

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра автоматизації та комп'ютерно-інтегрованих технологій та
робототехніки

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр

Освітній рівень

Роботизована мобільна платформа

Назва теми

КвРАКІТ.2021038.01.12 ПЗ

Галузь знань 15 «Автоматизація та приладобудування»

Шифр, назва

Спеціальність 151 «Автоматизація та комп'ютерно-інтегровані технології»

Шифр, назва

Освітня програма «Автоматизація та комп'ютерно-інтегровані технології»

Назва

Виконав:

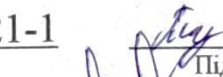
студент IV курсу, група АКІТ-21-1


Керівник

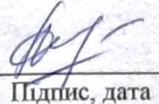
Нормоконтролер


До захисту допускаю:
зав. кафедри автоматизації
та комп'ютерно-інтегрованих
технологій та робототехніки

« 16 » червня 2025 р.


Підпис


Підпис, дата


Підпис, дата


Підпис, дата

Єлизавета ПИТЛИК
Ім'я, ПРІЗВИЩЕ

Денис МАКАРИШКІН
Ім'я, ПРІЗВИЩЕ

Людмила КОРЕЦЬКА
Ім'я, ПРІЗВИЩЕ

Валерій МАРТИНЮК
Ім'я, ПРІЗВИЩЕ

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій

Кафедра автоматизації, комп'ютерно-інтегрованих технологій та робототехніки

Рівень вищої освіти перший (бакалаврський)

Галузь знань 15 – Автоматизація та приладобудування

Спеціальність 151 – Автоматизація та комп'ютерно-інтегровані технології

Освітня програма Автоматизація та комп'ютерно-інтегровані технології

ЗАТВЕРДЖУЮ

Завідувач кафедри АКІТтаР

Валерій МАРТИНЮК

07 лютого 2025р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Питлик Єлизаветі Аркадіївні

Прізвище, ім'я, по батькові студента

1 Тема роботи Роботизована мобільна платформа

Керівник роботи канд. техн. наук., доцент Макаришкін Денис Анатолійович

Прізвище, ім'я, по батькові, науковий ступінь, учене звання

Затверджено наказом ректора університету від 07.02.2025 р. №23

2 Строк подання студентом роботи на кафедру 02.06.2025р.

3 Вихідні дані до роботи Завдання на кваліфікаційну роботу

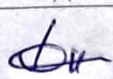
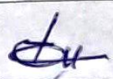

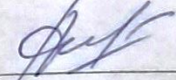
4 Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Вступ. Огляд методів розв'язання поставленої задачі. Розробка схемотехнічних рішень. Розробка алгоритму роботи програмного забезпечення. Висновки

5 Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

презентаційні матеріали (слайди)

6 Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Антиплагіат	Федула М.В., доцент кафедри АКІТтаР		
Нормоконтроль	Корецька Л.О., доцент кафедри АКІТтаР		

7 Дата видачі завдання 07 лютого 2025р.

КАЛЕНДАРНИЙ ПЛАН

Назва розділу кваліфікаційної роботи	Строк виконання	Примітка
1 Вибір та затвердження теми кваліфікаційної роботи; розробка завдання на кваліфікаційну роботу; складання календарного графіка виконання кваліфікаційної роботи	01.03.2025	Виконано
2 Вивчення предметної області, в якій планується використання системи автоматизації; аналіз вимог до системи автоматизації	15.03.2025	Виконано
3 Проектування та розробка загальної архітектури і структури системи автоматизації, інтерфейсу користувача; вибір засобів реалізації системи автоматизації	29.03.2025	Виконано
4 Програмна реалізація та тестування системи автоматизації	12.04.2025	Виконано
5 Написання тексту пояснювальної записки та розробка графічних матеріалів	19.04.2025	Виконано
6 Остаточне коригування кваліфікаційної роботи з урахуванням зауважень керівника; оформлення кваліфікаційної роботи як документа відповідно до вимог	11.04.2025	Виконано

Студент


Підпис

Єлизавета ПИТЛИК
Ім'я, ПІРІЗВИЩЕ

Керівник кваліфікаційної роботи


Підпис

Денис МАКАРИШКІН
Ім'я, ПІРІЗВИЩЕ

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Роботизована мобільна платформа».

Автор роботи: Питлик Єлизавета Аркадіївна

Керівник роботи: Макаришкін Денис Анатолійович

Пояснювальна записка: 63 с., 32 рис., 5 табл., 2 дод., 40 джерел.

Графічна частина: 8 презентаційних слайдів.

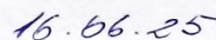
СИСТЕМИ КЕРУВАННЯ, ПИЛОСОС, МІКРОКОНТРОЛЕРНА СИСТЕМА, БЛОК-СХЕМА АЛГОРИТМУ.

Мета роботи створити робота-прибиральника з кількома варіантами автономного пересування, зчитуючи інформацію з датчиків про перешкоди перед ним.

Рух здійснюється за допомогою відповідного моторного приводу, який керує двома двигунами та відповідною парою коліс. Водій отримує сигнали від платформи Arduino Nano. Плата завантажена програмою, яка видає команди всім елементам конструкції. Написані спеціальні програмні алгоритми, які керують відключенням вентиляторів через заданий час. Робот живиться від підключеної батареї. Щоб запустити пилосос достатньо натиснути кнопку.







Підпис студента



Дата

ЗМІСТ

ВСТУП	3
1 СТАН ПИТАННЯ	5
1.1 Аналіз вихідних даних і відомих рішень	5
1.2 Висновки до першого розділу	14
2 ПРОЕКТНИЙ РОЗДІЛ	15
2.1 Розробка структурної і електричної принциповою схем.....	15
2.2 Вибір необхідних елементів	16
2.3 Висновки до другого розділу	36
3 ПРОГРАМНА ЧАСТИНА	37
3.1 Розробка алгоритму програми.....	37
3.2 Розробка керуючої програми.....	47
3.3 Конструкторсько-експериментальний розділ.....	48
3.4 Висновки до третього розділу	54
ВИСНОВКИ.....	56
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	58
Додаток А Програмний код	64

КвРАКІТ.2021038.01.12 ПЗ								
Зм.	Лист	№ докум.	Підпис	Дата		Літ.	Лист	Листів
Розроб.		Питлик Є.А.		16.06.25	Роботизована мобільна платформа Пояснювальна записка		2	
Перевір.		Макаришкін Д.А.		16.06.25				
Н. Контр.		Корецька Л.О.		16.06.25				
Затв.		Мартинюк В.В.		16.06.25				
ХНУ, АКІТ-21-1								

ВСТУП

Об'єктом дослідження даної роботи є мобільний робот-прибиральник — пристрій, який автоматизує процес збирання пилу та дрібного сміття в житлових приміщеннях. У сучасному світі впровадження новітніх технологій активно змінює наше повсякденне життя, забезпечуючи комфорт, зручність та економію часу. Зокрема, автоматизовані системи прибирання набувають все більшої популярності серед користувачів, оскільки здатні ефективно виконувати рутинні завдання з обслуговування житлових і робочих приміщень.

Сучасні роботи-прибиральники можуть не лише видаляти пил, крихти, шерсть і бруд з підлоги, але й виконувати додаткові функції, такі як зволоження повітря, ароматизація приміщення або дезінфекція поверхонь. Проте навіть базові моделі, здатні лише всмоктувати пил, дрібні частинки бруду та волосся, залишаються актуальними, особливо для власників домашніх тварин. Домашні улюбленці, такі як коти чи собаки, є джерелом постійного накопичення шерсті, тому попит на автономні прилади для прибирання серед населення зростає.

Процес прибирання традиційними методами часто вимагає значних фізичних зусиль, часу та концентрації уваги. У випадках, коли потрібно швидко впорядкувати приміщення, технічні засоби, зокрема недорогі моделі роботів, стають справжніми помічниками в побуті. Зниження вартості електронних компонентів та зростання доступності комплектуючих у вільному продажу зробило можливим створення подібних пристроїв навіть у домашніх умовах або в навчальному процесі.

У даній роботі реалізовано проєкт робота-прибиральника на базі апаратної платформи Arduino Nano, яка виконує функцію головного керувального модуля. Arduino Nano — це компактна, енергоефективна плата з мікроконтролером, яка широко використовується в аматорській та навчальній

електроніці завдяки простоті програмування, великій спільноті користувачів та великому вибору готових бібліотек і прикладів. За допомогою цієї плати надсилаються керувальні сигнали до виконавчих елементів робота, забезпечуючи його пересування, зміну напрямку руху, реагування на перешкоди.

Використання Arduino дає широкі можливості для модернізації та розширення функціоналу проєкту. У багатьох випадках такі проєкти слугують основою для навчання школярів і студентів основам програмування, електроніки, мехатроніки. Часто подібні розробки можна побачити на наукових виставках, STEM-форумах або хакатонах, де їх представляють діти, підлітки та молодь. Вони демонструють не лише технічні вміння, але й інженерне мислення, здатність до вирішення практичних задач побутового характеру. [1-5]

Таким чином, розробка робота-прибиральника на базі Arduino Nano має не лише практичне, а й навчальне значення, а також дозволяє долучити молодь до світу інженерних технологій.

Метою кваліфікаційної роботи є створення доступної моделі робота-прибиральника, щоб продемонструвати його можливості.

Для цього були поставлені та виконані наступні завдання:

- виконати огляд джерел та електронних ресурсів за темою кваліфікаційної роботи.
- проаналізувати існуючі моделі роботів-прибиральників.
- розробити електричні та конструктивні схеми та обрати відповідні елементи.
- сконструювати автономного робота-прибиральника та розробити програмний код для керування всією конструкцією.

1 СТАН ПИТАННЯ

1.1 Аналіз вихідних даних і відомих рішень

Робот-прибиральник – це пристрій для автоматичного прибирання приміщення. Технологія нагадує сферу, товщиною близько 10 см і діаметром 30 см. Існують також моделі, які мають форму квадрата, але із закругленими кутами. Робот-пилосос (менш відомий як робот-пилосос, робот-пилосос) — пилосос, оснащений штучним інтелектом і призначений для автоматичного прибирання приміщення. Побутові роботи та розумна техніка відносяться до категорії розумного будинку.

Робот-пилосос може прибирати відповідно до визначених сценаріїв або команд користувача. У процесі прибирання робот самостійно пересувається по призначеній поверхні, видаляючи з поверхні сміття. Коли робот зустрічає на своєму шляху перешкоду, він за спеціальним алгоритмом вирішує, як її подолати. Після завершення прибирання більшість роботів повертаються до свого зарядного пристрою самостійно.

З початку 21 століття багато компаній почали випуск «роботів-пилососів», таких як Electrolux Trilobite[en], Roomba[en], Robomaxx, Samsung Navibot, FloorBot та ін. Зазвичай пристрій являє собою диск діаметром 28-35 см і висотою 9-13 см. На передній панелі пристрою встановлено великий контактний датчик, а в центрі верхньої передньої панелі – інфрачервоний датчик. Залежно від моделі пристрій може бути оснащений інфрачервоним датчиком «віртуальної стіни». Моделі першого та другого покоління вимагали ручного введення квадратних метрів кімнати, але модель третього покоління цього не вимагає. Роботи-пилососи працюють від внутрішніх акумуляторів, тому їх потрібно регулярно заряджати. Новіші моделі можуть самостійно знаходити свій зарядний пристрій, спілкуючись із ним через інфрачервоний датчик. Час зарядки для пристроїв першого та другого покоління становить

					КвРАКІТ.2021038.01.12 ПЗ	5
		№ докум.	Підпис			

близько 12 годин, а для пристроїв третього покоління – близько 3 годин. Роботи-пилососи зазвичай невеликі та досить низькі, щоб поміститися під ліжку чи інші меблі. Коли робот-пилосос розуміє, що застряг, він перестає рухатися та починає подавати звукові сигнали, щоб допомогти власнику знайти його. [1-5]

Управління пристроєм

Робот-пилосос налаштовується за допомогою РК-дисплея і кнопок, розташованих на верхній панелі робота. Найчастіше вибирається одна з трьох програм очищення: звичайна, швидка і «точкова» - для площі до двох квадратних метрів (можуть бути відмінності в залежності від моделі). Сучасні роботи-пилососи дозволяють автоматизувати прибирання за допомогою планувальника завдань. Це може бути корисно для тих, хто хоче запланувати прибирання, коли їх немає вдома.

Деякі моделі роботів-пилососів використовують інфрачервоні пульти дистанційного керування для дистанційного регулювання пристрою, що корисно для людей з обмеженими фізичними можливостями.

