

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Навчально-науковий
інститут електроенергетики
(інститут)
Факультет інформаційних технологій
(факультет)
Кафедра інформаційних технологій та комп'ютерної інженерії
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня бакалавра

студента Голубицького Артема Сергійовича
(ПІБ)

академічної групи 123-21ск-1
(шифр)

спеціальності 123 Комп'ютерна інженерія
(код і назва спеціальності)

за освітньо-професійною програмою 123 Комп'ютерна інженерія
(офіційна назва)

на тему «Кіберфізична система забезпечення мікроклімату розумного будинку на базі ESP8266 з детальною розробкою веб-інтерфейсу.»
(назва за наказом ректора)

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	доц. Бешта Д.О			
спеціальної частини	доц. Бешта Д.О			
розділів:				
розробка апаратної частини	доц. Бешта Д.О			
розробка корпоративної мережі	ас. Панферова Я.В.			

Рецензент				
-----------	--	--	--	--

Нормоконтролер	проф. Цвіркун Л.І.			
----------------	--------------------	--	--	--

Дніпро
2024

ЗАТВЕРДЖЕНО:

завідувач кафедри
інформаційних технологій
та комп'ютерної інженерії

(повна назва)

Гнатушенко В.В.

(підпис)

(прізвище, ініціали)

« »

2024 року

ЗАВДАННЯ
на кваліфікаційну роботу
ступеня бакалавр

студента Голубицького А.С. академічної групи 123-21ск1
(прізвище та ініціали) (шифр)

спеціальності 123 Комп'ютерна інженерія

за освітньо-професійною програмою 123 Комп'ютерна інженерія
офіційна назва

на тему «Кіберфізична система забезпечення мікроклімату розумного будинку
на базі ESP8266 з детальною розробкою веб-інтерфейсу.»

затверджену наказом ректора НТУ «Дніпровська політехніка» від 29.04.2024 № 375-с

Розділ	Зміст	Термін виконання
Стан питання та постановка завдання	На основі матеріалів виробничих практик, інших науково-технічних джерел розглянути призначення та завдання побудови мережі керуючої компанії	10.05.2024
Розробка апаратної частини	Розробити вимоги до функцій, виконуваними системою забезпечення мікроклімату	17.05.2024
Розробка корпоративної мережі	Побудувати в Packet Tracer модель корпоративної мережі компанії, виконати налаштування та перевірку роботи системи	24.05.2024
Розробка компонента системи	Розробити кіберфізичну систему забезпечення мікроклімату	31.05.2024

Завдання видано

Бешта Д.О

(підпис керівника)

(прізвище, ініціали)

Дата видачі 25.01.2024

Дата подання до екзаменаційної комісії 14.06.2024

Прийнято до виконання

Голубицький А.С.

(підпис студента)

(прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 110 с., 45 рис., 8 табл., 2 додатки, 8 джерел.

КІБЕРФІЗИЧНА СИСТЕМА, МАРШРУТИЗАТОР, КОМУТАТОР, CISCO, CISCO PACKET TRACER, ІНТЕРНЕТ РЕЧЕЙ, ІОТ, МІКРОКЛІМАТ, РОЗУМНИЙ БУДИНОК.

Об'єкт розробки – кіберфізична система з реалізацією побудови та налаштування мережі комплексу контролю та моніторингу середовища мікроклімату.

Мета роботи – розробка та реалізація кіберфізичної системи, здатної надавати змогу для моніторингу мікроклімату в розумному будинку, враховуючи показники температури, вологості та інші параметри, а також створення зручного веб-інтерфейсу для моніторингу та керування системою.

Розроблено кіберфізичну систему, яка здатна збирати дані з датчиків, аналізувати їх та приймати рішення щодо керування кліматичними пристроями для підтримки заданих параметрів мікроклімату. Новизна роботи полягає у використанні доступного та енергоефективного мікроконтролера ESP8266, а також у створенні інтуїтивно зрозумілого веб-інтерфейсу з використанням фреймворку Python. Система може бути впроваджена в розумних будинках, офісах, готелях та інших приміщеннях, де важливо підтримувати комфортний мікроклімат.

ЗМІСТ

Перелік скорочень, умовних познач, одиниць і термінів	7
Вступ.....	8
1 Стан питання і постановка завдання	9
1.1 Стисла характеристика галузі (сфери)	9
1.2 Принципи та технічні способи інформаційного забезпечення кіберфізичної системи та мережі.....	15
1.3 Огляд існуючих кіберфізичних систем забезпечення мікроклімату	15
1.4 Завдання і мета роботи.....	17
1.5 Визначення можливих напрямків рішення поставлених завдань	17
1.5.1 Розробка архітектури мережі	17
1.5.2 Розробка кіберфізичної системи.....	18
1.6 Обґрунтування вибраного напрямку інженерного рішення.....	19
2 Розробка апаратної частини кіберфізичної системи	20
2.1 Технічні вимоги до кіберфізичної системи	20
2.1.1 Вимоги до системи в цілому	20
2.1.1.1 Вимоги до структури і функціонуванню системи	20
2.1.1.1.1 Перелік підсистем, їхнє призначення й основні характеристики, вимоги до числа рівнів ієрархії та ступені централізації системи.....	20
2.1.1.1.2 Вимоги до способів і засобів зв'язку між компонентами кіберфізичної системи	21
2.1.1.1.3 Вимоги до характеристик взаємозв'язків створюваної системи із суміжними системами.....	22
2.1.1.1.4 Вимоги до режимів функціонування системи	22
2.1.1.1.5 Вимоги до діагностування системи	23
2.1.1.1.6 Перспективи розвитку, модернізації системи	25
2.1.1.2 Вимоги до показників призначення	26
2.1.1.3 Вимоги до патентної чистоти.....	27

2.1.1.4	Додаткові вимоги.....	27
2.1.2	Вимоги функцій, виконуваним системою.....	28
2.1.2.1	Підсистема збору даних.....	28
2.1.3	Вимоги до видів забезпечення системи	30
2.1.3.1	Вимоги до математичного забезпечення	30
2.1.3.2	Вимоги до інформаційного забезпечення.....	30
2.1.3.3	Вимоги до лінгвістичного забезпечення.....	31
2.1.3.4	Вимоги до технічного забезпечення.....	32
2.1.3.5	Вимоги до організаційного забезпечення.....	33
2.1.3.6	Вимоги до методичного забезпечення	34
2.1.3.7	Вимоги до задач (налаштувань), які виконує система	34
2.2	Розробка апаратної частини комп'ютерної системи	36
2.2.1	Розробка загальної архітектури мережі комплексу	36
2.2.2	Вибір і обґрунтування структурної схеми комплексу технічних засобів системи	38
2.2.3	Розробка специфікації апаратних засобів кіберфізичної системи	40
2.2.4	Розрахунок інтенсивності трафіку вихідного трафіку найбільшої локальної мережі керуючої компанії	41
3	Розробка корпоративної мережі	44
3.1	Проектування логічної топології мережі	44
3.2	Вибір та опис мережного обладнання	46
3.3	Розрахунок схеми адресації корпоративної мережі	46
3.4	Вибір та налаштування способу маршрутизації	50
3.4.1	Базове налаштування конфігурації пристроїв	50
3.4.2	Налаштування протоколу OSPF та DHCP для маршрутизаторів корпоративної мережі	52
3.4.3	Налаштування роботи Інтернет	53
3.5	Налаштування мереж VLAN, маршрутизації між VLAN	54
3.6	Перевірка комп'ютерної мережі комплексу.....	56

4 Розробка компонента системи.....	63
4.1 Розробка контролеру кіберфізичної системи забезпечення мікроклімату	63
4.1.1 Опис датчиків та їх характеристик.....	63
4.1.2 Розробка схеми підключення датчиків.....	64
4.1.2.1 Розробка схеми підключення внутрішніх датчиків	64
4.1.2.2 Розробка схеми підключення зовнішніх датчиків	66
4.1.2.3 Розробка схеми підключення виконавчого пристрою	68
4.2 Розробка функціонального модуля	69
4.2.1 Розробка внутрішнього компонента системи.....	69
4.2.2 Розробка зовнішнього компонента системи	70
4.3 Розробка веб сторінок інтерфейсу користувача	71
4.3.1 Розробка головної сторінки	71
4.3.2 Розробка сторінки авторизації	73
4.3.3 Розробка сторінки з ретельною інформацією.....	75
4.3.4 Розробка сторінки налаштувань	78
Висновки	81
Список використаних джерел	82
Додаток А. Текст HTML коду веб сторінок	83
Додаток Б. Текст програми налаштування мережевого обладнання.....	22

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАК, ОДИНИЦЬ І ТЕРМІНІВ

IOT – Internet of Things Інтернет речей

ПК – персональний комп'ютер

HTTP – протокол передачі гіпертексту

JSON – JavaScript Object Notation Формат обміну даними

КФС – Кіберфізична система, що поєднує фізичні та обчислювальні компоненти

MQTT – Message Queuing Telemetry Transport, мережевий протокол обміну повідомленнями

LAN – локальна мережа

WAN – глобальна мережа

°C – градус Цельсія

% – відсоток

Rpm – кількість вуглецю

ГБ (GB) – гігабайт

МГц (MHz) – мегагерц

мс (ms) – мілісекунда

м² (m²) – квадратний метр

ВСТУП

Сучасні тенденції розвитку суспільства висувають на перший план питання енергоефективності та комфорту житлових приміщень. Розумні будинки, оснащені інтелектуальними системами управління, дозволяють вирішувати ці завдання, забезпечуючи оптимальний мікроклімат, знижуючи енергоспоживання та підвищуючи якість життя мешканців.

Світовий досвід демонструє зростаючий інтерес до кіберфізичних систем (КФС) у сфері автоматизації житла. КФС, що об'єднують фізичні компоненти (датчики, виконавчі механізми) з обчислювальними ресурсами, забезпечують гнучке та адаптивне управління мікрокліматом, враховуючи індивідуальні потреби користувачів та зовнішні умови.

Актуальність кваліфікаційної роботи полягає у необхідності розробки доступних та ефективних рішень для забезпечення мікроклімату в розумних будинках. Використання мікроконтролера ESP8266, що відрізняється низькою вартістю та широкими можливостями, дозволяє створити систему, доступну для широкого кола споживачів.

Метою кваліфікаційної роботи є розробка кіберфізичної системи забезпечення мікроклімату розумного будинку на базі ESP8266 та створення зручного веб-інтерфейсу для керування системою. Розроблена система дозволить автоматично підтримувати задані параметри температури та вологості, враховуючи дані з датчиків, а також забезпечує можливість віддаленого моніторингу та керування через веб-інтерфейс.

1 СТАН ПИТАННЯ І ПОСТАНОВКА ЗАВДАННЯ

1.1 Стисла характеристика галузі (сфери)

1.1.1 Огляд сфери застосування

Системи з регулювання та управління мікрокліматом є важливою частиною сучасних розумних будинків, офісних приміщень, готелів, медичних установ та інших об'єктів, де важливо підтримувати комфортні умови для проживання, роботи та відпочинку. Ці системи забезпечують не лише комфорт, але й енергоефективність, дозволяючи знизити витрати на опалення та кондиціонування повітря.

Призначення таких систем полягає в забезпеченні оптимальних умов мікроклімату у приміщенні, враховуючи потреби та зручність користувачів. Вони включають в себе автономний моніторинг температури, вологості, тиску, вміст вуглецю в повітрі, якість повітря, освітлення та іншими аспектами середовища, необхідними для здоров'я людей та приладів, а також надає можливість збору даних та управління через веб-інтерфейс. Всі ці функції включає у себе автономні IoT системи.

Дана галузь охоплює такі ключові напрямки, а саме:

Оптимізація мікроклімату що включає в себе розробку систем контролю за температурою, вологості, рівнем CO₂ та іншими параметрами для створення комфортного та здорового мікроклімату у приміщенні, що сприяє здоров'ю та продуктивності мешканців.

Використання передових технологій завдяки впровадженню технологій IoT, що дозволяють зв'язувати пристрої та забезпечувати їх взаємодію через Інтернет.

Розробка систем, що споживають мінімальну кількість енергії та максимально зменшують викиди шкідливих речовин, сприяючи екологічно чистому життю.

Створення ефективних систем управління та моніторингу систем управління та моніторингу які забезпечують зручність, ефективність та екологічність в управлінні мікрокліматом приміщення.

Об'єктом впровадження є офісні приміщення, житлові приміщення, квартири та будинки. Ключовими зацікавленими сторонами у впровадженні та експлуатації системи є об'єднання співвласників багатоквартирних будинків (ОСББ) та житлово-комунальні господарства (ЖКГ), керуючі компанії у сфері надання послуг обслуговування будинків та комплексів.

1.1.2 Характеристика умов застосування кіберфізичних систем у житлових комплексах

ОСББ та ЖКГ відповідають за управління та обслуговування житлових будинків, забезпечення їх належного технічного стану та комфорту мешканців. Впровадження кіберфізичної системи забезпечення мікроклімату дозволить їм вирішити ряд важливих завдань:

- Підвищення комфорту мешканців за рахунок забезпечення оптимальних параметрів температури та вологості в квартирах та будинках яке сприятиме підвищенню комфорту проживання та задоволеності мешканців;
- Автоматичне керування кліматичними пристроями, яке дозволить оптимізувати їх роботу та знизити витрати на опалення та кондиціонування;
- Можливість віддаленого моніторингу та керування системою дозволить ОСББ та ЖКГ оперативно реагувати на зміни умов та забезпечувати безперебійну роботу системи;

Умови застосування кіберфізичної системи в житлових приміщеннях характеризуються наступними особливостями:

- Температура та вологість зовнішнього повітря можуть значно змінюватися протягом доби та року, що вимагає адаптивності системи до цих змін;
- Система повинна бути захищена від несанкціонованого доступу та забезпечувати безпечну експлуатацію кліматичних пристроїв.

1.2 Характеристика об'єкта впровадження кіберфізичної системи забезпечення мікроклімату

1.2.1 Огляд організаційної структури компанії у сфері надання послуг обслуговування комплексів

Головний офіс обслуговуючої компанії складається з декількох підрозділів, та відстань від головного офісу та технічних приміщень до приміщень в яких розгортається система може бути розташована на відстані декілька кілометрів.

Організаційна структура включає декілька ключових ролей та взаємодій.

Мешканці квартир: Є кінцевими користувачами системи. Вони взаємодіють з системою через веб-інтерфейс, налаштовуючи бажані параметри мікроклімату та отримуючи інформацію про поточний стан системи.

ОСББ або ЖКГ відповідають за загальне управління розгортанням системи, включаючи її встановлення, налаштування та обслуговування. Вони можуть мати обмежений доступ до веб-інтерфейсу з можливістю моніторингу стану системи в усіх квартирах, а також отримання повідомлень про несправності та аварійні ситуації.

Технічні спеціалісти залучаються для встановлення, налаштування та обслуговування системи. Вони можуть мати доступ до більш детальної інформації про роботу системи та можливість вносити зміни до конфігурації.

Розробники системи забезпечують технічну підтримку, оновлення програмного забезпечення та консультації щодо роботи системи.

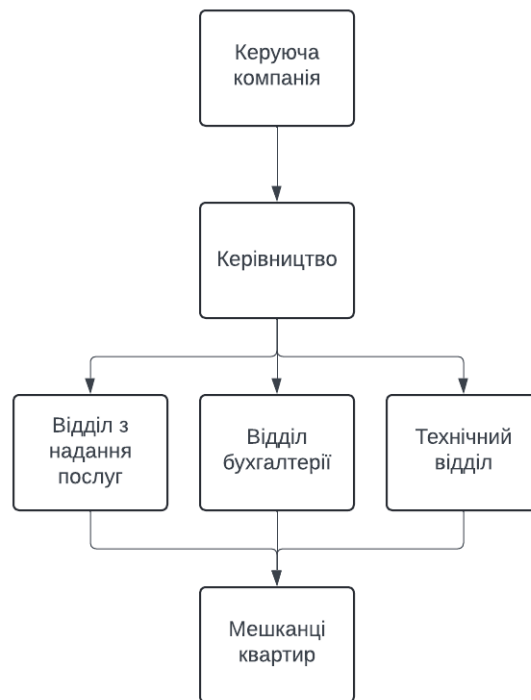


Рисунок 1.1 – Схема організаційної структури комплексу керуючої компанії

1.2.2 Топологія корпоративної мережі компанії у сфері надання послуг обслуговування комплексів

Огляд архітектури мережі:

- мережа відповідає топології на рисунку 1.2.
- передбачено можливість легкої адаптації мережі до зростання обсягів даних та кількості пристроїв;
- розраховано запровадження сучасних стандартів Ethernet та Wi-Fi для забезпечення швидкої та надійної передачі даних;
- передбачено інтеграцію передових технологій для підвищення ефективності управління мережевим трафіком у середовищі керуючої компанії.

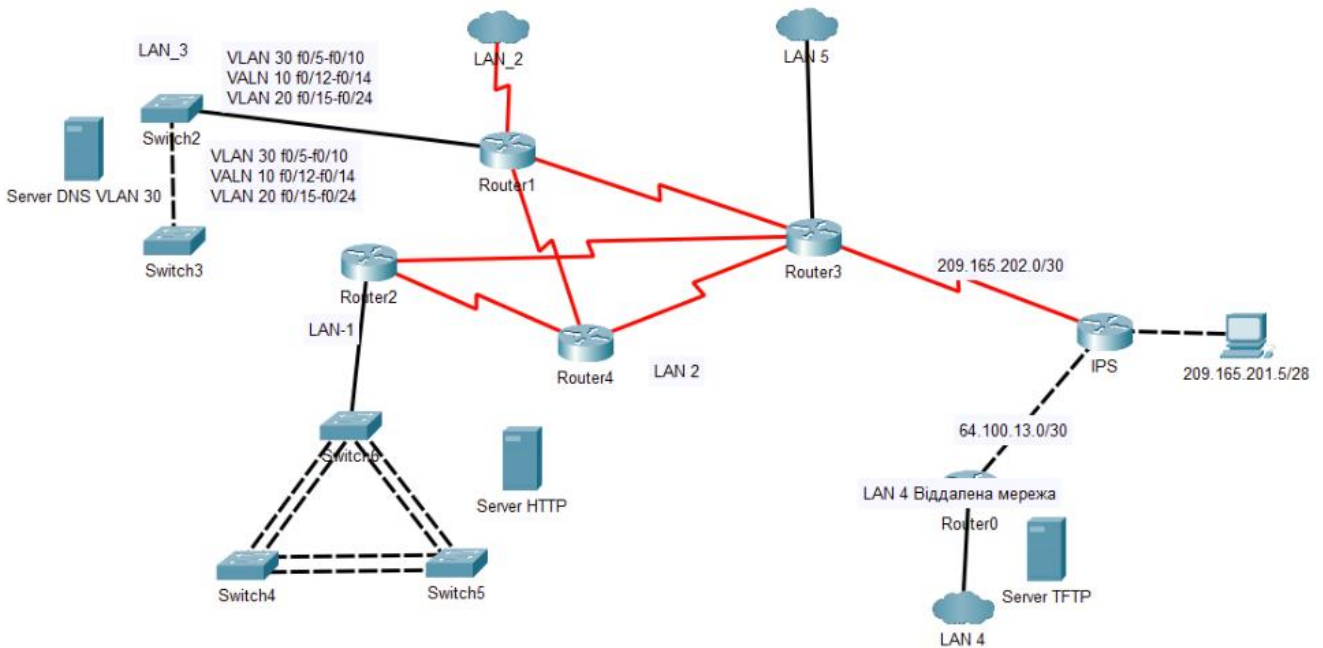


Рисунок 1.2 – Схема організаційної структури комплексу

1.2.3 Характеристика типових житлових приміщень

На рисунках 1.2 та 1.3 зображено типові квартири різної площі, що мають окремі входи з загального коридору. Кожна квартира має власний мікроклімат, який може відрізнятися від інших квартир в залежності від розташування, кількості вікон, зовнішніх стін, поверху та індивідуальних потреб мешканців.

Для ефективного контролю та управління мікрокліматом в кожній квартирі встановлюється комплект датчиків (температури, вологості та CO₂), які передають дані на центральний контролер. Реле та блоки живлення можуть бути розміщені в окремому щитку, або в іншому зручному місці, наприклад, в коридорі біля входу в квартиру.

Датчики температури, вологості та CO₂ встановлюються в кожній квартирі або приміщенні, де необхідно контролювати мікроклімат. Реле та блоки живлення можуть бути розміщені в окремому щитку або іншому зручному місці. Типові житлові приміщення зображені на (рис. 1.3 – 1.4).

1.2 Принципи та технічні способи інформаційного забезпечення кіберфізичної системи та мережі

Інформаційне забезпечення кіберфізичної системи мікроклімату ґрунтується на комплексному підході до збору, обробки та використання даних з використанням різноманітних датчиків, які постійно контролюють ключові параметри мікроклімату. Отримана інформація передається до центрального контролера, де вона піддається ретельній обробці та аналізу.

Для забезпечення точності та надійності даних використовуються різноманітні математичні методи та алгоритми. Фільтрація допомагає усунути випадкові похибки вимірювань, а калібрування забезпечує відповідність показань датчиків реальним значенням.

Зібрані та оброблені дані зберігаються в базі даних, що забезпечує можливість їх подальшого аналізу та використання. Крім того, дані передаються на веб-інтерфейс, де вони відображаються у зручному для користувача вигляді, дозволяючи здійснювати моніторинг та контроль над мікрокліматом у режимі реального часу.

Комп'ютерна мережа відіграє ключову роль у забезпеченні інформаційного обміну між компонентами кіберфізичних систем. Для передачі даних між датчиками, мікроконтролером та сервером використовується бездротовий протокол Wi-Fi. Цей протокол забезпечує достатню швидкість та надійність передачі даних, а також дозволяє легко масштабувати систему шляхом додавання нових пристроїв.

