

**Міністерство освіти і науки України**

**Луцький національний технічний університет**

(повне найменування закладу вищої освіти)

**Факультет комп'ютерних та інформаційних технологій**

(повне найменування факультету)

**Кафедра комп'ютерної інженерії та безпеки**

(повне найменування кафедри)

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

**СИСТЕМА АВТОМАТИЗОВАНОГО ВИМІРЮВАННЯ  
ВІДСТАНІ НА БАЗІ ARDUINO ТА УЛЬТРАЗВУКОВОГО  
ДАТЧИКА US-025**

**AUTOMATED DISTANCE MEASUREMENT SYSTEM BASED ON  
ARDUINO AND US-025 ULTRASONIC SENSOR**

спеціальність 123 Комп'ютерна інженерія  
(шифр і назва спеціальності)

освітня програма Комп'ютерна інженерія  
(назва освітньої програми)

Виконав: здобувач вищої освіти  
групи КІ-41  
Іщук Олександр Русланович

(підпис)

Керівник:  
к.т.н., доцент  
Христинець Наталія Анатоліївна

(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. Т. Терлецький

« 10 » 01 2025 р.

ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

*Іщуку Олександр Руслановичу*

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи *Система автоматизованого вимірювання відстані на базі*

*Arduino та ультразвукового датчика US-025*

Керівник роботи *к.т.н., доцент Христинець Наталія Анатоліївна*

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 *науково-технічна література та публікації в періодичних Виданнях по темі Інтернету речей, опубліковані роботи в даній області та різні інтернет-ресурси технічного спрямування*

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

*Аналітичний огляд джерел за обраною тематикою*

*Проектування системи вимірювання відстані*

*Реалізація та перевірка працездатності системи*

5. Перелік графічного (ілюстративного) матеріалу:

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
<i>Аналітичний огляд питань розробки системи автоматизованого вимірювання</i>	<i>Христинець Н.А, доцент</i>		
<i>Методи розробки та основні інструменти</i>	<i>Христинець Н.А, доцент</i>		
<i>Реалізація системи вимірювання відстані</i>	<i>Христинець Н.А, 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 з питань розробки</i>	до 10.02.2025 р.	Виконано
2.	<i>Вибір апаратної та програмної бази для проєкту</i>	до 02.03.2025 р.	Виконано
3.	<i>Проєктування та тестування системи моніторингу</i>	до 02.04.2025 р.	Виконано
4.	<i>Завдання на дослідження</i>	до 10.04.2025 р.	Виконано
5.	<i>Формування списку використаних джерел</i>	до 15.04.2025 р.	Виконано
6.	<i>Формування додатків та ілюстративного матеріалу</i>	до 10.05.2025 р.	Виконано
7.	<i>Представлення остаточного варіанту кваліфікаційної роботи керівнику</i>	до 15.05.2025 р.	Виконано
8.	<i>Нормоконтроль</i>	до 30.05.2025 р.	Виконано
9.	<i>Інструментальна перевірка на академічний плагіат</i>	до 03.06.2025 р.	Виконано
10.	<i>Представлення кваліфікаційної та всіх супровідних документів на кафедрі</i>	до 10.06.2025 р.	Виконано

Здобувач вищої освіти

\_\_\_\_\_  
(підпис)

Іщук О.Р.

\_\_\_\_\_  
(прізвище, ініціали)

Керівник кваліфікаційної роботи

\_\_\_\_\_  
(підпис)

Христинець Н.А.

\_\_\_\_\_  
(прізвище, ініціали)

## АНОТАЦІЯ

Іщук О. Р. Система автоматизованого вимірювання відстані на базі Arduino та ультразвукового датчика US-025. 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 дослідженню джерел з питань розробки систем автоматизованого вимірювання. Здійснено класифікацію датчиків та проведено аналіз апаратних платформ для побудови систем вимірювання.

У другому розділі сформовано технічні вимоги до системи, обґрунтовано вибір модулів системи – Arduino Uno R3, US-025, RGB-світлодіоди KY-016, LCD 1602 дисплей 16x2. Розглянуто особливості підключення системи та методи її перевірки та тестування роботи.

Третій розділ присвячено реалізації системи вимірювання відстані. Описано монтаж апаратної частини та програмування роботи в середовищі Arduino. Спроектовано та реалізовано візуалізацію результатів вимірювання через дисплей та проведено тестування роботи системи.

Ключові слова: вимірювання відстані, Arduino Uno R3, US-025, датчики, мікроконтролерна система, світлодіоди.

## ANNOTATION

Ishchuk O. Automated distance measurement system based on Arduino and US-025 ultrasonic sensor. Manuscript.

Bachelor's qualification work OP «Computer Engineering» specialty 123 Computer Engineering. Lutsk National Technical University. Lutsk, 2025.

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

The first section is devoted to the study of sources on the development of automated measurement systems. The classification of sensors is carried out and the analysis of hardware platforms for building measurement systems is carried out.

In the second section, technical requirements for the system are formed, the choice of system modules is justified – Arduino Uno R3, US-025, RGB LEDs KY-016, LCD 1602 display 16x2. The features of the system connection and methods of its verification and testing of operation are considered.

The third section is devoted to the implementation of the distance measurement system. The hardware assembly and programming of the work in the Arduino environment are described. The visualization of the measurement results through the display is designed and implemented, and the system operation is tested.

Keywords: distance measurement, Arduino Uno R3, US-025, sensors, microcontroller system, LEDs.

## ЗМІСТ

ВСТУП .....	7
РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД ПИТАНЬ РОЗРОБКИ СИСТЕМ АВТОМАТИЗОВАНОГО ВИМІРЮВАННЯ.....	9
1.1 Принципи роботи систем автоматизованого вимірювання відстані .....	9
1.2 Класифікація датчиків відстані .....	12
1.3 Порівняльна характеристика ультразвукових датчиків.....	14
1.4 Аналіз апаратних платформ для побудови систем вимірювання .....	19
1.5 Огляд типових застосувань.....	22
РОЗДІЛ 2. МЕТОДИКА РОЗРОБКИ ТА ОСНОВНІ ІНСТРУМЕНТИ.....	25
2.1 Технічні вимоги до системи.....	25
2.2 Обґрунтування вибору апаратних та програмних засобів.....	26
2.3 Схема з'єднання елементів системи на платформі Arduino .....	30
2.4 Опис алгоритму роботи системи.....	33
2.5 Підготовка програмного середовища та використання бібліотек .....	36
2.6 Методи перевірки та тестування роботи системи .....	40
РОЗДІЛ 3. РЕАЛІЗАЦІЯ СИСТЕМИ ВИМІРЮВАННЯ ВІДСТАНІ.....	41
3.1 Монтаж апаратної частини системи .....	41
3.2 Програмування роботи системи в середовищі Arduino .....	46
3.3 Візуалізація та виведення результатів вимірювання.....	51
3.4 Тестування роботи системи .....	54
3.5 Аналіз отриманих результатів .....	59
ВИСНОВКИ.....	61
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	63
ДОДАТКИ.....	64

## ВСТУП

Актуальність теми. У реаліях сьогодення стрімкого розвитку та популярності набувають різнотипів системи автоматизації або напів-автоматизації процесів. З огляду на широку сферу професійної діяльності, де повсякденним завданням є робота з відстанню та розмірами об'єктів, розробка системи автоматизованого безконтактного вимірювання відстані є вкрай актуальною та доцільною.

Такі системи використовуються у будівельних, логістичних, автомобільних, робототехнічних та охоронних сферах для вирішення завдань вимірювання відстаней, навігації, автоматизації процесів і підвищення точності виконання робіт.

Серед доступних та простих у використанні рішень значною популярністю користуються мікроконтролери Arduino, які забезпечують можливість оперативного створення прототипів і готових пристроїв. Ультразвуковий датчик US-025 у поєднанні з Arduino дає змогу реалізувати ефективну й водночас нескладну систему вимірювання відстані, яку за потреби можна інтегрувати в більш комплексні автоматизовані проекти.

Такий підхід є особливо актуальним як для освітніх цілей, так і для практичного застосування в інженерії.

Метою роботи є розробка системи автоматизованого вимірювання відстані на базі мікроконтролера Arduino та ультразвукового датчика US-025, яка забезпечує точне, стабільне та надійне визначення відстані до об'єктів у режимі реального часу.

Об'єкт дослідження – процес безконтактного вимірювання відстані до об'єктів з використанням ультразвукових сенсорів у складі автоматизованих систем на базі мікроконтролера Arduino.

Предмет дослідження – структура, принцип роботи та ефективність взаємодії ультразвукового датчика US-025 з мікроконтролером Arduino для реалізації системи вимірювання відстані.

Для досягнення мети поставлено наступні завдання:

- аналітично дослідити питання систем автоматизованих вимірювань та визначити способи реалізації рішень з використанням бюджетних технічних засобів на базі технологій Інтернету речей;
- реалізувати систему автоматизованого вимірювання відстані на базі мікроконтролера Arduino та ультразвукового датчика US-025;
- розробити алгоритм обробки даних з ультразвукового сенсора для точного визначення відстані в режимі реального часу;
- дослідити точність, стабільність та надійність вимірювань ультразвукового датчика US-025 у різних умовах експлуатації;
- візуалізувати результати вимірювань за допомогою дисплея або інших доступних засобів відображення інформації;
- спроектувати прототип пристрою для інтеграції в охоронні системи.

Апробація. Практичне застосування розробки проходило на базі переддипломної практики, на приватному підприємстві ЮАІНЕТ-КІБЕРНЕТ, що має невелике складське приміщення, де зберігаються матеріали та обладнання. Результати роботи представлені на VI Міжнародній науково-практичній конференції молодих вчених, аспірантів і студентів «Сучасні інформаційні технології та системи в управлінні», яка проходила з 10 по 11 квітня 2025 року [1] та представлені у додатку А кваліфікаційної роботи.

# РОЗДІЛ 1

## АНАЛІТИЧНИЙ ОГЛЯД ПИТАНЬ РОЗРОБКИ СИСТЕМ АВТОМАТИЗОВАНОГО ВИМІРЮВАННЯ

### 1.1 Принципи роботи систем автоматизованого вимірювання відстані

Основою роботи систем автоматизованого вимірювання відстані є засоби, які здійснюють фізичне вимірювання відстані, та програма, яка налагоджує роботу засобів в одній системі, відповідає за логіку роботи пристрою, аналіз, обробку та візуалізацію результатів.

У таких системах засобами називають компоненти, які відповідають за живлення пристрою, зчитування, обробку та вивід інформації користувачеві. Принципи роботи окремих компонентів можуть відрізнятися, однак їх призначення є типовим.

Головним компонентом систем автоматизованого вимірювання є контролер. Він відповідає за об'єднання компонентів в одну систему, за отримання і передачу сигналів даних, за створення керуючих сигналів, за обробку та коригування отриманих даних тощо. Його функціональні можливості не обмежуються певним набором команд і залежать лише від програми, яка в нього закладена. В окремих випадках контролер може відповідати за передачу сигналів живлення, а також за стабілізацію напруги та її перетворення зі змінної у постійну.

Кожен контролер повинен містити базовий перелік елементів, які є обов'язковими для роботи систем, а саме: процесор – відповідає за обробку даних, керує усіма пристроями системи, виконує алгоритм вимірювання; інтерфейси введення/виведення – використовуються для передачі даних та сигналів, слугують «містками» між головним контролером та іншими компонентами; пам'ять (оперативна та постійна) – відповідає за збереження тимчасових та постійних даних програми, а також може зберігати результати вимірювання; система живлення – відповідає за подачу живлення на контролер, стабілізацію та його розподіл між частинами та компонентами системи. У

випадку, якщо система використовує датчики, які видають аналоговий сигнал, то обов'язковою стає присутність у системі аналогово-цифрового перетворювача.

Наступним обов'язковим компонентом кожної автоматизованої системи вимірювання відстані є система живлення. Її основним призначенням є забезпечення електричною енергією всіх компонентів системи. Вона відповідає за перетворення типу струму, стабілізацію напруги або, при потребі, за зміну рівня живлення.

Основним елементом системи живлення є джерело живлення. Джерелами живлення можуть бути адаптери змінного струму, батареї, акумулятори, сонячні панелі тощо. У більш складних системах можуть використовуватися стабілізатори напруги, безперебійники та інші пристрої для гарантії безперебійної подачі енергії.

Живлення спочатку подається до головного контролера через спеціальні канали, а потім передається до всіх компонентів системи. Якщо система живлення дозволяє, то живлення компонентів може відбуватися безпосередньо від джерела після стабілізації.

Ключовим компонентом у системах автоматизованого вимірювання є датчик вимірювання відстані. За принципом дії вони бувають: ультразвукові, інфрачервоні, лазерні, оптичні та радіоімпульсні (радарні). Основним завданням датчика є аналіз відстані до найближчого об'єкта, на який він направлений.

Візуалізація результатів у системах автоматизованого вимірювання відстані є важливим чинником для забезпечення користувацької зручності, оскільки через неї користувач отримує доступ до даних. Візуалізація може здійснюватися різними способами, починаючи від простих LED-екранів до потужних телефонних додатків чи вебінтерфейсів.

У роботі систем автоматизованого вимірювання відстані можуть використовуватися інші, не обов'язкові компоненти. Вони можуть додати пристрою практичності, зручності у використанні або ж просто зробити

пристрій цікавішим. Наприклад, кнопки (для ввімкнення/вимкнення пристрою, для початку виміру, для навігації по інтерфейсу тощо), модулі бездротової передачі даних (Wi-Fi, Bluetooth), модуль SD-карт (для тривалого зберігання результатів вимірювань), діоди світлової індикації (для сигналізування різних станів пристрою), додаткові датчики вимірювання відстані (для покращення результатів вимірювання) тощо.

Всі вище описані компоненти формують принципи фізичної роботи будь-яких систем автоматизованого вимірювання відстані. Базова взаємодія компонентів у таких системах відображена на рисунку 1.1.

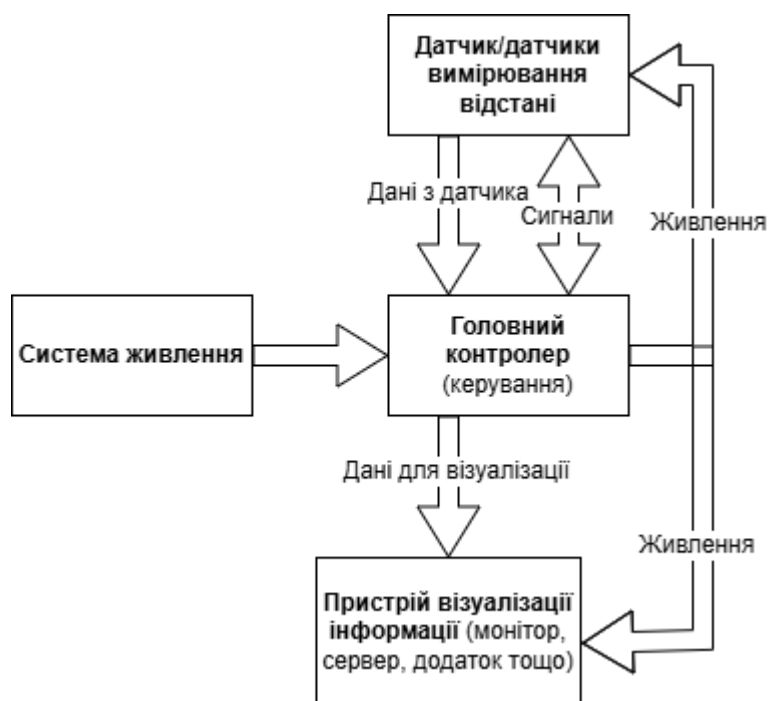


Рисунок 1.1 – Принцип роботи систем автоматизованого вимірювання відстані

Програмування – це друга важлива частина в реалізації роботи систем автоматизованого вимірювання відстані. Воно передбачає створення логіки роботи компонентів в одній злагодженій системі. Програма містить інструкції взаємодії компонентів (отримання та надсилання даних від одного компонента до іншого, передача сигналів тощо). На цьому етапі програмісту необхідно створити надійну, структуровану та ефективну програму, яка зможе виміряти відстань, обробити дані та вивести їх користувачу.

Отже, принципи роботи систем автоматизованого вимірювання відстані базуються на створенні взаємодії між компонентами та програмованим алгоритмом роботи цих компонентів. Основу такої системи складають датчики, контролер, засоби візуалізації та інші додаткові компоненти, які можуть покращити функціональність системи. Узгоджена взаємодія всіх компонентів забезпечує надійність та точність роботи такої системи.

