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»

АВТОНОМНИЙ РОБОТ НА ПЛАТФОРМІ ARDUINO ДЛЯ СЛІДУВАННЯ ЗА РУХОМИМ ОБ'ЄКТОМ

AUTONOMOUS ROBOT ON THE ARDUINO PLATFORM FOR TRACKINGA MOVING OBJECT

спеціальність 123 Комп'ютерна інженерія
(шифр і назва спеціальності)

освітня програма Комп'ютерна інженерія
(назва освітньої програми)

Виконав: здобувач вищої освіти
групи КІ-42
Драченко Назарій Олександрович

(підпис)

Керівник:
к.т.н., доцент
Бортник Катерина Яківна

(підпис)

Кваліфікаційну роботу
допущено до захисту
« 04 » 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
_____ доц. Тарас ТЕРЛЕЦЬКИЙ
« 10 » 01 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Драченко Назару Олександровичу
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи *Автономний робот на платформі Arduino для слідування за рухомим об'єктом*

Керівник роботи *к.т.н., доц. Бортник Катерина Яківна*

затверджені наказом закладу вищої освіти від «04» січня 2025 року № 11/01-02
2. Строк подання здобувачем вищої освіти кваліфікаційної роботи 10.06.2025р.

3. Вихідні дані до роботи *джерелом розробки є науково-технічна література та публікації в періодичних виданнях з даного питання, опубліковані зарубіжні та вітчизняні роботи в даній області та різні інтернет-ресурси технічного спрямування*

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):
Вступ
Аналіз стану розвитку автономних роботів
Апаратна реалізація системи
Розробка автономного робота та його програмування
Тестування автономного робота
Висновки

5. Перелік графічного (ілюстративного) матеріалу:
Плата Arduino UNO R3
Інтегроване середовище розробки Arduino (IDE)
Модуль Bluetooth HC-05
Макетна схема підключення елементів робота

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Огляд літератури про розвиток автономних роботів	*Бортник К.Я., доцент*		
Апаратна реалізація системи	*Бортник К.Я., доцент*		
Розробка автономного робота та його програмування	*Бортник К.Я., доцент*		
Нормоконтроль	*Багнюк Н.В., доцент*		
Гарант ОП	*Лавренчук С.В., доцент*		
Показник запозичень тексту	***___%***		
Академічна доброчесність	*Міскевич О.І.,ст.викладач*		

7. Дата видачі завдання 10.01.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	*Огляд літератури із досліджуваної проблеми, аналіз предметної області та наявних рішень*	до 10.02.2025 р.	Виконано
2.	*Апаратна реалізація системи*	до 02.03.2025 р.	Виконано
3.	*Розробка автономного робота та його програмування*	до 02.04.2025 р.	Виконано
4.	*Висновки та пропозиції*	до 10.04.2025 р.	Виконано
5.	*Формування списку використаних джерел*	до 15.04.2025 р.	Виконано
6.	*Формування додатків*	до 02.05.2025 р.	Виконано
7.	*Оформлення ілюстративного матеріалу*	до 10.05.2025 р.	Виконано
8.	*Представлення остаточного варіанту кваліфікаційної роботи бакалавра керівникові*	до 15.05.2025 р.	Виконано
9.	*Нормоконтроль*	до 30.05.2025 р.	Виконано
10	*Інструментальна перевірка на академічний плагіат*	до 03.06.2025 р	Виконано
11.	*Здача кваліфікаційної роботи та всіх супровідних документів на кафедру*	до 10.06.2025 р.	Виконано

Здобувач вищої освіти _____ Драченко Н.О.
(підпис) (прізвище, ініціали)

Керівник кваліфікаційної роботи _____ Бортник К.Я.
(підпис) (прізвище, ініціали)

АНОТАЦІЯ

Драченко Н. О. Автономний робот на платформі Arduino для слідування за рухомим об'єктом. 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, переліку використаних джерел, додатків.

У першому розділі розглянуто основні визначення, класифікацію та сучасний стан розвитку автономних роботів. Особливу увагу приділено архітектурі систем керування, методам ухвалення рішень та виявлення об'єктів, що є ключовими аспектами для побудови ефективної автономної системи.

У другому розділі роботи детально описано апаратну реалізацію розроблюваного прототипу, включаючи використані компоненти: плату Arduino Uno, ультразвуковий датчик відстані HC-SR04, мотор-щит Arduino Motor Shield L298P, а також сервопривід MG90S Micro Servo. Подано технічні характеристики, схеми підключення та принципи роботи кожного з модулів.

У третьому розділі висвітлено процес створення прототипу автономного робота, розробку та реалізацію схеми підключення модулів, а також програмування мікроконтролера Arduino. Окремо проведено тестування створеної системи, під час якого перевірено коректність роботи апаратного забезпечення та програмного забезпечення, а також відповідність розробленої системи технічним вимогам.

Ключові слова: автономний робот, Arduino Uno, HC-SR04, Motor Shield L298P, сервопривід MG90S, програмування Arduino, ультразвуковий сенсор.

ANNOTATION

Drachenko N. Autonomous robot on the Arduino platform for tracking a moving object. 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 chapters, conclusions, a list of used sources, and appendices.

The first chapter considers the basic definitions, classification, and current state of development of autonomous robots. Particular attention is paid to the architecture of control systems, methods of decision-making, and object detection, which are key aspects for building an effective autonomous system.

The second chapter of the work describes in detail the hardware implementation of the prototype being developed, including the components used: the Arduino Uno board, the HC-SR04 ultrasonic distance sensor, the Arduino Motor Shield L298P motor shield, and the MG90S Micro Servo servo drive. Technical characteristics, connection diagrams and operating principles of each module are presented.

The third section covers the process of creating a prototype of an autonomous robot, developing and implementing a module connection diagram, as well as programming the Arduino microcontroller. Separately, the created system was tested, during which the correct operation of the hardware and software was checked, as well as the compliance of the developed system with technical requirements.

Keywords: autonomous robot, Arduino Uno, HC-SR04, Motor Shield L298P, MG90S servo drive, Arduino programming, ultrasonic sensor..

ЗМІСТ

ВСТУП

У сучасних умовах розвитку інформаційних технологій і робототехніки особливого значення набувають автономні мобільні системи, здатні виконувати завдання без постійного втручання оператора. Такі роботи знаходять широке застосування в логістиці, охороні, військовій сфері, рятувальних операціях, а також у побуті. Здатність системи самостійно пересуватися, орієнтуватися в просторі й реагувати на зміну середовища відкриває нові можливості для автоматизації процесів, підвищення безпеки й ефективності діяльності людини.

Сучасні автономні роботи базуються на поєднанні сенсорних технологій, алгоритмів обробки даних, систем комп'ютерного зору та енергоефективних приводів. Особливе місце в їхній розробці посідають мікроконтролери, такі як Arduino, ESP32 або Raspberry Pi, які завдяки відкритим апаратно-програмним платформам сприяють швидкому впровадженню інновацій навіть у малих дослідницьких лабораторіях і навчальних закладах. Проте навіть через помітні досягнення в цій галузі, залишається низка технічних і методологічних викликів, пов'язаних із підвищенням автономності, надійності, точності роботи в умовах складного середовища та ефективного використання ресурсів.

Одним із найактуальніших напрямів є розробка роботів, які можуть слідувати за рухомим об'єктом у режимі реального часу. Такі системи потребують інтеграції різних типів сенсорів (ультразвукових, інфрачервоних, візуальних), алгоритмів обробки даних, ефективних механізмів управління приводами та енергоефективних рішень. Платформа Arduino, завдяки своїй простоті, відкритості та великій спільноті користувачів, є ідеальним вибором для розробки прототипів автономних роботів із функцією стеження.

Актуальність обраної теми обумовлена потребою в доступних і простих у реалізації рішеннях для створення автономних систем спостереження та супроводу, які можуть бути застосовані як у дослідницьких цілях, так і в практичних завданнях. Важливою є також навчальна значимість проєкту,

оскільки реалізація таких систем дозволяє глибше зрозуміти принципи роботи сенсорних модулів, алгоритмів управління та програмування мікроконтролерів.

Метою кваліфікаційної роботи є розробка автономного робота на платформі Arduino, здатного виявляти рухомий об’єкт та слідувати за ним, використовуючи сенсорні дані та алгоритми управління.

Об’єктом дослідження є автономна робототехнічна система.

Предметом дослідження є апаратні й програмні засоби побудови системи автономного слідування

Для досягнення даної мети передбачено вирішення наступних завдань:

– здійснити аналіз існуючих рішень і підходів до реалізації систем слідування;

– обґрунтувати вибір апаратних компонентів і програмного забезпечення;

– розробити та протестувати алгоритм автономного слідування;

– провести експериментальне дослідження розробленого автономного робота.

РОЗДІЛ 1
АНАЛІЗ СТАНУ РОЗВИТКУ АВТОНОМНИХ РОБОТІВ

1.1 Визначення та класифікація автономних роботів

За останнє десятиріччя автономні системи стали ключовими елементами концепції індустрії 4.0. Їхнє впровадження дозволяє значно підвищити ефективність виробництва, зменшити залежність від людського ресурсу та збільшити гнучкість у багатьох секторах економіки [1]. Широкий розвиток сенсорних технологій, мікроелектроніки, обчислювальних платформ і програмного забезпечення стимулює створення як простих побутових систем, так і складних автономних агентів для дослідницьких і військових задач.

Автономні роботи – це класи робототехнічних систем, які здатні виконувати завдання без безпосереднього керування людиною, використовуючи вбудовані сенсори, програмне забезпечення та алгоритми прийняття рішень. За визначенням [2] автономний робот – це система, яка може взаємодіяти з навколишнім середовищем, сприймати його зміни, аналізувати отримані дані та на основі цього приймати рішення про подальші дії. Такий підхід забезпечує здатність робота адаптуватися до змін умов, діяти у реальному часі та виконувати складні завдання навіть у непередбачуваних сценаріях.

Основні характеристики автономних роботів:

– наявність сенсорних систем для сприйняття навколишнього середовища (ультразвукові, оптичні, інфрачервоні, лазерні датчики, камери);

– інтеграція обчислювальних ресурсів (мікроконтролери, процесори, штучні нейронні мережі) для обробки інформації;

– виконавчі механізми (приводи, маніпулятори, колісні або гусеничні шасі);

– програмне забезпечення для автономного управління та адаптації поведінки.

Згідно джерела [3], автономні роботи класифікують за кількома ключовими ознаками, а саме:

– за типом пересування;

– за сферою використання;

– за рівнем автономності.

На рисунку 1.1 зображено блок-схему класифікацій автономних роботів за типом пересування, сферою використання та рівнем автономності.

Рисунок 1.1 – Класифікації автономних роботів

За типом пересування:

1) наземні автономні роботи – це системи, що пересуваються по поверхні землі, використовуючи колісні, гусеничні або крокуючі механізми. Вони широко застосовуються у логістиці (автоматичні візки, склади Amazon), сільському господарстві (автономні трактори), рятувальних операціях (роботи для виявлення людей під завалами) та військовій справі (розвідувальні платформи, розмінувальні роботи). На рисунку 1.2 зображено чотириногий робот Boston Dynamics Spot, що використовує алгоритми комп'ютерного зору для навігації на пересіченій місцевості;

Рисунок 1.2 – Чотириногий робот Boston Dynamics Spot [4]

2) повітряні роботи – це дрони та безпілотні літальні апарати (БПЛА), які виконують завдання на основі автоматичної навігації у повітряному просторі. Їх використовують для аерозйомки, моніторингу, доставки товарів, картографування, пошуково-рятувальних операцій, військової розвідки. На рисунку 1.3 зображено DJI Matrice 300 RTK – дрон із підтримкою штучного інтелекту, здатний здійснювати польоти за заданими маршрутами та уникати перешкод;

Рисунок 1.3 – DJI Matrice 300 RTK [5]

3) водні автономні роботи діляться на два класи: апарати, що діють на поверхні води (наприклад, розвідувальні катери), та підводні апарати (AUV – Autonomous Underwater Vehicles). Вони виконують завдання моніторингу екосистем, дослідження морського дна, обслуговування інженерних конструкцій, виявлення об'єктів. На рисунку 1.4 зображено автономний підводний апарат Bluefin-21, який застосовується для морських досліджень і пошукових операцій.

