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 Казакова Богдана Юрійовича
(ПІБ)

академічної групи 123-20-1
(шифр)

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

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

на тему «Кіберфізична система моніторингу температури розумного будинку за допомогою AWS IoT»

(назва за наказом ректора)

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

Рецензент				

Нормоконтролер	проф. Цвіркун Л.І.			

Дніпро
2024

ЗАТВЕРДЖЕНО:

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

_____ Гнатушенко В.В.
(підпис) (прізвище, ініціали)
« » _____ 2024 року

ЗАВДАННЯ
на кваліфікаційну роботу
ступеня бакалавр

студента Казаков Б.Ю. **академічної групи** 123-20-1
(прізвище та ініціали) (шифр)

спеціальності 123 Комп'ютерна інженерія

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

на тему «Кіберфізична система моніторингу температури розумного будинку за допомогою AWS IoT»

затверджену наказом ректора НТУ «Дніпровська політехніка» від 23.05.2024 № 469-с

Розділ	**Зміст**	**Термін виконання**
Стан питання та постановка завдання	На основі матеріалів виробничих практик, інших науково-технічних джерел розглянути призначення та завдання мікроконтролера у моніторингу температури в розумному будинку	05.05.2024
Розробка апаратної частини	Розробити вимоги до функцій, виконуваними системою, розробити структурну схему та специфікацію обладнання.	12.05.2024
Розробка корпоративної мереж	Побудувати в Packet Tracer модель мережі розумного будинку, виконати налаштування та перевірку роботи системи.	30.05.2024
Розробка компонента системи	Реалізувати та налаштувати систему моніторингу температури в розумному будинку з використаням платформи AWS IoT	20.06.2024

Завдання видано _____ доц. Бешта Д.О.
(підпис керівника) (прізвище, ініціали)

Дата видачі 06.02.2024

Дата подання до екзаменаційної комісії 01.07.2024

Прийнято до виконання _____ Казаков Б.Ю.
(підпис студента) (прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 78 с., 57 рис., 5 табл., 1 дод., 24 джерел.

КОНТРОЛЕР, IoT, ВІДСЛІДКОВУВАННЯ ТЕМПЕРАТУРИ, AWS, IoT CORE, RASPBERRY, DS18B20

Об’єкт розробки – процес передачі даних з мікроконтролера Raspberry Pi3 на платформу AWS IoT.

Мета роботи – створення та впровадження кіберфізичної системи для моніторингу температури в розумному будинку з використаням платформи AWS IoT.

Здійснено розробку IoT системи, що включає датчики температури, центральний хаб для обробки даних, а також інтеграцію з AWS IoT для збору, аналізу та зберігання даних. Система орієнтована на застосування в приватних будинках та офісних приміщеннях для забезпечення комфортної температури в приміщенні

Розробка комп’ютерної мережі виконана відповідно до завдання на кваліфікаційну роботу бакалавра.

Робота системи перевірена за допомогою моделі схеми корпоративної мережі із застосуванням програми Cisco Packet Tracer.

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

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАК, ОДИНИЦЬ І ТЕРМІНІВ

IoT – Internet of Things (Інтернет речей)

AWS IoT – Amazon Web Services Internet of Things (Інтернет речей Amazon Web Services)

GPIO – General Purpose Input/Output (загальний вхід/вихід)

TLS – Transport Layer Security (безпека транспортного рівня)

MQTT – Message Queuing Telemetry Transport (транспорт телеметрії з чергами повідомлень)

WPA – Wi-Fi Protected Access (специфікація шифрування даних для бездротової локальної мережі)

SBC – Subband Codec (стандартна технологія кодування аудіо, використовувана пристроями BLUETOOTH)

HTTP – HyperText Transfer Protocol (протокол передачі гіпертексту)

AI – artificial intelligence (штучний інтелект)

ВСТУП

Кіберфізична система моніторингу температури розумного будинку є невід'ємною частиною сучасних інтелектуальних систем керування житлом. Вона спрямована на забезпечення стабільного та комфортного температурного режиму у приміщеннях шляхом об'єднання фізичних пристроїв, таких як датчики температури з цифровими технологіями, включаючи алгоритми машинного навчання та інтернет речей (IoT). Датчики, розміщені у різних частинах будинку та на подвір’ї, постійно збирають дані про поточний рівень температури. Ці дані передаються у реальному часі на хмарну середовище, де їх обробляють, аналізують та візуалізують. Система використовує ці дані для виявлення відхилень від заданих параметрів та прийняття рішень щодо коригування.

Таким чином, система створює можливість не тільки автоматизованого підтримання оптимальних кліматичних умов, але й сприяє енергоефективності та зниженню витрат на енергоспоживання. Використання алгоритмів машинного навчання дозволяє системі адаптуватися до індивідуальних потреб мешканців, передбачати можливі зміни у кліматичних умовах та забезпечувати більш точне регулювання температури. Крім того, інтеграція з інтернетом речей забезпечує можливість віддаленого контролю та керування системою через мобільні пристрої або веб-інтерфейси, що додає додаткового комфорту та гнучкості в управлінні домашнім середовищем.

Важливим аспектом є безпека та захищеність даних, що передаються та зберігаються в системі. Забезпечення належного рівня кібербезпеки є критично важливим аспектом для запобігання несанкціонованого доступу та потенційних загроз для особистої інформації користувачів. Надійні методи шифрування та автентифікації допомагають підтримувати конфіденційність і цілісність даних, а також забезпечують захист від можливих атак.

1 СТАН ПИТАННЯ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1 Опис технології “Розумний будинок”

Система керування "розумний будинок" (англ. *smart home*) являє собою комплексну інтеграцію різних цифрових технологій, пристроїв і програмного забезпечення, що забезпечує автоматизоване та централізоване керування різними аспектами домашнього середовища. Основною метою системи розумного будинку є підвищення комфорту, безпеки та ефективності в управлінні житлом.

Основні компоненти системи керування розумним будинком включають:

а) Центральний контролер. Центральний пристрій або програмне забезпечення, яке координує роботу всіх підключених пристроїв та систем. Він виступає як «мозок» системи розумного будинку, забезпечуючи централізований контроль та управління.

б) Підключені пристрої. До системи розумного будинку підключаються різні пристрої, включаючи датчики, розумні термостати, освітлювальні прилади, замки, камери безпеки, побутову техніку та інші електронні пристрої. Ці пристрої можуть взаємодіяти один з одним та з центральним контролером.

в) Інтерфейс користувача. Це засіб взаємодії користувача з системою, зазвичай представлений мобільним додатком або веб-інтерфейсом. За допомогою інтерфейсу користувач може контролювати всі аспекти розумного будинку, встановлювати параметри та переглядати інформацію.

г) Автоматизація та сценарії. Система розумного будинку дозволяє налаштувати автоматичні сценарії та правила, що визначають, як повинні працювати підключені пристрої в залежності від часу доби, присутності людей або інших факторів. Наприклад, світло може автоматично вмикатися та вимикатися в залежності від розкладу чи руху мешканців.

д) Зв'язок та мережа. Підключення пристроїв та систем один з одним здійснюється за допомогою бездротових технологій, таких як Wi-Fi, Zigbee, або інших протоколів зв'язку. Це забезпечує стабільний обмін даними між

пристроями та центральним контролером.

е) Безпека та конфіденційність. Системи розумного будинку зазвичай включають функції безпеки, такі як шифрування даних, автентифікація користувачів та контроль доступу. Це допомагає забезпечити безпечне використання системи та захист конфіденційної інформації.

Рисунок 1.1 – Зображення систем використаних в “Розумний будинок”

Завдяки системі керування розумним будинком, мешканці можуть автоматизувати рутинні завдання, керувати різними пристроями з одного місця, та підвищувати ефективність використання енергії та ресурсів.

1.2 Опис технології IoT та її інтеграція в систему «Розумний будинок»

Термін IoT, або Інтернет речей, відноситься до колективної мережі підключених пристроїв та технології, що полегшує зв'язок між пристроями та хмарою, а також між самими пристроями. Завдяки появі недорогих комп'ютерних мікросхем та телекомунікацій з високою пропускною спроможністю ми маємо тепер мільярди пристроїв, підключених до Інтернету. Це означає, що повсякденні пристрої, такі як зубні щітки, пилососи, автомобілі та механічні установки – можуть використовувати датчики для збору даних та розумного реагування на дії користувачів.

Інтернет речей поєднує повсякденні «речі» з Інтернетом. Комп'ютерні інженери додають датчики та процесори до повсякденних предметів з 90-х років. Однак спочатку прогрес був повільним, тому що мікросхеми були більшими та громіздкими. Комп'ютерні чіпи малої потужності, які називають RFID-мітками, вперше використовувалися для відстеження дорогого обладнання. У міру того, як обчислювальні пристрої зменшувалися в розмірах, ці чіпи також з часом ставали менше, швидше і розумніше.

Вартість інтеграції обчислювальної потужності до невеликих об'єктів тепер значно знизилася. Наприклад, можна додати можливість підключення голосових сервісів до мікроконтролерів з вбудованою оперативною пам'яттю менше 1 МБ, наприклад для вимикачів світла. Виникла ціла індустрія, спрямована на наповнення наших будинків, підприємств та офісів пристроями IoT. Ці смарт-об'єкти можуть автоматично передавати дані до Інтернету та з Інтернету. Всі ці «невидимі обчислювальні пристрої» та пов'язані з ними технології разом називаються Інтернетом речей.

Пристроєм IoT або парком пристроїв можна керувати через графічний інтерфейс користувача. Загальні приклади включають мобільний додаток або веб-сайт, які можна використовувати для реєстрації та керування смарт-пристроями.

Пристрої «розумного» будинку в основному орієнтовані на підвищення ефективності та безпеки будинку, а також на покращення домашніх мереж. Такі пристрої, як розумні розетки, контролюють споживання електроенергії, а інтелектуальні термостати забезпечують підвищений контроль температури. Гідропонні системи можуть використовувати датчики IoT для керування садом, а датчики диму IoT можуть виявляти тютюновий дим. Домашні системи безпеки, такі як дверні замки, камери відеоспостереження та детектори витоку води, можуть виявляти та запобігати загрозам, а також надсилати попередження домовласникам.

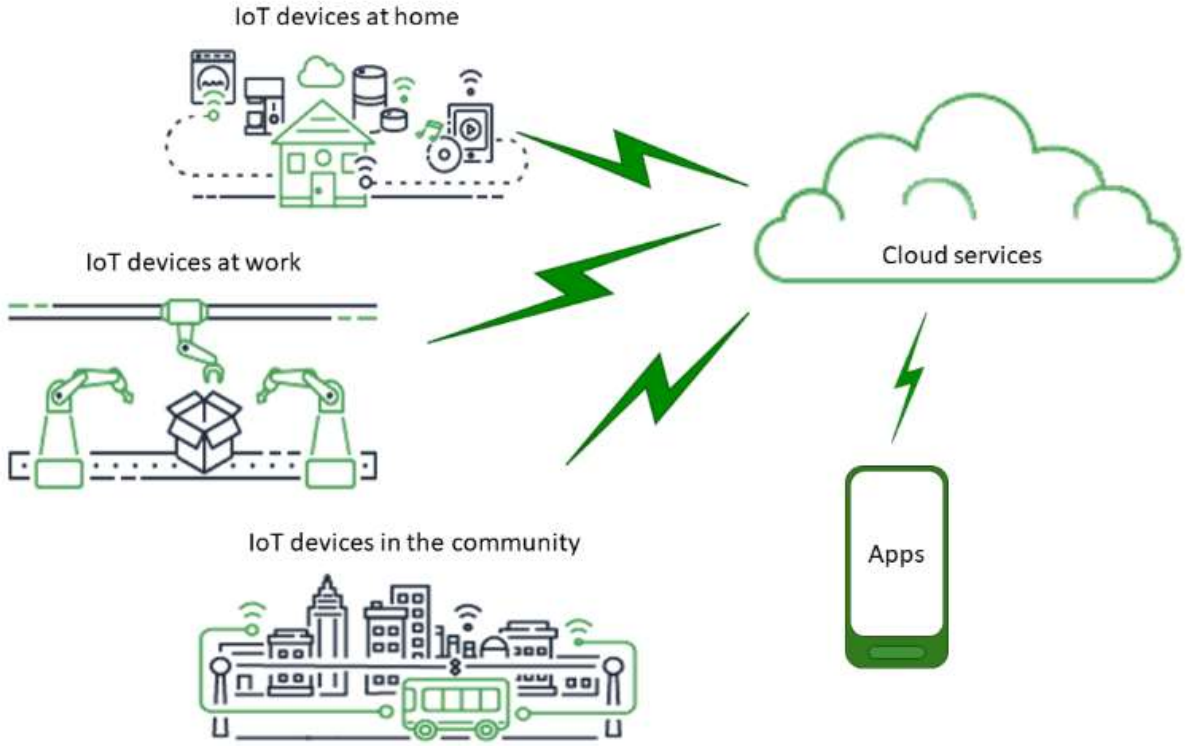

Рисунок 1.2 – Графічне зображення IoT

1.3 Опис умов використання кіберфізичних систем у «розумному будинку»

Компанія «SmartHouse», яка спеціалізується на розробці та впровадженні технологій розумного будинку, забезпечує інтеграцію різноманітних систем та пристроїв для створення автоматизованого і комфортного життєвого середовища. Впровадження кіберфізичної системи для забезпечення моніторингу температури в розумному домі дозволяє вирішити низку ключових завдань:

– забезпечення комфортного клімату: постійний моніторинг температури в різних приміщеннях будинку дозволяє підтримувати оптимальні умови для життя;

– віддалений доступ і управління: можливість віддаленого моніторингу температури через мобільні додатки або інтернет дозволяє мешканцям і компаніям швидко реагувати на будь-які зміни. Це забезпечує оперативне керування температурними параметрами і можливість їх налаштування в реальному часі.

Для використання кіберфізичної системи з контролю температури в розумних будинках необхідно враховувати наступні аспекти:

– пристосування до змін зовнішньої температури: зовнішні температури можуть значно варіюватися протягом доби та в різні пори року. Кіберфізична система повинна мати здатність адаптуватися до цих коливань, зберігаючи стабільний температурний режим у приміщенні. Це досягається шляхом аналізу зовнішньої температури та врахування факторів, таких як напрямок вітру, сонячна радіація та інші погодні умови;

– безпека і захист системи: система моніторингу температури повинна бути надійно захищена від несанкціонованого доступу та кібератак. Це включає в себе забезпечення безпеки даних користувачів і підтримку стабільної роботи всіх датчиків і пристроїв, які контролюють температуру.

1.4 Характеристика та структура керуючої компанії

Головний офіс компанії SmartHouse, яка є розробником «розумного дому», розташований у центральній частині міста Дніпро, зручної для співробітників та клієнтів і складається з декількох підрозділів. Інноваційні офісні приміщення з відкритим простором побудовані для сприяння комунікації та співпраці між відділами.

Організаційна структура має дивізіонну форму. Головною перевагою такої структури є незалежність операційної діяльності, тобто проблеми однієї компанії не загрожують існуванню інших.

Структура компанії включає в себе відділи: розробки, інтеграції та впровадження, продажів та маркетингу, досліджень та розвитку.

Відділ розробки (5 чол.):

– відповідає за створення та вдосконалення програмного забезпечення для розумних пристроїв та систем;

– складається з команд програмістів, архітекторів програмного забезпечення та інженерів з вбудованими системами.

Відділ інтеграції та впровадження (10 чол.):

– забезпечує встановлення та налаштування розумних систем в будинках клієнтів;

– включає в себе інженерів-інсталяторів, технічних консультантів та техніків з підтримки клієнтів.

Відділ продажів та маркетингу (2 чол.):

– відповідає за залучення нових клієнтів, рекламні кампанії та підтримку відносин з клієнтами;

– складається з менеджерів з продажу, маркетологів та спеціалістів зі зв'язків з громадськістю.

Відділ досліджень та розвитку (2 чол.):

– займається пошуком нових технологій та інновацій у сфері розумних будинків;

– складається з дослідників, аналітиків та інженерів-розробників.

Відділ технічної підтримки (8 чол.):

– надає підтримку клієнтам щодо вирішення технічних проблем та запитань після продажу;

– складається з клієнтських менеджерів та технічних спеціалістів.

Також маємо центр інновацій (5 чоло.), у якому проводяться демонстрації нових продуктів та технологій, а також тренінги для партнерів та клієнтів, він розташований у складі головного офісу.

Така структура дозволяє компанії ефективно працювати над розробкою, впровадженням та підтримкою продуктів для розумних будинків, забезпечуючи високий рівень сервісу для клієнтів.

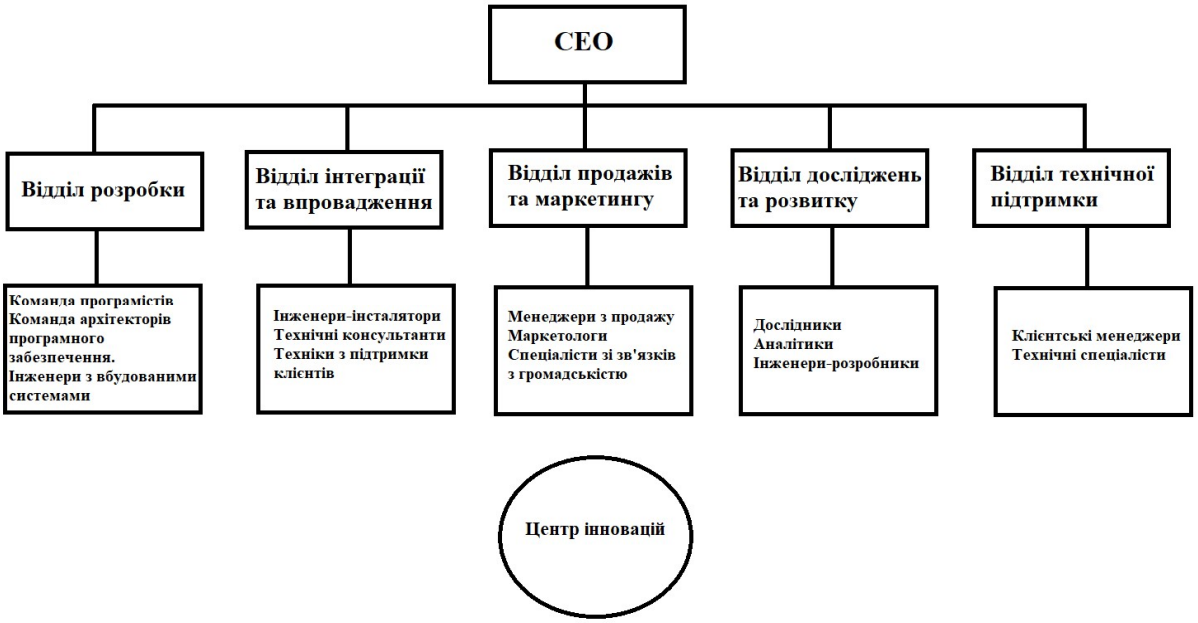

Рисунок 1.3 – Схема організаційної структури комплексу компанії

Ступінь комп'ютеризації робочих місць компанії приведена в табл.1.1. Загалом в компанії 37 комп'ютеризованих робочих місць

Таблиця 1.1 – Комп'ютеризовані робочі місця підприємства

Структурний підрозділ	Кількість		
	Працівники	ПК	Принтери
CEO	2	2	1
Відділ розробки	10	10	1
Відділ інтеграції та впровадження	10	9	1
Відділ продажів та маркетингу	2	2	1
Відділ досліджень та розвитку	2	2	-
Відділ технічної підтримки	8	4	1
Центр інновацій	5	2	
Загалом ПК		31	6

1.5 Характеристика типових житлових приміщень

Проект розумного будинку зображений на рисунку 1.4. Це одноповерховий будинок з терасою, який матиме 4 кімнати та буде знаходитись біля м. Дніпро у лісистій місцевості на лівобережжі. Такий вибір місця розташування забезпечить

мешканцям затишок і сприятливе середовище, далеко від міського шуму і забруднення, водночас забезпечуючи гарний доступ до міста.

Будинок спроектований таким чином, щоб максимально відповідати концепції розумного будинку. Всі кімнати з'єднані в єдину систему, яка дозволяє керувати внутрішніми процесами за допомогою центрального контролера. Контролер, до якого під’єднуються датчики температури, знаходитиметься у вітальній кімнаті. Блок живлення буде розміщено в коридорі біля входу в дім. Це розташування вибрано для зручного керування та моніторингу всієї системи.

У будинку передбачено використання декількох датчиків температури для забезпечення комфортного клімату в усіх приміщеннях. Максимальна довжина проводів датчиків становить 10 метрів, що накладає певні обмеження на їхнє розміщення. Тому, щоб забезпечити ефективний моніторинг температури в усьому будинку, було прийнято рішення розташувати датчики наступним чином:

– датчик на терасі, він буде вимірювати температуру навколишнього середовища. Це важливо для забезпечення точних даних про зовнішню температуру, що дозволяє адаптувати внутрішні кліматичні умови будинку відповідно до змін погоди;

– датчик у вітальній кімнаті, оскільки вона є найбільшою кімнатою у будинку і основним місцем перебування мешканців, цей датчик дозволить підтримувати оптимальну температуру саме в цій зоні, забезпечуючи максимальний комфорт.

Типове житлове приміщення зображено на рисунку 1.4.