## **1.2 Класифікація датчиків відстані**

При розробці системи автоматизованого вимірювання відстані необхідно вибрати датчик відстані, який допоможе найкраще виконати поставлене завдання. Кожен датчик має свої переваги та недоліки, тому опиратися потрібно на їх конкретні характеристики та сферу застосування, де буде використовуватися система вимірювання відстані.

Датчики вимірювання відстані працюють за допомогою надсилання сигналів та отримання зворотної відповіді і по суті є перетворювачами однієї енергії в іншу. Вони не мають фізичного контакту із об'єктом, на відміну від традиційних вимірювачів відстані. Зміни у більшості датчиків фіксуються у вигляді часу, необхідного для повернення надісланого сигналу, інтенсивності поверненого сигналу або зміни фази сигналу після відбиття.

Датчики вимірювання відстані використовують різні види вимірювальних технологій. Сигналами для їх роботи може слугувати світло, звук або радіо хвилі.

Усі датчики можна класифікувати за кількома характеристиками, а саме: за принципом дії, за типом вихідного сигналу, за діапазоном вимірювання, за умовами експлуатації та за типом монтажу.

За принципом дії можна виділити п'ять видів датчиків відстані:

Ультразвукові датчики – основою роботи таких датчиків є надсилання ультразвукової хвилі до напрямленого об'єкта та отримання зворотного ультразвукового сигналу (хвилі). Такий тип датчиків добре підходить для

середніх відстаней. Недоліком даних датчиків є втрата точності при вимірюванні відстані до м'яких або звукопоглинаючих поверхонь.

Інфрачервоні датчики – використовують інфрачервоне випромінювання. Світлодіод у таких датчиках посилає лазерний промінь, який відбивається і зчитується приймачем датчика. Цей тип при належних умовах має високу точність вимірювання, однак має обмежений діапазон і краще підходить для виміру малих дистанцій або розмірів. Недоліком інфрачервоних датчиків є втрата точності при сторонньому освітленні, а також при вимірі відстані до прозорих, світлопоглинаючих та світлозаломлюючих поверхонь.

Лазерні датчики – мають декілька технологій вимірювання, з яких можна виділити дві основні: розрахунок відстані через вимірювання часу польоту променю від початку надсилання до його відбиття від поверхні та вимірювання відстані шляхом аналізу відбитого лазерного променю. Такі датчики видають точніші та стабільніші результати і працюють з найбільшою кількістю різних поверхонь цілі вимірювання.

Оптичні датчики (датчики на базі камер) – можуть вимірювати не лише відстань, але й розпізнавати форму, колір чи інші візуальні властивості об'єктів; корисні при створенні складніших систем, які потребують вимірювання або аналізу об'єктів. Для коректної роботи таких датчиків необхідно забезпечити високий рівень освітлення.

Радіохвильові (радарні) датчики – дозволяють вимірювати відстані за допомогою радіохвиль. Вони ефективні в умовах пилу, пару, туману, задимлення тощо.

Датчики відстані також можна класифікувати за типом вихідного сигналу. Вихідний сигнал це, спосіб передачі даних чи сигналів. За типом вихідного сигналу розрізняють наступні види датчиків: аналогові (генерують безперервний сигнал, значення якого дорівнює вимірюваній відстані), цифрові (видають чіткі цифрові сигнали при досягненні певної відстані) та комунікаційні (забезпечують серійну передачу даних та сигналів).

За діапазоном вимірювання датчики бувають короткої (до 10-ти сантиметрів), середньої (від 10-ти сантиметрів до 5-ти метрів) та довгої (від 5-ти метрів) дії. Датчики короткої дії зазвичай мають більшу точність вимірювання і підходять для невеликих геометричних обчислень. Стандартними і найчастіше використовуваними вважаються датчики середньої дії. Вони задовольняють більшість користувацьких потреб. Датчики дальньої дії застосовуються при потребі виміру великих дистанцій чи розмірів.

За умовами експлуатації датчики вимірювання відстані бувають внутрішні та зовнішні. Внутрішні датчики призначені для виміру в умовах стабільного середовища, тобто всередині приміщень. Зовнішні датчики, натомість, призначені для вимірювання в спеціальних умовах. Вони мають захист від пилу, вологи, перепадів температур тощо.

За типом монтажу датчики вимірювання відстані бувають: поверхневі (легко встановлюються на пласкі поверхні), вбудовані (інтегруються в корпус вимірювального пристрою) та промислові (мають різьбу або спеціальне кріплення для встановлення у спеціальні місця).

Отже, датчики вимірювання відстані відрізняються за характеристиками, методами роботи та умовами застосування. Вибір потрібного датчика залежить від поставленого завдання, умов експлуатації та вимог до точності. Важливо врахувати всі переваги й обмеження кожного типу, щоб забезпечити ефективну роботу системи.

### **1.3 Порівняльна характеристика ультразвукових датчиків**

Ультразвукові датчики широко використовуються у системах вимірювання відстані завдяки своїй універсальності, достатній точності та доступності. Вони є оптимальним рішенням для більшості застосувань у побутових, промислових, освітніх та інженерних проектах.

В межах цієї роботи було розглянуто найпопулярніші варіанти ультразвукових датчиків:

US-025 – популярний ультразвуковий датчик з UART (універсальний асинхронний приймач-передавач), який має два режими роботи та підходить для розробок без складної роботи сигналу. Virізняється стабільністю, точністю вимірювання та відсутністю «сліпих зон». Має чотири піни підключення: VCC, Trig, Echo та GND. Зовнішній вигляд цього датчику відображено на рисунку 1.2.



Рисунок 1.2 – Ультразвуковий датчик US-025 [2]

HC-SR04 (рисунок 1.3) – один із найпоширеніших і найдешевших датчиків. Має цифровий інтерфейс, високий рівень сумісності з контролерами Arduino та ESP.



Рисунок 1.3 – Ультразвуковий датчик HC-SR04 [3]

HY-SRF05 (рис. 1.4) – розширена версія HC-SR04 з можливістю роботи в двох режимах. Цей датчик має більшу точність і дальність дії. Має п'ять пінів підключення: VCC, Trig, Echo, OUT та GND.



Рисунок 1.4 – Ультразвуковий датчик HY-SRF05 [4]

JSN-SR04T – вологозахищений датчик вимірювання відстані. Складається з двох частин (рисунок 1.5): сам датчик в герметичному корпусі з кабелем та плата для підключення до контролера і обробки інформації.



Рисунок 1.5 – Ультразвуковий датчик JSN-SR04T [5]

PING))) – ультразвуковий датчик, який ідеально підходить для вимірювання відстані між рухомими або не рухомими об'єктами. Вирізняється простотою підключення та адаптивністю під різні потреби. Точність вимірів та стабільність роботи цього датчика значно вища ніж у аналогів, тому ціна відповідна.

На вигляд PING))) (рисунок 1.6) не відрізняється від інших, однак має дещо інший принцип роботи та три піни: GND, 5V та SIG.



Рисунок 1.6 – Ультразвуковий датчик PING))) [6]

MaxBotix MB1010 (рисунок 1.7) – високоточний ультразвуковий датчик з кількома виходами: аналоговим, серійним та імпульсним. Підходить для розробок більш професійного рівню.



Рисунок 1.7 – Ультразвуковий датчик MaxBotix MB1010 [7]

SRF02 – це компактний ультразвуковий датчик відстані, який підтримує два інтерфейси передачі даних: UART і I2C. Ця функція дозволяє легко інтегрувати системи з декількома датчиками, зокрема через шину I2C, де кожному датчику можна призначити власну адресу.

SRF02 має циліндричну форму, яка зручна для вбудованого монтажу, та один ультразвуковий перетворювач, що використовується і для передавання, і для прийому сигналів (рисунок 1.8).

Модел ь	Діапазон вимірюв ання	Точ- ність	Кут огля ду	Інтерфейс	Напруг а жив- лення	Спожи- ваний струм	Ціна (грн )	Примітки
------------	-----------------------------	---------------	-------------------	-----------	---------------------------	--------------------------	-------------------	----------



Рисунок 1.8 – Ультразвуковий датчик SRF02 [8]

Для кращого розуміння особливостей роботи кожного з описаних датчиків, було створено порівняльну таблицю (таблиця 1.1) з описом основних експлуатаційних та функціональних характеристик.

Таблиця 1.1 – Порівняльна характеристика ультразвукових датчиків

PING)) )	2–300 см	±0,3 см	15°	Цифрови й	5 В	30 мА	~500	Висока точність, простота підключення
MB101 0	20–765 см	±1 см	30°	Аналогови й UART, PWM	2,5–5,5 В	2 мА	~800	Високоточний, з кількома виходами
SRF02	15–600 см	±1 см	55°	UART, I2C	5 В	4 мА	~600	Компактний, підтримує кілька інтерфейсів
US-025	2–600 см	±0,3 см	15°	UART	3,8–5,5 В	до 6 мА	39,9 9– 117	Стабільний, точний, без «сліпих зон»
HC- SR04	2–400 см	±0,3 см	15°	Цифрови й	5 В	до 15 мА	29,8 0–45	Популярний, недорогий, сумісний з Arduino
HY- SRF05	2–450 см	±0,2 см	<15°	Цифрови й	5 В	10–40 мА	83	Покращена версія HC- SR04 з двома режимами роботи
JSN- SR04T	25–500 см	±0,5 см	<50°	Цифрови й	5 В	30 мА	185– 528	Вологозахищен ий, підходить для зовнішнього використання

У ході аналізу було виявлено сильні і слабкі сторони кожного датчику. Вибір конкретного варіанту для використання залежить від сфери застосування та умов експлуатації.

#### 1.4 Аналіз апаратних платформ для побудови систем вимірювання

При розробці системи автоматизованого вимірювання відстані можливе використання різних апаратних платформ, починаючи від популярних мікроконтролерів і закінчуючи власно розробленими платами. Вибір апаратної платформи безпосередньо впливає на функціональні можливості системи, її практичність, точність, енергоспоживання, швидкодію та вартість реалізації. Було оглянуто мікроконтролери, які користуються великим попитом і популярністю серед розробників. Серед них: Arduino Uno, Arduino Nano, Arduino Mega, ESP32, ESP8266, STM32 та Raspberry Pi Pico [9]. Було оглянуто

наступні аспекти кожного мікроконтролера: кількість і тип входів/виходів, частота процесора, обсяг пам'яті, ціна та інші технічні особливості.

Для того щоб вибрати оптимальну апаратну основу для створення системи автоматизованого вимірювання відстані було проаналізовано технічні характеристики перерахованих вище платформ та відображено їх у таблиці 1.2.

Таблиця 1.2 – Порівняльна характеристика апаратних платформ для побудови системи автоматизованого вимірювання відстані

Платформа	Мікроконтролер	Частота, МГц	Flash / SRAM	GPIO / PWM / ADC	UART / I2C / SPI	Ціна (грн)	Особливості
Arduino Uno	ATmega328P	16	32 KB / 2 KB	14 / 6 / 6	1 / 1 / 1	200-500	Стандартна плата для початківців; проста в освоєнні.
Arduino Nano	ATmega328P	16	32 KB / 2 KB	22 / 6 / 8	1 / 1 / 1	~400	Компактний розмір; ідеально підходить для вбудованих систем.
Arduino Mega	ATmega2560	16	256 KB / 8 KB	54 / 15 / 16	4 / 1 / 1	~800	Велика кількість пінів; підходить для складних проектів.
ESP32	Xtensa LX6 (2 ядра)	160–240	4 MB Flash / 520 KB SRAM	34 / 16 / 18	3 / 2 / 4	~250	Висока продуктивність; вбудовані Wi-Fi та Bluetooth.
ESP8266	Tensilica L106	80–160	до 16 MB Flash / 80 KB SRAM	17 / 4 / 1	1 / 1 / 1	~150	Недорогий варіант з Wi-Fi; обмежена кількість пінів.
STM32 (F103C8T6)	ARM Cortex-M3	72	64 KB / 20 KB	37 / 15 / 10	3 / 2 / 2	~200	Висока енергоефективність; потребує досвіду для програмування.
Raspberry Pi Pico	RP2040 (2× Cortex-M0+)	до 133	2 MB Flash / 264 KB SRAM	26 / 16 / 3	2 / 2 / 2	~150	Підтримка MicroPython; гнучкі можливості вводу/виводу.

Ці параметри є ключовими у розробці системи вимірювання, оскільки вони визначають можливість підключення ультразвукових датчиків та інших компонентів системи, зберігання та передачу результатів вимірювання, а також ефективність і стабільність роботи з платформою.

Аналізуючи наведені характеристики апаратних платформ можна зробити висновки про те, які функції мікроконтролерів потрібні для реалізації системи автоматизованого вимірювання відстані з використанням ультразвукового датчика US-025.

Для передачі даних між компонентами системи у мікроконтролері обов'язково повинен бути інтерфейс зв'язку. Кожен з вище описаних мікроконтролерів має хоча б один з наступних інтерфейсів зв'язку: UART, I2C або SPI. Деякі платформи мають більше ніж один інтерфейс, тому вони зручніші при створенні системи, яка використовує більше ніж як один датчик. Однак, якщо система не потребує додаткових інтерфейсів зв'язку, то при її розробці доцільно використовувати варіант з меншою кількістю інтерфейсів, оскільки вони значно дешевші (рисунок 1.9).

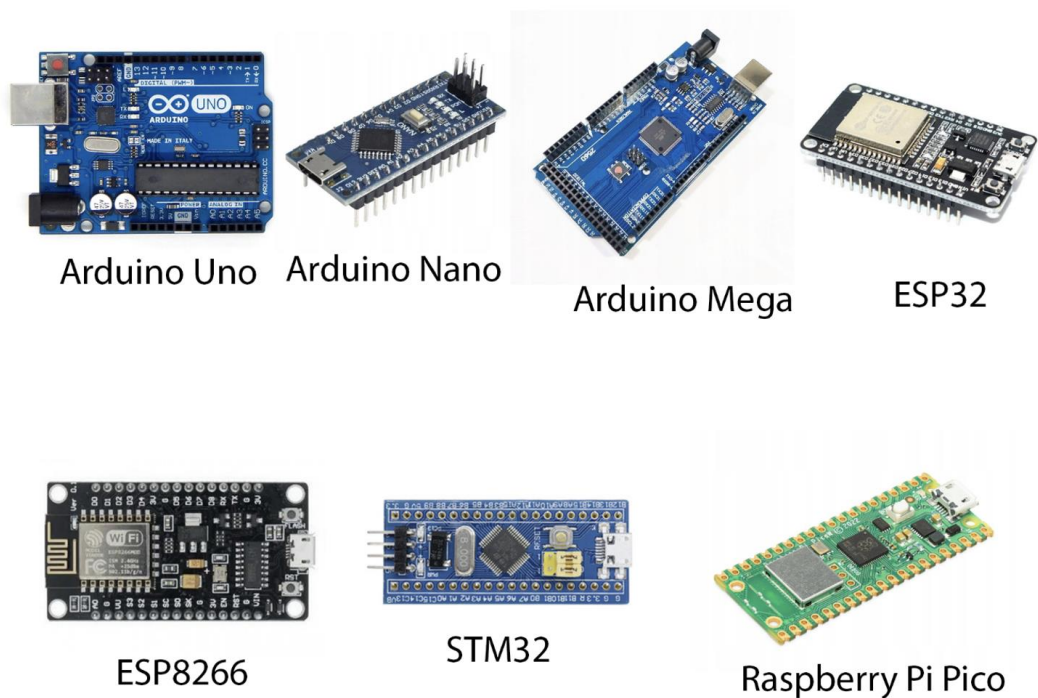


Рисунок 1.9 – Зовнішній вигляд найпопулярніших мікроконтролерів

Для системи, яка складається з декількох компонентів важливою є кількість роз'ємів підключення. Найбільшою кількістю роз'ємів вирізняється Arduino Mega. Однак при потребі можна використовувати спеціальні плати розширення, які роблять можливим підключення багатьох пристроїв. Для

систем автоматизованого вимірювання відстані при більшості потреб достатньо роз'ємів, які є на кожному з мікроконтролерів.

