

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ «ФАХОВИЙ КОЛЕДЖ**  
**ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**  
**НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**

**КУРСОВИЙ ПРОЄКТ**

**на тему:**

«Виготовлення охоронного пристрою на платі Arduino з використанням датчиків HC-SR501 і HC-SR04 та виводом інформації на LCD дисплей»

Виконав студент групи ІП-42

Олександр СЕНДАК

Керівник проєкту:

Остап ЮНАК

Курсовий проєкт перевірений

і допущений до захисту

“    ”    \_\_\_\_\_ 2025 р.

Курсовий проєкт при захисті оцінений

\_\_\_\_\_

Львів 2025

**ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ  
«ФАХОВИЙ КОЛЕДЖ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**

ЗАТВЕРДЖЕНО

на засіданні циклової комісії

«Фундаментальної підготовки»

Протокол № \_\_\_\_ від \_\_\_\_\_ 2025 р.  
Голова комісії

\_\_\_\_\_ Богдан ПЕЛЕЩАК

**ЗАВДАННЯ**

на курсовий проєкт

***Сендаку Олександр Віталійовичу***

(прізвище, ім'я та по батькові)

з навчальної дисципліни: ПРОГРАМНО-АПАРАТНІ ЗАСОБИ INTERNET-РЕЧЕЙ

Студент групи: **ІІІ-42**

**1. Тема проєкту:** Виготовлення охоронного пристрою на платі Arduino з використанням датчиків HC-SR501 I HC-SR04 та виводом інформації на LCD дисплей.

**2. Дата видачі завдання:** \_\_\_\_\_ ”14” жовтня 2025 р

**3. Термін здачі курсового проєкту:** \_\_\_\_\_ ”09” грудня 2025 р.

**4. Вихідні дані до проєкту:**

4.1. Провести аналіз предметної галузі і вимог (огляд аналогів, функціональні й нефункціональні вимоги).

4.2. Скласти технічне завдання (ТЗ) з переліком функцій, інтерфейсів і обмежень(див. ДОДАТОК А).

4.3 Підібрати WOM (Bill of Materials) для виконання проєкту (див. ДОДАТОК Б).

4.4 Розробити таблицю з'єднань пінів (Pin Connection Table) використаних компонентів для реалізації пристрою (див. ДОДАТОК В).

4.5 Реалізувати прошивку на Arduino (коментарі в коді, структура проекту, використані бібліотеки).

4.6. Зробити на монтажній платі безпечної монтажу прототип розробленого пристрою на базі Arduino.

4.7. Провести налаштування, калібрування і тестування.

4.8. Підготувати користувацьку інструкцію та технічний звіт (див. ДОДАТОК Г).

## **5 Перелік обов’язкових демонстраційних креслень:**

5.1 Принципова електрична схема підключення

## **6. Склад розрахунково – пояснювальної записки (перелік питань до розробки):**

### **ВСТУП**

1 ОГЛЯД ЛІТЕРАТУРИ/АНАЛОГІВ

2 ТЕХНІЧНЕ ЗАВДАННЯ

3 АПАРАТНА ЧАСТИНА

4 ПРОГРАМНА ЧАСТИНА

5 ТЕСТУВАННЯ ТА РЕЗУЛЬТАТИ

ВИСНОВКИ

ПЕРЕЛІК ПОСИЛАНЬ

### **Календарний план**

| Назва етапів                | Термін виконання | Примітка |
|-----------------------------|------------------|----------|
| Вступ                       |                  |          |
| 1 ОГЛЯД ЛІТЕРАТУРИ/АНАЛОГІВ |                  |          |
| 2 ТЕХНІЧНЕ ЗАВДАННЯ         |                  |          |
| 3 АПАРАТНА ЧАСТИНА          |                  |          |
| 4 ПРОГРАМНА ЧАСТИНА         |                  |          |
| 5 ТЕСТУВАННЯ ТА РЕЗУЛЬТАТИ  |                  |          |
| Висновки                    |                  |          |
| Перелік посилань            |                  |          |

