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

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

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня бакалавра

студента Торгольського Андрій Олександровича
(ПІБ)
академічної групи 123-21ск-1
(шифр)
спеціальності 123 Комп'ютерна інженерія
(код і назва спеціальності)

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

на тему "Кіберфізична система розумного складу з детальним опрацюванням побудови та налаштування корпоративної мережі"
(назва за наказом ректора)

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

Рецензент				

Нормоконтролер	проф. Цвіркун Л.І.			

Дніпро
2024

ЗАТВЕРДЖЕНО:
завідувач кафедри
інформаційних технологій
та комп'ютерної інженерії
(повна назва)
_____ Гнатушенко В.В.
(підпис) (прізвище, ініціали)

"25" січня 2024 року

ЗАВДАННЯ
на кваліфікаційну роботу
ступеня бакалавр

студента Торгольського А.О. **академічної групи** 123-21ск-1
(прізвище та ініціали) (шифр)

спеціальності 123 «Комп'ютерна інженерія»

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

на тему " Кіберфізична система розумного складу з детальним опрацюванням побудови та налаштування корпоративної мережі "

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

Розділ	Зміст	Термін виконання
Стан питання та постановка завдання	На основі матеріалів виробничих практик, інших науково-технічних джерел конкретизується предмет та мету роботи та виконується постановка завдання	10.05.2024
Розробка апаратної частини	На основі аналізу підприємства формулюються технічні вимоги до кіберфізичної системи та розробляється апаратна частина системи	17.05.2024
Розробка корпоративної мережі	Виконується розрахунок налаштувань корпоративної мережі та перевірка роботи системи, розробляються методи та налаштування обладнання для захисту інформації в системі	24.05.2024
Розробка компонента системи	Виконується детальна розробка компонента системи	31.05.2024

Завдання видано _____ проф. Олевський В.І.
(підпис керівника) (прізвище, ініціали)

Дата видачі 25.01.2024

Дата подання до екзаменаційної комісії 14.06.2024

Прийнято до виконання _____ Торгольський А.О.

РЕФЕРАТ

Пояснювальна записка: 117 с., 75 рис., 12 табл., 2 дод., 10 джерел.

КОМП'ЮТЕРНА МЕРЕЖА, СИСТЕМА IOT, API, KOTLIN, LAN, NAT, ROUTER, SERVER, SWITCH, VLAN, VPN.

Об’єкт розробки – кіберфізична система розумного складу з детальним опрацюванням побудови та налаштування корпоративної мережі.

Мета: створення кіберфізичної системи розумного складу з детальним опрацюванням побудови та налаштування корпоративної мережі.

В розділі «Стан питання і постановка задачі» наведена стисла характеристика галузі та умов застосування КС, відомості про технології збору та передачі інформації, розроблена організаційна структура, сформовані завдання та мета роботи та визначені основні напрямки рішення поставлених завдань.

В розділі «Розробка апаратної частини комп’ютерної системи» сформовані технічні вимоги та специфікація апаратних засобів для впроваджувальної КС; розроблена топологічна та структурна схема комплексу технічних засобів КС.

В розділі «Розробка корпоративної мережі» розрахована схема адресації підмереж та пристроїв, розроблена топологічна схема корпоративної мережі. Розроблена комп’ютерна мережа реалізована у вигляді моделі на симуляторі Cisco Packet Tracer з перевіркою її працездатності.

В розділі «Розробка компонента системи» обґрунтований вибору апаратних засобів та програмного забезпечення кіберфізичної системи; розроблені схеми підключення виконавчих пристроїв; наведений опис вікон мобільного застосунку. Мобільний застосунок забезпечує виконання наступних функцій: керування роботою окремих механізмів маніпулятору; підвищення ефективності виконання складських операцій.

Результати перевірки у вигляді таблиць, графіків описані і наводяться у пояснювальній записці та додатках.

ЗМІСТ

Перелік скорочень, умовних познак, одиниць і термінів 7
Вступ..... 8
1 Стан питання та постановка завдання..... 10
1.1 Стисла характеристика галузі та умов застосування КС..... 10
1.2 Характеристика і структура об’єкта впровадження..... 11
1.3 Стислі відомості про технології збору та передачі інформації..... 15
1.4 Принципи, технічні засоби та математичні методи інформаційного забезпечення..... 17
1.5 Огляд існуючих інженерних рішень КС в галузі..... 19
1.6 Завдання і мета роботи..... 22
1.7 Визначення можливих напрямків рішення поставлених завдань.. 23
2 Розробка апаратної частини комп’ютерної системи підприєства..... 26
2.1 Технічні вимоги до Системи..... 26
2.1.1 Вимоги до Системи в цілому..... 26
2.1.1.1 Вимоги до структури і функціонування Системи..... 26
2.1.1.1.1 Перелік підсистем, їх призначення та основні характеристики..... 26
2.1.1.1.2 Вимоги до способів і засобів зв’язку для інформаційного обміну між компонентами Системи..... 28
2.1.1.1.3 Вимоги до режимів функціонування Системи 29
2.1.1.1.4 Вимоги до діагностування Системи..... 29
2.1.1.1.5 Перспективи розвитку Системи..... 30
2.1.1.2 Вимоги до показників призначення Системи..... 30
2.1.1.3 Вимоги до експлуатації, технічного обслуговування, ремонту і збереження..... 31
2.1.1.3.1 Умови і регламент (режим) експлуатації, що повинні забезпечувати використання технічних засобів (ТЗ) Системи з заданими технічними показниками..... 31
2.1.1.3.1 Вимоги до параметрів мереж електроживлення..... 32

2.1.1.3.2 Вимоги до кількості, кваліфікація обслуговуючого персоналу і режимам його роботи 33
2.1.1.3.3 Вимоги до складу, розміщенню й умовам збереження комплексу запасних виробів та приладів 35
2.1.1.3.4 Вимоги до регламенту обслуговування 36
2.1.1.4 Вимоги до патентної чистоти 37
2.1.1.5 Додаткові умови 37
2.1.1.5.1 Вимоги до активного обладнання 37
2.1.1.5.2 Вимоги до кабель-каналів, інформаційним та електричним розеткам 38
2.1.1.5.3 Вимоги до комунікаційного обладнання і його розташування 39
2.1.1.5.4 Вимоги до однорідності 39
2.1.1.5.5 Вимоги до надійності 40
2.1.1.5.6 Вимоги до безпеки 40
2.1.1.5.7 Вимоги до захисту інформації від несанкціонованого доступу 41
2.1.2 Вимоги до функцій (задач), виконуваним Системою 42
2.1.3 Вимоги до видів забезпечення 45
2.1.3.1 Вимоги до математичного забезпечення 45
2.1.3.2 Вимоги до інформаційного забезпечення 46
2.1.3.3 Вимоги до лінгвістичного забезпечення 46
2.1.3.4 Вимоги до технічного забезпечення 47
2.1.3.5 Вимоги до організаційного забезпечення 48
2.1.3.6 Вимоги до методичного забезпечення 48
2.2 Розробка апаратної частини кіберфізичної системи 49
2.2.1 Розробка загальної структури кіберфізичної системи 49
2.2.2 Вибір і обґрунтування комплексу технічних кіберфізичної системи 51
2.2.3 Розрахунок інтенсивності вихідного трафіку найбільшої локальної мережі підприємства 56
3 Розробка корпоративної мережі 61
3.1 Розрахунок схеми адресації корпоративної мережі 61
3.2 Розрахунок схеми адресації пристроїв у корпоративній мережі ... 64
3.3 Розробка топологічної схеми корпоративної мережі 65

3.4 Налаштування моделі комп'ютерної мережі 67
3.4.1 Базове налаштування конфігурації пристроїв 67
3.4.2 Налаштування маршрутизаторів 67
3.4.3 Налаштування роботи Інтернет 68
3.5 Захист інформації в комп'ютерній Системі 69
3.5.1 Налаштування маршрутизаторів на підтримку служби AAA 69
3.5.2 Налаштування віртуальних локальних мереж VLAN 69
3.5.3 Налаштування віртуальної приватної мережі VPN 72
3.6 Перевірка комп'ютерної Системи підприємства 73
4 Розробка компонента системи 81
4.1 Загальні відомості 81
4.2 Обґрунтування технічних засобів 83
4.2.1 Обґрунтування апаратних засобів 83
4.2.2 Обґрунтування ПЗ 86
4.3 Розробка математичної моделі роботи маніпулятору 90
4.3.1 Рівняння прямої задачі кінематики 90
4.3.2 Рівняння зворотної задачі кінематики 92
4.4 Налаштування апаратних засобів та ПЗ 94
4.5 Опис розробленої програми 97
4.5.1 Загальні відомості 97
4.5.2 Головне вікно програми 98
4.5.3 Вікно Manual Control 98
Висновки 100
Перелік джерел посилання 101
Додаток А. Текст програми мобільного застосунку для керування роботою маніпулятору 102
Додаток Б. Текст програми контролеру та Wi-Fi модулю для керування маніпулятором 111

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАК, ОДИНИЦЬ І ТЕРМІНІВ

КС – кіберфізична система.

Комутатор – пристрій для з'єднання декількох вузлів комп'ютерної мережі.

Маршрутизатор – пристрій для поєднання мереж.

ПК – персональний комп’ютер.

ПЗ – програмне забезпечення

Скетч (Sketch) – це вихідні файли, що містять код на мові програмування C/C++.

ОС – операційна система.

LAN (Local Area Network) – це локальна комп'ютерна мережа.

VLAN (Virtual Local Area Network) – це віртуальна локальна комп'ютерна мережа.

WEP (Wired Equivalent Privacy) – стандарт захисту бездротового трафіку.

API (Application Program Interface) - набір інструкцій, для спілкування програмам між собою.

VPN (Virtual Private Network) – технологія захисту інтернет з’єднання.

NAT (Network Address Translation) – механізм перетворення IP-адреси в заголовку пакету.

ВСТУП

В умовах сучасного ведення бізнесу, ефективність управління складським простором стає одним із ключових факторів забезпечення успішної конкурентної діяльності компаній. Тому саме використання кіберфізичних комп'ютерних систем, що поєднують як фізичні, так і віртуальні складові оптимізації процесів управління та підвищення продуктивності виробничих процесів, відіграє важливу роль.

Роботизування систем призводить до універсалізації та автоматизації виробничих процесів, зменшуючи в них роль людини і, як наслідок, можливих збоїв через «людський фактор». Для кінцевого споживача такий процес інтеграції техніки в процеси виробництва виливається у зниження вартості продукту та підвищення його якості.

Оцінка сучасного стану дослідження свідчить про активний інтерес до впровадження аналогічних систем у виробничі процеси провідних компаній.

Компанія Amazon застосовує у своїй роботі власну систему розумного складу під назвою Amazon Robotics, що поєднує використання технологій штучного інтелекту та автономних роботів.

Компанія Honeywell Intelligrated застосовує у своїй роботі власну систему Warehouse Execution Software (WES), що дозволяє інтегрувати різноманітні технології, які включають системи автоматизованих сортувальних ліній, використання роботів, дронів та IoT-пристроїв.

Впровадження кіберфізичної комп'ютерної системи надає значні переваги:

− оптимізація процесів, що дозволить підвищити продуктивність, знизити час обробки товарів та зменшити витрати;

− покращення моніторингу та аналізу, що дозволить оперативно реагувати на будь-які зміни у виробничих процесах та навколишньому середовищі;

− підвищення точності та ефективності, що дозволяє зменшити фактор людського втручання, які знижую ймовірність помилок та підвищують якість обслуговування.

Отже, мета цієї роботи – детальне опрацювання створення кіберфізичної комп’ютерної системи розумного складу для оптимізації управління, ефективного використання складського приміщення та підвищення продуктивності виробничих процесів.

1 СТАН ПИТАННЯ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1 Стисла характеристика галузі та умов застосування КС

Кіберфізична система розумного складу – це сучасний підхід управління та оптимізації складських процесів.

Сфера складського господарства спеціалізується на розробці та впроваджені апаратний рішень та програмного забезпечення з метою оптимізації управління, ефективного використання складського приміщення та підвищення продуктивності складських процесів.

Умовами застосування даної КС є: наявність фізичних складських приміщень; потреба в ефективності та продуктивності управління виробничими процесами; потреба в автоматизації та інтеграції виробничих процесів; забезпечення відповідного рівня конкурентоспроможності.

Використання систем розумного складу можливе в складських приміщення різних розмірів, від невеликих магазинів до великих дистрибуцій них центрів.

Застосування систем розумного складу проявляється у різних галузях, таких як роздрібна торгівля, логістика, виробництво та інші, завдяки їхній гнучкості та можливостю інтеграції з власною інфраструктурою кожного підприємства.

Для вказаної галузі необхідно розробити та налаштувати корпоративну мережу логістичного підприємства та створити автоматизовану систему дистанційного управління роботою маніпуляторів складського приміщення.

Корпоративна мережа повинна зменшувати ризик випадкового або навмисного втручання із заподіянням шкоди майну складського приміщення.

Автоматизована система дистанційного управління роботою маніпуляторів логістичного підприємства повинна підвищувати

ефективність виконання складських операцій та зменшення ступені залученості робочого персоналу.

1.2 Характеристика і структура об’єкта впровадження

Об’єкт впровадження – логістичне підприємство.

Основна діяльність підприємства – це прийом, зберігання та відправлення товарів, комплектування замовлень, інвентаризація, організація перевезень і послуги із обслуговування клієнтів.

Підприємство має сім структурних відділів:

− фінансово-бухгалтерський відділ;

− відділ логістики;

− ІТ-відділ;

− відділ служби доставки;

− відділ роботи з клієнтами;

− відділ переробки та зберігання вантажу;

− відділ Call-центру.

Організаційна структура логістичного підприємства наведено на рисунку 1.1.

Директору філіалу керує всіма відділами.

Фінансово-бухгалтерський відділ спеціалізується на роботі з різними групами та категоріями кінцевих клієнтів. Відділ розробляє та впроваджує фінансові інструменти та системи, що дозволяють забезпечити прибутковість.

Відділ логістики спеціалізується на контролі міжміської логістики, міської логістики та термінальної логістики.

Відділ служби доставки спеціалізується на розробці оптимальних маршрутів доставки, впровадження технологічних інновацій для підвищення якості та ефективності доставки.

Відділ переробки та зберігання вантажу спеціалізується на питання отримання та відправлення вантажу.

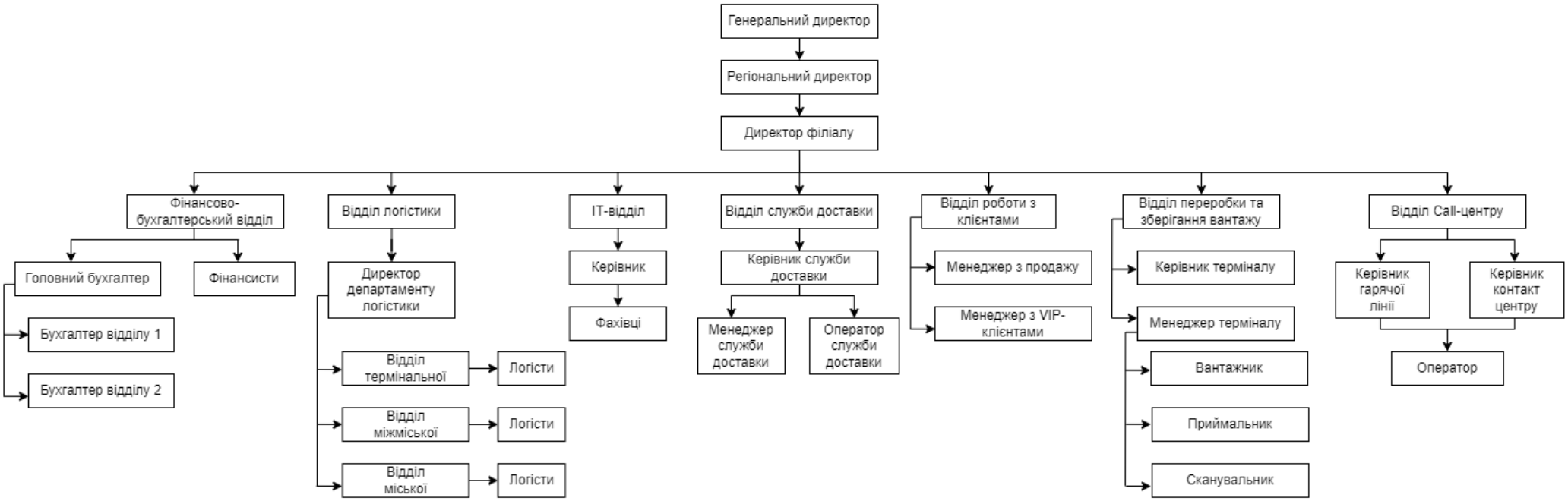

Рисунок 1.1 – Організаційна структура логістичного підприємства

IT-відділ займається безперервною підтримкою функціонування IT-інфраструктури.

Функції IT-відділу:

− моніторинг та вивчення нових інформаційних технологій;

− забезпечення безперебійної підтримки роботи обладнання та підтримка користувачів;

− проєктування, розробка, впровадження та підтримка корпоративної мережі.

Відділ по роботі з клієнтами спеціалізується на досягнені цілей з продажу, обслуговувані існуючих клієнтів та залученні нових дотримуючись стандартів якості обслуговування.

Функції відділу переробки та зберігання вантажу:

− прийом та видача вантажу клієнту;

− забезпечення ефективного способу;

− обробка запитів на повернення вантажу, та зміни даних щодо отримувача вантажу;

− надання клієнту інформації про діяльність підприємства.

Відділ Call-центру спеціалізується на забезпеченні оперативного та якісного зв’язку з клієнтами у питаннях з надання консультацій, обробки замовлень та вирішенні можливих проблем.

Для ефективного виконання складських операцій, таких як розвантаження, завантаження та переміщення товарів приміщення складу обладнане десятьма маніпуляторами, які працюють за принципом ручного керування (див. рис. 1.2).

Маніпулятори складського приміщення мають 3 ступені свободи (4DoF). Наявність 3-ох ступенів свободи дозволяє виконувати складні рухи та позиціювання в просторі.

Комплекс механізмів маніпулятору включає в себе наступні компоненти та їх відповідні ступені свободи:

– база, основна частина маніпулятору, яка здійснює обертання навколо вертикальної осі Z, забезпечуючи можливість повороту вліво та вправо;

– горизонтальне плече, секція маніпулятору, яка забезпечує переміщення вздовж горизонтальної осі X, забезпечуючи можливість руху вперед та назад;

– вертикальне плече, секція маніпулятору, яка забезпечує переміщення вздовж вертикальної осі Y, забезпечуючи можливість руху вгору та вниз;

– захоплювач, кінцева секція маніпулятору, яка забезпечує захоплення та утримання предмету для подальшого переміщення.

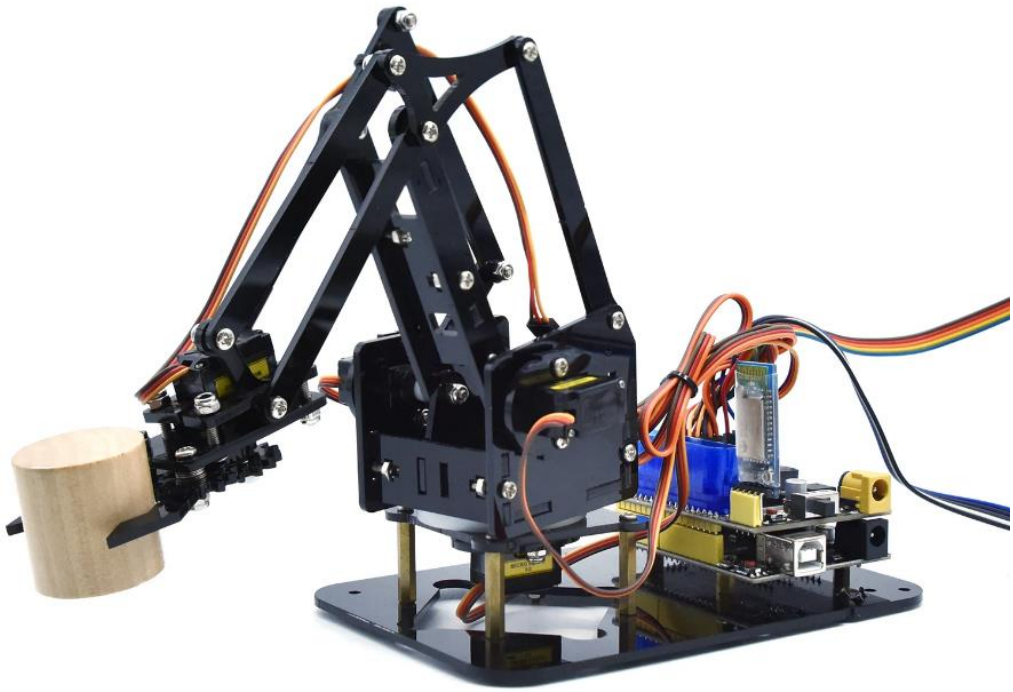

Рисунок 1.2 – Маніпулятор

Основою платою керування є плата Arduino Uno на базі мікроконтролеру ATmega328P та мікросхемою CP2102 (перетворювач USB-UART).

В якості виконавчих пристроїв маніпулятору використовуються сервоприводи Micro Servo 9G. Кількість сервоприводів на одному маніпуляторі – 4.

Технічні характеристики сервоприводу Micro Servo 9G:

− кут повороту: 180º;

− матеріал редуктору: нейлон (пластик);

− тип керування: PWM (широтно-імпульсна модуляція);

− живлення: 4.6 – 6 В.

1.3 Стислі відомості про технології збору та передачі інформації

Складське приміщення розташоване за адресою - Дніпропетровська область, Дніпропетровський район, смт. Слобожанське, вул. Жасмінова, 29 (див рис. 1.3).

Офіс логістичного підприємства, в якому розташовуються працівники, знаходиться на відстані 200 метрів від складського приміщення, за адресою - Дніпропетровська область, Дніпропетровський район, смт. Слобожанське, вул. Жасмінова, 29.

Рисунок 1.3 – Місцезнаходження складського приміщення та будівлі офісу на мапі Google Maps

Загальна площа складського приміщення - 3 га, на якій знаходяться 12 вантажних відділів (див. рис. 1.4).

Рисунок 1.4 – План складського приміщення

Офісна будівля логістичного підприємства має вигляд 2 поверхового офісу. Структурна схема розміщення підрозділів підприємства на першому поверсі (див. рис. 1.5).

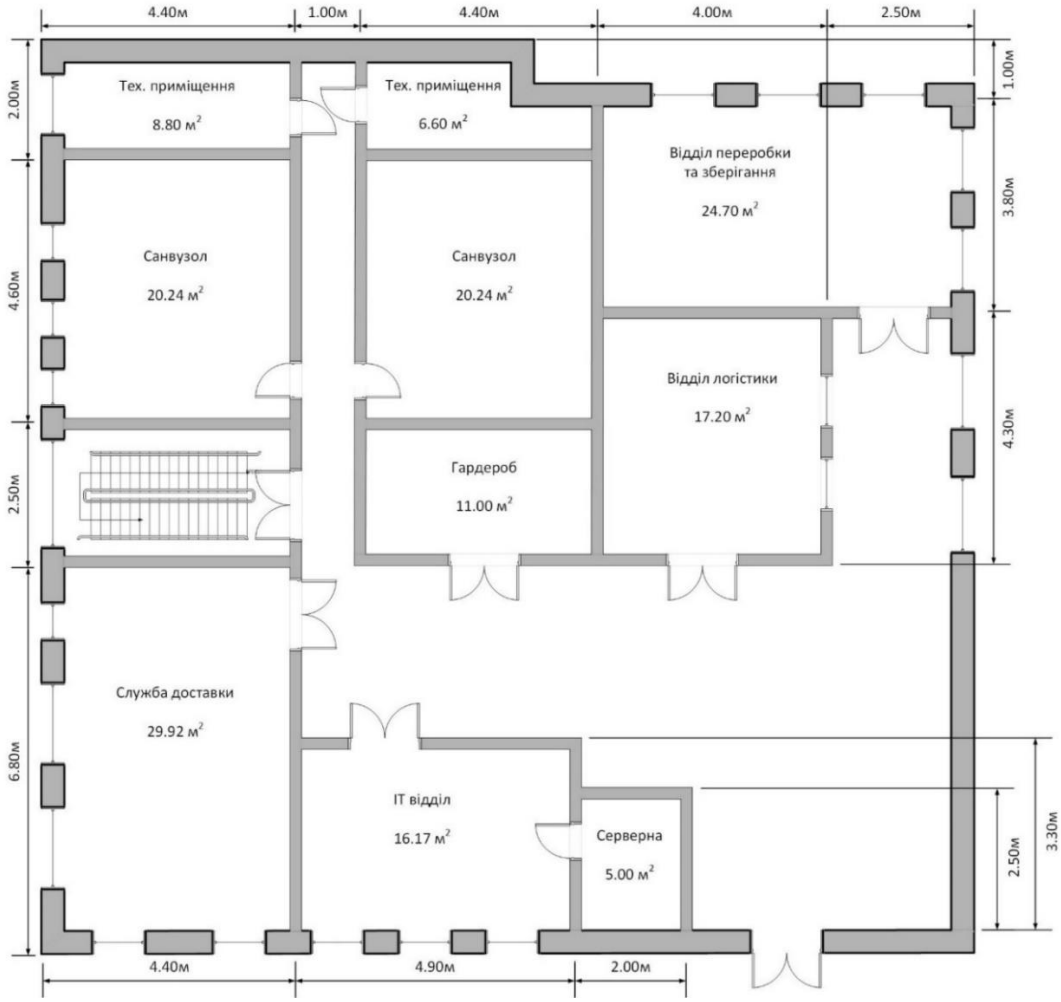

Рисунок 1.5 – Структурна схема розміщення підрозділів підприємства на першому поверсі

Структурна схема розміщення підрозділів підприємства на другому поверсі (див. рис. 1.6).

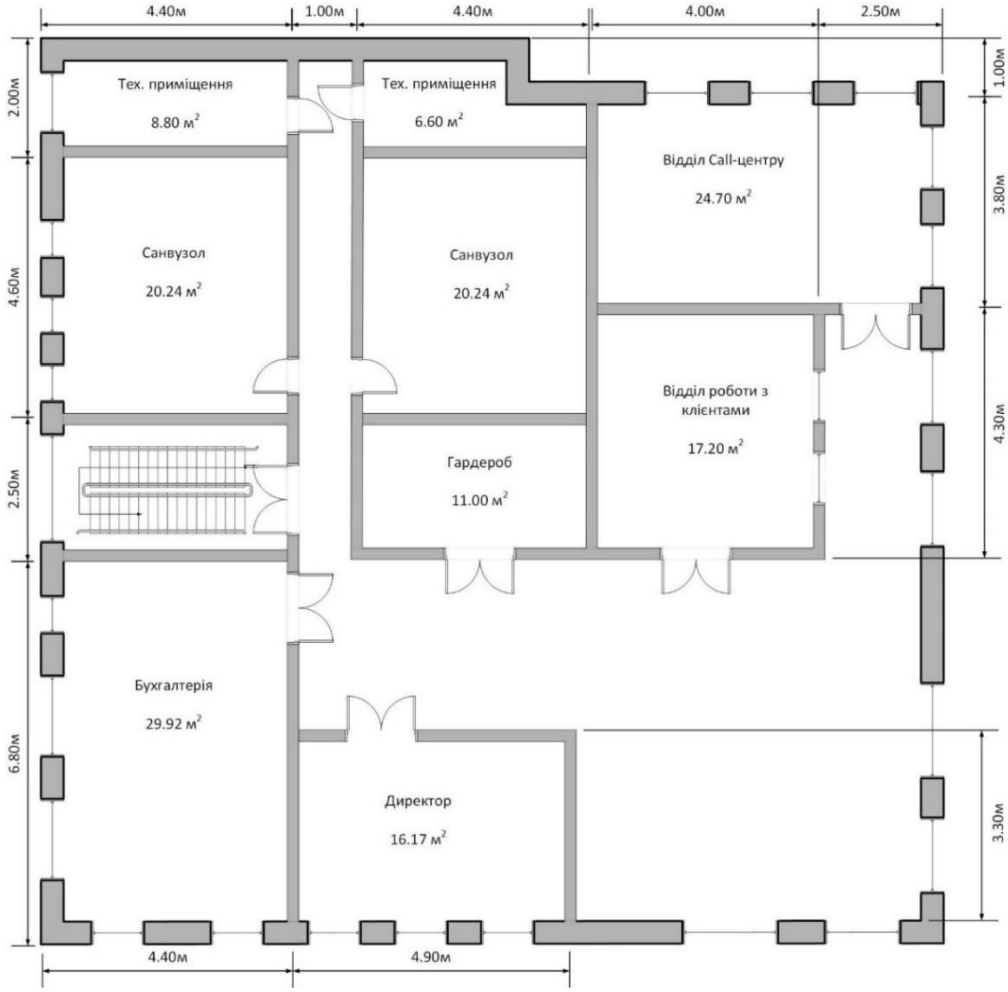

Рисунок 1.6 – Структурна схема розміщення підрозділів підприємства на другому поверсі

1.4 Принципи, технічні засоби та математичні методи інформаційного забезпечення

Існують два підходи до проєктування корпоративної мережі.

