

**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 ІП-42

Софія КУРИЛЯК

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

***Курилюк Софії Сергіївни***

(прізвище, ім'я та по батькові)

з навчальної дисципліни: ПРОГРАМНО-АПАРАТНІ ЗАСОБИ INTERNET-РЕЧЕЙ

Студент групи: **ІІІ-42**

**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:** \_\_\_\_\_ "14" жовтня 2025 р.

**3. Термін здачі курсового проєкту:** \_\_\_\_\_ "09" 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 WOM (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 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**

| 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 )

**Софія КУРИЛЮК**

\_\_\_\_\_  
( імя та прізвище )

**Керівник проекту**

\_\_\_\_\_  
( підпис )

**Остап ЮНАК**

\_\_\_\_\_  
( імя та прізвище )

## **ЗМІСТ**

|                                    |           |
|------------------------------------|-----------|
| <b>ВСТУП</b>                       | <b>3</b>  |
| <b>1 ОГЛЯД ЛІТЕРАТУРИ/АНАЛОГІВ</b> | <b>5</b>  |
| <b>2 ТЕХНІЧНЕ ЗАВДАННЯ</b>         | <b>9</b>  |
| <b>3 АПАРАТНА ЧАСТИНА</b>          | <b>11</b> |
| <b>4 ПРОГРАМНА ЧАСТИНА</b>         | <b>14</b> |
| <b>5 ТЕСТУВАННЯ ТА РЕЗУЛЬТАТИ</b>  | <b>20</b> |
| <b>ВИСНОВКИ</b>                    | <b>23</b> |
| <b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ</b>  | <b>25</b> |
| <b>ДОДАТКИ</b>                     | <b>26</b> |

## 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.** Принцип дії датчика DHT11 ґрунтується на вимірюванні температури та відносної вологості повітря за допомогою вбудованих сенсорних елементів. Для визначення вологості використовується резистивний гігроскопічний елемент, опір якого змінюється залежно від кількості водяної пари в повітрі.

Температура вимірюється за допомогою термістора. Всі виміряні дані обробляються внутрішнім мікроконтролером датчика та передаються у цифровому вигляді, що забезпечує просте підключення і використання в електронних системах.

**Модуль бездротового зв'язку NRF24L01.** Принцип дії модуля NRF24L01 ґрунтується на передачі та прийомі цифрових даних за допомогою радіохвиль у діапазоні 2,4 ГГц. Модуль використовує вбудований радіотрансивер, який перетворює електричні сигнали мікроконтролера в радіосигнали для бездротової передачі, а також виконує зворотне перетворення під час прийому. Обмін даними здійснюється через інтерфейс SPI, що забезпечує високу швидкість і надійність зв'язку. NRF24L01 підтримує адресацію, перевірку помилок і автоматичне підтвердження прийому даних, завдяки чому широко застосовується в системах бездротового керування та обміну інформацією.

**OLED-дисплей 0.96" (I2C, 128×64).** Принцип дії OLED-дисплея (Organic Light-Emitting Diode) базується на випромінюванні світла органічними світлодіодами при проходженні через них електричного струму. Кожен піксель дисплея світиться самостійно, тому підсвітка не потрібна, що забезпечує високий контраст і низьке енергоспоживання. Дисплей має роздільну здатність 128×64 пікселі та керується через інтерфейс I2C, який дозволяє передавати дані лише по двох сигнальних лініях. Він використовується для відображення текстової та графічної інформації в мікроконтролерних системах.

## **2 ТЕХНІЧНЕ ЗАВДАННЯ**

### **2.1 Назва роботи**

Виготовлення пристрою дистанційної передачі на модулі зв'язку NRF24L01 2.4 GHz даних датчика температури і вологості DHT11 на екран OLED-дисплея з використанням плати Arduino.

### **2.2 Мета роботи**

Виготовлення пристрою дистанційної передачі на модулі зв'язку NRF24L01 2.4 GHz даних датчика температури і вологості DHT11 на екран OLED-дисплея з використанням плати Arduino.

### **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);
- LED-дисплей 0.96" (I2C, 128×64);
- NRF24L01;
- DHT11;

- Резистори 220 Ом;
- З'єднувальні провідники типу «тато-тато» та «тато-мама».