**Студент**

\_\_\_\_\_  
( підпис )

**Олександр СЕНДАК**

\_\_\_\_\_  
( імя та прізвище )

**Керівник проекту**

\_\_\_\_\_  
( підпис )

**Остап ЮНАК**

\_\_\_\_\_  
( імя та прізвище )

## ЗМІСТ

|                                                           |    |
|-----------------------------------------------------------|----|
| ВСТУП.....                                                | 3  |
| 1 ОГЛЯД ЛІТЕРАТУРИ/АНАЛОГІВ.....                          | 5  |
| 1.1 Мова програмування C++ та середовище Arduino IDE..... | 5  |
| 1.2 Апаратно-програмна платформа Arduino.....             | 6  |
| 1.3 Принципи роботи використаних сенсорів.....            | 6  |
| 2 ТЕХНІЧНЕ ЗАВДАННЯ.....                                  | 9  |
| 2.1 Назва роботи.....                                     | 9  |
| 2.2 Мета роботи.....                                      | 9  |
| 2.3 Завдання.....                                         | 9  |
| 2.4 Технічні вимоги.....                                  | 10 |
| 2.5 Очікуваний результат .....                            | 10 |
| 3 АПАРАТНА ЧАСТИНА.....                                   | 11 |
| 3.1 Підбір ВОВ.....                                       | 11 |
| 4 ПРОГРАМНА ЧАСТИНА.....                                  | 13 |
| 4.1 Опис алгоритму роботи.....                            | 13 |
| 4.2 Код програми.....                                     | 14 |
| 5 ТЕСТУВАННЯ ТА РЕЗУЛЬТАТИ.....                           | 18 |
| 5.1 Підготовка до тестування.....                         | 18 |
| 5.2 Процес тестування.....                                | 18 |
| 5.3 Результати.....                                       | 19 |
| 5.4 Висновки.....                                         | 19 |
| ВИСНОВКИ.....                                             | 20 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....                           | 22 |
| ДОДАТОК 1.....                                            | 23 |

## ВСТУП

### **Мета роботи:**

Розробити згідно запропонованої теми апаратно-програмний пристрій на базі Arduino, що реалізує заданий набір функцій та підготувати звіт з техніко-конструкторською документацією та інструкцією з експлуатації.

### **Актуальність:**

Актуальність цієї курсової роботи обґрунтовується високим попитом на інтеграцію апаратного та програмного забезпечення у сучасних інженерних рішеннях, а також зростаючою роллю мікроконтролерних платформ у навчальному процесі та індустрії.

### **● Технологічна та Інженерна Актуальність**

- **Поширеність IoT (Internet of Things):** Сучасні системи автоматизації, моніторингу та управління критично залежать від компактних, економічних та енергоефективних пристроїв. Платформа Arduino є де-факто стандартом для швидкого прототипування та створення пристроїв, що взаємодіють з фізичним світом, що відповідає глобальному тренду розвитку Інтернету речей;
- **Демократизація розробки:** Arduino, як відкрита (Open-Source) апаратно-програмна платформа, значно знижує поріг входу для інженерів-початківців та сприяє створенню інновацій. Це дозволяє зосередитися на логіці проєкту та функціональності, а не на низькорівневому програмуванні та складному апаратному дизайні;
- **Міждисциплінарний характер:** Розробка апаратно-програмного пристрою вимагає інтеграції знань з електроніки, програмування мікроконтролерів (C/C++), теорії керування та проєктування інтерфейсів. Це забезпечує набуття комплексних інженерних навичок, які є критично важливими для працевлаштування у сферах робототехніки, автоматизації та вбудованих систем;

