

Міністерство освіти і науки України  
Державний заклад  
«Луганський національний університет імені Тараса Шевченка»

Навчально-науковий інститут математики та інформаційних технологій

Кафедра інформаційних технологій та систем

Ліщук Сергій Едуардович

**РОЗРОБКА СИСТЕМИ МОНІТОРИНГУ МІКРОКЛІМАТУ  
СЕРВЕРНОЇ КІМНАТИ НА БАЗІ ARDUINO**

**кваліфікаційна робота**

**здобувача вищої освіти першого (бакалаврського) рівня**

**освітньої програми «Комп'ютерна інженерія»**

**за спеціальністю 123 Комп'ютерна інженерія**

Особистий підпис \_\_\_\_\_ Сергій ЛІЩУК

Науковий керівник \_\_\_\_\_ Володимир ДОНЧЕНКО,  
старший викладач  
кафедри інформаційних технологій  
та систем

Завідувач кафедри \_\_\_\_\_ Микола СЕМЕНОВ,  
кандидат педагогічних наук, доцент  
кафедри інформаційних технологій  
та систем

Полтава – 2025

Міністерство освіти і науки України  
Державний заклад «Луганський національний університет  
імені Тараса Шевченка»

Факультет (інститут)	Навчально-науковий інститут математики та інформаційних технологій (повна назва)
Кафедра	Кафедра інформаційних технологій та систем (повна назва)
Освітній ступень	Бакалавр (код, назва)
Напрямок підготовки	123 Комп'ютерна інженерія (код, назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри ІТС  
М.А. Семенов

(підпис) (ініціали, прізвище)  
“ ” 2024 р.

ЗАВДАННЯ  
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТУ

Ліщуку Сергію Едуардовичу  
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Розробка системи моніторингу мікроклімату серверної кімнати на базі ARDUINO

Керівник кваліфікаційної роботи Донченко В.Ю. старший викладач  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджена наказом по університету від

2. Строк подання студентом проекту (роботи)

3. Вихідні дані до роботи (проекту) Розроблено систему моніторингу мікроклімату серверної кімнати на базі ARDUINO

(визначаються кількісні або (та) якісні показники, яким повинен відповідати об'єкт розробки)

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) АНАЛІТИЧНИЙ ОГЛЯД.

ВИБІР КОМПОНЕНТІВ ДЛЯ СИСТЕМИ МОНІТОРИНГУ МІКРОКЛІМАТУ СЕРВЕРНОЇ КІМНАТИ НА ОСНОВІ МІКРОКОНТРОЛЕРА ARDUINO. ПРОЄКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНО-АПАРАТНОЇ СИСТЕМИ.

(визначаються назви розділів або (та) перелік питань, які повинні увійти до тексту ПЗ)

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

## 6. Консультанти розділів проекту/роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

## 7. Дата видачі завдання „ 10 ” жовтня 2024р.

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи )	Примітка
	Вибір теми роботи, вивчення наукової літератури, затвердження теми та керівника.	До 15 жовтня	
	Аналіз літературних джерел за темою роботи. Розробка та апробація методики дослідно-експериментальної роботи. Подання структури теоретичної частини роботи та плану експериментальних досліджень.	Другий тиждень листопада (10 листопада )	
	Робота над теоретичною частиною. Подання теоретичної частини роботи для першого читання науковим керівником.	До 15 грудня	
	Усунення зауважень, урахування рекомендацій наукового керівника. Подання теоретичної частини роботи на друге читання.	До 28 січня	
	Проведення експериментальної роботи. Поетапний аналіз та обговорення її результатів. Перевірка стану виконання роботи.	Перший тиждень березня	
	Урахування рекомендацій наукового керівника, усунення недоліків, підготовка варіанта роботи до передзахисту. Розробка презентації.	До 31 березня	
	Попередній захист роботи на кафедрі	квітень	
	Доопрацювання роботи з урахуванням рекомендацій після передзахисту. Подання роботи науковому керівникові та рецензентові на підготовку відгуку та рецензії	За 10 днів до державної атестації	
	Подання на кафедру остаточного варіанта роботи, переплетеного та підписаного автором, науковим керівником і рецензентом.	За 5 днів до державної атестації	

Студент

підпис

Керівник проекту (роботи)

підпис

С.Е. Ліщук

(ініціали, прізвище)

В. Ю. Донченко

(ініціали, прізвище)

## **АНОТАЦІЯ**

**Ліщук С. Е.**

**Тема: Розробка системи моніторингу мікроклімату серверної кімнати на базі ARDUINO.**

**Спеціальність:** 123 «Комп'ютерна інженерія»

**Установа:** ЛНУ імені Тараса Шевченка, 2025 р.

**Бакалаврська робота містить:** 114 с., 72 рис., 3 табл., 30 джерел.

**Об'єкт дослідження** – процес контролю параметрів мікроклімату серверної кімнати.

**Предмет дослідження** – автоматизована система контролю параметрів мікроклімату серверної кімнати на базі мікроконтролера Arduino.

**Мета дослідження** – аналіз та розробка системи моніторингу мікроклімату серверної кімнати на базі Arduino.

**Для вирішення поставлених задач** використовувались методи моделювання та програмування; положення теорії автоматичного керування; методи створення автоматизованих вимірювальних мікропроцесорних систем; методи математичної статистики та обробки експериментальних результатів.

### **Результати роботи.**

Проведено огляд існуючих систем моніторингу мікроклімату серверних кімнат на базі Arduino, що дозволило виявити їхні переваги, недоліки та визначити актуальні тенденції в даній галузі. Здійснено детальне виявлення та формалізацію вимог до проєктованої системи моніторингу на основі потреб типового підприємства з серверною інфраструктурою. Розроблено архітектуру інформаційної системи, включаючи визначення її функціональних модулів, інтерфейсів та принципів взаємодії. Описана логіка функціонування програмних модулів системи моніторингу мікроклімату серверної кімнати на базі платформи Arduino. Розроблено мобільний застосунок "Blynk" для системи дистанційного моніторингу мікроклімату серверної кімнати. Створено прототип системи моніторингу мікроклімату серверної кімнати на базі Arduino та здійснено його попереднє впровадження для демонстрації працездатності та відповідності типовим вимогам.

**Ключові слова.** СЕРВЕРНА КІМНАТА, МІКРОКЛІМАТ, ARDUINO MEGA 2560, IOT (INTERNET OF THINGS), МІКРОПРОЦЕСОРНА СИСТЕМА, BLYNK.

## ABSTRACT

**Lishchuk Serhii Eduardovych**

**Theme: Development of a server room microclimate monitoring system based on ARDUINO.**

**Speciality:** 123 "Computer Engineering"

**Institution:** Luhansk Taras Shevchenko National University (LTSNU), 2025.

**Diploma work contains:** 114 pages, 72 Fig., 3 Table, 30 source.

**A research object is** the process of controlling the microclimate parameters of the server room.

**The article of research** automated system for monitoring server room microclimate parameters based on Arduino microcontroller.

**An aim of research is** analysis and development of a server room microclimate monitoring system based on Arduino.

To solve the tasks, modeling and programming methods were used; the principles of the theory of automatic control; methods for creating automated measuring microprocessor systems; methods of mathematical statistics and processing of experimental results.

**Job performances.** A review of existing server room microclimate monitoring systems based on Arduino was conducted, which allowed us to identify their advantages, disadvantages and determine current trends in this industry. A detailed identification and formalization of requirements for the designed monitoring system was carried out based on the needs of a typical enterprise with a server infrastructure. The architecture of the information system was developed, including the definition of its functional modules, interfaces and principles of interaction. The logic of the functioning of the software modules of the server room microclimate monitoring system based on the Arduino platform was described. The mobile application "Blynk" was developed for the remote monitoring system of the server room microclimate. A prototype of the server room microclimate monitoring system based on Arduino was created and its preliminary implementation was carried out to demonstrate its operability and compliance with typical requirements.

**Keywords.** SERVER ROOM, MICROCLIMATE, ARDUINO MEGA 2560, IOT (INTERNET OF THINGS), MICROPROCESSOR SYSTEM, BLYNK.

# Розробка системи моніторингу мікроклімату серверної кімнати на базі ARDUINO.

					ІТС.4КІ.0125.01-ВП							
Змн.	Арк.	№ докум.	Підпис	Дата	ВІДОМІСТЬ ПРОЄКТУ			Літ.	Арк.	Акрушіє		
Розроб.		Ліщук С.Е.									1	1
Керівник		Донченко В.Ю.						ЛНУ  Кафедра ІТС, Гр.4КІ				
Реценз.		Козуб Ю.Г.										
Н. Контр.												
Зав. каф.		Семенов М.А..										

<b>Міністерство освіти і науки України</b>	
<b>Державний заклад «Луганський національний університет</b>	
<b>імені Тараса Шевченка»</b>	
Факультет (інститут)	Навчально-науковий інститут
	математики та інформаційних технологій
	(повна назва)
Кафедра	Кафедра інформаційних технологій та
	систем
	(повна назва)
	(код, назва)

**ТЕХНІЧНЕ ЗАВДАННЯ**  
на виконання програмної розробки (ПР):  
**"РОЗРОБКА СИСТЕМИ МОНІТОРИНГУ МІКРОКЛІМАТУ**  
**СЕРВЕРНОЇ КІМНАТИ НА БАЗІ ARDUINO"**

**ІТС.4КІ.0125.02-ТЗ**

**ПОГОДЖЕНО**  
Керівник кваліфікаційної роботи

\_\_\_\_\_  
Донченко В.Ю.

“ \_\_\_\_\_ ” 2025р

**ВИКОНАВЕЦЬ**  
Студент групи 4КІ

\_\_\_\_\_  
Ліщук С. Е.

“ \_\_\_\_\_ ” 2025р

Полтава – 2025

## ЗМІСТ

ВСТУП.....	3
1. ПРИЗНАЧЕННЯ ПРОДУКЦІЇ .....	3
2. ТЕХНІЧНІ ХАРАКТЕРИСТИКИ.....	3
3. АЛГОРИТМ РОБОТИ ПРИСТРОЮ .....	5
4. ТЕХНІКО-ЕКОНОМІЧНІ ВИМОГИ ДО КІНЦЕВОГО ПРОДУКТУ	6
5. ВИМОГИ ДО МАТЕРІАЛІВ ТА КОМПЛЕКТУЮЧИХ.....	6
6. ЕТАПИ ВИКОНАННЯ ПР.....	6
7. ПРИЙМАННЯ .....	7
8. ПОРЯДОК ВНЕСЕННЯ ЗМІН ДО ТЕХНІЧНОГО ЗАВДАННЯ, ЩО ЗАТВЕРДЖЕНО.....	8

					ІТС.4КІ.0125.02-ТЗ					
Змн.	Арк.	№ докум.	Підпис	Дата	ТЕХНІЧНЕ ЗАВДАННЯ			Літ.	Арк.	Акрушів
Розроб.		Ліщук С.Е.								
Керівник		Донченко В.Ю.							2	7
Реценз.		Козуб Ю.Г.						ЛНУ Кафедра ІТС, Гр.4КІ		
Н. Контр.										
Зав. каф.		Семенов М.А..								

## ВСТУП

**1.1 Найменування: Розробка системи моніторингу мікроклімату серверної кімнати на базі ARDUINO.**

**1.2 Шифр ПР: АДР-5**

**1.3 Підстава до виконання ПР:** Підставою для виконання даної розробки є завдання на дипломний проєкт.

**1.4 Терміни розробки:**

1.4.1 Початок 30 жовтня 2024р.

1.4.2 Закінчення 20 квітня 2025р.

**1.5 Фінансується** за рахунок коштів замовника.

## 1. ПРИЗНАЧЕННЯ ПРОДУКЦІЇ

### 1.1. Призначення:

Цей пристрій призначений для управління наступними параметрами серверної кімнати:

- контроль температури і вологості;
- контроль доступу;
- управління світлом.

Мікрокомплекс створюється з метою:

- Відстеження параметрів мікроклімату приміщення.
- Оповіщення відповідальних осіб у разі порушення кліматичних умов та несанкціонованого доступу до серверної кімнати.

Завдання, що вирішуються

- Складання структурної схеми мікрокомплексу.
- Розробка технічних умов і вузлів мікрокомплексу.
- Створення основного програмного модуля.
- Створення мікроскладного макета.

## 2. ТЕХНІЧНІ ХАРАКТЕРИСТИКИ

					ІТС.4КІ.0125.02-ТЗ	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

Розроблено систему моніторингу мікроклімату серверної кімнати на базі ARDUINO. Прилади які живляться від 220В були підключенні до ардуіни за допомогою оптосимісторів та симісторів.

За допомогою безкоштовної програм Arduino IDE, було написано програмне забезпечення для функціонування системи моніторингу мікроклімату серверної кімнати.

№	Назва Arduino	Пояснення
1	Mega2560 + Ethernet Shield	Платформа для роботи з модулями і датчиками. Ethernet використовується для підключення до інтернету.
2	Інфрачервоний датчик руху (піроелектричний)	Використовується для вмикання/вимикання світла.
3	Зчитувач RFID MFRC-522	Використовується для здійснення пропускового режиму (світлодіод використовується як заміник замка).
4	Кнопка тактова	Використовується для перемикавання пунктів меню на LCD-екрані.
5	LCD дисплей 1602 (символьний на HD44780, з підсвіткою, зелений)	Використовується для відображення параметрів, що передаються датчиками (має 3 пункти меню).
6	Модуль з двома реле для Arduino	Використовується у зв'язку з ІЧ-датчиком руху і RFID для перемикавання світла (вкл/викл), а також відкриття/закриття електромеханічного замка.
7	Прецизійний годинник реального часу DS3231	Дозволяє рахувати поточний час і дату для запису даних на SD-карту.
8	Цифровий датчик температури і вологості DHT11	Використовується для вимірювання температури і вологості в приміщенні.
9	Резистор із опором 10 кОм	
10	Лампочка накаливання	Вмикання/вимикання світла.
11	Електромеханічний замок	Відкриття/закриття дверей.

№	Назва Arduino	Пояснення
12	Micro SD-карта	Використовується для зберігання параметрів у БД.
13	ІЧ-передавач	Служить для управління кондиціонером.

### 3. АЛГОРИТМ РОБОТИ ПРИСТРОЮ

Мікрокомплекс на базі мікроконтролера для контролю параметрів серверної кімнати повинен складатися з наступних модулів і їх функцій:

#### 1) Модуль температури та вологості

Він виконує функцію контролю та регулювання температури в серверній. Коли температура підвищується/знижується вище / нижче середнього значення, він повинен регулювати роботу кондиціонера.

#### 2) Модуль підсвічування

Він виконує функцію увімкнення/вимкнення світла у разі наявності/відсутності персоналу в серверній кімнаті відповідно.

#### 3) Модуль контролю доступу

Виконує функцію реєстрації осіб, які заходять на територію приміщення.

#### 4) Стаціонарний модуль візуалізації

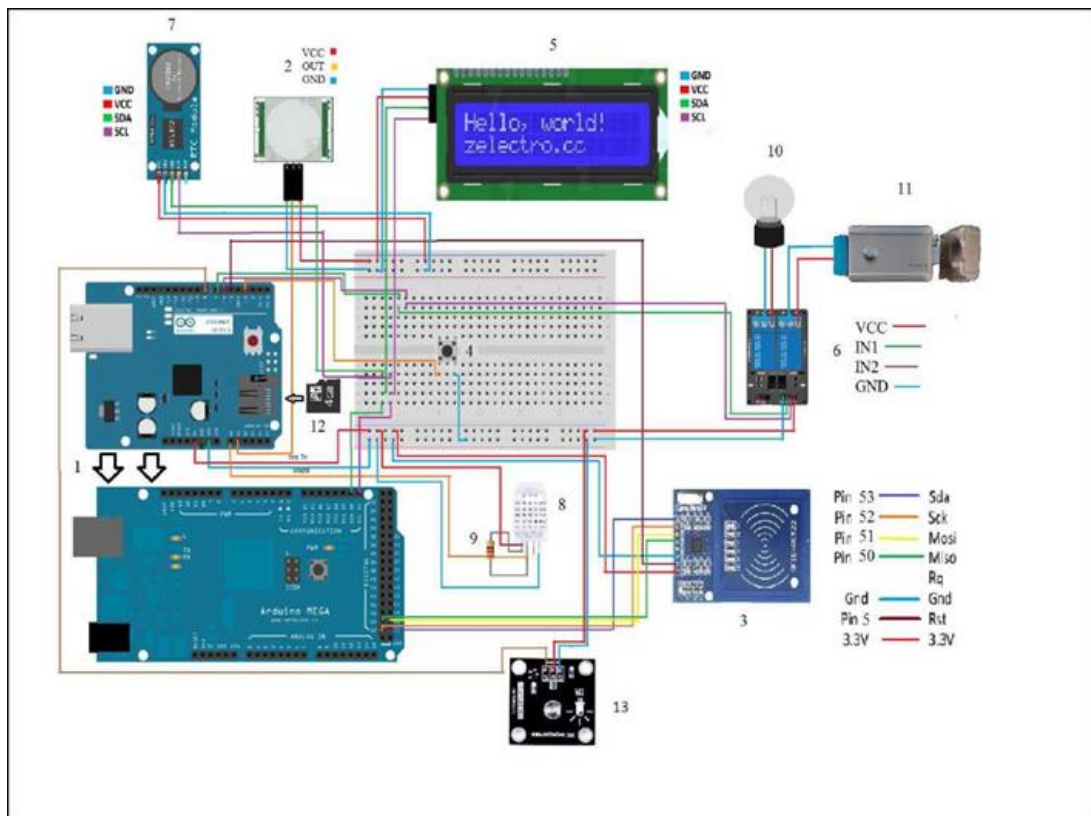
Він виконує функцію демонстрації основних параметрів приміщення на рідкокристалічному екрані пристрою візуального відображення інформації.

#### 5) Модуль візуалізації

Він виконує функцію демонстрації основних параметрів приміщення на веб-сторінці та в мобільному додатку Blink.

#### 6) Модуль зберігання

Виконує функцію зберігання даних з датчиків на SD-карті.



#### 4. ТЕХНІКО-ЕКОНОМІЧНІ ВИМОГИ ДО КІНЦЕВОГО ПРОДУКТУ

Вартість робіт по розробці даної ПР визначається згідно договору на розробку. Вартість запропонованих аналогів повинна забезпечити економічну доцільність їх застосування.

#### 5. ВИМОГИ ДО МАТЕРІАЛІВ ТА КОМПЛЕКТУЮЧИХ

5.1. Вимоги до екологічної безпечності під час експлуатації.

Не пред'являються.

5.2. Спеціальні вимоги до кінцевого продукту.

Не пред'являються.

5.3. Вимоги до безпеки для населення під час експлуатації продукції.

Не пред'являються.

#### 6. ЕТАПИ ВИКОНАННЯ ПР.

Етапи виконання ПР можуть уточнювати згідно календарного плану робіт по узгодженню між замовником та виконавцем

№	Етапи виконання роботи	Термін виконання та обсяг робіт	звітні матеріали
1	Аналіз розробки програмного комплексу та розробка першої версії. Аналіз вимог. Розробка структури. Попереднє тестування		Частковий програмний комплекс на ЕОМ замовника, що виконує всі основні функції та звітна документація п.8.2
2	Коректування структури. Розробка допоміжних функцій. Розробка остаточної версії програмного комплексу та його опрацювання. Тестування		Готовий програмний комплекс на ЕОМ замовника та звітна документація п.8.2
3	Доопрацювання окремих модулів та навчання користувачів. Розробка звітних матеріалів згідно п.8 цього ТЗ		звітні матеріали згідно пункту 8

## 7. ПРИЙМАННЯ

### 7.1. Необхідні вимоги для впровадження ПР та завершення робіт.

Оцінка результатів розробки і доцільність її продовження здійснюється замовником по представленню наступних матеріалів:

- встановлений програмний комплекс на ЕОМ замовника;
- перелік файлів на резервному носії;
- стислий опис роботи ПР та опис всіх файлів, які необхідні для роботи ПР.
- перелік документів
  - Технічне завдання
  - Пояснювальна записка

### 7.2. Перелік звітних документів, необхідних для прийняття етапів роботи:

- стислий опис результатів етапу у вигляді анотованого звіту(для 1 та 2 етапів);

					ІТС.4КІ.0125.02-ТЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

- частковий програмний комплекс на ЕОМ замовника згідно календарного плану робіт;
- акт приймання продукції.

7.3. Загальний перелік до приймання звітних документів, макетів, експериментальних зразків.

До приймання пред'являються: акт здачі-приймання продукції, акт впровадження ПР.

#### 7.4. Тестування ПР

Тестування виконується до "Програми та методики тестування", яка розробляється виконавцем та затверджується замовником

### **8. ПОРЯДОК ВНЕСЕННЯ ЗМІН ДО ТЕХНІЧНОГО ЗАВДАННЯ, ЩО ЗАТВЕРДЖЕНО.**

Дане технічне завдання може уточнюватися в процесі розробки ПР при узгодженні сторін з оформленням доповнень до ТЗ.

					<b>ІТС.4КІ.0125.02-ТЗ</b>	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**ДЗ «ЛУГАНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**  
**ІМЕНІ ТАРАСА ШЕВЧЕНКА»**

Навчально-науковий інститут математики та інформаційних  
технологій

(назва факультету, інституту)

Інформаційних технологій та систем

(назва кафедри)

**Пояснювальна записка**  
до дипломного проєкту (роботи)  
**БАКАЛАВРА**  
(освітньо-кваліфікаційний рівень)

на тему:  
**РОЗРОБКА СИСТЕМИ МОНІТОРИНГУ МІКРОКЛІМАТУ СЕРВЕРНОЇ**  
**КІМНАТИ НА БАЗІ ARDUINO**

Виконав: студент 4 курсу, групи \_\_\_\_  
напряму підготовки (спеціальності)  
**123 «Комп'ютерна інженерія»**  
(шифр і назва напряму підготовки, спеціальності)

**Ліщук С.Е.**  
(прізвище та ініціали)

Керівник **Донченко В. Ю.**  
(прізвище та ініціали)

Рецензент **Козуб Ю. Г.**  
(прізвище та ініціали)

Полтава – 2025 року

## ЗМІСТ

<b>ВСТУП.....</b>	<b>4</b>
<b>РОЗДІЛ І. АНАЛІТИЧНИЙ ОГЛЯД .....</b>	<b>7</b>
1.1. Дія кліматичних умов на внутрішній клімат серверного простору .....	7
1.2. Огляд наявних систем моніторингу кліматичних параметрів у приміщеннях .....	11
1.3. Стандарти та рекомендації з моніторингу умов у серверних приміщеннях .....	31
Висновки до розділу .....	34
<b>РОЗДІЛ 2. ВИБІР КОМПОНЕНТІВ ДЛЯ СИСТЕМИ МОНІТОРИНГУ МІКРОКЛІМАТУ СЕРВЕРНОЇ КІМНАТИ НА ОСНОВІ МІКРОКОНТРОЛЕРА ARDUINO .....</b>	<b>36</b>
2.1. Огляд основних компонентів системи .....	36
2.1.1. Огляд Arduino Mega .....	36
2.1.2. Опис PIR-датчика .....	41
2.1.3. Опис DHT11 .....	46
2.1.4. Дисплей LCD 1602 .....	48
2.1.5. Реле для комутації .....	53
2.1.6. RFID-зчитувач RC-522 .....	59
2.1.7. Ethernet Shield W5100 .....	62
2.2. Огляд середовищ розробки для програмування мікроконтролерів .....	65
2.2.1. Огляд Arduino IDE .....	65
2.2.2. Огляд Visual Studio .....	70
Висновки до розділу .....	72
<b>РОЗДІЛ 3. ПРОЄКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНО-АПАРATНОЇ СИСТЕМИ .....</b>	<b>73</b>
3.1. Розробка структури пристрою .....	73
3.2. Аналіз функціональних вимог .....	74

					<b>ІТС.4КІ.0125.03-ПЗ</b>		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Ліщук С.Е.			<b>ЗМІСТ</b>	Літ.	Арк.
Керівник		Донченко В.Ю.					2
Реценз.		Козуб Ю.Г.				ЛНУ	
Н. Контр.						Кафедра ІТС, Гр.4КІ	
Зав. каф.		Семенов М.А.					

3.3. Розробка програмно-апаратного комплексу .....	78
3.3.1. Проектування програмно-апаратного комплексу .....	78
3.3.2. Програмування датчиків та модулів .....	80
3.3.2.1. Використання бібліотек для розширення функціональності програмного забезпечення .....	83
3.3.2.2. Модуль температури та вологості .....	88
3.3.2.3. Модуль освітленості .....	90
3.3.2.4. Модуль контролю доступу .....	92
3.3.2.5. Модуль стаціонарної візуалізації .....	94
3.3.2.6. Модуль дистанційної візуалізації .....	97
3.3.3. Реалізація програмної частини мобільного застосунку .....	99
Висновки до розділу .....	107
<b>ВИСНОВКИ .....</b>	<b>109</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....</b>	<b>112</b>
<b>ДОДАТОК А.....</b>	<b>116</b>
<b>ДОДАТОК Б.....</b>	<b>119</b>
<b>ДОДАТОК В.....</b>	<b>120</b>
<b>ДОДАТОК Г .....</b>	<b>123</b>
<b>ДОДАТОК Д.....</b>	<b>126</b>
<b>ДОДАТОК Е.....</b>	<b>128</b>
<b>ДОДАТОК Є.....</b>	<b>131</b>
<b>ДОДАТОК Ж .....</b>	<b>133</b>
<b>ДОДАТОК З .....</b>	<b>135</b>
<b>ДОДАТОК И .....</b>	<b>138</b>
<b>ДОДАТОК І.....</b>	<b>140</b>

## ВСТУП

Сучасні підприємства значною мірою залежать від високотехнологічного інформаційного обладнання для забезпечення своєї операційної діяльності. Для гарантування надійного функціонування та безперебійної роботи цих критично важливих інформаційних систем, необхідним є розміщення телекомунікаційного обладнання у спеціально обладнаних серверних приміщеннях.

Ефективність та стабільність роботи серверних кімнат безпосередньо залежать від умов їх експлуатації, зокрема від мікрокліматичних параметрів. У цьому контексті набуває особливого значення розгляд інтегрованих систем контролю мікроклімату, які забезпечують оптимальні умови для підтримки надійності та ефективності функціонування обладнання.

Розвиток інформаційних технологій та широке впровадження концепції Інтернету речей (IoT) відкривають нові можливості для модифікації та удосконалення систем контролю параметрів мікроклімату. Це дозволяє досягти зменшення собівартості, розширення функціональних можливостей, а також підвищення точності та швидкодії таких систем.

**Об'єкт дослідження** – процес контролю параметрів мікроклімату серверної кімнати.

**Предмет дослідження** – автоматизована система контролю параметрів мікроклімату серверної кімнати на базі мікроконтролера Arduino.

**Мета дослідження** – аналіз та розробка системи моніторингу мікроклімату серверної кімнати на базі Arduino.

**Досягнення зазначеної мети передбачає вирішення таких основних завдань:**

- провести огляд літературних джерел та нормативних документів для розробки автоматизованої системи моніторингу мікроклімату серверних приміщень;

					ІТС.4КІ.0125.03-ПЗ				
Змн.	Арк.	№ докум.	Підпис	Дата	ВСТУП	Літ.	Арк.	Акрушів	
Розроб.		Ліщук С.Е.							
Керівник		Донченко В.Ю.					4	3	
Реценз.		Козуб Ю.Г.				ЛНУ Кафедра ІТС, Гр.4КІ			
Н. Контр.									
Зав. каф.		Семенов М.А.							

- зробити аналіз існуючих рішень для спостереження за мікрокліматом у серверних кімнатах;
- провести детальний аналіз основних компонентів та інструментів, необхідних для реалізації системи моніторингу мікроклімату серверної кімнати на базі платформи Arduino;
- розробити структурну схему пристрою;
- описати логіку функціонування програмних модулів системи моніторингу мікроклімату серверної кімнати на базі платформи Arduino.
- розробити мобільний застосунок "Blynk" для системи дистанційного моніторингу мікроклімату серверної кімнати.

Для вирішення поставлених задач використовувались методи моделювання та програмування; положення теорії автоматичного керування; методи створення автоматизованих вимірювальних мікропроцесорних систем; методи математичної статистики та обробки експериментальних результатів.

У першому розділі роботи було проведено огляд літературних джерел та нормативних документів для розробки автоматизованої системи моніторингу мікроклімату серверних приміщень, особливо з урахуванням потреб освітніх установ та малого бізнесу, які мають обмежені бюджети й не потребують надлишкових функцій, притаманних промисловим системам. Було ретельно досліджено законодавчу та нормативну базу, що регулює параметри внутрішнього клімату в серверних кімнатах. Розглянуто основні стандарти, зокрема ASHRAE, які визначають перелік параметрів, що підлягають обов'язковому контролю (температура, вологість), та їхні оптимальні значення для підтримки стабільної працездатності серверного обладнання. Окрім цього, було проведено зіставний аналіз існуючих рішень для спостереження за мікрокліматом у серверних кімнатах. Розглянуто їх функціональні можливості, переваги та недоліки. У підсумку, було сформульовано технічні вимоги до автоматизованої системи контролю мікроклімату серверного приміщення, які враховують не лише нормативні обмеження, але й потреби в гнучкості, доступності та простоті використання для малих організацій.

					ІТС.4КІ.0125.03-ПЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

У другому розділі було проведено детальний аналіз основних компонентів та інструментів, необхідних для реалізації системи моніторингу мікроклімату серверної кімнати на базі платформи Arduino. Було встановлено, що Arduino Mega2560 є оптимальним вибором для створення прототипів завдяки своїй доступності, простоті використання та широкій підтримці спільноти розробників. Датчики, такі як PIR-датчик, DHT11 та інші забезпечують збір необхідної інформації про навколишнє середовище, що є основою для прийняття рішень системою автоматизації. Для розробки програмного забезпечення було розглянуто Arduino IDE спеціалізоване середовище розробки, оптимізоване для роботи з платформою Arduino. Воно надає інтуїтивний інтерфейс та широкий набір бібліотек.

Третій розділ був присвячений вирішенню задачі прототипування системи моніторингу. В рамках цього етапу було розроблено структурну схему пристрою, здійснено обґрунтований вибір основного обладнання, проведено аналіз функціональних вимог з використанням діаграм варіантів використання, а також побудовано моделі, що описують різні аспекти системи, зокрема діаграму станів програмно-апаратного комплексу, діаграму компонентів та діаграму розгортання. Описана логіка функціонування програмних модулів системи моніторингу мікроклімату серверної кімнати на базі платформи Arduino. Розроблено мобільний застосунок "Blynk" для системи дистанційного моніторингу мікроклімату серверної кімнати.

Результати проведеного дослідження мають значний потенціал для практичного застосування. Отримані дані та розроблені алгоритми можуть бути використані для підвищення функціональності, надійності та енергоефективності існуючих систем "розумний будинок". Розроблені в рамках дослідження рішення можуть слугувати основою для створення нових, більш інтелектуальних та адаптивних систем моніторингу мікроклімату серверних кімнат на базі Arduino. Результати дослідження можуть сприяти популяризації технологій "розумного будинку" серед широкого кола користувачів, підвищуючи обізнаність про переваги та можливості таких систем.

					ІТС.4КІ.0125.03-ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

## РОЗДІЛ І. АНАЛІТИЧНИЙ ОГЛЯД

Серверні кімнати являють собою спеціалізовані простори, обладнані для розміщення серверних стійок і шаф з комунікаційним устаткуванням [1].



Рис. 1.1. Серверна кімната

Вихід обладнання з ладу або його тимчасове відключення можуть спричинити суттєві матеріальні та фінансові збитки [3]. Для стабільного функціонування необхідно здійснювати моніторинг і регулювання параметрів мікроклімату серверних приміщень у режимі реального часу. Розглянемо, яким чином атмосферні умови діють на внутрішній клімат серверної кімнати, та проаналізуємо доступні механізми керування цими показниками.

### 1.1. Дія кліматичних умов на внутрішній клімат серверного простору

Мікроклімат визначається як стан повітряного середовища всередині приміщення, що включає такі показники, як температура та вологість. Відповідно до джерела [3], запровадження дієвої системи спостереження та управління мікрокліматом у серверних приміщеннях сприяє запобіганню збоям у функціонуванні мережевого обладнання, пролонгації терміну його служби та оптимізації витрат на модернізацію серверної інфраструктури.

					<b>ІТС.4КІ.0125.03-ПЗ</b>		
Змн.	Арк.	№ докум.	Підпис	Дата	<b>РОЗДІЛ І. АНАЛІТИЧНИЙ ОГЛЯД</b>		
Розроб.	Ліщук С.Е.						
Керівник	Донченко В.Ю.						
Реценз.	Козуб Ю.Г.						
Н. Контр.							
Зав. каф.	Семенов М.А.				ЛНУ		
					Кафедра ІТС, Гр.4КІ		
					Лім.	Арк.	Акрушів
						7	28

Основні вимоги до організації серверних кімнат викладені у міжнародному стандарті TIA/EIA-569 [4], що був розроблений у співпраці Канади та США та здобув широке міжнародне визнання. Стандарт регламентує не лише архітектурні особливості приміщень, але й вимоги до освітлення, електропроводки, навантаження на підлогу, а також параметрів мікроклімату. Забезпечення належних кліматичних умов відповідно до встановлених параметрів є першочерговою вимогою стандарту.

В цілому, незалежно від особливостей конкретного серверного простору, обов'язковому моніторингу підлягають такі атмосферні параметри, як температура та відносна вологість повітря. Стандарт [4] регламентує їхні нормативні значення:

- температурний режим: від +18 °C до +27 °C;
- рівень відносної вологості (при 24 °C): від 40% до 60%.

Система підтримки мікроклімату повинна працювати безперебійно, забезпечуючи сталість цих показників протягом усього року. Далі буде розглянуто причини та наслідки впливу кліматичних факторів на серверне обладнання (Таблиця 1.1).

Таблиця 1.1

### Вплив кліматичних факторів на працездатність обладнання серверних

Кліматичний фактор	Причини впливу на обладнання серверної кімнати	Наслідки впливу кліматичних факторів на обладнання серверної кімнати
Температура	1. Висока теплогенерація активних компонентів: процесори, блоки живлення, жорсткі диски та інші елементи серверів під час роботи виділяють значну кількість тепла. 2. Неефективна робота систем охолодження: недостатня потужність кондиціонерів, забиті	1. Термічне пошкодження компонентів: перевищення допустимих температурних меж призводить до деградації напівпровідникових матеріалів, висихання конденсаторів, пошкодження підшипників вентиляторів. 2. Зниження продуктивності: при високих температурах може спостерігатися троттлінг

Кліматичний фактор	Причини впливу на обладнання серверної кімнати	Наслідки впливу кліматичних факторів на обладнання серверної кімнати
	<p>фільтри або несправність вентиляційних систем.</p> <p>3. Зовнішні теплові надходження: сонячне випромінювання через вікна, тепло від сусідніх приміщень або недостатня теплоізоляція стін.</p> <p>4. Скупчення обладнання: занадто щільне розміщення серверних стійок та недостатня циркуляція повітря між ними.</p>	<p>процесорів (зниження тактової частоти) для запобігання перегріву.</p> <p>3. Зменшення терміну експлуатації: постійний вплив високих температур значно скорочує середній час напрацювання на відмову (MTBF) електронних компонентів.</p> <p>4. Нестабільна робота: перегрів може викликати випадкові зависання, перезавантаження або некоректну роботу програмного забезпечення.</p>
<b>Вологість</b>	<p>1. Недостатня вентиляція: погана циркуляція повітря перешкоджає виведенню надлишкової вологи.</p> <p>2. Конденсація: утворення конденсату на холодних поверхнях обладнання при різкій зміні температури.</p> <p>3. Висока зовнішня вологість: проникнення вологого повітря ззовні при недостатній герметичності приміщення.</p> <p>4. Протікання або вологість у сусідніх приміщеннях: підвищена вологість може проникати через стіни, підлогу або стелю.</p>	<p>1. Електричні короткі замикання: конденсат або висока вологість можуть призвести до утворення провідних містків між електричними контактами.</p> <p>2. Корозія металевих елементів: підвищена вологість сприяє окисленню та корозії роз'ємів, контактів та інших металевих частин.</p> <p>3. Вихід з ладу електронних компонентів: корозія та короткі замикання можуть призвести до пошкодження мікросхем, плат та інших чутливих елементів.</p> <p>4. Зростання ймовірності відмов: нестабільна робота через вологість значно підвищує ризик виходу з ладу критично важливих компонентів.</p>

У дослідженнях [5-9] було проаналізовано вплив параметрів кліматичного забезпечення на енергоспоживання серверного обладнання (див. рис. 1.2).

					<b>ІТС.4КІ.0125.03-ПЗ</b>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

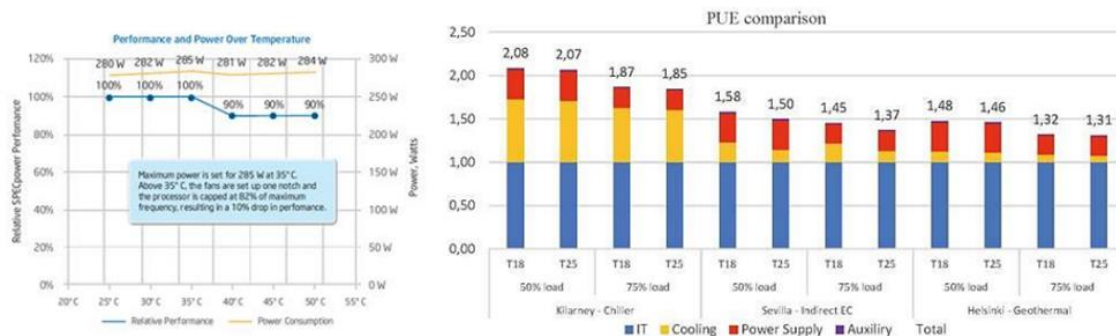


Рис. 1.2. Дослідження впливу температурно-вологісного режиму на енергоспоживання серверного обладнання

Рисунок 1.2 демонструє взаємозв'язок між температурним режимом внутрішнього повітряного середовища та двома ключовими аспектами: енергоспоживанням і ефективністю роботи серверного обладнання. Ефективність тут кількісно оцінюється за допомогою показника PUE (Power Usage Effectiveness), формула для обчислення якого наведена як (1.1) і підтверджена джерелами [9–17].

$$PUE = \frac{TFP}{ITEP} \quad (1.1)$$

де:

TFP (Total Facility Power) – загальна споживана потужність дата-центру;

ITEP (IT Equipment Power) – потужність, споживана ІТ-обладнанням (серверним обладнанням).

Результати досліджень, наведені на рисунку 1.2, свідчать, що підвищення температури внутрішнього повітряного середовища дата-центру призводить до зростання загального енергоспоживання (TFP), необхідного для підтримання нормативного функціонування серверного обладнання відповідно до його технічних вимог. Це, у свою чергу, спричиняє поступове зниження показника PUE, що свідчить про зменшення енергоефективності.

Оптимальні значення PUE визначаються згідно з концепцією, представленою на рисунку 1.3 [9–17].



Рис. 1.3. Концепція визначення оптимального рівня PUE (Power Usage Effectiveness)

На основі досліджень, представлених у роботах [5–9], встановлено, що зі зростанням температури та вологості повітря відбувається зниження показника енергоефективності PUE. Таким чином, підтримання мікрокліматичних параметрів у межах, регламентованих стандартом [4], є ключовим фактором для забезпечення стабільної та ефективної експлуатації серверного обладнання.

## 1.2. Огляд наявних систем моніторингу кліматичних параметрів у приміщеннях

Патент UA 112127 U демонструє приклад автоматизованої системи регулювання параметрів мікроклімату в приміщенні. У цій системі передбачено регулювання кліматичних умов за допомогою мікропроцесорного контролера, який отримує дані від сенсорів температури та вологості повітря. Крім того, система оснащена механізмом для відведення конденсату. Основними недоліками цієї конструкції є відсутність засобів контролю газового складу повітря, а також неможливість передавання даних через інтернет-мережу.

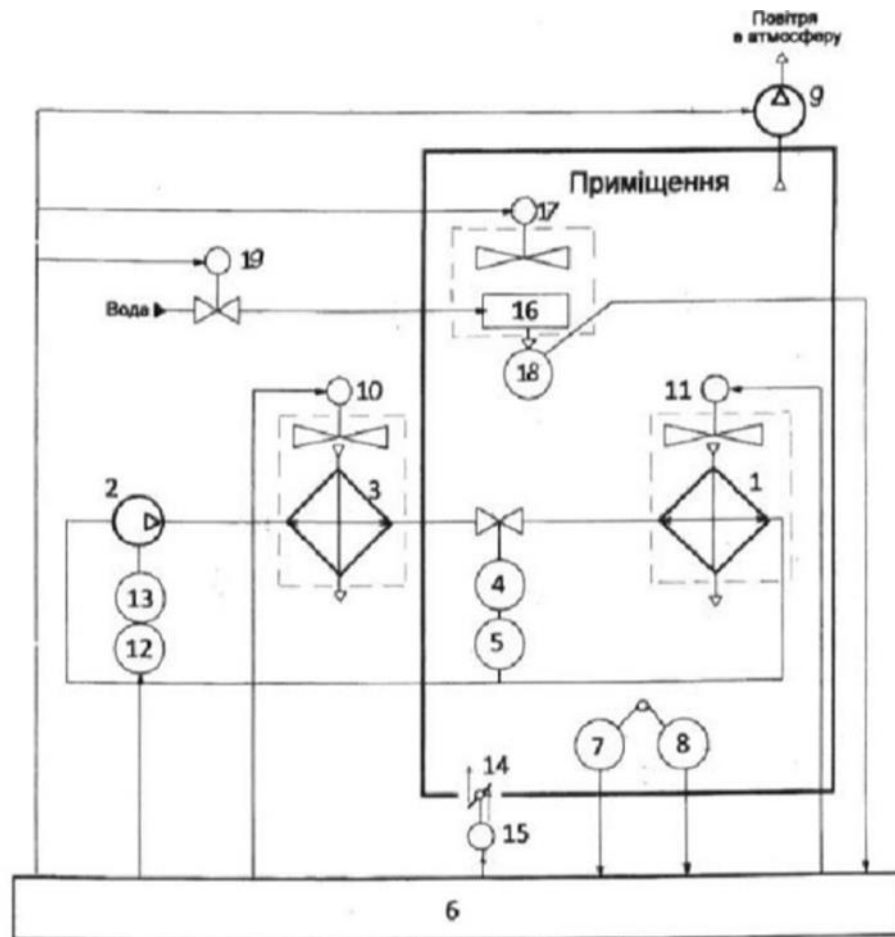


Рис. 1.4. Система автоматичного регулювання мікрокліматичних параметрів у приміщенні (патент UA 112127 U) [5]

Згідно з джерелом [5], основні функціональні блоки цієї системи включають:

**Холодильний контур:** випарник (1), компресор (2), конденсатор (3), терморегулюючий вентиль (4), термобалон (5).

**Система керування:** мікропроцесорний контролер (6), датчик температури (7), датчик відносної вологості повітря (8).

**Вентиляція:** витяжний вентилятор (9), вентилятор конденсатора (10), електродвигуни вентиляторів (17), кватирка для подачі свіжого повітря (14), виконавчий механізм кватирки (15).

