

**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

Богдан ПРОДАН

Керівник проєкту:

Остап ЮНАК

Курсовий проєкт перевірений

і допущений до захисту

“ \_\_\_\_ ” \_\_\_\_\_ 2025 р.

Курсовий проєкт при захисті оцінений

\_\_\_\_\_

**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

***Продану Богдану Ігоровичу***

(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 )

**Богдан ПРОДАН**

\_\_\_\_\_  
( імя та прізвище )

**Керівник проекту**

\_\_\_\_\_  
( підпис )

**Остап ЮНАК**

\_\_\_\_\_  
( імя та прізвище )

Львів 2025

## **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:

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.

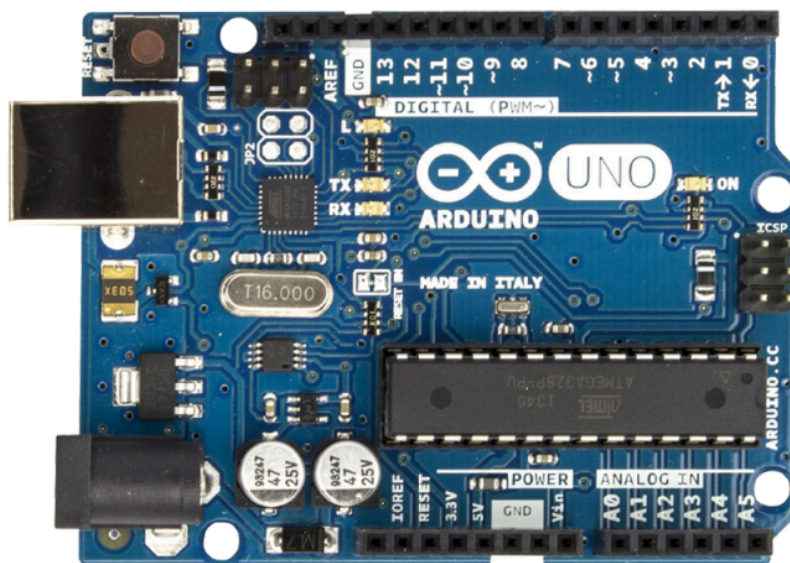


Рис. 1.1 Вигляд плати Arduino з мікроконтролером ATmega328P2

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.** Принцип роботи датчика DHT11 полягає у вимірюванні температури та відносної вологості повітря з використанням вбудованих чутливих елементів. Вологість визначається за допомогою гігроскопічного резистивного сенсора, електричний опір якого змінюється відповідно до вмісту водяної пари в повітрі. Вимірювання температури здійснюється термістором. Отримані значення обробляються вбудованим мікроконтролером і передаються у цифровому форматі, що спрощує підключення датчика та його застосування в електронних пристроях.

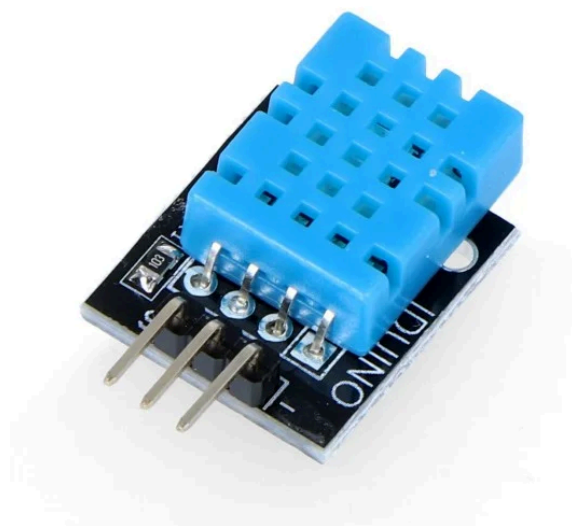


Рис 1.3.1 Вигляд датчику температури та вологості DHT11

**Модуль бездротового зв'язку NRF24L01.** Принцип роботи модуля NRF24L01 базується на бездротовій передачі та прийомі цифрової інформації за допомогою радіохвиль у частотному діапазоні 2,4 ГГц. Модуль оснащений радіотрансивером, який перетворює сигнали від мікроконтролера у радіосигнали для передавання, а під час прийому виконує зворотне перетворення. Передача даних здійснюється через інтерфейс SPI, що забезпечує високу швидкість та стабільність з'єднання. NRF24L01 підтримує механізми адресації, контролю помилок і автоматичного підтвердження отримання даних, що робить його зручним для використання в системах бездротового керування та обміну даними.

