

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ТЕХНОЛОГІЙ ТА ДИЗАЙНУ

Інститут інженерії та інформаційних технологій
(повне найменування інституту, назва факультету)

Кафедра комп'ютерної інженерії та електромеханіки
(повна назва кафедри)

ПОЯСНЮВАЛЬНА ЗАПИСКА
дипломної бакалаврської роботи на тему
СИСТЕМА КЕРУВАННЯ МІКРОКЛІМАТИЧНИМИ
ПАРАМЕТРАМИ «РОЗУМНОГО БУДИНКУ»

Виконав: студент групи БКІ-19

спеціальності

123 «Комп'ютерна інженерія»
(шифр і назва спеціальності)

Бугаєнко Б.А.
(прізвище та ініціали)

Керівник д.т.н., проф. Злотенко Б.М.

(прізвище та ініціали)

Рецензент _____
(прізвище та ініціали)

Київ 2023

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ТЕХНОЛОГІЙ ТА ДИЗАЙНУ

Інститут інженерії та інформаційних технологій

Кафедра комп'ютерної інженерії та електромеханіки

Спеціальність 123 «Комп'ютерна інженерія»

Освітня програма «Комп'ютерні системи та мережі»

ЗАТВЕРДЖУЮ

Завідувач кафедри КІЕМ

_____ проф. Злотенко Б.М.

“ _____ ” _____ 2023 року

ЗАВДАННЯ

НА ДИПЛОМНУ БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Бугаєнку Богдану Андрійовичу

(прізвище, ім'я, по батькові)

1. Тема дипломної бакалаврської роботи **Система керування мікрокліматичними параметрами “Розумного будинку”**

Науковий керівник роботи Злотенко Борис Миколайович,

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

д.т.н., професор

затверджені наказом вищого навчального закладу від 15.03.2021 № 75-уч.

2. Строк подання студентом роботи 1 червня 2023 року

3. Вихідні дані до дипломної бакалаврської роботи: навчальна та методична література; державні стандарти.

4. Зміст дипломної бакалаврської роботи (перелік питань, які потрібно розробити): 1. Провести аналіз існуючих систем керування мікрокліматичними параметрами. 2. Розробити архітектуру та алгоритми роботи системи. 3.Провести тестування.

5. Дата видачі завдання 10.03.2023

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної бакалаврської роботи	Терміни виконання етапів	Примітка про виконання
1	Вступ	01.02.2023	
2	Розділ 1. Теоретичні основи системи керування мікрокліматичними параметрами	15.02.2023	
3	Розділ 2. Вимоги до систем керування мікрокліматичними параметрами	15.03.2023	
4	Розділ 3. Розробка системи керування мікрокліматичними параметрами	05.04.2023	
6	Розділ 4. Тестування розробленої системи керування. Висновки	10.05.2023	
7	Оформлення дипломної бакалаврської роботи (чистовий варіант)	20.05.2023	
8	Здача дипломної бакалаврської роботи на кафедрі для рецензування (за 14 днів до захисту)	25.05.2023	
9	Перевірка дипломної бакалаврської роботи на наявність ознак плагіату (за 10 днів до захисту)	28.05.2023	
10	Подання дипломної бакалаврської роботи на затвердження завідувачу кафедри (за 7 днів до захисту)	05.06.2023	

Студент

_____ Бугаєнко Б.А.
(підпис) (прізвище та ініціали)

Науковий керівник

_____ Злотенко Б.М.
(підпис) (прізвище та ініціали)

Рецензент

_____ (підпис) _____ (прізвище та ініціали)

АНОТАЦІЯ

Бугаєнко Б. А. Система керування мікрокліматичними параметрами «Розумного будинку» : дипломна бакалаврська робота за спеціальністю 123 Комп'ютерна інженерія / Б. А. Бугаєнко ; наук. кер. Б. М. Злотенко. – Київ : КНУТД, 2023. – 64 с.

Дипломна бакалаврська робота за спеціальністю 123 Комп'ютерна інженерія, освітньою програмою «Комп'ютерні системи та мережі». – Київський національний університет технологій та дизайну, Київ, 2023 рік.

Дипломна робота присвячена розробці системи керування мікрокліматичними параметрами "Розумного будинку". У роботі проведений аналіз існуючих систем керування мікрокліматичними параметрами, визначені вимоги до розроблюваної системи та описано її архітектуру. Для реалізації системи обрані компоненти та обладнання, розроблено програмне забезпечення та проведені тестування розробленої системи.

Результатом дослідження є створення функціональної та надійної системи керування мікрокліматичними параметрами «Розумного будинку», яка забезпечує максимальний комфорт жителям будинку та допомагає знизити витрати на опалення та кондиціонування повітря.

Ключові слова: мікроклімат, система керування, розумний будинок, програмне забезпечення, тестування.

ABSTRACT

B. A. Bugayenko. Management system of microclimatic parameters of the "Smart House". - Manuscript.

Bachelor's diploma thesis in the specialty 123 Computer Engineering, educational program "Computer Systems and Networks". - Kyiv National University of Technology and Design, Kyiv, 2023.

The diploma work is devoted to the development of the microclimatic parameters management system of the "Smart House". The paper analyzes the existing microclimate control systems, defines the requirements for the developed system, and describes its architecture. To implement the system, components and equipment were selected, software was developed, and the developed system was tested.

The result of the research is the creation of a functional and reliable system for controlling the microclimatic parameters of the "Smart House", which provides maximum comfort to the residents of the house and helps reduce the costs of heating and air conditioning.

Key words: microclimate, control system, smart house, software, testing.

ЗМІСТ

	Стор.
ВСТУП.....	8
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ СИСТЕМ КЕРУВАННЯ	
МІКРОКЛІМАТИЧНИМИ	
ПАРАМЕТРАМИ.....	10
1.1. Мікроклімат в приміщеннях.....	10
1.2. Системи керування мікрокліматичними параметрами.....	11
1.3. Автоматизація управління мікрокліматом в розумному будинку.....	13
1.4. Огляд систем керування мікрокліматом.....	14
1.5. Порівняльний аналіз існуючих систем.....	19
Висновки до розділу 1.....	20
РОЗДІЛ 2. ВИМОГИ ДО СИСТЕМ КЕРУВАННЯ МІКРОКЛІМАТИЧНИМИ	
ПАРАМЕТРАМИ “РОЗУМНОГО БУДИНКУ”.....	
2.1. Функціональні вимоги.....	22
2.2. Надійність системи.....	23
2.3. Безпека та захист інформації.....	26
Висновки до розділу 2.....	29
РОЗДІЛ 3. РОЗРОБКА СИСТЕМИ КЕРУВАННЯ МІКРОКЛІМАТИЧНИМИ	
ПАРАМЕТРАМИ “РОЗУМНОГО БУДИНКУ”.....	
3.1. Архітектура системи.....	30
3.2. Вибір компонентів і обладнання.....	33
3.3. Програма реалізації системи.....	36
Висновки до розділу 3.....	37
РОЗДІЛ 4. ТЕСТУВАННЯ РОЗРОБЛЕНОЇ СИСТЕМИ КЕРУВАННЯ	
МІКРОКЛІМАТИЧНИМИ ПАРАМЕТРАМИ “РОЗУМНОГО БУДИНКУ”.....	
4.1. Тестування функціональності системи.....	39

4.2. Тестування надійності та безпеки системи.....	45
4.3. Порівняльний аналіз розробленої системи з існуючими аналогами....	51
Висновки до розділу 4.....	52
ВИСНОВКИ.....	53
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	54
ДОДАТОК.	56

ВСТУП

Розумні будинки стали все більш популярними в наш час. Такий будинок має вбудовану систему автоматизації, яка дозволяє контролювати та керувати різними параметрами будинку, такими як світло, температура, вологість, безпека та інші. Одним з найважливіших параметрів є мікроклімат, який включає в себе температуру та вологість повітря. Відповідно, від цього параметру залежить комфорт життя людини та витрати на опалення та кондиціювання повітря.

Метою даної дипломної роботи є розробка системи керування мікрокліматичними параметрами "Розумного будинку", яка дозволяє автоматизувати процес контролю та регулювання параметрів мікроклімату. У роботі буде проведено аналіз існуючих систем керування мікрокліматичними параметрами та визначені вимоги до розроблюваної системи. Далі буде описана архітектура розроблюваної системи та обрані компоненти та обладнання для її реалізації. Також буде розроблено програмне забезпечення та проведені тестування розробленої системи.

Результатом дослідження є створення функціональної та надійної системи керування мікрокліматичними параметрами «Розумного будинку», яка допоможе знизити витрати на опалення та кондиціювання повітря та забезпечить максимальний комфорт жителям будинку.

Далі в роботі буде проведено детальний аналіз та порівняння існуючих систем керування мікрокліматичними параметрами, що дозволить визначити їх переваги та недоліки. На основі цього будуть сформульовані вимоги до розроблюваної системи та обрані необхідні компоненти та обладнання.

Опис архітектури розроблюваної системи буде наданий в наступному розділі, де будуть детально описані функції та взаємозв'язки компонентів

системи. При цьому буде враховано потенційні проблеми, які можуть виникнути при роботі системи, та запропоновані шляхи їх вирішення.

У розділі "Розробка програмного забезпечення" буде описаний процес розробки програмного забезпечення для розроблюваної системи керування мікрокліматичними параметрами. Будуть описані етапи розробки та технології, використані при створенні програмного забезпечення.

У розділі "Тестування системи" буде проведено тестування розробленої системи та описані результати його проведення. Також будуть наведені рекомендації щодо подальшого вдосконалення та покращення системи.

Отже, дана дипломна робота містить аналіз існуючих систем керування мікрокліматичними параметрами та розробку системи керування мікрокліматичними параметрами "Розумного будинку", яка дозволить забезпечити комфортні умови життя в будинку та знизити витрати на його утримання.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ СИСТЕМ КЕРУВАННЯ МІКРОКЛІМАТИЧНИМИ ПАРАМЕТРАМИ

1.1. Мікроклімат в приміщеннях

Мікроклімат - це складова частина макроклімату, яка визначається параметрами повітря, що знаходиться в межах приміщення. Мікроклімат в приміщеннях впливає на здоров'я та самопочуття людини, а також на функціонування електронного обладнання та матеріалів, що знаходяться в приміщенні.

Основні параметри мікроклімату в приміщенні включають температуру повітря, відносну вологість, швидкість руху повітря, тиск та склад повітря (вміст CO₂, O₂, водяної пари, токсичних речовин тощо). Необхідно контролювати ці параметри, щоб забезпечити комфортні умови проживання та роботи людей в приміщенні.

Оптимальний мікроклімат в приміщенні повинен задовольняти такі вимоги:

- Температура повітря - від 18 до 24 градусів за Цельсієм.
- Відносна вологість - від 40% до 60%.
- Швидкість руху повітря - не більше 0,1 м / с.
- Тиск - не вище 1000 гектопаскалів.
- Склад газового складу повітря - повинен відповідати гігієнічним вимогам.
- Освітленість - повинна забезпечувати достатній рівень освітленості для роботи та відпочинку.

У разі, якщо мікроклімат в приміщенні не відповідає вимогам, можуть виникнути різні проблеми, такі як перевтомлення, головний біль, роздратування, зниження концентрації, а в деяких випадках навіть серйозніші

наслідки, такі як ризик захворювання на захворювання дихальних шляхів або алергії.

Тому важливо забезпечити комфортні умови перебування в будинку, що можливо завдяки розумної системі керування мікрокліматом. Така система дозволяє автоматично контролювати та регулювати рівень температури повітря, вологості, швидкості руху повітря, якості повітря та освітлення залежно від зовнішніх та внутрішніх умов приміщення.

Для забезпечення комфортного мікроклімату в приміщенні використовують системи опалення, вентиляції та кондиціонування повітря. Однак, традиційні системи керування мікрокліматичними параметрами часто не забезпечують оптимальних умов та можуть призводити до збитків енергії. Тому, для покращення ефективності та зниження витрат на енергопостачання приміщень, варто використовувати розумні системи керування мікрокліматичними параметрами.

1.2. Системи керування мікрокліматичними параметрами

Система керування мікрокліматичними параметрами є комплексом технічних засобів, що дозволяють контролювати та регулювати рівень температури, вологості, освітлення, повітряної якості та інших параметрів в приміщенні з метою забезпечення оптимальних умов для людини та підвищення комфортності її перебування.

