

Міністерство освіти і науки України

Луцький національний технічний університет

(повне найменування закладу вищої освіти)

Факультет комп'ютерних та інформаційних технологій

(повне найменування факультету)

Кафедра комп'ютерної інженерії та безпеки

(повне найменування кафедри)

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

**СИСТЕМА ЛАЗЕРНОГО ПОЗИЦІОНУВАННЯ НА БАЗІ LIDAR
ТА ARDUINO**

**LASER POSITIONING SYSTEM BASED ON LIDAR AND
ARDUINO**

спеціальність 123 Комп'ютерна інженерія

(шифр і назва спеціальності)

освітня програма Комп'ютерна інженерія

(назва освітньої програми)

Виконав: здобувач вищої освіти
групи КІ-41
Міркевич Вадим Васильович

(підпис)

Керівник:
асистент
Конкевич Людмила Миколаївна

(підпис)

Кваліфікаційну роботу
допущено до захисту
« 10 » червня 2025 р.

Гарант освітньої програми:

к.т.н., доцент
Лавренчук Світлана Василівна

(підпис)

Луцьк – 2025 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра комп'ютерної інженерії та безпеки

Ступінь вищої освіти: бакалавр

Галузь знань: 12 Інформаційні технології

Спеціальність: 123 Комп'ютерна інженерія

Освітня програма: «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

доц. Тарас ТЕРЛЕЦЬКИЙ

« 10 » 01 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Міркевичу Вадиму Васильовичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи Система лазерного позиціонування на базі Lidar та Arduino

Керівник роботи асистент, Конкевич Людмила Миколаївна

затвержені наказом закладу вищої освіти від «04» січня 2025 року № 11/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи 10.06.2025р.

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

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

Вступ

Аналіз предметної області та наявних рішень

Вибір апаратної та програмної бази для проєкту

Практична реалізація системи лазерного позиціонування

Висновки

5. Перелік графічного (ілюстративного) матеріалу:

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
<i>Аналіз предметної області та наявних рішень</i>	<i>Конкевич Л.М., асистент</i>		
<i>Вибір апаратної та програмної бази для проєкту</i>	<i>Конкевич Л.М., асистент</i>		
<i>Практична реалізація системи лазерного позиціонування</i>	<i>Конкевич Л.М., асистент</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 10.01.2025 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 10.02.2025 р.	Виконано
2.	<i>Вибір апаратної та програмної бази для проєкту</i>	до 02.04.2025 р.	Виконано
3.	<i>Практична реалізація системи лазерного позиціонування</i>	до 02.04.2025 р.	Виконано
4.	<i>Висновки та пропозиції</i>	до 10.04.2025 р.	Виконано
5.	<i>Формування списку використаних джерел</i>	до 15.04.2025 р.	Виконано
6.	<i>Формування додатків</i>	до 02.05.2025 р.	Виконано
7.	<i>Оформлення ілюстративного матеріалу</i>	до 10.05.2025 р.	Виконано
8.	<i>Представлення остаточного варіанту кваліфікаційної роботи керівникові</i>	до 15.05.2025 р.	Виконано
9.	<i>Нормоконтроль</i>	до 30.05.2025 р.	Виконано
10.	<i>Інструментальна перевірка на академічний плагіат</i>	до 03.06.2025 р.	Виконано
11.	<i>Здача кваліфікаційної роботи та всіх супровідних документів на кафедрі</i>	до 10.06.2025 р.	Виконано

Здобувач вищої освіти

_____ (підпис)

Міркевич В.В.

_____ (прізвище, ініціали)

Керівник кваліфікаційної роботи

_____ (підпис)

Конкевич Л.М.

_____ (прізвище, ініціали)

АНОТАЦІЯ

Міркевич В.В. Система лазерного позиціонування на базі Lidar та Arduino. Рукопис.

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

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

У першому розділі проаналізовано фізичні принципи вимірювання відстані за допомогою лазера, класифікацію та застосування Lidar-систем, а також переваги та обмеження даної технології.

У другому розділі здійснено вибір та обґрунтування складових системи: Arduino, сенсора, сервоприводів та програмних засобів розробки.

Третій розділ присвячено опису процесу розробки системи лазерного позиціонування на базі мікроконтролерної платформи Arduino та лазерного дальноміра типу Lidar.

Ключові слова: Lidar, Arduino, сервопривод, позиціонування, VL53L0X.

ANNOTATION

Mirkevych V. Laser positioning system based on Lidar and Arduino. Manuscript.

Qualifying work of a bachelor of EP «Computer Engineering» specialty 123 Computer Engineering. Lutsk National Technical University. Lutsk, 2025.

Qualification work consists of an introduction, three sections, conclusions, a references.

The first section is analyzes the physical principles of measuring distance using a laser, the classification and application of Lidar systems, as well as the advantages and limitations of this technology.

The second section selects and justifies the components of the system: Arduino, sensor, servos and development software.

The third section is devoted to describing the process of developing a laser positioning system based on the Arduino microcontroller platform and a Lidar-type laser rangefinder.

Keywords: Lidar, Arduino, servo, positioning, VL53L0X.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ ЛАЗЕРНОГО ПОЗИЦІОНУВАННЯ	8
1.1 Принципи роботи лазерних далекомірів	8
1.2 Класифікація Lidar-систем та їх особливості.....	12
1.3 Застосування Lidar у різних галузях	14
1.4 Системи позиціонування у мобільній робототехніці.....	17
РОЗДІЛ 2 ВИБІР КОМПОНЕНТІВ СИСТЕМИ	20
2.1 Вибір LiDAR-датчика.....	20
2.2 Вибір мікроконтролера.....	23
2.3 Вибір сервоприводів	26
2.4 Програмні засоби	28
РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ	32
3.1 Загальна блок-схема пристрою.....	32
3.2 Схема з'єднань LiDAR з контролером	33
3.3 Живлення елементів системи	35
3.4 Бібліотеки та драйвери для LiDAR	37
3.5 Збір і обробка даних про відстань	39
3.7 Алгоритми визначення позиції.....	43
3.8 Виведення результатів та збереження даних	44
ВИСНОВКИ.....	47
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	48

ВСТУП

Актуальність теми. У сучасних умовах зростає потреба у системах просторової орієнтації для автономних пристроїв, дронів, мобільних роботів та систем картографування. Одним із ключових технологічних рішень є використання лазерних систем позиціонування (LiDAR), які дозволяють з високою точністю формувати тривимірну модель навколишнього середовища. Завдяки доступності малогабаритних датчиків і відкритих апаратно-програмних платформ, подібні системи можуть бути реалізовані у вигляді доступних освітніх або наукових проєктів.

Метою роботи є створення простої, доступної та функціональної моделі, здатної виконувати вимірювання відстаней до об'єктів та будувати просторове уявлення про навколишнє середовище.

Об'єкт дослідження – система позиціонування.

Предмет дослідження – принципи побудови системи позиціонування на базі Lidar та Arduino.

Завдання, які необхідно виконати:

- провести огляд технологій лазерного вимірювання відстані;
- обґрунтувати вибір апаратних компонентів системи;
- підібрати відповідне програмне забезпечення для керування та збору даних;
- зібрати систему та реалізувати програмну логіку.

РОЗДІЛ 1

ТЕОРЕТИЧНІ ОСНОВИ ЛАЗЕРНОГО ПОЗИЦІОНУВАННЯ

1.1 Принципи роботи лазерних далекомірів

Лазерні системи позиціонування є ключовим компонентом у багатьох галузях техніки, зокрема в робототехніці, автономній навігації, системах автоматизованого контролю, транспорті та безпілотних платформах. Завдяки високій точності та здатності працювати в режимі реального часу, LiDAR (Light Detection and Ranging) активно витісняє традиційні методи визначення просторових координат, такі як ультразвукові чи інфрачервоні сенсори [1].

Лазерні далекоміри використовують лазерне випромінювання для вимірювання відстані до об'єкта. Технологія базується на різних фізичних принципах, серед яких найпоширенішими є методи імпульсний (Time of Flight, ToF), фазовий і триангуляція.

Метод ToF (рис. 1.1) полягає в вимірюванні часу, який лазерний пучок витрачає на подорож від передавача до відбивача і назад. Відстань визначається за формулою (1.1):

$$D = \frac{c \cdot \Delta t}{2}, \quad (1.1)$$

де D – відстань до об'єкта;

c – швидкість світла;

Δt – час проходження імпульсу до об'єкта і назад [2].

Цей метод є основним у багатьох сучасних системах Lidar завдяки своїй високій точності та швидкості вимірювання, що робить його ідеальним для застосувань у мобільних роботах і автономних транспортних засобах. Проте точність таких сенсорів зменшується при роботі з відбиваючими або прозорими поверхнями, а також при сильному зовнішньому освітленні [3].

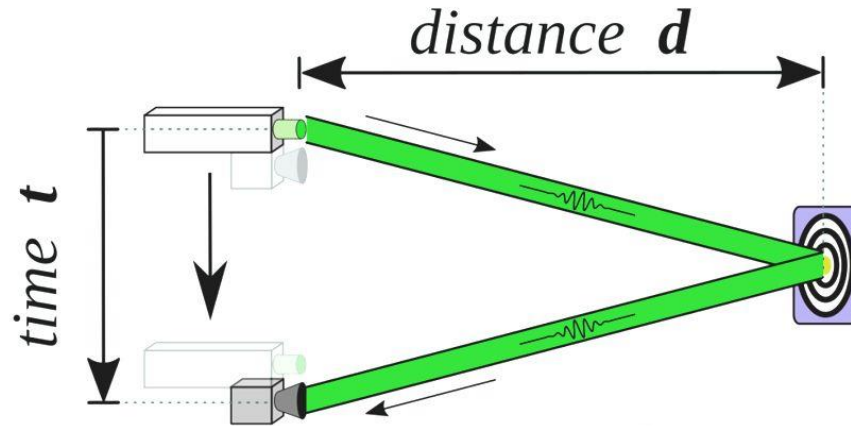


Рисунок 1.1 – Принцип дії лазерного далекоміра за методом ToF [2]

Фазовий метод базується на вимірюванні зміни фази світлової хвилі після її відбиття від об'єкта. Порівнюючи фазу переданого і прийнятого сигналу, можна обчислити відстань до об'єкта, оскільки зміна фази пропорційна шляху, який пройшов лазерний пучок. Відстань визначається за формулою (1.2):

$$D = \frac{\lambda}{4\pi} \Delta\varphi, \quad (1.2)$$

де D – відстань до об'єкта;

λ – довжина хвилі лазерного променя;

$\Delta\varphi$ – різниця фаз між переданим і прийнятим сигналом [4].

Цей метод має високу точність і дозволяє здійснювати вимірювання на значні відстані, але для отримання високої точності необхідна стабільність джерела лазерного випромінювання та точне налаштування апаратури.

Метод триангуляції є геометричним методом, що використовується в основному для вимірювання коротких відстаней. Він полягає в вимірюванні кута між лазерним променем і приймальним пристроєм при фіксованій відстані між передавачем і приймачем. Відстань до об'єкта обчислюється за допомогою формули (1.3):

$$D = \frac{L}{\tan(\theta)}, \quad (1.3)$$

де D – відстань до об'єкта;

L – відстань між лазерним передавачем і приймачем;

θ – виміряний кут [5].