Після прибирання власник повинен спорожнити пилозбірник робота-пилососа [1].

Описано кілька основних етапів роботи:

- збір даних: сканування кімнати лазером, щоб обчислити розмір і початкове положення меблів та інших перешкод;
- на основі отриманих даних складаються карти руху та формуються маршрути;
- увімкнення щіткової системи для збору сміття, засмоктування його в сміттєвий бункер і переміщення по визначеному маршруту;
- робот приводиться в рух електродвигунами і маленькими колесами.

Управляється кнопкою у верхній частині корпусу. Оснащений батареєю, ємність батареї визначає час її безперервної роботи.

Роботам потрібна навігація, щоб зрозуміти, де вони знаходяться. Якість навігації визначає якість її поведінки та виконання під час складання. Існує три найпоширеніші системи навігації:

Є чотири способи позиціонувати себе в просторі:

– використання далекоміра (ультразвукового), наприклад, робот-пилосос Electrolux Trilobite [2];

– використання контактних датчиків - датчики на бампері дозволяють реєструвати удари об стіни та меблі;

– використання «віртуальної стіни», простір для прибирання можна обмежити, як правило, за допомогою інфрачервоного передавача в «стіні» та інфрачервоного приймача;

– використання інфрачервоних датчиків у нижній частині пристрою може запобігти його падінню зі сходів, наприклад робота-пилососа Roomba.

Можливі аксесуари:

– спеціальна щітка для збору вовни - спеціальна щітка, яка більше підходить для збору шерсті домашніх тварин;

– дистанційне керування – використовується для дистанційного керування роботом-пилососом;

– планувальник – можливість запрограмувати робот-пилосос на роботу за розкладом;

– зарядний модуль з інфрачервоним маячком – дозволяє роботу-пилососу автоматично повертатися до зарядного пристрою;

– віртуальні стіни — це спеціальні датчики, що використовуються для обмеження площі прибирання;

– маяки – використовуються для поділу зон для послідовного очищення;

– Enhanced Power – акумуляторна батарея, яка дозволяє пристрою працювати більше 3 годин.

					КвРАКІТ.2021038.01.12 ПЗ	8
		№ докум.	Підпис			

Ці дії тривають, доки процесор не отримає сигнал про низький рівень заряду батареї, а потім перейде до док-станції для підзарядки.

База оснащена індукційною котушкою для бездротової зарядки акумулятора. Щоб роботизований пристрій швидко знаходив місце, він містить спеціальний інфрачервоний маяк. Робот рухається до цього маяка.

За типом завдань клінери поділяються на:

1. Для хімчистки. Обертіві бічні щітки захоплюють сміття і направляють його до головної центральної турбіни. Допоміжна турбощітка використовує потужний потік повітря для переміщення пилу та сміття в пилозбірник. Коли контейнер для пилу заповнений, пристрій сповіщає власника, що пора спорожнити контейнер і повернути його на базу.

2. Для вологого прибирання. Для чищення використовується серветка з мікрофібри. Основа так званої швабри зі змінною ганчіркою встановлюється в паз пристрою. Поставляється зі спеціальною секцією для води та миючого засобу. Змочується мікрофібру водою або іншою рідиною та зробіть вологе прибирання підлоги. У деяких моделях використовується засіб для ретельного очищення. Тобто робот покриває поверхню продуктом і натирає забруднене місце. Потім за допомогою всмоктування зберіть всю вологу в контейнер і видаліть залишки силіконовою лопаткою.

Перевага робота-прибиральника в тому, що він не потребує ручного керування прибиранням. Щоб підтримувати його в чистоті, не потрібна ручна праця. Це підходящий варіант для людей з обмеженими можливостями. Час збору можна організувати так, щоб не заважати іншим заходам. Пристрій має досить хорошу звукоізоляцію, і низький рівень шуму не буде заважати, особливо якщо вдома є маленькі діти.

Недоліком цього пристрою є те, що він вимагає регулярного обслуговування, наприклад очищення фільтрів, пилозбірника, корпусу та щіток. Крім того, бактерії можуть рости всередині пристрою, якщо ви чистите його вологою тканиною. Пилосос все ще не може видалити липкі або великі

шматки бруду, наприклад їжу. Замість цього він просто розмазав його по поверхні. Через конструкцію будинку кімнати та кути будинку можуть забруднитися. [6-8]

При аналізі всіх відомих виробників було виявлено кілька дорогих і популярних на даний момент моделей. Наприклад, номер моделі «Roborock E-352-00».

Ємність акумулятора цієї моделі Xiaomi становить 5200 мАг. Його цикл збору – до 150 метрів квадратних дорівнює 2 години. Через програму можна віддавати команди та контролювати її дії. Обладнання може виконувати як сухе, так і вологе прибирання. Зовнішній вигляд робота-прибиральника «Roborock E-352-00 » наведено на рисунку 1.4.



Рисунок 1.4 - Робот-прибиральник «Roborock E-352-00 »

Зовнішній вигляд робота-прибиральника модель «iClebo O5» наведено на рисунку 1.5.



Рисунок 1.5 - Модель «iClebo O5»

Корейський робот-пилосос може протирати підлогу вологою ганчіркою в ходу, але навіть за таку ціну він не має вбудованої функції миючого засобу. Ємність акумулятора становить 5200 мАг, а ємність для води становить 600 мл. Одного заряду вистачає на 2 години використання, а потужності вистачає на очищення 200 метрів квадратних за цикл. Рекомендується подавати команди через смартфон. Вигляд робота-пилососа «iRobot Roomba i7» наведено на рисунку 1.6.



Рисунок 1.6 - Робот-пилосос «iRobot Roomba i7»

Він має власну навігаційну систему iAdapt0, яка використовує vSLAM, яка допомагає безпомилково орієнтуватися на місцевості та створювати карти на основі власних результатів датчиків. Ці щітки легко видаляють шерсть домашніх тварин і килимів, що є великим плюсом. Голосове управління через Google Assistant, або управління через програмне забезпечення. Його мінус – відсутність вологого прибирання.

Проаналізувавши моделі, які пропонує людству сучасний ринок, встановлено, що цінова категорія таких пристроїв дуже висока. Вологе прибирання. Більшість пристроїв не мають інфрачервоних датчиків. У зворотному напрямку. [9-15]

1.2 Висновки до першого розділу

У першому розділі проведено аналіз задач що вирішують роботи-прибиральники. Визначено межі їх застосування. Показано основні принципи їх будови. Показано основні шляхи по проектуванню аналогічних приладів.

Проведено дослідження відомих моделей роботів-пилососів. Наведено їх технічні параметри та особливості їх будови.

					КвРАКІТ.2021038.01.12 ПЗ	
		№ докум.	Підпис			14

2 ПРОЕКТНИЙ РОЗДІЛ

2.1 Розробка структурної і електричної принциповою схем

Побудова принципів електричних і конструктивних схем.

Параметри, яким задовольняє модель автономного робота-прибиральника:

- живлення: 5-12В;
- використання датчики для переміщення за заданим алгоритмом;
- автоматично виконання функції всмоктування сміття.

Структурна діаграма та схематична діаграма встановленої моделі показані на рисунку 2.1 та рисунку 2.2 відповідно.

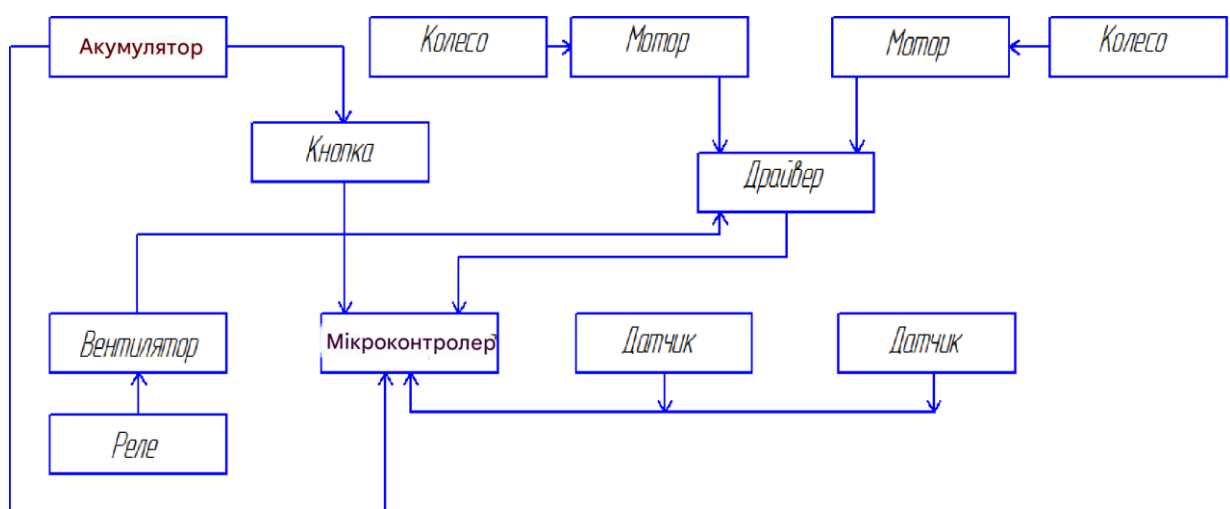


Рисунок 2.1 - Структурна схема робота

На представленій структурній схемі показано усі взаємозв'язки між структурними елементами. Показано їх логіка взаємодії для досягнення поставленої мети по керуванню роботом-пилососом. На основі даної структурної схеми розроблено схему принципову.

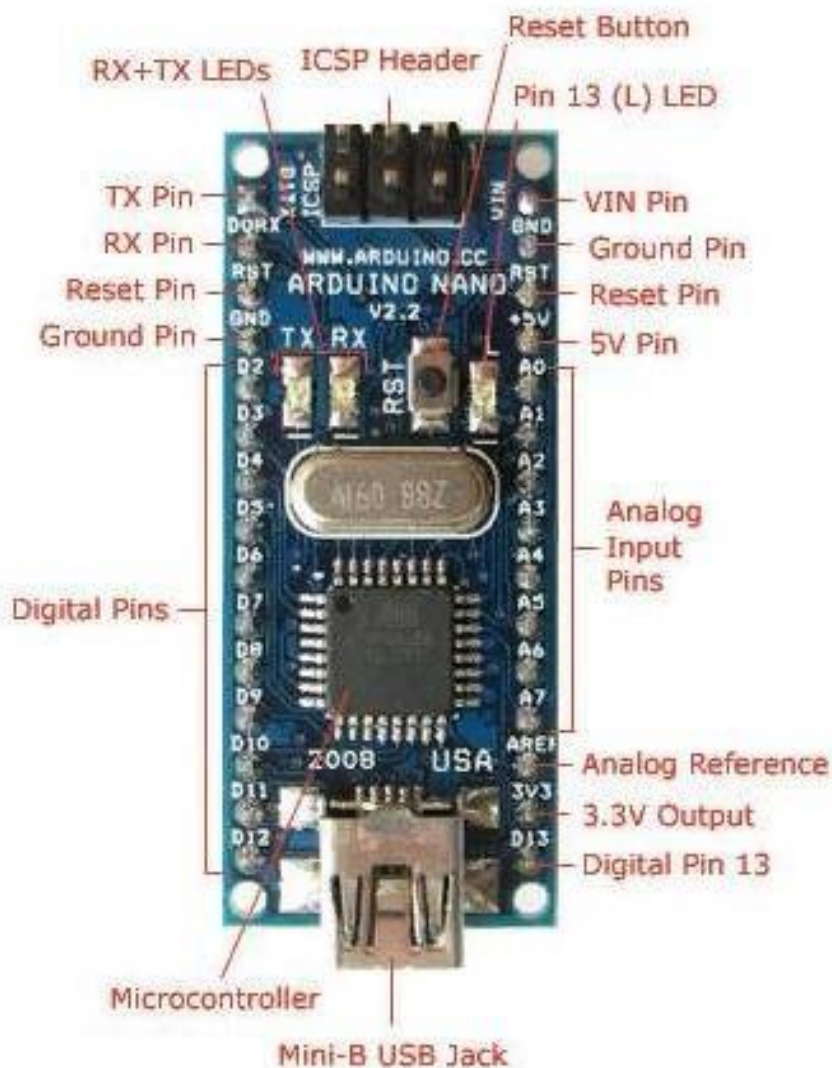


Рисунок 2.3 - Платформа Arduino Nano

Платформа має роз'єм mini USB для підключення до комп'ютера. Верхній шар має кнопку скидання та чотири індикатори. Така велика кількість аналогових входів дозволяє підключати більше аналогових датчиків, що є плюсом цього проекту.

