

Міністерство освіти і науки України
Державний заклад
«Луганський національний університет імені Тараса Шевченка»

Навчально-науковий інститут математики та інформаційних технологій

Кафедра інформаційних технологій та систем

Голод Данило Олегович

**ПРОГРАМНО-АПАРАТНІ ЗАСОБИ КЕРУВАННЯ МОДЕЛЮ
АВТОМОБІЛЯ**

кваліфікаційна робота

здобувача вищої освіти першого (бакалаврського) рівня

освітньої програми «Комп'ютерна інженерія»

за спеціальністю 123 Комп'ютерна інженерія

Особистий підпис _____ Данило ГОЛОД,

Науковий керівник _____ Володимир ДОНЧЕНКО,
старший викладач
кафедри інформаційних технологій
та систем

Завідувач кафедри _____ Микола СЕМЕНОВ,
кандидат педагогічних наук, доцент
кафедри інформаційних технологій
та систем

Полтава – 2025

Міністерство освіти і науки України	
Державний заклад «Луганський національний університет імені Тараса Шевченка»	
Факультет (інститут)	Навчально-науковий інститут математики та інформаційних технологій
	(повна назва)
Кафедра	Інформаційних технологій та систем
	(повна назва)
Освітній ступень	Бакалавр
	(код, назва)
Напрямок підготовки	123 Комп'ютерна інженерія
	(код, назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри ІТС

М.А. Семенов

(підпис)

(ініціали, прізвище)

“ ” 2025 р.

**ЗАВДАННЯ
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТУ**

Голоду Данилу Олеговичу

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) **Програмно-апаратні засоби керування моделлю автомобіля**

Керівник кваліфікаційної роботи **Донченко В.Ю. старший викладач**

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджена наказом по університету

від

2. Строк подання студентом проекту (роботи)

3. Вихідні дані до роботи (проекту)

Розробка програмно-апаратного

комплексу на базі Arduino Uno для керування рухомою моделлю автомобіля.

(визначаються кількісні або (та) якісні показники, яким повинен відповідати об'єкт розробки)

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) **НАУКОВО-ДОСЛІДНА ЧАСТИНА. ПРОЕКТУВАННЯ СИСТЕМИ КЕРУВАННЯ. РЕАЛІЗАЦІЯ ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ. ТЕСТУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ**

(визначаються назви розділів або (та) перелік питань, які повинні увійти до тексту ПЗ)

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

6. Консультанти розділів проекту/роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання „_____” _____ 2025р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
	Вибір теми роботи, вивчення наукової літератури, затвердження теми та керівника.	До 15 жовтня	
	Аналіз літературних джерел за темою роботи. Розробка та апробація методики дослідно-експериментальної роботи. Подання структури теоретичної частини роботи та плану експериментальних досліджень.	Другий тиждень листопада (10 листопада)	
	Робота над теоретичною частиною. Подання теоретичної частини роботи для першого читання науковим керівником.	До 15 грудня	
	Усунення зауважень, урахування рекомендацій наукового керівника. Подання теоретичної частини роботи на друге читання.	До 28 січня	
	Проведення експериментальної роботи. Поетапний аналіз та обговорення її результатів. Перевірка стану виконання роботи.	Перший тиждень березня	
	Урахування рекомендацій наукового керівника, усунення недоліків, підготовка варіанта роботи до передзахисту. Розробка презентації.	До 31 березня	
	Попередній захист роботи на кафедрі	квітень	
	Доопрацювання роботи з урахуванням рекомендацій після передзахисту. Подання роботи науковому керівникові та рецензентові на підготовку відгуку та рецензії	За 10 днів до державної атестації	
	Подання на кафедру остаточного варіанта роботи, переплетеного та підписаного автором, науковим керівником і рецензентом.	За 5 днів до державної атестації	

Студент

підпис

Д. О. Голод

(ініціали, прізвище)

Керівник проекту (роботи)

підпис

В. Ю. Донченко

(ініціали, прізвище)

АНОТАЦІЯ

Голод Д. О.

Тема: Програмно-апаратні засоби керування моделлю автомобіля.

Спеціальність: 123 «Комп'ютерна інженерія»

Установа: ЛНУ імені Тараса Шевченка, 2025 р.

Бакалаврська робота містить: 60 с., 12 рис., 2 табл., 20 джерел.

Об'єкт дослідження - процес керування моделлю автомобіля.

Предмет дослідження - системи програмно-апаратного керування мобільними моделями.

Мета роботи - розробка програмно-апаратного комплексу на базі Arduino Uno для керування рухомою моделлю автомобіля.

Результати роботи.

Створено прототип програмно-апаратного засобу керування моделлю автомобіля з графічним інтерфейсом для керування.

Ключові слова. ARDUINO UNO.

ABSTRACT

Holod Danylo

Theme: Car model control software and hardware.

Speciality: 123 "Computer Engineering"

Institution: Luhansk Taras Shevchenko National University (LTSNU), 2025.

Diploma work contains: 60 pages, 12 Fig., 2 Table, 20 source.

A research object is the process of controlling a car model.

An aim of work is hardware-software systems for mobile model control.

Job performances. A prototype of a software and hardware tool for controlling a car model with a graphical interface for control has been created.

Keywords. ARDUINO UNO.

ІТС.4КІ-МС.0125.01-ВП
ПРОГРАМНО-АПАРАТНІ ЗАСОБИ КЕРУВАННЯ МОДЕЛЮ
АВТОМОБІЛЯ

[illegible]

					ІТС.4КІ-МС.0125.03-ВП						
Змн.	Арк.	№ докум.	Підпис	Дата	Відомість проєкту	Літ.	Арк.	Акрушів			
Розроб.		Голод Д.О.									
Керівник		Донченко В.Ю.					1	1			
Реценз.		Козуб Ю.Г.				ЛНУ Кафедра ІТС, Гр.4КІ-МС					
Н. Контр.											
Зав. каф.		Семенов М.А.									

Міністерство освіти і науки України
Державний заклад «Луганський національний університет
імені Тараса Шевченка»

Факультет (інститут)

Навчально-науковий інститут
математики та інформаційних технологій
(повна назва)

Кафедра

Інформаційних технологій та систем
(повна назва)
(код, назва)

ТЕХНІЧНЕ ЗАВДАННЯ
на виконання програмної розробки (ПР):
" ПРОГРАМНО-АПАРАТНІ ЗАСОБИ КЕРУВАННЯ МОДЕЛЮ
АВТОМОБІЛЯ"

ІТС.4КІ-МС.0125.02-ТЗ

ПОГОДЖЕНО
Керівник кваліфікаційної роботи

Донченко В.Ю.

“ _____ ” 2025р

ВИКОНАВЕЦЬ
Студент групи 4КІ-МС

Голод Д. О.

“ _____ ” 2025р

Полтава – 2025

ЗМІСТ

ВСТУП.....	9
1. ПРИЗНАЧЕННЯ ПРОДУКЦІЇ.....	9
2. ТЕХНІЧНІ ХАРАКТЕРИСТИКИ	9
3. АЛГОРИТМ РОБОТИ ПРИСТРОЮ.....	9
4. ТЕХНІКО-ЕКОНОМІЧНІ ВИМОГИ ДО КІНЦЕВОГО ПРОДУКТУ	10
5. ВИМОГИ ДО МАТЕРІАЛІВ ТА КОМПЛЕКТУЮЧИХ	10
6. ЕТАПИ ВИКОНАННЯ ПР.....	10
7. ПРИЙМАННЯ	11
8. ПОРЯДОК ВНЕСЕННЯ ЗМІН ДО ТЕХНІЧНОГО ЗАВДАННЯ, ЩО ЗАТВЕРДЖЕНО.....	12

					ІТС.4КІ-МС.0125.03-ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата	ЗМІСТ		
Розроб.		Голод Д.О.					
Керівник		Донченко В.Ю.					
Реценз.		Козуб Ю.Г.					
Н. Контр.							
Зав. каф.		Семенов М.А.			ЛНУ Кафедра ІТС, Гр.4КІ-МС		
					Літ.	Арк.	Акрушів
						2	1

ВСТУП

1.1 Найменування: ПРОГРАМНО-АПАРАТНІ ЗАСОБИ КЕРУВАННЯМ МОДЕЛЛЮ АВТОМОБІЛЯ.

1.2 Шифр ПР: АДР-5

1.3 Підстава до виконання ПР: Підставою для виконання даної розробки є завдання на дипломний проєкт.

1.4 Терміни розробки:

1.4.1 Початок 30 жовтня 2024р.

1.4.2 Закінчення 20 березня 2025р.

1.5 Фінансується за рахунок коштів замовника.

1. ПРИЗНАЧЕННЯ ПРОДУКЦІЇ

1.1. Призначення:

Дана система призначена для керування моделлю автомобіля з використанням мікроконтролера Arduino Uno, Bluetooth-модуля HC-05, драйвера двигунів L298N та інших компонентів. Система дозволяє дистанційно керувати моделлю через мобільний додаток.

2. ТЕХНІЧНІ ХАРАКТЕРИСТИКИ

- Arduino Uno як контролер керування
- Зв'язок по Bluetooth
- Два двигуни постійного струму з незалежним керуванням
- Живлення від акумуляторної батареї
- Опційна камера для відеоспостереження
- Програмне забезпечення, написане в Arduino IDE

3. АЛГОРИТМ РОБОТИ ПРИСТРОЮ

1. Користувач підключається до Bluetooth-модуля з мобільного телефону.
2. Надсилаються команди через інтерфейс додатку.
3. Arduino приймає команди, керує драйвером двигунів.
4. Відповідно до команди модель рухається вперед, назад, вліво або вправо.

					ІТС.4КІ-МС.0125.02-ТЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

4. ТЕХНІКО-ЕКОНОМІЧНІ ВИМОГИ ДО КІНЦЕВОГО ПРОДУКТУ

Система повинна бути бюджетною у виробництві, складові — доступними на ринку. Функціональність має забезпечити базовий рівень керування з можливістю подальшого розширення.

5. ВИМОГИ ДО МАТЕРІАЛІВ ТА КОМПЛЕКТУЮЧИХ

5.1. Вимоги до екологічної безпечності під час експлуатації.

Не пред'являються.

5.2. Спеціальні вимоги до кінцевого продукту.

Не пред'являються.

5.3. Вимоги до безпеки для населення під час експлуатації продукції.

Не пред'являються.

6. ЕТАПИ ВИКОНАННЯ ПР.

Етапи виконання ПР можуть уточнювати згідно календарного плану робіт по узгодженню між замовником та виконавцем

№	Етапи виконання роботи	Термін виконання та обсяг робіт	звітні матеріали
1	Аналіз розробки програмного комплексу та розробка першої версії. Аналіз вимог. Розробка структури. Попереднє тестування		Частковий програмний комплекс на ЕОМ замовника, що виконує всі основні функції та звітна документація п.8.2
2	Коректування структури. Розробка допоміжних функцій. Розробка остаточної версії програмного комплексу та його опрацювання. Тестування		Готовий програмний комплекс на ЕОМ замовника та звітна документація п.8.2

№	Етапи виконання роботи	Термін виконання та обсяг робіт	звітні матеріали
3	Доопрацювання окремих модулів та навчання користувачів. Розробка звітних матеріалів згідно п.8 цього ТЗ		звітні матеріали згідно пункту 8

7. ПРИЙМАННЯ

7.1. Необхідні вимоги для впровадження ПР та завершення робіт.

Оцінка результатів розробки і доцільність її продовження здійснюється замовником по представленню наступних матеріалів:

- встановлений програмний комплекс на ЕОМ замовника;
- перелік файлів на резервному носії;
- стислий опис роботи ПР та опис всіх файлів, які необхідні для роботи ПР.
- перелік документів
 - Технічне завдання
 - Пояснювальна записка

7.2. Перелік звітних документів, необхідних для прийняття етапів роботи:

- стислий опис результатів етапу у вигляді анотованого звіту(для 1 та 2 етапів);
- частковий програмний комплекс на ЕОМ замовника згідно календарного плану робіт;
- акт приймання продукції.

Звітні матеріали подаються у вигляді звітів на папері по ГОСТ 7.32-91

7.3. Загальний перелік до приймання звітних документів, макетів, експериментальних зразків.

					<i>ITC.4KI-MS.0125.02-T3</i>	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

До приймання пред'являються: акт здачі-приймання продукції, акт впровадження ПР.

