

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

До захисту допущено:

Завідувач кафедри

_____ Олександр РОЛІК

« ___ » _____ 20__ р.

Дипломний проєкт

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Інтегровані інформаційні системи»

спеціальності 126 «Інформаційні системи та технології»

на тему: «Система навігації самохідної техніки на базі Arduino»

Виконав:

Студент 4 курсу, групи ІА-94

Саліхов Михайло Михайлович _____

Керівник:

к.т.н., доцент

Ткач Михайло Мартинович _____

Рецензент:

к.т.н., доцент

Ліхоузова Тетяна Анатоліївна _____

Засвідчую, що у цьому дипломному проєкті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____

Київ – 2023 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інтегровані інформаційні системи»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 20__ р.

ЗАВДАННЯ

на дипломний проєкт студенту
Саліхову Михайлу Михайловичу

1. Тема проєкту «Система навігації самохідної техніки на базі Arduino», керівник проєкту Ткач Михайло Мартинович, старший викладач, затверджені наказом по університету від «31» травня 2023 р. №2101-с.
2. Термін подання студентом проєкту: 12 червня 2023 року
3. Вихідні дані до проєкту: технічна документація апаратного і програмного комплексу системи навігації мобільного робота. Технології проєктування та створення програмованого робота на базі Arduino, що уникає перешкод.
4. Зміст пояснювальної записки: Опис предметної області, розгляд існуючих рішень. Реалізація апаратно програмного комплексу навігації та створення програмованого робота на базі Arduino.
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо): Блок-схема алгоритму системи навігації мобільного робота
6. Дата видачі завдання: 1 березня 2023 року

Календарний план

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1	Аналіз предметної області	17.04.2023	
2	Огляд існуючих рішень	20.04.2023	
3	Підготовка необхідних інструментів та бази апаратного забезпечення.	03.05.2023	
4	Вибір засобів та методів реалізації проекту	08.05.2023	
5	Розробка архітектури системи	9.05.2023	
6	Реалізація моделі робота	10.05.2023	
7	Розробка програмної частини проекту	23.05.2023	
8	Тестування та усунення недоліків моделі	25.05.2023	
9	Опис функцій та обмежень робота	27.05.2023	
10	Оформлення документації	05.06.2023	
11	Подання готового проекту	12.06.2023	

Студент

Михайло САЛІХОВ

Керівник

Михайло ТКАЧ

АНОТАЦІЯ

Саліхов М.М. Система навігації самохідної техніки на базі Arduino. КПІ ім. Ігоря Сікорського, Київ, 2023.

Пояснювальна записка дипломного проєкту складається з 3 розділів, містить 85 сторінок тексту, 59 рисунків, 3 таблиці, 4 додатки, 59 інформаційних посилань.

Ключові слова: навігація, самохідна техніка, мобільний робот, алгоритми уникнення перешкод, датчики, ультразвуковий датчик, Arduino.

Об'єктом розробки є система навігації самохідної техніки. Мета розробки – створення системи управління мобільного колісного робота з використанням Arduino, розробка мобільного колісного робота.

У дипломному проєкті створено та реалізовано апаратно-програмний комплекс мобільного робота-автомобіля здатного орієнтуватися у просторі у режимі реального часу та уникати перешкоди, спираючись на: мікроконтролерну плату Arduino UNO, шилд драйверу двигунів Adafruit Motor Shield; використовується код, написаний у програмному забезпеченні Arduino IDE із застосуванням бібліотек AF Motor та Servo.

Отримані результати можуть бути корисними при подальших дослідженнях та розробках різних систем навігації мобільних роботів, завдання яких – пересуватися у невизначеному середовищі з численними перешкодами.

SUMMARY

Salikhov M.M. Arduino-based self-driving vehicle navigation system. Igor Sikorsky KPI, Kyiv, 2023.

The explanatory note of the diploma project consists of 3 sections, contains 85 pages of text, 59 figures, 3 tables, 4 annex, 59 information references.

Keywords: navigation, self-driving vehicle, mobile robot, obstacle avoidance algorithms, sensors, ultrasonic sensor, Arduino.

Object of development is a navigation system of self-driving vehicle. The aim of the development is to create an Arduino-based mobile wheeled robot control system, to develop a four-wheeled mobile robot.

In this diploma project were created and implemented - a hardware and software complex mobile robot-car, able to navigate in its surroundings and avoid obstacles in real time, created with: Arduino UNO microcontroller board, Adafruit Motor Driver Shield, servomotor and DC motors; code was written in Arduino IDE software using AFMotor and Servo libraries.

The results obtained can be useful for further research and development of various navigation systems of autonomous mobile robots, which task is to move in an uncertain environment with numerous obstacles.

Номер рядка	Формат	Позначення	Найменування	Кільк. аркушів	Номер екзем.	Примітка
1			<u>Документація загальна</u>			
2						
3			Знову розроблена			
4						
5	A4	IA94.150БАК.004 ПЗ	Пояснювальна записка	106		
6	A3	IA94.150БАК.004 Д1	Габаритне креслення плати			
7			Arduino Uno R3			
8	A3	IA94.150БАК.004 Д2	Блок-схема алгоритму системи	1		
9			навігації мобільного робота			
10	A3	IA94.150БАК.004 Д3	Блок-схема процесу побудови	1		
11			мобільного робота			
12	A3	IA94.150БАК.004 Д4	Схема з'єднань компонентів	1		
13			мобільного робота з шилдом			
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						
28						

IA94.150БАК.004 ТП

Зм.	Аркуш	№ докум.	Підпис	Дата			
Розроб.		Саліхов			Літ.	Аркуш	Аркушів
Керівн.		Ткач				1	1
					КПІ ім. Ігоря Сікорського		
					Група IA-94		
Затв.							

Система навігації самохідної
техніки на базі Arduino
Відомість проекту

**Пояснювальна записка
до дипломного проекту
на тему: «Система навігації самохідної техніки
на базі Arduino»**

Київ – 2023

ЗМІСТ

ВСТУП	3
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ	5
1.1 Самохідна техніка: загальний огляд об'єкта дослідження	5
1.1.1 Понятійний апарат	5
1.1.2 Архітектурна модель	9
1.2 Огляд існуючих рішень	11
1.3. Проблеми забезпечення автономності транспортного засобу	20
1.4. Прототипування на базі Arduino	22
1.5 Постановка задачі	27
ВИСНОВКИ РОЗДІЛУ 1	28
2 АНАЛІЗ ОСНОВНИХ СПОСОБІВ НАВІГАЦІЇ САМОХІДНОЇ	29
ТЕХНІКИ З УХИЛЕННЯМ ВІД ПЕРЕШКОД	
2.1 Автономна навігація	29
2.2 Алгоритми навігації	35
2.3 Алгоритми планування маршруту	37
ВИСНОВКИ РОЗДІЛУ 2	44
3 ПРОЕКТУВАННЯ ТА СТВОРЕННЯ ПРОГРАМОВАНОГО РОБОТА,	46
ЩО УНИКАЄ ПЕРЕШКОД, НА БАЗІ ARDUINO	
3.1 Апаратні засоби проектування мобільного робота	46
3.2. Побудова мобільного робота, що уникає перешкод, на базі Arduino	64
3.3 Програмні засоби проектування мобільного робота	70
3.4 Тестування мобільного робота	73
3.5 Обговорення результатів проекту	74
ВИСНОВКИ РОЗДІЛУ 3	75
ВИСНОВКИ	77
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	79
ДОДАТКИ	85

ІА94.150БАК.004 ПЗ				
Зм.	Арк.	№ докум.	Підпис	Дата
Розробив		Саліхов М.М.		
Перевірив		Ткач М.М.		
Реценз.				
Н. Контр.				
Затвердив				
Система навігації самохідної техніки на базі Arduino				
Пояснювальна записка				
		Літ.	Аркуш	Аркушів
		2	2	85
«КПІ ім. Ігоря Сікорського» ФІОТ Група ІА-94				

ВСТУП

Актуальність дослідження. Технології безпілотного водіння наземних транспортних засобів вже кілька десятиліть перебувають у фокусі науковців та інженерів. Виробники автомобілів, компанії з випуску компонентів до них, а також представники сфери ІТ працюють над механізмами підвищення автоматизації водіння, стандартизацією отриманих технологічних рішень, їхньою комерціалізацією, створенням принципово нових машин – робокарів. Їх поширення змінить життя так само, як раніш це зробили паровози та автомобілі. Безпілотні автомобілі можуть забезпечити мобільність навіть тим людям, хто не може керувати машиною власноруч (наприклад, людям похилого віку чи інвалідам). Все це обумовлює необхідність подальшого вивчення технології потужних процесорів, якісних датчиків, складних алгоритмів, картографування і в цілому створення систем навігації автотранспорту. Такі наприми є пріоритетними у сучасних наукових роботах вчених різних галузей знань, в т.ч. інформаційних систем й технологій та автоматизації процесів керування, мета яких – вдосконалення існуючих та створення нових підходів до забезпечення автономності та безпеки водіння.

З огляду на зазначене вище, нагальним є вирішення проблем алгоритмізації систем навігації транспортних засобів та побудова моделей їх пересування. Прототипування та тестування нових рішень за допомогою Arduino може значно прискорити досягнення зазначених цілей через краще розуміння засад та принципів створення робокарів. Тому актуальною є розробка системи навігації самохідної техніки на базі Arduino, що обумовило тему, предметну область, мету та завдання даного дослідження.

Об'єктом розробки є система навігації самохідної техніки.

Предмет дослідження – методи та алгоритми навігації самохідної техніки, а також апаратно-програмне забезпечення навігації на базі Arduino.

Мета дослідження – створення системи управління мобільного колісного робота з використанням Arduino, розробка мобільного колісного робота.

					ІА94.150БАК.004 ПЗ	Арк.
						3
Зм.	Арк.	№ докум.	Підпис	Дат		

Досягнення поставленої мети передбачає розв'язання таких задач:

- 1) розкрити зміст понять «самохідна техніка» та «система навігації»; обґрунтувати доцільність використання платформи Arduino для створення прототипу самохідної техніки;
- 2) виявити проблеми автономності наземних транспортних засобів;
- 3) визначити базовий апаратний та програмний інструментарій створення прототипу самохідного транспортного засобу на базі Arduino;
- 4) створити програмне забезпечення для реалізації запропонованого алгоритму; здійснити кодування програм, використовуючи Arduino IDE та мову програмування C/C++; розробити схему з'єднання компонентів та створити прототип робота-автомобіля;
- 5) здійснити тестування моделі, її руху та здатності уникати перешкоди; оцінити отримані результати.

Практичне значення результатів роботи полягає у створенні алгоритму уникання перешкод та прототипу самохідного колісного робота на його основі, що може знайти застосування у процесі розробки роботів для виконання завдань у замкнених просторах. Зокрема, перспективним є застосування авторського доробку при розробках візків для пересування поранених, хворих, інвалідів, людей похилого віку у закладах охорони здоров'я.

Апробація роботи. Отримані результати дослідження було представлено на:

- X Всеукраїнській мультидисциплінарній науково-практичній Інтернет-конференції «Наукові досягнення та інновації: шлях до успіху» 31 травня 2023 р., м. Київ;
- Міжнародній науково-практичній конференції «Сучасні аспекти діджиталізації та інформатизації в програмній та комп'ютерній інженерії» 1- 03 червня 2023 р., м. Київ;
- VIII Міжнародній науково-практичній конференції «Концептуальні шляхи розвитку науки та освіти» 9-10 червня 2023 р., м. Львів;
- VIII Міжнародній науково-практичній конференції «Практичні та теоретичні питання розвитку науки та освіти» 19-20 червня 2023 р., м. Львів.

					ІА94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		4

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Самохідна техніка: загальний огляд об'єкта дослідження

1.1.1 Понятійний апарат

У традиційному розумінні самохідна техніка асоціюється, передусім, з автомобілем (від грец. αὐτός — «сам» і лат. mobilis — «той, що рухається») — самохідною колісною машиною, яка приводиться в рух встановленим у неї двигуном і призначена для перевезення людей, вантажу, буксирування транспортних засобів, виконання спеціальних робіт та перевезення спеціального устаткування безрейковими дорогами. Але з розвитком технологій, зокрема радіотехніки, автомобілі ставали автоматизованими. У 1920-х рр. було представлено машину з дистанційним управлінням, яку, на думку автора роботи [1], можна вважати першим безпілотним автомобілем, керований бездротовим способом через радіо (рис. 1.1)

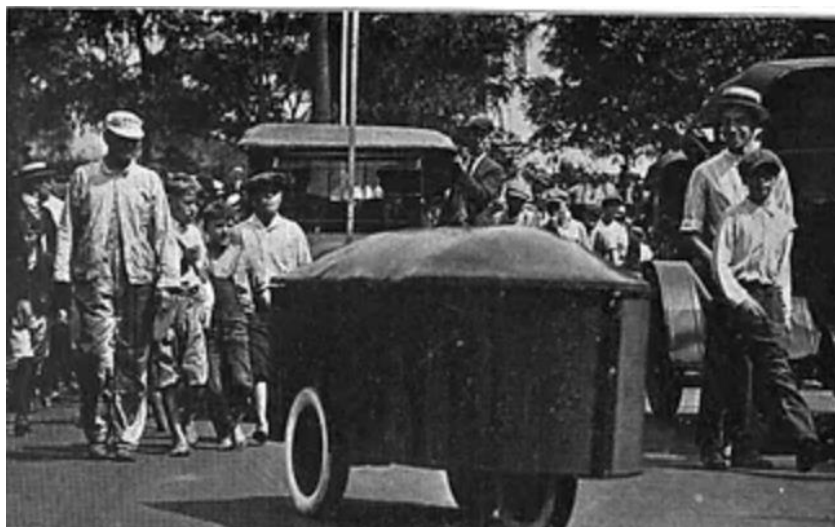


Рисунок 1.1 – Радіокерований автомобіль

На початку минулого століття журнал Scientific American опублікував концепт самохідного транспортного засобу, подавши його як «мрію автомобіліста – автомобіль, який керується набором кнопок». У статті було зазначено, що «... в майбутньому машина з кермом буде такою ж застарілою, як сьогодні машина з ручним насосом для бензину чи масла!» [1]. Автомобілебудування зазнало

					ІА94.150БАК.004 ПЗ	Арк.
						5
Зм.	Арк.	№ докум.	Підпис	Дат		

швидких перетворень у минулому столітті, але справжній прорив відбувся на початку 2000-х завдяки розробкам у галузі інформаційно-комунікаційних технологій, напівпровідників та робототехніці (рис. 1.2).



Рисунок 1.2 – Мрія та реальність: самохідна техніка

Значних зусиль доклало Агентство передових оборонних дослідницьких проєктів Міністерства оборони США (DARPA), яке стало піонером у цій галузі, представивши наприкінці 1980-х рр. нову розробку (рис. 1.3), яку назвали «автономний наземний транспортний засіб» (Autonomous Land Vehicle – ALV), – восьмиколійний робот заввишки 12 футів із кількома датчиками, здатний переміщатися з точки А в точку Б без втручання людини [2].

					ІА94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		6



Рисунок 1.3 – ALV, розроблений DARPA

Згодом у науковий обіг увійшов ще один термін – «безпілотний наземний транспортний засіб» (unmanned ground vehicle - UGV), котрий працює при контакті із землею та без присутності людини на борту [3].

Нині у документах ЄС застосовують такі поняття [4]:

Автоматизований транспортний засіб (Automated vehicle): автотранспортний засіб (автомобіль, вантажівка або автобус), оснащений технологіями (задля допомоги водієві), здатними передавати реалізацію елементів завдання водіння до комп'ютерної системи;

Автономний транспортний засіб (Autonomous vehicle): повністю автоматизований транспортний засіб, оснащений технологіями, здатними виконувати всі функції керування без втручання людини.

Національна Адміністрація Безпеки Дорожнього Руху США вважає автомобіль автономним, якщо він функціонує, не вимагаючи від водія безпосередньої участі у реалізації функцій водіння, включаючи «кермування, прискорення та гальмування» [5].

На думку автора роботи [6, с. 150], *автономний транспортний засіб* - це неофіційний термін, який використовується для опису автомобіля, обладнаного автоматизованою системою керування (automated driving system - ADS), який працює в автономному режимі, тобто автомобіль, що має високий рівень автоматизації керування.

					ІА94.150БАК.004 ПЗ	Арк.
						7
Зм.	Арк.	№ докум.	Підпис	Дат		

Як зазначають фахівці SAE International (раніше відома як Товариство автомобільних інженерів – SAE), деякі традиційно пов'язують автономію водіння безпосередньо з повною автоматизацією рівня 5; тоді як інші – застосовують це поняття до всіх рівнів автоматизації водіння; при цьому в законодавстві окремих штатів визначено, що автономність приблизно відповідає будь-якій ADS щонайменше рівня 3 [7].

Українські вчені Поліщук М.М. і Ткач М.М. зауважують, що «роботизовані транспортні засоби, що перебувають в експлуатації, можна розділити на дві основні групи: транспортні роботи із твердим шляхопроводом і робокари — транспортні візки з безконтактним індуктивним або оптичним шляхопроводом. ... Системи керування таких роботів можуть бути цикловими, позиційними з автоматичним адресуванням. Усі системи, як правило, побудовані на базі локального автомата з керуванням від центрального контролера ЕОМ нижнього й верхнього рівня. Вибір системи керування, як правило, визначається функціональним призначенням транспортного робота» [8, с.31]. Питання роботизованих транспортних засобів також розглянуто в роботах [9-11]. Оскільки компоненти таких автомобілів є як фізичними, так і віртуальними, вчені називають такі машини «кіберфізичні системи» (cyber-physical systems).

Водночас, експерти SAE проводили опитування 257 респондентів щодо загальної назви зазначених вище машин, яке показало, що 44,5% вибрали «self-driving car» (самокерований автомобіль), оскільки саме цей термін використовувався Waymo LLC (раніш відому як Google Self-Driving Car Project) з того часу, як компанія представила свій проєкт кілька років тому. [7].

Беручи до уваги такі висновки, у роботі автор використовує термін – «self-driving car» (SDC), припускаючи, що «самокерована» є більш точною характеристикою самохідної техніки, здатної рухатися без участі людини. Як синонім використовується термін «мобільний робот» чи «робот-автомобіль».

					ІА94.150БАК.004 ПЗ	Арк.
						8
Зм.	Арк.	№ докум.	Підпис	Дат		

1.1.2 Архітурна модель

Існують різні площини, за якими розглядають архітурну модель SDC: фізичні компоненти, етапи розробки, логічних функції, технологічні блоки тощо (рис.1.4).

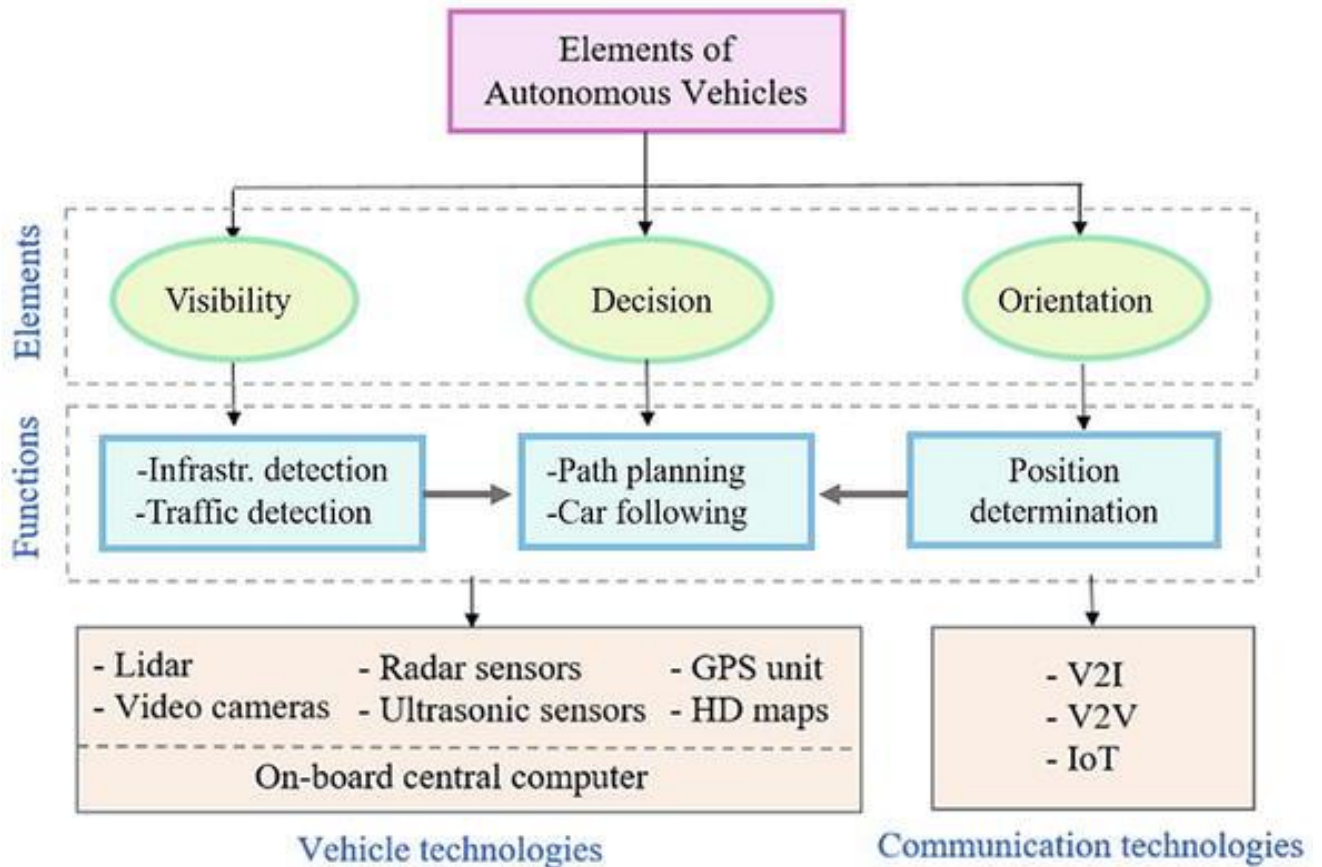


Рисунок 1.4 - Елементи SDC, їх функції та базові технології [12]

Для реалізації мети дослідження у цій роботі SDC розглянуто з технічної та функціональної точок зор.

А. Технічна точка зору

Апаратне та програмне забезпечення є двома основними площинами технічного вигляду архітектури SDC, і кожен шар включає компоненти, які представляють різні аспекти всієї системи [11-12] (рис. 1.5).

Для того, щоб обробляти інформацію в SDC використовують обчислювальні платформи з декількома ядрами, графічні процесори (Graphical Processing Units – GPUs).

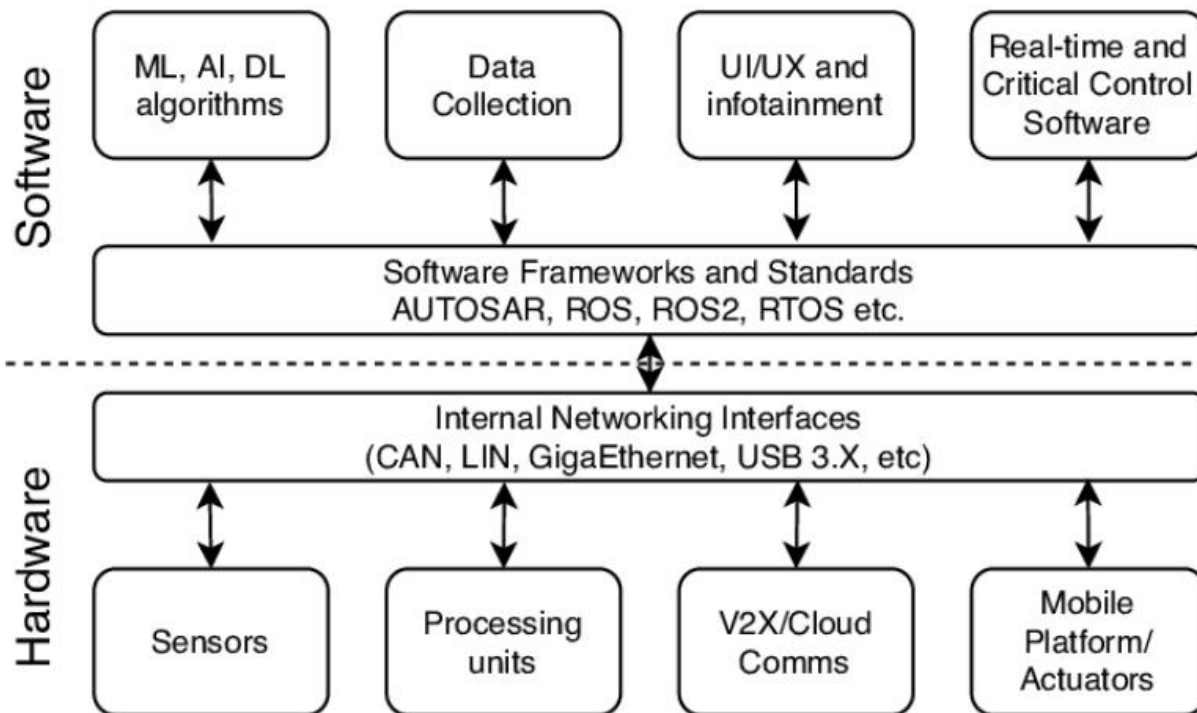


Рисунок 1.5 – Технічна архітектура системи SDC [11]

Разом із даними, що генеруються безпосередньо SDC, обробляються також зовнішні дані, доступні з Інтернету, інших транспортних засобів або інфраструктури, що називається V2X (Vehicle-to-everything). Апаратна частина також включає сам транспортний засіб, це мобільна платформа та певні пристрої, які можуть бути різних видів залежно від застосування та місцевості, де буде працювати система. Можливості обробки поточних даних SDC такі, що їх іноді називають суперкомп'ютерами на колесах.

Це твердження не далеке від реальності, оскільки через складність в апаратній частині, програмна – також розвивається. Як приклад – AUTOSAR (AUTomotive Open System Architecture) – це сучасна структура та стандарт для екосистеми автомобільної електроніки. Вона створена і управляється альянсом

таких гігантів галузі, як BMW, Toyota, GM, Chrysler, Continental, Bosch, Daimler, Volkswagen та багато інших. AUTOSAR заснований на мові C [13].

Б. Функціональна точка зору

В основі реалізації функцій – чотири основні блоки (зустрічаються у більшості запропонованих архітектур і рішень): сприйняття, планування і прийняття рішень, управління рухом і транспортним засобом, системний нагляд. Ці блоки представлені на рис. 1.6.

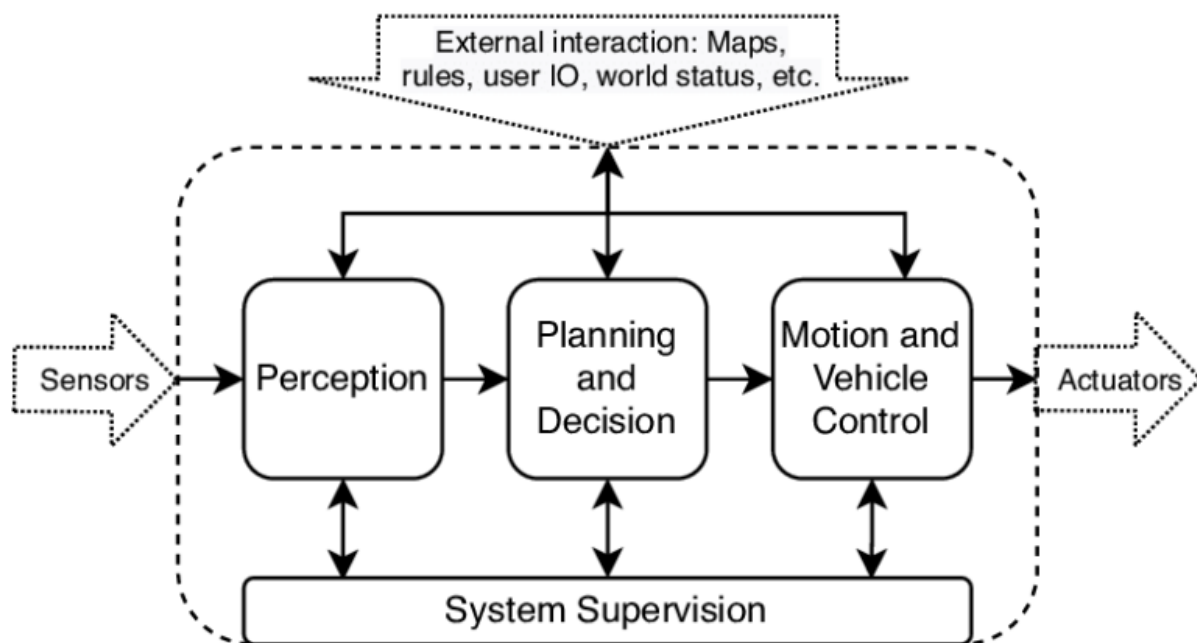


Рисунок 1.6 – Функціональна архітектура системи автономного водіння [11]

1.2 Огляд існуючих рішень

Розробники зосереджуються на певному наборі технологій для автономної навігації, зокрема, пов'язаних із уникненням перешкод. Це завдання зазвичай вирішується за допомогою різних типів датчиків і алгоритмів управління, що виконуються на мікроконтролерах або мікрокомп'ютерах.

Підґрунтям для переходу від автоматизації до автономності транспортного засобу є впровадження системи ADAS, яка являє собою групу електронних технологій, котрі допомагають водіям керувати автомобілем і паркуватися,

						ІА94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат			11

використовуючи автоматизовані технології, спираючись на датчики та камери (рис. 1.7), щоб виявляти перешкоди та реагувати відповідним чином [14].

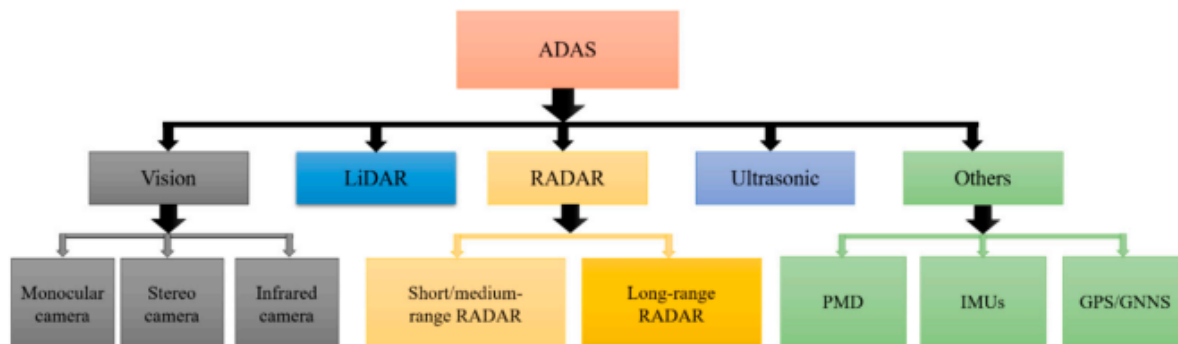


Рисунок 1.7 – Таксономія базових компонентів ADAS [14]

ADAS має різні типи систем: адаптивний круїз - контроль (ACC), адаптивне переднє світло (AFL), автоматичне аварійне гальмування (AEB), виявлення сліпих зон (BSD), попередження про перехресний рух (CTA), система моніторингу водія (DMS), попередження про лобове зіткнення (FCW), Інтелектуальна система допомоги при паркуванні (IPA), попередження про вихід зі смуги руху (LDW), система нічного бачення, система виявлення пішоходів (PDS), розпізнавання дорожніх знаків (RSR), система контролю тиску в шинах (TPMS), асистент руху у «пробці» (TJA).

Для надійної та безпечної навігації SDC зазвичай використовують комбінацію датчиків (рис. 1.8), які мають різну фізичну природу та базуються на різних технологіях (рис. 1.9), мають різне призначення та різне покриття оточуючого середовища (рис. 1.10).

Для сприйняття безпосереднього оточення автомобіля використовуються датчики виявлення об'єктів [15, 17]. Їх розрізають як:

- пасивні датчики – виявляють наявну енергію, як-от світло чи випромінювання, що відбивається від об'єктів у навколишньому середовищі;
- активні датчики – посиляють власний акустичний чи електромагнітний сигнал і сприймають його відбиття.

