

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ «ФАХОВИЙ
КОЛЕДЖ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**

КУРСОВИЙ ПРОЄКТ

на тему:

**«Організація дистанційної передачі на модулі зв'язку 2.4 GHz даних
датчика температури і вологості на екран OLED-дисплея з
використанням плати Arduino»**

Виконав студент групи МТ-41

Вадим ЧОПКО

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

<i>Чопку Вадиму Олеговичу</i>
(прізвище, ім'я та по батькові)

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

Студент групи: МТ-41

1. Тема проєкту: Організація дистанційної передачі на модулі зв'язку 2.4 GHz даних датчика температури і вологості на екран OLED-дисплея з використанням плати Arduino

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			Вадим ЧОПКО
	(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	12
4 ПРОГРАМНА ЧАСТИНА	14
5 ТЕСТУВАННЯ ТА РЕЗУЛЬТАТИ	18
ВИСНОВКИ	19
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	20
ДОДАТОК 1	21

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 температури та вологості (DHT11/DHT22). У роботі використовується цифровий датчик серії DHT. Він складається з ємнісного датчика вологості та терморезистора (NTC). Усередині також знаходиться простий 8-бітний чіп, який перетворює аналогові сигнали в цифровий вихід.

Особливістю датчика є те, що він видає вже калібрований цифровий сигнал. Для обміну даними використовується однопровідний протокол (One-wire proprietary), що спрощує підключення, але вимагає суворих часових діаграм при зчитуванні, що реалізується програмними бібліотеками.

Модуль радіозв'язку NRF24L01+. Це трансівер, що працює на частоті 2.4 GHz (ISM діапазон). Він використовує інтерфейс SPI для спілкування з мікроконтролером.

Модуль дозволяє передавати дані зі швидкістю до 2 Мбіт/с. Він підтримує автоматичну перевірку цілісності пакетів (CRC) та автоматичне підтвердження прийому (ACK), що робить передачу даних дуже надійною. Це ідеальне рішення для телеметрії в межах приміщення.

OLED-дисплей (SSD1306). Використовується монохромний графічний дисплей на базі контролера SSD1306 з діагоналлю 0.96 дюйма. Технологія OLED (органічні світлодіоди) не потребує підсвічування, оскільки пікселі світяться самі. Це забезпечує високий контраст та низьке енергоспоживання. Підключення здійснюється по шині I2C (всього два дроти: SDA та SCL), що економить піни мікроконтролера.

2 ТЕХНІЧНЕ ЗАВДАННЯ

2.1 Назва роботи

Організація дистанційної передачі на модулі зв'язку 2.4 GHz даних датчика температури і вологості на екран OLED-дисплея з використанням плати Arduino.

2.2 Мета роботи

Розробка та реалізація апаратно-програмного комплексу для бездротового моніторингу кліматичних параметрів (температури та вологості) у режимі реального часу з використанням радіоканалу частотою 2.4 ГГц.

2.3 Завдання

- Провести аналіз існуючих рішень та компонентної бази для бездротової передачі даних.
- Розробити електричну схему підключення датчиків та модулів до мікроконтролера Arduino.
- Реалізувати програмний алгоритм зчитування даних з сенсора DHT11/DHT22.
- Налаштувати радіообмін між двома вузлами (передавач та приймач) за допомогою модулів NRF24L01.
- Реалізувати виведення отриманих даних на графічний OLED-дисплей.
- Провести тестування системи на стабільність роботи та дальність передачі сигналу.

2.4 Технічні вимоги

- 5 В постійного струму;
- Arduino Uno R3;
- Breadboard (безпайкова макетна плата);
- Частота 2.400 – 2.525 GHz.;
- Графічний OLED дисплей 128x64 пікселів;
- Електролітичний конденсатор 10-100 мкФ;
- З'єднувальні провідники типу «тато-тато» та «тато-мама».

