

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

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень

Робот пилосос з автоматичним рухом

Назва теми

КВРАКІТ.2021044.01.17 ПЗ

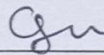
Галузь знань 15 «Автоматизація та приладобудування»
Шифр, назва

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

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

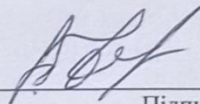
Виконав:

студент IV курсу, група АКІТ-21-1


Підпис

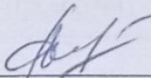
Андрій СІМ'ЯЧЕНКО
Ім'я, ПРІЗВИЩЕ

Керівник


Підпис, дата

Юрій ФОРКУН
Ім'я, ПРІЗВИЩЕ

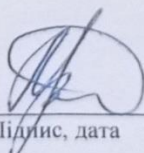
Нормоконтролер


Підпис, дата

Людмила КОРЕЦЬКА
Ім'я, ПРІЗВИЩЕ

До захисту допускаю:

зав. кафедри автоматизації
та комп'ютерно-інтегрованих
технологій та робототехніки


Підпис, дата

Валерій МАРТИНЮК
Ім'я, ПРІЗВИЩЕ

« 19 » червня 2025 р.

Хмельницький 2025

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

СІМ'ЯЧЕНКО Андрій Вікторович

Прізвище, ім'я, по батькові студента

1 Тема роботи Робот пилосос з автоматичним рухом

Керівник роботи канд.техн.наук., доцент Юрій ФОРКУН

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

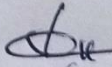
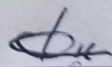
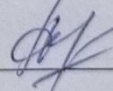
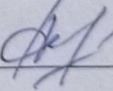
4 Зміст пояснювальної записки (перелік питань, які потрібно розробити)

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
7 Отримання супровідних документів (відгуку керівника, рецензії, довідки про перевірку на плагіат); нормоконтроль	30.05.2025	Виконано
8 Підготовка до захисту та захист кваліфікаційної роботи	03.06.2025	Виконано

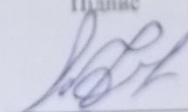
Студент



Підпис

Андрій СІМ'ЯЧЕНКО
Ім'я, ПРІЗВИЩЕ

Керівник кваліфікаційної роботи



Підпис

Юрій ФОРКУН
Ім'я, ПРІЗВИЩЕ

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Робот пилосос з автоматичним рухом».

Автор роботи: Андрій СІМ'ЯЧЕНКО

Керівник роботи: Юрій ФОРКУН

Пояснювальна записка: 66 с., 40 рис., 6 табл., 2 дод., 40 джерел.

Графічна частина: 15 презентаційних слайдів.

СИСТЕМИ КЕРУВАННЯ, РОБОТ_ПИЛОСОС, МІКРОКОНТРОЛЕРНИЙ ПРИБОР, БЛОК-СХЕМА АЛГОРИТМУ.

Метою роботи є розробка схемотехнічного рішення роботу пилососу з автоматичним рухом.

У бакалаврській роботі була розроблена автономна мобільна платформа, яка могла переміщатися по доріжці чорної лінії та відстежувалась за допомогою інфрачервоних датчиків. На трасі з'явилися несподівані перешкоди. Під час реалізації було враховано актуальність теми, а також існуючі роботизовані рішення у сфері складної логістики. Була розроблена блок-схема, яка показує кількість пристроїв у системі та їх взаємодію. Виберіть, порівняйте та виберіть необхідні компоненти, такі як: інфрачервоний датчик, РК-дисплей 1602, імпульсна кнопка тощо. Платформа зібрана, мікроконтролер встановлений, усі компоненти підключені згідно зі схемою, налаштований та відкалібрований інфрачервоний датчик схеми.



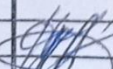

Підпис студента

19.06.25

Дата

ЗМІСТ

ВСТУП	3
1 СТАН ПИТАННЯ	5
1.1 Формулювання актуальності проекту	5
1.2 Аналіз вихідних даних і відомих рішень	5
1.3 Висновки до першого розділу	23
2 АПАРАТНА ЧАСТИНА	24
2.1 Розробка структурної схеми	24
2.2 Вибір необхідних компонентів	25
2.3 Розробка електричної принципової схеми	38
2.4 Висновки до другого розділу	38
3 ПРОГРАМНА ЧАСТИНА	40
3.1 Розробка алгоритму роботи	40
3.2 Розробка програмної частини пристрої	46
3.3 Висновки до третього розділу	52
4 КОНСТРУКТОРСЬКО-ЕКСПЕРИМЕНТАЛЬНИЙ РОЗДІЛ	53
4.1 Складання пристрої	53
4.2 Налагодження пристроїв і експериментальні дослідження	56
4.3 Висновок до четвертого розділу	59
ВИСНОВКИ	61
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ	64
Додаток А Програма керування автоматизованою мобільною платформи	67

КвРАКІТ.2021044.01.17 ПЗ											
<i>Зм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>							
<i>Розроб.</i>		<i>Сім'яченко А.В.</i>		<i>19.06.25</i>	Робот пиросос з автоматичним рухом Пояснювальна записка						
<i>Перевір.</i>		<i>Форкун Ю.В.</i>		<i>19.06.25</i>							
<i>Н. Контр.</i>		<i>Корецька Л.О.</i>		<i>19.06.25</i>							
<i>Затв.</i>		<i>Мартинюк В.В.</i>		<i>19.06.25</i>							
					<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%;"><i>Літ.</i></td> <td style="width: 20%;"><i>Лист</i></td> <td style="width: 60%;"><i>Листів</i></td> </tr> <tr> <td></td> <td style="text-align: center;">2</td> <td></td> </tr> </table>	<i>Літ.</i>	<i>Лист</i>	<i>Листів</i>		2	
<i>Літ.</i>	<i>Лист</i>	<i>Листів</i>									
	2										
					ХНУ, АКІТ-21-1						

ВСТУП

В даний час рівень розвитку робототехніки досяг надзвичайної висоти. Сучасна ситуація розвитку робототехніки така, що зупинити її рух у цьому напрямку неможливо. Це тому, що роботи вже зайняли своє місце в суспільному житті. Вони стали частиною сучасної промислової революції, яка характеризується широким впровадженням адаптивних технологій і роботизацією виробництва. Кількість компаній, що автоматизують свої процеси, зростає з кожним роком, тому фабрика, на якій працює всього кілька десятків людей і де всю основну роботу виконують роботи, вже нікого не дивує. Промислові роботи випускаються тисячами. Хоча цей ринок існує вже деякий час, справи тут погіршуються. Мотивація проактивної роботизації складів полягає в оптимізації простору, прискоренні операцій і виключенні людей з роботи на складі.

У нашій країні повна автоматизація промисловості і виробництва поки що далеке майбутнє. Поки що існує тенденція до автоматизації частин процесу без довгострокових витрат. Вилочні навантажувачі та роботизовані візки впроваджуються, щоб замінити людей на певному етапі процесу логістики.

Однією з проблем є забезпечення навігації в межах складу. Для цього були прийняті різні методи. Деякі вимагають підготовки складу – розмітки підлоги або встановлення спеціальних бирок (бездротових або світловідбиваючих) на стінах і полицях [1-5].

Метою цієї роботи є розробка автономної мобільної платформи, яка рухається за чорною лінією за певним алгоритмом і заданим маршрутом.

Завданнями роботи є:

- розробити блок-схему;
- вибрати необхідні компоненти;
- розробити електричні схеми;

- розробити алгоритми роботи;
- розробка програмної частини обладнання;
- зібрати обладнання;
- налаштування пристрою та експериментальні дослідження.
-

					КвРАКІТ.2021044.01.17 ПЗ	
		№ докум.	Підпис			4

1 СТАН ПИТАННЯ

1.1 Формулювання актуальності проекту

Автоматизація складських процесів і роботизація складу є одними з актуальних напрямків розвитку робототехніки. Системи, розроблені в цьому напрямку, можуть прискорити різні операції. Їх впровадження підвищує продуктивність праці, оскільки за той самий час машини можуть обробити на порядок більше продукції, ніж спеціально навчений персонал. Покращена безпека персоналу. Це особливо вірно при роботі з небезпечними та небезпечними матеріалами. Зменшуються трудовитрати.

Незважаючи на важливість цього напрямку, у світі існує дуже обмежена кількість готових рішень, які дозволяють створювати мобільні платформи, роботів і системи, здатні виконувати складські завдання [1-5].

1.2 Аналіз вихідних даних і відомих рішень

Розглянемо кілька готових рішень. Amazon Robotics була однією з перших компаній, яка запровадила роботизовані системи на складах. У компанії вирішили, що люди повинні залишатися в одному конкретному місці, а не бігати по складу в пошуках товару. Цю роботу за них мають виконувати роботи. Це значно полегшить працю людини. Невдовзі вони запустили розумну складську систему, у якій роботизована платформа доставляє невеликі стелажі певних товарів із місць зберігання до складських працівників. Отже, залишилося лише взяти продукт з полиці, відсканувати його та поставити на конвеєр. При цьому працівникам забороняється вхід на території складу, де зберігаються пересувні платформи та стелажі. Ці роботи являють собою невеликі візки на колесах з механізмами, які піднімають полиці на невелику відстань від землі для

					КвРАКІТ.2021044.01.17 ПЗ	5
		№ докум.	Підпис			

транспортування. Платформи позиціонуються за допомогою спеціальної розмітки, нанесеної на підлогу. Вони також оснащені датчиками перешкод, щоб запобігти випадковому зіткненню з іншими платформами. Платформа живиться від акумуляторної батареї, яка заряджається протягом п'яти хвилин щогодини. Розумні складські системи можуть відстежувати часто потрібні товари та розміщувати їх ближче до співробітників. За оцінками компанії, ця система приблизно подвоїла продуктивність співробітників. На рисунку 1.1 зображено робота від Amazon Robotics [6].



Рисунок 1.1 - Робот від компанії Amazon Robotics

Роботи Ronavi від Ronavi Robotics неймовірно доступні. Це прямокутний робот, який виглядає майже ідентично роботі Amazon. Його функція - переміщення вантажів вагою до півтори тонни. Це можуть бути ті самі стелажі, які завантажуються на Amazon або будь-якому іншому складі. Робот рухається відповідно до позначок на підлозі. Але компанія зможе побудувати спеціального робота, який використовує машинне зір для навігації в безпілотних зонах. Однією з особливостей робота є те, що він має спеціальні колеса, які дозволяють йому повертатися на одному місці або рухатися вбік. Робот живиться від акумулятора, який може працювати протягом шести годин. Програмне

забезпечення, розроблене для системи керування, дозволяє керувати кількома роботами одночасно. На рисунку 1.2 зображено робота від Ronavi Robotics.



Рисунок 1.2 - Робот Ronavi

У початкових даних ми будемо використовувати набір роботів серії TETRIX MAX для створення моделі з дистанційним керуванням, як показано на рисунку 1.3, і супровідний контролер робота TETRIX PRIZM [7].

Набір складається з наступних компонентів:

- алюмінієві та металеві деталі та кріплення, які використовуються для складання рами платформи;
- монтажні інструменти;
- колеса для пересування;
- 2 електродвигуна для руху платформи;
- 2 сервоприводи;
- цифровий інфрачервоний датчик;
- ультразвуковий датчик відстані;
- акумулятор Ni-MH 12 В;

- довжина: 134 мм
- передаточне число: 60:1;
- діаметр: 37 мм;
- діаметр валу: 6 мм;
- крутний момент: 4,9 Нм;
- швидкість обертання: 100 об/хв.



Рисунок 1.4 - Електродвигун TorqueNADO

Акумуляторна батарея. На рисунку 1.5 показано акумуляторний блок TETRIX.



Рисунок 1.5 - Акумуляторна батарея TETRIX

Ni-MH (нікель-металгідридний) акумулятор, 12 В, 3 Ач. Він складається з десяти елементів батареї, з'єднаних разом. Для безпеки вбудований змінний запобіжник на 20 А.

Технічні характеристики:

- вага: 580г
- розміри: 115×45×50 мм;
- напруга: 12В;
- ємність: 3Ah;
- номінальний запобіжник: 20 А.

Ультразвуковий датчик відстані. На рисунку 1.6 показано ультразвуковий датчик відстані TETRIX.



Рисунок 1.6 - Ультразвуковий датчик

Технічні характеристики:

- частота: 42 кГц;
- мінімальна відстань: 3 см;
- максимальна відстань: 400 см;
- напруга живлення: 5В.

Зарядний пристрій для акумулятора. На рисунку 1.7 показано зарядний пристрій TETRIX.



Рисунок 1.7 - Зарядний пристрій

Цей зарядний пристрій використовується для зарядки акумуляторів TETRIX на 12 вольт. Пристрій автоматично визначає напругу батареї, встановлює правильну напругу зарядки та автоматично вимикається, коли батарея повністю заряджена. Він також захищає батарею від перевантаження по струму, короткого замикання і зворотної полярності.

Технічні характеристики:

- вхідна напруга: АС 100-240V, 50/60Hz;
- вхідний струм: 0,6 А;
- вихідна напруга: постійна 16В;
- вихідний струм: 0.9/1.8А;

Контролер роботи TETRIX PRIZM. На рисунку 1.8 показано контролер TETRIX PRISM.

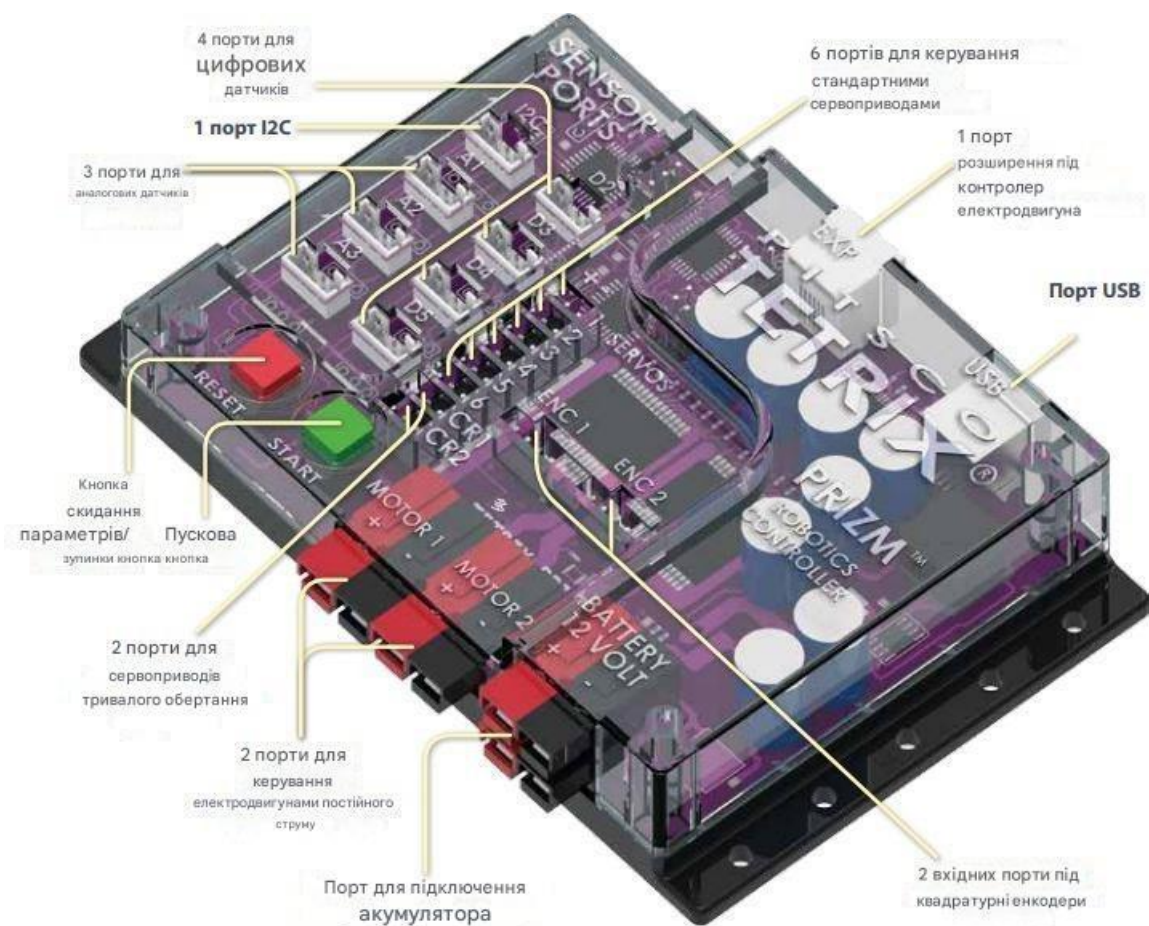


Рисунок 1.8 - Зовнішній вигляд контролера і призначення його портів

Сенсорний порт на контролері PRIZM. Контролер PRIZM використовує схему контактів, сумісну з Arduino UNO. Бібліотека мови Arduino для контролера PRIZM містить програмний код для датчиків, які можна автоматично налаштувати за допомогою функцій бібліотеки. Однак, якщо вам потрібно використовувати всі порти, функція компіляції програмного коду Arduino дозволяє використовувати їх безпосередньо. За винятком порту I2C, усі інші порти можна налаштувати як вхідні або вихідні сигнали за допомогою функції `pinMode` Arduino. На рисунку 1.9 показано один із портів, який використовується для підключення датчика до контролера. Таблиці 1.1-1.8 описують функції різних контактів усіх портів підключення датчиків контролера TETRIS PRIZM [8].