1.3 Огляд існуючих кіберфізичних систем забезпечення мікроклімату

Сучасний ринок пропонує різноманітні рішення реалізації розумних будинків, що базуються на кіберфізичних системах. Серед них можна виділити пропрієтарні системи від великих виробників, відкриті платформи та DIY-рішення, хмарні сервіси та рішення на базі мікроконтролерів. Пропрієтарні системи пропонують комплексні рішення, що включають датчики, термостати, контролери та програмне забезпечення. Ці системи відрізняються високою функціональністю,

але мають високу вартість та часто вимагають професійної установки та налаштування, тоді як відкриті платформи та DIY-рішення, такі як OpenHAB та Home Assistant дають більшу гнучкість, але вимагають певних технічних знань. Хмарні сервіси забезпечують віддалений доступ та зручний інтерфейс, але можуть мати обмежену функціональність, а рішення на базі мікроконтролерів, таких як Arduino, raspberry, ESP відрізняються доступністю та можливістю індивідуальної адаптації.

Кіберфізична система забезпечення мікроклімату будується на ключових принципах, що забезпечують її ефективність та адаптивність. Модульність дозволяє легко розширювати систему, додаючи нові датчики або пристрої, а також модифікувати її під конкретні потреби. Стандартизація компонентів гарантує їх взаємозамінність та сумісність, що спрощує обслуговування та модернізацію системи.

При побудові корпоративної мережі для управління кіберфізичною системою, важливими є принципи гнучкості, що дозволяють мережі легко адаптуватися до змін в кількості користувачів та обсягах даних. Резервування та відновлення забезпечують високу доступність системи та мінімізують ризики простоїв. Безпека є одним з пріоритетів, тому важливо впровадження сучасних заходів захисту, таких як файєрволи, системи виявлення вторгнень та шифрування даних.

На ринку існують перевірені рішення від провідних виробників, таких як Cisco Systems, Juniper Networks та HP Enterprise, які пропонують широкий спектр мережевого обладнання та програмного забезпечення для побудови корпоративних мереж. Ці рішення відрізняються високою продуктивністю, надійністю та безпекою, що робить їх придатними для використання в системах управління мікрокліматом.

1.4 Завдання і мета роботи

Метою роботи є розробка кіберфізичної системи для забезпечення мікроклімату розумного будинку на базі мікроконтролера ESP8266 та створення веб-інтерфейсу для моніторингу та керування системою.

Для вирішення поставленої задачі необхідно виконати наступні етапи:

- провести аналіз об'єкта впровадження;
- сформулювати технічні вимоги до мережі та кіберфізичної системи моніторингу та контролю мікроклімату;
- змодельовати комп'ютерну мережу комплексу;
- розробити проектні рішення для реалізації кіберфізичної системи та обрати обладнання;
- розробити програмні рішення для збору обробки та візуалізації інформації.

Результатом роботи повинна бути комп'ютерна мережа комплексу та функціональна кіберфізична система забезпечення мікроклімату, яка відповідає поставленим вимогам щодо точності, надійності та зручності використання.

1.5 Визначення можливих напрямків рішення поставлених завдань

1.5.1 Розробка архітектури мережі

Проектування гнучкої та масштабованої мережевої архітектури дозволяє забезпечити ефективну обробку зростаючого обсягу даних, що генеруються датчиками, та підключення нових пристроїв у майбутньому. Використання сучасних стандартів Ethernet та Wi-Fi гарантує високу швидкість та надійність передачі даних, що є критично важливим для оперативного реагування системи на зміни мікроклімату.

Для забезпечення безпечного доступу до системи з віддалених пристроїв впроваджуються віртуальні приватні мережі, що дозволяють створити захищений канал зв'язку між користувачем та сервером. Крім того, інтеграція технологій оптимізації маршрутизації трафіку дозволяє підвищити ефективність роботи

мережі та забезпечити пріоритетність передачі даних, критичних для роботи системи.

Вибір та інтеграція обладнання є важливим етапом розробки. Оптимальне поєднання мережевих комутаторів та маршрутизаторів, що підтримують необхідну пропускну здатність та забезпечують високий рівень безпеки, є запорукою стабільної роботи системи. Розгляд опцій мережевого зберігання даних дозволяє забезпечити ефективне архівування та швидкий доступ до накопиченої інформації, що може бути використана для аналізу та оптимізації роботи системи. Впровадження резервних систем, таких як джерела безперебійного живлення та резервні канали зв'язку, забезпечує високу доступність мережевих ресурсів та мінімізує ризики простоїв.

1.5.2 Розробка кіберфізичної системи

Існує декілька підходів до вирішення завдання створення системи керування мікрокліматом розумного будинку:

- використання готових рішень. На ринку представлені різноманітні готові системи від провідних виробників (Nest, Ecobee), вони пропонують широкий функціонал, але мають високу вартість та обмежені можливості індивідуальної адаптації;

- розробка системи на базі промислових контролерів. Такий підхід забезпечує високу надійність та продуктивність, але вимагає значних фінансових вкладень та кваліфікованого персоналу для налаштування та обслуговування;

- створення власної системи на базі мікроконтролерів. Цей підхід є найбільш гнучким та економічно вигідним, він дозволяє адаптувати систему під конкретні потреби та реалізувати індивідуальні алгоритми управління.

1.6 Обґрунтування вибраного напрямку інженерного рішення

Вибір оптимального інженерного рішення для розробки та впровадження системи базується на аналізі вимог до функціональності, надійності та ефективності, а також врахуванні специфіки житлових приміщень та потреб їх мешканців.

Пріоритетними вимогами до системи є висока продуктивність, стабільність роботи, безпека та можливість масштабування. Система повинна забезпечувати безперебійний контроль та управління мікрокліматом у квартирах та будинках, що вимагає постійного моніторингу та регулювання параметрів навколишнього середовища, таких як температура, вологість та рівень CO₂.

Виходячи з цього обрано розробку власної системи на базі мікроконтролера ESP8266. Цей вибір обумовлений наступними факторами:

- ESP8266 є одним з найбільш доступних мікроконтролерів з вбудованим модулем Wi-Fi. Це дозволяє створити систему з мінімальними витратами та великою гнучкістю [5];
- мікроконтролер має достатню обчислювальну потужність та обсяг пам'яті для реалізації необхідних алгоритмів управління мікрокліматом;
- завдяки вбудованому Wi-Fi модулю, мікроконтролер легко інтегрується з іншими пристроями розумного будинку та може бути частиною більш складної системи регулювання;
- розробка власної системи дозволяє врахувати всі особливості конкретного приміщення та реалізувати індивідуальні алгоритми управління мікрокліматом.

2 РОЗРОБКА АПАРАТНОЇ ЧАСТИНИ КІБЕРФІЗИЧНОЇ СИСТЕМИ

2.1 Технічні вимоги до кіберфізичної системи

2.1.1 Вимоги до системи в цілому

2.1.1.1 Вимоги до структури і функціонуванню системи

2.1.1.1.1 Перелік підсистем, їхнє призначення й основні характеристики, вимоги до числа рівнів ієрархії та ступені централізації системи

Система має бути складовою частиною комп'ютерної мережі керуючої компанії та надавати доступ до систем з верхніх рівнів мереж.

Кіберфізична система забезпечення мікроклімату розумного будинку має складатись з наступних підсистем:

– Підсистема збору даних, призначення якої має полягати у зборі даних про температуру та вологість повітря з датчиків. Для цього використовується декілька датчиків, які періодично опитуються з заданим інтервалом. Отримані дані передаються до центрального блоку керування через Wi-Fi.

– Підсистема керування яка відповідає за обробку даних з датчиків, прийняття рішень щодо керування виконавчими механізмами та взаємодію з веб-інтерфейсом. Система має бути реалізована на базі мікроконтролера ESP8266, який виконує алгоритми керування, має зберігати налаштування системи та історію даних, а також забезпечує віддалений доступ до системи через веб-інтерфейс.

– Підсистема виконавчих механізмів яка керуватиме кліматичними пристроями (опалення, кондиціонування, вентиляція) відповідно до команд центрального блоку керування. Для цього буде використовуватись реле, які вмикають/вимикають пристрої. Підсистема повинна мати можливість підключення різних типів кліматичних пристроїв.

– Веб-інтерфейс який забезпечуватиме зручний моніторинг та керування системою для користувача. Він повинен бути реалізований на базі Python та дозволить відображати поточну інформацію про мікроклімат та стан системи, змінювати налаштування системи та керувати виконавчими механізмами вручну.

Вимоги до числа рівнів ієрархії та ступені централізації:

Система повинна мати дворівневу ієрархію та складатись з нижнього рівня (блоку збору даних, блоку керування реле) та верхнього рівня, (центрального блоку керування та веб-інтерфейсу).

Ступінь централізації системи є високим, оскільки всі рішення щодо керування мікрокліматом приймаються центральним блоком керування на основі даних з датчиків та налаштувань користувача.

2.1.1.1.2 Вимоги до способів і засобів зв'язку між компонентами кіберфізичної системи

У локальній мережі LAN_1 буде застосовано технологію EtherChannel, яка об'єднує кілька фізичних з'єднань між комутаторами в одне логічне, підвищуючи пропускну здатність та забезпечуючи резервування каналів зв'язку.

У локальній мережі LAN_3 буде використано технологію VLAN для логічного розділення фізичної мережі на окремі віртуальні сегменти. Це дозволить ізолювати трафік різних підсистем, підвищити безпеку та ефективність роботи мережі.

Кіберфізична система повинна входити в склад віддаленої мережі LAN_4 та для забезпечення безпечної взаємодії віддаленої мережі LAN_4 з основною мережею буде застосовано технологію VPN.

У кіберфізичній системі зв'язок між блоком збору даних та центральним блоком керування повинен здійснюватись за допомогою бездротового зв'язку Wi-Fi з використанням протоколу MQTT. А інформація з датчиків кіберфізичної системи до керуючого мікроконтролера має передаватись за допомогою електричних сигналів номіналом 3.3 В.

Для визначення та прокладання оптимальних маршрутів між різними підмережами буде використано протокол динамічної маршрутизації OSPF. Цей протокол дозволяє автоматично адаптувати маршрути до змін у топології мережі, забезпечуючи високу надійність та ефективність передачі даних.

Підключення мережі до Інтернету здійснюватиметься через постачальника послуг з використанням технології NAT. NAT дозволяє приховати внутрішні IP-адреси пристроїв від зовнішньої мережі, підвищуючи безпеку системи.

2.1.1.1.3 Вимоги до характеристик взаємозв'язків створюваної системи із суміжними системами

Обмін інформацією в межах локальної мережі повинен здійснюватись за допомогою технології Ethernet та протоколу OSPF, що забезпечуватиме ефективну маршрутизацію та доставку даних між вузлами.

Для взаємодії між компонентами кіберфізичної системи, такими як датчики, виконавчі механізми та центральний контролер, повинен використовуватись спеціально налаштований IoT сервер та бездротові канали передачі даних Wi-Fi. Такий підхід дозволяє забезпечити гнучкість та масштабованість системи, а також спростити її розгортання та обслуговування.

2.1.1.1.4 Вимоги до режимів функціонування системи

Комп'ютерна мережа керуючої компанії, повинна підтримувати різні режими функціонування, що забезпечують її надійність, стабільність та безперебійну роботу:

– нормальний режим роботи, у якому мережа забезпечує повноцінний обмін даними між усіма пристроями, включаючи маршрутизатори, комутатори та сервер. Всі сервіси та протоколи працюють у штатному режимі, забезпечуючи доступ до веб-інтерфейсу та обмін даними з датчиками мікроклімату;

– режим обмеженої функціональності, коли у разі виникнення проблем зі зв'язком або відмови окремих компонентів мережі, система повинна автоматично перейти в режим обмеженої функціональності. У цьому режимі можуть бути обмежені деякі некритичні сервіси, такі як віддалений доступ, але основні функції маршрутизації та комутації повинні залишатися доступними;

– режим діагностики, який використовується у разі виникнення нестандартних ситуацій або проблем з роботою мережі, повинен бути передбачений режим діагностики. У цьому режимі адміністратор мережі може отримати доступ до розширеної інформації про стан мережі та її компонентів, що допоможе виявити та усунути проблему.

Кіберфізична система забезпечення мікроклімату повинна підтримувати наступні режими функціонування:

Автоматичний режим:

- система самостійно підтримує задані параметри температури та вологості, враховуючи дані з датчиків та зовнішні умови;
- користувач може налаштувати бажані параметри мікроклімату;
- система автоматично вмикає та вимикає кліматичні пристрої для підтримки заданих параметрів.

Ручний режим:

- користувач може вручну керувати кліматичними пристроями через веб-інтерфейс;
- система відображає поточний стан пристроїв та дозволяє користувачу вмикати/вимикати їх, а також змінювати режим роботи;

Аварійний режим:

- система автоматично переходить в цей режим при виникненні аварійних ситуацій (наприклад, несправність датчиків або виконавчих механізмів);
- в аварійному режимі система повинна вимикати всі кліматичні пристрої;
- користувач отримує повідомлення про аварійну ситуацію через веб-інтерфейс;

2.1.1.1.5 Вимоги до діагностування системи

Для забезпечення безперебійної роботи та швидкого виявлення потенційних проблем, комп'ютерна мережа повинна мати вбудовані механізми діагностування.

Це включає:

- аналіз мережевого трафіку завдяки збору та аналізу статистичних даних про мережевий трафік, який дозволяє виявляти перевантаження каналів зв'язку, підозрілу активність та інші потенційні проблеми.

- Логування подій через ведення детального журналу подій, пов'язаних з роботою мережі, включаючи зміни конфігурації, помилки, спроби несанкціонованого доступу тощо.

- Віддалений доступ для діагностики забезпечивши можливість віддаленого доступу до мережевого обладнання для проведення діагностики та налаштування. Це дозволяє адміністратору мережі оперативно реагувати на проблеми, навіть якщо він знаходиться поза межами офісу.

- Використання спеціалізованих інструментів та утиліт для діагностики мережі, таких як ping, traceroute, SNMP тощо. Ці інструменти дозволяють перевіряти доступність пристроїв, вимірювати затримки передачі даних, отримувати інформацію про стан мережевих інтерфейсів та інші параметри.

Система забезпечення мікроклімату повинна мати вбудовані засоби діагностування, які дозволять виявляти та локалізувати несправності, а також забезпечувати інформацію про поточний стан системи. Вимоги до діагностування включають:

Моніторинг стану компонентів:

Система повинна постійно відстежувати стан датчиків, виконавчих механізмів, блоку живлення та інших компонентів. При виявленні несправності система повинна генерувати повідомлення про помилку та інформувати користувача через веб-інтерфейс або інші канали зв'язку.

Самодіагностика:

Система повинна мати можливість проведення самодіагностики для виявлення прихованих несправностей Самодіагностика може включати перевірку працездатності датчиків, калібрування, тестування зв'язку між компонентами.

Логування подій:

Система повинна вести журнал подій, в якому фіксуються всі зміни стану системи, помилки, дії користувача. Журнал подій може бути використаний для аналізу роботи системи та виявлення причин несправностей.

Віддалений доступ для діагностики:

Система повинна забезпечувати можливість віддаленої діагностики через веб-інтерфейс або інші засоби зв'язку. Це дозволить фахівцям з технічної підтримки швидко виявляти та усувати несправності, не вимагаючи фізичної присутності на об'єкті.

Індикація стану:

Система повинна мати світлові або звукові індикатори, які відображають поточний стан системи та наявність помилок. Це дозволить користувачам швидко оцінити стан системи та вжити необхідних заходів у разі виникнення проблем.

2.1.1.1.6 Перспективи розвитку, модернізації системи

Перспективи розвитку та модернізації комп'ютерної мережі:

Інфраструктура системи повинна бути спроектована з урахуванням майбутнього зростання компанії, використовуючи мережеві архітектури, віртуалізацію та хмарні сервіси для забезпечення гнучкого масштабування ресурсів. При цьому, вона має адаптуватися до новітніх загроз, впроваджуючи передові технології захисту інформації, обліку та протоколів передачі даних. Система повинна бути відкритою до інтеграції з сучасними технологіями, такими як Інтернет речей. Архітектура системи має бути спроектована таким чином, щоб легко адаптуватися до нових вимог бізнесу та інтегрувати новітні технології.

Перспективи розвитку та модернізації кіберфізичної системи:

Розширення кількості та типів датчиків за допомогою включення датчиків освітленості, забрудненості повітря, присутності людей, відкриття вікон/дверей, для більш точного та адаптивного керування мікрокліматом.

Вдосконалення алгоритмів керування:

- розробка самонавчальних алгоритмів, які враховують індивідуальні уподобання користувачів та адаптуються до змін умов навколишнього середовища;
- реалізація функцій аналізу даних для виявлення закономірностей та прогнозування;
- використання даних про прогноз погоди для більш точного керування мікрокліматом та економії енергії.

Підвищення енергоефективності:

- застосування реле з низьким енергоспоживанням, датчиків з режимом сну, оптимізація роботи мікроконтролера;
- збір та аналіз даних про споживання електроенергії кліматичними пристроями для оптимізації роботи системи.

Розширення функціональності веб-інтерфейсу:

- можливість перегляду детальної статистики по споживанню енергії, налаштування індивідуальних профілів користувачів, створення сценаріїв регулювання на основі різних подій (час доби, погода, дані з інших датчиків);
- підключення до метеослужб для отримання прогнозу погоди та врахування його при керуванні мікрокліматом, інтеграція з календарем користувача для автоматичної зміни налаштувань в залежності від його розкладу;
- адаптація веб-інтерфейсу для зручного використання на смартфонах та планшетах.

2.1.1.2 Вимоги до показників призначення

Система має забезпечити працівникам та користувачам зручний доступ до даних з кіберфізичних систем та можливість обміну інформацією між хостами. Впровадити централізоване зберігання даних на серверах з додатковим резервним копіюванням у хмарі гарантуватиме збереження та відновлення важливої інформації. Безпека даних повинна бути забезпечена завдяки заходам захисту від

несанкціонованого доступу та шкідливого програмного забезпечення. Система має надавати можливість підключення до Інтернету та віддаленої роботи через VPN.

2.1.1.3 Вимоги до патентної чистоти

У процесі розробки кіберфізичної системи забезпечення мікроклімату розумного будинку не передбачається використання жодних запатентованих винаходів або технологій, що потребують ліцензування, система повинна базуватись на стандартних виробках. Всі компоненти системи (мікроконтролер ESP8266, датчики BME280, АНТ10, МН-Z19В, реле, блок живлення) є загальнодоступними та не є патентними винаходами.

Розроблене програмне забезпечення для веб-інтерфейсу має бути оригінальним. Воно не повинно містити комерційного коду або алгоритмів, що можуть порушувати права інтелектуальної власності третіх осіб.

2.1.1.4 Додаткові вимоги

Додаткові вимоги до комп'ютерної мережі не представляються.

Система забезпечення мікроклімату повинна відповідати наступним додатковим вимогам, пов'язаним з особливостями її експлуатації. Вимоги до умов експлуатації: Блок який стоїть всередині повинен бути стійким до впливу пилу, вологи та перепадів температур, характерних для житлових та офісних приміщень. Робочий діапазон температур повинен становити від 0°C до +40°C, а відносної вологості – від 20% до 80%.

Зовнішній блок повинен мати захист від прямих сонячних променів, попадання вологи та дощу. Робочий діапазон температур повинен становити від -20°C до +50°C, а відносної вологості – від 20% до 90%.

Вимоги до активного обладнання:

– активне обладнання (мікроконтролер, реле) повинно забезпечувати стабільну роботу протягом тривалого часу без необхідності перезавантаження або обслуговування;

- мікроконтролер повинен мати достатню кількість портів вводу/виводу для підключення всіх датчиків та виконавчих механізмів, а також запас портів для можливого розширення системи в майбутньому;

- система повинна мати можливість встановлення як на плоску горизонтальну поверхню, так і на стіну;

- активне обладнання повинно відповідати вимогам електромагнітної сумісності та не створювати перешкод для роботи інших пристроїв, особливо на частоті Wi-Fi стандарту 802.11n 2.4GHz.

Вимоги до комунікаційного обладнання та його розташування:

- комунікаційне обладнання (мікроконтролер, сервер) повинно бути розміщене в місці з надійним покриттям Wi-Fi сигналом;

- використання стандартних кабельних трас та конекторів;

- обладнання повинно бути розміщене в шафі з урахуванням вимог до охолодження та зручності доступу.

2.1.2 Вимоги функцій, виконуваним системою

2.1.2.1 Підсистема збору даних

Перелік функцій:

- збір даних з датчиків температури та вологості;

- фільтрація та усереднення даних для зменшення похибки вимірювання;

- формування пакетів даних для передачі;

- передача даних до центрального блоку керування через Wi-Fi.

Часовий регламент та вимоги до якості:

- частота опитування датчиків: не рідше ніж раз на хвилину;

- точність вимірювання: відповідно до вимог до системи в цілому ($\pm 0.5^{\circ}\text{C}$ для температури, $\pm 3\%$ для вологості);

- час передачі даних: не більше 1 секунди;

- формат даних: JSON або інший зручний для обробки формат.

2.1.2.2 Підсистема керування

Перелік функцій:

- прийом та обробка даних з блоку збору даних;
- аналіз отриманих даних та порівняння їх з заданими параметрами мікроклімату;
- прийняття рішень щодо керування виконавчими механізмами на основі алгоритмів керування;
- формування та передача команд керування на блок керування реле;
- прийом та обробка команд від веб-інтерфейсу;
- Зберігання налаштувань системи та історії даних;
- Забезпечення віддаленого доступу до системи через веб-інтерфейс.

Часовий регламент та вимоги до якості:

- час реакції на зміну даних з датчиків: не більше 5 секунд;
- час виконання алгоритмів керування: не більше 1 секунди;
- час передачі команд на блок керування реле: не більше 1 секунди;
- час відгуку веб-інтерфейсу: не більше 2 секунд;
- обсяг зберігання історії даних: не менше 1 місяця.

2.1.2.3 Веб-інтерфейс

Перелік функцій:

- відображення поточної інформації про мікроклімат (температура, вологість тиск,);
- відображення стану кліматичних пристроїв (ввімкнено/вимкнено, режим роботи);
- можливість зміни налаштувань системи (задані параметри мікроклімату, розклад роботи);
- можливість ручного керування кліматичними пристроями;
- відображення історії даних про мікроклімат;

- забезпечення авторизації та захисту від несанкціонованого доступу.

Часовий регламент та вимоги до якості:

- час оновлення даних на сторінці: не більше 5 секунд;
- час відгуку на дії користувача: не більше 2 секунд;
- інтуїтивно зрозумілий та зручний інтерфейс.