Кожен із чотирнадцяти цифрових контактів може бути налаштований як вхід (INPUT) або вихід (OUTPUT) і працює при напрузі 5 В.

Є два способи живлення Arduino Nano для належної роботи. Підключіть до джерела живлення 5 В або комп'ютера за допомогою відповідного кабелю

USB. В якості альтернативи використовуйте зовнішнє стабільне джерело, напруга якого повинна коливатися від 6 до 20 В. Напруга подається на вхід VIN.

(RX)-0 і (TX)-1 є виходами послідовної шини, які використовуються для отримання та передачі інформаційних даних TTL відповідно. Ці контакти підключаються до певних контактів на мікросхемі послідовної шини USB-TTL CH340G.

Існує налаштування штифта для зупинки на нижньому передньому або задньому фронті або змінному значенні, яке називається зовнішнім другим і третім перериваннями.

Для ШІМ використовуються наступні висновки: 3, 5, 6, 9, 10, 11. Кожен з них підходить для «широко-імпульсної модуляції» для управління живленням навантаження з восьмибітним дозволом. 10 - (SS), 11 - (MOSI), 12 - (MISO), 13 - (SCK) - Виводи, що використовуються для SPI. Саме з їх допомогою відбувається зв'язок SPI, яка підтримується апаратним забезпеченням, але не включена в мову Arduino.

Вбудований світлодіод - LED 13 підключається до тринадцятого цифрового виводу і при подачі на нього високого потенціалу буде показувати працездатність плати.

AREF – забезпечує опорну напругу лише для входу АЦП. RESET відповідає за надсилання сигналу низького рівня на контакт, що викликає перезавантаження мікроконтролера платформи.

Переваги оригінальної дошки є якісна збірка та те, що немає проблем із сумісністю чи драйверами. Недоліком є висока ціна. [16-17]

Перевагами плат вироблених у Китаї є низька ціна.

Недоліки:

– зібрані з дешевих піратських компонентів, часто виникають проблеми з підключенням і приводом;

– бувають випадки, коли чіп блокується операційною системою на апаратному рівні (для запобігання піратству). (рисунки 2.4 та 2.5)

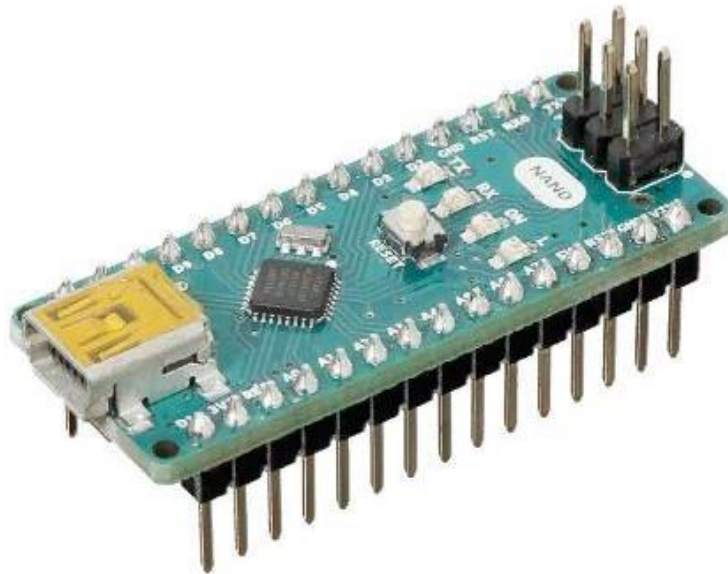


Рисунок 2.4 - Оригінальна плата Arduino Nano

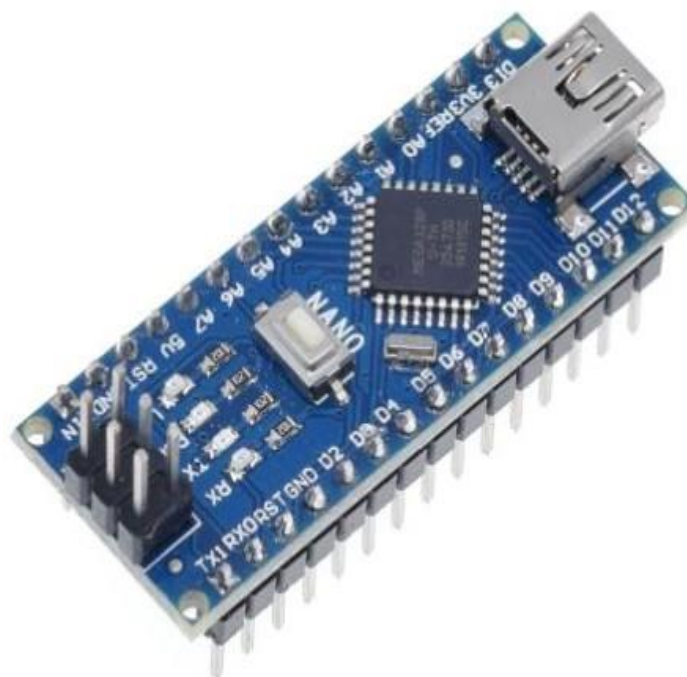


Рисунок 2.5 - Плата Arduino Nano

Технічні характеристики диктують, що модель робота-прибиральника буде використовувати необхідні елементи для пересування. Для того, щоб робот міг рухатися і виконувати маневри повороту, потрібні мотор-водії. Розберемо моделі приводів, які зараз випускаються.

Коли для якогось проекту потрібен двигун постійного струму або кроковий двигун, драйвери часто використовуються з Arduino. Мобільні роботи можуть бути сконструйовані на базі Arduino за допомогою драйверів на різних мікросхемах або за допомогою щитів двигунів.

Розберемо деякі популярні моделі приводів на основі різних мікросхем, які можна знайти.

Arduino має обмеження струму 800 мА, а кожен контакт має обмеження струму 40 мА. Не рекомендується підключати навіть найменші двигуни постійного струму безпосередньо до Arduino Nano. Будь-який двигун може генерувати пікові стрибки струму, що перевищують ліміт під час запуску або зупинки. Потрібно використовувати правильну схему підключення, таку як Н-подібний міст, щоб була можливість змінити напрямок руху двигуна.

Подовжувач козуха двигуна. Щоб забезпечити необхідну роботу різних двигунів постійного струму, з Arduino все частіше використовується плата розширення під назвою «Motor Shield». Такі мікросхеми, як L293D і L298N, є платами для екранування двигуна. Вони здатні керувати двома двигунами одночасно.

Цей щит забезпечує вибір джерел напруги, оскільки він здатний отримувати напругу як від Arduino, так і від зовнішніх джерел. Про працездатність пристрою свідчить світлодіод, розташований на платі.

Основа роботи Н-містка. Абсолютно кожен драйвер працює за принципом Н-bridge. Міст являє собою електронну схему, що складається з чотирьох перемикачів і навантаження. З назви схема нагадує букву Н. (рисунок 2.6)

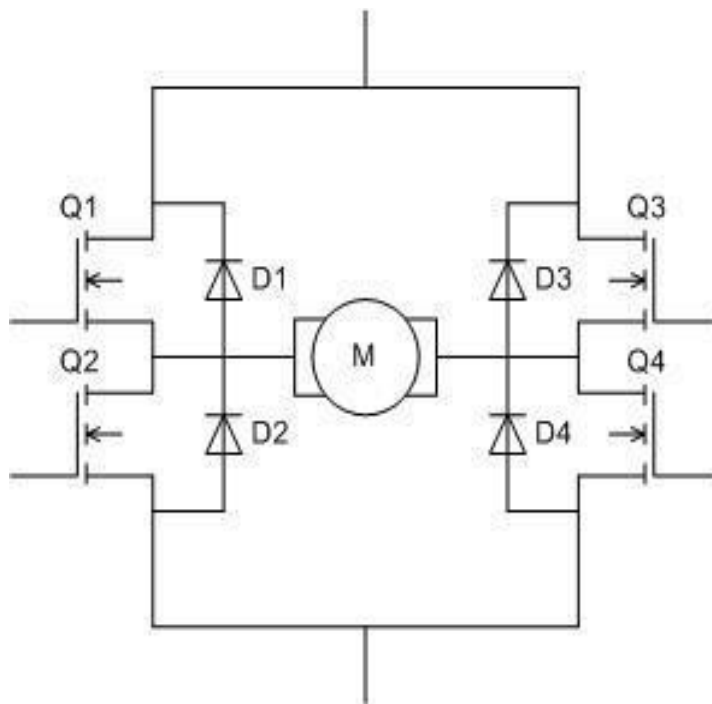


Рисунок 2.6 - Схематичний вигляд Н – мосту

Q1...Q4 — польові транзистори, біполярні транзистори або IGBT транзистори. Останній використовується в електромережах високої напруги. Біполярні транзистори рідко використовуються на практиці, вони можуть з'явитися в схемах малої потужності. Для великих струмів використовують польові транзистори з ізольованими затворами. Щоб уникнути короткого замикання джерела, обидва вимикачі не повинні бути замкнені одночасно. D1...D4 — струмообмежувальні діоди, зазвичай використовуються діоди Шотткі. Змінюючи стан перемикачів Н-мосту, можна регулювати напрямок руху двигуна та застосовувати гальмування. [18-19]

Драйвер на базі мікросхеми L298N. Модуль використовується для управління двома двигунами з напругою 5-35 В. Кожен двигун має навантаження 2А, але якщо підключити їх паралельно, можна досягти значення струму 4А. (рисунок 2.7)

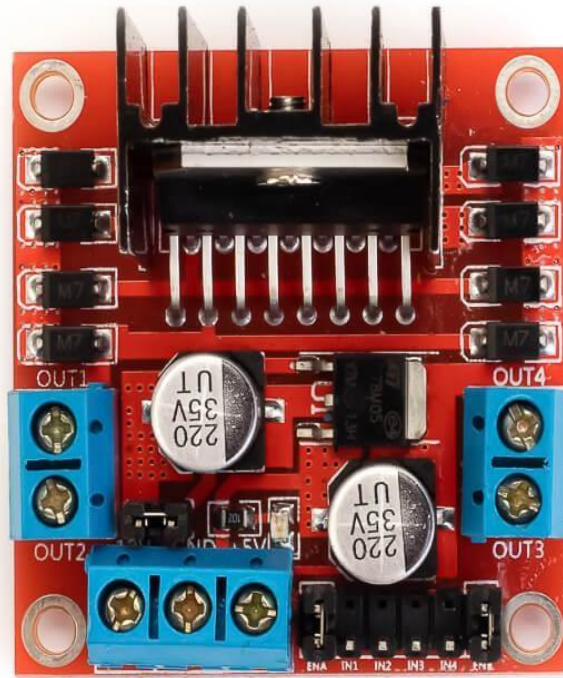


Рисунок 2.7 – Модуль на мікросхемі L298N

Позначення виводів мікросхеми L298N показано у таблиці 2.1.

Таблиця 2.1 - Позначення виводів мікросхеми L298N

Підключення зовнішнього живлення	VCC
Вивід на «землю»	GND
Управління швидкістю обертання моторів	IN1, IN2, IN3, IN4
Виходи з першого двигуна	OUT1, OUT2
Виходи зі другого двигуна	OUT3, OUT4
Перемикання живлення схеми від зовнішнього джерела	S1
Роздільна управління каналами у двох режимах: активний і пасивний	ENABLE A, B

Багато людей стикаються з проблемою обертання мотора в різні боки. Щоб цього уникнути, необхідно перевіряти полярність.

Драйвер на основі L293D. Цей драйвер L293D дуже простий у використанні з двигунами. Для керування двома двигунами передбачено два Н-містки. Мікросхема працює при напрузі до 36В і струмі до 600мА. Ця мікросхема подає на двигун максимальний струм 1,2А.

