Вінницький національний технічний університет
(повне найменування вищого навчального закладу)
Факультет інтелектуальних інформаційних технологій та автоматики
(повне найменування інституту, назва факультету (відділення))
Кафедра комп'ютерних наук
(повна назва кафедри (предметної, циклової комісії))

Бакалаврська дипломна робота на тему:

ПРОГРАМНИЙ МОДУЛЬ УПРАВЛІННЯ РОБОТОТЕХНІЧНИМ ПРИСТРОЄМ ПРИ ПОДОЛАННІ ПЕРЕШКОД

Виконав: студент 4 курсу, групи 2КН-20б
спеціальності 122 – Комп'ютерні науки
(шифр і назва напряму підготовки, спеціальності)
Фойда Д.О.
(прізвище та ініціали)

Керівник: к.т.н., доцент кафедри КН
Озеранський В.С.
(прізвище та ініціали)
«07» 06 2024 р.

Рецензент: к.т.н., доцент кафедри САІТ
Варчук І.В.
(прізвище та ініціали)
«07» 06 2024 р.

Допущено до захисту
Завідувач кафедри КН
д.т.н., проф. Яровий А.А.
(прізвище та ініціали)
«07» 06 2024 р.

Вінниця ВНТУ – 2024 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра Комп'ютерних наук
Рівень вищої освіти перший (бакалаврський)
Галузь знань – 12 «Інформаційні технології»
Спеціальність – 122 «Комп'ютерні науки»
Освітньо-професійна програма – «Комп'ютерні науки»

ЗАТВЕРДЖУЮ
Завідувач кафедри КН
проф., д.т.н. Яровий А.А.
"07" 06 2024 року

ЗАВДАННЯ
НА БАКАЛАВРСЬКУ ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Фойді Дмитру Олеговичу

1. Тема роботи: Програмний модуль управління робототехнічним пристроєм при подоланні перешкод.
керівник роботи: Озеранський Володимир Сергійович, к.т.н., доцент
затверджені наказом вищого навчального закладу "11" 03 2024 року №__

2. Строк подання студентом роботи 27.05.2024

3. Вихідні дані до роботи :
Кількість перешкод – не менше 5шт;
Кількість колісних пар – 1од;
Мова програмування – об'єктно-орієнтована;

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):
вступ, аналіз сучасних технологій управління робототехнічним пристроєм при подоланні перешкод, розробка алгоритму управління робототехнічним пристроєм при подоланні перешкод, висновки, перелік використаних джерел, додатки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень):
структура програмного модуля управління робототехнічним пристроєм при подоланні перешкод; схема алгоритму управління робототехнічним пристроєм при подоланні перешкод; загальна UML-діаграма класів; приклади роботи програми.

Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-3	Озеранський В.С., к.т.н., доцент		

6. Дата видачі завдання 07.03.2024

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва та зміст етапу	Термін виконання		Приміт
		початок	закінчення	
1	Обгрунтування доцільності розробки програмного модуля управління робототехнічним пристроєм при подоланні перешкод	06.05.2024	07.05.2024	
2	Розробка програмного модуля управління робототехнічним пристроєм при подоланні перешкод	08.05.2024	09.05.2024	
3	Програмна реалізація модуля управління робототехнічним пристроєм при подоланні перешкод	13.05.2024	14.05.2024	
4	Аналіз функціонування програмного модуля управління робототехнічним пристроєм при подоланні перешкод	19.05.2024	19.05.2024	
5	Оформлення додатків	23.05.2024	26.05.2024	
6	Розробка інструкції користувача	26.05.2024	27.05.2024	
7	Оформлення матеріалів до захисту БДР	04.06.2024	06.06.2024	

Студент _____ (підпис) Фойда Д.О. (прізвище та ініц

Керівник проекту (роботи) _____ (підпис) Озеранський (прізвище та ініц

АНОТАЦІЯ

УДК 621.374.415

Фойда Д.О. Програмний модуль управління робототехнічним пристроєм при подоланні перешкод. Бакалаврська дипломна робота складається з 67 сторінок формату А4, на яких є 31 рисунок, список використаних джерел містить 21 найменування.

В даній бакалаврській дипломній роботі досліджено процес розробки програмного модуля управління робототехнічним пристроєм при подоланні перешкод.

Проаналізовано основні види рухомих мобільних роботів та сучасні технології управління робототехнічними пристроями при подоланні перешкод. Створено узагальнену схему управління мобільним робототехнічним засобом, який оминає перешкоди, що трапляються на шляху його слідування. Обгрунтовано доцільність створення програмного модуля управління робототехнічним засобом при подоланні перешкод. Розроблено систему управління рухомим робототехнічним пристроєм при подоланні перешкод. Розроблено алгоритм управління робототехнічним пристроєм при подоланні перешкод. Здійснено моделювання роботи робототехнічного пристрою при подоланні перешкод в середовищі WeBots, яке довело правильність функціонування всіх його складових. Реалізовано робототехнічний пристрій, який оминає перешкоди, що трапляються на шляху у вигляді лабораторного макету.

Ключові слова: робот, моделювання, WeBots, Arduino, сонар, подолання перешкоди Arduino IDE.

ABSTRACT

Foyda D. Software module for controlling a robotic device when overcoming obstacles. The bachelor thesis consists of 67 pages of A4 format, on which there are 31 figures, the list of used sources contains 21 names.

In this bachelor's thesis, the process of developing a software module for controlling a robotic device when overcoming obstacles is investigated.

The main types of mobile mobile robots and modern control technologies of robotic devices when overcoming obstacles are analyzed. A generalized control scheme for a mobile robotic device has been created, which bypasses the obstacles that occur on the way of its tracking. The expediency of creating a software module for controlling a robotic device when overcoming obstacles is substantiated. A control system for a moving robotic device when overcoming obstacles has been developed. An algorithm for controlling a robotic device when overcoming obstacles has been developed. Simulation of the operation of the robotic device when overcoming obstacles in the WeBots environment was carried out, which proved the correct functioning of all its components. A robotic device that bypasses obstacles encountered on the way in the form of a laboratory model has been implemented.

Keywords: robot, simulation, WeBots, Arduino, sonar, obstacle course Arduino IDE.

ЗМІСТ

ВСТУП 7
1 ОБГРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ ПРОГРАМНОГО МОДУЛЯ УПРАВЛІННЯ РОБОТОТЕХНІЧНИМ ПРИСТРОЄМ ПРИ ПОДОЛАННІ ПЕРЕШКОД 9
1.1 Аналіз основних способів розробки мобільних роботів 9
1.2 Поняття мобільних роботів 11
1.3 Класифікація мобільних роботів за функціональним призначенням 14
1.4 Узагальнена структура рухомого мобільного робота 15
1.5 Висновок 16
2 РОЗРОБКА ПРОГРАМНОГО МОДУЛЯ УПРАВЛІННЯ РОБОТОТЕХНІЧНИМ ЗАСОБОМ ПРИ ПОДОЛАННІ ПЕРЕШКОД 17
2.1 Розробка системи управління рухомим роботом 17
2.2 Обгрунтування вимог до обладнання та вибір елементів рухомого робототехнічного пристрою 20
2.3 Розробка алгоритму управління робототехнічним пристроєм при подоланні перешкод 33
2.4 Висновок 35
3 ПРОГРАМНА РЕАЛІЗАЦІЯ МОДУЛЯ УПРАВЛІННЯ РОБОТОТЕХНІЧНИМ ПРИСТРОЄМ ПРИ ПОДОЛАННІ ПЕРЕШКОД 36
3.1 Обґрунтування вибору мови програмування та середовища розробки 36
3.2 Побудова структури робота при подоланні перешкоди 37
3.3 Моделювання робототехнічного пристрою при подоланні перешкод в середовищі Webots 38
3.4 Програмування робота при подоланні перешкод 44
3.5 Тестування розробленого програмного забезпечення управління робототехнічним пристроєм при подоланні перешкод 50
3.6 Висновок 51
ВИСНОВКИ 52

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ 53
ДОДАТОК А. Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень 56
ДОДАТОК Б. Інструкція користувача 57
ДОДАТОК В. Лістинг програми 58
ДОДАТОК Г. Графічна частина 62

ВСТУП

Актуальність досліджень. Характерною рисою сучасного етапу розвитку засобів обчислювальної техніки є безупинне збільшення функціональних можливостей і подальше ускладнення структури елементної бази ЕОМ. Постійно удосконалюється технологія виробництва виробів електронної техніки, розширюються її можливості. У результаті модернізації сучасних технологій підвищується ступінь інтеграції цифрових елементів ЕОМ, росте їхня номенклатура.

Робототехніка - це наука, яка стрімко розвивається. Навіть зараз неможливо уявити наше життя без роботів, вони використовуються там, де людині знаходитись дуже небезпечно, тому людство намагається в усіх сферах запровадити використання роботів.

Найпростішими є роботи, які оминають перешкоди. На основі них було створено безпілотні транспортні засоби. Наприклад, дрони або безпілотні машини. Все дедалі частіше можна побачити транспортні засоби, які не потребують водія. Також намагаються створити систему, яка буде уподібнена до людського мозку та приймати рішення схожі на рішення звичайних людей. Насправді роботизація досягла поширення в усіх сферах життя. Наприклад, у багатьох країнах скоротили кількість професій, використовуючи замість людей роботи, наприклад з кулінарії, прибиранні тощо. Звичайно ж роботи не працюють ідеально і їм потрібно багато чого навчитися. Тому людство створило роботів, які зможуть подальшому навчатись чомусь новому, що буде корисне людям, та повторювати побачене й удосконалювати свої власні навики.

Актуальність роботи полягає в тому, що роботи задіяні майже в усіх сферах життя людини. Основою автоматизації автотранспорту, авіатранспорту, військової справи, сучасної промисловості і в інших сферах діяльності людей є робототехнічні системи. Промислові роботи вже давно застосовуються в авіабудуванні, в суднобудуванні і в масі інших різних підприємств. У медичній практиці так само використовують робототехнічні системи, що дає можливість діагностувати різноманітні захворювання, а також проводити хірургічні операції. Не будемо

забувати про те що, існують не тільки промислові роботи, а й роботи широкого споживання, які використовуються в побуті, адже зараз активно ведуться розробки і випуск на світовий ринок таких роботів.

На сьогоднішній день автономні мобільні роботизовані системи є найбільш перспективним напрямком в робототехніці. Такі системи відрізняються від інших особливістю керування, вони повністю не залежать від оператора, ґрунтуються на аналізі показників датчиків і прийнятті рішень потрібних дій.

Незважаючи на те, що на сьогоднішній день проведений цілий ряд досліджень в галузі керування мобільними колісними роботами, універсальні підходи до синтезу систем автоматичного керування колісними роботами розроблені недостатньо.

Все це пояснює доцільність розробки програмного модуля управління робототехнічним пристроєм при подоланні перешкод.

Об'єкт дослідження – це процес управління робототехнічним пристроєм при подоланні перешкод.

Предмет дослідження – програмні та апаратні засоби управління робототехнічним пристроєм при подоланні перешкод.

Мета дослідження – підвищення якості управління робототехнічним пристроєм при подоланні перешкод.

Для досягнення поставленої мети необхідно вирішити такі задачі:

- проаналізувати сучасні технологій управління робототехнічним пристроєм при подоланні перешкод;

- розробити систему управління робототехнічним пристроєм при подоланні перешкод;

- здійснити моделювання робототехнічного пристрою при подоланні перешкод;

- розробити програмну та апаратну частини робототехнічного пристрою при подоланні перешкод;

- провести тестування нової розробки та проаналізувати отримані результати.

1 ОБГРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ ПРОГРАМНОГО МОДУЛЯ УПРАВЛІННЯ РОБОТОТЕХНІЧНИМ ПРИСТРОЄМ ПРИ ПОДОЛАННІ ПЕРЕШКОД

1.1 Аналіз основних способів розробки мобільних роботів

Автономні мобільні роботи в буденному житті як, правило мають набір давачів для аналізу навколишнього середовища, акумулятором і станцією для його заряджання також уособлюють в собі простоту дизайну, будучи складним механізмом, який здатні самостійно приймати рішення у певних поставлених для них задач. Чимале розповсюдження здобули промислові роботи, які стали головною технологічною базою машинобудівної, приладобудівної та електронних галузей світової промисловості. Зараз існує велика кількість працюючих промислових роботів, які виробляють промислові роботи для маніпулювання, зварювання, фарбування, упакування, шліфування, полірування і т.д. з великим спектром застосування й по точності, і по характері виконуваних операцій. Актуальність полягає в тому, що вони все більше використовуються в різних технологічних процесах на виробництві у всьому світі, для звільнення людини від виконання нетворчої, механічної чи небезпечної роботи.

Роботи - це автоматизовані машини, які здатні виконувати функції людини при взаємодії з навколишнім світом. Про них люди мріяли ще з давніх-давен, і ось зараз ці механізми входять у наше суспільство з величезною швидкістю. Основне їх призначення – зробити наше життя комфортнішим, покращити умови праці, звільнити «руки» від складних робочих процесів та збільшити продуктивність.

Роботи найчастіше зустрічаються у промисловості, де з їхньою допомогою вдалося повністю автоматизувати більшість виробничих завдань. Але, крім того, розумні машини все більше задіяні у військовій галузі, медицині, сфері обслуговування та споживчому секторі.

Найбільшою галуззю, де використовується робототехніка, є промисловість, зокрема автомобілебудування. Маніпулятори, що працюють на заводах, варіюються

від розмірів та функціональності залежно від типу виконуючої задачі – складальні, зварювальні, ріжучі, барвники. Поряд з ними на виробництві можна зустріти розвантажувально-навантажувальних роботів, пакувальників, сортувальників, формувальників та інші механізми, що замінюють людину в рутинних завданнях, що повторюються. Компаніями-лідерами в промисловій автоматизації є KUKA (Німеччина), Fanuc (Японія), Kawasaki (Японія), ABB (Швейцарія), Denso (Японія) та інші.

Роботи також активно використовуються у сільському господарстві. Найчастіше, це радіокеровані трактори та плуги, але дедалі ширшого застосування набувають безпілотних літальних апаратів, які аграрії використовують для картографування своїх угідь та регулярного огляду культур.

А які роботи служать у побуті? Безумовно, перше місце тут належить роботам-пилососам, які стали незамінними помічниками прибирання в будинку. Лідером серед виробників цих пристроїв є американська фірма iRobot та її пилососи Roomba. Останні моделі виробника відрізняються покращеною навігацією та поєднанням зі смартфоном. Це доповнення відкриває нові можливості для звичайних користувачів, які можуть через спеціальні програми додавати роботам більше функцій.

На рисунку 1.1 зображено класифікацію та порівняння різних видів роботів за способом їх використання і призначенням.

Рисунок 1.1 – Класифікація роботів

Роботи поділяються на такі категорії:

- Промислові роботи - призначені для автоматизації різноманітних технологічних операцій (наприклад зварювання, штампування, металообробка, складання готових виробів тощо);
- Будівельні роботи - дозволяють аналогічно автоматизувати величезну кількість різних операцій, що виконуються в процесі ремонту приміщень або будівництва нових об'єктів;
- Сільськогосподарські роботи - призначені для виконання трудомістких та монотонних процесів у сільському господарстві;
- Побутові роботи - застосовується в побуті та офісах. Яскравим прикладом побутової автоматизованої машини є робот-пилосос, що набрав пристойну популярність;
- Бойові (військові) роботи - покликані вивести збройні конфлікти на якісно інший рівень і призначені для мінімізації безпосередньої участі людини у бойових діях з метою скорочення або виключення зовсім людських втрат, а також для роботи в умовах, несумісних із можливостями людини у військових цілях;
- Перед охоронними роботами ставляться завдання захисту довірених територій або приміщень;
- Дослідницькі роботи використовуються для збору різноманітних видів інформації про досліджувані об'єкти, її переробки та передачу оператору.

1.2 Поняття мобільних роботів

Робот, як термін, можна розуміти як багатофункціональний автомат для виконання механічної роботи, схожу на ту, що виконує людина. Думка про створення робота з'явилася після прагнення замінити людину на робота в важких і небезпечних для життя ситуаціях. З плином часу поняття робот стало ширше, під ним стали представляти будь-яку автоматичну машину, яка може замінити людину, і яка буде нагадувати його розумну поведінку.

Мобільний робот вміє пересуватися в робочому просторі відповідно до програми керування. Такі роботи необхідно програмувати заздалегідь. Повинна бути присутня здатність самостійно орієнтуватися в навколишньому середовищі і здійснювати виконання завдання, спираючись тільки на власний штучний інтелект. Тому ці роботи і мають назву – мобільні, тому що вони не прив'язані до оператора.