2.1.3 Вимоги до видів забезпечення системи

2.1.3.1 Вимоги до математичного забезпечення

Вимоги до кіберфізичної системи.

- мають бути розроблені алгоритми фільтрації та усереднення даних з датчиків, алгоритми визначення поточного стану мікроклімату;
- мають бути розроблені алгоритми прийняття рішень щодо керування кліматичними пристроями на основі заданих параметрів та поточних даних, алгоритми регулювання потужності пристроїв;

2.1.3.2 Вимоги до інформаційного забезпечення

Вимоги до комп'ютерної мережі керуючої компанії:

Наявність чіткої та логічної структури даних, що охоплює всі необхідні елементи для ефективної роботи. Це передбачає використання уніфікованих форматів та стандартів (ISO) для забезпечення узгодженості та цілісності інформації, а також полегшення майбутньої масштабованості та інтеграції.

Обмін інформацією між різними частинами системи має бути організований так, щоб забезпечити швидкий та надійний доступ до даних, навіть за умов високого навантаження. Це вимагає впровадження ефективних протоколів передачі даних (TCP/IP, HTTP) та механізмів синхронізації (черги повідомлень, розподілені транзакції) між різними модулями системи, а також врахування можливих сценаріїв відмов та відновлення після них.

Підтримувати сумісність з іншими системами, що використовуються в компанії, а також з потенційними зовнішніми партнерами чи сервісами. Це означає

підтримку стандартів обміну даними, інтеграцію з існуючим програмним та апаратним забезпеченням, а також можливість безперешкодного та безпечного обміну інформацією з зовнішніми системами через різні канали зв'язку.

Інформаційне забезпечення кіберфізичної системи забезпечення мікроклімату повинно відповідати наступним вимогам.

Склад, структура та організація даних:

- дані про температуру та вологість повинні зберігатись в базі даних з мітками часу для подальшого аналізу та візуалізації;
- налаштування системи (задані параметри мікроклімату, розклад роботи) повинні зберігатись в окремому файлі конфігурації або в базі даних;
- історія подій (зміни стану системи, помилки, дії користувача) повинна зберігатись в журналі подій.

Інформаційний обмін між компонентами:

- для обміну даними між блоком збору даних та центральним блоком керування має бути використано протокол MQTT або HTTP;
- для взаємодії з веб-інтерфейсом використовуються протоколи HTTP та WebSocket.

Інформаційна сумісність із суміжними системами:

- система повинна підтримувати обмін даними з іншими системами розумного будинку за допомогою відкритих протоколів та інтерфейсів (MQTT);
- для забезпечення сумісності з різними системами можуть використовуватися шлюзи або конвертери протоколів.

2.1.3.3 Вимоги до лінгвістичного забезпечення

Лінгвістичне забезпечення кіберфізичної системи забезпечення мікроклімату повинно відповідати наступним вимогам:

Мови програмування:

- для програмування мікроконтролера ESP8266 повинна використовуватись мова MicroPython [9];

- для створення веб-інтерфейсу має бути використана мова Python з використанням фреймворку Flask [6].

Мови взаємодії користувачів і технічних засобів:

- веб-інтерфейс повинен бути реалізований з використанням HTML, CSS та JavaScript для забезпечення інтерактивності та зручності використання.

Кодування та декодування даних:

- для передачі даних між компонентами системи повинні використовуватись формати JSON або інші зручні для обробки формати;

- для забезпечення безпеки повинно використовуватися шифрування паролю користувача.

Засоби опису предметної області:

- для опису об'єкта регулювання (мікроклімату приміщення) повинні використовуватись фізичні величини (температура, вологість) та їх одиниці вимірювання.

Способи організації діалогу:

- веб-інтерфейс повинен забезпечувати інтуїтивно зрозумілий та зручний діалог з користувачем.

2.1.3.4 Вимоги до технічного забезпечення

Технічне забезпечення кіберфізичної системи забезпечення мікроклімату включає в себе різноманітні компоненти. Основою системи має бути мікроконтролер ESP8266 на платі Wemos D1 Mini або аналогічний, який повинен мати підтримку Wi-Fi, достатній обсяг флеш-пам'яті та оперативної пам'яті, а також підтримувати інтерфейси UART, I2C, SPI та GPIO. Для збору даних про мікроклімат мають бути використані сумісні з мікроконтролером датчики:

- датчик температури, вологості та атмосферного тиску, серії BМх або аналогічний;

- датчик температури та вологості серії АНТ або аналогічний;

- датчик концентрації вуглекислого газу серії МН або аналогічний.

Для забезпечення бездротового зв'язку між компонентами системи використовувати Wi-Fi роутер з підтримкою стандарту 802.11 b/g/n. Веб-інтерфейс та база даних можуть бути розміщені на окремому сервері або комп'ютері з операційною системою Linux або Windows.

2.1.3.5 Вимоги до організаційного забезпечення

Організаційне забезпечення кіберфізичної системи забезпечення мікроклімату передбачає чіткий розподіл ролей та відповідальності між учасниками проекту. Розробники відповідають за проектування, розробку, тестування, впровадження та налаштування системи на об'єкті надання консультації користувачам, проведення діагностики та усунення несправностей. Користувачі відповідають за експлуатацію системи, моніторинг її роботи та налаштування параметрів мікроклімату.

Організація функціонування системи починається з її встановлення та первинного налаштування на об'єкті технічними спеціалістами. Після цього проводиться навчання користувачів основам роботи з системою та веб-інтерфейсом, яке можуть здійснювати як розробники, так і технічна підтримка. Далі користувачі самостійно моніторять роботу системи через веб-інтерфейс, змінюють налаштування та керують кліматичними пристроями.

Веб-інтерфейс має бути розроблений таким чином, щоб бути максимально простим та зрозумілим для користувачів, що мінімізує ризик помилок. Система також повинна мати можливість налаштування прав доступу для різних користувачів, що запобігає несанкціонованим змінам налаштувань. Перед виконанням критичних дій система запитує підтвердження у користувача, а ведення журналу подій дозволяє відстежувати дії користувачів та виявляти можливі помилки. Крім того, система має можливість відновлення налаштувань за замовчуванням у разі неправильного налаштування.

2.1.3.6 Вимоги до методичного забезпечення

Для забезпечення ефективного використання та обслуговування кіберфізичної системи забезпечення мікроклімату, необхідно розробити детальну методичну документацію.

1) Інструкція з монтажу та встановлення, документ, який повинен містити покрокові інструкції щодо монтажу та встановлення всіх компонентів системи, включаючи підключення датчиків, реле, блоку живлення та налаштування Wi-Fi з'єднання. Він також повинен включати рекомендації щодо вибору місця встановлення датчиків та інших компонентів для забезпечення оптимальної роботи системи.

2) Інструкція з експлуатації, повинна містити детальний опис роботи системи, її функцій та режимів роботи. Повинна включати інструкції щодо налаштування бажаних параметрів мікроклімату, створення розкладів роботи кліматичних пристроїв, а також щодо використання веб-інтерфейсу для моніторингу та керування системою.

3) Інструкція з роботи з веб-інтерфейсом повинна містити детальний опис всіх елементів веб-інтерфейсу, їх призначення та способи використання. Повинна включати інструкції щодо перегляду поточної інформації про мікроклімат, зміни налаштувань системи, ручного керування кліматичними пристроями та перегляду історії даних.

4) Інструкція з технічного обслуговування, яка повинна містити інструкції щодо періодичного обслуговування системи, включаючи перевірку працездатності датчиків, реле та інших компонентів, а також оновлення програмного забезпечення.

2.1.3.7 Вимоги до задач (налаштувань), які виконує система

Під час розробки адресації для комп'ютерної мережі керуючої компанії налаштувати:

– мережа повинна складатися з 5 підмереж LAN1-LAN5 (рис.1.2).

– Кількість вузлів в кожній підмережі складатиме 57, 120, 161, 70 та 23 відповідно.

– Адресація для виділення підмереж 172.24.80.0/21.

– Середня інтенсивність вихідного трафіку в найбільш завантаженій мережі має становити 141 кадр/с.

Для виконання базового налаштування конфігурації пристроїв необхідно дотримуватися таких вимог:

– Назви пристроям за таким правилом: Прізвище студента_тип пристрою_номер пристрою.

– Встановити пароль cisco для доступу до консолі та ліній vty на всіх пристроях.

– Захистити привілейований режим паролем class на всіх пристроях.

– Зашифрувати всі паролі, що зберігаються у відкритому вигляді.

– Розробити інформаційний банер MOTD.

– Налаштувати використання протоколу SSH на всіх лініях vty.

– Створити локальних користувачів з паролем "admincisco" на всіх пристроях.

– Використовувати ім'я пристрою як ім'я домену.

– Згенерувати ключ RSA довжиною 1024 біти для шифрування даних.

– На DCE-інтерфейсах маршрутизаторів встановити тактову частоту 128000.

– Налаштувати аудит початку та завершення процесів ехес з використанням локальної бази.

– З метою підвищення пропускної здатності та надійності каналів у мережі LAN_1, необхідно виконати агрегацію фізичних ліній на комутаторах.[1]

На маршрутизаторах необхідно налаштувати протокол OSPF що дозволить використовувати декілька шляхів одночасно для передачі даних, забезпечить швидке відновлення мережі після збоїв та оперативне оновлення маршрутів при змінах в топології, мінімізуючи при цьому обсяг службового трафіку.

Під час налаштування маршрутизаторів необхідно встановити маршрут за замовчуванням на маршрутизаторі, який має пряме підключення до інтернет-

провайдера, та поширити цей маршрут через оновлення протоколу маршрутизації. Під час розробки кіберфізичної системи повинно врахувати наступні вимоги:

IP-адресу для запуску веб-інтерфейсу кіберфізичної системи необхідно вказати IP-адресу сервера, яка становить 172.24.85.61.

Веб-інтерфейс може бути доступний через порти 50, 80 або 5000 (за замовчуванням). Переконайтеся, що обраний порт не використовується іншими службами.

Після запуску веб-інтерфейсу необхідно вказати IP-адресу MQTT-брокера (сервера повідомлень) у налаштуваннях. Це забезпечить коректну взаємодію між веб-інтерфейсом, мікроконтролером та датчиками, дозволяючи збирати, обробляти та візуалізувати дані в реальному часі.

2.2 Розробка апаратної частини комп'ютерної системи

2.2.1 Розробка загальної архітектури мережі комплексу

Для забезпечення зв'язку між маршрутизаторами використовуються спеціальні кабелі Serial DTE або крос-кабелі, що гарантують швидкий та надійний обмін інформацією між різними частинами мережі. Ці кабелі відіграють важливу роль у забезпеченні стабільної роботи мережі та ефективної маршрутизації даних між підмережами.

Маршрутизатори підключаються до комутаторів за допомогою прямих кабелів, що забезпечує високу пропускну здатність та мінімізує втрати даних. Таке пряме з'єднання є простим та надійним способом обміну інформацією між цими ключовими компонентами мережі.

Робочі станції (ПК) також підключаються до комутаторів за допомогою прямих кабелів, що забезпечує швидкий та стабільний доступ до мережевих ресурсів. Це особливо важливо для забезпечення безперебійної роботи користувачів та швидкого доступу до необхідних даних та сервісів.

Для з'єднання комутаторів між собою використовуються крос-кабелі, що дозволяють створити додаткові шляхи для передачі даних та забезпечити

резервування каналів зв'язку. Це підвищує надійність мережі та забезпечує її стійкість до можливих збоїв або перевантажень.

На основі загальної характеристики комплексу, його архітектури, кількості підмереж розроблено структурну схему комплексу технічних засобів комп'ютерної системи, яка відображена на рисунку 2.1. Ця схема відображає всі ключові компоненти мережі, їхні взаємозв'язки та способи підключення, що дозволяє зрозуміти структуру та принципи роботи мережі.

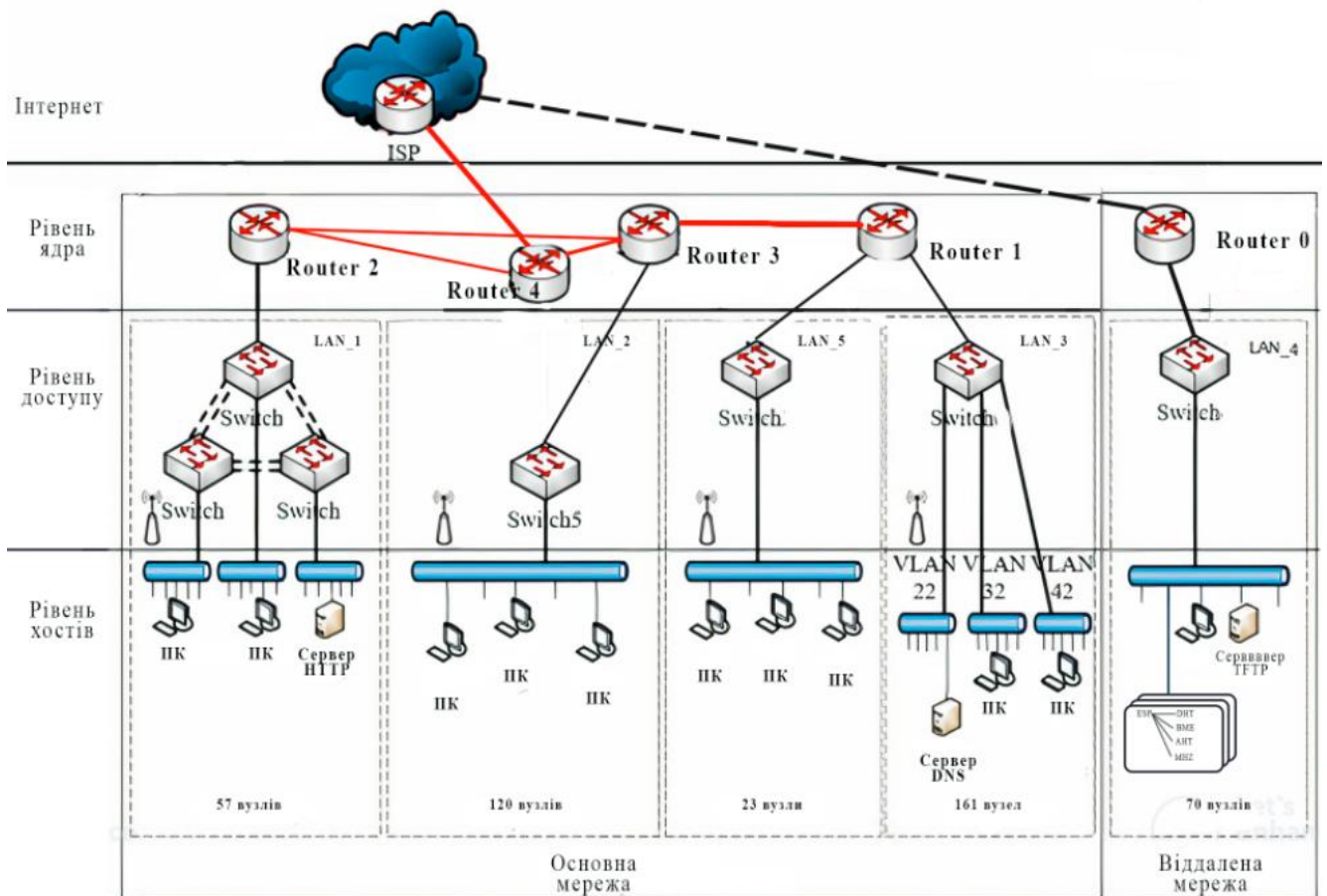


Рисунок 2.1 – Структурна схема комплексу технічних засобів комп'ютерної системи

Кіберфізична система забезпечення мікроклімату розумного будинку функціонує в рамках віддаленої локальної мережі, яка об'єднує всі її компоненти. Датчики та виконавчі механізми підключаються до мікроконтролера, який виконує роль центрального контролера. Мікроконтролер забезпечує збір даних з датчиків,

обробку цих даних, прийняття рішень щодо керування реле та передачу даних на сервер через Wi-Fi з'єднання. Сервер, реалізований на базі звичайного ПК під керуванням Windows або Linux, приймає дані від мікроконтролера, зберігає їх у базі даних та забезпечує роботу веб-інтерфейсу. Користувачі можуть отримати доступ до веб-інтерфейсу через будь-який пристрій з підтримкою Wi-Fi та браузера.

2.2.2 Вибір і обґрунтування структурної схеми комплексу технічних засобів системи

У якості маршрутизатора було обрано Cisco 2911. Ця модель має високу продуктивність та розширені можливості для обробки даних мережі. Технічні характеристики моделі наступні: має два WAN порти, дев'ять портів під підключення комутаторів та мережевий екран швидкістю до 5Gbps.

Для побудови локальної мережі обрано комутатор Cisco 2960-24TT. Цей комутатор, що входить до серії Cisco Catalyst, оснащений 24 портами Fast Ethernet та підтримує широкий спектр функцій, таких як VLAN, QoS, STP та ACLs. Завдяки цьому, комутатор забезпечує не лише швидку передачу даних, але й розширені можливості управління, безпеки та масштабованості. Це робить його ідеальним вибором для забезпечення надійного з'єднання робочих станцій, серверів та інших мережевих пристроїв у мережі керуючої компанії.

Для забезпечення доступу кіберфізичної системи до мережі керуючої компанії використовується маршрутизатор MikroTik RB952Ui-5ac2nD. Цей маршрутизатор спеціально розроблений для використання у вимогливих корпоративних середовищах та підтримує стандарт Wi-Fi 5, забезпечуючи високу швидкість бездротового зв'язку до 2000 Мбіт/с. Це дозволяє одночасно підключати та обслуговувати велику кількість користувачів та пристроїв без втрати якості зв'язку та швидкості.

MikroTik RB952Ui-5ac2nD також має вбудовані засоби безпеки, такі як можливість створення віртуальних приватних мереж (VPN), що дозволяє захистити передачу даних у мережі. Крім того, маршрутизатор має можливість підключення

додаткових мережеских пристроїв, що забезпечує гнучкість та масштабованість мережі.

Таблиця 2.1 – Специфікація обладнання

Позиція	Найменування	Тип, марка, позначення документа	Одиниці виміру	Кількість
1	Cisco 2911.	Holubitskiy_R1 Holubitskiy_R2 Holubitskiy_R3 Holubitskiy_R4 Holubitskiy_R0	Од.	5
2	Cisco 2960-24TT	Holubitskiy_SW1 Holubitskiy_SW2 Holubitskiy_SW3 Holubitskiy_SW4 Holubitskiy_SW5 Holubitskiy_SW6 Holubitskiy_SW7 Holubitskiy_SW8 Holubitskiy_SW9	Од.	9
3	MikroTik RB952Ui-5ac2nD	Holubitskiy_WiFi1 Holubitskiy_WiFi2 Holubitskiy_WiFi3	Од.	3

Кіберфізична система:

Структурна схема комплексу технічних засобів відображає взаємозв'язок між основними компонентами системи:

– датчики підключені до мікроконтролера ESP8266 за допомогою інтерфейсів I2C та UART;

– мікроконтролер ESP8266 отримує дані з датчиків, обробляє їх та передає на сервер через Wi-Fi. Також керує реле для регулювання роботи кліматичних пристроїв;

- реле підключені до мікроконтролера ESP8266 та використовуються для вмикання/вимикання кліматичних пристроїв;
- блок живлення забезпечує живлення компонентів системи;
- сервер отримує дані від мікроконтролера, зберігає їх у базі даних та надає доступ до веб-інтерфейсу;
- Wi-Fi роутер забезпечує бездротовий зв'язок між мікроконтролером та сервером;

2.2.3 Розробка специфікації апаратних засобів кіберфізичної системи

Відповідно до вимог в розділі 2.1.3.4 обрано такі датчики:

BME280, дозволяє одночасно вимірювати температуру, відносну вологість та атмосферний тиск. BME280 підтримує інтерфейси I2C та SPI, що робить його зручним для підключення до мікроконтролера ESP8266[2].

AHT10, спеціалізується на вимірюванні температури та вологості. Він відрізняється високою точністю вимірювання відносної вологості та стабільністю показань в широкому діапазоні температур. AHT10 також використовує інтерфейс I2C, що спрощує його підключення до системи[3].

MH-Z19B, призначений для вимірювання концентрації CO₂ в повітрі. Він використовує інфрачервоний (NDIR) метод вимірювання, який забезпечує високу точність. MH-Z19B підтримує інтерфейс UART та має вбудований температурний компенсатор, що дозволяє отримувати максимально точні показання незалежно від температури навколишнього середовища[4].

Керування кліматичними пристроями має здійснюватись за допомогою реле з нормально розімкнутими контактами, які здатні комутувати напругу 220 В та струм не менше 5 А з часом спрацьовування не більше 10 мс. Для стабільного живлення всіх компонентів системи використовувати стабілізований блок живлення з вихідною напругою 5 В, вихідним струмом не менше 1 А.

Таблиця 2.1 – Специфікація апаратних засобів

Компонент	Модель	Кількість	Призначення
Мікроконтролер	ESP8266 (Wemos D1 Mini)	2	Збір даних, управління реле, зв'язок з сервером
Датчик температури/вологості/тиску	BME280	1	Вимірювання температури, вологості та тиску
Датчик температури/вологості	AHT10	1	Вимірювання температури та вологості
Датчик CO2	MH-Z19B	1	Вимірювання концентрації CO2 в повітрі
Виконавчий пристрій (реле)	ESP01/01s relay	1	Перемикання живлення мережних пристроїв
Блок живлення	5 В, 1 А	2	Живлення компонентів системи
Сервер	ПК під керуванням Windows або Linux	1	Зберігання та обробка даних, забезпечення роботи веб-інтерфейсу

2.2.4 Розрахунок інтенсивності трафіку вихідного трафіку найбільшої локальної мережі керуючої компанії

Мережа об'єднує 161 ПК. Вихідний трафік з комутатора надсилається до роутера в лінію з пропускну здатністю, що становить 1000 Мбіт/с.

Щоб забезпечити стабільну роботу комутатора та уникнути його перевантаження, необхідно дотримуватися балансу між швидкістю надходження та відправлення пакетів.

З огляду на середню інтенсивність трафіку $\mu=141$ кадр/с та середню довжину повідомлення 1150 байт, можна розрахувати необхідну пропускну здатність

комутатора. Розглянемо сценарій коли всі користувачі найбільшої підмережі DLS одночасно активні в мережі.

