

**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 МТ-41

Владислав ЯЦОЛА

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

Яцолі Владиславу Висильовичу

(прізвище, ім'я та по батькові)

**навчальної дисципліни: ПРОГРАМНО-АПАРАТНІ ЗАСОБИ
INTERNET-РЕЧЕЙ**

Студент групи: МТ-41

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

2. Дата видачі завдання: _____ "22" вересня
2025 р.

3. Термін здачі курсового проєкту: _____ "15" 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 ВОР (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			Владислав ЯЦОЛА
	(підпис)		(імя та прізвище)
Керівник проекту			Остап ЮНАК
	(підпис)		(імя та прізвище)

ЗМІСТ

ВСТУП	3
1 ОГЛЯД ЛІТЕРАТУРИ/АНАЛОГІВ	5
2 ТЕХНІЧНЕ ЗАВДАННЯ	7
3 АПАРАТНА ЧАСТИНА	10
4 ПРОГРАМНА ЧАСТИНА	12
5 ТЕСТУВАННЯ ТА РЕЗУЛЬТАТИ	19
ВИСНОВКИ	21
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	23
ДОДАТКИ	24

ВСТУП

Мета курсового проєкту:

Основною метою виконання даної роботи є комплексне проєктування, програмна реалізація та практичне виготовлення апаратно-програмного пристрою на базі мікроконтролерної платформи Arduino1. Функціональне призначення розроблюваної системи полягає у забезпеченні охоронних функцій об'єкта. Окрім створення фізичного макета, невід'ємною частиною проєкту є підготовка технічної документації, яка включає аналіз схемотехнічних рішень, опис алгоритму роботи та інструкцію для користувача.

Актуальність теми:

Вибір даного напрямку дослідження зумовлений стрімким розвитком технологій автоматизації та зростаючим попитом на інтегровані інженерні рішення3. У сучасному світі спостерігається тенденція до поєднання апаратного забезпечення (hardware) та гнучких програмних алгоритмів (software), де роль мікроконтролерних систем стає ключовою як у промисловому секторі, так і в побутовій сфері.

Технологічна та інженерна значущість проєкту:

- **Розвиток концепції Інтернету речей (IoT):** Сучасні системи моніторингу та керування висувають високі вимоги до енергоефективності та компактності пристроїв. Платформа Arduino стала світовим стандартом для оперативного прототипування гаджетів, що

взаємодіють із фізичним світом, що повністю узгоджується з глобальними трендами розвитку IoT.

- **Доступність та відкритість (Open Source):** Завдяки відкритій архітектурі, Arduino суттєво знижує поріг входження в електроніку, дозволяючи розробникам фокусуватися на логіці проєкту, а не на складному проєктуванні друкованих плат чи низькорівневому кодуванні.
- **Міждисциплінарний підхід:** Реалізація такого пристрою вимагає синергії знань із різних галузей: цифрової електроніки, проєктування інтерфейсів та програмування мовою C++. Це дозволяє здобути практичні навички, необхідні для роботи у сферах вбудованих систем та робототехніки.

Практична цінність роботи:

- **Прикладний характер:** Курсова робота виходить за межі теорії, оскільки передбачає створення діючого фізичного прототипу, що виконує функції збору даних та автоматичного реагування.
- **Економічна ефективність:** Використання доступної елементної бази дозволяє мінімізувати витрати на створення пристрою, отримуючи при цьому повнофункціональний виріб.
- **Масштабованість:** Створений макет може слугувати основою для подальшої модернізації або переходу на промислові контролери.

1. ОГЛЯД ЛІТЕРАТУРИ ТА АНАЛОГІВ

1.1. Апаратно-програмна платформа Arduino

Arduino представляє собою платформу з відкритим вихідним кодом, призначену для розробки прототипів електронних пристроїв. Вона базується на принципах гнучкості та простоти використання. Система складається з фізичної друкованої плати з мікроконтролером та інтегрованого середовища розробки (IDE).

