

**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 ПП-41

Чуха МАКСИМ

Керівник проєкту:

Остап ЮНАК

Курсовий проєкт перевірений

і допущений до захисту

“ ___ ” _____ 2025 р.

Курсовий проєкт при захисті оцінений

Львів 2025

ЗМІСТ

ЗАВДАННЯ

на курсовий проект

Чуха Максим Андрійович

(прізвище, ім'я та по батькові)

з навчальної дисципліни: ПРОГРАМНО-АПАРАТНІ ЗАСОБИ
INTERNET-РЕЧЕЙ

Студент групи: **ІІІ-41**

1. Тема проекту: 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 проекту: _____ "15" грудня
2025 р.

4. Вихідні дані до проекту:

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 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 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

<u>Назва етапів</u>	<u>Термін виконання</u>	<u>Примітка</u>
<u>Вступ</u>		
<u>1 ОГЛЯД ЛІТЕРАТУРИ/АНАЛОГІВ</u>		
<u>2 ТЕХНІЧНЕ ЗАВДАННЯ</u>		
<u>3 АПАРАТНА ЧАСТИНА</u>		
<u>4 ПРОГРАМНА ЧАСТИНА</u>		
<u>5 ТЕСТУВАННЯ ТА РЕЗУЛЬТАТИ</u>		
<u>Висновки</u>		

Перелік посилань		
------------------	--	--

Студент

(підпис)

Чуха Максим

(імя та прізвище)

Керівник проекту

(підпис)

Остап ЮНАК

(імя та прізвище)

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:

Стрімке поширення технологій Інтернету речей (IoT), автоматизованих пристроїв та вбудованих систем сьогодні суттєво впливає як на побутову сферу, так і на промислові процеси. Саме тому виконання даної курсової роботи є особливо актуальним. Все більше сучасних моніторингових та охоронних систем проектуються на базі мікроконтролерних платформ, що дозволяє досягти компактності, енергоефективності та зручності модернізації. Завдяки цьому такі системи можуть легко адаптуватися до різних умов експлуатації та вимог користувачів, що робить мікроконтролерні рішення одним із провідних напрямів розвитку прикладної електроніки.

- **Технологічна та Інженерна Актуальність**

- **Активне впровадження IoT-рішень:** Більшість сучасних систем безпеки, автоматизації та моніторингу середовища ґрунтуються на використанні доступних і надійних апаратних платформ. Arduino належить до найпоширеніших рішень для створення прототипів пристроїв, що взаємодіють із фізичними об'єктами та сенсорами, і відповідає ключовим тенденціям розвитку технологій Інтернету речей;
- **Відкритість та доступність платформи:** Arduino є апаратно-програмною платформою з відкритою архітектурою, що забезпечує швидке проєктування та модифікацію електронних систем без необхідності глибокого занурення у низькорівневе програмування. Це дає змогу зосередити увагу на функціональній логіці пристрою, структурі програмного коду та підвищенні надійності роботи системи;
- **Міждисциплінарний характер розробки:** Створення охоронного пристрою поєднує елементи електроніки та схемотехніки, програмування мікроконтролерів мовою C/C++, опрацювання сигналів та організацію взаємодії з користувачем.
- **Прикладна та Практична Актуальність**
 - **Прикладна спрямованість роботи:** Курсова робота має не лише теоретичний характер, а передусім орієнтована на реалізацію реального діючого пристрою, який здатний визначати наявність руху, вимірювати відстань до об'єкта та формувати тривожні сигнали у разі виявлення загрози. Це дає можливість перевірити на практиці правильність обраних технічних рішень та алгоритмів роботи системи;

- **Економічна обґрунтованість:** Використання платформи Arduino разом із недорогими електронними компонентами забезпечує можливість створення ефективного охоронного пристрою з невеликими матеріальними витратами та в короткі терміни. Такий підхід є особливо корисним для навчальних, дослідницьких та експериментальних розробок;
- **Можливість подальшої модернізації:** Розроблений прототип може слугувати базовою платформою для розширення функціоналу — зокрема, інтеграції бездротових модулів, підключення мережевих сервісів або переходу до більш складних промислових рішень. Це підвищує цінність роботи як з практичної, так і з навчальної точки зору;

Отже, створення апаратно-програмного пристрою на основі платформи Arduino є актуальним, економічно раціональним та практично значущим дослідженням, яке поєднує засвоєння сучасних апаратних технологій із формуванням важливих інженерних компетенцій.