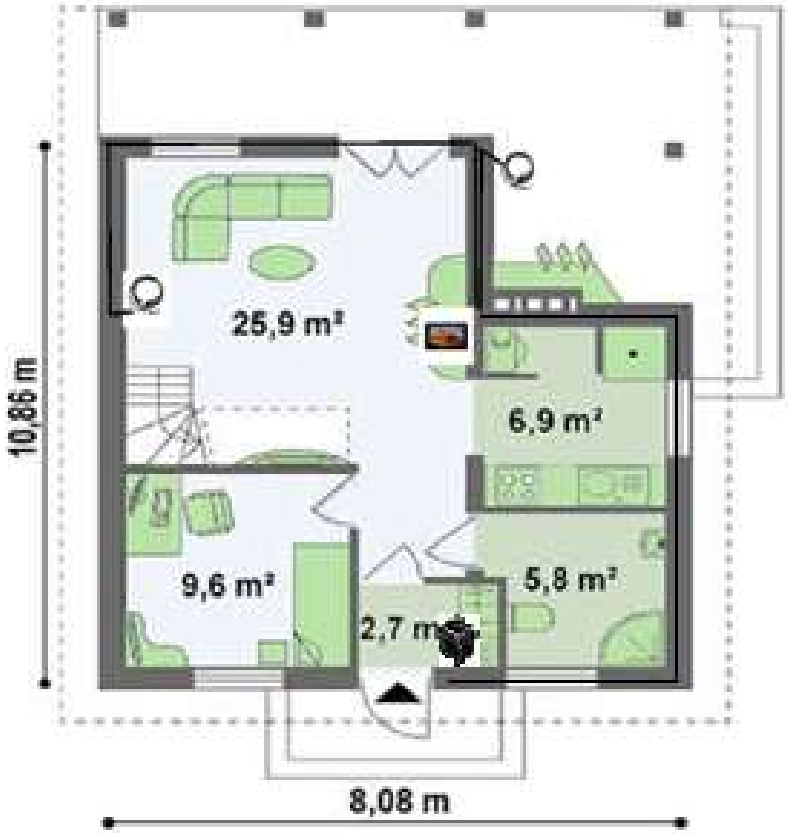

Рисунок 1.4 – Топологічна схема розміщення датчиків в житловому приміщенні

Окрім основної функції контролю температури, система «розумний будинок» дозволяє інтегрувати інші пристрої та сенсори, що забезпечують безпеку, освітлення, вентиляцію та інші функції.

Однією з ключових переваг розумного будинку є можливість оптимізації споживання енергії. Система контролю температури, вбудована у проект, сприятиме зменшенню витрат на опалення і кондиціонування повітря. Використання сучасних технологій та енергоефективних матеріалів при будівництві дозволить мінімізувати тепловтрати та забезпечить більш економне використання ресурсів.

1.6 Визначення можливих напрямків рішення поставлених завдань

Для вирішення проблеми моніторингу температури є багато варіацій. Це і апаратні рішення, по типу цифрових термометрів, систем моніторингу температури, тепловізорів та багато чого іншого. Але, окрім апаратних рішень

для моніторингу температури, існує ще безліч хмарних сервісів, які пропонують аналогічні функції. Dapatlicity, Monnit, SenseThink SenseCAP, Losant та AWS IoT. Вони мають багато переваг над апаратними рішеннями, адже вони економічніші, простіші у використанні та обслуговуванні, мають змогу у масштабуванні та гнучкості, а також мають підвищену безпеку. Тому зупинимось саме на них.

При виборі хмарного сервісу для моніторингу температури слід враховувати:

– функції, які нам потрібні: деякі сервіси пропонують базові функції моніторингу, а інші – більш розширені функції, такі як аналіз даних та звітність;

– вартість: деякі послуги пропонують безкоштовні плани, а інші – платні передплати;

– простота використання: деякі сервіси прості у використанні, інші потребують технічних знань;

– безпека: переконаймося, що вибраний сервіс використовує надійні заходи безпеки для захисту ваших даних.

Виходячи з усіх вище перерахованих аспектів, зупинимось на хмарному сервісі AWS IoT від Amazon. AWS IoT Core – це повністю керована платформа хмарних служб, призначена для підключення, керування та масштабування IoT-пристроїв. Вона пропонує широкий спектр функцій, які допомагають збирати, аналізувати та використовувати дані з датчиків, розташованих на мільйонах пристроїв по всьому світу.

Основні можливості AWS IoT Core:

– підтримує широкий спектр протоколів підключення, включаючи MQTT, HTTP, WebSockets і LoRaWAN. Це дає можливість підключати до платформи практично будь-який тип IoT-пристрою;

– пропонує багаторівневу систему безпеки, яка допомагає захистити ваші IoT-пристрої та дані. До цих функцій належать аутентифікація та авторизація пристроїв, шифрування даних, контроль доступу та моніторинг безпеки;

– надає широкий спектр функцій для управління IoT-пристроями, включаючи оновлення прошивки, віддалене керування та налаштування;

– пропонує інструменти для аналітики, які допомагають підприємствам отримувати insights з даних IoT. До цих інструментів належать Amazon Kinesis, Amazon S3, Amazon Redshift та Amazon Machine Learning;

– може масштабуватися, щоб підтримувати мільйони IoT-пристроїв.

Переваги використання AWS IoT Core:

– може допомогти знизити витрати на розгортання та експлуатацію IoT-рішень;

– пропонує широкий спектр функцій, які дають гнучкість у розробці та розгортанні IoT-рішень;

– може допомогти швидше виводити на ринок IoT-продукти та послуги;

– дає можливість розробляти інноваційні IoT-рішення, які допоможуть отримати конкурентну перевагу.

Для вирішення поставленого завдання одного хмарного середовища буде недостатньо. Є потреба в контролері.

Контролер – це електронний пристрій, який використовується для керування іншими пристроями або системами. Він може приймати сигнали від датчиків або користувачів, обробляти їх та надсилати команди іншим пристроям. Він нам допоможе отримувати дані з датчиків та передавати їх на хмарне середовище для подальшої візуалізації та аналізу. Їх різноманітність допоможе нам обрати найефективніший та підходящий для подальшої роботи.

1.6.1 Порівняння контролерів Banana Pi BPI-M5 та Raspberry Pi 4

Зупинимось на двох варіантах, які на даний час є популярними з точки зору технічних характеристик: Banana Pi BPI-M5 та Raspberry Pi 4. Це два популярні одноплатні комп'ютери (SBC), які часто використовуються для проектів "розумного будинку", робототехніки та інших вбудованих систем. Обидва контролери мають схожі характеристики, але й деякі ключові відмінності.

Контролер Banana Pi BPI-M5 має процесор Cortex-A55, а Raspberry Pi – Cortex-A72, у обох по 4 ядра, однакова архітектура та різна тактова частота, а також ті ж 4 ГБ оперативної пам'яті, але Banana Pi одну величезну перевагу –

16 ГБ вбудованої пам'яті eMMC (вбудована мультимедійна картка пам'яті). (див. рис. 1.5 – 1.6).

Рисунок 1.5 – Контролер Banana Pi BPI-M5

Подібно Raspberry Pi, на цій мініатюрній платі реалізовано досить багато функціоналу. Однак йому не вистачає вбудованих Wi-Fi та Bluetooth, що для нас може стати реальною проблемою. Тим не менш, 16 ГБ вбудованої пам'яті, джерело живлення USB-C та звичний 40-контактний GPIO підійде для тих, хто хоче запускати фізичні обчислювальні проекти, не покладаючись на карти SD для запуску операційної системи.

Тому порівнюючи ці два майже однакові контролери за ціною та функціональними показниками, ми можемо зробити висновок, що наш вибір зупинився на використанні саме Raspberry Pi через його функціонал та додаткові функції Bluetooth та Wi-fi , адже саме вони допомогають нам у нашій системі реалізовувати зв’язок з AWS IoT. Що стосується недостачі пам’яті у Raspberry у зрівнянні з конкурентним контролером, то її цілком вистачає для поставленної задачі.

Рисунок 1.6 – Контроллер Rasbperry Pi 4

Саме тому наш вибор пав на контролер Rasbperry Pi4 та хмарне середовище AWS IoT, які цілком задовольняють наші потреби та є оптимальним варіантам у

вирішенні поставленої задачі.

1.7 Постановка завдання

Завданням для кваліфікаційної роботи є розробка кіберфізичної системи моніторингу температури розумного будинку за допомогою AWS IoT на базі контроллера Raspberry Pi 4

Для вирішення цього завдання необхідно виконати наступні пункти:

– ознайомлення зі структурою роботи «Розумного будинку»;

– формулювання технічних вимог для контролера Raspberry Pi 4;

– вибір датчика для контролю температури;

– розробка системи для контролю заданого параметру;

– розробка програмного забезпечення для моніторингу температури;

– розробка програмних рішень для збору, обробки та візуалізації інформації в AWS.

Результатом виконання кваліфікаційної роботи є мережа «Розумного будинку» з підсистемою моніторингу температури температури будинку, та є гнучкою, надійною, масштабованою та безпечною.

2 РОЗРОБКА АПАРАТНОЇ ЧАСТИНИ КІБЕРФІЗИЧНОЇ СИСТЕМИ

2.1 Технічні вимоги до кіберфізичної системи

2.1.1 Вимоги до системи в цілому

2.1.1.1 Вимоги до структури і функціонуванню системи

2.1.1.1.1 Перелік підсистем, їхнє призначення й основні характеристики

Система повинна бути реалізована у розумному будинку для контролю усіх необхідних параметрів для життєдіяльності і являти собою готову систему для виміру та контролю цих параметрів.

Кіберфізична система моніторингу температури у «розумному будинку» має складатись з наступних підсистем:

– підсистема збору даних в «розумному будинку», яка відповідає за збір інформації про температуру за допомогою датчиків. Система використовує два датчики, які передають дані з певною періодичністю, встановленою у програмному коді на Python. Підсистема керування, завдання якої є обробка даних, отриманих від датчиків. Вона реалізована на базі мікроконтролера Raspberry Pi, який виконує управлінські алгоритми, зберігає налаштування системи та історію зібраних даних. Крім того, ця підсистема забезпечує можливість віддаленого доступу до системи;

– підсистема хмарного серверу: зібрані дані відправляються на хмарну платформу AWS для подальшої візуалізації та надсилаються через Wi-Fi на пристрій, який підключений до цього сервісу.

– підсистема локальної мережі керуючої компанії, структура якої повинна складатися з підмереж згідно організаційної структури. Для забезпечення надійності та майбутнього розширення мережі, рекомендується мати приблизно 10% запасних портів на випадок відмови портів або для майбутнього розширення та вдосконалення мережі.

2.1.1.1.2 Вимоги до числа рівнів ієрархії та ступені централізації

Підсистема збору даних повинна мати дворівневу ієрархію та складатись з нижнього рівня: блок збору даних, та верхнього – центральний блок керування (мікроконтролер) та web-інтерфейс.

Підситема хмарного серверу повинна підтримувати архітектуру AWS IoT з трьома основними рівнями: рівень пристроїв, рівень шлюзів та рівень хмарних служб.

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

Кіберфізична система повинна входити в склад локальної мережі розумного будинку. В кіберфізичній системі зв'язок між блоком збору даних і центральним блоком керування відбувається через бездротове з'єднання Wi-Fi. Дані з датчиків температури до мікроконтролера передаються по інтерфейсу 1-Wire. Протокол MQTT для обміну повідомленнями між IoT-пристроями.

HTTP/HTTPS для передачі даних через Інтернет на AWS з використанням стандартних веб-протоколів.

В мережі керуючої компанії для побудови локальної мережі використовувати дротове (FastErhernet або GigabitEthernet) або бездротове з'єднання Wi-Fi стандарту 802.11n.

Використання витої пари (Cat5e або Cat6) для підключення пристроїв у внутрішній мережі основної та віддаленої частин будинку. Для прокладання маршрутів між різними підмережами використовувати протокол маршрутизації OSPF.

2.1.1.1.4 Вимоги до режимів функціонування кіберфізичної системи

Кіберфізична система моніторингу температури повинна функціонувати в режимі моніторингу. Система автоматично кожні 3 с. повинна збирати дані про температуру за допомогою датчиків температури і передавати цю інформацію до контролера.

Користувач повинен мати можливість переглядати поточні значення температури та історію даних через на платформі AWS або через графіки візуалізації.

2.1.1.1.5 Вимоги до діагностування кіберфізичної системи

Кіберфізична система моніторингу температури повинна включати в себе вбудовані засоби діагностики, якими вдасться виявити та виправляти несправності, а також передавати інформацію про поточний стан системи. Вимоги до діагностування включають в себе:

– моніторинг стану компонентів: система повинна постійно відстежувати стан датчиків температури та інших компонентів. Інформація про стан системи повинна бути доступна для оперативного перегляду користувачем;

– діагностика несправностей: **с**истема має вбудовані засоби самодіагностики для виявлення потенційних несправностей датчиків та інших компонентів. Вона автоматично генерує повідомлення про виявлені проблеми та пропонує відповідні заходи для їх усунення;

– контроль інтеграції: система забезпечує контроль інтеграції всіх компонентів, включаючи зв'язок між датчиками, центральним контролером та іншими частинами системи. Цей контроль дозволяє забезпечити надійну і стабільну роботу системи моніторингу температури.

2.1.1.1.6 Перспективи розвитку, модернізації кіберфізичної системи

У процесі розвитку важливою є питання ефективності та надійності мережевої інфраструктури. Одним із ключових напрямків у цьому аспекті є модернізація мережі керуючої компанії, що полягає у розширенні її функціоналу. Для досягнення цієї мети розглянути можливість додання чотирьох підмереж нових філіалів керуючої компанії та їх з'єднання через множинні маршрутизатори та налаштування маршрутизації.

Для моделювання мережі компанії було обрано топологію на рисунку 2.1. Мережа LAN1 (далі MANCOMP1) є мережею керуючої компанії, інші 4 мережі

представляють можливі філіали в інших містах та додані для виконання учбового завдання з реалізації маршрутизації між підмережами. Кожна з підмереж повинна мати можливість доступу до Інтернет через NAT.

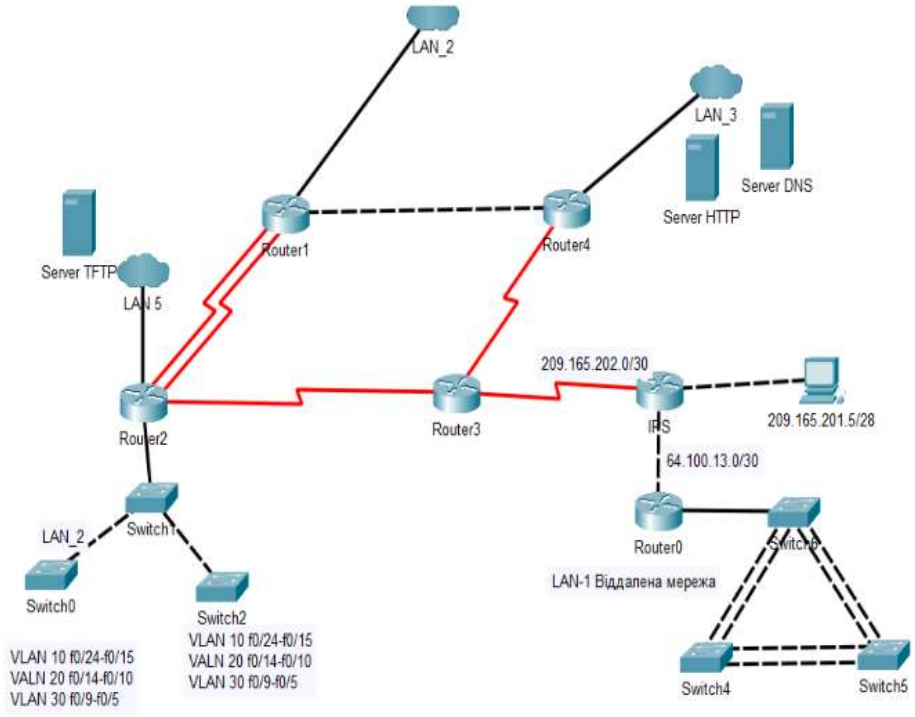

Рисунок 2.1 – Топологія мереж керуючої компанії

Для ефективного функціонування мережі важливо правильно налаштувати маршрутизатори, щоб забезпечити оптимальний шлях для передачі даних між підмережами та зовнішніми мережами.

Перспективи розитку та модернізації кіберфізичної системи:

– збільшення точності та чутливості. Вдосконалення сенсорів для моніторингу температури дозволить отримувати більш точні дані. Використання нових матеріалів і технологій може покращити чутливість та стабільність вимірювань;

– застосування штучного інтелекту і аналітики даних. Використання AI для аналізу даних про температуру може дозволити прогнозування змін, виявлення аномалій та автоматизацію регулювання системи опалення та кондиціонування;

– інтеграція з IoT і смарт-технологіями. Зв'язок з іншими смарт-пристроями у домі (освітлення, системи безпеки, побутові прилади) для автоматичного реагування на зміни температури і створення комфортних умов

для життя;

– енергоефективність і екологічність. Розробка систем, що мінімізують енергоспоживання, наприклад, за допомогою оптимального регулювання температури на основі зовнішніх умов і попередніх вимірів;

– захист від кіберзагроз. Зростаюча важливість забезпечення кібербезпеки для запобігання несанкціонованому доступу до системи моніторингу та її даних;

– модульність і гнучкість. Системи повинні бути модульними, щоб легко додавати нові функції або сенсори із зростанням потреб користувача;

– віддалений моніторинг і управління. Можливість віддаленого доступу через мобільні додатки для моніторингу та управління температурним режимом в будь-який час із будь-якого місця.

2.1.1.2 Вимоги до показників призначення

Система повинна забезпечувати зручний доступ користувачів до даних, збережених у кіберфізичних системах, а також можливість обміну цією інформацією. Це означає, що користувачі повинні мати змогу легко отримувати доступ до інформації про температуру у своєму будинку та ззовні через зручні інтерфейси.

Для забезпечення надійності і доступності даних, система має використовувати централізоване зберігання у хмарному середовищі. Це дозволить забезпечити надійне збереження і можливість відновлення важливої інформації у разі потреби.

З метою забезпечення безпеки даних, система повинна мати вбудовані заходи захисту від несанкціонованого доступу і шкідливого програмного забезпечення. Це включає в себе використання сучасних методів шифрування та аутентифікації для захисту інформації про температуру в будинку.

Також система має підтримувати можливість підключення до Інтернету і віддаленого доступу, щоб користувачі могли моніторити та управляти температурними умовами свого будинку з будь-якої точки світу через веб-інтерфейси або мобільні додатки.

2.1.1.3 Вимоги до патентної чистоти

Має бути забезпечена патентна чистота програмного та апаратного забезпечення на території України.

Програмне забезпечення для кіберфізичної системи мікроклімату розумного будинку, повинно бути розроблене для мікроконтролера Raspberry Pi, з використанням запатентованих технологій або комерційних алгоритмів.

Розроблене програмне забезпечення повинно бути оригінальним і не містити код або алгоритми, які порушують права інтелектуальної власності третіх осіб.

2.1.1.4 Додаткові вимоги

Щоб система моніторингу температури ефективно функціонувала, вона повинна враховувати умови експлуатації та характеристики обладнання.

Умови експлуатації:

– внутрішній блок має бути стійким до впливу різних чинників (перепадів температур, пилу, вологи, умовних непередбачених обставин) в житлових приміщеннях, а його можливий діапазон температур повинен сягати від -10°C до +50°C;

– зовнішній блок потребує захисту для нормальної роботи від навколишніх чинників (прямого сонячного проміння, опадів та вологи, забруднення), а його можливий діапазон температур має сягати від -30°C до +60°C.

Вимоги до обладнання:

– мікроконтролер повинен безперебійно функціонувати протягом тривалого часу без необхідності перезавантажень або обслуговування. Це гарантує точність та надійність моніторингу температурних показників;

– мікроконтролер повинен мати достатню кількість портів вводу/виводу для підключення всіх датчиків, виконавчих механізмів та можливого розширення системи в майбутньому. Це дозволяє гнучко адаптувати систему до мінливих потреб та інтегрувати додаткові функції;

– активне обладнання не повинно створювати перешкод для роботи інших пристроїв, particularly Wi-Fi 802.11n 2.4GHz. Це гарантує безперебійну роботу інших систем та запобігає втраті даних;

– мікроконтролер повинен знаходитися в зоні з стійким Wi-Fi сигналом. Це забезпечує безперебійну передачу даних про температуру з датчиків на хмарне середовище;

– використання стандартних протоколів та інтерфейсів для передачі даних полегшує інтеграцію системи з іншими системами та програмним забезпеченням;

– надійне та безпечне шифрування даних гарантує конфіденційність інформації та захищає її від несанкціонованого доступу.

2.1.2 Вимоги функцій, виконуваним системою

2.1.2.1 Підсистема збору даних

Перелік функцій, виконуваним системою:

– збір даних з датчиків температури;

– відправка даних у хмарне середовище через Wi-Fi;

– передача даних у систему візуалізації даних через Wi-Fi.

Часовий регламент і вимоги до якості:

– інтервал вимірювання датчиків: кожні 3 секунди;

– точність вимірювання температури: приблизно ±0.5°C;

– час передачі даних: у випадку бездротового з’єднання від декількох мілісекунд до секунди;

– формат даних: будь-який комфортний формат даних для обробки.

2.1.2.2 Підсистема керування

Перелік функцій підсистем керування:

– прийом та обробка даних з хмарного середовища збору даних;

– аналіз отриманих даних та порівняння їх з минулими;

– зберігання налаштувань системи та історії даних;

– забезпечення віддаленого доступу до системи через мережу Інтернет.