Метод триангуляції часто застосовується в 2D- і 3D-сканерах, де відстань вимірюється за допомогою декількох кутів і параметрів.

Принципи лазерного вимірювання відстані включають різні методи, кожен з яких має свої переваги й обмеження в залежності від специфіки застосування. Метод ToF є найбільш поширеним завдяки своїй швидкості і точності, тоді як фазовий метод використовується для високоточних вимірювань на великих відстанях. Метод триангуляції є ефективним для коротших дистанцій і застосовується у специфічних випадках.

Частотно-модульована безперервна хвиля (Frequency-Modulated Continuous-Wave, FMCW) – інший підхід до визначення відстані, який базується на безперервному випромінюванні лазерного сигналу з лінійно змінною частотою. При відбитті сигналу від об'єкта виникає зсув частоти, який пропорційний до пройденої відстані. Ця технологія дозволяє отримати не лише координати об'єкта, а й інформацію про його швидкість (за допомогою ефекту Доплера) [6].

На рисунку 1.2 подано принцип дії FMCW-датчика. Передавач випромінює безперервну хвилю, частота якої змінюється з часом. Вимірювання різниці частот між переданим і отриманим сигналами дозволяє обчислити відстань.

Серед основних переваг FMCW-технології – висока точність, стійкість до завад, можливість одночасного вимірювання відстані та швидкості. Однак такі системи є складнішими в реалізації та потребують більш дорогих компонентів і складної обробки сигналів [7].

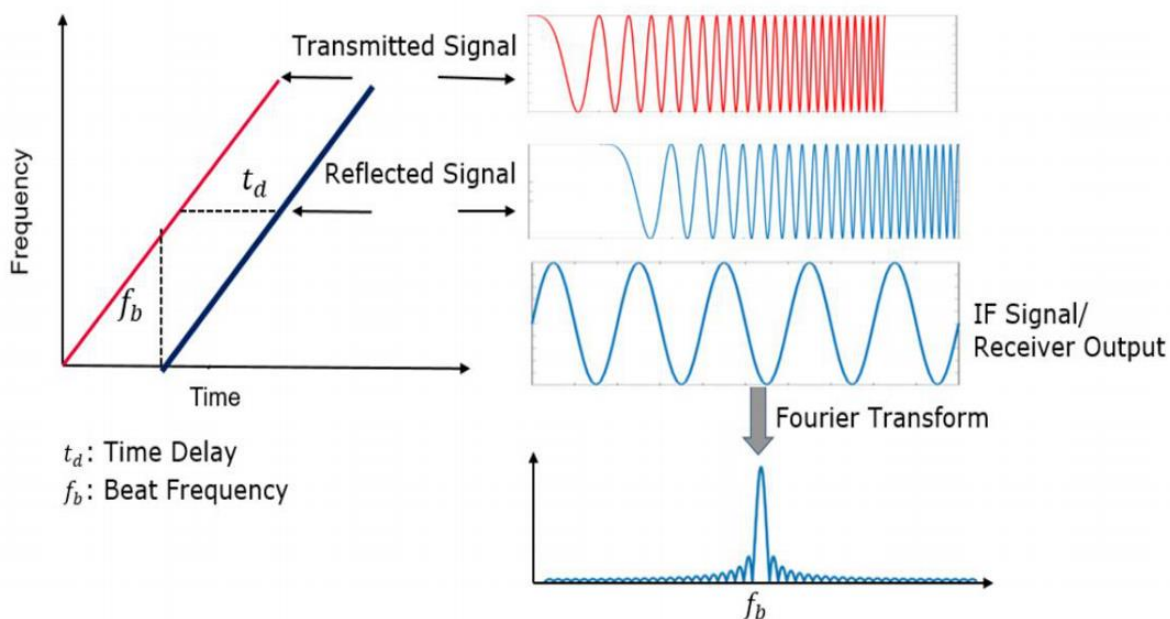


Рисунок 1.2 – Схема принципу роботи FMCW-LiDAR [6]

Порівняння характеристик ToF та FMCW-методів наведена нижче в таблиці 1.1.

Таблиця 1.1 – Порівняння основних характеристик ToF і FMCW-методів [3, 7]

Характеристика	ToF	FMCW
Тип сигналу	Імпульсний	Безперервний з модуляцією
Вимірювана величина	Відстань	Відстань і швидкість
Стійкість до завад	Середня	Висока
Точність	Середня	Висока
Вартість реалізації	Низька	Висока
Складність обробки	Низька	Висока

Таким чином, вибір методу визначення відстані залежить від вимог до точності, швидкості обробки, умов експлуатації та вартості системи. Для бюджетних рішень з базовими вимогами ідеально підходять ToF-сенсори. Натомість у системах високоточного позиціонування, наприклад, у безпілотних автомобілях, доцільно використовувати FMCW-LiDAR-системи.

1.2 Класифікація Lidar-систем та їх особливості

Lidar-системи класифікуються залежно від просторової роздільної здатності, яка визначається кількістю координат, що можуть бути зафіксовані у процесі сканування. У загальному вигляді розрізняють три основні типи: 1D, 2D та 3D Lidar-системи.

Одновимірні Lidar-системи (1D Lidar) – це найпростіший тип оптичних далекомірів, які вимірюють лише відстань до об'єкта вздовж одного променя. Вони не створюють зображень або карт, а лише надають дані про зміну положення об'єкта в одній осі. Такі сенсори часто застосовуються в промисловій автоматизації, для контролю висоти, визначення рівня заповнення контейнерів, а також у деяких дронах для стабілізації висоти [8].

1D Lidar зазвичай компактні, енергоефективні, дешеві у виробництві та обслуговуванні. Основним обмеженням є відсутність просторової інформації, що робить їх непридатними для складних навігаційних задач.

Двовимірні системи (2D Lidar) мають можливість сканувати в площині – зазвичай горизонтальній, що дозволяє створювати контурні мапи простору. Такі Lidar активно застосовуються в мобільній робототехніці, логістичних системах, а також в системах забезпечення безпеки на виробництві [9].

Принцип дії 2D Lidar полягає у швидкому обертанні або коливанні лазерного променя, що дозволяє фіксувати положення об'єктів у площині. Зазвичай ці пристрої мають високу частоту оновлення даних і забезпечують хорошу точність у межах кількох десятків метрів. Проте, незважаючи на двовимірність, 2D Lidar не може розпізнавати об'єкти, розташовані поза площиною сканування, що може створювати «сліпі зони».

Тривимірні системи (3D Lidar) дозволяють отримувати повноцінні тривимірні моделі навколишнього середовища. Вони використовують або кілька скануючих площин, або поворотні блоки, які охоплюють як горизонтальну, так і вертикальну площини, або багатоканальні лазери. Такі

Lidar-системи є ключовими в технологіях автономного транспорту, топографічного моделювання, археології та інспекції інфраструктури [10].

3D Lidar забезпечують максимальну інформативність, однак мають більшу складність конструкції, вищу вартість та потребують обчислювальних потужностей для обробки великих обсягів даних. У багатьох випадках вони поєднуються із системами машинного зору, GPS та інерційними навігаційними системами.

У таблиці 1.2 представлено порівняльну характеристику Lidar-систем за кількістю вимірюваних координат.

Іншим важливим критерієм класифікації є мобільність системи – її здатність до переміщення або інтеграції у мобільні платформи. Згідно з цим критерієм, Lidar-системи поділяють на стаціонарні, мобільні та портативні.

Таблиця 1.2 – Порівняльні характеристики Lidar-систем за кількістю координат [8-10]

Тип Lidar	Кількість координат	Призначення	Дальність, м	Переваги	Недоліки
1D	1 (відстань)	Промисловість, висотоміри	до 50	Простота, низька вартість	Немає просторової інформації
2D	2 (X-Y)	Робототехніка, логістика	до 100	Сканування площини	Обмеження по висоті
3D	3 (X-Y-Z)	Автомобілі, геодезія, дрони	до 300	Повна 3D-карта простору	Висока вартість, великі обсяги даних

1. Стаціонарні Lidar-системи фіксовано встановлюються на опорах або будівлях і призначені для детального сканування певної ділянки. Такі системи використовуються в архітектурній реставрації, інженерній геодезії, картографії, а також в охоронних системах для виявлення руху на об'єкті [11].

Вони відзначаються високою точністю, стабільністю вимірювань та широким кутом огляду, але обмежені нерухомістю – не здатні самостійно змінювати місце розташування.

2. Мобільні Lidar-системи встановлюються на транспортних засобах – автомобілях, літаках, безпілотниках, роботах. Вони забезпечують можливість сканування великих площ у динамічному режимі. Це ключовий тип Lidar для автономних автомобілів, а також для повітряного лазерного сканування територій [12].

Мобільні системи поєднуються з GPS, інерційними модулями, камерами для забезпечення просторової прив'язки даних. Основними перевагами є охоплення великих територій, адаптивність, економія часу. Недоліки включають складність синхронізації та високу вартість обладнання.

3. Портативні системи – це компактні пристрої, призначені для ручного або наплічного використання. Їх застосовують у складських приміщеннях, музеях, археологічних експедиціях. Пристрої можуть мати вбудовану обробку даних, що дозволяє отримувати 3D-модель в реальному часі [13].

Ці Lidar забезпечують гнучкість, доступ до важкодоступних місць, але поступаються за точністю та дальністю вимірюванням більшим мобільним або стаціонарним системам.

Класифікація Lidar-систем за кількістю координат та мобільністю має ключове значення для правильного вибору технології залежно від завдань. 1D-системи використовуються для простих завдань, 2D – у робототехніці та безпеці, 3D – у складному картографуванні та автономному управлінні. За мобільністю виділяють стаціонарні Lidar для детальних точкових досліджень, мобільні – для широкомасштабних динамічних сканувань, та портативні – для швидких локальних обстежень. Надалі розвиток Lidar-технологій пов'язаний з інтеграцією їх у смарт-системи та розширенням можливостей завдяки штучному інтелекту.

1.3 Застосування Lidar у різних галузях

Завдяки високій просторовій точності, здатності формувати детальні тривимірні карти та працювати в умовах недостатнього освітлення, Lidar-

системи набули широкого поширення в багатьох галузях зокрема: автономний транспорт, безпілотні літальні апарати (дрони), мобільна робототехніка, геодезія та системи безпеки.

Однією з найважливіших сфер застосування Lidar є автомобілебудування, зокрема – розробка автономних транспортних засобів. Lidar дозволяє створювати детальні 3D-моделі навколишнього середовища в режимі реального часу. Це дає змогу транспортному засобу точно розпізнавати об'єкти, відстані до них, дорожню розмітку, бордюри та пішоходів [14].

Такі компанії, як Waymo, Cruise та Aurora, активно інтегрують Lidar у свої системи автопілота. Зокрема, Lidar дозволяє уникати зіткнень завдяки побудові точних карт перешкод навіть в умовах туману, сутінків чи засвічення фарами інших автомобілів. Варто зазначити, що Lidar у цьому випадку часто працює у поєднанні з радарями, камерами та ультразвуковими сенсорами для досягнення максимальної надійності.

