

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

**Факультет програмних систем та прикладної математики
Кафедра системного програмування і спеціалізованих комп'ютерних систем**

«До захисту допущено»

Завідувач кафедри

(підпис) Віталій РОМАНКЕВИЧ

“ ____ ” _____ 2026 р.

**Дипломний проєкт
на здобуття ступеня бакалавра
за освітньо-професійною програмою
«Системне програмування та спеціалізовані комп'ютерні системи»
Спеціальність 123 «Комп'ютерна інженерія»**

на тему: Програмно-апаратний модуль сканування простору на основі оптичних датчиків відстані

Виконав: студент IV курсу, групи КВ-23
Молчембаєв Ярослав Андрійович

(підпис)

Керівник, ст.викл. каф.СПСКС
Дробязко Ірина Павлівна

(підпис)

Консультант з нормоконтролю, доц.каф.СПСКС,
к.т.н., доц. Клятченко Ярослав Михайлович

(підпис)

Рецензент

(підпис)

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

Студент

(підпис)

Київ – 2026 року

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

Факультет програмних систем та прикладної математики

Кафедра системного програмування і спеціалізованих комп'ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Освітньо-професійна програма

«Системне програмування та спеціалізовані комп'ютерні системи»

Спеціальність 123 «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ
Завідувач кафедри

_____ Віталій РОМАНКЕВИЧ
(підпис)

“ ____ ” _____ 2025 р.

ЗАВДАННЯ

на дипломний проєкт студента

Молчембаєв Ярослав Андрійович

1. Тема проєкту «Програмно-апаратний модуль сканування простору на основі оптичних датчиків відстані», керівник проєкту Дробязко І.П., старший викладач кафедри системного програмування і спеціалізованих комп'ютерних систем, затверджені наказом по університету від « ____ » _____ 202__ р. № ____
2. Термін подання студентом проєкту 8 червня 2026 р.
3. Вихідні дані до проєкту див. Технічне завдання
4. Зміст пояснювальної записки:
 - 1) Аналіз існуючих рішень;
 - 2) Вибір засобів реалізації;
 - 3) Розробка апаратної та програмної частин модуля
 - 4) Тестування модуля сканування простору

5. Перелік графічного матеріалу:

- 1) Програмно-апаратний модуль сканування простору. Програмна частина
Схема структурна;
- 2) Архітектура програмного забезпечення мікроконтролера. Схема
структурна;
- 3) Програмно-апаратний модуль сканування простору. Схема електрична
принципова;
- 4) Сканування простору у тривимірному режимі. Схема алгоритму;
- 5) Презентація за темою роботи.

Консультанти розділів проєкту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Клятченко Я. М., доцент кафедри СПіСКС, к.т.н., доцент		

7. Дата видачі завдання «12» листопада 2025 р.

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Терміни виконання етапів проєкту	Примітка
1.	Вивчення літератури за тематикою проєкту	25.11.2025	
2.	Аналіз існуючих рішень	12.12.2025	
3.	Розробка технічного завдання	05.01.2026	
4.	Підготовка матеріалів першого розділу дипломного проєкту	29.01.2026	
5.	Підготовка матеріалів другого розділу дипломного проєкту	13.02.2026	
6.	Підготовка матеріалів третього розділу дипломного проєкту	04.03.2026	
7.	Підготовка матеріалів четвертого розділу дипломного проєкту	14.04.2026	
8.	Підготовка графічної частини дипломного проєкту	11.05.2026	
9.	Оформлення документації дипломного проєкту	21.05.2026	
10.	Попередній огляд матеріалів на кафедрі	31.05.2026	

Студент

(підпис)

Ярослав МОЛЧЕМБАЄВ

Керівник проєкту

(підпис)

Ірина ДРОБЯЗКО

АНОТАЦІЯ

Кваліфікаційна робота включає пояснювальну записку (53 с., 41 рис. 2 табл., 4 додатки).

Об'єкт розробки – програмно-апаратний модуль сканування простору на основі оптичних датчиків відстані.

В ході виконання дипломного проєкту:

- проведено аналіз методів побудови існуючих програмно-апаратних систем просторового сканування та розроблено архітектуру модуля сканування;
- обрано мікроконтролер для реалізації функцій модуля;
- розроблено програмне забезпечення мікроконтролера для асинхронного керування датчиками та механічними елементами модуля, а також первинної обробки даних сканування;
- розроблено серверний додаток для взаємодії з модулем просторового сканування та передачі даних сканування;
- розроблено веб-сервіс для візуалізації даних сканування та задання конфігурації.

Модуль дозволяє:

- виконувати сканування заданого простору у різних режимах, виконувати первинну обробку та здійснювати передачу даних у реальному часі;
- фільтрувати помилкові та спотворені дані датчиків;
- функціонувати асинхронно з іншими пристроями;
- використовувати різні цільові платформи мікроконтролерів та апаратного забезпечення;

Інтеграція модуля в системи навігації у просторі дозволить їм отримувати якісні просторові дані при зменшенні енерговитрат та кінцевої вартості системи.

Ключові слова: ПРОСТОРОВЕ СКАНУВАННЯ, LiDAR, ESP32, ПРОГРАМНО-АПАРАТНИЙ МОДУЛЬ, WI-FI, UDP, ToF.

ABSTRACT

The qualification thesis includes an explanatory report (53 pages, 41 figures, 2 tables, 4 appendices).

The object of development is a hardware-software spatial scanning module based on optical distance sensors.

As part of the graduation project, the following tasks were completed:

- an analysis of existing spatial scanning systems and their construction methods was carried out, and the architecture of the scanning module was developed;
- a microcontroller was selected for implementing the module's functionality;
- microcontroller software was developed for asynchronous control of sensors and mechanical components of the module, as well as for primary processing of scanning data;
- a server application was developed for communication with the spatial scanning module and transmission of scanning data;
- a web service was developed for visualization of scanning data and configuration management.

The module provides the following capabilities:

- performing spatial scanning in various operating modes, carrying out primary data processing, and transmitting data in real time;
- filtering erroneous and distorted sensor data;
- asynchronous operation with other devices;
- support for various target microcontroller platforms and hardware configurations.

Integration of the module into spatial navigation systems can allow to acquire the qualitative spatial data with reduced power consumption and overall system cost.

Keywords: SPATIAL SCANNING, LiDAR, ESP32, HARDWARE-SOFTWARE MODULE, WI-FI, UDP, ToF.

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
	A4	ІАЛЦ.467200.002 ТЗ	Програмно-апаратний	4		
			модуль сканування			
			простору на основі			
			оптичних датчків			
			відстані			
			Технічне завдання			
		ІАЛЦ.467200.003 ТП	Програмно-апаратний	2		
			модуль сканування			
			простору на основі			
			оптичних датчків			
			відстані			
			Відомість технічного			
			проєкту			
		ІАЛЦ.467200.004 ПЗ	Програмно-апаратний	53		
			модуль сканування			
			простору на основі			
			оптичних датчків			
			відстані			
			Пояснювальна			
			записка			
			ІАЛЦ.467200.001 ОА			
Зм.	Арк	№ докум	Підпис	Дата	Програмно-апаратний модуль сканування простору на основі оптичних датчиків відстані Опис альбому Літ. Арк. Арк-ів 1 2 КПІ ім. Ігоря Сікорського, ФПСІМ гр. КВ-23	
Розроб.		Молчембаєв Я. А.				
Перевір.		Дробязко І.П.				
Н.контр.		Клятченко Я.М				
Затв.		Романкевич В. О.				

[illegible]

ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ.....	2
2. ПІДСТАВА ДЛЯ РОЗРОБКИ.....	2
3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ.....	2
4. ДЖЕРЕЛА РОЗРОБКИ.....	2
5. ТЕХНІЧНІ ВИМОГИ.....	2
5.1. Вимоги до програмного продукту, що розробляється.....	2
1.1. Вимоги до апаратного забезпечення.....	3
1.2. Вимоги до програмного та апаратного забезпечення користувача.....	3
6. ЕТАПИ РОЗРОБКИ.....	4

					ІАЛЦ.467200.002 ТЗ						
Зм.	Арк	№ докум	Підпис	Дата	Програмно-апаратний модуль сканування простору на основі оптичних датчиків відстані Технічне завдання			Літ.	Арк.	Арк-ів	
Розроб.		Молчембаєв Я. А.								1	4
Перевір.		Дробязко І. П.									
Н.контр.		Клятченко Я.М									
Затв.		Романкевич В. О.						КПІ ім. Ігоря Сікорського, ФПСМ гр. КВ-23			

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ

Назва розробки: «Програмно-апаратний модуль просторового сканування на основі оптичних датчиків відстані».

Галузь застосування: системи просторового сканування та навігації, системи наземного роботизованого керування.

2. ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на дипломне проектування на здобуття першого (бакалаврського) рівня вищої освіти, затверджене кафедрою системного програмування і спеціалізованих комп'ютерних систем Національного технічного університету України «Київський Політехнічний Інститут імені Ігоря Сікорського».

3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ

Метою даного проєкту є створення програмно-апаратного модуля для збору, первинної обробки та передачі даних для формування цифрового представлення простору у реальному часі, який забезпечує отримання якісних просторових даних при зменшенні енерговитрат і вартості розробки.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом інформації є технічна та науково-технічна література, технічна документація, публікації в періодичних виданнях та електронні статті у мережі Інтернет.

5. ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до програмного продукту, що розробляється

- збір, первинна обробка та передача просторових даних у потоковому режимі через мережу Wi-Fi протоколом UDP;

					ІАЛЦ.467200.002 ТЗ	Арк.
						2
Змн.	Арк.	№ докум.	Підпис	Дата		

- можливість задання конфігурації параметрів області та кроку сканування;
- сумісність з будь-якими механічними елементами, що дозволяють виконувати обертання, та датчиками відстані;
- сумісність з будь-якою 32-розрядною платформою мікроконтролера;
- можливість розширення користувачем функціональності модуля;
- можливість формування діагностичних повідомлень;

5.2. Вимоги до апаратного забезпечення

- 32-розрядний мікроконтролер ESP32 з підтримкою Wi-Fi;
- підтримка інтерфейсу UART;
- джерело живлення напругою 5V;
- захист від перепадів напруги;

5.3. Вимоги до програмного та апаратного забезпечення користувача

- ПК з підтримкою Wi-Fi;
- Операційна система Windows, Linux, macOS;
- Браузер з підтримкою технологій WebGL, WebSocket;
- Блок живлення 5В, мінімум 2А з захистом від короткого замикання;

					ІАЛЦ.467200.002 ТЗ	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

6. ЕТАПИ РОЗРОБКИ

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів
1.	Вивчення літератури за тематикою проекту	25.11.2025
2.	Аналіз існуючих рішень	12.12.2025
3.	Розробка технічного завдання	05.01.2026
4.	Підготовка матеріалів першого розділу дипломного проекту	29.01.2026
5.	Підготовка матеріалів другого розділу дипломного проекту	13.02.2026
6.	Підготовка матеріалів третього розділу дипломного проекту	04.03.2026
7.	Підготовка матеріалів четвертого розділу дипломного проекту	14.04.2026
8.	Підготовка графічної частини дипломного проекту	11.05.2026
9.	Оформлення документації дипломного проекту	21.05.2026
10.	Попередній огляд матеріалів на кафедрі	31.05.2026

					ІАЛЦ.467200.002 ТЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
	A4	ІАЛЦ.467200.004 ПЗ	Програмно-апаратний	53		
			модуль сканування			
			простору на основі			
			оптичних датчиків			
			відстані			
			Пояснювальна записка			
	A1	ІАЛЦ.467200.005 Д1	Програмно-апаратний	1		
			модуль сканування			
			простору. Програмна			
			частина			
			Схема структурна			
	A1	ІАЛЦ.467200.006 Д2	Архітектура програмного	1		
			забезпечення			
			мікроконтролера			
			Схема структурна			
	A1	ІАЛЦ.467200.007 Е3	Програмно-апаратний	1		
			модуль сканування			
			простору на основі			
			Схема електрична			
			принципова			
Зм.	Арк	№ докум	Підпис	Дата	ІАЛЦ.467200.003 ТП <i>Програмно-апаратний модуль сканування простору на основі оптичних датчиків відстані</i> Відомість технічного проекту	
Розроб.		Молчембаєв Я. А.				
Перевір.		Дробязко І.П.				
Н.контр.		Клятченко Я.М				
Затв.		Романкевич В. О.				
					Літ.	Арк.
						1
						2
					КПІ ім. Ігоря Сікорського, ФПСІМ гр. КВ-23	

[illegible]

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО ДИПЛОМНОГО ПРОЄКТУ**

на тему: «Програмно-апаратний модуль сканування простору на основі оптичних датчиків відстані»

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ.....	4
ВСТУП.....	5
1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБҐРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЕКТУ.....	6
1.1 Просторове сканування.....	6
1.2 Аналіз існуючих рішень для сканування простору.....	7
1.2.1 Класичні LiDAR-системи.....	7
1.2.2 ToF-модулі.....	9
1.2.3 Фотограмметричні системи просторового сканування.....	11
1.2.4 Стереометричні системи комп'ютерного зору.....	12
1.3 Обґрунтування теми дипломного проекту та постановка задачі.....	16
2. ПРОЄКТУВАННЯ АПАРАТНОЇ ЧАСТИНИ МОДУЛЯ.....	20
2.1 Вибір мікроконтролера.....	20
2.2 Вибір датчика відстані.....	24
2.3 Вибір механічних елементів.....	28
2.4 Схема живлення.....	31
3. РОЗРОБКА ПРОГРАМНОЇ ЧАСТИНИ МОДУЛЯ.....	33
3.1 Розробка прошивки мікроконтролера.....	33
3.1.1 План та послідовність етапів розробки.....	33

					ІАЛЦ.467200.004 ПЗ			
Зм.	Арк	№ докум	Підпис	Дата				
Розроб.		Молчембаєв Я. А.			<i>Програмно-апаратний модуль сканування простору на основі оптичних датчиків відстані</i> Пояснювальна записка	Літ.	Арк.	Арк-ів
Перевір.		Дробязко І.П.					1	4
						КПІ ім. Ігоря Сікорського, ФПСІМ гр. КВ-23		
Н. контр.		Клятченко Я. М.						
Затв.		Романкевич В. О.						

3.1.2 Архітектура програмного забезпечення мікроконтролера.....	33
3.1.3 Реалізація мережевої взаємодії.....	39
3.1.4 Логування подій програмної частини.....	41
3.2 Розробка програмного забезпечення для візуалізації роботи модуля.....	41
3.2.1 Реалізація сервера для взаємодії з модулем сканування.....	41
3.2.2 Реалізація інтерфейсу користувача.....	43
4. ТЕСТУВАННЯ МОДУЛЯ СКАНУВАННЯ.....	45
4.1 Запуск модуля сканування.....	45
4.2 Тестування простої сцени.....	46
4.3 Тестування сцени з декількома об'єктами.....	47
4.4 Тестування режиму одновимірного сканування.....	49
ВИСНОВКИ.....	50
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	52

					ІАЛЦ.467200.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		2

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

АЦП – Аналогово-цифровий перетворювач.

ОСРВ – Операційна Система Реального Часу.

ЦАП – Цифро-аналоговий перетворювач.

ШИМ – Широтно-імпульсна модуляція.

API (Application Programming Interface) – прикладний програмний інтерфейс.

CV (Computer Vision) – комп'ютерний зір.

HAL (Hardware Abstraction Library) – шар абстракції над апаратурою.

I2C (Inter-Integrated Circuit) – послідовний інтерфейс передачі даних.

LiDAR (Light Detection and Ranging) – технологія дистанційного зондування, що вимірює відстань до об'єктів за допомогою лазерних променів.

SDK (Software Development Kit) – набір для розробки програмного забезпечення.

SPI (Serial Peripheral Interface) – послідовний інтерфейс передачі даних.

SoC (System on Chip) – система, що в одному чипі містить різні компоненти, для яких, як правило, використовують окремі мікросхеми.

TCP/IP (Transmission Control Protocol / Internet Protocol) – стек мережних протоколів комп'ютерних мереж.

ToF (Time of Flight) – технологія вимірювання відстані на основі вимірювання часу польоту світлових хвиль.

UART (Universal Asynchronous Receiver-Transmitter) – послідовний інтерфейс передачі даних.