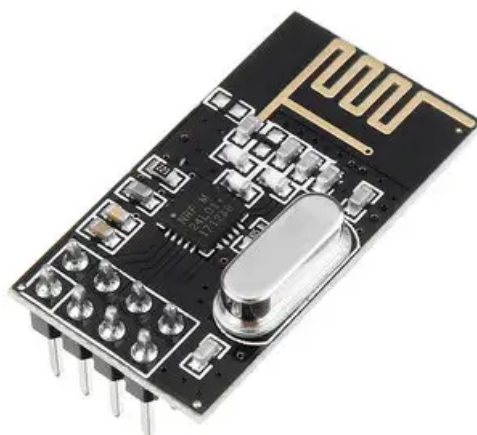


Рис 1.3.2 Вигляд модуля зв'язку NRF24L01

**OLED-дисплей 0.96" (I2C, 128×64).** Принцип роботи OLED-дисплея (Organic Light-Emitting Diode) полягає у випромінюванні світла органічними світлодіодами під дією електричного струму. Кожен піксель дисплея є самосвітним, тому додаткова підсвітка не потрібна, що забезпечує високий рівень контрастності та низьке енергоспоживання. Дисплей має роздільну здатність 128×64 пікселі та керується за допомогою інтерфейсу I2C, який дозволяє передавати інформацію лише по двох сигнальних лініях. OLED-дисплей застосовується для відображення текстових і графічних даних у мікроконтролерних системах.



Рис 1.3.2 Вид OLED-дисплей 0.96" (I2C, 128×64)

## **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 бездротової передачі даних здійснюється з використанням модуля зв'язку NRF24L01, що працює в частотному діапазоні 2,4 ГГц. Датчик температури та відносної вологості повітря DHT11, підключений до плати Arduino, виконує постійний моніторинг параметрів навколишнього середовища та передає виміряні значення до мікроконтролера. Надалі Arduino формує цифровий пакет даних і за допомогою інтерфейсу SPI передає його на радіомодуль для подальшої бездротової відправки. Приймальна частина системи, обладнана аналогічним модулем зв'язку, отримує дані, обробляє їх і відображає показники температури та вологості на OLED-дисплеї в режимі реального часу. Запропонований підхід забезпечує надійну дистанційну передачу інформації та зручне візуальне відображення даних без застосування 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 | 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 |
| 3 | DHT11            | 1         | U2                  | Датчик температури та вологості | VDD → 5V,<br>DATA → A1,<br>GND → GND                                                                               |
| 4 | Плата breadboard | 1         | -                   | 830 точок, безпайкова           | Безпайна плата для створення конструкції                                                                           |

| № | Назва компонента   | Кількість | Позначення на схемі | Характеристики                  | Примітка                                      |
|---|--------------------|-----------|---------------------|---------------------------------|-----------------------------------------------|
| 5 | Провідники         | 20        | -                   | Dupont (Male-Male, Male-Female) | Електричні з'єднання                          |
| 6 | Резистори          | 2         | R1, R2              | 220 Ом, 0.25 Вт                 | Обмеження струму LED                          |
| 7 | OLED-дисплей 0.96" | 1         | DS1                 | 128×64, I2C                     | VCC → 5V,<br>GND → GND,<br>SDA → A4, SCL → A5 |

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

Розроблення програмного забезпечення для мікроконтролера Arduino Uno виконувалося в інтегрованому середовищі розробки Arduino IDE. Як мову програмування використано C++ із залученням стандартних бібліотек Arduino, а також спеціалізованих бібліотек для роботи з бездротовим зв'язком, датчиками та OLED-дисплеєм.

Програмна частина системи складається з двох окремих скетчів:

- передавача (Arduino + DHT11 + NRF24L01);
- приймача (Arduino + NRF24L01 + OLED-дисплей).

Обмін даними між модулями реалізовано за допомогою бездротового каналу зв'язку в діапазоні 2,4 ГГц із використанням модулів NRF24L01.

### Використані бібліотеки

#### Для передавача:

- SPI.h — забезпечує обмін даними по шині SPI;
- nRF24L01.h, RF24.h — керування радіомодулем NRF24L01;
- DHT.h — зчитування значень температури та вологості з датчика DHT11.

