

Міністерство освіти і науки України

Луцький національний технічний університет

(повне найменування закладу вищої освіти)

Факультет комп'ютерних та інформаційних технологій

(повне найменування факультету)

Кафедра комп'ютерної інженерії та безпеки

(повне найменування кафедри)

**КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»**

**МОБІЛЬНА РОБОТИЗОВАНА ПЛАТФОРМА НА БАЗІ
ARDUINO**

MOBILE ROBOTIC PLATFORM BASED ON ARDUINO

спеціальність 123 Комп'ютерна інженерія

(шифр і назва спеціальності)

освітня програма Комп'ютерна інженерія

(назва освітньої програми)

Виконав: здобувач вищої освіти
групи КІЗ-41
Березинський Павло Юрійович

(підпис)

Керівник:
к.т.н., доцент
Христинець Наталія Анатоліївна

(підпис)

Кваліфікаційну роботу
допущено до захисту
« » червня 2025 р.
Гарант освітньої програми:
к.т.н., доцент
Лавренчук Світлана Василівна

(підпис)

Луцьк – 2025 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра комп'ютерної інженерії та безпеки

Ступінь вищої освіти: бакалавр

Галузь знань: 12 Інформаційні технології

Спеціальність: 123 Комп'ютерна інженерія

Освітня програма: «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

к. т. н. Т. Терлецький

« 10 » 01 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Березинському Павлу Юрійовичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи Мобільна роботизована платформа на базі Arduino

Керівник роботи Христинець Наталія Анатоліївна

затвержені наказом закладу вищої освіти від «11» квітня 2025 року № 209/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи 11.06.2025р.

3. Вихідні дані до роботи Науково-технічна література та публікації в періодичних Виданнях по темі Інтернету речей, опубліковані роботи в даній області та різні інтернет-ресурси технічного спрямування

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

Вступ

Огляд науково-технічних джерел з питань розробки мобільних роботизованих платформ

Матеріали, способи та методи проектування

Реалізація роботизованої системи

Висновки

5. Перелік графічного (ілюстративного) матеріалу:

Фото мікроконтролеру, схеми взаємодії компонентів системи, рисунки інтерфейсу

Користувача, листинги коду

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
<i>Огляд науково-технічних джерел з питань розробки мобільних роботизованих платформ</i>	<i>Христинець Н.А, доцент</i>		
<i>Матеріали, способи та методи проектування</i>	<i>Христинець Н.А, доцент</i>		
<i>Реалізація роботизованої системи</i>	<i>Христинець Н.А, доцент</i>		
<i>Нормоконтроль</i>	<i>Багнюк Н.В., доцент</i>		
<i>Гарант ОП</i>	<i>Лавренчук С.В., доцент</i>		
<i>Показник запозичень тексту</i>		____%	
<i>Академічна доброчесність</i>	<i>Міскевич О.І., ст.викладач</i>		

7. Дата видачі завдання 14.04.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	<i>Огляд літератури із досліджуваної проблеми, аналіз предметної області та наявних рішень</i>	15.04.2025 р.	Виконано
2.	<i>Вибір апаратної та програмної бази для проекту</i>	25.04.2025 р.	Виконано
3.	<i>Проектування та тестування системи визначення місцезнаходження</i>	04.05.2025 р.	Виконано
4.	<i>Висновки та пропозиції</i>	07.05.2025 р.	Виконано
5.	<i>Формування списку використаних джерел</i>	17.05.2025 р.	Виконано
6.	<i>Оформлення ілюстративного матеріалу</i>	20.05.2025 р.	Виконано
7.	<i>Представлення остаточного варіанту кваліфікаційної роботи керівнику</i>	01.06.2025 р.	Виконано
8.	<i>Нормоконтроль</i>	31.05.2025 р.	Виконано
9.	<i>Інструментальна перевірка на академічний плагіат</i>	04.06.2025 р.	Виконано
10.	<i>Представлення кваліфікаційної та всіх супровідних документів на кафедрі</i>	11.06.2025 р.	Виконано

Здобувач вищої освіти

(підпис)

Березинський П.Ю.

(прізвище, ініціали)

Керівник кваліфікаційної роботи

(підпис)

Христинець Н.А.

(прізвище, ініціали)

АНОТАЦІЯ

Березинський П. Ю. Мобільна роботизована платформа на базі Arduino. Рукопис.

Кваліфікаційна робота бакалавра ОП «Комп'ютерна інженерія» спеціальності 123 Комп'ютерна інженерія. Луцький національний технічний університет. Луцьк, 2025.

Кваліфікаційна робота складається з вступу, трьох розділів, висновків та списку використаних джерел.

Перший розділ присвячено аналізу популярних мікроконтролерів, принципам роботи плати Arduino Uno на мікроконтролері ATmega328P . Розглянуто середовища проектування та проведено аналіз конкурентних проектів колісних роботизованих систем.

У другому розділі розглянуто матеріали, способи та методи проектування мобільної платформи. Сформовано загальний алгоритм формування рооботизованої системи. Запланований алгоритм переміщення з послідовним виконанням простих дій. Розглянуто принципи стабілізації цифрових входів системи.

Третій розділ описує практичну реалізацію роботизованої системи. Наведені фрагменти програмної реалізації колісного мобільного робота. Розглянута фізична збірка пристрою та тестування платформи.

Ключові слова: Arduino Uno, роботизована платформа, робототехніка, мікроконтролер, Arduino IDE, датчики.

ANNOTATION

Berezinsky P. Mobile robotic platform based on Arduino. Manuscript.

Qualification work of the bachelor of the OP «Computer Engineering» specialty 123 Computer Engineering. Lutsk National Technical University. Lutsk, 2025.

Qualification work consists of an introduction, three sections, conclusions and a list of sources used.

The first section is devoted to the analysis of popular microcontrollers, the principles of operation of the Arduino Uno board on the ATmega328P microcontroller. The design environment is considered and an analysis of competitive projects of wheeled robotic systems is conducted.

The second section considers materials, methods and techniques for designing a mobile platform. A general algorithm for forming a robotized system is formed. A movement algorithm is planned with the sequential execution of simple actions. The principles of stabilizing the digital inputs of the system are considered.

The third section describes the practical implementation of the robotic system. Fragments of the software implementation of a wheeled mobile robot are given. The physical assembly of the device and testing of the platform are considered.

Keywords: Arduino Uno, robotic platform, robotics, microcontroller, Arduino IDE, sensors.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1. ОГЛЯД НАУКОВО-ТЕХНІЧНИХ ДЖЕРЕЛ З ПИТАНЬ РОЗРОБКИ МОБІЛЬНИХ РОБОТИЗОВАНИХ ПЛАТФОРМ	9
1.1 Аналіз популярних мікроконтролерів	9
1.2 Технічні характеристики Arduino UNO.....	14
1.3 Огляд мов програмування та середовищ проектування	17
1.4 Аналіз конкурентних проектів	19
РОЗДІЛ 2. МАТЕРІАЛИ, СПОСОБИ ТА МЕТОДИ ПРОЕКТУВАННЯ	22
2.1 Опис алгоритму формування системи.....	22
2.2 Визначення параметрів для вимірювання	25
2.3 Опис поведінки системи	27
2.4 Стабілізація цифрових входів	29
2.5 Особливості обміну даними по WI-FI	32
РОЗДІЛ 3. РЕАЛІЗАЦІЯ РОБОТИЗОВАНОЇ СИСТЕМИ	34
3.1 Опис логіки роботи програми	34
3.2 Програмна реалізація колісного мобільного робота.....	36
3.3 Фізична збірка пристрою	38
3.4 Тестування роботи платформи та вимірювання точності датчиків.....	41
ВИСНОВКИ	43
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	44

ВСТУП

Актуальність теми полягає в широких можливостях застосування роботизованих платформ на базі Arduino для навчання, досліджень і створення прототипів автоматизованих систем. Такі платформи сприяють розвитку практичних навичок у галузях програмування, електроніки та схемотехніки. Вони дозволяють швидко реалізовувати різноманітні проекти, поєднуючи апаратну частину з програмною логікою. Завдяки відкритому коду, модульності та великій спільноті користувачів, Arduino забезпечує доступний вхід у сферу робототехніки.

Конструкції на основі Arduino є простими в налаштуванні та масштабуванні, що робить їх зручними як для початківців, так і для досвідчених розробників. Доступність компонентів дозволяє реалізовувати інженерні ідеї без значних фінансових витрат. Такі проекти розвивають аналітичне мислення та вміння працювати з технічно складними системами. Отже, роботизовані платформи на Arduino мають високу освітню, практичну та інженерну цінність.

Мета роботи – розробити та реалізувати мобільну роботизовану платформу на базі Arduino для автоматизованого пересування.

Об'єкт дослідження – мобільна роботизована платформа на базі мікроконтролера Arduino.

Предметом дослідження є онструктивні особливості, алгоритми керування та програмне забезпечення мобільної роботизованої платформи на базі Arduino для забезпечення її автоматизованого пересування..

Для досягнення мети необхідно виконати наступні завдання:

- провести аналіз сучасних мікроконтролерних систем та аргументувати вибір платформи для розробки;
- розробити поетапний підхід проектування роботизованої системи і визначити ключові ланки у розробці;
- сформулювати алгоритм трекингу руху та визначити, яку послідовність дій має виконувати платформа запланованої конструкції;

- визначити, які електронні компоненти забезпечують стабілізацію роботи системи;
- реалізувати програмне забезпечення для проектування руху платформи;
- провести фізичну збірку роботизованої платформи;
- виконати експериментальне дослідження роботи створеної системи, оцінити точність зчитування даних, затримки передачі та енергоспоживання.

РОЗДІЛ 1

ОГЛЯД НАУКОВО-ТЕХНІЧНИХ ДЖЕРЕЛ З ПИТАНЬ РОЗРОБКИ МОБІЛЬНИХ РОБОТИЗОВАНИХ ПЛАТФОРМ

1.1 Аналіз популярних мікроконтролерів

Вибір мікроконтролера є одним із ключових завдань при проектуванні мобільних роботизованих систем. Серед великого різноманіття сучасних платформ цей етап є визначальним для успішної розробки пристрою. Загалом, різноманітність доступних платформ сьогодні дає змогу розробникам обирати оптимальні рішення залежно від вимог проекту – від простих навчальних пристроїв до складних промислових систем.

З аналізу інтернет-джерел відомо, що серед широкого спектру мікроконтролерів особливо виділяються такі популярні платформи, як Arduino, ESP8266, ESP32 та STM32, які відрізняються за архітектурою, функціональністю та сферою застосування. Розглянемо їх детальніше для того, щоб визначити переваги та недоліки кожної системи та обрати оптимальний варіант для подальшої розробки.