В цьому випадку пропускна здатність буде дорівнювати:

$$P_{p.p.} = \mu * L_{пов} * N * 8 = 141 * 1150 * 161 * 8 = 20.8 \text{ Мбіт/с} \quad (2.1)$$

де $L_{пов}$ – середня довжина повідомлення, N – кількість вузлів в мережі.

Оскільки отриманий результат не перевищує заданих параметрів мережі по вихідному каналу, можна зробити висновок, що перенавантаження комутатора не відбудеться. Комутатор також передає трафік до маршрутизатора зі швидкістю 1000 Мбіт/с. Отже, загальне навантаження на комутатор не повинно перевищувати:

$$\mu_{вих} = 10^9 / (1150 * 8) = 108\,696 \text{ пакетів/с} \quad (2.2)$$

Оскільки в середньому, кожне джерело виробляє 86 пакетів/с, то маршрутизатор обмежений кількістю приєднань, яку ми можемо дізнатись наступним чином:

$$N = \mu_{вих} / \mu = 108\,696 / 141 \approx 770 \text{ джерел} \quad (2.3)$$

Ця кількість задовольняє кількості вузлів у найбільшій локальній мережі, до якої входить 161 ПК.

Кожен з 161 ПК посилає потік заявок з інтенсивністю у 141 кадрів/с. Звідси, можна розрахувати інтенсивність вихідного трафіку:

$$\lambda = N * \mu = 161 * 141 = 22701 \text{ пакетів/с.} \quad (2.4)$$

Коефіцієнт затримки:

$$\rho = \frac{\lambda}{\mu_{вих}} = \frac{22701}{108696} = 0,208 \quad (2.5)$$

Коефіцієнт зайнятості маршрутизатора:

$$\frac{\rho}{1-\rho} = \frac{0,208}{1-0,208} = 0,262 \quad (2.6)$$

Середня затримка кадру дорівнює:

$$T = \frac{1}{(\mu-\lambda)} = \frac{1}{(108696 - 22701)} = 11,62 \text{ мкс} \quad (2.7)$$

Середня довжина черги:

$$L_{чер} = \frac{\rho^2}{1-\rho} = \frac{0,208^2}{1-0,208} = 0.0546 \quad (2.8)$$

Середній час перебування пакета у черзі:

$$T_{оч} = \frac{L_{чер}}{\lambda} = \frac{0,0546}{22701} = 0,24 \text{ мкс} \quad (2.9)$$

Це значення задовольняє вимогам до затримки в ЛМ.

3 РОЗРОБКА КОРПОРАТИВНОЇ МЕРЕЖІ

3.1 Проектування логічної топології мережі

Відповідно до організаційної структури керуючої компанії та топології мережі, що описані в першому розділі, в Packet Tracer побудовано модель мережі (рис.3.1).

Топологія організована з використанням комутаторів та маршрутизаторів, що забезпечують з'єднання між різними відділами та мережевими сегментами. Ключовим елементом є інтеграція VLAN, яка дозволяє ефективно розділити мережеві ресурси на основі функціональних потреб без зміни фізичної структури мережі. Це забезпечує високий рівень безпеки та оптимізацію трафіку, що особливо важливо для корпоративних мереж великих масштабів.

На рисунку 3.1 зображена логічна схема корпоративної мережі, що включає основну, віддалену та мережу ISP провайдера. Основна мережа є центральним елементом, віддалена забезпечує підключення віддалених офісів та співробітників, а мережа ISP провайдера надає доступ до Інтернету. Для з'єднання використовуються кабелі Serial Ethernet та Gigabit Ethernet[1].

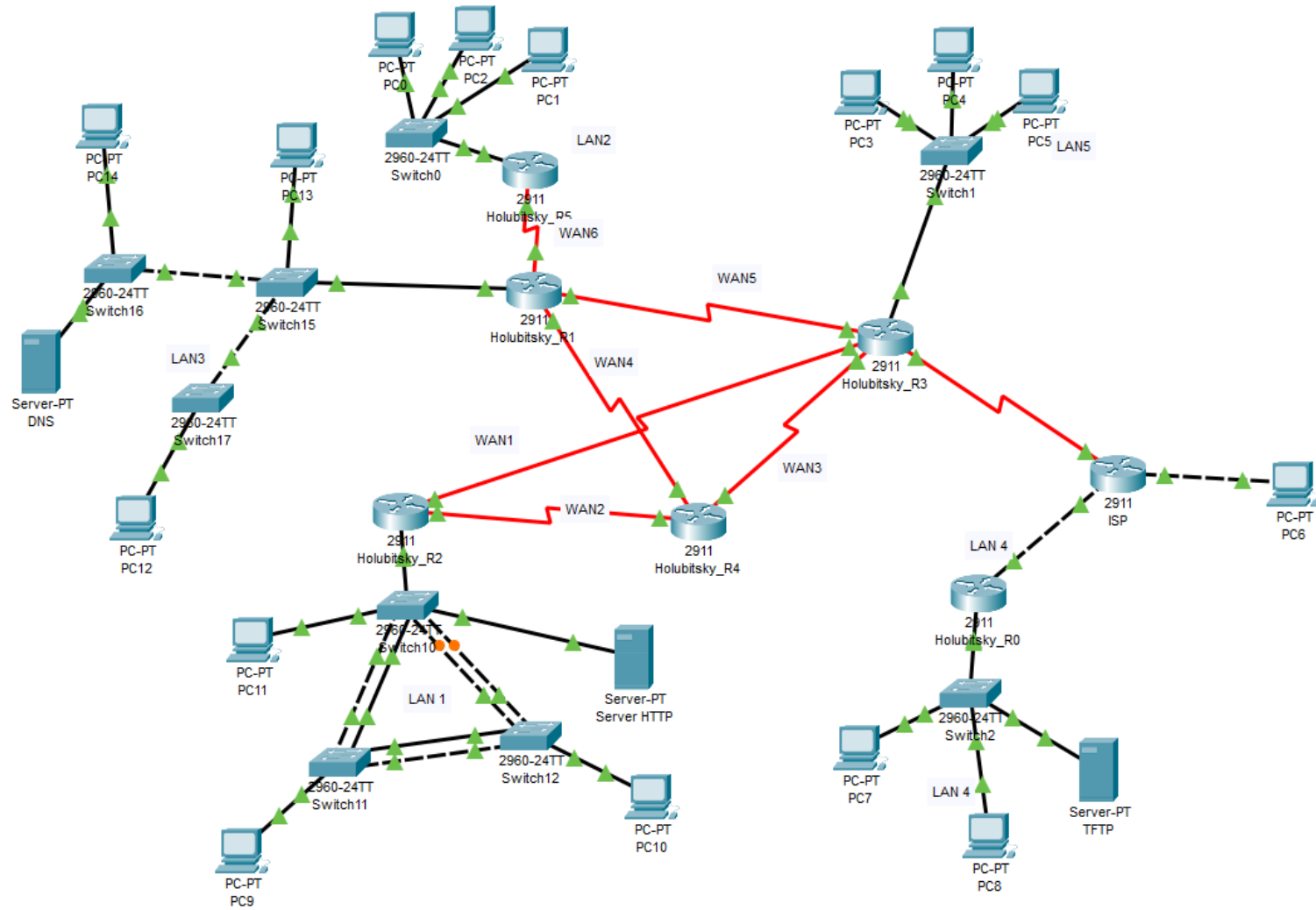


Рисунок 3.1 – Топологічна схема мережі комплексу

3.2 Вибір та опис мережного обладнання

Для побудови надійної та продуктивної мережі комплексу обрано комутатори та маршрутизатори від Cisco, відомого виробника мережевого обладнання.

Комутатори Cisco Catalyst 2960-24TT:

Ці комутатори забезпечують високу пропускну здатність завдяки підтримці Gigabit Ethernet, що є необхідним для обробки великих обсягів мережевого трафіку в корпоративному середовищі. Важливою особливістю цих комутаторів є підтримка технології Power over Ethernet (PoE), яка дозволяє жити підключені пристрої, безпосередньо через мережевий кабель. Це спрощує розгортання мережі та знижує витрати на додаткове електроживлення.

Маршрутизатори Cisco 2911:

Призначені для роботи в складних мережах з великою кількістю сервісів. Модель 2911 відрізняється високою продуктивністю та модульністю, що дозволяє легко розширювати її функціональність шляхом додавання нових інтерфейсів та модулів. Крім того, маршрутизатори Cisco 2911 забезпечують високий рівень безпеки передачі даних завдяки вбудованим засобам шифрування та підтримці VPN, що є критично важливим для захисту конфіденційної інформації компанії.

3.3 Розрахунок схеми адресації корпоративної мережі

Поділ мережі 172.24.80.0/21 на підмережі дозволяє розбити великий пул IP-адрес на менші, незалежні сегменти, що сприяє оптимізації використання адресного простору та покращенню управління мережею. У таблиці 3.1 наведено розподіл адресного блоку мережі з урахуванням кількості вузлів для кожної підмережі.

Таблиця 3.1 – Блок адрес мережі та кількість вузлів в кожній підмережі

Блок адрес	LAN1	LAN2	LAN3	LAN4	LAN5
172.24.80.0/24	57	120	161	70	23

Для створення п'яти підмереж, здатних обслуговувати 431 користувача, застосовано технологію VLSM (Variable Length Subnet Masking). Цей метод дозволяє створювати підмережі з різними довжинами масок, що забезпечує гнучкість та ефективність використання IP-адрес. VLSM дозволяє адаптувати розмір підмережі до потреб користувачів, уникаючи марнування адресного простору та забезпечуючи можливість масштабування мережі у майбутньому.

1) Підмережа LAN1 для 57 хостів

– Потрібна кількість хостів: 57

– Маска підмережі: /26

– Діапазон адрес: 172.24.80.0 - 172.24.80.63

– Розрахунок маски: Маска /26 означає 26 біт для мережі, 6 для хостів.

Бінарно: 11111111.11111111.11111111.11000000 Десяtkово: 255.255.255.192

2) Підмережа LAN2 для 120 хостів

– Потрібна кількість хостів: 120

– Маска підмережі: /25

– Діапазон адрес: 172.24.80.128 - 172.24.80.255

– Розрахунок маски: Маска /25 означає, що перші 25 бітів маски підмережі використовуються для адресації мережі, а решта 7 бітів — для хостів. Бітова адресація: 11111111.11111111.11111111.10000000
255.255.255.128 у десятковій системі.

3) Підмережа LAN3 для 161 хостів

– Потрібна кількість хостів: 161

– Маска підмережі: /24 (256 можливих хостів, $2^8 = 256$)

– Діапазон адрес: 172.24.81.0 - 172.24.81.255

– Розрахунок маски: Маска /24 означає 24 біти для мережі, 8 для хостів.

Бінарно: 11111111.11111111.11111111.00000000 Десяtkово: 255.255.255.0

4) Підмережа LAN4 для 70 хостів

– Потрібна кількість хостів: 70

– Маска підмережі: /25

– Діапазон адрес: 172.24.82.1 - 172.24.82.127

– Розрахунок маски: Маска /25 Бінарно:

11111111.11111111.11111111.10000000 Десятково: 255.255.255.128

5) Підмережа LAN5 для 23 хостів

– Потрібна кількість хостів: 23

– Маска підмережі: /27 (32 можливих хости, $2^5 = 32$)

– Діапазон адрес: 172.24.80.64 - 172.24.80.96

– Розрахунок маски: Маска /27 означає 27 біт для мережі, 5 для хостів.

Бінарно: 11111111.11111111.11111111.11100000 Десятково: 255.255.255.227

Розроблена адресація підмереж структурована в таблиці 3.2

Таблиця 3.2 – Схема адресації мережі

Назва підмережі	Необхідна кількість вузлів	Адреса підмережі	Маска підмережі	Діапазон допустимих IP-адрес вузлів
LAN1	57	172.24.80.0	/26	172.24.80.0 - 172.24.80.63
LAN2	120	172.24.80.128	/25	172.24.80.128 - 172.24.80.255
LAN3	161	172.24.81.0	/24	172.24.81.0 - 172.24.81.255
LAN4	70	172.24.82.0	/25	172.24.82.1 - 172.24.82.126
LAN5	23	172.24.80.64	/27	172.24.80.65 - 172.24.80.96

Для забезпечення зв'язку між маршрутизаторами виділено блок адрес 172.12.80.0/24. Цей адресний простір розподілено на п'ять підмереж, кожна з яких містить два вузли, використовуючи метод VLSM (Variable Length Subnet Masking). В таблиці 3.3 представлено схему адресації каналів між маршрутизаторами.

Таблиця 3.3 – Схема адресації каналів між маршрутизаторами

Назва мережі	Кількість вузлів	Номер мережі	Маска мережі	Початкове значення діапазону	Кінцеве значення діапазону
WAN1	2	172.24.85.0	/30	172.24.85.1	172.24.85.2
WAN2	2	172.24.85.4	/30	172.24.85.5	172.24.85.6
WAN3	2	172.24.85.8	/30	172.24.85.9	172.24.85.10
WAN4	2	172.24.85.12	/30	172.24.85.13	172.24.85.14
WAN5	2	172.24.85.16	/30	172.24.85.17	172.24.85.18
WAN6	2	172.24.85.20	/30	172.24.85.21	172.24.85.22

У таблиці 3.4 наведена адресація всіх маршрутизаторів мережі.

Таблиця 3.4 – Схема адресації пристроїв

Пристрій	Інтерфейс	IP-адреса	Маска
Holubitskiy _R1	Se0/2/0	172.24.85.21	255.255.255.252
	Se0/3/0	172.24.85.13	255.255.255.252
	Se0/3/1	172.24.85.17	255.255.255.252
	Gig0/0.10	172.24.81.1	255.255.255.192
	Gig0/0.20	172.24.81.65	255.255.255.192
	Gig0/0.30	172.24.81.129	255.255.255.192
	Gig0/0.99	172.24.81.193	255.255.255.240
Holubitskiy _R2	Gig0/0	172.24.80.1	255.255.255.192
	Se0/3/0	172.24.85.1	255.255.255.252
	Se0/3/0	172.24.85.5	255.255.255.252
Holubitskiy _R3	Gig0/0	172.24.80.65	255.255.255.224
	Se0/2/0	172.24.85.18	255.255.255.252
	Se0/2/1	209.165.202.2	255.255.255.252
	Se0/3/0	172.24.85.2	255.255.255.252
	Se0/3/1	172.24.85.10	255.255.255.252
Holubitskiy _R4	Se0/2/0	172.24.85.14	255.255.255.252
	Se0/3/0	172.24.85.6	255.255.255.252

Продовження таблиці 3.4.

	Se0/3/1	172.24.85.9	255.255.255.252
Holubitskiy _R5	Gig0/0	172.24.80.129	255.255.255.128
	Se0/3/0	172.24.85.22	255.255.255.252
Holubitskiy _R0	Gig0/0	172.24.82.1	255.255.255.128
	Gig0/1	64.100.13.1	255.255.255.252

3.4 Вибір та налаштування способу маршрутизації

3.4.1 Базове налаштування конфігурації пристроїв

Базове налаштування конфігурації пристроїв за прикладом маршрутизатору Holubitsky_R3:

```
hostname Holubitsky_R3 //задано назву пристрою
line console 0 //переход в конфігураційний режим
password cisco //пароль до консолі
login //вимкнено авторомний доступ
line vty 0 15//переход в конфігураційний режим
лінії VTY
password cisco //пароль до лінії VTY
login //вимкнено автономний доступ
enable secret class //зашифрований пароль до
превілейованого режиму
service password-encryption //вмикання шифру
паролів
banner motd #Holubitsky_R3# //призначено банер
line vty 0 15 // переход в конфігураційний режим
лінії VTY
transport input ssh // використовувати протокол
з'єднання SSH
login local // локальна автентифікація
username Holubitsky password admincisco// імя
користувачу та пароль
ip domain-name Holubitsky_R3 //імя домену
crypto key generate rsa // створення ключу
шифрування
1024 // встановлення довжини ключа
```

Технологія EtherChannel дозволяє об'єднати кілька фізичних інтерфейсів комутатора в один логічний канал, підвищуючи пропускну здатність та забезпечуючи резервування. На рисунках 3.4 – 3.5 відображено кроки для налаштування EtherChannel на комутаторі у мережі LAN_1.

```

hostname Switch
!
!
!
!
!
!
spanning-tree mode pvst
spanning-tree extend system-id
!
interface Port-channel1
 switchport mode trunk
!
interface Port-channel2
 switchport mode trunk
!
interface FastEthernet0/1
 switchport mode trunk
 channel-group 1 mode active
!
interface FastEthernet0/2
 switchport mode trunk
 channel-group 1 mode active
!
interface FastEthernet0/3
 switchport mode trunk
 channel-group 2 mode active
!
interface FastEthernet0/4
 switchport mode trunk
 channel-group 2 mode active
!
interface FastEthernet0/5
!
interface FastEthernet0/6
!

```

Рисунок 3.2 – Налаштування EtherChannel

```

!
interface FastEthernet0/20
!
interface FastEthernet0/21
!
interface FastEthernet0/22
!
interface FastEthernet0/23
!
interface FastEthernet0/24
!
interface GigabitEthernet0/1
!
interface GigabitEthernet0/2
!
interface Vlan1
 no ip address
 shutdown
!
!
!
!
line con 0
!
line vty 0 4
 login
line vty 5 15
 login

```

Рисунок 3.3 – Налаштування EtherChannel

Аналогічні команди налаштування EtherChannel виконано на всіх комутаторах, що беруть участь у створенні каналу.

3.4.2 Налаштування протоколу OSPF та DHCP для маршрутизаторів корпоративної мережі

Для автоматичного призначення IP-адрес комп'ютерам у мережі буде використано протокол DHCP (Dynamic Host Configuration Protocol). Це дозволить спростити адміністрування мережі та зменшити ризик помилок при налаштуванні пристроїв, оскільки користувачам не потрібно буде вручну вводити IP-адреси, маски підмережі та інші параметри. Крім того, DHCP забезпечує більш ефективне використання адресного простору, оскільки адреси призначаються динамічно і звільнюються після відключення пристрою від мережі.

Налаштування DHCP на прикладі маршрутизатора Holubitskiy_R5:

```
ip dhcp excluded-address 172.24.82.1 172.24.82.5
ip dhcp pool LAN-4
network 172.24.82.0 255.255.255.128
default-router 172.24.82.1
dns-server 172.24.81.190
```

Для забезпечення зв'язку між різними підмережами застосовано динамічну маршрутизацію за допомогою протоколу OSPF (Open Shortest Path First). Цей протокол дозволяє автоматично визначати та оновлювати маршрути між підмережами, що робить мережу більш гнучкою та стійкою до змін. OSPF використовує алгоритм для пошуку найкоротших шляхів, що забезпечує високу швидкість та ефективність передачі даних. Крім того, OSPF підтримує VLSM, що дозволяє використовувати різні маски підмережі в різних частинах мережі, оптимізуючи використання адресного простору.

Налаштування протоколу OSPF на прикладі маршрутизатора Holubitskiy _R0:

```
router ospf 1
network 172.24.82.0 0.0.0.127 area 0
network 64.100.13.0 0.0.0.3 area 0
network 172.24.81.0 0.0.0.63 area 0
network 172.24.81.64 0.0.0.63 area 0
network 172.24.81.128 0.0.0.63 area 0
network 172.24.81.192 0.0.0.63 area 0
```

На маршрутизаторі Holubitskiy _R3 налаштовано маршрут за замовчуванням до маршрутизатора ISP і виконано його розповсюдження, та додано статичний маршрут до мережі провайдера ISP:

```
ip route 209.165.201.0 255.255.255.240 209.165.202.2
```

Рисунок 3.4 – Налаштування маршруту за замовчуванням

3.4.3 Налаштування роботи Інтернет

Для забезпечення доступу системи до Інтернету та зовнішніх ресурсів на маршрутизаторі Holubitskiy_R3 налаштовано технологію NAT (Network Address Translation). NAT дозволяє приховати внутрішні IP-адреси пристроїв у локальній мережі, замінюючи їх на одну або декілька публічних IP-адрес при виході в Інтернет.

Для реалізації NAT-трансляції створено пул адрес NAT, який включає діапазон публічних IP-адрес від 209.165.202.5 до 209.165.202.30. Цей пул адрес буде використовуватися для перетворення внутрішніх адрес пристроїв у локальній мережі на публічні при відправленні запитів в Інтернет, та навпаки, при отриманні відповідей з Інтернету. Такий підхід дозволяє забезпечити доступ до Інтернету для всіх пристроїв у локальній мережі, використовуючи лише обмежену кількість публічних IP-адрес.

Налаштовано NAT на прикладі маршрутизатора Holubitskiy _R3:

```

ip access-list extended NAT5
deny ip 172.24.80.0 0.0.0.63 172.24.82.0 0.0.0.127
    пакетів з віддаленої мережі до основної мережі
deny ip 172.24.80.128 0.0.0.127 172.24.82.0
0.0.0.127
deny ip 172.24.81.0 0.0.0.255 172.24.82.0 0.0.0.127
deny ip 172.24.80.64 0.0.0.31 172.24.82.0 0.0.0.127
deny ip 172.12.85.0 0.0.0.255 172.24.82.0 0.0.0.127
permit ip 172.24.80.0 0.0.0.63 any
permit ip 172.24.80.128 0.0.0.127 any
permit ip 172.24.81.0 0.0.0.255 any
permit ip 172.24.80.64 0.0.0.31 any
permit ip 172.12.85.0 0.0.0.255 any
ip nat pool Internet 209.165.200.6 209.165.200.30
netmask 255.255.255.224
ip nat inside source list NAT12 pool Internet
ip nat inside source static 172.24.85.5
209.165.200.3
ip nat inside source static 172.24.81.190
209.165.200.3
ip nat inside source static 172.24.80.60
209.165.200.3
interface Serial0/2/1
ip nat outside
interface Serial0/2/0
ip nat inside
interface Serial0/3/0
ip nat inside
interface Serial0/3/1
ip nat inside
interface GigabitEthernet0/0
ip nat inside

```

3.5 Налаштування мереж VLAN, маршрутизації між VLAN

Технологія VLAN дозволяє створити кілька віртуальних локальних мереж на базі однієї фізичної інфраструктури. Це означає, що пристрої, які фізично підключені до одного комутатора, можуть бути розділені на різні логічні групи, ніби вони підключені до різних комутаторів. Такий підхід

забезпечує гнучкість та ефективність управління мережею, дозволяючи адаптувати її до потреб різних груп користувачів або пристроїв.

