

**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
модуля зв'язку GSM/GPRS, інфрачервоним датчиком руху і
ультразвуковим далекоміром.»**

Виконав студент групи ІІІ-41

Сергій САРАБУН

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

ПЕЛЕЩАК

Богдан

ЗАВДАННЯ
на курсовий проєкт

Сарабуну Сергію Степановичу

(прізвище, ім'я та по батькові)

з навчальної дисципліни: ПРОГРАМНО-АПАРАТНІ ЗАСОБИ
INTERNET-РЕЧЕЙ

Студент групи: **ІІІ-41**

1. Тема проєкту: Виготовлення охоронного пристрою на платі Arduino з використанням модуля зв'язку GSM/GPRS, інфрачервоним датчиком руху і ультразвуковим далекоміром.

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 проекту (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 безпечного 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)

Сергій САРАБУН

(імя та прізвище)

Керівник проекту

(підпис)

Остап ЮНАК

(імя та прізвище)

ЗМІСТ

ВСТУП	3
1 ОГЛЯД ЛІТЕРАТУРИ/АНАЛОГІВ	5
2 ТЕХНІЧНЕ ЗАВДАННЯ	9
3 АПАРАТНА ЧАСТИНА.....	11
4 ПРОГРАМНА ЧАСТИНА.....	14
5 ТЕСТУВАННЯ ТА РЕЗУЛЬТАТИ.....	20
ВИСНОВКИ.....	23
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	25
ДОДАТКИ.....	26

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 пов'язана з тим, що в сучасних інженерних проектах з'являється великий попит на поєднання апаратних та програмних засобів, а також зростає роль мікроконтролерних платформ як у навчальному процесі, так і в різних галузях промисловості.

• Технологічна та Інженерна Актуальність

- **Поширеність IoT (Internet of Things):** Сучасні системи автоматизації, контролю та управління використовують компактні, економічні та енергоефективні пристрої. Платформа Arduino є фактичним стандартом для швидкого створення прототипів та пристроїв, які взаємодіють з реальним світом, що відповідає загальному напрямку розвитку Інтернету речей;
- **Демократизація розробки:** Arduino, як відкрита платформа для роботи з апаратною та програмною частиною, значно знижує складність для новачків і покращує можливості для створення нових рішень. Це дає змогу більше уваги приділити логіці проекту та його функціям, а не витрачати час на складне програмування та дрібні деталі в апаратній реалізації;

- **Міждисциплінарний характер:** Розробка пристрою на базі програмного та апаратного забезпечення вимагає поєднання знань з електроніки, програмування мікроконтролерів на мовах C/C++, теорії керування та створення інтерфейсів. Це дозволяє набути розгорнуті навички інженера, які є важливими для працевлаштування у галузях робототехніки, автоматизації та систем з вбудованими пристроями;
- **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 працездатний пристрій з мінімальними витратами на компоненти та час розробки. Це має важливе значення для інноваційних стартапів та освітніх проєктів;
 - **Можливість подальшого розвитку:** Розроблений прототип є надійною основою для подальшого розширення, оновлення та переходу на промислові мікроконтролери, що надає науковий потенціал для виконання дипломної роботи або створення продукту для ринку.

Отже, створення апаратно-програмного пристрою на основі Arduino — це вчасне, економічно обґрунтоване та високопрактичне дослідження, яке поєднує вивчення сучасних апаратних засобів і розвиток ключових інженерних навичок.

1 ОГЛЯД ЛІТЕРАТУРИ/АНАЛОГІВ

1.1 Апаратно-програмна платформа Arduino

Arduino — це відкрита платформа для створення електронних пристроїв, яка легко використовується. Вона складається з фізичної друкованої плати, яка називається мікроконтролером, та спеціального програмного середовища, де пишуть код.