Часовий регламент та вимоги до якості:

– час реакції на зміну даних з датчиків: не більше 3 секунд;

– обсяг зберігання історії даних: не більше 1 місяця.

2.1.2.3 Підсистема локальної мережі керуючої компанії

Відповідно до розділу 2.1.1.1.6 та рисунку 2.1 мережу керуючої компанії необхідно розширити мережу шляхом додавання 4 підмереж та їх з'єднання через множинні маршрутизатори, та налаштувати маршрутизацію. Мережа компанії повинна мати єдиний простір IP-адресації 10.24.32.0/21. Адресація підмереж методом VLSM. Сегменти середовища (IP-підмережі) поділяються маршрутизаторами.

Кількість вузлів в певному сегменті мережі:

MANCOMP1 – 37;

MANCOMP2 – 79;

MANCOMP3 – 210;

MANCOMP4 – 78;

MANCOMP5 – 43.

Для виконання базового налаштування конфігурації пристроїв потрібно враховувати наступні вимоги [17]:

– назви пристроям за наступним правилом: Kozakov_тип пристрою_номер пристрою;

– на всіх пристроях повинен бути назначений пароль cisco до консолі і vty;

– на всіх пристроях повинен бути назначений пароль class до привілейованого режиму;

– усі паролі, що зберігаються у відкритому вигляді, потрібно зашифрувати;

– потрібно назначити на усіх лініях vty використання протоколу SSH;

– потрібно призначити ім’я користувача та пароль на всіх пристроях за правилом: 123201_Kazakov з паролем admincisco;

– в якості імені домена потрібно використати ім'я пристрою;

– для шифрування даних потрібно створювати ключ RSA завдовжки 1024 біт/.

В мережі компанії MANCOMP1 потрібно враховувати наступні вимоги:

– роутер в мережі повинен виконувати роль DHCP-сервера;

– застосувати технологію VLAN для розділення мережі на 2 равні сегменти: VLAN 20 (DEVELOPER) для відділу розробки, VLA

– N10 (OFFICE) для інших відділів;

– розподілити порти між VLAN за наступними правилами:

- f0/1-10: vlan10, access;
- f0/11-20: vlan20, access;
- f0/21-24, g0/1-2: vlan10, vlan 20, trunk.

– між комутаторами застосувати резервні зв'язки та агрегування каналів;

До маршрутизації врахувати наступні вимоги:

– на маршрутизаторах потрібно використовувати протокол динамічної маршрутизації OSPF;

– маршрут за умовчанням на маршрутизаторі з прямим підключенням до інтернет-провайдера (ISP) і розповсюдити його через оновлення маршрутизації;

– змінити еталонну пропускну спроможність для обчислення вартості за умовчанням для дозволу інтерфейсів Gigabit на значення = 1000.

Для налаштування роботи в Інтернет врахувати наступні вимоги:

– встановити одного провайдера послуг доступу до Інтернет (ISP);

– для виходу робочих станцій в Інтернет необхідно настроїти пограничний маршрутизатор з динамічним NAT з пулом адресів: 209.165.200.5 по 209.165.200.30;

– налаштувати сервер HTTP, щоб на вузлах при вводі в рядку браузера http://Kazakov.dp (http://209.165.200.4) відкривався веб-сайт з відомостями про тему та завдання на кваліфікаційну роботу студента.

2.1.3 Вимоги до видів забезпечення кіберфізичної системи

2.1.3.1 Вимоги до математичного забезпечення

Алгоритми обробки даних з датчиків, включаючи можливість фільтрації та усереднення, а також визначення поточного стану температури.

2.1.3.2 Вимоги до інформаційного забезпечення

Інформаційне забезпечення кіберфізичної системи моніторингу температури має повинен узгоджуватись таким вимогам:

Зберігання даних:

– система постійно вимірює температуру в домі. Ці показники зберігаються в базі даних разом із часом, коли вони були отримані. Це потрібно для того, щоб можна було потім переглянути історію температури та зробити аналіз або побудувати графіки;

– окрім температури, система також стежить за тим, що відбувається у вашому домі: зміни в роботі системи, будь-які помилки чи дії, які ви робите.

Сумісність з іншими системами:

– якщо у вас вдома є інші розумні пристрої (наприклад, освітлення або охоронна система), наша система може обмінюватися даними з ними. Для цього вона використовує відкриті протоколи, такі як MQTT. Це дозволяє всім пристроям розумного будинку працювати разом.

2.1.3.3 Вимоги до лінгвистичного забезпечення

Лінгвістичне забезпечення кіберфізичної системи моніторингу температури повинно відповідати наступним вимогам:

Мова програмування:

– для того, щоб наш мікроконтролер Raspberry Pi міг виконувати всі свої завдання, ми використовуємо мову програмування Python;

Взаємодія користувача з технічними засобами:

– коли компоненти системи обмінюються даними (наприклад, датчики передають показники контролеру), ми можемо використовувати будь-який

зручний формат, який дозволяє ефективно обробляти інформацію. Вибір формату залежить від потреб системи, але він повинен бути таким, щоб усі компоненти могли розуміти один одного;

– для забезпечення захисту інформації, особливо паролів користувачів, ми використовуємо шифрування. Це означає, що дані передаються в зашифрованому вигляді, і тільки ті, хто мають спеціальні ключі, можуть їх прочитати. Це важливо для запобігання несанкціонованому доступу до системи.

Опис контрольованих параметрів:

– щоб описати, як система виконує моніторинг температури, ми використовуємо її показники. Вся система побудована навколо контролю температури, тому температура і її одиниці вимірювання є ключовими для роботи та налаштування системи.

2.1.3.4 Вимоги до технічного забезпечення кіберфізичної системи

Технічне забезпеченням кіберфізичної системи моніторингу температури має враховувати в собі різноманітні компоненти, а саме:

– контролер Raspberry Pi, який є основою системи, у якому повинна бути підтримка Wi-Fi для роботи програми. Рекомендується використовувати модель з достатнім обсягом оперативної та флеш-пам'яті для забезпечення достатньої продуктивності для збору, обробки та відображення даних. Також контролер зобов’язаний забезпечувати принаймні підтримку інтерфейсів, які використовуються в роботі;

– датчик температури DS18B20 для сбору температури. Він є точним і зручним для вимірювання температури. Підключається до Raspberry Pi через один дріт і підтримує цифровий інтерфейс, що дозволяє легко зчитувати дані;

– блок для живлення має бути з віхідною напругою 5В та струмом не менше 3А. Необхідно також врахувати споживання енергії датчика DS18B20, хоча він використовує дуже малий струм (до 1 мА);

– резистор рекомендується використовувати 4.7 кОм для підключення лінії даних датчика DS18B20 до живлення, що забезпечує надійність передачі

сигналу;

– Wi-Fi роутер для забезпечення бездротового зв'язку між компонентами системи, що відповідає стандартам 802.11 b/g/n.

2.1.3.5 Вимоги до організаційного забезпечення

Організаційне забезпечення кіберфізичної системи моніторингу температури в розумному будинку передбачає чіткий розподіл відповідальностей між розробниками системи і її користувачами. Розробники відповідають за всі аспекти створення системи: від проектування і програмування до тестування, впровадження на об'єкті та подальшого підтримання. Вони також відповідають за надання консультацій користувачам і вирішення технічних питань, які можуть виникнути у процесі експлуатації.

Користувачі, у свою чергу, мають основну роль у щоденному користуванні системою. Вони моніторять температурні показники, адаптують налаштування для аналізу температури та покращення комфорту, і в разі необхідності вносять зміни у режим роботи пристроїв.

Організація функціонування системи починається з належного встановлення та налаштування обладнання на місці встановлення за участю технічно компетентних спеціалістів. Після цього важливим аспектом є проведення навчання користувачів з основ роботи з системою та її можливостями. Це дозволяє користувачам ефективно взаємодіяти з системою, розуміти її можливості та правильно використовувати для досягнення бажаних результатів.

2.2 Розробка апаратної частини комп’ютерної системи

2.2.1 Опису апаратних засобів кіберфізичної системи

Для кіберфізичної підсистеми система моніторингу температури для розумного будинку працює в межах локальної мережі, яка об'єднує всі її частини. В цій системі датчики температури DS18B20 підключаються до мікроконтролера Raspberry Pi, що виступає в ролі головного контролера. Цей мікроконтролер

збирає інформацію з датчиків, обробляє її, приймає рішення та відправляє дані на сервер IoT за допомогою Wi-Fi. Сервер отримує ці дані від мікроконтролера і зберігає їх у базі даних та у кінцевому результаті візуалізує їх (рис. 2.2).

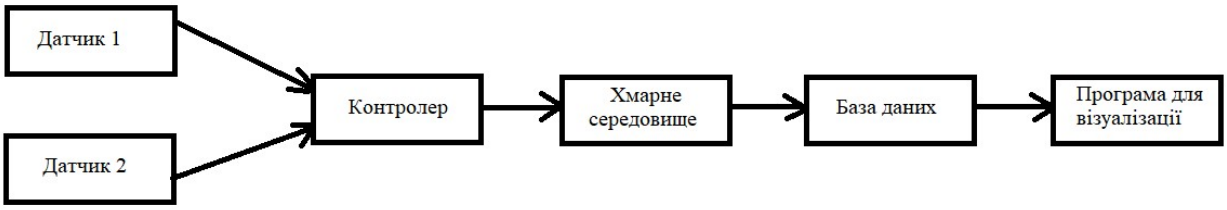

Рисунок 2.2 – Схема зв’язку компонентів системи

Основні компоненти та їх функції:

– датчики температури завдяки використанню протоколу 1-Wire підключаються до мікроконтролера;

– мікроконтролер збирає інформацію від датчиків, обробляє її та передає на сервер. Контролер також може здійснювати локальну обробку даних, наприклад, фільтрацію або усереднення значень температури, що дозволяє зменшити навантаження на сервер і забезпечити оперативну реакцію на зміни температури;

– хмарний середовище забезпечує віддалений доступ та обробку даних. Використовується для зберігання, обробки та аналізу даних, отриманих від мікроконтролера. Він також забезпечує доступ до даних з інших пристроїв та користувачів через мережу;

– база даних використовується для зберігання великих обсягів даних, що надходять від датчиків через хмарне середовище. Ці дані включають температуру. Вона надає швидкий доступ до даних, швидкий доступ до них, інтеграцію та масштабованість;

– сервіс для візуалізації даних дозволяє створювати інтерактивні графіки та інформаційні панелі, які можуть відображати поточні та історичні дані, що надходять з бази даних.

Датчик температури базується на популярній мікросхемі DS18B20. Він визначає температуру навколишнього середовища в діапазоні від -55 °C до

+125°C і передає дані у вигляді цифрового сигналу з 12-бітною роздільною здатністю за протоколом 1-Wire. Цей протокол дозволяє підключити величезну кількість таких датчиків, використовуючи лише один цифровий порт контролера та всього два дроти для всіх датчиків: землю та сигнал. У цьому випадку застосовується так зване паразитне харчування, при якому датчик отримує енергію прямо з лінії сигналу. Кожен датчик має унікальний прошитий на виробництві 64-бітний код, який може використовуватися мікроконтролером для спілкування із конкретним сенсором на загальній шині. Код кожного сенсора зчитується окремою командою.

У постійній пам'яті DS18B20 можна зберегти граничні значення температури, при виході з яких сенсор переходитиме в режим тривоги. На загальній шині з багатьох сенсорів мікроконтролер може відразу дізнатися, які з них перейшли в цей режим. Таким чином легко визначити проблемну ділянку в контрольованому середовищі.

Роздільна здатність показань налаштовується і становить від 9 до 12 біт. Менш роздільна здатність — вища швидкість перетворення.

Рисунок 2.3 – Датчик температури DS18B20

Датчик DS18B20 доступний також у водонепроникному варіанті, при якому він укладений у металеву циліндричну трубку. Ми його використовуватимемо у звичайному вигляді – у транзисторному корпусі. DS18B20 є однопровідним програмованим датчиком температури, що означає,

що йому необхідний лише один контакт даних, щоб передавати інформацію мікроконтролеру (у нашому випадку платі Raspberry Pi). Кожен датчик має свою унікальну адресу довжиною 64 біта, що дозволяє по одній і тій же шині (дроту) підключати до одного і того ж мікроконтролера безліч подібних датчиків - у цьому випадку звернення до конкретного датчика здійснюється на його адресу.

Таблиця 2.1 – Характеристики датчика DS18B20

Модуль:	DS18B20
Інтерфейс:	1-Wire
Діапазон вимірюваних температур:	−55…+125 °C
Точність:	±0,5 °C (не більше −10…+85 °C)
Роздільна здатність:	9/10/11/12 біт
Час отримання даних:	750 мс при 12-бітовій роздільній здатності 94 мс при 9-бітовій роздільній
Напруга живлення:	3-5,5 В
Діаметр гільзи:	6 мм
Довжина дроту:	100 см
Споживаний струм при бездіяльності:	750 нА
Споживаний струм при опитуванні:	1 мА

В Raspberry Pi 4 використовується однокристальна система Broadcom BCM2711. Кристал включає 4-ядерний 64-бітний процесор Cortex-A72 (ARM v8) з частотою 1,5 ГГц і графічний процесор GPU VideoCore VI з частотою 500 МГц. За даними виробника, система на новій архітектурі стала на 50% швидше, ніж попередні покоління RPi, а ще вона підтримує більше оперативної пам'яті - від 1 до 4 ГБ, залежно від версії плати.

Рисунок 2.4 – Контролер Raspberry Pi 4

Дана модифікація Raspberry Pi 4 Model B оснащується 4 ГБ пам'яті LPDDR4-2400 SDRAM, яка ділиться між CPU та GPU.

Raspberry Pi це повноцінний одноплатний комп'ютер розміром із банківську картку. Raspberry Pi 3 Модель B заснований на BCM2711 система-на-чіпі (SoC), яка включає 1,5 ГГц чотириядерний ARMv8 64-бітний процесор і потужний відеопроцесор VideoCore IV.

Raspberry Pi 4 Модель B має на борту бездротовий та Bluetooth зв'язок. Він має той же форм-фактор і розташування роз'ємів як у попередніх Raspberry Pi 2 Модель B і Raspberry Pi Модель B.

USB-пристрої, такі як клавіатури, миші та флешки можуть бути підключені через чотири порти USB плати.

З його 2,54 мм штирьовим роз'ємом GPIO і малим розміром, Raspberry Pi може також працює як програмований контролер у найрізноманітніших додатках робототехніки та електроніки.

2.2.2 Опису апаратних засобів комп’ютерної системи керуючої компанії

Посилаючись на завдання проекту були обрані апаратні рішення, які будуть відповідати всім поставленим питанням.

Конфігурація даного обладнання конфігурує в розробці проекту програми Cisco Packet Tracer та в налаштуванні типової мережі підприємства.

Для з’єднання всіх користувачів підмереж використовуємо комутатори Catalyst 2960, точки доступа Cisco, маршрутизатори Cisco 2911.

Комутатор Cisco Catalyst 2960 – це серія комутаторів Ethernet, які широко використовуються в бізнес-мережах для забезпечення швидкого і надійного підключення різних пристроїв до мережі. Вони мають різні моделі з різними характеристиками, але загалом Catalyst 2960 відомий своєю надійністю, продуктивністю і простотою управління. На рисунку 2.5 продемонстрований зовнішній вигляд комутатора Cisco Catalyst 2960.

Рисунок 2.5 – Зовнішній вигляд комутатора Cisco Catalyst 2960

Ключові характеристики Catalyst 2960:

– швидкість і пропускна здатність: Catalyst 2960 підтримує швидкість передачі даних до 10/100/1000 Мбіт/с на порт, залежно від моделі. Деякі моделі також можуть мати підтримку Gigabit Ethernet SFP портів;

– управління: Catalyst 2960 має різні можливості управління, включаючи Cisco IOS Software, яке дозволяє адміністраторам конфігурувати та моніторити мережеві параметри;

– підтримка безпеки: вони також забезпечують різні функції безпеки, такі як ACL (Access Control List і портова безпека;

– енергозбереження: деякі моделі Catalyst 2960 мають функції енергозбереження, які дозволяють ефективно використовувати електроенергію;

– стійкість і надійність: Catalyst 2960 відомий своєю стійкістю і надійністю, що робить їх популярними в бізнес-середовищах, де доступність мережі є критичною.В якості мережного обладнання було обрано маршрутизатор Cisco 2911, комутатор Cisco Catalyst 2960 та Wi-Fi роутер Linksys WRT-300N.

Модульний маршрутизатор Cisco 2911/K9 – це пристрій нового покоління, що відноситься до сімейства ISR G2. Модель дозволяє створювати безпечне широкосмугове підключення до мережі, проводити передачу мультимедійних даних, відео, здійснювати бездротовий зв'язок, а також застосовувати безліч додаткових функцій при мінімальному рівні витрат на придбання та утримання. На рис. 2.6 наведено зображення маршрутизатора Cisco 2911 в Packet Tracer.

Рисунок 2.6 – Маршрутизатор Cisco 2911

2.2.3 Розробка специфікації апаратних засобів

Для виконання поставленої задачі було розроблено специфікацію апаратних засобів кіберфізичної системи та комп’ютерної системи керуючої компанії MANCOMP1, у тому числі засобів збору та передачі інформації, інформації про які наведена в таблиці 2.2.

Таблиця 2.2 – Специфікація обладнання мережі та апаратних засобів

Компонент	Модель	Кількість, од.	Призначення
Мікроконтролер	Rasbperry Pi4	1	Збір даних, зв'язок з хмарним сервером
Датчик температури	DS18B20	2	Вимірювання температури
Блок живлення	5В, 2.5А	1	Живлення компонентів системи
Wi-Fi роутер	Cisco WRT300N	1	Забезпечення бездротового зв'язку
Комутатор	Cisco Catalyst 2950	3	Підключення кінцевих пристроїв
Маршрутизатор	Cisco 2911	1	Для доступу в Internet

2.3 Вибір і обгрунтування структурної схеми

На рис. 2.7 зображена структурна схема комплексу технічних засобів комп'ютерної системи мережі керуючої компанії. Ця схема відображає всі ключові компоненти мережі, їхні взаємозв'язки та способи підключення, що дозволяє зрозуміти структуру та принципи роботи мережі.

Хости об’єднуються в локальну мережу комутаторам другого рівня. Далі іде периметр мережі, який реалізований або маршрутизатором, який забезпечує захист мережі.

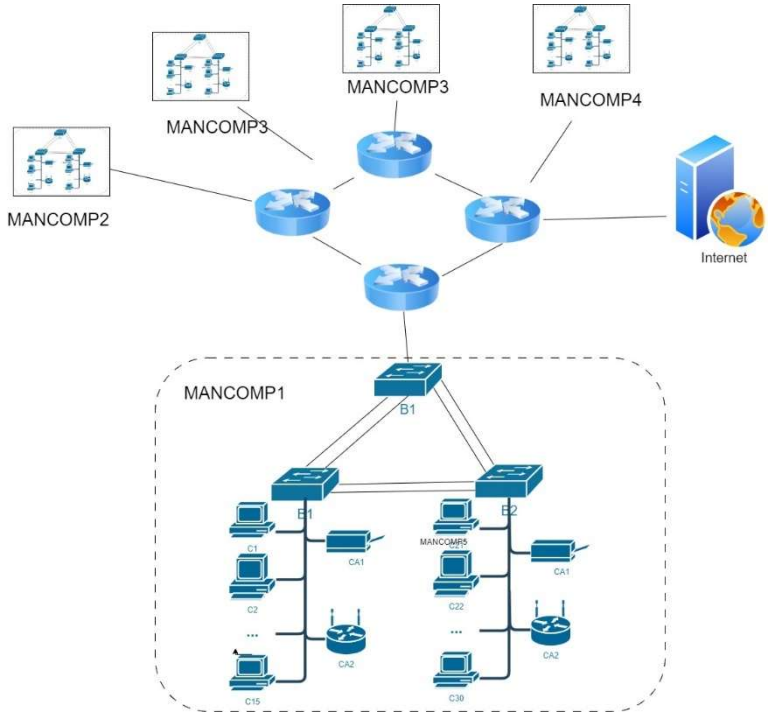

Рисунок 2.7 – Структурна схема комплексу технічних засобів комп'ютерної системи

Для забезпечення всіх вузлів у підмережі, буде встановлено відповідну кількість комутаторів. Так як у комутатора Cisco 2960 кількість портів 24 Fast + 2 Gigabit, та запас 10% достатньо 3 комутатори. Комутатор ядра з'єднується з маршрутизатором для доступу в Інтернет. На кожному поверсі передбачено Wi-Fi-маршрутизатор для безпровідного під'єднання через мобільні пристрої. Між комутаторами резервні зв'язки для додаткової надійності та збільшення пропускної здатності за рахунок агрегування каналів.

2.3.2 Розрахунок інтенсивності трафіку вихідного трафіку найбільшої локальної мережі підприємства

У мережі MANCOMP3 з 210 комп'ютерами трафік з комутатора надходить до роутера через лінк пропускною здатністю 1 Гбіт/с. Щоб уникнути перевантаження комутатора, швидкість прийому пакетів не повинна перевищувати швидкість їх відправлення. Припустимо, що всі користувачі (100%) одночасно активні. Середня інтенсивність трафіку становить 227 кадрів/секунду, а середня довжина повідомлення - 650 байт.