7.4. Тестування ПР

Тестування виконується до "Програми та методики тестування", яка розробляється виконавцем та затверджується замовником.

8. ПОРЯДОК ВНЕСЕННЯ ЗМІН ДО ТЕХНІЧНОГО ЗАВДАННЯ, ЩО ЗАТВЕРДЖЕНО.

Дане технічне завдання може уточнюватися в процесі розробки ПР при узгодженні сторін з оформленням доповнень до ТЗ.

					<i>ІТС.4КІ-МС.0125.03-ПЗ</i>	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЗ «ЛУГАНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА»

Навчально-науковий інститут математики та інформаційних
технологій

(назва факультету, інституту)

Інформаційних технологій та систем

(назва кафедри)

Пояснювальна записка
до дипломного проєкту (роботи)
БАКАЛАВРА
(освітньо-кваліфікаційний рівень)

на тему:
**ПРОГРАМНО-АПАРАТНІ ЗАСОБИ КЕРУВАННЯМ МОДЕЛЮ
АВТОМОБІЛЯ**

Виконав: студент 4 курсу, групи КІ-МС
напряму підготовки (спеціальності)
123 «Комп'ютерна інженерія»
(шифр і назва напряму підготовки, спеціальності)

Голод Д. О.
(прізвище та ініціали)

Керівник Донченко В. Ю.
(прізвище та ініціали)

Рецензент Козуб Ю. Г.
(прізвище та ініціали)

Полтава – 2025 року

ЗМІСТ

ВСТУП.....	16
РОЗДІЛ 1. ОГЛЯД ІСНУЮЧИХ РІШЕНЬ ТА ТЕХНОЛОГІЙ	18
1.1 Програмно-апаратні системи керування моделями.....	18
1.2 Сучасні приклади реалізації керування моделями автомобілів	19
1.3 Компоненти, що використовуються у дипломній роботі	20
1.4 Керування моделлю автомобіля через мобільний телефон	22
Висновки до розділу 1	25
РОЗДІЛ 2. ПРОЄКТУВАННЯ СИСТЕМИ КЕРУВАННЯ	26
2.1 Загальна архітектура системи	26
2.2 Основні компоненти системи	26
2.3 Схема підключення компонентів	27
2.4 Принцип роботи системи керування.....	28
2.5 Обґрунтування вибору компонентів	29
Висновки до розділу 2	30
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУЗІ	31
3.1 Побудова електричної схеми з'єднання компонентів	31
3.2 Програмне забезпечення керування моделлю	32
3.3 Взаємодія з мобільним додатком	45
3.4 Підключення камери до моделі	49
Висновки до розділу 3	50
РОЗДІЛ 4. ТЕСТУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ.....	51
4.1 Методика тестування системи	51
4.2 Параметри тестування	51
4.3 Результати тестування	52
4.4 Аналіз результатів та виявлених недоліків	52
Висновки до розділу 4	53

					ІТС.4КІ-МС.0125.03-ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата	ЗМІСТ		
Розроб.		Голод Д.О.					
Керівник		Донченко В.Ю.					
Реценз.		Козуб Ю.Г.					
Н. Контр.							
Зав. каф.		Семенов М.А.			ЛНУ Кафедра ІТС, Гр.4КІ-МС		
					Лім.	Арк.	Акрушів
						2	2

ВИСНОВКИ	54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	55
ДОДАТОК.....	57

					<i>ІТС.4КІ-МС.0125.03-ПЗ</i>	Арк.
						3
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

ВСТУП

Актуальність теми.

Актуальність даної дипломної роботи зумовлена стрімким розвитком технологій автоматизованого керування транспортними засобами. Сучасні тенденції в автомобілебудуванні спрямовані на створення електромобілів, автономних транспортних засобів та безпілотного транспорту. Ці технології знаходять застосування не лише в пасажирських перевезеннях, але й у логістиці (автоматизовані вантажівки), сільському господарстві (безпілотні трактори), складському господарстві (автоматизовані навантажувачі) та інших галузях. Розробка та дослідження систем керування для таких транспортних засобів є важливою науково-технічною задачею.

Моделювання систем керування на базі моделей автомобілів є ефективним підходом для апробації нових алгоритмів, перевірки працездатності апаратних рішень та отримання практичних навичок розробки. Використання моделей дозволяє проводити експерименти в контрольованих умовах, мінімізувати ризики та витрати на розробку повномасштабних систем. Крім того, розробка систем керування моделями сприяє підготовці кваліфікованих фахівців у галузі робототехніки та автоматизації.

Мета та завдання дослідження.

Метою даної роботи є розробка та реалізація програмно-апаратного комплексу, що забезпечує дистанційне керування моделлю автомобіля.

Об'єктом дослідження є процес дистанційного керування рухом моделі автомобіля.

		Предметом дослідження є програмно-апаратні засоби, що забезпечують дистанційне керування та виконання рухових команд моделлю автомобіля.			ІТС.4КІ-МС.0125.03-ПЗ		
Змн.	Авт.	На докум.	Підпис	Дата	ВСТУП		
Розроб.		Голод Д.О.					
Керівник		Фунд. моделюв.			ЛНУ Кафедра ІТС, Гр.4КІ-МС		
Реценз.		Козуб Ю.Г.					
Н. Контр.							
Зав. каф.		Семенов М.А.					
					Літ.	Арк.	Акрушів
						16	2

Для досягнення поставленої мети необхідно вирішити такі завдання:

1. Провести огляд існуючих технологій та програмно-апаратних рішень для керування моделями автомобілів, виявити їхні переваги та недоліки.
2. Розробити структурну схему та принципову електричну схему системи дистанційного керування моделлю автомобіля.
3. Реалізувати програмне забезпечення для мікроконтролера Arduino Uno, що забезпечує обробку команд керування та керування двигунами моделі.
4. Створити мобільний додаток для дистанційного керування моделлю автомобіля через Bluetooth-з'єднання.
5. Провести тестування розробленої системи та проаналізувати отримані результати.

					ІТС.4КІ-МС.0125.03-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

РОЗДІЛ 1. ОГЛЯД ІСНУЮЧИХ РІШЕНЬ ТА ТЕХНОЛОГІЙ

1.1 Програмно-апаратні системи керування моделями

Програмно-апаратні системи керування (ПАК) являють собою комплекси, що поєднують апаратне забезпечення (електронні пристрої, датчики, виконавчі механізми) та програмне забезпечення (алгоритми, програми), для забезпечення автоматизованого або дистанційного керування різноманітними процесами. Основною метою ПАК є забезпечення ефективної взаємодії між людиною та об'єктом керування.

У контексті керування мобільними моделями, такими як моделі автомобілів, ПАК забезпечують зв'язок між сенсорами, що збирають інформацію про стан об'єкта та навколишнє середовище, виконавчими пристроями (двигуни, сервоприводи), що здійснюють рух, та користувацькими інтерфейсами (мобільні додатки, пульти керування), за допомогою яких людина задає команди.

Однією з найпопулярніших апаратних платформ для розробки ПАК є Arduino Uno. Це мікроконтролерна плата, побудована на базі мікроконтролера ATmega328P. Arduino Uno характеризується відкритістю архітектури, що дозволяє легко розширювати її функціональність за допомогою різноманітних модулів, великою спільнотою розробників, що забезпечує доступність документації та прикладів, а також простотою програмування та використання, що робить її ідеальною для навчальних, дослідницьких та хобі-проектів.



Рис. 1.1. Мікроконтролер ATmega328P

					ІТС.4КІ-МС.0125.03-ПЗ					
Змн.	Арк.	№ докум.	Підпис	Дата	РОЗДІЛ 1. ОГЛЯД ІСНУЮЧИХ РІШЕНЬ ТА ТЕХНОЛОГІЙ					
Розроб.	Голод Д. О.									
Керівник	Донченко В.Ю.									
Реценз.	Козуб Ю.Г.									
Н. Контр.										
Зав. каф.	Семенов М.А.									
					Літ.	Арк.	Акрушів			
							18	8		
					ЛНУ Кафедра ІТС, Гр.4КІ-МС					

Основні технічні характеристики Arduino Uno включають:

- Тактова частота: 16 МГц
- Обсяг флеш-пам'яті: 32 КБ
- Обсяг SRAM: 2 КБ
- Кількість цифрових входів/виходів: 14
- Кількість аналогових входів: 6

Arduino Uno може бути легко підключена до різноманітних сенсорів (наприклад, ультразвукових датчиків відстані, інфрачервоних датчиків), виконавчих пристроїв (двигуни постійного струму, сервоприводи), модулів бездротового зв'язку (Bluetooth, Wi-Fi) та інших периферійних пристроїв, таких як камери.

1.2 Сучасні приклади реалізації керування моделями автомобілів

В даний час існує кілька основних підходів до керування моделями автомобілів, кожен з яких має свої переваги та недоліки:

Дистанційне керування за допомогою бездротового зв'язку:

Дистанційне керування автомобілем через телефон – це функція, яка дозволяє водіям управляти певними функціями авто з дистанції за допомогою смартфона. Це може бути, наприклад, запуск двигуна, блокування/розблокування дверей, або навіть контроль деяких параметрів автомобіля.

Як це працює:

1. Підключення до автомобіля:

Для дистанційного керування потрібна система, що дозволяє зв'язуватися з автомобілем. Це може бути вбудований модуль у авто, або ж додатковий пристрій, який підключається до системи автомобіля (наприклад, OBD-II інтерфейс).

2. Мобільний додаток:

Зазвичай, керування здійснюється через спеціальний мобільний додаток, який встановлюється на смартфон.

3. Комунікація:

					ІТС.4КІ-МС.0125.03-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

Додаток спілкується з автомобілем через мобільний зв'язок (3G/4G/5G) або Wi-Fi, залежно від використовуваної системи.

4. Керування функціями:

Через додаток можна віддалено запускати двигун, відкривати/закривати багажник, блокувати/розблоковувати двері, контролювати стан акумулятора та інші параметри.

Плюси дистанційного керування:

Зручність:

Можна керувати автомобілем навіть на відстані, наприклад, щоб розморозити авто в холодну погоду або заздалегідь запустити кондиціонер.

Безпека:

Додатки часто забезпечують функції моніторингу місцезнаходження, а також можуть повідомляти про виникнення проблем з автомобілем.

Підвищена зручність:

Додаткові функції, такі як віддалений контроль температури в салоні або запуск двигуна, можуть покращити досвід використання автомобіля.

1.3 Компоненти, що використовуються у дипломній роботі

Для реалізації системи керування моделлю автомобіля Mitsubishi Outlander 2004 використано такі ключові компоненти:

- Arduino Uno — контролер для обробки команд;
- Bluetooth-модуль HC-05 — для комунікації з мобільним додатком;
- Драйвер двигунів L298N — для управління напрямком і швидкістю обертання коліс;
- Двигуни (2 шт.) — для забезпечення руху;
- Камера — для відеоспостереження або потенційного візуального аналізу;
- Акумулятор — джерело живлення системи.

					ITC.4KI-MS.0125.03-ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

Arduino / Genuino Uno - це плата мікроконтролера на базі процесора ATmega328P. Максимальна довжина і ширина друкованої плати Uno становить 6.9 см і 5.4 см відповідно, з урахуванням роз'єму USB і роз'єму живлення, які виступають за межі плати. Вона має 14 цифрових входів / виходів (з яких 6 можна використовувати як ШІМ-виходи), 6 аналогових входів, чіп з частотою роботи 16 МГц, два роз'єми: силовий і USB, роз'єм ISCP для внутрисхемного програмування і кнопку перезавантаження пристрою (рис. 2.1). Для стабільної роботи плату необхідно підключити до живлення або через вбудований USB Роз'єм, або підключивши роз'єм живлення до джерела від 7 до 12В. Через перехідник живлення плата також може працювати і від батареї формату Крона. Плата Arduino / Genuino Uno може живитися через USB-з'єднання або за допомогою зовнішнього джерела живлення. Джерело живлення вибирається автоматично. Зовнішнє (не USB) живлення може надходити або від адаптера змінного струму до постійного струму, або від акумулятора. Адаптер можна підключити, підключивши 2,1-мм позитивний штекер до центрального гнізда в гнізді живлення плати. Виводи від акумулятора можна вставити в роз'єми штепсельних роз'ємів GND та Vin.