Архітектура базових плат Arduino [1] заснована переважно на мікроконтролерах сімейства AVR. Починаючи з 2016 року, сімейство AVR офіційно належить Microchip Technology, тому, усі продукти Atmel, зокрема мікроконтролери AVR, вже перейшли під управління Microchip. Серед цих продуктів найбільш відомою є модель Arduino Uno, яка використовує мікроконтролер ATmega328P. Це є 8-розрядний процесор і він має обмежений обсяг оперативної пам'яті, флеш-пам'яті і порівняно низьку тактову частоту [2]. Проте, таких характеристик цілком достатньо для виконання типових задач контролю, опитування датчиків і керування виконавчими пристроями. Плата Arduino Uno має 14 цифрових вихідних пінів, з яких 6 підтримують широтно-імпульсну модуляцію і 6 аналогових входів. Архітектура передбачає використання зовнішнього кристала або резонатора для стабілізації тактової частоти. На платі також розташовані елементи, що забезпечують живлення,

інтерфейс USB для програмування і налагодження, а також стабілізатори напруги. Зараз існують популярні плати Arduino UNO Rev3 та Arduino UNO R4, які широко використовуються для розробок електронних проєктів. UNO Rev3 оснащена мікроконтролером ATmega328P і підходить для базових застосувань і навчання. UNO R4 є сучаснішою версією з потужним ARM Cortex-M4 процесором, що забезпечує вищу продуктивність і розширені можливості для складніших проєктів (рис. 1.1).

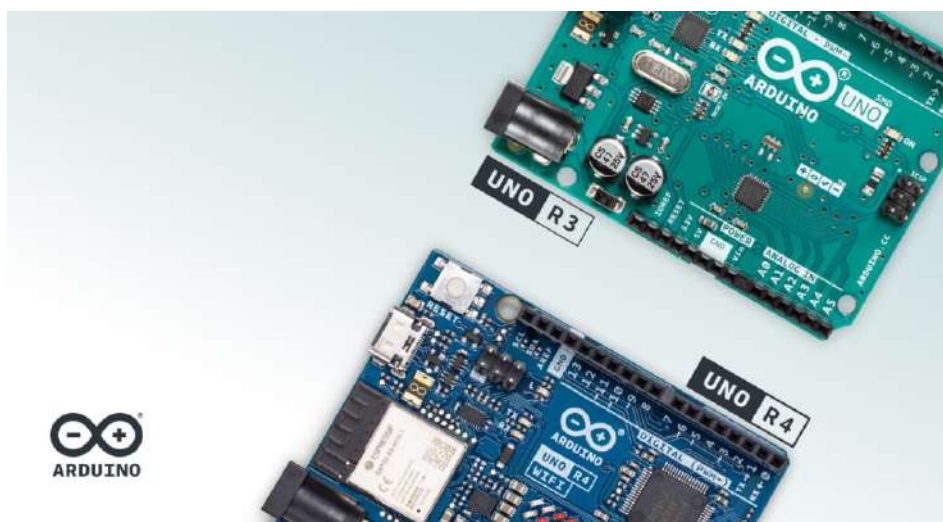


Рисунок 1.1 – Мікроконтролери Arduino Uno Rev3 та Rev4 [3]

Окрім цієї плати, зараз широку популярність має модуль ESP8266 китайського виробництва (рис. 1.2).



Рисунок 1.2 – Мікроконтролер ESP8266 [4]

Це недорогий та енергоефективний Wi-Fi модуль із вбудованим мікроконтролером. В основі його лежить 32-бітний процесор Tensilica Xtensa L106, який працює на частоті від 80 МГц до 60 МГц і це забезпечує достатню продуктивність для широкого спектра задач, включаючи обробку даних та керування периферією. Модуль має вбудований Wi-Fi стек із підтримкою протоколів TCP/IP, UDP, HTTP, FTP та SSL. Це дає змогу легко підключатися до мережі Інтернет і створювати власні точки доступу. ESP8266 обладнаний 17 GPIO-пінами, які можуть працювати з різними інтерфейсами, такими як SPI, I2C, UART, PWM і ADC, що значно розширює його функціональність для роботи з датчиками, виконавчими механізмами та іншими пристроями. Завдяки підтримці різних середовищ програмування, включаючи і Arduino IDE, і MicroPython і інші, розробники надають широкі можливості в плані вибору інструментів для швидкої та гнучкої реалізації власних проектів.

Окрім апаратних можливостей, ESP8266 характеризується низьким енергоспоживанням з підтримкою різних режимів сну, що важливо для автономних пристроїв, де проектується живлення від акумуляторів. Це досить економічно, тому цей модуль часто використовується для створення розумних домашніх систем, сенсорних мереж, автоматизації, дистанційного моніторингу та інших IoT-застосунків завдяки простоті інтеграції, великій спільноті користувачів і безлічі готових бібліотек.

На відміну від розглянутого ESP8266, мікроконтролер ESP32 має більш потужний двоядерний процесор Tensilica Xtensa LX6. Наявність такого потужного процесора, як для мікроконтролерної системи, дозволяє забезпечити значно вищу її обчислювальну продуктивність і можливість паралельної обробки задач. Крім підтримки Wi-Fi, ESP32 також оснащений вбудованим модулем Bluetooth, що розширює його сфери застосування у бездротовому зв'язку (рис. 1.3). ESP32 має більший обсяг оперативної пам'яті, більш розвинену периферію, включно з додатковими каналами ADC, DAC, сенсорними входами та апаратним прискоренням криптографічних алгоритмів.



Рисунок 1.3 – Мікроконтролер ESP32 [5]

В основному його необхідність є там, де потрібна одночасна обробка великої кількості інформації та багатофункціональна бездротова взаємодія.

Наступним популярним мікроконтролером є STM32, який базується на ядрі ARM Cortex-M і має 32-бітну архітектуру. Він має різні варіанти виготовлення з різною тактовою частотою, обсягом оперативної та флеш-пам'яті, а також підтримкою численних периферійних модулів, таких як UART, SPI, I2C, ADC і таймери. STM32 широко використовуються в промислових і вбудованих системах для керування пристроями, обробки сигналів та реалізації протоколів зв'язку в реальному часі (рис. 1.4).

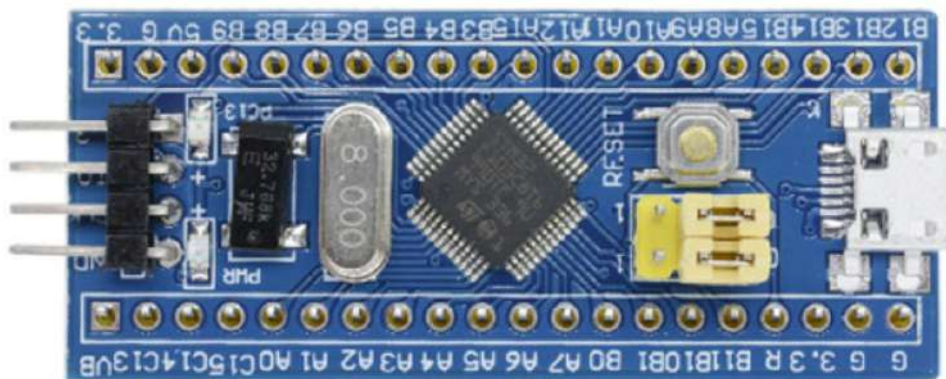


Рисунок 1.4 – Мікроконтролер SMT32 [6]

Аналіз їх технічних характеристик, можливостей і переваг дозволяє зрозуміти, які платформи найкраще підходять для конкретних завдань і як ефективно їх використовувати в різних завданнях (таблиця 1.1).

Таблиця 1.1 – Функції мікроконтролерів [7]

	Arduino (UNO)	ESP8266	ESP32	STM32F4 Series
Архітектура	8-бітна AVR	32-бітна	32-бітна	32-бітна ARM Cortex-M4
Частота процесора	16 MHz	80 MHz-160 MHz	до 240 MHz	до 180 MHz
RAM	2 KB	64 KB інструкції, 96 KB дані	520 KB SRAM	від 128 KB до 384 KB
Flash-пам'ять	32 KB	512 KB to 4 MB	4 MB to 16 MB	від 512 KB до 2 MB
Вольтаж	5V	3.3V	3.3V	3.3V
I/O Pins	14 digital, 6 analog	11 digital, 1 analog	34+ GPIO pins	Up to 100+ GPIO (differs for each board)
USB інтерфейси	+	+	+	+
Середовище програмування	Arduino IDE	Arduino IDE, ESP-IDF, PlatformIO	Arduino IDE, ESP-IDF, PlatformIO	STM32CubeIDE, Keil, IAR
Складність використання	для початківців	середня складність	високий рівень	професійні навички
Найкраще підходить для	Прості проекти, навчання	Базові програми IoT	Розширені IoT, завдання в режимі реального часу	Промислові, автомобільні, високопродуктивні системи

Як видно з таблиці, вибір правильного мікроконтролера залежить від складності та вимог розроблюваного проекту. Arduino є найкращим вибором для початківців та освітніх проектів. ESP8266 та ESP32 користуються великою популярністю для повноцінних IoT-додатків, де використовують Wi-Fi, завдяки своїй економічній ефективності та вбудованому бездротовому з'єднанню. STM32 виділяється взагалі для промислових застосувань, які потребують надійної

продуктивності, низького енергоспоживання та передових периферійних пристроїв.

Для розробки мобільної роботизованої системи було обрано платформу Arduino через її простоту використання та широку підтримку серед початківців і розробників-аматорів. Arduino забезпечує достатній набір входів/виходів для підключення датчиків і виконавчих механізмів, має просту архітектуру та працює на стабільній частоті, що робить її надійною для базових задач керування. Крім того, велика кількість готових бібліотек і прикладів дозволяє швидко розробляти і тестувати проєкт без потреби в глибоких знаннях складних середовищ програмування. Саме ці переваги роблять Arduino оптимальним вибором для створення елементарної мобільної роботизованої системи, де важлива швидкість розробки і надійність.

1.2 Технічні характеристики Arduino UNO

Універсальна мікроконтролерна плата Arduino Uno розроблена для створення прототипів вбудованих систем та автоматизованих пристроїв. Вона базується на мікроконтролері ATmega328P і є однією з найпопулярніших моделей у лінійці Arduino завдяки своїй простоті, доступності та широкій спільноті користувачів.

Обрана плата забезпечує необхідний мінімум для запуску мікроконтролера, включаючи USB-інтерфейс для програмування, стабілізатор живлення, кварцовий генератор та роз'єм для зовнішнього живлення.

Мікроконтролер, що використовується на платі Arduino UNO – 8-бітовий ATmega328P. Це мікроконтролер, який належить до серії AVR і був розроблений компанією Atmel. Він працює на тактовій частоті 16 мегагерц, що забезпечує стабільну та достатньо швидку роботу при обробці сигналів, обміні даними з периферійними пристроями та виконанні логіки керування. Об'єм флеш-пам'яті становить 32 кілобайти, з яких частина використовується для завантажувача, а решта – для основної програми. Для тимчасового зберігання змінних в

оперативній пам'яті доступно 2 кілобайти SRAM. Крім того, є 1 кілобайт EEPROM – пам'яті, що зберігає дані навіть після вимкнення живлення, що дозволяє зберігати критично важливі параметри чи налаштування.

