

Міністерство освіти і науки України
 Національний технічний університет
 «Дніпровська політехніка»
 Інститут електроенергетики
 (інститут)
 Факультет інформаційних технологій
 (факультет)
 Кафедра інформаційних технологій та комп'ютерної інженерії
 (повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня бакалавра

студента Груцин Євгеній Сергійович
 (П.І.Б.)

академічної групи 123-20ск-1
 (шифр)

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

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

на тему Комп'ютерна система моніторингу рівня кисню в приміщеннях
 ТОВ Дніпровський трубний завод з детальним опрацюванням побудови,
 налаштування та безпеки корпоративної мережі
 (назва за наказом ректора)

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
кваліфікаційної роботи	доц. Каштан В.Ю.			
розділів:				
розробка апаратної частини	доц. Ткаченко С.М.			
розробка корпоративної мережі	ас. Бешта Л.В.			

Рецензент				
------------------	--	--	--	--

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

Дніпро
2023

ЗАТВЕРДЖЕНО:
завідувач кафедри
інформаційних технологій
та комп'ютерної інженерії
(повна назва)

_____ Гнатушенко В.В.
(підпис) (прізвище, ініціали)

" ___ " _____ 2023 року.

ЗАВДАННЯ

на кваліфікаційну роботу ступеня бакалавр

студента Груцин Є.С. академічної групи 123-20ск-1
(прізвище, ініціали) (шифр)

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

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

на тему Комп'ютерна система моніторингу рівня кисню в приміщеннях
ТОВ Дніпровський трубний завод з детальним опрацюванням побудови,
налаштування та безпеки корпоративної мережі
(назва за наказом ректора)

затверджена наказом ректора НТУ «Дніпровська політехніка» від 16.05.20223 № № 350-с

Розділ	Зміст завдання	Термін виконання
Стан питання та постановка завдання	На основі матеріалів виробничих практик, інших науково-технічних джерел конкретизується предмет та мету роботи та виконується постановка завдання	01.05.2023
Розробка апаратної частини	На основі аналізу підприємства сформулюються технічні вимоги до комп'ютерної системи та розробляється апаратна частина системи	29.05.2023
Розробка корпоративної мережі	Виконується розрахунок налаштувань корпоративної мережі та перевірка роботи системи, розробляються методи та налаштування обладнання для захисту інформації в системі	12.06.2023
Розробка компонентасистеми	Виконується детальна розробка компонента системи	30.06.2023

Завдання видано

_____ (підпис керівника)

доц. Каштан В.Ю.
(прізвище та ініціали)

Дата видачі

01.04.2023 р.

Дата подання до атестаційної комісії

13.07.2023 р.

Прийнято до виконання

_____ (підпис студента)

Груцин Є.С.
(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка: 83 с., 32 рис., 4 табл., 2 дод., 10 джерел.

СИСТЕМА МОНІТОРИНГУ КИСНЕМ, ДАТЧИКИ, СОНЯЧНА ЕНЕРГІЯ, РІВЕНЬ ДОСТУПУ

Об'єкт розробки представляє собою систему, що включає датчики кисню, мережеве обладнання, IoT-пристрої та серверне програмне забезпечення, які взаємодіють між собою для забезпечення моніторингу рівня кисню і прийняття відповідних заходів за необхідності.

Предмет розробки: комп'ютерна система моніторингу рівня кисню в приміщеннях ТОВ Дніпровський трубний завод.

Мета дослідження: розробка, налаштування та безпека корпоративної мережі для моніторингу рівня кисню в приміщеннях заводу, що сприятиме забезпеченню безпеки працівників та покращенню умов праці.

У рамках роботи було розроблено комп'ютерну систему моніторингу, яка включає в себе датчики кисню, мережеве обладнання, IoT-пристрої та серверне програмне забезпечення. Було проведено побудову та налаштування корпоративної мережі, забезпечено безпеку мережі та даних.

Система моніторингу дозволяє операторам контролювати рівень кисню в реальному часі та при необхідності приймати відповідні заходи. Це сприяє покращенню безпеки працівників та забезпечує належні умови праці в приміщеннях заводу.

Результати дослідження підтверджують успішну реалізацію комп'ютерної системи моніторингу та підкреслюють важливість впровадження подібних систем у промислових середовищах для забезпечення безпеки та комфорту працівників..

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	6
ВСТУП	7
1 СТАН ПИТАННЯ І ПОСТАНОВКА ЗАВДАННЯ.....	9
1.2 Характеристика і структура ТОВ «Дніпровський трубний завод».....	10
1.3 Стислі відомості про технології збору та передачі інформації для ТОВ «Дніпровського трубного заводу».....	12
1.4 Принципи, технічні способи інформаційного забезпечення для ТОВ «Дніпровського трубного заводу».....	17
1.5 Аналітичний огляд існуючих способів обробки та передачі інформації, принципів побудови об'єкта проектування, відомих рішень у галузі.....	18
1.6 Завдання і мета роботи.....	19
1.7 Основні особливості та проблематика впровадження системи моніторингу рівня кисню	21
2 РОЗРОБКА АПАРАТНОЇ ЧАСТИНИ КОМП'ЮТЕРНОЇ СИСТЕМИ.....	23
2.1 Технічні вимоги до комп'ютерної системи.....	23
2.2 Вимоги до функцій, які виконує КС	24
2.3 Чисельність та склад персоналу	25
2.4 Вимоги до технічного забезпечення системи	26
2.5 Вимоги до інформаційного забезпечення.....	28
2.6 Розробка технічних вимог до апаратного забезпечення комп'ютерної системи...29	
2.7 Розробка загальної структури комп'ютерної системи ТОВ «Дніпровський трубний завод»	30
3 РОЗРОБКА КОРПОРАТИВНОЇ МЕРЕЖІ.....	32
3.1 Розрахунок IP-адрес та масок підмереж для ефективною адресації корпоративної мережі.....	32
3.2 Розрахунок схеми адресації пристроїв	34
3.3 Розробка схеми логічної топології корпоративної мережі	36

	5
3.4 Базове налаштування конфігурації пристроїв КС	37
3.4.1 Налаштування маршрутизаторів КС ТОВ «Дніпровський трубний завод»	39
3.4.2 Налаштування служби AAA на маршрутизаторах	41
3.4.3 Налаштування роботи Інтернету в КС ТОВ «Дніпровський трубний завод»	43
3.4.3 Налаштування VPN в КС ТОВ «Дніпровський трубний завод»	45
3.4.3 Налаштування агрегування каналів в відділі якості КС ТОВ «Дніпровський трубний завод»	46
3.4.4 Налаштування динамічного розподілу адрес.....	48
3.5 Налаштування VLAN в під мережі «Відділ закупівель»	50
3.6 Перевірка роботи комп'ютерної системи.....	51
4 РОЗРОБКА КОМПОНЕНТА СИСТЕМИ МОНІТОРИНГУ РІВНЯ КИСНЮ ТОВ ДНІПРОВСЬКИЙ ТРУБНИЙ ЗАВОД.....	53
4.1 Конфігурація апаратного забезпечення та налаштування сервісів у системі Інтернету речей	53
4.2 Конфігурація апаратного забезпечення та налаштування сервісів у системі Інтернету речей	56
4.3 Налаштування функціонування датчиків Інтернету речей	59
4.4 Тестування системи Інтернету речей ТОВ «Дніпровський трубний завод»....	67
ВИСНОВКИ.....	71
ПЕРЕЛІК ПОСИЛАНЬ.....	73

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ І ТЕРМІНІВ**

ТОВ	– Товариство з обмеженою відповідальністю
ІоТ	– Інтернет речей
MES	– Автоматизовані системи управління виробництвом.
SCADA	– Системи автоматичного збору даних.
ІоТ	– Системи інтернету речей.
QMS	– Системи управління якістю.
EIGRP	– Enhanced Interior Gateway Routing Protocol

ВСТУП

Комп'ютерні системи моніторингу стали необхідною складовою сучасного промислового середовища. Вони забезпечують ефективний контроль і керування важливими параметрами в різних галузях промисловості. Однією з таких галузей є ТОВ "Дніпровський трубний завод", який потребує системи моніторингу рівня кисню в приміщеннях для забезпечення безпечних умов праці та оптимального функціонування.

Кваліфікаційна робота бакалавра спрямована на детальне опрацювання побудови, налаштування та забезпечення безпеки корпоративної мережі для комп'ютерної системи моніторингу рівня кисню на ТОВ "Дніпровський трубний завод". Основною метою є забезпечення надійного та ефективного збирання, обробки та передачі даних про рівень кисню, а також забезпечення безпеки цієї інформації.

В рамках цієї роботи будуть розглянуті ключові аспекти, такі як вибір та налаштування апаратних засобів, розгортання мережевої інфраструктури, розробка програмного забезпечення для збирання та аналізу даних, а також заходи безпеки для захисту системи від несанкціонованого доступу та витоку конфіденційної інформації.

Запропоновано використати Інтернет речей (IoT) датчики, що дозволять забезпечити збір даних про рівень кисню в режимі реального часу. IoT датчики, сполучені з мережевим обладнанням, дозволяють отримати точну та актуальну інформацію про рівень кисню в приміщеннях. Для моделювання та віртуалізації цього середовища використовується Cisco Packet Tracer - популярний інструмент для створення, налаштування та симуляції мережевих середовищ.

Результати цієї роботи будуть сприяти ефективному контролю рівня кисню в приміщеннях ТОВ "Дніпровський трубний завод", що в свою чергу позитивно вплине на безпеку працівників, процеси виробництва та якість продукції. Крім того, цей дослід допоможе розширити розуміння процесів моніторингу та безпеки

у промислових середовищах, а також продемонструє можливості використання Інтернету речей для забезпечення ефективного моніторингу та контролю параметрів у промислових системах.

1 СТАН ПИТАННЯ І ПОСТАНОВКА ЗАВДАННЯ

1.1 Стисла характеристика галузі та умов застосування комп'ютерної системи ТОВ «Дніпровський трубний завод»

ТОВ "Дніпровський трубний завод" є підприємством, що займається виробництвом і постачанням труб із сталі. Галузь, в якій функціонує Дніпровський трубний завод, є важливою складовою інфраструктурних та будівельних проектів. Труби, виготовлені заводом, використовуються в таких сферах, як нафтогазова промисловість, будівництво трубопроводів, водопостачання та каналізація, електроенергетика, машинобудування і багато інших [1].

Комп'ютерна система в ТОВ "Дніпровський трубний завод" використовується для оптимізації та покращення багатьох процесів у виробництві. Основні умови застосування комп'ютерної системи на заводі включають [1]:

- управління виробництвом. Комп'ютерна система дозволяє відстежувати та керувати усіма етапами виробництва труб, починаючи від планування і закінчуючи контролем якості. Вона допомагає оптимізувати розклад виробництва, розподіл ресурсів та контроль за виконанням завдань;

- управління складом і логістикою. Комп'ютерна система допомагає контролювати запаси готової продукції, відстежувати рух сировини та матеріалів, а також планувати та оптимізувати логістичні процеси, пов'язані з постачанням і доставкою труб;

- контроль якості. Комп'ютерна система дозволяє автоматизувати процес контролю якості труб. Вона допомагає відстежувати параметри якості на кожному етапі виробництва, проводити випробування та аналізувати дані для забезпечення високої якості кінцевої продукції;

- управління даними. Комп'ютерна система забезпечує збір, збереження і аналіз даних про виробництво та якість труб. Це дозволяє здійснювати аналітику

та приймати рішення на основі достовірної інформації;

– автоматизація процесів. Комп'ютерна система дозволяє автоматизувати багато рутинних операцій і процесів на заводі, що сприяє зниженню витрат часу і зусиль, а також підвищенню ефективності виробництва.

Застосування комп'ютерної системи в ТОВ "Дніпровський трубний завод" допомагає покращити продуктивність, ефективність і якість виробництва, забезпечуючи конкурентні переваги на ринку трубопровідної промисловості.

1.2 Характеристика і структура ТОВ «Дніпровський трубний завод»

Трубний завод є виробничим об'єктом, спеціалізованим на виготовленні та постачанні труб з різних матеріалів, зазвичай зі сталі. Впровадження трубного заводу передбачає створення комплексу виробничих приміщень, обладнання та інфраструктури, необхідних для здійснення процесу виробництва труб. Основними етапами впровадження трубного заводу є проектування, будівництво, налагодження обладнання та пусконаладжувальні роботи.

Структура об'єкта впровадження трубного заводу представлена на рис.1.1. розглянемо детально структуру.

Виробничі приміщення. Трубний завод має ряд виробничих приміщень, що включають цехи для обробки сировини, гарячої та холодної прокатки, волочіння, зварювання, іспитів, обробки та обрізки труб. Кожен цех виконує свої функції і має відповідне обладнання та системи керування.

Обладнання. Трубний завод оснащений різноманітним обладнанням для виготовлення та обробки труб, включаючи преси, прокатні валки, машини для зварювання, верстати для обробки кінців труб, установки для іспитів та вимірювань, системи очищення та обробки поверхонь і багато іншого.

Інфраструктура. До структури об'єкта впровадження трубного заводу входить інфраструктура, яка забезпечує його функціонування. Це включає системи енергопостачання, водопостачання та каналізації, вентиляції та кондиціонування повітря, системи безпеки та протипожежного захисту, системи транспортування

матеріалів та готової продукції [1].



Рисунок 1.1 – Структурна схема Дніпровського трубного заводу

Лабораторії та контрольно-вимірювальні пункти. Трубний завод має спеціалізовані лабораторії та контрольно-вимірювальні пункти, де проводяться випробування та аналіз якості труб. Це дозволяє перевірити відповідність труб вимогам стандартів і забезпечити високу якість продукції.

Адміністративні та управлінські приміщення. Трубний завод також має адміністративні та управлінські приміщення, де розташовані офіси управління, відділи планування виробництва, продажів, бухгалтерії, контролю якості, дослідно-конструкторські підрозділи та інші служби, які забезпечують ефективну роботу заводу.

Ця структура об'єкта впровадження трубного заводу забезпечує організацію

та оптимальний хід процесу виробництва труб, а також забезпечує високий рівень якості продукції та ефективно управління заводом.

1.3 Стислі відомості про технології збору та передачі інформації для ТОВ «Дніпровського трубного заводу»

Технології збору та передачі інформації в Дніпровському трубному заводі включають в себе різні методи та системи, що допомагають ефективно управляти процесами виробництва, контролювати якість продукції та забезпечувати безперебійну комунікацію між різними підрозділами заводу. Розглянемо системи, які використовуються на заводі.

Автоматизовані системи управління виробництвом (MES) використовуються для планування, контролю та управління всіма етапами виробництва. Вони збирають дані з різних джерел, таких як машини, обладнання та сенсори, і передають цю інформацію в центральну систему керування. На основі цих даних можуть бути розроблені розклади виробництва, виконані контрольні операції та здійснено моніторинг параметрів якості.

Системи автоматичного збору даних (SCADA) використовуються для збору та відображення даних з різних датчиків, контролерів та інших пристроїв у режимі реального часу. Вони дозволяють операторам спостерігати стан обладнання, контролювати процеси та здійснювати віддалене керування. Наприклад, SCADA система може моніторити параметри температури, тиску, швидкості виробництва та автоматично регулювати режими роботи для досягнення оптимальних результатів [1].

Системи інтернету речей (IoT) дозволяють підключати різні пристрої та обладнання до мережі Інтернет для обміну даними і взаємодії. На трубному заводі це можуть бути датчики, які вимірюють параметри, наприклад, температури, вологості або вібрації, та передають ці дані в хмарні системи аналітики або внутрішні бази даних для подальшого аналізу та контролю [2].

Системи управління якістю (QMS) використовуються для керування процесами контролю якості на трубному заводі. Вони забезпечують збір і аналіз

даних про якість продукції, проведення іспитів та випробувань, документування результатів та впровадження заходів для поліпшення якості. Ці системи можуть бути інтегровані з іншими технологіями для автоматизації збору даних та забезпечення їх точності.

Електронна пошта та внутрішній портал: Електронна пошта та внутрішній портал використовуються для обміну інформацією та документами між різними підрозділами та співробітниками трубного заводу. Це забезпечує швидку і ефективну комунікацію, обмін даними та спільну роботу над проектами.

Відеоконференції та віддалений доступ. Технології відеоконференцій дозволяють здійснювати віртуальні зустрічі та наради між співробітниками, навіть якщо вони знаходяться в різних місцях. Віддалений доступ до систем та даних заводу дозволяє співробітникам працювати з будь-якого місця та забезпечує надійну комунікацію з віддаленими робітниками або підрозділами.

Трубний завод використовує системи моніторингу та аналізу даних для збору, обробки та аналізу інформації про процеси виробництва та якість продукції. Це дозволяє виявляти аномалії, прогнозувати проблеми та приймати рішення на основі фактичних даних.

Мобільні додатки та платформи дозволяють співробітникам трубного заводу отримувати доступ до інформації, контролювати процеси та здійснювати роботу на виробництві з використанням мобільних пристроїв. Наприклад, через мобільні додатки можна контролювати стан обладнання, переглядати розклади виробництва або відстежувати хід виконання завдань.

З огляду на важливість даних та інформації, трубний завод використовує технології та системи для забезпечення безпеки та захисту даних. Це включає системи шифрування, автентифікації, резервного копіювання та фізичного захисту пристроїв та інфраструктури.

Вищеописані технології збору та передачі інформації в Дніпровському трубному заводі допомагають оптимізувати процеси виробництва, забезпечити контроль якості та ефективну комунікацію між різними підрозділами. Вони сприяють підвищенню продуктивності, зниженню витрат та покращенню якості

продукції.

Фізичне розташування трубного заводу включає велику територію з виробничими приміщеннями, складськими просторами, адміністративними будівлями та інфраструктурою. Завод може мати декілька основних зон або цехів для виготовлення, обробки та контролю якості труб (рис.1.2).

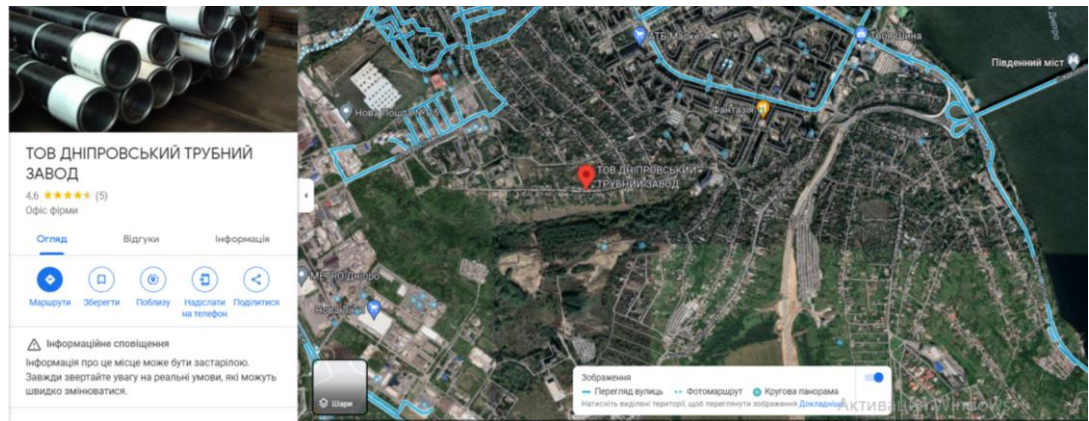


Рисунок 1.2 – Фізичне розташування заводу на Гугл карті [1]

Окрім основного заводу, Дніпровський трубний має окремий відділений офіс у іншій локації. Це офісний приміщення, де розташовані відділи управління, адміністрації, фінансів, продажів та інших служб, які забезпечують адміністративні та організаційні функції заводу. Офіс може бути розташований у центрі міста або в спеціально призначеній комерційній зоні.

Зв'язок між заводом і відділеним офісом забезпечений за допомогою технологій комунікацій, таких як електронна пошта, телефонія, відеоконференції, забезпечуючи ефективну комунікацію та обмін інформацією між різними підрозділами та співробітниками.

Важливо зазначити, що точне розташування заводу та відділеного офісу Дніпровського трубного заводу може змінюватися залежно від розширення, реконструкції або інших факторів, пов'язаних з розвитком підприємства.