Застосування VLAN дозволяє значно спростити фізичне розташування пристроїв та кабелів. Замість того, щоб прокладати окремі кабелі для кожної групи користувачів, VLAN дозволяє об'єднувати їх в одну фізичну мережу, розділяючи лише логічно. Це не тільки зменшує витрати на обладнання та кабелі, але й спрощує управління та підтримку мережі.

Важливою перевагою VLAN є підвищення рівня безпеки. Завдяки ізоляції трафіку між різними VLAN, зменшується ризик несанкціонованого доступу до чутливої інформації та поширення шкідливого програмного забезпечення. Кожна VLAN може мати свої власні правила доступу та політики безпеки, що дозволяє забезпечити індивідуальний підхід до захисту даних кожної групи користувачів.

Таблиця 3.5 – Адресація мереж VLAN

Назва	Мережева адреса	Маска	Діапазон використання
VLAN10	172.24.81.0/26	/26	172.24.81.1 - 172.24.81.62
VLAN20	172.24.81.64/26	/26	172.24.81.65 - 172.24.81.126
VLAN30	172.24.81.128/26	/26	172.24.81.129 - 172.24.81.190
VLAN99	172.24.81.192/26	/26	172.24.81.193 - 172.24.81.254

Налаштувано VLAN на прикладі комутатора з мережі LAN3:

```
int range fa0/5-10
switchport mode access
switchport access vlan 20
int range fa0/12-14
switchport mode access
switchport access vlan 10
int range fa0/15-24
switchport mode access
switchport access vlan 30
int range fa0/1-4
switchport mode trunk
switchport trunk native vlan 100
switchport trunk allowed vlan 20,10,30-100
```

3.6 Перевірка комп'ютерної мережі комплексу

Для забезпечення коректної роботи та безпеки мережі проведено перевірку базових налаштувань обладнання на прикладі маршрутизатора Holubitskiy_R5. Використовуючи команду `show running-config` у привілейованому режимі, проаналізовано ключові параметри конфігурації:

- перевірено унікальність імені маршрутизатора для його ідентифікації в мережі;
- проаналізовано налаштування паролів для доступу до консолі та ліній vty (віртуальні термінали), що використовуються для віддаленого управління маршрутизатором. Особливу увагу було приділено використанню надійних паролів та протоколу SSH для захисту від несанкціонованого доступу;
- перевірено наявність та складність паролю, необхідного для виконання привілейованих команд на маршрутизаторі;
- проаналізовано вміст банера, який відображається при підключенні до маршрутизатора, на предмет відповідності політикам безпеки компанії;
- перевірено коректність налаштування імені домену, що використовується для ідентифікації пристроїв у мережі.

```
-----
| !
| hostname Holubitsky_R5
| !
|
```

Рисунок 3.5 – Назва пристрою

```

| line con 0
| password 7 0822455D0A16
| -----

```

Рисунок 3.6 – Пароль до консолі


```

!
line vty 0 4
 password 7 0822455D0A16
 login local
 transport input ssh
line vty 5 15
 password 7 0822455D0A16
 login local
 transport input ssh
!

```

Рисунок 3.7 – Пароль ліній VTY та протоколу SSH

```

!
enable secret 5 $1$mERr$9cTjUIEqNGurQiFU.ZeCil

```

Рисунок 3.8 – Пароль до привілейованого режиму

```

!
banner motd ^CHolubitsky_R5^C
!

```

Рисунок 3.9 – Баннер MOTD

```

ip domain-name Holubitsky_5

```

Рисунок 3.10 – Ім'я домена

Усі паролі на маршрутизаторах зашифровано. Перевірка технології EtherChannel в LAN_1 відображена на рисунку 3.11.

```

Switch>enable
Switch#show eth
Switch#show etherchannel sum
Switch#show etherchannel summary
Flags: D - down          P - in port-channel
       I - stand-alone  s - suspended
       H - Hot-standby (LACP only)
       R - Layer3       S - Layer2
       U - in use       f - failed to allocate aggregator
       u - unsuitable for bundling
       w - waiting to be aggregated
       d - default port

Number of channel-groups in use: 2
Number of aggregators:          2

Group  Port-channel  Protocol    Ports
-----+-----+-----+-----
1      Po1(SU)          LACP       Fa0/1(P) Fa0/2(P)
2      Po2(SU)          LACP       Fa0/3(P) Fa0/4(P)

```

Рисунок 3.11 – Технологія EtherChannel

Перевірено налаштування маршрутизаторів на прикладі Holubitskiy_R5:

```
router ospf 1
 log-adjacency-changes
 passive-interface default
 no passive-interface GigabitEthernet0/0
 no passive-interface Serial0/3/0
 network 172.24.85.20 0.0.0.3 area 0
 network 172.24.80.128 0.0.0.127 area 0
.
```

Рисунок 3.12 – Налаштування OSPF

На рисунку 3.13 відображено перевірку зв'язку між різними підмережами. Результат успішний.

```
Successful VLA... PC6
Successful PC6 VLAN10
```

Рисунок 3.13 – Зв'язок між LAN_4 та LAN_5

Перевірено налаштування маршруту за замовчуванням на маршрутизаторі, підключеному до ISP провайдера.

```
Holubitsky_R3#show ip route st
Holubitsky_R3#show ip route static
S* 0.0.0.0/0 is directly connected, Serial0/3/1
```

Рисунок 3.14 – Налаштований маршрут за замовчуванням на маршрутизаторі

Перевірено налаштування DHCP на прикладі комп'ютера PC2, який знаходиться в підмережі LAN_2.

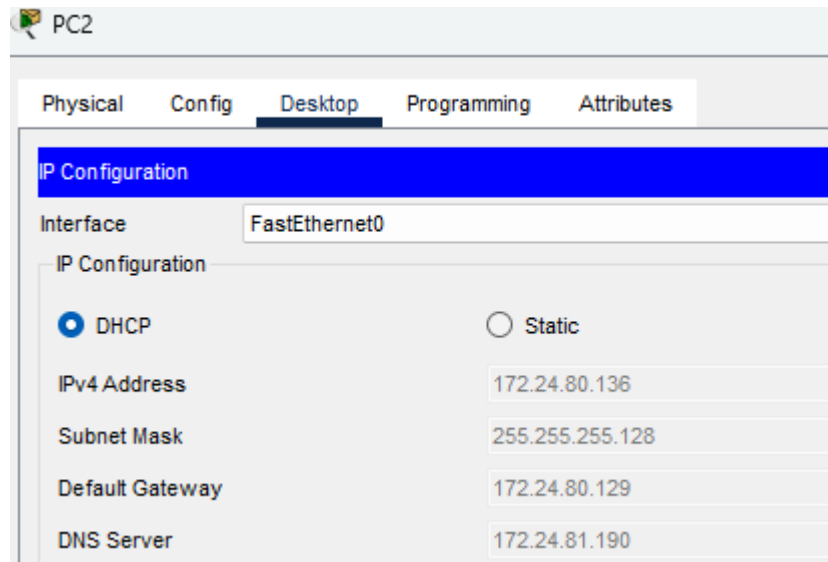


Рисунок 3.15 – IP-адреса PC2

Проведено перевірку призначених IP-адрес маршрутизаторам на прикладі Holubitskiy_R2.

```
Holubitsky_R2#show ip interface br
Holubitsky_R2#show ip interface brief
Interface                IP-Address      OK? Method Status      Protocol
GigabitEthernet0/0      172.24.80.1    YES manual up          up
GigabitEthernet0/1      unassigned      YES unset  administratively down down
GigabitEthernet0/2      unassigned      YES unset  administratively down down
Serial0/0/0             172.24.85.1    YES manual up          up
Serial0/0/1             172.24.85.5    YES manual up          up
Vlan1                   unassigned      YES unset  administratively down down
Holubitsky_R2#
```

Рисунок 3.16 – IP-адреси маршрутизатора

Проведено перевірку призначених імен різним VLAN та портах, які належать кожному VLAN.

VLAN Name	Status	Ports
1 default	active	Fa0/4, Fa0/5, Gig0/1, Gig0/2
10 VLAN0010	active	Fa0/6, Fa0/7, Fa0/8, Fa0/9 Fa0/10
20 VLAN0020	active	Fa0/11, Fa0/12, Fa0/13, Fa0/14 Fa0/15, Fa0/16
30 VLAN0030	active	Fa0/17, Fa0/18, Fa0/19, Fa0/20 Fa0/21, Fa0/22, Fa0/23, Fa0/24
1002 fddi-default	active	
1003 token-ring-default	active	
1004 fddinet-default	active	
1005 trnet-default	active	

VLAN	Type	SAID	MTU	Parent	RingNo	BridgeNo	Stp	BrdgMode	Trans1	Trans2
1	enet	100001	1500	-	-	-	-	-	0	0
10	enet	100010	1500	-	-	-	-	-	0	0
20	enet	100020	1500	-	-	-	-	-	0	0
30	enet	100030	1500	-	-	-	-	-	0	0
1002	fddi	101002	1500	-	-	-	-	-	0	0
1003	tr	101003	1500	-	-	-	-	-	0	0
1004	fdnet	101004	1500	-	-	-	ieee	-	0	0
1005	trnet	101005	1500	-	-	-	ibm	-	0	0

VLAN Type	SAID	MTU	Parent	RingNo	BridgeNo	Stp	BrdgMode	Trans1	Trans2
Remote SPAN VLANs									

Primary	Secondary	Type	Ports

Рисунок 3.17 – Імена та порти VLAN

```
Switch#show interfaces tr
Switch#show interfaces trunk
Port      Mode      Encapsulation  Status      Native vlan
Fa0/1     on        802.1q         trunking    1
Fa0/2     on        802.1q         trunking    1
Fa0/3     on        802.1q         trunking    1

Port      Vlans allowed on trunk
Fa0/1     1-1005
Fa0/2     1-1005
Fa0/3     1-1005

Port      Vlans allowed and active in management domain
Fa0/1     1,10,20,30
Fa0/2     1,10,20,30
Fa0/3     1,10,20,30

Port      Vlans in spanning tree forwarding state and not pruned
Fa0/1     1,10,20,30
Fa0/2     1,10,20,30
Fa0/3     1,10,20,30
```

Рисунок 3.18 – Транкові порти

Проведено перевірку налаштувань DHCP для VLAN на прикладі PC12, який знаходиться в VLAN10.

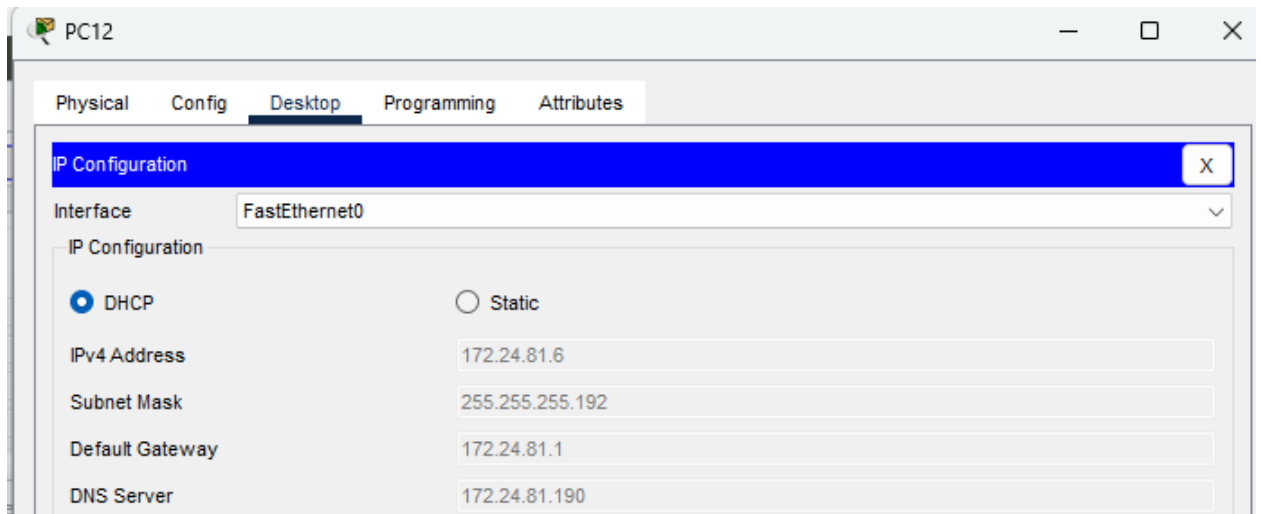


Рисунок 3.19 – IP-адреса PC12

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC13	PC14	ICMP		0.000	N	0	(edit)	(delete)
	Successful	PC14	PC13	ICMP		0.000	N	1	(edit)	(delete)

Рисунок 3.20 – Зв'язок між VLAN10 та VLAN20

Для перевірки коректності відображення веб-сайту з інформацією про тему та завдання кваліфікаційної роботи на прикладі ПК14, введено адресу. Результат відображено на рисунку 3.21.

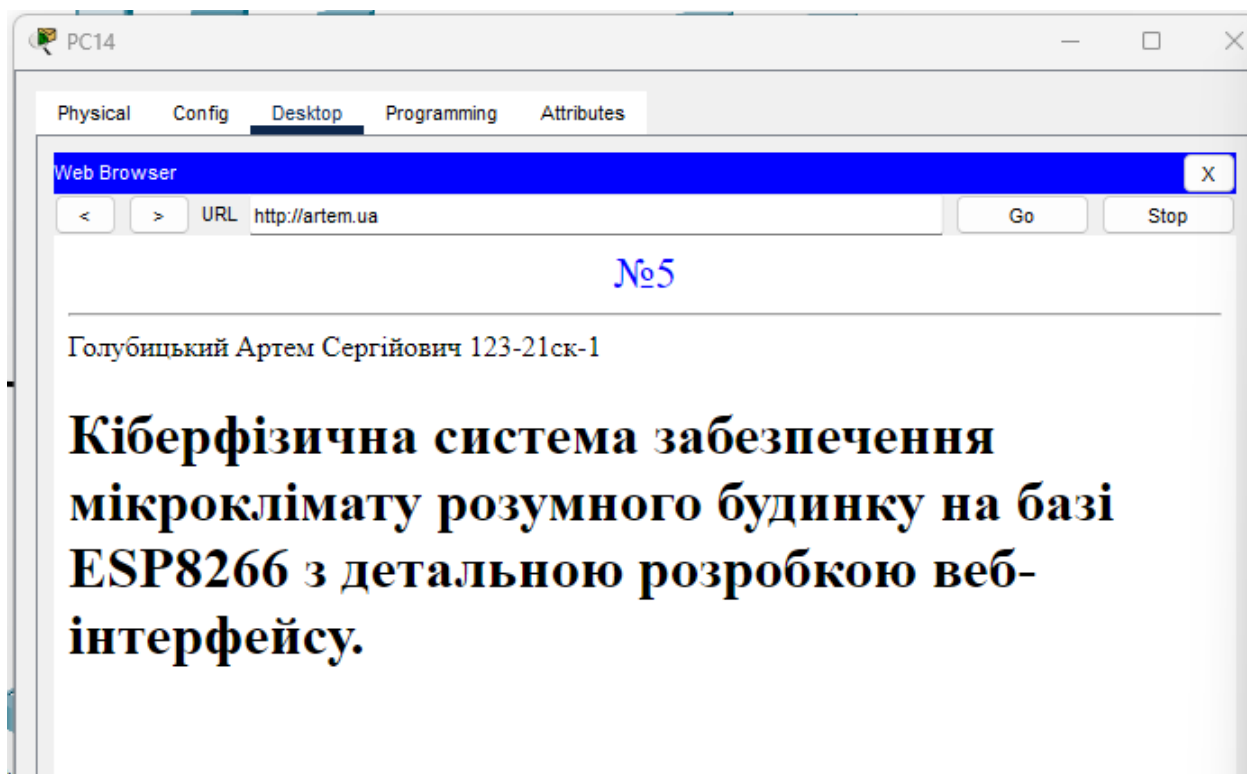


Рисунок 3.21 – Веб-сторінка з відомостями

4 РОЗРОБКА КОМПОНЕНТА СИСТЕМИ

4.1 Розробка контролера кіберфізичної системи забезпечення мікроклімату

4.1.1 Опис датчиків та їх характеристик

Відповідно до специфікації розробленій у розділі 2.3 у кіберфізичній системі забезпечення мікроклімату використовуються три типи датчиків для збору даних про температуру, вологість та концентрацію вуглекислого газу: ВМЕ280 (рис 4.2), АНТ10 (рис 4.3) та МН-Z19В (рис 4.4) та мікроконтролер Wemos D1 Mini ESP8266 (рис 4.1) відповідно. В таблиці 4.1 відображена структурна характеристика використаних датчиків.

Таблиця 4.1 – Основні характеристики використаних датчиків

Характеристика	ВМЕ280	АНТ10	МН-Z19В
±50 ppm ±5% показань	Температура, вологість, тиск	Температура, вологість	Температура, CO2
Інтерфейс	I2C, SPI	I2C	UART
Діапазон вимірювання температури	-40...+85 °C	-40...+80 °C	0...+50 °C
Точність вимірювання температури	±1 °C	±0.3 °C	±1.5 °C
Діапазон вимірювання вологості	0...100 %	20...80 %	-
Точність вимірювання вологості	±3 %	±2 %	-
Діапазон вимірювання CO2	-	-	400...5000 ppm
Точність вимірювання CO2	-	-	±50 ppm ±5%



Рисунок 4.1 – Мікроконтролер Wemos D1 Mini

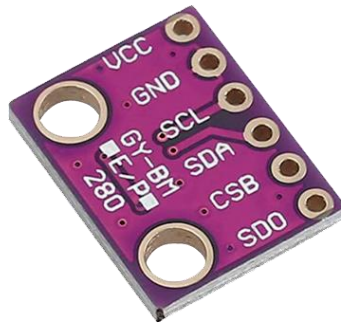


Рисунок 4.2 – Датчик BME280



Рисунок 4.3 – Датчик АНТ10

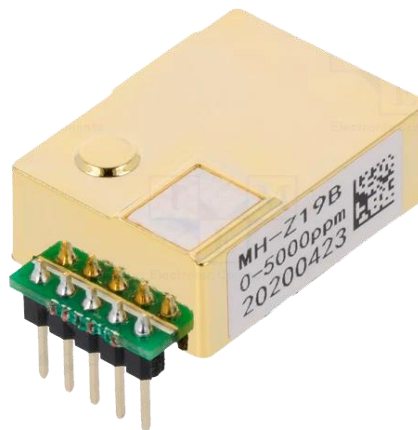


Рисунок 4.4 – Датчик MHZ-19B

4.1.2 Розробка схеми підключення датчиків

4.1.2.1 Розробка схеми підключення внутрішніх датчиків

Датчики розміщені на відстані один від одного, щоб мінімізувати взаємний вплив та забезпечити точність вимірювань.

BME280: Датчик температури, вологості та тиску підключений до мікроконтролера за допомогою інтерфейсу I2C. Використовуються наступні піни: VIN (3.3v), GND, SDA-SCL (D1) , SCL-SDA (D2).

MHZ19B: Інфрачервоний датчик CO2, підключений до мікроконтролера за допомогою інтерфейсу UART. Використовуються наступні піни: VIN (5v), GND, RX-TX, TX-RX.

На рисунку 4.5 відображено схему підключень датчиків. На рисунку 4.6 – 4.7 зображено зовнішній вигляд кінцевого модуля на платі.

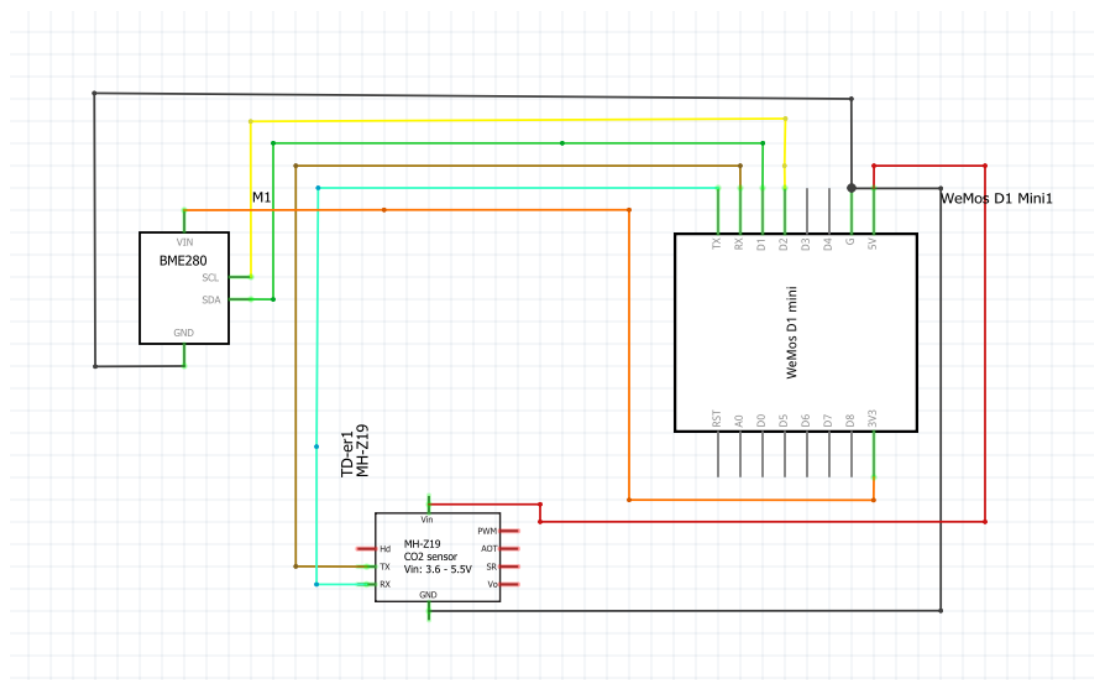


Рисунок 4.5 – Схема підключення внутрішнього модуля

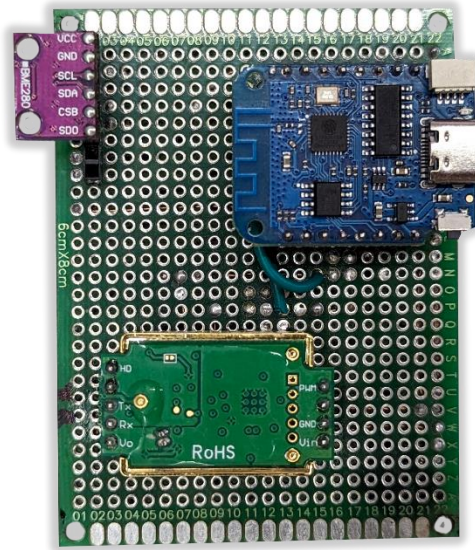


Рисунок 4.6 – Зібраний внутрішній модуль

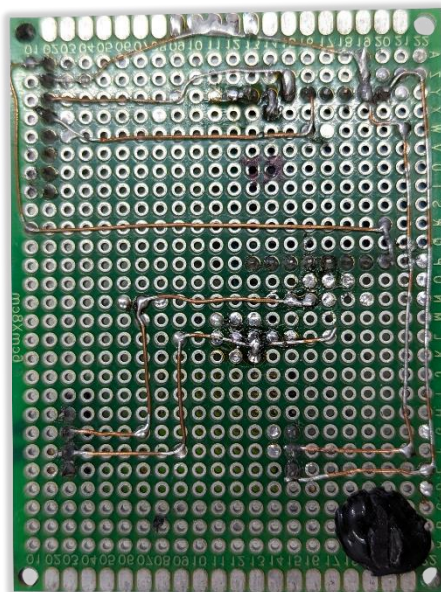


Рисунок 4.7 – Сторона пайки внутрішнього модуля

4.1.2.2 Розробка схеми підключення зовнішніх датчиків

АНТ10: Датчик температури та вологості підключений до другого мікроконтролера ESP8266 за допомогою інтерфейсу I2C. Для живлення датчика використовується пін VIN (3.3V), а для заземлення - пін GND. Лінії передачі даних SDA та SCL підключаються до пінів D1 та D2 мікроконтролера

відповідно. На рисунку 4.8 відображено схему підключень датчиків. На рисунку 4.9 – 4.10 зображено зовнішній вигляд кінцевого модуля на платі.