Схема містить 16 виходів. На відміну від мікросхеми L298N, підключений двигун працюватиме від 5 В, що надається платою Arduino, тому немає необхідності забезпечувати додаткове джерело живлення для L293D. (рисунок 2.8)

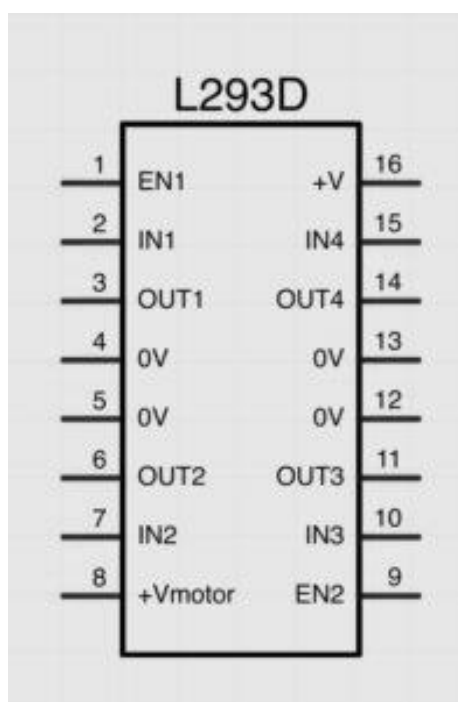


Рисунок 2.8 - Мікросхема L293D

Функціональне призначення висновків мікросхеми L293D показано у таблиці 2.2.

Таблиця 2.2 - Функціональне призначення висновків мікросхеми L293D

+V	живлення від 5В
+Vmotor	живлення для мотору до 36 В
0V	GND
En1, En2	включають і вимикають Н- мости
In1, In2	керують першим Н- мостом
Out1, Out2	підключення першого Н- моста
In3, In4	керують другим Н- мостом
Out3, Out4	підключення другого Н- моста

Драйвери на чіп L9110S. Мікросхема не відрізняється від попередніх, також забезпечує обертання незалежного колекторного двигуна. Може працювати з чотирипровідними або двофазними двигунами. (рисунок 2.9)

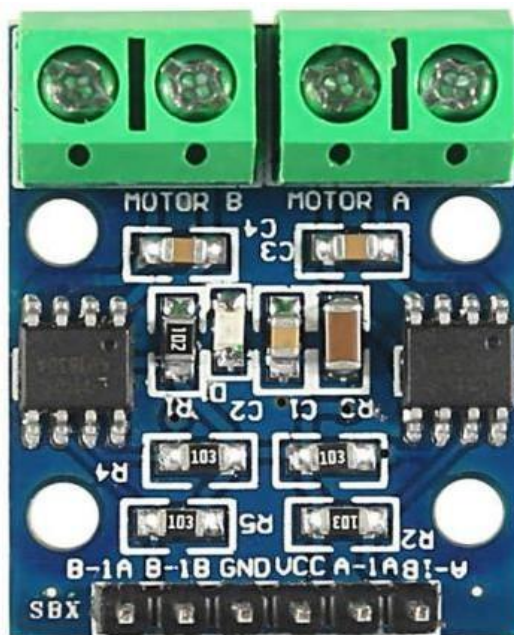


Рисунок 2.9 - Драйвер L9110S

Для легкого підключення до Arduino можна використовувати Motor Shield. Драйвер має два інтерфейси для підключення джерела живлення до

мікроконтролера і керованого пристрою. Інтерфейс для підключення керованого пристрою складається з двох елементів: двигуна А і двигуна В. Для подачі сигналів управління передбачено шість контактів:

- VCC – 5В;
- GND – контакт заземлення;
- А - IA - вихід 1 двигуна А;
- А - IB - Вихід 2 двигуна А;
- В - IA - Вихід двигуна 1;
- В - IB - Потужність двигуна 2.

Драйвер може керувати напрямком обертання кожного двигуна окремо, а також контролювати швидкість обертання двигунів за допомогою ШІМ. Дві мікросхеми L9110S, встановлені на платі, працюють за принципом Н-моста для зміни полярності та живлення двигунів.

Порівняння модулів. Максимальний струм, який забезпечує модуль на мікросхемі L293D, становить 1,2 А, але максимальний струм, який може забезпечити мікросхема L298N, становить 4 А. Крім того, L293D нагрівається досить швидко і набагато менш ефективний. Але L293D - це більш звичайна материнська плата, дуже дешева, близько 150-300 рублів, тоді як L298N не так поширена в сфері робототехніки, її важко знайти для покупки, ціна близько 100 рублів. В основному всі проекти Arduino реалізуються за допомогою чіпа L298N. Розглянуті модулі практично однакові; для мого робота-прибиральника я буду використовувати драйвер L298N.

Кроки для підключення драйвера до Arduino:

- перевірка полярності;
- підключення окреме джерело живлення 12В;
- підключення клеми GND;
- підключення клем IN1-D7, IN2-D6, IN3-D5, IN4-D4;
- забезпечення сигнали HIGH і LOW для початку і кінця обертання;

Мотор-редуктор являє собою мініатюрний потужний колекторний двигун розміром 130 з графітовими щітками і пластиковим редуктором. (рисунок 2.11) Підходить для різних роботів і електричних моделей. [20]



Рисунок 2.11 - Мотор-редуктор Gekko PR130-048

Специфікація мотора Gekko PR130-048 показана у таблиці 2.3.

Таблиця 2.3 - Специфікація мотора Gekko PR130-048

Напруга живлення	6В
Передавальне ставлення	48:1
Струм без навантаження	70 мА
Струм блокування	800 мА
Крутний момент	1,1 кг · см
Діаметр вала	5,4 мм
Розмір (ДхШхВ)	64,2 x 22,5 x 22,5 мм
Швидкість без навантаження	230 об/хв

Пара пластикових коліс, призначених для моделі PR130, підійде до двигуна. Вони забезпечать маленькому роботу рівномірну поїздку, як у роботі.

Колеса мають силіконові шини діаметром 65 мм і шириною 26 мм. (рисунок 2.12) [21]

Робот-прибиральник буде орієнтуватися за допомогою датчиків, які зчитують відстань до перешкод. [22-23]

У якості датчика обрано ультразвуковий далекомір, що призначений для роботи з Arduino.



Рисунок 2.12 - Колеса Gekko 65x26

В онлайн-середовищі під час пошуку датчиків, придатних для роботи, з'являються пропозиції щодо ультразвукових далекомірів. Виявляється, це доступний прилад, який може зчитувати відстань до перешкод, з діапазоном дослідження від 2 до 400 см. Далекомір може повідомляти про близькість і відстань до об'єктів різного розміру. Найбільш поширеним пристроєм є модуль HC-SR04.

Далекомір HC-SR04 (рисунок 2.13) пропускає ультразвукові хвилі та приймає ультразвукові хвилі, відбиті від об'єкта, через деякий час для визначення відстані.

На вимірювання дальності впливають:

- температура;
- вологість повітря;
- якість комплектуючих самого модуля;
- об'єкт поруч.

Щоб далекомір працював належним чином, відносно нього об'єкти повинні лежати в межах конуса діаграми напрямку. (рисунок 2.14)

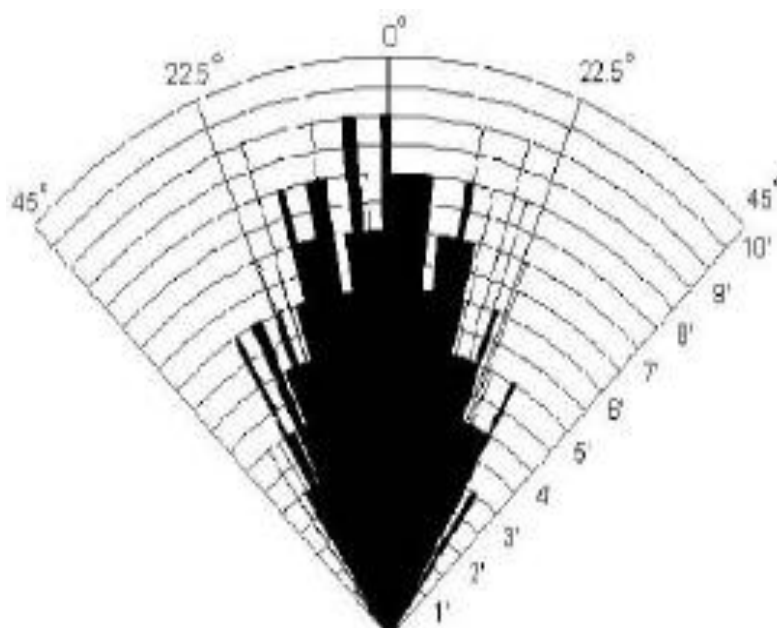


Рисунок 2.14 - Діаграма спрямованості HC - RS04

Щоб зменшити похибки вимірювань і похибки, значення усереднюються, тобто проводяться багаторазові вимірювання, спайки видаляються, а потім визначається середнє значення. За допомогою спеціальних датчиків визначається температура і фіксується поправочний коефіцієнт.

Ультразвуковий далекомір US-015

Ця модель практично нічим не відрізняється від попереднього датчика, але робота менш поширена. (рисунок 2.15)

Технічні параметри:

- напруга живлення – 5В;
- максимальний струм споживання - 20мА;
- струм в режимі очікування - 2,2 мА;
- рекомендований кут вимірювання 15°;
- максимальний кут вимірювання -30°;
- діапазон вимірювання відстані 2-700 см.

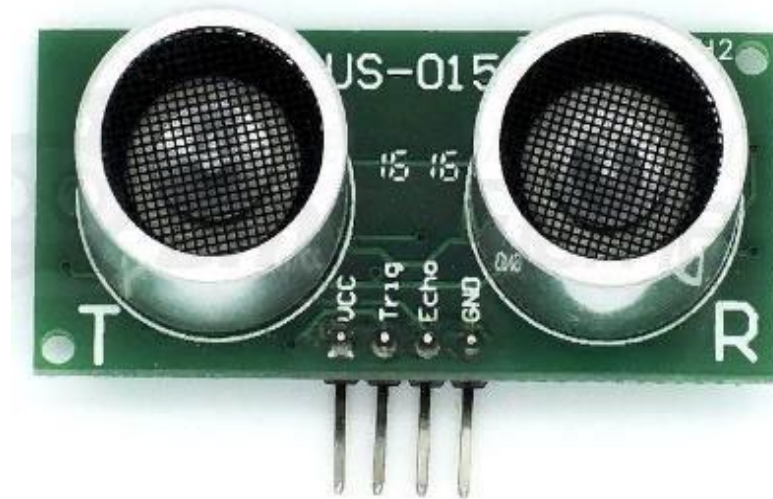


Рисунок 2.15 - Модуль US - 015

Після порівняльного аналізу робот-прибиральник виконує свої дії, отримуючи інформацію від двох ультразвукових далекомірів моделі HC-RS04. (рисунок 2.16)

Відсмоктування сміття буде досягнуто маленьким вентилятором у відеокарті комп'ютера. (рисунок 2.17)



Рисунок 2.20 - Зарядне пристрій Robiton LAC12-1000

Цей автоматичний зарядний пристрій підходить лише для свинцево-кислотних акумуляторів. Для зарядки використовується метод «постійний струм — постійна напруга». На першому етапі для зарядки акумулятора використовується постійний струм. Після досягнення необхідного значення напруги проводиться зарядка постійною напругою. [30]

Технічні характеристики:

- розмір зарядки: термінал типу T2/F2;
- акумуляторна електрохімічна система: свинцево-кислотний акумулятор;
- напруга акумулятора: 12В;
- струм і час зарядки: швидка зарядка (1,5 год-6 год);
- розмір і підключення до мережі: Пряме підключення до мережі (дроти не потрібні);
- спосіб зарядки: автоматичне відключення;
- додаткова функція: крапельна зарядка.

2.3 Висновки до другого розділу

У проєкті було реалізовано структурну і принципову електричну схеми робота-прибиральника, що забезпечує повноцінну логіку взаємодії компонентів. Для основи керування пристроєм обрано Arduino Nano — компактну, доступну та функціональну платформу. Платформа Arduino дозволяє підключати значну кількість аналогових та цифрових пристроїв, що є перевагою для робототехнічних систем. Для керування двигунами було досліджено та порівняно декілька моделей драйверів: L298N, L293D, L9110S. У підсумку вибрано драйвер L298N, як найбільш потужний, зручний у використанні та відповідний для даного навантаження. Двигуни типу Gekko PR130-048 обрано за їх компактність, надійність та відповідність параметрам живлення і крутного моменту. Для зчитування відстані до перешкод було використано ультразвуковий датчик, сумісний з Arduino. Використання Н-моста дозволяє реалізувати зміну напрямку обертання двигунів і забезпечує стабільність роботи. Запропонована схема дозволяє автономну роботу пристрою, що включає орієнтацію в просторі та прибирання сміття. Обрані компоненти забезпечують оптимальне співвідношення ціна/якість для створення ефективного та економічного робота-прибиральника.