Об'єднання заводу на відділеного офісу за допомогою однієї інфраструктури дозволяє забезпечити такі можливості:

- завдяки централізованій інфраструктурі, ресурси, такі як принтери, зовнішні зберігальні пристрої, модеми та інші периферійні пристрої, можуть бути

ефективно використовані з усіх підключених робочих станцій. Наприклад, кілька робочих станцій можуть спільно використовувати один принтер або зовнішній зберігальний пристрій;

- централізований доступ до баз даних дозволяє робочим станціям отримувати необхідну інформацію з периферійних місць, які потребують доступу до цих даних. Наприклад, робітники офісу можуть мати доступ до централізованої бази даних, яка містить інформацію про стан виробництва на заводі;

- централізоване використання програмних засобів дозволяє робочим станціям одночасно використовувати централізовані програми, які були встановлені на сервері. Наприклад, декілька користувачів можуть використовувати один і той же програмний пакет для редакції документів або обробки даних;

- централізоване використання ресурсів процесора дозволяє використовувати обчислювальну потужність для обробки даних іншими системами, що входять до мережі. Наприклад, деякі обчислювальні завдання можуть бути виконані на сервері, звільняючи ресурси процесора на робочих станціях.

- централізовані прикладні програмні засоби зазвичай встановлюються на сервері і можуть бути використані багатьма користувачами одночасно. Наприклад, декілька користувачів можуть використовувати централізоване програмне забезпечення для обробки даних або виконання спільних завдань.

Рекомендований вид кабелю для системи ТОВ «Дніпровського трубного заводу» - мережі Fast Ethernet - це кабель витої пари (twisted pair cable). Один з популярних типів кабелю витої пари, який рекомендується використовувати для Fast Ethernet, це кабель категорії 5e (Category 5e cable).

Категорія 5e (або CAT 5e) кабелю витої пари відповідає покращеним стандартам, які забезпечують підвищену пропускну здатність і зменшення перешкод. Він здатний передавати дані зі швидкістю до 1000 Мбіт/с (1 Гбіт/с), що відповідає стандарту Gigabit Ethernet. Кабель категорії 5e є покращеною версією кабелю категорії 5 і забезпечує кращу якість сигналу і меншу кількість помилок передачі даних.

Серія кабелю витой пари CAT 5e, рекомендована для використання у даній системі, може бути, наприклад, "UTP" (Unshielded Twisted Pair) або "FTP" (Foiled Twisted Pair). UTP - це кабель без екранування, що складається з неекраниваних скручених пар провідників. FTP - це кабель з екрануванням, що має фольгу або фольгу разом зі скрученими парами провідників.

Таким чином, рекомендовано використовувати CAT 5e кабель витой пари, зокрема UTP або FTP конфігурацію, для побудови мережі Fast Ethernet у даній системі [2].

На діаграмі рисунку 1.3 показано підтримку варіантів використання, які вимагають розширення доступу до корпоративної мережі в дуже безпечний спосіб до межі мережі IoT. Ця архітектура використовує безпечне з'єднання VPN від периферійних шлюзів до кластера VPN-агрегаторів у корпоративному центрі обробки даних. Це оптимально для підприємства так як потрібно передавати конфіденційну та конфіденційну інформацію між периферійними пристроями та корпоративною штаб-квартирою або потрібно безпечно розширити доступ до периферії для певних корпоративних мереж або служб.

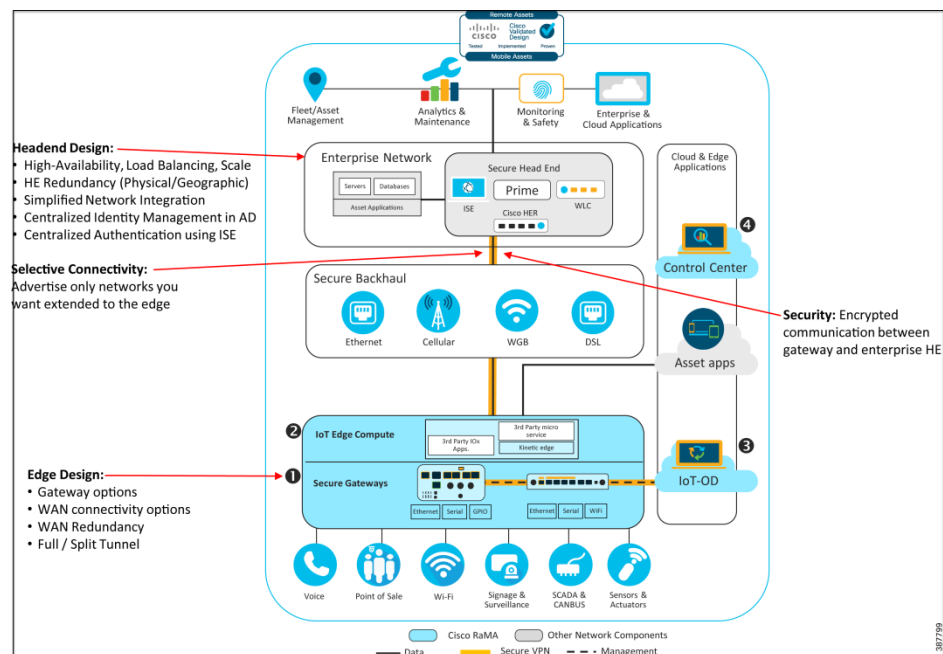


Рисунок 1.3 – Архітектура інтеграції корпоративної мережі з використанням IoT [2]

1.4 Принципи, технічні способи інформаційного забезпечення для ТОВ «Дніпровського трубного заводу»

Інформаційне забезпечення в комп'ютерних мережах включає технічні, комп'ютерні методи, які забезпечують передачу та обробку інформації між комп'ютерами. Основна мета інформаційного забезпечення полягає в забезпеченні безперебійної та надійної комунікації між різними вузлами комп'ютерної мережі.

Ключові складові інформаційного та технічного забезпечення комп'ютерної мережі для ТОВ «Дніпровського трубного заводу» складається з:

- комунікаційного обладнання. Це включає комутатори, маршрутизатори, мережеві концентратори та інші пристрої, які забезпечують з'єднання між різними вузлами мережі. Наприклад, комутатори використовуються для підключення робочих станцій, серверів та інших пристроїв до мережі;
- сервери використовуються для централізованого зберігання даних, надання послуг, які можуть бути доступні для користувачів мережі, а також для керування та адміністрування мережі. Наприклад, на трубному заводі можуть бути сервери для зберігання даних про виробництво, керування обладнанням, електронної пошти тощо;
- важливо мати належно організовану інфраструктуру кабелів для підключення пристроїв до мережі. Це може включати кабелі витої пари (twisted pair), оптоволокно (fiber optic) або бездротові з'єднання. Кабелі повинні бути встановлені з урахуванням вимог швидкості передачі даних та відстані між пристроями.
- з метою захисту важливої інформації та забезпечення конфіденційності, цілісності та доступності даних, на трубному заводі слід використовувати механізми безпеки мережі. Це може включати використання файрволів, віртуальних приватних мереж (VPN), систем аутентифікації та авторизації користувачів, а також резервного копіювання даних.
- систему моніторингу та керування. Ці системи дозволяють відстежувати стан мережі, виявляти проблеми та проводити керування мережевими пристроями. Наприклад, системи моніторингу можуть виявляти

відмови в роботі обладнання або перевантаження мережі, що допомагає оперативно реагувати та усунути проблеми.

- програмного забезпечення. На трубному заводі використовуються різні програмні засоби для керування виробництвом, ведення обліку, планування ресурсів, а також для спільної роботи і обміну даними між робочими станціями;
- протоколи передачі даних, такі як TCP (Transmission Control Protocol), який забезпечує передачу пакетів даних від джерела до призначення та вимагає встановлення з'єднання перед комунікацією між обчислювальними пристроями у мережі.

1.5 Аналітичний огляд існуючих способів обробки та передачі інформації, принципів побудови об'єкта проектування, відомих рішень у галузі

Аналітичний огляд існуючих способів обробки та передачі інформації, принципів побудови об'єкта проектування та відомих рішень для комп'ютерної системи ТОВ «Дніпровського трубного заводу», що забезпечує розгляд технологій, які можуть бути використані для покращення ефективності та безпеки мережі..

Обробка та передача інформації складається з:

- використання протоколів передачі даних, таких як TCP/IP, для забезпечення надійної та швидкої передачі даних у мережі;
- розгляд використання бездротових технологій, таких як Wi-Fi, для створення безпроводових з'єднань на трубному заводі;
- використання високопропускних мережевих комутаторів та маршрутизаторів для забезпечення швидкого обміну даними у великих масштабах.

Принципи побудови об'єкта проектування:

- розгляд використання розподіленої архітектури мережі, що дозволяє розподілити навантаження та забезпечити більшу масштабованість;
- застосування принципів модульності, що дозволяють розбити мережеве програмне забезпечення на незалежні модулі для більшої гнучкості та легкості супроводження.

Відомі рішення для комп'ютерних мереж трубного заводу:

- використання віртуалізації серверів для забезпечення ефективного використання обчислювальних ресурсів та зниження витрат на обладнання;
- застосування систем контролю доступу та ідентифікації для забезпечення безпеки мережі та захисту конфіденційної інформації. Складається з рівня доступу до мережі , рівня розподілу та рівня ядра;
- використання систем моніторингу мережі для постійного контролю та аналізу її стану, що допомагає виявляти проблеми та приймати своєчасні заходи для їх вирішення.

1.6 Завдання і мета роботи

Метою кваліфікаційної роботи є розробка комп'ютерної системи для ТОВ "Дніпровський трубний завод" з фокусом на побудові, налаштуванні та безпеці корпоративної мережі

Завдання роботи:

- побудова топології мережі. Визначення фізичної структури мережі, включаючи розташування мережевого обладнання та з'єднання між ними (рис.1.4);

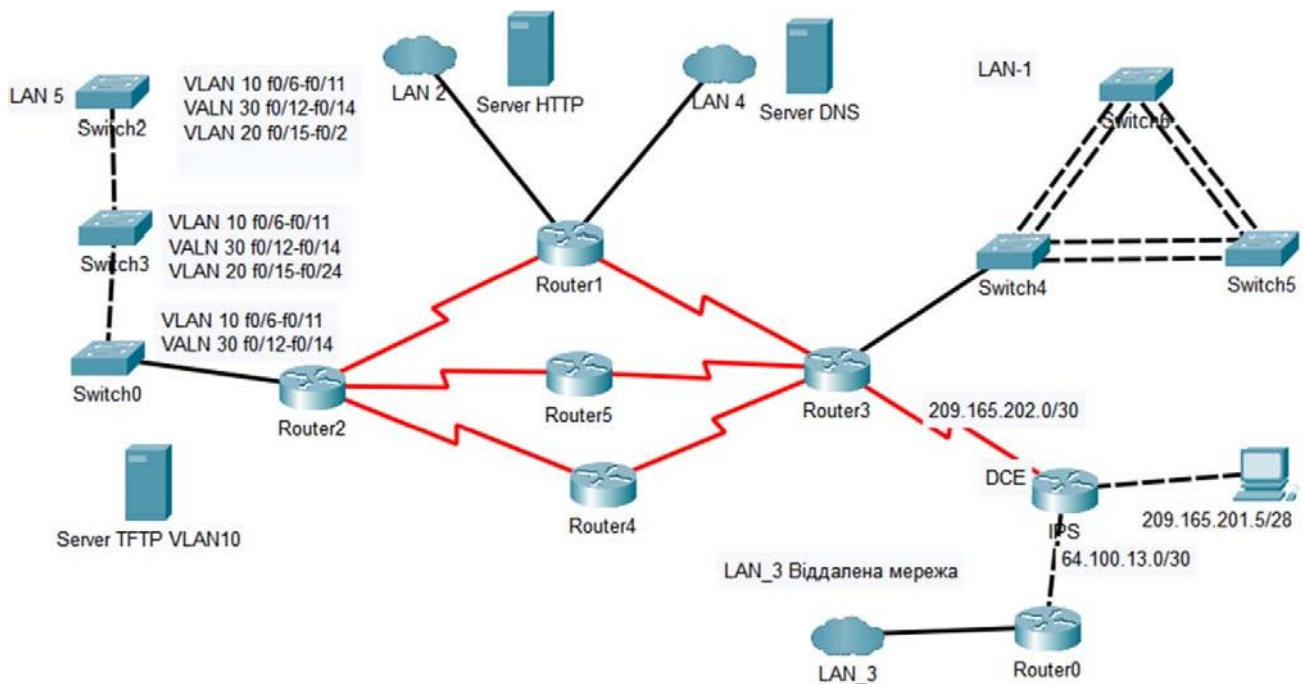


Рисунок 1.4 – Структура КМ ТОВ Дніпровський трубний завод

- розрахунок IP-адрес з використанням методу VLSM. Використовуючи

метод VLSM (маска зі змінною довжиною), встановлення IP-адрес для кожного пристрою в мережі, забезпечуючи ефективне використання доступних адрес;

- базове налаштування мережевого обладнання. Налаштування основних параметрів мережевого обладнання, таких як комутатори та маршрутизатори, включаючи IP-адреси, маски підмереж, розподіл DHCP, аутентифікацію та інші параметри;

- налаштування маршрутизації. Встановлення правил маршрутизації для забезпечення ефективного пересилання пакетів між різними підмережами в мережі;

- налаштування виходу до Інтернету з використанням динамічного NAT: Налаштування динамічного Network Address Translation (NAT) для забезпечення доступу до Інтернету з приватних IP-адрес у мережі;

- налаштування VLAN: Розбиття мережі на віртуальні LAN-сегменти для керування трафіком і забезпечення безпеки та ефективності;

- налаштування списків доступу. Встановлення правил доступу до ресурсів мережі для контролю та обмеження доступу користувачів до певних сервісів чи даних;

- налаштування сервісу AAA на сервері та маршрутизаторах. Встановлення сервісу автентифікації, авторизації та обліку (AAA) для керування доступом користувачів до мережевих ресурсів та забезпечення безпеки;

- налаштування VPN site-to-site між головною мережею та віддаленим відділом. Створення віртуальної приватної мережі (VPN) для безпечного обміну даними між головним офісом і віддаленим відділом;

- вибір та розміщення датчиків вимірювання рівня кисню у різних зонах трубного заводу. Розташування датчиків повинно бути стратегічно вибраною, щоб забезпечити належне покриття всієї території і детектувати потенційно небезпечні рівні кисню;

- підключення датчиків до мережі. Забезпечення правильного підключення датчиків рівня кисню до комп'ютерної мережі. Це може включати встановлення необхідного обладнання, наприклад, модулів збору даних або

преобразувачів, які дозволяють зчитувати дані з датчиків та передавати їх по мережі.

Усі ці завдання повинні бути виконані з використанням програмного забезпечення Cisco Packet Tracer, яке забезпечує можливість моделювання, налаштування та тестування мережевого середовища.

1.7 Основні особливості та проблематика впровадження системи моніторингу рівня кисню

Впровадження системи моніторингу рівня кисню на основі IoT (інтернет речей) датчиків в Cisco Packet Tracer має свої особливості та проблематику.

Cisco Packet Tracer обмежує вибір IoT датчиків, які можна використовувати для реалізації системи моніторингу. Тому, перед впровадженням системи, необхідно переконатися, що Cisco Packet Tracer підтримує необхідні типи IoT датчиків для моніторингу рівня кисню.

Впровадження IoT датчиків для системи моніторингу в Cisco Packet Tracer вимагає підтримки необхідних протоколів та мережевих пристроїв. Деякі IoT датчики можуть вимагати спеціального обладнання або протоколів (наприклад, Zigbee або LoRa), які можуть вимагати додаткової конфігурації та підтримки в мережевому середовищі.

Впровадження IoT датчиків для системи моніторингу пов'язане з питаннями безпеки. Оскільки IoT датчики можуть передавати чутливі дані через мережу, необхідно забезпечити захист цих даних, використовуючи механізми шифрування, аутентифікації та контролю доступу.

Впровадження системи моніторингу на базі IoT датчиків вимагає правильного налаштування збору та передачі даних з датчиків до центрального моніторингового вузла. Це може включати встановлення правильних параметрів збору даних, визначення інтервалів відправки даних та розміщення проміжних вузлів для передачі даних через мережу.

Cisco Packet Tracer може вимагати додаткової конфігурації для відображення та аналізу даних, зібраних з IoT датчиків. Це може включати розробку веб-інтерфейсу або програмного забезпечення для відображення реального часу та аналізу даних рівня кисню.

Необхідно враховувати можливості масштабування системи моніторингу, якщо потрібно впровадити більше IoT датчиків або розширити мережу. Cisco Packet Tracer повинен бути готовий до розширення та управління більшою кількістю датчиків у системі моніторингу.

Розробка системи моніторингу рівня кисню на основі IoT датчиків в Cisco Packet Tracer потребує уваги до деталей, таких як підтримка датчиків, сумісність з мережевим обладнанням, безпека даних, налаштування передачі даних та масштабованість. Успішне впровадження цієї системи вимагатиме дослідження та налаштування відповідних компонентів в Cisco Packet Tracer.

2 РОЗРОБКА АПАРАТНОЇ ЧАСТИНИ КОМП'ЮТЕРНОЇ СИСТЕМИ

2.1 Технічні вимоги до комп'ютерної системи

Комп'ютерна система ТОВ "Дніпровський трубний завод з системою моніторингу рівня кисню" призначена для контролю та нагляду за рівнем кисню в різних зонах заводу. Основна мета системи полягає у забезпеченні безпеки та ефективності процесів, пов'язаних з роботою з газами та рідинами, що містять кисень [1].

Особливості та функції системи моніторингу рівня кисню включають:

- встановлення датчиків кисню. Система передбачає розміщення датчиків кисню в стратегічних точках на території заводу. Ці датчики здатні точно вимірювати рівень кисню та передавати ці дані до центральної системи моніторингу;
- центральна система моніторингу. Центральна система забезпечує централізований збір, обробку та аналіз даних, що надходять від датчиків кисню. Вона відображає рівень кисню в реальному часі та надає операторам можливість моніторити стан та вчасно реагувати на будь-які зміни;
- попередження та тривоги. Система має можливість налаштовувати попередження та тривоги для різних рівнів кисню. У разі перевищення допустимих меж система автоматично відправляє сповіщення операторам, щоб вжити необхідні заходи для нормалізації ситуації;
- запис та збереження даних. Система моніторингу має можливість записувати та зберігати дані про рівень кисню протягом певного періоду часу. Це дозволяє здійснювати аналіз даних, виявляти тенденції та покращувати процеси роботи згідно зі зібраними відомостями;
- візуалізація та звітність. Система надає можливість візуалізувати дані про рівень кисню у вигляді графіків, діаграм або інших форматів, які

допомагають в оперативному аналізі та прийнятті рішень. Також система може генерувати звіти, які містять інформацію про статус, тренди та статистику рівня кисню на заводі.

Комп'ютерна система має складатися з наступних підсистем:

- відділ продажів та маркетингу – LAN1;
- відділ дослідження та розвитку – LAN12;
- відділ розробки та проектування – LAN3;
- відділ якості – LAN4;
- відділ закупівель – LAN5.

2.2 Вимоги до функцій, які виконує КС

Комп'ютерна система ТОВ «Дніпровський трубний завод» з системою моніторингу рівня кисню має виконувати ряд важливих функцій, пов'язаних з ефективним функціонуванням системи моніторингу. Основні вимоги до функцій мережі включають [2]:

- забезпечувати безперебійний та надійний зв'язок між датчиками кисню та центральною системою моніторингу. Це включає встановлення та налаштування мережевого обладнання, такого як комутатори, маршрутизатори та кабелі зв'язку;
- забезпечувати збір даних про рівень кисню з датчиків і передачу цих даних до центральної системи моніторингу. Це може включати використання протоколів передачі даних, таких як TCP/IP, для надійної передачі інформації;
- забезпечувати централізоване управління системою моніторингу рівня кисню. Це включає можливість збору, аналізу та візуалізації даних, налаштування параметрів системи, а також встановлення тривоги та сповіщень;

- повинна мати високий рівень безпеки для запобігання несанкціонованому доступу до системи моніторингу рівня кисню. Це може включати застосування механізмів аутентифікації, авторизації та шифрування даних, а також встановлення мережевих фаєрволів та інших захисних заходів;
- повинна бути масштабованою, щоб враховувати потенційне зростання кількості датчиків кисню та інших компонентів системи моніторингу. Це включає гнучкість та легкість розширення мережевої інфраструктури при необхідності;
- повинна бути надійною та доступною, щоб забезпечити постійний доступ до даних моніторингу рівня кисню. Це може включати використання резервних шляхів зв'язку, дублювання мережевого обладнання та резервування джерел живлення;
- здатна інтегруватися з існуючими системами ТОВ "Дніпровський трубний завод" для обміну даними та координації роботи. Це може включати інтеграцію з системами управління виробництвом, системами безпеки та іншими релевантними системами.