UDP (User Datagram Protocol) – протокол транспортного рівня стеку протоколів TCP/IP.

Wi-Fi (Wireless Fidelity) – стандарт передачі даних у бездротових мережах.

					ІАЛЦ.467200.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

У сучасних вбудованих системах все більшого поширення набувають засоби обробки просторових даних навколишнього середовища. Це пов'язано зі стрімким розвитком робототехніки, навігаційних систем, автономних комплексів. Для прийняття рішень такі системи у реальному часі потребують отримання просторової інформації про середовище, а саме: форма об'єктів у просторі, їх взаємне положення, відстань до цих об'єктів тощо. Одним із найпоширеніших способів отримання такої інформації є використання оптичних датчиків відстані, організація та первинна обробка даних від датчиків у зручному для подальшої обробки форматі.

Сучасні системи сканування простору дозволяють в реальному часі отримати досить точне представлення сцени у вигляді дискретного набору точок та передавати їх для обробки системам прийняття рішень, побудови карт місцевості, навігації автономних пристроїв та роботизованих систем. Часто впровадження та використання таких систем сканування ставить ряд технічних викликів, таких як: рівень споживання енергії, вартість впровадження та доступність для масового використання.

Метою дипломного проєкту є створення бюджетного, незалежного від кінцевої платформи, масштабованого та енергоефективного програмно-апаратного рішення для збору, первинної обробки та передачі просторових даних для подальшої обробки та прийняття рішень у реальному часі, з можливістю гнучко змінювати конфігурацію параметрів та програмно розширювати функціональність без втручання в ядро реалізації.

					ІАЛЦ.467200.004 ПЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБҐРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЕКТУ

1.1 Просторове сканування

Просторове сканування є процесом отримання інформації про геометричні характеристики навколишнього середовища та просторове розташування об'єктів. Основною метою такого сканування є формування цифрового представлення простору, придатного для подальшої обробки, аналізу або використання в автоматизованих системах.

Залежно від поставленої задачі просторове сканування може виконуватися різними методами. Одним із найпоширеніших є активне сканування, при якому система випромінює сигнал та аналізує його відбиття від об'єктів. До таких технологій належать лазерні далекоміри, LiDAR-системи, ультразвукові та радіолокаційні датчики. Іншим підходом є пасивне сканування, що базується на аналізі інформації, отриманої від камер та інших сенсорів без власного випромінювання сигналів. У цьому випадку для визначення форми та положення об'єктів застосовуються методи комп'ютерного зору та машинного навчання.

Результатом роботи систем просторового сканування можуть бути карти глибин, хмари точок, тривимірні моделі об'єктів або карти навколишнього середовища. Такі дані дозволяють визначати відстані між об'єктами, знаходити перешкоди, оцінювати геометричні параметри середовища та виконувати навігацію в автоматичному режимі.

Сучасні технології просторового сканування широко використовуються в різних галузях. У робототехніці вони забезпечують автономне пересування роботів та безпілотних транспортних засобів. У промисловості застосовуються для контролю якості продукції, побудови цифрових моделей виробничих об'єктів та автоматизації технологічних процесів. У будівництві та геодезії просторове сканування використовується для створення точних тривимірних моделей споруд і місцевості. Значного поширення такі технології набули також у системах безпеки,

					ІАЛЦ.467200.004 ПЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

медицині, технологіях віртуальної та доповненої реальності.

Постійне зростання вимог до точності, швидкості та доступності отримання просторових даних обумовлює розвиток нових систем сканування. Особливо актуальним є створення компактних та енергоефективних пристроїв, здатних працювати в реальному часі та інтегруватися з сучасними обчислювальними платформами. Це зумовлює необхідність розробки спеціалізованих модулів просторового сканування, які можуть використовуватися як складова робототехнічних комплексів, систем моніторингу та інших інтелектуальних технічних систем.

Таким чином, систему сканування простору можна визначити як апаратну або програмно-апаратну систему, призначену для отримання, обробки та передачі даних про геометричні характеристики навколишнього середовища та взаємне розташування об'єктів у просторі.

1.2 Аналіз існуючих рішень для сканування простору

1.2.1 Класичні LiDAR-системи

LiDAR – це технологія активного сканування навколишнього середовища за допомогою лазерних імпульсів. Традиційні LiDAR-системи використовують лазерний промінь для вимірювання відстані до об'єктів шляхом визначення часу повернення сигналу та обчислення відстані за формулою:

$$d = \frac{c \cdot t}{2}$$

де d – відстань, c – швидкість світла, t – проміжок часу, за який лазерний промінь був запущений та зафіксовано його повернення.

LiDAR-системи роблять мільйони таких запусків за секунду в різних напрямках, та на виході дають представлення навколишнього середовища у вигляді хмари точок (рис. 1.2.1).

					ІАЛЦ.467200.004 ТЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

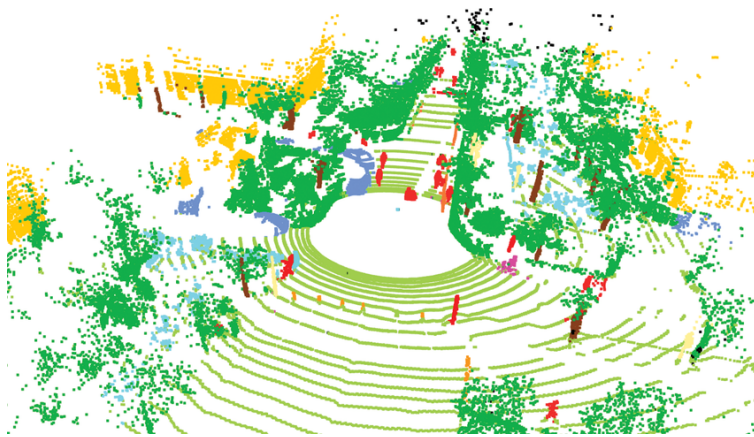


Рисунок 1.2.1 – Приклад хмари точок LiDAR-системи

Розрізняють декілька форм-факторів LiDAR-систем. До них відносяться:

- Механічні – зміна положення лазерних променів відбувається за рахунок механічного обертання системи. Такі пристрої часто використовуються в робототехніці та автоматизованих безпілотних системах. Як приклад системи, що дозволяє формувати тривимірні представлення зображень можна навести Velodyne HDL-64E, який використовується у безпілотних автомобілях (рис. 1.2.2).



Рисунок 1.2.2 – Velodyne HDL-64E

- Твердотільні (Solid-State LiDAR) – зміна напрямку лазерних променів здійснюється за допомогою без механічних елементів. Наприклад, MEMS (Micro-electromechanical system) системи використовують єдиний нерухомий лазер та дзеркало, яке коливається з фіксованою частотою одночасно вертикально та горизонтально, забезпечуючи проходження променя по всій області сканування. OPA (Optical Phased Array) системи

					ІАЛЦ.467200.004 ТЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

- використовують хвильову природу світла, виконуючи зсув фази хвиль світла за допомогою спеціальних електронних компонентів. Як приклад Solid-State систем можна навести Ouster OS1 (рис. 1.2.3).



Рисунок 1.2.3 – Ouster OS1

До переваг таких систем можна віднести високу швидкість функціонування, точність вимірювань та широкий кут огляду. З недоліків – значне енергоспоживання, складність конструкції та високу вартість, що робить їх впровадження у масові автономні системи проблематичним.

1.2.2 ToF-модулі

ToF – це технологія вимірювання відстані до об'єкту шляхом вимірювання часу польоту фотонів лазера. Принцип отримання відстані у ToF датчиках є аналогічним тому, який використовується у LiDAR-системах. Основною перевагою ToF-технології є можливість безконтактного вимірювання відстані з достатньо високою точністю при компактних розмірах сенсора та низькому енергоспоживанні. На відміну від LiDAR-систем, ToF-модулі не дають повноцінного представлення сцени, а лише вимірюють відстань до певної точки, або усереднену відстань до області, заданої кутом огляду. ToF-модулі, як правило, мають простий апаратний інтерфейс передачі даних (наприклад, I2C або UART), та забезпечують просту інтеграцію у вбудовані системи.

ToF-модулі мають просту будову: по своїй суті – це електричні схеми, що складаються з джерела лазерних імпульсів, датчика для вимірювання часу польоту

					ІАЛЦ.467200.004 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

та просту процесорну систему для перетворення даних світлової хвилі, на придатні для обробки дані про відстань. Найбільш поширеним прикладом промислових ToF-модулів є мікросхема Benewake LiDAR TFmini-S (хоча в назві і є слово LiDAR у цілях маркетингу, насправді це ToF датчик) (рис. 1.2.4).



Рисунок 1.2.4 – Benewake LiDAR TFmini-S

Основною перевагою ToF-датчиків є висока швидкодія вимірювань, що дозволяє отримувати дані практично в режимі реального часу. Це робить їх придатними для застосувань, де необхідна оперативна побудова карти простору або швидке реагування на зміну оточення. Крім того, такі сенсори мають малий форм-фактор та низьке енергоспоживання, що є критично важливим для мобільних та вбудованих систем. Ще однією важливою перевагою є простота інтеграції з мікроконтролерними платформами, оскільки більшість ToF-сенсорів використовують стандартні цифрові інтерфейси передачі даних, такі як I2C або SPI. Це значно спрощує апаратну реалізацію системи та зменшує вимоги до додаткових компонентів.

Однак, незважаючи на переваги, варто підкреслити, що такі модулі не дають повного зображення просторової сцени, та часто можуть давати значну похибку вимірювань. Також ToF-сенсори залежать від відбивної здатності поверхні, до яких вимірюють відстань. Часто такі сенсори мають широкий кут огляду, що може спотворювати результати вимірювань до поверхонь, що знаходяться під кутом до датчика, або розмивати межі об'єктів.

					ІАЛЦ.467200.004 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

1.2.3 Фотограмметричні системи просторового сканування

Фотограмметрія є методом побудови просторового представлення об'єктів або середовища на основі аналізу набору фотографій, отриманих з різних ракурсів. Цей підхід базується на алгоритмах комп'ютерного зору, які визначають положення характерних точок на зображеннях та відновлюють їх координати у тривимірному просторі. Сучасні фотограмметричні системи активно використовуються у картографії, архітектурному моделюванні, археології, робототехніці та створенні цифрових копій об'єктів. Для реалізації таких систем можуть застосовуватись як звичайні цифрові камери, так і спеціалізовані багатокамерні комплекси. Основною перевагою фотограмметрії є можливість отримання деталізованих тривимірних моделей без використання спеціалізованих лазерних сканерів. Також даний підхід дозволяє одночасно отримувати текстуровану модель об'єкта.

Метод фотограмметрії, на відміну від попередніх рішень, що використовуються для просторового сканування, не використовує послідовні вимірювання відстаней для побудови карти глибин сцени. Натомість, ця технологія застосовує принципи комп'ютерного зору та аналіз цифрових зображень. Як правило, такі системи використовують інструменти глибинного машинного навчання.

Основною ідеєю фотограмметричного підходу є пошук однакових характерних точок на різних зображеннях та подальше визначення їх просторового положення. Для цього використовуються алгоритми детекції особливостей зображення, порівняння ключових точок та оцінки геометричних параметрів сцени. Типовий процес побудови просторового представлення у фотограмметричних системах включає декілька основних етапів:

- отримання набору фотографій об'єкта або середовища з різних ракурсів;
- попередню обробку зображень;
- виявлення та зіставлення характерних точок;
- оцінку положення камери для кожного кадру;

					ІАЛЦ.467200.004 ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

- реконструкцію просторової структури сцени;
- формування хмари точок;
- побудову полігональної або текстурованої 3D-моделі.

Перевагою фотограмметрії є можливість отримання високо деталізованих та текстурованих тривимірних моделей (рис. 1.2.4) без використання лазерних сканерів.

Однак фотограмметричні системи мають ряд суттєвих недоліків. Якість реконструкції значною мірою залежить від умов освітлення, кількості та якості вихідних зображень, а також текстурованості поверхонь. Однотонні або блискучі об'єкти можуть суттєво ускладнювати процес реконструкції. Крім того, алгоритми побудови тривимірних моделей потребують значних обчислювальних ресурсів та часто не придатні для роботи у режимі реального часу на малопотужних вбудованих системах.



Рисунок 1.2.4 – Приклад реконструкції об'єкта методом фотограмметрії

1.2.4 Стереометричні системи комп'ютерного зору

Хоча принципи стереоскопічних систем більш тісно відносяться до задач комп'ютерного зору, їх цілком можливо використовувати в системах просторового сканування, формуючи карти глибин сцени. Принцип роботи таких систем

					ІАЛЦ.467200.002 ТЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

базується на використанні двох камер, розташованих на певній відстані одна від одної. Аналізуючи зміщення однакових об'єктів на двох зображеннях, система визначає відстань до них (рис. 1.2.5).

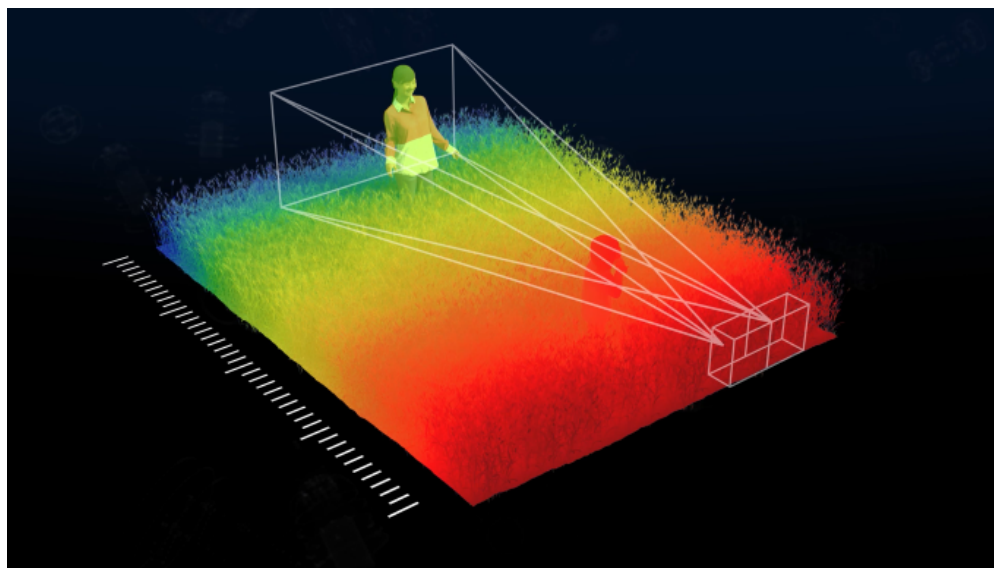


Рисунок 1.2.5 – Приклад роботи стереометричної системи

В основі визначення відстані у стереоскопічних системах лежить принцип триангуляції. Його суть полягає в тому, що один і той самий об'єкт спостерігається з двох різних точок простору. Через різне положення камер об'єкт проєктується на матриці зображень у різних місцях. Чим ближче об'єкт до камер, тим більше буде це зміщення, і навпаки — віддалені об'єкти матимуть менший зсув.

Математично глибина точки на сцені обчислюється за наступною формулою:

$$Z = \frac{fb}{d},$$

де Z – глибина, f – фокусна відстань камери, b – базис стереосистеми, або відстань між оптичними центрами камер, d – диспаритет – зміщення між проєкцією точки на одній камері порівняно з іншою. На рис. 1.2.6 зображене геометричне представлення принципу триангуляції, де диспаритету відповідає відстань між точками x та x' . У результаті обробки зображень з обох камер та математичних обчислень, на виході стереоскопічної системи формується зображення – теплова карта глибини (рис.1.2.7). На цій карті теплим відтінкам (червоний, жовтий) відповідають малі відстані до об'єктів, а холодним (зелений, блакитний, синій) – великі.