ATmega328P базується на архітектурі AVR, яка відома своєю простотою у вивченні, ефективним набором команд та низьким енергоспоживанням. Завдяки цьому мікроконтролер легко програмується мовою C або з використанням Arduino IDE, що відкриває широкі можливості для розробки систем автоматизації, моніторингу та керування. Архітектура AVR також дозволяє забезпечити високий рівень надійності та швидкий відгук системи на зовнішні події, що особливо важливо для реального часу у вбудованих пристроях. Розпіновка (рис. 1.5 а) та схема підключення живлення (рис. 1.5 б) до мікроконтролера ATmega328P є важливою складовою розробки електронних схем, що базуються на цьому чипі.

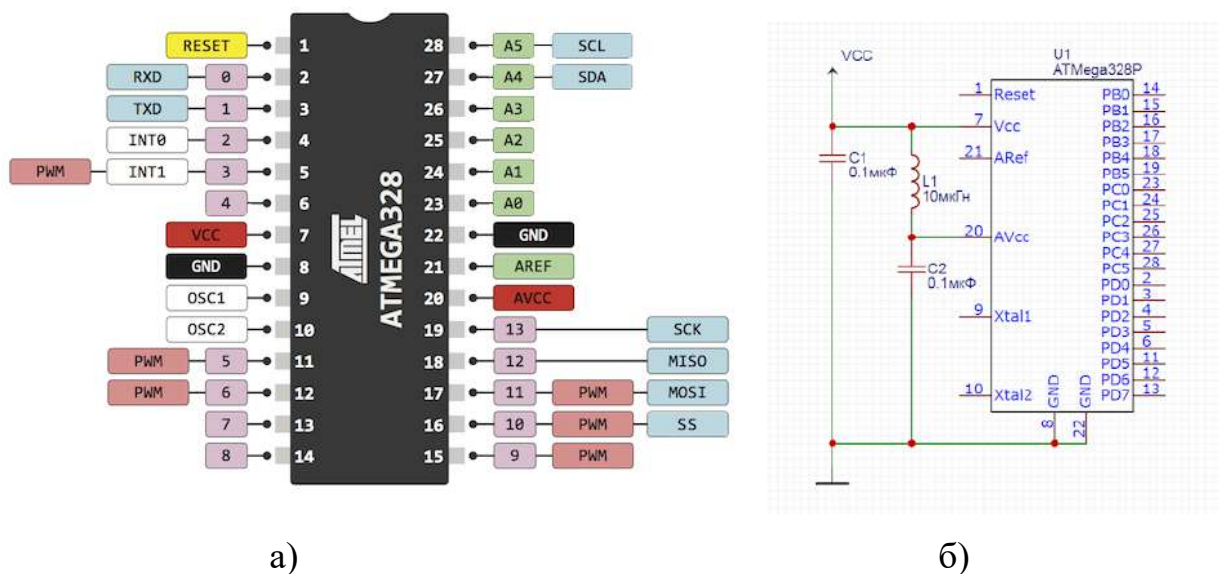


Рисунок 1.5 – Контактні піни (а) та схема підключення живлення (б) мікроконтролера ATmega328P [8]

Завдяки відкритій архітектурі, Arduino UNO активно використовується в освітніх проєктах, IoT-рішеннях, робототехніці та системах моніторингу.

Простота підключення датчиків, модулів і виконавчих пристроїв робить її ідеальним вибором для студентських кваліфікаційних робіт та навчальних цілей.

Плата, окрім мікроконтролера, оснащена зручним набором інтерфейсів введення та виведення, що дозволяє легко підключати датчики, виконавчі пристрої та модулі розширення (табл. 1.2).

Таблиця 1.2 – Інтерфейси введення/виведення плати Arduino UNO

Тип інтерфейсу I/O	Кількість	Призначення
Цифрові пінові входи/виходи	14	Введення або виведення цифрових сигналів (рівні лог. 0 або 1)
виходи широтно імпульсної модуляції ШІМ	6 із 14	Генерація широтно-імпульсних сигналів для керування потужністю
Аналогові входи	6	Зчитування аналогових сигналів з роздільною здатністю 10 біт
UART (послідовний порт, асинхронний приймач і передавач)	1 (TX/RX)	Обмін даними з комп'ютером або модулями через інтерфейс USB або Bluetooth
SPI	1 (підмножина пінів)	Швидкий синхронний обмін даними з периферійними пристроями
I ² C	1 (A4 – SDA, A5 – SCL)	Обмін даними з датчиками та модулями по двохдротовій шині

Усього плата має 14 цифрових пінів введення/виведення, які можуть використовуватись як для зчитування сигналів, так і для подачі керуючих імпульсів. З них шість пінів підтримують режим широтно-імпульсної модуляції (ШІМ), що дозволяє створювати аналогоподібні сигнали, наприклад, для керування яскравістю світлодіода або швидкістю обертання двигуна.

Окрім цифрових пінів, Arduino UNO має шість аналогових входів. Ці входи дозволяють зчитувати напругу у діапазоні від 0 до 5 вольт із роздільною здатністю 10 біт, тобто сигнал перетворюється на значення від 0 до 1023. Це дає змогу точно вимірювати параметри з аналогових датчиків, таких як температурні, світлові чи газові сенсори.

Завдяки наявності як цифрових, так і аналогових каналів введення/виведення, Arduino UNO забезпечує гнучкість у побудові систем керування, автоматизації та моніторингу, що робить її придатною для використання у різноманітних прикладних задачах кваліфікаційної роботи.

Основний принцип роботи Arduino UNO базується на виконанні послідовності інструкцій, записаних у програму, що зберігається у внутрішній флеш-пам'яті мікроконтролера. Мікроконтролер постійно зчитує вхідні сигнали з підключених датчиків через цифрові або аналогові входи, обробляє отримані дані згідно з алгоритмом і формує керуючі сигнали на вихідні піні. Завдяки своїй архітектурі та підтримці різноманітних інтерфейсів введення/виведення, плата здатна працювати у режимі реального часу, реагуючи на зміни в навколишньому середовищі та взаємодіяти з іншими пристроями.

У контексті виконуваної кваліфікаційної роботи бакалавра, Arduino UNO виступає, як основний контролер для збору та обробки даних для керування роботизованою системою, а також для керування підключеними модулями чи виконавчими пристроями. Її застосування дозволяє реалізувати автоматизовану систему, яка забезпечує моніторинг параметрів, таких як температура, освітленість чи рух, а також своєчасне реагування на події. Простота програмування і багатофункціональність роблять цю платформу ідеальним вибором для розробки прототипів і впровадження практичних рішень у навчальних та наукових роботах.

1.3 Огляд мов програмування та середовищ проектування

Мови програмування та середовища проектування роботизованих систем є важливою складовою процесу створення ефективних та надійних робототехнічних рішень. Сучасні роботизовані системи потребують гнучких, зручних і водночас потужних інструментів для розробки, які дозволяють реалізувати алгоритми керування, обробки даних з датчиків та взаємодії з апаратною частиною. Найпоширенішими мовами програмування в цій сфері є C, C++, Python, а також спеціалізовані середовища, які поєднують в собі графічні інтерфейси та можливості кодування.

Мова програмування C та її розширення C++ є основою для більшості вбудованих систем, включно з робототехнікою. Вони дозволяють максимально

контролювати апаратні ресурси, працювати з низькорівневими протоколами і реалізовувати оптимізовані алгоритми в реальному часі. Однак для початківців і для швидкої розробки прототипів вони можуть бути складними через необхідність врахування багатьох технічних деталей та обмежень апаратури. Саме тут на допомогу приходять спеціалізовані середовища розробки, такі як Arduino IDE.

Arduino UNO – це одна з найпопулярніших платформ для створення роботизованих систем серед аматорів, студентів і навіть професіоналів. Основною мовою програмування для Arduino є спрощений діалект C/C++, що забезпечує достатню потужність і одночасно зручність для новачків. Arduino IDE надає користувачам зрозумілий інтерфейс, який дозволяє швидко писати, компілювати та завантажувати код у мікроконтролер. Крім того, велика спільнота користувачів створила безліч бібліотек, які полегшують роботу з різноманітними датчиками, моторами та комунікаційними модулями.

Важливою перевагою Arduino UNO у контексті робототехніки є її апаратна гнучкість і широкий набір інтерфейсів введення-виведення, що дозволяють легко підключати сенсори, виконавчі механізми та комунікаційні модулі. Завдяки цьому платформа ідеально підходить для реалізації проектів з керування рухом, обробки сигналів, а також для побудови систем автономного управління. Водночас Arduino UNO підтримує програмування в середовищах, сумісних з Arduino, що розширює можливості розробки, наприклад, використання Visual Studio Code або PlatformIO.

Таким чином, огляд мов і середовищ проектування демонструє, що Arduino UNO є універсальним та доступним інструментом для розробки роботизованих систем. Вона поєднує простоту використання, потужність програмування на C/C++ і підтримку великої кількості апаратних компонентів, що робить її ідеальним вибором для навчання, досліджень та впровадження практичних рішень у сфері робототехніки.

1.4 Аналіз конкурентних проектів

Аналіз конкурентних проектів є важливим етапом у розробці будь-якої роботизованої системи, оскільки дозволяє оцінити існуючі рішення, їхні сильні та слабкі сторони, а також визначити унікальні аспекти власного проекту. На сучасному ринку представлено багато роботизованих платформ, що базуються на різних мікроконтролерах та одноплатних комп'ютерах. Наприклад, популярність набули системи на основі Arduino, Raspberry Pi та ESP8266 та ESP32, які широко використовуються як у навчальних, так і у промислових рішеннях [7, 9].

Одним із відомих прикладів є проект RobotDyn UNO, який використовує Arduino UNO як основну платформу для розробки робототехнічних систем. Цей проект підкреслює простоту програмування та велику кількість доступних бібліотек для роботи з датчиками та виконавчими механізмами. Завдяки відкритому апаратному забезпеченню та гнучкості Arduino UNO, користувачі можуть швидко створювати прототипи та реалізовувати індивідуальні алгоритми керування.