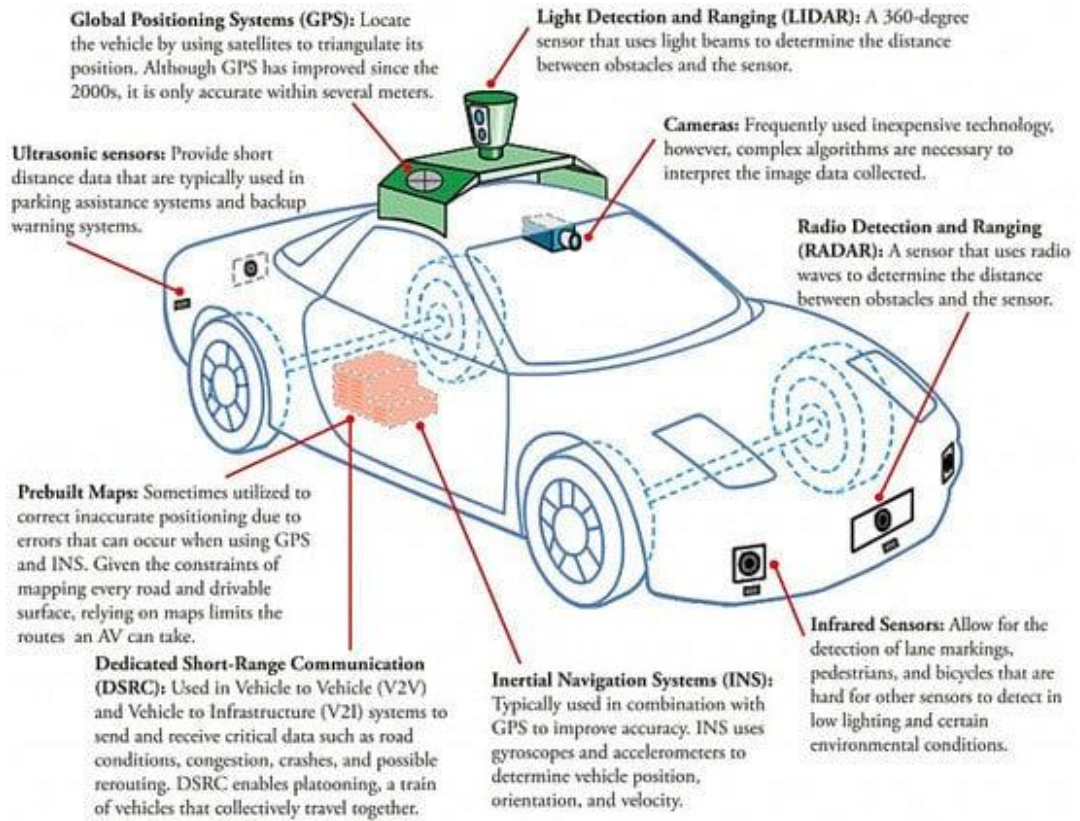


Рисунок 1.8 – Екосистема датчиків SDC [15]

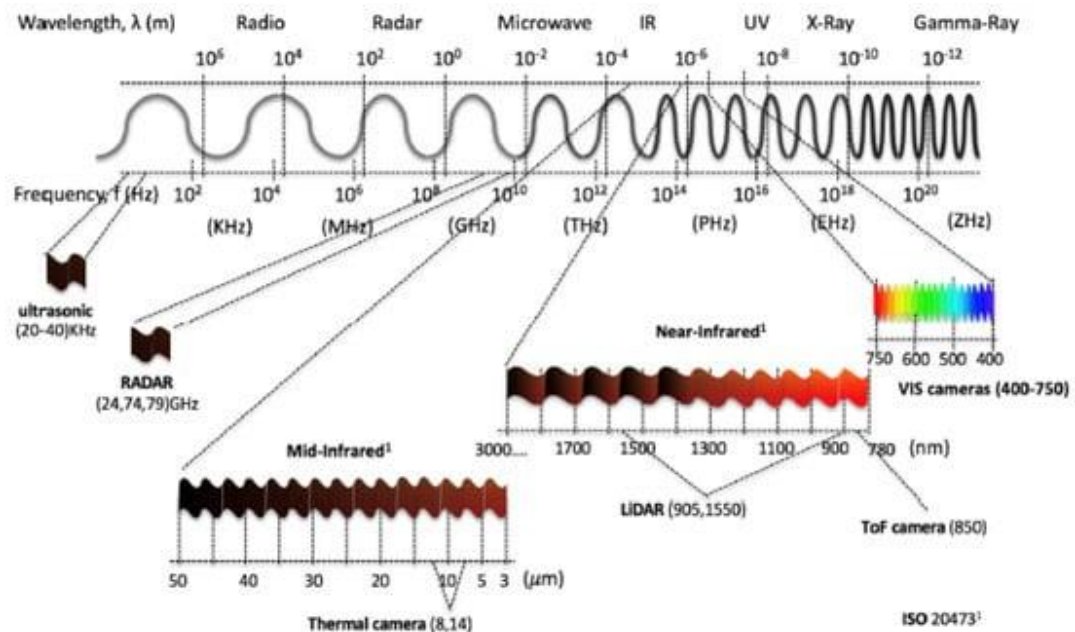


Рисунок 1.9 – Звукові та електромагнітні спектри, які використовуються датчиками SDC [15]

Зм.	Арк.	№ докум.	Підпис	Дат

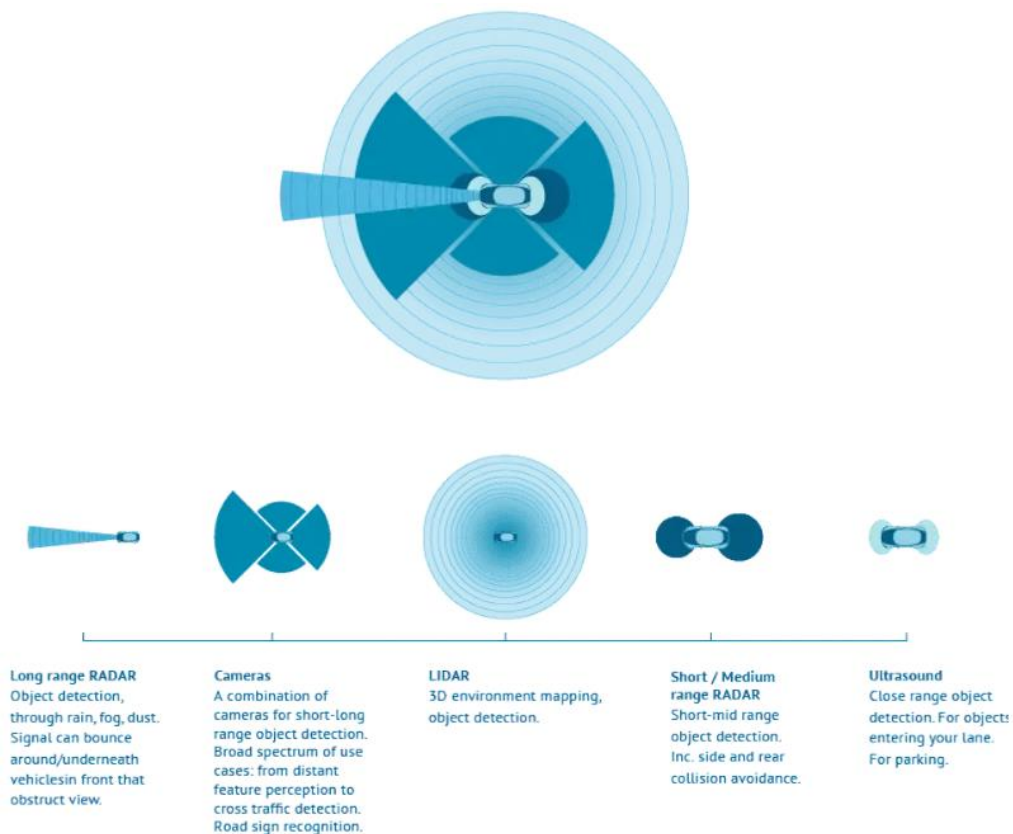


Рисунок 1.10 – Діапазон покриття датчиків [16].

До пасивних датчиків належать наступні.

Цифрові камери як пасивні датчики використовують сенсори зображення пристрою із зарядовим зв'язком (charge-coupled device – CCD) або комплементарний металоксид-напівпровідник (complementary metal-oxide semiconductor - CMOS) [18].

Камери використовуються для розпізнавання дорожніх знаків, читання ліній та іншої розмітки на дорозі, виявлення пішоходів, перешкод та багато іншого. Як приклад (рис. 1.11), BMW X5 використовує камеру TriFocal із MobilEye EyeQ4 (камера ZF S-Cam 3 має два основних компоненти – плату з процесором MobilEye EyeQ3 і плату з датчиком зображення CMOS від ON Semiconductor), що вважається прогресом у безпілотній навігації [19].

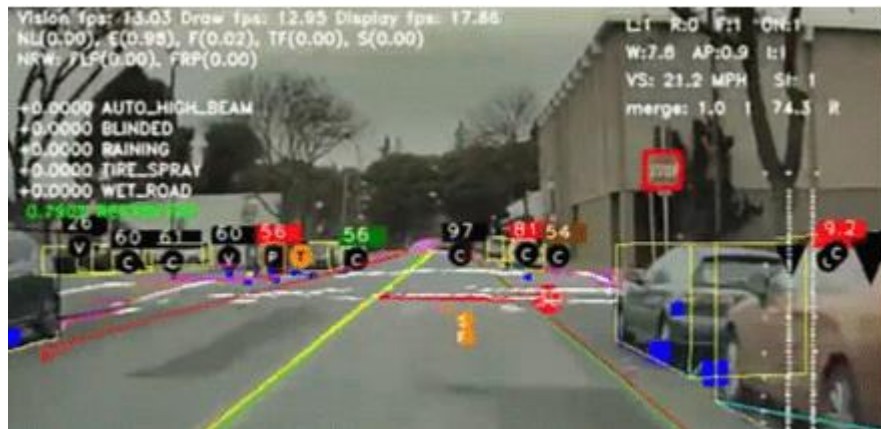
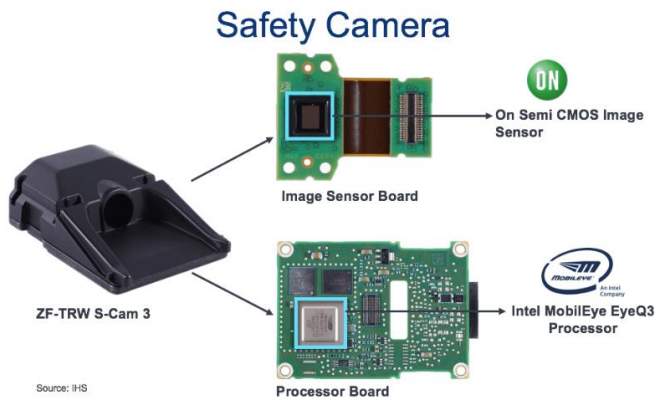


Рисунок 1.11 – Дані сприйняття камери, накладені на відеозображення [20]

Основні переваги пасивних датчиків: висока роздільна здатність у пікселях і колір по всій ширині поля зору; постійна «частота кадрів» у полі зору; дві камери можуть створювати 3D-стереоскопічне зображення; відсутність джерела передачі знижує ймовірність перешкод від іншого автомобіля; низька вартість завдяки вдосконаленій технології; зображення, створені цими системами, користувачі легко розуміють і з ними легко взаємодіяти.

Основні недоліки пасивних датчиків: низька ефективність при слабкому освітленні або поганих погодних умовах (через те, що вони не мають власного джерела передачі, вони не можуть легко адаптуватися до цих умов); генерування 0,5-3,5 Гбіт/с даних, що може бути багато для внутрішньої обробки або передачі даних у хмару (це більше, ніж обсяг даних, який генерують активні датчики).

До активних датчиків належать наступні.

Датчики RADAR (Radio Detection and Ranging) використовуються для виявлення великих об'єктів перед автомобілем. Вони зазвичай використовують частоту 76,5 ГГц. Використовуються два основні методи: пряме поширення, непряме поширення. Однак в обох випадках вони працюють за допомогою випромінювання цих радіочастот та вимірювання часу розповсюдження відбитих сигналів. Це дозволяє датчикам вимірювати як розмір та відстань до об'єкта, так і його відносну швидкість. RADAR може «бачити» крізь погану погоду та інші перешкоди видимості. Оскільки довжина їх хвиль становить лише кілька міліметрів, вони можуть виявляти об'єкти розміром у кілька сантиметрів. Ці датчики необхідні для попередження та пом'якшення наслідків зіткнення, виявлення сліпих зон (рис. 1.12), наданні допомоги при зміні смуги руху та при паркуванні тощо [21].

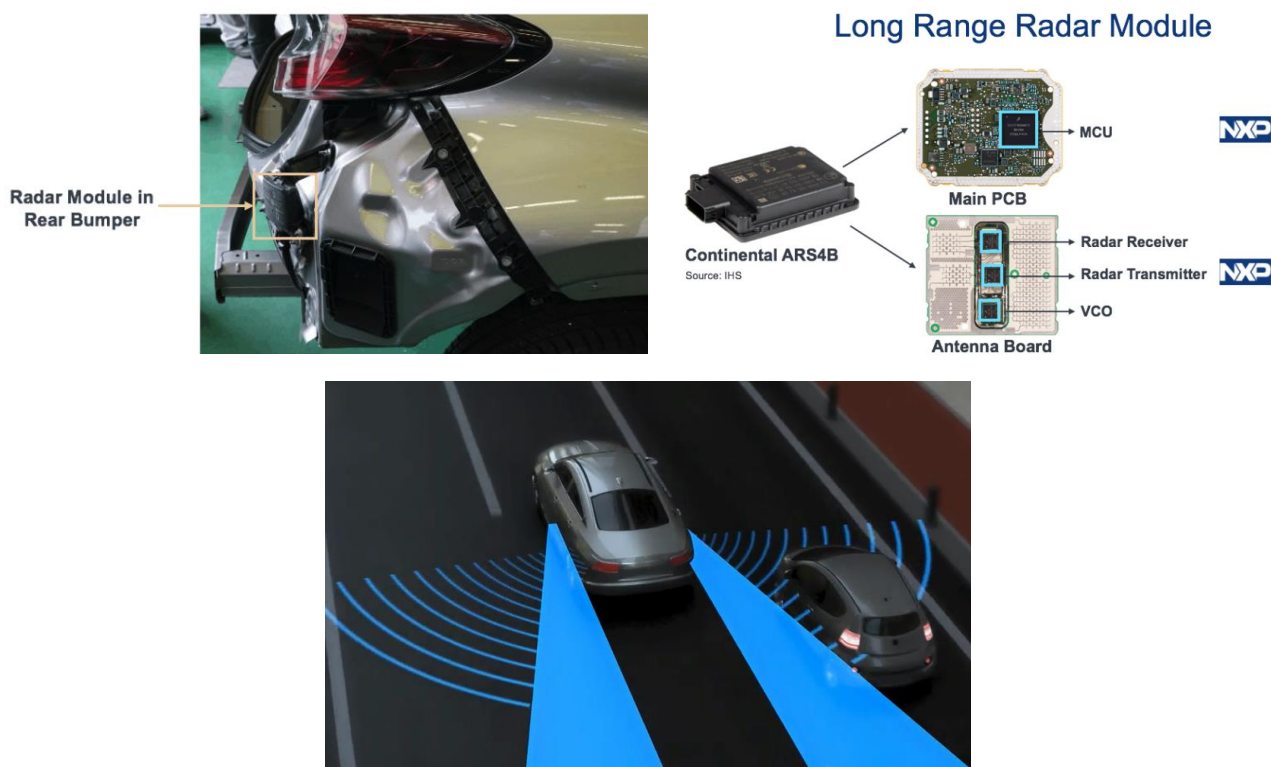


Рисунок 1.12 – Транспортний засіб, обладнаний радаром для виявлення сліпих зон [22]

Ультразвуковий датчик – це пристрій, що використовує принцип ехолокації. На рис. 1.13 зображена типова ультразвукова ехолокація датчика SDC. [23-25]

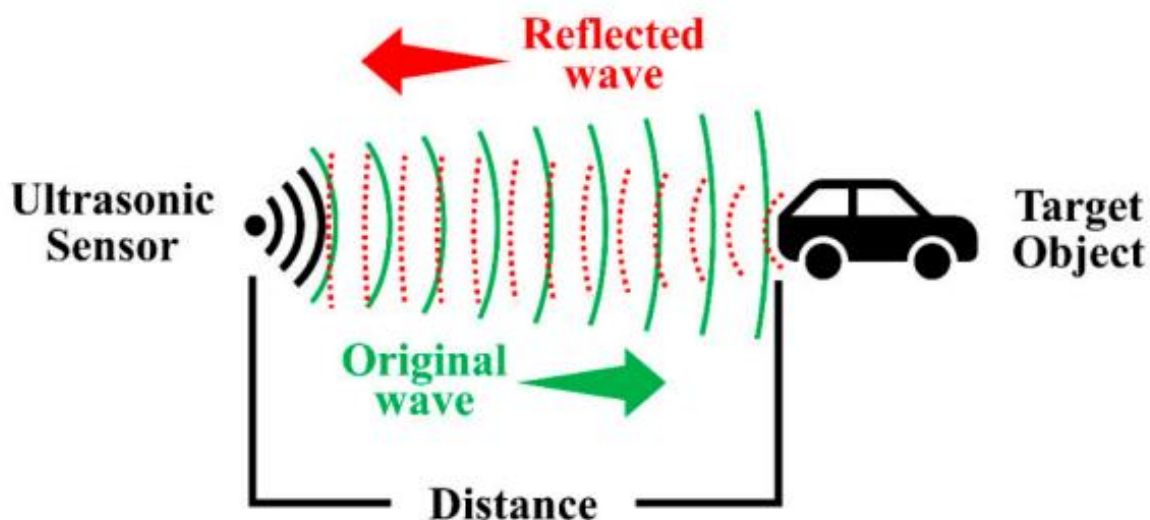


Рисунок 1.13 – Принцип дії ультразвукового датчика

Ці датчики використовуються для навігації та визначення дальності; генерують високочастотний звук близько 48 кГц, що більш ніж удвічі перевищує звичайний діапазон людського слуху, але мають обмежений діапазон порівняно із радаром. Але якщо об'єкт (перешкода) знаходиться на близькій відстані (для автомобіля – від 2,5 до 4,5 м), ультразвук є менш дорогою альтернативою радару. Характеристики цього датчика можуть погіршитися через надмірний сторонній шум і температуру середовища. Ультразвукові датчики мають сліпі зони в безпосередній близькості, що може призвести до неточних показань. Матеріали з високими коефіцієнтами поглинання звукових хвиль і демпфуючими властивостями, такі як акустична піна, також можуть впливати на показання ультразвукового датчика. Та попри зазначені обмеження, багато SDC, представлених нині на ринку, використовують ультразвукові датчики в інтелектуальній системі паркування (рис. 1.14).

LIDAR (Light Detection and Ranging) [27] — це датчик, що використовує імпульсні лазери для створення хмар точок, що дозволяє отримати 3D-зображення карти (рис. 1.15). Датчики LIDAR надсилають 50 000–200 000 імпульсів на секунду. Порівнюючи різницю в хмарах точок, що сприймаються один за одним, можна виявити об'єкти та їхній рух таким чином, щоб створити 3D-карту з діапазоном до 250 м [28].

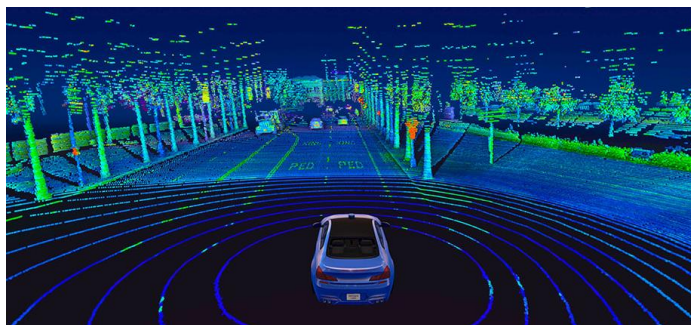


Рисунок 1.16 - Хмара точок, створена датчиком Alpha Prime компанії Velodyne Lidar [29]

Вони все частіше використовуються у поєднанні з камерами, тому що LiDAR не може розпізнавати кольори (наприклад, на світлофорах), а також не може читати текст так само добре, як камери. Порівняння функцій розглянутих датчиків подано у табл 1.

Таблиця 1 – Порівняльна характеристика датчиків SDC

Функція	LiDAR	RADAR	Camera	Ultrasonic
Первинна технологія	лазерний луч	радіохвиля	світло	звукова хвиля
Діапазон (м)	до 250	0.2-300	до 250	0.02-10
Роздільна здатність	добре	середньо	дуже добре	слабко
Уразливість від погодних умов	так	так	так	так
Впливає на умови освітлення	ні	ні	так	ні
Визначає швидкість	добре	дуже добре	слабко	слабко
Визначає відстань	добре	дуже добре	слабко	добре
Сприйнятливність до перешкод	добре	слабко	дуж. добре	добре
Розмір	великий	малий	малий	малий

1.3. Проблеми забезпечення автономності транспортного засобу

У 2014 році SAE представила стандарт J3016 «Рівні автоматизації водіння», який визначив шість рівнів – від SAE 0, де водій повністю контролює транспортний засіб, до рівня SAE 5, де транспортні засоби можуть контролювати всі аспекти динамічних завдань водіння без втручання людини. У 2021 р. було подано наступну ітерацію документу SAE J3016 (рис. 1.17).

	SAE РІВЕНЬ 0	SAE РІВЕНЬ 1	SAE РІВЕНЬ 2	SAE РІВЕНЬ 3	SAE РІВЕНЬ 4	SAE РІВЕНЬ 5
Що повинен робити водій?	Ви керуєте автомобілем, навіть якщо ноги не на педалях, а руки не на рулі.			Ви <u>не</u> керуєте автомобілем, якщо активовані функції автоматичного водіння.		
	Ви маєте слідувати за електронними помічниками: рулити, гальмувати або прискорюватись, якщо потрібно.			якщо система буде цього вимагати, необхідно керувати самостійно.	Електронні системи не вимагають від водія керувати автомобілем самостійно.	
	Системи допомоги водію			Системи безпілотного водіння		
Як вони працюють?	Помічники лише попереджують і надають короткотривалу допомогу.	Помічники допомагають керувати АБО прискорюватись/ гальмувати.	Помічники допомагають керувати ТА прискорюватись/ гальмувати.	Система може самостійно керувати автомобілем лише за умови одночасного виконання кількох умов.		Система може керувати автомобілем за будь-яких умов.
Приклади систем	Автоматичне аварійне гальмування Попередження про сліпі зони Попередження про покидання смуги	Утримання на смузі АБО Адаптивний круїз-контроль	Утримання на смузі ТА Адаптивний круїз-контроль одночасно	Помічник під час руху у дорожніх заторах.	Місцеве безпілотне таксі. Може бути відсутній руль або педаль.	Те саме, що і рівень 4, але можливість автоматизованого пересування зберігається

Рисунок 1.17 – Рівні автоматизації водіння за SAE J3016 [30]

Зазначені у рис. 1.17 рівні автоматизації мають свої особливості.

Рівень 0. Без автоматизації водіння. Перший рівень повне ручне керування. Людина забезпечує виконання динамічного завдання водіння.

Рівень 1. Допомога водієві. Найнижчий рівень автоматизації включає єдину автоматизовану систему допомоги водієві – адаптивний круїз-контроль.

Рівень 2. Часткова автоматизація водіння. Цей рівень передбачає наявність у машини елементів ADAS. Прикладом рівня 2 є такі розробки, як: Bluecruise від Ford, Full Self-Driving від Tesla, а також Super Cruise від General Motors (GM) [31].

Рівень 3. Автоматизація умовного водіння. Починаючи з третього рівня й вище автомобілі самостійно слідкують за ситуацією на дорозі за допомогою лідара. Транспортні засоби рівня 3 мають можливості «зчитування» (зондування) навколишнього середовища, що дозволяє їм отримати необхідну інформацію для ухвалення обґрунтованих рішень щодо подальших дій, в т.ч. уникання перешкод. Прикладом є Audi A8L 2019 р., який став першим у світі серійний автомобіль рівня 3 [32].

Рівень 4. Висока автоматизація водіння. Цей рівень не вимагає участі людини в управлінні автомобілем, оскільки він запрограмований на самостійну зупинку в разі збою системи. У конструкції транспортного засобу рівня 4 може не бути керма та педалей. Технологія автоматизації водіння рівня 4 призначена для використання в безпілотних таксі та службах громадського транспорту. У 2021 році Hyundai Motor Group запустила пілотний сервіс автономного водіння 4-го рівня на моделі автомобля марки Hyundai Ioniq 5 [33].

Рівень 5. Повна автоматизація водіння. Цей рівень – останній крок в автономності, якій усуває динамічне завдання водіння. У транспортних засобах рівня 5 не буде керма, педалей прискорення чи гальма. Ці робомобілі будуть всепогодними, необмеженими певними географічними кордонами і не потребуватимуть втручання людини. Як приклад, кілька років тому Apple запустила проєкт Titan зі створення SDC рівня 5. Але, як стало відомо наприкінці 2022 р., компанія скоротила амбітні плани та відклала прогнозовану дату запуску приблизно щонайменше до 2026 р., оскільки реалізація проєкту, за словами розробників, зіткнулася з реальністю: бачення повністю автономного транспортного засобу не може бути здійснено сучасними технологіями [34].

					ІА94.150БАК.004 ПЗ	Арк.
						21
Зм.	Арк.	№ докум.	Підпис	Дат		

1.4 Прототипування на базі Arduino

Як зазначено у роботах [35-37], Arduino - це електронна платформа з відкритим вихідним кодом, заснована на простому у використанні апаратному та програмному забезпеченні. Плати Arduino можуть зчитувати вхідні дані - світло на датчику, палець на кнопці або повідомлення в Твіттері – і перетворювати їх на вихідні дані – активувати двигун, включати світлодіод, публікувати будь що в Інтернеті. Для цього, використовуючи мову програмування Arduino (на основі Wiring) та програмне забезпечення Arduino. Процес прототипування з Arduino доволі простий і складається з кількох етапів (незалежно від варіанта або типу проекту) [36]:

1. Ідея (визначення) проблеми. На цьому етапі народжується ідея, втілення якої потребує реалізації проекту та створення прототипу.

2. Концептуалізація. Після того, як виникла ідея та визначено проблеми, які потрібно вирішити для її реалізації, будується концепція. Це спрощений вид проекту, який дозволяє відповісти на низку питань: Як саме це має працювати? Чи будуть необхідні обчислення для отримання остаточного результату? Який алгоритм реалізації проекту? та інші.

3. Прототипування. Прототип трансформує ідею (теорію) у специфікації реальної працюючої системи у вигляді створення зразка готового продукту. Сформульовані ідеї втілюються в апаратно-програмному продукті, які стають так званими альфа-класами, оскільки вони є першою версією проекту, який неможливо запустити доти, доки вони не будуть модифіковані для включення всіх необхідних функцій. Після включення всіх функцій прототип стає бета-версією, готового до тестування та подальшого розгортання для широкого використання. У робототехніці Arduino в основному використовується для керування двигунами, датчиками та виконавчими механізмами.

Щоб краще зрозуміти, яку роль Arduino може відігравати у роботизованих системах, розглянемо переваги та обмеження [36].

					ІА94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		22

1. Низька вартість та недорога платформа. Безпосередньо плати Arduino відносно економічні, особливо в порівнянні з іншими платформами мікроконтролерів, такими як Raspberry Pi та Nanode, логічно, що більшість новаторів купують компоненти саме Arduino та збирають свої конструкції Standard DIY (Do It Yourself), тобто «Зроби сам». Найдешевший модуль Arduino, який можна зібрати вдома, коштує менше 50 дол.

2. Кросплатформенність. Arduino IDE може працювати у більшості операційних систем, включаючи MacOS, Windows та Linux, що є ключовою відмінністю від більшості інших платформ мікроконтролерів, котрі працюють лише у Windows.

3. Енергоощадність. Arduino вимагає небагато енергії, адже працює з найнижчою напругою, заощаджуючи витрати користувача.

4. Швидкий процес прототипування. Повне прототипування системи Arduino відбувається достатньо швидко, тоді як на альтернативних обчислювальних платформах у більшості випадків потрібно набагато більше часу для створення прототипу пристрою аналогічного типу, що й на базі Arduino.

5. Просте та зрозуміле середовище програмування. Arduino IDE – це середовище програмування платформи, яке набуло поширення через цього простоту використання та швидке освоєння (яке не вимагає великої практики навіть для новачків). Водночас це не спрощення чи примітивізація, а просунуті інструменти, котрі допомагають професіоналам виконувати більш складні функції. Оскільки середовище розробки Arduino IDE було побудовано на певних функціях середовища програмування Processing, робота з Arduino IDE досить не складна для тих користувачів, хто вже знає мову Processing, що, наприклад, полегшує роботу викладачів, котрі навчають студентів, які вже вивчали програмування в Processing.

6. Відкритий вихідний код і програмне забезпечення, що розширюється. Arduino IDE та супутнє програмне забезпечення мають повністю відкритий вихідний код. Це полегшує роботу досвідченим програмістам з додавання розширень та модифікацій, які зроблять платформу кращою. Мова програмування Arduino також багато в чому запозичена з C++ і може бути розширена з бібліотеки

					IA94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		23

C++; технічні деталі мови також скопійовані з мови програмування AVR C, тому він приймає коди AVR-C, ніби вони були написані в Arduino IDE.

7. Впровадження більш інноваційних технологій. Коли Arduino прийшла на платформу мікроконтролерів, деякі інші платформи вже існували, але завдяки впровадженню інноваційних технологій, таких як висока швидкість обробки, Arduino змогла швидко стати домінуючою платформою та найкращою для професіоналів.

8. Простота інтерфейсу. Arduino є простим в інтерфейсі, адже має розширювані контакти, до яких можна підключити зовнішні модулі (такі як USB), що полегшує передачу ресурсів; також є доступним широкий спектр API-інтерфейсів або інтерфейсів прикладних програм.

9. Широкий набір датчиків. Arduino поставляється з кількома датчиками, що надає йому переваги при виборі користувачів інструментів для прототипування систем, які включають той чи інший тип візуалізації.

10. Величезна спільнота користувачів та ентузіастів-новаторів. Arduino об'єднав поціновувачів робототехніки, які обмінюються досвідом, ноу-хау, демонструють свої проекти, дають рекомендації.

11. Підходить навіть для початківців (школярів та студентів). Більшість функцій Arduino можна освоїти, переглянувши відеоролики в Інтернеті або поставивши запитання в онлайн-спільноті Arduino. Завдяки портативній платформі новачки можуть набути досвід, отримати нові навички та компетенції.

Водночас, поряд із зазначеними вище перевагами Arduino має низку обмежень [36].

1. Обчислювальна потужність. плати Arduino мають обмежену обчислювальну потужність проти іншими мікроконтролерами чи комп'ютерними системами. Це може обмежити здатність плати виконувати складні завдання та обчислення, особливо у додатках реального часу.

2. Пам'ять. Плати Arduino також мають обмежений обсяг пам'яті як оперативної, так і флеш-пам'яті. Це може бути обмеженням для проектів, яким

					ІА94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		24

потрібний великий обсяг сховища даних, або для додатків, які потребують багато пам'яті для операцій.

3. Вартість. Попри те, що плати Arduino відносно дешеві, вартість додаткових компонентів, таких як датчики, приводи та пристрої зв'язку, може швидко зрости і зробити загальну вартість проекту вищою.

4. Швидкість. Тактова частота плат Arduino обмежена, що може вплинути на продуктивність деяких програм реального часу, таких як високошвидкісний збір даних або керування двигуном.

Попри зазначені вище обмеження плати Arduino нині широко використовуються винахідниками та розробниками інноваційних продуктів для реалізації різних ідей та вирішення проблем у різних сферах діяльності, спектр яких настільки ж різноманітний, наскільки безмежні сфери застосування прототипів, створених за допомогою Arduino. Аналіз літератури показав, що Arduino – популярний вибір для прототипування рішень у таких сферах, як: проектування систем; додатки загального призначення; апаратний зв'язок; прототипування програмного забезпечення; домашня автоматизація та загальна автоматизація; сільське господарство; охорона здоров'я; гірничодобувна промисловість; енергія та захист навколишнього середовища; освіта тощо (більш детально подано автором у роботі [36]). Зокрема, як показало дослідження, широке застосування Arduino знайшло у створення прототипів автономних транспортних засобів (табл. 2 складено на базі [36]).

Для вдосконалення інструментарію притотипування мобільних роботів Arduino створило низку інновацій у 2022 р, зокрема Arduino IDE 2.0 (версія 2.0 перейшла з бета-версії на стабільну та внесла значні зміни), яка дозволяє розпочати з потужного сучасного редактора та зосередитися на специфічних для Arduino функціях.

Також нова IDE безпосередньо інтегрується з Arduino Cloud, дозволяючи користувачам продовжити роботу з місця, де вони зупинилися, у веб-редакторі на основі браузера. Нова IDE заснована на потужному інтерфейсі командного рядка Arduino, що значно спрощує базову бібліотеку. Також було створено новий

					ІА94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		25

інтерфейс командного рядка Arduino Cloud. Оскільки пристрої Інтернету речей (IoT) вимагають керування в масштабі, щоб їх ініціалізувати, отримувати оновлення та надсилати дані, новий Arduino створив свій інструмент Arduino Cloud CLI.

Таблиця 2 – Приклади моделей роботизованих автомобілів на базі Arduino

Продукт	Дослідження
	Мініатюрна модель автономного автомобіля з використанням Arduino UNO та Open CV
	Розробка роботизованої системи уникнення перешкод на основі Arduino для безпілотного автомобіля
	Розробка автономного автомобіля на основі рукавичок Arduino
	Роботизований автомобіль із голосовим керуванням на базі Arduino
	Платформа Robotic-agent для вбудовування програмних агентів з використанням плат Raspberry Pi та Arduino

Аналізуючи феномен популярності Arduino серед розробників, слід згадати аргументи, наведені ще у 2011 р. Філіпом Торроном [36]: ця платформа має великі перспективи, оскільки це відмінне програмне забезпечення для декількох систем, безліч бібліотек, драйвери, що працюють, простий, недорогий і з відкритим вихідним кодом [36].

1.5 Постановка задачі

Як показав аналіз предметної області дослідження, розробка системи навігації самохідної техніки ускладнена низкою технологічних проблем, серед яких створення програмного та апаратного забезпечення. Оскільки SDC мають не лише «зчитувати» інформацію з навколишнього середовища, але й розуміти, що вона собою являє, необхідним є створення алгоритмів, аби SDC могли миттєво вирішувати, що робити у навколишньому середовищі, яке має численні перешкоди. Оскільки не існує базових або складних алгоритмів, здатних охопити всі можливі ситуації на дорозі, кожна індивідуальна подія вимагає розробки відповідного алгоритму та реалізації його на базі певно платформи.

Створення інноваційних дослідних зразків нових самохідних машин потребує значного фінансування та вимагає багато робочого часу дослідників та експериментаторів. З огляду на це, є необхідність створення прототипу SDC – мобільного робота, який: уникає перешкоди, не стикаючись з оточуючими об'єктами; має можливість виявляти перешкоди на своєму шляху на основі заданої граничної відстані; після виявлення перешкоди може змінювати свій курс, автономно ухвалюючи рішення; не вимагає зовнішнього керування під час реалізації його функцій; зданий реагувати у режимі реального часу; діяти у невідомому середовищі.

Такого мобільного робота, що уникає перешкоди, можна буде використовувати практично у всіх системах навігації мобільних роботів, як приклад: для домашньої автоматизації (зокрема автоматизованого прибирання пилососом); використовувати у небезпечних умовах, де проникнення людини неможливе чи небезпечне тощо.

В основу цієї розробки доцільно покласти методологію алгоритмічного та програмного забезпечення основних компонентів системи навігації мобільного робота, яка базуватиметься на Arduino, що дозволить швидко та з мінімальними фінансовими витратами створити робочий прототип та здійснити експериментальні

					ІА94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		27

дослідження. Це дозволить зробити оцінку обраного підходу та створить підґрунтя для його вдосконалення та проведення подальших досліджень.

Висновки розділу 1

Узагальнюючи понятійний апарат, що міститься у наукових працях з автоматизації та роботизації транспортних засобів, можна сказати, що найбільш вживаними нині термінами для позначення самохідної техніки, яка пересувається без людини-водія, є: «автономний автомобіль», «безпілотний автомобіль», «самокерований автомобіль», «робот-автомобіль». Попри розповсюдженість цих термінів у літературі, у реальності, беручи до уваги визначені SAE рівнів автоматизації, автономних транспортних засобів рівня 5 немає. Представлені лише концептуальні розробки, що знаходяться на початковій стадії опрацювання, здебільшого концептуальні моделі. Машина рівня 4 нині являють собою поодинокі зразки, які проходять тестування. Автомобільні компанії намагаються забезпечити своїм виробам рівень 3 – першу сходинку автономного водіння. Очевидно, що у найближчій перспективі проекти, що матимуть шанс на широке впровадження, будуть зосереджуватися здебільшого на рівні 2 та його «просунутих» версіях.

Аналіз архітектури SDC показав, що їх техніко-технологічну основу формують великі складні системи, оснащені низкою датчиків (як для внутрішнього, так і зовнішнього моніторингу), генерації величезних обсягів даних за короткі проміжки часу. Функціональні блоки реалізуються на основі потоку інформації та етапів обробки, що реалізуються від збору даних (зондування) до управління транспортним засобом. Технологічні та функціональні блоки SDC є базою системи навігації SDC, котра являє собою здатність машини планувати свій маршрут і виконувати його без втручання людини, вирішуючи дві проблеми: знати, як рухатися до мети, і мати можливість уникати будь-яких перешкод на цьому шляху, спираючись на відповідні апаратні та програмні засоби, включаючи, розробку навігаційного алгоритму.

					ІА94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		28

2 АНАЛІЗ ОСНОВНИХ СПОСОБІВ НАВІГАЦІЇ САМОХІДНОЇ ТЕХНІКИ З УХИЛЕННЯМ ВІД ПЕРЕШКОД

2.1 Автономна навігація

У забезпеченні автономності пересування SDC чи мобільного робота система навігації спирається на низку специфічних засобів (пов'язаних із апаратними та програмними компонентами), що обумовлюють особливості їх динаміки і кінематики. Відтак, друге важливе питання, котре потребує розгляду для досягнення мети дослідження, – це автономна навігація, що зазвичай інтерпретується як здатність транспортного засобу планувати свій шлях та долати його без втручання людини. У деяких випадках у процесі планування використовуються дистанційні навігаційні засоби, у той час як в інших – єдиною інформацією, доступною для розрахунку шляху, є вхідні дані від датчиків на борту транспортного засобу [38-40].