Рисунок 1.4 – Автономний підводний апарат Bluefin-21 [6]

За сферою використання:

1) промислові автономні роботи використовуються на заводах, фабриках та автоматизованих виробничих лініях. Їхнє основне призначення – виконання повторюваних завдань із високою точністю та швидкістю, що дозволяє зменшити витрати на виробництво та підвищити якість продукції. На рисунку 1.5 зображено роботи-маніпулятори на автомобільному заводі;

Рисунок 1.5 – Робот-маніпулятор [7]

2) сервісні роботи призначені для взаємодії з людьми або виконання завдань у публічних просторах (готелі, лікарні, аеропорти, торгові центри). Вони можуть допомагати, консультувати, прибирати, доставляти дрібні вантажі. На рисунку 1.6 зображено інтелектуальний комерційний робот-гуманоїдний сервіс;

Рисунок 1.6 – Інтелектуальний комерційний робот [8]

3) військові автономні роботи є важливим класом робототехнічних систем, розроблених для виконання спеціалізованих завдань у сфері оборони та безпеки. Вони призначені для підвищення ефективності бойових операцій, зниження ризиків для життя військовослужбовців, а також автоматизації рутинних і небезпечних завдань у бойових і надзвичайних умовах. Ці системи здатні самостійно пересуватися, орієнтуватися на місцевості, аналізувати ситуацію та виконувати конкретні завдання на основі даних, що надходять із сенсорних систем. Однією з основних підгруп є розвідувальні автономні роботи, які забезпечують збір інформації з місця бойових дій, виявляють пересування противника, моніторять місцевість і передають дані в режимі реального часу.

Такі роботи можуть працювати як на суші, так і у повітрі, забезпечуючи оперативну підтримку командуванню. Прикладом є системи безпілотних літальних апаратів, таких як MQ-9 Reaper, які використовуються для проведення повітряної розвідки та точкових ударів, поєднуючи високу швидкість польоту з можливістю довготривалого патрулювання. Важливе місце серед військових роботів посідають системи розмінування, які оснащені маніпуляторами, камерами та спеціальними датчиками для виявлення вибухових пристроїв. Вони використовуються для очищення території, розмінування мінних полів, перевірки підозрілих об’єктів. Прикладом є платформи, подібні до QinetiQ TALON, які мають захист від вибухових хвиль, водонепроникний корпус і дистанційне керування. На рисунку 1.7 зображено гусеничний військовий робот TALON;

Рисунок 1.7 – Гусеничний військовий робот TALON [9]

4) дослідницькі автономні роботи застосовуються у наукових експедиціях, космічних місіях, морських дослідженнях. Вони збирають дані в умовах, що недоступні або небезпечні для людини. На рисунку 1.8 зображено Марсохід NASA.

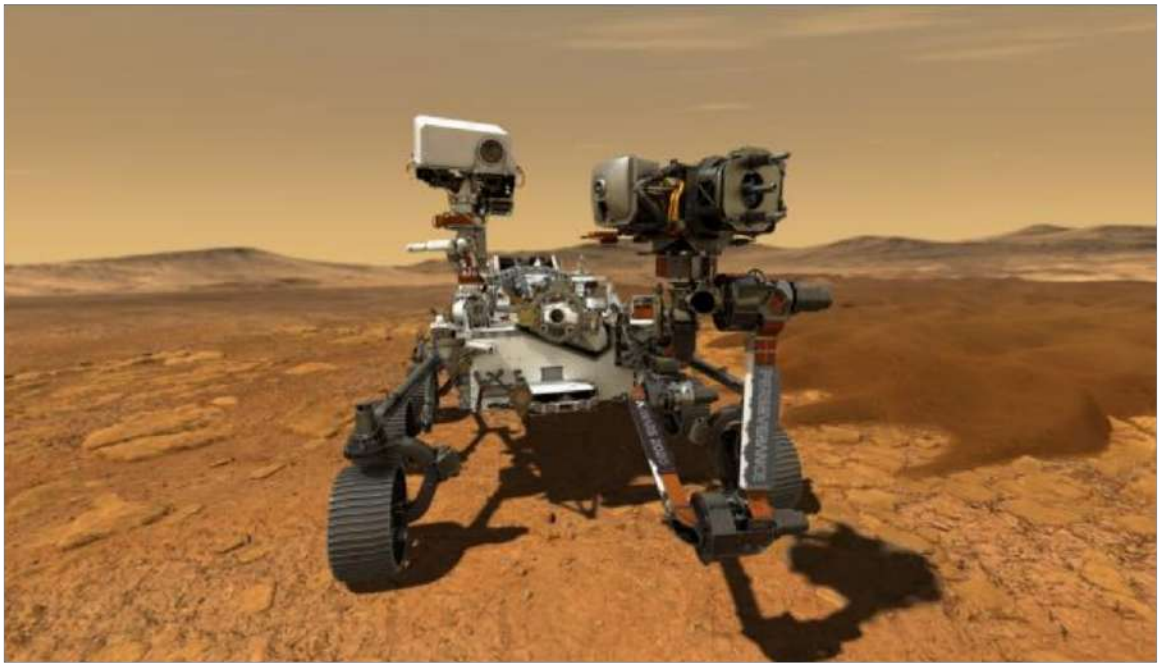

Рисунок 1.8 – Mars-2020 [10]

Класифікація автономних роботів за рівнем автономності є одним із ключових аспектів при аналізі їхніх функціональних можливостей. Рівень автономності визначається тим, наскільки система здатна виконувати завдання самостійно, без втручання оператора, та наскільки складні рішення вона може приймати в процесі своєї роботи. Важливо підкреслити, що ступінь автономності безпосередньо залежить від технічних характеристик робототехнічної системи, зокрема наявності обчислювальних ресурсів, сенсорних даних, алгоритмів управління й здатності до адаптації в реальному середовищі [11].

На нижчому рівні автономності перебувають дистанційно керовані системи. Такі роботи не приймають рішень самостійно й повністю залежать від команди оператора. Вони використовуються переважно там, де потрібна точна маніпуляція або контроль, наприклад, під час розмінування або виконання ремонтних робіт у складних умовах.

Частково автономні системи, або роботи з частковим рівнем автономності, можуть виконувати певні завдання без безпосереднього контролю з боку людини, але для корекції курсу, зміни поведінки чи прийняття рішень у

непередбачених ситуаціях потребують втручання оператора. Такі системи здебільшого функціонують за заданими алгоритмами, які дозволяють реалізувати прості сценарії поведінки, наприклад, утримання траєкторії, уникнення перешкод, автоматичне слідування за міткою. Прикладом можуть бути логістичні роботи на складах, які переміщаються між пунктами завантаження та розвантаження, використовуючи прості алгоритми планування маршруту [11].

Найвищий рівень займають повністю автономні системи. Вони здатні самостійно аналізувати ситуацію, приймати рішення в реальному часі, будувати карти середовища, визначати оптимальні траєкторії руху, а також змінювати свою поведінку залежно від зовнішніх факторів. У таких системах широко використовуються алгоритми штучного інтелекту, машинного навчання, нейронні мережі, а також багатосенсорні комплекси, що дозволяють отримувати повноцінну картину навколишнього середовища. Прикладами є сучасні автономні транспортні засоби, марсоходи NASA, військові системи, здатні виконувати тактичні завдання без участі оператора [11].

Таким чином, рівень автономності є ключовим параметром, який визначає технічні можливості, сценарії застосування та перспективи розвитку робототехнічних систем. Глибоке розуміння цього аспекту є необхідним для проектування ефективних, безпечних та етичних автономних рішень.

1.2 Архітектура й методи керування автономним роботом

Управління роботами є одним із ключових напрямів робототехніки, оскільки саме воно визначає, як система реалізує поставлені завдання, реагує на зміни навколишнього середовища й підтримує необхідну точність і стабільність роботи. Основою ефективного управління є розробка моделі робота – математичного чи програмного опису, що відображає його рух, механіку та взаємодію з об’єктами.

Кінематичні моделі описують співвідношення між керуючими сигналами (наприклад, швидкостями приводів) та рухом робота у просторі, що дозволяє будувати траєкторії та керувати положенням. Динамічні моделі розширюють ці описи, враховуючи сили, моменти, інерцію та інші фізичні впливи, що є критично важливим для точного прогнозування та компенсації рухів у реальних умовах експлуатації [12].

Методи управління роботами поділяють на два основні класи: зворотного зв'язку (feedback control) та без зворотного зв'язку (feedforward control). У системах зі зворотним зв'язком робот порівнює фактичний результат із заданим, визначаючи відхилення та коригуючи свою поведінку. Цей підхід широко застосовується в задачах стабілізації, відстеження траєкторії, а також компенсації зовнішніх збурень. Методи без зворотного зв'язку ґрунтуються на прогнозі поведінки системи та зазвичай використовуються там, де точні вимірювання стану неможливі або є затримки у сенсорних даних.

Архітектура системи керування роботом – це структура, що визначає спосіб організації апаратних і програмних компонентів, відповідальних за виконання завдань, обробку сенсорних даних та ухвалення рішень. Вона забезпечує взаємодію між блоками сприйняття, планування, прийняття рішень і управління виконавчими механізмами, створюючи єдину функціональну систему.

На нижньому рівні системи управління розташовані датчики (ультразвукові, інфрачервоні, візуальні, LIDAR), які збирають інформацію про навколишнє середовище. Далі ця інформація передається на рівень обробки даних, де за допомогою алгоритмів (наприклад, фільтрів Калмана або нечіткої логіки) вона очищується від шумів та формує основу для ухвалення рішень. На найвищому рівні система управління використовує ці дані для планування траєкторій, оптимізації маршрутів і визначення команд для виконавчих механізмів.

На рисунку 1.9 представлено загальну структурну схему роботи автономного робота. Сенсори виступають основним джерелом даних про

навколишнє середовище: ультразвукові забезпечують вимірювання відстані до об’єктів, інфрачервоні дозволяють виявляти теплові сигнали або перешкоди, візуальні сенсори (камери) збирають інформацію для аналізу образів, а LIDAR-датчики формують лазерне сканування простору, що дає змогу будувати точну картографію.

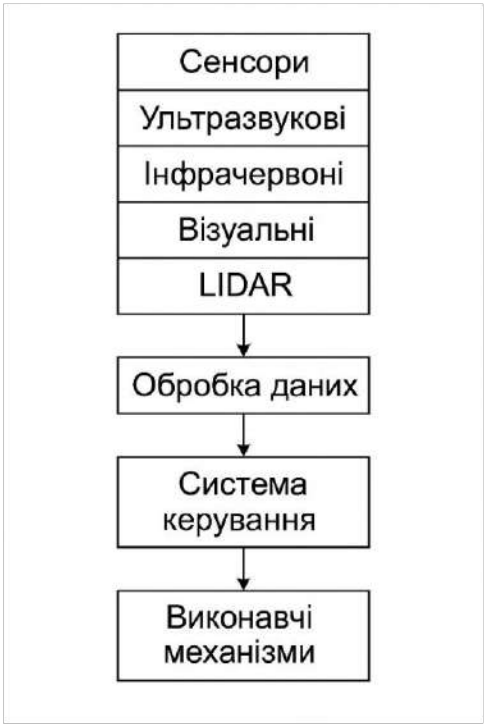

Рисунок 1.9 – Архітектура системи керування мобільним роботом

Отримані дані передаються на блок обробки даних, де вони проходять етапи фільтрації, нормалізації та аналізу. На цьому етапі система визначає ключові параметри, такі як наявність перешкод, відстань до цілі, швидкість руху об’єкта, а також просторові характеристики середовища.

Наступним елементом є система керування, що реалізує центральний алгоритм ухвалення рішень. На основі обробленої інформації система формує стратегію дій: визначає траєкторію руху, ухвалює рішення щодо зупинки, маневрування або зміни режиму роботи. Саме цей блок виступає логічним ядром усієї системи.

Завершальним етапом є блок виконавчих механізмів. До нього належать електромотори, сервоприводи, маніпулятори або інші пристрої, що фізично реалізують команди системи керування. Вони відповідають за рух платформи, повороти, зміни швидкості або виконання специфічних завдань, що поставлені перед роботом.

Сукупність цих блоків формує цілісну систему, здатну в режимі реального часу реагувати на зміну середовища, демонструвати автономність дій та виконувати завдання без постійного контролю з боку оператора.

Методи керування роботами можна розділити на класичні (лінійні) й сучасні (адаптивні, інтелектуальні). Серед класичних методів найпоширенішим є ПІД-регулювання (рис. 1.10), яке забезпечує стабільне відстеження заданих параметрів за рахунок пропорційного, інтегрального й диференціального впливів. Такий підхід ідеально підходить для систем із добре визначеними характеристиками, де важлива простота й надійність.