### **● Прикладна та Практична Актуальність**

- **Вирішення конкретної прикладної задачі:** Робота не обмежується теоретичним аналізом, а передбачає створення працюючого фізичного прототипу (моніторинг параметрів, автоматизоване управління, збір даних). Це безпосередньо демонструє здатність автора перетворювати теоретичні знання на практичне інженерне рішення;
- **Економічна ефективність:** Використання Arduino дозволяє створити функціональний пристрій з мінімальними витратами на компоненти та часом розробки. Це важливий аспект для інноваційних стартапів та освітніх проєктів;
- **Можливість подальшого розвитку:** Розроблений прототип слугує надійною базою для подальшого масштабування, модернізації та переходу на промислові мікроконтролери, що забезпечує наукову перспективу для дипломної роботи чи комерціалізації.

Таким чином, розробка апаратно-програмного пристрою на базі Arduino є своєчасним, економічно обґрунтованим і високопрактичним дослідженням, що поєднує вивчення сучасних апаратних засобів та формування ключових інженерних компетенцій.

## 1 ОГЛЯД ЛІТЕРАТУРИ/АНАЛОГІВ

### 1.1 Апаратно-програмна платформа Arduino

Arduino — це відкрита (open-source) платформа для створення прототипів електронних пристроїв, яка базується на простому у використанні апаратному та програмному забезпеченні. Платформа складається з фізичної друкованої плати (мікроконтролера) та середовища розробки (IDE) для написання коду.

Рис. 1.1 Вигляд плати Arduino з мікроконтролером ATmega328P2

Проект Arduino був започаткований у 2005 році в Інституті проєктування взаємодії (Interaction Design Institute Ivrea) в місті Івреа, Італія. Його засновниками стали Массімо Банці (Massimo Banzi), Девід Куартільєс, Том Іго, Джанлука Мартіно та Девід Мелліс. Основною метою створення платформи було забезпечення студентів дешевим та простим інструментом для створення інтерактивних проєктів, який би не вимагав глибоких знань у схемотехніці.

Для даної курсової роботи обрано плату Arduino Uno R3, яка побудована на базі мікроконтролера ATmega328P2. Її популярність зумовлена наступними факторами:

- **Доступність:** Низька вартість компонентів;
- **Кросплатформеність:** Середовище розробки працює на Windows, macOS та Linux;
- **Простота:** Зручний інтерфейс програмування через USB без необхідності використання зовнішніх програматорів
- **Відкрита архітектура:** Схеми плат знаходяться у вільному доступі, що дозволяє легко інтегрувати різноманітні датчики та модулі;

### 1.2 Мова програмування C++ та середовище Arduino IDE

Програмування мікроконтролера здійснюється мовою C++. Середовище розробки Arduino IDE використовує спеціальний спрощений діалект (фреймворк) під назвою **Wiring**.

**Фреймворк** — це набір готових інструментів, бібліотек, шаблонів та правил, який надає структуру для створення програмного забезпечення, спрощуючи та прискорюючи процес розробки. Він допомагає розробникам уникнути написання стандартного коду з нуля, дозволяючи зосередитися на унікальній бізнес-логіці продукту. Фреймворки використовуються для різних типів проєктів, включаючи веб, мобільні та десктопні додатки.

C++ — це універсальна мова програмування високого рівня, яка підтримує об'єктно-орієнтовану парадигму. В контексті розробки під Arduino, C++ дозволяє ефективно керувати пам'яттю мікроконтролера та працювати з апаратними регістрами, зберігаючи при цьому зрозумілий синтаксис.

Програма для Arduino називається «скетчем» (sketch) і має дві обов'язкові функції:

- `setup()` — виконується один раз при запуску для налаштування конфігурації пінів;
- `loop()` — виконується циклічно, реалізуючи основний алгоритм роботи пристрою.