					КвРАКІТ.2021038.01.12 ПЗ	36
		№ докум.	Підпис			

Що розуміти під «оптимальним» для цих умов і шляхів, залежить від конкретного завдання. [32]

Відстеження назад

Замість того, щоб починати з початкової точки, підхід починається з точки призначення (якщо це можливо) і визначає, як і куди туди дістатися. Далі слід шукати не шлях, який веде прямо до вашої мети, а шлях до «середини».

Програмування з відкатом

Іноді доводиться мати справу з проблемою пошуку оптимального рішення, коли жоден відомий алгоритм не може бути застосований, щоб допомогти знайти оптимальне рішення, і останнім засобом є використання вичерпного пошуку.

Методом проб і помилок

Багато проблем неможливо розв'язати аналітично, тому їх потрібно вирішувати методом проб і помилок, тобто спробувати всі можливі варіанти та відкинути їх, коли вони не вдаються. Якщо побудова рішення є складним процесом, то під час роботи фактично будується дерево можливих алгоритмічних кроків, а потім – у разі невдачі – відсікається відповідна гілка дерева, поки не буде встановлено шлях до успіху. Перетинати гілки дерева та відступати, якщо це не вдається, є винагородним алгоритмом.

Альфа-бета-відсікання

Дуже проста умова може позбавити вас від необхідності розглядати велику частину типового дерева гри. Цикл в алгоритмі може ігнорувати деякі випадки, і зазвичай ігнорує досить багато проблем.

Загальні правила відсікання вузлів пов'язані з поняттями кінцевих і наближених значень вузлів. Кінцеве значення називається посиленням. Приблизним значенням є верхня межа значення вузла в режимі MIN і нижня межа значення вузла в режимі MAX. Нижче наведено правила розрахунку цих значень.

1. Якщо всі відгалуження вузла були розглянуті або відсічені, то його значення апроксимації стає кінцевим.

2. Якщо приблизним значенням вузла в режимі MAX є v_1 , а кінцевим значенням одного з його відгалужень є v_2 , тоді встановіть приблизне значення вузла як $\max(v_1, v_2)$. Якщо вузол знаходиться в режимі MIN — $\min(v_1, v_2)$.

3. Припустимо, що p є вузлом у режимі MAX і має батьківський вузол q , а наближення вузла дорівнюють v_1 і v_2 відповідно, і $v_1 < v_2$, тоді якщо p є вузлом у режимі MAX, а q знаходиться в режимі MIN і $v_2 < v_1$, тоді ми можемо скоротити всіх неврахованих відгалужень вузла p .

Відгалуження та зв'язок

Метод розгалуження та зв'язування — це ще один спосіб скоротити розгалуження в дереві рішень, щоб не враховувати всі гілки дерева. Загальний підхід тут полягає в тому, щоб стежити за межами вже знайдених і можливих рішень. Якщо в якийсь момент найкраще рішення, яке було знайдено, є кращим, ніж найкраще рішення в нижній гілці, усі шляхи вниз від цього вузла можна проігнорувати.

Блок-схема алгоритму роботи автономного робота-прибиральника наведена на рисунку 3.1.

Алгоритм являє собою покроковий циклічний процес. Спочатку оголосимо змінну ультразвукового датчика, за допомогою якої мікроконтролер буде видавати команди драйверу двигуна. Потім встановіть змінні для двигуна, таймер антиприлипання для датчиків (якщо вони перестають реагувати на відстані 3 метрів), а також змінні реле, що відповідають за включення і виключення вентилятора.

Потім призначаються порти всіх елементів і починається робочий цикл програми, подається напруга і реле включає вентилятор. Далекомір надсилає та приймає ультразвукові сигнали та визначає напрямок руху робота на основі відстані між роботом і перешкодою.

Після того, як робот рухається протягом 10 хвилин, реле вимикає вентилятор, посилає низький сигнал на привід, і рух припиняється. [32]

Опис алгоритму керування мобільним роботом

Дана блок-схема відображає алгоритм керування мобільним роботом, який орієнтується у просторі, використовуючи два ультразвукових далекоміри, розташованих спереду ($cm1$) та ззаду ($cm2$). Робот здатен рухатися вперед, назад, а також здійснювати об'їзд перешкод ліворуч або праворуч.

Ініціалізація системи починається з задання номерів пінів для ультразвукових датчиків ($trigPin1$, $trigPin2$, $echoPin1$, $echoPin2$), а також задання початкових швидкостей обертання коліс ($speed1_1$, $speed2_1$). Змінні $moving$ і $rotate$ використовуються для визначення стану робота (рух або обертання), а змінна $endtime$ визначає граничний час роботи.

Робот постійно вимірює відстані спереду і ззаду за допомогою ультразвукових сенсорів. На основі цих вимірювань приймається рішення щодо подальшого руху.

1. Якщо передній датчик ($cm1$) показує відстань понад 10.6 см, робот рухається вперед.
2. Якщо передня відстань більша за 40 см, і задня також більша за 40 см, виконується рух уперед.
3. Якщо попереду є перешкода ($cm1 < 40$), але зліва вільно ($cm2 \geq 30$), виконується об'їзд праворуч.
4. Якщо попереду перешкода ($cm1 < 30$) і зліва менше 40, здійснюється об'їзд ліворуч.
5. Якщо відстань спереду менше 10 см, а ззаду також менше 10, виконується поворот на місці.
6. Якщо відстань спереду менше 5 см і ззаду менше 5 см — глухий кут, виконується рух назад.

7. Після кожного циклу перевіряється, чи досягнуто кінцевого часу роботи. Якщо так — робот зупиняється: вимикається вентилятор (який, імовірно, охолоджує драйвер двигунів), подається сигнал завершення роботи.

Такий підхід дозволяє роботу ефективно орієнтуватися у статичному середовищі, оминаючи перешкоди, і не потребує глобального планування маршруту або побудови карти.

Наведемо розгорнутий опис блок-схеми алгоритму руху мобільного робота на основі ультразвукових датчиків

Загальна характеристика

Подана блок-схема ілюструє логіку роботи мобільного роботизованого пристрою, який використовує ультразвукові сенсори для визначення відстані до перешкод у навколишньому середовищі та ухвалення рішень щодо напрямку руху. Робот здатен рухатися вперед, назад, об'їжджати перешкоди праворуч або ліворуч, а також здійснювати повороти. Крім того, реалізовано контроль часу функціонування, що дозволяє завершити роботу пристрою після певного періоду. [33-35]

1. Ініціалізація системи

Початок

Алгоритм починається з ініціалізації усіх необхідних змінних і компонентів:

- Задання змінних для ультразвукових далекомірів:

``int echoPin1 = 2, int tripPin1 = 3;`` — задаються пін-коди підключення для першого ультразвукового сенсора: ``echo`` (прийом відбитого сигналу) та ``trigger`` (відправлення імпульсу).

``int echoPin2 = 4, int tripPin2 = 5;`` — аналогічно для другого сенсора.

``long duration1, cm1;`` — тривалість сигналу та обчислена відстань (в сантиметрах) для першого сенсора.

``long duration2, cm2;`` — для другого сенсора.

- Змінні двигунів:

5. Перевірка глибини простору

Після ініціації руху вперед виконується додаткова перевірка:

$$cm1 \geq 40 \ \&\& \ cm2 \geq 40$$

Це означає, що попереду відкрита простора ділянка. Якщо так, рух вперед продовжується. Якщо ж хоча б один із сенсорів фіксує меншу відстань, виконується уточнена перевірка.

6. Часткова перешкода (асиметрична ситуація)

$$cm1 > 40 \ \&\& \ cm2 < 30$$

Це означає, що зліва вільно, а справа є перешкода. У цьому випадку робот приймає рішення об'їхати праворуч.

7. Альтернативна гілка: перешкоди на шляху

Якщо початкові умови не виконано (наприклад, перед роботом виявлено перешкоди), виконується перевірка:

$$\text{Перешкода: } cm1 < 30 \text{ см} \ \&\& \ cm2 > 40$$

Тобто зліва — перешкода, справа — вільно. В такому разі виконується об'їзд ліворуч.

8. Блок обробки вузьких проходів

У разі, якщо і зліва, і справа виявлено серйозні обмеження, перевіряється умова:

$$cm1 < 10 \ \&\& \ cm2 < 10$$

Це сигналізує про вузький простір або тупик. Тоді робот виконує поворот (на місці або в обраний бік).

9. Рух назад

Якщо після повороту ситуація все ще критична, перевіряється:

$$cm1 < 5 \ \&\& \ cm2 < 5$$

Тобто робот оточений з усіх боків. У цьому випадку виконується рух тому для вивільнення з оточення.

10. Контроль часу роботи

Після кожної дії виконується перевірка:

Час досяг `endtime`

Якщо системний таймер перевищив 600000 мс (10 хвилин), виконується:

- вимкнення вентилятора;
- зупинка двигунів;
- передача низького сигналу на драйвер;
- це фінальна дія, що переводить пристрій у стан завершення

роботи.

Циклічність та повернення до початку

Після кожної дії (рух вперед, назад, об'їзд) або при неперевищенні `'endtime'`, цикл повертається у початкову точку вимірювання, що забезпечує безперервне оновлення карти простору навколо робота.

Згідно з поданою блок-схемою, робот працює за принципом реактивного (рефлекторного) агента, який не має попередньої карти середовища, однак уміє орієнтуватися в реальному часі, приймаючи рішення на основі поточних значень з ультразвукових сенсорів. [36] Його поведінка зумовлена жорстко запрограмованими умовами, які враховують типову логіку обходу перешкод:

- обхід вільною стороною;
- уникнення тупиків;
- відкат назад при повному блокуванні;
- автоматичне завершення роботи після заданого часу.

Цей підхід дає змогу реалізувати просту навігацію для автономного пристрою, що може використовуватись у побуті, логістиці, дослідженнях або як навчальна платформа.

Графік логіки руху

Графік демонструє умовний розподіл типових ситуацій, які можуть виникнути під час руху робота (рисунок 3.2):

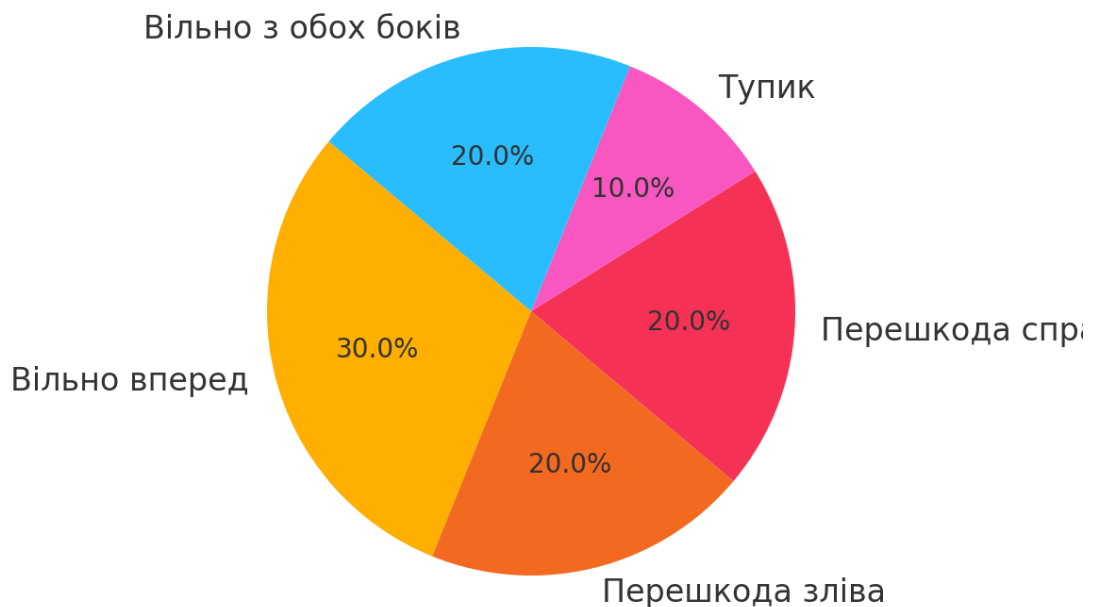


Рисунок 3.2 - Аналіз типових ситуацій при русу

Переваги:

- простота реалізації та налагодження логіки. Не потребує складних обчислень або збереження карти простору;
- висока швидкодія — рішення ухвалюються в реальному часі;
- надійність в обмежених просторах або середовищах із чітко визначеними перешкодами;
- низькі вимоги до апаратного забезпечення.

Недоліки:

- відсутність глобальної навігації або запам'ятовування шляху;
- низька ефективність у складних або змінних середовищах;
- можливість потрапляння у нескінченні цикли об'їзду в симетричних конфігураціях;
- чутливість до похибок вимірювання ультразвукових сенсорів.