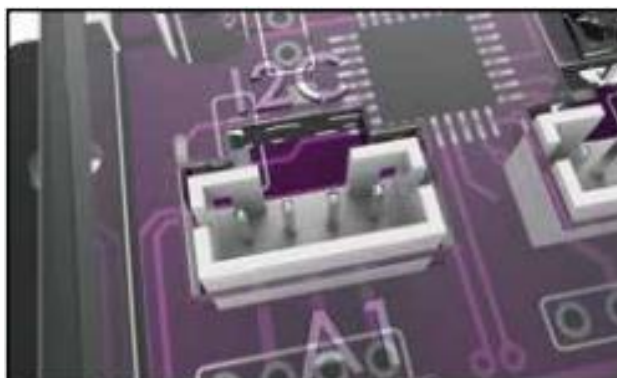


Рисунок 1.9 - Порт для датчиків на контролері PRIZM (Контакти зліва направо: 1, 2, 3, 4)

Таблиця 1.1 – Призначення контактів порту I2C

Номер контакту	Функція	Призначення контакту в ПЗ Arduino (IDE)
Контакт 1	Заземлення	Відсутня
Контакт 2	+5 В, 100 мА	Відсутня
Контакт 3	SDA (Послідовна передача даних через шину зв'язку I2C)	Канал введення ADC4 (A4)
Контакт 4	SCL (тактовий сигнал шини зв'язку I2C)	Канал введення ADC5 (A5)

Таблиця 1.2 - Порт для аналогових датчиків (A1)

Номер контакту	Функція	Призначення контакту в ПЗ Arduino (IDE)
Контакт 1	Заземлення	Відсутня
Контакт 2	+5 В, 100 мА	Відсутня
Контакт 3	Невикористовуваний	Відсутня
Контакт 4	Введення аналогових або введення-виведення цифрових сигналів	Введення аналогових сигналів (A1), введення-виведення цифрових сигналів (15)

Таблиця 1.3 - Порт для аналогових датчиків (А2)

Номер контакту	Функція	Призначення контактів в ПЗ Arduino (IDE) ()
Контакт 1	Заземлення	Відсутня
Контакт 2	+5 В, 100 мА	Відсутня
Контакт 3	Невикористовуваний	Відсутня
Контакт 4	Введення аналогових або введення-виведення цифрових сигналів	Введення аналогових сигналів (А2), введення-виведення цифрових сигналів (16)

Таблиця 1.4 - Порт для аналогових датчиків (А3)

Номер контакту	Функція	Призначення контактів в ПЗ Arduino (IDE) ()
Контакт 1	Заземлення	Відсутня
Контакт 2	+5 В, 100 мА	Відсутня
Контакт 3	Невикористовуваний	Відсутня
Контакт 4	Введення аналогових або введення-виведення цифрових сигналів	Введення аналогових сигналів (А3), введення-виведення цифрових сигналів (17)

послідовний пристрій керування передачею даних, у вікні якого відображаються значення даних, отримані від контролера PRIZM. Тут можна відобразити показання зі звичайних датчиків або кодерів, а при необхідності й інші типи програмних даних. Під час налагодження програмного коду здатність бачити значення даних часто буває корисною. Користувачі можуть створювати графічні інтерфейси за допомогою Java, Python або інших платформ програмування [9].

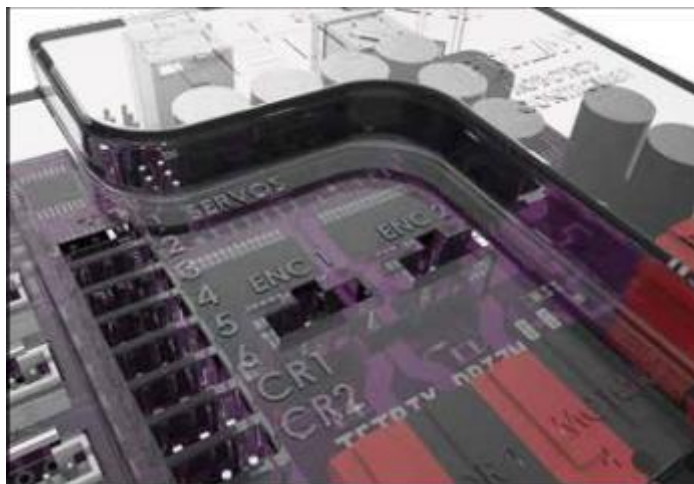


Рисунок 1.10 - Порти під квадратурні енкодери на контролері PRIZM - контакти зліва направо: 1, 2, 3, 4 (позначення: ENC1 та ENC2)

Кнопка скидання параметрів на контролері PRIZM.

Як показано на рисунку 1.12, червона кнопка скидання виконує дві функції. Натискання на неї зупиняє виконання будь-якого програмного коду та повертає всі показання кодера та інших датчиків до початкового стану. Натискання цієї кнопки перезавантажить всю систему так само, як і вимикання та ввімкнення живлення.

Кнопка запуску на контролері PRIZM. Зелена кнопка «Пуск», показана на рисунку 1.13, використовується для запуску програмного коду, завантаженого в контролер PRIZM. Коли ви створюєте код за допомогою Arduino IDE, функція PrizmBegin виконує всі команди, необхідні для запуску програми. Виклик цієї

функції встановлює необхідні параметри у відповідні стани та затримує виконання програмного коду до натискання зеленої кнопки «Пуск».



Рисунок 1.11 - Порт USB на контролері PRIZM, призначений для програмування та зв'язку



Рисунок 1.12 - Кнопка скидання параметрів на контролері PRIZM



Рисунок 1.13 - Пускова кнопка на контролері PRIZM

Порт двигуна постійного струму на контролері PRIZM.

Контролер PRIZM має два порти для підключення двигунів постійного струму. Вони позначені як Motor 1 і Motor 2. Кожен канал використовується для керування кутовою швидкістю та напрямком обертання двигуна постійного струму TETRIX за допомогою команд, визначених у Arduino (IDE) контролера

PRIZM. Кожен канал керування двигуном живиться від постійного струму 10 А та напруги 12 В. Кожен канал керування двигуном має червоно-чорний інтерфейс, де червона частина – це позитивний потенціал, а чорна частина – негативний потенціал. Кабелі живлення двигуна постійного струму TETRIX мають роз'єми відповідного кольору, які потрібно під'єднати до частини роз'єму того самого кольору, щоб усе працювало належним чином. На рисунку 1.14 показано порти, які використовуються для підключення двигуна постійного струму.

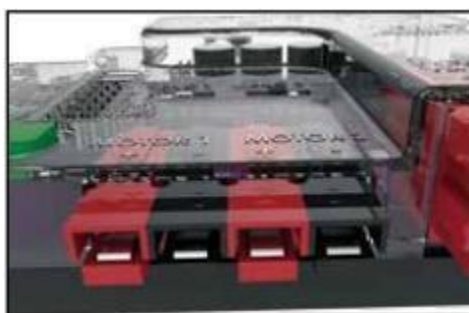


Рисунок 1.14 - Порти для електродвигунів постійного струму на контролері PRIZM

Порт підключення акумулятора на контролері PRIZM. Контролер PRIZM живиться від NiMH акумулятора серії TETRIX на 12 В. Контролер PRIZM поставляється з комплектом перемикача для підключення акумулятора до відповідного порту. Батареї можна підключати до верхнього або нижнього ряду роз'ємів, як показано на рисунку 1.15. Порт розширення використовується для підключення додаткових контролерів двигунів до контролера PRIZM і розподілу напруги до них через шлейфове з'єднання [10].

У таблиці 1.9 наведено технічні характеристики контролера TETRIX PRIZM.



Рисунок 1.15 - Порт введення/виводу напруги акумуляторної батареї контролера PRIZM

Таблиця 1.9 - Технічні характеристики контролера TETRIX PRIZM

Мікроконтролер:	Atmega328P з встановленим завантажувачем операційної системи Arduino Optiboot
Пам'ять:	програмоване ЕПЗУ обсягом 32 кб (Atmega328P)
Живлення напруга:	9... 18 У пост. струму
Порти для керування електродвигунами постійного струму:	2 з'єднувачі з роз'ємами Powerpole; бруківка схема управління при допомозі широтно-імпульсної модуляції; кожен канал розрахований на постійний струм силою 10 А максимальну навантаження по струму до 20 А
Рекомендований електродвигун:	Електродвигун постійного струму на 12 У TETRIX (39530)
Режими керування електродвигунами постійного струму:	Постійна потужність (від -100 % до 100 %) Постійна швидкість за допомогою ПІД-регулювання (від -720 до 720 градусів за секунду) ПІД-регулювання постійної швидкості для досягнення і утримання заданого кінцевого положення з урахуванням показань енкодера ПІД-регулювання постійної швидкості для досягнення та утримання заданого кінцевого положення з урахуванням показань енкодера в градусах Зупинка в режимі гальмування або вибігу Відстеження струму електродвигуна постійного струму (усі режими)

Кінець таблиці 1.9 - Технічні характеристики контролера TETRIX PRIZM

Порт розширення під додатковий контролер електродвигуна (EXP):	До цього порту за допомогою гірляндної схеми можна підключити додаткові модулі керування електродвигунами постійного струму і сервоприводами TETRIX.
Порт для підключення акумуляторної батареї:	підходить до роз'ємів Powerpole; додатковий порт для роздачі напруги акумуляторної батареї по гірляндний ланцюги контролерам електродвигунів, приєднаних до порту розширення.
1 зелена пускова кнопка (START)	Програмована кнопка
1 червона кнопка скидання параметрів/зупинки (RESET)	Непрограмована кнопка
1 червоний світлодіод:	Програмований світлодіод, службовець індикатором
1 зелений світлодіод:	Програмований світлодіод, службовець індикатором
1 синій світлодіод:	Вмикається на час подачі в ланцюг електроживлення

1.3 Висновки до першого розділу

У першому розділі було обґрунтовано актуальність впровадження роботизованих систем у складську логістику та розглянуто існуючі рішення в цій сфері. Проаналізовано приклади відомих компаній, зокрема Amazon Robotics та Ronavi Robotics, які продемонстрували ефективність автоматизації складських операцій за допомогою мобільних платформ. Визначено ключові переваги роботизованих систем: підвищення продуктивності, безпеки праці, а також зниження витрат і зменшення людського фактора. Представлено функціональні характеристики сучасних мобільних платформ і принципи їхньої навігації. Особливу увагу приділено розглядові конструктивних особливостей роботів, включаючи елементи шасі, типи приводу, енергетичне живлення та сенсорні

					КвРАКІТ.2021044.01.17 ПЗ	22
		№ докум.	Підпис			

системи. Визначено склад і можливості робототехнічного набору TETRIX MAX, який використовується для створення прототипу моделі з дистанційним керуванням. Окремо проаналізовано електродвигуни TorqueNADO, їх технічні параметри та можливості точної роботи в складі мобільної платформи. Надано детальний опис контролера TETRIX PRIZM, його портів, режимів роботи, інтерфейсів і програмних можливостей. Також розглянуто датчики, енкодери, елементи живлення та інші компоненти, необхідні для створення функціональної автономної платформи. У результаті опрацьованого матеріалу було зроблено висновок про доцільність вибору наявної компонентної бази для подальшої реалізації системи автономного мобільного робота.

					КвРАКІТ.2021044.01.17 ПЗ	
		№ докум.	Підпис			23

2 АПАРАТНА ЧАСТИНА

2.1 Розробка структурної схеми

Блок-схема (рисунок 2.1) розроблена на основі технічних специфікацій бакалаврської роботи. Для виконання завдання використано:

- мікроконтролер, який керує платформою;
- інфрачервоний датчик для відстеження чорних ліній;
- ультразвукові датчики для виявлення перешкод;
- електродвигуни для переміщення платформи;
- відображення налагодження та виведення інформації;
- платформу живить акумуляторна батарея.

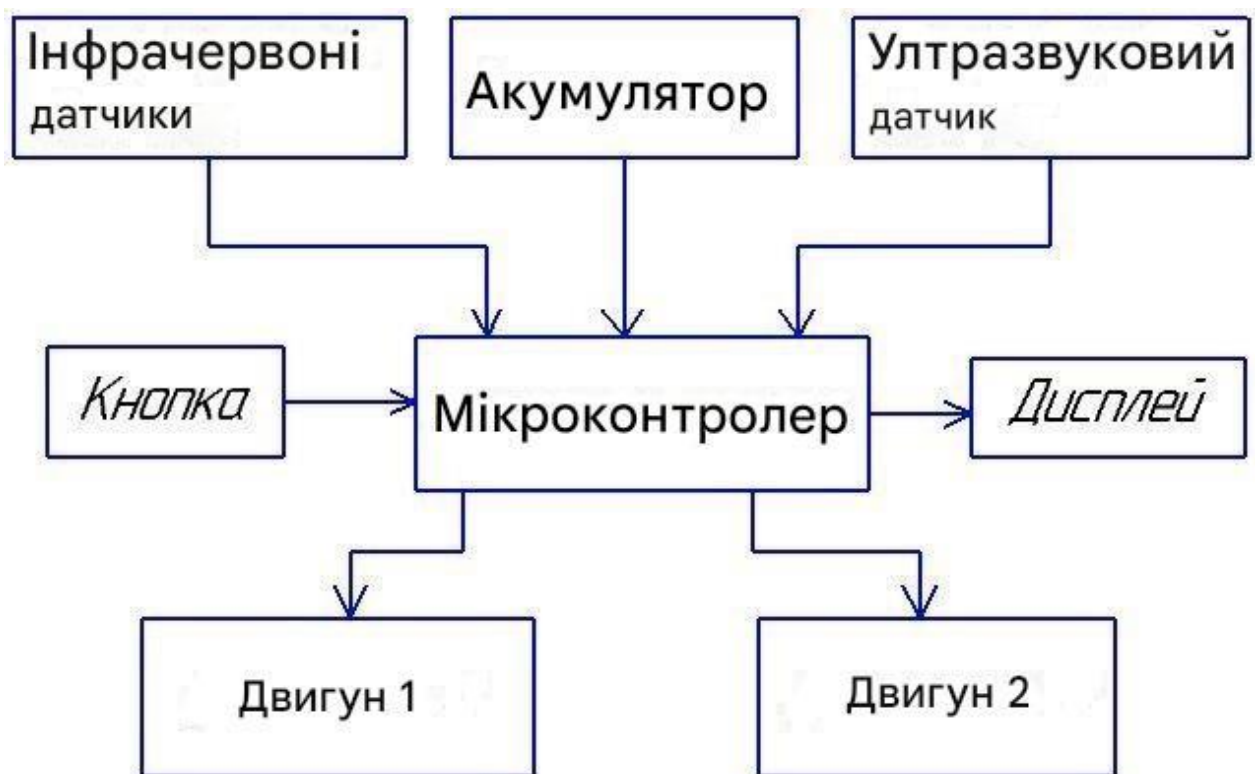


Рисунок 2.1 - Структурна схема платформи

2.2 Вибір необхідних компонентів

Відповідно до технічного завдання, ми використовуємо робот-набір для створення моделей дистанційного керування серії TETRIX MAX. При аналізі вихідних даних виявлено, що в комплект входять: корпус платформи, мікроконтролер, двигун, акумуляторна батарея з зарядним пристроєм, ультразвуковий датчик [11].

Відповідно до розробленої структурної схеми, для коректної роботи платформи нам не вистачає наступних елементів: інфрачервоного датчика, кнопок і дисплея.

Огляд, аналіз і вибір інфрачервоних датчиків.

За завданням платформа повинна була рухатися по чорній лінії, тому було вирішено використовувати інфрачервоний датчик. На рисунку 2.2 показаний один такий датчик.

