

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»

УЛЬТРАЗВУКОВИЙ РАДАР НА ОСНОВІ МІКРОКОНТРОЛЕРА
ARDUINO

ULTRASONIC RADAR BASED ON ARDUINO MICROCONTROLLER

спеціальність 123 Комп'ютерна інженерія

(шифр і назва спеціальності)

освітня програма Комп'ютерна інженерія

(назва освітньої програми)

Виконав: здобувач вищої освіти

групи КІз-41

Харчук Артем Олександрович

(підпис)

Керівник:

к.т.н., доцент

Поліщук Микола Миколайович

(підпис)

Кваліфікаційну роботу

допущено до захисту

« 04 » 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. Г. Терлецький

« 10 » 01 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Харчуку Артему Олександровичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи *Ультразвуковий радар на основі мікроконтролера Arduino*

Керівник роботи *к.т.н., доцент Поліщук Микола Миколайович*

затверджені 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

Теоретичні основи створення ультразвукового радара на основі Arduino

Апаратна реалізація ультразвукового радара на основі Arduino

Розробка програмно-апаратної частини ультразвукового радара

Висновки

5. Перелік графічного (ілюстративного) матеріалу:

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
<i>Теоретичні основи створення ультразвукового радара на основі Arduino</i>	<i>Поліщук М.М., доцент</i>		
<i>Апаратна реалізація ультразвукового радара на основі Arduino</i>	<i>Поліщук М.М., доцент</i>		
<i>Розробка програмно-апаратної частини ультразвукового радара</i>	<i>Поліщук М.М., доцент</i>		
<i>Нормоконтроль</i>	<i>Багнюк Н.В., доцент</i>		
<i>Гарант ОП</i>	<i>Лавренчук С.В., доцент</i>		
<i>Показник запозичень тексту</i>		_____%	
<i>Академічна доброчесність</i>	<i>Міскевич О.І., ст.викладач</i>		

7. Дата видачі завдання 10.01.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	<i>Огляд літератури із досліджуваної проблеми, основи створення ультразвукового радара на основі Arduino</i>	до 10.02.2025 р.	Виконано
2.	<i>Апаратна реалізація ультразвукового радара на основі Arduino</i>	до 02.03.2025 р.	Виконано
3.	<i>Розробка програмно-апаратної частини ультразвукового радара</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 10.05.2025 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 кафедру</i>	до 10.06.2025 р.	Виконано

Здобувач вищої освіти

(підпис)

Харчук А.О.

(прізвище, ініціали)

Керівник кваліфікаційної роботи

(підпис)

Поліщук М.М.

(прізвище, ініціали)

АНОТАЦІЯ

Харчук А. О. Ультразвуковий радар на основі мікроконтролера 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, додатку.

Перший розділ присвячено огляду предметної області, тут обґрунтовано актуальність обраної теми. Розглянуто принципи роботи ультразвукових сенсорів та основи побудови систем радарного типу. Зроблено огляд переваг мікроконтролерів Arduino в радарних системах та сучасних досліджень у сфері ультразвукового сканування на базі Arduino.

В другому розділі здійснено вибір та обґрунтування засобів розробки. Здійснено вибір апаратних компонентів для реалізації ультразвукового радара, обґрунтовано вибір мікроконтролера, проаналізовано ультразвуковий сенсор HC-SR04 та вибір сервоприводу для сканування SG90. Представлено схемотехнічне рішення компонентів до Arduino та можливі варіанти інтеграції системи.

Третій розділ присвячено розробці програмно-апаратної частини ультразвукового радара, зроблено огляд апаратного забезпечення, представлено складання ультразвукового радара та програмування мікроконтролера.

Ключові слова: радіолокаційні системи, Arduino Uno, ультразвуковий радар, компонент, датчик, мікроконтролер, плата, частота, сенсор

ANNOTATION

Kharchuk A. Ultrasonic radar based on the Arduino microcontroller. Manuscript.

Qualification work of the bachelor of the OP «Computer Engineering» specialty 123 Computer Engineering. Lutsk National Technical University. Lutsk, 2025.

Qualification work consists of an introduction, three chapters, conclusions, a list of sources used, an appendix.

The first chapter is devoted to an overview of the subject area, the relevance of the chosen topic is justified here. The principles of operation of ultrasonic sensors and the basics of building radar-type systems are considered. The advantages of Arduino microcontrollers in radar systems and modern research in the field of ultrasonic scanning based on Arduino are reviewed.

In the second chapter, the selection and justification of development tools are made. The selection of hardware components for the implementation of ultrasonic radar is made, the choice of microcontroller is justified, the HC-SR04 ultrasonic sensor and the choice of SG90 scanning servo drive are analyzed. The schematic solution of the components for Arduino and possible options for system integration are presented.

The third section is devoted to the development of the software and hardware part of the ultrasonic radar, an overview of the hardware is given, the assembly of the ultrasonic radar and programming of the microcontroller are presented.

Keywords: radar systems, Arduino Uno, ultrasonic radar, component, sensor, microcontroller, board, frequency, sensor

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ СТВОРЕННЯ УЛЬТРАЗВУКОВОГО РАДАРА НА ОСНОВІ ARDUINO	9
1.1 Принципи роботи ультразвукових сенсорів	9
1.2 Основи побудови систем радарного типу.....	13
1.3 Переваги мікроконтролерів Arduino в радарних системах	17
1.4 Огляд сучасних досліджень у сфері ультразвукового сканування на базі Arduino.....	19
РОЗДІЛ 2 АПАРАТНА РЕАЛІЗАЦІЯ УЛЬТРАЗВУКОВОГО РАДАРА НА ОСНОВІ ARDUINO	24
2.1 Вибір апаратних компонентів для реалізації ультразвукового радара.....	24
2.1.1 Обґрунтування вибору мікроконтролера Arduino Uno	24
2.1.2 Аналіз ультразвукового сенсора HC-SR04	28
2.1.3 Вибір сервоприводу для сканування SG90	29
2.2 Схемотехнічне рішення компонентів до Arduino.....	32
2.3 Можливі варіанти інтеграції системи в мобільні або стаціонарні рішення	34
РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНО-АПАРАТНОЇ ЧАСТИНИ УЛЬТРАЗВУКОВОГО РАДАРА	37
3.1 Огляд апаратного забезпечення.....	37
3.2 Складання ультразвукового радара на основі мікроконтролера	39
3.3 Візуалізація радарних даних на екрані або в реальному часі на ПК	44
3.4 Програмування мікроконтролера для керування ультразвуковим радаром	46
ВИСНОВКИ	49
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	51
ДОДАТКИ	55

ВСТУП

У сучасних умовах швидкого розвитку технологій особливу увагу приділяють радіолокаційним системам, які демонструють високий потенціал у зборі детальної інформації, необхідної для моніторингу та проведення операцій в умовах обмеженої видимості. Зокрема, хмарність значно обмежує можливості використання оптичних супутникових даних, водночас радарні системи продовжують ефективно працювати, забезпечуючи безперервний збір даних незалежно від погодних умов. Проте, незважаючи на доступність супутникових радарів, їхня повторюваність траєкторій не дозволяє ефективно виконувати завдання картування дрібномасштабних об'єктів у часових рамках, необхідних для оперативного реагування, наприклад, під час очищення територій чи морських операцій. Саме тому авіаційні та наземні системи набувають особливої значущості для детального картування та оперативного збору даних.

З огляду на це, особливої актуальності набуває розробка компактних міні-радарів, здатних візуалізувати дані в реальному часі, використовуючи доступні апаратно-програмні рішення, такі як платформа Arduino. Вибір Arduino обумовлений її відкритою архітектурою, широкою підтримкою спільноти розробників, доступністю програмних бібліотек та великою кількістю сумісних периферійних пристроїв, зокрема ультразвукових сенсорів.

Мета дослідження – розробка та реалізація міні-радарної системи на базі платформи Arduino для відображення та аналізу даних у реальному часі.

Об'єкт дослідження – міні-радарна система, що функціонує на основі мікроконтролера Arduino.

Предмет дослідження – апаратно-програмна платформа Arduino та її застосування для керування радарною системою, збору та візуалізації даних.

Для досягнення мети було поставлено ряд завдань:

– дослідити принципи роботи ультразвукових радарів, розглянути їхні особливості;

- проаналізувати основні напрями застосування ультразвукових радарів у сучасних технологічних;
- обґрунтувати доцільність вибору Arduino як апаратної платформи для реалізації ультразвукового радарного пристрою;
- здійснити підбір ключових компонентів з урахуванням технічних вимог до точності, енергоспоживання та масштабованості системи;
- розробити схему з'єднання компонентів та макет ультразвукового радара.

РОЗДІЛ 1

ТЕОРЕТИЧНІ ОСНОВИ СТВОРЕННЯ УЛЬТРАЗВУКОВОГО РАДАРА НА ОСНОВІ ARDUINO

1.1 Принципи роботи ультразвукових сенсорів

Ультразвукові сенсори – це електронні пристрої, що використовують звукові хвилі високої частоти (понад 20 кГц) для виявлення об'єктів і вимірювання відстані до них. Їхній принцип дії базується на ехолокації, сенсор випромінює короткий акустичний імпульс, який, вдаряючись об поверхню об'єкта, відбивається назад. За часом, який проходить між передачею й прийомом сигналу, система розраховує відстань до об'єкта за формулою (1.1):

$$D = \frac{v \times t}{2}, \quad (1.1)$$

де D – відстань до об'єкта;

v – швидкість звуку у повітрі (близько 343 м/с при температурі 20 °С);

t – час між випроміненням та прийомом сигналу [1].

Принцип дії ультразвукового сенсора схожий на той, який використовують кажани чи підводні сонари. Ця технологія дозволяє точно вимірювати відстані в межах від кількох сантиметрів до кількох метрів, що робить її придатною для інтеграції в системи моніторингу, автоматизованого керування, робототехніки та інтелектуального транспорту.

Сучасні ультразвукові сенсори складаються з наступних елементів:

– п'єзоелектричний передавач – перетворює електричний імпульс у звукову хвилю;

– приймач – реєструє зворотний сигнал та перетворює його на електричний імпульс;

– схема обробки сигналу – вимірює час затримки та виводить результат на мікроконтролер [2].

Найпоширенішим недорогим сенсором у проєктах з Arduino є HC-SR04, який має два окремі перетворювачі (передавач і приймач). Цей датчик працює в діапазоні 2-400 см, має точність до 0,3 см, і потребує напругу 5 В. Альтернативи включають JSN-SR04T (з водозахистом) і MaxBotix MB1040 (зі зменшеним електромагнітним шумом та інтегрованим трансд'юсером).



Рисунок 1.1 – Водонепроникний ультразвуковий датчик відстані JSN-SR04T [3]



Рисунок 1.2 – Ультразвуковий далекомір MaxBotix MB1040 [4]

Застосування ультразвукових сенсорів потребує врахування певних обмежень: відбиття сигналу залежить від кута нахилу об'єкта, поверхні (гладка чи жорстка), а також температури та вологості повітря. Для компенсації цих впливів іноді додають температурні датчики та фільтрацію сигналів у ПЗ [5].

Процес вимірювання відстані ультразвуковим сенсором базується на чіткій послідовності дій, які ініціюються та контролюються мікроконтролером (наприклад, Arduino Uno). Типовим прикладом такого сенсора є HC-SR04, який має два п'єзоелектричні перетворювачі: один – передавач (Trigger), другий – приймач (Echo). Алгоритм роботи охоплює такі основні кроки:

- ініціалізація імпульсу мікроконтролером. Arduino через цифровий пін (наприклад, D9) подає керуючий сигнал тривалістю 10 мікросекунд на вхід Trigger. Цей короткий імпульс активує генерацію ультразвукової хвилі. До подачі сигналу пін утримується в низькому логічному рівні (LOW);

- генерація ультразвукової хвилі. Сенсор HC-SR04 автоматично випромінює серію з 8 ультразвукових імпульсів з частотою приблизно 40 кГц. Ці хвилі поширюються через повітря у прямому напрямку з постійною швидкістю – близько 343 м/с при температурі +20° С;

- відбиття сигналу від об'єкта. Коли ультразвукова хвиля досягає об'єкта, вона відбивається назад. Хвиля може бути частково поглинута або розсіяна залежно від форми, матеріалу та кута поверхні об'єкта. Проте достатня частина сигналу, як правило, повертається у напрямку сенсора;

- фіксація часу повернення сигналу. Після передачі імпульсів сенсор автоматично активує пін Echo, переводячи його в високий рівень (HIGH). Цей пін утримується в стані HIGH до моменту реєстрації відбитого імпульсу. Тривалість HIGH-сигналу відповідає часу проходження звукової хвилі до об'єкта та назад (в обидва боки). Arduino фіксує цю тривалість за допомогою функції pulseIn();

- обчислення відстані до об'єкта. Знаючи час t , що пройшов між передачею й прийомом сигналу, контролер обчислює відстань за формулою (1.1)

або, враховуючи швидкість звуку в міліметрах на мікросекунду ($v \approx 0,343$ мм/мкс), спрощено за формулою (1.2) [1]:

$$D = \frac{t \times 0,343}{2} \quad (1.2)$$

В Arduino-проектах часто використовують округлену формулу (1.3) (для отримання результату в см) [5]:

$$D = \frac{t}{58,0} \quad (1.3)$$

Таким чином, якщо час $t = 580$ мкс, то відстань становитиме 10 см.

Для наочного розуміння процесу функціонування ультразвукового сенсора HC-SR04 у складі мікроконтролерної системи доцільно подати узагальнену схему алгоритму взаємодії між компонентами. На рисунку 1.3 наведено спрощену послідовність обміну сигналами між контролером Arduino та сенсором, яка демонструє принцип дії пристрою: від формування імпульсу до обчислення відстані на основі отриманих даних. Такий алгоритм є основою побудови більш складних систем ультразвукового сканування та радарного моніторингу.