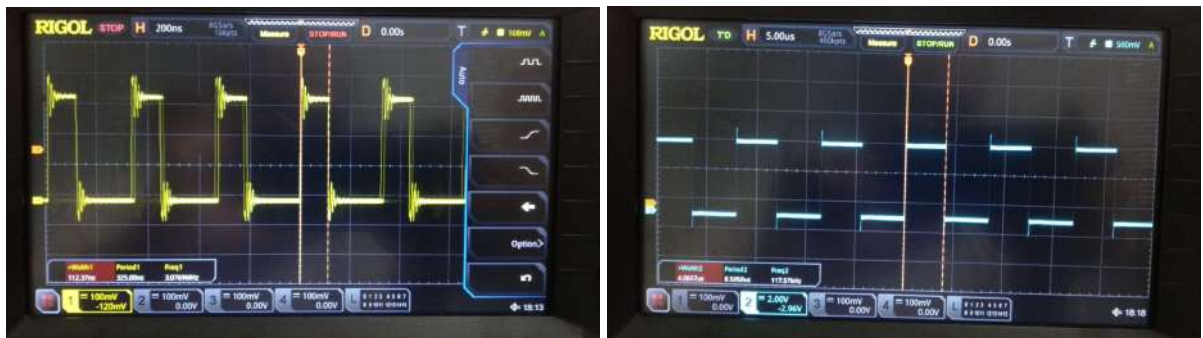
Ще одним прикладом є система на базі ESP32, яку можна знайти у проекті ESP32 Robot. Цей проект вирізняється підтримкою Wi-Fi та Bluetooth, що дає змогу створювати більш складні бездротові рішення. Проте, для початківців і навчальних цілей ESP32 може бути складнішою для освоєння через більшу складність апаратної архітектури і потребу більш глибоких знань програмування. Крім того, споживання енергії ESP32 вище, що не завжди доцільно для автономних роботизованих систем.

У режимі активної роботи ESP32 споживає близько 160-240 мА, тоді як Arduino UNO зазвичай споживає в межах 45-50 мА. У режимі сну Arduino також має нижче енергоспоживання – менше 1 мА, тоді як у ESP32 навіть у глибокому сні струм споживання може сягати 10 мА або більше, залежно від конфігурації.

Розглянемо важливий фактор споживання напруг різними типами мікроконтролерів (рис. 1.7). Робоча напруга мікропроцесорів ESP становить 3,3

В порівняно з робочою напругою Arduino, яка становить 5 В. Якщо плати використовуються підключеними до розетки, різниці в споживанні енергії не буде, оскільки струм буде зменшено для отримання такої ж кількості енергії.

Проведене дослідження [10], у якому аналізувалося виконання одного й того самого фрагмента коду на платах ESP32, ESP8266, Arduino Uno та Arduino Mega, дозволяє порівняти швидкодію цих мікроконтролерних платформ у контексті ключових інструкцій. Зокрема, увага зосереджена на двох важливих параметрах – часі виконання функції `digitalWrite` та періоді виконання порожнього циклу `loop()`, які безпосередньо впливають на швидкість обробки подій у реальному часі (рис. 1.6).



а)

б)

Рисунок 1.6 – Швидкодія платформ ESP32 (а) та Arduino Uno (б) під час виконання функції `digitalWrite` та періоді виконання циклу `loop()` [10]

Першим параметром є `digitalWrite Pulse width` – час, необхідний для зміни логічного рівня на цифровому піні. Цей показник є критичним у системах, де необхідне точне та швидке керування цифровими сигналами. Найшвидший результат показала плата ESP32 з часом лише 112,37 наносекунд, що свідчить про практично миттєве реагування. За нею йде ESP8266 із результатом 800 наносекунд, що хоч і повільніше за ESP32, але значно швидше, ніж у Arduino. Arduino Uno виконує цю операцію за 4,07 мікросекунди, а Arduino Mega – ще повільніше, приблизно за 5,44 мікросекунди, що пояснюється особливостями їхньої AVR-архітектури та складнішим доступом до портів.

Другим параметром є `Void loop period`, тобто час, за який мікроконтролер проходить через порожню функцію `loop()` без жодних інструкцій. Це дає уявлення про загальну «легкість» і швидкість виконання коду. У цьому аспекті ESP32 знову демонструє найкращий результат – лише 0,325 мікросекунди, що гарантує дуже швидку реакцію на зміни в середовищі. ESP8266 показав дещо гірший, але все ще досить високий показник – 5,16 мікросекунди. У свою чергу, Arduino Uno має час виконання порожнього циклу 8,51 мікросекунди, а Arduino Mega – найдовший цикл, який триває 11,25 мікросекунди.

Зіставлення цих даних демонструє суттєву перевагу ESP32 та ESP8266 у задачах, де критичною є швидкодія. Проте варто враховувати, що ці контролери складніші в налаштуванні, мають вищі енергоспоживання та потребують додаткової стабільності живлення. У той час як Arduino Uno або Mega, хоча й повільніші, залишаються стабільним вибором для проєктів, де не вимагається максимальна швидкість, але важливі передбачуваність, простота у використанні та енергоефективність. Саме така платформа буде використана у кваліфікаційній роботі.

Інші конкурентні розробки, наприклад, на базі Raspberry Pi [11], часто фокусуються на потужних обчислювальних можливостях і підтримці складних операційних систем, таких як Linux. Це дозволяє реалізовувати більш складні алгоритми, включно з комп'ютерним зором та штучним інтелектом. Однак такі системи вимагають більшої енергозабезпеченості, складнішого налаштування і менш підходять для простих керуючих задач з обмеженими ресурсами.

Отже, з урахуванням розробки схожих систем, вибір на користь Arduino UNO має низку переваг. По-перше, Arduino UNO є більш простим у використанні та має широку підтримку серед спільноти розробників, що значно полегшує розробку та налагодження. По-друге, вона споживає менше енергії, що важливо для автономних пристроїв. По-третє, достатня кількість цифрових і аналогових входів дозволяє ефективно підключати різні датчики та виконавчі пристрої без потреби в складних апаратних рішеннях.

РОЗДІЛ 2

МАТЕРІАЛИ, СПОСОБИ ТА МЕТОДИ ПРОЕКТУВАННЯ

2.1 Опис алгоритму формування системи

Проектування роботизованої системи передбачає поетапний підхід, що включає вибір апаратної платформи, мови програмування, середовища розробки, апаратних компонентів та етапів реалізації і тестування. Кожен із цих кроків критично впливає на ефективність, точність, масштабованість і адаптивність майбутнього роботизованого пристрою.

Першим етапом є вибір платформи, яка буде основою керування. У цьому проєкті основною обрано Arduino UNO, що є популярною мікроконтролерною платою з процесором ATmega328P, достатнім для обробки сигналів від сенсорів, керування виконавчими пристроями і забезпечення базової логіки роботи.

Альтернативами, звичайно, можуть бути Raspberry Pi – потужніший одноплатний комп'ютер для складніших обчислень, обробки зображень, ШІ, або інші мікроконтролери, як STM32, ESP32, Teensy, що дозволяють розширювати функціональність або працювати в реальному часі.

Наступним кроком є вибір мови програмування, який залежить від вибраної платформи. Для Arduino UNO зазвичай використовуються C та C++, оскільки Arduino IDE компілює програми саме в цих мовах. У випадку Raspberry Pi часто використовується Python через його простоту, велику кількість бібліотек для робототехніки, машинного навчання та роботи з сенсорами. Також у залежності від поставлених завдань можливе використання Java, JavaScript, MATLAB, зокрема, для інтеграції з візуальними або мережевими інтерфейсами чи математичного моделювання.

Для написання коду обирається середовище розробки, зручне для програміста. Для Arduino найчастіше застосовується Arduino IDE – просте й інтуїтивне середовище з готовими бібліотеками та серійною відлагоджувальною системою.

Для просунутіших проєктів із кількома файлами, модульною структурою або застосуванням зовнішніх бібліотек зручним є PlatformIO, яке інтегрується у Visual Studio Code. Для програмування іншими мовами можуть застосовуватись Eclipse, PyCharm, MATLAB тощо.

Ключовим етапом є визначення апаратних складових, які взаємодіятимуть із зовнішнім світом. Для сприйняття середовища використовуються різноманітні датчики: температурні для контролю клімату, ультразвукові для визначення відстані до перешкод, світлові для виявлення ліній або освітлення, інерційні (акселерометри, гіроскопи) – для трекінгу руху.

Виконавчі пристрої забезпечують фізичну взаємодію з середовищем – це серводвигуни для точного позиціонування, крокові двигуни для обертального руху з чіткою кількістю кроків, реле для перемикання потужних ланцюгів або світлодіоди для індикації.

Для зв'язку з користувачем або іншими пристроями використовуються комунікаційні модулі – Bluetooth для бездротового керування, Wi-Fi для віддаленого моніторингу або оновлення, GSM/3G/4G – для передачі даних через мобільні мережі.

Останній етап – розробка та тестування – включає написання коду згідно з логікою роботи системи, налаштування взаємодії між компонентами, запуск симуляцій (наприклад, в Tinkercad або Proteus), а також тестування всієї системи в реальних умовах. На цьому етапі відбувається відлагодження алгоритмів, перевірка реакції на зовнішні події, оцінка точності вимірювань і стабільності руху.

Важливим етапом перед переходом до технічного проєктування є розробка алгоритмів, які описують логіку роботи системи, її реакцію на зовнішні події та взаємодію між компонентами. Саме алгоритмічна структура визначає послідовність дій, на основі якої будуються апаратна архітектура і програмне забезпечення. Спрощена схема проєктування роботизованої системи, яка включає вибір платформи, мови програмування, середовища розробки та основних складових системи подана на рисунку 2.1.



Рисунок 2.1 – Алгоритм розробки роботизованої системи

Таким чином, схема проектування роботизованої системи демонструє інтеграцію програмної та апаратної складових із урахуванням обраної платформи, засобів розробки та функціональних вимог до системи.

2.2 Визначення трекингу руху

Процес трекингу руху роботизованої колісної платформи є важливим елементом поведінки усієї спроектованої системи. Він дозволяє аналізувати динаміку руху, контролювати маршрут, коригувати траєкторію і здійснювати навігацію у змінному середовищі.

Для перевірки працездатності системи та демонстрації основ керування двоколісною роботизованою платформою було заплановано та реалізовано спрощений алгоритм переміщення. Такий підхід дозволяє на початковому етапі оцінити взаємодію програмного коду з апаратною частиною та протестувати основні рухові функції пристрою.

Алгоритм передбачає послідовне виконання простих дій: прямолінійний рух вперед протягом 2 секунд, короткої зупинки на 1 секунду, потім розворот на місці шляхом обертання вліво протягом 1 секунди та остаточної зупинки. Цей набір дій є базовим, але він є і достатнім для початкового тестування функціональності приводів, керуючого мікроконтролера та логіки роботи з моторами.

У подальших розділах наведено програмну реалізацію цього алгоритму, опис конфігурації підключення виконавчих елементів та аналіз результатів виконання програми на фізичному прототипі.