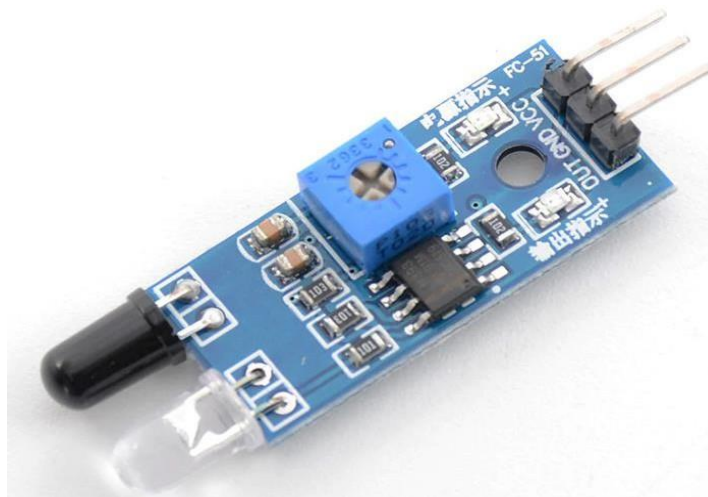
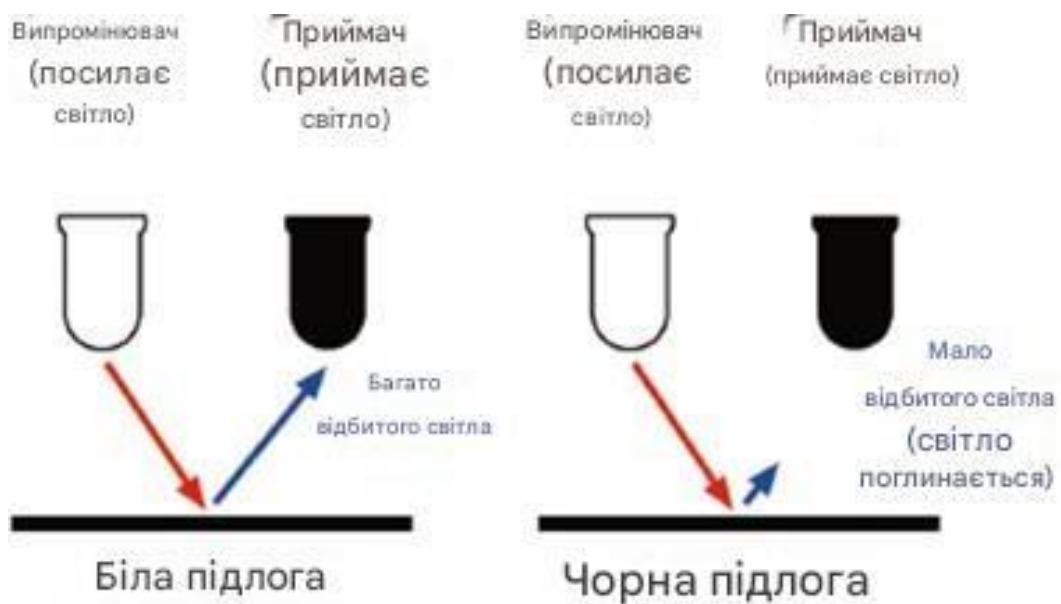


Рисунок 2.2 - Приклад інфрачервоного датчика лінії

Інфрачервоний датчик - це оптичний модуль, який використовується для виявлення перешкод у вигляді білих або чорних ліній. Його основними

компонентами є оптичні з'єднувачі, що складаються з інфрачервоного світлодіода та компонента, який вловлює інфрачервоне світло. Це може бути фотодіод, фототранзистор або фоторезистор. Інфрачервоні діоди випромінюють світло з довжиною хвилі 940-950 нм і можуть надійно виявляти перешкоди на різних відстанях, залежно від типу використовуваного датчика та оптрона.

Принцип роботи датчика показаний на рисунку 2.3 і заснований на виявленні інфрачервоного світла, відбитого від поверхні.



Розпізнавання світлого та темного

Рисунок 2.3 - Принцип роботи оптопар

Інфрачервоні датчики можна розділити на дві категорії:

а) за типом вихідного сигналу:

- моделювання;
- цифровий;

б) за типом фототрубки, яка використовується як детектор інфрачервоного світла:

- на фотодіод;

– на фототранзистор.

Аналоговий лінійний датчик складається з оптрона та резисторів, які обмежують струм світлодіода та чутливої фототрубки. На рисунку 2.4 показаний приклад схеми аналогового датчика.

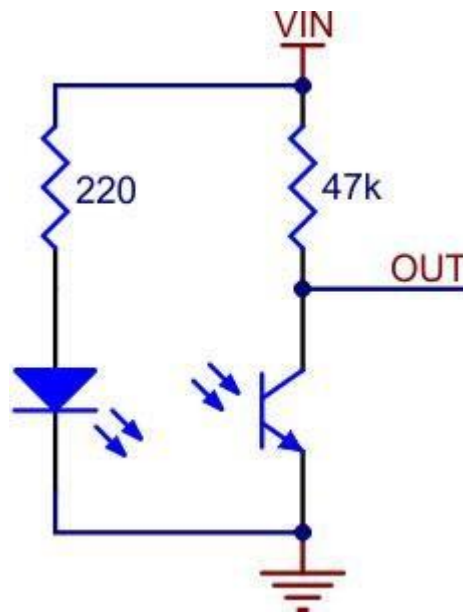


Рисунок 2.4 - Одна з можливих схем аналогового датчика

Аналогові датчики можуть не тільки розрізняти чорні та білі поверхні, а й різні відтінки сірого. Це дає точний контроль над переходом рамки від чорного до білого і навпаки.

При включенні сенсорного модуля на робочу поверхню направляється промінь довгохвильового інфрачервоного світлодіодного світла з довжиною хвилі 940-950 нм. Натомість промінь відбивається від поверхні та потрапляє на фотоелемент, розташований поруч із інфрачервоним світлодіодом. На виході отримуємо аналоговий сигнал. Чим яскравіша поверхня, тим сильніший сигнал. Чим темніша поверхня, тим слабший сигнал. Але можливо і навпаки. Все залежить від конструкції схеми, тобто обраного фотоелемента і джерела сигналу. Якщо взяти діаграму на рисунку 2.4, фототранзистор NPN діє як фотоелемент. Аналоговий сигнал береться з колектора. Отже, чим темніша поверхня, тим

сигнал, на вихідному кінці з'явиться логічний нуль (напруга низького рівня).
Принцип роботи цифрового лінійного датчика показаний на рисунку 2.7.



Рисунок 2.6 - Цифровий інфрачервоний датчик лінії

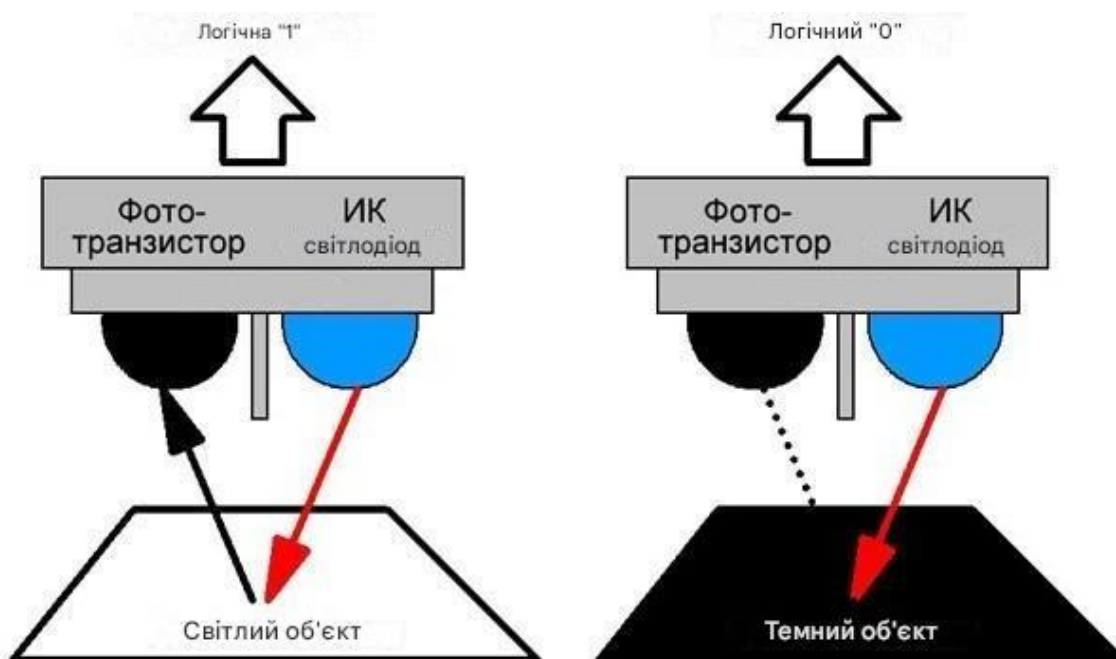


Рисунок 2.7 - Принцип роботи цифрового датчика лінії

За допомогою підстроювальних резисторів датчик можна відкалібрувати відповідно до певного сірого або будь-якого іншого кольору, а також

максимальної відстані. Для калібрування (регулювання чутливості) ручку тонкої настройки потрібно повернути в ту чи іншу сторону. Таким чином ви можете реагувати лише на найтемніші відтінки, або найсвітліші відтінки, або лише якщо колір перешкоди навпроти датчика залишається постійним - або лише якщо він знаходиться не ближче встановленої відстані.

Під час налаштування датчика ви можете стежити за світлодіодним індикатором, який загоряється, коли промінь приймається назад і його інтенсивність відповідає калібруванню. Якщо датчик розміщено занадто низько, прямий ІЧ-промінь буде виявлено передчасно та потрапить на перегородку безпосередньо між світлодіодом і фототранзистором, тому існує мінімальна відстань. Якщо датчик встановлений занадто далеко, промінь буде розсіюватися занадто рано і не досягне тильної сторони. Тому існує максимальна дистанція.

Згідно з технічними характеристиками, платформа повинна плавно слідувати за чорною лінією, розпізнавати різні повороти та перехрестя та проходити через них.

ПІД-регулювання буде використовуватися для досягнення плавного руху. Щоб виконати це завдання, нам потрібно змоделювати лінійний датчик. Цифрових лінійних датчиків достатньо для виявлення поворотів і перехресть.

Нижче представлені різні інфрачервоні датчики та модулі, які підходять для вирішення вищевказаних завдань, а також характеристики, зазначені виробником.

Лінійний інфрачервоний датчик KY-032 показаний на рисунку 2.8. Технічні характеристики наведені в таблиці 2.1.

Лінійний інфрачервоний датчик FC-51 показаний на рисунку 2.9. Технічні характеристики наведені в таблиці 2.2.

Таблиця 2.2 - Технічні характеристики датчика лінії FC-51

Режим роботи	цифровий
Напруга живлення, В	3.3 – 5
Дальність спрацювання, см	2 – 30

Інфрачервоний датчик серії TRT5000 показаний на рисунку 2.10.

Технічні характеристики наведені в таблиці 2.3.

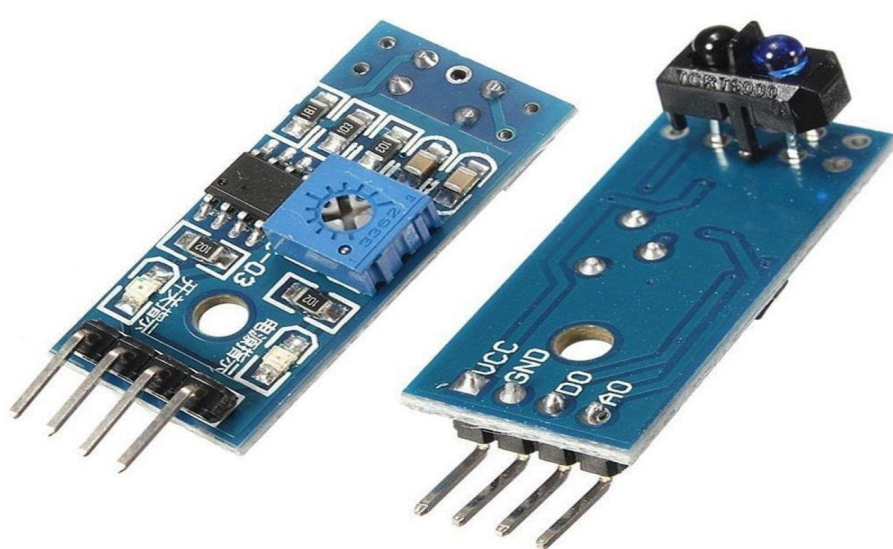


Рисунок 2.10 – Інфрачервоний датчик лінії TRT5000

Таблиця 2.3 - Технічні характеристики датчика лінії TRT5000

Режим роботи	аналоговий / цифровий
Напруга живлення, В	3.3 – 5
Струм споживання, мА	15
Дальність спрацювання, см	0.1 – 2.5

П'ятиканальний модуль TCRT5000 показаний на рисунку 2.11.

Технічні характеристики наведені в таблиці 2.4.

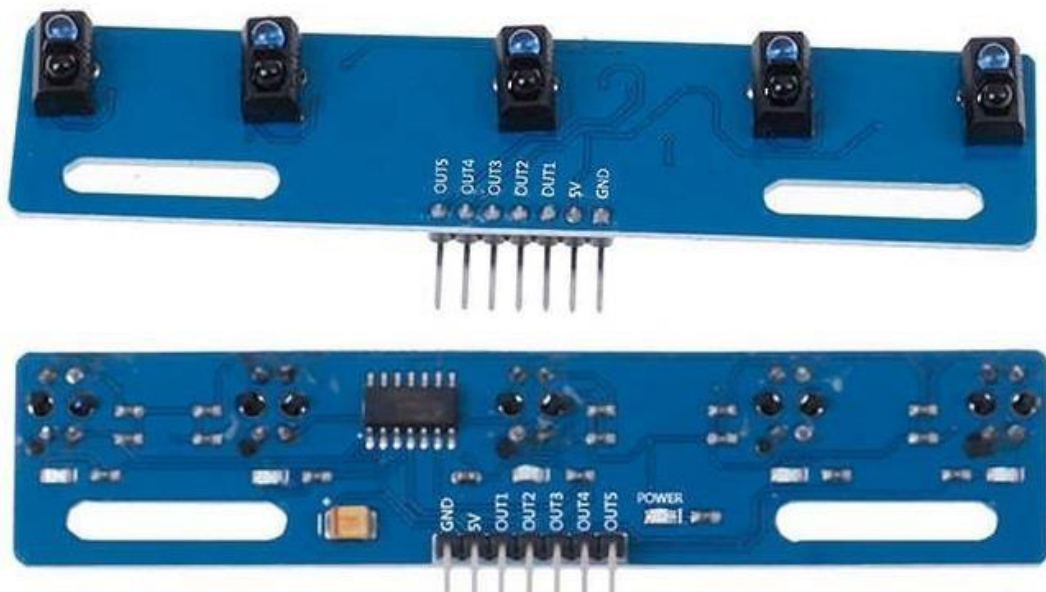


Рисунок 2.11 - П'ятиканальний модуль TCRT5000

Таблиця 2.4 – Технічні характеристики п'ятиканального модуля TCRT5000

Режим роботи	цифровий
Напруга живлення, В	3.3 – 5
Струм споживання, мА	15
Дальність спрацювання, см	0.1 – 2.5

На рисунку 2.12 зображено чотириканальний модуль YL-70 з датчиком YL-73. Технічні характеристики наведені в таблиці 2.5.