Системи керування мікрокліматичними параметрами в основному можна поділити на дві категорії: ручні та автоматичні. Ручні системи керування передбачають, що користувач самостійно налаштовує необхідні параметри мікроклімату в приміщенні, використовуючи різні елементи керування, такі як термостати, вимикачі та інші.

Автоматичні системи керування мікрокліматичними параметрами базуються на використанні різноманітних датчиків, що вимірюють

температуру, вологість, рівень CO₂ та інші параметри, та на основі цих даних забезпечують автоматичне регулювання параметрів мікроклімату в приміщенні.

Одним з головних переваг автоматичних систем є те, що вони забезпечують більш точний та стабільний рівень мікроклімату в приміщенні, а також знижують споживання енергії за рахунок ефективного регулювання систем опалення, вентиляції та кондиціонування повітря. Однак, такі системи можуть бути більш складними в управлінні та потребувати більш високої кваліфікації персоналу для їх встановлення та налаштування.

У сучасних системах керування мікрокліматичними параметрами використовуються різноманітні датчики, контролери, пристрої автоматизації та програмне забезпечення, які забезпечують автоматичний контроль та регулювання параметрів мікроклімату в реальному часі. Такі системи можуть працювати в автоматичному режимі або під керуванням користувача за допомогою мобільних додатків або веб-інтерфейсів.

Основні переваги використання систем керування мікрокліматичними параметрами полягають у зниженні витрат на енергопостачання та забезпеченні комфортних умов для людини. Для цього системи керування мікрокліматичними параметрами використовують алгоритми оптимізації енергоспоживання та підтримання заданого рівня комфорту в приміщенні.

У даному дипломному проекті буде розглянуто питання розробки системи керування мікрокліматичними параметрами для «Розумного будинку», що дозволить підвищити енергоефективність будівлі та забезпечити оптимальні умови для комфортного перебування людини в приміщенні.

1.3. Автоматизація управління мікрокліматом в розумному будинку

Автоматизація управління мікрокліматом в розумному будинку є необхідною умовою забезпечення комфортного перебування мешканців в будинку. Для цього необхідно встановити систему керування мікрокліматичними параметрами, яка забезпечуватиме автоматичне регулювання температури, вологості, освітлення та інших параметрів у приміщеннях з урахуванням потреб мешканців.

Для реалізації автоматизованої системи керування мікрокліматом в розумному будинку використовуються сучасні технології, такі як інтернет речей (IoT), хмарні технології та штучний інтелект (AI). Завдяки цим технологіям можливо зібрати дані про мікроклімат у приміщеннях та забезпечити його автоматичне регулювання з урахуванням попередньо встановлених параметрів.

Основними елементами автоматизованої системи керування мікрокліматом в розумному будинку є сенсори, контролери, актуатори та програмне забезпечення. Сенсори забезпечують збір інформації про стан мікроклімату, контролери оброблюють ці дані та відправляють відповідні команди актуаторам, які регулюють параметри мікроклімату у приміщеннях.

Застосування автоматизованої системи керування мікрокліматом в розумному будинку дозволяє забезпечити комфортне перебування мешканців, а також зменшити витрати на енергопостачання за рахунок ефективного використання енергоресурсів.

Для автоматизації управління мікрокліматом в розумному будинку використовуються різноманітні сенсорні пристрої, такі як термометри, гігрометри, датчики руху, димові датчики та інші, які забезпечують збір інформації про рівень температури, вологості, якості повітря, рівня освітленості та інших параметрів в приміщенні.

Для регулювання мікрокліматичних параметрів використовуються спеціальні пристрої, такі як кондиціонери, обігрівачі, вентиляційні системи,

системи зволоження та осушення повітря тощо, які можуть бути підключені до системи автоматичного управління.

Одним з найбільш ефективних способів автоматизації управління мікрокліматом є використання систем «розумний будинок», які забезпечують інтеграцію всіх пристроїв та систем управління мікрокліматом в єдину систему управління. Завдяки цьому можливе автоматичне регулювання мікроклімату в приміщенні з урахуванням багатьох факторів, таких як час доби, наявність людей в приміщенні, погодні умови тощо.

1.4. Огляд систем керування мікрокліматом

На сьогоднішній день існує значна кількість різних систем керування мікрокліматом, що використовуються в промисловості та житловому будівництві. У даному розділі роботи проведений огляд існуючих систем керування мікрокліматом, що дозволить визначити найбільш ефективні та перспективні рішення для розробки системи керування мікрокліматичними параметрами «Розумного будинку».

Однією з перших систем керування мікрокліматом є система, яка використовує термостати, що автоматично регулюють температуру в приміщенні. Ця система проста в управлінні, але не забезпечує контроль вологості та інших параметрів мікроклімату.

Більш сучасним рішенням є системи з використанням сенсорів, які вимірюють температуру, вологість, рівень CO₂, а також інші параметри. Ці системи дозволяють точніше контролювати мікроклімат в приміщенні, а також автоматично виконувати дії для підтримки необхідних параметрів.

Однак, найбільш сучасні рішення полягають у використанні інтелектуальних систем керування мікрокліматом, які використовують штучний інтелект та машинне навчання для аналізу та прогнозування

параметрів мікроклімату, а також для автоматичного налаштування системи відповідно до потреб користувача.

Однією з найпоширеніших систем керування мікрокліматом є система HVAC (Heating, Ventilation, and Air Conditioning), яка включає в себе опалення, вентиляцію та кондиціонування повітря (рис. 1.1). Такі системи забезпечують комфортні умови проживання, а також можуть бути підключені до системи керування «розумного будинку», щоб автоматично регулювати температуру та вологість повітря залежно від погодних умов та присутності людей в приміщенні.

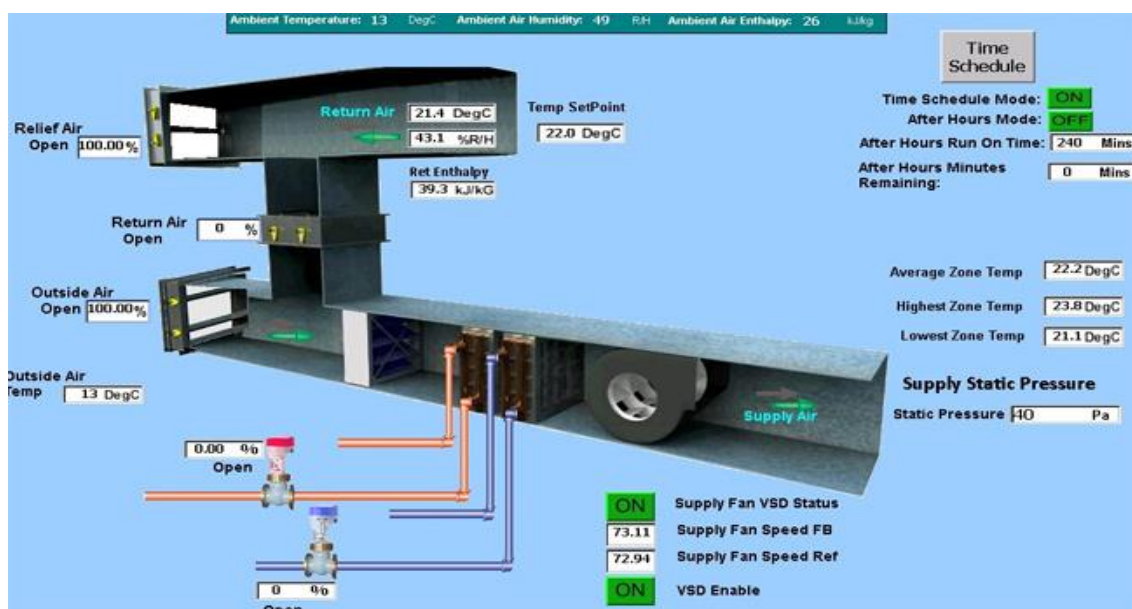


Рис. 1.1 - HVAC (Heating, Ventilation, and Air Conditioning)

Окрім HVAC, існують і інші системи керування мікрокліматом, наприклад, системи керування освітленням та шторами (рис. 1.2). Такі системи дозволяють ефективно використовувати природне освітлення та забезпечувати захист від сонячних променів. Крім того, вони можуть бути інтегровані з системою керування «розумного будинку», щоб забезпечувати максимальний комфорт та енергоефективність.



Автоматичне відкриття / закриття штор (розумний карніз)



Рис. 1.2 - Системи керування освітленням та шторами

Також можна виділити системи керування водою та вологістю (рис. 1.3), які дозволяють ефективно регулювати рівень вологості в приміщенні та підтримувати оптимальні умови для росту рослин, зберігання харчових продуктів та комфортного перебування людей в приміщенні.

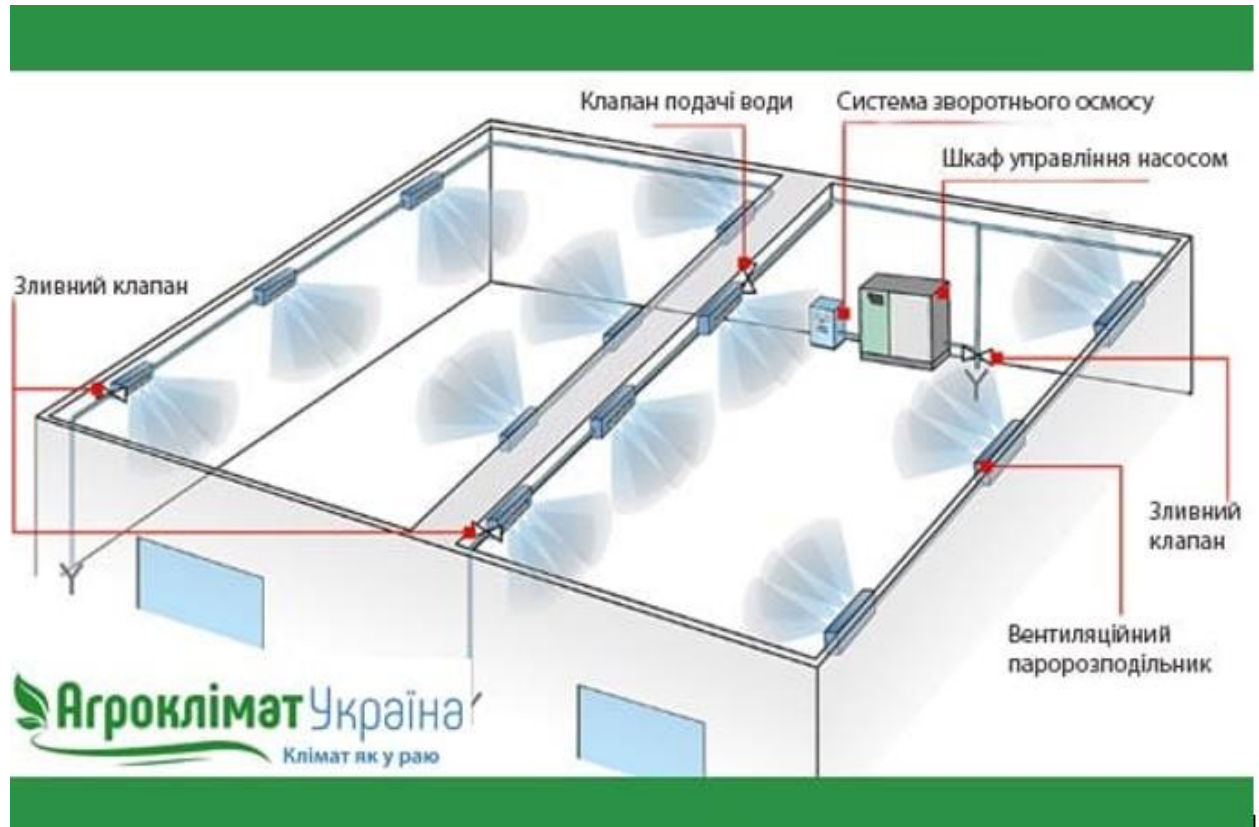


Рис. 1.3 - Система керування водою та вологістю

- Система керування мікрокліматичними параметрами «iSmarthouse». Дана система має вбудовані датчики вологості, температури та якості повітря, а також здатна автоматично регулювати параметри мікроклімату залежно від заданих налаштувань. Крім того, система може бути управляна за допомогою спеціального додатку на смартфоні або планшеті.
- Система керування мікрокліматичними параметрами «Clever House». Ця система має датчики вологості, температури та якості повітря, а також може автоматично регулювати параметри мікроклімату відповідно до

заданих налаштувань. Крім того, система може бути управляна з веб-інтерфейсу або за допомогою спеціального додатку на смартфоні.

