

**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 ПІ-41

Назар РОЖАК

Керівник проєкту:

Остап ЮНАК

Курсовий проєкт перевірений

і допущений до захисту

“ ____ ” _____ 2025 р.

Курсовий проєкт при захисті оцінений

Львів 2025

ЗАТВЕРДЖЕНО
на засіданні циклової комісії
«Фундаментальної підготовки»
Протокол № ____ від _____ 2025 р.
Голова комісії _____
Богдан ПЕЛЕЩАК

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 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)

НАЗАР РОЖАК

(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:

Метою курсової роботи є розробка апаратно-програмного пристрою на основі платформи Arduino, який виконує функції охоронної системи, а також підготовка технічної документації та опису принципу його роботи.

Актуальність:

Актуальність даної роботи зумовлена стрімким розвитком вбудованих систем та широким впровадженням мікроконтролерних рішень у сучасних технічних пристроях. Сьогодні апаратно-програмні комплекси активно використовуються у сфері автоматизації, безпеки, «розумних» будинків та Інтернету речей (IoT).

- **Технологічна та Інженерна Актуальність**

- **Використання Arduino відповідає сучасним тенденціям**

розвитку IoT-систем, які базуються на компактних, енергоефективних і недорогих пристроях. Платформа забезпечує зручну інтеграцію різноманітних датчиків і модулів, що робить її ідеальним інструментом для навчальних та прикладних проєктів.

- **Демократизація розробки: Arduino, як відкрита**

(Open-Source) апаратно-програмна платформа, значно знижує поріг входу для інженерів-початківців та сприяє створенню інновацій. Це дозволяє зосередитися на логіці проєкту та функціональності, а не на низькорівневому програмуванні та складному апаратному дизайні;

- **Міждисциплінарний характер:** Розробка

апаратно-програмного пристрою вимагає інтеграції знань з електроніки, програмування мікроконтролерів (C/C++), теорії керування та проєктування інтерфейсів. Це забезпечує набуття комплексних інженерних навичок, які є критично важливими для працевлаштування у сферах робототехніки, автоматизації та вбудованих систем;

- **Прикладна та Практична Актуальність**

- **Вирішення конкретної прикладної задачі:** Робота не

обмежується теоретичним аналізом, а передбачає створення працюючого фізичного прототипу (моніторинг параметрів, автоматизоване управління, збір даних). Це безпосередньо демонструє здатність автора перетворювати теоретичні знання на практичне інженерне рішення;

- **Економічна ефективність:** Використання Arduino дозволяє

створити функціональний пристрій з мінімальними витратами на компоненти та часом розробки. Це важливий аспект для інноваційних стартапів та освітніх проєктів;

- **Можливість подальшого розвитку:** Розроблений прототип

служує надійною базою для подальшого масштабування, модернізації та переходу на промислові мікроконтролери, що забезпечує наукову перспективу для дипломної роботи чи комерціалізації.

Таким чином, розробка апаратно-програмного пристрою на базі Arduino є своєчасним, економічно обґрунтованим і високопрактичним

дослідженням, що поєднує вивчення сучасних апаратних засобів та формування ключових інженерних компетенцій.

1 ОГЛЯД ЛІТЕРАТУРИ/АНАЛОГІВ

1.1 Апаратно-програмна платформа Arduino

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

Рис. 1.1 Вигляд плати Arduino з мікроконтролером ATmega328P2

Історія створення:

Проект Arduino був започаткований у 2005 році в Інституті проектування 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). Пасивний інфрачервоний датчик реагує на зміну теплового випромінювання у зоні контролю. При появі рухомого об'єкта змінюється рівень інфрачервоного випромінювання, що фіксується сенсором і перетворюється на електричний сигнал.

Рис 1.3.1 Вигляд датчику руху HC-SR501

Ультразвуковий датчик відстані HC-SR04. Датчик працює за принципом ехолокації. Він випромінює ультразвуковий імпульс, який відбивається від об'єкта та повертається назад. За часом проходження сигналу обчислюється відстань до перешкоди.

Рис 1.3.2 Вигляд датчику HC-SR04

Рідкокристалічний дисплей (LCD 1602) з I2C. Для відображення інформації використовується рідкокристалічний дисплей з інтерфейсом I2C, що дозволяє значно зменшити кількість з'єднувальних проводів і спростити підключення.