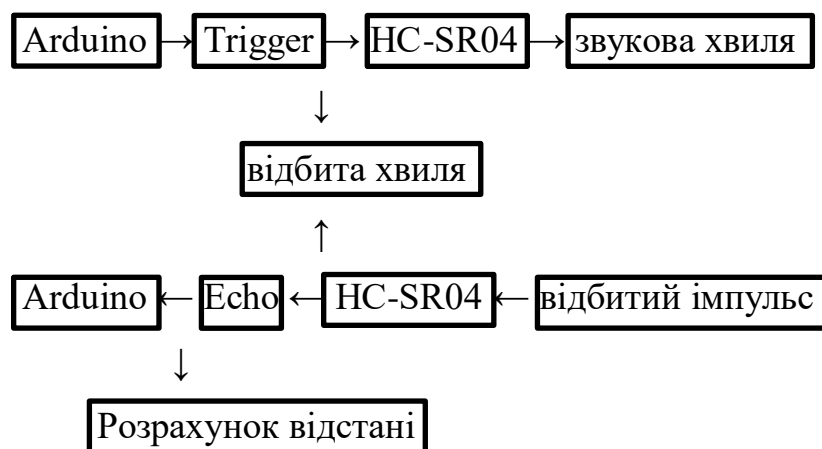


Рисунок 1.3 – Алгоритм функціонування сенсора HC-SR04 [5]

У результаті розгляду принципу дії ультразвукових сенсорів встановлено, що ці пристрої забезпечують точне й недороге вимірювання відстані шляхом аналізу часу проходження звукового імпульсу. Ультразвукові сенсори, зокрема HC-SR04, характеризуються високою чутливістю, простотою інтеграції та сумісністю з платформою Arduino. Алгоритм їхньої роботи базується на формуванні тригерного імпульсу, передачі й прийомі звукової хвилі та обчисленні часу її проходження. Такий принцип дає змогу створювати ефективні системи для виявлення об'єктів у навколишньому середовищі.

1.2 Основи побудови систем радарного типу

Термін «радар» виник у 1930-х роках у США та позначає технологію виявлення об'єктів на основі випромінювання та приймання радіохвиль. Радари використовувались здебільшого у військових цілях для виявлення літаків, кораблів, а також для навігації та прогнозування погоди. Основним принципом дії радара є вимірювання часу, за який радіохвиля, відбившись від об'єкта, повертається назад до приймача. Таким чином визначається відстань до об'єкта, а в деяких випадках – також швидкість його руху та розміри [6].

Сьогодні ідея радарного зондування була адаптована до інших типів хвиль – наприклад, ультразвуку, який замінює радіохвилі в компактних системах, таких як датчики відстані, навігаційні системи для роботів та домашні охоронні системи. Такі пристрої отримали назву ультразвукових радарів, або ехолокаційних сканерів, оскільки вони працюють за ідентичним фізичним принципом, але з використанням звукового діапазону.

Ультразвуковий радар – це система виявлення об'єктів і вимірювання відстані до них шляхом випромінювання та приймання звукових хвиль високої частоти (понад 20 кГц) [7]. В основі таких систем лежить ультразвуковий сенсор (наприклад, HC-SR04), який у парі з мікроконтролером (зазвичай Arduino Uno) та сервоприводом здійснює активне зондування простору в одному або кількох напрямках (рис. 1.4).

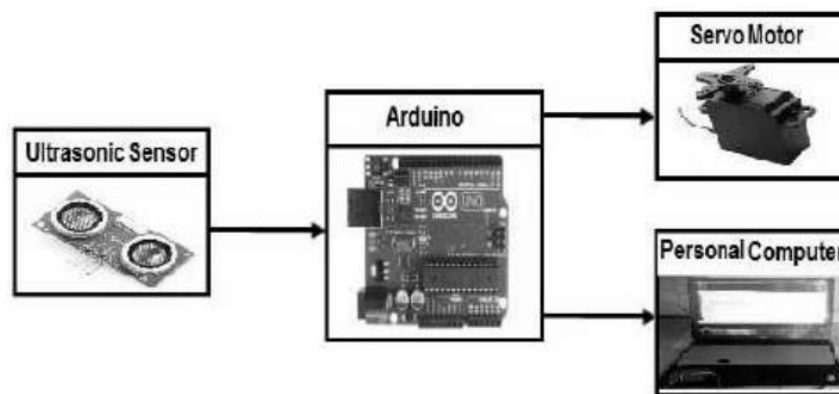


Рисунок 1.4 – Конструкція блоку для радіолокаційної системи малої дальності на базі Arduino [7]

Ключовими елементами системи є передавач сигналу (ультразвуковий модуль), приймач сигналу (вбудований у той самий модуль), механізм позиціонування (зазвичай сервопривід SG90), обчислювальний модуль (Arduino Uno) та інтерфейс виведення даних (LCD, серійний порт, Bluetooth, ПК).

Ультразвуковий радар здатен функціонувати у двох режимах:

- статичний одновимірний режим – сенсор встановлений нерухомо і вимірює відстань в одному напрямку;
- динамічний скануючий режим – сенсор обертається за допомогою сервопривода, скануючи простір у секторі, зазвичай до 180° , утворюючи полярну карту об'єктів.

Під час обертання сенсора система виконує серію вимірювань у фіксованих кутових положеннях (наприклад, кожні $1-2^\circ$), після чого отримані відстані прив'язуються до відповідного кута, формуючи полярну діаграму. Такий підхід дозволяє візуалізувати розташування об'єктів навколо пристрою у двовимірній площині.

Візуалізації часто реалізуються за допомогою мови програмування Processing або безпосередньо на TFT-дисплеях, підключених до Arduino. Це дозволяє створювати інтерактивні та наочні інтерфейси для відображення результатів сканування (рис. 1.5).

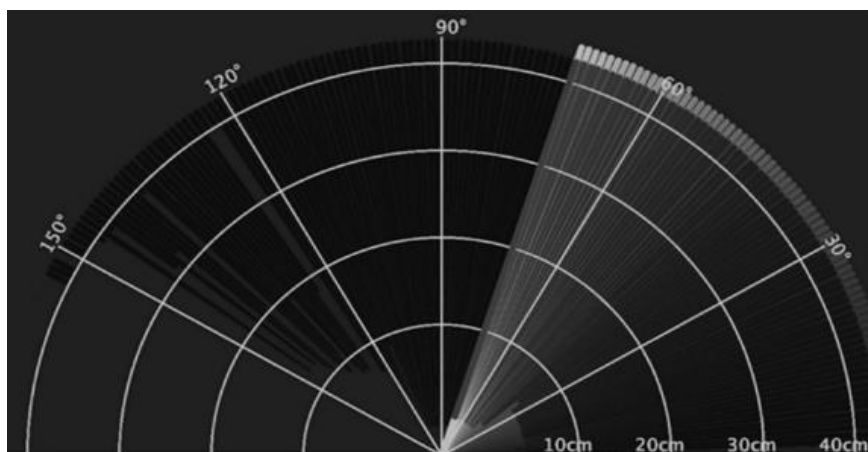


Рисунок 1.5 – Приклад полярної діаграми, яка ілюструє ультразвуковий радар на базі Arduino [8]

Типова ультразвукова радарна система на основі Arduino реалізується за модульним принципом, де кожен компонент виконує окрему функцію в загальній структурі. Така архітектура забезпечує гнучкість, масштабованість та зручність у налагодженні й обслуговуванні.

Мікроконтролер Arduino Uno є центральним елементом, що координує роботу сенсора, сервоприводу та обробляє дані. Ультразвуковий сенсор HC-SR04 відповідає за передавання і приймання звукового імпульсу, на основі якого вимірюється відстань. Сервопривід SG90 обертає сенсор у заданому секторі (наприклад, 0-180°) з певним кроком, створюючи карту простору та система виведення – дисплей, Bluetooth-модуль, USB-з'єднання або серійний монітор для передачі результатів сканування [9]. Такий поділ дозволяє реалізувати як одновимірні системи, так і двовимірні, де створюється повноцінна полярна карта простору з об'єктами на різних відстанях та під різними кутами.

У динамічному режимі сканування сервопривід повертає сенсор на фіксований кут, наприклад, кожні 2°, і в кожному положенні виконується вимірювання відстані. Отримані дані зберігаються у масиві, який згодом може бути виведений на екран у числовому вигляді, переданий на комп'ютер для візуалізації в середовищі обробки (наприклад, Processing, Python, MATLAB) або виведений у вигляді ASCII-графіки на серійному моніторі.

Така логіка дозволяє побудувати полярну діаграму розташування об'єктів у зоні огляду радара. При цьому кут повороту відповідає полярному куту (ϕ), виміряна відстань – радіальній координаті (r), система формує набір точок (r, ϕ), що створює картину навколишнього середовища.

Ультразвукові радари, побудовані на базі мікроконтролера Arduino, є універсальними пристроями, що знайшли широке застосування в різних галузях техніки та побуту. У таблиці 1.1 узагальнено основні напрями застосування ультразвукових радарів на основі Arduino з коротким описом функціонального призначення та ключових переваг використання.

Таблиця 1.1 – Сфери застосування ультразвукових радарів [10]

Сфера застосування	Опис	Типове призначення	Переваги використання
мобільна робототехніка	інтеграція в автономних роботах для дослідження та навігації у приміщеннях і на відкритому повітрі	уникнення зіткнень, побудова карти простору	простота інтеграції, невеликий розмір, реальна точність
автомобільні системи паркування	встановлення в авто для виявлення перешкод під час паркування	допоміжне паркування, зворотний зв'язок для водія	низька вартість, точне визначення об'єктів на близькій відстані
системи безпеки та охорони	використання в охоронних системах для детекції наближення до дверей або зон обмеженого доступу	автоматичне сповіщення про присутність сторонніх осіб	висока чутливість, можливість роботи в темряві
смарт-пристрої (розумний дім)	вбудовування у побутові системи для автоматичного вмикання світла, відкривання дверей тощо	адаптивне освітлення, управління пристроями без дотику	енергоефективність, швидка реакція на зміну середовища

Аналіз сфер застосування ультразвукових радарів на базі Arduino демонструє їхню універсальність та ефективність у вирішенні широкого спектра задач – від автономної навігації в мобільній робототехніці до підвищення комфорту й безпеки в побутових умовах. Завдяки поєднанню компактності, низького енергоспоживання, простоти програмування та високої чутливості, ці системи стали оптимальним вибором для прототипів IoT-пристроїв і функціональних автоматизованих рішень.

Отже, ультразвуковий радар на базі Arduino є прикладом спрощеної радарної системи, яка зберігає ключові принципи класичних радарів – імпульсне зондування, фіксацію часу затримки сигналу та побудову просторової карти об'єктів. Залежно від режиму роботи, кроку повороту й типу сенсора, система може досягати досить високої точності у виявленні перешкод. Наявність модуля сканування, сервопривода та можливостей обробки й виводу даних робить таку конструкцію функціональною, гнучкою та придатною для застосування в автономних пристроях.

1.3 Переваги мікроконтролерів Arduino в радарних системах

Arduino – це відкрита апаратно-програмна платформа, яка здобула широке поширення у сфері прототипування електронних систем, зокрема ультразвукових радарів [11]. Завдяки відкритому апаратному коду та широкому вибору сумісних плат і модулів, розробники можуть гнучко адаптувати архітектуру радарної системи до конкретних умов застосування. Важливою перевагою також є можливість оновлення прошивки та інтеграція з мовами вищого рівня (Python, MATLAB, LabVIEW), що відкриває перспективи для обробки великих обсягів даних та машинного навчання в реальному часі.

Розробка ультразвукових радарних систем на основі Arduino стала можливою завдяки низці технічних та програмних переваг, які відрізняють цю платформу від інших мікроконтролерних рішень. Arduino забезпечує гнучку архітектуру, високу сумісність із широким спектром модулів, а також спрощене програмування, що особливо важливо на етапі прототипування. У таблиці 1.2 узагальнено ключові переваги використання Arduino Uno, Nano та Mega в контексті побудови радарних систем, а також наведено типові приклади реалізації відповідних функціональних можливостей.

Таблиця 1.2 – Переваги використання Arduino в ультразвукових радарних системах [12]

Перевага	Опис	Типові приклади застосування
1	2	3
простота програмування	Arduino IDE підтримує інтуїтивно зрозумілий синтаксис на базі C/C++. Широкий набір бібліотек (Servo.h, NewPing.h, LiquidCrystal.h) прискорює розробку	зчитування даних із сенсорів, керування сервоприводами, обробка сигналів, виведення даних на дисплей або ПК
висока сумісність з модулями	підтримка широкого спектра периферійних пристроїв: ультразвукові сенсори (HC-SR04), сервомотори (SG90), дисплеї (OLED), бездротові модулі (ESP8266)	побудова радарних систем з візуалізацією або передачею даних по Bluetooth / Wi-Fi
доступність та енергоефективність	низька вартість, можливість живлення від батареї або USB. Споживання Arduino Uno – ~50-70 мА	створення автономних мобільних пристроїв або енергоефективних систем моніторингу
велика спільнота та документація	Arduino має глобальну спільноту, доступну документацію, відкриті схеми та приклади, що спрощує вивчення та відлагодження систем	прототипування, розробка навчальних і наукових проєктів, експериментальна інженерія
модульна масштабованість	Arduino легко розширюється завдяки використанню додаткових модулів (датчики, дисплеї, мережеві шини). Підтримка ESP, Ethernet, I2C	побудова розподілених або IoT-систем із логуванням, хмарною синхронізацією або багатоточковим скануванням

Мікроконтролери Arduino дозволяють суттєво спростити процес створення прототипу ультразвукового радара – від ідеї до робочого зразка. Завдяки широкій підтримці готових бібліотек та зразків коду, розробник може зосередитись на логіці роботи системи, а не на низькорівневому програмуванні.

У реальних умовах Arduino демонструє високу надійність і стабільність під час безперервного зондування простору. Зокрема, пристрої на базі Arduino активно застосовуються в безпілотних роботах для виявлення перешкод, у системах допоміжного паркування, у «розумному будинку» для автоматичного виявлення руху або наближення. Завдяки модульній структурі та можливості розширення функціоналу (наприклад, за рахунок ESP або GSM-модулів), Arduino залишається оптимальним вибором для побудови компактних, енергоефективних та функціонально гнучких радарних систем.

Отже, Arduino є оптимальною платформою для реалізації радарних систем різного рівня складності, її відкритість, гнучкість і широка підтримка зовнішніх модулів дозволяють легко масштабувати проекти, забезпечуючи не лише збір та обробку даних, а й подальшу інтеграцію в мережеві та IoT-орієнтовані рішення. Це обґрунтовує доцільність використання Arduino у якості базового контролера в системах ультразвукового сканування простору.

1.4 Огляд сучасних досліджень у сфері ультразвукового сканування на базі Arduino

Ультразвукове сканування простору за допомогою мікроконтролерів Arduino є активною темою досліджень у сфері прикладної робототехніки, автоматизації, мобільних систем навігації та розумного середовища. Із кожним роком розширюються технічні можливості таких систем, зокрема завдяки вдосконаленню сенсорів, обробки даних і інтеграції з бездротовими технологіями. Аналіз наукових публікацій за 2020-2025 роки дає змогу виокремити кілька основних напрямів розвитку ультразвукових радарних систем на основі Arduino.