### 1.3 Принципи роботи використаних сенсорів

**Датчик руху HC-SR501 (PIR Sensor).** Принцип дії пасивного інфрачервоного датчика (Passive Infrared Sensor) базується на виявленні змін інфрачервоного (теплового) випромінювання. Кожен об'єкт із температурою вище абсолютного нуля випромінює тепло. Піроелектричний елемент всередині датчика реагує на зміну інтенсивності цього випромінювання при переміщенні теплового об'єкта (людини) через сектори лінзи Френеля, генеруючи електричний сигнал тривоги.



Рис 1.3.1 Вигляд датчику руху HC-SR501

**Ультразвуковий датчик відстані HC-SR04.** Робота датчика базується на принципі ехолокації (SONAR). Пристрій складається з двох п'єзоелементів: випромінювача та приймача.

1. Випромінювач генерує пачку ультразвукових імпульсів частотою 40 кГц;
2. Звукова хвиля відбивається від перешкоди та повертається назад до приймача;
3. Мікроконтролер вимірює час проходження сигналу і розраховує відстань.

Рис 1.3.2 Вигляд датчику HC-SR04

**Рідкокристалічний дисплей (LCD 1602) з I2C.** Для візуалізації даних використано символний дисплей на рідких кристалах. Особливістю даної реалізації є використання адаптера I2C (Inter-Integrated Circuit). Цей послідовний протокол дозволяє передавати дані лише по двох сигнальних лініях (SDA та SCL), що значно спрощує схемотехніку пристрою порівняно з паралельним підключенням.

Рис 1.3.2 Вигляд рідкокристалічного дисплею (LCD 1602) з I2C

## **2 ТЕХНІЧНЕ ЗАВДАННЯ**

### **2.1 Назва роботи**

Виготовлення охоронного пристрою на платі Arduino з використанням датчиків HC-SR501 і HC-SR04 та виводом інформації на LCD дисплей.

### **2.2 Мета роботи**

Виготовити охоронний пристрій на платі Arduino з використанням датчиків HC-SR501 і HC-SR04 та виводом інформації на LCD дисплей.

### **2.3 Завдання**

- Ознайомитись з інструкцією до практичної роботи;
- Ознайомитись із принциповою електричною схемою охоронний пристрою на Arduino Uno;
- Зібрати на монтажній платі цифрову схему охоронного пристрою згідно електричної принципової схеми
- Написати скетч роботи пристрою на Arduino Uno;

### **2.4 Технічні вимоги**

- 5 В постійного струму;
- Arduino Uno R3;
- Breadboard (безпайкова макетна плата);
- LCD дисплей 1602 з I2C модулем;
- Світлодіоди: Зелений, Червоний;
- HC-SR501;
- HC-SR04;
- Резистори 220 Ом;
- П'єзозумер — для звукового сигналу;
- З'єднувальні провідники типу «тато-тато» та «тато-мама».

### **2.5 Очікуваний результат**

Алгоритм роботи охоронної системи базується на безперервному моніторингу зони відповідальності за допомогою пасивного інфрачервоного датчика руху (HC-SR501). У черговому режимі пристрій перебуває в стані спокою: активний зелений світлодіод, а на LCD-дисплеї відображається повідомлення про безпеку. При фіксації теплового випромінювання від рухомого об'єкта мікроконтролер миттєво активує режим «Тривога»: вмикається червоний світлодіод та звуковий оповіщувач, а ультразвуковий сенсор (HC-SR04) вимірює точну відстань до об'єкта, виводячи цю інформацію на екран у реальному часі. Після припинення руху система автоматично повертається у режим очікування.