Загалом, мережа для ТОВ «Дніпровський трубний завод» з системою моніторингу рівня кисню повинна забезпечувати надійну та безперебійну передачу даних, ефективне управління та контроль над системою моніторингу, високий рівень безпеки та інтеграцію з існуючими системами.

2.3 Чисельність та склад персоналу

Система повинна мати достатню кількість кваліфікованих працівників для забезпечення ефективного функціонування та обслуговування. Чисельність персоналу повинна бути достатньою для покриття всіх необхідних обов'язків та забезпечення безперебійності роботи системи.

Персонал, який обслуговує систему, повинен мати необхідні знання,

навички та досвід для ефективного виконання своїх обов'язків. Це може включати розуміння принципів роботи системи моніторингу рівня кисню, знання процедур обслуговування та ремонту, а також навички роботи з відповідним обладнанням та програмним забезпеченням. Персоналом, який обслуговує ТОВ «Дніпровський трубний завод»:

- інженери з електроніки та автоматизації. Це кваліфіковані фахівці з глибокими знаннями в галузі електроніки, автоматизації та контролю. Вони повинні мати розуміння принципів роботи датчиків рівня кисню, спроможність налагоджувати, монтувати та підтримувати цю систему;
- фахівці з мережевих технологій. Для належної роботи системи моніторингу рівня кисню необхідні фахівці з мережевих технологій, які мають досвід у налаштуванні та управлінні мережевим обладнанням, а також знання про протоколи мережевого зв'язку;
- аналітики даних. Це спеціалісти, які володіють навичками обробки і аналізу даних. Вони використовують спеціалізоване програмне забезпечення для моніторингу та аналізу даних про рівень кисню, що допомагає виявляти тенденції, проблеми та виробляти звіти.
- технічна підтримка. Завод має команду фахівців з технічної підтримки, які відповідають за забезпечення належної роботи системи, відладку проблем, відповіді на запити користувачів та надання допомоги у разі необхідності.

2.4 Вимоги до технічного забезпечення системи

Комутаційне обладнання адміністративної підсистеми структурованої кабельної системи має бути монтуванням на 19-дюймові направляючі у телекомунікаційних шафах.

Телекомунікаційні шафи повинні відповідати наступним вимогам [3]:

- шафа повинна мати двосторонній доступ;
- передні двері шафи повинні бути скляними, з кутом відкриття не менше 180 градусів або бути легко знімними;
- шафа має мати дві бічні суцільнометалеві неперфоровані стінки.
- шафа повинна мати встановлені пристрої для організації комутаційних шнурів (органайзери);
- шафа повинна бути заземлена усіма своїми компонентами;
- шафа має бути оснащена внутрішнім замком для закриття.

Для забезпечення рівномірного укладання комутаційних шнурів у телекомунікаційній шафі слід передбачити встановлення кабельних органайзерів.

Остаточну комплектацію кожної шафи слід визначати на етапі проектування.

Мережевий комутатор доступу для ТОВ "Дніпровський трубний завод з системою моніторингу рівня кисню" повинен мати такі характеристики:

- 48 портів стандартів 10BASE-T/100BASE-TX з пропускнуою спроможністю 10/100 Мбіт/с;
- два порти 1 Гбіт з роз'ємом на основі SFP для висхідних підключень;
- пропускна спроможність не менше 100 Гбіт/с;
- підтримка розміру кадру до 9000 байт (Jumbo frames) для всіх портів;
- максимальний розмір блоку (MTU) до 9000 байт;
- розмір таблиці MAC-адрес не менше 8000;
- максимальна кількість мереж VLAN не менше 255;
- мінімум 4000 ідентифікаторів VLAN;
- мінімум 128 Мб оперативної пам'яті;
- підтримка правил списків доступу на кожному порту;
- можливість застосування наступних функцій швидкого розгортання.
- Мережевий комутатор обрано фірму Cisco Catalyst 3650 Series з

характеристиками:

- 48 портів 10/100/1000BASE-T;
- підтримку SFP для висхідних підключень;
- підтримку VLAN;
- технологію QoS
- пропускна спроможність не менше 100 Гбіт/с.

Датчики, що використовуються в системі, мають задовольняти такі специфічні вимоги:

- мають мати високий рівень захисту від вологості, наприклад, відповідати стандарту IP65 або вище. це дозволить їм захистити від бризок, конденсації та інших вологих умов, які можуть бути присутні у паркінговому середовищі;

- мають працювати в широкому діапазоні робочих температур, наприклад, від -20°C до +60°C. це забезпечить надійну роботу як в холодних зимових умовах, так і в спекотному літньому кліматі;

- мають високу точність та швидкість вимірювання, що гарантує достовірну інформацію про вологість, температуру та задимленість. наприклад, точність вимірювання може бути в межах $\pm 1\%$ для вологості та $\pm 0,5^\circ\text{C}$ для температури;

- мають бути зручними та швидкими у монтажі, щоб забезпечити простоту установки у паркінговому середовищі;

- конструкція датчиків повинна бути міцною, щоб витримувати умови паркінгового середовища, включаючи вібрації, удари та потенційні пошкодження.

2.5 Вимоги до інформаційного забезпечення

Корпоративна мережа використовується для з'єднання комп'ютерів та серверів, які розташовані в різних відділах або віддалених будівлях, з метою забезпечення доступу до спільної інформаційної системи.

Об'єднання комп'ютерів в одну систему має на меті полегшити та збільшити ефективність роботи з даними та додатками. Користувачі отримують спільний доступ до ресурсів, таких як спільні папки або бази даних, що спрощує спільну роботу та обмін інформацією. Вони також можуть отримувати доступ до необхідних програм та даних з будь-якого комп'ютера в мережі.

Об'єднання комп'ютерів в одну інформаційну систему також спрощує впровадження та підтримку заходів безпеки. Застосовуються єдиною політикою безпеки, механізмами автентифікації та авторизації, а також системами моніторингу та виявлення інцидентів. Це допомагає запобігти несанкціонованому доступу, витоку даних та іншим загрозам безпеки.

Корпоративна мережа складається з головної та віддаленої локальної мережі. Головна будівля включає 5 відділів, які з'єднані за допомогою виті пари CAT5e. Зв'язок з віддаленою мережею здійснюється через Інтернет.

Корпоративна мережа використовуватиметься для обміну даними, документами та інформацією, що сприятиме спільній роботі та співпраці між відділами та співробітниками. Крім того, вона буде використовуватися для централізованого зберігання даних, таких як документи, бази даних та виробничі параметри.

2.6 Розробка технічних вимог до апаратного забезпечення комп'ютерної системи

Розробка специфікації апаратних засобів комп'ютерної системи включає в себе визначення конкретних компонентів, що використовуються описані в табл.2.1.

Таблиця 2.1 – Технічна специфікація обладнання

Вид апаратного засобу	Характеристика	Кількість
Персональні комп'ютери	Модель: HP ProDesk 600 G3 Процесор: Intel Core i5-7500 Оперативна пам'ять: 8 ГБ DDR4 Жорсткий диск: 256 ГБ SSD	50 шт

Мережеві карти:	Модель: Intel Ethernet Connection I219-LM Інтерфейс: Gigabit Ethernet	50 шт
Жорсткі диски:	Модель: Seagate Barracuda 2 ТБ HDD Інтерфейс: SATA 6 Гбіт/с	50 шт
Мережеве обладнання Cisco:	Мережевий комутатор: Cisco Catalyst 2960X-48TS-L Маршрутизатор: Cisco ISR4321/K9 Брандмауер: Cisco ASA 5506-X	16 шт

2.7 Розробка загальної структури комп'ютерної системи ТОВ «Дніпровський трубний завод»

Структура комп'ютерної системи складається з наступних основних елементів:

Сервери – це центральні обчислювальні вузли, що надають ресурси та послуги іншим комп'ютерам у мережі. Вони відповідають за централізоване зберігання даних, обробку та передачу інформації, а також виконання спеціалізованих завдань [3];

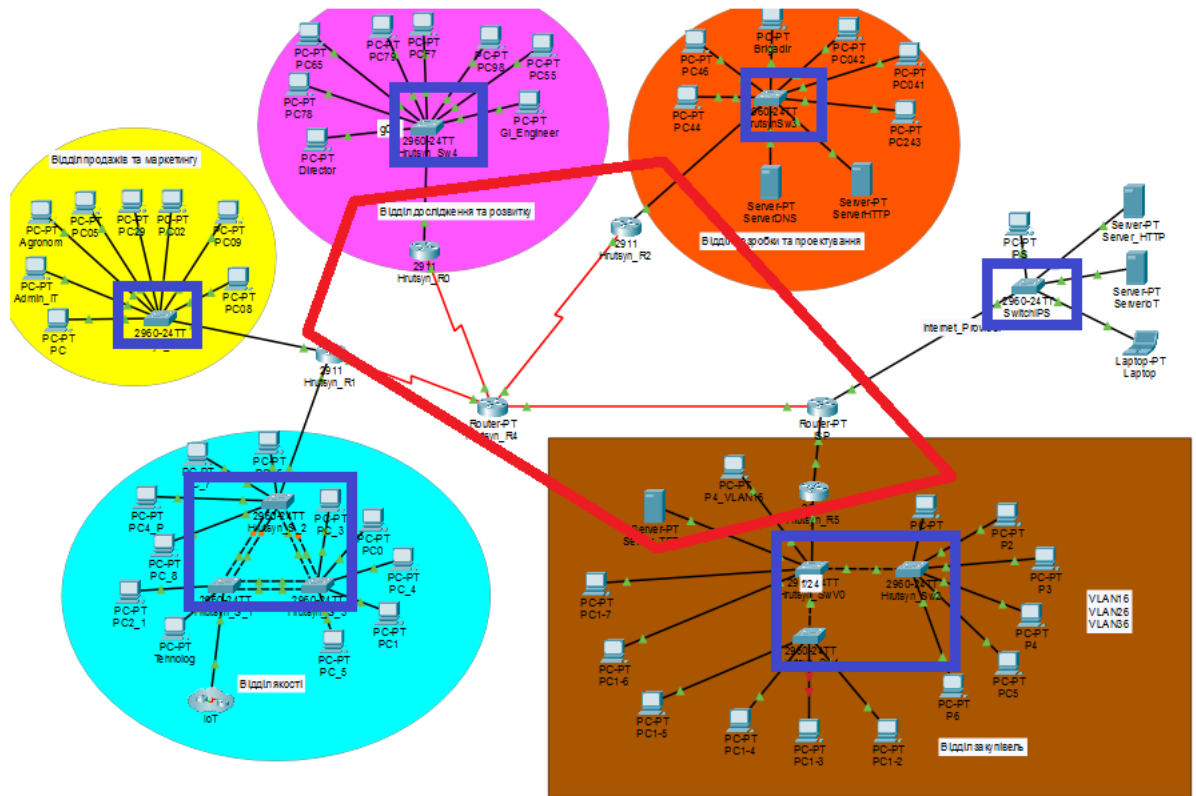
клієнтські комп'ютери (робочі станції) – це термінали, з яких користувачі отримують доступ до ресурсів та послуг, що надаються серверами;

мережеве обладнання – це комутатори, маршрутизатори та мережеві контролери, які забезпечують з'єднання та комунікацію між різними компонентами мережі.

Структурна схема комп'ютерної системи ТОВ «Дніпровський трубний завод» представлена на рис.2.1 та складається :

– рівень ядра. Використано 6 маршрутизаторів, які виконують маршрутизацію трафіку та підключені до мереж WAN. Шлюзовий маршрутизатор рівня ядра забезпечує підключення проєктованої мережі до Інтернету за допомогою технології VPN.

– рівень доступу включає тринадцять комутаторів, які формують LAN та VLAN підмережі. Цей підхід дозволяє безпосередньо передавати дані від кожного комутатора до отримувача, покращуючи продуктивність та забезпечуючи безпеку мережі. У підмережі " Відділ закупівель" використовуються два комутатори з технологією VLAN для ізольованого доступу до мережевих ресурсів цього відділу. У підмережі " Відділ якості" використовуються три комутатори з використанням технологій PAgP та LACP для збільшення пропускної здатності та надійності каналу передачі даних.



Умовні позначення:
■ - рівень ядра;
■ - рівень доступу;
 - кінцеві користувачі.

Рисунок 2.1 - Схематичне представлення структури КС ТОВ «Дніпровський трубний завод»

3 РОЗРОБКА КОРПОРАТИВНОЇ МЕРЕЖІ

3.1 Розрахунок IP-адрес та масок підмереж для ефективної адресації корпоративної мережі

Для побудови мережі ТОВ «Дніпровський трубний завод» був використаний IP-адресний діапазон 172.23.0.0/21, що надає нам 11 біт для розподілу підмереж та пристроїв у цій мережі.

Для ефективного використання доступних IP-адрес та мінімізації втрати адресного простору, використовується технологія VLSM (Variable Length Subnet Masking). VLSM дозволяє гнучко використовувати IP-адреси, розподіляючи їх на підмережі з різними розмірами масок.

Традиційно, стандартні мережеві конфігурації використовують однакові маски підмереж для всіх підмереж, незалежно від їхнього розміру або кількості пристроїв. Однак це може призводити до неефективного використання IP-адрес.

Завдяки VLSM ми можемо розподіляти IP-адреси більш гнучко, використовуючи маски підмереж різної довжини в мережі. Це дозволяє точніше визначати розмір кожної підмережі відповідно до потреб мережі, що забезпечує економію IP-адрес та краще використання доступних ресурсів.

Важливо зазначити, що адреси з діапазону 10.0.11.0/24 використовуються виключно для послідовних каналів, що з'єднують маршрутизатори. Ці адреси не включаються до загального IP-адресного простору 172.23.0.0/21.

LAN1 складається 47 вузлами. Потрібно 6 біт для представлення 47 вузлів ($2^6 - 2 = 62$ доступні адреси, враховуючи мережеву та широкомовну адресу). Маска підмережі: /26 (255.255.255.192). Перший адрес підмережі: 172.23.0.0, останній адрес: 172.23.0.63.

Для LAN2 треба 54 вузлами. Потрібно 6 біт для представлення 54 вузлів ($2^6 - 2 = 62$ доступні адреси, враховуючи мережеву та широкомовну адресу). Маска

підмережі: /26 (255.255.255.192). Перший адрес підмережі: 172.23.0.64, останній адрес: 172.23.0.127.

Для LAN3 з 39 вузлами потрібно 6 біт для представлення 39 вузлів ($2^6 - 2 = 62$ доступні адреси, враховуючи мережеву та широкомовну адресу). Маска підмережі: /26 (255.255.255.192). Перший адрес підмережі: 172.23.0.128, останній адрес: 172.23.0.191.

Розрахунок для мережі LAN4 з 86 вузлами потрібно 7 біт для представлення 86 вузлів ($2^7 - 2 = 126$ доступні адреси, враховуючи мережеву та широкомовну адресу). Маска підмережі: /25 (255.255.255.128). Перший адрес підмережі: 172.23.0.192, останній адрес: 172.23.0.255.

Для мережі LAN5 з 77 вузлами потрібно 7 біт для представлення 77 вузлів ($2^7 - 2 = 126$ доступні адреси, враховуючи мережеву та широкомовну адресу). Маска підмережі: /25 (255.255.255.128). Перший адрес підмережі: 172.23.1.1, останній адрес: 172.23.1.127.

Таким чином, ми розрахували маски підмереж для кожної мережі, що дозволить ефективно використовувати доступні IP-адреси та мінімізувати втрату адресного простору.

Всі підмережі обчислюються за однаковим принципом, і результати цих розрахунків представлені в таблиці 3.1.

Таблиця 3.1 – Схема адресації корпоративної мережі ТОВ «Дніпровський трубний завод»

Назва підмережі	Необхідна кількість вузлів	Номер мережі	Маска мережі	Діапазон доступних адрес
Відділ продажів та маркетингу – LAN1	47	172.23.0.0	255.255.255.192	172.23.0.1 – 172.23.152.62

Відділ дослідження та розвитку – LAN 2	54	172.23.0.63	255.255.255.192	172.23.0.64– 172.23.0.127
Відділ розробки та проектування – LAN3	39	172.23.0.127	255.255.255.192	172.23.0.128- 172.23.0.190
Відділ якості – LAN4	86	172.23.0.192	255.255.255.128	172.23.0.193- 172.23.0.254
Відділ закупівель – LAN5	57	172.23.1.0	255.255.255.128	172.23.1.1- 172.23.1.126

На основі отриманих розрахунків можна зробити висновок, що з 2046 доступних адрес було використано 294, хоча необхідна кількість становить 207. Тобто використано 15% доступного адресного простору, а це становить 70% від загальної потреби.

3.2 Розрахунок схеми адресації пристроїв

Для розрахунку адресації пристроїв в мережі, необхідно враховувати проектні вимоги: перші можливі IP-адреси призначаються інтерфейсам та підінтерфейсам маршрутизаторів у локальній мережі (LAN). Другі доступні IP-адреси призначаються комутаторам для кожної LAN. Останні використовувані IP-адреси призначаються вузлам (кінцевим пристроям). В мережах VLAN використовується протокол DHCP для адресації кінцевих пристроїв.

На основі проведених розрахунків, які представлені в таблиці 3.1, була розроблена схема адресації пристроїв, яку можна знайти у таблиці 3.2.

