

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ФАХОВИЙ КОЛЕДЖ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КУРСОВИЙ ПРОЄКТ

на тему:

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

Виконав студент групи МТ-41

Богдан Горін

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

Горіну Богдану Володимировичу

(прізвище, ім'я та по батькові)

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

Студент групи: MT-41

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

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 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			Богдан ГОРІН
	(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.....	11
4. ПРОГРАМНА ЧАСТИНА.....	14
5. ТЕСТУВАННЯ ТА РЕЗУЛЬТАТИ.....	20
6. ВИСНОВКИ.....	23
7. СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	25
8. 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 проєкту та функціональності, а не на низькорівневому програмуванні та складному апаратному дизайні;

– **Міждисциплінарний характер:** Розробка

апаратно-програмного пристрою вимагає інтеграції знань з електроніки, програмування мікроконтролерів (C/C++), теорії керування та проектування 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 проектування взаємодії (Interaction Design Institute Ivrea) в місті Івреа, Італія. Його засновниками стали Массімо Банці (Massimo Banzi), Девід Куартільєс, Том Іго, Джанлука Мартіно та Девід Мелліс. Основною метою створення платформи було забезпечення студентів дешевим та простим інструментом для створення інтерактивних проєктів, який би не вимагав глибоких знань у схемотехніці.

Причини вибору платформи:

Для даної курсової роботи обрано плату Arduino Uno R3, яка побудована на базі мікроконтролера ATmega328P2. Її популярність зумовлена наступними факторами:

- **Доступність:** Низька вартість компонентів;

- **Кросплатформеність:** Середовище розробки працює на Windows, macOS та Linux;
- **Простота:** Зручний інтерфейс програмування через USB без необхідності використання зовнішніх програматорів
- **Відкрита архітектура:** Схеми плат знаходяться у вільному доступі, що дозволяє легко інтегрувати різноманітні датчики та модулі;

1.2 Мова програмування C++ та середовище Arduino IDE

Програмування мікроконтролера здійснюється мовою C++. Середовище розробки Arduino IDE використовує спеціальний спрощений діалект (фреймворк) під назвою **Wiring**.

Фреймворк — це набір готових інструментів, бібліотек, шаблонів та правил, який надає структуру для створення програмного забезпечення, спрощуючи та прискорюючи процес розробки. Він допомагає розробникам уникнути написання стандартного коду з нуля, дозволяючи зосередитися на унікальній бізнес-логіці продукту. Фреймворки використовуються для різних типів проєктів, включаючи веб, мобільні та десктопні додатки.

C++ — це універсальна мова програмування високого рівня, яка підтримує об'єктно-орієнтовану парадигму. В контексті розробки під Arduino, C++ дозволяє ефективно керувати пам'яттю мікроконтролера та працювати з апаратними регістрами, зберігаючи при цьому зрозумілий синтаксис.

Програма для Arduino називається «скетчем» (sketch) і має дві обов'язкові функції:

- `setup()` — виконується один раз при запуску для налаштування конфігурації пінів;
- `loop()` — виконується циклічно, реалізуючи основний алгоритм роботи пристрою.

1.3 Принципи роботи використаних сенсорів

Датчик руху HC-SR501 (PIR Sensor). Принцип дії пасивного інфрачервоного датчика (Passive Infrared Sensor) базується на виявленні змін інфрачервоного (теплого) випромінювання. Кожен об'єкт із температурою вище абсолютного нуля випромінює тепло. Піроелектричний елемент всередині датчика реагує на зміну інтенсивності цього випромінювання при переміщенні теплого об'єкта (людини) через сектори лінзи Френеля, генеруючи електричний сигнал тривоги.



Рис 1.3.1 Вигляд датчику руху HC-SR501

Ультразвуковий датчик відстані HC-SR04. Робота датчика базується на принципі ехолокації (SONAR). Пристрій складається з двох п'єзоелементів: випромінювача та приймача.

1. Випромінювач генерує пачку ультразвукових імпульсів частотою 40 кГц;
2. Звукова хвиля відбивається від перешкоди та повертається назад до приймача;
3. Мікроконтролер вимірює час проходження сигналу і розраховує відстань.



Рис 1.3.2 Вигляд датчику HC-SR04

Рідкокристалічний дисплей (LCD 1602) з I2C. Для візуалізації даних використано символний дисплей на рідких кристалах. Особливістю даної реалізації є використання адаптера I2C (Inter-Integrated Circuit). Цей послідовний протокол дозволяє передавати дані лише по двох сигнальних лініях (SDA та SCL), що значно спрощує схемотехніку пристрою порівняно з паралельним підключенням.