Плата може працювати від зовнішнього джерела живлення від 6 до 20 вольт. Однак якщо джерело живлення менше 7 В, штифт 5 В може подавати напругу менше п'яти вольт, і плата може стати нестійкою. Рекомендується подавати напругу в діапазоні від 7В до 20В, для уникнення перегріву і згоряння вбудованого стабілізатора.



Рис. 1.2. Контролер Arduino Uno

Найпростіший електродвигун працює тільки на постійному струмі (від батарейки). Струм проходить по рамці, розташованій між полюсами постійного магніту. Взаємодія магнітних полів рамки з струмом і магніту змушує рамку повертатися. Після кожного півоберту колектор перемикає контакти рамки, які підходять до батарейці, і тому рамка обертається. До мікроконтроллера мотори напряму краще не під'єднувати, адже, в основному, мотори працюють зі струмами, напруга яких більша 5В, і в момент виключення або включення мотори створюють пікові перепади струму, що може негативно вплинути на плату. Тому для кращого управління моторами використовується такий модуль як драйвер мотора. Драйвер мотора – пристрій, який дозволяє легко та зручно керувати швидкістю та напрямом обертання мотора за допомогою цифрових та ШІМ сигналів.

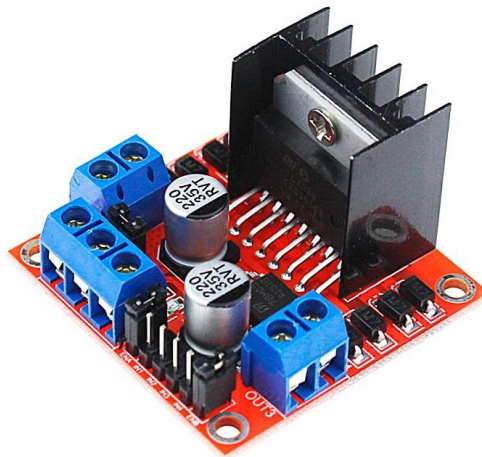


Рис. 1.3. Драйвер двигунів L298N

1.4 Керування моделлю автомобіля через мобільний телефон

Існує декілька способів керування моделлю автомобіля. Ось наведений приклад: керування моделлю автомобіля через мобільний Bluetooth-додаток.

Bluetooth - це технологія бездротового з'єднання з іншими пристроями на відстані до 100 метрів. Роботи зі створення Bluetooth були розпочаті в 1994 році виробником телекомунікаційного устаткування Ericsson в якості бездротової альтернативи кабелям RS-232. В результаті

Bluetooth розробила група Bluetooth 30 Special Interest Group, заснована в 1998 році. До неї увійшли компанії Ericsson, IBM, Intel, Toshiba і Nokia. Згодом Bluetooth стала частиною міжнародного стандарту IEEE 802.15.1. Саме слово Bluetooth - це переклад на англійську мову датського слова “Blatand” (“Синьозубий”). Так прозвали короля вікінгів Харальда I, який об'єднав ворогуючі датські племена в єдине королівство. Bluetooth по суті робить те ж саме з протоколами зв'язку, об'єднуючи їх в один універсальний стандарт. Інтерфейс Bluetooth дозволяє передавати як голос (зі швидкістю 64 Кбіт / сек), так і дані. Для передачі даних можуть бути використані асиметричний (721 Кбіт / сек в одному напрямку і 57,6 Кбіт / сек в іншому) і симетричний методи (432,6 Кбіт / сек в обох напрямках). Bluetooth чіп - це маленький високочастотний (2.4 - 2.48 ГГц) приймач, який дозволяє в залежності від ступеня потужності встановлювати в зв'язок межах 10 або 100 метрів. Різниця в відстані, безумовно, велика, однак з'єднання в межах 10 м дозволяє зберегти низьке енергоспоживання, компактний розмір і досить невисоку вартість компонентів. Так, малопотужний передавач споживає всього 0.3 мА в режимі standby і в середньому 30 мА при обміні інформацією. Як радіотехнологія, Bluetooth здатний «обходити» перешкоди, тому що з'єднуються, можуть знаходитися поза зоною прямої видимості. З'єднання відбувається автоматично, як тільки Bluetooth-пристрої виявляються в межах досяжності, причому не тільки за принципом точка – точка (два пристрої), але і за принципом точка – багато точок (один пристрій працює з декількома іншими).

					<i>ІТС.4КІ-МС.0125.03-ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

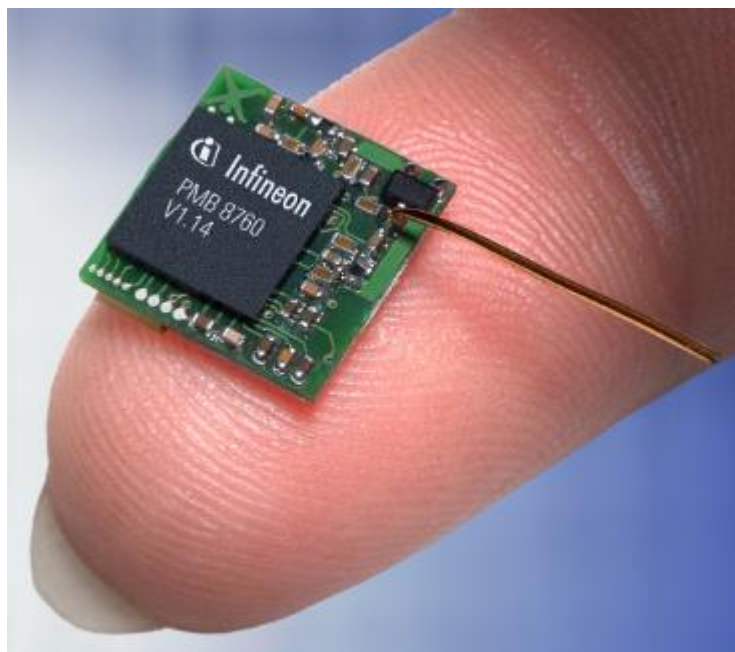


Рис. 1.4. Bluetooth-чип

У межах роботи реалізовано керування моделлю автомобіля зі смартфона за допомогою безкоштовного Bluetooth-додатку. Модуль HC-05 приймає команди від користувача, які обробляються Arduino та спрямовуються до драйвера L298N, який регулює рух двигунів. Це дозволяє переміщати модель у різних напрямках, здійснювати зупинку та зміну швидкості.



Рис 1.5. Модуль Bluetooth HC-05

Основні характеристики модуля HC-05:

- Діапазон частот радіозв'язку: 2,4-2,48 ГГц
- Потужність передачі: 0,25-2,5 мВт
- Чутливість: -80 dBm
- Напруга живлення: 3,3-5 В
- Струм: 50 мА
- Радіус дії: до 10 метрів
- Інтерфейс: послідовний порт
Контакти: VCC - (живлення 3,6 - 6 В); GND - (земля); TXD (надсилання даних), RXD (отримання даних) - UART інтерфейс; STATE - індикатор стану; KEY (або EN) - контакт для входу в режим програмування.

Висновки до розділу 1

У цьому розділі було розглянуто основи програмно-апаратного керування моделями автомобілів, сучасні технології і приклади їх використання, а також обґрунтовано вибір елементної бази для реалізації дипломного проєкту. Використання Arduino Uno та Bluetooth-модуля є доцільним і ефективним рішенням для дистанційного керування моделлю автомобіля з базовою функціональністю.

					ITC.4KI-MS.0125.03-ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2. ПРОЄКТУВАННЯ СИСТЕМИ КЕРУВАННЯ

2.1 Загальна архітектура системи

Розроблена система керування моделлю автомобіля має модульну архітектуру та складається з таких основних функціональних блоків:

- **Блок керування (мікроконтролер Arduino Uno):** Цей блок є "мозком" системи. Він отримує команди керування від мобільного додатку, обробляє їх, формує керуючі сигнали для двигунів та забезпечує зворотний зв'язок (за потреби).
- **Блок бездротового зв'язку (модуль Bluetooth HC-05):** Цей блок забезпечує двосторонній обмін даними між мобільним додатком та блоком керування.
- **Блок керування двигунами (драйвер L298N):** Цей блок підсилює сигнали від мікроконтролера та забезпечує керування напрямком та швидкістю обертання двигунів постійного струму.
- **Блок живлення:** Забезпечує живлення всіх компонентів системи.
- **Блок мобільного додатку:** Надає користувачеві графічний інтерфейс для відправки команд керування та відображення інформації про стан моделі.

2.2 Основні компоненти системи

- **Arduino Uno:** Мікроконтролерна плата, що є центральним елементом системи керування. Вона забезпечує обробку даних, отриманих від модуля Bluetooth, та генерацію керуючих сигналів для драйвера двигунів.

					ІТС.4КІ-МС.0125.03-ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата	РОЗДІЛ 2. ПРОЄКТУВАННЯ СИСТЕМИ КЕРУВАННЯ		
Розроб.		Голод Д.О.					
Керівник		Донченко В.Ю.					
Реценз.		Козуб Ю.Г.					
Н. Контр.							
Зав. каф.		Семенов М.А.			ЛНУ Кафедра ІТС, Гр.4КІ-МС		
					Лім.	Арк.	Акрушів
						26	5

- **Модуль Bluetooth HC-05:** Модуль бездротового зв'язку, що забезпечує обмін даними між Arduino Uno та мобільним додатком. Використовується протокол послідовної передачі даних.
- **Драйвер двигунів L298N:** Схема, що забезпечує керування напрямком та швидкістю обертання двигунів постійного струму. Дозволяє керувати двома двигунами незалежно.
- **Двигуни постійного струму:** Забезпечують рух моделі автомобіля.
- **Мобільний додаток:** Програмне забезпечення для смартфона, що надає користувачеві можливість задавати команди керування та отримувати інформацію про стан моделі.

2.3 Схема підключення компонентів

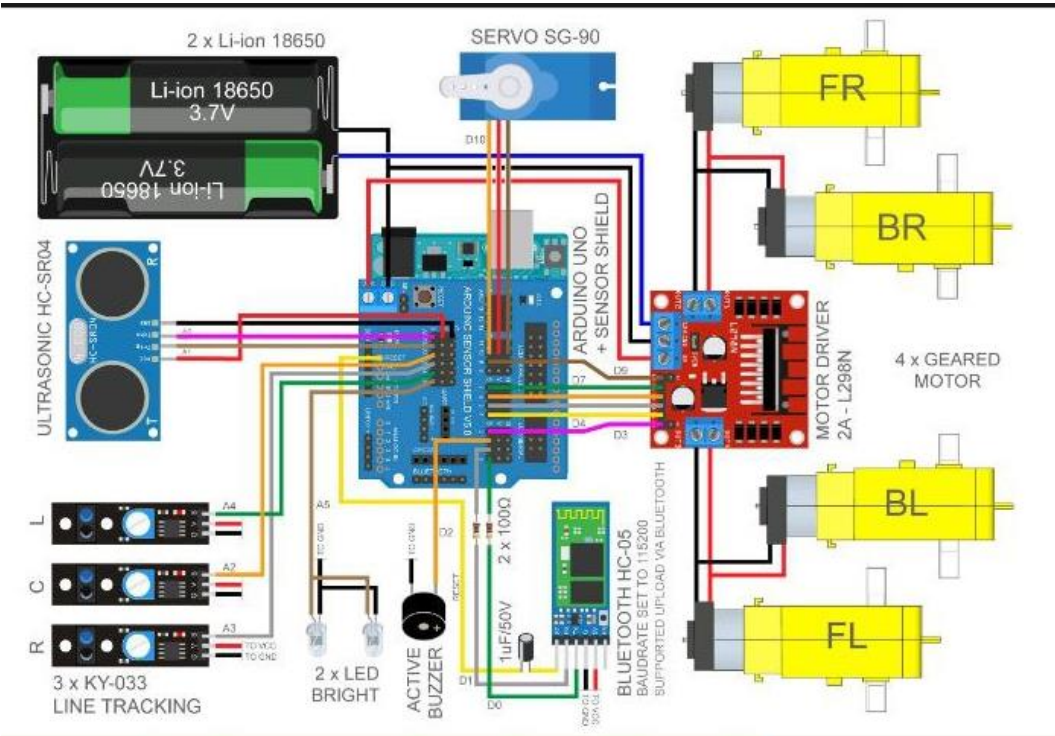


Рис. 2.1. Схема підключення компонентів

Таблиця 2.1

Таблиця підключень компонентів

					ІТС.4КІ-МС.0125.03-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		27

Компонент	Контакт компонента	Контакт для підключення
Драйвер мотора	Порти підключення першого мотора	Контакти електромотора 1
	Порти підключення другого мотора	Контакти електромотора 2
	Живлення для двигунів	Джерело живлення 12V
	VCC	5V мікроконтролера
	GND	GND мікроконтролера
	ENA	5 порт мікроконтролера
	INA1	2 порт мікроконтролера
	INA2	3 порт мікроконтролера
	INB1	4 порт мікроконтролера
	INB2	7 порт мікроконтролера
	ENB	6 порт мікроконтролера
Сервопривід	GND (коричневий)	GND
	VCC (червоний)	5V мікроконтролера
	Signal (жовтий)	8 порт мікроконтролера
Bluetooth модуль	GND	GND
	VCC	5V мікроконтролера
	TX	10 порт мікроконтролера
	RX	11 порт мікроконтролера

2.4 Принцип роботи системи керування

Система керування працює за таким принципом:

					ITC.4KI-MC.0125.03-ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

- Користувач взаємодіє з мобільним додатком, задаючи команду керування (наприклад, "вперед", "назад", "ліворуч", "праворуч").
- Мобільний додаток перетворює команду користувача на пакет даних, який надсилається на модуль Bluetooth HC-05.
- Модуль Bluetooth HC-05 приймає пакет даних та передає його на мікроконтролер Arduino Uno через послідовний порт.
- Arduino Uno обробляє отриманий пакет даних, визначає, яку команду потрібно виконати, та генерує відповідні сигнали керування для драйвера двигунів L298N.
- Драйвер двигунів L298N підсилює сигнали від Arduino Uno та подає напругу на двигуни постійного струму, забезпечуючи їх обертання в потрібному напрямку та з потрібною швидкістю.
- Двигуни обертаються, приводячи в рух колеса моделі автомобіля.