Історично проєкт зародився у 2005 році в Інституті проєктування взаємодії (м. Івреа, Італія). Команда засновників, до якої входили Массімо Банці та інші, ставила за мету надати студентам недорогий інструмент для реалізації інтерактивних проєктів.

Для даної роботи обрано плату **Arduino Uno R3** на базі мікроконтролера ATmega328P. Її перевагами є:

- **Доступність:** Низька вартість компонентів.
- **Універсальність:** ПЗ сумісне з Windows, Linux та macOS.
- **Зручність:** Завантаження коду через USB без зовнішніх програматорів.
- **Відкритість:** Доступні схеми спрощують підключення модулів.

1.2. Програмування: C++ та Arduino IDE

Програмування контролера здійснюється мовою C++ у середовищі Arduino IDE, яке використовує фреймворк Wiring. Фреймворк — це набір інструментів, що прискорюють розробку, дозволяючи уникнути рутинного коду.

Структура програми (скетчу) складається з двох основних блоків:

- `setup()` — виконується одноразово при старті для конфігурації портів;
- `loop()` — виконується циклічно, містячи основний алгоритм.

1.3. Принципи дії сенсорів

- **Датчик руху HC-SR501 (PIR):** Працює на основі фіксації змін теплового випромінювання. Будь-який об'єкт із температурою вище нуля випромінює тепло. Коли теплий об'єкт перетинає зони лінзи Френеля, піроелектричний елемент генерує сигнал тривоги.
- **Ультразвуковий далекомір HC-SR04:** Використовує принцип ехолокації. Випромінювач посилає сигнал частотою 40 кГц, який відбивається від перешкоди. Мікроконтролер вимірює час повернення сигналу та обчислює дистанцію.

1.4. Засоби відображення (LCD 1602)

Для виводу інформації обрано рідкокристалічний дисплей. Ключовою особливістю є використання модуля I2C, що дозволяє передавати дані лише двома проводами (SDA та SCL), значно спрощуючи схему.

2. ТЕХНІЧНЕ ЗАВДАННЯ

2.1. Найменування та тема проєкту

Розробка, макетування та налагодження автономної охоронної системи на базі мікроконтролера Arduino з інтеграцією датчиків руху (HC-SR501) та відстані (HC-SR04), а також виведенням інформаційних повідомлень на LCD-дисплей.

2.2. Мета та призначення роботи

Метою роботи є проєктування та створення функціонального прототипу пристрою безпеки, використовуючи апаратну обчислювальну платформу Arduino.

Призначення системи полягає у забезпеченні автоматизованого моніторингу визначеної зони (периметра). Головне завдання — налаштування коректної апаратної та програмної взаємодії між різнотипними сенсорами (інфрачервоним та ультразвуковим), забезпечення обробки вхідних сигналів мікроконтролером та організація зрозумілого інтерфейсу візуалізації поточного стану системи на рідкокристалічному екрані.

2.3. Основні етапи виконання

Для досягнення поставленої мети необхідно вирішити наступні інженерні задачі:

- **Теоретична підготовка:** Опрацювання методичних вказівок, аналіз документації (datasheets) на електронні компоненти.
- **Проектування схеми:** Розробка принципової електричної схеми

пристрою, визначення точок підключення периферії до портів контролера Arduino Uno.

- **Монтаж:** Виконання фізичної комутації електронних компонентів на безпайковій макетній платі (breadboard) відповідно до розробленої схеми.
- **Програмування:** Написання, компіляція та налагодження керуючого коду (скетчу), що реалізує логіку роботи системи.