Таблиця 3.2 – Схема призначення IP-адрес для пристроїв в мережі

Ім'я пристрою	Інтерфейс	IP-адреса	Маска	Шлюз	VLAN	Інтерфейс підключеного пристрою
LAN1						
Hrutsyn_R1	G0/1	172.23.0.1	/24	-	-	G0/1
	G0/2	10.22.211.129	/26	-	-	G0/2
	S0/0/0	10.0.6.1	/30	-	-	S0/0/0
Hrutsyn_Sw1	Vlan1	10.22.211.130	/26	10.22.211.129	-	G0/1
PC1 – PC11	NIC	10.22.211.190-10.22.211.179	/26	10.22.211.129	-	Fa0/1- Fa0/11
Printer1- Printer2	NIC	10.22.211.138	/26	10.22.211.129	-	Fa0/12- Fa0/13
LAN5						
Hrutsyn_R1	G0/1	172.23.1.1	/24	-	-	G0/1
Hrutsyn_Sw5	Vlan1	10.22.209.2	/24	10.22.209.1	-	G0/2
PC1- PC8	NIC	10.22.209.254-10.22.209.246	/24	10.22.209.1	-	F0/0-F0/8
LAN4						
Hrutsyn_R1	G0/1	172.23.0.193	/24	-	-	G0/1
	S0/0/0	10.0.6.5	/30	-	-	S3/0
Hrutsyn_Sw4	Vlan1	10.22.208.2	/24	10.22.208.1	-	G0/1
PC1- PC8	NIC	10.22.208.254-10.22.208.246	/24	10.22.208.1	-	F0/0-F0/8
Printer1	NIC	10.22.208.9	/24	10.22.208.1	-	Fa0/12
LAN3						
Hrutsyn_R4	G0/0	10.22.210.1	/24	-	-	G0/0
	S0/0/0	10.0.6.9	/30	-	-	S6/0
Hrutsyn_Sw3	Vlan1	10.22.210.2	/24	10.22.210.1	-	G0/0
PC1- PC7	NIC	10.22.210.254-10.22.210.248	/24	10.22.210.1	-	F0/0-F0/6
Server_HTTP	NIC	10.22.210.9	/24	10.22. 210.1	-	Fa0/23
Server_DNS	NIC	10.22.210.10	/24	10.22.210.1	-	Fa0/24
AccessPoint0	Port0	10.22.138.139	/24	10.22.210.1	-	Fa0/22
LAN2						
Hrutsyn_R0	G0/1	-	-	-	-	-
	G0/1.16	10.22.211.1	/27	-	16	G0/1
	G0/1.26	10.22.211.33	/27	-	26	G0/1
	G0/1.36	10.22.211.65	/27	-	36	G0/1
	G0/1.99	10.22.211.97	/28	-	99	G0/1
	F0/0	64.100.13.2	/30	-	-	F0/0
PC16.1-PC16.4	NIC	172.23.0.68-172.23.0.84	/27	-	16	Fa0/15- Fa0/18
PC26.1-PC26.4	NIC	172.23.0.85-10.22.211.97	/27	-	26	Fa0/10- Fa0/14
ServerTFTP	NIC	10.22.211.43	/27	-	26	Fa0/13
PC36.1-PC36.5	NIC	172.23.0.98-172.23.0.104	/27	-	36	Fa0/5- Fa0/9
Hrutsyn_SwV1	G0/1	172.23.0.105	/28	-	99	-
Hrutsyn_SwV2	F0/12	172.23.0.106	/28	-	99	-
Hrutsyn_SwV3	F0.11	172.23.0.107	/28	-	99	-
LAN1						
Hrutsyn_R3	S2/0	172.23.0.64	/30	-	-	S0/0/0
	S3/0	10.0.6.6	/30	-	-	S0/0/0

	S6/0	10.0.6.10	/30	-	-	S0/0/1
	F4/0	209.135.202.1	/30	-	-	F4/0
DLC100	Internet	209.165.200.10	/27	-	-	G0/1
IoT1 - IoT10	Wireless	192.168.25.101- 192.168.25.111	/24	192.168.25.1	-	Wireless
IPS						
Rout_IPS	F4/0	209.165.202.2	/30	-	-	F4/0
	F0/0	64.100.13.1	/30	-	-	G0/2
	F1/0	53.1.9.1	/24	-	-	G0/1
Server_IPS	NIC	53.1.9.10	/24	53.1.9.1	-	Fa0/1
ISP_Sw	Vlan1	53.1.9.2	/24	53.1.9.1	-	Fa0/2
Server_IoT	NIC	53.1.9.9	/24	53.1.9.1	-	Fa0/3
Server_CO	NIC	53.1.9.8	/24	53.1.9.1	-	Fa0/4

3.3 Розробка схеми логічної топології корпоративної мережі

Для створення топологічної схеми, спочатку необхідно визначитися з архітектурою мережі.

В якості архітектури була обрана пасивна зірка, яка базується на використанні центрального пристрою, що виступатиме в ролі комутатора. Пасивна зірка забезпечує високу ефективність мережі. Кожен пристрій має окреме підключення до центрального комутатора, що унеможливорює конфлікти та перевантаження мережі, які часто спостерігаються в інших типах топологій. У разі виникнення проблеми легко визначити, який саме пристрій спричиняє проблему, оскільки всі пристрої підключені безпосередньо до центрального комутатора.

На стороні серверів та у віддаленій мережі використовується розширена зірка. Розширена зірка також має центральний вузол зв'язку, проте на відміну від класичної зірки, вона дозволяє додавати додаткові вузли зв'язку або комутатори до центрального вузла, розширюючи можливості мережі. Розширена зірка забезпечує високу надійність. Якщо один із підцентрів виходить з ладу, інші підцентри та пристрої в мережі продовжують працювати без перебоїв.

З використанням топології та інформації з таблиці 3.1 та таблиці 3.2 була створена модель комп'ютерної мережі в Cisco Packet Tracer, яку можна оглянути на рис. 3.1.

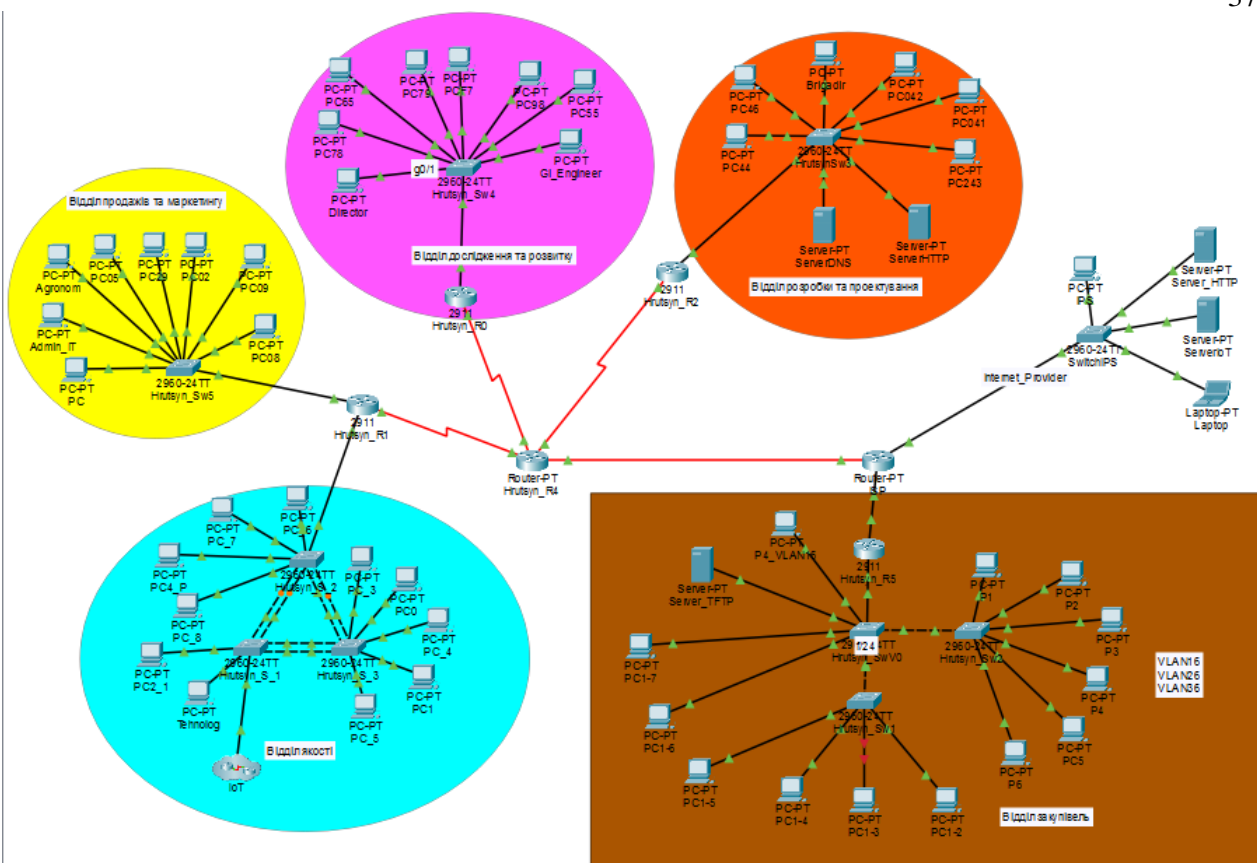


Рисунок 3.1 – Розроблена КМ ТОВ «Дніпровський трубний завод» в Cisco Packet Tracer

3.4 Базове налаштування конфігурації пристроїв КС

Процес базового налаштування конфігурації активних мережних пристроїв включає ряд кроків.

Крок 1. Використання сервісу шифрування паролів.

Крок 2. Захист привілейованого режиму ОС, консольного порту та ліній vty.

Крок 3. Призначення банера MOTD.

Крок 4. Використання протоколу SSH для віддаленого доступу до пристрою на лініях vty.

Крок 5. Створення локальних облікових записів зі збереженням паролів.

Крок 6. Встановлення доменного імені пристрою.

Крок 7. Створення ключа RSA для шифрування даних.

На рис.3.2 наведено приклад базових налаштувань на роутері Hrutsyn_R0:

```

Router0
Physical Config CLI Attributes
IOS Command Line Interface
--- System Configuration Dialog ---
Would you like to enter the initial configuration dialog? [yes/no]:
Press RETURN to get started!

Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname Hrutsyn_R0
Hrutsyn_R0(config)#
Hrutsyn_R0(config)#
Hrutsyn_R0(config)#interface GigabitEthernet0/0/0
Hrutsyn_R0(config-if)#ip address 172.23.1.1 255.255.0.0
Hrutsyn_R0(config-if)#ip address 172.23.1.1 255.255.255.192
Hrutsyn_R0(config-if)#no shutdown
Hrutsyn_R0(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0/0, changed state to up

Hrutsyn_R0(config-if)#ex
Hrutsyn_R0(config)#service password-encryption
Hrutsyn_R0(config)#enable secret class12320sk
Hrutsyn_R0(config)#line console 0
Hrutsyn_R0(config-line)#password cisco12320sk
Hrutsyn_R0(config-line)#login
Hrutsyn_R0(config-line)#exit
Hrutsyn_R0(config)#banner motd # 123-20sk-1 Grutsyn. Enter only have key#
Hrutsyn_R0(config)#ip domain-name Hrutsyn_R0
Hrutsyn_R0(config)#crypto key generate rsa
The name for the keys will be: Hrutsyn_R0.Hrutsyn_R0
Choose the size of the key modulus in the range of 360 to 4096 for your
General Purpose Keys. Choosing a key modulus greater than 512 may take
a few minutes.

How many bits in the modulus [512]: 1025
% Generating 1025 bit RSA keys, keys will be non-exportable...[OK]

Hrutsyn_R0(config)#line vty 0 4
*Mar 1 0:2:36.548: %SSH-5-ENABLED: SSH 1.99 has been enabled
Hrutsyn_R0(config-line)#login local
Hrutsyn_R0(config-line)#transport input ssh
Hrutsyn_R0(config-line)#
  
```

Рисунок 3.2 – Програмування Hrutsyn_R0

Було налаштовано:

- вимкнено пошуку DNS на маршрутизаторі;
- задано пристрою унікального імені;

- шифрування всіх паролів, що зберігаються у відкритому вигляді;
- встановлення паролю на вхід до привілейованого режиму та консольної лінії;
- налаштування запиту пароля при вході;
- налаштування банера MOTD;
- налаштування протоколу SSH та створення користувача;
- створення домену;
- генерація ключа RSA для шифрування даних;
- налаштування лінії VTY та встановлення необхідності введення логіну та пароля для входу;
- встановлення входу на лінію тільки по протоколу SSH;
- встановлення IPv4-адресу для інтерфейсу відповідно до вказаної таблиці;
- включення інтерфейсу для роботи.

3.4.1 Налаштування маршрутизаторів КС ТОВ «Дніпровський трубний завод»

Протокол EIGRP (Enhanced Interior Gateway Routing Protocol) є одним з протоколів маршрутизації внутрішньої шлюзової мережі (IGP) і використовується для обміну інформацією про маршрутизацію між маршрутизаторами у внутрішній мережі. EIGRP розроблений компанією Cisco і є пропрієтарним протоколом.

Для налаштування EIGRP на маршрутизаторі використовуються команди, такі як "router eigrp [AS number]" для активації протоколу EIGRP та визначення номера автономної системи (AS number), "network [network address]" для включення мережі до протоколу EIGRP та "eigrp router-id [router-id]" для призначення ідентифікатора маршрутизатора [4].

В Cisco-роутерах, EIGRP використовує такі метрики:

- bandwidth (Пропускна здатність): Використовується для вимірювання пропускної здатності каналу. Чим вище значення пропускної здатності, тим кращий маршрут;
- delay (Затримка): Вимірюється в мікросекундах і представляє час, необхідний для передачі пакету через мережу. Чим менше значення затримки, тим кращий маршрут;
- reliability (Надійність): Відображає ступінь надійності каналу зв'язку. Використовується як відсоткове значення, де 255 означає повну надійність, а 0 - повну ненадійність;
- load (Навантаження): Відображає ступінь завантаженості каналу. Використовується як відсоткове значення, де 255 означає максимальне навантаження, а 0 - немає навантаження;
- MTU (Максимальна одиниця передачі): Вказує максимальний розмір пакету, який може бути переданий через канал. Чим більше значення MTU, тим кращий маршрут.

```
Hrutsynov_R2(config)#router eigrp 3
```

```
Hrutsynov_R2(config-router)#eigrp router-id 3.3.3.3
```

```
Hrutsynov_R2(config-router)#network 172.23.0.1 0.0.0.64
```

```
Hrutsynov_R2(config-router)#network 172.23.0.64 0.0.0.64
```

```
Hrutsynov_R2(config-router) #passive-interface G0/1
```

```
Hrutsynov_R2(config-router) #ip route 0.0.0.0 0.0.0.0 209.165.202.2
```

```
Hrutsynov_R2(config-router) #no auto-summary
```

```
Hrutsynov_R2#copy running-config startup-config
```

```
Hrutsynov_R2#show ip route
```



```

Hrutysyn_R2
Physical Config CLI Attributes
IOS Command Line Interface
Hrutysyn_R2 (config)#banner motd # 123-2Usk-1 Grutysyn. Enter only have key#
Hrutysyn_R2 (config)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/0, changed state to down
%DUAL-5-NBRCHANGE: IP-EIGRP 6: Neighbor 10.0.6.10 (Serial0/0/0) is down: interface down
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/0, changed state to up
%DUAL-5-NBRCHANGE: IP-EIGRP 6: Neighbor 10.0.6.10 (Serial0/0/0) is up: new adjacency
Hrutysyn_R2 (config)#ex
Hrutysyn_R2#
%SYS-5-CONFIG_I: Configured from console by console
Hrutysyn_R2#sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route

Gateway of last resort is 209.165.202.2 to network 0.0.0.0

  10.0.0.0/8 is variably subnetted, 9 subnets, 5 masks
D   172.23.0.1/32 [90/21024000] via 10.0.6.10, 00:04:19, Serial0/0/0
D   10.0.6.4/30 [90/21024000] via 10.0.6.10, 00:04:19, Serial0/0/0
C   10.0.6.8/30 is directly connected, Serial0/0/0
L   10.0.6.9/32 is directly connected, Serial0/0/0
D   10.22.211.0/27 [90/20519680] via 10.0.6.10, 00:04:18, Serial0/0/0
D   10.22.211.32/27 [90/20519680] via 10.0.6.10, 00:04:18, Serial0/0/0
D   10.22.211.64/27 [90/20519680] via 10.0.6.10, 00:04:18, Serial0/0/0
D   10.22.211.96/28 [90/20519680] via 10.0.6.10, 00:04:18, Serial0/0/0
D   10.22.211.128/26 [90/21024256] via 10.0.6.10, 00:04:19, Serial0/0/0
D   53.0.0.0/8 [90/20517120] via 10.0.6.10, 00:04:19, Serial0/0/0
D 172.23.0.0/16 [90/20517120] via 10.0.6.10, 00:04:19, Serial0/0/0
C   172.23.0.128/25 is directly connected, GigabitEthernet0/1
L   172.23.0.129/32 is directly connected, GigabitEthernet0/1
D   209.165.202.0/24 [90/20514560] via 10.0.6.10, 00:04:19, Serial0/0/0
S*  0.0.0.0/0 [1/0] via 209.165.202.2

Hrutysyn_R2#

```

Рисунок 3.3 – Програмування Hrutysynov_R2

3.4.2 Налаштування служби AAA на маршрутизаторах

Служба AAA є механізмом, який використовується для керування доступом до мережевих ресурсів і включає функції аутентифікації, авторизації та обліку. Вона забезпечує високий рівень безпеки та контролю над користувачами, які намагаються отримати доступ до мережевих пристроїв.

Служба AAA дозволяє організаціям централізовано керувати доступом до мережевих ресурсів, спрощує процеси аутентифікації та авторизації, а також надає засоби для виконання аудиту і контролю за використанням мережі [5].

Відповідно до вимог, на маршрутизаторах була налаштована підтримка служби AAA та налаштований RADIUS-сервер з ключовим словом "radius123". Для облікових записів користувачів використовується ім'я пристрою з паролем "admin123".

Налаштування автентифікації AAA з використанням RADIUS-сервера складається з:

```
Hrutsyn_R2(config)#aaa new-model
Hrutsyn_R2(config)#aaa authentication login default group radius local
Hrutsyn_R2(config)#radius server serverRadius
Hrutsyn_R2(config-radius-server)#addr ipv4 172.23.0.64
Hrutsyn_R2(config-radius-server)#key radius
Hrutsyn_R2(config-radius-server)#exit
Hrutsyn_R2(config)#line console 0
Hrutsyn_R2(config-line)#login authentication default
Hrutsyn_R2(config-line)#line vty 0 15
Hrutsyn_R2(config-line)#login authentication default
```

На рисунку 3.4 наведено результат налаштувань.

```
123-20sk-1 Grutsyn. Enter only have key
User Access Verification
Username: Hrutsy_R2
Password:
Hrutsyn_R2>enable
Password:
Hrutsyn_R2#
.
Hrutsyn_R2 (config) #
```

Рисунок 3.4 – Перевірка налаштування AAA

На рисунку 3.5 наведено список користувачів та їх дані.

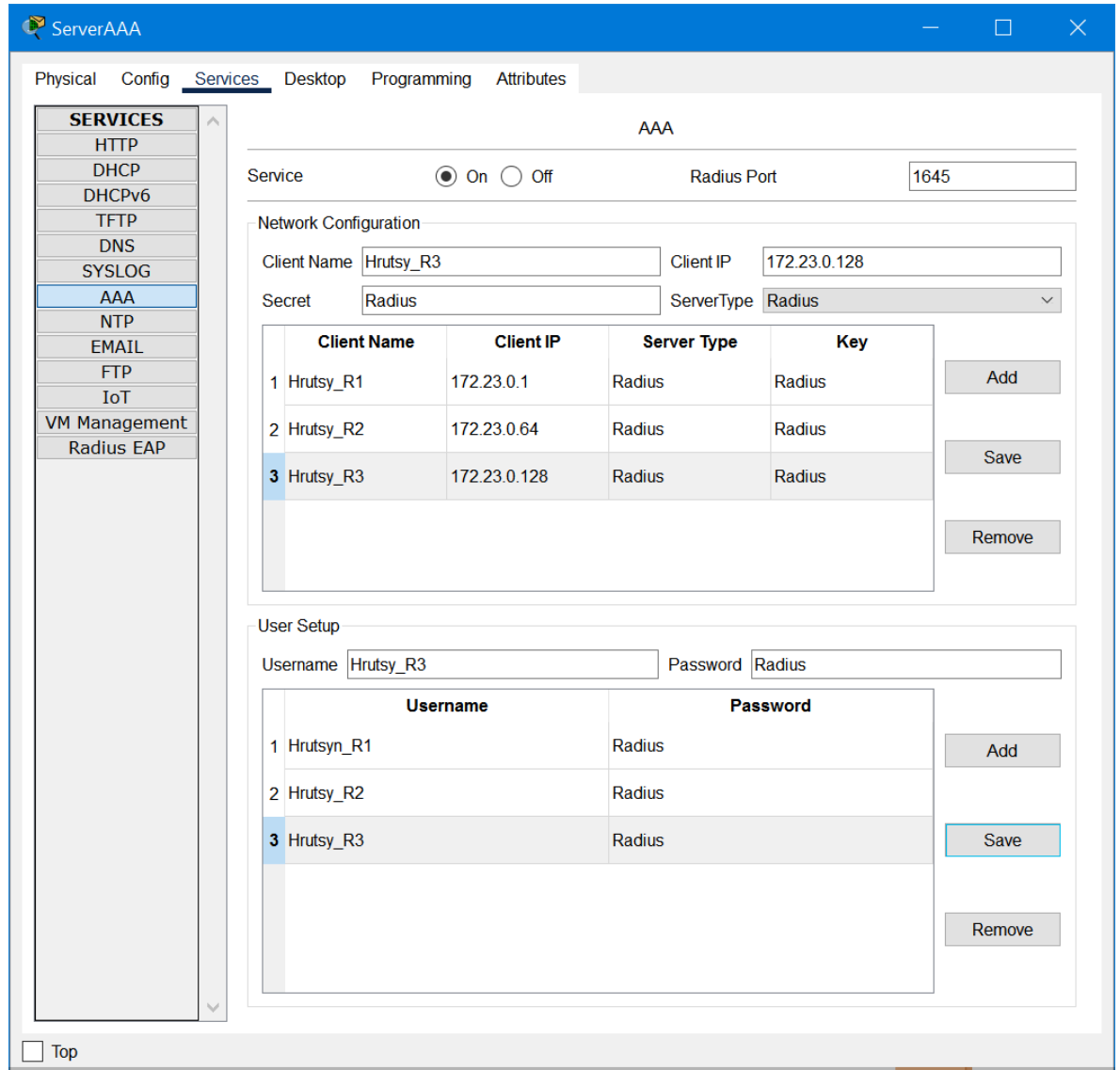


Рисунок 3.5 – Параметри сервера AAA

3.4.3 Налаштування роботи Інтернету в КС ТОВ «Дніпровський трубний завод»

Згідно вимог для розгортання корпоративної мережі, використовується блок адрес з приватних адрес, що належать до певного діапазону. З метою забезпечення доступу робочих станцій організації до Інтернету, потрібно використовувати технологію NAT на прикордонному маршрутизаторі.