Таблиця 2.5 - Технічні характеристики чотириканального модуля YL- 70

Режим роботи	аналоговий / цифровий
Напруга живлення, В	3.3 – 5
Дальність спрацювання, см	2 – 7

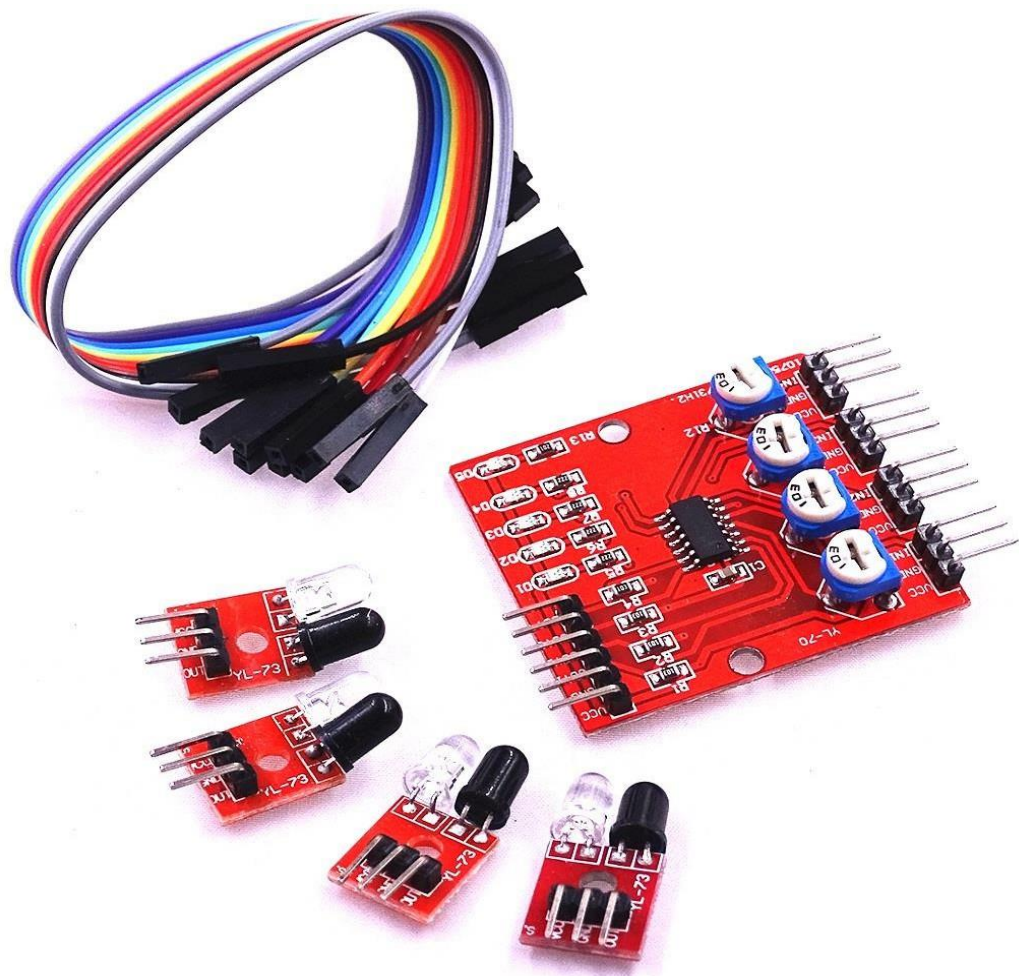


Рисунок 2.12 - Чотирьохканальний модуль YL-70 з датчиками YL- 73

Вісьмиканальний модуль МН показано на рисунку 2.13.

Технічні характеристики наведені в таблиці 2.6.

Таблиця 2.6 - Технічні характеристики вісьмиканальних модулів МГ

Режим роботи	аналоговий / цифровий
Напруга живлення, В	3.3 – 5
Дальність спрацювання, см	2 – 7

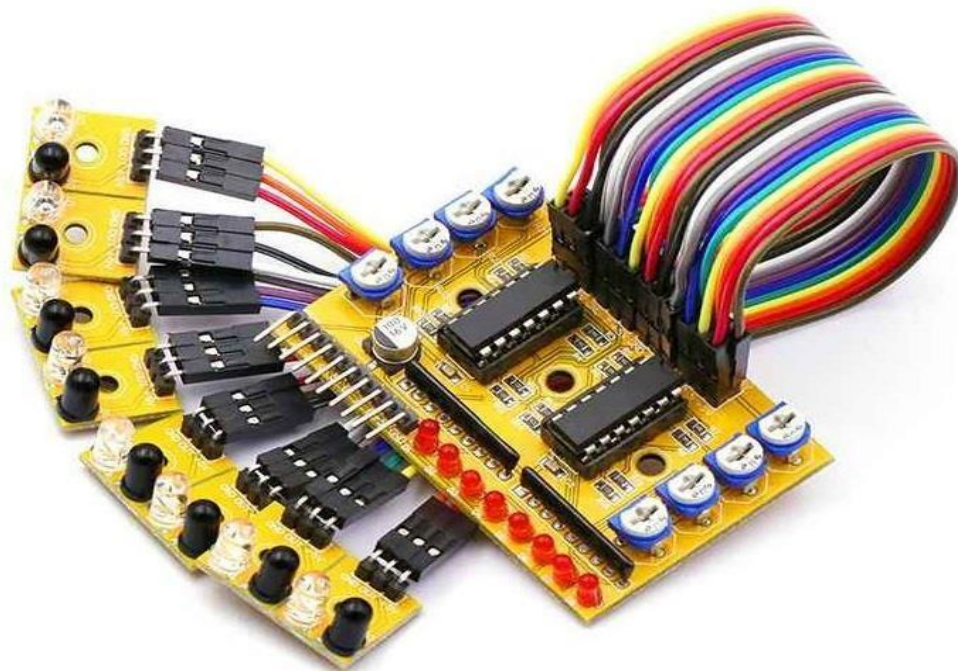


Рисунок 2.13 - Восьмиканальний модуль МН

З усіх представлених варіантів найбільший інтерес викликав чотириканальний модуль YL-70 і вісьмиканальний модуль МН [12].

Модуль YL-70 складається з чотирьох аналогових датчиків серії YL-73 і електронної плати. На платі встановлена мікросхема LM339, яка є чотириканальним компаратором. З його допомогою можна використовувати аналогові датчики як цифрові. Плата також оснащена чотирма підстроювальними резисторами та чотирма світлодіодними індикаторами, які дозволяють конфігурувати та калібрувати кожен окремий датчик до певного рівня запуску. Ще одна перевага цього модуля полягає в тому, що друкована плата всередині нього може розподіляти живлення на всі датчики одночасно.

Отже, якщо ми порівняємо цей модуль з усіма іншими датчиками, то для його підключення знадобиться менше проводів і контактів, які використовуються на платі керування платформою. При порівнянні YL-70 порівняно з п'ятиканальним модулем TCRT5000, останній матиме нижчу продуктивність через менший робочий діапазон. Крім того, в TCRT5000 всі

оптрони розташовані на одній платі електроніки і, на відміну від YL-70, їх неможливо розташувати в зручному порядку.

Вісьмиканальний модуль МН відрізняється від чотириканального модуля YL-70 лише тим, що може використовувати вісім датчиків одночасно. Але якщо ви придивитесь, то знайдете відмінності. Вони застосовні лише до аналогових датчиків, що входять до складу модуля. Вони мають різні номінали резистора для обмеження струму оптронної пари. У документації не вказано, який вони тип, тому можна припустити, що їхні оптрони також відрізняються.

Подальше порівняння вісьмиканального модуля МН з чотириканальним модулем YL-70 показало, що останній дуже чутливий до бічного «мерехтіння». Іншими словами, показання датчика можуть сильно відрізнятись в залежності від освітлення. У вісьмиканальному модулі МН показання датчика практично не змінюються від освітлення.

Після огляду та порівняння вищезазначених модулів і датчиків та їх характеристик було вирішено використовувати вісьмиканальний модуль МН у подальшому розвитку платформи.

Для відображення інформації було вирішено використовувати рідкокристалічний дисплей LCD 1602 з модулем I2C, як показано на рисунку 2.14, оскільки це найпопулярніший дисплей, а тому дешевший за відомі аналоги. Він має два рядки по шістнадцять символів у кожному.

Характеристики дисплея:

- тип відображення: персонаж з можливістю завантаження символів;
- контраст: регулюється потенціометром;
- напруга живлення: 5В;
- конвертер ПС/I2C: Є;
- інтерфейс: I2C;
- розміри: 80×36×12 мм;
- вага: 35 г. [13]

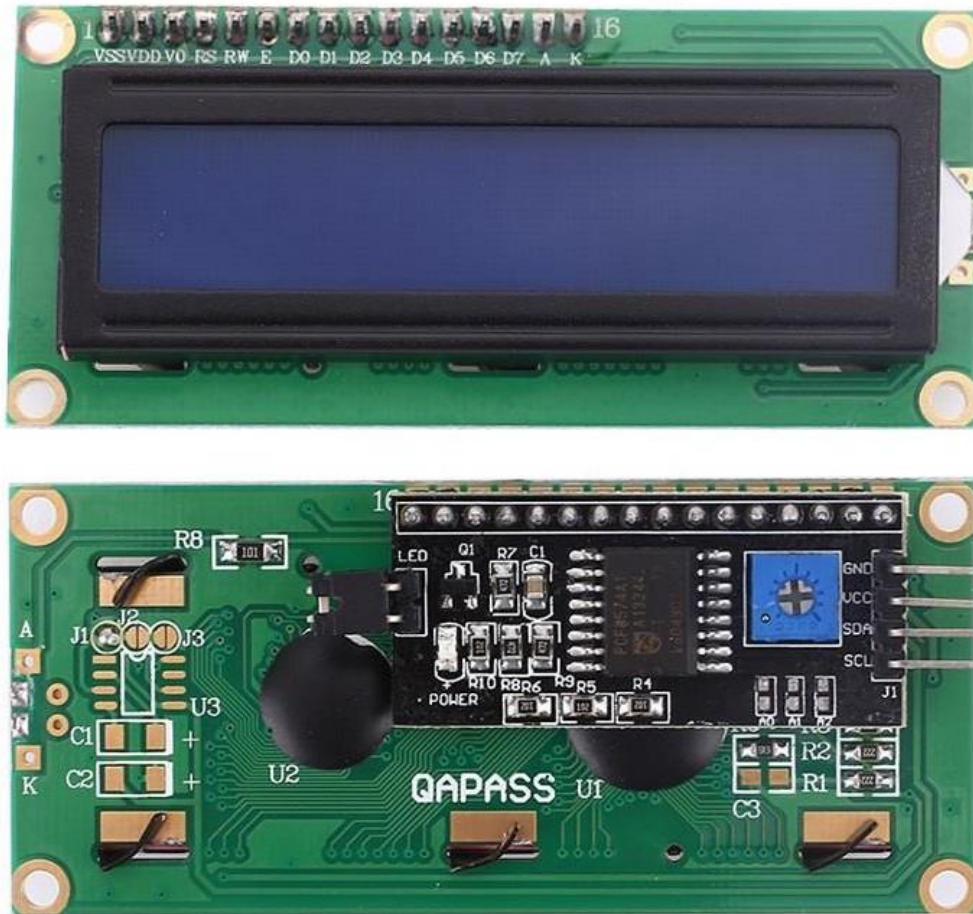


Рисунок 2.14 - Дисплей LCD 1602 з модулем I2C

Для керування платформою та регулювання її рухів під час руху було вирішено використовувати звичайні безфіксаційні тактильні кнопки, як показано на рисунку 2.15.



Рисунок 2.15 - Кнопка

2.3 Розробка електричної принципової схеми

На основі вихідних даних, конструктивних креслень і обраних вузлів була розроблена електрична схема. Схема електричного підключення показана на рисунку 2.16. Проектування схеми виконано в системі автоматизованого проектування КОМПАС 3D V16 [14-20].

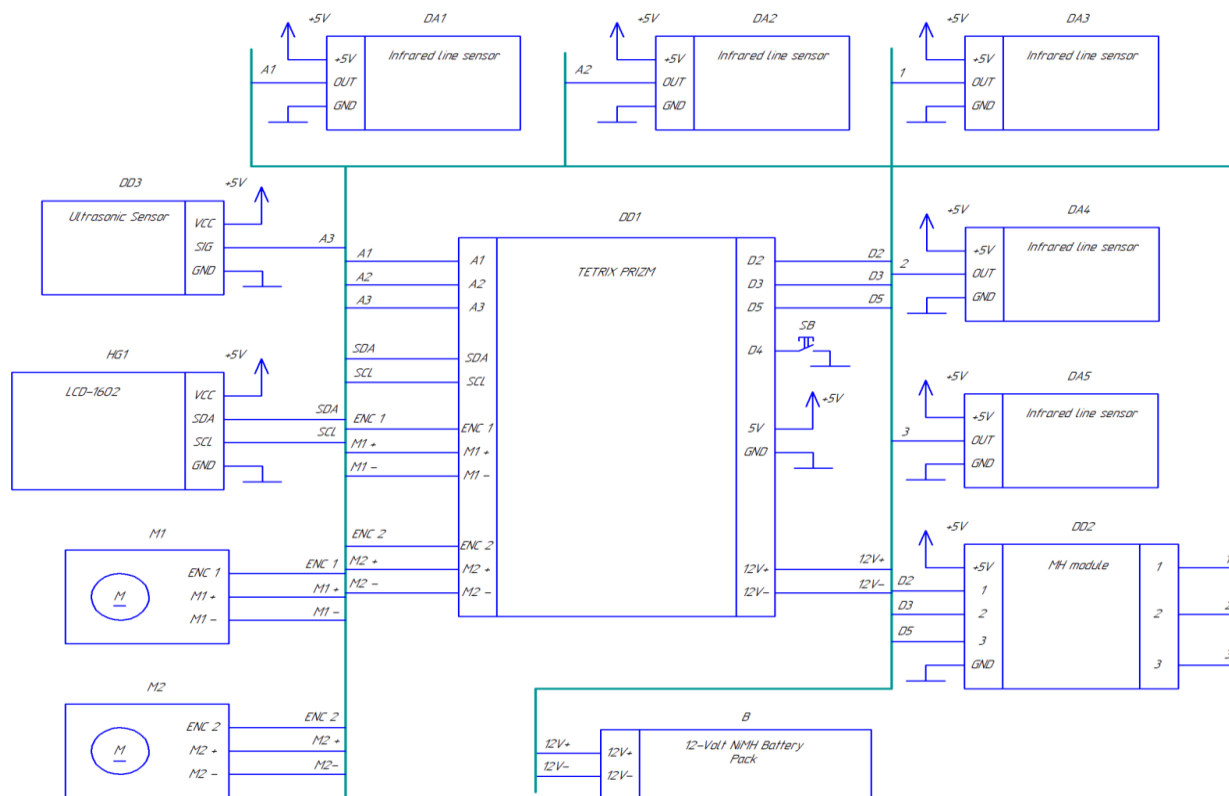


Рисунок 2.16 - Принципова схема автоматизованою мобільної платформи

2.4 Висновки до другого розділу

Таким чином, цей розділ присвячено розробці структурної схеми та принципової електричної схеми. Також було розглянуто фундаментальний принцип роботи інфрачервоних лінійних датчиків. Проведено процес відбору, порівняння та відбору датчиків для додаткового використання на платформі.

Для створення пристрою можна використовувати вісім датчиків, які працюють в аналоговому режимі.

– датчики не пов'язані один з одним і можуть розташовуватися в різному порядку;

– зменшити кількість проводів, необхідних для підключення;

– датчики не сприйнятливі до сонячного світла.

					КвРАКІТ.2021044.01.17 ПЗ	
		№ докум.	Підпис			39

3 ПРОГРАМНА ЧАСТИНА

3.1 Розробка алгоритму роботи

Згідно з технічними умовами, платформа повинна плавно рухатися по чорному маршруту, визначати перехрестя з пологими поворотами та проїжджати їх відповідно до маршруту, а також зупинятися біля певної розмітки та відображати на дисплеї певні написи.

Сама розмітка складатиметься з чорних ліній, що йдуть перпендикулярно низу злітно-посадкової смуги. Тільки після натискання кнопки платформа почне рухатися після зупинки біля маркера.

У той же час платформа повинна контролювати простір попереду, щоб зупинитися і запобігти зіткненню в разі появи несподіваної перешкоди [21].

На рисунку 3.1 представлена блок-схема розробленого алгоритму.

Алгоритм, зображений на блок-схемах, описує логіку роботи автономної мобільної платформи (робота), яка рухається маршрутом, розпізнає перешкоди, реагує на перехрестя та виконує дії згідно з визначеним масивом команд. Розглянемо послідовність роботи докладно:

Алгоритм починається з читання показань ультразвукового датчика, який використовується для визначення наявності перешкоди перед платформою. Якщо перешкода виявлена — платформа зупиняється.

Якщо перешкоди немає, відбувається зчитування аналогових та цифрових датчиків для аналізу навколишнього середовища. Після цього система перевіряє, чи знаходиться платформа на перехресті. Якщо ні — платформа рухається далі по прямій (сценарій зображено в діаграмі 4), при цьому перевіряється наявність мітки. Якщо мітка присутня, платформа зупиняється, на дисплей виводиться повідомлення, і система чекає натискання кнопки для подальшого руху.