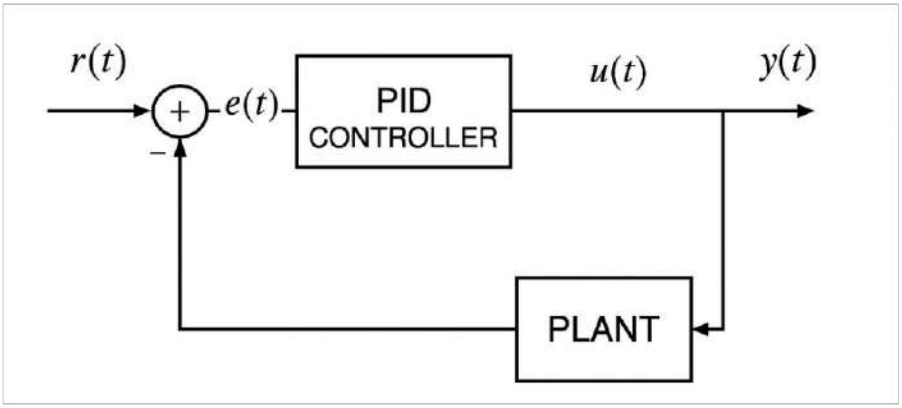

Рисунок 1.10 – Принципова схема ПІД-регулятора в системі управління роботом

Сучасні тенденції демонструють активне впровадження штучного інтелекту в системи управління роботами. Машинне навчання та нейронні мережі дозволяють створювати адаптивні моделі, які змінюють свою поведінку в залежності від накопиченого досвіду, а методи підкріплювального навчання допомагають формувати ефективні стратегії навіть у ситуаціях, де неможливо

передбачити всі можливі сценарії наперед. За даними джерела [3], такі підходи значно розширюють можливості автономних систем, роблячи їх гнучкими та здатними до самонавчання (рис. 1.11).

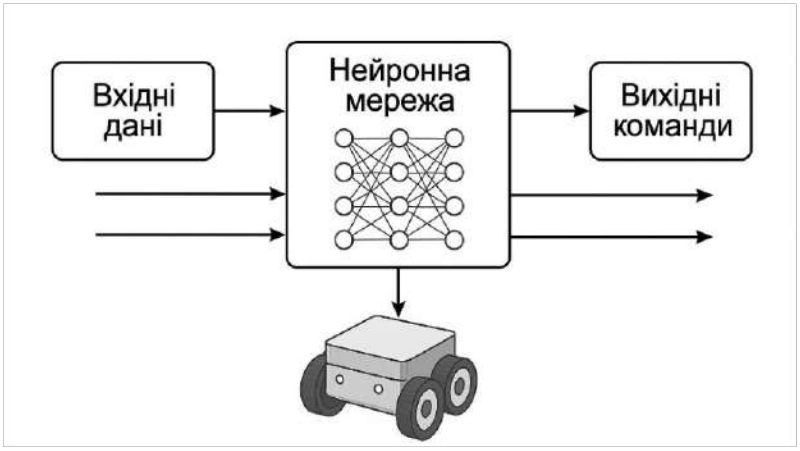

Рисунок 1.11 – Модель роботи нейронної мережі в управлінні мобільним роботом

Таким чином, вибір архітектури та методу управління залежить від задачі, що стоїть перед роботом, умов його експлуатації та технічних характеристик апаратних компонентів. Поєднання класичних і сучасних підходів забезпечує створення гнучких, адаптивних і ефективних систем, які здатні працювати в складних і мінливих середовищах.

1.3 Огляд методів виявлення об’єктів

Сучасні автономні робототехнічні системи потребують надійних методів виявлення об’єктів для забезпечення орієнтації в середовищі, ухвалення рішень і точного виконання завдань. Методи виявлення об’єктів поділяються на сенсорні й візуальні, кожен із яких має свої переваги, недоліки й сфери застосування.

Датчики руху є важливою складовою автономних робототехнічних систем, оскільки саме вони дозволяють роботу сприймати навколишнє середовище, виявляти перешкоди й формувати адекватні реакції на зміну умов. Серед

найпоширеніших у практиці автономної навігації є ультразвукові сенсори, зокрема модуль HC-SR04, що завдяки своїй простоті, низькій вартості й ефективності широко використовується в роботах, дронах, транспортних платформах.

Принцип роботи ультразвукового сенсора (рис. 1.12) ґрунтується на явищі ехолокації. Сенсор випромінює короткий ультразвуковий імпульс, який відбивається від об’єкта й повертається до приймача. Вимірюючи час затримки між випроміненням і прийомом сигналу, мікроконтролер розраховує відстань до перешкоди за допомогою відомої швидкості поширення звуку у повітрі (приблизно 343 м/с). Цей метод дозволяє визначати відстані в діапазоні приблизно від 2 см до 4 м із точністю до кількох міліметрів.

Рисунок 1.12 – Принципова схема роботи ультразвукового сенсора HC-SR04

Переваги ультразвукових сенсорів полягають у їхній нечутливості до кольору чи текстури об’єктів, здатності працювати у темряві, простоті підключення до мікроконтролерів (наприклад, Arduino) та наявності великої кількості бібліотек для програмування. Завдяки цим властивостям вони є основним інструментом для побудови простих систем уникнення перешкод,

автоматичного паркування, слідування за стіною або побудови двовимірних карт навколишнього простору.

Водночас обмеження ультразвукових сенсорів проявляються у вузькому куті огляду (приблизно 15°-30°), зниженій ефективності на нерівних або м'яких поверхнях (які можуть поглинати звук), а також у можливості виникнення завад від одночасної роботи кількох сенсорів (ефект перехресного сигналу). Щоб подолати ці проблеми, часто використовують багато сенсорні конфігурації або комбіновані системи, які поєднують ультразвук із інфрачервоними чи візуальними модулями [12].

Візуальні методи передбачають використання камер (рис. 1.13) для збору даних про оточення. Наприклад, модулі ESP32-CAM або спеціалізовані камери, як Pixy2, здатні визначати колірні мітки, контури й рух об'єктів. Такі методи забезпечують велику кількість інформації, даючи змогу проводити розпізнавання складних об'єктів, аналізувати рух, відстежувати траєкторії. Проте вони сильно залежать від якості освітлення та вимагають значних обчислювальних ресурсів [13].

Рисунок 1.13 – Система візуального розпізнавання об'єктів за допомогою модуля Pixy2

Інфрачервоні (IR) сенсори є важливим класом детекторів, які використовуються для виявлення теплового випромінювання об'єктів. Вони

працюють у спектральному діапазоні, невидимому людському оку (приблизно 700 нм – 1 мм), що дозволяє фіксувати теплові сигнали навіть у повній темряві. Така властивість робить IR-сенсори незамінними в багатьох сферах: системах безпеки, охорони периметра, автоматизованому патрулюванні, а також у медичних чи дослідницьких роботах, призначених для моніторингу рухомих теплових об'єктів (людей, тварин). Крім того, їх активно застосовують у побутовій електроніці для реалізації дистанційного керування різними пристроями.

Принцип роботи IR-сенсора ґрунтується на реєстрації електромагнітного випромінювання, яке кожен об'єкт випромінює пропорційно своїй температурі. У конструкції сенсора зазвичай присутній пірометричний детектор або термопара, яка змінює електричні параметри під дією інфрачервоного випромінювання. Це дозволяє перетворити тепловий сигнал на цифровий або аналоговий сигнал, який далі обробляється мікроконтролером.

Сильні сторони IR-сенсорів включають їхню здатність працювати без видимого освітлення, високу чутливість до біологічних об'єктів, можливість фіксувати навіть незначні зміни температури, а також відносну простоту інтеграції з іншими сенсорними системами (наприклад, з ультразвуковими або оптичними модулями). У системах автономних роботів IR-сенсори використовуються для побудови простих систем запобігання зіткненням, активації тригерів у присутності людини, або для навігації у приміщеннях зі змінними світловими умовами. Завдяки своїй компактності та енергоефективності, вони є оптимальним вибором для багатьох мобільних і портативних робототехнічних платформ.

Сучасні IR-системи можуть включати інтелектуальні алгоритми обробки даних, що підвищують точність детекції [12].

На рисунку 1.14 показано прототип високошвидкісного інфрачервоного детектора, встановленого на інструменті Pionier в Обсерваторії Паранал, ЄПО.

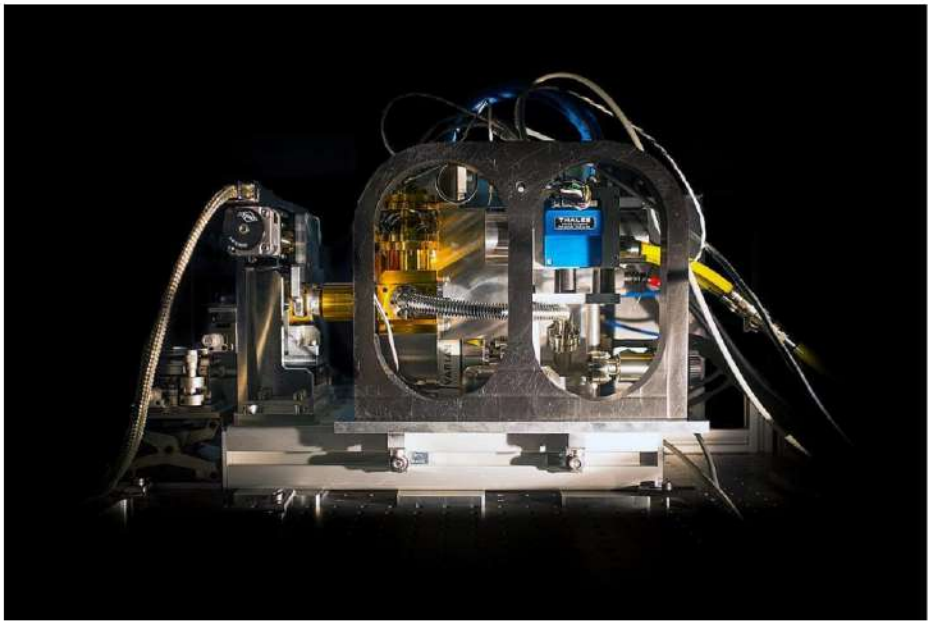

Рисунок 1.14 – Прототип високошвидкісного інфрачервоного детектора, встановленого на інструменті Pionier в Обсерваторії Паранал, ЄПО [12]

Особливої уваги заслуговує використання бібліотеки OpenCV – одного з найпотужніших програмних інструментів для комп'ютерного зору. OpenCV дозволяє реалізовувати алгоритми розпізнавання облич, аналізу руху, сегментації зображення, виявлення країв, відстеження ключових точок та інші складні задачі (рис. 1.15).

Рисунок 1.15 – Алгоритм виявлення об'єктів за допомогою OpenCV на прикладі мобільного робота

Перевага такого підходу полягає у високій гнучкості та можливості масштабування – алгоритми можуть працювати як на вбудованих пристроях, так і на потужних серверах, а також навчатися й удосконалюватися за допомогою нейронних мереж [14].

Аналіз наукових публікацій демонструє, що ефективність виявлення об’єктів значно підвищується при використанні багато сенсорних систем, які комбінують дані з ультразвукових, візуальних та інфрачервоних сенсорів, забезпечуючи стійкість до шумів, адаптивність і високу точність Такі системи відкривають шлях до створення автономних роботів, здатних працювати в динамічних та непередбачуваних середовищах.

РОЗДІЛ 2
АПАРАТНА РЕАЛІЗАЦІЯ СИСТЕМИ

2.1 Плата Arduino Uno R3

Процес проектування автономного мобільного робота починається з ретельного вибору апаратних компонентів.

Arduino UNO (рис. 2.1) – це плата мікроконтролера на базі ATmega328P . Вона має 14 цифрових вхідних/вихідних контактів (6 з яких можна використовувати як ШІМ-виходи), 6 аналогових входів, керамічний резонатор 16 МГц, USB-роз'єм, роз'єм живлення, роз'єм ICSP та кнопку скидання. Вона містить усе необхідне для підтримки мікроконтролера; просто підключіть його до комп'ютера за допомогою USB-кабелю або підживіть за допомогою адаптера змінного струму в постійний струм чи батареї, щоб розпочати роботу. Ви можете повозитися зі своїм UNO, не надто турбуючись про те, що зробите щось не так, у гіршому випадку ви можете замінити чіп за кілька доларів і почати все спочатку [15].

