Хмельницький національний університет
Факультет інформаційних технологій
Кафедра кібербезпеки

КВАЛІФІКАЦІЙНА РОБОТА

Ратушняка Арсена Юрійовича

на здобуття ступеня вищої освіти Бакалавра

Система захищеного комплексу віддаленого керування опалювальним котлом на базі IoT контролерів

Галузь знань 12 – Інформаційні технології

Спеціальність 123 – Комп'ютерна інженерія

Освітня програма Програмування та захист комп`ютерних систем і мереж

Шифр КРБКІ. 2001121.20.01.05 ПЗ

Виконав студент 4 курсу група КІ1-20-1 Арсен РАТУШНЯК

Керівник канд. техн. наук, доцент Микола СТЕЦЮК

Нормоконтролер старший викладач Сергій МОСТОВИЙ

До захисту допускаю:
Завідувач кафедри кібербезпеки Юрій КЛЬОЦ

2024 р.

Хмельницький 2024

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій

Кафедра Кібербезпеки

Рівень вищої освіти Бакалавр

Галузь знань 12 – Інформаційні технології

Спеціальність 123 – Комп'ютерна інженерія

Освітня програма Програмування та захист комп'ютерних систем і мереж

ЗАТВЕРДЖУЮ

Завідувач кафедри кібербезпеки

Юрій КЛЬОЦ

15 лютого 2024 р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Ратушняку Арсену Юрійовичу

1 Тема роботи Система захищеного комплексу віддаленого керування опалювальним котлом на базі IoT контролерів

Керівник роботи

Затверджено наказом ректора університету від 15 лютого 2024 № 8

2 Строк подання студентом кваліфікаційної роботи на кафедру

3 Вихідні дані до роботи

Система захищеного комплеску віддаленого керування опалювальним котлом на базі IoT котнтролерів

4 Зміст пояснювальної записки (перелік питань, які потрібно розробити)

1 Теоретичні Основи Мережевих Технологій та Інтернету Речей

2 Проектування мережі

3 Моделювання мережі котельні

5 Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

1 Алгоритм роботи сценаріїв контроллеру

2 Логічна топологія мережі

3 Схема підключення пристроїв до пінів

6 Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Мостовий С.В., старший викладач кафедри кібербезпеки		

7 Дата видачі завдання 16 лютого 2024 р.

КАЛЕНДАРНИЙ ПЛАН

Назва етапів (розділів) кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
Вибір і затвердження теми кваліфікаційної роботи	Січень-Лютий	
Ознайомлення з предметною областю	Лютий	
Дослідження існуючих рішень	Лютий	
Постановка задачі	Березень	
Визначення загальних принципів рішення задачі	Березень	
Деталізація принципів рішення залачі	Квітень	
Розробка проєктних рішень	Квітень	
Апробація проєктних рішень	Травень	
Оформлення пояснювальної записки згідно вимог	Травень	
Оформлення графічної частини	Червень	
Захист КР	Червень	

Студент _____ Арсен РАТУШНЯК

Керівник кваліфікаційної роботи _____ Микола СТЕЦЮК

Формат	Зона	Позиц.	Позначення	Найменування	Кільк.	Прим.
A4		1	КРБКІ.2001121.20.01.05 ПЗ	Система захищеного	56	
				комплексу віддаленого		
				керування опалювальним		
				котлом на базі IoT		
A2		2	КРБКІ.2001121.20.01.05 Е8	Алгоритм роботи	1	
				сценаріїв контролеру		
A2		3	КРБКІ.2001121.20.01.05 Е8	Логічна топологія мережі	1	
A2		4	КРБКІ.2001121.20.01.05 Е8	Схема підключення	1	
				пристроїв до пінів	1	

КРБКІ.2001121.20.01.05 ВП

Зм.	Арк.	№ Докум.	Підп.	Дата
Розробив		Ратушняк А.Ю.		19.06.24
Перев.		Стецюк М.В.		19.06.24
Н. контр.		Мостовий С.В.		19.06.24
Затв.		Кльоц Ю.П.		19.06.24

Система захищеного комплексу віддаленого керування опалювальним котлом на базі IoT

Відомість проєкту

Літера	Аркуш	Аркушів
в	1	1

ХНУ, КІ1-20-1

АНОТАЦІЯ

ІНФОРМАЦІЙНО-ТЕЛЕКОМУНІКАЦІЙНА СИСТЕМА, МАРШРУТИЗАТОР, РОЗПОДІЛЕНА СИСТЕМА, ВИКОРИСТАННЯ IoT, ОДНОПЛАТНИЙ ПК, МОНІТОРИНГ IoT В РЕАЛЬНОМУ ЧАСІ.

Записка: 73 стор., 18 рис., 40 джерел.

Мета роботи — розробка мережі для автоматизації роботи мережі котельні.

Об'єкт дослідження — Система захищеного комплексу віддаленого керування опалювальним котлом.

Предмет дослідження — сукупність засобів програмно-технічного характеру, які можуть бути спрямовані на автоматизацію роботи опалювального котла.

Методи дослідження — Застосовано методи теоретичного узагальнення для опису предметної області дослідження; аналізу та синтезу для оцінки захищеності наявної розподіленої інформаційно-телекомунікаційної; системного аналізу та експерименту для розробки моделі системи та налаштування пристроїв.

Результати — була успішно створена мережа котельні, яка об'єднує різні мережеві компоненти та пристрої Інтернету речей (IoT) для автоматичного моніторингу та контролю температури.

19.06.2024

ЗМІСТ

ВСТУП

Комплекси віддаленого керування на базі Інтернету речей (IoT) зробили значний внесок у організацію роботи та автоматизацію систем опалення та охолодження. Традиційні методи керування котлами, які базуються на ручному втручанні та простих автоматичних системах, стають менш ефективними та гнучкими в порівнянні з сучасними рішеннями на основі IoT. Сучасний світ стрімко рухається до цифрової трансформації, і це стосується всіх аспектів життя, включаючи управління енергією та ресурсами.

Захищені системи керування котлами опалення на основі IoT пристроїв пропонують значно більше можливостей для підвищення ефективності, безпеки та зручності. Завдяки інтеграції сенсорів, автоматизованих алгоритмів та можливості віддаленого моніторингу, такі системи дозволяють не лише оптимізувати споживання енергії, але й мінімізувати ризики аварійних ситуацій.

Інтернет речей дозволяє створити мережу взаємопов'язаних пристроїв, які можуть обмінюватися даними та здійснювати аналіз у режимі реального часу. У випадку опалювальних систем, це означає, що параметри роботи котла можуть постійно моніторитися, а дані аналізувати для виявлення можливих несправностей або оптимізації режимів роботи. Такий підхід значно підвищує надійність та безпеку роботи котлів, що є особливо важливим у контексті як побутових, так і промислових застосувань.

Таким чином, впровадження захищених систем керування котлами на основі IoT пристроїв стає невід'ємною частиною сучасної енергетики. Це не тільки сприяє підвищенню ефективності та безпеки, але й відкриває нові можливості для дистанційного управління та автоматизації процесів, що робить ці системи надзвичайно актуальними в умовах постійного розвитку технологій та зростання вимог до енергозбереження.

Крім того, сучасні IoT-системи дозволяють інтегрувати управління опаленням з іншими смарт-системами будинку чи підприємства, створюючи

єдину екосистему керування. Це відкриває можливості для комплексного підходу до енергозбереження, коли всі системи працюють злагоджено для досягнення загальної мети - зниження споживання енергії та підвищення комфорту. Наприклад, опалювальні системи можуть взаємодіяти з системами освітлення, вентиляції та кондиціонування, щоб забезпечити оптимальні умови в приміщенні з мінімальними витратами енергії.

Таким чином, розвиток IoT-технологій у сфері керування системами опалення представляє собою важливий крок у напрямку створення більш ефективних, безпечних та зручних енергетичних систем. Це є відображенням загальної тенденції до автоматизації та цифровізації, що охоплює всі аспекти нашого життя.

1 ТЕОРЕТИЧНІ ОСНОВИ МЕРЕЖЕВИХ ТЕХНОЛОГІЙ ТА ІНТЕРНЕТУ РЕЧЕЙ

1.1 Поняття “мережа” та її функції

Комп’ютерна мережа – це група взаємопов’язаних пристроїв: комп’ютери, сенсори, контролери та інші пристрої Інтернету речей, які об’єднані їх спільнім інтерфейсом з метою моніторингу та управління будівлею системи опалення [1]. Ці пристрої можуть бути фізично розташовані на значну відстань один від одного, однак завдяки мережі вони можуть взаємодіяти один з одним та обмінюватися даними для кращого моніторингу та управління котельнею. Мережа котельні складається з таких ключових компонентів: сервери, комп’ютери та маршрутизатори, щоб об’єднатися разом та взяти на себе контроль за процесами; сенсори температури, нагрівальні елементи та охолоджувачі підключені до контролерів SBC, які опрацьовують дані і керують пристроями; LCD-дисплеї, які показують стан котельні.

Основними складовими системи керування є: комп’ютери, сервери для моніторингу і периферійні пристрої, тобто сенсори температури, нагрівальні та охолоджувальні елементи, маршрутизатори, комутатори і інші IoT-пристрої. Пристрої за допомогою кабелів і бездротових технологій з’єднані між собою. Ethernet-кабелі з’єднують сенсори й контролери з маршрутизатором. При обміні даними між пристроями використовуються різні мережеві протоколи. Протоколи TCP/IP використовують при передаванні даних з комп’ютерів сенсорів та протоколи MQTT використовуються при обмінам даними між IoT-приставками.

Система складається з підключенних датчиків і контролерів до центрального маршрутизатора, який виконує роль вузла зв'язку. Така структура допомагає ефективно керувати усіма процесами в системі опалення. Основним призначенням є контроль температури, забезпечення оптимальної роботи нагрівальних і охолоджуючих елементів, а також забезпечення безпеки системи. Це дозволяє ефективно контролювати та коригувати параметри котла в режимі

реального часу. Масштаби мережі котелень можуть бути різними: від невеликих систем для індивідуальних будинків до великих складних систем для промислових установок. У кожному разі мережа підлаштовується під потреби конкретної системи опалення. Така мережа забезпечує надійне управління та моніторинг, дозволяючи котельні працювати ефективно та безпечно.

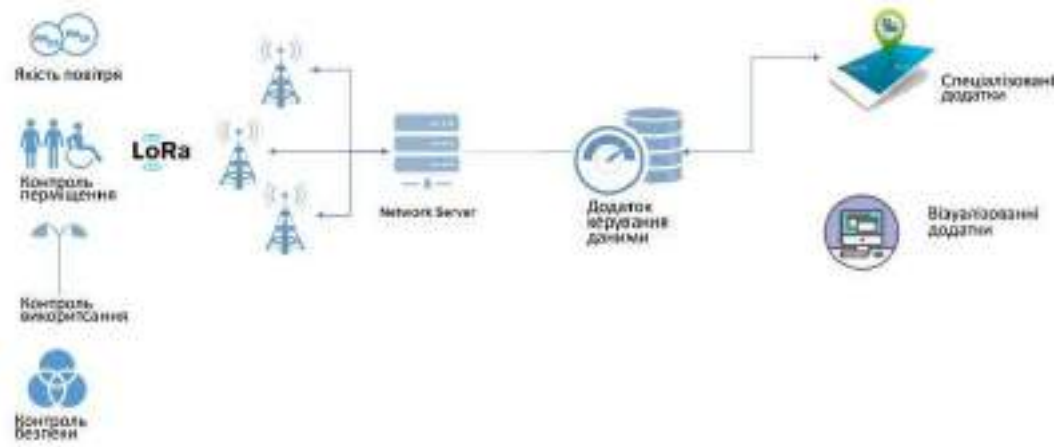

Рисунок. 1.1 – Приклад системи керування на основі інтернет речей

Підключення до комп'ютерної мережі можна здійснити за допомогою кабелів і різних мережевих пристроїв, таких як маршрутизатори та комутатори.

Маршрутизатор забезпечує функцію маршрутизації, тобто він може обробляти потік даних між різними мережами. Вони можуть приймати рішення про те, яким шляхом слідувати дані, щоб досягти певного пункту призначення.[2]. Маршрутизатори виконують різноманітні завдання, такі як функції брандмауера, серверів VPN та DHCP, що дозволяє значно розширити можливості вашої мережі, підвищуючи її безпеку та функціональність. Вони підтримують кілька мережевих технологій, включаючи Ethernet, Wi-Fi, DSL та кабельні з'єднання. Це забезпечує підключення різних пристроїв та використання різних типів з'єднань у вашій мережі, що робить її більш гнучкою, також пропонують розширені можливості конфігурації та управління. Вони дозволяють мережевим адміністраторам налаштовувати різні параметри мережі для забезпечення оптимальної роботи мережевого середовища. Це включає

налаштування пропускної здатності, пріоритету трафіку, безпеки та інших важливих аспектів мережевої інфраструктури, що робить управління мережею більш

Основною функцією маршрутизатора є маршрутизація даних між різними мережами. Він приймає дані з однієї мережі, призначені для іншої мережі, і вирішує, як передавати ці дані. Маршрутизатори використовують таблиці маршрутизації, щоб визначити найкращий маршрут для кожного пакета даних. Маршрутизатори можна використовувати для поділу мережі на підмережі або логічні сегменти. Це дозволяє визначати межі між різними частинами мережі, полегшуючи керування трафіком і забезпечуючи безпеку даних [3];

Маршрутизатори широко використовуються для підключення домашніх і робочих мереж до Інтернету. Вони діють як шлюзи між локальною мережею та постачальником послуг Інтернету, дозволяючи пристроям у мережі отримувати доступ до Інтернету, мають вбудовані засоби захисту, такі як брандмауери, які допомагають захистити мережу від несанкціонованого доступу та кіберзагроз. Що стосується дротового підключення, маршрутизатор можна підключити до мережі за допомогою кабелю Ethernet [4].

Це допомагає розподіляти мережевий трафік через дротове з’єднання, забезпечуючи більш стабільне та швидке з’єднання, особливо порівняно з бездротовим з’єднанням.

Як правило, дротові підключення використовуються, коли потрібна швидка та надійна передача даних, наприклад підключення до серверів або мережевих пристроїв в офісному середовищі.

Комутатори, або свічі, є важливими компонентами мережевої інфраструктури, які забезпечують з'єднання між різними пристроями в локальній мережі (LAN). Вони працюють на канальному рівні моделі OSI і використовують таблиці MAC-адрес для направлення трафіку безпосередньо до потрібного пристрою. Це значно підвищує ефективність мережі, оскільки трафік

передається тільки до необхідного сегмента мережі, зменшуючи колізії та підвищуючи загальну пропускну здатність.