Розрахуємо пропускну здатність на рівні доступу, якщо всі користувачі найбільшої підмережі одночасно використовують мережу.

$$Pp.p. = \mu * L_{пов} * N * 8 \quad (2.1)$$

$$Pp.p. = 227 * 650 * 210 * 8 = 24.7 \text{ Мбіт/с}$$

де $L_{пов}$ – середня довжина повідомлення;

N – кількість вузлів в мережі.

Отриманий результат не перевищуватиме заданих параметрів мережі по вихідному каналу, отже перенавантаження не трапиться.

Комутатор також передає трафік до маршрутизатора зі швидкістю 1000 Мбіт/с. Отже, загальне навантаження на комутатор не повинно перевищувати:

$$\mu_{вих} = 10^9 / (650 * 8) = 192\ 307 \text{ пакетів/с} \quad (4.2)$$

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

$$N = \mu вих / \mu = 108\ 696 / 141 \approx 847 \text{ джерел} \quad (4.3)$$

До нашої найбільшої локальної мережі входять 15 ПК, тому ця кількість джерела буде задовольняти нашу кількість вузлів

Кожен з 210 ПК посилає потік заявок з інтенсивністю у 227 кадрів/с.

Розраховуємо інтенсивність вихідного трафіку:

$$\lambda = N * \mu = 15 * 227 = 3405 \text{ пакетів/с.} \quad (4.4)$$

Коефіцієнт затримки:

$$\rho = \lambda / \mu_{вих} = 3405 / 192307 = 0{,}018 \quad (4.5)$$

Коефіцієнт зайнятості маршрутизатора:

$$\rho / (1 - \rho) = 0{,}018 / (1 - 0{,}018) = 0{,}018 \quad (4.6)$$

Середня затримка кадру, пов’язана з чергою M/M/1, дорівнює:

$$T = 1 / (\mu - \lambda) = 1 / (192307 - 3405) = 5{,}29 \text{ мкс} \quad (4.7)$$

Середня довжина черги:

$$L_{чер} = \rho^2 / (1 - \rho) = 0{,}018^2 / (1 – 0{,}018) = 0{,}00033 \quad (4.8)$$

Середній час перебування пакета у черзі:

$$T_{оч} = Lчер / \lambda = 0{,}00033 / 3405 = 0{,}0097 \text{ мкс} \quad (4.9)$$

Це значення задовольняє вимогам до затримки в локальній мережі.

3 РОЗРОБКА КОРПОРАТИВНОЇ МЕРЕЖІ

3.1 Розробка логічної топології

Перш за все перед налаштуванням мережі необхідно проаналізувати її логічну топологію і зрозуміти, як розташовані і підключені мережеві пристрої.

Логічна топологія розроблена з використанням середовища Cisco Packet Tracer. Ми побудували схему, використовуючи компоненти та інструменти цього середовища. Змодельовану типову корпоративну мережу можна побачити на рисунку 3.1.

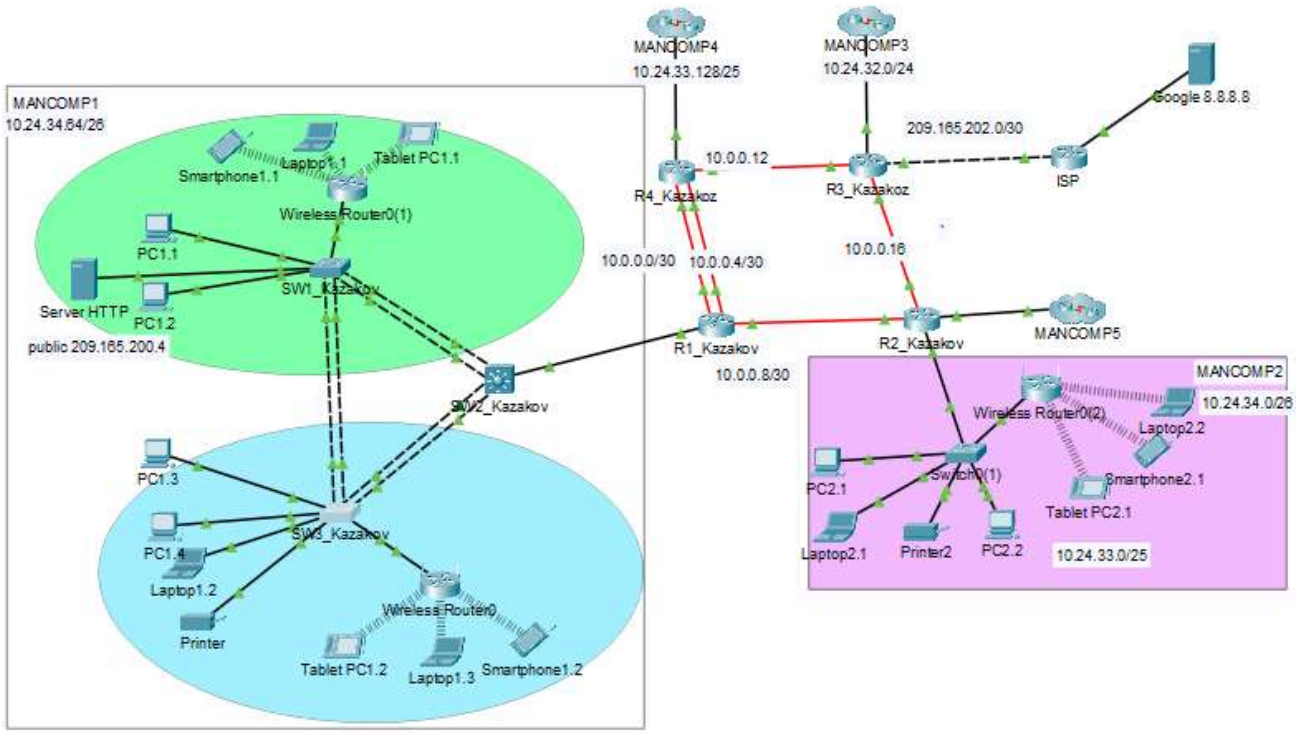

Рисунок 3.1 – Розроблена логічної топологія мережі

Топлогія складається 3 5-х мереж MANCOMP1–MANCOMP5 та, та віддаленого сервера Google, для перевірки доступу до Інтернет. В таблиці 3.1 схема адресації мереж. Для адресації мережі MANCOMP1 та MANCOMP2 використовується приватний блок класу А з блоку 10.24.32.0/21, а для адресації каналів WAN між маршрутизаторами блок адрес класу А 10.0.0.0/24. Мережа Internet представлена публічним адресним блоком 8.8.8.0/24 для моделювання

доступу до Інтернет. Для адресації використовувався метод VLSM, яким мережа розбивається на менші підмережі з масками різної довжини за розміром. В свою чергу мереж MANCOMP1 поділиться на 2 рівні підмережі для впровадження VLAN:VLAN 20 (DEVELOPER) для відділу розробки, VLAN10 (OFFICE) для інших відділів;

Таблиця 3.1 – Адресація підмереж

Назва	Кіл-ть хостів	Адреса мережі	Префікс	Діапазон допустимих адрес
MANCOMP3	210	10.24.32.0	/24	10.24.32.1 - 10.24.32.254
MANCOMP2	79	10.24.33.0	/25	10.24.33.1 - 10.24.33.126
MANCOMP4	78	10.24.33.128	/25	10.24.33.129 - 10.24.33.254
MANCOMP5	43	10.24.34.0	/26	10.24.34.1 - 10.24.34.62
MANCOMP1	37	10.24.34.64	/26	10.24.34.65 - 10.24.34.126
VLAN10	20	10.24.34.64	/27	10.24.34.65 - 10.24.34.94
VLAN20	20	10.24.34.96	/27	10.24.34.97 - 10.24.34.126
WAN1	2	10.0.0.0	/30	10.0.0.1 – 10.0.0.2
WAN2	2	10.0.4.0	/30	10.0.0.5 – 10.0.0.6
WAN3	2	10.0.8.0	/30	10.0.0.9 – 10.0.0.10
WAN4	2	10.0.12.0	/30	10.0.0.13 – 10.0.0.14
WAN5	2	10.0.16.0	/30	10.0.0.17 – 10.0.0.18

SW1_Kazakov та SW2_Kazakov комутатори доступа, які зєднують вузли в мережі MANCOMP1 в одну корпоративну мережу. Комутатор SW3_Kazakov комутатор ядра, і підключений до маршрутизатора R1_Kazakov.

В таблиці 3.2 наведено схема адресації пристроїв в MANCOMP1.

Таблиця 3.2 – Таблиця адресації пристроїв в MANCOMP1

Пристрій	Інтерфейс	IPv4 address	Subnet mask	Gateway
R1	GigabitEthernet0/0.10 GigabitEthernet0/0.20 GigabitEthernet0/3/0 GigabitEthernet0/1/0 GigabitEthernet0/2/0	10.24.34.65 10.24.34.97 10.0.0.2 10.0.0.3 10.0.0.9	255.255.255.224 255.255.255.224 255.255.255.252 255.255.255.252 255.255.255.252	–
Server HTTP	Fa0	10.24.34.70	255.255.255.192	10.24.34.65

3.2 Виконання базового налаштування конфігурації пристроїв

В мережі компанії MANCOMP1 аідповідно до вимог в розділі 2.1.2.3 необхідно задати:

– назви пристроям за наступним правилом: тип пристрою_номер пристрою_ Kozakov;

– на всіх пристроях повинен бути назначений пароль cisco до консолі і vty;

– на всіх пристроях повинен бути назначений пароль class до привілейованого режиму;

– усі паролі, що зберігаються у відкритому вигляді, потрібно зашифрувати;

– потрібно назначити на усіх лініях vty використання протоколу SSH;

– потрібно призначити ім’я користувача та пароль на всіх пристроях за правилом: 123201_Kozakov з паролем admincisco;

– в якості імені домена потрібно використати ім'я пристрою;

– для шифрування даних ключ RSA завдовжки 1024 біт.

На рисунку 3.2 наведено налаштування на R1. Аналогічні налаштування виконуються і на інших маршрутизаторах.

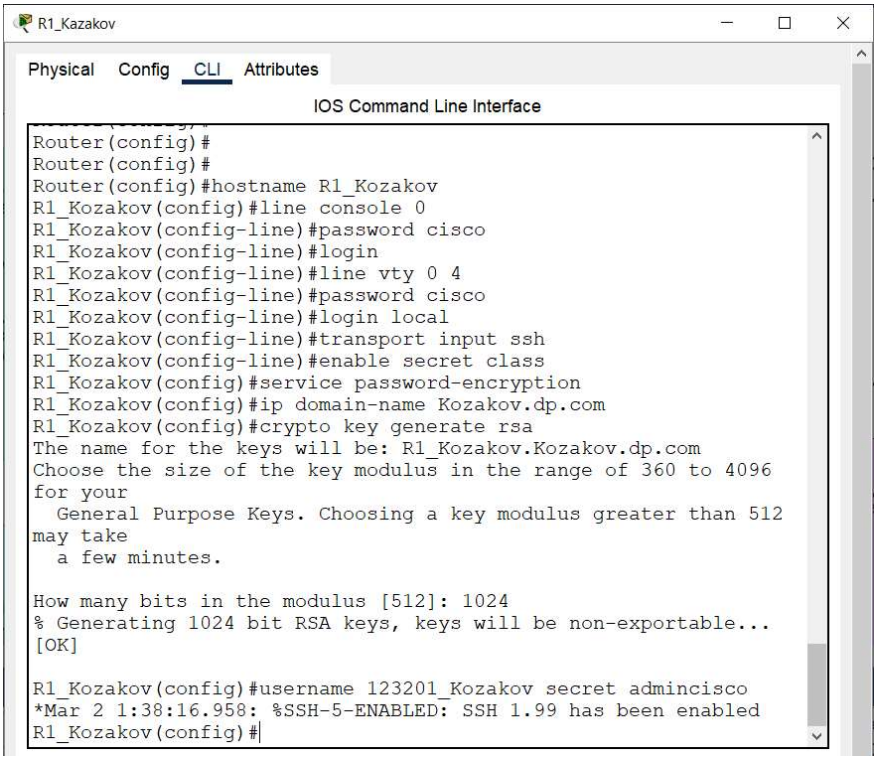

Рисунок 3.2 – Базові налаштування на R1

3.3 Налаштування та перевірка способу маршрутизації

Організація ефективної та стабільної мережі в корпоративній мережі є ключовим фактором для успіху. Для цього необхідно обрати спосіб маршрутизації та забезпечити надійний доступ до Інтернету.

OSPF (Open Shortest Path First) це протокол маршрутизації внутрішнього домену, який використовується для створення маршрутної таблиці на основі найкоротшого шляху.

Переваги OSPF:

– OSPF швидко адаптується до змін у мережі, що забезпечує мінімальні перебої в роботі;

– протокол може масштабуватись для великих мереж, що особливо важливо для геймінг-кафе з великою кількістю користувачів;

– OSPF забезпечує надійну маршрутизацію завдяки механізму виявлення помилок та резервування.

У цьому контексті протокол маршрутизації OSPF (Open Shortest Path First) є надійним та ефективним рішенням. OSPF дозволяє динамічно адаптувати маршрути передачі даних відповідно до змін у мережі, забезпечуючи оптимальний шлях для пакетів. Це особливо важливо в середовищі геймінг-кафе, де мережа може зазнавати значного навантаження та змін.

На рисунку 3.3 продемонстровано приклад налаштування OSPF на R1_Kazakov. Конфігурація передбачає, що маршрутизатор має інтерфейси, підключені до мереж з блоку 10.0.0.0/28 та мережі 10.24.34.64/26 :

– router ospf 1 – ця команда активує процес OSPF на маршрутизаторі з ідентифікатором процесу 1. Ідентифікатор процесу OSPF є локальним для маршрутизатора і не повинен збігатися з ідентифікаторами процесів інших маршрутизаторів у мережі;

– router-id 1.1.1.1 – використовується для встановлення унікального ідентифікатора маршрутизатора в домені OSPF і для ідентифікації маршрутизатора при виборі повідомлень OSPF і DR і BDR;

– auto-cost reference-bandwidth 1000 – визначає базову смугу пропускання, яка використовується для розрахунку вартості інтерфейсів. Чим нижча еталонна смуга пропускання, тим вища вартість інтерфейсів із меншою смугою пропускання;

– network 10.0.0.0 0.0.0.15 area 0 – ця команда оголошує мережу для участі у процесі OSPF. У цьому випадку мережа 10.0.0.0 з маскою підмережі 255.255.255.240 (зворотна маска підмережі 0.0.0.15) додається до області OSPF 0. Оголошення мережі в OSPF дозволяє маршрутизатору обмінюватися маршрутною інформацією з іншими OSPF-маршрутизаторами в тій же області;

– network 10.24.34.64 0.0.0.63 area 0 – цей рядок рекламує мережу 10.24.34.64/26 як мережу OSPF.

```
R1_Kozakov(config)#router ospf 1
R1_Kozakov(config-router)#router-id 1.1.1.1
R1_Kozakov(config-router)#auto-cost reference-bandwidth 1000
R1_Kozakov(config-router)#network 10.0.0.0 0.0.0.15 area 0
R1_Kozakov(config-router)#network 10.24.34.64 0.0.0.63 area 0
R1_Kozakov(config-router)#
```

Рисунок 3.3 – Налаштування OSPF на R1_Kazakov

Команда show ip ospf neighbor використовується для відображення стану маршрутизаторів-сусідів OSPF (див. рисунок 3.4). Вона надає інформацію про всіх сусідів OSPF, з якими поточний маршрутизатор встановив сусідські відносини.

```
R1_Kozakov#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
4.4.4.4           1   FULL/DR         00:00:35    10.0.0.5        GigabitEthernet0/1/0
4.4.4.4           1   FULL/DR         00:00:35    10.0.0.1        GigabitEthernet0/3/0
2.2.2.2           1   FULL/DR         00:00:35    10.0.0.10       GigabitEthernet0/2/0
R1_Kozakov#
```

Рисунок 3.4 – Сусідні ip адреса OSPF

Аналогічно було виконано налаштування і на інших маршрутизаторах в мережі. Після обміну інформацією в таблиці маршрутизації повинні відображатися записи з IP-адресам віддалених мереж з символом «O» на початку.

На рисунку 3.5 таблиця маршрутизації на R1_Kazakov відображає всі відомі йому мережі.

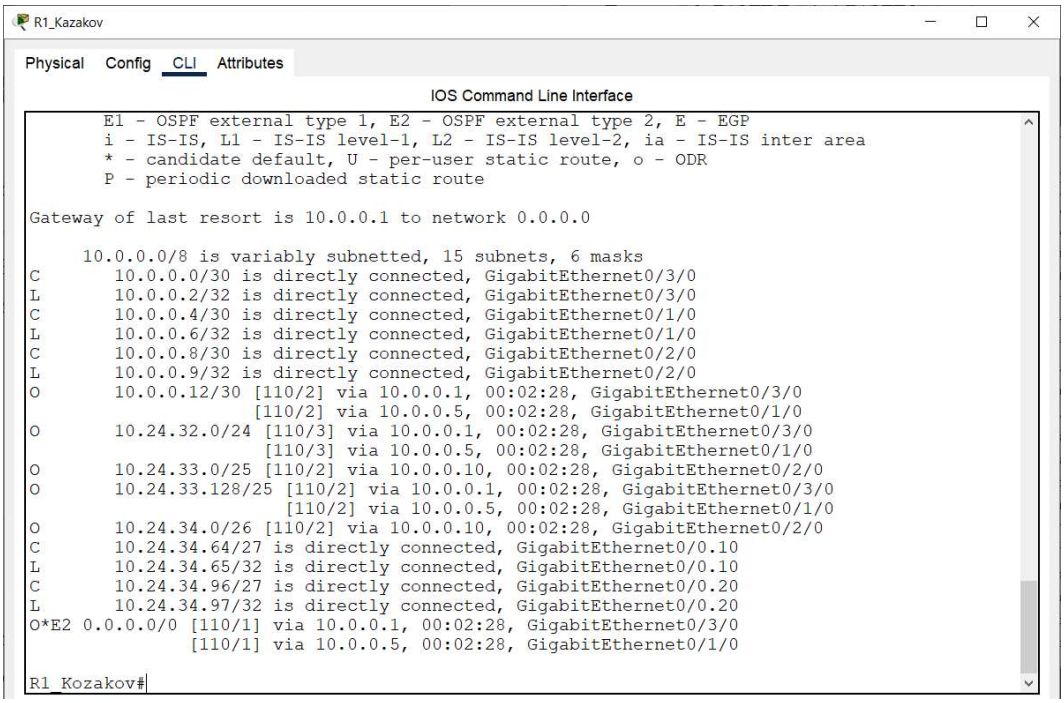

Рисунок 3.5 – Таблиця маршрутизації на R1_Kazakov

Виконаємо трасування маршруту від R1_Kazakov до всіх інших маршрутизаторів, щоб упевнитись у правильних налаштуваннях мережі та в тому, що мережі можуть обмінюватись між собою без перешкод (див. рисунок 3.5).

Отримані результати нас задовольняють, тому можна сказати, що побудова мережі та налаштування маршрутизації виконано успішно.

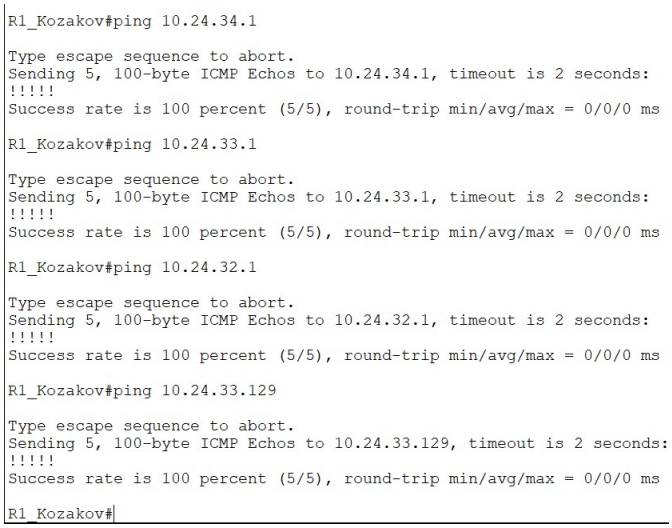

Рисунок 3.6 – Перевірка доступності маршрутизаторів від R1_Kazakov

3.4 Налаштування та перевірка доступу в Internet

Поряд із маршрутизацією, важливим аспектом є організація доступу до Інтернету. Маршрутизатор R3_Kazakov з'єднаний з провайдером ISP. Тому на цьому маршрутизаторі необхідно створити маршрут за замовчуванням і розповсюдити його через повідомлення OSPF. Також

```
R3_Kazakoz(config)#ip route 0.0.0.0 0.0.0.0 GigabitEthernet0/2
R3_Kazakoz(config)# router ospf 1
R3_Kazakoz(config-router)#default-information originate
R3_Kazakoz(config-router)#
```

Рисунок 3.7 – Налаштування маршруту до Інтернет

На рисунку 3.6 в табдиці маршрутихації на R1_Kazakov jстанній рядок O*E2 0.0.0.0/0 – це маршрут за замовчуванням, який створений на R3_Kazakov і розповсюджений через OSPF.

Для NAT надано пул адрес 209.165.202.5 209.165.202.30. Я для сервера HTTP статичний NAT з адресою 209.165.200.5.