Наступною важливою характеристикою є обсяг пам'яті. Пам'ять у мікроконтролерів відповідає за збереження даних програми або іншої інформації, яка повинна бути використана в процесі роботи системи. Якщо у системі вимірювання відстані реалізовується функціонал збереження вимірів, доцільно вибирати мікроконтролер з достатньою кількістю пам'яті. При потребі можна використовувати додаткові модулі для збільшення обсягу пам'яті (наприклад, модуль SD-карти).

Якщо у системі планується передача даних в додаток, на сайт або вебсервер, або віддалене керування системою, доцільно використовувати платформи, які мають технології бездротової передачі інформації (Wi-Fi, Bluetooth або інші).

Також, важливим фактором є кількість споживаної енергії мікроконтролером. Це більш важливо для мобільних та автономних систем з живленням від батареї. Серед вище описаних платформ найменше енергоспоживання мають STM32 або Raspberry Pi Pico. Інші платформи мають більше енергоспоживання, однак різниця не критична [10].

З огляду на всі наведені фактори зроблено висновки, що для нескладних систем вимірювання з простим функціоналом достатньо можливостей, які надають Arduino Uno чи Arduino Nano. Якщо у системі вимірювання відстані передбачається підключення багатьох датчиків, то доцільно використовувати Arduino Mega. Для передачі даних на сервер чи у додаток підійдуть мікроконтролери ESP32 та ESP8266. Якщо завданням є портативність та мобільність системи і важливе енергоспоживання, тоді більш підходящими є STM32 чи Raspberry Pi Pico. У будь-якому випадку, при потребі розширити функціонал мікроконтролеру доступні окремі модулі, які зможуть задовольнити всі потреби.

## 1.5 Огляд типових застосувань

Системи автоматизованого вимірювання відстані активно використовуються у багатьох сферах діяльності, а саме у промисловій, побутовій, інженерній та інших. Завдяки простоті реалізацій, доступності компонентів та ефективній роботі такі системи користуються великою популярністю.

Основні напрямки застосування включають: автомобільну промисловість, робототехніку, охоронні системи, інтелектуальні системи, медичні пристрої, сільське господарство тощо. У різних випадках використання систем вимірювання може мати свої особливості.

В автомобільній промисловості системи вимірювання відстані можуть використовувати у створення паркувальної системи для визначення відстані до перешкоди при задньому русі. Також, їх можна використовувати для автоматичного визначення перешкод у системах уникнення зіткнень.

В робототехніці такі системи використовуються для навігації дронів та роботів у приміщеннях і на відкритих територіях. Також, прикладом застосування систем вимірювання відстані, у цій сфері, може бути побудова карт місцевості та уникнення перешкод під час використання мобільних роботів.

У промисловості та сільському господарстві системи вимірювання відстані знайшли місце при визначенні рівня заповнення резервуарів, при автоматизації транспортувальних систем, для автоматичного наповнення кормушок та ємностей з водою тощо.

В сфері безпеки та охорони системи віддаленого вимірювання відстані можуть використовуватися для виявлення руху або несанкціонованого проникнення. Також, на основі датчиків руху можна побудувати систему сигналізації.

У інтелектуальних системах системи вимірювання відстані можуть використовуватися для автоматичного відкривання дверей, а також для ввімкнення або вимкнення світла при виявленні руху

Усі ці приклади демонструють широкий спектр застосувань систем автоматизованого вимірювання відстані, а також доводять їх практичність, ефективність, універсальність та затребуваність у різних сферах та умовах експлуатації. Завдяки простоті реалізації та доступності такі системи широко застосовуються у навчальних, дослідницьких та комерційних проєктах.

Отже, у ході аналізу основних питань розробки систем автоматизованого вимірювання відстані було розглянуто принципи роботи таких систем та виявлено засоби, які необхідні для реалізації. Також, було створено порівняльну характеристику ключових засобів, зокрема датчиків вимірювання та платформ розробки. Було оглянуто типові застосування систем автоматизованого вимірювання відстані та виявлено практичність та актуальність таких систем у різних сферах діяльності і на основі аналізу поставлено задачі на дослідження.

## РОЗДІЛ 2

### МЕТОДИКА РОЗРОБКИ ТА ОСНОВНІ ІНСТРУМЕНТИ

#### 2.1 Технічні вимоги до системи

При розробці системи автоматизованого вимірювання відстані слід враховувати функціональні вимоги, компоненти системи та їх технічні характеристики. Однак, важливим є не тільки функціонал, але й інтерфейс, який дозволить користувачеві краще працювати з системою.

Розроблювальна система повинна включати компоненти, які зможуть забезпечити наступні функціональні можливості: вимірювання відстані до об'єкту по натиску кнопки, виведення результатів вимірювання на дисплей, світлова індикація стану пристрою та система увімкнення та вимкнення пристрою.

Для реалізації такого функціоналу необхідні наступні компоненти: мікроконтролер, ультразвуковий датчик вимірювання, дисплей, світлодіоди різни кольорів або RGB-світлодіод, кнопка для увімкнення/вимкнення пристрою та кнопка для початку виміру.

У системах вимірювання відстані основним компонентом є датчик відстані, тому його технічні характеристики відіграють ключову роль. Першою з них є діапазон вимірювання. Він повинен міряти в діапазоні до 400 сантиметрів з точністю до 0,3 сантиметрів або краще. Також, датчик повинен взаємодіяти з користувачем через простий інтерфейс.

Для зручності і енергоефективності у системі вимірювання слід реалізувати систему увімкнення/вимкнення. Для цього можна використовувати окрему кнопку. При подачі живлення на систему вона повинна залишатися у вимкненому стані. При натисненні на кнопку система повинна вмикатися та бути доступною для початку вимірів. При натисненні на кнопку, коли система увімкнена, кнопка повинна вимикати систему.

Інша кнопка повинна відповідати за початок виміру і вивід результатів на дисплей. Спрацьовувати вона повинна тільки при увімкненій системі, при

цьому її дії повинні бути циклічними при повторному натисненні. Для покращення результатів вимірювання, при натисненні на кнопку початку виміру, датчик вимірювання може робити відразу декілька замірів, які потім обраховуються.

Також, для покращення користувацького досвіду на дисплеї можна реалізувати анімації, логотипи та інші графічні елементи, які дозволяє вивести вибраний дисплей.

Усі вищеописані вимоги необхідно врахувати при реалізації системи автоматизованого вимірювання відстані. Їх виконання забезпечить універсальну, просту та зручну систему вимірювання відстані.

## **2.2 Обґрунтування вибору апаратних та програмних засобів**

Для виконання технічних вимог до системи вимірювання відстані було обрано наступні апаратні засоби: плата Arduino Uno R3 (репліка), ультразвуковий датчик відстані US-025, LCD 1602 дисплей 16x2 (синій), модуль кнопки з ковпаком у кількості 2 штук та RGB-світлодіод KY-016. Такий набір компонентів забезпечить коректну роботу системи, вивід результатів вимірювання та світлову індикацію станів системи [11].

Репліка оригінальної плати Arduino Uno R3 на базі мікросхеми CH340 зарекомендувала себе надійністю, простотою в роботі та великою кількістю підтримуваних бібліотек і модулів. Її головна відмінність – використання CH340, як USB-UART перетворювача. Від попередників вона відрізняється наявністю додаткових роз'ємів SDA та SCL (інтерфейс зв'язку I2C) та іншими додатковими входами/виходами. Ця плата використовує мікроконтролер ATmega328, такий як і оригінальна плата, це дозволяє використовувати абсолютно ті самі бібліотеки, методи, конструкції коду та приклади, що і оригінальна плата. Суттєвою перевагою цієї репліки над оригіналом є ціна, яка в 5 разів менша.

Характеристики плати повністю задовольняють поставлені технічні вимоги. Плата має мікроконтролер ATmega328, 14 цифрових входів/виходів та 6 аналогових, флеш-пам'ять – 32 Кб, оперативна пам'ять – 2 Кб та тактова частота – 16 МГц.

Плата має невеликі розміри (рисунок 2.1), а саме 68 на 53 на 15 міліметрів. На платі є два зовнішні роз'єми: для живлення – Vin, для програмування або живлення – USB Micro-B. Також плата має вмонтовану кнопку для перезапуску плати Reset.

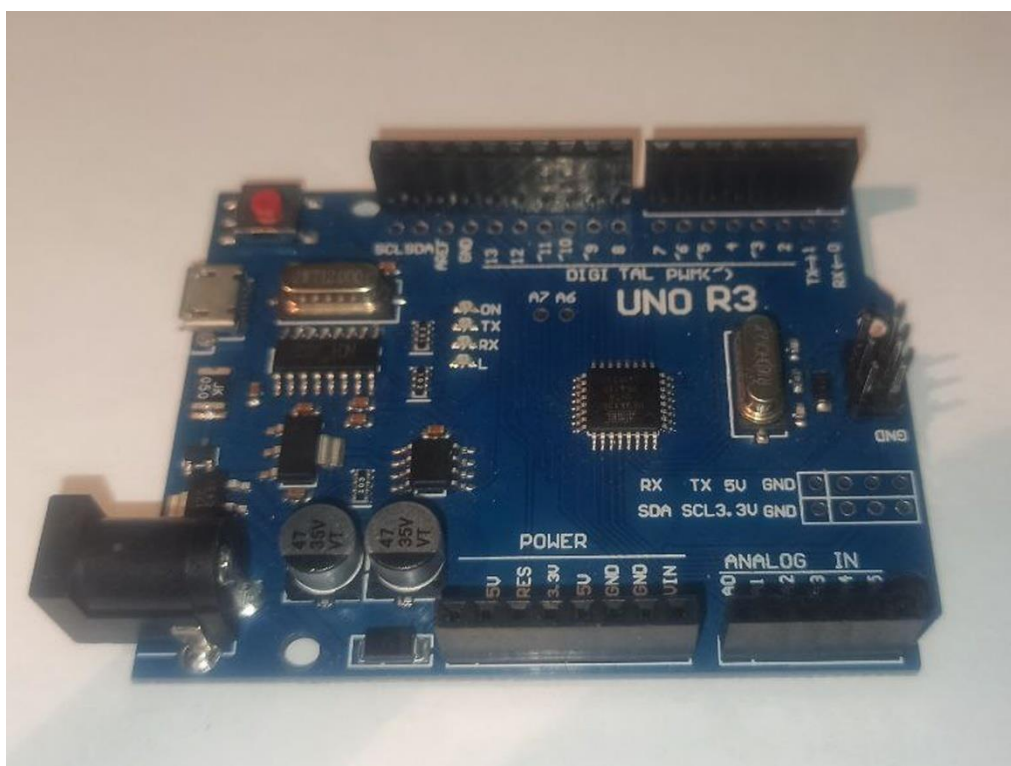


Рисунок 2.1 – Китайська версія плати Arduino Uno R3

Для вимірювання відстані було обрано ультразвуковий датчик вимірювання відстані US-025 (рисунок 2.2). Цей датчик відомий своєю точністю і відсутністю «сліпих зон». Він може вимірювати відстань від 2 до 600 сантиметрів з точністю до 3 мм. US-025 простий у підключенні. Він має 4 виходи: VCC, GND, Echo, Trig. Принцип роботи, як і у інших ультразвукових датчиках, полягає у відправленні ультразвукових імпульсів і отримання відлуння.

Даний датчик користується великою популярністю у системах вимірювання відстані. Завдяки своїй популярності він має багато написаних бібліотек для роботи з ним.

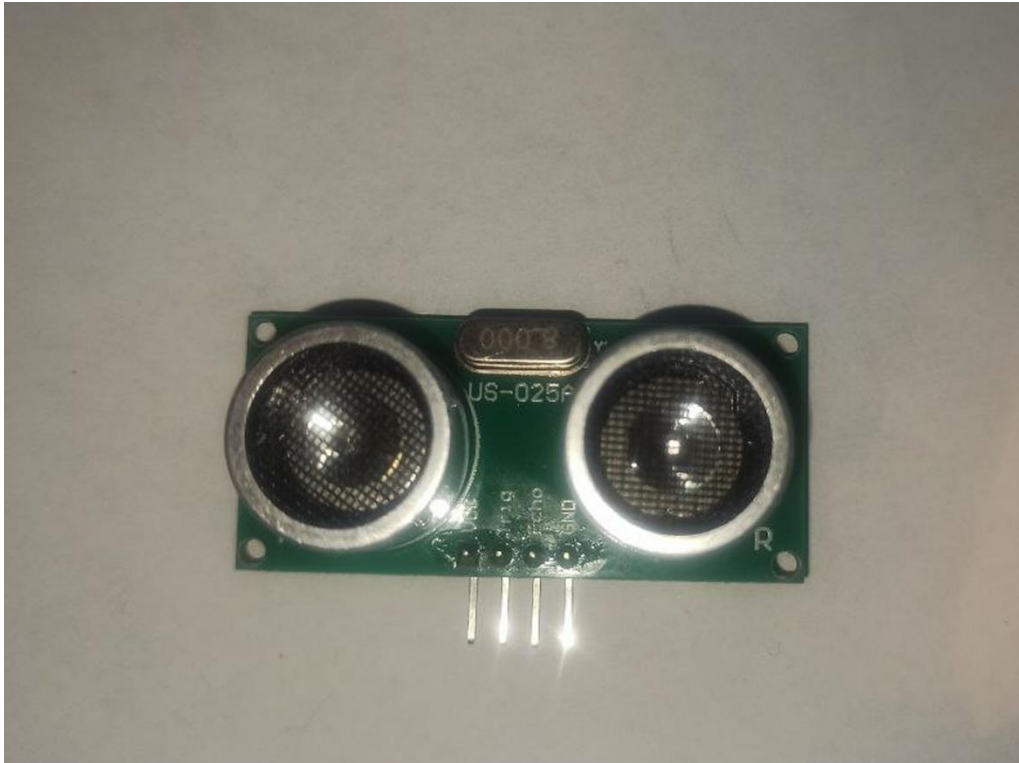


Рисунок 2.2 – Ультразвуковий датчик відстані US-025

Для керування пристроєм було використано два модулі кнопки типу push button, для індикації станів RGB-світлодіод KY-016 (рисунок 2.3).

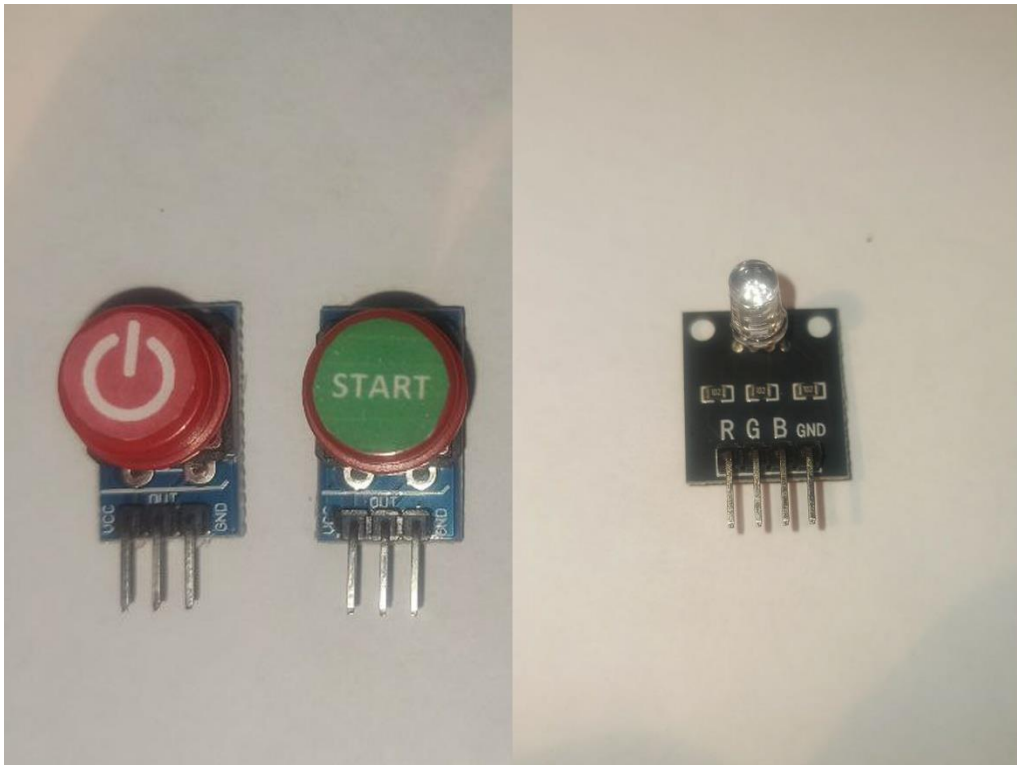


Рисунок 2.3 – Кнопки для керування пристроєм та RGB-світлодіод KY-016

Одна кнопка повинна відповідати за ввімкнення/вимкнення пристрою, інша – за початок вимірювання. Світлодіод повинен відповідати за індикацію станів пристрою. Переваги RGB-світлодіоду в тому, що він може показувати декілька кольорів з однієї лампочки.