Комутатори також пропонують різні розширені функції для покращення управління та безпеки мережі. Вони можуть підтримувати VLAN (віртуальні локальні мережі) для сегментації мережі на логічні групи, що покращує її структуру та безпеку. Крім того, комутатори можуть включати функції QoS (Quality of Service) для пріоритезації трафіку, що забезпечує стабільну роботу критично важливих додатків. Інтеграція з технологіями управління, такими як SNMP (Simple Network Management Protocol), дозволяє мережевим адміністраторам ефективно моніторити та керувати мережею, забезпечуючи її оптимальну продуктивність та надійність.

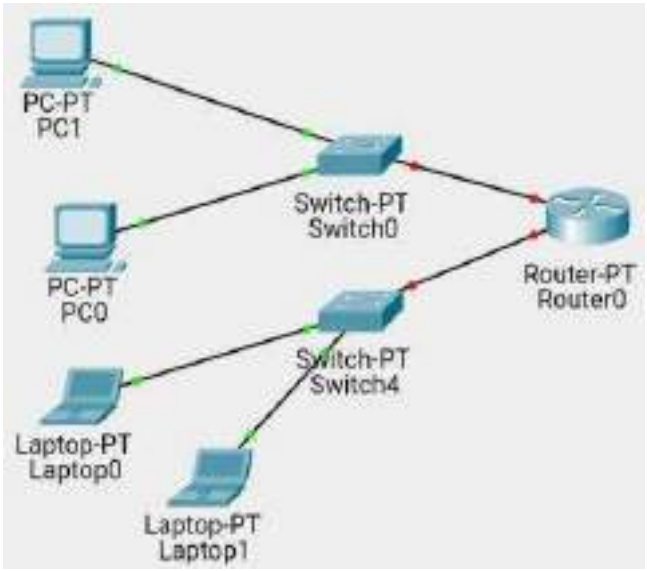

Рисунок 1.2 – Приклад невеликої локальної мережі

Потреба в контролі нагріваючих та охолоджуючих елементів в котельні приводить до використання такого пристрою як Smart Business Communications.

SBC (Smart Business Communications) — Одноплатний програмований компютер для виконання завдань поставлених певною програмою [5]:

– розширені можливості VoIP: SBC має розширені можливості моделювання для мереж голосової та Інтернет-телефонії (VoIP). Він дозволяє

створювати та налаштовувати IP-телефонію, відеоконференції та інші голосові служби;

– моделювання корпоративної мережі: SBC надає можливість моделювати різні аспекти корпоративних мереж, включаючи комутацію даних, бездротові мережі, безпеку мережі, віртуальну приватну мережу (VPN) тощо;

– advanced Devices and Services: SBC містить набір передових мережевих пристроїв і служб, таких як маршрутизатори, комутатори, брандмауери, IP-телефони, сервери тощо;

– підтримка технології IoT: SBC також підтримує Інтернет речей (IoT) і дозволяє моделювати підключення до різноманітних датчиків, пристроїв IoT і пов'язаних служб.

– навчальні матеріали: SBC містить навчальні матеріали та приклади сценаріїв, які допоможуть вам використовувати програмне забезпечення для навчання та дослідження корпоративної мережі.

Таким чином, SBC є потужним інструментом для моделювання та аналізу сучасних корпоративних мереж, який надає розширені функції.

У контексті SBC, керування IoT можна моделювати та аналізувати за допомогою різноманітних пристроїв і служб.

Ось деякі ключові аспекти керування Інтернетом речей у SBC: Моделювання пристроїв Інтернету речей: SBC надає можливість використовувати різні пристрої Інтернету речей, такі як температура, вологість, датчики руху, перемикачі та інші пристрої. Ці пристрої можна підключати до Інтернету та використовувати для збору даних [6].

Модель мережі IoT: SBC дозволяє створювати складні мережі IoT, включаючи з'єднання між пристроями IoT, маршрутизацію даних, безпеку та інші аспекти. Це дозволяє аналізувати різні сценарії використання та вплив мереж IoT на бізнес-процеси.

Моніторинг і керування пристроями: SBC можна використовувати для моделювання моніторингу та керування пристроями IoT. Це включає

зчитування даних віддаленого датчика, виконання різних дій на основі зібраних даних, наприклад активацію пристроїв, надсилання сповіщень тощо.

Аналіз даних: SBC допомагає аналізувати та використовувати дані, зібрані з пристроїв IoT, для прийняття рішень. Це може включати виявлення шаблонів, прогнозування подій, моніторинг продуктивності та інші аспекти.

Таким чином, SBC надає інструменти для моделювання та аналізу управління Інтернетом речей у корпоративних мережах, що дозволяє вивчати різні аспекти цієї технології та її вплив на бізнес-процеси.

SBC можна запрограмувати для керування Інтернетом речей (IoT) за допомогою мови програмування Python. Python є потужним і популярним інструментом для розробки програмного забезпечення, включно з додатками IoT. Використання Python у SBC дозволяє створювати сценарії та програми для керування пристроями IoT, збору й аналізу даних і взаємодії з мережею. Ось деякі ключові аспекти програмування для керування IoT у SBC за допомогою Python: Створення сценаріїв для автоматизації завдань: Python дозволяє створювати сценарії для виконання різноманітних завдань у мережі IoT [7].

Це може включати автоматичний збір даних із датчиків, керування пристроями та виконання різноманітних дій на основі умов і даних.

Взаємодія з API пристрою IoT: Python можна використовувати для взаємодії з API пристрою IoT, дозволяючи читати та надсилати дані на пристрої в мережі. Це відкриває можливості для інтеграції та взаємодії з багатьма різними пристроями та службами [8].

Аналіз даних і прийняття рішень: Python надає широку функціональність для аналізу даних, зібраних з пристроїв IoT, і прийняття рішень на основі цих даних. Це може включати виявлення шаблонів, прогнозування подій і розробку алгоритму керування.

Розробка програм дистанційного керування: Python можна використовувати для створення програм, які віддалено керують пристроями IoT

через мережу [9]. Це дозволяє керувати пристроями з будь-якого місця, де є доступ до Інтернету.

Корпоративна мережа - це мережева інфраструктура, яка з'єднує різні відділи, офіси та інші структурні частини компанії, які можуть бути розташовані на великій відстані один від одного.

Побудова корпоративної мережі відрізняється від локальної мережі (LAN) тим, що вона повинна забезпечувати ефективний обмін даними між різними місцями з урахуванням великих відстаней і різного обсягу даних, що передаються.

Основні аспекти побудови корпоративної мережі включають:

– мінімізація обсягу переданих даних: у корпоративній мережі важливо мінімізувати обсяг даних, що передаються онлайн. Цього можна досягти шляхом оптимізації мережевого трафіку, використання стиснення даних, кешування та інших методів;

– висока доступність і надійність: корпоративні мережі повинні мати високу доступність і надійність, особливо у випадку територіально розподілених вузлів. Це може вимагати використання резервних каналів зв’язку, а також механізмів автоматичного відновлення після відмови;

– безпека даних: забезпечення безпеки даних є одним із найважливіших аспектів бізнес-мережі. Це включає захист даних під час їх переміщення в мережі, захист мережевих ресурсів від несанкціонованого доступу та захист від різних кіберзагроз;

– управління мережею: ефективне керування корпоративною мережею дозволяє вам забезпечити оптимальну продуктивність і безпеку мережі. Це включає моніторинг мережевого трафіку, налагодження та обслуговування мережевих пристроїв, а також впровадження політик керування мережею;

– масштабованість: корпоративні мережі мають бути масштабованими, тобто готовими до збільшення трафіку та майбутнього розширення без втрати продуктивності та ефективності.

Тому при створенні корпоративної мережі важливо враховувати ці аспекти, щоб забезпечити ефективну роботу мережі та високий рівень безпеки та доступності [10]. Наш випадок стосується необхідності моніторингу стану пристроїв Інтернету речей (IoT) за межами локальної мережі, наприклад через мережевий інтерфейс. Це може бути проблемою для багатьох організацій, особливо якщо їхні пристрої IoT розташовані далеко одне від одного або від центральної мережі.

Ось докладніше про цю потребу та про те, як її можна задовольнити:

– Віддалений моніторинг: Оскільки пристрої IoT можуть бути розташовані далеко від центральної мережі, важливо, щоб існував спосіб віддалено контролювати їхній стан і діяльність. Це дозволяє мережевим операторам контролювати продуктивність пристрою, виявляти помилки та діагностувати проблеми.

– Мережевий інтерфейс: Моніторинг мережевого інтерфейсу означає використання мережевого підключення для доступу до пристроїв IoT. Цього можна досягти за допомогою мережевих протоколів, таких як TCP/IP або протоколів IoT, які дозволяють збирати дані та контролювати пристрої в мережі.

– Безпека зв’язку: Оскільки дані стану з пристроїв IoT можуть бути конфіденційними, важливо переконатися, що зв’язок між цими пристроями та базовою мережею є безпечним. Це може включати використання шифрування трафіку, автентифікації пристрою та інших методів безпеки.

– Система моніторингу та контролю: Для дистанційного моніторингу стану пристроїв IoT можна використовувати спеціальну систему моніторингу та контролю, яка дозволяє контролювати активність пристрою, отримувати сповіщення про несправності та перешкоди в роботі пристрою за допомогою пульта дистанційного керування.

– Масштабованість і ефективність: Важливо мати ефективний і масштабований спосіб моніторингу справності пристроїв IoT, особливо у

великих мережах із великою кількістю пристроїв. Це може включати використання розподілених систем моніторингу та керування, які дозволяють у режимі реального часу відстежувати стан пристрою та реагувати на події.

1.2 Одноплатні комп’ютери для контролю IoT

Одноплатні комп’ютери (SBC) — це повні комп’ютери, у яких усі компоненти, необхідні для роботи, інтегровані на одній платі.

Вони включають центральний процесор (CPU), оперативну пам’ять (RAM), інтегровані інтерфейси для підключення до периферійних пристроїв і часто інтегроване сховище даних або підключення до зовнішньої пам’яті.

Одноплатні комп’ютери призначені для різноманітних застосувань, від простих навчальних проектів до складних промислових систем.

Основні компоненти SBC

центральний процесор (CPU): Центральний процесор — це «мозок» комп’ютера, який відповідає за виконання програм і обробку даних. SBC зазвичай використовують енергоефективні процесори ARM, хоча є моделі на основі архітектури x86;

довільна пам’ять (RAM): RAM забезпечує тимчасове зберігання даних, на яких працює процесор. Обсяг пам'яті може варіюватися від кількох сотень мегабайт до кількох гігабайт;

зберігання даних: Флеш-пам'ять, SD-карта або жорсткий диск, підключений через інтерфейс USB або SATA, використовується для постійного зберігання даних і операційної системи;

інтерфейс вводу/виводу (I/O): Одноплатний комп’ютер, оснащений різними портами для підключення периферійних пристроїв:

usb: Для підключення клавіатури, миші, флеш-накопичувача тощо;

hdmi або VGA: Для виведення відео на екран;

ethernet або Wi-Fi: Для підключення до мережі;

GPIO (вхід/вихід загального призначення): призначений для підключення датчиків, двигунів, світлодіодів та інших електронних компонентів.

живлення: SBC зазвичай живиться від зовнішнього адаптера через порт microUSB, USB-C або спеціальний роз’єм живлення.

Типи одноплатних комп'ютерів

raspberry Pi: Один із найпопулярніших одноплатних комп’ютерів, який широко використовується в навчальних, хобі та професійних проектах;

Пропонує різноманітні моделі з різною потужністю та функціями [11].

Рисунок 1.5 – Raspberry Pi

arduino: Хоча технічно Arduino є мікроконтролером, а не комп’ютером загального призначення, його часто використовують у подібних проектах для керування електронними компонентами [12];

Рисунок 1.6 – Arduino

beaglebone: Ще один потужний SBC, який пропонує багато можливостей для розробників, особливо у сфері промислової автоматизації [13];

odroid, Banana Pi та інші: Альтернативи Raspberry Pi пропонують різні конфігурації та функції, часто зосереджені на вищій продуктивності або спеціалізованих програмах [14].

Основні програми SBC

освіта та навчання: Використовується для навчання базовим знанням програмування, електроніки та робототехніки. Зручний для проведення лабораторних робіт та проектних уроків;

хобі та саморобні проекти: Популярно серед ентузіастів, які будують різні проекти, такі як розумні будинки, медіа-центри, ігрові консолі тощо;

промислове та комерційне застосування: Використовується в автоматизації виробничих процесів, вбудованих системах, Інтернеті речей (IoT), де потрібні надійні та економічні рішення для економії енергії;

прототипування: допомагає розробникам швидко створювати та тестувати нові продукти та рішення перед масовим виробництвом.

Переваги одноплатного комп’ютера

компактність: Оскільки він об’єднує всі компоненти на одній платі, SBC має невеликі розміри, тому його можна використовувати у обмеженому просторі;

енергоефективність: Більшість SBC споживають дуже мало електроенергії, що робить їх ідеальними для проектів з автономним живленням;

ціна: Відносно низька вартість робить SBC доступним для широкого кола користувачів, включаючи студентів, ентузіастів і малий бізнес;

гнучкість і масштабованість: Завдяки різноманітним інтерфейсам і підтримці широкого спектру периферійних пристроїв SBC можна налаштувати для виконання різноманітних завдань.

Одноплатні комп'ютери стали невід'ємною частиною сучасної індустрії технологій завдяки своїй універсальності, доступності та потужності. Вони відкривають нові можливості для інновацій як у професійній сфері, так і в особистих проектах.

Переваги та недоліки використання SBC

Переваги:

Доступність для початківців –надає зручний інтерфейс, що дозволяє навіть початківцям швидко освоїти функції та основи роботи з одноплатними комп'ютерами (SBC). Завдяки інтуїтивно зрозумілому дизайну та зрозумілому розташуванню елементів користувачі можуть легко почати створювати та налаштовувати свої проекти;

детальна документація та ресурси - надано детальну документацію та велику кількість навчальних матеріалів, щоб допомогти користувачам швидко зрозуміти, як працювати з одноплатними комп'ютерами в симуляторі. Онлайн-курси, відеоуроки, форуми підтримки та інші освітні ресурси полегшують вивчення матеріалу;

моделювання реального середовища - Packet Tracer може імітувати реальні умови роботи SBC, дозволяючи користувачам тренуватися в умовах, наближених до реалістичних. Це корисно для навчання та тестування без ризику пошкодження обладнання або додаткових витрат на додаткове обладнання;

безпека та прибутковість - Використання емулятора дозволяє уникнути ризиків, пов'язаних із поганим з'єднанням або конфігурацією реального

пристрою. Крім того, це економить багато грошей, оскільки не вимагає придбання фізичних компонентів для навчання та тестування.

Візуалізація активності SBC в мережі

Візуальне представлення мережевих підключень - дозволяє візуально спостерігати за активністю одноплатних комп'ютерів у мережі. Користувачі можуть переглядати всі з’єднання між SBC та іншими мережевими пристроями, що значно полегшує розуміння мережевих топологій і процесів, що в них відбуваються.