### 3 АПАРАТНА ЧАСТИНА

#### 3.1 Підбір BOM

BOM (скорочення від Bill of Materials)- це список матеріалів і компонентів, потрібних для складання пристрою або виготовлення виробу, який може представлений як таблиця всіх деталей, які потрібно для проєкту

Таблиця 3.1.1 Перелік необхідних компонентів

| №  | Назва компонента      | Кількість | Позначення на схемі | Характеристики                  | Примітка                                 |
|----|-----------------------|-----------|---------------------|---------------------------------|------------------------------------------|
| 1  | Arduino Uno           | 1         | U1                  | Мікроконтролер ATmega328P       | Основна плата керування                  |
| 2  | Плата breadboard      | 1         | -                   | 830 точок, безпайкова           | Безпайна плата для створення конструкції |
| 3  | HC-SR501              | 1         | PIR1                | PIR Motion Sensor, 5 В          | Виявлення руху об'єктів (тепла)          |
| 4  | HC-SR04               | 1         | DIST1               | Ultrasonic Sensor, 2–400 см     | Вимірювання відстані до об'єкта          |
| 5  | LCD 1602 (I2C)        | 1         | U2                  | 16x2 символів, I2C інтерфейс    | Відображення статусу системи             |
| 6  | Світлодіод (Червоний) | 1         | D1_RED              | 5 мм, 2 В, 20 мА                | Індикація тривоги                        |
| 7  | Світлодіод (Зелений)  | 1         | D1_GREEN            | 5 мм, 2 В, 20 мА                | Індикація режиму "Охорона"               |
| 8  | П'єзозумер            | 1         | SP1                 | Активний, 5 В                   | Звукове оповіщення                       |
| 9  | Резистори             | 2         | R1, R2              | 220 Ом, 0.25 Вт                 | Обмеження струму LED                     |
| 10 | Провідники            | 20        | -                   | Dupont (Male-Male, Male-Female) | Електричні з'єднання                     |

Таблиця 3.1.2 Таблиця з'єднань пінів

| № | Компонент            | Вивід компонента | З'єднано з піном Arduino | Примітка / Функція    |
|---|----------------------|------------------|--------------------------|-----------------------|
| 1 | HC-SR04 (Ultrasonic) | VCC              | 5V                       | Живлення              |
|   |                      | Trig             | Digital Pin 9            | Сигнал запуску        |
|   |                      | Echo             | Digital Pin 10           | Прийом сигналу        |
|   |                      | GND              | GND                      | Земля                 |
| 2 | HC-SR501 (PIR)       | VCC              | 5V                       | Живлення              |
|   |                      | OUT              | Digital Pin 7            | Сигнал тривоги        |
|   |                      | GND              | GND                      | Земля                 |
| 3 | LCD 1602 (I2C)       | GND              | GND                      | Земля                 |
|   |                      | VCC              | 5V                       | Живлення              |
|   |                      | SDA              | Pin A4 (або SDA)         | Шина даних I2C        |
|   |                      | SCL              | Pin A5 (або SCL)         | Шина тактування I2C   |
| 4 | Світлодіод (Зелений) | Анод (+)         | Digital Pin 4            | Через резистор 220 Ом |
|   |                      | Катод (-)        | GND                      |                       |
| 5 | Світлодіод (Зелений) | Анод (+)         | Digital Pin 5            | Через резистор 220 Ом |
|   |                      | Катод (-)        | GND                      |                       |
| 4 | П'єзозумер           | Плюс (+)         | Digital Pin 6            | -                     |
|   |                      | Мінус (-)        | GND                      |                       |

## 4 ПРОГРАМНА ЧАСТИНА

Розробка програмного забезпечення для мікроконтролера здійснювалася в інтегрованому середовищі розробки Arduino IDE. Мова програмування — C++ з використанням стандартних бібліотек Wiring та спеціалізованих бібліотек для роботи з периферією (Wire.h, LiquidCrystal\_I2C.h).

Алгоритм роботи пристрою полягає у безперервному моніторингу простору за допомогою інфрачервоного датчика руху та вимірюванні відстані до об'єкта при виникненні тривоги, з подальшим виводом інформації на LCD дисплей та звуковим оповіщенням. Програма структурно поділена на три основні блоки: підключення бібліотек та оголошення констант, налаштування портів (ініціалізація) та основний цикл виконання.