2.5 Очікуваний результат

В результаті виконання роботи буде отримано повністю функціональний прототип бездротової метеостанції. Після подачі живлення на обидва модулі (передавач і приймач) система автоматично входить у робочий режим.

- **На стороні передавача:** Світлодіод на платі Arduino (L) може короткочасно блимати під час відправки пакету (якщо це передбачено налагоджувальним кодом, хоча у фінальній версії це не обов'язково). Датчик зчитує реальні параметри кімнати.
- **На стороні приймача:** Екран OLED миттєво ініціалізується. Спочатку він може бути порожнім або показувати нулі, доки не надійде перший пакет даних.
- **Динаміка роботи:** Кожну секунду (1000 мс) дані на дисплеї оновлюються. Якщо подути на датчик вологості, значення на дисплеї приймача "Н: xx %" почне зростати майже миттєво. Якщо нагріти датчик рукою, зміниться значення "Т: xx C".

- **Стабільність:** Система повинна працювати стабільно на відстані 10-15 метрів у приміщенні (через 1-2 стіни). При втраті зв'язку дисплей буде показувати останні отримані значення, доки не з'явиться новий сигнал.

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 проекту

Таблиця 3.1.1 Перелік необхідних компонентів

№	Назва компонента	Кількість	Позначення на схемі	Характеристики	Примітка
1	Arduino Uno	2	U1, U2	Мікроконтролер ATmega328P	Основна плата керування
2	Плата breadboard	2	-	830 точок, безпайкова	Безпайна плата для створення конструкції
3	Модуль NRF24L01+	2	RF1, RF2	2.4GHz, SPI інтерфейс, 3.3V	Бездротова передача/прийом даних
4	Датчик DHT11	1	DHT	20-90% RH, 0-50°C	Вимірювання температури та вологості
5	OLED Дисплей	1	DISP	0.96", 128x64, I2C (SSD1306)	Візуалізація отриманих даних
6	Дроти-джампери	20	-	5 мм, 2 В, 20 мА	Електричне з'єднання компонентів

Таблиця 3.1.2 Таблиця з'єднань пінів

№	Компонент	Вивід компонента	З'єднано з піном Arduino	Примітка / Функція
1	NRF24L01	VCC	3.3V	Живлення
		CE	Digital 9	Сигнал запуску
		CSN	Digital 10	Прийом сигналу
		GND	GND	Земля
		SCK	Digital 13	Serial Clock (тактування SPI)
		MOSI	Digital 11	Master Out Slave In (дані до модуля)
		MISO	Digital 12	Master In Slave Out (дані від модуля)
2	DHT11	VCC	5V	Живлення
		OUT	Digital 2	Вихід цифрового сигналу
		GND	GND	Земля
3	OLED	GND	GND	Земля
		VCC	5V	Живлення
		SCL	Pin A5 (або SCL)	Шина тактування I2C
		SDA	Pin A4 (або SDA)	Шина даних I2C

4 ПРОГРАМНА ЧАСТИНА

Програмне забезпечення розроблено в середовищі Arduino IDE. Логіка роботи системи розділена на два незалежних скетчі: для передавача та приймача.

Для роботи програми використовуються сторонні бібліотеки:

1. **RF24.h** — для керування радіомодулем. Вона забезпечує ініціалізацію SPI-інтерфейсу, налаштування каналу передачі, потужності сигналу та довжини адреси.
2. **DHT.h** — для взаємодії з датчиком вологості та температури. Бібліотека виконує перетворення сигналів з цифрового піна у зрозумілі значення температури (Celsius) та вологості (%).
3. **Adafruit_GFX.h** та **Adafruit_SSD1306.h** — для роботи з OLED дисплеєм. Вони дозволяють виводити текст, числа та графічні примітиви.

