

**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

Володимир САВКА

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

Савці Володимиру Андрійовичу

(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 BOM (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 безпального 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			Володимир САВКА
	(підпис)		(імя та прізвище)
Керівник проекту			Остап ЮНАК
	(підпис)		(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	19
ВИСНОВКИ	21
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	23
ДОДАТОК 1	24

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 проекту 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.

Історія створення:

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.

Причини вибору платформи:

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 (теплого) 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.

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.

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.

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

Розробити та реалізувати макет автоматизованої охоронної системи на базі мікроконтролера Arduino, яка здатна виявляти рух та визначати відстань до об'єкта, візуалізуючи стан системи на LCD дисплеї та за допомогою світлової індикації.

2.3 Завдання

- Проаналізувати принцип роботи датчиків HC-SR501 (PIR) та HC-SR04 (ультразвуковий).
- Розробити електричну схему з'єднання компонентів.
- Скласти алгоритм роботи пристрою (логіку реагування на події).
- Написати програмний код (скетч) для Arduino.
- Зібрати макет пристрою (або змодельовати його у середовищі Tinkercad) та провести тестування.

2.4 Технічні вимоги

- 5 В постійного струму;
- Arduino Uno R3;
- Breadboard (безпайкова макетна плата);
- LCD дисплей 1602 з I2C модулем;

- Світлодіоди: білий, зелений, жовтий, 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	PIR	PIR Motion Sensor, 5 В	Виявлення руху об'єктів
4	HC-SR04	1	SONAR	Ultrasonic Sensor, 2–400 см	Вимірювання відстані до об'єкта
5	LCD 1602 (I2C)	1	LCD	16x2 символів, I2C інтерфейс	Відображення статусу системи
6	Світлодіод (Червоний)	1	LED_Y	5 мм, 2 В, 20 мА	Індикація тривоги
7	Світлодіод (Зелений)	1	LED_R	5 мм, 2 В, 20 мА	Індикація режиму "Охорона"

№	Назва компонента	Кількість	Позначення на схемі	Характеристики	Примітка
8	П'єзозумер	1	SP1	Активний, 5 В	Звукове оповіщення
9	Резистори	2	R1-R4	220 Ом, 0.25 Вт	Обмеження струму LED
10	Провідники	20	-	Джампер (Тато-тато, Тато-мама)	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 2	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

№	Компонент	Вивід компонента	З'єднано з піном Arduino	Примітка / Функція
4	Світлодіод (Зелений)	Анод (+)	Digital Pin 4	Через резистор 220 Ом
		Катод (-)	GND	
5	Світлодіод (Червоний)	Анод (+)	Digital Pin 6	Через резистор 220 Ом
		Катод (-)	GND	
6	Світлодіод (Білий)	Анод (+)	Digital Pin 3	Через резистор 220 Ом
		Катод (-)	GND	
7	Світлодіод (Жовтий)	Анод (+)	Digital Pin 5	Через резистор 220 Ом
		Катод (-)	GND	
8	П'єзозумер	Плюс (+)	Digital Pin 6	-
		Мінус (-)	GND	

4 ПРОГРАМНА ЧАСТИНА

Програмне забезпечення (прошивка) для мікроконтролера розроблено у середовищі Arduino IDE мовою C++. Структура програми базується на класичній архітектурі вбудованих систем і складається з підключення бібліотек, оголошення глобальних змінних та двох основних функцій: `setup()` та `loop()`.

Підготовчий етап: На початку коду підключається бібліотека `LiquidCrystal_I2C.h`, яка необхідна для керування LCD дисплеєм через інтерфейс I2C. Це дозволяє суттєво зменшити кількість задіяних пінів мікроконтролера. Далі оголошуються константи для пінів підключення світлодіодів та датчиків, що спрощує подальше налаштування схеми.

Ініціалізація (`void setup`): У цьому блоці відбувається конфігурація портів введення-виведення. Піни, до яких підключені світлодіоди та керуючий пін ультразвукового датчика (`Trig`), налаштовуються на вихід (`OUTPUT`). Піни, що приймають сигнали від PIR-датчика та ехо-сигнал сонара (`Echo`), налаштовуються на вхід (`INPUT`). Також тут відбувається ініціалізація дисплея, увімкнення його підсвітки та запуск послідовного порту для можливого налагодження.

Основний цикл (`void loop`): Логіка роботи програми побудована циклічно. Кожна ітерація циклу виконує наступні кроки:

- **Збір даних:** Мікроконтролер генерує короткий імпульс (10 мкс) на піні `Trig` ультразвукового датчика. Після цього функція `pulseIn` вимірює тривалість повернення звукової хвилі на пін `Echo`. Отримай час конвертується у відстань у сантиметрах за формулою фізики. Паралельно зчитується цифровий сигнал із датчика руху HC-SR501;

- **Обробка даних та логіка розгалуження:** Використовується конструкція if-else if-else для визначення пріоритетності подій. Найвищий пріоритет має ультразвуковий датчик: якщо відстань менша за заданий поріг (20 см), система переходить у стан "Тривога". Якщо загрози по відстані немає, перевіряється датчик руху. При наявності сигналу система переходить у стан "Попередження". Якщо жодна з умов не виконується, система залишається у стані "Спокій";
- **Виведення інформації:** Залежно від обраного стану, мікроконтролер подає напругу на відповідний світлодіод і гасить інші, а також надсилає відповідні текстові повідомлення на LCD дисплей;