#### Для приймача:

- SPI.h — робота з інтерфейсом SPI;
- nRF24L01.h, RF24.h — приймання даних по радіоканалу;

- Wire.h — обмін даними по шині I2C;
- Adafruit\_GFX.h, Adafruit\_SSD1306.h — керування OLED-дисплеєм з роздільною здатністю 128×64 пікселі.

## Структура програми

Програма має стандартну модульну структуру та складається з трьох основних частин:

- підключення необхідних бібліотек і оголошення змінних;
- ініціалізація апаратних компонентів у функції setup();
- основний цикл виконання програми у функції loop().

## Опис програмної частини (передавач)

### 1. Оголошення констант і змінних

Для датчика температури та вологості задано пін підключення DHTPIN = 2 і тип датчика DHT11.

Для роботи з радіомодулем створено об'єкт RF24 radio(9, 10), де 9 — CE, а 10 — CSN.

Оголошено структуру DataPacket, яка містить змінні температури та вологості повітря і використовується для передавання даних по радіоканалу.

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

Під час запуску програми ініціалізується послідовний порт для виведення діагностичних повідомлень, запускається робота датчика DHT11,

налаштовується та ініціалізується модуль NRF24L01, задається адреса каналу зв'язку, після чого модуль переводиться в режим передавання даних (stopListening()).

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

У безперервному циклі здійснюється зчитування показників температури та вологості, перевірка коректності отриманих значень, запис даних у структуру DataPacket, передавання пакета по бездротовому каналу, виведення результатів у Serial Monitor та встановлення затримки 2 секунди для стабільності вимірювань.

#### **Опис програмної частини (приймач)**

##### **1. Оголошення констант і змінних**

Для OLED-дисплея задано роздільну здатність 128×64 пікселі та I2C-адресу 0x3C.

Модуль NRF24L01 налаштований на той самий канал і адресу, що й у передавачі. Для приймання даних використовується структура DataPacket.

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

Під час запуску ініціалізується OLED-дисплей, налаштовуються параметри відображення тексту, запускається модуль NRF24L01 і переводиться в режим прийому даних (startListening()).

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

У процесі роботи перевіряється наявність прийнятих даних, зчитується структура DataPacket, очищується дисплей і на OLED виводиться заголовок системи, а також значення температури й вологості. Отримані дані додатково відображаються в Serial Monitor для контролю.

#### **Загальний алгоритм роботи системи**

Датчик DHT11 вимірює температуру та вологість повітря, після чого Arduino-передавач надсилає ці дані через модуль NRF24L01.

Arduino-приймач отримує інформацію по радіоканалу та відображає результати вимірювань на OLED-дисплеї в режимі реального часу.

5

### 5.1 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 АНАЛІЗ РЕЗУЛЬТАТІВ**

Перевірка працездатності розробленого пристрою бездротового моніторингу температури та вологості виконувалася у два основні етапи: перевірка правильності складання апаратної частини та функціональне тестування програмного забезпечення з передачею даних по радіоканалу.

### **Підготовка до тестування**

Перед подачею живлення було проведено візуальний контроль електричних схем і макетної плати з метою виявлення можливих помилок монтажу. Було перевірено правильність підключення датчика DHT11 відповідно до виводів VCC, DATA та GND, коректність підключення модулів NRF24L01 по шині SPI (лінії MOSI, MISO, SCK, CE, CSN), а також правильність з'єднання OLED-дисплея через інтерфейс I2C (лінії SDA та SCL). Додатково перевірено відповідність рівнів живлення: 5 В для плати Arduino та датчика DHT11 і 3,3 В для радіомодуля NRF24L01. Також оцінено надійність контактів з'єднувальних провідників на макетній платі.

### **Процес тестування**

Після підключення плат Arduino Uno до персонального комп'ютера за допомогою USB-інтерфейсу було виконано компіляцію та завантаження програмних скетчів у пам'ять мікроконтролерів. Подальше тестування проводилося у два етапи.

### **Робота передавального вузла**

Після запуску програми виконується ініціалізація датчика DHT11 та модуля NRF24L01. З інтервалом у 2 секунди здійснюється зчитування значень температури та вологості повітря. Отримані дані передаються по бездротовому каналу на приймальний вузол. Поточні значення температури та вологості відображаються у Serial Monitor для контролю правильності вимірювань.