Для виводу даних було обрано LCD 1602 дисплей 16x2 (рисунок 2.4). Він надає достатній функціонал для виводу результатів вимірювання. Переваги його в тому, що він простий у використанні. Також, для роботи з ним є багато готових бібліотек. Для простоти підключення дисплею було обрано модуль I2C. Він зменшує кількість виходів з 16 до 4.



Рисунок 2.4 – LCD 1602 дисплей 16x2 з модулем I2C

Такий перелік апаратних засобів дозволить реалізувати систему автоматизованого вимірювання відстані. Таким чином він повністю задовольняє технічні вимоги щодо системи.

У роботі з апаратними засобами однозначним лідером є Arduino. Arduino – це відкрита платформа для роботи з різноманітними пристроями. Вона є безкоштовною і дає повний доступ до функцій та бібліотек для роботи з різними електричними пристроям. Таким чином ця платформа ідеально підходить для реалізації функціоналу системи автоматизованого вимірювання відстані.

У ході обґрунтування вибору апаратних і програмних засобів було обрано оптимальний набір компонентів що забезпечує повну відповідність технічним та функціональним вимогам до системи. Також, було обрано платформу програмної розробки Arduino, завдяки її доступності, зручності та широкій спільноті користувачів.

### **2.3 Схема з'єднання елементів системи на платформі Arduino**

Перед реалізацією системи автоматизованого вимірювання відстані було створено фізичну схему з'єднання елементів системи (рисунк 2.5) та принципову електричну схему. Це необхідно для розуміння і чіткої побудови системи. Фізична схема включає обрані компоненти, або аналоги, які по принципу підключення не відрізняються. Вона ілюструє розташування елементів та візуально показує як вони з'єднані між собою

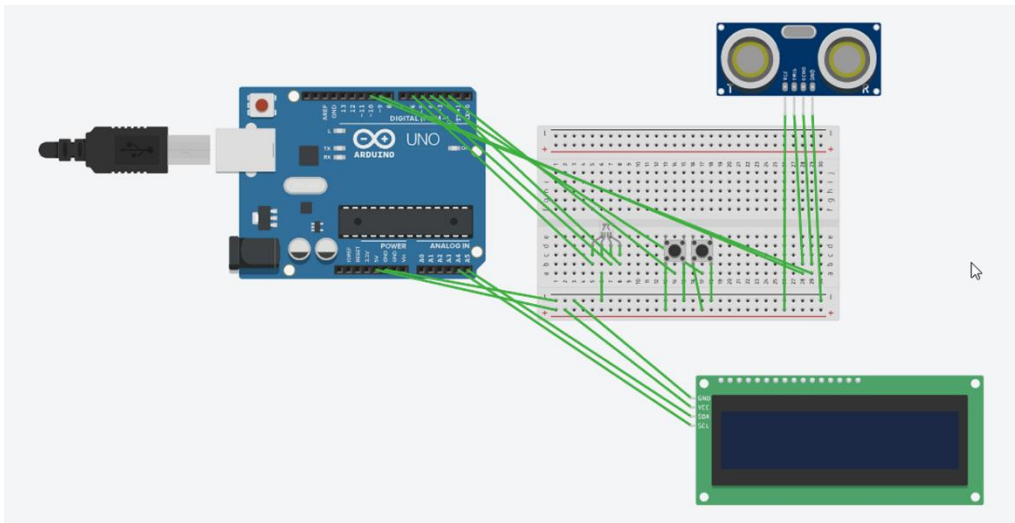


Рисунок 2.5 – Фізична схема з'єднання елементів системи

Схема передбачає з'єднання компонентів через макетну плату. Це спростить процес розробки та монтування системи, оскільки не потребує паяння елементів. Макетна плата відіграє роль «містка» між платою та компонентами.

Розглянемо принципи підключення кожного з компонентів. Перш за все на макетну плату необхідно подати живлення. За це на платі відповідають виходи 5V та GND, які підключаються до будь-яких з роз'ємів «+» (червона лінія) і «-» (синя лінія) відповідно. Тепер, з інших роз'ємів цих двох ліній доступні позитивна шина живлення та негативна (нульовий потенціал або «земля»).

Ультразвуковий датчик вимірювання відстані US-025 має піни підключення: VCC, GND, Trig та Echo. Усі ці піни втикаються у макетну плату

перпендикулярно осей підключення. VCC – це вхід, на який подається напруга живлення. На макетній платі його можна підключити до будь-якого з порожніх роз'ємів червоної лінії. GND – це загальний провід (нульовий потенціал), який використовується для замикання кола живлення. На макетній платі він підключається до синьої лінії. У будь-який з контактних рядів, до якого підключено Trig, необхідно вставити з'єднувальний кабель. З іншої сторони його потрібно підключити до будь-якого цифрового піна на платі. На схемі це D10. Trig – це вихідний пін. Сюди мікроконтролер подає короткий імпульс, щоб запустити вимірювання. Аналогічно потрібно підключити і вихідний пін Echo. На ньому з'являється імпульс, тривалість якого залежить від часу повернення звукової хвилі. На схемі цей пін підключається до цифрового роз'єму плати D9.

LCD 1602 дисплей з модулем I2C має теж 4 піни: VCC, GND, SDA та SCL. VCC та GND виконують ті ж самі функції, як і у датчику відстані (вони є стандартними пінами живлення для всіх компонентів). SDA – це роз'єм, який відповідає передаванню даних в модулі I2C. На схемі цей він підключається до аналогового роз'єму A5. SCL відповідає за синхронізацію і задає тактову частоту передавання даних. На схемі підключається до аналогового роз'єму A4.

Кнопки, які повинні вмикати/вимикати пристрій та починати вимірювання є ідентичними і мають по 3 піни підключення: VCC, GND та OUT. OUT відповідає за передавання сигналу стану кнопки (0 або 1). Для кнопки, яка вмикає/вимикає пристрій на схемі OUT підключений до цифрового роз'єму D2, а для кнопки початку виміру до D4.

RGB-світлодіод KY-016 має 4 піни: R, G, B та GND. Піни R, G та B відповідають за керування певним кольором (R – червоний, G – зелений, B – синій) і підключаються до цифрових пінів D3, D4, та D6.

Принципова електрична схема призначена для відображення з'єднання елементів з точки зору електроніки. Вона показує лінії сигналів, живлення, резистори, транзистори тощо. Принципову електричну схему для системи автоматизованого вимірювання відстані зображено на рисунку 2.6.

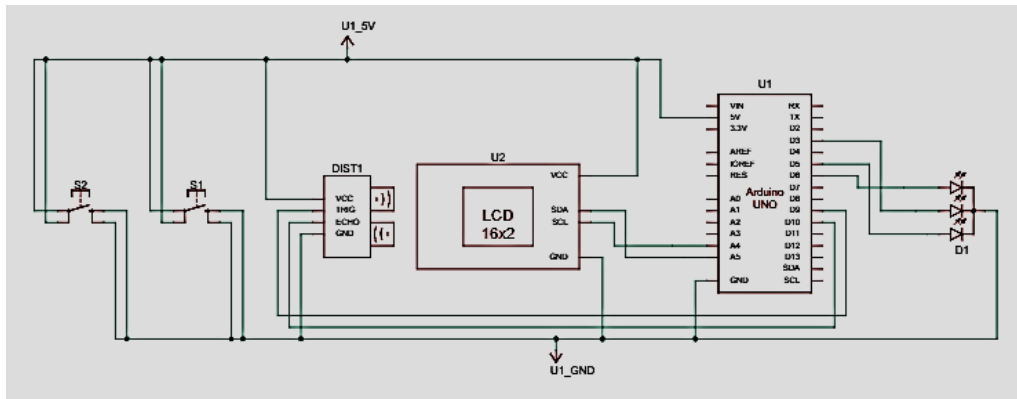


Рисунок 2.6 – Принципова електрична схема системи автоматизованого вимірювання відстані

Таким чином, побудовані схеми дозволяють чітко визначити логіку підключення усіх компонентів та перевірити можливість реалізації системи з обраних компонентів. Цей етап є необхідним перед реалізацією системи, оскільки він забезпечує зручність подальшого монтажу, тестування та обслуговування системи.

#### 2.4 Опис алгоритму роботи системи

Для забезпечення коректної роботи системи автоматизованого вимірювання відстані необхідно чітко визначити порядок дій, який буде виконуватися в процесі її функціонування. Алгоритм роботи системи описує логіку взаємодії між користувачем, ультразвуковим датчиком контролером Arduino та іншими елементами системи. На рисунку 2.7 наведено поетапний опис процесу роботи системи. Алгоритм дозволяє забезпечити узгоджену роботу всіх компонентів та реагування на дії користувача у реальному часі.

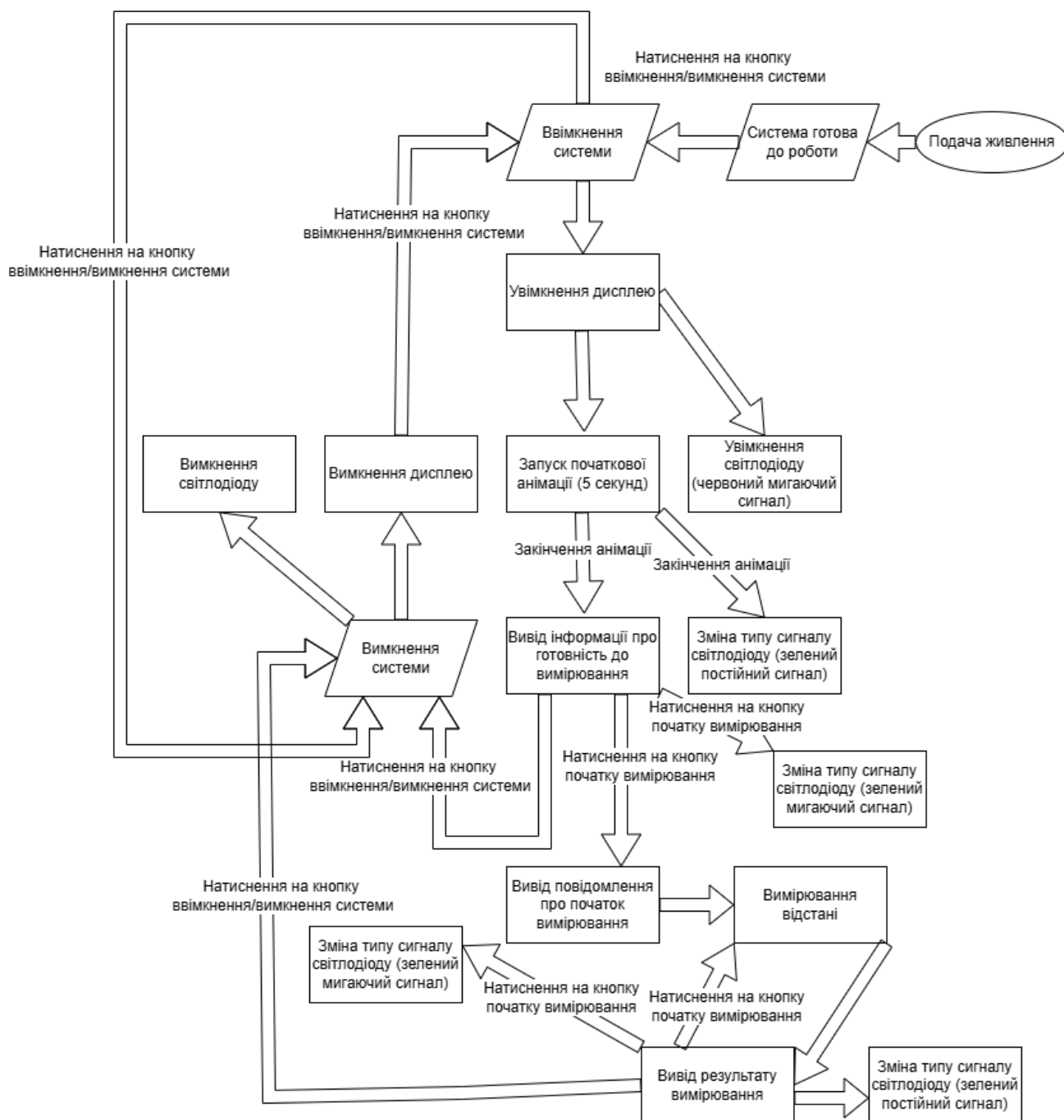


Рисунок 2.7 – Алгоритм роботи системи автоматизованого вимірювання відстані

Перший етап алгоритму роботи системи автоматизованого вимірювання відстані включає фізичну подачу живлення на мікроконтролер та всі компоненти. На цьому етапі необхідно підключити джерело живлення, в ролі якого може виступати блок живлення, повербанк, батарея, акумулятор тощо. Після подачі живлення система готова до роботи і виконання вказівок користувача.

Після ввімкнення системи усі компоненти залишаються у «сплячому» режимі. Це означає, що вони не працюють без вказівок. Єдина функція, яка доступна на цьому етапі – ввімкнення пристрою. Для того, щоб зробити це, користувач повинен натиснути на кнопку ввімкнення/вимкнення системи.

Після натиснення на кнопку ввімкнення/вимкнення починається запуск пристрою. На цьому етапі спочатку відбувається ввімкнення дисплею. Далі паралельно запускається початкова анімація на дисплеї та починає мигати червоний світлодіод. Сигнал світлодіоду означає, що пристрій запускається і ще не готовий до роботи.

Наступний етап починається після закінчення початкової анімації. На екран виводиться повідомлення про готовність системи до початку вимірювання. Також, сигнал RGB-світлодіоду змінюється на стабільний зелений. Якщо на цьому етапі натиснути кнопку ввімкнення/вимкнення системи, то система вимикається і повертається у «сплячий режим». Під час вимкнення системи гасне світлодіод та дисплей.

Також, на цьому етапі доступний функціонал початку виміру. Виконання функції вимірювання починається після натиснення на кнопку початку вимірювання. Після цього на дисплей виводиться повідомлення про те, що проводиться вимірювання, сигнал світлодіоду при цьому змінюється на зелений мигаючий.

Після закінчення вимірювання на дисплей виводиться результат. Світлодіод при цьому повертається до попереднього режиму (зелений постійний). Після цього можна знову запуснути вимірювання довільну кількість разів. Алгоритм при цьому буде такий самий. Також, можна натиснути кнопку вимкнення системи. Якщо на будь-якому етапі роботи джерело живлення відключається, система перестає працювати і просто вимикається.

Описаний алгоритм роботи системи автоматизованого вимірювання відстані забезпечує чітку логіку роботи системи та взаємодію між користувачем, програмними та апаратними засобами. Завдяки використанню поетапної логіки було спроектовано, яким чином система буде реагувати на дії

користувача, як буде виконуватися ввімкнення/вимкнення системи, запуск процесу вимірювання, індикація станів системи та вивід результатів на дисплей. Даний алгоритм робить систему інтуїтивно зрозумілою та зручною у користуванні. Такий підхід дозволяє передбачити наперед недопрацьованості системи та реалізувати функціонал системи автоматизованого вимірювання відстані найкраще.

## **2.5 Підготовка програмного середовища та використання бібліотек**

Для реалізації функціональних вимог системи автоматизованого вимірювання відстані на основі платформи Arduino необхідно використати відповідне програмне забезпечення, методи програмування та бібліотеки. Програмна частина системи забезпечує реакцію усіх компонентів на дії користувача, а саме: зчитування даних з ультразвукового датчика, виведення інформації на дисплей, керування світлодіодом, обробка натиснень на кнопки, а також реалізацію логіки згідно вище описаного алгоритму роботи системи.

Деякі компоненти системи є дещо складнішим і потребують спеціальних бібліотек для роботи з ними. Інші є елементарними, тому робота з ними простішою.