### 4.1 Опис алгоритму роботи

- **Підключення бібліотек та оголошення змінних:** Для роботи з дисплеєм по шині I2C підключено бібліотеку LiquidCrystal\_I2C. Для зручності читання коду та швидкого налаштування схеми, номери пінів призначені іменованим константам:
  - Датчики: pinTrig (9), pinEcho (10), pinPIR (7).
  - Індикація: pinLedGreen (4), pinLedRed (5), pinBuzzer (6). Також оголошено глобальні змінні для збереження часу відгуку ехо-сигналу (duration), обчисленої відстані (distance) та поточного стану датчика руху (pirState).
- **Ініціалізація:** При подачі живлення на мікроконтролер виконується функція налаштування:
  - Ініціалізується LCD дисплей та вмикається його підсвітка.
  - Порти для керування світлодіодами, зумером та тригером ультразвукового датчика (Trig) налаштовуються як виходи (OUTPUT).

- Порти для зчитування сигналів з PIR-датчика та ехо-сигналу (Echo) налаштовуються як входи (INPUT).
- На дисплей виводиться вітальне повідомлення про запуск системи.
- **Основний цикл:** Програма виконує нескінченний цикл, що реалізує логіку охоронної системи:
  - **Зчитування даних:** Мікроконтролер опитує цифровий пін датчика HC-SR501. Одночасно генерується імпульс тривалістю 10 мкс на пін Trig, і за допомогою функції pulseIn вимірюється час повернення сигналу на пін Echo. Цей час конвертується у відстань (см).
  - **Обробка логіки:**
    - **Режим «Тривога»:** Якщо датчик руху видає високий рівень (HIGH), програма вмикає червоний світлодіод, генерує звуковий сигнал (функція tone), а на дисплей виводить повідомлення "ALARM!" та поточну відстань до об'єкта.
    - **Режим «Охорона»:** Якщо руху не виявлено (LOW), активується зелений світлодіод, червоний світлодіод і зумер вимикаються, а на дисплеї відображається статус "SAFE".
  - Цикл повторюється з невеликою затримкою для стабілізації показань.

## 4.2 Код програми

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

const int pinTrig = 9;
const int pinEcho = 10;
const int pinPIR = 7;
const int pinLedGreen = 4;
const int pinLedRed = 5;
const int pinBuzzer = 6;

long duration;
int distance;
int pirState = LOW;

void setup() {
  pinMode(pinTrig, OUTPUT);
  pinMode(pinEcho, INPUT);
  pinMode(pinPIR, INPUT);
  pinMode(pinLedGreen, OUTPUT);
  pinMode(pinLedRed, OUTPUT);
  pinMode(pinBuzzer, OUTPUT);

  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
```



```

    lcd.print("Security System");
    lcd.setCursor(0, 1);
    lcd.print("System Active...");
    delay(2000);
    lcd.clear();
}

void loop() {
    pirState = digitalRead(pinPIR);

    digitalWrite(pinTrig, LOW);
    delayMicroseconds(2);
    digitalWrite(pinTrig, HIGH);
    delayMicroseconds(10);
    digitalWrite(pinTrig, LOW);
    duration = pulseIn(pinEcho, HIGH);
    distance = duration * 0.034 / 2;
    if (pirState == HIGH) {
        alertMode(distance);
    } else {
        safeMode();
    }
    delay(200);
}

void safeMode() {
    digitalWrite(pinLedGreen, HIGH);
    digitalWrite(pinLedRed, LOW);
    digitalWrite(pinBuzzer, LOW);

    lcd.setCursor(0, 0);

```

```
    lcd.print("Status: SAFE  ");
    lcd.setCursor(0, 1);
    lcd.print("No Motion  ");
}