Перший підхід базується на використанні стандартних рішень, які пропонують відомі компанії (Cisco, HP, Microsoft і т.д.). Використання цього підходу характеризується невеликими витратами на проєктування та впровадження, але, створена таким чином мережа, може не повною мірою задовольнити потреби закладу.

Другий підхід базується на використанні як стандартних рішень, так і впровадження унікальних розробок, що дозволяють максимально адаптувати мережу та задовільнити потреби закладу.

Першими кроками є аналіз центрів обробки даних та системи зберігання, а також розгляд системи передачі інформації між комп'ютерами.

Наступний крок це вивчення використання мережевих операційних систем, які відповідають за роботу програм на комп'ютерах та забезпечують доступ до ресурсів.

3-ох рівнева ієрархічна модель Cisco розроблена бути надійною, масштабованою та високоефективною мережевою конструкцією (див. рис. 1.7). Три рівня ієрархії складаються з: рівень ядра, рівень доступу та рівень хостів. Кожен рівень має свої унікальні характеристики та функціонал, що дозволяє ще більше спростити процеси в мережі.

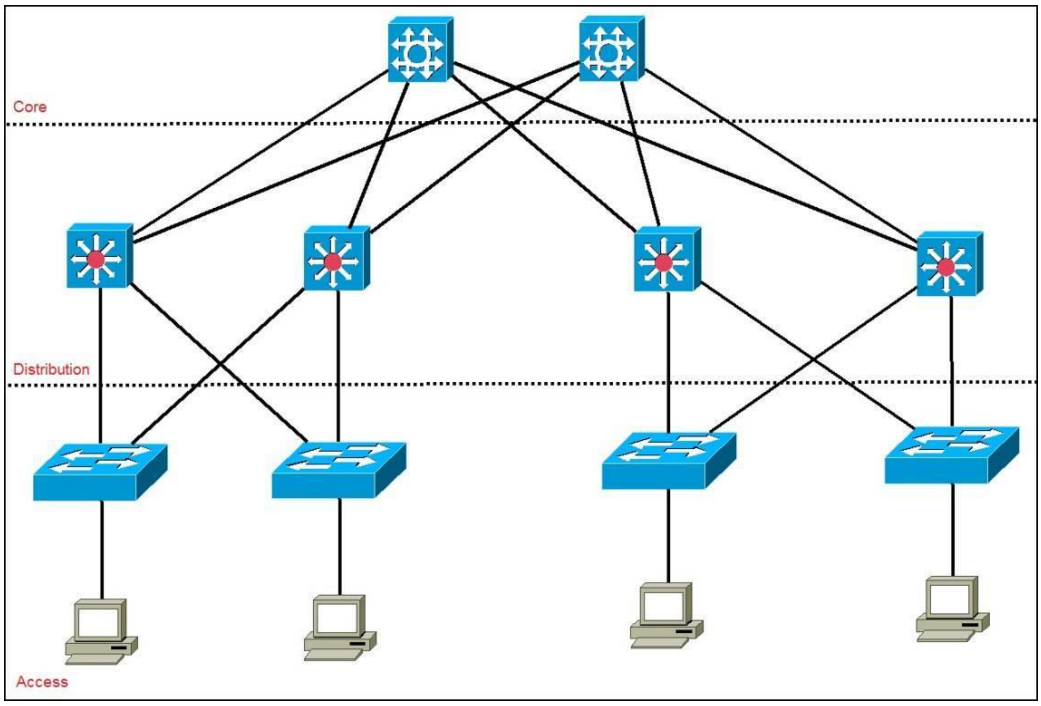

Рисунок 1.7 – Ієрархічна модель Cisco

Рівень ядра. Цей рівень представляє собою комплекс мережевих пристроїв, таких як маршрутизатори і комутатори, які забезпечують резервування каналів та швидку передачу даних між різними сегментами

мережі. Рівень ядра відповідає за ефективну та надійну передачу великого об’єму трафіку мережі.

Рівень доступу. Основним завданням цього рівня є забезпечення маршрутизації, фільтрації та доступу до WAN, а також візуалізація зв’язку між рівнями доступу та ядра.

Рівень хостів. Цей рівень слугує для підключення робочих станцій та серверів до мережі компанії. В більшості випадків рівень доступу представлений в мережі комутаторами другого рівня та точками доступу.

1.5 Огляд існуючих інженерних рішень КС в галузі

Amazon Robotics (раніше Kiva Systems) є провідним постачальним автоматизованих рішень складської логістики (див. рис. 1.8). До складу їх системи включають роботизовані пристрої і програмне забезпечення, за допомогою якого вантаж, що зберігається, переміщується та обробляється, у більш ефективний спосіб зі збільшення продуктивності роботи складського приміщення.

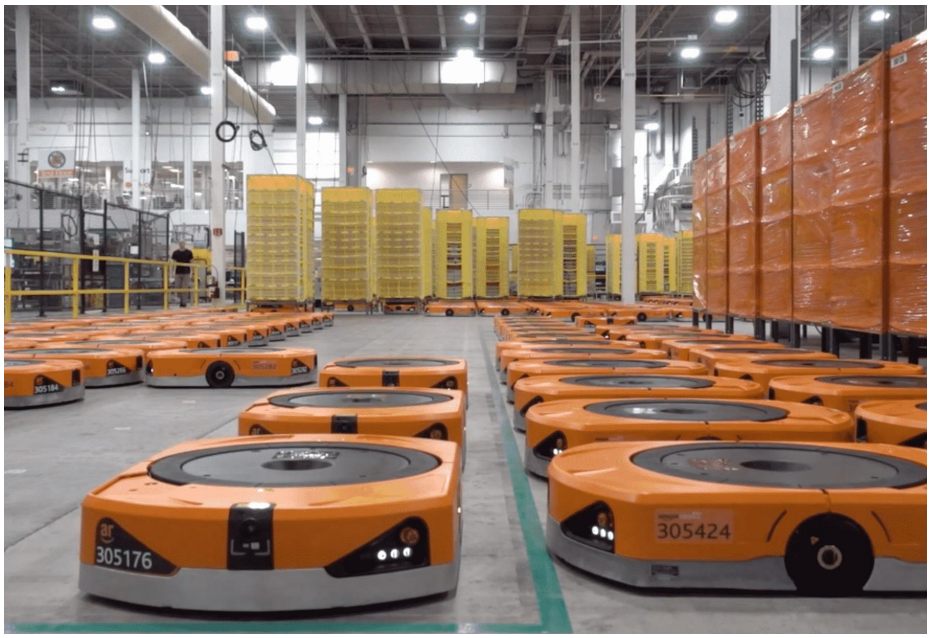

Рисунок 1.8 – Складське приміщення з Amazon Robotics

Основою системи Amazon Robotics є роботи, які здійснюють переміщення підлогою складу, піднімаючі полиці із вантажем та доставляючи їх до робочих станцій, де працівники складу виконують

необхідні дії, такі як сканування, збірка ти маркування. Цей процес автоматизовано та оптимізовано за допомогою ПЗ, що керує роботами, розподілом завдань та моніторингом роботи системи.

Особливістю роботи системи Amazon Robotics є здатність швидкого адаптування до змін попиту та обсягу роботи. Роботи можуть працювати ефективно працювати в різних умовах та навантаженнях, що, в свою чергу, забезпечує точну, швидку та ефективну роботу підприємства, навіть у період великого попиту.

Warehouse Execution Software (WES) від компанії Honeywell Intelligrated є інтегрованою програмною системою, що спрямована на оптимізацію та автоматизацію управління складськими операціями. (див. рис. 1.9).

Рисунок 1.9 – Складське приміщення з Warehouse Execution Software

Однією з функцій WES є оптимізація руху вантажних потоків в середині складського приміщення. Це включає в себе автоматизовану маршрутизацію та назначення завдань, для забезпечення оптимального розташування та пересування вантажу від приймального відділення до зони відвантаження. Відповідне ПЗ дає змогу динамічно реагувати на зміни в

роботі, а також надходження нового вантажу, з метою максимального використання ресурсів.

Дана система забезпечує контроль інвентаризації, відстеження вантажу та оптимізацію його розташування в складському приміщенні. Вона підтримує використання автоматизованих підйомників, конвеєрів та сортувальних систем, що дозволяє підвищити швидкість та ефективність складських операцій.

У багатьох галузях використовуються різні роботи та спеціалізовані маніпулятори, які здатні виконувати широкий спектр завдань.

Маніпулятори німецької компанії Kuka (див. рис. 1.10) – це роботи, що мають шість осей, доступних в різних розмірах та з різною вантажопідйомністю. Вони застосовуються під час виконання завдань, які потребують високої точності траєкторії (зварювання, склеювання та інші). Роботи Kuka з меншою вантажопідйомністю також відмінно справляються з легкими операціями, такими як тестування компонентів, монтаж дрібних деталей, шліфування, полірування, складання, а також завантаження та розвантаження машин.

Рисунок 1.10 – Маніпулятор Kuka

Датський маніпулятор UR3 (див. рис. 1.11) від компанії Universal Robots активно застосовується в дослідницьких, фармацевтичних, сільськогосподарських, електронних та технологічних сферах. Він відмінно справляється з такими завданнями монтажу дрібних деталей, зварювання, фарбування тощо.

Рисунок 1.11 – Маніпулятор UR3

1.6 Завдання і мета роботи

Метою кваліфікаційної роботи є розробка кіберфізичної системи розумного складу (далі Підсистеми 2) з детальним опрацюванням побудови та налаштуванням корпоративної мережі (далі Підсистема 1) логістичного підприємства (далі Система).

Для вирішення поставленої мети в кваліфікаційній роботі вирішуються наступні завдання:

− аналіз об’єкту впровадження;

− обґрунтування вибору мережевої архітектури;

− формування технічних вимог для розробки Системи;

− розробка специфікації апаратних засобів для реалізації функцій Системи;

– аналіз мережевого трафіку;

– розробка фізичної та логічної топології Підсистеми 1;

– розрахунок схеми адресації Підсистеми 1;

– налаштування та тестування мережевого обладнання;

– підбір обладнання для дистанційного керування маніпулятором;

– розробка програмного інтерфейсу дистанційного керування маніпулятором.

1.7 Визначення можливих напрямків рішення поставлених завдань

На цьому етапі топологія мережі логістичного підприємства має наступний вигляд (див. рис. 1.12).

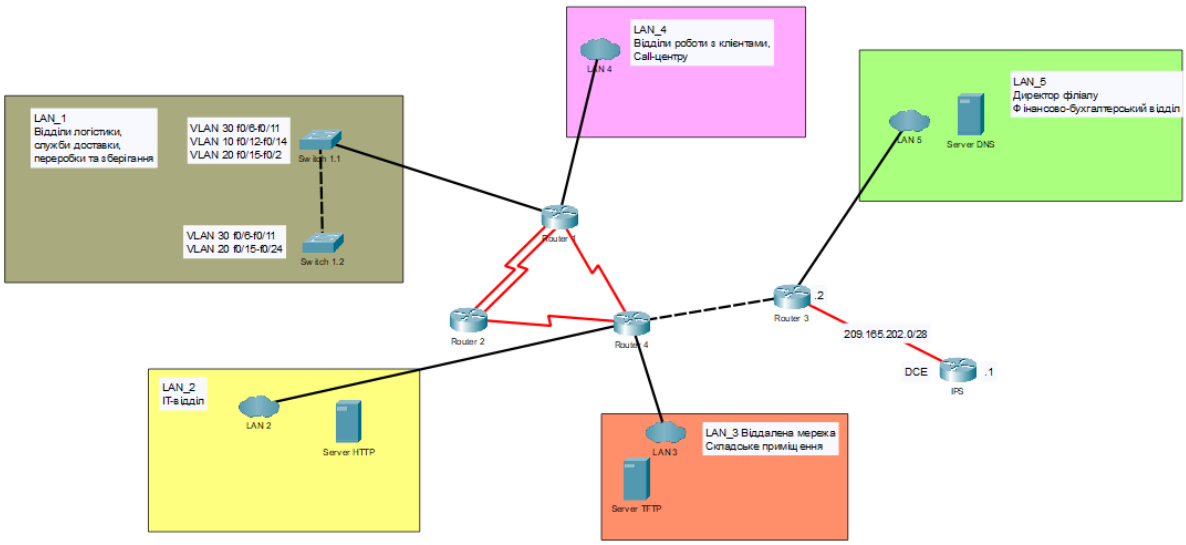

Рисунок 1.12 – Топологія мережі

Для вирішення поставлених завдань стосовно розробки комп'ютерної системи логістичного підприємства з реалізацією побудови та налаштування корпоративної мережі і підсистемою розумного складу можуть бути використані наступні напрямки рішення:

1) Вибір мережевої архітектури мережі.

Для Підсистеми 1 буде обрано розподілу архітектуру. Розподілена архітектура передбачає розподіл ресурсів мережі поміж взаємопов'язаними

вузлами. Мережа також передбачає ознаки централізованої архітектури, де серверне обладнання надає ресурси та послуги кінцевим користувача.

2) Вибір кабельної системи мережі.

Для забезпечення зв'язку між постачальником інтернет послуг та приміщенням офісу і складським приміщенням потрібно обрати оптоволоконний кабель. За допомогою оптоволоконного кабелю можна передавати великі об'єми даних на великі відстані з високою швидкістю, не зважаючи на електромагнітні перешкоди.

Для забезпечення зв'язку в середині мережі потрібно обрати кабель витої пари, що є стандартом для побудови Ethernet-мереж та забезпечує достатню швидкість передачі даних з відповідним рівнем надійності.

Для забезпечення зв'язку маніпуляторів з IoT-шлюзом потрібно обрати технологію Wi-Fi. За допомогою цієї технологіє буде можливе дистанційне керування роботою маніпуляторів з мобільного пристрою.

3) Аналіз мережевого трафіку.

Для аналізу мережевого трафіку потрібно встановити або мережеві монітори, що можуть моніторити мережевий трафік, або спеціальні програми (Wireshark), для розшифрування та аналізу трафіку.

4) Вибір способу управління мережею.

В логістичному підприємстві буде задіяне локальне управління мережею. Працівники IT- відділу будуть завжди знаходитись в офісному приміщенні, що дасть змогу відповідати за налаштування, моніторинг та оперативно вирішувати проблеми мережі.

5) Розробка фізичної та логічної топології мережі.

При розробці буде створено п'ять окремих підмереж, одна з яких буде віддаленою. Буде використано динамічний протокол маршрутизації мережі OSPF, впроваджено VPN для можливості віддаленого доступу.

6) Налаштування мережевого обладнання.

Маршрутизатори будуть мати відповідний рівень захисту доступу, використовувати складні паролі та шифрування віддаленого доступу за допомогою протоколу SSH.

Комутатори будуть мати відповідних рівень захисту доступу та технологію розподілення фізичної мережі на віртуальні локальні мережі VLAN.

7) Реалізація підсистеми розумного складу.

При реалізації підсистеми розумного складу слід обрати та встановити відповідні модулі розширення та керування. s

Розробити програмний інтерфейс дистанційної взаємодії персоналу із маніпулятором використовуючи мобільний пристрій.

Впровадити підсистему розумного складу до вже побудованої корпоративної мережі.

8) Тестування мережі в цілому та кожного компонента окремо.

Необхідно провести тестування функціональності, навантаження та безпеки мережі.

2 РОЗРОБКА АПАРАТНОЇ ЧАСТИНИ КОМП'ЮТЕРНОЇ СИСТЕМИ ПІДПРИЄСТВА

2.1 Технічні вимоги до Системи

Система організації інформаційного обміну та дистанційного керування маніпулятором включає в себе дві підсистеми.

Перша підсистема спрямована на забезпечення безпечного обміну, збереження та передачі інформації між структурними відділами підприємства і за його межі.

Друга підсистема спрямована на забезпечення підвищення ефективності виконання складських операцій (розвантаження, завантаження, переміщення) за допомогою мобільного застосунку дистанційного керування роботою маніпуляторів.

2.1.1 Вимоги до Системи в цілому

2.1.1.1 Вимоги до структури і функціонування Системи

2.1.1.1.1 Перелік підсистем, їх призначення та основні характеристики

Система складається із двох підсистем: Підсистема 1, призначена для організації середовища передачі інформації між структурними відділами та за межі підприємства і Підсистема 2, призначена для організації дистанційного керування роботою маніпулятору з мобільного пристрою та взаємодії із Підсистемою 1.

Підсистема 1 повинна мати наступні локальні підмережі, відповідно до організаційної структури підприємства та топологічної структури (Додаток А):

− LAN_1: відділи логістики, служби доставки та переробки і зберігання вантажу;

− LAN_2: IT-відділ;

− LAN_3: приміщення розумного складу;

− LAN_4: відділи роботи з клієнтами та Call-центру;

− LAN_5: керівництво та фінансово-бухгалтерський відділ.

Локальна підмережа LAN_1 повинна забезпечувати комунікацію до 14 вузлів.

Локальна підмережа LAN_2 повинна забезпечувати комунікацію до 109 вузлів. В цій підмережі повинен бути HTTP-сервер, для надання веб-сторінки.

Локальна підмережа LAN_3, яка є віддаленою, повинна забезпечувати комунікацію до 87 вузлів. В цій підмережі повинен бути TFTP-сервер, для передачі файлів.

Локальна підмережа LAN_4 повинна забезпечувати комунікацію до 94 вузлів.

Локальна підмережа LAN_5 повинна забезпечувати комунікацію до 40 вузлів. В цій підмережі повинен бути DNS-сервер, для перетворення доменних імен в IP-адреси та навпаки.

Врахувати, що середня інтенсивність трафіку найбільшої мережі LAN_2, повинна бути не меншою за 97 кадрів/с.

Підсистема 2 повинна складатись з:

− десяти маніпуляторів, кожен з яких оснащений платою розширення для керування сервоприводами, а також Wi-Fi модулем для дистанційного керування;

− IoT-шлюзу, для підключення маніпуляторів;

− мобільних пристроїв, для підключення оператору до IoT-шлюзу і подальшого дистанційного керування маніпулятором;

− мережевого обладнання, для взаємодії Підсистеми 2 з Підсистеми 1.

Всі підсистеми повинні взаємодіяти між собою, та створювати одну Систему, що зберігає працездатність і забезпечує відновлення своїх функцій при виникненні незвичайних ситуацій.

Загальна мережева інфраструктура підприємства повинна бути гнучкою та масштабованою для забезпечення можливості розширення і подальшої модернізації, враховуючи зростання потреб користувачів та розвитку нових технологій.

Канали передачі даних мають бути розраховані на максимальну завантаженість при пересиланні інформації мережею, що включає:

− середня довжина вихідного повідомлення в найбільшій мережі не повинна перевищувати 650 байт;

− затримка передачі пакету в найбільшій мережі не повинна перевищувати 6 мс;

− пропускна здатність повинна забезпечувати передачу даних зі швидкість не менше 50 МБ/с.

Для забезпечення відповідного рівня безпеки інформації та захисту її від несанкціонованого доступу повинні бути розробити та впроваджені заходи та стратегії безпеки.

2.1.1.1.2 Вимоги до способів і засобів зв'язку для інформаційного обміну між компонентами Системи

При створені Підсистеми 1 повинна бути використана технологія Ethernet, для взаємодії пристроїв один з одним шляхом передачі пакетів даних за допомогою кабелів.

Віддалена підмережа LAN_3 повинна взаємодіяти з основною за допомогою технології VPN.

В підмережі LAN_1 повинна використовувати технологію логічного розділення фізичної мережі на віртуальні мережі VLAN.

Для виявлення та організації прокладання маршрутів в мережі повинен використовуватись протокол динамічної маршрутизації OSPF.

Система повинна бути підключеною до глобальної мережі Інтернет через постачальника інтернет послуг (провайдера) використовуючи технологію NAT.

При створені Підсистеми 2 повинна бути використана технологія Wi-Fi, для взаємодії пристроїв один з одним шляхом передачі пакетів даних за допомогою електромагнітних хвиль.

Для віддаленого керування маніпулятором повинен використаний програмний інтерфейс (API).

Для взаємодії Підсистеми 2 та Підсистеми 1 повинна використовуватись технологія Ethernet.

2.1.1.1.3 Вимоги до режимів функціонування Системи

Вимоги до режимів роботи функціонування Системи:

− нормальний режим. Система повинна забезпечувати цілодобову безперебійну роботу підприємства;

− режим навантаження. Система повинна забезпечувати безперебійну роботу підприємства під час одночасної оброки, прийому та відправлення не менше 1000 товарів на годину;

− резервний режим. Активне обладнання Системи повинно забезпечувати безперебійну роботу підприємства під час відсутності електроживлення;

− профілактичний режим. Цей режим передбачає профілактику, ремонт або заміну компонентів Системи, під час проведення якого, відбувається обмеження функціональності Системи;

− режим відновлення після аварій. Цей режим передбачає відновлення роботи Системи, після або в разі її збою чи аварійної ситуацій.

2.1.1.1.4 Вимоги до діагностування Системи

Діагностування Системи повинно проводитись не рідше одного разу в пів року.

Вимоги до діагностування Підсистеми 1:

− перевірка з’єднань та підключень. Перевірка фізичного з’єднання пристроїв мережі та виявлення можливих проблем в кабельній системі;

− перевірка IP-адресації та мережевих налаштувань. Перевірка правильності призначення IP-адресації мережим пристроям та налаштувань мережевих пристроїв;

− діагностика пропускної здатності. Вимірювання швидкості передачі даних різних підмереж з метою виявлення можливого перевантаження ліній;

− перевірка безпеки. Перевірка наявності антивірусного ПЗ;

− перевірка мережевих служб. Перевірка доступу до веб-серверу, можливості обміну файлами та віддаленого доступу.

Вимоги до діагностування Підсистеми 2:

− перевірка з’єднань та підключень. Перевірка фізичного з’єднання пристроїв та виявлення можливих проблем;

− калібрування сервоприводів. Перевірка кута обертання з метою забезпечення точності та стабільності;

− перевірка мережевого з’єднання. Перевірка доступу до IoT-шлюзу, можливості підключення маніпуляторів.

2.1.1.1.5 Перспективи розвитку Системи

Перспективи розвитку Підсистеми 1:

− системи безпеки. Використання нових методів шифрування, авторизації та аутентифікації;

− масштабування. Розширення корпоративної мережі для забезпечення росту бізнесу та збільшення обсягу передачі даних.

Перспективи розвитку Підсистеми 2:

− використання камери з маніпулятором. За допомогою впровадження камери до маніпулятору стане можливим відслідковування дій оператора в реальному часі;

− додавання елементів розумного складу. Впровадження розумних сортувальних станцій та систем автоматизованого зберігання товарів.

2.1.1.2 Вимоги до показників призначення Системи

Вимоги до показників призначення Підсистеми 1:

− Підсистема 1 повинна надавати можливість доступу до мережі Інтернет;

− Підсистема 1 повинна надавати можливість віддаленого підключення з використання технології VPN;

− Підсистема 1 повинна надавати можливість обміну даними між структурними підрозділами підприємства;

− Підсистема 1 повинна надавати можливість реалізації веб-сторінки підприємства;

− Підсистема 1 повинна захищати дані від несанкціонованого доступу та шкідливого ПЗ.

Вимоги до показників призначення Підсистеми 2:

− Підсистема 2 повинна забезпечувати збір, передачу та обробку даних з плати керування маніпулятору;

− Підсистема 2 повинна забезпечувати підключення мобільного пристрою для керування роботою маніпулятор, використовуючи технологію Wi-Fi;

− Підсистема 2 повинна забезпечувати керування роботою маніпулятору із використання програмного інтерфейсу (API);

− Підсистема 2 повинна забезпечувати обмін інформацією з Підсистемою 1 з використанням маршрутизатору.

2.1.1.3 Вимоги до експлуатації, технічного обслуговування, ремонту і збереження

2.1.1.3.1 Умови і регламент (режим) експлуатації, що повинні забезпечувати використання технічних засобів (ТЗ) Системи з заданими технічними показниками

Підсистема 1 повинна забезпечувати свою функціональність при впливі наступних факторів:

− температура повітря: від +23°С до +25° С;

− відносна вологість повітря: від 40% до 60%;

− атмосферний тиск: від 95 кПа до 105 кПа;

− максимальна запиленість приміщення при розмірі часток більше 3 мкм: 0,75 мг/куб. м;

− напруженість зовнішнього електричного поля: <0,3 В/м;

− напруженість зовнішнього магнітного поля: <5,0 А/м;

− частота вібрацій при амплітуді зсувів 0,1 мм: <25 Гц.

Підсистема 2 повинна забезпечувати свою функціональність при впливі наступних факторів:

− температура повітря: від +15°С до +27° С;

− відносна вологість повітря: від 35% до 70%;

− атмосферний тиск: від 95 кПа до 105 кПа;

− максимальна запиленість приміщення при розмірі часток більше 3 мкм: 0,75 мг/куб. м;

− напруженість зовнішнього електричного поля: <0,3 В/м;

− напруженість зовнішнього магнітного поля: <5,0 А/м;

− частота вібрацій при амплітуді зсувів 0,1 мм: <30 Гц.

В повітрі приміщення не повинно бути агресивних речовин, які можуть спричинити корозію.

Розміщення обладнання, ТЗ повинно відповідати вимогам техніки безпеки, санітарним нормам і вимогам пожежної безпеки.

2.1.1.3.1 Вимоги до параметрів мереж електроживлення

Вимоги до параметрів мереж електроживлення Системи:

− напруга живлення в мережі повинна бути 220 В ± 5% від номінальної напруги (згідно ДСТУ IEC 60038:2015);

− частота живлення в мережі повинна бути 50 Гц ± 0.2 Гц; [1]

− мережа електроживлення повинна бути надійною, для уникнення збоїв роботи Системи;

− електроживлення повинно бути стабілізовано, для уникненню пошкодження або виходу з ладу обладнання Системи;

− Система повинна мати резервні джерела безперебійного живлення у випадку відсутності електроживлення;

− Система повинна забезпечувати заземлення електричних компонентів (розеток, мережевого обладнання, серверного обладнання, маніпуляторів) для захисту від електричних перешкод.

Вимоги до параметрів живлення маніпуляторів Підсистеми 2:

− живлення від двох послідовно з’єднаних акумуляторних батарей типу 18650, ємністю 3400 мА/год та напругою 3.7 В кожна;

− живлення від блоку живлення постійної напругою від 7-15 В та струмом до 5 А.

2.1.1.3.2 Вимоги до кількості, кваліфікація обслуговуючого персоналу і режимам його роботи

Стабільна робота Системи повинна забезпечуватись ІТ-відділом, який складається із 5 працівників: завідувач відділу, два інженери-програмісти, інженер-електронник, системний адміністратор.

Завідувач відділу забезпечує виконання наступних обов’язків:

− організація роботи відділу та керування ним згідно встановлених процедур;

− планування та контроль виконання завдань відділу;

− ведення звітності перед керівною ланкою закладу;

− встановлення цілей та завдань;

− координація комунікації між членами відділу та іншими структурними підрозділами.

Інженер-програміст забезпечує виконання наступних обов’язків:

− впровадження ПЗ для виконання поставлених завдань;

− тестування ПЗ на відповідність виконання поставлених завдань;

− підтримка та вдосконалення існуючих програмних рішень;

− надання технічної підтримки користувачам пов’язану із роботою ПЗ системи.

− забезпечення сумісність програмних модулів та апаратного забезпечення;

Інженер-електронник забезпечує виконання наступних обов’язків:

− дослідження та впровадження електронних компонентів для виконання поставлених завдань;

− налагодження та технічне обслуговування електронних компонентів;

− вивчення та вдосконалення нових технології у сфері електроніки для покращення функціонування системи;

− виконання монтажу та підключення електронних компонентів до системи із забезпеченням їх сумісності;

− надання технічної підтримки користувачам пов’язану із апаратною частиною роботи системи.

Системний адміністратор забезпечує виконання наступних обов’язків:

− виконання управління мережевим та серверним обладнанням;

− забезпечення відповідного рівня захисту інформації в мережі;

− надання технічної підтримки користувачам пов’язану із мережевою частиною роботи системи;

− здійснення моніторингу та діагностики поточного стану системи;

− забезпечення регулярного резервного копіювання даних та відновлення даних у разі її втрати, спотворення або пошкодження;

− розробка інструменти для автоматизації рутинних завдань, для підвищення ефективності.

Завідувач відділу повинен мати вищу освіту у галузі менеджменту або суміжній галузі, а також досвід роботи на керівній посаді більше 2-ох років.

Додатковим вимогами для завідувача відділу є наявність технічних знань у галузі інформаційних технологій.

Інженер-програміст повинен мати вищу професійно-технічну освіту чи початкову технічну освіту, а також знання та досвід у роботі із ПЗ більше 1-го року.

Інженер-електронник повинен мати вищу професійно-технічну освіту у галузі електроніки або суміжній галузі, а також знання та досвід у роботі із електронними компонентами більше 1-го року.