Безпілотні літальні апарати (дрони) з Lidar-сенсорами дозволяють отримувати високоточні геопросторові дані з повітря. Завдяки здатності проникати крізь рослинність, Lidar-дрони активно використовуються для:

- моніторингу лісових масивів;
- аналізу змін рельєфу;
- оцінки біомаси;
- побудови цифрових моделей місцевості (ЦММ) [15].

Особливо цінною є здатність Lidar формувати тривимірну «хмару точок», яка забезпечує точність до кількох сантиметрів. Це робить Lidar-дрони незамінними у геодезії, будівництві та агрономії. Компанії DJI, Yellowscan та RIEGL пропонують комерційні рішення, які вже використовуються в понад 70 країнах світу [16].

У сфері робототехніки Lidar є ключовим сенсором для автономної навігації, картографування та уникнення перешкод. Наприклад, роботи-прибиральники (як-от iRobot Roomba) або складські роботи Amazon використовують 2D або 3D Lidar для орієнтації у приміщеннях, побудови мап

методом SLAM (Simultaneous Localization and Mapping) та оптимізації маршрутів [17].

Перевагою Lidar є його здатність працювати без залежності від освітлення, що робить його кращим за камери у ряді задач. Завдяки цьому, Lidar дозволяє роботам безпечно пересуватись навіть у повній темряві або у задимлених приміщеннях.

Геодезія є однією з перших галузей, де Lidar-системи отримали практичне застосування. Завдяки високій точності та швидкості збору даних, Lidar дозволяє створювати детальні цифрові моделі рельєфу (ЦМР), моделі поверхні (ЦМП) та тривимірні карти територій. Основні застосування:

- топографічна зйомка;
- археологічне зондування;
- інженерні вишукування;
- моніторинг зсувів ґрунту [18].

У геодезії використовуються як повітряні Lidar (на літаках і дронах), так і наземні системи. Такі рішення дозволяють охоплювати великі площі в короткий термін із точністю до 2-5 см. Окрім того, Lidar здатен фіксувати об'єкти, які приховані лісовим покривом, що важко досягти іншими методами зйомки.

Lidar активно впроваджується у системи безпеки для охорони периметрів, виявлення вторгнень і відстеження руху в режимі реального часу. Завдяки точному виявленню об'єктів за координатами, Lidar можна використовувати:

- у військовій сфері для розпізнавання техніки та людей;
- на промислових об'єктах для охорони територій;
- у «розумних містах» для моніторингу натовпу або транспортних потоків [19].

Порівняно з камерами відеоспостереження, Lidar забезпечує більшу точність в умовах низької освітленості, а також не залежить від кольору або текстури об'єктів.

У таблиці 1.3 наведено стислий огляд ключових галузей використання Lidar та основних характеристик у кожній з них.

Таблиця 1.3 – Застосування Lidar у різних галузях [14-19]

Галузь	Основне застосування	Тип Lidar	Особливості використання
Автомобілі	Автономне водіння, навігація	3D ToF / FMCW	Висока роздільна здатність, виявлення руху
Дрони	Аерозйомка, моніторинг, ЦММ	Повітряний ToF	Можливість зондування через лісовий покрив
Робототехніка	Локалізація, SLAM, уникнення перешкод	2D / 3D ToF	Робота в реальному часі в приміщеннях
Геодезія	Топографія, археологія, ЦМР	Наземний / повітряний фазовий / ToF	Надвисока точність, покриття великих площ
Безпека	Охорона, спостереження, моніторинг	3D ToF / FMCW	Робота при поганому освітленні, реальний час

Lidar-системи є універсальним інструментом для збору просторових даних і виявлення об'єктів, що забезпечує їх активне впровадження у найрізноманітніших галузях. Від автономного транспорту до археологічних досліджень – Lidar демонструє здатність забезпечувати високу точність, надійність та ефективність навіть в екстремальних умовах. Очікується, що з розвитком технологій, здешевленням компонентів та вдосконаленням обробки даних застосування Lidar тільки зростатиме, охоплюючи нові сфери – від медицини до космічних місій.

1.4 Системи позиціонування у мобільній робототехніці

Системи позиціонування є ключовим елементом мобільної робототехніки, оскільки забезпечують здатність робота визначати власне положення у просторі, орієнтуватися в середовищі та планувати рух. Залежно від завдань та умов експлуатації, використовуються різні підходи до реалізації позиціонування: інерціальні, супутникові, візуальні та лазерні методи [20, 21]. У цьому підрозділі розглянуто особливості лазерного позиціонування на базі

LiDAR, його інтеграцію з мікроконтролерами (зокрема Arduino), а також порівняно з альтернативними технологіями.

Системи позиціонування в мобільних роботах зазвичай поділяють на такі категорії:

1) абсолютне позиціонування – визначає координати об'єкта у глобальній системі відліку (наприклад, GPS) [21];

2) відносне (локальне) позиціонування – базується на зміні положення відносно попередньої точки (наприклад, за допомогою енкодерів або IMU) [22];

3) сенсорне позиціонування – використовує дані від датчиків оточення, таких як камери чи LiDAR, для побудови карти середовища і визначення положення [20].

LiDAR-сенсори забезпечують високу точність вимірювання відстаней до об'єктів і дають змогу будувати детальні карти навколишнього середовища. Вони особливо ефективні у задачах, де важлива просторово-часова роздільна здатність та незалежність від зовнішнього освітлення [21, 22].

На рисунку 1.3 наведено основні застосування LiDAR у системі навігації мобільного робота.

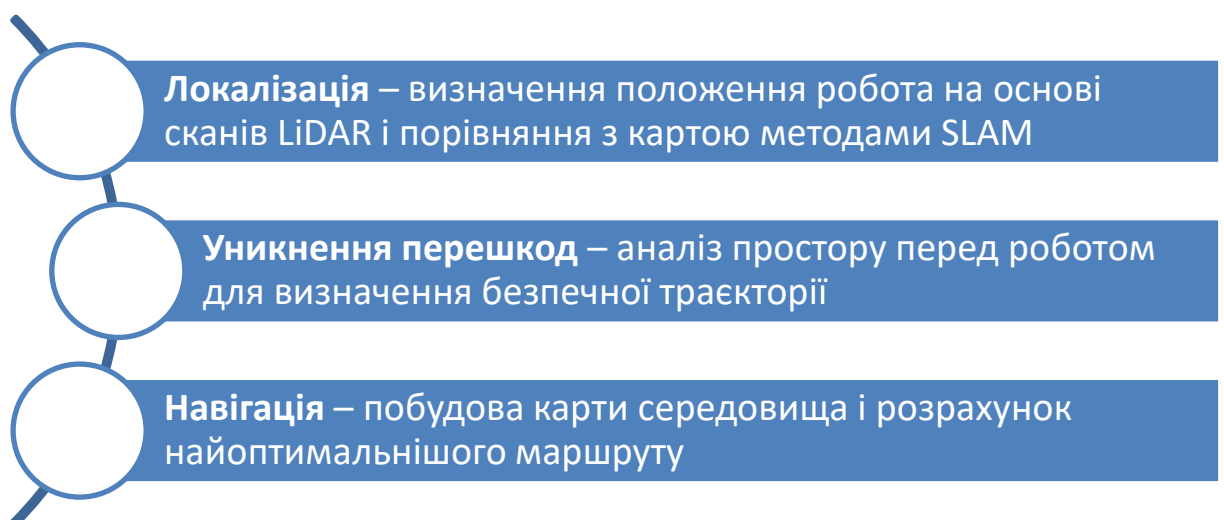


Рисунок 1.3 – Основні застосування LiDAR у робототехніці [20-21]

У цьому підрозділі розглянуто ключові методи позиціонування мобільних роботів, із фокусом на використанні LiDAR-технологій. Визначено, що LiDAR забезпечує високу точність та надійність у задачах навігації, особливо в комбінації зі SLAM-алгоритмами [20].

РОЗДІЛ 2

ВИБІР КОМПОНЕНТІВ СИСТЕМИ

2.1 Вибір LiDAR-датчика

Для реалізації системи лазерного позиціонування на базі Arduino важливо правильно обрати лазерний далекомір (LiDAR), який відповідатиме технічним та економічним вимогам проекту. До основних критеріїв вибору відносяться: точність вимірювання, дальність дії, швидкість оновлення даних, габарити, енергоспоживання, сумісність з платформою Arduino, наявність документації та прикладів реалізацій, а також вартість пристрою.

Серед поширених на ринку моделей компактних LiDAR-сенсорів, що підтримуються Arduino-спільнотою, найбільш придатними для використання в малогабаритних проектах є:

TFMini (рис. 2.1) – компактний одноточковий LiDAR, розроблений компанією Venewake, що працює за принципом time-of-flight (ToF). Пристрій забезпечує дальність вимірювання до 12 м на відкритому просторі зі швидкістю оновлення до 100 Гц. Підтримує UART та PWM-інтерфейси. Його основними перевагами є висока дальність вимірювання та стабільність сигналу при зовнішньому освітленні. Проте серед недоліків – більші габарити та вища вартість, що не завжди доцільно для бюджетних проектів [23].



Рисунок 2.1 – LiDAR-датчик TFMini [23]

VL53L0X (рис. 2.2) – лазерний далекомір від компанії STMicroelectronics, який використовує технологію ToF з вимірюванням часу проходження світлового імпульсу. Має надзвичайно малі габарити (4.4×2.4×1.0 мм), що дозволяє інтегрувати його в компактні мобільні системи. Дальність дії становить до 2 м у режимі «Long», точність вимірювання – до $\pm 3\%$. Датчик підтримує інтерфейс I²C, що забезпечує простоту підключення до Arduino. Наявна велика кількість бібліотек та прикладів реалізації з відкритим кодом [24].



Рисунок 2.2 – LiDAR-датчик VL53L0X [24]

VL53L1X – покращена версія VL53L0X із розширеним діапазоном вимірювання до 4 м, змінним полем зору (FoV) та вищою швидкістю вимірювання. Проте ця модель має більші енергоспоживання і складнішу конфігурацію, що може вимагати додаткових налаштувань при роботі з мікроконтролером Arduino [25].

Lidar Lite v3 – високоточний сенсор дальності від Garmin, що забезпечує дальність до 40 м із високою точністю (до ± 2.5 см). Проте цей пристрій є досить дорогим і потребує стабільного живлення на рівні 5 В з великим струмом. Також він має більші габарити та складнішу реалізацію в порівнянні з мініатюрними сенсорами, такими як VL53L0X [26].

HC-SR04 – ультразвуковий датчик, який часто використовується в недорогих робототехнічних проєктах. Незважаючи на широке використання, він не є LiDAR-сенсором, і його точність значно нижча у порівнянні з лазерними далекомірами. Додатково, ультразвукові датчики мають обмеження при роботі з м'якими або кутовими поверхнями та схильні до впливу шумів [27].

У таблиці 2.1 наведено порівняння ключових параметрів розглянутих LiDAR-модулів.