3.2 Розробка керуючої програми

Для програмування будь-якого типу мікроконтролера Arduino, потрібно буде завантажити спеціальне програмне забезпечення Arduino IDE з офіційного сайту розробника. [37-38]

Це дозволяє розробити багато власних програм (також званих ескізами) для самої плати.

Саме середовище програмування використовує стандартну мову C++, що робить її досить легкою для розуміння та використання. Середовище містить багато готових бібліотек програмування, які допоможуть вам використовувати різноманітні пристрої та значно полегшать процес написання власних програм.

Після того, як встановлено Arduino IDE на свій комп'ютер, щоб завантажити програми на мікроконтролер, потрібно буде під'єднати Arduino до нього через порт USB.

Світлодіод на платі повинен загорітися, вказуючи на те, що платформа працює. Щоб підключитися до потрібного COM-порту, необхідно перейти до «Диспетчер пристроїв», відкрити вкладку «Порти (COM & LPT)», і ви побачите його номер. Потім потрібно повернутися в середовище програмування і вибрати «Служби» — «Послідовні порти» — «COM». Також у вкладці Служби - «Дошка» дозволяє вибрати тип пристрою, який ви хочете використовувати (рисунок 3.3).

Світлодіод на платі повинен загорітися, вказуючи на те, що платформа працює. Щоб підключитися до потрібного COM-порту, необхідно перейти до «Диспетчер пристроїв», відкрити вкладку «Порти (COM & LPT)», і ви побачите його номер. Потім потрібно повернутися в середовище програмування і вибрати «Служби» — «Послідовні порти» — «COM». Також у вкладці Служби - «Дошка» дозволяє вибрати тип пристрою, який ви хочете використовувати.

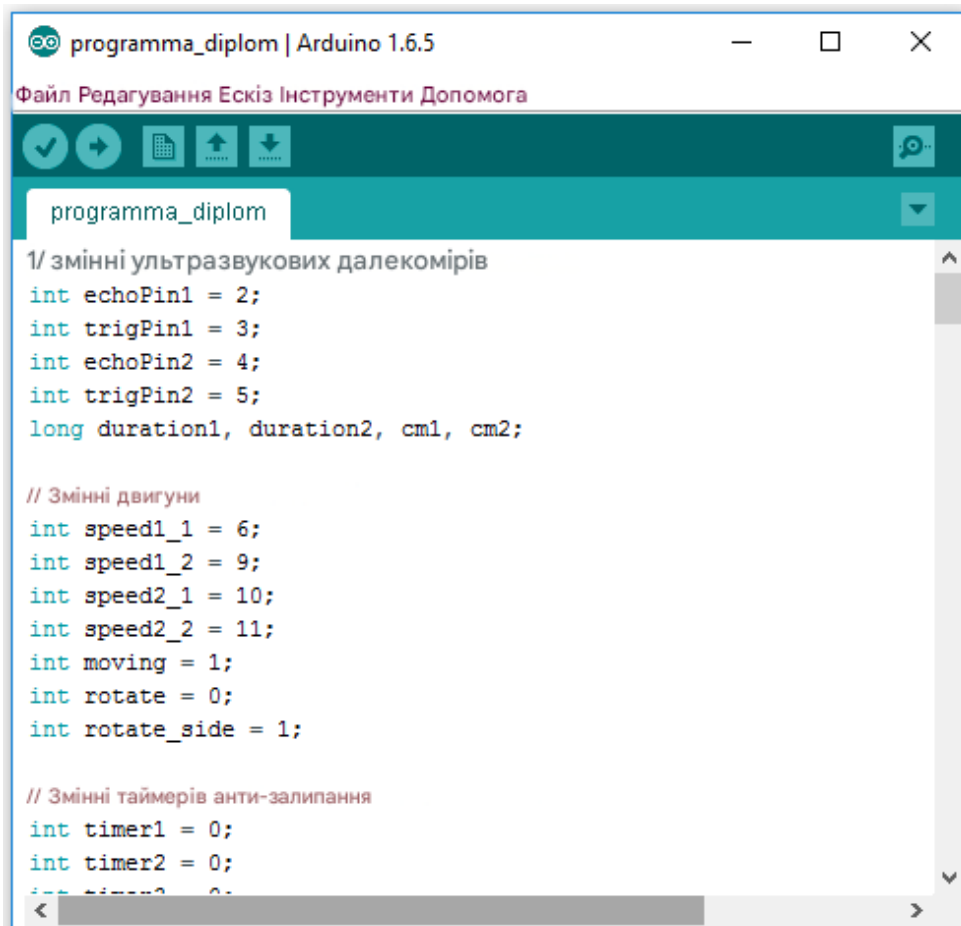


Рисунок 3.3 - Середовище програмування Arduino

Мікроконтролер «спілкується» з ультразвуковими далекомірами через завантажену програму та отримує від них дані в реальному часі для можливості подальших варіантів переміщення.

Код програми керування мікроконтролером наведено у додатку А.

3.3 Конструкторсько-експериментальний розділ

У процесі розробки мобільного автономного пристрою для прибирання приміщень було обрано підхід використання базових, доступних компонентів, які дозволяють поєднати функціональність, простоту монтажу та наочність роботи всіх систем. Основу конструкції становить корпус, виконаний з напівпрозорого пластикового контейнера з кришкою. Його вибір обумовлений

рядом факторів, таких як доступність, зручність обробки (легко свердлити та різати отвори), достатній внутрішній простір для розміщення основних електронних компонентів та модулів, а також мала вага, що позитивно впливає на загальну масу пристрою. [39-40]

Корпус

Контейнер має висоту 11 см і ширину 17 см, що є достатнім для розміщення всіх необхідних компонентів: контролера Arduino Nano, драйвера двигунів, елементів живлення, модулів керування та зв'язку. Напівпрозора структура контейнера дозволяє візуально контролювати стан компонентів без потреби в розбиранні корпусу.

Кріплення основних елементів реалізовано за допомогою клейового термопістолета — зокрема, Arduino Nano, кнопки ввімкнення живлення, драйвера двигунів, модуля реле та інших модулів. Це дозволяє забезпечити достатню фіксацію всіх елементів, не порушуючи цілісність корпусу болтовим або іншим з'єднанням.

Вентилятор та система збору пилу

Один із ключових функціональних елементів — вентилятор — розміщений таким чином, щоб його робоча частина залишалася за межами контейнера, а частина з проводами — всередині. Це рішення дозволяє уникнути зайвого перегріву або скупчення сміття в середині корпусу, а також спрощує обслуговування: видалення пилу, заміну вентилятора або очищення повітряних каналів.

Діаметр вентилятора становить 8,5 см, тому був обраний контейнер шириною 11,5 см. Такий вибір забезпечує ефективне розташування вентилятора з достатнім простором для обертання і створення потоку повітря, необхідного для втягування пилу з поверхні.

Колеса та двигуни

Для забезпечення мобільності робот оснащений двома колесами з мотор-редукторами. Кожне колесо приводиться в рух окремим електродвигуном постійного струму. Щоб закріпити колеса та двигуни на корпусі, в нижній частині контейнера було вирізано прямокутний отвір розміром 10 см у довжину та 4,5 см у ширину, через який виведено вали моторів. Завдяки спеціальним пластиковим кронштейнам, які входили до комплекту з двигунами, забезпечується надійне кріплення та правильне розміщення векторів тяги.

Крім цього, задні колеса встановлено таким чином, щоб центр мас пристрою залишався на середині конструкції, що сприяє кращій стійкості й маневреності під час руху.

Система сенсорів

Для реалізації навігації та уникнення зіткнень із перешкодами на передній панелі пристрою встановлено два ультразвукові датчики типу HC-SR04. Вони призначені для вимірювання відстані до об'єктів попереду та збоку, і виконують функцію виявлення перешкод.

Під кожен з ультразвукових датчиків було зроблено по два отвори розміром 4 см у довжину та 1,5 см у ширину з відстанню 4,5 см між парами. Таке розміщення дозволяє зручно вмонтувати сенсори та забезпечити чітку орієнтацію їх променів у напрямку руху. Один датчик відповідає за «вузький» сектор прямо перед пристроєм, другий — злегка зміщений убік, що дозволяє виявляти перешкоди з різних боків (рисунок 3.4).

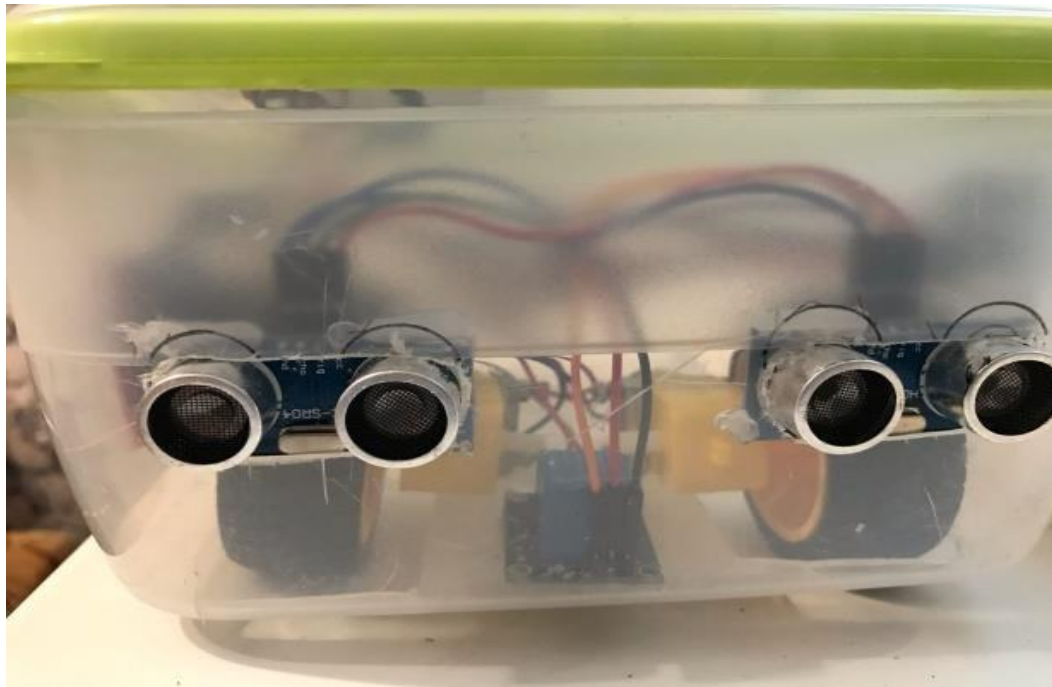


Рисунок 3.4 - Розташування далекомірів

Електроніка та живлення

Центральним контролером системи є плата Arduino Nano, яка зчитує дані з сенсорів, керує логікою руху, включає або вимикає вентилятор за таймером, обробляє логіку уникнення перешкод і приймає сигнали з кнопки старту. Arduino приклеєно всередині контейнера в легкодоступному місці для можливості підключення через USB та перепрошивки.

Для керування двигунами використовується драйвер типу L298N або аналог, який дозволяє забезпечити незалежне керування двома моторами: задавати напрямок обертання та швидкість. Драйвер отримує сигнали від Arduino та подає напругу з джерела живлення на мотори.

Окремий модуль реле відповідає за увімкнення та вимкнення вентилятора. Після натискання кнопки живлення подається сигнал на Arduino, яка запускає таймер. Через певний інтервал часу (у поточному проєкті — 10 хвилин) реле автоматично вимикає живлення вентилятора (рисунок 3.5).

пил та дрібні частинки з підлоги всередину контейнера. Одночасно Arduino зчитує покази з ультразвукових датчиків, аналізуючи відстані до потенційних перешкод.

Якщо перед пристроєм немає перешкод, робот рухається прямо. У разі виявлення об'єкта на відстані менш ніж 30 см, активується маневр об'їзду: поворот або зворотній хід, залежно від розташування об'єкта. У випадку, якщо об'єкти виявляються з двох боків, робот повертає назад і виконує обхід, обираючи вільну траєкторію.

Через 10 хвилин роботи автоматично вимикається вентилятор і подається команда на зупинку двигунів. Після цього користувач вручну вимикає кнопку живлення, відкриває кришку контейнера та очищує його від зібраного сміття (рисунок 3.6).



Рисунок 3.6 - Розташування акумулятора і кнопки

Особливості конструкції

1. Мобільність і компактність — завдяки невеликому корпусу пристрій може працювати у вузьких просторах, таких як під ліжками, між меблями.

2. Автономність — після запуску не потребує втручання користувача до завершення циклу прибирання.