Українські вчені зазначають: «Під навігацією розуміється теорія і практика спрямування за заданою траєкторією рухомих об'єктів» [41]. При цьому мета навігації полягає в знаходженні оптимальних (відповідно до заданих критеріїв) маршрутів переміщення між заданими точками простору з урахуванням стаціонарних (пасивних) та рухомих (активних) перешкод [41].

У робототехніці виділяють низку навігаційних схем, які поділяють на три категорії [9]:

- глобальна – визначення абсолютних координат пристрою під час руху довгими маршрутами;
- локальна – визначення координат пристрою по відношенню до деякої (зазвичай стартової) точки (ця схема затребувана розробниками наземних роботів, які виконують місії в межах наперед відомої області);
- персональна – позиціонування роботом частин свого «тіла» та взаємодія з прилеглими предметами, що актуально для пристроїв, забезпечених маніпуляторами.

					ІА94.150БАК.004 ПЗ	Арк.
						29
Зм.	Арк.	№ докум.	Підпис	Дат		

Як приклад локальної навігаційної системи на рис. 2.1 подано структурну схему, розроблену Рудиком А. В.

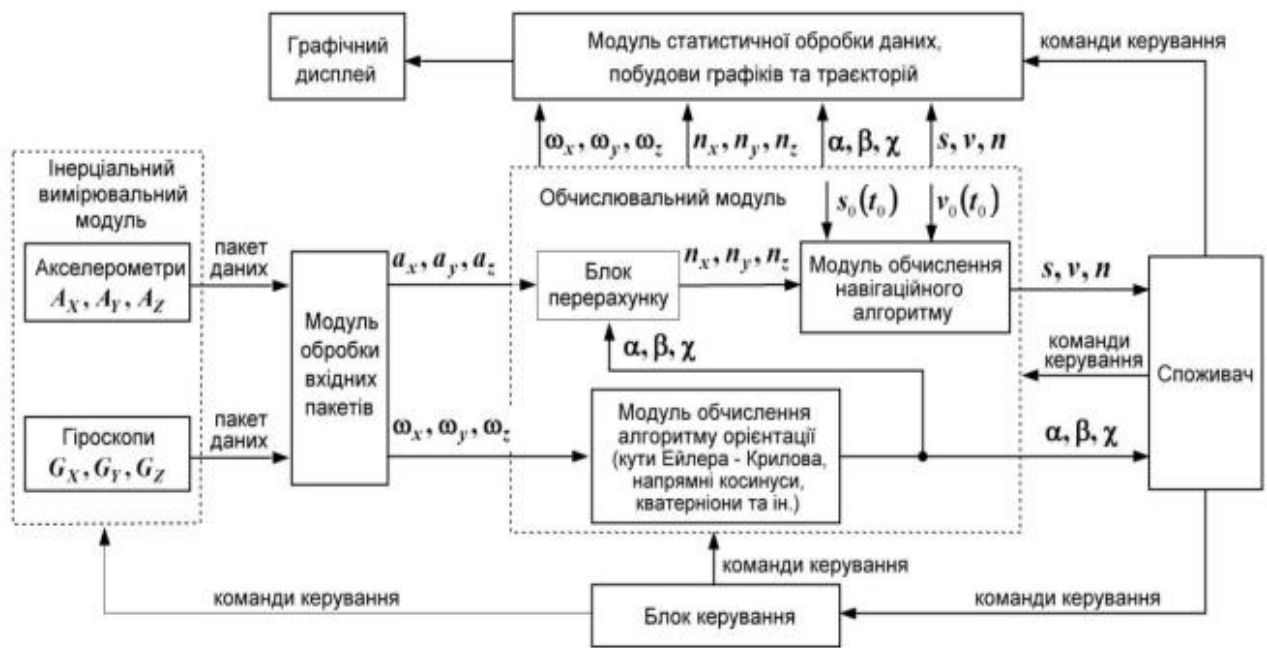


Рисунок 2.1 - Структурна схема локальної навігаційної системи [9, с. 341]

Системи навігації класифікуються ще за однією ознакою – вони можуть бути пасивними та активними. Пасивна система навігації передбачає прийом інформації про власні координати та інші характеристики свого руху від зовнішніх джерел, тоді як активна розрахована на визначення місця розташування лише самотужки – вона посилає сигнал і сприймає його відбиття. Як правило, всі глобальні схеми навігації пасивні, локальні бувають обох типів, персональні – завжди активні.

Здатність до навігації – головна особливість, яка вирізняє автономних мобільних роботів. «Загально визнано, що навігація – це методологія, яка дозволяє направляти робота в навколишньому середовищі, використовуючи інформацію датчиків», – зазначено у роботі [42, с. 77].

Є чотири компетенції, пов'язані зі здатністю автономної навігації: 1) побудова карти, 2) самолокалізація, 3) інтерпретація карти та 4) планування шляху. Компетенція побудови карти належить до репрезентації довкілля. Самолокалізація відноситься до здатності робота знати своє положення на карті. Нарешті,

інтерпретація карти стосується здатності робота використовувати карту для навігаційних завдань, таких як планування шляху. Обчислення шляху та навігація на основі орієнтирів – найпоширеніші методи навігації автономної машини.

На думку автора дослідження [43, с. 351], для забезпечення автономії у навігації найбільш широко використовують модульну структуру для сприйняття та обробки інформації. Складний механізм пересування транспортним засобом реалізується у рамках так званої *автономної навігаційної системи* (autonomous navigation system) – системи, здатної планувати свій шлях і виконувати свій план без втручання людини. Це комбінація складних систем, що допомагає приймати рішення з урахуванням зондування навколишніх ситуацій. Технології автономної навігаційної системи спираються на такі передові розробки: інерційна навігаційна система, супутникова навігаційна система, радары, камери, ультразвукова та акустична навігація, а також з використанням автономних навігаційних алгоритмів для точної та безпечної навігації транспортних засобів.

У роботі [44] система навігації роботом інтерпретується як «інтелектуальна система управління (ІСУ)», головне завдання якої полягає у фізичній реалізації руху машини та відповідного реагування на неконтрольовані зміни навколишнього середовища, що вимагає вирішення низки взаємодіючих підзадач. Бачення їх реалізації та специфіку керування роботом (де навігація є підсистемою) подано у вигляді тривірневої ієрархічної структури (рис. 2.2).

Відповідно до [45, с. 89], у системі управління інтелектуальними транспортними засобами система контролю ухилення від перешкод є однією з основних фундаментальних тем дослідженні у галузі автономної навігації, оскільки підтримання безпечної відстані від інших транспортних засобів та перешкод має вирішальне значення як для фізичної безпеки системи та навколишнього середовища, так і для прийняття рішень з контролю машини.

У процесі дослідження системи навігації самохідної техніки окрему увагу було приділено аналізу літератури з питання створення алгоритмів долаття перешкод [46-48].

Автор [49] узагальнив архітектурні рішення навігації SDC у статичному (структурованому) середовищі, поділяючи їх на 3 рівні: 1) стратегічний, 2) тактичний, 3) виконавчий, де на перших двох – розв’язують проблеми виявлення різних перешкод у навколишньому середовищі машини та забезпечення навігації; на третьому – відбувається втручання у процес навігації. Головним бар’єром у функціонуванні систем в динамічних середовищах є вимога ухвалювати рішення при наявності певних обмежень (часових та обчислювальних). Процес ускладнюється тим, що третій (виконавчий) рівень не володіє необхідними даними для ухвалення рішення, а перші два рівня не здатні вчасно приймати рішення.

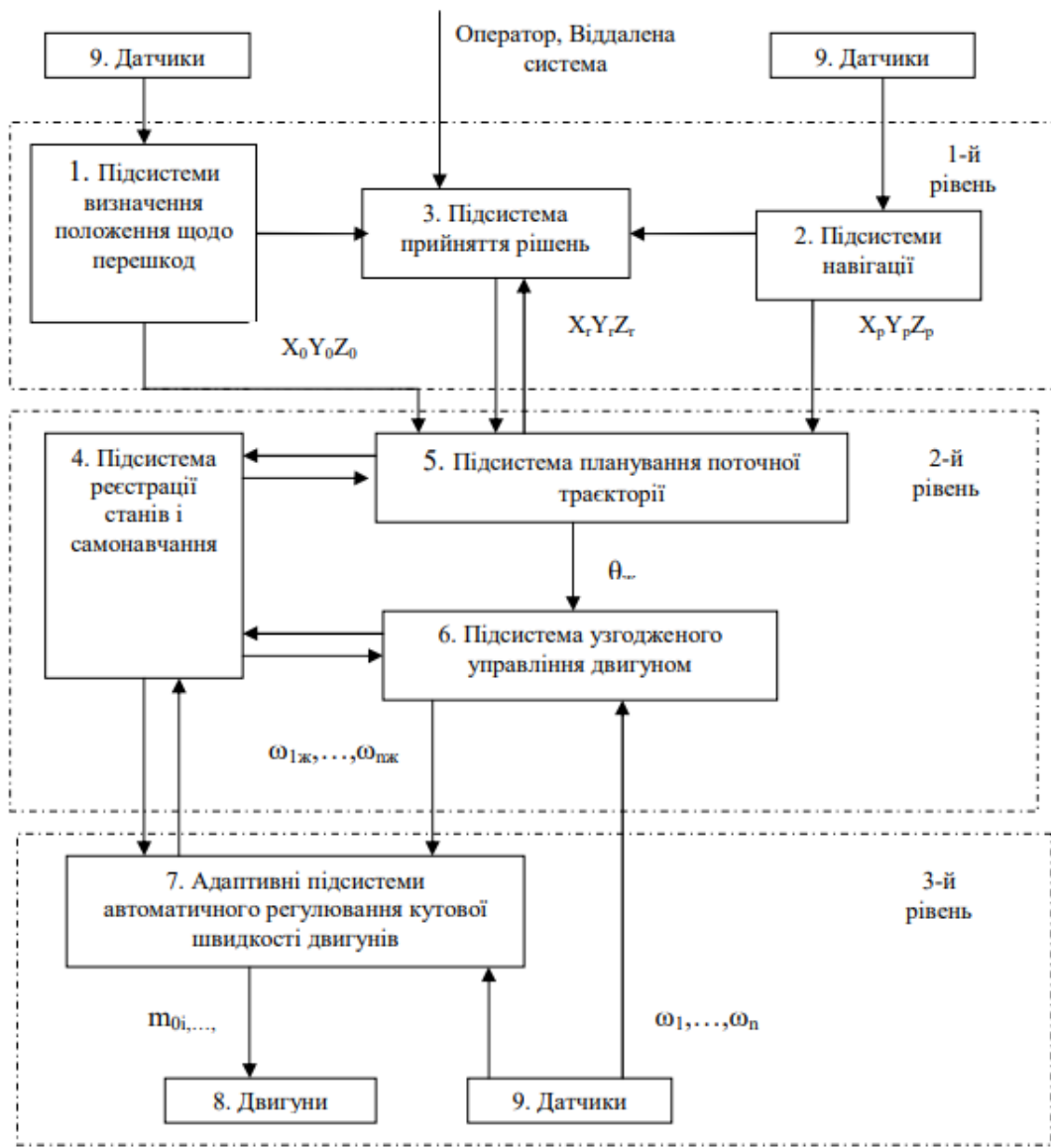


Рисунок 2.2 – Інтелектуальна система управління мобільним роботом [44]

Зм.	Арк.	№ докум.	Підпис	Дат

Для побудови концептуальної архітектури системи навігації SDC використовують декомпозицію функціональних елементів системи як-от: сприйняття, планування, виконання. Така трирівнева архітектура обумовлена методами керування мобільним роботом [50]:

- 1) реактивні методи керування, що приводять до відображення результатів сприйняття сенсорів безпосередньо на керування приводами без чи з невеликим зовнішнім впливом;
- 2) методи для виконання шаблонних послідовностей діяльності, яка екстенсивно залежить від внутрішнього стану апаратного забезпечення, причому пошуку оптимального рішення при цьому не проводиться;
- 3) складні обчислювальні методи (відносно рівнів зміни навколишнього середовища) для прийняття оптимальних рішень, наприклад, планувальники руху.

Беручи до уваги зазначені вище методи, автори роботи вказують [51], що таке архітектурне рішення має низку недоліків – в ньому не передбачено процеси оброблення сенсорної інформації, навчання та побудови локальної та глобальної карт середовища. О.П. Адамів аргументує, що переважна більшість відомих доробків стосується статичного і завчасно відомого середовища, і тому є необхідність у створенні оптимальної архітектури (з урахуванням задач і потоків даних, що виникають при пересуванні робота у неструктурованому динамічному середовищі, та взаємозв'язків її структурних елементів) [50] (рис. 2.3). Детальне пояснення запропонованої О.П. Адамовим архітектури подано у роботі [50].

Як зазначають японські вчені [52], система навігації реального SDC здійснюється через використання таких методів: зчитування (зондування), розпізнавання, прийняття рішень, контроль (рис. 2.4).

Зчитування (зондування) означає використання стереокамер, радарів, сонарів, лідарів та інших датчиків виявлення об'єктів.

Розпізнавання спирається на такі процеси, як: «злиття датчиків» (комбінування сенсорної інформації від різних датчик, що зондують середовище, як-от: транспортні засоби, пішоходів, знаки, дорожню розмітку тощо); точна ідентифікація розташування SDC; «злиття карт» (злиття з картою об'єктів,

					ІА94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		33

ідентифікованих шляхом зондування); «прогнозування руху об'єктів» (прогнозування поведінки об'єктів довкола SDC); і «формування динамічної карти» (створення шляху, просторові карти та карти ризиків, що відображають цю інформацію у вигляді даних, які можуть використовуватися для прийняття рішень та контролю).

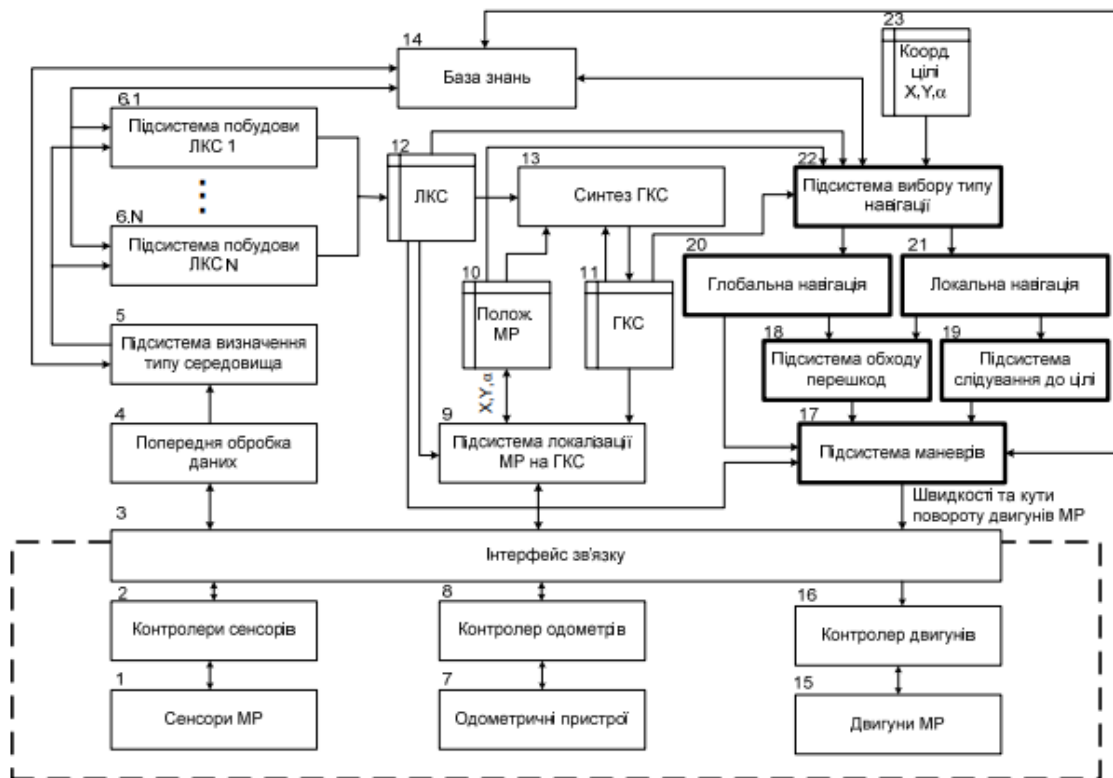
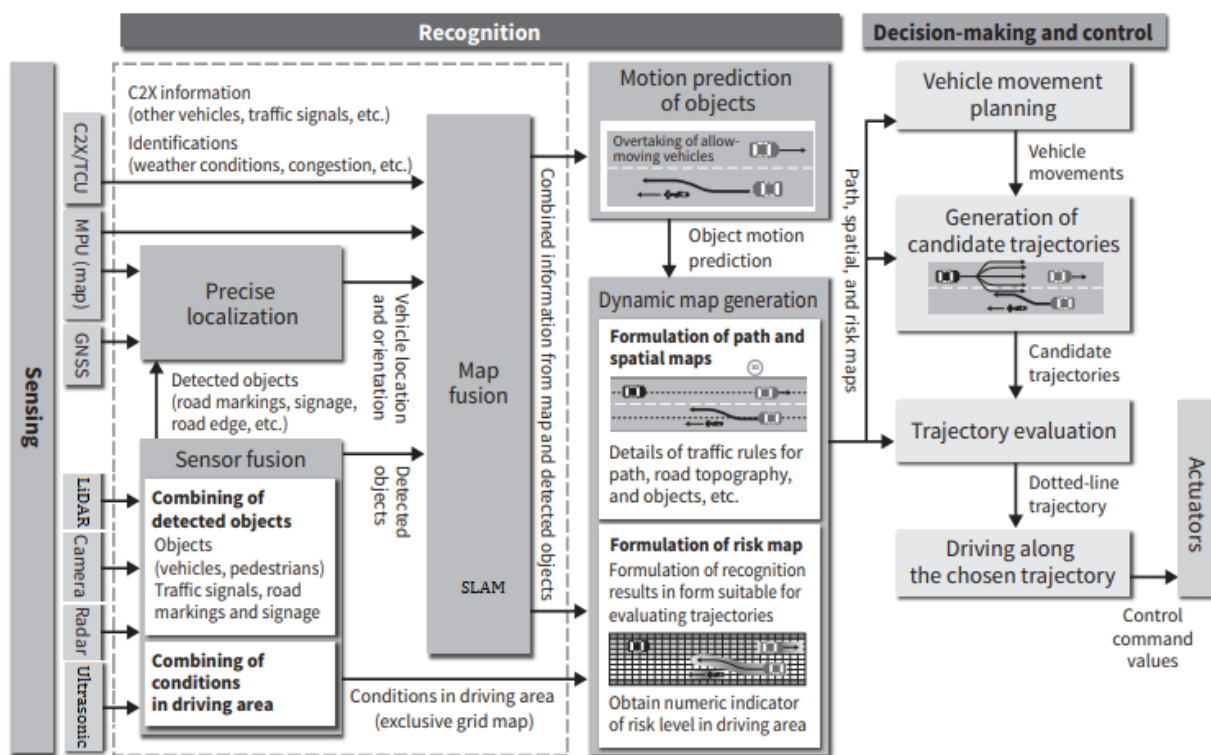


Рисунок 2.3 – Архітектура системи навігації у середовищі з перешкодами [50]

Процес прийняття рішень та контролю включає планування руху SDC, визначення можливих траєкторій, оцінку траєкторії та навігація SDC обраною траєкторією. Процес планування руху регулює стан всіх аспектів автономного керування та забезпечує загальний рух SDC. Формування можливих траєкторій проводиться з урахуванням таких чинників як динамічні характеристики SDC: оцінюючи траєкторії, беруться до уваги майбутні ризики під час вибору найкращої з можливих траєкторій. Навігація SDC з вибраної траєкторії здійснюється шляхом розрахунку значень команди управління, які мають бути відправлені на виконавчі механізми – актуатори (руль, педаль газу, гальма, двигун).

Традиційні навігаційні системи обробляють дані від датчиків за допомогою кількох модулів, налаштованих для таких завдань, як локалізація, картографування, виявлення об'єктів, планування руху та керування кермовим керуванням. Навігація має вирішальне значення в таких областях: створення та відстеження карт, планування маршруту, виявлення та запобігання перешкодам. Сенсорні системи лежать в основі здатності транспортних засобів бути автономними та забезпечувати точну локалізацію та планування шляху.



C2X: car to X TCU: telematics communication unit MPU: map position unit GNSS: global navigation satellite system

Рисунок 2.4 – Блок-схема системи навігації SDC [52].

2.2 Алгоритми навігації

У навігації мобільних роботів використовуються різні методи [53], які забезпечують уникнення перешкод, що є однією з фундаментальних проблем мобільної робототехніки. Ці методи поділяють на три групи (виходячи з базового алгоритму):

- детермінований алгоритм,
- недетермінований (стохастичний) алгоритм,
- еволюційний алгоритм (отриманий гібридизацією зазначених вище двох алгоритмів).

Рисунок 2.5 демонструє загальну класифікацію алгоритмів, які були реалізованими у системах навігації мобільних роботів різними авторами. Загальна класифікація детермінованого алгоритму, недетермінованого (стохастичного) алгоритму та еволюційного алгоритму, які використовуються для навігації мобільного робота.

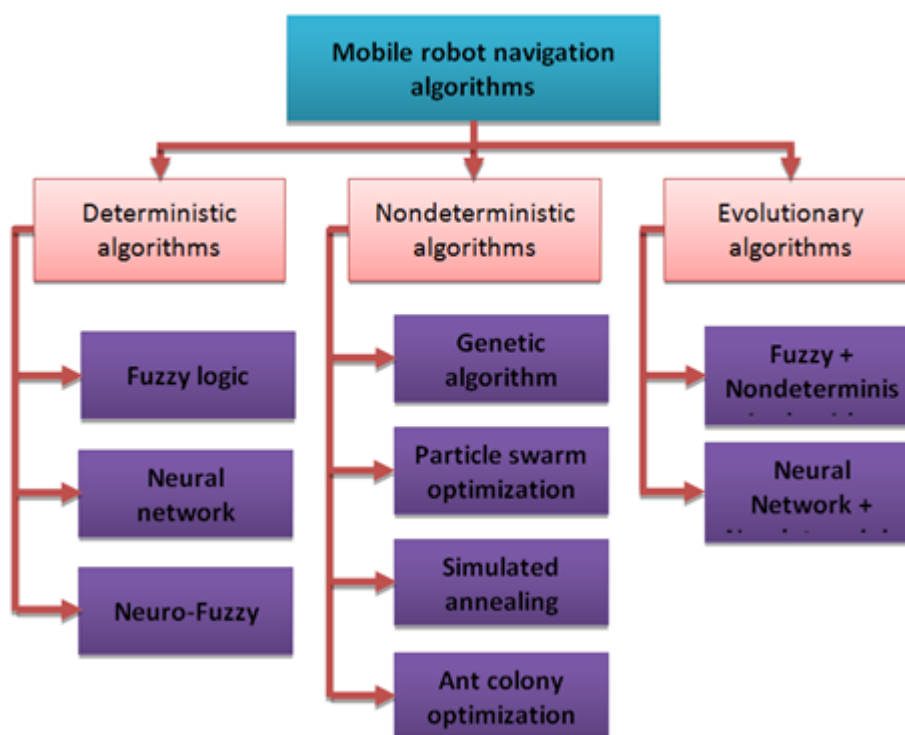


Рисунок 2.5 – Алгоритми навігації мобільних роботів [53]

Зокрема, за останні кілька років було запропоновано багато методів програмного обчислення для вирішення проблеми навігації роботів і уникнення перешкод у різних статичних і динамічних середовищах, серед них – метод нечіткої логіки (Fuzzy logic) для мобільної навігації роботом. Концепція нечіткої логіки була розроблена Заде (Zadeh) і є широко вживана у багатьох інженерних додатках, з поміж яких – мобільна робототехніка. Техніка нечіткої логіки була

успішно застосована багатьма дослідниками для контролю положення та орієнтації мобільного робота в навколишньому середовищі. Існуючі праці описують інтелектуальний контролер нечіткої логіки для вирішення проблеми навігації колісного мобільного робота в невідомому та мінливому середовищі; демонструють оптимальний нечіткий контролер Такагі-Сугено на основі градієнтного методу для налаштування параметрів функції приналежності та застосовують його для навігації мобільних роботів та уникнення перешкод; подають нечітку архітектуру на основі поведінки для навігації мобільних роботів у невідомих середовищах, спираючись на базові моделі поведінки: прагнення до мети, поведінку уникнення перешкод, поведінку відстеження тощо. для навігації мобільних роботів, випробувавши їх у різних симуляційних середовищах. Інші дослідники [54] об'єднали нечіткий генетичний алгоритм для вирішення проблеми планування шляху та керування автономним мобільним роботом, використовуючи інформацію датчика ультразвукового далекоміра (автори подали конфігурацію ультразвукових датчиків; побудували динамічну модель та кінетичні рівняння для мобільного робота; відповідно до визначених перешкод, надали поведінку та правила уникнення, запропонували правила вибору та уникнення перешкод та блок-схему навігації робота у середовищі з кількома перешкодами).

2.3 Алгоритми планування маршруту

Планування маршруту для самокерованих транспортних засобів полягає в пошуку геометричної інтерпретації його маршруту від початкової позиції до цілі, використовуючи зазвичай неповну карту перешкод, щоб уникнути зіткнення з цими перешкодами.

Нині розроблено низку алгоритмів для уникнення перешкод і планування маршруту, починаючи від простої зупинки автомобіля при виявленні перешкоди і закінчуючи зміною адаптивної поведінки автомобіля в залежності від типу, розмірів та інших особливостей виявленого блоку. Ці алгоритми відрізняються

					ІА94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		37

кількістю даних, необхідних для обробки, кількістю датчиків, операційною швидкістю, просторовою складністю, ефективністю та стратегією керування.

У роботі [55], детально викладено типові алгоритми, що використовуються в системах керування безпілотним транспортним засобом. Досліджені алгоритми можна розділити на такі групи: алгоритми помилок; наївні, або найпростіші алгоритми; алгоритми з використанням датчиків відстані; потенційні польові алгоритми; алгоритми на основі графів; формальні алгоритми (алгоритм Дейкстри, алгоритм Флойда-Воршала); евристичні алгоритми (наприклад, пошук по ширині); гібридні алгоритми пошуку (наприклад, A^* , D^*).

Більш детально зупинимось на алгоритмах знаходження шляху до мети з урахуванням заздалегідь невідомих перешкод – сімействі алгоритмів, що реалізують зазначений підхід, отримало назву алгоритмів знаходження шляху в русі [56] (алгоритми BUG). Вони набули поширення та мають велику кількість модифікацій: Alg1, Al2, DistBug, Class1, Rev1, Rev2, OneBug, LeaveBug та TangentBug та ін. Їхня популярність обумовлена простотою та дешевизною необхідного обладнання. Розглянемо окремі алгоритми цього сімейства.

Оригінальний алгоритм BUG, а точніше дві його модифікації Bug1 (рис. 2.6 – рис. 2.8) та Bug2 (рис. 2.9 – рис. 2.11) були запропоновані В. Люмельські та О. Степановим у 1987 р. Суть алгоритму Bug1 полягає в тому, що мобільний робот рухається безпосередньо до мети до тих пір, поки не зустрине перешкоду. Далі роботом досліджуються контури перешкоди до того часу, поки шлях руху до мети знову буде доступний. Як тільки датчик зіткнення фіксує наявність перешкоди, робот починає рух уздовж межі перешкоди проти годинникової стрілки для визначення точки виходу. Точкою виходу є найближча до мети точка межі перешкоди. Після досягнення точки виходу робот продовжує рух у напрямку мети доки не досягне її або не зустрінеться нова перешкода. У другому випадку повторюється процедура, описана вище. Метод має низьку ефективність, але гарантує досягнення роботом будь-якої доступної точки призначення.

					ІА94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		38

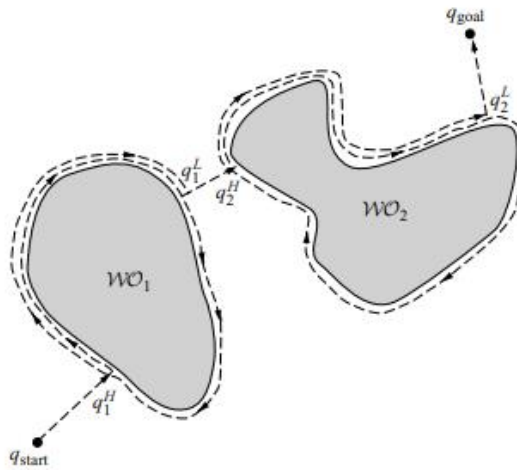


Рисунок 2.6 – Алгоритм Bug1 успішно знаходить мету

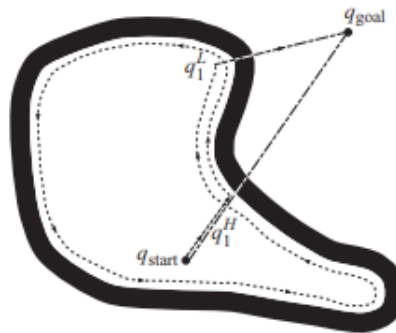


Рисунок 2.7 – Алгоритм Bug1 повідомляє, що мета недосяжна

Algorithm 1 Bug1 Algorithm

Input: A point robot with a tactile sensor

Output: A path to the q_{goal} or a conclusion no such path exists

- 1: **while** Forever **do**
- 2: **repeat**
- 3: From q_{i-1}^L , move toward q_{goal} .
- 4: **until** q_{goal} is reached **or** an obstacle is encountered at q_i^H .
- 5: **if** Goal is reached **then**
- 6: Exit.
- 7: **end if**
- 8: **repeat**
- 9: Follow the obstacle boundary.
- 10: **until** q_{goal} is reached **or** q_i^H is re-encountered.
- 11: Determine the point q_i^L on the perimeter that has the shortest distance to the goal.
- 12: Go to q_i^L .
- 13: **if** the robot were to move toward the goal **then**
- 14: Conclude q_{goal} is not reachable and exit.
- 15: **end if**
- 16: **end while**

Рисунок 2.8 – Алгоритм Bug1

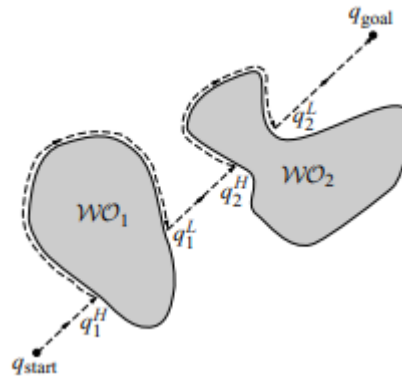


Рисунок 2.9 – Алгоритм Bug2 успішно знаходить мету

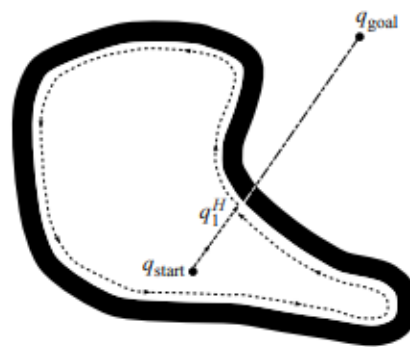


Рисунок 2.10 – Алгоритм Bug 2 повідомляє, що мета недосяжна

Algorithm 2 Bug2 Algorithm

Input: A point robot with a tactile sensor

Output: A path to q_{goal} or a conclusion no such path exists

- 1: **while** True **do**
- 2: **repeat**
- 3: From q_{i-1}^L , move toward q_{goal} along m -line.
- 4: **until**
 q_{goal} is reached **or**
an obstacle is encountered at *hit point* q_i^H .
- 5: Turn left (or right).
- 6: **repeat**
- 7: Follow boundary
- 8: **until**
 q_{goal} is reached **or**
 q_i^H is re-encountered **or**
 m -line is re-encountered at a point m such that
 $m \neq q_i^H$ (robot did not reach the hit point),
 $d(m, q_{goal}) < d(m, q_i^H)$ (robot is closer), and
If robot moves toward goal, it would not hit the obstacle
- 15: Let $q_{i+1}^L = m$
- 16: Increment i
- 17: **end while**

Рисунок 2.11 – Алгоритм Bug 2

Кожен із алгоритмів Bug1 і Bug2 реалізує досить ясну і просту ідею, і кожен має переваги та недоліки. Наприклад, алгоритм Bug1 ніколи не створює локальних циклів, але для пошуку точки виходу завжди буде досліджено весь периметр перешкоди. Алгоритм Bug2, у свою чергу, більш «інтуїтивний» і використовує простіші рішення, проте може виявитися менш ефективним у складних випадках.

Алгоритм Dist-Bug відрізняється від класичного Bug1 визначенням точки виходу, а саме тим, що при русі по перешкоді відстань до цілі в кожній точці пройденого шляху розраховується одночасно, а точка виходу – точка з мінімумом розрахункової відстані. DH-Bug має в своїй структурі «дорадчий» (deliberative) і «адаптивний» (reactive) шари. Дорадчий шар відповідає за створення попередньої оцінки шляху на основі неповної інформації, а адаптивний – за реакцію системи на невизначеність. Для скорочення довжини шляху DH-Bug дозволяє роботів рухатися в кожному з можливих напрямків, не більше 90° від цілі. Ця тактика унеможливує проходженню перешкоди вздовж її кордонів у класичному алгоритмі Bug2, що є основною причиною довгої генерації треків. І якщо рух по кордонах вже почався, він припиниться раніше. Алгоритм DH-Bug також дозволяє вирішити задачу обходу рухомих перешкод.

Оригінальний алгоритм Bug передбачає, що датчик (сенсор) може виявляти тільки зіткнення з перешкодами. Однак мобільний пристрій може бути оснащений більш інформативним датчиком, таким як далекомір. За допомогою цієї додаткової інформації можна робити спроби створити більш ефективні алгоритми побудови шляху до цілі. Алгоритм TangentBug є результатом однієї з перших таких спроб. TangentBug використовує додаткову структуру даних – так званий граф локальних дотичних (LTG) до меж перешкод, «видимих» далекоміром, у даний момент. TangentBug, як і оригінальний Bug, використовує два варіанти руху мобільного пристрою: до цілі і вздовж перешкоди. На кожному кроці алгоритму (в кожен дискретний момент часу) робот конструює LTG, на основі якого планується наступний крок. Коли робот рухається до цілі, він рухається в локально оптимальному напрямку згідно LTG. Найбільш значущі модифікації алгоритму Bug, які суттєво підвищують його ефективність, були реалізовані в алгоритмі

					ІА94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		41

СВВВ. Логічна структура алгоритму враховує розмір перешкоди (максимальну ширину плоского тіла довільної форми) і вирішує задачу оптимізації згенерованого шляху. Всі Bug-алгоритми подано на рисю 2.12.