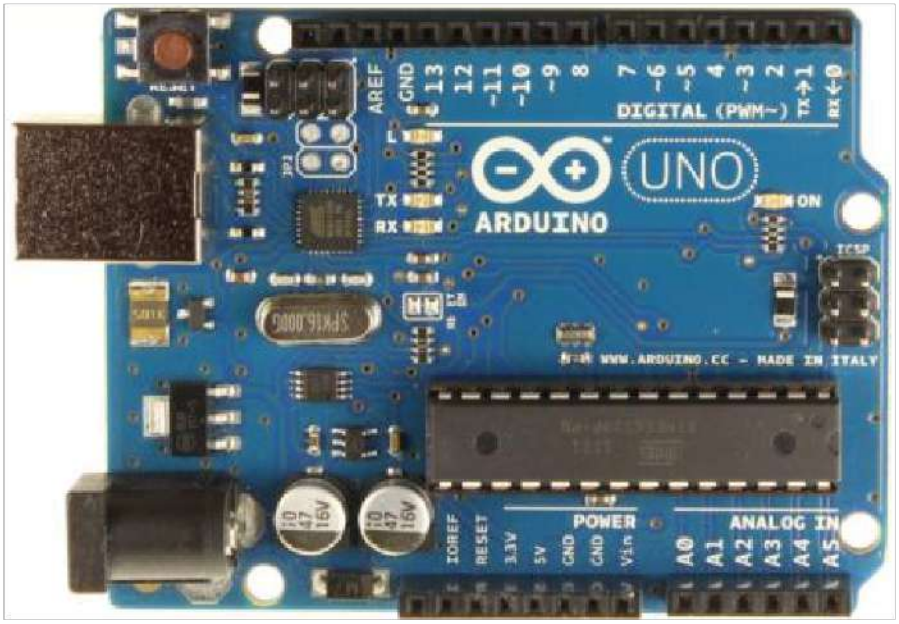

Рисунок 2.1 – Плата Arduino Uno [15]

На рисунку 2.2 зображено найменування входів/виходів Arduino Uno

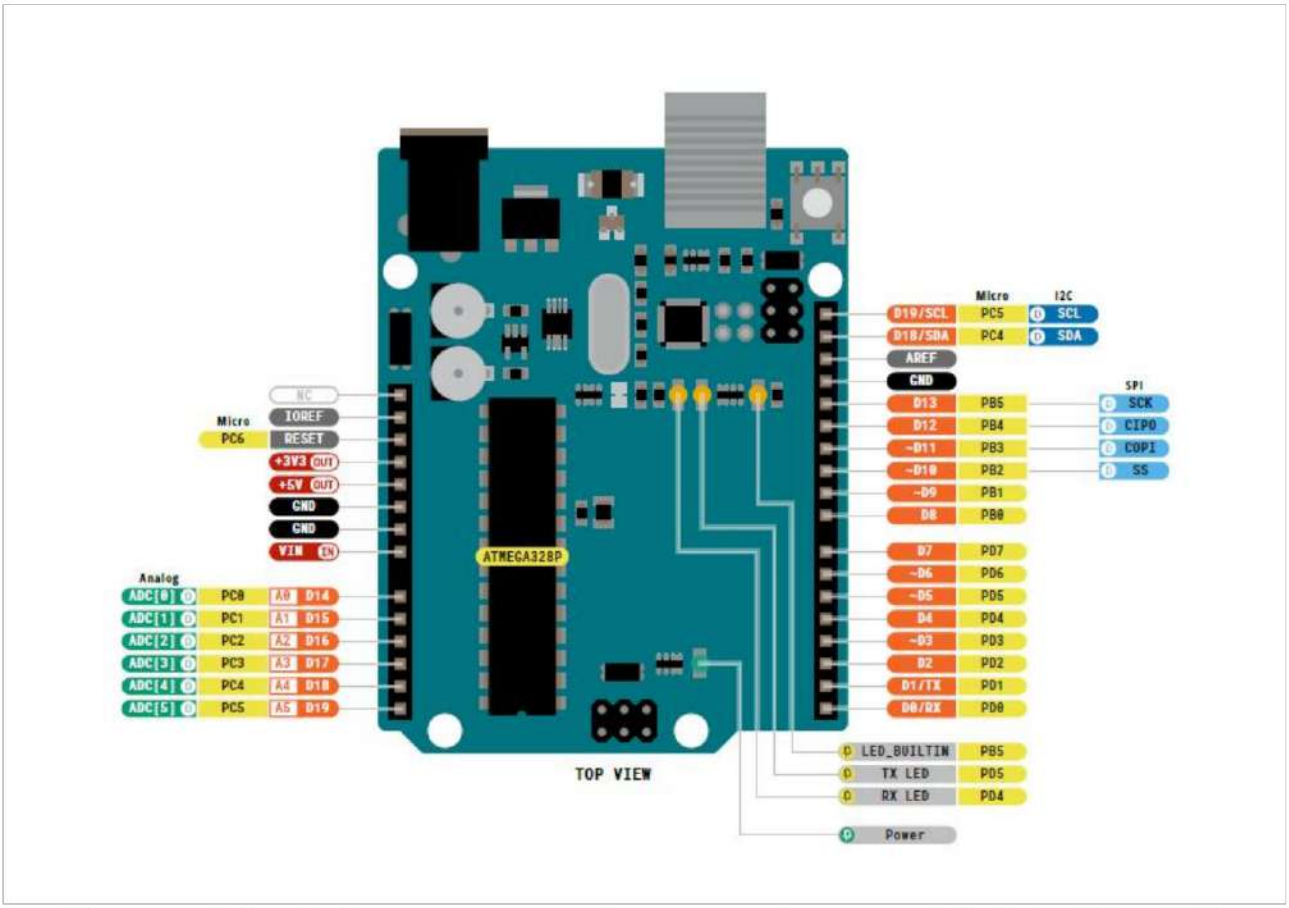

Рисунок 2.2 – Найменування входів та виходів на платі Arduino Uno [15]

Основними характеристиками пристрою Arduino Uno:

– пам’ять Flash - 32 Kb;

– частота - 16 МГц;

– мікроконтролер – ATmega328;

– вхідне живлення контролера 5 V - 12 V;

– цифрові входи та виходи – 14 шт., з них 6 шт. – ШІМ;

– інтерфейси – I2C/TWI, SPI, PWM;

– розмір: 68 × 53 × 15 мм.

2.2 Ультразвуковий датчик відстані HC SR04

Ультразвуковий датчик відстані HC-SR04 (рис. 2.3) – призначений для вимірювання відстаней від 2 до 400 см, причому межа точності може досягати до 3 мм. Модуль включає ультразвуковий передавач, приймач і вузол контролю.

На покази датчика практично не впливають сонячне випромінювання і електромагнітні шуми. На передній частині HC-SR04 розташовано два ультрозвукових датчика, перший з написом T (Transmiter) – це передавач ультрозвукових хвиль (TCT40-16T), а другий з написом R (Receive) – це приймач відбитих ультрозвукових хвиль (TCT40-16R), по центру розташований вивідний кварцовий генератор на 27 МГц.

Рисунок 2.3 – Загальний вигляд датчика HC SR04 [16]

Характеристики датчика HC SR04:

– діапазон вимірювання відстані 0,03 м - 4 м;

– частота ультразвуку 40 kHz;

– кут огляду 30;

– інтерфейс 2 логічні TTL лінії;

– вихідна інформація імпульс 0,15 - 25 mS;

– напруга живлення «Vcc» 5 V;

– струм споживання в активному режимі 15 мА;

– розмір модуля 45 x 20 x 15 mm.

Принцип роботи датчика наступний:

1) на вихід trig (тригер) посилаємо високий рівень протягом як мінімум 10 мкс;

2) модуль починає посилати ультразвукові імпульси з частотою 40 кГц і приймає їх назад, якщо в зоні видимості є будь-які перешкоди;

3) якщо сигнал повертається, модуль встановлює низький рівень на виході echo на 150 мс. За часом, який минув з п.1 до низького рівня на виході echo можна розрахувати відстань до перешкоди за формулою: відстань = (time * sound velocity)/2, де time – виміряний час імпульсу, sound velocity – швидкість звуку (340 м/с).

Точність вимірювання відстані датчиком HC SR04 залежить від декількох факторів:

– температури і вологості повітря;

– відстані до об’єкта;

– розташування щодо датчика (згідно діаграми випромінювання);

– якості виконання елементів модуля датчика.

В основу принципу дії будь-якого ультразвукового датчика закладено явище відображення акустичних хвиль, що поширюються в повітрі. Як відомо з курсу фізики, швидкість поширення звуку в повітрі залежить від властивостей цього самого повітря (в першу чергу від температури). Датчик ж, випускаючи хвилі і заміряючи час до їх повернення, не здогадується, в якому саме середовищі вони будуть поширюватися і бере для розрахунків деяку середню величину. В реальних умовах через фактор температури повітря HC-SR04 може помилятися від 1 до 3-5 см. Фактор відстані до об’єкта важливий, тому що росте ймовірність відбиття від сусідніх предметів, до того ж і сам сигнал загасає з відстанню [16].

Також для підвищення точності треба правильно направити датчик: зробити так, щоб предмет був в рамках конуса діаграми спрямованості. Простіше кажучи, «очі» HC-SR04 повинні дивитися прямо на предмет. Діаграма спрямованості HC-SR04 представлена на рисунку 2.4.

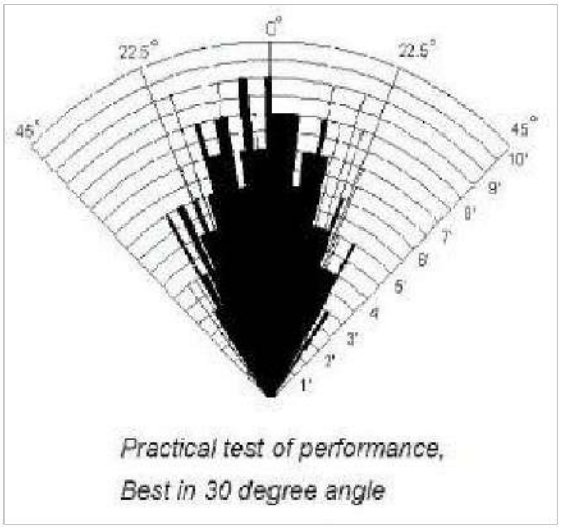

Рисунок 2.4 – Діаграма спрямованості HC SR04 [16]

Для зменшення помилок і похибки вимірювань зазвичай виконуються наступні дії:

– усереднюються значення (кілька разів заміряємо, прибираємо сплески, потім знаходимо середнє);

– за допомогою датчиків (наприклад, DHT11 або DHT22) визначається температура і вносяться поправочні коефіцієнти;

– датчик встановлюється на серводвигун, за допомогою якого ми «повертаємо голову», переміщаючи діаграму спрямованості вліво або вправо.

На рисунку 2.5 зображено принципову схему датчика.

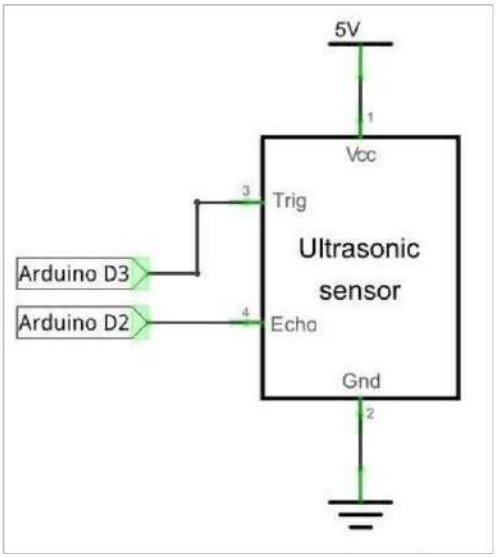

Рисунок 2.5 – Принципова схема датчика [16]

2.3 Arduino Motor Shield L298P

L298P Motor Shield (рис. 2.6) – це плата драйвера двигунів постійного струму, що використовує мікросхему потужного драйвера електродвигунів L298P, яка може безпосередньо управляти двома двигунами постійного струму; струм через навантаження – до 2 ампер. Вихідні інтерфейси управління двигунами використовують 8 високошвидкісних діодів Шоттки в якості захисту. Дана плата може бути встановлена безпосередньо на плату Arduino.