• ОГЛЯД ЛІТЕРАТУРИ/АНАЛОГІВ

• Апаратно-програмна платформа Arduino

Arduino — це універсальна апаратно-програмна платформа з відкритим кодом, призначена для розробки, тестування та прототипування електронних пристроїв і систем автоматизації. Вона поєднує мікроконтролерну плату та програмне середовище, що дає можливість оперативно створювати керуючі алгоритми без складної попередньої підготовки.

Історія створення та розвиток платформи:

Платформа Arduino була створена у 2005 році в Interaction Design Institute Ivrea (Італія) як доступний інструмент для навчання студентів електроніці та програмуванню інтерактивних систем. До засновників проєкту належать Массімо Банці, Девід Куартільєс, Том Іго, Джанлука Мартіно та Девід Мелліс.

Завдяки відкритій ліцензії, широкій спільноті розробників і простоті використання Arduino швидко отримала світове поширення та стала одним із базових стандартів для прототипування вбудованих систем.

Обґрунтування вибору Arduino Uno:

Для реалізації даного курсового проєкту обрано плату **Arduino Uno R3** на базі мікроконтролера **ATmega328P**. Такий вибір зумовлений низкою переваг:

- **Кросплатформеність:** середовище Arduino IDE підтримує Windows, Linux та macOS, що забезпечує зручність роботи на різних системах;
- **Фінансова доступність:** невисока вартість комплектуючих робить платформу оптимальною для навчальних, дослідницьких і прототипних розробок;
- **Простота підключення:** програмування здійснюється через USB-інтерфейс без використання зовнішніх програматорів;
- **Відкрита архітектура:** наявність відкритих схем та документації полегшує інтеграцію різноманітних модулів і датчиків;
- **Надійність та стандартизація:** форм-фактор Uno R3 сумісний із широким набором шилдів (Shields), що спрощує розширення функціоналу та забезпечує стабільну роботу пристрою.

- **Мова програмування C++ та середовище Arduino IDE**

Програмування мікроконтролера здійснюється мовою C++ у середовищі **Arduino IDE**, яке використовує спрощений фреймворк **Wiring**.

Фреймворк — це набір готових бібліотек, інструментів і шаблонів, що формують основу програмної системи та дозволяють прискорити розробку програмного забезпечення. Він зменшує обсяг рутинного коду та дає змогу зосередитися на реалізації основної логіки роботи пристрою. Подібні фреймворки широко застосовуються в розробці веб-, мобільних і десктопних застосунків.

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.

- **Принципи роботи використаних сенсорів**

- **Пасивний інфрачервоний датчик руху HC-SR501**

Датчик HC-SR501. Належить до класу пасивних інфрачервоних сенсорів (PIR) і призначений для виявлення руху теплових об'єктів. Його робота ґрунтується на фіксації змін інфрачервоного випромінювання в зоні контролю. При переміщенні людини або іншого теплового об'єкта змінюється тепловий фон, що реєструється піроелектричним елементом датчика та перетворюється на цифровий сигнал для обробки мікроконтролером.

Ультразвуковий датчик відстані HC-SR04. Датчик HC-SR04 застосовується для вимірювання відстані до об'єктів за принципом ехолокації. Після подачі керуючого імпульсу датчик випромінює ультразвукову хвилю, яка відбивається від перешкоди та повертається до приймача. Мікроконтролер визначає час проходження сигналу туди й назад і на його основі обчислює відстань до об'єкта.

- Випромінювач формує серію ультразвукових імпульсів із частотою 40 кГц;
- Ці звукові хвилі відбиваються від об'єкта та повертаються назад до приймача;
- Мікроконтролер фіксує час руху сигналу туди й назад та на його основі обчислює відстань до перешкоди.