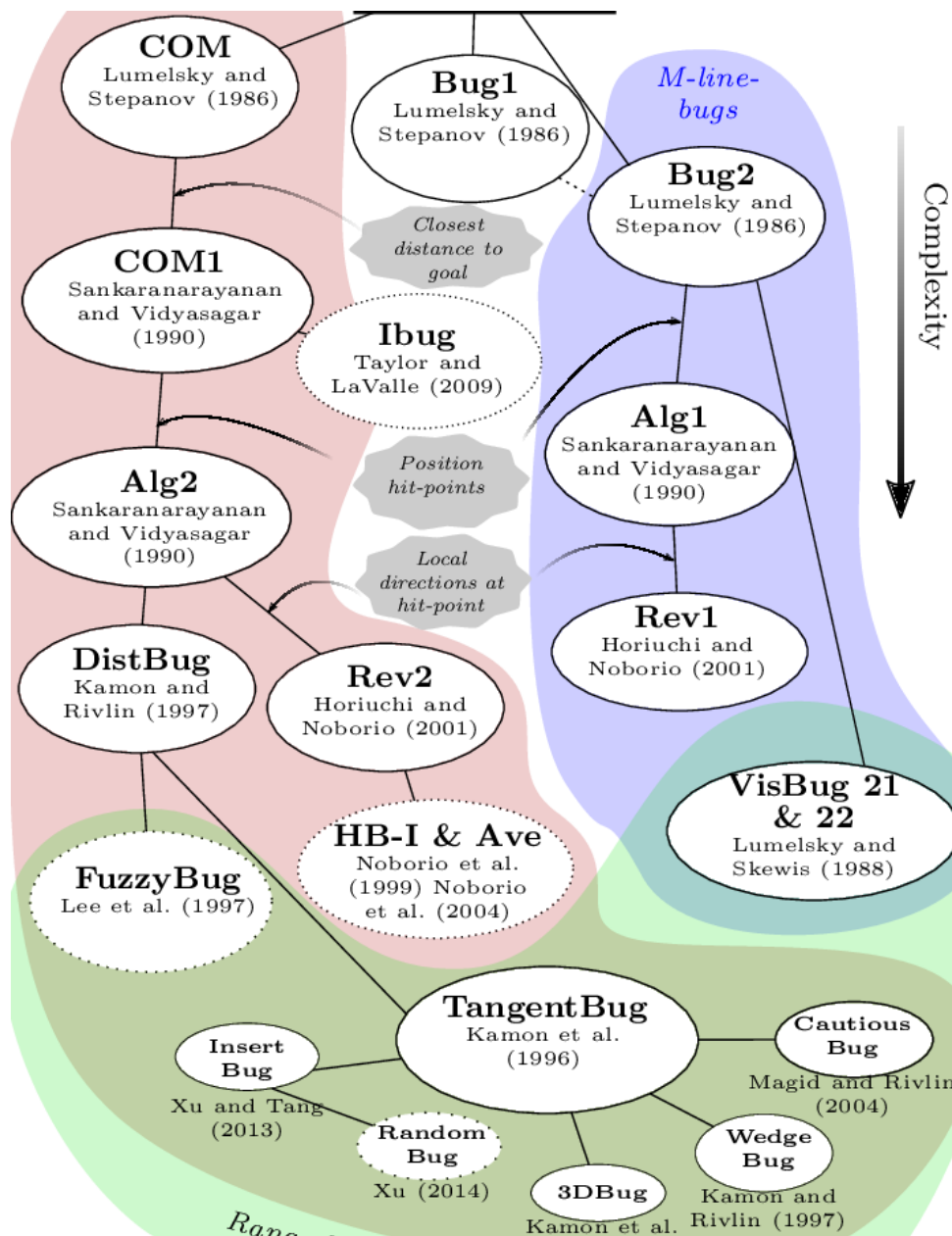


Рисунок 2.12 – Всі Bug-алгоритми [56]

Окрім алгоритмів Bug у роботі було досліджено інші алгоритми планування маршруту мобільних роботів, здатних оминати перешкоди. Їх порівняльний аналіз подано у табл.3

Таблиця 3 – Порівняння алгоритмів планування маршруту мобільних роботів, здатних оминати перешкоди

Алгоритм		Метод на основі датчика зору	“Follow the Gape” метод	Алгоритм гібридної навігації з прохідними стежками	Новий алгоритм гібридної навігації (NHNA)
Реалізація	необхідне обладнання	камера, гідроакустичний датчик(сонар), процесор, beagle board, ноутбук	ультразвукові, лідарні та датчики, оптичних швидкостей камери, npxi811108rt процесор rxi7954r fpga	лазерні та гідроакустичні датчики, роботи pioneer 3-at	лазерний датчик, мікропроцесор
	необхідні параметри	положення перешкод, кут і відстань	відстань і кут перешкоди	попередня інформація	попередня інформація
Продуктивність	ефективність	висока, розрахунки реальні і точні (залежить від обладнання)	висока, простий в налаштуванні, завжди вибирає найкоротший шлях, здатний уникнути симетричних перешкод	середня, генерує найкоротший шлях, але не обмежує відхилення від шляху	середній і більш ефективний, ніж hybrid, використовує dh-bug алгоритм
	конвергенція	може чи ні (залежить від характеру алгоритму)	немає (у тупиковому випадку, u-образних перешкод)	є, але в деяких сценаріях робот може зупинитися перед перешкодою	в основному сходиться, за винятком деяких сценаріїв, таких як тупик
	часова складність	залежить від роздільної здатності камери та додатка; займає більше часу для розрахунків	менше часу, оскільки рішення приймаються на основі поточного сприйняття	потребує більше часу для створення прикладу шляху (потрібен пошук a*)	витрата більше часу, оскільки пошук «а*» необхідний для створення шляху
Зауваження		не кращий варіант мініавто з мікроконтролером, вимагає ноутбука або процесора і конкретного застосунку, як matlab	не оминає u-образної перешкоди	вимагає складних розрахунків вимагає більше часу	складний розрахунок, витрачається більше часу

Алгоритм		Алгоритм Bug	Метод потенційного поля (VFF)	Гістограма векторного поля (VFH)
Реалізація	необхідне обладнання	датчики відстані (іч, гідролокатор), мікроконтролер	датчики відстані (іч, гідролокатор), мікроконтролер	сонарний датчик, процесор, великий обсяг пам'яті
	необхідні параметри	поточне та цільове положення	відстань цілі та перешкоди	відстань перешкоди
Продуктивність	ефективність	не дуже висока, може віддаляти робота від місця призначення	низька, розрахунок не точний, обмеження не враховуються	низька, розрахунок може бути точним, але споживає більше ресурсів, таких як пам'ять, процесор і енергія
	конвергенція	є, але потрібно більше часу	немає (у випадку u-образних і симетричних перешкод)	немає (у випадку u-образних і симетричних перешкод)
	часова складність	завжди рухається в одному напрямку, аби уникнути перешкоди, що збільшує часову складність	витрачається менше часу, завдяки вибору коротшого шляху	потрібно більше часу для створення 2d сітки і перетворення з 2d в 1d полярної гістограми
Зауваження	локальних мінімумів не відбувається; обертається найдовший шлях	локальний мінімум можливий	навантажує мікроконтролер, оскільки потрібні складні обчислення	

Висновки розділу 2

Наведені у літературі методи навігації та моделі керування SDC є узагальненням технічних рішень, які забезпечують SDC функціонування, його рух та досягнення цілі за певних умов. Виходячи з нових технічних завдань та вимог середовища, в якому має діяти SDC, є необхідність як вдосконалення, так і, в окремих випадках, створення принципово нових мобільних роботів (із власними

конструкційними особливостями) та розробки відповідного апаратного та програмного забезпечення їх навігації. Досліджені у роботі різні алгоритми обходу перешкод у деяких аспектах відрізняються один від одного. Наприклад, алгоритм Dist-Bug ефективніший, ніж метод потенційного поля, тому що він не має проблеми локального мінімуму. Він покриває невелику відстань у порівнянні з попередніми версіями, але може відводити робота від мети, тому що він не орієнтований на мету при прямованні за стіною. У той час як метод потенційного поля легко налаштувати, але він не є кращим, оскільки він призводить до мінімальної локальної помилки стека. Гістограма векторного поля - це алгоритм, що займає багато часу та місця, як і на 1-му кроці, він генерує 2D-гістограму, а потім перетворює її на 1D-гістограму для подальшого розрахунку. Більш детально розглянуті у роботі алгоритми Bug1 і Bug2, попри свою простоту у реалізації, мають низку обмежень, які полягають у наступному:

- кордони перешкод повинні являти собою замкнуті криві, такі, що пряма може перетнути кожен таку криву скінченне число разів;
- перешкоди не можуть торкатися одне одного, тобто точка, що належить одній перешкоді, не може так само належати будь-якій іншій перешкоді;
- мобільний робот розглядається як матеріальна точка, або, що еквівалентно, прохід між перешкодами вважається прохідним незалежно від ширини;
- датчики робота, крім факту контакту з перешкодою, також повинні надавати інформацію про поточні координати робота.

Більшість модифікованих версій базового алгоритму Bug спрямовані на подолання заданих обмежень ціною мінімізації складності алгоритму. Аналіз алгоритмів Bug показав, що окремі з них (як приклад TangentBug і DistBug) потребують датчиків діапазону, тоді як Bug1 і Bug2 – ні. В свою чергу Bug1 і Bug2 вимагають тактильних датчиків, але в окремих дослідженнях було використано ультразвукові для «зчитування» середовища і навігації в умовах невідомих перешкод. Отримані результати є підґрунтям, аби взяти за основу алгоритми обминання перешкод у розробці алгоритму для реалізації мети даного дослідження.

					ІА94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		45

3 ПРОЕКТУВАННЯ ТА СТВОРЕННЯ ПРОГРАМОВАНОГО РОБОТА, ЩО УНИКАЄ ПЕРЕШКОД, НА БАЗІ ARDUINO

3.1 Апаратні засоби проектування мобільного робота

Мікроконтролер Arduino виступає як управляючий мозок мобільного робота, забезпечує координацію рухів, сприймання даних з сенсорів і виконання заданої програми для досягнення бажаних функцій і завдань (рис. 3.1 – 3.2).

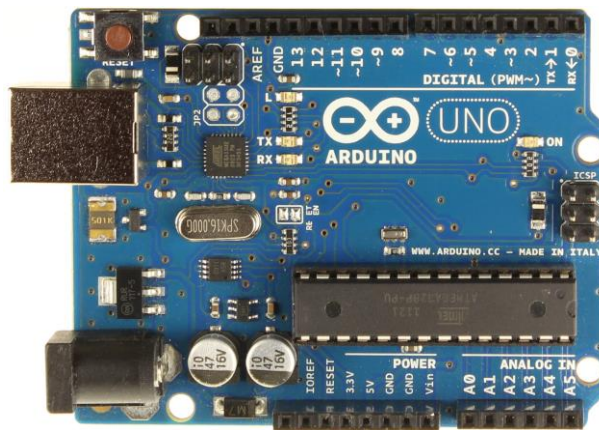


Рисунок 3.1 – Плата Arduino Uno

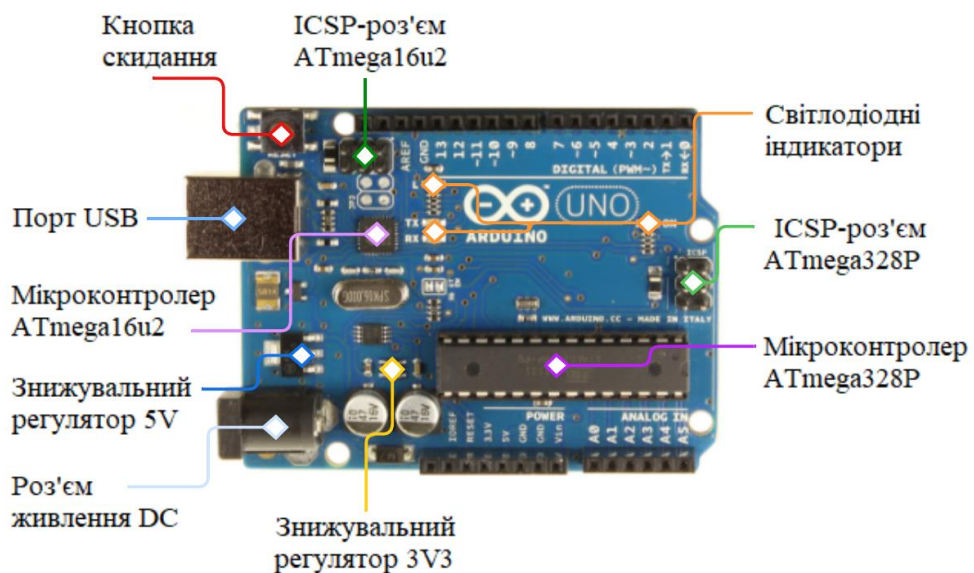


Рисунок 3.2 – Складові плати Arduino Uno

					IA94.150BAK.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		46

Характеристики Arduino Uno наступні.

Вхід/вихід: Arduino Uno має 14 цифрових вхідних/вихідних пінів. З них 6 можна налаштувати як ШІМ (Ширина Імпульсу Модуляції) виходи. Це дозволяє керувати різними пристроями, такими як світлодіоди, мотори, сервоприводи тощо.

Аналогові входи: Arduino Uno має 6 аналогових входів. Ці піни можуть вимірювати аналогові значення, наприклад, напругу або сигнали з датчиків.

Пам'ять: Arduino Uno має 32 кБ флеш-пам'яті, з яких 0,5 кБ використовується для зберігання програмного коду. Вона також має 2 кБ оперативної пам'яті (SRAM) і 1 кБ EEPROM для зберігання даних.

Інтерфейси: Arduino Uno має USB-порт для підключення до комп'ютера або інших пристроїв для програмування та зв'язку. Вона також має роз'єм ICSP (In-Circuit Serial Programming) для програмування мікроконтролера.

Живлення: Arduino Uno може бути живлена від зовнішнього джерела напруги в діапазоні від 7 до 12 вольт або від USB-порту. Вона має вбудований регулятор напруги, який забезпечує стабільну напругу 5 вольт для живлення мікроконтролера і підключених пристроїв.

Розміри: Arduino Uno має компактні розміри, що дорівнюють приблизно 68.6 мм x 53.4 мм. Це робить його зручним для використання в різних проектах.

Світлодіодна індикація:

- ON – індикатор живлення платформи.
- L – користувальницький світлодіод на 13 піні мікроконтролера. Використовує визначення LED_BUILTIN для роботи. При заданні значення високого рівня світлодіод вмикається, за низького – вимикається.
- RX, TX – блимають під час прошивки та обміну даними між Uno та комп'ютером. А також при використанні пінів 0 та 1.

Arduino Uno виконує ключову роль у цьому проекті як мікроконтролерна плата. Вона є центральним управляючим пристроєм, який керує роботом і забезпечує його функціональність.

У проекті використовується мікроконтролер ATmega328P, який знаходиться на платі Arduino Uno. Цей мікроконтролер має вбудований процесор і набір

					ІА94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		47

цифрових та аналогових входів/виходів, які можна програмувати для контролю різних пристроїв і сенсорів.

Arduino Uno забезпечує взаємодію з іншими компонентами робота, такими як мотори, сервоприводи, ультразвуковий сенсор і т.д. Завдяки своїм цифровим входам/виходам, вона може керувати рухом моторів, кутом обертання сервоприводів, отримувати відліки від ультразвукового сенсора, а також здійснювати інші операції залежно від програми, яка завантажена на мікроконтролер.

Arduino Uno програмується за допомогою мови Arduino, що дозволяє реалізовувати функціональність робота і взаємодію з додатковими компонентами.

Наступний компонент – Arduino Motor Shield L293D (рис. 3.3) має наступні характеристики.

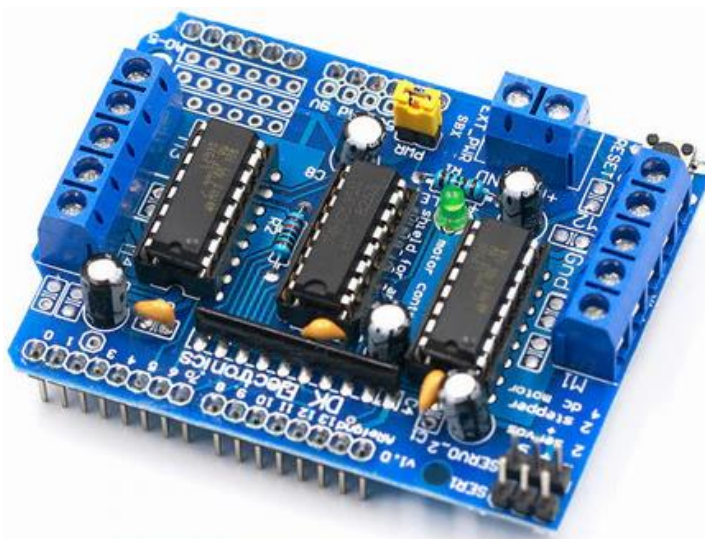


Рисунок 3.3 – Шилд драйверів двигунів на L293D

Інтегрована мікросхема L293D: двоканальний драйвер мотора, що забезпечує можливість керування двигунами постійного струму з максимальним струмом до 600 мА на кожний канал.

Підтримка двигунів постійного струму: шилд здатен керувати одночасно кількома двигунами постійного струму (від одного до чотирьох).

Напруга живлення: підтримує вхідну напругу живлення в діапазоні від 4.5 до 36 вольт.

Логічний рівень керування: 2 роз'єми для сервоприводів 5 В, підключених до спеціального таймера Arduino з високою роздільною здатністю

Індикатори стану: має LED-індикатори, які відображають стан каналів. Захист від перевантаження - Вбудований захист від перевантаження дозволяє запобігти пошкодженню шилда та підключених до нього двигунів.

Швидкість і напрямок керування: забезпечує можливість керування швидкістю та напрямком обертання двигунів з використанням PWM (ШИМ) сигналів.

L293D Based Arduino Motor Shield – це платформа, яка надає декілька переваг для керування двигунами постійного струму з використанням Arduino:

Однією з головних переваг цього шилда є його простота в використанні. Він має зручний інтерфейс, що дозволяє швидко підключити і керувати двигунами постійного струму без необхідності складних налаштувань.

Іншою перевагою є висока сумісність з Arduino платформою. L293D Based Arduino Motor Shield розроблений спеціально для використання з Arduino, що забезпечує його безпроблемну роботу з цією платформою. Це означає, що його можна інтегрувати в проекти без необхідності у додаткових модифікаціях.

Третя перевага полягає в його здатності керувати кількома двигунами одночасно. L293D Based Arduino Motor Shield має роз'єми для підключення до чотирьох двигунів постійного струму та два роз'єми сервоприводів, що дозволяє керувати багатьма моторами одночасно. Крім того, шилд надає можливість керувати швидкістю двигунів. Це надає змогу налаштувати швидкість обертання кожного двигуна окремо за допомогою аналогового сигналу, що дозволяє досягти більшої точності і контролю над рухом.

L293D Based Arduino Motor Shield має можливості розширення. Використання даного шилда дозволяє встановлювати додаткові модулі або датчики, щоб розширити його функціональність і використовувати в різних

					ІА94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		49

проектах. Це робить шилд багатофункціональним і гнучким інструментом для розробки робототехнічних проектів.

Узагальнюючи наведене вище, L293D Based Arduino Motor Shield є простим у використанні, сумісним з Arduino, здатним керувати декількома двигунами одночасно, підтримує керування швидкістю та має можливості розширення. Всі ці переваги роблять його відмінним вибором для розробки даного проекту, що включає керування двигунами постійного струму з використанням Arduino платформи.

Чіпсет драйвера. “Мозком” даного шилда є два драйвери двигунів L293D і регістр зсуву 74HC595 (рис. 3.4).

L293D — це двоканальний драйвер H-Bridge двигуна, який може керувати двома двигунами постійного струму або одним кроковим двигуном. Оскільки шилд містить два таких драйвери двигунів, він може керувати кількома двигунами: до чотирьох двигунів постійного струму або двох крокових двигунів.

Регістр зсуву 74HC595, з іншого боку, розширює чотири цифрові контакти Arduino до восьми контактів керування напрямком двох мікросхем L293D.

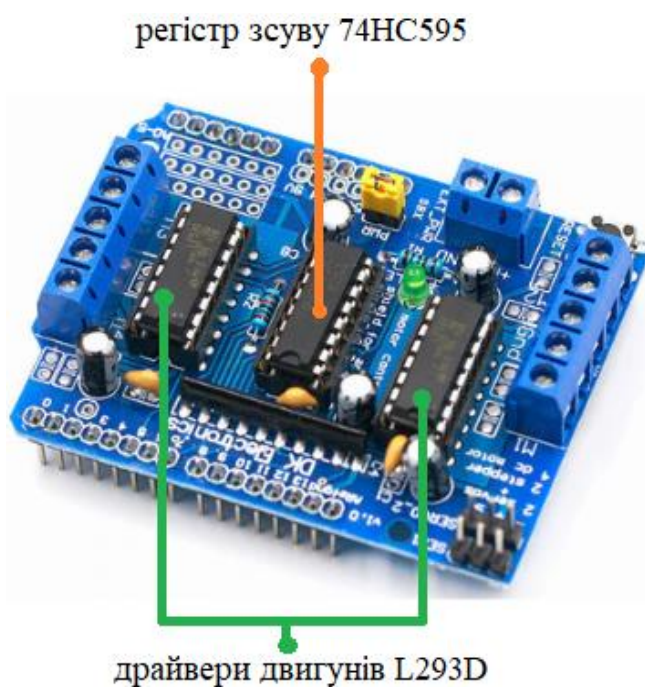


Рисунок 3.4 – Розміщення L293D драйверів двигунів і 74HC595 регістра зсуву

					IA94.150BAK.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		50

Підключення живлення двигуна. Шилд підтримує діапазон напруги двигунів від 4,5 В до 36 В. Цю потужність можна використовувати спільно з Arduino або окремо. Щоб вибрати між цими двома режимами, біля двоконтактного роз'єму живлення є спеціальна перемикач з позначкою PWR (рис. 3.5).



Рисунок 3.5 - Розміщення системи живлення шилда

Коли перемикач встановлена, живлення подається на двигуни через гніздо живлення постійного струму Arduino. У цьому випадку двигуни та Arduino фізично не ізольовані один від одного. Цей метод полегшує використання шилда, оскільки він вимагає лише одного джерела живлення. Однак слід зазначити що використовувати цей метод можна лише тоді, коли напруга живлення двигуна менше 12 В.

Коли перемикач видалена, живлення двигуна відключається від Arduino, що дозволяє фізично ізолювати двигуни від Arduino. Однак у цьому випадку необхідно забезпечити окреме джерело живлення двигуна для двопол'юсного роз'єму живлення, позначеного як EXT_PWR.

Підключення двигуна постійного струму. Вихідні канали обох мікросхем L293D розбиті на краях шилду за допомогою двох 5-контактних гвинтових клем

(рис. 3.6), позначених на платі як M1, M2, M3 і M4. До цих клем можна загалом підключити чотири двигуни постійного струму, що працюють від 4,5 до 25 В.

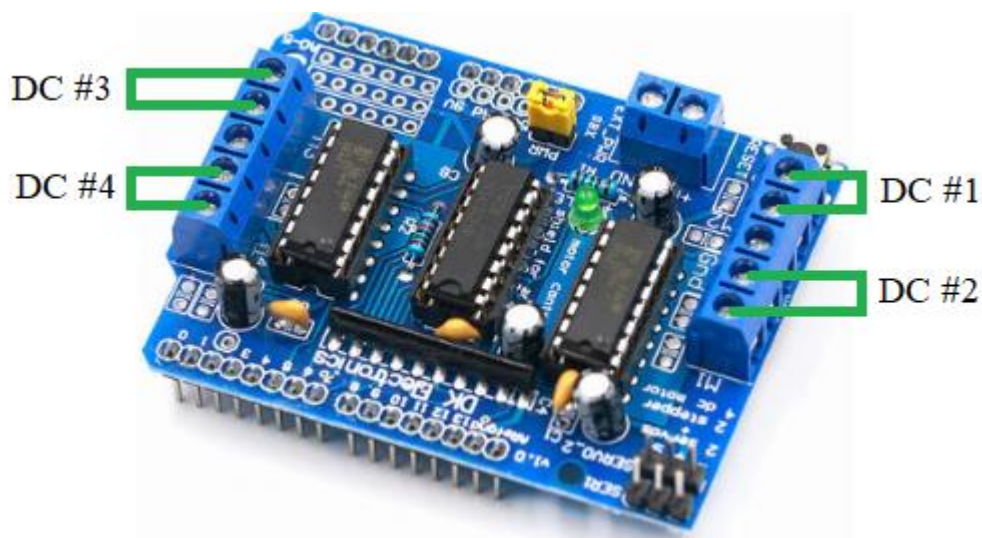


Рисунок 3.6 - Гвинтові клемки для підключення двигунів

Кожен канал на модулі може подавати струм до 600 мА (1,2 А пік) на двигун постійного струму. Величина струму, що подається на двигун, однак залежить від потужності джерела живлення двигуна.

Підключення сервоприводу відбувається наступним чином (рис. 3.7).

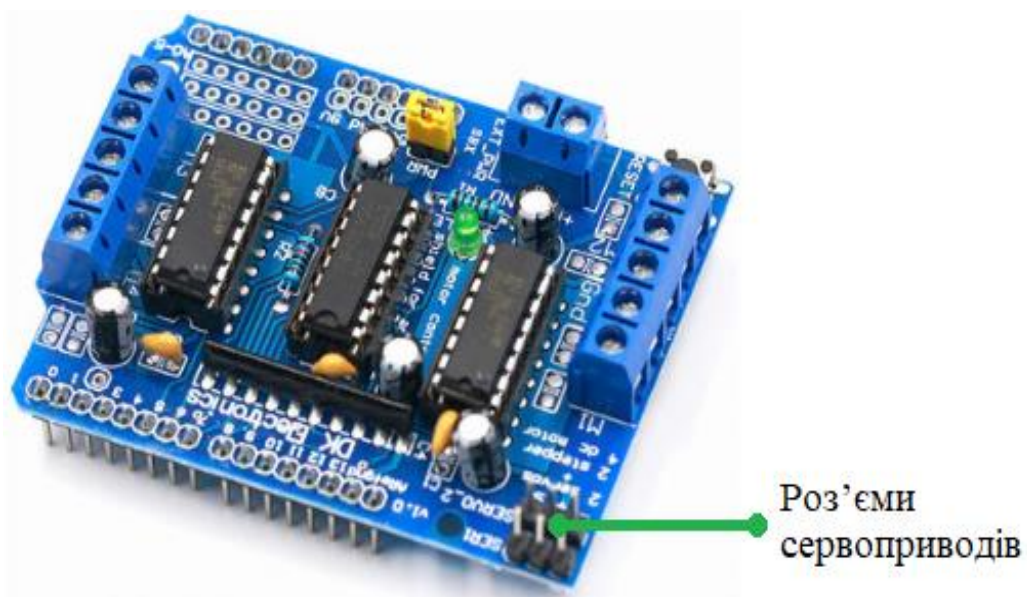


Рисунок 3.7 - Гвинтові клемки для підключення двигунів

Сервомотори живляться безпосередньо від джерела живлення 5 В Arduino, що, як правило, не є добре. Це може призвести до перегріву вбудованого регулятора 5 В Arduino, а також до електричного шуму в джерелі живлення 5 В. Позитивним моментом є конденсатор 100 мкФ на цих контактах живлення, що допомагає перешкоджати можливим проблемам. Отже, даний шилд краще використовувати лише з невеликими сервоприводами, такими як SG90. Загальна схема шилда подана на рис. 3.8.

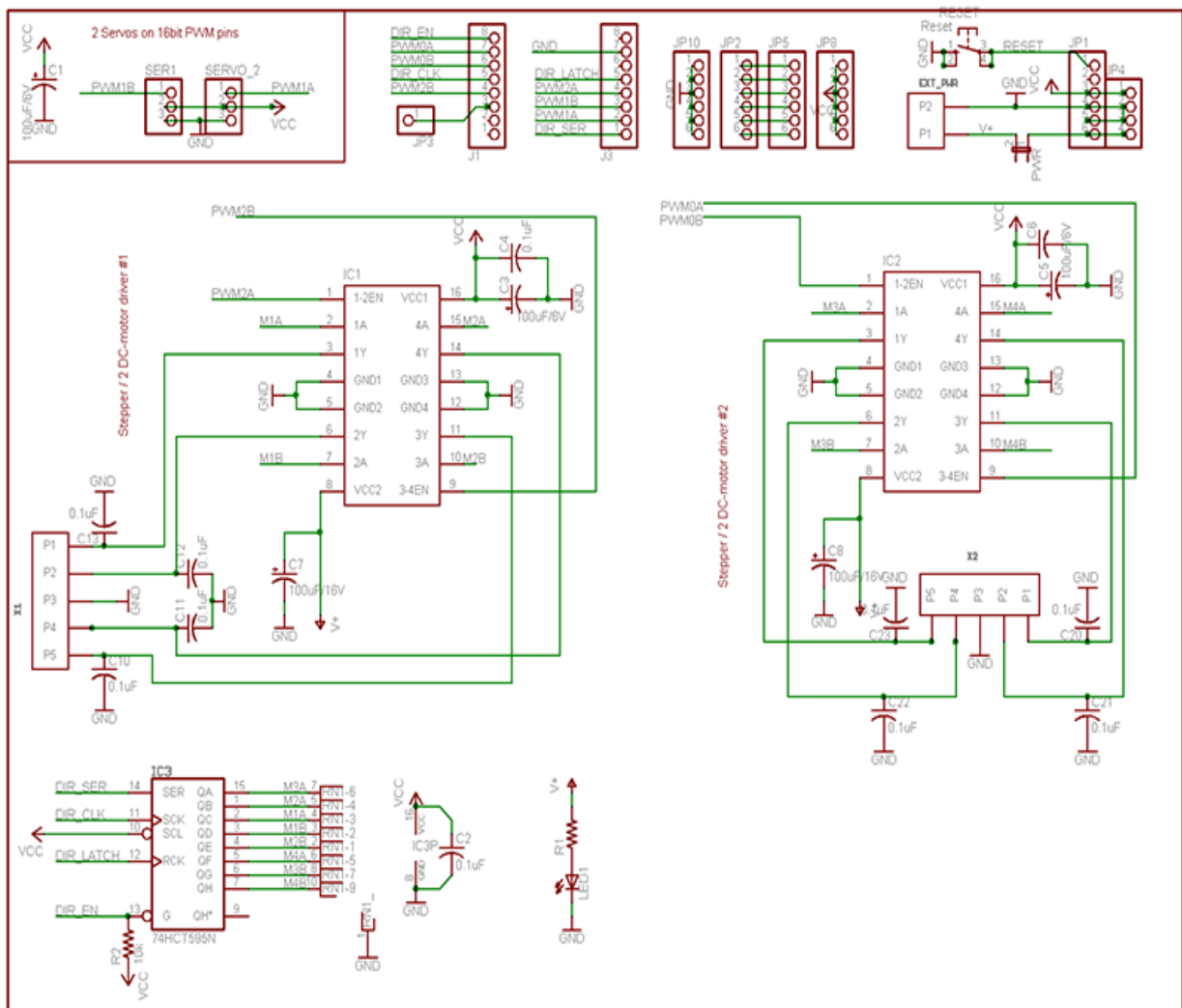


Рисунок 3.8 – Схематичне зображення шилда (Adafruit)

НС-SR04 Ультразвуковий сенсор (рис. 3.9) відіграє важливу роль у навігації і має наступні характеристики.

Робоча напруга: HC-SR04 працює з напругою 5 В. Це означає, що його можна безпечно підключити до Arduino або іншого контролера, який працює з такою ж напругою.

Дальність вимірювання: сенсор може вимірювати відстань до об'єктів у діапазоні від 2 сантиметрів до 400 сантиметрів. Це надає системі можливість вимірювати відстань від найближчих до віддалених об'єктів у вашому дослідженні.



Рисунок 3.9 - Ультразвуковий сенсор HC-SR04

Точність: HC-SR04 має високу точність вимірювання. Зазвичай, він має похибку приблизно ± 3 міліметри. Це дозволяє отримувати досить якісні результати вимірювання відстані.

Робоча частота: сенсор працює на ультразвуковій частоті 40 кГц. Він генерує ультразвукові хвилі для вимірювання відстані від об'єкта.

Роз'єми: HC-SR04 має чотири нижче зазначені піни:

- Vcc - цей пін забезпечує джерело живлення +5 В для датчика.
- Trigger - це вхідний пін, який використовується для ініціалізації вимірювання шляхом передачі ультразвукових хвиль протягом 10 мкс.
- Echo - це вихідний пін, який вмикається протягом певного періоду часу, що є еквівалентним часу, протягом якого хвиля повертається назад до датчика.
- Ground - це контакт GND, який використовується для підключення до системи GND.

					ІА94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		54

Інтерфейс: HC-SR04 використовує простий інтерфейс для взаємодії з контролером. Ви можете контролювати його за допомогою цифрових входів та виходів контролера.

У цьому проекті HC-SR04 (рис. 3.10) виконує важливу роль вимірювання відстані до об'єктів. Відправляючи ультразвуковий сигнал і отримуючи ехо-сигнал, HC-SR04 дозволяє точно визначити відстань до об'єкта за допомогою алгоритмів обробки сигналу.

Завдяки ультразвуковому сенсору робот здатен ефективно виявляти перешкоди перед собою, регулярно вимірюючи відстань за допомогою цього датчика. Якщо відстань до об'єкта менша за задане порогове значення, що було задано в коді, це свідчить про те, що переді ним є перешкода, і він має зупинитися, уникаючи зіткнення з об'єктами. Це гарантує безпеку робота. Крім того, використовуючи дані про відстань, отримані від HC-SR04, цей проект мобільного робота можна модифікувати, налаштовуючи під інші завдання. Крім використання даних для уникнення перешкод, робота можна вдосконалити, надаючи можливості автоматичної навігації або створення мапи оточуючого простору. HC-SR04 є незамінним компонентом цього проекту, який допомагає забезпечити безпеку та ефективну навігацію створеного у рамках цього проекту мобільного роботу завдяки його функціональності та надійності.

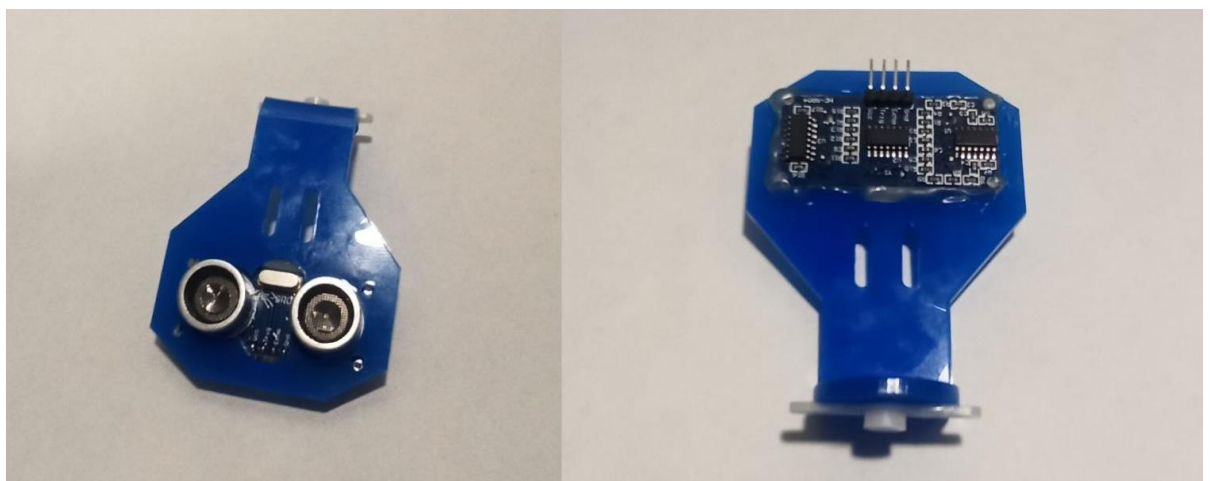


Рисунок 3.10 - Ультразвуковий датчик, закріплений на тримачі для подальшого закріплення на серводвигуні

Принцип роботи ультразвукового датчика відстані HC-SR04 наступні (рис.3.11 - 3.12). Процес виміру дальності починається з моменту, коли тригерний штифт встановлений у HIGH на 10 мкс. У відповідь датчик передає ультразвуковий сигнал з восьми імпульсів на частоті 40 кГц. Ця 8-імпульсна схема спеціально розроблена, щоб приймач міг відрізнити передані імпульси від навколишнього ультразвукового шуму. Ці вісім ультразвукових імпульсів поширюються від передавача. Тим часом ехо-штифт стає HIGH, щоб ініціювати зворотній ехо-сигнал. Якщо ці імпульси не відображаються назад, ехо-сигнал закінчується та стає LOW через 38 мілісекунд. Таким чином, імпульс тривалістю 38 мс свідчить про відсутність перешкод у радіусі дії датчика.

