

Міністерство освіти і науки України

Луцький національний технічний університет

(повне найменування закладу вищої освіти)

Факультет комп'ютерних та інформаційних технологій

(повне найменування факультету)

Кафедра комп'ютерної інженерії та охоронних систем

(повне найменування кафедри)

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

**СИСТЕМИ НАВІГАЦІЇ ТА МОНІТОРИНГУ СЕРЕДОВИЩА
ДЛЯ МОБІЛЬНОГО РОБОТА-ПАТРУЛЯ НА БАЗІ ІОТ-
ТЕХНОЛОГІЙ**

**NAVIGATION AND ENVIRONMENT MONITORING SYSTEMS
FOR A MOBILE PATROL ROBOT BASED ON IOT
TECHNOLOGIES**

спеціальність 123 Комп'ютерна інженерія

(шифр і назва спеціальності)

освітня програма Комп'ютерна інженерія

(назва освітньої програми)

Виконав: здобувач вищої освіти
групи КІ-42
Горалюк Максим Валерійович

(підпис)

Керівник:
к.т.н., доцент
Гринюк Сергій Васильович

(підпис)

Кваліфікаційну роботу
допущено до захисту
« » червня 2026 р.
Гарант освітньої програми:
к.т.н., доцент
Лавренчук Світлана Василівна

(підпис)

Луцьк – 2026 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій
Кафедра комп'ютерної інженерії та безпеки
Ступінь вищої освіти: бакалавр
Галузь знань: 12 Інформаційні технології
Спеціальність: 123 Комп'ютерна інженерія
Освітня програма: «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

доц. Т. Терлецький

« 23 » 12 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Горалюку Максиму Валерійовичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи Системи навігації та моніторингу середовища для мобільного робота-патруля на базі IoT-технологій

Керівник роботи к.т.н., доцент Гринюк Сергій Васильович

затверджені наказом закладу вищої освіти від «20» грудня 2025 року № 536/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи 28.05.2026 р.

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

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

Вступ

Аналіз та обґрунтування технічних рішень для мобільного еко-патруля

Аналіз існуючих систем сигналізації

Проектування архітектури та апаратного забезпечення системи

Практична реалізація прототипу мобільного робота для моніторингу

Висновки

5. Перелік графічного (ілюстративного) матеріалу:

Компонування основних вузлів гусеничного робота-патруля

Послідовність складання фізичної моделі гусеничного робота-патруля

Розміщення сенсорного модуля метеомоніторингу на гусеничному роботі-патрулі

Схема підключення сенсорного модуля та апаратних компонентів робота-патруля

Схема IoT-передавання даних метеомоніторингу від робота-патруля до хмарної латформи

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
<i>Аналіз та обґрунтування технічних рішень для мобільного еко-патруля</i>	<i>Гринюк С.В., доцент</i>		
<i>Проектування архітектури та апаратного забезпечення системи</i>	<i>Гринюк С.В., доцент</i>		
<i>Практична реалізація прототипу мобільного робота для моніторингу</i>	<i>Гринюк С.В., доцент</i>		
<i>Нормоконтроль</i>	<i>Багнюк Н.В., доцент</i>		
<i>Гарант ОП</i>	<i>Лавренчук С.В., доцент</i>		
<i>Показник запозичень тексту</i>	%		
<i>Академічна доброчесність</i>	<i>Міскевич О.І., ст. викладач</i>		

7. Дата видачі завдання 23.12.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	<i>Огляд літератури із досліджуваної проблеми, аналіз предметної області та наявних рішень</i>	до 10.02.2026 р.	
2.	<i>Аналіз та обґрунтування технічних рішень для мобільного еко-патруля</i>	до 02.03.2026 р.	
3.	<i>Проектування архітектури та апаратного забезпечення системи</i>	до 02.04.2026 р.	
4.	<i>Практична реалізація прототипу мобільного робота для моніторингу</i>	до 10.04.2026 р.	
5.	<i>Представлення остаточного варіанту кваліфікаційної роботи керівникові</i>	до 10.05.2026 р.	
6.	<i>Нормоконтроль</i>	до 23.05.2026 р.	
7.	<i>Інструментальна перевірка на академічний плагіат</i>	до 25.05.2026 р.	
8.	<i>Здача кваліфікаційної роботи та всіх супровідних документів на кафедру</i>	до 28.05.2026 р.	

Здобувач вищої освіти

(підпис)

Максим ГОРАЛЮК

(прізвище, ініціали)

Керівник кваліфікаційної роботи

(підпис)

Сергій ГРИНЮК

(прізвище, ініціали)

АНОТАЦІЯ

Горалюк М. В. Системи навігації та моніторингу середовища для мобільного робота-патруля на базі IoT-технологій. Рукопис.

Кваліфікаційна робота бакалавра ОП «Комп'ютерна інженерія» спеціальності 123 Комп'ютерна інженерія. Луцький національний технічний університет. Луцьк, 2026.

Кваліфікаційна робота складається з вступу, трьох розділів, висновків, переліку використаних джерел, додатків.

У першому розділі проаналізовано сучасні мобільні платформи для екологічного моніторингу, розглянуто переваги гусеничного шасі порівняно з колісними та іншими типами платформ.

У другому розділі розроблено архітектуру та апаратне забезпечення системи. Описано принципову схему підключення сенсорів, драйвера двигунів L298N, ультразвукового датчика, модуля ESP8266 та системи живлення. Розроблено алгоритм керування рухом і навігацією, який забезпечує виявлення перешкод, зупинку платформи та вибір безпечного напрямку руху. Також подано логіку роботи модуля збору та обробки даних, зокрема методи зчитування показників з аналогових і цифрових датчиків та застосування фільтрації шумів.

У третьому розділі описано технічну реалізацію фізичного прототипу гусеничного робота-патруля. Розглянуто процес складання моделі, компонування основних вузлів, монтаж сенсорного модуля метеомоніторингу, розробку вбудованого програмного забезпечення та налаштування передавання даних до хмарного середовища. Проведено експериментальну перевірку роботи прототипу, зокрема оцінено стабільність зчитування показників сенсорів, працездатність алгоритму виявлення перешкод, автономність роботи, прохідність гусеничної платформи та стабільність IoT-зв'язку.

Ключові слова: мобільний робот-патруль, гусенична платформа, екологічний моніторинг, IoT, Arduino Uno, ESP8266, DHT22, BMP280, MQ-135, LDR, ультразвуковий датчик, L298N.

ANNOTATION

Horaliuk M. Navigation and environment monitoring systems for a mobile patrol robot based on iot technologies. Manuscript.

Bachelor's qualification work of the EP «Computer Engineering» specialty 123 Computer Engineering. Lutsk National Technical University. Lutsk, 2026.

The qualification work consists of an introduction, three sections, conclusions, a list of sources used, and appendices.

The first section analyzes modern mobile platforms for environmental monitoring, considers the advantages of a tracked chassis compared to wheeled and other types of platforms.

The second section develops the architecture and hardware of the system. The schematic diagram of connecting sensors, the L298N motor driver, the ultrasonic sensor, the ESP8266 module and the power system is described. A motion and navigation control algorithm is developed that provides obstacle detection, platform stopping and selection of a safe direction of movement. The logic of the data collection and processing module is also presented, in particular, methods for reading indicators from analog and digital sensors and the use of noise filtering.

The third section describes the technical implementation of the physical prototype of a tracked patrol robot. The process of assembling the model, arranging the main nodes, installing the weather monitoring sensor module, developing the embedded software and configuring data transmission to the cloud environment is considered. An experimental verification of the prototype was carried out, in particular, the stability of sensor readings, the performance of the obstacle detection algorithm, the autonomy of operation, the cross-country ability of the tracked platform, and the stability of IoT communication were assessed.

Keywords: mobile patrol robot, tracked platform, environmental monitoring, IoT, Arduino Uno, ESP8266, DHT22, BMP280, MQ-135, LDR, ultrasonic sensor, L298N.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 АНАЛІЗ ТА ОБҐРУНТУВАННЯ ТЕХНІЧНИХ РІШЕНЬ ДЛЯ МОБІЛЬНОГО ЕКО-ПАТРУЛЯ	9
1.1 Аналіз існуючих мобільних платформ для екологічного моніторингу	9
1.2 Огляд сенсорів для вимірювання метеопараметрів.....	14
1.3 Вибір технології IoT для передачі даних	20
1.4 Обґрунтування програмного стеку.....	24
РОЗДІЛ 2 ПРОЕКТУВАННЯ АРХІТЕКТУРИ ТА АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ	29
2.1 Розробка принципової схеми робота-патруля	29
2.2 Алгоритм керування рухом та навігацією	33
2.3 Розробка модуля збору та обробки даних.....	38
РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ПРОТОТИПУ МОБІЛЬНОГО РОБОТА ДЛЯ МОНІТОРИНГУ	43
3.1 Збірка фізичної моделі робота на базі гусеничної платформи	43
3.2 Монтаж сенсорного модуля метеомоніторингу та підключення апаратних компонентів	50
3.3 Розробка вбудованого програмного забезпечення та налаштування IoT- передавання даних	55
3.4 Експериментальна перевірка роботи прототипу та аналіз результатів.....	60
ВИСНОВКИ	67
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	71
ДОДАТКИ	Помилка! Закладку не визначено.

ВСТУП

Актуальність теми. Сучасний етап розвитку науково-технічного прогресу характеризується масштабною інтеграцією роботизованих систем у всі сфери людської діяльності. Особливої ваги набуває концепція мобільних моніторингових платформ, здатних виконувати завдання у середовищах, де присутність людини є небажаною або небезпечною. Традиційні стаціонарні системи екологічного моніторингу, попри свою точність, мають суттєвий недолік – низьку просторову розрізненість даних. Використання мобільних роботів-патрулів дозволяє отримати динамічну карту розподілу показників довкілля у реальному часі, що є критично важливим для оперативного реагування на техногенні чи екологічні загрози.

Синергія робототехніки та технологій Інтернету речей (IoT) відкриває нові можливості для побудови кіберфізичних систем. Застосування мікроконтролерів сімейства Arduino у поєднанні з інтелектуальними сенсорами та хмарними сервісами дозволяє створювати гнучкі, енергоефективні та економічно вигідні рішення для автономного патрулювання. Саме тому розробка системи навігації та моніторингу для мобільної платформи на базі IoT-технологій є актуальним науково-прикладним завданням.

Мета роботи полягає у проектуванні, програмно-апаратній реалізації та дослідженні системи автономного патрулювання та екологічного моніторингу на основі гусеничної мобільної платформи, що забезпечує збір, обробку та дистанційну передачу параметрів навколишнього середовища.

Об'єкт дослідження – процеси автономної навігації та дистанційного моніторингу показників навколишнього середовища за допомогою мобільних роботизованих засобів.

Предмет дослідження – методи, алгоритми та програмно-апаратні засоби побудови системи керування мобільним роботом-патрулем та інтеграції сенсорної мережі в екосистему IoT.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- проаналізувати сучасні підходи до побудови мобільних робототехнічних комплексів та систем моніторингу середовища;
- обґрунтувати вибір апаратної архітектури на базі контролера Arduino Uno та відповідного сенсорного забезпечення;
- розробити алгоритмічне забезпечення для навігації мобільної платформи та обходу перешкод;
- реалізувати програмний комплекс для збору метеоданих та їх передачі через IoT-шлюз до хмарного інтерфейсу;
- провести експериментальну апробацію розробленого прототипу в реальних умовах та оцінити точність отриманих результатів.

Методи дослідження. Для розв'язання поставлених завдань використано методи теорії автоматичного керування, принципи побудови мікропроцесорних систем, методи об'єктно-орієнтованого програмування та експериментальні методи дослідження параметрів фізичних середовищ.

Наукова новизна та практичне значення отриманих результатів полягає у створенні комплексної архітектури мобільної метеостанції, яка поєднує високу прохідність гусеничного шасі з інтелектуальною обробкою екологічних показників. Розроблена система може бути використана як база для створення промислових засобів моніторингу складських приміщень, теплиць або прилеглих територій підприємств.

РОЗДІЛ 1

АНАЛІЗ ТА ОБҐРУНТУВАННЯ ТЕХНІЧНИХ РІШЕНЬ ДЛЯ МОБІЛЬНОГО ЕКО-ПАТРУЛЯ

1.1 Аналіз існуючих мобільних платформ для екологічного моніторингу

Екологічний моніторинг є важливим напрямом застосування сучасних мобільних роботизованих систем, оскільки потребує регулярного збирання даних про стан довкілля, параметри повітря, ґрунту, води, рослинності або техногенно навантажених територій. Традиційні методи екологічного контролю часто передбачають використання стаціонарних датчиків або ручне збирання показників, що обмежує просторове охоплення, потребує значних часових витрат і не завжди є безпечним для людини. У сучасних дослідженнях підкреслюється, що роботизовані системи для екологічного моніторингу дають змогу виконувати вимірювання у важкодоступних або небезпечних зонах, забезпечувати частіший збір даних і підвищувати гнучкість моніторингових процесів [1].

Мобільні платформи для екологічного моніторингу можна умовно поділити на кілька основних груп: колісні наземні роботи, гусеничні наземні роботи, крокуючі роботи, безпілотні літальні апарати та гібридні роботизовані системи. Кожен тип платформи має власні переваги, обмеження та сферу доцільного використання. Наприклад, безпілотні літальні апарати ефективні для огляду великих територій зверху, але мають обмежений час польоту, залежать від погодних умов і не завжди можуть виконувати контактні вимірювання біля поверхні землі. Колісні роботи є простими, енергоефективними та достатньо швидкими на рівних поверхнях, однак їхня прохідність знижується на пухкому ґрунті, траві, кам'янистих ділянках або нерівному рельєфі. Крокуючі роботи краще пристосовані до складного рельєфу, але мають складнішу механічну конструкцію, вищу вартість і потребують складніших алгоритмів керування [1].

Загальну класифікацію мобільних платформ, які можуть застосовуватися для екологічного моніторингу, доцільно подати на рисунку 1.1.

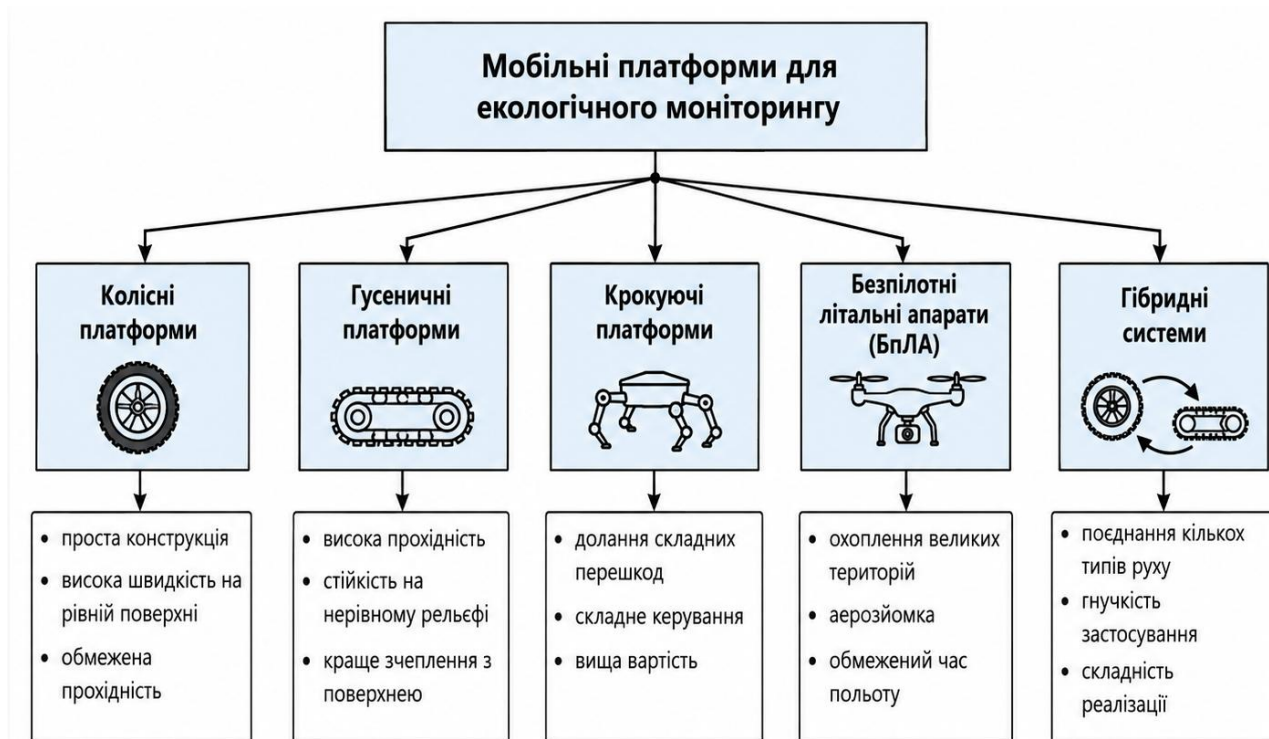


Рисунок 1.1 – Класифікація мобільних платформ для екологічного моніторингу [1]

Для екологічного моніторингу наземного типу особливе значення мають саме наземні мобільні роботи, оскільки вони можуть переміщуватися безпосередньо в контрольованому середовищі та збирати дані на рівні поверхні. Такі платформи можуть оснащуватися сенсорами температури, вологості, якості повітря, газів, пилу, освітленості, вологості ґрунту, камерами, GPS-модулями, інерційними датчиками та засобами бездротового зв'язку. В огляді роботизованих систем для екологічного моніторингу зазначено, що сенсори в таких системах повинні не лише забезпечувати точність вимірювань, а й бути інтегрованими з мобільною платформою та працювати в режимі, близькому до реального часу [1].

Колісні мобільні платформи є найпоширенішими у навчальних і прототипних роботизованих системах. Їхніми перевагами є простота конструкції, доступність комплектувальних, відносно низьке енергоспоживання, легкість керування та висока швидкість руху на рівних поверхнях. Такі платформи доцільно використовувати в приміщеннях, лабораторіях, теплицях, офісних

просторах або на підготовлених майданчиках. Проте під час роботи в природному середовищі колісне шасі може втрачати ефективність через пробуксовування, просідання коліс у м'який ґрунт, втрату зчеплення на вологій траві або складність подолання невеликих перешкод.

Гусеничні мобільні платформи мають іншу конструктивну логіку. На відміну від колісного шасі, гусениці забезпечують більшу площу контакту з поверхнею, завдяки чому вага робота розподіляється рівномірніше. Це зменшує ризик просідання в м'який ґрунт, покращує зчеплення з поверхнею та підвищує здатність платформи долати нерівності. У сучасних дослідженнях гусеничні роботи розглядаються як ефективні платформи для роботи на складному, м'якому або нерівному рельєфі, де колісні роботи можуть втрачати тягу, ковзати або застрягати [2, 3].

Перевага гусеничного шасі особливо помітна в умовах екологічного моніторингу відкритих територій. Робот може рухатися по ґрунтових ділянках, траві, піску, дрібному камінню, вологій поверхні або нерівному рельєфу. Для системи екологічного моніторингу це важливо, оскільки точки вимірювання не завжди розташовані на рівній або підготовленій поверхні. Наприклад, під час контролю стану пришкільної ділянки, тепличного господарства, промислової території, узбіччя, садової зони або ділянки після опадів колісна платформа може мати нижчу стабільність, тоді як гусенична конструкція забезпечує кращу прохідність.

