

Київський столичний університет імені Бориса Грінченка
Факультет інформаційних технологій та математики
Кафедра комп'ютерних наук

Допущено до захисту
Завідувач
кафедри комп'ютерних наук
К.Т.Н., доцент
(науковий ступінь, вчене звання)

Ірина МАШКІНА
(підпис)

«___» _____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня «бакалавр»
Спеціальність 122 Комп'ютерні науки
Освітня програма 122.00.01 Інформатика

Тема:

**РОЗРОБКА SMART-СИСТЕМИ ДЛЯ КЕРУВАННЯ МІКРОКЛІМАТОМ У
ПРИМІЩЕННІ НА БАЗІ ARDUINO**

Виконав (ла)
студент (ка) групи ІНб-1-21-4.0д
Прокопенко Олександр Анатолійович
(ПІБ)

(підпис)

Науковий керівник:

К.Т.Н., ст. досл.

Багацький О.В.
(прізвище, ініціали)

(підпис)

Київ 2025

Київський університет імені Бориса Грінченка
Факультет інформаційних технологій та математики
Кафедра комп'ютерних наук

«Затверджую»
Завідувач кафедри комп'ютерних
наук,
кандидат технічних наук, доцент
_____ Ірина МАШКІНА

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА
«Розробка smart-системи для керування мікрокліматом у приміщенні на базі
Arduino»

Виконавець – студент групи ІНб-1-21-4.0д
Прокопенко Олександр Анатолійович

1. Вихідні дані: *Законодавство та нормативно-правові акти України в інформаційній сфері. Публікації за напрямом дослідження.*
2. Основні завдання: *Здійснити огляд подібних інформаційних систем за темою роботи. Обґрунтувати мету, об'єкт, предмет дослідження. Розробити завдання, структуру та зміст бакалаврської дипломної роботи. Обрати та обґрунтувати методи та засоби розробки. Підготувати матеріали до розділів роботи відповідно до індивідуального завдання. Розробити структуру розробленої системи. Створити діючий прототип вузлів системи. Провести аналіз результатів, скласти висновки та оформити бакалаврську дипломну роботу відповідно до затверджених на кафедрі вимог. Підготувати презентацію за темою роботи.*
3. Пояснювальна записка: *Обсяг – до 45 стор. формату А4 комп'ютерного набору з дотриманням вимог стандарту і методичних рекомендацій кафедри.*
4. Графічні матеріали: *UML діаграми, скріншоти схеми підключення вузлів системи, презентація.*
5. Додатки: *програмна реалізація алгоритму роботи системи*
6. Строк подання роботи на кафедру: «___» _____ 2025 р.

Науковий керівник
к.т.н., ст.досл.
_____ Багацький О.В.

Виконавець:
студент групи ІНб-1-21-4.0д
_____ Прокопенко О. А.

Календарний план

№	Назва етапу дипломної роботи	Строк виконання етапу роботи	Примітка
1	Формування теми дипломної роботи бакалавра	до	Виконано
2	Ознайомлення з методичними рекомендаціями до дипломної роботи		Виконано
3	Складання змісту та календарного плану роботи		Виконано
4	Підбір літератури, складання завдання		Виконано
5	Написання реферату та вступу		Виконано
6	Написання першого розділу		Виконано
7	Написання другого розділу		Виконано
8	Написання третього розділу		Виконано
9	Написання четвертого розділу		Виконано
10	Написання висновків		Виконано
11	Виправлення зауважень		Виконано
12	Створення презентації		Виконано
13	Захист матеріалів дипломної роботи на засіданні кафедри	за 1 місяць до захисту	
14	Офіційний захист матеріалів дипломної роботи на засіданні екзаменаційної комісії	згідно графіку захистів	

Студент _____

Керівник роботи _____

РЕФЕРАТ

Пояснювальна записка до дипломної роботи «Розробка smart-системи для керування мікрокліматом у приміщенні на базі Arduino» містить 46 сторінок, 18 рисунків, список використаних джерел – 25 найменувань, 1 додаток.

Об'єкт дослідження: процес автоматизованого керування температурою та вологістю у замкнутому приміщенні.

Предмет дослідження: програмно-апаратна система на базі Arduino для моніторингу та регулювання мікрокліматичних параметрів.

Мета роботи: розробити ефективну smart-систему керування мікрокліматом із використанням мікроконтролера Arduino, датчиків температури та вологості, а також механізмів автоматичного регулювання.

Методи проектування: включають теоретичні аспекти, такі як синтез, узагальнення й концептуалізація, методи структурного аналізу, а також методи зворотного чи елементарно-теоретичного аналізу, порівняльного та системного аналізу, методи проектування апаратних та програмних систем (структурне моделювання, UML-діаграми); Експериментальні методи (тестування роботи датчиків, перевірка точності вимірювань, аналіз ефективності системи).

В результаті роботи розроблено програмно-апаратний комплекс, що забезпечує автоматичний моніторинг температури та вологості, керування опалювальними і вентиляційними пристроями згідно з заданими параметрами, а також можливість візуалізації даних через зручний інтерфейс. Комплекс може бути застосований у різноманітних сферах, зокрема для житлових та офісних приміщень, складів, теплиць, серверних кімнат та інших об'єктів, де важливий контроль мікроклімату.

КЛЮЧОВІ СЛОВА: SMART-СИСТЕМА, ARDUINO, МІКРОКЛІМАТ, ТЕМПЕРАТУРА, ВОЛОГІСТЬ, ДАТЧИКИ, АВТОМАТИЗАЦІЯ, ПРОГРАМНО-АПАРАТНИЙ КОМПЛЕКС.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. ОГЛЯД ІСНУЮЧИХ РІШЕНЬ ДЛЯ КОНТРОЛЮ МІКРОКЛІМАТУ В ПРИМІЩЕННЯХ	10
1.1. Аналіз апаратних рішень для контролю температури та вологості	10
1.2. Огляд програмних засобів для керування мікрокліматом	12
Висновки до розділу 1	15
РОЗДІЛ 2. ВИБІР АПАРАТНИХ ТА ПРОГРАМНИХ ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ СИСТЕМИ	16
2.1. Обґрунтування вибору апаратної платформи (Arduino)	16
2.2. Вибір датчиків для контролю мікроклімату	18
Висновки до розділу 2	21
РОЗДІЛ 3. ФУНКЦІОНАЛЬНИЙ ОПИС СИСТЕМИ ДЛЯ КЕРУВАННЯ МІКРОКЛІМАТОМ У ПРИМІЩЕННІ	23
3.1. Структурна схема системи	23
3.2. UML-діаграми (Use Case, класів або розгортання)	24
3.3. Алгоритми функціонування системи	27
Висновки до розділу 3	29
РОЗДІЛ 4. РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ	31
4.1. Апаратна реалізація (схема підключення, вибір компонентів)	31
4.2. Програмна реалізація (код, інтерфейс, логіка керування)	34
4.3. Тестування системи та аналіз результатів	37
Висновки до розділу 4	42
ВИСНОВКИ	44
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	45
Додаток 1	47

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

САПР - Система автоматизованого проектування

IoT - Internet of Things (Інтернет речей)

API - Application Programming Interface (Інтерфейс програмування застосунків)

GUI - Graphical User Interface (Графічний інтерфейс користувача)

DBMS - Database Management System (Система керування базами даних)

SQL - Structured Query Language (Мова структурованих запитів)

PWM - Pulse Width Modulation (Широтно-імпульсна модуляція)

PLC - Programmable Logic Controller (Програмований логічний контролер)

UART - Universal Asynchronous Receiver-Transmitter (Універсальний асинхронний приймач-передавач)

HTTP - HyperText Transfer Protocol (Протокол передачі гіпертексту)

Wi-Fi - Wireless Fidelity (Бездротова фідельність)

REST - Representational State Transfer (Перенос представлення стану)

ВСТУП

Актуальність теми дослідження обумовлена постійним зростанням потреби в ефективному управлінні мікрокліматом в різних типах приміщень, таких як житлові будинки, офіси, склади, теплиці, серверні кімнати та інші об'єкти, де важливо забезпечити стабільні умови для зберігання товарів, роботи техніки чи комфортного перебування людей. Погіршення якості повітря, неправильне регулювання температури та вологості може призвести до різних негативних наслідків, включаючи збільшення енергоспоживання, зниження продуктивності праці, а також шкоду для здоров'я людей. Тому виникає необхідність у створенні інтелектуальних систем, які здатні автоматично моніторити і регулювати мікроклімат у реальному часі.

Метою дослідження є розробка програмно-апаратного комплексу для автоматизації процесу моніторингу та регулювання температури й вологості, що дозволяє забезпечити оптимальні умови для різних типів приміщень.

Завданнями дослідження є:

1. Розробка алгоритмів автоматичного моніторингу та регулювання температури й вологості.
2. Створення інтерфейсу для візуалізації даних та керування пристроями.
3. Тестування та оптимізація програмно-апаратного комплексу для забезпечення стабільної роботи в реальних умовах.
4. Оцінка ефективності застосування комплексу в різних сферах.

Об'єктом дослідження є програмно-апаратний комплекс для автоматизації управління мікрокліматом у приміщеннях.

Предметом дослідження є технології моніторингу та регулювання температури і вологості, а також інтеграція систем для забезпечення оптимального мікроклімату.

Методи проєктування включають використання сучасних програмних і апаратних рішень, алгоритмів автоматичного контролю, а

також методи тестування і оцінки ефективності впроваджених технологій.

Застосування такого комплексу є актуальним для широкого кола об'єктів: житлових та офісних приміщень, складів, теплиць, серверних кімнат і будь-яких інших об'єктів, де важливий контроль за мікрокліматом для забезпечення комфортних і безпечних умов.

1 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ ДЛЯ КОНТРОЛЮ МІКРОКЛІМАТУ В ПРИМІЩЕННЯХ

1.1 Аналіз апаратних рішень для контролю температури та вологості

Сучасні системи контролю мікроклімату в приміщеннях базуються на різних апаратних рішеннях, які відрізняються за точністю вимірювань, функціональними можливостями та вартістю. Одними з найпоширеніших є готові комерційні рішення, такі як кліматичні контролери від компаній Honeywell, Nest або Xiaomi. Вони зазвичай інтегровані в систему "розумного дому" та дозволяють керувати опаленням, вентиляцією та кондиціонуванням через мобільний додаток. Такі системи мають вбудовані датчики температури, вологості, а іноді й якості повітря, проте їхня вартість є досить високою, а можливості кастомізації обмежені.

Альтернативою є модульні системи на базі мікроконтролерів, такі як Arduino або Raspberry Pi. Вони дозволяють створювати гнучкі рішення з використанням різноманітних датчиків (DHT22, BME280, DS18B20) та виконавчих пристроїв (реле, сервоприводи, вентилятори). Перевагою таких систем є можливість точної настройки логіки роботи, інтеграції з іншими пристроями та низька вартість компонентів. Однак вони вимагають технічних знань для налаштування та програмування.