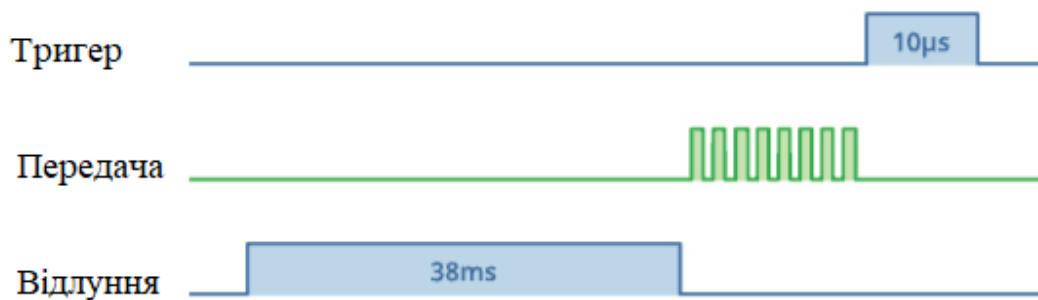


Рисунок 3.11 - Принцип роботи сигналу за відсутності перешкод

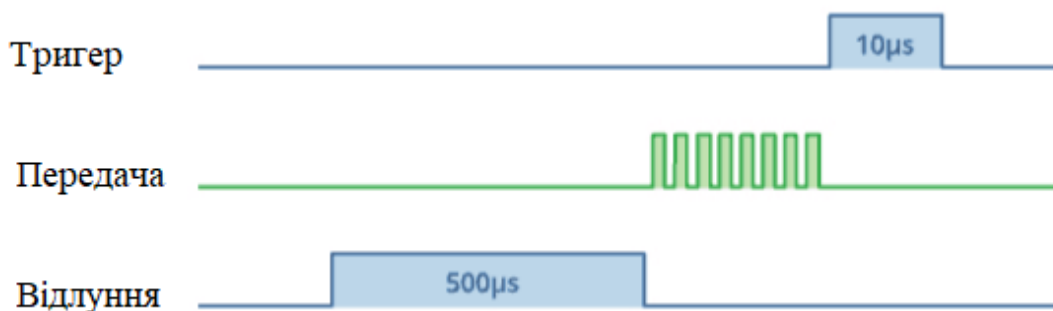


Рисунок 3.12 - Принцип роботи сигналу за наявності перешкод

Якщо ці імпульси відбиваються назад, ехо-штифт стає LOW, як тільки сигнал отримується. Це генерує імпульс на відлунні (Echo), діапазон якого

змінюється від 150 мкс до 25 мс залежно від часу, необхідного для отримання сигналу.

Розрахунок відстані. Ширина отриманого імпульсу використовується для обчислення відстані до відбитого об'єкта. Це можна розрахувати за допомогою рівняння відстані – швидкості – часу. Найпростіший спосіб зобразити рівняння – помістити літери в трикутник (рис. 3.13).



$$\text{Відстань} = \text{Швидкість} \times \text{Час}$$



$$\text{Час} = \text{Відстань} / \text{Швидкість}$$



$$\text{Швидкість} = \text{Відстань} / \text{Час}$$

Рисунок 3.13 – Графічне зображення розрахунку відстані

Візьмемо за приклад значення часу, тобто 500 мкс, і знаючи швидкість звуку - 340 м/с, обчислюємо відстань. Потрібно перетворити швидкість звуку в см/мкс. Це 0,034 см/мкс. Ця інформація надає змогу розрахувати відстань.

$$\text{Відстань} = 0,034 \text{ см/мкс} \times 500 \text{ мкс}$$

Проте це не кінцева відповідь, бо ехо-імпульс вказує час, який потрібен для відправлення та відбиття сигналу. Отже, щоб отримати відстань, доведеться розділити отриманий результат на два.

$$\text{Відстань} = (0,034 \text{ см/мкс} \times 500 \text{ мкс}) / 2$$

$$\text{Відстань} = 8,5 \text{ см}$$

Сервопривод SG90 (рис. 3.14). SG90 є малим та легким сервоприводом, який використовується для керування рухом механічних систем. Він є доступним, надійним та простим у використанні.

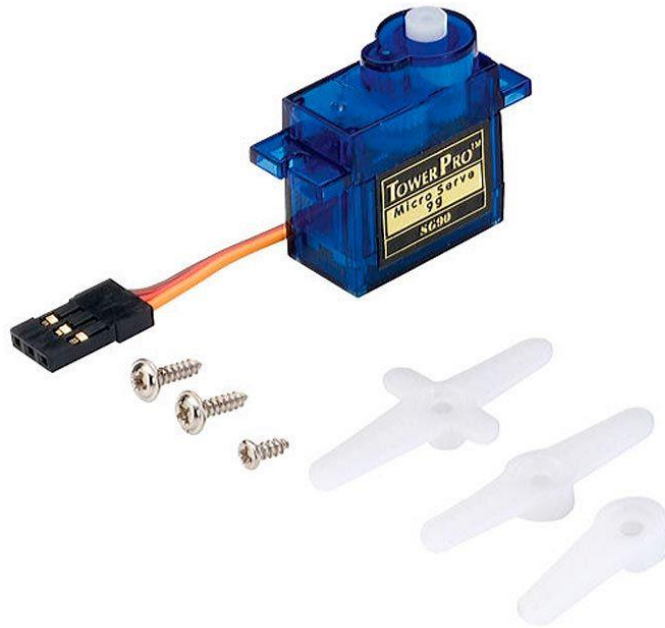


Рисунок 3.14 – Зовнішній вигляд сервоприводу SG90

Характеристики наступні.

Робочий діапазон: SG90 має обмежений робочий діапазон обертання, який зазвичай становить приблизно 180 градусів. Це означає, що він може повертатися від певного початкового положення до певного кінцевого положення.

Крутний момент: Враховуючи свою компактну конструкцію, SG90 має помірний крутний момент, здатний переміщати навантаження (як легкі, так і середні).

Напруга живлення: Зазвичай сервопривод SG90 працює в діапазоні напруги від 4,8 до 6 Вольт. Це означає, що він може бути живлений від батарей або блоків живлення з такою напругою.

Керування: SG90 приймає команди для керування положенням через ширину імпульсу PWM (ШИМ). Кут обертання визначається шириною імпульсу, який зазвичай становить від 1 до 2 мілісекунд. Зазвичай 1,5 мілісекунд відповідає нейтральному положенню (90 градусів), а ширина імпульсу більша або менша змінює кут обертання.

Розміри: SG90 має компактні розміри, зазвичай близько 23 мм x 12 мм x 29 мм. Це дозволяє легко використовувати його в різних проектах, де є обмежений простір.

Складові сервоприводу (рис. 3.15). Потенціометр забезпечує зворотний зв'язок положення з підсилювачем сигналу помилки у блоці керування, який порівнює поточне положення двигуна з цільовим положенням.

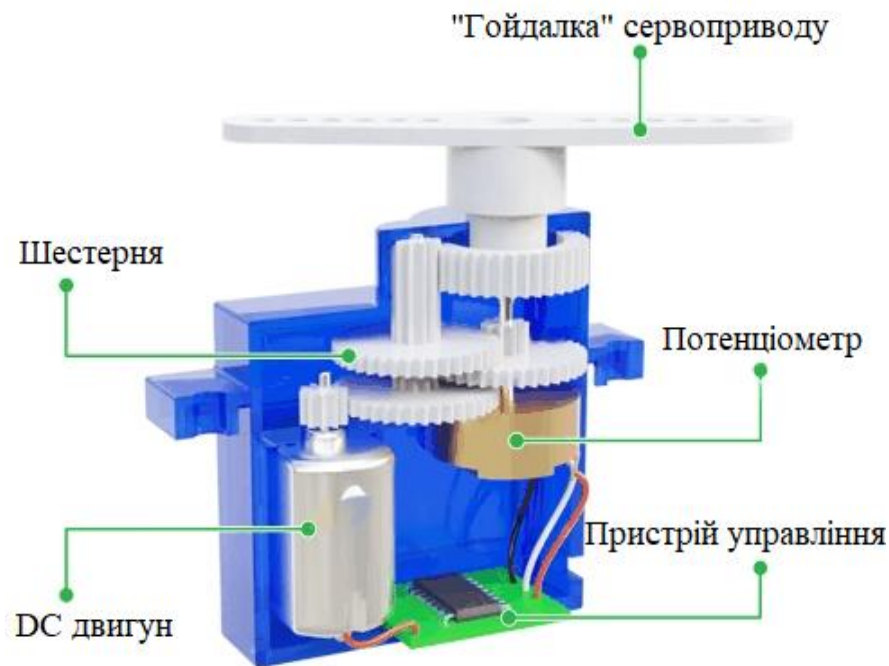


Рисунок 3.15 – Складові сервоприводу

У відповідь на помилку блок управління регулює поточне положення двигуна так, щоб воно відповідало бажаному положенню.

У техніці управління цей механізм відомий як сервомеханізм, або скорочено серво. Це замкнута система керування, яка використовує негативний зворотний зв'язок для регулювання швидкості та напрямку двигуна для досягнення бажаного результату. Сервоприводи зазвичай мають три з'єднання, як описано нижче (рис. 3.16).

- GND - служить спільним заземленням для двигуна та логіки.
- 5V - це позитивна напруга, яка живить сервопривід.
- Control - є входом для системи керування.

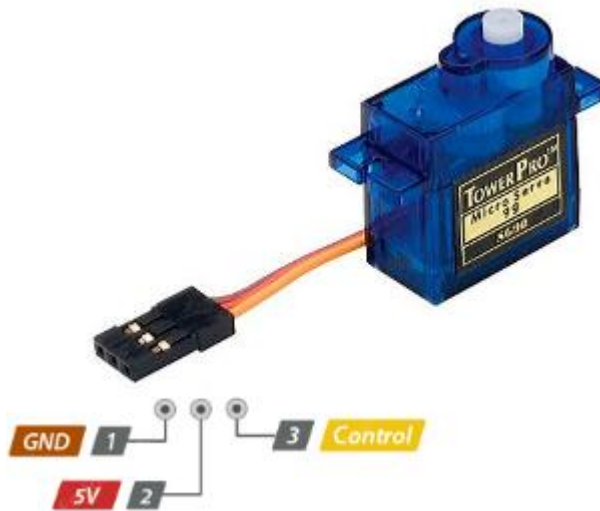


Рисунок 3.16 - Положення сервоприводу відносно довжини імпульсу

Сервоприводом можна керувати, посылаючи на нього серію імпульсів (рис. 3.17). Типовий сервопривод очікує імпульсу кожні 20 мілісекунд (тобто сигнал має бути 50 Гц). Довжина імпульсу визначає положення сервомотора.

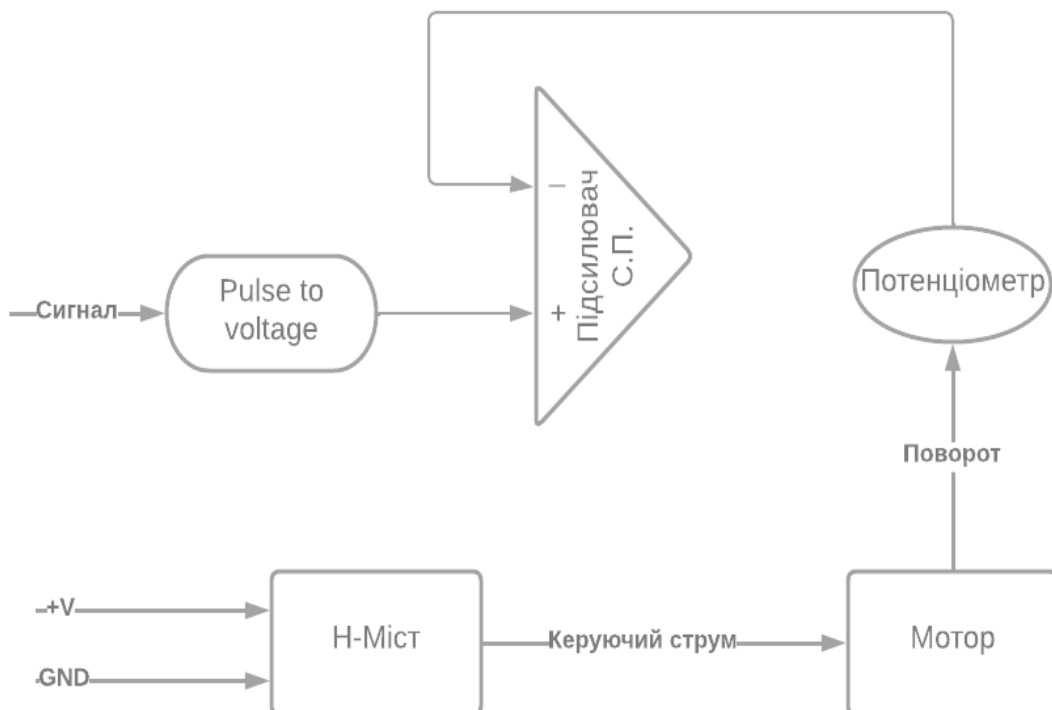


Рисунок 3.17 - Схема роботи сервоприводу

- Короткий імпульс тривалістю 1 мс або менше поверне сервопривід на 0 градусів (одна крайня).
- Тривалість імпульсу 1,5 мс поверне сервопривід на 90 градусів (середнє положення).
- Тривалість імпульсу 2 мс або близько того поверне сервопривід на 180 градусів (інша крайня).

Імпульси в діапазоні від 1 мс до 2 мс повертатимуть сервопривід у положення, пропорційне ширині імпульсу (рис. 3.18).

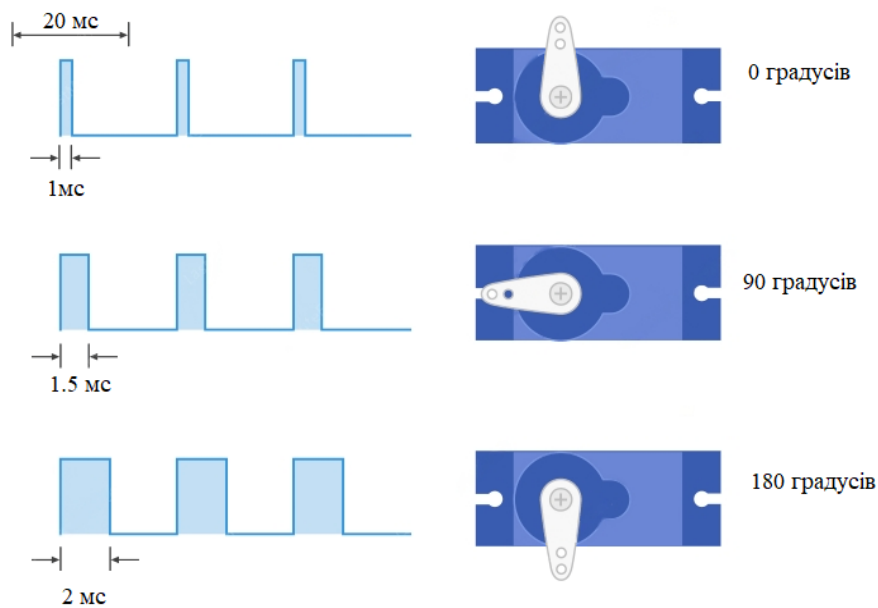


Рисунок 3.18 - Положення сервоприводу відносно довжини імпульсу

Двигун постійного струму (рис. 3.19). Мотори в цьому проекті відіграють ключову роль у руху та маневруванні робота. Основні мотори використовуються для обертання коліс, що дозволяє роботу рухатись вперед, назад та виконувати розворот. Вони забезпечують механічну силу та обертовий рух, необхідний для переміщення робота по поверхні.



Рисунок 3.19 - Зовнішній вигляд двигуна постійного струму

Кожен мотор керується мікроконтролером Arduino Uno і шилдом двигунів на базі L293D, який відправляє відповідні сигнали до мотора для контролю швидкості та напрямку обертання. Використання двигунів дозволяє роботу рухатись вперед та повертати у визначеному напрямку з відповідною швидкістю.

Крім основних моторів, в проєкті також використовується вищезазначений сервопривід, зокрема модель SG90. Сервопривід керує напрямком огляду, або "поглядом", робота. Він встановлений на спеціальну механічну конструкцію, яка дозволяє сервоприводу розвертатись в горизонтальній площині. За допомогою мікроконтролера Arduino Uno, сервопривід може контролювати свою позицію та напрямок огляду, що дає можливість роботу "дивитись" в різні боки.

Таким чином, мотори в злагодженій роботі з сервоприводом в цьому проєкті, забезпечують рух та орієнтацію робота в просторі, що є важливим для виконання завдань і переміщення по оточуючому середовищу.

Характеристики двигунів постійного струму:

Напруга живлення (Supply Voltage): Зазвичай, DC Gear Motor працюють в діапазоні від 3 до 12 вольт. Однак, це може різнитися в залежності від конкретної моделі.

Швидкість обертання (Rotation Speed): DC Gear Motor може мати різні швидкості обертання, вимірювана в оборотах в хвилину (RPM). Швидкість може бути від кількох десятків оборотів в хвилину до сотень оборотів в хвилину.

					ІА94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		62

Момент обертання (Torque): Момент обертання вимірюється у Нм (Ньютон-метр) або кгс·см (кілограм-сила на сантиметр). Це вказує на силу, з якою може обертатися вал DC Gear Motor. Вище значення моменту обертання означає більшу силу і здатність працювати з важкими навантаженнями.

Передавальне відношення (Gear Ratio): DC Gear Motor мають вбудовану систему шестерень, що дозволяє змінити швидкість обертання і збільшити момент обертання. Передавальне відношення вказує на співвідношення обертових швидкостей вала мотора та вала з виведеною силовою шестернею.

Потужність (Power): Потужність DC Gear Motor вимірюється в ватах (Вт) або міліватах (мВт) і вказує на електричну потужність, яку мотор може споживати під час роботи.

Ефективність (Efficiency): Ефективність DC Gear Motor вказує на його здатність ефективно перетворювати електричну енергію на механічну енергію з мінімумом втрат. Вона вимірюється в відсотках і чим вище значення, тим більш ефективним є мотор.

Двигун постійного струму (DC) — це обертовий електричний пристрій, який перетворює електричну енергію постійного струму в механічну (рис. 3.20). Індуктор (котушка) всередині двигуна постійного струму створює магнітне поле, яке створює обертовий рух, коли до його клеми подається напруга. Всередині двигуна знаходиться залізний вал, обмотаний дротом. Цей вал містить два фіксовані магніти, північний і південний, з кожного боку, які викликають як силу відштовхування, так і силу притягання, створюючи крутний момент.

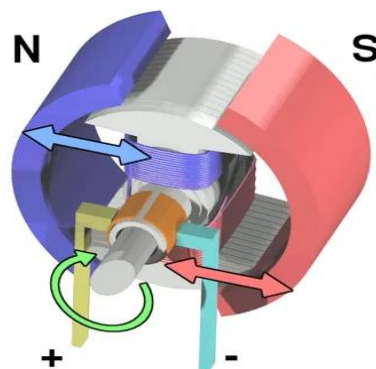


Рисунок 3.20 - Внутрішня частина DC мотору

						ІА94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат			63

Характеристик двигунів використаних в даній роботі:

Швидкість	230 ± 10% обертів на хвилину (без навантаження); 140 ± 10% обертів на хвилину (з навантаженням)
Колір	Жовтий
Матеріал	ABS-пластик, силікон, метал
Крутний момент	0.7 кг/см
Напруга	допустимі 3-6 В (робочі 6 В)
Розмір	19 x 22.5 x 64.5 мм
Струм	180-500 мА
Діаметр валу	5.5 мм
Довжина кабелю	13 см

Мотор-редуктор — це комплексна комбінація двигуна та коробки передач. Додавання редуктора до двигуна зменшує швидкість, одночасно збільшуючи вихідний крутний момент. Найважливішими параметрами щодо мотор-редукторів є швидкість (об/хв), крутний момент (фунт-дюйм) і ефективність (%).

3.2. Побудова мобільного робота, що уникає перешкод, на базі Arduino

Технічне завдання побудови мобільного робота полягає у наступному: робот має бути оснащеним ультразвуковим датчиком для вимірювання відстані до перешкод, що є основою для початку руху у невідомому середовищі. Коли відстань до перешкоди зменшується, робот інтерпретує це як наявність перешкоди, зупиняється, дивиться вліво та вправо, перш ніж обрати шлях, уникаючи перешкоди.

					ІА94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		64

Компоненти апаратного забезпечення (рис. 3.21):

- 1x Arduino UNO R3
- 1x 4-колісне шасі автомобіля
- 4x Двигун постійного струму з редуктором
- 1x Ультразвуковий датчик HC-SR04
- 1x Тримач ультразвукового датчика HC-SR04
- 1x Шилд драйвера двигунів L293D
- 1x SG90 Сервопривод
- 2x Батареї 9В
- 1x Кріплення для акумулятора
- 1x Вимикач живлення
- З'єднувальні дроти

Необхідні інструменти:

- 1x Паяльник
- Припой
- Клей
- Викрутка

Процес монтування моделі мобільного робота наступний.

Крок 1. До двигунів припаюються товсті червоні і чорні дроти до позитивних і негативних клем двигунів, затим двигуни закріплюються на шасі за допомогою гвинтів, дроти опресовуються (рис. 3.22).

Крок 2. На шасі фіксуються бокси (з використанням клею) для розміщення в них плати Arduino Uno та системи живлення, сервопривод за допомогою клею фіксується на передній частині шасі, на нього кріпиться тримач з ультразвуковим сенсором HC-SR04 (рис. 3.23).

Крок 3. Перемикач припаюється до кріплення для акумуляторів, фіксується клеєм, дроти опресовуються. В закріплені бокси встановлюються система живлення та плата Arduino Uno. Після цього шилд драйвера двигунів L293D встановлюється на плату Arduino UNO. В клеми шилда приєднуються дроти від двигунів та кріплення акумуляторів (рис. 3.24 – 3.25).

					ІА94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		65

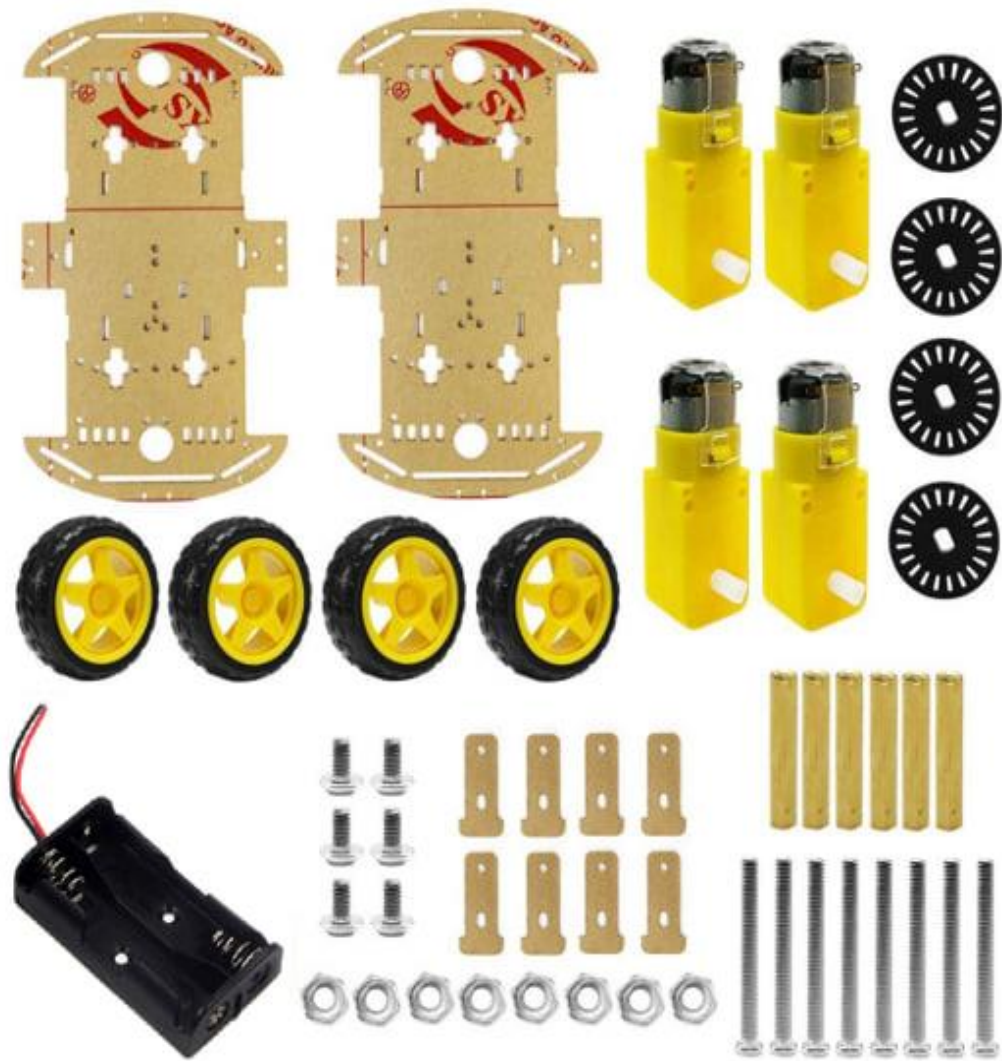


Рисунок 3.21 – Компоненти шасі робота

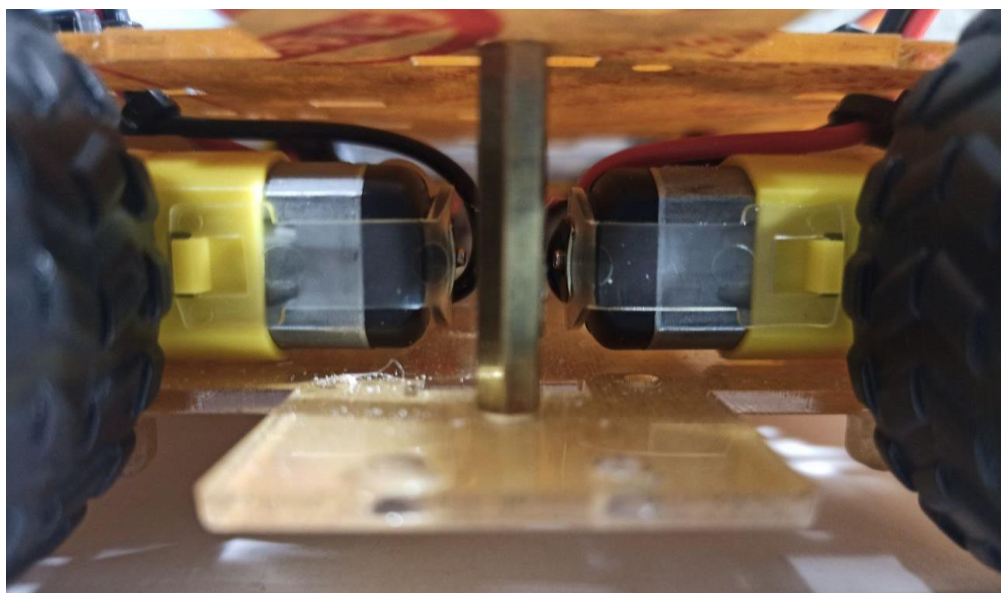


Рисунок 3.22 – Розміщення двигунів в серіїнні шасі

					ІА94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		66

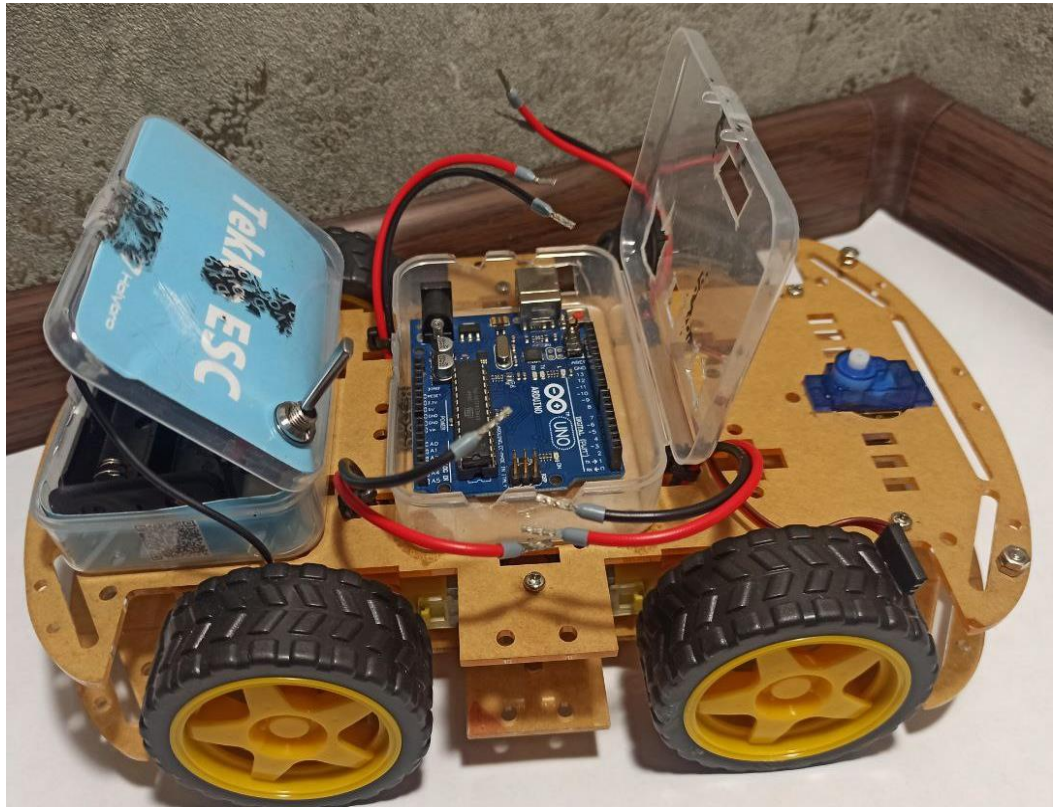


Рисунок 3.23 – Шасі з встановленими боксами для деталей робота та закріпленим сервоприводом

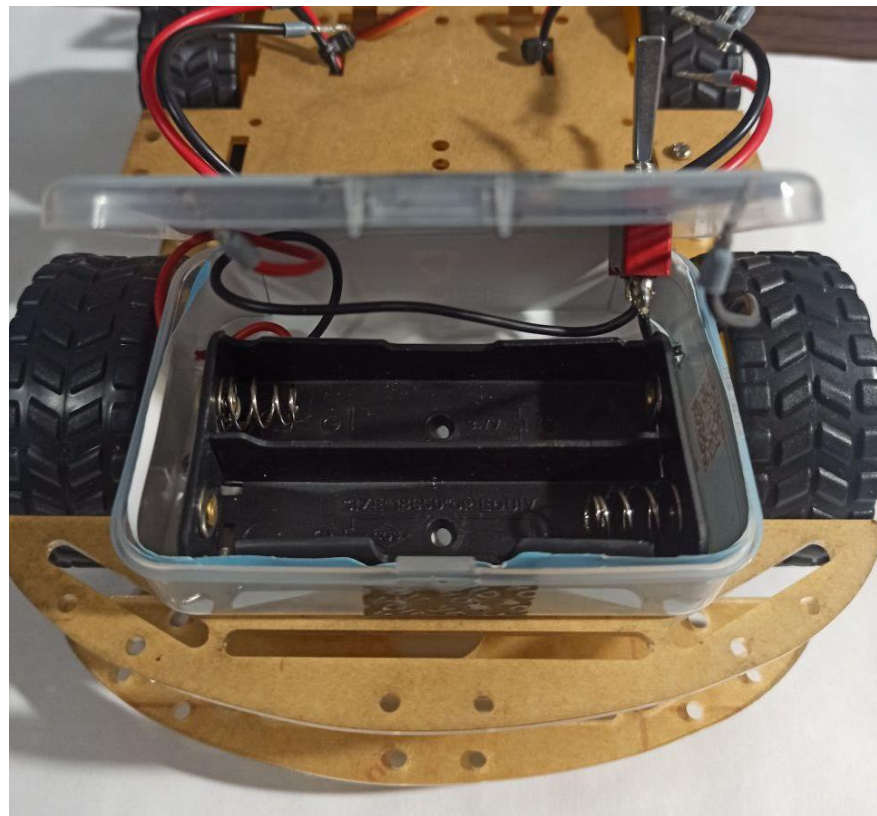


Рисунок 3.24.- З'єднання системи живлення

Зм.	Арк.	№ докум.	Підпис	Дат

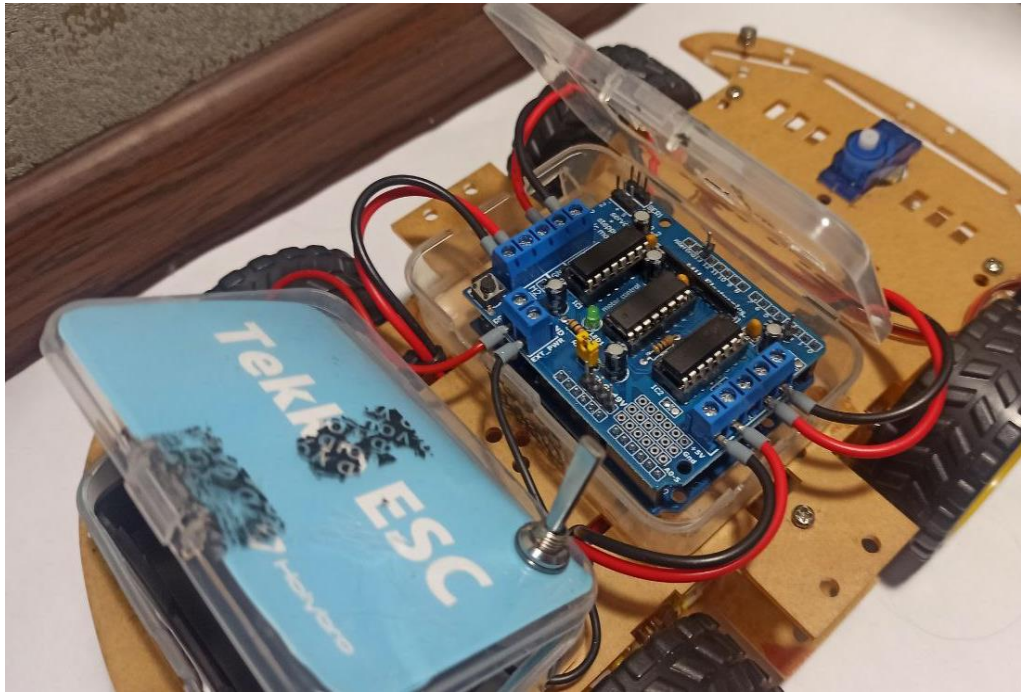


Рисунок 3.25 – З'єднання шилда з двигунами та системою живлення

Крок 4. До шилда приєднуються дроти сервоприводу, ультразвукового сенсору (рис. 3.26 – 3.27).