Системний адміністратор повинен мати вище професійну освіту у галузі адміністрування мереж або суміжній галузі, а також знання та досвід у роботі із мережевим та серверним обладнанням більше 2-ух років.

Працівники ІТ-відділу повинні дотримуватись правил техніки безпеки, охорони праці, виробничої санітарії та пожежної безпеки, які застосовуються в закладі.

Завідувач відділу працює в режимі нормованого робочого тижня згідно графіку з понеділка по п’ятницю з 9:00 до 17:00 разом із обідньою перервою.

Інженер-програміст працює в режимі ненормованого робочого тижня згідно графіку з понеділка по п’ятницю з 9:00 до 13:00 разом із обідньою перервою.

Інженер-електронник працює в режимі нормованого робочого тижня згідно графіку з понеділка по п’ятницю з 9:00 до 17:00 разом із обідньою перервою.

Системний адміністратор працює в режимі ненормованого робочого тижня згідно графіку з понеділка по п’ятницю з 9:00 до 13:00 разом із обідньою перервою.

2.1.1.3.3 Вимоги до складу, розміщенню й умовам збереження комплексу запасних виробів та приладів

Комплекс запасних виробів та приладів повинні зберігатися в технічному приміщенні офісної будівлі, доступ до якого мають відповідальні фахівці ІТ-відділу.

Технічне приміщення повинно мати контроль температури в межах від 22°C до 25°C та відсотку відносної волості повітря в межах від 40% до 60%.

Комплекс запасних виробів та приладів повинні складатися з:

− двох маршрутизаторів;

− трьох комутаторів;

− одного серверу;

− п’яти ПК;

− 100 м кабелю витої пари типу UTP;

− 15 конекторів типу RJ-45;

− двох блоків живлення маніпулятору напругою від 7 В до 15 В;

− одного модулю бездротового керування маніпулятором;

− двох акумуляторних батарей типу 18650, ємністю 3400 мА/год та напругою 3.7 В кожна;

− двох плат розширення для керування роботою сервоприводів та Wi-Fi модулю.

2.1.1.3.4 Вимоги до регламенту обслуговування

Виконання функціональних завдань Системи повинно бути розраховане на цілодобовий режим роботи, з відключенням необхідного сегменту для здійснення профілактичних заходів, але не частіше ніж 1 раз на пів року.

До технологічного обслуговування компонентів Системи повинно входити:

− зовнішній та внутрішній огляд;

− прочистка компонентів від пилу та бруду;

− протяжка різьбових з’єднань;

− заміна теплопровідного композиту;

− перевірка контактних з’єднань;

− перевірка та діагностика програмного забезпечення.

Відновлення працездатності ТЗ повинно проводитися відповідно до інструкцій розробника і постачальника ТЗ та документами по відновленню працездатності ТЗ і завершуватися проведенням їх тестування.

Виконання ремонтних робіт системи повинно виконуватися інженерами ІТ-відділу під наглядом завідувача.

ТЗ Системи, які вийшли з ладу повинні змінюватися новими. Складанням планів на закупівлю та закупівлею складових елементів системи займається завідувач ІТ-відділу.

2.1.1.4 Вимоги до патентної чистоти

Програмне забезпечення та ТЗ, що використовується в кіберфізичній системі, не повинно порушувати патентної чистоти.

При розробці кіберфізичної системи повинно бути використане ліцензійне ПЗ сертифіковане для використання на території України.

При розробці програмного інтерфейсу керування маніпулятором повинно бути використане середовище розробки Android Studio.

При розробці набору команд керування маніпулятором повинно бути використане середовище розробки Arduino IDE.

2.1.1.5 Додаткові умови

2.1.1.5.1 Вимоги до активного обладнання

Вимоги до активного обладнання повинні включати такі аспекти:

− надійність, що забезпечує стабільну роботу та забезпечує відповідність стандартам якості;

− сумісність, що забезпечує з’єднання з іншими пристроями в Системі;

− безпека, що забезпечує широкий набір механізмів безпеки, включаючи шифрування, віртуальні приватні мережі VPN, фаєрвол, контроль доступу та ін.;

− кількість портів та запас, що забезпечує потреби мережі у випадку розширення;

− розташування, активне обладнання повинно бути встановлено в технічному приміщенні, що знаходиться на оптимальній відстані від інших вузлів мережі, для мінімізації довжини кабельних трас;

− плата керування роботою маніпулятору повинна мати достатню кількість портів вводу/виводу для підключення сервоприводів, плати розширення та модулю дистанційного керування;

− IoT-шлюз Підсистеми 2 повинен підтримувати одночасне підключення десяти маніпуляторів.

2.1.1.5.2 Вимоги до кабель-каналів, інформаційним та електричним розеткам

Вимоги до кабель-каналів повинні включати:

− кабель-канали повинні мати захист від фізичних пошкоджень;

− кабель-канали повинні бути легко доступними для розміщення та обслуговування;

− кабель-канали повинні бути встановленими з урахування мінімального впливу електромагнітних перешкод;

− розміри кабель-каналів повинні бути не меншими ніж 25х25 мм для розміщення кабелів з урахування можливого розширення;

− кабельні траси повинні бути правильно побудованими для забезпечення належної організації, відповідати чиним стандартам, захищеними від впливу зовнішніх факторів;

− кабель-канали повинні відповідати нормам пожежної безпеки.

Вимоги до інформаційних розеток Підсистеми 1 повинні включати:

− сумісність зі стандартами передачі даних Ethernet Cat5e, Cat6, Cat6a;

− маркування та колірне кодування типу з’єднання кабелю витої пари;

− вологостійкість на рівні IP23;

− відповідність нормам пожежної безпеки.

Вимоги до інформаційних розеток Підсистеми 2 повинні включати:

− сумісність зі стандартами передачі даних Ethernet Cat5e, Cat6, Cat6a;

− маркування та колірне кодування типу з’єднання кабелю витої пари;

− вологостійкість на рівні IP34;

− відповідність нормам пожежної безпеки.

Вимоги до електричних розеток Підсистеми 1 повинні включати:

− вологостійкість на рівні IP23;

− перевірка не менші ніж раз на рік;

− забезпечення заземлення;

− наявність захисних шторок, для запобігання випадкового контакту;

− відповідність нормам пожежної безпеки.

Вимоги до електричних розеток Підсистеми 2 повинні включати:

− вологостійкість на рівні IP34;

− перевірка не менші ніж раз на пів року;

− забезпечення заземлення;

− наявність захисних шторок, для запобігання випадкового контакту;

− відповідність нормам пожежної безпеки.

2.1.1.5.3 Вимоги до комунікаційного обладнання і його розташування

Вимоги до комунікаційного обладнання:

− комунікаційне обладнання повинно знаходитись в спеціальній комутаційній шафі, що має відповідний рівень захисту від вологи та пилу. Комутаційна шафа повинна знаходитись в технічному приміщенні IT-відділу;

− комутаційна шафа повинна мати достатні розміри для розміщення комунікаційного обладнання;

− комутаційна шафа повинна бути стійкою до вібрацій;

− комутаційна шафа повинна мати можливість легкого доступу для обслуговування;

− обладнання, розташоване в середині комутаційної шафи, повинно бути надійно закріплено, розташовано для легкої ідентифікації та мати додатковий простір задля вентиляції та вільного доступу;

− корпус комутаційної шафи повинен бути заземлений;

− в технічному приміщенні розташування комутаційної шафи повинна бути забезпечена вентиляція та наявність кондиціонеру.

2.1.1.5.4 Вимоги до однорідності

Вимоги до однорідності повинні включати:

− використання єдиного типу кабелю для всієї мережі (для витої пари необхідно використовувати кабель не нижче 6 категорії);

− використання єдиного типу конекторів типу RJ-45 для підключення кабелю витої пари;

− використання мережевого обладнання фірми Cisco;

− використання стандартних протоколів TCP/IP, Ethernet, Wi-Fi.

2.1.1.5.5 Вимоги до надійності

Надійність Системи визначається на основі параметрів надійності окремих компонентів.

Надійність компонентів Системи (мережеве обладнання, кабельні траси, кінцеві вузли тощо) базується на гарантованих виробником технічних характеристик, що визначені у відповідних гарантійних паспортах.

Система повинна гарантувати збереження функціональності у разі виникнення перебоїв у живленні. Повинні бути резервні джерела безперебійного живлення такої потужності, щоб забезпечити можливість праці маніпуляторів не менше ніж на годину.

Серверне обладнання Системи має забезпечувати безперервну роботу та резервне копіювання даних із можливістю їх відновлення.

ПЗ Системи повинна забезпечувати безперервну роботу та швидке відновлення після виникнення непередбачуваних ситуацій.

Всі елементи системи повинні підлягати міжнародним стандартам, а також бути взаємозамінними.

2.1.1.5.6 Вимоги до безпеки

Вимоги з безпеки Системи визначають наступні правила:

− монтаж та експлуатація електротехнічного обладнання повинні відбуватися відповідно до інструкцій виробника;

− не експлуатувати електротехнічне обладнання, якщо порушені їх цілісність чи пошкоджений корпус;

− не користуватися нестандартним електротехнічним обладнанням;

− не розміщувати електричні прилади біля зон активного виділення тепла;

− не торкатися мокрими руками та не витирати вологою ганчіркою електричні кабелі, вимикачі, інші електроприлади які ввімкнені в електромережу;

− не залишати без нагляду ввімкнуті електричні прилади;

− повинен бути передбачений доступ до спільного електроду системи заземлення.

Умови роботи персоналу повинні відповідати державним санітарним правилам і нормам за ДСанПІН 3.3.2.007-98 «Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин».

На кожному поверху у офісі повинні бути комплект протипожежного обладнання та аптечки надання першої домедичної допомоги. Такий же набір повинен бути, безпосередньо у складському приміщенні.

2.1.1.5.7 Вимоги до захисту інформації від несанкціонованого доступу

Заходи захисту інформації повинні відповідати наступним вимогам:

− пароль для запуску системи повинен бути не менше 8 символів;

− пароль для входу в налаштування завантаження системи повинен бути не менше 8 символів

− паролі повинні змінюватись не рідше одного разу на три місяці;

− сервер не повинен виключати можливість підключення зовнішніх носіїв;

− реєстрація та контроль дій користувачів, обслуговуючого персоналу або сторонніх осіб;

− резервування ТЗ та дублювання носіїв інформації;

− застосування захисних фільтрів та криптографічних засобів захисту;

− систематичне оновлення ПЗ.

Вимоги щодо захисту інформації від несанкціонованого доступу повинні бути задокументовані. Документація повинна включати:

− політику інформаційної безпеки;

− інструкції для користувачів;

− журнал реєстрації подій;

− порядок дій у разі виникнення інцидентів інформаційної безпеки.

2.1.2 Вимоги до функцій (задач), виконуваним Системою

Вимоги до функцій, виконуваним Підсистемою 1:

− Підсистема 1 повинна забезпечувати безперебійну роботу веб-сторінки, серверу передачі файлів та DNS-серверу;

− Підсистема 1 повинна забезпечувати стабільний мережевий трафік відповідно до пропускної здатності;

− Підсистема 1 повинна надавати можливість резервного копіювання даних;

− Підсистема 1 повинна забезпечувати можливість масштабування для подальшого розвитку;

− Підсистема 1 повинна забезпечувати впровадження заходів безпеки для захисту інформації.

Вимоги до функцій, виконуваним Підсистемою 2:

− Підсистема 2 повинна забезпечувати стабільне підключення маніпуляторів до IoT-шлюзу з використання протоколу WebSoket;

− Підсистема 2 повинна забезпечувати керування роботою маніпулятору (переміщення захоплювачу вліво/вправо, вгору/вниз, вперед/назад та захват/утримання товарів) через програмний інтерфейс;

− Підсистема 2 повинна забезпечувати взаємодію IoT-шлюзу із Підсистемою 1 через маршрутизатор.

Вимоги під час розробки адресації пристроїв:

− перші доступні для використання IP-адреси повинні бути назначені інтерфейсам та підінтерфесам маршрутизаторів в мережах LAN;

− перші доступні для використання IP-адреси повинні бути назначені комутаторам в мережах LAN;

− серверам повинні бути назначені IP-адреси згідно правила: перша доступна для використання IP-адреса в мережі + 9 + 17. [2]

Вимоги під час базового налаштування мережевого обладнання:

− пристроям повинні бути назначені імена згідно правила: Прізвище студента_тип пристрою та номер пристрою;

− всі пристрої повинні мати пароль cisco для входу до консолі та віртуальних ліній;

− всі пристрої повинні мати пароль class для входу до привілейованого режиму;

− паролі, які зберігаються у відкритому вигляді, повинні бути зашифровані;

− всі пристрої повинні мати банер MOTD;

− на всіх віртуальних лініях vty повинен використовуватись протокол SSH;

− на маршрутизаторах повинен бути створений користувач з паролем згідно правила: група_прізвище з паролем admincisco;

− на маршрутизаторах, в якості імені домену, повинно використовуватись ім’я пристрою. Для шифрування даних повинен використовуватись RSA ключ довжиною в 1024 біт;

− на DCE-інтерфейсах маршрутизаторів повинна бути встановлена тактова частота 128000. [2]

Вимоги під час налаштування маршрутизації:

− для маршрутизації в мережі повинен використовуватись протокол динамічної маршрутизації OSPF;

− на маршрутизаторах повинні бути оголошені безпосередньо підключені мережі та вимкнено поширення оновлення маршрутизації на інтерфейси локальних мереж;

− для мереж VLAN повинен бути налаштований статичний маршрут, який буде оголошений іншим маршрутизаторам;

− еталонна пропускна спроможність для розрахунку вартості за замовчуванням Gigabit інтерфейсів повинна дорівнювати 1000;

− пропускна спроможність на послідовних інтерфейсах повинна дорівнювати 128 Кб/с, вартість метрики повинна дорівнювати 7500.

− на граничному маршрутизаторі повинен бути налаштований маршрут за замовчуванням з подальшим розповсюдженням його через оновлення маршрутизації;

− на граничному маршрутизаторі маршрут за замовчуванням повинен включати ручне підсумовування. [2]

Вимоги під час налаштування маршрутизаторів на підтримку служби AAA:

− для перевірки відключень до віртуальних ліній vty повинна використовуватись локальна база даних користувачів;

− для доступу до консолі повинна використовуватись аутентифікація з використанням протоколу Radius, при відсутності – локальну базу даних;

− Radius-сервер повинен використовувати ключове слово radius123; в якості облікового запису користувачів повинно використовуватись ім’я пристрою з паролем admin123. [2]

Вимоги під час налаштування роботи Інтернету:

− повинен бути встановлений один провайдер послуг до Інтернету (ISP);

− для виходу ПК в Інтернет повинен бути налаштований пограничний маршрутизатор з динамічним NAT згідно правила: ім’я пулу – Internet, пул адрес: 209.165.200.5 - 209.165.200.30.

− повинен бути налаштований сервер HTTP для відкриття веб-сайту з відомостями про тему та завдання кваліфікаційної роботи при введені в рядок браузеру http://123.dnipro.ua (http://209.165.200.4);

− повинна бути налаштована віртуальна приватна мережа VPN типу site-to-site з використанням IPsec для трафіку, який проходить між основною мережею та віддаленою мережею організації через Інтернет. [2]

Вимоги під час налаштування мереж VLAN їх маршрутизації:

− в мережі LAN_1 повинно бути створено мережі VLAN з номерами 27, 37, 47, 99 та 100 з відповідними іменами Logistics, Delivery, Store_and_remake, Management та Native;

− на комутаторах мережі LAN_1 повинні бути налаштовані транкові канали та порти доступу і додатково повинні бути вимкнуті фізично не використовувані порти;

− на комутаторах мережі LAN_1 повинні бути налаштовані IP-адреси SVI-інтерфейсів з мережі Management VLAN;

− повинна бути налаштована маршрутизація між мережами VLAN. [2]

Вимоги під час налаштування IP-адресації ПК в мережах VLAN:

− маршрутизатор, який здійснює маршрутизацію між VLAN, повинен бути налаштований в якості сервера DHCP для мереж VLAN;

− пули DHCP повинні мати назву згідно правила: pollvlan№, де № - номер мережі VLAN;

− з пулу DHCP повинні бути виключені перші 10 адрес мережі, вказана IP-адреса серверу DNS та шлюзу за замовчування. [2]

Вимоги до налаштування портів комутатору, підключених до серверів:

− повинен забезпечуватись доступ до порту тільки двом унікальним пристроям;

− MAC-адреса пристрою повинна розпізнаватись динамічно з подальшим додаванням в поточну конфігурацію;

− під час порушення системи безпеки повинно з’являтись повідомлення, а порт залишався включеним. [2]

2.1.3 Вимоги до видів забезпечення

2.1.3.1 Вимоги до математичного забезпечення

Вимоги до математичного забезпечення при розробці Підсистеми 1 повинні включати:

− розрахунок пропускної здатності мережі необхідний для визначення максимальної кількості інформації, яку мережа може передати за період часу, враховуючи її технічних характеристик та параметрів;

− розрахунок пропускної здатності мережі забезпечує оптимальне функціонування мережі з уникненням перевантажень, що може призвести до зниження її продуктивності.

Вимоги до математичного забезпечення при розробці Підсистеми 2 повинні включати:

− розрахунок рівняння прямої кінематики необхідної для визначення кінцевого положення та орієнтації на основі відомих параметрів положення ланок та з'єднань маніпулятору;

− розрахунок рівняння зворотної кінематики необхідної для визначення необхідних параметрів (кутів повороту, положення з'єднань) з метою досягнення заданого положення та орієнтації у просторі.

2.1.3.2 Вимоги до інформаційного забезпечення

Інформаційна система Підсистеми 1 повинна забезпечувати:

− мінімальний час відповіді на запити користувача;

− достатню пропускну здатність обробки запитів, що надходять одночасно, від різних користувачів;

− резервне копіювання інформації в разі виходу з аварійних ситуацій;

− можливість моніторингу в реальному часі.

Інформаційна система Підсистеми 2 повинна забезпечувати:

− обмін інформації маніпулятору з IoT-шлюзом по Wi-Fi;

− прийом параметрів сервоприводів;

− обробку даних від плати керування маніпулятору.

2.1.3.3 Вимоги до лінгвістичного забезпечення

Основною мовою взаємодії користувача з технічним забезпечення повинна бути українська, з можливістю переключитись на англійську.

Програмний інтерфейс повинен бути написаний або українською мовою або англійською. Коментарі до коду повинні бути українськими.

Програмний інтерфейс керування маніпулятором повинен бути розроблений мовою Kotlin з використання фреймворку Jetpack Compose.

2.1.3.4 Вимоги до технічного забезпечення

Вимоги до маршрутизаторів Системи:

− мінімальна кількість портів Gigabit Ethernet: 2;

− мінімальна кількість слотів розширення: 2;

− підтримка протоколів DHCP, NAT, VPN.

Вимоги до комутаторів Системи:

− мінімальна кількість портів Gigabit Ethernet: 16;

− підтримка технології VLAN.

Вимоги до ПК Системи:

− мінімальна кількість ядер процесору та тактова частота: 4 та 2 ГГц;

− мінімальний об’єм оперативної пам’яті: 8 ГБ;

− мінімальний об’єм накопичувача типу SSD: 256 ГБ;

− дискретним відеоадаптером;

− операційною системою Windows 10 або Windows 11.

Вимоги до серверів Системи:

− мінімальна кількість ядер процесору та тактова частота: 4 та 2 ГГц;

− мінімальний об’єм оперативної пам’яті: 16 ГБ;

− мінімальний об’єм накопичувача типу SSD: 256 ГБ;

− мінімальний об’єм накопичувача типу HDD: 1 ТБ;

− дискретним відеоадаптером;

− операційною системою Windows Server 2019 або Ubuntu Server 22.04.

Вимоги до IoT-шлюзу Підсистеми 2:

− підтримка технології Wi-Fi;

− підтримка дистанційного адміністрування;

− мінімальна кількість підключених пристроїв: 12.

Вимоги до технічного забезпечення маніпуляторів Підсистеми 2:

− підтримка Wi-Fi модулю зі стандартами 802.11 b/g/n;

− підтримка плати розширення для керування роботою сервоприводів та Wi-Fi модулем.

2.1.3.5 Вимоги до організаційного забезпечення

ІТ-відділ відповідає за стабільне функціонування та підтримку мережевого обладнання і моніторинг та аналіз трафіку.

Доступ до інформації повинен бути розмежованим за принципом найменшого привілею, забезпечуючи працівникам тільки той рівень доступу, який необхідний для виконання їх обов’язків.

Доступ до інформації структурних підрозділів повинен формуватися на основі групових політик та повинен мати можливість налаштування під час використання мережі.

Забезпечення проведення регулярних навчальних інструктажів для працівників з метою підвищення їх свідомості стосовно потенційних загроз та правил безпечного користування мережевими ресурсами.

Новий персонал має пройти етап навчання завчасно підготовленим інструкційним матеріалами та документацією, як для обладнання з яким він буде взаємодіяти, так і з інформаційним забезпеченням системи, перед тим як приступити до роботи.

2.1.3.6 Вимоги до методичного забезпечення

Повинні бути створені нормативи експлуатації, що включають правила та рекомендації щодо експлуатації комп’ютерної системи та її компонентів. Нормативи експлуатації повинні містити такі розділи: технічне обслуговування, діагностика, ремонт, модернізація, дії при аварійних ситуаціях.

Повинна бути створена документація для програмного інтерфейсу у роботі із маніпуляторами складського приміщення.

Повинні бути надані топологічні схеми мережі, специфікація мережевого обладнання, структурна схема комплексу технічних засобів та таблиця адресації пристроїв.

2.2 Розробка апаратної частини кіберфізичної системи

2.2.1 Розробка загальної структури кіберфізичної системи

На базі топологічної структури (див. рис. 1.12), організаційної архітектури підприємства (див рис. 1.1), кількості підмереж (див. п.2.1.1.1.1) та кількості вузлів в підмережах (див. п.2.1.1.1.1) була розроблена структурна схема технічних засобів комп'ютерної мережі на рисунку 2.1.

Для з'єднання між маршрутизаторами використовуються кабелі типу Serial DCE або крос-кабелі.

Для з'єднання між маршрутизаторами та комутаторами використовуються кабелі прямого типу.

Для з'єднання між комутаторами використовуються кабелі типу крос-кабелі.

Для з'єднання між комутаторами та ПК використовуються кабелі прямого типу.

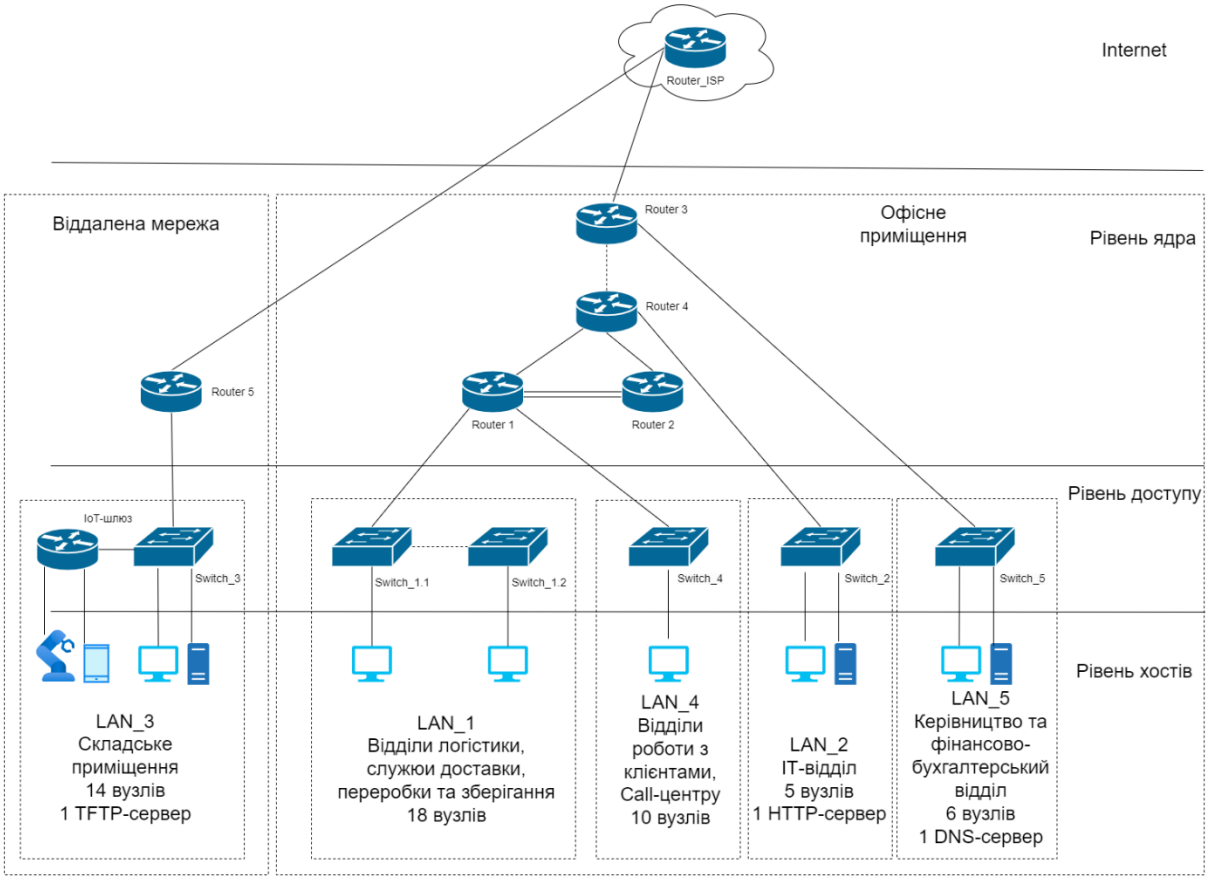

Рисунок 2.1 – Структурна схема технічних засобів комп'ютерної мережі

На рисунку 2.2 розроблена топологічна схема технічних засобів першого поверху офісної будівлі.

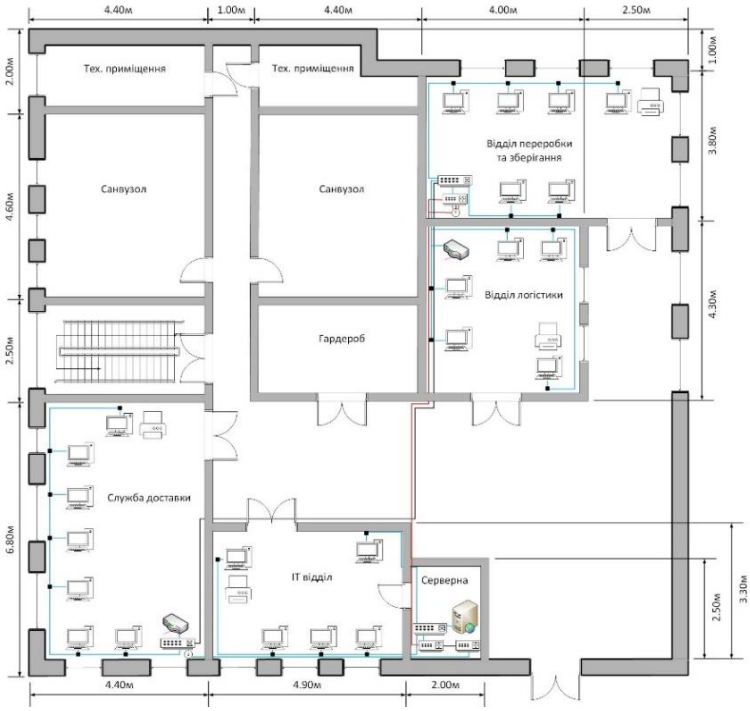

Рисунок 2.2 – Топологічна схема технічних засобів першого поверху офісної будівлі

На рисунку 2.3 розроблена топологічна схема технічних засобів другого поверху офісної будівлі.

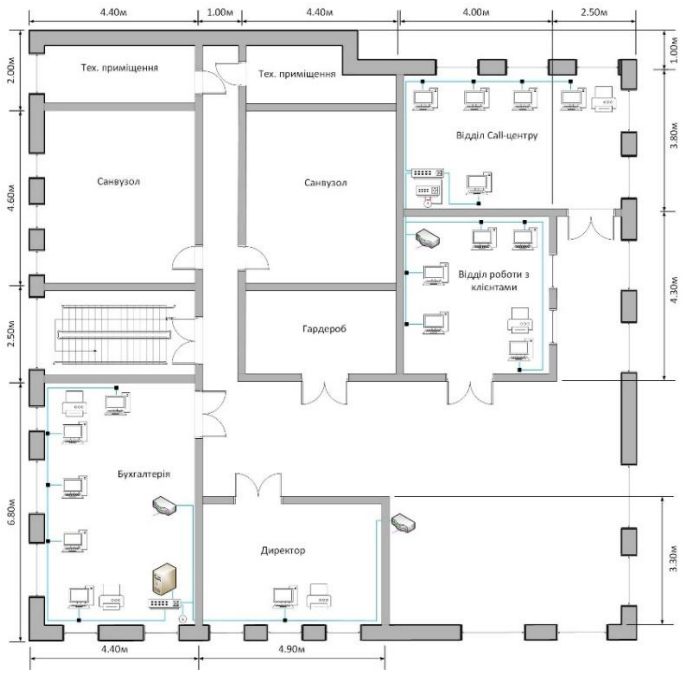

Рисунок 3.3 – Топологічна схема технічних засобів другого поверху офісної будівлі

На рисунку 2.4 розроблена структурна схема технічних засобів Підсистеми 2.