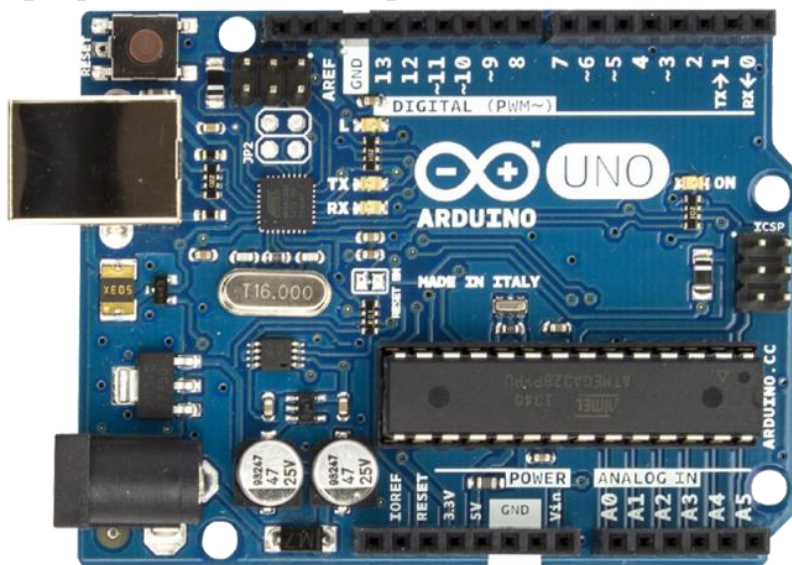


Рис. 1.1 Вигляд плати Arduino з мікроконтролером ATmega328P2

Історія створення:

Проект Arduino був започаткований у 2005 році в Інституті проєктування взаємодії (Interaction Design Institute Ivrea) у місті Івреа, Італія. Його засновники — Массімо Банці, Девід Куартільєс, Том Іго, Джанлука Мартіно та Девід Мелліс. Основна мета створення цієї платформи полягала в наданні студентам дешевого та простого інструменту для створення інтерактивних проєктів, який не потребував глибоких знань з схемотехніки.

Причини вибору платформи:

Для даної курсової роботи вибраний мікроконтролер Arduino Uno R3, який побудований на основі ATmega328P2.

Його популярність визначається такими факторами::

- **Доступність:** Низька ціна компонентів;

- **Кросплатформеність:** Середовище розробки функціонує на Windows, macOS та Linux;
- **Простота:** Зручний інтерфейс програмування через USB, не потрібно використовувати зовнішні програматори.
- **Відкрита архітектура:** Схеми плат доступні безкоштовно, що зручно для підключення різноманітних датчиків та модулів.

1.2 Мова програмування C++ та середовище Arduino IDE

Програмування мікроконтролера відбувається на мові C++. Середовище Arduino IDE використовує спеціальний спрощений вид цієї мови, який називається **Wiring**.

Фреймворк — це набір готових інструментів, бібліотек, шаблонів та правил, які надають структуру для створення програмного забезпечення, спрощують і прискорюють процес розробки. Він допомагає розробникам уникнути написання стандартного коду з нуля, дозволяючи зосередитися на унікальній бізнес-логіці продукту. Фреймворки використовуються для різних типів проектів, включаючи веб-додатки, мобільні та десктопні програми.

C++ — це універсальна мова програмування високого рівня, яка підтримує об'єктно-орієнтовану парадигму.

У контексті розробки для 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, щоб налаштувати конфігурацію пінів;
- `loop()` — виконується циклічно, реалізує основний алгоритм роботи пристрою..

1.3 Принципи роботи використаних сенсорів

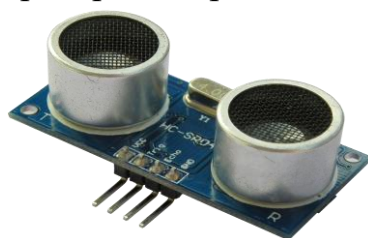
Датчик руху HC-SR501 (PIR Sensor). Принцип роботи пасивного інфрачервоного датчика (Passive Infrared Sensor) заснований на виявленні змін інфрачервоного, або теплового, випромінювання. Люди та інші предмети, які мають температуру вище абсолютного нуля, випромінюють тепло. Піроелектричний елемент усередині датчика реагує на зміну сили цього випромінювання, коли теплий об'єкт, наприклад людина, переходить через сектори лінзи Френеля, і генерує електричний сигнал тривоги.



Рис 1.3.1 Вигляд датчику руху HC-SR501

Ультразвуковий датчик відстані HC-SR04. Робота датчика заснована на принципі ехолокації (SONAR). Пристрій має два п'єзoelementи: випромінювач і приймач.

1. Випромінювач створює пакет ультразвукових імпульсів з частотою 40 кГц;
2. Ультразвукова хвиля відбивається від перешкоди і повертається до приймача;
3. Мікроконтролер вимірює час, за який сигнал подорожує, і



обчислює відстань.

