

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається  
Завідувач кафедри  
\_\_\_\_\_ Скарга-Бандурова І.С.  
« \_\_\_\_ » \_\_\_\_\_ 20\_\_ р.

**МАГІСТЕРСЬКА РОБОТА**

НА ТЕМУ:

**СИСТЕМА НАВІГАЦІЇ ІНТЕЛЕКТУАЛЬНИХ МОБІЛЬНИХ РОБОТІВ**

---

Освітньо-кваліфікаційний рівень “Магістр”  
Спеціальність 122 – “Комп'ютерні науки”

Науковий керівник роботи:

\_\_\_\_\_ (підпис)

Є. В. Щербаков

\_\_\_\_\_ (ініціали, прізвище)

Консультант з охорони праці:

\_\_\_\_\_ (підпис)

Я. О. Критська

\_\_\_\_\_ (ініціали, прізвище)

Студент:

\_\_\_\_\_ (підпис)

Є.В. Болтов

\_\_\_\_\_ (ініціали, прізвище)

Група:

\_\_\_\_\_ КН-17Дм

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки  
Кафедра Комп'ютерних наук та інженерії  
Освітньо-кваліфікаційний рівень магістр  
Напрямок підготовки \_\_\_\_\_  
(шифр і назва)  
Спеціальність 122 – «Комп'ютерні науки»  
(шифр і назва)

**ЗАТВЕРДЖУЮ:**

Завідувач кафедри \_\_\_\_\_  
I.C. Скарга-Бандурова  
« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ р.

**З А В Д А Н Н Я  
НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ**

Болтову Єгору Владиславовичу  
(прізвище, ім'я, по батькові)

1. Тема роботи Система навігації інтелектуальних мобільних роботів

керівник проекту (роботи) Щербаков Є. В., к.т.н., доцент  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від " " 2019 р. № \_\_\_\_\_

2. Термін подання студентом роботи \_\_\_\_\_

3. Вихідні дані до роботи Мобільний робот (гексапод) здатний до самостійного пересування за допомогою нової інтелектуальної системи навігації

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) \_\_\_\_\_

1) Аналіз існуючих систем навігації;

2) Проектування інтелектуальної системи навігації (ІСН);

3) Створення дослідного зразка з використанням спроектованої ІСН;

4) Питання охорони праці, екології.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада Консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Критська Яна Олександрівна		

7. Дата видачі завдання \_\_\_\_\_

Керівник \_\_\_\_\_

(підпис)

Завдання прийняв до виконання \_\_\_\_\_

(підпис)

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту ( роботи )	Примітка
1	Аналітичний огляд літератури за темою роботи	1.09.18 – 2.10.18	
2	Аналіз існуючих систем навігації	1.09.18 - 2.10.18	
3	Проектування інтелектуальної системи навігації	3.10.18 – 10.10.18	
4	Розробка гексапода (апаратна частина)	11.10.18 – 25.10.18	
5	Розробка гексапода (програмна частина)	26.10.18 – 26.11.18	
6	Розгляд питань охорони праці та основних напрямків їх дотримання	14.11.18 – 15.12.18	
7	Оформлення пояснювальної записки	22.12.18 – 29.12.18	
8	Оформлення презентації роботи	30.12.18 – 9.01.19	

**Студент**

\_\_\_\_\_ ( підпис )

\_\_\_\_\_ (прізвище та ініціали)

**Керівник**

\_\_\_\_\_ ( підпис )

\_\_\_\_\_ (прізвище та ініціали)

## АННОТАЦІЯ

Болтов Є.В. Система навігації інтелектуальних мобільних роботів в недетермінованих середовищах.

В роботі проведено аналіз існуючих систем навігації та запропоновано, розроблено і впроваджено нову інтелектуальну систему навігації, яка дозволяє скоротити час на виявлення перешкод під час руху. В роботі запропоновано використання компактного одноплатного комп'ютера та сучасної нейромережі, які на сьогоднішній день набирають дедалі більшої популярності.

**Ключові слова:** RASPBERRY PI, SHARP IR, LSC-24, RASPICAM, RTOS, XENOMAI, HEXAPOD, (E)OGMANEO

## АННОТАЦИЯ

Болтов Е.В. Система навигация интеллектуальных мобильных роботов в недетерминированных средах.

В работе проведен анализ существующих систем навигации и было предложено, разработано и внедрено новую интеллектуальной системы навигации, которая позволяет сократить время на выявление препятствия во время движения. В работе предлагается использование компактного одноплатного компьютера и современной нейросети, которые на сегодняшний день набирают все большую популярность.

**Ключевые слова:** RASPBERRY PI, SHARP IR, LSC-24, RASPICAM, RTOS, XENOMAI, HEXAPOD, (E)OGMANEO

## ABSTRACT

Boltov E.V. System navigation of intelligent mobile robots in non-deterministic environments.

In this paper, an analysis of existing navigation systems was carried out and a new intelligent navigation system was proposed, developed and implemented, which allows reducing the time for identifying obstacles during movement. The paper proposes the use of a compact single board computer and a modern neural network, which today are gaining increasing popularity.

**Keywords:** RASPBERRY PI, SHARP IR, LSC-24, RASPICAM, RTOS, XENOMAI, HEXAPOD, (E) OGMANEO

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ .....	7
ВСТУП .....	8
1 АНАЛІЗ ТА ПОСТАНОВКА ЗАДАЧІ.....	9
1.1 Аналіз технічного завдання.....	9
1.2 Аналіз предметної області.....	9
1.3 Аналіз існуючих систем навігації.....	11
1.3.1 Лазерні та ультразвукові датчики відстані .....	11
1.3.2 Одометрія .....	13
1.3.3 Інерційна навігація.....	15
1.3.4 Навігація за допомогою орієнтирів .....	15
1.3.5 Лінійна навігація .....	16
1.3.6 Картографічне позиціонування.....	16
1.4 Дослідження нейромережі (E)OgmaNeo .....	18
1.5 Оптимізація за допомогою Xenomai.....	21
1.6 Висновки до розділу 1 .....	25
2 АПАРАТНА ЧАСТИНА.....	26
2.1 Механічна модель .....	26
2.2 Сервоприводи .....	27
2.3 Сервоконтроллер .....	30
2.4 Одноплатний комп'ютер .....	31
2.5 Камера .....	34
2.6 IR датчик відстані.....	35
2.6.1 I2C АЦП на ADS1115 .....	36
2.7 Електроживлення .....	37
2.7.1 Акумулятор .....	37
2.7.2 DC-DC понижуючий конвертер 3А.....	39

	5
2.7.3 Вибір силових кабелів.....	40
2.8 Функціональна схема .....	41
2.9 Висновки до розділу 2 .....	42
3 ПРОГРАМНА ЧАСТИНА .....	43
3.1 Підготовка OS. Патч ядра.....	43
3.2 Дистанційне керування (сервер).....	44
3.2.1 Створення точки доступу .....	44
3.2.2 Програмний код сервера.....	45
3.3 Дистанційне керування (клієнт) .....	47
3.3.1 Графічний інтерфейс.....	47
3.3.2 Програмний код клієнта .....	53
3.4 Робота з протоколом зв'язку LSC-24 v1.2 .....	60
3.5 Tripod gait.....	63
3.6 Модуль роботи з нейромережею .....	64
3.7 Загальна схема програмного комплексу .....	66
3.8 Висновки до розділу 3 .....	66
4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ. ЕКОЛОГІЯ .....	67
4.1 Загальні питання з охорони праці.....	67
4.1.1 Правові та організаційні основи охорони праці.....	67
4.1.2 Організаційно-технічні заходи з безпеки праці .....	68
4.2 Аналіз стану умов праці .....	68
4.2.1 Вимоги до приміщень .....	68
4.2.2 Вимоги до організації місця праці.....	69
4.2.3 Навантаження та напруженість процесу праці.....	70
4.3 Виробнича санітарія.....	71
4.3.1 Аналіз небезпечних та шкідливих факторів при виробництві виробу .....	71
4.3.2 Пожежна безпека.....	73
4.3.3 Електробезпека.....	73

	6
4.4 Гігієнічні вимоги параметрів виробничого середовища .....	74
4.4.1 Мікроклімат .....	74
4.4.2 Освітлення.....	75
4.4.3 Вентилювання.....	77
4.5 Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій .....	77
4.5.1 Розрахунок захисного заземлення .....	79
4.6 Охорона навколишнього природного середовища .....	83
4.6.1 Загальні дані з охорони навколишнього середовища .....	83
4.6.2 Вимоги до збору, пакування та розміщення відходів ІТ галузі.....	84
4.6.3 Визначення впливу та заходів щодо поводження з відходами ІТ галузі.....	84
4.7 Висновки до розділу 4 .....	85
ВИСНОВКИ.....	86
ПЕРЕЛІК ПОСИЛАНЬ .....	87
ДОДАТОК А.....	89
ДОДАТОК Б .....	95



**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ**

<b>RT</b>	–	реальний час
<b>RTS</b>	–	система реального часу
<b>RTOS</b>	–	операційна система реального часу
<b>HC</b>	–	навігаційна система
<b>GPS</b>	–	система глобального позиціонування
<b>IR</b>	–	інфрачервоний датчик
<b>PSD</b>	–	детектор чутливості положення
<b>IREDD</b>	–	діод інфрачервоного випромінювання
<b>PCB</b>	–	друкована плата
<b>ADC</b>	–	аналого-цифровий перетворювач
<b>DC</b>	–	постійний струм
<b>OS</b>	–	операційна система
<b>RPi2</b>	–	одноплатний комп'ютер Raspberry Pi 2 Model B
<b>MCU</b>	–	мікроконтролер
<b>CSI</b>	–	послідовний інтерфейс камери
<b>I2C</b>	–	послідовна асиметрична шина
<b>SPI</b>	–	послідовний периферійний інтерфейс
<b>UART</b>	–	універсальний асинхронний приймач/передавач
<b>USB</b>	–	універсальна послідовна шина
<b>TTL</b>	–	транзисторно-транзисторна логіка
<b>CCD</b>	–	прилад із зарядним зв'язком
<b>XML</b>	–	розширювана мова розмітки
<b>IDE</b>	–	інтегроване середовище розробки
<b>SDK</b>	–	набір засобів розробки, утиліт та документації
<b>TCP</b>	–	протокол керування передачею
<b>IP</b>	–	інтернет протокол
<b>IPC</b>	–	міжпроцесний зв'язок
<b>API</b>	–	готовий інтерфейс програмування

## ВСТУП

На сьогоднішній день мобільні роботи мають широкий спектр застосування, але існує цілий ряд завдань, при яких їх застосування ускладнене або неможливе. Важко організувати надійний канал зв'язку з роботом в наступних обставинах:

- Є ймовірність втрати зв'язку з роботом при його перебуванні в недоступному місці.
- Виконуване роботом завдання не припускає ручного управління (роботи-прибиральники, роботи-офіціанти, роботи-кур'єри, тощо).
- Конструкція робота настільки складна, що людина не здатна контролювати весь функціонал робота.

Для вирішення цих завдань активно ведуться розробки нових автономних мобільних роботів. Більшість існуючих підходів до управління роботами обмежено проблемою тупикових ситуацій, які виникають, коли робот рухається до бажаної мети без знання навколишнього середовища. Також, при наявності перешкод, розташованих близько до цільової точки, можливе виникнення циклічних рухів робота. В силу того, що такі роботи, повинні працювати в заздалегідь невідомих умовах, а орієнтуватися в зовнішньому середовищі на підставі значень великої кількості датчиків, застосування класичних математичних підходів до таких систем становиться неможливим. Таким чином, рішення проблеми обходу роботом перешкод є актуальною науково-технічною проблемою.

## 1 АНАЛІЗ ТА ПОСТАНОВКА ЗАДАЧІ

### 1.1 Аналіз технічного завдання

Метою роботи є підвищення ефективності навігації мобільного робота за допомогою використання інтелектуальної системи управління. Дані системи діють не за заздалегідь заданою програмою, а самостійно приймають рішення на підставі інформації про навколишнє середовище і закладених в них знаннях.

Інтелектуальна система має такі відмінні риси:

- Володіє знаннями, які визначають її поведінку в складній і заздалегідь невідомій ситуації;
- Механізм логічного висновку вихідних сигналів на основі вхідних даних і знань, закладених в систему;
- Механізм навчання або самонавчання, який дозволяє поповнювати базу власних знань.

Досягнення поставленої мети вимагає вирішення наступних завдань:

- Аналіз існуючих систем навігації;
- Аналіз і дослідження відомих методів побудови інтелектуальних систем;
- Розробка алгоритмічно реалізованого методу обробки сенсорної інформації, для усунення тупикових ситуацій, зумовлених наявністю близького розташування перешкод;
- Розробка відповідного програмного забезпечення і загальних алгоритмів функціонування програмно-апаратного комплексу;
- Проведення експериментальних досліджень інтелектуальної системи управління переміщенням робота в середовищі з перешкодами.

### 1.2 Аналіз предметної області

Далеко не завжди умови навколишнього середовища дозволяють людині виконувати ту чи іншу дію безпосередньо. Це може бути робота з вибухонебезпечними матеріалами, отруйними речовинами, пожежогасіння і багато інших завдань. У таких ситуаціях на допомогу людині приходять мобільні роботи для використання у надзвичайних ситуаціях.

Мобільний робот – це робот, який має ряд сенсорів для сприйняття навколишнього середовища, ряд виконавчих пристроїв для впливу на середовище і систему управління, яка дозволяє роботу здійснювати цілеспрямовані і корисні дії.

За сферою застосування мобільні роботи можна поділити на 2 класи (рис. 1.1).

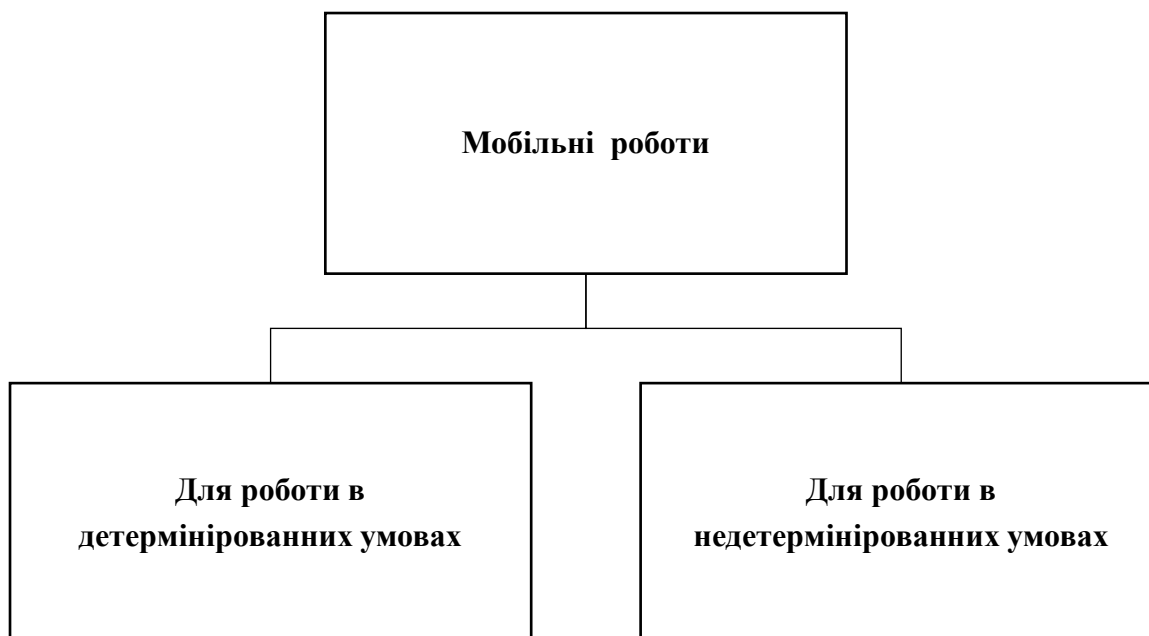


Рисунок 1.1 – Класифікація мобільних роботів за сферами застосування

Необхідно відзначити, що для роботи в недетермінованих умовах можуть використовуватися тільки мобільні роботи. Недетерміновані умови - це або недетерміновані середовища або недетерміновані процеси.

До недетермінованих середовищ пропонується відносити природні середовища і середовища, створювані аварійними ситуаціями, як в природних умовах, так і при руйнування будівель і споруд, спроектованих і створених людиною. До дій робота в природних середовищах відносяться дії при гірських розробках, польових сільськогосподарських роботах, лісопосадках, розвідці на місцевості, підводних і підземних роботах, визначеннях радіоактивної зараженості місцевості, військових діях під час розмінування і патрулювання, обслуговуванні сліпих як поводитирів на вулиці тощо.

До недетермінованих процесів відноситься кожен процес, перебіг і результат якого не залежить повністю від цілеспрямованої діяльності людини. Недетермінованими процесами є всі природні процеси, пожежі, вибухи, збирання врожаю сільськогосподарських культур, тощо.

За ступенем незалежності від участі людини-оператора всі мобільні роботи можуть бути класифіковані на дві великі групи:

- з автоматичним управлінням;
- з інтерактивним управлінням.

Мобільні роботи з автоматичним управлінням – це роботи, управління якими здійснюється системою управління без участі людини.

Для того щоб здійснювати дії без участі людини (зазвичай в недетермінованому, мінливому середовищі) роботи повинні володіти штучним інтелектом. У зв'язку з цим, на відміну від інших роботів, мобільні роботи з автоматичним управлінням в джерелах інформації зазвичай називаються автономними. Автономні мобільні роботи - це роботи, здатні самі (в залежності від сигналів від датчиків) приймати рішення в мінливій обстановці.

Роботи з інтерактивним управлінням слід визнати системами більш високого інтелектуального рівня, ніж автономні роботи, тому що можливості людського розуму на рівні прийняття рішення в критичних ситуаціях перевершують в даний час найрозвиненіші системи штучного інтелекту.

Мобільні роботи з інтерактивним управлінням - це роботи, які можуть працювати, і в автоматичному режимі, і управлятися людиною-оператором. На відміну від біотехнічних систем (систем з ручним керуванням), інтерактивні системи мають пристрої пам'яті для автоматичного виконання певних дій.

### **1.3 Аналіз існуючих систем навігації**

Системи навігації поділяються на 3 основні класи:

- глобальна НС - визначення абсолютних координат пристрою у процесі руху по довгих маршрутах;
- локальна НС - визначення координат пристрою по відношенню до деякої точки;
- персональна НС - позиціонування роботом частин свого тіла і взаємодія з сусідніми предметами.

Вважається, що чим крупніше мобільний робот, тим вище для нього важливість глобальної навігаційної системи і нижче - персональної.

Системи навігації класифікуються ще по одній ознаці - вони можуть бути пасивними і активними. Пасивна система навігації має на увазі прийом інформації про власні координати і інші характеристики свого руху від зовнішніх джерел, а активна розрахована на визначення місцеположення тільки за рахунок інтелектуальних властивостей мобільного робота. Як правило, глобальні навігаційні системи пасивні, локальні бувають і тими і іншими, а персональні - завжди активні.

#### **1.3.1 Лазерні та ультразвукові датчики відстані**

Одним із способів навігації є використання лазерних (IR) та ультразвукових датчиків відстані.

Ультразвукові датчики працюють за принципом відбиття звукових хвиль і використовуються для вимірювання відстані. Один датчик може виявити інші датчики, які працюють поблизу. Звукові хвилі випромінюються ультразвуковим датчиком, і вони відбиваються назад, якщо перед ним є об'єкт. Датчик виявляє ці хвилі і вимірює час, необхідний для передачі і прийому цих звукових хвиль. Потім відстань оцінюється часовим інтервалом між датчиком і об'єктом.

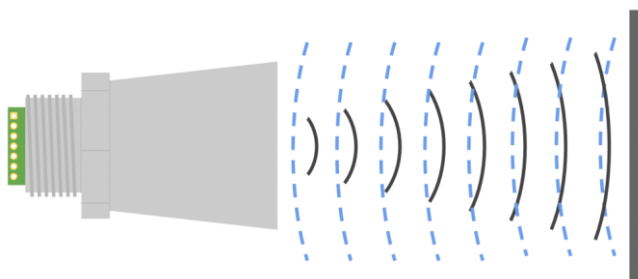


Рисунок 1.2 – Основна ілюстрація сонара. Перетворювач генерує звуковий імпульс, а потім прослуховує відлуння.

Інфрачервоні датчики працюють за принципом відбиття світлових хвиль та також можуть використовуватися для вимірювання відстані. Відбите світло виявляється, і потім обчислюється оцінка відстані між датчиком і об'єктом.

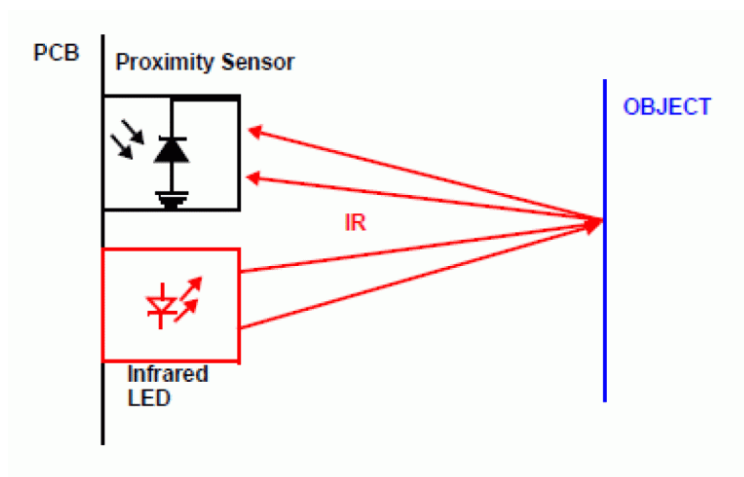


Рисунок 1.3 – Принцип роботи інфрачервоного датчика.

Основним недоліком лазерного датчика є те, що лазерний промінь допоможе отримати образ середовища тільки в зоні прямої видимості. Крім того, на шляху променю часто виникають дрібні перешкоди, що вносять погрішність в такий образ. А ультразвукові датчики

характеризуються великим часом відгуку, порядку десятих часткою секунди, що не дозволяє роботу переміщатися швидко. Швидкість звуку в різних умовах також може "плавати", впливаючи на точність оцінки відстані, в результаті в "голові" робота спотворюється загальна картина навколишнього середовища. Створення 3D-карт за допомогою лазерів в масштабі реального часу ще більш трудомістке і, як мінімум, вимагає значної обчислювальної потужності. По цих причинах цінність інформації, що поступає від бортових датчиків, невелика. Роботу необхідно перевести її в формальний і структурований "словесний" опис світу, що становиться задачею розпізнавання.

### 1.3.2 Одометрія

Одометрія – це спосіб оцінки переміщення за допомогою даних, отриманих із сенсорів руху. (вимірювання пройденого шляху). Одометрія застосовується при створенні роботів, якщо вони рухаються за допомогою ніг чи коліс для визначення їхньої поточної позиції відносно стартової.

Цей метод дуже чутливий до помилок через інтеграцію вимірювань швидкості впродовж всього часу оцінки позиції. Необхідні швидкі і точні збір та обробка даних, а також калібровка обладнання для ефективного використання одометрії [1].

Сьогодні, більшість роботів, що орієнтуються на місцевості, покладаються на одометрію як на основу навігаційної системи. Звичайний одометричний вимірювач включає в себе оптичні кодувальники, спарені з осями, що обертаються.