На рисунку наведено зміст налаштувань NAT в конфігураційному файлі R3_Kazakoz.

```
interface GigabitEthernet0/3/0
 ip address 10.0.0.17 255.255.255.252
 ip nat inside
!
interface Vlan1
 no ip address
 shutdown
!
router ospf 1
 router-id 3.3.3.3
 log-adjacency-changes
 auto-cost reference-bandwidth 1000
 network 10.0.0.0 0.0.0.15 area 0
 network 10.24.32.0 0.0.0.255 area 0
 default-information originate
!
ip nat pool NAT 209.165.202.5 209.165.202.30 netmask 255.255.255.224
ip nat inside source list NAT_LAN pool NAT overload
ip nat inside source static 10.24.34.70 209.165.200.4
ip classless
ip route 0.0.0.0 0.0.0.0 GigabitEthernet0/2
!
ip flow-export version 9
!
!
ip access-list standard NAT_LAN
 permit 10.24.0.0 0.0.255.255
!
```

Рисунок 3.8 – Налаштуванна NAT на R3_Kazakov

Щоб перевірити роботу динамічного NAT з будь-якого вузла в MANCOMP1 виконаємо трасування до 8.8.8.8 (рис.3.9).

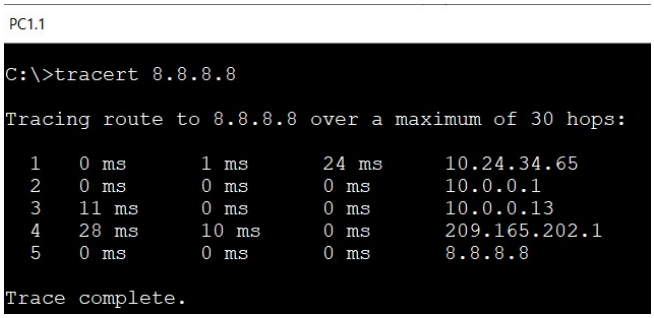

Рисунок 3.9 – Tracert до вузла Internet

І на R3_Kazakov створюється таблиця сесій NAT (рис.3.10).

```
R3_Kazakoz#sh ip nat  translations
Pro  Inside global     Inside local       Outside local      Outside global
icmp 209.165.202.5:62  10.24.34.75:62     8.8.8.8:62         8.8.8.8:62
icmp 209.165.202.5:63  10.24.34.75:63     8.8.8.8:63         8.8.8.8:63
icmp 209.165.202.5:64  10.24.34.75:64     8.8.8.8:64         8.8.8.8:64
icmp 209.165.202.5:65  10.24.34.75:65     8.8.8.8:65         8.8.8.8:65
icmp 209.165.202.5:66  10.24.34.75:66     8.8.8.8:66         8.8.8.8:66
icmp 209.165.202.5:67  10.24.34.75:67     8.8.8.8:67         8.8.8.8:67
---  209.165.200.4     10.24.34.70        ---                ---

R3_Kazakoz#
```

Рисунок 3.10 – Таблиця NAT на R3_Kazakov

Також перевіримо статичний NAT шляхом завантаження сторінки WEB-сервера за його публічною адресою (рис. 3.11). Сторінка завантажилася, значить завдання виконано вірно.

Рисунок 3.11 – Перевірка статичного NAT

3.5 Налаштування агрегування каналів

Між комутаторами резервні зв’язки для додаткової надійності та збільшення пропускної здатності за рахунок агрегування каналів. Застосуємо технологію PAgP та створимо інтерфейси до рисунку 3.12.

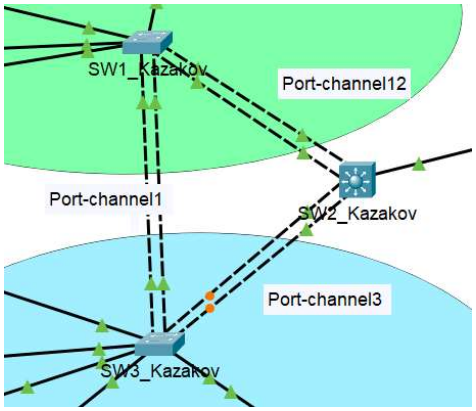

Рисунок 3.12 – Впровадження EtherChannel

На рисунку 3.13 результат перевірки створеного агрегованого каналу 1. Бачимо, що пропускна здатність його 200 Мб/с і створюють його порти f0/21-22.

```
SW1_Kazakov#show interfaces port-channel 1
Port-channel1 is up, line protocol is up (connected)
  Hardware is EtherChannel, address is 0040.0b49.6964 (bia 0040.0b49.6964)
  MTU 1500 bytes, BW 200000 Kbit, DLY 1000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Half-duplex, 200Mb/s
  input flow-control is off, output flow-control is off
  Members in this channel: Fa0/21 ,Fa0/22 ,
  ARP type: ARPA, ARP Timeout 04:00:00
```

Рисунок 3.13 – Відомості про port-channel 1

3.6 Налаштування безпеки мережі

3.6.1 Захист від несанкціонованого віддаленого доступу

Для налагодження функціонування протоколів віддаленого доступу (зокрема, Telnet та SSH) на пристроях Cisco використовуються як деякі загальні для всіх протоколів команди, так і характерні лише для певного протоколу команди. До загальних команд належать такі команди: password, username, login, transport.

Cisco VTY – віртуальний інтерфейс, за допомогою якого можна забезпечити віддалений доступ до пристрою.

Для налаштування на R1_Kazakov потрібно встановити домене ім’я, згенерувати RSA ключи для шифрування трафіку SSH та ввести розмір ключа, створити обліковий запис користувача, налаштувати лінії VTY для прийому

підключень через SSH, налаштувати SSH на 2 версію, а також можна налаштувати тайм-аут для підключення через SSH.

Сценарій налагодження віддаленого підключення за протоколом SSH до маршрутизатора TS наведено на рисунку 3.14.

```
R1_Kozakov(config)#
R1_Kozakov(config)#line vty 0 4
R1_Kozakov(config-line)#password cisco
R1_Kozakov(config-line)#transport input ssh
R1_Kozakov(config-line)#login local
R1_Kozakov(config-line)#logging synchronous
R1_Kozakov(config-line)#exec-timeout 60
R1_Kozakov(config-line)#ip domain-name Kazakov.dp.com
R1_Kozakov(config)#ip ssh version 2
R1_Kozakov(config)#ip ssh time-out 120
R1_Kozakov(config)#username 123201_Kazakov privilege 15 secret admincisco
```

Рисунок 3.14 – Налаштування безпечного доступу до R1

За допомогою команди ssh -l 123201_Kazakov 192.168.100.1 і введенням паролю я підключаюсь за протоколом SSH (див. рисунок 3.15):

```
C:\> ssh -l 123201_Kazakov 10.24.34.65

Password:


R1_Kozakov#
```

Рисунок 3.15 – Підключення за протоколом SSH

Після вводу пароля ми попадаємо в командний рядок маршрутизатора. Доступ налаштовано успішно.

3.6.2 Налаштування мереж VLAN

Впровадження VLAN значно підвищує рівень безпеки локальної мережі, оскільки забезпечує:

– ізоляцію трафіку між різними сегментами мережі, запобігаючи несанкціонованому доступу;

– можливість застосування індивідуальної політики безпеки та контролю доступу для кожного з VLAN;

– централізоване управління мережевими ресурсами, полегшуючи адміністрування та моніторинг;

– підвищення стійкості мережі до атак, оскільки компрометація одного сегмента не призводить до впливу на весь мережевий трафік.

В мережі MANCOMP1 застосуємо технологію VLAN для розділення мережі на 2 рівні сегменти: VLAN 20 (DEVELOPER) для відділу розробки, VLAN10 (OFFICE) для інших відділів;

На комаутаторах SW1_Kazakov та SW3_Kazakov розподілимо порти між VLAN за наступними правилами (рис. 3.16):

– f0/1-10: vlan10, access;

– f0/11-20: vlan20, access;

– f0/21-24, g0/1-2: vlan10, vlan 20, trunk.

```
SW1_Kazakov#sh vlan brief

VLAN Name                              Status    Ports
---- --------------------------------- --------- -------------------------------
1    default                           active    Gig0/1, Gig0/2
10   Office                            active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                 Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                                 Fa0/9, Fa0/10
20   Developer                         active    Fa0/11, Fa0/12, Fa0/13, Fa0/14
                                                 Fa0/15, Fa0/16, Fa0/17, Fa0/18
                                                 Fa0/19, Fa0/20
1002 fddi-default                      active
1003 token-ring-default                active
1004 fddinet-default                   active
1005 trnet-default                     active
SW1_Kazakov#
```

Рисунок 3.16 – Вдомості про налаштування VLAN на SW1_Kazakov

Комутатор SW1_Kazakov центральний, порти в ньому будуть в режимі trunk.

Для перевірки того, що мережа MANCOMP1 успішно розділена на VLAN, відправимо виконаємо tracert з ПК в сегменті OFFICE на ПК в сегмент Developer (рис. 3.17). Як бачимо, команда виконана успішно.

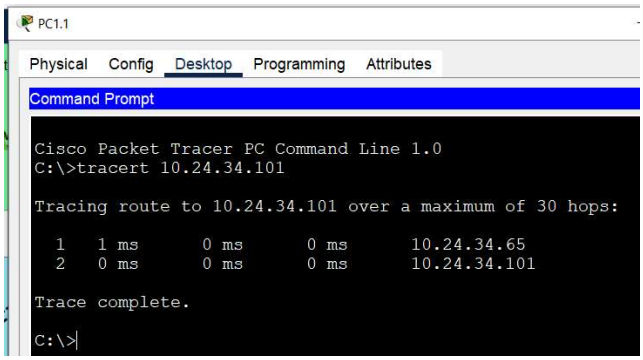

Рисунок 3.17 – Tracet з ПК в OFFICE на ПК в сегмент Developer

В режимі Simulation для передачі між сегментами пакет досягає підінтерфейса маршрутизатора та має заголовок 801.21q (рис. 3.18).

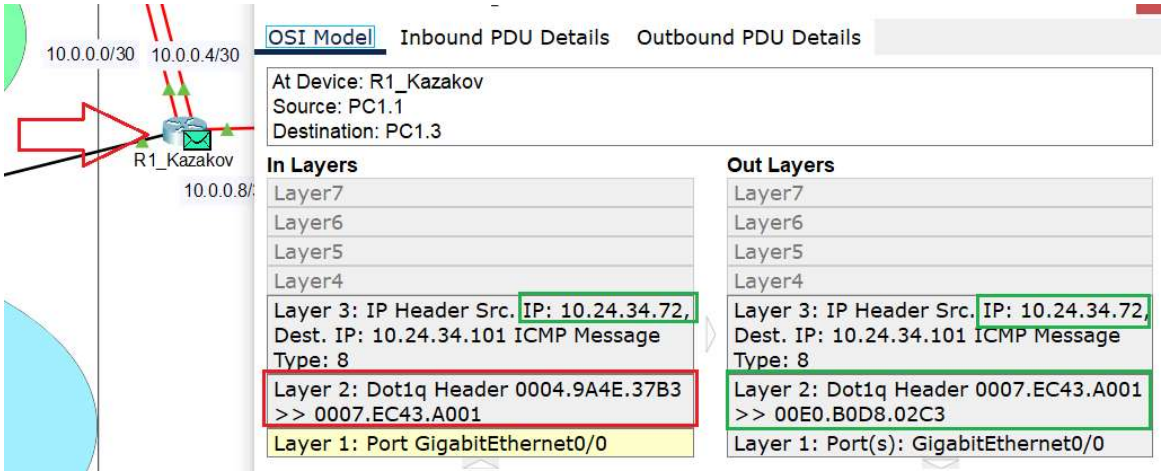

Рисунок 3.18 – Передача пакету між VLAN

4 ПРОЕКТУВАННЯ КІБЕРФІЗИЧНОЇ СИСТЕМИ МОНІТОРИНГУ ТЕМПЕРАТУРИ

4.1 Сценарій розробки кіберфізичної системи

Після огляду платформи та вибору компонентів створимо кіберфізичну систему моніторингу температури на базі платформи AWS IoT.

Сценарій розробки:

– підключення датчика температури до Onion2+;

– налаштування Amazon IoT таким чином, щоб Omega2+ був «річчю» в хмарі Amazon IoT;

– передача показзників температури на AWS IoT Core по протоколу MQTT;

– створення Roles, яка дозволить правилу записувати в таблицю потоку часу дані;

– налаштування бази даних DynamoDB на Amazon AWS для зберігання показників температури з Onion2+;

– налаштування веб-сторінки для читання даних із DynamoDB для їх представлення як діаграми.

На рисунку 4.1 показана архітектура систем.

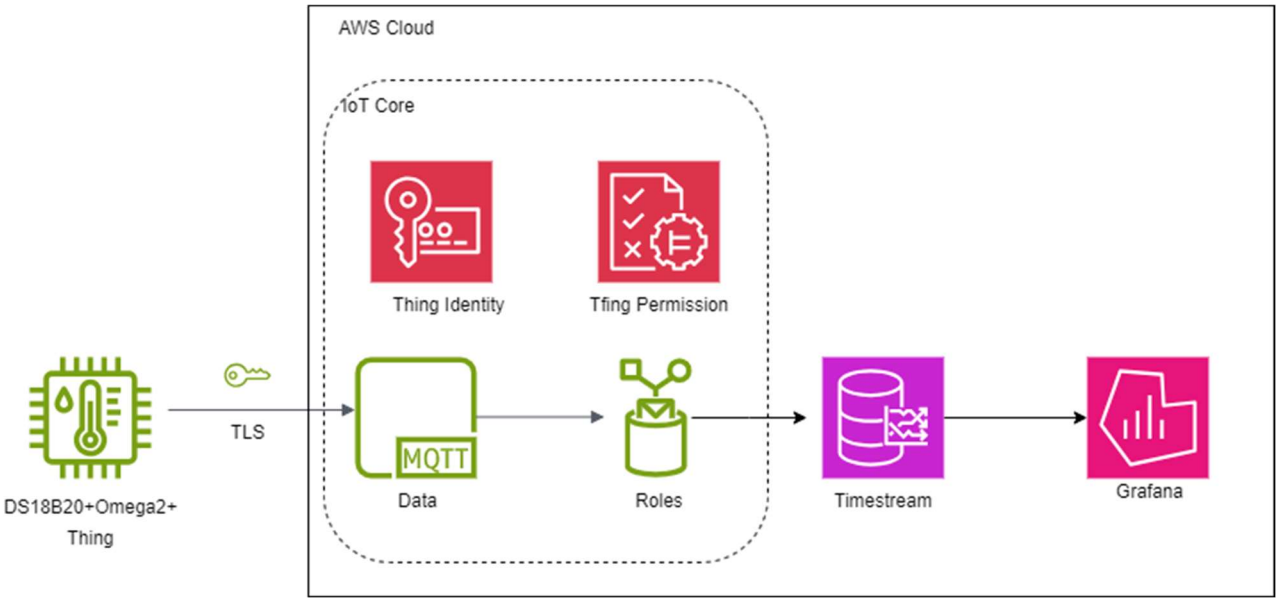

Рисунок 4.1 – Архітектура кіберфізичної системи моніторингу температури

4.2 Розробка кіберфізичної системи

Схема підключення датчиків температури DS18B20 до плати Raspberry Pi представлена на рисунку.(див. рис 4.2)

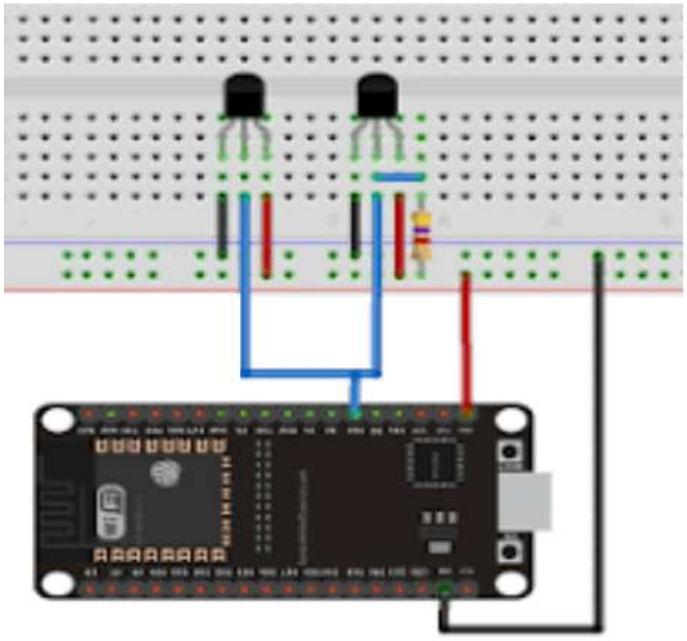

Рисунок 4.2 – Схема підключення датчику до Raspberry Pi 4

У представленій схемі датчик DS18B20 запитується від контакту 5V плати Raspberry Pi. Контакт даних датчика DS18B20 підключено до контакту GPIO 4 плати Raspberry Pi. Резистор 4,7 кОм, підключений до датчика, виконує роль резистора, що підтягує.

Об'єктові контролери підключаються до сервісу AWS IoT-Core за допомогою протоколу TLS. Адреса сервера, що використовується для підключення, містить унікальний обліковий запис ідентифікатор. Аутентифікація здійснюється за протоколом x509. Кожен пристрій використовує власний унікальний сертифікат.

4.3 Опис процесу підключення до AwS IoT

Для початку роботи нам потрібнен обліковий запис Amazon AWS. Для цього проводимо реєстрацію та створюємо обліковий запис користувача. Після чого переходимо к сервісу AWS IoT Core, натискаємо на «створити річ». Для виконання нашого завдання буде достатньо однієї речі, тому натискаємо

«створити річ» для того, щоб почати реєстрацію контролера в ядрі aws iot, після чого обираємо «створити одну річ» та надаємо їй ім’я. Називаємо її в честь нашого контролера, який буде використовуватися у роботі. (див. рис. 4.3)

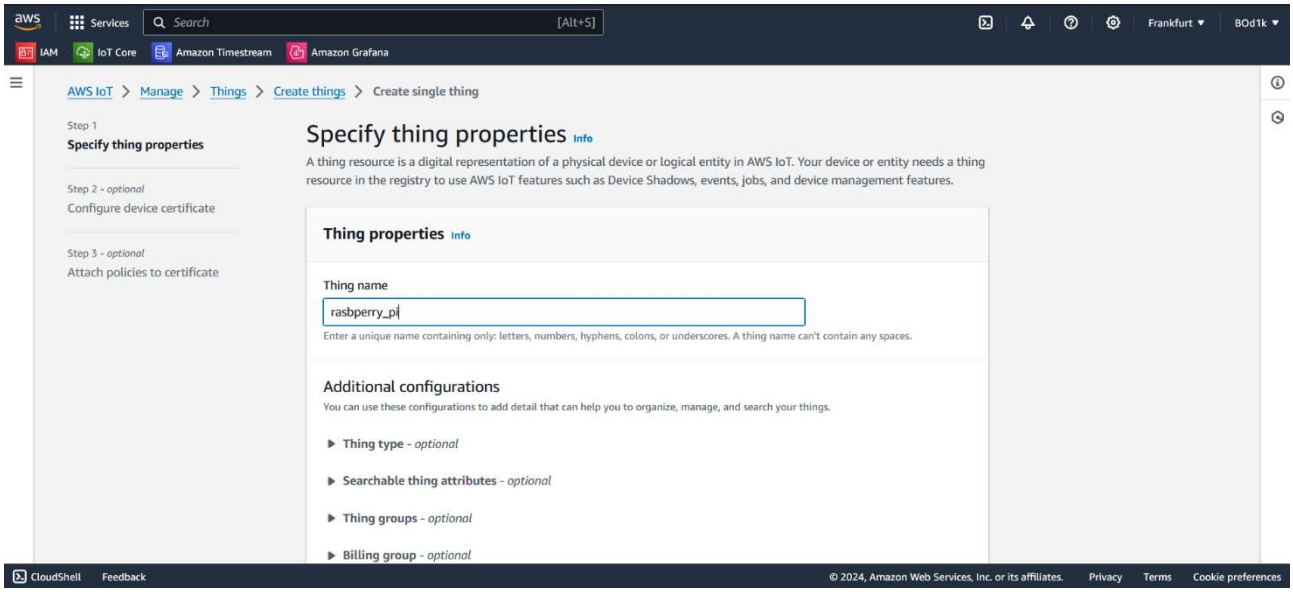

Рисунок 4.3 – Процес створення “Річ“

Наступним кроком буде створення засобу автентифікації зв’язку нашого Raspberry з AWS IoT. Скористаємося автоматичною генерацією сертифікату. (див. рис. 4.24) Далі нам буде запропоновано додати політику для нашої речі, але за замовчуванням вона буде відсутня, тому зробимо її та додамо вже після закінчення процесу створення. Далі натискаємо «створити річ». Слід зазначити, що після потрібно буде завантажити сертифікат, відкритий і закритий ключі на комп’ютер, а потім активувати їх для подальшої роботи. (див. рис. 4.4)