Окремо варто відзначити промислові системи контролю мікроклімату, такі як SCADA-системи або спеціалізовані контролери (Siemens), Schneider Electric тощо. Вони забезпечують високу точність і надійність, але їхнє використання в малих приміщеннях або побутових умовах є недоцільним через складність і високу вартість.

Таким чином, для розробки ефективної та доступної системи керування мікрокліматом оптимальним варіантом є використання модульної платформи на базі Arduino з підключенням недорогих, але точних датчиків. Це дозволить забезпечити необхідний рівень автоматизації при мінімальних

витратах.

Таблиця 1.1

Огляд апаратних рішень для систем контролю мікроклімату

Тип системи / Пристрій	Характеристики	Переваги	Недоліки
Готові комерційні рішення	Інтегровані кліматичні контролери (Honeywell, Nest, Xiaomi), вбудовані датчики (температура, вологість, іноді якість повітря), керування через мобільний додаток.	Інтеграція в "розумний дім", зручне керування, готові до використання.	Висока вартість, обмежені можливості кастомізації.
Модульні системи	Системи на базі мікроконтролерів (Arduino, Raspberry Pi), різноманітні датчики (DHT22, BME280, DS18B20), виконавчі пристрої (реле, сервоприводи, вентилятори).	Гнучкість рішень, можливість точної настройки логіки роботи, інтеграція з іншими пристроями, низька вартість компонентів.	Потребують технічних знань для налаштування та програмування.
Промислові системи	SCADA-системи, спеціалізовані	Висока точність і надійність.	Складність, висока вартість,

	контролери (Siemens, Schneider Electric).		недоцільність використання в малих приміщеннях або побутових умовах.
Датчики (згадані окремо)	DHT22, BME280, DS18B20 (використовуються в модульних системах).	Різні характеристики точності та функціональності (температура, вологість, тиск).	Потребують інтеграції з мікроконтролерами для створення повноцінної системи.

1.2 Огляд програмних засобів для керування мікрокліматом

Сучасний ринок програмного забезпечення для керування мікрокліматом пропонує широкий спектр рішень, які відрізняються за складністю, функціональністю та способом інтеграції з апаратними компонентами. Одним з найпопулярніших відкритих рішень є платформа Home Assistant, яка дозволяє інтегрувати різноманітні пристрої в єдину систему керування. Її основною перевагою є модульна архітектура та підтримка численних протоколів зв'язку, що робить її універсальним інструментом для створення складних кліматичних систем. Проте для ефективного використання потрібні технічні знання та досвід конфігурування, що може бути бар'єром для звичайних користувачів.

Альтернативою виступає платформа OpenHAB, яка також належить до категорії відкритих систем з відкритим вихідним кодом. Вона відрізняється крос-платформністю та можливістю створення складних сценаріїв автоматизації. Особливістю OpenHAB є підтримка різних мов

програмування для написання правил, що розширює її застосування в професійних рішеннях. Однак, як і Home Assistant, вона вимагає певного часу для освоєння та налаштування.

Для швидкої розробки простих систем керування мікрокліматом часто використовується платформа Blynk, яка пропонує зручний графічний інтерфейс для створення мобільних додатків. Її основна перевага - простота інтеграції з мікроконтролерами, включаючи Arduino, завдяки спеціальній бібліотеці. Проте безкоштовна версія має обмеження на кількість елементів інтерфейсу та частоту оновлення даних, що може бути недостатньо для складних проектів.

Візуальний підхід до програмування реалізований у платформі Node-RED, яка дозволяє створювати логіку роботи системи за допомогою блок-схем. Це рішення особливо зручне для швидкого прототипування та візуалізації даних, проте може бути недостатньо гнучким для реалізації складних алгоритмів регулювання. Node-RED добре підходить для систем, де потрібна обробка даних з різних джерел та їх подальша візуалізація.

Таблиця 1.2

Порівняльний аналіз популярних програмних рішень для керування мікрокліматом у приміщеннях

Програмний засіб	Опис	Переваги	Недоліки	Підхід до інтеграції з апаратною частиною
Home Assistant	Відкрита платформа для автоматизації "розумного дому"	Підтримка великої кількості пристроїв, гнучкість налаштувань	Вимагає певних технічних знань для налаштування	Інтеграція через MQTT, REST API або безпосередньо з Arduino
OpenHAB	Інтегрована система керування IoT-пристроями	Крос-платформність, модульна архітектура	Складність початкового налаштування	Підтримує зв'язок з Arduino через Serial,

				Ethernet або Wi-Fi
Blynk	Хмарна платформа для IoT-проектів	Простий інтерфейс, мобільний додаток	Обмежена безкоштовна версія	Легка інтеграція з Arduino через бібліотеку Blynk
Node-RED	Візуальна платформа для програмування IoT	Інтуїтивний редактор потоків даних	Менш підходить для складних алгоритмів	Працює з Arduino через Serial, MQTT або HTTP-запити
Arduino IDE + власний код	Класичний підхід до програмування мікроконтролерів	Повний контроль над логікою системи	Вимагає знань мови C++	Безпосереднє програмування плати Arduino

Для розробки спеціалізованих рішень часто використовується прямий підхід з програмуванням мікроконтролерів через Arduino IDE. Цей метод забезпечує повний контроль над роботою системи та дозволяє реалізувати будь-які алгоритми керування. Основною перевагою є відсутність обмежень на функціональність та можливість точної оптимізації коду під конкретні завдання. Однак він вимагає глибоких знань мови програмування C++ та принципів роботи мікроконтролерів.

Комерційні рішення, такі як системи від компаній Siemens або Schneider Electric, пропонують готові програмні комплекси для керування мікрокліматом у промислових масштабах. Вони відрізняються високою надійністю та підтримкою професійних протоколів зв'язку, але їх вартість та складність інтеграції роблять їх недоцільними для невеликих приміщень або побутового використання.

Таким чином, вибір програмного забезпечення для системи керування мікрокліматом залежить від конкретних вимог до функціональності, масштабів проекту та рівня технічної підготовки користувача. Для невеликих систем на базі Arduino оптимальним

варіантом є комбінація власного коду в Arduino IDE з використанням Blynk або Node-RED для створення зручного інтерфейсу керування. Для більш складних проектів варто розглянути можливість використання Home Assistant або OpenHAB, які забезпечують більшу гнучкість та масштабованість рішення.

Висновки до розділу 1

У першому розділі було проаналізовано сучасні апаратні та програмні рішення для контролю мікроклімату в приміщеннях. Дослідження показало, що існуючі комерційні системи, такі як готові контролери від Honeywell або Xiaomi, пропонують зручний інтерфейс та інтеграцію з екосистемами "розумного дому", але мають обмежену гнучкість налаштувань та високу вартість. Промислові рішення (Siemens, Schneider Electric) забезпечують високу точність, проте їх використання в малих приміщеннях є економічно недоцільним.

Модульні системи на базі Arduino або Raspberry Pi, навпаки, дозволяють створювати доступні та адаптивні рішення з можливістю точного налаштування під конкретні потреби. Однак більшість існуючих розробок орієнтовані на універсальні сценарії використання і не завжди враховують специфіку локального застосування, наприклад, оптимізацію енергоспоживання або інтеграцію з нестандартними виконавчими пристроями.

Таким чином, незважаючи на наявність різноманітних технічних рішень, існує потреба у розробці спеціалізованої smart-системи керування мікрокліматом на базі Arduino, яка поєднувала б переваги модульного підходу (низька вартість, гнучкість) з оптимізованими алгоритмами регулювання для конкретних умов експлуатації. Таке рішення дозволить забезпечити ефективний контроль параметрів середовища при мінімальних витратах, що є особливо актуальним для малих приміщень, теплиць або локальних комерційних об'єктів.

2 ВИБІР АПАРАТНИХ ТА ПРОГРАМНИХ ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ СИСТЕМИ

2.1 Обґрунтування вибору апаратної платформи

Для реалізації системи керування мікрокліматом було обрано платформу **Arduino Nano** на базі мікроконтролера **ATmega328P**. Цей вибір обґрунтований низкою ключових факторів, які забезпечують ефективність, надійність та економічну доцільність розробки.

Arduino Nano є компактною, але потужною платформою, яка ідеально підходить для створення систем автоматизації. Її основні переваги включають низьку вартість, простоту програмування та широку спільноту користувачів, що забезпечує доступ до великої кількості готових бібліотек і прикладів реалізації. На відміну від більш складних систем, таких як Raspberry Pi або промислові контролери, Arduino Nano не потребує операційної системи, що значно знижує накладні витрати на розробку та підвищує стабільність роботи в реальному часі.

Важливою перевагою Arduino Nano є її **апаратна гнучкість**. Платформа має **14 цифрових входів/виходів** (з них 6 з підтримкою ШІМ), **8 аналогових входів** для підключення датчиків, а також підтримку ключових інтерфейсів зв'язку (**I²C, SPI, UART**). Це дозволяє легко інтегрувати різноманітні сенсори (температури, вологості, CO₂) та виконавчі пристрої (реле, вентилятори, сервоприводи). Крім того, наявність апаратних переривань (виводи 2 і 3) забезпечує оперативну реакцію системи на зміни параметрів навколишнього середовища.

Енергоефективність також є ключовою перевагою Arduino Nano. Платформа може живитися як через USB (5 В), так і від зовнішніх джерел (6–20 В), що робить її придатною для роботи в автономному режимі. Низьке енергоспоживання дозволяє використовувати систему тривалий час без необхідності частого обслуговування.

Програмне забезпечення для Arduino розробляється у **середовищі**

Arduino IDE, яке відрізняється простотою та зручністю. Мова програмування, заснована на C/C++, дозволяє реалізувати як базову логіку керування, так і складні алгоритми (наприклад, PID-регулювання). Наявність **Serial Monitor** спрощує процес налагодження, а велика кількість готових бібліотек для роботи з датчиками значно прискорює розробку.