Мабуть, найпоширеніші сьогодні і корисні кожній домогосподарці роботи - це пилососи. Інфрачервоні датчики стежать, як за типом поверхні (паркет або килим), щоб вибрати відповідну щітку або режим роботи, так і за тим, щоб пилосос не скотився вниз сходами. Можна придбати робот-пилосос ROBOKING від LG (рис.1.2).

Рисунок 1.2 – Робот пилосос ROBOKING

Свій варіант пропонує і Panasonic (рис 1.3). Або, скажімо, Trilobit від Electrolux (рис. 1.4).

Рисунок 1.3 – Робот пилосос Panasonic

Рисунок 1.4 – Робот пилосос Trilobit

Робот-пилосос орієнтується в просторі так само, як кажани, - за допомогою ультразвуку: покритий тонкою золотою пластиною акустичний локатор з великою кількістю мікрофонів відчуває вібрації на частоті 60кГц. А напівкругла форма дає роботу кут огляду 180^0. Таким чином, Trilobite швидко визначає стіни, ніжки стільців, дверні отвори та сходи. Він легко долає проводи, кабелі і границю "підлога - килим". Зовні Trilobite найбільше схожий на круглу диванну подушку, а його невеликі розміри (висота 13 см і діаметр 35 см) дозволяють йому з легкістю прибирати навіть під ліжками та столами. Робот рухається на двох колесах, кожне з яких має свій двигун, а у Trilobite — чотири двигуни. Максимальна швидкість збирання - 40 квадратних сантиметрів за секунду. При тому, що людина з пилососом насправді прибирає пил лише з 60% доступної площі, Trilobite охоплює 95%. Під час збирання в електронних "мізках" робота-пилососа формується певна "карта місцевості". Крім того, робот обчислює, скільки часу потрібно для прибирання. Коли акумулятори робота "сідають", Trilobite сам знаходить зарядний пристрій (charging station) і їде заряджатися. Якщо батареї "сіли" раніше, ніж робот закінчив чистку, Trilobite - як тільки зарядиться - автоматично відновлює прибирання з того самого місця, де він її припинив. На жаль, робот не може сам вивантажити зібраний пил і, хоча мішки для збору пилу не потрібні - є "коробка багаторазового використання для пилу" (reusable dust box) - її господарям доведеться витрушувати самим. Керувати роботом-пилососом потрібно за допомогою LCD-дисплея - можна вибрати одну з трьох програм прибирання: звичайне, швидке та "місцеве", тобто на площі до двох квадратних метрів.

1.3 Класифікація мобільних роботів за функціональним призначенням

Розвиток технології штучного інтелекту передбачає широке використання роботів у різних галузях промисловості, інфраструктури та життєдіяльності людини. Прийняття рішень у виборі або розробці тієї чи іншої робототехнічної системи базується на аналізі та використанні системного підходу. Першим етапом є систематизація існуючих рішень, виділення значущих ознак та визначення переваг та недоліків таких рішень.

Відповідно до Згуровського М. З. – сьогодні відбувається інформаційна революція: перехід суспільства від індустріального до інформаційного, а надалі очікується його перетворення на суспільство знань. За різним оцінкам, 30–60 % робочих місць замінять робототехнічними системами до 2035 року.

У зв'язку з цим, необхідно вивчати та розвивати напрями, пов'язані з робототехнікою, зокрема – мобільними робототехнічними системами. Для вибору того чи іншого робота необхідно провести аналіз та класифікацію існуючих рішень, а також виявити їх переваги та недоліки (табл. 1.1).

Таблиця 1.1 – Класифікація мобільних роботів

Група	Базові вимоги	Базовий варіант, використання
Спеціального призначення	Компактність, безсумність	Автономні, один канал передачі відеоданих
Для військових та воєнізованих застосувань	Надійність, простота в управлінні, стандартизоване корисне навантаження	У складі розвідувальних, ударних і охоронних комплексів, багатоканальні системи передачі різноманітних даних
Для екстремальних ситуацій, наукових досліджень, кінематографічних застосувань	Стійкість до несприятливих зовнішніх впливів, універсальність по відношенню до корисного бортового навантаження	Автономні, багатоваріантність реалізації каналів даних
Для спортивних, промислових і побутових застосувань	Простота в управлінні, економічність, надійність	Автономні, один канал передачі відеоданих

1.4 Узагальнена структура рухомого мобільного робота

По конструкції багатофункціональні роботи представляють собою самохідні засоби, обладнані апаратурою розвідки, набором робочого обладнання, який можна замінити і інструментом. Вони мають на увазі наявність дистанційного керування і можливість вести відеоспостереження.

Рух мобільного робота в приміщенні, план якого заздалегідь невідомий має свої особливості. У приміщенні є як статичні перешкоди (стіни, столи, стільці), так і рухливі (люди, інші роботи). Мобільний робот оснащений скануючим лазерним далекоміром, який отримує стан рельєфу навколишніх об'єктів в площині, паралельній поверхні. Необхідно в режимі реального часу визначати положення мобільного робота в системі координат, пов'язаної з приміщенням (завдання локалізації), а також побудувати карту даного приміщення, що відображає рельєф стін і нерухомих об'єктів. Такого роду системи управління відомі як системи SLAM (System of Localization and Mapping). Функціональна структура навігаційної системи мобільного робота показана на рис. 1.5.

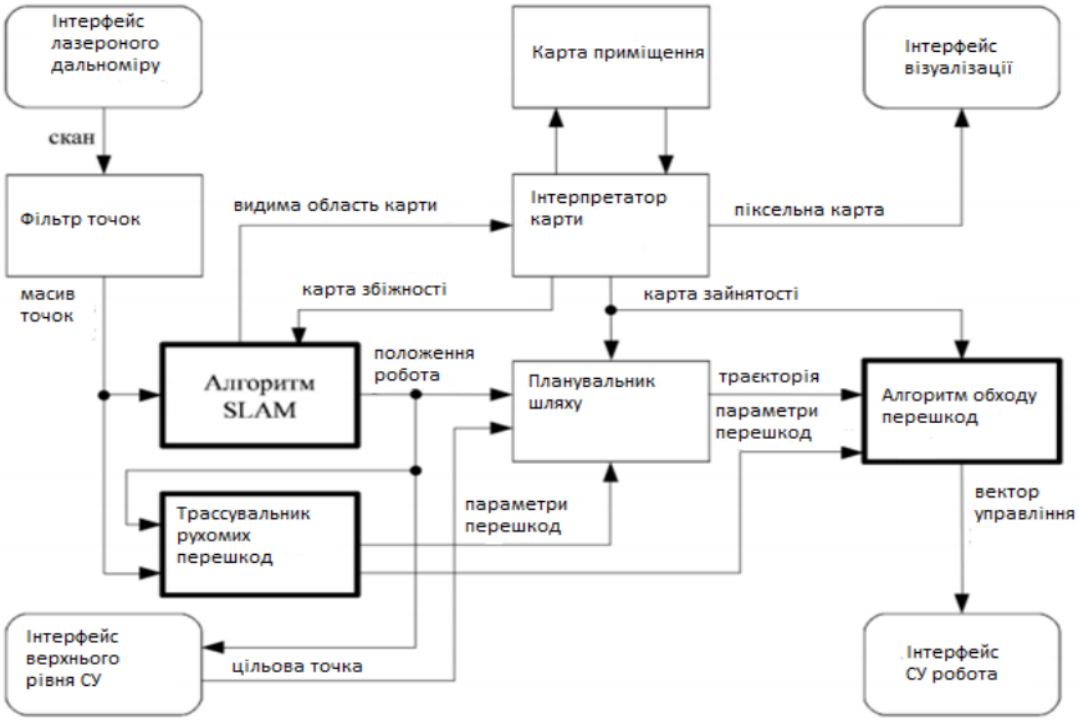


Рисунок 1.5 – Функціональна схема навігаційної системи мобільного робота

Особливістю запропонованої структури є незалежність від типу шасі мобільного робота, а також від наявності та типу датчиків одометра, що дозволяє використовувати розроблювану навігаційну систему на всіх типах мобільних роботів, які працюють в приміщенні.

1.5 Висновок

В даному розділі проаналізовано основні види рухомих мобільних роботів. Наведено класифікацію мобільних роботів за функціональним призначенням. Створено узагальнену схему управління мобільним робототехнічним засобом, який оминає перешкоди, що трапляються на шляху його слідування. Обгрунтовано доцільність створення програмного модуля управління робототехнічним засобом при подоланні перешкод.

2 РОЗРОБКА ПРОГРАМНОГО МОДУЛЯ УПРАВЛІННЯ РОБОТОТЕХНІЧНИМ ЗАСОБОМ ПРИ ПОДОЛАННІ ПЕРЕШКОД

2.1 Розробка системи управління рухомим роботом

Система управління дає можливість керувати рухом всього обладнання поміщеного в корпус робота. Але не варто забувати і про адаптивне керування ходової частини і про енергетичну установку з урахуванням взаємодії транспортної системи з оточенням.

Система управління включає в себе інформативно – керувальна частину. До неї входить: апаратура керування роботом, різні датчики і мікропроцесори; пост оператора мобільного робота (ПУ – пульт керування, персональний комп'ютер для можливості обробляти інформацію), а також приймально–передавальної апаратури, яка дає можливість передавати інформацію та команди керування.

У загальному випадку системою управління таких роботів виконуються наступні функції:

- за допомогою датчиків сприймає і розпізнає інформацію, яка надходить з навколишнього середовища;
- обмін інформацією з оператором;
- планування і перепланування дій;
- керування приводами;
- обмін інформацією з іншими роботами.

Отже, розглянемо ці особливості щодо мобільних роботів, які довільно переміщаються в будь–якій місцевості. В даному випадку модель навколишнього середовища буде представлена у вигляді карти місцевості, яка будується і уточнюється в процесі переміщення робота на основі сенсорної інформації. Така модель повинна мати два рівня за масштабом. Перший рівень призначений для меж досяжності сенсорних систем. Другий рівень призначений для визначення місцевості перед самим роботом.

Блок керування переміщенням реалізує керування тяговими приводами і приводами повороту шасі. Блок моделювання маршруту синтезує траєкторію руху, оптимізуючи її, по мінімуму втрати енергії, що вкрай важливо для роботів з автономним живленням, або по мінімуму часу завдання, коли в постановку задачі входить умова забезпечення максимальної швидкодії.

Блок моделювання плану місцевості показує її в такій формі, щоб було зручно реалізувати завдання вибору маршруту, з урахуванням перешкод і нерозпізнаних ділянок .

Якщо реалізація завдання вимагає знання більшої ділянки місцевості, ніж дають сенсорні системи в базовому положенні робота, карта місцевості формується і передається в блок побудови маршруту фрагментами в міру пересування робота. Такими ж ділянками послідовно реалізується і синтез траєкторії переміщення. В такому випадку перше наближення для всієї траєкторії робиться по апріорної інформації про місцевість або, як мінімум, визначається загальний напрямок переміщення.

На нижньому загальносистемному рівні системи керування є канал керування переміщенням в ближній зоні, що включає в себе найбільш детальну модель цієї зони і виконує алгоритм забезпечення безпеки переміщення .

Для виконання своїх завдань робот повинен рухатися по заданому маршруту і при цьому дотримуватися заходів безпеки, в тому числі, при наявності рухомих об'єктів в робочій зоні. Таким чином, робот переміщується автономно за допомогою навігаційної системи, при цьому оператор виконує тільки функцію постановки завдання. Не виключається і напівавтоматичний режим, наприклад, режим телеприсутності, при якому завдання оператора істотно спрощується. На першому етапі роботи навігаційної системи вирішується завдання фільтрації стану шляхом видалення помилкових вимірювань за допомогою спеціального фільтра. Для вирішення завдання аналізу моделі робочого середовища спочатку досліджувався метод нормальних розподілів (NDT – Normal Distribution 25 Transform). В цьому випадку карта приміщення розбивається на осередки, кожна з яких містить не самі точки, а параметри нормального розподілу всіх точок, що потрапили всередину.

Вирішуючи задачу мінімізації функції взаємної кореляції стану і карти, можна визначити положення робота, з якого був отриманий поточний стан.

Аналіз цього способу показав, що він істотно обмежує швидкість руху мобільного робота, оскільки всі обчислення виконуються в реальному масштабі часу. Тому було запропоновано новий спосіб, при якому отриманий за допомогою системи інформаційних датчиків скан перетворюється в сітковий функцію. При цьому кожна точка скану перетворюється в деяку безперервну функцію, потім вони об'єднуються за допомогою обраного принципу суперпозиції і накладаються на карту–сітку, тим самим утворюючи сітковий функцію. Метод сіткових функцій, так само як і метод нормальних 26 розподілів, заснований на зіставленні скану і отриманої карти за допомогою взаємної кореляційної функції. Для мінімізації цієї функції був використаний модифікований метод Ньютона. Після виконання завдання SLAM положення робота на мапі може бути обчислено з використанням перетворення з системи координат лазерного далекоміра в систему координат мобільного робота. Порівняльний аналіз двох розглянутих методів локалізації показав, що перевага нового методу сіткових функцій полягає в розширеній області збіжності, що дозволяє помітно збільшити швидкість руху мобільного робота. Специфіка управління мобільним роботом в динамічному середовищі полягає в тому, що рух перешкод можна розрахувати заздалегідь. Щоб уникнути зіткнень з рухомими перешкодами, необхідно знати їх положення і передбачати траєкторію їх руху. Тоді можна рухатися уздовж спланованою траєкторії, відхиляючись від неї в потрібний момент, щоб зробити маневр і об'їхати перешкоду. Пропонується алгоритм управління роботом в динамічному середовищі, заснований на трасуванні рухомих перешкод . На першому етапі вирішується завдання планування маршруту по побудованої методом сіткових функцій мапі приміщення. Для цього використовується добре відомий алгоритм A* . Далі вирішується задача трасування рухомих перешкод – визначення поточного вектора стану перешкоди в кожен момент часу, синхронізований з отриманням нового скана. Для побудови списку перешкод, спочатку проводиться класифікація і кластеризація точок скану. Кластеризація в даному випадку проводиться по Евклідовому віддалі між точками скану. Граничне значення обчислюється виходячи

з відстані до точки і кутового дозволу лазерного далекоміра. Кожен рухливий об'єкт обробляється з метою отримання відповідних йому параметрів габаритної окружності. Після знаходження габаритної окружності можна отримати оцінку повного вектора стану рухомого об'єкта (рис. 2.1).

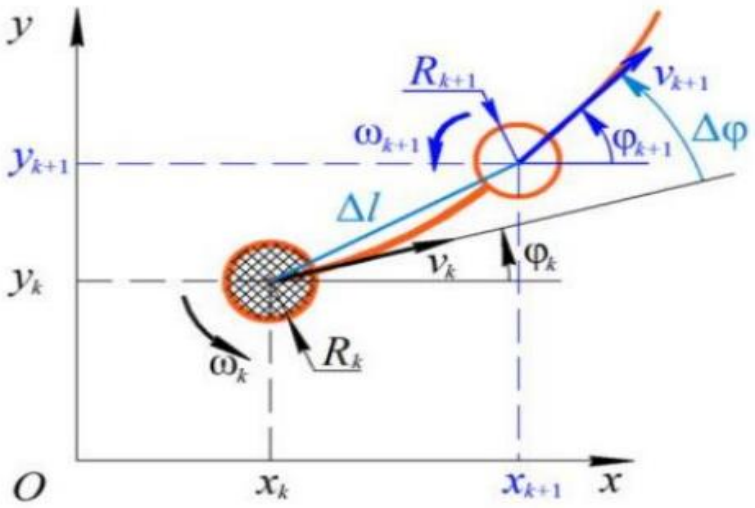


Рисунок 2.1 – Визначення положення рухомого об'єкту

2.2 Обгрунтування вимог до обладнання та вибір елементів рухомого робототехнічного пристрою

Для управління роботом при подоланні перешкод потрібен великий діапазон регулювання швидкості, висока точність підтримки швидкості обертання приводу. Звідси випливає, що необхідним є застосування двигуна постійного струму (ДПС) з редуктором (табл. 2.1).

Двигуном постійного струму (ДПС) називається електрична машина постійного струму, що перетворює електричну енергію постійного струму в механічну енергію. Існують різні типи двигунів постійного струму: незалежного збудження, паралельного збудження, послідовного збудження і змішаного збудження [4].

Основними параметрами систем двигуна постійного струму є діапазон регулювання швидкості, точність підтримки швидкості обертання приводу.