У дослідженні [13] запропоновано розробку мобільної платформи з функцією активного зондування за допомогою HC-SR04, яка використовується для навігації в невідомому середовищі. Застосування Arduino Uno, сервопривода SG90 і простого полярного алгоритму дозволило створити систему, здатну розпізнавати розташування перешкод із точністю до 2 см. Автори продемонстрували, що навіть базові системи сканування здатні забезпечити автономність руху без складної електроніки.

У роботі [14] представлено практичну реалізацію ультразвукової радарної системи (рис. 1.6), побудованої на основі Arduino Uno, що поєднує простоту реалізації з можливістю віддаленого моніторингу завдяки інтеграції модуля ESP8266.

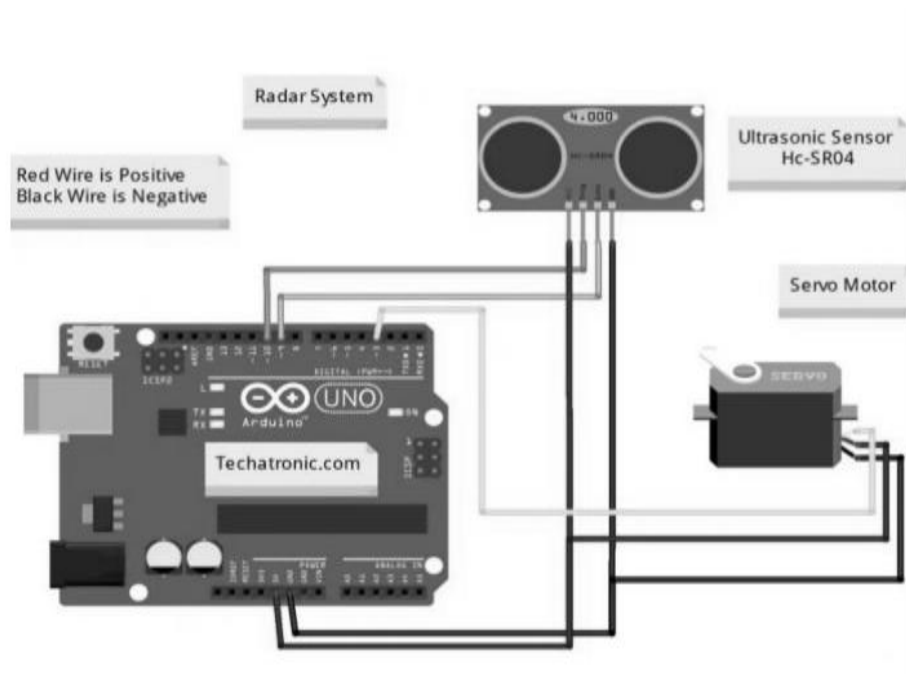


Рисунок 1.6 – Радіолокаційна система з використанням Arduino та ультразвукового датчика [14]

Автори демонструють, що обертання ультразвукового сенсора HC-SR04 за допомогою сервоприводу дозволяє виконувати сканування в секторі 180° , формуючи полярну карту простору (рис. 1.7).

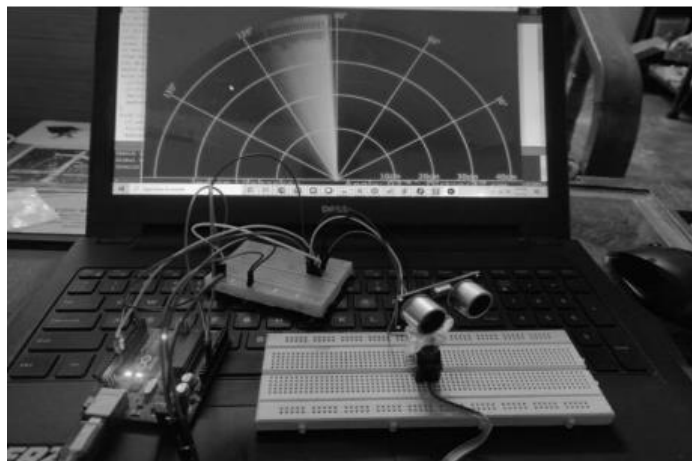


Рисунок 1.7 – Моніторинг і сповіщення в реальному часі [14]

Такий підхід є типовим для недорогих мобільних або стаціонарних систем виявлення об'єктів, які працюють в режимі реального часу. Застосовані алгоритми дозволяють із мінімальними обчислювальними ресурсами Arduino

визначати відстань до об'єкта з подальшою передачею результатів у хмару для візуалізації.

У статті [15] представлено розробку недорогого ультразвукового радарного пристрою на базі платформи Arduino Uno. Система використовує ультразвуковий сенсор HC-SR04 для вимірювання відстані до об'єктів, а сервопривід забезпечує обертання сенсора для сканування простору в секторі 180°. Отримані дані обробляються мікроконтролером Arduino та виводяться на серійний монітор або LCD-дисплей, що дозволяє в реальному часі візуалізувати положення об'єктів у навколишньому середовищі.

Це дослідження демонструє, що поєднання доступних компонентів, таких як Arduino, ультразвуковий сенсор і сервопривід, дозволяє створити функціональну радарну систему для виявлення об'єктів та вимірювання відстані. Простота реалізації та низька вартість роблять таку систему привабливою для освітніх цілей, а також для застосувань у робототехніці, системах уникнення перешкод і базовій навігації. Проект також слугує основою для подальших досліджень у сфері сенсорної інтеграції та вбудованих систем [15].

Дослідження [16] має цінність не лише як прототип низьковартісного радарного пристрою, а й як навчальний інструмент для ознайомлення з основами сенсорної інтеграції та просторової обробки даних. Автори зосереджуються на практичній доступності проекту, використовуючи недорогі апаратні компоненти та відкриті бібліотеки Arduino, а також підкреслюють, що подібні системи можуть знайти застосування в авіаційній навігації малого радіусу, науковому зондуванні, безпілотних системах та розумному моніторингу простору. Враховуючи відкритий код та універсальність компонентів, дана розробка є хорошим прикладом для студентських і науково-дослідницьких проектів з автоматизації та робототехніки.

У статті Hyunmin Jeong та Sangkil Kim [17] представили недорогу FMCW-радарну систему (рис. 1.8), призначену для виявлення об'єктів у приміщенні, зокрема через стіни. Система використовує комерційно доступні

компоненти, включаючи мікроконтролер Arduino Uno для збору даних, що робить її доступною для освітніх та дослідницьких цілей.

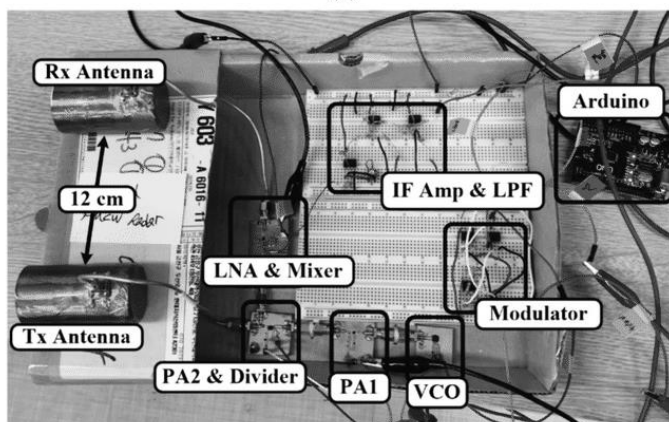
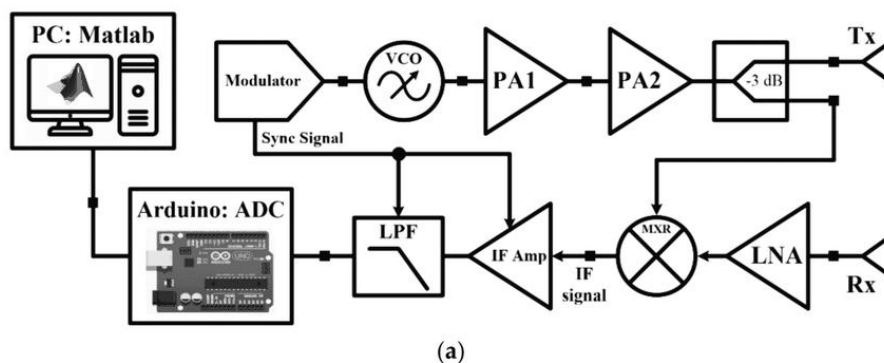


Рисунок 1.8 – Радіолокаційна система FMCW на базі Arduino-ADC для наскрізного детектування (прототип радара) [17]

Радар працює в С-діапазоні частот (5,83-5,94 ГГц) і здатен виявляти об'єкти на відстані до 56,3 метрів у приміщенні. У ході експериментів система успішно виявляла металеві об'єкти через 4-сантиметрову фанерну стіну з точністю менше 10 см. Це дослідження демонструє, що поєднання Arduino з FMCW-радарною технологією може забезпечити ефективне та економічне рішення для задач виявлення об'єктів у складних умовах, таких як через стіни, що відкриває нові можливості для застосування в системах безпеки та моніторингу.

Проаналізовані дослідження [13-17], підтверджують зростаючий науковий і практичний інтерес до побудови ультразвукових радарів на базі

мікроконтролерів Arduino. У всіх роботах підкреслюється універсальність цієї платформи як для освітніх, так і для прикладних цілей.

У роботах [14, 15] автори показали, як за допомогою Arduino Uno та обертального сканування можна створити полярну карту простору з кутовою точністю до $1-2^\circ$ і дальністю виявлення до 4 м. Практична реалізація таких систем ґрунтується на поєднанні алгоритмів обробки часу проходження сигналу та візуалізації результатів у середовищі Processing або через LCD. Дослідження [13, 17] додатково підкреслюють навчальну цінність таких систем, які дозволяють опанувати принципи просторового сканування, моделювання перешкод та інтеграції з ПК. Особливо перспективною є ідея, розглянута в роботі [16], де Arduino використовується у складі FMCW-радару для кризьстіночного сканування – такий підхід демонструє потенціал платформи й у високочастотних застосуваннях, зокрема в системах безпеки або медичних дослідженнях.

Загалом, спільними характеристиками всіх досліджених проєктів є:

- простота відтворення завдяки відкритому коду та доступності компонентів;
- модульність та можливість масштабування (додавання IoT, GPS, GSM, ESP);
- точність і стабільність вимірювання в умовах коротких відстаней;
- навчально-прикладна користь, що дозволяє поєднувати теорію з практикою у сфері вбудованих систем і сенсорних технологій.

Таким чином, аналіз підтверджує, що платформи Arduino, у поєднанні з ультразвуковими сенсорами, є потужним інструментом для створення прототипів адаптивних, економічно ефективних радарних систем, придатних для інтеграції у ширший контекст Smart Devices, IoT та автономної навігації.

РОЗДІЛ 2

АПАРАТНА РЕАЛІЗАЦІЯ УЛЬТРАЗВУКОВОГО РАДАРА НА ОСНОВІ ARDUINO

2.1 Вибір апаратних компонентів для реалізації ультразвукового радара

2.1.1 Обґрунтування вибору мікроконтролера Arduino Uno

Ультразвукова радарна система зазвичай складається з ультразвукового датчика відстані (наприклад, HC-SR04) та сервопривода (наприклад, SG90), що обертає датчик для сканування простору. Така система посилає ультразвукові імпульси і вимірює час відлуння, щоб визначати відстань до об'єктів, і за допомогою сервопривода здійснює круговий огляд, імітуючи роботу радара. Вибір мікроконтролера для керування датчиком і сервоприводом є критично важливим, оскільки він впливає на точність вимірювань, швидкодію та зручність розробки. Одним з найпопулярніших контролерів для подібних проектів є Arduino Uno, 8-розрядна плата на базі мікроконтролера ATmega328P (рис. 2.1). Arduino Uno часто розглядають поряд з іншими сучасними платами – такими як ESP32, STM32 (наприклад, плата Blue Pill), Raspberry Pi Pico (RP2040), а також іншими платами Arduino (наприклад, Nano або Mega) [18].

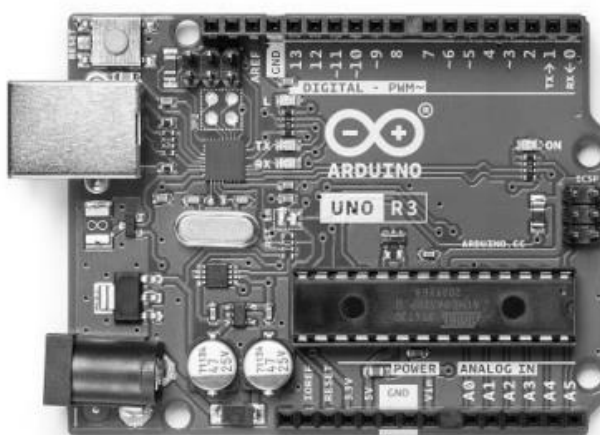


Рисунок 2.1 – Плата Arduino UNO [18]

Arduino Uno зарекомендував себе як стандарт у освіті та прототипуванні завдяки простоті використання та надійності [19], попри відносно скромні технічні характеристики (16 МГц, 32 КБ флеш-пам'яті, 2 КБ SRAM), Uno має достатню продуктивність для задач реального часу, зокрема вимірювання часу відлуння ультразвуку та генерування PWM-сигналів для керування сервоприводом. Щоб оцінити доцільність вибору Uno, розглянемо його у порівнянні з альтернативними контролерами (ESP32, STM32, Raspberry Pi Pico) та близькими аналогами з родини Arduino (Nano, Mega).

В таблиці 2.1 узагальнимо основні характеристики Arduino Uno та альтернатив, важливих в контексті використання з ультразвуковим датчиком та сервоприводом.

Таблиця 2.1 – Порівняння Arduino Uno з іншими альтернативними контролерами [20-24]

Плата	Мікроконтролер (MCU)	Тактова частота	Flash пам'ять	SRAM	Цифрові I/O (PWM)	Аналогові входи	Робоча напруга логіки
Arduino Uno R3	ATmega328P (8-біт AVR)	16 МГц	32 КБ	2 КБ	14 (6 PWM)	6	5 В
Arduino Nano	ATmega328P (8-біт AVR)	16 МГц	32 КБ	2 КБ	14 (6 PWM)	8	5 В
Arduino Mega 2560	ATmega2560 (8-біт AVR)	16 МГц	256 КБ	8 КБ	54 (15 PWM)	16	5 В
ESP32 Dev Kit	ESP32-WROOM-32 (2× Xtensa 32-біт)	до 240 МГц	4 МБ	520 КБ	36 (16 PWM)	до 18 (12-біт)	3,3 В
STM32 «Blue Pill»	STM32F103C8 T6 (ARM Cortex-M3)	72 МГц	64 КБ	20 КБ	37 (15 PWM)	10 (12-біт)	3,3 В
Raspberry Pi Pico	RP2040 (2× Cortex-M0+)	133 МГц	2 МБ	264 КБ	26 (16 PWM)	3 (12-біт)	3,3 В

Для ESP32 реальна кількість доступних аналогових входів залежить від використання Wi-Fi (частина ADC-каналів конфліктує) плата, Blue Pill може мати 128 КБ Flash у деяких ревізіях, Raspberry Pi Pico підтримує USB як хост або пристрій. Значення споживання струму 19 мА для Uno наведено для типового

режиму без навантаження, а 0,16 мА для ESP32 – у режимі глибокого сну (в активному режимі споживання ESP32 значно залежить від увімкнених периферій, але за відсутності передачі Wi-Fi становить ~20 мА). Всі вказані плати (окрім STM32) мають вбудований інтерфейс USB для програмування, STM32 Blue Pill потребує зовнішнього USB-UART або ST-Link для початкового завантаження прошивки.