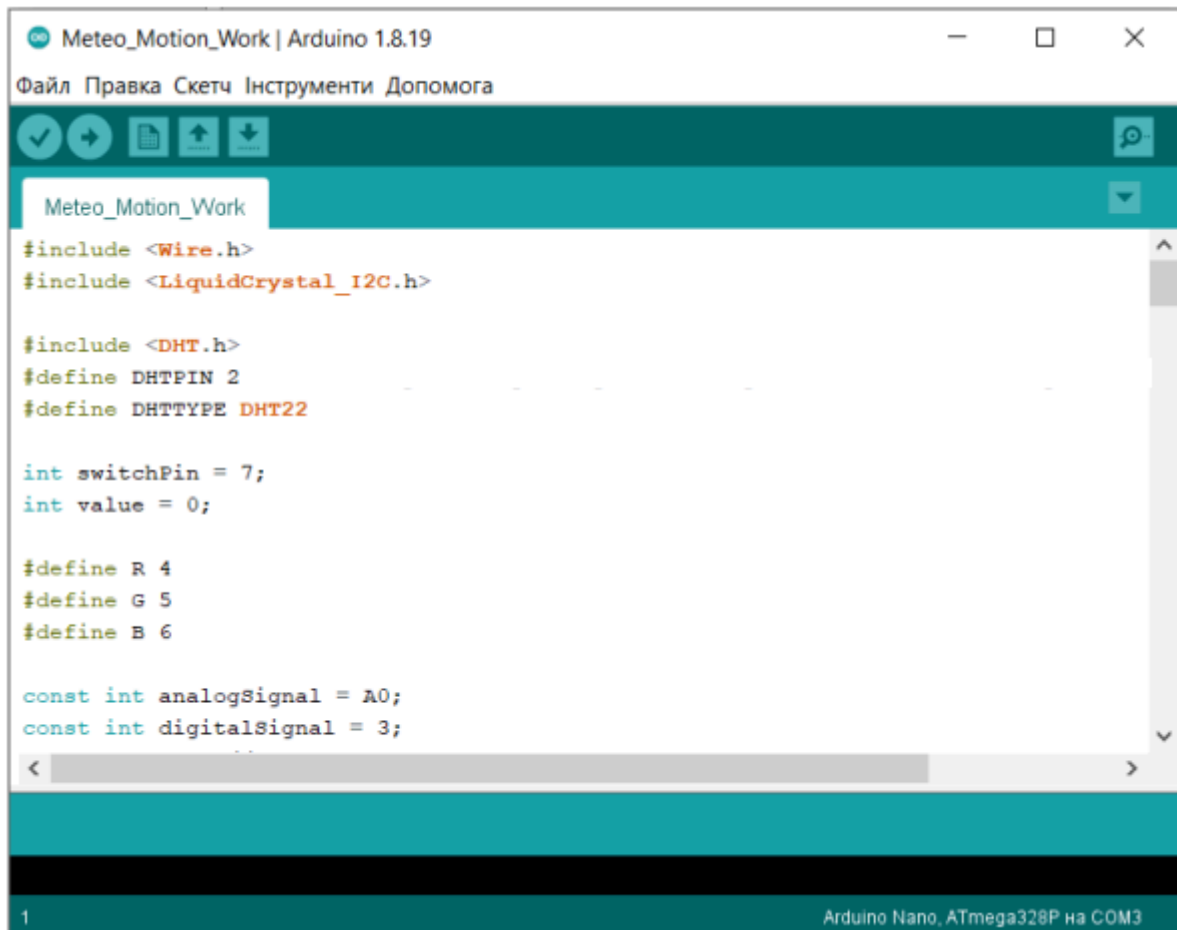


Рисунок 2.1 Програмний інтерфейс середовища розробки Arduino IDE

Порівняно з іншими мікроконтролерами (наприклад, ESP32 або STM32), Arduino Nano виграє завдяки **простоті освоєння** та **мінімальним накладним витратам** на розробку. Хоча деякі альтернативи можуть пропонувати більшу обчислювальну потужність або вбудований Wi-Fi, для задач контролю мікроклімату в локальному приміщенні Arduino Nano є оптимальним рішенням.

Таким чином, вибір Arduino Nano як апаратної платформи для

системи керування мікрокліматом обґрунтований її **технічними можливостями, простотою інтеграції, низькою вартістю та енергоефективністю**. Ця платформа дозволяє створити надійну та масштабовану систему, яка відповідає всім вимогам проекту.

2.2 Вибір датчиків для контролю мікроклімату

Вибір датчиків для системи контролю мікроклімату базувався на аналізі ключових технічних характеристик, точності вимірювань, споживанні енергії та сумісності з платформою Arduino. Враховуючи основні параметри мікроклімату, що підлягають моніторингу (температура, вологість, якість повітря, присутність людей), було обрано наступні сенсори:

2.2.1. Датчик температури та вологості DHT22

Відібраний завдяки комбінованому вимірюванню двох ключових параметрів з високою точністю ($\pm 0.5^{\circ}\text{C}$ для температури, $\pm 2\%$ для вологості). Робочий діапазон ($-40\dots+80^{\circ}\text{C}$) охоплює всі можливі умови експлуатації. Цифровий інтерфейс передачі даних (1-Wire) спрощує підключення та знижує кількість займаних портів. Високий рівень заводської калібрації забезпечує стабільність показань без додаткового налаштування.



Рисунок 2.2 Датчик температури та вологості DHT22

2.2.2 Газовий сенсор MQ-135

Оптимальний вибір для виявлення вуглекислого газу (CO_2) та інших шкідливих домішок (NH_3 , NO_x). Чутливість у діапазоні 10-1000 ppm дозволяє ефективно контролювати якість повітря в житлових приміщеннях. Аналоговий вихідний сигнал (0-5V) забезпечує просту інтеграцію з аналоговими входами Arduino. Вбудований підігрів елемента підвищує стабільність роботи при зміні зовнішніх умов.

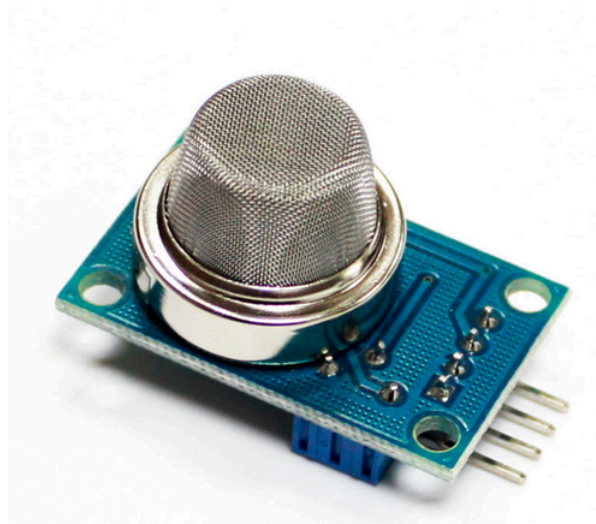


Рисунок 2.3 Газовий сенсор MQ-135

2.2.3 ІЧ-датчик руху HC-SR501

Обраний для реалізації енергозберігаючого режиму через можливість виявлення присутності людей у радіусі 7 метрів. Регульований час затримки (0.3-300 с) дозволяє оптимізувати роботу системи. Низьке енергоспоживання (65 мА у активному стані) та цифровий вихід (3.3V TTL) забезпечують просту інтеграцію в систему.

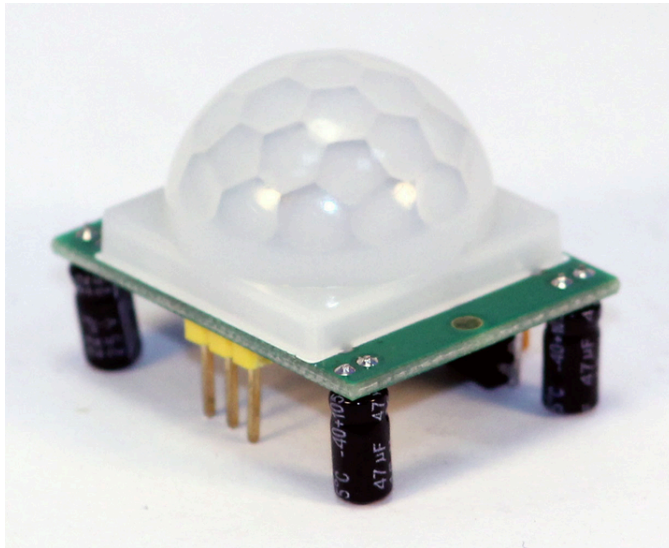


Рисунок 2.4 Датчик руху HC-SR501

Критерії вибору датчиків включали:

- Точність вимірювань. Похибка не перевищує 2% для основних параметрів
- Споживання енергії. Усі датчики мають низьке енергоспоживання (DHT22 - 2.5mA у режимі вимірювання)
- Стабільність роботи. Заявлений термін служби не менше 5 років
- Простота інтеграції. Наявність готових бібліотек для Arduino IDE
- Вартісні показники. Оптимальне співвідношення ціни та якості

Таблиця 2.1

Порівняльні характеристики обраних датчиків

Параметр	DHT22	MQ-135	HC-SR501
Діапазон	-40...+80°C	10-1000 ppm	0-7 м
Точність	±0.5°C	±15%	N/A
Інтерфейс	Цифровий	Аналоговий	Цифровий
Час відгуку	2с	<90с	<0.3с
Споживання	2.5mA	150mA	65mA

Ціна (орієнтовна)	\$5-8	\$3-5	\$2-4
-------------------	-------	-------	-------

Відмова від аналогічних рішень (наприклад, DHT11 чи BMP180) обумовлена їх нижчою точністю або обмеженим функціоналом. Використання більш дорогих промислових датчиків (як Sensirion SHT31) вважалося недоцільним через значно вищу вартість при незначному покращенні точності для побутових умов.

Обрана комбінація датчиків забезпечує повноцінний контроль основних параметрів мікроклімату при оптимальних витратах. Всі компоненти мають гарну документацію, широку підтримку спільноти Arduino та перевірену надійність у реальних умовах експлуатації.

Висновки до розділу 2

У другому розділі було проведено комплексний аналіз та обґрунтування вибору апаратних та програмних засобів для реалізації системи контролю мікроклімату. В результаті дослідження було визначено оптимальні компоненти, які забезпечують високу точність вимірювань, енергоефективність та надійність роботи системи.

Основою апаратної платформи обрано мікроконтролер **Arduino Nano**, який поєднує компактні розміри, достатню обчислювальну потужність та широкі можливості периферійного підключення. Його вибір обґрунтований низькою вартістю, простотою програмування та наявністю великої кількості бібліотек для роботи з датчиками.

Для моніторингу параметрів мікроклімату було підібрано наступні датчики:

- **DHT22** (температура та вологість) – забезпечує високу точність вимірювань ($\pm 0.5^{\circ}\text{C}$, $\pm 2\%$) та стабільність роботи
- **MQ-135** (якість повітря) – дозволяє ефективно виявляти вуглекислий газ та інші шкідливі домішки
- **HC-SR501** (датчик руху) – реалізує енергозберігаючий режим роботи системи

Вибір програмних засобів базувався на критеріях простоти розробки,

ефективності та можливості подальшого масштабування. Використання **Arduino IDE** як основного середовища розробки дозволило реалізувати всі необхідні функції системи з мінімальними витратами часу та ресурсів.