Структурно гусеничне шасі складається з рами, двох гусеничних стрічок, ведучих коліс, опорних роликів, двигунів із редукторами, блока живлення та кріплень для сенсорних модулів. Така конструкція дозволяє розміщувати на платформі не лише мікроконтролер і модулі зв'язку, а й додаткові сенсори для моніторингу навколишнього середовища. Доцільним є встановлення датчиків у верхній частині корпусу або на спеціальній панелі, щоб зменшити ризик їхнього забруднення під час руху по ґрунту. Камера або датчик відстані можуть встановлюватися у передній частині робота для аналізу траєкторії руху та виявлення перешкод.

Узагальнену структуру гусеничної мобільної платформи для екологічного моніторингу доцільно подати на рисунку 1.2.



Рисунок 1.2 - Узагальнена структура гусеничної мобільної платформи для екологічного моніторингу [2]

Важливою перевагою гусеничного шасі є підвищена стійкість під час руху. Завдяки більшій площі контакту з поверхнею платформа краще утримує рівновагу на нерівних ділянках, а центр мас може бути розміщений нижче, ніж у деяких колісних або крокуючих конструкціях. Це позитивно впливає на стабільність сенсорних вимірювань, оскільки різкі нахили, вібрації та удари можуть погіршувати точність показників. Дослідження гусеничних роботів із пасивною підвіскою показують, що вдосконалення механічної конструкції може покращувати контакт із нерівною поверхнею та підвищувати прохідність у складних умовах [2].

Разом із тим гусеничні платформи мають певні недоліки, які необхідно враховувати під час проектування. Вони можуть мати більше енергоспоживання порівняно з простими колісними платформами, складнішу механічну

конструкцію, більшу масу та вищий рівень тертя під час поворотів. Крім того, під час руху по нерівному рельєфу можуть виникати коливання, ковзання та відхилення від заданої траєкторії. У дослідженнях 2025 року зазначається, що для гусеничних роботів на нерівній поверхні актуальними залишаються задачі стабілізації руху, компенсації ковзання та точного відстеження траєкторії [4].

Для екологічного моніторингу гусеничне шасі доцільно поєднувати з модульною сенсорною системою. Наприклад, базовий варіант може містити датчик температури та вологості повітря, датчик газів або диму, датчик освітленості, ультразвуковий датчик відстані, GPS-модуль і Wi-Fi або GSM-модуль для передавання даних. У розширеному варіанті система може доповнюватися камерою, сенсором пилу, датчиком вологості ґрунту або метеорологічним модулем. Така структура дозволяє виконувати роботу не лише як рухому платформу, а як повноцінний мобільний вузол збору екологічних даних.

Порівняльну характеристику основних типів мобільних платформ для екологічного моніторингу подано в таблиці 1.1.

Таблиця 1.1 – Порівняння мобільних платформ для екологічного моніторингу

Тип платформи	Переваги	Обмеження	Доцільне використання
Колісна платформа	Проста конструкція, висока швидкість, низьке енергоспоживання	Обмежена прохідність на м'якому та нерівному ґрунті	Приміщення, лабораторії, теплиці, рівні майданчики
Гусенична платформа	Висока прохідність, краще зчеплення, стійкість на нерівностях	Більша маса, складніша конструкція, вище енергоспоживання	Ґрунт, трава, нерівні ділянки, відкриті території
Крокуюча платформа	Здатність долати складний рельєф і перешкоди	Висока складність керування, значна вартість	Кам'янисті, лісові, важкодоступні території
Дрон	Швидке охоплення великих площ, зйомка зверху	Обмежений час польоту, залежність від погоди, складність контактних вимірювань	Аерофотозйомка, огляд великих територій
Гібридна система	Поєднання переваг кількох платформ	Висока складність реалізації	Комплексний моніторинг великих і різномірних територій

Аналіз існуючих мобільних платформ показує, що вибір типу шасі повинен залежати від умов експлуатації, характеру контрольованої території, вимог до автономності, маси сенсорного обладнання та необхідної точності переміщення. Якщо робот призначений для руху в приміщенні або на рівній поверхні, колісне шасі є достатнім і економічно доцільним. Якщо ж система має працювати в умовах відкритого середовища, де можливі нерівності, трава, ґрунт, дрібні перешкоди або волога поверхня, гусеничне шасі має суттєві переваги [5].

Таким чином, для мобільної системи екологічного моніторингу гусеничне шасі є обґрунтованим конструктивним рішенням. Воно забезпечує кращу прохідність, стійкість, зчеплення з поверхнею та можливість роботи на складнішому рельєфі порівняно з базовими колісними платформами. Незважаючи на більшу масу та потенційно вище енергоспоживання, гусенична конструкція підвищує надійність руху в реальних умовах навколишнього середовища. Саме тому для проектування мобільного робота екологічного моніторингу доцільно обрати гусеничну платформу як основу, на якій можуть бути розміщені сенсорний блок, модуль навігації, система зв'язку та джерело автономного живлення.

1.2 Огляд сенсорів для вимірювання метеопараметрів

Для мобільної гусеничної платформи екологічного моніторингу важливим завданням є не лише переміщення контрольованою територією, а й збирання достовірних даних про стан навколишнього середовища. Саме тому сенсорний блок є однією з основних складових системи. Він забезпечує вимірювання метеопараметрів, оцінювання якості повітря, фіксацію рівня освітленості та передавання отриманих значень до блока керування для подальшої обробки. У межах розроблюваної системи доцільно використати набір доступних сенсорів: DHT22 для вимірювання вологості й температури повітря, BMP280 для визначення атмосферного тиску та орієнтовної висоти, MQ-135 для оцінювання якості повітря, а також LDR-фоторезистор для визначення рівня освітленості.

Сенсор DHT22, також відомий як AM2302, є цифровим датчиком температури та відносної вологості. Він містить ємнісний елемент для вимірювання вологості, термістор для вимірювання температури та вбудований контролер, який формує цифровий вихідний сигнал. Згідно з технічною документацією, AM2302 працює від напруги 3,3-5,5 В, вимірює вологість у діапазоні 0-100 % RH і температуру в діапазоні приблизно від -40 до +80° С [6]. Це робить його придатним для мобільних систем екологічного моніторингу, де необхідно контролювати мікроклімат на різних ділянках території.

У практичній реалізації DHT22 доцільно застосовувати для визначення вологості повітря в зоні руху робота. Показники вологості можуть бути корисними під час моніторингу теплиць, складських приміщень, навчальних лабораторій, відкритих ділянок або територій, де важливо контролювати зміну мікроклімату. Перевагою цього сенсора є просте підключення до мікроконтролерів Arduino, ESP32 або Raspberry Pi через цифровий інтерфейс. Недоліком є порівняно невисока швидкодія, тому сенсор краще використовувати для періодичного збирання даних, а не для високочастотних вимірювань у реальному часі [6].

Датчик BMP280 є цифровим барометричним сенсором, який використовується для вимірювання атмосферного тиску та розрахунку орієнтовної висоти. Він розроблений компанією Bosch Sensortec і призначений для мобільних та енергоефективних пристроїв. Згідно з технічним описом, BMP280 вимірює тиск у діапазоні 300-1100 гПа, що відповідає приблизному діапазону висот від +9000 м до -500 м відносно рівня моря. Також датчик має малий розмір, низьке енергоспоживання та підтримує цифрові інтерфейси I2C і SPI [7].

У мобільній платформі для екологічного моніторингу BMP280 може використовуватися для фіксації змін атмосферного тиску та приблизної оцінки висотних змін під час руху. Для наземного гусеничного робота такий сенсор не є основним навігаційним елементом, проте він розширює функції метеомоніторингу. Наприклад, під час роботи на відкритій території зміна

атмосферного тиску може бути використана як додатковий параметр для аналізу погодних умов. Крім того, BMP280 є компактним і не потребує складного підключення, що дозволяє легко інтегрувати його в загальний сенсорний блок.

Сенсор MQ-135 належить до напівпровідникових газових датчиків і використовується для оцінювання якості повітря. Він має чутливість до аміаку, оксидів азоту, бензолних парів, диму, CO₂ та інших газових домішок. У технічній документації MQ-135 зазначено, що сенсор призначений для систем контролю якості повітря в будівлях, офісах, побутових і промислових умовах [8]. Його робота ґрунтується на зміні електричного опору чутливого шару під дією газів, після чого вихідний сигнал може бути зчитаний мікроконтролером через аналоговий вхід.

Для мобільної платформи екологічного моніторингу MQ-135 може використовуватися як індикатор забруднення повітря або наявності потенційно шкідливих домішок. Наприклад, робот може переміщуватися територією й передавати значення з датчика на IoT-платформу. Це дозволить порівнювати якість повітря в різних точках маршруту. Водночас потрібно враховувати, що MQ-135 не є високоточним лабораторним газоаналізатором. Його показники залежать від калібрування, температури, вологості, часу прогріву та типу газової суміші. Тому в навчальному або прототипному проєкті MQ-135 доцільно використовувати для відносної оцінки якості повітря, тобто для виявлення змін або перевищення умовного порогу, а не для точного визначення концентрації кожного газу.

Для вимірювання рівня освітленості може використовуватися LDR-фоторезистор. LDR є світлозалежним резистором, електричний опір якого змінюється залежно від інтенсивності світла. У темряві його опір зростає, а при збільшенні освітленості – зменшується [9]. Завдяки простоті, низькій вартості та легкому підключенню LDR часто використовують у навчальних і прототипних системах автоматичного керування освітленням, робототехніці та пристроях екологічного моніторингу.

У складі мобільного робота LDR може виконувати допоміжну функцію моніторингу освітленості. Наприклад, система може визначати, чи працює робот у добре освітленій зоні, у затемненому приміщенні або на відкритій території. Ці дані можуть бути корисними для аналізу умов середовища, автоматичного вмикання підсвічування, налаштування камери або оцінювання добових змін освітленості. Недоліком LDR є невисока точність і залежність показників від конкретної схеми підключення, тому для точних вимірювань освітленості в люксах краще використовувати спеціалізовані цифрові датчики. Проте для базового навчального проєкту LDR є достатнім рішенням.

Загальну структуру сенсорного блока для вимірювання метеопараметрів та параметрів середовища доцільно подати на рисунку 1.3.

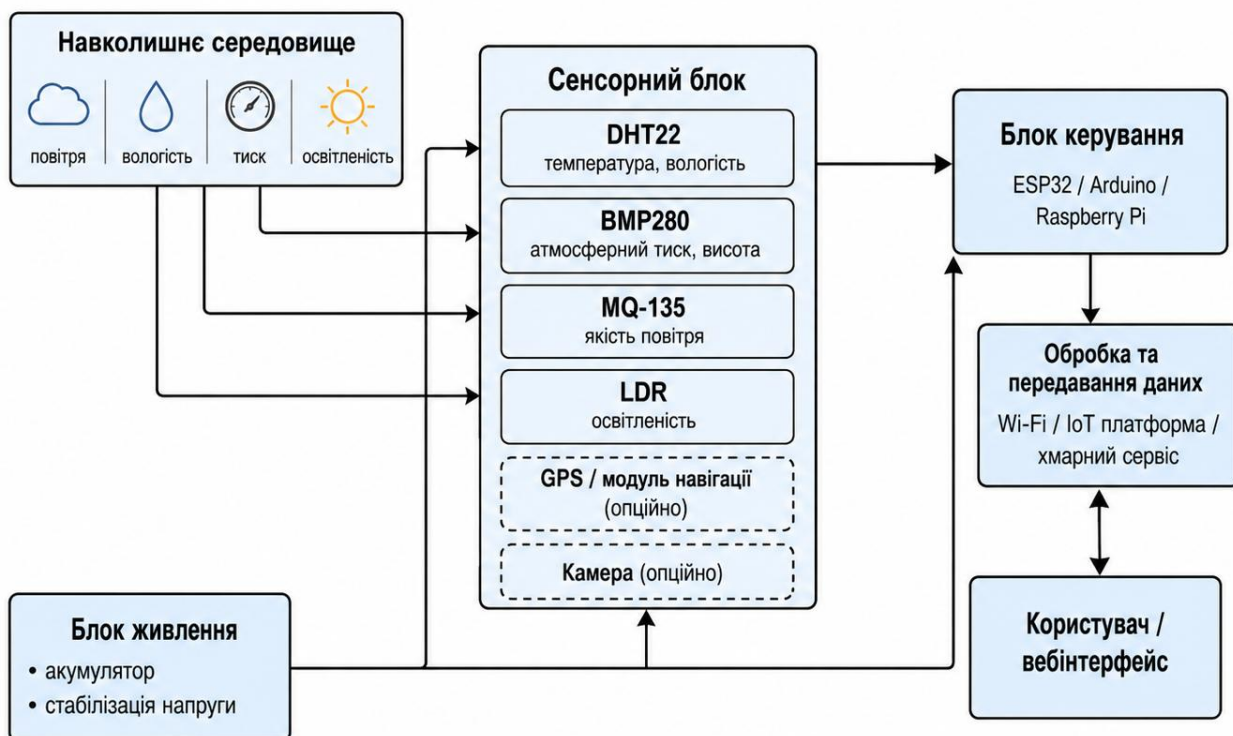


Рисунок 1.3 – Структура сенсорного блока мобільної платформи екологічного моніторингу [9]

Узагальнені технічні характеристики сенсорів, які можуть бути використані в системі екологічного моніторингу, подано в таблиці 1.2.

Таблиця 1.2 – Технічні характеристики сенсорів для мобільної платформи екологічного моніторингу

Сенсор	Вимірюваний параметр	Основні характеристики	Інтерфейс підключення	Призначення в системі
DHT22/AM2302	Вологість і температура повітря	Вологість 0-100 % RH; температура приблизно -40...+80 °С; живлення 3,3-5,5 В	Цифровий однопровідний сигнал	Контроль мікроклімату під час руху платформи
BMP280	Атмосферний тиск, орієнтовна висота	Тиск 300-1100 гПа; висока компактність; низьке енергоспоживання	I2C / SPI	Оцінювання метеоумов і змін атмосферного тиску
MQ-135	Якість повітря, газові домішки	Чутливість до NH ₃ , NO _x , бензольних парів, диму, CO ₂ та інших домішок	Аналоговий вихід / цифровий вихід на модулі	Виявлення погіршення якості повітря
LDR	Освітленість	Опір змінюється залежно від інтенсивності світла; просте підключення	Аналоговий сигнал через подільник напруги	Визначення рівня освітленості середовища

З погляду практичної реалізації сенсори доцільно розміщувати таким чином, щоб вони отримували достовірні показники та не зазнавали механічного пошкодження під час руху гусеничної платформи. DHT22 та BMP280 бажано встановити у верхній частині корпусу або на невеликій стійці, щоб зменшити вплив тепла від двигунів і електроніки. MQ-135 доцільно розмістити так, щоб повітря вільно надходило до чутливого елемента, але сам сенсор був захищений від пилу, вологи та прямого механічного контакту. LDR варто встановити у верхній або передній частині платформи, оскільки він повинен реагувати на зовнішнє освітлення.

Схематичне розміщення сенсорів на гусеничній мобільній платформі доцільно подати на рисунку 1.4.

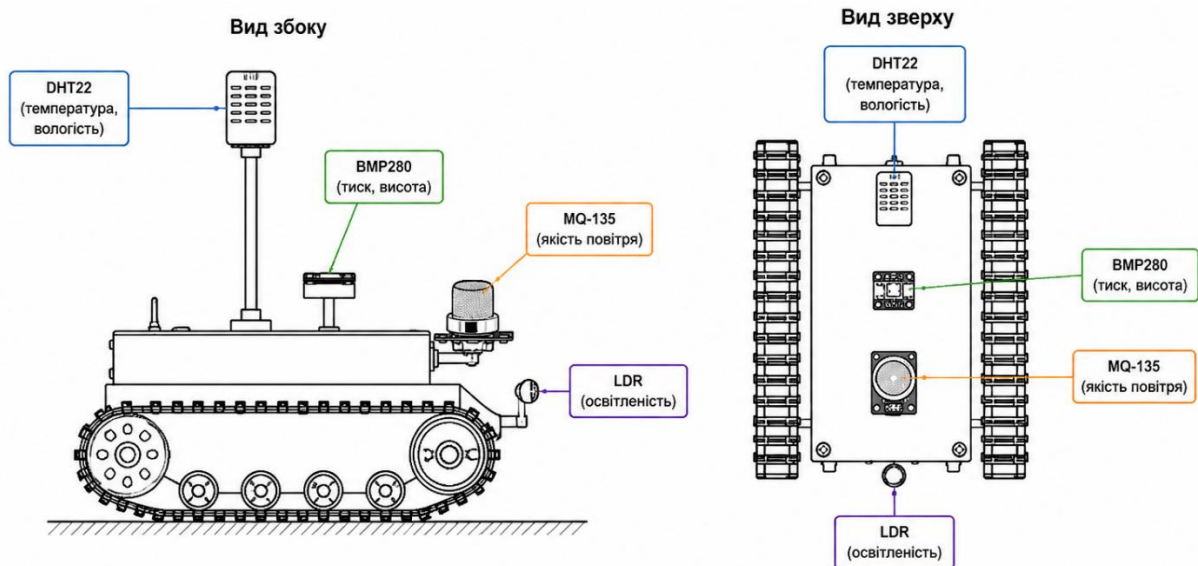


Рисунок 1.4 – Розміщення сенсорів DHT22, BMP280, MQ-135 та LDR на гусеничній мобільній платформі [9]

Усі зазначені сенсори можуть бути підключені до мікроконтролера ESP32 або Arduino, однак для IoT-системи перевагу доцільно надати ESP32, оскільки він має вбудовані засоби бездротового зв'язку Wi-Fi та Bluetooth. Це дає змогу не лише зчитувати показники сенсорів, а й передавати їх на сервер, вебпанель або хмарну платформу. Наприклад, дані DHT22 можуть передаватися як температура та вологість, BMP280 – як атмосферний тиск і висота, MQ-135 – як умовний індекс якості повітря, а LDR – як рівень освітленості. Після обробки ці дані можуть бути представлені у вигляді таблиці, графіків або повідомлень про перевищення заданих порогів.

Важливо враховувати, що частина сенсорів має різну природу вихідного сигналу. DHT22 передає цифрові дані, BMP280 працює через I2C або SPI, MQ-135 зазвичай формує аналоговий сигнал, а LDR підключається через подільник напруги до аналогового входу. Тому під час розробки системи потрібно передбачити правильне підключення, узгодження рівнів напруги, фільтрацію похибок, калібрування показників і програмну обробку даних. Особливо це стосується MQ-135, для якого необхідний час прогріву та налаштування порогових значень.

В результаті поєднання сенсорів DHT22, BMP280, MQ-135 та LDR дозволяє сформувати базовий сенсорний блок мобільної гусеничної платформи для екологічного моніторингу. DHT22 забезпечує контроль вологості та температури, BMP280 – вимірювання атмосферного тиску та орієнтовної висоти, MQ-135 – оцінювання якості повітря, а LDR – визначення рівня освітленості. Такий набір сенсорів є доступним, сумісним із поширеними мікроконтролерними платформами та достатнім для створення навчального або прототипного зразка мобільної системи моніторингу навколишнього середовища.