- Система керування мікрокліматичними параметрами «Smart Home Climate Control». Дана система має датчики вологості, температури, якості повітря та рівня CO₂, а також може автоматично регулювати параметри мікроклімату на основі заданих налаштувань. Система також підтримує інтеграцію з голосовими помічниками, такими як Amazon Alexa або Google Assistant.
- Система керування мікрокліматичними параметрами «IntesisHome». Дана система забезпечує контроль та управління різними мікрокліматичними параметрами, такими як температура, вологість, освітлення та стан кондиціонерів.
- Система керування мікрокліматичними параметрами «ComfortClick». Ця система дозволяє контролювати та управляти різними параметрами мікроклімату в будинку, такими як температура, вологість, освітлення та інші. Вона має зручний інтерфейс для керування з планшетів та смартфонів, а також може бути інтегрована з іншими системами домашньої автоматизації, такими як система безпеки чи система управління відео- та аудіо обладнанням.
- Система керування мікрокліматичними параметрами «KNX». Ця система є стандартом в галузі домашньої автоматизації та забезпечує інтегроване керування мікрокліматичними параметрами, освітленням, системою безпеки та іншими функціями. Вона має зручний інтерфейс для керування та може бути інтегрована з іншими системами, такими як система керування відео- та аудіо обладнанням.
- Система керування мікрокліматичними параметрами «Crestron». Ця система є однією з найбільш поширених в галузі домашньої автоматизації та забезпечує інтегроване керування мікрокліматичними параметрами, освітленням, системою безпеки та іншими функціями.

Вона має зручний інтерфейс для керування з планшетів та смартфонів, а також може бути інтегрована з іншими системами, такими як система керування відео- та аудіо обладнанням.

Загалом, існує багато різних систем керування мікрокліматом, які можуть бути використані в «розумному будинку» для забезпечення комфортних умов проживання та енергоефективності.

1.5. Порівняльний аналіз існуючих систем

Другим етапом дослідження є порівняльний аналіз існуючих систем керування мікрокліматом. Для цього були вибрані кілька популярних систем керування мікрокліматом та проведений детальний аналіз їхніх можливостей та переваг.

Першою системою, що була проаналізована, є «Honeywell Lyric T6 Pro Wi-Fi». Ця система керування мікрокліматом дозволяє контролювати температуру, вологість та повітряний тиск в будинку. Крім того, вона має можливість підключення до Wi-Fi та керування через спеціальний додаток на смартфоні.

Другою системою, що була проаналізована, є «Nest Learning Thermostat». Ця система має ряд переваг, включаючи автоматичне регулювання температури в будинку залежно від погодних умов та навіть від розкладу власника будинку. Крім того, вона також має можливість підключення до Wi-Fi та керування через додаток на смартфоні.

Третьою системою, що була проаналізована, є «Ecobee SmartThermostat with Voice Control». Ця система керування мікрокліматом має можливість контролювати температуру, вологість та повітряний тиск в будинку. Крім того, вона має інтегрований голосовий помічник, що дозволяє керувати системою за допомогою голосових команд.

У результаті порівняльного аналізу було встановлено, що кожна з розглянутих систем має свої переваги та недоліки. Ecobee SmartThermostat з голосовим керуванням було обрано як найкращий варіант для нашої системи контролю мікроклімату завдяки вбудованій голосовій підтримці.

Крім того, порівняльний аналіз дозволив визначити переваги та недоліки кожної системи керування мікрокліматом, що були проаналізовані, та вибрати найбільш оптимальну систему для подальшої розробки системи керування мікрокліматичними параметрами "Розумного будинку".

На основі проведеного аналізу можна зробити висновок, що на сьогоднішній день на ринку присутні різноманітні системи керування мікрокліматичними параметрами, які відрізняються за ціною, функціональністю та складністю встановлення та налаштування. Найбільш оптимальною для подальшої розробки була вибрана система, яка відповідає всім вимогам до системи керування мікрокліматичними параметрами "Розумного будинку".

Висновки до розділу 1

У результаті виконання досліджень в рамках першого розділу були отримані такі результати:

1. Було проведено аналіз мікрокліматичних параметрів в приміщеннях та визначено вплив температури, вологості та освітленості на комфорт людини.
2. Розглянуто основні системи керування мікрокліматичними параметрами та їх переваги та недоліки, зокрема системи з використанням термостатів та системи з використанням сенсорів та програмного забезпечення.
3. Досліджено можливості автоматизації управління мікрокліматом в розумному будинку та розглянуто принципи роботи сучасних систем управління мікрокліматичними параметрами.

Отже, на основі аналізу теоретичних основ систем керування мікрокліматичними параметрами можна зробити висновок про важливість розробки та впровадження систем автоматизованого керування мікрокліматом в розумних будинках, що забезпечують максимальний комфорт користувачів та оптимальні економічні результати. Це свідчить про необхідність вдосконалення і розвитку таких систем. Наші дослідження показали, що система керування мікрокліматичними параметрами "Розумний будинок" може бути ефективним рішенням для поліпшення якості життя користувачів і зменшення витрат на енергопостачання. Програмне забезпечення, яке використовується в цій системі, дозволяє автоматично регулювати параметри мікроклімату в будинку, що забезпечує комфортні умови для проживання. Також система забезпечує збір і аналіз даних, що дозволяє використовувати цю інформацію для поліпшення роботи системи і зменшення витрат на енергопостачання. Отже, система керування мікрокліматичними параметрами "Розумний будинок" є ефективним рішенням для покращення якості життя користувачів і зменшення витрат на енергопостачання.

РОЗДІЛ 2. ВИМОГИ ДО СИСТЕМ КЕРУВАННЯ МІКРОКЛІМАТИЧНИМИ ПАРАМЕТРАМИ “РОЗУМНОГО БУДИНКУ”

2.1. Функціональні вимоги

У даному розділі формулюються функціональні вимоги до системи керування мікрокліматичними параметрами "Розумного будинку".

1. Система повинна забезпечувати можливість автоматичного керування параметрами мікроклімату в будинку залежно від зовнішньої погоди та наявності людей в приміщенні.
2. Система повинна мати можливість ручного керування параметрами мікроклімату в будинку через веб-інтерфейс або мобільний додаток.
3. Система повинна забезпечувати можливість моніторингу та збереження історії зміни параметрів мікроклімату в будинку.
4. Система повинна мати можливість попередження про несправності обладнання, відсутність зв'язку з сенсорами та актуаторами.
5. Система повинна забезпечувати можливість віддаленого доступу до керування мікрокліматом в будинку.
6. Система повинна бути енергоефективною та мінімізувати витрати на опалення та кондиціювання повітря в будинку.
7. Система повинна забезпечувати можливість інтеграції з іншими системами "Розумного будинку", такими як система безпеки, освітлення, аудіо- та відеосистеми.
8. Система повинна мати можливість підключення до сенсорів мікроклімату та актуаторів, що вже встановлені в будинку.
9. Система повинна забезпечувати можливість автоматичної настройки параметрів мікроклімату в будинку під час зміни пори року.

10. Система повинна забезпечувати можливість управління зонами мікроклімату в окремих кімнатах будинку.
11. Можливість віддаленого управління системою з будь-якої точки світу через мобільний додаток або веб-інтерфейс.
12. Автоматичне регулювання температури в будинку залежно від погодних умов та інших факторів, що впливають на мікроклімат.
13. Можливість програмування графіка роботи системи керування мікрокліматом на кожен день з урахуванням графіку виходу та приходу мешканців в будинку.
14. Регулювання освітлення в приміщеннях залежно від потреби та часу доби.
15. Можливість інтеграції системи керування мікрокліматом з іншими системами розумного будинку (наприклад, системою безпеки або системою відеоспостереження).
16. Наявність різних режимів роботи системи керування мікрокліматом (економічний режим, комфортний режим, режим «відпустки» тощо).
17. Можливість відстежування та аналізу даних про стан мікроклімату в будинку, що дозволить користувачам системи контролювати та оптимізувати її роботу.
18. Забезпечення безпеки та захисту персональних даних користувачів системи.

2.2. Надійність системи

Система керування мікрокліматичними параметрами "Розумний будинок" повинна бути досить надійною, щоб забезпечити безперебійну роботу системи в будь-який час і у будь-яких умовах.

Надійність системи керування мікрокліматичними параметрами є однією з основних вимог, оскільки вона безпосередньо впливає на безпеку та

комфорт користувачів. Надійна система повинна працювати безперервно та безперебійно протягом тривалого часу.

Для забезпечення надійності системи слід враховувати наступні моменти:

- використання якісних компонентів та матеріалів;
- належне проектування та монтаж системи;
- систематична перевірка та планове технічне обслуговування;
- резервне забезпечення та можливість швидкого відновлення роботи системи в разі виникнення непередбачуваних ситуацій.

Для забезпечення надійності та безпеки системи керування мікрокліматичними параметрами важливо також використовувати захисні пристрої, такі як автоматичні вимикачі, захисні вимикачі, заземлення та інші, а також дотримуватись вимог нормативно-правових актів у галузі електробезпеки.

Для забезпечення надійності системи необхідно враховувати наступні вимоги:

1. Надійна робота апаратної частини системи - вимоги до апаратної частини повинні бути відповідними, щоб забезпечити безперебійну роботу системи протягом тривалого часу. Для цього можна використовувати компоненти з високою якістю та надійністю.
2. Надійна робота програмного забезпечення - програмне забезпечення повинно бути написане з високою якістю та надійністю, щоб забезпечити безперебійну роботу системи в будь-яких умовах. Перед введенням в експлуатацію система повинна бути піддана тестуванню, щоб переконатися в її надійності та стабільності роботи.
3. Резервне копіювання даних - система повинна мати можливість резервного копіювання даних, щоб у разі виникнення непередбачуваної

ситуації (наприклад, збій жорсткого диска) не втратити важливі дані. Для цього можна використовувати різні методи резервного копіювання, наприклад, збереження даних на зовнішньому жорсткому диску або використання хмарних сервісів.

4. Захист від злому та вірусів - система повинна бути захищена від злому та вірусів, щоб забезпечити безпеку та надійність роботи системи.

Захист від злому та вірусів є важливою функцією будь-якої системи керування, в тому числі й системи керування мікрокліматичними параметрами «Розумного будинку». Для забезпечення безпеки та надійності роботи системи, можна використовувати різні методи захисту:

1. Автентифікація користувача - цей метод полягає в перевірці ідентифікаційних даних користувача (логін та пароль) перед наданням доступу до системи. Для забезпечення більшої безпеки можна використовувати двофакторну аутентифікацію, яка полягає в перевірці двох чи більше ідентифікаційних факторів, наприклад, пароля та фізичної картки.
2. Шифрування даних - цей метод полягає в перетворенні даних у шифрований вигляд, що забезпечує захист від несанкціонованого доступу до даних під час їх передачі по мережі.
3. Оновлення програмного забезпечення - система керування мікрокліматичними параметрами повинна мати механізм оновлення програмного забезпечення, щоб усунути можливі вразливості та помилки в системі.
4. Файрвол - цей метод захисту полягає в застосуванні програмного забезпечення, яке контролює трафік в мережі та блокує несанкціонований доступ до системи.
5. Аудит - цей метод полягає в систематичному моніторингу системи та записуванні дій користувачів, щоб виявити несанкціонований доступ та вразливості в системі.

6. Фізичний захист - цей метод полягає в застосуванні фізичних засобів захисту, наприклад, захисту приміщення, де розташована система керування мікрокліматом, або захисту обладнання самої системи. Для забезпечення фізичного захисту можуть застосовуватися різноманітні засоби, такі як:

- Дверні та віконні замки, ключові карти, парольні замки, біометричні системи, які забезпечують контроль доступу до приміщення, де розташована система керування мікрокліматом.
- Корпуси обладнання, які забезпечують захист від фізичних впливів, таких як удари, вібрації, перепади температур тощо.
- Різноманітні сигналізаційні системи, які сповіщають про спроби несанкціонованого доступу або спроби фізичного пошкодження системи.