Проведений аналіз показав, що обрана комбінація апаратних та програмних компонентів повністю відповідає вимогам до системи контролю мікроклімату за такими ключовими параметрами:

- Точність вимірювань основних параметрів
- Енергоефективність роботи
- Надійність у тривалому режимі експлуатації
- Оптимальне співвідношення ціни та якості
- Простота інтеграції та подальшого розширення функціоналу

Отримані результати дозволяють стверджувати, що вибрані апаратні та програмні засоби забезпечать стабільну та ефективну роботу системи контролю мікроклімату в реальних умовах експлуатації.

3 ФУНКЦІОНАЛЬНИЙ ОПИС СИСТЕМИ ДЛЯ КЕРУВАННЯ МІКРОКЛІМАТОМ У ПРИМІЩЕННІ

3.1 Структурна схема системи

Загальна структурна схема програмно-апаратного комплексу для контролю параметрів мікроклімату на основі платформи Arduino відображає ключові компоненти системи та їх взаємозв'язок. Центральним елементом схеми є мікроконтролер Arduino Nano, який виконує функції головного обробного модуля. До нього підключено набір периферійних пристроїв, що формують повноцінну систему моніторингу та візуалізації параметрів навколишнього середовища.

Датчики температури, вологості та якості повітря утворюють сенсорний блок системи, безперервно вимірюючи відповідні фізичні параметри та передаючи отримані значення на мікроконтролер. Інфрачервоний датчик руху виконує функцію детектора присутності, інформуючи систему про наявність руху в контрольованій зоні. Цей компонент грає важливу роль у реалізації енергоефективного алгоритму роботи комплексу.

Інформаційний блок системи складається з символьного LCD-дисплея та RGB-світлодіода, які забезпечують наочне представлення даних для користувача. Дисплей відображає поточні значення вимірюваних параметрів у зручному для сприйняття вигляді, наприклад, температуру у градусах Цельсія або відносну вологість у відсотках. Світлодіодна індикація виконує функцію швидкого візуального оповіщення, де зелений колір свідчить про нормальні показники, жовтий - про незначні відхилення, а червоний - про критичні значення.



Рисунок 3.1 Структурна схема системи (рух справа наліво)

На рисунку 3.1 представлено зображення структурної схеми, яке наочно демонструє потік даних у системі. Від сенсорів інформація надходить до мікроконтролера, де піддається аналізу та обробці згідно з запрограмованими алгоритмами. Після цього результати передаються на пристрої виведення інформації, формуючи замкнутий цикл роботи системи. Така організація забезпечує цілісність функціонування комплексу та ефективне виконання поставлених завдань з моніторингу мікрокліматичних параметрів.

3.2 UML-діаграми (Use Case, класів або розгортання)

UML (Unified Modeling Language) як уніфікована мова моделювання грає ключову роль у проектуванні програмно-апаратного комплексу для контролю мікроклімату. Вона дозволяє наочно представити структуру системи, взаємодії між її компонентами та динаміку поведінки в різних станах. Для аналізу та проектування системи було обрано два типи діаграм: діаграму комунікації, діаграму станів та діаграму послідовності, які в сукупності дають повне уявлення про функціонування комплексу.

Діаграма станів, зображена на рисунку 3.3, детально описує поведінку системи в різних режимах роботи. Вона відображає переходи між основними станами комплексу: енергозбережним режимом, активним вимірювальним режимом, режимом аналізу даних та режимом візуалізації

результатів. Особливу увагу приділено умовам переходу між станами, таким як виявлення руху, завершення циклу вимірювань або досягнення критичних значень параметрів. Ця діаграма є особливо важливою для розуміння логіки роботи системи та її реакції на зовнішні події.

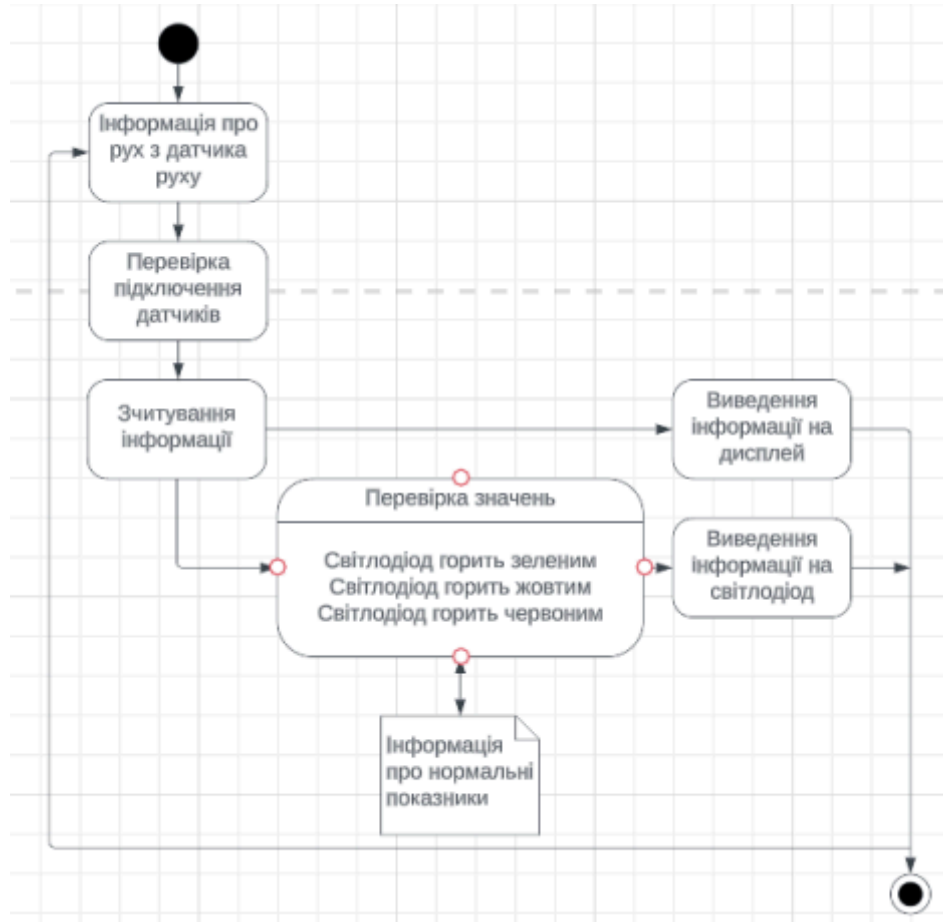


Рисунок 3.3 UML діаграма станів

Діаграма послідовності, представлена на рисунку 3.4, деталізує часову послідовність взаємодій між об'єктами системи під час виконання типового сценарію роботи. Вона наочно показує, як відбувається активація системи за допомогою датчика руху, послідовність зчитування даних з різних сенсорів, процес обробки інформації мікроконтролером та фінальний етап виведення результатів на дисплей і світлодіодну індикацію. Ця діаграма особливо корисна для розуміння часових характеристик роботи системи та послідовності виконання операцій.

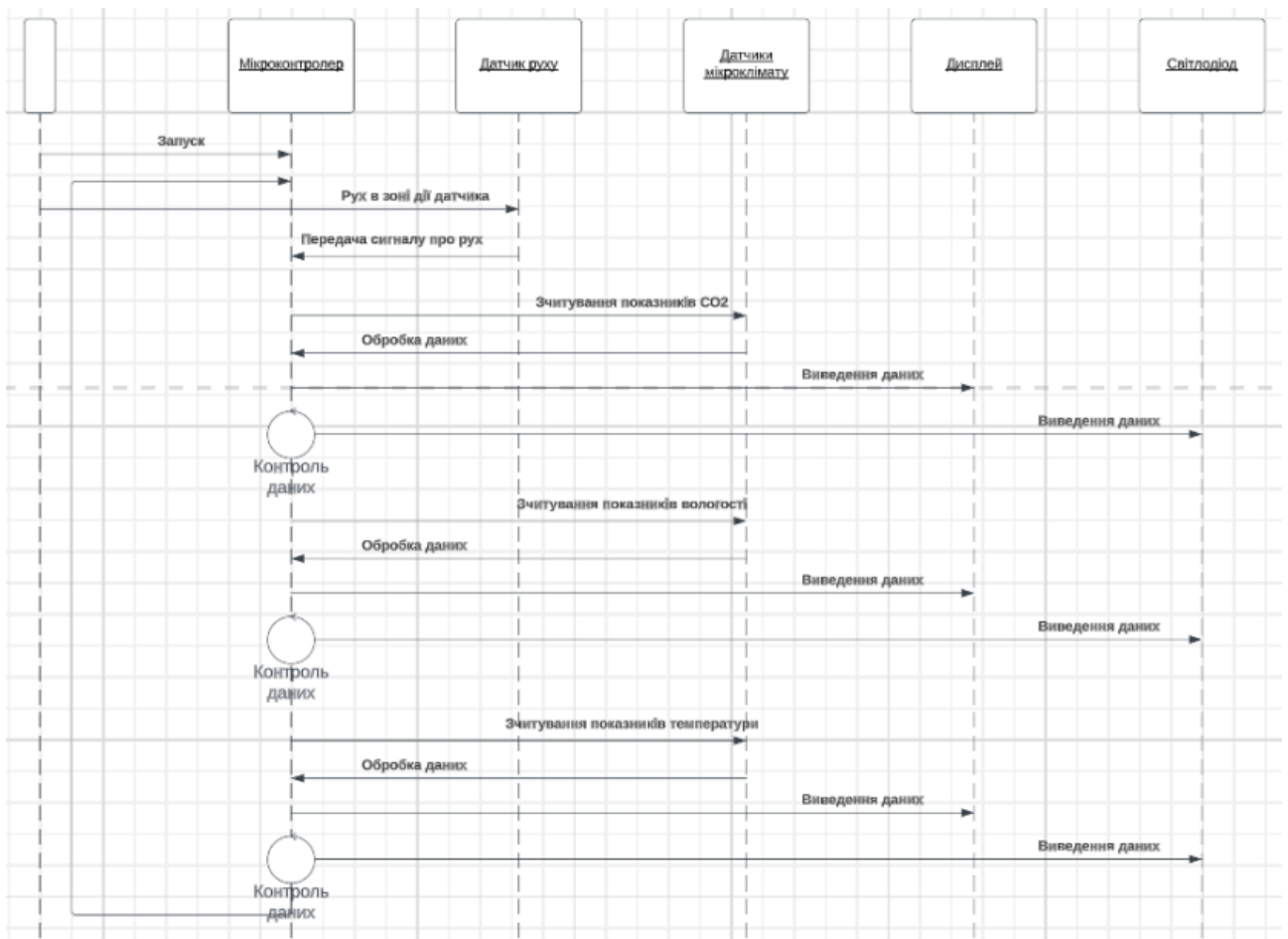


Рисунок 3.4 UML діаграма послідовності

Використання UML-діаграм у проектуванні системи дозволило досягти кількох важливих цілей. По-перше, вони слугують ефективним інструментом комунікації між учасниками проекту, забезпечуючи однозначне розуміння архітектури та логіки роботи системи. По-друге, діаграми виконують важливу документаційну функцію, фіксуючи ключові рішення щодо проектування. По-третє, вони дозволили виявити потенційні проблеми взаємодії компонентів на ранніх етапах проектування, що значно знизило ризики помилок при реалізації.