1.3 Вибір технології IoT для передачі даних

Передавання даних є однією з ключових функцій мобільної платформи екологічного моніторингу, оскільки зібрані сенсорами показники мають бути доступними для подальшого аналізу, збереження та відображення користувачу. У разі використання Arduino Uno як базового мікроконтролера потрібно враховувати, що ця плата не має вбудованих засобів бездротового зв'язку, тому для реалізації IoT-функцій необхідно застосовувати додаткові комунікаційні модулі. Arduino Uno Rev3 має 14 цифрових входів/виходів, 6 аналогових входів і побудована на мікроконтролері ATmega328P, що робить її зручною для зчитування даних із сенсорів, але для передавання інформації потрібні зовнішні модулі зв'язку [10].

У межах мобільної гусеничної платформи для екологічного моніторингу доцільно розглянути три основні способи бездротового передавання даних для Arduino Uno: Bluetooth-модуль, Wi-Fi-модуль ESP8266 та GSM-модуль. Кожен із цих варіантів має власні переваги та обмеження, які пов'язані з дальністю зв'язку, енергоспоживанням, складністю налаштування, потребою в додатковій інфраструктурі та можливістю передавання даних на віддалений сервер.

Першим варіантом є використання Bluetooth-модуля, наприклад HC-05. Такий модуль працює як послідовний бездротовий інтерфейс і дає змогу

передавати дані між Arduino Uno та смартфоном, ноутбуком або іншим пристроєм у межах невеликої відстані. У технічній документації HC-05 зазначено, що модуль підтримує Bluetooth V2.0+EDR, використовує профіль SPP для прозорого послідовного з'єднання та працює в діапазоні 2,4 ГГц [11]. Для мобільної платформи екологічного моніторингу Bluetooth є зручним під час локального тестування, налаштування та передавання даних на пристрій оператора поблизу робота. Проте цей варіант має обмежену дальність і не є повноцінним IoT-рішенням для віддаленого моніторингу через інтернет.

Другим варіантом є підключення Wi-Fi-модуля ESP8266 до Arduino Uno. ESP8266EX підтримує стандарти Wi-Fi 802.11 b/g/n у діапазоні 2,4 ГГц, може працювати в режимах Station і SoftAP, має вбудований TCP/IP-стек і призначений для мобільних, носимих та IoT-застосувань [12]. У системі екологічного моніторингу ESP8266 дає змогу передавати дані з Arduino Uno до локального сервера, вебінтерфейсу, хмарної платформи або MQTT-брокера. Такий підхід є доцільним, якщо мобільна платформа працює в зоні покриття Wi-Fi-мережі, наприклад у лабораторії, навчальному корпусі, теплиці, на території підприємства або в межах локальної дослідної ділянки.

Третім варіантом є використання GSM/GPRS-модуля, наприклад SIM800L. Такий модуль забезпечує передавання даних через мобільну мережу та є корисним у випадках, коли робот працює поза зоною Wi-Fi-покриття. Згідно з технічним описом, SIM800L підтримує чотиридіапазонний GSM/GPRS 850/900/1800/1900 МГц, GPRS class 12/10, SMS, TCP/UDP, HTTP та керування через AT-команди [13]. Для екологічного моніторингу на відкритих територіях GSM-модуль має суттєву перевагу, оскільки дає змогу передавати дані на віддалений сервер без прив'язки до локальної Wi-Fi-мережі. Водночас потрібно враховувати необхідність SIM-карти, покриття мобільного оператора, стабільного живлення та вищого енергоспоживання під час передавання даних.

Порівняння основних способів зв'язку для Arduino Uno подано в таблиці 1.3.

Таблиця 1.3 – Порівняння способів зв'язку для Arduino Uno у системі екологічного моніторингу

Спосіб зв'язку	Приклад модуля	Переваги	Обмеження	Доцільне використання
Bluetooth	HC-05	Просте підключення, низька вартість, зручний для локального тестування	Мала дальність, відсутність прямого доступу до інтернету	Налаштування робота, передавання даних на смартфон або ноутбук поблизу
Wi-Fi	ESP8266	Передавання даних у локальну мережу або хмару, підтримка TCP/IP, зручність для IoT	Потрібна Wi-Fi-мережа, складність налаштування порівняно з Bluetooth	Лабораторія, навчальний корпус, теплиця, локальна дослідна територія
GSM/GPRS	SIM800L	Передавання даних через мобільну мережу, робота поза Wi-Fi-покриттям	Потрібна SIM-карта, залежність від мобільного покриття, більше енергоспоживання	Відкриті території, польовий моніторинг, віддалені об'єкти

З погляду практичної реалізації Bluetooth є найпростішим способом передавання даних, однак його можливості обмежені локальним зв'язком. Він може бути корисним на етапі перевірки працездатності сенсорів DHT22, BMP280, MQ-135 та LDR, коли потрібно швидко отримувати значення на смартфон або комп'ютер. Проте для системи екологічного моніторингу, яка повинна працювати як IoT-пристрій, Bluetooth не забезпечує достатньої гнучкості, оскільки не дозволяє безпосередньо передавати дані на вебсервер або хмарну платформу без додаткового проміжного пристрою.

Wi-Fi через ESP8266 є найбільш збалансованим варіантом для навчального або прототипного проєкту на Arduino Uno. Цей модуль дає змогу передавати дані у мережу, реалізувати HTTP-запити, взаємодіяти з MQTT-брокером або відправляти показники на IoT-платформу. У такій архітектурі Arduino Uno виконує функцію зчитування даних із сенсорів і формування пакетів вимірювань, а ESP8266 забезпечує мережевий обмін. Перевагою цього підходу є доступність компонентів, велика кількість прикладів реалізації та можливість побудови вебінтерфейсу для перегляду даних.

GSM/GPRS-модуль доцільно використовувати тоді, коли мобільна платформа працює за межами локальної Wi-Fi-мережі. Наприклад, якщо робот виконує моніторинг відкритої території, сільськогосподарської ділянки, віддаленої промислової зони або території, де немає доступу до маршрутизатора, SIM800L може забезпечити передавання даних через мобільну мережу. Однак у такому випадку потрібно передбачити більш потужне джерело живлення, оскільки GSM-модулі можуть споживати значний струм у моменти передавання. Також необхідно враховувати стабільність мобільного покриття та витрати на обслуговування SIM-карти.

Для розроблюваної гусеничної мобільної платформи екологічного моніторингу найбільш доцільним варіантом є використання Wi-Fi-модуля ESP8266 у поєднанні з Arduino Uno. Такий вибір пояснюється тим, що платформа орієнтована на навчальну або прототипну реалізацію, де важливими є доступність, простота інтеграції, можливість передавання даних на локальний сервер або IoT-платформу та підтримка поширених мережевих протоколів. Bluetooth можна використовувати як допоміжний канал для локального налаштування, а GSM-модуль – як перспективний варіант модернізації для роботи на віддалених територіях.

Вибір технології IoT для передавання даних залежить від умов експлуатації мобільної платформи. Якщо система працює в межах лабораторії або навчального приміщення, достатньо Bluetooth або Wi-Fi. Якщо потрібно реалізувати повноцінний IoT-моніторинг із передаванням даних на сервер, доцільнішим є ESP8266. Якщо ж платформа має працювати на віддалених відкритих ділянках без Wi-Fi-покриття, доцільно використовувати GSM/GPRS-модуль. Для базового варіанта системи екологічного моніторингу оптимальним рішенням є Wi-Fi через ESP8266, оскільки він забезпечує найкраще співвідношення між функціональністю, вартістю, простотою реалізації та можливістю інтеграції з IoT-сервісами.

1.4 Обґрунтування програмного стеку

Програмний стек мобільної гусеничної платформи для екологічного моніторингу має забезпечувати зчитування показників із сенсорів, первинну обробку даних, керування обміном між Arduino Uno та комунікаційним модулем, передавання інформації через IoT-канал і подальше відображення результатів для користувача. Оскільки в попередньому підпункті було обґрунтовано доцільність використання Arduino Uno у поєднанні з Wi-Fi-модулем ESP8266, програмний стек доцільно будувати навколо середовища Arduino IDE, мови програмування Arduino C/C++, бібліотек для роботи із сенсорами та засобів мережевої взаємодії.

Базовим середовищем розробки для цього проєкту доцільно обрати Arduino IDE. Це середовище підтримує написання, компіляцію та завантаження програмного коду на мікроконтролерні плати Arduino, а також забезпечує роботу з бібліотеками, прикладами та послідовним монітором. Мова програмування Arduino ґрунтується на синтаксисі C/C++ і містить зручні функції для роботи з цифровими та аналоговими входами, таймерами, послідовним портом і периферійними пристроями [14]. Такий підхід є доцільним для навчального або прототипного проєкту, оскільки дає змогу швидко реалізувати алгоритм зчитування даних із DHT22, BMP280, MQ-135 та LDR, а також передавання сформованих значень до модуля зв'язку.

Важливою перевагою Arduino IDE є підтримка бібліотек. Бібліотеки розширюють можливості середовища розробки та дають змогу використовувати готові програмні модулі для роботи із сенсорами, дисплеями, мережевими модулями та протоколами обміну даними. В офіційній документації Arduino зазначено, що нові бібліотеки можна встановлювати через Library Manager, що спрощує підключення готових рішень до проєкту [15]. Для розроблюваної системи це важливо, оскільки робота з DHT22, BMP280 та IoT-платформою може бути реалізована не лише вручну, а й через перевірені бібліотеки, що зменшує ймовірність помилок у програмному коді.

Програмна частина системи повинна складатися з кількох логічних рівнів. Перший рівень відповідає за зчитування даних із сенсорів. Для DHT22 необхідно реалізувати отримання температури та вологості, для BMP280 – атмосферного тиску та орієнтовної висоти, для MQ-135 – зчитування аналогового значення якості повітря, а для LDR – визначення рівня освітленості через аналоговий вхід Arduino Uno. Другий рівень виконує первинну обробку даних: перевірку коректності значень, усереднення кількох вимірювань, перетворення аналогових сигналів у зручний для аналізу формат і формування пакета даних. Третій рівень забезпечує передавання даних через ESP8266 до IoT-платформи або локального сервера. Четвертий рівень пов'язаний із візуалізацією результатів для користувача.

Для роботи з метеосенсорами доцільно застосовувати готові бібліотеки Arduino. Наприклад, для DHT22 можна використати бібліотеку DHT sensor library, яка спрощує зчитування температури та вологості. Для BMP280 доцільно використати бібліотеку, що підтримує інтерфейс I2C або SPI, оскільки цей сенсор передає цифрові дані. MQ-135 і LDR можуть оброблятися через стандартну функцію `analogRead()`, оскільки ці компоненти у типовій схемі підключення формують аналоговий сигнал. Такий підхід забезпечує поєднання простоти реалізації та достатньої функціональності для навчальної мобільної платформи екологічного моніторингу.

Окрему увагу потрібно приділити програмній взаємодії Arduino Uno з ESP8266. Оскільки Arduino Uno не має вбудованого Wi-Fi, ESP8266 може використовуватися як зовнішній комунікаційний модуль. Передавання даних між Arduino Uno та ESP8266 може здійснюватися через послідовний інтерфейс UART. У простішій реалізації Arduino Uno формує текстовий рядок із показниками сенсорів, наприклад у форматі «`temperature=24.5;humidity=61;pressure=1008;air=380;light=720`», після чого передає його до ESP8266. Далі ESP8266 надсилає ці дані на сервер або IoT-платформу через Wi-Fi-з'єднання.

Для передавання даних до вебсервера можуть використовуватися HTTP-запити. Бібліотека `ArduinoHttpClient` призначена для виконання

HTTP-запитів типу GET, POST, PUT і DELETE до вебсервера, а також підтримує обмін через WebSocket [16]. У межах системи екологічного моніторингу HTTP-запит може використовуватися для надсилання поточних значень сенсорів до локального сервера, хмарного сервісу або спеціально створеного вебзастосунку. Наприклад, кожен пакет вимірювань може передаватися як набір параметрів: температура, вологість, тиск, рівень газових домішок та освітленість.

Як IoT-платформу для навчального прототипу доцільно розглядати ThingSpeak. В офіційній документації Arduino зазначено, що ThingSpeak є аналітичною IoT-платформою, яка дає змогу агрегувати, візуалізувати й аналізувати потоки даних у хмарі [17]. Для мобільної платформи екологічного моніторингу це є зручним рішенням, оскільки дані з сенсорів можна передавати у відповідні поля каналу, а потім переглядати їх у вигляді графіків. Наприклад, окремі поля можуть відповідати температурі, вологості, тиску, показнику MQ-135 та рівню освітленості.

Альтернативним варіантом для передавання даних є використання MQTT-протоколу. MQTT є легким протоколом обміну повідомленнями, побудованим за моделлю «видавець-підписник», і призначений для M2M- та IoT-комунікацій в умовах обмежених ресурсів [18]. У такій архітектурі мобільна платформа може публікувати повідомлення в теми на зразок «robot/temperature», «robot/humidity», «robot/air_quality» або «robot/light», а вебінтерфейс чи сервер може підписуватися на ці теми й отримувати актуальні значення. Для Arduino-проектів може використовуватися бібліотека PubSubClient, яка забезпечує клієнтську підтримку MQTT [19].

У межах базової реалізації доцільно застосувати HTTP-передавання даних на ThingSpeak або локальний вебсервер, оскільки цей варіант є простішим для навчального проєкту. MQTT варто розглядати як перспективний варіант удосконалення, якщо система буде масштабуватися, наприклад у разі підключення кількох роботизованих платформ або стаціонарних сенсорних вузлів. HTTP-запити простіші для початкової реалізації, тоді як MQTT краще

підходить для регулярного обміну короткими повідомленнями між багатьма пристроями.

Загальну структуру програмного стеку системи доцільно подати у вигляді кількох рівнів: рівень апаратного зчитування, рівень обробки даних, рівень комунікації, рівень IoT-платформи та рівень користувацького інтерфейсу. На першому рівні Arduino Uno отримує сигнали від DHT22, BMP280, MQ-135 та LDR. На другому рівні виконується перевірка, фільтрація та форматування даних. На третьому рівні ESP8266 забезпечує Wi-Fi-передавання. На четвертому рівні IoT-платформа приймає, зберігає та візуалізує дані. На п'ятому рівні користувач переглядає результати моніторингу у вигляді графіків, таблиць або повідомлень.

Орієнтовний програмний стек мобільної платформи екологічного моніторингу подано в таблиці 1.4.

Таблиця 1.4 – Програмний стек мобільної платформи екологічного моніторингу

Рівень програмного стеку	Програмні засоби	Призначення
Середовище розробки	Arduino IDE	Написання, компіляція та завантаження коду на Arduino Uno
Мова програмування	Arduino C/C++	Реалізація алгоритмів зчитування, обробки та передавання даних
Робота із сенсорами	DHT library, BMP280 library, analogRead()	Отримання показників вологості, температури, тиску, якості повітря та освітленості
Комунікаційний рівень	UART, SoftwareSerial, ESP8266	Обмін даними між Arduino Uno та Wi-Fi-модулем
Мережевий рівень	HTTP або MQTT	Передавання даних на сервер або IoT-платформу
IoT-платформа	ThingSpeak або локальний вебсервер	Зберігання, візуалізація та аналіз показників
Користувацький рівень	Вебінтерфейс, графіки, таблиці	Перегляд результатів екологічного моніторингу

Під час розробки програмного забезпечення важливо передбачити правильну послідовність роботи системи. Після запуску Arduino Uno ініціалізує сенсори, перевіряє доступність модуля ESP8266, встановлює параметри обміну та переходить у цикл вимірювання. У кожному циклі система зчитує значення з DHT22, BMP280, MQ-135 і LDR, виконує базову перевірку даних, формує пакет

вимірювань і передає його через ESP8266. Якщо з'єднання з мережею тимчасово недоступне, програмний алгоритм має передбачати повторну спробу підключення або локальне виведення даних через Serial Monitor для діагностики.

Для підвищення надійності програмного забезпечення доцільно реалізувати фільтрацію показників. Наприклад, для аналогових сенсорів MQ-135 і LDR можна виконувати кілька послідовних зчитувань і використовувати середнє значення. Це дасть змогу зменшити вплив випадкових коливань сигналу. Для DHT22 та BMP280 потрібно перевіряти, чи не повертає датчик некоректні або порожні значення. Також бажано встановити часові інтервали між вимірюваннями, щоб уникнути надмірного навантаження на сенсори та мережевий модуль.

Обґрунтований програмний стек має відповідати кільком вимогам: простоті реалізації, сумісності з Arduino Uno, підтримці вибраних сенсорів, можливості передавання даних через ESP8266, зручності налагодження та подальшого розширення. Arduino IDE і Arduino C/C++ забезпечують доступну основу для написання коду, бібліотеки спрощують роботу із сенсорами, а Wi-Fi-модуль ESP8266 забезпечує вихід у мережу. Використання ThingSpeak або локального сервера дозволяє візуалізувати результати моніторингу без необхідності створювати складну серверну інфраструктуру з нуля.

Отже, для мобільної гусеничної платформи екологічного моніторингу доцільно використати програмний стек, що складається з Arduino IDE, мови Arduino C/C++, бібліотек для DHT22 і BMP280, стандартних функцій `analogRead()` для MQ-135 та LDR, послідовного обміну з ESP8266 і HTTP-передавання даних на IoT-платформу. Такий стек є достатнім для реалізації базового прототипу, забезпечує зчитування метеопараметрів, передавання даних через Wi-Fi та їх подальшу візуалізацію. У перспективі систему можна розширити підтримкою MQTT, локальної бази даних, веб-панелі керування або мобільного застосунку.

РОЗДІЛ 2

ПРОЕКТУВАННЯ АРХІТЕКТУРИ ТА АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

2.1 Розробка принципової схеми робота-патруля

Розробка принципової схеми робота-патруля є одним із ключових етапів практичної реалізації мобільної гусеничної платформи екологічного моніторингу. На цьому етапі визначається спосіб електричного з'єднання мікроконтролера Arduino Uno з датчиками метеопараметрів, модулем зв'язку, драйвером двигунів, джерелом живлення та виконавчими елементами. На відміну від структурної схеми, яка відображає загальну логіку роботи системи, принципова схема деталізує підключення виводів, ліній живлення, сигнальних з'єднань і спільної землі.

У запропонованій системі Arduino Uno виконує роль центрального керуючого пристрою. Ця плата має 14 цифрових входів/виходів, з яких 6 можуть працювати як PWM-виходи, а також 6 аналогових входів, що дає змогу підключити як цифрові, так і аналогові сенсори [20]. Саме тому Arduino Uno доцільно використати для зчитування показників DHT22, BMP280, MQ-135 і LDR, формування керуючих сигналів для драйвера двигунів L298N та обміну даними з Wi-Fi-модулем ESP8266.

Підключення сенсорів до Arduino Uno залежить від типу вихідного сигналу. Датчик DHT22 формує цифровий сигнал і може бути підключений до одного з цифрових виводів, наприклад D2. Сенсор BMP280 працює через цифровий інтерфейс I2C або SPI, тому в базовій реалізації його доцільно підключити через I2C: лінію SDA до A4, а лінію SCL до A5 на Arduino Uno. Датчик MQ-135 зазвичай має аналоговий вихід, тому його доцільно підключити до аналогового входу A0. LDR-фоторезистор підключається через подільник напруги, а його аналоговий сигнал може надходити на вхід A1. Таке розміщення виводів дає змогу логічно розподілити сенсорні сигнали та залишити цифрові виводи для керування двигунами й модулем зв'язку.