На основі наведеного порівняння можна виділити низку причин, чому Arduino Uno є слушним вибором для проєкту ультразвукового радара з датчиком HC-SR04 та сервоприводом SG906 [19]:

– Arduino Uno має 14 цифрових та 6 аналогових входів – більш ніж достатньо для підключення одного ультразвукового датчика (потрібні 2 цифрові лінії для Trig/Echo) та одного сервоприводу (1 PWM-вихід). Залишається великий запас пінів для додаткових компонентів (наприклад, індикаторів, дисплея тощо). На відміну від Raspberry Pi чи інших одноплатних комп'ютерів, Uno забезпечує простий детермінований контроль GPIO в реальному часі, що важливо для точного вимірювання часу польоту звуку. Його 16-МГц тактова частота та апаратні таймери дозволяють точно відраховувати мікросекундні імпульси без багатозадачних затримок;

– Arduino Uno побудований на енергоефективному мікроконтролері ATmega328P. У типовому режимі плата споживає близько 19 мА, максимум – до 50 мА. Відсутність ресурсомістких модулів (Wi-Fi, відеовиходів тощо) означає відносно мале енергоспоживання, що важливо для автономних або батарейних застосувань. Дослідження відзначають низьке споживання як значну перевагу Uno в порівнянні з потужнішими контролерами у простих задачах. Наприклад, Raspberry Pi Pico споживає менше в сплячому режимі (≈ 6 мВт проти 15 мВт у Uno), але при активній роботі Pico може споживати більше через вищу тактову частоту та двоядерну архітектуру. Uno ж забезпечує хорошу автономність без складних схем енергозбереження;

– екосистема Arduino була спроектована з ухилом на легкість освоєння. Arduino Uno повністю підтримується Arduino IDE, яка надає простий інтерфейс

для написання і завантаження скетчів (програм) на плату. Бібліотеки для роботи з ультразвуковим датчиком (наприклад, популярна бібліотека NewPing) та сервоприводом (стандартна бібліотека Servo) доступні із мінімальними зусиллями. У середовищі Arduino багато деталей (ініціалізація таймерів, налаштування портів) приховані, що дозволяє сконцентруватися на логіці проекту. У випадку ультразвукового радара є численні доступні приклади коду під Arduino Uno, що пришвидшує розробку;

– за більш ніж десятиліття існування Arduino Uno навколо нього сформувався багатий набір готових бібліотек для будь-яких модулів. Зокрема, для HC-SR04 можна використовувати як пряме вимірювання функцією `pulseIn()`, так і сторонні бібліотеки, що спрощують обчислення дистанції. Для сервоприводів в Arduino IDE є вбудована бібліотека `Servo.h`, яка дозволяє задавати кут повороту в градусах без детального програмування ширини імпульсу. Це суттєво полегшує інтеграцію компонентів: розробник може використати готові рішення з відкритим кодом. Спільнотою Arduino накопичено великий банк знань – від офіційних туторіалів до форумів. Uno має найбільшу кількість навчальних матеріалів і проектів у відкритому доступі серед всіх плат Arduino, що підтверджується його статусом «the most used and documented board» за даними офіційної документації. Це означає, що для задачі побудови радара легко знайти приклади та адаптувати їх під свої потреби, мінімізуючи ризик помилок;

– Arduino Uno працює на рівні логіки 5 В і має лінійні стабілізатори живлення на 5 В та 3,3 В, що дозволяє жити різноманітні модулі. Більшість типових сенсорів і виконавчих пристроїв розраховані саме на 5-вольтове живлення (історично склалося в аматорській електроніці). Ультразвуковий модуль HC-SR04, зокрема, потребує 5 В для надійної роботи та видає логічний сигнал ЕСНО також 5-вольтовий. Uno може напряму зчитувати цей сигнал, тоді як для мікроконтролерів на 3,3 В (ESP32, STM32, Pico) необхідно передбачати захист входу (ділник напруги, спеціальна версія датчика чи інший ультразвуковий модуль). Крім того, UNO має стандартний роз'єм живлення

«барель» для подачі 7-12 В, що спрощує живлення мобільного пристрою від батареї чи адаптера. Також форм-фактор Uno передбачає підтримку шилдів – плат-розширень, які можуть додавати функціональність (наприклад, екран для відображення радара, модуль Bluetooth для передачі даних тощо) просто встановленням на роз'єми Uno. Хоч у цьому проекті шилди не задіяні, сама можливість легкої апаратної розширюваності є плюсом. Сервопривід SG90 також сумісний з логікою 5 В і живленням 5 В, тому управління ним від Uno є тривіальним: достатньо підключити живлення до 5V/GND та сигнальний вивід до PWM-виходу Arduino.

Порівняльний аналіз показує, що Arduino Uno володіє збалансованим набором характеристик, достатніх для реалізації ультразвукової радарної системи малого радіуса дії. Uno поступається сучасним 32-розрядним платформам у максимальній швидкодії та обсягах пам'яті, але перевершує їх за простотою, сумісністю та наявністю підтримки. У контексті задачі з датчиком HC-SR04 та сервоприводом, його 5-вольтова архітектура і готові рішення дозволяють швидко та надійно сконструювати робочу систему.

2.1.2 Аналіз ультразвукового сенсора HC-SR04

Ультразвуковий сенсор HC-SR04 працює за принципом ехолокації. Після подачі на вхід Trig імпульсу тривалістю 10 мкс, модуль генерує 8 ультразвукових імпульсів з частотою 40 кГц. Ці імпульси відбиваються від об'єктів і повертаються до приймача. Час між передачею та прийомом імпульсу дозволяє обчислити відстань до об'єкта за формулою (2.1):

$$\text{Відстань (см)} = (\text{Час затримки (мкс)} \times 0,0343) / 2 \quad (2.1)$$

Цей метод забезпечує точне вимірювання відстані в межах від 2 см до 400 см з точністю до 3 мм [25].

Для підключення HC-SR04 до Arduino Uno використовуються чотири виходи:

- VCC – підключається до 5 В на Arduino;

- GND – підключається до GND на Arduino;
- Trig – підключається до цифрового піну (наприклад, D9);
- Echo – підключається до цифрового піну (наприклад, D10).

Для забезпечення гнучкості проєктування та адаптації ультразвукового радара до різних умов експлуатації варто розглянути альтернативи сенсору HC-SR04. У таблиці 2.2 наведено ключові особливості популярних моделей ультразвукових сенсорів, які застосовуються в мікроконтролерних системах, зокрема на платформі Arduino.

Таблиця 2.2 – Порівняння з аналогами [26]

Модель	Діапазон (см)	Кут огляду	Особливості
HC-SR04	2-400	~15°	базова модель, низька вартість
JSN-SR04T	20-600	~30°	водонепроникний, один трансдюсер
MaxBotix HRLV-EZ1	30-500	~42°	висока точність, аналоговий вихід

Отже, HC-SR04 залишається оптимальним вибором завдяки простоті, низькій вартості та достатній точності, тоді як його аналоги доцільні у спеціалізованих умовах, що потребують захисту від вологи чи розширеного діапазону.

2.1.3 Вибір сервоприводу для сканування SG90

Сервоприводи – це електромеханічні пристрої, які забезпечують обертання або позиціонування вала в межах заданого кута. В основі роботи сервопривода лежить принцип імпульсно-широотно-модульованого (PWM) управління, коли кут повороту вала залежить від ширини вхідного імпульсу. Наприклад, для стандартного 180° сервопривода (як SG90), імпульс тривалістю 1 мс відповідає положенню 0°, 1,5 мс – 90°, а 2 мс – 180° [27].

Сервоприводи ідеально підходять для задач, де потрібно точно контролювати кут повороту, зокрема, для обертання ультразвукового датчика HC-SR04 у складі мобільного або стаціонарного радара, що сканує навколишній простір.

SG90 – один із найбільш популярних мікросервоприводів у проєктах з Arduino (рис. 2.2). Його основними характеристиками є компактність, мала вага, низьке енергоспоживання і досить висока точність позиціонування [27].

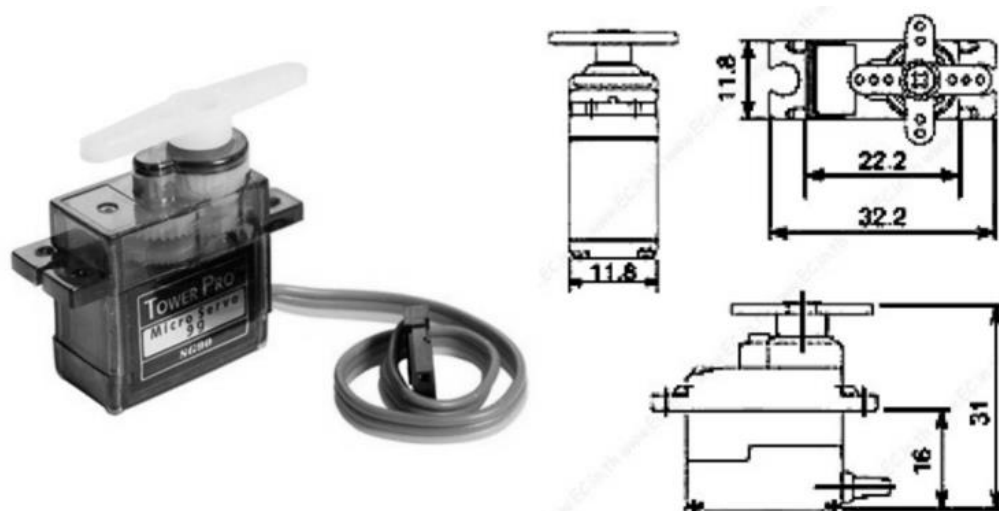


Рисунок 2.2 – Мікросервопривід SG90 [27]

SG90 має три дроти: коричневий (GND), червоний (VCC) та помаранчевий (Signal). Сервопривод підключається до Arduino Uno наступним чином:

- GND → GND на Arduino;
- VCC → 5V на Arduino;
- Signal → будь-який цифровий пін (зазвичай D9) [27].

Для забезпечення належної продуктивності та надійності системи обертання ультразвукового сенсора важливо обґрунтовано підійти до вибору сервопривода. Оскільки на ринку представлено кілька популярних моделей – таких як SG90, MG90S та MG996R, доцільно виконати порівняльний аналіз їхніх технічних характеристик. У таблиці 2.3 наведено ключові параметри цих сервомоторів.

Таблиця 2.3 – Порівняння параметрів сервомоторів [27]

Параметр	SG90	MG90S	MG996R
матеріал шестерень	пластик	метал	метал
крутний момент	1,8 кг·см	2,2 кг·см	10-13 кг·см
напруга	4,8-6,0 В	4,8-6,0 В	4,8-7,2 В
вага	~9 г	~13 г	~55 г
кут обертання	180°	180°	180°
споживання струму	до 650 мА	до 800 мА	до 2,5 А (під навантаженням)
ціна (орієнтовно)	\$1-3	\$3-6	\$7-10

З наведеного порівняння видно, що сервопривід SG90 є найкращим вибором для легких, малопотужних застосувань з обмеженим простором, тоді як

MG90S і MG996R доцільні для проєктів, що потребують вищої механічної міцності та крутного моменту.

2.1.4 Периферійні компоненти

Для повноцінного функціонування ультразвукового радара на базі Arduino, окрім основних елементів (Arduino Uno, HC-SR04, SG90), необхідно застосовувати допоміжні периферійні компоненти, які забезпечують зручність розробки, виведення інформації та стабільне живлення системи.

Найпоширенішими периферійними елементами є:

- дисплеї (LCD, OLED) – для виводу результатів вимірювання відстані в реальному часі;
- живлення (адаптери, батарейні блоки, USB) – забезпечення стабільної роботи в польових умовах;
- макетні плати, дроти, резистори – для зручного складання прототипу без паяння;
- кнопки, світлодіоди, бузери – для керування режимами сканування або індикації подій;
- бездротові модулі (опційно) – наприклад, Bluetooth HC-05 для передавання даних на смартфон, або ESP8266 для надсилання результатів сканування в мережу Wi-Fi.

Для забезпечення повноцінного функціонування ультразвукового радара на базі Arduino необхідно використовувати низку допоміжних апаратних елементів. Ці компоненти не лише забезпечують зручність у побудові прототипу, а й розширюють функціональні можливості системи – від виводу інформації на дисплей до організації бездротового зв'язку. У таблиці 2.4 представлено основні периферійні модулі, які застосовуються в проєкті, разом із прикладами їхнього практичного призначення.

Таблиця 2.4 – Периферійні компоненти ультразвукового радара [28]

Компонент	Призначення	Приклад моделі
1	2	3
LCD-дисплей	виведення результатів сканування	LCD 1602 + I2C модуль
макетна плата	збирання прототипу без пайки	Breadboard 830 точок

Продовження таблиці 2.4

1	2	3
проводи підключення	комунікація між компонентами	Dupont Male-Male
джерело живлення	автономне або стаціонарне живлення	USB, 9В батарея, адаптер 5 В
Bluetooth модуль	бездротова передача даних на смартфон	HC-05
Wi-Fi модуль	підключення до локальної мережі	ESP8266 (ESP-01)
світлодіод	індикація подій (наприклад, тривога)	LED 5 мм
кнопка	перемикання режимів сканування	Tact Switch

Отже, застосування периферійних компонентів дозволяє адаптувати ультразвуковий радар до конкретних умов використання, підвищити його функціональність, зручність взаємодії та здатність до інтеграції з іншими пристроями.

2.2 Схемотехнічне рішення компонентів до Arduino

Ультразвуковий радар на базі Arduino Uno реалізується шляхом поєднання основних та периферійних модулів. Основу складають мікроконтролер Arduino Uno, ультразвуковий сенсор HC-SR04 і сервопривід SG90, що обертає сенсор у горизонтальній площині для сканування простору. Допоміжні компоненти – такі як LCD-дисплей, кнопка, світлодіоди та бездротові модулі (HC-05, ESP8266) – розширюють функціональність пристрою, дозволяючи виводити результати, взаємодіяти з користувачем або передавати дані на віддалені пристрої.