У контексті використання платформи Arduino Uno та базових сенсорів, реалізація трекингу може здійснюватися за допомогою кількох методів. Найпростіший з них – одометричний підхід, що базується на підрахунку обертів коліс. Для цього на кожне колесо встановлюються енкодери. Якщо у якості енкодерів взяти інфрачервоні з прорізним диском, сигнали з яких надходять на цифрові входи Arduino, то, аналізуючи кількість імпульсів, система буде розраховувати відстань, яку пройшов робот, а також можна буде визначити повороти шляхом диференціації обертів правого і лівого колеса. Такий підхід дозволяє вираховувати приблизні координати відносно стартової точки у двовимірному просторі.

Іншим методом є використання інерціальних сенсорів, зокрема гіроскопа і акселерометра у вигляді модуля MPU6050. Вони мають змогу забезпечувати дані про кутові швидкості та прискорення і, інтегруючи ці значення, можна обчислити зміну положення і напрямок руху робота.

Поєднання одометричних і інерціальних даних дозволяє підвищити точність трекінгу колісного робота. Цей підхід відомий, як сенсорне злиття. Для цього можуть використовуватись фільтри Калмана [12], чи комплементарні фільтри, які враховують похибки кожного окремого сенсора (рис. 2.2).

У спрощеному варіанті трекінг може виконуватись за зовнішнім маркуванням траєкторії. Якщо платформа обладнана ІЧ-датчиками або камерою, вона здатна слідкувати за чорною лінією на світлому фоні. Такий метод називається лінійним трекінгом (line following) і часто використовується для демонстрацій або в освітніх проєктах. У цьому випадку система не потребує глобального позиціонування, а лише визначає напрямок відносно лінії й регулює швидкість коліс для збереження орієнтації.

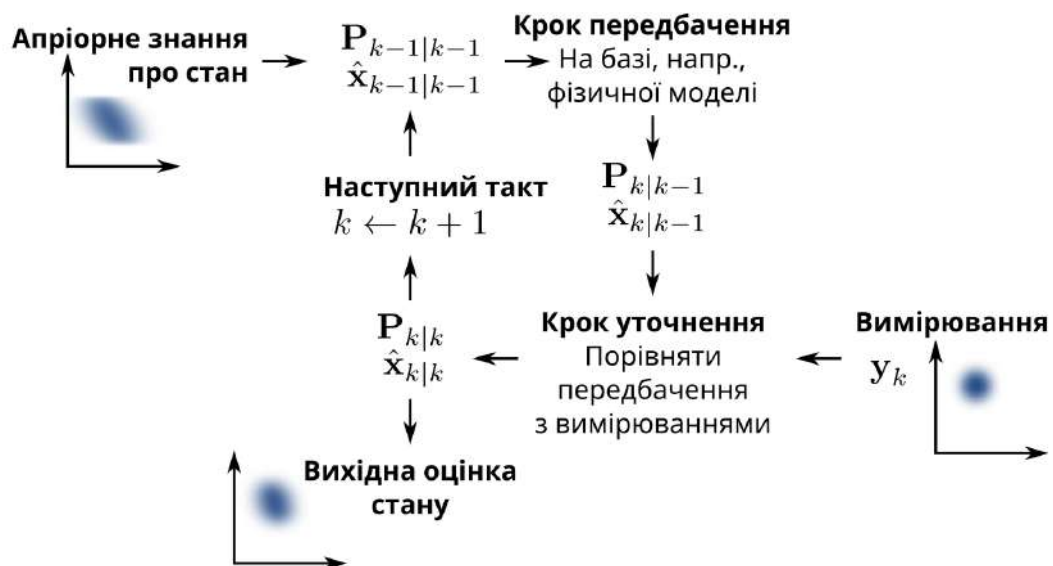


Рисунок 2.2 – Фільтр Калмана [12]

Використання зовнішніх систем позиціонування, таких як GPS або маяки з технологією Bluetooth в контексті простих навчальних роботів з Arduino Uno зазвичай не використовується через складність та високе енергоспоживання.

Проте, вибір методу трекінгу залежить від мети проекту, точності, якої необхідно досягти, та технічної бази платформи. Для досліджуваної системи з ультразвуковим сенсором та Arduino Uno базовий одометричний підхід із потенційним доповненням інерціального модуля є оптимальним рішенням.

2.3 Опис поведінки системи

Керування невеликим двигуном, який входить до складу мобільної роботизованої системи, можна реалізувати досить просто. Якщо потужність двигуна невелика, то його можна підключати безпосередньо до цифрового виводу плати Arduino і керування в цьому випадку буде здійснюватись шляхом подачі логічного рівня через увімкнення і вимкнення сигналу. Такий підхід дозволяє організувати базу роботи з виконавчими пристроями, але він не є оптимальним для практичного використання. Тому, доцільно переходити до керування через транзистори або драйвери, бо вони дозволяють працювати з більшими навантаженнями без ризику для мікроконтролерної плати.

У даному проєкті буде реалізовано підключення компактного вібраційного моторчика (рис. 2.3) безпосередньо до Arduino.



Рисунок 2.3 – Вібромоторчик з ексцентриковим навантаженням

Для реалізації схеми знадобляться: плата Arduino, підключена до комп'ютера через USB, резистор номіналом 220 Ом і сам мініатюрний вібропривод з ексцентриковим навантаженням. Використаний в роботі вібромотор працює при напрузі живлення 3-4,5 В. Він розміщений у пластиковому корпусі загальною довжиною 15 мм. Діаметр самого двигуна становить 4,17 мм, а зовнішній діаметр силіконового покриття – 5 мм.

Схема підключення моторчика виглядає наступним чином. Вібромотор має два провідники живлення. Один із них потрібно підключити до контакту GND на платі Arduino. Немає принципової різниці, який саме провід буде підключено до «землі», бо моторчик не має полярності. Другий провід моторчика з'єднується з вибраним цифровим виходом Arduino через резистор. Цей резистор виконує захисну функцію – він обмежує струм, що проходить через вивід плати, тим самим запобігаючи перевантаженню або пошкодженню мікроконтролера. Оскільки плата Arduino не розрахована на безпосереднє керування силовими навантаженнями, таке рішення забезпечує її стабільну роботу.

У кваліфікаційній роботі було задіяно лише один вібраційний мотор безпосередньо до контролера, однак, можна було під'єднати ще кілька таких моторчиків. Цифрові піни Arduino з 2 по 4 можна використати для незалежного керування навіть трьома різними двигунами і кожен цифровий вихід Arduino здатний окремо активувати один мотор. Слід враховувати, що таке рішення не є оптимальним, бо збільшення кількості підключених двигунів спричиняє зростання навантаження на саму плату, а це може призвести до перевищення допустимого струму. Тому, для розробки використано один двигун, щоб забезпечити безпечну і стабільну роботу системи.

Електродвигуни постійного струму за своєю природою є індуктивними навантаженнями. Це означає, що після вимкнення живлення або при обертанні вручну, вони можуть генерувати зворотну напругу. Такий імпульс здатен пошкодити підключену електроніку, якщо не забезпечити належний захист. Щоб цього уникнути, то між цифровим виходом і джерелом живлення 5 В

встановлюється захисний діод, який «гасить» зворотну ЕРС, скеровуючи її безпосередньо на плюс живлення.

Щоб запобігти пошкодженню електронних компонентів через зворотну електрорушійну силу, яка виникає при відключенні живлення індуктивного навантаження двигуна, між цифровим виходом Arduino та джерелом живлення 5 В підключено захисний діод. Цей діод розташовано в зворотному напрямку відносно основного струму: анод – до 5 В, а катод – до виходу Arduino. Коли двигун перестає отримувати живлення, він генерує зворотну напругу, яка може піти назад у мікроконтролер. У такому випадку діод відкривається і перенаправляє зворотний струм назад до джерела живлення, замкнувши його безпечним шляхом. Таким чином, діод ефективно гасить зворотну напругу, не даючи їй потрапити до чутливих елементів плати та запобігаючи їх виходу з ладу.

Кожен пін Arduino має вбудований діод, тому потреби в додаткових зовнішніх захисних елементах немає.

2.4 Стабілізація цифрових входів

Електронні компоненти, здатні керувати потужним струмом за допомогою відносно слабкого керувального сигналу, відіграють ключову роль у цифрових схемах мікроконтролерних розробок, у тому числі колісних мобільних систем. Зокрема, при подачі невеликого струму на керуючий вивід такого елемента, між двома іншими виводами може виникнути значно потужніший струм. Величина цього струму прямо залежить від сили керуючого сигналу: чим більший вплив на вхід, тим більший струм протікає через решту схеми.

Ця властивість широко застосовується для підсилення сигналів, перемикання логічних станів, а також – що особливо важливо у цифровій електроніці – для стабілізації входів логічних елементів. У цифрових системах коректне розпізнавання логічних рівнів, і високого, і низького, є критично важливим для надійної роботи мікроконтролерів, логіки та комунікаційних шин.

У випадках, коли вхідний контакт залишається «в підвішеному стані», тобто який не з'єднаний ні з логічною «1», ні з «0», виникає загроза неконтрольованого перемикання або збоїв у логіці. Саме тому виникає потреба у стабілізації вхідного сигналу, яку можна реалізувати за допомогою спеціальних схем із використанням керованих електронних компонентів.

При подачі логічної одиниці на цифровий вихід мікроконтролера, зокрема Arduino, на базу транзистора типу NPN подається струм, який відкриває транзистор, дозволяючи основному струму протікати від колектора до емітера. У випадку встановлення логічного нуля на виході мікроконтролера, струм до бази не надходить, транзистор залишається закритим, і струм через колекторно-емітерний перехід не протікає. Однією з ключових особливостей таких елементів є здатність керування значно більшим струмом при допомозі порівняно малого керуючого струму. Коефіцієнт підсилення струму транзистора, характеризує співвідношення між струмом колектора і струмом бази. Для типових малопотужних NPN-транзисторів цей коефіцієнт може сягати значення близько 200, що означає, що при струмі бази 1 мА транзистор може комутувати струм до 200 мА через колекторно-емітерне з'єднання (рис. 2.4).

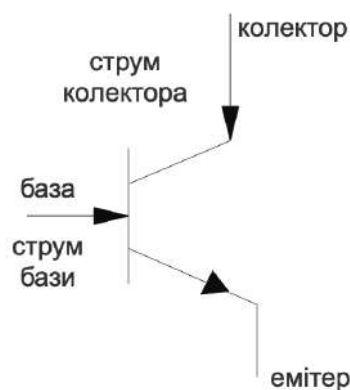


Рисунок 2.4 – Структурна схема nnp-транзисторів

У випадку керування індуктивним навантаженням, наприклад, електродвигуном, обов'язковим елементом захисту є зворотний діод, підключений паралельно до індуктивності. Під час різкого вимкнення живлення на індуктивному елементі виникає зворотна електрорушійна сила, яка може

призвести до пробую транзистора через перенапругу. Діод, з'єднаний у зворотній полярності до джерела живлення, виконує роль шунта, розсіюючи паразитну енергію в безпечний спосіб. Це забезпечує стабільну й надійну роботу транзистора у схемах комутації індуктивних навантажень та запобігає його виходу з ладу.