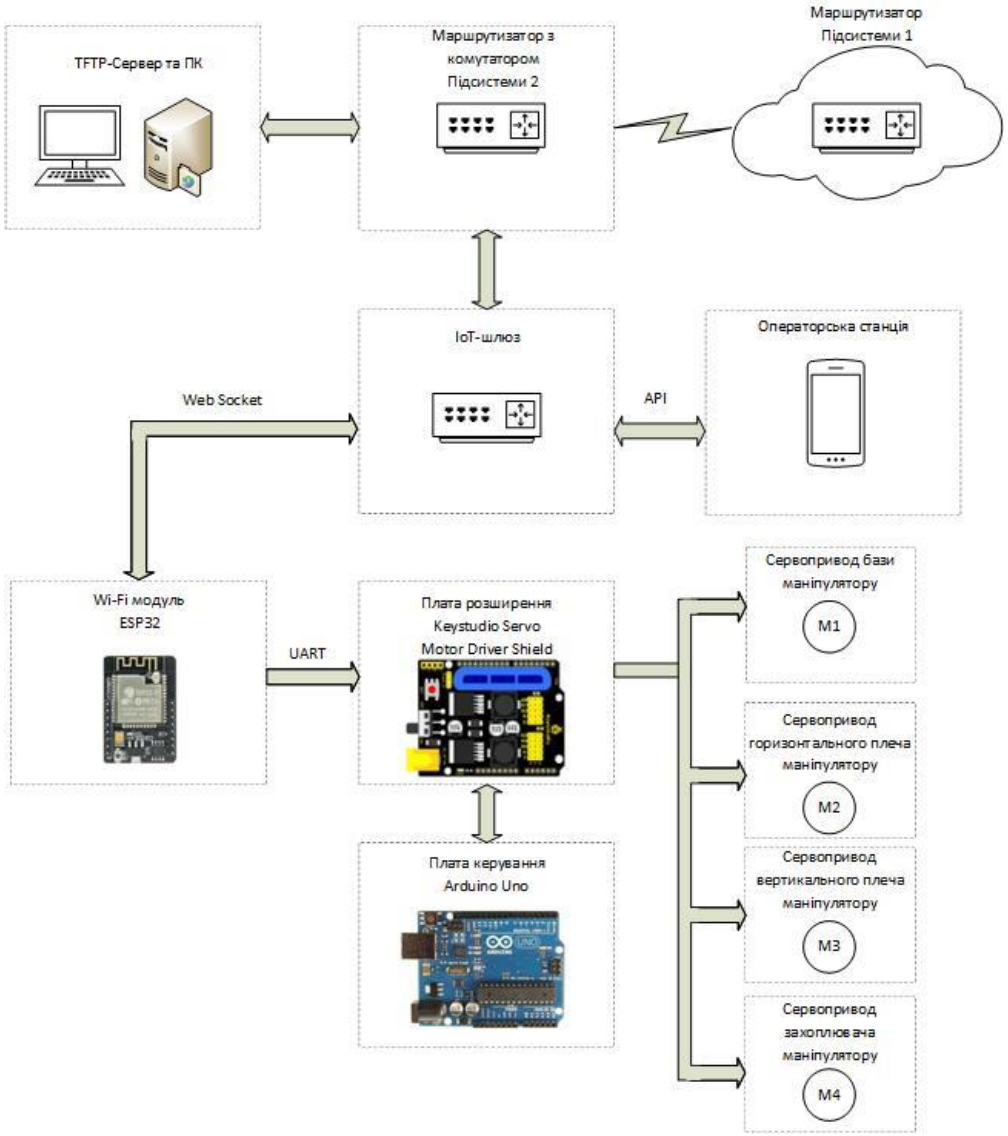

Рисунок 2.4 – Структурна схема технічних засобів Підсистеми 2

2.2.2 Вибір і обґрунтування комплексу технічних кіберфізичної системи

В якості граничного маршрутизатору Підсистеми 1, підключення до Інтернету, обрано маршрутизатор Cisco ISR 4331 (ISR4331/K9) та додатково встановлений модуль NIM-2T (2-портова послідовна інтерфейсна карта WAN). Cisco ISR 4331 є потужним та надійним пристроєм для побудови мережі у середніх та великих підприємствах.

В якості маршрутизаторів мережі обрано маршрутизатори Cisco 2911 (CISCO2911/K9) та додатково встановлені модулі NIM-2T (2-портова

послідовна інтерфейсна карта WAN). Маршрутизатори серії Cisco 2900 мають 4-и порти та слотами розширення. Маршрутизатори цієї серії призначені для використання на малих та середніх підприємствах, що потребують сучасних та технологічні рішень.

Особливість маршрутизаторів 2900 серії можливість переходу портів Gigabyte Ethernet на оптичні (SFP-роз'єми), додаткові слоти розширення EHWIC, що сумісні з модулями попереднього покоління VWIC/ HWIC.

В якості комутаторів мережі обрано Cisco Catalyst 2960-L 16 (WS-C2960L-16PS-LL). Комутатор 2960-L 16 - це Gigabit Ethernet комутатор з фіксованою конфігурацією на другому рівні моделі OSI. Вони призначенні для використання на малих та середніх підприємствах.

В якості серверів мережі обрано 3 Cisco UCS C220 M3 LFF в різних конфігураціях, наведених в таблиці 2.1. Cisco UCS C220 M3 LFF – сервер, що забезпечує достатню продуктивність різноманітних бізнес-додатків, веб-серверів, розподілених баз даних та файлових серверів. Система управління Cisco Unified Computing System (UCS) Manager надає централізоване та вбудоване керування всіма компонентами Cisco UCS (програмними та апаратними).

В якості IoT-шлюзу обрано Aeotec Smart Home Hub (GP-AEOHUBV3EU). Даний IoT-шлюз дає змогу одночасного підключення 100 пристроїв. Підтримує технології Wi-Fi? ZigBee 3.0 та Z-Wave.

В якості ДЖБ обрано KUSTAR UB 6000VA (UB60). KUSTAR UB 6000VA призначений для використання в якості резервного джерела живлення, що забезпечує постійне електроживлення протягом обмеженого періоду часу в разі відключення основного джерела електроенергії.

В якості персональних комп’ютерів працівників обрано готову збірку VINGA ADVANCED A0198. Ця збірка призначена для різноманітних завдань, від повсякденного використання до високопродуктивних завдань. Оснащена потужним процесором і достатньою кількістю оперативної пам'яті для швидкої обробки завдань.

В якості принтера мережі обрано Canon i-SENSYS LBP710CX, цей кольоровий лазерний принтер, забезпечує високу швидкість друку та велику місткість паперу, має невелику площу та простий у використанні і обслуговуванні.

В якості маршрутизатор для бездротової мережі обрано TP-LINK ARCHER-AX12. Використання технології Wi-Fi 6 забезпечує вищу швидкість, меншу затримку та більшу ємність, забезпечуючи більше одночасних з’єднань. Останній протокол безпеки Wi-Fi, WPA3, надає нові можливості для покращення кібербезпеки. Більш надійне шифрування в безпеці паролів Wi-Fi і покращений захист від атак грубої сили.

В якості Ethernet кабелю обрано кабель компанії ATcom, що відноситься до категорії 6 та призначений для використання в локальних мережах. Кабель складається з 8 провідників AWG 23 з кольоровим маркуванням в поліетиленовій ізоляції. Вита пара Atcom має надійну захисну оболонку, добре протистоїть вологі, морозостійка, при перепадах температури зберігає свої фізичні властивості.

В якості плати розширення для керування роботою сервоприводів та Wi-Fi модулю обрано плату розширення Keyestudio TB6612FNG Motor/Servo Drive Shield. Ця плата розширення, оснащена двома модулями скидання напруги LM2596S-5.0V DC-DC, які дозволяють легко керувати кількома сервоприводами.

В якості блока живлення маніпуляторів обрано блок живлення постійної напруги S-60-12. S-60-12 AC-DC імпульсний блок живлення напругою від 9 до 12В да силою струму до 5А. Основні особливості таких пристроїв включають підтримку стабільного рівня вихідної напруги, стабілізацію напруги за допомогою негативного зворотного зв'язку, а також високий коефіцієнт корисної дії (ККД) тощо.

В якості Wi-Fi модулю керування роботою маніпуляторів обрано ESP32-CAM. Плата відрізняється скромним споживанням електроенергії та компактними розмірами. Дана модель налагоджувальної плати має потужний двоядерний 32-бітовий процесор Xtensa із частотою від 80 до 240

МГц, а також модуль від Espressif. Плата підтримує зв'язок через Bluetooth та WiFi, має слот для мікро карти пам'яті SD-карти об'ємом до 4 Гб.

В таблиці 2.1 наведена специфікація технічного кіберфізичної системи.

Таблиця 2.1 – Специфікація обладнання

Позиція	Найменування і технічна характеристика	Тип, марка, позначення документа, опитувального листа	Одиниці виміру	Кількість	Примітки
1	2	3	4	5	6
1.	Маршрутизатор Мережева технологія: 10/100/1000 Base-T Кількість слотів розширення: 6 Кількість портів: 3 Об’єм ОЗП: 4 ГБ Об’єм Flash-пам’яті: 4 ГБ	Cisco ISR 4331 (ISR4331/K9)	шт.	1	Граничний маршрутизатор, додатково NIM-2T модуль
2.	Маршрутизатор Мережева технологія: 10/100/1000 Base-T Кількість слотів розширення: 10 Кількість портів: 3 Об’єм ОЗП: 512 ГБ Об’єм Flash-пам’яті: 256 ГБ	Cisco 2911 (Cisco2911/K9)	шт.	3	Додатково NIM-2T модуль
3.	Комутатор Кількість портів: 16 Об’єм ОЗП: 512 МБ Об’єм Flash-пам’яті: 256 МБ Консольний порт: 1	Cisco Catalyst 2960-L 16 (WS-C2960L-16PS-LL)	шт.	6	Підтримка PoE+ (Pover over Ethernet Plus)
4.	Сервер Кількість процесорів: 2x E5-2650 v2 1.7-2.1 ГГц Об’єм ОЗП: 32 ГБ 1333 ГГц Об’єм SSD накопичувача: 128 ГБ Об’єм HDD накопичувача: 2x 1 ТБ	Cisco UCS C220 M3 LFF	шт.	1	HTTP сервер

Продовження таблиці 2.1

5.	Сервер Кількість процесорів: 2x E5-2650 v2 1.7-2.1 ГГц Об’єм ОЗП: 16 ГБ 1333 ГГц Об’єм SSD накопичувача: 128 ГБ Об’єм HDD накопичувача: 2x 2 ТБ	Cisco UCS C220 M3 LFF	шт.	1	TFTP сервер
6.	Сервер Cisco UCS C220 Кількість процесорів: 2x E5-2650 v2 1.7-2.1 ГГц Об’єм ОЗП: 32 ГБ 1333 ГГц Об’єм SSD накопичувача: 128 ГБ Об’єм HDD накопичувача: 1 ТБ	Cisco UCS C220 M3 LFF	шт.	1	DNS сервер
7.	ДБЖ Вихідна потужність: 6 кВА/5.4 кВт Тип батареї: 12В / 7А*г Діапазон вхідної напруги: 120-276 В	KSTAR UB 6000VA (UB60)	шт.	15	Для серверів, маршрутизаторів та ПК
8.	ПК Процесор: Intel Core i5 10100 ОЗП: 16 ГБ DDR4, 2666 МГц Накопичувач: 320 ГБ SSD	VINGA ADVANCED A0198	шт.	50	Для працівників
9.	Принтер Роздільна здатність: 9600x600 Формат: А4 Технологія друку: лазерна Кількість картриджів: 4	Canon i-SENSYS LBP710CX	шт.	8	Друк з мобільних пристроїв
10.	Маршрутизатор Кількість антен: 4 Стандарт: 802.11ax (Wi-Fi 6) Максимальна швидкість з’єднання: 1201+300 Мбіт/с Кількість діапазонів: 2	TP-LINK ARCHER-AX12	шт.	5	Для гостьової бездротової мережі

Продовження таблиці 2.1

11.	Вита пара Матеріал: мідь Діаметр провідника: 0,51 мм Зовнішня ізоляція: ПВХ Захисний єкран: UTP Довжина: 305 м	ATCOM UTP 305M CAT.6	шт.	1	Категорія 6
12.	Кабельний канал 40х25 мм	Елекор	м.	110	Для витої пари
13.	Розетка комп'ютерна RJ-45	Schneider Electric Asfora	шт.	50	Для підключення ПК
14.	Комутаційна коробка	IEK	шт.	6	Для комутаторів
15.	Блок живлення Напруга живлення: 7-12В Струм живлення: до 5А	S-60-12	шт.	11	Для живлення маніпуляторів
16.	Модуль керування Покриття: до 100м Живлення: 3.6-6 В Антена: вбудована	ESP32-CAM	шт.	11	Модуль керування маніпулятора ми

2.2.3 Розрахунок інтенсивності вихідного трафіку найбільшої локальної мережі підприємства

Найбільшою мережею є підмережа IT відділу, тобто LAN_2. Дані для розрахунку вхідного трафіку:

− кількість вузлів в підмережі: 109;

− середня інтенсивність трафіку: $\mu = 97$ кадрів/с;

− середня довжина повідомлення: $l = 650$ байт;

− затримка передачі пакету: ≤ 6 мс;

− кількість портів комутатору: 16.

Формула для визначення пропускної здатності мережі на рівні доступу (2.1).

$$P_{p.p} = \mu * l * 8 * n, \quad (2.1)$$

де $P_{p.p}$ – пропускна здатність, біт/с;

μ – інтенсивність обслуговування, кадрів/с;

l – середня довжина повідомлення, байт;

n – кількість портів комутатору.

Рішення:

μ = 97 кадрів/с;

l = 650 байт;

n = 16.

$$P_{р.р} = 97 * 650 * 8 * 16 = 8\,070\,400 \approx 8.07 \text{ (Мбіт/с)}$$

Формула для визначення значення інтенсивності виходу (2.2). При розрахунку враховується, що навантаження на комутаторі розраховується через лінію 1000 Мбіт/с.

$$\mu_{вих} = C / (8 * l), \quad (2.2)$$

де C – пропускна здатність лінії, біт/с;

l – середня довжина повідомлення, байт.

Рішення:

C = 1 000 000 000 біт/с;

l = 650 байт.

$$\mu_{вих} = 1\,000\,000\,000 / (8 * 650) = 192\,307 \text{ (пакетів/с)}$$

Формула для визначення максимальної кількості вузлів, яку можна приєднати до комутатору рівня розподілу на основі заданої середньої інтенсивності трафіку (2.3).

$$N = \mu_{вих} / \mu, \quad (2.3)$$

де N – кількість вузлів, яку можна приєднати;

$\mu_{вих}$ – інтенсивність виходу трафіку, пакетів/с;

μ – середня інтенсивність трафіку, пакетів/с.

Рішення:

$\mu_{вих}$ = 192 307 пакетів/с;

μ = 97 пакетів/с.

$$N = 192\,307 / 97 = 1982.55 \text{ (вузлів)}$$

Отже, за наданими значеннями, кількість вузлів, яку можна приєднати до комутатора рівня розподілу, становить приблизно 1982.55. Оскільки кількість вузлів зазвичай є цілим числом, можна округлити результат до

найближчого цілого значення. Тому максимальна кількість вузлів, яку можна приєднати, складатиме 1983.

Формула для визначення загальної інтенсивності трафіку від усіх користувачів (2.4).

$$\lambda = x * \mu, \quad (2.4)$$

де λ – загальна кількість трафіку, пакети/с;

x – коефіцієнт, який представляє кількість користувачів або вузлів в мережі;

μ – середня інтенсивність трафіку, пакети/с;

Рішення:

x = 109 вузлів;

μ = 97 кадрів/с.

$$\lambda = 109 * 97 = 10\,573 \text{ (пакетів/с)}$$

Формула для розрахунку коефіцієнту затримки на рівні розподілу (2.5)

$$\rho = \lambda / \mu_{вих}, \quad (2.5)$$

де ρ – коефіцієнт затримки на рівні розподілу;

λ – загальна кількість трафіку, пакети/с;

$\mu_{вих}$ – інтенсивність виходу (кількість пакетів, що виходить з комутатору за одиницю часу), пакети/с.

Рішення:

λ = 10 573 пакетів/с;

$\mu_{вих}$ = 192 307 пакетів/с;

$$\rho = 10\,573 / 192\,307 \approx 0.055$$

Формула для розрахунку коефіцієнта зайнятості комутатору на рівні розподілу (2.6).

$$r = \rho / (1 - \rho), \quad (2.6)$$

де r – коефіцієнт зайнятості комутатору;

ρ – коефіцієнт затримки на рівні розподілу.

Рішення:

$$r = 0.055 / (1 – 0.055) \approx 0.058$$

Формула для розрахунку середньої затримки кадру (2.7).

$$T = 1 / (\mu_{вих} - \lambda), \quad (2.7)$$

де T – середня затримка кадру, с;

λ – загальна кількість трафіку, пакети/с;

$\mu_{вих}$ – інтенсивність виходу (кількість пакетів, що виходить з комутатору за одиницю часу), пакети/с.

Рішення:

λ = 10 573 пакетів/с;

$\mu_{вих}$ = 192 307 пакетів/с.

$$T = 1 /(192\ 307 – 10\ 573) \approx 0.0000055 \text{ (секунд)} = 5.5 * 10^{-6} \text{(секунд)}$$

Формула для розрахунку середньої довжини черги (2.8).

$$L_{черги} = \rho^2 / (1 - \rho), \quad (2.8)$$

де $L_{черги}$ – середня довжина черги;

ρ – коефіцієнт затримки на рівні розподілу.

Рішення:

ρ = 0.055.

$$L_{черги} = (0.055)^2 / (1 – 0.055) \approx 0.003025 / 0.945 \approx 0.0032$$

Формула для розрахунку середнього часу перебування пакета в черзі (2.9).

$$T_{очік} = L_{черги} / \lambda, \quad (2.9)$$

де $T_{очік}$ – середній час перебування пакета в черзі, с;

$L_{черги}$ – середня довжина черги;

λ – загальна кількість трафіку, пакети/с.

Рішення:

$L_{черги}$ = 0.0032;

λ = 10 573 пакетів/с.

$$T_{очік} = 0.0032 / 10\ 573 = 0.303 \text{ (мс)}$$

Значення $T_{очік}$ менше ніж у наданих вимогах (6 мс), а отже вимоги виконані.

Формула для розрахунку пропускної здатності каналу (2.10).

$$b = \lambda * l * 8, \quad (2.10)$$

де b – пропускна здатність каналу, біт/с;

λ – інтенсивність трафіку, пакетів/с;

l – середня довжина пакету, байт.

Рішення:

λ = 10 573 пакетів/с;

l = 650 байт;

$$b = 10\,573 * 650 * 8 = 54\,979\,600 \text{ біт/с} \approx 54.98 \text{ Мбіт/с}$$

Результат розрахунку пропускної здатності каналу - 54.98 Мбіт/с, співпадає з вихідною пропускною здатністю каналу – 1000 Мбіт/с.

В результаті проведення розрахунків з’ясовано, що розраховані значення показників задовільняють технічні вимоги.

3 РОЗРОБКА КОРПОРАТИВНОЇ МЕРЕЖІ

3.1 Розрахунок схеми адресації корпоративної мережі

Для побудови мережі Підсистеми 1 був використаний блок адрес 172.24.184.0/21. В таблиці 3.1 наведено блок адрес мережі та кількість вузлів в кожній з підмереж.

Таблиця 3.1 – Блок адрес мережі та кількість вузлів в кожній з підмереж

№	Блок адрес	LAN_1	LAN_2	LAN_3	LAN_4	LAN_5
17	172.24.184.0/21	14	109	87	94	40

Необхідно забезпечити функціонування 5 підмереж для 344 користувачів.

Розподіл мережі на підмережі забезпечується з використання методу маски підмережі змінної довжини (VLSM). Цей метод дає змогу ефективно розподіляти адресний простір відповідно до потреб мережі, забезпечуючи економію IP-адрес та краще використання вільних ресурсів.

Для розподілу мережі 172.23.184.0/21 на п’ять підмереж, потрібно визначити маску для кожної з підмереж.

При побудові мережі Підсистеми 1 буде використовуватись протокол IPv4, тому мережу можна розподілити підмережі, що складаються з 1 (маска /32), 2 (маска /31), 4 (маска /30), 8 (маска /29), 16 (маска /28), 32 (маска /27), 64 (маска /26), 128 (маска /25), 256 (маска /24) адрес.

Потрібно врахувати, що дві адреси (адреса мережі та широкомовна адреса) в підмережі не можуть бути використані.

Потрібно врахувати кількість комутаторів в підмережах, для виділення адрес SVI-інтерфейсу.

З урахуванням масштабованості та розвитку, для підмережі LAN_1 з 14 вузлами, обрано 256 (маска /24) адрес, враховуючи наявність віртуальних локальних мережі (VLAN).

Для підмережі LAN_2 з 109 вузлами, обрано 128 (маска /25) адрес.

Для підмережі LAN_3 з 87 вузлами, обрано 128 (маска /25) адрес.

Для підмережі LAN_4 з 94 вузлами, обрано 128 (маска /25) адрес.

Для підмережі LAN_5 з 40 вузлами, обрано 64 (маска /26) адрес.

Адресу мережі переведено в двійковий вид та відрізано частину, в яку входить обрана маска.

Для підтримки найбільшої ділянки мережі LAN_1, потрібно відрізати 8 біт праворуч (з вузлової частини). В результаті отримаємо 2^3=8 підмереж по 2^8-2=254 вузлів у кожній з них з мережевим префіксом /24.

172.24.10111000.|00000000 – підмережа №1.

172.24.10111001.|00000000 – підмережа №2.

172.24.10111010.|00000000 – підмережа №3.

Заповнивши відокремлену частину одиницями отримано кінцевий IP-адрес для кожної з підмережі.

172.24.10111000.|11111111 – кінцевий IP-адрес підмережі №1.

172.24.10111001.|11111111 – кінцевий IP-адрес підмережі №2.

172.24.10111010.|11111111 – кінцевий IP-адрес підмережі №3.

Обрано першу вільну адресу підмережі №1 172.24.184.0/24 з діапазоном вільних адрес 172.24.184.1 – 172.24.184.254 з широкомовною адресою – 172.24.184.255.

Для підтримки наступної ділянки мережі LAN_2, потрібно відрізати 7 біт праворуч з наступної вільної адреси підмережі №2. В результаті отримуємо 2^1=2 підмережі по 2^7-2=126 вузлів у кожній з них з мережевим префіксом /25.

172.24.10111001.0|0000000 – підмережа №2.1.

172.24.10111001.1|0000000 – підмережа №2.2.

Заповнивши відокремлену частину одиницями отримано кінцевий IP-адрес для кожної з підмережі.

172.24.10111001.0|1111111 – кінцевий IP-адрес підмережі №2.1.

172.24.10111001.1|1111111 – кінцевий IP-адрес підмережі №2.2.

Обрано першу вільну адресу підмережі №2.1 172.24.185.0/25 з діапазоном вільних адрес 172.24.185.1 – 172.24.185.126 з широкомовною адресою – 172.24.185.127.

Для підтримки наступної ділянки мережі LAN_4, обрано другу вільну адресу підмережі №2.2 172.24.185.128/25 з діапазоном вільних адрес 172.24.185.129 – 172.24.185.254 з широкомовною адресою – 172.24.185.255.

Для підтримки наступної ділянки мережі LAN_3, потрібно відрізати 7 біт праворуч з наступної вільної адреси підмережі №3. В результаті отримуємо 2^1=2 підмережі по 2^7-2=126 вузлів у кожній з них з мережевим префіксом /25.

172.24.10111010.0|0000000 – підмережа №3.1.

172.24.10111010.1|0000000 – підмережа №3.2.

Заповнивши відокремлену частину одиницями отримано кінцевий IP-адрес для кожної з підмережі.

172.24.10111010.0|1111111 – кінцевий IP-адрес підмережі №3.1.

172.24.10111010.1|1111111 – кінцевий IP-адрес підмережі №3.2.

Обрано першу вільну адресу підмережі №3.1 172.24.186.0/25 з діапазоном вільних адрес 172.24.186.1 – 172.24.186.126 з широкомовною адресою – 172.24.186.127.

Для підтримки наступної ділянки мережі LAN_5, потрібно відрізати 6 біт праворуч з наступної вільної адреси підмережі №3. В результаті отримуємо 2^1=2 підмережі по 2^6-2=62 вузла у кожній з них з мережевим префіксом /26.

172.24.10111010.10|000000 – підмережа №3.2.1.

172.24.10111010.11|000000 – підмережа №3.2.2.

Заповнивши відокремлену частину одиницями отримано кінцевий IP-адрес для кожної з підмережі.

172.24.10111010.10|111111 – кінцевий IP-адрес підмережі №3.2.1.

172.24.10111010.11|111111 – кінцевий IP-адрес підмережі №3.2.2.

Обрано першу вільну адресу підмережі №3.2.1 172.24.186.128/26 з діапазоном вільних адрес 172.24.186.129 – 172.24.186.190 з широкомовною адресою – 172.24.186.191.

Схема адресації підмереж підприємства наведена в таблиці 3.2.

Таблиця 3.2 – Схема адресації підмереж підприємства

Назва мережі	Необхідна кількість вузлів	Виділена кількість вузлів	Адреса підмережі	Префікс	Діапазон допустимих IP-адрес вузлі	Широкомовна адреса
LAN_1	14	254	172.24.184.0	/24	172.24.184.0 - 172.24.184.254	172.24.184.255
LAN_2	109	126	172.24.185.0	/25	172.24.185.1 - 172.24.185.126	172.24.185.127
LAN_3	87	126	172.24.185.128	/25	172.24.184.129- 172.24.184.254	172.24.185.255
LAN_4	94	126	172.24.186.0	/25	172.24.186.1 - 172.24.186.126	172.24.186.127
LAN_5	40	62	172.24.186.128	/26	172.24.186.129- 172.24.186.190	172.24.186.191

Для каналів між маршрутизаторами буде використовуватись блок адрес 10.1.17.0/24. Розподіл на підмережі відбувається за допомогою методу VLSM.

Для підтримки ділянки мережі, що складається з 2 вузлів, необхідна підмережа з маскою /30 (2^2-2=2). Відрізаємо 2 біти праворуч. У результаті отримуємо 2^6=64 підмереж по 2^2-2=2 вузла у кожній. В таблиці 3.3 представлено схему адресації каналів між маршрутизаторами.

Таблиця 3.3 – Схема адресації каналів між маршрутизаторами

Назва підмережі	Адреса підмережі	Префікс	Діапазон допустимих IP-адрес вузлів	Широкомовна адреса
WAN_1	10.1.17.0	/30	10.1.17.1-10.1.17.2	10.1.17.3
WAN_2	10.1.17.4	/30	10.1.17.5-10.1.17.6	10.1.17.7
WAN_3	10.1.17.8	/30	10.1.17.9-10.1.17.10	10.1.17.11
WAN_4	10.1.17.12	/30	10.1.17.13-10.1.17.14	10.1.17.15
WAN_5	10.1.17.16	/30	10.1.17.17-10.1.17.18	10.1.17.19

3.2 Розрахунок схеми адресації пристроїв у корпоративній мережі

У таблиці 3.4 наведена схема адресація всіх маршрутизаторів мережі.

Таблиця 3.4 – Схема адресації маршрутизаторів

Пристрій	Інтерфейс	IP-адрес	Маска мережі	Префікс
Torholskyi_Router1	Gig0/0	172.24.184.1	255.255.255.0	/24
	Gig0/1	172.24.186.1	255.255.255.128	/25

Продовження таблиці 3.4

Torholskyi_Router1	Serial0/1/0	10.1.17.9	255.255.255.252	/30
	Serial0/0/1	10.1.17.13	255.255.255.252	/30
	Serial0/0/0	10.1.17.17	255.255.255.252	/30
Torholskyi_Router2	Serial0/0/0	10.1.17.18	255.255.255.252	/30
	Serial0/0/1	10.1.17.14	255.255.255.252	/30
	Serial0/1/0	10.1.17.5	255.255.255.252	/30
Torholskyi_Router3	Gig0/0/0	10.1.17.1	255.255.255.252	/30
	Gig0/0/1	172.24.186.129	255.255.255.192	/26
	Serial0/1/0	209.165.202.2	255.255.255.240	/28
Torholskyi_Router4	Gig0/0	10.1.17.2	255.255.255.252	/30
	Gig0/1	172.24.185.1	255.255.255.128	/25
	Serial0/0/0	10.1.17.10	255.255.255.252	/30
	Serial0/0/1	10.1.17.6	255.255.255.252	/30
Torholskyi_Router5	Gig0/0	172.24.185.129	255.255.255.128	/25
	Serial0/0/0	64.100.13.1	255.255.255.252	/30
Router_ISP	Gig0/0	209.165.201.1	255.255.255.240	/28
	Serial0/0/0	209.165.202.1	255.255.255.240	/28
	Serial0/0/1	64.100.13.2	255.255.255.252	/30

Адреси SVI-інтерфейсів комутаторів наведено в таблиці 3.5

Таблиця 3.5 – Схема адресації SVI-інтерфейсів комутаторів

Підмережа	Пристрій	IP-адрес	Маска мережі	Адреса шлюзу
LAN_1	Torholskyi_Switch1.1	172.24.184.195	255.255.255.240	172.24.184.193
	Torholskyi_Switch1.2	172.24.184.194	255.255.255.240	172.24.184.193
LAN_2	Torholskyi_Switch2	172.24.185.2	255.255.255.128	172.24.185.1
LAN_3	Torholskyi_Switch3	172.24.185.130	255.255.255.128	172.24.185.129
LAN_4	Torholskyi_Switch4	172.24.186.2	255.255.255.128	172.24.186.1
LAN_5	Torholskyi_Switch5	172.24.186.130	255.255.255.192	172.24.186.129

3.3 Розробка топологічної схеми корпоративної мережі

На рисунку 3.1 розроблена логічна схема корпоративної мережі. Логічна схема складається з мережі офісного приміщення та віддаленої мережі складського приміщення, де розміщенні маніпулятори, а також мережа провайдеру послуг Інтернету.