2.4. Вимоги до апаратного забезпечення

Проектована система повинна включати наступний перелік компонентів та модулів:

- **Живлення:** Джерело стабілізованого живлення постійного струму (напруга 5 В) або живлення від USB-порту.
- **Керування:** Мікроконтролерна плата Arduino Uno R3 (на базі чіпа ATmega328P).
- **Монтаж:** Макетна плата (Breadboard) для швидкого прототипування без використання паяльного обладнання.
- **Візуалізація:** Символьний дисплей LCD 1602, оснащений адаптером послідовного інтерфейсу I2C.
- **Світлова індикація:** Світлодіоди зеленого (режим "Охорона") та червоного (режим "Тривога") кольорів.
- **Сенсори:**
 - Пасивний інфрачервоний сенсор руху HC-SR501 (PIR).
 - Ультразвуковий далекомір HC-SR04 (Sonar).
- **Звукова сигналізація:** П'єзоелектричний зумер (активний або пасивний).

- **Пасивні компоненти:** Резистори номіналом 220 Ом (для обмеження струму світлодіодів).
- **Комутація:** Набір з'єднувальних проводів (джамперів) типів «штекер-штекер» та «штекер-гніздо».

2.5. Сценарій функціонування системи

Логіка роботи пристрою базується на циклічному опитуванні датчиків.

Передбачено наступні стани системи:

- **Режим спокою (Standby):** За умови відсутності руху в зоні дії PIR-датчика, система сигналізує про безпеку увімкненням зеленого світлодіода. На дисплеї відображається текстове повідомлення про нормальний стан ("SAFE").
- **Режим тривоги (Alert):** У момент детектування теплового об'єкта відбувається миттєва активація червоного світлодіода та вмикається звукова сирена. Одночасно запускається вимірювання дистанції сонаром HC-SR04.
- **Інформування:** Отримані дані про дистанцію до порушника транслюються на LCD-екран у реальному часі.
- **Відновлення:** Після припинення руху система автоматично повертається у режим очікування.

3. АПАРАТНА ЧАСТИНА

3.1. Обґрунтування вибору компонентної бази (BOM)

Ключовим етапом проєктування є формування специфікації матеріалів (Bill of Materials — BOM). У рамках даного проєкту було здійснено підбір компонентів, виходячи з критеріїв сумісності, надійності та простоти інтеграції.

- **Мікроконтролер:** Плата Arduino Uno обрана як центральний вузол обробки даних завдяки наявності достатньої кількості цифрових портів (GPIO) та підтримці апаратних переривань.
- **Сенсор HC-SR501:** Використовується для ініціалізації тривоги. Його принцип дії дозволяє ігнорувати статичні об'єкти та реагувати виключно на рух теплих тіл, що мінімізує хибні спрацювання.
- **Сенсор HC-SR04:** Забезпечує точне вимірювання відстані (від 2 до 400 см). Використання ультразвуку дозволяє працювати незалежно від рівня освітлення приміщення.
- **LCD 1602 з I2C:** Вибір модифікації дисплея з I2C-конвертером є критично важливим для оптимізації схеми, оскільки це дозволяє вивільнити 6 цифрових портів контролера, які були б зайняті при стандартному паралельному підключенні.

3.2. Схемотехнічні рішення та комутація

Фізичне з'єднання компонентів виконано на макетній платі відповідно до розробленої принципової схеми. Розподіл портів мікроконтролера реалізовано наступним чином:

- **Підсистема вимірювання відстані (HC-SR04):**

- *Trig (Тригер)*: Підключено до цифрового піна **D9**. Через цей контакт подається стартовий імпульс тривалістю 10 мкс.
- *Echo (Приймач)*: Підключено до цифрового піна **D10**. Цей вхід фіксує тривалість зворотного сигналу.
- *Живлення*: VCC (+5V) та GND (Земля).

- **Підсистема детекції руху (HC-SR501):**

- *Signal Output*: Вихід датчика підключено до цифрового піна **D7**.
Високий логічний рівень на цьому піні сигналізує про наявність руху.

- **Інтерфейс відображення (LCD 1602 I2C):**

- *SDA (Data)*: Лінія даних підключена до аналогового піна **A4**.
- *SCL (Clock)*: Лінія тактування підключена до аналогового піна **A5**.