інтерактивна візуалізація даних - Інтегровані інструменти моніторингу та аналізу даних у реальному часі дозволяють користувачам бачити, як дані передаються між пристроями, стан порту, як він змінюється, і як мережа реагує на різні команди та події. Це сприяє глибшому розумінню мережевих протоколів і механізмів обміну даними;

можливість створювати складні мережеві сценарії - можна моделювати складні мережеві сценарії, включаючи взаємодію між різними типами пристроїв і протоколів. Це дозволяє користувачам тестувати різні конфігурації та сценарії, щоб зрозуміти, як вони впливають на продуктивність мережі;

анімація процесу - надає можливість анімувати процеси обміну даними, роблячи навчання більш інтуїтивно зрозумілим і цікавим. Користувачі можуть спостерігати за тим, як пакети даних переміщуються по мережі, як передаються сигнали та як різні пристрої реагують на ці сигнали.

Недоліки

обмежена функціональність при емуляції - Незважаючи на багатий набір функцій, Packet Tracer не може повністю замінити справжній SBC. Деякі особливі функції та можливості, доступні на реальних пристроях, можуть бути недоступні в емуляторі. Це може обмежити можливість тестування деталей і коригування складних проектів;

відсутність підтримки деяких конкретних пристроїв і протоколів - не підтримує мережеві протоколи. Це може ускладнити роботу користувачів, які

хочуть імітувати поведінку певного обладнання або використовувати певні мережеві протоколи;

потрібні оновлення та підтримка - Для підтримки в актуальному стані потрібні регулярні оновлення. Відсутність своєчасних оновлень може призвести до відставання від реальних технологій і невідповідності між моделюванням і реальністю.

Таким чином, використання одноплатного комп'ютера має такі переваги, як легкість навчання, візуалізація операцій і можливість імітувати складні мережеві сценарії.

У той же час існують певні обмеження, пов'язані з функціональністю і його сумісністю з пристроями.

1.4 Використання IoT

Інтернет речей (IoT) — це концепція мережі фізичних об'єктів, обладнаних датчиками, програмним забезпеченням та іншими технологіями для підключення та обміну даними з іншими пристроями та системами через Інтернет. IoT дозволяє фізичним об'єктам взаємодіяти з цифровим світом, надаючи можливість автоматизувати, контролювати та аналізувати різні процеси та системи [18].

Основні можливості IoT включають підключення, збір даних, аналіз даних і автоматизацію. Фізичні пристрої підключено до Інтернету, що дозволяє їм обмінюватися даними. Вони оснащені датчиками, які збирають дані про навколишнє середовище або об'єкт, до якого вони прикріплені. Зібрані дані обробляються для отримання корисної інформації та прийняття рішень.

Пристрій може автоматично виконувати дії на основі отриманих даних і встановлених алгоритмів. Основні принципи IoT включають інтеграцію фізичних пристроїв із цифровими системами, використання різноманітних

датчиків і приводів для взаємодії з фізичним світом, забезпечення зв’язку між пристроями для обміну даними через стандартні протоколи зв’язку (Wi-Fi, Bluetooth, Zigbee, LTE тощо), збирати великі обсяги даних для аналізу та прийняття рішень, а також виконувати дії та керувати процесами без участі людини.

Основні компоненти систем Інтернету речей включають датчики та датчики, які збирають дані з навколишнього середовища, агенти (приводи), які виконують дії у фізичному світі на основі отриманих команд, контролери та шлюзи, які обробляють дані від датчиків і надсилають команди до приводів, надаючи зв'язок, з'єднання між локальною мережею пристроїв IoT та Інтернетом, протоколи зв'язку, що забезпечують зв'язок між пристроями IoT і хмарними службами, а також сервісні та аналітичні хмари, які зберігають, обробляють і аналізують дані з пристроїв IoT, надаючи можливість доступу та контролю дані пристроїв через Інтернет [19].

Додатки IoT включають розумні будинки, де освітлення, опалення та безпека автоматизовані, промисловість, де відстежується стан обладнання та контролюються виробничі процеси, охорона здоров’я, де відстежується стан пацієнтів і медичне обладнання, сільське господарство, де відстежується стан ґрунту, іригаційні системи автоматизовані, а транспорт, де контролюються транспортні засоби, контролюється та контролюється рух. Таким чином, Інтернет речей (IoT) є потужним інструментом для створення інтелектуальних, автономних і взаємопов’язаних систем, здатних покращувати різні аспекти нашого життя.

Архітектура системи IoT базується на кількох ключових компонентах, включаючи датчики та датчики. Вони є ключовими елементами будь-якої системи IoT, оскільки дозволяють збирати дані про середовище або об’єкти, з якими вони взаємодіють.

Датчики та перетворювачі — це електронні пристрої, здатні вимірювати фізичні параметри, такі як температура, вологість, світло, тиск, рух, склад газу,

рівень звуку тощо. Вони перетворюють ці фізичні параметри в електричні сигнали, які можуть бути оброблені іншими компонентами системи IoT для подальшого аналізу та прийняття рішень.

Датчики температури є одними з найпоширеніших типів датчиків. Вони використовуються для вимірювання температури навколишнього середовища або певних об'єктів. Наприклад, термістори та термопари – це типи датчиків температури, які широко використовуються в промислових, сільськогосподарських системах, системах опалення, вентиляції та кондиціонування повітря (HVAC), а також у багатьох інших галузях. Дані, отримані від датчиків температури, можна використовувати для контролю мікроклімату в приміщенні, моніторингу стану обладнання або попередження про можливі несправності [20].

Іншим важливим типом датчика є датчик вологості. Вони вимірюють вологість повітря або ґрунту, необхідну для сільського господарства, де важливо підтримувати оптимальні умови для росту рослин [21]. Ці датчики також використовуються в системах кондиціонування повітря, де вони допомагають підтримувати комфортний рівень вологості в будинку.

Світловий датчик використовується для вимірювання інтенсивності світла. Вони можуть автоматично регулювати внутрішнє освітлення, забезпечуючи енергоефективність і комфорт користувача. Такі датчики використовуються в системах розумного будинку, де вони можуть автоматично вмикати або вимикати світло в залежності від рівня природного освітлення.

Датчики руху є ще одним важливим компонентом систем IoT. Вони використовуються для виявлення руху в певній зоні, що важливо для систем безпеки. Наприклад, датчики руху можна використовувати для автоматичного ввімкнення світла при виявленні людини або спрацьовування тривоги в разі несанкціонованого доступу.

Датчики газу вимірюють концентрацію різних газів у повітрі, що важливо для моніторингу якості повітря в приміщеннях або виявлення небезпечних

витоків газу в промислових середовищах. Ці датчики можуть виявляти такі гази, як вуглекислий газ (CO2), метан (CH4), чадний газ (CO) та інші.

Сучасні системи IoT часто використовують складні датчики, які поєднують кілька функцій в одному пристрої. Наприклад, датчики навколишнього середовища можуть одночасно вимірювати температуру, вологість, тиск і якість повітря. Це забезпечує більш повний і точний моніторинг умов, дозволяючи системам IoT ефективніше реагувати на зміни навколишнього середовища.

Дані, зібрані датчиками, передаються на контролер і шлюз, які обробляють інформацію та передають її в хмарні служби для подальшого аналізу. На основі аналізу даних можна приймати автоматизовані рішення або сповіщати користувачів про необхідність втручання. Тому сенсори та датчики є важливими компонентами систем IoT, забезпечуючи збір і передачу даних, необхідних для прийняття рішень і автоматизації процесів.

Основні функції та можливості для моделювання Інтернету речей (IoT) дозволяють користувачам створювати та тестувати складні мережі, які включають пристрої IoT, датчики, контролери та інші компоненти.

Користувачі можуть вибирати з бібліотеки пристроїв Cisco та сторонніх виробників, таких як датчики, контролери, шлюзи та агенти, і розміщувати їх на віртуальній мережевій картці. Це дозволяє створювати різні сценарії та топології для моделювання реальних систем IoT.

Використання нагрівальних елементів у системах IoT:

Моніторинг і контроль: Використовуючи технологію IoT, нагрівальні елементи можна підключати до Інтернету, що дозволяє відстежувати стан і контролювати можливості дистанційного керування ними. Наприклад, користувачі можуть увімкнути або вимкнути електрообігрівач у котельні за допомогою мобільного додатку.

Оптимізація споживання енергії: Збір даних від нагрівальних елементів за допомогою датчиків та їх аналіз дозволяє оптимізувати споживання енергії.

Наприклад, система IoT може автоматично регулювати температуру в кімнаті залежно від зовнішніх умов і режиму роботи.

Попередження про несправності: За допомогою технології IoT нагрівальний елемент можна підключити до системи моніторингу, яка автоматично виявляє несправності та надсилає повідомлення про них власнику компонента або в службу сервісної служби. Управління ресурсами: системи Інтернету речей дозволяють ефективніше використовувати такі ресурси, як газ або електроенергія для опалення приміщень.

Завдяки збору та аналізу даних системи IoT можуть автоматично регулювати роботу нагрівальних елементів, враховуючи зовнішні умови, графіки використання кімнат та інші фактори. Наприклад, система може вимикати обігрівач, коли в кімнаті нікого немає, або знижувати температуру вночі для економії енергії.

Розумне дистанційне керування: Використовуючи IoT, власники будинків можуть отримати доступ до своєї системи опалення з будь-якого місця за допомогою смартфона або комп'ютера. Це дозволяє їм вимикати або регулювати кімнатну температуру, навіть коли їх немає вдома, забезпечуючи більшу зручність і контроль.

Інтеграція з іншими системами: системи IoT можна інтегрувати з іншими системами управління будинком, такими як системи безпеки, автоматизація освітлення та погодні служби. Це допомагає створити інтелектуальне й автоматизоване середовище, яке оптимізує комфорт і економить енергію.

У контексті Інтернету речей (IoT) компоненти охолодження відіграють важливу роль у забезпеченні оптимальних температурних умов для приміщень, обладнання та електроніки. Вони забезпечують ефективне охолодження, запобігають перегріву та забезпечують правильну роботу пристрою [24].

Типи охолоджуючих елементів:

вентилятори: Вентилятори є одними з найпоширеніших охолоджуючих елементів. Вони створюють потік повітря, який допомагає відводити тепло від

приладів або приладів. Вентилятори можна використовувати всередині приміщень або в системах вентиляції;

кондиціонер: Кондиціонер використовується для активного охолодження повітря в приміщенні. Вони інтегровані в системи центрального опалення та охолодження або можуть бути переносними для використання в певних приміщеннях;

термоелектричний модуль: Ці елементи використовують принцип Пельтьє для охолодження або нагрівання. Вони забезпечують ефективне охолодження електронних пристроїв або модулів.

Використання охолоджуючих елементів у системах IoT: Моніторинг і контроль: Використовуючи технологію IoT, охолоджуючі елементи можна підключати до Інтернету, що дозволяє відстежувати стан і дистанційно керувати ними.

Оптимізація енергоспоживання: Збір даних із компонентів охолодження та їх аналіз дозволяє оптимізувати енергоспоживання. Система IoT може автоматично регулювати роботу кондиціонера або вентилятора, враховуючи зовнішні умови та графік використання приміщення.

Датчики температури відіграють важливу роль у забезпеченні оптимальних умов і контролю температури в системах опалення та гарячого водопостачання. Вони допомагають виявляти зміни температури, дозволяючи системі автоматично регулювати роботу котла та іншого обладнання для забезпечення комфортної та ефективної роботи всієї системи.

Типи датчиків температури:

термістор: Це один із найпоширеніших типів датчиків температури. Термистор - резистор, опір якого змінюється залежно від температури. За допомогою термістора можна точно виміряти температуру і використовувати ці дані для управління системою опалення;

термопара: Це тип датчика температури, який використовує принцип термоелектричного ефекту для вимірювання температури. Термопари здатні

працювати в широкому діапазоні температур і забезпечують високу точність вимірювань.

Використання датчиків температури в системах IoT: Моніторинг і контроль температури: Використовуючи датчики температури, системи IoT можуть безперервно вимірювати температуру в котельнях та інших зонах системи опалення. Це дає можливість реагувати на зміни температури та автоматично регулювати роботу котла чи насоса для забезпечення оптимальних температур. Оптимізація енергоспоживання: Вимірювання температури допомагає оптимізувати роботу котлів та іншого обладнання для економії енергії. Наприклад, якщо температура в котельні перевищує встановлений поріг, система може автоматично знизити потужність котла, щоб запобігти перегріву.

Приклад застосування:

Моніторинг температури в режимі реального часу: система IoT може надсилати дані від датчика температури до центральної системи моніторингу для аналізу даних у режимі реального часу. Це дозволяє оператору системи швидко реагувати на будь-які зміни температури та вживати необхідних заходів [25].

Оптимізація роботи котельні: Аналіз даних датчика температури може допомогти оптимізувати роботу котельні, щоб забезпечити ефективну та економічну роботу. Наприклад, система може автоматично регулювати температуру води в системі опалення в залежності від зовнішніх умов і навантаження системи.

2 ПРОЕКТУВАННЯ МЕРЕЖІ

2.1 Підбір пристроїв

Комп'ютерна мережа котельні включає в себе набір пристроїв і компонентів, які допомагають ефективно управляти системою опалення і контролювати температурний режим. Основою мережі є центральний маршрутизатор, який забезпечує підключення всіх пристроїв. До цього маршрутизатора підключаються різні компоненти системи, включаючи сервери, комп’ютери та деякі пристрої IoT.

Сервер служить основним центром обробки даних. Він зберігає інформацію про робочі параметри системи опалення, забезпечує інтерфейс моніторингу та керування, а також забезпечує можливість обробки та зберігання даних від датчиків температури та інших пристроїв IoT. Сервер підключається до маршрутизатора через кабель Ethernet, що забезпечує стабільне та швидке з'єднання.

Комп'ютер, підключений до тієї ж мережі, дозволяє користувачеві отримати доступ до інтерфейсу керування системою опалення. Це може бути як стаціонарний комп’ютер, так і ноутбук, що дозволяє професіоналам керувати налаштуваннями системи, переглядати дані моніторингу та вносити необхідні налаштування в режимі реального часу.

Одноплатний комп’ютер (SBC) відповідає за обробку даних датчиків і керування елементами нагрівання й охолодження. Він отримує дані від датчика температури, аналізує дані та, залежно від отриманого значення, активує нагрівальний або охолоджуючий елемент. Це забезпечує підтримку оптимальної температури в котельні.

Датчик температури вимірює температуру в реальному часі та передає ці дані на SBC. Нагрівальний елемент відповідає за підвищення температури в системі, а охолоджуючий призначений для зниження температури в разі її

перевищення. Ці елементи працюють разом з SBC, дозволяючи точно контролювати температурний режим.