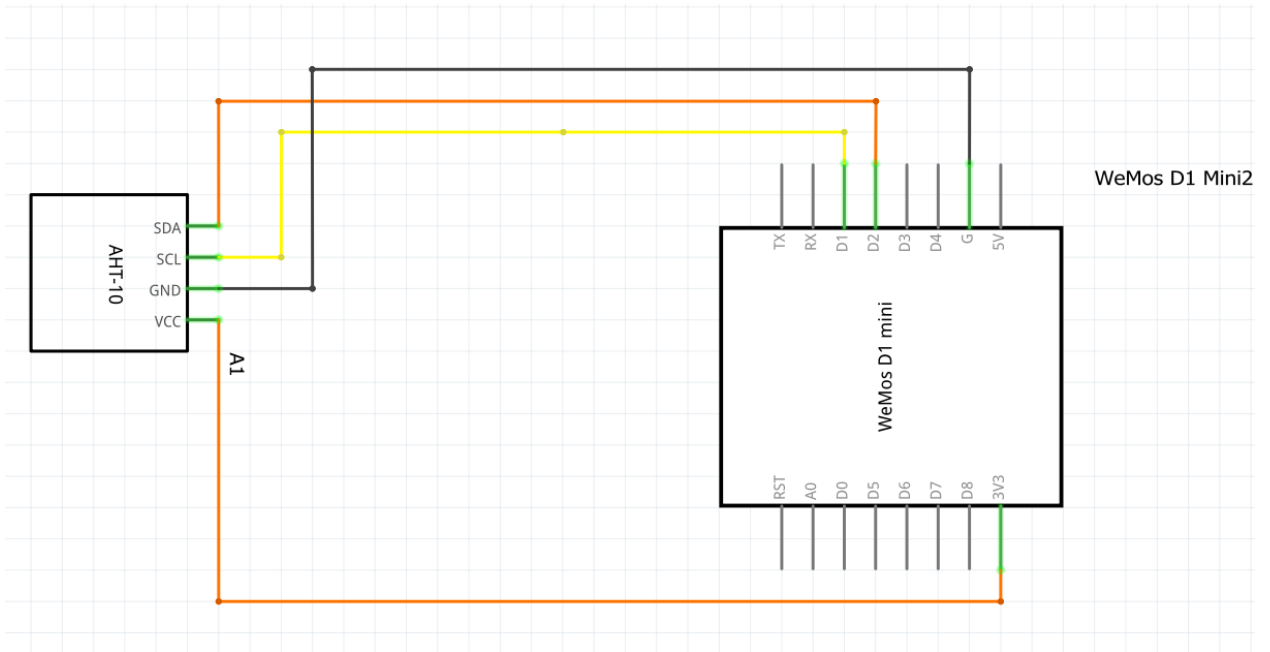


Рисунок 4.8 – Схема підключення зовнішнього модуля

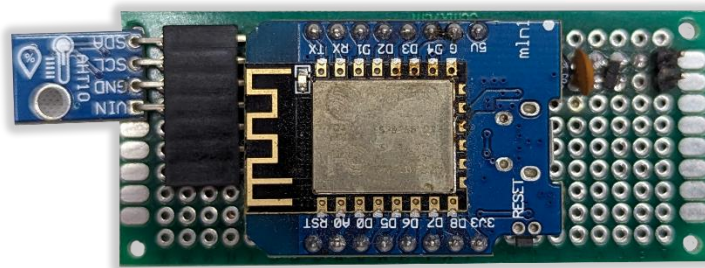


Рисунок 4.9 – Зібраний зовнішній модуль

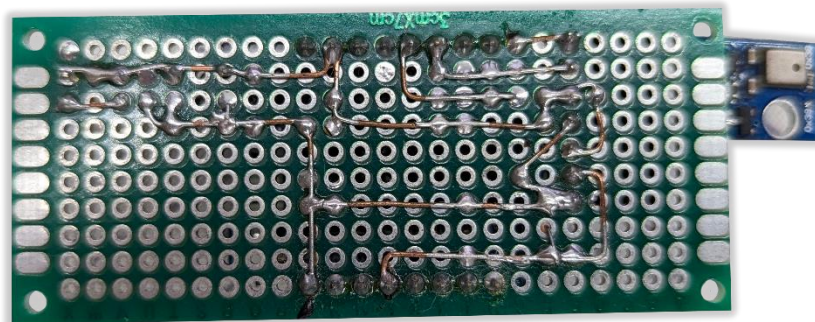


Рисунок 4.10 – Сторона пайки зовнішнього модуля

4.1.2.3 Розробка схеми підключення виконавчого пристрою

На рисунку 4.11 зображено підключення виконавчого пристрою (реле) до мікроконтролера ESP8266 через транзисторний ключ.

На схему подається напруга 5 В від джерела живлення.

Мікроконтролер ESP живиться від джерела 3.3 В через адаптер ESP-01. Один з пінів мікроконтролера (GPIO) використовується для управління транзистором.

Транзистор використовується як електронний ключ для керування реле. Коли на базу транзистора подається високий рівень сигналу (3.3 В) з мікроконтролера, транзистор відкривається, дозволяючи струму протікати через обмотку реле. Резистор обмежує струм бази транзистора.

Реле використовується для комутації зовнішнього навантаження (споживача) змінним струмом 220 В. Коли транзистор відкритий, струм протікає через обмотку реле, замикаючи його контакти та вмикаючи споживача.

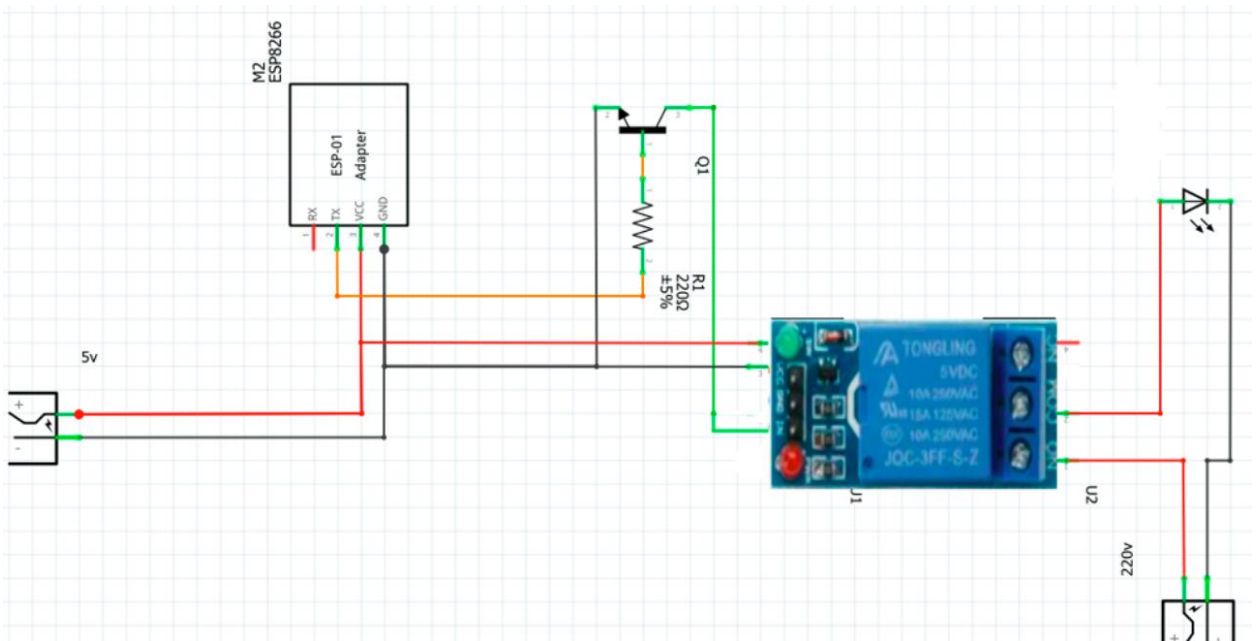


Рисунок 4.11 – Схема підключення зовнішнього модуля

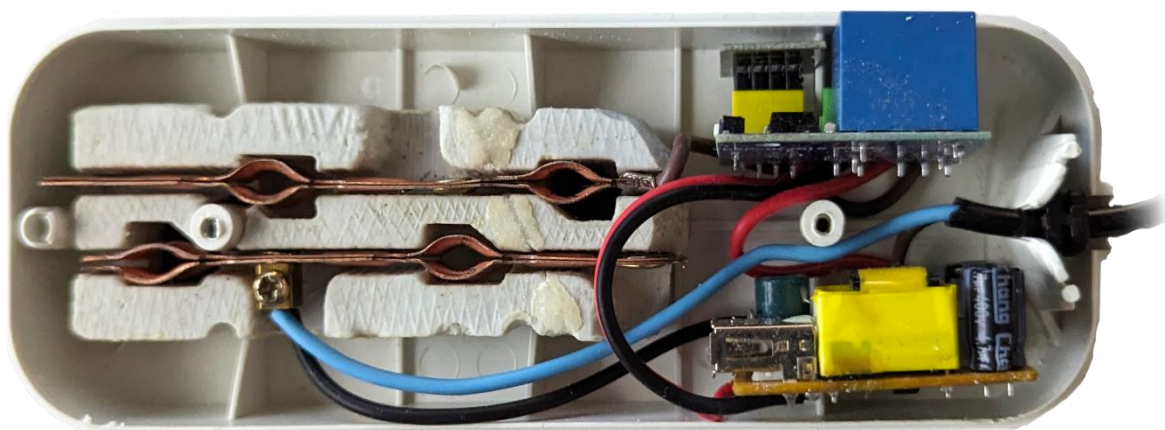


Рисунок 4.12 – Зібраний виконавчий пристрій

4.2 Розробка функціонального модуля

4.2.1 Розробка внутрішнього компонента системи

Внутрішній компонент системи повинен розташовуватися у власному корпусі з достатнім рівнем проходження потоків повітря задля точних вимірювань. Задля цього корпус має достатній рівень вентиляційних отворів. На рисунку 4.13 відображено зовнішній вигляд кінцевого внутрішнього модулю.



Рисунок 4.13 – Внутрішній компонент системи

4.2.2 Розробка зовнішнього компонента системи

Зовнішній компонент системи розташовується на вулиці, закріплений на стіні або подовжуючій рійці. Корпус розроблений таким чином, щоб мінімізувати шкоду навколишнього (дощу та вітру) середовища на датчик. Мікроконтролер розташований на віддалені від датчика в безпечному від вологи місці, та під'єднаний до датчика за допомогою шлейфа. На рисунку 4.14 відображено зовнішній вигляд кінцевого зовнішнього модулю.

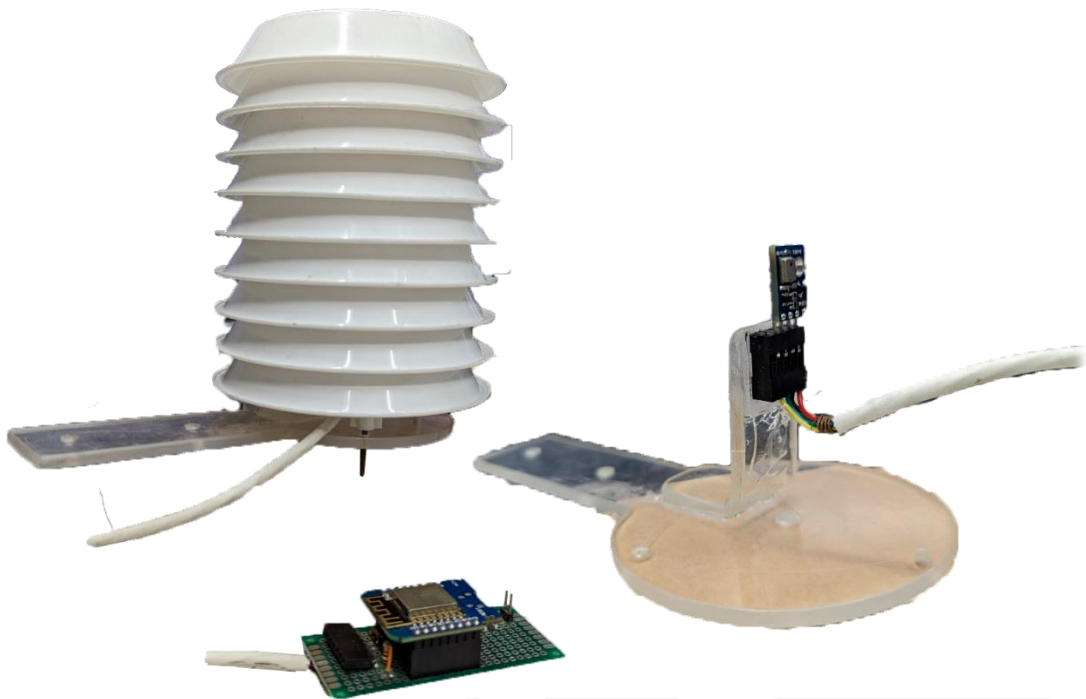


Рисунок 4.14 – Зовнішній компонент системи

4.3 Розробка веб сторінок інтерфейсу користувача

4.3.1 Розробка головної сторінки

Головна сторінка `index.html` служить основним інформаційним та керуючим центром для користувача кіберфізичної системи забезпечення мікроклімату. Її основні цілі:

- відображення поточного стану мікроклімату, а саме надання користувачу актуальної інформації про температуру, вологість, рівень CO₂ та стан пристроїв (увімкнено/вимкнено) в режимі реального часу;
- можливість керування системою за допомогою інтуїтивно зрозумілого інтерфейсу для ручного керування кліматичними пристроями, зміни налаштувань та встановлення розкладу роботи;
- забезпечення швидкого доступу до інших сторінок веб-інтерфейсу (сторінок детальної інформації, графіків, налаштувань);
- відображення статусу системи через інформування користувача про стан підключення системи датчиків до сервера (онлайн/офлайн).

Ключові моменти розробки:

При розробці головної сторінки `index.html` приділено особливу увагу створенню логічної та зручної структури, використовуючи розмітку HTML5. Застосування CSS дозволило створити привабливий та інтуїтивно зрозумілий дизайн, адаптований для різних пристроїв. Інтерактивність сторінки забезпечується за допомогою JavaScript, що дозволяє оновлювати дані в режимі реального часу, перемикає стан пристроїв та змінювати параметри кліматичної системи без перезавантаження сторінки. Для оптимізації роботи та зменшення навантаження на сервер використовується кешування даних. Крім того, реалізовано механізми обробки помилок, які інформують користувача про можливі проблеми та допомагають у їх вирішенні. На рисунках 4.15 та 4.16 відображено зовнішній вигляд головної сторінки.

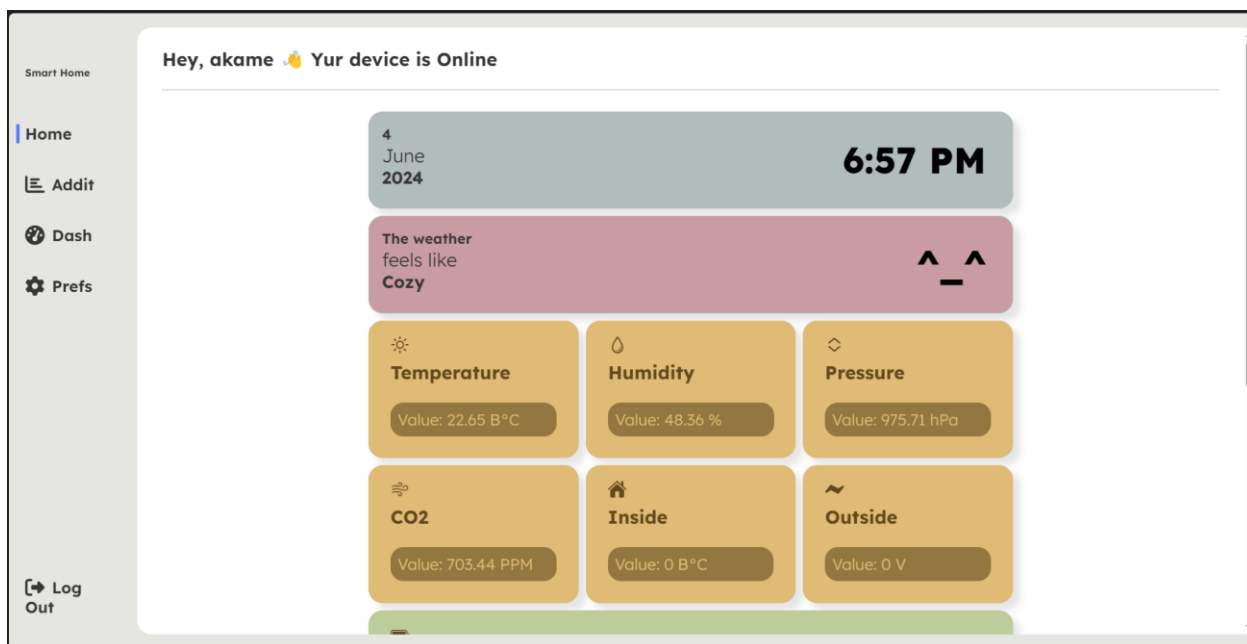


Рисунок 4.15 – Інтерфейс головної сторінки

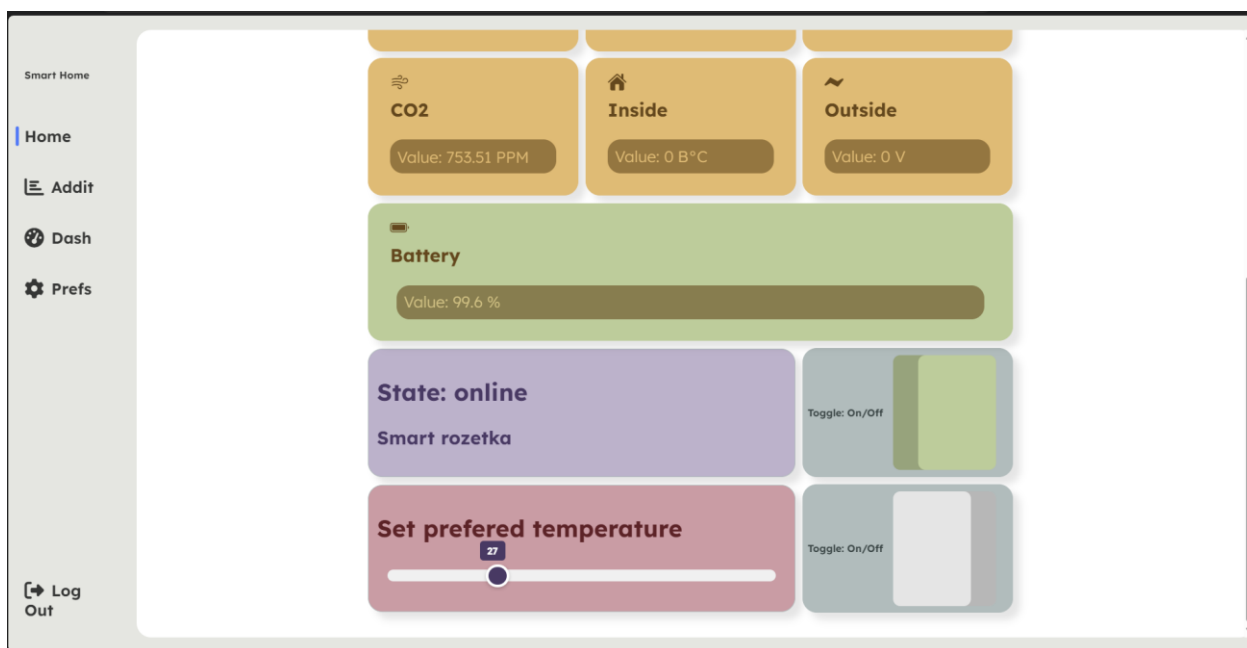


Рисунок 4.16 – Інтерфейс головної сторінки

Опис JavaScript функцій головної сторінки:

Функція `toggleSwitch()` використовується для перемикавання стану реле (ввімкнення/вимкнення) Функціонал:

– Надсилає POST-запит на сервер (`/toggle_switch`) з даними про стан реле.