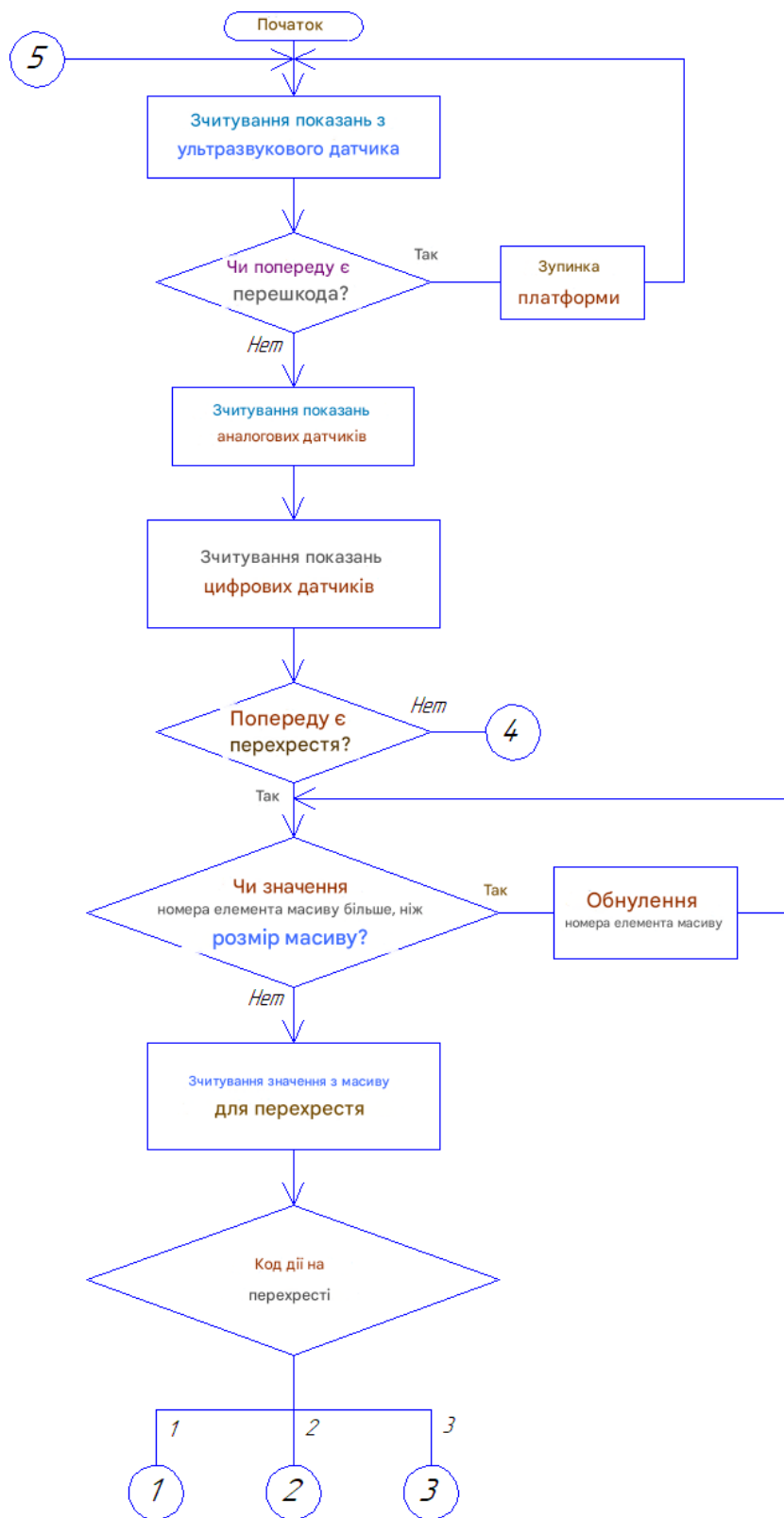


Рисунок 3.1 - Блок-схема алгоритму роботи платформи

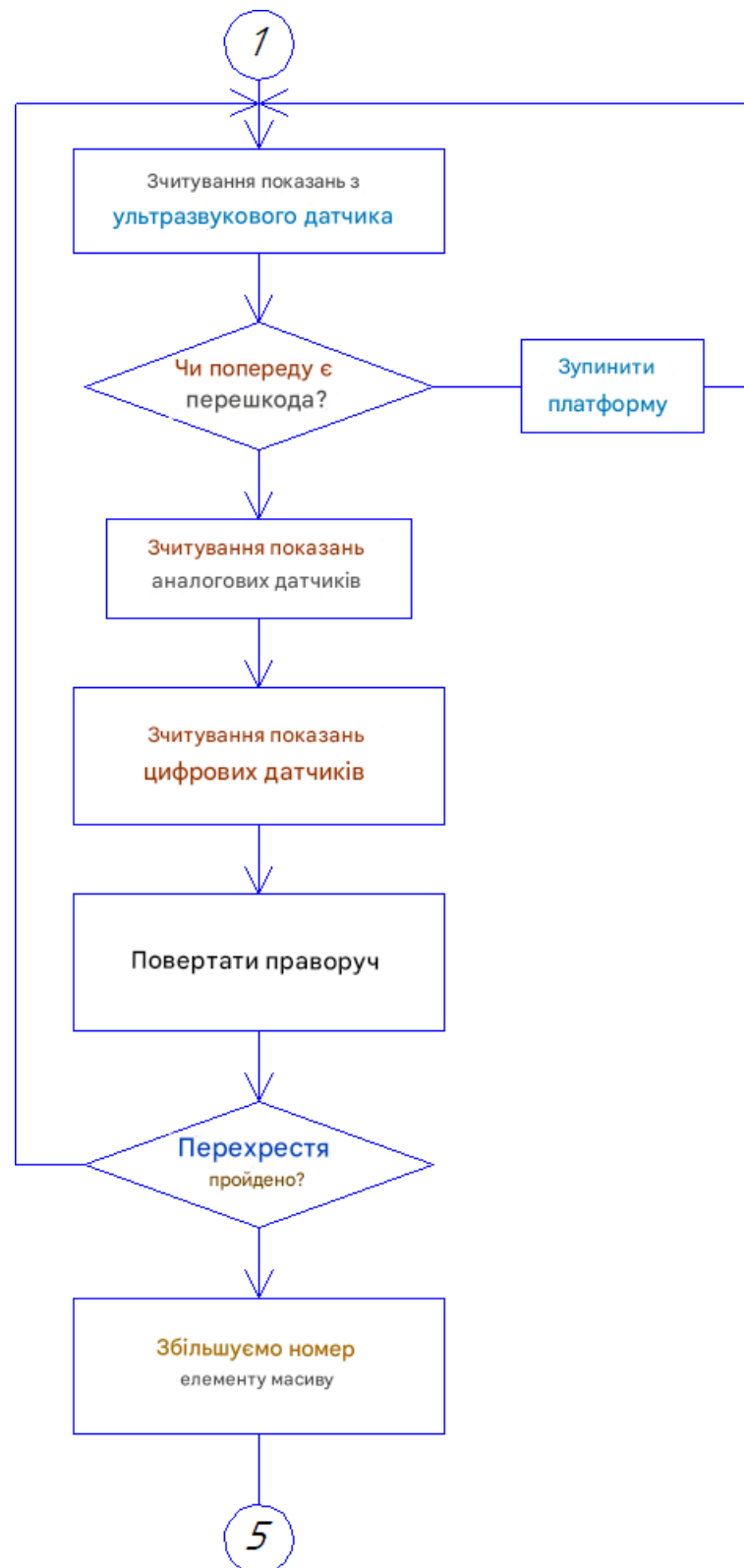


Рисунок 3.1 – Блок-схема алгоритму роботи платформи (Продовження)

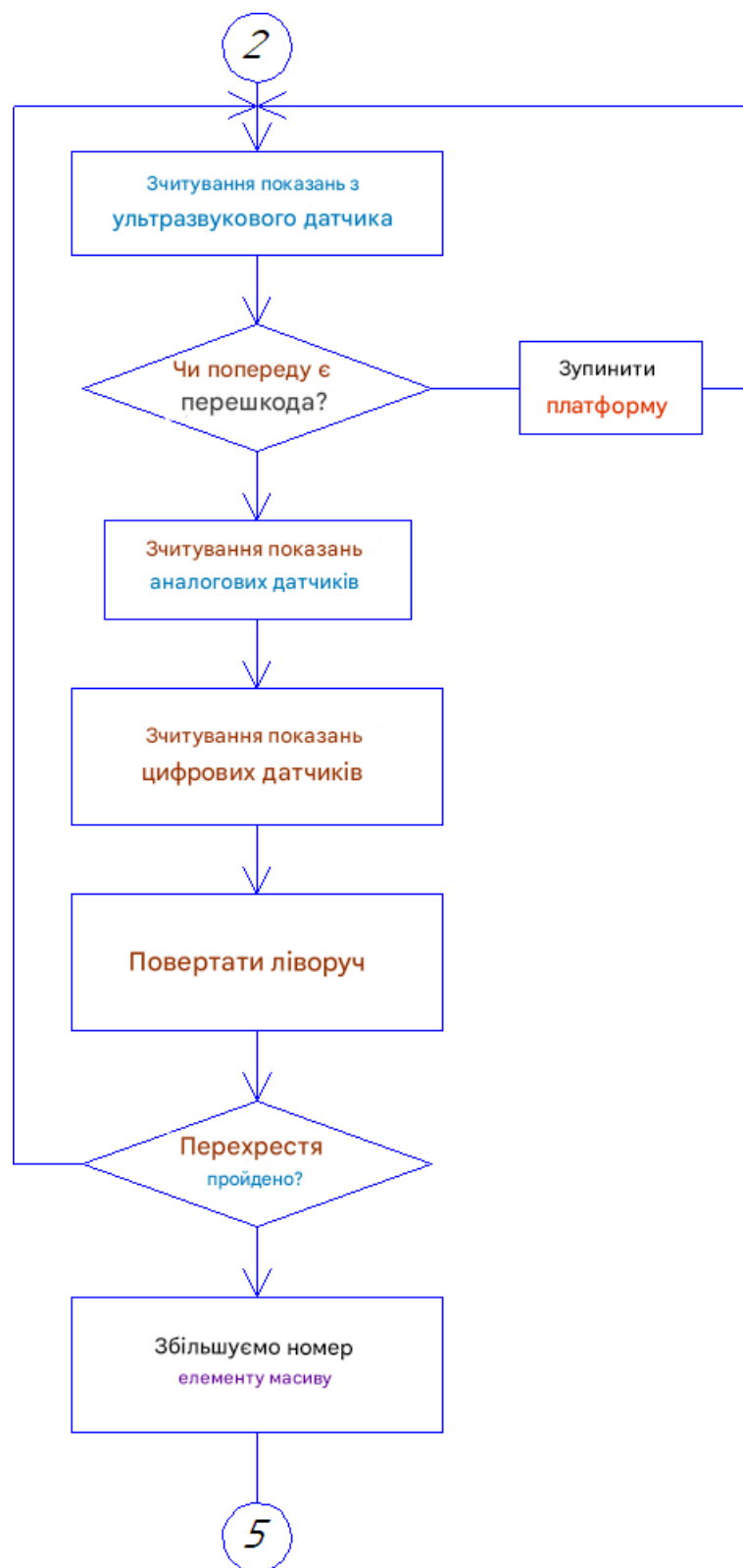


Рисунок 3.1 - Блок-схема алгоритму роботи платформи (Продовження)



Рисунок 3.1 - Блок-схема алгоритму роботи платформи (Продовження)



Рисунок 3.1 - Блок-схема алгоритму роботи платформи (Закінчення)

Якщо ж виявлено перехрестя, контролер перевіряє, чи не вичерпано масив команд, які задають дії на перехрестях. Якщо кінець масиву досягнуто — відбувається обнулення лічильника команд. Інакше — зчитується елемент з масиву команд, який задає напрямок дії на поточному перехресті (1 — поворот праворуч, 2 — поворот ліворуч, 3 — рух прямо). В залежності від значення команди платформа виконує відповідний підалгоритм (на окремих схемах 1, 2, 3).

У кожному з трьох підалгоритмів:

Платформа знову перевіряє перешкоди перед собою, а при їх наявності зупиняється.

Знову зчитуються дані аналогових та цифрових датчиків.

Виконується одна з дій: поворот праворуч, поворот ліворуч або прямий проїзд перехрестя.

Після проходження перехрестя лічильник команд у масиві збільшується, і цикл повторюється (повернення до головного циклу з міткою 5).

Таким чином, алгоритм є циклічним та підтримує багатоступеневе переміщення мобільної платформи через перехрестя з можливістю адаптації руху в залежності від перешкод, міток і попередньо запрограмованого маршруту. Така структура дозволяє системі гнучко реагувати на змінні умови, забезпечуючи безпечне й ефективне пересування [22].

3.2 Розробка програмної частини пристрої

Контролер PRIZM підключається до комп'ютера за допомогою кабелю USB. Напруга забезпечується зовнішньою акумуляторною батареєю NiMH TETRIX з номінальною напругою 12 В. Контролер оснащений подвійною системою приводу двигуна постійного струму з великим струмом, кожна з яких підтримує квадратурний кодер для точного ПІД-контролю швидкості та положення вала двигуна постійного струму. Крім того, є шість стандартних сервопортів і два довгих обертових сервопорту. Для зовнішніх датчиків є чотири цифрові порти, два аналогових порти та один порт I2C. Цифровий порт D2 також можна налаштувати для послідовної передачі даних за допомогою відповідної бібліотеки, включеної в програмне забезпечення Arduino. Цифрові датчики можуть бути налаштовані як вхідні або вихідні сигнали. Порти аналогового датчика можна налаштувати для введення аналогових сигналів і виведення цифрових сигналів. - По одній червоній і зеленій для візуальної сигналізації. Управління функціями контролера PRIZM здійснюється мікроконтролером Atmega328P. Завантажувач Optiboot використовується для завантаження програмного коду в нього через роз'єм USB. Сервоконтролер. Ці мікроконтролери обмінюються сигналами з головним мікроконтролером обробки через комунікаційну шину I2C. Головний процесор не обтяжений цим і може вільно контролювати виконання програмного коду. Зелена кнопка Пуск

(Пуск) використовується для запуску програми, завантаженої в контролер. Червона кнопка скидання використовується для зупинки програми [23].

Контролер PRIZM побудований на схемі Arduino, тому він працює з Arduino-сумісними мовами програмування.

Інтегроване середовище розробки Arduino (IDE) — це програмне забезпечення з відкритим вихідним кодом, яке дозволяє писати керуючі програми та надсилати їх до приводів. Програмне забезпечення Arduino (IDE) є зручним для розробників інструментом для створення програм у текстовому форматі та може використовуватися на різноманітних пристроях під керуванням операційних систем Windows, Macintosh та Linux. Це робить програмне забезпечення Arduino (IDE) дуже універсальним. Спілкуйтеся з PRIZM з Arduino (IDE) за допомогою мови програмування на основі C. В Arduino IDE окрема програма називається скетчем.

Обрано використовувати бібліотеку для програмування контролера платформи та керування самою платформою. Бібліотека — це файл (набір файлів), який містить той самий код, для якого ми пишемо ескізи. Можна підключити бібліотеку до нашого коду та використовувати функціональність, яку вона надає, і є багато варіантів: готові «інструменти» для роботи із зовнішніми датчиками та модулями, готові «інструменти» для роботи з внутрішніми периферійними пристроями мікроконтролера (таймери, АЦП, пам'ять), бібліотеки різноманітних математичних інструментів і так далі.

Перевага використання бібліотеки полягає в тому, що нам не потрібно знати, як працює код усередині; ми використовуємо готові інструменти, надані розробниками бібліотеки. Це економить час на розробку певних частин коду, а також звільняє пам'ять мікроконтролера. Як правило, бібліотеки постачаються з описом або документацією та прикладами використання.

Виробники серії робототехнічних наборів TETRIX MAX для створення моделей з дистанційним керуванням також розробили власну бібліотеку PRIZM для використання з контролерами, датчиками та двигунами.

Для використання дисплея ми також будемо використовувати бібліотеку LiquidCrystal_I2C. Дисплей буде підключено до мікроконтролера через інтерфейс I2C, тому для використання цього інтерфейсу нам знадобиться додаткова бібліотека Wire.

Завдання полягає в тому, щоб платформа рухалася плавно, дотримуючись чорної лінії. Для виконання цього завдання ми будемо використовувати пропорційне інтегрально-похідне керування, або скорочено ПІД-регулювання.