Для роботи з платою Arduino Uno R3 необхідно підготувати середовище Arduino IDE до роботи. Для цього спочатку потрібно підключити плату до комп'ютера де встановлено Arduino IDE. Після цього потрібно перевірити чи комп'ютер (ОС Windows) ідентифікує плату. Для цього потрібно зайти в «Диспетчер пристроїв» та в розділі «Порти» знайти пристрій, який буде називатися відповідно до процесора використовуваного платою. Оскільки в системі використовується репліка оригінальної плати Arduino Uno R3 на базі мікросхеми CH340, то в списку пристроїв повинен розпізнаватися пристрій саме з підписом CH340 (рисунок 2.8). Для операційних систем macOS та більшості дистрибутивів Linux підтримка CH340 часто вже вбудована в ядро системи, або може бути встановлена через менеджер пакетів.

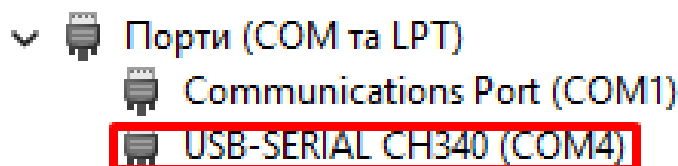


Рисунок 2.8 – Розпізнана плата Arduino Uno R3 (CH340)

Якщо після підключення плати в «Диспетчері пристроїв» не з'являється новий пристрій, або є іконка чи надпис про нерозпізнаний пристрій, то необхідно встановити драйвер для процесору плати. У випадку з реплікою плати Arduino Uno R3 на базі мікросхеми CH340, потрібно встановити драйвер для CH340, який є у вільному доступі в інтернеті, або надається продавцем плати.

Після того як комп'ютер ідентифікував плату потрібно ініціювати пакети необхідні для роботи з нею в Arduino IDE та вибрати необхідні налаштування для роботи з платою. У вікні вибору плат потрібно вибрати порт, до якого підключена плати Arduino Uno R3 (рисунок 2.9).

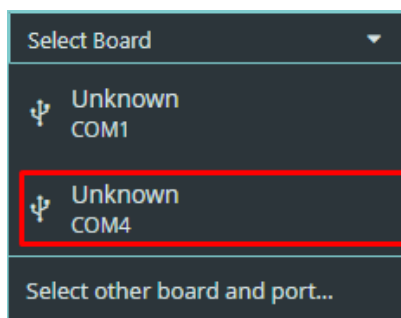


Рисунок 2.9 – Список доступних для роботи плат і портів, до яких вони підключені

У наступному вікні яке відкриється потрібно знайти плату Arduino Uno, вибрати порт, до якого підключена плата та натиснути «ОК». Цей тип є стандартним і підходить для плати Arduino Uno R3, яка використовується в системі вимірювання відстані (рисунок 2.10).

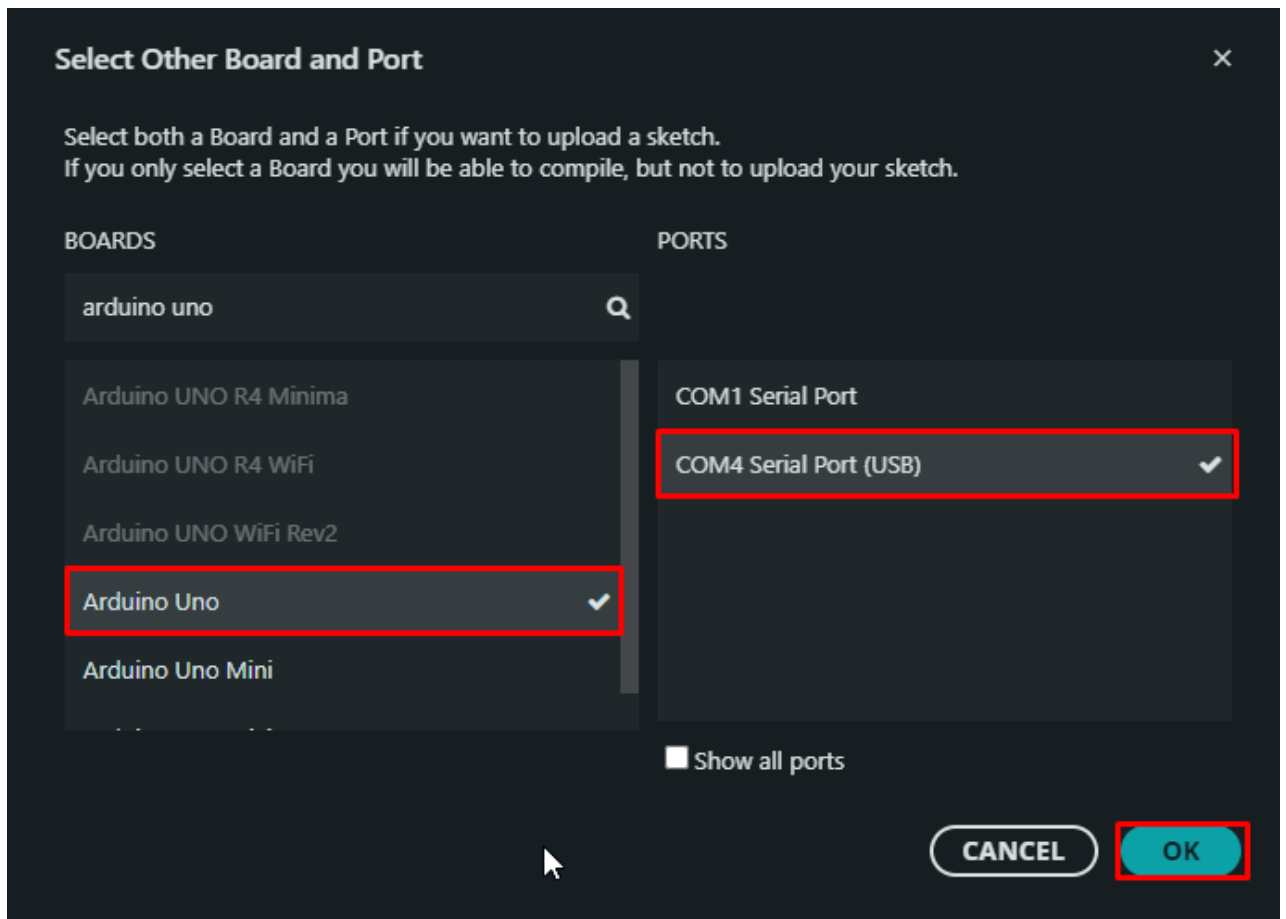


Рисунок 2.10 – Вікно вибору плати

Після цього у списку вибору буде доступна плата з назвою Arduino Uno. Для того, щоб перевірити роботу плати можна скористатися прикладами коду, які є стандартними для Arduino IDE, або ж написати тестовий код, який перевірить чи може програма взаємодіяти з платою і виведе звіт у Serial Monitor.

Один і з найпростіших способів перевірки надає саме середовище. Можна завантажити приклад «Blink» з меню Файл/Приклади/01.Basics/Blink (рисунок 2.11).

Цей код примусить вбудований світлодіод на платі мигати з інтервалом в одну секунду. Якщо світлодіод блимає, то плата готова до подальшої роботи.

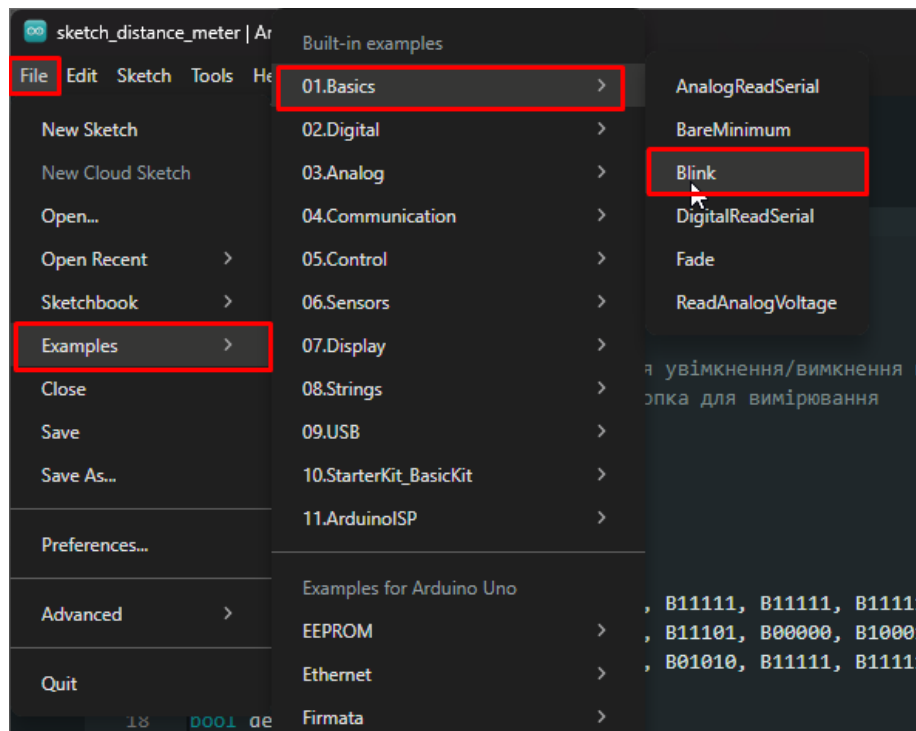


Рисунок 2.11 – Вибір тестового коду Blink

Для того, щоб скомпілювати код та перевірити його на помилки в середовищі Arduino IDE потрібно натиснути на кнопку 2, яка зображена на рисунку 2.12, а для завантаження коду в плату використовується кнопка 1.

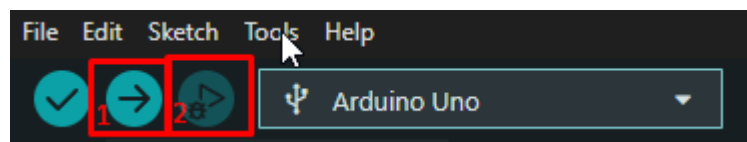


Рисунок 2.12 – Кнопки для компіляції і завантаження коду в середовищі Arduino IDE

Для роботи з ультразвуковим датчиком вимірювання відстані потрібно використовувати додаткові бібліотеки. Перша з них бібліотека Wire. Вона відповідає за роботу з інтерфейсом I2C, який використовує дисплей LCD 1602. Функції цієї бібліотеки в забезпечені обміну даними між Arduino і дисплеєм та в ініціалізації зв'язку чи відправлення/отримання даних по шині I2C.

Друга бібліотека необхідна для роботи з дисплеєм LiquidCrystal\_I2C. Вона дозволяє керувати дисплеєм LCD 1602 через I2C-перехідник. Також, ця

бібліотека значно спрощує роботу з дисплеєм, зменшуючи кількість пінів, які необхідні для підключення, а також надає зручні методи для роботи з дисплеєм, такі як: `.begin()`, `.print()`, `.setCursor()`, `.clear()` тощо.

## 2.6 Методи перевірки та тестування роботи системи

Під час розробки системи автоматизованого вимірювання відстані можуть виникнути проблеми з роботою певних компонентів. Для впевненості у правильному функціонуванні було сформовано методи тестування, як компонентів по окремоті, так і системи вцілому.

Перший метод пов'язаний з перевіркою апаратного з'єднання компонентів. Для тестування можна використовувати мультиметр в режимі прозвонки. Це дозволяє перевірити цілісність провідників, місць пайки, а також переконатися у відсутності коротких замикань.

Другий метод передбачає перевірку кожного компонента системи за допомогою простих тестових скретчів у середовищі Arduino IDE. Наприклад, для перевірки дисплею можна використати приклади скриптів, які доступні разом із бібліотекою `LiquidCrystal_I2C`. Він дозволяє вивести тестовий текст на дисплей.

Окрему увагу слід приділити тестуванню кнопок. Для цього можна написати скетч, який при натисненні кнопки виводить повідомлення в монітор порту або змінює стан світлодіода. Це дозволяє впевнитися, що кнопки коректно підключені та реагують на дії користувача.

Заключним етапом є перевірка системи на правильність роботи вцілому. На цьому етапі необхідно вручну перевірити роботу системи згідно запланованого алгоритми. Також, потрібно перевірити правильність вимірювання через вимірювання заздалегіть відомої відстані до об'єкту.

Таким чином, застосування поетапного підходу до тестування дозволяє вчасно виявити та усунути несправності, забезпечуючи стабільну та надійну роботу всієї системи.

## РОЗДІЛ 3

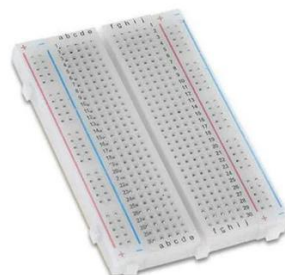
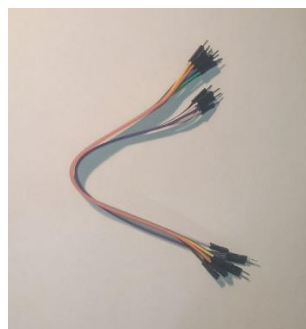
### РЕАЛІЗАЦІЯ СИСТЕМИ ВИМІРЮВАННЯ ВІДСТАНІ

У цьому розділі розглянуто практичну реалізацію системи автоматизованого вимірювання відстані. Детально описано процес збирання апаратної частини, налаштування мікроконтролера Arduino та програмування системи згідно з розробленим алгоритмом. Також наведено приклади виведення результатів вимірювання, проведено випробування системи в різних умовах та здійснено аналіз точності, стабільності й швидкодії отриманих результатів.

#### 3.1 Монтаж апаратної частини системи

Монтаж апаратної частини системи автоматизованого вимірювання відстані використовує безпачний метод. Цей метод передбачає з'єднання компонентів системи через макетну плату, за допомогою спеціальних кабелів (джамперів). Переваги такого методу в простоті реалізації, швидкості збирання та можливості багаторазового використання компонентів без їх пошкодження. Крім того, безпачне з'єднання дозволяє легко змінювати конфігурацію системи під час налагодження чи тестування.

Для монтування системи було використано макетну плату MB-102 на 400 контактів та джампери типу «Тато-Тато» і «Тато-Мама», довжиною 20 сантиметрів. Такого набору достатньо для забезпечення з'єднань між компонентами (рисунок 3.1).



### Рисунок 3.1 – Макетна плата та джампери для монтування системи

Також, для створення з'єднань між контактами макетної плати були використані спеціальні з'єднувальні дроти. Вони корисні тим, що мають тверді контакти, які зручно розміщувати на макетній платі.

Монтаж системи автоматизованого вимірювання відстані починається з підключення системи живлення з мікроконтролеру Arduino Uno R3 на макетну плату. Для цього потрібно на контролері знайти піни 5V (позитивна фаза) та GND (нульова або негативна фаза). Позитивна фаза підключається до червоної лінії на макетній платі, негативна – до синьої (рисунок 3.2). Після підключення на будь-якому з роз'ємів буде доступна відповідна фаза.

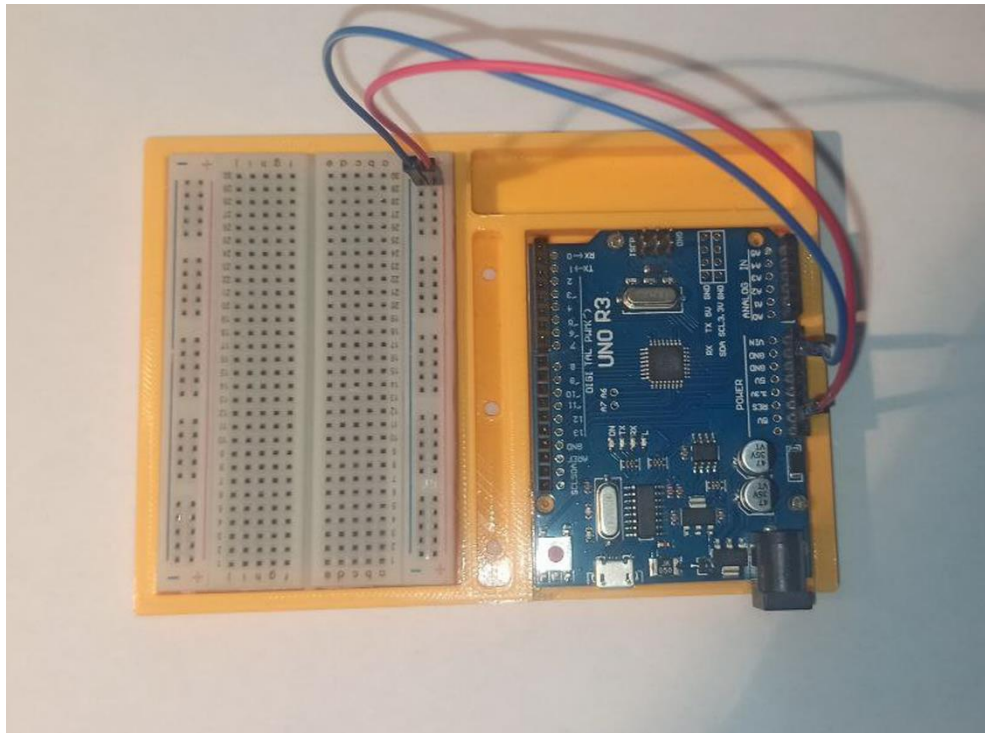


Рисунок 3.2 – Підключення живлення від мікроконтролера до макетної плати

Далі необхідно підключити ультразвуковий датчик вимірювання відстані US-025. Спочатку вставити датчик у відповідні роз'єми на макетній платі. Потім потрібно з'єднати пін VCC датчика до червоної лінії живлення, на яку раніше було підключено кабелі живлення від контролера. Також, потрібно підключити пін GND датчика до синьої лінії макетної плати. Процес

підключення датчика відстані та подачу живлення на нього зображено на рисунку 3.3. Пін TRIG потрібно підключити через макетну плату і джампери «Тато-Тато» до цифрового піну D9, а пін ECHO – до D10.