Всі елементи підключаються до Arduino Uno з урахуванням обмежень по живленню, сигналам та інтерфейсам (I2C, PWM, UART). З'єднання здійснюються через макетну плату з використанням Dupont-дротів.

Щоб забезпечити коректну взаємодію всіх компонентів ультразвукового радара на базі Arduino Uno, необхідно чітко визначити призначення кожного виводу мікроконтролера та відповідних периферійних модулів. У таблиці 2.5 наведено стислий перелік основних з'єднань, який допоможе зрозуміти логіку підключення та спростить реалізацію проєкту як у прототипі, так і в фінальній версії пристрою.

Таблиця 2.5 – Перелік основних з'єднань компонентів ультразвукового радара з Arduino Uno [29]

Компонент	Пін на Arduino Uno	Примітка
HC-SR04 Trig	D9	цифровий вихід для запуску сигналу
HC-SR04 Echo	D8	вхід для прийому відбитого сигналу
SG90 сервопривід	D10	PWM-керування положенням сенсора
LCD 1602 (SDA)	A4	передача даних по шині I2C
LCD 1602 (SCL)	A5	синхронізація годинника по I2C
кнопка	D2	перемикання режимів сканування
світлодіод	D3	індикація подій
HC-05 TXD	D4	передача даних на Arduino (UART)
HC-05 RXD	D5	прийом даних з Arduino (через дільник)
ESP8266 TX	D6	передача даних до Arduino
ESP8266 RX	D7	прийом з Arduino (через дільник напруги)
усі модулі (VCC)	5 V або 3,3 V	живлення відповідно до вимог модуля
усі модулі (GND)	GND	спільна «земля»

Таким чином, чітке визначення з'єднань між компонентами дозволяє забезпечити стабільну роботу ультразвукового радара, уникнути конфліктів при спільному використанні інтерфейсів та полегшує процес налаштування й налагодження системи як для стаціонарного, так і для мобільного застосування.

На рисунку 2.3 представлено структурну схему з'єднань компонентів ультразвукового радара.

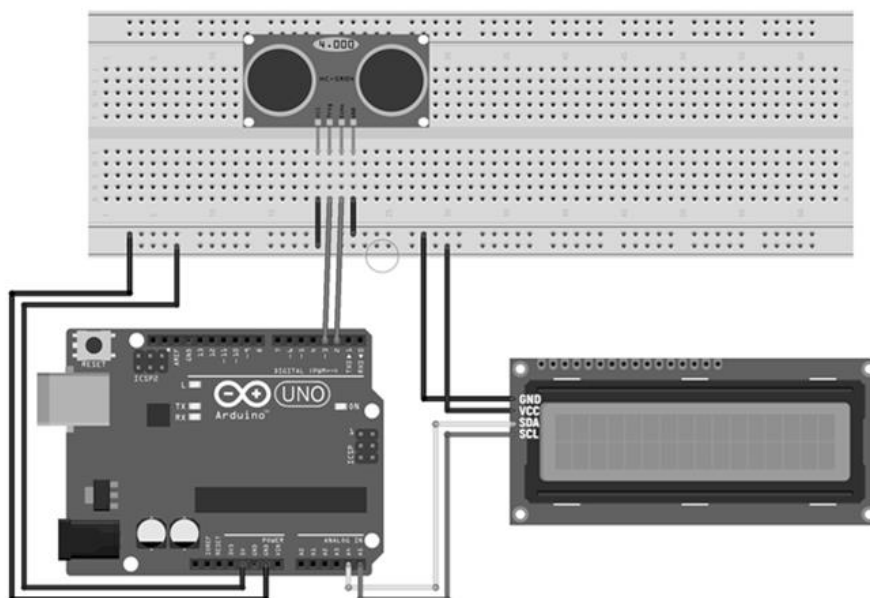


Рисунок 2.3 – Загальна електрична схема підключення компонентів до Arduino Uno (з додатковими модулями HC-05 та ESP8266) [29]

Схемотехнічне рішення ультразвукового радару демонструє гнучку модульну архітектуру, засновану на поєднанні стандартних інтерфейсів Arduino Uno: цифрових входів/виходів, PWM, I2C та UART. Такий підхід забезпечує легку інтеграцію сенсорів, сервомеханізмів та засобів виводу або зв'язку.

2.3 Можливі варіанти інтеграції системи в мобільні або стаціонарні рішення

Ультразвуковий радар на базі Arduino має широкі можливості інтеграції в різні типи пристроїв – як мобільні, так і стаціонарні. Гнучкість архітектури системи, доступність модулів і простота налаштування дозволяють адаптувати рішення під конкретні потреби користувача або середовища застосування.

У мобільних рішеннях (роботи, гусеничні платформи, дрони) радарна система відіграє роль навігаційного сенсора:

- уникнення зіткнень – сенсор постійно сканує простір і фіксує перешкоди на шляху руху;
- побудова карти середовища – за допомогою обертання сенсора створюється полярна діаграма оточення;
- автономна навігація – у поєднанні з алгоритмами керування мікроконтролер може приймати рішення про зміну напрямку руху [30].

Система живиться від акумуляторів (наприклад, Li-ion 7,4 В з модулем стабілізації), з можливістю передавати дані по Bluetooth або Wi-Fi на інші вузли системи.

У фіксованих установках ультразвуковий радар може виконувати такі функції:

- паркувальні системи – вимірювання відстані до перешкод з виведенням на дисплей або передачею до автомобільної системи;
- контроль доступу – активація звукового або світлового сигналу при наближенні об'єкта до контрольованої зони;

– смарт-домівки – вмикання світла або пристроїв за рухом об’єкта (наприклад, відкриття дверей, запуск вентиляції);

– логістичні центри – контроль простору біля вантажних рамп або автоматизованих зон розвантаження [30].

Такі системи можуть бути живлені від мережі або USB-адаптера, інтегруватися з контролерами доступу, охоронними системами або IoT-хабами.

Залежно від середовища використання та вимог до функціональності, ультразвуковий радар може бути адаптований до різних типів пристроїв. У таблиці 2.6 наведено узагальнене порівняння типових варіантів інтеграції системи.

Таблиця 2.6 – Порівняння варіантів інтеграції ультразвукового радара [30]

Застосування	Тип платформи	Живлення	Передача даних	Основна функція
робот-сканер	мобільна	акумулятор 7,4 В	Bluetooth / Wi-Fi	навігація та уникнення перешкод
гусеничний танк	мобільна	Power Bank (5 В)	серійний порт	стеження за об’єктами
стаціонарна сигналізація	стаціонарна	мережевий адаптер	без передачі або Wi-Fi	виявлення руху біля дверей/вікон
паркувальна система	стаціонарна	12 В адаптер	LCD / CAN / UART	відображення відстані до перешкоди
смарт-дім (автоматизація)	стаціонарна	USB 5 В	MQTT / Wi-Fi	увімкнення світла, керування пристроями

Як показано в таблиці 2.6, універсальність ультразвукового радара на базі Arduino дозволяє ефективно адаптувати його до найрізноманітніших застосувань – від мобільних автономних платформ до стаціонарних систем автоматизації та безпеки.

У мобільному рішенні ультразвуковий радар інтегрується до платформи, яка пересувається в просторі, наприклад, гусеничного робота або робота на колесах. До складу входить мікроконтролер Arduino Uno, ультразвуковий сенсор HC-SR04 для визначення відстані, сервопривід SG90 для повороту сенсора, а також модуль Bluetooth HC-05, що забезпечує бездротову передачу даних на смартфон або інший контролер. Живлення здійснюється від акумулятора 7,4 В або Power Bank, через стабілізатор напруги до 5 В. Така схема дозволяє

реалізувати систему об'їзду перешкод, побудову карти оточення або дистанційне керування.

У стаціонарному варіанті схема більш орієнтована на моніторинг простору в межах фіксованої зони, наприклад, у кімнаті, біля дверей або в зоні паркування. У цьому випадку до Arduino Uno підключається датчик HC-SR04, а результати виводяться на LCD-дисплей з інтерфейсом I2C. Для передачі даних у локальну мережу використовується модуль ESP8266, що дозволяє надсилати показники відстані, наприклад, на вебінтерфейс або до хмарного сервісу. Живлення в такій схемі зазвичай здійснюється через стаціонарний адаптер живлення на 5 В, що забезпечує стабільну безперебійну роботу.

Як бачимо, завдяки універсальності платформи Arduino та широкій сумісності з периферією, ультразвуковий радар легко інтегрується в різноманітні типи систем.

РОЗДІЛ 3

РОЗРОБКА ПРОГРАМНО-АПАРАТНОЇ ЧАСТИНИ УЛЬТРАЗВУКОВОГО РАДАРА

3.1 Огляд апаратного забезпечення

Для розробки апаратної частини було вибрано ряд комплектуючих:

- мікроконтролер Arduino Uno (рис. 3.1);

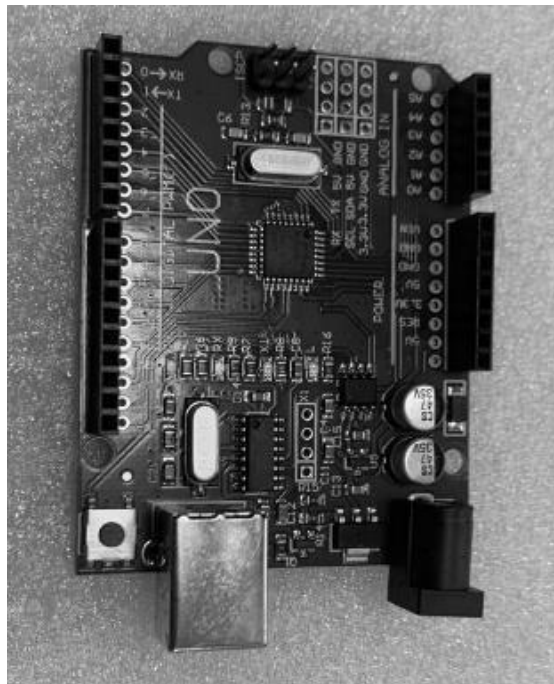


Рисунок 3.1 – Мікроконтролер Arduino Uno

- ультразвуковий модуль HC-SR04 (рис. 3.2);

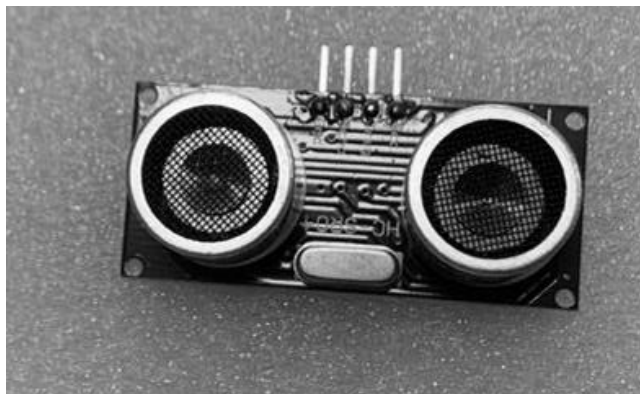


Рисунок 3.2 – ультразвуковий модуль HC-SR04

- сервопривід TS90A (рис. 3.3);

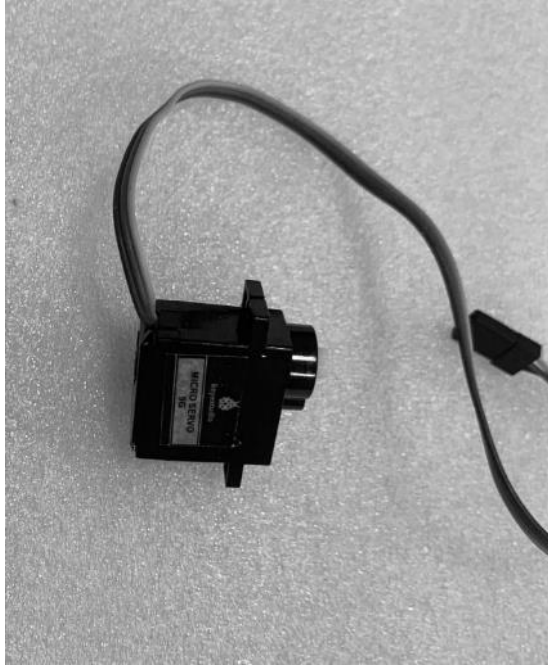


Рисунок 3.3 – сервопривід TS90A

- основа для ультразвукового радара (рис. 3.4);

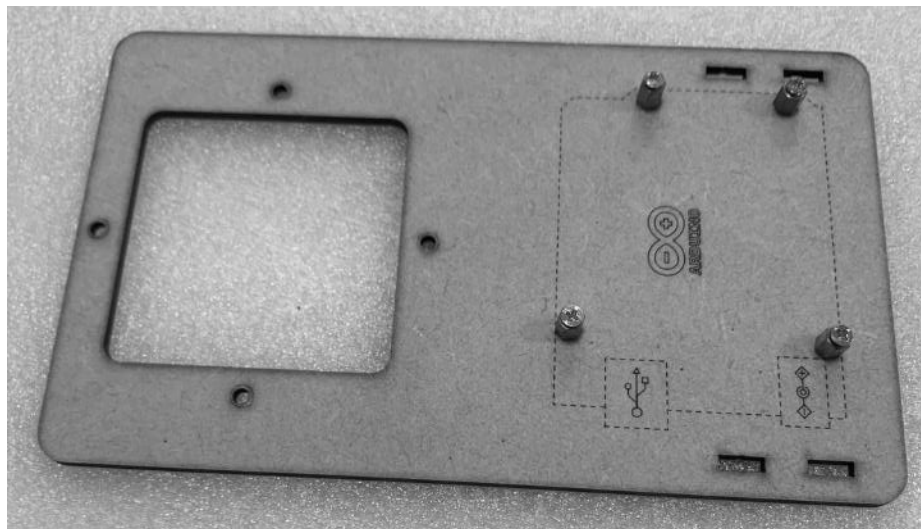


Рисунок 3.4 – Основа для ультразвукового радара

- кріпильні елементи та дюпони (рис. 3.5).