Таблиця 2.1 – Порівняння технічних характеристик LiDAR-датчиків [23-27]

Назва датчика	Дальність вимірювання	Точність	Інтерфейс	Розміри (мм)	Вартість (USD)	Сумісність з Arduino
VL53L0X	до 200 см	±3%	I ² C	4.4×2.4×1.0	~5-7	Висока
VL53L1X	до 400 см	±2%	I ² C	4.9×2.5×1.6	~8-10	Висока
TFMini	до 1200 см	±6 см	UART, PWM	42×15×16	~20-30	Висока
Lidar Lite v3	до 4000 см	±2.5 см	PWM, I ² C	48×20×40	~100-120	Висока
HC-SR04	до 400 см	±1 см	Trigger/Echo	45×20×15	~2	Висока (не LiDAR)

Беручи до уваги цілі створення системи лазерного позиціонування на базі Arduino, доцільним є використання сенсора VL53L0X. Основні переваги цього модуля полягають у його компактності, низькому енергоспоживанні та простоті інтеграції. Пристрій ідеально підходить для роботи в умовах обмеженого простору, як-от на мобільних платформах, де важлива кожна частка ваги та об'єму. Датчик має вбудований лазер класу 1 (безпечний для очей) та забезпечує досить точні результати для середовища робототехніки, де вимірювання в діапазоні до 2 м є цілком прийнятним.

Ще однією перевагою є активна підтримка спільноти розробників. Для VL53L0X доступні офіційні бібліотеки для Arduino, що значно спрощує розробку програмного забезпечення. Додатково, відкритість протоколу I²C дає

змогу використовувати цей сенсор разом із кількома іншими пристроями на одній шині.

На основі порівняльного аналізу технічних характеристик, вартості, габаритів та сумісності з Arduino-платформою, для реалізації системи лазерного позиціонування обрано сенсор VL53L0X. Цей модуль поєднує в собі достатню точність, малі розміри, простоту програмування та високу енергоефективність, що робить його оптимальним вибором для реалізації компактних мобільних систем.

2.2 Вибір мікроконтролера

Мікроконтролер є центральним елементом будь-якої вбудованої системи, що відповідає за обробку вхідних сигналів, керування периферійними пристроями, взаємодію з сенсорами та передачу даних. У системах лазерного позиціонування, побудованих на базі LiDAR, мікроконтролер відіграє ключову роль у зборі та попередній обробці даних з далекомірного сенсора, а також у забезпеченні координації між обчислювальними, комунікаційними та виконавчими модулями – сервоприводами.

Вибір мікроконтролера у контексті створення системи лазерного позиціонування ґрунтується на комплексному аналізі низки ключових технічних параметрів. Передусім береться до уваги кількість цифрових та аналогових входів і виходів, що є визначальним чинником при підключенні кількох периферійних пристроїв, зокрема лазерного далекоміра, сервоприводів, дисплеїв та допоміжних сенсорів. Важливою є й конфігурація пам'яті – як обсяг оперативної (SRAM), так і програмної (flash) – оскільки вони безпосередньо впливають на здатність мікроконтролера стабільно працювати з об'ємними бібліотеками та опрацьовувати великі масиви даних, що надходять від сенсорних елементів.

Крім того, до важливих критеріїв належить кількість та тип підтримуваних комунікаційних інтерфейсів, зокрема таких як I²C, UART і SPI,

які забезпечують ефективну взаємодію між мікроконтролером та периферійними модулями. Значущою характеристикою виступає також тактова частота, оскільки вона визначає загальну швидкість системи, що особливо критично для задач, що виконуються у реальному часі. Додатково враховується сумісність з програмним середовищем Arduino IDE, яке є стандартом для розробки прототипів на основі відкритої апаратної платформи. Нарешті, конструктивні особливості, включаючи габаритні розміри, енергоспоживання та загальну вартість, також відіграють роль у виборі, особливо в контексті розробки мобільної або автономної системи.

Серед Arduino-сумісних мікроконтролерів, що активно використовуються у системах керування та позиціонування, доцільно розглянути наступні моделі.

Arduino Uno R3 – найпоширеніша плата на основі мікроконтролера ATmega328P, має 14 цифрових входів/виходів (з них 6 з підтримкою ШІМ), 6 аналогових входів, 32 КБ флеш-пам'яті, 2 КБ SRAM та 1 КБ EEPROM. Uno підтримує інтерфейси I²C, UART та SPI. Основними перевагами є низька вартість і висока поширеність серед початківців. Проте обмеженість пам'яті та кількості портів ускладнює підключення кількох сенсорів та сервоприводів одночасно [28].

Arduino Nano – мініатюрна плата, що базується на тому ж мікроконтролері ATmega328P. Має схожі характеристики з Uno, але компактніші розміри. Незважаючи на зручність у вбудованих конструкціях, Nano також страждає від обмеженого обсягу пам'яті та портів [29].

Arduino Leonardo – заснований на ATmega32U4, має 20 цифрових портів (з них 7 ШІМ), 12 аналогових входів, 32 КБ флеш-пам'яті та можливість працювати як USB-пристрій без додаткового чипа. Підтримує UART, SPI та I²C. Підходить для HID-проектів, однак його архітектура іноді викликає сумісні проблеми з деякими бібліотеками Arduino [30].

ESP32 DevKit – 32-бітна плата з мікроконтролером Tensilica LX6, має 520 КБ SRAM, до 4 МБ флеш-пам'яті, вбудовані Wi-Fi та Bluetooth, до 34 GPIO. Тактова частота досягає 240 МГц. Перевагою є висока продуктивність і

бездротова комунікація. Однак ESP32 вимагає глибших знань програмування, а також сумісність із деякими Arduino-бібліотеками потребує адаптації [31].

Arduino Mega 2560 – плата на основі ATmega2560, має 54 цифрові входи/виходи (15 з підтримкою ШІМ), 16 аналогових входів, 256 КБ флеш-пам'яті, 8 КБ SRAM і 4 КБ EEPROM. Оснащена чотирма апаратними UART, що є великою перевагою для одночасного підключення кількох модулів (наприклад, LiDAR, Bluetooth, GPS). Підтримує інтерфейси I²C, SPI та сумісна з усіма стандартними бібліотеками Arduino IDE [32].

У таблиці 2.2 наведено порівняння технічних характеристик наведених вище мікроконтролерів Arduino.

Таблиця 2.2 – Порівняння технічних характеристик мікроконтролерів Arduino [28-32]

Назва плати	Пам'ять Flash / SRAM / EEPROM	К-ть I/O	UART	I ² C	SPI	Аналогові входи	ШІМ-виходи	Ціна (USD)
Arduino Uno R3	32 КБ / 2 КБ / 1 КБ	14	1	1	1	6	6	~10
Arduino Nano	32 КБ / 2 КБ / 1 КБ	22	1	1	1	8	6	~8
Arduino Leonardo	32 КБ / 2.5 КБ / 1 КБ	20	1	1	1	12	7	~12
ESP32 DevKit	до 4 МБ / 520 КБ / -	до 34	3	1	4	до 18	до 16	~6-10
Arduino Mega 2560	256 КБ / 8 КБ / 4 КБ	70	4	1	1	16	15	~15

На основі порівняльного аналізу, оптимальним варіантом для реалізації системи лазерного позиціонування є Arduino Mega 2560. Насамперед, технічні характеристики дозволяють уникнути обмежень, пов'язаних з нестачею пам'яті або портів, що є типовою проблемою для плат Arduino Uno або Nano. Наявність 54 цифрових портів, серед яких 4 апаратні UART, дозволяє підключати одночасно LiDAR-модуль, кілька сервоприводів, модуль Bluetooth або Wi-Fi, дисплей, енкодери тощо без потреби в додаткових мультиплексах чи порт-експандерах.

Також важливою перевагою є наявність 16 аналогових входів, що відкриває можливості для інтеграції аналогових сенсорів (наприклад, ІЧ-дальномірів або потенціометрів), які можуть доповнювати інформацію, отриману від LiDAR. Обсяг флеш-пам'яті в 256 КБ та 8 КБ SRAM забезпечує стабільне виконання програм навіть за використання об'ємних бібліотек для керування графікою, сервоприводами або алгоритмами навігації.

Arduino Mega 2560 має широку підтримку серед розробників, велику кількість прикладів і готових бібліотек для роботи з сенсорами VL53L0X, сервоприводами та засобами комунікації.

Ураховуючи технічні вимоги до побудови системи лазерного позиціонування, зокрема кількість периферійних пристроїв, потребу в кількох апаратних UART, обсяги пам'яті та програмну сумісність, було обрано платформу Arduino Mega 2560. Вона забезпечує необхідну обчислювальну потужність, функціональність та гнучкість для реалізації складних алгоритмів позиціонування та керування компонентами системи.

2.3 Вибір сервоприводів

У системах лазерного позиціонування на базі мікроконтролерів та LiDAR-сенсорів механічна орієнтація датчика у просторі відіграє важливу роль. Для реалізації повороту або сканування в горизонтальній та вертикальній площинах широко застосовуються сервоприводи – компактні електромеханічні пристрої, здатні точно контролювати кутове положення вала. У даному проєкті сервоприводи використовуються для обертання лазерного далекоміра з метою забезпечення можливості сканування простору та формування карти оточення. Вибір конкретної моделі сервопривода ґрунтується на аналізі низки технічних параметрів, таких як кут повороту, крутний момент, робоча напруга, швидкість обертання, сумісність із мікроконтролером, а також надійність і ціна.

У процесі дослідження було розглянуто кілька популярних моделей сервоприводів, які активно застосовуються у робототехнічних та

автоматизованих системах: SG90, MG90S, MG996R та DS3218. Кожна з цих моделей має свої переваги та обмеження, що зумовлює доцільність їх використання в залежності від вимог конкретного проєкту.

Модель SG90 є надзвичайно популярною завдяки своїй низькій вартості та компактним розмірам. Проте її крутного моменту (близько 1.8 кг·см при 6 В) недостатньо для стабільного керування масивнішими датчиками або для забезпечення точного обертання за умов підвищеного навантаження [33]. MG90S, яка є металевим аналогом SG90, має дещо кращі показники потужності, однак також не забезпечує достатнього крутного моменту, необхідного для обертання LiDAR-сенсора зі стабільною швидкістю.

У свою чергу, сервопривод DS3218 має високий крутний момент – до 20 кг·см, металевий корпус, що робить його доречним для застосування у великих роботизованих системах. Однак такий пристрій є значно важчим, дорожчим, та споживає більше енергії, що є недоліком у контексті розробки портативної та економної системи лазерного позиціонування [34].

Найбільш оптимальним у цьому контексті виявився сервопривод MG996R. Цей пристрій забезпечує крутний момент – 9.4 кг·см при 6 В, що є більш ніж достатнім для ефективного обертання лазерного сенсора навіть за наявності додаткового механічного навантаження. При цьому він зберігає прийнятні габарити та масу, дозволяє працювати з широким діапазоном напруг (4.8-7.2 В), має металеві шестерні, що забезпечує довговічність та високу механічну стійкість [35]. Швидкість обертання також є задовільною – близько 0.17 с/60° при 6 В, що дозволяє реалізовувати швидке сканування навколишнього простору. Крім того, цей сервопривод підтримується великою кількістю бібліотек Arduino, що значно полегшує програмну реалізацію керування ним [36].