4.1 Код програми

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

LiquidCrystal_I2C lcd(0x27, 16, 2);

const int pirPin = 2;
const int trigPin = 9;
const int echoPin = 10;

const int ledWhite = 3;
const int ledGreen = 4;
const int ledYellow = 5;
const int ledRed = 6;

long duration;
int distance;
int pirState = LOW;

void setup() {
  pinMode(pirPin, INPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  pinMode(ledWhite, OUTPUT);
  pinMode(ledGreen, OUTPUT);
  pinMode(ledYellow, OUTPUT);
  pinMode(ledRed, OUTPUT);

  digitalWrite(ledWhite, HIGH);

  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("Security System");
```

```

lcd.setCursor(0, 1);
lcd.print("Loading...");
delay(2000);
lcd.clear();

Serial.begin(9600);
}

void loop() {
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    duration = pulseIn(echoPin, HIGH);
    distance = duration * 0.034 / 2;

    pirState = digitalRead(pirPin);

    if (distance > 0 && distance < 20) {
        digitalWrite(ledRed, HIGH);
        digitalWrite(ledYellow, LOW);
        digitalWrite(ledGreen, LOW);

        lcd.setCursor(0, 0);
        lcd.print("ALARM! ALARM!  ");
        lcd.setCursor(0, 1);
        lcd.print("Dist: " + String(distance) + "cm  ");
    }
    else if (pirState == HIGH) {
        digitalWrite(ledRed, LOW);
        digitalWrite(ledYellow, HIGH);
        digitalWrite(ledGreen, LOW);

        lcd.setCursor(0, 0);

```

```
    lcd.print("Motion Detected!");  
    lcd.setCursor(0, 1);  
    lcd.print("Dist: " + String(distance) + "cm    ");  
}  
else {  
    digitalWrite(ledRed, LOW);  
    digitalWrite(ledYellow, LOW);  
    digitalWrite(ledGreen, HIGH);  
  
    lcd.setCursor(0, 0);  
    lcd.print("System Secure    ");  
    lcd.setCursor(0, 1);  
    lcd.print("Monitoring...    ");  
}  
  
    delay(200);  
}
```

5 ТЕСТУВАННЯ ТА РЕЗУЛЬТАТИ

Тестування розробленого пристрою проводилося у декілька етапів для перевірки працездатності всіх вузлів схеми.

- **Етап 1: Перевірка живлення та ініціалізації.** При підключенні пристрою до USB-порту комп'ютера загоряється білий світлодіод, що свідчить про наявність живлення 5В. На дисплеї з'являється стартове повідомлення, після чого система переходить у режим очікування. Зелений світлодіод активний. *Результат:* Позитивний. Схема зібрана вірно, коротких замикань немає.
- **Етап 2: Тестування інфрачервоного датчика (PIR).** Для перевірки було здійснено рух рукою на відстані 1-2 метри від датчика. Система миттєво відреагувала: зелений світлодіод згас, увімкнувся жовтий, на екрані з'явився напис "Motion Detected!". Через 3-5 секунд після припинення руху система автоматично повернулася у стан спокою. *Результат:* Датчик HC-SR501 коректно фіксує теплові збурення.
- **Етап 3: Тестування ультразвукового датчика (HC-SR04).** Перед датчиком було розміщено перешкоду (книгу). При наближенні перешкоди на відстань менше 20 см система перейшла у червоний режим тривоги. На дисплеї відобразилася точна відстань до об'єкта. Було порівняно показники на дисплеї з вимірами лінійкою. *Результат:* Похибка вимірювання становить не більше 1-2 см, що є допустимим для охоронних систем такого класу.

- **Етап 4: Комплексне тестування.** Симулювалася ситуація проникнення: спочатку вхід у зону дії PIR-датчика (жовтий сигнал), потім наближення до охоронюваного об'єкта (червоний сигнал). Система коректно перемикала режими згідно з пріоритетами, не допускаючи зависань.

5.1 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 спроектовано та реалізовано прототип автоматизованої охоронної системи на базі мікроконтролера Arduino Uno. Було вирішено такі завдання:

- Освоєно принципи роботи з датчиками фізичних величин (PIR та ультразвук).
- Набуто практичних навичок складання електронних схем на макетній платі Breadboard.
- Розроблено алгоритм та написано програмний код, який забезпечує логічне опрацювання сигналів та взаємодію з користувачем через візуальні інтерфейси.

Створений пристрій повністю відповідає технічним вимогам: він здатний виявляти рух та контролювати периметр, інформуючи про рівень небезпеки. Робота підтвердила ефективність використання платформи Arduino для створення бюджетних систем безпеки.

5.2 Перспективи розвитку:

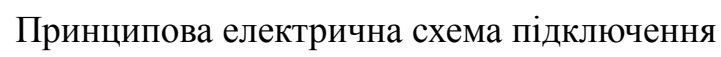
Розроблений пристрій є базовим прототипом, який має значний потенціал для модернізації та перетворення на повноцінну систему «Розумний дім» або комерційну сигналізацію. До основних напрямів подальшого вдосконалення можна віднести:

- **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

- **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.*



Принципова електрична схема підключення