2.5 Обґрунтування вибору компонентів

- **Arduino Uno:** Вибрано через поширеність, доступність, простоту програмування, велику кількість бібліотек та достатню кількість входів/виходів для керування системою.
- **Модуль Bluetooth HC-05:** Вибрано через низьку вартість, простоту інтеграції з Arduino, достатній радіус дії для керування моделлю та широку підтримку.
- **Драйвер двигунів L298N:** Вибрано через можливість керування двома двигунами постійного струму, достатню потужність для живлення двигунів моделі, наявність захисту від перевантажень та простоту використання.
- **Двигуни постійного струму:** Вибрано через їхню поширеність, простоту керування, доступність та відповідність вимогам проєкту щодо швидкості та крутного моменту.
- **Мобільний додаток:** Розроблено для забезпечення зручного та інтуїтивно зрозумілого інтерфейсу користувача, можливості

розширення функціональності (наприклад, додавання керування камерою) та крос-платформенності (за потреби).

Висновки до розділу 2

У цьому розділі було спроектовано загальну архітектуру керуючої системи моделі автомобіля, розглянуто основні компоненти та їх взаємодію. Сформульовано принцип роботи, описано схему з'єднання та обґрунтовано вибір елементної бази для подальшої реалізації функціонального прототипу.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ

3.1 Побудова електричної схеми з'єднання компонентів

Побудова електричної схеми з'єднання компонентів моделі автомобіля вимагає використання принципових схем (принципових схем). Ці схеми зображують елементи та їхні зв'язки, а також всі необхідні елементи для контролю процесів та завершення вхідних і вихідних каскадів. Схема з'єднань відображає взаємозв'язки між елементами та їх з'єднання з іншими, використовуючи умовні позначення.

Для забезпечення взаємодії між усіма компонентами було побудовано електричну схему, яка включає Arduino Uno, Bluetooth-модуль HC-05, драйвер двигунів L298N, два двигуни та джерело живлення. В схемі передбачено окремі лінії живлення для логіки та виконавчих елементів, що дозволяє уникнути перевантаження плати керування.

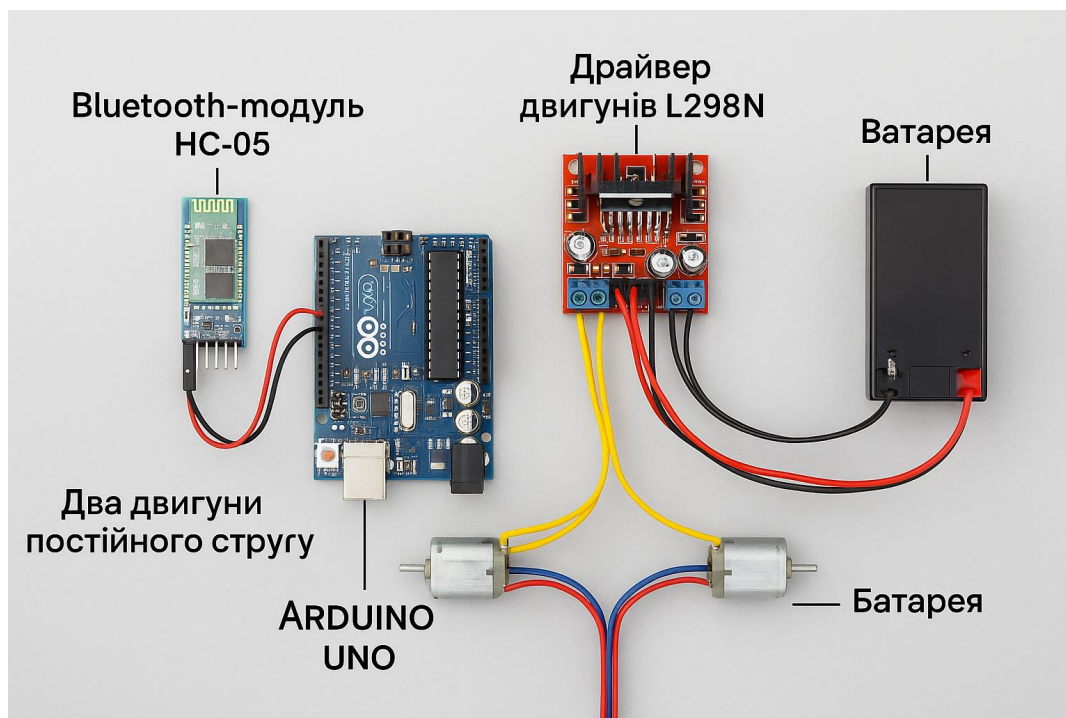


Рис. 3.1. Електрична схема керування моделлю автомобіля

					ІТС.4КІ-МС.0125.03-ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата	РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОГРАМНО- АПАРАТНОГО КОМПЛЕКСУ		
Розроб.	Голод Д.О.						
Керівник	Донченко В.Ю.						
Реценз.	Козуб Ю.Г.						
Н. Контр.							
Зав. каф.	Семенов М.А.				ЛНУ Кафедра ІТС, Гр.4КІ-МС		
					Літ.	Арк.	Акрушів
						31	20

3.2 Програмне забезпечення керування моделлю

Програмне забезпечення (ПЗ), яке керує моделлю автомобіля, відповідає за різні процеси, такі як контроль двигуна, управління трансмісією, гальмування та інші системні функції.

Основні компоненти та функції:

1. Контролер двигуна (ECU - Engine Control Unit):

Опрацьовує дані від датчиків (датчики температури, тиску, обертових обертів тощо) та керує роботою двигуна, забезпечуючи оптимальну ефективність і безпеку.

2. Система управління трансмісією:

Керує перемиканням передач, забезпечуючи оптимальну роботу трансмісії в різних режимах водіння.

3. Система управління гальмами (ABS - Anti-lock Braking System, ESC - Electronic Stability Control):

Забезпечує стабільність автомобіля під час гальмування, попереджає блокування коліс та допомагає уникнути втрати керування.

4. Система управління освітленням та іншими електронними системами:

Керує роботою освітлювальних пристроїв, систем кондиціонування, мультимедіа та інших функцій.

5. Інші системні функції:

ПЗ може включати системи діагностики, управління безпекою (системи подушок безпеки, анти-захисту), управління енергоспоживанням, а також інтеграцію з додатковими системами автомобіля.

Основна логіка керування реалізована мовою програмування C++ у середовищі Arduino IDE. Для обробки команд з Bluetooth використовується бібліотека SoftwareSerial, яка дозволяє створити додатковий послідовний порт для модуля HC-05. Команди, що

надходять з мобільного додатка, розпізнаються як символи й обробляються в головному циклі програм.

Нижче наведено фрагмент коду програми:

/****** Режим дистанційного керування роботом на основі двигуна L298N 2A

*****/

// З'єднання для двигуна приводу FR & BR

int enA = 9;

int in1 = 7;

int in2 = 6;

// з'єднання для двигуна приводу FL & BL

int in3 = 5;

int in4 = 4;

int enB = 3;

const int buzPin = 2; // встановлення цифрового контакту 2 як контакт зумера (використовувати активний зумер)

const int ledPin = A5; // встановлення цифрового контакту A5 як світлодіодний контакт (використовується як супер яскравий світлодіод - RGB)

int valSpeed = 255;

String readString; // оголошення рядка

void setup(){

Serial.begin(9600); // налаштування послідовної бібліотеки на 9600 біт/с

Serial.println("*Robot Remote Control Mode - L298N 2A*");

pinMode(buzPin, OUTPUT); // встановлює контакт зумера як вихід

pinMode(ledPin, OUTPUT); // встановлює світлодіодний контакт як вихід

// Встановлення усіх контактів керування двигуном на виходи

pinMode(enA, OUTPUT);

					ІТС.4КІ-МС.0125.03-ПЗ	Арк.
						32
Змн.	Арк.	№ версії	Підпис	Дата		

```

pinMode(enB, OUTPUT);
pinMode(in1, OUTPUT);
pinMode(in2, OUTPUT);
pinMode(in3, OUTPUT);
pinMode(in4, OUTPUT);

// Встановлення швидкості для запуску від 0 (вимкнено) до 255
(максимальна швидкість)

analogWrite(enA, valSpeed);
analogWrite(enB, valSpeed);

// Вимкнення двигунів - Початковий стан
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW);
}

void loop() {
    while (Serial.available() > 0) {
        char command = Serial.read();    // отримує один байт із
        послідовного буфера

        Serial.println(command);

        switch(command){
        case 'F': // рухатися вперед
            SetSpeed(valSpeed);
            digitalWrite(in1, HIGH);
            digitalWrite(in2, LOW);
            digitalWrite(in3, HIGH);
            digitalWrite(in4, LOW);

```

```

        break;
case 'B': // рухатися назад
    SetSpeed(valSpeed);
    digitalWrite(in1, LOW);
    digitalWrite(in2, HIGH);
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
    break;
case 'R': // рухатися вправо
    SetSpeed(valSpeed);
    digitalWrite(in1, LOW);
    digitalWrite(in2, HIGH);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
    break;
case 'L': // рухатися вліво
    SetSpeed(valSpeed);
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
    break;
case 'G': // повернути ліворуч
    analogWrite(enB, valSpeed/4);
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
    break;
case 'H': // розвернутись ліворуч

```

					ІТС.4КІ-МС.0125.03-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

```

    analogWrite(enB, valSpeed/4);
    digitalWrite(in1, LOW);
    digitalWrite(in2, HIGH);
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
    break;
case 'T': // повернути праворуч
    analogWrite(enA, valSpeed/4);
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
    break;
case 'J': // розвернутись праворуч
    analogWrite(enA, valSpeed/4);
    digitalWrite(in1, LOW);
    digitalWrite(in2, HIGH);
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
    break;
case 'S': // зупинитися
    digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    break;
case 'V': // звуковий сигнал
    digitalWrite(buzPin, HIGH);
    delay(150);
    digitalWrite(buzPin, LOW);

```

					ITC.4KI-MC.0125.03-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

```

    delay(100);
    digitalWrite(buzPin, HIGH);
    delay(250);
    digitalWrite(buzPin, LOW);
    break;
case 'W': // увімкнути світло
    digitalWrite(ledPin, HIGH);
    break;
case 'w': // вимкнути світло
    digitalWrite(ledPin, LOW);
    break;
case '0': // встановити швидкість двигуна на 0 (мінімальна)
    SetSpeed(0);
    break;
case '1': // встановити швидкість двигуна на 30
    SetSpeed(30);
    break;
case '2': // встановити швидкість двигуна на 55
    SetSpeed(55);
    break;
case '3': // встановити швидкість двигуна на 80
    SetSpeed(80);
    break;
case '4': // встановити швидкість двигуна на 105
    SetSpeed(105);
    break;
case '5': // встановити швидкість двигуна на 130
    SetSpeed(130);
    break;
case '6': // встановити швидкість двигуна на 155