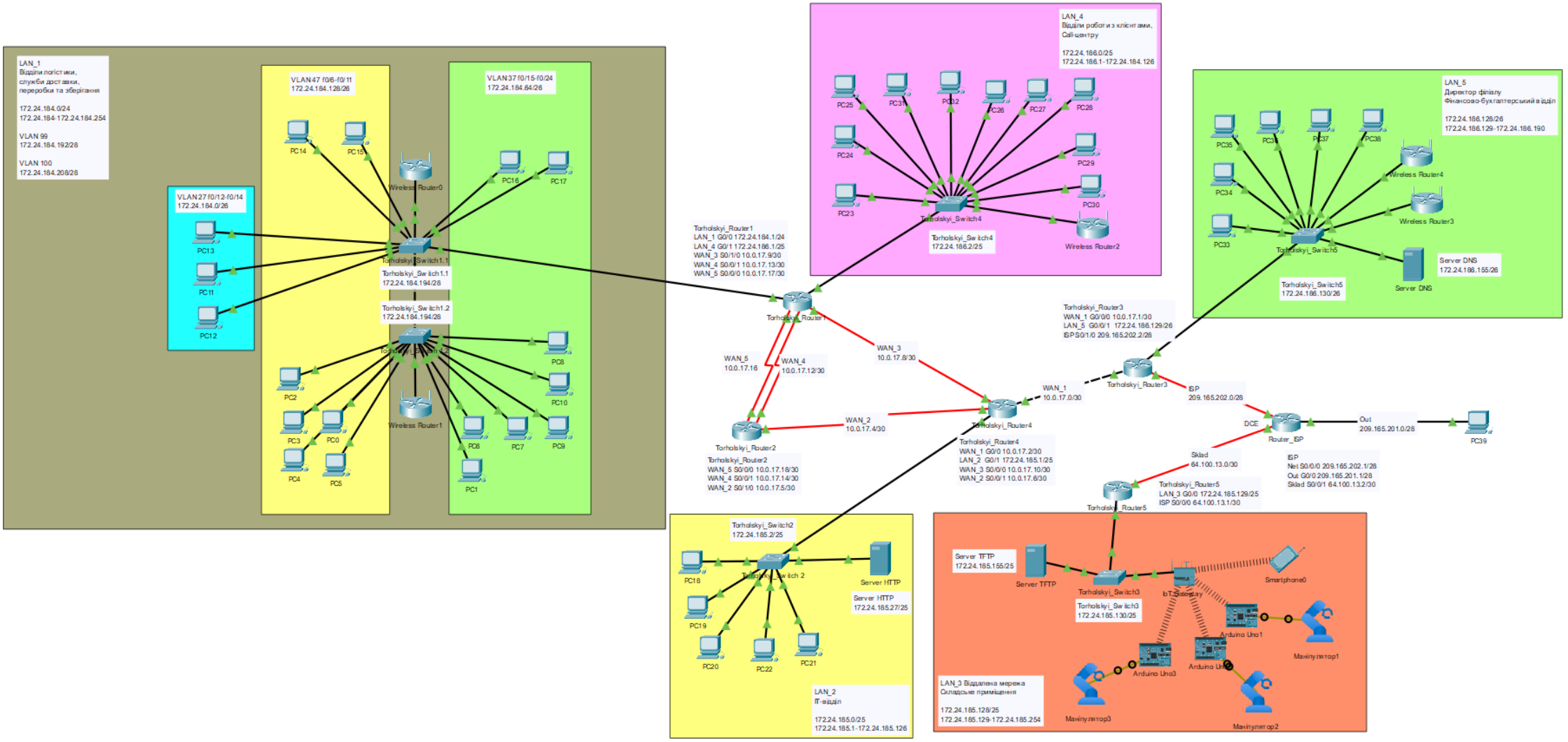

Рисунок 3.1 – Логічна схема корпоративної мережі логістичного підприємства

3.4 Налаштування моделі комп’ютерної мережі

3.4.1 Базове налаштування конфігурації пристроїв

Згідно технічних вимог було проведено базове налаштування мережевого обладнання Системи.

Приклад базового налаштування маршрутизатору Torholskyi_Router1 (див. рис. 3.2):

```
no ip domain-lookup     // заборона перетворень доменних імен у випадку помилкового введення
                        // в командний рядок не перетворюваних слів
hostname Torholskyi_Router1 // призначення унікального імені пристрою
line console 0          // вхід до конфігурації лінії консолі
password cisco          // призначення паролю лінії консолі
login                   // вимкнення анонімного доступу
line vty 0 15           // вхід до конфігурації ліній VTY
password cisco          // призначення паролю ліній VTY
login                   // вимкнення анонімного доступу
enable secret class     // призначення шифрованого паролю для привілейованого режиму
service password-encryption     // шифрування паролів, які зберігаються у відкритому вигляді
banner motd #Torholskyi_Router1. Staff access only!#    // налаштування банеру MOTD
line vty 0 15           // вхід до конфігурації ліній VTY
transport input ssh     // призначення використання протоколу SSH
login local             // налаштування локальної аутентифікації
username 12321sk1_Torholskyi password admincisco        // створення користувача та паролю
ip domain-name Torholskyi_Router1                       // налаштування доменного імені
crypto key generate rsa         // створення ключа шифрування
1024                    // налаштування довжини ключа шифрування
interface s0/0/0        // вибір DCE-інтерфейсу
bandwidth 128           // налаштування пропускної спроможності
ip ospf cost 7500       // налаштування метрики
clock rate 128000       // налаштування тактової частоти
interface s0/0/1
bandwidth 128
ip ospf cost 7500
clock rate 128000
interface s0/1/0
bandwidth 128
ip ospf cost 7500
clock rate 128000
```

Рисунок 3.2 – Базове налаштування маршрутизатору Torholskyi_Router1

3.4.2 Налаштування маршрутизаторів

Згідно технічних вимог для маршрутизації пристроїв використовується протокол динамічної маршрутизації OSPF (Open Short Path First).

Перевагою протоколу:

− підтримка мережевої маски змінної довжини (VLSM);

− використання алгоритму Дейсктри для визначення найкоротшого шляху до кожного з вузлів мережі;

− автоматичне оновлення маршрутів.

Приклад налаштування протоколу OSPF маршрутизатору Torholskyi_Router4 (див. рис. 3.3).

```
router ospf 17                          // ввімкнення протоколу динамічної маршрутизації
network 172.24.185.0 0.0.0.127 area 0   // оголошення необхідних для маршрутизації мереж
network 10.1.17.0 0.0.0.3 area 0
network 10.1.17.4 0.0.0.3 area 0
network 10.1.17.8 0.0.0.3 area 0
passive-interface default               // вимкнення поширення оновлень на всіх портах
no passive-interface s0/0/0             // ввімкнення поширення оновлень на портах, підключених з іншими маршрутизаторами
no passive-interface s0/0/1
no passive-interface g0/0
```

Рисунок 3.3 – Налаштування OSPF маршрутизатору Torholskyi_Router4

На граничному маршрутизаторів Torholskyi_Router3 оголошено маршрут за замовчування до маршрутизатору ISP (Internet Service Provider) та оголошено його поширення (див. рис. 3.4).

```
ip route 0.0.0.0 0.0.0.0 209.165.202.1                  // оголошення маршруту за замовчуванням
ip route 209.165.201.0 255.255.255.240 209.165.202.1    // оголошення статичного маршруту до мережі ISP
router ospf 17                                          // ввімкнення протоколу динамічної маршрутизації
default-information originate                           // ввімкнення поширення маршруту за замовчуванням
```

Рисунок 3.4 – Налаштування маршруту за замовчуванням маршрутизатору Torholskyi_Router3

Для налаштування адресації пристроїв в мережі на маршрутизаторах використовується протокол динамічної конфігурації DHCP (Dynamic Host Configuration Protocol). Використання цього протоколу спрощує налаштування та адміністрування пристроїв.

Приклад налаштування протоколу динамічної конфігурації DHCP маршрутизатору Torholskyi_Router1 (див. рис. 3.5).

```
ip dhcp pool LAN_4                      // створення пулу DHCP для підмережі LAN_4
network 172.24.186.0 255.255.255.128    // оголошення IP-адреси для підмережі LAN_4
default-router 172.24.186.1             // оголошення IP-адрес шлюзу за замовчуванням для підмережі LAN_4
dns-server 172.24.186.155               // оголошення IP-адрес DNS серверу для підмережі LAN_4
ip dhcp excluded-address 172.24.186.1 172.24.186.10     // виключення вказаних адрес з пулу DHCP
```

Рисунок 3.5 – Налаштування протоколу DHCP маршрутизатору Torholskyi_Router1

3.4.3 Налаштування роботи Інтернет

Для доступу Системи до мережі Інтернет використовується технологія NAT (Network Address Translation). Технологія NAT перетворює один простір IP-адрес в інший шляхом зміни інформації про мережеву адресу в IP-заголовку пакетів, коли вони передаються через пристрій маршрутизації трафіку. Ця технологія використовувався для обходу необхідності призначати нову адресу кожному вузлу під час переміщення

мережі або заміни постачальника послуг Інтернету, але не міг маршрутизувати адресний простір мережі. [3]

Приклад налаштування технології NAT граничного маршрутизатору Torholskyi_Router3 (див. рис. 3.6).

```
ip access-list extended NAT            // створення ACL-списку для технології NAT
deny ip 172.24.184.0 0.0.0.255 172.24.185.128 0.0.0.127  // заборона надходження пакетів з віддаленої мережі до мережі офісу
deny ip 172.24.185.0 0.0.0.127 172.24.185.128 0.0.0.127
deny ip 172.24.186.0 0.0.0.127 172.24.185.128 0.0.0.127
deny ip 172.24.186.128 0.0.0.63 172.24.185.128 0.0.0.127
deny ip 10.1.17.0 0.0.0.255 172.24.185.128 0.0.0.127
permit ip 172.24.184.0 0.0.0.255 any    // дозвіл надходження будь-яких пакетів з мережі офісу
permit ip 172.24.185.0 0.0.0.127 any
permit ip 172.24.186.0 0.0.0.127 any
permit ip 172.24.186.128 0.0.0.63 any
permit ip 10.1.17.0 0.0.0.255 any
ip nat pool Internet 209.165.200.5 209.165.200.30 netmask 255.255.255.224      // створення пулу адрес NAT
ip nat inside source list NAT pool Internet                  // прив'язка списку доступу з пулом адресів Internet
ip nat inside source static 172.24.185.27 209.165.200.4      // призначення IP-адреси NAT для серверу HTTP
ip nat inside source static 172.24.186.155 209.165.200.3     // призначення IP-адреси NAT для серверу DNS
interface g0/0/0                                             // вибір внутрішнього інтерфейсу
ip nat inside                                                // налаштування інтерфейсу як внутрішній інтерфейс NAT
interface g0/0/1
ip nat inside
interface s0/1/0                                             // вибір зовнішнього інтерфейсу
ip nat outside                                               // налаштування інтерфейсу як зовнішній інтерфейс NAT
```

Рисунок 3.6 – Налаштування технології NAT маршрутизатору Torholskyi_Router3

3.5 Захист інформації в комп'ютерній Системі

3.5.1 Налаштування маршрутизаторів на підтримку служби AAA

Служба AAA (Authentication Authorization Accounting) – це механізм авторизації, аутентифікації та обліку, який використовується для керування доступом до мережевих ресурсів. Ця служба дає змогу контролю користувачів, що намагаються отримати доступ до мережевих пристроїв.

Приклад налаштування служби AAA маршрутизатору Torholskyi_Router1 (див. рис. 3.7).

```
aaa new-model                          // ввімкнення служби AAA
aaa authentication login default group radius local     // налаштування аутентифікації для консольного доступу
                                                        // до мережевого пристрою з використанням серверу Radius
radius server serverRadius             // налаштування серверу Radius
address ipv4 172.24.186.155 auth-port 1645              // оголошення IP-адреси та порту підключення серверу Radius
key radius123                          // оголошення ключа аутентифікації
line console 0                         // вхід до конфігурації лінії консолі
login authentication default           // встановлення методу аутентифікації для доступу до консольного порту
line vty 0 15                          // вхід до конфігурації віртуальних ліній
login authentication default           // встановлення методу аутентифікації для доступу до віртуальних ліній
```

Рисунок 3.7 – Налаштування служби AAA маршрутизатору Torholskyi_Router1

3.5.2 Налаштування віртуальних локальних мереж VLAN

Для розподілу фізичної локальної мережі на віртуальні локальні мережі використовується технологія VLAN. Перевагою використання

технології VLAN є використання одного фізичного інфраструктурного шару для розподілення трафіку між різними група працівників, що в свою чергу зменшує потреби в розміщенні додаткового комутаційного обладнання. [4]

Підмережа LAN_1 була розподілена на три підмережі VLAN. Номери та назви мереж VLAN представлено в таблиці 3.6.

Таблиця 3.6 – Мережі VLAN

Номер VLAN	Ім’я VLAN	Примітка
1	Default	Не використовується
27	Logistics	Для відділу логістики
37	Delivery	Для відділу доставки
47	Store_and_remake	Для відділу переробки та зберіганн
99	Managment	Для управління пристроями
100	Native	Власна мережі

Схема адресації мереж VLAN представлено в таблиці 3.7.

Таблиця 3.7 – Схема адресації мереж VLAN

Назва	Розмір	Адреса	Маска мережі	Діапазон адрес	Широкомовна адреса
Logistics	62	172.24.184.0	255.255.255.192	172.24.184.1–172.24.184.62	172.24.184.63
Delivery	62	172.24.184.64	255.255.255.192	172.24.184.65–172.24.184.126	172.24.184.127
Store_and_remake	62	172.24.184.128	255.255.255.192	172.24.184.129–172.24.184.190	172.24.184.191
Managment	14	172.24.184.192	255.255.255.240	172.24.184.193–172.24.184.206	172.24.184.207
Native	14	172.24.184.208	255.255.255.240	172.24.184.209–172.24.184.222	172.24.184.223

Розподіл портів для окремих мереж VLAN представлено в таблиці 3.8.

Таблиця 3.8 – Розподіл портів для окремих мереж VLAN

Назва	VLAN	Порти
Logistics	27	f0/12-f0/14
Delivery	37	f0/15-f0/24
Store_and_remake	47	f0/6-f0/11

Адресація пристроїв в підмережі LAN_1 представлена в таблиці 3.9.

Таблиця 3.9 – Адресація пристроїв в підмережі LAN_1

Пристрій	Інтерфейс	Адреса	Маска мережі	Шлюз	VLAN
Switch 1.1	SVI	172.24.184.194	255.255.255.240	172.24.184.193	99
Switch 1.2	SVI	172.24.184.195	255.255.255.240	172.24.184.193	99
Router1	G0/0.27	172.24.184.1	255.255.255.192	-	27
	G0/0.37	172.24.184.65	255.255.255.192	-	37
	G0/0.47	172.24.184.129	255.255.255.192	-	47
	G0/0.99	172.24.184.193	255.255.255.240	-	99

Приклад налаштування технології VLAN маршрутизатору Torholskyi_Switch1.1 (див. рис. 3.8).

```
vlan 27                 // створення мережі VLAN
name Logistics          // налаштування імені мережі VLAN
vlan 37
name Delivery
vlan 47
name Store_and_remake
vlan 99
name Managment
vlan 100
name Native
interface range fa0/6-11        // вибір портів
switchport mode access          // налаштування режиму роботи портів
switchport access vlan 47       // налаштування портів для роботи з VLAN 47
interface range fa0/12-14
switchport mode access
switchport access vlan 27
interface range fa0/15-24
switchport mode access
switchport access vlan 37
interface range fa0/1-5
switchport mode trunk           // налаштування транкового каналу
switchport trunk native vlan 100        // налаштування власної мережі на транковому каналі
switchport trunk allowed vlan 47,27,37,99-100   // налаштування переліку мереж VLAN доступних на транковому каналі
interface vlan 99               // вибір VLAN
ip address 172.24.184.194 255.255.255.240       // оголошення IP-адреси
ip default-gateway 172.24.184.193       // оголошення шлюзу за замовчуванням
```

Рисунок 3.8 – Налаштування технології VLAN маршрутизатору Torholskyi_Switch1.1

Приклад налаштування підінтерфейсів для мереж VLAN маршрутизатору Torholskyi_Router1 (див. рис. 3.9).

```
interface g0/0.47               // вибір підінтерфейсу
encapsulation dot1Q 47          // встановлення мітки мережі VLAN для порту
ip address 172.24.184.129 255.255.255.192       // оголошення IP-адреси підінтерфейсу
interface g0/0.27
encapsulation dot1Q 27
ip address 172.24.184.1 255.255.255.192
interface g0/0.37
encapsulation dot1Q 37
ip address 172.24.184.65 255.255.255.192
interface g0/0.99
encapsulation dot1Q 99
ip address 172.24.184.193 255.255.255.240
```

Рисунок 3.9 – Налаштування підінтерфейсів для мереж VLAN маршрутизатору Torholskyi_Router1

Приклад налаштування протоколу динамічної конфігурації в мережах VLAN маршрутизатору Torholskyi_Router1 (див. рис. 3.10).

```
ip dhcp excluded-address 172.24.184.1 172.24.184.11      // виключення вказаних IP-адрес
ip dhcp excluded-address 172.24.184.193 172.24.184.195
ip dhcp excluded-address 172.24.184.65 172.24.184.71
ip dhcp excluded-address 172.24.184.129 172.24.184.135
ip dhcp pool LAN_1_poolvlan27                           // створення пулу DHCP для VLAN Logistics
network 172.24.184.0 255.255.255.192                    // оголошення IP-адреси мережі для VLAN Logistics
default-router 172.24.184.1                             // оголошення IP-адреси шлюзу за замовчуванням для VLAN Logistics
dns-server 172.24.186.155                               // оголошення IP-адреси серверу DNS для VLAN Logistics
ip dhcp pool LAN_1_poolvlan37
network 172.24.184.64 255.255.255.192
default-router 172.24.184.65
dns-server 172.24.186.155
ip dhcp pool LAN_1_poolvlan47
network 172.24.184.128 255.255.255.192
default-router 172.24.184.129
dns-server 172.24.186.155
```

Рисунок 3.10 – Налаштування протоколу динамічної конфігурації в мережах VLAN маршрутизатору Torholskyi_Router1

На порті комутатору Torholskyi_Switch3, який підключений до серверу TFTP, налаштовано заходи безпеки (див. рис. 3.11).

```
interface f0/3                  // вибір порту підключення серверу TFTP
switchport mode access          // налаштування режиму порту
switchport port-security        // ввімкнення захисту порту
switchport port-security maximum 2      // оголошення кількості унікальних пристроїв, яким наданий доступ
switchport port-security mac-address sticky     // налаштування автоматичного розпізнавання MAC-адресу
                                                // з додаванням його в поточну конфігурацію
switchport port-security violation restrict     // налаштування дій комутатору на випадок порушення безпеки
```

Рисунок 3.11 – Налаштування заходів безпеки комутатору Torholskyi_Switch3

3.5.3 Налаштування віртуальної приватної мережі VPN

Згідно технічних вимог було проведено налаштування технології VPN з використанням набору протоколів IPsec. Використання VPN (Virtual Privat Network) дозволяє безпечно обмінюватись даними між різними підрозділами підприємства, в тому числі і віддаленими. [5]

Приклад налаштування технології VPN маршрутизатору Torholskyi_Router5 (див. рис. 3.12).

```
license boot module c2900 technology-package securityk9         // активація модулю sekurityk9
ip access-list extended VPN                                     // створення ACL-списку для VPN
permit ip 172.24.185.128 0.0.0.127 172.24.184.0 0.0.0.255       // дозвіл на проходження пакетів з мережі офісу до віддаленої мережі
permit ip 172.24.185.128 0.0.0.127 172.24.185.0 0.0.0.127
permit ip 172.24.185.128 0.0.0.127 172.24.186.0 0.0.0.127
permit ip 172.24.185.128 0.0.0.127 172.24.186.128 0.0.0.63
permit ip 172.24.185.128 0.0.0.127 10.1.17.0 0.0.0.255
crypto isakmp policy 10                                         // створення криптографічної політики
encryption aes 256                                              // вибір алгоритму шифрування з довжиною ключа 256 біт
hash sha                                                        // вибір алгоритму хеш-суми
authentication pre-share                                        // вибір методу аутентифікації попередньо обмінюваних ключів
group 2                                                         // створення групи для обміну ключами
crypto isakmp key cisco address 209.165.202.2                   // налаштування ключа ISAKMP з адресою вихідного
                                                                // інтерфейсу граничного маршрутизатору мережі офісу
crypto ipsec transform-set VPN-SET esp-aes esp-sha-hmac         // створення набору перетворень
crypto map MAP 10 ipsec-isakmp                                  // створення криптографічного зіставлення
set peer 209.165.202.2                                          // оголошення IP-адреси піру, для VPN з'єднання
set transform-set VPN-SET                                       // вибір набору перетворень для зіставлення
match address VPN                                               // зв'язування зі ACL-списком VPN
interface s0/0/0                                                // вибір інтерфейсу
crypto map MAP                                                  // зв'язування криптографічного зіставлення
                                                                // MAP з вихідним інтерфейсом
```

Рисунок 3.12 – Налаштування технології VPN маршрутизатору Torholskyi_Router5

3.6 Перевірка комп’ютерної Системи підприємства

Перевірка базового налаштування пристроїв на прикладі маршрутизатору Torholskyi_Router1 за допомогою команди show running-config. Перевірка назви пристрою (див. рис. 3.13), імені і паролю користувача (див. рис. 3.14), імені домену (див. рис. 3.15), паролю до привілейованого режиму (див. рис. 3.16), банеру MOTD (див. рис. 3.17) паролю до консолі (див. рис. 3.18), паролю до віртуальних ліній vty та використання на них протоколу SSH (див. рис. 3.19) та шифрування паролів, які знаходяться у відкритому вигляді (див. рис 3.20).

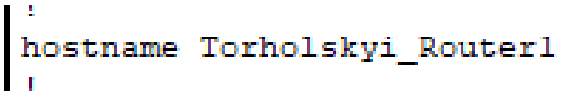

Рисунок 3.13 – Назва пристрою

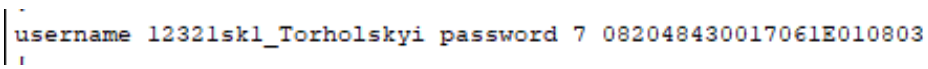

Рисунок 3.14 – Ім’я та паролю користувача

Рисунок 3.15 – Ім’я домену

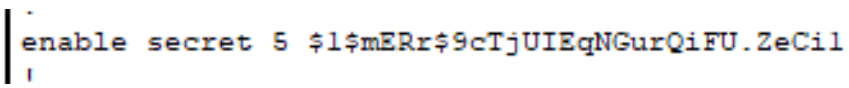

Рисунок 3.16 – Пароль привілейованого режиму

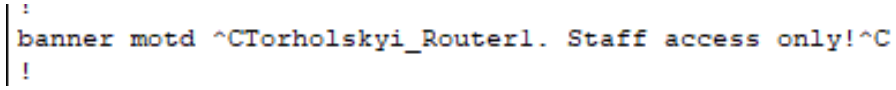

Рисунок 3.17 – Банер MOTD

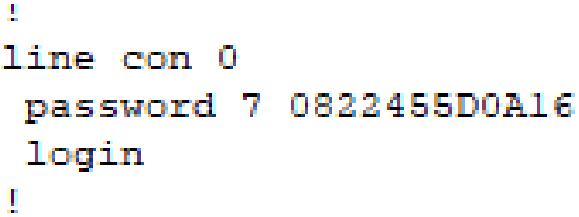

Рисунок 3.18 – Пароль консолі

```
line vty 0 4
 password 7 0822455D0A1
 login local
 transport input ssh
line vty 5 15
 password 7 0822455D0A1
 login local
 transport input ssh
```

Рисунок 3.19 – Пароль віртуальних ліній vty та протокол SSH

```
service password-encryption
```

Рисунок 3.20 – Шифрування відкритих паролів

Перевірка налаштування IP-адреси на послідовному інтерфейсі, його тактової частоти, метрики та пропускної здатності (див. рис. 3.21).

```
interface Serial0/1/0
 bandwidth 128
 ip address 10.1.17.9 255.255.255.252
 ip ospf cost 7500
 clock rate 128000
```

Рисунок 3.21 – Перевірка Serial інтерфейсу

Перевірка налаштування маршрутизації з використання протоколу OSPF на прикладі маршрутизатору Torholskyi_Router4. Перевірка здійснюється за допомогою команди show ip protocols (див. рис. 3.22).

```
Torholskyi_Router4#show ip protocols

Routing Protocol is "ospf 17"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 172.24.185.1
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    172.24.185.0 0.0.0.127 area 0
    10.1.17.0 0.0.0.3 area 0
    10.1.17.4 0.0.0.3 area 0
    10.1.17.8 0.0.0.3 area 0
  Passive Interface(s):
    Vlan1
    GigabitEthernet0/1
    GigabitEthernet0/2
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.1.17.18           110      00:03:15
    172.24.185.1         110      00:02:38
    172.24.186.1         110      00:03:16
    209.165.202.2        110      00:02:42
  Distance: (default is 110)
```

Рисунок 3.22 – Перевірка протоколу OSPF

Перевірка маршрутизації з використанням протоколу OSPF на прикладі персонального комп'ютера з LAN_2 – PC18 та персонального комп'ютера з LAN_4 – PC23 (див. рис. 3.23).

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Failed	PC18	PC23	ICMP		0.000	N	0	(edit)	
	Successful	PC18	PC23	ICMP		0.000	N	1	(edit)	

Рисунок 3.23 – Перевірка маршрутизації

Перевірка налаштування статичних маршрутів на прикладі маршрутизатору Torholskyi_Router3. Перевірка здійснюється за допомогою команди show ip route static (див. рис. 3.24).

```
Torholskyi_Router3#show ip route static
     209.165.201.0/28 is subnetted, 1 subnets
S       209.165.201.0 [1/0] via 209.165.202.1
     209.165.202.0/24 is variably subnetted, 3 subnets, 3 masks
S       209.165.202.0/30 is directly connected, Serial0/1/0
S*   0.0.0.0/0 [1/0] via 209.165.202.1
```

Рисунок 3.24 – Перевірка статичних маршрутів

Перевірка служби AAA на прикладі Torholskyi_Router3 (див. рис. 3.25).

```
Torholskyi_Router3. Staff access only!

User Access Verification

Username: Torholskyi_admin
Password:
Torholskyi_Router3>
```

Рисунок 3.25 – Перевірка служби AAA

Перевірка протоколу DHCP на прикладі персонального комп'ютера PC23 (див. рис. 3.26).

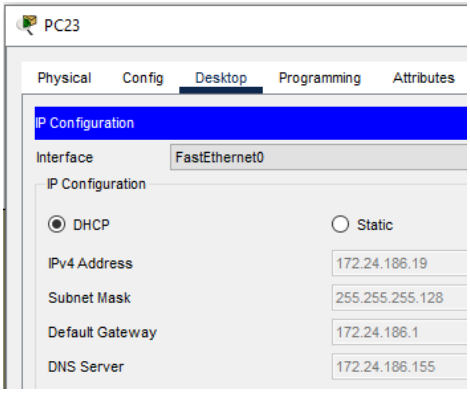

Рисунок 3.26 – Протокол DHCP

Перевірка налаштувань серверу Radius з клієнтами та користувачами (див. рис. 3.27).