Рис 1.3.2 Вигляд

рідкокристалічного дисплею (LCD 1602) з I2C

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

Організація дистанційної передачі на модулі зв'язку 2.4 GHz даних датчика температури і вологості 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

- **Напруга живлення логіки:** 5 В (USB від ПК або PowerBank).
- **Напруга живлення радіомодуля:** 3.3 В (Критично важливо! 5В пошкодить модуль NRF24L01).

- **Мікроконтролер:** Arduino Uno R3 (або Nano V3.0) — 2 шт. (один для передачі, один для прийому).
- **Робоча частота:** 2.4 GHz (ISM band).
- **Датчик параметрів середовища:** Цифровий датчик температури та вологості (DHT11 або DHT22).
- **Пристрій відображення:** OLED дисплей 0.96" (протокол I2C, 128x64 пікселів).
- **З'єднувальні елементи:** Макетна плата (Breadboard), провідники-перемички (джампери).

2.5 Очікуваний результат

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

3. АПАРАТНА ЧАСТИНА

3.1 Підбір BOM

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

Таблиця 3.1.1 Перелік необхідних компонентів

Найменування	К-ть	Позначення	Опис
Arduino Uno R3	2	U1, U2	Головний Мікроконтролер ATmega328P
Модуль NRF24L01+	2	RF1, RF2	Радіомодуль 2.4 GHz (SPI інтерфейс)
Адаптер для NRF24L01	2	-	(Рекомендовано) Стабілізатор 3.3В для стабільної роботи радіо
Датчик DHT11	1	S1	Датчик температури та вологості
OLED Дисплей	1	D1	Екран 128x64, драйвер SSD1306, I2C
Плата breadboard	2	-	Безпайна макетна плата для монтажу
Дроти-джампери	20+	-	Тато-тато, Тато-мама для з'єднань
Резистор 10 кОм	1	R1	Підтягуючий резистор для DHT11 (якщо датчик без плати)

Блок Передавача:

Компонент	Пін компонента	Пін Arduino / Живлення	Примітка
NRF24L01	VCC	3.3V	НЕ 5V!
	GND	GND	Загальна земля
	CE	Digital Pin 9	Chip Enable
	CSN	Digital Pin 10	Chip Select Not
	SCK	Digital Pin 13	SPI Clock
	MOSI	Digital Pin 11	SPI MOSI
	MISO	Digital Pin 12	SPI MISO
DHT11	VCC (+)	5V	Живлення датчика
	GND (-)	GND	Загальна земля
	DATA (Out)	Digital Pin 2	Сигнальний провід

Блок Приймача:

Компонент	Пін компонента	Пін Arduino / Живлення	Примітка
NRF24L01	VCC	3.3V	НЕ 5V!
	GND	GND	Загальна земля
	CE	Digital Pin 9	
	CSN	Digital Pin 10	
	SCK	Digital Pin 13	
	MOSI	Digital Pin 11	
	MISO	Digital Pin 12	
OLED Display	VCC	5V	Живлення дисплея
	GND	GND	Загальна земля
	SCL	Analog Pin A5	I2C Clock
	SDA	Analog Pin A4	I2C Data

4. ПРОГРАМНА ЧАСТИНА

Розробка програмного забезпечення для мікроконтролерів здійснювалася в інтегрованому середовищі розробки Arduino IDE. Мова програмування — C++ з використанням стандартних бібліотек Wiring.

Оскільки система складається з двох фізично рознесених вузлів, програмна реалізація поділяється на дві незалежні частини (скетчі): програма Передавача (Sensor Node) та програма Приймача (Receiver Node).

1. Загальний опис та використовувані бібліотеки

Для роботи з периферійними пристроями використовуються спеціалізовані бібліотеки, що забезпечують абстракцію від низькорівневих регістрів: ● <SPI.h>: забезпечує апаратний інтерфейс SPI для зв'язку з радіомодулем.

- <RF24.h> (nRF24L01): реалізує протокол керування трансивером, налаштування каналів та пакетну передачу даних.
- <DHT.h>: забезпечує зчитування та обробку сигналів з цифрового датчика температури та вологості.
- <Wire.h>, <Adafruit_GFX.h>, <Adafruit_SSD1306.h>: набір бібліотек для керування OLED-дисплеєм через шину I2C та формування графічного інтерфейсу.

2. Алгоритм роботи Передавача

Програма передавача виконує циклічний опитування датчика та відправку пакетів даних.