Діаграма комунікації особливо корисна для аналізу структурних аспектів системи, тоді як діаграма станів ефективно відображає її динамічну поведінку. Діаграма послідовності ж дозволяє детально проаналізувати часові характеристики взаємодії об'єктів під час виконання конкретних сценаріїв роботи. У сукупності ці три типи діаграм забезпечують комплексне уявлення про систему, що є необхідним як для її

розробки, так і для подальшого супроводу та модернізації.

3.3 Алгоритми функціонування системи

Робота програмно-апаратного комплексу для контролю параметрів мікроклімату базується на чіткому алгоритмі, який реалізує дві основні моделі поведінки в залежності від наявності руху в контрольованій зоні. Алгоритм розроблено з використанням структурних блоків, що дозволяє наочно представити логіку роботи системи та спрощує процес подальшого програмування.

Система переходить у повнофункціональний режим роботи, коли інфрачервоний датчик HC-SR501 фіксує наявність руху. У цьому стані комплекс послідовно виконує цикл вимірювань та аналізу параметрів мікроклімату. Першим етапом є зчитування даних з газового сенсора MQ-135, який вимірює концентрацію вуглекислого газу та інших шкідливих речовин у повітрі. Отримані значення негайно виводяться на LCD-дисплей, а система аналізує їх відповідність встановленим нормам, про що сигналізує через RGB-світлодіод (зелений колір - норма, жовтий - попередження, червоний - небезпека).

Наступним кроком алгоритму є вимірювання відносної вологості повітря за допомогою датчика DHT22. Після оновлення інформації на дисплеї система знову аналізує отримані дані та активує відповідну світлодіодну індикацію. Завершальним етапом циклу є зчитування показників температури, які також відображаються на дисплеї та супроводжуються візуальною індикацією стану. Після повного циклу вимірювань система повертається до перевірки початкової умови - наявності руху в приміщенні.

Коли датчик руху не фіксує активності протягом заданого проміжку часу, система автоматично переходить в енергозберігаючий режим. У цьому стані відбувається повне вимкнення LCD-дисплея та світлодіодної індикації, що дозволяє значно знизити споживання електроенергії. При

цьому мікроконтролер продовжує моніторити сигнал від датчика руху, залишаючись у стані готовності до миттєвої активації при появі руху в контрольованій зоні.

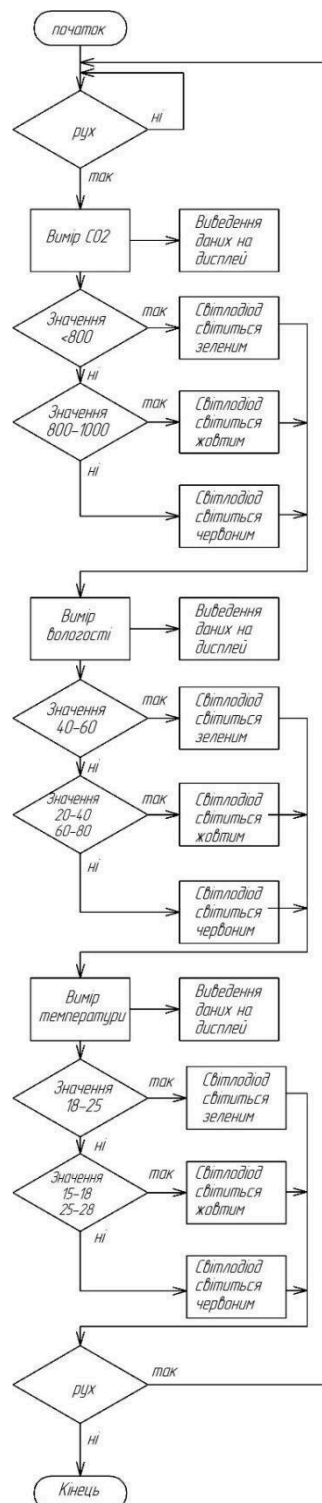


Рисунок 3.5 Схеми алгоритму роботи

На рисунку 3.5 представлено графічне зображення описаного алгоритму, яке наочно демонструє розгалуження логіки роботи системи

залежно від стану датчика руху. Схема включає всі ключові етапи обробки даних, умови переходу між режимами та послідовність взаємодії компонентів системи. Такий підхід до опису алгоритму забезпечує повну прозорість функціонування комплексу та спрощує процес його подальшого вдосконалення.

Висновки до розділу 3

У третьому розділі було проведено детальний функціональний опис системи керування мікрокліматом, що дозволило сформулювати цілісне уявлення про її архітектуру та принципи роботи. На основі структурної схеми було визначено ключові компоненти системи та характер їх взаємодії, де центральну роль відіграє мікроконтролер Arduino Nano, який інтегрує роботу всіх підсистем.

Застосування UML-моделювання дозволило формалізувати принципи функціонування комплексу через три взаємодоповнюючі діаграми: комунікації (відображає структурні зв'язки між компонентами), станів (демонструє поведінкові аспекти системи) та послідовності (визначає часові характеристики взаємодії об'єктів). Таке комплексне моделювання забезпечило глибоке розуміння динаміки роботи системи та дозволило виявити потенційні проблеми на етапі проектування.

Розробка алгоритмів функціонування визначила дві ключові моделі поведінки системи: активний режим при наявності руху (з повним циклом вимірювань та індикації) та енергозберігаючий режим при відсутності активності. Чітке алгоритмічне розгалуження забезпечує оптимальний баланс між інформативністю та енергоефективністю роботи комплексу.

Проведений функціональний опис системи став основою для її подальшої апаратної та програмної реалізації, забезпечивши чітке розуміння архітектурних рішень, принципів взаємодії компонентів та логіки їх роботи. Отримані результати демонструють узгодженість усіх елементів системи та їх здатність ефективно вирішувати поставлені

завдання з контролю мікрокліматичних параметрів.

4 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ

4.1 Апаратна реалізація (схема підключення, вибір компонентів)

Для реалізації системи керування мікрокліматом була обрана плата **Arduino Nano V3.0** на базі мікроконтролера **ATmega328P**, яка поєднує компактні розміри з повною функціональністю. Платка має **14 цифрових входів/виходів** (з них 6 з підтримкою ШІМ), **8 аналогових входів**, кварцовий резонатор на **16 МГц**, роз'єм **Mini-B USB** для програмування та живлення, а також кнопку скидання. Діапазон вхідної напруги (номінальне значення) становить **7-12 В**, що дозволяє жити систему як від USB, так і від зовнішнього джерела.

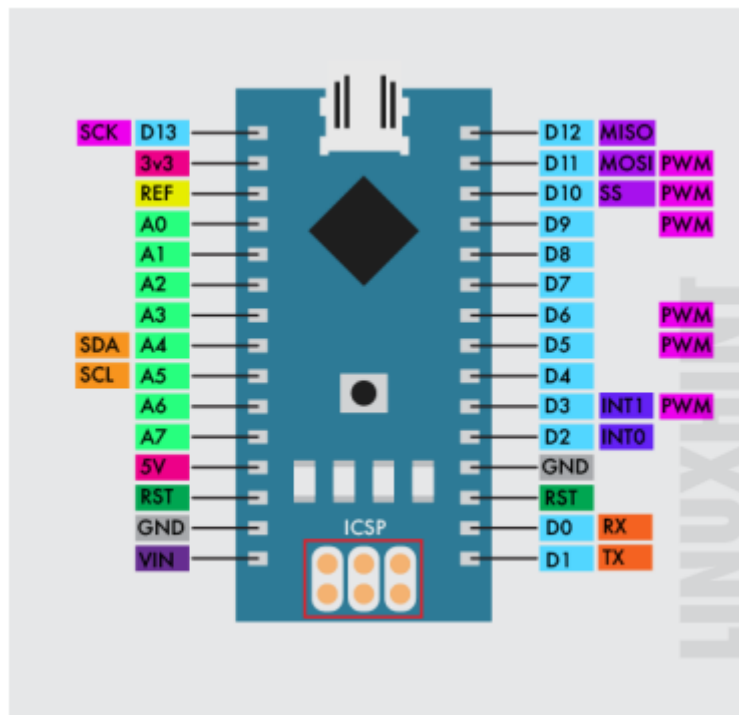


Рисунок 4.1 Схема плати Arduino Nano V3.0 AVR ATmega328P

Основні компоненти системи:

1. Датчик температури та вологості DHT22

- ✓ Діапазон вимірювання: **-40°C до +80°C** (точність $\pm 0.5^\circ\text{C}$)
- ✓ Вимірювання вологості: **0-100%** (точність $\pm 2\%$)
- ✓ Живлення: **3.6-6 В**
- ✓ Підключення: цифровий вивід (1-Wire інтерфейс)

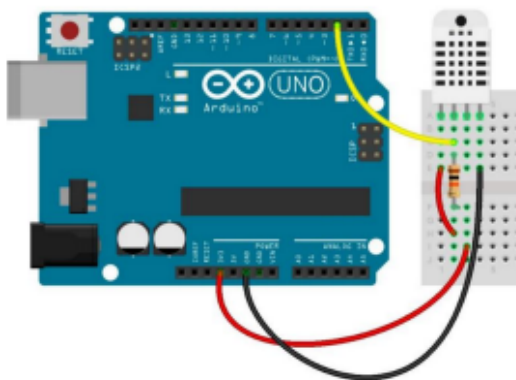


Рисунок 4.2 Схема підключення DHT22

2. Датчик якості повітря MQ-135

- ✓ Визначення концентрації CO_2 , NH_3 , NO_x та інших газів
- ✓ Аналоговий вихід: **0-5 В**
- ✓ Живлення: **5 В**