Алгоритм роботи передавача: Ініціалізація датчика та радіомодуля відбувається у блоці `setup()`. У нескінченному циклі `loop()` мікроконтролер зчитує дані з датчика. Для забезпечення цілісності даних при передачі використовується структура (`struct`), що містить дві змінні типу `float`: температура та вологість. Ця структура "пакується" в пакет байтів і відправляється через метод `radio.write()`. Після цього відбувається затримка (наприклад, 1-2 секунди) для енергозбереження та стабільності датчика.

Алгоритм роботи приймача: Приймач постійно сканує радіоефір на наявність вхідних пакетів за допомогою функції `radio.available()`. Якщо пакет отримано, він зчитується у ідентичну структуру даних. Отримані значення передаються на OLED-дисплей через функцію `display.print()`, попередньо очистивши екран від старих даних.

4.1 Код програми (Мікроконтролер 1. Відправник)

```
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
#include <DHT.h>

#define DHTPIN 2
#define DHTTYPE DHT11

RF24 radio(9, 10);
DHT dht(DHTPIN, DHTTYPE);

const byte address[6] = "00001";

struct Data_Package {
    float temperature;
    float humidity;
};

Data_Package data;

void setup() {
    radio.begin();
    radio.openWritingPipe(address);
    radio.setPALevel(RF24_PA_MIN);
    radio.stopListening();
    dht.begin();
}

void loop() {
    data.humidity = dht.readHumidity();
    data.temperature = dht.readTemperature();
```



```

radio.write(&data, sizeof(Data_Package));

delay(1000);
}

```

4.2 Код програми (Мікроконтролер 2. Отримувач)

```

#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT,
&Wire, OLED_RESET);
RF24 radio(9, 10);

const byte address[6] = "00001";

struct Data_Package {
    float temperature;
    float humidity;
};

Data_Package data;

void setup() {
    display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
    display.clearDisplay();
}

```

```
radio.begin();
radio.openReadingPipe(0, address);
radio.setPALevel(RF24_PA_MIN);
radio.startListening();
}

void loop() {
  if (radio.available()) {
    radio.read(&data, sizeof(Data_Package));

    display.clearDisplay();
    display.setTextSize(1);
    display.setTextColor(WHITE);

    display.setCursor(0, 0);
    display.println("REMOTE SENSOR DATA:");

    display.setTextSize(2);
    display.setCursor(0, 20);
    display.print("T: ");
    display.print(data.temperature);
    display.print(" C");

    display.setCursor(0, 45);
    display.print("H: ");
    display.print(data.humidity);
    display.print(" %");

    display.display();
  }
}
```

5 ТЕСТУВАННЯ ТА РЕЗУЛЬТАТИ

Тестування системи проводиться у кілька етапів для перевірки надійності апаратної та програмної частин.

Етап 1: Перевірка сенсорів. Спочатку перевіряється коректність показників датчика DHT11 через Serial Monitor без використання радіоканалу. Порівнюються показники датчика з еталонним побутовим термометром. *Результат:* Відхилення температури склало $\pm 1^{\circ}\text{C}$, що в межах норми.

Етап 2: Тестування радіоканалу. Перевірка "пінгу" між модулями. На передавачі відправляється лічильник пакетів, на приймачі перевіряється, чи не губляться пакети. *Результат:* При розташуванні модулів поруч втрат пакетів немає (0%).

Етап 3: Тестування дальності. Модуль передавача встановлено стаціонарно. Модуль приймача переміщується по приміщенню.

- Відстань 1-5 метрів (пряма видимість): Прийом стабільний.
- Відстань 10 метрів (одна цегляна стіна): Прийом стабільний.
- Відстань 15+ метрів (дві стіни): Спостерігаються затримки оновлення даних, можлива втрата пакетів.

Етап 4: Інтеграційне тестування. Перевірка повної системи: зчитування -> передача -> відображення. Симулюється зміна умов (нагрівання датчика). *Результат:* Дисплей відображає зміни з затримкою не більше 1 секунди після циклу вимірювання. Текст на OLED чіткий, мерехтіння відсутнє.