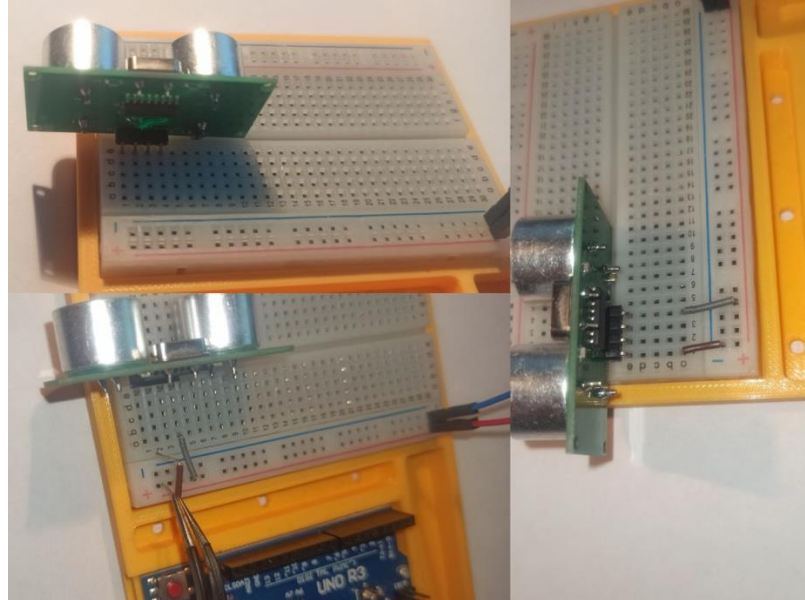


Рисунок 3.3 – Підключення датчика відстані та подача живлення на нього

Наступним етапом монтажу системи є підключення кнопок для увімкнення/вимкнення пристрою та для початку виміру (рисунок 3.4).

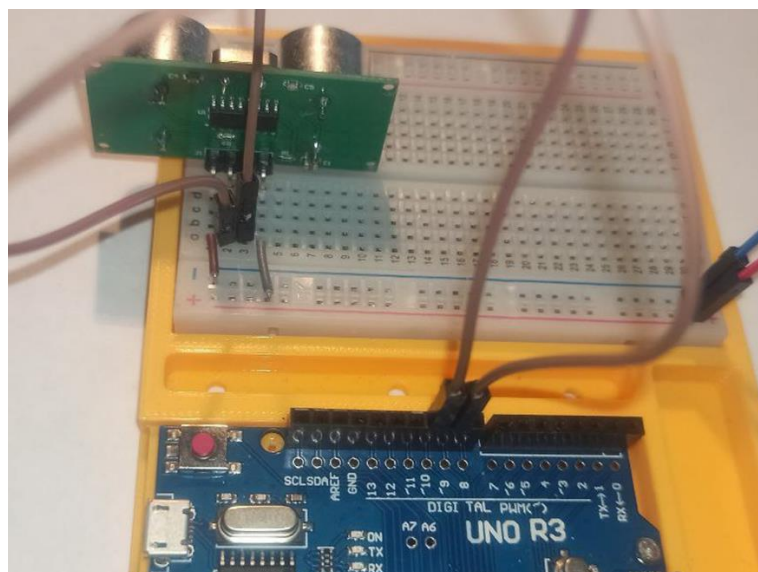


Рисунок 3.4 – Підключення пінів датчика відстані Echo та Trig до мікроконтролера

Для цього потрібно вставити кнопки у вільні роз'єми на макетній платі, підключити до ліній живлення піни VCC та GND, а також підключити піни OUT до вільних цифрових пінів на мікроконтролері. Пін OUT кнопки увімкнення/вимкнення було підключено до цифрового піна D2 на мікроконтролері, а для кнопки початку виміру – до D4. Усі етапи підключення двох кнопок зображено на рисунку 3.5.

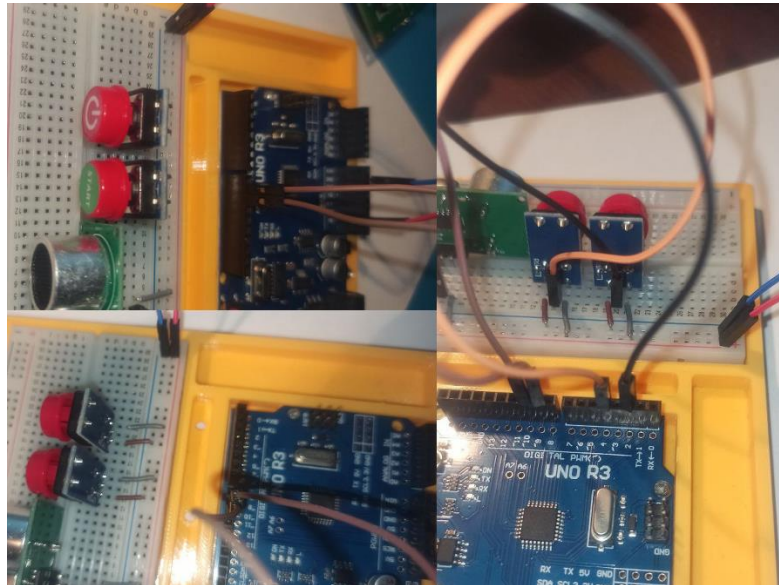


Рисунок 3.5 – Етапи підключення кнопок у систему

Далі було підключено RGB-світлодіод до макетної плати та до контролера. Пін GND світлодіода було з'єднано з синьою лінією на макетній платі. Піни R, G та B було підключено до цифрових пінів D6, D5 та D3 відповідно (рисунок 3.6).

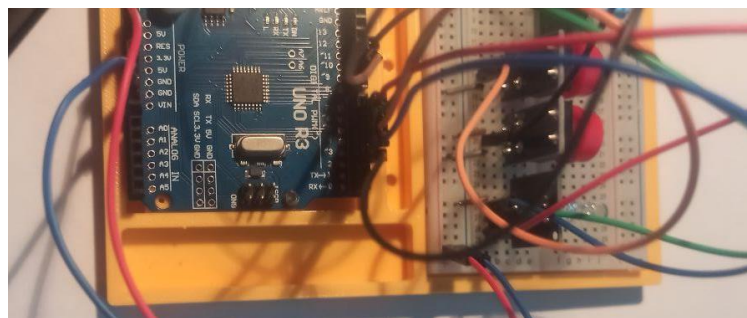


Рисунок 3.6 – Монтуванні RGB-світлодіоду в систему вимірювання

Перед підключенням LCD дисплею було припаяно I2C адаптер, оскільки він поставлявся окремо від дисплею (рисунок 3.7). Процес паяння зображено на рисунку 3.8.

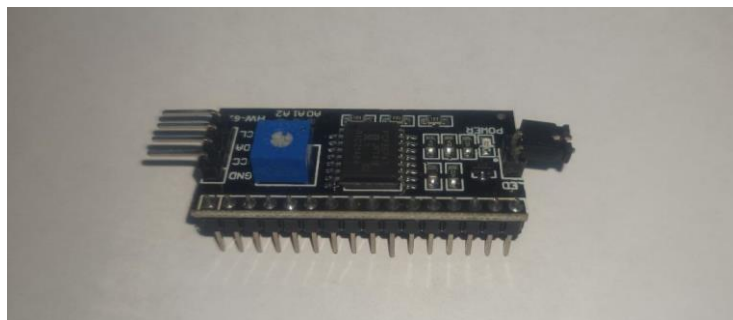


Рисунок 3.7 – Модуль інтерфейсу I2C для LCD дисплею

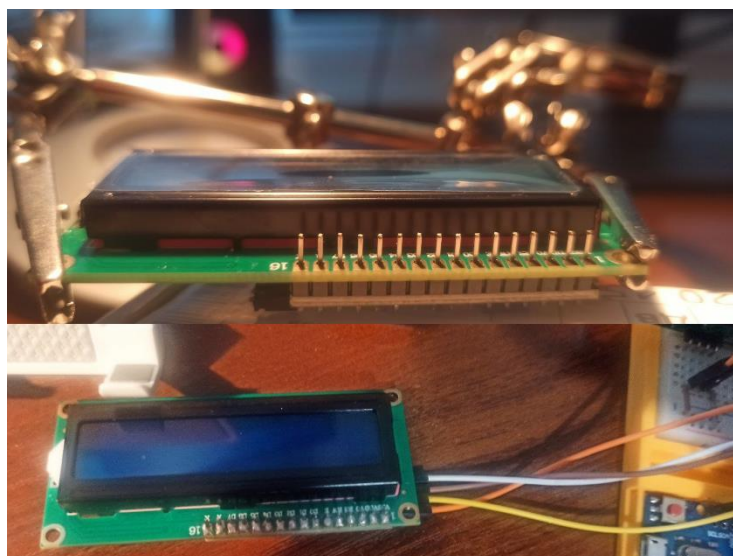


Рисунок 3.8 – Процес паяння модуля I2C до LCD дисплею

Для початку до дисплею потрібно підключити лінії живлення. У даній системі, до дисплею, було підключено живлення через окремі джампери «Тато-Мама» з мікроконтролеру. Пін SDA дисплею було підключено до аналогового роз'єму A4 мікроконтролеру, SCL – до A5. Лінії живлення, відповідно VCC і GND, з'єднано з контактами 5V та GND на Arduino Uno R3. Таким чином, дисплей отримує як живлення, так і сигнальні дані через інтерфейс I2C, що дозволяє значно зменшити кількість необхідних пінів та спростити з'єднання.

Після підключення дисплей готовий до ініціалізації програмними засобами в середовищі Arduino IDE. Підключення дисплею зображено на рисунку 3.9.

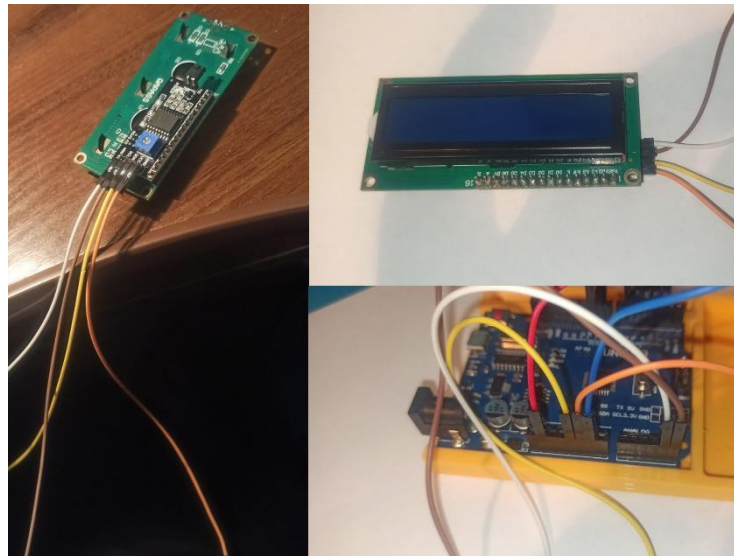


Рисунок 3.9 – Процес підключення LCD дисплею до мікроконтролеру Arduino Uno R3

У результаті виконаного монтажу апаратної частини системи було успішно реалізовано з'єднання всіх необхідних компонентів без використання пайки згідно попередньо розробленої схеми. Використання макетної плати, джамперів та інших з'єднувальних елементів дало змогу швидко зібрати працездатну систему. Усі основні модулі були надійно інтегровані з мікроконтролером Arduino Uno R3. Завдяки попередньому припаянню I2C-адаптера до дисплея, вдалося зменшити кількість підключень та зменшити використовуваний простір на макетній платі. Таким чином, система повністю готова до програмування в середовищі Arduino IDE.

### **3.2 Програмування роботи системи в середовищі Arduino**

Програмування системи автоматизованого вимірювання відстані здійснювалося в середовищі розробки Arduino IDE, яке зарекомендувало себе, як зручний інструмент для написання, компіляції та завантаження коду на

мікроконтролер. Основною метою програмування було реалізувати функціональні можливості системи згідно алгоритму описаному в пункті 2.4, а саме зчитування показників з ультразвукового датчика, обробка натискань кнопок, керування RGB-світлодіодом та вивід результатів вимірювання на дисплей.

Для реалізації даного функціоналу було використано стандартні бібліотеки Arduino, адаптована під програмування контролерів мова C++ та додаткові бібліотеки для роботи з компонентами.

Для роботи системи було використано дві бібліотеки: Wire (для роботи з I2C інтерфейсами) та LiquidCrystal\_I2C (для роботи з дисплеєм). Для того щоб підключити бібліотеки використовується конструкція `#include <library_name.h>`.

Далі необхідно оголосити змінні пінів, які використовуються у всіх компонентах системи. Для цього використовується конструкція `const int namePin = pin_number`. На рисунку 3.10 наведено всі оголошені піни для роботи системи.

---

```

4  const int trigPin = 9;           // Пін для сигналу TRIG ультразвукового датчика
5  const int echoPin = 10;        // Пін для сигналу ECHO ультразвукового датчика
6  const int buttonPin = 2;       // Кнопка для вмикання/вимикання пристрою
7  const int measureButtonPin = 4; // Кнопка для запуску вимірювання
8  const int redPin = 6;          // Пін керування червоним світлодіодом
9  const int greenPin = 5;        // Пін керування зеленим світлодіодом
10 const int bluePin = 3;         // Пін керування синім світлодіодом

```

---

Рисунок 3.10 – Оголошення пінів

Також, було створено ряд змінних, які будуть використовуватися у процесі обчислень системи згідно логіки роботи (рисунок 3.11).

---

```

27 bool deviceOn = false;          // Стан пристрою: увімкнений чи вимкнений
28 unsigned long lastDebounceTime = 0; // Час останнього зміни стану кнопки (для debounce)
29 unsigned long debounceDelay = 50;  // Затримка для debounce в мс
30 int lastButtonState = HIGH;        // Останній зчитаний стан кнопки
31 int buttonStateStable = HIGH;      // Стабільний стан кнопки після debounce

```

---

Рисунок 3.11 – Оголошення змінних, необхідних для роботи системи

Функція `setup()` є обов'язковою частиною кожного скретчу в Arduino. Вона виконується один раз під час запуску або перезавантаження мікроконтролера. В системі автоматизованого вимірювання відстані вона містить початкові налаштування усіх компонентів.

Кожен пін, до якого підключені компоненти повинен бути відповідно налаштований згідно призначення. Для цього використовується функція `pinMode()`. Вона визначає яким чином працює пін: як вхід (INPUT) чи як вихід (OUTPUT). На рисунку 3.12 показано, яким чином налаштовані відповідні піни у системі вимірювання відстані.

---

```

36   pinMode(trigPin, OUTPUT);           // TRIG як вихід
37   pinMode(echoPin, INPUT);           // ECHO як вхід
38   pinMode(buttonPin, INPUT);         // Кнопка ввімкнення як вхід
39   pinMode(measureButtonPin, INPUT);  // Кнопка вимірювання як вхід
40   pinMode(redPin, OUTPUT);           // Світлодіоди як вихід
41   pinMode(greenPin, OUTPUT);
42   pinMode(bluePin, OUTPUT);

```

---

Рисунок 3.12 – Налаштування пінів в функції `setup()`

Під час розробки чи тестування компонентів системи корисним інструментом є Serial Monitor, який доступний з інтерфейсу Arduino IDE. Він дозволяє виводити значення з датчиків, отримувати повідомлення, відповіді тощо.