**Регулювання продуктивності:** частотний перетворювач (12), електродвигун компресора (13), випарники (11).

**Зволоження повітря:** ультразвукові зволожувачі повітря (16), датчик температури повітря на виході зволожувача (18), клапан подачі води (19).

У патенті UA 101943 U представлено іншу систему автоматизованого регулювання мікроклімату, що функціонує з урахуванням циркадного ритму (структурна схема зображена на рис. 1.5). Дана система орієнтована на застосування в аграрному секторі, зокрема для контролю мікроклімату в тваринницьких приміщеннях. Управління здійснюється мікроконтролером, що забезпечує моніторинг температури, відносної вологості повітря, а також концентрації газів, зокрема аміаку.

Основними недоліками цієї системи є відсутність можливості регулювання швидкості руху повітря, а також, як і у попередньому рішенні, відсутність функції передавання даних через мережу Інтернет.

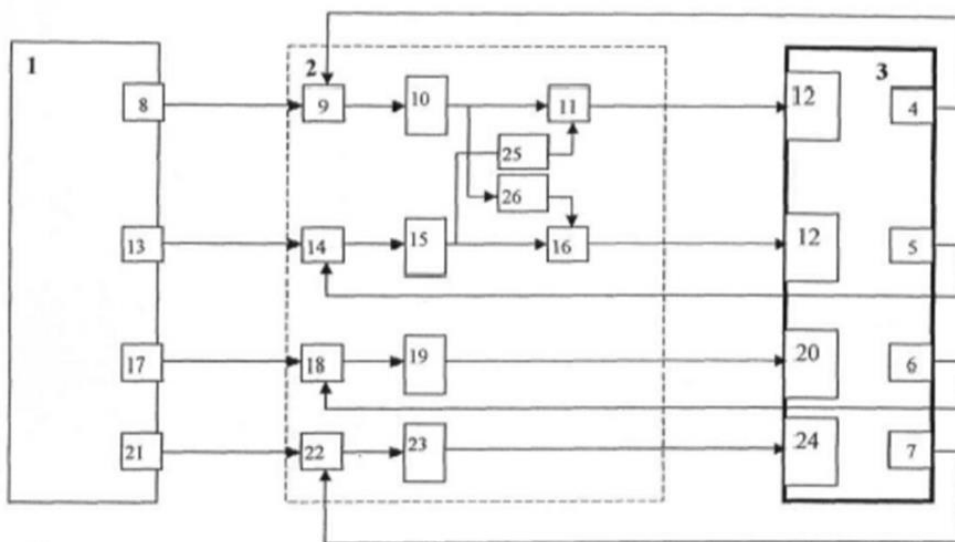


Рис. 1.5. Система автоматичного регулювання мікрокліматичних параметрів у приміщенні (патент UA 101943 U) [6]

Система складається з таких основних елементів [6]:

- 1 – мікропроцесор;
- 2 – блок аналізу та регулювання параметрів;
- 3 – тваринницьке приміщення;
- 4 – датчик температури повітря;
- 5 – датчик відносної вологості повітря;
- 6 – датчик концентрації аміаку;
- 7 – датчик рівня вуглекислого газу;
- 8 – задавач температури для мікропроцесора;

- 9,14,18,22 – блоки порівняння (суматори);
- 10,15,19,23 – регулятори;
- 11,16 – коригувальні суматори;
- 12 – виконавчий механізм для регулювання температури та вологості;
- 13 – задавач вологості для мікропроцесора;
- 17 – задавач допустимого рівня вуглекислого газу;
- 20 – виконавчий механізм регулювання концентрації вуглекислого газу;
- 21 – задавач допустимого рівня аміаку;
- 24 – виконавчий механізм регулювання концентрації аміаку;
- 25,26 – блоки коригувальних сигналів зворотного зв'язку.

У патенті UA 72565 U запропоновано метод автоматичного регулювання мікроклімату в приміщеннях шляхом керування подачею теплоносія в секцію нагріву кондиціонера. Дана система орієнтована на застосування в будівництві та експлуатації виробничих і житлових будівель. Подібно до попередніх рішень, регулювання мікрокліматичних параметрів (температури та вологості повітря) здійснюється мікропроцесором. Однак на відміну від інших систем, ця конструкція дозволяє додатково керувати швидкістю повітряного потоку. Водночас основним недоліком залишається відсутність функції передачі даних через Інтернет.

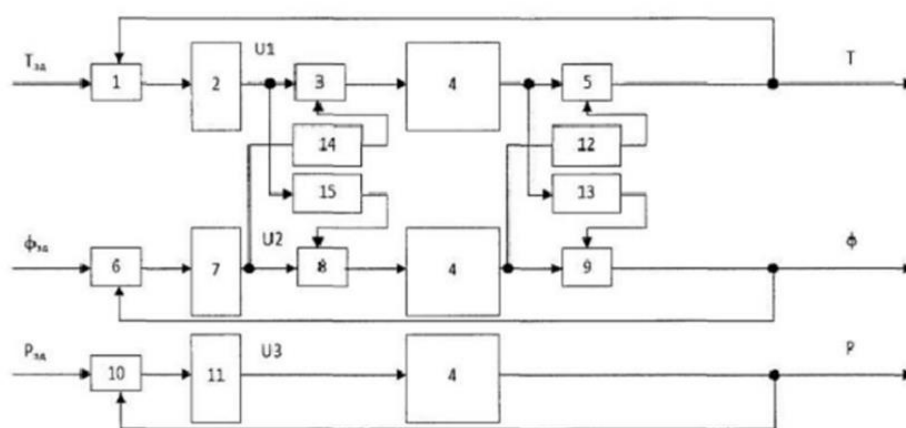


Рис. 1.5. Система автоматичного регулювання мікрокліматичних параметрів у приміщенні (патент UA 72565 U) [7]

Система складається з таких основних елементів [7]:

1, 3, 5, 6, 8, 9, 10 – суматори;

					ІТС.4КІ.0125.03-ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

2, 7, 11 – регулятори;

4 – об'єкт управління;

12, 13 – елементи перехресного зв'язку;

14, 15 – коригувальні зв'язки;

U1, U2, U3 – керуючі дії (у відсотках від умовного регульованого об'єму);

Tзд – задане значення температури;

Фзд – задане значення вологості повітря;

Pзд – задане значення розрідження повітря;

T, Ф, P – відповідно поточні значення температури, вологості та розрідження повітря.

Патентний аналіз [10–15] показав, що за своєю структурою більшість існуючих систем контролю мікроклімату мають схожі архітектурні рішення. В цілому системи регулювання кліматичних умов виробничих приміщень, як правило, складаються з трьох основних типів компонентів:

Контролер – керуючий пристрій, що координує роботу системи та забезпечує її взаємодію із зовнішнім середовищем;

Датчики – пристрої, які фіксують поточні параметри навколишнього середовища;

Виконавчі механізми – елементи, що реалізують команди контролера для регулювання параметрів мікроклімату.

Різниця у функціональності систем, розглянутих у роботах [10–15], визначається перш за все типами використаних датчиків, методами опрацювання зібраних даних та архітектурою реалізації зворотного зв'язку між окремими блоками.

Наприклад, у роботі [10] описується комп'ютеризована система контролю та регулювання мікроклімату робочого місця, яка здійснює моніторинг температури, вологості та рівня освітленості завдяки використанню комп'ютера з відповідним програмним забезпеченням. Структурну схему цієї системи наведено на рисунку 1.6.

					ІТС.4КІ.0125.03-ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

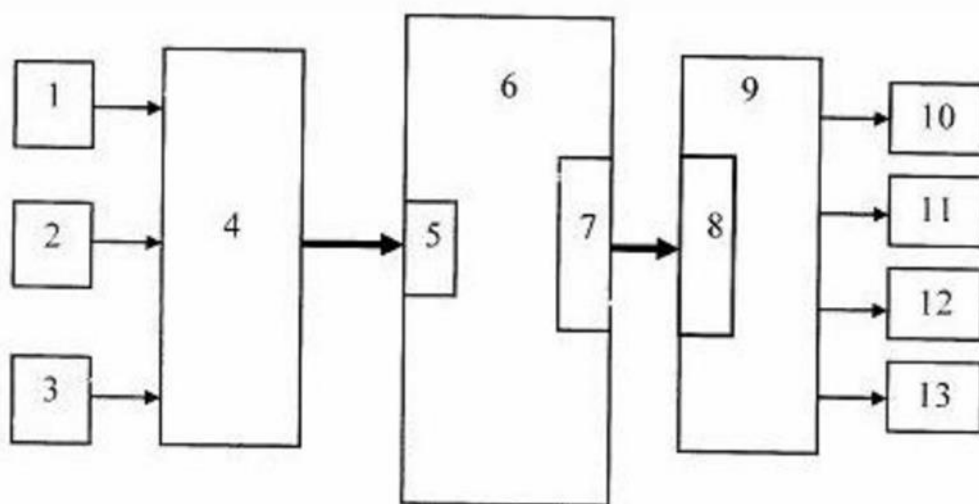


Рис. 1.6. Структурна схема комп'ютеризованої системи моніторингу та керування мікрокліматом робочого місця

Архітектура системи включає: сенсор температури (1), датчик освітленості (2) та гігрометр (3), вихідні сигнали яких інтегруються в мікроконтролерному інформаційно-вимірювальному блоці (4). Дані з цього блоку передаються через послідовний інтерфейс (5) до обчислювальної машини (6). Паралельний порт (7) комп'ютера через комунікаційний канал (8) з'єднаний з входом керуючого пристрою (9), що здійснює управління виконавчими механізмами: освітлювальною системою робочої зони (10), нагрівачем (11), зволожувачем (12) та охолоджувачем повітря (13).

В описуваній системі інформація, отримана від сенсорів, передається до комп'ютера для подальшого аналізу. На основі опрацьованих даних генеруються керуючі сигнали, що використовуються для регулювання параметрів мікроклімату.

У роботі [11] представлено пристрій, призначений для визначення параметрів мікроклімату та концентрації шкідливих речовин у повітрі. Зазначена система забезпечує одночасне автоматизоване вимірювання дванадцяти показників, серед яких: температура, відносна вологість, атмосферний тиск, рівень освітленості, а також вміст таких небезпечних газів, як монооксид вуглецю, діоксид вуглецю, аміак, сірководень, оксиди азоту (I) та (IV), метан і формальдегід. Збір та обробка первинних даних здійснюються

мікроконтролером, а формування керуючих сигналів покладається на вторинний прилад. Структурна схема цієї системи детально зображена на рисунку 1.7.

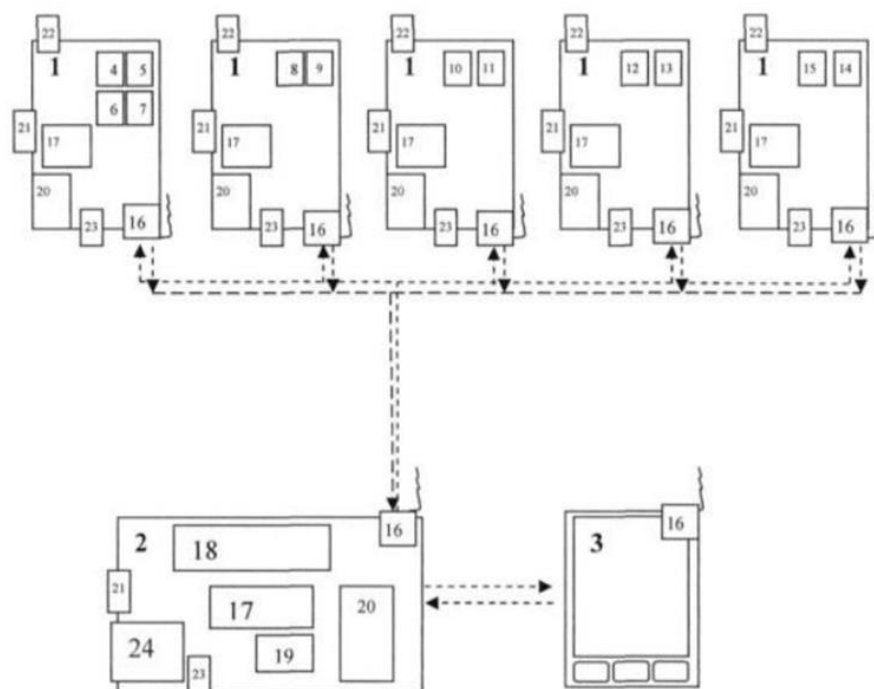


Рис. 1.7. Структурна схема системи для моніторингу мікрокліматичних параметрів та вмісту шкідливих речовин у повітрі

До складу системи входять п'ять блоків для здійснення вимірювань (1), а також модулі, що відповідають за керування (2) та обробку даних (3). Для забезпечення бездротової взаємодії між цими елементами використовується радіоадаптер бездротової мережі (16), що дозволяє підключати всі вимірювальні блоки до модулів керування та аналізу даних через радіоканал. Це суттєво розширює можливості застосування подібної системи в різних умовах.

Попри те, що система має надлишковий набір сенсорів для завдань моніторингу мікроклімату серверних приміщень, її концепція використання кількох ідентичних модулів забезпечує розширений контроль простору із застосуванням лише одного мікроконтролера. Додатковою перевагою є можливість бездротової передачі даних, що значно полегшує встановлення сенсорів у важкодоступних місцях.

Серед загальних недоліків систем, що розглядаються в джерелах [10–13, 15], слід виділити складну архітектуру, низьку гнучкість у застосуванні та значне енергоспоживання.

Патент [14] пропонує електронний аналізатор параметрів мікроклімату для системи "розумний дім", який, на відміну від попередніх рішень, не має складної архітектури, є більш гнучким у використанні та має знижене енергоспоживання. Цей прилад включає модулі для вимірювання температури, тиску, вологості та освітленості, а також блок обробки даних і радіочастотний модуль для обміну інформацією та отримання керуючих сигналів. Структурну схему цієї розробки можна побачити на рисунку 1.8.

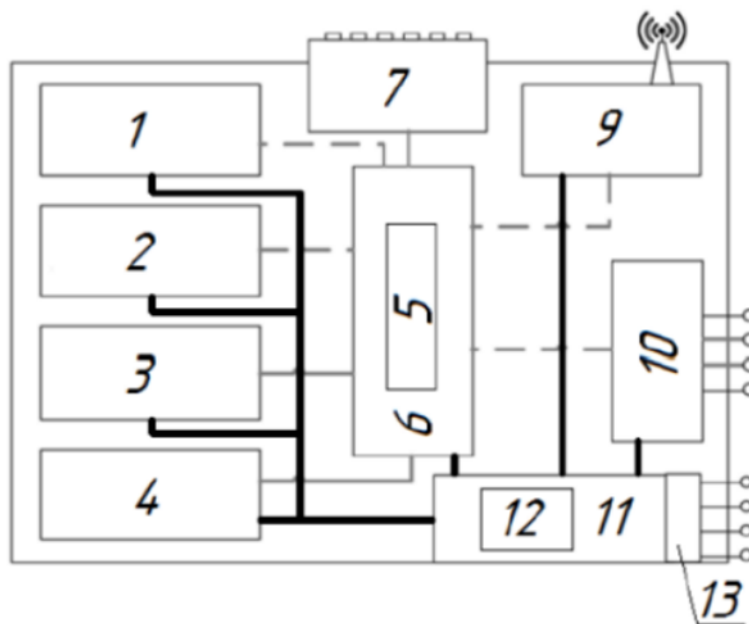


Рис. 1.8. Схема будови аналізатора мікроклімату для "розумного дому"

Електронний аналізатор мікроклімату приміщення містить такі основні компоненти: датчик температури (1), сенсор атмосферного тиску (2), датчик вологості повітря (3) та сенсор освітленості (4). Усі вони підключені до модуля обробки інформації (5), який обладнаний пристроєм для конвертації сигналів і формування інформаційного пакету (6). Цей модуль також підключений до блоку для введення ідентифікаторів (7), який має двопозиційні перемикачі (8), а також до радіопередавача (9) та шини для обміну даними (10). Живлення всіх елементів аналізатора забезпечує блок електроживлення (11), який включає контролер подачі живлення (12) та порт Micro-USB (13), що підключається до спільної енергомережі.

Для ідентифікації типів з'єднань у схемі використано такі позначення:

- суцільна лінія – аналоговий провідний канал обміну даними;

- штрихова лінія – цифровий провідний канал обміну даними;
- радіосигнал – радіочастотний канал обміну даними;
- товста суцільна лінія – мережа живлення.

На рисунку 1.9 наведено схему конструкції перемикачів, призначених для налаштування ідентифікаторів електронного аналізатора [14].

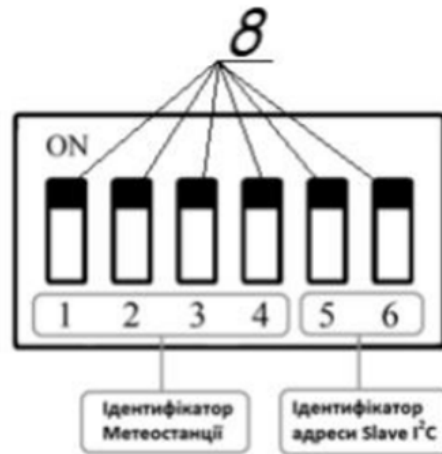


Рис. 1.9. Схема конструкції перемикачів

Запропонований електронний аналізатор передбачає можливість налаштування рівня потужності радіочастотного сигналу та частоти передавання даних. Такі функції дозволяють оптимізувати режим роботи пристрою відповідно до специфіки його застосування та забезпечити раціональне використання електроенергії.

Конструкція цієї метеостанції дає змогу підключати до одного контролера до 16 пристроїв, що можуть розташовуватись на відстані до 80 метрів від нього. Це дозволяє здійснювати моніторинг мікрокліматичних параметрів на значних площах, зокрема понад 19 000 м<sup>2</sup> [14].

Завдяки використанню простих інтерфейсів для передавання даних та відсутності вбудованого контролера аналізу, конструкція пристрою є спрощеною, що позитивно впливає на зниження вартості його виготовлення. Завдяки можливості підключення метеостанції до зовнішнього контролера, її функціонал значно розширюється.

Таким чином, розроблений електронний аналізатор мікроклімату для "розумного дому" є простим, надійним, недорогим у виробництві та зручним у використанні.

					ІТС.4КІ.0125.03-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

Пристрій здатен ефективно збирати всі необхідні дані щодо мікроклімату, що дає змогу формувати сценарії для автоматизованого управління інженерною інфраструктурою будинку з метою підтримання оптимальних умов [14].

Цей аналізатор є універсальним за способами підключення та передавання даних до контролера системи «розумний будинок», а також має гнучкі можливості щодо живлення. Пристрій характеризується високою енергоефективністю, автономністю та автоматизованим налаштуванням параметрів роботи. Конструктивно передбачені ідентифікатори дозволяють охопити значну площу контролю мікроклімату та передавати дані до єдиного контролера [14].

У дослідженні [16] авторами було реалізовано систему моніторингу мікроклімату на базі двох мікроконтролерів Arduino, де для вимірювання температури використовувалися датчики на основі терморезистора DHT11.

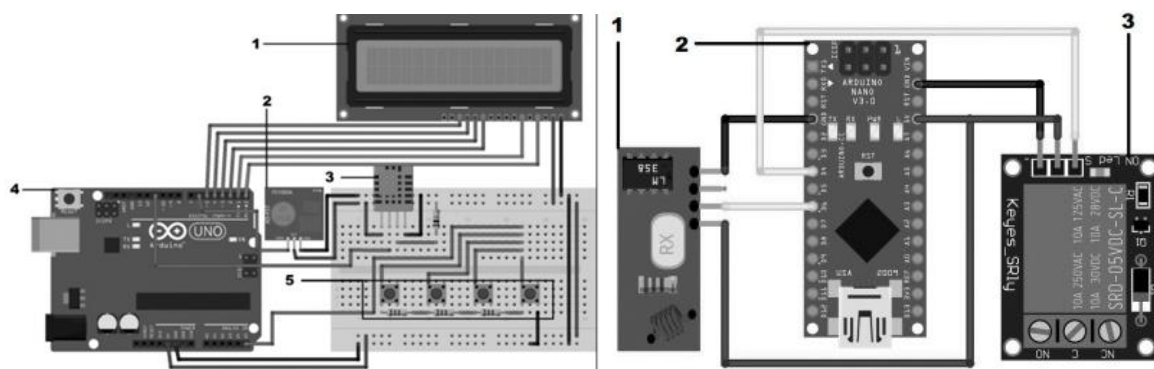


Рис. 1.10. Передатчик та приймач

Ця система відзначається простотою конструкції та конкурентоспроможною собівартістю, однак вона не є придатною для використання в серверних кімнатах (СК), оскільки не забезпечує контроль вологості повітря. На основі аналізу можна дійти висновку, що за структурною організацією системи моніторингу мікрокліматичних параметрів у приміщеннях загалом підходять для застосування в СК. Вони, як правило, побудовані на основі мікроконтролерів або мікропроцесорів, до яких підключаються сенсори різних типів.

Вибір типу сенсорів, а також їх кількість залежать від конкретних задач, конфігурації та розмірів серверного приміщення. Водночас параметри

температури та вологості повітря залишаються базовими й контролюються в усіх випадках незалежно від особливостей системи.

Наприклад, у дослідженні [17] авторами була розроблена система моніторингу мікрокліматичних параметрів серверної кімнати, побудована на основі популярної безкоштовної платформи для автоматизації Home Assistant з відкритим вихідним кодом (див. рис. 1.11).

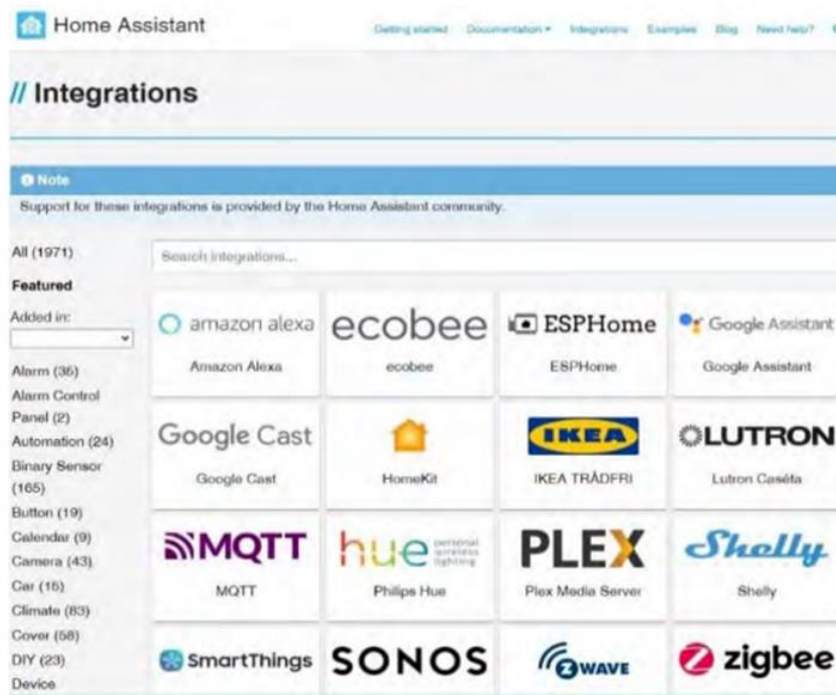


Рис. 1.11. Головна сторінка платформи Home Assistant

За допомогою платформи Home Assistant ефективно організовується автоматизований контроль серверних приміщень, що забезпечує моніторинг, створення автоматичних сценаріїв та сповіщення про події через SMS або відео. Для безперервного спостереження за станом серверної інфраструктури використовується комплекс датчиків: два сенсори температури та вологості, детектор задимлення, датчик контролю чистоти повітря, а також дві відеокамери з функціями інфрачервоного бачення та звукового оповіщення при фіксації руху. Для контролю температурного режиму в серверному приміщенні використовуються інтелектуальні Wi-Fi датчики, зокрема Aqara Aags-soi та Wsdcgqi Ilm. У разі несправностей порівняння даних із кількох сенсорів дозволяє оперативно виявити відхилення у показаннях.

Інтеграція всіх сенсорів та пристроїв в єдиному інтерфейсі Home Assistant, що доступний як через веб-браузер, так і мобільний додаток, створює зручну платформу для побудови автоматизованих систем моніторингу мікроклімату в серверних кімнатах. Проте цей підхід вимагає використання високотехнологічних датчиків і постійного підключення до платформи, що не завжди є доцільним для реалізації систем кліматичного контролю.

На українському ринку системи моніторингу мікроклімату серверних кімнат пропонуються кількома компаніями. Наприклад, компанія Кластер [18] розробила систему спостереження за температурними та вологісними параметрами СК. При виявленні перевищення допустимих значень сенсори активують механізм автоматичного надсилання SMS-повідомлення відповідальному користувачу, інформуючи про поточний стан температури та вологості в серверному приміщенні. Крім того, ця система може керувати кондиціонерами та інтегруватися із системою звукового сповіщення. Серед основних переваг рішення – простота конструкції, доступність та невисока вартість. До недоліків відноситься відсутність можливості здійснювати моніторинг у режимі реального часу.

Ще одна розробка – система SNMP-моніторингу серверної кімнати від компанії KVAZARMICRO [19], призначена для обслуговування невеликих серверних приміщень, що вміщують від п'яти до десяти локальних серверів (див. рисунок 1.12). Головна мета цієї системи полягає у безперервному локальному контролі як стану серверного обладнання, так і параметрів мікроклімату в серверній кімнаті. Інформування про критичні ситуації здійснюється в режимі реального часу як локально (за допомогою звукових сигналів та повідомлень на дисплеї пристрою), так і віддалено через мобільний додаток (Android/iOS) та месенджер Telegram.

					ІТС.4КІ.0125.03-ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		



Рис. 1.12. Система SNMP моніторингу мікроклімату серверної кімнати

Дана система забезпечує моніторинг серверного обладнання із використанням протоколу SNMP. Вона дозволяє отримувати широкий спектр даних про роботу серверів, зокрема:

- температурні показники процесорів, чіпсетів та мостів;
- швидкість обертання вентиляторів;
- стан блоків живлення;
- тривалість безперервної роботи серверів (uptime).

Окрім цього, система надає інформацію про uptime операційної системи, що дозволяє виявити ситуації, коли сервер функціонує, але ОС зависла. Також здійснюється моніторинг мікроклімату серверної кімнати, включаючи температурний режим, рівень вологості, стан дверей (із можливістю підключення датчика руху) та контроль витоків води. Система підтримує інтеграцію з UPS для фіксації переходу на автономне живлення, а також здійснює моніторинг критичних мережевих пристроїв, таких як Switch або Router (за наявності SNMP-підтримки).

Серед основних переваг цієї системи:

- Оповіщення операторів про аварійні ситуації у реальному часі;
- Попередження збоїв у роботі обладнання (наприклад, завчасне повідомлення про перегрів процесора);
- Високий рівень автономності, що зменшує залежність від людського фактору та дозволяє обійтися без постійної присутності локального адміністратора;

- Простота інсталяції та експлуатації.

Основним недоліком є висока вартість системи.

У джерелі [20] наведено результати аналітичного огляду програмних рішень, призначених для моніторингу параметрів навколишнього середовища в серверних кімнатах. Оцінка проводилася на основі наступних критеріїв:

- наявність локальних сповіщень для оператора;
- підтримка інформування через електронну пошту або SMS-повідомлення;
- специфіка програмно-апаратної реалізації;
- простота конфігурації та технічного обслуговування;
- здатність інтеграції з різноманітними типами сенсорів;
- наявність безкоштовної пробної версії або гарантії повернення коштів протягом тестового періоду;
- оптимальне співвідношення вартості та функціональності.

На підставі зазначених параметрів було визначено найкращі системи моніторингу середовища для серверних приміщень, однією з яких є NETMON [21].

**NETMON** – це рішення для моніторингу навколишнього середовища в дата-центрах та серверних кімнатах, яке забезпечує контроль таких параметрів, як температура, швидкість повітряного потоку, витoki води, електроживлення, безпека, а також виявлення диму. Система управляється через централізовану панель управління (див. рис. 1.13).

NETMON застосовує серію сенсорних пристроїв Аксп SensorProbe, спеціально розроблених для моніторингу серверних стійок та оперативного інформування про зміни параметрів середовища.

Серед переваг цієї системи:

- наявність фізичного пристрою для збору даних;
- адаптація для серверних кімнат різного масштабу;
- сповіщення оператора локально на робочому місці;
- підтримка надсилання SMS-повідомлень користувачам;

					<i>ITC.4KI.0125.03-ПЗ</i>	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

- можливість моніторингу продуктивності системи;
- широкий спектр сенсорів для контролю всіх необхідних параметрів середовища;
- наявність демоверсії для тестування.

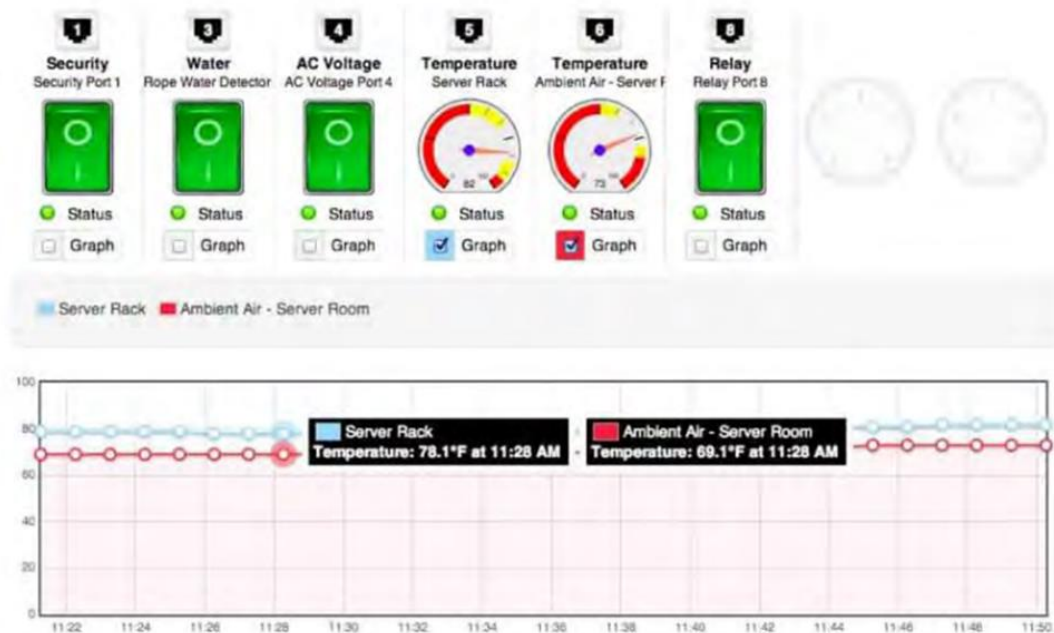


Рис. 1.13. Централізована панель управління системи Netmon

Основним недоліком цієї системи є її висока вартість.

### TEMPCUBE [22]

TEMPCUBE (див. рис. 1.14) – це бюджетне рішення для передачі даних сенсорів серверної кімнати до системи моніторингу. Пристрій використовує бездротову технологію для обміну даними, а програмне забезпечення сумісне з мобільними пристроями, що дозволяє здійснювати моніторинг із будь-якої точки.

Це рішення підходить для підприємств різного масштабу, хоча найефективніше проявляє себе в умовах великих організацій із розгалуженою серверною інфраструктурою.

Основні переваги TEMPCUBE:

- Автономність – пристрій може бути встановлений у будь-якому місці завдяки вбудованому акумулятору;
- Збір даних щодо температури та вологості через Wi-Fi з подальшою передачею до системи моніторингу;

Змн.	Арк.	№ докум.	Підпис	Дата

ІТС.4КІ.0125.03-ПЗ

Арк.

25

- Дистанційний контроль – можливість перевіряти стан системи через мобільний додаток;
- Гнучке налаштування порогових значень температури та вологості з автоматичним оповіщенням у разі їх перевищення;
- Сповіщення здійснюється за допомогою SMS або електронної пошти.

Це робить TEMPCUBE доступною альтернативою для організації базового контролю мікроклімату в серверних кімнатах.

Недоліки: не має пробної безкоштовної версії; ціна.

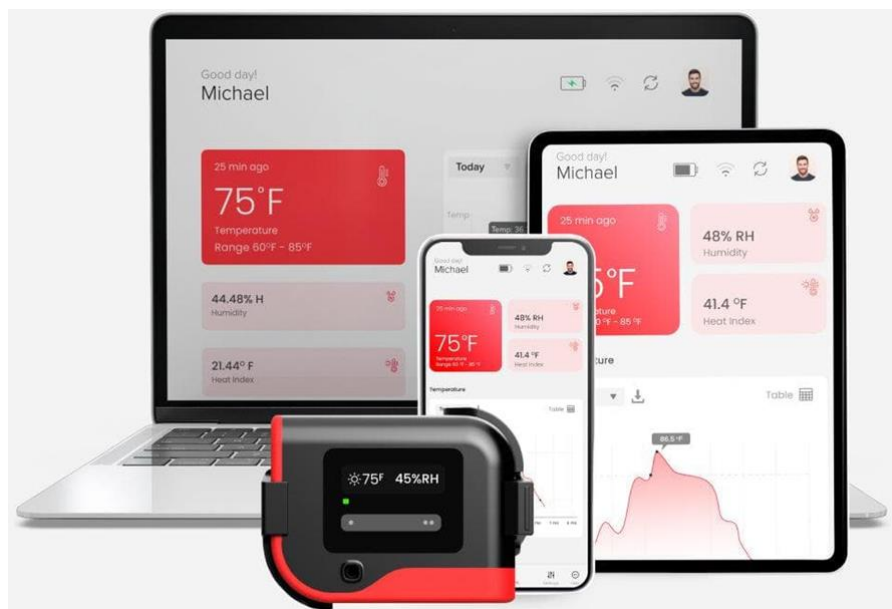


Рис. 1.14. Сенсорний пристрій tempCube

## MONNIT [23]

Система MONNIT призначена для комплексного моніторингу як мікрокліматичних умов, так і стану обладнання, розміщеного в серверних кімнатах. Рішення базується на використанні бездротових сенсорів, які фіксують показники навколишнього середовища та передають дані через шлюз на централізовану платформу моніторингу (див. рис. 1.15).

Незважаючи на те, що система виглядає доволі складною, її розгортання займає менше 15 хвилин, що значно спрощує процес налаштування. Після встановлення управління системою здійснюється віддалено, а для зручності

					ІТС.4КІ.0125.03-ПЗ	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

моніторингу існують спеціалізовані мобільні додатки для платформ Android та iOS.

#### Переваги MONNIT:

- здатність контролювати до 70 різних параметрів;
- використання бездротових технологій передачі даних;
- можливість моніторингу мікрокліматичних умов у реальному часі.

#### Недоліки:

- відносно висока вартість;
- складність конфігурації для деяких користувачів.

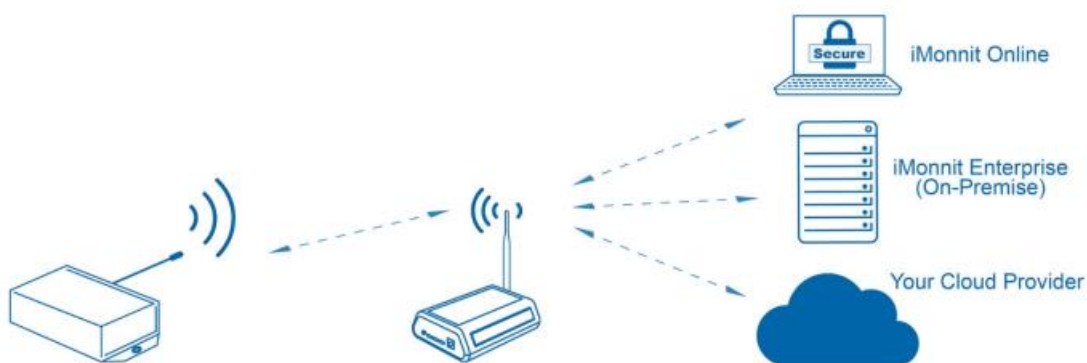


Рис. 1.15. Система моніторингу з використанням бездротових сенсорів **ITWATCHDOGS** [24]

Система моніторингу ITWatchDogs є високоефективним рішенням для контролю умов у серверних кімнатах, що забезпечує надійне керування інфраструктурою даних [24].

Продукт ITWatchDogs від Vertiv Geist (див. рис. 1.16) розроблено компанією Vertiv Group, яка спеціалізується на обладнанні для серверних приміщень і центрів обробки даних.

Однією з основних переваг системи є її здатність до комплексного контролю параметрів навколишнього середовища. ITWatchDogs забезпечує точне вимірювання температури, рівня вологості, швидкості повітряного потоку, а також має можливість виявляти присутність води або інших рідин у серверному приміщенні. Завдяки цьому адміністраторам вдається оперативно реагувати на зміни мікроклімату, що можуть загрожувати стабільній роботі обладнання,

зокрема в разі підвищення температури, яка може призвести до перегріву серверів.



Рис. 1.16. Зовнішній вигляд системи ITWatchDogs

Система ITWatchDogs додатково оснащена функціями забезпечення фізичної безпеки, такими як датчики руху та магнітні контакти для контролю стану дверей, що створює додатковий рівень захисту від несанкціонованого доступу до серверних приміщень. Однією з ключових переваг цього рішення є висока гнучкість у налаштуванні сповіщень і тривоги: користувач може вибирати способи отримання повідомлень – електронною поштою, через SMS або за допомогою мобільних додатків.

Ще однією важливою особливістю ITWatchDogs є можливість інтеграції з іншими системами моніторингу та управління дата-центрами, що дозволяє централізовано контролювати всю ІТ-інфраструктуру підприємства. Така інтеграція забезпечує ефективну взаємодію між різними компонентами системи, спрощуючи адміністрування та посилюючи загальну надійність роботи.

Завдяки поєднанню широких можливостей моніторингу довкілля, функцій безпеки та адаптивних налаштувань оповіщення, ITWatchDogs є ефективним інструментом для забезпечення стабільності й захищеності серверних кімнат. Проте через свою високу вартість це рішення більше орієнтоване на великі комерційні структури, і є малопридатним для впровадження в освітніх установах.

Наступна система демонструє приклад більш бюджетного рішення для моніторингу температури, вологості та наявності руху [3]. Це самостійно зібране рішення, яке має певні обмеження порівняно з комерційними системами.

Система складається з таких компонентів:

- сенсори температури та вологості DHT-22;
- датчик диму MQ-2;
- мікроконтролер Arduino із модулем Ethernet Shield;
- вентилятор постійного струму;
- центр моніторингу (див. рис. 1.17).

Програмна частина включає веб-додаток, серверну частину та базу даних, що дозволяє здійснювати моніторинг і керування через вебінтерфейс. Система також підтримує віддалене сповіщення через GSM-мережу, що забезпечує зручний доступ до інформації та оперативне отримання повідомлень про критичні зміни. Завдяки поєднанню вебінтерфейсу та GSM-оповіщень рішення є доступним і зручним у використанні, хоча й поступається комерційним аналогам за рівнем функціональності та надійності.

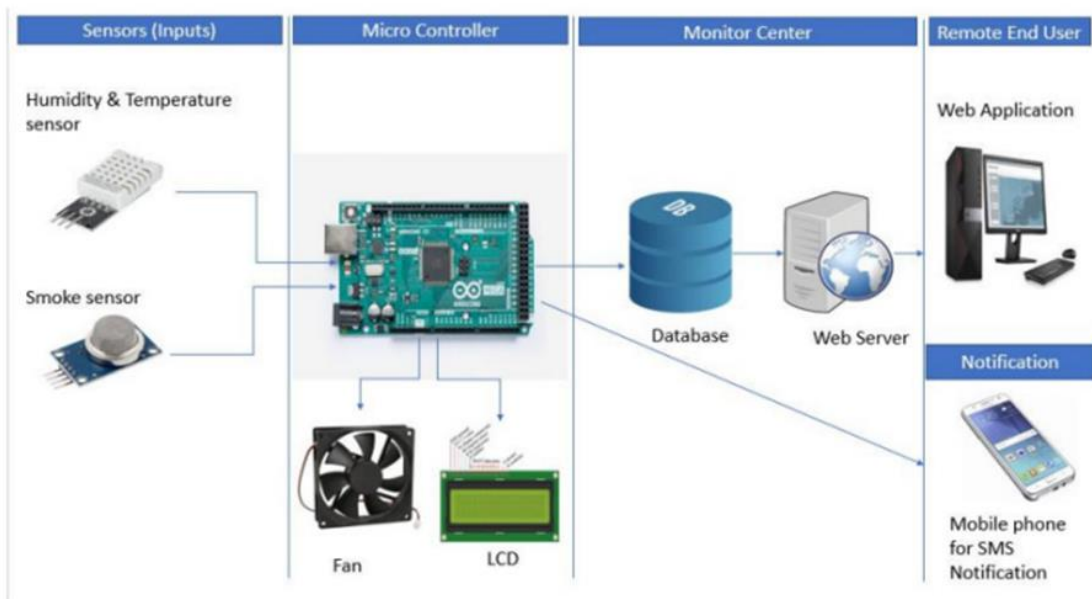


Рис. 1.17. Схема роботи системи

Система AVTECH є одним із провідних рішень для моніторингу серверних приміщень, що відзначається високим рівнем контролю та керування критично важливою ІТ-інфраструктурою [4]. Це рішення орієнтоване на забезпечення максимальної надійності, безпеки та стабільності умов роботи серверного обладнання.