ПІД-регулятор створює керуючий сигнал, який є сумою трьох членів: пропорційного, інтегрального та похідної.

Пропорційний елемент P відповідає за пропорційне керування. Його принцип роботи полягає в тому, що вихідний сигнал регулятора компенсує зміну величини регулювання відносно певного заданого значення. Чим більша помилка невідповідності, тим більше відхилення команди контролера. Але недолік ланки пропорційного керування полягає в тому, що регулятор ніколи не може стабілізуватися на заданому значенні. Подальше збільшення коефіцієнта завжди призводить до автоколивань. Щоб уникнути цього, всі використовують зв'язки інтеграції та диференціації. Інтегральна складова I накопичує похибку регулювання і таким чином дозволяє ПІД-регулятору усувати статичні похибки. Іншими словами: процес інтеграції постійно вносить деяку упередженість, і система схильна до деяких постійних помилок. Але якщо ці похибки не існують або їх можна проігнорувати, то інтегральна складова сама вносить похибки зміщення. Найважливішим недоліком інтегральної ланки керування є ефект насичення інтегратора. Похідна складова D пропорційна швидкості зміни відхилення контрольованої змінної і призначена для компенсації відхилення

прогнозованого майбутнього від цільового значення. Слід зазначити, що похідна складова усуває затухаючі коливання.

Похідне керування особливо ефективно для процесів із великими затримками. Недоліком диференціального закону керування є його нестійкість через шум.

Розглянемо ПІД-регулювання на прикладі нашої платформи. Тут помилка $e(t)$ буде різницею між значеннями двох змодельованих інфрачервоних датчиків AL і AR, розташованих по обидві сторони лінії. Якщо значення датчика однакові, то контрольне значення u дорівнює нулю і платформа повинна рухатися прямолінійно. Якщо значення різні, то u буде відрізнятися від нуля, тому платформу потрібно буде повернути в якомусь напрямку. Щоб зчитати значення з датчика, ми скористаємося функцією `analogRead()`. Для керування двигуном ми будемо використовувати бібліотечну функцію `prizm.setMotorPower(motor#, потужність)`, яка використовується для регулювання рівня потужності та напрямку. Параметр потужності визначає потужність обертання валу двигуна, а діапазон налаштувань становить від 0 до 100. Це у відсотках, тобто 0 означає повне вимкнення, а 100 означає повну потужність. Крім того, існує два способи зупинки двигуна: зупинка накатом і зупинка на гальмування. Якщо потужність встановлена на «0», двигун буде працювати на холостому ході, а потім зупиниться. Якщо потужність встановлена на «125», двигун зупиниться після гальмування. Для пропорційного перенесення значення керуючого впливу з поточного діапазону значень до діапазону значень потужності двигуна скористаємося функцією `map()` [24].

Для виявлення перехресть використано три цифрові інфрачервоні датчики, розташовані, як показано на рисунку 3.2. Щоб прочитати значення датчика, використано функцію `digitalRead()`. Якщо всі три датчики спрацьовують одночасно, або одночасно спрацьовують середній і правий датчики, або середній

і лівий датчики, то перед нами буде позначка або перехрестя. Для створення такої умови буде використано умовний оператор if.



Рисунок 3.2 - Розташування інфрачервоних датчиків лінії

Для створення маршрутів платформи використовується масив, який вручну заповнюється певними значеннями. Крім того, коли буде виявлено перехрестя, платформа отримує доступ до масиву та побачить, які дії потрібно виконати. Щоб виконати це завдання, використовується оператор switch. Він порівнює значення змінної масиву зі значенням, визначеним у операторі case. Коли значення оператора case дорівнює значенню змінної, виконується

програмний код оператора. Нижче наведено приклад програмної реалізації оператора switch [25]:

```
switch (var) { Випадок 1:  
//Якщо var дорівнює 1, виконати break;  
Випадок 2:  
//Якщо var дорівнює 2, виконати break;  
за замовчуванням:  
// Виконати, якщо альтернатива не вибрана  
// Необов'язковий за умовчанням  
}
```

Для вирішення задачі обходу плавних перехресть був розроблений наступний алгоритм. Припустимо, що потрібно повернути ліворуч. Рух припинився з датчиком аналогової лінії з правого боку і продовжився рух тільки з лівого боку. У цей час датчик праворуч відстежує, чи перестала платформа рухатися по прямій перед поворотом. У разі перетину подальший рух виконується знову за допомогою обох датчиків. Розрахунок для визначення значення помилки для керуючого впливу дещо відрізняється. Визначається середнє між онлайн- і офлайн-свідченнями лівого датчика і під час розрахунку похибки ми віднімаємо знайдене середнє з поточного показання датчика.

Щоб зупинитися перед перешкодою за допомогою ультразвукового датчика, використано бібліотечну функцію `prizm.readSonicSensorCM()`. Ця функція зчитує значення датчика та перетворює його на значення відстані в сантиметрах. Далі, використовуючи оператор `if`, створюється умова, за якої платформа має зупинитися на певній відстані [26-30].

3.3 Висновки до третього розділу

Можна зробити такі висновки:

– контролер має лише три аналогових виходи, які зайняті, тому подальша модернізація платформи може викликати труднощі;

– для керування платформою за допомогою ПД-регулятора було вирішено не використовувати інтегральний компонент, оскільки якби система була схильна до якоїсь постійної помилки, інтегральна ланка постійно вносила б певні відхилення, а отже, платформа поводитиметься некоректно на прямій частині маршруту;

– бібліотека програмування є дуже корисною та дозволяє дуже легко використовувати датчики та двигуни, керовані контролером.

					КВРАКІТ.2021044.01.17 ПЗ	52
		№ докум.	Підпис			

4 КОНСТРУКТОРСЬКО-ЕКСПЕРИМЕНТАЛЬНИЙ РОЗДІЛ

4.1 Складання пристрої

Відповідно до принципової електричної схеми відбувається збирання платформи, встановлення усіх датчиків та виконано підключення. Зібрана платформа, з різних ракурсів показана на рисунках 4.1-4.5 [31-35].