Рис 1.3.2 Вигляд датчику HC-SR04

Рідкокристалічний дисплей (LCD 1602) з I2C. Для зображення даних використано екран на основі рідких кристалів. У цій реалізації застосовано адаптер I2C (Inter-Integrated Circuit). Цей послідовний протокол передає дані через дві сигнальні лінії (SDA та SCL), що значно спрощує виготовлення схеми пристрою у порівнянні з паралельним підключуванням.



Рис 1.3.2 Вигляд рідкокристалічного дисплею (LCD 1602) з I2C

2 ТЕХНІЧНЕ ЗАВДАННЯ

2.1 Назва роботи

Виготовлення захисного пристрою на основі Arduino, який використовує датчики HC-SR501 та HC-SR04, а також виводить інформацію на дисплей LCD.

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

- Вивчити інструкцію до практичного завдання;
- Ознайомитись із електричною схемою охоронного пристрою, який побудовано на Arduino Uno;
- Зробити на монтажній платі цифрову схему охоронного пристрою за схемою;
- Запрограмувати пристрій використовуючи скетч 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 ґрунтується на постійному спостереженні за зоною відповідальності за допомогою пасивного інфрачервоного датчика руху (HC-SR501). У режимі очікування пристрій перебуває у спокої: горить зелений світлодіод, а на екрані відображається повідомлення про те, що система безпечна. Коли виявляється теплове випромінювання рухомого об'єкта, мікроконтролер швидко переключається в режим тривоги: з'являється червоний світлодіод, увімкниться звуковий сигнал, а ультразвуковий сенсор (HC-SR04) вимірюватиме відстань до об'єкта і виводитиме цю данину на екран в режимі реального часу. Після припинення руху система самостійно повертається у режим очікування.

3 АПАРАТНА ЧАСТИНА

3.1 Підбір BOM

BOM (скорочення від Bill of Materials) — це список всіх матеріалів і компонентів, які потрібні для того, щоб зробити пристрій або виріб. Цей список можна подати у вигляді таблиці, в якій вказуються усі деталі, які використовуються в проекті.

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

№	Назва компонента	Кількість	Позначення на схемі	Характеристики	Примітка
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		Анод (+)	Digital Pin 5	

№	Компонент	Вивід компонента	З'єднано з піном Arduino	Примітка / Функція
	Світлодіод (Зелений)	Катод (-)	GND	Через резистор 220 Ом
4	П'єзозумер	Плюс (+)	Digital Pin 6	-
		Мінус (-)	GND	

4 ПРОГРАМНА ЧАСТИНА

Розробка програмного забезпечення для мікроконтролера виконувалася в середовищі Arduino IDE. Як мова програмування використовувалася C++, з впровадженням стандартних бібліотек Wiring та спеціалізованих бібліотек для роботи з периферією, таких як Wire.h та LiquidCrystal_I2C.h.

Основна функція пристрою полягає в постійному спостереженні за простором за допомогою інфрачервоного датчика руху, а також вимірюванні відстані до об'єкта у разі тривоги, з подальшим виведенням отриманих даних на екран LCD та вимкненням звукового сигналу.

Програма має три основні частини: підключення бібліотек та оголошення констант, налаштування портів (ініціалізація) та основний цикл роботи..

4.1 Опис алгоритму роботи

- **Підключення бібліотек та оголошення змінних:** Для роботи з дисплеєм через шину I2C використовується бібліотека 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:

- LCD дисплей ініціалізується, а його підсвітка вмикається.
- Порти, які використовуються для управління світлодіодами, зумером та тригером ультразвукового датчика (Trig), налаштовуються як виходи (OUTPUT).
- Порти, які використовуються для зчитування сигналів з PIR-датчика та ехо-сигналу (Echo), налаштовуються як входи (INPUT).
- На екран виводиться вітання, яке підтверджує запуск системи.