					ІАЛЦ.467200.002 ТЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

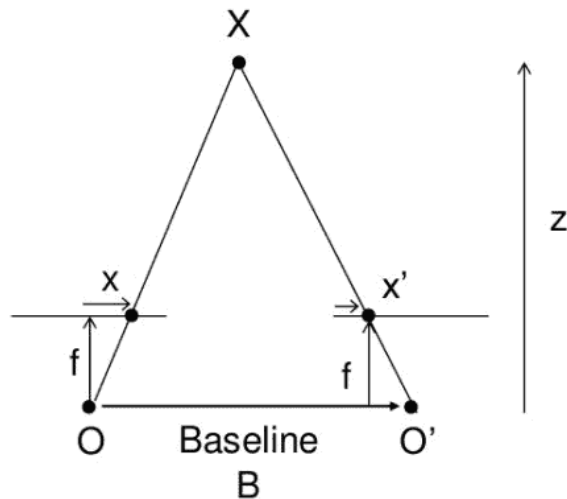


Рисунок 1.2.6 – Візуальне представлення принципів тріангуляції

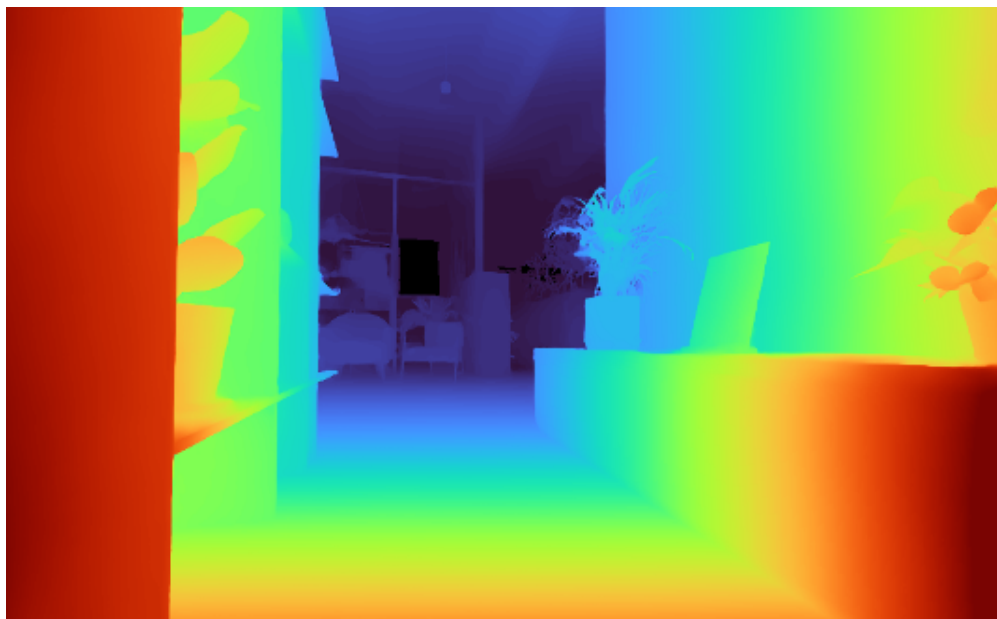


Рисунок 1.2.7 – Приклад роботи стереометричної системи

Основною перевагою стереоскопічних систем є можливість отримання просторової інформації без застосування активного випромінювання, що дозволяє зменшити енергоспоживання та спростити апаратну частину пристрою. Крім того, такі системи здатні одночасно отримувати як карту глибини, так і звичайне кольорове зображення сцени, що є важливим для задач навігації, робототехніки та комп'ютерного аналізу середовища. Також при достатніх обчислювальних потужностях, стереоскопічні системи можуть працювати в режимі реального часу.

Стереоскопічні системи активно використовуються у мобільній робототехніці,

					ІАЛЦ.467200.002 ТЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

автономному транспорті, безпілотних літальних апаратах та системах доповненої реальності. Прикладами готових рішень є Intel RealSense (рис.1.2.8) та інші системи стереозору, що поєднують декілька камер і алгоритми цифрової обробки зображень. Такі рішення дозволяють отримувати карту глибини сцени в режимі реального часу, що є критично важливим для задач навігації та уникнення перешкод. Завдяки розвитку апаратного прискорення та вбудованих SoC, стереоскопічні системи стають все більш доступними для вбудованих застосувань і компактних робототехнічних платформ.



Рисунок 1.2.8 – Intel RealSense

Незважаючи на переваги, стереоскопічним системам характерний ряд недоліків, подібних до недоліків фотограмметрії. По-перше, якість результатів роботи стереокамер прямо залежить від якості освітлення сцени, текстурованості об'єктів, відбиваючих властивостей та якості калібрування камер. Однотонні або слабкоструктуровані об'єкти часто призводять до значних похибок вимірювань. Цей недолік властивий будь-яким системам, що працюють на основі комп'ютерного зору, оскільки знаходження відстаней до точок відбувається методами машинного навчання, на відміну від, наприклад, ToF, де відстань обчислюється за допомогою формули та відомих параметрів швидкості світла.

Таким чином, стереоскопічні системи є ефективним засобом побудови карт глибин та просторового аналізу середовища, однак їх використання потребує достатньої обчислювальної потужності та сприятливих умов зйомки.

					ІАЛЦ.467200.004 ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

1.3 Обґрунтування теми дипломного проєкту та постановка задачі

У підрозділі 1.2 представлено аналіз основних принципів функціонування систем сканування простору. Порівняння основних характеристик описаних вище методів просторового сканування надаються нижче в таблиці 1.

Таблиця 1 – Порівняльні характеристики найбільш поширених методів просторового сканування

Характеристика	LiDAR-системи	ToF-сенсори	Фотограмметрія	Стереоскопія
Точність вимірювань	Висока	Середня	Висока	Середня
Формат результатів	Хмара точок	Відстань до точки (області)	Тривимірна текстурована модель	Карта глибин та зображення сцени
Вартість інтеграції	Висока	Низька	Висока	Середня
Вимоги до обчислювальних потужностей	Низькі	Низькі	Високі	Високі
Складність алгоритмічної обробки	Середня	Низька	Висока	Висока
Можливість використання у вбудованих системах	Так	Так	Ні	Так

Також проаналізовано основні технічні виклики при інтеграції та експлуатації систем просторового сканування для вбудованих систем, оскільки розуміння цих факторів дозволяє більш досконало оцінити ризики та уникнути потенційних помилок при проєктуванні. Зокрема, зосереджено значну увагу саме тим чинникам, які значно впливають на якість та ефективність роботи вбудованих систем. Результати порівняльного аналізу показані в таблиці 2.

Таблиця 2 – Особливості інтеграції різних систем просторового сканування

Характеристика	LiDAR-системи	ToF-модулі	Фотограмметрія	Стереоскопія
Апаратна інтеграція	Вимагає готового модуля, часто підтримку високошвидкісних інтерфейсів (Ethernet, CAN)	Проста інтеграція через поширені послідовні інтерфейси (I2C, UART, SPI)	Потрібна цифрова камера та стабільна механічна система, або багатокамерна конфігурація	Залежно від моделі, можлива інтеграція через послідовні інтерфейси, проте деякі вимагають таких інтерфейси, як USB, Ethernet, MIPI CSI-2
Налаштування	Готові драйвери	Готові драйвери	Калібрування камер, контроль освітлення сцени	Калібрування камер, контроль освітлення сцени
Обробка даних	При достатніх обчислювальних потужностях можлива на самій вбудованій системі	Проста, можлива навіть без інтелектуальних методів	Потребує повноцінного CV-пайплайна (SfM, MVS) та значних обчислювальних ресурсів	Для роботи в реальному часі вимагає середніх обчислювальних ресурсів GPU
Масштабування	Високе, можливість додавання IMU, GPS, комплексів з додаткових LiDAR-систем модулів тощо	Високе, легке впровадження додаткових модулів	Низьке	Високе для апаратної частини, низьке для алгоритмічної

Продовження таблиці 2

Характеристика	LiDAR системи	ToF-модулі	Фотограмметрія	Стереоскопія
Енергоефективність	Низька: потребує значного енергоспоживання лазерних модулів	Висока: низьке енергоспоживання сенсорів та мікроконтролерів	Низька: потребує високої електричної потужності як і для вимірювальних модулів (камер), так і для засобів інтелектуальної обробки	Середня: залежить від обчислювальної платформи та архітектури обробки даних
Гнучкість	Низька: замкнута архітектура, неможливість програмного розширення функціоналу, обмеженість у виборі апаратних засобів	Низька: початковий функціонал є спеціалізованим та обмеженим	Висока: можливість керування CV-пайплайном	Висока: можливість апаратних та програмних модифікацій

Проведений аналіз існуючих підходів до просторового сканування, а також оцінка особливостей їх апаратної та програмної інтеграції показали, що жоден із розглянутих методів не є універсальним рішенням для використання у компактних та недорогих вбудованих системах реального часу.

LiDAR-системи забезпечують високу точність та готові дані у вигляді хмари точок, однак їх застосування обмежується високою вартістю та складністю апаратної інтеграції. Фотограмметричні підходи дозволяють отримувати детальні тривимірні моделі, проте є непридатними для задач реального часу через значні обчислювальні витрати та складність обробки. ToF-датчики, у свою чергу, мають недостатню точність та обмежену інформативність для побудови повноцінного просторового представлення.

					ІАЛЦ.467200.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

Отримані результати аналізу свідчать про наявність проміжного класу задач, для яких необхідна система, що поєднує:

- низьку вартість реалізації;
- можливість забезпечення роботи в реальному часі;
- достатню точність для задач локальної навігації та виявлення перешкод;
- програмну гнучкість та можливість розширення функціоналу
- простоту інтеграції у вбудовані платформи з обмеженими ресурсами.

Отже, актуальною є розробка програмно-апаратних засобів для сканування простору, яка дозволить досягти балансу між низькою вартістю реалізації та простотою інтеграції, з одного боку, і високим рівнем контролю над алгоритмами обробки та формуванням просторового представлення — з іншого.

Для реалізації програмно-апаратного проєкту пропонується підхід, орієнтований на компенсацію основних обмежень існуючих рішень шляхом перенесення значної частини функціональності в програмний рівень. Це дозволить забезпечити розширюваність системи, можливість впровадження нових алгоритмів обробки даних та адаптацію під різні сценарії використання без зміни апаратної частини.

Програмно-апаратний модуль повинен складатися з двох частин:

- Апаратної – мікроконтролер, датчик, механічні елементи, пасивні електричні елементи, схема живлення;
- Програмної – прошивка, драйвери, користувацький інтерфейс, бізнес-логіка модуля.

					ІАЛЦ.467200.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

2. ПРОЄКТУВАННЯ АПАРАТНОЇ ЧАСТИНИ МОДУЛЯ

2.1 Вибір мікроконтролера

Для реалізації модуля сканування простору, що мав би переваги над існуючими рішеннями, складено список вимог, який містить наступні параметри:

- Вартість – головний недолік більшості описаних систем сканування простору;
- Енергоефективність – система має споживати мінімум енергії для забезпечення функціональності;
- Легкість впровадження – система має забезпечувати можливість комунікації з програмним забезпеченням обробки даних;
- Можливість роботи в реальному часі – передбачається функціонал, що підтримує потокову передачу даних, а отже необхідно забезпечити можливість роботи в реальному часі як для збору та обробки, так і передачі даних;
- Широкий вибір бібліотек для взаємодії з периферією.

Перед розробкою апаратної частини проводився аналіз найпопулярніших платформ мікроконтролерів, що використовуються для розробки пристроїв IoT та інших спеціалізованих комп'ютерних систем, які могли б задовольнити описаним вище вимогам та мати оптимальну обчислювальну потужність для коректного функціонування системи. У процесі аналізу розглядалися мікроконтролери сімейств AVR, STM32 та ESP32.

Серед мікроконтролерів AVR-архітектури розглянуто ATmega2560 (рис.2.1.1). Цей мікроконтролер є широко використовуваним у навчальних та прототипних системах завдяки простоті архітектури та великій кількості програмних бібліотек для взаємодії з периферією.

					ІАЛЦ.467200.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

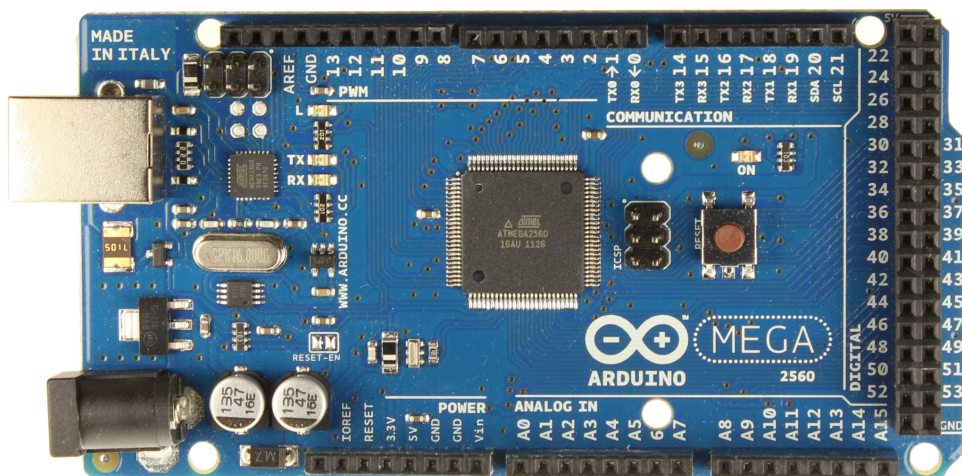


Рисунок 2.1.1 – Плата розробника Arduino MEGA з ATmega2560

ATmega2560 має наступний набір характеристик, що є актуальними в розрізі поставленої задачі:

- Ядро та архітектура: 8-бітний AVR RISC;
- Тактова частота: 16 МГц;
- Робоча напруга: 4.5 – 5.5 В;
- Флеш-пам'ять: 256 КБ (з них 8 КБ зарезервовані під завантажувач);
- Оперативна пам'ять (SRAM): 8 КБ;
- Сумісність з екосистемою Arduino.

Також мікроконтролер забезпечує підтримку найбільш поширених послідовних інтерфейсів, а саме: UART, I2C, SPI. Однак через 8-бітну архітектуру, низьку тактову частоту та обмежений обсяг оперативної пам'яті його продуктивність є недостатньою для ефективної обробки даних просторового сканування та роботи з потоковою передачею даних одночасно. Крім того, для підтримки мережевих інтерфейсів стеку протоколів TCP/IP, зокрема стандарту бездротових мереж IEEE 802.11, апаратна схема потребуватиме окремого співпроцесора для роботи з Wi-Fi, що створює додаткові вимоги до живлення та підвищує загальну вартість системи.

Розглядалися мікроконтролери сімейства STM32, зокрема STM32F411, побудований на базі ядра ARM Cortex-M4 (рис.2.1.2). мікроконтролер забезпечує

високу продуктивність, підтримує апаратний FPU, DMA та має широкий набір послідовних інтерфейсів. STM32F411 є ефективним рішенням для задач цифрової обробки сигналів та роботи в режимі реального часу. Крім того, платформа має нижче енергоспоживання та більш передбачувану роботу периферії у порівнянні з ESP32.

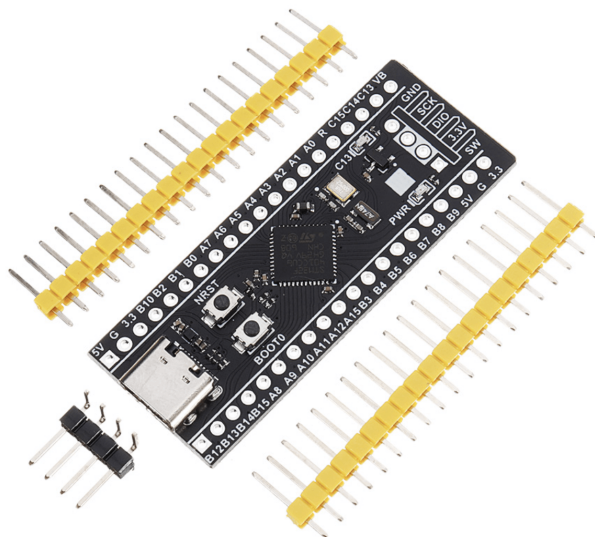


Рисунок 2.1.2 – Плата розробника STM32F411

Проте, незважаючи на переваги, сімейство мікроконтролерів STM32 вимагає глибокого розуміння особливостей роботи з периферією, налаштувань виводів, інтерфейсів та тактових генераторів. Крім того, процес розробки ускладнюється необхідністю роботи з низькорівневими налаштуваннями (HAL/LL або навіть регістровий рівень), що збільшує час розробки та підвищує ймовірність помилок на етапі конфігурації. Також слід враховувати відносно складну структуру екосистеми та меншу зручність швидкого прототипування у порівнянні з більш високорівневими платформами. Слід зазначити, що мікроконтролери сімейства STM32 не мають широкого поширення у вигляді рішень з вбудованим інтерфейсом Wi-Fi. Замість цього виробник орієнтується на використання зовнішніх модулів бездротового зв'язку або альтернативні технології, такі як Bluetooth Low Energy та LoRa. Це зумовлено високими вимогами до апаратних ресурсів Wi-Fi стеку та

складністю його інтеграції у мікроконтролерні системи реального часу.