void alertMode(int dist) {
    digitalWrite(pinLedGreen, LOW);
    digitalWrite(pinLedRed, HIGH);
    tone(pinBuzzer, 1000);
    lcd.setCursor(0, 0);
    lcd.print("Status: ALARM! ");
    lcd.setCursor(0, 1);
    lcd.print("Dist: ");
    lcd.print(dist);
    lcd.print(" cm  ");
    delay(100);
    noTone(pinBuzzer);
    digitalWrite(pinLedRed, LOW);
    delay(100);
}
```

## 5 ТЕСТУВАННЯ ТА РЕЗУЛЬТАТИ

Перевірка працездатності розробленого пристрою «Автономна охоронна система» проводилася у два етапи: перевірка монтажу апаратної частини та функціональне тестування програмного забезпечення з імітацією тривожних подій.

### 5.1 Підготовка до тестування

Перед подачею живлення було проведено візуальний огляд макетної плати для виявлення можливих помилок монтажу:

- перевірено правильність підключення датчиків HC-SR501 та HC-SR04 (відповідність пінів VCC та GND, щоб уникнути короткого замикання);
- переконано у коректності підключення I2C-модуля LCD дисплея (лінії SDA та SCL);
- перевірено полярність світлодіодів та наявність струмообмежувальних резисторів у їхніх колах;
- перевірено надійність з'єднань провідників на платі breadboard.

### 5.2 Процес тестування

Після підключення плати Arduino UNO до персонального комп'ютера через USB-інтерфейс було виконано компіляцію та завантаження скетчу в пам'ять мікроконтролера. В ході тестування було змодельовано роботу системи у двох режимах:

- **Режим «Очікування» (відсутність руху):**
  - Система ініціалізується, на LCD дисплеї з'являється стартове повідомлення.
  - Через 2 секунди активується зелений світлодіод.
  - На дисплеї відображається напис «Status: SAFE», звуковий сигнал відсутній.

- **Режим «Тривога» (імітація проникнення):**

- При помаху рукою перед лінзою датчика HC-SR501 система миттєво реагує зміною стану.
- Зелений світлодіод гасне, починає блимати червоний світлодіод.
- Активується п'єзозумер, подаючи переривчастий звуковий сигнал.
- На дисплей виводиться повідомлення «Status: ALARM!».
- У нижньому рядку дисплея в реальному часі відображається відстань до об'єкта (руки), виміряна ультразвуковим датчиком HC-SR04 (наприклад, «Dist: 15 cm»).

### 5.3 Результати

Під час випробувань було встановлено:

- **Чутливість:** PIR-датчик коректно фіксує рух на відстані до 3 метрів (в межах кімнати);
- **Точність вимірювань:** Ультразвуковий датчик надає покази відстані з похибкою не більше 1-2 см, що є достатнім для визначення положення об'єкта;
- **Індикація:** Інформація на LCD дисплеї відображається чітко, контрастність налаштована коректно. Логіка перемикання світлодіодів (Червоний/Зелений) повністю відповідає стану системи;
- **Стабільність:** Збоїв у роботі мікроконтролера при одночасній роботі двох датчиків та дисплея не виявлено.

### 5.4. Висновки

Результати тестування підтверджують, що розроблена охоронна система працює стабільно і виконує поставлене технічне завдання. Забезпечено коректну взаємодію датчиків руху та відстані з засобами візуального та звукового сповіщення. Пристрій готовий до демонстрації як діючий макет охоронної сигналізації.

## ВИСНОВКИ

У ході виконання курсової роботи було успішно розроблено та реалізовано макет автономної охоронної системи на базі мікроконтролерної платформи Arduino Uno з функцією вимірювання відстані до об'єкта.

- **Апаратна реалізація:** Опрацьовано принципи побудови систем безпеки з використанням різнотипних сенсорів. Успішно реалізовано спільну роботу пасивного інфрачервоного датчика (HC-SR501) для виявлення руху та активного ультразвукового сонара (HC-SR04) для визначення відстані. Використання інтерфейсу I2C для підключення LCD-дисплея дозволило значно спростити схемотехніку та заощадити цифрові порти мікроконтролера;