Ключовою особливістю AVTECH є розширений моніторинг середовища, що охоплює контроль таких важливих параметрів, як температура, вологість,

					ІТС.4КІ.0125.03-ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

рівень шуму, а також виявлення наявності води або диму в приміщенні. Це дозволяє ефективно попереджати аварійні ситуації, такі як перегрів серверів чи виникнення загорянь, які можуть призвести до порушення роботи системи.

Крім моніторингу мікроклімату, AVTECH також інтегрує засоби фізичної безпеки, включаючи датчики руху та контролери доступу до серверної кімнати. Це забезпечує додатковий захист як обладнання, так і конфіденційних даних від несанкціонованого втручання.

Окремо варто виділити систему сповіщень і тривоги, яка налаштовується відповідно до потреб користувача. Оповіщення можуть надсилатися електронною поштою, SMS або через мобільні застосунки, що дозволяє адміністраторам своєчасно реагувати на критичні зміни в роботі серверного приміщення.

Ще однією перевагою AVTECH є можливість інтеграції з іншими системами управління дата-центрами, що забезпечує централізований моніторинг і керування всією IT-інфраструктурою. Це рішення легко адаптується до різних IT-систем і дозволяє ефективно взаємодіяти з наявним обладнанням.

Завдяки своїм розширеним можливостям AVTECH є ідеальним вибором для підприємств, які потребують надійного та всеосяжного контролю умов у серверних кімнатах. Від спостереження за мікрокліматом до інтеграції з охоронними системами – це комплексне рішення забезпечує стабільну роботу критичних IT-ресурсів. Проте, через свою високу вартість, AVTECH є малопридатним для впровадження в освітніх установах або малому бізнесі.

Проведений аналіз показав, що ринок рішень для моніторингу мікроклімату серверних кімнат як в Україні, так і за кордоном представлений досить широко. Існує значне розмаїття систем, які відрізняються функціональними можливостями: від базового контролю температури й вологості до комплексних рішень, що одночасно виконують функції систем безпеки (моніторинг дверей, вікон) та контролюють роботу серверного обладнання.

Більшість таких систем використовують бездротову передачу даних між сенсорами та центральними серверами, інтегруючись із хмарними платформами

					ІТС.4КІ.0125.03-ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

для забезпечення доступу з будь-якої точки. Водночас загальними недоліками цих рішень є їх прив'язаність до фірмових сенсорів конкретного виробника, що ускладнює адаптацію з іншими компонентами. Більшість систем орієнтовані на великі дата-центри або великі серверні кімнати, а їхня вартість, разом із регулярними платежами за використання фірмового програмного забезпечення, часто перевищує бюджет малого бізнесу чи освітніх закладів.

Крім того, складність налаштування та велика кількість параметрів, які потребують контролю, роблять процес впровадження таких систем досить ресурсоємним.

Найкращим варіантом для навчальних закладів і малого бізнесу є створення власного рішення для моніторингу мікроклімату, яке дозволяє задовольнити базові потреби в контролі ключових параметрів, зберігаючи при цьому доступність та простоту в обслуговуванні.

### **1.3. Стандарти та рекомендації з моніторингу умов у серверних приміщеннях**

У даному розділі представлено рекомендації щодо визначення кількості та оптимального розташування датчиків температури та вологості в серверних приміщеннях, розроблені на основі вимог стандарту ANSI/ASHRAE/IES 90.1-2022 «Стандарт енергоефективності для будівель, за винятком малоповерхових житлових об'єктів» [25].

Відповідно до відомостей, розміщених на ресурсі [26], обов'язковому контролю підлягають наступні параметри:

- температура та вологість у межах серверного приміщення;
- температура в безпосередній близькості від систем опалення, вентиляції та кондиціонування (HVAC);
- температура повітря на вході та виході серверної стійки.

Рекомендації стосовно оптимального розміщення та типів сенсорних пристроїв наведено у Таблиці 1.2 [26].

Відповідно до стандартів ASHRAE [25] та даних, представлених у Таблиці 1.2, для ефективного моніторингу мікрокліматичних параметрів на рівні серверної стійки необхідно встановити не менше шести температурних сенсорів.

					<b>ITC.4KI.0125.03-ПЗ</b>	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		

Датчики повинні розташовуватися по обидва боки стійки – на передній та задній частинах, а також на трьох рівнях: у верхній, середній та нижній зонах.

Таблиця 1.2

**Рекомендації з оптимального розміщення датчиків температури та вологості в серверному приміщенні**

<b>Область застосування</b>	<b>Місце встановлення сенсорів</b>	<b>Нормативні значення параметрів</b>	<b>Рекомендовані типи датчиків</b>
Вологість у серверній кімнаті	У зонах з підвищеним ризиком зміни вологості	40% – 60%	Датчик вологості
Температура в серверній кімнаті	У місцях з підвищеною температурою	18–27 °C	Датчик температури
Обігрів, вентиляція, кондиціонування (HVAC)	Поруч із кожним пристроєм HVAC для контролю їхнього стану	Температура: 18–27 °C; вологість: 40% – 60%	Датчик температури та вологості
Температура на вході в серверну стійку	<b>ASHRAE</b> рекомендує по 3 сенсори на передній частині кожної стійки (зверху, посередині, внизу)	18–27 °C	Датчик температури
Температура на виході зі стійки	<b>ASHRAE</b> рекомендує по 3 сенсори на задній частині кожної стійки (зверху, посередині, внизу)	Менше ніж 20 °C	Датчик температури

Моніторинг мікрокліматичних параметрів у серверних приміщеннях здійснюється шляхом встановлення датчиків температури та вологості.

Розміщення цих сенсорів передбачає:

- встановлення у зонах із потенційним перегрівом усередині серверної кімнати;
- розташування поблизу кондиціонерів для своєчасного виявлення можливих несправностей систем охолодження.

Для невеликих серверних кімнат допускається використання як дротових, так і бездротових датчиків. Провідні сенсори дозволяють знизити загальну вартість системи, водночас бездротові рішення забезпечують більшу гнучкість і простоту встановлення.

В результаті здійсненого аналізу (розділи 1.2 та 1.3) були сформульовані фундаментальні вимоги до системи спостереження за мікрокліматом у серверному приміщенні:

1. Система повинна забезпечувати підтримання оптимальних рівнів температури та вологості для гарантованої працездатності серверного обладнання.
2. Система має працювати в цілодобовому режимі протягом усього року без перерв.
3. У випадку використання повітряної системи охолодження, вимірювання температурних і вологісних показників необхідно здійснювати на висоті 1,5 метра від поверхні підлоги.
4. Повинна бути передбачена можливість гнучкого налаштування порогових значень температури та вологості в процесі експлуатації.
5. Система має забезпечувати передачу даних на віддалені пристрої через Wi-Fi.
6. Повинна бути можливість подальшого розширення функціоналу системи (модульність).
7. Максимально допустиме відхилення при вимірюванні температури –  $\pm 0,5$  °C.
8. Вимірювальний пристрій повинен охоплювати температурний спектр від 0 до 60 °C.
9. Точність вимірювання вологості повинна знаходитися в межах 2–5%.
10. Система повинна забезпечувати вимірювання вологості в діапазоні 0–100%
11. Інтервал між вимірюваннями температури та вологості має становити від 30 секунд до 1 хвилини.

					<i>ITC.4KI.0125.03-ПЗ</i>	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дата		

Ці вимоги забезпечують належний рівень контролю параметрів мікроклімату для підтримання стабільної роботи серверного обладнання та своєчасного реагування на зміни умов у серверному приміщенні.

### **Висновки до розділу**

Проведений огляд літературних джерел та нормативних документів підтвердив актуальність розробки автоматизованої системи моніторингу мікроклімату серверних приміщень, особливо з урахуванням потреб освітніх установ та малого бізнесу, які мають обмежені бюджети й не потребують надлишкових функцій, притаманних промисловим системам. Враховуючи специфіку таких об'єктів, розробка подібної системи має базуватися на модульній архітектурі, яка дозволяє адаптувати функціонал під конкретні вимоги користувача та забезпечує можливість масштабування.

В межах першого розділу було ретельно досліджено законодавчу та нормативну базу, що регулює параметри внутрішнього клімату в серверних кімнатах. Розглянуто основні стандарти, зокрема ASHRAE, які визначають перелік параметрів, що підлягають обов'язковому контролю (температура, вологість), та їхні оптимальні значення для підтримки стабільної працездатності серверного обладнання.

Окрім цього, було проведено зіставний аналіз існуючих рішень для спостереження за мікрокліматом у серверних кімнатах. Розглянуто їх функціональні можливості, переваги та недоліки. Встановлено, що більшість комерційних систем орієнтовані на великі дата-центри або серверні приміщення значних розмірів. Такі системи характеризуються високою вартістю, складністю налаштування, обмеженою сумісністю з нестандартними сенсорами, а також необхідністю регулярної оплати за використання фірмового програмного забезпечення.

В рамках аналізу також розглянуто стандарти щодо кількості, типів і розташування сенсорів залежно від контрольованих зон (загальний простір серверної кімнати, серверні стійки, системи HVAC). Це дозволило сформулювати чітке уявлення про основні технічні вимоги до системи моніторингу.

					<i>ITC.4KI.0125.03-ПЗ</i>	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

У підсумку, було сформульовано технічні вимоги до автоматизованої системи контролю мікроклімату серверного приміщення, які враховують не лише нормативні обмеження, але й потреби в гнучкості, доступності та простоті використання для малих організацій. Така система має забезпечувати базові функції моніторингу температури та вологості, підтримувати можливість віддаленого доступу через Wi-Fi, гарантувати точність вимірювань та дозволяти оперативне налаштування порогових значень.

Таким чином, розробка доступної, енергоефективної та адаптивної системи моніторингу мікроклімату серверної кімнати є перспективним напрямом, що дозволяє забезпечити оптимальні умови експлуатації ІТ-обладнання навіть у невеликих серверних приміщеннях.

					<i>ІТС.4КІ.0125.03-ПЗ</i>	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

## РОЗДІЛ 2. ВИБІР КОМПОНЕНТІВ ДЛЯ СИСТЕМИ МОНІТОРИНГУ МІКРОКЛІМАТУ СЕРВЕРНОЇ КІМНАТИ НА ОСНОВІ МІКРОКОНТРОЛЕРА ARDUINO

### 2.1. Огляд основних компонентів системи

#### 2.1.1. Огляд Arduino Mega

Для реалізації проєкту комп'ютерної системи моніторингу мікроклімату серверної кімнати в рамках даної роботи було обрано мікроконтролерну плату Arduino Mega. Це рішення обумовлене низкою факторів, включаючи доступність, гнучкість та широку підтримку спільноти розробників.

Arduino Mega 2560 є розширеною версією популярної платформи Arduino, побудованою на базі потужного мікроконтролера ATmega2560 [10]. Ця плата пропонує значно більше ресурсів порівняно зі своїми попередниками, включаючи більшу кількість цифрових та аналогових входів/виходів, а також кілька послідовних портів UART. Така архітектура дозволяє реалізувати складніші проєкти, що вимагають великої кількості периферійних пристроїв.

Однією з ключових переваг Arduino Mega 2560 є її відкритість. Плата має відкритий апаратний і програмний код, що дозволяє глибоко вивчити її роботу та модифікувати під власні потреби. Крім того, існує велика спільнота розробників, яка створює різноманітні бібліотеки, щити та інші ресурси, що спрощують процес розробки.

Простота використання Arduino Mega 2560 також є важливим фактором. Інтуїтивно зрозуміле середовище розробки Arduino IDE дозволяє швидко створювати та завантажувати програми на плату, навіть для користувачів без глибоких знань в області програмування.

Таким чином, вибір Arduino Mega в якості основної платформи для проєкту комп'ютерної системи моніторингу мікроклімату серверної кімнати є обґрунтованим з точки зору її технічних характеристик, доступності, гнучкості та широкої підтримки спільноти розробників.

					ІТС.4КІ.0125.03-ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	РОЗДІЛ 2. ВИБІР КОМПОНЕНТІВ ДЛЯ СИСТЕМИ МОНІТОРИНГУ МІКРОКЛІМАТУ СЕРВЕРНОЇ КІМНАТИ НА ОСНОВІ МІКРОКОНТРОЛЕРА ARDUINO	Літ.	Арк.	Акрушів
Розроб.		Ліщук С.Е.					36	37
Керівник		Донченко В.Ю.				ЛНУ Кафедра ІТС, Гр.4КІ		
Реценз.		Козуб Ю.Г.						
Н. Контр.								
Зав. каф.		Семенов М.А.						

Плата оснащена всім необхідним для роботи: кварцовим резонатором, роз'ємами живлення та програмування, а також кнопкою скидання. Для живлення можна використовувати як зовнішній блок живлення, так і підключення до комп'ютера через USB-кабель. Arduino Mega 2560 сумісна з більшістю щитів, розроблених для попередніх версій платформи, що дозволяє розширювати її функціональність за допомогою додаткових модулів.

Використання Arduino Mega дозволить реалізувати широкий спектр функцій в системі моніторингу мікроклімату серверної кімнати, включаючи управління освітленням, контролювання температури, забезпечення безпеки та багато іншого (рис. 2.1).

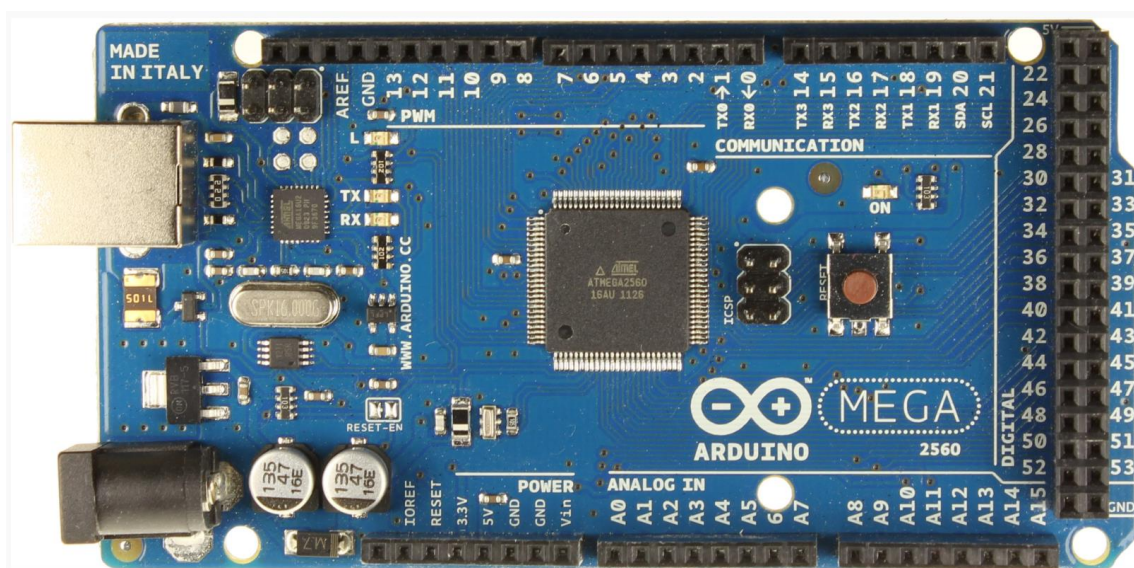


Рис. 2.1. Мікроконтролер Arduino Mega 2560

Плата Arduino Mega 2560 є однією з найпотужніших платформ в сімействі Arduino, що відрізняється значно розширеною функціональністю порівняно зі своїми попередниками. Однією з ключових особливостей Mega 2560 є використання мікроконтролера ATmega16U2 (або ATmega8U2 в ранніх версіях) для реалізації інтерфейсу USB-UART замість традиційних мікросхем FTDI [10]. Таке рішення спрощує процес оновлення прошивки та забезпечує більшу інтеграцію з платформою Arduino.

Опис пінів плати на базі ATmega16U2 показаний на рисунку 2.2.

# ARDUINO MEGA PINOUT DIAGRAM

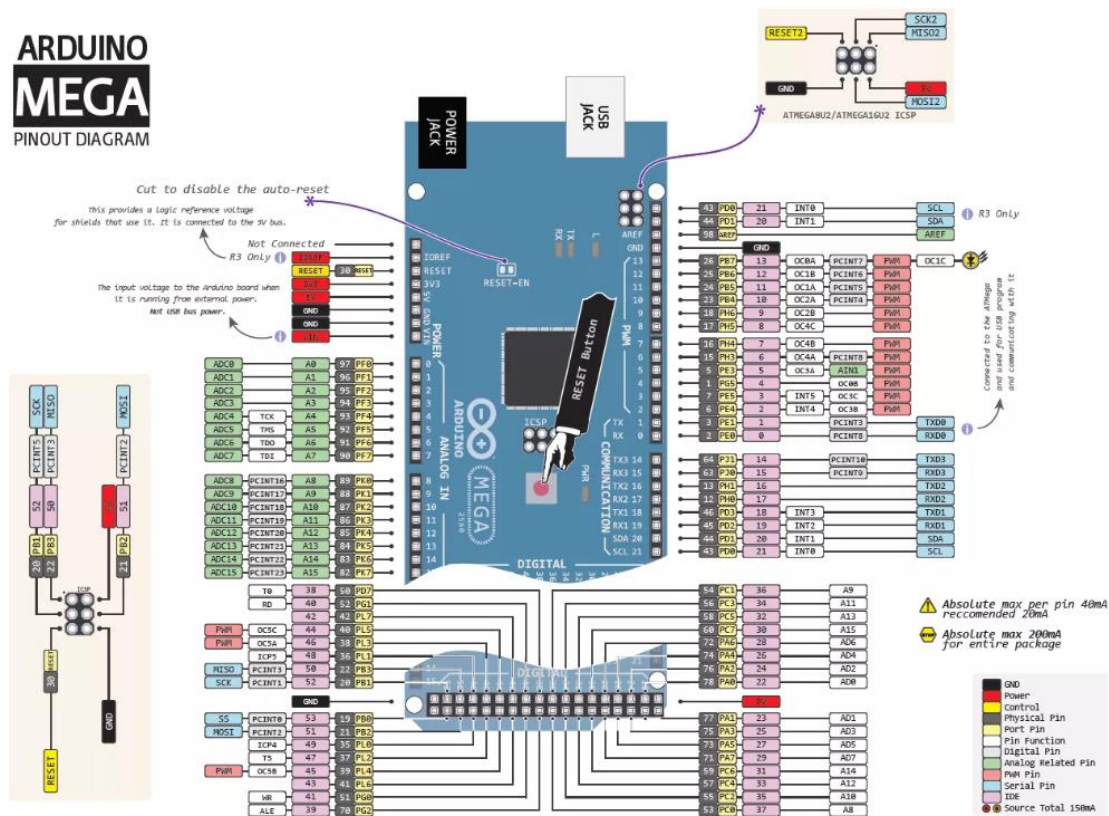


Рис. 2.2. Опис пінів плати Arduino Mega

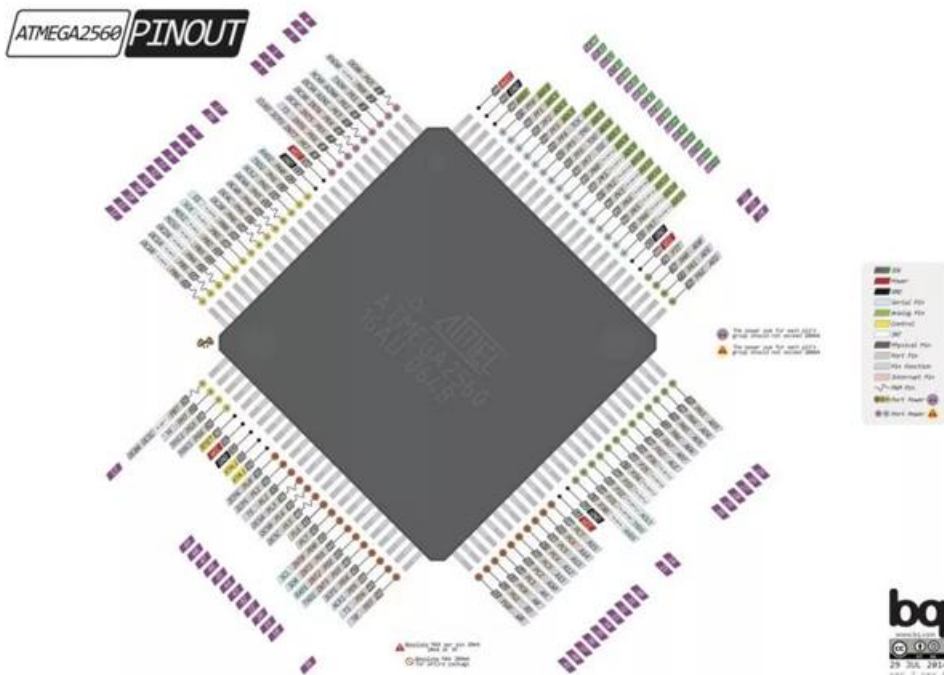


Рис. 2.3. Распиновка чипа Atmega2560

Починаючи з версії R2, на платі Mega 2560 був доданий резистор підтяжки до землі лінії HWB мікроконтролера 8U2, що додатково спростило процедуру переходу пристрою в режим DFU. Версія R3 принесла ще більше удосконалень: з'явилися виводи SDA і SCL для підключення зовнішніх пристроїв, а також новий

вивід IOREF, який дозволяє платам розширення адаптуватися до різної напруги живлення (5 В або 3.3 В).

Arduino Mega 2560 сумісна з більшістю щитів, розроблених для попередніх версій платформи Arduino (Uno, Diecimila, Duemilanove) завдяки однаковому розташуванню основних компонентів, таких як цифрові та аналогові входи/виходи, роз'єми живлення та ICSP. Це дозволяє легко розширювати функціональність плати за допомогою додаткових модулів.

Технічні характеристики Arduino Mega 2560:

- Мікроконтролер: ATmega2560
- Напруга живлення: 5 В (номінальна), діапазон робочих напруг 7-20 В
- Цифрові входи/виходи: 54 (з них 15 з функцією ШІМ)
- Аналогові входи: 16
- Пам'ять: флеш-пам'ять 256 КБ, SRAM 8 КБ, EEPROM 4 КБ
- Тактова частота: 16 МГц

Зовнішні джерела живлення можуть бути представлені мережевими адаптерами змінного струму або акумуляторами. Для підключення зовнішнього джерела живлення використовується спеціальний роз'єм на платі. Рекомендується використовувати адаптери з напругою в діапазоні від 7 до 20 В. Вихідна напруга адаптера 5 В подається на відповідний контакт на платі, забезпечуючи живлення всіх компонентів.

Важливо зазначити, що використання напруги живлення нижче 7 В може призвести до зниження вихідної напруги 5 В і, як наслідок, до нестабільної роботи плати. З іншого боку, надмірна напруга (понад 12 В) може призвести до перегріву стабілізатора напруги та виходу плати з ладу. Тому, для забезпечення оптимальної роботи Arduino Uno, слід дотримуватися рекомендованих параметрів живлення.

Підключення зовнішнього джерела живлення здійснюється шляхом під'єднання позитивного полюса до контакту VIN, а негативного - до контакту GND на платі. При використанні акумулятора також необхідно під'єднати його до цих контактів.

					<i>ІТС.4КІ.0125.03-ПЗ</i>	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

Вибір джерела живлення залежить від конкретних умов експлуатації плати Arduino Mega. Для стаціонарних застосувань можна використовувати мережевий адаптер, а для портативних пристроїв - акумулятори або батареї.

Цифрові виводи Arduino Mega, які можуть працювати як входи, так і виходи, забезпечують гнучкість при розробці електронних проєктів. Кожен вивід здатний працювати з напругою до 5 В і витримувати струм до 40 мА. Для зручності роботи з датчиками передбачена можливість підключення внутрішніх підтягуючих резисторів. Функції `pinMode()`, `digitalWrite()` та `digitalRead()` дозволяють легко керувати станом цих виводів (рис. 2.4).

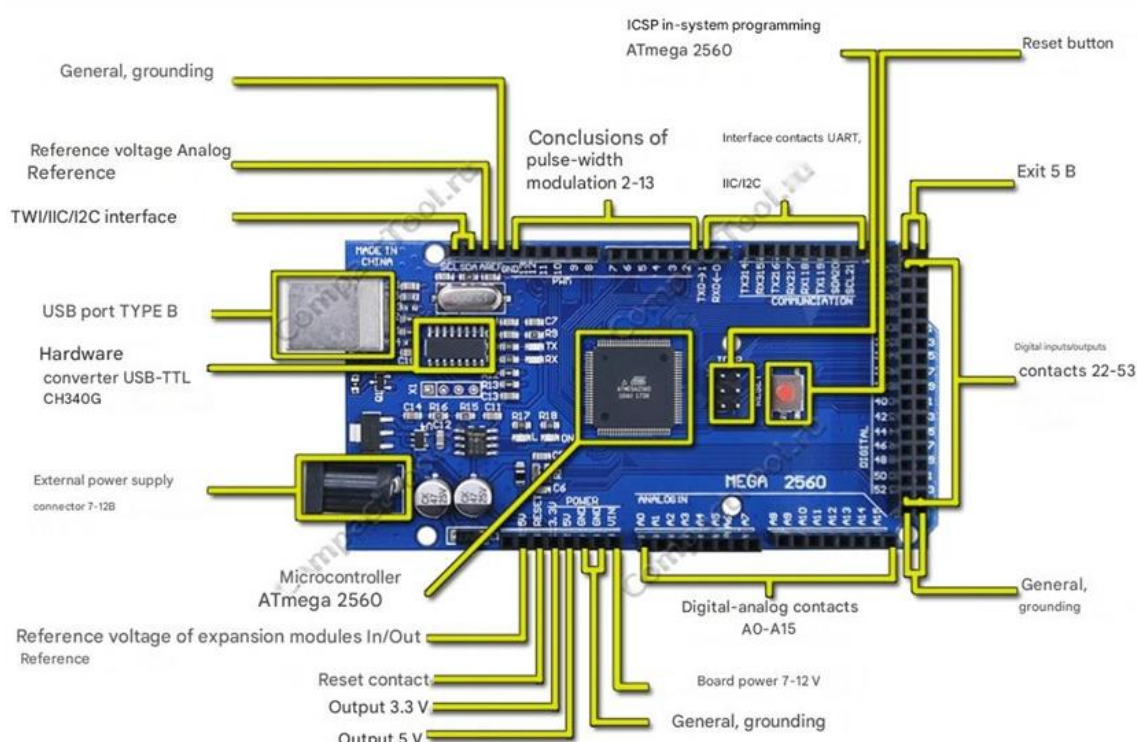


Рис. 2.4. Виводи Arduino Mega

Для захисту мікроконтролера та підключених до нього пристроїв від перевантажень, плата Arduino Uno оснащена запобіжником. Запобіжник розрахований на максимальний струм 500 міліампер. У випадку перевищення цього значення, запобіжник перегорає, розриваючи електричний ланцюг та запобігаючи пошкодженню плати. Цей додатковий рівень захисту є важливим, оскільки навіть сучасні комп'ютери не завжди можуть забезпечити повний захист від перевантажень за лінією USB.

Принципова схема плати показана на рисунку 2.5.

					ІТС.4КІ.0125.03-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		40

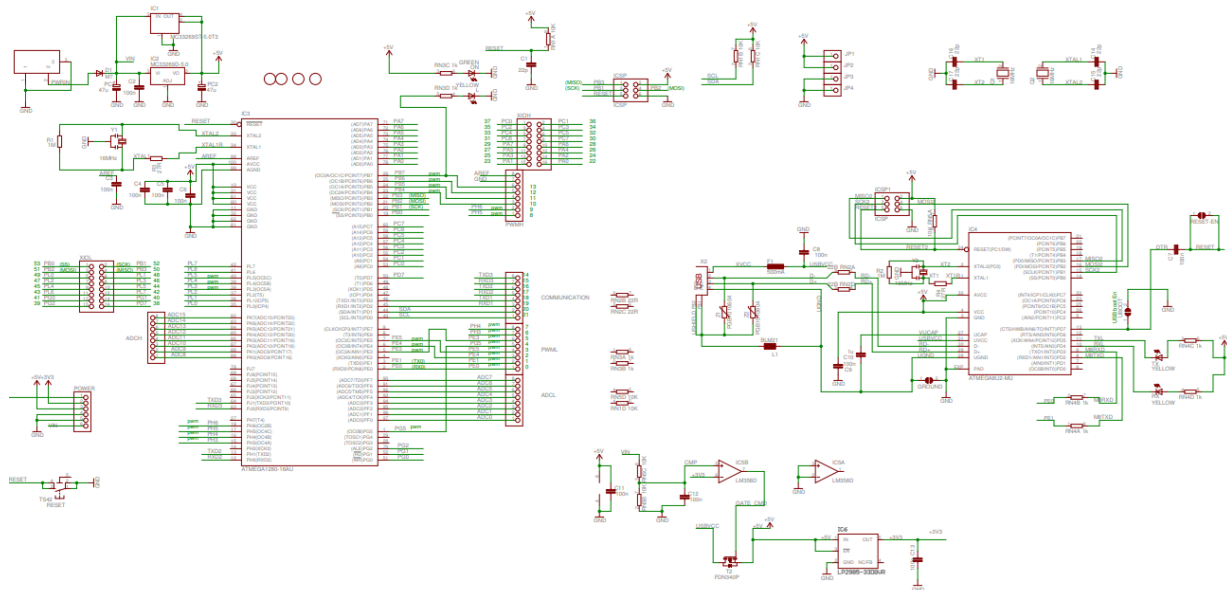


Рис. 2.5. Принципова схема плати Arduino Mega 2560

Arduino Mega 2560 є потужною і універсальною платформою для розробки електронних пристроїв. Вона поєднує в собі високу продуктивність, розширені можливості та сумісність з великою кількістю додаткових модулів. Завдяки своїм характеристикам, Arduino Mega 2560 є відмінним вибором для реалізації складних проєктів у сферах робототехніки, автоматизації та Інтернету речей.

### 2.1.2. Опис PIR-датчика

Піроелектричний інфрачервоний датчик (PIR-датчик) є високочутливим електронним пристроєм, призначеним для виявлення інфрачервоного випромінювання, яке випускають усі теплі тіла, включаючи людей і тварин. Принцип роботи PIR-датчика заснований на використанні піроелектричного матеріалу, який генерує електричний заряд при зміні температури [13].

Конструктивно PIR-датчик складається з лінзи Френеля, яка фокусує інфрачервоне випромінювання на піроелектричний елемент. Коли в зоні дії датчика з'являється рухомий об'єкт, що випромінює тепло, його інфрачервоне випромінювання змінює температуру піроелектричного елемента, що призводить до виникнення електричного сигналу. Цей сигнал обробляється електронною схемою датчика і може використовуватися для подальшої передачі або активації інших пристроїв (рис. 2.6).



Рис. 2.6. PIR-датчик

Піроелектричні інфрачервоні датчики (PIR-датчики) є ефективним засобом для виявлення руху в навколишньому середовищі. Принцип їх роботи заснований на реєстрації змін теплового випромінювання, яке виникає при русі об'єктів.

Однією з основних функцій PIR-датчиків є виявлення руху. Коли в зоні дії датчика з'являється об'єкт, що випромінює тепло (наприклад, людина або тварина), датчик реєструє зміну теплового потоку і генерує електричний сигнал. Цей сигнал може використовуватися для активації різних пристроїв та систем.

Застосування PIR-датчиків є досить широким і охоплює різні сфери:

- Системи безпеки. PIR-датчики використовуються для створення охоронних систем, які спрацьовують при виявленні несанкціонованого проникнення.
- Автоматизація освітлення. Датчики дозволяють автоматично вмикати та вимикати освітлення залежно від присутності людей, що сприяє економії енергії.
- Системи контролю доступу. PIR-датчики можуть використовуватися для відкривання дверей або шлагбаумів при виявленні авторизованих осіб.
- Робототехніка. Датчики руху використовуються для створення роботів, які можуть орієнтуватися в просторі та уникати перешкод.

Ключові характеристики PIR-датчиків:

1. Чутливість: здатність датчика виявляти навіть незначні зміни теплового випромінювання.

					ІТС.4КІ.0125.03-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		42

2. Дальність дії: відстань, на якій датчик може виявити рух.
3. Кут огляду: розмір зони, яку датчик контролює.
4. Час реакції: швидкість, з якою датчик реагує на зміну теплового потоку.

Більшість сучасних PIR-датчиків дозволяють користувачеві налаштовувати такі параметри, як чутливість, час затримки реакції та кут огляду. Це дозволяє оптимізувати роботу датчика для конкретних умов застосування.

Налаштування модуля HC-SR501 є доволі простим та зводиться до врахування трьох основних параметрів.

По-перше, необхідно встановити час затримки після спрацювання датчика. Упродовж цього періоду буде виконуватись певна подія, наприклад, вмикання освітлення або сигналізації. Час затримки можна регулювати в діапазоні від 5 до 300 секунд (у деяких джерелах вказується до 200 секунд, однак на практиці модулі часто підтримують до 300 секунд).

По-друге, налаштовується чутливість датчика, тобто відстань, з якої він здатен виявити рух. Цей параметр регулюється в межах від 3 до 7 метрів. Наприклад, для коридору шириною 2 метри оптимальним буде середній рівень чутливості.

По-третє, необхідно визначити положення перемички режиму роботи (джампера). Згідно з технічною документацією, передбачено два режими: L (Low) і H (High).

У режимі L: коли датчик фіксує рух, він активує навантаження (наприклад, вмикає лампу), після чого не реагує на подальший рух доти, доки не завершиться встановлений час затримки. Після завершення затримки модуль переходить у режим блокування тривалістю приблизно 2,5 секунди, під час якого рух також ігнорується. По завершенні блокування датчик знову готовий до фіксації руху — щойно рух буде виявлено, сценарій активується повторно.

У режимі H: після фіксації руху активується таймер затримки, як і в режимі L. Однак у цьому режимі, якщо протягом цього періоду виявляється новий рух, таймер скидається й запускається заново. Таким чином, поки поблизу датчика

					ІТС.4КІ.0125.03-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43

відбувається рух, навантаження залишається активним. Щойно рух припиняється й час затримки вичерпується, навантаження вимикається. Після цього датчик також входить у короткочасний режим блокування на 2,5 секунди, а потім знову переходить у стан очікування.

На рисунку 2.7 показано розташування елементів регулювання та перемички режиму роботи.

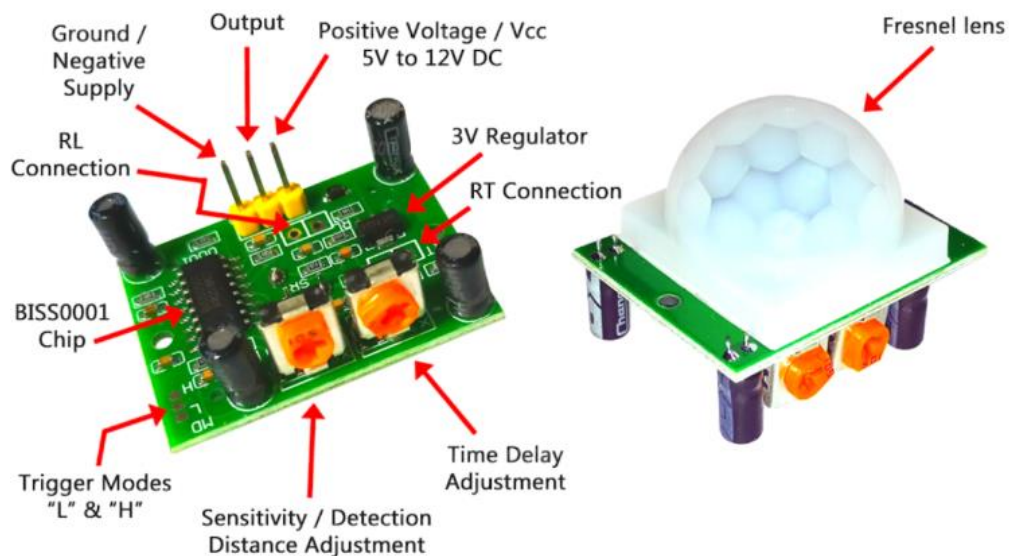


Рис. 2.7. Розташування елементів регулювання та перемички режиму роботи

Усі регулювання, здійснюються за годинниковою стрілкою, призводять до збільшення чутливості або часу затримки. Відповідно, обертання проти годинникової стрілки — зменшує ці параметри.

Додаткова важлива інформація під час встановлення: сам чутливий елемент розташований безпосередньо під білим куполом, який є лінзою Френеля. Ця лінза має складну багатогранну структуру, що забезпечує коректне заломлення та фокусування інфрачервоних променів на сенсор, підвищуючи ефективність виявлення руху.



Рис. 2.8. Лінза Френеля

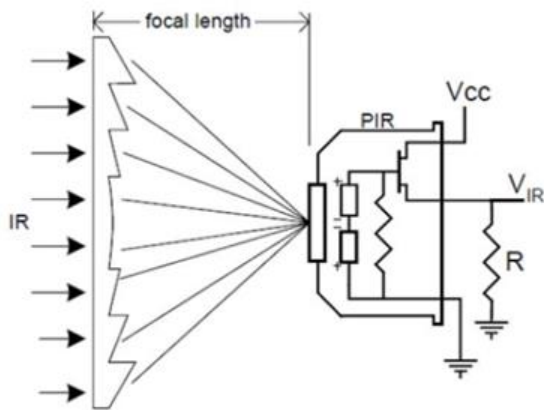


Рис. 2.9. Проходження променів крізь лінзу

Слід зазначити, що датчик може функціонувати й без лінзи (перевірено експериментально). Втім, у такому випадку його чутливість суттєво знижується. Сам сенсор має прямокутне подвійне вікно.

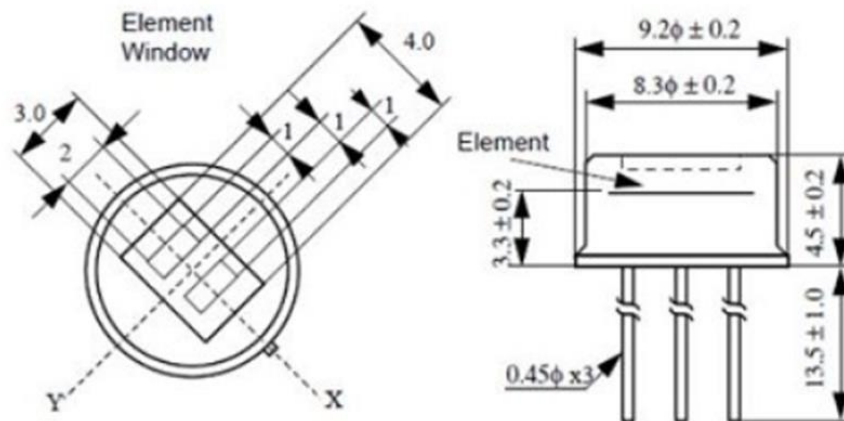


Рис. 2.10. Приклад роботи сенсорів

Датчик реагує на різницю сигналів, що надходять через два вікна сенсора. Чим більшою є ця різниця, тим вища чутливість пристрою. Саме тому вікна датчика слід орієнтувати паралельно до ймовірного напрямку руху об'єктів.

Ще один важливий аспект: до перешкод, що можуть спричинити хибні спрацювання, належать сонячне світло та вітер. Тому при встановленні модуля рекомендується уникати прямого впливу сонячних променів, а також бокового потоку повітря, наскільки це можливо.