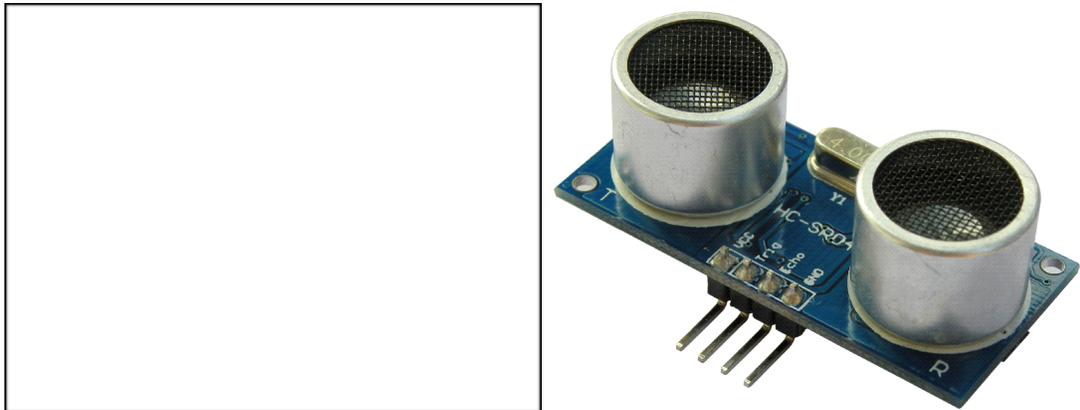


Рис 1.3.2 Вигляд датчику HC-SR04

Рідкокристалічний дисплей (LCD 1602) з I2C. Для відображення інформації у проєкті використовується символьний LCD-дисплей формату 16×2 з модулем I2C. Застосування послідовного інтерфейсу значно зменшує кількість задіяних пінів мікроконтролера, спрощує підключення та підвищує надійність роботи пристрою.

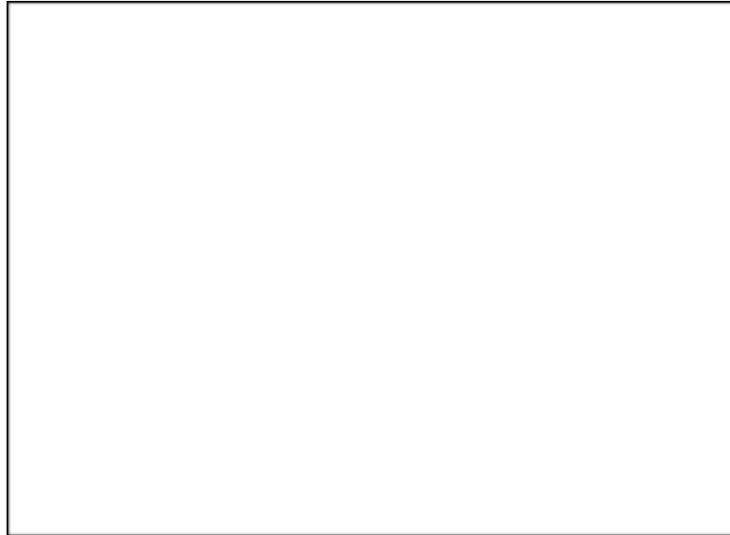


Рис 1.3.2 Вигляд рідкокристалічного дисплею (LCD 1602) з I2C

- **ТЕХНІЧНЕ ЗАВДАННЯ**

- **Назва роботи**

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

- **Мета роботи**

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

- **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;

- **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».

- **Очікуваний результат**

Охоронний пристрій функціонує шляхом постійного контролю заданої зони за допомогою пасивного інфрачервоного сенсора **HC-SR501**. У штатному режимі очікування система перебуває у спокої: світиться зелений

світлодіод, а на рідкокристалічному дисплеї відображається повідомлення про відсутність загрози. При виявленні руху теплового об'єкта мікроконтролер миттєво переводить систему у режим «Тривога». У цей момент активується червоний світлодіод і спрацьовує звуковий сигналізатор, сповіщаючи про потенційну загрозу.

• АПАРАТНА ЧАСТИНА

• Підбір BOM

BOM BOM (**Bill of Materials**) — це структурований список компонентів, необхідних для складання пристрою. До нього входять керуючий мікроконтролер, сенсори, елементи індикації, джерела живлення та з'єднувальні провідники. Такий підхід дозволяє систематизувати процес складання, а також спрощує подальше обслуговування або модернізацію пристрою.

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	Приєм 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	

• ПРОГРАМНА ЧАСТИНА

Програмне забезпечення для мікроконтролера розроблялось у середовищі **Arduino IDE** з використанням мови **C++**. У кодї застосовуються стандартні бібліотеки **Wiring** та додаткові бібліотеки для роботи з периферійними модулями, зокрема **Wire.h** і **LiquidCrystal_I2C.h**.

Опис алгоритму роботи

- **Підключення необхідних бібліотек та оголошення змінних:**

На початку коду підключаються необхідні бібліотеки для роботи периферійних пристроїв. Для керування LCD-дисплеєм через I2C

використовується бібліотека **LiquidCrystal_I2C**.