### **Робота приймального вузла**

Arduino-приймач переходить у режим очікування даних з радіоканалу. Після надходження пакета даних інформація приймається, обробляється та відображається на OLED-дисплеї з діагоналлю 0,96 дюйма в режимі реального часу. Для діагностики дані також дублюються у послідовному порту.

### **Результати тестування**

У ході експериментальної перевірки встановлено стабільну роботу бездротового зв'язку між модулями NRF24L01, при цьому втрати пакетів даних не спостерігалися. Датчик DHT11 забезпечує коректні показники температури та вологості в межах паспортної похибки. Інформація на OLED-дисплеї відображається чітко та оновлюється без затримок. Програмне забезпечення працює стабільно без збоїв і зависань під час тривалої безперервної роботи.

### **Висновки за результатами тестування**

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

## **ЗАГАЛЬНІ ВИСНОВКИ**

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

### **Апаратна реалізація**

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

### **Програмна реалізація**

Розроблено програмний алгоритм мовою C++ із використанням спеціалізованих бібліотек DHT, RF24, Adafruit\_SSD1306 та Wire. Програмне забезпечення забезпечує стабільне зчитування даних з датчика, їх передачу по бездротовому каналу та відображення вимірюваних параметрів у реальному часі.

### **Працездатність системи**

Тестування підтвердило повну відповідність роботи пристрою вимогам технічного завдання. Система стабільно функціонує в умовах безперервної роботи та забезпечує надійну передачу вимірювальної інформації.

