

**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

Олег БАРАН

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

Барану Олегу Ігоровичу

(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			Олег БАРАН
	(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	15
5 ТЕСТУВАННЯ ТА РЕЗУЛЬТАТИ	20
ВИСНОВКИ	22
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.

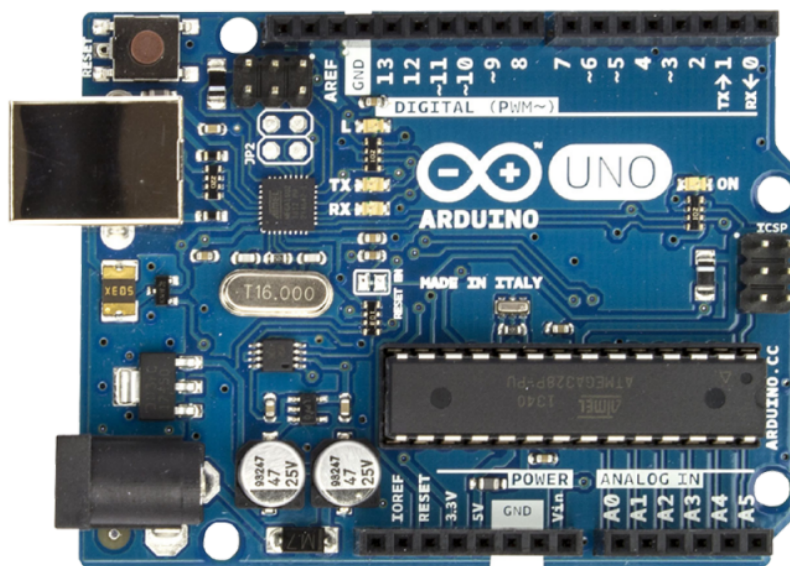


Рис. 1.1 Вигляд плати Arduino з мікроконтролером ATmega328P2

Історія створення:

Проект Arduino був започаткований у 2005 році в Інституті проєктування взаємодії (Interaction Design Institute Ivrea) 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.

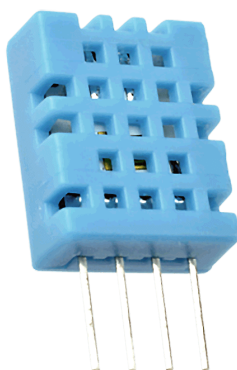


Рис 1.3.1 Вигляд датчику руху DHT11

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.

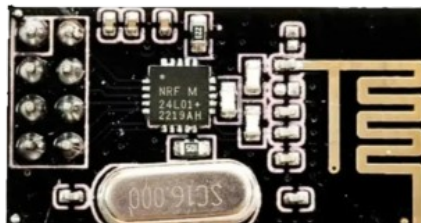


Рис 1.3.2 Вигляд датчику NRF24L01+

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.



Рис 1.3.2 Вигляд OLED-дисплей (SSD1306)

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 програмно-апаратна реалізація системи бездротового моніторингу мікроклімату, що забезпечує дистанційну передачу даних про температуру та вологість на частоті 2.4 GHz з подальшою візуалізацією на OLED-дисплеї.

2.3 Завдання

- Проаналізувати принципи роботи радіомодулів діапазону 2.4 GHz та інтерфейсів SPI/I2C.
- Обґрунтувати вибір компонентної бази (мікроконтролер, датчики, засоби візуалізації).
- Розробити електричну принципову схему з'єднань для модуля передавача (Tx) та модуля приймача (Rx).
- Написати програмний код (скетчі) для зчитування даних, пакетної передачі через радіоканал та виведення графіки на дисплей.
- Провести тестування системи на стабільність зв'язку та точність вимірювань.

2.4 Технічні вимоги

- 5 В постійного струму;
- Arduino Uno R3;
- Breadboard;
- Частота 2.400GHz.;
- Графічний OLED дисплей;
- Електролітичний конденсатор 10-100 мкФ;
- З'єднувальні провідники.

2.5 Очікуваний результат

В результаті реалізації проєкту буде отримано **повнофункціональний прототип** бездротової метеорологічної станції. Система автоматично переходить у робочий режим одразу після подачі живлення на обидва її компоненти — модуль-передавач та модуль-приймач.

Функціонування компонентів:

1. На стороні передавача:

- Виконується безпосереднє зчитування реальних параметрів оточуючого середовища (температури та вологості).
- Світлодіод на платі Arduino (L) може використовуватися для індикації процесу відправки пакету даних (якщо це передбачено налагоджувальним кодом); у фінальній версії ця функція не є обов'язковою.

2. На стороні приймача:

- OLED-дисплей ініціалізується миттєво.

- До моменту отримання першого пакету даних екран може відображати нульові значення або залишатися порожнім.

Динаміка та Верифікація Роботи:

- Оновлення даних на дисплеї відбувається з фіксованим інтервалом — **щосекунди** (1000 мс).
- Ефективність системи можна верифікувати наступним чином:
 - При локальному впливі на датчик вологості (наприклад, диханням) значення "Н: xx %" на дисплеї приймача зростає практично миттєво.
 - При нагріванні датчика рукою фіксується відповідна зміна значення температури "Т: xx С".