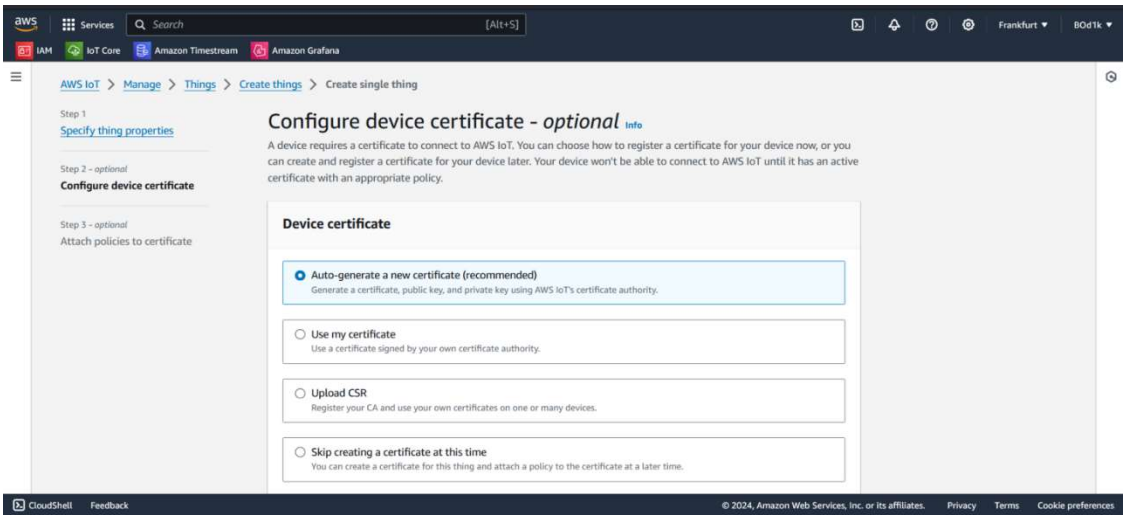

Рисунок 4.4 – Автоматична генерація сертифікату

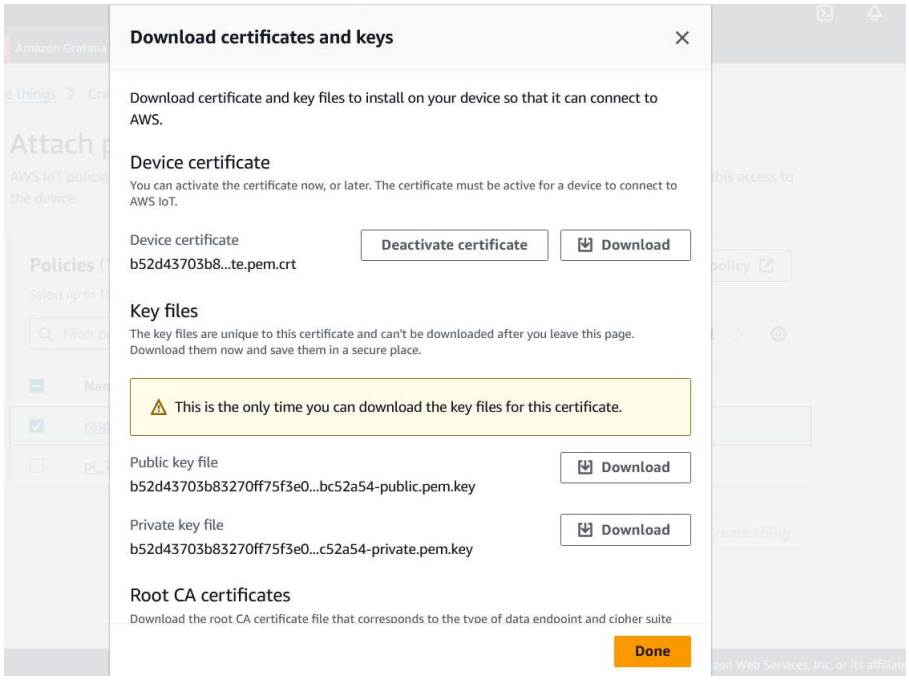

Рисунок 4.5 – Вікно завантаження ключів та сертифікату

Завантажені ключі та сертифікат ми помістимо у відповідно теку під назвою «aws2», вона нам знадобиться для подальшої роботи з контролером, по суті це буде тека нашого проекту, яку ми потім повинні будемо перемістити на контроллер. (див. рис. 4.6)

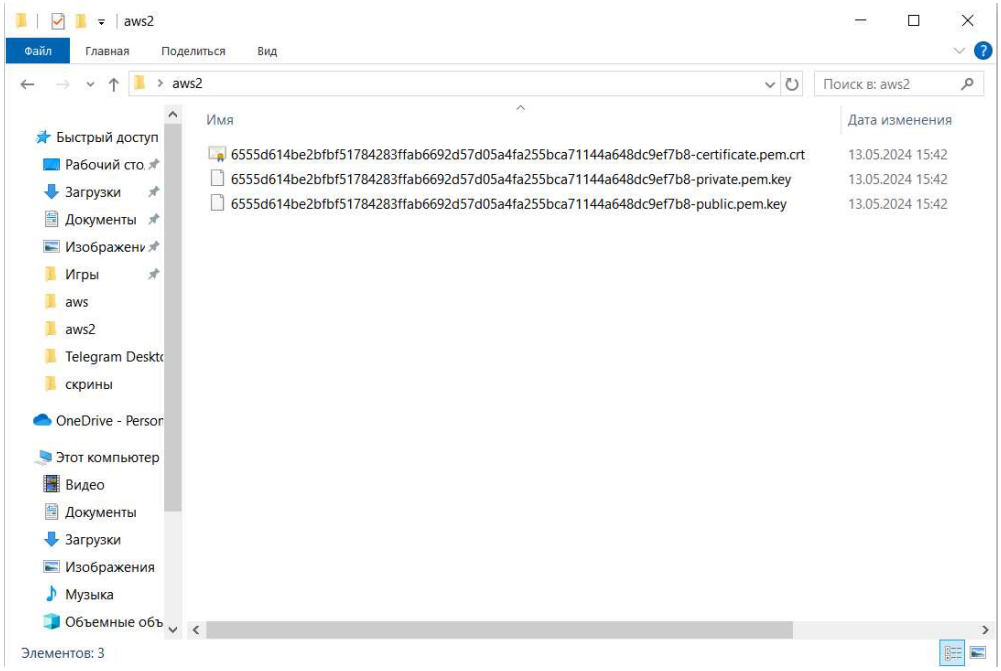

Рисунок 4.6 – Папка проекту

Тепер наша річ створена і ми повернулись до основної консолі AWS IoT. Так як у попередньому налаштуванні ми пропустили політику через її

відстутність під час реєстрації речі, виконаємо це зараз. Переходимо в меню «безпека», і потім у підменю «політика» і натискаємо «створити політику». На сторінці «Створення політики» у полі «Ім'я» вводимо ім'я політики, у нашому випадку це «raspberry_thing_policy». У полі “Дія” вводимо “iot”: *. У полі ресурсу ARN вводимо *.

Встановлюємо прапорець "Дозволити". Це дозволяє нашому Raspberry Pi публікувати повідомлення в AWS IoT.

iot: * - політика підписки та публікації з використанням цього сертифіката;

* - всі клієнти можуть публікувати / підписуватись на цю річ, використовуючи цей сертифікат. (див. рис. 4.7)

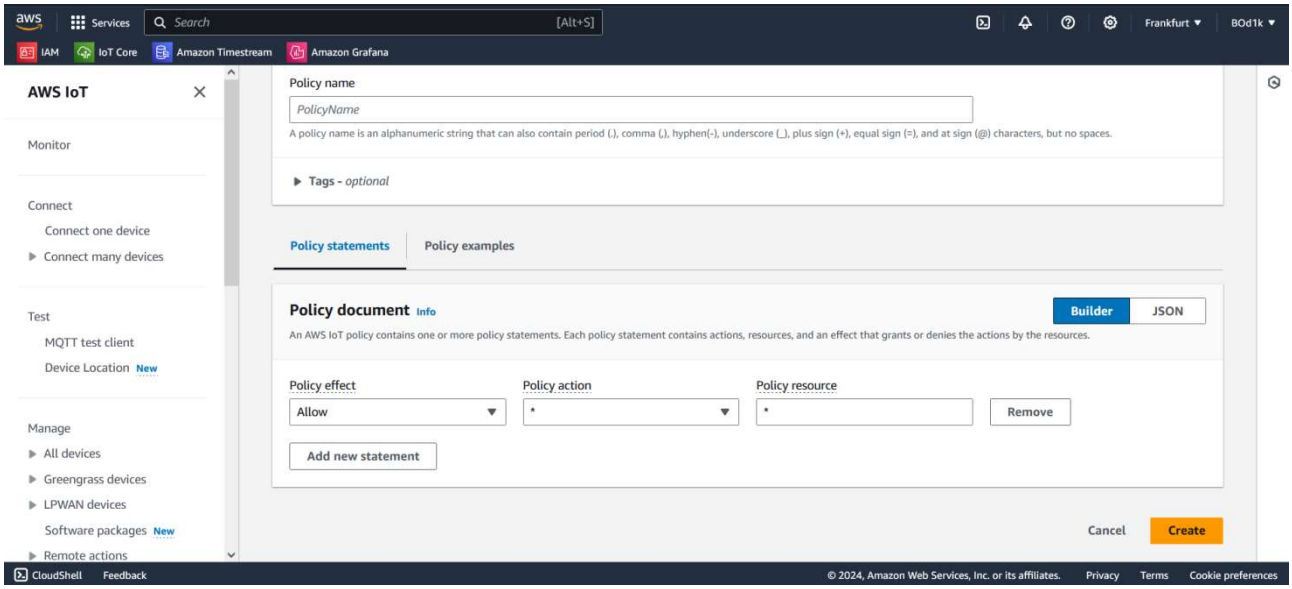

Рисунок 4.7 – Опис політики

Політика, яку можна використовувати з безліччю сертифікатів, встановлює дозволи для пристроїв, які перевіряються за допомогою сертифіката. Вона дозволяє здійснювати з'єднання через протокол MQTT від клієнтів, ідентифікатори яких відповідають іменам речей.

Короткий огляд дозволів, які надає наша політика:

– дозволяє здійснювати лише одне MQTT-з'єднання з AWS IoT за допомогою цього сертифіката;

– дозволяє під'єднаному пристрою отримувати та публікувати повідомлення в усіх темах MQTT, що належать до тіні вашої речі;

– дозволяє підключеному пристрою підписуватися на всі теми, пов'язані з оновленням і отриманням тіні «речі»;

– дозволяє під'єднаному пристрою отримувати та оновлювати тінь Істоти.

Після цього потрібно під’єднати завантажений сертифікат до створеної «речі» та політики. (див. рис. 4.8)

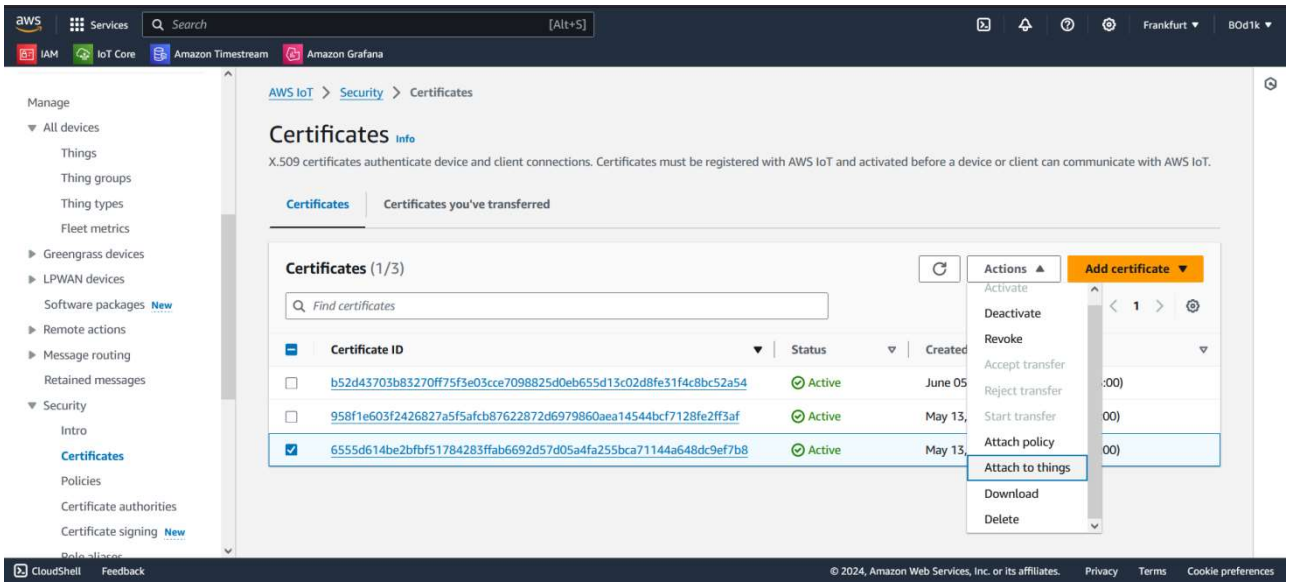

Рисунок 4.8 – Додавання сертифікату до політики та «речі»

На цьому процес налаштування у ядрі AWS IoT завершено, переходимо до реалізації зв’язування контролера та комп’ютера.

Для реалізації завдання потрібно написати програму на мові Python для того, щоб зчитувати температуру з двох датчиків температури та надсилати з нашого девайсу на ядро AWS IoT. Написану програму потрібно завантажити до вже створеної теки “aws2”, також не забуваємо про кореневий спільний файл «AmazonRootCA1.pem». (див. рис. 4.9) Це файл кореневого сертифіката від Amazon, який є частиною системи управління цифровими сертифікатами. Він забезпечує безпеку комунікацій між клієнтами та серверами Amazon, підтверджуючи автентичність серверів.

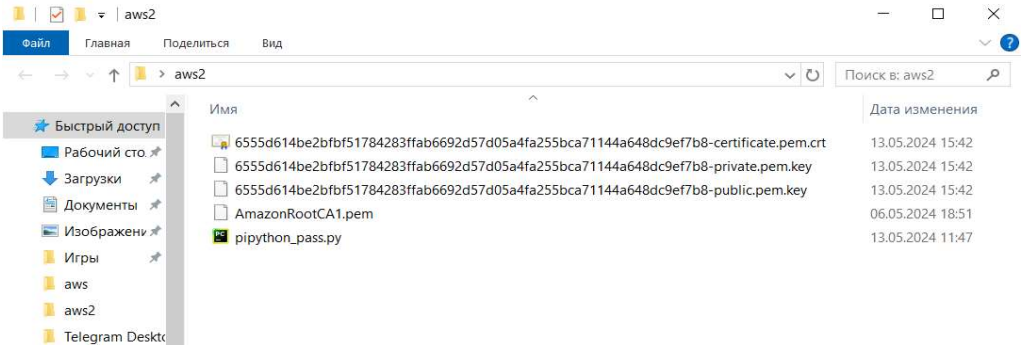

Рисунок 4.9 – Фінальний вигляд папки перед відправкою на контролер

Написаний код на Python:

– налаштовує MQTT-клієнт для підключення до AWS IoT з використанням шифрування та аутентифікації;

– читає дані з двох цифрових сенсорів температури DS18B20;

– безперервно публікує отримані дані на AWS IoT кожні 3 секунди.

Реалізовуємо підключення до нашого серверу та контролеру за допомогою Bitvise SSH Client та завантажуємо усю створену теку «aws2» в кореневий файл контролера для роботи системи (ри. 4.10).

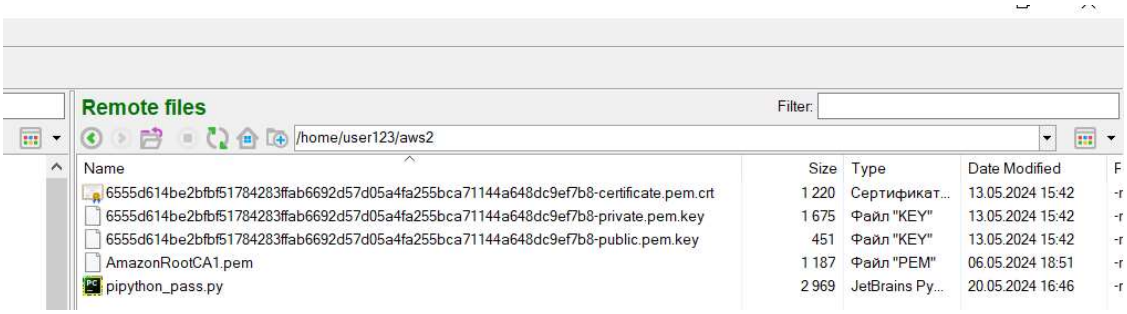

Рисунок 4.10 – Зображення робочого вікна теки

Для правильної роботи програми виконуємо налаштування контролера за допомогою деяких команд і переходимо до її запуску з консолі.

Переходимо у теку з програмою та запускаємо код на Python. Бачимо, що програма запустилася і відображає нам температуру навколишнього середовища з двох сенсорів а також повідомляє, що повідомлення на AWS IoT відправляється у тему, яка вказано у коді програми, а саме «device/data». (див. рис. 4.11) Це

означає, що вдалося успішно під’єднати контролер до нашого облікового запису.

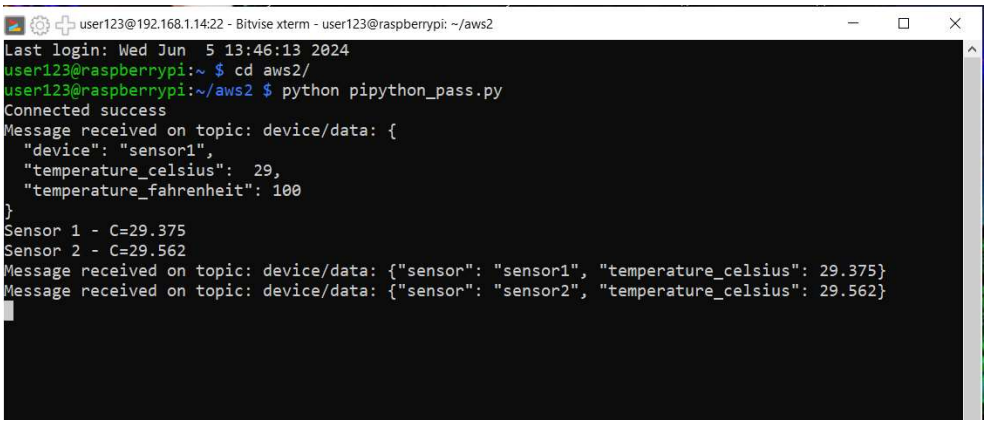

Рисунок 4.11 – Робоча консоль з даними

Далі переходимо до облікового запису та підписуємося на тему, яку вказали у коді програми щоб переконатися у тому, що дані передаються на віддалену хмару.

Переходимо у вікно «MQTT test client» та вводимо тему. Натискаємо кнопку «підписатись» та бачимо відображення даних, які надходять з контролера на ядро AWS IoT, обидва сенсора відправляють дані про температуру. (див. рис. 4.12) На цьому етап з’єднання та віправлення даних вважаємо завершеним.

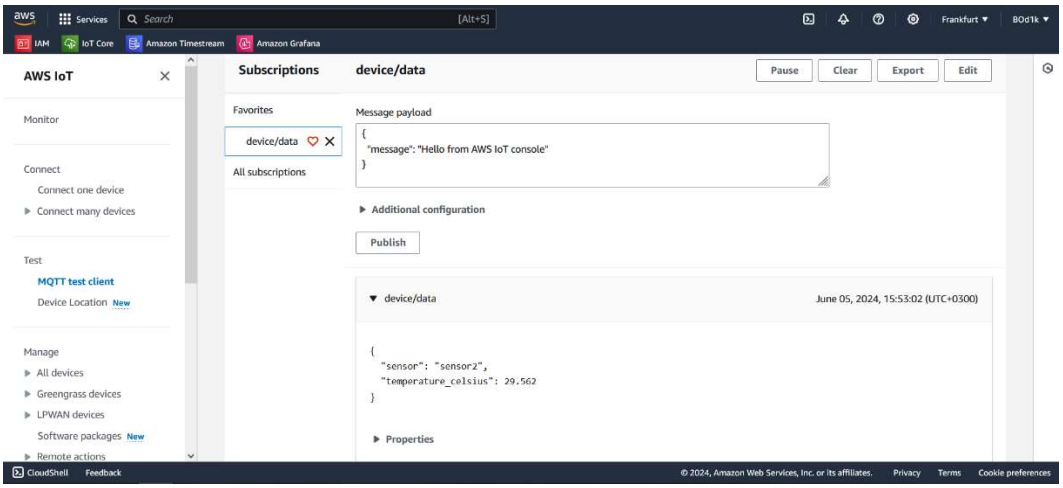

Рисунок 4.12 – Результат відображення даних

4.4 Зберігання даних та їх візуалізація за допомогою Amazon Timestream та Grafana

Отримані дані потрібно десь зберігати для їх подальшої візуалізації а також

аналізу.

Зупинимося на базі даних від Amazon Timestream. Це спеціально розроблена повністю керована база даних тимчасових рядів, яка допомагає заощаджувати час користувача: в ній не потрібно виконувати трудомісткі інфраструктурні завдання накшталт встановлення, оновлення, зберігання, копіювання та резервного копіювання.