Цей Шилд особливо стане в нагоді тим, хто збирає роботів і робоплатформи з використанням колекторних двигунів. На платі, крім самого драйвера двигунів L298, зручно розміщені роз'єми для підключення Bluetooth модуля, сервоприводу і датчиків. Зручне підключення, гарна якість монтажу вигідно відрізняють даний Шилд від аналогічних.Серводвигун, на якому закріплена камера, повертається на заданий кут відповідно до того, який саме датчик зафіксував рух. Це дозволяє спрямувати камеру точно в напрямку потенційної загрози для отримання чіткого зображення об'єкта [17].

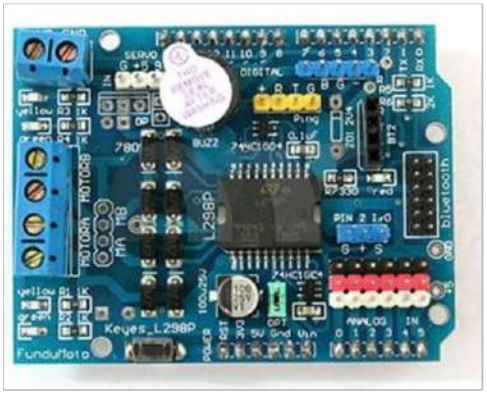

Рисунок 2.6 – L298P Motor Shield [17]

Характеристики L298P Motor Shield:

– вхідна напруга логіки VD: 5 В;

– вхідна напруга для моторів: VIN 6,5 - 12 В, PWR IN 4,8 - 24 В;

– струм споживання логіки ISS: ≤ 36 мА;

– струм споживання моторів IO: ≤ 2 А;

– максимальна потужність: 25 Вт (T = 75° C);

– електричні рівні вхідних сигналів: логічна одиниця (2,3 В ≤ Vin ≤ 5 В) та логічний нуль (0,3 В ≤ Vin ≤ 1,5 В);

– робоча температура: -25 – 130° C.

Особливості:

1) основою модуля є мікросхема драйвера двигунів L298P, тому ви можете використовувати цифровий інтерфейс вводу/виводу D10, D11, D12, D13 без додаткових складних схем підключення;

2) на платі встановлений зумер (D4), який можна використовувати, наприклад, для формування сигналу тривоги;

3) на платі передбачений спеціальний роз'єм для модулів Bluetooth, що дозволяє встановлювати модулі без додаткових провідників безпосередньо на плату;

4) на платі виведений інтерфейс шести, не задіяних, цифрових портів D2, D3, D5, D6, D7, D9;

5) на платі виведений інтерфейс шести, не задіяних, аналогових портів A0, A1, A2, A3, A4, A5;

6) на платі встановлені світлодіоди для індикації прямого або зворотнього напрямку обертання.

На рисунку 2.7 зображено інтерфейси плати.

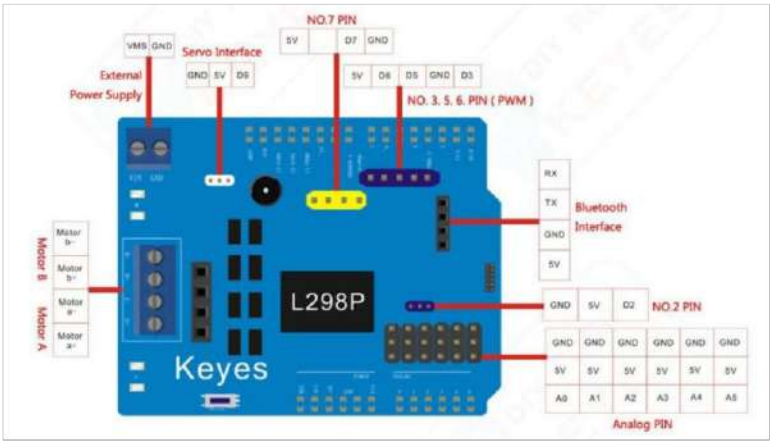

Рисунок 2.7 – Інтерфейси плати [17]

На рисунку 2.8 показано принципову схему плати L298P.

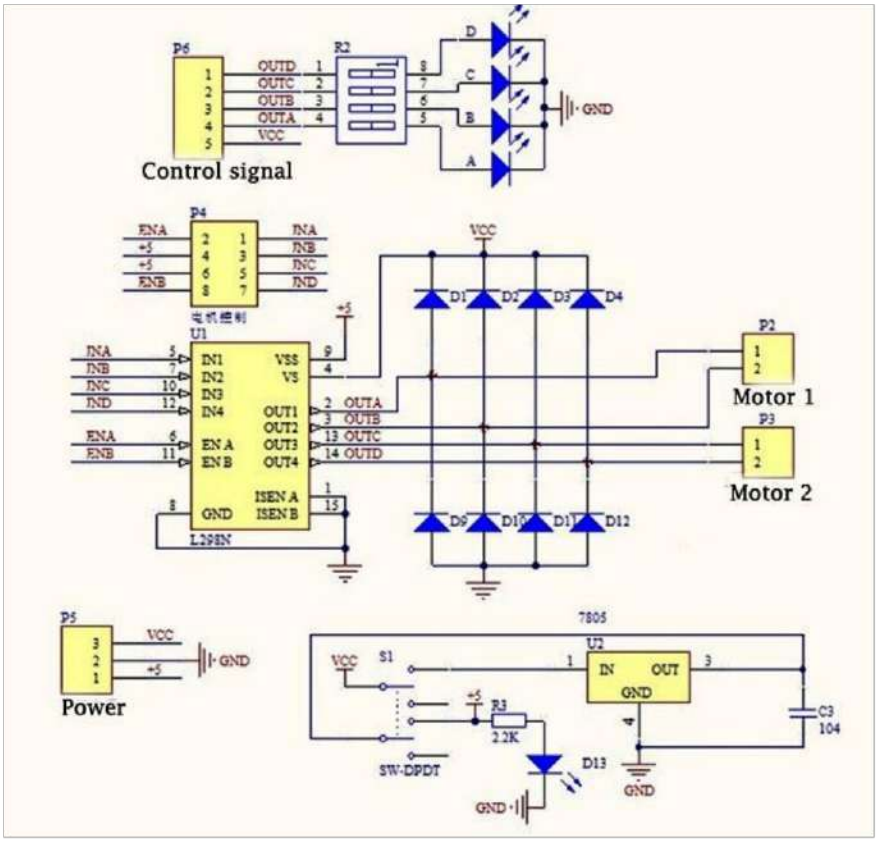

Рисунок 2.8 – Принципова схема плати L298P [17]

2.4 Сервопривід MG90S Micro Servo

Сервопривід SG90 є одним із найпопулярніших мікросервоприводів для початкових робототехнічних проєктів, освітніх задач і хобі-конструювання. Він поєднує компактні розміри, низьке енергоспоживання та прийнятну точність позиціонування, що робить його зручним для інтеграції в легкі механічні системи. Завдяки простоті підключення SG90 широко використовується разом із мікроконтролерами Arduino, Raspberry Pi, ESP32 тощо [18].

Основні технічні характеристики SG90:

– напруга живлення: 4,8 - 6,0 В;

– кут повороту: близько 180° (залежить від керуючого сигналу);

– швидкість: ~0,12 с/60° при 4,8 В;

– вага: близько 9 г;

– розміри: 22,2 × 11,8 × 31 мм.

SG90 (рис. 2.8) управляється за допомогою ШІМ-сигналу (широтно-імпульсної модуляції), де тривалість імпульсу визначає кут повороту вихідного валу. Сервопривід підключається до цифрового виходу мікроконтролера та живиться від зовнішнього джерела (або безпосередньо від плати, якщо енергоспоживання дозволяє). У типовому застосуванні він використовується для повороту сенсорів, камер, легких механічних захоплювачів, сервомеханізмів, що регулюють положення крила або керма в моделях літаків тощо.

Рисунок 2.8 – Сервопривід SG90 [18]

Основною перевагою SG90 є його простота інтеграції в навчальні проєкти, оскільки він повністю сумісний з Arduino, а в інтернеті доступна велика кількість бібліотек, прикладів та відеоуроків. Проте через пластиковий редуктор сервопривід має певні обмеження щодо довговічності та стійкості до механічних навантажень, особливо при багатогодинному використанні в циклічних режимах або при роботі з важкими механізмами. В порівнянні з моделями, що мають металеві шестерні, такими як MG90S, SG90 поступається за показниками максимальної сили, проте залишається чудовим вибором для легких і середньонавантажених завдань, де критичними є маса, енергоефективність та ціна.

Таким чином, SG90 є оптимальним рішенням для початкових робототехнічних розробок, де потрібне недороге, легке та просте в управлінні рішення для виконання базових маневрів, а його популярність і доступність на ринку роблять його невід'ємним компонентом багатьох навчальних лабораторних наборів і прототипів.

РОЗДІЛ 3
РОЗРОБКА ТА ПРОГРАМУВАННЯ АВТОНОМНОГО РОБОТА

3.1 Створення прототипу автономного робота

Процес створення прототипу автономного робота включав кілька послідовних етапів, кожен з яких мав важливе значення для забезпечення функціональності й надійності системи.

На першому етапі (рис. 3.1) було проведено підготовку механічної частини конструкції. Було зібрано основу гусеничної платформи, встановлено мотор-редуктори, закріплено колісні модулі та гусеничні траки, перевірено їхню натяжку та свободу обертання. Особливу увагу приділено забезпеченню правильної геометрії шасі, щоб уникнути перекосів та перевантажень на двигуни під час руху.

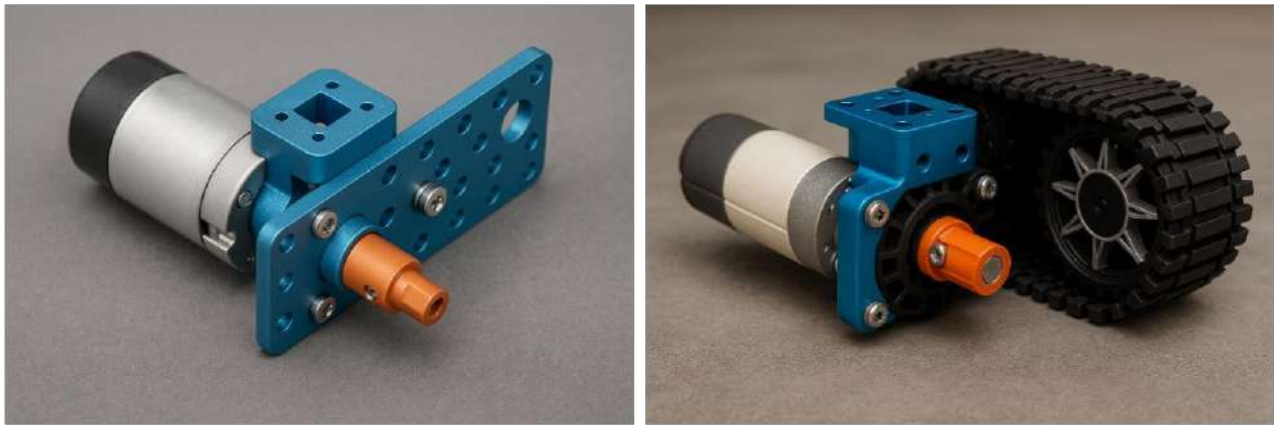

Рисунок 3.1 – Встановлення нижнього мотора та ведучого колеса

Другий етап (рис. 3.2) передбачав монтаж електронних компонентів. На верхній платформі конструкції закріплювалися мікроконтролер Arduino Uno, моторний драйвер L298P Shield, а також блок живлення й акумуляторний модуль. Кабельні з’єднання акуратно розводилися з урахуванням потреб у живленні та сигналізації, щоб уникнути електромагнітних завад і забезпечити стабільність роботи системи.