NAT, або Network Address Translation, є технологією, яка використовується для перетворення IP-адрес між різними мережевими доменами. Основна мета NAT полягає в забезпеченні доступу до Інтернету для приватних IP-адрес, які

самостійно не можуть маршрутизуватися через Інтернет [6].

Застосування технології NAT дозволяє робочим станціям з приватними IP-адресами комунікувати зі світовою мережею, перетворюючи їхні адреси на публічні IP-адреси, які використовуються в Інтернеті. Це дозволяє приватним мережам ефективно використовувати доступні публічні IP-адреси, сприяє забезпеченню безпеки та конфіденційності мережі, а також допомагає зменшити витрати на придбання додаткових публічних IP-адрес.

Таким чином, застосування технології NAT на прикордонному маршрутизаторі дозволить реалізувати доступ до Інтернету для робочих станцій організації, які використовують приватні IP-адреси.

```
Hrutsyn_R4(config)#access-list 14 permit 172.23.0.0 0.0.7.255
```

```
Hrutsyn_R4(config)#ip nat pool Internet 209.165.202.5 209.165.202.30 netmask 255.255.255.224
```

```
Hrutsyn_R4(config)#ip nat inside source list 14 pool Internet
```

```
Hrutsyn_R4(config)#ip nat inside source static 172.23.0.46 209.165.202.4
```

```
Hrutsyn_R4(config)#int se0/1/0
```

```
Hrutsyn_R4(config-if)#ip nat outside
```

```
Hrutsyn_R4(config-subif)#in se0/2/0
```

```
Hrutsyn_R4(config-if)#ip nat inside
```

Перевіряємо налаштування та виводимо таблицю NAT (рис.3.6).

NAT Table for Pasichna_Router_4				
Protocol	Inside Global	Inside Local	Outside Local	Outside Global
icmp	209.165.202.7:5	172.23. 0.35:5	209.165.201.5:5	209.165.201.5:5
icmp	209.165.202.7:6	172.23.0 .35:6	64.100.13.2:6	64.100.13.2:6
icmp	209.165.202.5:8	172.23.0 .45:7	64.100.13.2:8	64.100.13.2:8

Рисунок 3.6 – Таблиця NAT

3.4.3 Налаштування VPN в КС ТОВ «Дніпровський трубний завод»

Використання віртуальної приватної мережі site-to-site VPN дозволяє організації інтегруватися з наявною мережевою інфраструктурою. Це надає можливість забезпечити безпечний обмін даними між різними підрозділами підприємства, навіть якщо вони знаходяться на віддалених місцях.

Віртуальна приватна мережа (VPN) site-to-site дозволяє створити захищений тунель між різними мережами організації, що забезпечує конфіденційність та цілісність передаваних даних. Цей тунель дозволяє віддаленим підрозділам підприємства підключатися до центральної мережі безпосередньо, як будто вони фізично знаходяться в одному місці.

Завдяки використанню віртуальної приватної мережі site-to-site VPN, організація може розширити свою мережу, об'єднати різні відділи та підрозділи в єдину комунікаційну інфраструктуру. Це сприяє полегшенню обміну даними, спільній роботі та забезпечує централізований контроль над мережею.

Використання віртуальної приватної мережі site-to-site VPN є особливо корисним для організацій, які мають розподілену інфраструктуру з віддаленими підрозділами, філіями або віддаленими робочими місцями. Це дозволяє забезпечити безпеку та зручність спілкування між цими місцями, а також ефективно використовувати ресурси мережі.

```
Hrutsyn_R4(config)#ip access-list extended 100
```

```
Hrutsyn_R4(config-ext-nacl)#permit ip any 209.165.200.0 0.0.0.31
```

```
Hrutsyn_R4(config-ext-nacl)#permit ip 172.23.0.0 0.0.7.255 209.165.202.0
0.0.0.31
```

```
Hrutsyn_R4(config-ext-nacl)#permit ospf any any
```

```
Hrutsyn_R4(config)#ip access-list extended VPN
```

```
Hrutsyn_R4(config-ext-nacl)#permit ip 172.23.02.0 0.0.7.255 172.23.153.0
0.0.0.31
```

```
Hrutsyn_R4(config)#crypto isakmp policy 1
```

```
Hrutsyn_R4(config-isakmp)#encryption aes 256
```

```

Hrutsyn_R4(config-isakmp)#authentication pre-share
Hrutsyn_R4(config-isakmp)#group 1
Hrutsyn_R4(config)#crypto isakmp key cisco address 64.100.13.3
Hrutsyn_R4(config)#crypto ipsec transform-set VPN-IPSEC-SET esp-aes esp-
sha-hmac
Hrutsyn_R4(config-crypto-map)#set peer 64.100.13.3

```

3.4.3 Налаштування агрегування каналів в відділі якості КС ТОВ «Дніпровський трубний завод»

Шляхом використання технології EtherChannel, були створені агреговані з'єднання фізичних портів на комутаторах в мережі відділі якості з метою підвищення пропускної здатності та надійності каналів. Це дозволяє об'єднати кілька фізичних портів в один логічний канал, що принесло кілька значних переваг.

Головною перевагою такого об'єднання є збільшення швидкості передачі даних завдяки паралельному використанню кількох портів одночасно. Це дозволяє досягти високої пропускної здатності та забезпечити рівномірне розподілення навантаження на доступні порти. Крім того, в разі відмови одного з портів, інші порти продовжують працювати без перебоїв, що значно покращує надійність і безперебійність каналу.

Використання технології EtherChannel [7] сприяє оптимальному використанню ресурсів мережі та забезпечує ефективну передачу даних. Збільшена швидкість передачі та забезпечення надійності каналу робить її ідеальним рішенням для вимогливих застосунків і високонавантажених середовищ. Балансування навантаження і збільшення пропускної здатності створюють оптимальні умови для безперебійного функціонування мережі та задоволення потреб користувачів.

```

Hrutsyn_S_1(config)#int range fa0/2-3, fa0/6-7
Hrutsyn_S_1(config-if-range)#switchport mode trunk

```

Налаштування EtherChannel за допомогою протоколу Cisco PAgP на прикладі команд на комутаторі Hrutsyn_S_2:

```
Hrutsyn_S_2(config)#int range fa0/6-7
Hrutsyn_S_2(config-if-range)#sh
Hrutsyn_S_2(config-if-range)#channel-group 3 mode desirable
Hrutsyn_S_2(config-if-range)#
Creating a port-channel interface Port-channel 3
Hrutsyn_S_2(config-if-range)#no sh
Hrutsyn_S_2(config-if-range)#exit
Hrutsyn_S_2(config)#int port-channel 3
Hrutsyn_S_2(config-if)#switchport mode trunk
Hrutsyn_S_3(config-if)#int range fa0/4-5
Hrutsyn_S_ (config-if-range)#sh
Hrutsyn_S_ (config-if-range)#channel-group 2 mode passive
Hrutsyn_S_ (config-if-range)#
Creating a port-channel interface Port-channel 2
Hrutsyn_S_ (config-if-range)#no sh
Hrutsyn_S_ (config-if-range)#exit
Hrutsyn_S_ (config)#int port-channel 2
Hrutsyn_S_ (config-if)#switchport mode trunk
```

Вводимо в CLI команду *show etherchennal summary* та перевіряємо налаштування PAgP та LACP на комутаторах (рис.3.7) [8].

```
Hrutsyn_S_2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Hrutsyn_S_2(config)#do sh eth sum
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)         PAgP       Fa0/1(P) Fa0/2(P)
3      Po3(SD)         PAgP       Fa0/3(I) Fa0/4(I)
Hrutsyn_S_2(config)#
```

Рисунок 3.7 – Налаштування порт-каналів на комутаторі

3.4.4 Налаштування динамічного розподілу адрес

Налаштування динамічного розподілу адрес включає в себе використання протоколу DHCP (Dynamic Host Configuration Protocol). DHCP є механізмом, який автоматично призначає IP-адреси та інші мережеві параметри пристроям у мережі.

Для налаштування DHCP необхідно налаштувати DHCP-сервер, який буде надавати адреси пристроям, що підключаються до мережі. Клієнти, які потребують IP-адресу, можуть надсилати запити до сервера, і він відповідає, надаючи необхідні параметри мережі.

Таким чином, використання протоколу DHCP спрощує процес налаштування мережі, забезпечує ефективне використання адресного простору та забезпечує зручність управління мережевими налаштуваннями.

На рисунку 3.8 та рис.3.9 наведено фрагмент розподілу IP-адрес для під мережі «Відділ розробки та проектування» (розтер 2) та під мережі «Відділ продажів та маркетингу» (розтер 1)ю

```
Hrutsyn_R2#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Hrutsyn_R2(config)#interface GigabitEthernet0/1
Hrutsyn_R2(config-if)#ip address 172.23.0.129 255.255.255.128
Hrutsyn_R2(config-if)#ip dhcp pool Project
Hrutsyn_R2(dhcp-config)#network 172.23.0.128 255.255.255.128
Hrutsyn_R2(dhcp-config)#default-router 172.23.0.129
Hrutsyn_R2(dhcp-config)#dns-server 172.23.0.130
Hrutsyn_R2(dhcp-config)#ip dhcp ex 172.23.0.129 172.23.0.131
Hrutsyn_R2(config)#
Hrutsyn_R2(config)#exit
```

Рисунок 3.8 – Динамічний розподіл адрес кінцевим користувач для «Відділу розробки та проектування»

```
-----
Hrutsyn_R1(config)#interface GigabitEthernet0/1
Hrutsyn_R1(config-if)#ip address 10.22.209.1 255.255.255.0
Hrutsyn_R1(config-if)#ip address 172.23.1.1 255.255.255.0
Hrutsyn_R1(config-if)#ip address 172.23.1.1 255.255.255.192
Hrutsyn_R1(config-if)#ip address 172.23.1.1 255.255.255.192
Hrutsyn_R1(config-if)#ip dhcp pool LAN_Marketing
Hrutsyn_R1(dhcp-config)#network 172.23.1.0 255.255.255.192
Hrutsyn_R1(dhcp-config)#default-router 172.23.1.1
Hrutsyn_R1(dhcp-config)#dns-server 172.23.1.2
Hrutsyn_R1(dhcp-config)#ip dhcp ex 172.23.1.1 172.23.1.4
Hrutsyn_R1(config)#
```

Рисунок 3.9 – Динамічний розподіл адрес кінцевим користувач для

«Відділу продажів та маркетингу»

На рисунку 3.10 та рис.3.11 наведено результат роботи протоколу DHCP.

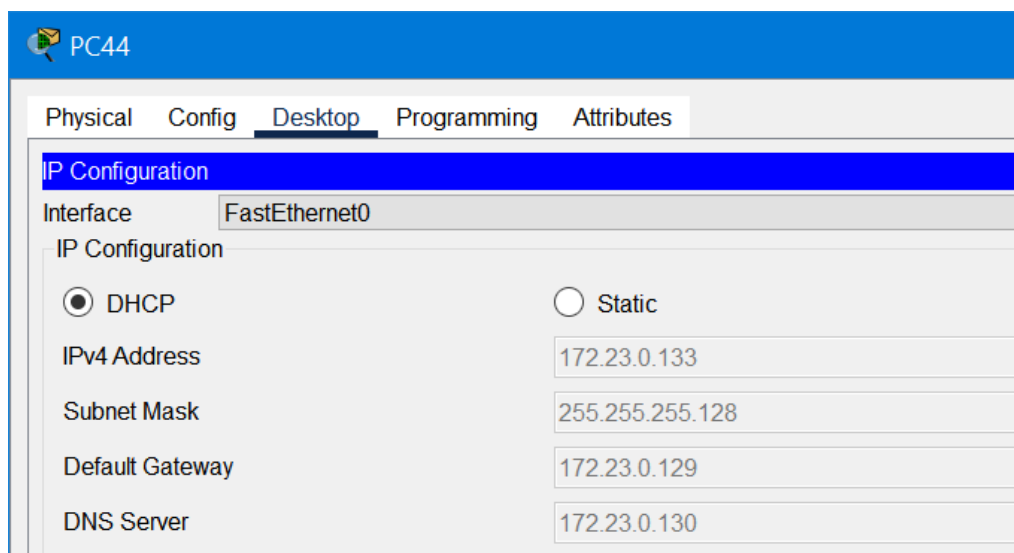


Рисунок 3.10 – Результат налаштування DHCP для «Відділу продажів та маркетингу»

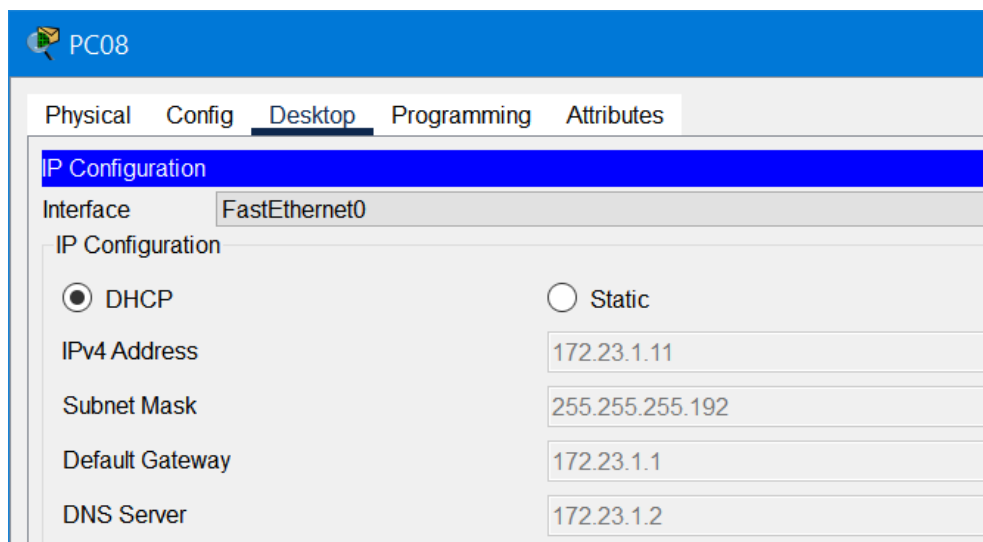


Рисунок 3.11 – Результат налаштування DHCP для «Відділу продажів та маркетингу»

3.5 Налаштування VLAN в під мережі «Відділ закупівель»

Таблиця 3.3 містить розподіл підмережі на віртуальні локальні мережі (VLAN).

Таблиця 3.3 – Розподіл VLAN для підмережі

Номер VLAN	Ім'я VLAN	Примітка
1	Default	Не використовується
16	VLAN16	Підрозділ стратегічних закупівель
26	VLAN26	Підрозділ операційних закупівель
36	VLAN36	Підрозділ стратегічного аналізу та контролю закупівель
99	Management	Для пристроїв
100	Native	Власна

Відділ закупівель поділено на три підрозділи (рис.3.12):

- підрозділ стратегічних закупівель. Відповідає за визначення стратегії закупівель, укладення довгострокових угод та співпрацю з ключовими постачальниками. Його головною метою є забезпечення стабільного постачання важливих ресурсів та матеріалів для компанії;
- підрозділ операційних закупівель: Відповідає за забезпечення поточних потреб компанії в ресурсах та матеріалах. Цей підрозділ відповідає за укладення короткострокових угод, переговори з постачальниками та здійснення закупівельних операцій;
- підрозділ стратегічного аналізу та контролю закупівель: Відповідає за аналіз ефективності закупівельних процесів, виявлення потенційних змін та вдосконалення стратегій закупівель. Цей підрозділ забезпечує контроль над виконанням угод та забезпечує оптимальне використання ресурсів компанії.

```
Hrutsyn_SwV0#sh vlan brief
```

VLAN Name	Status	Ports
1 default	active	Fa0/1, Fa0/2, Fa0/3, Fa0/4 Gig0/2
16 VLAN_16	active	Fa0/15, Fa0/16, Fa0/17, Fa0/18 Fa0/19, Fa0/20, Fa0/21, Fa0/22
26 VLAN_26	active	Fa0/10, Fa0/11, Fa0/12, Fa0/13 Fa0/14
36 VLAN_36	active	Fa0/5, Fa0/6, Fa0/7, Fa0/8 Fa0/9
99 Management	active	
100 Native	active	
1002 fddi-default	active	
1003 token-ring-default	active	
1004 fddinet-default	active	
1005 trnet-default	active	

```
Hrutsyn_Sw2#sh vlan brief
```

VLAN Name	Status	Ports
1 default	active	Fa0/1, Fa0/2, Fa0/3, Fa0/4 Gig0/1, Gig0/2
16 VLAN_16	active	Fa0/15, Fa0/16, Fa0/17, Fa0/18 Fa0/19, Fa0/20, Fa0/21, Fa0/22 Fa0/24
26 VLAN_26	active	Fa0/10, Fa0/11, Fa0/12, Fa0/13 Fa0/14
36 VLAN_36	active	Fa0/5, Fa0/6, Fa0/7, Fa0/8 Fa0/9
99 Management	active	
100 Native	active	
1002 fddi-default	active	
1003 token-ring-default	active	
1004 fddinet-default	active	
1005 trnet-default	active	

```
Hrutsyn_Sw2#
```

```
Hrutsyn_Sw1#sh vlan brief
```

VLAN Name	Status	Ports
1 default	active	Fa0/1, Fa0/2, Fa0/3, Fa0/4 Gig0/1, Gig0/2
16 VLAN_16	active	Fa0/15, Fa0/16, Fa0/17, Fa0/18 Fa0/19, Fa0/20, Fa0/21, Fa0/22 Fa0/23
26 VLAN_26	active	Fa0/10, Fa0/11, Fa0/12, Fa0/13 Fa0/14
36 VLAN_36	active	Fa0/5, Fa0/6, Fa0/7, Fa0/8 Fa0/9
99 Management	active	
100 Native	active	
1002 fddi-default	active	
1003 token-ring-default	active	
1004 fddinet-default	active	
1005 trnet-default	active	

```
Hrutsyn_Sw1#
```

Рисунок 3.12– Результат налаштувань VLAN на комутаторах відділу закупівель

3.6 Перевірка роботи комп'ютерної системи

Для перевірки працездатності внутрішньої мережі було використано команду ping, яка відправила ехо-запит з мережі «Відділ дослідження та розвитку» до підмережі «Відділ розробки та проектування» (рис 3.13).