- **Система індикації та оповіщення:**

- *Зелений LED*: Анод підключено до піна **D4** через резистор 220 Ом (для індикації безпеки).
- *Червоний LED*: Анод підключено до піна **D5** через резистор 220 Ом (для індикації тривоги).
- *Зумер (Buzzer)*: Позитивний контакт підключено до піна **D6**.

Така топологія схеми забезпечує надійне розмежування сигнальних ліній та кіл живлення, що гарантує стабільну роботу пристрою.

4. ПРОГРАМНА ЧАСТИНА

4.1. Середовище та інструментарій розробки

Розробка керуючої програми (прошивки) для мікроконтролера здійснювалася в інтегрованому середовищі Arduino IDE. Код написано високорівневою мовою C++ із використанням об'єктно-орієнтованого підходу.

Для забезпечення коректної взаємодії мікроконтролера з периферійними пристроями було залучено зовнішні бібліотеки:

- Wire.h — базова бібліотека для організації обміну даними по послідовній шині I2C.
- LiquidCrystal_I2C.h — спеціалізована бібліотека для керування символьним дисплеєм, яка містить готові методи для ініціалізації екрана, керування підсвіткою та позиціювання курсора.

4.2. Алгоритмічна структура програми

Загальна логіка роботи програмного забезпечення побудована за модульним принципом і складається з трьох ключових блоків: декларації глобальних змінних, процедури ініціалізації та основного виконавчого циклу.

Блок 1: Декларація констант та змінних

На початку скетчу (програми) виконується підключення бібліотек та оголошення об'єкта дисплея з адресою 0x27 та параметрами екрана 16x2. Для підвищення читабельності коду та спрощення подальшої модернізації, фізичні номери пінів мікроконтролера присвоєно іменованим константам:

- Сенсори: pinTrig (9), pinEcho (10), pinPIR (7).

- Виконавчі пристрої: pinLedGreen (4), pinLedRed (5), pinBuzzer (6).

Також оголошено глобальні змінні duration (для збереження часу проходження звукового сигналу) та distance (для результату обчислення відстані).

Блок 2: Ініціалізація системи (Setup)

Функція void setup() виконується одноразово при подачі живлення. У ній реалізовано наступний алгоритм:

- **Конфігурація портів вводу-виводу (GPIO):**
 - Піни pinTrig, pinLedGreen, pinLedRed, pinBuzzer налаштовано як виходи (OUTPUT), оскільки вони керують зовнішніми навантаженнями.
 - Піни pinEcho та pinPIR налаштовано як входи (INPUT) для зчитування сигналів від сенсорів.
- **Запуск дисплея:** Виконується метод lcd.init() та вмикається підсвітка lcd.backlight().
- **Тестова індикація:** На екран виводиться вітальне повідомлення "Security System", яке через 2 секунди зникає, сигналізуючи про готовність системи до роботи.

Блок 3: Головний цикл (Loop)

Функція void loop() містить нескінченний цикл опитування датчиків:

- **Зчитування PIR-сенсора:** Контролер перевіряє логічний стан на піні 7.
- **Робота з сонаром:**
 - Генерується короткий імпульс (10 мкс) на піні pinTrig.
 - Функція pulseIn вимірює довжину імпульсу відповіді на піні pinEcho.

○ Отриманий час конвертується у відстань за формулою: $S \text{ (см)} = \text{duration} \times 0.034 / 2$.

● **Умовна логіка:**

0 Якщо датчик руху повертає HIGH (рух виявлено), викликається функція обробки тривоги `alertMode()`.

○ В іншому випадку виконується функція штатного режиму `safeMode()`.

4.3. Опис функцій реакції

Для структурування коду логіка індикації винесена в окремі підпрограми:

- `safeMode()`: Встановлює високий рівень на зеленому світлодіоді, вимикає червоний світлодіод і зумер. На LCD виводиться статус "SAFE".
- `alertMode(int dist)`: Вмикає червоний світлодіод, генерує звуковий сигнал частотою 1000 Гц (`tone()`) та виводить напис "ALARM!" разом із поточною відстанню до об'єкта.