На сьогоднішній день існують наступні обертальні сенсори, що вимірюють переміщення і швидкість:

- енкодери з щітковими контактами;
- потенціометри;
- оптичні енкодери;
- магнітні енкодери;
- індуктивні енкодери;
- ємкісні енкодери.

У основі сучасних оптичних сенсорів лежить зменшений сенсор, що визначає близькість по перериванню променя. У ньому сфокусований і направлений на певний фотодетектор промінь світла періодично уривається диском зі спеціальними прорізами, що обертається на валу.

Існує два основних типи оптичних енкoderів:

- інкрементний - вимірює швидкість обертання і може визначити відносне положення;
- абсолютний - вимірює точне кутове положення і може визначити швидкість.

Одноканальний тахометр - найпростіший вигляд інкрементного енкодера. У основі механіки - дискретне джерело світла, пульсуюче певну кількість разів за один оберт валу. Збільшення кількості імпульсів за оберт збільшує дозвіл енкодера (і його вартість). Цей пристрій добре підходить як вимірювач швидкості із зворотним зв'язком в середньо і високошвидкісних системах управління. Але у них з'являються проблеми з перешкодами і стабільністю на малих швидкостях через помилки дискретизації. До цих проблем додається те, що одноканальний тахометр не здатний визначити напрям обертання і, як наслідок, не може бути використаний як позиціонуючий сенсор.

Абсолютні оптичні енкодери звичайно використовуються в додатках з повільним обертанням, для яких не допустима втрата інформації про положення через тимчасову втрату живлення. Найкраще підходить для систем з повільним і/або нечастим обертанням, в яких кодується кут повороту (протилежність обчисленням, пов'язаним з тривалим високошвидкісним обертанням, необхідні для обчислення переміщення по шляху проходження).

Потенційна незручність абсолютного енкодера - паралелізм вихідних даних, який вимагає більш складний інтерфейс через велику кількість проводів.

Одометрія дає хорошу короточасну точність, недорога і володіє дуже великою частотою дискретизації. Але початкова ідея одометрії - об'єднання рухової інформації, що збільшується у часі, яка неминуче приводить до накопичення помилок. На практиці, помилки орієнтації, що накопичуються є причиною більшості помилок позиціонування, кількість яких збільшується пропорційно шляху, пройденому роботом. Однак широко прийнято, що одометрія - дуже важлива частина навігаційної системи робота і задача навігації спростилася, якщо точність одометрії була б збільшена.

Декілька причин, по яким одометрія використовується в мобільних роботах:

- дані одометрії можуть бути об'єднані технологією абсолютного позиціонування (і іншими технологіями) для отримання кращої і більш точної оцінки положення;
- одометрія може бути використана в абсолютному позиціонуванні, поліпшеному орієнтирами (маяками) на місцевості. Даючи необхідну точність позиціонування, підвищуючи точність одометрії - це дозволяє зменшити частоту оновлень в абсолютному позиціонуванні. Як наслідок - для даного маршруту потрібно менше маяків;
- в деяких ситуаціях одометрія застосовна тільки як навігаційний інформатор.



### 1.3.3 Інерційна навігація

Альтернативний метод одометрії – інерційна навігація. Принцип роботи полягає у безперервному отриманні навіть найменшого прискорення по кожній з трьох осей напрямів і переміщення у часі, щоб обчислити і положення. Платформа сенсора стабілізується гіроскопом, це необхідне для збереження суворої орієнтації трьох акселерометрів протягом всього процесу.

Хоч концепція методу проста, специфіка реалізації вельми вимоглива. Головним чином це викликано помилками, причиною яких є стабільність (її відсутність), для забезпечення якої, щоб гарантувати коректність обчислення положення, використовуються гіроскопи [2].

Однією з переваг інерційної навігації є можливість забезпечувати швидкі, низько латентні динамічні вимірювання.

Однак головним недоліком є те, що кутові і швидкісні дані повинні бути включені один і два рази (відповідно) для визначення орієнтації і лінійного позиціонування (відповідно).

### 1.3.4 Навігація за допомогою орієнтирів

Навігація за допомогою орієнтирів може відбуватися за допомогою пошуку як штучних орієнтирів, так і природніх.

Головна проблема навігації по природніх орієнтирах полягає у визначенні і зіставленні характерних особливостей в даних, отриманих від сенсорів. Такі сенсори являють собою машинний зір. Більшість систем машинного зору засновані на визначенні довгих відрізків прямих, наприклад як в дверних отворах або точок з'єднання стін і стелі.

У системі позиціонування по природніх орієнтирах виділяють наступні базові компоненти:

- сенсор (звичайно зоровий) визначаючий орієнтири на сцені;
- метод порівняння, отриманих внаслідок спостереження, особливостей з картою відомих орієнтирів;
- метод обчислення місцеположення і локалізації помилок від порівнянь.

Набагато легше визначити штучні орієнтири, оскільки вони розробляються з певною специфікою. Додатково, для штучного орієнтира заздалегідь відомі точні розміри і форма. Багато які системи позиціонування по штучних орієнтирах засновані на машинному зорі, а як орієнтири частіше за все використовуються чорний прямокутник з білими точками по кутах, сфера з вертикальними і горизонтальними колами для калібрування, що дозволяє визначити просторове положення по одному зображенню.

Точність описаного вище методу залежить від того, з якою точністю геометричні параметри орієнтирів будуть витягнуті із зображення сцени, яка, в свою чергу, залежить від відносного положення і кута між роботом і орієнтиром.

### 1.3.5 Лінійна навігація

Ще один вид навігації по орієнтирах, що широко використовується в індустрії – це лінійна навігація. Вона може розглядатися як навігація по безперервних орієнтирах, а через те, що в більшості випадків сенсор, що використовується в системі, повинен знаходитися дуже близько до лінії, габарити пристрою обмежені тим, що воно повинно знаходитися в безпосередній близькості від лінії. Ця технологія довгі роки використовувалася в задачах промислової автоматизації, але вона не була детально вивчена і, як наслідок, не дозволяла пристрою рухатися вільно.

Основні реалізації лінійної навігації:

- електромагнітне управління;
- управління відбивною або оптичною стрічкою;
- феритове управління, де використовується феромагнітний пил;
- управління по термальним маркерах.

Основні особливості навігації по орієнтирах:

- навігація по природних маяках вимоглива до постійності навколишнього оточення;
- навігація по штучних маяках;
- максимальна відстань між роботом і орієнтиром значно менше, ніж в системах з активними маяками;
- точність позиціонування залежить від відстані і кута між роботом і орієнтиром;
- необхідна велика обчислювальна потужність, чим в системах з активними маяками;
- зовнішні умови (такі як освітленість) можуть бути причиною помилок. Наприклад: орієнтир не може бути розпізнаний, або деякий об'єкт помилково прийнятий за орієнтир;
- в навігації по орієнтирах потрібно, щоб б робот знав своє зразкове початкове положення для того, щоб він знав де шукати орієнтири;
- база даних маяків і їх розташування в просторі повинна весь час підтримуватися.

### 1.3.6 Картографічне позиціонування

Картографічне позиціонування - це технологія, по якій робот використовує сенсори для побудови локальної карти місцевості. Ця локальна карта потім порівнюється з глобальною,

заздалегідь збереженою в пам'яті. Після знаходження збігів робот обчислює своє поточне положення і орієнтацію на місцевості.

Основні переваги картографічного позиціонування приведені нижче:

- використовується на місцевості зі структурою типовою для приміщення і отримує інформацію про положення в оточенні, не змінюючи його;
- може бути використана для створення і оновлення карти місцевості. Карти місцевості грають велику роль при глобальному плануванні шляху;
- дозволяє роботу вивчити нову місцевість і підвищує точність позиціонування при її дослідженні.

Недоліки картографічної навігації пов'язані з наступними вимогами:

- на місцевості повинна бути достатня кількість стаціонарних, добре помітних деталей;
- мобільний робот повинен мати значні обчислювальні потужності.

Уявлення, що використовується для карти, повинно забезпечувати можливість об'єднання на карті нової інформації, що поступає від сенсорів. Також воно повинно доставляти необхідну інформацію для планування маршруту та обходу перешкод.

Три головних кроки в обробці сенсорних даних для побудови карти:

- видобування характерних ознак з необроблених сенсорних даних;
- об'єднання даних від сенсорів різних типів;
- автоматизоване створення абстрактної моделі місцевості.

Один з самих енергоємних аспектів картографічної навігації - зіставлення карт. Зіставлення відбувається при первинному видобуванні характерних ознак, далі визначається точна відповідність між зображенням і характеристиками моделі. Робота по зіставленню карт в сфері машинного зору частіше за все фокусується на загальній проблемі зіставлення зображення, отриманого з випадкового положення і орієнтації, по відношенню до моделі.

Алгоритми порівняння можна розділити на алгоритми, засновані на аналізі зображення і засновані на аналізі характерних ознак. Перші відрізняються від других тим, що, по-перше, по карті зіставити дані дуже інформативної точки на місцевості простіше, ніж дані що являють собою малий набір особливостей. Обчислення в алгоритмах заснованих на аналізі характерних ознак швидше, ніж в алгоритмах заснованих на аналізі зображення і не вимагають хороших попередніх головних обчислень. Обчислення в алгоритмах заснованих на аналізі зображення можуть керуватися не ідеальною моделлю місцевості і є більш точними, а також можуть виконуватися на меншій кількості точок, ніж потрібно для обчислення в алгоритмах заснованих на аналізі характерних ознак.

Як і в навігації по орієнтирах, вигідно використати приблизне обчислення положення, засноване на одометрії, для створення зразкової візуальної сцени (по карті, що є), яку буде

«бачити» робот. Далі, ця створена сцена порівнюється з тим, що зараз бачить робот. Ця процедура ефективно знижує час необхідний для знаходження відповідності.

У картографічному позиціонуванні виділяють два загальних способи представлення карт: геометричне і топологічний. На геометричній карті об'єкти представляються відповідно до їх абсолютних геометричних відносин. Це може бути сіткова карта або більш абстрактна лінійна або полігональна карта. Топологічний підхід більше базується на протоколюванні геометричних відносин між вистеженими особливостями, аніж на їх абсолютне положення в координатах відносно деякої системи відліку. Як результат, цей підхід може використовуватися для інтеграції карт великих територій, оскільки всі зв'язки між вузлами швидше відносні, чим абсолютні.

#### 1.4 Дослідження нейромережі (E)OgmaNeo

Створення інтелектуального мобільного робота потребує використання високопродуктивної нейромережі, яка дозволить використовувати її на одноплатному комп'ютері. Тому було прийнято рішення використовувати сучасну нейромережу з онлайн навчанням - (E)OgmaNeo [3].

(E)OgmaNeo являє собою бібліотеку на мові програмування C++, яка має безпосередньо 2 версії: EOgmaNeo та OgmaNeo. Обидні бібліотеки працюють за одним і тим самим алгоритмом. Відмінність полягає у тому, що EOgmaNeo заточена під використання на вбудованих системах (ARM, одноплатні комп'ютери), тому вона використовує тільки потужності CPU, а інша бібліотека (OgmaNeo) використовує потужності GPU (заточена під пристрої з OpenCL).

Особливості:

- Впроваджені розріджені прогностичні ієрархії (SPH).
- Підтримка створення world-model
- Повністю онлайн/інкрементне навчання

У порівнянні з Deep Learning, SPH мають наступні відмінні риси:

- Не використовує зворотне поширення
- Має біологічну-подібність
- Місцеві розріджені властивості
- Моделі динамічних систем через зміну розріджених розподілених уявлень (SDR) мають множину напрямків (як просторові, так і тимчасові)

У порівнянні з НТМ, бібліотека OgmaNeo має наступні переваги:

- Представлення часу через експоненціальну пам'ять, на відміну від послідовної у НТМ.
- Використання додаткових спрощених моделей нейронів
- Прогнозування через декодування
- Використання ієрархії
- Використання безперервних синапсів
- Синапси не виростають під час виконання, а створюються відразу (менше виділення/звільнення) пам'яті

Нейронний шар OgmaNeo (рис. 1.4) має наступну структуру:

- Шари являють собою 3D сітки клітин
- Кожен шар має ширину, висоту та розмір стовпця, таким чином це являє собою 2D сітку стовпців
- Один шар приймає деякий вхід, створює приховане представлення **H** (у 3D сітці стовпців), а потім знову відображається на вході, але вже на один часовий крок попереду.
- Шар виконує відображення  $X_t \rightarrow H \rightarrow X_{(t+1)}$ , де **H** приховане зображення стовпців
- Шар також має зворотний зв'язок з більш вищими шарами (використовується для уточнення прогнозів). Таким чином, кожен шар має інформацію, що проходить в обох напрямках, вгору і вниз
- Вхідні дані та прогнозування знаходяться на «дні», прихований стан і зворотний зв'язок знаходяться на «вершині»
- Є вагові коефіцієнти передачі ( $X_t \rightarrow H$ )
- Є ваги зворотного зв'язку ( $H, F \rightarrow X_{(t+1)}$ ), де F - зворотній стан

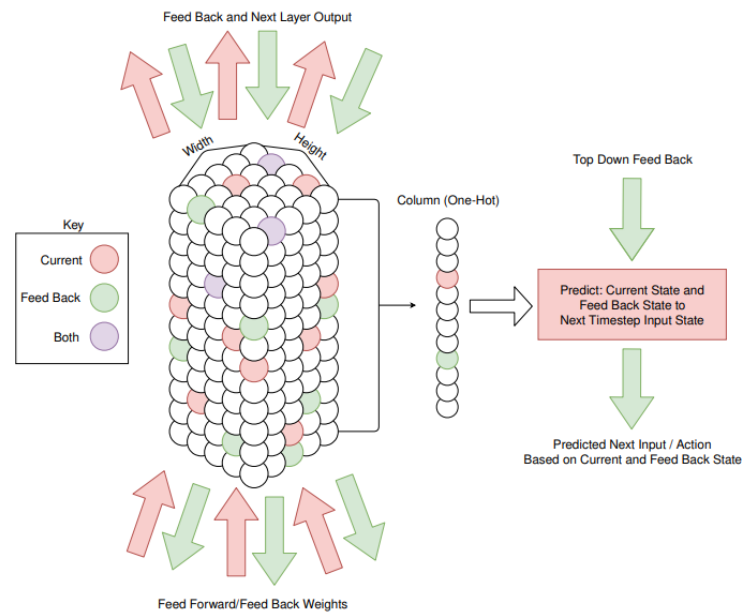


Рисунок 1.4 – Структура нейронного шару OmniaNeo (5x5x12).

Так як  $\mathbf{H}$  є бінарним, то більшість розріджених алгоритмів кодування вимагають модифікації для роботи. Таким чином використовуються найкращий алгоритм – колонний бінарний розріджений код (CBSC), який представлено на рисунку 1.5.

```

input : input states  $I$ , weight matrix  $W$ 
output: sparse binary columnar cell states  $O$ 
 $A \leftarrow 0$  // For each column cell, initialize activation to 0
 $R \leftarrow 0$  // For each input, initialize reconstruction to 0
for numIters do
    // Determine stimulus  $S$ 
     $S \leftarrow W^T \cdot (I - R)$ 
    // Add on to activations
     $A \leftarrow A + S$ 
    // Inhibit activations to determine states
     $O \leftarrow \text{ColumnWiseOneHot}(A)$  // One active unit per column
    // Reconstruct
     $R \leftarrow W \cdot O$ 
end
return  $O$ 

```

Рисунок 1.5 – Алгоритм CBSC.

## 1.5 Оптимізація за допомогою Xenomai

Використання нейромережі та різноманітних програмних модулів дає значне чисельне навантаження, а так як планується циклічне виконання, то визначення перешкоди без затримок являє собою дуже важливу задачу. Так, якщо мобільний робот не виявить вчасно перешкоду, то відбудеться зіткнення і це може мати негативні для робота наслідки. Тому пропонується модернізація OS одноплатного комп'ютера за допомогою використання системи реального часу.

Системи жорсткого реального часу мають жорсткі вимоги щодо часу, щоб відповідати термінам або може відбутися нерегулярна поведінка. Рисунок 1.6 показує обмеження часу в жорсткій системі реального часу (RTS) з використанням функції часу-корисності. Приклади систем жорсткого реального часу:

- Система автопілота літака. Коли автопілот відчуває потенційне зіткнення з сусіднім об'єктом то він змінює напрямок руху літака негайно.

Деякі з основних вимог жорстких систем реального часу:

- мінімальна затримка під час перемикання завдань
- мінімальний джиттер
- завершення виконання
- пріоритетна багатозадачність
- пріоритетне успадкування
- принципово відповідність строгим термінам

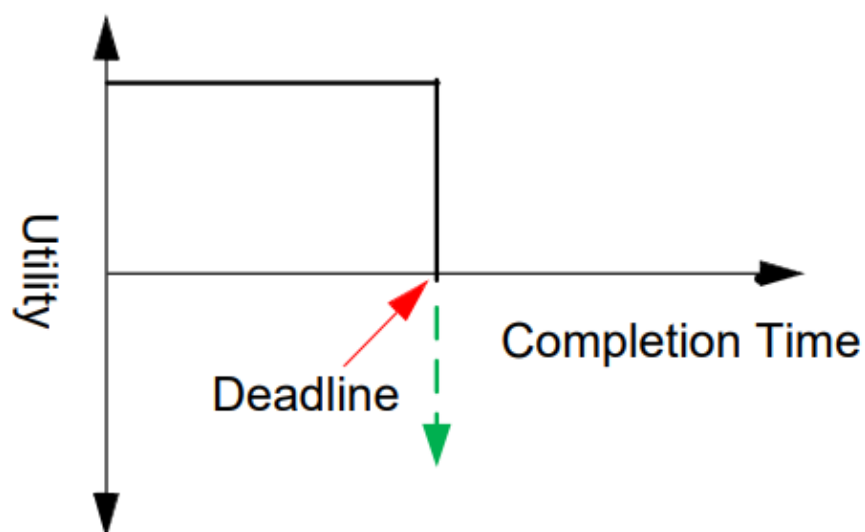


Рисунок 1.6 – Обмеження часу у жорсткій RTS

Для м'яких систем реального часу також є важливим дотримання термінів, але відсутність дотримання може бути не катастрофічною. На рис. 1.7 показані тимчасові обмеження у м'якій системі реального часу з використанням функції часу-корисності. Програмне забезпечення пропускає терміни. Прикладом м'якої системи реального часу є програвач компакт-дисків, який, у випадку вилучення, може затримати фактичний час відповіді на запит вилучення на декілька мілісекунд.

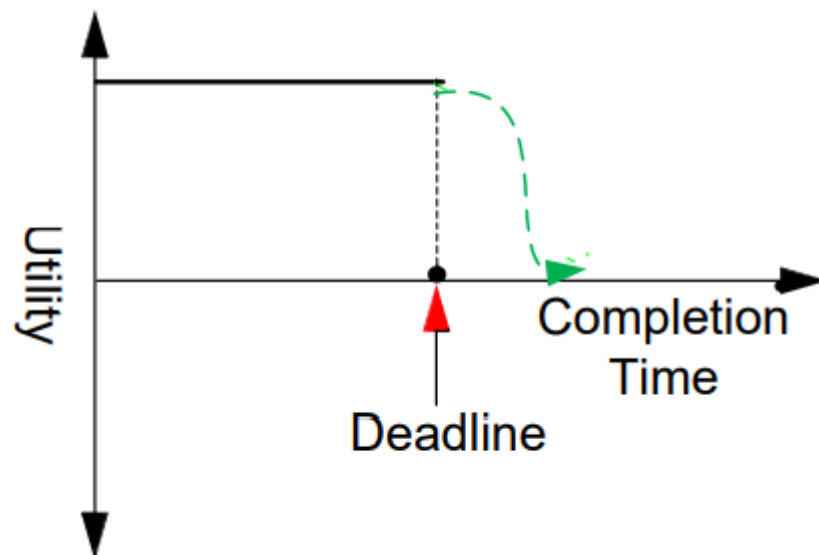


Рисунок 1.7 – Обмеження часу у м'якій RTS

Так як у роботі планується використання одноплатного комп'ютера на базі OS Linux, то слід розглянути її з точки зору RT. Linux не є жорсткою операційною системою реального часу, оскільки не гарантує виконання задач у поставлені строки. Ядро може призупинити виконання задачі під час її роботи і вона може залишатися призупиненою на будь-який довгий час (наприклад, при обслуговуванні переривання).

Основна ідея створення стандартного жорсткого середовища Linux у реальному часі полягає в тому, що невелике високоякісне ядро реального часу працює між апаратним і стандартним Linux. Завдання в реальному часі виконуються цим ядром реального часу (від виконання до завершення) і звичайні процеси Linux припиняються протягом цього терміну. У режимі реального часу планувальник ядра реального часу розглядає стандартне ядро Linux як простий процес, яке, коли їм дають можливість запускувати, виконує свій власний планувальник графік нормальних процесів Linux. Але так як ядро реального часу працює на більш високому пріоритеті, звичайні процеси Linux можуть бути в будь-який час витіснені завданням в реальному часі. Управління переривання є іншим фактором, яким керується ядро реального часу. Коли переривання запускається під час виконання завдання в реальному часі, воно



спочатку приймається ядром реального часу і зберігається. Коли ядро реального часу робиться, переривання передається стандартному Linux ядро. Якщо є пов'язаний обробник реального часу для переривання, він є виконується ядром реального часу. Інакше, якщо немає більше реального часу виконання завдань, збережене переривання передається на звичайний Linux. Інший

Механізми використовуються для передачі переривань від ядра реального часу до нормального.

Таким чином пропонується використання фреймворку Xenomai.

Xenomai - це фреймворк для розробки додатку реального часу, пов'язаний з ядром Linux, для того, щоб забезпечити всеохоплюючу, з відкритим інтерфейсом реального часу, підтримку додатків, легко інтегрованих в Linux [4].

Проект Xenomai розпочався у серпні 2001 року. У 2003-му він був об'єднаний з проектом RTAI, щоб забезпечити вільну платформу промислового рівня для Linux, названу RTAI / fusion, на базі ядра Xenomai для абстрактної операційної системи реального часу. В кінцевому результаті проект RTAI / fusion став незалежним від RTAI в 2005 році під назвою Xenomai.