Дисплей і монітор температури також підключені до мережі. На дисплеї відображається поточний стан системи, включаючи інформацію про те, чи триває нагрівання, тоді як монітор температури показує поточну температуру. Це дозволяє користувачам отримувати важливу інформацію в зручній формі та швидко реагувати на будь-які зміни.

Усі пристрої підключено до маршрутизатора для забезпечення їх сумісності. Пристрої використовують дротові та бездротові з'єднання залежно від потреб системи та розташування пристрою. Налаштування мережевих протоколів, таких як TCP/IP, забезпечує ефективний обмін даними між пристроями, дозволяючи їм працювати як єдина система.

Безпека мережі забезпечується фільтрацією трафіку, яка контролює доступ до мережі та захищає її від небажаних вторгнень. Шифрування даних забезпечує захист переданої інформації, а процеси автентифікації гарантують, що лише авторизовані користувачі мають доступ до критичних елементів системи.

Моніторинг та управління мережею здійснюється за допомогою спеціалізованого програмного забезпечення, що дозволяє відстежувати стан усіх підключених пристроїв у режимі реального часу, аналізувати їх активність та оперативно реагувати на кожен інцидент. Автоматизація процедур контролю температури та інших параметрів системи опалення здійснюється за допомогою скриптів або програм, що значно підвищує ефективність і надійність мережі.

Створена таким чином мережа для котельні включає всі необхідні компоненти для забезпечення стабільної та ефективної роботи системи опалення, з можливістю гнучкого управління та моніторингу температурного режиму в режимі реального часу. Щоб котельня працювала належним чином і забезпечувала оптимальний температурний режим, необхідно ретельно підбирати мережеве обладнання, яке буде використовуватися для моніторингу та управління системою. У сучасних умовах, коли важливо не тільки підтримувати

стабільну роботу системи, але і забезпечити безпеку і надійність, важливим фактором стає вибір технічних характеристик пристрою.

Основою мережі є центральний маршрутизатор, який повинен мати високу пропускну здатність і підтримувати сучасні мережеві протоколи, щоб забезпечити швидке і надійне з'єднання всіх компонентів системи. До маршрутизатора підключаються різні пристрої, включаючи сервер, який служить основним центром обробки даних, і функціональний комп'ютер для управління та моніторингу.

Сервер повинен мати достатню обчислювальну потужність для обробки великих обсягів даних від датчиків температури та інших пристроїв IoT. Він також повинен забезпечувати можливість зберігати дані та підтримувати безперервну роботу в умовах високого навантаження.

Одноплатний комп’ютер (SBC), відповідальний за аналіз даних датчиків і керування елементами нагрівання й охолодження, повинен мати необхідну продуктивність і енергоефективність. Важливим аспектом є можливість швидкої обробки даних і передачі команд іншим компонентам системи.

Датчики температури, життєво важливі для моніторингу умов у котельні, мають бути дуже точними та надійними, щоб забезпечувати точні вимірювання температури в реальному часі. Нагрівальний елемент і охолоджуючий елемент повинні мати достатню потужність для підтримки необхідних температурних параметрів в різних умовах експлуатації.

Дисплей і монітор температури відіграють важливу роль у відображенні поточного стану системи та наданні користувачеві інформації про температуру та стан нагрівальних елементів. Вони мають бути простими у використанні та надавати точну та актуальну інформацію.

Крім того, усі ці компоненти мають бути інтегровані в єдину мережу за допомогою відповідних мережевих протоколів і технологій, таких як TCP/IP, що забезпечує ефективний обмін даними між пристроями. Кібербезпека також є

важливим аспектом, включаючи фільтрацію трафіку, шифрування даних і автентифікацію користувачів.

Тому технічні характеристики мережевого обладнання, яке буде використовуватися в системі котельні, мають особливе значення для забезпечення стабільної, ефективної та безпечної роботи. Нижче наведено детальний опис технічних характеристик кожного елемента мережі, що дозволить більш детально ознайомитися з їх параметрами та можливостями.

Сервер: HP Proliant DL 380 Gen9 (8x2.5)

SFF SFF 16GB (4x4GB) DDR4 ECC Registered 2400 Mhz

2 x Intel XEON 8 Core E5-2630 V3 [2.40GHz - 3.20GHz]

RAID-контролер HP P440AR + Cache FBWC 2GB (NO BATT) [26]

Рисунок 2.1 – Сервер

Моноблок Asus Vivo AiO V241EAK-BA180M (90PT02T2-M01BW0) Gold

Миша Logitech G102 Клавіатура дротова Genius SlimStar 126 USB Black UKR

Процесор Двоядерний Intel Core i3-1115G4 (3.0 — 4.1 ГГц)

Відеокарта Intel UHD Graphics Обсяг оперативної пам'яті 16 ГБ

Рисунок 2.2 – Комп’ютер

Датчик температури CISCO MT10-HW

Об'єкт детекції температура та вологість

Бездротовий протокол Wi-Fi

Тип елемента живлення 2 x AA

Частота передачі 2.4 ГГц

Інтерфейси: Ethernet, Wi-Fi

Рисунок 2.3 – Датчик температури

Система охолодження Thermocold MEX SEA 050 Z C

Тип куллеру Повітря – Вода Холодопродуктивність, кВт 49,6

Температура рідини на виході, °С -7...+18

Коефіцієнт ефективності EER 2,72

Споживана потужність, кВт 18,2

Витрата рідини, м³/год 8,5

Рисунок 2.4 – Система охолодження

Маршрутизатор LINKSYS MR6350

Характеристики бездротової мережі

Робоча частота 2.4 / 5 ГГц

Захист інформації WPA2-PSK

Рисунок 2.5 – Маршрутизатор

Модуль Нагріву МН-80

Максимальний робочий тиск теплоносія МПа (бар) 0,6 (6)

Максимальна температура теплоносія °С 95

Температура продуктів згорання на виході з модуля, не меньше> °С110

Рисунок 2.6 – Модуль Нагріву

SBC RASPBERRY PI 5 8GB (RPI5-8GB)

Процесор Arm Cortex-A76

Кількість ядер процесора 4

Оперативна пам'ять 8 ГБ [27]

Рисунок 2.7 – SBC

Система моніторингу температури testo Saveris 2 T2

Діапазон вимірювань -50…+150 °C (залежить від зонда, що підключається)

Похибка від ±0,3 °C (залежить від зонда, що підключається)

Рисунок 2.8 – Система моніторингу температури

Щоб забезпечити ефективну роботу мережевої інфраструктури котельні, необхідно ретельно підійти до вибору обладнання, яке ляже в основу цієї системи. Відповідність технічних характеристик вибраного обладнання потребам і вимогам котельні є особливо важливим аспектом для досягнення стабільної та надійної роботи всієї мережі. Нижче ви знайдете детальний опис основних компонентів, що складають мережеву інфраструктуру, та їх основні технічні характеристики.

Основним компонентом мережевої інфраструктури є сервер, представлений моделлю SFF HP Proliant DL 380 Gen9 (8x2.5). Цей сервер оснащений оперативною пам'яттю DDR4 ECC 16 ГБ, зареєстрованою з частотою 2400 МГц, що забезпечує високу швидкість обробки даних і стабільну роботу. Два процесори Intel XEON 8 Core E5-2630 V3 з тактовою частотою від 2,40 ГГц до 3,20 ГГц і енергоспоживанням 85 Вт, кожен з яких забезпечує достатню обчислювальну потужність для виконання складних обчислювальних завдань.

RAID-контролер HP P440AR з кеш-пам'яттю 2 ГБ (акумулятор не входить) забезпечує високу надійність і швидкість доступу до даних.

Моноблок Asus Vivo AiO V241EAK-BA180M з комплектом периферії, таким як миша Logitech G102 та дротова клавіатура Genius SlimStar 126 USB Black UKR використовується для зручного доступу до системи та системи управління системою. Моноблок оснащений двоядерним процесором Intel Core i3-1115G4 з тактовою частотою 3,0-4,1 ГГц і вбудованою відеокартою Intel UHD Graphics, що дозволяє ефективно здійснювати управління і моніторинг ігрової системи. Обсяг оперативної пам'яті 16 ГБ забезпечує швидке виконання багатозадачних процесів.

Датчик CISCO MT10-HW використовується для моніторингу температури, здатний визначати температуру і вологість. Він працює через бездротовий протокол Bluetooth і живиться від двох батарейок типу AA. Датчик підтримує передачу даних на частоті 2,4 ГГц і може підключатися через інтерфейси Ethernet і Wi-Fi, забезпечуючи гнучкість встановлення та використання.

Система охолодження представлена кулером Thermocold MEX SEA 050 Z C, який працює за принципом повітря-вода. Потужність охолодження системи становить 49,6 кВт, температура рідини на виході від -7 до +18°C. EER становить 2,72, а споживана потужність 18,2 кВт. Кулер оснащений двома компресорами і одним контуром, що забезпечує ефективне охолодження в різних умовах експлуатації.

Маршрутизатор LINKSYS MR6350 забезпечує надійне з'єднання і високу швидкість передачі даних завдяки підтримці дротового і бездротового режимів роботи. Він працює на частотах 2,4 і 5 ГГц і має дві антени для кращого покриття. Система захисту WPA2-PSK забезпечує безпеку передачі даних. Маршрутизатор має чотири порти LAN (RJ-45), які підтримують стандарти Ethernet до 1000BASE-T.

Нагрівальний модуль MH-80, 100, 120 ECO M може працювати при максимальному тиску теплоносія до 0,6 МПа (6 бар) і досягати температури

теплоносія до 95 °С. Температура продуктів згоряння на виході з модуля становить не менше 110°С, що дозволяє ефективно підтримувати необхідний температурний режим.

RASPBERRY PI 5 8 ГБ одноплатний комп'ютер (RPI5-8 ГБ) оснащений чотирьохядерним процесором Arm Cortex-A76 і 8 ГБ оперативної пам'яті. Це забезпечує високу продуктивність обробки даних і низьке енергоспоживання, що робить його ідеальним для використання в системах IoT.

Система моніторингу температури представлена реєстратором температури testo Saveris 2 T2 WiFi, який має два зовнішні канали вимірювання температури. Діапазон вимірювань становить від -50 до +150°С з похибкою ±0,3°С залежно від підключеного зонда, що забезпечує дуже точний моніторинг температурних умов.

РК-дисплей Adafruit I2C 16x2 з розмірами 36 мм у висоту, 8 мм в ширину і 80 мм в довжину дозволяє переглядати інформацію про стан системи і поточні показники температури, забезпечуючи зручність і точність при використанні.

Після об'єднання всіх пристроїв в одну мережу отримаємо результат, який зображений на рисунку 2.1

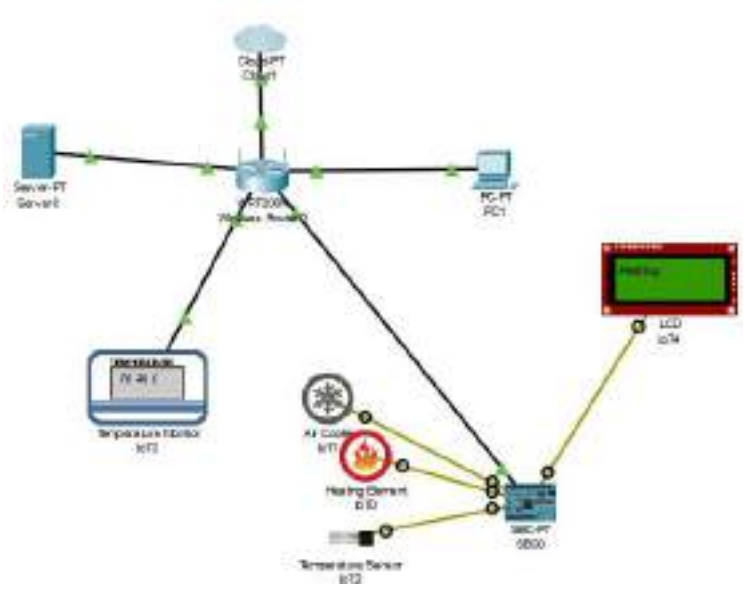

Рисунок 2.9 – Схема мережі

2.2 Вибір інструментарію

Мови та інструменти програмування

Python — це інтерпретована мова програмування високого рівня, відома своїм простим і зрозумілим синтаксисом. Його часто використовують для навчання програмуванню початківців, але це також потужний інструмент для розробки складних програмних систем.

Переваги Python:

– простота використання: Зрозумілий синтаксис Python полегшує вивчення та використання;

– багатий набір стандартних бібліотек: Python має великий набір стандартних бібліотек, що спрощує розробку різноманітних програмних рішень;

– велика спільнота: Численні ресурси, документація та форуми підтримки роблять Python доступним як для початківців, так і для експертів;

– підтримує кілька доменів програм: Від веб-розробки до аналізу даних, машинного навчання та вбудованих систем;

– інтеграція з іншими мовами: Python легко інтегрується з C, C++, Java, що розширює його можливості.

Приклад використання Python на SBC:

– Робота з GPIO: Контроль різних датчиків та пристроїв.

– Автоматизація завдань: Написання сценаріїв для автоматизації повторюваних завдань.

– Мережеве програмування: Розробка програм керування мережею та обміну даними [15].

C++ — це потужна мова програмування, яка забезпечує високий рівень контролю над ресурсами комп’ютера. Він часто використовується для розробки операційних систем, драйверів і складних програм, які потребують високої продуктивності [16].

Переваги C++:

– висока продуктивність: Програми на C++ працюють дуже швидко завдяки компіляції в машинний код;

– контроль ресурсів: Можливість керувати пам'яттю та апаратними ресурсами;

– об'єктно-орієнтоване програмування: підтримує парадигму ООП, дозволяючи модульний і багаторазовий код.

Приклад використання C++ на SBC:

– розробка драйверів: Створення драйверів для нових апаратних пристроїв;

– виконання високопродуктивних обчислень: Створення програм, які потребують високої продуктивності;

– ігри та графічні програми: Створення складних ігор і графічних програм.

JavaScript — є основною мовою веб-програмування, яка використовується для створення інтерактивних веб-сторінок. Його також можна використовувати поза веб-браузером, наприклад на сервері чи вбудованій системі.

Переваги JavaScript:

– гнучкість: Можна використовувати як на стороні клієнта, так і на стороні сервера;

– широко використовується: Основна мова веб-розробки, яка дуже популярна;

– асинхронний: Вбудована підтримка асинхронних операцій, що дозволяє ефективно обробляти мережеві запити та інші завдання введення-виведення [17].

Приклад використання SBC JavaScript:

– веб-інтерфейс керування пристроєм: Створення веб-інтерфейсу для керування та моніторингу SBC;

– інтернет речей (IoT): Розробка програм для керування пристроями IoT через веб-інтерфейс;

– завдання автоматизації: Створення сценаріїв для автоматизації через Node.js.