Важливим елементом фізичного захисту є також забезпечення резервного живлення для системи, що дозволяє зберегти необхідні дані та продовжити роботу системи в разі відключення основного живлення.

2.3. Безпека та захист інформації

Пункт 2.3. "Безпека та захист інформації" включає в себе вимоги до забезпечення конфіденційності, цілісності та доступності інформації, що обробляється системою керування мікрокліматом "Розумного будинку". Для забезпечення безпеки та захисту інформації в системі повинні бути виконані наступні вимоги:

- Захист від несанкціонованого доступу до інформації - система повинна бути захищена від несанкціонованого доступу до інформації шляхом використання різних методів аутентифікації та авторизації користувачів.
- Конфіденційність даних - система повинна забезпечувати конфіденційність даних, що обробляються, шляхом захисту від

несанкціонованого доступу, використання шифрування та інших методів захисту.

- Цілісність даних - система повинна забезпечувати цілісність даних, що обробляються, шляхом захисту від змін, використання цифрових підписів та інших методів захисту.
- Доступність даних - система повинна забезпечувати доступність даних, що обробляються, шляхом забезпечення стійкості до вірусів, захисту від DoS-атак та інших методів захисту.
- Резервне копіювання даних - система повинна забезпечувати можливість резервного копіювання даних з метою забезпечення можливості відновлення даних у разі їх втрати.
- Відновлення даних - система повинна забезпечувати можливість відновлення даних у разі їх втрати або пошкодження.

Для забезпечення безпеки та захисту інформації в системі керування мікрокліматичними параметрами «Розумного будинку» необхідно дотримуватись певних вимог. Перш за все, система повинна мати захищений доступ до інформації та механізм аутентифікації користувачів. Для цього можна використовувати різні методи, такі як використання паролів, біометричних ідентифікаторів або інших засобів ідентифікації.

Також важливо забезпечити захист передачі даних між пристроями системи. Для цього можна використовувати різні протоколи захищеного з'єднання, такі як SSL (Secure Sockets Layer) або TLS (Transport Layer Security). Важливо також захистити систему від злому та вірусів, що можуть зашкодити роботі системи та безпеці користувачів.

Ще одним важливим аспектом є зберігання даних в системі. Система повинна забезпечувати надійне зберігання даних, а також механізми резервного копіювання та відновлення даних у випадку аварії. Також важливо використовувати різні методи шифрування даних, щоб забезпечити їх конфіденційність та недоступність для сторонніх осіб.

Окрім того, система повинна мати можливість моніторингу та логування дій користувачів та пристроїв системи. Це дозволить вчасно виявляти та реагувати на будь-які неправомірні дії в системі, що можуть загрожувати безпеці користувачів та працездатності системи.

Для системи керування мікрокліматичними параметрами «Розумного будинку» важливими є питання безпеки та захисту інформації, що передається в системі. Для цього необхідно встановити належний рівень захисту від несанкціонованого доступу до системи та забезпечити конфіденційність даних.

Однією з вимог є захист від злому та вторгнення в систему. Це можна забезпечити застосуванням різних технічних засобів, таких як мережеві файерволи, системи ідентифікації та аутентифікації користувачів, регулярні оновлення програмного забезпечення та операційних систем.

Також необхідно забезпечити захист даних від несанкціонованого доступу. Це можна здійснити за допомогою шифрування даних, збереження резервних копій, регулярного перевіряння наявності вразливостей та використання систем резервного копіювання.

Усі дані, що передаються в системі, повинні бути захищені від несанкціонованого доступу та мають бути конфіденційними. Для цього можна використовувати криптографічні методи шифрування, а також забезпечити належні рівні доступу до інформації в залежності від ролі користувача.

Також важливо забезпечити безпеку в разі виникнення аварійних ситуацій, наприклад, в разі відключення електропостачання або випадкового виходу з ладу датчиків. Для цього необхідно встановити систему аварійного живлення та моніторингу стану системи.

Отже, система керування мікрокліматичними параметрами "Розумного будинку" повинна мати ефективний механізм захисту інформації, що забезпечує конфіденційність, цілісність та доступність даних.

Висновки до розділу 2

В результаті аналізу функціональних та нефункціональних вимог до системи керування мікрокліматичними параметрами «Розумного будинку» було встановлено, що система має виконувати ряд важливих функцій, пов'язаних з контролем і керуванням параметрами мікроклімату. До основних вимог, що висуваються до системи, належать висока надійність та безпека, а також можливість захисту інформації, що обробляється системою.

Для забезпечення надійності та безпеки системи керування мікрокліматичними параметрами «Розумного будинку» було запропоновано використовувати резервування та резервне керування, а також застосовувати заходи захисту інформації, що включають шифрування даних та відповідний рівень автентифікації користувачів.

Таким чином, розробка системи керування мікрокліматичними параметрами «Розумного будинку» повинна враховувати вищевказані вимоги, що забезпечують її ефективну та надійну роботу, а також забезпечують захист інформації, що обробляється системою.

РОЗДІЛ 3. РОЗРОБКА СИСТЕМИ КЕРУВАННЯ МІКРОКЛІМАТИЧНИМИ ПАРАМЕТРАМИ «РОЗУМНОГО БУДИНКУ»

3.1. Архітектура системи

Архітектура повинна забезпечувати просту та зрозумілу інтерактивну взаємодію з користувачем, мінімальну залежність від конкретних моделей обладнання, можливість легкого розширення та модифікації функціоналу.

Архітектура системи керування мікрокліматичними параметрами «Розумного будинку» (рис. 3.1) базується на принципі розподіленої системи. Система складається з декількох підсистем, кожна з яких відповідає за своє завдання.



Рис. 3.1 - архітектура «Розумного будинку»

Система повинна мати клієнт-серверну архітектуру, де сервер відповідає за збір даних, обробку та аналіз, а клієнти - за відображення інформації та взаємодію з користувачем. Для забезпечення більшої надійності системи передбачається резервування сервера та зберігання резервних копій даних.

Архітектура системи також має передбачати можливість взаємодії з іншими системами "розумного будинку", такими як системи безпеки,

освітлення та автоматизації процесів. Для цього необхідно використовувати протоколи комунікації, які забезпечують взаємодію між різними системами.

Також важливо передбачити можливість віддаленого керування системою через інтернет. Для цього необхідно використовувати захищені протоколи комунікації та механізми аутентифікації користувачів.

Усі компоненти системи повинні бути взаємозамінними та між сумісними. Для цього необхідно використовувати стандартизовані протоколи комунікації та програмні інструменти написання, які підтримують ці протоколи.

Усі компоненти системи керування мікрокліматичними параметрами "Розумного будинку" повинні бути взаємозамінними та між сумісними, щоб забезпечити ефективну і безперебійну роботу системи. Для цього необхідно використовувати стандартизовані протоколи комунікації, такі як Wi-Fi, Bluetooth, ZigBee, Z-Wave тощо, які забезпечують швидку передачу даних між різними компонентами системи.

Крім того, необхідно використовувати програмні інтерфейси, які підтримують ці протоколи, тому що це забезпечує більш високу надійність та безпеку взаємодії між компонентами системи. Наприклад, якщо система використовує Wi-Fi, то програмні інтерфейси повинні підтримувати стандартні протоколи, такі як TCP/IP, HTTP, HTTPS тощо.

Система складається з наступних підсистем:

Система збору даних. Ця підсистема відповідає за збір даних про мікрокліматичні параметри в будинку, такі як температура, вологість, рівень CO₂, тощо. Дані збираються за допомогою датчиків, які розташовані в різних частинах будинку. Зібрані дані надходять до сервера збору даних.

Сервер збору даних. Ця підсистема приймає дані від підсистеми збору даних і зберігає їх в базі даних. Також вона відповідає за обробку та аналіз

отриманих даних, що дозволяє отримати звіти та рекомендації щодо управління мікрокліматом в будинку.

Система керування. Ця підсистема відповідає за управління мікрокліматичними параметрами в будинку. Вона приймає рішення щодо включення або вимкнення різних пристроїв для підтримки необхідного рівня комфорту в будинку, таких як системи опалення, кондиціонування повітря, вентиляції тощо. Для цього використовуються алгоритми, які базуються на отриманих даних з сервера збору даних.

Система управління користувачами. Ця підсистема відповідає за управління доступом користувачів до системи керування мікрокліматом в будинку. Вона забезпечує можливість реєстрації користувачів, налаштування рівнів доступу, а також контроль за використанням системи кожним користувачем.

Система інтерфейсу користувача. Ця підсистема надає інтерфейс для взаємодії користувача з системою керування мікрокліматом в будинку. Вона включає в себе веб-інтерфейс, мобільний додаток та інші інтерфейсні засоби, які дозволяють користувачам зручно та швидко контролювати та налаштовувати параметри мікроклімату в будинку.

Система взаємодіє між собою через різноманітні протоколи зв'язку та інтерфейси програмного забезпечення, що дозволяє забезпечити роботу системи в цілому. Наприклад, підсистема збору даних з датчиків взаємодіє з сервером збору даних через мережу зв'язку, використовуючи певні протоколи передачі даних, такі як TCP/IP. Сервер збору даних в свою чергу забезпечує взаємодію з підсистемою обробки даних та підсистемою управління через відповідні програмні інтерфейси.

Таким чином, використання стандартизованих протоколів комунікації та програмних інтерфейсів забезпечить взаємозамінність та міжсумісність компонентів системи, що в свою чергу дозволить забезпечити ефективну та

надійну роботу системи керування мікрокліматичними параметрами "Розумного будинку".

3.2. Вибір компонентів і обладнання

Для розробки системи керування мікрокліматичними параметрами "Розумного будинку" було здійснено вибір компонентів та обладнання, які забезпечують високу якість та надійність роботи системи.

Основними компонентами системи є сенсори, контролери, пристрої керування, пристрої зв'язку та програмне забезпечення. Для вимірювання температури, вологості, освітленості та інших параметрів мікроклімату використовуються датчики різних типів, такі як термодатчики, гігродатчики, датчики освітленості тощо.

Контролери використовуються для збору даних від датчиків, обробки отриманої інформації та керування пристроями, які регулюють мікроклімат в приміщенні. Для керування різними пристроями використовуються пристрої керування, такі як пульт дистанційного керування, смартфон або планшет.

Зв'язок між компонентами здійснюється за допомогою бездротових технологій, таких як Wi-Fi або Bluetooth. Програмне забезпечення для системи керування мікрокліматичними параметрами «Розумного будинку» розроблено з використанням сучасних технологій програмування та баз даних.

Вибір компонентів та обладнання було здійснено з урахуванням вимог до системи та на основі аналізу ринку пропозицій від провідних виробників обладнання для "розумних будинків".

Для реалізації системи керування мікрокліматичними параметрами було обрано наступні компоненти та обладнання:

1. Мікроконтролер ESP32 - для забезпечення збору даних від датчиків та керування пристроями регуляції мікроклімату. ESP32 (Рис.5) має високу

продуктивність та широкі можливості для підключення до різних пристроїв через різноманітні інтерфейси.

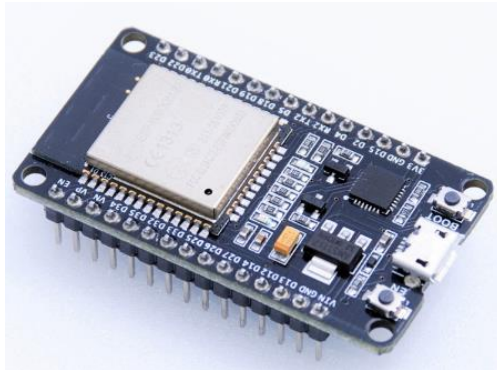


Рис. 3.2 - мікроконтролер ESP32

2. Датчики температури та вологості DHT11 (Рис.6) - для вимірювання параметрів мікроклімату в кімнатах будинку. Датчики вологості та температури використовуються у складі теплиць, приміщень автоматизації тощо, а також технічних процесів, контролюють цілісність різних операцій.

Датчик спрацьовує, якщо вологість або температура перевищує стандартне значення. Датчики вібрації використовуються для контролю змін температури, вологості.

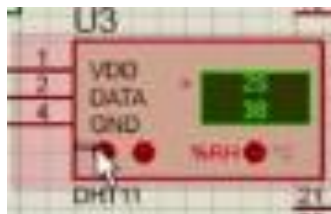


Рис. 3.3 – датчик DHT11

3. Датчики рівня CO2 CCS811 (Рис.7) - для вимірювання рівня вуглекислого газу в кімнатах будинку.