Для керування гусеничною платформою доцільно використати драйвер двигунів L298N або аналогічний модуль на базі подвійного H-моста. Мікроконтролер Arduino Uno не може безпосередньо жити двигуни, оскільки його виводи призначені лише для формування логічних сигналів. Драйвер L298 призначений для керування індуктивними навантаженнями, зокрема двигунами постійного струму, і підтримує роботу з двома каналами керування [21]. У схемі робота-патруля один канал драйвера може керувати лівою гусеницею, а другий - правою. Завдяки цьому робот може рухатися вперед, назад, повертати ліворуч, праворуч і зупинятися.

Для керування L298N можна використати цифрові виводи Arduino Uno. Наприклад, ENA підключається до PWM-виводу D5, IN1 – до D7, IN2 – до D8, IN3 – до D9, IN4 – до D10, а ENB – до PWM-виводу D6. Виводи ENA та ENB відповідають за ввімкнення каналів і регулювання швидкості двигунів за допомогою PWM-сигналу. Виводи IN1-IN4 визначають напрям обертання двигунів. Якщо IN1 та IN2 мають різні логічні стани, двигун обертається в одному напрямі або в іншому. Якщо обидва входи перебувають в однаковому стані, двигун зупиняється або переходить у режим гальмування залежно від логіки керування драйвером.

Окрему увагу потрібно приділити підключенню ESP8266, оскільки цей модуль працює з логічними рівнями 3,3 В, тоді як Arduino Uno використовує логіку 5 В. ESP8266 має UART-інтерфейси для передавання даних, тому може взаємодіяти з Arduino через послідовний порт [22]. Проте пряме підключення TX Arduino до RX ESP8266 без узгодження рівнів напруги є небажаним. Для безпечної роботи потрібно використати логічний перетворювач рівнів або подільник напруги для лінії, що надходить до ESP8266. Також ESP8266 потребує стабільного живлення 3,3 В, тому його не варто живити від слабкого виводу 3,3 В Arduino Uno, якщо модуль споживає значний струм під час Wi-Fi-передавання.

Система живлення робота-патруля повинна бути розділена на дві основні частини: живлення логічних компонентів і живлення двигунів. Логічна частина

охоплює Arduino Uno, сенсори, ESP8266 та інші низькопотужні модулі. Виконавча частина охоплює двигуни постійного струму та драйвер L298N. Для двигунів доцільно використати окремий акумуляторний блок, наприклад 7,4 В або 12 В залежно від характеристик моторів. Для Arduino та сенсорів потрібно забезпечити стабільні 5 В, а для ESP8266 – стабільні 3,3 В. Усі блоки живлення повинні мати спільну землю, оскільки без спільного GND сигнальні рівні між Arduino, драйвером двигунів і сенсорами можуть визначатися некоректно.

Принципову схему робота-патруля доцільно подати на рисунку 2.1.

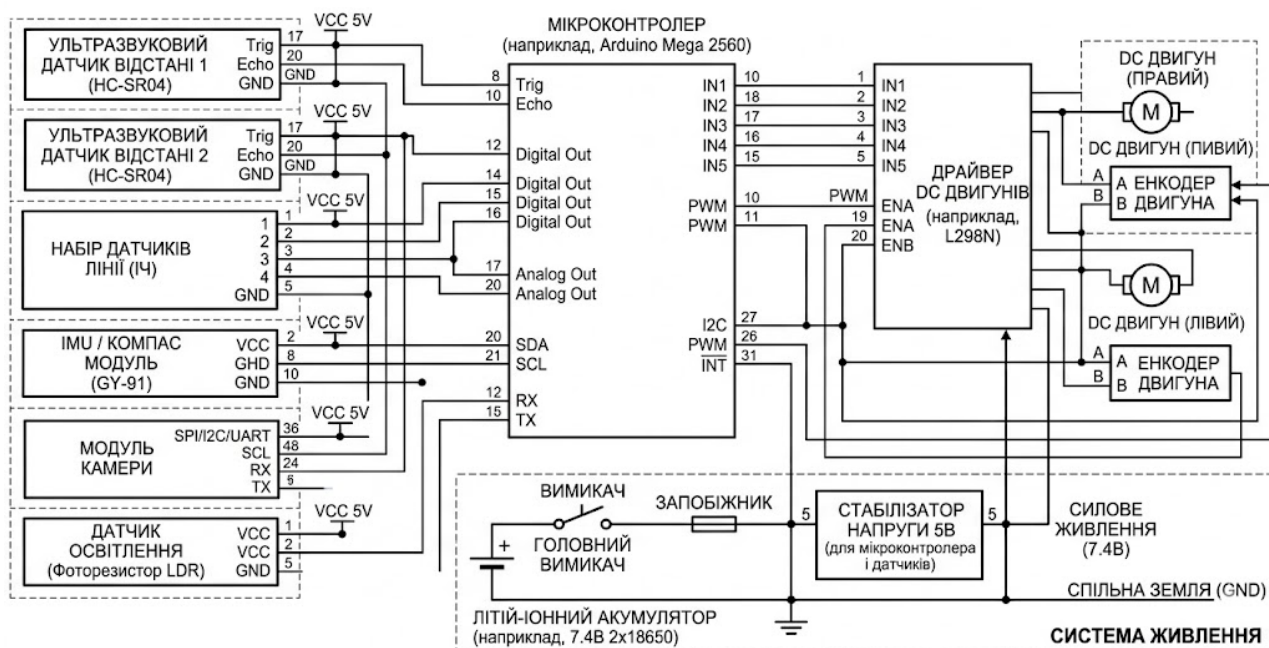


Рисунок 2.1 – Принципова схема підключення датчиків, драйвера двигунів і системи живлення робота-патруля

Орієнтовне призначення виводів Arduino Uno у принциповій схемі подано в таблиці 2.1.

Таблиця 2.1 – Орієнтовне підключення компонентів робота-патруля до Arduino Uno

Компонент	Вивід компонента	Вивід Arduino Uno	Призначення
DHT22	DATA	D2	Зчитування температури та вологості
BMP280	SDA	A4	Передавання даних через I2C
BMP280	SCL	A5	Тактова лінія I2C

Продовження таблиці 2.1

Компонент	Вивід компонента	Вивід Arduino Uno	Призначення
MQ-135	АО	A0	Аналоговий сигнал якості повітря
LDR	Вихід подільника напруги	A1	Аналогове визначення освітленості
L298N	ENA	D5	PWM-керування швидкістю лівого двигуна
L298N	ENB	D6	PWM-керування швидкістю правого двигуна
L298N	IN1	D7	Напрямок обертання лівого двигуна
L298N	IN2	D8	Напрямок обертання лівого двигуна
L298N	IN3	D9	Напрямок обертання правого двигуна
L298N	IN4	D10	Напрямок обертання правого двигуна
ESP8266	TX / RX	D3 / D4 або окремий UART	Обмін даними між Arduino та Wi-Fi-модулем
Живлення	GND	GND	Спільна земля всієї системи

Під час розробки принципової схеми важливо враховувати електричну сумісність компонентів. DHT22 може працювати з живленням 3,3-5,5 В, тому його зручно підключати до Arduino Uno [6]. BMP280 має підтримку цифрових інтерфейсів I2C та SPI, що дозволяє інтегрувати його з мікроконтролером без використання аналогових входів [23]. MQ-135 і LDR підключаються до аналогових входів, тому для них доцільно передбачити стабільне живлення та програмну фільтрацію показників. Для MQ-135 додатково потрібно враховувати час прогріву та можливу залежність показників від температури і вологості.

У схемі керування двигунами важливо не підключати мотори безпосередньо до Arduino Uno. Двигуни створюють значне струмове навантаження та електричні завади, які можуть спричинити перезавантаження мікроконтролера або некоректну роботу сенсорів. Тому двигуни мають живитися через драйвер L298N від окремої силової лінії. Для підвищення стабільності роботи системи доцільно використовувати конденсатори фільтрації живлення, стабілізатор напруги для логічної частини та якісне з'єднання GND між усіма модулями.

З погляду логіки роботи принципова схема забезпечує послідовний обмін даними між усіма елементами системи. Сенсори передають показники на Arduino Uno, мікроконтролер обробляє ці дані та формує керуючі сигнали. Якщо система виявляє зміну параметрів середовища або перешкоду, Arduino може змінити режим руху, зменшити швидкість, зупинити платформу або передати повідомлення через ESP8266. Драйвер L298N виконує команди Arduino та керує двигунами гусеничної платформи. Блок живлення забезпечує автономність роботи системи, а Wi-Fi-модуль реалізує передавання результатів моніторингу до користувача або IoT-платформи.

Принципова схема робота-патруля повинна забезпечувати надійне з'єднання сенсорного блока, блока керування, драйвера двигунів, комунікаційного модуля та системи живлення. Для базової реалізації доцільно використати Arduino Uno як центральний контролер, DHT22, BMP280, MQ-135 і LDR як сенсори екологічного моніторингу, L298N як драйвер двигунів, ESP8266 як Wi-Fi-модуль та акумуляторний блок із стабілізаторами напруги. Така схема є достатньою для створення навчального прототипу гусеничного робота-патруля, здатного збирати метеодані, контролювати якість повітря, визначати освітленість, керувати рухом і передавати інформацію через IoT-канал.

2.2 Алгоритм керування рухом та навігацією

Алгоритм керування рухом та навігацією є основою функціонування мобільного робота-патруля, оскільки саме він визначає послідовність дій платформи під час переміщення, виявлення перешкод, вибору безпечного напрямку руху та виконання вимірювань параметрів середовища. Для гусеничної мобільної платформи екологічного моніторингу алгоритм повинен поєднувати дві основні групи завдань: забезпечення автономного руху та організацію збору даних із сенсорів у визначених точках маршруту.

У запропонованій системі керування рухом реалізується на базі Arduino Uno, драйвера двигунів L298N, двох двигунів гусеничної платформи та

ультразвукового датчика відстані. Arduino Uno формує керуючі сигнали для драйвера L298N, який змінює напрям обертання лівого й правого двигунів. Завдяки цьому робот може рухатися вперед, назад, виконувати поворот ліворуч або праворуч, а також зупинитися [20, 21]. Ультразвуковий датчик використовується як основний елемент виявлення перешкод перед роботом.

Принцип роботи ультразвукового датчика полягає у випромінюванні короткого ультразвукового імпульсу та вимірюванні часу, за який відбитий сигнал повертається до приймача. На основі цього часу мікроконтролер обчислює відстань до об'єкта. У типовому модулі HC-SR04 використовується два сигнальні виводи: Trig для запуску імпульсу та Echo для приймання відбитого сигналу. Згідно з технічним описом, датчик може вимірювати відстань у діапазоні приблизно від 2 см до 400 см, що є достатнім для базової системи виявлення перешкод у мобільному роботі [24].

Для реалізації автономного обходу перешкод у програмному алгоритмі задаються порогові значення відстані. Наприклад, якщо відстань до об'єкта перевищує 30 см, робот продовжує рух уперед. Якщо відстань зменшується до 20-30 см, система може зменшити швидкість руху та виконати повторне вимірювання. Якщо відстань менша ніж 20 см, робот повинен зупинитися, виконати аналіз доступних напрямків і обрати безпечний варіант подальшого руху. Такі порогові значення можуть змінюватися залежно від розмірів робота, швидкості руху, типу поверхні та точності датчика.

Логіка обходу перешкод може бути реалізована за простим реактивним принципом. Спочатку робот рухається вперед і періодично вимірює відстань до об'єкта перед собою. Якщо перешкоди немає, рух продовжується. Якщо датчик фіксує перешкоду на небезпечній відстані, робот зупиняється, виконує короткий рух назад або залишається на місці, після чого повертається ліворуч або праворуч. У простому варіанті напрям повороту може задаватися програмно, наприклад спочатку робот пробує повернути праворуч. У більш досконалому варіанті можна використовувати сервопривід для повороту ультразвукового

датчика ліворуч і праворуч, щоб порівняти відстані в обох напрямках і вибрати той, де простір є більш вільним.

Алгоритм руху робота-патруля можна подати як послідовність таких дій: ініціалізація мікроконтролера, сенсорів і драйвера двигунів, запуск основного циклу, вимірювання відстані до перешкоди, прийняття рішення щодо руху, керування двигунами, фіксація точки заміру, зчитування метеопараметрів, передавання даних через модуль зв'язку, повторення циклу. Така структура дає змогу поєднати навігацію та екологічний моніторинг в одному програмному процесі.

Загальну логіку алгоритму керування рухом і навігацією робота-патруля доцільно подати на рисунку 2.2.

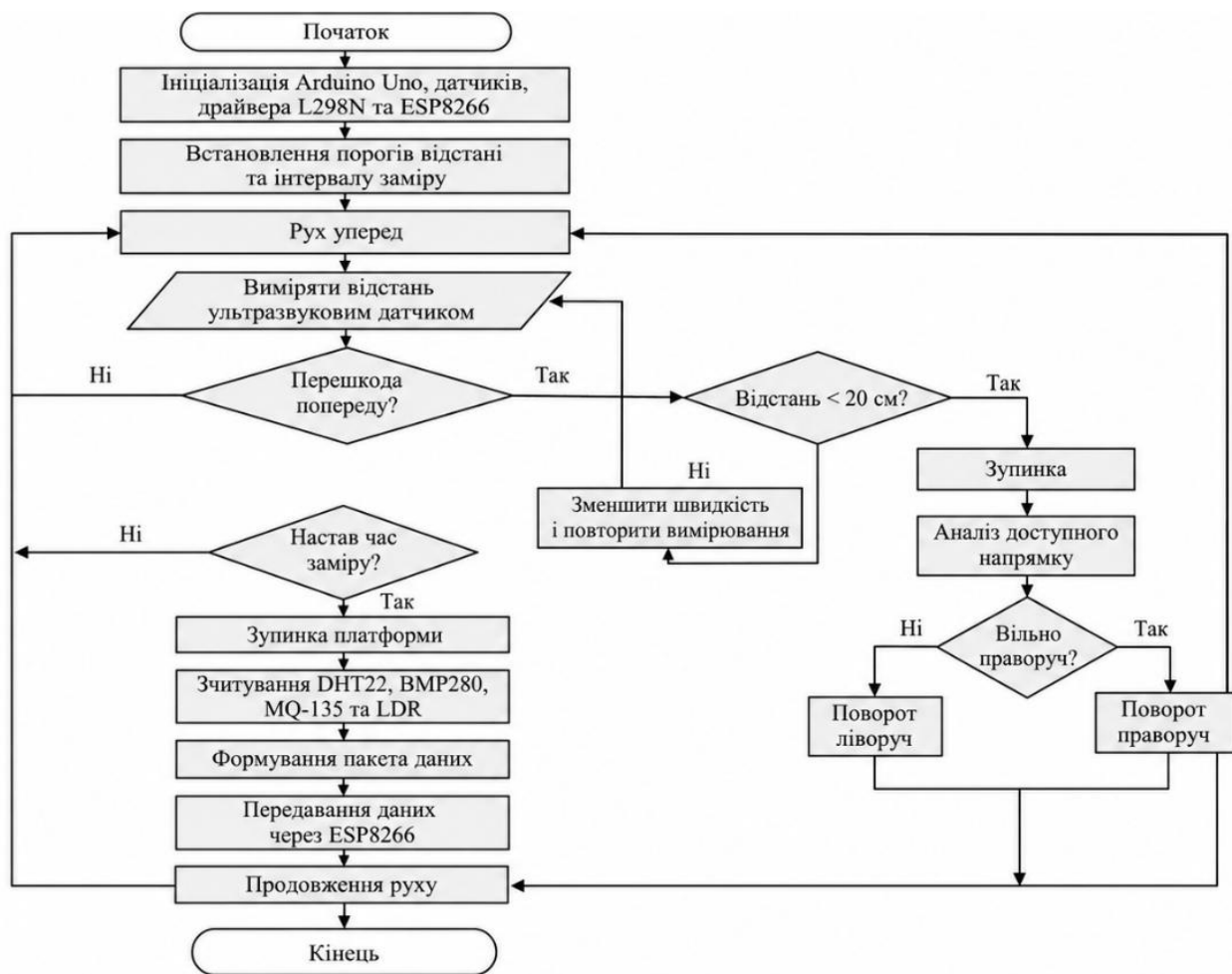


Рисунок 2.2 – Блок-схема алгоритму керування рухом і навігацією робота-патруля

Важливим завданням є вибір точок заміру параметрів середовища. Оскільки робот-патруль є мобільною платформою, він може виконувати вимірювання не лише в одному місці, а в різних ділянках контрольованої території. Це підвищує інформативність екологічного моніторингу, оскільки дає змогу порівнювати температуру, вологість, атмосферний тиск, якість повітря та освітленість у різних точках маршруту.

У межах навчального або прототипного проекту можна використати кілька підходів до вибору точок заміру. Перший підхід – часовий, коли робот виконує вимірювання через однакові проміжки часу, наприклад кожні 10 або 30 секунд. Другий підхід – маршрутний, коли заміри здійснюються після проходження певної ділянки або після зміни напрямку руху. Третій підхід – подієвий, коли система виконує додатковий замір у разі виявлення перешкоди, різкої зміни показників сенсорів або зупинки робота. Для базової реалізації найбільш доцільним є комбінований підхід, за якого основні вимірювання виконуються через заданий часовий інтервал, а додаткові – у разі виникнення події.

Під час вибору точки заміру важливо забезпечити стабільність показників. Якщо робот рухається по нерівній поверхні, вібрації можуть впливати на якість вимірювань, особливо для сенсорів, які реагують на повітряні потоки або потребують стабільного положення. Тому доцільно виконувати зчитування метеопараметрів після короткої зупинки платформи. Наприклад, робот зупиняється на 2-3 секунди, зчитує значення DHT22, BMP280, MQ-135 та LDR, формує пакет даних і передає його через ESP8266. Після цього рух продовжується.

Орієнтовну логіку режимів руху та вимірювання подано в таблиці 2.2.

Таблиця 2.2 – Режими роботи робота-патруля під час руху та екологічного моніторингу

Умова роботи	Дія системи	Призначення
Відстань до перешкоди більша за 30 см	Рух уперед	Патрулювання території без зміни маршруту
Відстань 20-30 см	Зменшення швидкості, повторне вимірювання	Уточнення наявності перешкоди

Продовження таблиці 2.2

Умова роботи	Дія системи	Призначення
Відстань менша за 20 см	Зупинка, вибір нового напрямку	Запобігання зіткненню
Вільний простір праворуч	Поворот праворуч і продовження руху	Обхід перешкоди
Вільний простір ліворуч	Поворот ліворуч і продовження руху	Обхід перешкоди
Досягнуто інтервалу заміру	Зупинка, зчитування сенсорів	Отримання стабільних даних моніторингу
Виявлено різку зміну параметра	Додатковий замір і передавання повідомлення	Фіксація потенційно важливої події
Немає зв'язку з IoT-модулем	Локальне продовження руху або зупинка	Захист від втрати даних і аварійної ситуації

Для програмної реалізації алгоритму доцільно використовувати циклічну структуру, у якій кожна ітерація містить блок навігації та блок моніторингу. Блок навігації відповідає за вимірювання відстані, порівняння її з пороговими значеннями та керування двигунами. Блок моніторингу відповідає за визначення моменту заміру, зчитування сенсорів і формування пакета даних. Такий підхід дозволяє уникнути розділення системи на дві незалежні частини та забезпечує узгоджену роботу руху й екологічного контролю.