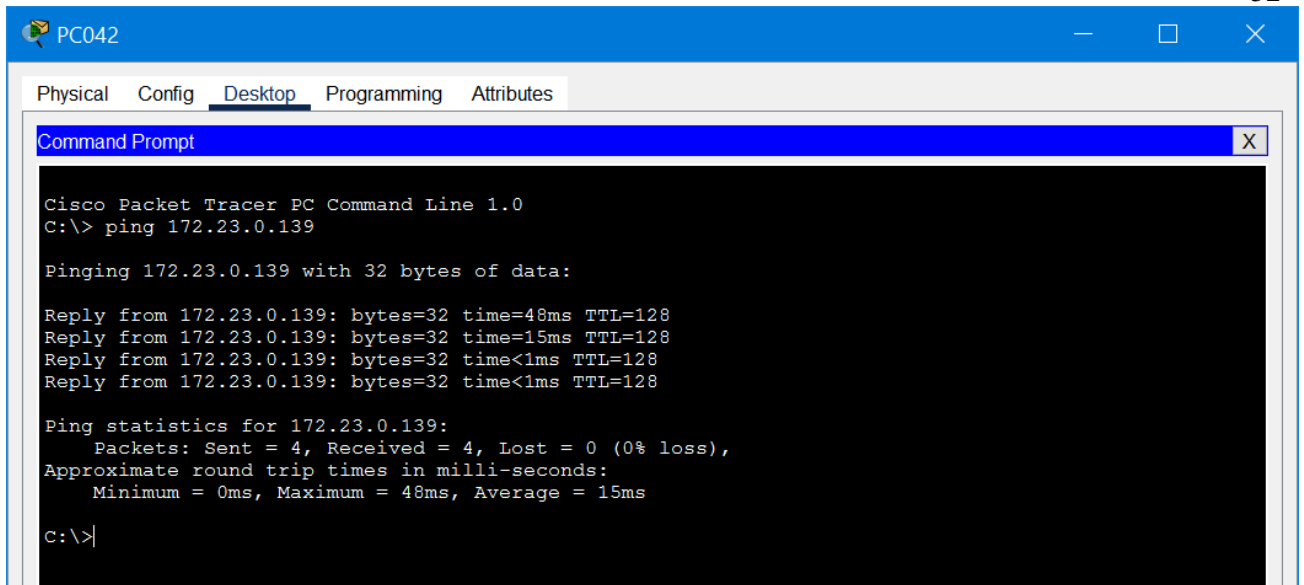


Рисунок 3.13 – Тестування мережі

На рис.3.14 наведено перевірку роботи відділеної мережі.

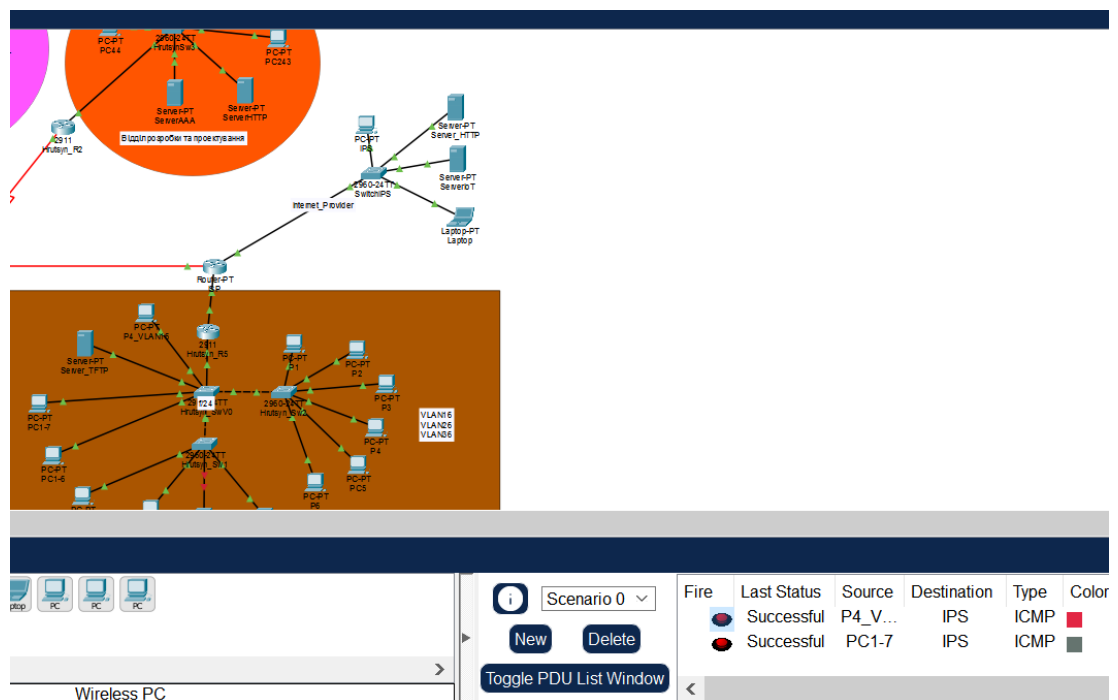


Рисунок 3.14 – Тестування відділеної мережі

4 РОЗРОБКА КОМПОНЕНТА СИСТЕМИ МОНІТОРИНГУ РІВНЯ КИСНЮ ТОВ ДНІПРОВСЬКИЙ ТРУБНИЙ ЗАВОД

4.1 Конфігурація апаратного забезпечення та налаштування сервісів у системі Інтернету речей

Даний розділ присвячений опису процесу налаштування апаратних компонентів та сервісів в рамках системи Інтернету речей (IoT) використовуючи Cisco Packet Tracer [9].

В рамках даного розділу, були реалізовані основні функціональні компоненти:

- автоматичне обприскування води;
- автоматизоване освітлення;
- сирена;
- сонячну панель для генерації електроенергії;
- систему моніторингу киснем.

Конфігурація апаратного забезпечення та налаштування сервісів були виконані з використанням Cisco Packet Tracer, що дозволило симулювати та перевірити роботу цих компонентів в імітаційному середовищі [10].

Для автоматичного поливу було встановлено відповідне апаратне обладнання, таке як датчики вологості, насоси та клапани. Застосовано відповідні налаштування, які дозволяють визначати рівень вологості ґрунту та автоматично активувати полив для забезпечення оптимальних умов росту та розвитку рослин.

Для автоматизованого освітлення використовувалися датчики руху та світла, які контролюють присутність людей та рівень освітленості в приміщенні. На основі цих параметрів було налаштовано освітлювальні

прилади, щоб забезпечити оптимальне освітлення відповідно до потреб користувачів.

Сирена в системі IoT використовується для негайного сповіщення про небезпеку або надзвичайну ситуацію. Було налаштовано відповідний сигнальний пристрій та встановлено вимоги щодо активації сирени в разі виявлення небезпеки [11].

Конфігурація апаратного забезпечення та налаштування сервісів в системі IoT виконані з метою забезпечення оптимальної та безпечної роботи цих функціональних компонентів. Цей розділ надає детальний опис процесу налаштування та пояснює взаємодію між апаратними пристроями та сервісами у системі IoT.

У системі було використано сонячну панель, яка генерує електроенергію і надсилає її до акумулятора. Цей акумулятор служить для зберігання електроенергії. За допомогою підключеного між ними лічильника електроенергії, енергія, зібрана сонячною панеллю, вимірюється і реєструється.

Після цього, розподілення енергії відбувається з батареї до підключених пристроїв. Батарея віддає збережену електроенергію відповідно до потреб і вимог підключених пристроїв, що забезпечує живлення цих пристроїв. Таким чином, енергія розподіляється з акумулятора до підключених пристроїв у системі автоматизованого поливу та освітлення.

У комп'ютерній системі також присутня система моніторингу рівня кисню. Ця система призначена для постійного контролю рівня кисню в приміщеннях. Вона використовує спеціальні датчики, які постійно вимірюють рівень кисню в повітрі.

Отримані дані з датчиків передаються до центрального контрольного пристрою або системи збору даних, де вони аналізуються і відображаються в

реальному часі. Це дозволяє операторам системи вчасно виявляти будь-які зміни рівня кисню і приймати необхідні заходи.

Система моніторингу кисню може бути підключена до централізованої системи управління, яка автоматично реагує на відхилення рівня кисню і активує відповідні заходи, такі як підвищення подачі свіжого повітря або сповіщення персоналу про необхідність втручання.

Ця система моніторингу кисню є важливим компонентом системи безпеки та комфорту в приміщеннях, особливо в умовах, де належна концентрація кисню має значення для здоров'я та безпеки людей.

Структурна схема IoT пристроїв ТОВ «Дніпровський трубний завод» представлена на рисунку 4.1.

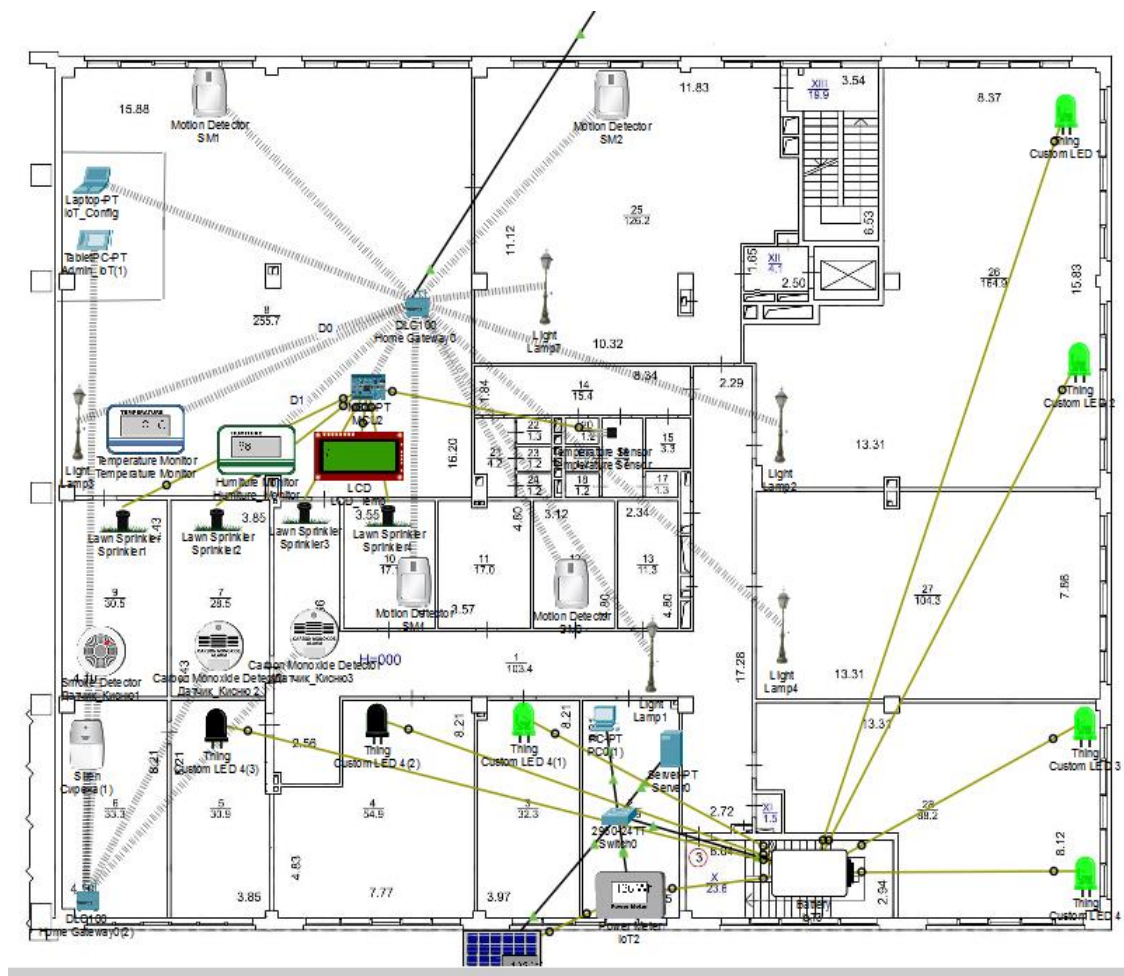


Рисунок 4.1 – Структурна схема IoT датчиків

Комп'ютерна система використовує технології IoT хмарних обчислень. У якості мережного шлюзу для системи використовується маршрутизатор DLC100. Для реалізації хмарних обчислень використовуються "розумні пристрої", такі як датчик кисню, сонячна панель, розетки, тощо.

Запропонована структура розміщення пристроїв та використання IoT технологій дозволяє ефективно виявляти та реагувати на небезпечні ситуації, забезпечуючи безпеку та захист приміщення.

4.2 Конфігурація апаратного забезпечення та налаштування сервісів у системі Інтернету речей

Вибір датчиків для різних систем в системі "інтернету речей" (IoT) залежить від їх призначення та вимог щодо функціональності. Давайте розглянемо кожен систему окремо.

При виборі датчиків для автоматичного поливу необхідно враховувати параметри ґрунту та рослин, що вирощуються. Датчики вологості ґрунту здатні вимірювати рівень вологості та вчасно активувати полив, коли ґрунт стає сухим. Також можуть бути використані датчики дощу для вимірювання кількості опадів та регулювання режиму поливу.

Для автоматизованого освітлення можуть використовуватися датчики руху та освітленості. Датчики руху реагують на присутність людини в приміщенні та активують освітлення. Датчики освітленості вимірюють рівень освітленості і дозволяють регулювати яскравість освітлення в залежності від зовнішніх умов.

Система сирени використовує датчики руху або датчики виявлення звуку. Датчики руху сприймають рух або виключні події, тоді як датчики звуку реагують на підвищений рівень шуму. При спрацюванні сигналу з датчика, сирена активується для відповідного сповіщення або виклику уваги.

Для генерації електроенергії з сонячної панелі, використовуються фотоелектричні датчики, які конвертують сонячне випромінювання в електричну енергію. Ці датчики сприймають фотони світла та генерують електричний струм, який потім може бути використаний для живлення системи або зберігатися у акумуляторі.

Для системи моніторингу кисню можуть використовуватися електрохімічні датчики кисню, які здатні вимірювати концентрацію кисню у повітрі. Ці датчики виявляють наявність кисню та вимірюють його рівень. Вони можуть бути використані для моніторингу рівня кисню у різних середовищах, наприклад, у воді або у повітрі. Інформація, отримана від цих датчиків, може бути використана для контролю якості повітря, води або для виявлення можливих проблем, пов'язаних з недостатнім рівнем кисню.

Вибір датчиків для кожної системи залежить від конкретних вимог та потреб проекту. Наприклад, враховуються тип пристрою, середовище, в якому він буде використовуватися, точність вимірювання, надійність та інші фактори. Оптимальний вибір датчиків допомагає забезпечити ефективну та надійну роботу системи "Інтернету речей".

Комунікація між пристроями здійснюється за допомогою бездротової технології WiFi, що забезпечується маршрутизатором DLC100. Для керування мережею та забезпечення доступу до веб-інтерфейсу системи безпеки, користувачі налаштували параметри MCU2 та IoT-сервера.

MCU2 відповідає за управління під'єднаними пристроями та надає розподіл IP-адрес з приватного адресного блоку 192.168.26.10-192.168.26.124 за допомогою протоколу DHCP. Це дозволяє автоматично призначати IP-адреси пристроям у мережі, спрощуючи процес налаштування та забезпечуючи їхню взаємодію (табл.4.1).

Таблиця 4.1 – Мережні налаштування домашнього шлюзу

Параметр	Значення
IP-адреса домашнього шлюзу	192.168.26.10
Маска домашньої підмережі	255.255.255.0
SSID бездротової домашньої мережі	GazonP
Метод автентифікації	WPA2-PSK AES
Ключ автентифікації (<i>пароль</i>)	HrutsynKey

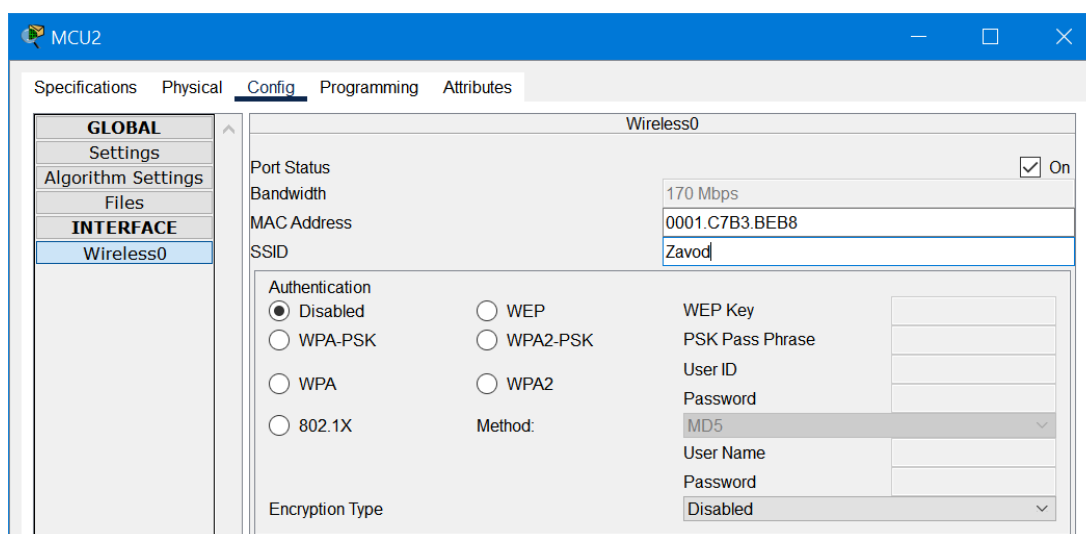


Рисунок 4.1 – Параметри налаштування MCU2

В якості IoT-сервера використовується сервер провайдера зі зазначеним IP-адресою 1.0.0.1/24. На головній сторінці веб-сайту сервера можна побачити список доступних IoT-пристроїв, для кожного з яких є можливість віддаленого керування (увімкнення/вимкнення) або спостереження.

Адміністратор через інтерфейс для управління IoT-пристроями з використання веб технологій слідкує за датчиками. Результат наведено на рис.4.2 та рис.4.3.

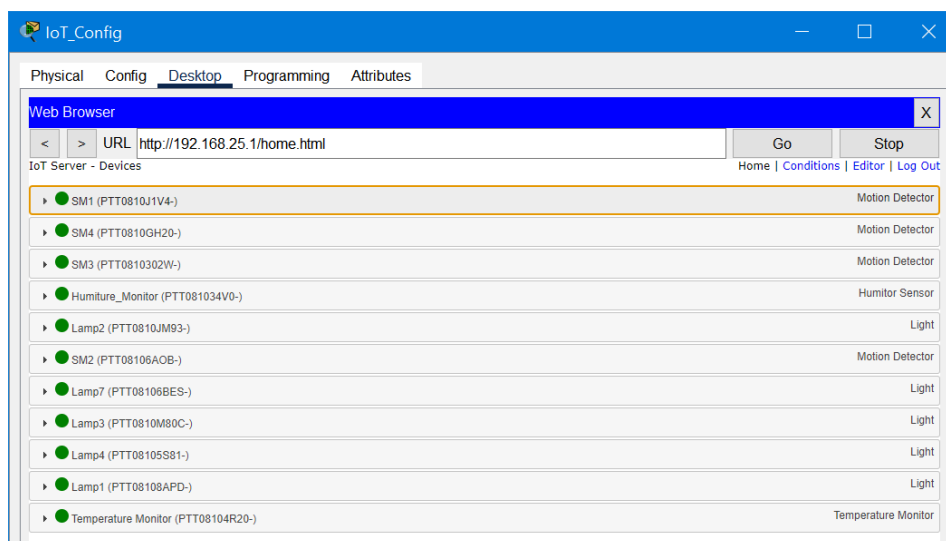


Рисунок 4.2 – Параметри налаштування IoT-пристроїв

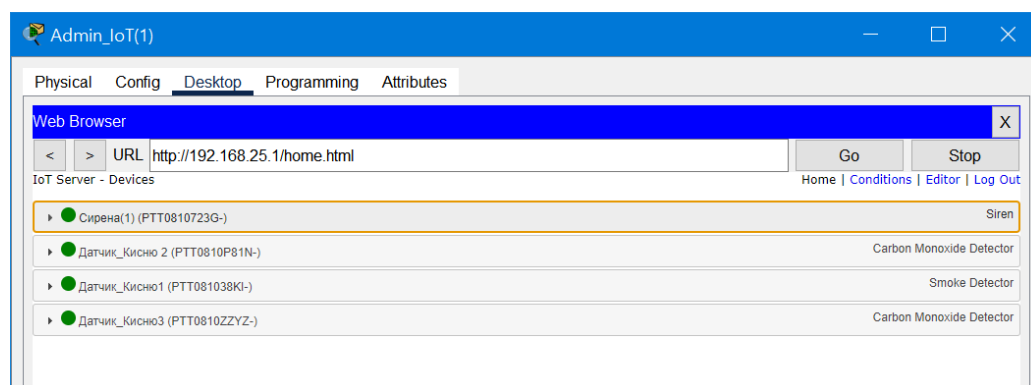


Рисунок 4.3 – Параметри налаштування IoT-пристроїв для системи кисню

4.3 Налаштування функціонування датчиків Інтернету речей

За допомогою веб-інтерфейсу сервера Інтернету речей сконфігуровано сценарій для управління системою кисню. На рисунку 4.4 наведено сценарій датчиків для системи моніторингу рівня кисню.