## 2.5 Очікуваний результат

Організація дистанційної передачі даних базується на використанні модуля бездротового зв'язку NRF24L01, який працює у діапазоні 2,4 ГГц. Датчик температури та вологості повітря (DHT11), підключений до плати Arduino, здійснює безперервні вимірювання параметрів навколишнього середовища та передає отримані значення до мікроконтролера. У подальшому Arduino формує цифровий пакет даних і через інтерфейс SPI надсилає його на радіомодуль для бездротової передачі. Приймаюча частина системи, оснащена аналогічним модулем зв'язку, приймає інформацію, обробляє її та виводить показники температури і вологості на OLED-дисплей у реальному часі. Такий підхід забезпечує стабільну дистанційну передачу даних і наочне відображення інформації без використання дротових з'єднань.

### 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         | A1 / A2             | Мікроконтролер ATmega328P       | Основна плата керування                                                                                            |
| 2 | Плата breadboard | 1         | -                   | 830 точок, безпайкова           | Безпайна плата для створення конструкції                                                                           |
| 3 | NRF24L01         | 1         | U1\U3               | 2.4 ГГц RF модуль               | VDD, VDD_PA → 3.3V,<br>GND(VSS) → GND, MOSI → D11, MISO → D12, SCK → D13, CSN → D10, CE → D9, IRQ → не підключений |
| 4 | DHT11            | 1         | U2                  | Датчик температури та вологості | VDD → 5V,<br>DATA → A1,<br>GND → GND                                                                               |

| №  | Назва компонента   | Кількість | Позначення на схемі | Характеристики                  | Примітка                                      |
|----|--------------------|-----------|---------------------|---------------------------------|-----------------------------------------------|
| 5  | OLED-дисплей 0.96" | 1         | DS1                 | 128×64, I2C                     | VCC → 5V,<br>GND → GND,<br>SDA → A4, SCL → A5 |
| 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 | DHT11     | VCC              | 5V                       | Живлення датчика     |
|   |           | DATA             | Digital Pin 2            | Лінія передачі даних |
|   |           | GND              | GND                      | Земля                |
| 2 | NRF24L01  | VCC              | 3.3V                     | Живлення             |
|   |           | CE               | Digital Pin 9            | Увімкнення модуля    |
|   |           | CSN              | Digital Pin 10           | Вибір SPI-пристрою   |
|   |           | MOSI             | Digital Pin 11           | SPI — передача даних |

| № | Компонент        | Вивід компонента | З'єднано з піном Arduino | Примітка / Функція    |
|---|------------------|------------------|--------------------------|-----------------------|
|   |                  | MISO             | Digital Pin 12           | SPI — прийом даних    |
|   |                  | SCK              | Digital Pin 13           | SPI — тактовий сигнал |
|   |                  | GND              | GND                      | Земля                 |
| 3 | OLED 0.96” (I2C) | GND              | GND                      | Земля                 |
|   |                  | VCC              | 5V                       | Живлення              |
|   |                  | SCL              | Pin A5 (SCL)             | Лінія тактування I2C  |
|   |                  | SDA              | Pin A4 (або SDA)         | Шина даних I2C        |

## 4 ПРОГРАМНА ЧАСТИНА

Розробка програмного забезпечення для мікроконтролера Arduino Uno здійснювалася в інтегрованому середовищі Arduino IDE.

Мова програмування — C++, з використанням стандартних бібліотек Arduino та спеціалізованих бібліотек для роботи з бездротовим зв'язком, датчиками та OLED-дисплеєм.

Програмне забезпечення складається з двох окремих скетчів:

- передавач (Arduino + DHT11 + NRF24L01),
- приймач (Arduino + NRF24L01 + OLED-дисплей).

Передача даних між модулями здійснюється бездротовим каналом 2.4 ГГц за допомогою модулів NRF24L01.

### Використані бібліотеки

Для передавача:

- **SPI.h** — забезпечує роботу по шині SPI;
- **nRF24L01.h**, **RF24.h** — керування бездротовим модулем NRF24L01;
- **DHT.h** — зчитування температури та вологості з датчика DHT11.