### **Навчальна цінність**

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

### **Перспективи розвитку**

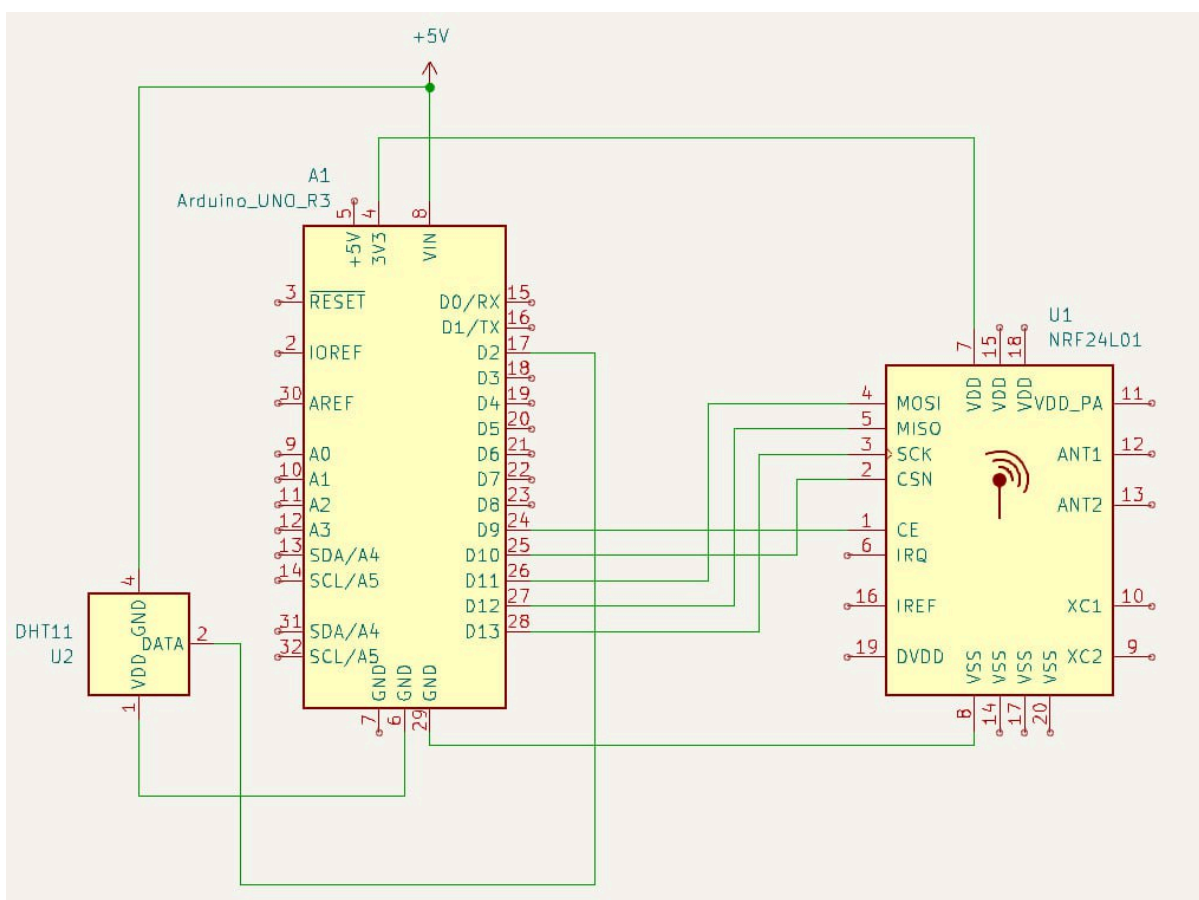
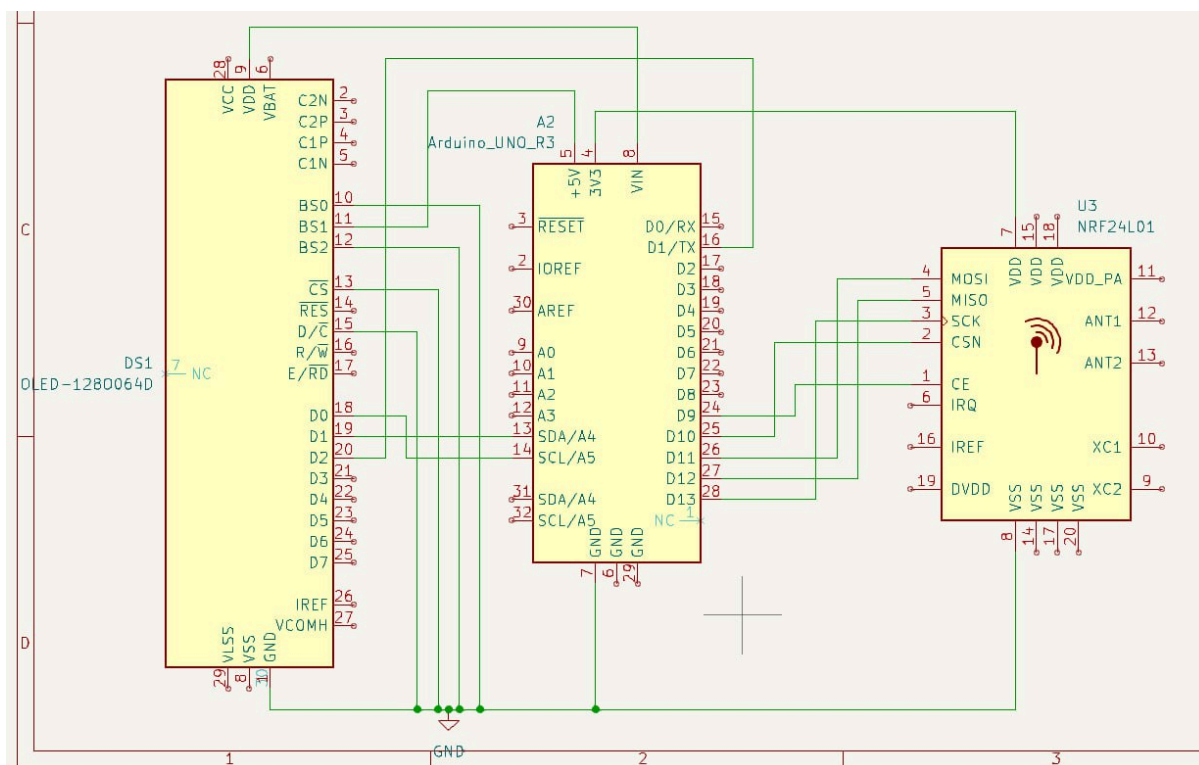
Розроблений пристрій є базовим прототипом і може бути вдосконалений шляхом інтеграції Wi-Fi або GSM-модуля, реалізації збереження даних на SD-карті, використання більш точних датчиків (DHT22, BME280), оптимізації енергоспоживання для автономної роботи, а також створення друкованої плати та корпусу пристрою.

## **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