Також, серед мікроконтролерів сімейства ESP32 знайдено модель ESP32 WROOM від виробника Espressif Systems (рис. 2.1.3). Цей контролер є одним із найбільш поширених модулів на базі ESP32, який поєднує мікроконтролер та радіочастотний модуль у єдиному компактному рішенні.

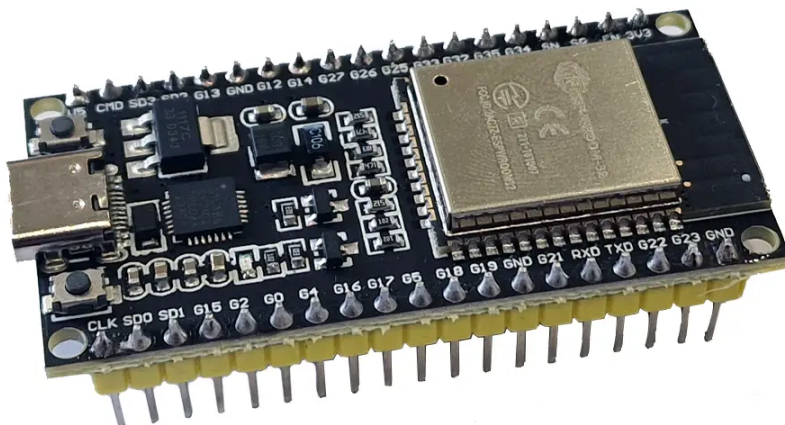


Рисунок 2.1.3 – Плата розробника на основі ESP32 WROOM

Він побудований на двоядерному 32-бітному процесорі Xtensa LX6 із тактовою частотою до 240 МГц, що забезпечує достатню продуктивність для обробки даних у реальному часі, включаючи роботу з сенсорами, фільтрацію сигналів та виконання мережових задач. Наявність вбудованих модулів Wi-Fi та Bluetooth робить його зручним для побудови систем з бездротовою передачею даних без необхідності використання додаткових компонентів. Крім того, мікроконтролер підтримує значну кількість послідовних інтерфейсів: UART, SPI, I2C. Також мікроконтролер має вбудовані мікросхеми АЦП, ЦАП, а також ШІМ-контролер. Варто зазначити, що сімейство ESP32 має активну підтримку спільноти, результатом чого є велика кількість різноманітних бібліотек та керівництв з розробки та відлагодження.

Однак, незважаючи на переваги та високу обчислювальну потужність, мікроконтролер має певні недоліки:

					ІАЛЦ.467200.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

- складна архітектура та велика кількість вбудованої периферії ускладнює загальну передбачуваність роботи, що може стати значною перешкодою при розробці вбудованих систем жорсткого реального часу, де затримки відгуку можуть мати критичні наслідки;
- при активному використанні модуля бездротового зв'язку, мікроконтролер використовує значну частку обчислювальних ресурсів, що також призводить до меншої стабільності та детермінованості виконання інших задач;

Проте, задача розробки модуля сканування простору як прототипа є задачею м'якого реального часу та допускає можливі затримки, результатом яких може бути знижена плавність передачі представлення сцени. Отже, найбільш оптимальним варіантом для розробки обрано мікроконтролер ESP32 WROOM, оскільки він має достатні обчислювальні потужності як для обробки даних з датчиків, так і потокової передачі; його використання дозволить не ускладнювати схему додатковими модулями, а також не призведе до значного підвищення вартості загальної розробки.

2.2 Вибір датчика відстані

Єдине та основне джерело просторової інформації у розроблювальній системі – це датчик відстані. Для вибору датчика сформовано наступні вимоги:

- Достатня частота вимірювань;
- Низька вартість;
- Низьке споживання енергії;
- Простота інтеграції з мікроконтролером;
- Невеликі габарити;
- Відносно висока точність вимірювань.

При виборі датчика виконувався аналіз найпопулярніших рішень для прототипування та комерційної розробки. Серед мікросхем для вимірювання відстані, що відповідали б вимогам низької вартості загальної розробки,

					ІАЛЦ.467200.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

на ринку представлені датчики на основі ToF технології, та ультразвукові датчики.

Ультразвукові датчики є одним із найбільш поширених та доступних засобів вимірювання відстані у вбудованих системах та робототехніці. Принцип їх роботи базується на випромінюванні ультразвукового імпульсу та подальшому вимірюванні часу його повернення після відбиття від об'єкта. На основі часу проходження сигналу визначається відстань до перешкоди. Формула обчислення відстані є ідентичною до ToF-датчиків, але обчислення виконуються на основі швидкості звуку. Серед ультразвукових датчиків найбільш популярним є мікросхема HC-SR04 (рис. 2.2.1).



Рисунок 2.2.1 – Ультразвуковий датчик відстані HC-SR04

Цей сенсор широко використовується у навчальних проєктах, робототехніці та системах автоматизації завдяки низькій вартості, простоті підключення та достатній точності вимірювання для базових задач. Датчик HC-SR04 має діапазон вимірювання приблизно від 2 до 400 см та забезпечує точність на рівні кількох міліметрів за сприятливих умов. Для роботи сенсор використовує два основні сигнали: Trigger — для запуску вимірювання, та Echo — для передачі імпульсу тривалості, пропорційної відстані до об'єкта.

					ІАЛЦ.467200.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		24

HC-SR04 має відносно невеликий кут огляду – 30° , має низькі вимоги до споживання енергії та дуже низьку ціну, що робить його привабливим рішенням для багатьох систем. Датчик не використовує послідовні інтерфейси для взаємодії з мікроконтролером, замість цього він має 2 виводи для запуску ультразвукової хвилі (TRIG) та отримання значення відстані у вигляді тривалості високого значення вихідного піну (ECHO). Часова діаграма роботи датчика HC-RS04 зображена на рисунку 2.2.2.

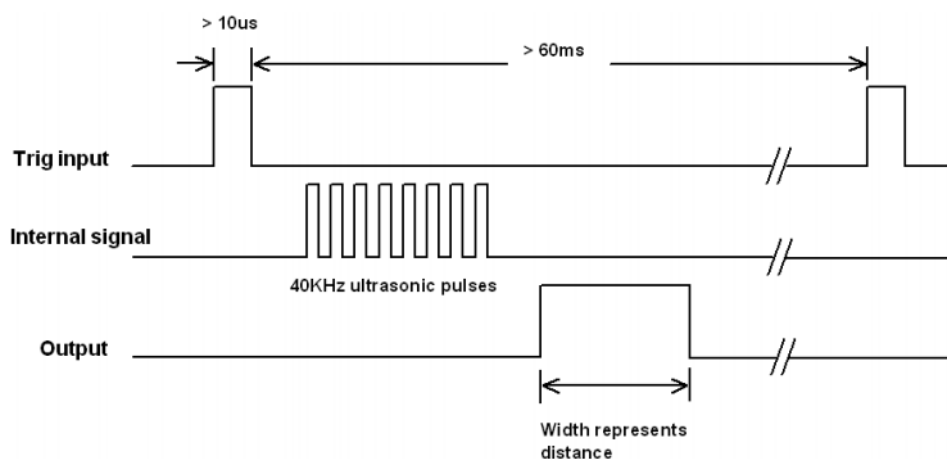


Рисунок 2.2.2 – Часова діаграма передачі значення відстані HC-SR04

Проте, особливості роботи датчика передбачають його роботу більше як датчика присутності об'єкта в області роботи, ніж для точних вимірювань, що робить його малопридатним для системи сканування простору. Це зумовлено низькою частотою вимірювання сигналів: для стабільного отримання відносно точних результатів, необхідно враховувати ехо від попередніх вимірювань, що призводить до того, що оптимальний час опитування датчика – раз на 50-60 мс. Крім того, точність роботи датчика залежить від властивостей повітря, матеріалів та кута падіння хвилі до поверхонь. Ці фактори можуть значно впливати на характеристики ультразвукової хвилі, що можуть значним чином спотворювати результати.

Серед ToF-датчиків, найбільш популярним рішенням для подібних проєктів

					ІАЛЦ.467200.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

є мікросхема VL53L0X (рис. 2.2.3).

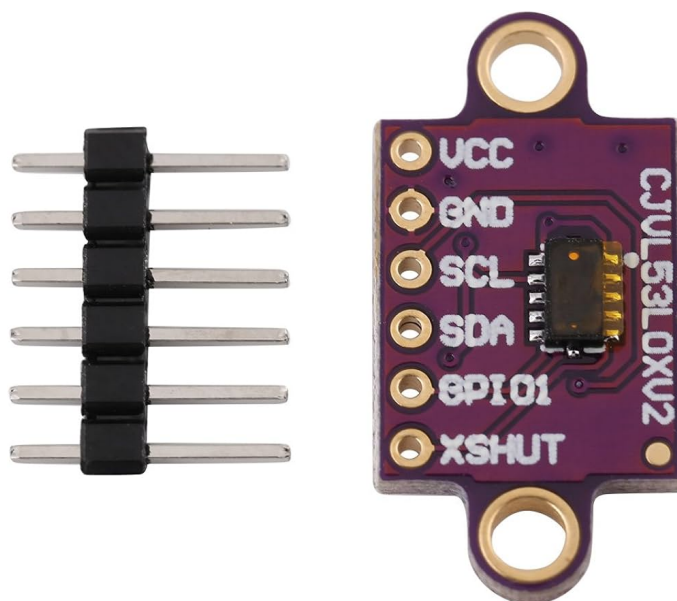


Рисунок 2.2.3 – VL53L0X

Ця мікросхема має компактні розміри – як правило, габарити друкованої плати цього модуля не перевищують розмірів 25*10 мм. Мікросхема живиться від джерела напругою 3 В та споживає не більш ніж 30 мА струму у режимі активної роботи. Модуль підтримує послідовний інтерфейс I2C, а також може забезпечувати частоту вимірювання до 50 Гц, або до 30 мс на одне вимірювання. Варто зазначити, що частота вимірювань є конфігурованою, та впливає на загальну точність вимірювання, тому можна зменшити час вимірювання для підвищення точності. Виробник зазначає, що проміжок відстаней, який може вимірювати модуль становить від 50 до 2000 мм.

При зазначених перевагах, також присутній ряд недоліків:

- невисока дальність вимірювань;
- якість вимірювань може знижуватись в залежності від умов освітлення;
- залежність від світловідбиваючих властивостей поверхні.

Таким чином, цей ToF-датчик забезпечує хороший баланс між точністю, швидкістю та складністю інтеграції, що робить його найбільш оптимальним рішенням у межах розробки дипломного проекту.

					ІАЛЦ.467200.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

2.3 Вибір механічних елементів

Для функціоналу зміни кута падіння лазера від датчика, необхідно забезпечити можливість обертання системи навколо горизонтальної та вертикальних осей. При виборі механічних елементів враховано наступні вимоги:

- низька вартість;
- можливість витримувати навантаження від корпусу та датчика;
- точність керування;
- низькі вимоги до енергоспоживання.

Серед елементів, що могли б відповідати зазначеним вище вимогам, розглянуто два способи механічного керування системою.

Крокові двигуни є електромеханічними пристроями, які перетворюють послідовність електричних імпульсів у дискретні кутові переміщення ротора (рис. 2.3.1). На відміну від класичних двигунів постійного струму, кроковий двигун обертається не безперервно, а фіксованими кроками, що дозволяє забезпечити високу точність позиціонування без необхідності використання складних систем зворотного зв'язку.

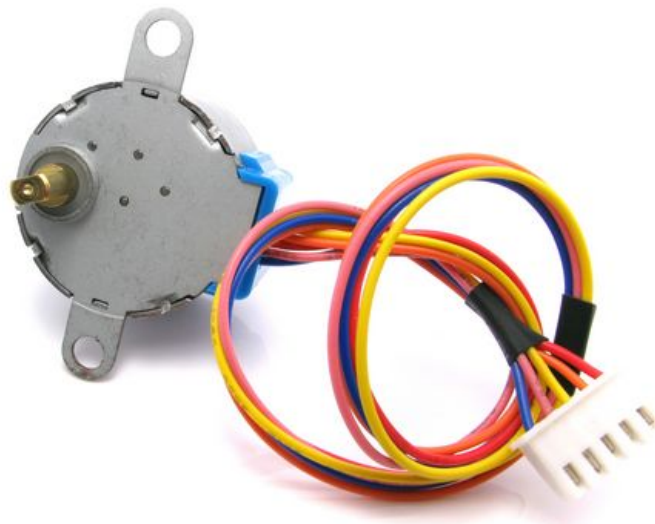


Рисунок 2.3.1 – Кроковий двигун

					ІАЛЦ.467200.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		27

У системах просторового сканування крокові двигуни застосовуються для керованого повороту сенсорного модуля з заданим кутовим кроком, що забезпечує формування регулярної сітки вимірювань. Це дозволяє отримувати стабільні та повторювані дані для побудови карти простору. Додатковою перевагою є можливість мікрокрокового керування, що підвищує роздільну здатність сканування та зменшує вібрації під час руху. При цьому крокові двигуни мають відносно хороші показники витримуваних навантажень.

Водночас, крокові двигуни мають відносно високі вимоги до живлення, а також потребують додаткову мікросхему для керування – драйвер (рис. 2.3.2). Це також підвищує споживання електроенергії, та ускладнює апаратну схему розробки. Крім цього, на високих швидкостях роботи на крокових двигунах спостерігаються “втрати” кроків, і ці втрати з часом тільки накопичуються, що може призводити до зниження точності. Для компенсації ефекту накопичення помилки, часто використовують схеми для калібрування пристроїв, наприклад через датчик Холла.

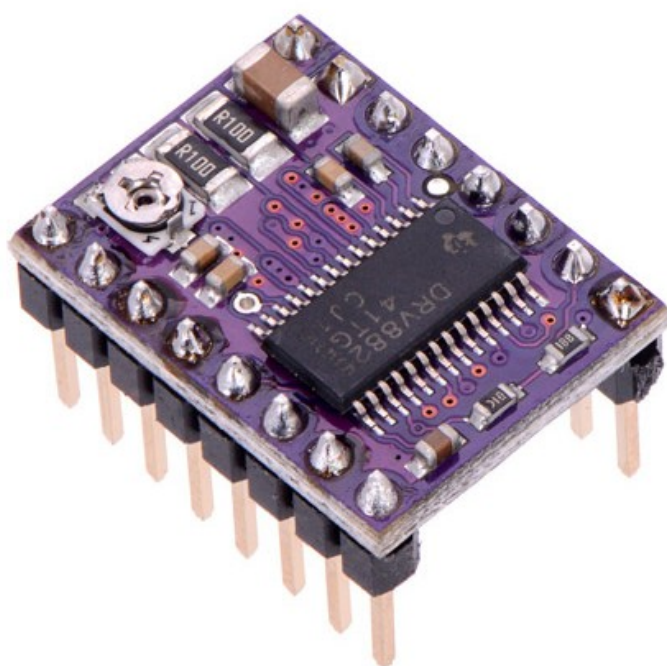


Рисунок 2.3.2 – Драйвер крокового двигуна

					ІАЛЦ.467200.004 ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

Гідною альтернативою кроковим двигунам є сервоприводи. Вони представляють з себе інтегровані електромеханічні системи, які забезпечують точне позиціонування вихідного вала за рахунок вбудованої системи зворотного зв'язку. Типовий сервопривід містить двигун постійного струму, редуктор, потенціометр або інший датчик положення, а також контролер, який регулює положення вала відповідно до вхідного керуючого сигналу.