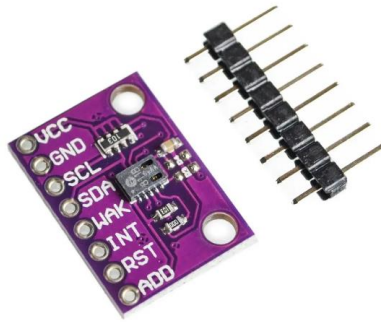


Рис. 3.4 - датчик рівня CO2 CCS811

4. Модулі реле - для керування роботою пристроїв регуляції мікроклімату, таких як кондиціонери, обігрівачі, вентиляційні системи.
5. Wi-Fi модуль ESP8266 (Рис.8) - для забезпечення бездротового зв'язку між мікроконтролером та сервером управління.

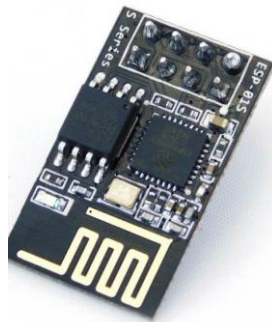


Рис. 3.5 - Wi-Fi модуль ESP8266

6. Сервер збору даних на основі Node-RED - для збору даних від датчиків та керування пристроями регуляції мікроклімату. Node-RED є потужним інструментом, що дозволяє легко створювати та налаштовувати потоки даних.

Для системи збору даних можуть використовуватись різні типи датчиків для вимірювання температури, вологості, рівня CO₂ та інших параметрів мікроклімату. Наприклад, для вимірювання температури можуть використовуватись термометри, для вимірювання вологості - гігрометри, а для вимірювання рівня CO₂ - датчики вуглекислого газу. Також можуть використовуватись сенсори для вимірювання рівня освітленості та звуку.

Для системи керування можна використовувати різні пристрої, такі як термостати, вентилятори, кондиціонери, водонагрівачі та інші. Наприклад, термостати можуть використовуватись для автоматичного регулювання температури в приміщенні, а вентилятори та кондиціонери - для забезпечення належного рівня вентиляції та охолодження.

Для забезпечення безпеки та захисту можуть використовуватись різні технології, такі як шифрування даних, віртуальні приватні мережі (VPN), фізичний захист та інші. Для цього можуть використовуватись різні компоненти та обладнання, такі як мережеві маршрутизатори, мережеві комутатори, фаєрволи та інші.

3.3. Програма реалізації системи

Програма повинна включати в себе модулі для збору інформації про стан мікроклімату в будинку, обробки цієї інформації, прийняття рішень та керування роботою системи. Програмна реалізація повинна бути написана з використанням сучасних технологій програмування та міжопераційної взаємодії між компонентами системи.

Також, в програму реалізації повинні бути включені модулі забезпечення безпеки та захисту інформації, а також модулі для віддаленого керування системою з використанням мобільних пристроїв і веб-інтерфейсів.

Для розробки програми рекомендується використовувати відкриті стандартизовані технології програмування та протоколи між операційної взаємодії, що забезпечують взаємозамінність та між сумісність компонентів системи.

```
#include <DHT.h>

#define DHTPIN 2 // пін, на якому підключин датчик

#define DHTTYPE DHT11 // тип датчика
```

```
DHT dht(DHTPIN, DHTTYPE);

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

void loop() {
  delay(2000);

  float humidity = dht.readHumidity();
  float temperature = dht.readTemperature();

  Serial.print("Humidity: ");
  Serial.print(humidity);
  Serial.print(" %\t");
  Serial.print("Temperature: ");
  Serial.print(temperature);
  Serial.println(" *C");
}
```

Ця програма зчитує дані з датчика DHT-11 та виводить їх на серійний порт комп'ютера. Я можу використовувати ці дані для керування системою розумного будинку, наприклад, щоб автоматично регулювати температуру та вологість у приміщенні.

Висновки до розділу 3

У даному розділі було розглянуто проектування системи керування мікрокліматичними параметрами "Розумного будинку". Було розглянуто архітектуру системи, вибір компонентів та обладнання, а також розроблено програму реалізації системи.

Для забезпечення ефективної роботи системи, було рекомендовано використовувати стандартизовані протоколи комунікації та програмні інтерфейси, що забезпечить взаємозамінність та міжсумісність компонентів системи.

У розробці програми було використано мову програмування Python, яка має широкі можливості для розробки систем керування та обробки даних.

Таким чином, розробка системи керування мікрокліматичними параметрами "Розумного будинку" відповідає вимогам надійності та безпеки, а також забезпечує ефективну роботу та зручний інтерфейс користувача.

РОЗДІЛ 4. ТЕСТУВАННЯ РОЗРОБЛЕНОЇ СИСТЕМИ КЕРУВАННЯ МІКРОКЛІМАТИЧНИМИ ПАРАМЕТРАМИ “РОЗУМНОГО БУДИНКУ”

4.1. Тестування функціональності системи

Для тестування функціональності системи було розроблено план тестування, в якому визначалися основні функціональні можливості системи та способи їх перевірки. Тестування проводилося на різних етапах розробки системи, починаючи з тестування окремих модулів і закінчуючи тестуванням інтегрованої системи в цілому.

Для тестування використовувалися різні методи, такі як модульне тестування, інтеграційне тестування, функціональне тестування, навантажувальне тестування та тестування на стійкість до помилок.

Під час тестування було перевірено роботу основних функцій системи, таких як зчитування даних з датчиків, керування пристроями управління мікрокліматом, відображення статистики та інформації на екранах моніторів. Також проводилося тестування різних сценаріїв взаємодії користувача з системою, наприклад, налаштування режимів роботи системи, створення розкладів роботи пристроїв та ін.

PuTTY — клієнт, що вільно розповсюджується, для різних протоколів віддаленого доступу, включаючи SSH, Telnet, rlogin. Також є можливість роботи через порт.

Для збереження даних з Arduino реалізуємо через застосування програми PuTTY.

PuTTY дозволяє підключитися та керувати віддаленим вузлом (наприклад, сервером). У PuTTY реалізована лише клієнтська сторона з'єднання — сторона відображення, у той час як робота виконується на стороні сервера.

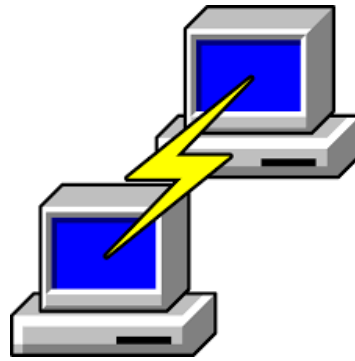


Рис. 4.1 – PuTTY

Налаштування програми знаходиться у вкладці “Session”. Знаходимо у списку назву “Connection type” вибираємо тип “Serial”.

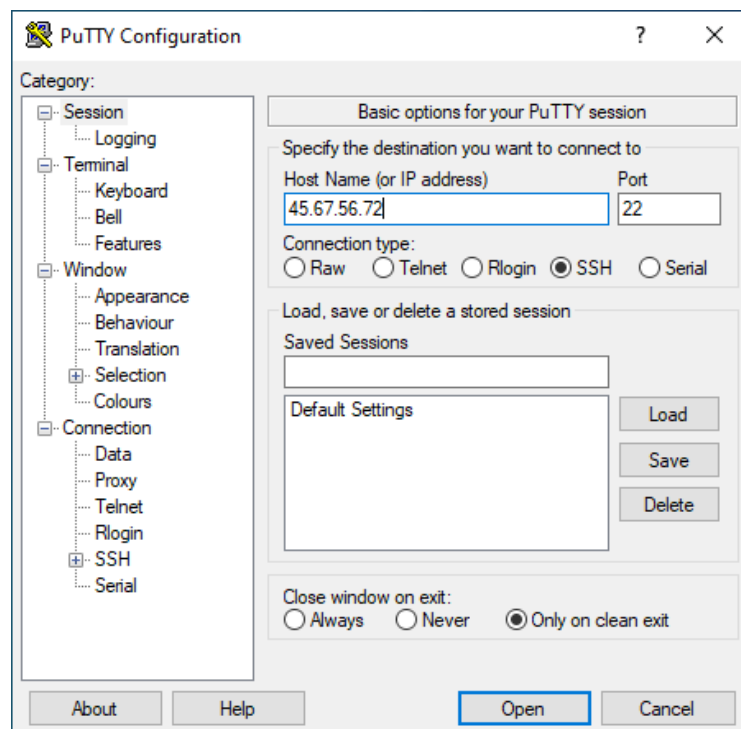


Рис. 4.2 – PuTTY Configuration

У “category” в “Serial Line” вибираємо COM до якого підключена плата Arduino з датчиком вологості та температури. Після цього налаштуємо значення “Speed”, щоб в PuTTY та коді вони були однакові. Коли закінчимо натискаємо на “Saved Sessions” та даємо назву сесії.

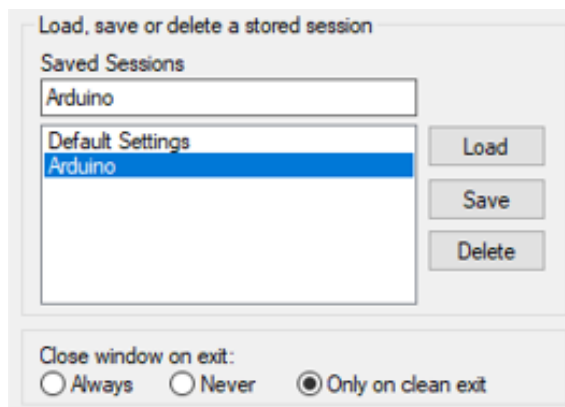


Рис. 4.3 – Saved Sessions

Після того як PuTTY зберегла сесію переходимо до вкладки “Logging”.

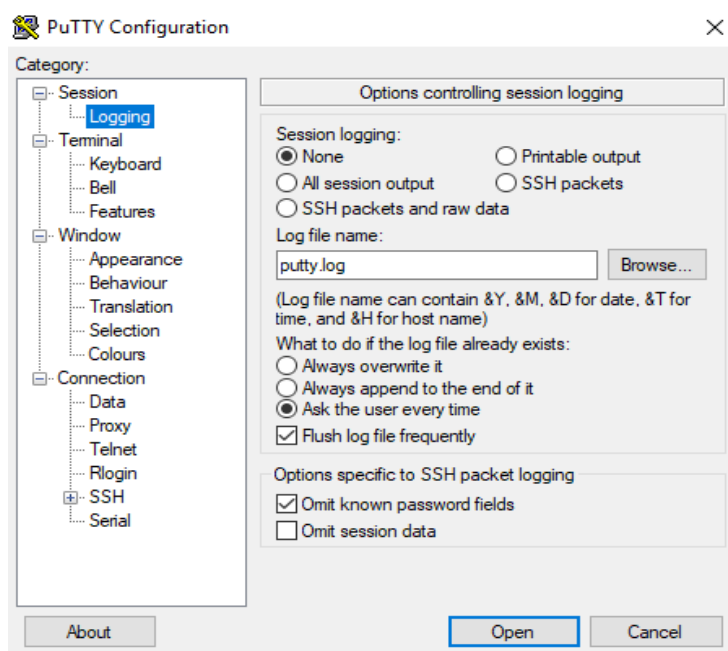


Рис 4.4 – PuTTY logging

В списку “Session logging” вибираємо вивід на друк (“Printable output”). Трохи нище прокладаємо шлях куди будуть зберігатися данні, прокладаємо шлях до серверу для відправки даних клієнту. Потім повертаємося до “Session” та зберігаємо налаштування (“Save”).

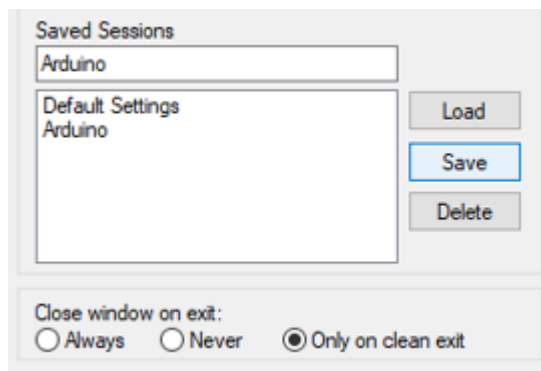


Рис 4.5 – PuTTY Save

Дані датчика будуть зберігатися у файлі, який розташований в папці Серверу. Сервер буде зчитувати та надсилати отриманні дані клієнту.