В контексті розробки цієї системи Serial Monitor використовувався для виведення даних та інформації до підключення дисплею, тому у функції `setup()` є обов'язковим налаштування режиму його роботи.

Функція `loop()` в Arduino є безкінечним циклом, який виконується знову і знову після завершення `setup()`. Саме тут реалізується основна логіка роботи системи автоматизованого вимірювання відстані.

На початку циклу викликається функція `handleButtonWithDebounce()`, яка відповідає за обробку кнопки увімкнення/вимкнення системи. Вона реалізована

із використанням програмної затримки, яка забезпечує стабільну роботу кнопки і стримує від хибних спрацювань.

У функції `loop()` закладена умова роботи двох кнопок: щоб якщо пристрій включався запускалася функція `blinkRedLed()` та якщо пристрій увімкнений і натиснута кнопка вимірювання, то необхідно запустити функцію вимірювання `measureDistance()`. Вміст коду роботи функції `loop()` відображено на рисунку 3.13.

---

```

53 void loop() {
54     handleButtonWithDebounce();           // Обробка кнопки увімкнення з debounce
55     if (deviceOn) {
56         | blinkRedLed();                   // Миготіння червоного світлодіода при увімкненні
57         |
58     }
59     if (deviceOn && digitalRead(measureButtonPin) == HIGH) { // Якщо пристрій увімкнено і натиснута кнопка
60         | measureDistance();               // Запуск вимірювання відстані
61         |
62     }

```

---

Рисунок 3.13 – Код функції циклу `loop()`

Функція `handleButtonWithDebounce()` (рисунку 3.14) містить реалізацію реакції натиснення на кнопку увімкнення з урахуванням затримки. Вона моніторить зміну кнопки та вимірює час натиснення кнопки для того, щоб позбавити систему від помилкових спрацювань.

---

```

79 // Обробка кнопки з debounce для уникнення хибного спрацювання
80 void handleButtonWithDebounce() {
81     int reading = digitalRead(buttonPin);
82     if (reading != lastButtonState) {
83         | lastDebounceTime = millis();
84         |
85     }
86     if ((millis() - lastDebounceTime) > debounceDelay) {
87         | if (reading != buttonStateStable) {
88         |     | buttonStateStable = reading;
89         |     | Serial.println(buttonStateStable);

```

---

Рисунок 3.14 – Код функції циклу `handleButtonWithDebounce ()`

Також, всередині цієї функції є умова, яка дозволяє при натисненні на кнопку увімкнення/вимкнення, якщо пристрій знаходиться в увімкненому стані,

його вимкнути. Таким чином, однією кнопкою можна і вмикати і вимикати систему (рисунок 3.15).

---

```

89     if (buttonStateStable == HIGH) {
90         if (!deviceOn) {
91             turnOnDevice();      // Увімкнення пристрою
92         } else {
93             turnOffDevice();     // Вимкнення пристрою
94         }
95     }
96 }
97 }
98 lastButtonState = reading;
99 }

```

---

Рисунок 3.15 – Код вимкнення системи

Для увімкнення системи була створена спеціальна процедура `turnOnDevice()` (рисунок 3.16), яка викликається при натисненні на кнопку увімкнення/вимкнення при умові, якщо пристрій вимкнений. Ця процедура виводить інформацію про стан пристрою змінюючи його, вмикає дисплей, змінює сигнали світлодіода та виводить початкову анімацію і інформацію для користувача.

---

```

101 // Процедура увімкнення пристрою
102 void turnOnDevice() {
103     Serial.println("Turning ON the device");
104     deviceOn = true;
105     deviceTurnOnTime = millis();
106     lcd.backlight();
107     showLoadingScreen();           // Анімація завантаження
108     digitalWrite(redPin, LOW);
109     digitalWrite(greenPin, HIGH); // Ввімкнення зеленого світлодіода
110     digitalWrite(bluePin, LOW);
111     lcd.clear();
112     lcd.setCursor(0, 0);
113     lcd.print("Gotovo do vymiryuvannya");
114     lcd.setCursor(0, 1);
115     lcd.print("Natysnit' knopku");
116 }

```

---

Рисунок 3.16 – Код процедури `turnOnDevice()`

Також, була створена процедура вимкнення системи автоматизованого вимірювання відстані `turnOffDevice()`. Вона відповідає за зміну стану пристрою у змінній, яка за це відповідає та вивід відповідного повідомлення в `Serial Monitor`. Ще вона відповідає за очищення та вимкнення дисплею і вимкнення всіх кольорів RGB-світлодіоду. Повний код процедури `turnOffDevice()` відображено у попередньому коді лістингу.

Для безпосереднього вимірювання відстані було створено процедуру `measureDistance()`, яка викликається в циклі `loop()` після натискання кнопки запуску вимірювання. Її основним призначенням є ініціювання ультразвукового сигналу, вимірювання часу повернення відбитого імпульсу та обчислення середнього значення відстані на основі кількох вимірювань. Такий алгоритм роботи є стандартним для більшості ультразвукових датчиків вимірювання відстані.

Таким чином, у середовищі `Arduino IDE` було реалізовано повноцінну програмну логіку для автоматизованої системи вимірювання відстані. Код організовано у вигляді модульних функцій, кожна з яких виконує окрему задачу. Така структура дозволить легко змінювати, вдосконалювати код та розширювати систему в майбутньому. У ході програмування було виконано всі вимоги, поставлені до функціональних можливостей системи.

### **3.3 Візуалізація та виведення результатів вимірювання**

Одним із ключових елементів зручності та інформативності системи вимірювання в є засоби візуалізації результатів вимірювання та іншої інформації. Для цього в цій системі використовується LCD-дисплей із інтерфейсом I2C та бібліотеки для роботи з ним. Також, для візуалізації станів у системі є RGB-світлодіод.

LCD 1602 дисплей 16x2, з модулем I2C використовує для роботи бібліотеку `LiquidCrystal_I2C`. Після підключення дисплею його потрібно оголосити. В коді це робиться наступним чином: `LiquidCrystal_I2C lcd(0x27, 16,`

2). Це означає, що оголошується дисплей типу `Lcd`, з використанням бібліотеки `LiquidCrystal_I2C`, який доступний за адресою `0x27` та має розміри матриці виводу 16 на 2.

Для покращення візуальної привабливості інтерфейсу системи автоматизованого вимірювання відстані було створено спеціальні символи, які використовуються в стартовій анімації завантаження системи. Такі символи називаються кастомними символами і зберігаються в пам'яті дисплею. Оголошення цих символів відображено на рисунку 3.17.

---

```

14 byte progressBar[8] = { // Користувацький символ для прогрес-бара
15     B11111, B11111, B11111, B11111,
16     B11111, B11111, B11111, B11111
17 };
18 byte customChar1[8] = { // Користувацький символ (логотип ЛНТУ)
19     B10101, B10111, B11101, B00000,
20     B10001, B11011, B01110, B00100
21 };
22 byte customChar2[8] = { // Користувацький символ (серце)
23     B00000, B00000, B01010, B11111,
24     B11111, B01110, B00100, B00000
25 };

```

---

Рисунок 3.17 – Кастомні символи, які використовуються у створенні стартової анімації

В функції `void setup()` здійснюється налаштування дисплею та створення кастомних символів описаних вище. Для цього дисплей спочатку ініціалізується та підсвітка і створюються кастомні символи елементу прогрес-бару (`progressBar`), логотипу ЛНТУ (`customChar1`) та символу серця (`customChar2`) (рисунок 3.18).

---

```

44 lcd.init(); // Ініціалізація LCD
45 lcd.backlight(); // Вмикаємо підсвічування
46 lcd.createChar(0, progressBar); // Створення символу прогрес-бара
47 lcd.createChar(1, customChar1); // Створення символу 1
48 lcd.createChar(2, customChar2); // Створення символу 2
49 turnOffDevice(); // Початково пристрій вимкнений

```

---

Рисунок 3.18 – Кастомні символи, які використовуються у створенні стартової анімації

Для відображення кастомних символів було створено спеціальний макрос в програмі MS EXEL, який спочатку перетворює візуально оформлену область в нулі та одиниці, а потім генерує готовий код символу та копіює його в буфер обміну комп'ютера. Для відображення символів дисплей використовує матриці розміром 5 на 8 (рисунок 3.19).

1	0	1	0	1	0	0	0	0	0	1	1	1	1	1
1	0	1	1	1	0	0	0	0	0	1	1	1	1	1
1	1	1	0	1	0	1	0	1	0	1	1	1	1	1
0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1	1	1	1	1	1	1	1
1	1	0	1	1	0	1	1	1	0	1	1	1	1	1
0	1	1	1	0	0	0	1	0	0	1	1	1	1	1
0	0	1	0	0	0	0	0	0	0	1	1	1	1	1

Рисунок 3.19 – Матриці виводу кастомних символів в програмі MS EXEL

Під час процедури увімкнення пристрою `turnOnDevice()` вмикається підсвітка дисплею та початкова анімація запуску. За анімацію відповідає функція `showLoadingScreen()`. Анімація виконується 5 секунд і включає вивід логотипу, надпису «LNTU» та символу серця у першому рядку дисплею і прогрес-бар у другому (рисунок 3.20).

```

129 // Анімація завантаження на дисплеї
130 void showLoadingScreen() {
131     lcd.clear();
132     lcd.setCursor(5, 0);
133     lcd.write(1); // Вивід користувацького символу
134     lcd.setCursor(6, 0);
135     lcd.print("LNTU");
136     lcd.setCursor(10, 0);
137     lcd.write(2);
138     const int loadingTime = 5000;
139     const int steps = 16;
140     const int stepTime = loadingTime / steps;

```

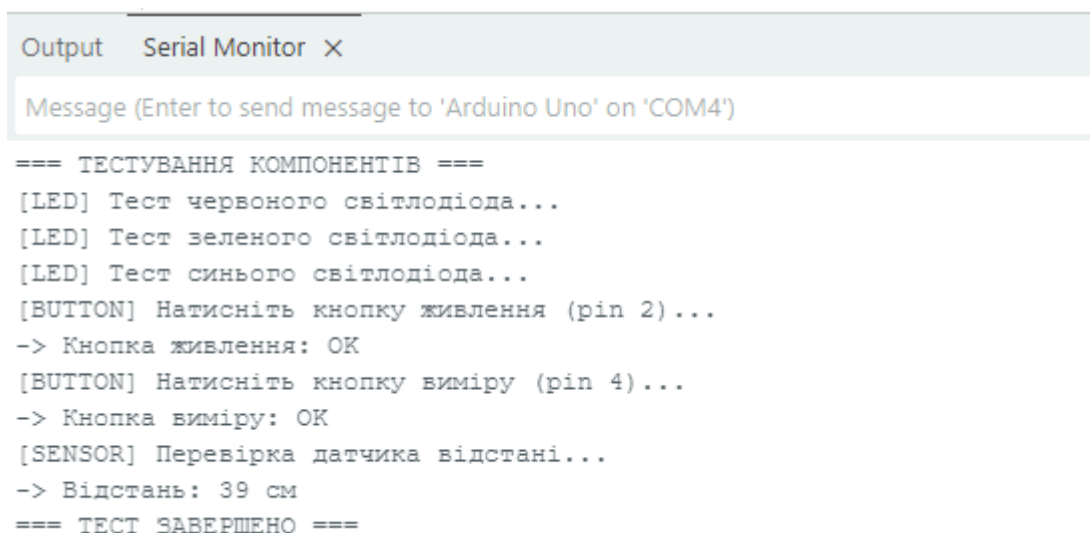
Рисунок 3.20 – Кастомні символи, які використовуються у створенні стартової анімації

Також, під час анімації подається світловий сигнал про те, що пристрій ще не готовий до роботи (червоний мигаючий сигнал). Сигнал змінюється на зелений одразу після завантаження анімації. В процесі вимірювання сигнал змінюється на зелений мигаючи. Під час вимірювання на дисплей виводиться надпис «Zachekayte». Результати вимірювання виводяться у функції `measureDistance()` та вказуються у сантиметрах.

Таким чином, було створено візуальний вивід інформації та індикацію станів пристрою, яке підвищить зручність та комфорт використання системи автоматизованого вимірювання відстані. Також, забезпечений вивід результатів вимірювання та повідомлень користувачу у зручному форматі.

### 3.4 Тестування роботи системи

Метою проведення тестування системи є перевірка правильності роботи всіх її компонентів, виявлення багів та перевірка точності і стабільності системи вимірювання (рисунок 3.21).



```
Output  Serial Monitor X
Message (Enter to send message to 'Arduino Uno' on 'COM4')
=== ТЕСТУВАННЯ КОМПОНЕНТІВ ===
[LED] Тест червоного світлодіода...
[LED] Тест зеленого світлодіода...
[LED] Тест синього світлодіода...
[BUTTON] Натисніть кнопку живлення (pin 2)...
-> Кнопка живлення: ОК
[BUTTON] Натисніть кнопку виміру (pin 4)...
-> Кнопка виміру: ОК
[SENSOR] Перевірка датчика відстані...
-> Відстань: 39 см
=== ТЕСТ ЗАВЕРШЕНО ===
```

Рисунок 3.21 – Результат тестування компонентів через Serial Monitor в середовищі Arduino IDE

В процесі тестування необхідно перевірити кожен з функціональних можливостей, яка була закладена в логіку роботи системи.

Тестування компонентів було проведено в середовищі Arduino IDE. Воно включало перевірку взаємодії компонента із зовнішніми чинниками впливу, такими як натиснення користувачем кнопок. Всі результати тестування виводилися у Serial Monitor.

Тестування компонентів пройшло успішно. В результаті тестування всі компоненти були підключені до системи правильно і доступні для взаємодії.

Перевірка системи автоматизованого вимірювання відстані в цілому включає замірювання розмірів фізичними пристроями вимірювання та системою і порівняння результатів між ними. Також, важливим принципом є перевірка стабільності результатів при багаторазовому запуску.

Етап ввімкнення системи автоматизованого вимірювання відстані та етап готовності до вимірювання відображено на рисунку 3.22 та рисунку 3.23 відповідно.

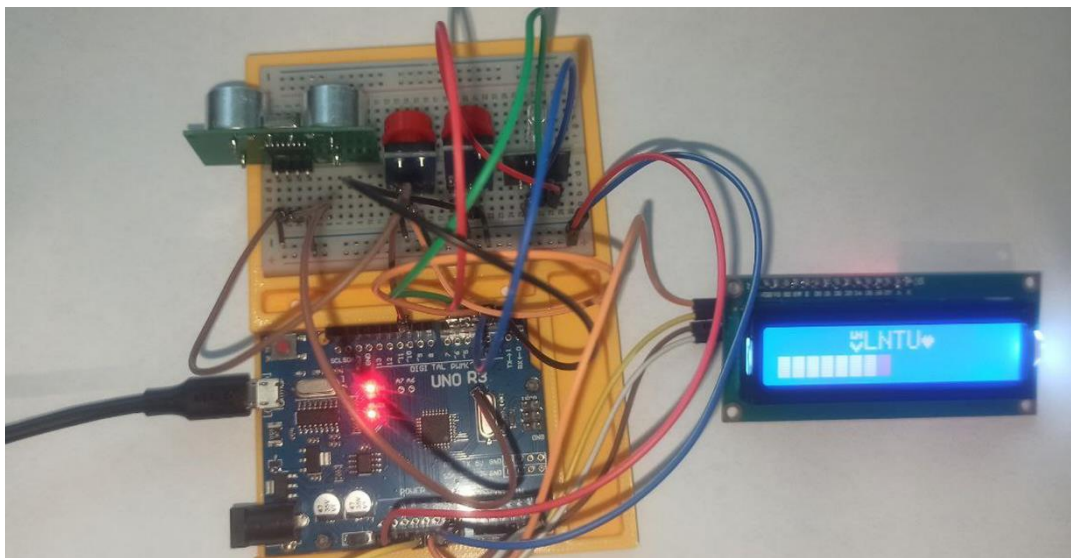


Рисунок 3.22 – Етап ввімкнення системи автоматизованого вимірювання відстані

Для початку було протестовано вивід інформації на дисплей та вивід сигналів на RGB-світлодіод. Після натискання на кнопку живлення пристрій

успішно виводив анімацію завантаження, що включала логотип, текст «LNTU» та прогрес-бар, а також миттєво вмикалася підсвітка дисплея.