У таблиці 2.3 наведено порівняльні характеристики чотирьох розглянутих моделей сервоприводів. На основі аналізу представлених даних було зроблено висновок, що сервопривод MG996R найкраще відповідає технічним вимогам розроблюваної системи. Його застосування дозволяє забезпечити надійне та

стабільне позиціонування LiDAR-сенсора, що є критичним для точності вимірювань та формування просторової карти.

Таблиця 2.3 – Порівняльна характеристика сервоприводів для системи позиціонування [33-35]

Модель	Крутний момент (кг·см)	Напруга живлення (В)	Швидкість обертання (с/60°)	Маса (г)	Тип шестерень
SG90	1.8 (при 6 В)	4.8-6.0	0.12	9	Пластикові
MG90S	2.2 (при 6 В)	4.8-6.0	0.11	13	Металеві
MG996R	9.4 (при 6 В)	4.8-7.2	0.17	55	Металеві
DS3218	20.0 (при 6.8 В)	4.8-6.8	0.16	60	Металеві

Слід зазначити, що надійність механічного компонента, яким є сервопривод, значною мірою впливає на тривалість безперебійної роботи всієї системи. З урахуванням режимів експлуатації (безперервне сканування, робота в динамічному середовищі, вплив вібрацій) використання моделі MG996R також доцільне з огляду на її витривалість. За даними користувацьких оглядів, пристрій добре зарекомендував себе у практичному застосуванні в аматорських та напівпрофесійних робототехнічних системах [37].

Крім того, значним фактором у виборі стала вартість сервопривода. На ринку модель MG996R пропонується за відносно невисоку ціну, що забезпечує хороше співвідношення «ціна-якість» для проектів з обмеженим бюджетом.

Таким чином, сервопривод MG996R є технічно доцільним та економічно обґрунтованим вибором для реалізації поворотного механізму у системі лазерного позиціонування. Його застосування дозволяє задовольнити основні вимоги до точності, надійності та швидкодії сканування, що є ключовими для ефективної роботи LiDAR-сенсора у складі розроблюваного пристрою.

2.4 Програмні засоби

Розробка системи лазерного позиціонування передбачає застосування відповідного програмного забезпечення для написання, компіляції, налагодження та візуалізації коду, що забезпечує взаємодію між апаратною

частиною (мікроконтролером, датчиком та сервоприводами) та програмною логікою. Ефективність розробки проекту значною мірою залежить від правильного вибору інтегрованого середовища розробки (IDE), яке має забезпечувати зручний інтерфейс, підтримку відповідних мов програмування, сумісність з апаратною платформою, наявність інструментів для налагодження та гнучкість у розширенні функціональності.

Arduino IDE є офіційним середовищем розробки для платформ Arduino, яке підтримує програмування мовою C/C++ з розширенням Arduino API. Завдяки своїй простоті, відкритому коду та великій спільноті користувачів, це середовище широко застосовується у навчальних, наукових та хобі-проектах. Серед основних функціональних можливостей Arduino IDE можна виділити автоматичну компіляцію, монітор послідовного порту для обміну даними між мікроконтролером і ПК, а також підтримку численних бібліотек, включаючи бібліотеки для роботи з LiDAR-датчиками, сервоприводами, інтерфейсами I²C, SPI та UART [38].

Arduino IDE не вимагає складного встановлення або додаткових налаштувань, що робить його ідеальним вибором для початкової фази розробки. Водночас він дозволяє реалізовувати складні алгоритми обробки даних, таймери, програмну логіку контролю положення, а також реалізувати протоколи обміну з візуалізаційними інструментами на ПК. Значною перевагою є можливість використання великої кількості готових бібліотек і прикладів, які спрощують інтеграцію компонентів у єдину систему. Завдяки активній спільноті та розвиненій документації, Arduino IDE дозволяє швидко адаптувати нові апаратні модулі без глибоких знань мікроелектроніки [39].

MATLAB традиційно використовується для наукових досліджень завдяки своїм потужним інструментам візуалізації та математичному апарату. Спеціальні функції MATLAB дозволяють будувати карти оточення, аналізувати розподіл точок, а також моделювати сигнали, отримані з LiDAR-сенсорів [40]. Вбудована підтримка 3D-візуалізації та функцій машинного навчання робить

його зручним для дослідницьких задач, проте вартість ліцензії обмежує його використання у студентських та DIY-проєктах.

NetBeans IDE, своєю чергою, є інтегрованим середовищем розробки, орієнтованим насамперед на мову програмування Java. Це середовище має потужні засоби для розробки графічних інтерфейсів, обробки вхідних даних, мережевої взаємодії та візуалізації. Зокрема, Java-програма, написана в середовищі NetBeans, може приймати дані від Arduino через послідовний порт, обробляти ці дані, зберігати їх у форматах CSV або JSON, будувати графіки та реалізовувати інтуїтивно зрозумілий GUI [41].

NetBeans дозволяє побудувати окремий застосунок на Java, що функціонуватиме як центр візуалізації даних з лазерного далекоміра. Це дає змогу зручно спостерігати за зміною відстаней у реальному часі, аналізувати результати вимірювань або будувати схематичне зображення сканованого простору. Також середовище підтримує розширення за допомогою плагінів, багатовіконний режим, можливість підключення до баз даних і багатопотокову обробку, що відкриває перспективи подальшої модернізації системи, наприклад, для автоматизованого керування об'єктами на основі просторового аналізу [42].

PlatformIO хоча й пропонує розширені функції налагодження та підтримку великої кількості платформ, потребує складнішого налаштування та інтеграції з Visual Studio Code. Atmel Studio більше орієнтований на AVR-програмування низького рівня, що ускладнює роботу з Arduino-сумісними бібліотеками. Середовище Processing, хоч і підходить для візуалізації, значно поступається за гнучкістю Java у створенні повноцінних застосунків [43].

У процесі аналізу було розглянуто декілька програмних платформ, зокрема Arduino IDE, PlatformIO, Atmel Studio, Eclipse, а також середовища, орієнтовані на візуалізацію даних, зокрема Processing, NetBeans Java IDE та Python із бібліотеками Matplotlib і PySerial. Кожен із цих інструментів має свої переваги у певних аспектах, проте з огляду на вимоги до сумісності, простоти

налагодження та доступності, оптимальними для реалізації системи виявилися Arduino IDE та NetBeans Java IDE.

Таким чином, поєднання двох програмних середовищ – Arduino IDE і NetBeans – дозволяє ефективно реалізувати програмну архітектуру системи лазерного позиціонування. Їхня сумісність, доступність, зручність та широкі можливості для масштабування дозволяють не лише вирішити актуальні задачі, а й закласти основу для подальшого розвитку та інтеграції нових функціональних модулів.

РОЗДІЛ 3

ПРАКТИЧНА РЕАЛІЗАЦІЯ

3.1 Загальна блок-схема пристрою

Загальна блок-схема системи лазерного позиціонування, що представлена на рисунку 3.1, демонструє основні функціональні модулі проекту та логіку їх взаємодії. Основною ідеєю реалізації такої системи є поєднання апаратної частини, що виконує безпосередній збір і передавання просторових даних, та програмного забезпечення, яке відповідає за їх подальшу обробку та візуалізацію у тривимірному просторі. Система має чітку модульну структуру, в якій кожен елемент виконує визначену роль і пов'язаний із суміжними компонентами за допомогою фізичних або логічних інтерфейсів.

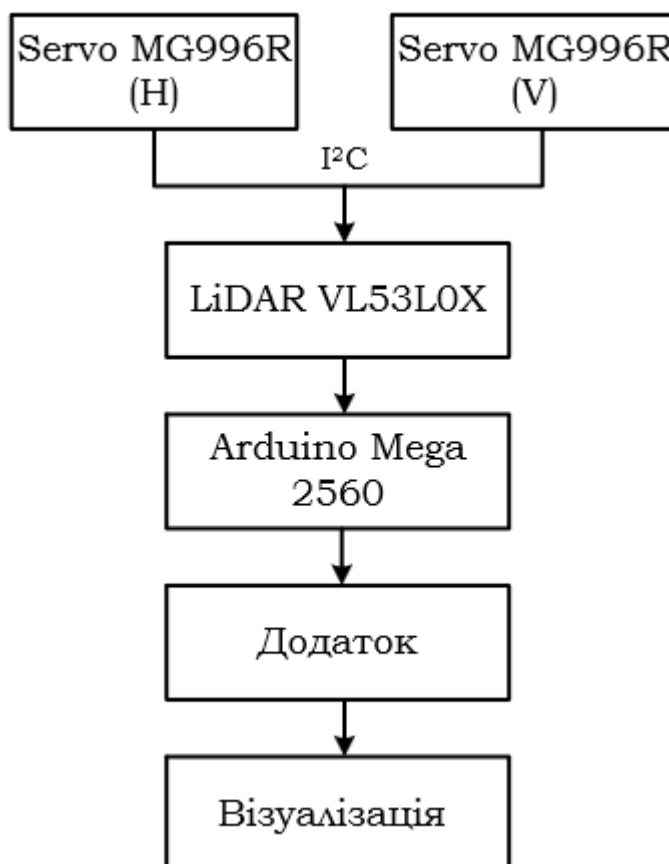


Рисунок 3.1 – Загальна блок-схема системи лазерного позиціонування

Наступним ключовим елементом системи є лазерний далекомір VL53L0X, реалізований у вигляді модуля GY-530. Цей датчик функціонує на основі принципу вимірювання часу проходження світлового імпульсу до об'єкта та назад (Time-of-Flight, ToF). VL53L0X підключається до Arduino Mega 2560 через інтерфейс I²C, що дозволяє забезпечити ефективний обмін даними з низьким енергоспоживанням. Однією з переваг цього сенсора є його компактність, висока точність на малих відстанях (до 2 метрів) та можливість швидкого оновлення даних, що є критичним у задачах позиціонування.

Мікроконтролер Arduino Mega 2560 виступає центральним вузлом апаратної частини системи. Він координує всі операції між виконавчими механізмами (серводвигунами), сенсорними пристроями (LiDAR) та забезпечує передачу зібраної інформації до комп'ютерної частини системи. Завдяки великій кількості цифрових входів/виходів, розширеному обсягу оперативної пам'яті та підтримці кількох послідовних портів Arduino Mega є оптимальним вибором для реалізації такого роду проєктів. Комунікація між Arduino та зовнішнім комп'ютером відбувається через послідовний інтерфейс UART із використанням вбудованого USB-конвертера, що спрощує процес налагодження та зменшує потребу у додатковому обладнанні.

Таким чином, загальна блок-схема пристрою включає взаємозв'язок між сенсором, сервоприводами, контролером і комп'ютерною системою з ПЗ вищого рівня. Цей підхід дозволяє реалізувати повноцінне тривимірне сканування та аналіз навколишнього середовища в реальному часі.