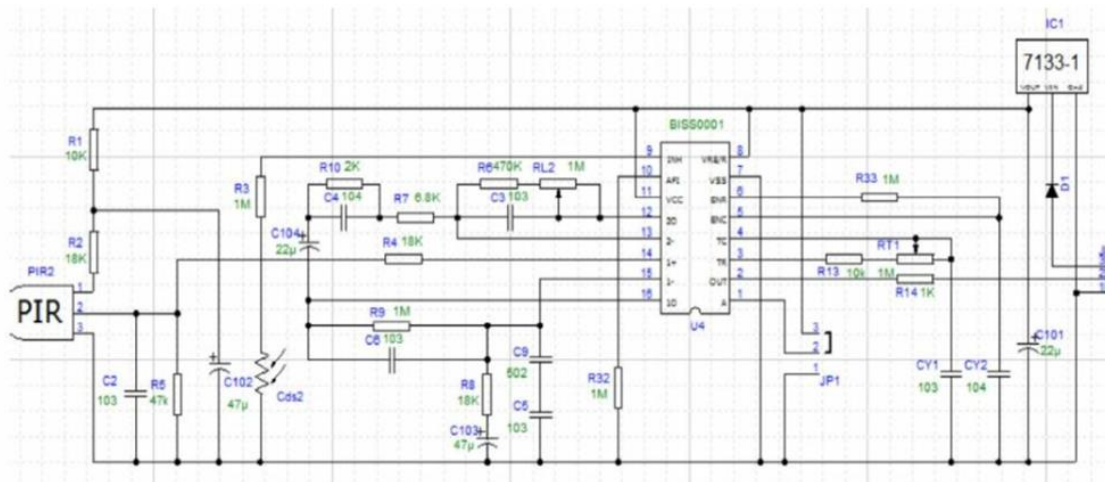


Рис. 2.11. Схема модуля

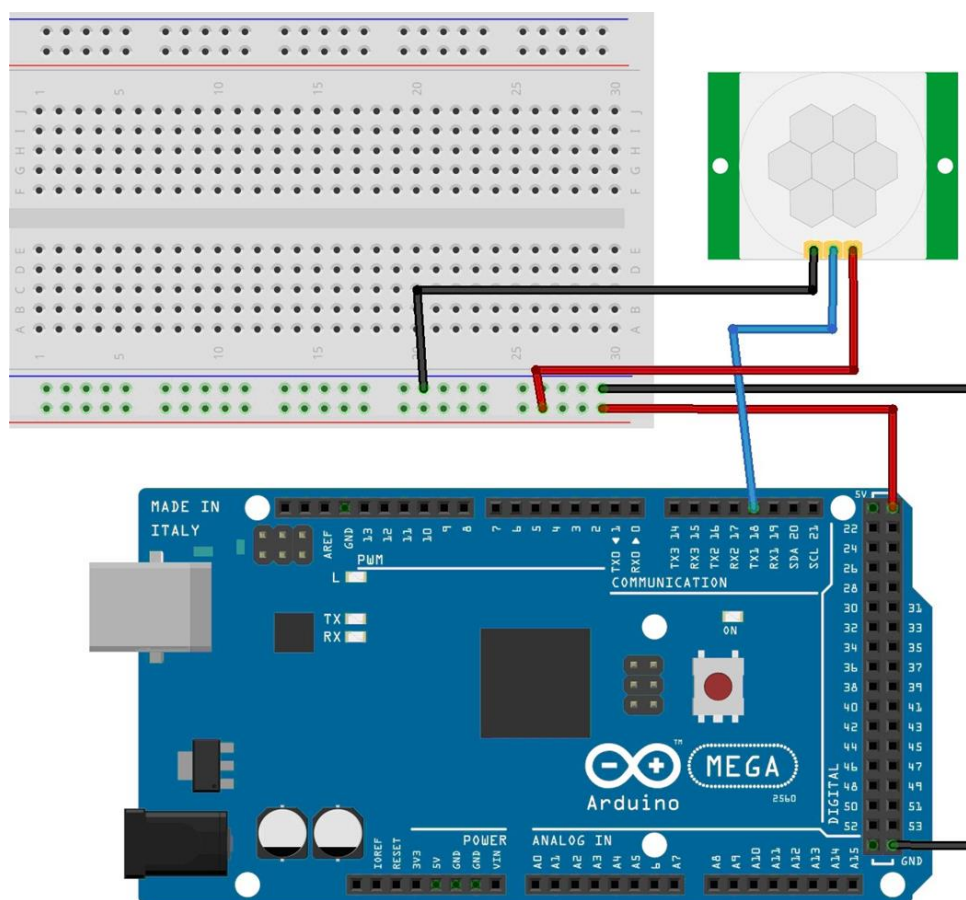


Рис. 2.12. Схема підключення датчика руху HC-SR501

### 2.1.3. Опис DHT11

У процесі розробки прототипу пристрою для контролю мікроклімату в серверних кімнатах перевагу було надано датчику DHT11 завдяки його поєднанню доступності та ефективності. Цей датчик, який є популярним рішенням для вимірювання температури та вологості, використовує ємнісний принцип для визначення температури та гігрометр для вологості.

Вбудований аналогово-цифровий перетворювач (АЦП) забезпечує перетворення аналогових вимірювань у цифровий формат, що полегшує інтеграцію в загальну схему пристрою. Цифровий інтерфейс датчика базується на однопровідній шині з відкритим колектором, що передбачає обов'язкове використання підтягуючого резистора (10 кОм). Зовнішній вигляд DHT11 ілюстровано на рисунку 2.13. Вибір саме цього датчика пояснюється його здатністю точно фіксувати ключові параметри мікроклімату, а також його економічністю, що робить його ідеальним вибором для створення системи моніторингу.

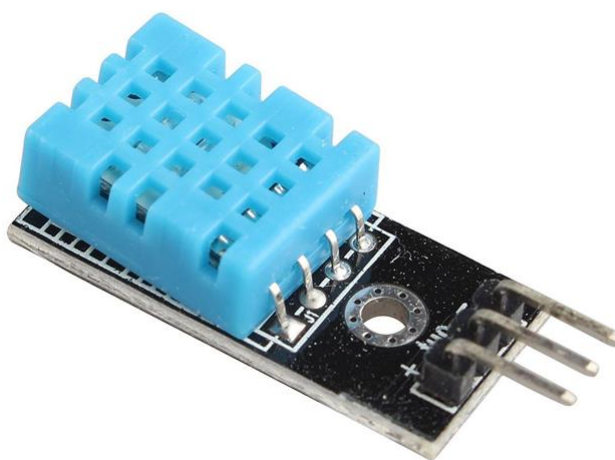


Рис. 2.13. DHT11 датчик

DHT11 складається з двох основних елементів: терморезистора для вимірювання температури та ємнісного датчика для вимірювання вологості. Принцип роботи датчика заснований на зміні електричних характеристик цих елементів залежно від температури та вологості навколишнього середовища. Отримані дані перетворюються в цифровий сигнал, який передається на мікроконтролер.

Основні характеристики DHT11:

1. Діапазон вимірювань. Датчик здатний вимірювати температуру в діапазоні від 0 до 50 градусів Цельсія з точністю  $\pm 2$  градуси, а відносну вологість - від 20% до 80% з точністю  $\pm 5\%$ .
2. Цифровий інтерфейс. Для взаємодії з мікроконтролерами DHT11 використовує простий цифровий інтерфейс, що дозволяє легко інтегрувати його в різні електронні системи.

					<b>ІТС.4КІ.0125.03-ПЗ</b>	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

3. Низьке енергоспоживання. Датчик має низький рівень споживання енергії, що робить його придатним для використання в автономних пристроях, що живляться від батарей.
4. Компактні розміри. Малі габарити дозволяють легко розміщувати датчик у різних пристроях та системах [14].

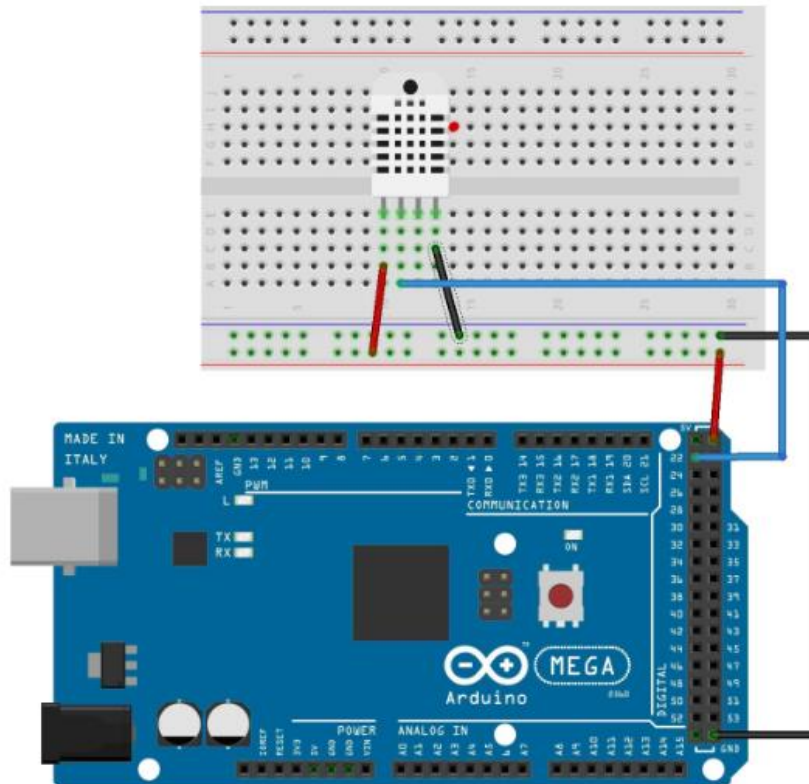


Рис. 2.10. Схема підключення датчика DHT11

#### 2.1.4. Дисплей LCD 1602

Рідкокристалічні дисплеї LCD 1602 є поширеним вибором для виведення текстової інформації у багатьох електронних проєктах завдяки своїй доступності та широкому спектру модифікацій [27]. Однак, їхнє застосування з мікроконтролерами типу Arduino Uno або Nano має певні обмеження. Класична схема підключення LCD 1602 вимагає використання мінімум 6 цифрових виводів мікроконтролера, що може бути значним обмеженням для проєктів з обмеженою кількістю доступних портів. Це пов'язано з тим, що для керування кожним сегментом символів на дисплеї потрібен окремий сигнал керування.

Для подолання цього обмеження часто використовують LCD модулі з інтерфейсом I2C. Цей інтерфейс дозволяє підключити дисплей до мікроконтролера лише за допомогою двох проводів (SCL і SDA), суттєво

зменшуючи кількість зайнятих портів. Це робить LCD дисплеї з інтерфейсом I2C більш універсальними та зручними у використанні з платформами Arduino.

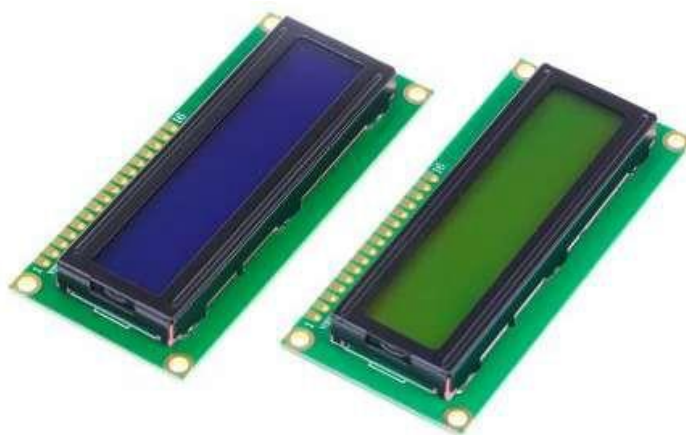


Рис. 2.11. Дисплей LCD 1602

Кожен з виводів рідкокристалічного дисплея відіграє критичну роль у його функціонуванні:

- **GND (земля):** слугує точкою відліку для електричних потенціалів у схемі.
- **5 В (живлення):** забезпечує необхідну напругу для роботи електроніки дисплея.
- **Контраст:** дозволяє регулювати яскравість зображення шляхом зміни напруги на цьому виводі.
- **RS (Register Select):** визначає, чи буде наступна передана інформація інтерпретована як команда (наприклад, встановити курсор) чи як дані для відображення на екрані.
- **RW (Read/Write):** використовувався в старих моделях дисплеїв для читання даних з дисплея. В сучасних моделях зазвичай заземлений.
- **Enable (E):** Сигнал, який інформує дисплей про те, що дані на лініях даних готові для прийому.
- **DB0-DB7 (Data Bits):** група з 8 ліній, за якими передаються біти даних, що визначають, які сегменти рідких кристалів мають бути увімкнені для формування зображення.
- **A (анод) та K (катод):** виводи для підключення зовнішнього джерела живлення для підсвічування дисплея.



Рис. 2.12. Виводи LCD 1602

Рідкокристалічний дисплей (LCD) має наступні технічні характеристики:

- Тип відображення: символів, з можливістю завантаження користувацьких символів.
- Підсвічування: світлодіодне, що забезпечує яскраве та рівномірне підсвічування символів.
- Контролер: HD44780, що відповідає за керування роботою дисплея.
- Живлення: 5 вольт постійного струму.
- Розмір дисплея: 16 символів по ширині та 2 рядки по висоті.
- Робоча температура: від -20°C до +70°C.
- Температура зберігання: від -30°C до +80°C.
- Кут огляду: 180 градусів, що забезпечує гарну видимість з різних кутів.

Стандартна схема приєднання монітора безпосередньо до мікроконтролера Ардуіно без I2C зображена на рисунку 2.13.

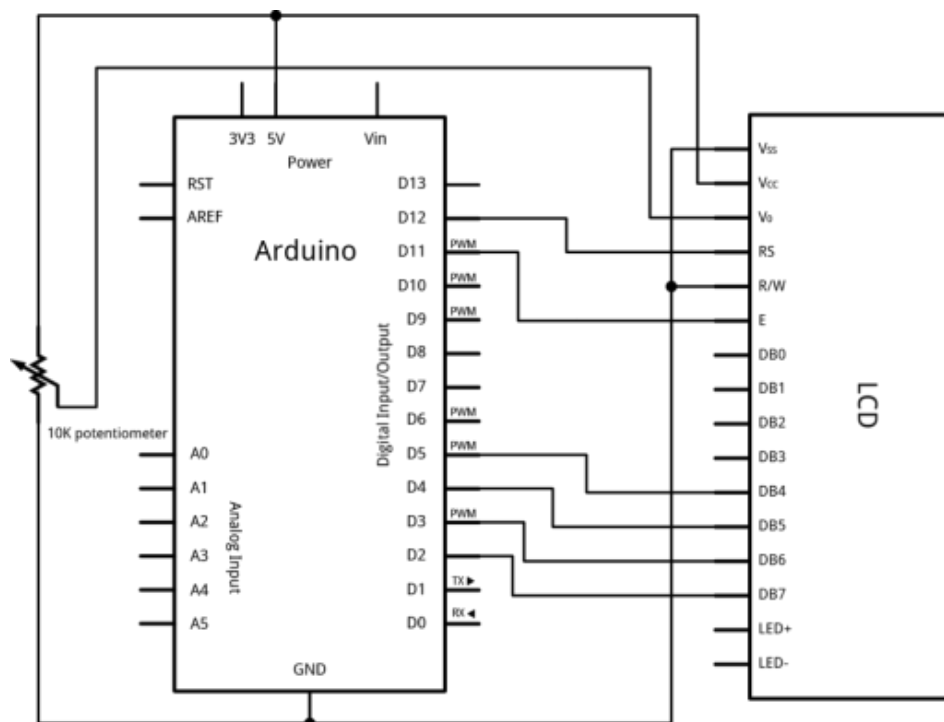


Рис. 2.13. Схема підключення дисплея до Arduino без I2C

I2C (Inter-Integrated Circuit) – це популярний послідовний інтерфейс, який використовується для зв'язку між різними електронними компонентами в системах. Однією з головних переваг I2C є його простота: для підключення багатьох пристроїв достатньо лише двох ліній – лінії даних (SDA) та лінії тактування (SCL). Це дозволяє суттєво економити порти мікроконтролера, що особливо важливо для платформ з обмеженими ресурсами, таких як Arduino. Крім того, I2C підтримує підключення великої кількості пристроїв до однієї шини, що робить його зручним для створення складних електронних систем.

Ще одним важливим плюсом є широка підтримка I2C сучасними мікроконтролерами. Більшість платформ, включаючи Arduino, мають вбудовані модулі для роботи з I2C, що значно спрощує процес розробки.

Однак, як і будь-яка технологія, I2C має свої обмеження. Одним з них є ємнісне обмеження шини. Наявність занадто великої кількості пристроїв або довгих проводів може призвести до спотворення сигналів і, як наслідок, до нестабільної роботи системи. Крім того, програмування мікроконтролера для роботи з великою кількістю різних пристроїв, підключених до I2C, може бути досить складним завданням. Особливо це стосується ситуацій, коли виникають збої в роботі шини, оскільки їх локалізація може зайняти багато часу.

Підсумовуючи, I2C є потужним інструментом для розробки електронних пристроїв, але його використання вимагає розуміння його особливостей та обмежень. Перед вибором I2C для конкретного проєкту необхідно ретельно оцінити такі фактори як кількість підключених пристроїв, довжина шини та складність програмного забезпечення.

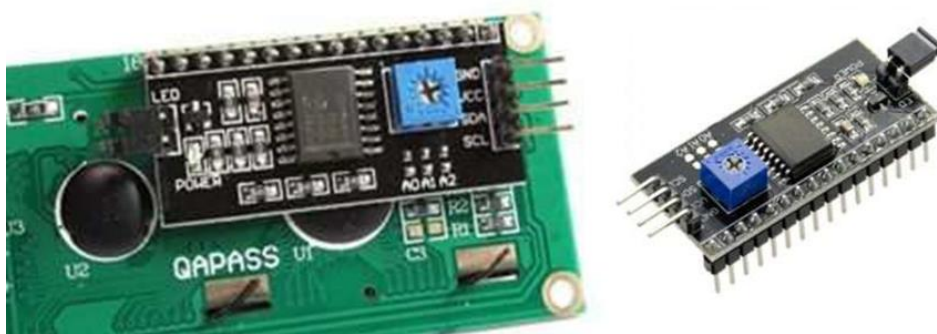


Рис. 2.14. Модуль i2c для LCD 1602

Для швидкого та зручного підключення рідкокристалічного дисплея до плати Arduino рекомендується використовувати готові модулі, які мають вбудовану підтримку протоколу I2C. Це значно спрощує процес підключення, оскільки вимагає лише під'єднання двох проводів (SDA та SCL) до відповідних пінів на Arduino. Для забезпечення підсвічування дисплея може знадобитися додаткове джерело живлення. Контрастність зображення можна легко налаштувати за допомогою вбудованого потенціометра. Схема підключення такого модуля до Arduino детально проілюстрована на рисунку 2.15.

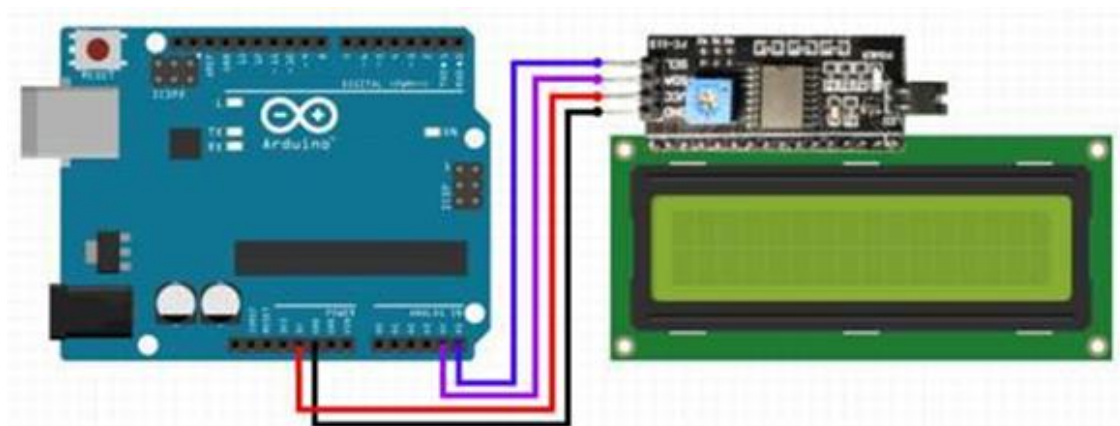


Рис. 2.15. Підключення LCD екрану до Arduino по I2C

Рідкокристалічний дисплей з інтерфейсом I2C підключається до плати за допомогою чотирьох проводів: два для передачі даних (SDA та SCL) і два для

живлення (GND та VCC). Конкретніше, вивід GND дисплея з'єднується з загальним проводом (GND) на платі, вивід VCC підключається до 5-вольтового живлення, а лінії даних SDA та SCL підключаються до відповідних аналогових портів A4 та A5 на платі.

### 2.1.5. Реле для комутації

Реле: це електро cơchanічний або електронний пристрій, який дозволяє керувати потужними електричними ланцюгами за допомогою низьковольтних сигналів. Простіше кажучи, реле – це своєрідний перемикач, який може вмикати і вимикати сильні струми, що протікають через навантаження, наприклад, двигуни, лампи розжарювання чи насоси.

Мікроконтролери, такі як Arduino, здатні обробляти інформацію та приймати рішення, але вони не спроєктовані для безпосереднього керування потужними навантаженнями. Підключення таких навантажень безпосередньо до виходів мікроконтролера може призвести до його пошкодження. Реле ж дозволяє ізолювати мікроконтролер від потужного навантаження, забезпечуючи надійну та безпечну роботу системи.

Основний принцип роботи електромагнітного реле полягає в наступному: коли через котушку реле протікає електричний струм, створюється магнітне поле, яке притягує рухомий контакт (якорь). В результаті цього контакту замикається або розмикається електричний ланцюг навантаження.

Основні характеристики реле:

- Напруга і струм спрацювання: мінімальні значення напруги або струму, необхідні для спрацювання реле.
- Напруга і струм відпускання: значення напруги або струму, при яких реле повертається в початковий стан.
- Час спрацювання і відпускання: швидкість, з якою реле реагує на зміну вхідного сигналу.
- Робочі струм і напруга: максимально допустимі значення струму і напруги в контактній групі реле.

- Внутрішній опір: опір котушки реле.

Існують два основних типи реле:

Електромагнітні реле: класичні реле, в яких рухомий контакт приводиться в дію електромагнітом.

Твердотільні реле: сучасні реле, в яких замість механічних контактів використовуються напівпровідникові елементи. Вони мають більш тривалий термін служби та менші розміри.

### Електромагнітне реле

Електромагнітне реле: це електромеханічний пристрій перемикання, дія якого ґрунтується на принципі електромагнетизму. Принцип роботи реле полягає в тому, що при проходженні електричного струму через котушку, створюється магнітне поле, яке притягує до себе феромагнітний яркір. Рух ярко́ра призводить до механічного перемикання контактів, що дозволяє керувати електричним ланцюгом навантаження.



Рис. 2.16. Електромагнітне реле

Електромагнітне реле: це електромеханічний пристрій, що використовується для дистанційного керування електричними ланцюгами. Принцип його роботи базується на взаємодії електромагнітного поля та механічних сил.

При подачі керуючого сигналу на котушку реле, створюється магнітне поле, яке притягує до себе феромагнітний яркір, переміщуючи його. Рух ярко́ра призводить до механічного замикання або розмикання електричних контактів, таким чином змінюючи стан керованого ланцюга. У відсутності керуючого сигналу, яркір повертається у вихідне положення під дією пружини або сили тяжіння.

					ІТС.4КІ.0125.03-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		54

Електромагнітні реле широко застосовуються в автоматичних системах керування, електроприводах, релейному захисті та інших галузях електротехніки. Вони дозволяють:

- Регулювати напругу та струм в електричних ланцюгах.
- Виконувати функції накопичувачів та перетворювачів електричної енергії.
- Фіксувати відхилення параметрів від заданих значень.

Електромагнітні реле класифікують за різними ознаками:

- За родом струму: на реле постійного та змінного струму.
- За конструкцією: на якірні та герконові реле.
- За швидкістю дії: розрізняють реле з різною швидкістю спрацювання.
- За ступенем захисту: реле можуть бути герметичними, закритими та відкритими.

### Твердотільні реле

Твердотільне реле (ТТР) – це електронний комутаційний пристрій, створений на основі напівпровідникових компонентів, таких як транзистори, симістори або тиристори. На відміну від електромагнітних реле, ТТР не має рухомих механічних частин, що забезпечує високу надійність та довговічність. Конструктивно ТТР часто виконуються за гібридною технологією [20], що передбачає комбінування різних електронних компонентів на єдиній підкладці. Така конструкція дозволяє створювати компактні та функціонально насичені пристрої.



Рис. 2.17. Твердотільне реле

Твердотільні реле: це електронні пристрої комутації, що базуються на

					ИТС.4КІ.0125.03-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		55

напівпровідникових компонентах, зокрема транзисторах, симісторах та тиристорах. На відміну від електромеханічних реле, твердотільні реле не мають рухомих частин, що забезпечує високу надійність, довговічність та безшумність роботи.

Переваги твердотілих реле:

1. Довгий термін служби: відсутність механічних контактів мінімізує знос і підвищує надійність.
2. Висока швидкість спрацювання: швидке перемикання без затримок.
3. Компактні розміри: можливість розміщення на друкованих платах.
4. Відсутність акустичних перешкод: тиха робота без виникнення іскор та дуг.
5. Низьке енергоспоживання: ефективне використання енергії.
6. Висока ізоляція: забезпечує безпеку роботи.
7. Стійкість до вібрацій та ударів: збереження працездатності в складних умовах.
8. Можливість використання у вибухонебезпечних середовищах: відсутність іскроутворення.

Твердотільне реле працює на основі оптичної ізоляції та керування силовими ключами. Керуючий сигнал подається на світлодіод, світловий потік якого через оптичну систему потрапляє на фотодіод. Сигнал з фотодіода керує силовим ключем (транзистором, симістором або тиристором), який і комутує навантаження.

Недоліки:

- Нагрівання: при великих струмах і напругах можливе значне нагрівання пристрою, що обмежує його робочі характеристики.
- Вартість: зазвичай твердотільні реле дорожчі за електромагнітні.

Твердотільні реле класифікують за такими ознаками:

- Тип навантаження: однофазні та трифазні.
- Спосіб управління: постійним або змінним струмом, ручне або автоматичне.

					ІТС.4КІ.0125.03-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

- Метод комутації: контроль переходу через нуль, довільне включення, фазове управління.

Твердотільні реле широко застосовуються в автоматичі, електроніці, побутовій техніці та промисловості. Одним з популярних прикладів є модулі типу SONGLE SRD-05VDC, які використовуються для керування навантаженнями в схемах на основі мікроконтролерів, таких як Arduino.

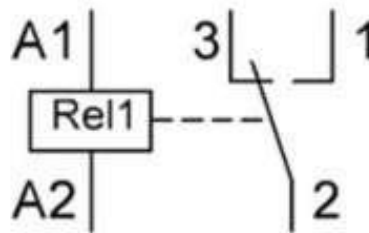


Рис. 2.18. Схема реалізації реле

Між A1 і A2 знаходиться металевий сердечник. При ввімкненні електричного струму, до нього притягнеться якір (2). 1, 3 - нерухомі контакти. За відсутності струму якір є близько контакту 3.

Схема підключення реле до Ардуїно зображена на рисунку 2.19

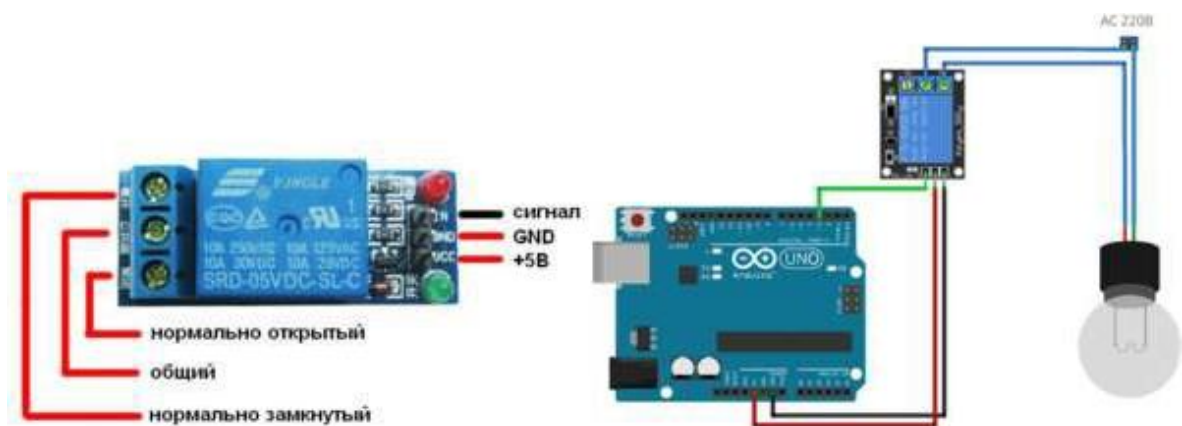


Рис. 2.19. Схема підключення реле до Arduino

Реле - це електромеханічний пристрій, який дозволяє керувати електричними ланцюгами за допомогою невеликого керуючого сигналу. Для підключення реле до мікроконтролера Arduino, як правило, використовують три основні контакти: загальний (COM), нормально відкритий (NO) та нормально замкнутий (NC).

Процес підключення:

					ІТС.4КІ.0125.03-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		57

1. Заземлення: контакт GND реле з'єднується з аналогічним контактом на платі Arduino.
2. Живлення: контакт VCC реле підключається до джерела живлення 5В на Arduino.
3. Керування: контакт In реле під'єднується до цифрового виходу Arduino, який буде подавати керуючий сигнал.

Більшість реле працюють за принципом інверсії: коли на вхід In подається високий логічний рівень (логічна одиниця), котушка реле знеструмлюється, а коли подається низький рівень (логічний нуль), котушка вмикається. Це означає, що для керування реле ми фактично змінюємо логічний стан на його вході.

Для більш плавного та надійного керування котушкою реле часто використовують транзистор. Коли на базу транзистора подається низький логічний рівень, він відкривається, дозволяючи протікати струму через котушку реле. Таким чином, реле включається. Для вимкнення реле на базу транзистора подається високий логічний рівень, що призводить до закриття транзистора і знеструмлення котушки.

Для візуального контролю за роботою реле часто використовують світлодіоди. Червоний світлодіод може сигналізувати про подачу живлення на реле, а зелений - про замикання контактів.

Для керування станом реле в програмі для Arduino використовується функція `digitalWrite()`. Встановивши логічний нуль на відповідному піні, ми вмикаємо реле, а встановивши логічну одиницю - вимикаємо.

Спрощена схема роботи:

При подачі живлення на Arduino транзистор, який керує реле, знаходиться у закритому стані.

Коли ми хочемо включити реле, ми виконуємо команду `digitalWrite()`, яка встановлює низький логічний рівень на піні, підключеному до бази транзистора.

Транзистор відкривається, струм протікає через котушку реле, і контакти реле змінюють свій стан відповідно до їх призначення (нормально відкриті замикаються, нормально замкнуті розмикаються).

					ІТС.4КІ.0125.03-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		58

Для вимкнення реле ми виконуємо команду `digitalWrite()`, яка встановлює високий логічний рівень на піні, закриваючи транзистор і знеструмлюючи котушку реле.

Підключення і керування реле за допомогою Arduino є відносно простим процесом, який вимагає розуміння основних принципів роботи цифрових виходів мікроконтролера та роботи самого реле. Використання транзистора для керування котушкою реле забезпечує надійну ізоляцію між мікроконтролером і потужним навантаженням, яке може підключатися до контактів реле.

#### 2.1.6. RFID-зчитувач RC-522

**Радіочастотна ідентифікація (RFID)** — це технологія, що широко застосовується в багатьох сферах для автоматичного розпізнавання об'єктів. Її можливості охоплюють відстеження персоналу, контроль доступу, оптимізацію ланцюгів поставок, управління бібліотечним фондом, реалізацію бонусних програм та інші прикладні завдання [23]. Візуальне представлення пристрою наведено на рисунку 2.20.



Рис. 2.20. Зчитувач карток RFID RC522

Система RFID фундаментально складається з двох ключових компонентів: **транспондера** (або мітки), що прикріплюється до об'єкта, та **пристрою для зчитування карток** (приймача). Їхня взаємодія показана на рисунку 2.21.

Прикладом такого зчитувача є RFID RC522.

					ITC.4KI.0125.03-ПЗ	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дата		

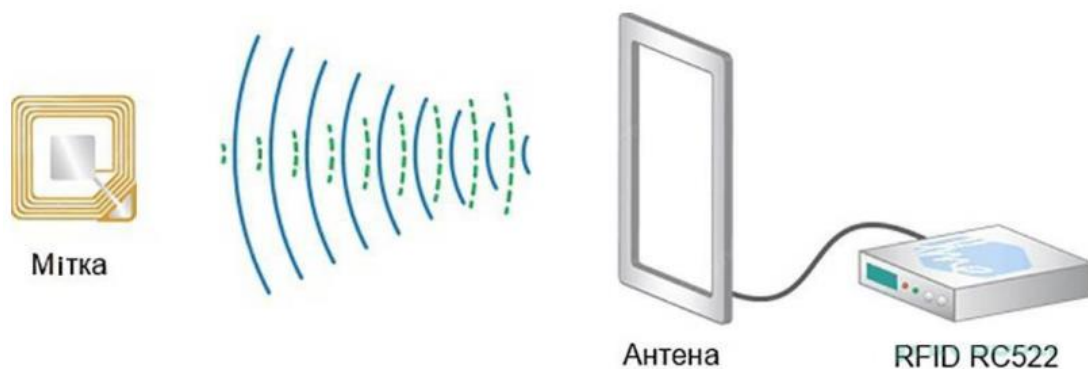


Рис. 2.21. Компоненти системи RFID

Зчитувач RFID-карток складається з радіочастотного модуля, блоку управління та антени, що генерує високочастотне електромагнітне поле. На відміну від зчитувача, RFID-мітка зазвичай є пасивним елементом, що містить лише антену та мікросхему.

Коли мітка (наприклад, карта або брелок) потрапляє в зону дії електромагнітного поля зчитувача, в її антені виникає індукційна напруга. Ця напруга слугує джерелом живлення для мікросхеми мітки, як це проілюстровано на рисунку 2.22.

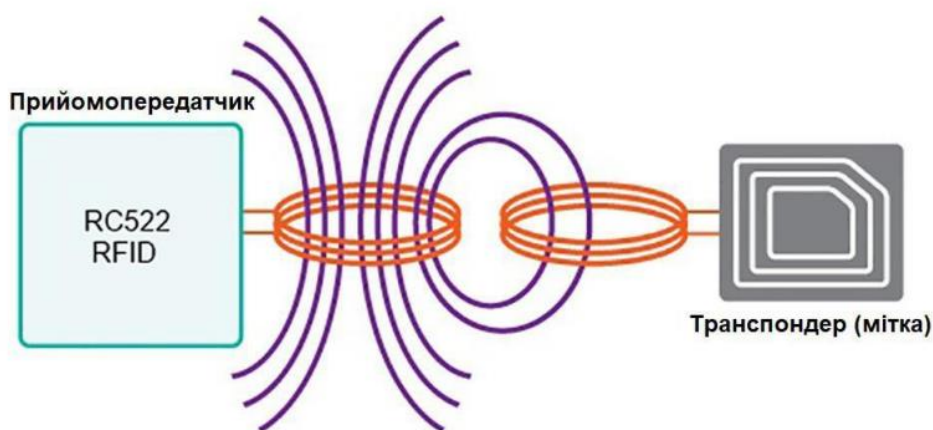


Рис. 2.22. Приймопередатчик та транспондер

Модуль RFID RC522, побудований на мікросхемі NXP MFRC522, призначений для радіочастотної ідентифікації. У комплект зазвичай входять 1 КБ RFID-картка та брелок.

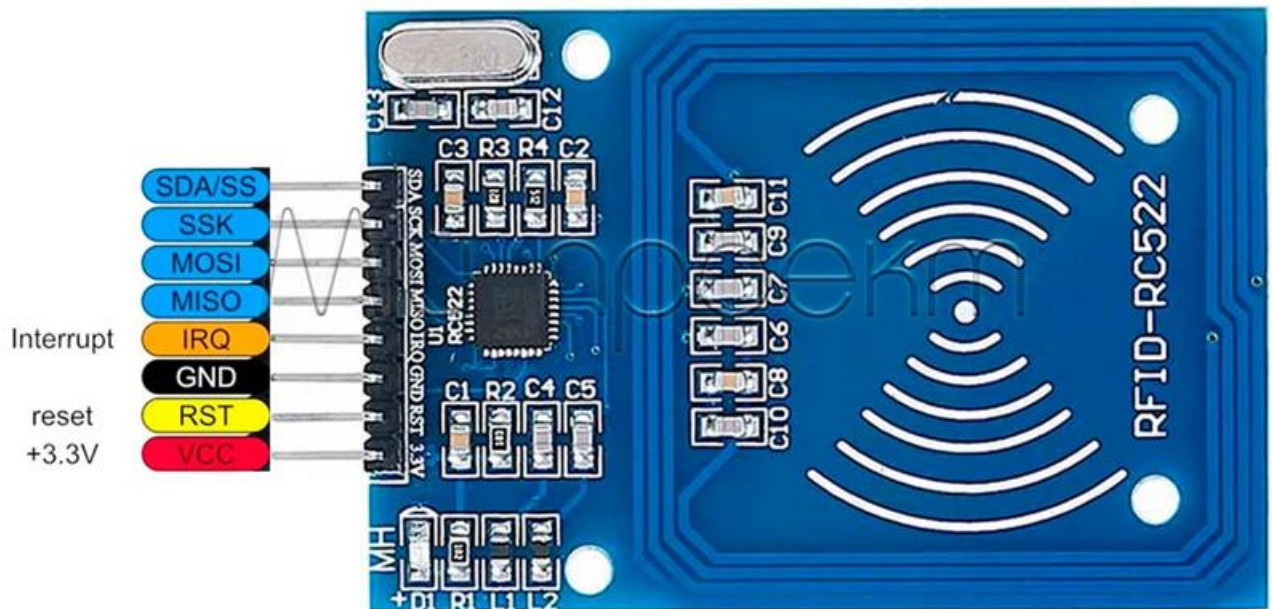


Рис. 2.23. Опис пінів плати RFID RC522

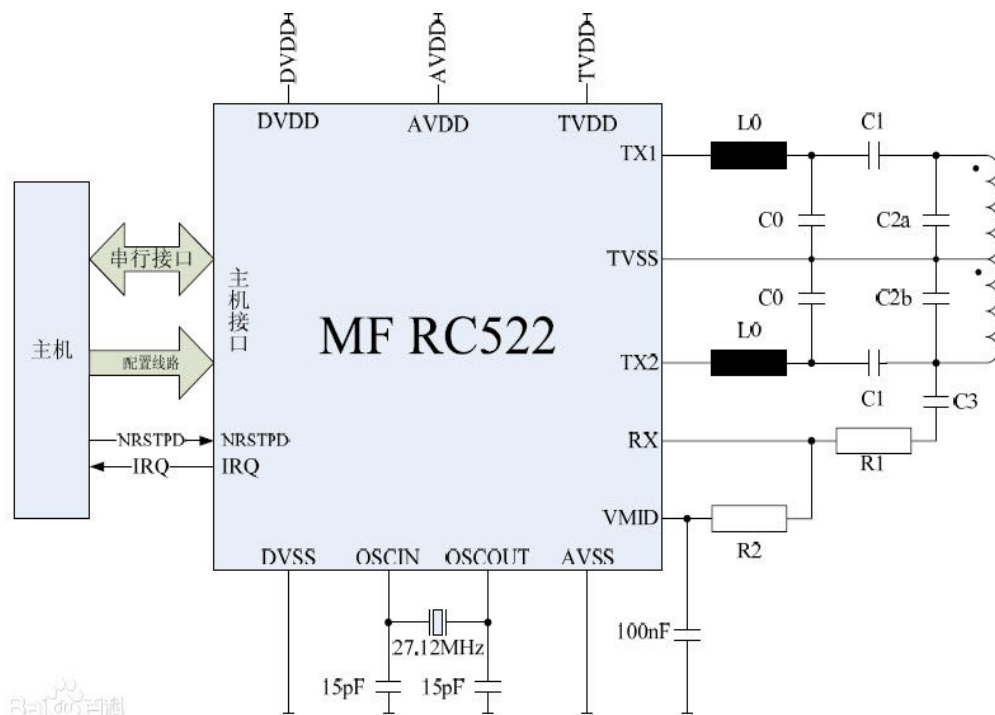


Рис. 2.24. Принципова схема модуля

Пристрій генерує електромагнітне поле на частоті 13,56 МГц для зв'язку з RFID-мітками стандарту ISO 14443A. Для інтеграції з мікроконтролерами модуль оснащений 4-контактним інтерфейсом SPI, а також може використовувати I2C та UART. Окремий контакт IRQ дозволяє оптимізувати роботу, активуючи опитування лише при виявленні мітки [24].

Роз'єми та живлення:

VCC: Вхід живлення (2.5-3.3 В).

RST: Управління скиданням і живленням (низький рівень вимикає модуль).

GND: Заземлення.

IRQ: Вивід для переривань.

MISO / SCL / Tx: Багатофункціональний вивід для SPI, I2S, UART.

MOSI: Вхід даних SPI.

SCK: Вхід тактових імпульсів.

SDA / SS / Rx: Багатофункціональний вивід для SPI, I2S, UART.

Незважаючи на робочу напругу 2,5-3,3 В, логічні виводи модуля є 5-вольт-сумісними. Це забезпечує пряме підключення до платформи Arduino та інших 5 В мікроконтролерів без потреби в додаткових перетворювачах логічного рівня.

Схема підключення RFID RC522 до Ардуїно зображена на рисунку 2.25.