Керування сервоприводів у більшості випадків виконується за допомогою ШІМ-сигналу, підтримка якого є у більшості сучасних мікроконтролерів, зокрема сімейства ESP32, що робить їх більш привабливим вибором з точки зору простоти електричної схеми.. Найбільш популярним серед сервоприводів для прототипування та нескладних механічних систем є модель SG90 (рис. 2.3.3).



Рисунок 2.3.3 – Сервопривід SG90

Цей сервопривід живиться від джерела напругою 4.2 В від 6 В, має обертальний момент до 1.8 кг, та споживає до 700 мА у пікових навантаженнях. Керування здійснюється ШІМ-сигналом, а також можлива регуляція кута повороту з кроком в 1 градус. Кут повороту цього сервопривода складає 180 градусів, а також він має низьку ціну.

Незважаючи на переваги, сервоприводи потребують додаткового захисту від перепадів напруги, їхня точність залежить від внутрішньої системи керування, та

					ІАЛЦ.467200.004 ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

та мають, як правило, нижчий обертальний момент, ніж крокові двигуни.

Отже, у результаті аналізу для механічної системи модуля сканування обрано сервопривід SG90. Його характеристики відповідають поставленим вимогам, їх використання не ускладнює електричну схему та вони споживають менше електроенергії у порівнянні з кроковими двигунами.

Для захисту від перепадів напруг біля кожного сервопривода необхідно додати електролітичні конденсатори, які допоможуть згладити різкі перепади споживання струму при старті та навантаженні, а також стабілізувати стрибки напруги при ввімкненні та вимкненні живлення.

Також, для сервоприводів SG90 існують готові корпуси, які дозволяють обертати систему в обох осях. Для розроблюваного модуля обрано кронштейн для керування камерою, до якого також можна приєднати ToF-датчик (рис.2.3.4).



Рисунок 2.3.4 – Кронштейн для сервоприводів SG90

2.4 Схема живлення

Незважаючи на те, що описані вище компоненти є достатньо енергоефективними, все одно варто зосередити увагу на безпечну подачу живлення. Наприклад, хоч сервоприводи споживають відносно небагато енергії,

					ІАЛЦ.467200.004 ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

навіть живлення одного привода від виводів контролера є небезпечним для обох мікросхем, а оскільки у проєкті використовується два сервоприводи, необхідно враховувати їх сумарне максимальне споживання енергії та обрати таке джерело, яке з надлишком зможе забезпечити необхідний струм на виході.

Для живлення обрано змішаний підхід: мікроконтролер живиться від ПК, до якого під'єднаний за допомогою USB; ToF-датчик живиться від вивода +3 В мікроконтролера; сервоприводи використовують окремий блок живлення з інтерфейсом USB-C напругою 5 В та струмом 3 А на виході. Для з'єднання джерела живлення сервоприводів використано плату-адаптер (рис. 2.4.1), яка виводить провідники живлення та землі, а також сигнали даних. На адаптері розпаяно штировий роз'єм 2.54 мм, виводи якого під'єднано на лінії живлення та землі макетної плати. Також між землею та живленням встановлено 2 електролітичні конденсатори ємністю 470 мкФ.

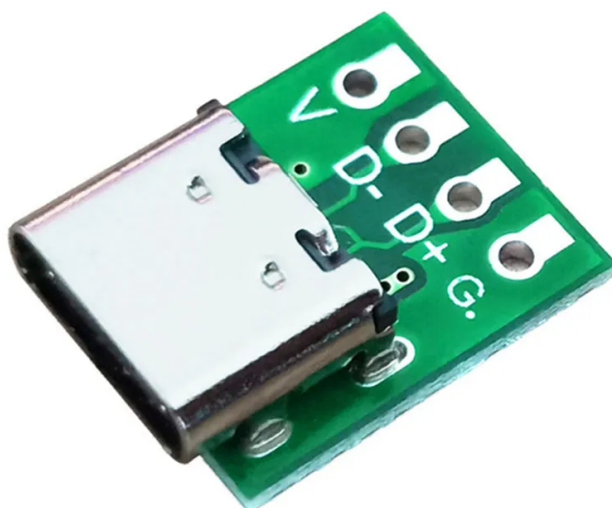


Рисунок 2.4.1 – Адаптер USB-DIP 2.54

					ІАЛЦ.467200.004 ПЗ	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		

3. РОЗРОБКА ПРОГРАМНОЇ ЧАСТИНИ МОДУЛЯ

3.1 Розробка прошивки мікроконтролера

3.1.1 План та послідовність етапів розробки

Перед безпосередньою розробкою програмного забезпечення прошивки мікроконтролера, здійснювалося формулювання переліку вимог. Розроблювана програмна частина модуля повинна:

- Бути абстрагованою від цільової платформи – логіка обробки та алгоритмів не є платформи-залежною, отже має бути ізольованою від деталей реалізації взаємодії контролера з периферією;
- Підтримувати можливість реалізації власних алгоритмів обробки даних та алгоритмів сканування;
- Підтримувати неблокуючу поведінку навіть для одноядерних контролерів або систем без планувальників задач;
- Не використовувати динамічне виділення пам'яті – в цілях оптимізації, забезпечення детермінованості виконання та сумісності з програмними платформами, які можуть не підтримувати операції з динамічним керуванням пам'яті.

Проектування кожної зі складових компонентів програмного забезпечення модуля складається з наступних етапів:

- Декомпозиція предметної області – розбиття компонентів системи на програмні об'єкти (структури) та забезпечити інтерфейс для їх використання;
- Проектування архітектури системи – розділення загальної програми на окремі модулі взаємодії з периферією, бізнес-логіки, та первинної обробки даних;
- Проектування шару абстракції над взаємодією з периферією (HAL);
- Проектування узагальненого керуючого об'єкту системи, що

					ІАЛЦ.467200.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		32

відповідатиме за ініціалізацію модуля сканування, введення конфігурації, функціонування системи під час виконання програми мікроконтролера;

- Проєктування неблокуючої моделі поведінки роботи системи;
- Проєктування розширюваної підсистеми первинної обробки даних;
- Проєктування моделі передачі даних для подальшої обробки.

Крім цього, важливим є вибір інструментів для розробки: у випадку вбудованих систем на мікроконтролерах вибір є доволі невеликим – лише мови, що забезпечують високу швидкодію, дозволяють низькорівневе керування пам'яттю та можуть працювати без runtime-оточення, або мають нативну реалізацію такого оточення виробником мікроконтролера. До таких мов програмування належать C/C++, Rust, MicroPython та деякі інші мови.

У випадку розроблюваної системи та з урахуванням списку вимог, найбільш придатним для використання є саме C, оскільки, наприклад, C++ має меншу підтримку серед платформ через складність проєктування трансляторів з підтримкою усього стандартного функціоналу мови; Rust є занадто молодого мовою програмування, тому має як обмежену підтримку різних платформ, так і обмежену кількість бібліотек для мікроконтролерів; інші мови програмування займають нішеві позиції у якості альтернативи C для конкретної платформи.

У випадку C, будь-яка існуюча архітектура та платформа має власний компілятор мови C, через простоту самої мови та через можливості низькорівневого керування системою; також приємним бонусом є сумісність синтаксису мов C та C++, що значно спрощує інтеграцію програмного коду.

3.1.2 Архітектура програмного забезпечення мікроконтролера

У результаті декомпозиції предметної області розроблюваного проєкту виділено наступні сутності:

- Пристрій сканування простору – головний та єдиний спосіб взаємодії цільової платформи з логікою керування;
- Пристрій сервопривода або іншого обертового компонента;
- Пристрій датчика відстані;

					ІАЛЦ.467200.004 ПЗ	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дата		

- Скінченна машина станів для різних режимів сканування;
- Стратегія обробки даних.

Мова С початково проєктувалася як високорівневе представлення мови асемблера і є суто процедурною мовою. Як наслідок з цього, проєкти мовою С можуть здаватися складними та заплутаними, особливо при порушенні принципів структурного програмування. Натомість, у багатьох сучасних мов, в тому числі призначених для системного програмування, однією з найпопулярніших парадигм є об'єктно-орієнтоване програмування.

Незважаючи на обмежений функціонал ядра мови С на рівні стандарту, існують способи відтворити основні концепції ООП. В процесі розробки програмного забезпечення використано саме такий підхід, що імітує об'єктно-орієнтовану архітектуру засобами мови С. Кожен функціональний компонент системи представлений у вигляді структури даних, яка інкапсулює внутрішній стан, а взаємодія з ним здійснюється через набір функцій-інтерфейсів.

Розглянемо, як приклад, об'єкт сервопривода (рис 3.1.1). Цей об'єкт має поле з позицією у градусах, яке є спільним для будь-якої реалізації пристрою для обертання системи. Поле `priv` є безтиповим вказівником, оскільки різні платформно-залежні реалізації мають різний набір полів даних для керування пристроєм.

Такий підхід дозволяє реалізувати принцип приховування, який є однією з основних архітектурних практик при проєктуванні програм мовою С. Також такий спосіб зберігання внутрішнього стану частково реалізує принцип успадкування, який, по суті, реалізовано через композицію.

```
struct cdss_servo {
    void *priv;
    int pos;
    struct cdss_servo_ops *ops;
};
```

Рисунок 3.1.1 – Об'єктне представлення сервопривода

Поле `ops` є структурою, яка містить в собі вказівники на функції, які мають

					ІАЛЦ.467200.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		34

роль інтерфейсу взаємодії з пристроєм (рис. 3.1.2). Це поле одночасно реалізує як поліморфізм, так і доповнює успадкування, а в комбінації з полем `priv` при правильному використанні реалізує і інкапсуляцію. Таким чином, сукупність цих підходів хоч і не у повній мірі, але дає можливість імітувати об'єктно-орієнтоване програмування засобами мови C, надаючи можливості для зручного та чистого проектування складних програмних систем, зниження зв'язності логічно відокремлених модулів, та підтримки можливості розширення функціоналу без “спагетті-коду”.

```
struct cdss_servo_ops {
    int (*init)(struct cdss_servo *servo, int axis);
    int (*destroy)(struct cdss_servo *servo);
    int (*move)(struct cdss_servo *servo, enum cdss_servo_dir dir, int value);
    int (*initial_position)(struct cdss_servo *servo);
};
```

Рисунок 3.1.2 – Інтерфейс взаємодії з пристроєм сервопривода

Залежна від платформи реалізація визначається в окремих C-модулях та відображається на структуру операцій (рис. 3.1.3). Таким чином, через єдиний інтерфейс програма може керувати будь-якими периферійними пристроями без розуміння їх реалізації (рис. 3.1.4).

```
struct cdss_servo_ops esp32_ops = {
    .init = esp32_servo_init,
    .destroy = esp32_servo_destroy,
    .move = esp32_servo_move,
    .initial_position = esp32_servo_initial_position
};
```

Рисунок 3.1.3 – Реалізація HAL для ESP32

```
static inline void cdss_servo_move(struct cdss_servo *servo, enum cdss_servo_dir dir, int value) {
    if (servo->ops && servo->ops->move)
        servo->ops->move(servo, dir, value);
}
```

Рисунок 3.1.4 – Виклики платформно-залежного коду з єдиним інтерфейсом

Інші об'єкти системи також використовують цей підхід, що підвищує

					ІАЛЦ.467200.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

читабельність коду. Варто зазначити, що багато сучасних проєктів з великою спільнотою та активною підтримкою розробників також використовують “ручне ООП”. Найбільш популярним прикладом використання цього підходу є ядро ОС GNU Linux, особливо драйверна модель.

Проектування неблокуючої логіки роботи системи передбачає можливість конкурентної (псевдопаралельної) роботи на платформах, які не використовують ОСРВ. Для цієї мети найбільш оптимальним підходом виявилось проектування системи з кооперативною багатозадачністю, де задача сама вирішує, коли повертати керування в головний цикл. Найпростіший та найбільш ефективний спосіб забезпечити таку поведінку є використання скінченного автомата (машини станів). За один крок задача виконує одну дію, визначає стан автомату для наступного такту роботи, та повертає керування головному циклу.

Як приклад, розглянемо реалізацію машини станів режиму двовимірного сканування поверхні. Скінченні автомати режимів сканування мають наступний інтерфейс (рис. 3.1.5). Вони підтримують операції встановлення початкового та обробки поточного стану.

```
static int cdss_device_3d_handle_state(void *vdev) {
    struct cdss_device *device = (struct cdss_device*)vdev;
    if (device->state_machine.state >= MAX_STATE) {
        cdss_log(LOG_DEBUG, msg: "Invalid state machine value %d\n", device->state_machine.state);
        return -1;
    }
    return handlers[device->state_machine.state](device);
}

static struct cdss_mode_ops ops_3d = {
    .next = cdss_device_3d_handle_state,
    .reset = reset
};
```

Рисунок 3.1.5 – API скінченного автомату

Кожен стан автомату представляє собою функцію-обробник, задачами якої є реалізація бізнес-логіки обробки поточного стану та встановлення наступного стану системи для наступного такту роботи. Приклад реалізації обробника стану зображено на рисунку 3.1.6. Робота таких обробників передбачає якомога швидше

					ІАЛЦ.467200.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

виконання, тому у цьому прикладі при отриманні некоректного значення відстані (що сигналізує про те, що датчик ще не готовий повернути нові дані), керування повертається до точки виклику, а стан автомату залишається незмінним.

```
static int handle_read_sensor(struct cdss_device* device) {
    int distance = cdss_distance_sensor_get_distance(device->distance_sensor);
    if (distance > 0) {
        push_point(device,
            (struct cdss_point) {
                .r = distance,
                .theta = device->x_servo->pos,
                .phi = device->y_servo->pos }
        );
        cdss_device_sm_next_state(&device->state_machine, state: MOVE_HORIZONTAL);
    }
    return 0;
}
```

Рисунок 3.1.6 – Обробник стану “Зчитати дані з датчика”

Система обробки даних використовує запозичений та модифікований з об’єктно-орієнтованих патернів проєктування Стратегія (рис.3.1.7).

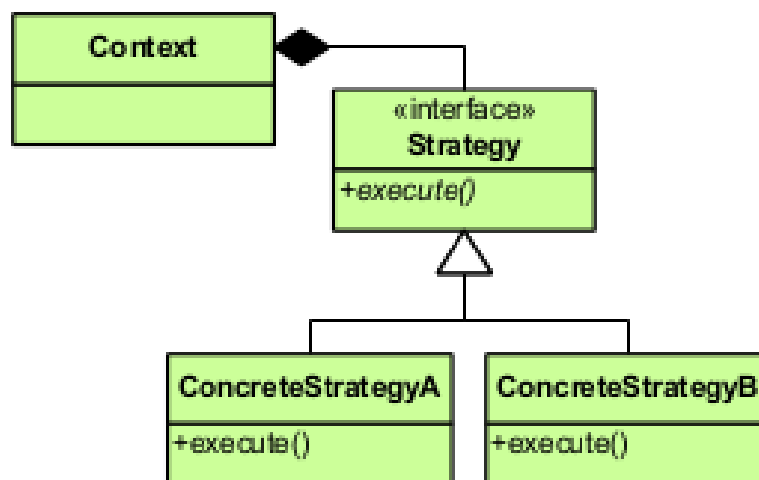


Рисунок 3.1.7 – Патерн стратегія

Суть патерна полягає в тому, що певний алгоритм представляється у вигляді об’єкта, а не функції. Окремі складові цього алгоритма, які можуть відрізнятися, розбиваються на абстрактні функції, і кожна з реалізацій стратегії реалізує власний спосіб виконання складових алгоритма. У випадку проєктування системи обробки даних, реалізовано лише стратегію для видалення аномальних точок, проте

інтерфейс модуля передбачає можливість розширення можливих операцій та їх користувацьку реалізацію. Також виконана модифікація патерну, яка полягала в тому, що внутрішній стан “контексту” перенесено до самого об’єкту стратегії шляхом використання безтипового вказівника, а власне виконання повного алгоритму реалізовано у вигляді функції (рис. 3.1.8).

```
static inline void cdss_point_processing_strategy_execute(struct cdss_point_processing_strategy *strategy,
                                                         struct cdss_point *points,
                                                         size_t num_points,
                                                         size_t start_idx) {
    if (strategy->filter_anomalies) strategy->filter_anomalies(strategy, points, num_points, start_idx);
    /* TBD: add new strategy implementations */
}
```

Рисунок 3.1.8 – Код виконання алгоритму зі стратегією

Проектування системи збереження масиву точок та передачу їх мережею передбачало забезпечення абстракції від способу збереження точок в об’єкті модуля сканування, а також створення зручного інтерфейсу для потокової передачі. Це досяглося використанням кільцевого буфера, доступ до якого здійснюється за допомогою інтерфейсної функції, що дозволяє отримати усі точки, що вже оброблені та готові до відправки (рис. 3.1.9).

```
/* Continuous chunk */
if (device->start_of_row >= device->read_ptr) {
    elems_to_copy = MIN(device->start_of_row - device->read_ptr, out_len);
    memcpy(out, &device->txbuf[device->read_ptr], Size: elems_to_copy * sizeof(struct cdss_point));
    device->read_ptr = (device->read_ptr + elems_to_copy);
    sum = elems_to_copy;
}
/* Non-continuous chunk:
 * [data_to_copy], ..., (data_to_copy)*/
else {
    elems_to_copy = MIN(TXBUF_SIZE - device->read_ptr, out_len);
    memcpy(out, &device->txbuf[device->read_ptr], Size: elems_to_copy * sizeof(struct cdss_point));
    p = out + elems_to_copy;
    sum = elems_to_copy;

    elems_to_copy = MIN(device->start_of_row, out_len - sum);
    memcpy(p, device->txbuf, Size: elems_to_copy * sizeof(struct cdss_point));
    device->read_ptr = elems_to_copy;
    sum += elems_to_copy;
}
return sum;
```

Рисунок 3.1.9 – Код функції отримання готових до відправки точок

					ІАЛЦ.467200.004 ПЗ	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		

3.1.3 Реалізація мережевої взаємодії

Мережева підсистема має платформно-залежний характер та для різних платформ має різний інтерфейс, та залежно від потреб користувача дані від модуля можуть навіть не мати потреби передаватися мережею, тому вирішено використовувати мережеві функції цільової платформи мікроконтролера, у якості прикладу реалізації клієнтського коду взаємодії контролера з системою сканування.