Рис 1.3.2 Вигляд рідкокристалічного дисплею (LCD 1602) з I2C

2 ТЕХНІЧНЕ ЗАВДАННЯ

2.1 Назва роботи

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

2.2 Мета роботи

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

2.3 Завдання

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

2.4 Технічні вимоги

- 5 В постійного струму;
- Arduino Uno R3;
- Breadboard (безпайкова макетна плата);
- LCD дисплей 1602 з I2C модулем;
- Світлодіоди: Зелений, Червоний;
- HC-SR501;
- HC-SR04;
- Резистори 220 Ом;
- П'єзозумер — для звукового сигналу;

- З'єднувальні провідники типу «тато-тато» та «тато-мама».

2.5 Очікуваний результат

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

3 АПАРАТНА ЧАСТИНА

3.1 Підбір BOM

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

Таблиця 3.1.1 Перелік необхідних компонентів

№	Назва компонента	Кількість	Позначення на схемі	Характеристики	Примітка
1	Arduino Uno	1	U1	Мікроконтролер ATmega328P	Основна плата керування
2	Плата breadboard	1	-	830 точок, безпайкова	Безпайна плата для створення конструкції
3	HC-SR501	1	PIR1	PIR Motion Sensor, 5 В	Виявлення руху об'єктів (тепла)
4	HC-SR04	1	DIST1	Ultrasonic Sensor, 2–400 см	Вимірювання відстані до об'єкта
5	LCD 1602 (I2C)	1	U2	16x2 символів, I2C інтерфейс	Відображення статусу системи
6	Світлодіод (Червоний)	1	D1_RED	5 мм, 2 В, 20 мА	Індикація тривоги
7	Світлодіод (Зелений)	1	D1_GREEN	5 мм, 2 В, 20 мА	Індикація режиму

№	Назва компонента	Кількість	Позначення на схемі	Характеристики	Примітка
					"Охорона"
8	П'єзозумер	1	SP1	Активний, 5 В	Звукове оповіщення
9	Резистори	2	R1, R2	220 Ом, 0.25 Вт	Обмеження струму LED
10	Провідники	20	-	Dupont (Male-Male, Male-Female)	Електричні з'єднання

Таблиця 3.1.2 Таблиця з'єднань пінів

№	Компонент	Вивід компонента	З'єднано з піном Arduino	Примітка / Функція
1	HC-SR04 (Ultrasonic)	VCC	5V	Живлення
		Trig	Digital Pin 9	Сигнал запуску
		Echo	Digital Pin 10	Прийом сигналу
		GND	GND	Земля
2	HC-SR501 (PIR)	VCC	5V	Живлення
		OUT	Digital Pin 7	Сигнал тривоги
		GND	GND	Земля
3	LCD 1602 (I2C)	GND	GND	Земля
		VCC	5V	Живлення
		SDA	Pin A4 (або SDA)	Шина даних I2C
		SCL	Pin A5 (або SCL)	Шина тактування

№	Компонент	Вивід компонента	З'єднано з піном Arduino	Примітка / Функція
				I2C
4	Світлодіод (Зелений)	Анод (+)	Digital Pin 4	Через резистор 220 Ом
		Катод (-)	GND	
5	Світлодіод (Зелений)	Анод (+)	Digital Pin 5	Через резистор 220 Ом
		Катод (-)	GND	
4	П'єзозумер	Плюс (+)	Digital Pin 6	-
		Мінус (-)	GND	

4 ПРОГРАМНА ЧАСТИНА

Програмне забезпечення реалізовано в середовищі Arduino IDE з використанням стандартних бібліотек. Алгоритм роботи побудований за принципом постійного контролю стану датчика руху та вимірювання відстані у разі спрацювання сигналізації.

Система працює у двох режимах:

- **очікування** — відсутність руху, активний зелений світлодіод;
- **тривога** — виявлення руху, звуковий сигнал, вивід відстані до об'єкта.

Алгоритм функціонування пристрою ґрунтується на постійному контролі навколишнього простору за допомогою інфрачервоного датчика руху. У разі виявлення активності система додатково визначає відстань до об'єкта, після чого виводить відповідні дані на LCD-дисплей і подає звуковий сигнал. Програмне забезпечення умовно поділяється на три логічні частини: підключення необхідних бібліотек і задання констант, ініціалізацію апаратних портів та основний цикл виконання програми.

4.1 Опис алгоритму роботи

- **Підключення бібліотек та оголошення змінних:** Для роботи з рідкокристалічним дисплеєм через інтерфейс I2C використовується бібліотека **LiquidCrystal_I2C**. З метою підвищення зручності читання програми та спрощення її налаштування всі задіяні виводи мікроконтролера описані у вигляді іменованих констант.
До сенсорів віднесено: **pinTrig (9)**, **pinEcho (10)** та **pinPIR (7)**.
Елементи індикації підключені до виводів: **pinLedGreen (4)**, **pinLedRed (5)** і **pinBuzzer (6)**. Також у програмі оголошено змінні

для збереження тривалості ехо-сигналу, обчисленої відстані та поточного стану датчика руху.