Щоб підтримувати жорсткі можливості реального часу для ядра Linux, Xenomai реалізує мікро-ядро між апаратним забезпеченням і ядром Linux (рис. 1.8). Це мікроядро відповідає за виконання жорстких завдань у реальному часі та переривання перехоплень, блокуючи їх від ядра Linux, отже перешкоджаючи ядрі Linux витікати жорстке мікроядро реального часу. По суті, ядро Linux виконується на тлі цього мікро-ядра як завдання з низьким рівнем пріоритету, який виконує завдання, що не є реальними. Xenomai забезпечує жорсткий механізм розробки в режимі реального часу з підтримкою в реальному часі для користувацьких космічних додатків і реалізує мікро-ядро з планувальником реального часу. Вона також надає різні API для створення RT завдань, таймерів, об'єктів синхронізації і т.д., і імітує різні API, які називаються шкінами (див. Xenomai Skins) для застосування в реальному часі, включаючи інтерфейс POSIX \*, RTAI, VxWorks \* і т.д., що дозволяє полегшення перенесення існуючих додатків RT до Linux. Xenomai використовує наноядро ADEOS для обробки диспетчеризації переривань у реальному часі. Додаткові відомості про Xenomai та їх використання для створення жорстких завдань у реальному часі див. Примітка: Є багато інших схем жорсткого реального часу, доступних для Linux (наприклад, RTLinux (Wind River \* Linux), RTAI і т.д., але цей документ охоплює виключно Xenomai). [5]

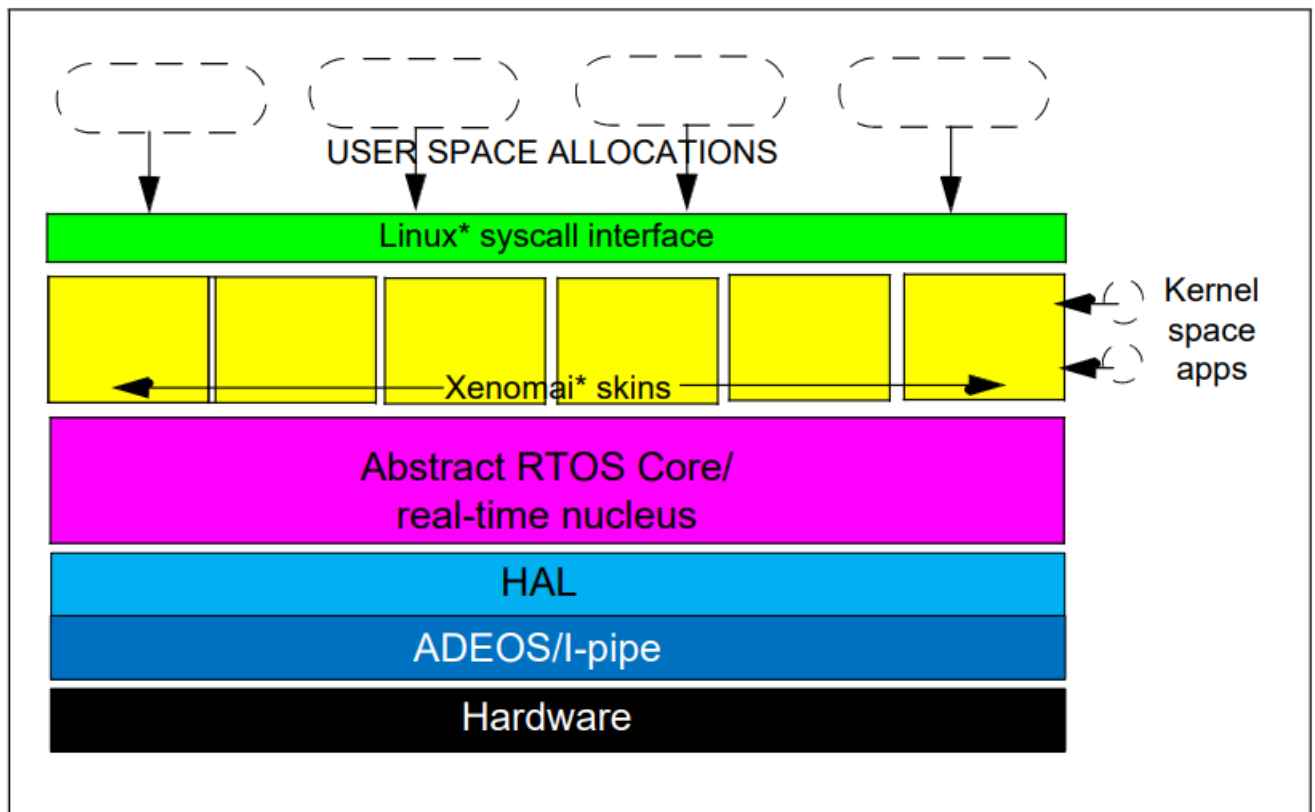


Рисунок 1.8 – Архітектура Xenomai

Ядро Xenomai - це абстрактна RTOS, яка надає ресурси операційної системи та загальні блоки для побудови різних інтерфейсів RTOS, які називаються шкінами. Ці шкіни імітують різні API RTOS, що дозволяє легко переносити існуючі програми на Xenomai. Ці блоки утворюють ядро Xenomai. Xenomai поставляється з наступними шкінами:

- POSIX
- pSOS
- RTAI
- uITRON
- VRTX
- VxWorks
- Native API Xenomai

## 1.6 Висновки до розділу 1

У цьому розділі були поставлені цілі для досягнення мети роботи, а саме підвищення ефективності навігації мобільного робота за допомогою використання інтелектуальної системи управління. Була спроектована інтелектуальна система навігація яка використовує камер та IR сенсор. Також були вивчені та проаналізовані основні методи та технології навігації мобільних роботів а також предметна область.

Відповідно до поставлених цілей була проаналізована оптимізація за рахунок використання фреймворку Xepomai, що дозволяє покращити результати роботи мобільного робота.

Наступним кроком роботи стане розробка апаратної частини роботи – дослідного зразка, відповідно до поставленого завдання.

## 2 АПАРАТНА ЧАСТИНА

У цьому розділі розглядається апаратна частина проекту. Відповідно до технічного завдання на розробку інтелектуальної системи управління, були обрані та розглянуті необхідні компоненти та модулі для реалізації дослідного зразка. Основними критеріями вибору стали вартість та доступність компонентів на ринку. Також у розділі представлена функціональна схема підключення та механічна конструкція апаратного комплексу.

### 2.1 Механічна модель

У якості механічної моделі було прийнято рішення використовувати гексапод. Гексапод – являє собою механічний транспортний засіб, який може пересуватися за допомогою шістьох ніг. Гексапод має велику гнучкість в тому, як він може рухатися, оскільки він може бути статично стабільним на трьох (і більше) ногах. Якщо ноги стають недієздатними, гексапод все одно може ходити. Крім того, не всі ноги робота необхідні для стабільності, тому вільні ноги у подальшому можна використовувати для маніпулювання корисним навантаженням.

Для створення гексапода використовувалися готові алюмінієві частини від компанії «Lobot», які представлені на рис. 3.1.



Рисунок 3.1 – Алюмінієві частини гексапода.

За допомогою представленого набору алюмінієвих частин можна зібрати гексапод с 3 ступенями свободи. З точки зору біологічної моделі ноги павука (рис. 3.2), 3 ступені свободи будуть відповідати за такі члени ніг як Coxa, Femur та Tibia. Такої кількості ступенів свободи досить для позиціонування ноги у трьохвимірному просторі та забезпечення достатньої свободи та плавності руху.

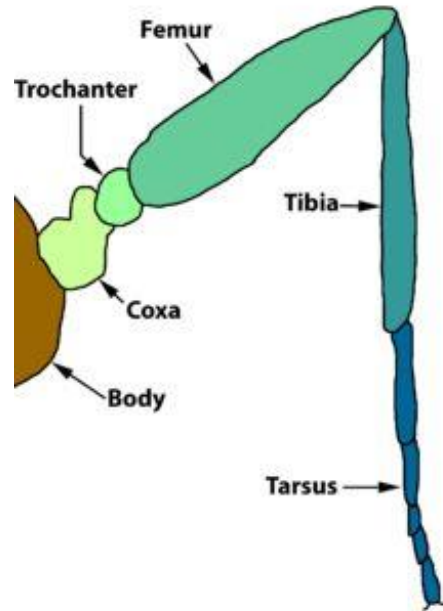


Рисунок 3.2 – Біологічна модель ноги павука.

## 2.2 Сервоприводи

Сервопривод — допоміжний двигун для дистанційного автоматичного керування або регулювання машин, апаратів, закриття і відкриття засувки, клапанів тощо. Сервомотори виділяються у окрему групу моторів у зв'язку з тим, що, як правило, діапазон роботи ротора такого мотора менше одного повного оберту. Сервопривод буває електричним, гідравлічним і пневматичним. Основні характеристики: маса, динаміка двигуна, рівномірність руху, енергоефективність [6].

Сервопривод працює від імпульсів змінної тривалості, які отримує через сигнальний дріт. Коли тривалість імпульсів становить близько 1,5 мілісекунди, то сервопривід перебуває в нейтральному положенні (тобто у нього однаковий потенціал обертання в обидва напрями). Кут повороту сервоприводу залежить від тривалості імпульсу. Чим триваліший імпульс, тим швидше працює двигун. Коли сервопривід виконує команду переміститися, то яка-небудь зовнішня сила, що при цьому спробує його спинити буде відчувати сильний спротив — це і є та

максимальна сила, яку витримуватиме сервопривід. Проте сервопривод не постійно підтримує вказане положення — для цього йому необхідні імпульси, на які він чекає протягом 20 мс. Що стосується тривалості імпульсу, то якщо вона менше 1,5 мілісекунд — сервопривід повертає вал на декілька градусів проти годинникової стрілки і намагається зафіксувати положення. Якщо ж вона більше, то навпаки — за годинниковою стрілкою. У середньому для роботи сервоприводу необхідний діапазон тривалості імпульсу від 1 мс до 2 мс. Крім того, важливий параметр, що характеризує сервопривід — це швидкість обертання (той час, за яке сервопривід переходить з одного положення в інше)

Головні частини сервоприводу — це його двигун, елементи керування і передача. Крім того, в ньому є також дрібніші периферійні пристрої — блокування, сигналізація, система включення/виключення, елементи зворотного зв'язку. Як правило, сервоприводи можуть працювати, на відміну від систем сельсин/давач — сельсин/приймач, тільки від зовнішніх джерел енергії, оскільки потужності внутрішніх джерел енергії недостатньо для ефективного функціонування сервоприводу (дуже вже енергоємну роботу йому доводиться виконувати).

Багато сервоприводів є обертові механізми, проте їх використовують для відтворення лінійного руху. У деяких з цих випадків, використовують датчики лінійних переміщень.[2] Ці сервоприводи запобігають неточностей у перетворенні обертового руху в лінійний під час переміщення об'єкта, проте їх дизайн став набагато складнішим, і їх набагато важче виготовляти у готовому вигляді, адже вони мають виконувати дуже специфічні завдання (тому до кінцевого «готового до застосування» вигляду їх підганяють у польових умовах).

Сервоприводи поділяються на кілька видів (залежно від джерела енергії) — механічні, гідравлічні і електричні. Електричні сервоприводи є найсучаснішими. Часто в них використовуються синхронні двигуни. Їхній обертовий момент і швидкість регулюються спеціальними сервоперетворювачами.

Сучасні сервоприводи характеризуються доброю спроможністю до регулювання, стабілізацією швидкості і обертового моменту, високою динамікою і точністю позиціонування, відносно невеликими вагою і габаритами, а також великою стійкістю до різноманітних перешкод.

Завдяки своїм високим технічним показникам, сервоприводи активно використовують у робототехніці. Їхнє використання допомагає підвищити функціональність роботів, зробити їхні рухи чіткішими і плавнішими.

Функціонування механічної моделі гексапода потребує 18 сервоприводів (3 ступеня свободи на кожному із 6 ніг). Таким чином були обрані бюджетні та надійні сервоприводи MG995 (зовнішній вигляд сервоприводу наведено на рис. 3.3).



Рисунок 3.3 – Сервопривод MG995.

MG995 - це найвідоміші сервоприводи від фірми «TowerPro», які являють собою цифрові металеві приводи з високим крутним моментом. Специфікація та технічні характеристики MG995 наведені у таблиці 3.1.

Таблиця 3.1 – Специфікація MG995.

Вага	55 g
Габарити	40.7×19.7×42.9mm
Крутний момент	9.4kg/cm (4.8V) 11kg/cm (6V)
Робоча швидкість	0.20sec/60degree (4.8V) 0.16sec/60degree (6.0V)
Робоча напруга	4.8 ~ 6.6V
Температурний діапазон	0 - 55deg
Споживання струму (холостий хід)	10mA
Робочий струм	170mA
Споживання струму (обертання заблоковано)	1200mA

## 2.3 Сервоконтроллер

Для того, щоб одночасно керувати усіма цифровими сервоприводами гексапода необхідно використовувати спеціалізовану плату, а саме – сервоконтроллер. Спираючись на поставлені вимоги та бюджет, було обрано сучасний 24 каналних сервоконтроллер від фірми «Lobot», представлений на рис. 3.4 [7].

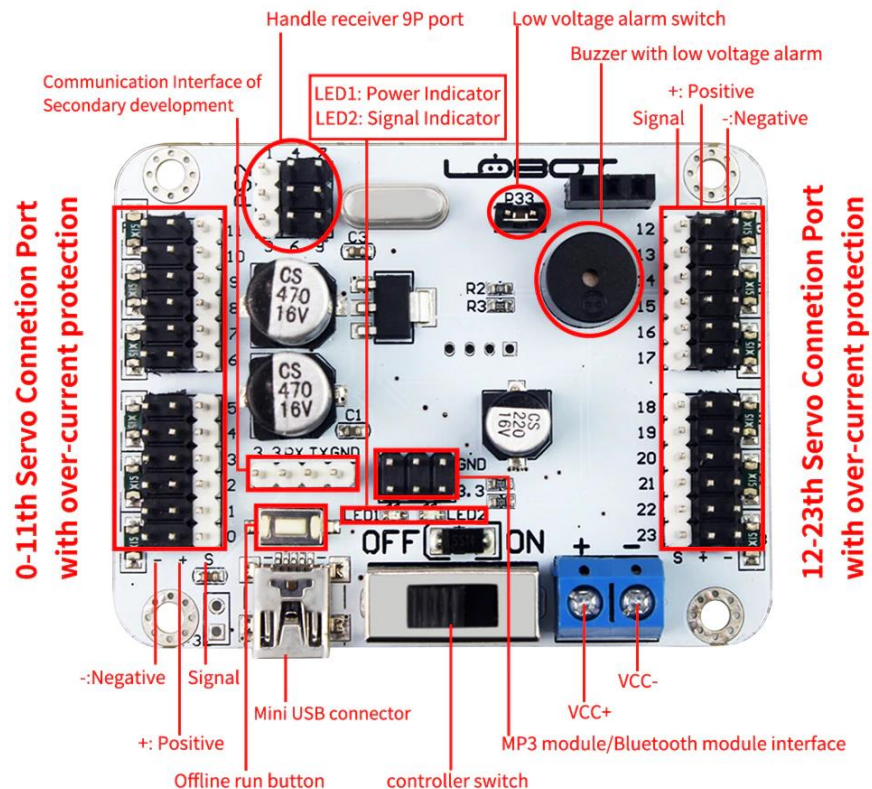


Рисунок 3.4 – Сервоконтроллер LSC-24 з позначенням портів.

Особливості сервоконтроллера:

- Процесор, що використовує високопродуктивний мікрокомп'ютер STM32 з однією мікросхемою ARM Cortex-M3, високоточне керування 24-каналами, швидкість регулювання;
- 16М пам'ять, яка може містити 230 груп дій, кожна група дій може містити до 510 дій;
- Всі 24-канальні інтерфейси сервосистеми мають захист від перевантаження, значно знижує ризик вичерпання сервоприводу через заблокований ротор;
- Програмне забезпечення для ПК, яке підтримує графічне онлайн налагодження; можливість керування 24-каналами одночасно;



- Сигналізація низької напруги. Якщо живлення становитиме нижче 5В, то сервоконтроллер автоматично сповістить про це звуковим сигналом;
- Підтримка TTL послідовного зв'язку, сервоконтроллер може бути з'єднаний з іншими чіпами, які можуть відправляти інструкції до контролера для досягнення інтелектуального управління (рис. 3.5). Якщо до сервоконтроллера надійдуть некоректні дані, то він оповістить про це звуковим сигналом, як і у випадку з низькою напругою.

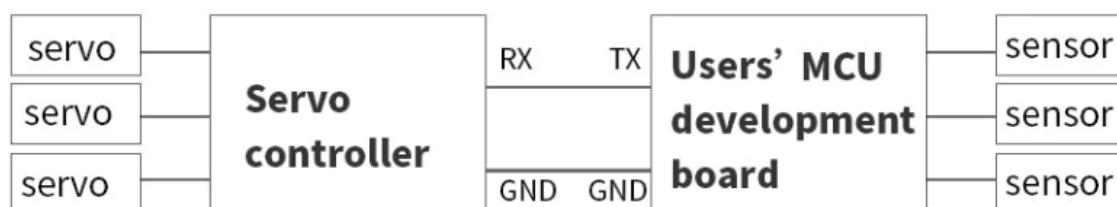


Рисунок 3.5 – Варіант можливого використання TTL сервоконтроллера.

Технічні характеристики сервоконтроллера наведені у таблиці 3.2.

Таблиця 3.2 – Специфікація LSC-24.

Вага	26 g
Розміри	65.0×50.0mm
Кількість каналів	24
Пам'ять	16M
Робоча напруга	5V
Напруга для сервоприводів	4.2-7.2V
CPU	32 Bits Baud rate: 115200 (USB), 9600 (UART)
CPU voltage	3.3V

## 2.4 Одноплатний комп'ютер

У якості керуючої плати було прийнято використовувати одноплатний комп'ютер Raspberry Pi2 Model B rev 1.1 (рис. 3.6).

Raspberry Pi – це найпопулярніші одноплатні комп'ютери для ентузіастів, які мають низьку ціну, відкритий характер розробки та орієнтацію на навчання та експерименти.

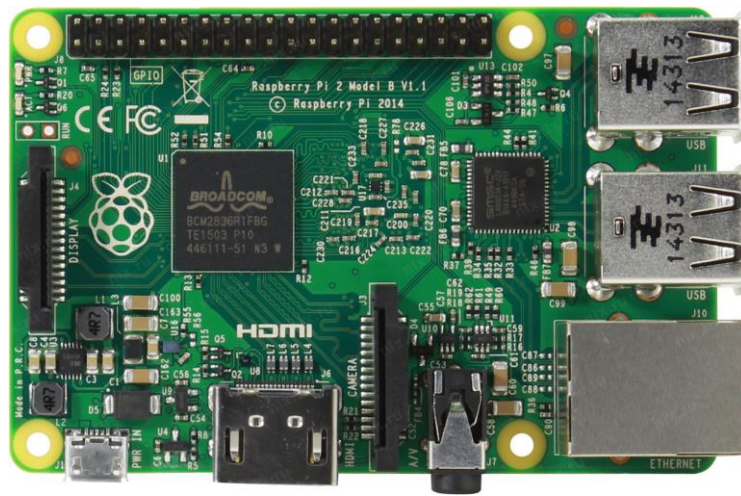


Рисунок 3.6 – Зовнішній вигляд Raspberry Pi 2.

Raspberry Pi 2B використовує 32-розрядний чотирьохядерний процесор ARM Cortex-A7 900 МГц та графічний процесор VideoCore IV (GPU). Він має кеш рівня 1 (L1) 16 Кб, а кеш рівня 2 (L2) 128 Кб. Кеш-пам'ять 2 рівня використовується насамперед графічним процесором. З повною специфікацією можна ознайомитися у таблиці 3.3 [8].

Таблиця 3.3 – Специфікація RPi2.

Вага	45 g
Розміри	85.6×56.5mm
SoC	Broadcom BCM2836 (CPU, GPU, DSP, SDRAM)
CPU:	900 MHz quad-core ARM Cortex A7 (ARMv7 instruction set)
GPU:	Broadcom VideoCore IV @ 250 MHz
Memory:	1 GB (shared with GPU)
Video input	15-pin MIPI camera interface (CSI) connector
Power rating	800 mA (4.0 W)
Power source	5 V via MicroUSB or GPIO header

На рис. 3.7 представлена діаграма, що показує розташування виводів RPi2 на інтегральній схемі та їх функції. Як можна побачити з діаграми, RPi2 має такі інтерфейси як I2C, SPI та UART.

Raspberry Pi 2 Model B (J8 Header)					
GPIO#	NAME			NAME	GPIO#
	3.3 VDC Power	1		2	5.0 VDC Power
<b>8</b>	GPIO 8 SDA1 (I2C)	3		4	5.0 VDC Power
<b>9</b>	GPIO 9 SCL1 (I2C)	5		6	Ground
<b>7</b>	GPIO 7 GPCLK0	7		8	GPIO 15 TxD (UART) <b>15</b>
	Ground	9		10	GPIO 16 RxD (UART) <b>16</b>
<b>0</b>	GPIO 0	11		12	GPIO 1 PCM_CLK/PWM0 <b>1</b>
<b>2</b>	GPIO 2	13		14	Ground
<b>3</b>	GPIO 3	15		16	GPIO 4 <b>4</b>
	3.3 VDC Power	17		18	GPIO 5 <b>5</b>
<b>12</b>	GPIO 12 MOSI (SPI)	19		20	Ground
<b>13</b>	GPIO 13 MISO (SPI)	21		22	GPIO 6 <b>6</b>
<b>14</b>	GPIO 14 SCLK (SPI)	23		24	GPIO 10 CE0 (SPI) <b>10</b>
	Ground	25		26	GPIO 11 CE1 (SPI) <b>11</b>
<b>30</b>	SDA0 (I2C ID EEPROM)	27		28	SCL0 (I2C ID EEPROM) <b>31</b>
<b>21</b>	GPIO 21 GPCLK1	29		30	Ground
<b>22</b>	GPIO 22 GPCLK2	31		32	GPIO 26 PWM0 <b>26</b>
<b>23</b>	GPIO 23 PWM1	33		34	Ground
<b>24</b>	GPIO 24 PCM_FS/PWM1	35		36	GPIO 27 <b>27</b>
<b>25</b>	GPIO 25	37		38	GPIO 28 PCM_DIN <b>28</b>
	Ground	39		40	GPIO 29 PCM_DOUT <b>29</b>

**Attention!** The GPIO pin numbering used in this diagram is intended for use with WiringPi / Pi4J. This pin numbering is not the raw Broadcom GPIO pin numbers.

<http://www.pi4j.com>

Рисунок 3.7 – Raspberry Pi 2 pinout.

RPi2 може функціонувати майже на будь-якій OS, але найзручніше використовувати її зі спеціалізованим дистрибутивом Linux – Raspbian.

Raspbian - заснована на дистрибутиві Debian. З 2015 року Raspbian офіційно представлена Raspberry Pi Foundation в якості основної операційної системи для одноплатних комп'ютерів

Raspberry Pi. Raspbian оптимізована для низько продуктивних процесорів ARM , які використовуються в лінійці комп'ютерів Raspberry Pi.

## 2.5 Камера

Для наділення гексапода зором використовується камера Raspi CAM v2 (рис. 3.8). Це офіційна камера від Raspberry Pi Foundation, яка має 8 Мр сенсор Sony IMX219 та покращений фіксований фокус. Камера дозволяє роботи статичні фото с роздільною здатністю 3280 x 2464 пікселів, знімати відео у трьох режимах – 1080р@ 30fps, 720р @ 60fps та 640 x 480р @ 90fps. Поставляється в комплекті зі шлейфом для зручного підключення до RPі.



Рисунок 3.8 – Зовнішній вигляд камери Raspi CAM v2.

Технічні характеристики камери наведені у таблиці 3.4.

Таблиця 3.4 – Характеристики Raspi CAM v2.

Габарити	25.0×23.0×9.0mm
Дозвіл	8 Мр
Максимальна роздільна здатність	3280 x 2464
Інтерфейс підключення	CSI
Сенсор	Sony IMX219
Розмір матриці CCD	¼ дюйма 1.4µm x 1.4µm пікселів з технологією OmniBSI для високої продуктивності (висока чутливість, низький рівень перехресних перешкод, низький рівень шуму)

## 2.6 IR датчик відстані

У якості датчика наближення використовується датчик Sharp GP2Y0A02YK0F (рис.3.9).

GP2Y0A02YK0F - це блок вимірювання дальності, який складається з інтегрованої комбінації PSD, IRED та схеми обробки сигналів. Різноманітність відбивної здатності об'єкта, температура навколишнього середовища та тривалість роботи не оказують сильного впливання на виявлення відстані, тому що використовується метод тріангуляції. Цей пристрій виводить напругу, що відповідає за виявлену відстань до об'єкта, тому цей датчик також можна використовувати як датчик наближення.