У ході розробки визначено, що найбільш ефективним способом передачі даних з датчика буде використання локальної бездротової мережі Wi-Fi. Відправка даних через мережу здійснюється через протокол транспортного рівня UDP.

Цільова платформа цього проєкту використовує високорівневний API для взаємодії з TCP/IP стеком, які надаються платформою Arduino. Це зручна користувацька обгортка поверх низькорівневних бібліотек для мікроконтролерів, наприклад `lwIp`, а також SDK виробників контролерів. Вона дозволяє не зосереджуватись на реалізації драйверів, організації паралельної роботи та керування пам'яттю; замість цього вона дозволяє зосередитися безпосередньо на бізнес-логіці передачі даних. Ініціалізація мережевої підсистеми системи виглядає наступним чином (рис. 3.1.7). У ході ініціалізації відбувається з'єднання з безпроводною мережею та відкриття UDP порту для прослуховування. Використання високорівневого способу взаємодії з мережею дозволяє уникнути збільшення кількості коду, деталей роботи стандарту IEEE 802.11 а також спростити використання пам'яті, оскільки бібліотека бере на себе ці задачі.

```
WiFi.begin(SSID, WPA2_PASSWORD);
Serial.println("Trying to connect");
while (retries-- && WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
if (!retries) {
    Serial.print("W-Fi connection timeout: failed to connect to ");
    Serial.println(SSID);
    return -1;
}
udp.begin(LOCAL_PORT);
```

Рисунок 3.1.7 – Ініціалізація мережевої підсистеми

Структура пакету має наступний вигляд (рис. 3.1.8). Пакет містить в собі кількість точок, межі для кутів вертикальної та горизонтальної осей градусах від 0 до 180, а також сам масив точок. Отримання масиву точок та кількості цих точок відбувається через згаданий вище інтерфейс об'єкта модуля сканування, а границі кутів є полями структури самого пристрою.

```
#pragma pack(push, 1)
struct points_packet {
    int cnt;
    int lim_x;
    int lim_y;
    struct cdss_point points[MAX_NUM_POINTS];
} pkt;
#pragma pack(pop)
```

Рисунок 3.1.8 – Структура UDP дейтаграми для передачі масиву точок

Структура конфігурації системи має наступний вигляд (рис. 3.1.9). Усі поля дозволяють динамічну зміну протягом роботи системи, не потребуючи перезапуску або перепрошивки. Ця ж структура використовується для передачі динамічної конфігурації з ПК до контролера, також за допомогою UDP пакетів.

```
struct cdss_device_runtime_config {
    int step;
    int angle_lim_x;
    int angle_lim_y;
    enum cdss_operating_mode mode;
};
```

Рисунок 3.1.9 – Структура UDP дейтаграми для передачі конфігурації

Невід'ємною частиною будь-якої програмної системи є можливість відслідковувати стан системи та коректність її функціонування. Для цього використовуються спеціальні діагностичні повідомлення. Оскільки способи виведення діагностичної інформації є платформо-залежною, вирішено винести функціонал логування в HAL-модуль системи.

					ІАЛЦ.467200.004 ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

3.1.4 Логування подій програмної частини

Розроблене програмне забезпечення підтримує 5 рівнів докладності логування, серед яких рівень помилок, попереджень, інформації, розширеної інформації та відлагодження. Використання інтерфейсу для виведення діагностичних повідомлень передбачає передачу рівня докладності та форматного рядка з відповідними аргументами, аналогічно до використання функції printf() стандартної бібліотеки мови С. Формат виведеного повідомлення містить в собі мітку з іменем модуля, рівень докладності та саме повідомлення (рис. 3.1.10). Такий функціонал дозволяє спростити обробку логів подій, наприклад, розділення повідомлень з різних пристроїв у різні сховища. Також користувач може обмежити рівень виведення повідомлень під час збірки проєкту за допомогою визначення макроса CDSS_LOGMASK.

```
[CDSS][INFO]Reconfig: step=1, x_lim=90, y_lim=90, mode=1
[CDSS][DEBUG]Reset device to apply new configuration...
[CDSS][INFO]Reconfig: step=1, x_lim=90, y_lim=20, mode=1
[CDSS][DEBUG]Reset device to apply new configuration...
[CDSS][INFO]Reconfig: step=2, x_lim=90, y_lim=20, mode=1
[CDSS][DEBUG]Reset device to apply new configuration...
```

Рисунок 3.1.10 – Приклад виведення діагностичної інформації модуля

3.2 Розробка програмного забезпечення для візуалізації роботи модуля

3.2.1 Реалізація сервера для взаємодії з модулем сканування

Рациональним рішенням для проєктування системи візуалізації є використання клієнт-серверної архітектури, де клієнт відповідає за візуалізацію, ввід даних від користувача та відправку до сервера, а сервер керує мережевим з'єднанням з контролером, виконує обробку (опціонально) даних, та передає дані клієнту.

Для сервера визначено наступний функціонал:

- Встановлення з'єднання з мікроконтролером;
- Прийом та передача пакетів даних з мікроконтролера, передача

					ІАЛЦ.467200.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		41

конфігурації;

- Обробка даних;
- Передача даних на клієнт візуалізації.

Для реалізації серверної частини використано мову Python. Цю мову обрано через велику кількість бібліотек та багатий функціонал, що надається інструментами мови та стандартною бібліотекою. Також мова Python має низький поріг входу як для програмування, так і для розгортання серверних додатків, а також підходить для швидкого та ефективного прототипування.

Встановлення з'єднання з мікроконтролером виконується за допомогою відкриття UNIX-сокета. Сокет – це абстрактний файл, який представляє мережеве з'єднання та підтримує простий інтерфейс отримання та передачі даних. Сервер керує як передачею конфігурації на мікроконтролер, так і має в собі цикл опитування дейтаграм (рис. 3.2.1).

```
def reconfig(self, data): 1 usage
    packet = struct.pack( fmt: "<iiii", *v: int(data["step"]), int(data["x_lim"]), int(data["y_lim"]), int(data["mode"]))
    self.sock.sendto(packet, (MCU_IP, MCU_PORT))

def udp_loop(self, handler): 1 usage
    while True:
        count = 0
        data, addr = self.sock.recvfrom(4096)

        cnt, lim_x, lim_y = struct.unpack_from( format: "<iii", data, offset: 0)
        offset = 12
        points = []
        for i in range(cnt):
            r, theta, phi = struct.unpack_from( format: "<iii", data, offset)
            points.append(model.Point(r, theta, phi, lim_x, lim_y, default_processor))
            count = count + 1
            offset += 12

        handler("{\\"cnt\\": %s, \\"lim_x\\": %s, \\"lim_y\\": %s, \\"points\\": %s}" %(count, lim_x, lim_y, points.__repr__()))
```

Рисунок 3.2.1 – Організація обміну даних з мікроконтролером

У циклі опитування UNIX-сокета відбувається парсинг пакету з мікроконтролера, формування json-рядка та його передача на програмне забезпечення візуалізації через Websocket. Тут може відбуватися збір даних в один масив та обробка більш складними алгоритмами, наприклад, кластеризація точок або інші математичні операції, як наприклад виділення границь об'єктів.

					ІАЛЦ.467200.004 ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

3.2.2 Реалізація інтерфейсу користувача

Для клієнтського програмного забезпечення візуалізації визначена наступна функціональність:

- Візуалізація точок з мікроконтролера;
- Можливість огляду візуалізації з різних ракурсів;
- Встановлення з'єднання з серверною частиною, з можливістю передавати дані в обидві сторони;
- Користувацький інтерфейс для передачі конфігурації на серверну частину.

Для задоволення потреб, описаних вище, найбільш оптимальним рішенням обрано використання веб-орієнтованого інтерфейсу. Такий підхід дозволяє спростити верстку користувацького інтерфейсу, відображення даних та організацію мережевої взаємодії. Відповідно до обраного підходу, єдиними можливими технологіями є стандартні браузерні інструменти – HTML5, CSS, та JavaScript. Це відкриті стандарти, що підтримуються усіма сучасними браузерами, не залежать від клієнтської платформи та дозволяють швидко та ефективно проєктувати інтерактивні користувацькі інтерфейси, на відміну від технологій для розробки настільних додатків, таких як QT, JavaFX, WinForms тощо. Для мережевої взаємодії з серверною частиною обрано технологію Websocket – вона дозволяє передавати дані в обидві сторони, та підтримується усіма сучасними браузерами. Websocket надає ідентичний UNIX-сокетами інтерфейс, дозволяючи асинхронний обмін даними.

Оскільки список виконуваних задач клієнтської частини є невеликим, а також є потреба у швидкому прототипуванні, вирішено не використовувати Frontend-фреймворки, оскільки такий функціонал легко досягти стандартними засобами, а також оптимізувати роботу користувацького інтерфейсу за рахунок відсутності накладних витрат з боку фреймворку.

Для візуалізації обрано Javascript бібліотеку THREE.js. Це популярна кросс-браузерна бібліотека з відкритим вихідним кодом для відображення тривимірної графіки прямо на веб-сторінці. На відміну від стандартних браузерних

					ІАЛЦ.467200.004 ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

засобів програмування, таких як маніпуляції з HTML елементом canvas, який використовує CPU для обчислень графіки, ця бібліотека використовує специфікацію WebGL, яка дозволяє задіяти обчислювальні потужності графічного прискорювача комп'ютера, забезпечуючи високу швидкодію для відображення складних з точки зору паралельних обчислень сцен. Вона має зручний API, підтримує використання шейдерів та впровадження тривимірної анімації. Завдяки їй, зникає необхідність використовувати сторонні плагіни браузерів, а код, написаний з використанням цієї бібліотеки працює на будь-яких браузерах, що підтримують HTML5.

Отже, розроблено:

- Прошивку мікроконтролера з підтримкою функцій, описаних у вимогах до програмного забезпечення контролера;
- Сервер для організації передачі даних між мікроконтролером та клієнтом візуалізації;
- Клієнтський веб-сервіс для візуалізації хмари точок з контролера у тривимірному вигляді.

					ІАЛЦ.467200.004 ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

4. ТЕСТУВАННЯ МОДУЛЯ СКАНУВАННЯ

4.1 Запуск модуля сканування

Для коректного запуску необхідно виконати наступні дії:

- Запустити сервер обробки даних;
- Відкрити веб-сторінку з наступним URL: localhost:8080/static;
- Подати живлення на сервоприводи;
- З'єднати контролер та ПК.

При коректному запуску сервера, при переході на веб-інтерфейс є можливість спостерігати координатні осі та меню конфігурації (рис. 4.1.1).

Наступним кроком подається живлення на сервоприводи шляхом з'єднання блоку живлення з платою-адаптером. В кінці подається живлення на мікроконтролер.

Під час запуску, після ініціалізації модуля, сервоприводи мають зайняти початкове положення, а термінал, на який виводяться дані з UART, має показувати відповідні повідомлення (рис. 4.1.2). При успішному запуску та підключенню мікроконтролера до Wi-Fi мережі, графічний інтерфейс забезпечує відображення точок сканування в реальному часі.

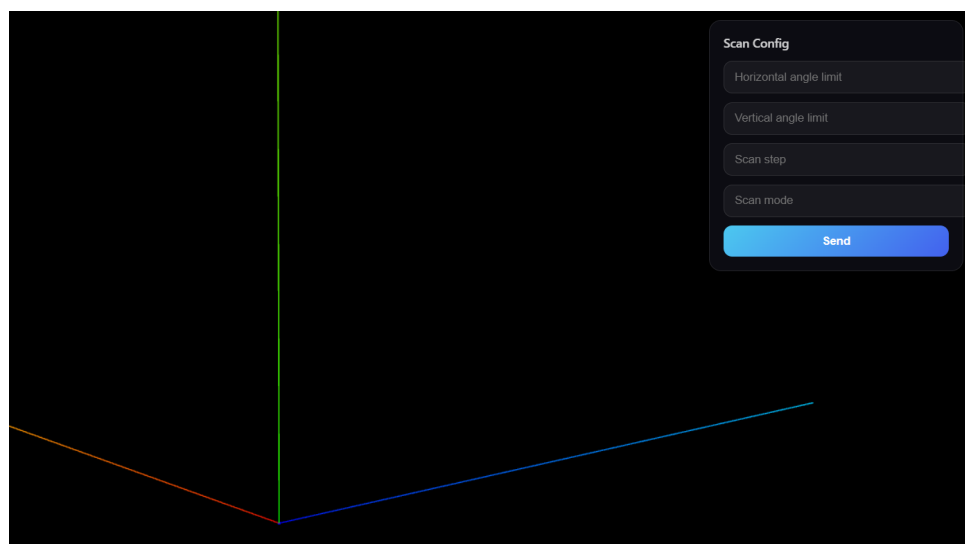


Рисунок 4.1.1 – Вигляд веб-інтерфейсу

					ІАЛЦ.467200.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45

```
[CDSS][INFO LVL2]Initializing servos...
[CDSS][INFO LVL2]Initializing distance sensor...
[CDSS][INFO LVL2]Initializing state machine with mode SCAN 3D...
[CDSS][INFO LVL2]Initializing point processing strategy with mode default...
[CDSS][INFO]Device initialization succeeded!
```

Рисунок 4.1.2 – Вивід даних з UART при успішному запуску

4.2 Тестування простої сцени

Головною метою модуля датчика просторового сканування є надання інформації про просторове середовище, співвідношення об'ємів та глибин поверхонь сцени, тому для тестування необхідно підготувати такі умови, де це співвідношення можна буде оцінити візуально.

У якості прикладу створено тестову сцену (рис. 4.2.1). Сцена складається з прямокутного об'єкту, який відхилено від центру, та стіни у якості фону.