Перевагою алгоритму є його простота та придатність для реалізації на Arduino Uno. Він не потребує складних обчислень, картографування або використання комп'ютерного зору, однак забезпечує базову автономність роботи. Система здатна рухатися контрольованою територією, реагувати на перешкоди, уникати зіткнень і виконувати вимірювання параметрів середовища. Для навчального проєкту цього достатньо, щоб продемонструвати принцип роботи мобільної IoT-платформи екологічного моніторингу.

Разом із тим алгоритм має потенціал для подальшого вдосконалення. Наприклад, до системи можна додати енкодери для точнішого визначення пройденої відстані, GPS-модуль для фіксації координат точок заміру, сервопривід для повороту ультразвукового датчика, карту контрольованої території або алгоритм руху за контрольними точками. Також можна реалізувати передавання

не лише поточних значень сенсорів, а й журналу маршруту, часу заміру та стану акумулятора.

Алгоритм керування рухом та навігацією робота-патруля повинен забезпечувати безпечне переміщення гусеничної платформи, автономний обхід перешкод і виконання вимірювань параметрів середовища у визначених точках. У базовій реалізації доцільно використати ультразвуковий датчик для виявлення перешкод, драйвер L298N для керування двигунами та часово-подієвий підхід до вибору точок заміру. Такий алгоритм є простим, зрозумілим і придатним для практичної реалізації на Arduino Uno, а також може бути розширений у майбутньому шляхом додавання складніших навігаційних і моніторингових функцій.

2.3 Розробка модуля збору та обробки даних

Модуль збору та обробки даних є однією з основних складових мобільного робота-патруля для екологічного моніторингу, оскільки саме він забезпечує отримання показників із сенсорів, їх первинну перевірку, фільтрацію шумів, підготовку до передавання через IoT-модуль та подальше використання в системі аналізу. У запропонованій системі цей модуль реалізується на базі Arduino Uno, до якого підключаються цифрові та аналогові датчики: DHT22, BMP280, MQ-135, LDR, а також, за потреби, ультразвуковий датчик для навігації.

З технічного погляду модуль збору даних виконує кілька послідовних операцій. Спочатку мікроконтролер ініціалізує всі підключені сенсори та перевіряє їх доступність. Після цього виконується зчитування показників, перетворення сигналів у цифрові значення, фільтрація випадкових коливань, формування пакета даних і передавання результатів до комунікаційного модуля ESP8266. Така послідовність дозволяє уникнути хаотичного опитування сенсорів і забезпечує стабільну роботу системи в режимі періодичного моніторингу.

У системі використовуються два основні типи датчиків: цифрові та аналогові. До цифрових належать DHT22 та BMP280. Датчик DHT22 передає вже

сформовані цифрові значення температури й вологості, тому для його зчитування доцільно застосовувати відповідну бібліотеку Arduino. Офіційна документація Arduino для DHT sensor library зазначає, що бібліотека сумісна з усіма архітектурами Arduino, що спрощує її використання у навчальних і прототипних проєктах [25].

Датчик BMP280 є цифровим барометричним сенсором, який може працювати через інтерфейси I2C або SPI. У межах цієї роботи доцільно використовувати інтерфейс I2C, оскільки він потребує лише двох сигнальних ліній – SDA та SCL, що дає змогу економити виводи Arduino Uno. У технічній документації Bosch зазначено, що BMP280 підтримує I2C та SPI, має режими фільтрації та використовується для точного вимірювання атмосферного тиску [26].

До аналогових датчиків у системі належать MQ-135 та LDR. MQ-135 використовується для оцінювання якості повітря, а LDR - для визначення рівня освітленості. Їхні вихідні сигнали змінюються у вигляді напруги, яка подається на аналогові входи Arduino Uno. Для зчитування таких сигналів використовується функція `analogRead()`. В офіційній документації Arduino зазначено, що Arduino Uno має 10-бітний аналого-цифровий перетворювач, який перетворює вхідну напругу в діапазон значень від 0 до 1023 [27].

Для цифрових датчиків логіка зчитування відрізняється від аналогових. Якщо датчик передає логічний стан, використовується функція `digitalRead()`, яка зчитує значення HIGH або LOW з обраного цифрового виводу [28]. У випадку DHT22 та BMP280 частіше застосовуються спеціалізовані бібліотеки, оскільки ці сенсори передають не просто логічний рівень, а структуровані дані, які потрібно коректно прийняти, розпізнати та перетворити у фізичні величини.

Загальна логіка роботи модуля збору даних передбачає періодичне опитування сенсорів. Наприклад, один цикл вимірювання може виконуватися кожні 5-10 секунд. У цьому циклі Arduino Uno зчитує температуру та вологість із DHT22, атмосферний тиск із BMP280, аналогове значення MQ-135, аналогове

значення LDR, а також, за потреби, відстань до перешкоди з ультразвукового датчика. Після цього отримані дані проходять первинну перевірку та фільтрацію.

У процесі зчитування даних із сенсорів важливо враховувати можливість виникнення шумів і похибок. Для аналогових датчиків шум може виникати через нестабільність живлення, електромагнітні завади від двигунів, коливання контактів, зміну температури або особливості самого сенсора. Особливо це стосується MQ-135 та LDR, оскільки їхні сигнали залежать від зовнішніх умов і можуть змінюватися навіть при незначних коливаннях напруги. Тому пряме використання одного вимірювання без обробки може призвести до нестабільних результатів.

Найпростішим методом фільтрації є усереднення кількох послідовних вимірювань. Наприклад, для MQ-135 і LDR можна виконати 10 зчитувань з короткою затримкою між ними, після чого обчислити середнє значення. Такий метод зменшує вплив випадкових коливань і робить показники стабільнішими. Для навчального прототипу цей підхід є найбільш доцільним, оскільки він простий у реалізації та не потребує значних обчислювальних ресурсів.

Орієнтовну структуру модуля збору та обробки даних доцільно подати на рисунку 2.3.

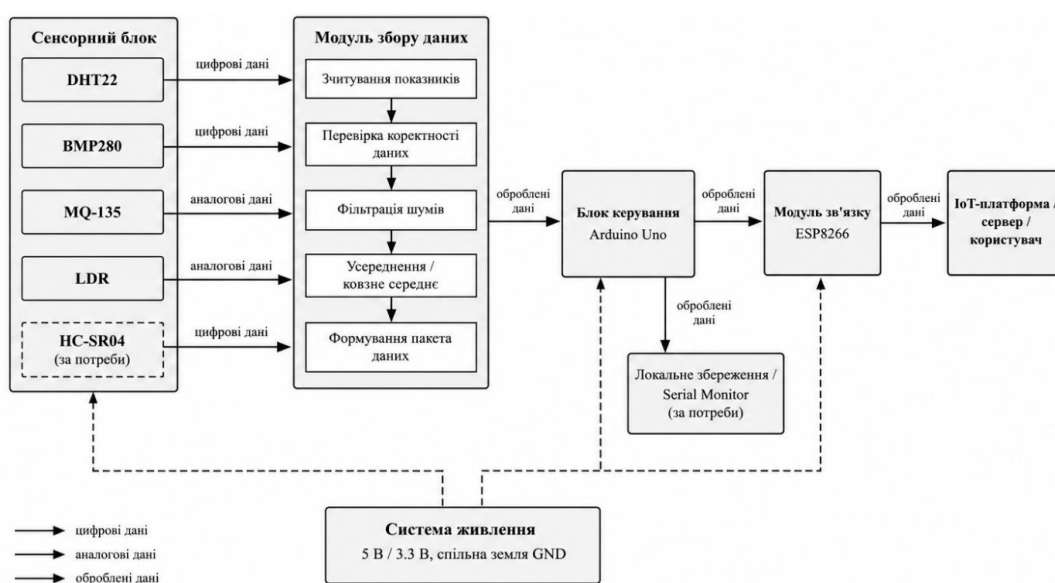


Рисунок 2.3 – Структура модуля збору та обробки даних мобільного робота-патруля

Найпростішим методом фільтрації є усереднення кількох послідовних вимірювань. Наприклад, для MQ-135 і LDR можна виконати 10 зчитувань з короткою затримкою між ними, після чого обчислити середнє значення. Такий метод зменшує вплив випадкових коливань і робить показники стабільнішими. Для навчального прототипу цей підхід є найбільш доцільним, оскільки він простий у реалізації та не потребує значних обчислювальних ресурсів.

Іншим методом є ковзне середнє. У цьому випадку система зберігає кілька останніх значень сенсора та на кожному циклі обчислює середнє значення з цього набору. Наприклад, якщо використовується вікно з 5 вимірювань, нове значення додається до масиву, найстаріше видаляється, а результатом вважається середнє з останніх п'яти показників. Такий підхід корисний для плавного відображення даних на графіку або веб-панелі.

Для виявлення різких змін можна використовувати порогову фільтрацію. Наприклад, якщо нове значення MQ-135 суттєво відрізняється від попереднього, система не повинна одразу вважати це реальною зміною якості повітря. Доцільно виконати повторне вимірювання і лише після підтвердження зміни сформулювати повідомлення. Така логіка зменшує ймовірність помилкових спрацювань через короткочасні завади.

Основні методи зчитування та обробки даних подано в таблиці 2.3.

Таблиця 2.3 – Методи зчитування та обробки показників сенсорів

Сенсор	Тип сигналу	Метод зчитування	Потреба у фільтрації	Особливості обробки
DHT22	Цифровий	Через бібліотеку DHT	Низька	Перевірка коректності температури та вологості
BMP280	Цифровий I2C / SPI	Через бібліотеку BMP280	Низька / середня	Перевірка значень тиску, можливе усереднення
MQ-135	Аналоговий	analogRead()	Висока	Усереднення, порогова обробка, калібрування
LDR	Аналоговий	analogRead() через подільник напруги	Середня	Усереднення, нормалізація рівня освітленості
HC-SR04	Цифровий імпульсний	Trig / Echo	Середня	Повторне вимірювання при підозрілих значеннях

Модуль збору та обробки даних забезпечує перехід від окремих сенсорних сигналів до структурованої інформації, придатної для екологічного моніторингу. Для цифрових сенсорів DHT22 і BMP280 доцільно використовувати готові бібліотеки, а для аналогових MQ-135 і LDR - функцію `analogRead()` з подальшим усередненням і пороговою обробкою. Застосування фільтрації шумів підвищує стабільність показників і зменшує ймовірність помилкових спрацювань. У результаті сформований пакет даних може передаватися через ESP8266 до IoT-платформи, де користувач отримує доступ до актуальних параметрів середовища.

РОЗДІЛ 3

ПРАКТИЧНА РЕАЛІЗАЦІЯ ПРОТОТИПУ МОБІЛЬНОГО РОБОТА ДЛЯ МОНІТОРИНГУ

3.1 Збірка фізичної моделі робота на базі гусеничної платформи

Збірка фізичної моделі робота на базі гусеничної платформи є першим практичним етапом технічної реалізації прототипу мобільної системи екологічного моніторингу. На цьому етапі формується механічна основа пристрою, виконується розміщення електронних модулів, підключення виконавчих механізмів, монтаж сенсорного блока та підготовка платформи до подальшого програмного налагодження. Особливістю обраної конструкції є використання гусеничного шасі, яке забезпечує підвищену прохідність, стійкість під час руху та можливість переміщення не лише по рівній поверхні, а й по ділянках із незначними нерівностями.

Фізична модель робота побудована на базі компактної гусеничної платформи, яка містить дві незалежні гусениці, електродвигуни постійного струму, несучу раму, верхню монтажну панель і місця для встановлення електронних компонентів. Така конструкція є зручною для створення мобільного робота-патруля, оскільки дозволяє реалізувати диференційне керування рухом: якщо обидві гусениці обертаються в одному напрямку, робот рухається вперед або назад, а якщо гусениці обертаються з різною швидкістю або в протилежних напрямках, платформа виконує поворот.

На рисунку 3.1 показано зібрану гусеничну роботизовану платформу, на якій розміщено основні апаратні компоненти системи. У передній частині конструкції встановлено ультразвуковий датчик, який виконує функцію виявлення перешкод під час руху. Завдяки фронтальному розташуванню датчик може контролювати простір перед роботом і передавати інформацію до мікроконтролера для прийняття рішення щодо зупинки або зміни напрямку руху. Верхня частина платформи використовується для монтажу плати керування, комунікаційного модуля, проводів з'єднання та сенсорного блока.

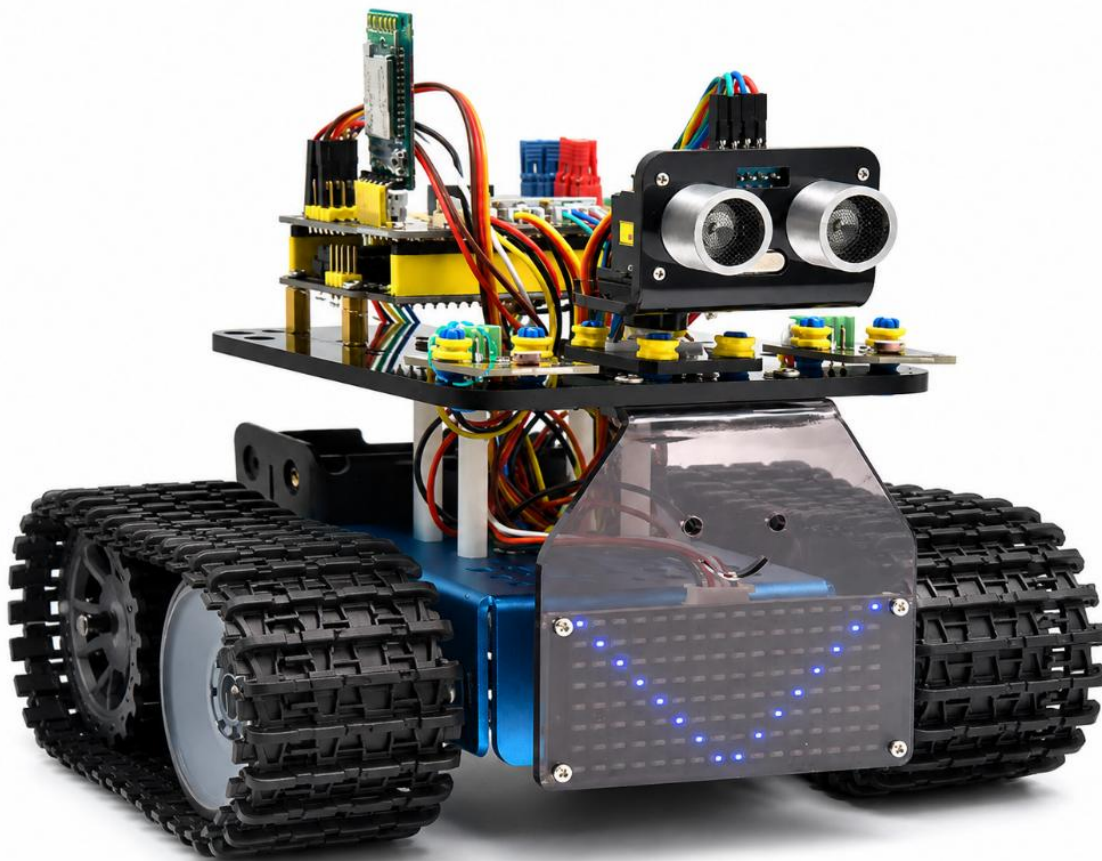


Рисунок 3.1 – Загальний вигляд фізичної моделі робота на базі гусеничної платформи

Основними елементами фізичної моделі є гусеничне шасі, плата Arduino Uno, драйвер двигунів, ультразвуковий датчик, сенсорний модуль екологічного моніторингу, модуль зв'язку, акумуляторний блок і монтажні елементи. Гусеничне шасі виконує роль механічної основи робота, на якій закріплюються всі інші вузли. Arduino Uno використовується як центральний контролер, що обробляє сигнали від сенсорів і формує команди для виконавчих пристроїв. Драйвер двигунів забезпечує керування електродвигунами гусеничної платформи, оскільки вихідні виводи Arduino не розраховані на безпосереднє живлення моторів.

Компонування основних вузлів робота доцільно подати на рисунку 3.2.

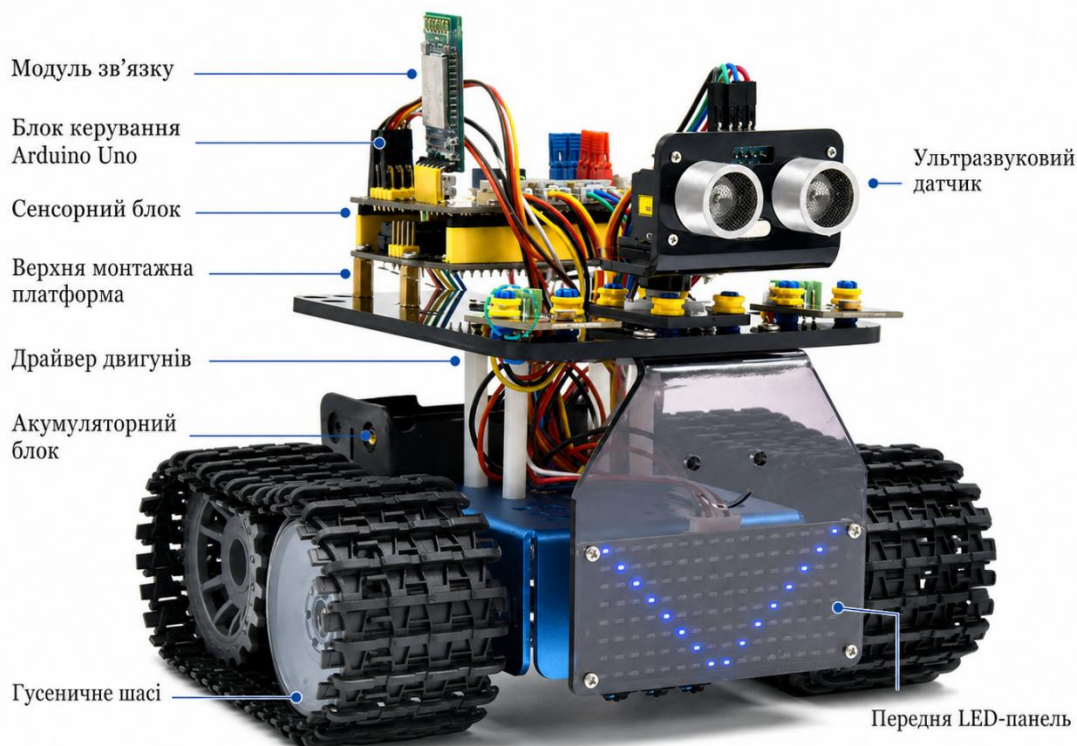


Рисунок 3.2 – Компонування основних вузлів гусеничного робота-патруля

У процесі складання насамперед формується механічна частина платформи. До основи шасі встановлюються двигуни постійного струму з редукторами, ведучі колеса, опорні елементи та гусеничні стрічки. Після цього перевіряється легкість обертання гусениць, відсутність перекосів і достатній натяг гусеничних стрічок. Надмірно слабке натягнення може призвести до ковзання гусениці під час повороту, а надмірно сильне – до збільшення навантаження на двигуни та швидшого розрядження акумулятора. Тому на етапі механічної збірки важливо забезпечити рівномірне розміщення гусениць і стабільний контакт із поверхнею.