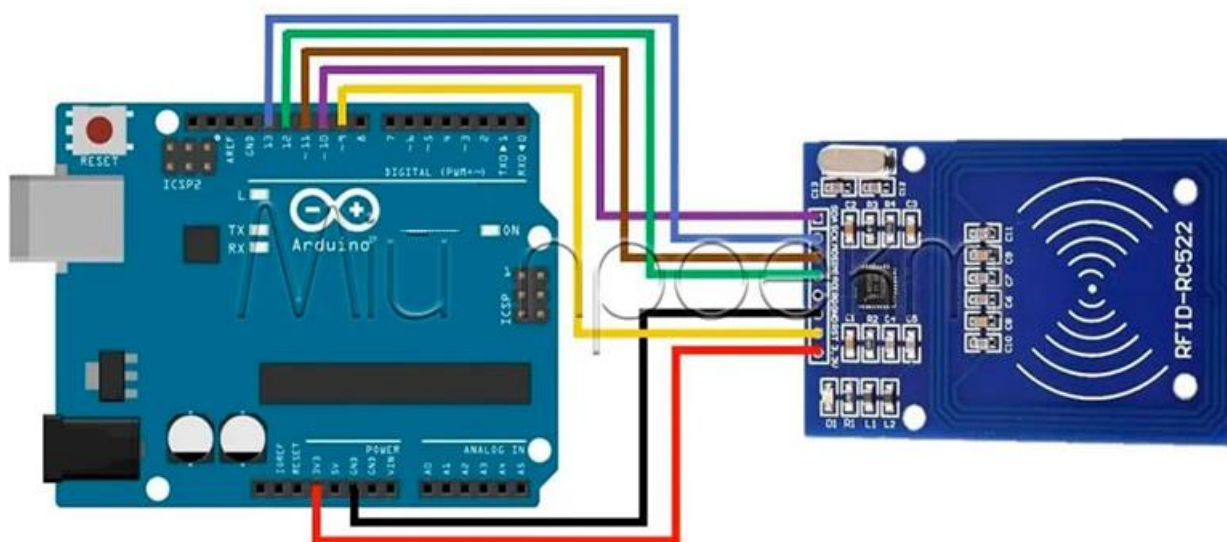


Рис. 2.25. Схема підключення RFID RC522 до Arduino

### 2.1.7. Ethernet Shield W5100

Мережевий шилд Ethernet Shield W5100 розширює функціональні можливості контролерів Arduino MEGA, перетворюючи їх на повноцінні мережеві пристрої. Це дозволяє використовувати Arduino як простий веб-сервер або здійснювати віддалене читання/запис його цифрових та аналогових входів/виходів через Інтернет. Шилд підтримується офіційною бібліотекою Ethernet Arduino, що спрощує його інтеграцію. Візуальне представлення пристрою наведено на рисунку 2.26.

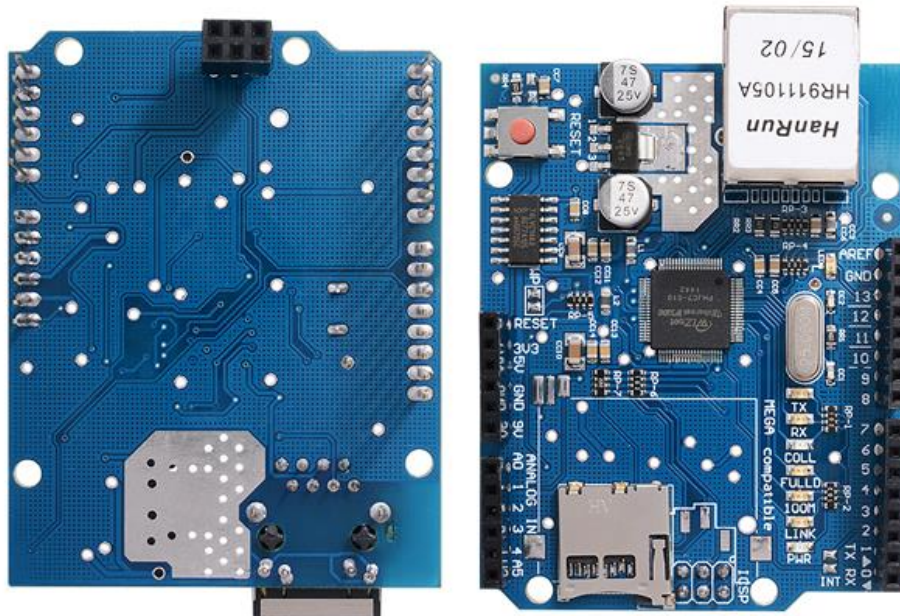


Рис. 2.26. Ethernet Shield W5100

Окрім мережевих функцій, плата підтримує читання та запис даних на міні-SD-карти (TF-карти), для яких передбачено відповідний роз'єм. Дизайн шилда є масштабованим: його можна не лише безпосередньо підключати до плати Arduino, а й встановлювати поверх нього інші шилди, створюючи багатошарові конфігурації.

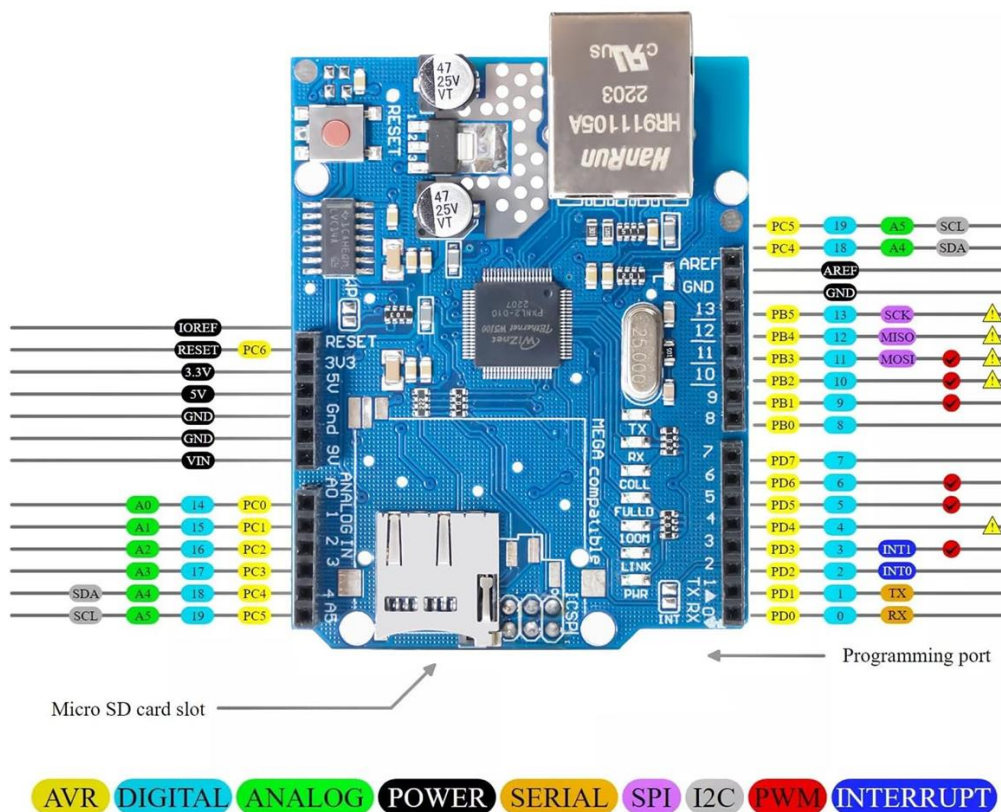


Рис. 2.27. Опис пінів Ethernet Shield W5100

Варто зазначити, що на платі встановлені резистори 51 Ом, що відповідає вимогам даташиту мікросхеми W5100. Однак це призводить до значного тепловиділення, тому для стабільної роботи може знадобитися додатковий радіатор.

Шилд надійно функціонує в мережах 10/100 Мбіт Ethernet, проте його сумісність з деякими 1000 Мбіт роутерами не гарантується.

Комунікація з шилдом, включно з взаємодією з мікросхемою W5100 та microSD-картою, відбувається через шину SPI. Це залучає виводи 11, 12, 13 для плат на Atmega8/168/328 або 50, 51, 52 для плат серії Mega (на Atmega1280/2560), які, відповідно, не можуть бути використані в програмах користувача (скетчах).

Для коректної роботи модуля SPI на платах серії Mega (Atmega1280/2560) необхідно додатково налаштувати пін 53 в режим виходу. Щоб забезпечити сумісність з усіма варіантами плат, з'єднання з шиною SPI реалізовано виключно через роз'єм ICSP (ISP).

Взаємодія з мікросхемою W5100 та картою microSD можлива лише по черзі, оскільки вони використовують спільну шину SPI. Вибір W5100 здійснюється активацією піна 10, а microSD — активацією піна 4.

Оскільки доступ до кнопки скидання на платі Arduino стає неможливим при встановленому шилді, на платі Ethernet Shield передбачена дублююча кнопка Reset для зручності використання.

Для візуальної індикації роботи на платі розміщено вісім світлодіодів:

- PWR: світиться, коли шилд отримує живлення (3.3 В).
- LINK: дублює жовтий світлодіод на роз'ємі RJ45, вказуючи на наявність мережевого з'єднання. Блимає під час прийому або передачі даних.
- 100M: дублює зелений світлодіод на роз'ємі RJ45, сигналізуючи про встановлення 100 Мбіт/с з'єднання.
- FULLD: вказує на активацію повнодуплексного (двонаправленого) з'єднання.
- COLL: блимає при виявленні колізій у мережі.
- RX: блимає під час прийому даних шилдом.

					ІТС.4КІ.0125.03-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		64

- TX: блимає під час передачі даних шилдом.
- L: дублює світлодіод "L" на платі Arduino, в даному випадку індикуючи вибір чіпа W5100.

Схема підключення Ethernet Shield W5100 до Ардуіно зображена на рисунку 2.28.

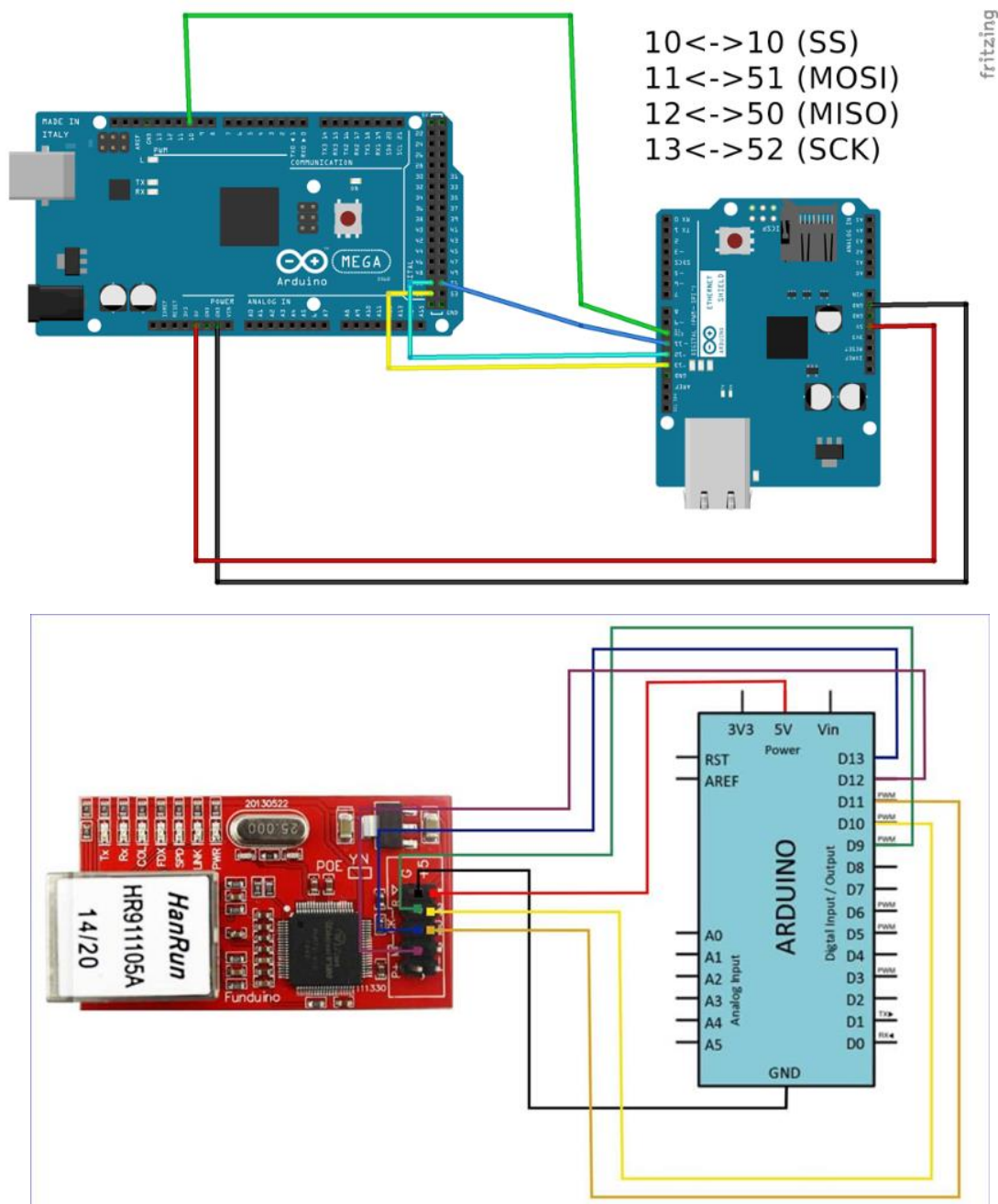


Рис. 2.28. Схема підключення RFID RC522 до Arduino

## 2.2. Огляд середовищ розробки для програмування мікроконтролерів

### 2.2.1. Огляд Arduino IDE

Arduino IDE - це інтегроване середовище розробки, призначене для створення програмного забезпечення для платформи Arduino. Воно надає

					ITC.4KI.0125.03-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		65

зручний інтерфейс для написання коду, його компіляції та завантаження в мікроконтролер. Середовище підтримує широкий спектр платформ і містить бібліотеки функцій для роботи з різноманітними датчиками та виконавчими пристроями.

#### Особливості Arduino IDE:

1. Середовище розробки Arduino IDE має лаконічний дизайн, що дозволяє швидко освоїти його навіть новачкам в програмуванні.
2. Arduino IDE використовує спрощену версію мови C++, що робить процес написання коду більш доступним для широкого кола користувачів.
3. Arduino IDE включає в себе велику кількість стандартних бібліотек, які спрощують роботу з різноманітними сенсорами, актуаторами та іншими периферійними пристроями.
4. Arduino IDE підтримує широкий спектр платформ Arduino, що дозволяє використовувати єдине середовище розробки для різних проектів.
5. Цей інструмент дозволяє відстежувати дані, що передаються між комп'ютером і платою Arduino, що значно спрощує процес налагодження програм.
6. Arduino IDE є проектом з відкритим кодом, що дозволяє користувачам вносити зміни до його функціональності та розширювати його можливості.

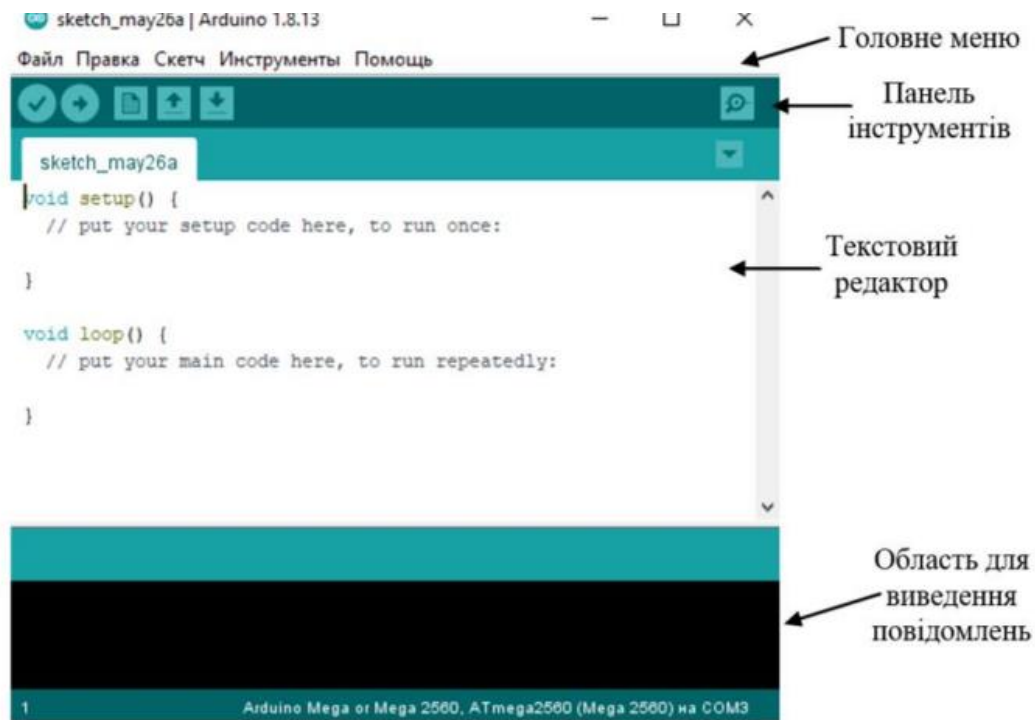


Рис. 2.29. Інтерфейс Arduino IDE

Скетчі, написані в середовищі розробки Arduino IDE, зберігаються у файлах з розширенням .ino. Редактор коду в IDE забезпечує стандартний набір функцій для роботи з текстом: копіювання, вставка, пошук і заміна. Інформація про хід виконання програми та можливі помилки відображається в області повідомлень. Вибір моделі плати Arduino та послідовного порту здійснюється в нижній частині вікна програми. Панель інструментів надає швидкий доступ до основних функцій середовища розробки. На панелі інструментів знаходиться 6 кнопок (рис. 2.30):



Рис. 2.30. Панель інструментів

### Основні функції Arduino IDE

- **Перевірити.** Дана опція дозволяє перевірити синтаксис написаного коду.

У разі виявлення помилок вони відображаються в області повідомлень.

- Завантажити. Використовується для компіляції програми та її завантаження у мікроконтролер Arduino.

- Створити. Дозволяє створити новий файл для написання скетчу.

- Відкрити. Призначена для відкриття існуючого файлу зі скетчем.

- Зберегти. Дозволяє зберегти поточний скетч у файл.

- Монітор послідовного порту. Використовується для відкриття програми Serial Monitor, яка відображає дані, що надходять від Arduino на комп'ютер через послідовний інтерфейс. Монітор підтримує роботу як із USB-варіантами плати, так і зі звичайними версіями Arduino. Для передачі даних на зовнішній пристрій у вікні монітора вводиться текст, після чого натискається кнопка "Відправити" або клавіша Enter. Швидкість передачі даних потрібно налаштувати відповідно до параметрів, вказаних у функції `Serial.begin()` у вашому коді. Функція `Serial.print()` дозволяє виводити текст у вікно монітора.

Бібліотеки значно розширюють функціональність програм, надаючи додаткові можливості, такі як робота з апаратними засобами або обробка даних. Для додавання бібліотеки слід перейти до меню Sketch, вибрати опцію Include Library і обрати необхідну бібліотеку. Після цього в програму додається оператор `#include`, а бібліотека компілюється разом зі скетчем. Важливо зазначити, що кожна підключена бібліотека використовує частину пам'яті мікроконтролера.

Більшість бібліотек уже попередньо встановлені разом із програмним забезпеченням Arduino, однак додаткові бібліотеки можна завантажити з зовнішніх джерел.

Для завантаження скетчу необхідно вибрати відповідну плату та порт (див. рис.2.31), які використовуються для вашої операційної системи. Це здійснюється через меню Tools, де у розділі Board вибирається модель плати, а в розділі портів — відповідний COM-порт (наприклад, COM1, COM2, COM4, COM5, COM7 тощо) або USB-з'єднання.

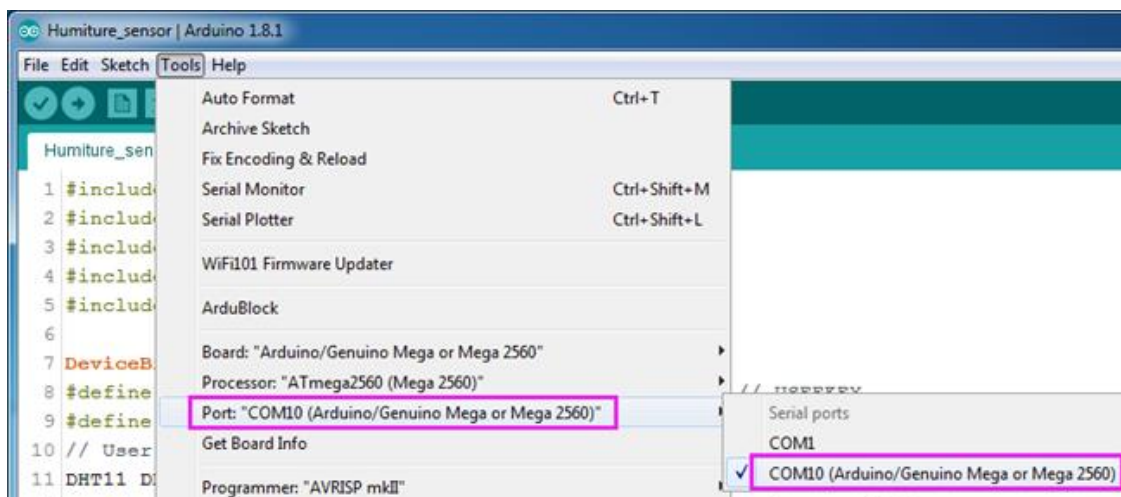


Рис. 2.31. Зображення вибору плати

Для завантаження програми в плату Arduino необхідно встановити з'єднання між комп'ютером і платою за допомогою USB-кабелю. В середовищі розробки Arduino IDE слід вибрати відповідну модель плати та послідовний порт, до якого підключено Arduino. Ця інформація необхідна для того, щоб комп'ютер міг правильно спілкуватися з платою.

Після вибору потрібних параметрів можна приступати безпосередньо до процесу прошивки.

Для цього необхідно скомпілювати написаний код. Компілятор перетворить ваш код, написаний на спрощеній версії C++, в машинний код, який розуміє мікроконтролер Arduino. Скомпільований код передається в буфер завантажувача - спеціальної програми, яка зашита в пам'ять мікроконтролера. Завантажувач, у свою чергу, переписує отриманий код в основну пам'ять мікроконтролера і запускає його на виконання.

Процес прошивки супроводжується скиданням мікроконтролера та миганням світлодіодів, що індикують активність передачі даних. Для забезпечення зручності користувача середовище розробки Arduino IDE надає інтуїтивно зрозумілий інтерфейс та автоматизує більшість рутинних операцій.

Функції `setup()` та `loop()` є основою будь-якого скетчу для Arduino.

- `setup()` виконується один раз на початку програми і використовується для ініціалізації портів введення-виведення, встановлення початкових значень змінних та підключення бібліотек.

- `loop()` виконується циклічно і містить основний алгоритм програми.

Все, що має повторюватися, розміщується саме тут.

Основні функції для роботи з портами введення-виведення:

- Цифрові порти: `pinMode()`, `digitalWrite()`, `digitalRead()`. Ці функції дозволяють керувати цифровими виходами (наприклад, світлодіодами) та зчитувати значення з цифрових входів (наприклад, кнопок).
- Аналогові порти: `analogWrite()`, `analogRead()`, `analogReference()`. Ці функції використовуються для роботи з аналоговими сигналами, такими як напруга.
- Спеціальні функції: `tone()`, `noTone()` для генерації звукових сигналів, `millis()` для вимірювання часу.

Arduino IDE широко використовується для розробки різноманітних електронних проектів, включаючи системи автоматизації будинку, робототехніку, інтернет речей та багато інших. Завдяки своїй простоті та універсальності, Arduino IDE стала популярним інструментом як для професійних розробників, так і для любителів електроніки.

Arduino IDE є потужним і зручним інструментом для розробки програмного забезпечення для платформи Arduino. Завдяки своїм можливостям та доступності, Arduino IDE стала стандартом де-факто для багатьох розробників, які працюють з мікроконтролерами.

### 2.2.2. Огляд Visual Studio

Microsoft Visual Studio - це комплексна платформа для розробки програмного забезпечення, що пропонує широкий спектр інструментів і функцій для створення різноманітних додатків. Цей продукт від компанії Microsoft є одним з найпопулярніших середовищ розробки у світі, що обумовлено його потужністю, універсальністю та постійним розвитком.

Visual Studio дозволяє розробникам створювати як консольні програми, так і програми з графічним інтерфейсом користувача. Підтримуються різноманітні технології, включаючи Windows Forms, ASP.NET для веб-розробки, а також мобільні платформи. Це робить Visual Studio універсальним інструментом для

					ІТС.4КІ.0125.03-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		70

розробки програмного забезпечення під різні операційні системи та пристрої [17].

Однією з ключових особливостей Visual Studio є наявність інтегрованого середовища розробки (IDE), яке об'єднує в собі редактор коду, компілятор, налагоджувач та інші необхідні інструменти. Редактор коду забезпечує інтелектуальне автодоповнення коду, підсвічування синтаксису та інші функції, що полегшують процес написання коду. Вбудований налагоджувач дозволяє ефективно виявляти та усувати помилки в програмах.

Крім того, Visual Studio включає в себе набір візуальних інструментів для створення графічних інтерфейсів, дизайну баз даних та веб-сайтів. Це дозволяє розробникам зосередитися на логіці програми, делегуючи рутинні завдання автоматизованим інструментам (рис. 2.32).

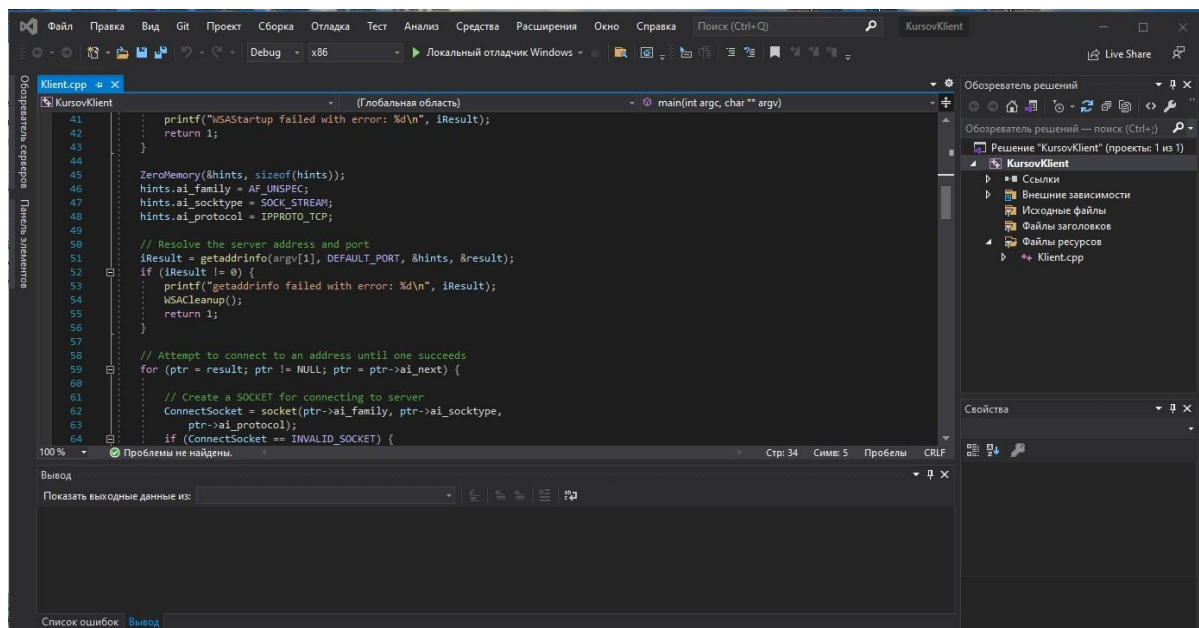


Рис. 2.32. Visual Studio

Visual Studio є одним з найпопулярніших інтегрованих середовищ розробки (IDE), що надають розробникам потужні інструменти для створення різноманітних програмних продуктів. Ця платформа підтримує широкий спектр мов програмування, включаючи C++, C#, F# та Visual Basic, що дозволяє розробникам обирати найбільш підходящий інструмент для вирішення конкретних завдань.

За допомогою Visual Studio можна розробляти як традиційні десктопні додатки для операційної системи Windows, так і сучасні веб-додатки, мобільні

					ИТС.4КІ.0125.03-ПЗ	Арк.
						71
Змн.	Арк.	№ докум.	Підпис	Дата		

програми та хмарні сервіси. Платформа пропонує широкий набір інструментів для створення користувацьких інтерфейсів, роботи з базами даних, а також для відлагодження та тестування програмного коду.

Однією з ключових переваг Visual Studio є підтримка різноманітних технологій і платформ. Це дозволяє розробникам створювати кросплатформні додатки, які можуть працювати на різних операційних системах і пристроях. Крім того, Visual Studio тісно інтегрується з іншими продуктами Microsoft, такими як Azure, що полегшує розробку хмарних додатків [18].

### **Висновки до розділу**

У другому розділі було проведено детальний аналіз основних компонентів та інструментів, необхідних для реалізації системи моніторингу мікроклімату серверної кімнати на базі платформи Arduino.

Було встановлено, що Arduino Mega2560 є оптимальним вибором для створення прототипів завдяки своїй доступності, простоті використання та широкій підтримці спільноти розробників. Датчики, такі як PIR-датчик, DHT11 та інші забезпечують збір необхідної інформації про навколишнє середовище, що є основою для прийняття рішень системою автоматизації.

Для розробки програмного забезпечення для Arduino було розглянуто кілька інструментів:

- Arduino IDE спеціалізоване середовище розробки, оптимізоване для роботи з платформою Arduino. Воно надає інтуїтивний інтерфейс та широкий набір бібліотек.
- Visual Studio універсальне середовище розробки, яке підтримує широкий спектр мов програмування та платформ. Воно може використовуватися для створення більш складних програмних додатків, які взаємодіють з системою "розумний дім".

Вибір конкретного інструмента залежить від складності проєкту, досвіду розробника та вимог до кінцевого продукту. Комбінація цих інструментів дозволяє створювати ефективні та надійні системи автоматизації.

## РОЗДІЛ 3. ПРОЄКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНО-АПАРATНОЇ СИСТЕМИ

### 3.1. Розробка структури пристрою

Структура пристрою являє собою сукупність елементарних ланок об'єкта та зв'язків між ними. Під елементарною ланкою розуміють частину об'єкта, системи керування тощо, яка реалізує елементарну функцію. Структурна схема призначена для наочного відображення основних функціональних вузлів пристрою, їхнього призначення та ролі в загальному процесі.

Загальна структурна схема включатиме такі функціональні блоки:

1. Блок живлення — пристрій, що здійснює перетворення напруги загальної електромережі до рівня, необхідного для функціонування системи.
2. Датчик температури — пристрій, призначений для моніторингу температурних змін у середовищі та генерування сигналу в разі досягнення критичних значень.
3. Інфрачервоний датчик руху — електронний сенсор, що вимірює інтенсивність інфрачервоного випромінювання, яке генерується об'єктами в його полі зору.
4. Зчитувач електронного ключа — пристрій, що здійснює зчитування ідентифікаційних параметрів для подальшої авторизації або відмови в доступі до приміщення.
5. ІЧ-передавач — пристрій, що забезпечує дистанційну передачу керуючих сигналів у поєднанні з інфрачервоним приймачем.
6. Модуль з двома реле — електромеханічний пристрій, що комутує електричне коло навантаження механічним шляхом за допомогою електромагнітного приводу.
7. Мікроконтролер — програмований пристрій, призначений для керування електронними компонентами системи.

					<b>ІТС.4КІ.0125.03-ПЗ</b>		
<b>Змн.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підпис</b>	<b>Дата</b>	<b>РОЗДІЛ 3. ПРОЄКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНО- АПАРATНОЇ СИСТЕМИ</b>		
Розроб.		Ліщук С.Е.					
Керівник		Донченко В.Ю.					
Реценз.		Козуб Ю.Г.					
Н. Контр.							
Зав. каф.		Семенов М.А.					
					<b>Літ.</b>	<b>Арк.</b>	<b>Акрушіє</b>
						73	36
					<b>ЛНУ</b>		
					<b>Кафедра ІТС, Гр.4КІ</b>		

8. Мережевий модуль — контролер Ethernet з функцією автоматичної конфігурації.
9. РК-дисплей (LCD) — один із типів плоскопанельних дисплеїв, що використовує рідкі кристали для візуалізації інформації.
10. SD-карта — енергонезалежний пристрій зберігання даних на основі флеш-пам'яті.
11. Веб-застосунок (Web-application) — програмний інтерфейс, доступний через веб-браузер, для відображення даних, отриманих від сенсорів.
12. Мобільний застосунок (Mobile-application) — програмний додаток, призначений для мобільних пристроїв, що забезпечує візуалізацію даних, отриманих від сенсорів.

### 3.2. Аналіз функціональних вимог

Діаграма варіантів використання являє собою графічне представлення взаємодії користувача з системою, що відображає зв'язок між користувачем та різними сценаріями використання, в яких він бере участь. Діаграма варіантів використання дозволяє ідентифікувати різні типи користувачів системи та відповідні варіанти використання, і часто доповнюється іншими типами діаграм для більш детального опису.

Основні компоненти діаграми варіантів використання включають:

**Актори:** користувачі, що взаємодіють із системою. Актор може бути фізичною особою, організацією або зовнішньою системою, яка обмінюється даними з досліджуваним застосунком або системою. Вони є зовнішніми сутностями, що генерують або споживають інформацію.

**Прецедент (Варіант використання):** конкретна послідовність дій та взаємодій між акторами та системою, спрямована на досягнення певної мети. Прецедент також може розглядатися як сценарій використання.

**Цілі:** кінцевий результат, якого прагне досягти користувач у більшості випадків використання. Ефективна діаграма варіантів використання повинна чітко описувати дії та альтернативні шляхи досягнення поставленої мети.

Для відображення взаємозв'язків між акторами та прецедентами в мові UML передбачено декілька стандартних типів відношень:

Відношення асоціації (association relationship): використовується для позначення специфічної ролі актора під час його взаємодії з окремим варіантом використання.

Відношення включення (include relationship): відображає залежність між двома варіантами використання, де поведінка одного варіанта (включеного) є невід'ємною частиною послідовності дій іншого (базового) варіанта використання.

Відношення розширення (extend relationship): визначає зв'язок між базовим варіантом використання та іншим варіантом, функціональна поведінка якого активується базовим лише за умови виконання певних додаткових критеріїв.

Відношення узагальнення (generalization relationship): застосовується для відображення ієрархічної залежності між варіантами використання, вказуючи на те, що дочірні варіанти успадковують усі характеристики поведінки батьківських варіантів [6].

При побудові діаграми варіантів використання ставилася мета продемонструвати основні дії, які зможе виконувати користувач та сама система моніторингу. Загальна діаграма варіантів використання з боку користувача представлена на рисунку 3.1.

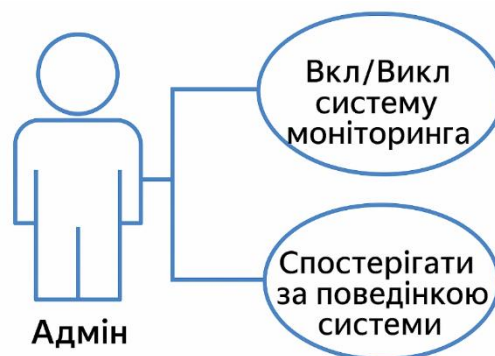


Рис. 3.1. Загальна діаграма варіантів використання з боку користувача

Конкретизація варіантів використання:

1) Увімкнення/вимкнення системи моніторингу:

Основна дійова особа (актор): адміністратор.

					ІТС.4КІ.0125.03-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		75

Інші учасники прецеденту: відсутні.

Зв'язки з іншими варіантами використання: відсутні.

Короткий опис: даний варіант використання забезпечує адміністратору можливість ініціації або припинення функціонування системи моніторингу.

2) Спостереження за функціонуванням системи:

Основна дійова особа (актор): адміністратор.

Інші учасники прецеденту: відсутні.

Зв'язки з іншими варіантами використання: відсутні.

Короткий опис: цей варіант використання надає адміністратору можливість здійснювати моніторинг процесів, що відбуваються в межах системи моніторингу.

На рисунку 3.2. зображено функції, які виконуються всередині системи автоматично, без безпосередньої участі користувача.

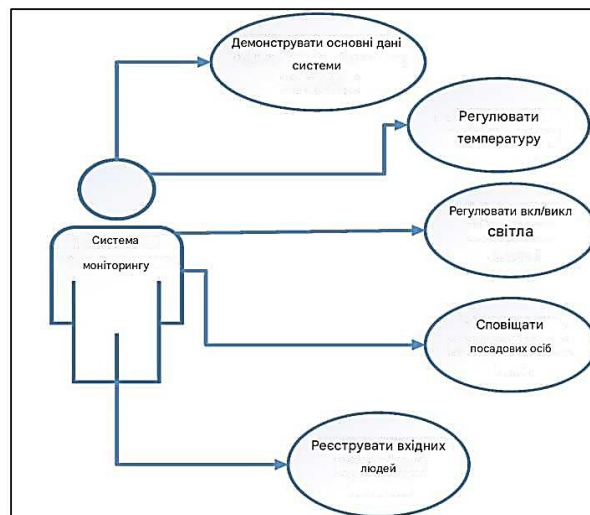


Рис. 3.2. Use-case діаграма (процеси всередині системи)

Конкретизація варіантів використання:

1) Демонстрація основних даних системи:

Основна дійова особа (актор): система моніторингу.

Інші учасники прецеденту: відсутні.

Зв'язки з іншими варіантами використання: відсутні.

Короткий опис: даний варіант використання забезпечує автоматизованій інформаційній системі (АІС) можливість відображення ключових параметрів, зібраних системою моніторингу.

2) Регулювання температури:

Основна дійова особа (актор): система моніторингу.

Інші учасники прецеденту: відсутні.

Зв'язки з іншими варіантами використання: відсутні.

Короткий опис: цей варіант використання дозволяє АІС автоматично регулювати роботу системи кондиціонування повітря при виявленні відхилення температури від заданого діапазону (перевищенні або зниженні відносно середнього значення).

3) Керування освітленням (увімкнення/вимкнення):

Основна дійова особа (актор): система моніторингу.

Інші учасники прецеденту: відсутні.

Зв'язки з іншими варіантами використання: відсутні.

Короткий опис: даний варіант використання забезпечує АІС функцію автоматичного ввімкнення освітлення при фіксації присутності персоналу в серверному приміщенні та його вимкнення після того, як приміщення буде залишено.

4) Оповіщення відповідальних осіб:

Основна дійова особа (актор): система моніторингу.

Інші учасники прецеденту: відсутні.

Зв'язки з іншими варіантами використання: відсутні.

Короткий опис: цей варіант використання надає АІС можливість автоматичного інформування відповідальних осіб у наступних випадках:

Вихід значення температури в серверному приміщенні за критичні межі, визначені в пункті 3 технічного завдання.

Несанкціоноване проникнення сторонніх осіб до приміщення.

5) Реєстрація осіб, що входять:

Основна дійова особа (актор): система моніторингу.

Інші учасники прецеденту: відсутні.

Зв'язки з іншими варіантами використання: відсутні.

Короткий опис: даний варіант використання забезпечує АІС функцію реєстрації всіх осіб, що входять до серверного приміщення.

					ІТС.4КІ.0125.03-ПЗ	Арк.
						77
Змн.	Арк.	№ докум.	Підпис	Дата		

### 3.3. Розробка програмно-апаратного комплексу

Основні вимоги до розроблюваної інформаційної системи викладено в технічному завданні.

#### 3.3.1. Проєктування програмно-апаратного комплексу

Для наочного представлення динаміки функціонування програмно-апаратного комплексу було розроблено модель станів. Ця модель відображає переходи об'єкта з одного стану в інший та слугує для моделювання динамічних аспектів системи. Модель станів представлена на рисунку 3.3.

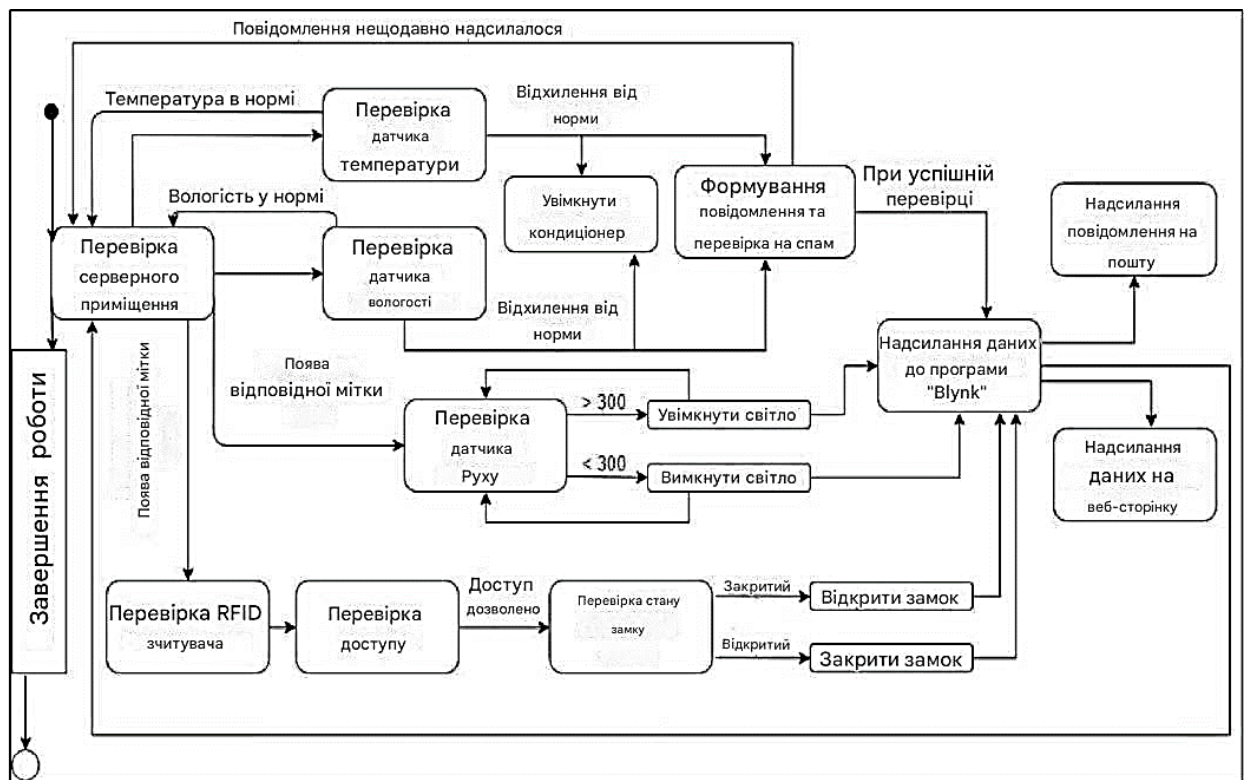


Рис. 3.3. Модель станів програмно-апаратного комплексу

Модель компонентів дозволяє визначити архітектуру системи, що розробляється, встановивши залежності між програмними компонентами, в ролі яких може виступати вихідний, бінарний і виконуваний код. Багато середовищах розробки модуль або компонент відповідає файлу. Пунктирні стрілки, що з'єднують модулі, показують відносини взаємозалежності, аналогічні до тих, що мають місце при компіляції вихідних текстів програм. На рисунку 3.4 показана модель компонентів системи, що розробляється.