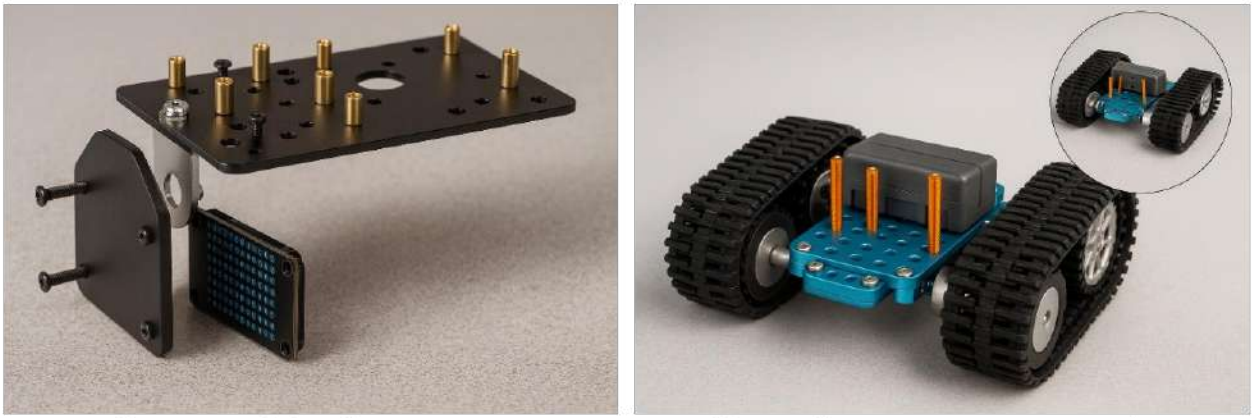

Рисунок 3.2 – Встановлення тримача батареї та плати та датчиків

На третьому етапі проводився монтаж сенсорного блоку. Ультразвуковий датчик HC-SR04 було встановлено на сервоприводі MG90S для забезпечення можливості активного сканування простору перед роботом (рис. 3.3). Таке рішення дозволило розширити кут огляду та підвищити точність навігації, оскільки система отримала здатність оцінювати обстановку не лише по фронтальній осі, а й у бокових секторах.

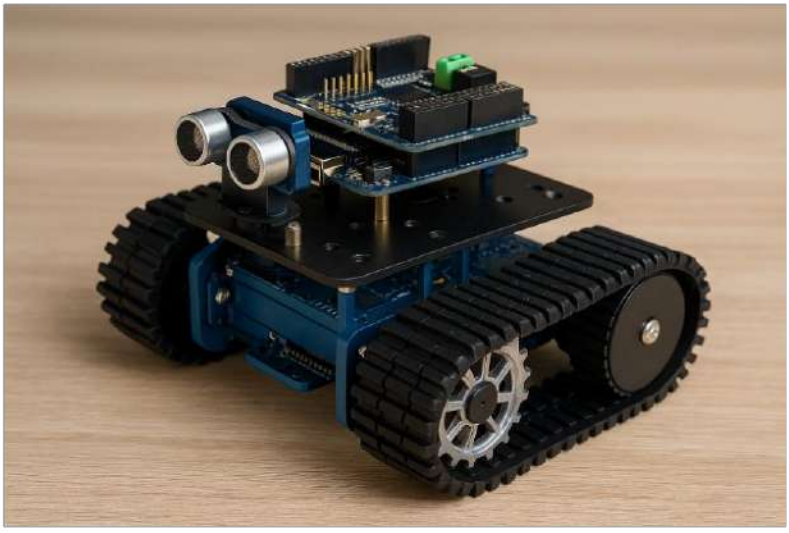

Рисунок 3.3 – Встановлення датчиків та плат

Четвертий етап включав тестування окремих компонентів. Було перевірено обертання двигунів, калібрування кутів сервоприводів, а також зчитування даних з ультразвукового сенсора. Для цього використовувалися

прості тестові скетчі Arduino, які дозволяли поетапно перевірити кожен вузол системи.

На п’ятому етапі здійснювалася інтеграція всієї системи: підготовка програмного забезпечення, що об’єднує роботу всіх компонентів, перевірка взаємодії між мікроконтролером, сенсорами, приводами та блоками живлення. Було налаштовано алгоритми ухвалення рішень для навігації, уникнення перешкод і слідування за заданим маршрутом.

Завершальний етап передбачав комплексне тестування прототипу в реальних умовах. Було перевірено здатність робота пересуватися по різних поверхнях, об’їжджати перешкоди, стабільність зв’язку між модулями та ефективність алгоритмів управління. Результати тестування показали працездатність системи та заклали основу для подальших досліджень і вдосконалення.

3.2 Схема підключення модулів

На рисунку 3.4 представлено макетну схему, яка ілюструє під’єднання джерела живлення та електромоторів.

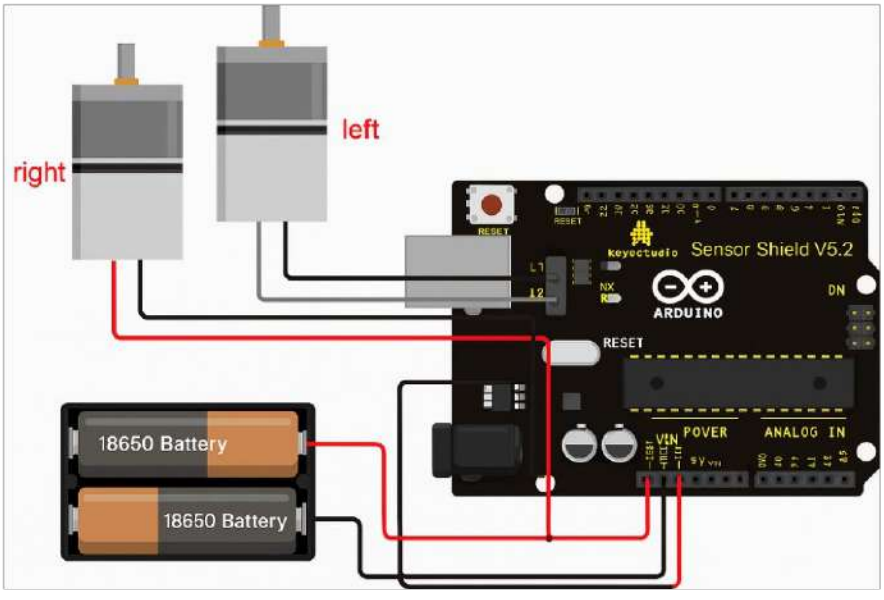

Рисунок 3.4 – Схема підключення живлення та електромоторів

На рисунку 3.5 представлено макетну схему, яка ілюструє підключення ультразвукового датчика HC-SR04.

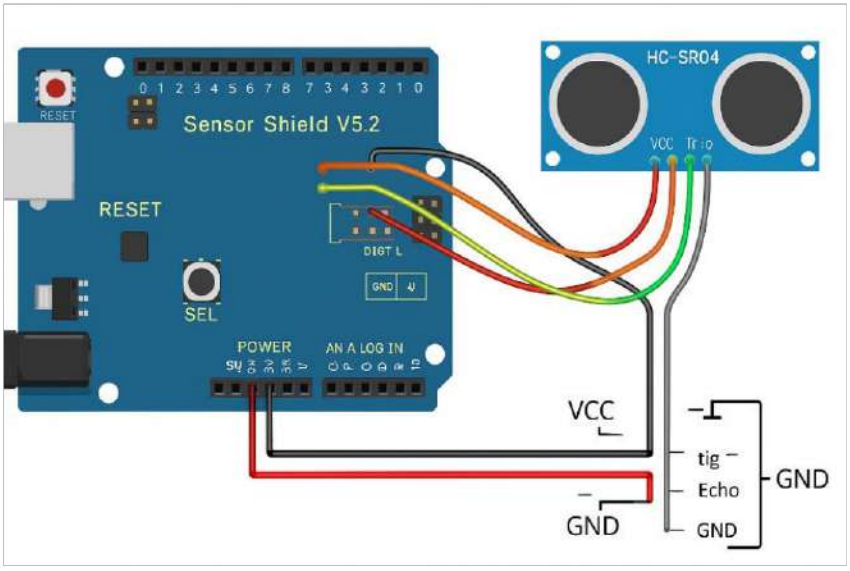

Рисунок 3.5 – Схема підключення ультразвукового датчика HC-SR04

Логіка роботи ультразвукового робота, який слідує за об'єктом показана на блок-схемі (рис. 3.6).

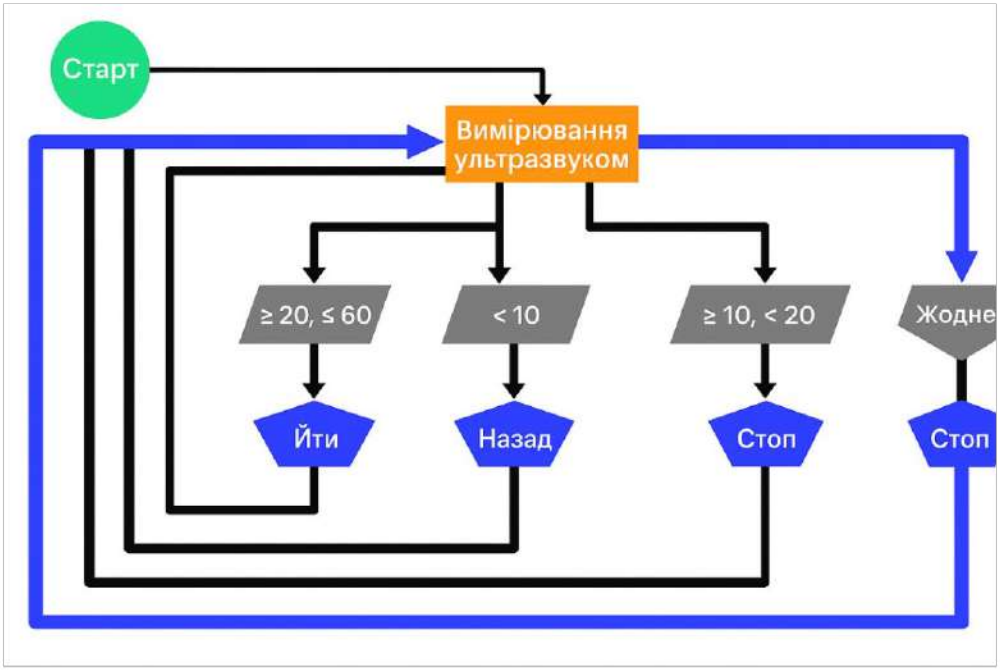

Рисунок 3.6 – Блок-схема роботи ультразвукового робота, який слідує за об'єктом

На рисунку 3.7 представлено загальну схему підключення модулів автономного робота.

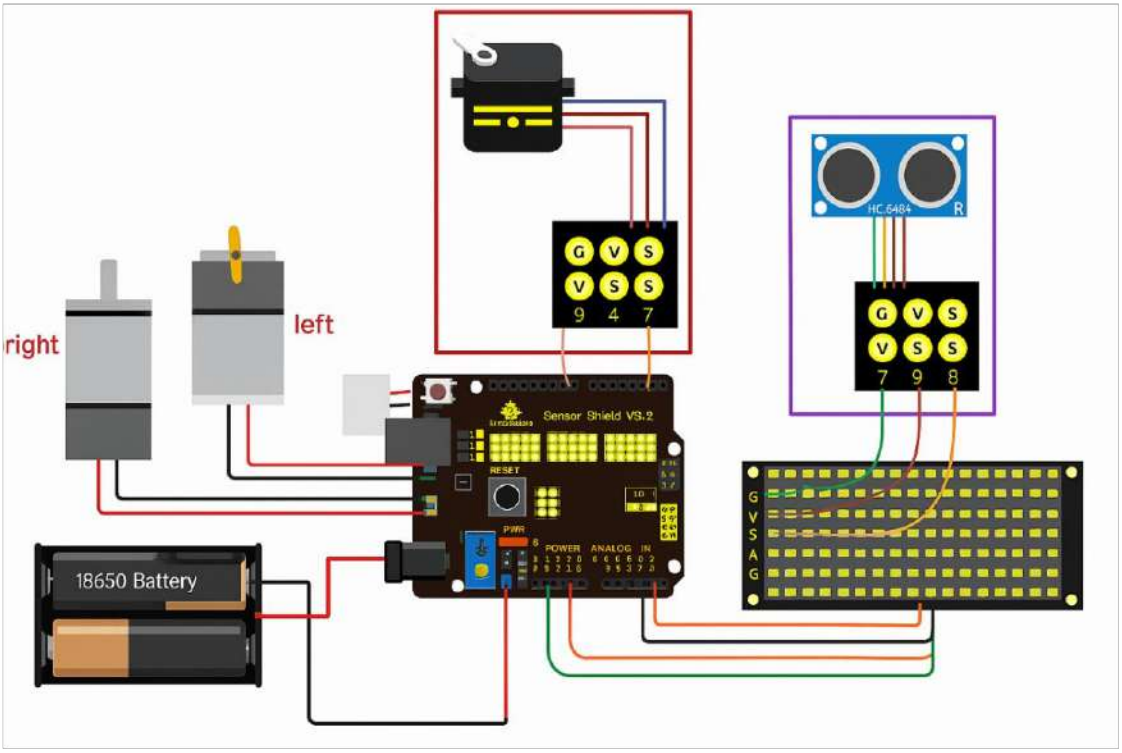

Рисунок 3.7 – Загальна схема підключення модулів автономного робота

3.3 Програмування Arduino

На даному етапі розробки автономного робота відбувається програмування мікроконтролера Arduino Uno, яке забезпечує керування усіма підключеними модулями та виконавчими механізмами. Для написання та завантаження програмного коду застосовувалося середовище розробки Arduino IDE (рис. 3.8), що підтримує мову програмування на основі C/C++. Саме це середовище надає простий спосіб написання скетчів (програм), завдяки вбудованим бібліотекам для роботи з сенсорами, моторами, сервоприводами, світлодіодами та іншими компонентами.