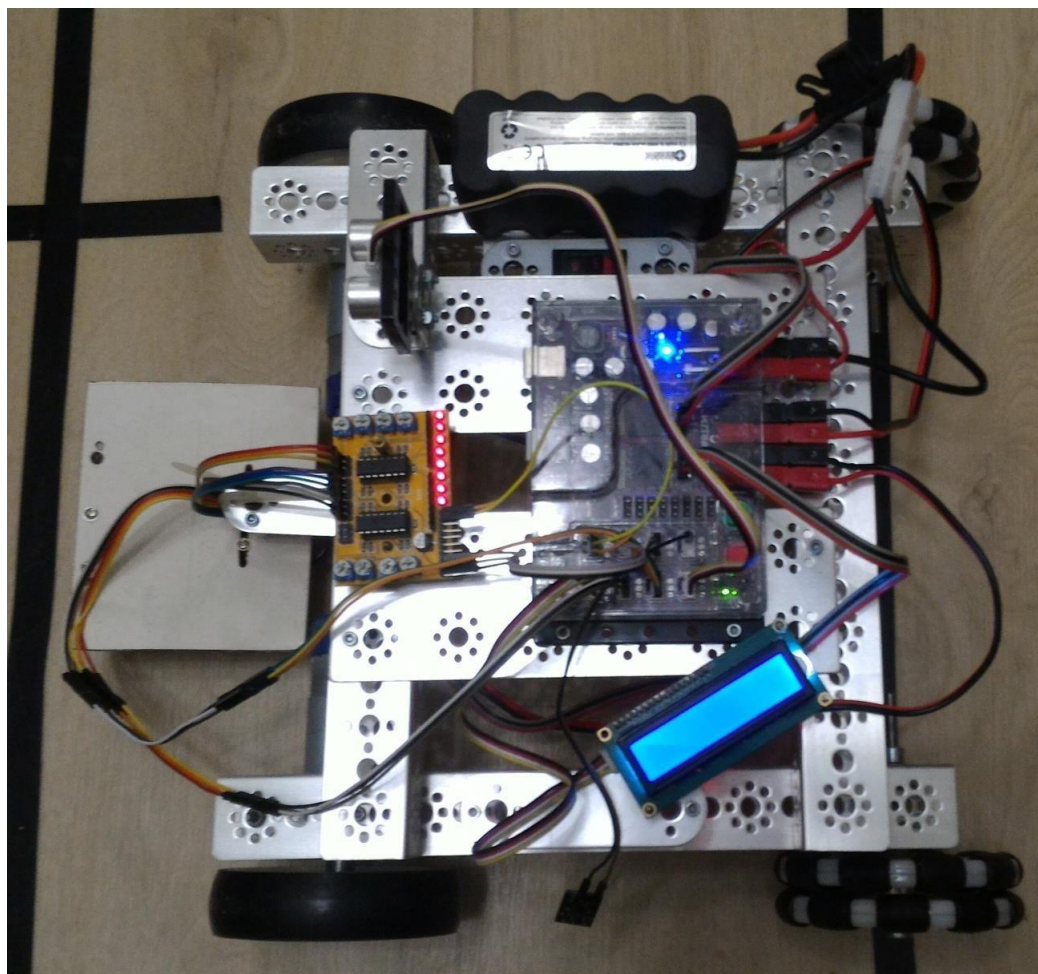


Рисунок 4.1 - Зібрана платформа, вид зверху

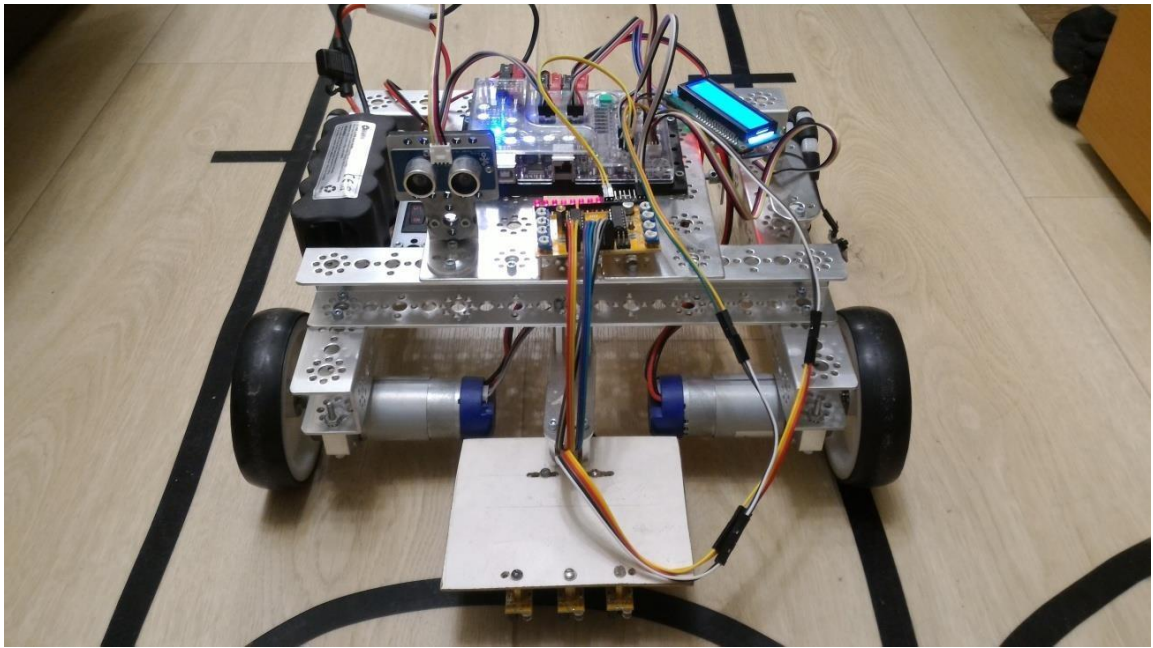


Рисунок 4.2 - Зібрана платформа, вид спереду

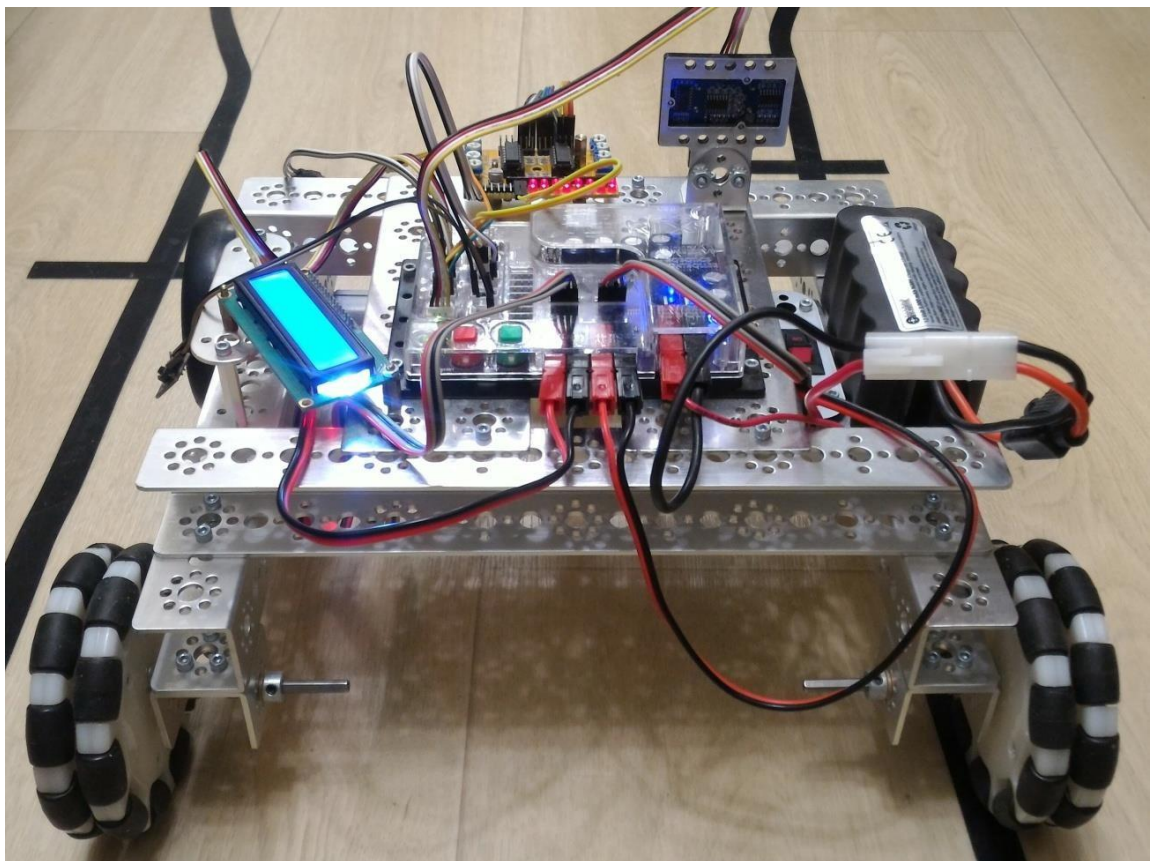


Рисунок 4.3 - Зібрана платформа, вид ззаду

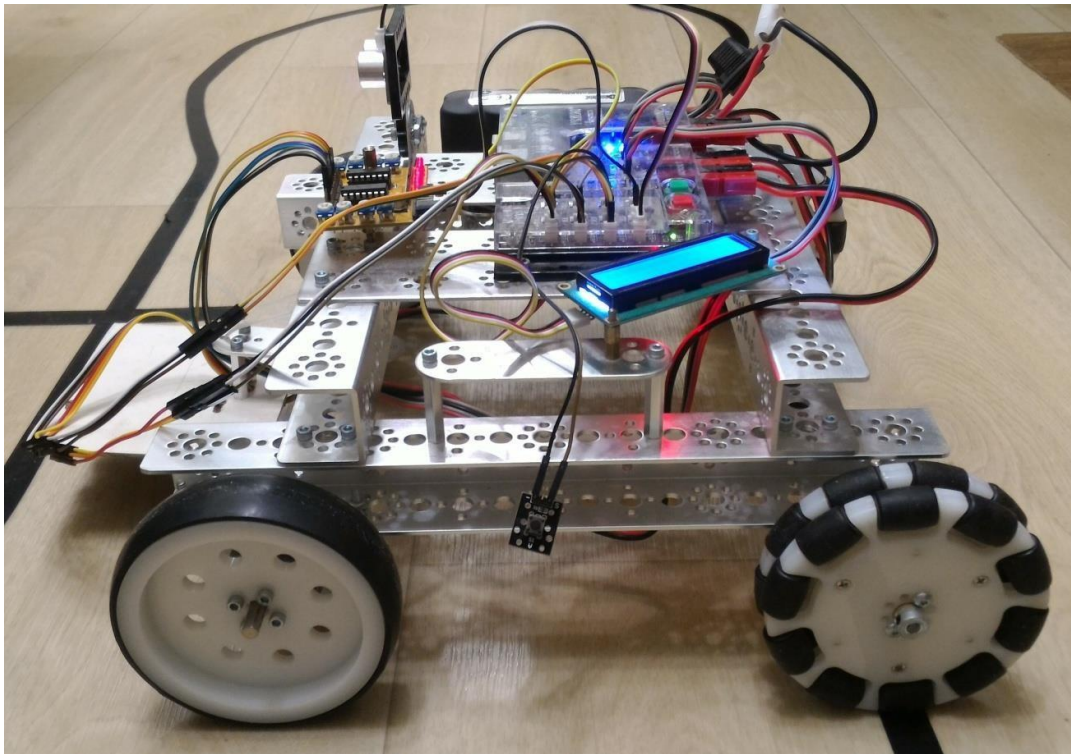


Рисунок 4.4 - Зібрана платформа, вид збоку

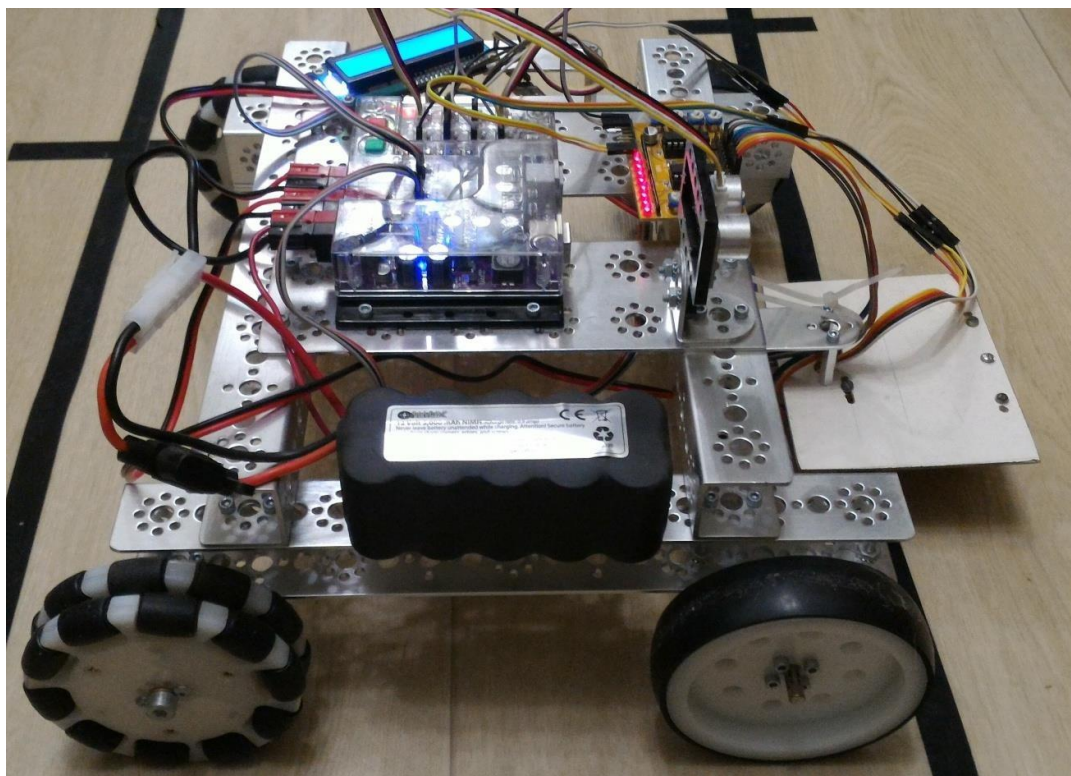


Рисунок 4.5 - Зібрана платформа, вид збоку

4.2 Налагодження пристроїв і експериментальні дослідження

По-перше, потрібно налаштувати ПД-регулятор, щоб досягти плавного руху платформи. Це можна зробити кількома способами, але використано один із них, який передбачає ручне послідовне налаштування кожного коефіцієнта. Спочатку обнуляються всі коефіцієнти, виводиться платформа в онлайн і запускається. Платформа, швидше за все, буде швидко рухатися по трасі. Збільшуючи пропорційний коефіцієнт, платформа може працювати стабільно без сходу з рейок. Далі за аналогією обрано два інших коефіцієнта, по черзі збільшуємо і зменшуємо їх для досягнення плавності руху платформи [36-40].

Після налаштування ПД-регулятора необхідно налаштувати 3 цифрових лінійних датчика. Це робиться за допомогою підстроювальних резисторів на вісьмиканальному модулі МН. Модуль МН має світлодіодний індикатор, за допомогою якого можна налаштувати датчик. Слід повертати потенціометр, поки світлодіод не вимкнеться. Це означає, що датчик бачить лінію. Після цього виконані описані вище операції з двома іншими датчиками. Процес налаштування показаний на рисунку 4.6.

Ці датчики використовуються для виявлення перехресть, тому також потрібно регулювати відстань між ними. Це зроблено для того, щоб коли платформа рухається в зоні без перехресть, обидва датчики не надсилали помилкових тривог одночасно. Якщо це станеться, платформа вважатиме, що попереду є перехрестя. Для цього додано до нашої програми блок коду, який контролює датчики та відображає повідомлення, якщо вони спрацьовують одночасно. Далі платформу ставиться на доріжку, запускається, а при одночасному спрацьовуванні збільшується відстань між датчиками і знову повторюються описані вище дії.

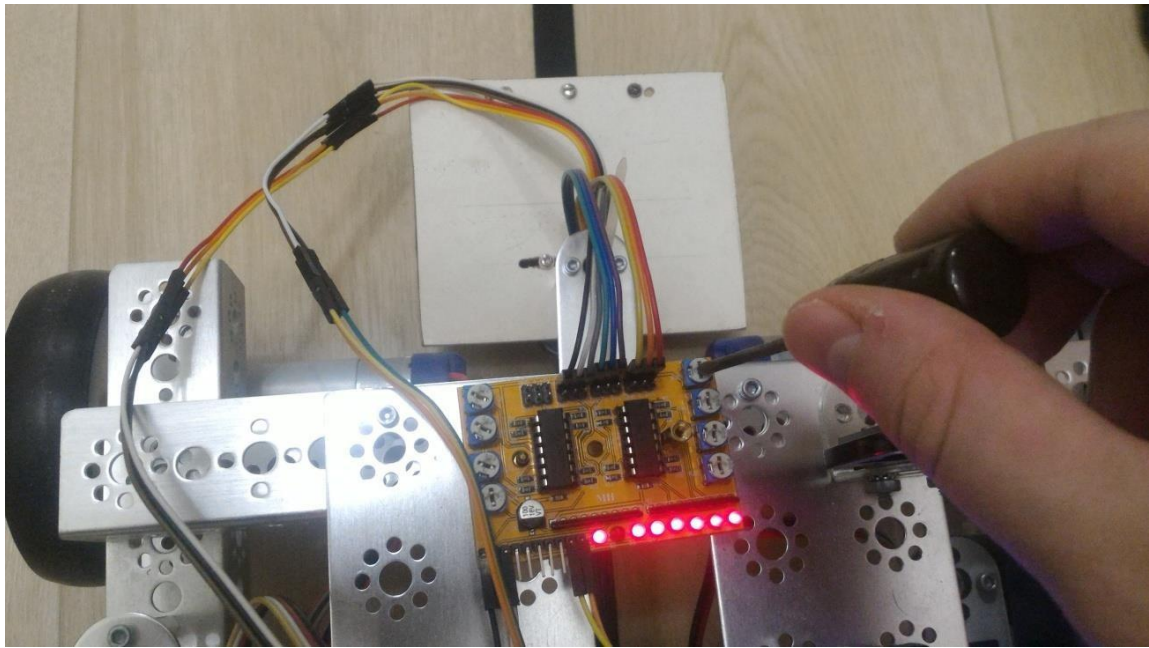


Рисунок 4.6 - Регулювання цифрових датчиків

Трек, показаний на рисунку 4.7, використовувався для повномасштабного тестування платформи.

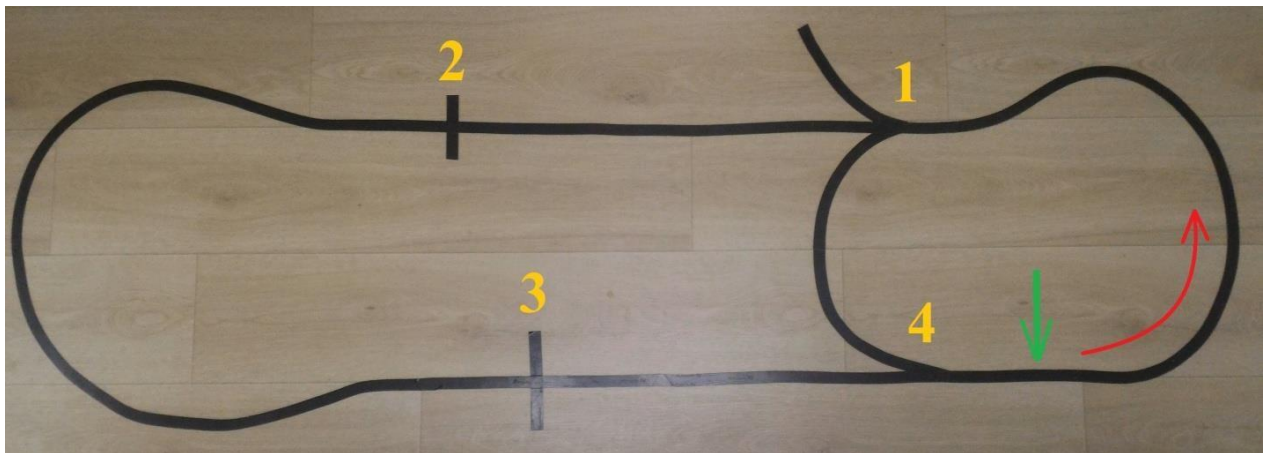


Рисунок 4.7 - Траса для випробувань платформи

Цифри вище позначають позначені перехрестя з пологими поворотами. Початок руху позначається зеленою стрілкою, а напрямок руху після його початку – червоною стрілкою.

Щоб пройти цим маршрутом, потрібно вручну заповнити послідовність дій для кожного перехрестя. Програма написана таким чином, що платформа не розрізняє мітки та перетини, тому для неї будь-яка гілка є перетином.

Маршрут руху наступний: їдемо через перше перехрестя прямо; на другому перехресті зупиняємося (це буде знак стоп) і чекаємо натискання кнопки для продовження; на третьому перехресті зупиняємося (це буде ще один знак стоп) і чекаємо натискання кнопки; на четвертий день їдемо прямо; потім на першому перехресті повернути ліворуч; на четвертому їдемо прямо. Платформа буде обертатися цим маршрутом.

У програмі масив заповнюється числами, що відповідають певним діям на перетині. У таблиці 4.1 наведено дії на перетині та відповідні масиви чисел.

Таблиця 4.1 – Дії перетину

Дія на перехресті	Відповідне число масиву
Прямо	0
Праворуч	1
Ліворуч	2
Зупинка	3

Знову ж таки, відбувається просто перерахування дій та відповідні номери на кожному перетині по порядку:

- прямо (число 0);
- зупинка (число3);
- зупинка (число3);
- прямо (число 0);
- ліворуч (No2);
- прямо (число 0).

Ось як виглядає заповнений масив у програмі для наведеного вище маршруту: `int action[] = {0, 3, 3, 0, 2, 0}`

З огляду на особливості програми, а також для більш правильного і плавного руху, на перехресті з одним поворотом (наприклад: перехрестя 4) рекомендується не їхати прямо, а вибрати поворот, протилежний повороту на перехресті. Рекомендується повертати праворуч і повертати ліворуч від першого перехрестя до четвертого перехрестя. `int action[] = {0, 3, 3, 1, 2, 2}`

Під час тестування виявилось, що коли платформа працювала належним чином і не відхилялася від колії, її максимальна швидкість становила 6 см/с або 0,06 м/с. Це тому, що зі збільшенням швидкості цифрові датчики лінії не встигають помітити перехрестя.

При виконанні плавного повороту мінімальний радіус цього повороту виявився 15 см. Під час руху платформа жодного разу не відхилилася від маршруту та правильно проїхала всі перехрестя та розмітку.

Під час тестів під навантаженням проблем не спостерігалось. Завдяки енкодеру двигуна контролер може підтримувати задану швидкість постійною при будь-якому навантаженні. Ємності акумулятора для двигуна більш ніж достатньо.

4.3 Висновок до четвертого розділу

У цьому розділі було здійснено збирання автономної платформи згідно з попередньо розробленою електричною схемою. Проведено монтаж усіх необхідних модулів, включно з датчиками, контролером та виконавчими елементами. Платформа була протестована в реальних умовах руху по підготовленій трасі з різними типами перехресть і знаків. Було налаштовано ПІД-регулятор для забезпечення плавного та стабільного руху, що значно зменшило ризик сходу з траєкторії. Особливу увагу приділено точному калібруванню

цифрових датчиків і коректному розміщенню їх на платформі, що дозволило уникнути хибних спрацювань.

Програмне забезпечення було адаптоване для обробки масиву дій на перехрестях, що дозволило створити гнучку навігаційну логіку. Було реалізовано обробку команд з масиву дій: поворотів, зупинок та прямолінійного руху. Під час експериментального запуску підтверджено, що всі функції платформи працюють відповідно до заданого алгоритму, включаючи реакцію на знаки «стоп» і точне дотримання маршруту. Досягнуто стабільної роботи при швидкості до 0,06 м/с, що відповідає вимогам надійності системи. Випробування підтвердили ефективність використаної апаратної платформи та обраної стратегії управління.

					КВРАКІТ.2021044.01.17 ПЗ	
		№ докум.	Підпис			60

ВИСНОВКИ

У ході виконання роботи було повністю реалізовано проєкт автоматизованої мобільної платформи, яка здатна автономно переміщуватися за заданим маршрутом, орієнтуючись по чорній лінії з використанням інфрачервоних датчиків. Проєкт об'єднав як теоретичні, так і практичні аспекти розробки систем управління мобільними роботами, що включає аналіз актуальності, вибір технічних засобів, моделювання алгоритмів, розробку програмного забезпечення, складання та налагодження обладнання.

На початку роботи було обґрунтовано актуальність застосування роботизованих платформ у логістичних та промислових задачах. Проведено детальний аналіз існуючих рішень (Amazon Robotics, Ronavi Robotics), що дозволило зрозуміти перспективи подальшого розвитку цієї галузі. Виявлено необхідність у створенні простих і доступних мобільних платформ для досліджень і навчання.

У другому розділі розроблено структурну схему пристрою, підібрано всі необхідні компоненти, зокрема контролер PRIZM, електродвигуни TorqueNADO, ультразвукові та інфрачервоні датчики, акумуляторну батарею та дисплей. Детально проаналізовано переваги та недоліки різних типів інфрачервоних датчиків, що дозволило обґрунтовано обрати восьмиканальний модуль МН як найбільш стабільний у роботі.

Важливим етапом була розробка принципової електричної схеми, що охоплює всі з'єднання між контролером і модулями. Завдяки ретельно продуманій схемі вдалося уникнути помилок у процесі збирання пристрою. Технічні параметри кожного компонента було ретельно підібрано відповідно до вимог до платформи.

У програмному розділі реалізовано алгоритм руху з використанням ПІД-регулятора. Було розроблено цикл опрацювання інформації від сенсорів,

					КвРАКІТ.2021044.01.17 ПЗ	61
		№ докум.	Підпис			

обробку команд масиву дій на перехрестях і контроль стану руху. Програма реалізована у середовищі Arduino IDE з використанням бібліотек PRIZM та LiquidCrystal_I2C, що забезпечило зручність керування та відображення інформації на дисплеї. Реалізація алгоритму у вигляді блок-схеми допомогла структуровано організувати логіку роботи пристрою.

У четвертому розділі зібрано реальний прототип платформи відповідно до електричної схеми, проведено налагодження ПІД-регулятора та калібрування цифрових датчиків. Під час експериментальних досліджень на підготовленій трасі перевірено точність роботи алгоритму та стабільність руху. За результатами тестування встановлено, що максимальна швидкість платформи складає 6 см/с, а мінімальний радіус повороту – 15 см.