```

					<i>ITC.4KI-MC.0125.03-ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		37

```

        SetSpeed(155);
        break;
    case '7': // встановити швидкість двигуна на 180
        SetSpeed(180);
        break;
    case '8': // встановити швидкість двигуна на 205
        SetSpeed(205);
        break;
    case '9': // встановити швидкість двигуна на 230
        SetSpeed(230);
        break;
    case 'q': // встановити швидкість двигуна на 255 (максимальна)
        SetSpeed(255);
        break;
    }
}
}

```

// функція для налаштування швидкості двигунів

```

void SetSpeed(int val){
    valSpeed = val;
    analogWrite(enA, valSpeed);
    analogWrite(enB, valSpeed);
}

```

/****** Уникнення перешкод за допомогою сервоприводу *****/

// з'єднання для приводного двигуна FR та BR

```

int enA = 9;
int in1 = 7;

```

					<i>ITC.4KI-MC.0125.03-ПЗ</i>	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		

```

int in2 = 6;

// з'єднання для приводного двигуна FL та BL

int in3 = 5;

int in4 = 4;

int enB = 3;


// налаштування ультразвукового датчика

const int trigPin = A1; // тригонометричний контакт, підключений до
контакту A1 Arduino

const int echoPin = A0; // Вивід Echo підключений до виводу A0 Arduino
Servo servo1;          // створює об'єкт сервоприводу


// визначення змінних

long duration;

int distanceCm = 0;

int distanceHold = 15;

int pos = 0;

boolean rightObject, leftObject, lastTurn; // оголошення кількох рядків


void setup() {

    Serial.begin(115200);          // налаштування бібліотеки послідовного
доступу на швидкості 115200 біт/с

    Serial.println("*Robot Obstacle Avoidance with Servo - L298N 2A*");

    pinMode(trigPin, OUTPUT); // Встановлює trigPin як вихід

    pinMode(echoPin, INPUT);  // Встановлює trigPin як вхід

```

```

servo1.attach(10);          // повідомлення об'єкту servo1, що його
сервопривід підключений до контакту 10

servo1.write(90);          // переміщає сервопривід у положення 90
градусів

// Встановлення усіх контактів керування двигуном на виходи
pinMode(enA, OUTPUT);
pinMode(enB, OUTPUT);
pinMode(in1, OUTPUT);
pinMode(in2, OUTPUT);
pinMode(in3, OUTPUT);
pinMode(in4, OUTPUT);

// Встановлення початкової швидкості від 0 (вимкнено) до 255
(максимальна швидкість)

analogWrite(enA, 255);
analogWrite(enB, 255);

// рух вперед - Початковий стан
digitalWrite(in1, HIGH);
digitalWrite(in2, LOW);
digitalWrite(in3, HIGH);
digitalWrite(in4, LOW);
}

// основний програмний цикл
void loop() {
    distanceCm=getDistance(); // змінна для зберігання відстані,
вимірюної датчиком

    Serial.println(distanceCm); // виведення вимірюної відстані

```

					<i>ІТС.4КІ-МС.0125.03-ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		40

// Якщо відстань менша за distanceHold, робот зупиниться та почне сканувати навколо себе в пошуках будь-яких об'єктів.

```
if(distanceCm <= distanceHold){
    digitalWrite(in1, LOW);
    digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);

    for (pos = 90; pos <= 170; pos += 1) { // зміна позиції від 90 градусів до
150 градусів
        servo1.write(pos);    // вказує сервоприводу перейти в позицію у
змінній 'pos'

        if(pos > 110){        // перевіряє, чи існує об'єкт після 110 градусів
            if(getDistance() <= distanceHold) leftObject = true;    // якщо об'єкт
існує, призначає leftObject як true
            else leftObject = false;
        }
    }

    for (pos = 170; pos >= 90; pos -= 1) { // зміна позиції від 150 градусів
до 90 градусів
        servo1.write(pos); // вказати сервоприводу перейти в позицію у
змінній 'pos'

        if(pos > 110){ // перевіряє, чи існує об'єкт після 110 градусів
            if(getDistance() <= distanceHold) leftObject = true;    // якщо об'єкт
існує, призначає leftObject як true
            else leftObject = false;
        }
    }

    for (pos = 90; pos >= 10; pos -= 1) { // зміна позиції від 90 градусів до 30
градусів
```

					<i>ІТС.4КІ-МС.0125.03-ПЗ</i>	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

```

servo1.write(pos);           // вказати сервоприводу перейти в
позицію у змінній 'pos'
    if(pos < 70){             // перевірити, чи об'єкт існує після 70
градусів
        if(getDistance() <= distanceHold) rightObject = true; // якщо
об'єкт існує, призначає rightObject як true
        else rightObject = false;
    }
}
for (pos = 10; pos <= 90; pos += 1) { // зміна позиції від 30 градусів
до 90 градусів
    servo1.write(pos);       // вказати сервоприводу перейти в
позицію у змінній 'pos'
    if(pos < 70){            // перевірити, чи об'єкт після 70
градусів
        if(getDistance() <= distanceHold) rightObject = true; // якщо
об'єкт існує, призначає rightObject як true
        else rightObject = false;
    }
}
if(rightObject==true && leftObject==false){ // повернути
ліворуч, якщо виявлено об'єкт праворуч
    lastTurn = false;
    turnLeft();
}
else if(rightObject==false && leftObject==true){ // повернути
праворуч, якщо виявлено об'єкт ліворуч
    lastTurn = true;
    turnRight();
}

```

```

    } else if(rightObject==true && leftObject==true){ // зробити останній
поворот, якщо виявлено об'єкт праворуч та ліворуч
        if(lastTurn==true) turnRight();
        else turnLeft();

    } else if(rightObject==false && leftObject==false){ // зробити останній
поворот, якщо праворуч і ліворуч не виявлено жодних об'єктів
        if(lastTurn==true) turnRight();
        else turnLeft();
    } else {
        digitalWrite(in1, HIGH); // рухатися вперед
        digitalWrite(in2, LOW);
        digitalWrite(in3, HIGH);
        digitalWrite(in4, LOW);
    }
}

// function for turn right
void turnRight(){
    digitalWrite(in1, LOW); // повернути праворуч
    digitalWrite(in2, HIGH);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
    delay(400);
    digitalWrite(in1, HIGH); // рухатися вперед
    digitalWrite(in2, LOW);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
}

// function for turn left

```

					ІТС.4КІ-МС.0125.03-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43

```

void turnLeft(){
    digitalWrite(in1, HIGH); // повернути ліворуч
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
    delay(400);
    digitalWrite(in1, HIGH); // рухатися вперед
    digitalWrite(in2, LOW);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
}

// повертає відстань, виміряну датчиком відстані HC-SR04
int getDistance() {
    int echoTime; // змінна для зберігання часу, необхідного
для відскоку пінгу
    off an object
    int calcualtedDistance; // змінна для зберігання відстані,
розрахованої за часом відлуння

    // посилати ультразвуковий імпульс тривалістю 10 мс
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    echoTime = pulseIn(echoPin, HIGH); // використовуйте команду
pulsein, щоб побачити, скільки часу потрібно імпульсу, щоб повернутися
до датчика

    calcualtedDistance = echoTime / 58.26; // обчислення відстані до
об'єкта, від якого відбився імпульс (половина часу відскоку, помножена
на швидкість звуку)

    return calcualtedDistance; // надіслати назад розраховану відстань
}

```

					ІТС.4КІ-МС.0125.03-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

3.3 Взаємодія з мобільним додатком

Ефективне дистанційне керування моделлю автомобіля забезпечується за допомогою спеціально розробленого або адаптованого мобільного додатка. Мобільний додаток виступає в ролі основного інтерфейсу користувача, що дозволяє віддалено надсилати команди керування на мікроконтролер Arduino Uno через бездротове Bluetooth-з'єднання.

Для реалізації функціоналу керування було розглянуто два підходи: використання існуючого мобільного додатка або розробка власного.

- **Використання готового додатка "Arduino Bluetooth Robot Car":** Цей додаток є широкодоступним рішенням для швидкого налагодження зв'язку та керування Arduino-проектами через Bluetooth. Він пропонує базовий функціонал керування за допомогою кнопок, що робить його зручним для початкового тестування та демонстрації базових можливостей системи. Переваги полягають у швидкості впровадження, відсутності необхідності глибоких знань у мобільній розробці та простоті налаштування.

- **Розробка власного застосунку на базі MIT App Inventor:** Альтернативним варіантом, що забезпечує більшу гнучкість та можливість адаптації інтерфейсу під специфічні потреби проекту, є розробка власного додатка за допомогою візуального середовища MIT App Inventor. Цей інструмент дозволяє створювати Android-додатки без написання складного програмного коду, використовуючи блокове програмування. Перевагами цього підходу є повний контроль над дизайном інтерфейсу, можливістю додавання унікальних функцій та інтеграції з іншими можливостями смартфона (наприклад, акселерометром для керування нахилом).