Основний цикл: Програма виконує нескінченний цикл, що реалізує логіку охоронної системи:

- **Зчитування даних:** Мікроконтролер перевіряє цифровий вивід датчика HC-SR501. Під час цього на вивід Trig надходить імпульс тривалістю 10 мкс, а за допомогою функції pulseIn вимірюється час, протягом якого сигнал повертається на вивід Echo. Цей час потім переводиться в відстань у сантиметрах.
- **Обробка логіки:**
 - **Режим «Тривога»:** Якщо датчик руху показує високий рівень (HIGH), програма вмикає червоний світлодіод, генерує звуковий сигнал (функція tone), а на екрані виводиться повідомлення "ALARM!" разом із відстанню до об'єкта.
 - **Режим «Охорона»:** якщо не виявлено руху (LOW), вмикається зелений світлодіод, червоний вимикається, а на екрані з'являється статус "SAFE".
 - Цикл повторюється із невеликою затримкою для того, щоб показання були стабільнішими.

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 роботи пристрою «Автономна охоронна система» відбулася у дві частини: перевірка встановлення апаратної частини та тестування програмного забезпечення з навмисним викликом тривожних ситуацій.

Підготовка до тестування

- Було здійснено візуальний огляд макетної плати перед подачею живлення, щоб виявити можливі помилки при монтажі:
- перевірено, чи правильно підключені датчики HC-SR501 та HC SR04, у тому числі відповідність пинів VCC та GND, щоб уникнути короткого замикання;
- переконанося, що I2C-модуль LCD дисплея підключено правильно, з урахуванням ліній SDA та SCL;
- перевірено полярність світлодіодів та наявність струмообмежувальних резисторів в їхніх колах;
- перевірено, чи є надійними з'єднання провідників на платі breadboard..

5.1 Процес тестування

Після того як Arduino UNO підключили до комп'ютера через USB, здійснили компіляцію та завантаження коду у пам'ять мікроконтролера. Під час тестування система перевірялася в двох режимах:

- **Режим «Очікування» (відсутність руху):**

- Система запускається, на екрані лінійного дисплея з'являється повідомлення про запуск.
- Через 2 секунди загоряється зелений світлодіод.
- На екрані пише «Status: SAFE», звук не вмикатиметься.

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

- Якщо промахнутися рукою перед датчиком HC-SR501, система миттєво змінює стан.
- Зелений світлодіод гасне, з'являється червоний, який починає мигати.
- Запускається п'єзозумер, відтворюється переривчастий звук.
- На екрані з'являється повідомлення «Status: ALARM!»
- У нижньому рядку екрану в реальному часі показується відстань до об'єкта (руки), виміряна ультразвуковим датчиком HC-SR04 (наприклад, «Dist: 15 см»).

- **Результати**

Під час тестування виявлено:

Чутливість: датчик PIR добре визначає рух у 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 екрані відображається чітко, контраст встановлений правильно. Робота світлодіодів (червоний/зелений) відповідає стану системи.
- **Стабільність:** при одночасній роботі двох датчиків та екрану не виявлено помилок в роботі мікроконтролера..

5.2 Висновки

Тксти показали, що створена система охорони працює стабільно і виконує всі вимоги технічного завдання.

Забезпечена правильна робота датчиків руху та відстані разом із засобами візуального та звукового сповіщення. Присадок готовий до демонстрації як робочий макет системи охорони.

ВИСНОВКИ

У процесі виконання курсової роботи було створено та реалізовано прототип автономної системи охорони на основі платформи Arduino Uno, яка має можливість вимірювати відстань до об'єкта.

- Апаратна частина: Вивчено та застосовано принципи побудови системи безпеки з використанням різних сенсорів.

Вдалося здійснити взаємодію між пасивним інфрачервоним датчиком (HC-SR501), який виявляє рух, та активним ультразвуковим сонаром (HC-SR04), який визначає відстань. Використання інтерфейсу I2C для підключення дисплея дозволило спростити схему та зберегти цифрові порти мікроконтролера.

- Програмна частина: Розроблено гнучкий алгоритм на мові C++, який забезпечує реальний час відповіді датчиків.

Застосування спеціалізованих бібліотек (LiquidCrystal_I2C, Wire) дозволило правильно виводити текстову інформацію, а логічні умови забезпечують вірну реакцію системи на тривожні події.

- Функціональність системи: Під час тестування підтвердилася повна відповідність роботи пристрою технічному завданню.

Система правильно переходить з режиму очікування в режим тривоги при виявленні руху, супроводжуючи це світлово-звуковою індикацією та відображенням точного значення відстані до порушника на екрані.