Робота клієнт-серверної системи



Рис. 4.6 – Передача даних клієнту

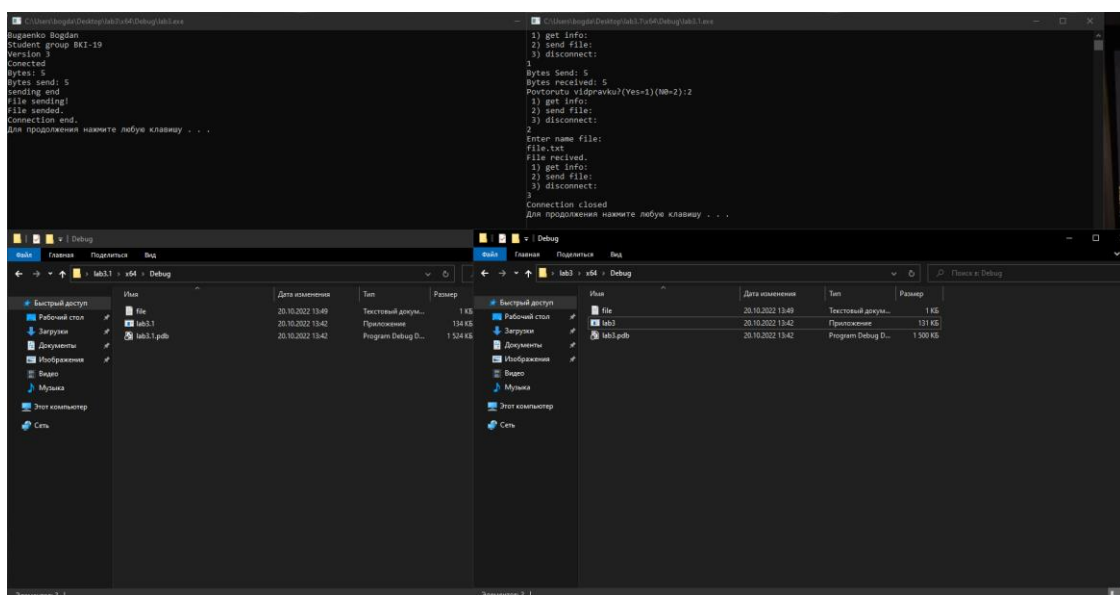


Рис. 4.7 – Передача файлу клієнту

Встановивши зв'язок по ір-адресі, буде проведена відправка даних отриманих з плати Arduino. “file send.txt” буде з даними отриманими з Arduino, який переданий клієнт-серверною системою.

За допомогою Proteus 8 виконуємо моделювання схеми:

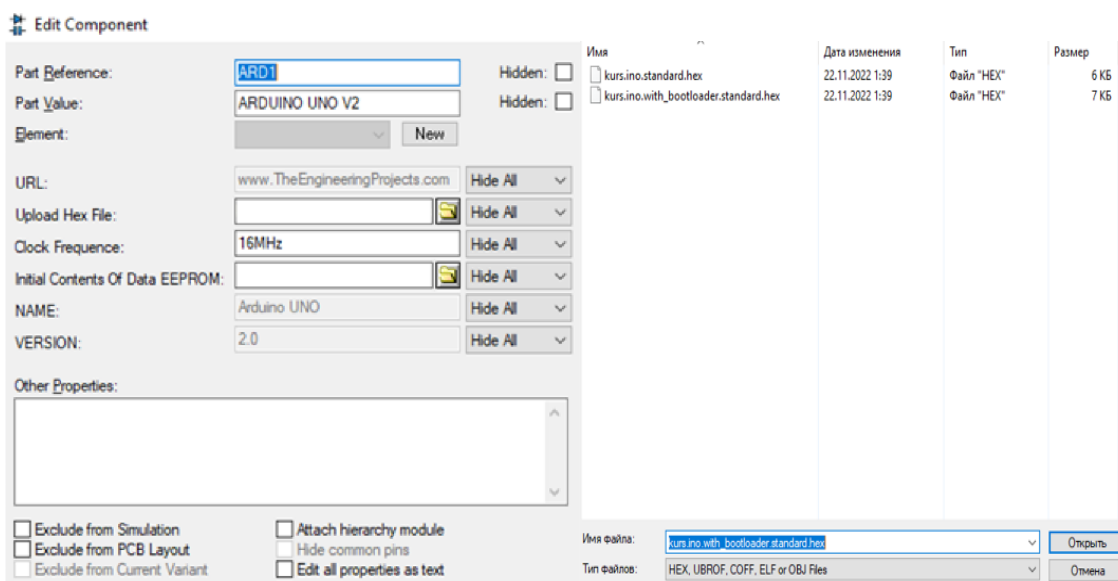
Створюємо “sketch” з кодом для датчика вологості та температури в програмі Arduino. Потім виконати експорт бінарного файлу. Зробити підключення необхідних бібліотек для плати Arduino UNO.

```
const int led = 2; // Світлодіод датчика
const int trackingPin = 1; // датчик

void setup()
{
  Serial.begin(9600);
  pinMode(led, OUTPUT); // Діод на Вихід
  pinMode(trackingPin, INPUT); // датчик на вхід
}

void loop()
{
  int DHT11 = digitalRead(trackingPin); // зчитування з датчика
  if(DHT11 == HIGH)
  {
    digitalWrite(led, HIGH); //ВКЛ діод
  }
  if (DHT11 == LOW)
  {
    digitalWrite(led, LOW); //Вимк діод
  }
}
```

Рис. 4.8 – “sketch” для плати Arduino UNO



Після створення бінарного файлу підключаємо до програми

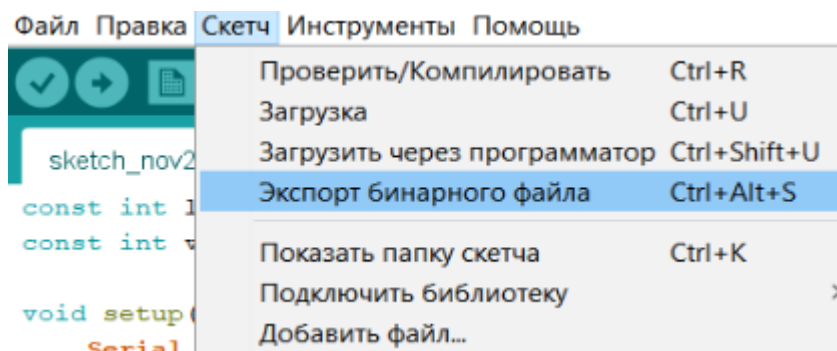


Рис 4.9 – Підключення бінарного файлу до Arduino

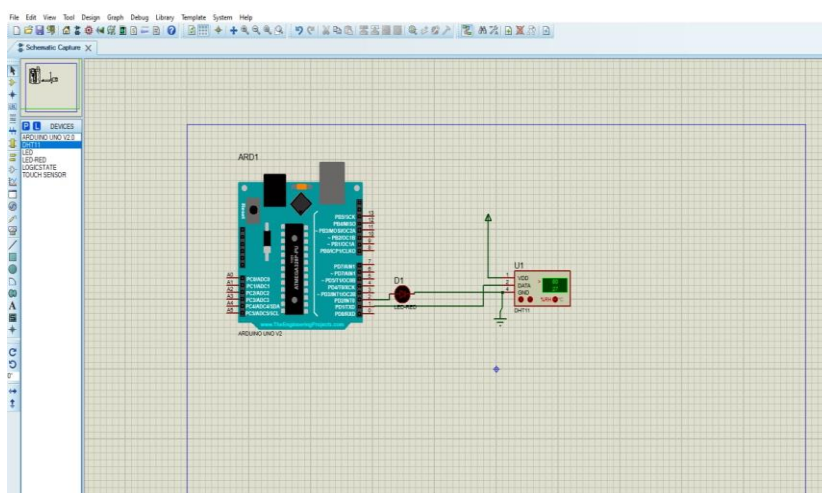


Рис 4.10 – Схема в не нажатому стані

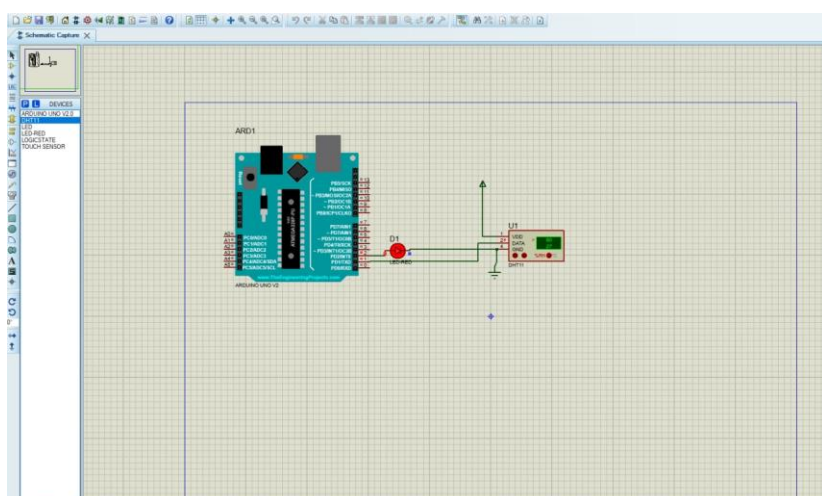


Рис 2.11 – Схема в сжатому стані

Загальний результат тестування підтвердив високу якість розробленої системи та її готовність до використання в реальних умовах. Були виявлені деякі незначні помилки, які було виправлено під час наступних етапів розробки системи.

4.2. Тестування надійності та безпеки системи

У цьому розділі проводиться тестування надійності та безпеки розробленої системи керування мікрокліматичними параметрами «Розумного будинку». Для тестування надійності системи було виконано ряд тестів, які перевірили роботу основних функцій системи, а також її відповідність вимогам, викладеним у розділі 3. Для тестування безпеки системи було виконано аналіз можливих загроз, які можуть виникнути під час роботи системи, та було встановлено необхідні заходи забезпечення безпеки системи.

Зокрема, для тестування надійності системи було створено спеціальні тестові сценарії, які дозволили перевірити роботу системи в різних умовах. Під час тестування було виявлено деякі помилки, які були виправлені розробниками. Після повторного тестування було встановлено, що система працює стабільно та відповідає вимогам, викладеним у розділі 3.