Вибір Python через його переваги Ми обираємо Python як основну мову програмування для роботи з SBC, оскільки він має багато переваг:

– простий і легкий для вивчення: Python має синтаксис, який робить його ідеальним для новачків і прискорює розвиток;

– велика кількість бібліотек: Багатий набір бібліотек для різних завдань (наприклад, RPi.GPIO для роботи з GPIO на Raspberry Pi), що дозволяє швидко та ефективно розробляти програми;

– розширена підтримка спільноти: Велика спільнота розробників, багаті ресурси та документація допоможуть вам швидко знайти рішення будь-якої проблеми;

– кросплатформенність: Можливість запуску коду на різних платформах без істотних змін;

– інтерактивне середовище: Інтерактивні середовища розробки, такі як Jupyter Notebook, дозволяють швидко тестувати та налагоджувати код.

3 МОДЕЛЮВАННЯ МЕРЕЖІ КОТЕЛЬНІ

3.1 Налаштування пристроїв

Виконано налаштування роутера через веб-інтерфейс. Ось детальний опис кроків та налаштувань, які застосовано [28].

Налаштування Інтернет-з'єднання

Тип з'єднання: Вибрав "Static IP" (Статична IP).

IP-адреса Інтернету:

Ввів IP-адресу: 209.104.10.13

Встановлено маску підмережі: 255.255.255.0

Ввів шлюз за замовчуванням: 192.168.1.1

Вказано основний DNS-сервер: 192.168.1.10

Додаткові налаштування:

Параметри DNS 2 і DNS 3 залишено порожніми, оскільки вони не обов'язкові.

Поле "Host Name" залишено порожнім.

Поле "Domain Name" залишено порожнім.

MTU (Maximum Transmission Unit) залишено за замовчуванням - 1500.

Налаштування мережі

IP-адреса роутера:

Встановоено IP-адресу: 192.168.0.1

Встановоено маску підмережі: 255.255.255.0

Налаштування DHCP-сервера:

Увімкнув DHCP-сервер, вибравши опцію "Enabled".

Вказано стартову IP-адресу для видачі DHCP: 192.168.0.1

Встановоено максимальну кількість користувачів: 50

Визначено діапазон IP-адрес, які видає DHCP-сервер: від 192.168.0.1 до 192.168.0.50

Залишено час оренди клієнта (Client Lease Time) на 0 хвилин, що означає один день за замовчуванням.

Статичні DNS-сервери для DHCP:

Поля для статичних DNS 1, DNS 2, DNS 3 залишено порожніми.

WINS:

Поле для WINS-сервера залишено порожнім.

Таким чином, виконано налаштування роутера, Встановоеноши статичну IP-адресу для підключення до інтернету, налаштувавши DHCP-сервер для внутрішньої мережі, та забезпечивши коректну конфігурацію мережевих параметрів. Тепер моя мережа має стабільне підключення до інтернету та коректну роботу DHCP-сервера для підключених пристроїв.

Щоб налаштувати мережеве підключення PC1, вибрано статичну IP-адресу. Це означає, що комп'ютер використовуватиме певну IP-адресу, яку призначено вручну, замість автоматичного отримання IP-адреси через DHCP [35] (протокол динамічної конфігурації хоста). Нижче наведено детальний опис встановлених налаштувань:

Конфігурація IP: вибрано опцію «Статичний», що означає статичну конфігурацію IP. Цей вибір робиться у випадках, коли комп'ютер завжди має однакову IP-адресу. Статичні IP-адреси часто використовуються для серверів або інших пристроїв, які повинні мати фіксоване розташування в мережі.

Адреса IPv4:

Встановоено IP-адресу 192.168.1.14. Ця адреса належить до діапазону приватних IP-адрес, які зазвичай використовуються в локальних мережах. Вибрана IP-адреса забезпечує унікальну ідентифікацію комп'ютера в локальній мережі. Діапазон 192.168.1.x зазвичай використовується в багатьох домашніх і бізнес-мережах. У цьому випадку адреса 192.168.1.14 вказує на те, що комп'ютер є чотирнадцятим пристроєм у цій підмережі, хоча це не обов'язково означає фактичне місце серед інших пристроїв.

Маска підмережі:

Встановлено маску підмережі на 255.255.255.0. Ця маска підмережі визначає діапазон IP-адрес, які належать одній мережі. З цією маскою підмережі всі пристрої з IP-адресами, що починаються з 192.168.1, вважатимуться частиною однієї локальної мережі. Маска 255.255.255.0 означає, що перші три байти (192.168.1) є ідентифікатором мережі, а останній байт (у цьому випадку 0,14) є ідентифікатором певного пристрою в цій мережі. Це дозволяє мати до 254 пристроїв в одній мережі (192.168.1.1 - 192.168.1.254) [36].

Використання статичної IP-адреси дозволяє мені завжди знати точне розташування ПК1 в мережі, що полегшує керування мережею, особливо коли мені потрібно отримати доступ до певного комп'ютера для обслуговування або встановлення. Цей підхід також дозволяє уникнути потенційних конфліктів IP-адрес, які можуть виникнути під час використання DHCP у великих мережах із багатьма пристроями.Щоб налаштувати мережеве підключення Server0, вибрано статичну IP-адресу та налаштував параметри шлюзу за замовчуванням і DNS-сервера. Це забезпечує стабільне і постійне мережеве розташування сервера, що важливо для його роботи в мережі. Нижче наведено детальний опис параметрів:

Конфігурація IP: вибрано опцію «Статичний», що означає статичну конфігурацію IP. Це дає серверу фіксовану IP-адресу, яка не змінюється після перезапуску або відключення від мережі.

Адреса IPv4: Встановоено IP-адресу 192.168.1.10. Це приватна IP-адреса, яка використовується для ідентифікації сервера в локальній мережі. Використання адреси 192.168.1.10 гарантує, що сервер буде доступним з цієї адреси з будь-якої точки локальної мережі.

Маска підмережі: Встановоено маску підмережі на 255.255.255.0. Це стандартна маска підмережі класу C, яка дозволяє використовувати до 254 пристроїв у цій мережі (адреси від 192.168.1.1 до 192.168.1.254). Маска 255.255.255.0 вказує, що перші три байти (192.168.1) ідентифікують мережу, а останній байт (.10 у цьому випадку) ідентифікує певний пристрій (хост) у цій мережі.

Параметри безпеки: Бездротова безпека: налаштовано WPA2-PSK для бездротового шифрування на своєму бездротовому маршрутизаторі VRT300N [37]. Встановоено надійний пароль для доступу до вашої мережі Wi-Fi.

Контроль доступу: Налаштовано фільтрування MAC-адрес, щоб дозволити доступ лише авторизованим пристроям.

Безпека мережевого пристрою: Логін і пароль маршрутизатора за замовчуванням змінено. ACL (списки контролю доступу) встановлені на маршрутизаторі для обмеження доступу до певних частин мережі [29].

Виконано тести безпеки: Тестування доступу за допомогою ACL: Тестування операцій ACL, щоб переконатися, що доступ до певних IP-адрес або портів обмежено відповідно до конфігурації [38].

Конфігурація шлюзу IPv4/DNS: також вибрано параметр «Статичний», щоб установити шлюз за замовчуванням і сервер DNS, забезпечуючи стабільне та передбачуване мережеве середовище для сервера.

Шлюз за замовчуванням: Встановоено шлюз за замовчуванням 192.168.0.1. Це IP-адреса маршрутизатора або іншого пристрою, який забезпечує доступ з локальної мережі до інших мереж, включаючи Інтернет. Правильне налаштування шлюзу за замовчуванням має важливе значення для того, щоб сервер міг спілкуватися з пристроями за межами локальної мережі.

DNS-сервер: Поле конфігурації DNS-сервера залишається порожнім. Це означає, що сервер не має DNS-сервера, призначеного для визначення доменних імен в IP-адреси. У більшості випадків заповнення цього поля є необхідним, щоб забезпечити здатність сервера виконувати запити доменного імені. Зазвичай це поле містить адресу DNS-сервера, надану вашим Інтернет-провайдером, або загальнодоступний DNS-сервер, який використовується (наприклад, Google DNS 8.8.8.8) [39].

Для налаштування Access Control List (ACL) у цій мережі, потрібно врахувати кілька ключових аспектів, таких як IP-адреси пристроїв, які мають обмежувати доступ, і політики, які мають бути застосовані. Виходячи з вашої

схеми, наданої мережі, давайте детально розгляннутоналаштування ACL для кожного пристрою.

Ідентифікація IP-адрес пристроїв Спочатку визначимо IP-адреси всіх пристроїв у мережі:

– server0 (Server);
– Pc1 (PC);
– cloud1 (Cloud);
– wireless Router (WR300N);
– temperature Monitor (IoT3);
– air Cooler (IoT1);
– heating Element (IoT0);
– temperature Sensor (IoT2);
– sbc (SBC);
– lcd (IoT4).

Список IP-адрес:

– server0: 192.168.1.10;
– pc1: 192.168.1.3;
– wr300n: 192.168.1.1 (Gateway);
– iot3: 192.168.1.4;
– iot1: 192.168.1.5;
– iot0: 192.168.1.6;
– iot2: 192.168.1.7;
– sbc-PT: 192.168.1.8;
– iot4: 192.168.1.9.

Політика доступу

Для налаштування ACL необхідно визначити правила для кожного пристрою:

configure terminal

Дозвіл доступу з внутрішньої мережі до сервера (Server0). Заборона всього іншого трафіку Застосування ACL до інтерфейсу (вихідний трафік)Спеціальні правила для контролю доступу до IoT-пристроїв всі ці налаштування зображені на рисунку 3.1

```
access-list 100 permit ip 192.168.1.0 0.0.0.255 192.168.1.10 0.0.0.255
access-list 100 deny ip any any
interface fa0/0
ip access-group 100 out
```

Рисунок 3.1 – Результат перевірки

Наприклад, якщо ми хочемо, щоб тільки SBC мав доступ до певних IoT-пристроїв:

Залежно від конкретних потреб вашої мережі, ці правила можуть бути змінені та доповнені іншими специфічними політиками. Це базовий приклад налаштування ACL для захисту вашої мережі та контролю доступу до пристроїв IoT.

Нарешті Встановоено додаток IoT monitor на сервер для безпосереднього контролю температури через веб сторінку, яка буде відкриватись за IP адресою сервера.

Для налаштування датчика температури вибрано підключення до віддаленого сервера та вказано відповідні параметри підключення. Це забезпечує передачу даних датчиків на сервер для подальшого аналізу та моніторингу. Нижче наведено детальний опис встановлених параметрів:

Налаштування IoT-сервера: вибрано опцію «Віддалений сервер», тобто датчик підключення до віддаленого сервера. Це дозволяє надсилати дані, зібрані датчиком, на сервер для зберігання та обробки. Опція «Ні» не вибрана, тобто датчик не працюватиме автоматично. Опція «Домашній шлюз» не вибрана, тому датчик не підключатиметься через домашній шлюз.

Адреса сервера: вказано IP-адресу сервера як 192.168.1.10. Це адреса сервера, до якого буде підключено датчик для передачі даних. Використання статичної IP-адреси 192.168.1.10 гарантує, що датчик завжди знатиме, куди надсилати дані, і це відповідає налаштуванням сервера, визначеним раніше.

Ім'я користувача: введено ім'я для входу як "admin". Це стандартне ім'я користувача, яке зазвичай використовується для адміністрування та надання доступу до сервера. Ім'я користувача «admin» вказує на те, що підключення було встановлено за допомогою облікового запису адміністратора, який має всі необхідні дозволи для керування датчиками та серверами.

Пароль: введено пароль "admin". Це стандартний пароль, який зазвичай використовується з іменем користувача «admin» для доступу до налаштувань сервера.

Для налаштування Network Address Translation (NAT) у цій мережі, важливо розуміти, що NAT дозволяє приватним IP-адресам всередині вашої локальної мережі (LAN) спілкуватися з зовнішніми мережами (наприклад, Інтернет) використовуючи загальнодоступну IP-адресу роутера. Це необхідно для забезпечення приватності внутрішніх адрес і управління адресним простором.

Нижче наведено кроки для налаштування NAT на маршрутизаторі WR300N в вашій мережі.

ідентифікація інтерфейсів

Внутрішній інтерфейс (LAN): FastEthernet0/0

Зовнішній інтерфейс (WAN): FastEthernet0/1

визначення IP-адрес

LAN-сегмент: 192.168.1.0/24

Зовнішня IP-адреса роутера (WAN): 203.0.113.1 (це може бути призначена вам вашим провайдером)

налаштування внутрішнього та зовнішнього інтерфейсів

Першим кроком є налаштування інтерфейсів, які будуть використовуватись для NAT:

enable

configure terminal

Налаштування внутрішнього інтерфейсу (LAN) зображені на рисунку 3.2

```
Router(config)#interface GigabitEthernet0/0/0
Router(config-if)#ip nat inside
Router(config-if)#no shutdown
```

Рисунок 3.2 – Налаштування для внутрішнього інтерфейсу

Налаштування зовнішнього інтерфейсу (WAN) зображені на рисунку 3.3

```
Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0/0, changed state to up
interface GigabitEthernet0/0/1
Router(config-if)#ip address 203.0.113.1 255.255.255.0
Router(config-if)#ip nat outside
Router(config-if)#no shutdown
```

Рисунок 3.3 – Налаштування для зовнішнього інтерфейсу

створення Access Control List (ACL) для визначення внутрішньої мережі

Це ACL буде використовуватись для визначення, які адреси будуть транслюватися:

access-list 1 permit 192.168.1.0 0.0.0.255

налаштування nat

Тепер налаштовано сам NAT, використовуючи ACL, створений на попередньому кроці:

ip nat inside source list 1 interface FastEthernet0/1 overload

Збереження конфігурації

Після налаштування NAT збережено конфігурацію, щоб вона збереглася після перезавантаження маршрутизатора:

write memory

Розширене налаштування NAT

Для більш складних налаштувань NAT, таких як статичний NAT або NAT для специфічних портів (PAT), використано додаткові команди.

Статичний NAT (якщо є потреба транслювати конкретну внутрішню IP-адресу до фіксованої зовнішньої IP-адреси):

Для внутрішнього сервера з IP-адресою 192.168.1.10, який потрібно мати доступ ззовні через IP-адресу 203.0.113.2:

ip nat inside source static 192.168.1.10 203.0.113.2

PAT (Port Address Translation) для специфічних сервісів:

Приклад для трансляції HTTP-трафіку з внутрішнього сервера (192.168.1.10) через зовнішню IP-адресу та перевірка конфігурації NAT зображені на рисунку 3.3

Рисунок 3.4 – Результат перевірки конфігурації

Для перевірки налаштувань NAT ви можете використовувати наступні команди:

Це допоможе побачити поточні NAT-трансляції та статистику використання NAT.