Рисунок 3.9 – Зовнішній вигляд Sharp GP2Y0A02YK0F.

Нижче представлена технічна специфікація (табл. 3.5) та блок схема пристрою (рис. 3.10).

Таблиця 3.5 – Специфікація Sharp GP2Y0A02YK0F.

Габарити	29.5×13.0×21.6mm
Тип вихідного сигналу	Аналоговий (до 3V)
Діапазон вимірювання відстані	20-150cm
Робоча напруга (рекомендована)	4.5-5.5V
Робочий струм	33mA

## ■ Block diagram

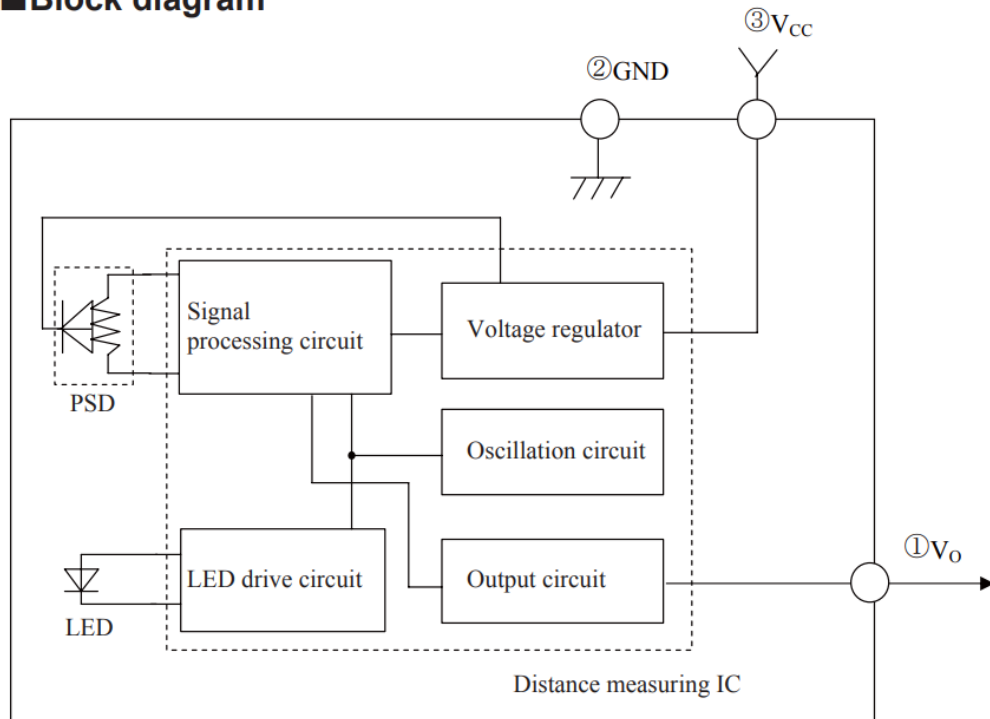


Рисунок 3.9 – Блок схема Sharp GP2Y0A02YK0F.

### 2.6.1 I2C АЦП на ADS1115

Так як Sharp GP2Y0A02YK0F має аналоговий вихідний сигнал, а RPi2 не має аналогових входів, то необхідно використовувати аналого-цифровий перетворювач. Для цього був обраний 16-бітний АЦП на мікросхемі ADS1115/ADS1015 (рис. 3.10), який має 4 аналогові входи та працює через шину I2C.

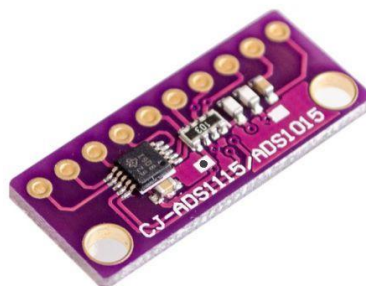


Рисунок 3.10 – Зовнішній вигляд перетворювача.

Нижче представлені технічні характеристики (табл. 3.6) перетворювача.

Таблиця 3.6 – Характеристики АЦП ADS1115/ADS1015.

Розміри	25.5×9.0mm
Мікросхема модулю	ADS1115
Інтерфейс	I2C
Кількість каналів	4
Робоча напруга	2-5V
Робочий струм	150µA
Діапазон вхідної напруги	від ±256mV до ±6.144V
Розрядність АЦП	16 (зі знаком)
Діапазон адресів	від 0x49 до 0x4B
Діапазон робочих температур	від -40 до +125

АЦП на ADS1115 передбачає установку адреси для I2C шини за допомогою замикання виведення ADDR на цифрові сигнальні виводи або виводи шини живлення [9]. Установка різних адрес I2C призначена для того, щоб при підключенні двох і більше різних пристроїв не відбувалося конфлікту між ними. В кращому випадку модулі у яких адреси співпадають просто не будуть працювати, а в гіршому – робота всіх модулів на шині може стати непередбачуваною.

## 2.7 Електроживлення

### 2.7.1 Акумулятор

Використання 18 сервоприводів, управляючих плат та модулів потребує значною кількості електричного струму. При максимальній загрузці максимальний струм може становити 23600mA, тому вкрай важливо використовувати якісний акумулятор з високою токовіддачею. У якості живлення гексапода використовується Li-Po акумулятор на 4500mAh з максимально можливою токовіддачею 30C (рис.3.11).



Рисунок 3.11 – Зовнішній вигляд акумулятора.

Нижче представлені повні технічні характеристики (табл. 3.7) акумулятора.

Таблиця 3.7 – Характеристики акумулятора.

Габарити	138.0×46.0×25.0mm
Тип	Li-Po
Ємність	4500mAh
Напруга	7.4V
Максимальна токовіддача	30C
Вага	199g



### 2.7.2 DC-DC понижуючий конвертер 3А

Так як прийнято використовувати акумулятор на 7.4V, а RPi2 вимагає стабільні 5V, то слід знизити напругу яка буде подаватися на RPi2. Для цього був обраний понижуючий конвертер постійного струму на 3A (рис.3.11), якого більш ніж достатньо для живлення RPi2. Конвертер видає на виході стабільні 5V та має USB інтерфейс, що робить процес підключення до RPi2 дуже простою задачею.



Рисунок 3.11 – Понижуючий конвертер.

Нижче представлені технічні характеристики (табл. 3.8) конвертера.

Таблиця 3.8 – Характеристики понижуючого конвертера.

Габарити	26.0×15.0×8.0mm
Тип перетворення	синхронне випрямлення
Вхідна напруга	6-24V
Тип вхідної напруги	постійна
Вихідна напруга	5.1-5.2V
Максимальний вихідний струм	3A
ККД перетворювача	97.5%
Робоча температура	від 40 C до +85 C
Струм холостого ходу	0.85mA
Стабілізація напруги	± 0.5%

### 2.7.3 Вибір силових кабелів

Так як сервоприводи та одноплатний комп'ютер потребують значних токів (21.6А та 2А відповідно), то слід підібрати відповідні кабелі. Для цього була використана специфікація гнучких кабелів в високотемпературній силіконовій ізоляції, яка представлена на рис.3.12.

Flexible Silicone Wire Cable Specification (0.08mm Series)

Size	Conductor Number	Conductor Diameter	Outer Diameter	Insulation Thickness	Conductor Resistance	Electric Current	Length Per Roll
30AWG	11 pcs	0.30 mm	0.8 mm	0.05 mm	331Ω /km	0.8 A	500m
28AWG	16 pcs	0.32 mm	1.2 mm	0.08 mm	227Ω /km	1.2 A	500m
26AWG	30 pcs	0.44 mm	1.5 mm	0.15 mm	123Ω /km	3.5 A	400m
24AWG	40 pcs	0.58 mm	1.6 mm	0.20 mm	97Ω /km	5.0 A	400m
22AWG	60 pcs	0.72 mm	1.7 mm	0.30 mm	88Ω /km	8.7 A	400m
20AWG	100 pcs	0.92 mm	1.8 mm	0.50 mm	62Ω /km	13 A	400m
18AWG	150 pcs	1.20 mm	2.3 mm	0.55 mm	39Ω /km	22 A	200m
16AWG	252 pcs	1.53 mm	3.0 mm	0.80 mm	24Ω /km	35 A	200m
14AWG	400 pcs	1.75 mm	3.5 mm	0.90 mm	15Ω /km	55 A	200m
12 AWG	680 pcs	2.50 mm	4.5 mm	1.00 mm	9.0Ω /km	88 A	100m
10 AWG	1050 pcs	3.00 mm	5.5 mm	1.20 mm	6.3Ω /km	140 A	100m
8 AWG	1650 pcs	4.40 mm	6.3 mm	1.20 mm	4.0Ω /km	190 A	100m
7 AWG	2400 pcs	4.60 mm	7.2 mm	1.20 mm	1.2Ω /km	205 A	100m
6 AWG	3200 pcs	5.20 mm	8.5 mm	1.65 mm	1.2Ω /km	230 A	100m

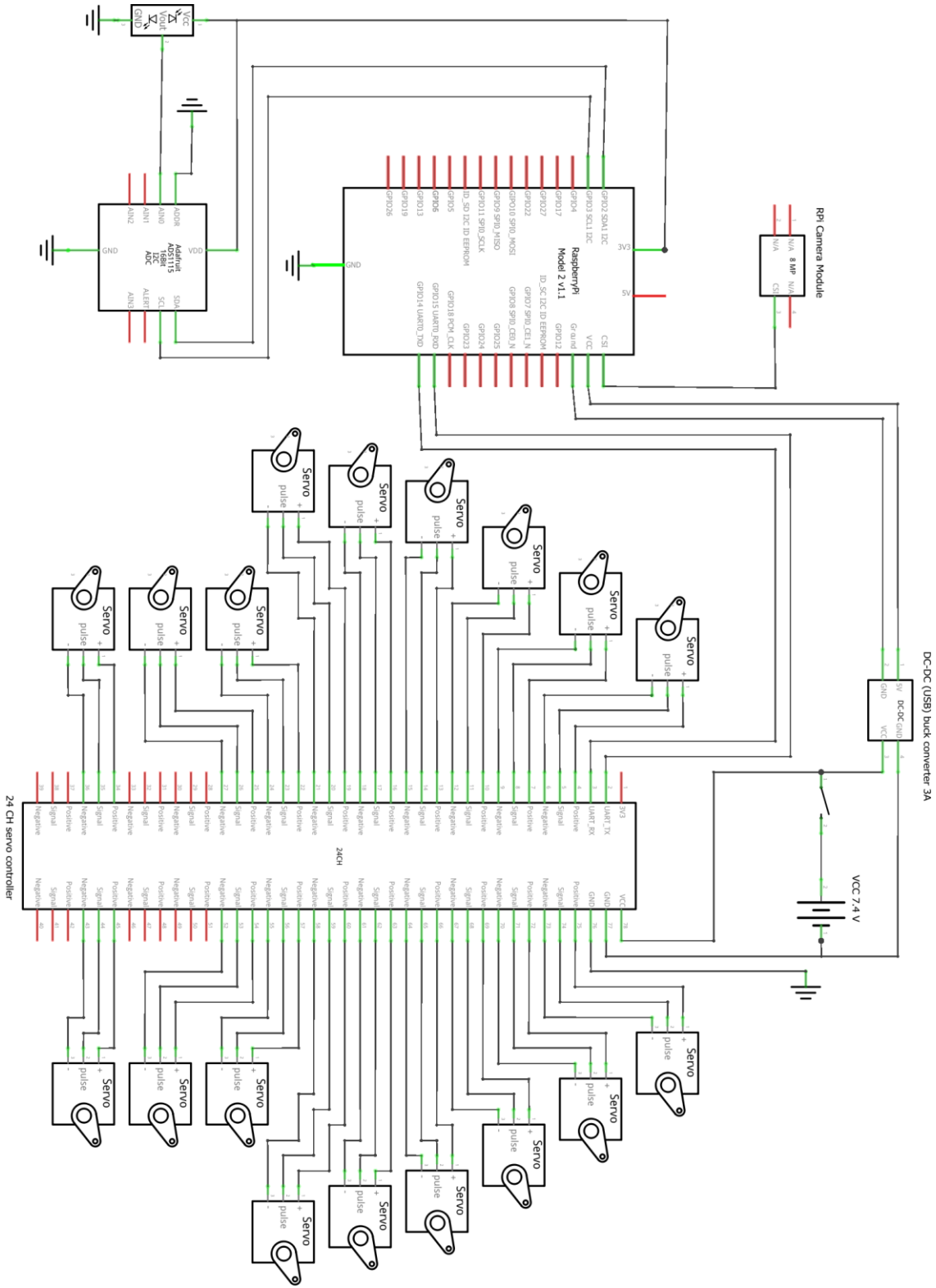
Note: There is a certain tolerance of the data, reference only.

Рисунок 3.12 – Специфікація силових кабелів.

Відповідно до цієї специфікації були обрані кабелі марки 26AWG (для RPi2 та понижуючого конвертера) та 18AWG (для сервоконтролера, сервоприводів).

### 2.8 Функціональна схема

На рисунку 3.13 представлена функціональна схема з'єднання компонентів гексапода, яка виконана у програмному інструменті «Fritzing».



fritzing

Рисунок 3.13 – Функціональна схема з'єднання.

## 2.9 Висновки до розділу 2

У даному розділі були розглянуті технічні характеристики та специфікація використаних модулів та компонентів, а також проілюстрована функціональна схема з'єднання. Результатом роботи по апаратній частині став мобільний робот – гексапод, здатний до самостійного пересування. Усі частини були успішно протестовані на працездатність і відмовостійкість.

Загальні економічні витрати на створення гексапода представлені у таблиці нижче.

Електронна комплектуюча / Модуль	Вартість
Алюмінієвий набір	65.5\$
Сервоприводи MG995 (18 шт.)	169.2\$
Сервоконтроллер LSC-24	32.6\$
Одноплатний комп'ютер RPi2 Model B	40.0\$
Камера Raspberry v2 (8MP)	35.0\$
IR Sharp GP2Y0A02YK0F	13.5\$
ADC ADS1115/ASD1015	3.2\$
Акумулятор Li-Po 2S 4500mA 30C	35.0\$
Понижуючий (USB) конвертер 3A	1.5\$
Накладні витрати, витратні матеріали, тощо	40.0\$
<b>Загальна вартість: 435.5 \$</b>	

Наступним кроком стане налаштування OS гексапода та програмування програмного комплексу.

## 3 ПРОГРАМНА ЧАСТИНА

У цьому розділі розглядається програмна частина дослідного зразка. Відповідно до технічного завдання були виконані ряд заходів по оптимізації працездатності операційної системи гексапода. Була відтворена клієнт-сервера архітектура, яка носить характер налагодження та виступає у якості допоміжного засобу при онлайн навчанні та корекції рухів гексапода. Окрім реалізації основного завдання, також розглядається робота з протоколом зв'язку сервоконтролера та алгоритм ходи гексапода.

### 3.1 Підготовка OS. Патч ядра

Для початку необхідно оновити систему до останньої версії та інсталиювати необхідні компоненти для подальшої роботи.

```
sudo apt-get update && apt-get upgrade

sudo apt-get install mc
sudo apt-get install git
sudo apt-get install bc
sudo apt-get install libncurses-dev

sudo apt-get install dh-autoreconf

sudo apt-get install hostapd
sudo apt-get install dnsmasq
```

Далі скачуємо компоненти Xenomai та відповідний образ Raspbian. Потім виконуємо патч та перезбірку ядра.

```
cd /opt

wget https://www.xenomai.org/downloads/ipipe/v4.x/arm/older/ipipe-core-4.9.38-arm-3.patch

wget http://xenomai.org/downloads/xenomai/stable/xenomai-3.0.5.tar.bz2

tar xvjf xenomai-3.0.5.tar.bz2

git clone https://github.com/raspberrypi/linux.git

cd linux

git checkout f0cd77ded5127168b1b83ca2f366ee17e9c0586f

/opt/xenomai-3.0.5/scripts/prepare-kernel.sh --arch=arm --ipipe=/opt/ipipe-core-4.9.38-arm-3.patch

mkdir /opt/kernel
```

```

export KERNEL=kernel7
make multi_v7_defconfig O=/opt/kernel
make menuconfig
make -j4 zImage
modules O=/opt/kernel

make dtbs

sudo make modules_install

sudo cp /opt/kernel/arch/arm/boot/zImage /boot/kernel7.img
sudo cp /opt/kernel/arch/arm/boot/dts/bcm2836-rpi-2-b.dtb /boot
mcedit /boot/config.txt

    kernel=kernel7.img
    device_tree=bcm2836-rpi-2-b.dtb

sudo reboot

```

Після патча ядра встановлюємо API Xenomai:

```

cd /opt/xenomai-3.0.5
./scripts/bootstrap
./configure --enable-smp [-enable-pshared]
make
sudo make install
mcedit /root/.bashrc

append: export PATH="$PATH:/usr/xenomai/bin"

```

Підготовку OS можна вважати завершеною.

## 3.2 Дистанційне керування (сервер)

### 3.2.1 Створення точки доступу

Для того, щоб можна була зв'язуватися з гексаподом, використовуючи Wi-Fi, перш за все необхідно створити точку доступу. Для цього треба відреагувати сервісний файл служби dhcp та hostapd з необхідними параметрами та виконати перезапуск системи.

```

systemctl stop dnsmasq
systemctl stop hostapd

mcedit /etc/dhcpd.conf

interface wlan0
    static ip_address=192.168.0.1/24
    nohook wpa_supplicant

systemctl daemon-reload

mcedit /etc/dnsmasq.conf

dhcp-range=192.168.0.2,192.168.0.20,255.255.255.0,24h

mcedit /etc/hostapd/hostapd.conf

interface=wlan0
driver=nl80211
ssid=hexapod
hw_mode=g
channel=1
wmm_enabled=0
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=epsilon13
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP

mcedit /etc/default/hostapd
DAEMON_CONF="/etc/hostapd/hostapd.conf"

service hostapd start
service dnsmasq start

reboot

```

Після запуску системи створюється точка доступу Wi-Fi, яка має ім'я (SSID) hexapod та пароль epsilon13.

### 3.2.2 Програмний код сервера

Після того як створюється точка доступу, основний програмний модуль має розпочати прослуховування з'єднання для подальшої взаємодії з клієнтом. Для цього використовується клієнт-сервера архітектура з використанням протоколу TCP [10].

Відповідно до завдання була написана бібліотека для роботи з сокетом, з повним кодом якої можна ознайомитись у додатку А.

Використовуючи бібліотеку роботи з сокетом, створюється сервер, який прослуховує порт 8080, а потім циклічно очікує нового клієнта. Після того як клієнт підключився, запускається

потік який з частотою 2Hz відправляє дані про стан гексапода клієнту. Паралельно цьому потоку відбувається очікування та обробка команди від клієнта.

```

...
#define CMD_SHUTDOWN    0x11
#define CMD_REBOOT      0x12
...

int main()
{
    net::serversocket* server = new net::serversocket(8080);

    int result = server->listen();

    if (result != 0)
    {
        std::cout << strerror(result) << std::endl;
        return 0;
    }

    while (true)
    {
        net::socket *client = server->accept();

        if (!client->valid())
        {
            delete client;
            continue;
        }

        net::socketaddress *address = client->get_socketaddress();
        std::cout << address->get_address() << ":" << address->get_port() <<
std::endl;

        std::thread *thread = new std::thread(worker, client, hexapod);

        while (true)
        {
            std::string msg;

            if (client->read(msg) <= 0)
            {
                break;
            }

            char cmd = msg[0];

            if (cmd == CMD_SHUTDOWN)
            {
                sync();
                reboot(LINUX_REBOOT_CMD_POWER_OFF);
            }

            if (cmd == CMD_REBOOT)
            {

```



```

        sync();
        reboot(LINUX_REBOOT_CMD_RESTART);
    }

    ...

}

thread->detach();

delete thread;
delete client;

std::cout << "disconnected" << std::endl;
}

delete server;

return 0;
}

```

Так як TCP має послідовність байт «Big Endian», то була написана функція для конвертації послідовності байт. Ця функція використовується на стороні сервера при кожній відправці даних клієнту.

```

void *endian_swap(void *arg)
{
    unsigned int n = *((int *) arg);

    n = ((n >> 8) & 0x00ff00ff) | ((n << 8) & 0xff00ff00);
    n = ((n >> 16) & 0x0000ffff) | ((n << 16) & 0xffff0000);

    *((int *) arg) = n;

    return arg;
}

```

### 3.3 Дистанційне керування (клієнт)

У якості клієнту передбачається використання пристроїв на OS Android, тому клієнт був запрограмований на мові Java, використовуючи IDE «Android Studio».

#### 3.3.1 Графічний інтерфейс

Інтерфейс додатку було написано вручну, використовуючи мову розмітки XML. Інтерфейс має службові кнопки управління (такі як підключення до гексапода, вимкнення гексапода, перезавантаження, тощо) та кнопки корекції руху (при онлайн навчанні). Також додаток має інформацій поля з поточною напругою акумулятора та температурою центрального процесора

гексапода. Поля «temp» та «angle x,y,z» не використовуються (додані для майбутніх досліджень).