Для приймача:

- **SPI.h** — робота з SPI;  
**nRF24L01.h**, **RF24.h** — прийом даних по радіоканалу;
- **Wire.h** — робота з шиною I2C;

- `Adafruit_GFX.h`, `Adafruit_SSD1306.h` — керування OLED-дисплеєм 128×64.

## Опис структури програми

Програма структурно поділена на три основні блоки:

1. Підключення бібліотек та оголошення змінних;
2. Ініціалізація апаратних модулів (функція `setup()`);
3. Основний цикл роботи (`loop()`).

## Опис програмної частини (передавач)

### 1. Оголошення констант і змінних

Для датчика температури та вологості визначено:

- пін підключення `DHTPIN = 2`,
- тип датчика `DHT11`.

Для роботи з NRF24L01 створюється об'єкт:

- `RF24 radio(9, 10)`, де  
9 — CE, 10 — CSN.

Оголошено структуру `DataPacket`, яка містить:

- `temperature` — температура повітря;
- `humidity` — вологість повітря.

Ця структура використовується для передачі даних по радіоканалу.

### 2. Ініціалізація (функція `setup()`)

Під час запуску програми:

- ініціалізується послідовний порт для виводу діагностичних повідомлень;
- запускається робота датчика DHT11;
- ініціалізується модуль NRF24L01;
- налаштовується адреса каналу зв'язку;
- модуль переводиться у режим передавача (`stopListening()`).

### 3. Основний цикл роботи (функція `loop()`)

У нескінченному циклі виконується:

1. Зчитування температури та вологості з датчика DHT11;
2. Перевірка коректності отриманих даних;
3. Запис значень у структуру `DataPacket`;
4. Передача структури по бездротовому каналу;
5. Вивід значень температури та вологості в Serial Monitor;
6. Затримка 2 секунди для стабілізації вимірювань.

## Опис програмної частини (приймач)

### 1. Оголошення констант і змінних

Для OLED-дисплея визначено:

- роздільну здатність 128×64 пікселів;
- I2C-адресу дисплея `0x3C`.

Для NRF24L01 використовується той самий канал зв'язку та адреса, що й у передавачі.

Також використовується структура `DataPacket` для прийому температури та вологості.

## 2. Ініціалізація (функція `setup()`)

При запуску:

- ініціалізується OLED-дисплей;
- налаштовуються параметри відображення тексту;
- запускається модуль NRF24L01;
- модуль переводиться у режим приймача (`startListening()`).

## 3. Основний цикл роботи (функція `loop()`)

У процесі роботи:

1. Перевіряється наявність прийнятих даних;  
Зчитується структура `DataPacket`;
2. Очищується дисплей;
3. На OLED виводиться:
  - заголовок системи,
  - значення температури,
  - значення вологості;
4. Дані дублюються у Serial Monitor для контролю.

## Загальний алгоритм роботи системи

- 5 Датчик DHT11 вимірює температуру та вологість;  
Arduino-передавач передає дані через NRF24L01;
- 6 Arduino-приймач приймає інформацію по радіоканалу;
- 7 Отримані дані відображаються на OLED-дисплеї в реальному часі.

## 7.1 Код програми

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

#define DHTPIN 2
#define DHTTYPE DHT11

RF24 radio(9, 10); // CE, CSN
const byte address[6] = "00001";

DHT dht(DHTPIN, DHTTYPE);

struct DataPacket {
    float temperature;
    float humidity;
};

DataPacket data;

void setup() {
    Serial.begin(9600);
    dht.begin();

    radio.begin();
```

```

radio.openWritingPipe(address);
radio.setPALevel(RF24_PA_LOW);
radio.stopListening();
}

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

    if (isnan(data.humidity) || isnan(data.temperature)) {
        Serial.println("Помилка зчитування DHT11");
        return;
    }

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

    Serial.print("Temp: ");
    Serial.print(data.temperature);
    Serial.print(" °C | Hum: ");
    Serial.print(data.humidity);
    Serial.println(" %");

    delay(2000);
}

#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

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
-1);