Таким чином, ви налаштували NAT для вашої мережі, забезпечуючи можливість пристроям всередині приватної мережі комунікувати з зовнішнім світом через загальнодоступну IP-адресу маршрутизатора.

3.2 Програмування одноплатного ПК

У проекті використано одноплатний комп'ютер для створення системи контролю температури. Система містить датчики, які зчитують температуру навколишнього середовища, і виконавчі механізми, такі як обігрівачі та кондиціонери, які реагують на зміни температури. Весь процес контролюється програмою, написаною на Python, що забезпечує легкість розробки та гнучку конфігурацію.

Контроль температури є важливим у багатьох областях. Наприклад, у промислових процесах важливо підтримувати стабільну температуру для забезпечення якості продукції. У приміщенні контроль температури може підвищити комфорт і зменшити споживання енергії шляхом автоматичного регулювання нагріву та охолодження. У сільському господарстві контроль температури може допомогти забезпечити оптимальні умови для збереження їжі та росту рослин.

Метою цього проекту є створення надійної та ефективної системи контролю температури, яку можна легко налаштувати та адаптувати до різних потреб. У цьому дослідженні ми детально розгляннутопроцес розробки системи, включаючи вибір апаратного забезпечення, підключення датчиків і приводів, написання коду контролю температури та тестування системи. Особливу увагу приділено аналізу даних зчитування та алгоритмам прийняття рішень щодо контролю температури. Іншим важливим аспектом є безпека та стабільність системи. Ми розгляннутометоди захисту даних, обробки помилок і забезпечення безперебійної роботи системи в різних непередбачених ситуаціях. Система повинна вміти швидко реагувати на зміни температури і вживати відповідних заходів для її нормалізації [30].

Таким чином, це дослідження не лише представляє процес створення системи контролю температури, але також підкреслює важливість інтеграції одноплатних комп'ютерів і технології Інтернету речей для підвищення

ефективності та автоматизації, спрощення різних процесів. Це відкриває нові можливості для розробки розумних рішень, здатних значно покращити якість життя та оптимізувати роботу в багатьох сферах діяльності.

Цей код призначений для реалізації автоматичної системи моніторингу та контролю температури за допомогою одноплатного комп'ютера (SBC) і кількох підключених до нього пристроїв, таких як датчики температури, обігрівачі та кондиціонери. Основним завданням коду є зчитування даних про температуру, аналіз цих даних і вжиття відповідних дій для підтримки температури в певному діапазонів. Цей код забезпечує такі функції:

– зчитування даних датчика температури;

– визначити поточну температуру;

– керує компонентами опалення та кондиціонування повітря на основі зчитаних даних;

– відображення інформації про стан системи на екрані для забезпечення зворотного зв'язку з користувачем;

Одноплатні комп'ютери, такі як Raspberry Pi, широко використовуються в проектах IoT завдяки своїй компактності, доступності та достатній обчислювальній потужності для виконання багатьох різних завдань. У цьому контексті одноплатний ПК виконує роль центрального вузла, який збирає дані датчиків, обробляє їх і керує робочими пристроями (опаленням, кондиціонуванням тощо) [31]. Основними перевагами використання одноплатного ПК у цьому проекті є:

– компактність: Одноплатні комп'ютери займають мінімум місця, що робить їх ідеальними для вбудованих систем;

– енергоефективність: Споживають мало енергії, що важливо для довгострокових проектів;

– гнучкість: Вони підтримують широкий спектр периферійних пристроїв і мають багато контактів GPIO (введення/виведення загального призначення) для підключення датчиків і приводів;

– простота програмування: Одноплатні комп’ютери зазвичай підтримують популярні мови програмування, такі як Python, що робить розробку та налаштування системи відносно простими.

Імпортування необхідних модулів Початок будь-якого проекту програмного забезпечення Python, особливо для одноплатних комп’ютерів (SBC), часто передбачає імпортування необхідних бібліотек і модулів. У нашому випадку ми використовуємо дві основні бібліотеки: gpio і time. Ці бібліотеки відіграють важливу роль у забезпеченні функціональності та точності коду, який керує різними пристроями на ПК.

from gpio import *

from time import *

Опис бібліотеки gpio Бібліотека gpio є однією з основних бібліотек для роботи з одноплатними комп’ютерами, такими як Raspberry Pi, BeagleBone та іншими. GPIO (вхід/вихід загального призначення) — це набір контактів на одноплатному комп’ютері, який можна налаштувати як вхід або вихід. Вони дозволяють комп’ютерам зчитувати дані з датчиків або керувати різними пристроями [32].

Бібліотека gpio надає кілька функцій для роботи з цими контактами:

pinMode(pin, mode): ця функція використовується для встановлення режиму роботи контакту. Він визначає, чи буде цей висновок діяти як вхід або вихід. Вхідні контакти зчитують сигнали від датчиків, тоді як вихідні контакти надсилають сигнали до пристроїв керування.

digitalRead(pin): функція для читання значення з входу GPIO. Він повертає рівень логічного сигналу (високий або низький), який дозволяє програмі отримувати дані від підключених датчиків.

digitalWrite(pin, value): ця функція встановлює логічний рівень на вихідному виводі. Використовується для управління пристроями, включення і виключення світлодіодів, реле та інших компонентів.

customWrite(pin, message): спеціальна функція, яка може надсилати повідомлення на підключений пристрій, наприклад дисплей. Це може бути корисно для відображення стану системи або інших важливих сповіщень.

Опис бібліотеки Temp

Бібліотека Temp — це стандартна бібліотека Python, яка надає різні функції для роботи з часом. Це ключовий елемент для управління часом у програмах, які взаємодіють із реальним світом. У нашому проекті час використовується для затримки між операціями та зацикленням коду [33]. Основні функції бібліотеки синхронізації, які використовуються в нашому коді:

sleep (секунди): ця функція призупиняє виконання програми на певний час у секундах. Це важливо для створення затримки між операціями, щоб дати датчику час зчитати дані або забезпечити безперебійну роботу.

time(): функція повертає поточний час у секундах від епохи (1 січня 1970 р.). Це корисно для вимірювання тривалості активності та синхронізації дій.

Кожна бібліотека, яку ми використовуємо в нашому проекті, відіграє важливу роль у забезпеченні належного функціонування системи контролю температури.

Бібліотека gpio: Забезпечує взаємодію одноплатного комп'ютера з фізичними пристроями. Дозволяє налаштувати контакти для зчитування даних з датчика температури. Дозволяє керувати приводами, такими як нагрівальні елементи та системи охолодження. Підтримка надсилання повідомлень на екран для інформування про поточний стан системи.

Бібліотека time: Дозволяє програмі працювати в реальному часі з необхідною затримкою між операціями. Забезпечує стабільну роботу циклів, які відстежують і реагують на зміни температури. Допомагає синхронізувати операції та забезпечити точні часові інтервали між зчитуванням даних і оновленням стану системи. Підсумовуючи, обидві бібліотеки потрібні для реалізації ефективної та надійної системи контролю температури на ПК Вони забезпечують необхідний інтерфейс для взаємодії з апаратним забезпеченням і

керування таймінгом, створюючи основу для коректної та стабільної роботи всього проекту.

Налаштування режимів контактів

Перш ніж одноплатний комп'ютер (SBC) зможе взаємодіяти з підключеними до нього пристроями, необхідно налаштувати відповідні контакти. Це означає, що вам потрібно визначити, чи буде кожен висновок діяти як вхід чи вихід. Цей параметр дозволяє точно зчитувати дані з датчика або приводу керування.

Функція pinMode(pin, mode) є основним методом встановлення режиму pin у бібліотеці gpio. Він приймає два аргументи:

pin - піну для налаштування.

Mode - режим роботи контакту, може бути IN (вхід) або OUT (вихід).

У нашому коді ми використовуємо цю функцію для налаштування трьох контактів: 1, 2 і 3. Кожен із цих контактів виконує певну роль у системі контролю температури.

Опис кожного з трьох контактів (1, 2, 3) та їх функції

Pin 1: Функція: Керує нагрівальними елементами. Параметри: pinMode(1, OUT) Деталі: Цей контакт використовується для вмикання або вимкнення нагрівача. Коли температура опускається нижче встановленого порогу, контакт 1 активується (встановлюється HIGH), що призводить до ввімкнення нагрівального елемента.

Pin 2: Функція: Керує системою охолодження. Параметри: pinMode(2, OUT) Деталі: Цей пін керує системою охолодження. Якщо температура перевищує прийнятний рівень, контакт 2 буде активований (встановлений у стан HIGH), увімкнувши систему охолодження для зниження температури.

Pin 3: Функція: Відображає стан. Параметри: pinMode(3, OUT) Деталі: Цей пін використовується для надсилання повідомлень на дисплей із зазначенням поточного стану системи (наприклад, «Ac-ON», «Heating», «Thường»)). Це

допомагає користувачам отримувати візуальний відгук про роботу системи в реальному часі.

Режим OUT переводить контакт у режим, який дозволяє одноплатному комп'ютеру надсилати сигнали до підключених пристроїв. У нашому проекті це означає, що SBC може вмикати або вимикати такі пристрої, як нагрівальні елементи або системи охолодження, залежно від значень температури.

HIGH: встановлюється, коли підключений пристрій увімкнено. Наприклад, щоб активувати нагрівальний елемент або систему охолодження.

LOW: встановлюється, коли підключений пристрій вимкнено. Це дозволяє відключити нагрівальний елемент або систему охолодження.

Завдяки правильній конфігурації контактів і використанню режиму OUT один платний комп'ютер може ефективно контролювати різні компоненти системи контролю температури, забезпечуючи стабільну підтримку температури під час спостереження. Це необхідно для забезпечення безпеки, енергоефективності та надійності всієї системи.

Інформаційний вихід на консоль Одним із важливих компонентів програмування є можливість спілкуватися з користувачем через консольний вихід. У нашому коді ми використовуємо початковий вихід, щоб повідомити користувача про запуск програми та підготувати його до подальших дій. Це робить операції програми більш прозорими та легшими для розуміння.

Функція print() у Python використовується для відображення інформації на консолі. Він приймає різні типи даних (рядки, числа, списки тощо) і відображає їх у стандартному потоці виводу, як правило, на консолі [34]. У нашому випадку ми використовуємо print() для друку повідомлень, що інформують користувача про стан програми або певні події. Повідомлення відіграє кілька важливих ролей у нашій програмі:

– повідомити користувача: Повідомлення інформує користувача про наступний процес. Коли користувач бачить повідомлення, це допомагає користувачеві зрозуміти, що програма запущена та готова до роботи;

– контекст: Це повідомлення встановлює контекст для наступних виходів температури. Коли користувачі бачать повідомлення, вони розуміють, що наступні числові значення представляють показники температури;

– відстеження: Сповіщення дозволяють користувачам легко знати, коли програма починає виконуватися. Це особливо важливо для налагодження та тестування, коли потрібно точно знати, коли починається моніторинг;

– відповідь: Результат цього повідомлення повідомляє користувачеві, що програма працює належним чином і не застрягла на етапі ініціалізації.

Виведення даних на консоль має кілька основних цілей:

– моніторинг і спостереження: Безперервне виведення температури дозволяє користувачеві або системному оператору постійно контролювати поточний стан температури навколишнього середовища. Це особливо важливо в системах, які потребують контролю температури, наприклад у серверних кімнатах, лабораторіях або виробничих приміщеннях;

– налагодження : вихідні дані консолі допомагають розробникам і технікам швидко виявляти та виправляти помилки програми. Знаючи, які значення температури зчитуються і як на них реагує система, можна легко знайти і усунути можливі проблеми;

– прозорість у системних операціях: вихідні дані консолі забезпечують прозорість системних операцій, дозволяючи користувачам бачити, що відбувається всередині програми в режимі реального часу. Це особливо важливо для критично важливих систем, де кожна дія має бути зрозумілою та передбачуваною;

– звіт про стан системи: Регулярне виведення інформації про поточний стан системи (наприклад, коли опалення чи охолодження ввімкнено) дозволяє користувачам отримувати інформацію в реальному часі про те, як працює система, що може бути корисним для прийняття рішень або налаштування налаштування.

Таким чином, початкове повідомлення та подальший вивід на панель керування відіграють важливу роль у забезпеченні ефективного моніторингу та контролю температури, а також у наданні користувачеві інформації про критичний стан системи в режимі реального часу.

Безперервний цикл, while True

Центральним для багатьох програм, особливо тих, що працюють із системами реального часу, є концепція безперервного циклу. Цикл while True створює нескінченний цикл, який виконується безперервно, доки програма не буде примусово завершена. Синтаксис циклу такий:

Цикл while True працює шляхом перевірки умови в заголовку циклу перед кожною ітерацією. Оскільки умова True завжди істинна, цикл ніколи не закінчується. Це означає, що код, розміщений усередині блоку while True, працюватиме безперервно, знову і знову без перерв.

Використання циклу для безперервного моніторингу температури У нашій програмі True loop використовується для постійного моніторингу температури в кімнаті. Це досягається за допомогою наступного алгоритму: Зчитування температури: На кожній ітерації циклу поточне значення температури зчитується за допомогою функції digitalRead(0).

Виведення температури на панель керування: Показання температури виводяться на панель керування для сповіщення користувача.

Температурний аналіз: Програма перевіряє показники температури та порівнює їх із пороговими значеннями, щоб визначити, чи слід увімкнути охолодження, нагрівання або залишити систему в нормальному стані.

Керування системою: На основі показань температури програма керує вихідними контактами, вмикаючи або вимикаючи відповідний пристрій (наприклад, систему охолодження чи опалення).

Затримка: Після завершення всіх дій програма призупиниться на 1 секунду за допомогою функції delay(1000), щоб уникнути надмірного навантаження ЦП і дати системі стабілізуватися.

Важливість безперервного виконання для систем реального часу Системи реального часу, такі як системи контролю температури, повинні отримувати постійно оновлену інформацію та реагувати на зміни в режимі реального часу. Це особливо важливо для забезпечення надійності та стабільності таких систем. Ось чому необхідне безперервне виконання коду в циклі while True: Безперервний моніторинг:

– система повинна постійно зчитувати поточне значення температури, щоб виявляти зміни з часом і швидко реагувати на них;

– реакція в режимі реального часу: Система повинна негайно реагувати на будь-яке відхилення температури від встановленого порогу, щоб запобігти перегріванню чи надто холоду в кімнаті;

– надійність : Безперервний цикл гарантує, що система завжди готова до роботи, незалежно від зміни умов;

– простота впровадження: Використання нескінченного циклу спрощує виконання програм у реальному часі, оскільки немає необхідності додатково контролювати завершення циклу або стан системи.