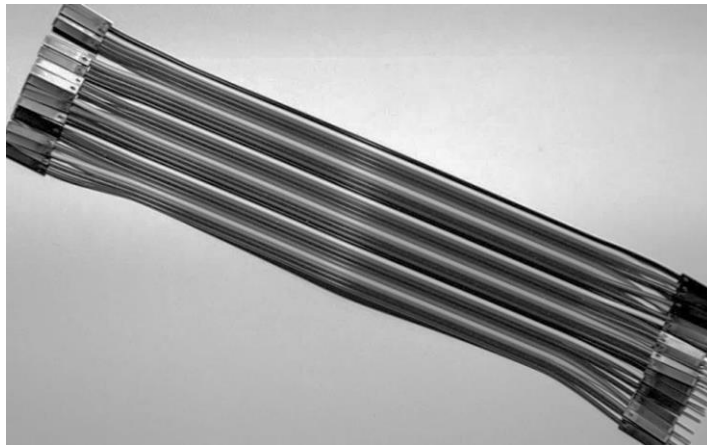


Рисунок 3.5 – Дюпони

3.2 Складання ультразвукового радара на основі мікроконтролера

На першому етапі було проведено підготовку всіх необхідних компонентів для збирання макету ультразвукового радара на базі мікроконтролера. Для реалізації конструкції використовувалися різні допоміжні елементи: монтажна основа, кріпильні гвинти, мідні дистанційні стійки та інші комплектуючі, які представлено на рисунку 3.6.

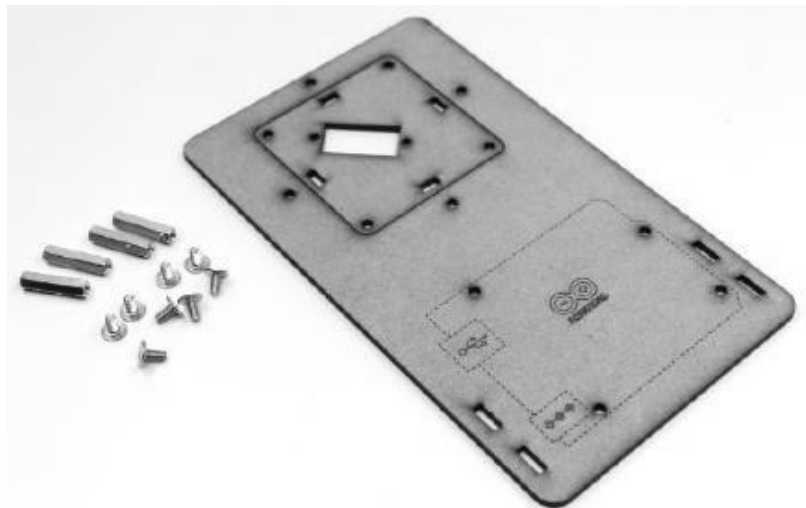


Рисунок 3.6 – Елементи кріплення міні-радара

На початковому кроці було здійснено фіксацію подвійних мідних стійок, які гвинтами закріплювалися до монтажної основи, як представлено на рисунку 3.7.

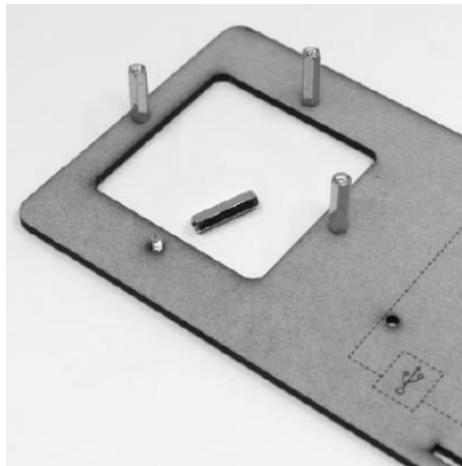


Рисунок 3.7 – Кріплення стійок до основи

Наступним кроком стало встановлення сервоприводу на основу, який згодом слугує механізмом обертання ультразвукового датчика HC-SR04. Сервопривід фіксувався за допомогою спеціальних гвинтів, при цьому його поворотний вал мав бути розташований у центральній точці конструкції, як на рисунку 3.8.

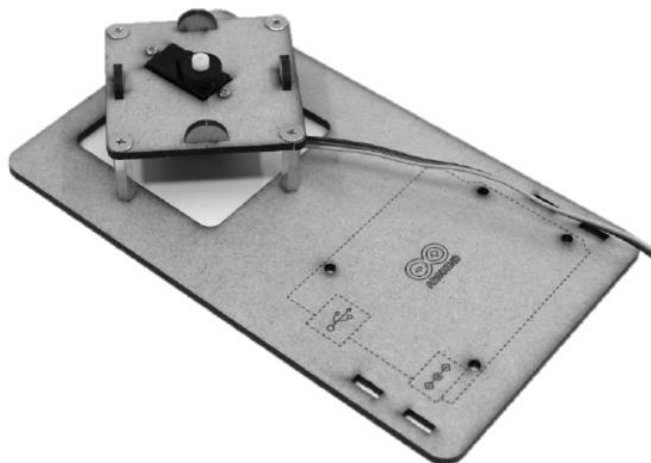
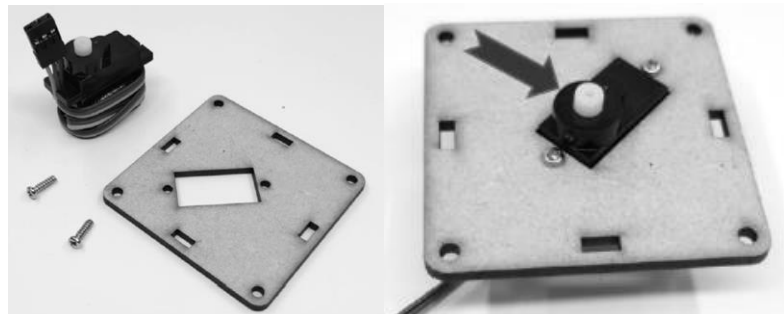


Рисунок 3.8 – Монтаж сервоприводу

Третій етап полягав у формуванні радіолокаційного вузла, де буде встановлено ультразвуковий модуль (рис. 3.9).

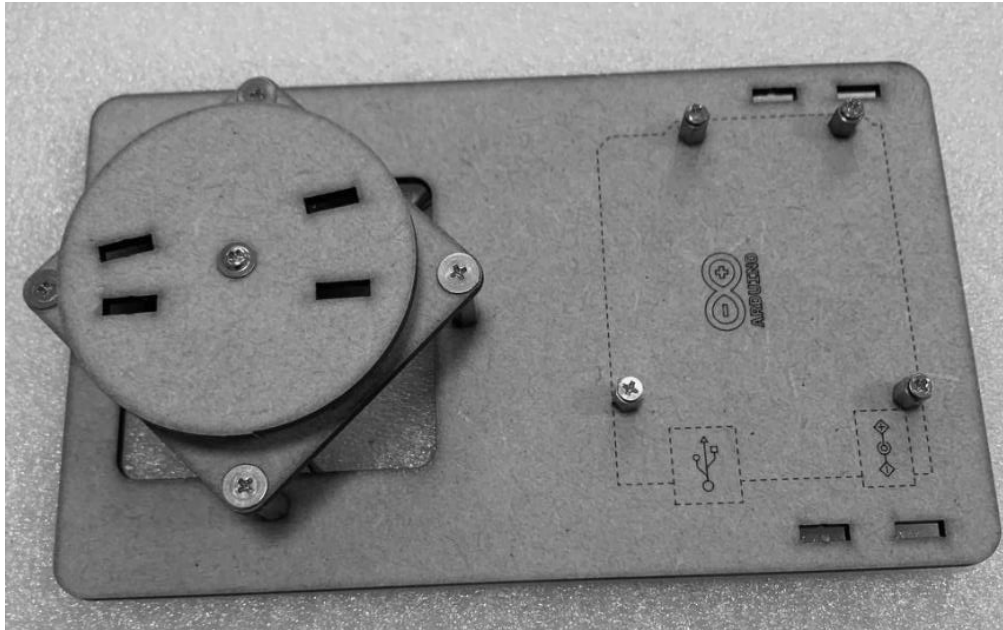


Рисунок 3.9 – Радіолокаційний вузол

На четвертому етапі на основу закріплювалася мікроконтролерна плата Arduino UNO. Для цього застосовувалися короткі мідні стійки, що забезпечували надійну фіксацію плати, як на рисунку 3.10.

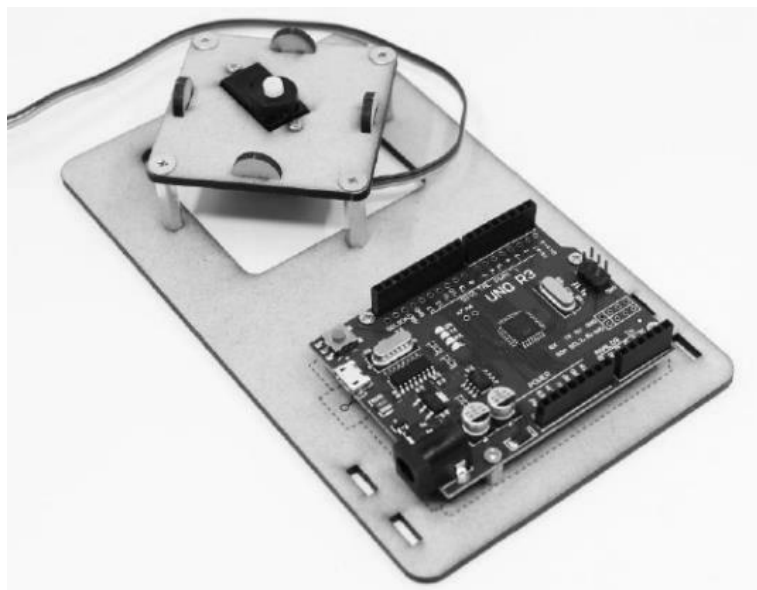


Рисунок 3.10 – Встановлення плати Arduino UNO

Після цього здійснювалося підключення модуля HC-SR04 дюпон (рис. 3.11), що забезпечувало електричну комутацію між окремими компонентами системи.

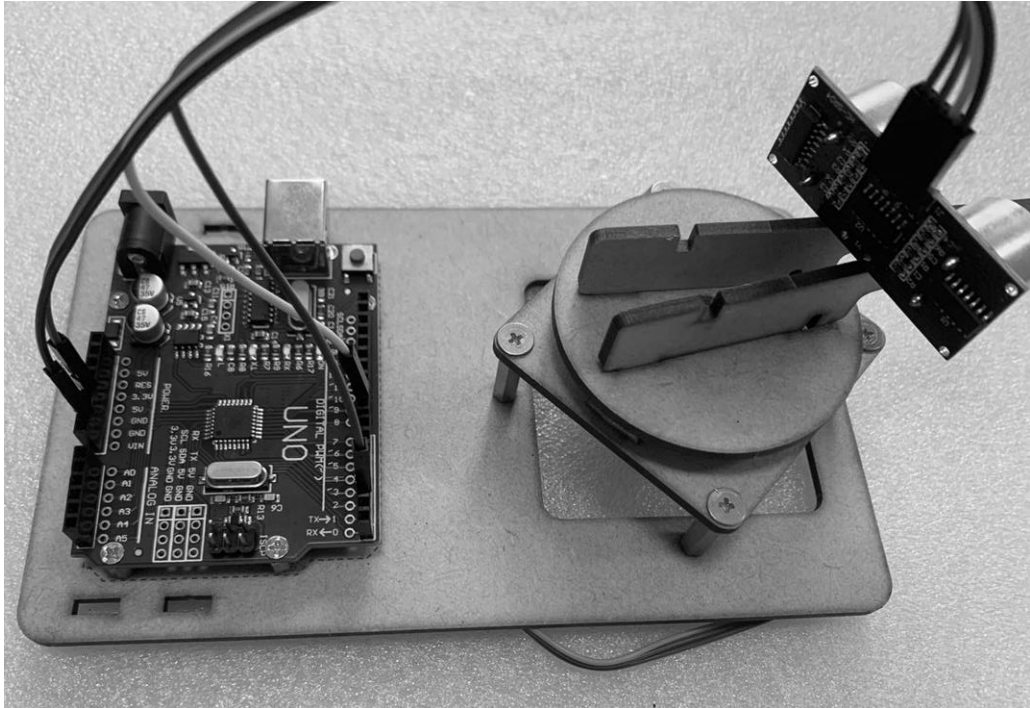


Рисунок 3.11 – Монтаж та підключення модуля HC-SR04

На наступному кроці, використовуючи важіль рульового механізму, визначався робочий діапазон обертання, після чого основа радара фіксувалася у потрібному положенні (рис. 3.12).

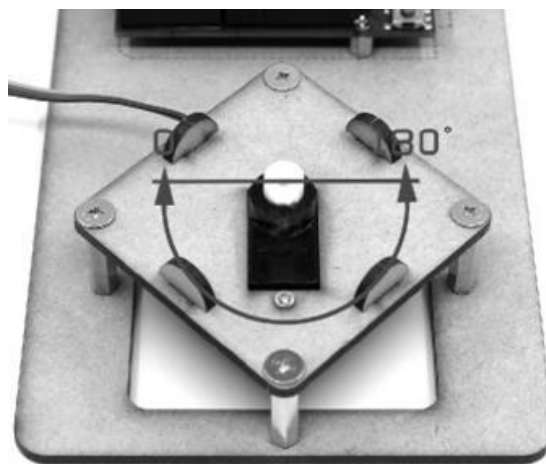


Рисунок 3.12 – Налаштування діапазону обертання

Після встановлення всіх комплектуючих ультразвукового радара підключаються всі складові, як на рисунку 3.13. Тоді макет набуде вигляду, як представлено на рисунку 3.14.