- Оголошення змінних та структур: Створюється структура даних (struct DataPackage), яка містить дві змінні типу float (температура та вологість). Це дозволяє передавати дані єдиним пакетом бітів, а не окремими посилками. Визначаються піни підключення CE (9) та CSN (10).
- Ініціалізація (setup):
 - Ініціалізується робота датчика DHT.
 - Налаштовується радіомодуль: встановлюється канал зв'язку, рівень підсилення (PA Level) та відкривається "труба" (pipe) на запис.
- Основний цикл (loop):
 - Виконується зчитування показників вологості та температури.

- Здійснюється перевірка коректності даних (функція `isnan`): якщо датчик повернув помилку, передача блокується до наступного циклу.
- Дані упаковуються в структуру і відправляються методом `radio.write()`.
- Встановлюється затримка (`delay`) для енергозбереження та запобігання перевантаженню каналу.

3. Алгоритм роботи Приймача

Програма приймача працює в режимі прослуховування ефіру та оновлення інформації на дисплеї.

- Оголошення змінних: Аналогічно передавачу, оголошується ідентична структура даних для коректного розпакування прийнятих байтів. Ініціалізується об'єкт дисплея з роздільною здатністю 128x64 пікселів.
- Ініціалізація (`setup`):
 - Запускається шина I2C та OLED-дисплей. Екран очищується.
 - Радіомодуль налаштовується на ту саму частоту та адресу, що й передавач, але відкривається "труба" на читання.
 - Активується режим прослуховування (`startListening`).
- Основний цикл (`loop`):
 - Функція `radio.available()` постійно перевіряє наявність вхідних пакетів даних у буфері.
 - При надходженні сигналу метод `radio.read()` записує отримані біти у локальну структуру змінних.
 - Отримані значення виводяться на OLED-екран за допомогою функцій бібліотеки Adafruit (встановлення курсору, розміру шрифту та друк значень).
 - Інтерфейс оновлюється лише при надходженні нових даних, що забезпечує відсутність мерехтіння екрана.

4.1 Код Передатчика

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

#define DHTPIN 2
#define DHTTYPE DHT11

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

const byte address[6] = "00001";

struct DataPackage {
  float temperature;
  float humidity;
};

DataPackage data;

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

  radio.begin();
  radio.openWritingPipe(address);
  radio.setPALevel(RF24_PA_MIN);
  radio.stopListening();
}

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

```

if (isnan(data.humidity) || isnan(data.temperature)) {
  Serial.println("Помилка зчитування з DHT сенсора!");
  return;
}

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

Serial.print("T: "); Serial.print(data.temperature);
Serial.print(" H: "); Serial.println(data.humidity);

delay(2000);
}

```

4.2 Код Приймача

```

#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);
const byte address[6] = "00001";

struct DataPackage {
  float temperature;
  float humidity;
};

```

```
DataPackage data;

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

  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println(F("SSD1306 allocation failed"));
    for(;;);
  }
  display.clearDisplay();
  display.setTextColor(WHITE);

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

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

    Serial.print("Recv T: "); Serial.print(data.temperature);
    Serial.print(" Recv H: "); Serial.println(data.humidity);

    updateDisplay();
  }
}

void updateDisplay() {
  display.clearDisplay();

  display.setTextSize(1);
  display.setCursor(0, 0);
  display.println("Monitor System 2.4G");
  display.drawLine(0, 10, 128, 10, WHITE);
}
```

```
display.setTextSize(2);  
display.setCursor(0,20);  
display.print("T: ");  
display.print(data.temperature,1);  
display.cp437(true);  
display.write(167);//Знакградуса  
display.println("C");
```

```
display.setCursor(0,45);  
display.print("H: ");  
display.print(data.humidity,1);  
display.println("%");
```

```
display.display();  
}
```

5. ТЕСТУВАННЯ ТА РЕЗУЛЬТАТИ

Перевірка працездатності розробленої системи бездротового моніторингу проводилася у два етапи: перевірка монтажу апаратної частини двох вузлів (передавача та приймача) та комплексне функціональне тестування програмного забезпечення.

1. Підготовка до тестування

Перед подачею живлення було проведено ретельний візуальний огляд макетних плат обох вузлів для виявлення можливих помилок монтажу:

- **Контроль живлення радіомодулів:** Перевірено коректність підключення модулів NRF24L01 до виводів 3.3 В (оскільки подача 5 В може миттєво вивести їх з ладу).
- **Перевірка шин даних:** Пересвідченося у правильності комутації інтерфейсу SPI (піни 9, 10, 11, 12, 13) для радіомодулів та інтерфейсу I2C (піни A4, A5) для OLED-дисплея.
- **Підключення сенсора:** Перевірено полярність підключення датчика DHT11 (VCC, GND, Signal) та наявність підтягуючого резистора (якщо використовується окремий компонент).
- **Надійність контактів:** Перевірено цілісність з'єднань провідників-джамперів на платах breadboard для уникнення «брязкоту» контактів.