Нижче приведено код розмітки графічного інтерфейсу:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#000000"
    android:orientation="vertical"
    tools:context=".main">

    <LinearLayout
        style="@style/Widget.AppCompat.Button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <Button
            style="?metaButtonBarButtonStyle"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="0.25"
            android:onClick="handler_connect"
            android:text="@string/connect"
            android:textColor="#000000" />

        <Button
            style="?metaButtonBarButtonStyle"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="0.25"
            android:onClick="handler_disconnect"
            android:text="@string/disconnect"
            android:textColor="#000000" />

        <Button
            style="?metaButtonBarButtonStyle"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="0.25"
            android:onClick="handler_shutdown"
            android:text="@string/shutdown"
            android:textColor="#000000" />

        <Button
            style="?metaButtonBarButtonStyle"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="0.25"
            android:onClick="handler_reboot"
            android:text="@string/reboot"
            android:textColor="#000000" />

        <Button
            style="?metaButtonBarButtonStyle"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="0.25"
            android:onClick="handler_stop"
            android:text="@string/stop"
            android:textColor="#000000" />
    </LinearLayout>
</LinearLayout>
```

```

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:paddingLeft="10dp"
    android:paddingRight="10dp"
    android:paddingBottom="10dp">
    <TableLayout
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="25"
        android:orientation="vertical"
        android:stretchColumns="*">
        <TableRow
            android:paddingTop="10dp"
            android:paddingBottom="10dp">
            <TextView
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:text="@string/voltage"
                android:textColor="#33b5e5"
                android:textSize="14sp"
                android:textStyle="bold" />
            <TextView
                android:id="@+id/voltage"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:text="@string/unknown"
                android:textColor="#33b5e5"
                android:textSize="14sp"
                android:textStyle="bold" />
        </TableRow>
        <TableRow
            android:paddingTop="10dp"
            android:paddingBottom="10dp">
            <TextView
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:text="@string/cpu_temp"
                android:textColor="#33b5e5"
                android:textSize="14sp"
                android:textStyle="bold" />
            <TextView
                android:id="@+id/cpu_temp"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:text="@string/unknown"
                android:textColor="#33b5e5"
                android:textSize="14sp"
                android:textStyle="bold" />
        </TableRow>
        <TableRow
            android:paddingTop="10dp"
            android:paddingBottom="10dp">
            <TextView
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:text="@string/temp"
                android:textColor="#33b5e5"

```

```

        android:textSize="14sp"
        android:textStyle="bold" />
    <TextView
        android:id="@+id/temp"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:text="@string/unknown"
        android:textColor="#33b5e5"
        android:textSize="14sp"
        android:textStyle="bold" />
    </TableRow>
</TableLayout>
<TextView
    android:id="@+id/log"
    android:layout_width="0dp"
    android:layout_height="150dp"
    android:layout_weight="50"
    android:background="#000000"
    android:textColor="@android:color/holo_green_dark" />
<TableLayout
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="25"
    android:orientation="vertical"
    android:stretchColumns="*">
    <TableRow
        android:paddingTop="10dp"
        android:paddingBottom="10dp">
        <TextView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:gravity="end"
            android:text="@string/angle_x"
            android:textColor="#33b5e5"
            android:textSize="14sp"
            android:textStyle="bold" />
        <TextView
            android:id="@+id/angle_x"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:gravity="end"
            android:text="@string/unknown"
            android:textColor="#33b5e5"
            android:textSize="14sp"
            android:textStyle="bold" />
    </TableRow>
    <TableRow
        android:paddingTop="10dp"
        android:paddingBottom="10dp">
        <TextView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:gravity="end"
            android:text="@string/angle_y"
            android:textColor="#33b5e5"
            android:textSize="14sp"
            android:textStyle="bold" />
        <TextView
            android:id="@+id/angle_y"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:gravity="end"

```

```

        android:text="@string/unknown"
        android:textColor="#33b5e5"
        android:textSize="14sp"
        android:textStyle="bold" />
</TableRow>
<TableRow>
    android:paddingTop="10dp"
    android:paddingBottom="10dp">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="end"
        android:text="@string/angle_z"
        android:textColor="#33b5e5"
        android:textSize="14sp"
        android:textStyle="bold" />
    <TextView
        android:id="@+id/angle_z"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="end"
        android:text="@string/unknown"
        android:textColor="#33b5e5"
        android:textSize="14sp"
        android:textStyle="bold" />
    </TableRow>
</TableLayout>
</LinearLayout>

<View
    android:layout_width="match_parent"
    android:layout_height="1dp"
    android:background="#33b5e5" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:baselineAligned="false"
    android:orientation="horizontal">
    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center_vertical"
        android:layout_weight="50"
        android:orientation="vertical">
        <Button
            android:layout_width="match_parent"
            android:layout_height="70dp"
            android:onClick="handler_up"
            android:text="@string/up"
            android:textColor="#000000" />
    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <Button
            android:layout_width="0dp"
            android:layout_height="70dp"
            android:layout_weight="50"
            android:onClick="handler_turn_left"
            android:text="@string/turn_left"
            android:textColor="#000000" />

```

```

        <Button
            android:layout_width="0dp"
            android:layout_height="70dp"
            android:layout_weight="50"
            android:onClick="handler_turn_right"
            android:text="@string/turn_right"
            android:textColor="#000000" />
    </LinearLayout>
    <Button
        android:layout_width="match_parent"
        android:layout_height="70dp"
        android:onClick="handler_down"
        android:text="@string/down"
        android:textColor="#000000" />
</LinearLayout>
<LinearLayout
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center_vertical"
    android:layout_weight="50"
    android:orientation="vertical">
    <Button
        android:layout_width="match_parent"
        android:layout_height="70dp"
        android:onClick="handler_forward"
        android:text="@string/forward"
        android:textColor="#000000" />
    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <Button
            android:layout_width="0dp"
            android:layout_height="70dp"
            android:layout_weight="50"
            android:onClick="handler_left"
            android:text="@string/left"
            android:textColor="#000000" />
        <Button
            android:layout_width="0dp"
            android:layout_height="70dp"
            android:layout_weight="50"
            android:onClick="handler_right"
            android:text="@string/right"
            android:textColor="#000000" />
        </LinearLayout>
    <Button
        android:layout_width="match_parent"
        android:layout_height="70dp"
        android:onClick="handler_backward"
        android:text="@string/backward"
        android:textColor="#000000" />
    </LinearLayout>
</LinearLayout>
</LinearLayout>

```

На рисунку 3.1 представлено знімок з екрану додатка (для демонстрації графічного інтерфейсу).

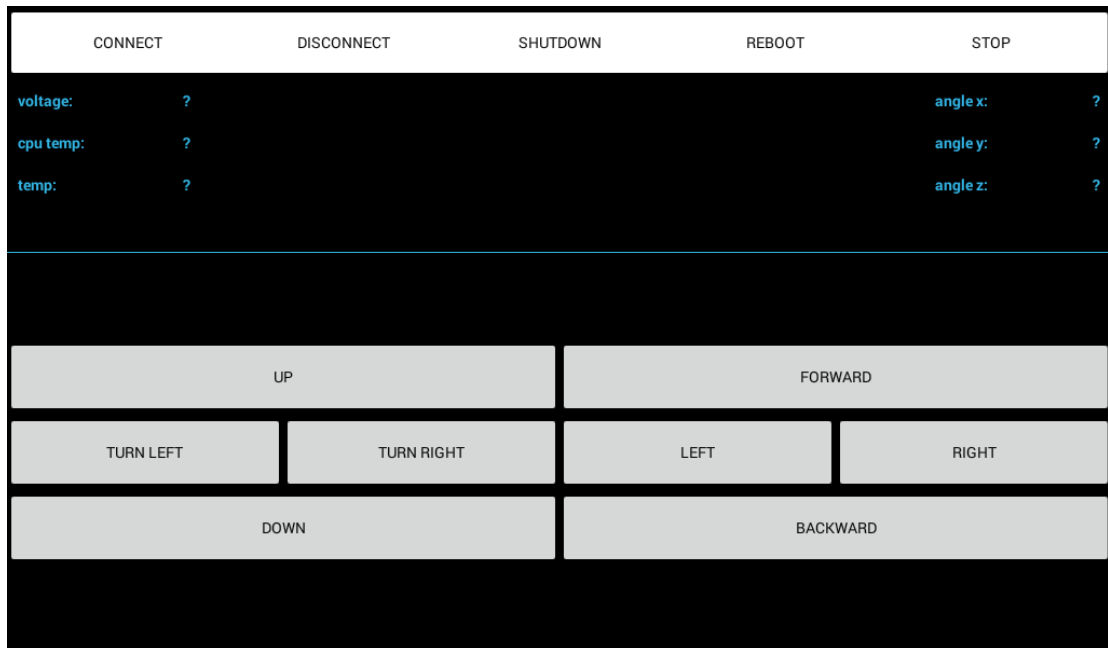


Рисунок 3.1 – Графічний інтерфейс додатка.

### 3.3.2 Програмний код клієнта

Перш за все у маніфест додатка необхідно додати дозвіл на використання TCP з'єднання та Wi-Fi [11].

```
| <uses-permission android:name="android.permission.INTERNET" />
```

Наступним кроком стало написання класу асинхронного з'єднання, який відкриває з'єднання з сервером та, через java interface, забезпечує постійне читання даних (якщо вони є). Також клас має метод для послідовного запису (передавання) даних та метод закриття з'єднання.

```
package exist009.hexapod;

import java.io.BufferedWriter;
import java.io.DataInputStream;
import java.io.IOException;
import java.io.OutputStreamWriter;
import java.net.InetSocketAddress;
import java.net.Socket;
import java.net.UnknownHostException;

import android.util.Log;

public class AsyncConnection extends android.os.AsyncTask<Void, String, Exception>
{
    private String url;
```

```

private int port;
private int timeout;
private ConnectionHandler connectionHandler;

private DataInputStream in;
private BufferedWriter out;
private Socket socket;
private boolean interrupted = false;

private String TAG = getClass().getName();

AsyncConnection(String url, int port, int timeout, ConnectionHandler
connectionHandler)
{
    this.url = url;
    this.port = port;
    this.timeout = timeout;
    this.connectionHandler = connectionHandler;
}

@Override
protected void onProgressUpdate(String... values)
{
    super.onProgressUpdate(values);
}

@Override
protected void onPostExecute(Exception result)
{
    super.onPostExecute(result);
    Log.d(TAG, "Finished communication with the socket. Result = " + result);
}

@Override
protected Exception doInBackground(Void... params)
{
    Exception error = null;

    try
    {
        Log.d(TAG, "Opening socket connection.");

        socket = new Socket();
        socket.connect(new InetSocketAddress(url, port), timeout);

        in = new DataInputStream(socket.getInputStream());
        out = new BufferedWriter(new
OutputStreamWriter(socket.getOutputStream()));

        connectionHandler.didConnect();

        while (!interrupted)
        {
            float voltage = in.readFloat();
            float cpu_temp = in.readFloat();
            float temp = in.readFloat();

            float angle_x = in.readFloat();
            float angle_y = in.readFloat();
            float angle_z = in.readFloat();

```



```

        connectionHandler.didReceiveData(voltage, cpu_temp, temp, angle_x,
angle_y, angle_z);
    }
}
catch (UnknownHostException e)
{
    Log.e(TAG, "doInBackground(): " + e.toString());
    error = interrupted ? null : e;
}
catch (IOException e)
{
    Log.d(TAG, "doInBackground(): " + e.toString());
    error = interrupted ? null : e;
}
catch (Exception e)
{
    Log.e(TAG, "doInBackground(): " + e.toString());
    error = interrupted ? null : e;
}
finally
{
    try
    {
        socket.close();
        out.close();
        in.close();
    }
    catch (Exception e)
    {
        Log.e(TAG, "close error: " + e.toString());
    }
}

connectionHandler.didDisconnect(error);

return error;
}

void write(final byte command)
{
    try
    {
        out.write(command);
        out.flush();
    }
    catch (NullPointerException | IOException e)
    {
        Log.e(TAG, "write(): " + e.toString());
    }
}

void disconnect()
{
    try
    {
        Log.d(TAG, "Closing the socket connection.");

        interrupted = true;

        if(socket != null)
        {
            socket.close();

```

```

        }

        if(out != null & in != null)
        {
            out.close();
            in.close();
        }
    }
    catch (IOException ex)
    {
        Log.e(TAG, "disconnect(): " + ex.toString());
    }
}

```

Java interface який використовується для отримання даних від сервера (та статусу підключення) приведено нижче.

```

package exist009.hexapod;

public interface ConnectionHandler {

    void didReceiveData(float voltage, float cpu_temp, float temp, float angle_x,
float angle_y, float angle_z);

    void didDisconnect(Exception error);

    void didConnect();
}

```

Заключним етапом розробки мобільного додатка стало написання точки входу програми, яка відповідає за взаємодію кнопок (тобто має обробники кнопок, які були вказані при розробці інтерфейсу) та роботу з логом (для налагодження).

```

package exist009.hexapod;

import android.support.v7.app.ActionBar;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.Layout;
import android.text.method.ScrollingMovementMethod;
import android.view.View;
import android.widget.TextView;

import java.text.SimpleDateFormat;
import java.util.Locale;

public class main extends AppCompatActivity implements ConnectionHandler
{
    private static final byte CMD_SHUTDOWN    = 0x11;
    private static final byte CMD_REBOOT     = 0x12;
    private static final byte CMD_STOP       = 0x13;
    private static final byte CMD_UP         = 0x21;
    private static final byte CMD_DOWN      = 0x22;
    private static final byte CMD_TURN_LEFT = 0x23;
    private static final byte CMD_TURN_RIGHT = 0x24;
}

```

```

private static final byte CMD_FORWARD    = 0x25;
private static final byte CMD_BACKWARD  = 0x26;
private static final byte CMD_LEFT     = 0x27;
private static final byte CMD_RIGHT    = 0x28;

private TextView log;
private TextView _voltage;
private TextView _cpu_temp;
private TextView _temp;
private TextView _angle_x;
private TextView _angle_y;
private TextView _angle_z;

private AsyncConnection connection;
private boolean connection_state = false;

@Override
public void didReceiveData(final float voltage, final float cpu_temp, final float
temp, final float angle_x, final float angle_y, final float angle_z)
{
    runOnUiThread(new Runnable() {
        @Override
        public void run() {
            _voltage.setText(String.format(Locale.US, "%.2f", voltage));
            _cpu_temp.setText(String.format(Locale.US, "%.2f", cpu_temp));
            _temp.setText(String.format(Locale.US, "%.2f", temp));
            _angle_x.setText(String.format(Locale.US, "%.2f", angle_x));
            _angle_y.setText(String.format(Locale.US, "%.2f", angle_y));
            _angle_z.setText(String.format(Locale.US, "%.2f", angle_z));
        }
    });
}

@Override
public void didConnect()
{
    connection_state = true;
    logWrite("Подключено");
}

@Override
public void didDisconnect(Exception error)
{
    connection_state = false;
    logWrite(error != null ? "Ошибка: " + error.toString() : "Отключено");
}

private void logWrite(final String msg)
{
    if(msg == null || msg.isEmpty())
    {
        return;
    }

    runOnUiThread(new Runnable()
    {
        @Override
        public void run()
        {
            long date = System.currentTimeMillis();
            SimpleDateFormat sdf = new SimpleDateFormat("[kk:mm:ss.SSS] ",
Locale.US);

```

```
        String dateString = sdf.format(date);
        log.append(dateString + msg + "\n");

        final Layout layout = log.getLayout();

        if(layout != null)
        {
            int scrollDelta = layout.getLineBottom(log.getLineCount() - 1) -
log.getScrollY() - log.getHeight();

            if(scrollDelta > 0)
            {
                log.scrollBy(0, scrollDelta);
            }
        }
    });
}

public void handler_connect(View v)
{
    if (connection_state) return;
    connection = new AsyncConnection("192.168.0.1", 8080, 1000, this);
    connection.execute();
}

public void handler_disconnect(View v)
{
    if (!connection_state) return;
    connection.disconnect();
}

public void handler_shutdown(View v)
{
    if (!connection_state) return;
    connection.write(CMD_SHUTDOWN);
}

public void handler_reboot(View v)
{
    if (!connection_state) return;
    connection.write(CMD_REBOOT);
}

public void handler_stop(View v)
{
    if (!connection_state) return;
    connection.write(CMD_STOP);
}

public void handler_up(View v)
{
    if (!connection_state) return;
    connection.write(CMD_UP);
}

public void handler_down(View v)
{
    if (!connection_state) return;
    connection.write(CMD_DOWN);
}
```

```
public void handler_turn_left(View v)
{
    if (!connection_state) return;
    connection.write(CMD_TURN_LEFT);
}

public void handler_turn_right(View v)
{
    if (!connection_state) return;
    connection.write(CMD_TURN_RIGHT);
}

public void handler_forward(View v)
{
    if (!connection_state) return;
    connection.write(CMD_FORWARD);
}

public void handler_backward(View v)
{
    if (!connection_state) return;
    connection.write(CMD_BACKWARD);
}

public void handler_left(View v)
{
    if (!connection_state) return;
    connection.write(CMD_LEFT);
}

public void handler_right(View v)
{
    if (!connection_state) return;
    connection.write(CMD_RIGHT);
}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    log = findViewById(R.id.log);
    log.setMovementMethod(new ScrollingMovementMethod());

    _voltage = findViewById(R.id.voltage);
    _cpu_temp = findViewById(R.id.cpu_temp);
    _temp = findViewById(R.id.temp);
    _angle_x = findViewById(R.id.angle_x);
    _angle_y = findViewById(R.id.angle_y);
    _angle_z = findViewById(R.id.angle_z);
}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    ActionBar actionBar = getSupportActionBar();

    if (actionBar != null)
    {
        actionBar.hide();
    }
}
```

```

        hideSystemUI();
    }

    @Override
    protected void onResume()
    {
        super.onResume();
        hideSystemUI();
    }

    private void hideSystemUI() {
        View decorView = getWindow().getDecorView();
        decorView.setSystemUiVisibility(
            View.SYSTEM_UI_FLAG_IMMERSIVE_STICKY
            | View.SYSTEM_UI_FLAG_LAYOUT_STABLE
            | View.SYSTEM_UI_FLAG_LAYOUT_HIDE_NAVIGATION
            | View.SYSTEM_UI_FLAG_LAYOUT_FULLSCREEN
            | View.SYSTEM_UI_FLAG_HIDE_NAVIGATION
            | View.SYSTEM_UI_FLAG_FULLSCREEN);
    }
}

```

### 3.4 Робота з протоколом зв'язку LSC-24 v1.2

Наступним кроком стало написання програми для взаємодії сервера (далі основний модуль) та сервоконтроллера. Сервоконтроллер LSC-24 має власний протокол для спілкування по UART (рис.3.2).

#### LSC Series Servo Controller Communication Protocol V1.2

Serial Communication, Baud Rate is 9600 baud

Header	Data Length	Command	Parameter
0x55 0x55	Length	Cmd	Prm 1...Prm N

Header: Received two consecutive 0x55, indicates that the data packets arrived.

Length: Equal to the parameter number N plus a command and plus the byte length occupied by the data length itself. That means the data length is equal to the parameter N plus 2.

(Length=N +command+a byte length = N +2)

Command: Various control instructions.

Parameter: In addition to the command, the need to add control information

Рисунок 3.2 – LSC-24 communication protocol.

Основному модулю потрібно отримувати напругу на акумуляторі (для виявлення розрядження) та задавати кут відхилення потрібного сервопривода, для цього в протоколі LSC-24 існують команди CMD\_GET\_BATTERY\_VOLTAGE (рис. 3.3) та CMD\_SERVO\_MOVE (рис. 3.4).

5. Command name: CMD\_GET\_BATTERY\_VOLTAGE Command value: 11  
Length(data length value): 2

Description: Get the servo controller's battery voltage in unit millivolts, the controller will immediately return the data after the command has been sent to it, the returned data is a data packet with two parameter values.

Parameter1: no parameter

Transmit:

Header	Length	Command	Parameter
0x55 0x55	0x02	0x0F	No parameter

Return: The parameter 1 of the data returned by servo controller represents the lower 8 bits of the voltage value, and parameter 2 represents the higher 8 bits of the voltage, such as the return voltage of 7500mV

Header	Length	Command	Parameter
0x55 0x55	0x04	0x0F	0x4C 0x1D

Рисунок 3.3 – Команда CMD\_GET\_BATTERY\_VOLTAGE.

1. Command name: CMD\_SERVO\_MOVE Command value: 3  
Length(Data Length):

Description: Control the rotation of any servo(you can control anyone as you like),  
Length = the number of control servo\*3+5

Parameter 1: The number of servo to be controlled

Parameter 2: Lower 8 bits of time value

Parameter 3: Higher 8 bits of time value

Parameter 4: Servo ID number

Parameter 5: Lower 8 bits of the angle position value

Parameter 6: Higher 8 bits of the angle position value

Parameter.....: The format is the same as the parameter 4, 5 and 6, control the angle

Рисунок 3.4 – Команда CMD\_SERVO\_MOVE.

Нижче представлено метод основного класу Hexapod, який дозволяє отримати дані напруги від LSC-24 відповідно до його протоколу:

```
float Hexapod::get_voltage()
{
    uint8_t tx[4];

    tx[0] = 0x55;
    tx[1] = 0x55;
    tx[2] = 0x02;
    tx[3] = 0x0F;

    if (write(this->UART, tx, sizeof(tx)) < 0)
    {
        return -1;
    }

    uint8_t rx[6];

    struct pollfd poll_struct = {this->UART, POLLIN, 0};

    if (poll(&poll_struct, 1, 100) && read(this->UART, rx, sizeof(rx)))
    {
        return *(int16_t *)&rx[4] / 1000.;
    }

    return -2;
}
```

Метод для позиціонування сервопривода має наступний вид:

```
bool Hexapod::set_servo_angle(Servo servo, uint16_t time)
{
    uint8_t tx[10];

    tx[0] = 0x55;
    tx[1] = 0x55;
    tx[2] = 0x08;
    tx[3] = 0x03;
    tx[4] = 0x01;

    memcpy(&tx[5], &time, sizeof(time));

    tx[7] = servo.id;

    memcpy(&tx[8], &servo.angle, sizeof(servو.angle));

    if (write(this->UART, tx, sizeof(tx)) < 0)
    {
        return true;
    }

    return false;
}
```



### 3.5 Tripod gait

Для того щоб гексапод міг пересуватися потрібно запрограмувати послідовність роботи сервоприводів. Для того, щоб розібратися як ходить гексапод, слід звернутися до рисунку 3.5, на якому чітко видно основні алгоритми ходи гексаподів. Для даного проекту було обрано алгоритм Tripod gait, тому що він являється найшвидшим, але при цьому навантаження на центральні ноги більше, ніж на передні і задні, що погано позначається на тривалість життя сервоприводів [12].

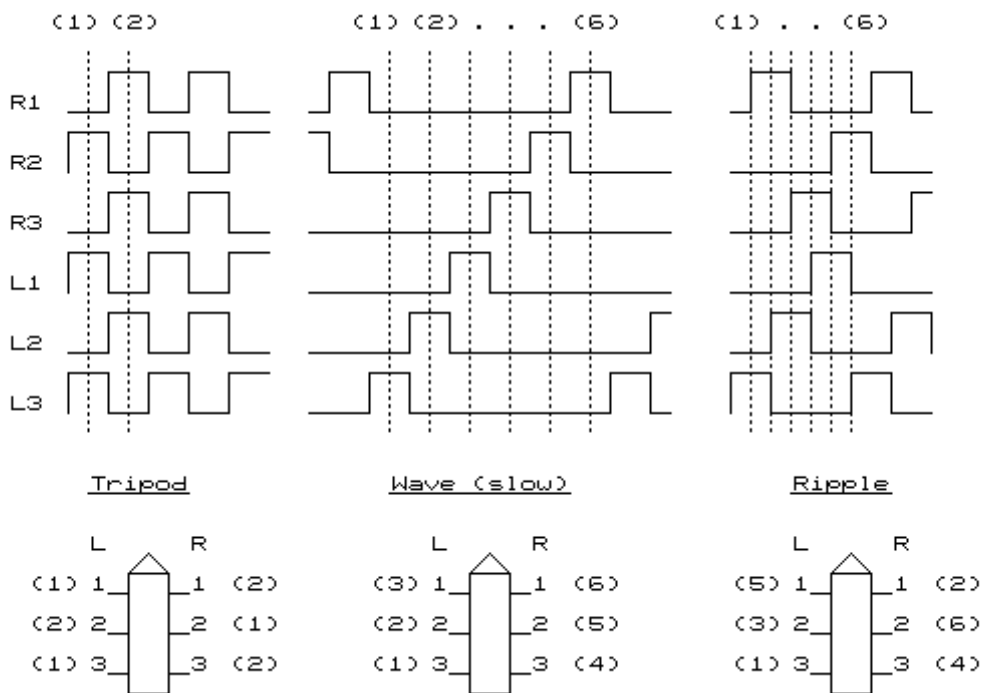


Рисунок 3.5 – Основні алгоритми пересування гексапода.

Приклад спрощеного алгоритму для пересування вперед наведено нижче:

```

if (cmd == CMD_FORWARD)
{
    hexapod->set_servo_angle({RF_FEMUR, 1300}, 0);
    hexapod->set_servo_angle({RR_FEMUR, 1300}, 0);
    hexapod->set_servo_angle({LM_FEMUR, 1700}, 0);

    //usleep(200000);

    hexapod->set_servo_angle({RF_COXA, 1300}, 0);
    hexapod->set_servo_angle({RR_COXA, 1300}, 0);
    hexapod->set_servo_angle({LM_COXA, 1700}, 0);

    //usleep(200000);

    hexapod->set_servo_angle({RF_FEMUR, 1500}, 0);

```

```

        hexapod->set_servo_angle({RR_FEMUR, 1500}, 0);
        hexapod->set_servo_angle({LM_FEMUR, 1500}, 0);