3.2 Схема з'єднань LiDAR з контролером

Після формування загальної структурної блок-схеми системи було реалізовано фізичне з'єднання (рис. 3.2) мікроконтролера Arduino Mega 2560 із сенсором відстані GY-530, що базується на чипі VL53L0X. Цей модуль належить до типу Time-of-Flight (ToF) лазерних сенсорів і підтримує обмін даними за інтерфейсом I²C, що суттєво спрощує підключення до Arduino. У

процесі збирання апаратної частини було з'єднано контакти SDA і SCL датчика з відповідними виводами Arduino Mega 2560: контакт SDA датчика – з цифровим піном 20 контролера, а SCL – з піном 21 відповідно.

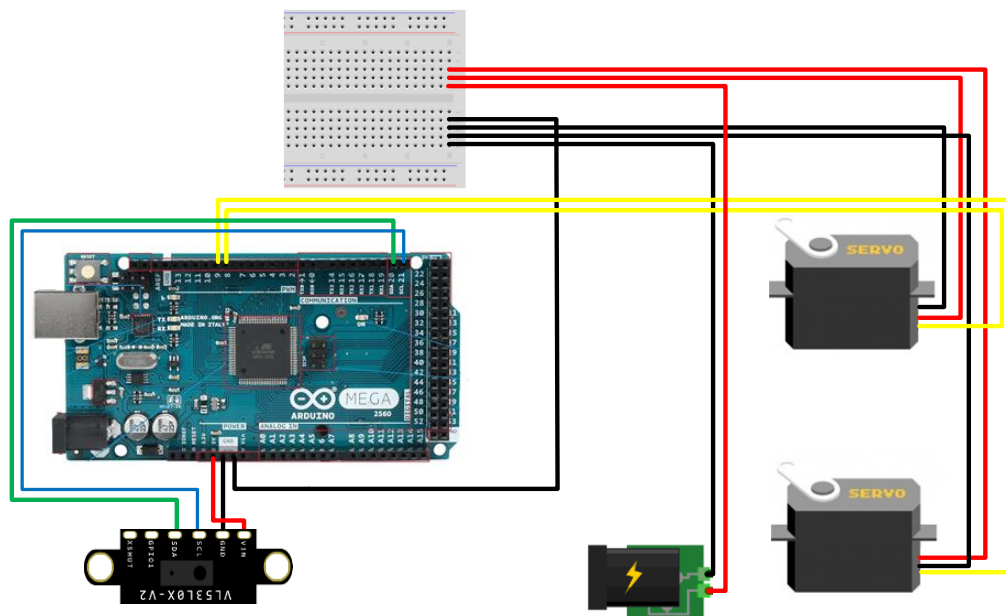


Рисунок 3.2 – Схема з'єднань

Для забезпечення стабільної роботи сенсора також було підключено живлення: пін VIN модуля GY-530 підключено до виходу 5 В на Arduino, а пін GND – до загальної землі (GND). У результаті створено надійний та безпечний зв'язок, який дозволяє контролеру опитувати сенсор у режимі реального часу.

Важливою частиною стало врахування допустимого рівня напруги логічних сигналів, оскільки чип VL53L0X номінально працює з рівнем 3.3 В, а Arduino Mega 2560 має логічний рівень 5 В. У цій конструкції було використано модуль GY-530 із вбудованим понижувальним резистивним делителем і стабілізатором, який дозволяє безпечно підключати пристрій без додаткових логічних перетворювачів рівнів.

Щодо сервоприводів MG996R, підключення здійснювалося через окремі цифрові піни Arduino. Було використано пін 9 для першого сервомотора та пін 10 для другого, що відповідає базовій конфігурації бібліотеки Servo.h. Обидва серводвигуни мають три контакти: сигнальний (жовтий або білий), живлення

(червоний) і землю (чорний або коричневий). Сигнальні дроти з'єднано із зазначеними піновими виходами на Arduino Mega 2560. Живлення та земля сервомоторів були підключені до зовнішнього джерела живлення (5-6 В), через розгалуження з урахуванням струмових характеристик (близько 2,5 А на мотор під навантаженням).

Щоб захистити контролер від можливого перевантаження по струму, сервоприводи підключено не безпосередньо до 5 В лінії Arduino, а до зовнішнього блока живлення, спільного з Arduino по землі. Це забезпечує стабільність роботи навіть під час активного повороту платформи.

Узгодження сигнальних рівнів і правильна організація шини I²C мають суттєве значення, особливо в умовах підвищених перешкод або довгих з'єднувальних проводів. У проєкті використано екрановані провідники та зменшену довжину з'єднань для уникнення флуктуацій сигналу. Відповідна апаратна реалізація підвищує надійність системи в динамічних умовах позиціонування, де будь-який збій комунікації призводить до втрати точності.

Таким чином, фізичне підключення сенсора VL53L0X до Arduino Mega 2560 через I²C-шину здійснено з урахуванням рівнів напруги, стабільності живлення та розмежування функцій між піновими виходами, а серводвигуни керуються окремими цифровими каналами із незалежним живленням.

3.3 Живлення елементів системи

При створенні системи лазерного позиціонування надзвичайно важливо правильно організувати живлення всіх компонентів, щоб уникнути перевантаження живильної лінії, перегріву чи нестабільної роботи пристрою.

Першочергово, потрібно усвідомити обмеження самої плати Arduino Mega 2560. Її вбудований регулятор напруги дозволяє живлення як від USB (5 В, максимум 500 мА), так і через роз'єм живлення (Vin), де рекомендована напруга становить 7-12 В. Проте вбудований стабілізатор на платі не

розрахований на високі струмові навантаження, особливо під час активного використання сервоприводів.

Сервоприводи MG996R є потужними елементами, які при навантаженні можуть споживати до 2,5 А кожен. Це означає, що при одночасному використанні двох сервоприводів система може потребувати струму понад 5 А. У таких умовах застосування живлення безпосередньо з плати Arduino є небажаним, оскільки це призведе до просідання напруги або навіть пошкодження плати.

Щоб уникнути цього, було застосовано окреме джерело живлення для сервомоторів. Зовнішній модуль живлення з вихідною напругою 5 В та струмом щонайменше 5 А було з'єднано з сервоприводами через розподільчу шину живлення. Важливо було не забути про спільне заземлення всіх компонентів системи: землю (GND) зовнішнього джерела необхідно було підключити до спільної землі Arduino, щоб уникнути потенційної різниці напруг та шумів.

Для живлення самого мікроконтролера Arduino Mega 2560 було застосовано або USB-кабель від комп'ютера (під час відлагодження коду) та блок живлення 9 В через роз'єм живлення. Такий підхід гарантував, що потужні елементи не перевантажували саму плату Arduino.

Лідарний сенсор має дуже низьке енергоспоживання (звичайно менше ніж 20 мА), тому його живлення могло бути безпечно підключене до пінів 5 В або 3.3 В Arduino. У нашій конфігурації був обраний пін 5 В, оскільки модуль GY-530 містить вбудований стабілізатор і може працювати з напругою 5 В, що значно спрощує інтеграцію.

Стабільність системи у великій мірі залежала від вибору джерела живлення. У тестуванні було помічено, що навіть невелика нестабільність живлення приводила до спотворення сигналів з датчика відстані або до нестабільної роботи сервомоторів, особливо у фазі активного сканування, коли відбувався постійний поворот платформи.

3.4 Бібліотеки та драйвери для LiDAR

Успішна реалізація системи лазерного позиціонування потребувала коректного підключення відповідних програмних засобів – бібліотек і драйверів – до середовища розробки Arduino IDE. Оскільки датчик належить до класу Time-of-Flight сенсорів, які взаємодіють із контролером через інтерфейс I²C, було обрано низку перевірених бібліотек, що забезпечують зчитування відстані, керування режимами роботи сенсора та виконання калібрування.

Початково було встановлено офіційну бібліотеку від компанії Pololu, яка розробила модуль GY-530 на основі чіпа VL53L0X. Цю бібліотеку можна знайти безпосередньо через менеджер бібліотек Arduino IDE, у вікні «Library Manager», здійснивши пошук за ключовим словом VL53L0X. Було обрано саме версію Pololu VL53L0X, оскільки вона надає повний набір функцій, включаючи керування таймінгом вимірювання, читання кількох сенсорів на одній шині (у випадку масштабування) та вбудовані функції для діагностики.

Після встановлення бібліотеки було здійснено імпорт відповідних заголовочних файлів у скетч Arduino (рис. 3.3)

```
#include <Wire.h>
#include <VL53L0X.h>
```

Рисунок 3.3 – Імпорт файлів у скетч Arduino

Ініціалізація об'єкта сенсора виконувалась у секції setup() (рис. 3.4).

```
void setup() {
  Serial.begin(115200);
  Wire.begin();
  if (!lox.begin()) {
```

Рисунок 3.4 – Ініціалізація об'єкта сенсора

Використання методу `startContinuous()` дозволило реалізувати безперервне зчитування відстані без необхідності запускати кожне окреме вимірювання вручну. У тілі циклу `loop()` здійснювалось регулярне опитування сенсора (рис. 3.5):

```
void loop() {  
  uint16_t distance = lidar.readRangeContinuousMillimeters();  
  Serial.print("Відстань: ");  
  Serial.print(distance);  
  Serial.println(" мм");  
  delay(100);  
}
```

Рисунок 3.5 – Опитування сенсора

Ці дані передавались у зовнішнє програмне середовище через послідовний порт, де вони оброблялись далі в середовищі NetBeans.

З метою забезпечення сумісності з іншими бібліотеками та уникнення конфліктів, у коді було заборонено одночасне використання інших пристроїв, які могли б блокувати I²C-шину. Зокрема, серводвигуни MG996R не потребують I²C і керуються за допомогою ШІМ-сигналів, що дозволило уникнути перешкод.

Особливої уваги заслуговує налаштування часу таймауту (`setTimeout(500)`), яке запобігає зависанню програми в разі втрати зв'язку з датчиком. Без цього параметра вбудовані функції очікували б нескінченно довго на відповідь від сенсора, що критично у реальних умовах експлуатації, наприклад, при трясках, зміні температури чи перешкодах.

Було також випробувано альтернативну бібліотеку `Adafruit_VL53L0X`, яка має зручний інтерфейс, проте на момент тестування вона не забезпечувала достатньої гнучкості в керуванні режимами вимірювання та не підтримувала всі функції, необхідні для багатоточкового сканування.

Для реалізації взаємодії з сенсором через середовище Java NetBeans було використано стандартну обробку серійного потоку. Програма, написана на Java, приймала текстові значення з порту COM і зберігала їх у масиви для подальшої візуалізації, аналізу та формування карти простору.

3.5 Збір і обробка даних про відстань

Для здійснення механічного сканування навколишнього середовища датчик VL53L0X було закріплено на рухомій платформі, яка обертається за допомогою двох сервомеханізмів MG996R. Перший сервопривід забезпечує обертання сенсора по горизонталі, другий – по вертикалі. Таке розташування надало змогу отримувати просторову інформацію про відстань до об'єктів, що дозволило перейти від одномірного до тривимірного аналізу середовища.

Після встановлення апаратної частини системи, включно з підключенням лазерного датчика VL53L0X на модулі GY-530 до плати Arduino Mega 2560, наступним кроком є реалізація програмного механізму для збору та первинної обробки інформації про відстань до об'єктів.