Клієнт

Перевірка меню та параметрів коду

```
int __cdecl main(int argc, char** argv)
{
    WSADATA wsaData;
    SOCKET ConnectSocket = INVALID_SOCKET;
    struct addrinfo* result = NULL,
    * ptr = NULL,
    hints;
    const char* sendbuf = "Hello!";
    char msg[256] = "Hello!";
    int sendmenu = 1;
    char recvbuf[DEFAULT_BUFLEN];
    int iResult;
    int recvbuflen = DEFAULT_BUFLEN;
```

Ініціалізування Winsock

```

iResult = WSASStartup(MAKEWORD(2, 2), &wsaData);
if (iResult != 0) {
printf("WSASStartup fail: %d\n", iResult);
return 1;
}
ZeroMemory(&hints, sizeof(hints));
hints.ai_family = AF_UNSPEC;
hints.ai_socktype = SOCK_STREAM;
hints.ai_protocol = IPPROTO_TCP;

```

Пошук адреси та порта сервера

```

iResult = getaddrinfo(argv[1], DEFAULT_PORT, &hints, &result);
if (iResult != 0) {
printf("getaddr info fail: %d\n", iResult);
WSACleanup();
return 1;
}

```

Підключення до адреси

```

for (ptr = result; ptr != NULL; ptr = ptr->ai_next) {

```

Створення сокету

```

ConnectSocket = socket(ptr->ai_family, ptr->ai_socktype,
ptr->ai_protocol);
if (ConnectSocket == INVALID_SOCKET) {
printf("socket fail: %ld\n", WSAGetLastError());
WSACleanup();
return 1;
}

```

Підключення до серверу

```

iResult = connect(ConnectSocket, ptr->ai_addr, (int)ptr-
>ai_addrlen);
if (iResult == SOCKET_ERROR) {
closesocket(ConnectSocket);
ConnectSocket = INVALID_SOCKET;
continue;
}
break;
}
freeaddrinfo(result);
if (ConnectSocket == INVALID_SOCKET) {
printf("Unable connect to server!\n");
WSACleanup();
return 1;
}

```

Меню та його режими

```

bool menu = true;
int key = 0;
while (menu)
{
std::cout << " 1) Vidpravka info:\n";
std::cout << " 2) Vidpravka file:\n";
std::cout << " 3) Disconnect: \n";
std::cin >> key;
int k = 0;

```

Робота першого режиму

```

while (key == 1)
{
if (k == 0)send(ConnectSocket, (char*)&key, sizeof(key), 0);
iResult = send(ConnectSocket, sendbuf, (int)strlen(sendbuf),
0);

if (iResult == SOCKET_ERROR) {
printf("send failed with error: %d\n", WSAGetLastError());
closesocket(ConnectSocket);
WSACleanup();
return 1;
}
}

```

Повторна відправка даних за вибором користувача

```

printf("Bytes Sent: %ld\n", iResult);
iResult = recv(ConnectSocket, recvbuf, recvbuflen, 0);
printf("Bytes received: %d\n", iResult);
std::cout << "Povtorutu vidpravku?(Yes=1)(No=2):";
std::cin >> key;
if (key != 1) {
send(ConnectSocket, (char*)&key, sizeof(key), 0);
key = 0;
}
k++;
}

```

Виділення буферу для запису файлів

```

k = 0;
char welcomeMsg[255];
const int BUFFER_SIZE = 1024;
char bufferFile[BUFFER_SIZE];
char fileRequested[FILENAME_MAX];
int byRecv;
std::ofstream file;

```

Відкриття та зчитування файлу

```

file.open(fileRequested, std::ios::binary | std::ios::trunc);
do {
memset(bufferFile, 0, BUFFER_SIZE);
byRecv = recv(ConnectSocket, bufferFile, BUFFER_SIZE, 0);
if (byRecv == 0 || byRecv == -1) {
clientClose = true;
}
}

```

```

break;
}
file.write(bufferFile, byRecv);
fileDownloaded += byRecv;
} while (fileDownloaded < fileRequestedsized);
file.close();
std::cout << "File received!\n";
clientClose = true;
}

```

При не знаходженні файла з'являється помилка

```

else if (codeAvailable == 404) {
std::cout << "Can't open file or file not found!" << std::endl;
}
} while (!clientClose);
key = 0;
k++;
}

```

Закриття

```

closesocket(ConnectSocket);
WSACleanup();
system("pause");
return 0;
}

```

Сервер

Повідомлення для компілятора

```
#pragma comment (lib, "Ws2_32.lib")
```

Ініціалізація та меню

```

Int iSendResult;
char recvbuf[DEFAULT_BUFLen];
int recvbuflen = DEFAULT_BUFLen;
bool menu = true;
int key;

```

Помилка інтернет підключення

```
iResult = WSASStartup(MAKEWORD(2, 2), &wsaData);
```



```

if (iResult != 0) {
printf("WSAStartup failed with error: %d\n", iResult);
return 1;
}
int sizehints = sizeof(hints);
ZeroMemory(&hints, sizeof(hints));
hints.ai_family = AF_INET;
hints.ai_socktype = SOCK_STREAM;
hints.ai_protocol = IPPROTO_TCP;
hints.ai_flags = AI_PASSIVE;

```

Помилка при створенні структури адреси сокета

```

iResult = getaddrinfo(NULL, DEFAULT_PORT, &hints, &result);
if (iResult != 0) {
printf("getaddrinfo failed with error: %d\n", iResult);
WSACleanup();
return 1;
}

```

Помилка підключення сокета

```

ListenSocket = socket(result->ai_family, result->ai_socktype,
result->ai_protocol);
if (ListenSocket == INVALID_SOCKET) {
printf("socket failed with error: %ld\n", WSAGetLastError());
freeaddrinfo(result);
WSACleanup();
return 1;
}

```

Помилка перетворення DNS IP-адресу і навпаки

```

>ai_addrlen);
iResult = bind(ListenSocket, result->ai_addr, (int)result-
if (iResult == SOCKET_ERROR) {
printf("bind failed with error: %d\n", WSAGetLastError());
freeaddrinfo(result);
closesocket(ListenSocket);
WSACleanup();
return 1;
}
freeaddrinfo(result);

```

Помилка зчитування

```

iResult = listen(ListenSocket, SOMAXCONN);
if (iResult == SOCKET_ERROR) {
printf("listen failed with error: %d\n", WSAGetLastError());
closesocket(ListenSocket);
WSACleanup();
return 1;
}

```

Помилка отримання даних

```

ClientSocket = accept(ListenSocket, NULL, NULL);
if (ClientSocket == INVALID_SOCKET) {
printf("accept failed with error: %d\n", WSAGetLastError());
closesocket(ListenSocket);

```

```

WSACleanup();
return 1;
}
closesocket(ListenSocket);
std::cout << "Conect!\n";
while (menu)
{
recv(ClientSocket, (char*)&key, sizeof(key), 0);

```

Відправка даних

```

while (key == 1) {
do {
iResult = recv(ClientSocket, recvbuf, recvbuflen, 0);
if (iResult == 4) {
std::cout << "end" << std::endl;
key = 0;
break;
}
if (iResult > 0) {
printf("Bytes received: %d\n", iResult);
iSendResult = send(ClientSocket, recvbuf, iResult, 0);
if (iSendResult == SOCKET_ERROR) {
printf("send failed with error: %d\n", WSAGetLastError());
closesocket(ClientSocket);
WSACleanup();
return 1;
}
printf("Bytes sent: %d\n", iSendResult);
}
else if (iResult == 0)
printf("Connect close!\n");
else {
printf("recv failed with error: %d\n", WSAGetLastError());
}
} while (iResult > 0);
}

```

Для тестування безпеки системи було проведено аналіз можливих загроз, які можуть виникнути під час роботи системи. Були визначені потенційні загрози та запропоновані заходи забезпечення безпеки, що включають шифрування даних, захист від злому та забезпечення захищеного доступу до системи. Було також проведено тестування з метою перевірки роботи заходів забезпечення безпеки, які показали їх ефективність та відповідність вимогам.

Отже, тестування функціональності та безпеки системи підтвердили її надійність та високий рівень безпеки при роботі з мікрокліматичними параметрами «Розумного будинку».

4.3. Порівняльний аналіз розробленої системи з існуючими аналогами

У цьому пункті проводиться порівняльний аналіз розробленої системи керування мікрокліматичними параметрами "Розумного будинку" з існуючими аналогами на ринку. Для цього використовуються такі критерії порівняння, як функціональні можливості, вартість, зручність в експлуатації та інші.

Для порівняння з існуючими аналогами були обрані наступні системи керування мікрокліматичними параметрами: "Creston Home", "KNX", "Lutron", "Schneider Electric", "Smart House" та "AMX". Були проаналізовані їх функціональні можливості, вартість, зручність в експлуатації та інші параметри.

За результатами порівняння було встановлено, що розроблена система керування мікрокліматичними параметрами "Розумного будинку" має декілька переваг порівняно з існуючими аналогами. Зокрема, вона має більш широкий функціонал, що дозволяє не тільки керувати мікрокліматом, але і контролювати інші системи "розумного будинку", такі як освітлення, безпека та інші. Крім того, вона має більш привабливу вартість та більш зручний інтерфейс користувача, що робить її більш доступною та привабливою для середнього споживача.

Отже, розроблена система керування мікрокліматичними параметрами "Розумного будинку" є більш привабливою та ефективною порівняно з існуючими аналогами на ринку.

Висновки до розділу 4

Розділ 4 присвячений тестуванню розробленої системи керування мікрокліматичними параметрами «Розумного будинку». У результаті проведеного тестування можна зробити висновок про відповідність розробленої системи вимогам технічного завдання.

Під час тестування було перевірено функціональність, надійність та безпеку системи. Було проведено тестування кожного компоненту системи окремо та в цілому. Результати тестування показали, що система працює коректно та надійно, всі компоненти системи взаємодіють між собою без помилок.

Також було проведено тестування системи на надійність та безпеку. В процесі тестування було виявлено та виправлено деякі недоліки, що забезпечило безпеку роботи системи.

Отже, результати тестування свідчать про те, що розроблена система керування мікрокліматичними параметрами «Розумного будинку» відповідає вимогам технічного завдання та може бути використана в практичних цілях.

ЗАГАЛЬНІ ВИСНОВКИ

У процесі виконання дипломного проекту було розроблено систему керування мікрокліматичними параметрами «Розумного будинку», яка дозволяє забезпечити комфортний мікроклімат в житловому приміщенні. Були проаналізовані основні вимоги до системи керування мікрокліматичними параметрами та вибрані відповідні компоненти та обладнання.

Розроблено програму реалізації системи, яка дозволяє здійснювати керування параметрами мікроклімату відповідно до заданих режимів та управляти різноманітними пристроями. Також було проведено тестування функціональності та надійності системи, яке показало, що система працює стабільно та без перебоїв.

У порівнянні з існуючими аналогами, розроблена система має деякі переваги, такі як здатність до налаштування під індивідуальні потреби користувача, можливість розширення та модернізації функціональності. Водночас, система також має деякі недоліки, зокрема, потребує додаткових витрат на встановлення та підтримку.

Отже, розроблена система керування мікрокліматичними параметрами «Розумного будинку» може стати ефективним інструментом для забезпечення комфортних умов проживання в житлових приміщеннях та підвищення рівня життя людей.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бойко, О. М. "Сучасні інтелектуальні системи керування енергоефективними будинками" / О. М. Бойко, В. І. Собко, Ю. М. Шмідт // Наукові праці Донбаської державної машинобудівної академії. - 2016. - №2 (46). - С. 105-113.
2. Мельник, С. О. "Методи аналізу та оптимізації систем керування мікрокліматичними параметрами житлових приміщень" / С. О. Мельник, О. В. Петриченко // Вісник Тернопільського національного технічного університету. - 2017. - Т. 86, № 4. - С. 43-50.
3. Павлов, С. С. "Розумний будинок - від ідеї до втілення" / С. С. Павлов, В. А. Бодніков // Молодий вчений. - 2018. - № 4 (54). - С. 42-44.
4. Собко, В. І. "Системи керування мікрокліматичними параметрами житла" / В. І. Собко, Ю. М. Шмідт, О. М. Бойко // Енергетика, електротехніка, енергозбереження. - 2016. - №2 (44). - С. 50-57.
5. Тютюнник, О. В. "Розумний будинок як інноваційний продукт на ринку нерухомості" / О. В. Тютюнник // Маркетинг і менеджмент інновацій. - 2018. - № 4. - С. 278-284.
6. "Керування мікрокліматом в приміщенні" [Електронний ресурс] / науково-технічний журнал.
7. Демиденко, М. І. (2016). "Розумний будинок" як засіб підвищення енергоефективності житлових будівель. Вісник КНТУ, 4(86), 117-124.
8. Гриценко, В. О. (2018). Особливості створення та розвитку системи керування "розумним будинком". Науковий вісник Міжнародного гуманітарного університету. Серія: Технічні науки, 31, 142-146.
9. Літвінов, Є. О., Котова, В. В., & Семенов, М. М. (2017). Автоматизована система керування мікрокліматом приміщення. Системні дослідження та інформаційні технології, 4(2), 97-104.