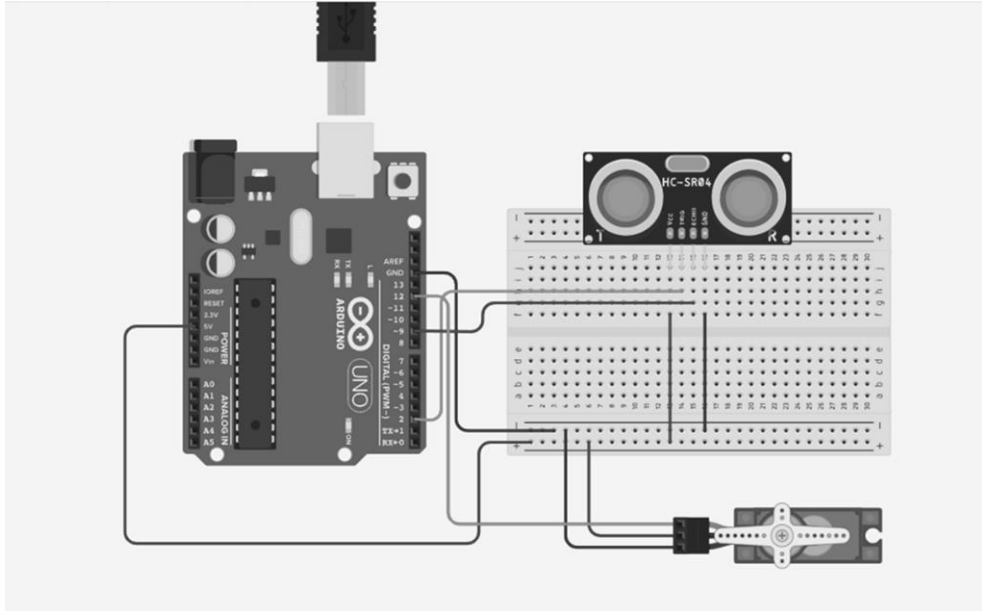


Рисунок 3.13 – Схема підключення ультразвукового радара

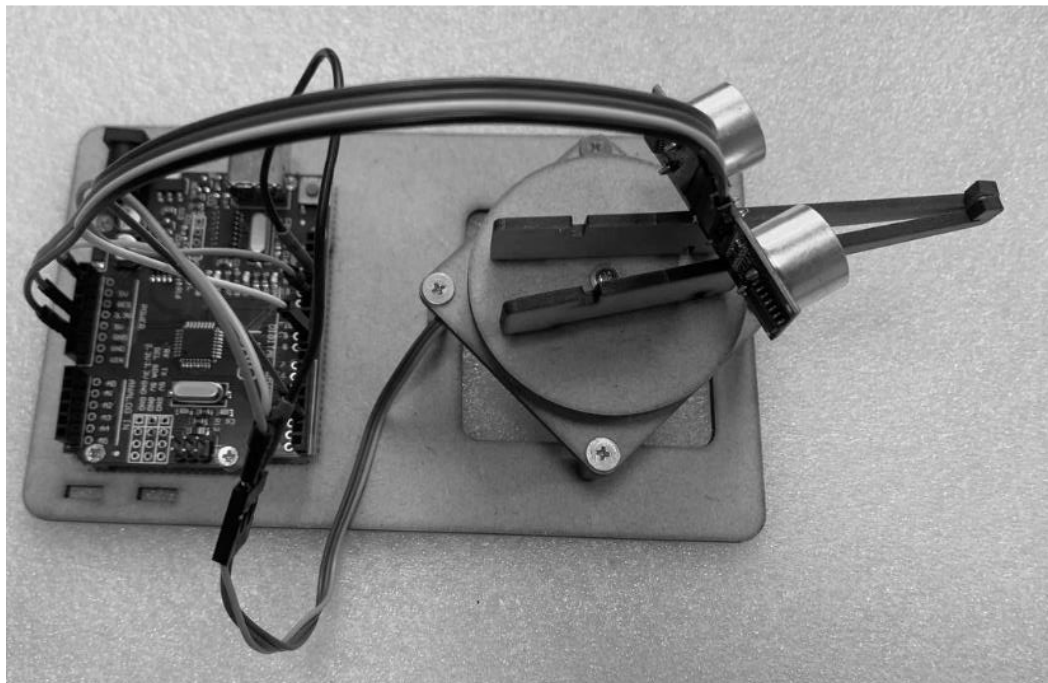


Рисунок 3.14 – Макет ультразвукового радара на основі мікроконтролера

На завершальному етапі вал сервоприводу вводився в центральний отвір основи радіолокатора. Для уникнення пошкодження деталей при монтажі рульовий механізм утримувався руками, після чого фіксувався гвинтом.

3.3 Візуалізація радарних даних на екрані або в реальному часі на ПК

Візуалізація радарних даних є одним із ключових етапів у роботі ультразвукового радара на основі мікроконтролера Arduino Uno. Отримана інформація про відстань до об'єктів не має практичної цінності без зручного способу її представлення користувачеві. Сучасні системи дозволяють реалізувати два основні підходи: відображення даних безпосередньо на фізичному дисплеї, підключеному до Arduino, або передавання інформації через послідовний порт на комп'ютер для візуалізації в реальному часі.

На першому етапі можна застосувати, наприклад, 4-розрядний дисплей TM1637 або LCD-дисплей 16x2, підключений до Arduino. Для цього Arduino послідовно передає значення, отримані з ультразвукового сенсора, на дисплей, де вони відображаються у вигляді цифрового значення відстані (рис. 3.15). У випадку використання дисплея це було б актуально, але в нашому випадку було використано ПК для відображення даних.

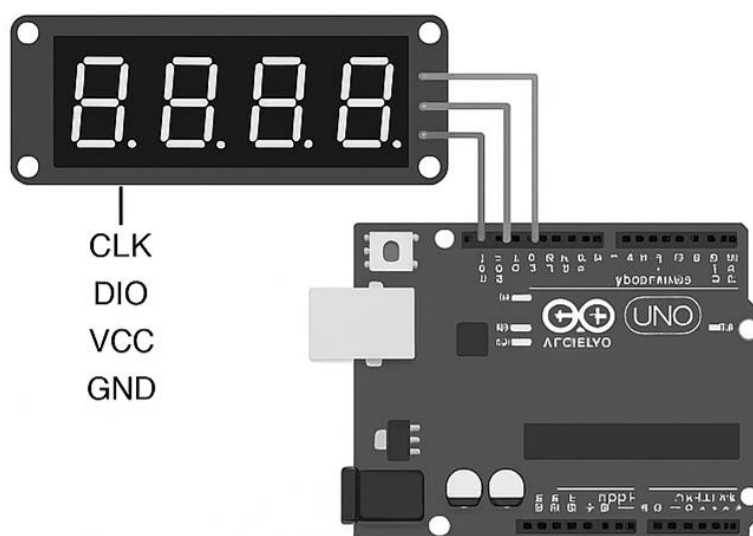


Рисунок 3.15 – Схема підключення дисплея TM1637 до Arduino.

В іншому підході дані передаються через послідовний порт (Serial) на комп'ютер, де відбувається обробка та візуалізація за допомогою спеціального програмного забезпечення. Для реалізації цього методу можна використовувати середовище Processing, яке здатне читати серійні дані від Arduino та створювати графічне представлення радарного променя на екрані комп'ютера. Програма в середовищі Processing обробляє значення відстані та кута, генеруючи радіальну діаграму, яка оновлюється в режимі реального часу (рис. 3.16).

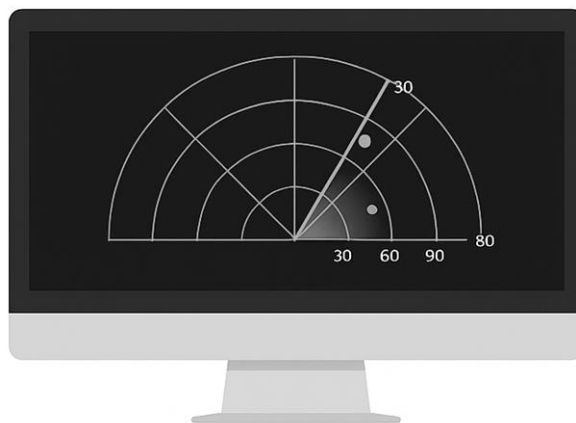


Рисунок 3.16 – Приклад візуалізації даних радара на екрані ПК у середовищі Processing

Особливістю такого підходу є можливість масштабування даних, накладання графічних шарів, а також інтерактивності, наприклад, зміни параметрів відображення безпосередньо з комп'ютера. Крім того, використання ПК для візуалізації дає змогу вести запис отриманих даних, що важливо для подальшого аналізу та досліджень.

На практиці система формує дані про відстань у вигляді пари значень (кут, радіус), які Arduino передає через порт. На комп'ютері створюється полярна діаграма, де кожна точка на графіку відповідає виявленому об'єкту. Це дозволяє користувачеві бачити розташування об'єктів у просторі у зручному та наочному вигляді (рис. 3.17).

Таким чином, візуалізація даних є важливим компонентом ультразвукового радара, що перетворює сирі цифрові дані на зрозумілу

інформацію. Вибір способу візуалізації залежить від завдань проєкту: для простих рішень підходить локальний дисплей, тоді як для складніших досліджень або демонстрацій ефективніше використовувати комп'ютерну візуалізацію в реальному часі.



Рисунок 3.17 – Схематичне зображення полярної діаграми радара

3.4 Програмування мікроконтролера для керування ультразвуковим радаром

Для програмування плати Arduino Uno використовується спеціальне інтегроване середовище розробки – Arduino IDE. Після встановлення цього програмного забезпечення на персональний комп'ютер необхідно підключити плату Arduino Uno за допомогою USB-кабелю. Далі у середовищі Arduino IDE потрібно обрати відповідну плату, перейшовши в меню «Інструменти» → «Плати» → «Arduino/Genuino Uno», а також вказати правильний COM-порт через меню «Інструменти» → «Порт».

Програмування Arduino Uno здійснюється мовою Arduino, що базується на платформі Wiring. Для першого запуску плати та перевірки її працездатності зазвичай використовують демонстраційний приклад – програму для керування вбудованим світлодіодом. Починаючи підключення обладнання потрібно запам'ятати номер контакту під час написання коду.

Було створено функцію для вимірювання відстані від датчика, а потім у циклі дав команду обертати серводвигун зліва направо, як представлено у лістингу 3.1.

Лістинг 3.1 – Функція для вимірювання відстані від датчика

```
#include<Servo.h>
#define trigPin 8
#define echoPin 9
long duration;
int distance ;
Servo myservo;
int calculateDistance()
{
    digitalWrite(trigPin,LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin,HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin,LOW);
    duration = pulseIn(echoPin, HIGH);
    distance = duration*0.034/2;
    return distance;
}
void setup()
{
    pinMode(trigPin , OUTPUT);
    pinMode(echoPin, INPUT);
    myservo.attach(11);
    Serial.begin(9600);
}
void loop()
{
    int i ;
    for (i=15; i<=165; i++)
    {
        myservo.write(i);
        delay(15);
        calculateDistance();
        Serial.print(i);
        Serial.print(",");
        Serial.print(distance);
        Serial.print(".");
    }
    for(i=165; i>=15; i--)
    {
        myservo.write(i);
        delay(15);
        calculateDistance();
        Serial.print(i);
        Serial.print(",");
        Serial.print(distance);
        Serial.print(".");
    }
}
```

кінець лістингу 3.1

Щоб завантажити цей приклад, необхідно вибрати в меню «Файл» → «Приклади» → «Основи» → «Blink». Після відкриття прикладу у середовищі розробки достатньо натиснути кнопку «Завантажити» на верхній панелі, щоб передати програму на мікроконтролер.

Якщо в заданому діапазоні немає об'єкта, що перед датчиком нічого немає на відстані приблизно 40 см, то на радарі відображаються всі зелені лінії, але шого боку, якщо ми розмістимо об'єкт у діапазоні 40 см, то датчик виявить його, і це буде позначено червоними лініями на радарі, інтерфейс представлено на рисунку 3.18.

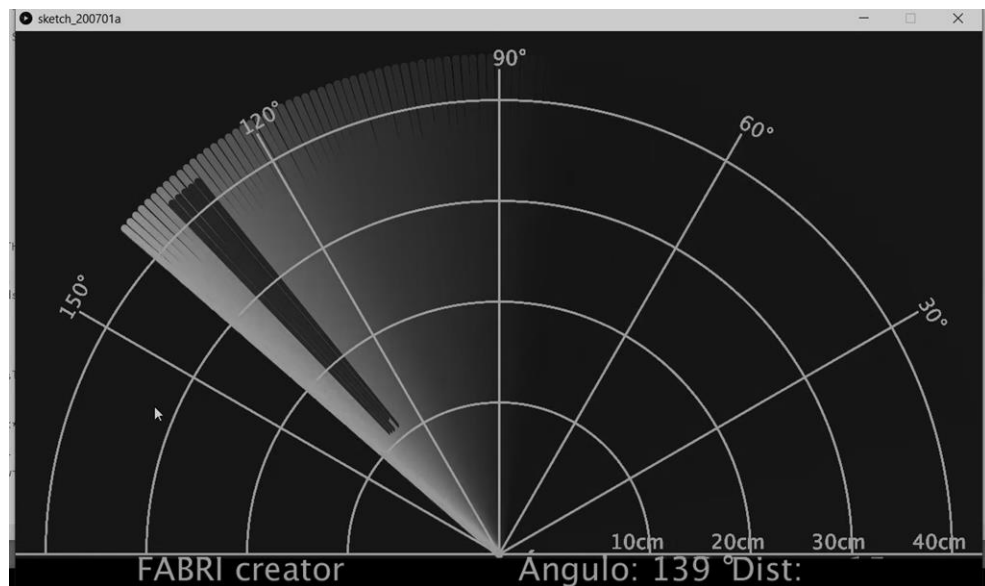


Рисунок 3.18 – Інтерфейс Processing IDE.

Радар відображається за допомогою середовища розробки Processing IDE. Ультразвуковий датчик встановлений на серводвигуні, який обертається вбік. Датчик сканує область в межах своєї відстані.

ВИСНОВКИ

У ході виконання бакалаврської кваліфікаційної роботи було досягнуто поставлену мету – реалізовано ультразвуковий радарний пристрій з використанням апаратної платформи Arduino, який дозволяє визначати відстань до об'єктів і візуалізувати дані у зручному форматі. Робота охопила всі ключові етапи: від теоретичного дослідження принципів дії ультразвукових сенсорів до практичного втілення апаратно-програмного прототипу.

Під час дослідження принципів роботи ультразвукових радарів було встановлено, що ці пристрої мають низку переваг: простота реалізації, відносно невелика вартість, безпечність для здоров'я людини та прийнятна точність при роботі на коротких відстанях. Ультразвук, як фізичне явище, надає змогу точно вимірювати дистанцію до об'єктів за допомогою аналізу часу проходження сигналу до об'єкта і назад.

У процесі аналізу сфер застосування ультразвукових радарів з'ясовано, що вони активно використовуються у транспорті (системи паркування), робототехніці (обхід перешкод), промисловості (моніторинг рівня речовин), а також у системах безпеки. Ці сфери обумовлюють високий попит на компактні й енергоефективні рішення.

Доцільність вибору мікроконтролера Arduino Uno була підтверджена завдяки його відкритій архітектурі, широкій підтримці бібліотек, простоті програмування та наявності великої спільноти розробників. Arduino є доступним і достатньо функціональним для реалізації задач ультразвукового зондування на базовому рівні.

На основі визначених технічних критеріїв (точність вимірювання, енергоспоживання, гнучкість підключень) було здійснено оптимальний підбір апаратних компонентів. Основу системи склали мікроконтролер Arduino Uno, ультразвуковий сенсор HC-SR04, сервопривід SG90 для обертання сенсора.