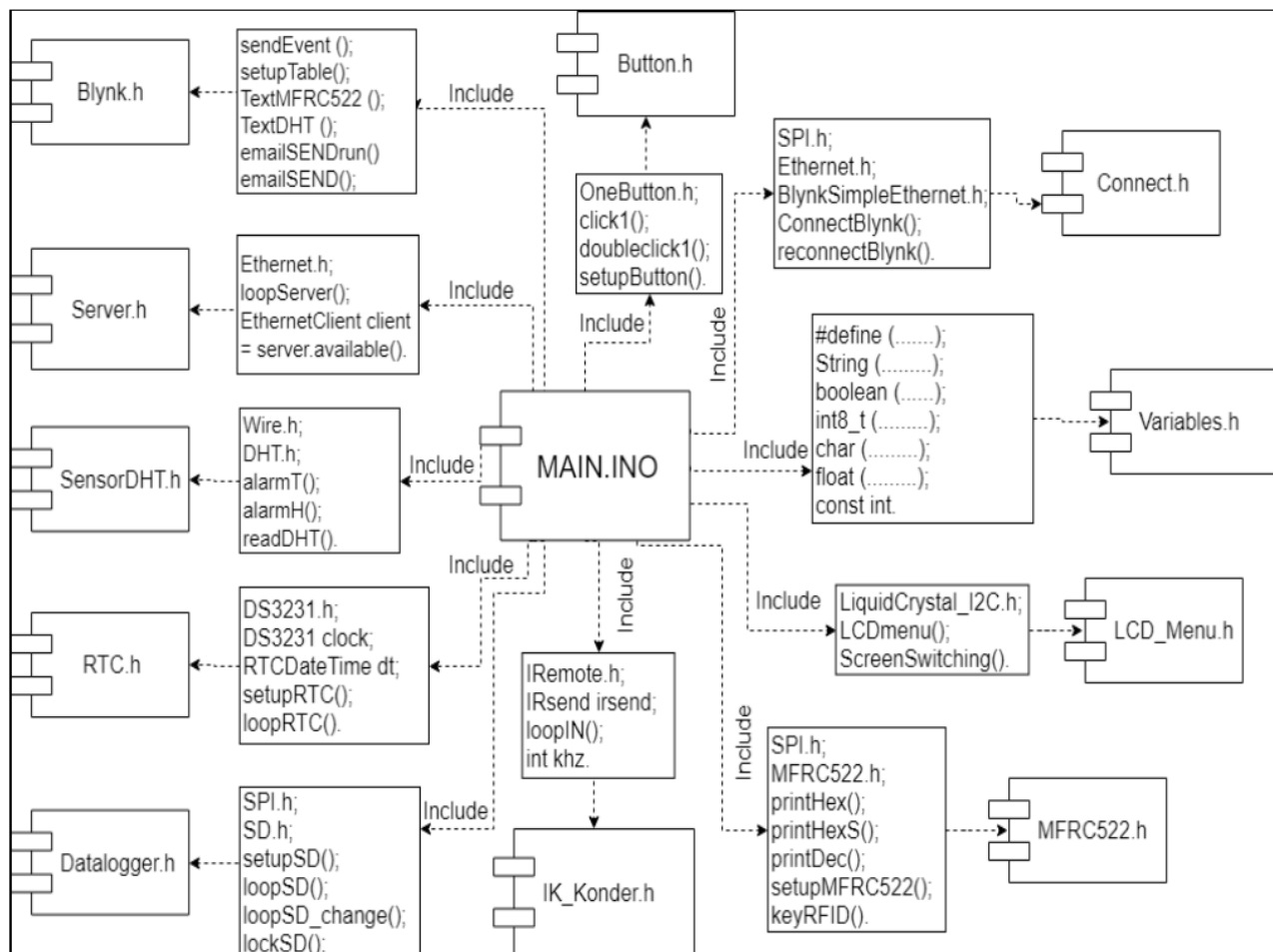


Рис. 3.4. Модель компонентів

Також була побудована архітектура комплексу, що містить графічні зображення датчиків, пристроїв, процесів та зв'язків між ними. Дана діаграма показана рисунку 3.5.

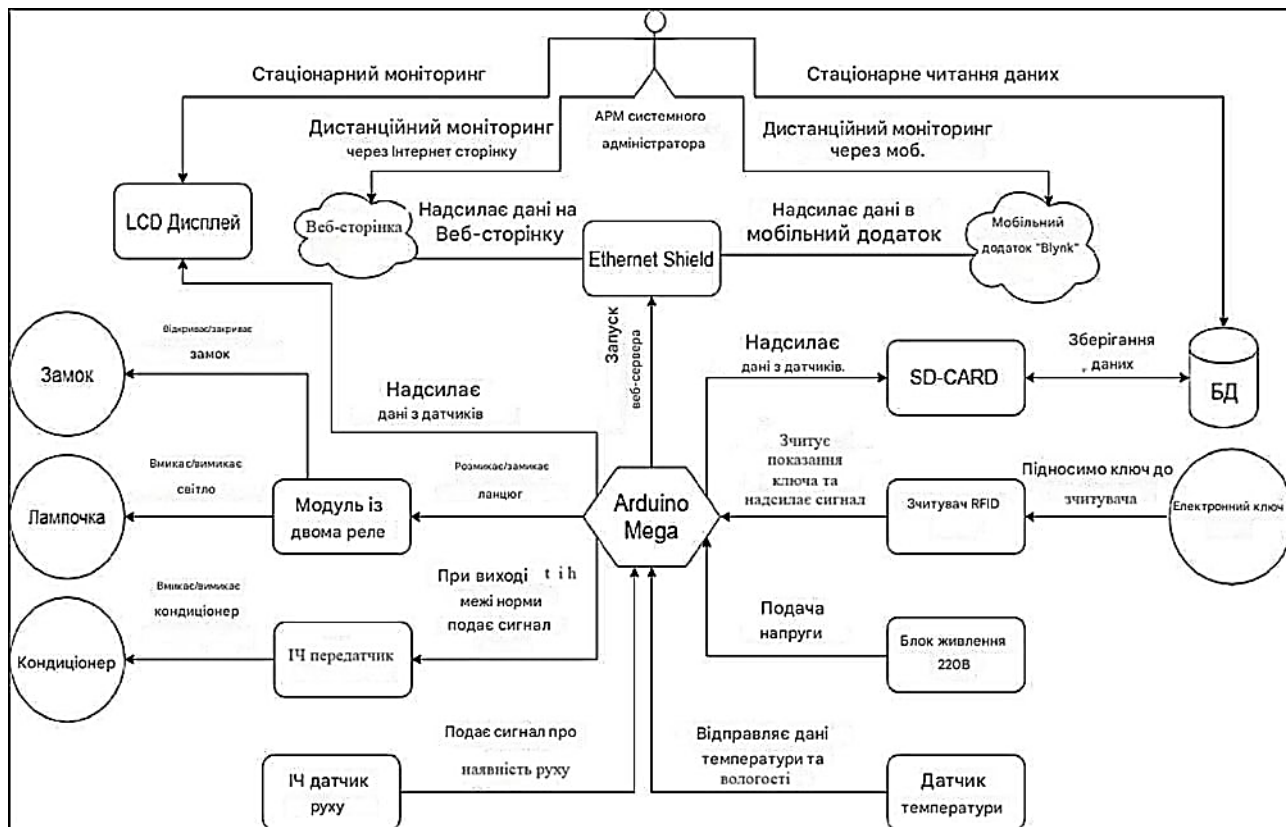


Рис. 3.5. Архітектура комплексу

Пояснювальна записка до архітектури комплексу представлена на рисунку

3.6.

	Пристрій, який організовує виконання всіх процесів системи
	Робоче місце співробітника
	Датчики, що підключаються до пристрою
	Сторонні компоненти, які взаємодіють із системою
	База даних
	Віддалене зберігання і візуалізація оброблених системою даних

Рис. 3.6. Пояснювальна записка до архітектури комплексу

### 3.3.2. Програмування датчиків та модулів

Перед етапом фізичної збірки макету мікрокомплексу було розроблено загальну схему підключення компонентів рисунок 3.7.

Змн.	Арк.	№ докум.	Підпис	Дата

ІТС.4КІ.0125.03-ПЗ

Арк.

80

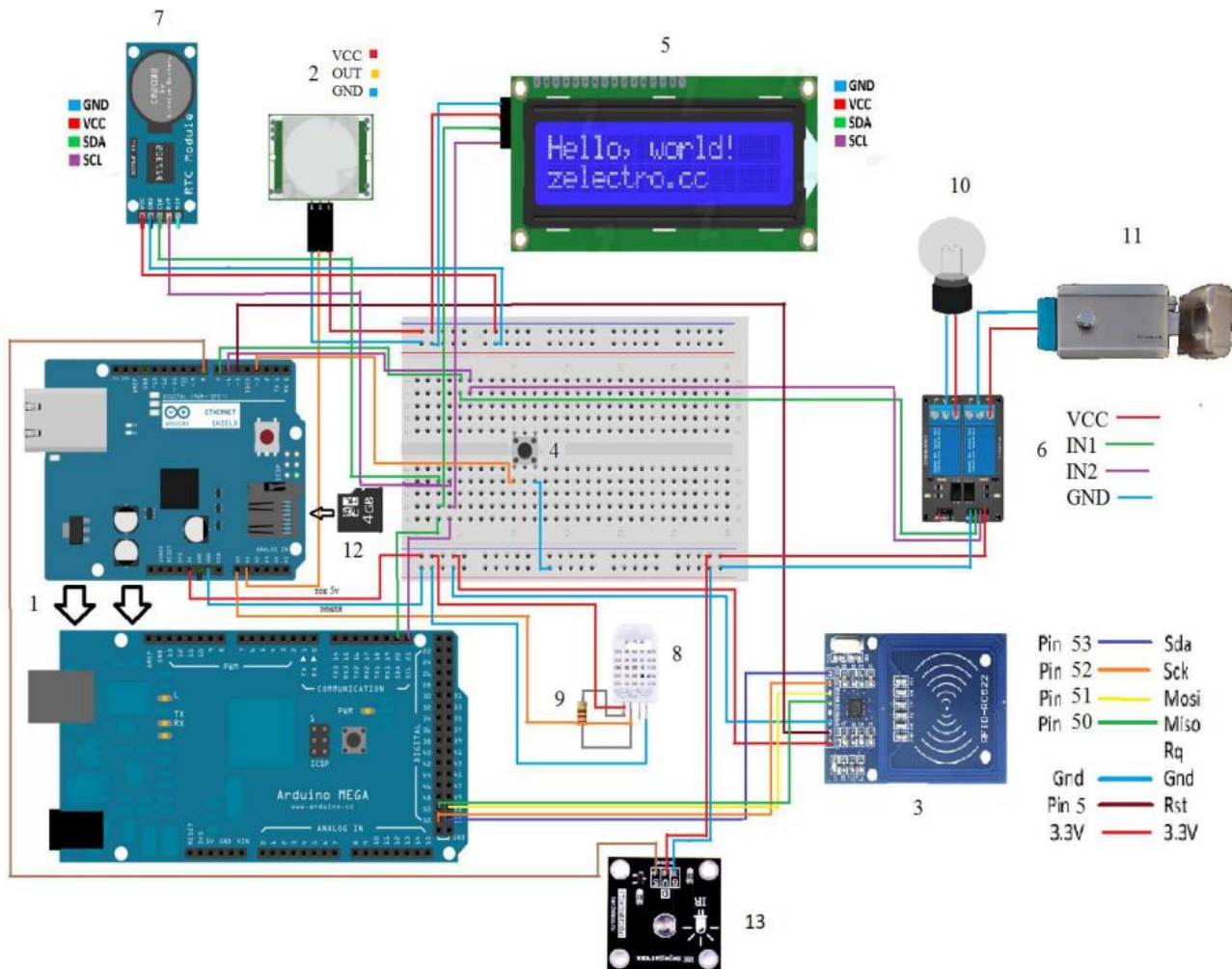


Рис. 3.7. Схема підключення датчиків та модулів

Таблиця 3.1

### Перелік компонентів

№	Назва	Пояснення
1	Mega2560 + Ethernet Shield	Платформа для роботи з модулями і датчиками. Ethernet використовується для підключення до інтернету.
2	Інфрачервоний датчик руху (піроелектричний)	Використовується для вмикання/вимикання світла.
3	Зчитувач RFID MFRC-522	Використовується для здійснення пропускнуго режиму (світлодіод використовується як замінник замка).
4	Кнопка тактова	Використовується для перемикання пунктів меню на LCD-екрані.
5	LCD дисплей 1602 (символьний на HD44780, з підсвіткою, зелений)	Використовується для відображення параметрів, що передаються датчиками (має 3 пункти меню).

№	Назва	Пояснення
6	Модуль з двома реле для Arduino	Використовується у зв'язку з ІЧ-датчиком руху і RFID для перемикання світла (вкл/викл), а також відкриття/закриття електромеханічного замка.
7	Прецизійний годинник реального часу DS3231	Дозволяє рахувати поточний час і дату для запису даних на SD-карту.
8	Цифровий датчик температури і вологості DHT11	Використовується для вимірювання температури і вологості в приміщенні.
9	Резистор із опором 10 кОм	
10	Лампочка накаливання	Вмикання/вимикання світла.
11	Електромеханічний замок	Відкриття/закриття дверей.
12	Micro SD-карта	Використовується для зберігання параметрів у БД.
13	ІЧ-передавач	Служить для управління кондиціонером.

Мікрокомплекс на базі мікроконтролера для контролю параметрів серверної кімнати складається з наступних модулів і їх функцій:

### 1. Модуль температури та вологості

Він виконує функцію контролю та регулювання температури в серверній. Коли температура підвищується/знижується вище / нижче середнього значення, він повинен регулювати роботу кондиціонера.

### 2. Модуль підсвічування

Він виконує функцію увімкнення/вимкнення світла у разі наявності/відсутності персоналу в серверній кімнаті відповідно.

### 3. Модуль контролю доступу

Виконує функцію реєстрації осіб, які заходять на територію приміщення.

### 4. Стационарний модуль візуалізації

Він виконує функцію демонстрації основних параметрів приміщення на рідкокристалічному екрані пристрою візуального відображення інформації.

### 5. Модуль візуалізації

Він виконує функцію демонстрації основних параметрів приміщення на веб-сторінці та в мобільному додатку Blink.

## **6. Модуль зберігання**

Виконує функцію зберігання даних з датчиків на SD-карті.

### **3.3.2.1. Використання бібліотек для розширення функціональності програмного забезпечення**

Для ефективної розробки програмного забезпечення для системи було використано механізм бібліотек. Бібліотека Arduino представляє собою набір функцій та структур даних, що зберігається окремо від основного коду програми [30]. Цей підхід дозволяє модуляризувати програмний код, спрощуючи його структуру та підтримку.

Використання бібліотек значно спрощує роботу з різноманітними периферійними пристроями, такими як датчики, актуатори, дисплеї тощо. Бібліотеки надають готові функції для взаємодії з цими пристроями, що дозволяє розробнику зосередитися на розв'язанні конкретного завдання, не заглиблюючись у деталі низькорівневого програмування.

Для реалізації функціональності системи були використані такі бібліотеки: LiquidCrystal\_I2C.h, OneWire.h, GyverButton.h, що дозволило значно скоротити час розробки та підвищити надійність програмного забезпечення.

#### **LiquidCrystal\_I2C.h**

Для візуалізації інформації в розробленій системі було використано рідкокристалічний дисплей, керований за допомогою бібліотеки LiquidCrystal\_I2C [31]. Ця бібліотека є частиною стандартного набору Arduino IDE і забезпечує зручний інтерфейс для взаємодії з дисплеями, що підключені за протоколом I2C.

Бібліотека LiquidCrystal\_I2C надає широкий спектр функцій для управління дисплеєм, включаючи ініціалізацію, очищення, встановлення курсора, виведення тексту, створення користувацьких символів та керування підсвічуванням. Детальний опис основних функцій наведено нижче:

- Ініціалізація дисплея: функція `begin(cols, rows, [char_size])` використовується для початкової настройки дисплея, включаючи

					<i>ІТС.4КІ.0125.03-ПЗ</i>	Арк.
						83
Змн.	Арк.	№ докум.	Підпис	Дата		

встановлення кількості стовпців і рядків, а також розміру символів.

- Очищення та позиціонування курсора: функції `clear()`, `home()`, `setCursor(col, row)` дозволяють очистити дисплей, встановити курсор у початкову позицію або перемістити його в довільну точку дисплея.
- Керування відображенням: функції `display()`, `noDisplay()`, `blink()`, `noBlink()`, `cursor()`, `noCursor()` дозволяють керувати видимістю дисплея, миганням курсора та підкресленням символів.
- Прокручування дисплея: функції `scrollDisplayLeft()` та `scrollDisplayRight()` дозволяють прокручувати вміст дисплея вліво або вправо.
- Керування напрямком прокручування тексту: функції `leftToRight()` та `rightToLeft()` дозволяють встановити напрямок прокручування тексту.
- Створення користувацьких символів: функція `createChar(num, array)` дозволяє створити власні символи та зберігати їх у пам'яті дисплея.
- Виведення тексту: функція `print(text)` використовується для виведення текстової інформації на дисплей.
- Керування підсвічуванням: функції `backlight()`, `noBacklight()` та `setBacklight(flag)` дозволяють керувати підсвічуванням дисплея.

Для підключення дисплея до мікроконтролера Arduino необхідно вказати його адресу на шині I2C, кількість стовпців і рядків. Приклад коду для ініціалізації дисплея:

```
#include <Wire.h>

#include <LiquidCrystal_I2C.h> LiquidCrystal_I2C lcd (address, col, row); void
setup () {
    lcd.init ();
}
```

Де параметр:

- address: Адреса дисплея на шині I2C - 0x27 або 0x3F;
- col: кількість стовпців, які реалізовані у дисплея;3

					<i>ІТС.4КІ.0125.03-ПЗ</i>	Арк.
						84
Змн.	Арк.	№ докум.	Підпис	Дата		

- row: кількість рядків, які реалізовані у дисплея.

Використання бібліотеки LiquidCrystal\_I2C значно спрощує процес розробки програмного забезпечення для керування рідкокристалічними дисплеями, дозволяючи розробникам зосередитися на вирішенні більш складних завдань.

### OneWire.h

Інтерфейс 1-Wire відрізняється простотою реалізації, оскільки для обміну даними вимагає лише одного сигнального проводу та заземлення. Однією з унікальних особливостей цього інтерфейсу є можливість паразитного живлення підключених пристроїв без використання додаткового джерела живлення. Сигнальний провід одночасно слугує для передачі даних та живлення пристроїв, що мають достатню ємність для накопичення енергії. Така конструкція дозволяє спростити схему підключення та зменшити кількість необхідних компонентів.

Для початку сеансу обміну даними на шині 1-Wire необхідно виконати процедуру скидання (reset). Майстер (контролер) ініціює скидання шляхом встановлення логічного нуля на сигнальному проводі протягом мінімум 480 мкс. Після цього майстер відпускає лінію, дозволяючи їй перейти в невизначений стан. Завдяки резистору підтяжки, підключеному до лінії даних, вона зазвичай встановлюється в логічну одиницю.

Якщо на шині присутні пристрої, готові до обміну даними, вони реагують на сигнал скидання. Для підтвердження готовності пристрої активізують свій вихід, тимчасово опускаючи рівень сигналу на лінії до логічного нуля на проміжок часу від 60 до 240 мкс. Таким чином, майстер, прочитавши стан лінії після скидання, може визначити, чи присутні на шині пристрої, готові до взаємодії.

Ключові особливості ініціалізації:

- Скидання (reset): активний низький сигнал тривалістю не менше 480 мкс, що ініціює сеанс обміну.
- Підтвердження скидання: активний низький сигнал від пристроїв тривалістю 60-240 мкс, що свідчить про їхню готовність до обміну.

					ІТС.4КІ.0125.03-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		85

- Резистор підтяжки: забезпечує встановлення лінії даних у логічну одиницю в неактивному стані.

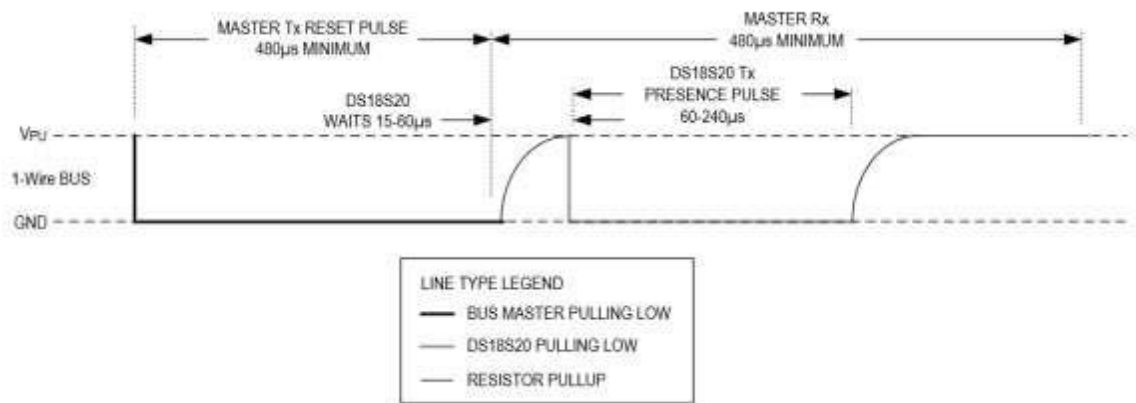


Рис. 3.8. Старт обміну даними

Інтерфейс 1-Wire використовує нестандартний підхід до передачі даних: замість традиційного кодування бітів рівнями сигналу, він застосовує тимчасове кодування. Логічна одиниця передається коротким імпульсом низького рівня тривалістю 15 мкс, після чого лінія переходить у високий стан під дією підтягувального резистора. Логічний нуль передається більш тривалим імпульсом низького рівня – від 60 до 120 мкс. Такий метод кодування дозволяє зменшити кількість необхідних елементів у схемах пристроїв.

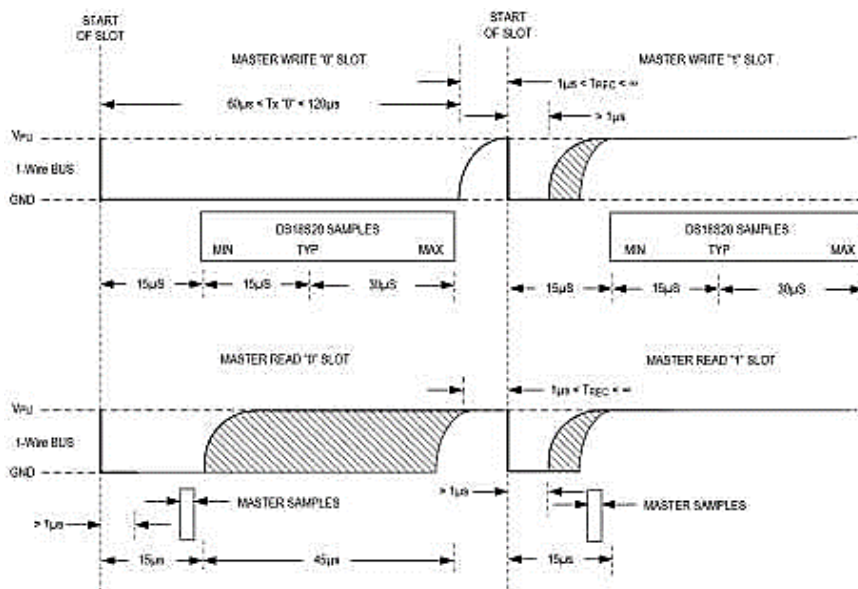


Рис. 3.9. Передача даних 1-Wire

## GyverButton.h

Бібліотека GyverButton.h призначена для спрощення роботи з тактовими кнопками в середовищі Arduino. Вона дозволяє ефективно обробляти

					ІТС.4КІ.0125.03-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		86

різноманітні події, пов'язані з натисканням кнопки, такі як одиночне натискання, утримання, подвійне та потрійне натискання.

Функціональні можливості бібліотеки:

- універсальність: бібліотека підтримує роботу як з нормально замкнутими, так і з нормально розімкнутими кнопками, а також дозволяє використовувати різні схеми підключення з підтягувальними резисторами.
- фільтрація дрижання контактів: вбудований механізм дебаунсингу дозволяє усунути помилкові спрацювання, спричинені дрижанням контактів кнопки.
- налаштовані параметри: можливість налаштування часу дебаунсингу, таймауту утримання, інтервалу між повторними натисканнями та інших параметрів.
- різноманітні події: бібліотека дозволяє відстежувати такі події, як натискання, відпускання, утримання, одиночне, подвійне та потрійне натискання, а також зміну стану кнопки з заданим кроком за часом.
- просте використання: інтуїтивно зрозумілий інтерфейс та набір функцій, що дозволяють легко інтегрувати бібліотеку в будь-який Arduino проєкт.

### Основні класи та функції

**Клас *GButton*:** основний клас для роботи з кнопкою. При створенні об'єкта цього класу необхідно вказати номер цифрового виводу, до якого підключена кнопка, а також додаткові параметри, такі як тип кнопки (нормально замкнута або нормально розімкнута) та схема підключення (з підтягувальним резистором або без).

Функції для налаштування:

`setDebounce()`: встановлює час дебаунсингу.

`setTimeout()`: встановлює таймаут утримання.

`setStepTimeout()`: встановлює інтервал між зміною стану при утриманні кнопки.

`setType()`: встановлює тип кнопки.

					ІТС.4КІ.0125.03-ПЗ	Арк.
						87
Змн.	Арк.	№ докум.	Підпис	Дата		

setTickMode(): вибирає режим опитування кнопки (ручний або автоматичний).

Функції для отримання інформації:

isPress(), isRelease(), isClick(), isHoled(), isHold(), isSingle(), isDouble(), isTriple(), hasClicks(), getClicks(), isStep(): Повертають інформацію про стан кнопки та події, що відбулися.

Функція tick(): виконує опитування стану кнопки та оновлює внутрішні змінні бібліотеки.

### 3.3.2.2. Модуль температури та вологості

Завдання модуля температури та вологості полягає у здійсненні моніторингу та регулюванні температури і вологості в серверному приміщенні, а також у формуванні повідомлень для подальшого оповіщення відповідальних осіб у разі досягнення критичних значень.

Для реалізації цього завдання було використано:

- Датчик температури та вологості DHT11.
- З'єднувальні проводи для макетних плат (типу "мама-тато").

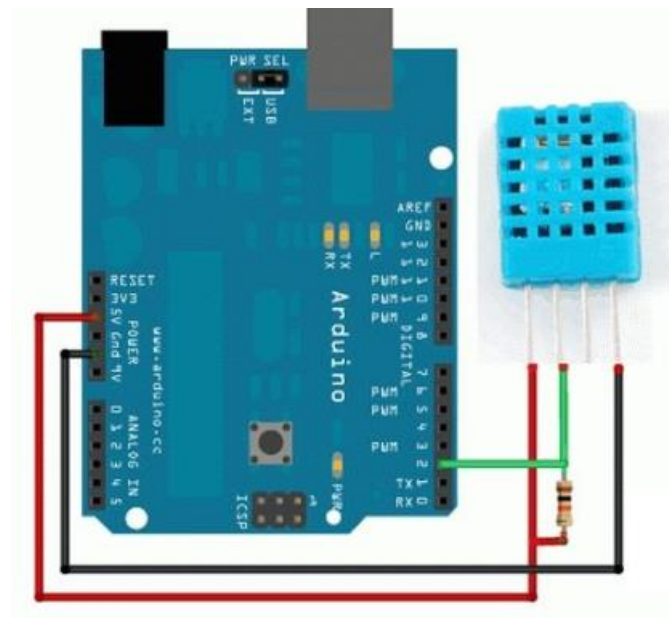


Рис. 3.10. Схема підключення датчика DHT11

Логіка роботи модуля температури та вологості продемонстрована на рисунку 3.11.

```

#include <Wire.h>
#include "DHT.h" //Підключаємо бібліотеку термодатчик.
#define DHTTYPE DHT21 // DHT 21 (AM2301) //Тут вибираємо який у нас датчик.
DHT dht (DHTPIN, DHTTYPE); // Оголошуємо датчик
// Перевіряємо, чи знаходиться температура в нормі, чи вийшла за встановлені межі
// Піднімаємо та опускаємо відповідні прапори
// Опускаємо прапор, дозволяємо повторну відправку повідомлення на пошту) при приході температури в норму
void alarmT(){
    if (t > t_max) {t_alarm_max=true;} else {t_alarm_max=false; send_t_alarm_max=false;}
    if (t < t_min) {t_alarm_min=true;} else { t_alarm_min = false ; send_t_alarm_min = false ;}
}
// Те саме для вологості
void alarmH() {
    if (h > h_max) {h_alarm_max = true;} else {h_alarm_max = false; send_h_alarm_max = false ;}
    if (h < h_min) {h_alarm_min = true;} else {h_alarm_min = false; send_h_alarm_min = false ;}
}
// Читаємо дані з датчика DHT, Відправляємо дані до Blynk
// і у випадку аварії відправляємо повідомлення на пошту
void readDHT(){
    h = dht.readHumidity();
    t = dht.readTemperature(); // або dht.readTemperature(true) for Fahrenheit
    // Якщо дані лічені, то рядки нижче будуть виконані
    if (isnan(h) || isnan(t)) {
        Serial.println(F("Failed to read from DHT sensor!"));
        return;
    }
    Blynk.virtualWrite(VPIN_LABEL_TEMP, t);
    Blynk.virtualWrite(VPIN_LABEL_HUMID, h); // Отправляем дані в Blynk
    alarmT(); // Перевіряємо аварії для Температури
    alarmH(); // Перевіряємо аварії для Вологості

/* Якщо (прапор аварії і чи не було відправки повідомлення про аварію)
 * то відправимо повідомлення і піднімо прапор про те, що ми вже відправили повідомлення, для захисту від спаму
 на електронну пошту */
    if (t_alarm_max && !send_t_alarm_max)
        {emailSEND(TextDHT ("Обережно! Перевищено норму температури", String(t)));
    send_t_alarm_max=true;}
    if (t_alarm_min && !send_t_alarm_min)
        {emailSEND(TextDHT ("Обережно! Температура нижче норми", String(t))); send_t_alarm_min=true;}
    if (h_alarm_max && !send_h_alarm_max)
        {emailSEND (TextDHT ("Обережно! Перевищена норма вологості", String(h))); send_h_alarm_max=true;}
    if (h_alarm_min && !send_h_alarm_min)
        {emailSEND(TextDHT ("Обережно! Вологість нижче за норму", String(h))); send_h_alarm_min=true;}
}

```

Рис. 3.11. Логіка роботи модуля температури та вологості

Змн.	Арк.	№ докум.	Підпис	Дата

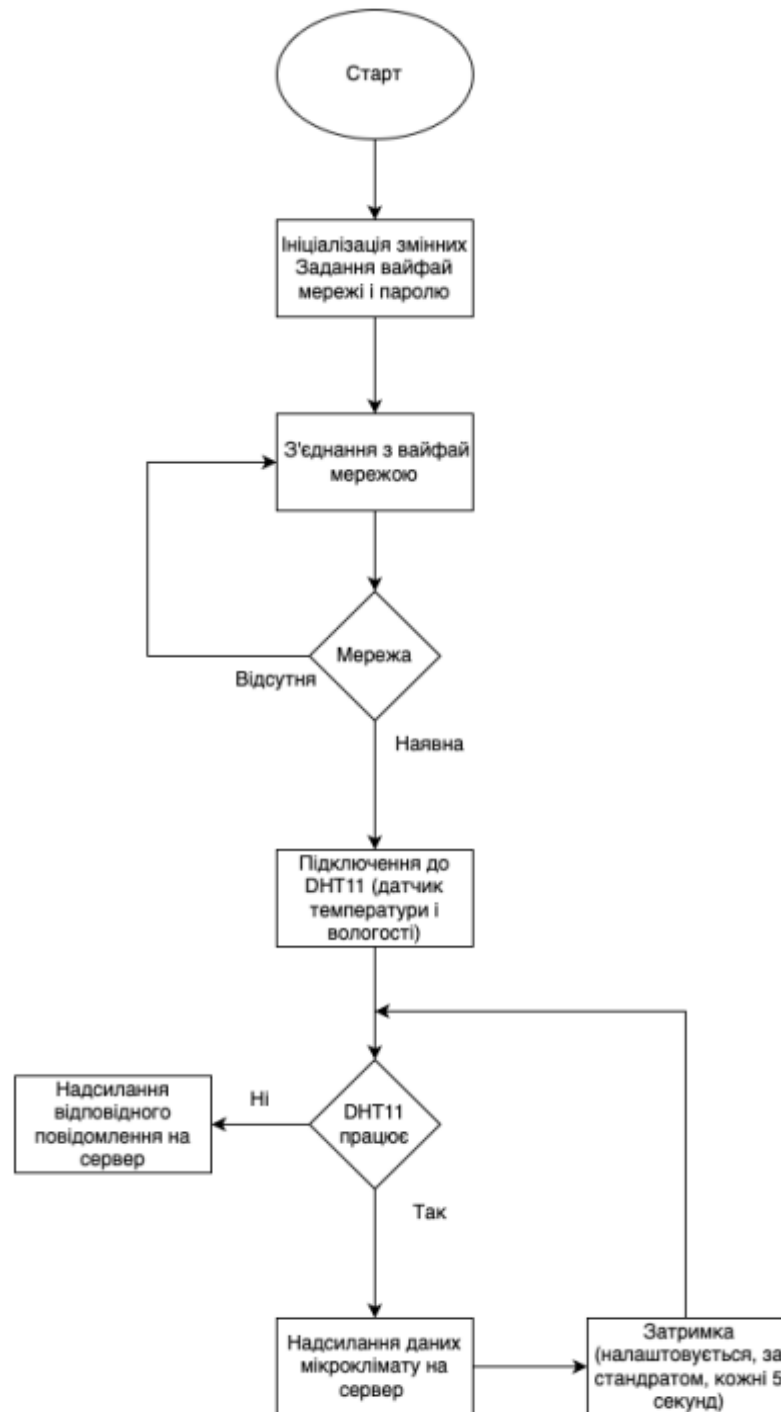


Рис. 3.12. Діаграма алгоритму роботи модуля для контролю мікроклімату

### 3.3.2.3. Модуль освітленості

Завдання модуля освітленості полягає в автоматичному керуванні освітленням серверного приміщення шляхом його ввімкнення при виявленні присутності персоналу та вимкнення після того, як приміщення буде залишено.

Для реалізації цього завдання було застосовано наступні компоненти:

Інфрачервоний датчик руху.

Лампа розжарювання (як демонстраційне навантаження).

Модуль з двома реле для комутації електричного кола освітлення.

З'єднувальні проводи для макетних плат (типу "мама-тато").

Конструкція датчика передбачає наявність двох змінних резисторів, один з яких використовується для регулювання чутливості сенсора, а інший визначає тривалість періоду активності після фіксації руху.

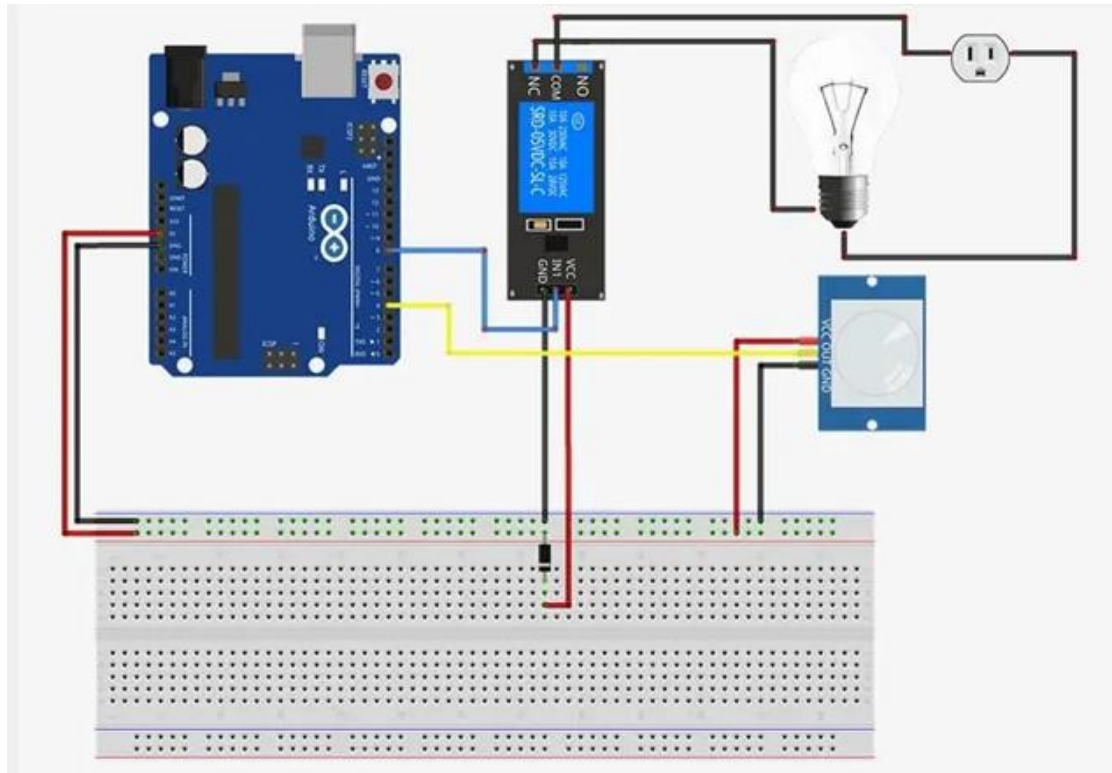


Рис. 3.13. Схема підключення

Логіка роботи модуля освітленості показана на рисунку 3.14.

```
void PIRsensor() { // Функція зчитування сигналів з датчика руху PIR
    //Serial.println(analogRead(PIN_PIR));
    //зчитуємо стан датчика якщо сигнал вище 300 означає піднімаємо прапор statePIR,
    інакше опускаємо statePIR
    if (analogRead(PIN_PIR)>300){statePIR = HIGH;}
    else {statePIR = LOW;}
    //якщо є рух включити Освітлення інакше вимкнути Освітлення
    if (statePIR == HIGH) {digitalWrite(PIN_RELE_SVET, HIGH);}
    else {digitalWrite(PIN_RELE_SVET, LOW);}
}
```

Рис. 3.14. Робота модуля освітленості

Блок-схема модуля освітленості зображена на рисунку 3.15.

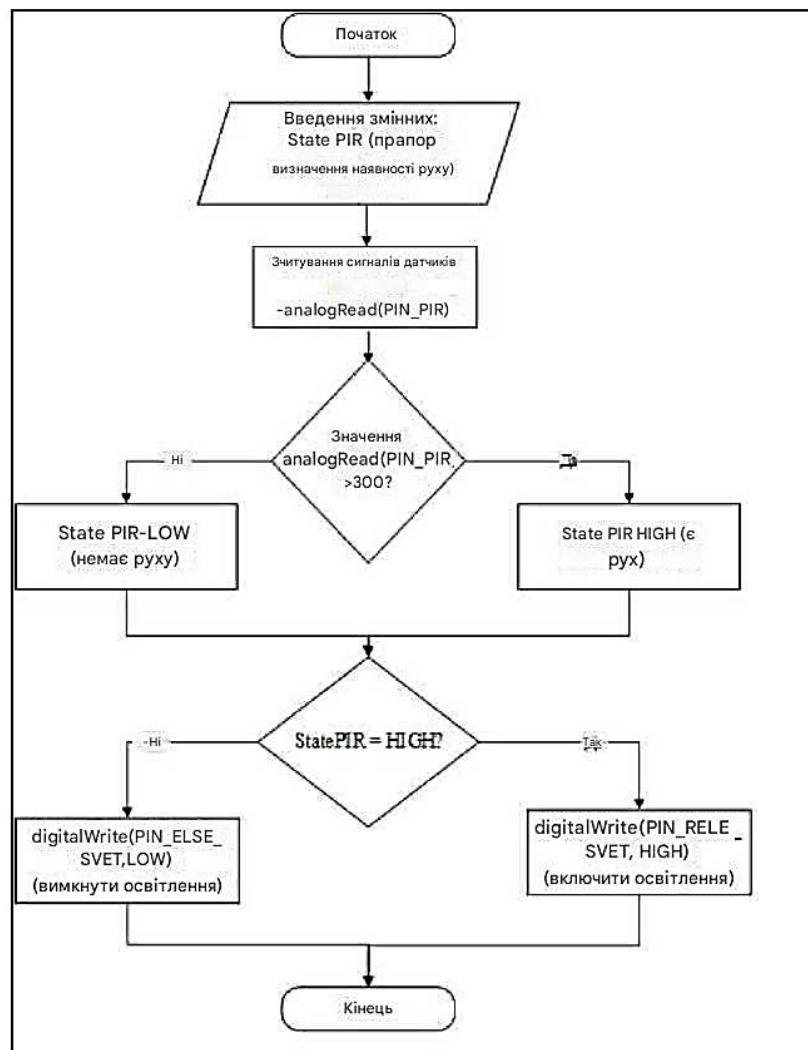


Рис. 3.15. Блок-схема модуля освітленості

#### 3.3.2.4. Модуль контролю доступу

Завдання модуля контролю доступу полягає у здійсненні реєстрації осіб, які входять до приміщення.