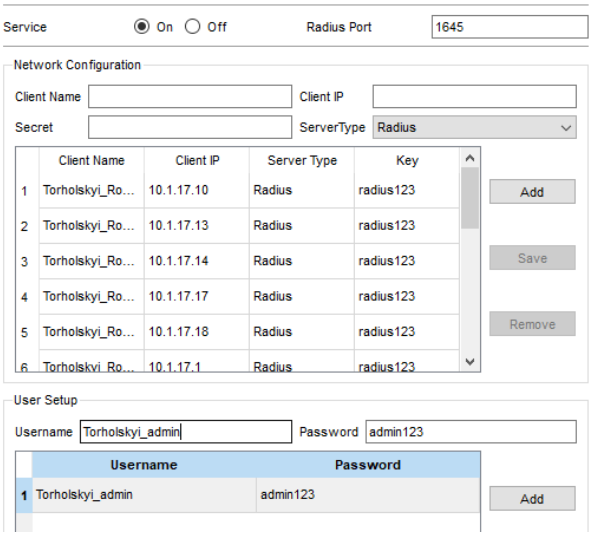

Рисунок 3.27 – Налаштування серверу Radius

Перевірка статичних IP-адрес серверу HTTP (див. рис. 3.28), серверу DNS (див. рис. 3.29) та серверу TFTP (див. рис. 3.30).

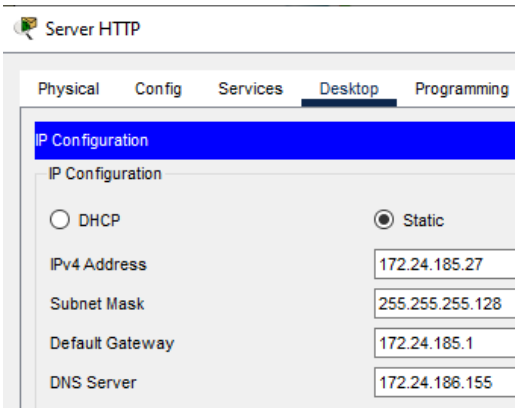

Рисунок 3.28 – IP-адреса серверу HTTP

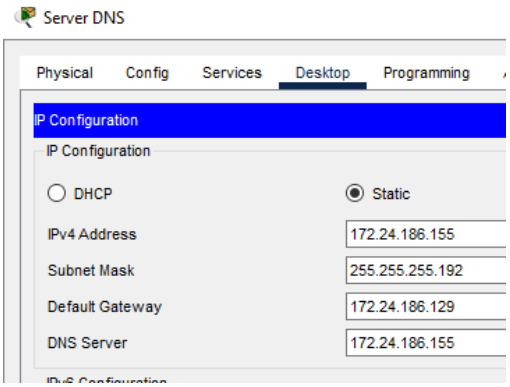

Рисунок 3.29 – IP-адреса серверу DNS

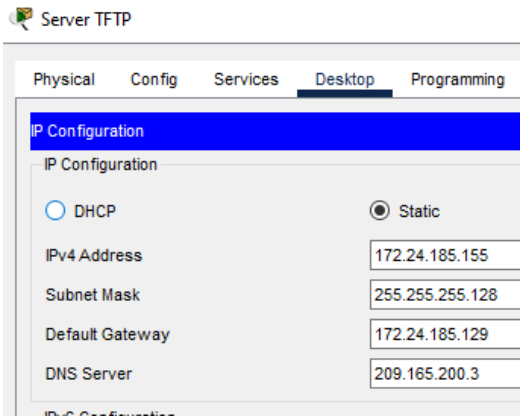

Рисунок 3.30 – IP-адреса серверу TFTP

Перевірка IP-адрес на SVI-інтерфейсах на прикладі комутатору Torholskyi_Switch2. Перевірка здійснюється за допомогою команди show interface vlan1 (див. рис. 3.31).

```
Torholskyi_Switch2#show interfaces vlan1
Vlan1 is up, line protocol is up
  Hardware is CPU Interface, address is 0001.6
  Internet address is 172.24.185.2/25
```

Рисунок 3.31 – IP-адреса на SVI-інтерфейсі

Перевірка налаштувань безпеки порту комутатору з’єднаного з сервером TFTP на прикладі Torholskyi_Switch3. Перевірка здійснюється за допомогою команди show port-security (див. рис. 3.32).

```
Torholskyi_Switch3#show port-security
Secure Port MaxSecureAddr CurrentAddr SecurityViolation Security Action
               (Count)       (Count)        (Count)
--------------------------------------------------------------------------
        Fa0/3        2           1                0          Restrict
--------------------------------------------------------------------------
```

Рисунок 3.32 – Перевірка безпеки порту

Перевірка налаштованих мереж VLAN та портів, що належать кожній мережі VLAN на прикладі комутатору Torholskyi_Switch1.1. Перевірка здійснюється за допомогою команди show vlan (див. рис. 3.33).

```
Torholskyi_Switch1.1#show vlan

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/3, Fa0/4, Fa0/5, Gig0/1
                                                Gig0/2
27   Logistics                        active    Fa0/12, Fa0/13, Fa0/14
37   Delivery                         active    Fa0/15, Fa0/16, Fa0/17, Fa0/18
                                                Fa0/19, Fa0/20, Fa0/21, Fa0/22
                                                Fa0/23, Fa0/24
47   Store_and_remake                 active    Fa0/6, Fa0/7, Fa0/8, Fa0/9
                                                Fa0/10, Fa0/11
99   Managment                        active
100  Native                           active
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active
```

Рисунок 3.33 – Мережі VLAN та їх порти

Перевірка транкового каналу VLAN на прикладі комутатору Torholskyi_Switch1.2. Перевірка здійснюється за допомогою команди show interface trunk (див. рис. 3.34).

```
Torholskyi_Switch1.2#show interface trunk
Port        Mode         Encapsulation  Status        Native vlan
Fa0/2       on           802.1q         trunking      100

Port        Vlans allowed on trunk
Fa0/2       27,37,47,99-100

Port        Vlans allowed and active in management domain
Fa0/2       27,37,47,99,100

Port        Vlans in spanning tree forwarding state and not pruned
Fa0/2       27,37,47,99,100
```

Рисунок 3.34 – Транковий канал

Перевірка протоколу DHCP в мережі VLAN 27 на прикладі персонального комп’ютера PC13 (див. рис. 3.35).

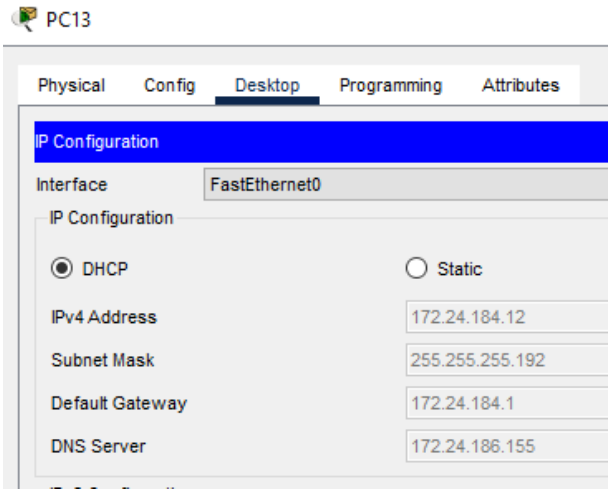

Рисунок 3.35 – Протокол DHCP в мережі VLAN 27

Перевірка маршрутизації між мережами VLAN на прикладі персонального комп'ютера з мережі VLAN 37 – PC16 та персонального комп'ютера з мережі VLAN 47 – PC5 (див. рис. 3.36).

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC16	PC5	ICMP		0.000	N	0	(edit)	

Рисунок 3.36 – Перевірка маршрутизації між VLAN

Перевірка роботи технологію NAT на прикладі Torholskyi_Router3. Перевірка здійснюється за допомогою команди show ip nat statistics, після відправки двох пакетів з персонального комп'ютера PC5 в мережі VLAN 47 до віддаленого персонального комп'ютера PC39 (див. рис. 3.37).

```
Torholskyi_Router3#show ip nat statistics
Total translations: 5 (2 static, 3 dynamic, 3 extended)
Outside Interfaces: Serial0/1/0
Inside Interfaces: GigabitEthernet0/0/0 , GigabitEthernet0/0/1
Hits: 2  Misses: 1236
Expired translations: 0
Dynamic mappings:
-- Inside Source
access-list NAT pool Internet refCount 3
 pool Internet: netmask 255.255.255.224
       start 209.165.200.5 end 209.165.200.30
       type generic, total addresses 26 , allocated 1 (3%), misses 0
```

Рисунок 3.37 – Перевірка технології NAT

Перевірка роботи технології VPN на прикладі маршрутизатору Torholskyi_Router5. Перевірка здійснюється за допомогою команди show crypto ipsec sa, після відправки двох пакетів з персонального комп'ютера PC40 в віддаленої мережі LAN_3 до персонального комп'ютера PC33 в мережі офісу LAN_5 (див. рис. 3.38).

```
Torholskyi_Router5#show crypto ipsec sa

interface: Serial0/0/0
    Crypto map tag: MAP, local addr 64.100.13.1

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (172.24.185.128/255.255.255.128/0/0)
   remote  ident (addr/mask/prot/port): (172.24.184.0/255.255.255.0/0/0)
   current_peer 209.165.202.2 port 500
    PERMIT, flags={origin_is_acl,}
   #pkts encaps: 1, #pkts encrypt: 1, #pkts digest: 0
   #pkts decaps: 2, #pkts decrypt: 2, #pkts verify: 0
   #pkts compressed: 0, #pkts decompressed: 0
   #pkts not compressed: 0, #pkts compr. failed: 0
   #pkts not decompressed: 0, #pkts decompress failed: 0
   #send errors 1, #recv errors 0

     local crypto endpt.: 64.100.13.1, remote crypto endpt.:209.165.202.2
     path mtu 1500, ip mtu 1500, ip mtu idb Serial0/0/0
     current outbound spi: 0x0(0)

     inbound esp sas:

 --More--
```

Рисунок 3.38 – Перевірка технології VPN

Перевірка роботи веб-сайту з відомостями про тему, мету та завдання на кваліфікаційну роботу з використанням доменного імені на прикладі персонального комп’ютера PC33 з підмережі LAN_5 (див. рис. 3.39).

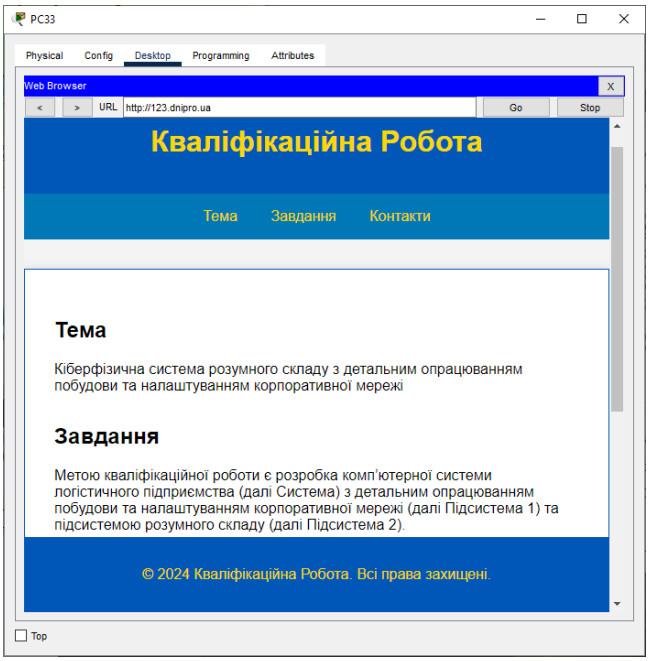

Рисунок 3.39 – Веб-сайт підприємства

Перевірка роботи веб-сайту з відомостями про тему, мету та завдання на кваліфікаційну роботу з використанням зовнішньої IP-адреси на прикладі персонального комп’ютера PC39 (див. рис. 3.40).

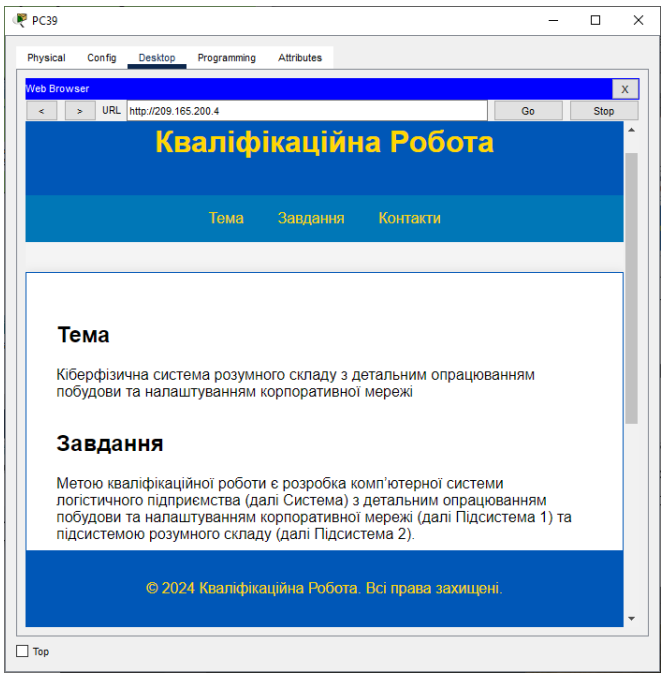

Рисунок 3.40 – Веб-сайт підприємства

4 РОЗРОБКА КОМПОНЕНТА СИСТЕМИ

4.1 Загальні відомості

Кіберфізична система розумного складу – це комплекс інтелектуальних пристроїв та технологій, призначених для підвищення продуктивності та автоматизації виконання складських операцій.

Кіберфізична система складається з десяти маніпуляторів, оснащених додатковими платами розширення для керування роботою сервоприводів та Wi-Fi модулями для дистанційного керування за допомогою мобільного додатку.

Основною платою керування виступає плата Arduino Uno на базі мікроконтролеру ATmega328P (див. рис. 4.1). Пам’ять мікроконтролеру ATmega328P: 32 Кб Flash-пам’яті, 2 Кб SRAM та 1 Кбайт EEEPROM пам’яті.

В таблиці 4.1 наведені основні компоненти плати Arduino Uno.

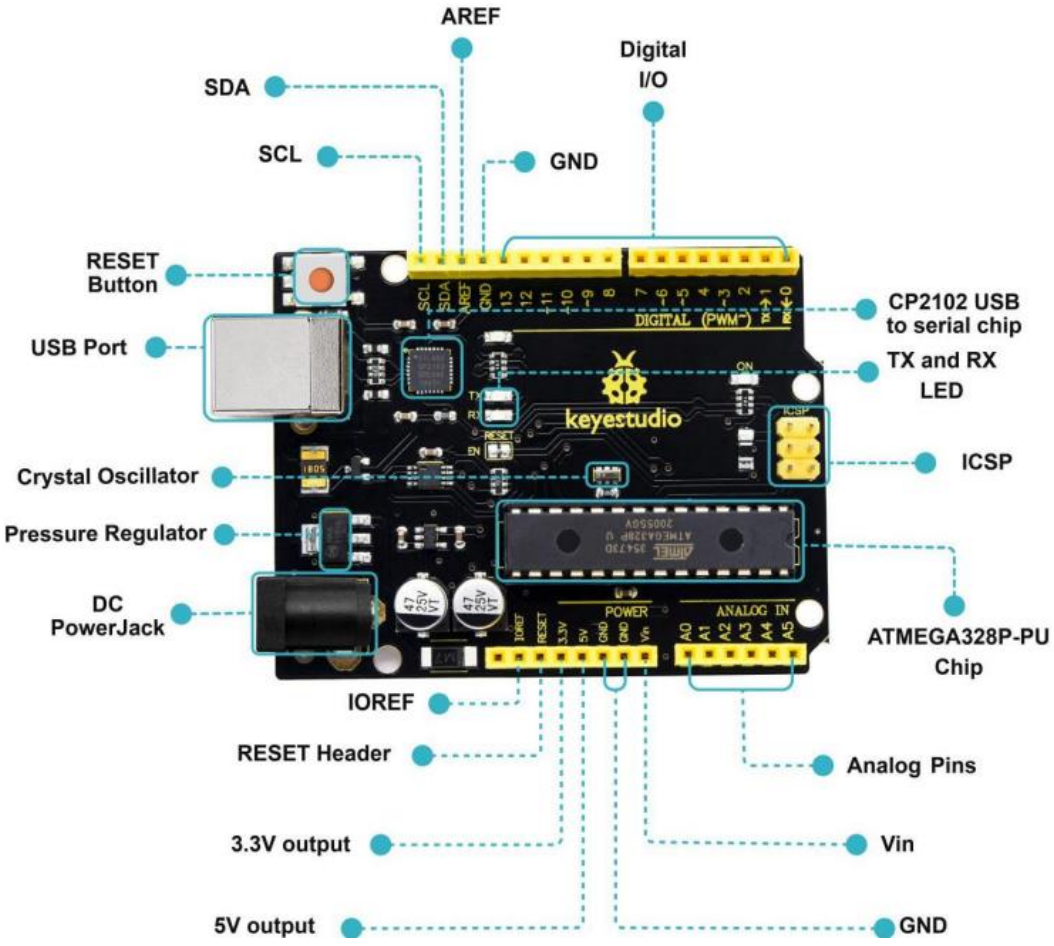

Рисунок 4.1 – Arduino Uno

Таблиця 4.1 – Основні компоненти плати Arduino Uno

Назва порту	Призначення
ICSP	Інтерфейс SPI
IOREF	Налаштування опорної напруги з якою працює мікроконтролер
AREF	Аналоговий опорний вхід. Налаштування зовнішньої опорної напруги для вхідних аналогових сигналів
GND	Загальний (нульовий)
RX	Прийом серійного інтерфейсу UART. Отримання даних серійним інтерфейсом
TX	Передача серійного інтерфейсу UART. Передача даних серійним інтерфейсом
Digital I/O	Цифрові входи/виходи
Analog Pins	Аналогові входи/виходи
SDA	Лінія даних (I2C)
SCL	Тактова лінія (I2C)
5V output	Вихід живлення напругою 5 В
3.3V output	Вихід живлення напругою 3.3 В
RESET button	Кнопка скидання мікроконтролера
RESET header	Підключення зовнішньої кнопки скидання
Vin	Вхід зовнішнього джерела живлення
DC Power Jack	Додатковий вхід для зовнішнього джерела живлення 7-12 В
Pressure Regulator	Контроль напруги, яка подається на плату
CP2102	Перетворювач USB-UART
Crystal Oscillator	Розрахунок часу
ATMEGA328P-PU Chip	Мікроконтролер плати

В якості виконавчих пристроїв маніпулятору використовуються чотири сервоприводи Micro Servo 9G (див. рис. 4.2).

Рисунок 4.2 – Сервопривод маніпулятору

Керування позиціюванням сервоприводів відбувається за допомогою ШІМ (широтно-імпульсної модуляції).

Технічні характеристики сервоприводу Micro Servo 9G:

− робоча напруга: 4.8-6 В;

− максимальний крутний момент: 1.8 кг/см (при напрузі 4.8 В);

− швидкість повороту: 0.12 с/60° (при напрузі 4.8 В);

− кут повороту: 180° (±90° від центральної точки);

− матеріал редуктору: нейлон (пластик);

− споживаний струм в русі: 50-80 мА;

− споживаний струм в утриманні: 5-10 мА

− тип керування: PWM (широтно-імпульсна модуляція).

4.2 Обґрунтування технічних засобів

4.2.1 Обґрунтування апаратних засобів

Для зручного керування роботою сервоприводів та Wi-Fi модулю було обрано плату розширення Keyestudio TB6612FNG Servo Motor Driver Shield.

Keyestudio TB6612FNG Servo Motor Driver Shield – це плата розширення, що підключається до основної плати керування Arduino, з метою простого та зручного керування сервоприводами та підключення Wi-Fi модулю. Вона значно спрощує процес розробки, забезпечуючи зручний інтерфейс для одночасного підключення декількох сервоприводів (виведено окремі групи пінів для підключення периферії), Wi-Fi модулю. Плата має додатковий роз’єм живлення, що дозволяє використовувати зовнішнє джерело живлення напругою від 7 до 12 В та силою струму до 3 А не перевантажуючи плату Arduino. Перемикання на зовнішнє джерело живлення відбувається за допомогою відповідного перемикача, який розташований поруч з роз’ємом.

Для стабільної роботи сервоприводів плата розширення має драйвер PCA9685. Вбудований тактовий генератор цього драйверу дозволяє зняти навантаження з мікроконтролера.

Схема підключення сервоприводів маніпулятору до плати розширення наведена на рисунку 4.3.

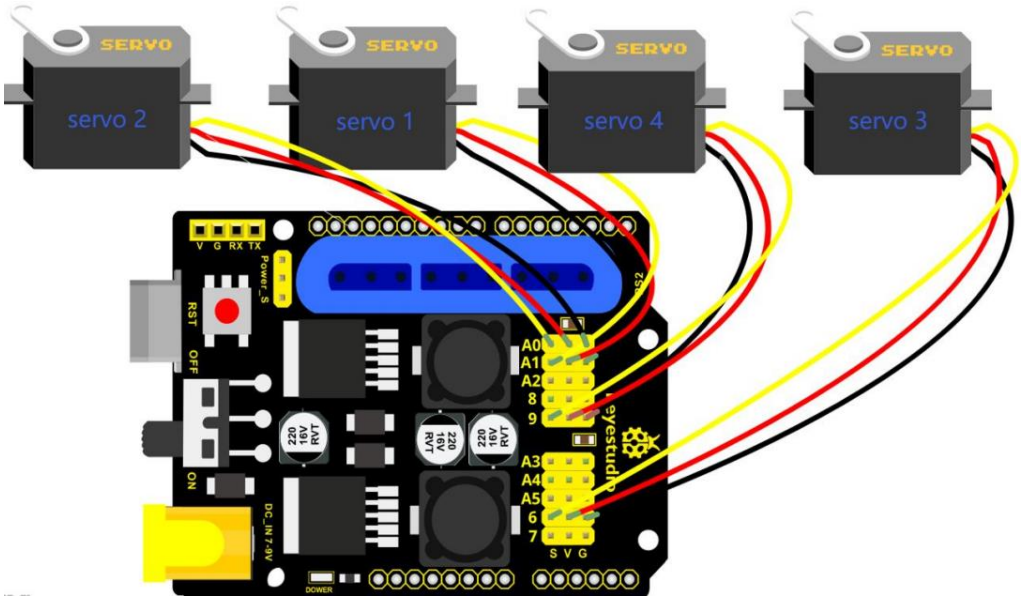

Рисунок 4.3– Схема підключення сервоприводів до плати розширення

Для дистанційного керування роботою маніпуляторів був обраний Wi-Fi модуль ESP32-CAM.

ESP32-CAM – це модуль, що поєднує плату розробки ESP32 та камеру OV2640 на 2 Мп (див. рис. 4.4). Модуль має компактні розміри та має невелике електроспоживання, що дає змогу під’єднати напряму її до плати розширення без додаткового джерела живлення.

Завдяки наявності роз’єму для підключення камери, цей модуль в подальшому може бути використаний як камера спостереження за діями маніпулятору, як система розпізнавання облич і жестів або сканування QR-кодів тощо.

ESP32-CAM має потужний 2-ух ядерний 32-бітовий процесор Xtensa з тактовою частотою в діапазонів від 80 до 240 МГц. Модуль підтримує зв’язок через бездротові технології Wi-Fi та Bluetooth, додатково на ньому розташований слот для карт пам’яті типу Micro-SD до 4 Гб.

В таблиці 4.2 наведені призначення виводів модулю ESP32-CAM.

Рисунок 4.4 – Wi-Fi модуль ESP32-CAM

Таблиця 4.2 – Призначення виводів модулю ESP32-CAM

Назва порту	Призначення
5V	Живлення 5 В
GND	Загальний (нульовий)
GPIO12	Прийом серійного інтерфейсу UART. Отримання даних серійним інтерфейсом
GPIO13	Передача серійного інтерфейсу UART. Передача даних серійним інтерфейсом
GPIO15-14	Загальний вхід/вихід
GPIO2-4	Загальний вхід/вихід
3V3	Живлення 3.3 В
GPIO16	Загальний вхід/вихід
GPIO0	Режим завантаження (Boot Mode)
VCC	Живлення 3.3, 5 В
GPIO1,3	Загальний вхід/вихід

Технічні характеристики ESP32-CAM:

− стандарт Wi-Fi: 802.11 b/g/n;

− стандарт Bluetooth: 4.2 LE з друкованою антеною;

− об’єм SPI Flash-пам’яті: 32 Мбіт;

− максимальний об’єм Micro-SD карти пам’яті: до 4 Гб;

− розміри: 40.5*27*4.5 мм;

− формат зображення: JPEG (підтримка лише OV2640), BMP у відтінках сірого;

− роз’єм підключення камери: FPC;

− роздільна здатність камери: 2 Мп.

Схема підключення модулю ESP32-CAM до плати розширення наведена на рисунку 4.5.

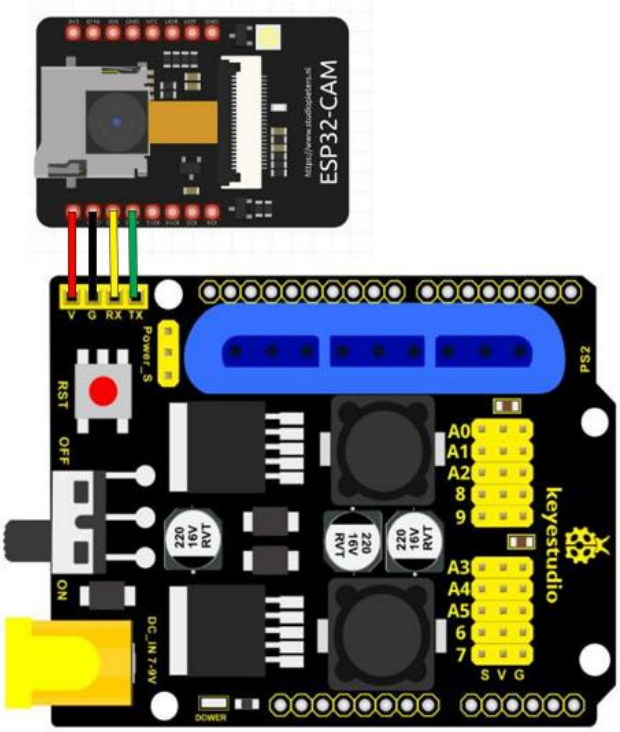

Рисунок 4.5 – Схема підключення Wi-FI модулю до плати розширення

4.2.2 Обґрунтування ПЗ

Для програмування основної плати Arduino Uno та модулю ESP32-CAM було використано середовище розробки Arduino IDE. Arduino IDE – це інтегроване середовище розробки, яке забезпечує легке програмування та налаштування мікроконтролерів. Переваги використання середовища розробки Arduino IDE:

− інтуїтивно зрозумілий інтерфейс, що робить його гарним вибором як для новачків, так і для досвідчених розробників;

− великий вибір апаратних платформ, включаючи різноманітні плати Arduino, ESP, Raspberry Pi та інші;

– великий вибір бібліотек, що спрощують розробку, використовуючи вже готові рішення типових задач;

– крос-платформеність, що дозволяє запускати середовище розробки на різних операційних системах включаючи Linux, macOS, Windows;

– оновлення, Arduino IDE постійно вдосконалюється та оновлюється, що дозволяє використовувати нові функції та можливості.

В налаштуваннях Arduino IDE вказано посилання на менеджер плат ESP, а саме https://dl.espressif.com/dl/package_esp32_index.json, що наведено на рисунку 4.6 та встановлено пакет плат esp32 від Espressif Systems 2.0.17, що наведено на рисунку 4.7.