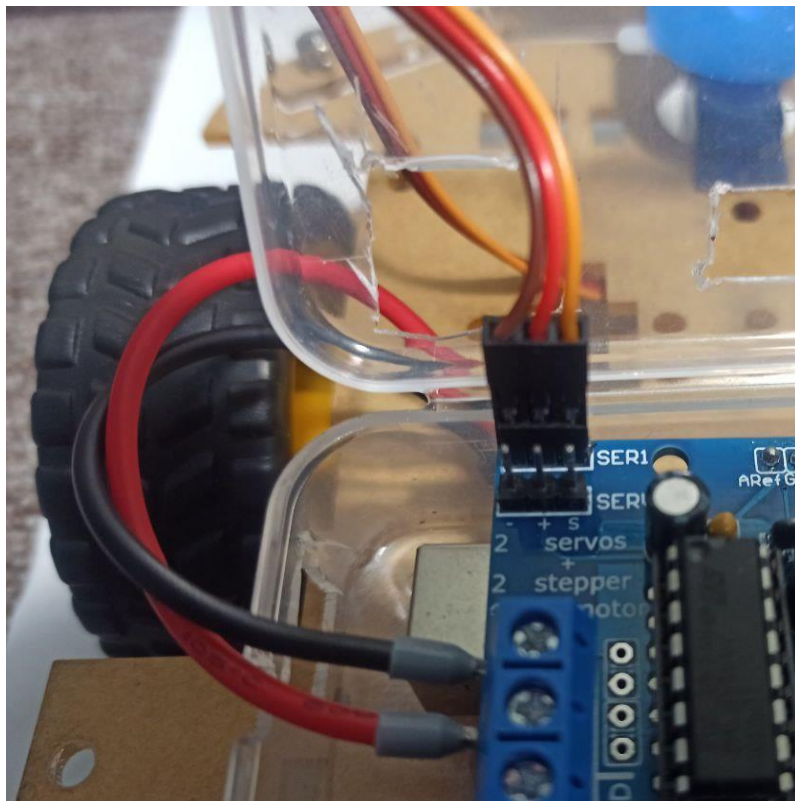


Рисунок 3.26 – Підключення сервоприводу до шилда

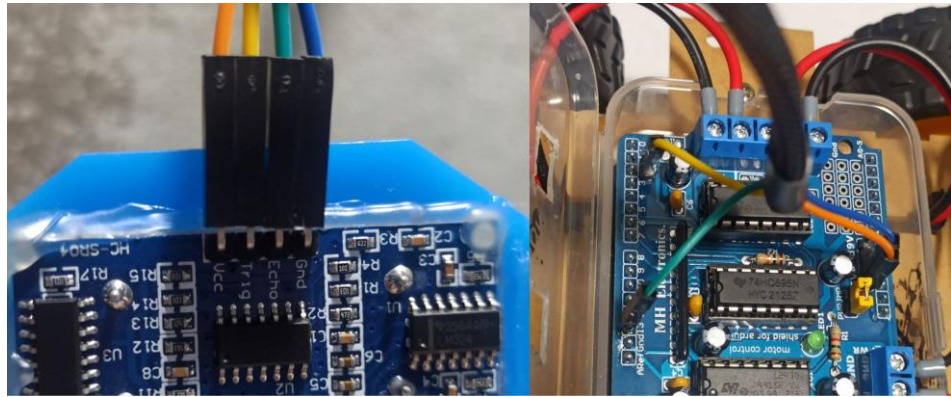


Рисунок 3.27 – Підключення ультразвукового датчика до шилда

На рис. 3.28 подано загальну схему з'єднання компонентів робота-автомобіля.

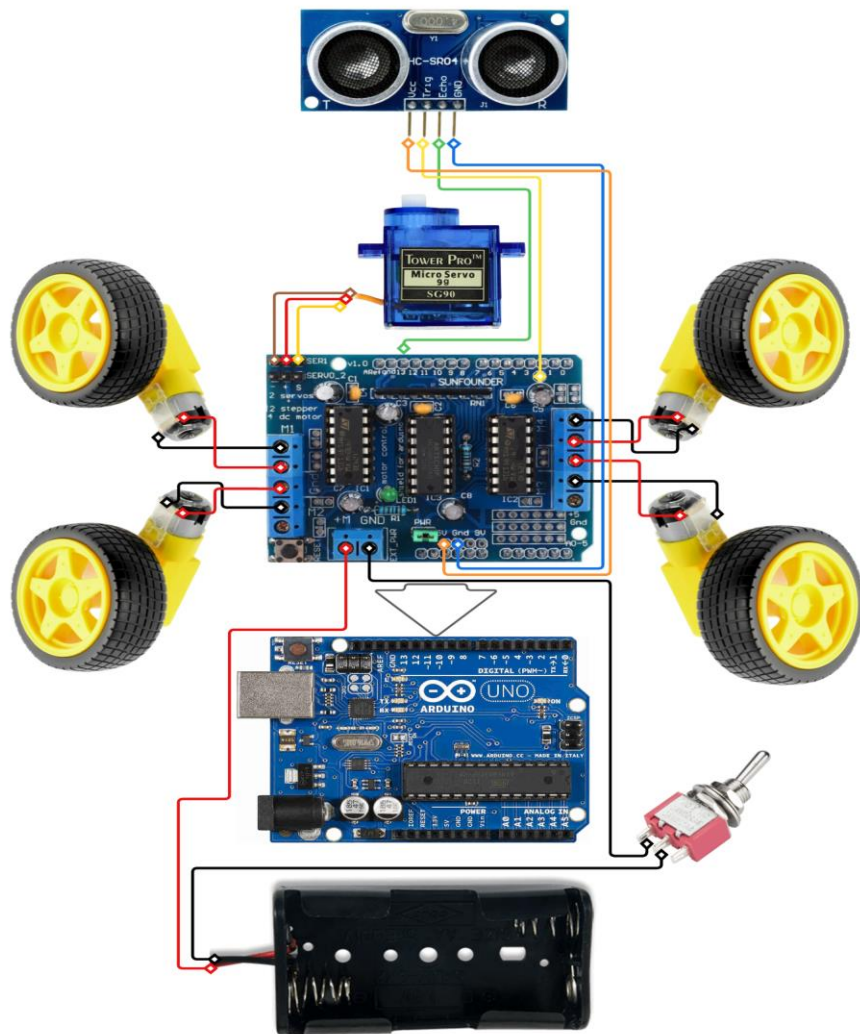


Рисунок 3.28 – Схема з'єднання компонентів робота-автомобіля

									Арк.
									69
Зм.	Арк.	№ докум.	Підпис	Дат	ІА94.150БАК.004 ПЗ				

3.3 Програмні засоби проектування мобільного робота

Мікроконтролер Arduino поєднується з ПК через USB-з'єднання. Дані передаються між платою та ПК побітово. Адаптер використовується для живлення плати, а програматор USB – для запису апаратної програми (написаної в Arduino IDE) на платі.

Кодування Arduino. Для реалізації мети проекту було здійснено кодування програм для платформи Arduino. Arduino IDE є основним інструментом для розробки програм для Arduino і надає простий інтерфейс для написання, компіляції та завантаження програм до плати Arduino. Етапи написання програми наступні.

1. Створення нового проекту. Після встановлення Arduino IDE програму було відкрито і створено новий проект. Проект можна створити, натиснувши на кнопку "New" або вибравши "File" > "New". Це створить новий файл програми, який буде відкритий в Arduino IDE.

2. Синтаксис Arduino. Arduino використовує мову програмування, яка базується на мові C/C++. Основний код програми Arduino складається з двох функцій - setup() та loop(). Функція setup() виконується один раз при запуску плати Arduino, тоді як функція loop() виконується безкінечно, постійно повторюючи свій код. Весь код програми повинен бути написаний всередині цих функцій.

3. Основні функції та бібліотеки. Arduino надає ряд вбудованих функцій та бібліотек, які допомагають взаємодіяти з різними компонентами та модулями. Наприклад, можна використати функції pinMode() для встановлення режиму піна (наприклад, ввід або вивід), digitalRead() для зчитування стану цифрового піна, digitalWrite() для запису значення на цифровий пін, або бібліотеки, такі як Servo для керування сервоприводом.

4. Взаємодія з платою Arduino. Arduino дає змогу звертатися до конкретних пінів та компонентів плати, використовуючи їх номери або назви. Наприклад, digitalRead(13) зчитує стан піна 13, або analogWrite(9, 255) встановлює аналогове значення 255 на пін 9. Також є можливість використати змінні для збереження значень та працювати з ними у коді.

					ІА94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		70

5. Компіляція та завантаження програми. Після написання коду програми в Arduino IDE її було скомпільовано, для цього знадобилося натиснути кнопку "Verify" або вибрати "Sketch" > "Verify/Compile". Якщо програма успішно компілюється, вона готова до завантаження на плату Arduino, підключивши плату до комп'ютера та натиснувши кнопку "Upload" або "Sketch" > "Upload". Arduino IDE компілює написану програму та завантажує її на плату Arduino через з'єднання USB.

6. Тестування та налагодження. Після завантаження програми на плату Arduino можна виконати тестування та налагодження, щоб переконатися, що робот працює як і очікувалося. Потрібно спостерігати за поведінкою робота та перевіряти, чи відповідає він заданим вимогам.

На цьому етапі здійснюється процес кодування програм для Arduino, використовуючи Arduino IDE та мову програмування C/C++. Це дозволяє забезпечити взаємодію різних компонентів робота та здійснити керування їхньою поведінкою за допомогою програмного коду. Кодування Arduino є важливим етапом у досягненні мети проекту, оскільки це надає можливість програмувати функціональність робота та забезпечує його працездатність.

На рис. 3.29 подано блок-схему алгоритму системи навігації мобільного робота. Імплементацію програми системи навігації мобільного робота подано в Додатку В.

					ІА94.150БАК.004 ПЗ	Арк.
						71
Зм.	Арк.	№ докум.	Підпис	Дат		

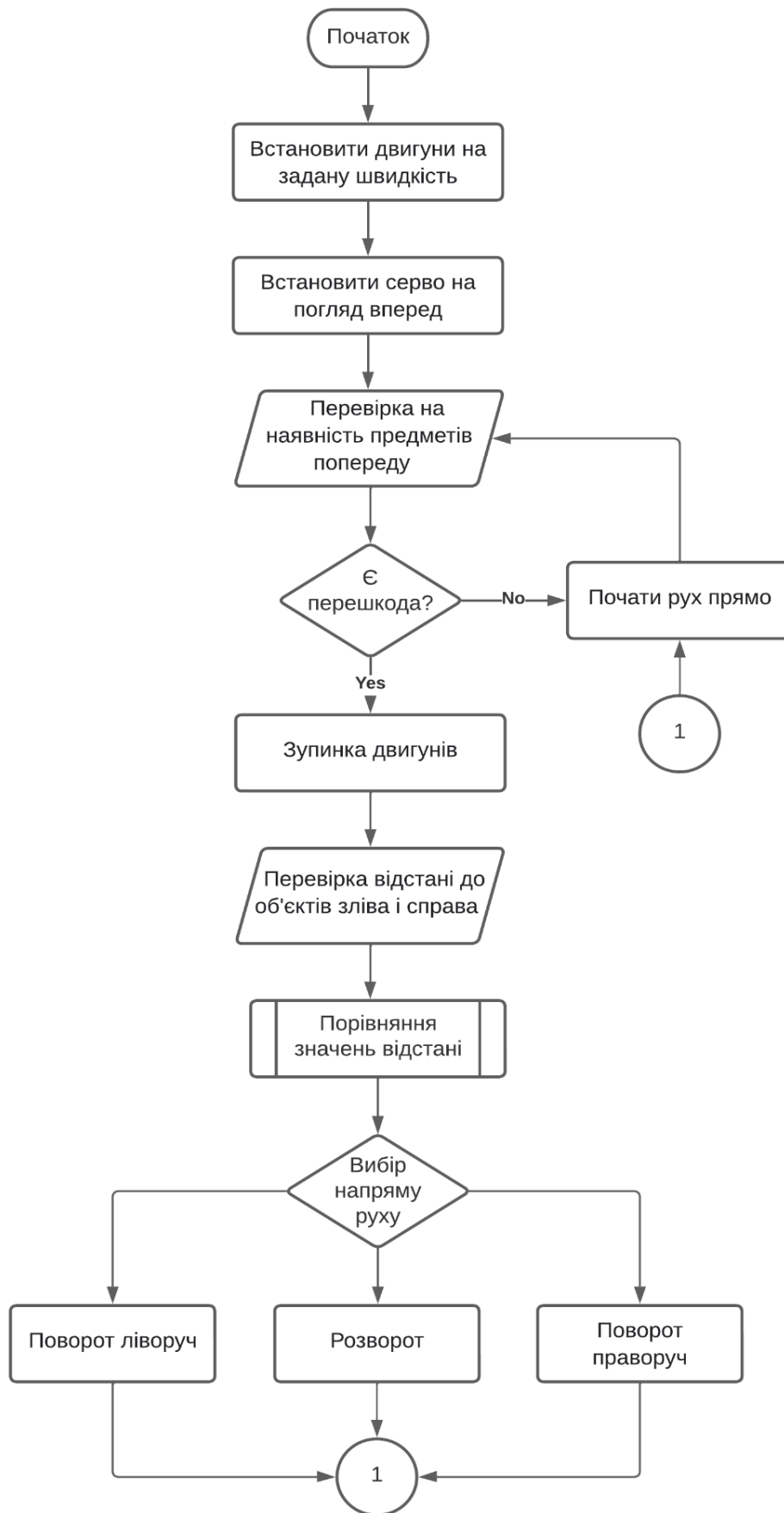


Рисунок 3.29 – Блок-схема алгоритму системи навігації мобільного робота

3.4. Тестування мобільного робота

Результатом роботи є мобільний робот, керований Arduino, який пересувається, виявляючи перешкоди на своєму шляху та уникаючи їх (рис. 3.30). Під час роботи робота сонар посилає ультразвукову хвилю в переднє положення (90 градусів), ліворуч (180 градуси) і праворуч (0 градусів). Коли звукова хвиля стикається, зіштовхнувшись із перешкодою, відбивається, відстань зберігається для правого та лівого положення. Після цього мікроконтролер порівнює значення на основі свого алгоритму і визначає, в якому напрямку здійснити розворот.

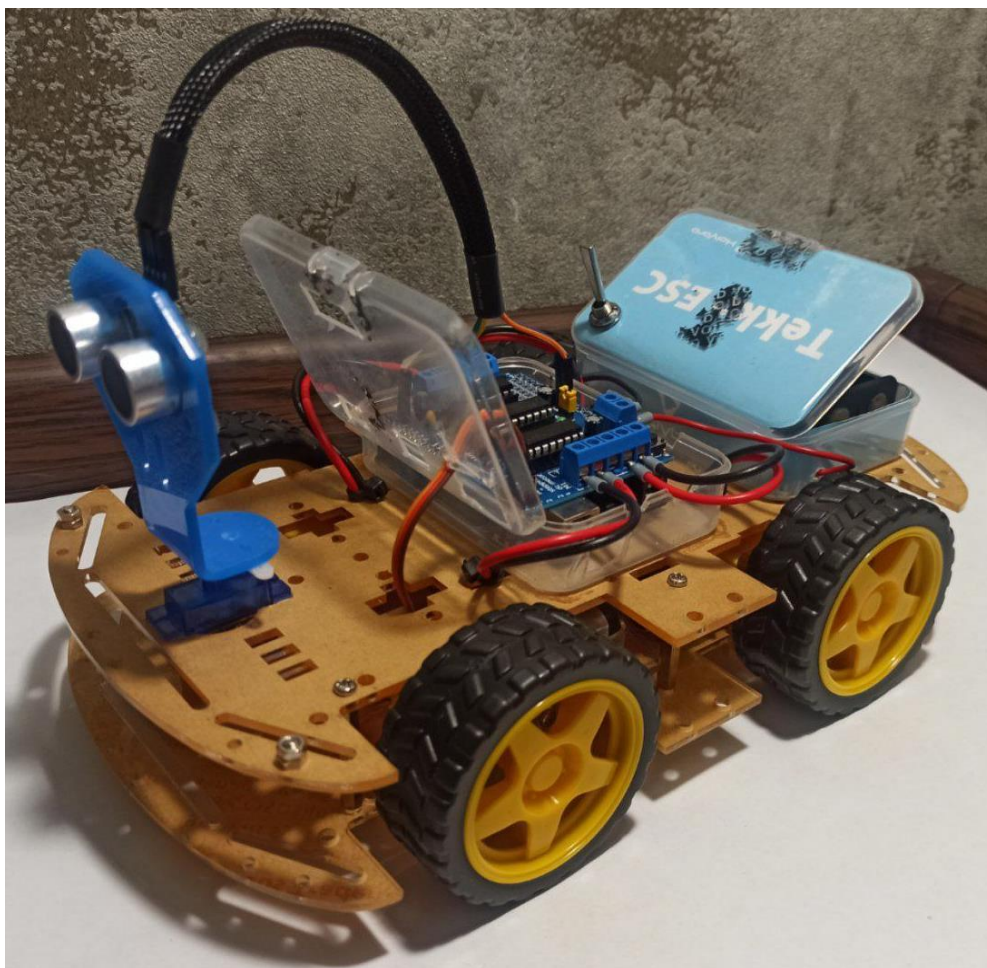


Рисунок 3.30 – Апаратна модель мобільного робота, що уникає перешкод

Тести, проведені на створеному апаратному забезпеченні, встановили обмеження алгоритму розпізнавання. Це стосувалося випадків, коли деякі

перешкоди не виявлялися через те, що датчик не може вимірювати перешкоди за межами діапазону вимірювання. Коли об'єкт (перешкода) є на шляху робота і цей об'єкт не знаходиться в зоні «видимості» датчика, перешкоду не буде виявлено, що призведе до зіткнення. Щоб уникнути цього, випробування проводилися в замкнутій зоні, де стіна є єдиною перешкодою, і автомобіль міг вільно рухатися без зіткнень. Щоб створити робота, який буде виявляти численні перешкоди та уникати їх, потрібно використовувати більшу кількість різних датчиків, щоб охопити ширший діапазон виявлення перешкод.

3.5. Обговорення результатів проєкту

Ефективність датчика: використовуваний датчик є одним із найважливіших компонентів. Це дуже важливий компонент, оскільки продуктивність створеного робота критично залежить від точності датчика. Ультразвуковий датчик HC-SR04 працює, випромінюючи високочастотну звукову хвилю (40 кГц) через один із своїх п'єзоелектричних перетворювачів і виявляючи зворотні імпульси (echo) через інший перетворювач. Коли перешкоди на шляху робота не перебували під кутом 30 градусів, звукові хвилі, які випромінював датчик, не могли виявити об'єкт. Це було проблемою, оскільки робот може зіткнутися з об'єктом, який розташований нижче датчика, або об'єктами, які перебувають на шляху робота, але не потрапляють у зону дії датчика. За виключанням цієї ситуації з діапазоном, в інших випадках датчик міг виявляти об'єкти в межах свого діапазону для уникнення перешкод.

Напруга батареї: батарея була сильно навантажена компонентами, особливо двигунами. Двигуни потребують багато енергії для своєї роботи, особливо дешеві двигуни, оскільки вони мають менший ККД. Щит драйвера двигуна живиться окремо від Arduino. Для Arduino потрібне джерело живлення 9-12 В, і він живиться від однієї батареї 9 В, тоді як екран драйвера двигуна потребує 5-36 В і живиться від двох батарей 9 В, з'єднаних послідовно. Двигуни, серводвигун і датчик потребують 3-6 В кожен.

					IA94.150BAK.004 ПЗ	Арк.
						74
Зм.	Арк.	№ докум.	Підпис	Дат		

Вплив тепла на екран драйвера двигуна: безперервний струм, який може подавати екран, залежить від того, наскільки добре екран зберігається прохолодним. Плата сконструйована таким чином, щоб відводити тепло від мікросхем драйвера двигуна, але продуктивність все одно можна покращити, додавши радіатори. Коли температура мікросхеми драйвера двигуна наближається до своєї межі, до двигуна подається менше потужності, що призведе до того, що двигуни не зможуть обертатися. Екран був протестований при кімнатній температурі без примусового повітряного потоку або радіаторів, і він міг забезпечити постійний струм до чотирьох каналів двигуна, перш ніж тепловий захист почав зменшувати струм. Шилд випромінював достатньо енергії, щоб робот рухався протягом близько 15 хвилин, після чого потужності стало недостатньо для подальшого пересування.

Висновки розділу 3

Виявлення та запобігання перешкодам у процесі руху розглядається як центральна проблема при розробці мобільних роботів. Її розв'язання полягає у наділенні роботів сенсорами для навігації у невідомому середовищі. Отриманий результат та здійснені експерименти підтвердили, що, базуючись на Arduino UNO та Adafruit Motor Shield із кодом написаним у програмному забезпеченні Arduino IDE, можливо створити мобільного робота. Будучи повністю автономним (після завантаження коду, передбачуваного методологією), не вимагаючи участі оператора у процесі пересування, робот, як показали експерименти, здатен успішно маневрував у невідомому середовищі без зіткнень. Використана методологія (зокрема, апаратно-програмне забезпечення навігації) дозволило реалізувати поставлені у п.1.5 завдання. Створений мобільний робот здатний:

- орієнтуватися у відкритому просторі, виявляючи перешкоди на своєму шляху на основі заданої граничної відстані;
- змінювати свій курс, маневрувати та уникати зіткнень, ухвалюючи рішення у режимі реального часу;

					IA94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		75

- діяти у невідомому середовищі, не вимагаючи зовнішнього керування під час реалізації його функцій.

Апаратне забезпечення, яке було використано під час реалізації проєкту, є широко доступне і недорогим, що робить прототип легко відтворюваним. Базові принципи створення мобільного робота, котрий уникає перешкоди, застосовані під час даної бакалаврської роботи, можна використати у подальших дослідженнях різних систем навігації, розширюючи функції робота, виходячи з поставлених нових завдань.

					ІА94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		76

ВИСНОВКИ

Мобільна робототехніка як технологічна галузь та галузь наукових досліджень останні два десятиліття зазнала неймовірних перетворень, перед усім, завдяки інформаційно-комунікаційним технологіям. Це відкрило нові можливості для стрімкої автоматизації транспортних засобів та створення SDC та мобільних роботів різного функціонального призначення, які здатні пересуватися без участі людини-водія. Водночас, проекти у галузі робототехніки, зокрема такого масштабу як SDC, вимагають чимало ресурсів (фінансових, трудових) та практичних інструментів реалізації ідей та наукових доробок. До того ж винахідники витрачають чимало часу на створення пілотних зразків інженерних рішень і для конкретної конфігурації обладнання часто вимушені «винаходити велосипед» у процесі створення дослідного зразка. Платформа Arduino, маючи низку новацій і переваг у застосуванні (що підтверджує це дослідження), дає можливість достатньо легко створити прототип мобільних роботів заданої конфігурації. Це прискорює отримання результатів дослідницьких проектів та здійснення їх тестування. Завдяки недорогим, зручним у користуванні та налаштуванні роботизованим рішенням, Arduino сприяє новаторству у робототехніці, відкриваючи широкі можливості для реалізації нових проектів, зокрема і проекту «Система навігації самохідної техніки на базі Arduino». Цей проект реалізовано з використанням ультразвукового датчика для виявлення об'єктів (статичних перешкод) у навколишньому середовищі, Motor Driver Shield для керування чотирма двигунами постійного струму для руху коліс робота за допомогою мікроконтролера Arduino.

Серед факторів, які впливають на чітку навігацію сконструйованого робота, – середовище, в якому відбувалося тестування, а також кількість наявних перешкод у тестовому просторі. Ці фактори впливають на датчик (у даному випадку ультразвук), що вказує на те, що точності пересування мобільного робота залежить від якості датчика.

					ІА94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		77

Запропонований підхід має окремі обмеження: продуктивність і точність створеного мобільного робота значною мірою залежать від моделі ультразвукового датчика, який застосовується, та його якості. Застосований датчик не може точно виміряти відстань між роботом і об'єктом (перешкодою), коли: відстань сягає більше 3 метрів, кут занадто малий, об'єкт занадто малий, поверхня перешкоди погано відбиває (чи зовсім не відбиває) ультразвуковий сигнал.

Для повної автономності робот повинен мати більше можливості адаптуватися до навколишнього середовища (в т.ч. у відкритому просторі); бути спроможним працювати протягом тривалого часу без зовнішнього втручання, уникати загрозливих для нього ситуацій тощо. Для кращих результатів і підвищеної точності пересування створений мобільний робот вимагає додаткових датчиків. Крім того, використання кращих приводів забезпечить більш швидку навігацію та ефективнішу роботу. Рекомендації для подальших досліджень полягають у наступному. Система навігації буде якіснішою, якщо до проєкту додати: камери – це дозволить вивести робота за межі прямої видимості (крім того, камеру можна використовувати для обробки зображень); бездротові технології – це забезпечить зовнішній зв'язок і керування роботом із віддаленого комп'ютера.

Положення та результати дослідження подано у публікаціях автора [36, 58-59].

					ІА94.150БАК.004 ПЗ	Арк.
						78
Зм.	Арк.	№ докум.	Підпис	Дат		

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Sabry F. Self Driving Car: Solving Full Self-driving Need Solving Real-world Artificial Intelligence. One Billion Knowledgeable. 2022 . 293 p.
2. Shapiro D. G. Three Anecdotes from the DARPA Autonomous Land Vehicle Project. AI Magazine, 2008. 29(2), 40. <https://doi.org/10.1609/aimag.v29i2.2108>
3. Unmanned ground vehicle. URL: https://en.wikipedia.org/wiki/Unmanned_ground_vehicle.
4. Pillath S. Automated vehicles in the EU. URL: [https://www.europarl.europa.eu/RegData/etudes/BRIE/2016/573902/EPRS_BRI\(2016\)573902_EN.pdf](https://www.europarl.europa.eu/RegData/etudes/BRIE/2016/573902/EPRS_BRI(2016)573902_EN.pdf)
5. U.S. Department of Transportation Releases Policy on Automated Vehicle Development. URL: <https://www.transportation.gov/briefing-room/usdepartment-transportation-releases-policy-automated-vehicle-development>.
6. Young R. Critical Analysis of Prototype Autonomous Vehicle Crash Rates Six Scientific Studies from 2015-2018 SAE International. 2021. 252 p
7. Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles J3016_202104. URL: https://www.sae.org/standards/content/j3016_202104/
8. Поліщук М. М., Ткач М.М. Робототехнічні системи: проектування і моделювання: навч. посіб. Київ: КПІ ім. Ігоря Сікорського, 2021. 112 с.
9. Рудик А. В. Наукові основи та принципи побудови приладової системи вимірювання прискорення мобільного робота : дис. ... д-ра техн. наук : 05.11.01 – Прилади та методи вимірювання механічних величин. Київ, 2018. – 460 с.
10. Мигаль В. Д. Інтелектуальні системи в технічній експлуатації автомобілів: монографія. Х.: Майдан, 2018. 262 с.
11. Velasco-Hernandez, G. et al. Autonomous Driving Architectures, Perception and Data Fusion: A Review. In Proceedings of the 2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP 2020), Cluj-Napoca, Romania, 3–5 Sept. 2020. DOI: [10.1109/ICCP51029.2020.9266268](https://doi.org/10.1109/ICCP51029.2020.9266268)

					ІА94.150БАК.004 ПЗ	Арк. 79
Зм.	Арк.	№ докум.	Підпис	Дат		

12. Said E. et al. Visibility-Based Technologies and Methodologies for Autonomous Driving. In book: "Self-driving Vehicles and Enabling Technologies" edited by Prof. Marian Gaiceanu. 2020. pp.1-25.
13. Staron M. Automotive Software Architectures: An Introduction. Second Edition Springer International Publishing A&G. 2022. 274 p
14. Raouf, I. et al. Sensor-Based Prognostic Health Management of Advanced Driver Assistance System for Autonomous Vehicles: A Recent Survey. Mathematics 2022, 10, 3233. <https://doi.org/10.3390/math10183233>
15. Vargas J. et al. An Overview of Autonomous Vehicles Sensors and Their Vulnerability to Weather Conditions. Sensors. 2021; 21(16):5397. <https://doi.org/10.3390/s21165397>
16. Wevolver 2020 Autonomous Vehicle Technology Report. URL: <https://www.coursehero.com/file/62601304/Wevolver2020AutonomousVehicleTechnologyReportpdf/> (accessed 2023 June 1)
17. Yeong, D.J. et al. Sensor and Sensor Fusion Technology in Autonomous Vehicles: A Review. Sensors 2021, 21, 2140. <https://doi.org/10.3390/s21062140>
18. Background Information on CCD and CMOS Technology. URL: https://www.tedpella.com/cameras_html/ccd_cmos.aspx (accessed 2023 June 1)
19. A Look Inside ADAS Modules. URL: <https://amkor.com/semiconductor-story/a-look-inside-adas-modules/>
20. AV and ADAS Sensors. URL: <https://community.sw.siemens.com/s/article/AV-and-ADAS-Sensors> (accessed 2023 June 1)
21. A Look Inside ADAS Modules. . URL: <https://amkor.com/semiconductor-story/a-look-inside-adas-modules/> (accessed 2023 June 1)
22. Drive and stay cool changing the lane with Blind Spot Detection. URL: <https://www.continental-automotive.com/en-gl/Passenger-Cars/Autonomous-Mobility/Functions/Cruising-Driving/Blind-Spot-Detection>

23. Гуржій А. М. Основи автоматики та робототехніки: Навчальний посібник/ А. М. Гуржій, А. Т. Нельга, В. М. Співак, О. С. Ітякін:—Дніпро:«Гарант СВ», 2021.- 243с.
24. Lim, B.S. et al. Autonomous Vehicle Ultrasonic Sensor Vulnerability and Impact Assessment. In Proceedings of the IEEE World Forum on Internet of Things (WF-IoT), Singapore, 5–8 February 2018; pp. 231–236.
25. Xu W. et al. Analyzing and Enhancing the Security of Ultrasonic Sensors for Autonomous Vehicles, in IEEE Internet of Things Journal, vol. 5, no. 6, pp. 5015-5029, Dec. 2018. DOI: [10.1109/JIOT.2018.2867917](https://doi.org/10.1109/JIOT.2018.2867917)
26. Ultrasonic Sensors in Self-Driving Cars Babak Shahian Jahromi Babak Shahian Jahromi. URL: <https://medium.com/@BabakShah/ultrasonic-sensors-in-self-driving-cars-d28b63be676f> (accessed 2023 June 1)
27. LIDAR Sensor in Autonomous Vehicles: Why it is Important for Self-Driving Cars? URL: <https://www.cogitotech.com/blog/lidar-sensor-in-autonomous-vehicles-why-it-is-important-for-self-driving-cars/>
28. Agnihotri N. What are the sensors used in self-driving cars? URL: <https://www.engineersgarage.com/what-are-the-sensors-used-in-self-driving-cars/>
29. A Guide to Lidar Wavelengths for Autonomous Vehicles and Driver Assistance. URL: <https://velodynelidar.com/blog/guide-to-lidar-wavelengths/>
30. SAE Levels of Driving Automation™ Refined for Clarity and International Audience. URL: <https://www.sae.org/blog/sae-j3016-update> (accessed 2023 June 3)
31. GM Super Cruise. URL: <https://gmauthority.com/blog/gm/general-motors-technology/general-motors-autonomous-technology/gm-super-cruise/> (accessed 2023 June 1)
32. Padgett M. Here's how the 2019 Audi A8 will become the first Level 3 self-driving car* URL: https://www.motorauthority.com/news/1111528_heres-how-the-2019-audi-a8-will-become-the-first-level-3-self-driving-car# (accessed 2023 June 1)

					IA94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		81

33. Hyundai Motor Group's Roboride Experience: A Peek into the Autonomous Driving Era. URL: <https://www.hyundaimotorgroup.com/story/CONT0000000000082639> 0 (accessed 2023 June 1)
34. Mullen D. Watch waymo self-driving car navigating san francisco in heavy rain. URL: <https://www.driving.co.uk/news/technology/watch-waymo-self-driving-car-navigating-san-francisco-in-heavy-rain/> (accessed 2023 June 7)
35. Q&A: Arduino founder Massimo Banzi. URL: <https://www.helsinkitimes.fi/business/22390-q-a-arduino-founder-massimo-banzi.html>
36. Саліхов М. М. Arduino – перспективний інструмент протитопування у робототехніці // Концептуальні шляхи розвитку науки та освіти: матеріали VIII Міжнародної науково-практичної конференції м. Львів, 9-10 червня 2023 року. – Львів: Львівський науковий форум, 2023. – с. 73-79.
37. Bräunl T. Embedded Robotics: From Mobile Robots to Autonomous Vehicles with RaspberryPi and Arduino Springer Nature, 2022. 519 p.
38. Navigation URL: <https://www.britannica.com/technology/navigation-technology>
39. Навігаційні системи [Електронний ресурс] : навч. посіб. для студ. спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» / С.Л. Лакоза; КПІ ім. Ігоря Сікорського, 2021. — 80 с.
40. Imad M. et al. Navigation system for autonomous vehicle: A survey. JCSTS., vol. 2, no. 2, pp. 20–35, 2020.
41. Ковалець І. В. та ін. Технологія планування траєкторій руху мобільних об'єктів з урахуванням перешкод на складній місцевості. Енергетика і автоматика. - 2017. - № 1. - С. 110-122.
42. Aguilo I. et al. Artificial Intelligence Research and Development Hardcover by IOS Press. 2003. 500 p.
43. Rathin Chandra Shit Precise localization for achieving next-generation autonomous navigation: State-of-the-art, taxonomy and future prospects. Computer

					IA94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		82

Communications Volume 160, 1 July 2020, Pages 351-374.