Для реалізації цієї функції застосовано технологію радіочастотної ідентифікації (RFID). RFID являє собою метод безконтактної передачі, запису та зберігання даних за допомогою радіосигналів.

Кожна RFID-система складається зі зчитувача (ридера) та RFID-мітки, що містить ідентифікаційні дані. Мітка включає інтегральну схему для зберігання та обробки інформації, а також антену для прийому та передачі радіосигналів.

Для реалізації даного модуля було використано наступні компоненти:

- RFID-зчитувач MFRC-522.
- RFID-мітка.

- З'єднувальні проводи для макетних плат (типу "мама-тато").

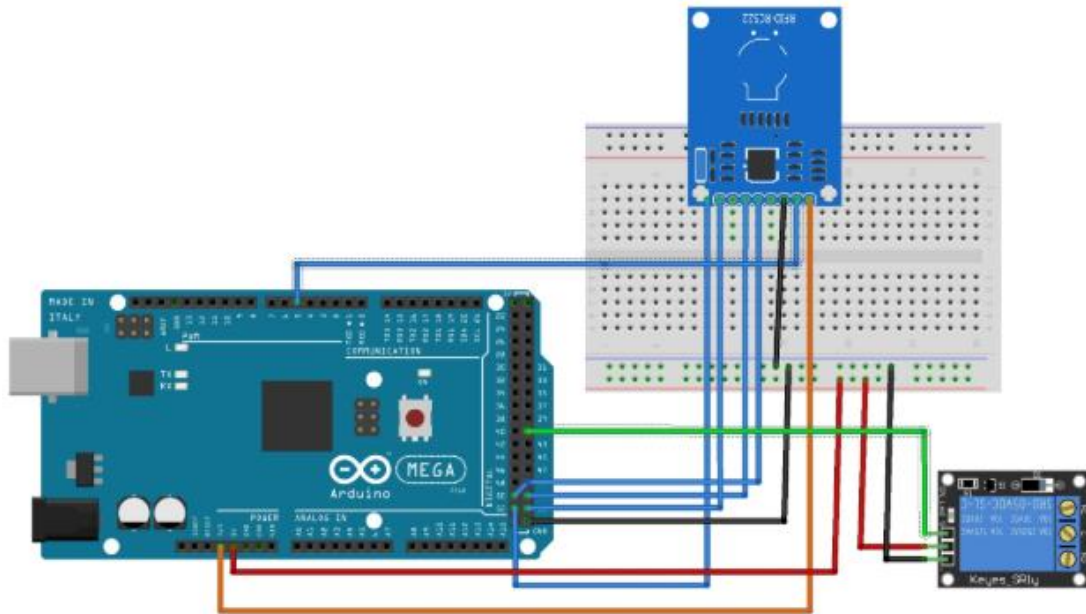


Рис. 3.16. Схема підключення RFID-зчитувача MFRC-522

Логіка функціонування RFID-зчитувача є досить складною, тому її повний опис наведено в Додатку Е. Основна структурна схема алгоритму його роботи така:

```
// Ініціалізація лічильника RFID міток
недійсний setupMFRC522() {
    SPI.begin();          // Ініціалізуємо інтерфейс SPI
    mfrc522.PCD_Init();    // Ініціалізуємо модуль MFRC522
    mfrc522.PCD_DumpVersionToSerial(); // Виводимо версію прошивки модуля на
    монітор серійного порта
    // Виводимо повідомлення про те, що модуль готовий до розрахунку і очікує мітку
    Serial.println(F("Будь ласка, розмістіть RFID-мітку на зчитувачі"));
}
// Читання карт і ключів з RFID-мітками
// Логіка така: провели ключем відкритий замок, щоб закрити — потрібно провести
ключем ще раз
void keyRFID() {
    digitalWrite(PIN_RELE_LOCK, StateKeyRFID); // Управляємо реле в залежності
    від стану замка
    Blynk.virtualWrite(VPIN_LOCK, StateKeyRFID); // Ішлемо дані у таблиці Blynk
}
```

```
// Очікуємо мітку, і поки її немає, перериваємо подальше виконання цієї функції
(ключ RFID)
якщо (!mfrc522.PICC_IsNewCardPresent()) {
    повернення;
}
// Спроба прочитати мітку, і поки це не вийшло, перериваєм подальше виконання цієї
функції (keyRFID)
якщо (!mfrc522.PICC_ReadCardSerial()) {
    повернення;
}
```

### 3.3.2.5. Модуль стаціонарної візуалізації

Завдання модуля стаціонарної візуалізації полягає у відображенні параметрів датчиків, встановлених у серверному приміщенні, на рідкокристалічному дисплеї (LCD).

Інтерфейс LCD-екрана оснащено меню, навігація по якому здійснюється за допомогою кнопок, забезпечуючи окремий перегляд кожного параметра. В автономному режимі передбачено автоматичне циклічне перемикання пунктів меню.

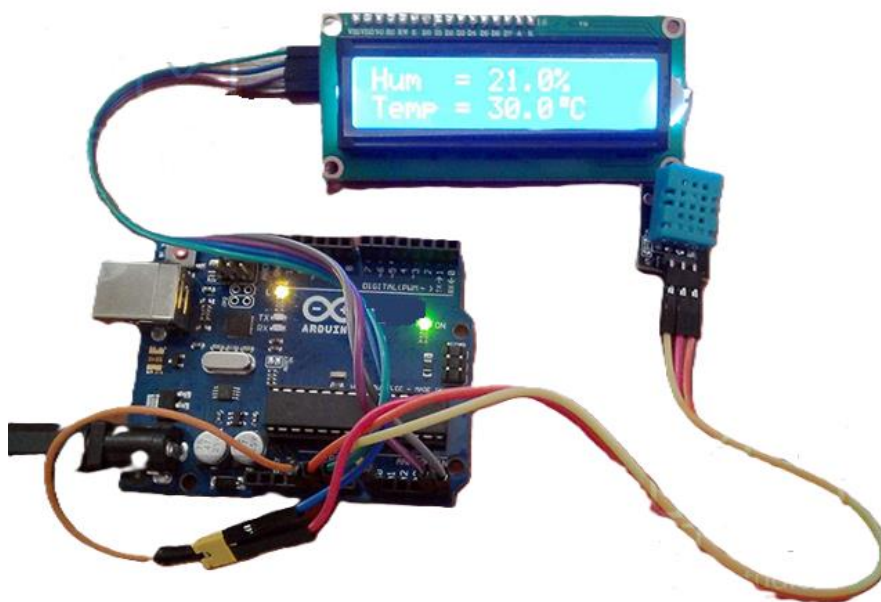


Рис. 3.17. Схема підключення

Структура меню LCD-екрана включає наступні пункти:

Екран 1: Відображення даних, отриманих від датчика температури та вологості DHT22 (вологість, температура).

					ІТС.4КІ.0125.03-ПЗ	Арк.
						94
Змн.	Арк.	№ докум.	Підпис	Дата		

Екран 2: Відображення стану освітлення (увімкнено/вимкнено).

Екран 3: Відображення стану системи контролю доступу на основі RFID (двері відчинено/зачинено).

Для реалізації даного модуля було використано наступні компоненти:

- LCD-дисплей.
- Перетворювач інтерфейсу LCD в I2C для спрощення підключення до мікроконтролера.
- З'єднувальні проводи для макетних плат (типу "мама-тато").

Логіка роботи модуля стаціонарної візуалізації подана нижче.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h> // Бібліотека для LCD екрана
LiquidCrystal_I2C lcd(0x27,16,2); // Оголошуємо екран за адресою 0x27 на шині I2C // 2
рядки по 16 символів

// Прототип функції, щоб компілятор знав, що така функція існує
void LCDmenu();

// Функція перемикавання екранів
void ScreenSwitching(){
    MenuPos++;
    if(MenuPos > 2){MenuPos=0;}
    LCDmenu();
}

// Функція виводу інформації на LCD екран
void LCDmenu(){
    static int8_t LastMenuPos = 100;
    switch (MenuPos) {
        case 0:
            // Виконується, коли MenuPos дорівнює 0
            if (LastMenuPos != MenuPos) {
                // Якщо минулого разу був інший екран, очищуємо і виводимо новий
                lcd.clear();
                lcd.setCursor(0, 0);    // Встановлюємо курсор на початок 1 рядка
                lcd.print(F("Hum = %")); // Виводимо текст
                lcd.setCursor(7, 0);    // Встановлюємо курсор на 7 символ
                lcd.print(h, 1);        // Виводимо значення вологості
```

```

        lcd.setCursor(0, 1);    // Встановлюємо курсор на 2 рядок
        lcd.print(F("Temp = \1C")); // Виводимо текст, \1 – символ градуса
        lcd.setCursor(7, 1);    // Встановлюємо курсор на 7 символ
        lcd.print(t, 1);        // Виводимо значення температури
    } else {
        // Інакше просто оновлюємо інформацію в потрібних місцях
        lcd.setCursor(7, 0);
        lcd.print(h, 1);
        lcd.setCursor(7, 1);
        lcd.print(t, 1);
    }
    break;
case 1:
    // Виконується, коли MenuPos дорівнює 1
    lcd.clear();
    lcd.setCursor(0, 0);
    if(statePIR == HIGH) {
        lcd.print(F("SVET VKL")); // Світло ввімкнене
    } else {
        lcd.print(F("SVET OTKL")); // Світло вимкнене
    }
    break;
case 2:
    // Виконується, коли MenuPos дорівнює 2
    lcd.clear();
    lcd.setCursor(0, 0);
    if(!StateKeyRFID) {
        lcd.print(F("CLOSED")); // Закрито
    } else if (StateKeyRFID) {
        lcd.print(F("OPEN")); // Відкрито
    }
    break;
default:
    break;
}
}

```

### 3.3.2.6. Модуль дистанційної візуалізації

Завдання модуля дистанційної візуалізації полягає у відображенні основних параметрів серверного приміщення на веб-сторінці та в мобільному застосунку "Blynk".

Для реалізації цього завдання використовується модуль Ethernet Shield W5100 та мережевий кабель.

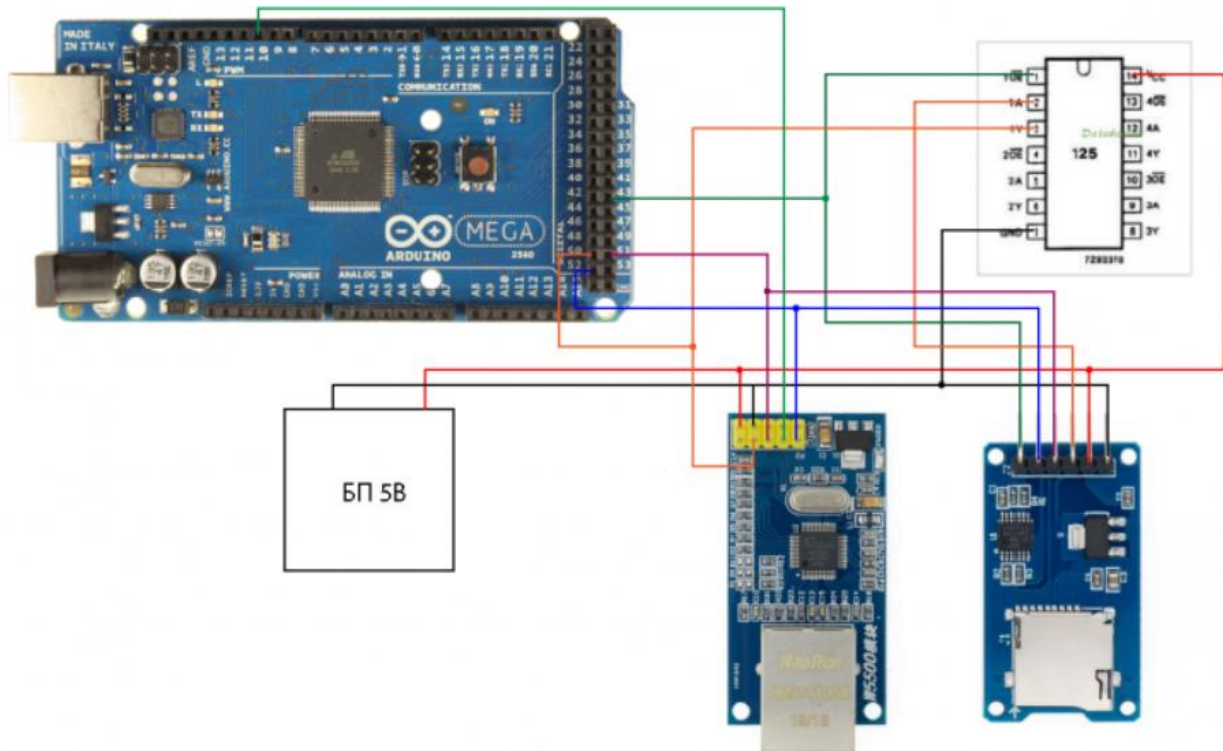


Рис. 3.18. Схема підключення модуля Ethernet Shield W5100

Ethernet Shield забезпечує можливість підключення Arduino до мережі Інтернет. Цей модуль розширює функціональні можливості Arduino, дозволяючи передавати та отримувати дані з будь-якої точки світу, де є доступ до Інтернету.

Веб-сервер ініціалізується на Ethernet Shield за допомогою команди `Ethernet.begin(arduino_mac, device_ip, dns_ip, gateway_ip, subnet_mask)`.

Де:

- `arduino_mac` – MAC-адреса плати Arduino.
- `device_ip` – IP-адреса плати Arduino в локальній мережі.
- `dns_ip` – IP-адреса DNS-сервера.
- `gateway_ip` – IP-адреса шлюзу IPv4.
- `subnet_mask` – маска підмережі IPv4.

Веб-сторінка розробляється у середовищі програмування Arduino IDE у вигляді коду HTML.

Логіка формування веб-сторінки з температурою, вологістю та станом освітлення, яка оновлюється кожні 5 секунд:

```
// Виведення HTML-сторінки
if (c == '\n' && currentLineIsBlank) {
  client.println(F("HTTP/1.1 200 OK"));           // Заголовочна інформація
  client.println(F("Content-Type: text/html"));   // Тип контенту
  client.println(F("Connection: close"));         // Закрити з'єднання
  client.println(F("Refresh: 5"));                // Автоматичне оновлення кожні 5 сек
  client.println();
  client.println(F("<!DOCTYPE HTML>"));           // HTML тип документа
  client.println(F("<html>"));                     // Відкриття тега HTML
  client.println(F("<head>"));                     // Відкриття тега HEAD
  client.println(F("<meta http-equiv='Content-Type' content='text/html; charset=utf-8'/>"));
  client.println(F("<title>Мікрокомплекс</title>")); // Назва сторінки
  client.println(F("</head>"));                     // Закриття HEAD
  client.println(F("<body>"));                     // Відкриття BODY
  client.println(F("<h1>Мікрокомплекс</h1>"));       // Заголовок сторінки
  client.println(F("<br>"));                         // Перенесення на новий рядок
  client.println(F("<hr>"));                         // Розділювальна лінія
  client.println(F("Температура = "));           // Температура з DHT 11
```

```

client.println(t); // Змінна температури
client.println(F(" °C"));
if (t_alarm_max) {
    client.println(F("Увага! Перевищення норми температури"));
} else if (t_alarm_min) {
    client.println(F("Увага! Температура нижче норми"));
}
client.println(F("<br>")); // Перенесення на новий рядок
client.println(F("Вологість = ")); // Вологість з DHT 11
client.println(h); // Змінна вологості
client.println(F(" %"));
if (h_alarm_max) {
    client.println(F("Увага! Перевищення норми вологості"));
} else if (h_alarm_min) {
    client.println(F("Увага! Вологість нижче норми"));
}
client.println(F("<br>")); // Перенесення на новий рядок
client.println(F("<hr>")); // Розділювальна лінія
if (statePIR == HIGH) {
    client.println(F("Світло в приміщенні увімкнено"));
} else if (statePIR == LOW) {
    client.println(F("Світло в приміщенні вимкнено"));
}
client.println(F("<br>"));
client.println(F("<hr>"));
}

```

### 3.3.3. Реалізація програмної частини мобільного застосунку

Для того, щоб IoT-рішення функціонувало за межами локальної Wi-Fi мережі, необхідно залучити IoT-платформу. **IoT-платформа** — це інтегроване рішення, що забезпечує уніфіковану взаємодію між кінцевими пристроями Інтернету речей та сервісами, які обробляють зібрані дані.

Важливість таких платформ можна пояснити, звернувшись до статистики. Дослідження Cisco показало, що 75% IoT-проектів зазнають невдачі. В опитуванні брали участь понад 1800 керівників компаній та IT-лідерів, метою

					ІТС.4КІ.0125.03-ПЗ	Арк.
						99
Змн.	Арк.	№ докум.	Підпис	Дата		

якого було виявлення основних перешкод для впровадження Інтернету речей на підприємствах. Згідно з висновками, ключовими бар'єрами є витрати та терміни реалізації проєктів, а також обмеженість експертних знань штатних співробітників.

Саме IoT-платформи допомагають усунути ці проблеми, забезпечуючи уніфіковану взаємодію між пристроями IoT та сервісами обробки даних. Наприклад, якщо компанія вже має значний парк обладнання, його інтеграція в нову IoT-інфраструктуру може бути складною. Деякі "старі" пристрої можуть успішно виконувати свої функції, але не мати можливості підключення до Інтернету. Заміна такого обладнання на IoT-сумісне потягне за собою значні витрати, що збільшить термін окупності проєкту, адже доведеться списувати цілком працездатні верстати та агрегати.

Навіть якщо обладнання сумісне з IoT, залишається низка важливих питань: які дані слід збирати й використовувати, як проводити їхній поглиблений аналіз та забезпечити оперативний зворотний зв'язок. Саме IoT-платформи вирішують ці завдання, забезпечуючи безшовну інтеграцію апаратних засобів з використанням різних типів підключень, а також ефективну передачу даних на підключені пристрої або між ними.

Сучасний ринок пропонує безліч IoT-платформ від провідних високотехнологічних та IT-компаній. Приміром, компанія Toshiba розробила платформу SPINEX для інтеграції IoT-пристроїв і сервісів. Її створення ґрунтувалося на значному досвіді Toshiba в енергетиці, виробництві напівпровідникових компонентів, а також у сферах Інтернету речей, штучного інтелекту, розпізнавання голосу та відео.

Програмування мікроконтролерних платформ, таких як Arduino або Raspberry Pi, стало одним із найцікавіших напрямків розвитку двадцять першого століття. Побудова пристроїв на їхній основі давно вийшла за межі можливостей лише вузьких фахівців. Вони застосовуються для створення верстатів, роботів, 3D-принтерів, квадрокоптерів, серверів та різноманітних IoT-пристроїв.

Однак, цей ринок є значно сегментованим: платформи програмуються в різних середовищах та через різні інтерфейси. Проєкт Blynk може стати вирішенням цієї проблеми, об'єднуючи підходи до програмування.

Blynk є хмарним сервісом, що забезпечує інтуїтивно зрозуміле створення графічних інтерфейсів для управління мікрокомп'ютерами та мікроконтролерами. Ця платформа значно спрощує збір інформації з датчиків, мінімізуючи потребу в розробці складних програмних рішень.

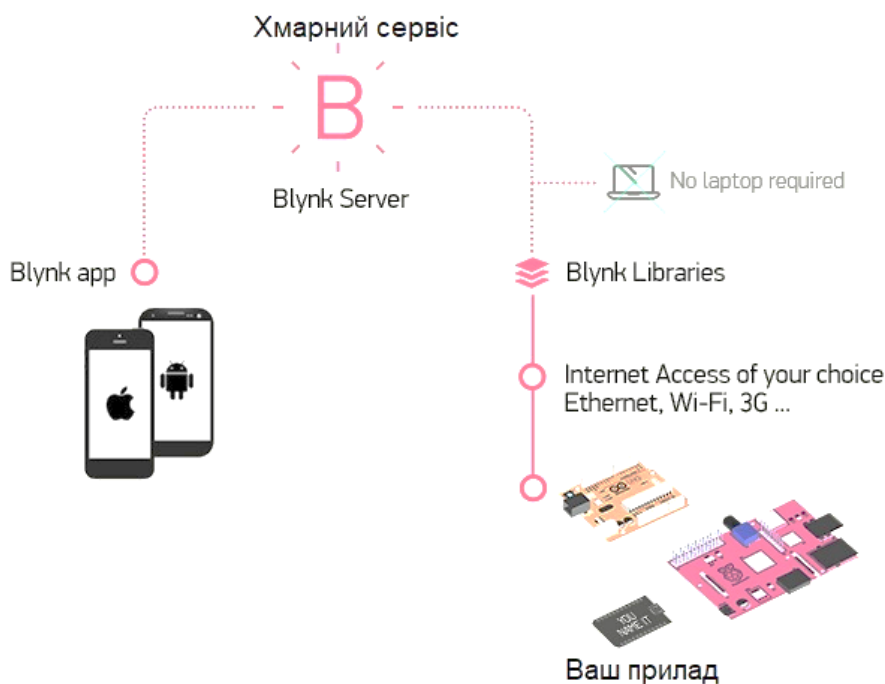


Рис. 3.19. Хмарний сервіс Blynk

Платформа складається з трьох ключових компонентів:

- Blynk App — це мобільний додаток, який дозволяє користувачам створювати інтуїтивно зрозумілі інтерфейси для своїх IoT-проєктів за допомогою різноманітних віджетів.
- Blynk Server — цей сервер відповідає за всю комунікацію між смартфоном і апаратним забезпеченням. Користувачі можуть обирати між використанням хмарного сервера Blynk або розгортанням власного приватного сервера локально, залежно від своїх потреб.
- Blynk Libraries — ці бібліотеки, доступні для всіх популярних апаратних платформ (Arduino, Raspberry Pi, ESP8266, Energia,

Particle та ін.), забезпечують зв'язок із сервером та обробляють всі вхідні й вихідні команди пристрою.

Для використання Blynk необхідно встановити мобільний додаток та пройти обов'язкову реєстрацію. Це пов'язано з хмарною архітектурою сервісу, що гарантує контрольований доступ до підключених пристроїв. Варто зазначити, що система передбачає можливість локального розгортання сервера, що виключає залежність від інтернет-з'єднання.

Для ініціалізації та управління програмами потрібні базові навички з'єднання мікроконтролера або мікрокомп'ютера з ПК. Blynk підтримує широкий спектр комунікаційних інтерфейсів, забезпечуючи гнучкість інтеграції:

- Official Ethernet Shield (W5100)
- Adafruit CC3000 WiFi
- ENC28J60
- RN-XV WiFly
- ESP8266
- USB (Serial)
- SeeedStudio Ethernet Shield V2.0 (W5200)
- Official Arduino WiFi Shield
- ESP8266 (WiFi modem)

Blynk позиціонується як універсальний інструмент для швидкої розробки застосунків управління мікроконтролерними та мікрокомп'ютерними проєктами — від збору даних з метеостанцій до автоматизації розумного будинку та робототехніки. Це робить його привабливим як для початківців, так і для досвідчених фахівців, що цінують оптимізацію часу.

Документація та інструкції доступні на офіційному сайті. Як open-source проєкт, Blynk заохочує активну участь спільноти у розробці нових можливостей.

На даний момент Blynk підтримує роботу з такими апаратними платформами:

- ESP8266
- Raspberry Pi

					ІТС.4КІ.0125.03-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		102

- Arduino
- Wicked WildFire (CC3000)
- Particle (exSpark Core)
- TinyDuino (CC3000)

Для автономної роботи з платформою Blynk необхідно розгорнути локальний сервер. Процес включає такі етапи:

1. *Отримання файлів сервера:* завантажте основний архів сервера та відповідний JAR-файл з офіційних репозиторіїв Blynk (<https://github.com/blynkkk/blynk-server/archive/master.zip> та <https://github.com/blynkkk/blynk-server/releases> відповідно).
2. *Підготовка робочого середовища:* розпакуйте архів у бажану папку та створіть у ній директорію data для зберігання службових даних.
3. *Запуск сервера:* відкрийте командний рядок, перейдіть до папки з розпакованими файлами та виконайте команду: `java -jar server/server-0.41.11-java8.jar -dataFolder data`.
4. *Верифікація роботи:* після запуску, для перевірки функціональності сервера, перейдіть у веб-браузері за адресою <https://10.30.2.33:9443/admin>. Відображення інтерфейсу (як на рис. 3.20) підтвердить успішний запуск.

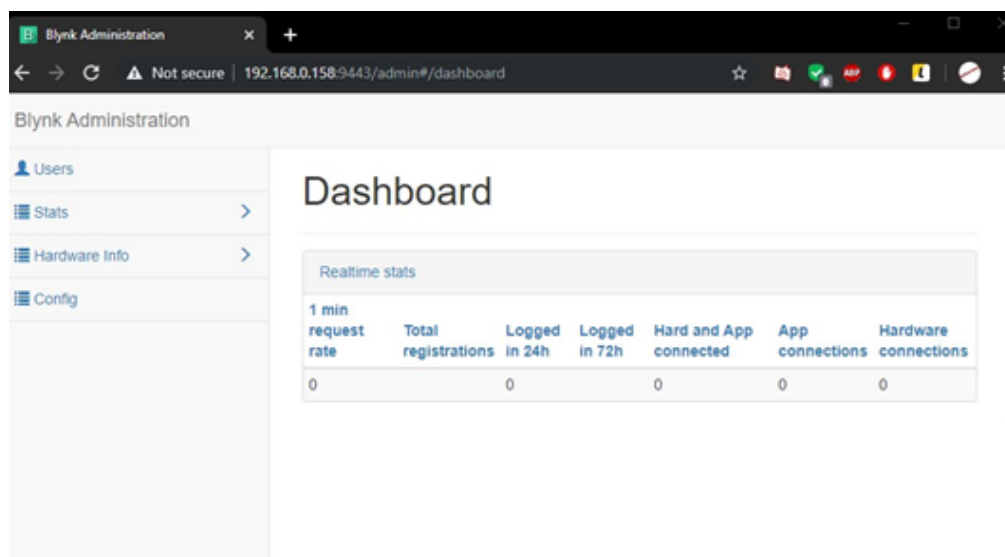


Рис. 3.20. «Blynk - сервер»

5. Далі додати до існуючої програми код для коректної роботи з Wi-Fi:

					ІТС.4КІ.0125.03-ПЗ	Арк.
						103
Змн.	Арк.	№ докум.	Підпис	Дата		

```
#include <ESP8266WiFi.h> const char* ssid = "TS_ITS";
const char* password = "KafTSITS";
void setup(void)
{
  Serial.begin(115200);          WiFi.begin(ssid,          password);
  Serial.print("Connecting WiFi AP: ");
  while (WiFi.status() != WL_CONNECTED) { delay(1000);
  Serial.print(".");
  }
  Serial.println("");          Serial.println("WiFi          connected");
  Serial.println(WiFi.localIP());
}
void loop() {
}
```

Для забезпечення взаємодії веб-додатку з пристроями використовуються наступні ключові програмні елементи:

*Blynk.begin("auth", "ssid", "pass")*: функція відповідає за налаштування мережевого підключення до Wi-Fi точки доступу (визначається SSID та паролем) та автентифікацію пристрою на Blynk-сервері за допомогою auth token.

Підключення до Blynk-сервера ініціюється після першого виконання *Blynk.run()* або *Blynk.connect()*. Для управління станом з'єднання (відключення/повторне підключення) застосовуються функції *Blynk.disconnect()* та *Blynk.connect()*, які повертають логічний результат.

*Blynk.run()*: є центральною функцією для постійної синхронізації пристрою з Blynk-сервером та прийому команд з мобільного додатку. Вона зазвичай інтегрується в головний цикл *loop()*. Однією з вагомих переваг цієї функції є можливість прямого керування фізичними входами та виходами плати ESP (у режимі "Standalone") без потреби писати явні команди *digitalRead*, *digitalWrite*, *analogRead*, *analogWrite* у коді. Достатньо просто налаштувати відповідні елементи управління в мобільному додатку та пов'язати їх з необхідними контактами ESP8266.

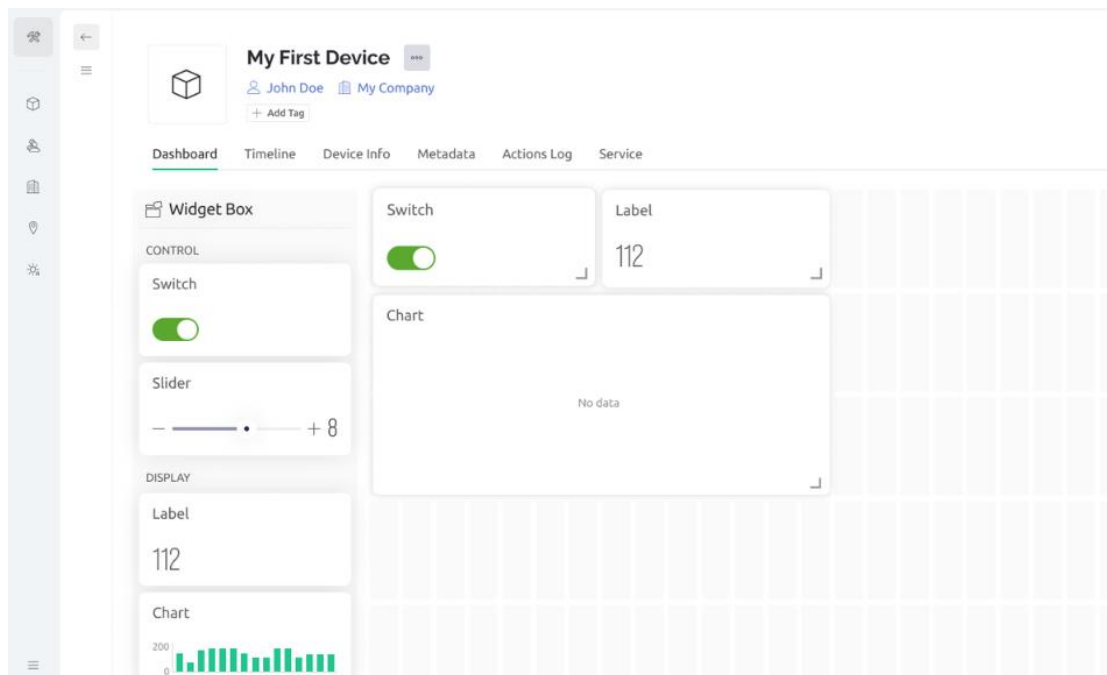


Рис. 3.21. Сторінка веб-додатку для керування системою моніторингу мікроклімату серверної кімнати

Хоча веб-додаток пропонує велику кількість віджетів та керуючих блоків, для наших цілей достатньо таких елементів:

- Чотири світлові індикатори "LED" (Virtual Pins 1, 29, 30, 31).
- Дві клавіші (Virtual Pins 2, 3).
- Термінал для обміну інформацією (Virtual Pin 4).
- "Value Display" для відображення даних з датчика (Virtual Pin 8).
- "History Graph" для відображення історичних даних на графіку (Virtual Pin 8).

В якості середовища розробки було обрано **Blynk.Apps** (Рисунок 3.22). Ця платформа надає нативні мобільні додатки для операційних систем **iOS** та **Android**, що забезпечують дистанційне керування підключеними пристроями та візуалізацію отриманих від них даних.



Рис. 3.22. Blynk.Apps

Функціонування додатку реалізується у двох основних режимах:

### 1. Режим розробника

Основне призначення режиму розробника полягає у створенні та редагуванні **графічного інтерфейсу користувача (GUI)** мобільної інформаційної панелі, що призначена для конкретного шаблону пристрою. Мобільний дашборд формується за допомогою **віджетів** – модульних елементів інтерфейсу, які можуть бути розміщені на робочому полотні. Кожен віджет виконує специфічну функцію (наприклад, кнопка, повзунок, графік) та має індивідуальні налаштування, що залежать від його функціональності.

### 2. Режим кінцевого користувача

Цей режим призначений як для виробників, так і для кінцевих користувачів. Його фокус зосереджений на перегляді та керуванні пристроями, налаштуванні автоматизації та отриманні сповіщень. Взаємодія відбувається через віджети та додаткові екрани, що містять актуальну інформацію про дані, які встановлюються, надсилаються або отримуються з **Blynk.Cloud** та безпосередньо з пристроїв.

Код для коректної роботи з Blynk:

```
#define BLYNK_PRINT Serial // Enables Serial Monitor
/* Fill in information from Blynk Device Info here */
// #define BLYNK_TEMPLATE_ID      "TMPxxxxxx"
// #define BLYNK_TEMPLATE_NAME    "Device"
// #define BLYNK_AUTH_TOKEN       "YourAuthToken"
```

					ITC.4KI.0125.03-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		106

```

// Following includes are for Arduino Ethernet Shield (W5100)
// If you're using another shield, see Boards_* examples
#include <SPI.h>
#include <Ethernet.h>
#include <BlynkSimpleEthernet.h>
void setup()
{
  // See the connection status in Serial Monitor
  Serial.begin(9600);
  // Here your Arduino connects to the Blynk Cloud.
  Blynk.begin(BLYNK_AUTH_TOKEN);
}
void loop()
{
  // All the Blynk Magic happens here...
  Blynk.run();
}

```



Рис 3.23. Інтерфейс додатку Blynk

### Висновки до розділу

Підсумовуючи результати даного розділу, слід зазначити створення функціональної системи моніторингу мікроклімату серверної кімнати на базі платформи Arduino, вирішенню задачі прототипування системи моніторингу. В рамках цього етапу було розроблено структурну схему пристрою, здійснено обґрунтований вибір основного обладнання, проведено аналіз функціональних вимог з використанням діаграм варіантів використання, а також побудовано

					ІТС.4КІ.0125.03-ПЗ	Арк.
						107
Змн.	Арк.	№ докум.	Підпис	Дата		

моделі, що описують різні аспекти системи, зокрема діаграму станів програмно-апаратного комплексу, діаграму компонентів та діаграму розгортання.

Описана логіка функціонування деяких програмних модулів системи моніторингу мікроклімату серверної кімнати на базі платформи Arduino, логіка функціонування інших програмних модулів детально описана у відповідних додатках:

- Логіка ініціалізації компонентів системи – Додаток А.
- Логіка обробки натискання кнопки перемикання меню LCD-екрана – Додаток Б.
- Логіка роботи з мобільним застосунком Blynk – Додаток В.
- Логіка роботи з SD-картою – Додаток Г.
- Логіка роботи LCD-дисплея – Додаток Д.
- Логіка роботи RFID-зчитувача MFRC522 – Додаток Е.
- Логіка роботи годинника реального часу – Додаток Є.
- Логіка роботи з датчиком температури та вологості DHT – Додаток Ж.
- Логіка взаємодії з сервером – Додаток З.
- Логіка роботи ІЧ-передавача – Додаток И.

Розроблено мобільний застосунок "Blynk" для системи дистанційного моніторингу мікроклімату серверної кімнати.

## ВИСНОВКИ

Дана робота присвячена розробці системи моніторингу мікроклімату серверної кімнати на базі Arduino, тому що забезпечення оптимальних параметрів повітря у кімнаті є актуальним завданням, оскільки при роботі обладнання серверної кімнати виділяється значна кількість тепла. Крім того, недотримання рекомендованих нормативів, щодо температури та вологості повітря може призвести до виходу з ладу обладнання, що в свою чергу, не лише зупинить роботу підприємства, але і призведе до значних матеріальних збитків.

У результаті виконання бакалаврської роботи було досягнуто поставленої мети, а саме забезпечення моніторингу та контролю стану мікроклімату серверного приміщення для підприємства, що використовує інформаційні технології у своїй діяльності.

Для досягнення визначеної мети було успішно вирішено наступні завдання:

- Проведено огляд існуючих систем моніторингу, що дозволило виявити їхні переваги, недоліки та визначити актуальні тенденції в даній галузі.
- Здійснено детальне виявлення та формалізацію вимог до проєктованої системи моніторингу на основі потреб типового підприємства з серверною інфраструктурою.
- Розроблено архітектуру інформаційної системи, включаючи визначення її функціональних модулів, інтерфейсів та принципів взаємодії.
- Проведено детальний аналіз основних компонентів та інструментів, необхідних для реалізації системи моніторингу мікроклімату серверної кімнати на базі платформи Arduino.
- Розроблено структурну схему пристрою, описано логіку функціонування програмних модулів системи моніторингу мікроклімату серверної кімнати на базі платформи Arduino.

					ІТС.4КІ.0125.03-ПЗ				
Змн.	Арк.	№ докум.	Підпис	Дата					
Розроб.	Ліщук С.Е.				ВИСНОВКИ		Літ.	Арк.	Акрушів
Керівник	Донченко В.Ю.							109	3
Реценз.	Козуб Ю.Г.						ЛНУ Кафедра ІТС, Гр.4КІ		
Н. Контр.									
Зав. каф.	Семенов М.А.								

- Створено прототип системи моніторингу та здійснено його попереднє впровадження для демонстрації працездатності та відповідності типовим вимогам.
- Розроблено мобільний застосунок "Blynk" для системи дистанційного моніторингу мікроклімату серверної кімнати.

Процес розробки продемонстрував, що вибір використаного обладнання був оптимальним та повністю виправдав себе, забезпечивши можливість створення повноцінної системи моніторингу, яка відповідає визначеним вимогам.

Розроблена система моніторингу може бути впроваджена на підприємствах, що мають власну серверну інфраструктуру, про що свідчать результати її прототипування. Це підтверджує практичну цінність та потенціал розробленої системи для забезпечення надійного функціонування критично важливого обладнання.

Подальший розвиток розробленої системи моніторингу може бути спрямований на розширення її функціональних можливостей та підвищення ефективності. Можливими напрямками вдосконалення є:

- Інтеграція з платформами хмарних сервісів: забезпечення можливості централізованого моніторингу розподіленої інфраструктури.
- Впровадження розширеної системи оповіщень: налаштування гнучких сценаріїв сповіщень через різні канали зв'язку (SMS, електронна пошта, месенджери).
- Розробка інструментів візуалізації даних: створення інтерактивних графіків та дашбордів для зручного аналізу історичних даних та поточного стану системи.
- Підтримка протоколів віддаленого керування: реалізація можливості віддаленого перезавантаження обладнання або виконання інших базових адміністративних дій.

					ІТС.4КІ.0125.03-ПЗ	Арк.
						110
Змн.	Арк.	№ докум.	Підпис	Дата		

- Адаптація до різних типів сенсорів та обладнання: забезпечення гнучкості системи для інтеграції з ширшим спектром пристроїв.
- Розробка API для інтеграції зі сторонніми системами: надання можливості іншим програмним продуктам отримувати дані моніторингу.

Реалізація зазначених напрямків розвитку дозволить значно підвищити функціональність, адаптивність та комерційну привабливість розробленої системи моніторингу для широкого кола підприємств.

					ІТС.4КІ.0125.03-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		111

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Вимоги до інженерної інфраструктури технологічних приміщень (серверних) для розміщення серверного та телекомунікаційного обладнання ВЦТМ : наказ Державної служби України з надзвичайних ситуацій № 608 від 23.10.2019 [Електронний ресурс]. – Режим доступу: <https://dsns.gov.ua/nakaziz-osnovnoyi-diynosti/100411>.
2. PRTG: Моніторинг серверної кімнати [Електронний ресурс]. – Режим доступу: <https://idealsoft.com.ua/vendors/paessler/paessler-kachestvo-uslug-oborudovanie/monitoring-serverroom/>.
3. Requirements for a flexible and scalable Automatic Server Room Environmental Conditions Monitoring and Control system [Електронний ресурс]. – Режим доступу: <https://zenodo.org/records/1284899>.
4. AVTECH – Monitor Temperature and Environment Conditions with Room Alert [Електронний ресурс]. – Режим доступу: <https://avtech.com>.
5. Муратов В. Г., Ананський Д. В. Спосіб автоматичного регулювання параметрів мікроклімату в приміщенні : пат. 112127 Україна ; заявл. 2016.
6. Зубенко О. В., Бойко О. В., Небилиця М. С. Спосіб автоматизованого управління мікрокліматом в приміщенні в режимі циркадного ритму : пат. 101943 Україна ; заявл. 2015.
7. Гараган Н. Ю., Мазур О. В. Спосіб автоматичного управління мікрокліматом в приміщенні : пат. 72565 Україна ; заявл. 2012.
8. ANSI/TIA-942-2005 Telecommunications Infrastructure Standard for Data Centers [Електронний ресурс]. – Режим доступу: <https://ansi.org>.
9. Dutt D. G. EVPN in the Data Center. O'Reilly Media, 2018. – 106 p.
10. Комп'ютерна система контролю і керування мікрокліматом робочого місця : пат. 23908 Україна ; заявл. 05.02.2007 ; опубл. 11.06.2007. – 3 с. – Режим доступу:

					<b>ІТС.4КІ.0125.03-ПЗ</b>		
<b>Змн.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підпис</b>	<b>Дата</b>	<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ</b>		
Розроб.		Ліщук С.Е.					
Керівник		Донченко В.Ю.					
Реценз.		Козуб Ю.Г.					
Н. Контр.							
Зав. каф.		Семенов М.А.			ЛНУ Кафедра ІТС, Гр.4КІ		
					<b>Лім.</b>	<b>Арк.</b>	<b>Акрушів</b>
						112	4

<https://base.uipv.org/searchINV/search.php?action=viewdetails&IdClaim=106749&chapter=biblio>.

11. Пристрій для визначення показників мікроклімату та вмісту шкідливих речовин в атмосферному повітрі : пат. 108466 Україна ; заявл. 17.06.2015 ; опубл. 25.07.2016. – Режим доступу: <https://base.uipv.org/searchINV/search.php?action=viewdetails&IdClaim=225473&chapter=description>.
12. Пристрій для вимірювання параметрів мікроклімату приміщень : пат. 99874 Україна ; заявл. 25.06.2015 ; опубл. 25.06.2015. – Режим доступу: <https://base.uipv.org/searchINV/search.php?action=viewdetails&IdClaim=213682&chapter=description>.
13. Спосіб автоматичного управління мікрокліматом в приміщенні : пат. 72565 Україна ; заявл. 16.01.2012 ; опубл. 27.08.2012. – Режим доступу: <https://base.uipv.org/searchINV/search.php?action=viewdetails&IdClaim=176991&chapter=description>.
14. Електронний аналізатор параметрів мікроклімату приміщення для системи "розумний дім" : пат. 146633 Україна ; заявл. 24.04.2019 ; опубл. 10.03.2021. – Режим доступу: <https://base.uipv.org/searchINV/search.php?action=viewdetails&IdClaim=274793&chapter=description>.
15. Спосіб моделювання та оптимізації режимів моніторингу мікроклімату, зокрема, в приміщеннях музеїв або художніх галерей : пат. 97047 Україна ; заявл. 29.10.2010 ; опубл. 26.12.2011. – Режим доступу: <https://base.uipv.org/searchINV/search.php?action=viewdetails&IdClaim=167781&chapter=description>.
16. Заєць Я. Б., Злотенко Б. М. Енергоефективна система керування мікрокліматом у приміщенні // Технології та дизайн. Мехатронні системи та комп'ютерні технології. – 2017. – № 1 (22). – С. 1–6.
17. Şuvar M.-C., Munteanu L., Cioară C. Optimal Monitoring of Server Rooms with Home Assistant Platform // MATEC Web of Conferences. – 2022. – Vol. 373. – P. 00044. – DOI: 10.1051/matecconf/202237300044.

					ІТС.4КІ.0125.03-ПЗ	Арк.
						113
Змн.	Арк.	№ докум.	Підпис	Дата		

- 18.Моніторинг серверної кімнати [Електронний ресурс]. – Режим доступу:  
<https://klaster.ua/ua/reshenija/dispatcherizacija/byudzhethnyy-variant/monitoringservernoy-komnaty/>.
- 19.Система SNMP моніторингу серверної кімнати [Електронний ресурс]. – Режим доступу: <https://kmm.md/ro/content/система-snmp-мониторинга>.
- 20.The Best Server Room Environmental Monitoring Systems & Tools for 2023 [Електронний ресурс]. – Режим доступу:  
<https://www.comparitech.com/netadmin/server-room-environmental-monitoring-systems/>.
- 21.Innovators in Network Monitoring [Електронний ресурс]. – Режим доступу:  
<https://netmon.com/>.
- 22.TempCube [Електронний ресурс]. – Режим доступу: <https://tempcube.io/>.
- 23.Monnit [Електронний ресурс]. – Режим доступу:  
<https://www.monnit.com/applications/data-center-server-room-monitoring/>.
- 24.Vertiv Geist™ Екологічний моніторинг [Електронний ресурс]. – Режим доступу: <https://www.vertiv.com/en-us/products/brands/geist/vertiv-geist-environmental-monitoring/>.
- 25.Standard 90.1-2022—Energy Standard for Sites and Buildings Except Low-Rise Residential Buildings [Електронний ресурс]. – Режим доступу:  
<https://www.ashrae.org/technical-resources/bookstore/standard-90-1>.
- 26.Моніторинг серверної кімнати і ЦОД. Рекомендовані стандарти та найкращі практики [Електронний ресурс]. – Режим доступу:  
[https://netping.com.ua/monitoring\\_servernoi\\_kimnaty\\_i\\_cod.html](https://netping.com.ua/monitoring_servernoi_kimnaty_i_cod.html).
- 27.Chen D., Cho K., Han S., Jin Z., Shin K. Invisible Sensing of Vehicle Steering with Smartphones / D. Chen, K. Cho, S. Han, Z. Jin, K. Shin // Department of Electrical Engineering and Computer Science, University of Michigan, 2015. – P. 13.
- 28.Schwartz M. Internet of Things with ESP8266. – Packt Publishing Ltd, 2016. – ISBN 1786466678.

29. Brown E. Web Development with Node and Express: Leveraging the JavaScript Stack. – O'Reilly Media, 2014. – ISBN 1491902302.
30. Chodorow K. MongoDB: The Definitive Guide. 2nd ed. – O'Reilly, 2013. – ISBN 1449344682.

					<i>ITC.4KI.0125.03-ПЗ</i>	Арк.
						115
Змн.	Арк.	№ докум.	Підпис	Дата		

## ДОДАТОК А

### Ініціалізація компонентів системи