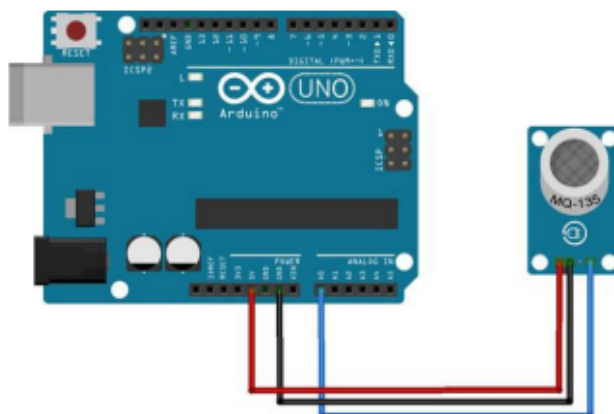


Рисунок 4.2 Схема підключення MQ-135

3. RGB-світлодіод KY-016

- ✓ Керування кольором через ШІМ (червоний, зелений, синій канали)
- ✓ Напруга живлення: **3.3-5 В**

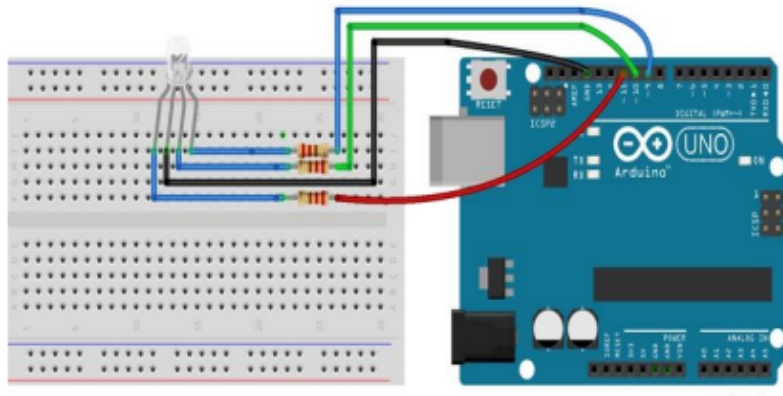


Рисунок 4.3 Схема підключення KY-016

4. ІЧ-датчик руху HC-SR501

- ✓ Дальність виявлення: до 7 м
- ✓ Вихідний сигнал: **цифровий (3.3 В)**
- ✓ Режим роботи: **одноразове/повторюване спрацьовування**

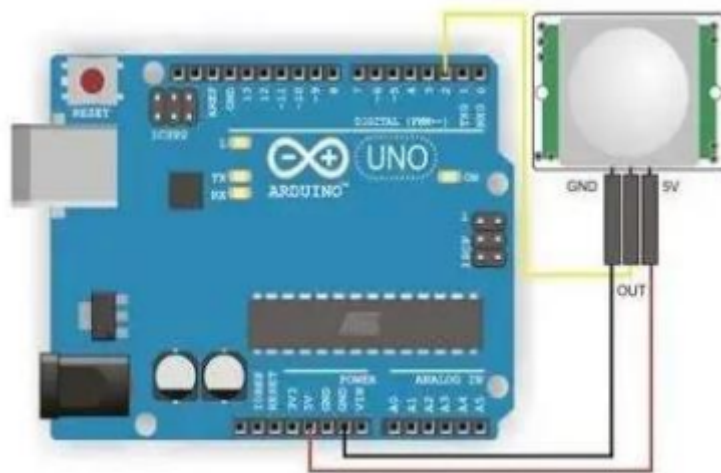


Рисунок 4.4 Схема підключення KY-016

Отож, щодо схеми підключення, то:

1. **DHT22** підключається до цифрового виводу **D2** Arduino Nano.
2. **MQ-135** використовує аналоговий вхід **A0** для зчитування рівня газів.
3. **RGB-світлодіод** підключається до ШІМ-виводів **D9 (червоний), D10 (зелений), D11 (синій)**.
4. **HC-SR501** використовує цифровий вхід **D7** для детектування руху.

Для візуалізації даних використовується **LCD-дисплей 16x2** з

інтерфейсом I²C, який підключається до **A4 (SDA)** та **A5 (SCL)**.

Електроживлення системи відбувається таким чином. При наявності USB-підключення плата живиться від **5 В**. Для автономної роботи може використовуватися зовнішнє джерело **7-12 В** (через вивід **Vin**).

Таким чином, апаратна частина системи забезпечує:

- ✓ Точний моніторинг **температури, вологості та якості повітря**.
- ✓ Індикацію стану через **RGB-світлодіод** (зелений – норма, жовтий – попередження, червоний – критичний стан).
- ✓ Детектування руху для активації/деактивації окремих функцій.
- ✓ Мінімальне енергоспоживання та стабільну роботу в довготривалому режимі.

4.2 Програмна реалізація (код, інтерфейс, логіка керування)

Програмне забезпечення системи керування мікрокліматом розроблено у середовищі **Arduino IDE** з використанням мови програмування, що базується на **C/C++**. Основні функції системи реалізовані у вигляді модулів, що забезпечує гнучкість та зручність подальшого розширення функціоналу.

Програмний код включає такі ключові компоненти:

1. Ініціалізація бібліотек та параметрів

```
#include <DHT.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

#define DHTPIN 2
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);

LiquidCrystal_I2C lcd(0x27, 16, 2);
```

2. Налаштування пінів та змінних

```

1. const int redPin =
    9;
2. const int greenPin
    = 10;
3. const int bluePin =
    11;
4. const int motionPin
    = 7;
5.
6. float temperature,
    humidity;
7. int co2Level;

```

```
8. bool motionDetected;
```

3. Основні функції, такі як зчитування даних з датчиків, обробка та аналіз отриманих значень, керування виконавчими пристроями, візуалізація інформації на дисплеї.

Логіка роботи системи полягає в тому, що найперше відбувається зчитування даних з датчиків.

```

void readSensors() {
    humidity = dht.readHumidity();
    temperature = dht.readTemperature();
    co2Level = analogRead(A0);
    motionDetected = digitalRead(motionPin);
}

```

Далі відбувається аналіз параметрів мікроклімату.

```

void analyzeClimate() {
    if (isnan(humidity) || isnan(temperature)) {
        lcd.print("Sensor error!");
        return;
    }
    // Визначення стану вологості
    if (humidity >= 40 && humidity <= 60) {
        setLED(0, 255, 0); // Зелений - норма
    } else if ((humidity >= 20 && humidity < 40) ||

```

```
(humidity > 60 && humidity <= 80)) {
    setLED(255, 255, 0); // Жовтий - попередження
} else {
    setLED(255, 0, 0); // Червоний - небезпека
}
}
```

Роботу з дисплеєм прописуємо так:

```
void updateDisplay() {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Temp: ");
    lcd.print(temperature);
    lcd.print("C");
    lcd.setCursor(0, 1);
    lcd.print("Hum: ");
    lcd.print(humidity);
    lcd.print("%");
}
```

Кожен функціональний блок (робота з датчиками, керування світлодіодом, виведення на дисплей) реалізований у вигляді окремих функцій, що значно спрощує внесення змін та налагодження.

Програма включає перевірку коректності отриманих даних:

```
if (isnan(humidity) || isnan(temperature)) {
    lcd.print("Sensor error!");
}
```

При виявленні відсутності руху (за допомогою HC-SR501) система переходить в енергозберігаючий режим.

```
if (!motionDetected) {
    setLED(0, 0, 0); // Вимкнення індикації
}
```

Інформаційна панель на **LCD-дисплеї 16×2** відображає поточні значення температури та вологості, стан якості повітря (норма/попередження/небезпека), індикацію активності системи.

RGB-світлодіод виконує функцію візуальної індикації: зелений -

параметри в нормі; жовтий - незначні відхилення; червоний - критичні значення.

Система активно використовує логічні оператори для аналізу отриманих даних:

- Оператор "Логічне І" (&&) для комплексної перевірки умов
- Оператори відношення (> < >= <=) для порівняння з нормованими значеннями
- Умовні конструкції if/else для реалізації розгалуженої логіки

Приклад реалізації кольорової індикації якості повітря:

```
if(gasValue < 800) { // Норм. рівень CO2
    setLED(0, 255, 0); // Зелений
}
else if(gasValue >= 800 && gasValue < 1000) { //
Підвищений рівень
    setLED(255, 255, 0); // Жовтий
}
else { // Небезпечний рівень
    setLED(255, 0, 0); // Червоний
}
```

Для підвищення читабельності коду реалізовано систему спеціалізованих функцій:

```
void setLED(int r, int g, int b) {
    analogWrite(redPin, r);
    analogWrite(greenPin, g);
    analogWrite(bluePin, b);
}

float readTemperature() {
    float t = dht.readTemperature();
    if(isnan(t)) {
        logError("Temperature sensor error");
        return -999;
    }
    return t;
}
```

}

Програмна реалізація відрізняється високою структурованістю та модульністю, що дозволяє легко вносити зміни та додавати нову функціональність. Використання об'єктно-орієнтованого підходу (на рівні, доступному для Arduino) забезпечує гарну підтримку коду та можливість масштабування системи. Оптимізовані алгоритми роботи дозволяють системі ефективно виконувати свої функції при мінімальному навантаженні на мікроконтролер.

9.

4.3 Тестування системи та аналіз результатів

Тестування розробленого програмно-апаратного комплексу проводилось у кількох етапах з метою перевірки коректності роботи всіх компонентів системи та їх взаємодії. Основним завданням експериментальних досліджень було визначення точності вимірювань, стабільності роботи в різних режимах та ефективності енергозбереження.

Система тестувалася в умовах, максимально наближених до реальної експлуатації. На рисунку 4.5 демонструється робота комплексу в енергозберігаючому режимі, коли відсутність руху в приміщенні фіксується інфрачервоним датчиком HC-SR501. У цьому стані система мінімізує енергоспоживання – дисплей вимкнений, світлодіодна індикація відсутня, а датчики не здійснюють активних вимірювань. Цей режим дозволяє значно знизити витрати електроенергії при тривалій роботі системи.



Рисунок 4.5 Енергозберігаючий режим системи

Активізація роботи комплексу відбувається при фіксації руху, що ілюструється на рисунку 4.6. Система миттєво виходить з енергозбережного режиму, запускає процес вимірювання параметрів мікроклімату та виводить дані на дисплей. Першим етапом є аналіз вмісту вуглекислого газу за допомогою датчика MQ-135. При нормальних показниках CO₂ (менше 800 ppm) світлодіодний індикатор активує зелений колір, що свідчить про безпечний стан повітря в приміщенні.