Під час запуску мікроконтролера виконується етап ініціалізації. У цей момент відбувається запуск LCD-дисплея та ввімкнення його підсвітки. Виводи, що керують світлодіодами, зумером і тригером ультразвукового датчика, налаштовуються як вихідні, тоді як входи PIR-датчика та лінія Echo конфігуруються для приймання сигналів. Після завершення налаштування на екрані виводиться повідомлення про старт системи.

Основна частина програми реалізована у вигляді нескінченного циклу, що відповідає за роботу охоронного пристрою. У процесі виконання мікроконтролер постійно зчитує сигнал із PIR-датчика. Одночасно формується імпульс тривалістю 10 мкс на виводі Trig, після чого за допомогою функції `pulseIn()` визначається час повернення ультразвукової хвилі. Отримане значення використовується для розрахунку відстані до об'єкта.

Далі відбувається аналіз отриманих даних:

- у режимі «Тривога», коли фіксується рух, вмикається червоний світлодіод, активується звуковий сигнал, а на дисплеї відображається повідомлення «ALARM!» разом із виміряною відстанню;
- у режимі «Охорона», за відсутності руху, світиться зелений світлодіод, зумер та червоний індикатор вимикаються, а на екрані відображається статус «SAFE».

Цикл роботи повторюється з невеликою паузою, що забезпечує стабільність показників та коректну роботу системи.

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

Після збирання пристрою було проведено перевірку коректності підключення всіх компонентів та тестування роботи програми.

5.1 Підготовка до тестування

У процесі випробувань встановлено:

- стабільну роботу датчика руху;
- точність вимірювання відстані в межах 1–2 см;
- коректне відображення інформації на дисплеї;
- надійну реакцію системи на появу руху.

5.2 Процес тестування

Після з'єднання плати Arduino Uno з персональним комп'ютером за допомогою USB-кабелю було здійснено компіляцію програмного коду та його завантаження у пам'ять мікроконтролера. У процесі перевірки працездатності системи було змодельовано її функціонування у двох основних режимах роботи.

- **Режим «Очікування» (відсутність руху):**
 - Система ініціалізується, на LCD дисплеї з'являється стартове повідомлення.
 - Через 2 секунди активується зелений світлодіод.

- На дисплеї відображається напис «Status: SAFE», звуковий сигнал відсутній.
- **Режим «Тривога» (імітація проникнення):**
 - У разі руху перед інфрачервоним датчиком HC-SR501 система миттєво переходить у режим тривоги.
 - При цьому зелений індикатор вимикається, а червоний світлодіод починає блимати.
 - Одночасно вмикається п'єзозумер, який подає переривчастий звуковий сигнал.
 - На рідкокристалічному дисплеї відображається повідомлення «Status: ALARM!», а в нижньому рядку в реальному часі виводиться відстань до виявленого об'єкта, визначена ультразвуковим сенсором HC-SR04 (наприклад, «Dist: 15 cm»).

5.3 Результати

У процесі проведення тестування було отримано такі результати:

- **Чутливість системи:** PIR-датчик стабільно виявляє рух у межах приміщення на відстані до 3 метрів, що відповідає заявленим характеристикам пристрою.
- **Точність вимірювань:** Ультразвуковий сенсор визначає відстань до об'єкта з похибкою не більше 1–2 см, що є достатнім для коректного функціонування охоронної системи.

- **Індикація:** Дані на LCD-дисплеї відображаються чітко та читабельно, контрастність налаштована оптимально. Перемикання світлодіодів (червоного та зеленого) повністю відповідає поточному режиму роботи системи.
- **Стабільність роботи:** Під час одночасної взаємодії датчиків та дисплея збоїв у роботі мікроконтролера не зафіксовано, що свідчить про надійність реалізованого рішення.

5.4 Висновки

У результаті виконання курсової роботи було створено працездатну охоронну систему на базі Arduino. Реалізовано взаємодію між датчиками, виконавчими елементами та засобами індикації.

Робота підтвердила можливість використання Arduino як ефективної платформи для побудови систем безпеки. Отримані результати демонструють практичну цінність проєкту та можливість його подальшого розвитку, зокрема шляхом додавання бездротового зв'язку, системи логування або автономного живлення.