Таким чином, використання транзистора в парі з захисним діодом утворює ефективний і безпечний засіб керування навантаженнями у цифрових пристроях.

Керуючий вивід транзистора, відомий як база, має високу чутливість до електричних сигналів. Через це навіть незначне електростатичне наведення, наприклад від дотику, може призвести до хибного спрацьовування – запуску двигуна або іншого навантаження без подачі сигналу з мікроконтролера. Щоб уникнути таких небажаних ефектів і забезпечити стабільну роботу схеми, в електричний ланцюг бази вводиться резистор, підключений до нульового потенціалу – землі. Такий елемент має назву підтягуючого резистора, його схема подана на рисунку 2.5.

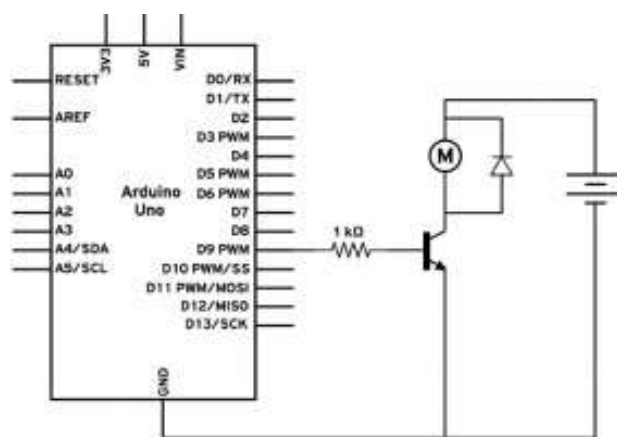


Рисунок 2.5 – Структурна схема підтягуючого резистора [13]

Суть його роботи полягає в тому, що він створює надійний шлях для витоку струму в ті моменти, коли активний сигнал на базу відсутній. У результаті роботи, базовий потенціал зберігається на низькому рівні і транзистор залишається у закритому стані. Резистор такого типу, зазвичай, має номінал

близько 10 кОм, що забезпечує достатній струм витoku без істотного навантаження на логічний вихід.

Використання цього резистора дозволяє виключити вплив паразитних сигналів і електромагнітних завад, які могли б призвести до непередбачуваних запусків двигуна або інших компонентів, що керуються транзистором. Таким чином, підтягуючий резистор виконує важливу функцію стабілізації логічного рівня на базі та захисту схеми від небажаних станів.

2.5 Особливості обміну даними по WI-FI

Обмін даними через Wi-Fi у двоколійній роботизованій системі на базі Arduino має низку важливих особливостей, які необхідно враховувати для забезпечення стабільної та надійної роботи [14]. Насамперед, мобільний характер платформи вимагає бездротового зв'язку з мінімальною затримкою, оскільки навіть незначні затримки можуть впливати на точність керування і стабільність руху. Для цього зазвичай використовуються Wi-Fi модулі на основі мікросхем, які можуть працювати, як клієнт або створювати власну точку доступу.

Залежно від реалізації, обмін може здійснюватися як із зовнішнім сервером, наприклад, через локальну мережу, так і безпосередньо між пристроями. У випадку двостороннього обміну даними часто реалізується протокол передачі команд і зворотного зв'язку для моніторингу стану роботи, наприклад, рівня заряду, швидкості, напрямку руху або показників з датчиків.

Важливу роль відіграє стабільність з'єднання, адже при втраті сигналу робот може залишитися без керування. Тому в програмному забезпеченні зазвичай реалізується система повторних спроб підключення, а також алгоритми поведінки при обриві зв'язку. Нерідко додається буфер команд або таймер аварійної зупинки.

Ще однією особливістю є вплив електромагнітних завад від моторів і живлення, що може спричинити перебої в передачі даних. Для мінімізації цих

ефектів використовуються стабілізовані джерела живлення для модуля Wi-Fi, розділення цифрових і силових кіл, а також фільтруючі елементи.

Оскільки двоколісна система є динамічно нестабільною, точність та швидкість обміну даними безпосередньо впливають на алгоритми балансування та стабілізації [15]. Саме тому, Wi-Fi модуль має не лише підтримувати швидке передавання даних, але й бути надійним у мобільному середовищі з частими змінами положення і рівня сигналу.

У цілому, використання Wi-Fi у такій роботизованій системі дозволяє реалізувати гнучке керування, віддалений моніторинг і навіть інтеграцію з хмарними сервісами, проте вимагає уважного підходу до технічної реалізації, щоб забезпечити необхідний рівень швидкодії, надійності й безпеки.

РОЗДІЛ 3

РЕАЛІЗАЦІЯ РОБОТИЗОВАНОЇ СИСТЕМИ

3.1 Опис логіки роботи програми

Алгоритм налаштування роботизованої системи на базі Arduino Uno полягає в наступному.

Для початку підключення використовується стандартний USB-кабель типу А-В. Кабель буде з'єднувати плату Arduino Uno з USB-портом комп'ютера. Після з'єднання живлення на платі автоматично активується, що супроводжується миготінням вбудованого світлодіода – індикатора живлення. Далі потрібно виконати запуск середовища розробки Arduino IDE. Спочатку здійснюється перевірка наявності підтримки плати Arduino Uno в програмованому середовищі IDE [16].

У більшості випадків, в тому числі у версії 2025 року Arduino IDE 2.3.6, підтримка плати Arduino Uno вже вбудована (рис. 3.1). На більш старих версіях можна відкрити меню Tools, перейти до пункту Board і переконатися, що обрано Arduino Uno зі списку доступних.

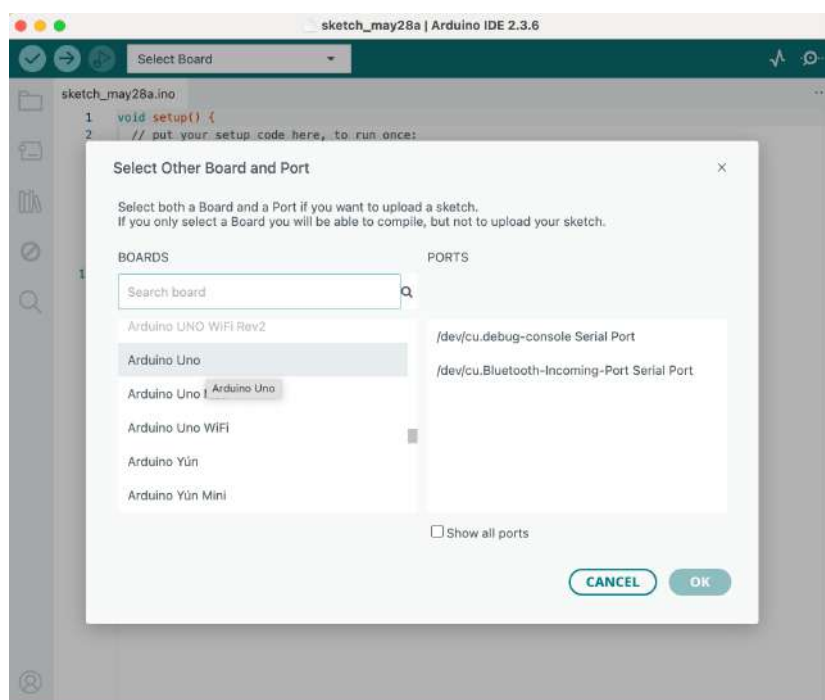


Рисунок 3.1 – Середовище проектування Arduino IDE 2.3.6

Наступний крок полягає у встановленні та виборі COM-порту. Після фізичного підключення плати система автоматично призначає їй COM-порт. У меню Tools вибирається пункт Port і там потрібно вказати порт, до якого підключена плата. Зазвичай він позначений як COMx, де x – це номер порту. Якщо порт не відображається, то необхідно встановити драйвер CH340 або інший, відповідно до типу чипа USB-UART на платі.

Далі зазвичай іде завантаження тестового прикладу для перевірки зв'язку. Для перевірки працездатності середовища потрібно скористатися вбудованим прикладом з меню Examples Arduino IDE, де серед базових прикладів вибирається скетч Blink (рис. 3.2). Цей код призначений для циклічного вмикання й вимикання вбудованого світлодіода на 13-му піні.



Рисунок 3.2 – Успішне завантаження скетчу Blink

Компіляція та завантаження програми на плату відбувається, коли натискається кнопка Verify компіляції коду. Якщо компіляція проходить без помилок, то код можна завантажити через Upload. Успішне завантаження підтверджується повідомленням у нижній частині вікна IDE та зміною поведінки світлодіода на платі.

Ініціалізація системи та перехід до основного коду керування роботом проходить після перевірки базової функціональності. Тоді можна переходити до завантаження основного скетчу, який реалізує логіку роботи роботизованої

системи. Це – код керування двигунами, потім обробки команд з датчиків і обміну даними по бездротовому інтерфейсу.

Після ініціалізації проходить тестування роботизованої платформи у реальних умовах. Після завантаження основної програми потрібно перевірити функціональність платформи у фізичному середовищі, а саме: перевірити роботу моторів, живлення, обробку команд, та відповідність дій заданій логіці.

3.2 Програмна реалізація колісного мобільного робота

Розробка колісного робота на платформі Arduino Uno передбачає рух за раніше зазначеним алгоритмом.

На початку основного циклу програми функцією loop виконується команда руху вперед. Для цього на пін IN1 подається високий рівень HIGH, а на пін IN2 – низький LOW. Це призводить до того, що мотор починає обертатися у визначеному напрямку, який механічно відповідає руху вперед мобільної платформи. Команда delay(2000) задає тривалість руху – 2000 мілісекунд, тобто 2 секунди. Протягом цього часу робот буде переміщуватися прямолінійно або дугоподібно, залежно від механіки коліс і розташування мотора. Фрагмент реалізації керування пінам IN1 та IN2 для руху вперед наведено у лістингу 3.1.

Лістинг 3.1 – Фрагмент коду для руху вперед

```
void loop() {
  // Рух вперед
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  delay(2000); // час руху 2 секунди
}
```

кінець лістингу 3.1

Після завершення руху робот повинен зробити зупинку. Для цього обидва керувальні піна IN1 і IN2 встановлюються в логічний нуль LOW. Це забезпечує повну відсутність сигналу на обмотках двигуна, внаслідок чого він перестає обертатися. Водночас викликається затримка delay(1000) – одна секунда, яка

гарантує паузу перед наступним рухом. Така пауза може бути корисною для стабілізації положення робота або очікування нової команди (лістинг 3.2).

Лістинг 3.2 – Фрагмент коду для зупинки

```
// Зупинка
digitalWrite(IN1, LOW);
digitalWrite(IN2, LOW);
delay(1000); // час зупинки 1 секунда
```

кінець лістингу 3.2

Після паузи знову викликається команда на рух вперед: IN1 = HIGH, IN2 = LOW, так само як у першому русі. Єдина відмінність полягає в тому, що цього разу тривалість руху становить лише 1000 мілісекунд. Це дозволяє, наприклад, доїхати до певної точки чи зробити скоригований маневр. Такий повторний рух може застосовуватись у сценаріях, де потрібно здійснити рух у два етапи – з перевіркою або паузою між ними (лістинг 3.3).