Amazon Timestream пропонує повністю керовані спеціалізовані двигуни баз даних тимчасових рядів для робочих навантажень: від запитів з малою затримкою до отримання великомасштабних даних. У сервісі доступні вбудовані функції аналізу часових рядів, які допомагають виявляти тенденції та закономірності майже в реальному часі. (див. рис. 4.12)

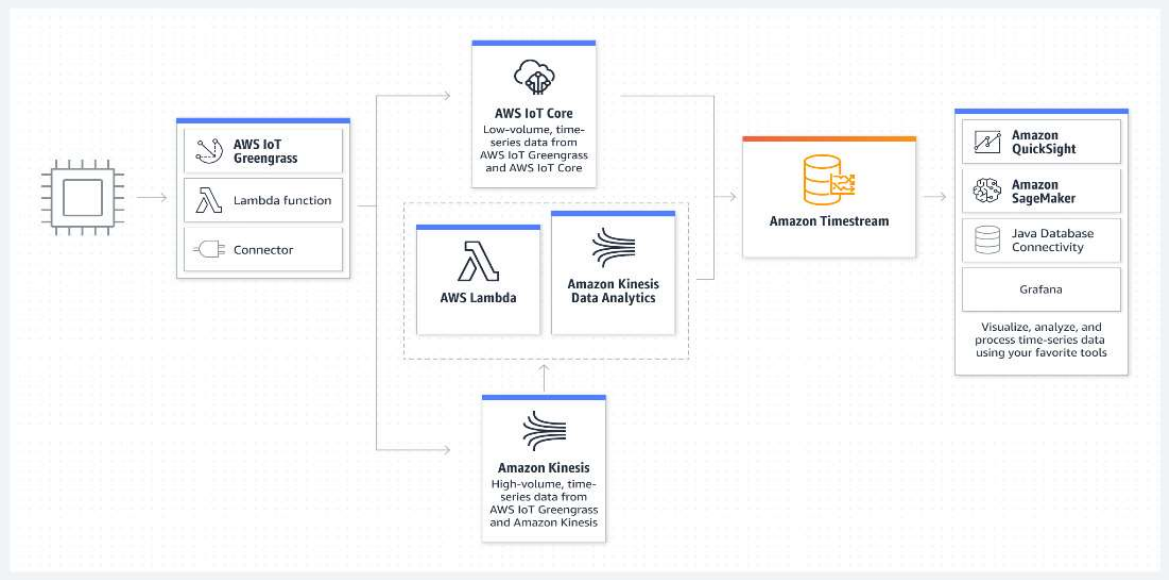

Рисунок 4.13 – Наглядне відображення додатку інтернету речей

4.4.2 Створення таблиці в Amazon Timestream та IoT правило

Перше, що нам потрібно виконати, це створити базу даних тимчасового потоку та таблицю для зберігання вхідних даних. Після цього ми створимо правило IoT для маршрутизації телеметрії в таблицю потоків часу.

Переходимо у вікно Amazon Timestream та натискаємо «створення бази даних» (див. рис. 4.14).

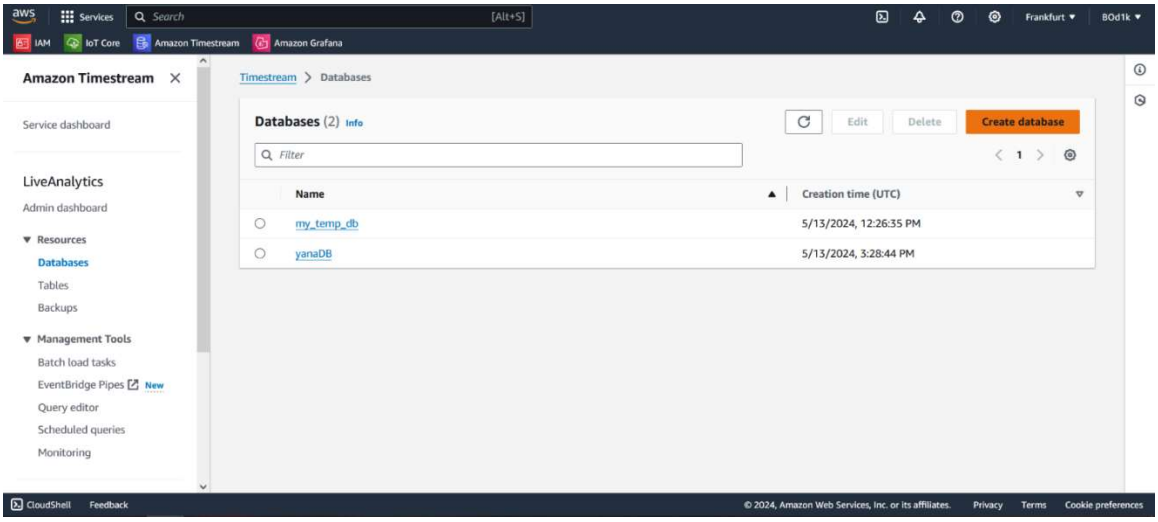

Рисунок 4.14 – Процес створення бази даних

Надаємо ім’я нашій базі даних, у нашому випадку це “my_temp_db” та натискаємо «створити базу даних». Тепер, коли в нас є база даних, ми можемо створити таблицю, яка буде зберігати вхідні дані.

Переходимо на створену базу даних, натискаємо «таблиці» та «створення таблиці». (див. рис. 4.15)

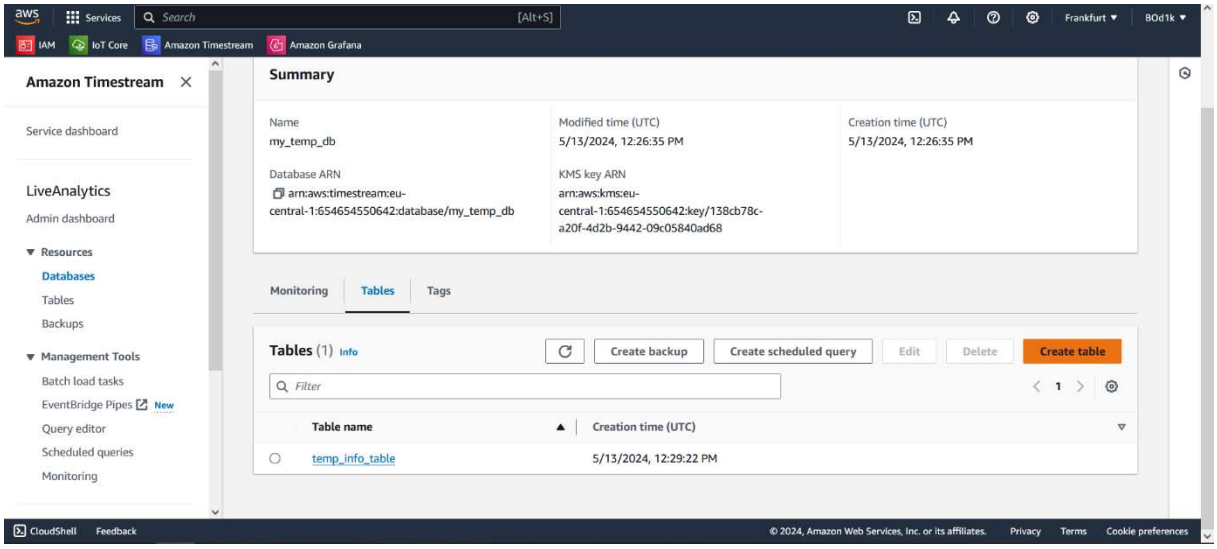

Рисунок 4.15 – Процес створення таблиці для зберігання даних

Називаємо нашу таблицю “temp_info_table” та вказуємо їй, як довго дані будуть зберігатися у пам’яті та коли вони будуть видалені з диску та натискаємо «створити таблицю».

Тепер, коли маємо демонстраційну таблицю об’єктів, можемо створити правило у ядрі AWS IoT, яке буде направляти вхідні дані в таблицю. Механізм

основних правил IoT дуже потужний і дозволяє нам обирати, оцінювати, доповнювати та перетворювати вхідні дані, а потім спрямовувати дані в нижчестоящі служби, використовуючи одне або декілька дій правила.

Повертаємося у ядро IoT, переходимо до «маршрутизації повідомлень», там обираємо «правила» та натискаємо «створити правило» (див. рис. 4.16)

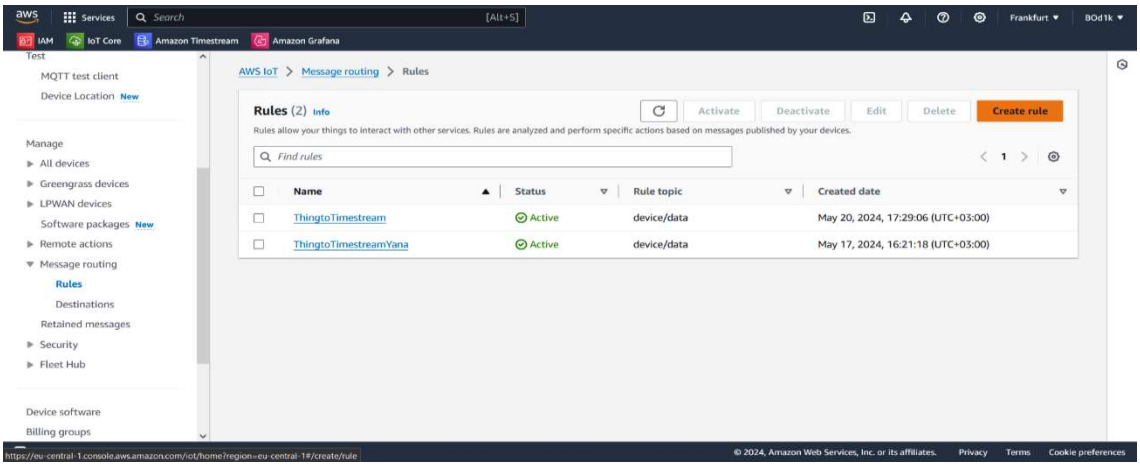

Рисунок 4.16 – Процес створення правила

Називаємо наше правило “ThingToTimestream”, переходимо далі, у наступному вікні ми використаємо простий вибір інструкцій та наскаємо «далі». (див. рис. 4.17)

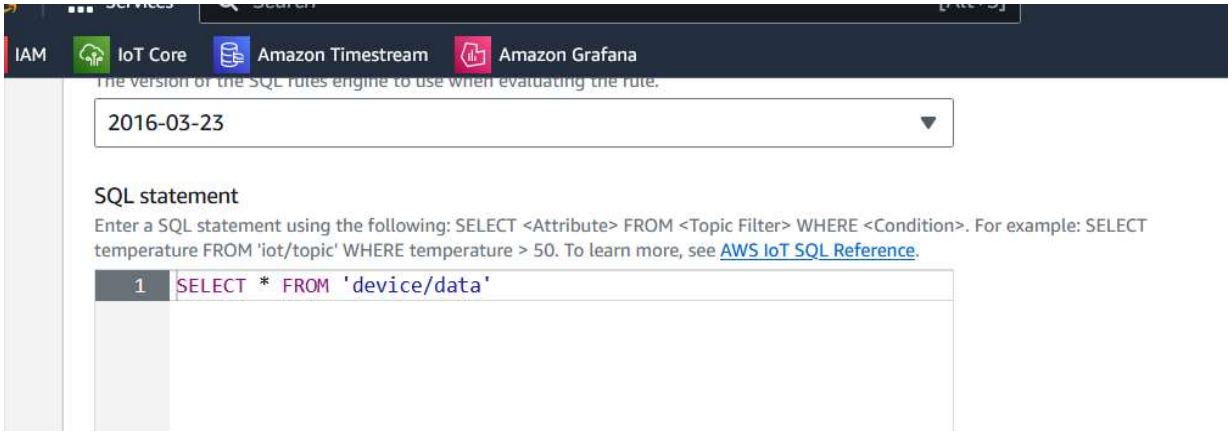

Рисунок 4.17 – Приклад використання інструкція для подальшого створення правила

Далі нам потрібно обрати дію потоку часу, а саме “Timestream table”, та налаштувати розміри таблиці бази даних і значення позначок часу. Найголовніше, це правильно використати змінні підстановки для отримання значень

безпосередньо з вхідного корисного навантаження (див. рис. 4.18).

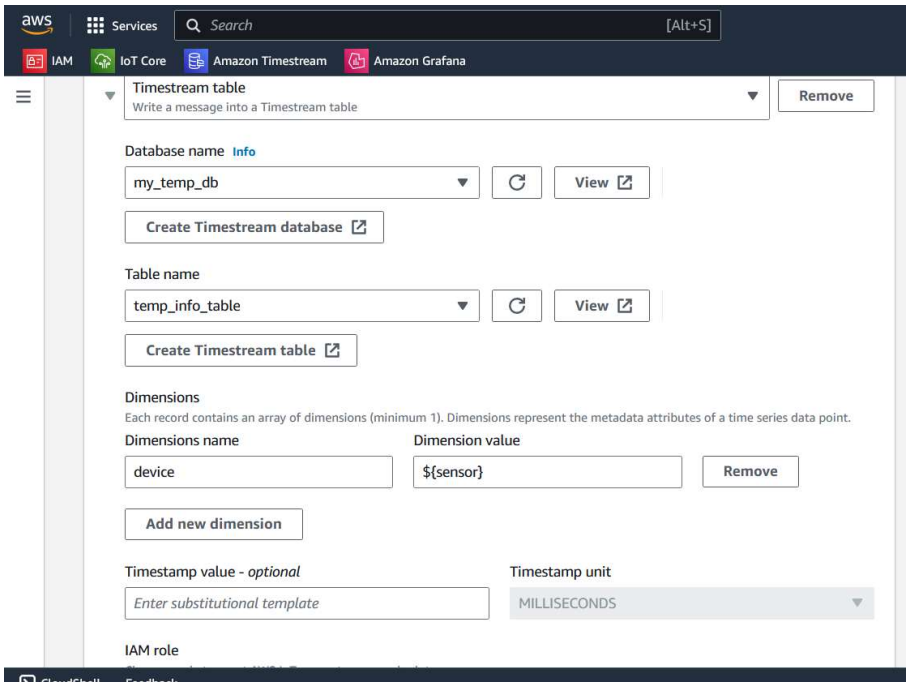

Рисунок 4.18 – Налаштування правила

Також нам потрібно створити IAM роль, яка дозволить правилу записувати в таблицю потоку часу дані. У вікні натискаємо «створити нову роль», надаємо їй назву “thing_to_TimestreamRole” та настикаємо «створити». Після усіх виконаних дій натискаємо «далі» та перевіряємо усі дані, які ми використали для створення правила та обираємо «створити».

Тепер, коли наше нове правило активне, ми можемо виконати запит у бази даних, щоб перевірити, чи правильно зберігаються дані та чи надходять вони туди. Повертаємось у Amazon Timestream, обираємо «засоби управління» та переходимо в «редактор запитів». Вводимо відповідну команду та натискаємо кнопку «запустити» (див. рис. 4.19 та 4.20).

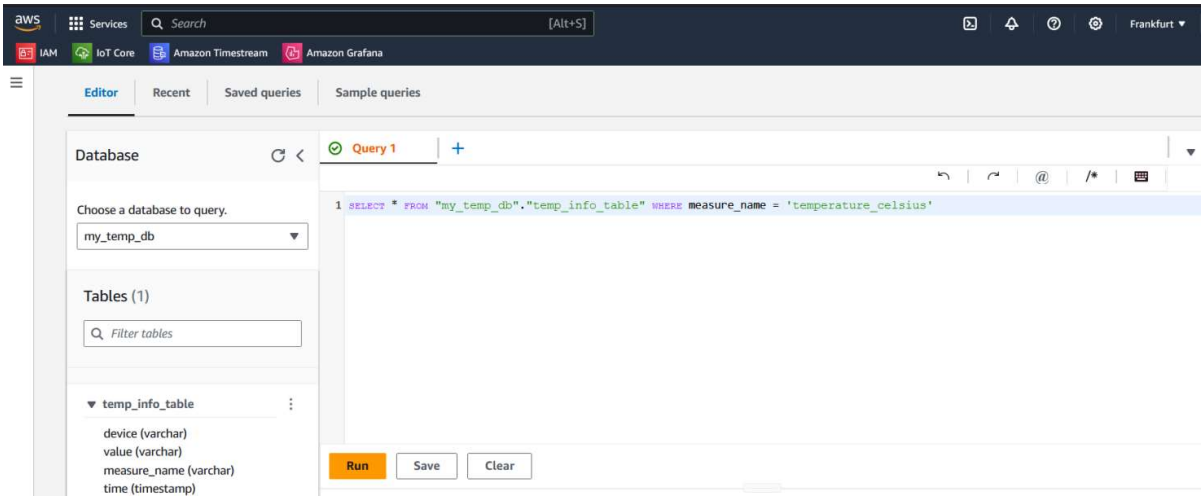

Рисунок 4.19 – Відображення запиту для отримання даних

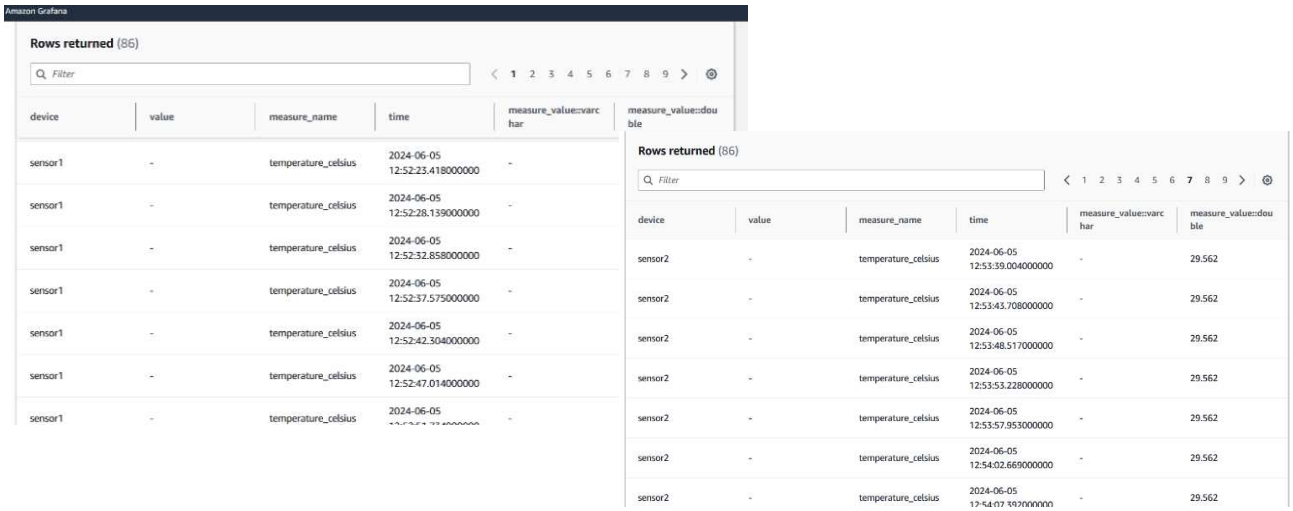

Рисунок 4.20 – Покази з обох датчиків

4.4.3 Візуалізація даних за допомогою Amazon Grafana

А зараз, коли ми зберігаємо результати зчитування показників з датчиків в базі даних потоку часу, ми можемо створити інформаційну панель для відображення даних у вигляді простого часового ряду, а саме лінійний графік.

Результати вимірювання зберігаються у розділі «Rows returned», де показують час, коли було зроблено вимірювання та значення.

Реалізуємо вивід цих значень в режимі реального часу за допомогою сервісу Grafana.

Amazon Managed Service for Grafana (AMG) – це повністю керований та безпечний сервіс для візуалізації даних, який дозволяє клієнтам швидко запитувати, зіставляти та візуалізувати операційні метрики, логи та трасування

своїх додатків із кількох джерел. AMG заснований на проекті Grafana з відкритим кодом.