ВИСНОВКИ

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

- **Апаратна частина:** Розглянуто та реалізовано принципи побудови систем безпеки з використанням різних типів сенсорів. Забезпечено коректну взаємодію пасивного інфрачервоного датчика руху HC-SR501 та ультразвукового датчика відстані HC-SR04. Застосування інтерфейсу I2C

для підключення LCD-дисплея дало змогу спростити електричну схему та зменшити кількість задіяних портів мікроконтролера;

- **Програмна частина:** Створено адаптивний програмний алгоритм мовою C++, що забезпечує опитування датчиків у режимі реального часу.

Використання бібліотек LiquidCrystal_I2C та Wire дозволило організувати коректне відображення інформації на дисплеї, а логічна обробка даних гарантує надійну реакцію системи на тривожні події;

- **Працездатність системи:** У ході тестування підтверджено відповідність роботи пристрою поставленим вимогам. Система коректно переходить із режиму очікування в режим тривоги при фіксації руху, активує світлову та звукову сигналізацію, а також відображає точну відстань до об'єкта на екрані;

- **Навчальна цінність:** Виконання роботи сприяло закріпленню знань з електроніки та схемотехніки, зокрема щодо підключення навантажень через транзисторні ключі й резистори. Також було поглиблено практичні навички роботи з протоколами передачі даних і налагодження програмного забезпечення.

5.5 Перспективи розвитку

Створений пристрій є початковою моделлю, яка має широкі можливості для подальшого вдосконалення та може бути адаптована до повноцінної системи «Розумний дім» або охоронного комплексу комерційного рівня. Основними напрямками розвитку є:

- **Дистанційне інформування:** Впровадження GSM-модуля (наприклад, SIM800L) або Wi-Fi модуля (ESP8266) для надсилання SMS-повідомлень чи push-сповіщень у месенджери користувача у разі спрацювання сигналізації;

- **Система контролю доступу:** Оснащення пристрою матричною клавіатурою (Keypad 4×4) або RFID-зчитувачем, що дозволить здійснювати постановку та зняття з охорони за допомогою коду чи електронної картки;

- **Автономне живлення:** Розробка системи резервного живлення з використанням літій-іонних акумуляторів та модуля заряджання, а також

оптимізація програмного забезпечення шляхом реалізації режимів енергозбереження для збільшення часу автономної роботи;

- **Журналювання подій:** Підключення модуля реального часу (RTC) разом із SD-картою для збереження інформації про всі спрацювання системи (дата, час, відстань до об'єкта) у вигляді лог-файлів;
- **Конструктивне вдосконалення:** Перехід від макетної плати до друкованої плати (PCB) та розробка захисного корпусу з використанням 3D-друку для підвищення надійності та захисту електронних компонентів від зовнішніх впливів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- **Блум Д.** *Вивчаємо Arduino: інструменти та методи технічного чарівництва* / Джеремі Блум. — Київ: БХВ, 2016. — 336 с.
(Посібник, присвячений роботі з платформою Arduino, її можливостям та практичному застосуванню.)
- **Шилдт Г.** *C++: базовий курс* / Герберт Шилдт. — 3-тє вид. — Москва: Вільямс, 2019. — 624 с.
(Навчальний матеріал з основ програмування мовою C++.)
- **Петренко А. І.** *Основи мікропроцесорної техніки та програмування мікроконтролерів* : навчальний посібник. — Київ: Каравела, 2020. — 288 с.
(Розглянуто принципи роботи мікроконтролерів та методи їх програмування.)
- **Рюмик С. М.** *Мікроконтролери AVR: практикум для початківців*. — Харків: Ранок, 2018. — 112 с.
(Практичний посібник для вивчення архітектури та програмування AVR-контролерів.)
- **Галкін В. І.** *Промислова електроніка та мікросхемотехніка* : підручник. — Київ: Вища школа, 2017. — 320 с.
(Опис основ промислової електроніки та принципів роботи мікросхем.)
- **Alvins B.** *Mastering Arduino Programming*. — 2021.
(Присвячено програмуванню мікроконтролерів та роботі з мовою C/C++.)
- **Schwartz M.** *Arduino Electronics Blueprints*. — 2019.
(Матеріал з інтеграції електронних компонентів, використання бібліотек і створення скетчів.)
- **Axelson D.** *Embedded Systems Design*. — 2020.
(Розглянуто фундаментальні принципи проектування вбудованих систем та їх архітектуру.)
- **Marr B.** *Internet of Things (IoT) in Business*. — 2022.
(Опис концепцій Інтернету речей, мережевих протоколів і стандартів для IoT-пристроїв.)

ДОДАТОК 1

Принципова електрична схема підключення