Таблиця 2.1 – Переваги та недоліки двигунів постійного струму

Переваги	Недоліки
1. Широкий і безступінчатий діапазон регулювання частоти обертання; 2. Простота регулювання частоти обертання; 3. Жорсткі механічні характеристики, що задовольняють вимогам до приводів верстатів, зокрема, у двигунів з незалежним, паралельним збудженням; 4. Велику перевантажну здатність з коефіцієнтом перевантаження в межах Кп = Мп / Мн = 2,5 ... 10; де Мп – пусковий момент, Мн – номінальний момент. 5. Порівняно мала інерційність (достатню швидкодію) [3].	1.Складність конструкції; 2. При їх застосуванні потрібен перетворювач змінного струму в постійний.

Для розроблювального робототехнічного пристрою необхідно застосування приводів, які відповідають таким вимогам:

- швидкість обробки заданого положення;

- надійність;

- мала маса і малі габарити приводу;

- наявність стандартного посадкового місця для зручності розміщення і монтажу;

- висока енергоефективність;

- доступність.

Для вибору необхідного двигуна проведемо наступні розрахунки:

Припустимо, що вага робота буде – 2 кг.

1. Приймаємо швидкість руху робота – 45 см / сек.

2. Розрахунок швидкості обертання ведучого колеса. Здійснюється за формулою:

$$nk = \frac{V}{(3.14 \cdot D)},$$

де V – швидкість руху робота (см / сек), а D – діаметр ведучого колеса (см).

Діаметр обраних ведучих коліс – 4,5 см, відповідно, їх швидкість обертання складе

$$nk = \frac{V}{(3.14 \cdot D)} = \frac{45}{(3.14 \cdot 4.5)} = 3.2 \text{ об/сек}$$

Необхідний двигун повинен мати вбудований понижаючий редуктор. Швидкість обертання валу редуктора при номінальному навантаженні – не менш 200 об / хв (табл. 2.2).

Таблиця 2.2 – Характеристики двигунів доступних у продажу

GM37–520		Напруга 12 В; Максимальна швидкість 320 об./хв; Швидкість при номінальному навантаженні 200 об/хв; Струм при номінальному навантаженні 150 мА; Передаточне число редуктора 1:30; Вага 0,152 кг; Розмір 60мм; Ціна 1 512,75 руб.
GM25–370		Напруга 9 В; Максимальна швидкість 150 об./хв; Швидкість при номінальному навантаженні 100 об/хв; Струм при номінальному навантаженні 1200 мА; Передаточне число редуктора 1:10; Вага 0,1 кг; Розмір 25мм; Ціна 460 руб.
GA12–N20		Напруга 6 В; Максимальна швидкість 450 об./хв; Швидкість при номінальному навантаженні 300 об / хв; Струм при номінальному навантаженні 170 мА; Передаточне число редуктора 1:30; Вага 0,012 кг; Розмір 24мм; Ціна 550 руб.

З вище перерахованих двигунів для робота були обрані двигуни постійного струму типу GA12–N20 (рис. 2.2).

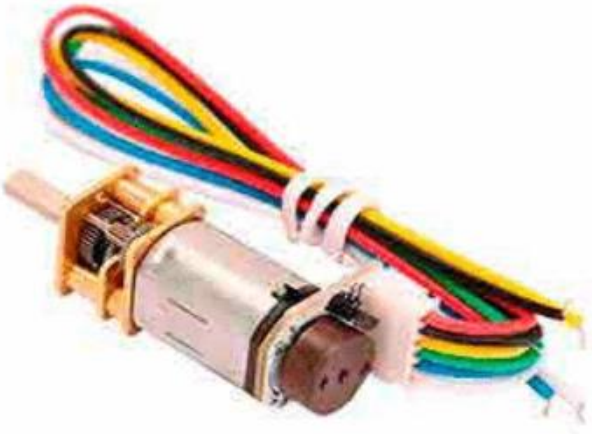


Рисунок 2.2 – Двигун постійного струму типу GA12–N20

Даний тип електродвигунів був обраний, тому що на відміну від вище перерахованих двигунів вони досить компактних розмірів (довжина – 25 мм, ширина – 12 мм), мають гранично малу вагу (0, 012 кг). Є в наявності енкодер. Передавальне число таких двигунів 1:30, 450 об / хв, 210 імпульсів на оборот. При виборі значну роль зіграла економічна частина, тому що ціна таких моторів значно нижче своїх аналогів. А також, характеристики даного двигуна близькі до значень розрахунків. В роботі будуть використані двигуни в кількості двох штук.

Для реалізації систем керування електроустаткуванням, в т.ч. і електроприводами, застосовують обчислювальні пристрої, зокрема мікроконтролери.

Мікроконтролер (англ. Micro Controller Unit, MCU) – мікросхема, призначена для керування електронними пристроями. Типовий мікроконтролер поєднує на одному кристалі функції процесора і периферійних пристроїв, містить ОЗУ і (або) ПЗУ. По суті, це однокристальний комп'ютер, здатний виконувати досить прості завдання. Відрізняється від мікропроцесора інтегрованими в мікросхему пристроями введення – виведення, таймерами і іншими периферійними пристроями.

Використання в сучасному мікроконтролері достатнього потужного обчислювального пристрою з широкими можливостями, побудованого на одній мікросхемі замість цілого набору, значно знижує розміри, енергоспоживання і вартість побудованих на його базі пристроїв.

Програмування мікроконтролерів зазвичай здійснюється на мові асемблера або C++, хоча існують компілятори для інших мов.

У промисловості набули поширення різні мікроконтролери, це AVR і SAM3 фірми Atmel, STM8 і STM32 фірми STMicroelectronics, PIC фірми Microchip, Arduino.

Для виконання системи керування робота, необхідний мікроконтролер, якій відповідають наступним вимогам:

- наявність не менше 8 КБ пам'яті програм;

- наявність незалежної пам'яті EEPROM для запису і зберігання робочих даних;

- наявність не менше 5 апаратних таймерів з можливістю формування ШІМ сигналу;

- наявність апаратного інтерфейсу для зв'язку з ПК;

- наявність не менше 15 GPIO;

- низьке енергоспоживання;

- забезпечення вихідного струму через порт не менше 20 мА, для керування силовими ключами і пристроями індикації без застосування узгоджувальних каскадів;

- підтримка безкоштовними IDE;

- можливість програмування, як на мовах низького рівня, так і на мовах високого рівня;

- можливість швидкого виконання математичних операцій, таких як множення, ділення, а також тригонометричних функцій.

Arduino Nano – це повнофункціональний мініатюрний пристрій на базі мікроконтролера ATmega328 (Arduino Nano 3.0) або ATmega168 (Arduino Nano 2.x), адаптований для використання з макетної плати. За функціональністю пристрій схожий на Arduino Duemilanove, і відрізняється від нього розмірами, відсутністю роз'єму живлення, а також іншим типом (Mini-B) USB-кабелю. Arduino Nano (рис. 2.3) розроблено і випускається фірмою Gravitech.

Рисунок 2.3 – Зовнішній вигляд контролера Arduino Nano

Головна відмінність цієї мініатюрної плати полягає у відсутності гнізда для зовнішнього джерела живлення, натомість використовуються VIN. Коли йдеться про створення мініатюрного пристрою, розмір Arduino Nano v3 ATmega328 / ATmega168 відіграє вирішальну роль при виборі платформи.

Технічні характеристики контролера Arduino Nano (рис. 2.4):

- мікроконтролер: ATmega328;
- тактова частота: 16 МГц;
- напруга логічних рівнів: 5 В;
- вхідна напруга живлення: 7-12 В;
- портів введення-виведення загального призначення: 22;
- максимальний струм з піна введення-виведення: 40 мА;
- максимальний вихідний струм піна 3.3V: 50 мА;
- максимальний вихідний струм піна 5V: 800 мА;
- портів з підтримкою ШІМ: 6;
- портів, підключених до АЦП: 8;
- розрядність АЦП: 10 біт;
- Flash-пам'ять: 32 КБ;
- EEPROM-пам'ять: 1 КБ;
- SRAM-пам'ять: 2 КБ;
- габарити: 18×45 мм.

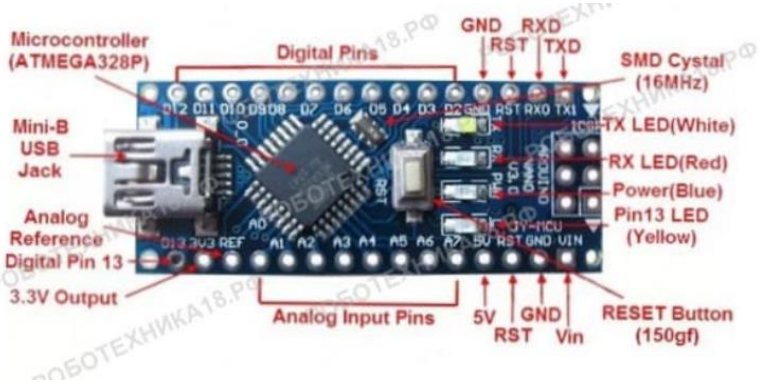


Рисунок 2.4 – Структура контролера Arduino Nano

Драйвер двигунів MX1508 – це аналог драйвера двигунів L298N. Пристрій створений на основі мікросхеми MX1508, а принцип роботи драйвера досить простий: всередині пристрою знаходиться два транзистора, а при зміні полярності напруги відбувається зміна напрямку обертання двигунів. За допомогою такого драйвера можна створювати мобільних роботів, автономні автомобілі на основі мікроконтролерів, а також інші роботизовані пристрої з механічними модулями.

Модуль двигунів MX1508 (рис 2.5) має стандартний інтерфейс, за допомогою якого легко та швидко можна підключитися до пристроїв, створених на основі Arduino або інших мікроконтролерів. Живлення модуля здійснюється за допомогою керуючого пристрою або від зовнішньої батареї. Дана плата також дозволяє можливість ШІМ-управління для плавної зміни швидкості обертання двигуна. MX1508 може управляти двома двигунами постійного струму або 4-дротяними двофазними крокovими двигунами, в тому числі дозволяє регулювати швидкість обертання чи міняти напрям обертання, здійснювати реверс.

Характеристики:

- мікросхема: MX1508;
- напруга живлення модуля: 2-10 В;
- напруга вхідного сигналу: 1.8-7 В;
- робочий струм: 1.5 А, піковий до 2.5 А;
- розміри: 25x21x6 мм;
- вага: 3 г.

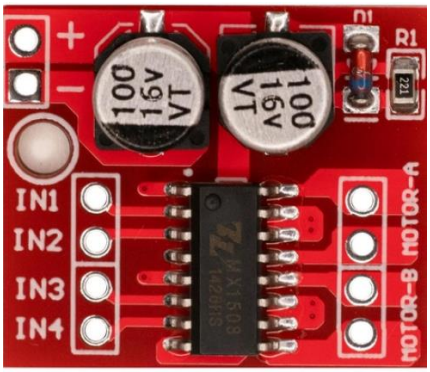


Рисунок 2.5 – Драйвер двигунів MX1508

Драйвер двигуна L298N для Arduino (рис. 2.6) використовується для управління двома малопотужними колекторними двигунами постійного струму або малопотужним 4-х дротяним двофазним кроковим двигуном. Практичне застосування: управління двигунами невеликих колісних роботів або двигунами пересувних іграшок.

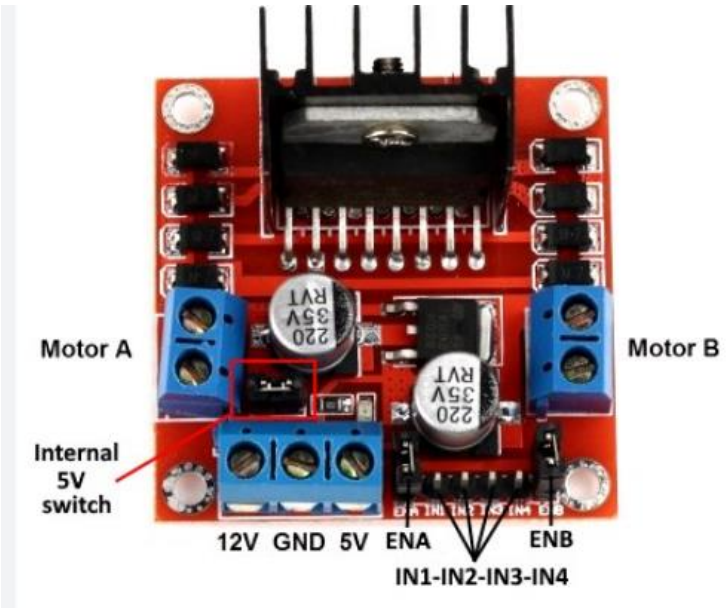


Рисунок 2.6 – Драйвер двигунів L298

Для закріплення модуля на плоскій поверхні в платі передбачено один монтажний отвір.

Управління драйвером L298N здійснюється або від контролера Arduino, або від іншого мікропроцесорного керуючого пристрою. Мікросхема модуля L298N працює за принципом Н-моста і використовуються для зміни полярності живлення мотора, що дає можливість реверсу.

Живлення драйвера здійснюється або від Arduino контролера, або іншого мікропроцесорного керуючого пристрою, або зовнішнього джерела живлення (блоку живлення, батареї). Напруга живлення становить 2 - 9 В постійного струму.

Характеристики:

- мікросхема: L298N;
- особливість: Н-міст, можливість реверсу;
- напруга живлення: 2 - 9 В постійного струму;
- керуючий сигнал: 1,8 - 7 В постійного струму;
- максимальний споживаний струм двигунів: до 1,5 А;
- розміри: 25 х 21 х 6 мм;
- вага: 3 г.

Оптичний енкодер H206 (рис. 2.7) дуже зручний в підключенні до Arduino, тому що має на борту всю розв'язку, що дозволяє живити від 3,3 до 5 вольт. Завдяки наявності у схемі тригера Шмітта, вихідний сигнал повністю «цифровий». Також є світлодіод для індикації розмикання лінії. Він буде корисний при налаштуванні взаємного положення енкодера та крильчатки осі двигуна.

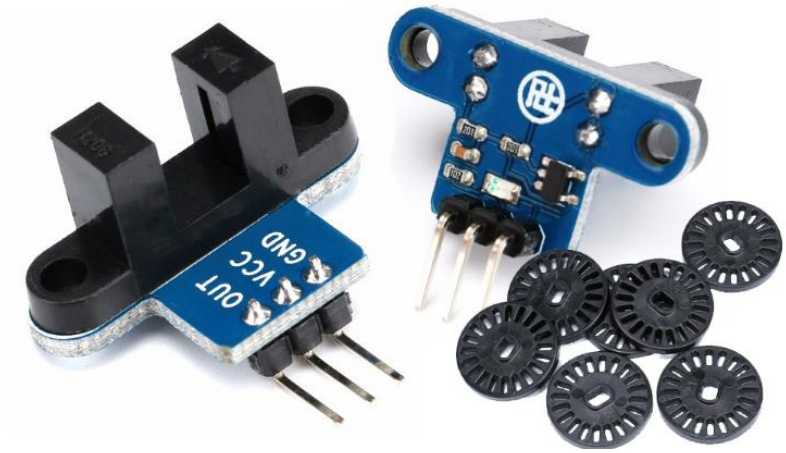


Рисунок 2.7 – Оптичний енкодер H206

Кнопки для Arduino (рис. 2.8) складаються з невеликої текстолітової плати, на якій розпаяні: тактова кнопка 12*12 мм, притягуючий резистор і 3 піна для підключення до Arduino. Також в комплект входять ковпачки для кнопок різних кольорів. Всього в комплекті 5 модулів кнопок та 5 різнокольорових ковпачків.

Рисунок 2.8 – Кнопки для Arduino

Серед усіх ультразвукових датчиків, які сумісні з Arduino Nano, HC-SR04 (рис. 2.9) ідеально підходить для розробки робототехнічного пристрою при подоланні перешкод, тому що:

– вартість датчика дуже низька;
– висока точність вимірювань;
– діапазон вимірювань від 2 см до 400 см;
– електромагнітні шуми не впливають на точність вимірювань;
– має власну бібліотеку для мови С під назвою NewPing.

Рисунок 2.9 – Зовнішній вигляд ультразвукового датчика HC-SR04

Принцип роботи заснований на ехолокації. Передавач складається з випромінювача сигналу, що генерує ультразвукові хвилі, приймача, що виявляє відбиті сигнали «ехо», і додаткових електронних елементів для коректної роботи модуля. Даний ультразвуковий датчик генерує хвилі з частотою 40 кГц. Спрощено принцип роботи цього датчика представлено на рис. 2.10.