        //usleep(200000);

        hexapod->set_servo_angle({RF_COXA, 1500}, 0);
        hexapod->set_servo_angle({RR_COXA, 1500}, 0);
        hexapod->set_servo_angle({LM_COXA, 1500}, 0);

        hexapod->set_servo_angle({LF_FEMUR, 1700}, 0);
        hexapod->set_servo_angle({LR_FEMUR, 1700}, 0);
        hexapod->set_servo_angle({RM_FEMUR, 1300}, 0);

        //usleep(200000);

        hexapod->set_servo_angle({LF_COXA, 1700}, 0);
        hexapod->set_servo_angle({LR_COXA, 1700}, 0);
        hexapod->set_servo_angle({RM_COXA, 1300}, 0);

        //usleep(200000);

        hexapod->set_servo_angle({LF_FEMUR, 1500}, 0);
        hexapod->set_servo_angle({LR_FEMUR, 1500}, 0);
        hexapod->set_servo_angle({RM_FEMUR, 1500}, 0);

        //usleep(200000);

        hexapod->set_servo_angle({LF_COXA, 1500}, 0);
        hexapod->set_servo_angle({LR_COXA, 1500}, 0);
        hexapod->set_servo_angle({RM_COXA, 1500}, 0);
    }

```

### 3.6 Модуль роботи з нейромережею

Модуль роботи з нейромережею відповідає не тільки за створення, загрузку та збереження нейромережі, а також і за роботу з камерою та інфрачервоним сенсором. Модуль написан на мові програмування Python. За роботу з IR сенсором (та ADS1115) відповідає спеціалізована бібліотека wiringpi.

Робота з IR сенсором:

```

import wiringpi

ADS1115_PIN = 100
ADS1115_I2C_ADDRESS = 0x48

wiringpi.wiringPiSetup()
wiringpi.ads1115Setup(ADS1115_PIN, ADS1115_I2C_ADDRESS)

def get_proximity():
    distance = 60.374 * pow(wiringpi.analogRead(ADS1115_PIN) * 0.000125, -1.16)

    if distance > 50:
        return 51
    if distance < 25:
        return 25

```

```
return distance
```

Робота з камерою та неймережею (роздільна здатність камери, кількість шарів неймережі та їх параметри були підібрані емпірично, зважаючи на споживану оперативну пам'ять та чисельні можливості RPi2):

```
import picamera
import pyogmaneo

cam_width = 64
cam_height = 64

camera = picamera.PiCamera()

camera.resolution = (cam_width, cam_height)
camera.framerate = 24

cs = pyogmaneo.PyComputeSystem(4)

inputColumnSize = 64

for i in range(9): # 9 layers with exponential memory
    ld = pyogmaneo.PyLayerDesc()

    ld._hiddenSize = pyogmaneo.PyInt3(36, 36, 6)

    lds.append(ld)

h = pyogmaneo.PyHierarchy(cs, [pyogmaneo.PyInt3(64, 64, 1)],
[pyogmaneo._inputTypePred], lds)

vis = np.empty((cam_height, cam_width, 3), dtype=np.uint8)

camera.capture(vis, "rgb", use_video_port=True)

vis = vis[resizedHeight - int(hiddenHeight):, :, :]

visGrey = 0.333 * ((vis[:, :, 0].T / 255.0) + (vis[:, :, 1].T / 255.0) + (vis[:, :,
2].T / 255.0))

filtered = visGrey.T.ravel().tolist()

h.activate(filtered, cs)