Після складання механічної основи виконується монтаж верхньої платформи. Вона використовується для розміщення електронних компонентів і повинна забезпечувати достатню жорсткість конструкції. На верхній панелі доцільно розміщувати Arduino Uno, плату розширення або макетну плату, модуль ESP8266, сенсори метеостанції та з'єднувальні дроти. Важливо, щоб електронні

компоненти не контактували з рухомими частинами гусениць, не зазнавали механічного тиску та були захищені від випадкового короткого замикання.

Окрему увагу під час складання потрібно приділити розміщенню ультразвукового датчика. Його доцільно встановлювати у передній частині робота на невеликій висоті над поверхнею. Таке розташування дозволяє виявляти перешкоди перед платформою та своєчасно передавати сигнал до контролера. Датчик має бути спрямований горизонтально вперед, оскільки нахил униз може призвести до помилкового виявлення поверхні підлоги, а нахил угору – до пропуску низьких перешкод. У разі потреби датчик можна встановити на невеликий кронштейн або сервопривід, що дасть змогу сканувати простір ліворуч і праворуч.

Розміщення ультразвукового датчика в передній частині робота подано на рисунку 3.3.

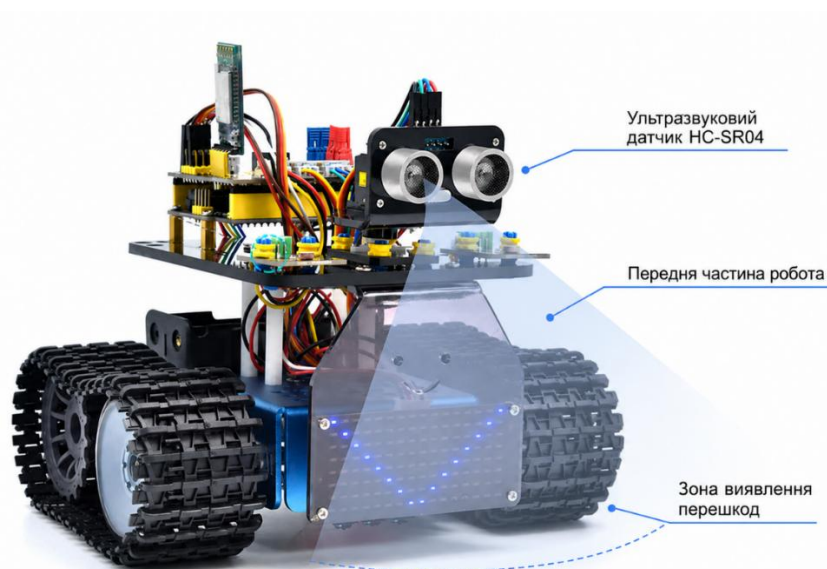


Рисунок 3.3 – Розміщення ультразвукового датчика для виявлення перешкод

Сенсорний модуль екологічного моніторингу, або умовна «метеостанція», монтується у верхній частині конструкції. Таке розміщення є доцільним, оскільки датчики температури, вологості, атмосферного тиску, якості повітря та освітленості повинні мати доступ до навколишнього середовища. Якщо розмістити сенсори занадто близько до двигунів, акумулятора або силових

модулів, їхні показники можуть спотворюватися через нагрівання електроніки, електромагнітні завади або обмежену циркуляцію повітря. Тому DHT22 і BMP280 бажано розміщувати у верхній частині робота, MQ-135 – збоку або спереду для вільного доступу повітря, а LDR – у відкритій верхній або передній зоні, де датчик може реагувати на зміну освітленості.

Схематичне розміщення сенсорного блока «метеостанції» на гусеничній платформі подано на рисунку 3.4.

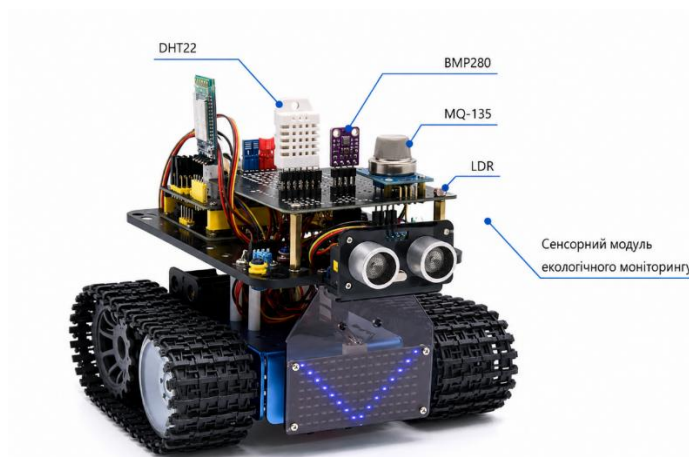


Рисунок 3.4 – Монтаж сенсорного модуля екологічного моніторингу на платформі робота

Під час монтажу сенсорного блока потрібно враховувати не лише зручність підключення, а й умови майбутніх експериментальних досліджень. Датчик DHT22 повинен розташовуватися так, щоб на нього не потрапляло тепло від плати керування або стабілізаторів напруги. Датчик BMP280 бажано розміщувати у стабільному положенні, оскільки він використовується для вимірювання атмосферного тиску та орієнтовної висоти. MQ-135 потребує доступу повітря до чутливого елемента, тому його не варто закривати корпусом або встановлювати в місці з обмеженою вентиляцією. LDR-фоторезистор повинен бути відкритим для світла, інакше показники освітленості будуть заниженими.

Під час компонування електроніки важливо дотримуватися принципу розділення силової та сигнальної частин. Двигуни та драйвер L298N створюють

більші струмові навантаження, тому їх бажано розміщувати окремо від чутливих сенсорів. Сигнальні дроти від датчиків варто прокласти так, щоб вони не перетиналися безпосередньо з силовими лініями живлення двигунів. Це зменшує ймовірність появи шумів у показниках аналогових датчиків, зокрема MQ-135 та LDR. Також важливо забезпечити спільну землю GND для всіх компонентів, оскільки без цього обмін сигналами між Arduino, драйвером двигунів і сенсорами може бути нестабільним.

Послідовність складання фізичної моделі робота можна подати у вигляді таких етапів. Спочатку виконується складання гусеничного шасі та перевірка механічної частини. Далі встановлюється монтажна панель, на яку кріпляться плата Arduino Uno, драйвер двигунів і допоміжні модулі. Після цього виконується підключення двигунів до драйвера, монтаж ультразвукового датчика та встановлення сенсорів метеостанції. Наступним етапом є підключення системи живлення, перевірка стабілізованих ліній 5 В і 3,3 В, а також контроль правильності полярності живлення. Завершальним етапом є перевірка цілісності з'єднань, тестове ввімкнення та підготовка системи до програмного налагодження.

Основні етапи складання фізичної моделі робота подано на рисунку 3.5.

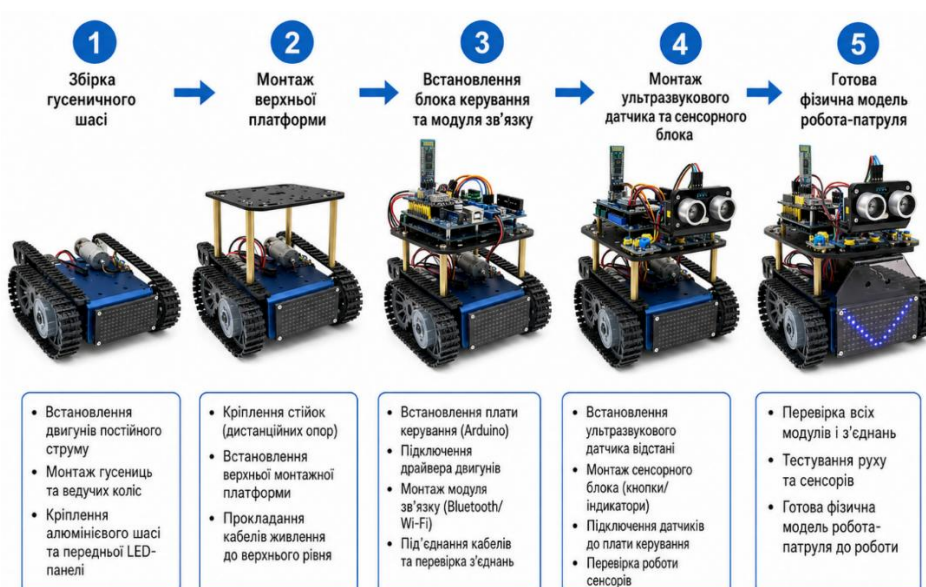


Рисунок 3.5 – Послідовність складання фізичної моделі гусеничного робота-патруля

Узагальнений склад вузлів фізичної моделі робота подано в таблиці 3.1.

Таблиця 3.1 – Основні вузли фізичної моделі гусеничного робота-патруля

Вузол	Основні компоненти	Призначення
Механічна платформа	Гусеничне шасі, рама, опорні елементи, ведучі колеса	Забезпечення руху та розміщення компонентів
Блок керування	Arduino Uno	Обробка даних сенсорів і формування керуючих сигналів
Виконавчий блок	Драйвер L298N, двигуни постійного струму	Керування рухом лівої та правої гусениці
Навігаційний блок	Ультразвуковий датчик	Виявлення перешкод перед роботом
Сенсорний блок	DHT22, BMP280, MQ-135, LDR	Збір параметрів температури, вологості, тиску, якості повітря та освітленості
Комунікаційний блок	ESP8266 або Bluetooth-модуль	Передавання даних користувачу або на IoT-платформу
Блок живлення	Акумулятор, стабілізатори 5 В і 3,3 В	Автономне живлення логічної та силової частин

Після завершення механічної збірки необхідно виконати первинну перевірку прототипу. На цьому етапі перевіряється, чи надійно закріплені всі вузли, чи не торкаються дроти рухомих частин, чи правильно встановлені гусениці та чи немає перекосу платформи. Далі перевіряється живлення логічної частини без запуску двигунів. Це дозволяє переконатися, що Arduino Uno, сенсори та комунікаційний модуль отримують стабільну напругу. Лише після цього доцільно перевіряти роботу двигунів через драйвер, оскільки помилки в силовій частині можуть призвести до перезавантаження контролера або пошкодження окремих компонентів.

Важливим результатом етапу складання є отримання фізичного прототипу, придатного до подальшого програмного налагодження. На цьому етапі робот ще не виконує повністю автономні функції, але вже має завершену конструктивну основу: гусеничне шасі, блок керування, блок руху, навігаційний датчик, сенсорний модуль екологічного моніторингу та систему живлення. Надалі саме ця фізична модель використовується для завантаження вбудованого програмного забезпечення, перевірки алгоритму руху, тестування зчитування даних із сенсорів і передавання показників через IoT-канал.

Збірка фізичної моделі робота на базі гусеничної платформи передбачає поетапне формування механічної, електронної та сенсорної частин пристрою. Обрана конструкція забезпечує достатню стійкість, прохідність і зручність для розміщення апаратних компонентів. Гусеничне шасі виконує роль основи руху, Arduino Uno забезпечує керування, драйвер двигунів реалізує переміщення платформи, ультразвуковий датчик відповідає за виявлення перешкод, а сенсорний модуль «метеостанції» дає змогу збирати параметри навколишнього середовища. Така фізична модель є придатною для подальшого експериментального дослідження роботи мобільного робота екологічного моніторингу.

3.2 Монтаж сенсорного модуля метеомоніторингу та підключення апаратних компонентів

Монтаж сенсорного модуля метеомоніторингу є важливим етапом технічної реалізації гусеничного робота-патруля, оскільки саме цей модуль забезпечує збирання даних про параметри навколишнього середовища. У межах розроблюваного прототипу сенсорний модуль виконує функцію компактною «метеостанції», яка встановлюється безпосередньо на мобільну платформу та взаємодіє з центральним блоком керування. До складу сенсорного модуля входять датчики DHT22, BMP280, MQ-135 та LDR, які дають змогу вимірювати температуру, вологість, атмосферний тиск, орієнтовну висоту, якість повітря та рівень освітленості.

Основною вимогою до монтажу сенсорного модуля є правильне розміщення датчиків на платформі робота. Датчики мають бути встановлені так, щоб вони отримували достовірні показники, не зазнавали механічного пошкодження під час руху та не потрапляли під вплив тепла від двигунів, драйвера або стабілізаторів напруги. Тому сенсорний модуль доцільно розміщувати у верхній частині гусеничної платформи, на окремій монтажній

платі або стійці. Таке розташування забезпечує кращий доступ датчиків до повітряного середовища та зменшує вплив нагрівання від силових елементів.

Загальне розміщення сенсорного модуля метеомоніторингу на гусеничній платформі доцільно подати на рисунку 3.6.

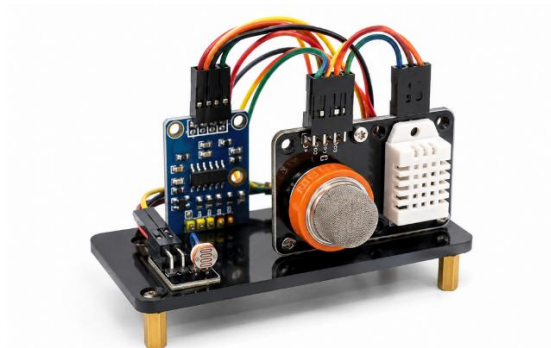


Рисунок 3.6 – Розміщення сенсорного модуля метеомоніторингу на гусеничному роботі-патрулі

Датчик DHT22 використовується для вимірювання температури та відносної вологості повітря. Його доцільно встановити у верхній частині платформи, на певній відстані від двигунів і силових модулів. Це пояснюється тим, що нагрівання електронних компонентів може спотворювати результати вимірювання температури. Крім того, датчик не повинен бути закритий корпусом або проводами, оскільки для коректного вимірювання вологості потрібний вільний доступ повітря до чутливого елемента. У схемі підключення DHT22 з'єднується з Arduino Uno через цифровий вивід, наприклад D2, а також підключається до ліній живлення 5 V і GND.

Датчик BMP280 призначений для вимірювання атмосферного тиску та розрахунку орієнтовної висоти. Його бажано монтувати на верхній монтажній панелі поруч із DHT22, але з урахуванням того, щоб сенсор не був закритий іншими деталями конструкції. BMP280 передає дані через цифровий інтерфейс I2C, тому для підключення до Arduino Uno використовуються лінії SDA та SCL. У платі Arduino Uno лінія SDA відповідає виводу A4, а SCL - виводу A5. Додатковою перевагою такого підключення є економія цифрових виводів

мікроконтролера, оскільки для обміну з BMP280 достатньо двох сигнальних ліній.

Датчик MQ-135 використовується для оцінювання якості повітря. Його чутливий елемент реагує на наявність газових домішок, диму та забруднювальних речовин, тому цей сенсор доцільно розмішувати в зоні, де повітря вільно надходить до поверхні датчика. У конструкції гусеничного робота MQ-135 можна встановити на передній або бічній частині верхньої монтажної платформи. Не рекомендується повністю закривати цей датчик корпусом, оскільки це може зменшити швидкість реакції на зміну якості повітря. У типовій схемі MQ-135 підключається до аналогового входу Arduino Uno, наприклад A0, а його значення зчитується функцією `analogRead()`.

LDR-фоторезистор використовується для визначення рівня освітленості. Його потрібно встановити у відкритій зоні, бажано у верхній або передній частині робота, щоб датчик міг реагувати на зміну зовнішнього освітлення. LDR підключається до Arduino Uno через подільник напруги, оскільки сам фоторезистор змінює опір залежно від інтенсивності світла. Вихід подільника напруги підключається до аналогового входу, наприклад A1. За отриманим аналоговим значенням можна оцінити умовний рівень освітленості: низький, середній або високий.

Орієнтовне підключення датчиків сенсорного модуля до Arduino Uno подано в таблиці 3.2.

Таблиця 3.2 – Підключення сенсорного модуля метеомоніторингу до Arduino Uno

Компонент	Вимірюваний параметр	Тип сигналу	Підключення до Arduino Uno	Призначення
DHT22	Температура, вологість	Цифровий	DATA - D2, VCC - 5 В, GND - GND	Контроль мікроклімату
BMP280	Атмосферний тиск, висота	I2C	SDA - A4, SCL - A5, VCC - 3,3 В або 5 В залежно від модуля, GND - GND	Вимірювання тиску та оцінка висоти
MQ-135	Якість повітря	Аналоговий	A0 - A0, VCC - 5 В, GND - GND	Виявлення зміни якості повітря
LDR	Освітленість	Аналоговий	Вихід подільника - A1, VCC - 5 В, GND - GND	Оцінювання рівня освітленості

Під час підключення апаратних компонентів важливо дотримуватися принципу спільної землі. Усі модулі – Arduino Uno, драйвер двигунів, сенсори, ESP8266 та стабілізатори напруги – повинні мати спільне з'єднання GND. Це забезпечує коректне визначення логічних рівнів і стабільний обмін сигналами між компонентами. Якщо спільна земля не буде реалізована, Arduino може неправильно зчитувати значення сенсорів або некоректно передавати сигнали до драйвера двигунів і модуля зв'язку.

Окрему увагу потрібно приділити підключенню силової частини. Двигуни гусеничної платформи не підключаються безпосередньо до Arduino Uno, оскільки вони споживають значно більший струм, ніж може забезпечити мікроконтролер. Для керування рухом використовується драйвер двигунів L298N або аналогічний модуль. Arduino Uno формує лише логічні сигнали керування, а драйвер подає необхідну напругу та струм на лівий і правий двигуни. Завдяки цьому можна реалізувати рух уперед, назад, повороти та зупинку платформи [21].

Схему взаємодії сенсорного модуля, блока керування, драйвера двигунів і системи живлення доцільно подати на рисунку 3.7.

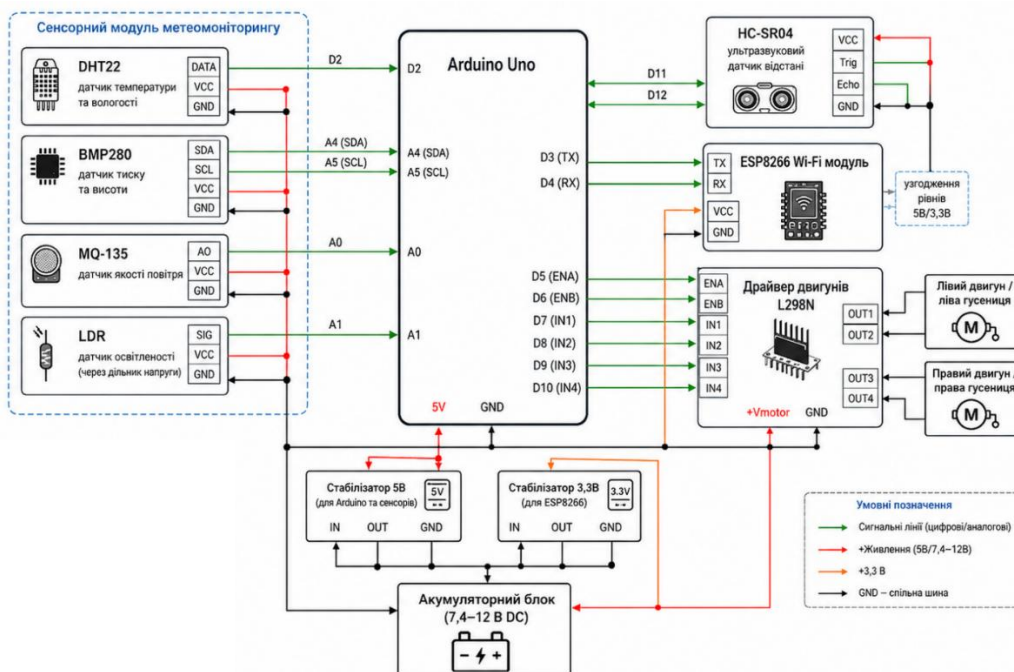


Рисунок 3.7 – Схема підключення сенсорного модуля та апаратних компонентів роботи-патруля

Система живлення прототипу повинна забезпечувати стабільну роботу логічної та силової частин. Для двигунів може використовуватися акумуляторний блок із напругою 7,4-12 В залежно від характеристик моторів. Для Arduino Uno та більшості сенсорів потрібне стабілізоване живлення 5 В. Для ESP8266 необхідно забезпечити стабільні 3,3 В, оскільки цей модуль чутливий до просідання напруги під час Wi-Fi-передавання. Тому доцільно використовувати окремий стабілізатор 3,3 В із достатнім струмом навантаження, а не покладатися лише на вивід 3,3 В плати Arduino Uno.