Рисунок 2.10 – Принцип роботи ультразвукового датчика HC-SR04

Після того, як ці хвилі відбиваються від об'єкта і повертаються до приймача, перетворювач дасть інформацію про загальний час, за який звук пройшов від перетворювача до об’єкта і назад. За даним принципом працюють більшість датчиків цього типу.

Недоліком такого датчика є збільшення похибки вимірювання до ±3 см при виявленні перешкод на відстані більше 4м. А також при несприятливих погодних умовах, таких як висока вологість або низька температура, виникають похибки вимірюваних показників. Цю проблему можна вирішити за допомогою модернізованого аналога з вбудованою температурною корекцією значень, наприклад, недорогого датчика US-100 [7].

US-100 – далекомір, який теж працює за принципом ехолокації (рис. 2.11). Більшість ультразвукових датчиків не обчислюють самі відстань до об’єкта, а відправляють у мікроконтролер інформацію про час повернення сигналу на датчик.

Всі необхідні дані для розрахунку відстані виконує мікропроцесор, підключений до датчика. Але є такі далекоміри, які самі виконують необхідні обчислення і на виході видають готовий результат.

Датчик US-100 підтримує обидва режими. З одного боку це дозволяє використовувати його в схемах і програмах, призначених для датчиків першого типу (наприклад, HC-SR04). З іншого - можна звільнити контролер від обчислень, описаних вище, і отримати від датчика готовий результат вимірюву через послідовний інтерфейс UART. А враховуючи, що US-100 має вбудовану

термокомпенсацію, результати розрахунків на стороні датчика будуть більш точними. А разом з цим можна отримувати дані про відстань та інформацію про температуру навколишнього середовища, що, у свою чергу, дозволяє використовувати датчик для інших цілей. Наприклад, до виміряних значень з інших датчиків можна внести необхідні поправки [7].

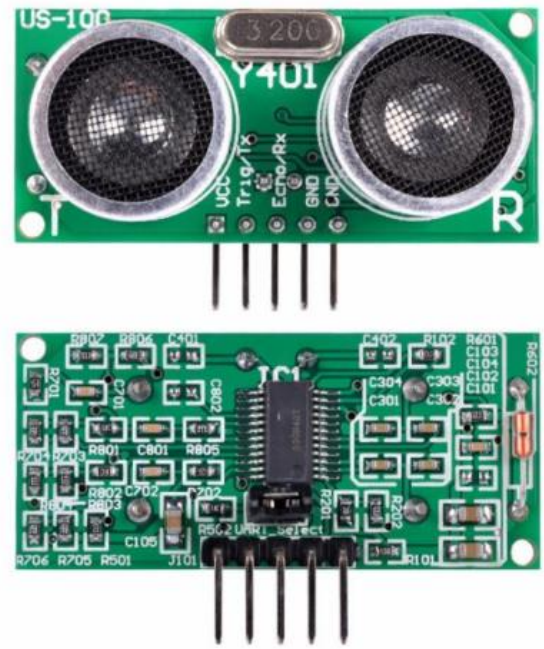


Рисунок 2.11 – Зовнішній вигляд ультразвукового датчика US-100

Для моделювання руху робота, що оминає перешкоди використаємо відоме середовище Webots.

Для забезпечення нормального функціонування робота необхідно використати 3 основні бібліотеки: DistanceSensor.hpp, Motor.hpp, Robot.hpp.

- DistanceSensor.hpp

Ця бібліотека використовується саме для сенсорів відстані. Тобто, за допомогою функцій даної бібліотеки можливо визначити відстань до об’єктів. Вузол DistanceSensor можна використовувати для моделювання загального датчика, інфрачервоного датчика, датчика сонара або лазерного далекоміра. Це моделювання пристрою виконується шляхом виявлення зіткнень між одним або кількома сенсорними променями та об’єктами навколишнього середовища. У випадку загального, сонарного та лазерного типу зіткнення відбувається з обмежуючими

об’єктами твердих вузлів, тоді як виявлення зіткнень інфрачервоних променів використовує самі тверді вузли.

Для обчислення відстані від об'єкта використовуються два різні методи. Метод «Середнє значення» обчислює середнє значення відстаней, виміряних усіма променями, тоді як метод «Найближче значення» використовує найкоротшу відстань, виміряну будь-яким променем.

- Motor.hpp;

Дана бібліотека використовується для забезпечення функціонування моторів коліс робота для його переміщення та руху по заданій траєкторії.

Моторний вузол — це абстрактний вузол (не створений), похідними класами якого є RotationalMotor і LinearMotor. Ці класи можна використовувати в механічній симуляції для приводу руху суглоба, створюючи таким чином рух вздовж або навколо однієї з його осей.

Як вже було описано вище – давачі відстані бувають різних типів, вони зображені в таблиці 3.1

Таблиця 3.1 – характеристика і порівняння давачів відстані різних типів

	універсальний	інфрачервоний	сонар	лазер
Кількість сенсорів	>0	>0	>0	1
Розрахунок відстані	Середнє значення	Середнє значення	Найближче значення	Найближче значення
Гаусова ширина	+	+	-	-
Чутливість до червоних предметів	-	+	-	-
Малює червону пляму	-	-	-	+
Ігнорує прозорі предмети	-	+	-	+

RotationalMotor може живити HingeJoint (відповідно Hinge2Joint), якщо встановлено всередині поля пристрою (відповідно device або device2) цих вузлів. Потім він здійснює обертальний рух навколо обраної осі. Аналогічно, LinearMotor може привести в дію SliderJoint, створюючи ковзний рух вздовж своєї осі.

Стандартний спосіб управління двигуном - це безпосереднє керування положенням. Користувач вказує цільове положення за допомогою функції wb_motor_set_position, потім P-контролер враховує бажану швидкість, прискорення і силу двигуна, щоб перемістити двигун в цільове положення.

У Webots управління положенням здійснюється у три етапи. Перший етап виконується заданим користувачем контролером, який вирішує, яке положення, швидкість, прискорення та силу двигуна повинні використовуватись. Другий етап виконується P-контролером двигуна, який обчислює поточну швидкість двигуна. Нарешті третій етап виконується на фізичному симуляторі.

- Robot.hpp;

Вузол Робот може бути використаний як основа для створення робота, наприклад, шарнірного робота, гуманоїдного робота, колісного робота.

Прикладами функцій цієї бібліотеки можуть бути:

- wb_robot_step функція. Ця функція синхронізує дані датчика та виконавчого механізму між Webot та контролерами. Якщо wb_robot_step функція не викликається, у Webots не буде спрацьовування та оновлення датчиків у контролері.

У duration визначає кількість часу, виражену в мілісекундах, яка не повинна бути змодельована, поки функція wb_robot_step повертає. Залежно від складності моделювання та режиму виконання, функція може повертатися швидко. Коли він повернеться, час моделювання закінчиться. Іншими словами, фізика працює протягом зазначеного часу: об'єкти можуть рухатися, двигуни можуть працювати, значення датчиків можуть змінюватися і т. д.

2.3 Розробка алгоритму управління робототехнічним пристроєм при подоланні перешкод

Узагальнену схему алгоритму управління робототехнічним пристроєм при подоланні перешкод представлено на рисунку 2.12. Дана схема представляє опис узагальнених кроків, які виконує робот при навігації до цілі на високому рівні деталізації.

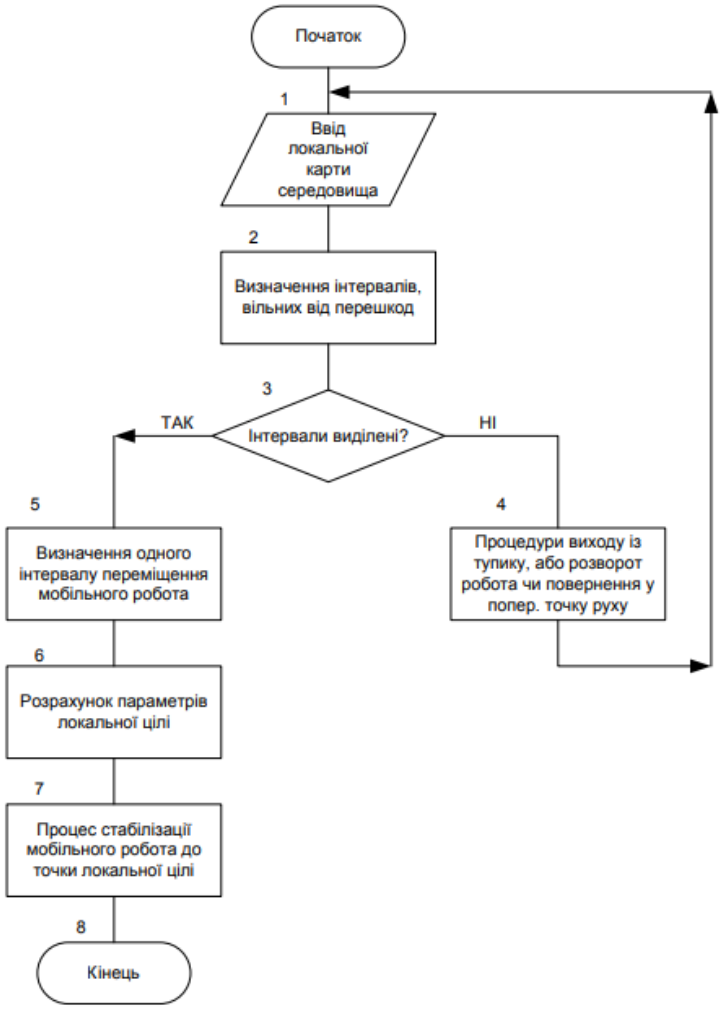


Рисунок 2.12 – Узагальнена схема алгоритму управління робототехнічним пристроєм при подоланні перешкод

Вхідною інформацією для виконання алгоритму є локальна мапа середовища, що представлена у полярній системі координат. На такій локальній мапі представляються лінійні відстані до перешкод у заданому кутовому діапазоні (0…1800). На зазначеній мапі відображаються перешкоди, що фіксуються сенсорними пристроями робота на межі їх чутливості, а тому вона, як правило, обмежується деяким визначеним радіусом для усунення шуму. Таким чином на локальній мапі представляються положення перешкод відносно положення мобільного робота. Такий спосіб є достатньо компактним і зручним. Після даного кроку здійснюється перехід до сітко-базованої карти, де визначаються декартові координати розміщення граней об’єктів, які є перешкодами. Такий перехід до сітко-базованого представлення викликаний необхідністю визначення напряму руху до цілі

робота пилососа, яка задається декартовими координатами (представляється сітко-базованим способом), а також для виконання задач локалізації робота. Таким чином, вихідною інформацією даного блоку є позначення позицій перешкод у декартових координатах.

Наступним блоком алгоритму є становлення інтервалів, вільних від перешкод. Базуючись на розміщенні перешкод, що виконані на попередньому кроці, зазвичай, виділяються декілька ділянок, за якими робот може переміщуватись. Тому, результатом виконання даного блоку є визначення інтервалів можливого переміщення робота. Наступним блоком схеми алгоритму є умова, що встановлює факт існування інтервалів, вільних від перешкод для подальшого переміщення робота. Такі інтервали можуть бути відсутні. Тому метою зазначеної умови є визначення тупикових ситуацій, в яких може опинитись робот в результаті переміщень. У випадку наявності тупику (випадок хибності умови), робот визначає процедури виходу із тупику, або ж здійснює розворот чи повернення у попередню точку свого руху.

У випадку істинності умови, встановлюється факт існування інтервалів, куди робот може перемістити для наближення до цілі. Після визначення інтервалу для подальшого переміщення робота, визначаються декартові координати точки локальної цілі, та кута під яким необхідно перебувати роботу при досягненні локальної цілі. Такий процес передбачає спосіб досягнення точки цілі під визначеним у попередньому блоці кутом. Після завершення виконання описаних вище кроків, положення робота вважається у точці локальної цілі, після чого, покроково визначаються подальші локальні цілі до моменту досягнення глобальної точки руху робота.

2.4 Висновок

В даному розділі розроблено систему управління рухомим робототехнічним пристроєм при подоланні перешкод. Обгрунтовано вимоги до обладнання та вибору елементної бази для створюваного пристрою. Розроблено алгоритм управління робототехнічним пристроєм при подоланні перешкод.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ МОДУЛЯ УПРАВЛІННЯ РОБОТОТЕХНІЧНИМ ПРИСТРОЄМ ПРИ ПОДОЛАННІ ПЕРЕШКОД

3.1 Обґрунтування вибору мови програмування та середовища розробки

Програму для функціонування робота написано на мові C++.

C++ було розроблено як універсальну мову зі статичними типами даних, ефективністю та переносимістю мови C.

Розроблений так, щоб безпосередньо та всебічно підтримувати безліч стилів програмування (процедурне програмування, абстракцію даних, об'єктно-орієнтоване програмування та узагальнене програмування).

Розроблений так, щоб давати програмісту свободу вибору, навіть якщо це дає можливість вибирати неправильно.

Розроблений так, щоб максимально зберегти сумісність із C, тим самим уможливлюючи легкий перехід від програмування на C.

Уникає таких особливостей, які залежать від платформи чи не є універсальними.

Не накладає жодного надлишкового навантаження на програму, яка не використовує жодних можливостей.

Розроблено так, щоб не вимагати надто ускладненого середовища програмування.

C++ вважається однією з дуже швидких мов програмування.

Переваги C++ над C:

- Велика безпека.
- Можливість писати узагальнений код за допомогою шаблонів.
- Можливість використовувати об'єктно-орієнтований підхід.
- Управління ресурсами за допомогою RAII.
- Спрощення коду за рахунок перевантаження функцій та операторів.
- Простіша обробка помилок за рахунок винятків.

Недоліки:

- Довга компіляція.
- Більший обсяг згенерованого машинного коду.

3.2 Побудова структури робота при подоланні перешкоди

Перш ніж приступити до виготовлення робота необхідно спроєктувати його структуру. На рисунку 3.1 зображено структуру робототехнчного пристрою при подоланні перешкод.

Як можна побачити з рисунку, необхідно створити тіло робота, яке буде геометричного, hingeJoint (4 штуки) та прикріпити до них 4 колеса. Більш того, необхідно створити датчики, які мають форму куба.

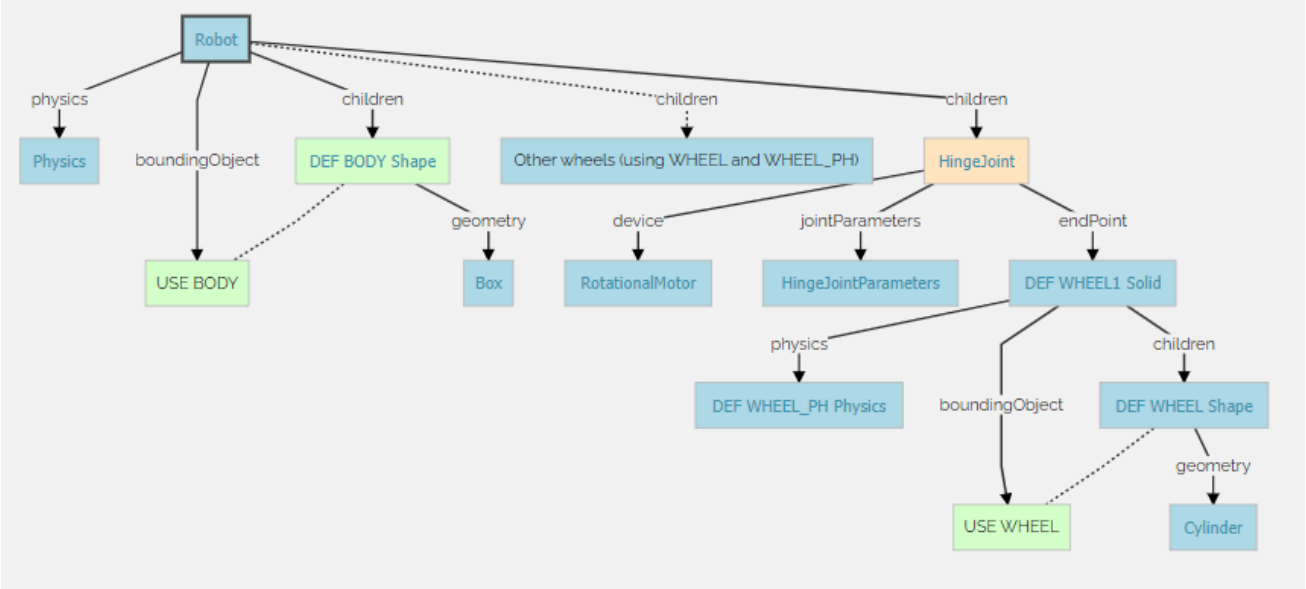


Рисунок 3.1 – Структура робототехнчного пристрою при подоланні перешкод

Початкове положення колеса визначається полями трансляції та обертання вузла Solid. Початок обертання (anchor) і вісь обертання (axis) визначаються дочірнім елементом HingeJointParameters вузла HingeJoint.