<https://doi.org/10.1016/j.comcom.2020.06.007>

44. Гребенюк Б. А. Розробка підсистеми управління інтелектуальним роботом / Б. А. Гребенюк // «Automation and Development of Electronic Devices» ADED-2023: Collection of Students' Scientific Paper. – Kharkiv : Kind of Kharkiv National University of Radio Electronics [electronic edition], 2023. – Part 1. –336p. P. 263-269
45. Erick J. Rodríguez-Seda, Dušan M. Stipanović Chapter 4 - Guaranteed Collision Avoidance with Discrete Observations and Limited Actuation // Advances in Intelligent Vehicles. Academic Press, 2014, p. 89-110.
<https://doi.org/10.1016/B978-0-12-397199-9.00004-5>
46. Becker M. et al Obstacle avoidance procedure for mobile robots. ABCM Symposium series in Mechatronics. Vol. 2, 2006. P. 250-257.
47. Nowakowski M., Kurylo, J. Usability of Perception Sensors to Determine the Obstacles of Unmanned Ground Vehicles Operating in Off-Road Environments. Appl. Sci. 2023, 13, 4892. <https://doi.org/10.3390/app13084892>
48. Yasin, J.N., Mohamed, S.A.S., Haghbayan, M.H. et al. Low-cost ultrasonic based object detection and collision avoidance method for autonomous robots. Int. j. inf. technol. 13, 97–107 (2021). <https://doi.org/10.1007/s41870-020-00513-w>
49. Звенігородський О.С. Інтелектуальна система планування тактики руху автономного робота в квазістаціонарному середовищі: Дис... канд. техн. наук: 05.13.23. — Д., 2002. — 127с.
50. Адамів, О.П. Моделі та інтелектуальні засоби адаптивного керування автономним мобільним роботом. Дис. ... канд.техн. наук. Одеса, 2007. 124 с.
51. Koval V., Adamiv O., Proc. of the Third IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'2005). – Sofia (Bulgaria). – 2005. – P. 120- 124.
52. Kojima T. et al. Intelligent Technology for More Advanced Autonomous Driving. 2018. Hitachi Review Vol. 67, No. 1. p. 58–63

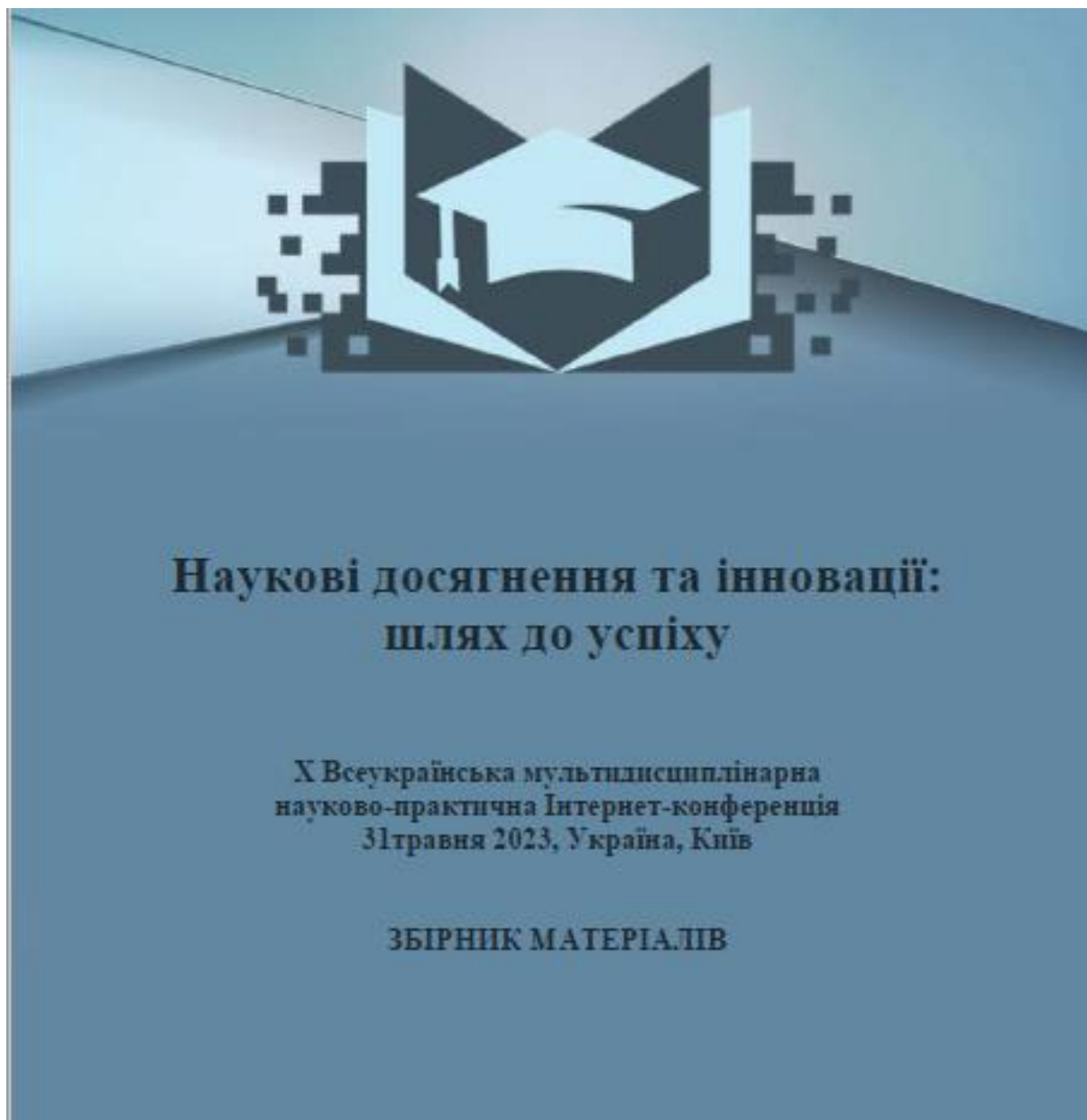
					ІА94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		83

53. Pandey A, Pandey S, Parhi DR. Mobile robot navigation and obstacle avoidance techniques: A review. *Int Rob Auto J.* 2017;2(3):96-105.
DOI: [10.15406/iratj.2017.02.00023](https://doi.org/10.15406/iratj.2017.02.00023)
54. Liu Q, Lu YG, Xie C. Optimal Genetic Fuzzy Obstacle Avoidance Controller of Autonomous Mobile Robot Based on Ultrasonic Sensors. *IEEE International Conference on Robotics and Biomimetics.* 2006. p. 125–129.
55. Mamchur, D.; Peksa, J.; Kolodinskis, A.; Zigunovs, M. The Use of Terrestrial and Maritime Autonomous Vehicles in Nonintrusive Object Inspection. *Sensors* 2022, 22, 7914. <https://doi.org/10.3390/s22207914>
56. McGuire K. et al. A Comparative Study of Bug Algorithms for Robot Navigation. *Robotics and Autonomous Systems.* Vol. 121, 2019.
<https://doi.org/10.1016/j.robot.2019.103261>
57. Саліхов М. М. Від автоматизованих до автономних транспортних засобів // Наукові досягнення та інновації: шлях до успіху. X Всеукраїнська мультидисциплінарна науково-практична Інтернет-конференція, 31 травня 2023, Україна, Київ : зб. матеріалів — Електрон. дан. — Київ : Ярочé нко Я. В., 2023. — с. 136-142.
58. Саліхов М. М. Технологічні бар'єри розвитку безпілотних транспортних засобів // зб. наук. пр. Міжнародної науково-практичної конференції «Сучасні аспекти діджиталізації та інформатизації в програмній та комп'ютерній інженерії», 01-03 червня 2023 р., м. Київ. С.101-102.
59. Саліхов М.М. Самокеровані автомобілі та системи їх навігації// Практичні та теоретичні питання розвитку науки та освіти: матеріали VIII Міжнародної науково-практичної конференції: м. Львів, 19-20 червня 2023 року. – Львів: Львівський науковий форум, 2023. – С.53-59.

					ІА94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		84

ДОДАТКИ

Додаток А



**Наукові досягнення та інновації:
шлях до успіху**

X Всеукраїнська мультидисциплінарна
науково-практична Інтернет-конференція
31 травня 2023, Україна, Київ

ЗБІРНИК МАТЕРІАЛІВ

					ІА94.150БАК.004 ПЗ	Арк.
						85
Зм.	Арк.	№ докум.	Підпис	Дат		

ЗМІСТ

ВСТУП.....	7
Айзенбарт Л. М.	
Стратегії побудови культурної ідентичності в літературних текстах галицьких письменників кін. XIX – поч. XX ст. (М. Яцків, Й. Рот, Б. Шульц)	8
Безручко І. О.	
Генеza монетарного союзу зони євро. Перспективи приєднання до Єврозони для України	11
Білецький Г. В.	
Аналіз конструкцій висівних апаратів катушкового типу для сівби дрібно насінневих культур.....	15
Борисенко Н. О., Маріуца А. Е., Сироватка Н. Ю.	
Аналіз генетичної структури рамчастих коропів антонінсько-зозуленецького внутрішньопородного типу	20
Васильченко Р. В.	
Ідеї Памфіла Юркевича в контексті розвитку філософії кордоцентризму	24
Весна А. Є.	
Договори про спільну діяльність. Види договорів щодо розпорядження майновими правами інтелектуальної власності. Договір комерційної концесії	29
Вороніна С. В., Наумова Н. А.	
Роль гендеру у сприйнятті відчуття небезпеки під час воєнного конфлікту	35
Геєр І. О., Стицюк О. М.	
Розвиток будівельної галузі в умовах воєнного часу	41
Гнедюк В. Л.	
Особливості адміністративно-правової охорони кібернетичної безпеки	46
Горбонос Н. О., Нехаєва А. Г.	
Альтернативне житло на базі морських контейнерів.....	50
Єрємін П. А., науковий керівник Миронець О.М.	
Воєнний стан як стимул для швидкого розвитку науки та техніки: історичний аналіз.....	55
Ільніцька Т. С.	
Підготовка фахівців до професійної діяльності в умовах цифровізації освіти медичних коледжів.....	57
Казіміров М. М., науковий керівник Женгліньська Г. О.	
Реалізація норм права	61
Каліна І. Г.	
Теоретичні основи обліку і контроль оренди основних засобів	63
Квятківська А. П.	
Застосування штучного інтелекту. Світовий досвід.....	67
Коломієць Д. В., Колодка А. В.	
Новітні методи вивчення англійської мови в Україні	72
Король А. А.	
Особливості перекладу антропонімів у дитячій художній літературі.....	75
Короткевич Є. Ю.	
Вплив стресу на харчову поведінку українців під час воєнного стану	79

4

					IA94.150BAK.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		86

Кравченко Д. М., науковий керівник Мороз О.І. Теоретичні аспекти проблеми дослідження творчої активності в юнацькому віці.....	86
Крикун В. Д. Технологія віртуальної реальності для розвитку м'яких навичок військовослужбовців	92
Лисенко О. Ю. Особливості доказування в кримінальному процесі під час воєнного стану	95
Локшин В. С. Формування фахової компетентності майбутніх педагогів професійного навчання в умовах комплексної діджиталізації суспільства	99
Локшин В. С. Професійна управлінська культура менеджерів СКС	102
Мірошниченко А. С., науковий керівник Бондар О. С. Історія розвитку приватного права та його становлення в Україні	115
Неділько А. О. Апаратні неінвазивні атаки та запобігання ним.....	121
Осіїчук К. П., науковий керівник Яцюрик А. О. Динаміка емоційно-ціннісного ставлення молодших школярів до навчання ..	126
Панченко І. Г. Комунікативний підхід до вивчення мови у закладах професійної (професійно- технічної) освіти	128
Пахольчук А. М., науковий керівник Мозговий О. М. Фінансовий компонент екосистем бізнесу	130
Радомська Г. Д., Бобицька Г. В., Каньовська Л. В. Діагностична цінність прокальцитоніну у лікуванні хронічного обструктивного захворювання легень	134
Саліхов М. М. Від автоматизованих до автономних транспортних засобів.....	136
Сидоренко А. В., науковий керівник Попов О. С. Сучасні тенденції розвитку транснаціональних корпорацій і їх вплив на світову економіку	143
Соколенко С. М. Довоєнний стан заробітної плати в Сумській області. Чому українці їхали працювати за кордон	148
Солонуха Д. Л. Місце банківського права в системі національного права.....	155
Степаненко А. О. Правова природа контекстуальних елементів у аспекті доказування воєнних злочинів в умовах збройного конфлікту	157
Стеценко В. А., науковий керівник Бондар О. С. Цивільне право як важливий елемент в системі правових галузей	163
Топчий В. І. Бачення обов'язків командира кавалерії в давніх Афінах у творі Ксенофонта Афінського «Гіппарх».....	166

САЛІХОВ Михайло Михайлович,

студент 4 курсу

Національного технічного університету України

«Київський політехнічний інститут імені Ігоря Сікорського»

м. Київ, Україна

ВІД АВТОМАТИЗОВАНИХ ДО АВТОНОМНИХ ТРАНСПОРТНИХ ЗАСОБІВ

Сьогодні транспортний сектор зазнає швидких перетворень, значною мірою через нові форми автоматизації водіння, котрі стали можливими завдяки розробкам у галузі інформаційно-комунікаційних технологій та робототехніці. «Автономні транспортні засоби, або безпілотні пасажирські автомобілі та вантажівки, коли вони набудуть широкого поширення в майбутньому, можуть призвести до величезних змін у моделях мобільності та поведінці» [1, с.164].

У літературі з питань автоматизації транспортних засобів вживається термін «самохідна техніка» під яким розуміють безпілотний наземний транспортний засіб (unmanned ground vehicle – UGV), котрий працює при контакті із землею та без присутності людини на борту [2]. UGV можуть використовуватися в багатьох випадках, коли участь людини-оператора може бути незручною, небезпечною або неможливою. UGV має набір датчиків для спостереження за навколишнім середовищем і здатний автономно приймати

136

					ІА94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		88

рішення щодо своїх дій, або передавати інформацію людині-оператору, якій знаходиться поза транспортним засобом. Поряд із UGV до категорії «самохідна техніка» відносять також «безпілотні літальні апарати» (unmanned aerial vehicles - UAV) та «безпілотні підводні апарати» (Unmanned underwater vehicles - UUV).

У документі ЄС до категорії самохідна техніка віднесли [3]:

Автоматизований транспортний засіб (Automated vehicle): автотранспортний засіб (автомобіль, вантажівка або автобус), оснащений технологіями, що допомагають водію, аби елементи завдання водіння могли бути передані до комп'ютерної системи;

Автономний транспортний засіб (Autonomous vehicle): повністю автоматизований транспортний засіб, оснащений технологіями, здатними виконувати всі функції керування без втручання людини.

Під'єднаний транспортний засіб (Connected vehicle): автомобіль, оснащений пристроями для зв'язку з іншими транспортними засобами або інфраструктура через Інтернет.

Національна Адміністрація Безпеки Дорожнього Руху США вважає транспортний засіб автономним, якщо він функціонує, не вимагаючи від водія безпосередньої участі у реалізації функцій водіння, включаючи «кермо, прискорення та гальмування» [4]. Крім того, водій цих транспортних засобів не повинен постійно стежити за дорогою під час самостійного водіння.

На думку автора роботи [5, с. 150], автономний транспортний засіб - це неофіційний термін, який використовується для опису автомобіля, обладнаного автоматизованою системою керування (automated driving system - ADS), який працює в автономному режимі, тобто автомобіль, що має високий рівень автоматизації керування. При цьому вчений звертає увагу [5, с. 149], що «автономний» є застарілим терміном, який тривалий час використовується спільнотами, котрі займаються ДіР у галузі робототехніки та штучного інтелекту, для позначення систем, що мають здатність і повноваження приймати рішення незалежно та самодостатньо. Згодом це

					ІА94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		89

трактування було розширено, аби охопити не лише ухвалення рішень, а й всю функціональність системи, в результаті чого «автономний» став синонімом поняття «автоматизований». Науковець вказує, що взаємозамінне використання цих понять затемнює питання про те, чи залежить так званий «автономний транспортний засіб» від зв'язку та/або кооперації із зовнішніми об'єктами для реалізації важливих функцій (таких як збирання, отримання, обробка даних). Деякі системи автоматизації водіння дійсно можуть бути автономними, якщо вони виконують усі свої функції незалежно та самодостатньо, але якщо вони залежать від зв'язку та/або кооперації із зовнішніми об'єктами, їх слід вважати взаємодіючими, а не автономними [5, с. 149]. І, як зазначають фахівці Товариства автомобільних інженерів (SAE), «деякі народні звичаї пов'язують автономію безпосередньо з повною автоматизацією водіння рівня 5, тоді як інші звичаї застосовують її до всіх рівнів автоматизації водіння, а в законодавстві деяких штатів визначено, що вона приблизно відповідає будь-якій ADS на рівні 3 або вище (або будь-якого транспортного засобу, оснащеного таким ADS)». [6, с. 34]

Дослідження праць з властивостей самохідної техніки, зокрема автомобілів, показало, що найбільш вживаними термінами (які використовують як синоніми) є: «автономний автомобіль» (autonomous car); «безпілотний автомобіль» (driverless car); «самокерований автомобіль» (self-driving car); «робот-автомобіль» (robotic car, robo-car). Оскільки компоненти таких автомобілів є як фізичними, так і віртуальними, вчені визначають «безпілотники» як «кіберфізичні системи» (cyber-physical systems).

Водночас, експерти SAE проводили опитування 257 респондентів щодо загальної назви зазначених вище машин, яке показало, що 44,5% вибрали «self-driving car» (самокерований, безпілотний автомобіль), оскільки саме цей термін використовувався Waymo LLC (раніш відому як Google Self-Driving Car Project) з того часу, як компанія представила свій проект кілька років тому. [6, с.164]. Виходячи з цього, у цій роботі автор використовується термін «безпілотний автомобіль» (driverless car – DLC).

					ІА94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		90

Рівень автономності DLC визначається масштабом автоматизації його функцій. Однією з основних передумов автономності DLC є запровадження так званої «Передової системи допомоги водію» (Advanced Driver Assistance Systems – ADAS). Поштовхом до розробки ADAS стала необхідність підвищення безпеки шляхом запобігання нещасним випадкам і зменшення навантаження на водія [7, 8]. ADAS – це пасивні та активні системи безпеки, призначені для мінімізації та усунення впливу людського фактору під час керування транспортним засобом. У системах ADAS використовуються передові технології, які допомагають водієві під час водіння, покращуючи його роботу. ADAS використовує комбінацію сенсорних технологій для сприйняття навколишнього середовища навколо автомобіля, а потім або надає інформацію водієві, або при потребі виконує необхідні завдання. Основні додатки ADAS, які мають важливе значення для безпеки, такі: Адаптивний круїз – контроль (ACC); Адаптивне переднє світло (AFL); Автоматичне аварійне гальмування (AEB); Виявлення сліпих зон (BSD); Попередження про перехресний рух (CTA); Система моніторингу водіїв (DMS); Попередження про лобове зіткнення (FCW); Інтелектуальна система допомоги при паркуванні (IPA); Попередження про вихід зі смуги руху (LDW); (ALC); Система виявлення пішоходів (PDS); Розпізнавання дорожніх знаків (RSR); Система контролю тиску в шинах (TPMS); Асистент руху у «пробці» (TJA), Система зв'язку 5G і V2X.

Тривалий час ADAS часто використовувався для позначення DLC. Проте представлені його функції стосуються саме систем автоматизованої допомоги водієві. Оскільки, як вже зазначалося, ці два терміни часто використовуються як взаємозамінні, було висунуто вимогу до автовиробників подавати ADAS як «підтримка водія», а не використовувати терміну «автономне водіння».

Для формалізації рівнів «автономності» DLC було розроблено відповідні стандарти SAE J3016 [6], за якими межа між ADAS і автономним водінням знаходиться між рівнем 2 і рівнем 3 (Рис. 1, Рис. 2).

					ІА94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		91



Рисунок 1. Межа між ADAS і автономним водінням. [9]

	SAE РІВЕНЬ 0	SAE РІВЕНЬ 1	SAE РІВЕНЬ 2	SAE РІВЕНЬ 3	SAE РІВЕНЬ 4	SAE РІВЕНЬ 5
Що повинен робити водій?	Ви керуєте автомобілем, навіть якщо нога не на педаль, а рука не на руль.			Ви ще керуєте автомобілем, якщо активовані функції автономного водіння.		
	Ви можете сидівати за електронними панелями, рулити, гальмувати або прискорюватись, якщо потрібно.			Якщо система буде новою водієм,	Електронні системи не вимагають від жодя керувати автомобілем самостійно.	
	Системи допомоги водія			Системи безпильного водіння		
Як вони працюють?	Панельки лише попереджують і надають короткотривалу допомогу.	Панельки допомагають керувати ABS прискорюватись/гальмувати.	Панельки допомагають керувати TA прискорюватись/гальмувати.	Система може самостійно керувати автомобілем лише за умови одностороннього виконання певних умов.		Система може керувати автомобілем за будь-яких умов.
Приклади систем	Автоматичне аварійне гальмування Попередження про смітани Попередження про поїздження смуги	Утримання на смугі ABS Адаптивний контроль круїз-контролю	Утримання на смугі TA Адаптивний круїз-контроль одночасно	Панельки під час руху у доріжних зонах.	Місцева безпильна поїздка. Можливо, бути відсутній руль або педалі.	Те саме, що і рівень 4, але можливість автономного керування зберігається.

Рисунок 2. Рівні автоматизації водіння відповідно до SAE J3016. [6]

Сьогодні автомобільні компанії забезпечують своїм виробам початковий рівень автономності. Прогнозується [10], що у 2025 р. близько 63% усіх автомобілів, що продаватимуться у всьому світі, матимуть автономію рівня 2 або вище. Продажі автомобілів рівня 2 та пілотних зразків рівня 3 становитимуть відповідно близько 12 та 4% від загального обсягу продажів автомобілів. Очікується, що DLC рівня 4 становитимуть 1% у 2025 р.,

поступово збільшуватимуть свою частку і до 2030 р. сягатимуть 5% світового ринку. Щодо DLC рівня 5, його дотепер не має жоден автомобіль. Досягнення такої автономності має багато техніко-технологічних проблем, зокрема в частині забезпечення водіння без керма та педалей, яке, за словами представників компанії Apple нині не може бути реалізовано сучасними технологіями (це власне кажучи й загальмувало проект Titan компанії Apple, спрямований на створення автомобіля рівня 5) [11].

З початку 2000-х рр. завдяки значному прогресу у галузі інформаційно-комунікаційних технологій та робототехніці DLC стають реальністю. Прототипи DLC подолали мільйони миль у тестовому режимі, а провідні високотехнологічні компанії вклали приголомшливі інвестиції у збільшення автоматизації водіння. Однак для досягнення повної автономії є багато як технічних, так і нетехнічних проблем, пов'язаних зі страхуванням, етичними та моральними особливостями використання DLC і відповідним законодавчим забезпеченням. Для вирішення цих проблем потрібні продумані рішення, які відповідають вимогам, нормам та інтересам ключових сейкхолдерів.

Список бібліографічних посилань:

1. Wee B., Milakis D., Thomopoulos N. Policy Implications of Autonomous Vehicles Elsevier Science. 2020.354 p.
2. Unmanned ground vehicle. URL: https://en.wikipedia.org/wiki/Unmanned_ground_vehicle.
3. Pillath S. Automated vehicles in the EU. URL: [https://www.europarl.europa.eu/RegData/etudes/BRIE/2016/573902/EPRS_BRI\(2016\)573902_EN.pdf](https://www.europarl.europa.eu/RegData/etudes/BRIE/2016/573902/EPRS_BRI(2016)573902_EN.pdf).
4. U.S. Department of Transportation Releases Policy on Automated Vehicle Development. URL: <https://www.transportation.gov/briefing-room/us-department-transportation-releases-policy-automated-vehicle-development>.
5. Young R. Critical Analysis of Prototype Autonomous Vehicle Crash Rates Six Scientific Studies from 2015-2018 SAE International. 2021. 252 p.

					ІА94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		93

6. Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles J3016_202104. URL: https://www.sae.org/standards/content/j3016_202104/.
7. Samantaray R. ADAS Sensor Data Handling in the World of Autonomous Mobility. SAE Technical Paper 2023-01-0993, 2023. <https://doi.org/10.4271/2023-01-0993>.
8. Yan Li, Hualiang Shi Advanced Driver Assistance Systems and Autonomous Vehicles: From Fundamentals to Applications Springer 2022. 627 p.
9. On ADAS, Autonomous Driving, and Vehicle Safety Technologies. URL: <https://www.rohm.com/blog/-/blog/id/8030373>.
10. Level 2-4 autonomous vehicle sales as a share of total vehicle sales in 2025 and 2030, by automation level. URL: <https://www.statista.com/statistics/1230101/level-2-autonomous-vehicle-sales-worldwide-as-a-share-of-total-vehicle-sales-by-automation-level/>.
11. Gurman M. Apple Scales Back Self-Driving Car and Delays Debut Until 2026. URL: <https://www.bloomberg.com/news/articles/2022-12-06/apple-scales-back-self-driving-car-and-delays-launch-until-2026>.

					ІА94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		94

ЛЬВІВСЬКИЙ НАУКОВИЙ ФОРУМ

МАТЕРІАЛИ

VIII МІЖНАРОДНОЇ НАУКОВО-ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ



КОНЦЕПТУАЛЬНІ ШЛЯХИ РОЗВИТКУ НАУКИ ТА ОСВІТИ

9 – 10 червня 2023 року

					ІА94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		95

ЗМІСТ

ДЕРЖАВНЕ УПРАВЛІННЯ	7
<i>Джурик Ю.А., Труш А.С.</i> ПЕРЕДУМОЙ ВИНИКНЕННЯ КОРПОРАТИВНИХ ВІДНОСИН	7
ЕКОНОМІЧНІ НАУКИ	12
<i>Іваненко О.О.</i> РОЗВИТОК БІРЖОВОЇ ДІЯЛЬНОСТІ В УКРАЇНІ	12
<i>Іванюк О.М.</i> ОРГАНІЗАЦІЯ ОБЛІКУ ДОХОДІВ І ВИТРАТ	14
<i>Качмар Г.Я.</i> БІЗНЕС В УКРАЇНІ ПІД ЧАС ВІЙНИ	16
<i>Коваленко Н., Циба О., Кудряцєв В., Шаніро Р.</i> УДОСКОНАЛЕННЯ МЕТОДИЧНИХ ПІДХОДІВ ДО ОЦІНКИ ЕФЕКТИВНОСТІ РОЗВИТКОМ ПІДПРИЄМСТВ	18
<i>Левницький О.О.</i> ОРГАНІЗАЦІЯ ФІНАНСОВОЇ РОБОТИ НА ПІДПРИЄМСТВАХ АПК	20
<i>Москвичов Б.І.</i> МЕТОДИ ДОСЛІДЖЕННЯ ПРОБЛЕМ ТА РЕГУЛЮВАННЯ НАДМІРНОГО ЗРОСТАННЯ ДЕРЖАВНОГО БОРГУ УКРАЇНИ	22
<i>Сургай Д.</i> ПЛАНУВАННЯ МАРКЕТИНГОВОЇ ДІЯЛЬНОСТІ ПІДПРИЄМСТВА В СУЧАСНИХ УМОВАХ	24
<i>Чигринєць С., Бойко Ю.</i> ФІНАНСОВЕ ЗАБЕЗПЕЧЕННЯ РОЗВИТКУ ПІДПРИЄМСТВ	26
ІСТОРИЧНІ НАУКИ	30
<i>Ляшенко Є.</i> РІЗНОМАНІТТЯ КУЛІНАРНИХ ТРАДИЦІЙ ПОКУТЯН В СВІТЛІ ПОВСЯКДЕННОГО ХАРЧУВАННЯ	30
МЕДИЧНІ НАУКИ	35
<i>Чемельова В.М.</i> РЕАБІЛІТАЦІЯ У ДОРОСЛИХ ПАЦІЄНТІВ ПІСЛЯ COVID-19 У ПІСЛЯ ГОСТРІЙ ФОРМІ ЛІКУВАННЯ ТЕРАПЕВТИЧНИМИ ВПРАВАМИ	35
МИСТЕЦТВОЗНАВСТВО	37
<i>Винокур А.М.</i> СТАНОВЛЕННЯ ЧОЛОВІЧОГО ОДЯГУ ВОЛИНІ	37
<i>Чайка М.В.</i> РІЗЬБЛЕННЯ ПО ДЕРЕВУ НА ТЕРЕНАХ УКРАЇНИ: МИНУЛЕ І СЬОГОДЕННЯ	40

НАЦІОНАЛЬНА БЕЗПЕКА.....	43
<i>Казіміров М.М.</i> РОЛЬ ВОГНЕВОЇ ПІДГОТОВКИ У ФОРМУВАНІ НАВИЧОК ПОВОДЖЕНІ ЗІ ЗБРОЄЮ ПОЛІЦЕЙСЬКИМ.....	43
<i>Савенко А.Б.</i> ЗАБЕЗПЕЧЕННЯ ПРАВОПОРЯДКУ В УМОВАХ ВОЄННОГО СТАНУ.....	45
<i>Ткачова Ю.В., Тимофеев В.П.</i> РОЛЬ ПОЛІЦЕЙСЬКИХ У ЗАБЕЗПЕЧЕННІ БЕЗПЕКИ ТА ПРАВОПОРЯДКУ ПІД ЧАС ВОЄННОГО СТАНУ: АНАЛІЗ ОСНОВНИХ ВИКЛИКІВ	47
ПЕДАГОГІЧНІ НАУКИ.....	49
<i>Богатирьова Г.А., Литвиненко Ж.Р.</i> ШЛЯХИ ФОРМУВАННЯ ПОЛКУЛЬТУРНОЇ КОМУНІКАТИВНОЇ КОМПЕТЕНТНОСТІ МАЙБУТНІХ ФАХІВЦІВ З ТУРИЗМУ.....	49
<i>Богатирьова Г.А., Митрофанова Ю.Д.</i> ФОРМУВАННЯ УМІНЬ ЕФЕКТИВНОЇ КОМУНІКАЦІЇ МАЙБУТНІХ ФАХІВЦІВ З ТУРИЗМУ В УМОВАХ КУЛЬТУРНОГО РІЗНОМАНІТТЯ ТУРИСТИЧНОЇ ДІЯЛЬНОСТІ.....	52
<i>Халтобіна О.У.</i> УМОВИ ТА ВИМОГИ СТВОРЕННЯ ЗДОРОВ'ЯЗБЕРЕЖУВАЛЬНОГО СЕРЕДОВИЩА В ЗАКЛАДІ ДОШКІЛЬНОЇ ОСВІТИ.....	55
<i>Шварц Н.В., Сабатовська-Фролкіна І.С., Тимошук Г.В.</i> ОСНОВНІ ПРИНЦИПИ ГЕЙМІФІКАЦІЇ В ОСВІТІ.....	57
ПОЛІТИЧНІ НАУКИ.....	59
<i>Мироненко В.В.</i> ОРГАНІЗАЦІЙНО-ІНСПЕКЦІЙНА ДІЯЛЬНІСТЬ МАГАТЕ З ПРОТИДІЇ ПОШИРЕННЮ РАКЕТНО-ЯДЕРНИХ ОЗБРОЄНЬ.....	59
ПСИХОЛОГІЧНІ НАУКИ.....	62
<i>Кісільова В.І.</i> ОСОБЛИВОСТІ ВПЛИВУ ВІЙНИ НА ПСИХІКУ МАЛОЛІТНІХ....	62
<i>Ткачова Ю.В.</i> ПСИХОЛОГІЧНІ РИЗИКИ ПРАЦІВНИКІВ НАЦІОНАЛЬНОЇ ПОЛІЦІЇ УКРАЇНИ В УМОВАХ ВОЄННОГО СТАНУ	64
ТЕХНІЧНІ НАУКИ.....	66
<i>Дитко Т.В., Джумеля К.М.</i> ПРОЄКТУВАННЯ ТА РОЗРОБКА ВЕБ-СЕРВІСУ ОБЛІКУ ОСОБИСТИХ ФІНАНСІВ.....	66
<i>Ланін А.В., Грінчук І.О.</i> БЕЗПЛОТНИЙ МІСЬКИЙ ТРАНСПОРТ ЯК СКЛАДОВА «РОЗУМНОГО МІСТА»	68

					ІА94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		97

<i>Мішко М.Ю., Вєждев В.В.</i> РОЗРОБКА ВЕБСЕРВІСУ ДОСТАВКИ СУШІ НА ОСНОВІ ТЕХНОЛОГІЇ REACT	71
<i>Саліхов М.М.</i> ARDUINO – ПЕРСПЕКТИВНИЙ ІНСТРУМЕНТ ПРОТИТОПУВАННЯ У РОБОТОТЕХНІЦІ	73
<i>Шпак О.І., Русин М.В.</i> РОЗРОБКА ВЕБ-СЕРВІСУ ТА МОБІЛЬНОГО ДОДАТКУ ДЛЯ БРОНЮВАННЯ КВИТКІВ НА МАСОВІ ЗАХОДИ	80
<i>Шпак О.І., Ходанич П.Я.</i> СИСТЕМА ІНФОРМУВАННЯ РИЗИКІВ ТЕХНОГЕННИХ ТА ПРИРОДНИХ КАТАСТРОФ	85
ФІЗИКО-МАТЕМАТИЧНІ НАУКИ.....	88
<i>Максимова С.І.</i> ЗВ'ЯЗОК ТЕОРІЇ ВІДНОСНОСТІ ТА НЕЕВКЛІДОВИХ ГЕОМЕТРІЙ	88
<i>Попович К.В., Ходзінська Н.Ю., Чеснік Н.М.</i> ФОРМУВАННЯ МАТЕМАТИЧНИХ КОМПЕТЕНТНОСТЕЙ У МАЙБУТНІХ ФАХІВЦІВ З КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ	90
ФІЗИЧНЕ ВИХОВАННЯ І СПОРТ	93
<i>Антонюк В.Є.</i> PECULIARITIES OF PHYSICAL TRAINING OF FEMALE POLICE OFFICERS	93
<i>Антонюк В.Є.</i> ФІЗИЧНА ПІДГОТОВКА - ЕФЕКТИВНИЙ ЗАСІБ ПІДВИЩЕННЯ ПРОФЕСІЙНОЇ МАЙСТЕРНОСТІ ПРАЦІВНИКІВ ПОЛІЦІЇ	96
ФІЛОЛОГІЧНІ НАУКИ.....	100
<i>Антонюк В.Є.</i> THE ROLE OF ENGLISH FOR LEGAL PROFESSIONS	100
<i>Баштанікова А.А.</i> РОЗВИТОК КОМПЕТЕНТНОСТЕЙ НА УРОКАХ УКРАЇНСЬКОЇ МОВИ В ПОЧАТКОВІЙ ШКОЛІ НУШ	101
<i>Подсевак К.С.</i> ЛЕКСИКО-СЕМАНТИЧНІ ТРАНСФОРМАЦІЇ В ПЕРЕКЛАДІ ФРАЗЕОЛОГІЗМІВ СПОРТИВНОГО МЕДІАДИСКУРСУ	106
<i>Шостак Н.С.</i> РОЗВИТОК ЗВ'ЯЗНОГО МОВЛЕННЯ МОЛОДШИХ ШКОЛЯРІВ НУШ НА УРОКАХ УКРАЇНСЬКОЇ МОВИ ТА ЧИТАННЯ	108
ЮРИДИЧНІ НАУКИ.....	112
<i>Галкін К.</i> КОНСТИТУЦІЙНІ ТА МІЖНАРОДНО-ПРАВОВІ МЕХАНІЗМИ ЗАБЕЗПЕЧЕННЯ ПРАВ ЛЮДИНИ В УМОВАХ ЗБРОЙНИХ КОНФЛІКТІВ	112
<i>Жовтобрюх Н.</i> МІЖНАРОДНА СИСТЕМА ЗАХИСТУ ПРАВ ЛЮДИНИ	114

					ІА94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		98

УДК 004.775:625.75

*Саліхов М.М.,
студент 4 курсу
Національного технічного університету України
«Київський політехнічний інститут імені Ігоря Сікорського»,
м. Київ, Україна
e-mail: actualhooman@gmail.com*

ARDUINO – ПЕРСПЕКТИВНИЙ ІНСТРУМЕНТ ПРОТИТОПУВАННЯ У РОБОТОТЕХНІЦІ

Мобільна робототехніка — це технологічна галузь та галузь досліджень, де за останні два десятиліття відбулися неймовірні технологічні прориви, включаючи безпілотні транспортні засоби. Водночас, у дослідженнях і розробках галузі робототехніки часто відчувається потреба у практичних інструментах інтеграції для реалізації цінних наукових здобутків. Робототехніки в кінцевому підсумку витрачають немало часу на реалізацію інженерних рішень для конкретної конфігурації обладнання, часто вимушені «винаходити велосипед» реалізуючи свої ідеї. З метою розв'язання цієї проблеми та спрощення процесу створення прототипів з'явилася низка різних мобільних роботизованих платформ, здатних підтримувати дослідницьку роботу. Одна з таких – платформа Arduino [1].