4.4. Лістинг програми

Нижче наведено повний код прошивки мікроконтролера:

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

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

```
// Ініціалізація дисплея (адреса 0x27, 16 символів, 2 рядки)
```

```
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

```
// Декларація пінів підключення
```

```
const int pinTrig = 9; const int
```



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

```
// Глобальні змінні
```

```
long duration; int  
distance; int  
pirState = LOW;
```

```
void setup() {
```

```
    // Налаштування режимів роботи пінів
```

```
    pinMode(pinTrig, OUTPUT);  
    pinMode(pinEcho, INPUT);  
    pinMode(pinPIR, INPUT);  
    pinMode(pinLedGreen, OUTPUT);  
    pinMode(pinLedRed, OUTPUT);  
    pinMode(pinBuzzer, OUTPUT);
```

```
    // Ініціалізація інтерфейсів
```

```
    Serial.begin(9600);  
    lcd.init(); lcd.backlight();
```

```
    // Стартова заставка
```

```
    lcd.setCursor(0, 0);
```

```

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

```

```

void loop() {
    // 1. Зчитування стану датчика руху
    pirState = digitalRead(pinPIR);

    // 2. Вимірювання відстані (Sonar)
    digitalWrite(pinTrig, LOW);
    delayMicroseconds(2);
    digitalWrite(pinTrig, HIGH);
    delayMicroseconds(10);
    digitalWrite(pinTrig, LOW);

    duration = pulseIn(pinEcho, HIGH);    distance =
duration * 0.034 / 2; // Конвертація у см

    // 3. Логіка вибору режиму  if (pirState
== HIGH) {    alertMode(distance); //
Режим ТРИВОГИ
    } else {    safeMode();        // Режим
СПОКОЮ }

```

```

    delay(200); // Затримка для стабільності
}

// Функція режиму "Спокій" void safeMode() {
    digitalWrite(pinLedGreen, HIGH);
    digitalWrite(pinLedRed, LOW);
    digitalWrite(pinBuzzer, LOW); // Вимкнути звук

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

// Функція режиму "Тривога" void alertMode(int
dist) { digitalWrite(pinLedGreen, LOW);
digitalWrite(pinLedRed, HIGH); tone(pinBuzzer,
1000); // Звуковий сигнал 1 кГц

    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); //
```

Перерывчастий звук