- **Програмна реалізація:** Розроблено гнучкий програмний алгоритм мовою C++, який забезпечує опитування датчиків у реальному часі. Використання спеціалізованих бібліотек (LiquidCrystal\_I2C, Wire) дозволило налаштувати коректний вивід текстової інформації, а логічні умови забезпечують безпомилкову реакцію системи на тривожні події;
- **Працездатність системи:** Тестування підтвердило повну відповідність роботи пристрою технічному завданню. Система коректно переходить з режиму очікування в режим тривоги при виявленні руху, супроводжуючи це світлозвуковою індикацією та відображенням точної відстані до порушника на екрані.
- **Навчальна цінність:** Робота дозволила закріпити знання з основ електроніки та схемотехніки, зокрема особливостей підключення навантажень через транзисторні ключі або резистори, а також поглибити навички роботи з протоколами передачі даних та налагодженням програмного коду;

**Перспективи розвитку:**

Розроблений пристрій є базовим прототипом, який має значний потенціал для модернізації та перетворення на повноцінну систему «Розумний дім» або комерційну сигналізацію. До основних напрямів подальшого вдосконалення можна віднести:

- **Дистанційне сповіщення:** Інтеграція GSM-модуля (наприклад, SIM800L) або Wi-Fi модуля (ESP8266) для надсилання SMS-повідомлень або сповіщень у месенджер власника при спрацюванні сигналізації;
- **Контроль доступу:** Додавання матричної клавіатури (Keypad 4x4) або RFID-зчитувача для можливості постановки та зняття об'єкта з охорони за допомогою пароля або картки доступу;
- **Енергонезалежність:** Розробка схеми безперебійного живлення з використанням літій-іонних акумуляторів та модуля зарядки, а також програмна оптимізація енергоспоживання (режим сну) для тривалої автономної роботи;
- **Логування подій:** Підключення модуля годинника реального часу (RTC) та SD-карти для запису історії спрацювань (дата, час та відстань до об'єкта) у текстовий файл;
- **Конструктивне виконання:** Перенесення схеми з макетної плати (breadboard) на друковану плату (PCB) та розробка корпусу методом 3D-друку для захисту електроніки від зовнішніх впливів.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. **Блум Д.** Вивчаємо **Arduino**: інструменти та методи технічного чарівництва / Джеремі Блум. — К.: БХВ, 2016. — 336 с.
2. **Шилдт Г.** C++: базовий курс / Герберт Шилдт. — 3-тє вид. — М.: Вільямс, 2019. — 624 с.
3. **Петренко А. І.** Основи мікропроцесорної техніки та програмування мікроконтролерів: навчальний посібник / А. І. Петренко. — К.: Каравела, 2020. — 288 с.
4. **Рюмик С. М.** Мікроконтролери AVR: практикум для початківців / С. М. Рюмик. — Х.: Ранок, 2018. — 112 с.
5. **Галкін В. І.** Промислова електроніка та мікросхемотехніка: підручник / В. І. Галкін. — К.: Вища школа, 2017. — 320 с.
6. **Alvins, B.** Mastering Arduino Programming (2021). *Призначення: Програмування мікроконтролерів, синтаксис C/C++.*
7. **Schwartz, M.** Arduino Electronics Blueprints (2019). *Призначення: Інтеграція компонентів, використання бібліотек, створення скетчів.*
8. **Axelson, D.** Embedded Systems Design (2020). *Призначення: Фундаментальні основи проєктування вбудованих систем, архітектура.*
9. **Marr, B.** Internet of Things (IoT) in Business (2022). *Призначення: Концепції IoT, мережеві протоколи та стандарти для пристроїв.*

**ДОДАТОК 1**

Принципова електрична схема підключення