Ініціалізацію пристрою розпочато із оголошення відповідного об'єкта VL53L0X в коді програми. У функції `setup()` виконано послідовність команд, необхідних для перевірки з'єднання з сенсором та його калібрування. Код ініціалізації наведено нижче на рисунку 3.6.

Цей алгоритм дозволяє зчитувати показники дистанції з періодичністю приблизно 10 вимірів за секунду. Такий підхід забезпечує достатню щільність точок для побудови елементарної карти навколишнього середовища. Покази лазерного далекоміра можуть бути нестабільними або містити шум – тобто випадкові стрибки або неточності, що не відображають реальної зміни відстані. Щоб зробити дані більш точними та «плавними», використовується метод ковзного середнього. Замість використання кожного окремого виміру одразу, система зберігає останні, наприклад, 5 або 10 значень, і кожного разу обчислює їх середнє арифметичне. Це середнє значення і є результатом, що потім

використовується для побудови карти чи позиціонування. Таким чином, випадкові викиди в даних згладжуються. Відповідний фрагмент коду для обчислення середнього значення показано на рисунку 3.7.

```
void setup() {
  Serial.begin(115200);
  while (!Serial); // Очікуємо з'єднання із монітором порту (опційно)
  delay(100);      // Коротка затримка для стабільності

  Wire.begin();   // Ініціалізуємо I2C
  if (!I2C.begin()) {
    Serial.println(F("Помилка ініціалізації датчика VL53L0X"));
    while (1); // Зупинка програми
  }

  Serial.println(F("Датчик VL53L0X успішно ініціалізовано"));
}
```

Рисунок 3.6 – Код перевірки з'єднання з сенсором та його калібрування

```
#define BUFFER_SIZE 5
int distanceBuffer[BUFFER_SIZE];
int index = 0;

int getFilteredDistance(int newVal) {
  distanceBuffer[index] = newVal;
  index = (index + 1) % BUFFER_SIZE;
  int sum = 0;
  for (int i = 0; i < BUFFER_SIZE; i++) {
    sum += distanceBuffer[i];
  }
  return sum / BUFFER_SIZE;
}
```

Рисунок 3.7 – Обчислення середнього значення

Функція `getFilteredDistance()` застосовується після зчитування значення з сенсора, перш ніж ці дані передаються у блок побудови карти. Такий метод зменшує ймовірність виникнення артефактів, пов'язаних із одиничними помилковими зчитуваннями.

Особливу увагу звернув на те, що GY-530 дозволяє налаштовувати таймінги вимірювання. Збільшення часу вимірювання (наприклад, до 50 мс) підвищує точність, але зменшує частоту оновлення даних. У даній реалізації був використаний режим з типовою тривалістю вимірювання 33 мс, що дозволило досягти компромісу між точністю та швидкістю. Підготовлені дані передаються далі у модуль побудови двовимірної мапи, де відбувається їх геометрична інтерпретація з урахуванням положення сервоприводів.

3.6 Побудова карти навколишнього середовища

Після отримання стабілізованих даних про відстань з лазерного сенсора, система переходить до етапу їх геометричної інтерпретації. Застосовано просторову трансформацію координат, яка дозволяє перетворити показники відстані у двовимірну полярну сітку. Оскільки лазерний сенсор розташований на обертовій платформі, керованій сервоприводом MG996R, кожне значення відстані відповідає певному куту повороту, що дозволяє виконати розгортання вимірювань у площині.

Під час сканування навколишнього простору сервопривод поетапно змінює кут повороту в межах 0° - 180° з кроком у 1° . Відповідний кут задано за допомогою функції `servo.write(angle)` з бібліотеки `Servo.h`, що дозволяє точно позиціонувати сенсор на заданому напрямку. Для кожного положення серводвигуна виконано зчитування відстані, після чого збережено отримані значення разом із відповідними кутами у масив даних.

Основою побудови карти стала реалізація алгоритму полярного сканування, в якому кожен вектор від центра (тобто точки встановлення сенсора) представляє напрямок лазерного променя, а його довжина визначається зчитаною відстанню. Для переведення таких векторів у прямокутну систему координат застосовував класичні тригонометричні перетворення (рис. 3.8).

```
float angle_rad = angle * PI / 180.0;
float x = distance * cos(angle_rad);
float y = distance * sin(angle_rad);
```

Рисунок 3.8 – Переведення векторів у прямокутну систему координат

Після трансформації координати кожної точки передавано через послідовний порт для подальшої візуалізації у середовищі NetBeans Java. Для накопичення даних використовував масив структур, у якому збережені значення кута, відстані та координат (рис. 3.9).

```
struct PointData {
    int angle;
    int distance;
    float x;
    float y;
};

PointData scanData[181];
```

Рисунок 3.9 – Накопичення даних у масиві структур

З метою уникнення розмиття карти внаслідок випадкових коливань у даних, додатково використано поріг достовірності. Дані, в яких $distance < 30$ мм або $distance > 2000$ мм, відкидав як потенційно некоректні. Застосовано також фільтрацію координат через обмеження радіуса області сканування.

Передача зібраної карти здійснювалася через UART, використовуючи стандартний інтерфейс Serial. Структуровано передані дані у вигляді рядків (рис. 3.10).

```
angle:distance:x:y
```

Рисунок 3.10 – Дані у вигляді рядків

Цей формат дозволяє зручно парсити вхідні дані в середовищі NetBeans Java IDE. Додатково додано маркер завершення серії сканування, що дає

зможу Java-програмі визначити момент завершення одного повного розгортання карти.

Після формування карти в пам'яті комп'ютера відбувається її візуалізація. Завдяки модульності та високій частоті оновлення даних, розроблена система дозволяє відобразити динамічні зміни простору в режимі майже реального часу, що є критично важливим для задач мобільної робототехніки та просторової орієнтації пристроїв.

3.7 Алгоритми визначення позиції

Розробка алгоритмів визначення позиції у мобільному середовищі із застосуванням системи лазерного позиціонування вимагала використання комбінації геометричних, евристичних та фільтраційних підходів. Основна мета полягала у тому, щоби забезпечити точне визначення просторового положення датчика в реальному часі шляхом аналізу набору відстаней, отриманих у процесі сканування навколишнього середовища.

Після того, як під час одного повороту сервоприводу було зібрано всі дані про відстані, система шукала серед них стабільні та впізнавані об'єкти – так звані реперні точки, які можна використовувати як опорні координати для точнішого визначення положення і побудови карти навколишнього середовища. У простих умовах тестування, таких як прямокутні кімнати з чітко вираженими стінами, застосовував метод визначення напрямків з максимальними градієнтами змін відстаней. Цей підхід дозволив умовно виділити межі простору та зробити висновок про орієнтацію сенсора.

В основі алгоритму лежало припущення про відносну незмінність навколишнього середовища протягом короткого проміжку часу. Кожна серія сканувань зберігалась у масиві, який порівнювався із попередніми за допомогою метрики середньоквадратичного відхилення. Якщо знайдена відповідність перевищувала заданий поріг, виконувалась корекція орієнтації

шляхом обертання сервоприводу на малий кут у зворотному напрямку та повторного сканування.

Крім того, використано елементи простого векторного аналізу: після перетворення отриманих відстаней на координати у декартовій площині (за допомогою тригонометричних функцій), виконано розрахунок центра ваги для визначення загального напрямку руху або відхилення від нього. Зміщення центру ваги між двома послідовними скануваннями трактував як вектор зсуву, що дозволяло апроксимувати рух системи. У спрощеному вигляді обчислення реалізовано наступним чином (рис. 3.11):

```
float cx = 0, cy = 0;
for (int i = 0; i < num_points; i++) {
    cx += points[i].x;
    cy += points[i].y;
}
cx /= num_points;
cy /= num_points;
```

Рисунок 3.11 – Обчислення вектор зсуву

Система працює на основі евристичних правил та найпростіших геометричних міркувань, що є прийнятним для прототипу на базі Arduino. Варто також зазначити, що через обмежену обчислювальну потужність Arduino Mega складні обчислення реалізовувалися не в реальному часі, а із затримками, що впливало на плавність оновлення позиції.

Застосування таких методів дозволило побудувати базову модель навігації з використанням лише лазерного сенсора та сервоприводу, без потреби у складних IMU або GPS-модулях.

3.8 Виведення результатів та збереження даних

Останній етап функціонування системи лазерного позиціонування полягав у візуалізації результатів та організації збереження отриманих даних. Враховуючи

обмеженість апаратних ресурсів Arduino Mega 2560, застосовано комбінований підхід: базову обробку виконано на мікроконтролері, а візуалізацію і довготривале зберігання – на комп'ютері, до якого Arduino передавав дані через інтерфейс UART.

У програмі на Arduino реалізовано серійну передачу масиву відстаней у форматі, зручному для подальшого розбору у Java-додатку. Наприклад, передача кожної точки відбувалась у вигляді рядків.

Використано середовище розробки NetBeans із бібліотекою RXTX для реалізації Java-програми, яка приймала дані, зберігала їх у структурованому вигляді та відображала карту навколишнього середовища. У графічному інтерфейсі побудовано умовну 2D-візуалізацію у полярній системі координат, перетворену в декартову площину для зручності сприйняття користувачем.

Після обчислення координат за формулами (рис. 3.12), вони відображались на площині за допомогою компоненту JPanel, перетвореного на графічне полотно. Візуалізація (рис. 3.13) оновлювалась у реальному часі, що дозволило контролювати роботу пристрою та оперативно реагувати на зміни в середовищі.

```
point.x = point.distance * Math.cos(Math.toRadians(point.angle));  
point.y = point.distance * Math.sin(Math.toRadians(point.angle));
```

Рисунок 3.12 – Обчислення координат

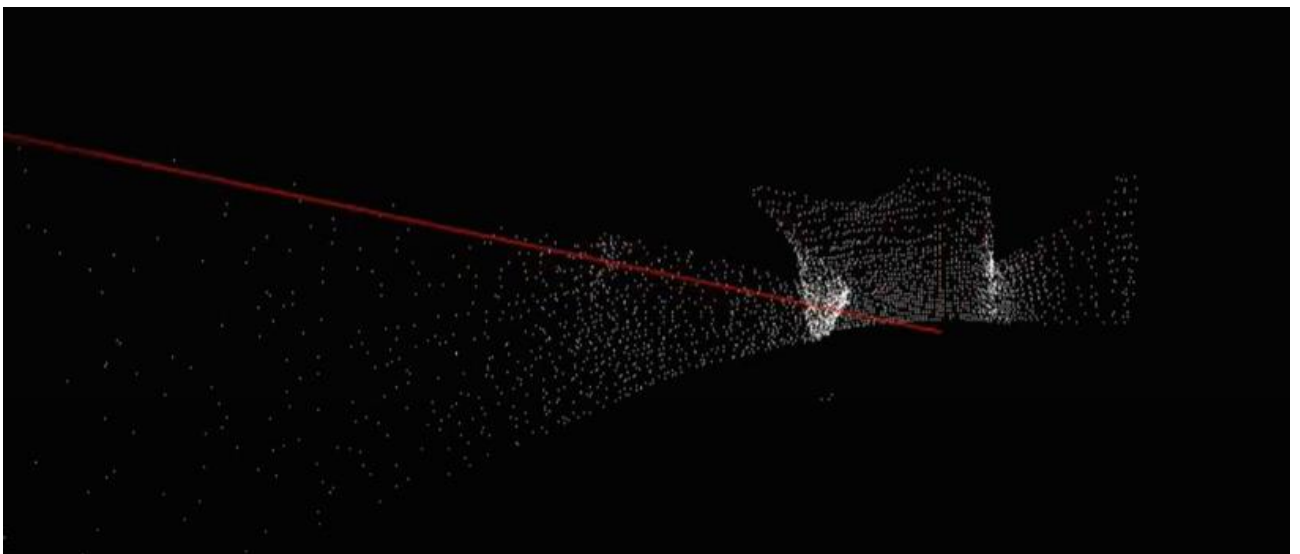


Рисунок 3.13 – Візуалізація даних

Збереження даних (рис. 3.14) реалізовано у форматі CSV з часовими мітками, що дозволило пізніше аналізувати сканування, будувати карти або повторно використовувати отримані значення для симуляцій.

```
PrintWriter writer = new PrintWriter(new FileWriter("scanlog.csv", true));  
writer.println(timestamp + "," + angle + "," + distance);  
writer.close();
```

Рисунок 3.14 – Збереження даних з часовими мітками

Таким чином, реалізовано повний цикл від збору даних сенсором до їх візуалізації, зберігання та можливості використання для подальших алгоритмів навігації.