RF24 radio(9, 10); // CE, CSN
const byte address[6] = "00001";

struct DataPacket {
    float temperature;
    float humidity;
};

DataPacket data;

void setup() {
    Serial.begin(9600);

    display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
    display.clearDisplay();
    display.setTextSize(1);
    display.setTextColor(SSD1306_WHITE);
```

```
radio.begin();
radio.openReadingPipe(0, address);
radio.setPALevel(RF24_PA_LOW);
radio.startListening();
}

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

    display.clearDisplay();
    display.setCursor(0, 0);
    display.println("Wireless Monitor");

    display.setCursor(0, 20);
    display.print("Temp: ");
    display.print(data.temperature);
    display.println(" C");

    display.setCursor(0, 35);
    display.print("Hum: ");
    display.print(data.humidity);
    display.println(" %");

    display.display();

    Serial.print("Temp: ");
    Serial.print(data.temperature);
```

```
Serial.print(" | Hum: ");  
Serial.println(data.humidity);  
}  
}
```

## ТЕСТУВАННЯ ТА РЕЗУЛЬТАТИ

Перевірка працездатності розробленого пристрою **бездротового моніторингу температури та вологості** проводилася у два етапи:

перевірка коректності монтажу апаратної частини та функціональне тестування програмного забезпечення з передачею даних по радіоканалу.

### Підготовка до тестування

Перед подачею живлення було виконано візуальний огляд електричних схем та макетної плати з метою виявлення можливих помилок монтажу:

- перевірено правильність підключення датчика **DHT11** (відповідність виводів VCC, DATA та GND);
- перевірено коректність підключення модулів **NRF24L01** по шині SPI (лінії MOSI, MISO, SCK, CE, CSN);
- переконалися у правильності підключення **OLED-дисплея** по інтерфейсу I2C (лінії SDA та SCL);
- перевірено відповідність рівнів живлення (5 В для Arduino та DHT11, 3.3 В для NRF24L01);