- Отримує JSON-відповідь від сервера зі станом реле.
- Оновлює стан чекбокса відповідно до стану реле, отриманого від сервера.

Функція `checkTemperatureAndControlRelay()` автоматично керує реле в залежності від поточної температури та встановленого бажаного значення.

Функціонал:

- перевіряє, чи увімкнений режим автоматичного керування (`controlSwitch.checked`);
- отримує поточну температуру з сервера (`/get_temperature`);
- порівнює поточну температуру з бажаною температурою, встановленою за допомогою повзунка;
- викликає функцію `toggleSwitch` для увімкнення або вимкнення реле, якщо необхідно;
- встановлює затримку 30 секунд для повторної перевірки температури.

Додаткові функції:

- `updateSeekBarValuePosition()` оновлює позицію значення `seekbar` на основі поточного значення;
- `updateDateTime()` оновлює час на сторінці;
- `updateData()` отримує дані з сервера та оновлює значення датчиків на сторінці;
- `checkDataAndAlert()` перевіряє, чи не перевищують значення датчиків встановлені пороги, та виводить сповіщення, якщо це так;

4.3.2 Розробка сторінки авторизації

Сторінка `login.html` служить для аутентифікації користувачів та надання їм доступу до керування кіберфізичною системою мікроклімату. Основні цілі сторінки:

- забезпечення безпечного входу через запит та перевірку облікових даних користувача (логін та пароль) для захисту системи від несанкціонованого доступу;

- виведення повідомлень про помилки у разі некоректного введення даних (неіснуючий користувач, неправильний пароль);

- надання можливості перегляду даних без авторизації (режим "Гість").

Ключові моменти розробки:

- застосовано розширення Flask-WTF для створення та валідації форми входу, що спрощує обробку даних, що вводяться користувачем, та підвищує безпеку[6];

- для захисту паролів користувачів використовується хешування з використанням алгоритму PBKDF2 з SHA256. Це дозволяє зберігати паролі у безпечному вигляді, не розкриваючи їх у відкритому вигляді;

- використано сесії для зберігання стану аутентифікації користувача. Після успішного входу в систему, користувач отримує ідентифікатор сесії, який використовується для подальшої авторизації на інших сторінках;

- з міркувань безпеки, при спробі реєстрації з логіном, який вже існує, дані з файлу в якому зберігається уся зібрана статистика очищуються. Це запобігає потенційному доступу нового користувача до даних попереднього власника облікового запису.

Забезпечення безпеки:

- паролі користувачів не зберігаються у відкритому вигляді, а лише їх хеші, що ускладнює їх розшифрування у разі несанкціонованого доступу до файлу з паролями;

- очищення даних при повторній реєстрації запобігає випадковому доступу до даних, що могли залишитися від попереднього користувача з таким самим іменем;

– використання сесій, що дозволяє відстежувати стан аутентифікації користувача та обмежувати доступ до певних функцій системи лише для авторизованих користувачів.

На рисунку 4.17 відображено вигляд сторінки авторизації

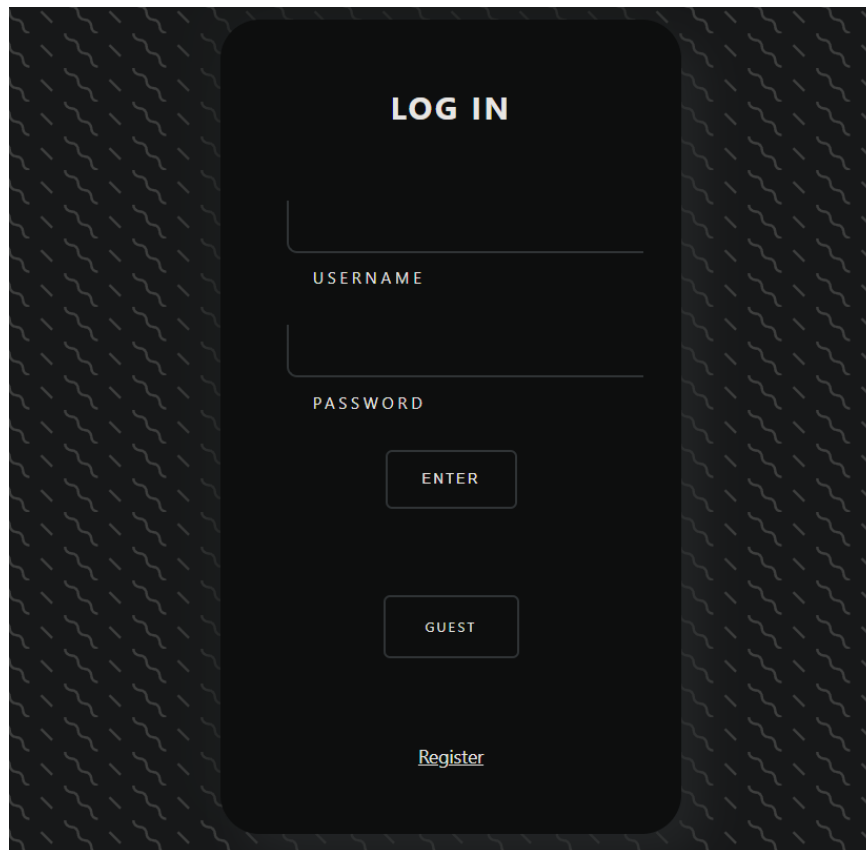


Рисунок 4.17 – Інтерфейс сторінки авторизації

4.3.3 Розробка сторінки з ретельною інформацією

Сторінка `sensor_graph.html` призначена для детального аналізу даних, отриманих з конкретного датчика. Вона дозволяє користувачеві візуалізувати зміни показників у часі, відслідковувати тенденції та виявляти аномалії.

Основні цілі сторінки:

- візуалізація даних за допомогою відображення графіку зміни показників датчика у часі з можливістю масштабування та панорамування;
- обчислення та відображення середнього значення вибраного діапазону даних;

- можливість вибору часового інтервалу для відображення даних за допомогою повзунка;
- можливість обмеження кількості точок, що відображаються на графіку, для кращої читабельності;
- навігація між графіками різних датчиків за допомогою кнопки для переходу до графіків попереднього та наступного датчиків;
- можливість експортувати вибрані записи в таблицю для подальшої обробки або друку.

Ключові моменти розробки:

- застосування бібліотеки Chart.js для створення інтерактивного графіку. Вона дозволяє налаштовувати зовнішній вигляд графіку, додавати підписи до осей, масштабувати та панорамувати графік за допомогою миші або сенсорного екрану;
- дані для графіка завантажуються асинхронно з CSV-файлу за допомогою бібліотеки PapaParse. Це дозволяє уникнути необхідності перезавантаження сторінки при зміні діапазону відображення або кількості видимих точок;
- JavaScript-код розраховує середнє значення вибраного діапазону даних та відображає його на сторінці;
- повзунок дозволяє користувачу змінювати початок та кінець часового інтервалу, дані за який відображаються на графіку;
- поле введення дозволяє користувачу обмежити кількість точок на графіку, що корисно при великому обсязі даних;



Рисунок 4.18 – Інтерфейс сторінки з графіком конкретного сенсора

Опис JavaScript функцій сторінки з графіками:

Функція `getData()` відповідає за отримання та обробку даних з CSV-файлу та створення графіка. Приймає аргумент `sensorName`, який містить в собі назву датчика, дані якого потрібно отримати.

Функціональність:

- відправляє запит на отримання даних з CSV-файлу;
- збирає отриманий текст у формат CSV за допомогою бібліотеки `PapaParse`;
- фільтрує дані, залишаючи тільки записи, що відповідають вказаному датчику;

- формує набір даних для графіка, перетворюючи рядки CSV у точки з координатами часу та значення;
- створює графік за допомогою Chart.js, використовуючи отримані дані та конфігурацію.

4.3.4 Розробка сторінки налаштувань

Сторінка `settings.html` призначена для налаштування параметрів кіберфізичної системи та її компонентів. Основні цілі сторінки:

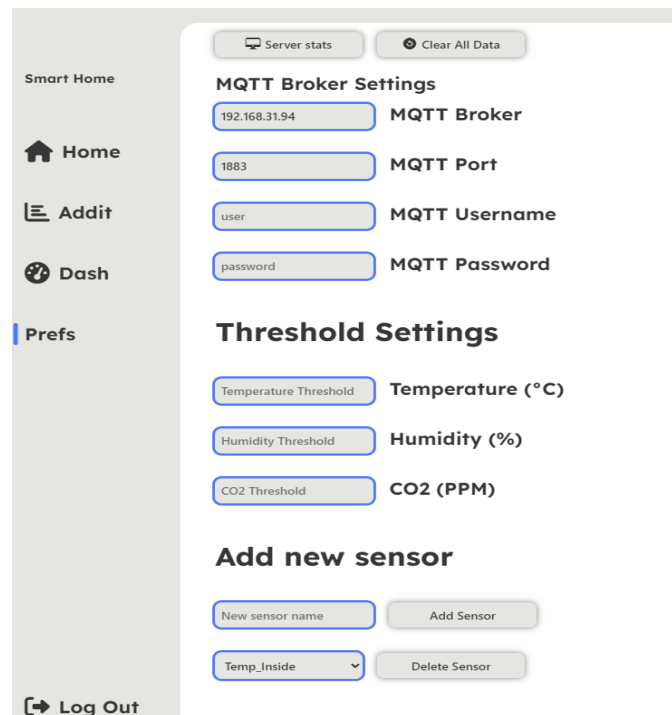
- встановлення адреси та порту MQTT-брокера, а також облікових даних (логін та пароль) для підключення до нього;
- встановлення граничних значень температури, вологості та рівня CO₂, при перевищенні яких система буде автоматично вмикати або вимикати кліматичні пристрої та виводити термінове аварійне повідомлення з назвою датчику та критичним значенням;
- можливість додавання нових датчиків до системи та видалення існуючих;
- зміна назви, одиниць вимірювання та іконок для відображення даних з датчиків на головній сторінці;
- можливість очищення історії даних, що зберігається в системі.

Ключові моменти розробки:

- для введення та відправки налаштувань на сервер використовуються HTML-форми з різними типами полів вводу (текстові поля, числові поля, кнопки);
- налаштування зберігаються в CSV-файлі, що дозволяє легко їх редагувати та зберігати;
- перед збереженням налаштування перевіряються на коректність (наприклад, чи є значення полів числовими);
- після збереження налаштувань користувачу виводиться повідомлення про успішне збереження або про помилку, що виникла;

- сторінка доступна лише авторизованим користувачам;
- передбачено кнопку "Clear All Data", яка дозволяє очистити історію даних, що зберігається в системі;
- передбачено відображення системних параметрів серверу таких як ім'я серверу, операційна система, версія оточення, використання оперативної пам'яті, використання центрального процесору.

Сторінка налаштувань є важливим інструментом для адміністрування кіберфізичної системи забезпечення мікроклімату. Вона дозволяє користувачам налаштовувати параметри системи під свої потреби та забезпечувати її оптимальну роботу.



The screenshot displays a web interface for a smart home system. On the left is a sidebar with navigation options: Smart Home, Home, Addit, Dash, Prefs (highlighted), and Log Out. The main content area is titled 'Smart Home' and contains several sections:

- Server stats** and **Clear All Data** buttons at the top.
- MQTT Broker Settings**:
 - MQTT Broker: 192.168.31.94
 - MQTT Port: 1883
 - MQTT Username: user
 - MQTT Password: password
- Threshold Settings**:
 - Temperature Threshold: Temperature (°C)
 - Humidity Threshold: Humidity (%)
 - CO2 Threshold: CO2 (PPM)
- Add new sensor**:
 - New sensor name: Temp_Inside
 - Add Sensor button
 - Delete Sensor button

Рисунок 4.19 – Інтерфейс сторінки налаштувань

Added sensors

Temp_Inside:	<input type="text" value="B°C"/>	<input type="text" value="thermometer"/>
Temp Outside:	<input type="text" value="B°C"/>	<input type="text" value="thermometer_sun"/>
Pressure:	<input type="text" value="hPa"/>	<input type="text" value="chevron_up_chevron_down"/>
Humidity_Inside:	<input type="text" value="%"/>	<input type="text" value="drop"/>
Humidity Outside:	<input type="text" value="%"/>	<input type="text" value="drop_fill"/>
CO2:	<input type="text" value="PPM"/>	<input type="text" value="heart_fill"/>
Battery:	<input type="text" value="%"/>	<input type="text" value="battery_100"/>
<input type="text" value="New sensor name"/>	<input type="text" value="Unit"/>	<input type="text" value="Icon"/>
<input type="button" value="Submit new sensors"/>		

Preview of New Card:

Рисунок 4.20 – Інтерфейс сторінки налаштувань

ВИСНОВКИ

У результаті проведеної роботи розроблено та реалізовано кіберфізичну систему забезпечення мікроклімату розумного будинку на базі мікроконтролера ESP8266. Система успішно виконує функції збору даних з датчиків температури, вологості та концентрації CO₂, обробки цих даних та автоматичного керування кліматичними пристроями для підтримки заданих параметрів мікроклімату.

Розроблений веб-інтерфейс забезпечує зручний та інтуїтивно зрозумілий спосіб моніторингу та керування системою, дозволяючи користувачам віддалено контролювати мікроклімат у приміщенні та налаштовувати параметри роботи системи.

Отримані результати відповідають сучасному рівню розвитку технологій Інтернету речей та кіберфізичних систем. Використання мікроконтролера ESP8266 дозволило створити економічно ефективне рішення, доступне для широкого кола споживачів. Застосування відкритих протоколів та інтерфейсів забезпечує можливість інтеграції системи з іншими компонентами розумного будинку.

Розроблена система може бути використана в житлових приміщеннях та офісах для підвищення комфорту, енергоефективності та забезпечення здорового мікроклімату. Наукова та науково-технічна значущість роботи полягає у розробці та пробі нових підходів до створення кіберфізичних систем управління мікрокліматом на базі доступних компонентів.

Подальший розвиток може бути спрямований на розширення функціональності системи, інтеграцію з більшою кількістю датчиків та кліматичних пристроїв, впровадження більш складних алгоритмів керування на основі машинного навчання та штучного інтелекту, а також на підвищення енергоефективності та безпеки системи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Методичні рекомендації до виконання кваліфікаційної роботи бакалавра студентами галузі знань 12 Інформаційні технології спеціальності 123 Комп'ютерна інженерія / Л.І. Цвіркун, С.М. Ткаченко, Я.В. Панферова, Д.О. Бешта, Л.В. Бешта. – Д.: НТУ «ДП», 2023-2024. – 62 с.
2. Технічний паспорт VMx280 [Електронний ресурс] – Режим доступу до ресурсу: <https://mouser.com/datasheet/2/783/BST-BME280-DS002-1509607.pdf> (дата звернення 19.04.2024)
3. Технічний паспорт АНТ10 [Електронний ресурс] – Режим доступу до ресурсу: <https://robu.in/wp-content/uploads/2021/01/АНТ10.pdf> (дата звернення 19.04.2024)
4. Технічний паспорт MHZ-19-B [Електронний ресурс] – Режим доступу до ресурсу: https://winsen-sensor.com/d/files/infrared-gas-sensor/mh-z19b-co2-ver1_0.pdf (дата звернення 19.04.2024)
5. Документація мікроконтролера ESP8266 [Електронний ресурс] – Режим доступу до ресурсу: <https://www.espressif.com/en/products/socs/esp8266> (дата звернення 21.04.2024)
6. Flask project [Електронний ресурс] – Режим доступу до ресурсу: <https://flask.palletsprojects.com/en/2.1.x/> (дата звернення 29.04.2024)
7. Технічні характеристики обладнання Cisco – Режим доступу: <https://stack-systems.com.ua/> (дата звернення 24.05.2024)
8. MQTT Essentials [Електронний ресурс] – Режим доступу до ресурсу: <https://www.hivemq.com/mqtt-essentials/> (дата звернення 22.04.2024)
9. Python Software Foundation [Електронний ресурс] – Режим доступу до ресурсу: <https://www.python.org/> (дата звернення 19.04.2024)

Додаток А.
Текст HTML коду веб сторінок

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
“ДНІПРОВСЬКА ПОЛІТЕХНІКА”

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ
НАЛАШТУВАННЯ ВЕБ ІНТЕРФЕЙСУ КІБЕРФІЗИЧНОЇ СИСТЕМИ

Текст програми

804.02070743.24005-01 12 01

Листів 22

АНОТАЦІЯ

Дана програма містить в собі програмний код веб сторінок інтерфейсу користувача для кіберфізичної системи моніторингу та контролю мікроклімату в розумних будинках.

Система призначена для збору, обробки та візуалізації даних з датчиків температури, вологості та CO₂, а також для віддаленого керування кліматичними пристроями.

Веб-інтерфейс створений з використанням HTML, CSS та JavaScript, забезпечуючи зручний та інтуїтивно зрозумілий спосіб взаємодії користувача з системою.

ЗМІСТ

1	Файл головної сторінки index.html	4
2	Файл сторінки авторизації login.html	11
3	Файл сторінок з ретельною інформацією sensor_graph.html.....	12
4	Файл сторінки налаштувань settings.html	16

Текст файлу index.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="icon" type="image/x-icon" href="{{ url_for('static',
filename='/favicon.ico') }}">
  <link rel="stylesheet" href="{{ url_for('static',
filename='styles/styles.css') }}">
  <link rel="stylesheet" href="styles/styles.css">
  <link rel="data" href="{{ url_for('static', filename='data/dataa.csv') }}">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.5.1/css/all.css">
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/framework7-
icons/css/framework7-icons.css">
  <script src="https://kit.fontawesome.com/7e84202197.js"
crossorigin="anonymous"></script>
  <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
  <script
src="https://cdn.jsdelivr.net/npm/papaparse@5.3.1/papaparse.min.js"></script>
  <title>Smart Home Monitor</title>
</head>
<body>
  <div id="notification"></div>
  <div class="bottom-menu">
    <a href="#" class="menu-item"><i class="fas fa-home fa-2x"></i></a>
    <a href="/overview" class="menu-item"><i class="fas fa-chart-bar fa-
2x"></i></a>
    <a href="/charts" class="menu-item"><i class="fas fa-tachometer-alt fa-
2x"></i></a>
    {% if not is_guest %}
    <a href="/settings" class="menu-item"><i class="fas fa-gear fa-
2x"></i></a>
    {% endif %}
    {% if hello_username == 'Guest' %}
    <a href="/login" action="/login" class="menu-item"><i class="fas fa-
sign-in-alt fa-2x"></i></a>
    {% else %}
    <a href="/logout" class="menu-item"><i class="fas fa-sign-out-alt fa-
2x"></i></a>
    {% endif %}
  </div>

```

```

</div>
<div class="side-menu">
  <div class="menu-header">Smart Home</div>
  <a href="#" class="menu-item menu-item-active"><i ></i>Home</a>
  <a href="/overview" class="menu-item"><i class="fas fa-chart-bar fa-
lg"></i>Addit</a>
  <a href="/charts" class="menu-item"><i class="fas fa-tachometer-alt fa-
lg"></i>Dash</a>
  {% if not is_guest %}
  <a href="/settings" class="menu-item"><i class="fas fa-gear fa-
lg"></i>Prefs</a>
  {% endif %}
  {% if hello_username == 'Guest' %}
  <a href="/login" action="/login" class="menu-item"><i class="fas fa-
sign-in-alt fa-lg"></i>Log In</a>
  {% else %}
  <a href="/logout" class="menu-item"><i class="fas fa-sign-out-alt fa-
lg"></i>Log Out</a>
  {% endif %}
</div>
<div class="content">
  <h3 class="heading">Hey, {{ hello_username }} 🙌 Yur device is <span
id="status" data-status="{{ status }}">{{ status }}</span></h3>
  <hr class="rounded">
  <div class="grid-container" id="agg">
  <div class="big-card">
    <div class="date-time-container">
      <p class="day">There</p>
      <p class="month">sould be</p>
      <p class="date">your date</p>
    </div>
    <p class="time">sad...</p>
  </div>
  <div class="big-card cozy-color">
    <div class="date-time-container">
      <p class="d">The weather</p>
      <p class="m">feels like</p>
      <p class="dd" id="temperature-description">...</p>
    </div>
    <div class="t" id="temperature-emoji"></div>
  </div>
</div>
<div class="grid">

```



```

    {% for sensor, data in sensor_data.items() %}
        <div class="card" onclick="goToSensorGraph('{{ sensor }}')">
            <i class="f7-icons icon-in-card">{{ data.icon }}</i>
            <h3 class="text">{{ sensor }}</h3>
            <p class="text">
                Value:
                <span id="{{ sensor }}Value">{{ data.value }}</span>
                {{ data.unit }}
            </p>
        </div>
    {% endfor %}
</div>
<script>
    function goToSensorGraph(sensorName) {
        window.location.href = "/sensor_graph/" + sensorName;
    }
</script>
    <div class="grid">
        <div class="big-card some-color">
            <h2 class="some-color">State: {{ state }}</h2>
            <h3 class="some-color">Smart rozetka</h3>
        </div>

        <div class="big-card">
            <div class="slider-container">
                <label for="switch">Toggle: On/Off</label>
                <input type="checkbox" id="switch" class="switch"
onchange="toggleSwitch(this)" {% if current_state == "ON" %}checked{% endif
%}>
            </div>
        </div>

    </div>
<div class="grid paddingg">
    <div class="big-card somer-color">
        <h2 class="somer-color">Set prefered temperature</h2>
        <div class="seekbar-container">
            <input type="range" min="0" max="100" value="{{ last_value }}"
class="seekbar" id="{{ sensor_name }}Seekbar">
            <div class="seekbar-value" id="{{ sensor }}SeekbarValue"></div>
        </div>
    </div>
</div>

```

```

<div class="big-card">
  <div class="slider-container">
    <label for="switch">Toggle: On/Off</label>
    <input type="checkbox" id="switch" class="switch"
onchange="toggleSwitch(this)" {% if current_state == "ON" %}checked{% endif
%}>
  </div>
</div>
</div>
</div>
</div>
  <script>
const checkbox = document.getElementById("switch");
function toggleSwitch(checkbox) { //Функція перемикання перемикача
  const state = checkbox.checked ? "on" : "off";
  fetch("/", { // Відправка стану переикача на '/'
    method: "POST",
    headers: { "Content-Type": "application/x-www-form-urlencoded" },
    body: `switch=${state}`
  })
  .then(response => {
    if (!response.ok) {
      throw new Error('Network response was not ok.');
```