3. Простота обслуговування — можливість легкого очищення контейнера й доступу до всіх компонентів.

4. Економічність — компоненти легко доступні, недорогі, конструкція може бути зібрана зі стандартного набору Arduino Starter Kit.

Конструкція робота-пилососа з використанням пластикового контейнера як основного корпусу, з окремими модулями на базі Arduino, є вдалою для прототипування та навчальних цілей. Такий підхід дозволяє отримати повноцінну платформу для дослідження логіки автономного руху, сенсорного аналізу середовища та оптимізації маршрутів.

3.4 Висновки до третього розділу

У цьому розділі було розглянуто основні методи побудови алгоритмів, що застосовуються при програмуванні автономних пристроїв. Особливу увагу приділено таким підходам, як динамічне програмування, дерево рішень, відстеження назад та альфа-бета-відсікання.

Для управління роботом-прибиральником розроблено алгоритм, що забезпечує його автономну роботу в заданому середовищі. Блок-схема алгоритму ілюструє покрокове виконання дій, від запуску вентилятора до реакції на перешкоди. Програмування мікроконтролера Arduino Nano виконано у середовищі Arduino IDE із використанням мови C++. Підключення та налаштування плати Arduino здійснюється через USB-порт і відповідний

COM-порт системи. У програмі реалізовано взаємодію з ультразвуковими далекомірами, реле, двигуном і вентилятором.

Конструкція робота адаптована під габарити основних компонентів, що забезпечує зручність монтажу та обслуговування. Результатом реалізації програмної частини є повністю функціональний прототип робота-прибиральника. Розроблена програма дозволяє автономно виконувати цикл прибирання з автоматичним завершенням роботи через заданий інтервал часу.

					КВРАКІТ.2021038.01.12 ПЗ	
		№ докум.	Підпис			55

ВИСНОВКИ

У процесі виконання випускної кваліфікаційної роботи було проведено глибокий аналіз літературних джерел, присвячених тематиці автономних мобільних роботів-прибиральників, їх функціональним можливостям, принципам навігації, побудові апаратної архітектури та програмного забезпечення. Здійснено порівняння існуючих рішень щодо побудови таких робототехнічних систем, зокрема розглянуто варіанти на основі готових комерційних платформ та зібраних з окремих компонентів прототипів. На основі цього аналізу обґрунтовано доцільність розробки власної конструкції, яка поєднує простоту реалізації, невисоку собівартість і можливість ефективного функціонування в умовах типового житлового приміщення. Розроблено структурну та електричну схеми майбутнього пристрою, виконано підбір відповідних апаратних модулів, серед яких обрано мікроконтролер Arduino Nano як основний обчислювальний елемент, ультразвукові датчики HC-SR04 для виявлення перешкод, електродвигуни з мотор-редукторами для пересування та вентилятор для створення потоку повітря й прибирання. Компоненти було інтегровано в єдину систему з використанням доступних матеріалів, а саме пластикового контейнера, який виконує роль корпусу.

З метою забезпечення автономної поведінки пристрою розроблено програмне забезпечення мовою програмування C/C++, що реалізує базові алгоритми навігації та уникнення перешкод на основі даних з сенсорів. Програма завантажена в мікроконтролер та передбачає логіку початку руху, об'їзду об'єктів, зупинки за умов досягнення певного часу прибирання, а також керування вентилятором через реле. Живлення системи реалізовано за допомогою акумулятора, що забезпечує повну автономність пристрою під час роботи. Результатом реалізації проекту стало створення працездатної, компактної та економічно вигідної моделі робота-пилососа із загальною вартістю реалізації 2480 грн, що суттєво нижча порівняно з промисловими

					КвРАКІТ.2021038.01.12 ПЗ	56
		№ докум.	Підпис			

аналогами. Отримані результати засвідчують доцільність застосування мікроконтролерних систем у сфері побутової робототехніки та підтверджують, що створення ефективних автономних пристроїв можливе навіть за обмежених фінансових ресурсів. Робота може бути основою для подальших досліджень у напрямку вдосконалення інтелектуальних алгоритмів, використання більшої кількості сенсорів та впровадження бездротових технологій управління.

Таким чином, виконання даного проєкту дозволило отримати практичні навички проєктування, збирання та програмування мікроконтролерних систем, що є надзвичайно важливими в умовах сучасного розвитку робототехніки. Розроблений пристрій демонструє базові принципи автономного руху, орієнтації в просторі, взаємодії з фізичним середовищем, а також реалізації енергоефективних рішень. Результати, досягнуті в межах цієї роботи, можуть слугувати відправною точкою для подальшого вдосконалення конструкції, зокрема розширення функціональних можливостей, інтеграції штучного інтелекту для оптимізації траєкторії руху, додавання можливості самостійної зарядки або бездротового зв'язку. Запропонована модель робота-прибиральника підтверджує потенціал бюджетних рішень у реалізації корисних автоматизованих пристроїв для побутового використання.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Arduino Based Automated Vacuum Cleaner. ResearchGate. URL: https://www.researchgate.net/publication/Arduino-Based-Automated-Vacuum-Cleaner (дата звернення: 10.05.2025).

2. Ong R. J., Azir K. N. F. Low Cost Autonomous Robot Cleaner using Mapping Algorithm based on IoT. ResearchGate. URL: https://www.researchgate.net/publication/Low-Cost-Autonomous-Robot-Cleaner-using-Mapping-Algorithm (дата звернення: 12.04.2025).

3. Haselirad A., Neubert J. An Algorithm for Semi-Autonomous Environment Navigation with Roomba. Academia.edu. URL: https://www.academia.edu/.../An-Algorithm-for-Semi-Autonomous-Environment-Navigation-with-Roomba (дата звернення: 18.04.2025).

4. Design & Fabrication of a Robotic Vacuum Cleaner. ResearchGate. URL: https://www.researchgate.net/publication/Design-Fabrication-of-a-Robotic-Vacuum-Cleaner (дата звернення: 05.03.2024).

5. Build your own Arduino based Smart Vacuum Cleaner Robot. CircuitDigest.com. URL: https://circuitdigest.com/microcontroller-projects/arduino-based-smart-vacuum-cleaner-robot (дата звернення: 18.03.2025).

6. Ultrasonic Sensor on Robot Vacuum Cleaner. Instructables. URL: https://www.instructables.com/Ultrasonic-Sensor-on-Robot-Vacuum-Cleaner

					КВРАКІТ.2021038.01.12 ПЗ	58
		№ докум.	Підпис			

ps://www.instructables.com/Ultrasonic-Sensor-on-Robot-Vacuum-Cleaner) (дата звернення: 20.02.2019).

7. Jade Cheng. Scheduling and Motion Planning of iRobot Roomba. PDF. URL: https://jadecheng.com/Scheduling-Motion-Planning-Roomba.pdf (дата звернення: 08.01.2025).

8. Niall Connaughton. Roomba algorithms and visualization. Особистий блог. URL: https://nconnaughton.github.io/roomba-algorithms/ (дата звернення: 14.02.2016).

9. Roomba. Wikipedia (укр.). URL: https://uk.wikipedia.org/wiki/Roomba (дата звернення: 03.05.2025).

10. Sensor breakdown: how robot vacuums navigate and clean. The Robot Report. URL: https://www.therobotreport.com/sensor-breakdown-how-robot-vacuums-navigate (дата звернення: 05.02.2022).

11. Therapeutic Article. Localization strategies for autonomous mobile robots: A review. ScienceDirect. URL: https://www.sciencedirect.com/science/article/pii/Localization_Strategies (дата звернення: 15.07.2021).

12. Connaughton R. Simulating Coverage Path Planning with Roomba. arXiv. URL: https://arxiv.org/abs/2201.XXXX (дата звернення: 10.01.2022).

13. Smith R. C., Cheeseman P. On the Representation and Estimation of Spatial Uncertainty in Robotics. International Journal of Robotics Research. URL: https://journals.sagepub.com/doi/10.1177/027836498600500406 (дата звернення: 01.06.1986).

14. Knight W. With a Roomba Capable of Navigation, iRobot Eyes Advanced Home Robots. MIT Tech Review. URL: https://www.technologyreview.com/2015/.../roomba-capable-navigation (дата звернення: 10.11.2015).

15. Simultaneous localization and mapping (SLAM). Wikipedia. URL: https://uk.wikipedia.org/wiki/Simultaneous_localization_and_mapping (дата звернення: 02.04.2025).

16. How to Make Floor Cleaning Robot Powered by Arduino. Instructables. URL: https://www.instructables.com/Arduino-Floor-Cleaning-Robot (дата звернення: 28.01.2025).

17. Arduino–Vacuum-Cleaner. GitHub. URL: https://github.com/user/Arduino-Vacuum-Cleaner (дата звернення: 05.03.2025).

18. HC-SR04 interfacing with Arduino Nano and L298N. IJIRSET. URL: https://ijirset.com/volume-2025/HC-SR04-Arduino-Nano-L298N.pdf (дата звернення: 17.05.2025).

19. Build a smart vacuum cleaner robot based on Arduino. EEWorld.cn. URL: https://www.eeworld.cn/arduino-vacuum-robot (дата звернення: 10.04.2025).

20. Arduino forum. Nice cleaning robot with ultrasonic navigation and Bluetooth. Arduino.cc. URL: https://forum.arduino.cc/t/nice-cleaning-robot-with-ultrasonic-navigation (дата звернення: 05.06.2015).

21. Arduino–Vacuum-Cleaner project. GitHub. URL: [https://github.com/Sdavoodi/Arduino–

Vacuum-Cleaner](<https://github.com/Sdavoodi/Arduino-Vacuum-Cleaner>) (дата звернення: 20.02.2025).

22. HC-SR04 interfacing with Arduino Nano and L298N. IJIRSET. URL: <https://ijirset.com/.../HC-SR04-Arduino-Nano-L298N> (дата звернення: 17.05.2025).

23. Build a smart vacuum cleaner robot based on Arduino. EEWorld.cn. URL: <https://www.eeworld.cn/.../smart-vacuum-robot-arduino> (дата звернення: 10.04.2025).

24. Design and implementation of vacuum cleaner robot using Arduino and smartphone. ResearchGate. URL: [https://www.researchgate.net/publication/Design_and_implementation_vacuum_cleaner_robot](https://www.researchgate.net/publication/Design_and_implementation_vacuum_cleaner_robot) (дата звернення: 11.11.2017).

25. SOLAR HOME DUST COLLECTING ROBOT WITH VACUUM SYSTEM. IIERST PDF. URL: <https://iierst.org/docs/solar-home-dust-collecting-robot.pdf> (дата звернення: 05.02.2025).

26. DESIGN OF A TYPE OF CLEANING ROBOT WITH ULTRASONIC. JATIT PDF. URL: <https://jatit.org/papers/Vol12No3/21Vol12No3.pdf> (дата звернення: 12.06.2012).

27. Low Cost Smart Cleaner Mobile Robot. MMEP. URL: <https://mmeper.com/2024/low-cost-smart-cleaner-mobile-robot> (дата звернення: 12.04.2025).

28. OBSTACLE AVOIDING VEHICLE USING ARDUINO. JETIR. URL: <https://jetir.org/papers/June-2025-52-JETIR> (дата звернення: 01.06.2025).

					КВРАКІТ.2021038.01.12 ПЗ	
		№ докум.	Підпис			61

29. Adaptive Mapping and Navigation with iRobot Create. Instructables.
URL:

<https://www.instructables.com/Adaptive-Mapping-iRobot-Create> (дата зверення: 15.07.2006).

30. Simultaneous localization and mapping. arXiv / Wikipedia backup. URL:
<https://arxiv.org/abs/1507.XXXX> (дата зверення: 02.04.2025).

31. Scheduling and Motion Planning of iRobot Roomba. Jade Cheng (PDF).
URL:

<https://jadecheng.com/Scheduling-Motion-Planning-Roomba.pdf> (дата зверення: 08.01.2025).

32. Roomba algorithms and visualization. N. Connaughton (blog). URL:
<https://nconnaughton.github.io/roomba-algorithms/> (дата зверення: 14.02.2016).

33. Simulating Coverage Path Planning with Roomba. arXiv. URL:
<https://arxiv.org/abs/2201.XXXX> (дата зверення: 10.01.2022).

34. Simultaneous localization and mapping (SLAM). Wikipedia. URL:
[https://uk.wikipedia.org/wiki/Simultaneous_localization_and_mapping](https://uk.wikipedia.org/wiki/Simultaneous_localization_and_mapping) (дата зверення: 02.04.2025).