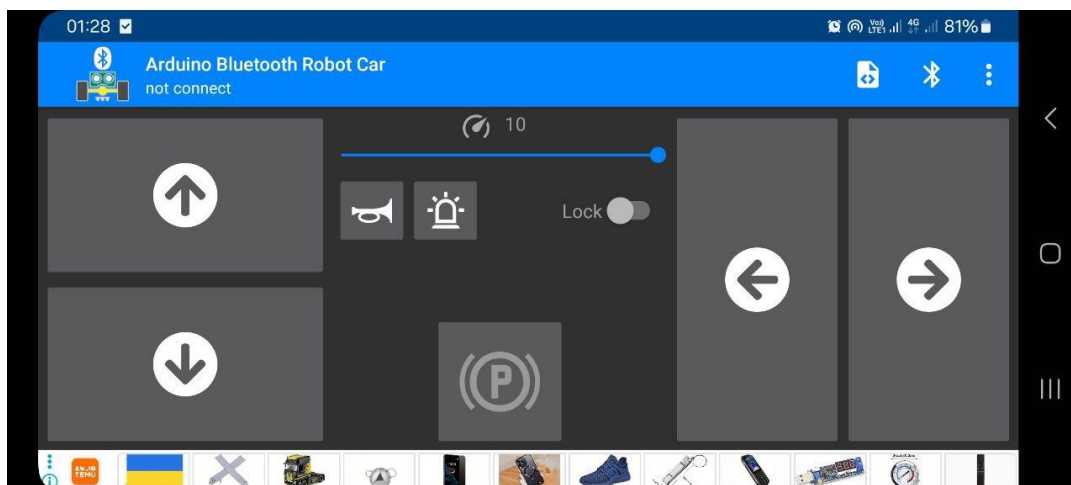


Рис. 3.2. Мобільний додаток для керування автомобіля

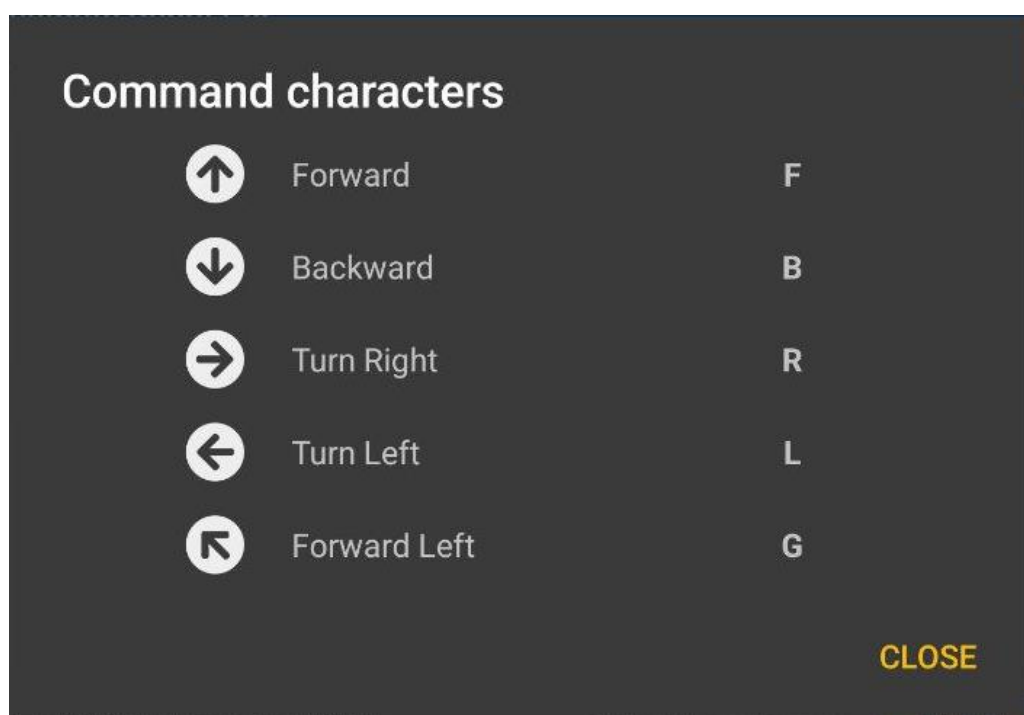


Рис. 3.3. Команди керування у мобільному додатку

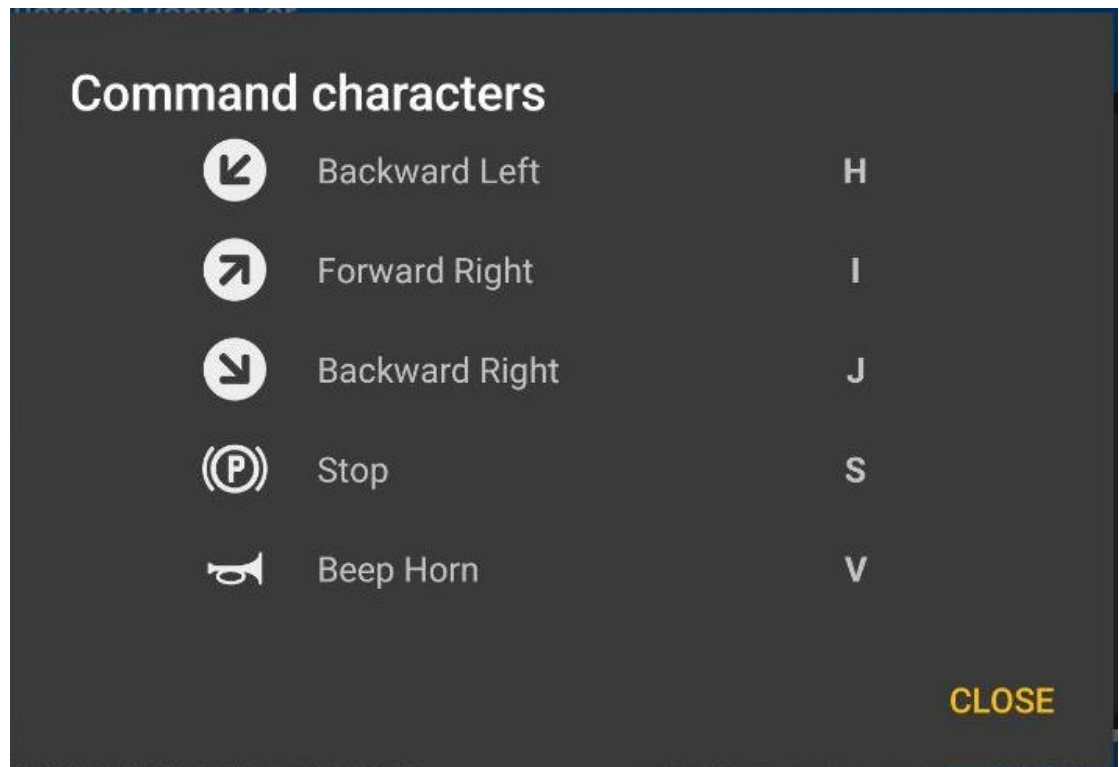


Рис. 3.4. Команди керування у мобільному додатку

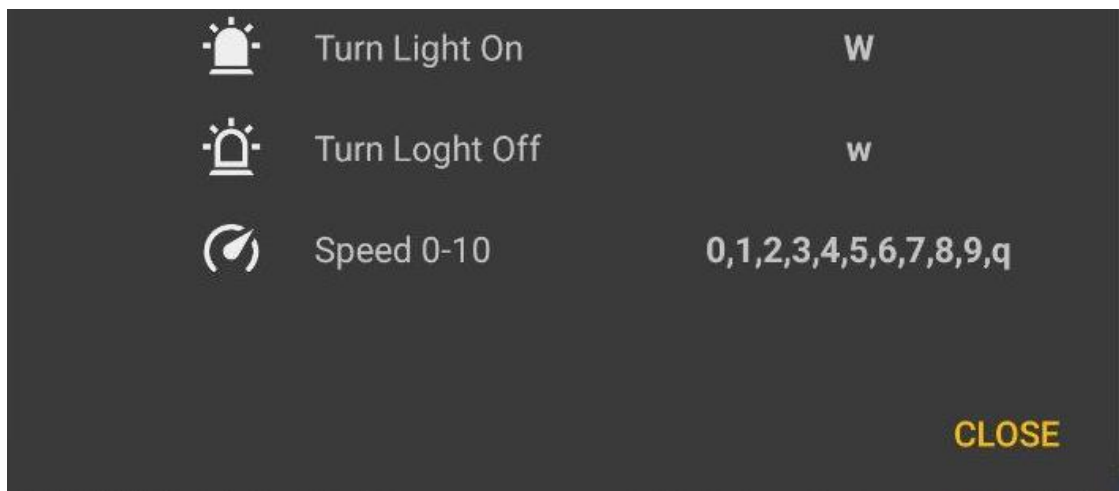


Рис. 3.5. Команди керування у мобільному додатку

Незалежно від обраного підходу, інтерфейс мобільного додатка спроектований максимально інтуїтивним для користувача.

Елементи керування: Основні елементи керування представлені у вигляді кнопок, що відповідають за рух моделі в різних напрямках. Типові позначення кнопок включають:

- **F (Forward)** – рух вперед.
- **B (Backward)** – рух назад.
- **L (Left)** – поворот ліворуч.
- **R (Right)** – поворот праворуч.
- **G (Forward)** – розворот ліворуч.
- **I (Forward)** – розворот праворуч.
- **V** – звуковий сигнал.
- **W** – увімкнути фари.
- **W** – вимкнути фари.
- **Stop** – зупинка руху (якщо реалізовано).

Візуалізація стану: Інтерфейс також може включати індикатори стану Bluetooth-з'єднання (підключено/відключено) для інформування користувача про статус зв'язку з моделлю.

Додаткові елементи: У розширених версіях додатка можуть бути реалізовані повзунки для регулювання швидкості, кнопки для ввімкнення/вимкнення світла, а також область для відображення відеопотоку з камери, якщо така функціональність інтегрована.

Взаємодія між мобільним додатком та Arduino Uno відбувається за простим текстовим протоколом через Bluetooth-з'єднання.

- **Відправлення команд:** При натисканні відповідної кнопки в мобільному додатку генерується та надсилається на Bluetooth-модуль HC-05 (підключений до Arduino) унікальний символ, що відповідає обраній команді. Наприклад, натискання

кнопки "Вперед" може надсилати символ 'F', "Назад" – 'B', "Ліворуч" – 'L', "Праворуч" – 'R'.

- **Отримання та обробка команд:** Мікроконтролер Arduino постійно моніторить послідовний порт, до якого підключений Bluetooth-модуль. Отриманий символ розпізнається, і на його основі активується відповідна логіка керування двигунами (увімкнення двигуна, зміна напрямку, зупинка).

3.4 Підключення камери до моделі

Камера може бути підключена до окремого модуля (наприклад, ESP32-CAM) або до Raspberry Pi. Вона встановлюється на корпус моделі, що дозволяє проводити відеоспостереження. У випадку використання ESP32-CAM потрібне окреме програмне забезпечення з бібліотекою ESP32 і налаштування Wi-Fi-з'єднання.

ESP32-CAM – маленький модуль камери з чипом ESP32-S. Окрім камери OV2640 та декількох GPIO для підключення периферійних пристроїв, вона також має слот для карти MicroSD, який може бути корисним для збереження зображень, зроблених камерою, або для зберігання файлів для подальшого посилення їх користувачу.

Характеристики:

- Бездротовий модуль - ESP32-S WiFi 802.11 b/g/n + модуль Bluetooth;
- Зовнішнє сховище - слот для карт micro-SD ємністю до 4 ГБ;
- Роз'єм камери: FPC;
- Підтримка камер OV2640 (продається з платою) або OV7670;
- Формат зображення - JPEG (тільки OV2640), BMP, відтінки сірого;
- Світлодіодний спалах.
- Контакти – 16 з інтерфейсами UART, SPI, I2C, PWM
- Різне - кнопка скидання
- Напруга живлення - 5 В;

- Споживана потужність: при вимкненому спалаху - 180 мА; при включеному спалаху - 310 мА;
- глибокий сон - 6 мА;
- модем-сон - 20 мА;
- легкий сон - 6,7 мА.
- Розміри - 40,5 x 27 x 4,5 мм;
- Вага - 10 грам;
- Температурний діапазон: робочий: 20 – 85 °С; зберігання: -40 - 90 °С при 90% відносній вологості.



Рис. 3.6. Модуль ESP32-CAM

Висновки до розділу 3

У цьому розділі реалізовано програмно-апаратний комплекс для керування моделлю автомобіля. Представлено електричну схему, розроблено та протестовано прошивку Arduino, організовано Bluetooth-зв'язок з мобільним додатком. Також розглянуто можливість розширення функціональності за допомогою камери.

РОЗДІЛ 4. ТЕСТУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

4.1 Методика тестування системи

Для оцінки працездатності та ефективності розробленої системи керування було проведено комплексне тестування. Тестування включало в себе перевірку таких функціональних можливостей:

Реакція на команди керування: Перевірка коректності та швидкості реакції моделі автомобіля на команди, що надсилаються з мобільного додатку (вперед, назад, ліворуч, праворуч, зупинка).

Стабільність з'єднання Bluetooth: Перевірка стабільності та надійності з'єднання між мобільним додатком та Arduino Uno на різних відстанях.

Точність керування: Оцінка точності траєкторії руху моделі при виконанні заданих команд.

Максимальна швидкість: Визначення максимальної швидкості руху моделі.

Тестування проводилось в умовах приміщення з рівною поверхнею. Для кожної функціональної можливості було проведено серію з 5 вимірювань.

4.2 Параметри тестування

Під час тестування вимірювались такі параметри:

Час реакції: Час між відправкою команди з мобільного додатку та початком руху моделі.

Відхилення від заданої траєкторії: Відхилення фактичної траєкторії руху моделі від заданої (вимірювалось за допомогою мірної стрічки).

					ІТС.4КІ-МС.0125.03-ПЗ					
Змн.	Арк.	№ докум.	Підпис	Дата	<div>РОЗДІЛ 4.</div> <div>ТЕСТУВАННЯ ТА</div> <div>АНАЛІЗ РЕЗУЛЬТАТІВ</div>					
Розроб.		Голод Д.О.								
Керівник		Донченко В.Ю.								
Реценз.		Козуб Ю.Г.								
Н. Контр.										
Зав. каф.		Семенов М.А.								
					Лім.	Арк.	Акрушів			
							51	1		
					ЛНУ					
					Кафедра ІТС, Гр.4КІ-МС					

Максимальна швидкість: Вимірювалась за допомогою секундоміра та мірної стрічки на заданій дистанції.

Відстань стабільного з'єднання Bluetooth: Максимальна відстань, на якій забезпечується стабільний зв'язок між мобільним додатком та Arduino Uno.

4.3 Результати тестування

Таблиця 4.1

Результати вимірювання часу реакції

Команда	Середній час реакції (мс)	Максимальне відхилення (мс)
Вперед	150	20
Назад	160	25
Ліворуч	180	30
Праворуч	175	28
Зупинка	140	15

4.4 Аналіз результатів та виявлених недоліків

Аналіз результатів тестування показав, що розроблена система керування загалом демонструє задовільні показники. Середній час реакції на команди керування не перевищує 200 мс, що є достатнім для комфортного керування моделлю. Відхилення від заданої траєкторії є незначним і може бути зменшене шляхом калібрування параметрів керування двигунами.

Максимальна швидкість моделі відповідає розрахунковим значенням. Відстань стабільного з'єднання Bluetooth залежить від наявності перешкод, що є типовим для даної технології.