```

    const monthNames = ["JAN", "FEB", "MAR", "APR", "MAY", "JUN", "JUL",
"AUG", "SEP", "OCT", "NOV", "DEC"];
    const dayElement = document.querySelector(".day");
    const dateElement = document.querySelector(".date");
    const monthElement = document.querySelector(".month");
    const timeElement = document.querySelector(".time");
    const now = new Date();
    const yea = { year: "numeric", };
    const mo = { month: "long",};
    const da = { day: "numeric",};
    const ti = { hour: "numeric",
                minute: "numeric",};
    dayElement.textContent = now.toLocaleDateString("en-US", da);
    dateElement.textContent = now.toLocaleDateString("en-US", yea);
    monthElement.textContent = now.toLocaleDateString("en-US", mo);
    timeElement.textContent = now.toLocaleTimeString("en-US", ti);
}
updateDateTime();
setInterval(updateDateTime, 1000);

var status = document.getElementById("status").getAttribute("data-
status");
function updateData() {
    if (status !== "Offline") {
        fetch('/data')
            .then(response => {
                if (response.ok) {
                    return response.json();
                } else {
                    throw new Error('Помилка отримки даних');
                }
            })
            .then(data => {
                Object.keys(data).forEach(sensor => {
                    console.log(`${sensor}Value`);
                    document.getElementById(`${sensor}Value`).textContent =
data[sensor].value;
                    console.log(data[sensor].value);
                });
                Object.keys(data).forEach(sensor => {
                    const element = document.getElementById(`${sensor}Value`);
                    if (element) {

```

```

        element.innerText = data[sensor].value;
    } else {
        console.error(`Element with id "${sensor}Value" not
found`);
    }
});

var settings = {
    "backgroundColor": "#333333",
    "chartColor": "#cdb3d5",
    "chartLinesColor": "#fff",
    "textColor": "#fff"
};

const checkbox = document.getElementById("switch");
checkbox.checked = data.current_state === "ON";
})

.catch(error => {
    console.error(error);
});
}
}

updateData();
setInterval(updateData, 5000);
async function checkDataAndAlert() {
    // Fetch
    const settingsResponse = await fetch("/static/data/settings.csv");
    const settingsText = await settingsResponse.text();
    const settingsData = Papa.parse(settingsText, { header: true }).data;
    // Extract
    const thresholds = {};
    for (const row of settingsData) {
        thresholds[row.sensor_name] = parseFloat(row.value);
    }
    // Fetch
    const response = await fetch("/static/data/dataa.csv");
    const text = await response.text();
    const data = Papa.parse(text, { header: true }).data;

    const latestData = data.reduce((acc, row) => {
        acc[row.sensor_name] = row.value;
        return acc;
    }, {});
    // Check alerts

```

```

    let alertMessage = "";
    for (const sensor in thresholds) {
        if (latestData[sensor] && parseFloat(latestData[sensor]) >
thresholds[sensor]) {
            alertMessage += `${sensor} is too high
(${latestData[sensor]})\n`;
        }
    }
    if (latestData.Temp_Inside && parseFloat(latestData.Temp_Inside) > 40)
{
        alertMessage += `Temperature is too high
(${latestData.Temp_Inside}°C)\n`;
    }
    if (latestData.Humidity && parseFloat(latestData.Humidity) >
humidityThreshold) {
        alertMessage += `Humidity is too high (${latestData.Humidity}%) \n`;
    }
    if (latestData.CO2 && parseFloat(latestData.CO2) > ppmThreshold) {
        alertMessage += `CO2 level is too high (${latestData.CO2} PPM)\n`;
    }
    if (alertMessage) {
        alert(alertMessage);
        // fetch("/clear_data", { method: "POST" })
        //     .then(response => { /* ... */ });
    }
}
checkDataAndAlert();
setInterval(checkDataAndAlert, 60000);
const seekbar = document.getElementById(`${ sensor }SeekBar");
const seekbarValue = document.getElementById(`${ sensor }SeekBarValue");
seekbar.addEventListener('input', function() {
    const value = this.value;
    seekbarValue.textContent = value;
    const percent = (value / this.max) * 100;
    const seekbarRect = seekbar.getBoundingClientRect();
    const offset = (percent / 100) * seekbarRect.width -
seekbarValue.offsetWidth / 2;
    seekbarValue.style.left = `${offset}px`;
});
</script>
</body>
</html>

```

Текст файлу login.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="icon" type="image/x-icon" href="{{ url_for('static',
filename='/favicon.ico') }}">
  <link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='styles/auth.css') }}">
  <title>Login</title>
</head>
<body>
  <div class="loader-container">
    <div class="loader"></div>
  </div>
  <div class="container">
    <form action="/login" method="post">
      <div class="card">
        <a class="login">Log in</a>
        <div class="inputBox">
          <input type="text" name="username" required="required">
          <span class="user">Username</span>
        </div>
        <div class="inputBox">
          <input type="password" name="password"
required="required">
          <span>Password</span>
        </div>
        <button type="submit" class="enter">Enter</button>
        <a href="{{ url_for('index') }}"?guest=true"
class="ag">Guest</a>
        <a href="{{ url_for('register') }}" class="register-
link">Register</a>
      </div>
    </form>
    <a class="error">

```

```

        {% if error_message %}
            <p>{{ error_message }}</p>
        {% endif %}
    </a>
</div>
</body>
</html>

```

Текст файлу sensor_graph.html

```

<!DOCTYPE html>
<html>
<head>
    <title>{{ sensor_name }} Graph</title>
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/framework7-
icons/css/framework7-icons.css">
    <link rel="stylesheet" href="{{ url_for('static',
filename='styles/graph_styles.css') }}">
    <link rel="icon" type="image/x-icon" href="{{ url_for('static',
filename='/favicon.ico') }}">
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
    <script
src="https://cdn.jsdelivr.net/npm/papaparse@5.3.2/papaparse.min.js"></script>
    <script
src="http://cdnjs.cloudflare.com/ajax/libs/hammer.js/2.0.8/hammer.min.js"></s
cript>
</head>
<body>
    <div class="side-menu">
    <div class="button-container">
        <button class="back-button" onclick="window.location.href='/'">
            <i class="f7-icons">arrowtriangle_up_fill</i>Back to
Home</button>
        <button class="next-button" onclick="goToPreviousSensor('{{
sensor_name }}')">
            Prev Sensor<i class="f7-
icons">arrowtriangle_left_fill</i></button>
        <button class="next-button" onclick="goToNextSensor('{{ sensor_name
}}')">
            <i class="f7-icons">arrowtriangle_right_fill</i>Next
Sensor</button>
        <button class="next-button" onclick="goToNextSensor('{{ sensor_name
}}')">

```

```

        <i class="f7-icons">grid</i>Export to table</button>
    </div>
</div>
<div class="content">
    <div class="over-card temp-color">
        <div class="card-header">
            <h2>{{ sensor_name }}</h2>
            <div id="averageValue" class="smol"></div>
            <div id="minValue" class="smol"></div>
            <div id="maxValue" class="smol"></div>
        </div>
        <div class="card-body">
            <div class="chart-container">
                <canvas id="{{ sensor_name }}Canvas"></canvas>
            </div>
        </div>
        <div class="seekbar-container">
            <input type="range" min="0" max="100" value="100" class="seekbar"
id="{{ sensor_name }}Seekbar">
        </div>
        <div class="visible-points-container">
            <label for="visiblePointsInput">Visible Points:</label>
            <input class="input" type="number" id="visiblePointsInput"
min="1" value="20">
        </div>
    </div>
</div>
    <script>
    async function getData(sensorName) {
    const response = await fetch("/static/data/dataa.csv");
    const text = await response.text();
    const data = Papa.parse(text, { header: true }).data;
    const filteredData = data.filter(row => row.sensor_name === sensorName);
    const datasets = [{
    label: sensorName,
    fill: true,
    data: filteredData.map(row => ({
    x: new Date(row.timestamp),
    y: parseFloat(row.value)
    })),
    borderColor: "#2F81F7",
    backgroundColor: "rgba(47,129,247,0.3)",

```



```

borderWidth: 2,
pointRadius: 2,
tension: 0.2,
}]);
let visiblePoints = 20;
const ctx = document.getElementById(`${sensorName}Canvas`).getContext("2d");
const averageValueElement = document.getElementById("averageValue");
const maxValueElement = document.getElementById("maxValue");
const minValueElement = document.getElementById("minValue");
function createChart(startIndex) {
const endIndex = Math.min(startIndex + visiblePoints,
datasets[0].data.length);
const visibleValues = datasets[0].data.slice(startIndex, endIndex).map(d =>
d.y);
const average = visibleValues.reduce((sum, value) => sum + value, 0) /
visibleValues.length;
averageValueElement.textContent = `Average: ${average.toFixed(2)}`;
const max = visibleValues.reduce((maxVal, currentVal) => currentVal > maxVal
? currentVal : maxVal, -Infinity);
maxValueElement.textContent = `Maximum: ${max.toFixed(2)}`;
const min = visibleValues.reduce((minVal, currentVal) => currentVal < minVal
? currentVal : minVal, Infinity);
minValueElement.textContent = `Minimum: ${min.toFixed(2)}`;
const chart = new Chart(ctx, {
type: "line",
data: {
datasets: datasets,
},
options: {
responsive: true,
maintainAspectRatio: false,
pan: {
enabled: true,
mode: 'x',
},
plugins: {
legend: {
display: true,
position: "top",
},
},
scales: {

```

```

x: {
  type: 'linear',
  min: datasets[0].data[startIndex].x.getTime(),
  max: datasets[0].data[endIndex - 1].x.getTime(),
  ticks: {
    autoSkip: true,
    maxTicksLimit: 10,
    callback: function(value, index, values) {
      return new Date(value).toLocaleTimeString([], { hour: '2-digit', minute: '2-
digit' });
    }
  },
  title: {
    display: true,
    text: 'Time'
  }
},
y: {
  title: {
    display: true,
    text: sensorName
  },
  suggestedMin: Math.min(...datasets[0].data.map(d => d.y)) - 2,
  suggestedMax: Math.max(...datasets[0].data.map(d => d.y)) + 2,
}
},
pan: {
  enabled: true,
  mode: 'x',
  onPan: function({chart}) {
    chart.update('none');
  }
},
});
return chart;
}

let chart = createChart(datasets[0].data.length - visiblePoints);

const seekbar = document.getElementById(`${ sensor_name }Seekbar`);
seekbar.addEventListener('input', function() {
const totalPoints = datasets[0].data.length;

```

```

const newStartIndex = Math.round((totalPoints - visiblePoints) * (this.value
/ 100));
chart.destroy();
chart = createChart(newStartIndex);
});
const visiblePointsInput = document.getElementById("visiblePointsInput");
visiblePointsInput.addEventListener('input', function() {
visiblePoints = parseInt(this.value, 10) || 20;
chart.destroy();
chart = createChart(datasets[0].data.length - visiblePoints);
});
}

getData('{{ sensor_name }}');

function goToNextSensor(currentSensor) {
const sensorOrder = ["Temp_Inside", "Temp Outside", "Pressure",
"CO2", "Humidity_Inside", "Humidity Outside", "CO2"];
const currentIndex = sensorOrder.indexOf(currentSensor);
const nextIndex = (currentIndex + 1) % sensorOrder.length;
const nextSensor = sensorOrder[nextIndex];
window.location.href = "/sensor_graph/" + nextSensor;
}

function goToPreviousSensor(currentSensor) {
const sensorOrder = ["Temp_Inside", "Temp Outside", "Pressure",
"CO2", "Humidity_Inside", "Humidity Outside", "CO2"];
const currentIndex = sensorOrder.indexOf(currentSensor);
const prevIndex = (currentIndex - 1 + sensorOrder.length) %
sensorOrder.length;
const prevSensor = sensorOrder[prevIndex];
window.location.href = "/sensor_graph/" + prevSensor;
}
</script>
</body>
</html>

```

Текст файла settings.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

```

```

<link rel="icon" type="image/x-icon" href="{{ url_for('static',
filename='/favicon.ico') }}">
<link rel="stylesheet" href="{{ url_for('static',
filename='styles/styles.css') }}">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.5.1/css/all.css">
<script src="https://kit.fontawesome.com/7e84202197.js"
crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/framework7-
icons/css/framework7-icons.css">
<link rel="stylesheet" href="{{ url_for('static',
filename='styles/styles.css') }}">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.5.1/css/all.css">
<title>Settings</title>
</head>
<body>
<div class="bottom-menu">
<a href="#" class="menu-item"><i class="fas fa-home fa-2x"></i></a>
<a href="/overview" class="menu-item"><i class="fas fa-chart-bar fa-
2x"></i></a>
<a href="/charts" class="menu-item"><i class="fas fa-tachometer-alt fa-
2x"></i></a>
<a href="/login" action="/login" class="menu-item"><i class="fas fa-sign-
out-alt fa-2x"></i></a>
</div>

<div class="side-menu">
<div class="menu-header">Smart Home</div>
<a href="/" class="menu-item"><i class="fas fa-home fa-lg"></i>Home</a>
<a href="/overview" class="menu-item"><i class="fas fa-chart-bar fa-
lg"></i>Addit</a>
<a href="/charts" class="menu-item"><i class="fas fa-tachometer-alt fa-
lg"></i>Dash</a>
{% if not is_guest %}
<a href="#" class="menu-item menu-item-active"><i ></i>Prefs</a>
{% endif %}
{% if hello_username == 'Guest' %}
<a href="/login" action="/login" class="menu-item"><i class="fas fa-
sign-in-alt fa-lg"></i>Log In</a>
{% else %}

```

```

        <a href="/logout" class="menu-item"><i class="fas fa-sign-out-alt fa-
lg"></i>Log Out</a>
        {% endif %}
    </div>
    <div class="content">
    <div></div>
<div></div>
    <button class="back-button"
onclick="window.location.href='/server_info'">
    <i class="f7-icons">desktopcomputer</i>Server stats</button>
    <button id="clearDataButton"><i class="f7-
icons">arrow_counterclockwise_circle_fill</i>Clear All Data</button>
    <form action="/settings" method="post">
        <div class="settingsElement">
            <a class="title">MQTT Broker Settings</a>
            <div class="inputBox">
                <input type="text" name="mqtt_broker" placeholder="127.0.0.1"
class="input" value="{{ mqtt_broker }}">
                <span class="label">MQTT Broker</span>
            </div>
            <div class="inputBox">
                <input type="text" name="mqtt_port" placeholder="1883"
class="input" value="{{ mqtt_port }}">
                <span class="label">MQTT Port</span>
            </div>
            <div class="inputBox">
                <input type="text" name="mqtt_username" placeholder="user"
class="input" value="{{ mqtt_user }}">
                <span class="label">MQTT Username</span>
            </div>
            <div class="inputBox">
                <input type="password" name="mqtt_passwd" placeholder="password"
class="input" value="{{ mqtt_passwd }}">
                <span class="label">MQTT Password</span>
            </div>
        </div>
        <div class="settingsElement">
        </div>
    <div class="settingsElement">
        <h2>Threshold Settings</h2>
        <div class="inputBox">

```

```

        <input type="number" name="temperature_threshold"
placeholder="Temperature Threshold" class="input" value="{{
thresholds.get('Temperature', '') }}">
        <span class="label">Temperature (°C)</span>
</div>
<div class="inputBox">
        <input type="number" name="humidity_threshold"
placeholder="Humidity Threshold" class="input" value="{{
thresholds.get('Humidity', '') }}">
        <span class="label">Humidity (%)</span>
</div>
<div class="inputBox">
        <input type="number" name="co2_threshold" placeholder="CO2
Threshold" class="input" value="{{ thresholds.get('CO2', '') }}">
        <span class="label">CO2 (PPM)</span>
</div>

<h2>Add new sensor</h2>
<form method="POST">
        <input type="text" name="new_key" id="new_key" placeholder="New
sensor name" oninput="updatePreview()" class="input">
        <button type="submit" name="add">Add Sensor</button>
</form>

<form method="POST">
        <select name="del_key" class="input">
                {% for key in sensor_data.keys() %}
                        <option value="{{ key }}">{{ key }}</option>
                {% endfor %}
        </select>
        <button type="submit" name="delete">Delete Sensor</button>
<h2>Added sensors</h2>
<form method="POST">
        <table>
                {% for key, value in sensor_data.items() %}
                        <tr>
                                <td>{{ key }}:</td>
                                <td><input type="number" name="{{ key }}_value"
value="{{ value['value'] }}" class="input" style="display: none;"></td>
                                <td><input type="text" name="{{ key }}_unit"
value="{{ value['unit'] }}" class="input"></td>

```

```

                <td><input type="text" name="{{ key }}_icon"
value="{{ value['icon'] }}"class="inputtt"></td>
            </tr>
        {% endfor %}
    <tr>
        <td><input type="text" id="new_key" name="new_key"
placeholder="New sensor name" oninput="updatePreview()"class="inputtt"></td>
        <td><input type="number" id="new_value"
placeholder="Value" oninput="updatePreview()"class="inputtt"style="display:
none;"></td>
        <td><input type="text" id="new_unit" placeholder="Unit"
oninput="updatePreview()"class="inputtt"></td>
        <td><input type="text" id="new_icon" placeholder="Icon"
oninput="updatePreview()"class="inputtt"></td>
    </tr>
</table>
<button type="submit">Submit new sensors</button>
</form>
<h2>Preview of New Card:</h2>
<div class="grid" style="margin-right:90%;">
    <div class="card">
        <i class="f7-icons icon-in-card" id="previewIcon"></i>
        <h3 class="text" id="previewSensorName"></h3>
        <p class="text">
            Value: <span id="previewValue"></span> <span
id="previewUnit"></span>
        </p>
    </div>
</div>
</body>
</div>
<script>
function updatePreview() {
    var newSensorName = document.getElementById("new_key").value;
    var newValue = document.getElementById("new_value").value;
    var newUnit = document.getElementById("new_unit").value;
    var newIcon = document.getElementById("new_icon").value;

    document.getElementById("previewSensorName").textContent = newSensorName;
    document.getElementById("previewValue").textContent = newValue;
    document.getElementById("previewUnit").textContent = newUnit;
    document.getElementById("previewIcon").textContent = newIcon;
}

```

```

}
function addTopic() {
  const newTopic = prompt("Enter the new topic name:");
  if (newTopic) {
    const newDiv = document.createElement("div");
    newDiv.className = "inputBox";
    newDiv.innerHTML = `
      <input type="text" name="topic_${newTopic}_value"
placeholder="Value" class="input">
      <input type="text" name="topic_${newTopic}_unit"
placeholder="Unit" class="input">
      <input type="text" name="topic_${newTopic}_icon"
placeholder="Icon" class="input">
      <span class="label">${newTopic}</span>`;
    document.querySelector('.settingsElement').appendChild(newDiv);
  }
}
const clearDataButton = document.getElementById("clearDataButton");
clearDataButton.addEventListener("click", () => {
  if (confirm("Are you sure you want to clear all data? This action
cannot be undone.)) {
    fetch("/clear_data", { method: "POST" })
      .then(response => {
        if (response.ok) {
          alert("Data cleared successfully!");
          location.reload();
        } else {
          alert("Error clearing data.");
        }
      })
  });
});
</script>
</script>
</body>
</html>

```


Додаток Б.**Текст програми налаштування мережевого обладнання**

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
“ДНІПРОВСЬКА ПОЛІТЕХНІКА”

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ
НАЛАШТУВАННЯ МЕРЕЖІ КОМП'ЮТЕРНОЇ СИСТЕМИ

Текст програми налаштування мережного обладнання.

804.02070743.24005-01 12 01

Листів 2

АНОТАЦІЯ

Ця програма містить комплексний набір команд, спеціально розроблених для налаштування маршрутизаторів та комутаторів у корпоративній мережі. Вона охоплює всі аспекти конфігурації, починаючи від базового призначення IP-адрес до детального налаштування ключових мережевих сервісів та протоколів.

У програмі передбачено конфігурацію DHCP для автоматизації процесу розподілу IP-адрес, NAT для безпечного доступу до інтернету. Крім того, вона включає налаштування протоколу OSPF для динамічної маршрутизації в складних мережевих середовищах.

ЗМІСТ

1 Конфігурація Holubitskiy _R4 26

1. Конфігурація Holubitskiy_R4

```
Current configuration : 1461 bytes
version 15.1
no service timestamps log datetime msec
no service timestamps debug datetime msec
service password-encryption
!
hostname Holubitsky_R4
!
enable secret 5 $1$mERr$9cTjUIEqNGurQiFU.ZeCi1
!
no ip cef
no ipv6 cef
!
username Holubitsky password 7 082048430017061E010803
!
license udi pid CISCO2911/K9 sn FTX15243TH8-
!
ip ssh version 1
ip domain-name Holubitsky_4
!
spanning-tree mode pvst
!
interface GigabitEthernet0/0
no ip address
duplex auto
speed auto
shutdown
!
interface GigabitEthernet0/1
no ip address
duplex auto
speed auto
shutdown
!
interface GigabitEthernet0/2
no ip address
duplex auto
speed auto
shutdown
!
interface Serial0/2/0
ip address 172.24.85.14 255.255.255.252
!
interface Serial0/2/1
no ip address
clock rate 2000000
shutdown
!
interface Serial0/3/0
ip address 172.24.85.6 255.255.255.252
```

```
clock rate 2000000
interface Serial0/3/1
ip address 172.24.85.9 255.255.255.252
interface Vlan1
no ip address
shutdown
router ospf 1
log-adjacency-changes
network 172.24.85.8 0.0.0.3 area 0
network 172.24.85.4 0.0.0.3 area 0
network 172.24.80.12 0.0.0.3 area 0
ip classless
ip flow-export version 9
banner motd ^CHolubitsky_R4^C
line con 0
password 7 0822455D0A16
login
line aux 0
line vty 0 4
password 7 0822455D0A16
login local
transport input ssh
line vty 5 15
password 7 0822455D0A16
login local
transport input ssh
```