Принцип роботи HingeJoint зображено на рисунку 3.2, 3.3.

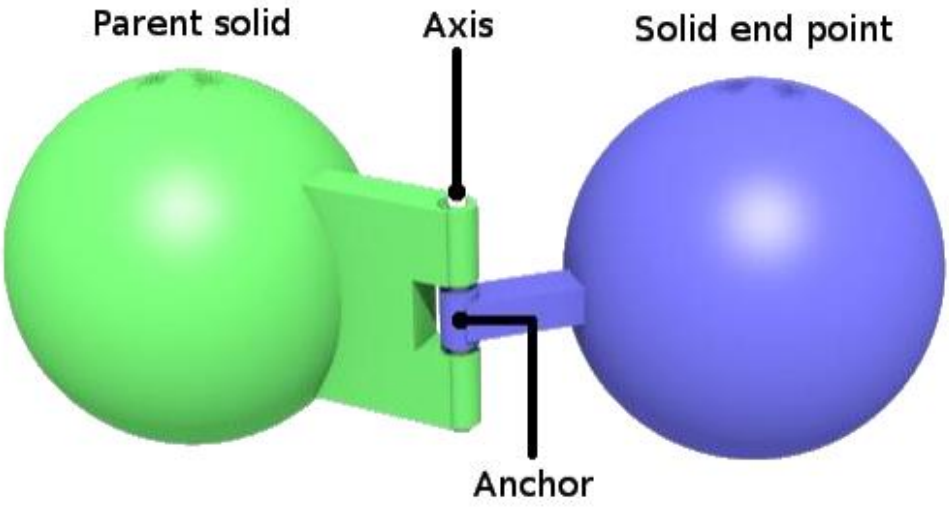


Рисунок 3.2 – Графічне представлення вузла HingeJoint

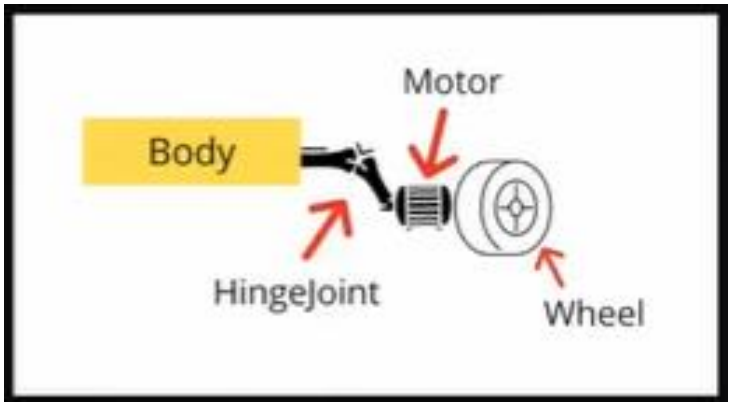


Рисунок 3.3 – Принцип кріплення колеса до «тіла» робота

3.3 Моделювання робототехнічного пристрою при подоланні перешкод в середовищі Webots

Для розробки та програмування робота необхідно його створити. Це включає в себе кілька етапів, кожен з яких є критично важливим для успішного функціонування робота. Щоб зробити робота, який буде долати перешкоди, треба виділити наступні компоненти машини:

- тіло робота;
- 4 колеса;
- датчики відстані.

Модель та план робота можна побачити на рисунках 3.4, 3.5.

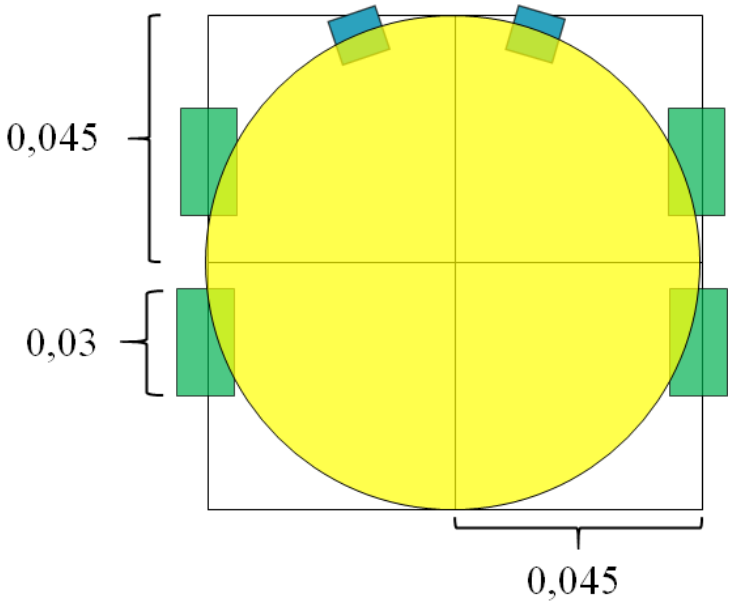


Рисунок 3.4 – Зображення зовнішнього виду робота (вид зверху)

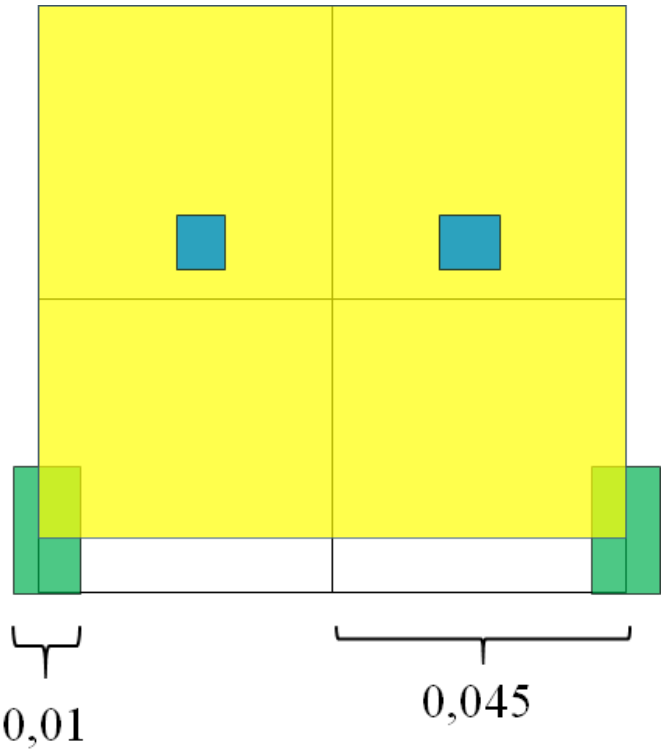


Рисунок 3.5 – Зображення зовнішнього виду робота (вид спереду)

Як видно з наведених вище рисунків жовтим кольором зображено тіло робота, зеленим – колеса, а синім – давачі відстані.

Процес створення робота починаємо з основи – корпуса або тіла робота. В даному проєкті воно зображено як циліндр.

Спочатку необхідно створити самого робота, для цього потрібно виконати наступні кроки:

Add -> body -> robot

Robot-> children -> add-> base node -> transform

Після виконання даних дій з’являється на арені робот, але він не має форми, тому ми його не можем побачити. Наступним кроком стане саме надання роботу форми та інших властивостей.

Transform -> children -> add -> base node -> shape

Shape -> geometry -> cylinder

Тепер на арені з’явився циліндр, для якого необхідно далі задати параметри: розмір, розміщення, колір і т. д.

Потрібно задати такі параметри:

Cylinder -> height = 0.08.

Cylinder -> radius = 0.045.

Задані параметри вже бути обґрунтовані в минулих підрозділах, також там можна найти план побудови робота.

Далі необхідно задати колір/текстуру для циліндра.

Shape -> appearance -> PBRAppearance.

PBRAppearance -> r=1; g=1; b=0.

PBRAppearance -> roughness = 1.

PBRAppearance -> metalness = 0.

Потім необхідно додати інші властивості та фізику. Для цього виконуємо наступні дії:

Bounding object -> use -> body.

Physics -> add.

Також необхідно задати розміщення фігури. Вона має бути трохи вище полу, оскільки ще будуть колеса.

Після виконання всіх вищевказаних дій, буде створене тіло робота (циліндр). Його можна побачити на рисунку 3.6.

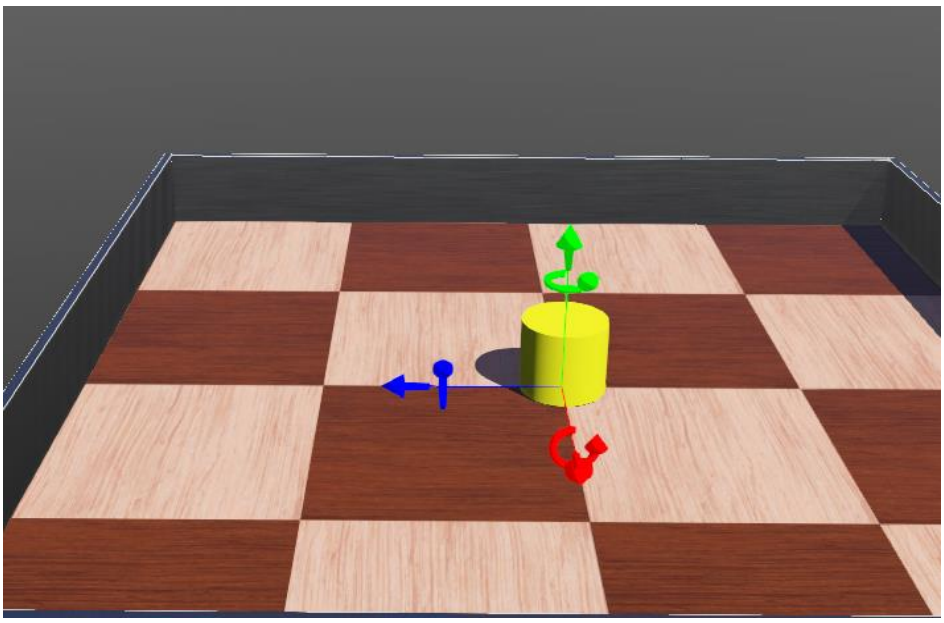


Рисунок 3.6 – Зображення створеного тіла робота

Для забезпечення нормального функціонування робота, потрібно створити засоби для його переміщення. Тому наступним етапом є створення коліс.

Як вже було описано і обґрунтовано раніше, необхідно створювати вузли hingeJoint. Будову коліс описано у підрозділі вище. можна зробити, виконавши наступні кроки:

Robot -> children -> add ->base node -> hinge Joint.

Hinge Joint -> hingejoint parameters.

Тепер потрібно додати мотор до колеса, щоб надати йому змогу обертатись, щоб робот пересувався.

Hinge Joint -> device -> rotation motor.

Тепер вже всі підготовчі кроки виконано і можна приступити безпосередньо до створення коліс.

Hinge Joint -> endpoint -> Solid

Колесо створене, але його не видно, бо в нього нема параметрів. Тому наступним кроком буде надавання форми.

Endpoint -> children -> shape

Shape -> geometry -> cylinder.

Також необхідно вказати параметри фігури коліс, тобто параметри циліндра.

Cylinder -> hight = 0.01

Cylinder -> radius = 0.025

Далі вказуємо позицію та розміщення даного колеса у просторі.

Rotation -> x=0; y=0; z=1; ang=1.57

Transition -> x=0.045; y=0.025

Також потрібно вказати такі ж параметри як і в transition в вузлі joint parameters

Hinge Joint ->join parameters -> abchor -> x=0.045; y=0.025

Тепер надаємо колір та інші властивості колесу, виконавши наступні дії:

Shape -> appearance -> add -> PBRappearence

PBRappearence -> colour -> r=0; g=1; b=0

PBRappearence -> roughness = 1

PBRappearence -> mealness = 0

Отже, в результаті виконання даних дій було створено колесо. Результат створення колеса можна побачити на рисунку 3.7.

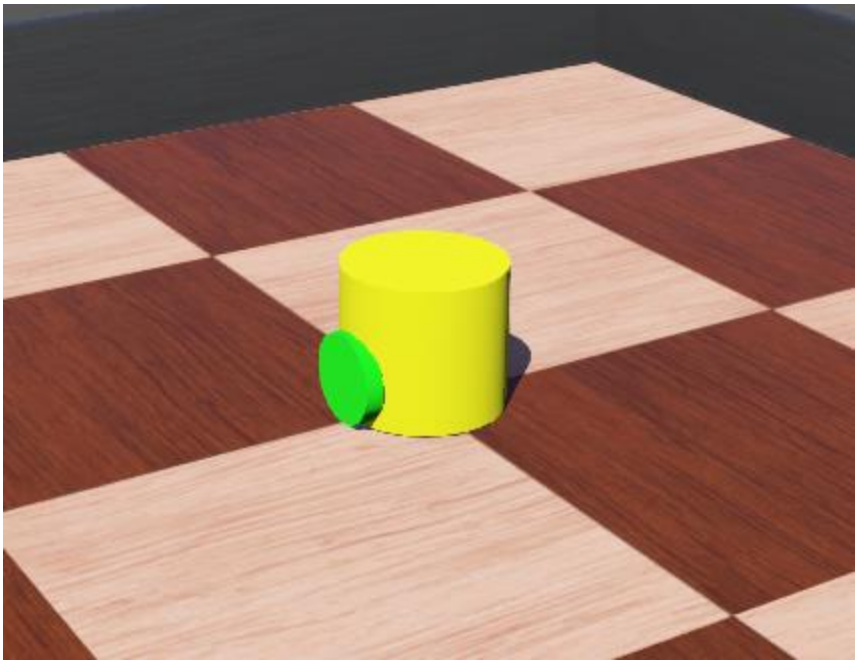


Рисунок 3.7 – Зображення результатів створення колеса

Оскільки для роботи робота необхідно 4 колеса, то потрібно повторити вказані вище кроки ще 3 рази.

Результат створення 4 коліс можна побачити на рисунку 3.8.

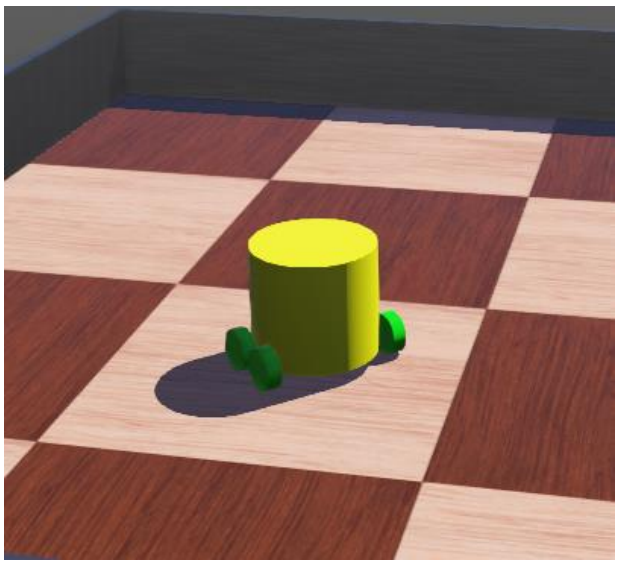


Рисунок 3.8 – Зображення результату створення всіх коліс робота, що оминає перешкоди

Останнім кроком в створенні робота є створення датчиків відстані. Для цього потрібно виконати наступні операції:

Robot -> children ->add -> base node -> distance sensor

Далі потрібно надати форму об’єкту:

Distance sensor -> children -> add ->shape

Shape -> geometry -> add -> box

Тепер ми маємо створені датчики, потрібно задати параметри розміру для куба.

Shape -> size -> x=y=z=0.001

Shape -> appearance -> PBApperence

PBApperence -> base colour -> r=0; g=1; b=1;

PBApperence -> roughness = 1

PBApperence -> metalness = 0

Після виконання цих операцій приступим до задання розташування сенсорів та надання їм фізичних властивостей.

Оскільки датчиків 2, то необхідно повторити ці дії двічі, змінюючи лище розташування датчиків (вісь OX)

Результат виконання даних операцій та вже готового робота можна побачити на рисунках 3.9, 3.10.