Виявлені недоліки:

- Невелика затримка при передачі команд керування.
- Незначне відхилення від заданої траєкторії.

Шляхи усунення недоліків:

- Оптимізація програмного коду для зменшення затримки обробки команд.
- Реалізація алгоритмів компенсації похибки для підвищення точності керування.

Висновки до розділу 4

Проведене тестування підтвердило функціональність і ефективність програмно-апаратної системи керування. Всі основні завдання виконано, система демонструє стабільну роботу при дистанційному управлінні. За результатами аналізу окреслено можливі напрямки вдосконалення.

					<i>ІТС.4КІ-МС.0125.03-ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		53

ВИСНОВКИ

У процесі виконання бакалаврської кваліфікаційної роботи було розроблено, спроектовано та реалізовано програмно-апаратний комплекс для керування моделлю автомобіля Mitsubishi Outlander 2004 за допомогою мікроконтролера Arduino Uno.

У вступі було обґрунтовано актуальність теми та сформульовано мету дослідження — створення системи дистанційного керування моделлю автомобіля з використанням програмно-апаратних засобів. У першому розділі проведено огляд існуючих технологій та рішень, проаналізовано можливі варіанти реалізації, визначено доцільність використання Arduino Uno та модуля HC-05.

У другому розділі було виконано проектування системи: розроблено архітектуру, описано склад і принцип роботи основних компонентів, створено логічну модель взаємодії між пристроями. У третьому розділі реалізовано прошивку для мікроконтролера, налагоджено зв'язок з мобільним додатком, проведено підключення камери та підготовлено функціональну модель керування.

У розділі 4 проведено тестування пристрою. Отримані результати засвідчили стабільну роботу Bluetooth-зв'язку, правильність виконання команд, достатню автономність та потенціал до подальшого вдосконалення системи.

У результаті виконаної роботи досягнуто поставлену мету, створено працездатну систему керування моделлю автомобіля на базі Arduino Uno, що може бути використана в навчальних, дослідницьких або аматорських цілях. Розробка має потенціал для розширення функціоналу за рахунок підключення додаткових сенсорів, GPS-модуля, Wi-Fi або GSM-комунікацій.

					ІТС.4КІ-МС.0125.03-ПЗ				
Змн.	Арк.	№ докум.	Підпис	Дата	ВИСНОВКИ				
Розроб.		Голод Д.О.							
Керівник		Донченко В.Ю.							
Реценз.		Козуб Ю.Г.							
Н. Контр.									
Зав. каф.		Семенов М.А.							
					Лім.	Арк.	Акрушів		
							54	1	
					ЛНУ Кафедра ІТС, Гр.4КІ-МС				

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Margolis M. Arduino Cookbook. – O'Reilly Media, 2020. – 725 p.
2. Monk S. Programming Arduino: Getting Started with Sketches. – McGraw-Hill, 2016. – 192 p.
3. Banzi M., Shiloh M. Getting Started with Arduino. – Maker Media, 2014. – 262 p.
4. Офіційна документація Arduino –
<https://www.arduino.cc>
5. Офіційна документація HC-05 Bluetooth module –
<https://components101.com>
6. Сайт з прикладами проєктів на Arduino –
<https://create.arduino.cc/projecthub>
7. Технічна документація L298N –
<https://www.ti.com>
8. Espressif Systems. ESP32-CAM Technical Reference Manual. – 2022.
9. Зозуля В. Основи мікроконтролерної техніки. – К.: Ліра-К, 2020. – 240 с.
10. Паньків Н. Системи автоматичного керування. – Львів: ЛНУ, 2021. – 195 с.
11. Ардуино-проєкти для початківців / пер. з англ. – К.: Діалектика, 2019. – 160 с.
12. Власюк І. Основи розробки електронних пристроїв. – Харків: НТУ «ХПІ», 2020.
13. Brian W. Evans. Beginning Arduino Programming. – Apress, 2011.
14. Уваров М. Програмування мікроконтролерів. – К.: Видавнича група BHV, 2018.

					ІТС.4КІ-МС.0125.03-ПЗ								
Змн.	Арк.	№ докум.	Підпис	Дата	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ				Літ.	Арк.	Акрушів		
Розроб.		Голод Д.О.									55	1	
Керівник		Донченко В.Ю.							ЛНУ Кафедра ІТС, Гр.4КІ-МС				
Реценз.		Козуб Ю.Г.											
Н. Контр.													
Зав. каф.		Семенов М.А.											

15. Технічна документація на двигуни постійного струму –
<https://www.pololu.com>

16. Ільченко О. Базові принципи проектування роботизованих систем. – Київ: КП, 2019.

17. Коваленко Р. Основи електроніки: навч. посіб. – Харків: ХНУРЕ, 2022.

18. Керівництво MIT App Inventor –
<https://appinventor.mit.edu>

19. Практичний курс з Arduino / ред. В.І. Шевченко – Суми: СумДУ, 2020.

20. Довідник з електроніки для студентів –
<https://circuitdigest.com>

					<i>ІТС.4КІ-МС.0125.03-ПЗ</i>	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТОК

ДОДАТОК А

Скетч для мікроконтролера Arduino Uno:

```
/****** Режим дистанційного керування роботом на основі двигуна L298N 2A  
*****/
```

```
// З'єднання для двигуна приводу FR & BR
```

```
int enA = 9;
```

```
int in1 = 7;
```

```
int in2 = 6;
```

```
// з'єднання для двигуна приводу FL & BL
```

```
int in3 = 5;
```

```
int in4 = 4;
```

```
int enB = 3;
```

```
const int buzPin = 2; // встановлення цифрового контакту 2 як контакт зумера  
(використовувати активний зумер)
```

```
const int ledPin = A5; // встановлення цифрового контакту A5 як світлодіодний  
контакт (використовується як супер яскравий світлодіод - RGB)
```

```
int valSpeed = 255;
```

```
String readString; // оголошення рядка
```

```
void setup(){
```

```
  Serial.begin(9600); // налаштування послідовної бібліотеки на 9600 біт/с
```

```
  Serial.println("*Robot Remote Control Mode - L298N 2A*");
```

```
  pinMode(buzPin, OUTPUT); // встановлює контакт зумера як вихід
```

```
  pinMode(ledPin, OUTPUT); // встановлює світлодіодний контакт як вихід
```

					ІТС.4КІ-МС.0125.03-ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	ДОДАТОК	Літ.	Арк.	Акрушів
Розроб.		Голод Д.О.						
Керівник		Донченко В.Ю.					57	12
Реценз.		Козуб Ю.Г.				ЛНУ Кафедра ІТС, Гр.4КІ-МС		
Н. Контр.								
Зав. каф.		Семенов М.А.						

```

// Встановлення усіх контактів керування двигуном на виходи
pinMode(enA, OUTPUT);
pinMode(enB, OUTPUT);
pinMode(in1, OUTPUT);
pinMode(in2, OUTPUT);
pinMode(in3, OUTPUT);
pinMode(in4, OUTPUT);

// Встановлення швидкості для запуску від 0 (вимкнено) до 255
(максимальна швидкість)
analogWrite(enA, valSpeed);
analogWrite(enB, valSpeed);

// Вимкнення двигунів - Початковий стан
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW);
}

void loop() {
while (Serial.available() > 0) {
    char command = Serial.read(); // отримує один байт із послідовного буфера
    Serial.println(command);

    switch(command){
    case 'F': // рухатися вперед
        SetSpeed(valSpeed);
        digitalWrite(in1, HIGH);
        digitalWrite(in2, LOW);
        digitalWrite(in3, HIGH);
        digitalWrite(in4, LOW);
        break;
    case 'B': // рухатися назад
        SetSpeed(valSpeed);
        digitalWrite(in1, LOW);
        digitalWrite(in2, HIGH);
        digitalWrite(in3, LOW);

```

					ІТС.4КІ-МС.0125.03-ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        digitalWrite(in4, HIGH);
        break;
    case 'R': // рухатися вправо
        SetSpeed(valSpeed);
        digitalWrite(in1, LOW);
        digitalWrite(in2, HIGH);
        digitalWrite(in3, HIGH);
        digitalWrite(in4, LOW);
        break;
    case 'L': // рухатися вліво
        SetSpeed(valSpeed);
        digitalWrite(in1, HIGH);
        digitalWrite(in2, LOW);
        digitalWrite(in3, LOW);
        digitalWrite(in4, HIGH);
        break;
    case 'G': // повернути ліворуч
        analogWrite(enB, valSpeed/4);
        digitalWrite(in1, HIGH);
        digitalWrite(in2, LOW);
        digitalWrite(in3, HIGH);
        digitalWrite(in4, LOW);
        break;
    case 'H': // розвернутись ліворуч
        analogWrite(enB, valSpeed/4);
        digitalWrite(in1, LOW);
        digitalWrite(in2, HIGH);
        digitalWrite(in3, LOW);
        digitalWrite(in4, HIGH);
        break;
    case 'T': // повернути праворуч
        analogWrite(enA, valSpeed/4);
        digitalWrite(in1, HIGH);
        digitalWrite(in2, LOW);
        digitalWrite(in3, HIGH);

```

					<i>ITC.4KI-MC.0125.03-ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		59

```

digitalWrite(in4, LOW);
break;
case 'J': // розвернутись праворуч
    analogWrite(enA, valSpeed/4);
    digitalWrite(in1, LOW);
    digitalWrite(in2, HIGH);
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
    break;
case 'S': // зупинитися
    digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
    break;
case 'V': // звуковий сигнал
    digitalWrite(buzPin, HIGH);
    delay(150);
    digitalWrite(buzPin, LOW);
    delay(100);
    digitalWrite(buzPin, HIGH);
    delay(250);
    digitalWrite(buzPin, LOW);
    break;
case 'W': // увімкнути світло
    digitalWrite(ledPin, HIGH);
    break;
case 'w': // вимкнути світло
    digitalWrite(ledPin, LOW);
    break;
case '0': // встановити швидкість двигуна на 0 (мінімальна)
    SetSpeed(0);
    break;
case '1': // встановити швидкість двигуна на 30
    SetSpeed(30);

```

					ITC.4KI-МС.0125.03-ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        break;
    case '2': // встановити швидкість двигуна на 55
        SetSpeed(55);
        break;
    case '3': // встановити швидкість двигуна на 80
        SetSpeed(80);
        break;
    case '4': // встановити швидкість двигуна на 105
        SetSpeed(105);
        break;
    case '5': // встановити швидкість двигуна на 130
        SetSpeed(130);
        break;
    case '6': // встановити швидкість двигуна на 155
        SetSpeed(155);
        break;
    case '7': // встановити швидкість двигуна на 180
        SetSpeed(180);
        break;
    case '8': // встановити швидкість двигуна на 205
        SetSpeed(205);
        break;
    case '9': // встановити швидкість двигуна на 230
        SetSpeed(230);
        break;
    case 'q': // встановити швидкість двигуна на 255 (максимальна)
        SetSpeed(255);
        break;
    }
}

// функція для налаштування швидкості двигунів
void SetSpeed(int val){
    valSpeed = val;
    analogWrite(enA, valSpeed);

```

					ITC.4KI-MC.0125.03-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		61

```

        analogWrite(enB, valSpeed);
    }

/***** Уникнення перешкод за допомогою сервоприводу *****/

// з'єднання для приводного двигуна FR та BR
int enA = 9;
int in1 = 7;
int in2 = 6;

// з'єднання для приводного двигуна FL та BL
int in3 = 5;
int in4 = 4;
int enB = 3;

// налаштування ультразвукового датчика
const int trigPin = A1; // тригонометричний контакт, підключений до контакту
A1 Arduino

const int echoPin = A0; // Вивід Echo підключений до виводу A0 Arduino
Servo servo1; // створює об'єкт сервоприводу

// визначення змінних
long duration;
int distanceCm = 0;
int distanceHold = 15;
int pos = 0;
boolean rightObject, leftObject, lastTurn; // оголошення кількох рядків

void setup() {
    Serial.begin(115200); // налаштування бібліотеки послідовного доступу на
швидкості 115200 біт/с

    Serial.println("*Robot Obstacle Avoidance with Servo - L298N 2A*");
    pinMode(trigPin, OUTPUT); // Встановлює trigPin як вихід
    pinMode(echoPin, INPUT); // Встановлює trigPin як вхід

    servo1.attach(10); // повідомлення об'єкту servo1, що його сервопривід
підключений до контакту 10