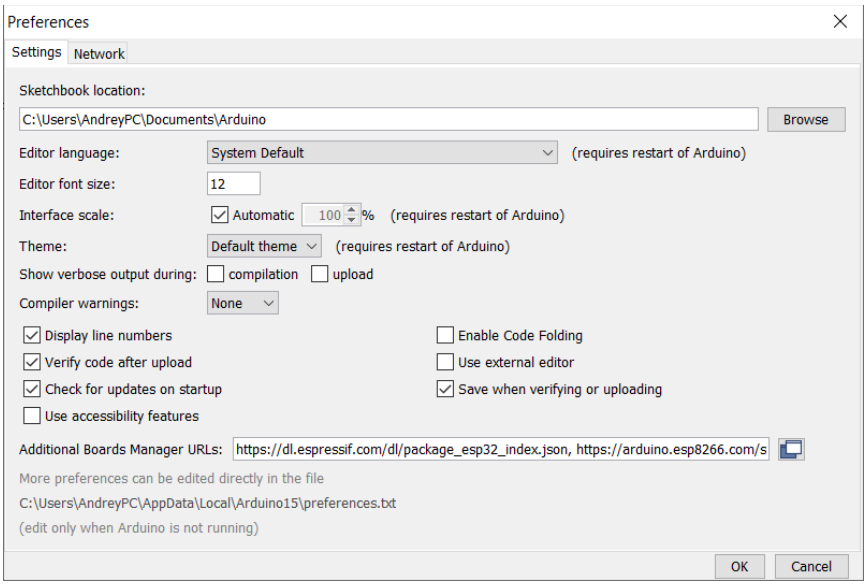

Рисунок 4.6 – Налаштування посилання на менеджер плат ESP

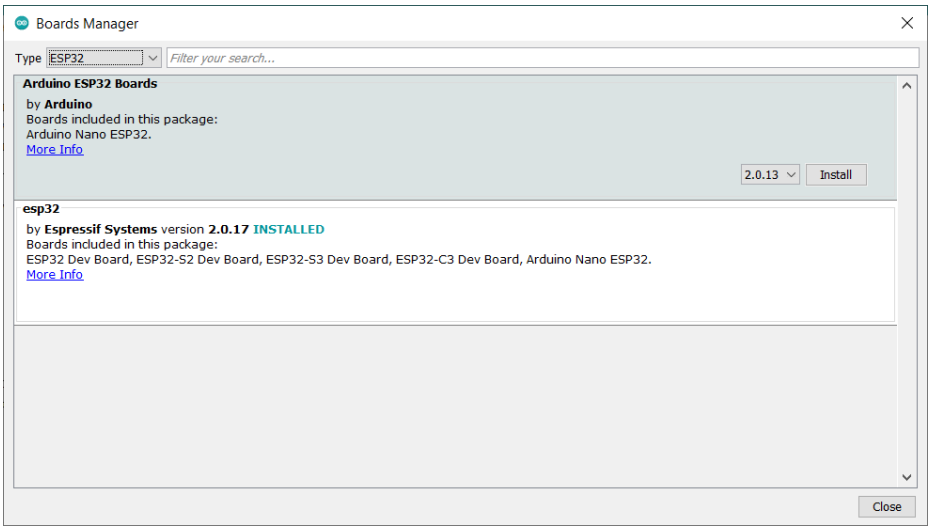

Рисунок 4.7 – Менеджер плат

При роботі з модулем ESP32-CAM додатково була встановлена бібліотека ArduinoWebsockets 0.5.3. Це бібліотека для написання сучасних websockets-додатків за допомогою Arduino.

При роботі з платою Arduino Uno додатково була встановлена бібліотека ServoSmooth 3.9. Це бібліотека для плавного управління сервоприводами за допомогою Arduino.

На рисунку 4.8 наведені встановлені бібліотеки в середовищі Arduino IDE.

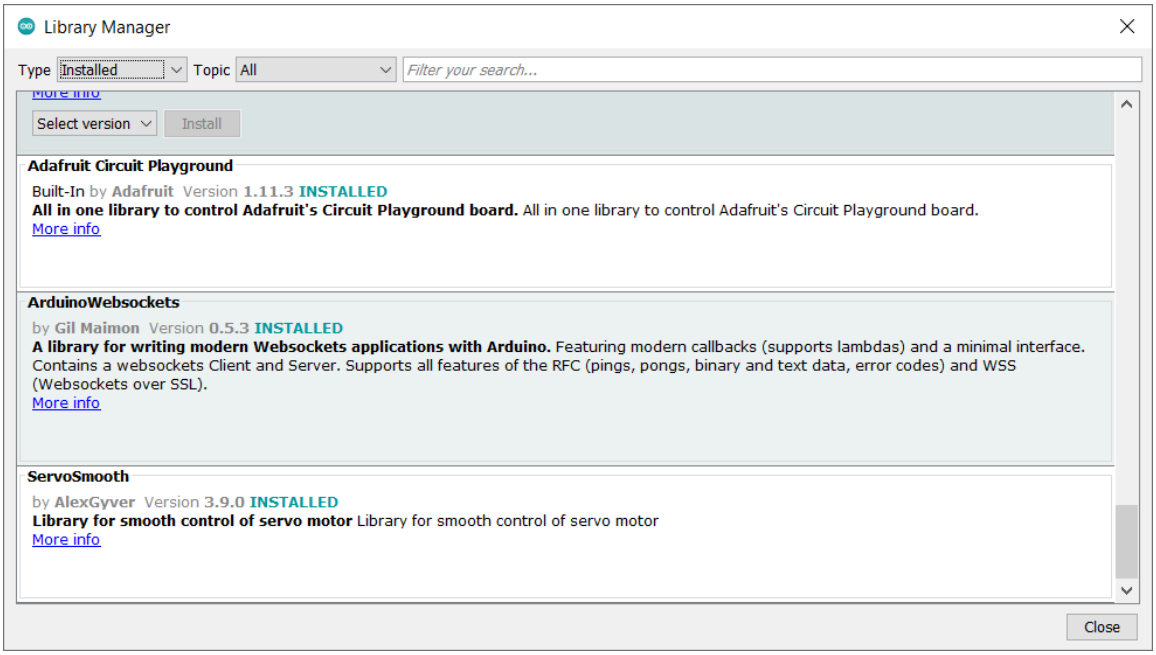

Рисунок 4.8 – Встановлені бібліотеки в середовищі Arduino IDE

Для розробки програмного інтерфейсу мобільного застосунку, було використане середовище розробки Android Studio. Android Studio – це інтегроване середовище розробки, яке потужні інструменти у створені, тестуванні та налагоджені додатків для мобільної платформи Android. Переваги використання середовища розробки Android Studio:

– підтримка великої кількості плагінів та розширень, що дозволяють інтегрувати нові функції;

– наявність регулярних оновлень, що дозволяє використовувати нові функції та можливості;

– підтримка мов програмування Java та більш сучасної Kotlin;

− наявність вбудованого емулятору Android, що дозволяє тестувати розроблені додатки;

− підтримка різних версій Android, що забезпечує сумісність та тестування на різних пристроях;

− наявність інтеграції з GitHub, що дозволяє зробити спільну розробку мобільного додатку більш зручнішою.

Мовою програмування програмного інтерфейсу застосунку обрано Kotlin. Kotlin – мова програмування, яка широко використовується для розробки сучасних Android застосунків.

Переваги використання мови програмування Kotlin:

− сумісність з мовою програмування Jave, що дозволяє користуватись її бібліотеками та навпаки;

− безпека типів, що дозволяє знаходити помилки на етапі компіляції проєкту;

− наявність постійних оновлень;

− підтримка в середовищі розробки Android Studio;

− дозволяє писати більш стиснутий та зрозумілий код, що дозволяє використовувати менше шаблонного код.

При побудові програмного інтерфейсу був використаний фреймворк Jet Compose. Jetpack Compose – це фреймворк, для розробки графічного інтерфейсу Android застосунків, який використовує декларативний підхід.

Переваги використання фреймворку Jetpack Compose:

− декларативний підхід, що дозволяє описувати зовнішній вигляд інтерфейсу у вигляді компонентів та їх зв’язків;

− швидкість та ефективність розробки графічних інтерфейсів;

− мультиплаформеність, що дозволяє розробляти під Android та IOS.

В якості протоколу прикладного рівня передачі даних модулю ESP32-CAM та мобільного пристрою обраний протокол WebSockets. WebSockets – це протокол прикладного рівня, що забезпечує 2-ох сторонню взаємодію клієнту та серверу з використанням одного з’єднання TCP.

Переваги використання протоколу WebSockets:

− WebSockets встановлює постійне з’єднання клієнт-сервер, без необхідності встановлення нового з’єднання;

− низька затримка передачі даних;

− крос-платформеність, що дозволяє взаємодію різних пристроїв таких як, веб-браузери, сервери, мобільні пристрої.

4.3 Розробка математичної моделі роботи маніпулятору

Маніпулятор можна розглядати як ланцюг, який складається з декількох твердих тіл (ланок), які послідовно з’єднані та приводяться в рух виконавчими пристроями. Один кінець такого ланцюгу з’єднується з основою, а інший кінець вільний та з’єднується з виконавчим пристроєм (захоплювачем), що надає змогу впливати на об’єкти, здійснювати їх переміщення, захоплення тощо. [7]

Кінематика маніпулятору вивчає геометрію руху щодо попередньо визначеної абсолютної системи координат, не беручи до уваги сили та моменти, які породжують цей рух.

Існує дві основні задачі кінематики маніпулятору:

− за відомим вектором приєднаних кутів – узагальнених координат $q(t)=(q1(t), ..., qn(t))T$ та визначеним геометричними параметрами ланок (n – кількість ступенів свободи) розрахувати положення та орієнтацію захоплювача маніпулятору до абсолютної системи координат. Це пряма задача кінематики;

− за відомими геометричними параметрами ланок розрахувати можливі вектори змінних маніпулятору, що забезпечують задані положення та орієнтацію захоплювача до абсолютної системи координат. Це зворотна задача кінематики.

4.3.1 Рівняння прямої задачі кінематики

Для планування рухів робота, необхідно зрозуміти взаємозв'язок між виконавчими пристроями та кінцевим положенням робота в робочому

середовищі. Для стаціонарних маніпуляторів всі відносно просто: знаючи положення або кут кожного суглоба, можна обчислити положення кінцевих виконавчих пристроїв (кінцівок) за допомогою тригонометричних розрахунків

Отже основне завдання рівняння прямої кінематики це визначення поточного положення виконавчого пристрою (захоплювача).

Для вирішення рівняння була розроблена кінематична модель дволанкового маніпулятору з двома обертовими парами (див. рис. 4.9).

Кожна з ланок є абсолютно жорсткою, довжиною L. Перша ланка L_1 закріплена до основи та розташовується під кутом Q_1 до неї. Друга ланка L_2 закріплена до кінця першої ланки та розташовується під кутом Q_2 до неї. На кінці другої ланки розташований виконавчий пристрій (захоплювач).

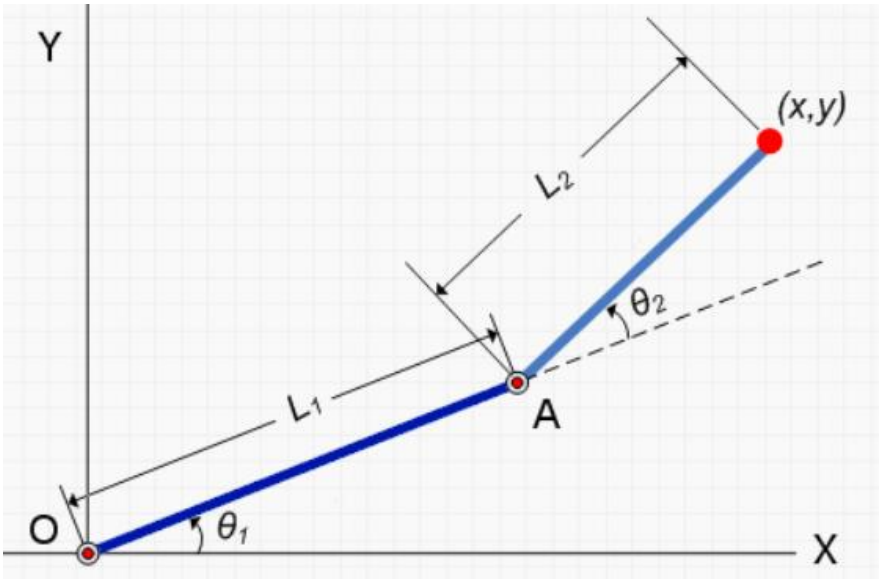

Рисунок 4.9 – Кінематична модель дволанкового маніпулятору

Виходячи з цього маємо першу систему відліку, яка пов’язана з початком координат в точці кріплення ланки першої ланки, в точці O.

Для початку необхідно визначити зміщення другої системи відліку відносно (координати точки A) першої (4.1):

$$x_A = l * \cos(q_1);\ y_A = l * \sin(q_1) \tag{4.1}$$

Розрахунок координат (x,y) в системі відліку першої ланки (4.2):

$$x^{"} = l * \cos(q_2);\ y^{"} = l * \sin(q_2) \tag{4.2}$$

З рисунку 4.13 видно, що стосовно точки O, друга ланка повернута відносно першої на (Q_1+Q_2) (4.3):

$$x' = l * \cos(q_1 + q_2);\ y' = l * \sin(q_1 + q_2) \tag{4.3}$$

Формула для рішення, задачі прямої кінематики (4.4):

$$\begin{aligned} x &= x_A + x' = l * \cos(q_1) + l * \cos(q_1 + q_2) \\ y &= y_A + y' = l * sin(q_1) + l * \sin(q_1 + q_2) \end{aligned} \tag{4.4}$$

4.3.2 Рівняння зворотної задачі кінематики

Для досягненні заданого положення виконавчого пристрою (захоплювачу) необхідно визначити положення кожного суглобу та інвертувати співвідношення між суглобами та виконавчим пристроєм.

Отже основне завдання рівняння зворотної кінематики це розрахунок кута та положення суглобів, необхідного для досягнення виконавчим пристроєм (захоплювачем) заданої точки.

Для вирішення рівняння була розроблена кінематична модель дволанкового маніпулятору з двома обертовими парами (див. рис 4.10).

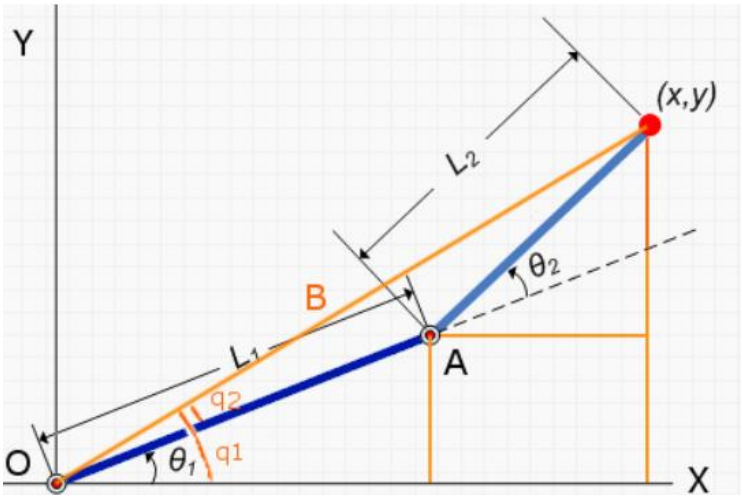

Рисунок 4.10 – Кінематична модель дволанкового маніпулятору

З відомими декартовими координатами (x, y) виконавчого пристрою за допомогою зворотної задачі кінематики можна знайти узагальнені координати Q_1 та Q_2. Проведемо додаткові побудови та позначимо кут q_2, для якого за теоремою косинусів отримуємо рівняння (4.5):

$$B^2 = l^2 + (x^2 + y^2) - 2 * l * \sqrt{x^2 + y^2} * \cos(q_2) \quad (4.5)$$

Виразимо q_2 та отримаємо вираз (4.6):

$$q_2 = arccos \frac{l^2 + (x^2 + y^2) - B^2}{2 * l * \sqrt{x^2 + y^2}} \quad (4.6)$$

З малюнка 4.13 видно: $tg(Q_1 + q_2) = y/x$. Отримуємо формулу (4.7):

$$Q_1 = arctg \frac{y}{x} - q_2 \quad (4.7)$$

Розпишемо отриманий вираз через тангенс (4.8):

$$tg(Q_1 + Q_2) = \frac{y - l * \sin(Q_1)}{x - l * \cos(Q_1)} \quad (4.8)$$

Виразимо Q_2 та отримаємо вираз (4.9):

$$Q_2 = arctg \frac{y - l * \sin(Q_1)}{x - l * \cos(Q_1)} - Q_1 \quad (4.9)$$

Таким чином, рішення зворотного завдання кінематики матиме вигляд (4.10):

$$Q_1 = arctg \frac{y}{x} - arccos \frac{l^2 + (x^2 + y^2) - B^2}{2 * l * \sqrt{x^2 + y^2}}$$
$$Q_2 = arctg \frac{y - l * \sin(Q_1)}{x - l * \cos(Q_1)} - Q_1 \quad (4.10)$$

Отримані рівняння дозволяють визначити необхідне положення ланок маніпулятора виходячи базуючись на його поточних координат. Перше рівняння дозволяє нам знайти координату Q_1 за відомими координатами (x, y), а друге кут Q_2.

Важливою особливістю зворотної кінематики є специфіка розрахунку тригонометричних функцій. В результаті таких обчислень не виключене виникнення інтервалів, для яких не буде розв’язків (особливо при роз розрахунку функції арккосинусу).

При наявності трьох та більше ланок задачі зворотної кінематики можуть необмежену кількість рішень.

Універсального розв’язку для задачі зворотної кінематики не існує. Різні кінематичні схеми потребують різних підходів для вирішення.

4.4 Налаштування апаратних засобів та ПЗ

Для завантаження прошивки на модуль ESP32-CAM був використаний програматор. [8]

Перед початком завантаження прошивки в налаштуваннях Arduino IDE було обрано кінцеву плату AI Thinker ESP32-CAM (рис. 4.11) та порт на який підключений програматор (див. рис. 4.12).

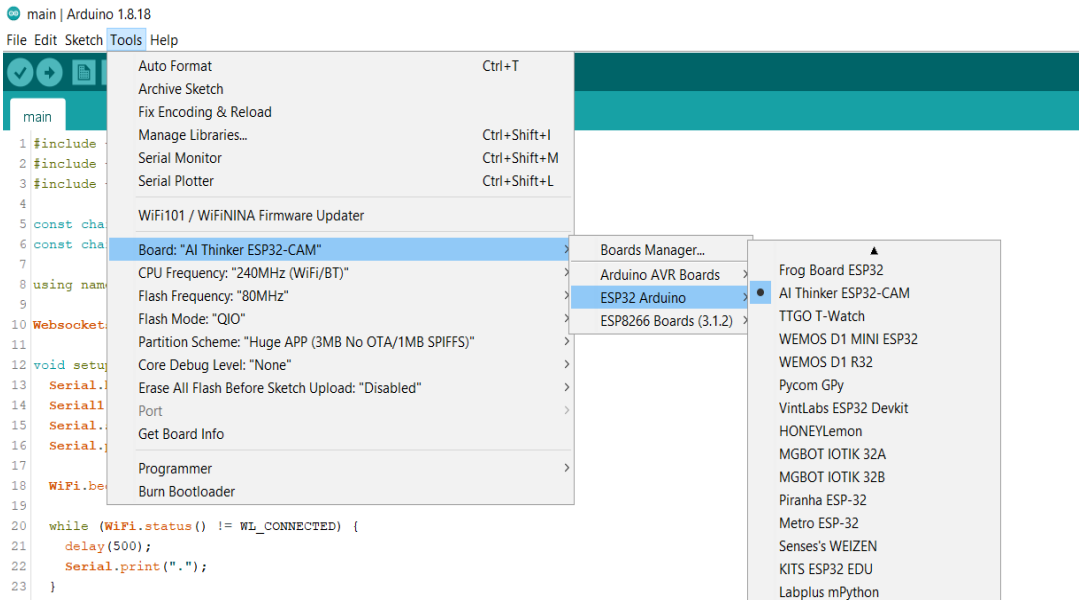

Рисунок 4.11 – Налаштування плати для прошивання

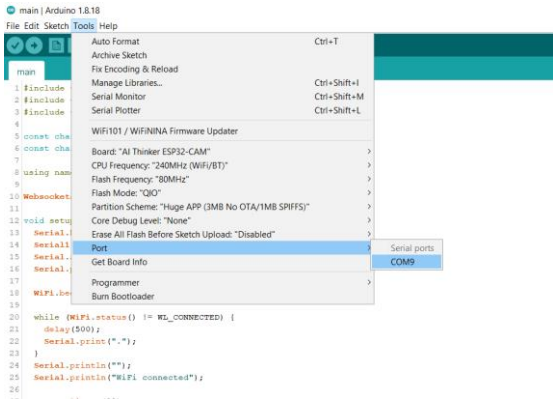

Рисунок 4.12 – Налаштування порту для прошивки

Вхід в режим запису модулю відбувається замиканням контактів GND та GPIO0 з одночасним натисканням кнопки скидання (див. рис. 4.13).

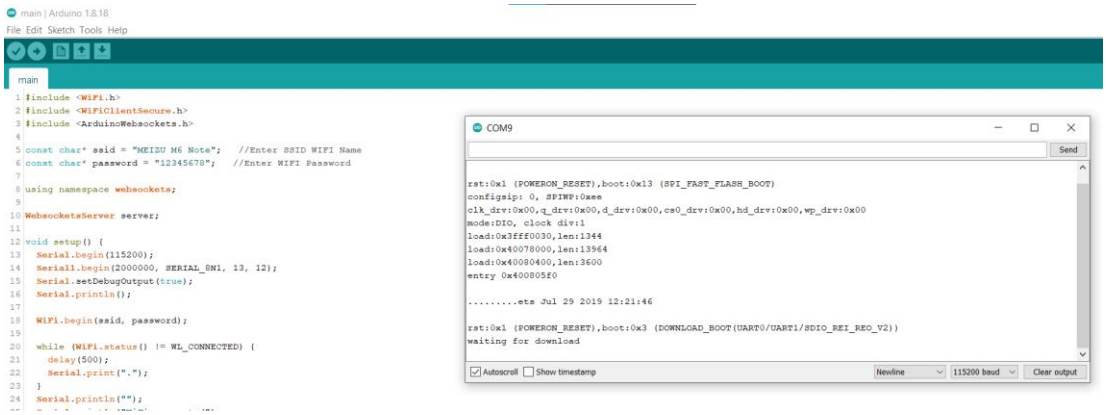

Рисунок 4.13 – Режим запису модулю ESP32-CAM

В якості прошивки було завантажено приклад з бібліотеки esp (див. рис. 4.14). Як видно з послідовного монітору, модулю був назначена IP-адреса 192.168.43.254.

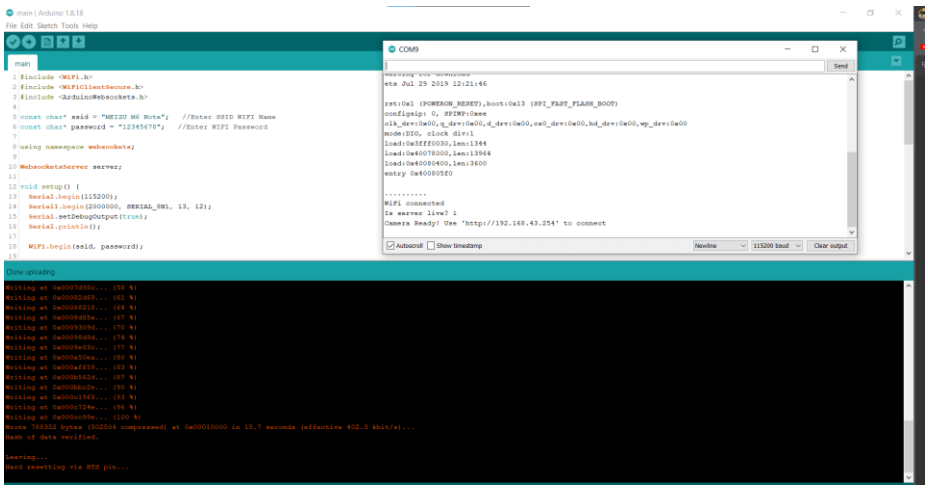

Рисунок 4.14 – Прошитий модуль ESP32-CAM

Для завантаження прошивки на плату Arduino Uno потрібен тільки кабель та підключення до ПК (див. рис. 4.15). В якості прошивки було завантажено приклад з бібліотеки Arduino Blink. [9]

Рисунок 4.15 – Прошита плата Arduino Uno

В Android Studio створений шаблон проєкту Basic View Activity (див. рис. 4.16). Був використаний вбудований емулятор, для відображення роботи проєкту. [10]

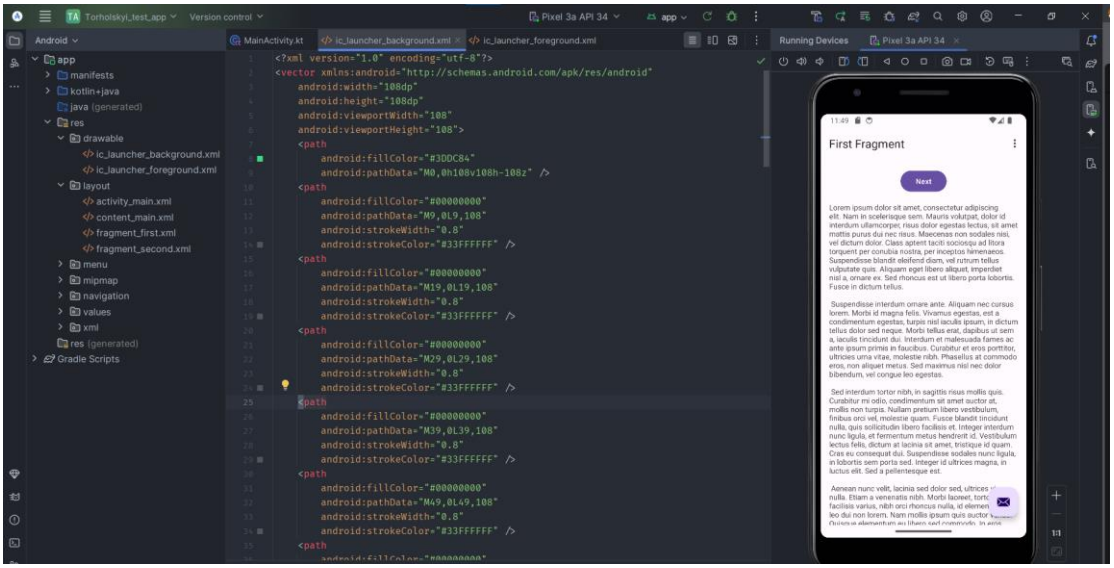

Рисунок 4.16 – Шаблон проєкту Basic View Activity

4.5 Опис розробленої програми

4.5.1 Загальні відомості

Для контролю, передачі команд управління і даних, між плагінами та маніпулятором, був розроблений програмний інтерфейс (API), що контролює параметри команд, які передаються від плагінів до маніпулятора.

API містить функції, які спрощують розробку плагінів для програми. Функції такі як пряма та зворотна кінематика, були реалізовані один раз, в інтерфейсі.

Інтерфейс надає розробнику методи для керування маніпулятором (пересування захоплювачу по всім осям з заданим кроком, переміщення захоплювачу в задану позицію на координатній площині відносно самого бази маніпулятора).

Програма керування маніпулятором надає такі функції:

− поворот захоплювача вліво/вправо;

− переміщення захоплювача вперед/назад;

− переміщення захоплювача вгору/вниз;

− захоплення/утримання захоплювача.

Діаграма класу інтерфейсу зображена на рисунку 4.17.

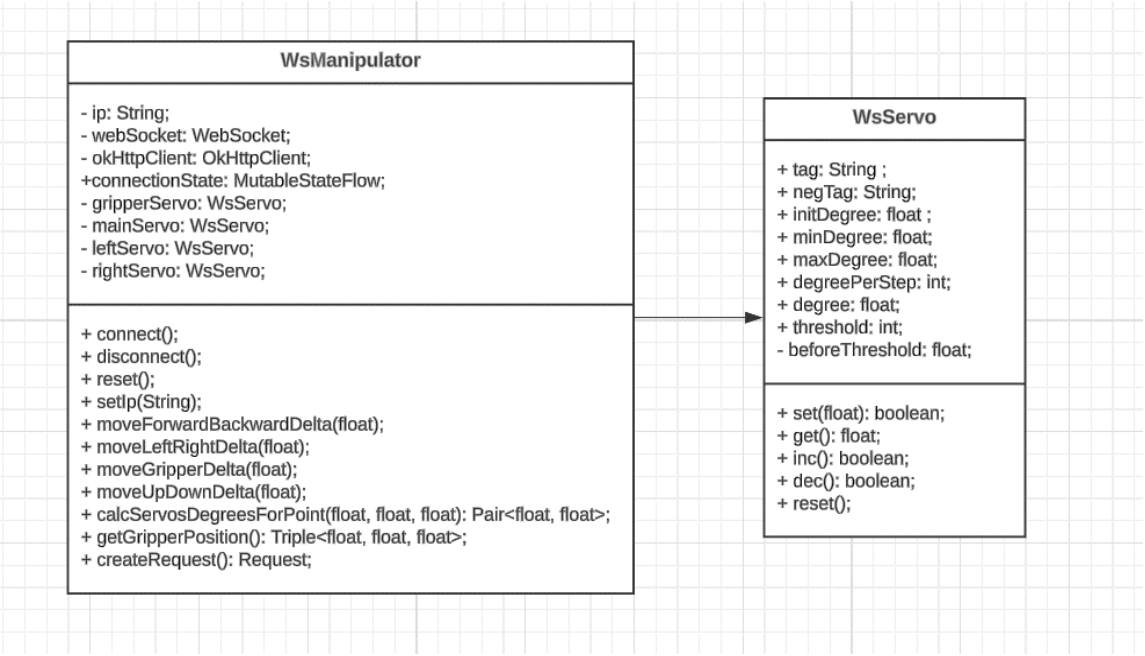

Рисунок 4.17 – Діаграма класу інтерфейсу

4.5.2 Початкове вікно програми

Початкове вікно програми, в якому містяться всі пункт меню ручного керування Manual Control та 5 полів для введення IP адреси маніпуляторів. Початкове вікно зображене на рисунку 4.18.

Для з’єднання з маніпулятором, необхідно ввести IP-адресу модулю ESP32-CAM, що з’являлась під час прошивки та натиснути кнопку Connect.