Лістинг 3.3 – Фрагмент коду для продовження руху

```
// Знову рух вперед
digitalWrite(IN1, HIGH);
digitalWrite(IN2, LOW);
delay(1000); // продовження руху на 1 секунду
```

кінець лістингу 3.3

На завершальному етапі виконання коду мотор остаточно зупиняється шляхом встановлення обох пінів у стан LOW, як і раніше. Однак цього разу додатково використовується конструкція `while (true);`, яка безкінечно затримує виконання програми (лістинг 3.4).

Лістинг 3.4 – Фрагмент коду для повної зупинки робота

```
// Повна зупинка
digitalWrite(IN1, LOW);
digitalWrite(IN2, LOW);

while (true); // Зупинити виконання
```

кінець лістингу 3.4

Це гарантує, що рух не буде повторюватись знову, оскільки код більше не переходить до початку функції `loop()`. Такий підхід дозволяє створити одноразовий маневр, який не повторюється без перезапуску контролера.

Якщо потрібно, щоб робот рухався знову за заданим алгоритмом, не зупиняючись, то можна зробити програму циклічною, але тоді дії будуть повторюватись нескінченно. Для цього достатньо прибрати рядок перевірки умови `while (true)` в кінці фрагмента коду зупинки.

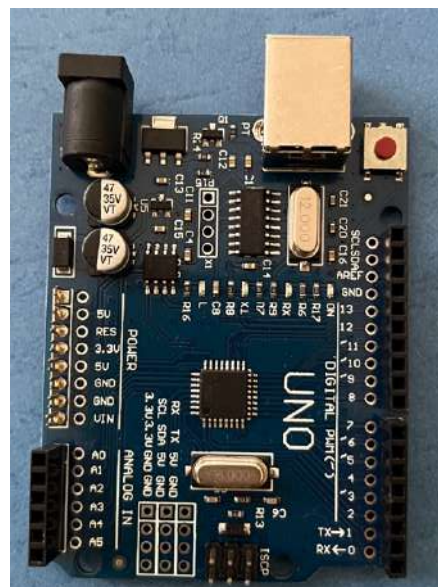
3.3 Фізична збірка пристрою

У процесі реалізації мобільного робота було здійснено послідовне збирання апаратної частини системи на базі двоколісного шасі.

Основу складала готова монтажна платформа (рис. 3.3 а), де отвори в пластині дозволяють гнучко розміщувати деталі відповідно до конкретного проєкту. Також основним мозком платформи є плата Arduino (рис. 3.3 б), до якої підключено виконавчі елементи та модулі.



а)



б)

Рисунок 3.3 – Основні елементи системи: платформа (а) та плата (б)

Форма з вирізами по краях – це стандартна деталь для складання двоколісних роботів, вона призначена для монтажу приводних коліс та забезпечення їх вільного обертання без перешкод. Плата Arduino розміщується на верхній частині шасі для забезпечення легкого доступу до інтерфейсів програмування та живлення.

Для руху використовуються два привідні колеса (рис. 3.4 а) розміром 27х66 мм, жорстко з'єднані з двигунами постійного струму, з'єднувальні дюпон кабелі типу мама-мама, мама-тато, тато-тато довжиною 10 см (рис. 3.4 б) і ультразвуковий датчик відстані HC-SR04 (рис. 3.4 в).

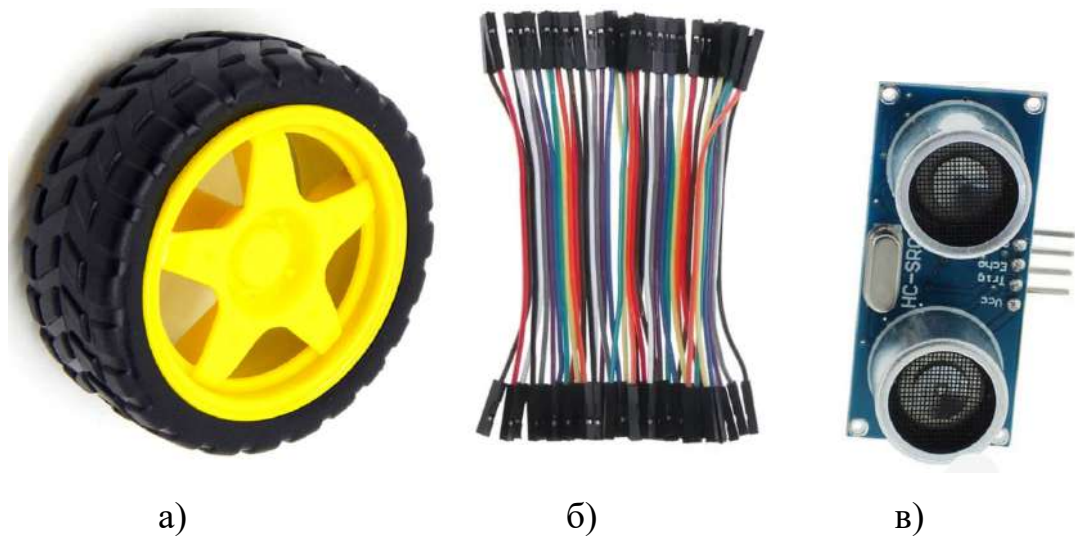


Рисунок 3.4 – Додаткові компоненти платформи: робототехнічні колеса (а), з'єднувальні кабелі (б) та ультразвуковий датчик відстані (в)

Конструкція забезпечує стійкість та здатність до маневрування за рахунок розміщення коліс з обох боків корпусу та наявності додаткової опори або ролика ззаду.

Живлення системи здійснюється за допомогою акумуляторного блоку, розміщеного під платформою, або через зовнішнє джерело, підключене до роз'єму живлення контролера. Окремо встановлено драйвер для керування мотором, який відповідає за правильну комутацію напрямку обертання. Всі

елементи закріплено на корпусі за допомогою гвинтів або двостороннього монтажного скотчу.

Після встановлення усіх основних елементів конструкції, важливо правильно з'єднати мікроконтролер з усіма сенсорами та модулями за допомогою проводів у вигляді dupont-кабелів, які забезпечують надійний контакт без пайки.

На рисунку 3.5 видно, що до плати Arduino Uno підключені кольорові дроти, які йдуть від різних датчиків і модулів.

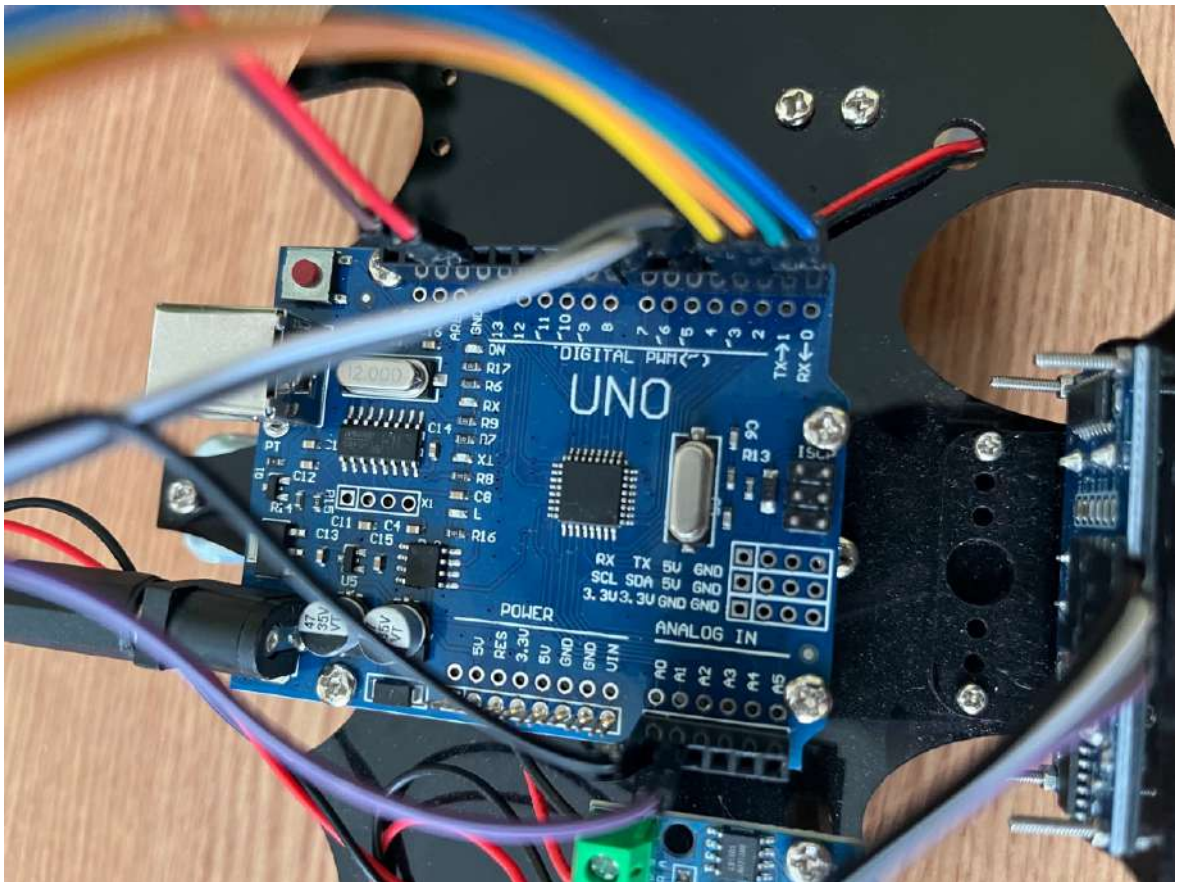


Рисунок 3.5 – З'єднання елементів на основі мікроконтролера

Основні з'єднання виконані через пінові ряди наступним чином. Живлення 5V подається з плати до модулів, GND виступає, як спільна земля для всіх підключених пристроїв, а цифрові входи/виходи використовуються для керування датчиками, двигунами та прийому сигналів (наприклад, з ультразвукового датчика). Для підключення I2C та UART пристроїв використано пари SCL/SDA або TX/RX відповідно.

Завершення збірки включає перевірку надійності з'єднань, тестування моторів та ініціалізацію мікроконтролера через Arduino IDE. Фізична структура робота є компактною (рис. 3.6), мобільною і придатною для подальших експериментів, розширення функціоналу або встановлення додаткових модулів.