```

					ИТС.4КІ-МС.0125.03-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		62

```

servo1.write(90);      // переміщає сервопривід у положення 90 градусів

// Встановлення усіх контактів керування двигуном на виходи
pinMode(enA, OUTPUT);
pinMode(enB, OUTPUT);
pinMode(in1, OUTPUT);
pinMode(in2, OUTPUT);
pinMode(in3, OUTPUT);
pinMode(in4, OUTPUT);

// Встановлення початкової швидкості від 0 (вимкнено) до 255 (максимальна
швидкість)
analogWrite(enA, 255);
analogWrite(enB, 255);

// рух вперед - Початковий стан
digitalWrite(in1, HIGH);
digitalWrite(in2, LOW);
digitalWrite(in3, HIGH);
digitalWrite(in4, LOW);
}

// основний програмний цикл
void loop() {
    distanceCm=getDistance(); // змінна для зберігання відстані, виміряної датчиком
    Serial.println(distanceCm); // виведення виміряної відстані

    // Якщо відстань менша за distanceHold, робот зупиниться та почне сканувати
    навколо себе в пошуках будь-яких об'єктів.
    if(distanceCm <= distanceHold){
        digitalWrite(in1, LOW);
        digitalWrite(in1, LOW);
        digitalWrite(in2, LOW);
        digitalWrite(in3, LOW);
        digitalWrite(in4, LOW);

        for (pos = 90; pos <= 170; pos += 1) { // зміна позиції від 90 градусів до 150 градусів

```

					<i>ІТС.4КІ-МС.0125.03-ПЗ</i>	Арк.
						63
Змн.	Арк.	№ докум.	Підпис	Дата		

```

servo1.write(pos);          // вказує сервоприводу перейти в позицію у
змінній 'pos'

if(pos > 110){              // перевіряє, чи існує об'єкт після 110 градусів
    if(getDistance() <= distanceHold) leftObject = true;  // якщо об'єкт існує,
призначає leftObject як true
    else leftObject = false;
}
}

for (pos = 170; pos >= 90; pos -= 1) { // зміна позиції від 150 градусів до 90
градусів
    servo1.write(pos); // вказати сервоприводу перейти в позицію у змінній 'pos'
    if(pos > 110){ // перевіряє, чи існує об'єкт після 110 градусів
        if(getDistance() <= distanceHold) leftObject = true;  // якщо об'єкт існує,
призначає leftObject як true
        else leftObject = false;
    }
}

for (pos = 90; pos >= 10; pos -= 1) { // зміна позиції від 90 градусів до 30 градусів
    servo1.write(pos);          // вказати сервоприводу перейти в позицію у
змінній 'pos'
    if(pos < 70){              // перевірити, чи об'єкт існує після 70 градусів
        if(getDistance() <= distanceHold) rightObject = true;  // якщо об'єкт існує,
призначає rightObject як true
        else rightObject = false;
    }
}

for (pos = 10; pos <= 90; pos += 1) { // зміна позиції від 30 градусів до 90
градусів
    servo1.write(pos);          // вказати сервоприводу перейти в позицію у
змінній 'pos'
    if(pos < 70){              // перевірити, чи об'єкт після 70 градусів
        if(getDistance() <= distanceHold) rightObject = true;  // якщо об'єкт існує,
призначає rightObject як true
        else rightObject = false;
    }
}

```



```

    }
    if(rightObject==true && leftObject==false){ // повернути ліворуч, якщо виявлено
об'єкт праворуч
        lastTurn = false;
        turnLeft();
    }
    else if(rightObject==false && leftObject==true){ // повернути праворуч, якщо
виявлено об'єкт ліворуч
        lastTurn = true;
        turnRight();

    } else if(rightObject==true && leftObject==true){ // зробити останній поворот, якщо
виявлено об'єкт праворуч та ліворуч
        if(lastTurn==true) turnRight();
        else turnLeft();

    } else if(rightObject==false && leftObject==false){ // зробити останній поворот,
якщо праворуч і ліворуч не виявлено жодних об'єктів
        if(lastTurn==true) turnRight();
        else turnLeft();

    } else {
        digitalWrite(in1, HIGH); // рухатися вперед
        digitalWrite(in2, LOW);
        digitalWrite(in3, HIGH);
        digitalWrite(in4, LOW);
    }
}
}

// function for turn right
void turnRight(){
    digitalWrite(in1, LOW); // повернути праворуч
    digitalWrite(in2, HIGH);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);

```

```

delay(400);
digitalWrite(in1, HIGH); // рухатися вперед
digitalWrite(in2, LOW);
digitalWrite(in3, HIGH);
digitalWrite(in4, LOW);
}
// function for turn left
void turnLeft(){
    digitalWrite(in1, HIGH); // повернути ліворуч
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
    delay(400);
    digitalWrite(in1, HIGH); // рухатися вперед
    digitalWrite(in2, LOW);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
}
// повертає відстань, виміряну датчиком відстані HC-SR04
int getDistance() {
    int echoTime; // змінна для зберігання часу, необхідного для
відскоку пінгу
    off an object
    int calcualtedDistance; // змінна для зберігання відстані, розрахованої за
часом відлуння
    // посилати ультразвуковий імпульс тривалістю 10 мс
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    echoTime = pulseIn(echoPin, HIGH); // використовуйте команду pulsein,
щоб побачити, скільки часу потрібно імпульсу, щоб повернутися до датчика
    calcualtedDistance = echoTime / 58.26; // обчислення відстані до об'єкта, від
якого відбився імпульс (половина часу відскоку, помножена на швидкість звуку)
    return calcualtedDistance; // надіслати назад розраховану відстань
}

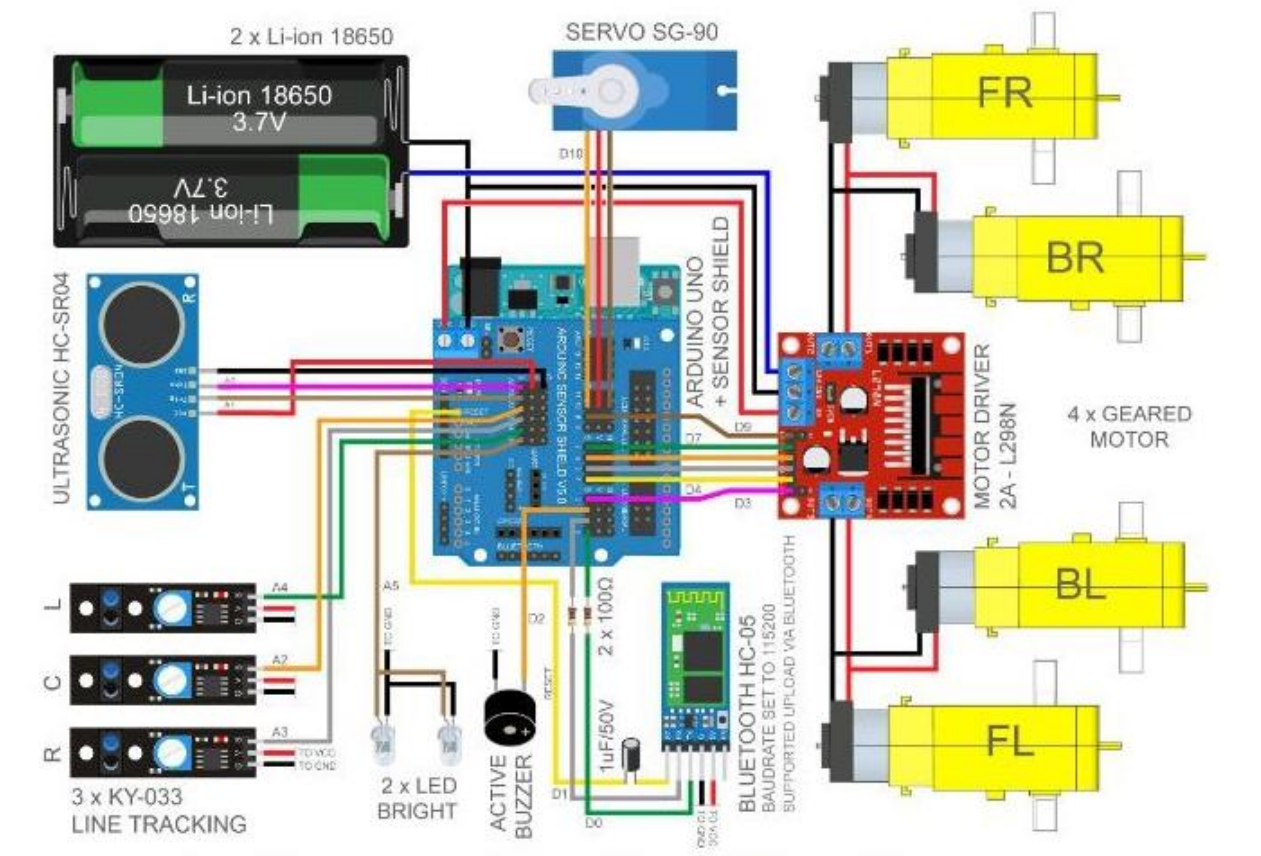
```

					<i>ІТС.4КІ-МС.0125.03-ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		66

ДОДАТОК Б

Схема підключення компонентів

На рисунку зображено повну схему підключення елементів до плати Arduino Uno.



Змн.	Арк.	№ докум.	Підпис	Дата

ITC.4KI-MC.0125.03-ПЗ

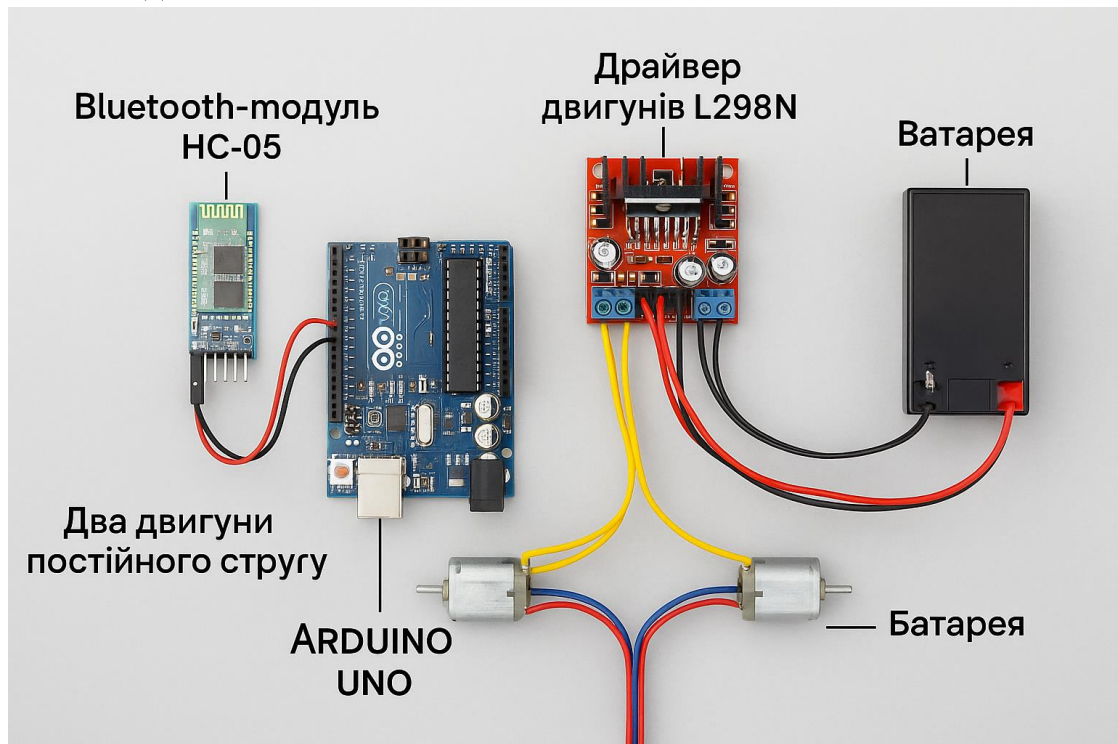
Арк.

67

ДОДАТОК В

Фото моделі автомобіля під час тестування

На фото зображено макет моделі під час тестування у приміщенні з активним Bluetooth-з'єднанням.



Змн.	Арк.	№ докум.	Підпис	Дата

ITC.4KI-МС.0125.03-ПЗ

Арк.

68