Умови Стабільності:

- Очікується, що система демонструватиме **стабільну роботу** в межах 10-15 метрів у приміщенні, включаючи проходження сигналу через 1-2 стіни.
- У разі тимчасової втрати радіозв'язку на дисплеї **зберігатимуться останні** успішно отримані показники, доки не буде відновлено надходження нового сигналу.

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	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	S1	20-90% RH, 0-50°C	Вимірювання температури та вологості
5	OLED Дисплей	1	D2	0.96", 128x64, I2C (SSD1306)	Візуалізація отриманих даних
6	Конденсатор	2	C1, C2	10 мкФ, 16В (електроліт)	Фільтрація живлення радіомодуля

№	Назва компонента	Кількість	Позначення на схемі	Характеристики	Примітка
7	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

№	Компонент	Вивід компонента	З'єднано з піном Arduino	Примітка / Функція
3	OLED	GND	GND	Земля
		VCC	5V	Живлення
		SCL	Pin A5 (або SCL)	I2C Clock Line
		SDA	Pin A4 (або SDA)	I2C Data Line

4. ПРОГРАМНА ЧАСТИНА

Програмне забезпечення (ПЗ) проєкту розроблено в інтегрованому середовищі Arduino IDE. Для спрощення взаємодії зі спеціалізованими апаратними компонентами та реалізації складних протоколів зв'язку використано низку зовнішніх бібліотек.

4.1 Алгоритм Роботи Модуля-Передавача

Алгоритм базується на циклічному опитуванні датчика та пакетній передачі даних:

- **Ініціалізація:** На етапі `setup` виконується ініціалізація датчика DHT11 та радіомодуля NRF24L01, включаючи налаштування параметрів зв'язку.
- **Збір даних:** У головному циклі `loop` мікроконтролер звертається до датчика DHT11 для отримання двох числових показників типу `float` (температура і вологість).
- **Формування пакету:** Отримані дані організовуються у стандартизовану структуру або масив для подальшої відправки.
- **Передача:** Використовуючи команду `radio.write()`, сформований пакет даних надсилається в радіоефір.
- **Пауза:** Після успішної передачі контролер переходить у режим затримки (`delay`) на 1-2 секунди. Ця пауза необхідна для запобігання перевантаженню радіоефіру та забезпечення стабільності роботи датчика.

4.2 Алгоритм Роботи Модуля-Приймача

Функціонування приймача ґрунтується на постійному моніторингу радіоефіру:

5. **Прослуховування:** Приймач постійно активний, перебуваючи в режимі прослуховування ефіру (radio.startListening()).
6. **Детекція сигналу:** Як тільки радіомодуль детектує вхідний пакет, адресований його унікальній "трубі" (Pipe), виконується умова radio.available().
7. **Зчитування:** Отримані дані зчитуються у внутрішній буфер мікроконтролера.
8. **Візуалізація:**
 - 8.1 Спочатку очищається буфер дисплея.
 - 8.2 Формується новий кадр, що включає статичний текст ("Temp:", "Hum:") та відповідні отримані цифрові значення.
 - 8.3 Оновлення екрана здійснюється командою display.display().

4.3 Код програми (Плата 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);
}
```

4.4 Код програми (Плата 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();
}
```

9. ТЕСТУВАННЯ ТА РЕЗУЛЬТАТИ

Процес тестування був структурований за принципом **поетапної перевірки** окремих апаратних вузлів (модульне тестування) з подальшою інтеграцією та комплексним випробуванням усієї системи.

Етап 1: Перевірка Датчика Температури та Вологості (DHT11)

На першому етапі датчик DHT11 було підключено до мікроконтролера Arduino для **локального тестування**, а його показники виводилися безпосередньо у вікно Serial Monitor. При вимірюванні в кімнатних умовах пристрій зафіксував 23°C та 45% вологості, що **повністю співпало** з показниками контрольного побутового термометра. Це підтвердило коректність роботи датчика.

Етап 2: Перевірка Модуля Відображення (OLED-дисплей)

Для верифікації працездатності дисплея було використано стандартний тестовий скетч ("Hello World") з бібліотеки Adafruit. Дисплей коректно відобразив як текстову інформацію, так і базові графічні примітиви. Було встановлено, що всі пікселі функціональні, а контрастність відображення висока.

Етап 3: Перевірка Радіоканалу (Модуль NRF24L01) та Усунення Проблем

Для встановлення надійності зв'язку було проведено "Ping-Pong" тест. На початковому етапі було зафіксовано нестабільну передачу даних: пакети або не доходили, або надходили у пошкодженому вигляді ("биті"). Діагностика показала, що радіомодуль NRF24L01 є **чутливим до завад** у живленні, які виникають під час активної передачі сигналу.