Для зменшення впливу електричних шумів сигнальні дроти сенсорів бажано прокладати окремо від силових ліній двигунів. Двигуни під час запуску та зміни напрямку обертання можуть створювати завади, які впливають на аналогові вимірювання MQ-135 та LDR. Тому доцільно використовувати короткі сигнальні дроти, фіксувати їх на платформі, уникати хаотичного переплетення провідників і за потреби встановлювати фільтрувальні конденсатори на лініях живлення. Це підвищує стабільність роботи сенсорного модуля та зменшує ймовірність випадкових коливань показників.

Процес монтажу апаратних компонентів виконувався поетапно. Спочатку на гусеничній платформі закріплювався блок керування Arduino Uno та драйвер двигунів. Далі підключалися лівий і правий двигуни до відповідних виходів драйвера L298N. Після перевірки правильності підключення силової частини встановлювався ультразвуковий датчик у передній частині робота. Наступним етапом був монтаж сенсорного модуля метеомоніторингу з датчиками DHT22, BMP280, MQ-135 та LDR. Після цього підключався модуль ESP8266 для передавання даних через Wi-Fi. Завершальним етапом була перевірка живлення, спільної землі та правильності з'єднання всіх сигнальних ліній.

Після завершення монтажу проводиться первинна перевірка апаратної частини. Спочатку перевіряється наявність живлення на Arduino Uno, сенсорах та модулі зв'язку. Далі через середовище Arduino IDE можна перевірити зчитування показників кожного датчика окремо. Після цього перевіряється робота двигунів через драйвер L298N: рух уперед, назад, поворот ліворуч,

поворот праворуч і зупинка. Такий порядок перевірки дає змогу швидше виявити помилки підключення та уникнути пошкодження компонентів.

Монтаж сенсорного модуля метеомоніторингу та підключення апаратних компонентів забезпечують формування повноцінної апаратної основи гусеничного робота-патруля. Датчики DHT22, BMP280, MQ-135 та LDR дають змогу збирати основні параметри середовища, Arduino Uno виконує функцію центрального блока керування, L298N забезпечує керування двигунами, ESP8266 реалізує IoT-передавання даних, а система живлення підтримує автономну роботу прототипу. Правильне розміщення сенсорів, розділення силової та сигнальної частин, стабілізація напруги та спільна земля є основними умовами надійної роботи апаратної системи

3.3 Розробка вбудованого програмного забезпечення та налаштування IoT-передавання даних

Розробка вбудованого програмного забезпечення є одним із ключових етапів технічної реалізації гусеничного робота-патруля екологічного моніторингу, оскільки саме програмна частина забезпечує взаємодію між сенсорами, блоком керування, драйвером двигунів, модулем зв'язку та IoT-платформою. У межах прототипу програмне забезпечення виконує кілька основних функцій: ініціалізацію апаратних компонентів, зчитування показників із датчиків, фільтрацію та попередню обробку даних, керування рухом платформи, формування пакета метеоданих і передавання інформації через Wi-Fi-модуль ESP8266.

Для програмування робота-патруля доцільно використовувати середовище Arduino IDE, оскільки воно підтримує мову Arduino C/C++, роботу з бібліотеками, послідовний монітор для налагодження та завантаження програмного коду на плату Arduino Uno [14, 15]. Arduino Uno у цій системі виконує роль центрального контролера, який отримує дані з сенсорів DHT22,

BMP280, MQ-135, LDR та ультразвукового датчика, а також формує керуючі сигнали для драйвера двигунів L298N.

Програмна архітектура прототипу має модульний характер. Окремо виділяється модуль ініціалізації, модуль зчитування сенсорів, модуль обробки даних, модуль керування рухом, модуль формування пакета даних і модуль передавання інформації через ESP8266. Такий підхід спрощує налагодження системи, оскільки кожен програмний блок можна перевіряти окремо. Наприклад, спочатку перевіряється коректність зчитування DHT22 і BMP280, далі – робота аналогових входів MQ-135 та LDR, після цього – керування двигунами через L298N, а на завершальному етапі - передавання даних через Wi-Fi.

Основний цикл роботи програми складається з послідовних операцій. Після запуску системи виконується ініціалізація портів Arduino Uno, бібліотек для роботи з датчиками, послідовного інтерфейсу для обміну з ESP8266, а також встановлення початкових параметрів руху. Далі програма переходить у нескінченний цикл `loop()`, у якому періодично виконується вимірювання відстані до перешкоди, зчитування метеопараметрів, аналіз отриманих значень, керування рухом і передавання даних на IoT-платформу.

Для роботи з датчиком DHT22 доцільно використовувати відповідну бібліотеку Arduino, яка спрощує отримання температури та вологості у фізичних одиницях. Датчик BMP280 підключається через інтерфейс I2C і дає змогу отримувати значення атмосферного тиску, а також розраховувати орієнтовну висоту. Аналогові датчики MQ-135 та LDR зчитуються за допомогою функції `analogRead()`, яка на платі Arduino Uno повертає значення в діапазоні від 0 до 1023. Для зменшення випадкових коливань аналогових сигналів доцільно виконувати кілька послідовних вимірювань і розраховувати середнє значення.

Загальну структуру вбудованого програмного забезпечення робота-патруля доцільно подати на рисунку 3.8.

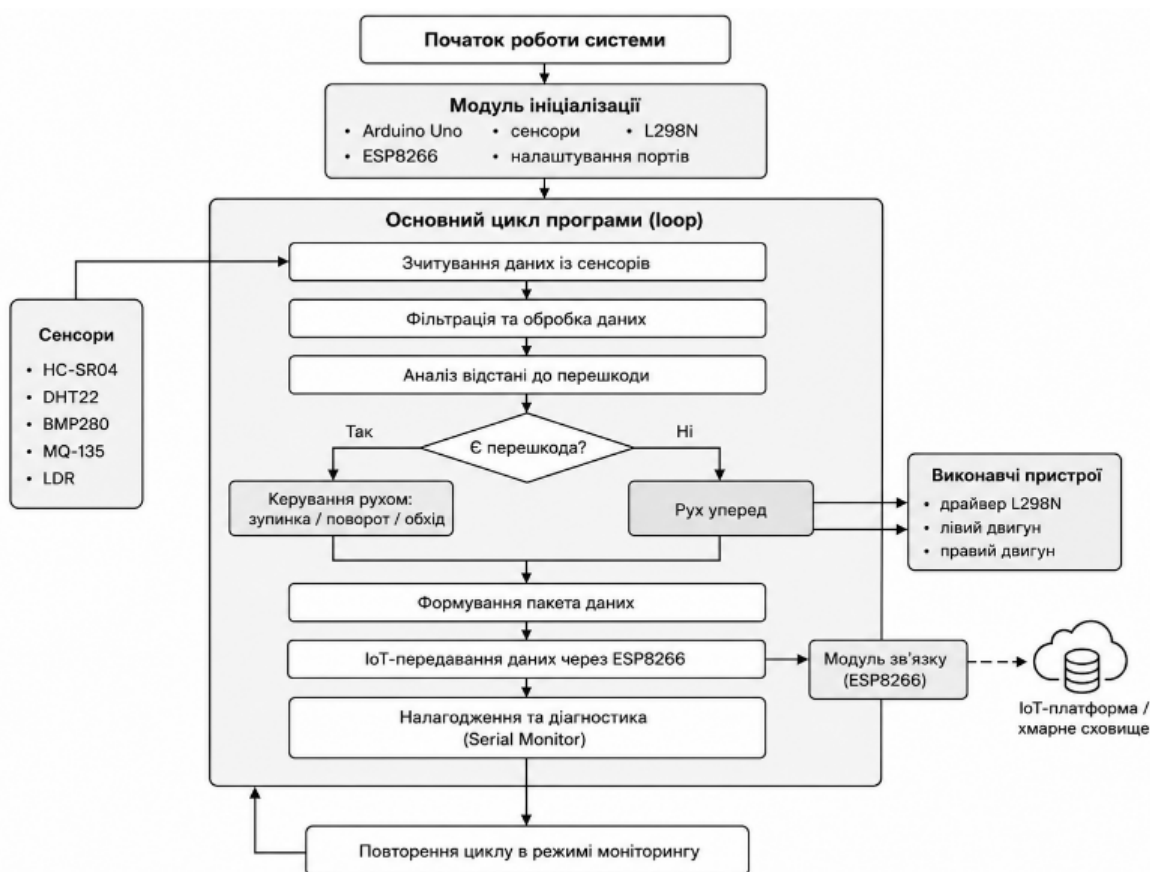


Рисунок 3.8 – Структура вбудованого програмного забезпечення гусеничного робота-патруля

Орієнтовну структуру програмних модулів подано в таблиці 3.3.

Таблиця 3.3 – Основні програмні модулі вбудованого ПЗ робота-патруля

Програмний модуль	Основні функції	Результат роботи
Модуль ініціалізації	Налаштування портів, запуск бібліотек, перевірка сенсорів	Підготовка системи до роботи
Модуль зчитування сенсорів	Отримання даних із DHT22, BMP280, MQ-135, LDR і HC-SR04	Первинні значення параметрів середовища
Модуль фільтрації	Усереднення аналогових показників, перевірка допустимих меж	Стабілізовані дані для обробки
Модуль керування рухом	Формування сигналів для L298N, зупинка або поворот при перешкоді	Безпечне переміщення платформи
Модуль IoT-передавання	Формування рядка або JSON-пакета, передавання через ESP8266	Дані передані на сервер або хмарну платформу
Модуль налагодження	Виведення показників у Serial Monitor	Контроль роботи під час тестування

Під час розробки firmware важливо правильно організувати часові інтервали. Датчики метеомоніторингу не потребують зчитування з дуже високою

частотою, тому оптимальним є період вимірювання 5-10 секунд. Водночас ультразвуковий датчик для виявлення перешкод повинен опитуватися частіше, оскільки від цього залежить безпечний рух робота. Тому в програмі доцільно розділити швидкий цикл навігації та повільніший цикл метеомоніторингу. Такий підхід дає змогу не перевантажувати мікроконтролер і забезпечити своєчасну реакцію на перешкоди.

Для передавання даних у хмарне сховище або на IoT-платформу використовується модуль ESP8266. Він забезпечує підключення до Wi-Fi-мережі та може передавати дані на сервер за допомогою HTTP-запитів або MQTT-протоколу. Для навчального прототипу найпростішим варіантом є HTTP-передавання даних на платформу ThingSpeak або локальний сервер, оскільки цей підхід не потребує складної серверної архітектури та дозволяє швидко візуалізувати показники у вигляді графіків. MQTT доцільно розглядати як перспективний варіант для подальшого розширення системи, особливо якщо планується підключення кількох роботів або сенсорних вузлів.

Схему передавання даних від сенсорів до IoT-платформи доцільно подати на рисунку 3.9.

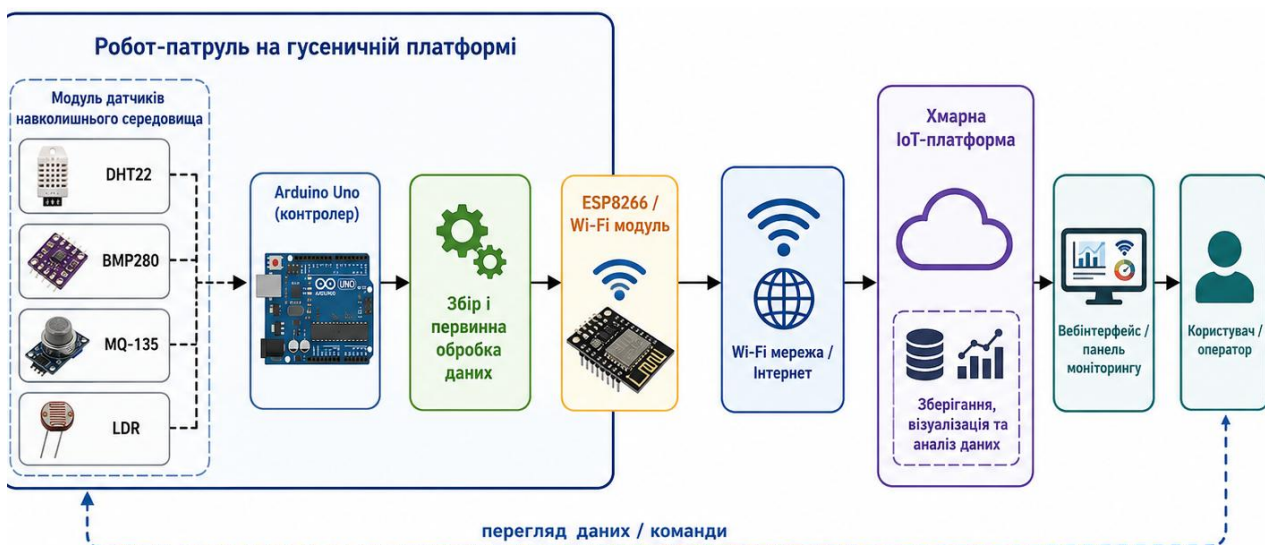


Рисунок 3.9 – Схема IoT-передавання даних метеомоніторингу від робота-патруля до хмарної платформи

Під час налаштування IoT-передавання даних спочатку задаються параметри Wi-Fi-мережі: назва мережі, пароль доступу, адреса сервера або API-ключ IoT-платформи. Після цього ESP8266 виконує підключення до мережі та очікує на отримання пакета даних від Arduino Uno. Arduino передає сформовані показники через послідовний інтерфейс UART. ESP8266 приймає ці дані, формує HTTP-запит або MQTT-повідомлення та надсилає їх до хмарного сервісу. У разі помилки з'єднання модуль повинен виконати повторну спробу передавання.

Для налагодження програмного забезпечення використовувався послідовний монітор Arduino IDE. Через нього можна перевіряти поточні значення температури, вологості, тиску, якості повітря, освітленості та відстані до перешкоди. Також у Serial Monitor доцільно виводити службові повідомлення, наприклад «Wi-Fi connected», «Sensor error», «Data sent», «Obstacle detected». Це спрощує пошук помилок, оскільки дає змогу бачити, на якому етапі виникає проблема: під час зчитування сенсора, формування пакета чи передавання даних.

Окрему увагу потрібно приділити обробці помилок. Якщо датчик DHT22 тимчасово не повертає коректне значення, програма не повинна аварійно завершувати роботу. У такому випадку доцільно пропустити поточне вимірювання або використати попереднє достовірне значення. Якщо ESP8266 втрачає Wi-Fi-з'єднання, система може продовжувати локальне зчитування сенсорів і повторювати спробу підключення через певний інтервал часу. Якщо ультразвуковий датчик фіксує критично малу відстань, керування рухом має мати вищий пріоритет, ніж передавання метеоданих.

Вбудоване програмне забезпечення гусеничного робота-патруля забезпечує узгоджену роботу всіх апаратних компонентів системи. Arduino Uno виконує зчитування сенсорів, обробку даних і керування рухом, а ESP8266 забезпечує передавання сформованих показників до IoT-платформи. Запропонована структура firmware дозволяє реалізувати базову функціональність мобільної метеостанції: автономний рух, виявлення перешкод, збирання

параметрів середовища, попередню фільтрацію даних і віддалений моніторинг результатів.

3.4 Експериментальна перевірка роботи прототипу та аналіз результатів

Експериментальна перевірка роботи прототипу є завершальним етапом технічної реалізації гусеничного робота-патруля екологічного моніторингу. Її метою є оцінювання працездатності фізичної моделі, правильності підключення апаратних компонентів, стабільності зчитування даних із сенсорного модуля, ефективності алгоритму руху, автономності живлення та якості передавання даних через IoT-канал. У межах перевірки особлива увага приділялася роботі сенсорів DHT22, BMP280, MQ-135 і LDR, функціонуванню ультразвукового датчика для виявлення перешкод, а також взаємодії Arduino Uno з модулем ESP8266.

Загальну послідовність експериментальної перевірки прототипу доцільно подати на рисунку 3.10.



Рисунок 3.10 – Послідовність експериментальної перевірки гусеничного робота-патруля

Експериментальні дослідження проводилися поетапно. Спочатку перевірялася працездатність окремих апаратних компонентів: плати Arduino Uno, драйвера двигунів, ультразвукового датчика, сенсорів метеомоніторингу та модуля зв'язку. Після цього виконувалося тестування зібраної системи в цілому. Такий підхід дозволив виявляти можливі помилки підключення поступово, не перевантажуючи систему одночасною роботою всіх модулів.

На першому етапі виконувалося тестування сенсорного модуля метеомоніторингу. Датчик DHT22 використовувався для вимірювання температури та вологості повітря, BMP280 – для визначення атмосферного тиску, MQ-135 – для оцінювання якості повітря, а LDR – для визначення рівня освітленості. Для перевірки коректності вимірювань показники сенсорів порівнювалися з умовно еталонними або контрольними приладами: побутовим термометром, гігрометром, барометром або даними з метеосервісу для атмосферного тиску. Показники MQ-135 і LDR оцінювалися переважно у відносних значеннях, оскільки ці датчики потребують додаткового калібрування для отримання точних фізичних одиниць. Результати перевірки показників сенсорного модуля показано в таблиці 3.4.

Таблиця 3.4 – Результати перевірки показників сенсорного модуля метеомоніторингу

Параметр	Сенсор	Контрольне значення	Значення сенсора	Відхилення	Висновок
Температура повітря, °C	DHT22	24,0	24,4	0,4	Відхилення допустиме
Вологість повітря, %	DHT22	58	60	2	Показник стабільний
Атмосферний тиск, гПа	BMP280	1009	1008	1	Відхилення незначне
Якість повітря, ум. од.	MQ-135	-	365	-	Використовується як відносний показник
Освітленість, ум. од.	LDR	-	742	-	Реагує на зміну освітлення

За результатами тестування встановлено, що датчики DHT22 і BMP280 забезпечують стабільні значення, придатні для використання у прототипі мобільної метеостанції. Незначні відхилення температури, вологості та тиску

можуть бути пов'язані з особливостями розміщення сенсорів на платформі, нагріванням електронних компонентів, циркуляцією повітря та похибками самих вимірювальних модулів. Для підвищення точності вимірювань доцільно передбачити коротку зупинку роботи перед зчитуванням метеоданих, а також програмне усереднення кількох послідовних вимірювань.