- перевірено надійність контактів провідників на макетній платі breadboard.

### Процес тестування

Після підключення плат Arduino Uno до персонального комп'ютера через USB-інтерфейс було виконано компіляцію та завантаження відповідних скетчів у пам'ять мікроконтролерів.

Тестування проводилося у два етапи:

#### 1. Робота передавального вузла

- Після запуску програми Arduino ініціалізує датчик **DHT11** та модуль **NRF24L01**.
- З періодичністю 2 секунди здійснюється зчитування значень температури та вологості.
- Отримані дані передаються по бездротовому каналу на приймальний модуль.

- Поточні значення температури та вологості відображаються у **Serial Monitor** для контролю правильності вимірювань.

## 2. Робота приймального вузла

- Arduino-приймач переходить у режим прослуховування радіоканалу.
- При надходженні пакета даних значення температури та вологості приймаються та обробляються.
- Дані виводяться на **OLED-дисплей 0.96"** у реальному часі.
- Інформація дублюється у послідовному порту для діагностики.

## Результати тестування

У ході експериментальної перевірки встановлено:

- **Коректність передачі даних:** Бездротовий зв'язок між модулями NRF24L01 є стабільним, втрати пакетів даних не спостерігалися;
- **Точність вимірювань:** Датчик DHT11 забезпечує коректні показники температури та вологості в межах паспортної похибки;
- **Індикація:** Дані на OLED-дисплеї відображаються чітко та оновлюються без затримок;
- **Стабільність роботи:** Програма працює без зависань та помилок при тривалій безперервній роботі системи.

## Висновки

Результати тестування підтверджують, що розроблена система **бездротового моніторингу температури та вологості** працює стабільно та відповідає поставленому технічному завданню. Забезпечено коректну взаємодію датчика DHT11, радіомодулів NRF24L01 та OLED-дисплея. Пристрій готовий до демонстрації як діючий макет системи дистанційного контролю мікроклімату.