- Навчальна цінність: Робота допомогла 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 налагодження програмного коду;

5.3 Перспективи розвитку:

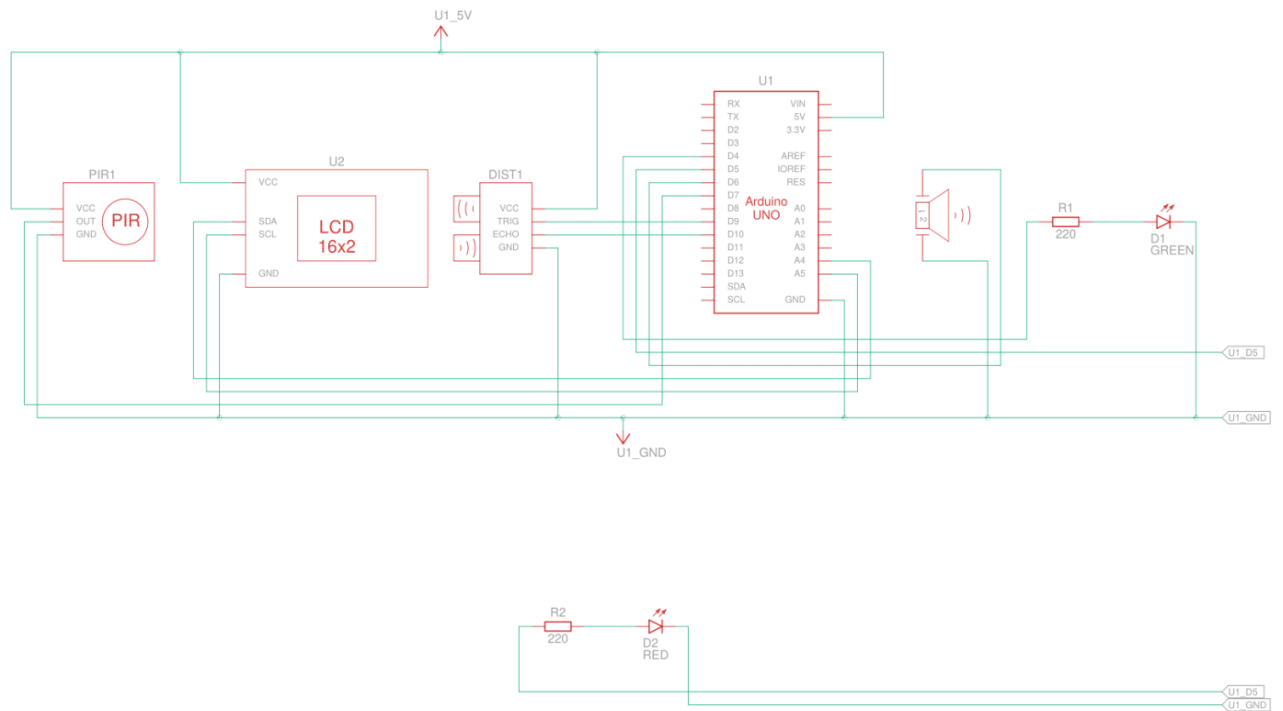
Розроблений пристрій є базовим прототипом, який має великий потенціал для вдосконалення та перетворення на повноцінну систему «Розумний дім» або комерційну сигналізацію. До основних напрямів подальшого покращення можна віднести:

- Дистанційне сповіщення: Інтеграція модуля GSM (наприклад, SIM800L) або Wi-Fi (наприклад, ESP8266), щоб відправляти SMS-повідомлення або сповіщення у месенджер власника коли сигналізація включається;
- Контроль доступу: Додавання матричної клавіатури (Keypad 4x4) або читального модуля RFID, щоб дозволити встановлення або відкінення об'єкта з охорони за допомогою пароля або картки;
- Енергонезалежність: Створення схеми живлення, щоб пристрій працював безперервно за допомогою літій-іонних акумуляторів, модуля зарядки, а також програмних засобів для зменшення споживання енергії (режим сну), щоб працював довго без підключення до електричної мережі;
- Логування подій: Підключення модуля годинника реального часу (RTC) та SD-карти для запису історії подій (дата, час і відстань до об'єкта) у текстовий файл;
- Конструктивне виконання: Перенесення схеми з макетної плати (breadboard) на друковану плату (PCB) та створення корпусу через 3D-друк для захисту електроніки від впливу зовнішнього середовища.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

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

ДОДАТОК 1



Принципова електрична схема підключення