Середовище розробки Arduino IDE (рис. 3.8) є одним із ключових інструментів для створення, компіляції та завантаження програмного забезпечення на мікроконтролери платформи Arduino. Arduino IDE (Integrated

Development Environment) – це вільно поширюваний кросплатформний інструмент із графічним інтерфейсом, який підтримує операційні системи Windows, macOS та Linux. Завдяки своїй простоті та відкритості Arduino IDE здобуло широку популярність як серед початківців, так і серед професіоналів у сфері робототехніки та електроніки.

Головна панель середовища складається з редактора коду, панелі інструментів для компіляції та завантаження скетчів, а також монітора послідовного порту, що дозволяє проводити відлагодження програм. Мова програмування в Arduino IDE базується на спрощеному синтаксисі C/C++ з використанням спеціальних бібліотек, які значно спрощують доступ до апаратних функцій мікроконтролера, таких як управління пін-входами/виходами, робота з PWM, серійну комунікацію, підключення датчиків та дисплеїв.

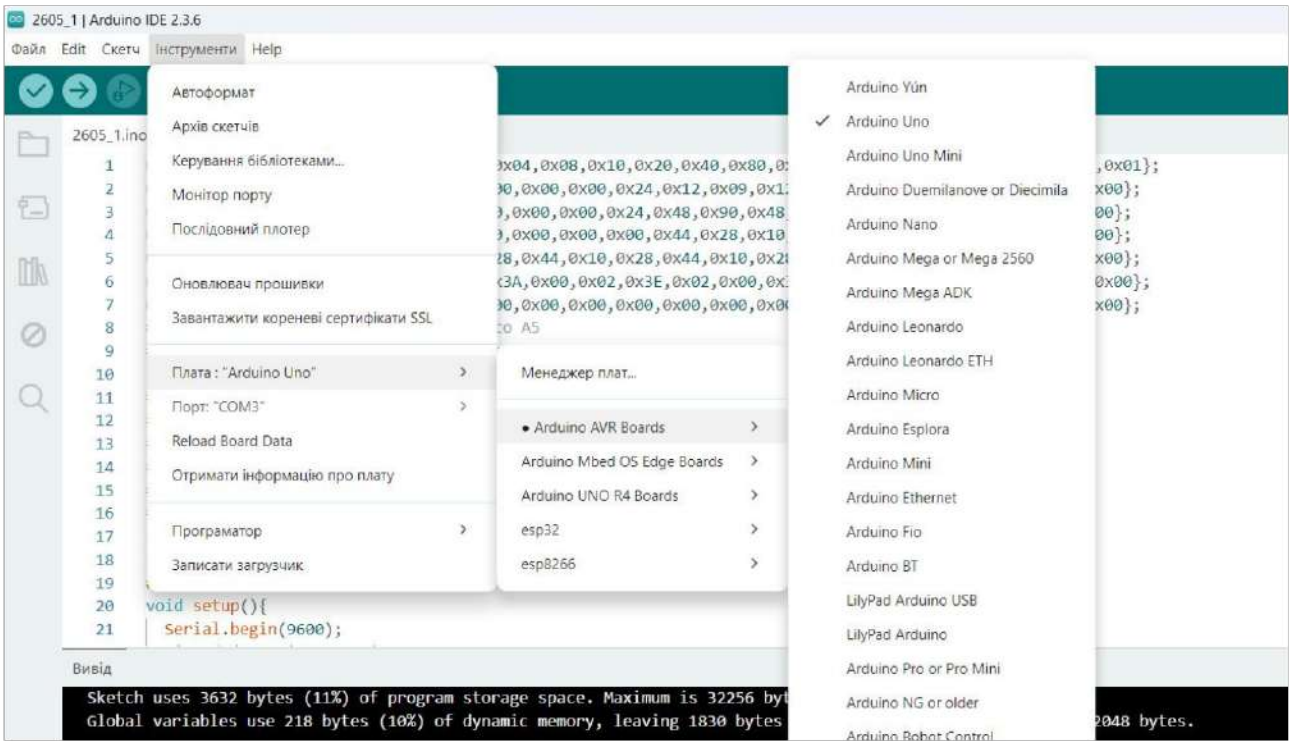

Рисунок 3.8 – Середовище розробки Arduino IDE

Розроблений скетч містить алгоритми обробки сигналів ультразвукового датчика HC-SR04, сервопривода SG90, двигунів постійного струму та матриці світлодіодів. Програмний код виконує вимірювання відстані до об’єктів,

визначає режим роботи робота (рух вперед, назад, зупинка), керує обертами моторів залежно від вхідних даних, а також виводить інформацію на світлодіодну матрицю.

Приклад програмного фрагмента, що забезпечує обробку даних з ультразвукового сенсора HC-SR04 та керування мотором, наведений на лістингу 3.1.

Лістинг 3.1 – Обробку даних з ультразвукового сенсора HC-SR04

```
#define ML_Ctrl 13  //define the direction control pin of left motor
#define ML_PWM 11   //define PWM control pin of left motor
#define MR_Ctrl 12  //define the direction control pin of right motor
#define MR_PWM 3   //define PWM control pin of right motor
#define Trig 5  //ultrasonic trig Pin
#define Echo 4  //ultrasonic echo Pin
int distance;
int pulsewidth;
#define servoPin 9  //servo Pin
void setup(){
  Serial.begin(9600);
  pinMode(SCL_Pin,OUTPUT);
  pinMode(SDA_Pin,OUTPUT);
  matrix_display(clear); //Clear the display
  matrix_display(start01);  //display start pattern
  pinMode(servoPin, OUTPUT);
  procedure(180); //set servo to 90°
  pinMode(Trig, OUTPUT);
  pinMode(Echo, INPUT);
  pinMode(ML_Ctrl, OUTPUT);
  pinMode(ML_PWM, OUTPUT);
  pinMode(MR_Ctrl, OUTPUT);
  pinMode(MR_PWM, OUTPUT);
}
void loop(){
  distance = checkdistance();  //assign the distance detected by ultrasonic sensor
to distance
  if (distance >= 20 && distance <= 60) //range to go front
  {
    Car_front();
  }
  else if (distance > 10 && distance < 20)  //range to stop
{
    Car_Stop();
  }
  else if (distance <= 10)  //range to go back
```

Продовження лістингу 3.1

```
}
  else if (distance <= 10)  //range to go back

  {
    Car_Stop();
  }
  else if (distance <= 10)  //range to go back
  {
    Car_back();
  }
  else  //other situations, stop
  {
    Car_Stop();
  }
}
```

Кінець лістингу 3.1

Програмування здійснюється шляхом оголошення глобальних змінних, налаштування пінів вводу/виводу та написання циклу loop(), що забезпечує безперервну роботу алгоритмів. Окремими функціями реалізоване керування серво, обробка результатів віддалеміра, а також відображення даних на дисплеї. Крім того, застосовуються бібліотеки Servo.h для роботи з SG90, а також стандартні бібліотеки Arduino для обробки сигналів PWM.

У скетчі застосовуються захисні механізми, наприклад, таймаути для уникнення зависань у разі відсутності сигналу з ультразвукового датчика, а також обмеження швидкості зміни кута сервопривода для запобігання механічним перевантаженням. Перед завантаженням програми на мікроконтролер проводиться компіляція та перевірка на помилки у середовищі Arduino IDE. Після успішного завантаження програма автоматично запускається на платі Arduino Uno та починає взаємодію з усіма підключеними компонентами системи.

Завдяки модульності програмного коду, можливе розширення функціональності системи шляхом додавання нових функцій, наприклад, Bluetooth-керування, GPS-навігації або обробки візуальних даних з камери. Такий підхід дозволяє швидко адаптувати систему під нові задачі та

експериментальні сценарії, що є особливо важливим для навчальних і дослідницьких проектів.

3.4 Тестування автономного робота

Після завершення етапів апаратної збірки та програмування проведено тестування автономного робота (рис. 3.9) з метою перевірки його працездатності, коректності виконання алгоритмів та відповідності заданим технічним вимогам.

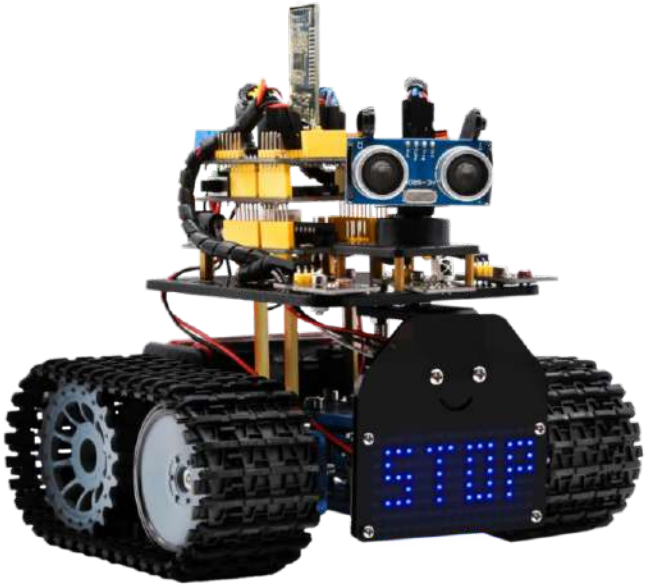

Рисунок 3.9 – Автономний робот

Тестування здійснювалося в лабораторних умовах на рівній поверхні з використанням спеціально підготовлених тестових сценаріїв.

Першочергово перевірено роботу модулів живлення та підключення двигунів: при подачі живлення мотор-щит коректно розподіляв енергію на обидва електродвигуни, забезпечуючи керованість робота у всіх режимах – рух вперед, назад, зупинка. Також перевірено роботу сервопривода SG90, який демонстрував стабільні обертальні рухи у межах заданих кутів, що було важливо для виконання функцій сканування або напрямного керування.

Також проведено тестування ультразвукового сенсора HC-SR04 (рис. 3.10). У ході випробувань перевірено точність вимірювання відстані до об’єктів різних розмірів та матеріалів. Виявлено, що на відстанях до 1 метра сенсор демонстрував відхилення не більше ніж ±1 см, що є прийнятним результатом для задачі слідування за об’єктами.

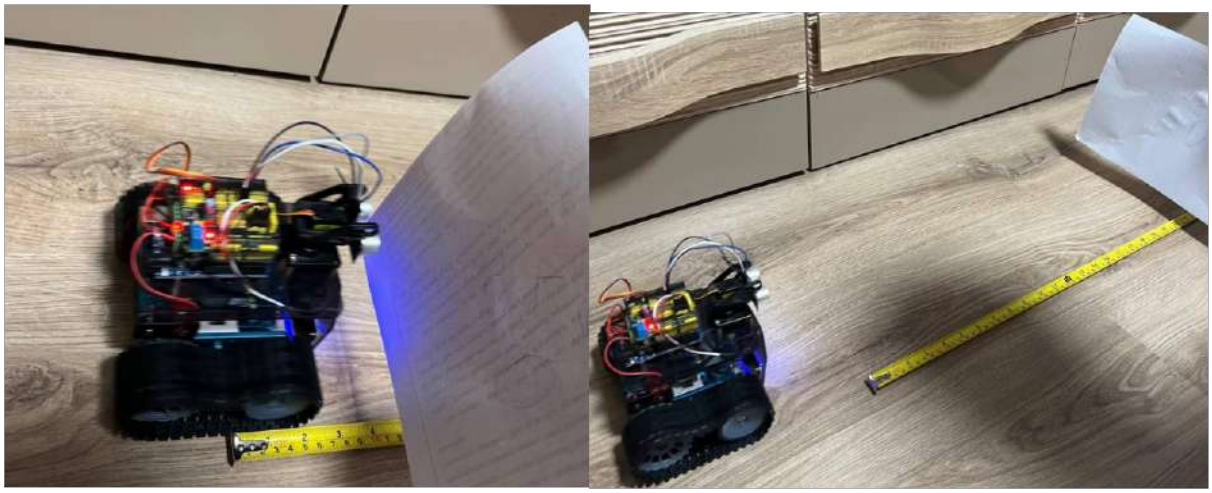

Рисунок 3.10 – Результати тестування

Програмне забезпечення тестувалося шляхом перевірки коректності алгоритмів ухвалення рішень. Зокрема, тестувалося, чи робот переходить у відповідні стани (рух вперед, назад, стоп) при зміні відстані до перешкоди. Для цього використовувалися серійні монітори в Arduino IDE, які дозволяли відслідковувати у режимі реального часу показники вимірювань і поточні дії системи.