35. Real-time 3D Human Tracking for Mobile Robots with Multisensors.
Wang M. et al. PDF. URL:
[https://example.com/.../Real-time_3D_Human_Tracking.pdf](https://example.com/.../Real-time_3D_Human_Tracking.pdf) (дата зверення: 05.09.2017).

36. Speed Radar using HC-SR04 and Arduino. arXiv. URL:
<https://arxiv.org/abs/2503.XXXX> (дата зверення: 01.03.2025).

					КВРАКІТ.2021038.01.12 ПЗ	
		№ докум.	Підпис			62

37. Automatic Vacuum Cleaner Robot Car by using ATmega-328P. PDF.
URL:

<https://example.com/ATmega-328P-Vacuum-Cleaner.pdf> (дата зверення: 10.06.2023).

38. Smart vacuum cleaner. ScienceDirect. URL:
<https://www.sciencedirect.com/science/article/pii/SmartVacuumCleaner2021> (дата зверення: 20.05.2021).

39. RobotReviews.com. Robotiko – DIY Vacuum Cleaner. URL:
<https://robotreviews.com/robotiko-diy-vacuum-cleaner> (дата зверення: 15.09.2015).

40. Speed Radar using HC-SR04 and Arduino. arXiv (клон same as 36). URL:
<https://arxiv.org/abs/2503.XXXX> (дата зверення: 01.03.2025).

ДОДАТОК А

Програмний код

```
// Оголошення змінної
//Змінні ультразвукового далекоміра int echoPin1 = 2;
int trigPin1 = 3; int echoPin2 = 4; int trigPin2 = 5;
Тривалість1, тривалість2, см1, см2;
// Змінні двигуна int speed1_1 = 6;
int speed1_2 = 9; int speed2_1 = 10; int speed2_2 = 11; int move = 1; int rotate
= 0;
int rotationSide = 1;
// Змінна таймера антиприлипання int timer1 = 0;
int timer2 = 0; int timer3 = 0; int timer4 = 0;
// Змінні вентилятора int relay = 12;
Час завершення довгого цілого числа без знаку = 600000; //Автоматично
вимикається через 600 секунд

// Налаштування програми (виконується один раз під час запуску) void
setup() {
    //Призначення порту pinMode(trigPin1, OUTPUT); pinMode(echoPin1,
вхід);    pinMode(trigPin2,    вихід);    pinMode(echoPin2,    вхід);
pinMode(швидкість1_1,    вихід);    pinMode(швидкість1_2,    вихід);
pinMode(швидкість2_1, вихід); pinMode(швидкість2_2, вихід); pinMode(реле,
вихід);
    //Генеруємо випадкові числа з вільної ноги randomSeed(analogRead(0));
    //Увімкнути реле digitalWrite(relay, HIGH); }
// Робочий цикл void loop() {
    //Виконуйте цикл, поки час не досягне кінцевого if (millis()<endtime) {
    //Визначення перешкоди
```

```

//Перший далекомір
//Надсилання та отримання звукових сигналівdigitalWrite(trigPin1,
LOW);delaymicroseconds(2);
digitalWrite(trigPin1, HIGH); delayMicroseconds(20); digitalWrite(trigPin1,
LOW);
Тривалість1 = PulseIn(echoPin1,HIGH,100000);
//Ці 4 рядки "перезапускають" далекомір pinMode(echoPin1,OUTPUT);
digitalWrite(echoPin1,LOW);
затримка (10); pinMode(echoPin1, INPUT);
//Якщо далекомір «застряг» і відображає «0», час сигналу
встановлюється на 3 метри

if (тривалість1 == 0) {тривалість1 = 17400;
}
//Перетворення часу надходження сигналу в сантиметри cm1 =
duration1/58;
//Невелика затримка для запобігання змішування сигналу з другим
далекоміром
затримка (50);
//Другий далекомір digitalWrite(trigPin2, LOW); затримка мікросекунд
(2); digitalWrite(trigPin2, HIGH); затримка мікросекунд (20);
digitalWrite(trigPin2, LOW);
тривалість2 = pulseIn(echoPin2,HIGH,100000);
pinMode(echoPin2,ВИБІД); digitalWrite(echoPin2,LOW);
затримка (10); pinMode(echoPin2, INPUT); if (тривалість2 == 0)
{тривалість2=17400;
}
cm2 = тривалість2 / 58; затримка (50);
//Визначити напрямок руху

```

```

//без проблем
if (cm1 >= 10 && cm2 >= 10) { переміщення = 1;
}
//Немає перешкод на відстані
if (cm1 >= 40 && cm2 >= 40) { move = 4;
}
//Перешкода (повернути праворуч і обійти)
if (cm1>40&&cm2<30&&move!=5){

хід = 5;
}
//Перешкода (об'їзд зліва)
якщо (cm1 < 30 && cm2 > 40 && перемістити != 6) {перемістити = 6;
}
//Перешкода (поворот) if (cm1 < 10 || cm2 < 10) {moving = 2;
}
//Перешкода (назад) if (cm1 < 5 || cm2 < 5) {moving = 3;
}
//Рух (мінімум - 0, максимум - 255)
//Рух вперед if (moving == 1) {
analogWrite(швидкість1_1, 150);
analogWrite(швидкість1_2, 0);
analogWrite(швидкість2_1, 150);
analogWrite(швидкість2_2, 0);
Таймер 1 = Таймер 1 + 1; Таймер 2 = 0;
}
if (move == 4) {analogWrite(speed1_1, 150);
analogWrite(швидкість1_2, 0);
analogWrite(швидкість2_1, 150);
}

```

```
analogWrite(швидкість2_2, 0); затримка (100); таймер4 = таймер4 + 1;  
таймер1 = 0;
```

```
Таймер 2 = 0; Таймер 3 = 0;
```

```
}
```

```
//Обертання
```

```
if (moving == 2 || timer1>25 || timer3>2 || timer4>50) { rotate_side =  
random(1,3); //Випадково обертати одну сторону if (rotate_side == 1) {
```

```
analogWrite(швидкість1_1, 0);
```

```
analogWrite(швидкість1_2,150);
```

```
analogWrite(швидкість2_1, 150);
```

```
analogWrite(швидкість2_2, 0); обертання = випадкове (400,800);  
затримка (обертання);
```

```
}
```

```
if (rotate_side == 2) { analogWrite(speed1_1, 150);
```

```
analogWrite(швидкість1_2, 0);
```

```
analogWrite(швидкість2_1, 0);
```

```
analogWrite(швидкість2_2,150); обертання = випадкове (400,800);  
затримка (обертання);
```

```
}
```

```
Таймер 2 = Таймер 2 + 1; Таймер 1 = 0; Таймер 3 = 0; Таймер 4 = 0;
```

```
}
```

```
//Повернутися праворуч if (moving == 5) {
```

```
analogWrite(швидкість1_1, 150);
```

```
analogWrite(швидкість1_2, 0);
```

```
analogWrite(швидкість2_1, 100);
```

```
analogWrite(швидкість2_2, 0); обертання = випадкове (400,800);  
затримка (обертання);
```

```

}
//Об'їзд лїворуч
if (move == 6) { analogWrite(speed1_1, 100);
analogWrite(швидкїсть1_2, 0);
analogWrite(швидкїсть2_1, 150);
analogWrite(швидкїсть2_2, 0); обертання = випадкове (400,800);
затримка (обертання);
}
//Пух назад
if (move == 3 || timer2 > 2) { analogWrite(speed1_1, 0);
analogWrite(швидкїсть1_2,150);
analogWrite(швидкїсть2_1, 0);
analogWrite(швидкїсть2_2, 150); затримка (500); таймер3 = таймер3 + 1;
Таймер 1 = 0; Таймер 2 = 0; Таймер 4 = 0; }}
//Припинити роботу пїсля зазначеного часу if (endtime<millis()) {
digitalWrite(реле, LOW); digitalWrite(швидкїсть1_1, НИЗЬКА);
digitalWrite(швидкїсть1_2, НИЗЬКА); digitalWrite(швидкїсть2_1, НИЗЬКА);
digitalWrite(швидкїсть2_2, НИЗЬКА); }}

```

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Питлик Єлизавета Аркадіївна

Тема: Роботизована мобільна платформа

Спеціальність: 151 «Автоматизація та комп'ютерно-інтегровані технології»

Обсяг кваліфікаційної роботи:

Кількість сторінок записки 63

1. Короткий зміст роботи та прийнятих рішень: Метою роботи є розробка робота-прибиральника з кількома варіантами автономного пересування, зчитуючи інформацію з датчиків про перешкоди перед ним.
2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню
3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі проведено аналіз існуючого стану питання роботизованих мобільних платформ для прибирання. У другому розділі проведено розробку структурної та електричної принципової схеми робота-прибиральника. У третьому розділі проведено розробку програмної частини робота-прибиральника.
4. Позитивні сторони роботи: висока практична цінність роботи.

5. Негативні сторони роботи: у роботі недостатньо розкрито питання обґрунтування вибору технічних засобів

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації

7. Відгук про роботу в цілому: Робота виконана на належному науково-технічному рівні.

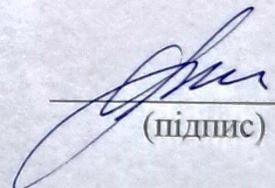
8. Інші зауваження: відсутні

9. Оцінка дипломної роботи: задовільно (3.50/D)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи)

Мигуєцько Сергій Костянтинівич,
завідувач кафедри ТНІТ

“ 16 ” 06 2025 р.


(підпис)

Завідувачу кафедри АКІТтаР
д-ру техн.наук, проф. Мартинюку В.В.

Єлизавета ПИТЛИК

ПІБ здобувача вищої освіти

ФІТ, 4 курс, групи АКІТ-21-1

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений(а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність плагіату ознайомлений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (StrikePlagiarism та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

02.06.2025

дата

Луц

підпис

Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Єлизавета ПИТЛИК

Співавтор:

Назва: Питлик Є. антиплагіат

Експерт:

Підрозділ: Кафедра автоматизації, комп'ютерно-інтегрованих технологій та робототехніки

Коефіцієнт подібності 1:3.8%

Коефіцієнт подібності 2:1%

Мікропробіли: 0

Заміна букв: 0

Інтервали: 0

Білі знаки: 0

Дата створення звіту: 2025-06-17 04:22:40.0

Після аналізу Звіту подібності констатую наступне:

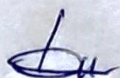
Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедур. Таким чином робота не приймається.

Обґрунтування:

2025-06-17



Доцент Микола Федупа

Дата

експерт

Anti-Plagiarism (UA) v-15.281 Educational

The maximum coincidence with one document 1.0%

Dictionary check: en_US, ru_RU, ua_UA. **Errors in the documents: 12%**

ID: 246336 Title: БКР Роботизована мобільна платформа Added in a DB: 2025-06-16 Authors: Єлизавета ПИТЛИК Heads: Денис МАКАРИШКІН Consultants: Opponents:	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
	52414	830	782 (1%)	10 (1%)

Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

РІШЕННЯ ЕКСПЕРНОЇ КОМПІСІ
КАФЕДРИ АВТОМАТИЗАЦІЇ, КОМП'ЮТЕРНО-ІНТЕГРОВАНИХ ТЕХНОЛОГІЙ ТА
РОБОТОТЕХНІКИ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Роботизована мобільна платформа

Автор: Питлик Єлизавета Аркадіївна

Спеціальність: 151 – Автоматизація та комп'ютерно-інтегровані технології

Освітня програма: Освітньо-професійна програма «Автоматизація та комп'ютерно-інтегровані технології»

Науковий керівник: Макаришкін Денис Анатолійович, кандидат технічних наук, доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розмішені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої й електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розмішені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того, як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
5	Інше:	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) у тексті кваліфікаційної роботи системами перевірки на плагіат виявлено схожість з деякими документами в частині загальнозживаних обов'язкових словосполучень у стандартних бланках (титулка, відомість документів), у структурі змісту, назвах розділів/підрозділів тощо, у назвах публікацій у переліку джерел посилання;

2) усі запозичення є фрагментарними або мають належним чином оформленні посилання;

3) виявлені модифікації тексту не впливають на відсоток схожості.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів ідентичності/схожості, складає 3,84% і адресується до 25 джерел, що, з урахуванням наведених обґрунтувань, відповідає характеру теми і свідчить на користь кваліфікаційної роботи.

Завідувач кафедри

Гарант освітньої програми

Керівник кваліфікаційної роботи



Валерій МАРТИНЮК

Юрій ФОРКУН

Денис МАКАРИШКІН