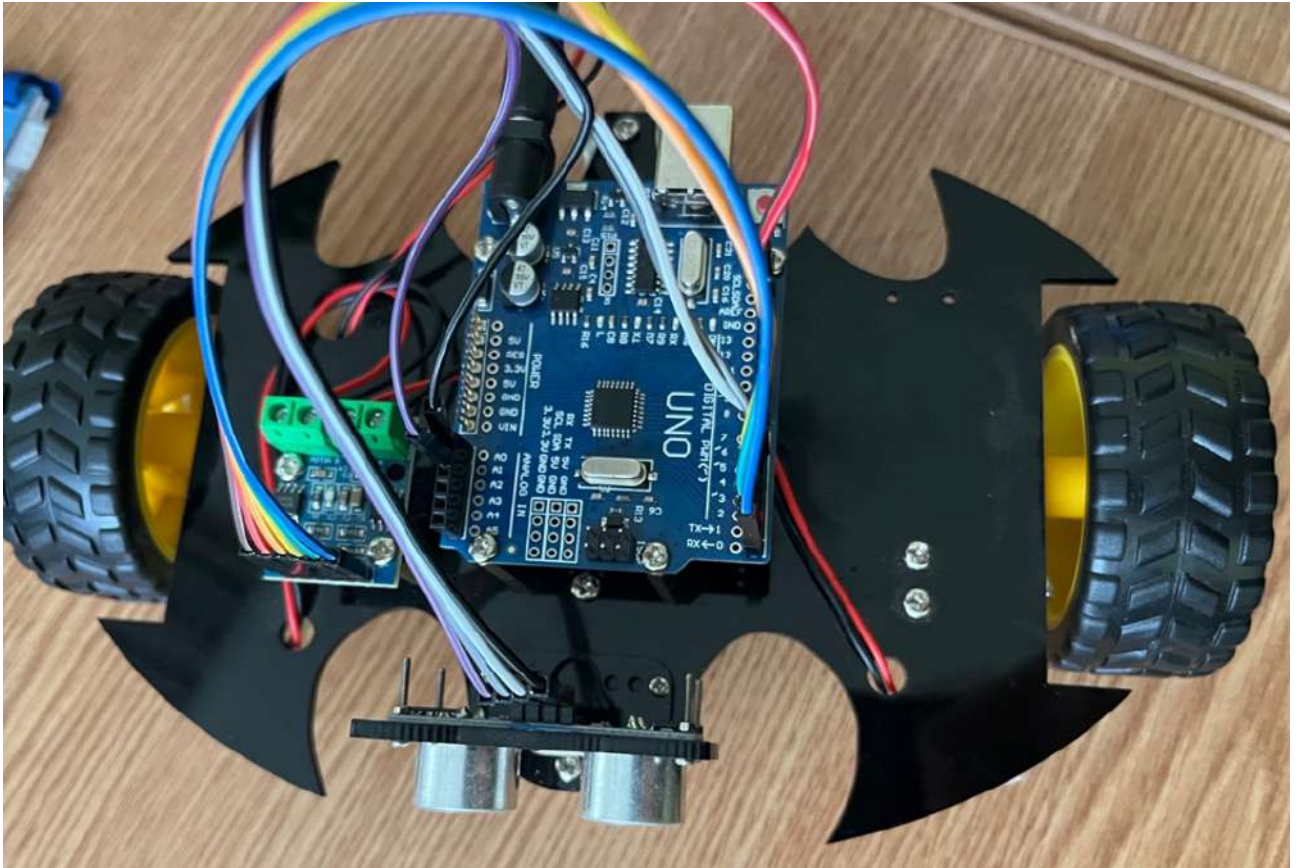


Рисунок 3.6 – Загальний вигляд роботизованої платформи

Таким чином, після підключення всіх компонентів та перевірки з'єднань, конструкція повністю готова до подальшого програмування та тестування.

3.4 Тестування роботи платформи та вимірювання точності датчиків

Після завершення збірки роботизованої платформи було проведено первинне тестування її функціональності. Основна увага приділялась перевірці правильності з'єднань, стабільності живлення та реакції мікроконтролера на

введені команди. Для цього використовувалися прості програми з бібліотек Arduino IDE, які дозволяли поетапно перевіряти роботу кожного модуля.

Особливу увагу приділено роботі ультразвукового датчика, оскільки точність його вимірювань є критичною для навігації. Було створено тестову програму, яка зчитувала дані з датчика та виводила результати у серійний монітор. Порівнюючи відстані, що визначав датчик, із реальними вимірами, вдалося оцінити похибку. В середньому вона становила близько 1-2 см, що є прийнятним для більшості задач початкового рівня.

Також перевірялась реакція платформи на перешкоди. Під час наближення до об'єктів робот вчасно реагував на зміни відстані, зупиняючись або змінюючи напрямок руху. Це свідчить про коректну інтеграцію програмного забезпечення з сенсорною частиною. Було також протестовано різні сценарії – прямий рух, об'їзд об'єкта, зупинка – що допомогло переконатися у стабільній роботі датчика у різних умовах.

Важливою частиною тестування стало спостереження за тим, як змінюється точність датчика залежно від умов освітлення, поверхні об'єкта та кута нахилу. Найкращі результати спостерігались при наявності плоских, перпендикулярних поверхонь. При нахилених або м'яких поверхнях точність дещо знижувалась, однак залишалась у допустимих межах для роботи в межах навчального проєкту.

Загалом тестування підтвердило, що платформа працює відповідно до заданих параметрів, а точність ультразвукового датчика достатня для базової навігації. Надалі можливе вдосконалення шляхом калібрування, додавання додаткових сенсорів або використання фільтрації даних для покращення стабільності показників.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було спроектовано та розроблено мобільну роботизовану платформу на базі мікроконтролера Arduino, що забезпечує автоматизоване пересування.

Проведений аналіз сучасних мікроконтролерних систем показав, які платформи найкраще підходять для конкретних завдань і як ефективно їх використовувати. Визначено, що платформа Arduino забезпечує достатній набір входів/виходів для підключення датчиків і виконавчих механізмів, має просту архітектуру та працює на стабільній частоті, що робить її надійною для базових задач керування

Розроблено поетапний підхід проектування роботизованої системи і визначено ключові ланки у розробці: вибір платформи, мови програмування, середовища розробки, визначення апаратних складових (датчики, виконавчі пристрої, комунікаційні модулі), збірку та тестування.

Сформовано алгоритм трекингу руху та визначено, що платформа має виконувати таку послідовність дій: прямолінійний рух вперед протягом 2 секунд, короткої зупинки на 1 секунду, потім розворот на місці шляхом обертання вліво протягом 1 секунди та остаточної зупинки.

Визначено, що використання транзистора в парі з захисним діодом утворює ефективний і безпечний засіб керування навантаженнями та забезпечують стабілізацію роботи системи.

Реалізовано програмне забезпечення для руху та зупинки платформи з різними часовими проміжками.

Проведено фізичну збірку роботизованої платформи, що складається з монтажної платформи, плати Arduino, привідних коліс, з'єднувальних дюпон кабелів, ультразвукового датчика відстані HC-SR04 та акумуляторного блоку.

Виконано експериментальне дослідження роботи створеної системи, визначено, що платформа працює відповідно до заданих параметрів, а точність ультразвукового датчика достатня для базової навігації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Smart agriculture monitoring system using ATMEGA328P. *International Research Journal of Modernization in Engineering Technology and Science*. 2023. URL: <https://doi.org/10.56726/irjmets32859> (date of access: 27.04.2025).
2. Radar Using Arduino UNO. Thele et al. *International Journal of Research Publication and Reviews*. 2024. Vol. 5, no. 4. P. 706-764.
3. UNO Rev3 or UNO R4? Choosing the perfect Arduino for your project. *Arduino*. URL: <https://blog.arduino.cc/2024/12/18/uno-rev3-or-uno-r4-choosing-the-perfect-arduino-for-your-project/?queryID=undefined> (date of access: 14.05.2025).
4. ESP8266. *Вікіпедія*. URL: <https://uk.wikipedia.org/wiki/ESP8266> (date of access: 25.04.2025).
5. ESP32 Wi-Fi and Bluetooth SoC Espressif Systems. *Wireless SoCs, Software, Cloud and AIoT Solutions*. URL: <https://www.espressif.com/en/products/socs/esp32> (date of access: 25.04.2025).
6. Контролер для розробки STM32F103C8T6 ARM 32 Cortex-M3 STM32. *ROZETKA*. URL: <https://surl.lu/wzmyfv> (дата звернення: 26.04.2025).
7. Choosing the Right Microcontroller: A Comparison of Arduino, ESP8266, ESP32, and STM32f4 series. *Echcrusading*. URL: <https://surl.li/nryitb> (date of access: 28.04.2025).
8. Radar Using Arduino UNO / S. Thele et al. *International Journal of Research Publication and Reviews*. 2024. Vol. 5, no. 4. P. 760-793. URL: <https://doi.org/10.55248/gengpi.5.0424.10113> (date of access: 28.05.2025).
9. Braunl T. Raspberry Pi. *Embedded Robotics*. Singapore, 2022. P. 67-83. URL: https://doi.org/10.1007/978-981-16-0804-9_4 (date of access: 08.05.2025).
10. Tailor H. ESP32 vs Arduino Speed Comparison. *Makerguides*. URL: <https://www.makerguides.com/esp32-vs-arduino-speed-comparison/> (date of access: 12.05.2025).
11. Murugan M. M. S., E. R., M. V. Colour Detection Based Robotic Cart Using Raspberry Pi. *International Journal of Advanced Research in Science, Communication*

and Technology. 2021. P. 615–620. URL: <https://doi.org/10.48175/ijarsct-1069> (date of access: 13.05.2025).

12. Фільтр Калмана – Вікіпедія. *Вікіпедія*. URL: https://uk.wikipedia.org/wiki/Фільтр_Калмана (дата звернення: 28.04.2025).

13. Arduino – управління двигунами постійного струму. *Geekmatic*. URL: https://geekmatic.in.ua/ua/arduino_motor_control (дата звернення: 01.05.2025).

14. Олещенко Л.М., Мошенський А.О. Апаратно-програмна система моніторингу та аналізу показників навколишнього середовища на основі контролера ESP8266. *Вчені записки ТНУ імені В.І. Вернадського. Серія: Технічні науки*. Т. 34 (73). № 4. 2023. С. 84-92.

15. Aziz A., Zahra A. Prototype Design of Landfill Gas Pipe Leak Monitoring System Based on Microcontroller Node MCU ESP8266 with the Internet of Things Method. *International Journal of Intelligent Systems and Applications in Engineering*. 2023. Vol. 11(2). P. 133-147.

16. Thopate K., Shirbavikar K., Kulkarni R.V., Phursule R., Kale V., Karjule S., Suhas K., Kale S. Enhancing Ultrasonic Sensor Goggles for Blinds Using Node MCU ESP8266 Microprocessor. *International Journal of Intelligent Systems and Applications in Engineering*. 2023. Vol. 12(1). P. 611-618.