h.learn(0.1, cs) # навчання

#h.save("neural.net") # збереження
```

### 3.7 Загальна схема програмного комплексу

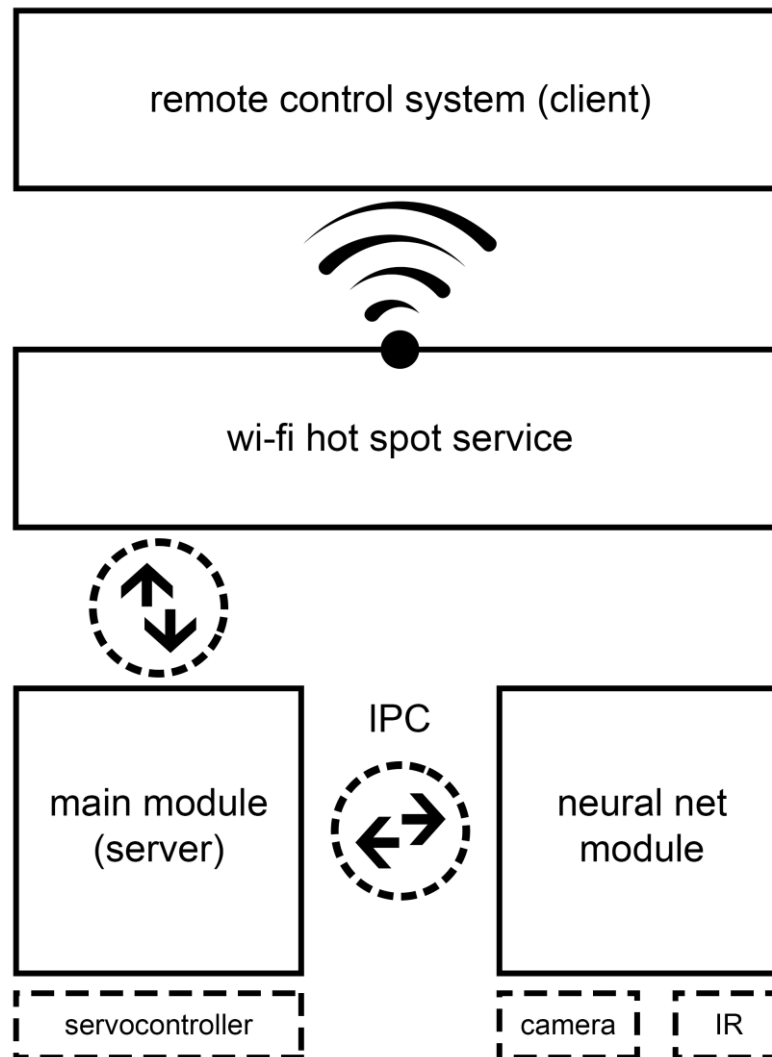


Рисунок 3.6 – Структурна схема програмного комплексу.

### 3.8 Висновки до розділу 3

У даному розділі було розглянуто процес проектування та написання відповідного програмного забезпечення для функціонування гексапода.

Цей розділ став фінальним етапом у практичному впровадженні нової інтелектуальної системи навігації. Тестування показало, що ІСН успішно функціонує та показує досить не погані результати в області обходу перешкод.

## **4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ. ЕКОЛОГІЯ**

В даному розділі проведено аналіз потенційних небезпечних та шкідливих виробничих факторів, причин пожеж. Розглянуті заходи, які дозволяють забезпечити гігієну праці і виробничу санітарію. На підставі аналізу розроблені заходи з техніки безпеки та рекомендації з пожежної профілактики.

Так як в процесі виконання магістерської роботи розроблявся дослідний зразок спроектованої інтелектуальної системи, то аналіз потенційно небезпечних і шкідливих виробничих чинників виконується для приміщення, де проводились роботи над дипломним проектом.

### **4.1 Загальні питання з охорони праці**

В законі України «Про охорону праці» визначається, що охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності.

При роботі з обчислювальною технікою змінюються фізичні і хімічні фактори навколишнього середовища: виникає статична електрика, електромагнітне випромінювання, змінюється температура і вологість, рівень вміст кисню і озону в повітрі. Повітря забруднюється шкідливими хімічними речовинами антропогенного походження за рахунок деструкції полімерних матеріалів, які використовуються для обробки приміщень та обладнання. Неправильна організація робочого місця сприяє загальному і локальній напрузі м'язів ший, тулуба, верхніх кінцівок, викривлення хребта і розвитку остеохондрозу. На всіх підприємствах, в установах, організаціях повинні створюватися безпечні і нешкідливі умови праці.

#### **4.1.1 Правові та організаційні основи охорони праці**

Основним організаційним напрямом у здійсненні управління в сфері охорони праці є усвідомлення пріоритету безпеки праці і підвищення соціальної відповідальності держави, і особистої відповідальності працівників.

Користувачі персональних комп'ютерів, для яких ця робота є головною, підлягають медичним оглядам: попереднім — під час влаштування на роботу і періодичним — протягом професійної діяльності раз на два роки. Жінок з часу встановлення вагітності та в період годування дитини грудьми до роботи з ПК не допускають.

Обов'язки працівників щодо додержання вимог нормативно-правових актів з охорони праці (ст. 14), відповідальність робітників всіх категорій за порушення вимог щодо охорони праці (ст. 44) та структура організації/виробництв системи управління охорони праці визначені у [13].

#### **4.1.2 Організаційно-технічні заходи з безпеки праці**

В організації/підприємстві проводиться навчання і перевірка знань з питань охорони праці відповідно до вимог Типового положення про порядок проведення навчання і перевірки знань з питань охорони праці, затвердженого наказом Держнаглядохоронпраці України від 26.01.2005 N 15, зареєстрованого в Міністерстві юстиції України 15.02.2005 за N 231/10511 [14].

Також впроваджені організаційні заходи з пожежної безпеки - навчання і перевірку знань відповідно до вимог Типового положення про інструктажі, спеціальне навчання та перевірку знань з питань пожежної безпеки на підприємствах, в установах та організаціях України, затвердженого наказом Міністерства України з питань надзвичайних ситуацій та у справах захисту населення від наслідків Чорнобильської катастрофи від 29.09.2003 N 368, зареєстрованого в Міністерстві юстиції України 11.12.2003 за N 1148/8469 [15].

### **4.2 Аналіз стану умов праці**

Робота над створенням дослідного зразка (гексапода) проходить в побутовому приміщенні. Для даної роботи достатньо однієї людини, для якої надано робоче місце зі стаціонарним комп'ютером.

#### **4.2.1 Вимоги до приміщень**

Геометричні розміри приміщення зазначені у таблиці 4.1. Для зручності спільної роботи з іншими працівниками (обговорення ідей, з'ясування проблем і т.д.) в кімнаті є диван і журнальний стіл, обставлені живими квітами. Також робочий процес пов'язаний з багатьма документами, теками, журналами для чого приміщення облаштоване принтером і шафою для зручності. Задля дотримання визначеного рівня мікроклімату в будівлі встановлено систему опалення та кондиціонування. Для забезпечення потрібного рівня освітленості кімната має вікно та систему загального рівномірного освітлення, що встановлена на стелі. Для дотримання вимог пожежної безпеки встановлено порошковий вогнегасник та систему автоматичної пожежної сигналізації.

Таблиця 4.1 – Розміри приміщення

Найменування	Значення
Довжина, м	6
Ширина, м	4
Висота, м	2.5
Площа, м <sup>2</sup>	24
Об'єм, м <sup>3</sup>	60

Згідно до санітарних норм мікроклімату виробничих приміщень [16], розмір площі для одного робочого місця оператора персонального комп'ютера має бути не менше 6 кв. м, а об'єм — не менше 20 куб. м. Отже, дане приміщення цілком відповідає зазначеним нормам.

#### 4.2.2 Вимоги до організації місця праці

При порівнянні відповідності характеристик робочого місця нормативним основні вимоги до організації робочого місця [17] (табл. 4.2) і відповідними фактичними значеннями для робочого місця, констатуємо повну відповідність.

Таблиця 4.2 – Характеристики робочого місця

Найменування параметра	Фактичне значення	Нормативне значення
Висота робочої поверхні, мм	750	680 ÷ 800
Висота простору для ніг, мм	730	≥ 600
Ширина простору для ніг, мм	660	≥ 500
Глибина простору для ніг, мм	700	≥ 650
Висота поверхні сидіння, мм	470	400 ÷ 500
Ширина сидіння, мм	400	≥ 400
Глибина сидіння, мм	400	≥ 400
Висота поверхні спинки, мм	600	≥ 300
Ширина опорної поверхні спинки, мм	500	≥ 380
Радіус кривини спинки в горизонтальній площині, мм	400	400
Відстань від очей до екрану дисплея, мм	800	700 ÷ 800

Робочий стіл на досліджуваному місці також містить достатньо простору для ніг. Крісло, що використовується в якості робочого сидіння, є поворотним, має підлокітники і можливість регулювання за висотою і кутом нахилу спинки, також воно м'яке і виконане з екологічної шкіри, що дає можливість працювати у комфорті. Екран монітору знаходиться на відстані 0.8 м, клавіатура має можливість регулювання кута нахилу 5-15°. Отже, за всіма параметрами робоче місце відповідає нормативним вимогам. Приміщення кабінету знаходиться на першому поверсі двох поверхової будівлі і має об'єм 60 м<sup>3</sup>, площу – 24 м<sup>2</sup>. У цьому кабінеті обладнано одне робоче місце, яке укомплектовано 2 ПК, один з котрих сервер без наявності пристроїв I/O інформації. Температура в приміщенні протягом року коливається у межах 18– 24°C, відносна вологість — близько 50%. Швидкість руху повітря не перевищує 0,2 м/с. Шум у приміщенні знаходиться на рівні 50 дБА. Система вентилявання приміщення — природна, а опалення — централізоване. Розміщення вікон забезпечує природне освітлення з коефіцієнтом природного освітлення не менше 1,5%, а загальне штучне освітлення, яке здійснюється за допомогою восьми люмінесцентних ламп, забезпечує рівень освітленості не менше 200Лк. У кабінеті є електрична мережа з напругою 220 В, яка створює небезпеку ураження електричним струмом. ПК та периферійні пристрої можуть бути джерелами електромагнітних випромінювань, аерозолів та шкідливих речовин (часток тонеру, оксидів нітрогену та озону). За ступенем пожежної безпеки приміщення належить до категорії В. Кабінет оснащений переносним вуглекислотним вогнегасником ВВК-5. Наявна аптечка для надання долікарської допомоги, а також у кабінеті роблять вологе прибирання та щоденно провітрюють приміщення.

#### **4.2.3 Навантаження та напруженість процесу праці**

За фізичним навантаженням робота відноситься до категорії легкі роботи (Ia), її виконують сидячи з періодичним ходінням. Щодо характеру організування виконання дипломної роботи, то він підпадає під нав'язаний режим, оскільки певні розділи роботи необхідно виконати у встановлені конкретні терміни. За ступенем нервово-психічної напруги виконання роботи можна віднести до II – III ступеня і кваліфікувати як помірно напружений – напружений за умови успішного виконання поставлених завдань.

Під час виконання робіт використовують ПК та периферійні пристрої (лазерні та струменеві), що призводить до навантаження на окремі системи організму. Такі перекосяти у напруженні різних систем організму, що трапляються під час роботи з ПК, зокрема, значна напруженість зорового аналізатора і довготривале малорухоме положення перед екраном, не тільки не зменшують загального напруження, а навпаки, призводять до його посилення і появи стресових реакцій.



Найбільшому ризику виникнення різноманітних порушень піддаються: органи зору, м'язово-скелетна система, нервово-психічна діяльність, репродуктивна функція у жінок. Тобто наявне психофізіологічні небезпечні та шкідливі фактори:

- a) фізичного перевантаження:
  - a. статичного;
  - b. динамічного;
- b) нервово-психічного перевантаження:
  - a. розумового перенапруження;
  - b. монотонності праці;
  - c. перенапруження аналізаторів
  - d. емоційних перевантажень.

Рекомендовано застосування екранних фільтрів, локальних світлофільтрів (засобів індивідуального захисту очей) та інших засобів захисту, а також інші профілактичні заходи [17].

Роботу за дипломним проектом визнано, таку, що займає 50% часу робочого дня та за восьмигодинної робочої зміни рекомендовано встановити додаткові регламентовані перерви: – для розробників програм тривалістю 15 хв через кожну годину роботи.

### **4.3 Виробнича санітарія**

На підставі аналізу небезпечних та шкідливих факторів при виробництві (експлуатації), пожежної безпеки можуть бути надалі вирішені питання необхідності забезпечення працюючих достатньою кількістю освітлення, вентиляції повітря, організації заземлення, тощо.

#### **4.3.1 Аналіз небезпечних та шкідливих факторів при виробництві виробу**

Аналіз небезпечних та шкідливих виробничих факторів виконується у табличній формі (табл. 4.3). Роботу, пов'язану з ЕОМ з ВДТ, у тому числі на тих, які мають робочі місця, обладнані ЕОМ з ВДТ і ПП, виконують із забезпеченням виконання [18], які встановлюють вимоги безпеки до обладнання робочих місць, до роботи із застосуванням ЕОМ з ВДТ і ПП. Переважно роботи за проектами виконують у кабінетах чи інших приміщеннях, де використовують різноманітне електрообладнання, зокрема персональні комп'ютери (ПК) та периферійні пристрої. Основними робочими характеристиками персонального комп'ютера є:

- робоча напруга  $U = +220\text{В} \pm 5\%$ ;
- робочий струм  $I = 3\text{А}$ ;
- спожива потужність  $P = 750\text{Вт}$ .

Таблиця 4.3 – Аналіз небезпечних і шкідливих виробничих факторів

Небезпечні і шкідливі виробничі фактори	Джерела факторів (види робіт)	Кількісна оцінка	Нормативні документи
1	2	3	4
фізичні			
Підвищена температура поверхонь обладнання	експлуатація ЕОМ, серверного обладнання для роботи	2	[19]
підвищений рівень шуму на робочому місці	-//-	2	[20]
підвищена або знижена вологість повітря	-//-	2	[19]
підвищена або знижена рухливість повітря	-//-	1	[19]
підвищений рівень напруги електричної мережі, замикання якої може відбутися через тіло людини	-//-	4	[21] [22]
недостатність природного світла	порушення умов праці (вимог до приміщень)	2	[23]
недостатнє освітлення робочої зони	порушення гігієнічних параметрів виробничого середовища	3	[23]
підвищена яскравість світла	порушення умов праці (організації місця праці - налагодження моніторів)	1	[17]
понижена контрастність	-//-	1	[17]
психофізіологічні:			
нервово-психічне перевантаження (розумове, перенапруження аналізаторів-зорових)	- пошук інформації для постановки теми; - пошук та аналіз аналогів і літератури; - пошук наявних технологій, моделювання та аналіз алгоритмів; - виконання роботи за темою диплома, тестування; - оформлення роботи	4	[18] [17]
фізичні (статичне – сидіння)	порушення умов праці та організації робочого часу	2	[18] [17]

Робочі місця мають відповідати вимогам Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин, затверджених

постановою Головного державного санітарного лікаря України від 10.12.98 N 7 [17]. За умов роботи з ПК виникають наступні небезпечні та шкідливі чинники: несприятливі мікрокліматичні умови, освітлення, електромагнітні випромінювання, забруднення повітря шкідливими речовинами (джерелом, яких можуть бути: принтер, сканер та інші джерела виділення багатьох хімічних речовин - наприклад, озону, оксидів азоту та аерозолів високодисперсних частинок тонера), шум, вібрація, електричний струм, електростатичне поле, напруженість трудового процесу та інше.

#### 4.3.2 Пожежна безпека

В приміщенні наявна затверджена «План-схема евакуації з кабінету (приміщення)».

Пожежна безпека при застосуванні ЕОМ забезпечується:

- системою запобігання пожежі,
- системою протипожежного захисту,
- організаційно-технічними заходами.

Згідно [24] таке приміщення, площею 24 м<sup>2</sup>, відноситься до категорії "В" (пожежонебезпечної) та для протипожежного захисту в ньому проектом передбачено устаткування автоматичною пожежною сигналізацією із застосуванням датчиків-сповіщувачів РІД-1 (сповіщувач димовий ізоляційний) в кількості 1 шт., і застосуванням первинних засобів пожежогасіння. Відповідно до норм первинних засобів пожежогасіння пропонується використовувати:

- ручний вуглекислий вогнегасник ОУ-5 в кількості 1 шт.;
- ковдра 1 м<sup>2</sup>, кошму 2×1,5 м<sup>2</sup> або азбестове полотно 2×2 м<sup>2</sup> в кількості 1 шт.

#### 4.3.3 Електробезпека

На робочому місці виконуються наступні вимоги електробезпеки: ПК, периферійні пристрої та устаткування для обслуговування, електропроводи і кабелі за виконанням та ступенем захисту відповідають класу зони за ПУЕ (правила улаштування електроустановок), мають апаратуру захисту від струму короткого замикання та інших аварійних режимів. Лінія електромережі для живлення ПК, периферійних пристроїв і устаткування для обслуговування, виконана як окрема групова три- провідна мережа, шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів. Штепсельні з'єднання та електророзетки крім контактів фазового та нульового робочого провідників мають спеціальні контакти для підключення нульового захисного провідника. Електромережа штепсельних розеток для

живлення персональних ПК, укладено по підлозі поруч зі стінами відповідно до затвердженого плану розміщення обладнання та технічних характеристик обладнання. Металеві труби та гнучкі металеві рукави заземлені. Захисне заземлення включає в себе заземлюючих пристроїв і провідник, який з'єднує заземлюючий пристрій з обладнанням, яке заземлюється - заземлюючий провідник.

#### 4.4 Гігієнічні вимоги параметрів виробничого середовища

##### 4.4.1 Мікроклімат

Мікроклімат робочих приміщень – це клімат внутрішнього середовища цих приміщень, що визначається діючої на організм людини з'єднанням температури, вологості, швидкості переміщення повітря. В даному приміщенні проводяться роботи, що виконуються сидячи і не потребують динамічного фізичного напруження, то для нього відповідає категорія робіт Ia. Отже оптимальні значення для температури, відносної вологості й рухливості повітря для зазначеного робочого місця відповідають [19] і наведені в табл. 4.4:

Таблиця 4.4 – Норми мікроклімату робочої зони об'єкту

Період року	Категорія робіт	Температура С <sup>0</sup>	Відносна вологість %	Швидкість руху повітря, м/с
Холодна	легка-1 а	22 - 24	40 – 60	0,1
Тепла	легка-1 а	23 - 25	40 – 60	0,1

Дане приміщення обладнане системами опалення, кондиціонування повітря або припливно-втяжною вентиляцією. У приміщенні на робочому місці забезпечуються оптимальні значення параметрів мікроклімату: температури, відносної вологості й рухливості повітря у відповідності до [19]. Рівні позитивних і негативних іонів у повітрі мають відповідати [19]. Для забезпечення оптимальних параметрів мікроклімату в приміщенні проводяться перерви в роботі співробітників, з метою його провітрювання. Існують спеціальні системи кондиціонування, які забезпечують підтримання в приміщенні балансу оптимальних параметрів мікроклімату.

Контроль параметрів мікроклімату в холодний і теплий період року здійснюється не менше 3-х разів на зміну (на початку, середині, в кінці).

#### 4.4.2 Освітлення

Світло є природною умовою існування людини. Воно впливає на стан вищих психічних функцій і фізіологічні процеси в організмі. Хороше освітлення діє тонізуюче, створює гарний настрій, покращує протікання основних процесів вищої нервової діяльності.

Збільшення освітленості сприяє поліпшенню працездатності навіть в тих випадках, коли процес праці практично не залежить від зорового сприйняття. При поганому освітленні людина швидко втомлюється, працює менш продуктивно, виникає потенційна небезпека помилкових дій і нещасних випадків.

Освітленість приміщення має велике значення при роботі на ПЕОМ. Вона багато в чому визначається колірною і мережевий обстановкою. Для зменшеного поглинання світла стеля і стіни вище панелей (1,5 – 1,7 м.). Якщо вони не облицьовані звукопоглинальним матеріалом, фарбуються білою водоемульсійною фарбою (коефіцієнт відбиття повинен бути не менше 0,7).

Для забарвлення стіни панелей рекомендується віддавати перевагу світлим фарбам.

Основний потік природного світла при цьому повинен бути зліва. Не допускається спрямування основного світлового потоку природного світла праворуч, ззаду і спереду працівника на ПЕОМ.

Робота на ПЕОМ може здійснюватися за таких видах освітлення:

- загальному штучному освітленні, коли відео монітори розташовуються по периметру приміщення або при центральному розташуванні робочих місць у два ряди по довжині кімнати з екранами, звернені в протилежні сторони;
- суміщене освітлення (природне + штучне) тільки при одному і трьох рядном розташуванні робочих місць, коли екран і поверхня робочого столу знаходяться перпендикулярно світла несучій стіні. При цьому штучне освітлення буде виконане стельовими або підвісними люмінесцентними світильниками, рівномірно розміщеними по стелі рядами паралельно світловим прорізам так, щоб екран відео монітора знаходився в зоні захисного кута світильника, і його проекції не доводилися на екран. Працюючі на ПЕОМ не повинні бачити відображення світильників на екрані. Застосовувати місцеве освітлення при роботі на ПЕОМ не рекомендується.
- природне освітлення, коли робочі місця з ПЕОМ розташовуються в один ряд по довжині приміщення на відстані 0,8 – 1,0 м від стіни з віконними прорізами, і екрани знаходяться перпендикулярно цієї стіни. Основний потік природного світла при цьому повинен бути зліва. Не допускається спрямування основного світлового потоку природного світла праворуч, ззаду і спереду. Оптимальна відстань очей до екрана відео монітора повинна становити 60-70 см. Розглядати інформацію ближче 50 см не рекомендується.

У проекті, що розробляється, передбачається використовувати суміщене освітлення. У світлий час доби використовуватиметься природне освітлення приміщення через віконні отвори, в решту часу використовуватиметься штучне освітлення. Штучне освітлення створюється газорозрядними лампами.

Штучне освітлення в робочому приміщенні передбачається здійснювати з використанням люмінесцентних джерел світла в світильниках загального освітлення, оскільки люмінесцентні лампи мають високу потужність (80 Вт), тривалий термін служби (до 10000 годин), та спектральний склад випромінюваного світла близький до сонячного. При експлуатації ЕОМ виконується зорова робота IVв розряду точності (середня точність). При цьому нормована освітленість на робочому місці ( $E_n$ ) рівна 200 лк. Джерелом природного освітлення є сонячне світло.

У приміщенні, де розташовані ЕОМ передбачається природне бічне освітлення, рівень якого відповідає нормам [23]. Джерелом природного освітлення є сонячне світло. Регулярно повинен проводитися контроль освітленості, який підтверджує, що рівень освітленості задовольняє ДБН і для даного приміщення в світлий час доби достатньо природного освітлення.

### **Розрахунок освітлення**

Для виробничих та адміністративних приміщень світловий коефіцієнт приймається не менше  $1/8$ , в побутових –  $1/10$ :

$$S_b = \left( \frac{1}{5} \div \frac{1}{10} \right) \times S_n, \quad (4.1)$$

де  $S_b$  – площа віконних прорізів,  $m^2$ ;

$S_n$  – площа підлоги,  $m^2$ .

$$S_n = a \cdot b = 6 \cdot 4 = 24 \text{ м}^2,$$

$$S = 1/10 \cdot 24 = 2,4 \text{ м}^2.$$

Приймаємо 1 вікно площею  $S=2,4 \text{ м}^2$ .

Світильники загального освітлення розташовуються над робочими поверхнями в рівномірно-прямокутному порядку. Для організації освітлення в темний час доби передбачається обладнати приміщення, довжина якого складає 6 м, ширина 4 м, світильниками ЛПО2П, оснащеними лампами типу ЛБ (дві по 80 Вт) з світловим потоком 3200 лм кожна.

Розрахунок штучного освітлення виробляється по коефіцієнтах використання світлового потоку, яким визначається потік, необхідний для створення заданої освітленості при загальному рівномірному освітленні. Розрахунок кількості світильників  $n$  виробляється по формулі (4.2):

$$n = \frac{E \times S \times Z \times K}{F \times U \times M}, \quad (4.2)$$

де  $E$  – нормована освітленість робочої поверхні, визначається нормами – 300 лк;

$S$  – освітлювана площа, м<sup>2</sup>;  $S = 24$  м<sup>2</sup>;

$Z$  – поправочний коефіцієнт світильника (1,1 для люмінесцентних ламп);

$K$  – коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації – 1,5;

$U$  – коефіцієнт використання, залежний від типу світильника, показника індексу приміщення і т.п. – 0,575

$M$  – число люмінесцентних ламп в світильнику – 2;

$F$  – світловий потік лампи – 3200лм.

Підставивши числові значення у формулу (4.2), отримуємо:

$$n = \frac{300 \times 24 \times 1.1 \times 1.5}{3200 \times 0.575 \times 2} = 3,2$$

Приймаємо освітлювальну установку, яка складається з 3-х світильників, які складаються з 2-х люмінесцентних ламп загальною потужністю 80 Вт, напругою – 220 В.

#### 4.4.3 Вентилювання

У приміщенні, де знаходяться ЕОМ, повітрообмін реалізується за допомогою природної організованої вентиляції (вентиляційні шахти), тобто при  $V$  приміщення  $> 40$  м<sup>3</sup> на одного працюючого допускається природна вентиляція. Цей метод забезпечує приток потрібної кількості свіжого повітря, що визначається в СНіП.

Також має здійснюватися провітрювання приміщення, в залежності від погодних умов, тривалість повинна бути не менше 10 хв. Найкращий обмін повітря здійснюється при наскрізному провітрюванні.

#### 4.5 Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій

Відповідно до санітарно-гігієнічних нормативів та правил експлуатації обладнання наводимо приклади деяких заходів безпеки.

1) Заходи безпеки під час експлуатації персонального комп'ютера та периферійних пристроїв передбачають:

- правильне організування місця праці та дотримання оптимальних режимів праці та відпочинку під час роботи з ПК;
- експлуатацію сертифікованого обладнання;
- дотримання заходів електробезпеки;
- забезпечення оптимальних параметрів мікроклімату;
- забезпечення раціонального освітлення місця праці (освітленість робочого місця не перевищувала 2/3 нормальної освітленості приміщення);

облаштовуючи приміщення для роботи з ПК, потрібно передбачити припливно-витяжну вентиляцію або кондиціонування повітря:

- якщо об'єм приміщення 20 м<sup>3</sup>, то потрібно подати не менш як 30 м<sup>3</sup>/год повітря;
- якщо об'єм приміщення у межах від 20 до 40 м<sup>3</sup>, то потрібно подати не менш як 20 м<sup>3</sup>/год повітря;
- якщо об'єм приміщення становить понад 40 м<sup>3</sup>, допускається природна вентиляція, у випадку, коли немає виділення шкідливих речовин.

зниження рівня шуму та вібрації:

- у джерелі виникнення, шляхом застосування раціональних конструкцій, нових матеріалів і технологічних процесів;
- звукоізолювання устаткування за допомогою глушників, резонаторів, кожухів, захисних конструкцій, оздоблення стін, стелі, підлоги тощо;
- використання засобів індивідуального захисту).

2) Заходи безпеки під час експлуатації інших електричних приладів передбачають дотримання таких правил:

- постійно стежити за справним станом електромережі, розподільних щитків, вимикачів, штепсельних розеток, лампових патронів, а також мережевих кабелів живлення, за допомогою яких електроприлади під'єднують до електромережі;
- постійно стежити за справністю ізоляції електромережі та мережевих кабелів, не допускаючи їхньої експлуатації з пошкодженою ізоляцією;
- не тягнути за мережевий кабель, щоб витягти вилку з розетки;
- не закривати меблями, різноманітним інвентарем вимикачі, штепсельні розетки;
- не підключати одночасно декілька потужних електропристроїв до однієї розетки, що може викликати надмірне нагрівання провідників, руйнування їхньої ізоляції, розплавлення і загоряння полімерних матеріалів;
- не залишати включені електроприлади без нагляду;



- не допускати потрапляння всередину електроприладів крізь вентиляційні отвори рідин або металевих предметів, а також не закривати їх та підтримувати в належній чистоті, щоб уникнути перегрівання та займання приладу;
- не ставити на електроприлади матеріали, які можуть під дією теплоти, що виділяється, загорітися (канцелярські товари, сувенірну продукцію тощо).

Від ураження струмом застосовують різні електричні захисні засоби:

- Ізолюючі – ізолюють людини від струмоведучих або заземлених частин, а так-же від землі. Вони діляться на основні та додаткові.
- Основні – володіють ізоляцією, здатної довго витримувати робоче напругу електроустановки і тому ними дозволяється стосуватися струмоведучих частин, знаходячи-трудящих під напругою.
- Запобіжні – володіють ізоляцією нездатною витримати робоча напруга електроустановки, і тому вони не можуть самостійно захищати людину від ураження струмом під цим напругою. Їх значення - посилити захисні дії основних і ізолюючих засобів, разом з якими вони повинні застосовуватися, при чому при використанні основних захисних засобів достатньо застосування одного запобіжного захисного засобу.

#### 4.5.1 Розрахунок захисного заземлення

Згідно з класифікацією приміщень за ступенем небезпеки ураження електричним струмом, приміщення в якому проводяться всі роботи відноситься до першого класу (без підвищеної небезпеки). Під час роботи використовуються електроустановки з напругою живлення 36 В, 220 В, та 360 В. Опір контуру заземлення повинен мати не більше 4 Ом.

Розрахунок проводять за допомогою методу коефіцієнта використання (екранування) електродів. Коефіцієнт використання групового заземлювача  $\eta$  – це відношення діючої провідності цього заземлювача до найбільш можливої його провідності за нескінченно великих відстаней між його електродами. Коефіцієнт використання вертикальних заземлювачів  $\eta_v$  в залежності від розміщення заземлювачів та їх кількості знаходиться в межах 0,4...0,99. Взаємну екрануючу дію горизонтального заземлювача (з'єднувальної смуги) враховують за допомогою коефіцієнта використання горизонтального заземлювача  $\eta_c$ .

#### Послідовність розрахунку:

1) Визначається необхідний опір штучних заземлювачів  $R_{шт.з.}$ :

$$R_{\text{шт.з.}} = \frac{R_{\text{д}} \cdot R_{\text{пр.з.}}}{R_{\text{пр.з.}} - R_{\text{д}}}, \quad (4.3)$$

де  $R_{\text{пр.з.}}$  – опір природних заземлювачів;  $R_{\text{д}}$  – допустимий опір заземлення. Якщо природні заземлювачі відсутні, то  $R_{\text{шт.з.}} = R_{\text{д}}$ .

Підставивши числові значення у формулу (4.3), отримуємо:

$$R_{\text{шт.з.}} = \frac{4 \cdot 40}{40 - 4} \approx 4 \text{ Ом}$$

2) Опір заземлення в значній мірі залежить від питомого опору ґрунту  $\rho$ , Ом·м. Приблизне значення питомого опору глини приймаємо  $\rho = 40$  Ом·м (табличне значення).

3) Розрахунковий питомий опір ґрунту,  $\rho_{\text{розр.}}$ , Ом·м, визначається відповідно для вертикальних заземлювачів  $\rho_{\text{розр.в}}$ , і горизонтальних  $\rho_{\text{розр.г}}$ , Ом·м за формулою:

$$\rho_{\text{розр.}} = \Psi \cdot \rho \quad (4.4)$$

де  $\Psi$  – коефіцієнт сезонності для вертикальних заземлювачів і кліматичної зони з нормальною вологістю землі, приймається для вертикальних заземлювачів  $\rho_{\text{розр.в}} = 1,7$  і горизонтальних  $\rho_{\text{розр.г}} = 5,5$  Ом·м.

$$\rho_{\text{розр.в}} = 1,7 \cdot 40 = 68 \text{ Ом} \cdot \text{м}$$

$$\rho_{\text{розр.г}} = 5,5 \cdot 40 = 220 \text{ Ом} \cdot \text{м}$$

4) Розраховується опір розтікання струму вертикального заземлювача  $R_{\text{в}}$ , Ом, за (4.5).

$$R_{\text{в}} = \frac{\rho_{\text{розр.в}}}{2 \cdot \pi \cdot l_{\text{в}}} \cdot \left( \ln \frac{2 \cdot l_{\text{в}}}{d_{\text{ст}}} + \frac{1}{2} \cdot \ln \frac{4 \cdot t + l_{\text{в}}}{4 \cdot t - l_{\text{в}}} \right), \quad (4.5)$$

де  $l_{\text{в}}$  – довжина вертикального заземлювача (для труб – 2 – 3 м;  $l_{\text{в}} = 3$  м);

$d_{\text{ст}}$  – діаметр стержня (для труб – 0,03 – 0,05 м;  $d_{\text{ст}} = 0,05$  м);

$t$  – відстань від поверхні землі до середини заземлювача, яка визначається за ф. (4.6):

$$t = h_E + \frac{1_E}{2}, \quad (4.6)$$

де  $h_B$  – глибина закладання вертикальних заземлювачів (0,8 м); тоді

$$t = 0,8 + \frac{3}{2} = 2,3 \text{ м};$$

$$R_B = \frac{68}{2 \cdot \pi \cdot 3} \cdot \left( \ln \frac{2 \cdot 3}{0,05} + \frac{1}{2} \cdot \ln \frac{4 \cdot 2,3 + 3}{4 \cdot 2,3 - 3} \right) = 18,5 \text{ Ом}$$

- 1) Визначається теоретична кількість вертикальних заземлювачів  $n$  штук, без урахування коефіцієнта використання  $\eta_B$ :

$$n = \frac{2R_E}{R_D} = \frac{2 \times 18,5}{4} = 9,25 \quad (4.7)$$

І визначається коефіцієнт використання вертикальних електродів групового заземлювача без врахування впливу з'єднувальної стрічки  $\eta_B = 0,57$  (табличне значення).

- 2) Визначається необхідна кількість вертикальних заземлювачів з урахуванням коефіцієнта використання  $\eta_B$ , шт:

$$n = \frac{2 \cdot R_E}{R_D \cdot \eta_B} = \frac{2 \cdot 18,5}{4 \cdot 0,57} \approx 16 \quad (4.8)$$

- 3) Визначається довжина з'єднувальної стрічки горизонтального заземлювача  $l_C$ , м:

$$l_C = 1,05 \cdot L_B \cdot (n_B - 1), \quad (4.9)$$

де  $L_B$  – відстань між вертикальними заземлювачами, (прийняти за  $L_B = 3\text{м}$ );

$n_B$  – необхідна кількість вертикальних заземлювачів.

$$l_C = 1,05 \cdot 3 \cdot (16 - 1) \approx 48 \text{ м}.$$

Визначається опір розтіканню струму горизонтального заземлювача (з'єднувальної стрічки)

$R_{\Gamma}$ , Ом:

$$R_{\Gamma} = \frac{\rho_{\text{розр.}\Gamma}}{2 \cdot \pi \cdot l_c} \cdot \ln \frac{2 \cdot l_c^2}{d_{\text{см}} \cdot h_{\Gamma}}, \quad (4.10)$$

де  $d_{\text{см}}$  – еквівалентний діаметр смуги шириною  $b$ ,  $d_{\text{см}} = 0,95b$ ,  $b = 0,15$  м;

$h_{\Gamma}$  – глибина закладання горизонтальних заземлювачів (0,5 м);

$l_c$  – довжина з'єднувальної стрічки горизонтального заземлювача  $l_c$ , м

$$R_{\Gamma} = \frac{220}{2 \cdot \pi \cdot 48} \cdot \ln \frac{2 \cdot 48^2}{0,95 \cdot 0,15 \cdot 0,5} = 8,1 \text{ Ом}$$

4) Визначається коефіцієнт використання горизонтального заземлювача  $\eta_c$  відповідно до необхідної кількості вертикальних заземлювачів  $n_B$ .

Коефіцієнт використання з'єднувальної смуги  $\eta_c = 0,3$ .

Розраховується результуючий опір заземлювального електроду з урахуванням з'єднувальної смуги:

$$R_{\text{заг.}} = \frac{R_E \cdot R_{\Gamma}}{R_E \cdot n_c + R_{\Gamma} \cdot n_E \cdot n_E} \leq R_d, \quad (4.11)$$

Висновок: дане захисне заземлення буде забезпечувати електробезпеку будівлі, так як виконується умова:  $R_{\text{заг.}} < 4$  Ом, а саме:

$$R_{\text{заг.}} = \frac{18,5 \cdot 8,1}{18,5 \cdot 0,3 + 8,1 \cdot 16 \cdot 0,57} = 1,9 \leq R_d$$

При виникненню пожеж при роботі на ПЕОМ від таких можливими джерел запалювання як:

- іскри і дуги коротких замикань;
- перегрів провідників, резисторів та інших радіодеталей ПЕОМ, від тривалої перевантаження та наявність перехідного опору;
- іскри при розмиканні і розмиканні ланцюгів;
- розряди статичної електрики;

- необережному поводженню з вогнем, а також вибухи газо-повітряних і паро-повітряних сумішей.

Важливу увагу слід звернути на пожежну безпеку підприємства в цілому і окремих його приміщень. В приміщеннях не повинно накопичуватися сміття, непотрібний папір, мотлох та ін. речі, які не використовуються у виробничому процесі. Наявний вільний аварійний вихід за межі приміщення в разі пожежі, бути передбачені вогнегасники. Вони повинні бути в робочому стані і перевірятися згідно з нормами. У приміщеннях повинна бути пожежна сигналізація, вогнегасник. У разі виникнення пожежі необхідно повідомити в найближчу пожежну частину, убезпечити інших працівників і по можливості прийняти кроки по запобіганню можливих наслідків та усуненню пожежі.

## **4.6 Охорона навколишнього природного середовища**

### **4.6.1 Загальні дані з охорони навколишнього середовища**

Діяльність за темою магістерської роботи в процесі її виконання впливає на навколишнє природне середовище і регламентується нормами діючого законодавства: Законом України «Про охорону навколишнього природного середовища», Законом України «Про забезпечення санітарного та епідемічного благополуччя населення», Законом України «Про відходи», Законом України «Про охорону атмосферного повітря», Законом України «Про захист населення і територій від надзвичайних ситуацій техногенного та природного характеру», Водний кодекс України.

Основним екологічним аспектом в процесі діяльності за даними спеціальностями є процеси впливу на атмосферне повітря та процеси поводження з відходами, які утворюються, збираються, розміщуються, передаються на видалення (знешкодження), утилізацію, тощо в ІТ галузі.

Вплив на атмосферне повітря при нормальних умовах праці не оказує, бо не має в приміщенні сканерів, принтерів та інших джерел викиду забруднюючих речовин в повітря робочої зони.

В процесі діяльності розробника математичної моделі за допомогою ПК виникають процеси поводження з відходами ІТ галузі. Нижче надано перелік відходів, що утворюються в процесі роботи:

- Відпрацьовані люмінесцентні лампи - I клас небезпеки
- Акумулятор для джерел безперебійного живлення – III клас безпеки
- Макулатура - IV клас небезпеки

- Побутові відходи - IV клас небезпеки

#### **4.6.2 Вимоги до збору, пакування та розміщення відходів ІТ галузі**

Наводяться вимоги зберігання виявлених за своєю роботою відходів відповідно до вимог [25].

Відходи в міру їх накопичення збирають у тару, відповідну класу небезпеки, з дотриманням правил безпеки, після чого доставляють до місця тимчасового зберігання відходів відповідно до затвердженої схеми їх розміщення. Зазначені для зберігання відходів місця чи об'єкти повинні використовуватися лише для заявлених відходів.

Не допускається зберігання відходів у невстановлених схемою місцях, а також перевищення норм тимчасового зберігання відходів.

Способи тимчасового зберігання відходів визначаються видом, агрегатним станом і класом небезпеки відходів:

- Відходи I класу небезпеки зберігаються в герметичній тарі (сталеві бочки, контейнери). У міру наповнення тару з відходами закривають герметично сталевий кришкою;
- Відходи II класу небезпеки в залежності від агрегатного стану зберігаються в поліетиленових мішках, бочках, сховищах та інших видах тари, яка запобігає поширенню шкідливих речовин;
- Відходи III класу небезпеки зберігаються в тарі, яка забезпечує локалізацію зберігання, дозволяє виконувати вантажно-розвантажувальні і транспортні роботи і виключає поширення в ОС шкідливих речовин;
- Відходи IV класу небезпеки можуть зберігатися відкрито на промисловому майданчику у вигляді конусоподібної купи, звідки їх автотранспортом перевантажують у самоскид і до-ставляють на місце утилізації або захоронення;

#### **4.6.3 Визначення впливу та заходів щодо поводження з відходами ІТ галузі**

З метою визначення та прогнозування впливу відходів на навколишнє середовище, своєчасного виявлення негативних наслідків, їх запобігання відповідно до Закону України «Про відходи» повинен здійснюватися моніторинг місць утворення, зберігання, і видалення відходів. Відомості про місце утворення та місце розташування відходів зазначаються та наводяться у таблиці 4.5.

Таблиця 4.5 – Відомості про місце утворення та місце розташування відходів

№ з/п	Код та найменування відходів за ДК -005-96	Технологічний процес або виробництво, де утворюються відходи / клас небезпеки	Місце розташування відходу, тара та її кількість, місткість, розміри у разі наявності майданчиків розташування відходів необхідно зазначити тип покриття та наявність даху)
1	7710.3.1.26 Лампи люмінесцентні, та відходи, які містять ртуть, інші зіпсовані або відпрацьовані	1	буд.46, кв. 96
2	7710.3.1.01 Макулатура паперова та картонна (Макулатура)		буд.46, кв. 96
3	Акумулятор для джерел безперебійного живлення	3	буд.46, кв. 96

#### 4.7 Висновки до розділу 4

В результаті проведеної роботи було зроблено аналіз умов праці, шкідливих та небезпечних чинників, з якими стикається робітник. Було визначено параметри і певні характеристики приміщення для роботи над запропонованим проектом написаному в дипломній роботі, описано, які заходи потрібно зробити для того, щоб дане приміщення відповідало необхідним нормам і було комфортним і безпечним для робітника. Приведені рекомендації щодо організації робочого місця, а також важливу інформацію щодо пожежної та електробезпеки. Були наведені розміри приміщення та значення температури, вологості й рухливості повітря, необхідна кількість і потужність ламп та інші параметри, значення яких впливає на умови праці робітника, а також – наведені інструкції з охорони праці, техніки безпеки при роботі на комп'ютері.

А також визначені основні екологічні аспекти впливу на навколишнє природне середовище та зазначені заходи щодо поводження з ними

## ВИСНОВКИ

У даній магістерській атестаційній роботі були проаналізовані існуючі системи навігацій та запропонована та спроектована власна інтелектуальна система навігації, яка підходить до використання у недетермінованих середовищах.

Відповідно до спроектованої системи навігації було розроблено дослідний зразок – гексапод, який здатен до самостійного пересування. Створення гексапода полягало у проектуванні та впровадженні як апаратної частини проекту, так і програмної.

Відповідно до поставлених умов, гексапод може керуватися дистанційно за допомогою мобільного додатку на ОС Android. Це дає можливість використовувати гексапод в режимі налагодження та дозволяє використовувати мобільний додаток для корекції онлайн-навчання запропонованої нейромережі. Дистанційне керування використовує клієнт-серверну архітектуру з використанням пакетної передачі даних (сокетів) на базі стеку протоколів TCP/IP.

Тестування гексапода у недетермінованому середовищі підтвердило дієздатність спроектованої інтелектуальної системи, тому головну задачу магістерської роботи можна вважати виконаною. За допомогою інфрачервоного сенсора, гексапод може постійно коректувати свою нейромережу (перенастроювати ваги синапсів), що дозволяє у подальшому уникати перешкод та відмовитись від постійного контролю та корекції вчителя.

Наступним етапом досліджень та розробки стане використання акселерометру/гіроскопа та датчиків торкання (на усіх ногах гексапода). Це дозволить уникати вироджених випадків, коли гексапод стоїть частково на ногах та не має про це уяви. Також передбачається використання акселерометру/гіроскопу у якості вхідних даних нейромережі.

В ході виконання роботи були використані та покращенні навички з більшості предметів, отримані під час навчання. Також були отримані необхідні знання з охорони праці та екології навколишнього середовища.



**ПЕРЕЛІК ПОСИЛАНЬ**

- [1] Вікіпедія, «Одометрія,» [Онлайновий]. Available: <https://uk.wikipedia.org/wiki/%D0%9E%D0%B4%D0%BE%D0%BC%D0%B5%D1%82%D1%80%D1%96%D1%8F>. [Дата звернення: 2018].
- [2] С. И. С. В. Г. Б. Тузов В.А., «АВИОНИКА,» КИРОВОГРАДСКАЯ ЛЁТНАЯ АКАДЕМИЯ НАЦИОНАЛЬНОГО АВИАЦИОННОГО УНИВЕРСИТЕТА, 2013.
- [3] OgmaAI, «Ogma Intelligent Systems Corp – Neuroscience based Machine Intelligence,» [Онлайновий]. Available: <http://ogma.ai>. [Дата звернення: 2018].
- [4] Xenomai, «Xenomai | Real-time system,» [Онлайновий]. Available: <https://xenomai.org/>. [Дата звернення: 2018].
- [5] Amarpreet Singh Ugal (BIOS/Firmware Engineer Intel Corporation), «Hard Real Time Linux\* using Xenomai\* on Intel® Multi-Core Processors,» 2009.
- [6] Вікіпедія, «Сервопривод,» [Онлайновий]. Available: <https://uk.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%BE%D0%BF%D1%80%D0%B8%D0%B2%D0%BE%D0%B4>.
- [7] Lewan Soul, «LSC-24 Channel Servo Controller,» [Онлайновий]. Available: <http://www.lewansoul.com/product/detail-13.html>. [Дата звернення: 2018].
- [8] Raspberry Pi, «Raspberry Pi 2 Model B - Raspberry Pi,» [Онлайновий]. Available: <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>. [Дата звернення: 2018].
- [9] Adafruit Industries, «ADS1115 16-Bit ADC - 4 Channel with Programmable Gain Amplifier,» [Онлайновий]. Available: <https://www.adafruit.com/product/1085>. [Дата звернення: 2018].
- [10] S. Mann, Linux TCP/IP Network Administration, Prentice Hall, 2001.
- [11] E. Harold, Java Network Programming, 4th Edition, O'Reilly Media, 2013, p. 506.
- [12] RoboCraft, «Hexapod-робот под управлением ROS,» 2018. [Онлайновий]. Available: <http://robocraft.ru/blog/3113.html>.
- [13] НПАОП 0.00-6.03-93 «Порядок опрацювання та затвердження власником нормативних актів про охорону праці, що діють на підприємстві».
- [14] НПАОП 0.00-4.12-05 «Типове положення про порядок проведення навчання і перевірки знань з питань охорони праці».
- [15] НАПБ Б.02.005-2003 «Типове положення про інструктажі, спеціальне навчання та

перевірку знань з питань пожежної безпеки на підприємствах, в установах та організаціях України».

- [16] ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень».
- [17] ДСанПіН 3.3.2.007-98 «Правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин».
- [18] НПАОП 0.00.-1.28-10 «Правил охорони праці під час експлуатації електронно-обчислювальних машин».
- [19] ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень».
- [20] ДСН 3.3.6.037-99 «Санітарні норми виробничого шуму, ультразвуку та інфразвуку».
- [21] ГОСТ 12.1.030-81 «Електробезпека. Захисне заземлення, занулення».
- [22] ГОСТ 13109-97 «Електрична енергія. Сумісність технічних засобів. Норми якості електричної енергії в системах електропостачання загального призначення».
- [23] ДБН В.2.5-28:2015 «Державні Будівельні Норми України. Природне і штучне освітлення».
- [24] НАПБ Б.03.002-2007 «Норми визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою».
- [25] ДСанПіН 2.2.7.029 «Гігієнічні вимоги щодо поводження з промисловими відходами та визначення їх класу небезпеки для здоров'я населення»..

## ДОДАТОК А

## «Лістинг класу роботи з ТСП з'єднанням»

**(serversocket.hpp)**