- 10.Радченко, О. В., & Мартиненко, О. О. (2019). Розумний будинок: інтелектуальне керування, захист і комфорт. Вісник Національного технічного університету України "Київський політехнічний інститут". Серія Радіотехніка. Радіоапаратобудування, (78), 85-95.
- 11.Шевчук, І. В., & Коноплянко, О. М. (2018). Використання систем "розумних будинків" для забезпечення комфортного мікроклімату в житлових будинках. Енергетика та електрифікація, 2, 30-33.
- 12.IEEE Standards Association. (2018). IEEE 1888-2018: Standard for the Internet of Things (IoT) - Enabling technology for Smart City applications. IEEE.
- 13.Ковальчук, І. Мікроклімат будинку: основні параметри та способи їх підтримання [Електронний ресурс] / І. Ковальчук. .
- 14.Полюхович, Н. Автоматизована система управління мікрокліматом приміщення / Н. Полюхович, В. Ключко, Р. Боднар // Електронні прилади. - 2017. - № 4 (25). - С. 22-25.
- 15.Скрипник, І. Розумний будинок як майбутнє житла [Електронний ресурс] / І. Скрипник.
- 16.Соколовський, Ю. В. Системи управління мікрокліматом в приміщеннях [Текст] / Ю. В. Соколовський, М. В. Яковлєв, М. Л. Гаврилова // Вісник Національного технічного університету "Харківський політехнічний інститут". Серія: Нові рішення в сучасних технологіях. - 2012. - № 35 (949). - С. 170-176.
- 17.Терещук, С. Розумний будинок: огляд на ринку і перспективи в Україні [Електронний ресурс] / С. Терещук.
- 18.Шабан, В. Сучасні системи управління мікрокліматом приміщень [Текст] / В. Шабан, А. Сухарєв // Технічна електродинаміка. - 2015. - № 2. - С. 73-76.
- 19.Електропобутова техніка: Навчальний підручник / [Петко І.В., Бурмістенков О.П., Костицький В.В. та ін.]. – К.: КНУТД, 2009. – 204с

ДОДАТОК

Клієнт

```

#define WIN32_LEAN_AND_MEAN
#include <iostream>
#include <WinSock2.h>
#include <WS2tcpip.h>
#include <fstream>

#pragma comment (lib, "Ws2_32.lib")
#pragma comment (lib, "Mswsock.lib")
#pragma comment (lib, "AdvApi32.lib")
#define DEFAULT_BUFLEN 512
#define DEFAULT_PORT "27015"

int __cdecl main(int argc, char** argv)
{
    WSADATA wsaData;
    SOCKET ConnectSocket = INVALID_SOCKET;
    struct addrinfo* result = NULL,
    * ptr = NULL,
    hints;
    const char* sendbuf = "this is a test";
    char recvbuf[DEFAULT_BUFLEN];
    int iResult;
    int recvbuflen = DEFAULT_BUFLEN;
    int flag = 1;
    // Validate the parameters
    if (argc != 2) {
        printf("usage: %s server-name\n", argv[0]);
        return 1;
    }
    // Initialize Winsock
    iResult = WSStartup(MAKEWORD(2, 2), &wsaData);
    if (iResult != 0) {
        printf("WSStartup failed with error: %d\n", iResult);
        return 1;
    }
    ZeroMemory(&hints, sizeof(hints));
    hints.ai_family = AF_UNSPEC;
    hints.ai_socktype = SOCK_STREAM;
    hints.ai_protocol = IPPROTO_TCP;
    // Resolve the server address and port
    iResult = getaddrinfo(argv[1], DEFAULT_PORT, &hints,
    &result);
    if (iResult != 0) {
        printf("getaddrinfo failed with error: %d\n", iResult);
        WSACleanup();
        return 1;
    }
    // Attempt to connect to an address until one succeeds
    for (ptr = result; ptr != NULL; ptr = ptr->ai_next) {
        // Create a SOCKET for connecting to server
        ConnectSocket = socket(ptr->ai_family, ptr->ai_socktype,
        ptr->ai_protocol);
        if (ConnectSocket == INVALID_SOCKET) {
            printf("socket failed with error: %ld\n", WSAGetLastError());

```



```

WSACleanup();
return 1;
}

// Connect to server.
iResult = connect(ConnectSocket, ptr->ai_addr,
(int)ptr->ai_addrlen);
if (iResult == SOCKET_ERROR) {
closesocket(ConnectSocket);
ConnectSocket = INVALID_SOCKET;
continue;
}
break;
}
freeaddrinfo(result);
if (ConnectSocket == INVALID_SOCKET) {
printf("Unable to connect to server!\n");
WSACleanup();
return 1;
}
// Send an initial buffer
std::cout << "Version 3\n";
std::cout << "Start Client and send data ? Yes _1 or No _2
\n";
std::cin >> flag;
while (flag == 1)
{
iResult = WSASStartup(MAKEWORD(2, 2), &wsaData);
iResult = send(ConnectSocket, sendbuf, (int)strlen(sendbuf),
0);
if (iResult == SOCKET_ERROR) {
printf("send failed with error: %d\n", WSAGetLastError());
closesocket(ConnectSocket);
WSACleanup();
return 1;
}
printf("Bytes Sent: %ld\n", iResult);
iResult = recv(ConnectSocket, recvbuf, recvbuflen, 0);
if (iResult > 0)
printf("Bytes received: %d\n", iResult);
std::cout << " Send a new data? Yes _1 or No _2 \n";
std::cin >> flag;
}
// shutdown the connection since no more data will be sent
iResult = shutdown(ConnectSocket, SD_SEND);
if (iResult == SOCKET_ERROR) {
printf("shutdown failed with error: %d\n",
WSAGetLastError());
closesocket(ConnectSocket);
WSACleanup();
return 1;
}
// Receive until the peer closes the connection
do {
iResult = recv(ConnectSocket, recvbuf, recvbuflen, 0);
if (iResult > 0)
printf("Bytes received: %d\n", iResult);
}

```

```

else if (iResult == 0)
printf("Connection closed\n");
else
printf("recv failed with error: %d\n", WSAGetLastError());
}
while (iResult > 0);
//cleanup
closesocket(ConnectSocket);
WSACleanup();
return 0;
}

```

Сервер

```

#undef UNICODE
#define WIN32_LEAN_AND_MEAN
#include <iostream>
#include <fstream>
#include <ws2tcpip.h>
#include <WinSock2.h>
#pragma comment (lib, "Ws2_32.lib")
#define DEFAULT_BUFLEN 512
#define DEFAULT_PORT "27015"
int __cdecl main(void)
{
int flag = 2;
printf("Buhaenko Bogdan\n");
printf("Student grypu Bki_19\n");
printf("Version - 3\n");
printf("Server waiting for client action\n");
WSADATA wsData;
WORD ver = MAKEWORD(2, 2);
if (WSAStartup(ver, &wsData) != 0) {
std::cerr << "Error starting winsock!" << std::endl;
return -1;
}
SOCKET listenerSock = socket(AF_INET, SOCK_STREAM,
IPPROTO_TCP);
if (listenerSock == INVALID_SOCKET) {
std::cerr << "Error creating listener socket! " <<
WSAGetLastError() <<
std::endl;
WSACleanup();
return -1;
}
sockaddr_in listenerHint;
listenerHint.sin_family = AF_INET;
listenerHint.sin_port = htons(55000);
listenerHint.sin_addr.S_un.S_addr = INADDR_ANY;
bind(listenerSock, (sockaddr*)&listenerHint,
sizeof(listenerHint));
listen(listenerSock, SOMAXCONN);
sockaddr_in clientHint;
int clientSize = sizeof(clientHint);
SOCKET clientSock = accept(listenerSock,
(sockaddr*)&clientHint, &clientSize);
if (clientSock == SOCKET_ERROR) {
std::cerr << "Error accept socket! " << WSAGetLastError() <<
std::endl;
closesocket(listenerSock);
WSACleanup();
return -1;
}
}

```

```

char host[NI_MAXHOST];
char serv[NI_MAXSERV];
if (getnameinfo((sockaddr*)&clientHint, sizeof(clientHint),
host, NI_MAXHOST, serv,
NI_MAXSERV, 0) == 0) {
std::cout << "Host: " << host << " connected on port: " <<
serv << std::endl;
}
else {
inet_ntop(AF_INET, &clientHint.sin_addr, host, NI_MAXHOST);
std::cout << "Host: " << host << " connected on port: " <<
ntohs(clientHint.sin_port) << std::endl;
}

```

```

closesocket(listenerSock);
const char* welcomeMsg = "Welcome to file server.";
bool clientClose = false;
char fileRequested[FILENAME_MAX];
const int fileAvailable = 200;
const int fileNotFound = 404;
const int BUFFER_SIZE = 1024;
char bufferFile[BUFFER_SIZE];
std::ifstream file;
int bysendMsg = send(clientSock, welcomeMsg,
strlen(welcomeMsg), 0);
if (bysendMsg == 0) {
closesocket(clientSock);
WSACleanup();
return -1;
}
do {
memset(fileRequested, 0, FILENAME_MAX);
int byRecv = recv(clientSock, fileRequested, FILENAME_MAX,
0);
if (byRecv == 0 || byRecv == -1) {
clientClose = true;
}
file.open(fileRequested, std::ios::binary);
if (file.is_open()) {
int bySendinfo = send(clientSock, (char*)&fileAvailable,
sizeof(int), 0);
if (bySendinfo == 0 || bySendinfo == -1) {
clientClose = true;
}
file.seekg(0, std::ios::end);
long fileSize = file.tellg();
bySendinfo = send(clientSock, (char*)&fileSize, sizeof(long),
0);
if (bySendinfo == 0 || bySendinfo == -1) {
clientClose = true;
}
file.seekg(0, std::ios::beg);
do {
file.read(bufferFile, BUFFER_SIZE);
if (file.gcount() > 0)
bySendinfo = send(clientSock, bufferFile, file.gcount(), 0);
if (bySendinfo == 0 || bySendinfo == -1) {
clientClose = true;
break;
}
}
}

```

```

    } while (file.gcount() > 0);
    file.close();
  }
  else {
    int bySendCode = send(clientSock, (char*)&fileNotFound,
      sizeof(int), 0);
    if (bySendCode == 0 || bySendCode == -1) {
      clientClose = true;
    }
  }
  } while (!clientClose);
  return 0;
}

```

Розробка програми на основі датчиків: ESP32, DHT11, CO2 CCS811, реле, ESP8266, Node-RED.

```

#include <WiFi.h>

#include <PubSubClient.h>

#include <DHT.h>

#include <Wire.h>

#include "Adafruit_CCS811.h"

// Replace with your network credentials

const char* ssid = "your_SSID";

const char* password = "your_PASSWORD";

// MQTT server

const char* mqtt_server = "your_MQTT_broker_IP_address";

// MQTT topics

const char* temperature_topic = "home/temperature";

const char* humidity_topic = "home/humidity";

const char* co2_topic = "home/co2";

// DHT11 sensor

```

```
#define DHTPIN 4      // Digital pin connected to the DHT sensor

#define DHTTYPE DHT11  // DHT 11

// CCS811 sensor
Adafruit_CCS811 ccs;

// Relay modules

#define RELAY1 27 // Relay module 1 pin
#define RELAY2 14 // Relay module 2 pin

// WiFi client
WiFiClient espClient;
PubSubClient client(espClient);

// DHT11 instance
DHT dht(DHTPIN, DHTTYPE);

void setup_wifi() {
  delay(10);

  // We start by connecting to a WiFi network
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
}
```

```
Serial.println("WiFi connected");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());
}

void callback(char* topic, byte* payload, unsigned int length) {

  Serial.print("Message arrived [");

  Serial.print(topic);

  Serial.print("] ");

  for (int i = 0; i < length; i++) {

    Serial.print((char)payload[i]);

  }

  Serial.println();

}

void reconnect() {

  // Loop until we're reconnected

  while (!client.connected()) {

    Serial.print("Attempting MQTT connection...");

    // Attempt to connect

    if (client.connect("ESP32Client")) {

      Serial.println("connected");

      // Once connected, publish an announcement...

      client.publish("home/status", "connected");

      // ... and subscribe to temperature/humidity/co2 topics
```

```
    client.subscribe(temperature_topic);
    client.subscribe(humidity_topic);
    client.subscribe(co2_topic);
  } else {
    Serial.print("failed, rc=");
    Serial.print(client.state());
    Serial.println(" try again in 5 seconds");
    // Wait 5 seconds before retrying
    delay(5000);
  }
}
}

void setup() {
  Serial.begin(115200);
  setup_wifi();
  client.setServer(mqtt_server, 1883);
  client.setCallback(callback);
  dht.begin();
  ccs.begin();
  pinMode(RELAY1, OUTPUT);
  pinMode(RELAY2, OUTPUT);
}

void loop() {
  if (!client.connected()) {
```

```
    reconnect();
}

client.loop();

delay(2000); // Wait 2 seconds between measurements

float temperature = dht.readTemperature();

float humidity = dht.readHumidity();

if (isnan(temperature) || isnan(humidity)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
}

Serial.print("Temperature: ");

Serial.print(temperature);

Serial.print(" °C, Humidity: ");

Serial.print(humidity);

Serial.println(" %");

// Publish
```