## ЗАГАЛЬНІ ВИСНОВКИ

У ході виконання курсової роботи було успішно розроблено та реалізовано макет **бездротової системи вимірювання температури та вологості** на базі мікроконтролерної платформи Arduino Uno.

## Апаратна реалізація

Опрацьовано принципи побудови розподілених вимірювальних систем з використанням цифрових датчиків та бездротових модулів. Реалізовано взаємодію датчика DHT11 з Arduino, а також організовано радіоканал передачі даних за допомогою NRF24L01. Використання інтерфейсу I2C для підключення OLED-дисплея дозволило зменшити кількість задіяних виводів мікроконтролера.

### **Програмна реалізація**

Розроблено програмний алгоритм мовою C++ з використанням спеціалізованих бібліотек (**DHT**, **RF24**, **Adafruit\_SSD1306**, **Wire**). Програма забезпечує стабільне зчитування даних з датчика, передачу їх по бездротовому каналу та відображення інформації на дисплеї в реальному часі.

### **Працездатність системи**

Тестування підтвердило повну відповідність роботи пристрою технічному завданню. Система стабільно функціонує в умовах безперервної роботи та забезпечує надійну передачу вимірювальних даних.

### **Навчальна цінність**

У процесі виконання роботи було закріплено практичні навички роботи з мікроконтролерами Arduino, цифровими сенсорами, бездротовими інтерфейсами SPI та I2C, а також навички налагодження та тестування вбудованих систем.

### **Перспективи розвитку**

Розроблений пристрій є базовим прототипом і може бути вдосконалений у таких напрямках:

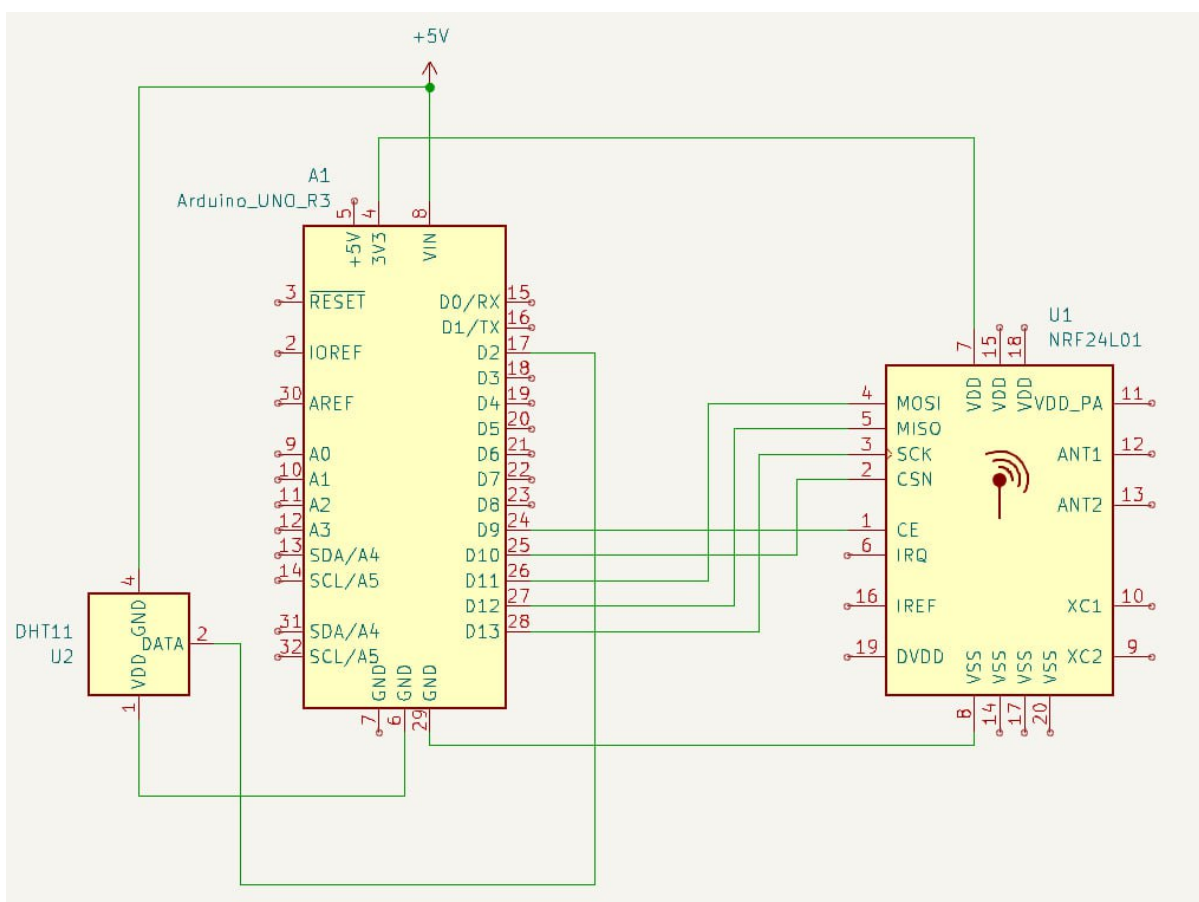
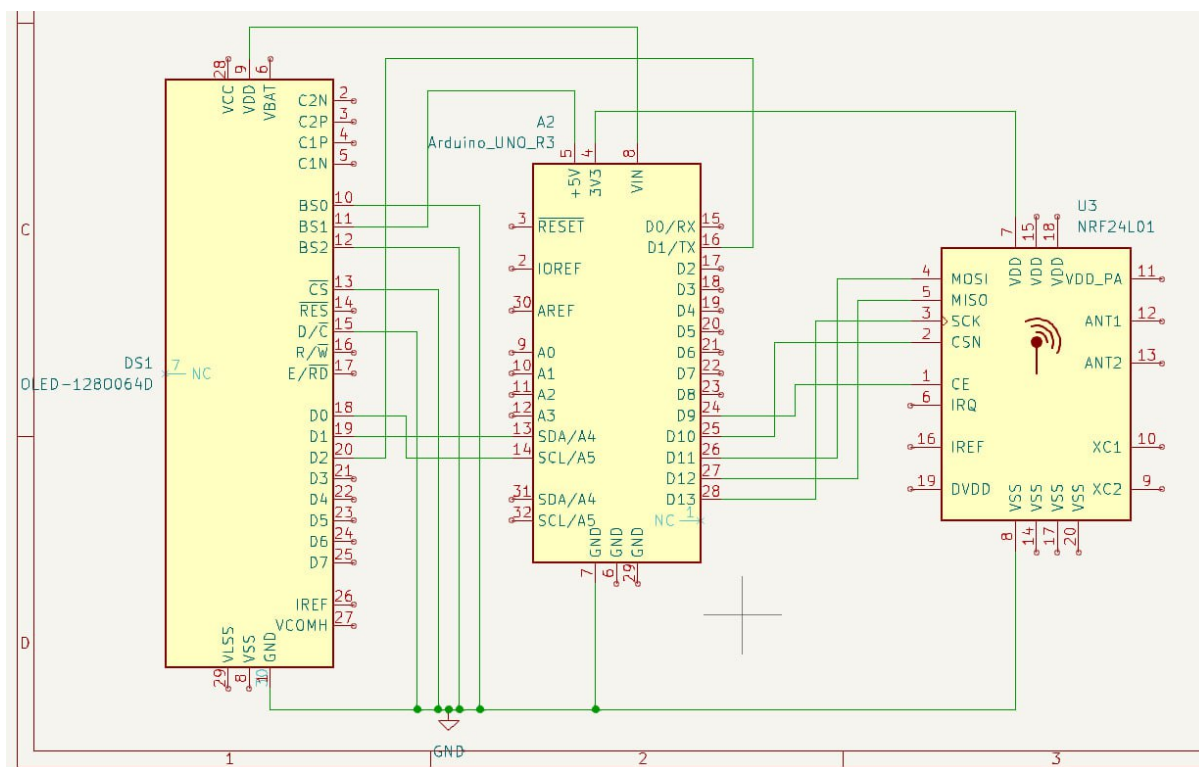
- інтеграція Wi-Fi або GSM-модуля для віддаленого моніторингу;
- додавання збереження даних на SD-карту;
- використання більш точних датчиків (DHT22, BME280);
- оптимізація енергоспоживання для автономної роботи;
- створення друкованої плати (PCB) та корпусу пристрою.



## 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