```
#include "Variables.h" // Підключаємо вкладку з Змінними
#include "Connect.h"

BlynkTimer timer; // Створюємо об'єкт BlynkTimer з іменем timer

#include "RTC.h" // Підключаємо вкладку з таймером реального часу
#include "BlynkCOD.h" // Підключаємо вкладку з кодом функцій Blynk
#include "DataLogger.h" // Підключаємо вкладку Запису на SD карту
#include "SensorDHT.h" // Підключаємо вкладку Датчика температури й вологості
#include "Modul_mfrc522.h" // Підключаємо вкладку Зчитувача RFID міток
#include "LCD_Menu.h" // Підключаємо вкладку LCD екрана
#include "Button.h" // Підключаємо вкладку Фізичної кнопки
#include "Server.h" // Підключаємо вкладку Сервера
#include "IK_Konder.h" // Підключаємо вкладку Керування кондиціонером

// Функція зчитування сигналів з датчика руху PIR
void PIRsensor(){

    // Зчитуємо стан датчика — якщо сигнал вище 300, піднімаємо прапорець statePIR,
    інакше опускаємо

    if (analogRead(PIN_PIR)>300){
        statePIR = HIGH;
    } else {
        statePIR = LOW;
    }

    // Якщо є рух — вмикаємо освітлення, інакше вимикаємо
    if (statePIR == HIGH){
        digitalWrite(PIN_RELE_SVET, HIGH);
    } else {
        digitalWrite(PIN_RELE_SVET, LOW);
    }
}

// Блок при вмиканні Arduino — ініціалізація всієї периферії
void setup() {
```

					<b>ІТС.4КІ.0125.03-ПЗ</b>		
Змн.	Арк.	№ докум.	Підпис	Дата	<b>ДОДАТОК А</b>		
Розроб.	Ліщук С.Е.						
Керівник	Донченко В.Ю.						
Реценз.	Козуб Ю.Г.						
Н. Контр.							
Зав. каф.	Семенов М.А.						
					Літ.	Арк.	Акрушів
						116	3
					ЛНУ		
					Кафедра ІТС, Гр.4КІ		

```

// Налаштування входів і виходів
pinMode(PIN_RELE_SVET, OUTPUT); // Реле освітлення
pinMode(PIN_RELE_LOCK, OUTPUT); // Реле замка
setupButton(); // Кнопка перемикання екранів (Button.h)
// Ініціалізуємо серійний монітор
Serial.begin(9600);
while (!Serial); // Чекаємо, поки не ініціалізується монітор серійного порту
Serial.println(F("Start"));
// Ініціалізація годинника реального часу (RTC.h)
setupRTC();
// Ініціалізація LCD (LCD_Menu.h)
lcd.init();
lcd.backlight(); // Вмикаємо підсвітку
// Ініціалізуємо датчик температури й вологості (SensorDHT.h)
Serial.println(F("DHTxx test!"));
dht.begin();
// Ініціалізація SD карти (DataLogger.h)
setupSD();
delay(1000);
// Ініціалізуємо модуль MFRC522 (Modul_mfrc522.h)
setupMFRC522();
delay(1000);
// Викликаємо функцію підключення до Blynk (Connect.h)
reconnectBlynk();
delay(1000);
setupTable(); // Очистити таблицю при старті
// Запускаємо сервер (Server.h)
server.begin();
// Налаштовуємо таймери
timer.setInterval(60000L, reconnectBlynk); // Перепідключення Blynk при розриві
з'єднання
timer.setInterval(2000, readDHT); // Читаємо температуру кожні 2 секунди
ID_Timer_keyRFID = timer.setInterval(500, keyRFID); // Читаємо RFID мітки та
відкриваємо/закриваємо замок
ID_Timer_ScreenSwitching = timer.setInterval(2000, ScreenSwitching); // Перемикаємо
меню екранів LCD

```

					<i>ИТС.4КІ.0125.03-ПЗ</i>	Арк.
						117
Змн.	Арк.	№ докум.	Підпис	Дата		

```

timer.disable(ID_Timer_ScreenSwitching); // Вимикаємо перемикання екранів LCD
timer.setInterval(50, [](){button1.tick();}); // Перевіряємо, чи натиснута кнопка
timer.setInterval(2000, LCDmenu); // Оновлюємо дані на LCD екрані
timer.setInterval(1000, PIRsensor); // Читаємо стан з датчика руху
timer.setInterval(1000, loopRTC); // Читаємо час з модуля RTC DS3231
timer.setInterval(1000, loopServer); // Обробляємо запити з сервера
timer.setInterval(1000, emailSENDrun); // Надсилаємо повідомлення, якщо потрібно
}
// Головний цикл
void loop() {
  if (Blynk.connected()) { // Якщо є підключення до сервера Blynk
    Blynk.run(); // Тік Blynk
  }
  timer.run(); // Тік таймера
}

```

					<i>ІТС.4КІ.0125.03-ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		118

## ДОДАТОК Б

### Скетч роботи кнопки перемикання меню LCD екрана

```
#include "OneButton.h" // Підключаємо бібліотеку для обробки кнопок
OneButton button1(PIN_BUTTON_MENU, true); // Оголошуємо екземпляр класу // При
одиначному натисканні викликаємо перемикання екрана
void click1() {
    Serial.println("ScreenSwitching");
    ScreenSwitching(); // Функція перемикання екранів
}
// При подвійному кліку вмикаємо або вимикаємо автоматичне перемикання екранів
void doubleclick1() {
    static boolean stateScreenSwitching = false; // Прапорець стану автоматичного
перемикання екранів
    // Завдяки static зміна значення зберігається
    stateScreenSwitching = !stateScreenSwitching; // Інвертуємо прапорець
    if (stateScreenSwitching) {
        timer.enable(ID_Timer_ScreenSwitching);
        Serial.println("ScreenSwitching AUTO ON");
    } else {
        timer.disable(ID_Timer_ScreenSwitching);
        Serial.println("ScreenSwitching AUTO OFF");
    }
}
// Ініціалізація кнопки
void setupButton() {
    button1.attachClick(click1); // Призначаємо функцію click1 для одиначного кліку
    button1.attachDoubleClick(doubleclick1); // Призначаємо функцію doubleclick1 для
подвійного кліку
}
```

					ІТС.4КІ.0125.03-ПЗ				
Змн.	Арк.	№ докум.	Підпис	Дата					
Розроб.	Ліщук С.Е.				ДОДАТОК Б		Літ.	Арк.	Акрушів
Керівник	Донченко В.Ю.							119	1
Реценз.	Козуб Ю.Г.						ЛНУ Кафедра ІТС, Гр.4КІ		
Н. Контр.									
Зав. каф.	Семенов М.А.								

## ДОДАТОК В

### Скетч підключення до мережі для проєкту на Arduino з Blynk

```
#define BLYNK_PRINT Serial
#include <SPI.h>
#include <Ethernet.h>
#include <BlynkSimpleEthernet.h>

// MAC-адреса має бути унікальною для кожного пристрою в локальній мережі
byte arduino_mac[] = { 0xDE, 0xED, 0xBA, 0xFE, 0xFE, 0xED };
IPAddress device_ip (192, 168, 10, 224);
IPAddress dns_ip (192, 168, 10, 1);
IPAddress gateway_ip (192, 168, 10, 1);
IPAddress subnet_mask(255, 255, 255, 0);

// Авторизаційний ключ Blynk приходить на пошту з додатку
#define AUTH "akls3e41dk1349sdajwqe123jfsdf"

// #define USE_LOCAL_SERVER // Розкоментуйте, якщо потрібен локальний сервер
Blynk

#ifdef USE_LOCAL_SERVER
    #define SERVER IPAddress (192, 168, 10, 9) // Вкажіть вашу локальну IP-адресу
    #define SERVER "Blynk.dlinkddns.com" // Або назву вашого хоста (приклад)
#endif

// Функція для підключення до Blynk
void ConnectBlynk() {
#ifdef USE_LOCAL_SERVER
    // Ініціалізація Ethernet
    Ethernet.begin(arduino_mac, device_ip, dns_ip, gateway_ip, subnet_mask);

    // Перевірка наявності Ethernet обладнання
    if (Ethernet.hardwareStatus() == EthernetNoHardware) {
        Serial.println(F("Ethernet шилд не знайдено. Робота неможлива :("));
        while (true) {
            delay(1); // Очікування – робота без Ethernet немає сенсу
        }
    }
}
```

					ІТС.4КІ.0125.03-ПЗ				
Змн.	Арк.	№ докум.	Підпис	Дата					
Розроб.	Ліщук С.Е.				ДОДАТОК В		Лім.	Арк.	Акрушіє
Керівник	Донченко В.Ю.							120	3
Реценз.	Козуб Ю.Г.						ЛНУ Кафедра ІТС, Гр.4КІ		
Н. Контр.									
Зав. каф.	Семенов М.А.								

```

// Перевірка підключення Ethernet кабелю
if (Ethernet.linkStatus() == LinkOFF) {
    Serial.println(F("Кабель Ethernet не підключено."));
} else {
    Serial.println(Ethernet.localIP());
    Blynk.config(AUTH, SERVER, 8080);
    Blynk.connect(10000UL);
}
#else
// Ініціалізація Ethernet (без локального сервера)
Ethernet.begin(arduino_mac, device_ip, dns_ip, gateway_ip, subnet_mask);
// Перевірка наявності Ethernet шилду
if (Ethernet.hardwareStatus() == EthernetNoHardware) {
    Serial.println(F("Ethernet шилд не знайдено. Робота неможлива :("));
    while (true) {
        delay(1);
    }
}
// Перевірка Ethernet кабелю
if (Ethernet.linkStatus() == LinkOFF) {
    Serial.println(F("Кабель Ethernet не підключено."));
} else {
    Serial.println(Ethernet.localIP());
    Blynk.config(AUTH, "blynk-cloud.com", 80);
    Blynk.connect(10000UL);
}
#endif
}

// Перепідключення при втраті з'єднання
void reconnectBlynk() {
    if (!Blynk.connected()) {
        BLYNK_LOG("Зараз відключено");
        ConnectBlynk();
        if (Blynk.connected()) {
            BLYNK_LOG("Підключено повторно");
        }
    }
}

```

					ІТС.4КІ.0125.03-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		121

```
} else {  
    BLYNK_LOG("Повторне підключення не вдалося");  
}  
}  
}
```

					ІТС.4КІ.0125.03-ПЗ	Арк.
						122
Змн.	Арк.	№ докум.	Підпис	Дата		

# ДОДАТОК Г

## Скетч роботи SD-карти

```
#include <SPI.h>
#include <SD.h>
// Ініціалізація SD карти
void setupSD() {
    Serial.print(F("Ініціалізація SD карти..."));
    // Перевіряємо, чи присутня карта і чи її можна ініціалізувати
    if (!SD.begin(SDCARD_CS)) {
        Serial.println(F("Помилка: карта не знайдена або не ініціалізується"));
        // return; // далі нічого не виконуємо
    }
    Serial.println(F("Карта ініціалізована.));
}
/* * Дві функції:
* loopSD() – зберігає дані про температуру та вологість
* loopSD_change() – зберігає дані лише якщо температура чи вологість змінились */
// Зберігаємо дані про температуру та вологість
void loopSD() {
    // Формуємо рядок для журналу
    String dataString = "";
    dataString += Data;
    dataString += ",";
    dataString += Time;
    dataString += ",";
    dataString += "Humi ";
    dataString += String(h);
    dataString += ", Temp ";
    dataString += String(t);
    // Відкриваємо файл (одночасно можна відкрити лише один файл!)
    File dataFile = SD.open("loght.txt", FILE_WRITE);
    // Якщо файл доступний, записуємо в нього:
```

					ІТС.4КІ.0125.03-ПЗ					
Змн.	Арк.	№ докум.	Підпис	Дата	ДОДАТОК Г			Літ.	Арк.	Акрушів
Розроб.		Ліщук С.Е.								
Керівник		Донченко В.Ю.							123	1
Реценз.		Козуб Ю.Г.						ЛНУ Кафедра ІТС, Гр.4КІ		
Н. Контр.										
Зав. каф.		Семенов М.А.								

```

if (dataFile) {
    dataFile.println(dataString);
    dataFile.close();
    Serial.print("DataLogger: ");
    Serial.println(dataString);
} else {
    // Якщо файл не відкрився, виводимо повідомлення про помилку
    Serial.println(F("Помилка відкриття loght.txt"));
}
}

// Зберігаємо дані лише якщо температура або вологість змінилися
void loopSD_change() {
    String dataString = "";
    static float last_h = 0;
    static float last_t = 0;
    if (last_h != h && last_t != t) {
        last_h = h;
        last_t = t;
        dataString += Data;
        dataString += ",";
        dataString += Time;
        dataString += ",";
        dataString += "Humi ";
        dataString += String(h);
        dataString += ", Temp ";
        dataString += String(t);
        File dataFile = SD.open("loght.txt", FILE_WRITE);
        if (dataFile) {
            dataFile.println(dataString);
            dataFile.close();
            Serial.print("DataLogger: ");
            Serial.println(dataString);
        } else {
            Serial.println(F("Помилка відкриття loght.txt"));
        }
    }
}

```

					ІТС.4КІ.0125.03-ПЗ	Арк.
						124
Змн.	Арк.	№ докум.	Підпис	Дата		

```

}

// Зберігаємо подію про відкриття або закриття замка
void lockSD(String UID, String Value) {
    // Формуємо рядок для журналу
    String dataString = "";
    dataString += Data;
    dataString += ",";
    dataString += Time;
    dataString += ",";
    dataString += "Ключ ";
    dataString += ",";
    dataString += UID;
    dataString += ",";
    dataString += Value;
    File dataFile = SD.open("lock.txt", FILE_WRITE);
    if (dataFile) {
        dataFile.println(dataString);
        dataFile.close();
        Serial.print("DataLogger: ");
        Serial.println(dataString);
    } else {
        Serial.println(F("Помилка відкриття lock.txt"));
    }
}

```

					ITC.4KI.0125.03-ПЗ	Арк.
						125
Змн.	Арк.	№ докум.	Підпис	Дата		

## ДОДАТОК Д

### Скетч роботи LCD-дисплея

```
#include <Wire.h>

#include <LiquidCrystal_I2C.h> // Бібліотека для LCD екрана

LiquidCrystal_I2C lcd(0x27, 16, 2); // Оголошуємо екран за адресою 0x27 на шині I2C, 2
рядки по 16 символів

// Прототип функції, щоб компілятор знав про її існування
void LCDmenu();

// Функція, що відповідає за перемикання екранів
void ScreenSwitching() {
    MenuPos++;
    if (MenuPos > 2) {
        MenuPos = 0;
    }
    LCDmenu();
}

// Функція виведення інформації на LCD екран
void LCDmenu() {
    static int8_t LastMenuPos = 100;
    switch (MenuPos) {
        case 0:
            // Виконується, коли MenuPos дорівнює 0
            if (LastMenuPos != MenuPos) {
                // Якщо попередній екран був інший — очищаємо все і виводимо новий
                lcd.clear();
                lcd.setCursor(0, 0); lcd.print(F("Hum = % "));
                lcd.setCursor(7, 0); lcd.print(h);
                lcd.setCursor(0, 1); lcd.print(F("Temp = 1C "));
                lcd.setCursor(7, 1); lcd.print(t);
            }
        }
    }
```

					ІТС.4КІ.0125.03-ПЗ					
Змн.	Арк.	№ докум.	Підпис	Дата	ДОДАТОК Д			Лім.	Арк.	Акрушів
Розроб.		Ліщук С.Е.								
Керівник		Донченко В.Ю.							126	1
Реценз.		Козуб Ю.Г.						ЛНУ Кафедра ІТС, Гр.4КІ		
Н. Контр.										
Зав. каф.		Семенов М.А.								

```

    } else {
        // Інакше оновлюємо лише значення
        lcd.setCursor(7, 0); lcd.print(h);
        lcd.setCursor(7, 1); lcd.print(t);
    }
    break;
case 1:
    // Виконується, коли MenuPos дорівнює 1
    lcd.clear();
    lcd.setCursor(0, 0);
    if (statePIR == HIGH) {
        lcd.print(F("SVET VKL"));
    } else {
        lcd.print(F("SVET OTKL"));
    }
    break;
case 2:
    // Виконується, коли MenuPos дорівнює 2
    lcd.clear();
    lcd.setCursor(0, 0);
    if (!StateKeyRFID) {
        lcd.print(F("CLOSED"));
    } else if (StateKeyRFID) {
        lcd.print(F("OPEN"));
    }
    break;
default:
    // Якщо не обрано жодну з альтернатив
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print(F("MenuPos Error"));
    break;
}
LastMenuPos = MenuPos;
}

```

					ІТС.4КІ.0125.03-ПЗ	Арк.
						127
Змн.	Арк.	№ докум.	Підпис	Дата		

## ДОДАТОК Е

### Скетч роботи RFID-зчитувача MFRC522

```
#include <SPI.h>

#include <MFRC522.h> // Підключаємо бібліотеку для роботи з модулем RC522
MFRC522 mfrc522(MFRC522_SS_PIN, MFRC522_RST_PIN); // Створюємо об'єкт для
роботи

// Допоміжна функція — виводить масив байтів у HEX-форматі
void printHex(byte *buffer, byte bufferSize) {
    for (byte i = 0; i < bufferSize; i++) {
        Serial.print(buffer[i] < 0x10 ? " 0" : " ");
        Serial.print(buffer[i], HEX);
    }
}

// Повертає UID у вигляді рядка HEX
String printHexs(byte *buffer, byte bufferSize) {
    String UID = "";
    for (byte i = 0; i < bufferSize; i++) {
        UID += buffer[i] < 0x10 ? " " : "";
        UID += String(buffer[i], HEX);
    }
    return UID;
}

// Виводить масив у десятковому форматі
void printDec(byte *buffer, byte bufferSize) {
    for (byte i = 0; i < bufferSize; i++) {
        Serial.print(buffer[i] < 0x10 ? " 0" : " ");
        Serial.print(buffer[i], DEC);
    }
}

// Ініціалізація зчитувача RFID
void setupMFRC522() {
```

					<b>ІТС.4КІ.0125.03-ПЗ</b>			
<b>Змн.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підпис</b>	<b>Дата</b>	<b>ДОДАТОК Е</b>	<b>Лім.</b>	<b>Арк.</b>	<b>Акрушіє</b>
Розроб.		Ліщук С.Е.						
Керівник		Донченко В.Ю.					128	1
Реценз.		Козуб Ю.Г.				ЛНУ		
Н. Контр.						Кафедра ІТС, Гр.4КІ		
Зав. каф.		Семенов М.А.						

```

SPI.begin();           // Ініціалізуємо SPI
mfr522.PCD_Init();     // Ініціалізуємо модуль RC522
mfr522.PCD_DumpVersionToSerial(); // Виводимо прошивку
Serial.println(F("Піднесіть RFID-карту до зчитувача"));
}

// Основна функція обробки RFID
void keyRFID() {
    // Якщо немає нової мітки — виходимо
    if (!mfr522.PICC_IsNewCardPresent()) return;
    // Якщо мітку не прочитано — виходимо
    if (!mfr522.PICC_ReadCardSerial()) return;
    // Вивід UID в монітор порту
    Serial.println(F("UID карти"));
    Serial.println(F("HEX: "));
    printHex(mfr522.uid.uidByte, mfr522.uid.size);
    Serial.println();
    Serial.print(F("DEC: "));
    printDec(mfr522.uid.uidByte, mfr522.uid.size);
    Serial.println();
    // Перевірка UID по базі ключів
    int flagCheck = 0;
    for (byte j = 0; j < countKey; j++) {
        for (byte i = 0; i < 4; i++) {
            if (uidCard[j][i] != mfr522.uid.uidByte[i]) {
                flagCheck++;
                i = 4;
            }
        }
    }
    // Якщо хоч один ключ збігся
    if (flagCheck < countKey) {
        StateKeyRFID = !StateKeyRFID; // Змінюємо стан замка
        if (StateKeyRFID) {
            Serial.println(F("OPEN"));
            sendEvent(printHexs(mfr522.uid.uidByte, mfr522.uid.size), F("OPEN"));
        }
    }
}

```

```

        emailSEND(TextMFRC522(printHexs(mfrc522.uid.uidByte,          mfrc522.uid.size),
F("OPEN"))));
        lockSD(printHexs(mfrc522.uid.uidByte, mfrc522.uid.size), F("OPEN"));
    } else {
        Serial.println(F("CLOSE"));
        sendEvent(printHexs(mfrc522.uid.uidByte, mfrc522.uid.size), F("CLOSE"));
        emailSEND(TextMFRC522(printHexs(mfrc522.uid.uidByte,          mfrc522.uid.size),
F("CLOSE"))));
        lockSD(printHexs(mfrc522.uid.uidByte, mfrc522.uid.size), F("CLOSE"));
    }
}

// Керуємо реле та надсилаємо в Blynk
digitalWrite(PIN_RELE_LOCK, StateKeyRFID);
Blynk.virtualWrite(VPIN_LOCK, StateKeyRFID);
// Захист: опитування лише 1 раз на 3 секунди
timer.disable(ID_Timer_keyRFID);
timer.setTimeout(3000, []() {
    timer.enable(ID_Timer_keyRFID);
});
}

```

## ДОДАТОК Є

### Скетч роботи годинника реального часу

```
#include <Wire.h>
#include <DS3231.h>

DS3231 clock;      // Оголошуємо екземпляр класу
RTCDatetime dt;    // Структура, в яку зберігаються всі дані з модуля RTC
// Ініціалізація модуля DS3231
void setupRTC() {
    Serial.println(F("Ініціалізація DS3231"));
    clock.begin();
    // Встановити час компіляції (можна розкоментувати для одноразового встановлення)
    // clock.setDateTime(__DATE__, __TIME__);
    // Вимикаємо тривоги та очищаємо аварійні сигнали (для прикладу)
    // У нормальних умовах їх варто скидати після запуску живлення або
перезавантаження
    clock.armAlarm1(false);
    clock.armAlarm2(false);
    clock.clearAlarm1();
    clock.clearAlarm2();
}
void loopRTC() {
    dt = clock.getDateTime(); // Зчитуємо дату й час у структуру dt
    // Формуємо строки
    Data = String(dt.day) + "." + String(dt.month) + "." + String(dt.year);
    Time = String(dt.hour) + ":" + String(dt.minute) + ":" + String(dt.second);
    DateTime = Data + " " + Time;
    // Надсилаємо час у віджет Blynk
    Blynk.virtualWrite(VPIN_LABEL_TIME, DateTime);
    // Виводимо в консоль
    Serial.println(DateTime);
    // Додатковий вивід поелементно:
    Serial.print("Raw data: ");
```

					ІТС.4КІ.0125.03-ПЗ									
Змн.	Арк.	№ докум.	Підпис	Дата	ДОДАТОК Є				Лім.		Арк.	Акрушіє		
Розроб.	Ліщук С.Е.										131	1		
Керівник	Донченко В.Ю.													
Реценз.	Козуб Ю.Г.													
Н. Контр.														
Зав. каф.	Семенов М.А.				ЛНУ Кафедра ІТС, Гр.4КІ									

```

Serial.print(dt.year); Serial.print("-");
Serial.print(dt.month); Serial.print("-");
Serial.print(dt.day); Serial.print(" ");
Serial.print(dt.hour); Serial.print(":");
Serial.print(dt.minute); Serial.print(":");
Serial.println(dt.second);
}

```

					<b>ITC.4Kl.0125.03-ПЗ</b>	Арк.
						132
Змн.	Арк.	№ докум.	Підпис	Дата		

## ДОДАТОК Ж

### Скетч роботи з датчиком температури та вологості DHT

```
#include <Wire.h>

#include <DHT.h> // Підключаємо бібліотеку термодатчика

#define DHTTYPE DHT21 // Вказуємо модель датчика (наприклад, DHT21 або DHT22)

DHT dht(DHTPIN, DHTTYPE); // Оголошуємо датчик

// Функція перевірки температури на вихід за межі
void alarmT() {
    if (t > t_max) { t_alarm_max = true; } else { t_alarm_max = false; }
    if (t < t_min) { t_alarm_min = true; } else { t_alarm_min = false; }
    send_t_alarm_max = false;
    send_t_alarm_min = false;
}

// Функція перевірки вологості на вихід за межі
void alarmH() {
    if (h > h_max) { h_alarm_max = true; } else { h_alarm_max = false; }
    if (h < h_min) { h_alarm_min = true; } else { h_alarm_min = false; }
    send_h_alarm_max = false;
    send_h_alarm_min = false;
}

// Зчитуємо дані з датчика, надсилаємо в Blynk, відправляємо аварійні листи
void readDHT() {
    h = dht.readHumidity();
    t = dht.readTemperature(); // або dht.readTemperature(true) для Фаренгейтів
    // Якщо не вдалось зчитати дані — вихід
    if (isnan(h) || isnan(t)) {
        Serial.println(F("Помилка зчитування з DHT датчика!"));
        return;
    }
    // Відправляємо в Blynk
    Blynk.virtualWrite(VPIN_LABEL_TEMP, t);
    Blynk.virtualWrite(VPIN_LABEL_HUMID, h);
}
```

					ІТС.4КІ.0125.03-ПЗ						
Змн.	Арк.	№ докум.	Підпис	Дата	ДОДАТОК Ж			Літ.	Арк.	Акрушів	
Розроб.	Ліщук С.Е.										
Керівник	Донченко В.Ю.									133	1
Реценз.	Козуб Ю.Г.							ЛНУ Кафедра ІТС, Гр.4КІ			
Н. Контр.											
Зав. каф.	Семенов М.А.										

```

// Перевірка аварій
alarmT();
alarmH();

// Відправка повідомлень про аварії температури
if (t_alarm_max && !send_t_alarm_max) {
    emailSEND(TextDHT("Увага! Перевищено норму температури", String(t)));
    send_t_alarm_max = true;
}

if (t_alarm_min && !send_t_alarm_min) {
    emailSEND(TextDHT("Увага! Температура нижча за норму", String(t)));
    send_t_alarm_min = true;
}

// Відправка повідомлень про аварії вологості
if (h_alarm_max && !send_h_alarm_max) {
    emailSEND(TextDHT("Увага! Перевищено норму вологості", String(h)));
    send_h_alarm_max = true;
}

if (h_alarm_min && !send_h_alarm_min) {
    emailSEND(TextDHT("Увага! Вологість нижча за норму", String(h)));
    send_h_alarm_min = true;
}
}

```

## ДОДАТОК 3

### Скетч взаємодії з сервером

```
EthernetServer server(80); // Ініціалізуємо сервер на порту 80

void loopServer() {
    EthernetClient client = server.available();
    if (client) { // Якщо клієнт з'єднався
        boolean currentLineIsBlank = true;
        while (client.connected()) {
            if (client.available()) {
                char c = client.read();
                if (newInfo && c == ' ') {
                    newInfo = 0; // Скидаємо прапорець нової інформації
                }
                if (c == '$') { // Отримали символ "$", початок нових даних
                    newInfo = 1;
                }
                // Якщо є нова інформація, перевіряємо її вміст
                if (newInfo == 1) {
                    Serial.println(c);
                    if (c == '1') {
                        Serial.println(F("Увімкнути"));
                        digitalWrite(PIN_RELE_SVET, HIGH); // Вмикаємо освітлення
                    }
                    if (c == '2') {
                        Serial.println(F("Вимкнути"));
                        digitalWrite(PIN_RELE_SVET, LOW); // Вимикаємо освітлення
                    }
                }
            }
            // Відслідковуємо кінець рядка
            if (c == '\n') {
                currentLineIsBlank = true;
            } else if (c != '\r') {
                currentLineIsBlank = false;
            }
        }
    }
}
```

					ІТС.4КІ.0125.03-ПЗ				
Змн.	Арк.	№ докум.	Підпис	Дата					
Розроб.		Ліщук С.Е.			ДОДАТОК 3	Літ.	Арк.	Акрушів	
Керівник		Донченко В.Ю.					135	1	
Реценз.		Козуб Ю.Г.				ЛНУ Кафедра ІТС, Гр.4КІ			
Н. Контр.									
Зав. каф.		Семенов М.А.							

```

currentLineIsBlank = false;
}
// Коли отримали порожній рядок — формуємо відповідь HTML
if (c == '\n' && currentLineIsBlank) {
    client.println(F("HTTP/1.1 200 OK"));
    client.println(F("Content-Type: text/html"));
    client.println(F("Connection: close"));
    client.println(F("Refresh: 5")); // автооновлення кожні 5 сек
    client.println();
    client.println(F("<!DOCTYPE HTML>"));
    client.println(F("<html>"));
    client.println(F("<head>"));
    client.println(F("<meta http-equiv='Content-Type' content='text/html; charset=utf-8'/>"));

    client.println(F("<title>Microcomplex</title>"));
    client.println(F("</head>"));
    client.println(F("<body>"));
    client.println(F("<h1>Мікрокомплекс</h1><hr>"));
    // Температура
    client.print(F("Температура = "));
    client.print(t);
    client.print(F(" °C"));
    if (t_alarm_max) {
        client.println(F(" Увага! Температура перевищує норму"));
    } else if (t_alarm_min) {
        client.println(F(" Увага! Температура нижча за норму"));
    }
    client.println(F("<br>"));
    // Вологість
    client.print(F("Вологість = "));
    client.print(h);
    client.println(F(" %"));
    if (h_alarm_max) {
        client.println(F(" Увага! Вологість перевищує норму"));
    } else if (h_alarm_min) {

```

```

        client.println(F(" Увага! Вологість нижча за норму"));
    }
    client.println(F("<br>"));
    // Стан освітлення
    if (statePIR == HIGH) {
        client.println(F("Світло в приміщенні увімкнено"));
    } else {
        client.println(F("Світло в приміщенні вимкнено"));
    }
    client.println(F("<br>"));
    // Стан дверей (замка)
    if (!StateKeyRFID) {
        client.println(F("Серверне приміщення зачинене"));
    } else {
        client.println(F("Серверне приміщення відчинене"));
    }
    client.println(F("</body>"));
    client.println(F("</html>"));
    client.println(F("<hr>"));
    break; // Завершення відповіді
}
}
}
delay(1);    // Очікування на завершення обміну
client.stop(); // Закриваємо з'єднання
}
}

```

## ДОДАТОК И

### скетч для керування кондиціонером через ІЧ-випромінювач (бібліотека IRremote)

```
#include <IRremote.h>

IRsend irsend;

void loopIN() {
    int khz = 38; // задаємо частоту 38 кГц
    // Сигнал ІЧ-пульта для увімкнення кондиціонера (20°C, 40% вологість)
    unsigned int irSignal_on[] = {
        9000, 4500, 560, 560, 560, 560, 560, 1690, 560, 560, 560, 560, 560, 560,
        560, 560, 560, 560, 560, 1690, 560, 1690, 560, 560, 560, 1690, 560, 1690,
        560, 560, 560, 560, 560, 560, 560, 560, 560, 1690, 560, 560, 560, 1690,
        560, 560, 560, 560, 560, 560, 1690, 560, 1690, 560, 1690, 560, 1690,
        560, 560, 560, 1690, 560, 560, 560, 1690, 560, 560, 560, 1690, 560, 1690,
        560, 1690, 560, 1690, 560, 560, 39416, 9000, 4500, 560
    };

    // Сигнал ІЧ-пульта для вимкнення кондиціонера
    unsigned int irSignal_off[] = {
        8000, 4300, 500, 400, 450, 450, 560, 1690, 560, 560, 560, 560, 460, 560,
        560, 1690, 560, 560, 1590, 560, 1590, 560, 560, 560, 560, 560, 560, 560,
        560, 560, 560, 560, 560, 560, 560, 560, 560, 560, 1690, 560, 1690, 560,
        1690, 560, 1690, 560, 1690, 560, 1690, 560, 560, 560, 560, 560, 1690, 560,
        560, 560, 1690, 560, 560, 560, 1690, 560, 560, 560, 1690, 560, 560, 560,
        560, 1690, 560, 560, 560, 1690, 560, 1690, 560, 560, 560, 39416, 8000, 2210, 460
    };

    // Увімкнення при перевищенні температури
    if (t > t_max) {
        irsend.sendRaw(irSignal_on, sizeof(irSignal_on) / sizeof(irSignal_on[0]), khz); //
        Вмикаємо кондиціонер
    } else {
```

					<b>ІТС.4КІ.0125.03-ПЗ</b>		
Змн.	Арк.	№ докум.	Підпис	Дата	<b>ДОДАТОК И</b>		
Розроб.	Ліщук С.Е.						
Керівник	Донченко В.Ю.						
Реценз.	Козуб Ю.Г.						
Н. Контр.							
Зав. каф.	Семенов М.А.				<b>ЛНУ</b> <b>Кафедра ІТС, Гр.4КІ</b>		
					Лім.	Арк.	Акрушіє
						138	1

```

        irsend.sendRaw(irSignal_off, sizeof(irSignal_off) / sizeof(irSignal_off[0]), khz);    //
Вмикаємо кондиціонер
    }
    delay(10000); // Пауза 10 секунд
    // Увімкнення при перевищенні вологості
    if (h > h_max) {
        irsend.sendRaw(irSignal_on, sizeof(irSignal_on) / sizeof(irSignal_on[0]), khz);    //
Вмикаємо кондиціонер
    } else {
        irsend.sendRaw(irSignal_off, sizeof(irSignal_off) / sizeof(irSignal_off[0]), khz);    //
Вмикаємо кондиціонер
    }
    delay(50000); // Пауза 50 секунд
}

```

					ІТС.4КІ.0125.03-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		139

## ДОДАТОК І

### Скетч оголошення змінних

```
// Пристрої, що працюють через шину I2C, підключаємо до:
// I2C SCL A5 (UNO) / 21 (Mega)
// I2C SDA A4 (UNO) / 20 (Mega)

// Пристрої, підключені через SPI
#define MFRC522_RST_PIN 5 // Пін підключення RST модуля MFRC522
#define MFRC522_SS_PIN 53 // Пін підключення SDA(SS) модуля MFRC522
// SPI W5100 SD MFRC522

// MOSI 51
// MISO 50
// SCK 52

#define SDCARD_CS 4 // CS пін для SD-карти

// Інші пристрої
#define DHTPIN A0 // Пін, до якого підключено датчик DHT
#define PIN_RELE_SVET 7 // Реле освітлення
#define PIN_RELE_LOCK 6 // Реле замка
#define PIN_PIR A1 // Вхід датчика руху
#define PIN_BUTTON_MENU 3 // Кнопка перемикання екранів

// Віртуальні піни для Blynk
#define VPIN_LABEL_TIME V0
#define VPIN_LABEL_TEMP V1
#define VPIN_LABEL_HUMID V2
#define VPIN_LOCK V5
#define VPIN_TABLE V10

// Глобальні змінні
// Дані часу
String Data;
String Time;
String DataTime;

// Стан PIR сенсора
boolean statePIR = 0;
```

					ІТС.4КІ.0125.03-ПЗ				
Змн.	Арк.	№ докум.	Підпис	Дата					
Розроб.	Ліщук С.Е.				ДОДАТОК І		Літ.	Арк.	Акрушіє
Керівник	Донченко В.Ю.							140	1
Реценз.	Козуб Ю.Г.						ЛНУ  Кафедра ІТС, Гр.4КІ		
Н. Контр.									
Зав. каф.	Семенов М.А.								

```

// Позиція меню
int8_t MenuPos = 0;

// Прапорець отримання нової інформації з сайту
boolean newInfo = 0;

// Дані з датчика температури та вологості
float h; // Вологість
int h_min = 20;
int h_max = 70;
boolean h_alarm_min = false;
boolean h_alarm_max = false;
boolean send_h_alarm_min = false;
boolean send_h_alarm_max = false;
float t; // Температура
int t_min = 18;
int t_max = 25;
boolean t_alarm_min = false;
boolean t_alarm_max = false;
boolean send_t_alarm_min = false;
boolean send_t_alarm_max = false;

// Ідентифікатори таймерів
int ID_Timer_keyRFID;
int ID_Timer_ScreenSwitching;

// Надсилання листів (через Blynk або іншу систему)
#define USE_MAIL_SEND
#define BLYNK_MAX_SENDBYTES 1000
char MAIL[] = "arduino@gmail.com"; // Email для повідомлень
char Subject[] = "Subject: Microcomplex"; // Тема повідомлення
String Message = ""; // Тіло повідомлення
boolean stateSEND_Mail_Timer = false; // Дозвіл на надсилання
boolean waitSEND = false; // Очікування відправки

// Стан RFID замка
boolean StateKeyRFID = false;

// Зчитування UID карток (через скетч DumpInfo_My)
const int countKey = 2;

// UID ключів у HEX форматі

```

					ІТС.4КІ.0125.03-ПЗ	Арк.
						141
Змн.	Арк.	№ докум.	Підпис	Дата		

```

byte uidCard[countKey][4] = {
    {0x05, 0xB6, 0x97, 0xBB},
    {0x50, 0x27, 0x9E, 0x7C}
};

// Імена для UID (опціонально)
String NameUID[countKey] = {
    "Петро",
    "Василь"
};

```

					ІТС.4КІ.0125.03-ПЗ	Арк.
						142
Змн.	Арк.	№ докум.	Підпис	Дата		