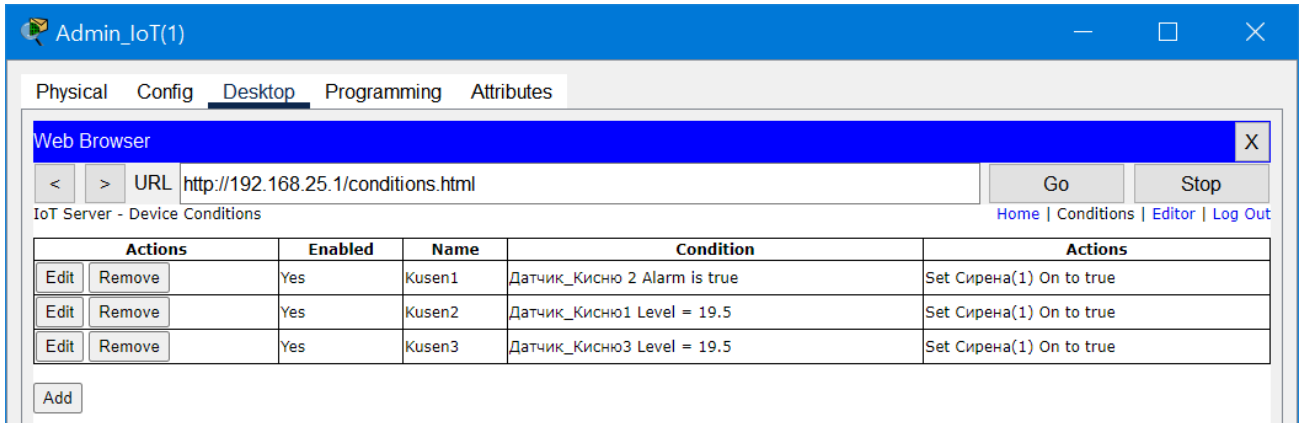


Рисунок 4.4 – Сценарії для системи моніторингу киснем

Алгоритм роботи:

Спочатку необхідно налаштувати датчик кисню, підключений до системи. Це включає встановлення правильних параметрів датчика, таких як мінімальний та максимальний рівні кисню, порогові значення, тривалість вимірювань тощо.

Після налаштування датчика кисню можна запустити процес моніторингу. Це включає вимірювання рівня кисню в приміщенні та збереження цих даних для подальшого аналізу.

Отримані дані про рівень кисню підлягають аналізу для виявлення будь-яких аномалій або відхилень від прийнятих нормативів. Це може включати порівняння зі стандартами безпеки, встановлення тривалості перебування в безпечному діапазоні кисню тощо.

Якщо виявлено відхилення від нормативів або небезпечний рівень кисню, система може автоматично сповістити відповідні особи або персонал через різні канали сповіщень, наприклад, електронну пошту, SMS-повідомлення, веб-систему чи звукові сигнали. Не допустимий рівень кисню для життя людини відноситься до значень, коли концентрація кисню в повітрі впадає нижче 19.5 відсотка об'єму. Це вважається критично низьким рівнем

кисню, який може призвести до кисневого голоду або задухи. У таких умовах людина може відчувати нездужання, погіршення концентрації, втому, запаморочення та інші симптоми. Оптимальний рівень кисню для життя людини становить близько 20.9 відсотка об'єму повітря. Проте, слід зазначити, що точні значення можуть варіюватися залежно від індивідуальних факторів, таких як вік, стан здоров'я та інші умови.

Дані про моніторинг рівня кисню можуть бути записані для подальшої звітності та аналізу. Це може бути корисним для ведення статистики, виявлення тенденцій або використання в подальших дослідженнях.

На рисунку 4.5 наведено сценарій датчиків для системи ввімкнення освітлення, сирени (свіщення важливої інформації), поливу.

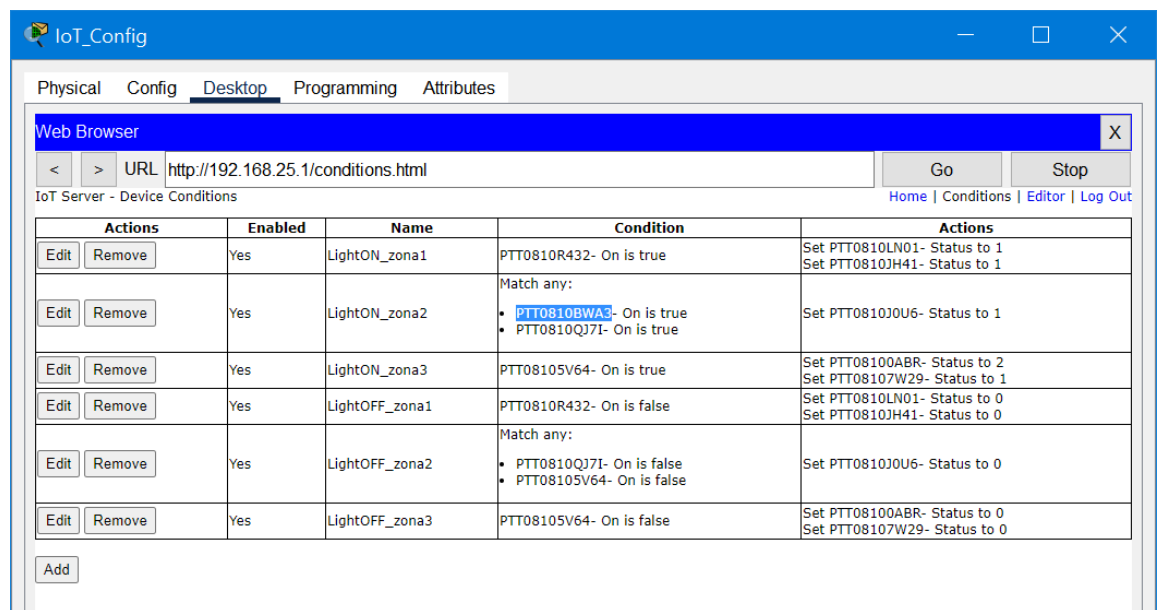


Рисунок 4.5 – Сценарії для системи освітлення, сповіщення

Третій блок це система електроенергії на основі сонячної панелі. сонячна панель виконує функцію зарядки. Вона отримує енергію від сонця і перетворює її на електричну енергію. Ця електроенергія подається до акумулятора для зберігання.

Для вимірювання електричної енергії, яку виробляє сонячна панель, в системі встановлений лічильник електроенергії. Цей лічильник зчитує електричну потужність, яку виробляє сонячна панель, і реєструє ці дані.

Також в системі присутня батарея, яка відповідає за розподіл енергії. Вона отримує електроенергію з акумулятора і розподіляє її між підключеними пристроями. Це дозволяє забезпечити живлення цих пристроїв.

Реєстраційний сервер виконує функцію зчитування потужності, виробленої сонячною панеллю, потужності, яку вимірює лічильник електроенергії, і потужності, яка доступна на акумуляторі. Він отримує ці дані та реєструє їх для подальшого аналізу та моніторингу роботи системи.

Ця система дозволяє ефективно використовувати енергію сонця, забезпечувати живлення підключених пристроїв та здійснювати моніторинг енергетичних показників.

На рис.4.6 наведено статус заряду електромережі.

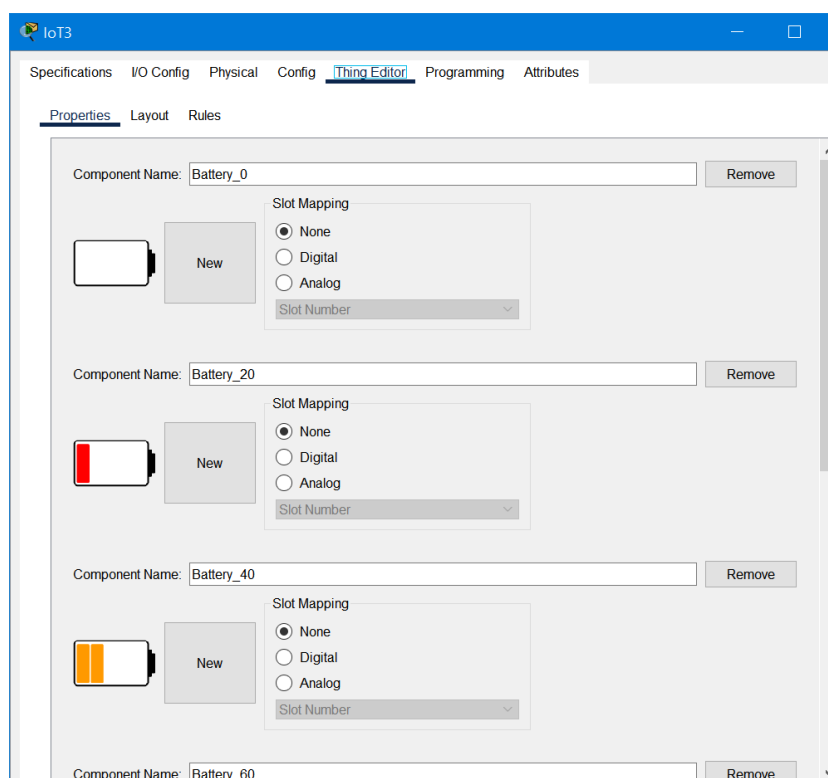


Рисунок 4.6 – Стан електромережі в КС

Для реалізації роботи датчиків використано Python, який представляє систему керування енергопостачанням пристрою. Він використовує різні модулі та функції для керування електричними сигналами та моніторингу рівня доступної енергії. Ось опис коду:

Код починається з імпорту необхідних модулів: `gpio`, `time`, `ioeclient`, `environment`, `physical`. Ці модулі надають функціональні можливості, пов'язані з керуванням GPIO, управлінням часом, комунікацією з IoT-клієнтом, моніторингом оточення та операціями з фізичними пристроями:

```
from gpio import *  
  
from time import *  
  
from ioeclient import *  
  
from environment import *  
  
from physical import *  
  
import math
```

Визначаються декілька глобальних змінних для зберігання значень конфігурації та інформації про енергопостачання.

Функція `setup()` відповідає за ініціалізацію системи. Вона налаштовує IoT-клієнта, визначає тип пристрою та його стани, та отримує початковий рівень доступної енергії. Також вона надсилає звіт на сервер.

```
def setup():  
  
    setImage('100')  
  
    IoEClient.setup({  
  
        "type": "Battery",
```

```
"states": [  
  {  
    "name": "Available power",  
    "type": "number",  
    "unit": '%',  
    "controllable": False  
  }  
]  
})
```

Функція `onInputReceiveDone()` є зворотним викликом, який спрацьовує при отриманні вхідних даних. Вона обробляє отримані вхідні дані.

Функція `restoreProperty()` використовується для відновлення властивості пристрою з сервера. Вона отримує значення вказаної властивості та повертає його. Якщо властивість не знайдена, повертається задане значення за замовчуванням.

Функція `main()` є головною точкою входу в скрипт. Вона спочатку викликає функцію `setup()`, а потім увіймає в безкінечний цикл.

У межах циклу обчислюється рівень доступної енергії у відсотках, і на основі цього значення встановлюється відповідне зображення, що відображає стан акумулятора.

```
INPUT_PIN = 0
```

```
OUTPUT_PIN = 1
```

```
MAX_STORED_POWER = 1500.0 #2000 watts
```



```

MULTIPLIER = float(255)/1023

LOG_BASE = 1.0749111034571373359815489867558

availablePowerPercent = 0.0

availablePower = float(MAX_STORED_POWER)/2.0

receivedPower = 0

AVAIL_POWER_PROP="Available power (%)"

CHARGE_RATE = 19

LOOP_DELAY = 500

```

Функція `sendReport()` надсилає звіт на сервер про поточний стан пристрою та рівень доступної енергії.

Функція `receivePower()` обчислює отриману енергію на основі аналогового сигналу. Вона застосовує логарифмічне перетворення та обмежує отриману енергію до максимальної швидкості зарядки. Отримана енергія додається до рівня доступної енергії.

Функція `sendPower()` розподіляє доступну енергію між підключеними пристроями. Вона отримує інформацію про споживану потужність для кожного пристрою, підключеного до певного слоту, та перевіряє, чи є достатньо доступної енергії для живлення пристрою. Якщо так, відповідна кількість енергії виводиться на пристрій. Якщо доступної енергії недостатньо, на пристрій виводиться мінімальна кількість енергії, що вказує на нестачу ресурсу.

```
(float(availablePower)/MAX_STORED_POWER) * 100
```

```
if availablePowerPercent < 10:
```

```
    setImage('Battery_0')
```

```
elif availablePowerPercent < 30:
```

```
    setImage('Battery_20')  
  
elif availablePowerPercent < 50:  
    setImage('Battery_40')  
  
elif availablePowerPercent < 70:  
    setImage('Battery_60')  
  
elif availablePowerPercent < 90:  
    setImage('Battery_80')  
  
elif availablePowerPercent <= 100:  
    setImage('Battery_100')  
  
  
sendReport()  
  
delay(LOOP_DELAY)  
  
receivePower(LOOP_DELAY*3)  
  
sendPower()  
  
delay(LOOP_DELAY*2)
```

На завершення, код переконується, що рівень доступної енергії не становить від'ємного значення і є обмеженим до нуля, якщо він менше нуля.

На рис.4.7 наведено програмування датчика вимірювання рівня кисню

```

Датчик_Кисню 2
Specifications I/O Config Physical Config Thing Editor Programming Attributes
Carbon Monoxide Detector (Python) - pyjs.py
Open New Delete Rename Import Run Clear Outputs Help
Reload Copy Paste Undo Redo Find Replace Zoom: + -
..
main.py
pyjs.py
5 from time import *
6 from ioclient import *
7 from environment import *
8 from physical import *
9
10 INPUT_PIN = 0
11 OUTPUT_PIN = 1
12 MAX_OXYGEN_LEVEL = 100.0 # Максимальний рівень кисню в системі (у відсотках)
13 MIN_OXYGEN_LEVEL = 19.5 # Мінімальний припустимий рівень кисню в системі (у відсотках)
14 oxygenLevel = 0.0 # Поточний рівень кисню в системі (у відсотках)
15 oxygenSensorPin = 0 # Пін для підключення датчика кисню
16
17 def setup():
18     IoClient.setup({
19         "type": "Oxygen Monitor",
20         "states": [
21             {
22                 "name": "Oxygen Level",
23                 "type": "number",
24                 "unit": "%",
25                 "controllable": False
26             }
27         ]
28     })
29
30 # Налаштування датчика кисню
31 pinMode(oxygenSensorPin, INPUT)
32 attachInterrupt(oxygenSensorPin, onOxygenChange, CHANGE)
33
34 # Відновлення попередньо збереженого рівня кисню (якщо існує)
35 oxygenLevel = restoreProperty("Oxygen Level", 0.0)
36
37 def onOxygenChange():
38     global oxygenLevel
39     oxygenValue = analogRead(oxygenSensorPin)
40     # Конвертуємо значення датчика від 0 до 1023 у відсотки кисню
41     oxygenLevel = (oxygenValue / 1023) * 100
42
43 def main():
44     setup()
45     while True:
46         if oxygenLevel < MIN_OXYGEN_LEVEL:
47             turnOnVentilation()
48         else:
49             turnOffVentilation()
50
51         sendReport()
52         delay(1000)
53
54 def turnOnVentilation():

```

Рисунок 4.7 – Програмування датчика кисню

4.4 Тестування системи Інтернету речей ТОВ «Дніпровський трубний завод»

Виконуємо тестування системи з сонячною панеллю. Таке рішення корисним в умова блекаутів, щоб завод міг функціонувати. Для цього ми спочатку ми вимикаємо всі розетки, що показати як статус заряду з 0% буде підвищуватись до 100%. Результат на рис.4.8-4.10.

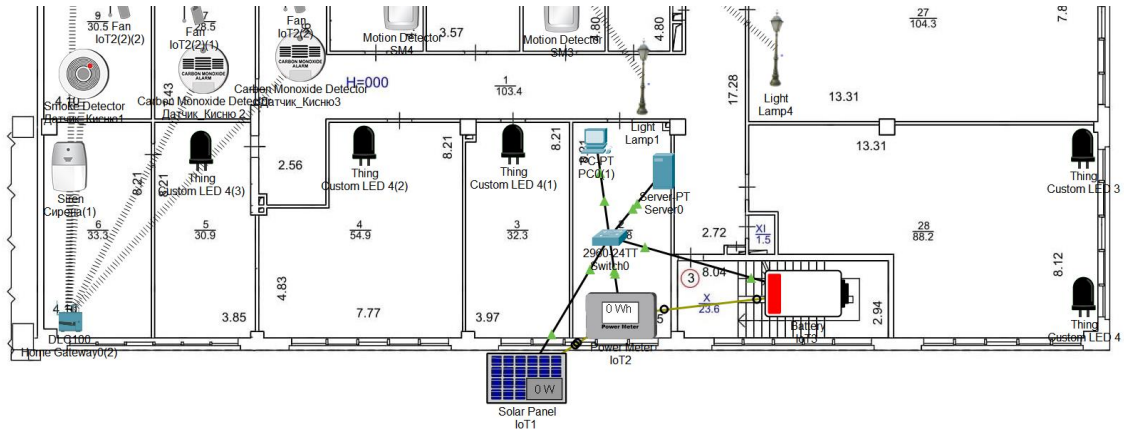


Рисунок 4.8 – Ємність батареї 20%

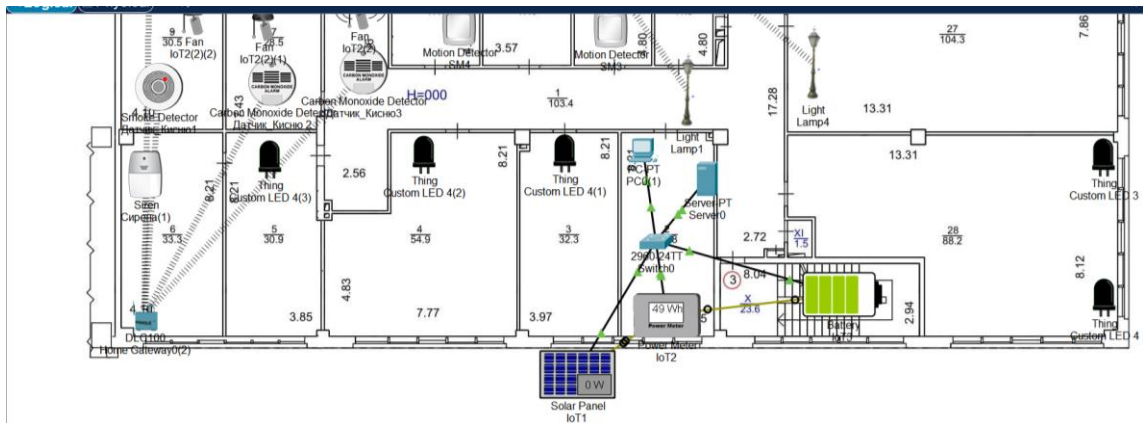


Рисунок 4.9 – Ємність батареї 60%

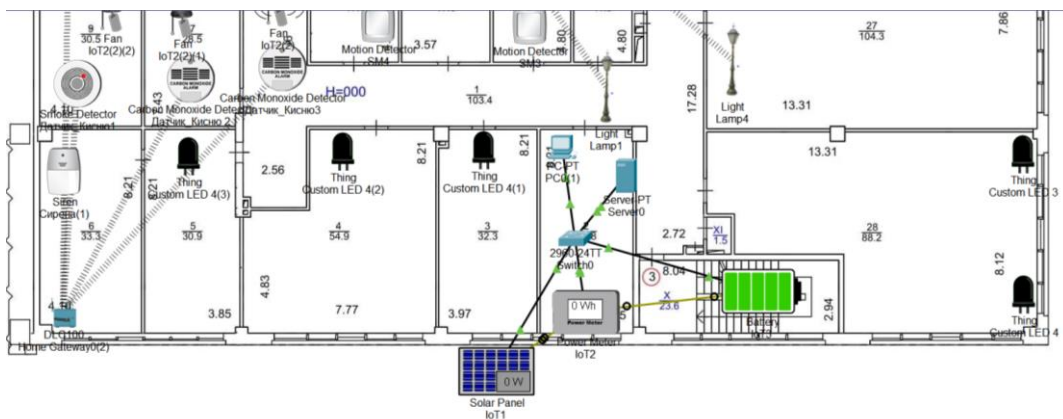


Рисунок 4.10 – Ємність батареї 100%

Під'єднуємось до мережі і розетки змінюють колір чорного на зелений, що

ВИСНОВКИ

У кваліфікаційній роботі "Комп'ютерна система моніторингу рівня кисню в приміщеннях ТОВ Дніпровський трубний завод з детальним опрацюванням побудови, налаштування та безпеки корпоративної мережі" була розроблена комп'ютерна система моніторингу рівня кисню в приміщеннях ТОВ Дніпровський трубний завод. Система складається з датчиків кисню, мережевого обладнання, IoT-пристроїв та серверного програмного забезпечення.