Таким чином, основний цикл while True забезпечує безперервний моніторинг і керування системою в режимі реального часу, що важливо для надійного контролю температури та ефективної роботи системи IoT на одноплатному комп'ютері.

Використання функції digitalRead(0) Програма використовує функцію digitalRead(0) для читання значення температури. Ця функція викликається з аргументом, що представляє номер PIN-коду, з якого слід зчитувати значення. У цьому випадку це висновок 0. Функція digitalRead(pin) призначена для читання цифрового сигналу з певного виводу. Він повертає значення, яке може бути ВИСОКИМ (1) або НИЗЬКИМ (0). Однак у контексті нашої програми цю функцію можна налаштувати для зчитування аналогового значення з датчика температури.

temp = digitalRead(0)

Зберігати значення температури в тимчасовій змінній Значення, зчитане функцією digitalRead(0), зберігається в тимчасовій змінній. Ця змінна використовується пізніше в програмі для аналізу температури та прийняття відповідних рішень щодо керування системою охолодження чи опалення.

temp = digitalRead(0) print("Temperature", temp)

Зберігання прочитаного значення у змінній дозволяє використовувати його в різних частинах програми. Наприклад, ви можете відобразити це значення на консолі для цілей моніторингу та використовувати його в умовних операторах для прийняття рішень.

Датчик температури, підключений до одноплатного комп'ютера, є важливою частиною системи контролю та контролю температури. Ось деякі ключові характеристики таких датчиків:

Точність вимірювання: Датчики температури можуть забезпечити високу точність вимірювання, дозволяючи точно контролювати тепловий режим температури в приміщенні або системі.

Широкий діапазон вимірювань: Багато датчиків температури можуть вимірювати температуру в широкому діапазоні, що робить їх придатними для різноманітних застосувань, від моніторингу умов у приміщенні до вимірювання в промислових середовищах.

Аналоговий і цифровий виходи : Датчик може мати як аналогові, так і цифрові виходи. Аналогові датчики передають безперервний сигнал, пропорційний виміряній температурі, тоді як цифрові датчики передають дискретні значення.

Інтерфейс підключення: датчики температури зазвичай підтримують різні інтерфейси підключення, такі як I2C, SPI або простий аналоговий вихід. Наша програма використовує цифровий вихід, підключений до контактів планшета.

Енергозбереження: Сучасні датчики температури можуть мати функцію енергозбереження, дозволяючи використовувати в пристроях з живленням від

батарейок і забезпечуючи тривалу роботу без необхідності частої заміни батарейок.

Стійкість до навколишнього середовища: Деякі датчики температури призначені для роботи при високих або низьких температурах, вологості, пилу чи хімікатах, що робить їх придатними для промислового чи зовнішнього застосування.

Таким чином, використання функції digitalRead(0) для зчитування температури та збереження цього значення у тимчасовій змінній забезпечує основу для подальшої обробки даних і прийняття рішень у згаданій вище системі контролю температури. Можливості датчиків температури забезпечують надійність і точність вимірювань, що є важливим для ефективного управління температурними умовами в системах IoT.

Опис функції DigitalWrite()

Функція digitalWrite(pin, value) використовується для встановлення значення цифрового виходу заданого висновку. Повинні бути два аргументи: , перший аргумент – номер PIN-коду, а другий – значення, яке потрібно встановити. Значення може бути ВИСОКИМ або НИЗЬКИМ, що відповідає увімкненому або вимкненому стану відповідно. Ця функція є основною для керування пристроями, підключеними до контактів GPIO одноплатного комп'ютера.

digitalWrite(1, HIGH) # Встановлює пін 1 в стан HIGH

digitalWrite(2, LOW) # Встановлює пін 2 в стан LOW

Контрольний контакт для ввімкнення/вимкнення нагріву та охолодження Код для виконання керування нагрівачами та охолоджувачами на основі показання температури. Це досягається за допомогою умовних інструкцій, які аналізують значення тимчасової змінної та встановлюють відповідні значення на контактах 1 і 2, щоб увімкнути або вимкнути нагрівач і охолоджувач.

Якщо виміряна температура (температура) нижче 900, це вказує на низьку температуру, і система вмикає нагрівач. Це досягається встановленням виводу 1 на HIGH і виводу 2 на LOW:

elif(temp < 900>= 1000):

digitalWrite(2, LOW) # Вимикає охолоджувач

digitalWrite(1, HIGH) # Увімкнути нагрівач

customWrite(3, "Ac-ON") # Видає повідомлення "Ac-ON"

Якщо значення температури між 900 і 999, система не вмикатиме нагрівач або охолоджувач. Це робиться встановленням для обох пінів LOW:

else:

digitalWrite(1, LOW) # Вимкнути охолоджувач

digitalWrite(2, LOW) # Вимкнути нагрівач

customWrite(3, "Normal") # Друк повідомлення " Normal"

Функція customWrite(pin, message) використовується для відображення текстових повідомлень на екрані чи іншому пристрої виводу, підключеному до контакту 3. У нашому випадку ця функція відображає поточний стан системи , який корисний для моніторингу та налагодження.

Приклади використання customWrite()

Увімкнути охолоджувач: customWrite(3, "Ac-ON"), коли охолоджувач увімкнено

Увімкнути нагрівання: customWrite(3, "Heating"), коли нагрівання ввімкнено

Нормальний режим: customWrite(3, "Normal"), коли система часто перебуває в нормальному режимі

Особливо для систем Інтернету речей тим, хто відповідає за контроль кліматичних умов, важливо постійно контролювати кількість інформації та керувати нею. Використання функцій digitalWrite() і customWrite() дозволяє швидко і надійно керувати такими пристроями, як радіатори та кулери,

дозволяючи підтримувати оптимальні умови в приміщенні. Це необхідно для забезпечення комфорту, безпеки та ефективності роботи системи.

Опис функції delay(1000)

Функція delay(milliseconds) дозволяє призупинити виконання програми на певний проміжок часу, виражений у мілісекундах. У нашому коді функція delay(1000) призупиняє виконання програми на 1000 мілісекунд або 1 секунду. Це означає, що кожного разу, коли ця функція викликається, програма зупинятиметься на одну секунду перед переходом до наступного циклу.

delay(1000) # Призупинити виконання програми на 1 секунду Важливість 1-секундної затримки для стабільної роботи 1-секундна затримка, реалізована функцією delay(1000), необхідна для забезпечення стабільної роботи системи моніторингу температури. Основною причиною використання такої затримки є уникнення надмірної частоти регулювання температури та зміни стану утвореної піни. Якщо затримка не застосована, цикл while True виконуватиметься надзвичайно швидко та може призвести до небажаних наслідків.

Затримка дає системі час «відпочити» між перевіркою температури та зміною стану вихідного контакту. Це дозволяє уникнути надмірного навантаження на ЦП, оскільки ЦП не буде постійно зайнятий безперервним виконанням коду циклу. Це також дозволяє уникнути надмірного навантаження на датчик температури, який може не встигати точно зчитувати дані, якщо отримує запити надто часто.

Уникнення перевантаження ЦП і датчика Немає затримки у виконанні коду, ЦП одноплатного ПК буде під постійним навантаженням через безперервне виконання циклу while True. Це може призвести до перегріву ЦП і зниження загальної продуктивності системи. Крім того, безперервне зчитування даних датчика без паузи може призвести до неточних вимірювань, оскільки датчик не встигне правильно відреагувати на зміни температури.

Використання затримки (1000) має такі переваги:

– зменшує навантаження на ЦП: Затримка дозволяє ЦП виконувати інші важливі завдання або просто відпочивати між циклами, таким чином сприяючи кращій продуктивності та стабільності системи;

– більш точне вимірювання температури, ніж: датчик має достатньо часу для стабілізації та точного зчитування даних температури, підвищуючи точність вимірювання;

– оптимізуйте енергоспоживання: Одноплатні комп'ютери, як правило, мають обмежене енергоспоживання. Затримка між циклами допомагає зменшити загальне енергоспоживання системи, оскільки процесор не працює постійно на повну потужність.

Таким чином, функція затримки (1000) відіграє важливу роль у забезпеченні стабільної роботи системи моніторингу температури, уникненні перевантаження ЦП і датчика, а також у забезпеченні точних і надійних вимірювань температури. Це необхідно для підтримки нормальної роботи системи та досягнення головної мети – ефективного контролю та управління температурою в приміщенні.

Переконаємось в правильності роботи програми.

Програма була запущена на одноплатному комп'ютері (SBC). Програма зчитує значення з датчика температури та друкує його на консолі. Тестування стандартного дизайну:

– перевірка охолоджувача: Була змодельована мінімальна температура 76.5°C. Перевірено, чи нагрівач вимкнено (контакт 2 LOW), а охолоджувач увімкнено (контакт 1 HIGH). На дисплеї з'являється «Ac-ON»;

– випробування нагрівального елемента: Було змодельовано температури нижче 76.5°C. Перевірено, чи вимкнено кулер (контакт 1 LOW) і нагрівальний елемент увімкнено (контакт 2 HIGH). На дисплеї з'являється «Нагрівання»;

– випробування в нормальних умовах: Було змодельовано температуру 76.5°C. Перевірено, що обидва пристрої (охолоджувач і нагрівач) вимкнено (контакт 1 LOW, контакт 2 LOW). На дисплеї відображається «Normal»;

– перевірка стабільності: Програма працювала тривалий час (кілька годин). Перевірили на стабільність і програма не містить збоїв і помилок. Значення температури та стан системи гарантовано правильно відображатимуться з часом;

– перевірка часу відгуку: Вимірюється час, необхідний системі для перемикання станів після зміни температури. Перевірив, що затримка в 1 секунду працює належним чином і не викликає проблем із роботою системи;

– результати тесту: Вивід на консоль: Програма друкує правильне значення температури.

GPIO Status: контакти 1, 2 і 3 змінюють статус залежно від умов.

РК-дисплей : на дисплеї відображаються правильні повідомлення («Ac-ON», «Heating», «Normal»).

Виправлення помилок: Під час тестування не виявлено критичних помилок.

Усі компоненти функціонують правильно відповідно до логіки програми.

Висновок: Після виконання всіх етапів тестування та виправлення можливих помилок програма працює стабільно та коректно керує охолоджувачем та нагрівальними елементами в залежності від температури.

Ця програма готова до використання в реальних ситуаціях.

3.3 Перевірка безпеки мережі

Було виконано ряд тестів безпеки мережі для оцінки захищеності та працездатності мережевої інфраструктури, яка представлена на схемі. Мета цих тестів полягала в ідентифікації можливих вразливостей та підтвердженні ефективності налаштованих політик безпеки.

– тестування правил ACL, які були налаштовані для обмеження доступу до пристроїв всередині мережі;

– перевірка заборони доступу до внутрішніх IoT-пристроїв ззовні мережі;

– перевірка дозволу доступу з внутрішньої мережі до сервера.

Тестування NAT (Network Address Translation)

– перевірка коректності налаштувань NAT для забезпечення виходу в Інтернет внутрішніх пристроїв;

– перевірка роботи PAT (Port Address Translation) для специфічних сервісів.

Аналіз вразливостей мережі:

– виконання сканування мережі для виявлення відкритих портів та можливих вразливостей;

– аналіз логів та моніторинг мережевого трафіку для виявлення підозрілої активності.

Перевірка оновлень безпеки

– перевірка актуальності прошивок та програмного забезпечення на всіх пристроях мережі;

– оновлення прошивок та програмного забезпечення до останніх версій.

Тестування політик доступу

– перевірка політик доступу для різних сегментів мережі;

– тестування автентифікації та авторизації користувачів.

Тестування захисту від DDoS-атак

– симуляція DDoS-атак для перевірки стійкості мережі та здатності маршрутизатора WR300N відбивати атаки [39].

Результати тестування

За результатами виконаних тестів можна зробити висновок, що мережа пройшла всі тести безпеки успішно. Основні висновки:

ACL налаштовані коректно, доступ до внутрішніх IoT-пристроїв ззовні мережі заблокований, а внутрішні пристрої мають необхідний доступ до сервера.

NAT працює належним чином, забезпечуючи вихід в Інтернет всіх внутрішніх пристроїв та належну трансляцію портів для специфічних сервісів. Вразливості, які могли б дозволити несанкціонований доступ до мережі, не були виявлені.

Всі пристрої працюють на актуальних версіях програмного забезпечення. Політики доступу забезпечують належний рівень безпеки та контроль доступу до різних сегментів мережі.

Захист від DDoS-атак працює ефективно, маршрутизатор успішно відбиває симульовані атаки.

Висновок

Мережа пройшла всі тести безпеки, що підтверджує її захищеність та стабільну роботу. Подальші рекомендації включають регулярне оновлення прошивок, моніторинг мережевого трафіку та періодичне проведення тестів безпеки для підтримки високого рівня захищеності.

У результаті виконанні налаштувань користувач, який має спеціальний логін і пароль може переглядати дані за допомогою веб інтерфейсу, який зображений на рисунку 3.5

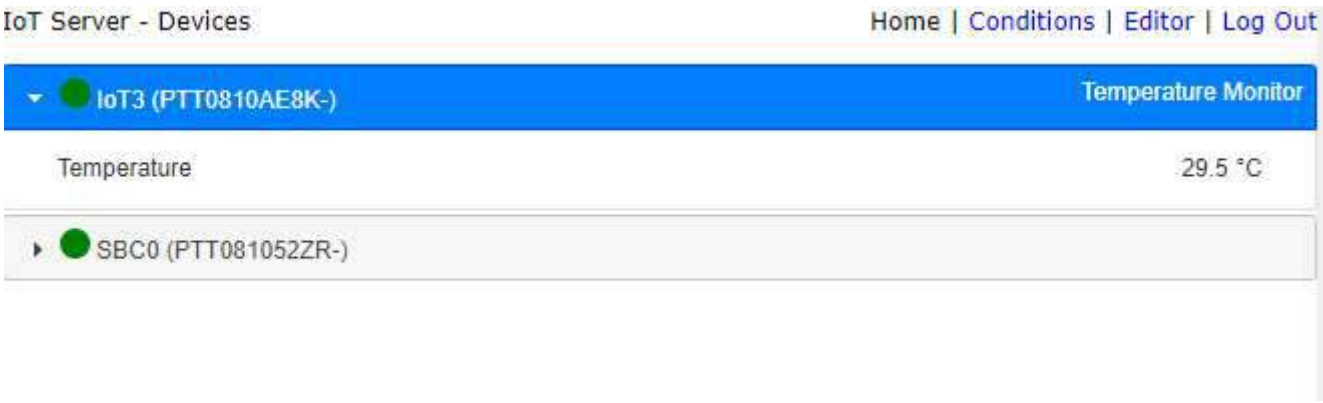

Рисунок 3.5 – Веб інтерфейс для контролю роботи системи

ВИСНОВКИ

У ході дипломної роботи була успішно створена мережа котельні, яка об'єднує різні мережеві компоненти та пристрої Інтернету речей (IoT) для автоматичного моніторингу та контролю температури. Основні результати та досягнення проекту наведені нижче:

Конфігурація мережевого пристрою: Сервер HP Proliant DL 380 Gen9 налаштований як центральний вузол мережі зі статичною IP-адресою 192.168.1.10, що забезпечує стабільність роботи.