У завершальній частині проєкту розроблено електричну схему з'єднання усіх компонентів та зібрано повноцінний макет ультразвукового радару. Була створена програма в середовищі Arduino IDE, яка забезпечує збір даних з сенсора, обробку результатів та їх виведення у вигляді координат і відстаней на дисплеї або комп'ютері.

Таким чином, робота підтвердила можливість створення доступного та функціонального ультразвукового радарного пристрою, що може бути інтегрований у ширше коло розробок в галузі автоматизації, робототехніки та інтелектуальних систем спостереження.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Seda Üstün Ercan, Mohammed Sufyan Mohammed. Distance Measurement and Object Detection System Based On Ultrasonic Sensor and XBee. *Düzce University Journal of Science & Technology*. Vol. 8. 2020. P. 1706-1719.
2. R. Suryawanshi, A. Thosar, A. Singh, N. Mehendale. Object Detection System Using Arduino and Ultra Sonic sensor. *Journal of object detection system*. URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5009923 (дата звернення: 23.01.2025).
3. Waterproof JSN-SR04T Ultrasonic Distance Sensor with Arduino Tutorial. *Makerguides*. URL: <https://www.makerguides.com/jsn-sr04t-arduino-tutorial/> (дата звернення: 25.01.2025).
4. MaxBotix MB1040 LV-MaxSonar-EZ4 Ultrasonic Rangefinder. *3DMakerWorld*. URL: <https://3dmakerworld.com/products/ultrasonic-sensor-for-uavs-robots-autonomous-navigation-distance-measuring-and-more-mb1040-000-lv-maxsonar-ez4-range-from-6-inches-to-254-inches-maxbotix-inc> (дата звернення: 28.01.2025).
5. Qiongfeng Shi, Zhongda Sun, Xianhao Le, Jin Xie, Chengkuo Lee. Soft Robotic Perception System with Ultrasonic Auto-Positioning and Multimodal Sensory Intelligence. *ACS Nano*. Vol 17. Issue 5. 2023. URL: <https://pubs.acs.org/doi/abs/10.1021/acsnano.2c12592> (дата звернення: 02.02.2025).
6. Радіолокаційна станція (РЛС) або радар. *Вікіпедія*. URL: <https://uk.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B4%D0%B0%D1%80> (дата звернення: 08.02.2025).
7. Sh. Chowdhury, S. Mazumdar, S. Giri, A. Bhattacharya. Radar System Utilizing Arduino. *International Journal of Research Publication and Reviews*. Vol. 5. № 5. 2024. P. 12732-12738.
8. Angona Biswas, Sabrina Abedin, Md. Ahasan Kabir. Moving Object Detection Using Ultrasonic Radar with Proper Distance, Direction, and Object Shape

Analysis. *Journal of Information Systems Engineering and Business Intelligence*. Vol. 6. № 2. 2020. P. 99-111.

9. Ultrasonic Radar with Arduino. *Arduino*. URL: <https://projecthub.arduino.cc/nimishac/ultrasonic-radar-with-arduino-19baa3> (дата звернення: 22.02.2025).

10. Dale Ensminger, Leonard J. Bond. *Ultrasonics: Fundamentals, Technologies, and Applications*. CRC Press, 2024. URL: https://books.google.com.ua/books?hl=uk&lr=&id=EVnqEAAAQBAJ&oi=fnd&pg=PP1&ots=TIJWg2ZnZT&sig=pvB2Sg-U_YK35kI7IIVK-cXkBMI&redir_esc=y#v=onepage&q&f=false (дата звернення: 25.11.2024 р.)

11. IoT projects Using Arduino with Codes and Schematics. *Arduino Maker Space*. URL: <https://arduinomakerspace.com/iot-projects-using-arduino/> (дата звернення: 22.02.2025).

12. M. Rajeswari, J. Lingappa, M. G. Basha, V. Deepa, V. Sravani. K. Srilatha. To Design and Implement a Radar System using Arduino and Ultrasonic Sensor. *2nd International Conference on Automation, Computing and Renewable Systems (ICACRS)*. Pudukkottai, India, 2023. P. 208-212.

13. S. M. Basir, R. C. Ismail, K. H. Yusof, N. I. A. Katim, M. N. M. Isa, S.Z. M. Naziri. An implementation of Short Time Fourier Transform for Harmonic Signal Detection. *Journal of Physics: Conference Series*. Vol. 1755. 2021. URL: <https://iopscience.iop.org/article/10.1088/1742-6596/1755/1/012013/pdf> (дата звернення: 28.02.2025).

14. Ya. Garud, A. Fatke, S. Chavan, A. Ghuge. Design and Implement a Radar System using Arduino and Ultrasonic Sensor. *International Journal on Science and Technology*. Vol. 16. Issue 2. 2025. URL: <https://www.ijst.org/papers/2025/2/3328.pdf> (дата звернення: 28.02.2025).

15. Ya. Bendal, A. Pawar, K. Raut, P. Thorat, Sh. Bendre. Arduino based radar system. *International Research Journal of Modernization in Engineering Technology and Science*. Vol. 07. Issue 03. 2025. URL: <https://www.irjmets.com/>

uploadedfiles/paper//issue_3_march_2025/69961/ final/ fin_irjmets1742719471.pdf?utm_source=chatgpt.com (дата звернення: 11.03.2025).

16. Sh. Chowdhury, S. Mazumdar, S. Giri, A. Bhattacharya. Radar System Utilizing Arduino. *International Journal of Research Publication and Reviews*. Vol. 5. № 5. 2024. P. 12732-12738. URL: https://www.researchgate.net/publication/381119575_Radar_System_Utilizing_Arduino (дата звернення: 19.03.2025).

17. Jeong H., Kim S. Educational Low-Cost C-Band FMCW Radar System Comprising Commercial Off-the-Shelf Components for Indoor Through-Wall Object Detection. *Electronics*. Vol. 10(22). 2021. URL: <https://www.mdpi.com/2079-9292/10/22/2758> (дата звернення: 21.03.2025).

18. Arduino UNO Rev3 – Technical Documentation. *Arduino Docs* (2025). URL: <https://docs.arduino.cc/hardware/uno-rev3/> (дата звернення: 21.03.2025).

19. Yash Goyal. Comparative Study of Microcontoller: Arduino Uno, Raspberry Pi 4, ESP 32. *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*. Vol. 12. Issue VII. 2024. URL: <https://www.ijraset.com/best-journal/comparative-study-of-microcontroller-esp32-arduino-uno-raspberry-pi-4#:~:text=A,2> (дата звернення: 27.03.2025).

20. Arduino Uno vs ESP32. *Tutorialspoint*. URL: <https://www.tutorialspoint.com/arduino-uno-vs-esp32> (дата звернення: 29.03.2025).

21. Arduino Uno vs STM32duino Blue Pill. *Tutorialspoint*. URL: <https://www.tutorialspoint.com/arduino-uno-vs-stm32duino-blue-pill> (дата звернення: 29.03.2025).

22. Arduino Uno vs Arduino Nano. *Tutorialspoint*. URL: <https://www.tutorialspoint.com/arduino-uno-vs-arduino-nano> (дата звернення: 29.03.2025).

23. Arduino Uno vs Arduino Mega. *Tutorialspoint*. URL: <https://www.tutorialspoint.com/arduino-uno-vs-arduino-mega> (дата звернення: 10.04.2025).

24. Raspberry Pi Pico. *Raspberry Pi Trading Ltd*. URL: <https://datasheets.raspberrypi.com/> (дата звернення: 15.04.2025).

25. Ultrasonic Ranging Module HC – SR04. *SparkFun*. URL: https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf?utm_source (дата звернення: 26.04.2025).

26. Comparing LV-MaxSonar-EZ and HC-SR04 Sonar Range Finders With Arduino. *Autodesk*. URL: https://www.instructables.com/Comparing-LV-MaxSonar-EZ-and-HC-SR04-Sonar-Range-F/?utm_source=chatgpt.com (дата звернення: 28.04.2025).

27. SG90 9 g Micro Servo. *FriendlyWire*. URL: <https://www.friendlywire.com/projects/ne555-servo-safe/SG90-datasheet.pdf> (дата звернення: 01.05.2025).

28. Complete Guide for Ultrasonic Sensor HC-SR04 with Arduino. *RandomNerdTutorials*. URL: <https://randomnerdtutorials.com/complete-guide-for-ultrasonic-sensor-hc-sr04/> (дата звернення: 04.05.2025).

29. HC-SR04 Module. *AmperMarket*. URL: <https://ampermarket.kz/base/ex17-ultrasonic/> (дата звернення: 07.05.2025).

30. Arduino Radar Project. *HowToMechatronics*. URL: <https://howtomechatronics.com/projects/arduino-radar-project/> (дата звернення: 09.05.2025).

ДОДАТКИ

Додаток А

Код програмування для Processing IDE

```

import processing.serial.*; // imports library for serial communication
import java.awt.event.KeyEvent; // imports library for reading the data from the serial
port
import java.io.IOException;
Serial myPort; // defines Object Serial
// defubes variables
String angle="";
String distance="";
String data="";
String noObject;
float pixsDistance;
int iAngle, iDistance;
int index1=0;
int index2=0;
PFont orcFont;
void setup() {

    size (1200, 700); // ***CHANGE THIS TO YOUR SCREEN RESOLUTION***
    smooth();
    myPort = new Serial(this, "COM7", 9600); // starts the serial communication
    myPort.bufferUntil('.'); // reads the data from the serial port up to the character '.'.
    So actually it reads this: angle,distance.
}
void draw() {

    fill(98, 245, 31);
    // simulating motion blur and slow fade of the moving line
    noStroke();
    fill(0, 4);
    rect(0, 0, width, height-height*0.065);

    fill(98, 245, 31); // green color
    // calls the functions for drawing the radar
    drawRadar();
    drawLine();
    drawObject();
    drawText();
}
void serialEvent (Serial myPort) { // starts reading data from the Serial Port
    // reads the data from the Serial Port up to the character '.' and puts it into the
    String variable "data".
    data = myPort.readStringUntil('.');
    data = data.substring(0, data.length()-1);

    index1 = data.indexOf(","); // find the character ',' and puts it into the variable
    "index1"
    angle= data.substring(0, index1); // read the data from position "0" to position of the
    variable index1 or thats the value of the angle the Arduino Board sent into the Serial
    Port

```

```

    distance= data.substring(index1+1, data.length()); // read the data from position
"index1" to the end of the data pr thats the value of the distance

    // converts the String variables into Integer
    iAngle = int(angle);
    iDistance = int(distance);
}
void drawRadar() {
    pushMatrix();
    translate(width/2, height-height*0.074); // moves the starting coordinats to new
location
    noFill();
    strokeWeight(2);
    stroke(98, 245, 31);
    // draws the arc lines
    arc(0, 0, (width-width*0.0625), (width-width*0.0625), PI, TWO_PI);
    arc(0, 0, (width-width*0.27), (width-width*0.27), PI, TWO_PI);
    arc(0, 0, (width-width*0.479), (width-width*0.479), PI, TWO_PI);
    arc(0, 0, (width-width*0.687), (width-width*0.687), PI, TWO_PI);
    // draws the angle lines
    line(-width/2, 0, width/2, 0);
    line(0, 0, (-width/2)*cos(radians(30)), (-width/2)*sin(radians(30)));
    line(0, 0, (-width/2)*cos(radians(60)), (-width/2)*sin(radians(60)));
    line(0, 0, (-width/2)*cos(radians(90)), (-width/2)*sin(radians(90)));
    line(0, 0, (-width/2)*cos(radians(120)), (-width/2)*sin(radians(120)));
    line(0, 0, (-width/2)*cos(radians(150)), (-width/2)*sin(radians(150)));
    line((-width/2)*cos(radians(30)), 0, width/2, 0);
    popMatrix();
}
void drawObject() {
    pushMatrix();
    translate(width/2, height-height*0.074); // moves the starting coordinats to new
location
    strokeWeight(9);
    stroke(255, 10, 10); // red color
    pixsDistance = iDistance*((height-height*0.1666)*0.025); // covers the distance from the
sensor from cm to pixels
    // limiting the range to 40 cms
    if (iDistance<40) {
        // draws the object according to the angle and the distance
        line(pixsDistance*cos(radians(iAngle)), -pixsDistance*sin(radians(iAngle)), (width-
width*0.505)*cos(radians(iAngle)), -(width-width*0.505)*sin(radians(iAngle)));
    }
    popMatrix();
}
void drawLine() {
    pushMatrix();
    strokeWeight(9);
    stroke(30, 250, 60);
    translate(width/2, height-height*0.074); // moves the starting coordinats to new
location
    line(0, 0, (height-height*0.12)*cos(radians(iAngle)), -(height-
height*0.12)*sin(radians(iAngle))); // draws the line according to the angle
    popMatrix();
}

```

```

void drawText() { // draws the texts on the screen

  pushMatrix();
  if (iDistance>40) {
    noObject = "Out of Range";
  } else {
    noObject = "In Range";
  }
  fill(0, 0, 0);
  noStroke();
  rect(0, height-height*0.0648, width, height);
  fill(98, 245, 31);
  textSize(25);

  text("10cm", width-width*0.3854, height-height*0.0833);
  text("20cm", width-width*0.281, height-height*0.0833);
  text("30cm", width-width*0.177, height-height*0.0833);
  text("40cm", width-width*0.0729, height-height*0.0833);
  textSize(40);
  text("N_Tech ", width-width*0.875, height-height*0.0277);
  text("Angle: " + iAngle + " ", width-width*0.48, height-height*0.0277);
  text("Distance: ", width-width*0.26, height-height*0.0277);
  if (iDistance<40) {
    text("      " + iDistance + " cm", width-width*0.225, height-height*0.0277);
  }
  textSize(25);
  fill(98, 245, 60);
  translate((width-width*0.4994)+width/2*cos(radians(30)), (height-height*0.0907)-
width/2*sin(radians(30)));
  rotate(-radians(-60));
  text("30", 0, 0);
  resetMatrix();
  translate((width-width*0.503)+width/2*cos(radians(60)), (height-height*0.0888)-
width/2*sin(radians(60)));
  rotate(-radians(-30));
  text("60", 0, 0);
  resetMatrix();
  translate((width-width*0.507)+width/2*cos(radians(90)), (height-height*0.0833)-
width/2*sin(radians(90)));
  rotate(radians(0));
  text("90", 0, 0);
  resetMatrix();
  translate(width-width*0.513+width/2*cos(radians(120)), (height-height*0.07129)-
width/2*sin(radians(120)));
  rotate(radians(-30));
  text("120", 0, 0);
  resetMatrix();
  translate((width-width*0.5104)+width/2*cos(radians(150)), (height-height*0.0574)-
width/2*sin(radians(150)));
  rotate(radians(-60));
  text("150", 0, 0);
  popMatrix();
}

```