На заключному етапі виконано комплексне тестування всього робота, включаючи одночасну роботу моторів, серво, сенсорів і матриці індикації. Усі компоненти працювали узгоджено, робот демонстрував правильне реагування на зміну середовища, забезпечувався безпечний режим роботи (з автоматичною зупинкою при виникненні критичних ситуацій). За результатами тестування можна зробити висновок про функціональну готовність розробленого пристрою до використання у реальних умовах.

ВИСНОВКИ

У кваліфікаційній роботі проведено дослідження та розробку автономного робота на основі платформи Arduino, здатного виявляти рухомий об’єкт та слідувати за ним у режимі реального часу. Під час виконання роботи здійснено аналіз сучасного стану розвитку автономних мобільних систем, досліджено архітектури управління та методи інтеграції сенсорних даних, розглянуто основні типи сенсорів, що застосовуються у робототехніці для виявлення та відстеження об’єктів.

На основі проведеного аналізу було обґрунтовано вибір апаратних компонентів, серед яких мікроконтролер Arduino Uno, ультразвуковий сенсор HC-SR04, мотор-щит L298P та сервопривід MG90S, а також розроблено схему їх інтеграції для створення прототипу автономного робота. Було написано та протестовано програмне забезпечення, що забезпечує обробку сенсорних даних, реалізацію алгоритмів ухвалення рішень, керування приводами та відображення інформації на світлодіодній матриці.

Результати тестування продемонстрували працездатність розробленої системи, коректність реалізованих алгоритмів та відповідність поставленим цілям і завданням. Робота підтвердила, що платформа Arduino є ефективним та доступним інструментом для створення навчальних і дослідницьких прототипів автономних робототехнічних систем.

Розроблений прототип може бути використаний як основа для подальших досліджень і розширення функціональності, зокрема для впровадження комп’ютерного зору, використання більш точних сенсорів або підключення до бездротових мереж. Отримані результати мають як практичну, так і навчальну цінність, сприяючи формуванню компетенцій у сфері робототехніки, електроніки та програмування.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Future trends in autonomous robotics. *Researchgate*. URL: https://www.researchgate.net/publication/379231219_An_overview_of_emerging_trends_in_robotics_and_automation (дата звернення: 07.02.2025).

2. Autonomous robotic systems: A review. *Sciencedirect*. URL: https://www.sciencedirect.com/science/article/abs/pii/S0168169907001688 (дата звернення: 07.02.2025).

3. Classification and applications of autonomous robots. *Researchgate*. URL https://www.researchgate.net/figure/ClassificationofAutonomousrobots_fig1_365590143 (дата звернення: 07.03.2025).

4. Spot robot. *Bostondynamics*. URL: https://bostondynamics.com/products/spot/ (дата звернення: 07.03.2025).

5. DJI Matrice 300 RTK. *Dji*. URL: https://www.dji.com/matrice-300 (дата звернення: 07.04.2025).

6. Автономний підводний апарат Bluefin-21. *Gdmissionsystems*. URL: https://gdmissionsystems.com/products/underwater-vehicles/bluefin-21-autonomous-underwater-vehicle (дата звернення: 07.04.2025).

7. Робот маніпулятор. *Termobud* .URL: https://surl.li/jwymje (дата звернення: 07.04.2025).

8. Інтелектуальний комерційний робот. *Reemanrobot*. URL: https://ua.reemanrobot.com/ humanoid-service-robot/ (дата звернення: 07.04.2025).

9. Наземні бойові роботи: лідери та Україна. *Lb*. URL: https://lb.ua/news/2021/11/17/498795_nazemni_boyovi_roboti_lideri.html (дата звернення: 07.04.2025).

10. Mars Perseverance Rover. *Mars*. URL: https://mars.nasa.gov/mars2020 (дата звернення: 07.04.2025).

11. Levels of autonomy in robotic platforms. *Andplus*. URL: https://www.andplus.com /blog/the-5-levels-of-autonomy (дата звернення: 07.05.2025).

12. Robot Modeling and Control. *Wiley*. URL: https://surl.li/awzerb (дата звернення: 07.05.2025).

13. Learning OpenCV 4: Computer Vision with Python. *Github*. URL: https://github.com/PacktPublishing/Learning-OpenCV-4-Computer-Vision-with-Python-Third-Edition (дата звернення: 07.05.2025).

14. Які різні типи датчиків IO IP? *Savgoodtech*. URL: https://www.savgoodtech.com/uk/news/what-are-the-different-types-of-eo-ir-sensors-/(дата звернення: 07.05.2025).

15. Arduino Uno. *Arduino*. URL: https://docs.arduino.cc/hardware/uno-rev3/ (дата звернення: 07.04.2025).

16. Ультразвуковий датчик відстані Arduino HC. *Wiki*. URL: https://wiki.tntu.edu.ua_Arduino_HC_SR04 (дата звернення: 07.05.2025).

17. Arduino Motor Shield L298P. *Wiki.*. URL: https://wiki.tntu.edu.ua/Arduino_Motor_Shield_L298P(дата звернення: 07.05.2025)

18. Сервопривід SG90. *Towerpro*. URL: https://www. towerpro.com.tw/product/sg90/ (дата звернення: 07.05.2025).

ДОДАТКИ

Додаток А

Код Arduino Uno

```
unsigned char start01[] =
{0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80,0x80,0x40,0x20,0x10,0x08,0x04,0x02,0x01};
unsigned char front[] =
{0x00,0x00,0x00,0x00,0x00,0x24,0x12,0x09,0x12,0x24,0x00,0x00,0x00,0x00,0x00,0x00};
unsigned char back[] =
{0x00,0x00,0x00,0x00,0x00,0x24,0x48,0x90,0x48,0x24,0x00,0x00,0x00,0x00,0x00,0x00};
unsigned char left[] =
{0x00,0x00,0x00,0x00,0x00,0x00,0x44,0x28,0x10,0x44,0x28,0x10,0x44,0x28,0x10,0x00};
unsigned char right[] =
{0x00,0x10,0x28,0x44,0x10,0x28,0x44,0x10,0x28,0x44,0x00,0x00,0x00,0x00,0x00,0x00};
unsigned char STOP01[] =
{0x2E,0x2A,0x3A,0x00,0x02,0x3E,0x02,0x00,0x3E,0x22,0x3E,0x00,0x3E,0x0A,0x0E,0x00};
unsigned char clear[] =
{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00};
#define SCL_Pin  A5  //Set clock pin to A5
#define SDA_Pin  A4  //Set data pin to A4

#define ML_Ctrl 13  //define the direction control pin of left motor
#define ML_PWM 11   //define PWM control pin of left motor
#define MR_Ctrl 12  //define the direction control pin of right motor
#define MR_PWM 3   //define PWM control pin of right motor
#define Trig 5  //ultrasonic trig Pin
#define Echo 4  //ultrasonic echo Pin
int distance;
int pulsewidth;
#define servoPin 9  //servo Pin
void setup(){
  Serial.begin(9600);
  pinMode(SCL_Pin,OUTPUT);
  pinMode(SDA_Pin,OUTPUT);
  matrix_display(clear); //Clear the display
  matrix_display(start01);  //display start pattern
  pinMode(servoPin, OUTPUT);
  procedure(180); //set servo to 90°
  pinMode(Trig, OUTPUT);
  pinMode(Echo, INPUT);
  pinMode(ML_Ctrl, OUTPUT);
  pinMode(ML_PWM, OUTPUT);
  pinMode(MR_Ctrl, OUTPUT);
  pinMode(MR_PWM, OUTPUT);
}
void loop(){
  distance = checkdistance();  //assign the distance detected by ultrasonic sensor
to distance
  if (distance >= 20 && distance <= 60) //range to go front
  {
    Car_front();
  }
  else if (distance > 10 && distance < 20)  //range to stop
  {
    Car_Stop();
  }
  else if (distance <= 10)  //range to go back
  {
    Car_back();
  }
  else  //other situations, stop
  {
    Car_Stop();
```

```
  }
}
/***********the function for motor running***************/
void Car_front()
{
  digitalWrite(MR_Ctrl,LOW);
  analogWrite(MR_PWM,200);
  digitalWrite(ML_Ctrl,LOW);
  analogWrite(ML_PWM,200);
}
void Car_back()
{
  digitalWrite(MR_Ctrl,HIGH);
  analogWrite(MR_PWM,200);
  digitalWrite(ML_Ctrl,HIGH);
  analogWrite(ML_PWM,200);
}
void Car_left()
{
  digitalWrite(MR_Ctrl,LOW);
  analogWrite(MR_PWM,200);
  digitalWrite(ML_Ctrl,HIGH);
  analogWrite(ML_PWM,200);
}
void Car_right()
{
  digitalWrite(MR_Ctrl,HIGH);
  analogWrite(MR_PWM,200);
  digitalWrite(ML_Ctrl,LOW);
  analogWrite(ML_PWM,200);
}
void Car_Stop()
{
  digitalWrite(MR_Ctrl,LOW);
  analogWrite(MR_PWM,0);
  digitalWrite(ML_Ctrl,LOW);
  analogWrite(ML_PWM,0);
}

/******************dot matrix*******************/
// the function for dot matrix display
void matrix_display(unsigned char matrix_value[])
{
  IIC_start(); // call the function that data transmission start
  IIC_send(0xc0);  //Choose address

  for(int i = 0;i < 16;i++) //pattern data has 16 bits
  {
     IIC_send(matrix_value[i]); //data to convey patterns
  }

  IIC_end();   //end to convey data pattern

  IIC_start();
  IIC_send(0x8A);  //select pulse width4/16, control display
  IIC_end();
}

//The condition starting to transmit data
void IIC_start()
{
  digitalWrite(SCL_Pin,HIGH);
  delayMicroseconds(3);
  digitalWrite(SDA_Pin,HIGH);
  delayMicroseconds(3);
```

```
  digitalWrite(SDA_Pin,LOW);
  delayMicroseconds(3);
}

// transmit data
void IIC_send(unsigned char send_data)
{
  for(char i = 0;i < 8;i++)  //Each byte has 8 bits
  {
      digitalWrite(SCL_Pin,LOW);  //pull down clock pin SCL Pin to change the
signals of SDA
delayMicroseconds(3);
      if(send_data & 0x01)  //set high and low level of SDA_Pin according to 1 or
0 of every bit
      {
        digitalWrite(SDA_Pin,HIGH);
      }
      else
      {
        digitalWrite(SDA_Pin,LOW);
      }
      delayMicroseconds(3);
      digitalWrite(SCL_Pin,HIGH); //pull up clock pin SCL_Pin to stop transmitting
data
      delayMicroseconds(3);
      send_data = send_data >> 1;  // detect bit by bit, so move the data right by
one
  }
}
//The sign that data transmission ends
void IIC_end()
{
  digitalWrite(SCL_Pin,LOW);
  delayMicroseconds(3);
  digitalWrite(SDA_Pin,LOW);
  delayMicroseconds(3);
  digitalWrite(SCL_Pin,HIGH);
  delayMicroseconds(3);
  digitalWrite(SDA_Pin,HIGH);
  delayMicroseconds(3);
}
/***************end dot matrix display*****************/
//The function to control servo
void procedure(int myangle) {
  for (int i = 0; i <= 50; i = i + (1)) {
    pulsewidth = myangle * 11 + 500;
    digitalWrite(servoPin,HIGH);
    delayMicroseconds(pulsewidth);
    digitalWrite(servoPin,LOW);
    delay((20 - pulsewidth / 1000));
  }}
//The function to control ultrasonic sensor function controlling ultrasonic
float checkdistance() {
  digitalWrite(Trig, LOW);
  delayMicroseconds(2);
  digitalWrite(Trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(Trig, LOW);
  float distance = pulseIn(Echo, HIGH) / 58.20;  //58.20, that is , 2*29.1=58.2
  delay(10);
  return distance;
}
```