2. Процес тестування

Після підключення обох плат Arduino до джерел живлення (USB-порти ПК або PowerBank), було виконано компіляцію та завантаження відповідних скетчів: «Transmitter» у вузол з датчиком та «Receiver» у вузол з дисплеєм.

В ході тестування було виконано наступні кроки:

1. **Перевірка ініціалізації:** Через вбудований інструмент «Монітор порту» (Serial Monitor) перевірено статус ініціалізації модулів (повідомлення про успішний старт радіо та дисплея).
2. **Тест передачі даних:** Спостереження за OLED-екраном приймача. Зафіксовано появу початкових значень температури та вологості, що відповідають кімнатним умовам.
3. **Динамічний тест сенсора:** Для перевірки реакції системи на зміни середовища було здійснено штучне підвищення вологості та температури (шляхом видиху теплого повітря на датчик DHT11).
4. **Тест дальності зв'язку:** Вузол приймача поступово віддалявся від передавача для перевірки стабільності радіоканалу на відстані в межах приміщення.

3. Результати

Під час випробувань було встановлено:

- **Точність відображення:** Значення на дисплеї приймача повністю відповідають даним, що зчитуються датчиком на стороні передавача (синхронізація підтверджена через Serial Monitor).
- **Швидкодія:** Оновлення інформації на екрані відбувається коректно, із заданим програмним інтервалом (2 секунди), без видимих затримок чи зависань.
- **Чутливість до змін:** Система успішно зафіксувала різке зростання вологості під час динамічного тесту та відобразила зміни на графічному інтерфейсі OLED.
- **Стабільність:** Зв'язок на частоті 2.4 ГГц залишався стійким, втрати пакетів даних у межах робочої зони не виявлено.

Висновки

Результати тестування підтверджують, що розроблена бездротова система працює стабільно і виконує поставлене технічне завдання. Апаратна частина (датчики, трансивери, дисплей) та програмна логіка успішно інтегровані. Пристрій забезпечує дистанційний моніторинг параметрів мікроклімату та готовий до експлуатації.

6. ВИСНОВКИ

У ході виконання курсової роботи було успішно розроблено та реалізовано макет системи бездротового дистанційного моніторингу параметрів навколишнього середовища на базі платформи Arduino та радіомодулів діапазону 2.4 ГГц.

Висновки:

- 1. Апаратна реалізація:** Опрацьовано принципи побудови розподілених систем, що складаються з незалежних вузлів (передавача та приймача). Отримано практичні навички роботи з різними інтерфейсами передачі даних: SPI (для радіомодулів NRF24L01), I2C (для OLED-дисплея) та однопровідного інтерфейсу датчиків DHT. Особливу увагу приділено узгодженню рівнів напруги (3.3 В та 5 В) для забезпечення стабільної роботи компонентів.
- 2. Програмна реалізація:** Розроблено два незалежні програмні алгоритми (скетчі) мовою C++. Реалізовано механізм пакетної передачі даних за допомогою структур (struct), що дозволило синхронно передавати значення температури та вологості. Використання спеціалізованих бібліотек (RF24, Adafruit_SSD1306) забезпечило ефективну взаємодію з периферією.
- 3. Працездатність системи:** Тестування підтвердило повну відповідність роботи пристрою технічному завданню. Система забезпечує стабільний радіозв'язок на частоті 2.4 ГГц, коректне зчитування фізичних параметрів датчиком DHT11 та їх чітку візуалізацію на графічному OLED-екрані приймача в режимі реального часу.
- 4. Навчальна цінність:** Робота дозволила поглибити знання в галузі бездротових технологій (IoT), схемотехніки та протоколів обміну

даними, а також отримати досвід налагодження радіоелектронної апаратури.