Для того щоб почати роботу з Grafana , переходимо у сервіс “Amazon Grafana ” та натискаємо кнопку «створити робочу область», після чого надаємо їй ім’я “grafana_workspace” та натискаємо «далі». На наступному етапі треба обрати «AWS IAM Identity Center» та зареєструвати користувача, щоб ця функція була активована. Цьому новому користувачу надаємо права адміністратора для доступу в консолі для подальших дій. (див. рис. 4.21)

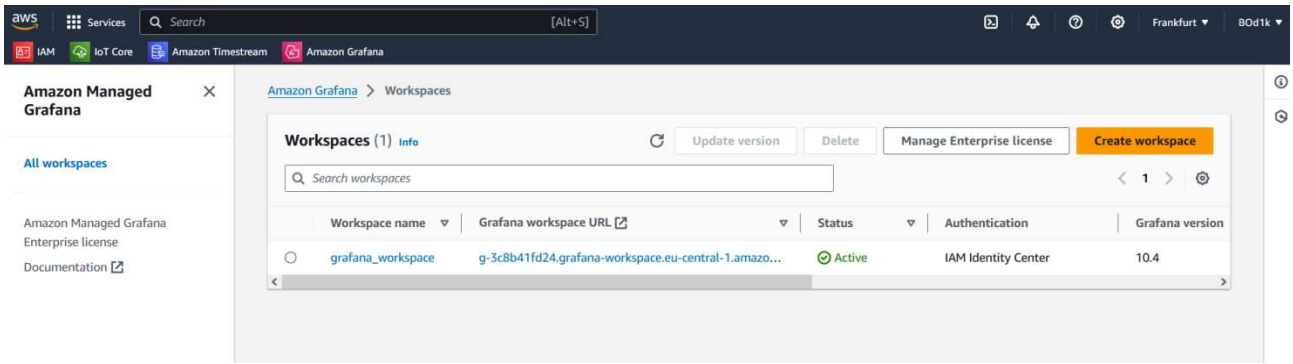

Рисунок 4.21 – Створена робоча область

Отримуємо посилання, за допомогою якого переходимо в сервіс та авторизуємося через дані створеного користувача та з’являємося у головному меню додатку. (див. рис. 4.22)

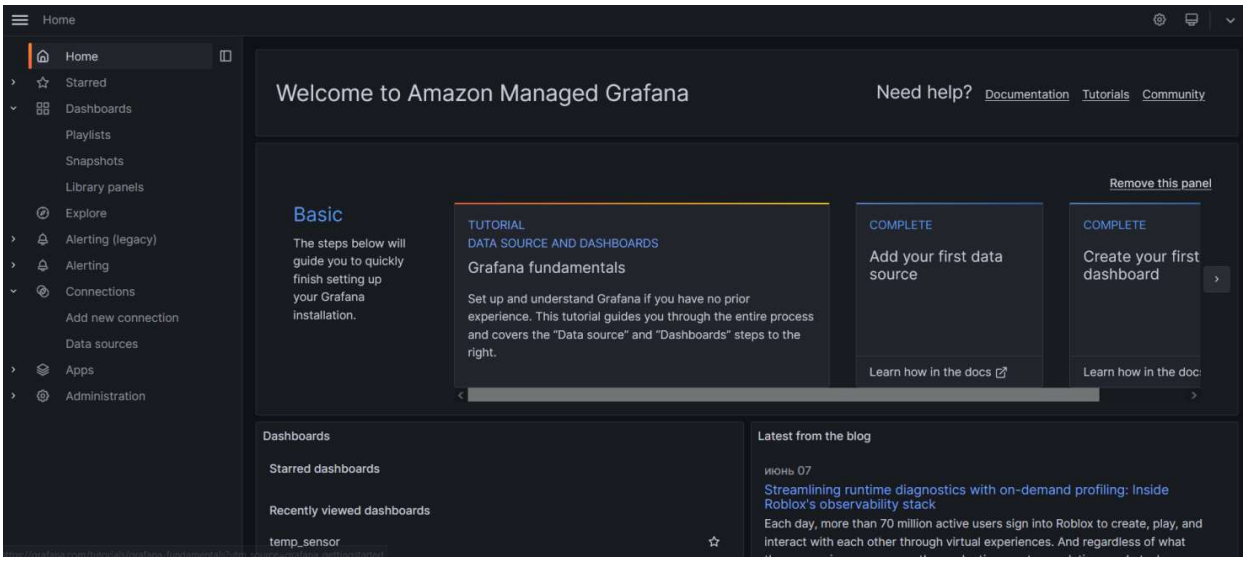

Рисунок 4.22 – Головне вікно Amazon Grafana

Завершальним етапом буде додавання джерела даних для нашої таблиці.

Додаємо джерело даних, а саме “Amazon Timestream”. У цьому вікні вибираємо регіон і перевіряємо з’єднання. Обираємо створену базу даних, таблицю та міру та натискаємо «save & test». (див. рис. 4.23)

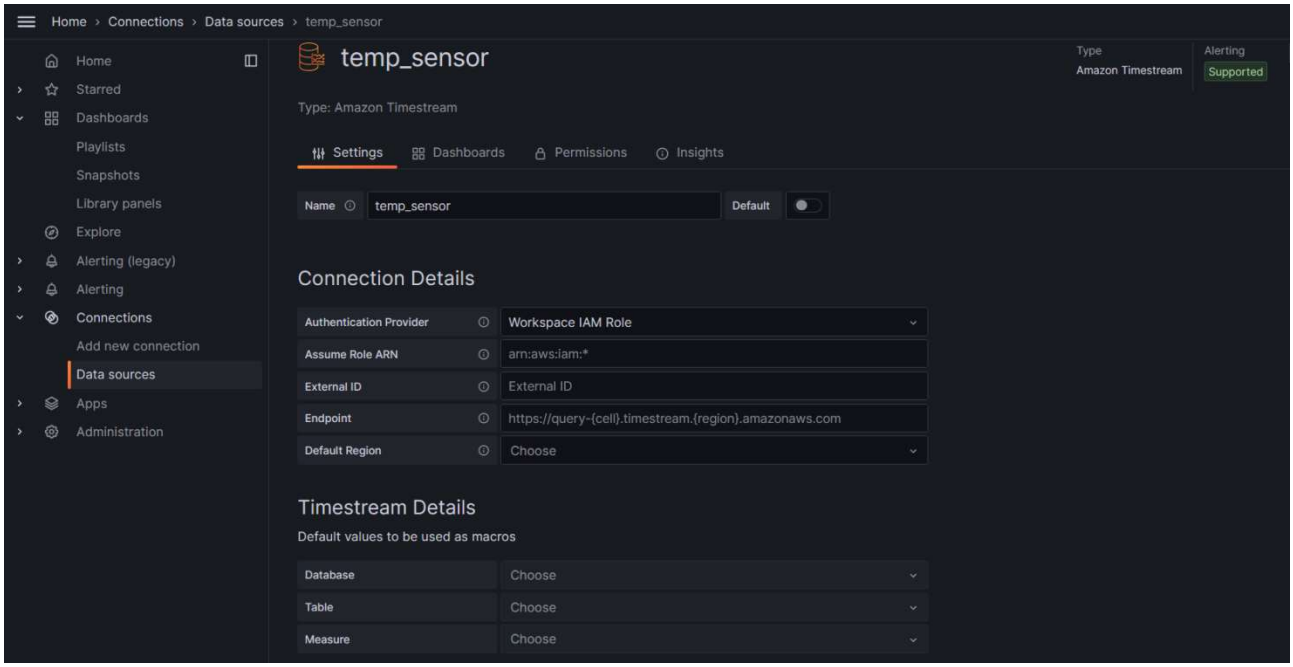

Рисунок 4.23 – Налаштування джерела даних

Тепер, коли у нас є джерело даних ми можемо створити нову інформаційну панель. Обираємо «додати нову панель», далі «Amazon Timestream», та пишемо запит для панелі, а у нашому випадку запитів 2, адже для кожного сенсора має бути свій графік, тому нижче буде представлені запити для кожного сенсора. (див. рис. 4.24 та 4.25)

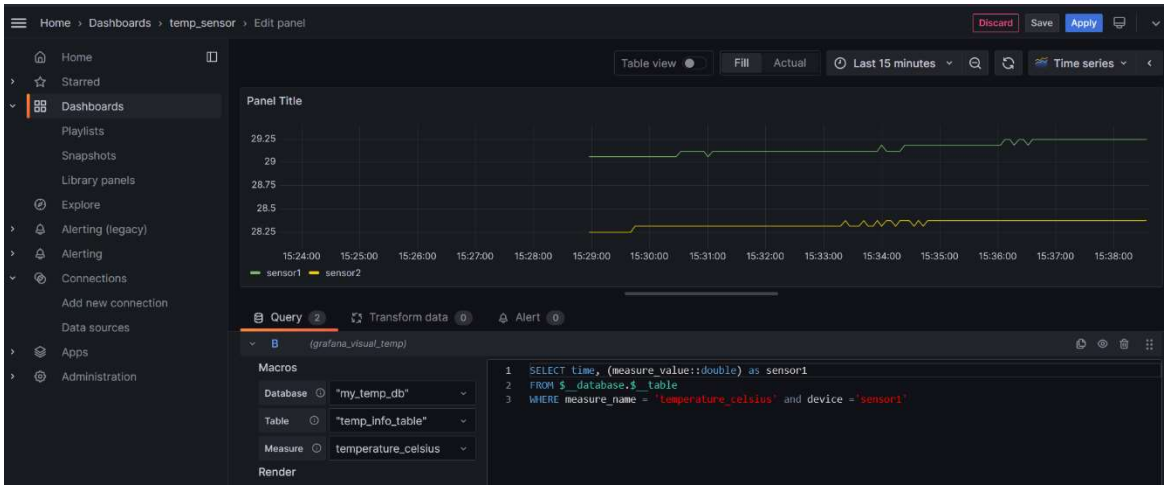

Рисунок 4.24 – Результат запиту для датчику «sensor1»

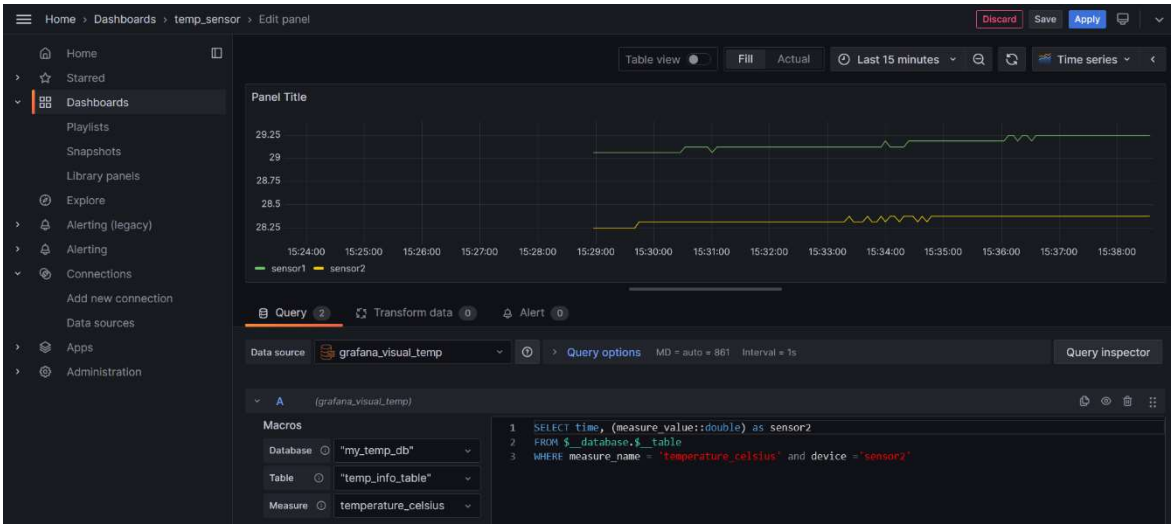

Рисунок 4.25 – Результат запиту для датчику «sensor2»

Результатом роботи та підключення до серверу є передача даних в режимі реального часу з двох датчиків та їх графічна візуалізація з коректними значеннями у вигляді двох графіків.

Отримані дані можна переформатовувати у потрбіний для вас вигляд. (див. рис. 4.26)

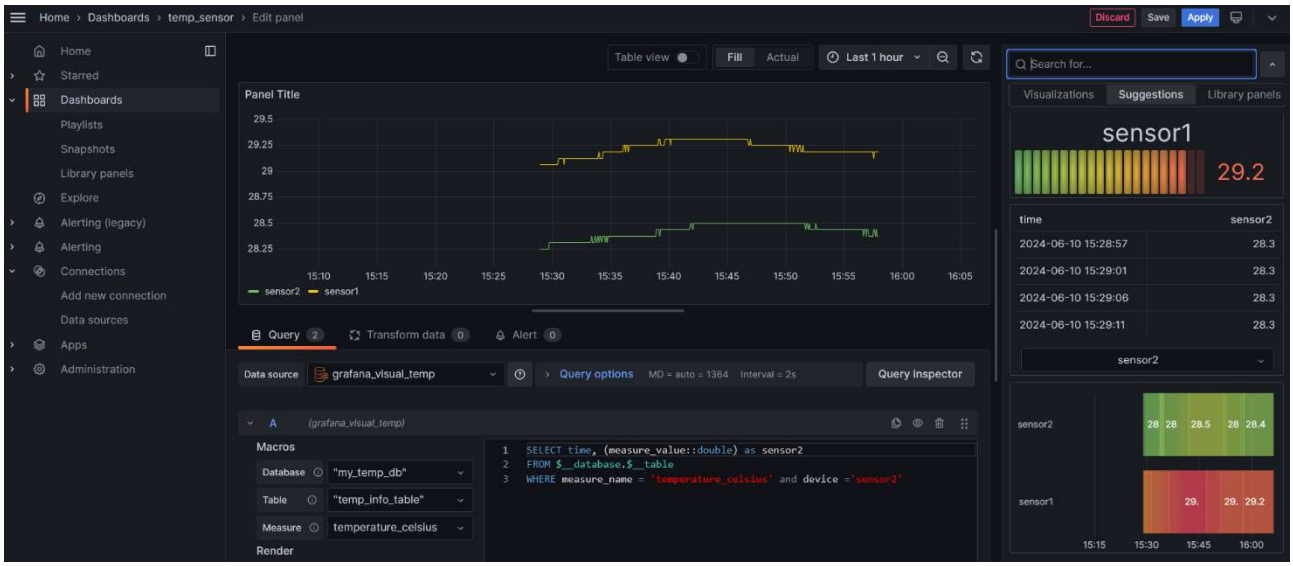

Рисунок 4.26 – Відображення варіантів форматування

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було створено кіберфізичної системи моніторингу температури для розумного будинку. Були визначені умови використання та застосування системи. Вона виконує функції збору даних з температурних датчиків та візуалізує їх.

Отримана візуалізація забезпечує зручний спосіб моніторингу.

Виконані в рамках кваліфікаційної роботи завдання та досягнуті результати повністю відповідають сучасному рівню розвитку технологій Інтернету речей та кіберфізичних систем. Використання мікроконтролера Rasbperry Pi у поєднанні з платформою AWS IoT дозволило створити економічно вигідне та надійне рішення. Система може легко інтегруватися з іншими компонентами розумного будинку, що робить її універсальним інструментом для підвищення комфорту та ефективності управління.

Ця робота робить вагомий внесок у розвиток технологій Інтернету речей (IoT) та кіберфізичних систем, демонструючи практичне застосування доступних компонентів для створення ефективних та надійних рішень. Використання AWS IoT відкриває широкі можливості для інтеграції з іншими сервісами та системами, що сприяє подальшому розширенню функціональності розумних будинків.

Розроблена система може бути базою для більш складних систем моніторингу та управління мікрокліматом, що включатимуть додаткові функції та можливості. Майбутні дослідження можуть бути спрямовані на інтеграцію системи з більшою кількістю датчиків, впровадження технологій штучного інтелекту для більш точного прогнозування та автоматизації управління, а також підвищення енергоефективності та безпеки.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. IoT опис URL: https://aws.amazon.com/what-is/iot/

2. Мікрокомп’ютер Raspberry Pi URL: https://networkdiscount.com.ua/ua/p1351615191-mikrokompyuter-raspberry-model.html

3. Технічні характеристики Raspberry Pi 4 URL: https://www.kosmodrom.ua/odnoplatnyy-kompyuter-raspberry/raspberry-pi-4-model-b-4gb.html

4. DS18B20 – повний опис датчика та його можливостей URL: https://mpr-kip.com/ua/p829821687-germetichnyj-datchik-temperatury.html

5. Підключення датчику URL: https://qazf.com.ua/ds18b20-arduino/

6. Raspberry Pi для розумного будинку (Smart Home) URL: https://evo.net.ua/raspberry-pi-dlya-avtomatizatsii-umnogo-doma-smart-home/

7. Використання контролеру в системі розумного будинку URL: https://livolo.in.ua/kontrolery-dlia-rozumnoho-budynku/

8. Опис контролера BANANA URL: https://miniboard.com.ua/boards/418-banana-pi-m64.html

9. Елементи контролеру Raspberry Pi 4 URL: https://evo.net.ua/izuchaem-raspberry-pi.-chast-1.-znakomstvo/

10. Елементи контролеру Raspberry Pi 4 URL: https://networkdiscount.com.ua/ua/p1072298159-mikrokompyuter-raspberry-model.html

11. Хамбракен, Д. Комп'ютерні мережі: Пер. з англ. / Д. Хамбракен. – М.: ДМК Прес, 2004. – 449 с.

12. Мережева академія cisco – https://www.netacad.com/

13. Як працює Інтернет речей. URL: https://osvita.home.cx.ua/ukraincyam/shho-take-internet-of-things-ta-yak-vono-pracyuie.html

14. Інтернет речей. URL: http://surl.li/aezfc

15. Опис Bitvise SSH Client. URL: https://www.softportal.com/software-41688-

bitvise-ssh-client.html

16. Опис загальної роботи Grafana. URL: https://grafana.com/docs/grafana-cloud/monitor-infrastructure/aws/monitor-svcs/ (дата звернення 24.05.2024)

17. Методичні рекомендації до виконання кваліфікаційної роботи бакалавра студентами галузі знань 12 інформаційні технології спеціальності 123 комп'ютерна інженерія / Л.І. Цвіркун, В.В. Гнатушенко, С.М. Ткаченко ; М-во освіти і науки України, НТУ "ДП" – Дніпро: 2023. – 29 с

18. Amazon Timestream. URL: https://aws.amazon.com/timestream/?nc1=h_ls (дата звернення 30.05.2024)

19. Michel Bakni, Yudith Cardinale, Luis Manuel Moreno. An Approach to Evaluate Network Simulators: An Experience with Packet Tracer // Revista VenezoMANCOMPa de Computación. — 2018-06. — Т. 5. — С. 29 – 36.

20. Mykyychuk M. M., Stadnyk B. I., Yatshyshyn S. P., Lutsyk Ya. T. (2017). Measuring Smart Means for Cyber-Physical Systems. Measuring Technology and Metrology: Issue 77, pp. 3–17.

21. AWS IoT - Device data to dashboard. URL: https://www.youtube.com/watch?v=z8T4hAERuOg&ab_channel=AWSIoT

22. AWS IoT Core with Raspberry Pi4. URL: https://www.youtube.com/watch?v=adKuyckikuw&ab_channel=BINARYUPDATES

23. Smart House сайт компанії. URL: https://www.smarthouse.ua/o_nas.html

24. Сайт Amazon IoT Core. URL: https://aws.amazon.com/iot-core/?nc1=h_ls

ДОДАТОК А

Текст Python коду мікроконтролера для моніторингу температури

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

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ НАЛАШТУВАННЯ КІБЕРФІЗИЧНОЇ МЕРЕЖІ МОНІТОРИНГУ ТЕМПЕРАТУРИ

Текст програми

804.02070743.24005-01 12 01

Листів

2024

АНОТАЦІЯ

Дана програма містить в собі програмний код для мікроконтролеру кіберфізичної системи моніторингу температури в розумному будинку.

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

Програма написана мовою Python, відлагоджена і призначена для застосування на пристроях, які мають доступ до Wi-Fi.

ЗМІСТ

```
# -*- coding: utf-8 -*-
import time
import paho.mqtt.client as mqtt
import ssl
import json
import RPi.GPIO as GPIO
import os
import glob

# AWS IoT Endpoint, you can find this in the AWS IoT console
AWS_IOT_ENDPOINT       =       "a2zu7yyfj2hovi-ats.iot.eu-central-
1.amazonaws.com"
PORT = 8883  # MQTT port for secure communication

# Path to your certificate and private key files
CERT_FILE                                                          =
"6555d614be2bfbf51784283ffab6692d57d05a4fa255bca71144a648dc9ef7b8-
certificate.pem.crt"
KEY_FILE                                                           =
"6555d614be2bfbf51784283ffab6692d57d05a4fa255bca71144a648dc9ef7b8-
private.pem.key"
CA_FILE = "AmazonRootCA1.pem"  # Root CA certificate

GPIO.setmode(GPIO.BCM)
GPIO.setup(4, GPIO.OUT)


def on_connect(client, userdata, flags, rc, properties):
    if rc == 0:
        print("Connected success")
        # Subscribe to a topic when connected
        client.subscribe("device/data")
    else:
        print(f"Connected fail with code {rc}")


def on_message(client, userdata, message):
    print("Message received on topic: " + message.topic + ": " +
str(message.payload.decode()))


# Create a client instance
client = mqtt.Client(mqtt.CallbackAPIVersion.VERSION2)

# Set the callbacks
client.on_connect = on_connect
client.on_message = on_message

# Set the certificates and key
client.tls_set(ca_certs=CA_FILE,               certfile=CERT_FILE,
```

```
keyfile=KEY_FILE, cert_reqs=ssl.CERT_REQUIRED,
               tls_version=ssl.PROTOCOL_TLSv1_2, ciphers=None)

client.tls_insecure_set(True)

# Connect to the AWS IoT endpoint
client.connect(AWS_IOT_ENDPOINT, port=PORT, keepalive=60)

# Start the MQTT loop
client.loop_start()

# Publish a message
client.publish("test_topic", "Hello from Kazakov")

os.system('modprobe w1-gpio')
os.system('modprobe w1-therm')

base_dir = '/sys/bus/w1/devices/'
device1_path = glob.glob(base_dir + '28-57e77d1f64ff')[0]  # Путь к
первому датчику
device2_path = glob.glob(base_dir + '28-66e57d1f64ff')[0]  # Путь ко
второму датчику


def read_temp_raw(device_path):
    with open(device_path + '/w1_slave', 'r') as f:
        valid, temp = f.readlines()
    return valid, temp


def read_temp(device_path):
    valid, temp = read_temp_raw(device_path)

    while 'YES' not in valid:
        time.sleep(0.2)
        valid, temp = read_temp_raw(device_path)

    pos = temp.index('t=')
    if pos != -1:
        temp_string = temp[pos + 2:]
        temp_c = float(temp_string) / 1000.0
        return temp_c


while True:
    c1 = read_temp(device1_path)
    c2 = read_temp(device2_path)

    print('Sensor 1 - C={:,.3f}'.format(c1))
    print('Sensor 2 - C={:,.3f}'.format(c2))
```

```
    client.publish("device/data", payload=json.dumps(
        {"sensor": "sensor1", "temperature_celsius": c1}), qos=0,
retain=False)
    client.publish("device/data", payload=json.dumps(
        {"sensor": "sensor2", "temperature_celsius": c2}), qos=0,
retain=False)
    time.sleep(3)
```