Массімо Банзі співрозробник Arduino зауважив, що мета створення Arduino у 2003 р. полягала у надання винахідникам і просто творчим людям без значного досвіду в програмуванні або апаратному забезпеченні інструментів, які б дозволили реалізовувати їхні ідеї зі створення інноваційних продуктів у галузі робототехніки [2]. Щодо назви, історія наступна. «Коли ми розробляли продукт з моїми співзасновниками, – казав Банзі, – я перебував у Ізраїлі. В Ізраїлі є багато речей, які називаються Arduino. Тому що Arduino був одним із королів Італії у 1000 році. Отже, нам потрібне було ім'я, і був крайній термін, і, по суті, сталося те, що я сказав: «Добре, знаєте що, назвемо його Arduino, як бар, куди я ходжу випити. І тому ми назвали його так тільки тому, що це було перша випадкова назва, яка не була зайнята. Я хотів щось невиразно унікальне, і тому я ніби облажався, думаючи, що ми завжди зможемо виправити це пізніше. У результаті вона залишилась Arduino» [2].

Як зазначено на офіційному сайті [3], Arduino - це електронна платформа з відкритим вихідним кодом, заснована на простому у використанні апаратному та програмному забезпеченні. Плати Arduino можуть зчитувати вхідні дані - світло на датчику, палець на кнопці або повідомлення в Твіттері – і перетворювати їх на вихідні дані – активувати двигун, включити світлодіод, публікувати будь що в Інтернеті. Для цього, використовуючи мову програмування Arduino (на основі Wiring) та програмне забезпечення Arduino (IDE) на основі Processing, слід лише відправити набір інструкцій мікроконтролеру на плату. Примітно, що плата не була розроблена виробником

					IA94.150BAK.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		99

мікросхем. Чому це важливо? Виробники мікросхем часто хочуть показати, чим відрізняється їхній інноваційний продукт, і тому додають дивні елементи, аби виділити його та показати «новизну». При цьому Arduino акцентує увагу на спільних рисах мікроконтролерів, а не на їхніх відмінностях, що означає наступне – Arduino є ідеальною платформою для початківців: все, що можна навчитися зробити на базі Arduino, може бути корисним у роботі з будь-яким іншим мікроконтролером. Слід зазначити, що багато плат, що є на ринку, історично надзвичайно складні, із безліччю додаткових частин, таких як LCDs, кнопки, LEDs, 7-сегментний дисплей тощо, котрі демонструють все, на що ці плати здатні; тоді як Arduino має мінімум. І якщо винахіднику потрібно більше, є можливість використовувати шилди – існують сотні шилдів Arduino від LCD до Wi-Fi, котрі можна додати, аби отримати додаткову функціональність.

Процес прототипування з Arduino доволі простий і складається з кількох етапів (незалежно від варіанта або типу проекту).

1. Ідея/визначення проблеми. На цьому етапі народжується ідея, втілення якої потребує реалізації проекту та створення прототипу.

2. Концептуалізація. Після того, як виникла ідея та визначено проблеми, які потрібно вирішити для її реалізації, будується концепція. Це спрощений вид проекту, який дозволяє відповісти на низку питань: Як саме це має працювати? Чи будуть необхідні обчислення для отримання остаточного результату? Який алгоритм реалізації проекту? та інші.

3. Прототипування. Прототип трансформує ідею (теорію) у специфікації реальної працюючої системи у вигляді створення зразка готового продукту. Сформульовані ідеї втілюються в апаратно-програмному продукті, які стають так званими альфа-класами, оскільки вони є першою версією проекту, який неможливо запустити доти, доки вони не будуть модифіковані для включення всіх необхідних функцій. Після включення всіх функцій прототип стає бета-версією, готового до тестування та подальшого розгортання для широкого використання.

У робототехніці Arduino в основному використовується для керування двигунами, датчиками та виконавчими механізмами. Двигуни використовуються для керування рухом робота та можуть керуватися сигналами PWM (широтно-імпульсна модуляція) Arduino. Датчики використовуються для виявлення змін у навколишньому середовищі та надання вхідних даних Arduino, який може використовувати цю інформацію для управління поведінкою робота. Приводи, такі як серводвигуни можуть керуватися Arduino для виконання певних дій або рухів.

Щоб краще зрозуміти, яку роль Arduino може відігравати у роботизованих системах, розглянемо переваги та обмеження [1, 4].

1. *Низька вартість та недорога платформа.* Безпосередньо плати Arduino відносно економічні, особливо в порівнянні з іншими платформами мікроконтролерів, такими як Raspberry Pi та Nanode, логічно, що більшість новаторів купують компоненти саме Arduino та збирають свої конструкції Standard DIY (Do It Yourself), тобто «Зроби сам». Найдешевший модуль Arduino, який можна зібрати вдома, коштує менше 50 дол.

2. *Кросплатформенність.* Arduino IDE може працювати у більшості операційних систем, включаючи Macintosh OSX, Windows та Linux, що є ключовою відмінністю від більшості інших платформ мікроконтролерів, котрі працюють лише у Windows.

3. *Енергоощадність.* Arduino вимагає небагато енергії, адже працює з найнижчою напругою, заощаджуючи витрати користувача.

4. *Швидкий процес прототипування.* Повне прототипування системи Arduino відбувається достатньо швидко, тоді як на альтернативних обчислювальних платформах у більшості випадків потрібно набагато більше часу для створення прототипу пристрою аналогічного типу, що й на базі Arduino.

5. *Просте та зрозуміле середовище програмування.* Arduino IDE – це середовище програмування платформи, яке набуло поширення через цього простоту використання та швидке освоєння (яке не вимагає великої практики навіть для новачків). Водночас це не спрощення чи примітивізація, а просунуті інструменти, котрі допомагають професіоналам виконувати більш складні функції. Оскільки середовище розробки Arduino IDE було побудовано на певних функціях середовища програмування Processing, робота з Arduino IDE досить не складна для тих користувачів, хто вже знає мову Processing, що, наприклад, полегшує роботу викладачів, котрі навчають студентів, які вже вивчали програмування в Processing.

6. *Відкритий вихідний код і програмне забезпечення, що розширюється.* Arduino IDE та супутнє програмне забезпечення мають повністю відкритий вихідний код. Це полегшує роботу досвідченим програмістам з додавання розширень та модифікацій, які зроблять платформу кращою. Мова програмування Arduino також багато в чому запозичена з C++ і може бути розширена з бібліотеки C++; технічні деталі мови також скопійовані з мови програмування AVR C, тому він приймає коди AVR-C, ніби вони були написані в Arduino IDE.

7. *Відкритий вихідний код і апаратне забезпечення, що розширюється.* Не лише програмне забезпечення Arduino має відкритий вихідний код, але й апаратне забезпечення, котре засноване на мікроконтролерах Atmel ATMEGA8 та ATMEGA168, також має відкритий вихідний код, а його модулі публікуються під ліцензією Creative Commons. Завдяки відкритому вихідному коду, досвідчені розробники схем можуть створити свою версію модуля, розширити або поліпшити його, тоді як недосвідчені користувачі можуть навіть зібрати макетну версію модуля. Це найчастіше допомагає їм краще зрозуміти, як працює модуль, а також заощадити гроші.

8. *Впровадження більш інноваційних технологій.* Коли Arduino прийшла на платформу мікроконтролерів, деякі інші платформи вже існували, але завдяки впровадженню інноваційних технологій, таких як висока швидкість обробки, Arduino змогла швидко стати домінуючою платформою та найкращою для професіоналів.

9. *Простота інтерфейсу.* Arduino є простим в інтерфейсі, адже має розширювані контакти, до яких можна підключити зовнішні модулі (такі як USB), що полегшує передачу ресурсів; також є доступним широкій спектр API-інтерфейсів або інтерфейсів прикладних програм.

					ІА94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		101

10. *Широкий набір датчиків.* Arduino поставляється з кількома датчиками, що надає йому переваги при виборі користувачів інструментів для прототипування систем, які включають той чи інший тип візуалізації.

11. *Величезна спільнота користувачів та ентузіастів-новаторів.* Arduino об'єднав поціновувачів робототехніки, які обмінюються досвідом, ноу-хау, демонструють свої проєкти, дають рекомендації.

12. *Підходить навіть для початківців (школярів та студентів).* Більшість функцій Arduino можна освоїти, переглянувши відеоролики в Інтернеті або поставивши запитання в онлайн-спільноті Arduino. Завдяки портативній платформі новачки можуть набути досвід, отримати нові навички та компетенції.

Водночас, поряд із зазначеними вище перевагами Arduino має низку обмежень.

1. *Обчислювальна потужність.* плати Arduino мають обмежену обчислювальну потужність проти іншими мікроконтролерами чи комп'ютерними системами. Це може обмежити здатність плати виконувати складні завдання та обчислення, особливо у додатках реального часу.

2. *Пам'ять.* Плати Arduino також мають обмежений обсяг пам'яті як оперативної, так і флеш-пам'яті. Це може бути обмеженням для проєктів, яким потрібний великий обсяг сховища даних, або для додатків, які потребують багато пам'яті для операцій.

3. *Вартість.* Попри те, що плати Arduino відносно дешеві, вартість додаткових компонентів, таких як датчики, приводи та пристрої зв'язку, може швидко зрости і зробити загальну вартість проєкту вищою.

4. *Швидкість.* Тактова частота плат Arduino обмежена, що може вплинути на продуктивність деяких програм реального часу, таких як високошвидкісний збір даних або керування двигуном.



Рис. 1 Деякі сфери використання Arduino [4].

Попри зазначені вище обмеження плати Arduino нині широко використовуються винахідниками та розробниками інноваційних продуктів для реалізації різних ідей та вирішення проблем у різних сферах діяльності, спектр яких настільки ж різноманітний, наскільки безмежні сфери застосування прототипів, створених за допомогою Arduino. Аналіз літератури показав, що Arduino – популярний вибір для прототипування рішень у таких сферах, як: проектування систем; додатки загального призначення; апаратний зв'язок; прототипування програмного забезпечення; домашня автоматизація та загальна автоматизація; сільське господарство; охорона здоров'я; гірничодобувна промисловість; енергія та захист навколишнього середовища; освіта тощо (див. рис. 1). Як показало авторське дослідження, широке застосування Arduino знайшло у створення прототипів автономних транспортних засобів (табл. 1).

Таблиця 1

Роботизовані автомобілі на базі Arduino

Продукт	Дослідження
	Мініатюрна модель автономного автомобіля з використанням Arduino UNO та Open CV [5]
	Розробка роботизованої системи уникнення перешкод на основі Arduino для безпілотного автомобіля [6]
	Розробка автономного автомобіля що керується за допомогою Arduino рукавички [7]
	Роботизований автомобіль із голосовим керуванням на базі Arduino [8]
	Платформа Robotic-agent для вбудовування програмних агентів з використанням плат Raspberry Pi та Arduino [9]

Джерело: складено автором.

Прикладом інновацій Arduino є розробки 2022 р [10]. Випущена наприкінці 2008 року версія 0012 Arduino IDE представила багато корисних функцій, які з тих пір стали основою Arduino IDE, включаючи базове підсвічування, бібліотеки та включення інструментарій avrdude для компіляції коду та завантаження його на приєднану плату. У версії 1.0 було оновлено тему, деякі вбудовані бібліотеки та спосіб звітування про хід компіляції. У вересні 2022 р. було анонсовано Arduino IDE 2.0 (версія 2.0 перейшла з бета-версії на стабільну та внесла значні зміни) [11], заснований на фреймворку Eclipse Theia (Arduino не просто перескочила з однієї IDE на основі Java на іншу, оскільки Theia заснована на тому ж базовому коді, що і надзвичайно популярний Visual Studio Code від Microsoft, що розширюється), а створила інновацію, яка дозволяє розпочати з потужного сучасного редактора та зосередитися на специфічних для функціях Arduino. На додаток до розширених функцій (завершення коду та підтримка), нова IDE безпосередньо інтегрується з Arduino Cloud, дозволяючи користувачам продовжити роботу з місця, де вони зупинилися, у веб-редакторі на основі браузера. Нова IDE заснована на потужному інтерфейсі командного рядка Arduino, що значно спрощує взаємодію базових бібліотек та управління залежностями. Інші покращення включають вбудоване налагодження, покращений послідовний монітор та плотер, які можна використовувати навіть одночасно. Також було створено новий інтерфейс командного рядка Arduino Cloud [10]. Оскільки пристрої Інтернету речей (IoT) вимагають керування в масштабі, щоб їх ініціалізувати, отримувати оновлення та надсилати дані, новий Arduino створив свій інструмент Arduino Cloud CLI. Завдяки цьому розробники можуть легко взаємодіяти з Cloud API програмним способом. Зрештою, мета полягає в тому, щоб почати з одного прототипу, відтворити його властивості на багатьох інших фізичних пристроях, а потім розширити їх із можливістю отримання майбутніх оновлень «по повітрю» (over-the-air - OTA), індивідуально або масово. (Оновлення «по повітрю» — це різні методи поширення нових версій програм, налаштування та оновлення ключів шифрування).

У 2022 р. Arduino запропонувала новинку в галузі програмованих логічних контролерів (ПЛК – галузеві комп'ютери, які були захищені для використання у виробництві, роботах або інших програмах, які вимагають високої надійності та відмовостійкості). У відповідь на зростання попиту на автоматизацію, зокрема ПЛК, було створено спеціальну IDE Arduino для ПЛК, яка дозволяє писати програми ПЛК на п'яти мовах, визначених стандартом IEC 61131-3.

Аналізуючи феномен популярності Arduino серед розробників, слід згадати, що ще у лютому 2011 р. Філіп Торрон у статті «Чому Ардуіно перемогла і чому вона тут, щоб залишитись» [12] обґрунтував, що ця платформа має великі перспективи, оскільки це «відмінне програмне забезпечення для декількох систем, безлічі бібліотек, драйверів, що працюють, просте, недороге і з відкритим вихідним кодом». Підтвердженням цього прогнозу є поширення нині відомих продуктів Arduino та створення нових.

В подальшому очікується, що у зв'язку з попитом, що зростає, на недорогі, зручні і настроювані роботизовані рішення, Arduino продовжуватиме відігравати важливу роль у майбутньому робототехніки. Крім того, досягнення в галузі технологій, таких як IoT та машинне навчання, ймовірно, сприятимуть подальшому зростанню та інноваціям у використанні Arduino для робототехніки. Підсумовуючи, можна сказати, що Arduino у робототехніці матиме динамічну реалізацію, адже відкриває широкі можливості для інноваторів та інновацій.

						IA94.150BAK.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат			104

СПИСОК ЛІТЕРАТУРИ:

1. Bräuml T. Embedded Robotics: From Mobile Robots to Autonomous Vehicles with Raspberry Pi and Arduino Springer Nature, 2022. 519 p.
2. Q&A: Arduino founder Massimo Banzi. URL: <https://www.helsinki.fi/business/22390-q-a-arduino-founder-massimo-banzi.html>
3. What is Arduino? URL: <https://www.arduino.cc/en/Guide/Introduction>
4. Kondaveeti H. K. et al. A systematic literature review on prototyping with Arduino: Applications, challenges, advantages, and limitations. Computer Science Review Volume 40, May 2021, 100364. <https://doi.org/10.1016/j.cosrev.2021.100364>
5. Walke T. et al. Miniature Model of Autonomous Vehicle Using Arduino UNO and Open CV. Journal For Research in Applied Science and Engineering Technology. Volume 10 Issue IV Apr 2022 p.3294-3305 <https://doi.org/10.22214/ijraset.2022.42074>
6. Alli K. S. et al., "Development of an arduino-based obstacle avoidance robotic system for an unmanned vehicle," ARPJ Journal of Engineering and Applied Sciences, vol. 13, no. 3, 2018. P. 886-892 http://eprints.abuad.edu.ng/744/1/jeas_0218_6751.pdf
7. Tazhkenov R. Et al. Development of Arduino glove-based autonomous car. 2017 21st International Conference on System Theory, Control and Computing (ICSTCC), Sinaia, Romania, 2017, pp. 212-216, DOI: 10.1109/ICSTCC.2017.8107036
8. Saravanan M. et al. Arduino Based Voice Controlled Robot Vehicle. IOP Conf. Series: Materials Science and Engineering 993 (2020) 012125. URL: <https://iopscience.iop.org/article/10.1088/1757-899X/993/1/012125/pdf>
9. Lazarin N.M., Pantoja, C.E. A robotic-agent platform for embedding software agents using raspberry pi and arduino boards. In: 9th Software Agents, Environments and Applications School (2015). URL: https://www.researchgate.net/publication/277403727_A_Robotic-agent_Platform_for_Embedding_Software_Agents_Using_Raspberry_Pi_and_Arduino_Boards
10. Arduino Open Source Report 2022. URL: <https://content.arduino.cc/assets/Arduino%20Open%20Source%20Report%202022.2.pdf>
11. Halfacree G. Arduino IDE 2.0 Leaves Beta, Brings Shiny New Features to the Masses as the Default IDE. URL: <https://www.hackster.io/news/arduino-ide-2-0-leaves-beta-brings-shiny-new-features-to-the-masses-as-the-default-ide-d4f2211ae94b>
12. Torrone P. Why the Arduino Won and Why It's Here to Stay. URL: <https://makezine.com/article/technology/arduino/why-the-arduino-won-and-why-its-here-to-stay/>

					IA94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		105

Додаток В

Імплементація програми системи навігації мобільного робота

Після аналізу вимог та проектування робота, можна переходити до реалізації програмного коду для Arduino. У цьому розділі детально описується структура програми та основні функціональні блоки, які були реалізовані.

Введення. В цьому розділі буде пояснюватися загальна структура програми та її організація. У цьому контексті, Arduino Uno використовується як основний мікроконтролер для управління роботом. Він підключається до різних компонентів робота, таких як мотори, сенсори, сервоприводи та інші пристрої.

Опис основних функціональних блоків. У програмі для робота використовуються дві бібліотеки: AFMotor і Servo. Розглянемо їх як окремі функціональні блоки:

Бібліотека AFMotor - надає функціональність для керування DC-моторами та кроковими двигунами через моторний щиток на базі чіпа L293D.

Ця бібліотека надає змогу контролювати швидкість, напрямок та зупинку моторів нашого робота. Нижче наведений приклад коду для ініціалізації об'єкту AFMotor і керування двигуном:

```
#include <AFMotor.h> //Імпорт бібліотеки для керування моторним шилдом
AF_DCMotor motor(1); // Створення об'єкту для першого мотора

void setup() {
  // Ініціалізація моторного щитка
  motor.setSpeed(255); // Встановлення швидкості мотора
}

void loop() {
  motor.run(FORWARD); // Рух уперед
  delay(1000);
}
```

					ІА94.150БАК.004 ПЗ	Арк.
						106
Зм.	Арк.	№ докум.	Підпис	Дат		


```
motor.run(RELEASE); // Зупинка мотора
delay(1000);
}
```

У цьому прикладі використано об'єкт AF_DCMotor для управління першим мотором (підключеного до піна M1 на моторному щитку). За допомогою методів setSpeed та run ми встановлюємо швидкість та напрямок руху мотора.

Бібліотека Servo має низку особливостей. Бібліотека Servo дозволяє керувати сервоприводами, які використовуються для точного кутового позиціонування. За допомогою цієї бібліотеки ми можемо встановлювати кут повороту сервопривода та виконувати плавні рухи. Ось приклад коду для керування сервоприводом:

```
#include <Servo.h>
Servo servoMotor; // Створення об'єкту для сервоприводу
void setup() {
  // Підключення сервоприводу до піна 9
  servoMotor.attach(9);
}
void loop() {
  servoMotor.write(90); // Встановлення кута повороту 90 градусів
  delay(1000);
  servoMotor.write(0); // Встановлення кута повороту 0 градусів
  delay(1000);
}
```

В цьому прикладі ми використовуємо об'єкт Servo для керування сервоприводом, підключеного до піна 9 на Arduino. За допомогою методу write ми встановлюємо кут повороту сервопривода.

Обидві бібліотеки дуже корисні для керування рухом та позиціонування різних компонентів робота, і їх використання в програмі Arduino дозволяє досягти потрібної функціональності та керованості роботом.

Нижче подано методологію написання програми.

I. Імпорт бібліотек використаних в роботі.

					ІА94.150БАК.004 ПЗ	Арк.
						107
Зм.	Арк.	№ докум.	Підпис	Дат		

Крок 1. По-перше, код має включати всі бібліотеки, необхідні для роботи.

Код програми:

```
#include <AFMotor.h> //Імпорт бібліотеки для керування  
моторним шилдом  
#include <Servo.h> //Імпорт бібліотеки для керування  
сервоприводом
```

Крок 2: Створення об'єктів для керування окремими двигунами та сервоприводом. У цьому блоку коду створюються об'єкти для керування кожним окремим двигуном та сервоприводом. Кожен об'єкт відповідає конкретному мотору або сервоприводу із підключених компонентів до Arduino.

Код програми:

```
AF_DCMotor RBMotor(1); //Створюються об'єкти для керування  
кожним окремим двигуном  
AF_DCMotor RFMotor(2);  
AF_DCMotor LFMotor(3);  
AF_DCMotor LBMotor(4);  
Servo servoWatch; //Створюється об'єкт для керування  
сервоприводом
```

Крок 3: Встановлення параметрів системи. Цей блок коду визначає параметри, які використовуються для налаштування системи руху робота.

Параметри ультразвукового датчика:

- byte TRIG_PIN = 2; - призначення піна для вихідного сигналу (тригер) ультразвукового датчика.
- byte ECHO_PIN = 13; - призначення піна для вхідного сигналу (ехо) ультразвукового датчика.
- byte distWatch = 150; - встановлення максимальної відстані, на яку датчик реагує. Об'єкти, що знаходяться далі, ігноруються.
- byte rangeStop = 50; - встановлення мінімальної відстані до об'єкту, при якій рух зупиняється (у сантиметрах).

					ІА94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		108

- float timeEcho = 2*(distWatch+10)/100/340*1000000; - обчислення максимального часу очікування зворотного сигналу. Цей час використовується для визначення, коли датчик вважає, що немає перешкоди.

Параметри двигуна наступні:

- byte MAX_SPEED = 100; - встановлення максимальної швидкості двигуна.
- int OFFSET_SPEED = 10; - коефіцієнт, який враховує різницю у потужності між двигунами по різних сторонах.

Параметри повороту:

- TURN_SPEED = 50; - сума, яка додається до швидкості двигуна під час повороту. Це дозволяє роботу повертатії.

Код програми:

```
byte TRIG_PIN = 2; //Призначення пінів ультразвукового датчика
byte ECHO_PIN = 13;
byte distWatch = 150; //Максимальна відстань, на яку датчик реагує(об'єкти що знаходяться далі ігноруються)
byte rangeStop = 50; //Мінімальна відстань до об'єкту, при якій рух зупиняється (в сантиметрах)
float timeEcho = 2*(distWatch+10)/100/340*1000000; //Максимальний час очікування зворотного сигналу

byte MAX_SPEED = 100; //Максимальна швидкість двигуна
int OFFSET_SPEED = 10; //Коефіцієнт для врахування того, що одна сторона є більш потужною
int TURN_SPEED = 50; //Сума, яка додається до швидкості двигуна під час повороту
```

Крок 4: Налаштування і початкова ініціалізація.

					ІА94.150БАК.004 ПЗ	Арк.
						109
Зм.	Арк.	№ докум.	Підпис	Дат		

У цьому блоку коду виконуються деякі початкові налаштування і ініціалізація компонентів системи. Цей код виконується один раз при запуску програми (у функції `setup()`).

Налаштування швидкостей двигунів: `rightBack.setSpeed(motorSpeed);` - встановлює швидкість для правого заднього двигуна.

`rightFront.setSpeed(motorSpeed);` - встановлює швидкість для правого переднього двигуна.

`leftFront.setSpeed(motorSpeed+motorOffset);` - встановлює швидкість для лівого переднього двигуна з додаванням коефіцієнта `motorOffset`.

`leftBack.setSpeed(motorSpeed+motorOffset);` - встановлює швидкість для лівого заднього двигуна з додаванням коефіцієнта `motorOffset`.

Зупинка всіх моторів:

`RBMotor.run(RELEASE);` - забезпечує зупинку правого заднього мотора.

`RFMotor.run(RELEASE);` - забезпечує зупинку правого переднього мотора.

`LFMotor.run(RELEASE);` - забезпечує зупинку лівого переднього мотора.

`LBMotor.run(RELEASE);` - забезпечує зупинку лівого заднього мотора.

Ініціалізація сервоприводу:

`servoWrite(90);` - призначає пін 10 для сервоприводу, що дозволяє керувати його положенням.

Налаштування пінів ультразвукового датчика:

`pinMode(TRIG_PIN,OUTPUT);` - встановлює режим вихідного піна `trig` у режим "OUTPUT" (вихідний).

`pinMode(ECHO_PIN,INPUT);` - встановлює режим вхідного піна `echo` у режим "INPUT" (вхідний).

Крок 5: Основний цикл програми.

У цьому кроці я описую функцію `loop()`, яка є основним циклом програми. Цей код виконується безперервно після ініціалізації (у функції `setup()`).

У циклі `loop()` зазвичай розміщується основний алгоритм керування роботом. Залежно від контексту програми, цей алгоритм може включати рух, взаємодію з датчиками, прийом команд зовнішнього контролера тощо.

					ІА94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		110

В цьому циклі можуть бути використані різні команди, функції та умовні оператори для керування роботом залежно від поставленої задачі. Цей код повторюється безкінечно, доки програма не завершиться або не буде вручну зупинений.

Крок 6: Виконання основного алгоритму керування в циклі loop().

У цьому кроці виконується основна логіка керування роботом на основі отриманих відстаней та виконання необхідних дій.

Блок коду 1: Перевірка передньої відстані

```
servoWatch.write(90);           //Встановлення сервоприводу на погляд  
прямо вперед  
delay(750);  
int distance = getRange();      //Перевірка, що попереду немає предметів  
if(distance >= rangeStop)      //Якщо в межах шляху немає предметів -  
рух вперед  
{  
    movementForward();  
}
```

Блок коду 2: Зупинка руху

```
while(distance >= rangeStop)    //Продовження перевірки відстані до  
об'єкту, поки вона не стане менше мінімальної відстані зупинки  
{  
    distance = getRange();  
    delay(250);  
}  
stopMove();                     //Зупинка двигунів
```

Блок коду 3: Визначення напрямку повороту

```
int directionTurn = checkDir(); //Перевірка відстані до об'єктів зліва і справа  
та отримання інструкції щодо повороту  
Serial.print(directionTurn);  
switch (directionTurn)          //Поворот ліворуч, праворуч або здійснити
```

					ІА94.150БАК.004 ПЗ	Арк.
						111
Зм.	Арк.	№ докум.	Підпис	Дат		

розворот залежно від інструкції

```
{
case 0:                                //Поворот ліворуч
    turnLeft (400);
    break;
case 1:                                //Розворот
    turnLeft (700);
    break;
case 2:                                //Поворот праворуч
    turnRight (400);
    break;
}
```

- Функція accelerate(): Прискорення двигунів від 0 до повної швидкості.
- Функція decelerate(): Уповільнення двигунів від повної швидкості до нуля.
- Функція movementForward(): Встановлення всіх двигунів на рух вперед.
- Функція stopMove(): Встановлення всіх двигунів на зупинку.
- Функція turnLeft(): Встановлення двигунів на поворот ліворуч на вказаний час і зупинка їх після повороту.
- Функція turnRight(): Встановлення двигунів на поворот праворуч на вказаний час і зупинка їх після повороту.
- Функція getRange(): Вимірювання відстані до перешкоди за допомогою ультразвукового датчика.
- Функція checkDir(): Перевірка відстаней до об'єктів зліва і справа, прийняття рішення щодо напрямку повороту.

Блок коду 4: Функція accelerate()

```
void accelerate()                        //Функція для прискорення двигунів від 0 до
повної швидкості
{
for (int i=0; i<MAX_SPEED; i++)         //Цикл від 0 до повної швидкості
{
```

					ІА94.150БАК.004 ПЗ	Арк.
						112
Зм.	Арк.	№ докум.	Підпис	Дат		

```

    RBMotor.setSpeed(i);                //Встановлення двигунів на поточну
швидкість циклу
    RFMotor.setSpeed(i);
    LFMotor.setSpeed(i+OFFSET_SPEED);
    LBMotor.setSpeed(i+OFFSET_SPEED);
    delay(10);
}
}

```

Блок коду 5: Функція decelerate()

```

void decelerate()                        //Функція уповільнення двигунів від повної
швидкості до нуля
{
    for (int i=MAX_SPEED; i!=0; i--)     //Цикл від повної швидкості до 0
    {
        RBMotor.setSpeed(i);           //Встановлення двигунів на поточну
швидкість циклу
        RFMotor.setSpeed(i);
        LFMotor.setSpeed(i+OFFSET_SPEED);
        LBMotor.setSpeed(i+OFFSET_SPEED);
        delay(10);
    }
}

```

Блок коду 6: Функція movementForward()

```

void movementForward()                  //Встановлення всіх двигунів на рух
вперед
{
    RBMotor.run(FORWARD);
    RFMotor.run(FORWARD);
    LFMotor.run(FORWARD);
}

```

					IA94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		113

```
LBMotor.run(FORWARD);  
}
```

Блок коду 7: Функція stopMove()

```
void stopMove() //Встановлення всіх двигунів на зупинку  
{  
  RBMotor.run(RELEASE);  
  RFMotor.run(RELEASE);  
  LFMotor.run(RELEASE);  
  LBMotor.run(RELEASE);  
}
```

Блок коду 8: Функція turnLeft()

```
void turnLeft(int duration) //Встановлення двигунів на поворот  
ліворуч на вказаний час і зупинити їх  
{  
  RBMotor.setSpeed(MAX_SPEED+TURN_SPEED); //Встановлення всіх  
двигунів на задану швидкість  
  RFMotor.setSpeed(MAX_SPEED+TURN_SPEED);  
  LFMotor.setSpeed(MAX_SPEED+OFFSET_SPEED+TURN_SPEED);  
  LBMotor.setSpeed(MAX_SPEED+OFFSET_SPEED+TURN_SPEED);  
  RBMotor.run(FORWARD);  
  RFMotor.run(FORWARD);  
  LFMotor.run(BACKWARD);  
  LBMotor.run(BACKWARD);  
  delay(duration);  
  RBMotor.setSpeed(MAX_SPEED); //Встановлення всіх двигунів на  
задану швидкість  
  RFMotor.setSpeed(MAX_SPEED);  
  LFMotor.setSpeed(MAX_SPEED+OFFSET_SPEED);
```

					ІА94.150БАК.004 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дат		114


```

LBMotor.setSpeed(MAX_SPEED+OFFSET_SPEED);
RBMotor.run(RELEASE);
RFMotor.run(RELEASE);
LFMotor.run(RELEASE);
LBMotor.run(RELEASE);
}

```

Блок коду 9: Функція turnRight()

```

void turnRight(int duration) //Встановлення двигунів на поворот
праворуч на вказаний час і зупинити їх
{
  RBMotor.setSpeed(MAX_SPEED+TURN_SPEED); //Встановлення всіх
двигунів на задану швидкість
  RFMotor.setSpeed(MAX_SPEED+TURN_SPEED);
  LFMotor.setSpeed(MAX_SPEED+OFFSET_SPEED+TURN_SPEED);
  LBMotor.setSpeed(MAX_SPEED+OFFSET_SPEED+TURN_SPEED);
  RBMotor.run(BACKWARD);
  RFMotor.run(BACKWARD);
  LFMotor.run(FORWARD);
  LBMotor.run(FORWARD);
  delay(duration);
  RBMotor.setSpeed(MAX_SPEED); //Встановлення всіх двигунів на
задану швидкість
  RFMotor.setSpeed(MAX_SPEED);
  LFMotor.setSpeed(MAX_SPEED+OFFSET_SPEED);
  LBMotor.setSpeed(MAX_SPEED+OFFSET_SPEED);
  RBMotor.run(RELEASE);
  RFMotor.run(RELEASE);
  LFMotor.run(RELEASE);
}

```

					ІА94.150БАК.004 ПЗ	Арк.
						115
Зм.	Арк.	№ докум.	Підпис	Дат		

```
LBMotor.run(RELEASE);  
}
```

Блок коду 10: Функція `getRange()`

Вимірює відстань до об'єкта за допомогою ультразвукового сенсора.

```
int getRange()                //Вимір відстані до предмета  
{  
    unsigned long pulseDuration;    //Створення змінної для зберігання  
    часу шляху імпульсу  
    int distance;                //Створення змінної для зберігання обчисленої  
    відстані  
    digitalWrite(TRIG_PIN, HIGH);    //Генерація імпульсу тривалістю 10  
    мікросекунд  
    delayMicroseconds(10);  
    digitalWrite(TRIG_PIN, LOW);  
    pulseDuration = pulseIn(ECHO_PIN, HIGH, timeEcho);    //Вимір часу  
    повернення імпульсу  
    distance = (float)pulseDuration * 340 / 2 / 10000;    //Обчислення відстані до об'єкта  
    на основі часу імпульсу  
    return distance;  
}
```

У цій функції спочатку генерується короткий імпульс на пін `trig` для запуску ультразвукового сенсора. Потім вимірюється час `pulseTime`, який потрібен для повернення ехо-сигналу від об'єкта до сенсора. На основі цього часу обчислюється відстань до об'єкта з урахуванням швидкості звуку. Отримане значення відстані повертається як результат функції.

Ця функція допомагає роботу визначати відстань до перешкод і приймати рішення про подальші дії на основі цих вимірів.

					ІА94.150БАК.004 ПЗ	Арк.
						116
Зм.	Арк.	№ докум.	Підпис	Дат		

Блок коду 11: Функція checkDir ()

Перевіряє відстані в лівому та правому напрямках, щоб визначити, у якому напрямку повернути робота.

```
int checkDir() //Перевірка напрямку ліворуч і праворуч,
прийняти рішення, у який бік повернути
{
  int rangeArray [2] = {0,0}; //Відстань ліворуч і праворуч
  int directionTurn = 1; //Напрямок повороту, 0 ліворуч, 1
розворот, 2 праворуч
  servoWatch.write(180); //Поворот серво, щоб подивитися
ліворуч
  delay(500);
  rangeArray [0] = getRange(); //Отримати відстань до об'єкта
зліва
  servoWatch.write(0); //Поворот серво, щоб подивитися
праворуч
  delay(1000);
  rangeArray [1] = getRange(); //Отримати відстань до об'єкта
справа
  if (rangeArray[0]>=200 && rangeArray[1]>=200) //Якщо обидва
напрямки вільні - поворот ліворуч
    directionTurn = 0;
  else if (rangeArray[0]<=rangeStop && rangeArray[1]<=rangeStop) //Якщо обидва
напрямки заблоковані, розворот
    directionTurn = 1;
  else if (rangeArray[0]>=rangeArray[1]) //Якщо ліворуч більше місця,
поворот ліворуч
    directionTurn = 0;
  else if (rangeArray[0]<rangeArray[1]) //Якщо праворуч більше
місця, поворот праворуч
```

```
directionTurn = 2;
return directionTurn;
}
```

У цій функції спочатку встановлюється сервопривод на позицію, щоб подивитися в лівий напрямок. Потім вимірюється відстань до об'єкта в лівому напрямку за допомогою функції `getDistance()` і зберігається в `rangeArray[0]`.

Потім сервопривод повертається в позицію праворуч, щоб подивитися в правий напрямок. Після цього вимірюється відстань до об'єкта в правому напрямку за допомогою функції `getRange()` і зберігається в `rangeArray[1]`.

Після цього виконується порівняння отриманих відстаней. Якщо обидва напрямки мають відстань більше або рівну 200, то поворот здійснюється ліворуч (`directionTurn = 0`). Якщо обидва напрямки мають відстань менше або рівну `rangeStop`, то здійснюється розворот (`directionTurn = 1`). Якщо відстань зліва більше, ніж відстань справа, то поворот виконується ліворуч (`directionTurn = 0`). Якщо відстань зліва менше, ніж відстань справа, то поворот виконується праворуч (`turnDir = 2`).

На основі цих перевірок визначається напрямок повороту `directionTurn`, який повертається як результат функції.