Була проведена детальна опрацювання побудови системи. Датчики кисню були розміщені в стратегічних точках приміщень заводу, що дозволяє вимірювати рівень кисню у реальному часі.

Для забезпечення функціональності системи була побудована корпоративна мережа, яка включає в себе комутатори, маршрутизатори, сервери та інші мережеві пристрої. Було проведено детальне налаштування мережевого обладнання для забезпечення безперебійної роботи системи.

Розроблено програмне забезпечення для IoT-сервера, яке забезпечує моніторинг та аналіз даних, отриманих від датчиків кисню. Це дозволяє операторам системи контролювати рівень кисню та при необхідності приймати відповідні заходи.

Важливою складовою роботи була забезпечення безпеки корпоративної мережі. Було встановлено механізми автентифікації, авторизації та обліку (AAA), а також застосовано технології шифрування для забезпечення конфіденційності та цілісності даних.

Результати роботи свідчать про успішне впровадження комп'ютерної системи моніторингу рівня кисню в приміщеннях ТОВ Дніпровський трубний завод. Система дозволяє оперативно контролювати рівень кисню та вчасно реагувати на відхилення.

Впровадження системи моніторингу рівня кисню сприяє покращенню безпеки працівників ТОВ «Дніпровський трубний завод» та забезпечує належні

умови праці.

Результати дослідження підтверджують ефективність та важливість застосування комп'ютерних систем моніторингу для забезпечення безпеки та комфорту в промислових приміщеннях.

ПЕРЕЛІК ПОСИЛАНЬ

1. Дніпровський трубний завод. [Електронний ресурс] – Режим доступу до ресурсу: http://www.dpw.com.ua/index.php?option=com_content&view=frontpage&Itemid=61&lang=uk
2. New York, NY, Jan. 22, 2021. Facts and Factors. “Global Cloud Computing Market Size & Share Will Reach USD 1025.9 Billion by 2026” [Електронний ресурс] – Режим доступу до ресурсу: <https://www.globenewswire.com/newsrelease/2021/01/22/2162789/0/en/Global-Cloud-Computing-Market-Size-Share-Will-Reach-USD-1025-9-Billion-by-2026-Facts-Factors.html>
3. Chris Richardson, “Pattern: Microservice Architecture”, 2020, [Електронний ресурс] – Режим доступу до ресурсу: <https://microservices.io/patterns/microservices.html#6>.
4. What is EIGRP? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.metaswitch.com/knowledge-center/reference/what-is-open-shortest-path-first-ospf>
7. What is AAA? | FreeRADIUS Documentation – NetworkRADIUS)? [Електронний ресурс] – Режим доступу до ресурсу: <https://networkradius.com/doc/current/concepts/introduction/AAA.html>
8. Wat is NAT en hoe werkt het? – VOIPZeker [Електронний ресурс] – Режим доступу до ресурсу: <https://voipzeker.nl/alles-over-bellen-met-voip/nat>
9. LACP vs PAGP: What's the Difference? – GeeksforGeeks [Електронний ресурс] – Режим доступу до ресурсу: <https://www.geeksforgeeks.org/lacp-vs-pagp-whats-the-difference/#:~:text=LACP%20is%20an%20Open%20standard%2C%20i.e.%2C%20supported%20by%20most%20vendors,be%20used%20between%20Cisco%20devices.&text=LACP%20has%20two%20modes%2C%20i.e.,ports%20by%20exchanging%20LACP%20packets.>)
10. Hassan, Mosfiqun & Semantha, Farida Habib & Islam, Mohammed &

Siddique, Abdul & Faisal, Fahad & Hasan, Mehedi. (2020). An IoT Based Environment Monitoring System. 10.1109/ICISS49785.2020.9316050.

11. Barrenetxea, Guillermo & Ingelrest, Francois & Schaefer, Gunnar & Vetterli, Martin. (2008). Wireless Sensor Networks for Environmental Monitoring: The SensorScope Experience. 1. 98 - 101. 10.1109/IZS.2008.4497285.

Додаток А
Текст програми

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

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ НАЛАШТУВАННЯ МЕРЕЖІ
КОМП'ЮТЕРНОЇ СИСТЕМИ
ТОВ ДНІПРОВСЬКИЙ ТРУБНИЙ ЗАВОД**

Текст програми

804.02070743.23005-01 12 01

Листів 5

2023

АНОТАЦІЯ

Розроблена програма включає в себе фрагмент програмного коду, який використовується для налаштування компонентів корпоративної мережі комп'ютерної системи третього апеляційного адміністративного суду. Її головна мета - забезпечити налаштування різноманітних аспектів мережі, таких як динамічна маршрутизація, DHCP, AAA, інтерфейси, протокол маршрутизації NAT, консольні і vty лінії, а також створення віртуальних приватних мереж (VPN), налаштування домену та протоколу SSH в комп'ютерній системі

ЗМІСТ

	стор.
1. Налаштування датчика кисню	4
2. Налаштування сонячної панелі	6

```

1      Налаштування датчика кисню
!
from gpio import *
from time import *
from ioeclient import *
from environment import *
from physical import *

INPUT_PIN = 0
OUTPUT_PIN = 1
MAX_OXYGEN_LEVEL = 100.0 #
Максимальний рівень кисню в системі (у
відсотках)
MIN_OXYGEN_LEVEL = 19.5 #
Мінімальний припустимий рівень кисню в
системі (у відсотках)
oxygenLevel = 0.0 # Поточний рівень кисню
в системі (у відсотках)
oxygenSensorPin = 0 # Пін для підключення
датчика кисню

def setup():
    IoEClient.setup({
        "type": "Oxygen Monitor",
        "states": [
            {
                "name": "Oxygen Level",
                "type": "number",
                "unit": "%",
                "controllable": False
            }
        ]
    })

    # Налаштування датчика кисню
    pinMode(oxygenSensorPin, INPUT)
    attachInterrupt(oxygenSensorPin,
onOxygenChange, CHANGE)

    # Відновлення попередньо збереженого
рівня кисню (якщо існує)
    oxygenLevel = restoreProperty("Oxygen
Level", 0.0)

def onOxygenChange():
    global oxygenLevel
    oxygenValue =
analogRead(oxygenSensorPin)
    # Конвертуємо значення датчика від 0 до
1023 у відсотки кисню
    oxygenLevel = (oxygenValue / 1023) * 100

def main():
    setup()
    while True:

```

```

if oxygenLevel < MIN_OXYGEN_LEVEL:
    turnOnVentilation()
else:
    turnOffVentilation()

    sendReport()
    delay(1000)

def turnOnVentilation():
    # Код для ввімкнення вентиляційної системи
    pass

def turnOffVentilation():
    # Код для вимкнення вентиляційної системи
    pass

def sendReport():
    IoEClient.reportStates({
        "Oxygen Level": oxygenLevel
    })

if __name__ == "__main__":
    main()

```

2 Налаштування сонячної панелі

```

from gpio import *
from time import *
from ioeclient import *
from environment import *
from physical import *
import math

INPUT_PIN = 0
OUTPUT_PIN = 1
MAX_STORED_POWER = 1500.0 #2000 watts
MULTIPLIER = float(255)/1023
LOG_BASE =
1.0749111034571373359815489867558
availablePowerPercent = 0.0
availablePower =
float(MAX_STORED_POWER)/2.0
receivedPower = 0
AVAIL_POWER_PROP="Available power
(%)"
CHARGE_RATE = 19
LOOP_DELAY = 500

def setup():
    setImage('100')
    IoEClient.setup({
        "type": "Battery",
        "states": [
            {
                "name": "Available power",
                "type": "number",
                "unit": '%',
                "controllable": False
            }
        ]
    })

IoEClient.onInputReceive(onInputReceiveDone)

    global availablePowerPercent
    global availablePower

    init_percent =
availablePower/MAX_STORED_POWER*100
    availablePowerPercent =
restoreProperty(AVAIL_POWER_PROP,
init_percent)
    if availablePowerPercent < 0 or
availablePowerPercent > 100:
        availablePowerPercent = init_percent
        availablePower =
availablePowerPercent*MAX_STORED_POWE
R/100

    sendReport()

```

```

def onInputReceiveDone(input):
    processData(input, True)

def restoreProperty(propertyName, defaultValue):
    value = getDeviceProperty(getName(),
propertyName)
    if not (value is "" or value is None):
        if type(defaultValue) is float :
            value = float(value)

        setDeviceProperty(getName(), propertyName,
value)
    return value
    return defaultValue

def main():
    global availablePowerPercent
    global availablePower
    global MAX_STORED_POWER
    setup()
    while True:
        availablePowerPercent =
(float(availablePower)/MAX_STORED_POWER) *
100
        if availablePowerPercent < 10:
            setImage('Battery_0')
        elif availablePowerPercent < 30:
            setImage('Battery_20')
        elif availablePowerPercent < 50:
            setImage('Battery_40')
        elif availablePowerPercent < 70:
            setImage('Battery_60')
        elif availablePowerPercent < 90:
            setImage('Battery_80')
        elif availablePowerPercent <= 100:
            setImage('Battery_100')

        sendReport()
        delay(LOOP_DELAY)
        receivePower(LOOP_DELAY*3)
        sendPower()
        delay(LOOP_DELAY*2)

def receivePower(timeDeltaMsec):
    global availablePower
    global receivedPower
    receivedPowerRaw = analogRead(0)
    if receivedPowerRaw > 0:
        receivedPower = math.pow(LOG_BASE,
receivedPowerRaw * MULTIPLIER)
        receivedPower = min(receivedPower,
CHARGE_RATE * timeDeltaMsec * 0.001)
    else:

```



```

    receivedPower = 0
    availablePower += receivedPower
    if availablePower >
MAX_STORED_POWER :
    availablePower =
MAX_STORED_POWER

```

```

def sendPower():
    global availablePower
    totalPowerUsed = 0
    currentWattage = 0
    numberOfSlots = int(getSlotsCount())
    deviceAtPort = False
    for i in range (1, numberOfSlots):
        #Start at 1 because 0 is the input port
        if getAttributeOfDeviceAtSlot('wattage', i):
            currentWattage =
float(getAttributeOfDeviceAtSlot('wattage', i))
            deviceAtPort = True
        else:
            currentWattage = 0

```

```

    totalPowerUsed += currentWattage
    if deviceAtPort == True:
        if availablePower > currentWattage:
            outputElectricity(i, currentWattage)
            availablePower -= currentWattage
        else:
            outputElectricity(i, 1)

```

```

    deviceAtPort = False
    if availablePower < 0:
        availablePower = 0

```

```

def sendReport():
    global availablePowerPercent
    IoEClient.reportStates(availablePowerPercent)
    setDeviceProperty(getName(),
AVAIL_POWER_PROP,
availablePowerPercent)

```

```

def setImage(component):
    setComponentOpacity(component, 1)

    if not 'Battery_0' == component :
        setComponentOpacity('Battery_0', 0)
    if not 'Battery_20' == component :
        setComponentOpacity('Battery_20', 0)
    if not 'Battery_40' == component :

```

```

        setComponentOpacity('Battery_40', 0)
    if not 'Battery_60' == component :
        setComponentOpacity('Battery_60', 0)
    if not 'Battery_80' == component :
        setComponentOpacity('Battery_80', 0)
    if not 'Battery_100' == component :
        setComponentOpacity('Battery_100', 0)

```

```

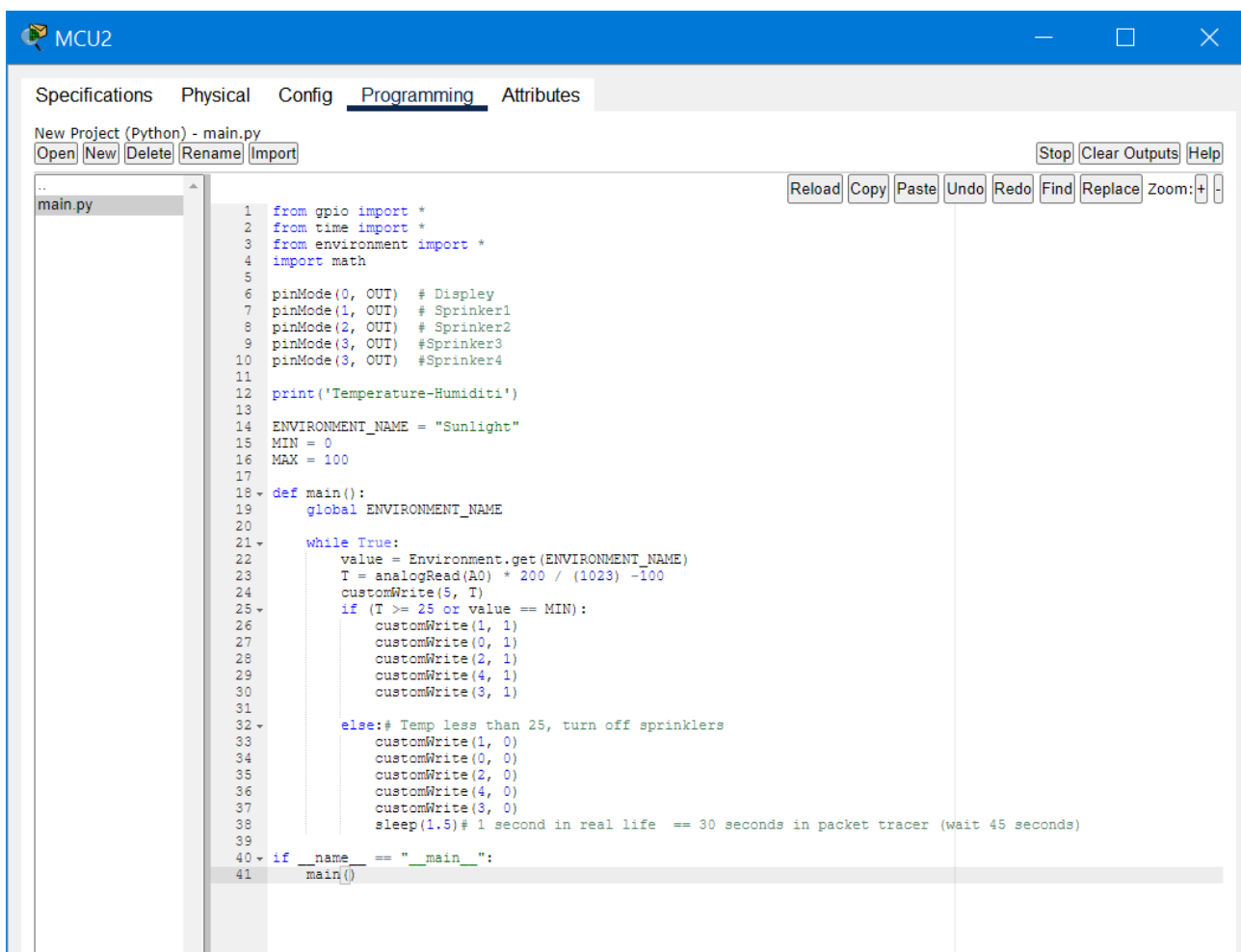
def outputElectricity(port, value):
    global LOG_BASE
    el_log =
math.floor(float(math.log(value))/math.log(LOG_B
ASE))
    if el_log < 0:
        el_log = 0
    elif el_log > 255:
        el_log = 255
    analogWrite(port, el_log)

if __name__ == "__main__":
    main()

```

Додаток Б

Налаштування контролера



```
MCU2
Specifications Physical Config Programming Attributes
New Project (Python) - main.py
Open New Delete Rename Import Stop Clear Outputs Help
Reload Copy Paste Undo Redo Find Replace Zoom: + -
..
main.py
1 from gpio import *
2 from time import *
3 from environment import *
4 import math
5
6 pinMode(0, OUT) # Display
7 pinMode(1, OUT) # Sprinkler1
8 pinMode(2, OUT) # Sprinkler2
9 pinMode(3, OUT) #Sprinkler3
10 pinMode(3, OUT) #Sprinkler4
11
12 print('Temperature-Humiditi')
13
14 ENVIRONMENT_NAME = "Sunlight"
15 MIN = 0
16 MAX = 100
17
18 def main():
19     global ENVIRONMENT_NAME
20
21     while True:
22         value = Environment.get(ENVIRONMENT_NAME)
23         T = analogRead(A0) * 200 / (1023) -100
24         customWrite(5, T)
25         if (T >= 25 or value == MIN):
26             customWrite(1, 1)
27             customWrite(0, 1)
28             customWrite(2, 1)
29             customWrite(4, 1)
30             customWrite(3, 1)
31
32         else:# Temp less than 25, turn off sprinklers
33             customWrite(1, 0)
34             customWrite(0, 0)
35             customWrite(2, 0)
36             customWrite(4, 0)
37             customWrite(3, 0)
38             sleep(1.5)# 1 second in real life == 30 seconds in packet tracer (wait 45 seconds)
39
40 if __name__ == "__main__":
41     main()
```

Налаштування батареї

The screenshot shows the IoT1 programming environment with the 'Programming' tab selected. The file 'main.py' is open, displaying a Python script for a solar panel simulation. The script includes imports for Options, time, math, physical, gpio, Environment, and IoEClient. It defines several variables for simulation parameters and a setup function for the IoEClient.

```

1 from options import Options
2 from time import *
3 import math
4 from physical import *
5 from gpio import *
6 from environment import Environment
7 from ioeclient import IoEClient
8 #from pyjs import *
9
10
11 #Solar Panel
12 #Read the sunlight levels
13 #Output electricity based on sunlight
14 #Panel will be 160Watts per square meter
15
16 #Features output to IoE Server:
17 # number of kWh of energy produced since turning on
18 # number of kWh per minute
19 # current production
20 ENVIRONMENT_NAME = "Sunlight" # var ENVIRONMENT_NAME
21 MULTIPLIER = 255. / 1023 # var MULTIPLIER
22 MAX_POWER = 1000. #1000 Watts of power based on one meter solar panel at noon at the equator # var MAX
23 EFFICIENCY = 0.16 #About a 16 percent efficiency per solar panel # var EFFICIENCY
24 PANEL_POWER = MAX_POWER * EFFICIENCY # var PANEL_POWER
25 LOG_BASE = 1.0749111034571373359815489867558 # var LOG_BASE
26
27 state = 1 # var state
28 electricity = 0 # var electricity
29 #tick = 0 # var tick
30
31
32 def setup ():
33
34     IoEClient.setup({
35         "type": "Solar",
36         "states": [{
37             "name": "Status",
38             "type": "number",
39             "unit": 'Wh',
40             "controllable": False
41         }]
42     })
43
44     IoEClient.onInputReceive ( lambda rinput: processData(rinput, True) )
45
46     sendReport()
47
48
49
50

```