Платформа впевнено проходила всі етапи маршруту, включаючи виявлення перехресть і зупинку за міткою. Система коректно обробляла кожен елемент масиву дій, забезпечуючи точне проходження заздалегідь визначеного маршруту. Під час роботи платформи не зафіксовано жодного критичного збою або відхилення від траєкторії.

Одним із важливих досягнень є стабільна робота системи за різних умов освітлення, що стало можливим завдяки ретельному калібруванню датчиків. Крім того, було враховано ефекти шумів та перешкод, які могли впливати на роботу сенсорів, та реалізовано програмні фільтри.

Успішне тестування довело ефективність використаної схемотехніки та програмної реалізації. Завдяки енкодерам двигунів платформа зберігала постійну швидкість навіть під навантаженням, а контролер PRIZM ефективно управляв всіма елементами системи.

Отриманий результат свідчить про практичну придатність розробленої платформи для задач внутрішньої логістики, навігації по маркованих маршрутах і навчального використання. Структура програми легко розширюється, що дозволяє інтегрувати нові функції, як-от Wi-Fi-керування або машинне зір.

Платформа має потенціал для модернізації. Зокрема, можливе додавання нових сенсорів, розширення пам'яті або створення більш складного маршруту. Використання ПД-регулювання дозволяє підтримувати високу точність і адаптивність платформи в реальному часі.

Таким чином, всі поставлені в роботі задачі були успішно виконані. Розроблена мобільна платформа поєднує простоту конструкції, ефективно управління та гнучку програмну архітектуру. Отримані знання і практичний досвід можуть бути використані в подальших наукових або інженерних проектах, пов'язаних з робототехнікою та автоматизацією.

					КВРАКІТ.2021044.01.17 ПЗ	
		№ докум.	Підпис			63

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Бабенко І. П. Розвиток робототехніки в промисловості України. Київ : Наукова думка, 2019. 256 с.
2. Петренко О. Л. Автоматизація складських процесів: український досвід : *Логістика і менеджмент*. № 2 (2021). С. 45–56.
3. Мельник В. С. Основи робототехніки для інженерів. Львів : Світ, 2020. — 312 с.
4. Іванюк А. Т. “Вплив роботизації на ринок праці” : *Економіка України*, № 7 (2022). С. 78–84.
5. Савчук Н. І. Тенденції промислової роботизації в Україні. Харків : Торсінг, 2021. 184 с.
6. Коваленко М. О. “Сучасна промислова революція та Україна” : *Промисловий вісник*, № 5 (2020). С. 32–44.
7. Müller T. et al. “Robotic trends in warehousing” : *EU Robotics Journal*, 2021. — pp. 112–129.
8. Schmidt H. *Automation in Modern Industry*. Berlin : Springer, 2018. 248 p.
9. Brown A. “Mobile robots in logistics” : *IEEE Trans. Robot.*, vol. 35, no. 4 (2019). pp. 1120–1130.
10. Smith J. *Warehouse Robotics and Automation*. New York : Wiley, 2020. 304 p.
11. Башук М. С. “Мобільні платформи в інтелектуальних складах” : *Автоматизація*, № 4 (2022). С. 50–61.
12. Кондратенко Ю. І. *Логістика 4.0: перспективи* Київ : КНЕУ, 2021. 220 с.
13. Олійник В. “Системи орієнтації для роботів” : *Системи і сенсори*, № 3 (2020). С. 9–18.

					КВРАКІТ.2021044.01.17 ПЗ	
		№ докум.	Підпис			64

14. Мартинюк Т. Маркування складів та навігація. Львів : Центр Європа, 2019. 272 с.
15. Зайцев О. Г. Технології ідентифікації на складах : Логістика України, № 2 (2021). С. 15–27.
16. Novak P. et al. “Marker-based navigation in European warehouses” : Robotics EU, 2022. pp. 88–102.
17. Rossi L. “Line-following robots review” : Italian Journal of Robotics, 2021. pp. 45–59.
18. Johnson L. “AGV systems in America” : Automation Today, vol. 12 (2020). pp. 75–84.
19. Garcia M. “Warehouse automation ROI analysis” : Logistics USA, 2019. pp. 63–72.
20. EVS – European Vehicle Standard. “AGV standards” : EVS Documentation, 2021. — 130 p.
21. Кузнецов В. П. Датчики лінії для робототехніки. Львів : Техніка, 2020. 198 с.
22. Савельєв А. В. Основи електроніки для роботів. Харків : Лабораторія, 2019. 350 с.
23. Литвиненко Д. М. “Вибір датчиків для автономних платформ” : Робоінженер, № 1 (2021). С. 22–34.
24. Романенко П. І. Принципові схеми мобільних платформ. Київ : Наукпоп, 2022. 216 с.
25. Мельничук Л. “Аналіз модулів TCRT5000 та YL-70” : РобоЖурнал, № 5 (2021). С. 40–53.
26. TETRIX PRIZM Robotics Controller Datasheet. Pitsco Education (USA), 2018. 45 p.
27. TorqueNADO DC Motor Datasheet. Pitsco Education (USA), 2017. 32 p.

					КВРАКІТ.2021044.01.17 ПЗ	
		№ докум.	Підпис			65

28. Bosch Sensortec. “Ultrasonic sensor documentation” (Germany), 2019. 28 p.
29. Arduino. “LiquidCrystal_I2C Library Reference.” Arduino.cc (USA), 2019. — 87 p.
30. European Commission. “Sensor standards in robotics” : EC Release, 2021. 140 p.
31. Іваненко С. Г. “PID-регулювання в мобільних роботах” : Системи управління, 2021. С. 95–105.
32. Кравченко О. В. Програмування Arduino для роботів. Львів : Центр Європа, 2020. 310 с.
33. Петросян А. “Алгоритми навігації на основі масиву команд” : РобоАрхітектура, № 3 (2022). С. 14–26.
34. Шевченко І. “Бібліотеки PRIZM і їх застосування” : РобоСофт, № 2 (2021). С. 30–42.
35. Arduino. “PRIZM API Reference.” Arduino.cc (USA), 2020. 102 p.
36. Schmidt H. Implementing PID in Embedded Systems. Berlin : Springer, 2018. 210 p.
37. Dupont J. “Finite state machines for robotics” : French Robotics Review, 2022. pp. 50–63.
38. Müller C. “Testing embedded code on mobile platforms” : EuroRobotics Conf. Proc., 2021. pp. 215–228.
39. NASA. “State machine design guide” : NASA website (USA), 2020. 95 p.
40. RobotLab GmbH. “Debugging embedded systems” : RobotLab Docs (Germany), 2019. 110 p.

ДОДАТОК А

Програма керування автоматизованою мобільною платформи

```
#include <GyverButton.h> #include <LiquidCrystal_I2C.h> #include <Wire.h>
LiquidCrystal_I2C lcd(0x3F, 16, 2); #include <PRIZM.h>
PRIZM prizm;

#define maxSpeed 21
#define L1 9 // крайній лівий датчик #define R1 2 // правий крайній датчик
int sensorWhite[2] = {40, 60}; // показання датчиків поза лінією в порядку A0, A1 ... int
sensorBlack[2] = {690, 690}; // свідчення датчиків на лінії в порядку A0, A1...

int action[] = {0, 3, 3, 1, 2, 2}; // масив для дії на перехресті int i = 0; //індекс для масиву
int d = 0;
//дії на перехрестях #define STATE_FORWARD      0
#define STATE_LEFT      2
#define STATE_RIGHT     1
#define STATE_PER       5
#define STATE_STOP      3
#define STATE_МЕТКА    4
GButton myButt(4, HIGH_PULL, NORM_OPEN);
// прототипи використовуваних функції void runForward();
void steerRight(); void steerLeft(); void steerStop(); void steerPer();

//ПІД

// Пропорційна
float up; // пропорційна складова
float kp = 1.7; // коефіцієнт пропорційною складовою

// Диференціальна
float ud; // диференційна складова float dif = 0;

float kd = 60; // коефіцієнт диференціальної складової
```

```

int oldErr; // значення помилки, отриманої на попередньої ітерації unsigned long time =
0; // час отримання поточної помилки
unsigned long pastTime = 0; // час отримання попередньої помилки

unsigned long time1 = 0; unsigned long lastTime1 = 0;

unsigned long time4 = 0; long lastTime4 = 0;

int err; // Різниця свідчень датчиків (помилка) int pause = 20;

int flag; int Left ;
int targetState = STATE_FORWARD;

68astime button_flag = 0; 68astime button;

68astime Dleft;
68astime Dright;
68astime Dfor;

int AL; int AR; int sonic; int Lmap; int Rmap; float u;
int gost = 0; int encod = 0; int fkl = 0;

void setup()
{

pinMode(17, INPUT); pinMode(2, INPUT); pinMode(5, INPUT); pinMode(3, INPUT);
pinMode(4, INPUT_PULLUP);

lcd.init(); lcd.backlight();

prizm.PrizmBegin(); prizm.setMotorInvert(1, 1); Serial.begin(9600);
}

void loop() { myButt.tick();

sonic = prizm.readSonicSensorCM(17); Dleft = digitalRead(2);

```

```
Drigh = digitalRead(5); Dfor = digitalRead(3); button = !digitalRead(4);
```

```
AL = analogRead(A1); AR = analogRead(A2);
```

```
Lmap = map(AL, sensorWhite[0], sensorBlack[0], 0, 100); Rmap = map(AR, sensorWhite[1],  
sensorBlack[1], 0, 100);
```

```
Lmap = constrain(Lmap, 0, 100);
```

```
Rmap = constrain(Rmap, 0, 100);
```

```
switch (gost)
```

```
{
```

```
case 0:
```

```
err = Lmap - Rmap; up = kp * err;
```

```
time = millis(); // поточний час
```

```
ud = kd * (err - 69asti) / (time - 69astime); // de/dt 69asti = err;
```

```
69astime = time;
```

```
u = up + ud; // керуюче вплив u = constrain (u, -200, 200);
```

```
Left = map(u, 0, 110, maxSpeed, 0);
```

```
Left = constrain(Left, -maxSpeed, maxSpeed); Right = map(u, 0, -110, maxSpeed, 0);
```

```
Right = constrain(Right, -maxSpeed, maxSpeed); break;
```

```
case 1:
```

```
err = 50 - Rmap; up = kp * err;
```

```
time = millis(); // поточний час
```

```
ud = kd * (err - 70asti) / (time - 70astime); // de/dt 70asti = err;
```

```
70astime = time;
```

```
u = up + ud; // керуюче вплив u = constrain (u, -200, 200);
```

```
Left = map(u, 0, 110, maxSpeed, 0);
```

```
Left = constrain(Left, -maxSpeed, maxSpeed); Right = map(u, 0, -110, maxSpeed, 0);
```

```
Right = constrain(Right, -maxSpeed, maxSpeed); break;
```

```
case 2:
```

```

err = Lmap - 50; up = kp * err;
time = millis(); // поточний час
ud = kd * (err - 70asti) / (time - 70astime); // de/dt 70asti = err;
70astime = time;
u = up + ud; // керуюче вплив u = constrain (u, -200, 200);
Left = map(u, 0, 65, maxSpeed, 0);
Left = constrain(Left, -maxSpeed, maxSpeed); Right = map(u, 0, -65, maxSpeed, 0);
Right = constrain(Right, -maxSpeed, maxSpeed);
break;
}

if ((Dleft && Dfor && flag == 0) || (Dright && Dfor && flag == 0) || (Dleft && Dfor &&
DRight && flag == 0))
{
flag = 4;
targetState = STATE_PER;
}

if (myButt.isPress() && flag == 4) { lcd.clear();
targetState = STATE_FORWARD; flag = 5;
}

if (i > 5) i = 0; if (Sonic < 30) {
prizm.setMotorPower(1, 125);

prizm.setMotorPower(2, 125);

} else {

switch (targetState)

{
case STATE_FORWARD: runForward(); break; case STATE_LEFT: steerLeft();
break;

```

```

        case STATE_RIGHT:          steerRight();          break;          case
STATE_STOP:          steerStop();          break; case STATE_PER:          steerPer();
        break; case STATE_METKA:          steerStop();
        break;
    }
}
}

```

```
void steerPer()
```

```
{
switch (action[i])
{
```

```
case 0:          runForward(); break;
```

```
case 1:          steerRight(); break;
```

```
case 2:          steerLeft(); break;
```

```
case 3:          steerStop(); break;
```

```
}
```

```
}
```

```
void runForward()
```

```
{
```

```
lcd.setCursor(0, 1); lcd.print("F");
```

```
gost = 0;
```

```
prizm.setMotorPower(1, Left); prizm.setMotorPower(2, Right);
```

```
if (flag == 4 && (DLeft == 0 && DFor) && (DRight == 0 && DFor)) { targetState =
STATE_FORWARD;
```

```
flag = 0; i++;
```

```
}
```

```
if (flag == 5) {
```

```
targetState = STATE_FORWARD; flag = 0;
```

```
i++;
```

```
}
```

```

    }
    void steerRight()
    {
        gost = 1;
        lcd.setCursor(0, 1); lcd.print("R"); prizm.setMotorPower(1, Left); prizm.setMotorPower(2,
Right);
        if (AL > 400 && d == 0) d++; if (AL < 100 && d == 1) {
            flag = 0;
            targetState = STATE_FORWARD; i++;
            d = 0;
        }
    }
    void steerLeft()
    {
        gost = 2;
        lcd.setCursor(0, 1); lcd.print("L"); Right); fkl = 1;

        if (fkl == 1 && d == 0) { d++;
            prizm.resetEncoders();
        }
        encod = prizm.readEncoderDegrees(2); if (encod > 360 && d == 1) {
            flag = 0;
            targetState = STATE_FORWARD; i++;
            fkl = 0; prizm.resetEncoders(); d = 0;
        }
    }

    void steerStop()
    {
        prizm.setMotorPower(1, 0);
        prizm.setMotorPower(2, 0);

        lcd.setCursor(4, 0); lcd.print("Please,"); lcd.setCursor(1, 1); lcd.print("press button!");
    }

```

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Сім'яченко Андрій Вікторович

Тема: Робот пилосос з автоматичним рухом

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

Обсяг кваліфікаційної роботи:

Кількість сторінок записки 66

1. Короткий зміст роботи та прийнятих рішень: Метою роботи є розробка схемотехнічного рішення роботу пилососу з автоматичним рухом
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 стану питання щодо типових рішень побудови робота пилососа. У другому розділі проведено апаратну розробку робота пилососа з автоматичним рухом. У третьому розділі розроблено програмне забезпечення для робота пилососа з автоматичним рухом.
4. Позитивні сторони роботи: висока практична цінність роботи.

5. Негативні сторони роботи: у роботі недостатньо уваги приділяється аналізу існуючих технічних рішень

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.

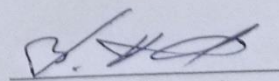
8. Інші зауваження: відсутні

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

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

Петрушко Володимир Степанович, доц. каф.
ТМІТ

" 19 " 06 2025 р.


(підпис)

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

Андрій СІМ'ЯЧЕНКО

ПІБ здобувача вищої освіти

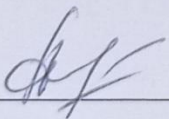
ФІТ, 4 курс, групи АКІТ-21-1

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.



дата



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: Андрій СІМ'ЯЧЕНКО

Співавтор:

Назва: Сім'яченко антиплагіат

Експерт:

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

Коефіцієнт подібності 1: 5.8%

Коефіцієнт подібності 2: 4.7%

Мікропробіли: 0

Заміна букв: 0

Інтервали: 0

Білі знаки: 0

Дата створення звіту: 2025-06-19 13:24:50.0

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 процедурам. Таким чином робота не приймається.

Обґрунтування:

2025-06-19

Доцент Микола Федула

Дата

експерт

Anti-Plagiarism (UA) v-15.281 Educational

The maximum coincidence with one document 2.0%

Dictionary check: en_US, ru_RU, ua_UA. **Errors in the documents: 9%**

ID: 246892 Title: БКР Робот пилосос з автоматичним рухом Added in a DB: 2025-06-19 Authors: Андрій СІМ'ЯЧЕНКО Heads: Юрій ФОРКУН Consultants: Opponents:	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
	41389	670	1116 (3%)	11 (2%)

Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ
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: Робот пилосос з автоматичним рухом

Автор: Сім'яченко Андрій Вікторович

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

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

Науковий керівник: Форкун Юрій Вікторович, 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, розміщені 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 розміщені 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 загальноживаних 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 5,84% і адресується до 17 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

Юрій ФОРКУН