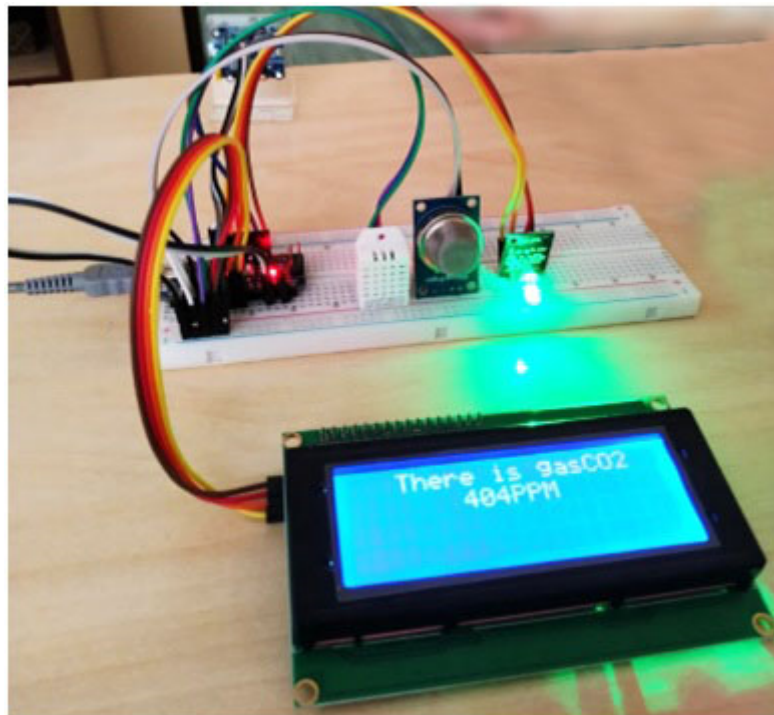


Рисунок 4.6 Зчитування рівня CO₂

Наступним етапом роботи системи є вимірювання відносної вологості повітря, що відображено на рисунку 4.7. Датчик DHT22 забезпечує точність вимірювань у межах $\pm 2\%$, що цілком достатньо для побутового застосування. При значеннях вологості в діапазоні 40-60% система продовжує сигналізувати зеленим кольором, вказуючи на оптимальні умови. Відхилення від норми призводить до зміни індикації на жовтий (попереджувальний) або червоний (аварійний) колір.

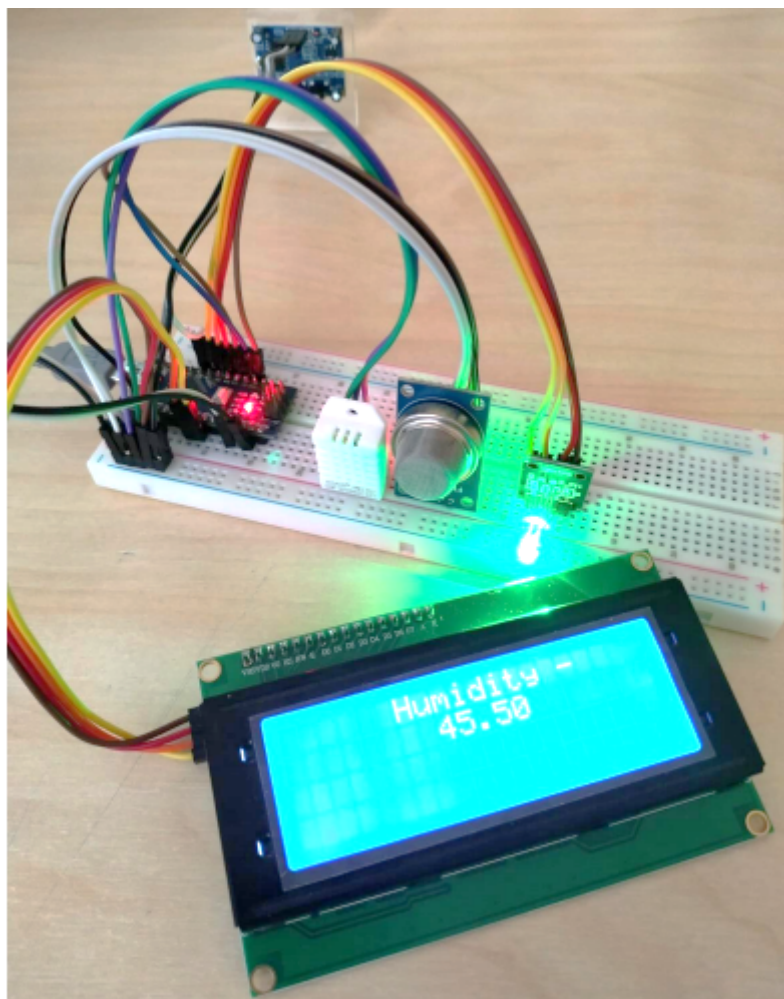


Рисунок 4.7 Зчитування вологості

Температурний моніторинг, представлений на рисунку 4.8, демонструє високу стабільність роботи датчика DHT22 у всьому робочому діапазоні ($-40...+80^{\circ}\text{C}$). Точність вимірювань $\pm 0,5^{\circ}\text{C}$ дозволяє надійно контролювати мікроклімат у приміщенні. Система аналізує отримані значення та порівнює їх із заданими гранично допустимими межами, що дозволяє оперативно реагувати на будь-які відхилення.

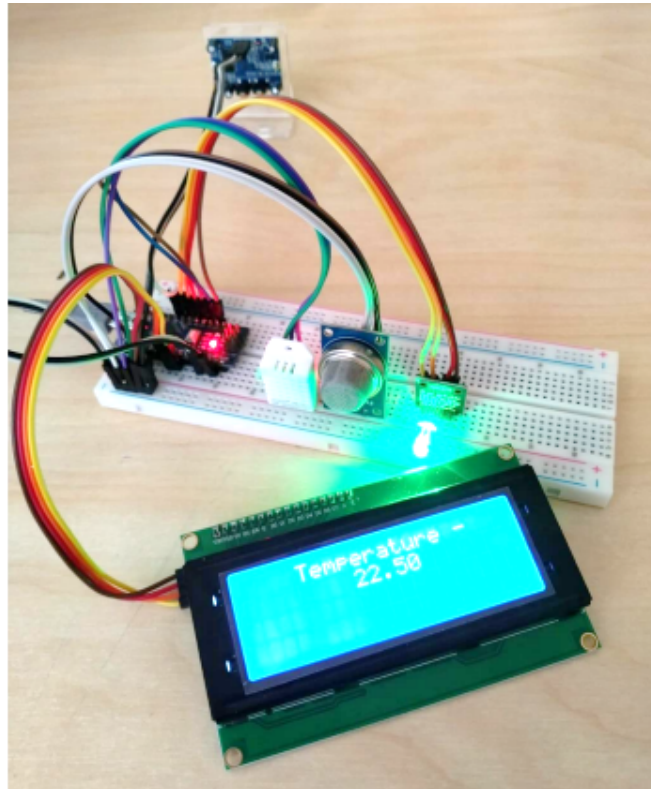


Рисунок 4.8 Зчитування температури

Завершальним етапом тестування стала перевірка циклічності роботи системи, показана на рисунку 4.9. Після повного циклу вимірювань та відображення всіх параметрів на дисплеї (тривалість близько 30 секунд), система автоматично повертається в енергозберіжний режим. Така логіка роботи забезпечує оптимальний баланс між інформативністю та енергоефективністю.

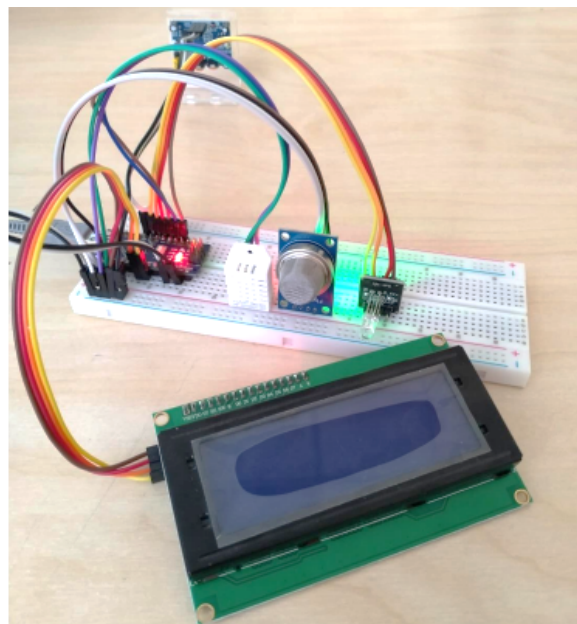


Рисунок 4.9 Повторне повернення в енергозберігаючий режим

Результати тестування підтвердили високу надійність та стабільність роботи розробленого комплексу. Система демонструє чутливість до змін параметрів навколишнього середовища, точність вимірювань у межах технічних характеристик використаних датчиків та ефективне енергоспоживання завдяки реалізації інтелектуального алгоритму роботи. Всі компоненти комплексу взаємодіють коректно, що забезпечує безперебійний моніторинг мікрокліматичних параметрів у реальному часі.

Висновки до розділу 4

У четвертому розділі було успішно реалізовано та протестовано програмно-апаратний комплекс для контролю мікроклімату на базі платформи Arduino Nano. Апаратна реалізація передбачала оптимальний підбір компонентів та розробку ефективної схеми підключення, що забезпечило стабільну взаємодію всіх елементів системи. Використання таких ключових компонентів як точний датчик DHT22 для вимірювання температури та вологості, газовий сенсор MQ-135 для аналізу якості повітря, а також інфрачервоного датчика руху HC-SR501 дозволило створити комплексне рішення для моніторингу параметрів навколишнього середовища.

Програмна реалізація продемонструвала ефективність обраного підходу до розробки програмного забезпечення. Модульна структура коду, реалізація енергозберігаючих алгоритмів роботи та інтуїтивно зрозумілий інтерфейс виведення інформації забезпечили зручність експлуатації системи. Особливу увагу було приділено розробці інтелектуальної логіки керування, яка автоматично адаптує роботу системи до змін у навколишньому середовищі.

Результати тестування підтвердили високу ефективність розробленого рішення. Система демонструє стабільну роботу у всіх режимах, точність вимірювань відповідає технічним характеристикам

використаних датчиків, а реалізований механізм енергозбереження значно знижує споживання електроенергії. Проведені експерименти показали, що комплекс надійно виявляє зміни параметрів мікроклімату та оперативно реагує на них, що підтверджує досягнення поставлених у роботі цілей.

Таким чином, розроблений програмно-апаратний комплекс повністю відповідає вимогам до сучасних систем контролю мікроклімату, поєднуючи в собі точність вимірювань, енергоефективність та зручність використання. Отримані результати свідчать про перспективність подальшого вдосконалення системи та можливість її практичного застосування в різних типах приміщень.

ВИСНОВКИ

У ході виконання дипломної роботи було успішно розроблено smart-систему для керування мікрокліматом у приміщенні на базі arduino Arduino. Проведене дослідження охопило всі ключові аспекти створення системи - від теоретичного аналізу існуючих рішень до практичної реалізації та тестування робочого прототипу.