ВИСНОВКИ

У ході виконання курсової роботи було успішно розроблено та протестовано систему дистанційної передачі даних про мікроклімат. Було вирішено задачу організації бездротового зв'язку між двома мікроконтролерами Arduino за допомогою модулів NRF24L01 на частоті 2.4 ГГц.

Основні досягнення:

- Освоєно роботу з протоколом SPI та бібліотекою RF24.
- Налаштовано взаємодію з I2C OLED дисплеєм.
- Створено модульний код, який легко масштабувати.

Система показала свою ефективність для побутового використання. Вона є дешевою альтернативою комерційним метеостанціям.

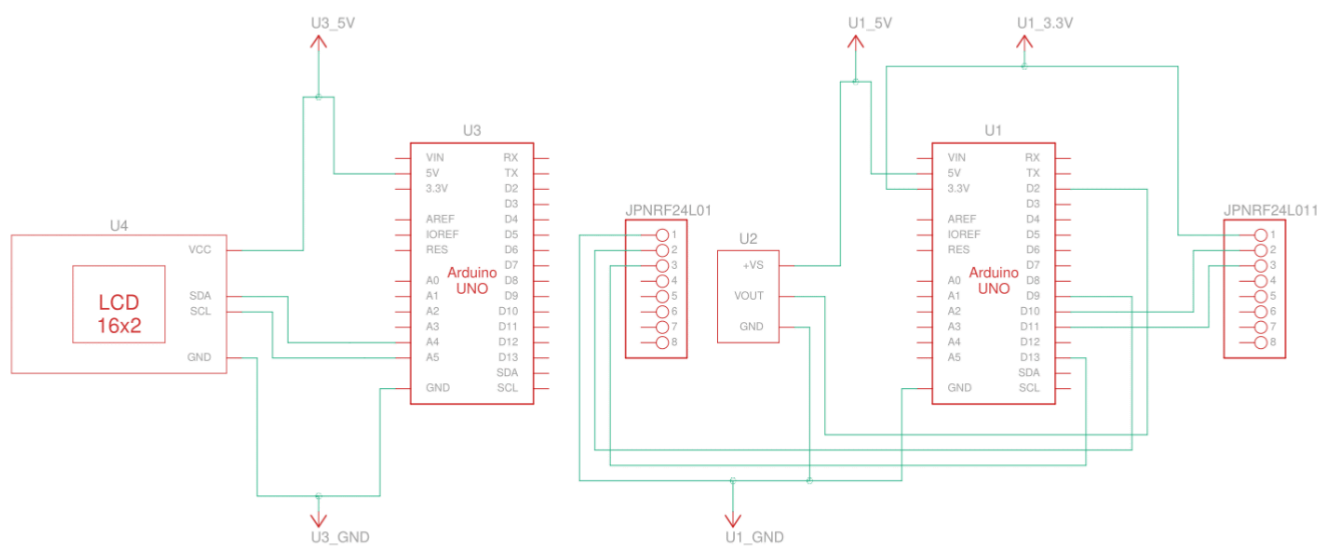
Перспективи розвитку:

1. **Енергоефективність:** Впровадження режиму глибокого сну (Deep Sleep) для мікроконтролера передавача, що дозволить працювати від батарейок кілька місяців.
2. **Розширення функціоналу:** Додавання датчика тиску (BMP180/BMP280) або датчика якості повітря (MQ-135).
3. **IoT інтеграція:** Заміна приймача Arduino на ESP8266/ESP32 для передачі даних не тільки на екран, а й у веб-сервер або мобільний додаток через Wi-Fi.
4. **Зворотний зв'язок:** Додавання реле на приймач для автоматичного включення зволожувача повітря, якщо вологість падає нижче заданого рівня.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- **Блум Д. Вивчаємо Arduino:** інструменти та методи технічного чарівництва / Джеремі Блум. — К.: БХВ, 2016. — 336 с.
- **Шилдт Г. C++:** 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