ВИСНОВКИ

За результатами виконання кваліфікаційної роботи можна зробити відповідні висновки:

На основі теоретичних досліджень було розглянуто принципи функціонування лазерних далекомірів, класифікацію LiDAR-систем та сфери їх застосування. Особливу увагу приділено використанню LiDAR у мобільній робототехніці, де точне позиціонування має вирішальне значення.

Проведено порівняльний аналіз різних моделей LiDAR-датчиків, контролерів та сервоприводів. З урахуванням критеріїв точності, сумісності, енергоспоживання та вартості обрано датчик VL53L0X, мікроконтролер Arduino Mega 2560 і сервоприводи MG996R як оптимальні компоненти для реалізації проєкту.

Проведений огляд програмних засобів продемонстрував доцільність використання комбінації середовищ Arduino IDE для прошивки мікроконтролера та NetBeans Java IDE для побудови десктопного застосунку візуалізації. Такий підхід дозволив досягти гнучкого керування апаратними компонентами та створити ефективне середовище для обробки, аналізу та виведення результатів вимірювання.

На основі попередніх технічних рішень побудовано схему підключення компонентів, виконано з'єднання відповідно до технічних характеристик. Створено програмний код, що забезпечує опитування датчика, обертання сервоприводів та передачу зібраних даних на ПК.

Оцінено можливості Arduino IDE для програмування мікроконтролера, а також Java-середовища NetBeans для візуалізації результатів. Обрані інструменти забезпечують достатню функціональність і зручність у реалізації логіки керування системою позиціонування.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Low-drift and real-time lidar odometry and mapping. *SpringerLink*. URL: <https://link.springer.com/article/10.1007/s10514-016-9548-2>. (дата звернення: 12.01.2025).
2. News – Basic Principle and application of TOF(Time of Flight) System. *Lumispot-tech*. URL: <https://www.lumispot-tech.com/news/tof-time-of-flight-definition-and-principle/> (дата звернення: 12.01.2025)
3. Time-of-Flight principle. *Terabee*. URL: <https://www.terabee.com/time-of-flight-principle/> (дата звернення: 13.01.2025).
4. Understanding the Hardware: LiDAR & Phase-Based Scanning | Existing Conditions. *Existing Conditions 3D Laser Scanning, CAD & BIM Modeling Services*. URL: <https://www.existingconditions.com/knowledge-center-articles/understanding-the-hardware-lidar-phase-based-scanning> (дата звернення: 13.01.2025).
5. Statistical modeling and calibration of triangulation Lidars. *Researchgate.net*. URL: https://www.researchgate.net/publication/303462157_Statistical_Modeling_and_Calibration_of_Triangulation_Lidars (дата звернення: 13.01.2025).
6. Validation of Quasi-Optical Anti reflective Structures for Radar Lenses through Experimentation and Simulation. *Simple search*. URL: <https://www.diva-portal.org/smash/get/diva2:1772453/FULLTEXT02> (дата звернення: 14.01.2025)
7. What is the composition, working principle and application of Quanhom® laser rangefinder. *Quanhom.com*. URL: <https://www.quanhom.com/f739216/What-is-the-composition-working-principle-and-application-of-Quanhom-laser-rangefinder.htm> (дата звернення: 14.01.2025).
8. 1D LiDAR Based on Large Aperture FPCB Mirror. *IEEE Xplore*. URL: <https://ieeexplore.ieee.org/abstract/document/8925013> (дата звернення: 15.01.2025).
9. Neuvition. How 2D LiDAR is Revolutionizing the Robotics Industry. *Neuvition*. URL: <https://cdn.neuvition.com/media/blog/how-2d-lidar-is-revolutionizing-the-robotics-industry.html> (дата звернення: 15.01.2025).

10. LiDAR in Cars: How LiDAR technology is making self driving cars a reality. *MRL Consulting Group*. URL: <https://www.mrlcg.com/resources/blog/lidar-in-cars-how-lidar-technology-is-making-self-driving-cars-a-reality/> (дата звернення: 15.01.2025).

11. 7 interesting LiDAR applications. *CloudFactory*. URL: <https://www.cloudfactory.com/blog/interesting-lidar-applications> (дата звернення: 15.01.2025).

12. Mobile LiDAR Scanning: Everything You Need to Know. *YellowScan*. URL: <https://www.yellowscan.com/knowledge/mobile-lidar-scanning-everything-you-need-to-know/> (дата звернення: 15.01.2025).

13. Handheld LiDAR Scanner. *Exyn Technologies*. URL: <https://www.exyn.com/handheld-lidar-scanner> (дата звернення: 15.01.2025).

14. How Lidar Works in Autonomous Cars. *Waymo*. URL: <https://waymo.com/tech> (дата звернення: 17.01.2025).

15. Lidar for Forest Inventory. *USDA Forest Service*. URL: <https://www.fs.usda.gov/research/trees/lidar> (дата звернення: 17.01.2025).

16. Lidar for Surveying. *DJI Enterprise*. URL: <https://enterprise.dji.com/> (дата звернення: 17.01.2025).

17. Lidar in Robotics. *Mapix technologies*. URL: <https://www.mapix.com/lidar-applications/lidar-robotics/> (дата звернення: 17.01.2025).

18. 3D Elevation Program. *USGS*. URL: <https://www.usgs.gov/3DEP> (дата звернення: 17.01.2025).

19. Lidar for Security and Smart Cities. *Quanergy*. URL: <https://www.quanergy.com/solutions/security> (дата звернення: 17.01.2025).

20. Nikodem Bartnik. *See how Nikodem Bartnik integrated LIDAR room mapping into his DIY robotics platform*. Arduino Blog, 2022. URL: <https://blog.arduino.cc/2022/01/31/see-how-nikodem-bartnik-integrated-lidar-room-mapping-into-his-diy-robotics-platform/> (дата звернення: 18.01.2025).

21. TFMini – Micro LiDAR Module Hookup Guide. *SparkFun Learn*. URL: <https://learn.sparkfun.com/tutorials/tfmini---micro-lidar-module-hookup-guide/all> (дата звернення: 18.01.2025).

22. How to Use the RPLIDAR 360° Laser Scanner With Arduino. *Instructables*. URL: <https://www.instructables.com/How-to-Use-the-RPLIDAR-360-Laser-Scanner-With-Ardu/> (дата звернення: 18.01.2025).

23. TFMini Lidar Sensor Datasheet. *Benewake*. URL: <https://www.benewake.com/en/tfmini.html> (дата звернення:: 14.03.2025).

24. VL53L0X Time-of-Flight ranging sensor. *STMicroelectronics*. URL: <https://www.st.com/en/imaging-and-photonics-solutions/vl53l0x.html> (дата звернення:: 13.01.2025).

25. VL53L1X Product Datasheet. *STMicroelectronics*. URL: <https://www.st.com/resource/en/datasheet/vl53l1x.pdf> (дата звернення:: 12.03.2025).

26. LIDAR-Lite v3 Operation Manual and Technical Specifications. *Garmin*. URL: https://static.garmin.com/pumac/LIDAR_Lite_v3_Operation_Manual_and_Technical_Specifications.pdf (дата звернення:: 14.03.2025).

27. HC-SR04 Ultrasonic Sensor Datasheet. *RobotDyn*. URL: <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf> (дата звернення:: 14.03.2025).

28. Arduino Uno Rev3. *Arduino*. URL: <https://store.arduino.cc/products/arduino-uno-rev3> (дата звернення:: 14.03.2025).

29. Arduino Nano. *Arduino*. URL: <https://store.arduino.cc/products/arduino-nano> (дата звернення:: 14.03.2025).

30. Arduino Leonardo. *Arduino*. URL: <https://store.arduino.cc/products/arduino-leonardo> (дата звернення:: 15.03.2025).

31. ESP32 Series Datasheet. *Espressif Systems*. URL: <https://www.espressif.com/en/products/socs/esp32> (дата звернення:: 15.03.2025).

32. Arduino Mega 2560 Rev3. *Arduino*. URL: <https://store.arduino.cc/products/arduino-mega-2560-rev3> (дата звернення:: 16.03.2025).

33. SG90 Micro Servo. *ServoCity*. URL: <https://www.servocity.com/sg90-micro-servo/> (дата звернення: 16.03.2025)
34. DS3218 Servo Motor Specification. *Amazon*. URL: <https://www.amazon.com/DS3218-Digital-Servo/dp/B07G2PY8ND> (дата звернення: 16.03.2025)
35. MG996R Specification. *Tower Pro*. URL: <http://www.towerpro.com.tw/product/mg996r/> (дата звернення: 17.03.2025)
36. Using MG996R with Arduino. *Arduino Forum*. URL: <https://forum.arduino.cc/t/using-mg996r-servo/657897> (дата звернення: 17.03.2025)
37. MG996R Servo Review for Robotics. *Hackster.io*. URL: <https://www.hackster.io/news> (дата звернення: 17.03.2025)
38. Software. *Arduino IDE*. URL: <https://www.arduino.cc/en/software> (дата звернення: 18.03.2025).
39. Getting Started with Arduino IDE 2. *Arduino*. URL: <https://docs.arduino.cc/software/ide-v2/tutorials/getting-started-ide-v2/> (дата звернення: 18.03.2025).
40. LiDAR Toolbox. *Mathworks*. URL: <https://www.mathworks.com/products/lidar.html> (дата звернення: 18.03.2025)
41. Welcome to Apache NetBeans. *NetBeans IDE Official Site*. URL: <https://netbeans.apache.org/> (дата звернення: 18.03.2025)
42. What is jSerialComm? *JSerialComm Library Documentation*. URL: <https://fazecast.github.io/jSerialComm/> (дата звернення: 18.03.2025)
43. IDE PlatformIO для VSCode. *PlatformIO*. URL: <https://docs.platformio.org/en/latest/integration/ide/vscode.html> (дата звернення: 19.03.2025)