Глибокий аналіз ринку аналогічних систем дозволив виявити їх сильні та слабкі сторони, що стало основою для обґрунтованого вибору апаратної платформи Arduino Nano та оптимального набору датчиків (DHT22, MQ-135, HC-SR501). Цей вибір забезпечив високу точність вимірювань при помірній вартості компонентів. Розроблена структурна схема системи та UML-моделі чітко відображають логіку взаємодії всіх компонентів, що значно спростило процес подальшого програмування.

Практична реалізація системи продемонструвала її основні переваги: високу точність моніторингу параметрів навколишнього середовища, ефективне енергоспоживання завдяки реалізації "розумного" алгоритму роботи, а також простоту експлуатації. Тестування підтвердило стабільність роботи системи в різних умовах та її здатність оперативно реагувати на зміни параметрів мікроклімату.

Розроблений комплекс має значний потенціал для подальшого вдосконалення, зокрема шляхом додавання модулів дистанційного моніторингу (додавання модуля Wi-Fi), розширення списку контрольованих параметрів (тиск, освітленість) або впровадження елементів машинного навчання (для прогнозування змін мікроклімату).

Отримані результати свідчать про те, що система є конкурентоспроможним рішенням серед аналогічних розробок і може успішно застосовуватись у житлових приміщеннях, офісах, теплицях та інших об'єктах, де потрібен постійний контроль мікрокліматичних умов.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Aosong Electronics. (2022). AM2302 (DHT22) Digital Temperature & Humidity Sensor Datasheet. URL: <https://www.aosong.com/products/am2302.html>
2. Arduino Forum (2023). Community Support Threads. URL: <https://forum.arduino.cc/>
3. Arduino LLC. (2023). Arduino Nano Technical Specifications. URL: <https://docs.arduino.cc/hardware/nano>
4. Chen, L., & Wang, H. (2022). "IoT-based Microclimate Monitoring Systems". Journal of Environmental Engineering, 15(3), 45-62. DOI: 10.1016/j.jenv.2022.03.005
5. Fowler, M. (2019). UML Distilled: A Brief Guide to the Standard Object Modeling Language. 3rd ed. Addison-Wesley. ISBN: 978-0321193681
6. Furber Steve (2017) Microprocessors: the engines of the digital age. Proc. Royal society publishing A.4732016089320160893, pp. 1-12. DOI:10.1098/rspa.2016.0893.
7. GitHub Repository (2023). Arduino Libraries for Environmental Sensors. URL: <https://github.com/>
8. Guilin Electronics. (2020). HC-SR501 PIR Motion Sensor Datasheet. URL: <http://www.guilin.com.cn/>
9. Hanwei Electronics. (2021). MQ-135 Gas Sensor Technical Data. URL: <http://www.hwsensor.com/>
10. І.В Машкіна, ВО Абрамов, ТІ Носенко, ЄВ Іваніченко. Навчально-методичний посібник до виконання і захисту кваліфікаційної роботи бакалавра спеціальностей 122 Комп'ютерні науки для здобувачів вищої освіти денної форми навчання, Київський столичний університет імені Бориса Грінченка. Київ, 2024.- 43 с.
11. ОВ Бушма, АВ Турукало Evaluation of parameters in software implementation bar graph display devices - Cybersecurity: Education, Science, Technique, 2022, Vol 4 (16), p. 142-158

12. Теоретичні та практичні аспекти використання математичних методів та інформаційних технологій в освіті й науці: моногр. / за заг. ред. О. Литвин. — К.: Київ. ун-т ім. Б. Грінченка, 2021. — 332 с.
- 13.
14. Kurniawan A. (2021) "Arduino Nano 33 IoT Board Development." In: Beginning Arduino Nano IoT. pp. 23-78. DOI:10.1007/978-1-4842-6446-1_2
15. Monk, S. (2022). Programming Arduino: Getting Started with Sketches. 3rd ed. McGraw-Hill Education. ISBN: 978-1264676989
16. O. Svystun, I. Yurchak. (2021), "Recommendation Dialog System for Selecting the Computer Hardware Configuration." Advances in Cyber-Physical Systems, Volume 6, Number 1, pp. 70 - 76. DOI: 10.23939/acps2021.01.070.
17. Petyn V.A. Datchyky dlia Arduino y Raspberry Pi v proektakh Internet of Things. BKhV-Peterburh, 2016. 320 p.
18. Smith, P. (2020). Low-Power Design Techniques for Embedded Systems. Springer. ISBN: 978-3030385743
19. Wilson, J. (2021). Smart Home Automation with Arduino. Packt Publishing. ISBN: 978-1803236234
20. А. Г. Винничук, В. В. Гринюк "Розроблення інформаційно-вимірювальної системи Контролю мікроклімату тепличних приміщень" // Методи та прилади контролю якості № 2 (43) 2019 , с. 32-40. DOI:10.31471/1993-9981-2019-2(43)-32-40
21. А. Р. Купінський, І. Ю. Юрчак "Принципи побудови метеостанції для спостереження за мікрокліматом у приміщенні на платформі Arduino", Computer systems and networks, Vol. 3, No. 1, 2021. с.68-79. DOI:10.23939/csn2021.01/068
22. ДБН В.2.5-67:2018 "Опалення, вентиляція та кондиціонування повітря"
23. ДСТУ ISO 7243:2017 "Ергономіка. Оцінка теплового навантаження на організм людини"
24. Кузьмук Розумне керування мікрокліматом, 2022. – Доступно: <https://aw-therm.com.ua/keruvannyamikroklimatom-u-primishenni/>

25.О. Колодій Рекуператор повітря: здоровий мікроклімат та збереження тепла, 2022. URL: <https://aw-therm.com.ua/rekuperator-povitrya-zdorovij-mikroklimat-ta-zberezhenня-tepla/>

26.Петин В.І Проекти з використанням контролера Arduino. 2014 р. 39с.

Додаток 1

Лістинг програми

```
- #include <Wire.h>
- #include <LiquidCrystal_I2C.h>
- #include <DHT.h> // Підключення бібліотеки для роботи з датчиком DHT
-
- #define DHTPIN 2      // Пін для підключення датчика DHT22
- #define DHTTYPE DHT22 // Тип використовуваного датчика
-
- // Піни для керування RGB світлодіодом
- #define R_PIN 4      // Червоний канал
- #define G_PIN 5      // Зелений канал
- #define B_PIN 6      // Синій канал
-
- // Піни для газового сенсора MQ-135
- const int ANALOG_GAS_PIN = A0; // Аналоговий пін
- const int DIGITAL_GAS_PIN = 3;  // Цифровий пін
- const int MOTION_SENSOR_PIN = 7; // Пін датчика руху
-
- // Глобальні змінні
- bool gasDetected = false; // Прапорець виявлення газу
- int gasConcentration = 0;  // Рівень концентрації газу
- int motionState = 0;       // Стан датчика руху
-
- DHT dht(DHTPIN, DHTTYPE);
- LiquidCrystal_I2C lcd(0x27, 20, 4); // Ініціалізація LCD 20x4
-
- void setup() {
-   dht.begin(); // Ініціалізація датчика DHT22
-
-   // Налаштування пінів на вихід
-   pinMode(R_PIN, OUTPUT);
-   pinMode(G_PIN, OUTPUT);
-   pinMode(B_PIN, OUTPUT);
-   pinMode(DIGITAL_GAS_PIN, INPUT);
-   pinMode(MOTION_SENSOR_PIN, INPUT);
-
-   // Ініціалізація LCD
-   lcd.init();
-   lcd.backlight();
-   lcd.setCursor(0, 0);
-   lcd.print("Система контролю");
-   lcd.setCursor(0, 1);
-   lcd.print("мікроклімату");
-   delay(2000);
- }
-
- void loop() {
-   motionState = digitalRead(MOTION_SENSOR_PIN);
-
-   if (motionState == HIGH) {
-     checkEnvironment();
-   } else {
-     // Енергозберігаючий режим
-     lcd.noBacklight();
-     setLedColor(LOW, LOW, LOW);
-   }
- }
-
- void checkEnvironment() {
-   lcd.backlight();
-
-   // Перевірка якості повітря
```

```

-   checkAirQuality();
-   delay(3000);
-
-   // Перевірка вологості
-   checkHumidity();
-   delay(3000);
-
-   // Перевірка температури
-   checkTemperature();
-   delay(3000);
-
-   lcd.clear();
- }
-
- void checkAirQuality() {
-   gasDetected = digitalRead(DIGITAL_GAS_PIN);
-   gasConcentration = analogRead(ANALOG_GAS_PIN);
-
-   lcd.clear();
-   lcd.setCursor(3, 0);
-   lcd.print("CO2: ");
-   lcd.print(gasConcentration);
-   lcd.print(" ppm");
-
-   if (gasConcentration < 800) {
-     setLedColor(LOW, HIGH, LOW); // Зелений - норма
-   }
-   else if (gasConcentration >= 800 && gasConcentration < 1000) {
-     setLedColor(HIGH, HIGH, LOW); // Жовтий - попередження
-   }
-   else {
-     setLedColor(HIGH, LOW, LOW); // Червоний - небезпека
-   }
- }
-
- void checkHumidity() {
-   float humidity = dht.readHumidity();
-
-   lcd.clear();
-   if (isnan(humidity)) {
-     lcd.print("Помилка датчика!");
-     return;
-   }
-
-   lcd.setCursor(0, 0);
-   lcd.print("Вологість: ");
-   lcd.print(humidity);
-   lcd.print("%");
-
-   if (humidity >= 40 && humidity <= 60) {
-     setLedColor(LOW, HIGH, LOW); // Зелений - норма
-   }
-   else if ((humidity > 20 && humidity < 40) || (humidity > 60 && humidity <
80)) {
-     setLedColor(HIGH, HIGH, LOW); // Жовтий - попередження
-   }
-   else {
-     setLedColor(HIGH, LOW, LOW); // Червоний - небезпека
-   }
- }
-
- void checkTemperature() {
-   float temperature = dht.readTemperature();
-
-

```

```

-   lcd.clear();
-   if (isnan(temperature)) {
-       lcd.print("Помилка датчика!");
-       return;
-   }
-
-   lcd.setCursor(0, 0);
-   lcd.print("Температура: ");
-   lcd.print(temperature);
-   lcd.print("°C");
-
-   if (temperature >= 18 && temperature <= 25) {
-       setLedColor(LOW, HIGH, LOW); // Зелений - норма
-   }
-   else if ((temperature > 15 && temperature < 18) || (temperature > 25 &&
temperature < 28)) {
-       setLedColor(HIGH, HIGH, LOW); // Жовтий - попередження
-   }
-   else {
-       setLedColor(HIGH, LOW, LOW); // Червоний - небезпека
-   }
- }
-
- void setLedColor(bool red, bool green, bool blue) {
-     digitalWrite(R_PIN, red);
-     digitalWrite(G_PIN, green);
-     digitalWrite(B_PIN, blue);
- }

```