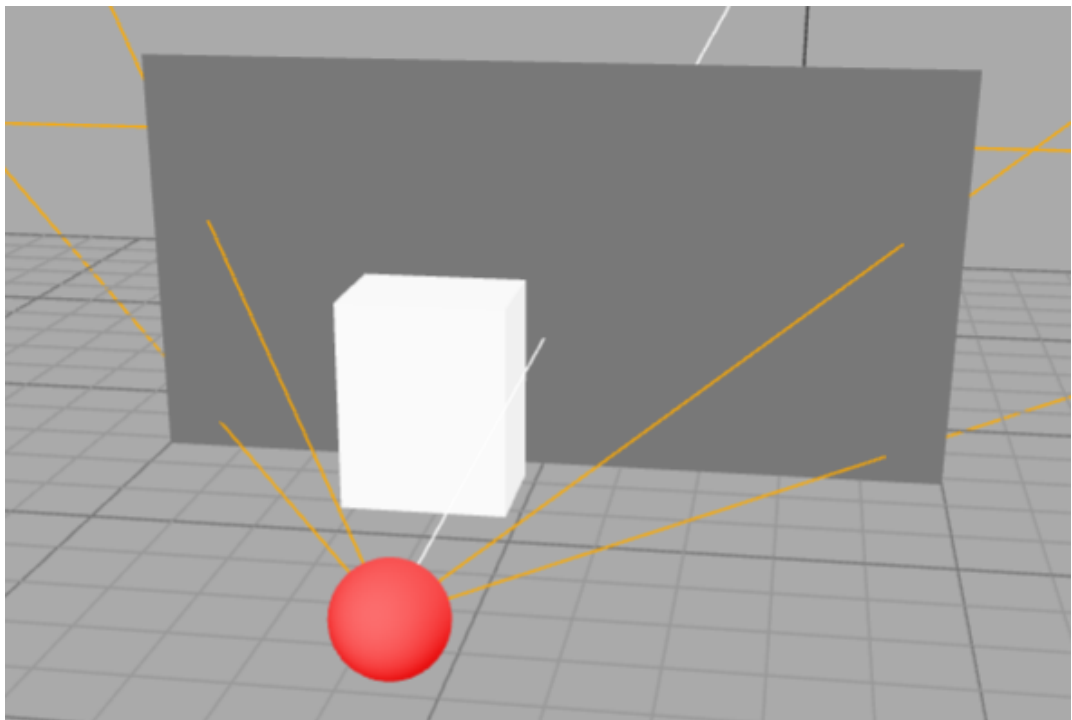


Рисунок 4.2.1 – Схематичне зображення тестової сцени

Запустимо систему сканування з параметрами за замовчуванням та залишимо її працювати на певний час. Після завершення деякого проміжку часу веб-інтерфейс відобразить отриману хмару точок (рис. 4.2.1). Як видно,

					ІАЛЦ.467200.004 ПЗ	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

зображення дає чітке розуміння, що сцена складається з плоского фону (стіни), та певного об'єкту, який має меншу глибину. Додатково програма візуалізації підбирає кольори для відображення точок в залежності від їх віддаленості від модуля сканування. Тому на хмарі точок видно область фіолетового кольору.

Через особливості роботи обраного датчика, описані в розділі 1.2.2, спостерігається висока плавність переходів між площинами, що зумовлюється кутом огляду та апаратними обмеженнями мікросхеми. Однак, оскільки програмне забезпечення мікроконтролера дозволяє використовувати будь-яке інше апаратне забезпечення, високу плавність можна усунути, використавши датчик з можливістю вимірювання відстані у вузькій області або ж додавши інші стратегії обробки даних.

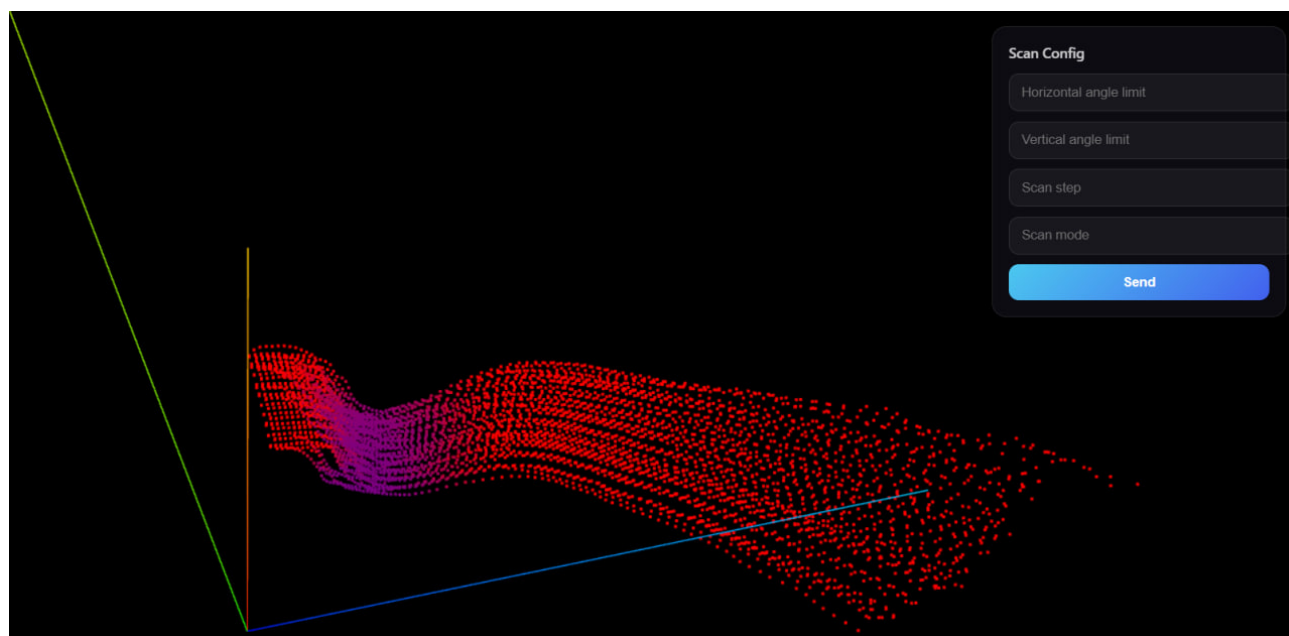


Рисунок 4.2.2 – Хмара точок, отримана в результаті роботи модуля

4.3 Тестування сцени з декількома об'єктами

Для наступної сцени використаємо 2 прямокутних об'єкти, а також змінимо кут, під яким система сканування розташована відносно стіни (рис. 4.3.1). Така сцена також дозволить оцінити, як змінюються отримані результати сканування та як інтерпретуються результати для об'єктів з різною відстанню до модуля.

					ІАЛЦ.467200.004 ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

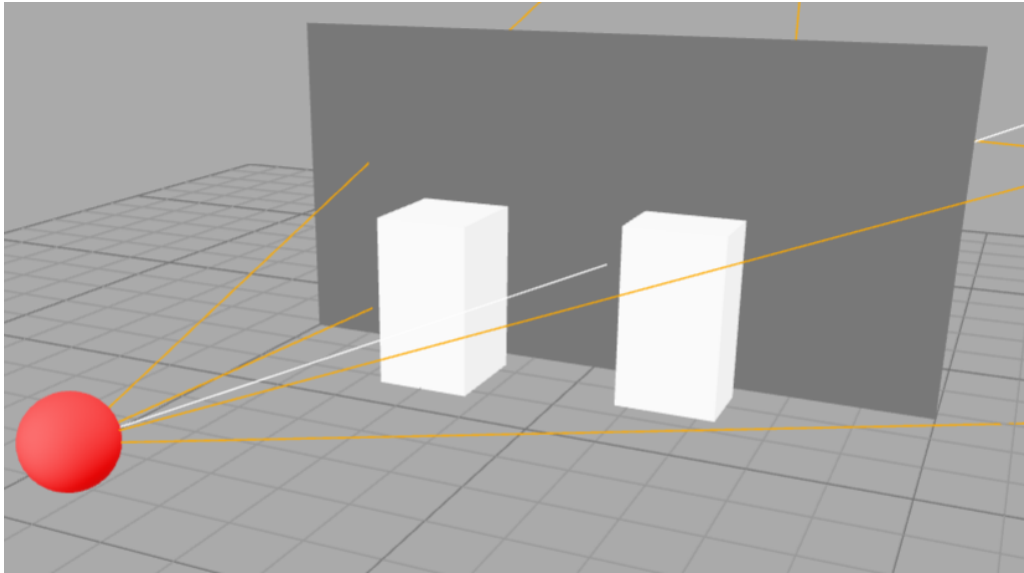


Рисунок 4.3.1 – Сцена з декількома об’єктами

Запустимо сканування та переглянемо на веб-сторінці результати роботи після завершення хоча б одного повного циклу сканування. Як результат, можна побачити (рис. 4.3.2), що хмара точок має дві чітко виражені області з меншою відстанню до модуля сканування. Таким чином, можна стверджувати, що функціонування модуля надає достатньо достовірні дані для їх подальшої алгоритмічної обробки.

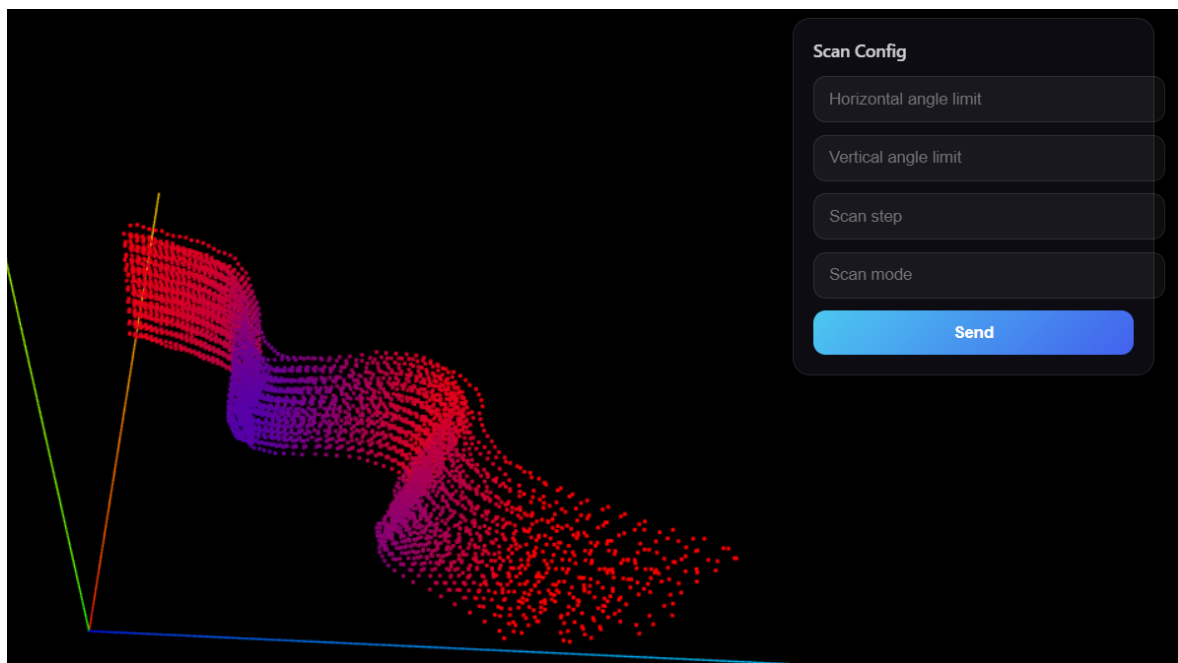


Рисунок 4.3.2 –Хмара точок для сцени з двома об’єктами

					ІАЛЦ.467200.004 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48

4.4 Тестування режиму одновимірного сканування

Для наступного етапу тестування можна залишити таку ж сцену з двома об'єктами. Для запуску одновимірного режиму сканування необхідно в меню конфігурації веб-інтерфейсу обрати опцію “1D scan”. Запустимо цей режим та переглянемо результати візуалізації (рис. 4.4.1). Вони показують, що вертикальна координата є сталою, отже режим працює коректно. Цей режим може використовуватися системами, що не потребують обробки просторових даних з урахуванням висоти. Його перевагою є більша швидкість роботи через меншу кількість необхідних ітерацій сканування.

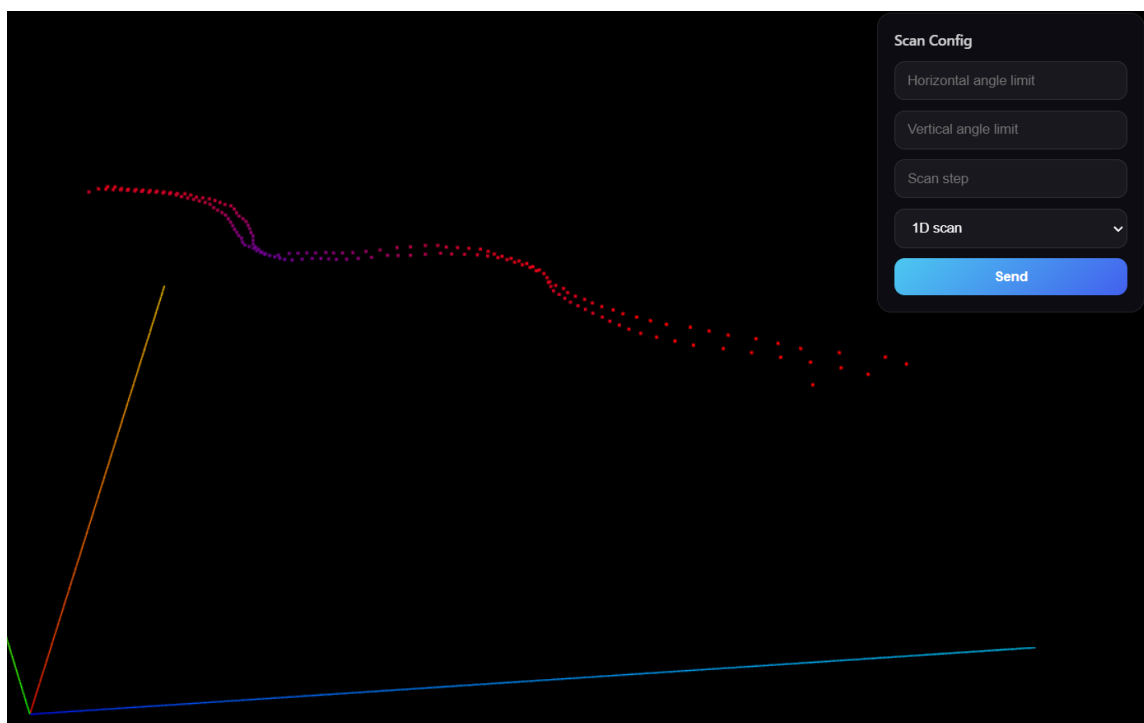


Рисунок 4.4.1 – Візуалізація в режимі одновимірного сканування

Отже, у ході тестування виконано запуск програмно-апаратного модуля, отримано візуалізацію результатів роботи модуля, продемонстровано виконання основних його функцій та можливостей, а також підтверджена коректність роботи розробленого модуля.

ВИСНОВКИ

У ході виконання дипломного проєкту розроблено та реалізовано програмно-апаратний модуль просторового сканування на основі ToF-датчиків з підтримкою зміни конфігурації через мережу Wi-Fi.

Основною метою проєкту є створення модуля, що забезпечує функціональність, співставну до промислових систем, проте з меншою собівартістю та енерговитратністю. Розробка модуля є актуальною задачею.

Основні вимоги до розробки програмно-апаратного модуля в результаті аналіз існуючих підходів та систем сканування простору.

Розробка модуля складалася з проєктування як апаратного, так і програмного забезпечення. Після оцінки різних варіантів апаратної конфігурації обрано оптимальні для виконання поставлених вимог компоненти, а саме:

- мікроконтролер ESP32 WROOM – дешевий, але має високу обчислювальну потужність, має низьке споживання енергії, вбудований Wi-Fi чип та велику кількість бібліотек для периферії;
- Оптичний ToF-датчик VL53L0X – оптимальність ціни, простота використання, швидкість роботи, низьке енергоспоживання та висока точність вимірювань.

Програмна частина розробки включала розробку прошивки мікроконтролера, а також створення засобів візуалізації результатів сканування. Програмне забезпечення мікроконтролера задовольняє сучасним вимогам до швидкодії, багатозадачності, гнучкості, кросплатформеності та можливості розширення функціональності. Також досягнуто можливість переконфігурації модуля через бездротову мережу без необхідності перезавантаження модуля. Реалізація серверу та клієнта візуалізації забезпечує зручний інтуїтивний інтерфейс та можливість введення власної конфігурації.