Оголошуються пін-коди для пристроїв:

- **Ініціалізація:** Під час подачі живлення на плату Arduino виконується функція **setup()**, де налаштовуються апаратні ресурси пристрою:
- **Основний цикл програми:** Основна логіка охоронного пристрою реалізована в нескінченному циклі **loop()**, який працює протягом всього часу роботи системи:
 - **Зчитування даних:** Мікроконтролер опитує PIR-датчик. Одночасно на пін Trig подається імпульс тривалістю 10 мкс, а функція **pulseIn** вимірює час повернення сигналу на пін Echo. На основі цього обчислюється відстань до об'єкта (см).
 -
 - **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);  
}
```

• ТЕСТУВАННЯ ТА РЕЗУЛЬТАТИ

Перевірка функціонування розробленого пристрою здійснювалась у два етапи: спершу проводився контроль правильності складання апаратної частини, а після цього — функціональне тестування програмного забезпечення з моделюванням тривожних ситуацій.

• Підготовка до тестування

Перед подачею живлення на макетну плату виконано детальний візуальний огляд для виявлення можливих помилок монтажу:

- Перевірено правильність підключення датчиків **HC-SR501** та **HC-SR04**, зокрема відповідність контактів **VCC** та **GND** для уникнення короткого замикання;
- Підтверджено коректне підключення I2C-інтерфейсу **LCD-дисплея** (лінії **SDA** та **SCL**);
- Перевірено полярність світлодіодів і наявність струмообмежувальних резисторів у їхніх колах;
- Оцінено надійність з'єднань провідників на макетній платі (breadboard).

• Процес тестування

Після підключення плати **Arduino UNO** до ПК через USB виконано компіляцію та завантаження програмного коду. Тестування проводилось у двох режимах:

- **Режим «Очікування» (відсутність руху)**
- **Режим «Тривога» (імітація проникнення)**

- **Висновки**

Результати перевірки вказують, що розроблений пристрій може бути використаний як повноцінний демонстраційний макет системи сигналізації. У ході перевірки підтверджено надійну роботу охоронної системи та її відповідність заданим технічним вимогам. Забезпечується коректна та узгоджена взаємодія датчиків руху і відстані з візуальними та звуковими засобами сповіщення.

ВИСНОВКИ

У межах курсової роботи було розроблено та успішно реалізовано прототип автономної охоронної системи на базі платформи Arduino Uno, що дозволяє вимірювати відстань до об'єкта..

- **Апаратна реалізація:** Опановано принципи побудови систем безпеки з різними типами сенсорів. Реалізовано спільну роботу пасивного PIR-датчика (HC-SR501) для виявлення руху та активного ультразвукового сенсора (HC-SR04) для визначення відстані. Використання I2C-інтерфейсу для LCD-дисплея спростило схемотехніку та зекономило цифрові порти мікроконтролера;
- **Програмна реалізація:** Створено гнучкий алгоритм на C++, який забезпечує опитування датчиків у реальному часі. Бібліотеки **LiquidCrystal_I2C** та **Wire** дозволили організувати коректний вивід

інформації, а логічні умови гарантують надійну реакцію системи на тривожні події;

- **Працездатність системи:** Тестування підтвердило відповідність технічним вимогам. Пристрій стабільно переходить із режиму очікування до режиму тривоги, супроводжуючи це світлозвуковою індикацією та відображенням точного значення відстані до об'єкта;
- **Навчальна цінність:** **Навчальна цінність:** робота дозволила закріпити знання з електроніки та схемотехніки, особливостей підключення навантажень через транзисторні ключі та резистори, а також поглибити навички роботи з протоколами передачі даних і налагодженням програмного коду.

- **Перспективи розвитку:**

Розроблений макет є базовим прототипом із потенціалом для модернізації та інтеграції у системи «Розумний дім» або комерційні сигналізації. Основні напрями вдосконалення:

- **Дистанційне сповіщення:** інтеграція GSM-модуля (SIM800L) або Wi-Fi модуля (ESP8266) для надсилання SMS або сповіщень у месенджери;
- **Контроль доступу:** додавання матричної клавіатури (Keypad 4x4) або RFID-зчитувача для постановки/зняття об'єкта з охорони;
- **Енергонезалежність;**

- **Логування подій:** використання літій-іонних акумуляторів та модуля зарядки, оптимізація програмного забезпечення (режим сну) для тривалої автономної роботи;
- **Конструктивне виконання:** перенесення електроніки на друковану плату (PCB) та виготовлення корпусу методом 3D-друку для захисту 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, V.** *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