```
#ifndef SERVERSOCKET_H
#define SERVERSOCKET_H

#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <unistd.h>
#include <cerrno>
#include "socket.hpp"

namespace net {
    class serversocket {
    protected:
        int port;
        int backlog;
        std::string address;

        int socketfd;

    public:

        serversocket(int port) {
            this->port = port;
            backlog = 10;
            address = "0.0.0.0";
        }

        serversocket(int port, int backlog) {
            this->port = port;
            this->backlog = backlog;
            address = "0.0.0.0";
        }

        serversocket(int port, int backlog, std::string address) {
            this->port = port;
            this->backlog = backlog;
            this->address = address;
        }

        ~serversocket();

        int listen();

        net::socket* accept();

        void close() {
            if (socketfd == -1) {
                return;
            }
        }
    };
}
```

```

        }

        ::close(socketfd);
    }

    bool valid() {
        return socketfd != -1;
    }

    int get_socket() {
        return socketfd;
    }
};
}

#endif

```

### **(serversocket.cpp)**

```

#include "serversocket.hpp"

net::serversocket::~serversocket() {
    close();
}

int net::serversocket::listen() {

    net::socketaddress* sockaddr = new net::socketaddress(address, port);
    struct sockaddr_in addr = sockaddr->get_struct();

    socketfd = ::socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);

    if (socketfd == -1) {
        return errno;
    }

    int yes = 1;
    if (::setsockopt(socketfd, SOL_SOCKET, SO_REUSEADDR, &yes, sizeof(int)) != 0) {
        close();
        return errno;
    }

    if (::bind(socketfd, (struct sockaddr*)&addr, sizeof(struct sockaddr)) != 0) {
        close();
        return errno;
    }

    if (::listen(socketfd, backlog) != 0) {
        close();
        return errno;
    }

    return 0;
}

net::socket* net::serversocket::accept() {

```

```

    struct sockaddr_in from;
    socklen_t l = sizeof(from);
    int clientfd = ::accept(socketfd, (struct sockaddr*)&from, &l);

    return new net::socket(clientfd, from);
}

```

### **(socket.hpp)**

```

#ifndef SOCKET_H
#define SOCKET_H

#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <string>
#include <unistd.h>
#include <cerrno>

#include "socketaddress.hpp"

namespace net {
    class socket {
    protected:
        int socketfd;
        struct sockaddr_in address;
        socketaddress* sockaddr;

    public:
        socket() {

        }

        socket(int socket, struct sockaddr_in addr) {
            socketfd = socket;
            address = addr;

            sockaddr = new socketaddress(addr);
        }

        ~socket();

        int read(std::string&);

        int read(char*, int);

        int send(std::string);

        int send(const char*, int, int);

        void set_blocking();

        void set_unblocking();

        void close() {
            if (socketfd == -1) {

```

```

        return;
    }

    ::close(socketfd);
}

bool valid() {
    return socketfd != -1;
}

int get_socket() {
    return socketfd;
}

socketaddress* get_socketaddress() {
    return sockaddr;
}
};
}

#endif

```

### **(socket.cpp)**

```

#include "socket.hpp"
#include <string>
#include <fcntl.h>

#define DEFAULT_SOCKET_BUFFER 128

net::socket::~~socket() {
    delete sockaddr;
    close();
}

void net::socket::set_blocking() {
    int opts = fcntl(socketfd, F_GETFL);
    opts = opts & (~O_NONBLOCK);
    fcntl(socketfd, F_SETFL, opts);
}

void net::socket::set_unblocking() {
    fcntl(socketfd, F_SETFL, O_NONBLOCK);
}

int net::socket::read(std::string& msg) {
    int bytes_total = 0;
    char buffer[DEFAULT_SOCKET_BUFFER];

    int bytes_read = recv(socketfd, buffer, DEFAULT_SOCKET_BUFFER, 0);

    if (bytes_read <= 0) {
        return bytes_read;
    }

    msg.append(std::string(buffer, 0, bytes_read));
}

```

```

bytes_total += bytes_read;

set_unblocking();

while (bytes_read > 0) {
    memset(buffer, 0, DEFAULT_SOCKET_BUFFER);
    bytes_read = recv(socketfd, buffer, DEFAULT_SOCKET_BUFFER, 0);

    if (bytes_read < 0) {
        break;
    }

    msg.append(std::string(buffer, 0, bytes_read));
    bytes_total += bytes_read;
}

set_blocking();

return bytes_total;
}

int net::socket::read(char* buf, int len) {
    return ::recv(socketfd, buf, len, 0);
}

int net::socket::send(std::string data) {
    return send(data.c_str(), data.length(), 0);
}

int net::socket::send(const char* buf, int len, int flags) {
    return ::send(socketfd, buf, len, flags);
}

```

### **(socketaddress.hpp)**

```

#ifndef SOCKET_ADDRESS_H
#define SOCKET_ADDRESS_H

#include <netinet/in.h>
#include <arpa/inet.h>
#include <string>
#include <cstring>

namespace net {
    class socketaddress {
    protected:
        int port;
        std::string address;

    public:
        socketaddress(struct sockaddr_in addr) {
            port = addr.sin_port;

            char ip[INET_ADDRSTRLEN];

```





ДОДАТОК Б

«Комп’ютерна презентація»

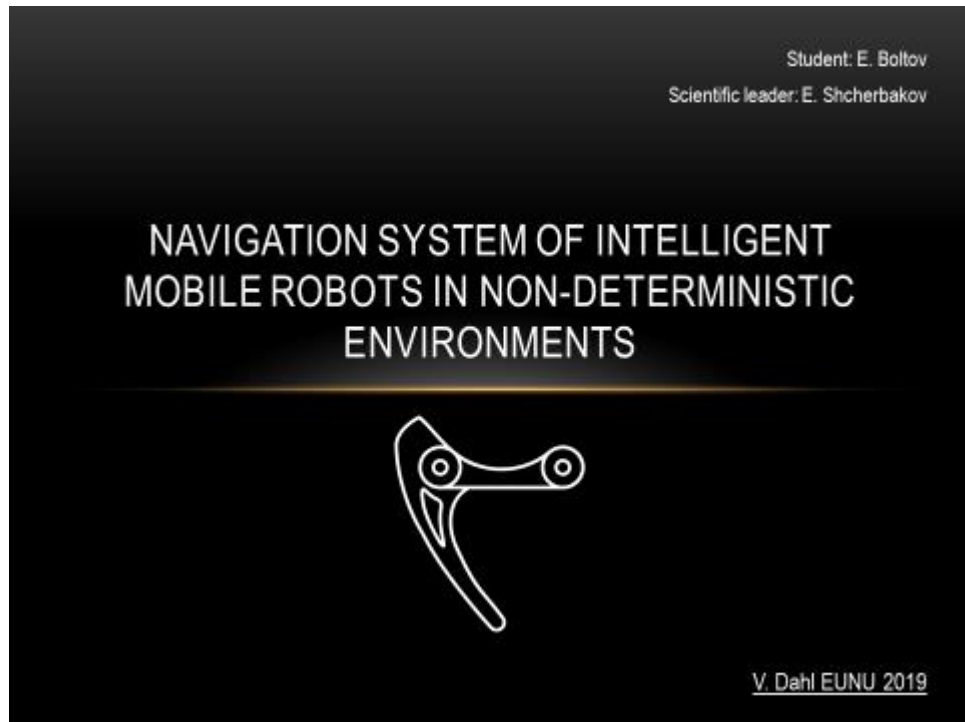


Рисунок Б.1 – Title.

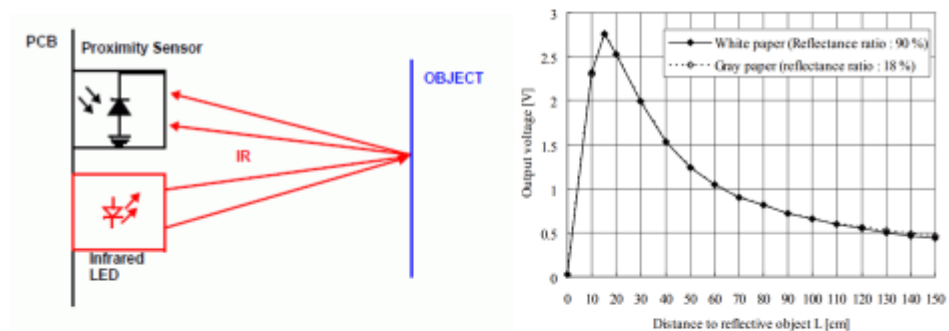
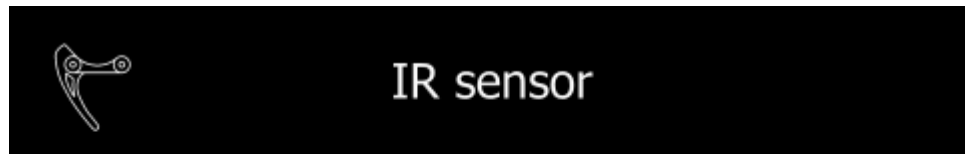


Рисунок Б.2 – IR sensor.

# Functional diagram

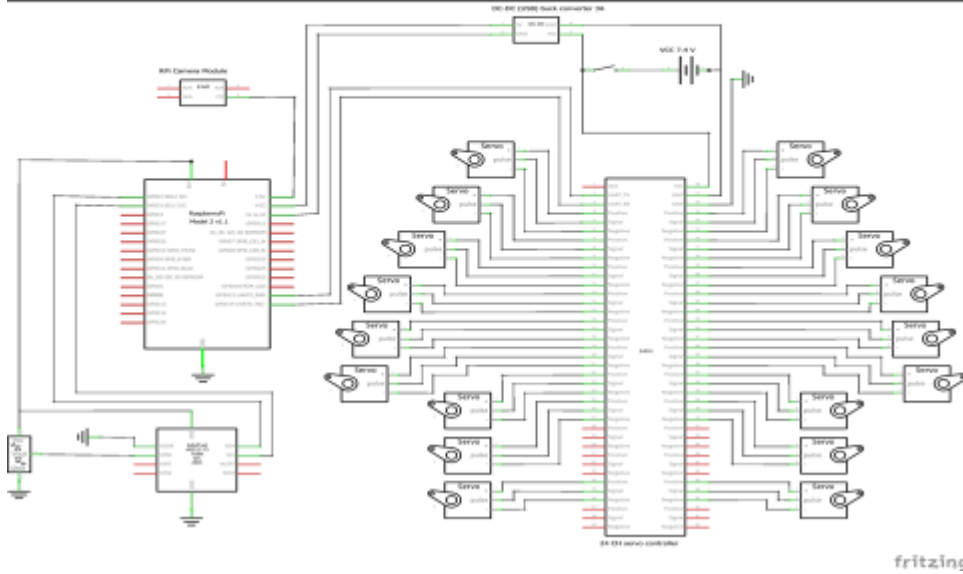


Рисунок Б.3 – Functional diagram.

# Hexapod gait

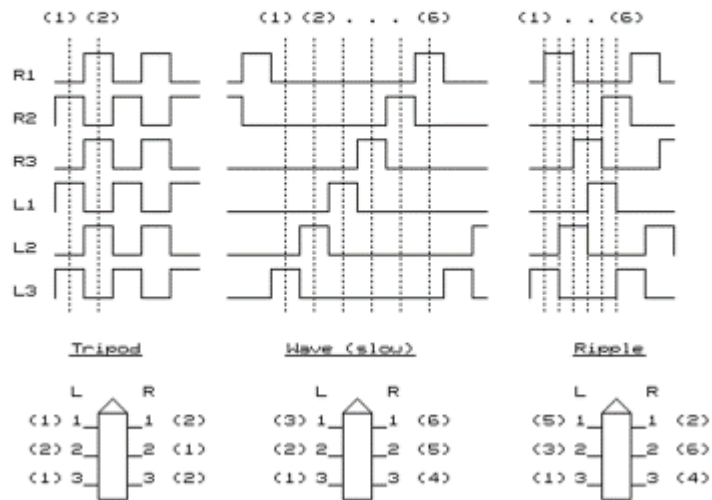


Рисунок Б.4 – Hexapod gait.

# (E)OgmaNeo layer

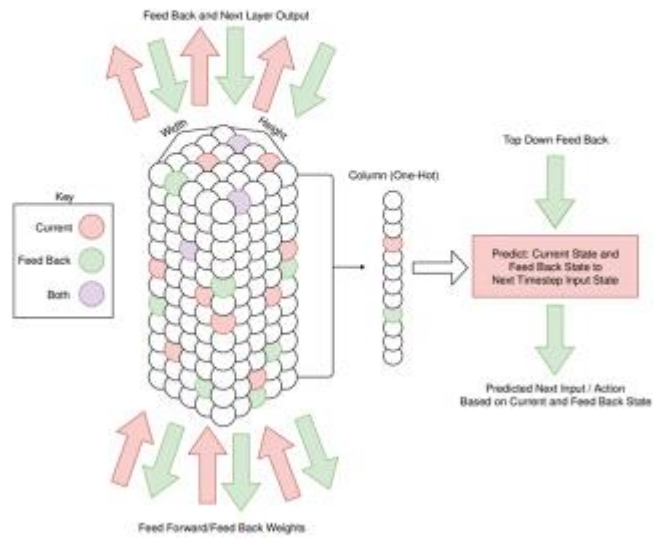


Рисунок Б.5 – (E)OgmaNeo layer.

# Xenomai architecture

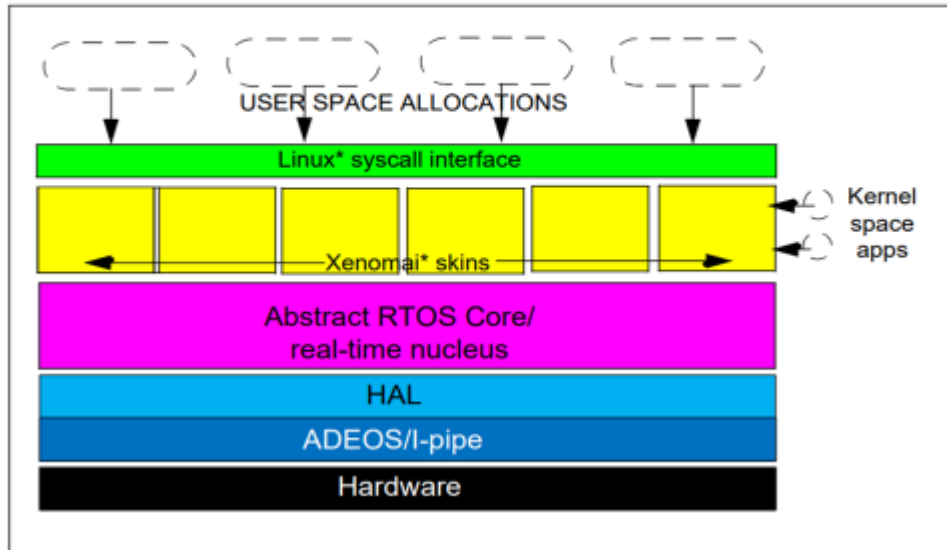


Рисунок Б.6 – Xenomai architecture.

# Software diagram

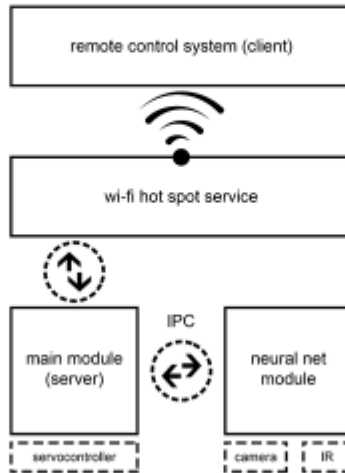


Рисунок Б.7 – Software diagram.

# Control application interface

CONNECT	DISCONNECT	SHUTDOWN	REBOOT	PREDICTIVE
voltage: ?				angle x: ?
cpu temp: ?				angle y: ?
proximity: ?				angle z: ?
UP		FORWARD		
TURN LEFT	TURN RIGHT	LEFT	RIGHT	
DOWN		BACKWARD		

Рисунок Б.8 – Control application interface.



## Economic costs

Electronic component / Module	Cost
Aluminum set	65.5\$
Servo MG995 (x3)	169.2\$
Servo Controller LSC-24	32.6\$
Single board computer RPi2 Model B	40.0\$
Camera Raspberry v2 (BMP)	35.0\$
IR Sharp GP2Y0A02YK0F	13.5\$
ADC ADS1115/ASD1015	3.2\$
Battery Li-Po 2S 4500mAh 30C	35.0\$
DC-DC (USB) buck converter 3A	3.5\$
Overhead, consumables, etc.	40.0\$
<b>Total cost: 435.5 \$</b>	

Рисунок Б.9 – Control application interface.

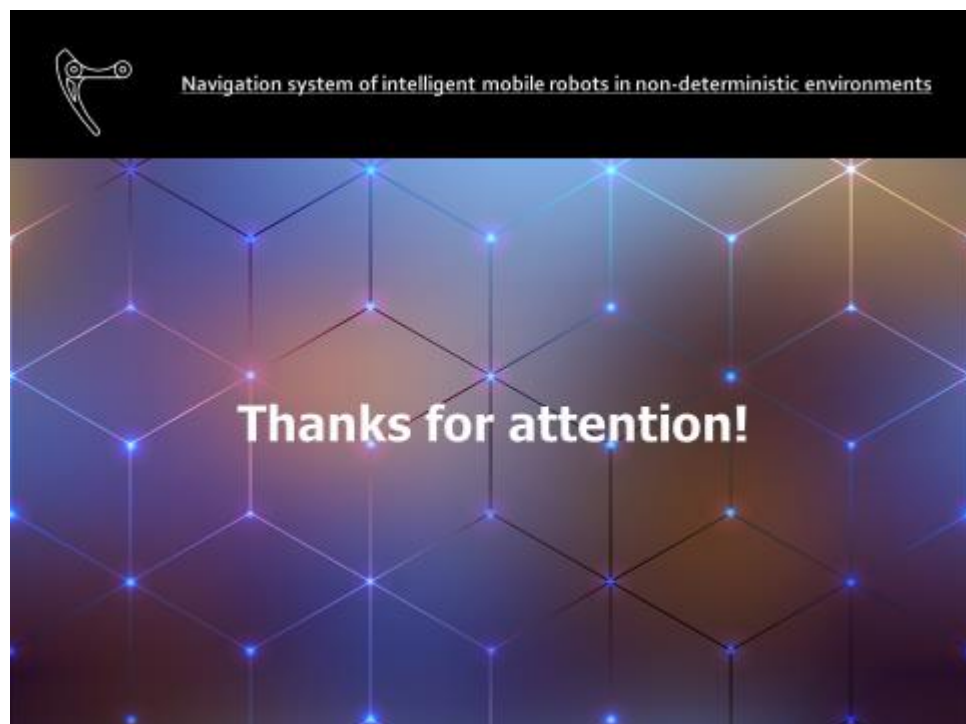


Рисунок Б.10 – Final slide.