					ІАЛЦ.467200.004 ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

Тестування модуля сканування простору підтвердило його функціональні можливості, а саме:

- Сканування простору у різних режимах (отримання даних від датчика та їх попередня обробка перед передачею мережею для подальшого використання системами навігації);
- Поточкову передачу даних у реальному часі;
- Можливість переконфігурації модуля під час роботи (зміна області та кроку сканування, вибір режиму роботи);

Розроблений модуль має низьку собівартість та енергозатратність за рахунок оптимального вибору його апаратних компонентів.

Програмно-апаратний модуль сканування простору може бути використаний в системах наземного роботизованого керування, автономних транспортних системах, для створення тривимірних комп'ютерних моделей, при побудові карт місцевості.

В подальшому можливе вдосконалення програмно-апаратного модуля за рахунок використання потужніших мікроконтролерів, додавання нових алгоритмів і функцій сканування та обробки.

					ІАЛЦ.467200.004 ПЗ	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. How the Solid-State LiDAR works (and why everyone bets on it) – Think autonomous [Електронний ресурс].
URL: <https://www.thinkautonomous.ai/blog/solid-state-lidar/> (дата звернення: 17.05.2026).
2. Лідар – Вікіпедія [Електронний ресурс].
URL: <https://uk.wikipedia.org/wiki/Лідар/> (дата звернення: 17.05.2025).
3. What is photogrammetry – Artec3D [Електронний ресурс].
URL: <https://www.artec3d.com/learning-center/what-is-photogrammetry> (дата звернення: 17.05.2026).
4. Understanding I2C – TI [Електронний ресурс].
URL:
https://www.ti.com/lit/an/slva704/slva704.pdf?ts=1778963605702&ref_url=https%253A%252F%252Fwww.google.com%252F (дата звернення: 17.05.2026).
5. VL53L0X Datasheet – Alldatasheet [Електронний ресурс].
URL:
<https://www.alldatasheet.com/datasheet-pdf/view/948120/STMICROELECTRONICS/VL53L0X.html> (дата звернення: 17.05.2026).
6. SG-90 9 g Micro Servo [Електронний ресурс].
URL:
<https://www.friendlywire.com/projects/ne555-servo-safe/SG90-datasheet.pdf> (дата звернення: 26.05.2026)
7. ESP-WROOM-32 Datasheet (PDF) - ESPRESSIF SYSTEMS (SHANGHAI) CO., LTD. [Електронний ресурс].
URL:
<https://www.alldatasheet.com/datasheet-pdf/pdf/1179101/ESPRESSIF/ESP-WROOM-32.html> (дата звернення: 26.05.2026).

					ІАЛЦ.467200.004 ПЗ	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дата		

8. Christopher Preschern *Fluent C: Principles, Practices, and Patterns*. O'Reilly Media: Santa Rosa, CA, 2022. 308 с. (дата звернення: 27.05.2026).
9. [THREE.js](https://threejs.org/docs/) docs [Електронний ресурс].
URL: <https://threejs.org/docs/> (дата звернення: 28.05.2026).
10. Tanenbaum A. S. *Modern Operating Systems*. Herbert Bos. 5th ed. Hoboken : Pearson, 2022. 1136 с. (дата звернення: 28.05.2026).
11. State Machines with Function Pointers [Електронний ресурс].
URL: <https://www.beningo.com/158-state-machines-with-function-pointers/>
(дата звернення: 28.05.2026)
12. WebSocket API (WebSockets) [Електронний ресурс].
URL: https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API
(дата звернення: 29.05.2026)
13. Python asyncio — Asynchronous I/O [Електронний ресурс].
URL: <https://docs.python.org/3/library/asyncio.html> (дата звернення: 29.05.2026)
14. Python FastAPI Reference [Електронний ресурс].
URL: <https://fastapi.tiangolo.com/reference/> (дата звернення: 29.05.2026)

					ІАЛЦ.467200.004 ПЗ	Арк.
						53
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТОК 1

Програмно-апаратний модуль сканування простору на основі оптичних датчиків
відстані.

Програмно-апаратний модуль сканування простору.

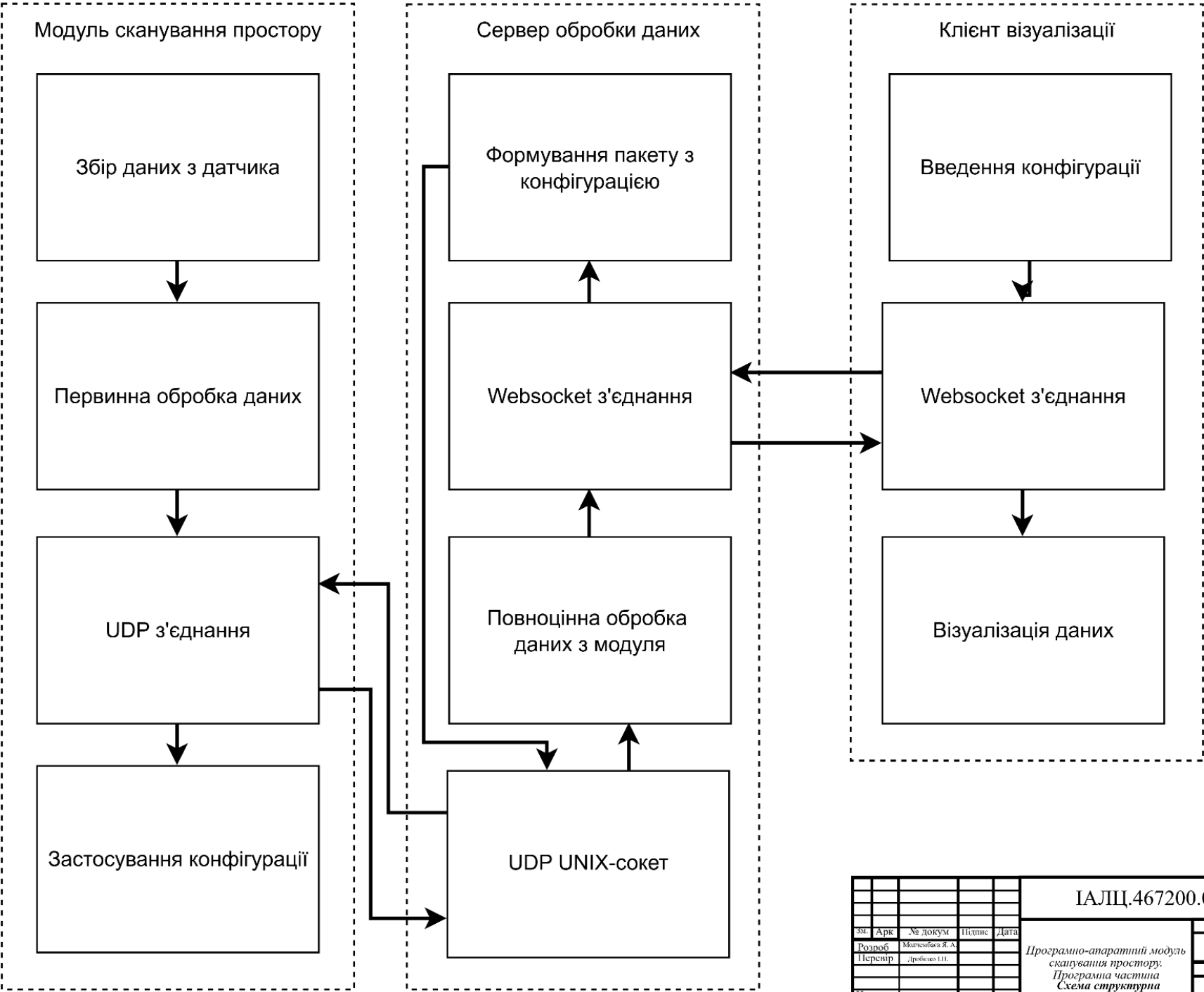
Програмна частина

Схема структурна

ІАЛЦ.467200.005 Д1

Аркушів – 1

Київ 2026



ІАЛЦ.467200.005 Д1				
Зм.	Арк.	№ докум.	Підпис	Дата
Розроб		Молоткоба Я. А.		
Перепір		Дробіско І. П.		
Н. контр.		Калаченко Я. М.		
Затв.		Романович В. О.		
Програмно-апаратний модуль сканування простору. Програмна частина. Схема структурна				
Лист	Маса	Масштаб		
Лист	Аркушів			
КПІ ім. Ігоря Сікорського, ФІІСПМ, гр. КВ-23				

ДОДАТОК 2

Програмно-апаратний модуль сканування простору на основі оптичних датчиків
відстані.

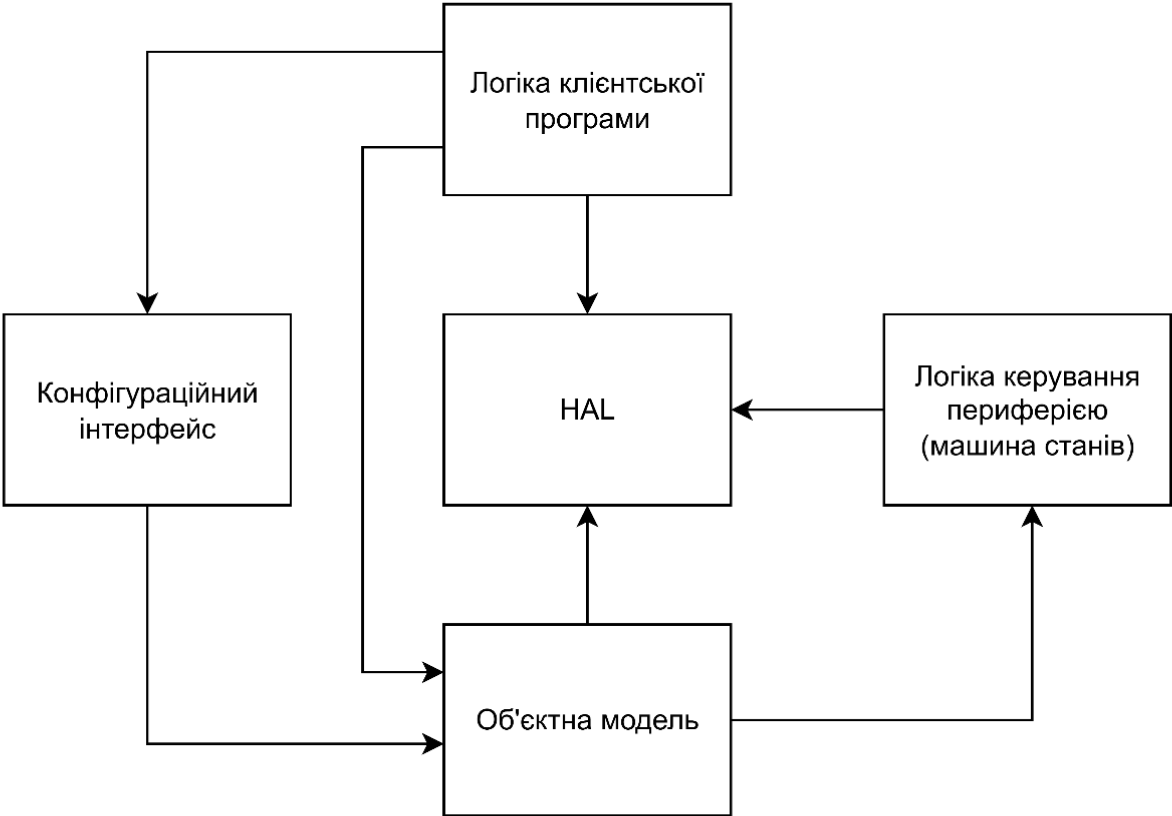
Архітектура програмного забезпечення мікроконтролера

Схема структурна

ІАЛЦ.467200.006 Д2

Аркушів – 1

Київ 2026



ДОДАТОК 3

Програмно-апаратний модуль сканування простору на основі оптичних датчиків
відстані.

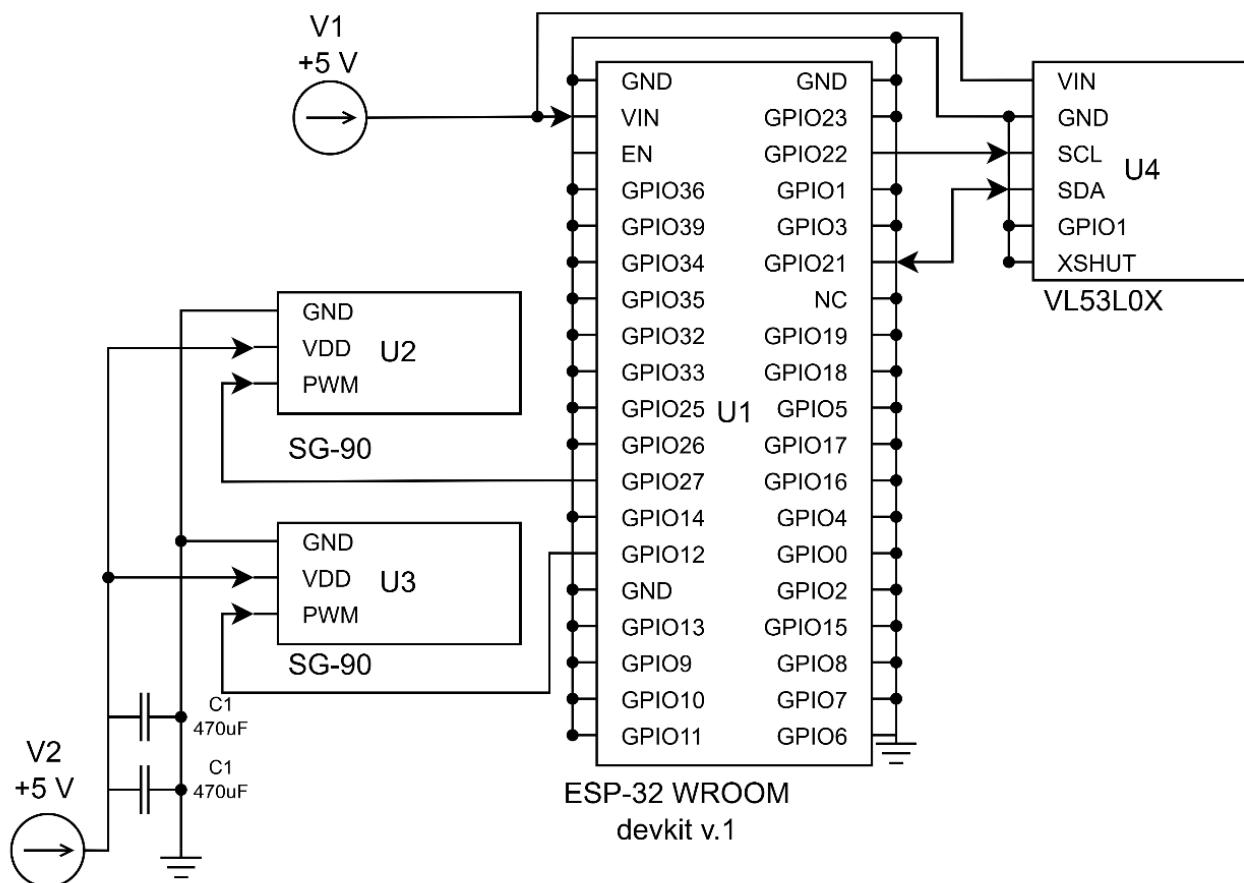
**Програмно-апаратний модуль сканування простору на
основі оптичних датчиків відстані**

Схема електрична принципова

ІАЛЦ.467200.007 ЕЗ

Аркушів – 1

Київ 2026



ІАЛЦ.467200.007 Е3					
№	Арк	№ докум	Підпис	Дата	
Розроб		Морозовська Я. А.			
Перевір		Дробіш І. І.			
Н. контр.		Скочко І. М.			
Затв.		Романенко В. О.			
Програмно-апаратний модуль сканування простору на основі оптичних датчиків відстані			Лист		
Схема принципова			Архив		
			КПІ ім. Ігоря Сікорського, ФІСІПМ, гр. КВ-23		

ДОДАТОК 4

Програмно-апаратний модуль сканування простору на основі оптичних датчиків
відстані.

Сканування простору у тривимірному режимі

Схема алгоритму

ІАЛЦ.467200.008 ДЗ

Аркушів – 1

Київ 2026