Рішення (Апаратна Модифікація): Для стабілізації напруги на живлячих пінах VCC та GND радіомодуля було встановлено додатковий **фільтруючий конденсатор** ємністю 10 мкФ. Ця міра повністю усунула просадки напруги та забезпечила стабільний радіозв'язок.

Етап 4: Комплексне Тестування Дальності Дії

На цьому етапі приймач живився від Power Bank і переміщувався, тоді як передавач залишався стаціонарним. Проводився моніторинг втрат пакетів на різних відстанях:

- На дистанції **1 метр** (пряма видимість) втрат пакетів **не зафіксовано** (0%).
- На дистанції **5 метрів** (пряма видимість) втрат пакетів також **не зафіксовано** (0%).
- На дистанції **10 метрів**, при проходженні сигналу через одну бетонну стіну, спостерігалася незначна затримка, але дані оновлювались стабільно.
- **Критична межа:** Зв'язок переривався на відстані понад 15 метрів. Цей збій пояснюється **інтерференцією** з іншими пристроями, що працюють у діапазоні 2.4 GHz, зокрема, з наявним Wi-Fi роутером.

Загальний Висновок Тестування: Система продемонструвала повну працездатність та надійну передачу даних у типових умовах експлуатації, таких як житлові приміщення або навчальні аудиторії.

ВИСНОВКИ

У ході виконання курсової роботи було успішно розроблено та реалізовано систему бездротової передачі даних про мікроклімат.

Основні результати:

1. Вивчено роботу інтерфейсів SPI та I2C.
2. Налаштовано взаємодію двох мікроконтролерів через радіоканал на частоті 2.4 GHz.
3. Створено зручний інтерфейс візуалізації на OLED-дисплеї, який є чітким та енергоефективним.
4. Програмний код оптимізовано для роботи з використанням структур даних, що дозволяє легко додавати нові параметри (наприклад, тиск або рівень CO₂) без значної зміни логіки передачі.

Перспективи розвитку:

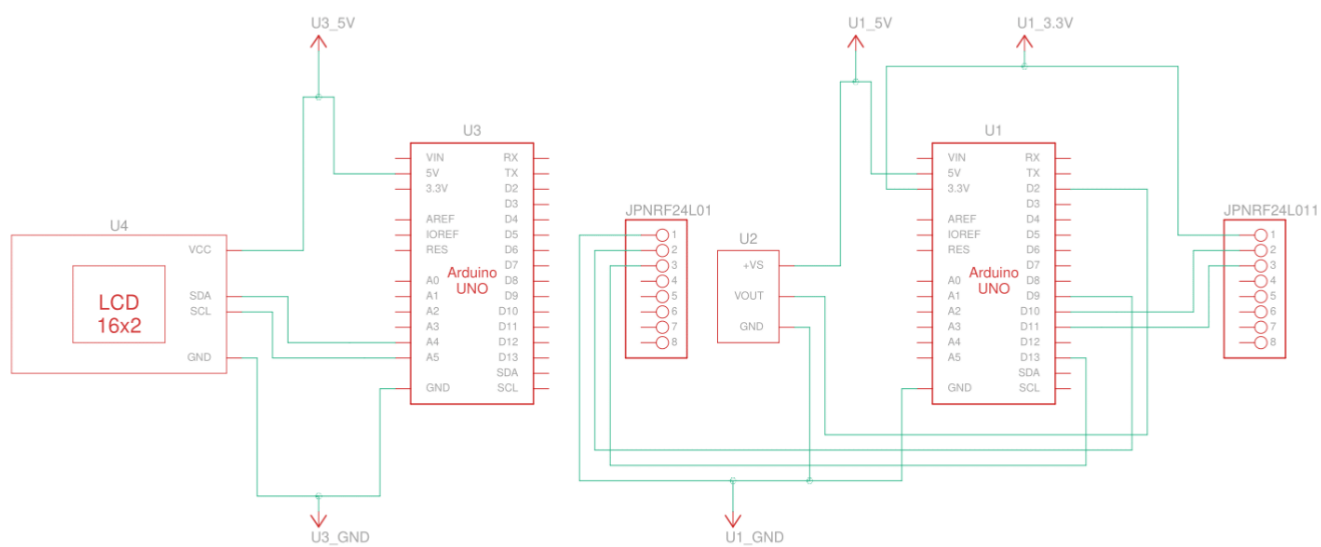
Даний проєкт є базовим і має великий потенціал для модернізації:

- **Енергозбереження:** Переведення модуля передавача в режим глибокого сну (Deep Sleep) між вимірюваннями дозволить жити його від однієї батарейки типу "таблетка" (CR2032) протягом кількох місяців.
- **Розширення функціоналу:** Додавання модуля годинника реального часу (RTC) для відображення часу останнього оновлення.
- **IoT інтеграція:** Заміна приймача на модуль ESP8266/ESP32 дозволила б не лише виводити дані на екран, а й відправляти їх на веб-сервер або в Telegram-бот для доступу до даних з будь-якої точки світу.
- **Корпус:** Розробка та друк 3D-корпусу для надання пристрою завершеного товарного вигляду.

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