На другому етапі перевірялася робота алгоритму руху та виявлення перешкод. Для цього робот встановлювався на рівну поверхню, після чого перед ним розміщувалися перешкоди на різній відстані. Ультразвуковий датчик HC-SR04 вимірював відстань до об'єкта, а Arduino Uno на основі отриманих даних формувала команду для драйвера двигунів L298N. Якщо відстань була більшою за встановлений безпечний поріг, робот продовжував рух. Якщо відстань зменшувалася до критичного значення, система зупиняла платформу та виконувала поворот для обходу перешкоди.

Результати перевірки роботи алгоритму обходу перешкод можна подати в таблиці 3.5.

Таблиця 3.5 – Результати тестування виявлення перешкод і реакції робота

Відстань до перешкоди, см	Стан системи	Реакція робота	Результат
50	Перешкода далеко	Рух уперед	Рух стабільний
30	Наближення до перешкоди	Зменшення швидкості	Перешкоду виявлено
20	Критична зона	Зупинка	Зіткнення не відбулося
15	Небезпечна зона	Зупинка та поворот	Робот змінив напрямок
10	Дуже близька перешкода	Аварійна зупинка	Рух заблоковано

Отримані результати свідчать, що ультразвуковий датчик забезпечує достатню точність для базового виявлення перешкод у межах навчального прототипу. Найкращі результати досягаються тоді, коли перешкода має рівну поверхню та розташована безпосередньо перед датчиком. Якщо об'єкт має складну форму, нахилену поверхню або погано відбиває ультразвуковий сигнал, показники можуть бути менш стабільними. Тому для підвищення надійності

системи доцільно виконувати повторне вимірювання відстані перед прийняттям рішення про поворот або зупинку.

На третьому етапі досліджувалася автономність роботи прототипу. Час роботи залежить від ємності акумулятора, споживання двигунів, кількості активних сенсорів, частоти передавання даних через ESP8266 та режиму руху. Найбільше навантаження створюють двигуни гусеничної платформи, особливо під час старту, поворотів і руху по поверхні з підвищеним опором. Тому тестування автономності доцільно проводити у кількох режимах: без руху, під час рівномірного руху та під час руху з частими поворотами. Результати тестування автономності прототипу показано в таблиці 3.6.

Таблиця 3.6 – Результати тестування автономності роботи прототипу

Режим роботи	Умови тестування	Орієнтовний час роботи	Характеристика результату
Сенсорний режим	Робот нерухомий, працюють сенсори та ESP8266	2 год 40 хв	Найменше енергоспоживання
Рівномірний рух	Рух по рівній поверхні з періодичним передаванням даних	1 год 25 хв	Стабільна робота системи
Патрулювання	Рух, повороти, зупинки, передавання даних	1 год 05 хв	Найбільше навантаження на акумулятор
Рух із частими перешкодами	Часті зупинки та повороти	50 хв	Зростає споживання двигунів

Результати тестування автономності показують, що час роботи прототипу суттєво залежить від режиму руху. У сенсорному режимі енергоспоживання є нижчим, оскільки двигуни не працюють. Під час активного патрулювання час автономної роботи зменшується через постійне навантаження на гусеничні приводи. Для підвищення автономності доцільно використовувати акумулятор більшої ємності, оптимізувати швидкість руху, зменшити частоту передавання даних або реалізувати енергоощадні режими роботи модуля ESP8266.

На четвертому етапі оцінювалася прохідність гусеничної платформи. Тестування проводилося на різних типах поверхонь: рівна підлога, килимове покриття, плитка, гумове покриття та поверхня з незначними нерівностями.

Гусенична конструкція забезпечила стабільний контакт із поверхнею та кращу стійкість під час поворотів порівняно з типовими малими колісними платформами. Водночас на дуже гладкій або слизькій поверхні можливе часткове проковзування гусениць, що впливає на точність руху. Результати оцінювання прохідності показано в таблиці 3.7.

Таблиця 3.7 – Результати оцінювання прохідності гусеничної платформи

Тип поверхні	Рух уперед	Повороти	Стабільність платформи	Загальна оцінка
Рівна тверда поверхня	Стабільний	Стабільні	Висока	Добре
Килимове покриття	Стабільний	Незначне уповільнення	Висока	Добре
Плитка	Стабільний	Можливе ковзання	Середня	Задовільно
Нерівна поверхня	Рух можливий	Повороти ускладнені	Середня	Задовільно
Поверхня з дрібними перешкодами	Рух частково можливий	Потребує корекції	Середня	Задовільно

На п'ятому етапі перевірялася стабільність IoT-передавання даних. Для цього робот передавав показники сенсорів через ESP8266 до хмарної платформи або локального сервера. Оцінювалися такі параметри: наявність з'єднання з Wi-Fi-мережею, регулярність оновлення даних, час затримки між вимірюванням і появою значення на платформі, а також стабільність передавання на різних відстанях від маршрутизатора. У межах базового прототипу оптимальним є інтервал оновлення даних 5-10 секунд. Результати тестування стабільності показано в таблиці 3.8.

Таблиця 3.8 – Результати тестування стабільності IoT-передавання даних

Відстань від маршрутизатора	Стан з'єднання	Оновлення даних	Характеристика роботи
2 м	Стабільне	Кожні 5 с	Дані передаються без затримок
5 м	Стабільне	Кожні 5-7 с	Передавання стабільне
10 м	Переважно стабільне	Кожні 7-10 с	Можливі незначні затримки
15 м	Нестабільне	Нерегулярне	Потрібне покращення покриття
За перешкодою	Залежить від умов	Можливі втрати	Потрібне тестування в реальному середовищі

За результатами тестування встановлено, що ESP8266 забезпечує достатню стабільність передавання даних у межах локальної Wi-Fi-мережі. Найкращі результати отримуються за прямої видимості між роботом і маршрутизатором. У разі збільшення відстані або наявності стін якість зв'язку може погіршуватися, що призводить до затримок або втрати окремих пакетів даних. Для підвищення стабільності IoT-зв'язку можна використовувати маршрутизатор із кращим покриттям, зовнішню антену, повторювач Wi-Fi або GSM-модуль для роботи поза локальною мережею.

У процесі експериментальної перевірки було також оцінено загальну працездатність системи. Прототип виконував базові функції мобільного екологічного моніторингу: рухався на гусеничній платформі, виявляв перешкоди, зупинявся або змінював напрямок руху, зчитував показники сенсорів і передавав їх через IoT-канал. Найбільш стабільними виявилися цифрові сенсори DHT22 та BMP280. Аналогові сенсори MQ-135 і LDR потребують програмного усереднення та калібрування, оскільки їхні значення залежать від умов середовища, живлення та схеми підключення.

Загалом результати експериментів підтвердили працездатність розробленого прототипу. Гусеничне шасі забезпечує достатню стійкість і прохідність для навчального робота-патруля, сенсорний модуль дає змогу отримувати основні параметри середовища, а IoT-модуль забезпечує передавання даних користувачу. Водночас у процесі тестування було визначено напрями подальшого вдосконалення системи: підвищення точності калібрування сенсорів, збільшення автономності живлення, покращення стабільності Wi-Fi - зв'язку, оптимізація алгоритму обходу перешкод і додавання GPS-модуля для фіксації координат точок заміру.

Експериментальна перевірка показала, що створений прототип гусеничного робота-патруля екологічного моніторингу є функціонально придатним для виконання базових завдань. Він може використовуватися як навчальна модель мобільної IoT-системи, яка поєднує механічну платформу, сенсорний блок, вбудоване програмне забезпечення, алгоритм автономного руху

та передавання даних у хмарне середовище. Отримані результати підтверджують доцільність подальшого розвитку системи в напрямі підвищення точності вимірювань, розширення набору сенсорів і вдосконалення алгоритмів навігації.

ВИСНОВКИ

У кваліфікаційній роботі було розв'язано науково-прикладне завдання, пов'язане з проєктуванням, програмно-апаратною реалізацією та експериментальним дослідженням системи автономного патрулювання й екологічного моніторингу на основі гусеничної мобільної платформи. Розроблена система поєднує засоби мобільної робототехніки, мікроконтролерного керування, сенсорного вимірювання параметрів середовища та IoT-передавання даних, що дає змогу розглядати її як прототип мобільної метеостанції для контролю стану навколишнього середовища.

У процесі дослідження було проаналізовано сучасні підходи до побудови мобільних робототехнічних комплексів і систем моніторингу середовища. Встановлено, що традиційні стаціонарні системи екологічного контролю, хоча й забезпечують достатню точність вимірювань, мають обмежене просторове охоплення. На відміну від них, мобільні роботи-патрулі можуть переміщуватися між різними точками контрольованої території, виконувати вимірювання в динаміці та формувати більш повну картину зміни екологічних показників. У роботі розглянуто різні типи мобільних платформ, серед яких колісні, гусеничні, крокуючі, безпілотні та гібридні системи. Обґрунтовано доцільність використання саме гусеничного шасі, оскільки воно забезпечує підвищену прохідність, стійкість і кращий контакт із поверхнею під час руху по нерівних або недостатньо підготовлених ділянках.

У роботі обґрунтовано апаратну архітектуру прототипу на базі контролера Arduino Uno. Цей мікроконтролер обрано як центральний блок керування, оскільки він має достатню кількість цифрових і аналогових входів/виходів, підтримує підключення поширених сенсорів, драйвера двигунів і комунікаційного модуля. Для формування сенсорного модуля метеомоніторингу використано датчики DHT22, BMP280, MQ-135 та LDR. Вони забезпечують вимірювання температури, вологості, атмосферного тиску, оцінювання якості повітря та рівня освітленості. Такий набір сенсорів є достатнім для створення

базової мобільної системи екологічного моніторингу та може бути розширений у майбутньому додатковими вимірювальними модулями.

Під час розробки апаратної частини було сформовано принципову схему підключення основних компонентів робота-патруля. Описано підключення сенсорного модуля, ультразвукового датчика, драйвера двигунів L298N, Wi-Fi - модуля ESP8266 та системи автономного живлення. Визначено, що для стабільної роботи прототипу необхідно розділяти силову й логічну частини, використовувати стабілізовані лінії живлення 5 В і 3,3 В, забезпечувати спільну землю GND для всіх компонентів та враховувати вплив електричних завад від двигунів на аналогові показники сенсорів.

Алгоритмічне забезпечення системи розроблено з урахуванням потреб автономного руху, виявлення перешкод і виконання вимірювань параметрів середовища. Для навігації використано ультразвуковий датчик, який визначає відстань до перешкоди перед роботом. На основі отриманих значень система приймає рішення щодо подальшого руху: продовжує переміщення, зменшує швидкість, зупиняється або змінює напрямок. Запропонований алгоритм є простим для реалізації на Arduino Uno та забезпечує базову автономність руху без використання складних методів картографування, комп'ютерного зору чи GPS-навігації.

У межах програмної реалізації створено комплекс для збирання, попередньої обробки та передавання метеоданих. Вбудоване програмне забезпечення забезпечує ініціалізацію сенсорів, зчитування показників, фільтрацію аналогових сигналів, формування пакета даних і передавання інформації через модуль ESP8266 до хмарного інтерфейсу. Для цифрових сенсорів DHT22 і BMP280 використано бібліотечний підхід, що спрощує отримання температури, вологості та атмосферного тиску. Для аналогових датчиків MQ-135 та LDR застосовано зчитування через аналогові входи Arduino з подальшим усередненням значень, що дозволяє зменшити вплив випадкових шумів і коливань сигналу.

У процесі технічної реалізації було створено фізичну модель гусеничного робота-патруля. На платформі розміщено блок керування Arduino Uno, драйвер двигунів L298N, ультразвуковий датчик для виявлення перешкод, сенсорний модуль метеомоніторингу, модуль зв'язку ESP8266 та систему живлення. Під час монтажу сенсорного блока враховано необхідність правильного розміщення датчиків: DHT22 і BMP280 встановлено у зоні з вільним доступом повітря, MQ-135 розміщено так, щоб забезпечити надходження повітря до чутливого елемента, а LDR розташовано у відкритій зоні для коректного визначення освітленості.

Експериментальна перевірка підтвердила працездатність розробленого прототипу. Було перевірено функціонування сенсорного модуля, системи руху, алгоритму виявлення перешкод, автономного живлення та IoT-передавання даних. Встановлено, що датчики DHT22 і BMP280 забезпечують достатньо стабільні показники для навчального прототипу мобільної метеостанції. Датчики MQ-135 та LDR доцільно використовувати як індикаторні сенсори, оскільки їхні значення залежать від умов калібрування, стабільності живлення, температури, вологості та зовнішнього освітлення.

За результатами експериментів встановлено, що гусенична платформа забезпечує достатню стійкість і прохідність на рівних та помірно нерівних поверхнях. Система виявлення перешкод на основі ультразвукового датчика дає змогу своєчасно зупиняти платформу або змінювати напрямок руху. Передавання даних через ESP8266 забезпечує можливість дистанційного моніторингу показників середовища, однак стабільність зв'язку залежить від відстані до маршрутизатора, наявності перешкод і якості Wi-Fi-покриття.

Практичне значення отриманих результатів полягає у створенні функціонального прототипу мобільної метеостанції на базі гусеничного робота-патруля. Розроблена система може бути використана як основа для подальшого створення засобів екологічного моніторингу складських приміщень, теплиць, навчальних лабораторій, виробничих зон або прилеглих територій підприємств. Крім того, запропонований прототип має навчальну цінність, оскільки

демонструє комплексне застосування робототехніки, мікроконтролерного керування, сенсорних мереж, вбудованого програмного забезпечення та IoT-технологій.

Спроектвано, реалізовано та експериментально перевірено систему автономного патрулювання й екологічного моніторингу на основі гусеничної мобільної платформи. Розроблений прототип забезпечує автономний рух, виявлення перешкод, збирання параметрів навколишнього середовища, їх первинну обробку та дистанційне передавання користувачу. Перспективами подальшого вдосконалення є додавання GPS-модуля для фіксації координат точок вимірювання, підвищення автономності живлення, покращення алгоритму навігації, розширення набору сенсорів і створення зручнішої панелі візуалізації результатів моніторингу.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Review on Robotic Systems for Environmental Monitoring. *IEEE Open Journal of Instrumentation and Measurement*. URL: <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10806561> (date of access:18.01.2026).
2. Design, Analysis and Control of Tracked Mobile Robot with Passive Suspension on Rugged Terrain. *Actuators*. URL: <https://www.mdpi.com/2076-0825/14/8/389> (date of access:18.01.2026).
3. Research on Trajectory Tracking Control Method for Crawler Robot Based on Improved PSO Sliding Mode Disturbance Rejection Control. *Sensors*. URL: <https://www.mdpi.com/1424-8220/25/7/2113> (date of access:18.01.2026).
4. Construction and benchmark of an autonomous tracked mobile robot system. *Robotic Systems and Applications*. URL:https://www.researchgate.net/publication/358248333_Construction_and_benchmark_of_an_autonomous_tracked_mobile_robot_system (date of access:18.01.2026).
5. Sensor-Fusion Based Navigation for Autonomous Mobile Robot. *Sensors*. URL: <https://www.mdpi.com/1424-8220/25/4/1248> (date of access:18.01.2026).
6. Autonomous Navigation and Collision Avoidance for Mobile Robots: Classification and Review. *arXiv*. URL: <https://arxiv.org/abs/2410.07297> (date of access:18.01.2026).
7. Autonomous Mobile Robot Path Planning Techniques – A Review: Metaheuristic and Cognitive Techniques. *Robotics*. URL: <https://www.mdpi.com/2218-6581/15/1/23> (date of access:18.01.2026).
8. Autonomous Mobile Robot Path Planning Techniques – A Review: Classical and Heuristic Techniques. *IEEE Access*. URL: <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=11036715> (date of access:18.01.2026).
9. Distributed Multi-Mobile Robot Path Planning and Obstacle Avoidance Based on ACO-DWA in Unknown Complex Terrain. *Electronics*. URL: <https://www.mdpi.com/2079-9292/14/16/3290> (date of access:25.02.2026).

10. Obstacle Avoidance Technique for Mobile Robots at Autonomous Human-Robot Collaborative Warehouse Environments. *Sensors*. URL: <https://www.mdpi.com/1424-8220/25/8/2387> (date of access:25.02.2026).
11. Design and Research of an Articulated Tracked Firefighting Robot. *Sensors*. URL: <https://www.mdpi.com/1424-8220/22/14/5086> (date of access:25.02.2026).
12. Analysis of kinematic characteristics of a mobile caterpillar robot. *Transport Technologies*. URL: <https://science.lpnu.ua/tt/all-volumes-and-issues/vipusk-4-nomer-2-2023/analysis-kinematic-characteristics-mobile> (date of access:25.02.2026).
13. UNO R3. *Arduino Documentation*. URL: <https://docs.arduino.cc/hardware/uno-rev3/> (date of access:01.03.2026).
14. Getting Started with Arduino UNO R3. *Arduino Documentation*. URL: <https://docs.arduino.cc/tutorials/uno-rev3/getting-started/> (date of access:04.03.2026).
15. Getting Started with Arduino IDE 2. *Arduino Documentation*. URL: <https://docs.arduino.cc/software/ide-v2/tutorials/getting-started-ide-v2> (date of access:07.03.2026).
16. Language Reference. *Arduino Documentation*. URL: <https://docs.arduino.cc/language-reference/> (date of access:10.03.2026).
17. Installing Libraries. *Arduino Documentation*. URL: <https://www.arduino.cc/en/Guide/libraries> (date of access:13.03.2026).
18. DHT sensor library. *Arduino Documentation*. URL: <https://docs.arduino.cc/libraries/dht-sensor-library/> (date of access:16.03.2026).
19. BMP280. *Arduino Documentation*. URL: <https://docs.arduino.cc/libraries/bmp280/> (date of access:19.03.2026).
20. AnalogRead(). *Arduino Documentation*. URL: <https://docs.arduino.cc/language-reference/en/functions/analog-io/analogRead/> (date of access:22.03.2026).
21. DigitalRead(). *Arduino Documentation*. URL: <https://www.arduino.cc/reference/en/language/functions/digital-io/digitalread/> (date of access:25.03.2026).
22. ArduinoHttpClient. *Arduino Documentation*. URL: <https://docs.arduino.cc/libraries/arduinohttpclient/> (date of access:28.03.2026).

23. ThingSpeak Communication Library for Arduino, ESP8266 and ESP32. *MathWorks GitHub Repository*. URL: <https://github.com/mathworks/thingspeak-arduino> (date of access:31.03.2026).

24. ESP8266 Technical Reference. Version 1.7. *Espressif Systems*. URL: https://www.espressif.com/sites/default/files/documentation/esp8266-technical_reference_en.pdf (date of access:03.04.2026).

25. ESP8266 Wi-Fi SoC. *Espressif Systems*. URL: <https://www.espressif.com/en/products/socs/esp8266> (date of access:06.04.2026).

26. MQTT 5.0: 7 New Features and a Migration Checklist. *EMQX*. URL: <https://www.emqx.com/en/blog/introduction-to-mqtt-5> (date of access:10.04.2026).

27. How to use the L298N motor driver module. *Arduino Project Hub*. URL: <https://projecthub.arduino.cc/hibit/how-to-use-the-l298n-motor-driver-module-0bb697> (date of access:15.04.2026).

28. Ultrasonic sensor with Arduino – Complete Guide. *Arduino Project Hub*. URL: <https://projecthub.arduino.cc/lucasfernando/ultrasonic-sensor-with-arduino-complete-guide-284faf> (date of access:20.04.2026).