```
}
```

5. ТЕСТУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

5.1. Методика проведення випробувань

Комплексна перевірка працездатності розробленої системи «Автономна охоронна сигналізація» здійснювалася у два послідовні етапи: верифікація апаратного монтажу (Hardware Check) та функціональне тестування програмного алгоритму в реальних умовах.

Етап 1: Попередня діагностика

Перед першим запуском пристрою було проведено ретельний візуальний та інструментальний контроль макетної плати для уникнення критичних апаратних збоїв:

- **Інспекція живлення:** Перевірено коректність комутації ліній VCC (+5V) та GND, оскільки помилка полярності могла призвести до виходу з ладу сенсорів HC-SR501 та HC-SR04.
- **Перевірка інтерфейсів:** Підтверджено надійність з'єднань шини I2C (лінії SDA/SCL) для забезпечення стабільного зв'язку з дисплеєм.
- **Компонентна база:** Перевірено наявність струмообмежувальних резисторів у колах світлодіодів.

Етап 2: Сценарне тестування

Після завантаження прошивки в мікроконтролер було відтворено базові сценарії експлуатації:

- Штатний режим (Режим очікування):

При подачі живлення система успішно ініціалізується, виводячи стартову заставку. Після цього активується зелений індикатор, а на дисплеї

відображається статус «Status: SAFE», що свідчить про відсутність руху в контрольованій зоні.

- **Режим реагування (Тривога):**

При імітації проникнення (поява людини в зоні дії PIR-сенсора) система миттєво (затримка < 0.5 с) переходить в активний стан:

- Вимикається зелений і вмикається червоний світлодіод.
- Активується переривчастий звуковий сигнал зумера.
- На дисплей виводиться попередження «ALARM!» та динамічно оновлюється відстань до об'єкта.

5.2. Результати дослідження

У ході експериментів було зафіксовано наступні технічні показники роботи системи:

- **Радіус ефективної дії:** Інфрачервоний датчик HC-SR501 демонструє впевнену детекцію об'єктів на дистанції до 3–4 метрів.
- **Точність вимірювань:** Ультразвуковий сонар забезпечує визначення відстані з похибкою не більше 1–2 см, що є достатнім показником для локалізації порушника.
- **Інформативність:** Використання LCD-дисплея дозволяє оперативно отримувати точні метричні дані, а світлозвукова індикація забезпечує надійне сповіщення.
- **Стабільність:** Під час тривалого тестування (1 година безперервної роботи) збоїв у роботі мікроконтролера чи конфліктів переривань не виявлено.

ВИСНОВКИ

У результаті виконання курсового проєкту було успішно вирішено поставлене технічне завдання: розроблено, змонтовано та протестовано діючий прототип автономної охоронної системи на базі платформи Arduino Uno.

Підсумки реалізації проєкту:

- Апаратна архітектура:

Вдалося реалізувати ефективну схему, що поєднує сенсори різного фізичного принципу дії (інфрачервоний та ультразвуковий). Впровадження протоколу I2C для керування дисплеєм дозволило оптимізувати використання портів мікроконтролера та спростити монтаж.

- Програмне забезпечення:

Створений алгоритм на мові C++ забезпечує надійну обробку сигналів у реальному часі. Використання модульного підходу та спеціалізованих бібліотек (LiquidCrystal_I2C, Wire) гарантує стабільність роботи пристрою та легкість його подальшої модифікації.

- Практична цінність:

Система повністю відповідає вимогам технічного завдання, демонструючи чітке перемикання між режимами охорони та тривоги. Робота над проєктом дозволила закріпити інженерні навички у сферах схемотехніки, програмування вбудованих систем та налагодження електроніки.

Перспективи розвитку:

Розроблений макет є базою для створення повноцінного комерційного продукту. Подальша модернізація може включати:

- **Дистанційний моніторинг:** Інтеграція GSM або Wi-Fi модулів для відправки повідомлень на смартфон власника.
- **Автономність:** Додавання блоку безперебійного живлення (Li-Ion акумулятори) для роботи при знеструмленні мережі.
- **Конструктивне виконання:** Перенесення схеми на друковану плату (PCB) та розробка 3D-моделі корпусу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Блум Д. *Вивчаємо Arduino: інструменти та методи технічного чарівництва* / Джеремі Блум. — Київ : БХВ, 2016. — 336 с.
2. Галкін В. І. *Промислова електроніка та мікросхемотехніка: підручник* / В. І. Галкін. — Київ : Вища школа, 2017. — 320 с.
3. Петренко А. І. *Основи мікропроцесорної техніки та програмування мікроконтролерів: навчальний посібник* / А. І. Петренко. — Київ : Каравела, 2020. — 288 с.
4. Рюмик С. М. *Мікроконтролери AVR: практикум для початківців* / С. М. Рюмик. — Харків : Ранок, 2018. — 112 с.
5. Шилдт Г. *C++: базовий курс* / Герберт Шилдт. — 3-тє вид. — Москва : Вільямс, 2019. — 624 с.
6. Alvins B. *Mastering Arduino Programming*. Packt Publishing, 2021.
7. Axelson D. *Embedded Systems Design*. O'Reilly Media, 2020.
8. Marr B. *Internet of Things (IoT) in Business*. Wiley, 2022.
9. Schwartz M. *Arduino Electronics Blueprints*. Packt Publishing, 2019.