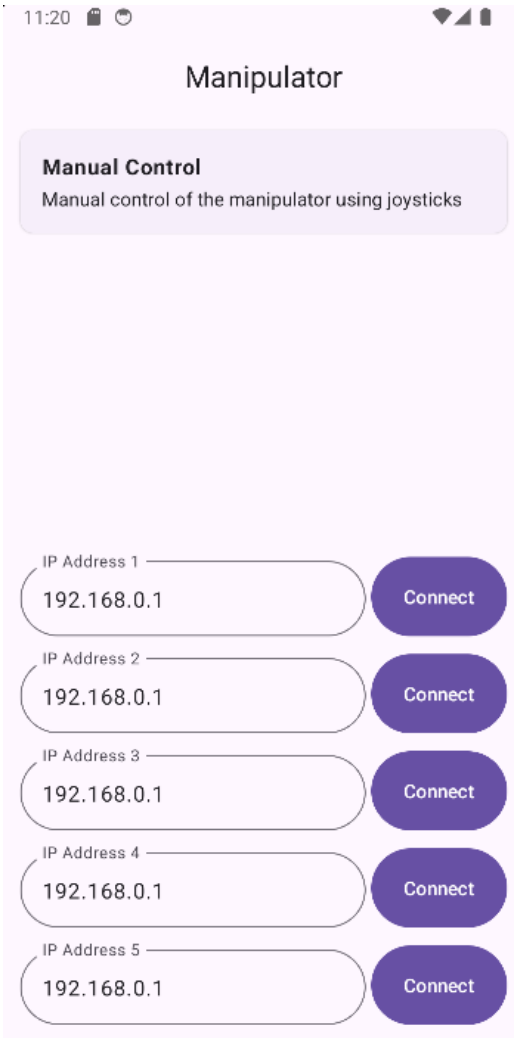

Рисунок 4.18 – Початкове вікно програми

4.5.3 Вікно Manual Control

Вікно Manual Control зображене на рисунку 4.19. При невдалому підключені до маніпулятору, кнопки керування будуть не активні.

Для повторного підключення до маніпулятору використовується іконка маніпулятору, розташована зліва згори. Після вдалого підключення маніпулятору, кнопки керування стануть активними (див. рис. 4.20).

Кнопка (Up\Down) здійснює переміщення захоплювачу вгору та вниз.

Кнопка (Forw\Back) здійснює переміщення захоплювачу вперед та назад.

Кнопка (Grip) здійснює захоплення та утримання захоплювачу.

Кнопка (Left\Right) здійснює переміщення захоплювачу вліво та вправо.

Для повернення до головного меню використовується стрілка назад зліва вгорі.

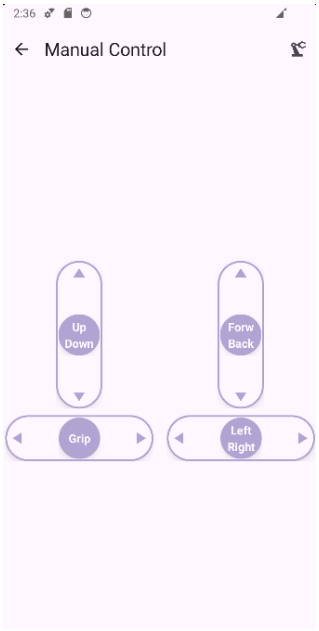

Рисунок 4.19 – Вікно Manual Control з відключеним маніпулятором

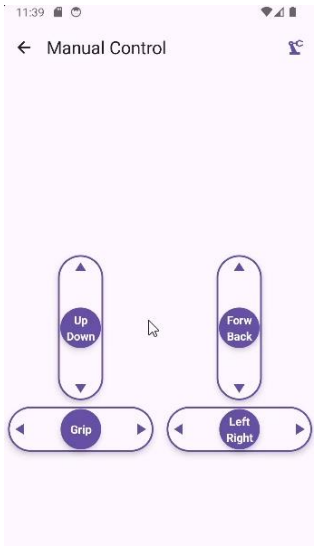

Рисунок 4.20 – Вікно Manual Control з підключеним маніпулятором

ВИСНОВКИ

В даній кваліфікаційній роботі була створена кіберфізична комп’ютерна системи розумного складу з детальним опрацюванням побудови та налаштування корпоративної мережі.

Був проведений аналіз об’єкту та сформульовані технічні вимоги до Системи. Відповідно до цих вимог обрано необхідне апаратне забезпечення.

Згідно завдання була розрахована адресація підмереж, базові налаштування активного обладнання, налаштування технологій DHCP, NAT, VLAN, VPN, AAA, OSPF.

Після створення основної мережі, було розроблено кіберфізичну систему дистанційного керування роботою маніпулятору за допомогою мобільного застосунку. Застосунок забезпечує користувачу зручний інтерфейс керування рухами маніпулятору.

Був обґрунтований вибір апаратних та програмних засобів, розроблена математична модель роботи маніпулятору, налаштовані апаратні засоби, середовища розробки та описаний загальний вигляд мобільного додатку та його станів.

В кінці роботи було проведено тестування окремих компонентів Системи та роботи Системи в цілому. Тестування показало, що Система відповідає всім поставленим вимогам та забезпечує надійне та ефективне керування роботою маніпулятора за допомогою мобільного застосунку.

Кваліфікаційна робота виконана в повній відповідності до теми і поставлених завдань. Оформлення роботи відповідає всім нормативним вимогам та рекомендаціям, встановленим методичним керівництвом.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ДСТУ IEC 60038:2015 Еталонна напруга за IEC – [Електронний ресурс] - Режим доступу: http://surl.li/ufxko.

2. Атестація здобувачів вищої освіти. Методичні рекомендації до виконання кваліфікаційної роботи бакалавра студентами галузі знань 12 Інформаційні технології спеціальності 123 Комп'ютерна інженерія / Л.І. Цвіркун, С.М. Ткаченко, Я.В. Панферова, Д.О. Бешта, Л.В. Бешта. – Д.: НТУ «ДП», 2024. – 63 с.

3. Налаштування NAT – Education [Електронний ресурс] – Режим доступу до ресурсу: http://surl.li/ufwxg.

4. Налаштування VLAN – Education [Електронний ресурс] – Режим доступу до ресурсу: http://surl.li/ufwxs.

5. Налаштування VPN – Education [Електронний ресурс] – Режим доступу до ресурсу: http://surl.li/ufwyf.

6. Вступ до автономних роботів – Education [Електронний ресурс] - Режим доступу до ресурсу: http://surl.li/ufxhk.

7. Механіка промислових роботів/під ред. К. В. Фролова, Є. І. Воробйова. Том 1. Кінематика та динаміка. - К: Вища школа, 1988. -304

8. Програмування модулю ESP32-CAM - Education [Електронний ресурс] – Режим доступу до ресурсу: http://surl.li/ufxgb.

9. Програмування контролеру – Education [Електронний ресурс] – Режим доступу до ресурсу: http://surl.li/ufxey.

10. Програмування мобільного додатку - Education [Електронний ресурс] – Режим доступу до ресурсу: http://surl.li/ufxcd.

Додаток А

Текст програми мобільного застосунку для керування роботою маніпулятору

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

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ МОБІЛЬНОГО ЗАСТОСУНКУ КЕРУВАННЯ МАНІПУЛЯТОРОМ

Текст програми

804.02070743.24017-01 12 01

Листів 8

2024

АНОТАЦІЯ

Дана програма містить в собі частини програмного коду для кіберфізичної системи розумного складу. Одна з частин призначена для графічного відображення компонентів на екрані, а інша реалізовує функціонал.

Програма написана мовою Kotlin у середовищі розробки Android Studio та призначена для застосування на мобільних пристроях з операційною системою не нижче версії 7.1.

Програма призначена для дистанційного керування роботою маніпулятору (розвантаження, завантаження, переміщення).

ЗМІСТ

1 Файл MainActivity.kt……...4

2 Файл WsManipulator.kt……4

3 Файл WsServo.kt……..6

1 Файл MainActivity.kt

```
class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        // Встановлення вмісту (інтерфейсу) активності
        setContent {

            // Встановлення теми застосунку
            ManipulatorTheme {
                // Контейнер Surface з використанням колірної схеми MaterialTheme
                Surface(
                    modifier = Modifier.fillMaxSize(), // Налаштування модифікатору
розміру
                    color = MaterialTheme.colorScheme.background // Встановлення
колірної схеми
                ) {
                    val navController = rememberNavController() // Створення
контролера навігації

                    // Встановлення точок навігації для контролера навігації
                    NavHost(
                        navController = navController, // Передача контролера
навігації
                        startDestination = "menu" // Початкова точка навігації
                    ) {
                        composable("menu") {

                            // Відображення екрану меню та передача фунуцкції
навігації для екрану ручного керування
                            MenuScreen(
                                navigateToManualControl = {
navController.navigate("manual_control") },
                            )
                        }
                        composable("manual_control") {

                            // Відображення екрану ручного керування та встановлення
обробника натискання кнопки назад
                            ManualControlScreen(
                                onBackClick = navController::popBackStack
                            )
                        }
                    }
                }
            }
        }}}
```

2 Файл WsManipulator.kt

```
class WsManipulator(context: Context) : WebSocketListener() {
    // Об'єкти класу WsServo, що відповідають за рух сервоприводів
    private var gripperServo = WsServo(tag = "g", initDegree = 80F, maxDegree = 80F)
    private var mainServo = WsServo(tag = "m", initDegree = 90F, minDegree = 5F,
maxDegree = 175F)
    private var leftServo = WsServo(tag = "l", initDegree = 130F, maxDegree = 160F)
    private var rightServo = WsServo(tag = "r", maxDegree = 90F)
    private var ip = "192.168.234.111"

    // Стан підключення маніпулятору
    val connectionState =
MutableStateFlow<ManipulatorState>(ManipulatorState.Disconnected(""))

    // Об'єкт okHttpClient для взаємодії з WebSocket
    private val okHttpClient = OkHttpClient()
```

```
    // WebSocket для з'єднання з маніпулятором
    private lateinit var webSocket: WebSocket

    // Метод для встановлення IP-адреси маніпулятору
    fun setIp(ip: String) {
        this.ip = ip
    }

    // Метод для руху маніпулятору вперед/назад
    fun moveForwardBackwardDelta(delta: Float) {
        var (x, y, z) = getGripperPosition()
        x += delta
        val (a1, a2) = calcServosDegreesForPoint(x, y, z)

        if (leftServo.set(leftServo.degree - a1)) {
            moveServoDelta(leftServo)
        }
        if (rightServo.set(rightServo.degree - a2)) {
            moveServoDelta(rightServo)
        }
    }

    // Метод для руху маніпулятору ліворуч/праворуч
    fun moveLeftRightDelta(delta: Float) {
        if (mainServo.set(delta)) {
            moveServoDelta(mainServo)
        }
    }

    // Метод для розтискання/стискання захоплювача маніпулятору
    fun moveGripperDelta(delta: Float) {
        if (gripperServo.set(delta)) {
            moveServoDelta(gripperServo)
        }
    }

    // Метод для руху маніпулятору вгору/вниз
    fun moveUpDownDelta(delta: Float) {
        var (x, y, z) = getGripperPosition()
        z += delta
        val (a1, a2) = calcServosDegreesForPoint(x, y, z)
        if (leftServo.set(leftServo.degree - a1)) {
            moveServoDelta(leftServo)
        }
        if (rightServo.set(rightServo.degree - a2)) {
            moveServoDelta(rightServo)
        }
    }

    // Метод для відправдення команди руху
    private fun moveServoDelta(servo: WsServo) {
        val delta = servo.get()
        val tag = if (delta > 0) servo.tag else servo.negTag
        val deltaInt = abs(delta.roundToInt())

        Log.d(TAG, "$tag => $deltaInt")

        if (::webSocket.isInitialized) {
            try {
                webSocket.send("$tag$deltaInt")
            } catch (e: Exception) {
                connectionState.value = ManipulatorState.Disconnected("Fail connect to
manipulator")
                Log.d(TAG, "Manipulator connection error: ${e.message.toString()}")
            }
        }
    }

    // Метод для обчислення кутів сервоприводів для точки
    private fun calcServosDegreesForPoint(x: Float, y: Float, z: Float): Pair<Float,
Float> {
```

```
        val x = x + 0.01F
        val y = y + 0.01F
        val z = z + 0.01F

        val d = sqrt(x.pow(2) + z.pow(2))
        val q1 = atan(z / x)
        var q2 = (d.pow(2) + L1.pow(2) - L2.pow(2)) / (2 * L1 * d)
        var q3 = (L1.pow(2) + L2.pow(2) - d.pow(2)) / (2 * L1 * L2)
        q2 = acos(q2.coerceIn(-1F, 1F))
        q3 = acos(q3.coerceIn(-1F, 1F))
        val a1 = (q1 + q2).toDegree()
        var a2 = (PI.toFloat() - q3).toDegree()

        a2 -= a1
        return Pair(a1, a2)
    }

    // Метод для обчислення позиції захоплювача маніпулятору
    private fun getGripperPosition(): Triple<Float, Float, Float> {
        val x =
            L1 * cos(leftServo.degree.toRadian()) + L2 * cos((leftServo.degree -
(rightServo.degree + leftServo.degree)).toRadian())
        val z =
            L1 * sin(leftServo.degree.toRadian()) + L2 * sin((leftServo.degree -
(rightServo.degree + leftServo.degree)).toRadian())
        Log.d(TAG, "Gripper pos: x: %.2f, y: 0, z: %.2f".format(x, z))
        return Triple(x, 0F, z)
    }

    // Метод для підключення до WebSocket
    fun connect() {
        connectionState.value = ManipulatorState.Connecting
        try {
            webSocket = okHttpClient.newWebSocket(createRequest(), this)
        } catch (e: Exception) {
            connectionState.value = ManipulatorState.Disconnected("Fail connect to
manipulator")
            Log.d(TAG, "Manipulator connection error: ${e.message.toString()}")
        }
    }

    // Метод для відключення від WebSocket
    fun disconnect() {
        try {
            webSocket.close(1000, "Canceled manually.")
            connectionState.value = ManipulatorState.Disconnected("Disconnected")
        } catch (e: Exception) {
            connectionState.value = ManipulatorState.Disconnected("Force
disconnected")
            Log.d(TAG, "Fail disconnect manipulator: ${e.message.toString()}")
        }
    }

    // Метод скидання маніпулятору до початкового стану
    fun reset() {
        try {
            mainServo.reset()
            gripperServo.reset()
            leftServo.reset()
            rightServo.reset()
            webSocket.send("X1")
        } catch (e: Exception) {
            connectionState.value = ManipulatorState.Disconnected("Fail connect to
manipulator")
            Log.d(TAG, "Manipulator connection error: ${e.message.toString()}")
        }
    }

    // Метод для створення запиту до WebSocket
    private fun createRequest(): Request {
        val websocketURL = "ws://${ip}:$PORT"
```

```
        return Request.Builder().url(websocketURL).build()
    }

    // Метод обробки подій WebSocket
    override fun onOpen(webSocket: WebSocket, response: Response) {
        super.onOpen(webSocket, response)
        connectionState.value = ManipulatorState.Connected
        reset()
        Log.d(TAG, "Connected")
    }

    // Метод отримання повідомлення WebSocket від серверу
    override fun onMessage(webSocket: WebSocket, text: String) {
        super.onMessage(webSocket, text)
        Log.d(TAG, "Received: $text")
    }

    // Метод закриття з'єднання  WebSocket
    override fun onClosing(webSocket: WebSocket, code: Int, reason: String) {
        super.onClosing(webSocket, code, reason)
        Log.d(TAG, "onClosing: $code $reason")
    }

    // Метод закритого з'єднання WebSocket
    override fun onClosed(webSocket: WebSocket, code: Int, reason: String) {
        super.onClosed(webSocket, code, reason)
        connectionState.value = ManipulatorState.Disconnected(reason)
        Log.d(TAG, "Closed: $code $reason")
    }

    // Метод обробки помилки під час роботи з WebSocket-з'єднанням
    override fun onFailure(webSocket: WebSocket, t: Throwable, response: Response?) {
        connectionState.value = ManipulatorState.Disconnected("Fail connect to
manipulator")
        Log.d(TAG, "Failure: ${t.message} $response")
        super.onFailure(webSocket, t, response)
    }

    // Статичний член класу, що є спільним для всіх екземплярів
    companion object {
        // Довжина першого сегменту маніпулятору
        private const val L1 = 8F

        // Довжина другого сегменту маніпулятору
        private const val L2 = 8F

        // Порт з'єднаня з WebSocket-з'єдання
        private const val PORT = "82"

        // Тег для логування
        private const val TAG = "Manipulator"

        @Volatile
        // Змінна для збереження єдиного екземпляру класу WsManipulator
        private var INSTANCE: WsManipulator? = null

        // Метод для отримання єдиного екземпляра класу WsManipulator
        fun getInstance(context: Context) = INSTANCE ?: synchronized(this) {

            // Якщо екземпляр не існує, створюємо його та зберігаємо в INSTANCE
            INSTANCE ?: WsManipulator(context).also {
                INSTANCE = it
            }
        }
    }
}
```

3 Файл WsServo.kt

```
class WsServo(
    val tag: String,                // Тег, який ідентифікує сервопривід
    val initDegree: Float = 0F,     // Початкове значення сервоприводу
```

```
    val minDegree: Float = 0F,      // Мінімальне значення сервоприводу
    val maxDegree: Float = 180F,    // Максимальне значення сервоприводу
    val degreePerStep: Int = 2,     // Кількість градусів за крок переміщення
) {
    val negTag = tag.uppercase()    // Тег для зворотнього напрямку
    var threshold = 1               // Поріг зміни на скільки градусів повинен
змінитися серводвигун
    private var beforeThreshold = initDegree; // Збереження попереднього значення
ступеня перед зміною

    var degree = initDegree         // Початкове значення сервоприводу
        set(value) {
            field = value.coerceIn(minDegree, maxDegree) // Забезпечує, щоб значення
ступеня було в межах minDegree та maxDegree
            Log.d(TAG, "$tag: $degree/$maxDegree") // Відображення значення ступеня у
відлагоджувальних повідомленнях
        }

    // Метод для встановлення нового значення сервоприводу
    fun set(value: Float): Boolean {
        var tmp = degree + value    // Тимчасова змінна для нового значення
        tmp = tmp.coerceIn(minDegree, maxDegree) // Забезпечує, щоб нове значення було
в межах minDegree та maxDegree
        return if (tmp == degree) {             // Перевірка, чи нове значення
співпадає з поточним
            false                               // Якщо так, повертається логічне
значення false
        } else {
            degree = tmp                        // Якщо ні, встановлюється нове
значення
            abs(degree - beforeThreshold) >= threshold  // Повертається логічне
значення
        }
    }

    // Метод для отримання різниці між поточним та попереднім значенням сервоприводу
    fun get(): Float {
        return (degree - beforeThreshold).also {
            beforeThreshold = degree     // Оновлення попереднього значення ступеня
        }
    }

    // Метод для інкрементування значення серводвигуна
    fun inc(): Boolean {
        if (degree == maxDegree) return false

        degree += degreePerStep

        return true
    }

    // Метод для декрементування значення серводвигуна
    fun dec(): Boolean {
        if (degree == minDegree) return false

        degree -= degreePerStep

        return true
    }

    // Метод для скидання значення початкового значення серводвигуна
    fun reset() {
        degree = initDegree
        beforeThreshold = initDegree
    }

    // Перевизначений метод toString для повернення тегу серводвигуна
    override fun toString() = tag
}
```

Додаток Б

Текст програми контролеру та Wi-Fi модулю для керування маніпулятором

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

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ КОНТРОЛЕРУ ТА WI-FI МОДУЛЮ ДЛЯ КЕРВУВАННЯ МАНІПУЛЯТОРОМ

Текст програми

804.02070743.24017-01 12 01

Листів 6

2024

АНОТАЦІЯ

Дана програма містить в собі частини програмного коду для кіберфізичної системи розумного складу. Одна з частин призначена програмування плати керування, а інша для програмування Wi-Fi модулю.

Програма написана у середовищі розробки Arduino IDE та призначена для плат керування Arduino Uno та Wi-Fi модулю ESP32-CAM.

Програма призначена для прошивки контролеру плати керування та прошивки Wi-Fi модулю для обміну даними між платою керування та мобільним пристроєм.

ЗМІСТ

1 Файл arduino_serial.ino……..……4

2 Файл main.ino…….6

1 Файл arduino_serial.ino

```
#include <ServoSmooth.h>  // Бібліотека для плавного керування сервоприводами

#define TIMEOUT 100  // таймаут в мілісекундах на відпрацювання неправильно надісланих даних

#define MAIN_SERVO_PIN 6 // визначення піна для підключення сервоприводу бази
#define GRIPPER_SERVO_PIN 7 // визначення піна для підключення сервоприводу захоплювача
#define LEFT_SERVO_PIN 8 // визначення піна для підключення лівого сервоприводу
#define RIGHT_SERVO_PIN 9 // визначення піна для підключення правого сервоприводу

ServoSmooth mainServo, gripperServo, leftServo, rightServo;

// задання початкового, мінімального та максимального положення сервоприводу бази
#define M_INIT 90
#define M_MIN 5
#define M_MAX 175
int mainServoDegree = M_INIT;

// задання початкового, мінімального та максимального положення сервоприводу захоплювачу
#define G_INIT 80
#define G_MIN 0
#define G_MAX G_INIT
int gripperServoDegree = G_INIT;

// задання початкового, мінімального та максимального положення лівого сервоприводу
#define L_INIT 130
#define L_MIN 0
#define L_MAX 160
int leftServoDegree = L_INIT;

// задання початкового, мінімального та максимального положення правого сервоприводу
#define R_INIT 0
#define R_MIN 0
#define R_MAX 90
int rightServoDegree = R_INIT;

int intValue;     // Значення числа для парсингу
char header;      // Заголовок команди
boolean recievedFlag, startParse;   // Флаги для контролю парсингу
unsigned long parseTime;  // Час парсингу

/**
 * uint positive direction step
 * m = main
 * g = gripper
 * l = left
 * r = right
 * uint negative direction step
 * M = main
 * G = gripper
 * L = left
 * R = right
 *
 * X = reset
 */

// Функція для парсингу отриманих даних через послідовний інтерфейс
void parsing() {
  if (Serial.available() > 0) {     // Перевірка наявності даних
    char thisChar = Serial.read();  // Читання байту через послідовний інтерфейс
    if (startParse) {               // Якщо парсинг розпочато
      if (!isDigit(thisChar)) {     // Якщо поточний символ не є цифрою
        switch (header) {           // Обробляємо команди відповідно до їх заголовка
          case 'm':                 // Рух в позитивному напрямку
            mainServoDegree += intValue;
            if (mainServoDegree > M_MAX) mainServoDegree = M_MAX;
            mainServo.setTargetDeg(mainServoDegree);
            break;
          case 'M':                 // Рух в негативному напрямку
            mainServoDegree -= intValue;
            if (mainServoDegree < M_MIN) mainServoDegree = M_MIN;
            mainServo.setTargetDeg(mainServoDegree);
            break;
          case 'g':
            gripperServoDegree += intValue;
            if (gripperServoDegree > G_MAX) gripperServoDegree = G_MAX;
            gripperServo.setTargetDeg(gripperServoDegree);
            break;
          case 'G':
            gripperServoDegree -= intValue;
```

```
            if (gripperServoDegree < G_MIN) gripperServoDegree = G_MIN;
            gripperServo.setTargetDeg(gripperServoDegree);
            break;
          case 'l':
            leftServoDegree += intValue;
            if (leftServoDegree > L_MAX) leftServoDegree = L_MAX;
            leftServo.setTargetDeg(leftServoDegree);
            break;
          case 'L':
            leftServoDegree -= intValue;
            if (leftServoDegree < L_MIN) leftServoDegree = L_MIN;
            leftServo.setTargetDeg(leftServoDegree);
            break;
          case 'r':
            rightServoDegree += intValue;
            if (rightServoDegree > R_MAX) rightServoDegree = R_MAX;
            rightServo.setTargetDeg(rightServoDegree);
            break;
          case 'R':
            rightServoDegree -= intValue;
            if (rightServoDegree < R_MIN) rightServoDegree = R_MIN;
            rightServo.setTargetDeg(rightServoDegree);
            break;
          case 'X':                 // Встановлення початкового положення
            mainServoDegree = M_INIT;
            gripperServoDegree = G_INIT;
            leftServoDegree = L_INIT;
            rightServoDegree = R_INIT;

            mainServo.setTargetDeg(M_INIT);
            gripperServo.setTargetDeg(G_INIT);
            leftServo.setTargetDeg(L_INIT);
            rightServo.setTargetDeg(R_INIT);

            break;
        }
        // Serial.println("G: " + String(gripperServoDegree) + String("/") + G_MAX);
        recievedFlag = true;                          // Підтвердження отримання даних
        startParse = false;                           // Завершення парсингуі
      } else {                                        // Якщо отриманий символ є цифрою
        intValue = intValue * 10 + (thisChar - '0');  // Формуємо числове значення з отриманих
цифр
      }
    }
    if (isAlpha(thisChar) && !startParse) {  // Якщо отримано букву, але парсинг не розпочато
      header = thisChar;                     // Записуємо заголовок команди
      intValue = 0;                          // Обнуляємо значення числа
      startParse = true;                     // Розпочинаємо парсинг
      parseTime = millis();                  // Записуємо час початку парсингу
    }
  }
  if (startParse && (millis() - parseTime > TIMEOUT)) {
    startParse = false;                     // Завершення парсингу через таймаут
  }
}

void setup() {
  // Serial.begin(115200);
  Serial.begin(2000000);         // Ініціалізація послідовного інтерфейсу
  Serial.setTimeout(50);         // Встановлення таймауту для послідового інтерфейсу

  // Підключення сервоприводів до відповідних пінів та встановлення початкого положення
  gripperServo.attach(GRIPPER_SERVO_PIN, G_INIT);
  // gripperServo.smoothStart();
  gripperServo.setSpeed(180);     // Встановлення швидкості руху сервоприводу
  gripperServo.setAccel(0.5);     // Встановлення прискорення руху сервоприводу
  gripperServo.setAutoDetach(false);  // Вимкнення автоматичного відключення сервоприводу

  mainServo.attach(MAIN_SERVO_PIN, M_INIT);
  // mainServo.smoothStart();
  mainServo.setSpeed(180);
  mainServo.setAccel(0.5);
  mainServo.setAutoDetach(false);

  leftServo.attach(LEFT_SERVO_PIN, L_INIT);
  // leftServo.smoothStart();
  leftServo.setSpeed(180);
  leftServo.setAccel(0.5);
  leftServo.setAutoDetach(false);

  rightServo.attach(RIGHT_SERVO_PIN, R_INIT);
```

```
  // rightServo.smoothStart();
  rightServo.setSpeed(180);
  rightServo.setAccel(0.5);
  rightServo.setAutoDetach(false);
}

void loop() {
  parsing();      // Виклик функції для парсингу отриманих даних

  // Обробка руху сервоприводів
  mainServo.tick();
  gripperServo.tick();
  leftServo.tick();
  rightServo.tick();
}
```

2 Файл main.ino

```
#include <WiFi.h>     // Бібліотека для роботи з Wi-Fi
#include <WiFiClientSecure.h>   // Бібліотека для безпечного клієнтського з'єднання Wi-Fi
#include <ArduinoWebsockets.h>  // Бібліотека для роботи з WebSockets

const char* ssid = "MEIZU M6 Note";   // Назва Wi-Fi мережі
const char* password = "12345678";   // Пароль Wi-Fi мережі

using namespace websockets;    // Використання простору імен для бібліотеки ArduinoWebsockets

WebsocketsServer server;      // Створення об'єкту WebsocketsServer

void setup() {
  Serial.begin(115200);       // Ініціалізація з'єднання з монітором швидкістю 115200
  Serial1.begin(2000000, SERIAL_8N1, 13, 12); // Ініціалізація з'єднання з ESP32 та
налаштуваннями для UART
  Serial.setDebugOutput(true);  // Ввімкнення відлакдки
  Serial.println();

  WiFi.begin(ssid, password);   // Підключення до Wi-Fi

  while (WiFi.status() != WL_CONNECTED) {   // Очікування встановлення з'єднання з Wi-Fi мережею
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");   // Повідомлення про успішне підключення до Wi-Fi

  server.listen(82);             // Створення серверу WebSockets на порті 82
  Serial.print("Is server live? ");
  Serial.println(server.available()); // Перевірка доступності серверу

  Serial.print("Camera Ready! Use 'http://");
  Serial.print(WiFi.localIP());     // Виведення IP-адреси для з'єднання
  Serial.println("' to connect");
}

void loop() {
    WebsocketsClient client = server.accept();  // Прийняття клієнтського з'єднання
    while(client.available()) {        // Перевірка доступності даних від клієнта
      WebsocketsMessage msg = client.readBlocking();    // Зчитування блокуючого повідомлення від
клієнта

      // log
    //  Serial.print("Got Message: ");
    //  Serial.println(msg.data());

      Serial1.flush();        // Очищення буфера обміну для передачі даних
      Serial1.println(msg.data());    // Надсилання отриманого повідомлення
    }

}
```