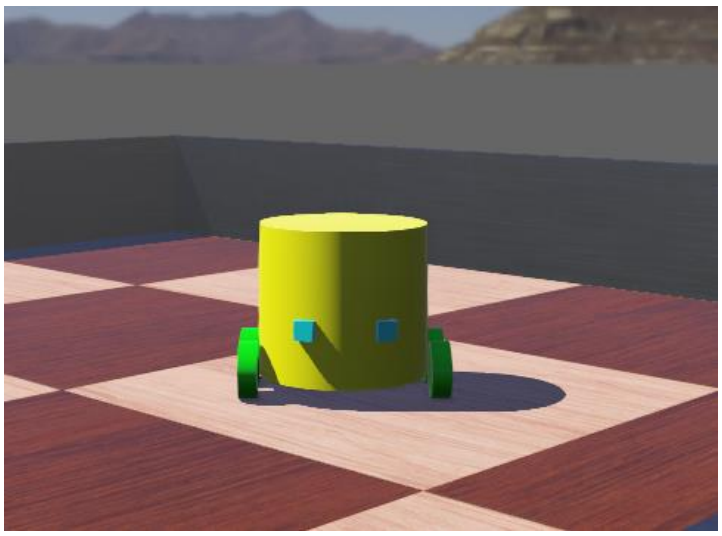
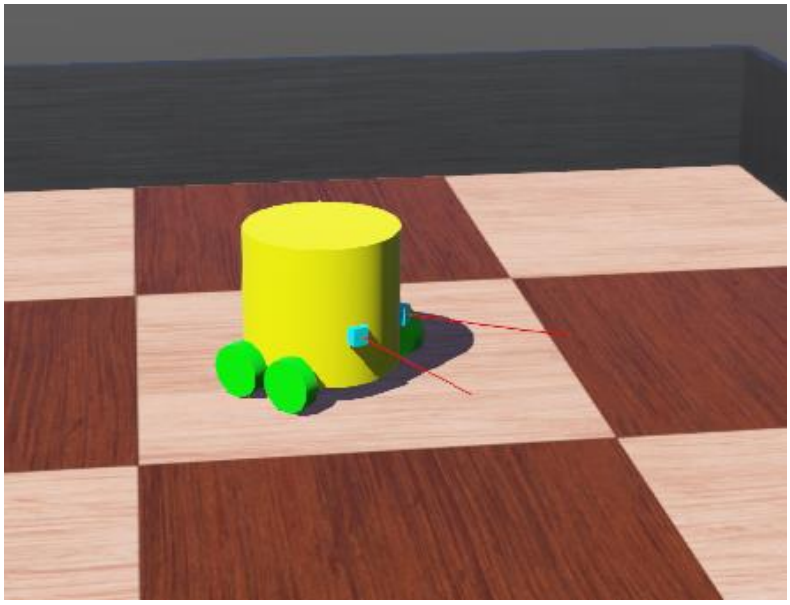


Рисунок 3.9 – Зображення результату створення всіх коліс робота, що оминає перешкоди



Рисунки 3.10 – Зображення готового робота

Як можна побачити на рисунку, відстань, на яку реагують датчики зображено червоною прямою

3.4 Програмування робота при подоланні перешкод

#include <webots/DistanceSensor.hpp>

#include <webots/Motor.hpp>

#include <webots/Robot.hpp>

Додавання бібліотек, для забезпечення роботи функцій програми

```
#define TIME_STEP 64
```

Оголошення кроку робота

```
using namespace webots;
```

Оголошення робота

```
int main(int argc, char **argv) {
  Robot *robot = new Robot();
  DistanceSensor *ds[2];
  char dsNames[2][10] = {"ds_right", "ds_left"};
```

Оголошення та запуск робота і його сенсорів

```
  for (int i = 0; i < 2; i++) {
    ds[i] = robot->getDistanceSensor(dsNames[i]);
    ds[i]->enable(TIME_STEP);
  }
```

Опис дій для датчиків відстані.

```
  Motor *wheels[4];
  char wheels_names[4][8] = {"wheel1", "wheel2", "wheel3", "wheel4"};
```

Оголошення моторів та створення змінних для коліс.

```
  for (int i = 0; i < 4; i++) {
    wheels[i] = robot->getMotor(wheels_names[i]);
    wheels[i]->setPosition(INFINITY);
    wheels[i]->setVelocity(0.0);
  }
```

Забезпечення постійної роботи двигунів без зупинок.

```
  int avoidObstacleCounter = 0;
  while (robot->step(TIME_STEP) != -1) {
    double leftSpeed = 4.0;
    double rightSpeed = 4.0;
```

Оголошення швидкостій для обох двигунів.

```
    if (avoidObstacleCounter > 0) {
      avoidObstacleCounter--;
```

```
    leftSpeed = 1.0;
    rightSpeed = -1.0;    }
```

Швидкість повороту

```
else {
    for (int i = 0; i < 2; i++) {
      if (ds[i]->getValue() < 950.0)
      avoidObstacleCounter = 100;      }
```

Зчитуванян даних сенсорів.

```
   wheels[0]->setVelocity(leftSpeed);
   wheels[1]->setVelocity(rightSpeed);
   wheels[2]->setVelocity(leftSpeed);
   wheels[3]->setVelocity(rightSpeed);
```

Вказуємо швидкість для усіх коліс.

Щоб протестувати роботу робота в віртуальному середовищі, створимо перешкоди на його шляху. У даному проєкті використовуються куби різних розмірів у якості перешкод. Арена з перешкодами зображена на рисунку 3.11.

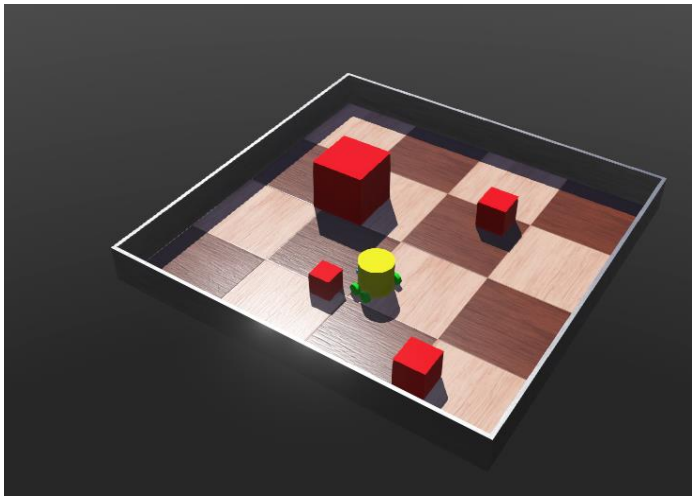


Рисунок 3.11 – Зображення арени віртуального середовища з перешкодами у вигляді кубів різних розмірів

Далі необхідно запустити програму і прослідкувати за зміною траєкторії робота (рис. 3.12)

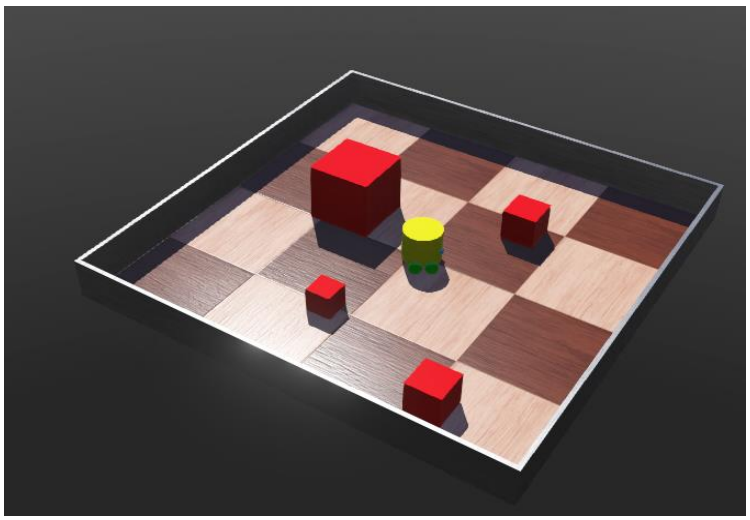


Рисунок 3.12 – Зображення руху робота

Як видно з рисунку 3.12, робот змінив своє положення, огинаючи перешкоди на своєму шляху, щоб уникнути зіткнення.

Для програмування апаратної частини робототехнічного пристрою при подоланні перешкод на базі контроллера Arduino використовують інтегроване середовище розробки Arduiono IDE, яке можна завантажити на офіційному сайті проекту [5]. Мовою програмування є C++. Програми для контроллера Arduino прийнято називати скетчами. Для компіляції скетчу слід натиснути кнопку «Перевірити» на панелі швидкого доступу, для завантаження скетчу на плату необхідно з'єднати плату з комп'ютером за допомогою USB кабелю та натиснути кнопку «Вивантажити» (рис 3.13).

Програма повинна містити дві обов'язкові функції setup() та loop(). Функція setup() викликається одноразово при старті, а функція loop() виконується як нескінченного циклу.

Мова програмування Arduino складається з трьох основних частин. Перш за все, функції, які дозволяють керувати дошкою. Використання функцій можна розбирати символи, виконувати математичні операції та виконувати різні інші завдання, такі як digitalRead() і digitalWrite() дозволяє читати або записувати значення.

Кожний скетч, що написаний для Arduino, містить дві функції. Це SetUp() і loop(). Скетч завжди починається з setUp(), який виконується один раз після ввімкнення чи перезавантаження плати. Після його створення використовують loop(), щоб повторити програму багато разів, доки не вимкнеться живлення.

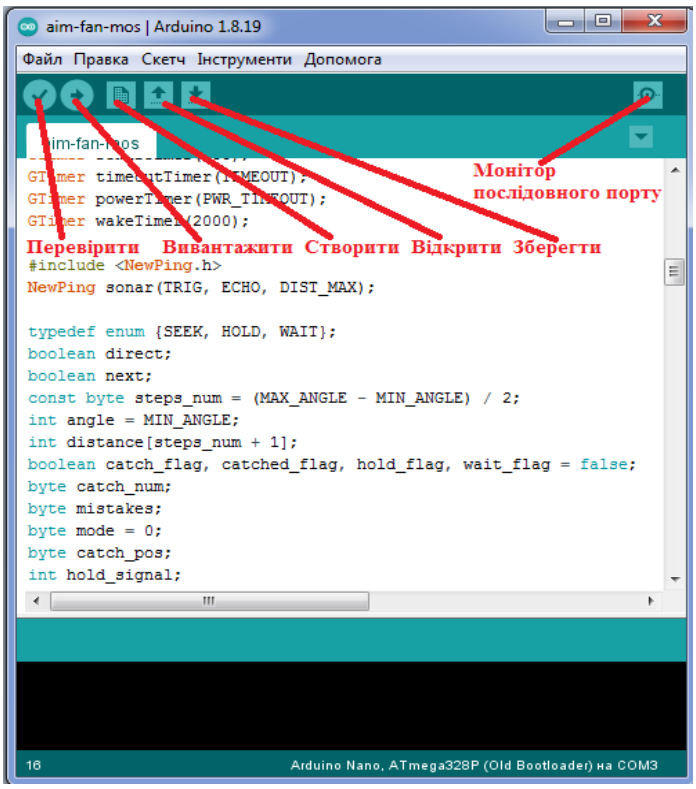


Рисунок 3.13 – Зовнішній вигляд вікна Arduino IDE

Існують також значення Arduino, які представляють константи та змінні. Більшість типів даних (масив, bool, char, float тощо) схожі на існуючі у C++. Можливе перетворення типу. Остання частина мови Arduino називається структурою. Вона містить невеликі фрагменти коду, наприклад оператори

Отже, необхідно зібрати апаратну частину за схемою на рис. 3.14.

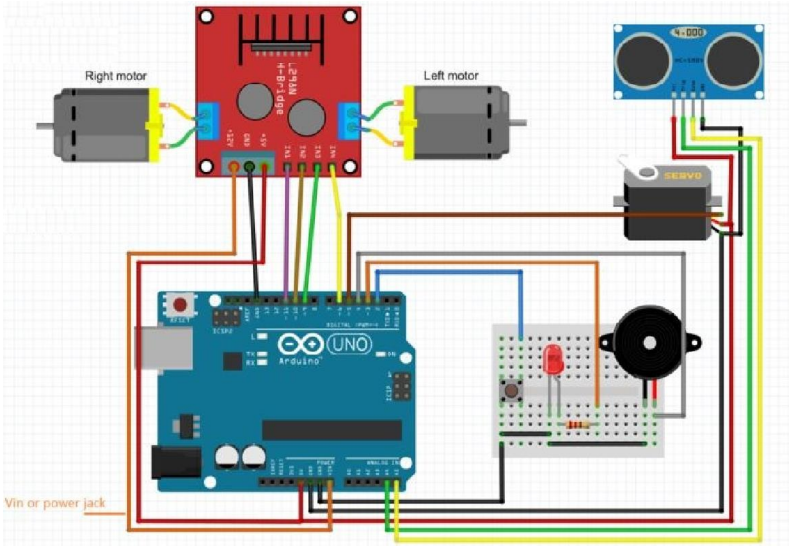


Рисунок 3.14 – Монтажна схема робототехнічного пристрою при подоланні перешкод

Використовуємо при цьому 1 плату Arduino Uno, драйвер двигунів L298, 2 двигуна з колесами, 1 серво-двигун для повороту ультразвукового давача HC-SR04, монтажну плату, та корпус нашого робототехнічного пристрою (рис. 3.15).

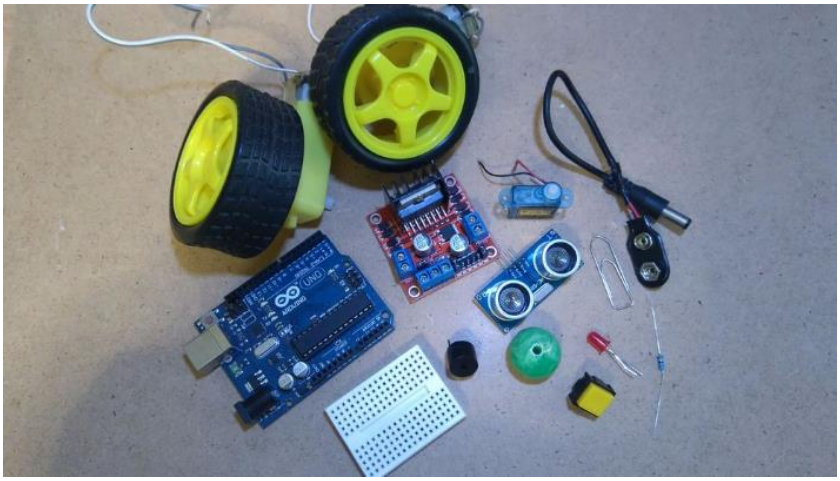


Рисунок 3.15 – Апаратна частина робототехнічного пристрою при подоланні перешкод

Після правильного поєднання всіх складових апаратна частина робототехнічного пристрою при подоланні перешкод матиме вигляд (рис. 3.16).

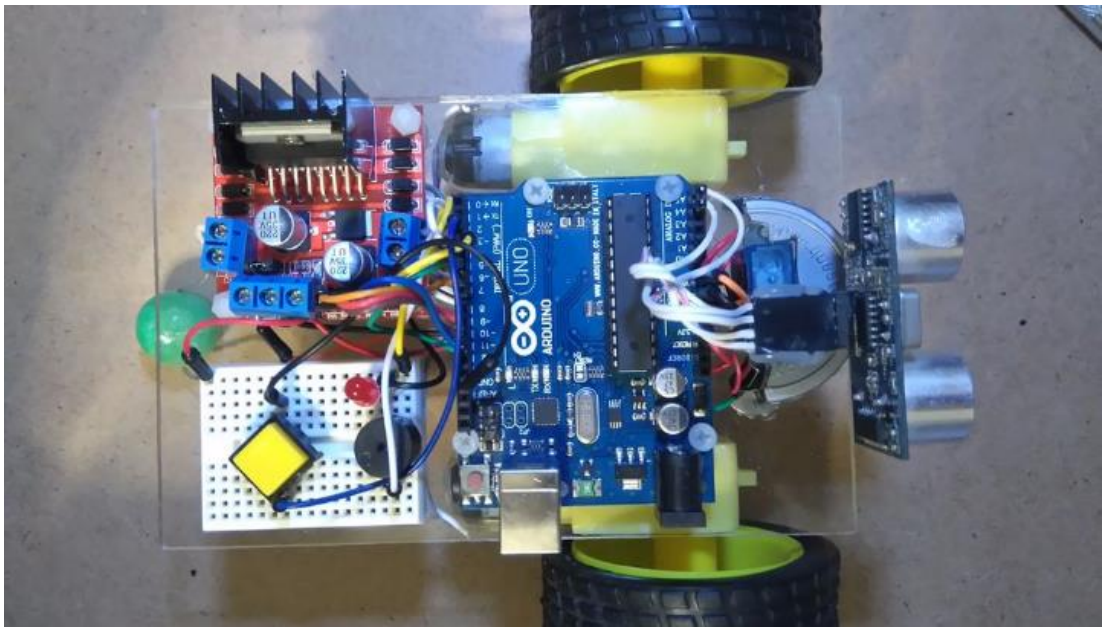


Рисунок 3.16 – Зовнішній вигляд робототехнічного пристрою при подоланні перешкод

Для правильної роботи робототехнічного пристрою при подоланні перешкод необхідно завантажити в Arduino Uno програму для його управління.