Перспективи розвитку:

Розроблений пристрій є базовим прототипом системи «Розумний дім», який має значний потенціал для модернізації та розширення функціоналу. До основних напрямів подальшого вдосконалення можна віднести:

- 1. Масштабування мережі (Mesh-мережа):** Розширення системи шляхом додавання кількох передавачів (сенсорних вузлів), розташованих у різних кімнатах. Приймач зможе збирати та відображати дані з декількох точок одночасно, ідентифікуючи їх за унікальними адресами каналів.
- 2. Автоматизація клімат-контролю:** Впровадження виконавчих механізмів (реле). Наприклад, автоматичне вмикання вентилятора або зволожувача повітря, якщо отримані показники температури чи вологості виходять за встановлені межі.
- 3. Логування даних:** Додавання модуля годинника реального часу (RTC) та слота для SD-карти на стороні приймача для запису історії змін температури у файл (data logger) для подальшого аналізу.
- 4. Інтеграція з мережею Інтернет:** Заміна Arduino на приймачі платою ESP8266 або ESP32, що дозволить передавати отримані дані на веб-сервер або у мобільний додаток (Blynk, Telegram-бот) для віддаленого контролю зі смартфона.
- 5. Енергоефективність:** Оптимізація програмного коду передавача шляхом використання режимів глибокого сну (Deep Sleep) мікроконтролера та радіомодуля. Це дозволить пристрою працювати від акумуляторної батареї протягом кількох місяців без підзарядки.

7. 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, В.** *Mastering Arduino Programming (2021). Призначення: Програмування мікроконтролерів, синтаксис C/C++.*
- **Schwartz, М.** *Arduino Electronics Blueprints (2019). Призначення: Інтеграція компонентів, використання бібліотек, створення скетчів.*
- **Axelson, D.** *Embedded Systems Design (2020). Призначення: Фундаментальні основи проектування вбудованих систем, архітектура.*
- **Marr, В.** *Internet of Things (IoT) in Business (2022). Призначення: Концепції IoT, мережеві протоколи та стандарти для пристроїв.*

8. ДОДАТКИ

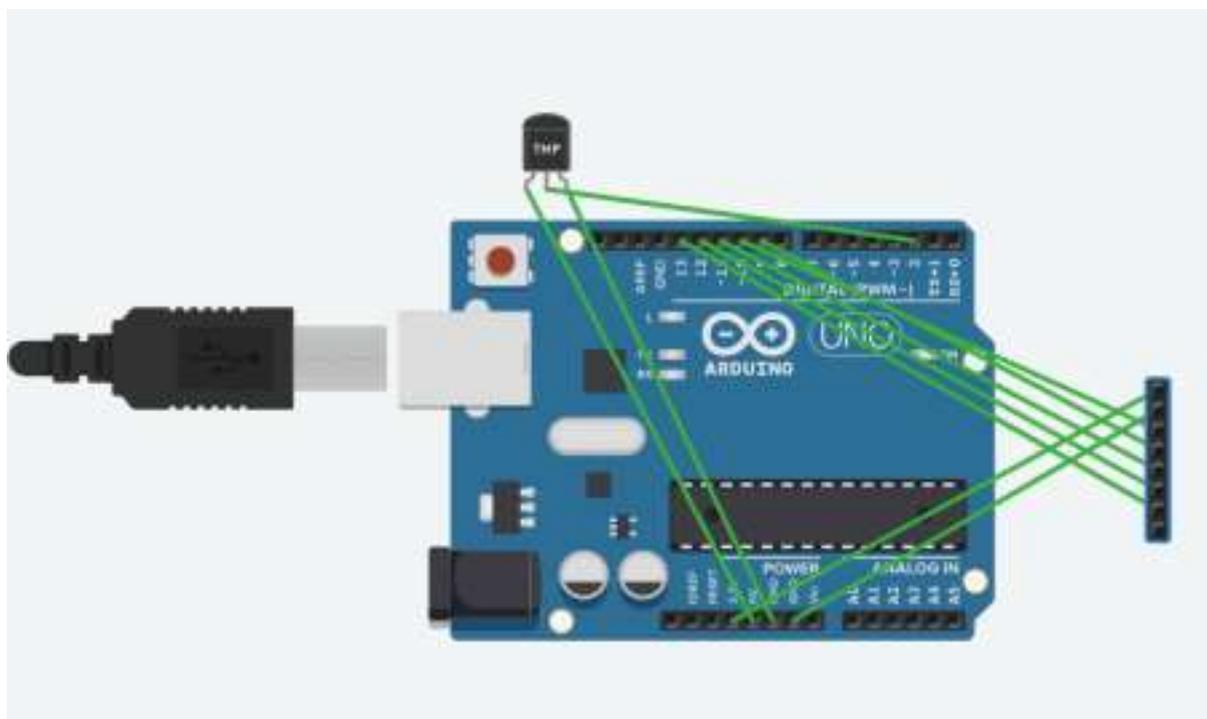


Схема 1: Передатчик