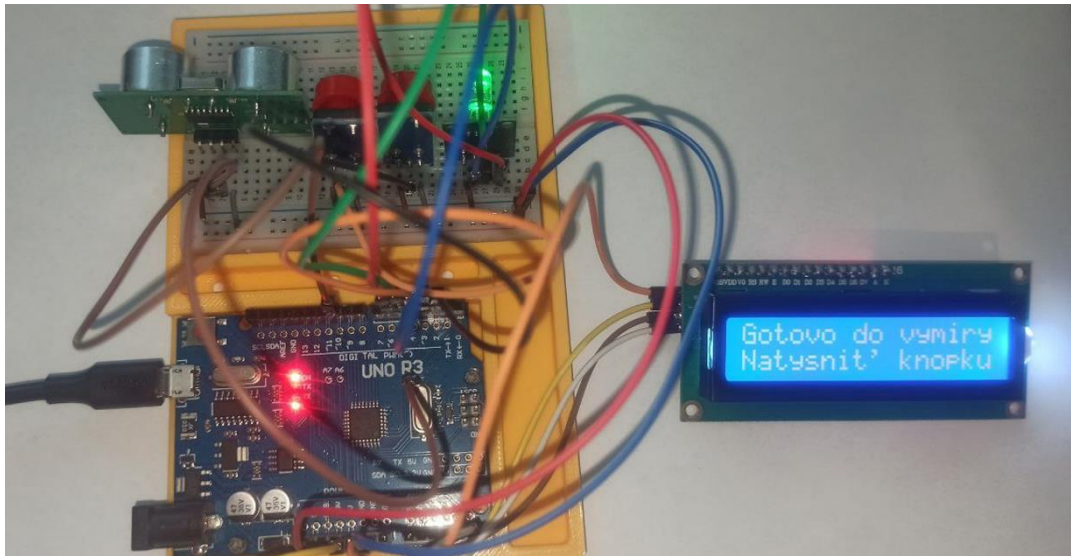


Рисунок 3.23 – Етап готовності системи до вимірювання

Паралельно на RGB-світлодіоді з'являвся червоний мигаючий сигнал. Це означає що система запускається і ще не готова до вимірювання. Після завершення анімації індикація змінювалася на зелений колір. Це означає, що система готова до початку виміру.

На рисунку 3.24 зображено реакція системи під час вимірювання.

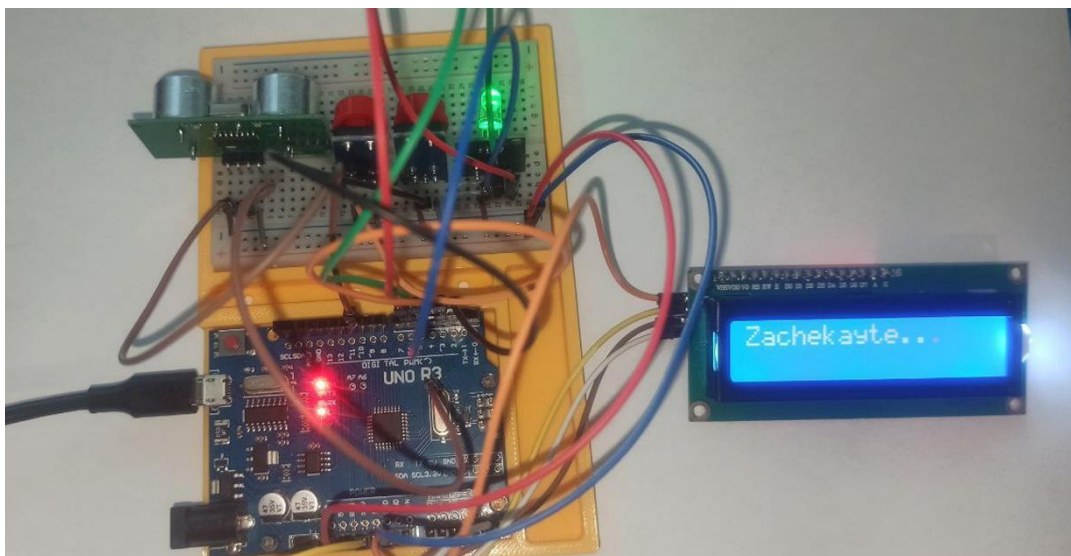


Рисунок 3.24 – Етап вимірювання відстані

При натисненні на кнопку виміру запускається етап вимірювання. В цей момент на екран виводиться надпис «Zachekayte» та виконується процес вимірювання. Також, зелений світлодіод починає мигати. Це означає, що система вимірює відстань.

Останнім етапом візуальних можливостей системи є вивід результатів. В цей момент на екран виводиться надпис «Vidstan':» та відстань в сантиметрах. Цей етап відображено на рисунку 3.25. Після виводу світлодіод знову показує зелений постійний сигнал.

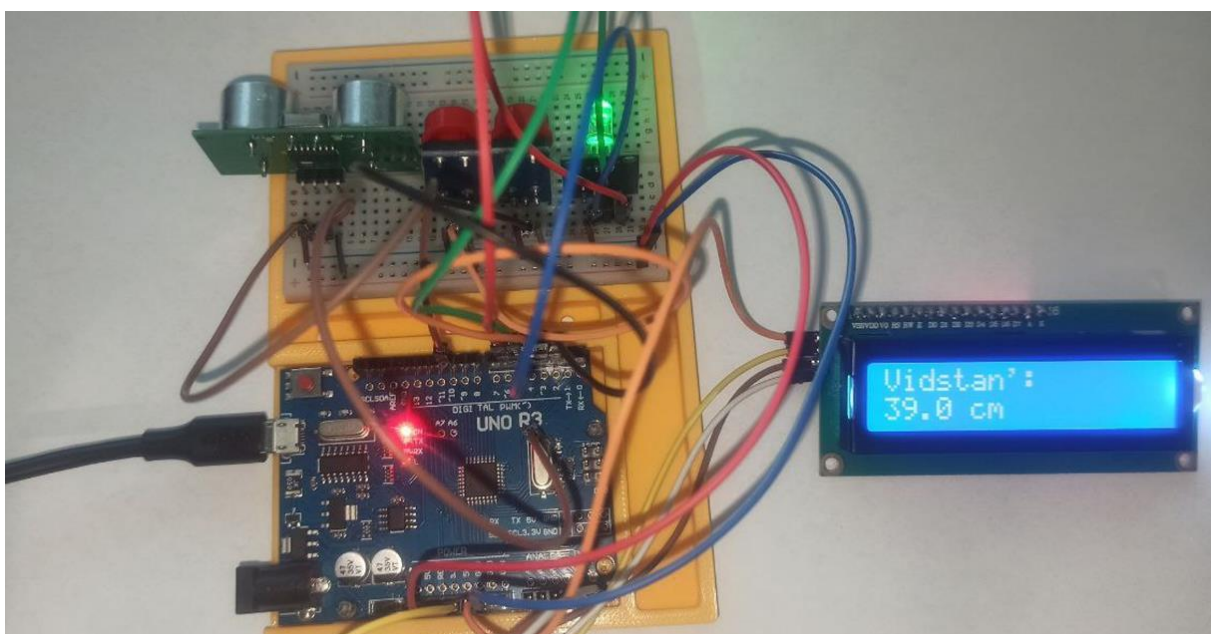


Рисунок 3.25 – Етап виводу результатів вимірювання

Далі було проведено ряд вимірювань із статичного положення системи для перевірки стабільності системи. Система вимірює з точністю до сантиметрів. Усі 5 замірів дали однакове значення (рисунок 3.26). Також, було протестовано вимірювання відстані на більших дистанціях від 10 см до 100 см з інтервалом у 10 см. Для перевірки точності використовувалась вимірювальна рулетка як зразок. На кожній з обраних дистанцій проводилося по 5 вимірювань, після чого визначалося середнє значення та порівнювалося з реальними даними. В переважній більшості випадків система показувала

відстань із допустимою похибкою в межах 1-2 см, що відповідає технічним характеристикам ультразвукового датчика.



Рисунок 3.26 – Тестування системи на стабільність вимірів

Під час експерименту також перевірялося, як змінюється точність у випадку зміни кута розміщення об'єкта перед датчиком. При нахилі об'єкта під кутом до  $15^\circ$  система продовжувала стабільно визначати відстань із незначним відхиленням. Проте при більших кутах результати могли змінюватися більш суттєво через фізичні властивості відбиття ультразвукових хвиль.

Ще один важливий аспект тестування полягає в перевірці поведінки системи при зміні типу об'єкта (матеріалу, кольору, форми). У ході тестів використовувалися різні предмети: дерев'яна дошка, пластикова кришка, книжка з глянцевою обкладинкою. Усі об'єкти визначались без проблем, хоча в окремих випадках (наприклад, при дуже гладких поверхнях) часом спостерігалася незначна нестабільність значень.

На завершення було здійснено тестування всієї системи в умовах моделювання типової ситуації: користувач вмикає систему, чекає на готовність, натискає кнопку і бачить результат. Тестування показало, що всі етапи

проходять логічно, а реакція пристрою на дії користувача швидка та передбачувана.

Таким чином, результати тестування підтвердили правильність роботи апаратної частини та логіки програмного забезпечення, що забезпечує надійну роботу системи автоматизованого вимірювання відстані в заданих умовах.

### **3.5 Аналіз отриманих результатів**

Після завершення розробки та тестування системи автоматизованого вимірювання відстані можна зробити кілька важливих висновків щодо її роботи, точності та зручності використання.

Перш за все, система показала себе надійною та стабільною під час усіх проведених перевірок. Вона коректно реагує на дії користувача, вмикається, ініціалізується з виводом анімації, чітко подає сигнали готовності за допомогою RGB-світлодіода, і виконує вимірювання після натискання кнопки. Це означає, що логіка взаємодії всіх компонентів реалізована правильно.

Що стосується точності вимірювання, то результати виявились доволі точними, з похибкою в межах 1-2 см. Це дуже хороший показник для недорогого ультразвукового датчика в умовах звичайного приміщення. Також приємно відзначити, що система не просто «міряє», а й робить це повторювано – результати кількох вимірів поспіль на одній і тій самій відстані були практично однаковими. Це свідчить про хорошу стабільність. Додатковим плюсом є візуальна частина – інтерфейс із кастомними символами та анімацією виглядає приємно, а підсвітка дисплея та світлові сигнали додають інтуїтивності у сприйнятті роботи системи. Це важливо, бо навіть проста система повинна бути зручною у використанні.

Окремо варто згадати ситуації, в яких точність могла змінюватися – наприклад, коли об'єкт мав гладку або нахилену поверхню. Це природно, враховуючи фізичні властивості ультразвукових хвиль. Але навіть у таких випадках відхилення були незначними і не впливали критично на результат.

Таким чином, система в цілому успішно виконує свої функції, відповідає поставленим завданням і може бути використана як для навчальних, так і для практичних цілей. Отримані результати підтверджують, що обрана апаратна база та програмна логіка є оптимальними для реалізації подібного проєкту.

## ВИСНОВКИ

У результаті виконання кваліфікаційної роботи досягнуто мети та реалізовано поставлені завдання.

Аналітично досліджено питання систем автоматизованих вимірювань та визначено, що принципи роботи систем автоматизованого вимірювання відстані базуються на створенні взаємодії між компонентами та програмованим алгоритмом роботи цих компонентів на основі технології надсилання сигналів та отримання зворотної відповіді.

Реалізовано систему автоматизованого вимірювання відстані на базі мікроконтролера Arduino та ультразвукового датчика US-02. У якості додаткових компонентів було обрано наступні апаратні засоби: LCD 1602 дисплей 16x2 (синій), модуль кнопки з ковпаком у кількості 2 штук та RGB-світлодіод KY-016.

Розроблено алгоритм обробки даних з ультразвукового сенсора для точного визначення відстані в режимі реального часу, який описує логіку взаємодії між користувачем, ультразвуковим датчиком контролером Arduino та іншими елементами системи. Завдяки використанню поетапної логіки було спроектовано, яким чином система буде реагувати на дії користувача, як буде виконуватися ввімкнення/вимкнення системи, запуск процесу вимірювання, індикація станів системи та вивід результатів на дисплей.

Досліджено точність, стабільність та надійність вимірювань ультразвукового датчика US-025 у різних умовах експлуатації – здійснено тестування всієї системи в умовах моделювання типової ситуації. Тестування показало, що всі етапи проходять логічно, а реакція пристрою на дії користувача швидка та передбачувана.

Візуалізовано результати вимірювань за допомогою дисплея та створено візуальний вивід інформації, індикацію станів пристрою, яке дозволяє підвищити зручність та комфорт використання системи автоматизованого

вимірювання відстані. Забезпечено вивід результатів вимірювання та повідомлень у зручному форматі.

Спроековано прототип пристрою для інтеграції в охоронні системи. Практично перевірено роботу системи на гладких та похилих поверхнях та зафіксовано відхилення точності вимірювань в межах 5%, що свідчить про адекватність роботи.

На основі вище зазначеного, можна зробити висновок, що розроблена мікроконтролерна система вимірювання відстані є надійною, стабільною, економічно вигідною та точною. Ідеї, що покладені в основу розробки, а також і усю систему, можна використовувати і у системах безпеки та сигналізації, для безконтактного вимірювання в складних умовах агресивних, брудних середовищ, для контролю рівня заповнення ємностей, а також в навчальних дисциплінах за напрямком Інтернету речей.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Іщук О. Особливості проектування системи точного вимірювання відстані. «Сучасні інформаційні технології та системи в управлінні» : Матеріали VI Міжнар. науково-практ. конф. молодих вчен., аспірантів і студентів, м. Київ, 10-11 квіт. 2025 р. / Наук. керівник Н. Христинець. Київ, 2025. С. 12–14.
2. 4 Types of Distance Sensors & How to Choose the Right One | KEYENCE America. *KEYENCE CORPORATION OF AMERICA*. URL: <https://www.keyence.com/products/measure/resources/measurement-sensors-resources/4-types-of-distance-sensors-and-how-to-choose-the-right-one.jsp> (date of access: 11.05.2025).
3. Ультразвуковий датчик відстані US-025. *Arduino в Україні*. URL: <https://arduino.ua/prod3710-yltrazvykovoi-datchik-rasstoyaniya-us-025> (дата звернення: 11.05.2025).
4. Ультразвуковий датчик відстані HC-SR04 купити в Києві та Україні. *Arduino в Україні*. URL: <https://arduino.ua/prod182-yltrazvykovoi-datchik-rasstoyaniya-hc-sr04> (дата звернення: 11.05.2025).
5. Ультразвуковий датчик відстані. *Мій проект*. URL: <https://myproject.com.ua/hy-srf05-ultrazvukovij-datchik-vidstani-ua.html> (дата звернення: 11.05.2025).
6. IOT-based transformer monitoring health system. *International Research Journal of Modernization in Engineering Technology and Science*. 2023. Vol. 1, no. 5. P. 124-137
7. Adhau D. S. Empowering Homes with Esp-32. *International Scientific Journal of Engineering and Management*. 2024. Vol. 03, no. 04. P. 1-9.
8. Baumann P. Temperature Sensors. *Selected Sensor Circuits*. Wiesbaden, 2022. P. 1-22. URL: [https://doi.org/10.1007/978-3-658-38212-4\\_1](https://doi.org/10.1007/978-3-658-38212-4_1) (date of access: 11.05.2025).

9. Rajashekar M. IoT Based Transformer Monitoring System using ESP-32. *International Journal for Research in Applied Science and Engineering Technology*. 2024. Vol. 12, no. 5. P. 1-23

10. Бордюк М., Шевчук Т., Бордюк В. Ультразвукові дослідження в лабораторному практикумі медичної та біологічної фізики. *New pedagogical thought*. 2023. Т. 114, № 2. С. 89–96. URL: <https://doi.org/10.37026/2520-6427-2023-114-2-89-96> (дата звернення: 25.02.2025).

11. Exploiting Arduino Features to Develop Programming Competencies / M. Tupac-Yupanqui et al. *IEEE Access*. 2022. Vol. 10. P. 20602–20615. URL: <https://doi.org/10.1109/access.2022.3150101> (date of access: 25.02.2025).