3.5 Тестування розробленого програмного забезпечення управління робототехнічним пристроєм при подоланні перешкод

Для тестування розробленого робототехнічного пристрою запускаємо його кнопкою ввімкнення, яка знаходиться у верхній частині його корпуса. Робот починає рухатися вперед до моменту, поки його ультразвуковий давач не зустріне перешкоду. При цьому ультразвуковий давач повертається вправо-вліво на кут приблизно 90^0 за допомогою серво-двигуна, на якому він і знаходиться. Якщо відстань до перешкоди менша за критичну (приблизно 10см), програма керування зупинить двигуни робота, сформує сигнал повернути в сторону, протилежну центру перешкоди на 10^0, якщо при цьому ультразвуковий давач все одно «бачить» перешкоду (наприклад, якщо перешкода широка), то команда повороту повторюється. Якщо по прямому курсу руху робототехнічного пристрою перешкод немає (ультразвуковий давач їх «не бачить»), то програма керування вмикає двигуни і робот продовжує рух вперед. Таким чином робототехнічний пристрій може рухатися в певному просторі абсолютно автономно (рис .3.17).

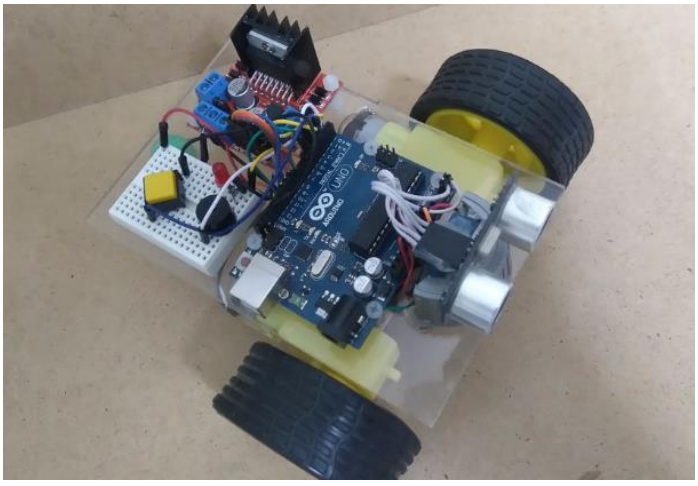


Рисунок 3.17 – Робототехнічний пристрій для подолання перешкод

Розроблений робототехнічний пристрій був протестований спеціалістами в галузі розробки робототехніки. Для тестування на достовірність правдивості відповідей було обрано десять спеціалістів, що працюють в галузі розробки робототехніки. Оцінка роботи виставлялась від 1 до 10, в залежності від того була дана розробка корисною чи ні. Результати тестування наведені в таблиці 3.1

Таблиця 3.1 - Результати тестування робота, при подоланні перешкод

Спеціаліст	1	2	3	4	5	6	7	8	9	10
Оцінка	8	8	8	10	8	9	8	7	9	10

Проаналізувавши дані, маємо загальну оцінку 85 із 100. Отже, можна дійти висновку що якість управління робототехнічним пристроєм при подоланні перешкод складає 85%. Це свідчить про досить високий рівень підвищення якості управління роботом при подоланні перешкод.

3.6 Висновок

В даному розділі обгрунтовано вибір мови програмування та середовища розробки робототехнічного пристрою при подоланні перешкод. Побудована структура робота, що оминає перешкоди. Проведено моделювання робота, що оминає перешкоди в середовищі WeBots. Здійснено реалізацію розробленого робота у вигляді лабораторного макету. Проведено тестування розробленого програмного забезпечення для управління робототехнічним пристроєм при подоланні перешкод, яке підтвердило, що якість управління робототехнічним пристроєм при подоланні перешкод складає 85%.

ВИСНОВКИ

Під час виконання бакалаврської дипломної роботи було розроблено програмний модуль управління робототехнічним пристроєм при подоланні перешкод.

Проаналізовано основні види рухомих мобільних роботів та сучасні технології управління робототехнічними пристроями при подоланні перешкод. Створено узагальнену схему управління мобільним робототехнічним засобом, який оминає перешкоди, що трапляються на шляху його слідування. Обгрунтовано доцільність створення програмного модуля управління робототехнічним засобом при подоланні перешкод.

Розроблено систему управління рухомим робототехнічним пристроєм при подоланні перешкод. Обгрунтовано вимоги до обладнання та вибору елементної бази для створюваного пристрою. Розроблено алгоритм управління робототехнічним пристроєм при подоланні перешкод. Здійснено моделювання роботи робототехнічного пристрою при подоланні перешкод в середовищі WeBots, яке довело правильність функціонування всіх його складових. Реалізовано робототехнічний пристрій, який оминає перешкоди, що трапляються на шляху у вигляді лабораторного макету. Проведено тестування розробленого програмного забезпечення для управління робототехнічним пристроєм при подоланні перешкод, яке підтвердило, що якість управління робототехнічним пристроєм при подоланні перешкод складає 85%.

Отже всі поставлені задачі виконано, мету роботи досягнуто.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Програмні засоби контролю ЕОМ. − Режим доступу: http://rdmcn.ptngu.com/lekcyi/13-lekcya-8.html.
2. Лувішіс І. Коротко про історію розвитку комп'ютерів/ Лувішіс І., Зарубін Ю., Мазо Б.: Комп'ютер Прес - 1996 - N5.
3. Іванов О.В. Історія створення комп'ютерів/ Іванов О.В.: Комп'ютер Прес - 1996 - N5.
4. 3DMark − Режим доступу: https://uk.wikipedia.org/wiki/3DMark (дата звернення черв. 2022). − Назва з екрана.
5. Мельник А.О. Архітектура комп’ютера . Наукове видання. – Луцьк: Волинська обласна друкарня, 2008. – 470 с.
6. Заміховський Л.М., Калявін В.П. Основи теорії надійності і технічної діагностики систем: Навчальний посібник.–Івано-Франківськ: Вид-во “Полум’я”, 2009.– 360 с.
7. Тхір І.Л., Калушка В.П., Юзьків А.В. Посібник користувача ПК. Третє видання. — Тернопіль: “Підручники та посібники”, 2006. -1024с.
8. Шимон О. М. Лекція №1. Апаратне та програмне забезпечення ПК.: Конспект лекцій; Житомирський держ. унів. ім. І. Франка. – Житомир, 2006 – 17с.:Режим доступу: http://eprints.zu.edu.ua/18/1/Konspect_modul_1_Windows.pdf
9. Dynamic-link libraries [Electronic resource] – 2013. Access mode: https://msdn.microsoft.com/en-us/library/windows/desktop/ms682589(v=vs.85).aspx. – Title from the screen.
10. How to change the color of progressbar in C# .NET 3.5? Режим доступу: https://stackoverflow.com/questions/778678/how-to-change-the-color-of-progressbar-in-c-sharp-net-3-5.
11. Unified Modeling Language – Вікіпедія: веб-сайт. - Режим доступу: URL: https://uk.wikipedia.org/wiki/ Unified_Modeling_Language.
12. Microsoft Visual Studio – Вікіпедія: веб-сайт. Режим доступу: URL: https://uk.wikipedia.org/wiki/Microsoft_Visual_Studio

13. Ситник В. Ф., Писаревська Т. А., Єрьоміна Н. В., Краєва О. С. Основи інформаційних систем: Навчальний посібник Київ, 2005. 308 с.
14. Josuttis, Nicolai M. The C++ standard library : a tutorial and reference / Nicolai M. Josuttis.—2nd ed. c. 78-80.
15. Свароп С. Г. A Bite Of Python v2.01. 2013. 159 с. с. 20-23. PyCharm [Електронний ресурс] – Режим доступу: https://www.jetbrains.com/pycharm/
16. Chollet Francois Deep Learning with Python. 2018. — 400 с.
17. Miguel Grinberg Flask Web Development. Published by O'Reilly Media, Inc. 2018.
18. C++ 17 STL. Стандартна бібліотека шаблонів. Яцек Галовіц. 2018. ISBN: 978-5-4461-0680-6
19. Arduino Pro Mini [Електронний ресурс] // doc.arduino.ua – 2021. – Режим доступу до ресурсу: https://doc.arduino.ua/ru/hardware/ProMini.
20. Arduino Nano [Електронний ресурс] // arduino.ua – 2021. – Режим доступу до ресурсу: https://arduino.ua/prod166-arduino-nano-v3-0-avratmega 328p-s-raspayannimi-razemami.
21. Методичні вказівки до виконання бакалаврських кваліфікаційних робіт для студентів спеціальності 122 «Комп'ютерні науки» [Електронний ресурс] / уклад.: А. А. Яровий, О. В. Сілагін, І. В. Богач. – Вінниця : ВНТУ, 2023. – (PDF, 54 с.).

ДОДАТКИ

ДОДАТОК А
ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Програмний модуль управління робототехнічним пристроєм при подоланні перешкод

Тип роботи: бакалаврська дипломна робота
(БДР, МКР)

Підрозділ кафедра комп'ютерних наук, ФІТА
(кафедра, факультет)

Показники звіту подібності Unicheck

Оригінальність 81,2% Схожість 18,8%

Аналіз звіту подібності (відмітити потрібне):

✓ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.

□ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.

□ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку Озеранський В.С.

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи Фойда Д.О.

Керівник роботи Озеранський В.С.

ДОДАТОК Б

ІНСТРУКЦІЯ КОРИСТУВАЧА

Для старту робототехнічного пристрою запускаємо його кнопкою ввімкнення, яка знаходиться у верхній частині його корпуса. Робот починає рухатися вперед до моменту, поки його ультразвуковий давач не зустріне перешкоду. При цьому ультразвуковий давач повертається вправо-вліво на кут приблизно 90^0 за допомогою серво-двигуна, на якому він і знаходиться. Якщо відстань до перешкоди менша за критичну (приблизно 10см), програма керування зупинить двигуни робота, сформує сигнал повернути в сторону, протилежну центру перешкоди на 10^0, якщо при цьому ультразвуковий давач все одно «бачить» перешкоду (наприклад, якщо перешкода широка), то команда повороту повторюється. Якщо по прямому курсу руху робототехнічного пристрою перешкод немає (ультразвуковий давач їх «не бачить»), то програма керування вмикає двигуни і робот продовжує рух вперед. Таким чином робототехнічний пристрій може рухатися в певному просторі абсолютно автономно (рис. Б.1).

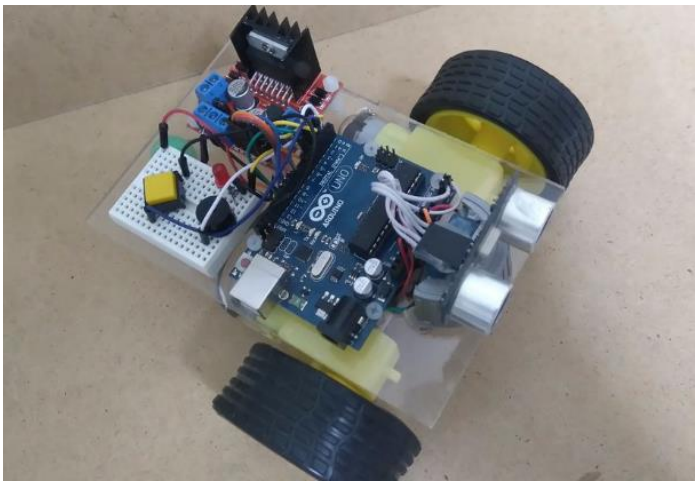


Рисунок Б.1 – Робототехнічний пристрій для подолання перешкод

ДОДАТОК В
ЛІСТИНГ ПРОГРАМИ

```
using System; using
#define Trig 8
#define Echo 9
#include <Servo.h>

Servo servo;

int kyt = 90;
int dyv_left = 0;
int dyv_right = 0;
int dyv_vpered = 0;
int chas;

const int in11 = 0;    // L298N-1 pin 1
const int in12 = 1;    // L298N-1 pin 2
const int in13 = 2;    // L298N-1 pin 2
const int in14 = 3;    // L298N-1 pin 3

const int in21 = 4;    // L298N-2 pin 1
const int in22 = 5;    // L298N-2 pin 2
const int in23 = 6;    // L298N-2 pin 2
const int in24 = 7;    // L298N-2 pin 3

void setup()
{
  servo.attach(10); // підключаємо серво на 10 пін

  pinMode(Trig, OUTPUT); // вихід
  pinMode(Echo, INPUT); // вхід

  pinMode(in11, OUTPUT);      // вихід на L298n
  pinMode(in12, OUTPUT);      // вихід на L298n
  pinMode(in13, OUTPUT);      // вихід на L298n
  pinMode(in14, OUTPUT);      // вихід на L298n
  pinMode(in21, OUTPUT);      // вихід на L298n
  pinMode(in22, OUTPUT);      // вихід на L298n
  pinMode(in23, OUTPUT);      // вихід на L298n
  pinMode(in24, OUTPUT);      // вихід на L298n
}
```

```
void u_vpered(){

    digitalWrite(in11, LOW);
    digitalWrite(in12, HIGH);

    digitalWrite(in13, LOW);
    digitalWrite(in14, HIGH);

    digitalWrite(in21, LOW);
    digitalWrite(in22, HIGH);

    digitalWrite(in23, HIGH);
    digitalWrite(in24, LOW);
}

void u_right()
{
    digitalWrite(in21, LOW);
    digitalWrite(in22, HIGH);

    digitalWrite(in23, HIGH);
    digitalWrite(in24, LOW);
}

void u_left()
{
    digitalWrite(in21, HIGH);
    digitalWrite(in22, LOW);

    digitalWrite(in23, LOW);
    digitalWrite(in24, HIGH);
 }

void stop()
{           // стоп
    digitalWrite(in11, LOW);
    digitalWrite(in12, LOW);

    digitalWrite(in13, LOW);
    digitalWrite(in14, LOW);

    digitalWrite(in21, LOW);
    digitalWrite(in22, LOW);

    digitalWrite(in23, LOW);
```

```
    digitalWrite(in24, LOW);
}

void mir_left(){
  digitalWrite(Trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(Trig, LOW);
  chas = pulseIn(Echo, HIGH);
  dyv_left = chas/58;
}

void mir_vpered(){
   digitalWrite(Trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(Trig, LOW);
  chas  = pulseIn(Echo, HIGH);
  dyv_vpered = chas/58;
}

void mir_right(){
  digitalWrite(Trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(Trig, LOW);
  chas = pulseIn(Echo, HIGH);
  dyv_right = chas/58;
}

void loop(){
  mir_right();
  if(dyv_vpered<30){
      stop();
    delay(100);
    for(kyt=90;kyt>=10; kyt--){
      servo.write(kyt);
      delay(5);
    }

   mir_right();
   delay(100);
for(kyt=10; kyt<=170; kyt++){
  servo.write(kyt);
  delay(5);
}

  mir_left();
```

```
  delay(100);
  for(kyt=170; kyt>=90; kyt--){
    servo.write(kyt);
    delay(5);
  }

  if(dyv_right < dyv_left){

    u_right();
    delay(400);
    stop();
  }

  else{
    u_left();
    delay(400);
   stop();
  }
  }
else{
u_vpered();
}
       }
```

ДОДАТОК Г
ГРАФІЧНА ЧАСТИНА

Рисунок Г.1 – Класифікація роботів

Рисунок Г.2 – Приклади мобільних роботів

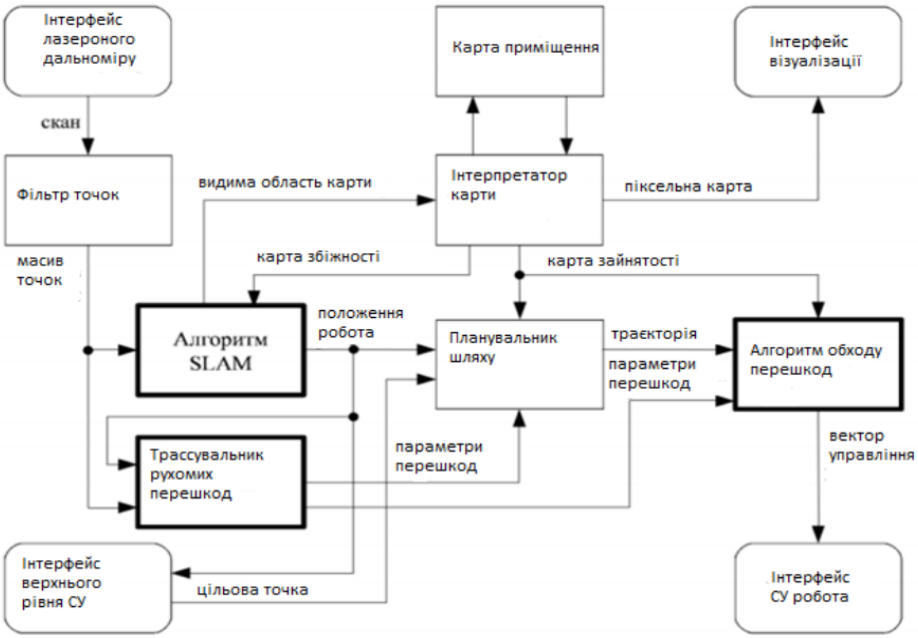


Рисунок Г.3 – Функціональна схема навігаційної системи мобільного робота

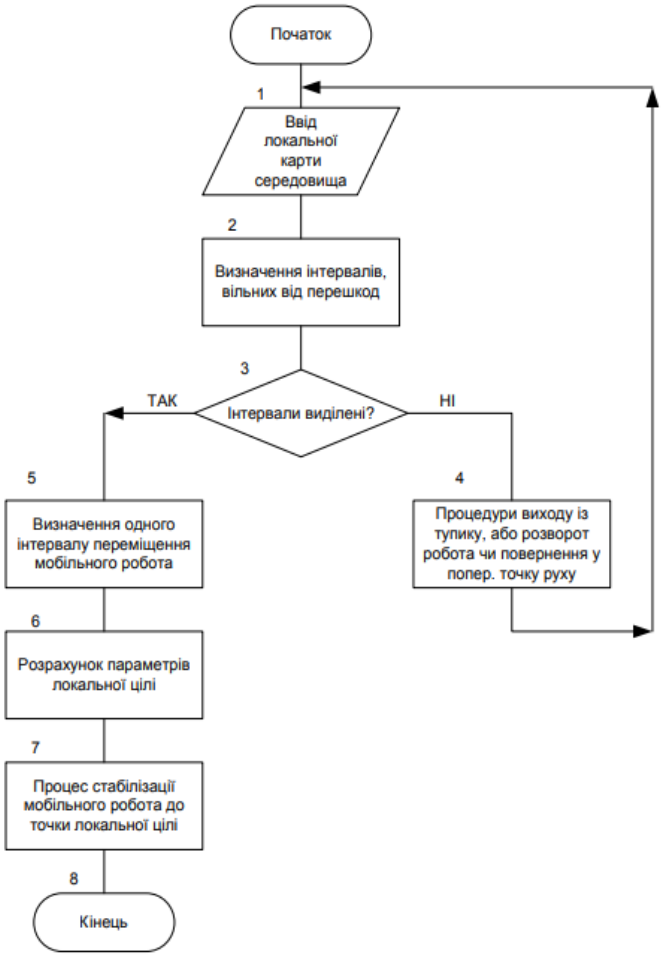


Рисунок Г.4 – Схема алгоритму управління робототехнічним пристроєм при подоланні перешкод

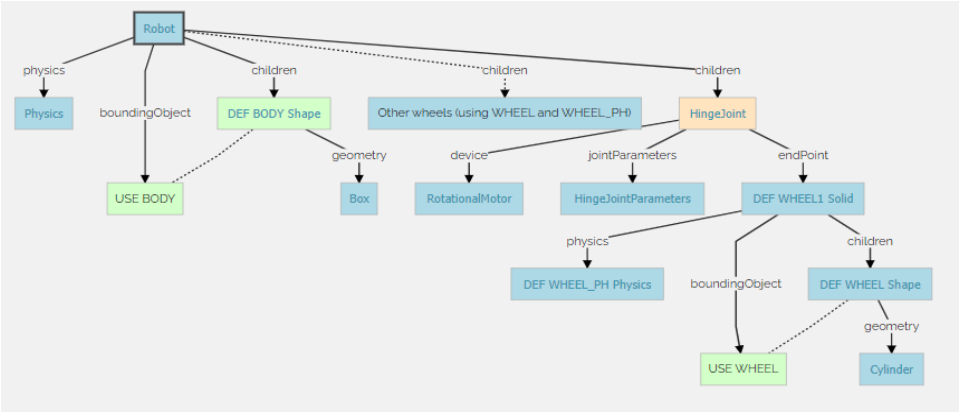
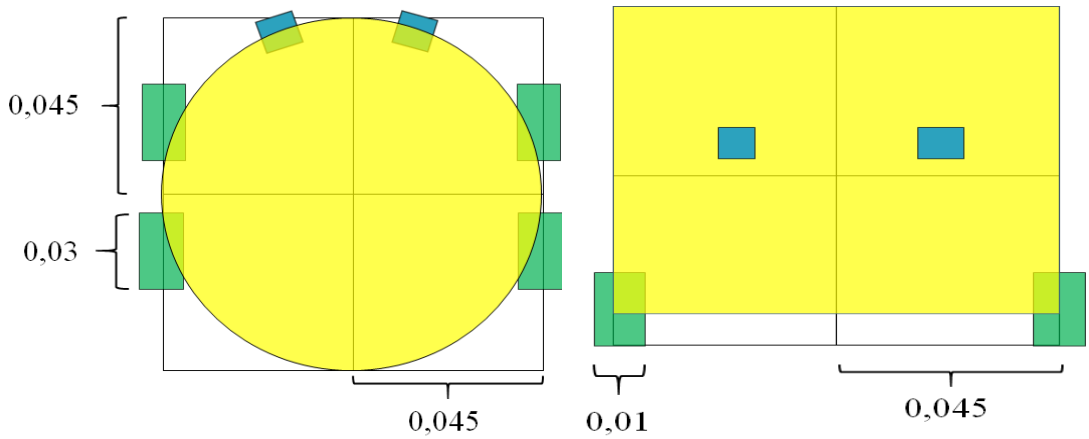


Рисунок Г.5 – Структура робототехнчного пристрою при подоланні перешкод



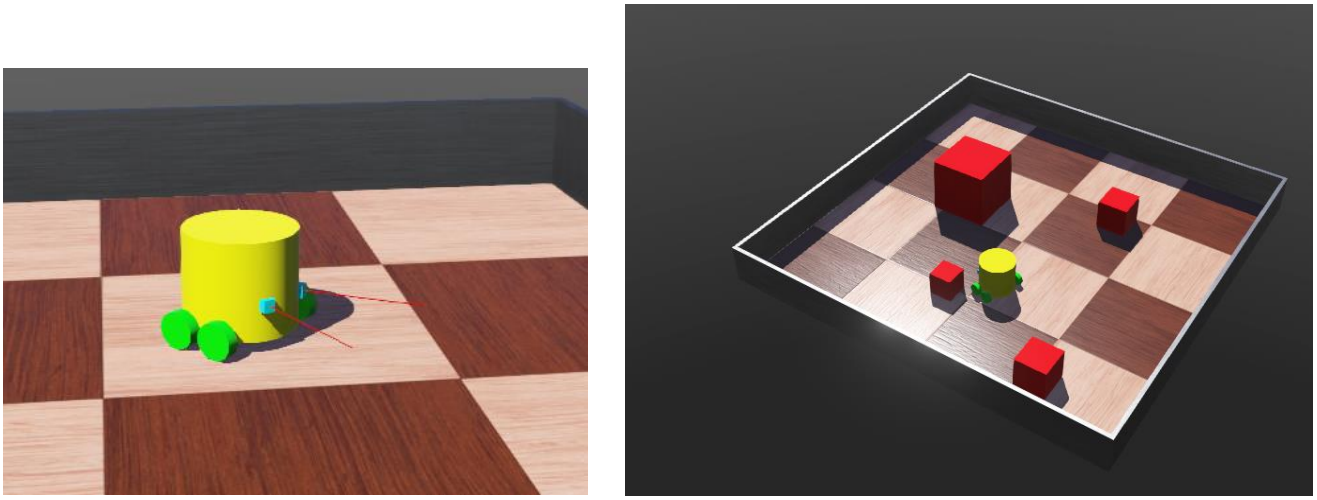


Рисунок Г.6 – Моделювання робототехнічного пристрою при подоланні перешкод в середовищі WeBots

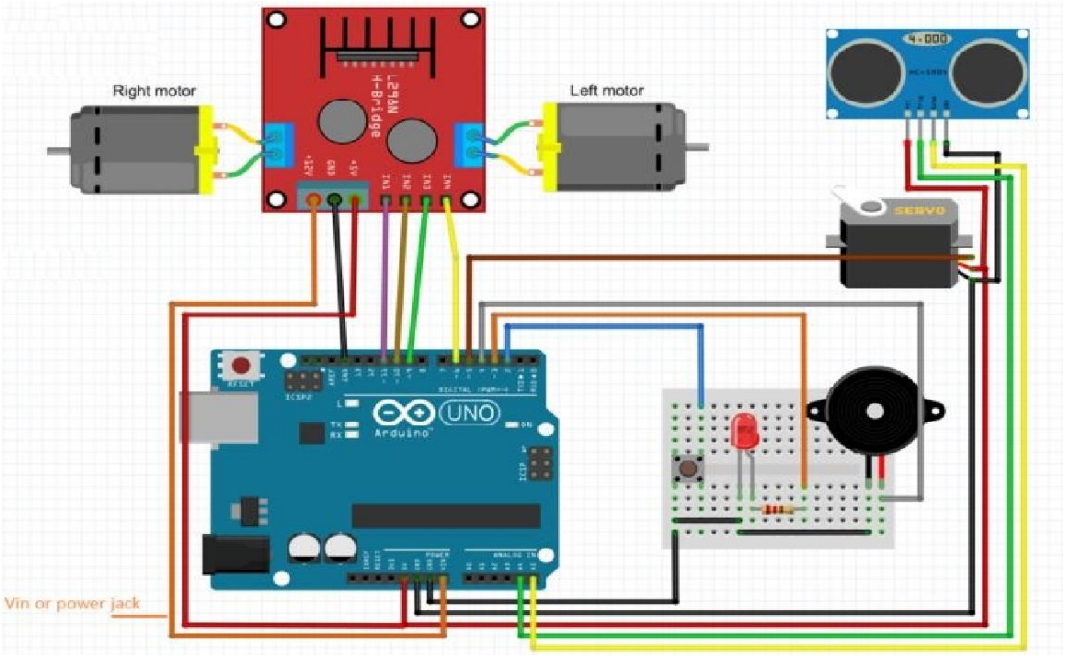


Рисунок Г.7 – Монтажна схема робототехнічного пристрою при подоланні перешкод

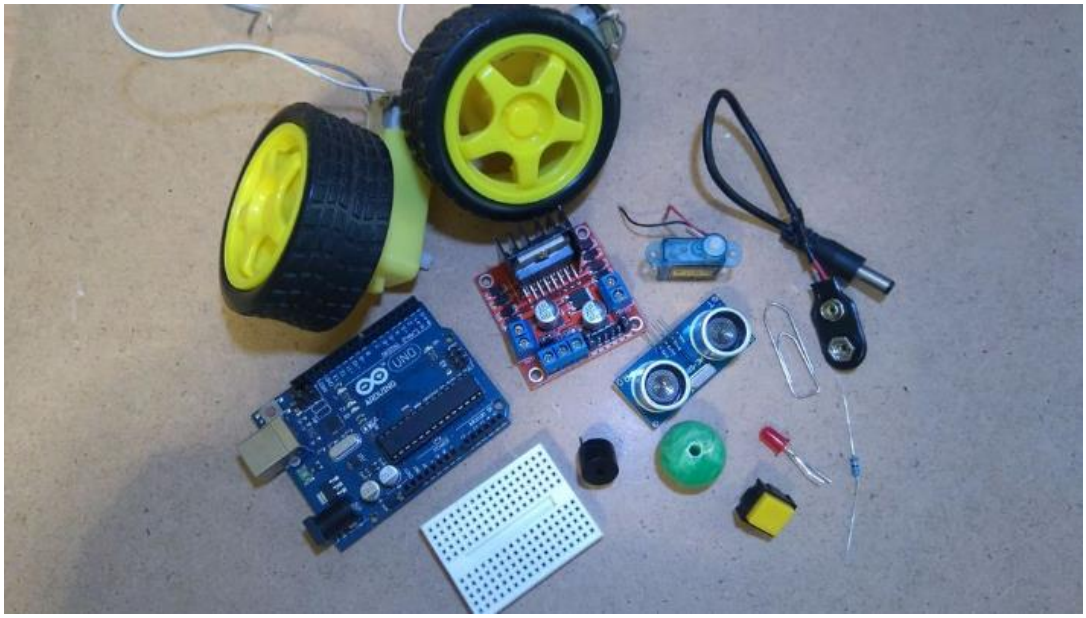


Рисунок Г.8 – Апаратна частина робототехнічного пристрою при подоланні перешкод

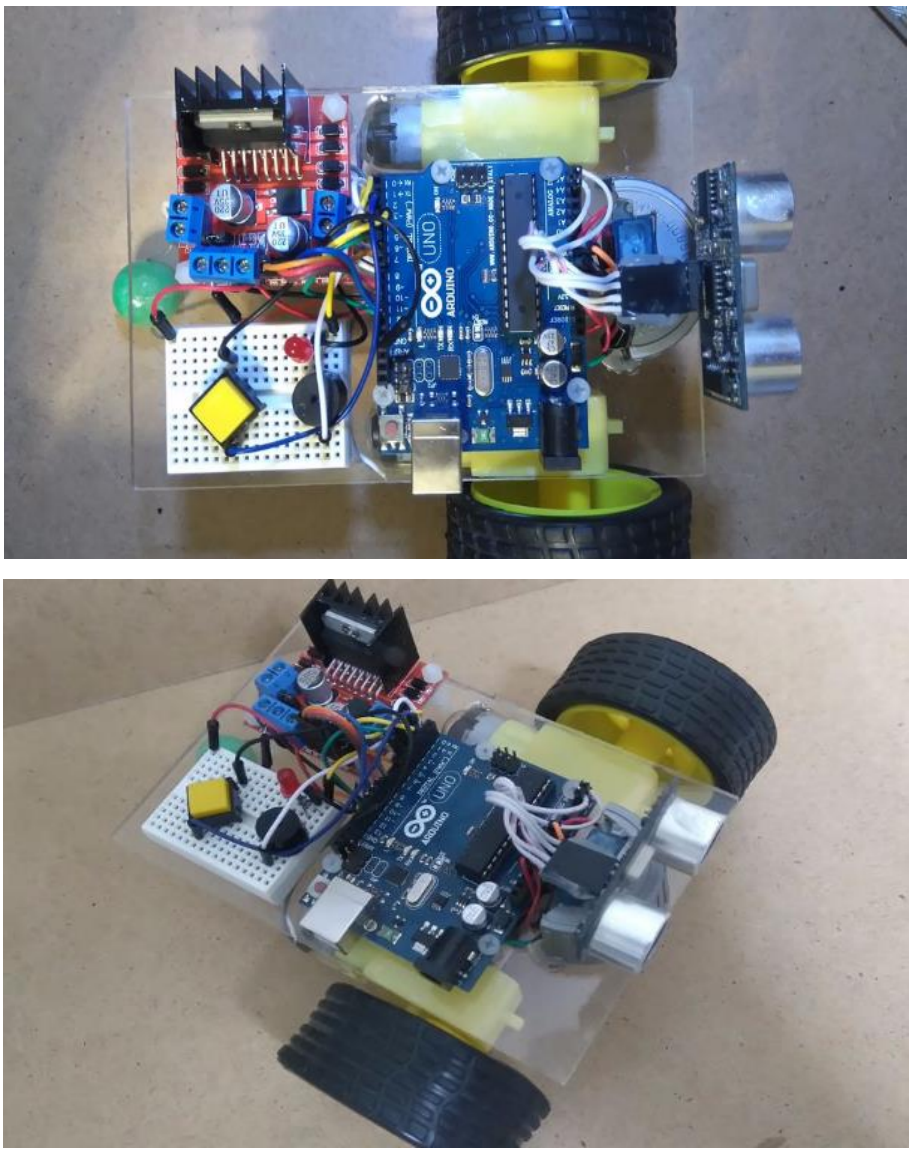


Рисунок Г.9 – Зовнішній вигляд робототехнічного пристрою при подоланні перешкод