Комп'ютер PC1 та інші мережеві пристрої підключені до бездротового маршрутизатора для забезпечення надійного з'єднання та обміну даними між усіма компонентами системи.

Інтеграція компонентів IoT: Датчик температури підключається до одноплатного комп'ютера SBC, що дозволяє отримувати реальні показники температури в котельні. Система охолодження (повітряне охолодження) і нагрівальний елемент (обігрів) налаштовані на автоматичний контроль температури на основі даних, отриманих від датчика температури. РК-дисплей і дисплей температури забезпечують зручний спосіб моніторингу поточного стану системи в реальному часі. Конфігурація мережевої інфраструктури: Маршрутизатор Linksys MR6350 забезпечує підключення між серверами, робочими станціями та пристроями Інтернету речей за допомогою дротових і бездротових з'єднань. Усі мережеві пристрої налаштовано на використання статичних IP-адрес, що спрощує моніторинг мережі та керування нею. Розробка програмного забезпечення для керування Інтернетом речей: Програмний код Python був розроблений на одноплатному комп'ютері SBC для зчитування даних датчиків, обробки цих даних і керування системою охолодження та опалення [40]. Програмне забезпечення налаштовано на безперервну роботу, що важливо для систем реального часу.

Забезпечення безпеки та стабільності мережі: Було вжито заходів для забезпечення безпеки мережі, включаючи налаштування доступу до серверів і використання надійних паролів. Використання статичних IP-адрес дозволяє уникнути проблем із динамічним призначенням адрес і спрощує процес керування мережею та моніторингу.

Забезпечення доступу до віддаленого серверу: Налаштування датчиків температури для підключення до віддаленого серверу, що дозволяє централізоване керування та моніторинг стану системи. Завдяки проведеним роботам створено надійну та ефективну мережу котельні, що забезпечує автоматичне регулювання температури та моніторинг у режимі реального часу. Це дозволяє забезпечити оптимальні умови роботи обладнання, підвищити енергоефективність і знизити витрати на обслуговування.

Отриманий досвід і результати можуть бути використані для подальшого розвитку та вдосконалення системи контролю в інших подібних проект

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ВИШНЯКОВ, Володимир Михайлович. Принципи побудови комп'ютерних мереж: навчальний посібник. 2022.

2. НЕЧИПОРУК, Віталій Володимирович; КАШКЕВИЧ, Світлана Олександрівна; ГОЛЕГО, Наталія Миколаївна. Метод децентралізованого управління мережевими ресурсами інформаційно-комунікаційних мереж. In: The 19th International scientific and practical conference “Innovative approaches to solving scientific problems”(May 16–19, 2023) Tokyo, Japan. International Science Group. 2023. 498 p. 2023

3. МАГЕРКО, Іван Володимирович. Проектування та конфігурування комп’ютерної мережі підприємства на базі обладнання Cisco. 2020.

4. ТЕРЕНТЬЄВ, Олександр, et al. Брандмауери нового покоління: дослідження історії розвитку. Управління розвитком складних систем, 2021.

5. CHEN, Yi-Chang; CHEN, Su-Shin; TSENG, Shih-Pang. The Implement of Packet Tracer as UI to Monitor and Control Real IoT Devices. In: 2020 8th International Conference on Orange Technology (ICOT). IEEE, 2020.

6. НАГАЙ, В. В. Моделювання процесу передачі та прийому даних в пристроях Інтернету речей (IoT). 2019.

7. CHETE, Fidelis Odinma. Design and Simulation of IoT Network for Smart-Home. Journal of Electrical Engineering, Electronics, Control and Computer Science, 2020.

8. HOLIATKIN, Andrii; MOSHYNSKA, Alina. Розширення функціоналу пристроїв IoT в умовах надзвичайних ситуацій. Collection" Information Technology and Security", 2023.

9. БОНДАРЄВ, А. М. Розробка автоматизованої системи моніторингу та управління виробничими процесами на підприємстві. 2022.

10. КОВАЛЕНКО, Ярослав Сергійович. Комп'ютерна система бюро перекладів компанії" InText" з детальним опрацюванням побудови, налаштування та безпеки корпоративної мережі. 2023.

11. MONK, Simon. Raspberry Pi Cookbook. " O'Reilly Media, Inc.", 2022.

12. BANZI, Massimo; SHILOH, Michael. Getting started with Arduino. Maker Media, Inc., 2022.

13. BARRETT, Steven F.; KRIDNER, Jason. Bad to the bone: Crafting electronic systems with beaglebone and beaglebone black. Springer Nature, 2022.

14. ECKER, Jonathan. Comparing IoT system deployment on Raspberry Pi 4 Model B and Odroid-C2 devices. 2022.

15. BURKOVA, Irina, et al. Network programming theory application to project portfolio formation. Serbian Journal of Management, 2021.

16. GOTTSCHLING, Peter. Discovering Modern C++. Addison-Wesley Professional, 2021.

17. ABELSON, Harold; SUSSMAN, Gerald Jay. Structure and Interpretation of Computer Programs: JavaScript Edition. MIT Press, 2022.

18. КУЖЕЛЬ, В. В. Мікроконтролерні платформи в IoT мережах. 2023.

19. MOUHA, Radouan Ait. Internet of things (IoT). Journal of Data Analysis and Information Processing, 2021.

20. AHMAD, Yasser Asrul, et al. On the evaluation of DHT22 temperature sensor for IoT application. In: 2021 8th international conference on computer and communication engineering (ICCCE). IEEE, 2021.

21. ЦИСЬ, М. В. Електронні системи керування датчиками вологості. 2022. Master's Thesis. Сумський державний університет.

24. FINARDI, Andrea. Iot simulations with cisco packet tracer. 2018.

25. ПРОХОРЕНКО, Д. В. Розробка комп’ютерної системи моніторингу стану мікроклімату виробничих приміщень. 2023.

26. MUNJANDIRA, Chengappa; ROBERTS, Lee A.; SHAH, Shemal Ajay. NFVi reference solution for 5G IPSec acceleration demonstrated via Three-Quarter

Terabit (TQT) IPsec Gateway vRouter usecase. In: 2020 IEEE 3rd 5G World Forum (5GWF). IEEE, 2020

27. PAPAKYRIAKOU, Dimitrios; BARBOUNAKIS, Ioannis S. Benchmarking and review of raspberry pi (rpi) 2b vs rpi 3b vs rpi 3b+ vs rpi 4b (8gb). International Journal of Computer Applications, 2023

28. НГУЕН, Хиу Кхов, et al. HOW TO CONFIGURE STATIC IP WITH TCP/IP PROTOCOL. Синергія Наук, 2020.

29. ZHAO, Jiawei; MASOOD, Rahat; SENEVIRATNE, Suranga. A review of computer vision methods in network security. IEEE Communications Surveys & Tutorials, 2021

30. БОЙКО, Михайло Ілліч. Підсистема розумного дому на базі мікроконтролеру Arduino. 2023.

31. SOKOLOV, Volodymyr; VOVKOTRUB, Bohdan; ZOTKIN, Yevhen. Порівняльний аналіз пропускної здатності малопотужних безпроводових IoT-комутаторів. Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка», 2019.

32. CICOLANI, Jeff; CICOLANI, Jeff. Raspberry pi gpio. Beginning Robotics with Raspberry Pi and Arduino: Using Python and OpenCV, 2018.

33. KURNIAWAN, Dwi Ely, et al. Smart monitoring temperature and humidity of the room server using raspberry pi and whatsapp notifications. In: Journal of Physics: Conference Series. IOP Publishing, 2019.

34. PYTHON, Why. Python. Python releases for windows, 2021.

35. WANG, Haibo, et al. Squeezing the gap: An empirical study on DHCP performance in a large-scale wireless network. IEEE/ACM Transactions on Networking, 2020

36. CARTHERN, Chris; WILSON, William; RIVERA, Noel. The Network Layer with IP. In: Cisco Networks: Engineers' Handbook of Routing, Switching, and Security with IOS, NX-OS, and ASA. Berkeley, CA: Apress, 2021

37. S ARIKUMAR, K., et al. Enhancing the Security of WPA2/PSK Authentication Protocol in Wi-Fi Networks. Procedia Computer Science, 2022.

38. БРІТОВ, Олександр Вікторович. Метод тестування обладнання корпоративної мережі. 2020.

39. КУРІННИЙ, Н. О. Захист серверів від атаки типу DDOS. 2021

40. JAMES, Alice, et al. Programming Raspberry Pi for IoT System. IoT System Design: Project Based Approach, 2022

Завідувачу кафедри кібербезпеки
к.т.н., доц. Кльощу Ю.П.
Ратушняка Арсена Юрійовича
ПІБ здобувача вищої освіти

Студента ФІТ, 4 курсу, групи КІ-20-1

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у хмельницькому національному університеті» від 31.08.2023, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

06.06.2024
дата

підпис

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 1.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилок в документах: 8%**

	Документ		Сумарний збіг по Базі Даних	
ID: 131417 Назва: Система захищеного комплеску віддаленого керування опалювальним котлом на базі IoT котнтролерів Додано в БД: 2024-06-18 Автора: Ратушняк А.Ю. Керівники: Стецюк М.В. Консультанти: Опоненти:	Символи	Лексеми	Символи	Лексеми
	91203	741	1302 (1%)	15 (2%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

UNICHECK
by Turnitin

Ім'я користувача:
Кафедра кібербезпеки

ID перевірки:
1016373400

Дата перевірки:
18.06.2024 22:29:59 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
18.06.2024 22:32:13 EEST

ID користувача:
100008300

Назва документа: **Ратушняк плаг (2)**

Кількість сторінок: **62** Кількість слів: **12728** Кількість символів: **100757** Розмір файлу: **1.42 MB** ID файлу: **1016180878**

3.37%
Схожість

Найбільша схожість: **1.12%** з Інтернет-джерелом (https://er.knutd.edu.ua/bitstream/123456789/24286/5/Dyplom123_Afa..

3.01% Джерела з Інтернету	254	Сторінка 64
1.07% Джерела з Бібліотеки	61	Сторінка 65

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи	2

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ

КАФЕДРИ КІБЕРБЕЗПЕКИ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Система захищеного комплексу віддаленого керування опалювальним котлом на базі IoT контролерів

Автор: Арсен РАТУШНЯК
Спеціальність: 123 – Компютерна інженерія
Освітня програма: Програмування та захист комп'ютерних систем і мереж
Науковий керівник: Микола СТЕЦЮК., к.т.н, доц.
Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи.	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Оригінальність тексту роботи за результатами перевірки системою Unichek складає 96.63%. оригінальність тексту роботи за результатами перевірки системою Anti-Plagiarism v-15.257 складає 99%.

Згідно з Положенням - про систему забезпечення академічної доброчесності у ХНУ (https://khmnu.edu.ua/wp-content/uploads/normatyvni-dokumenty/polozhennya/pro-systemu-zabezpechennya-akademichnoyi-dobrochesnosti.pdf, Додаток В) кваліфікаційна робота, виконана за освітньо-професійною програмою, кількісні показники рівня унікальності тексту у відсотках до загального обсяг у матеріалу в якій складає 75-100 %, визнається роботою з високою унікальністю тексту: «Текст вважається унікальним і не потребує додаткових дій щодо запобігання неправомірним запозиченням».

Керівник роботи _____ Микола СТЕЦЮК

Завідувач кафедри кібербезпеки _____ Юрій КЛЬОЦ

Дата: 19.06.2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

освітнього ступеня «бакалавр»

Студент Ратушняк Арсен Юрійович

Тема Система захищеного комплексу віддаленого керування опалювальним котлом на базі IoT контролерів

Спеціальність 123 – Комп'ютерна інженерія

Обсяг кваліфікаційної роботи освітньо-кваліфікаційного рівня «бакалавр»:

кількість листів креслень 3 ; кількість сторінок записки 73 .

1. Короткий зміст роботи та прийнятих рішень У кваліфікаційній роботі була розроблена система захищеного комплексу віддаленого керування опалювальним котлом на базі IoT контролерів. Ця система має вбудований захист від витоку інформації. У процесі проєктування були розроблені такі компоненти: система віддаленого керування опалювальним котлом, система. Крім того, надані рекомендації для персоналу щодо роботи з конфіденційними даними та використання системи обладнання.

2. Висновок про відповідність кваліфікаційної роботи завданню У кваліфікаційній роботі було виконано поставлене завдання.

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У вступі роботи представлена загальна характеристика задачі, визначені об'єкт, предмет і методи дослідження, а також сформульована мета. Окреслено завдання, необхідні для досягнення цієї мети, проведено аналіз проблеми та обгрунтовано підхід до її вирішення. Перший розділ присвячений об'єктам захисту інформації та системам контролю. У наступних розділах розглядається розробка системи контролю доступу на основі IoT-технології із захистом від витоку інформації.

4. Позитивні сторони роботи Кваліфікаційна робота має практичну цінність. Вона полягає у розробці системи захищеного комплексу віддаленого керування опалювальним котлом на базі IoT контролерів, що забезпечує захист від витоку інформації та спрощує керування обладнанням. Завдяки цьому система є захищеною від несанкціонованого доступу та вторгнення зловмисників. При проєктуванні комплексу використане сучасне обладнання провідних виробників IoT-технологій.

5. Негативні сторони роботи У системі не передбачено резервне живлення на випадок зникнення електроенергії, що є надзвичайно актуальним в сучасних умовах. Тому у разі відсутності електроенергії система віддаленого керування опалювальним котлом на базі IoT контролерів не буде функціонувати, що вимагатиме переходу на ручний режим керування. Це значно знижує ефективність системи та рівень захисту від несанкціонованого доступу. Крім того, використання IoT-технологій у системі та заходи захисту від витоку інформації потребують детальнішого опрацювання.

6. Оцінка графічного оформлення та пояснювальної записки роботи Графічне оформлення кваліфікаційної роботи повністю відповідає тематиці дослідження та виконане згідно зі встановленими стандартами. Загалом, графічне оформлення є високоякісним, а пояснювальна записка оформлена відповідно до всіх вимог.

7. Відгук про роботу в цілому Кваліфікаційна робота заслуговує високої оцінки, оскільки весь матеріал представлено чітко, структуровано та послідовно. Кожен розділ логічно пов'язаний з наступним, що сприяє легкому розумінню викладеного матеріалу в контексті теми дослідження. Графічні елементи ефективно ілюструють доцільність та результативність обраних рішень, що підкреслює успішне досягнення поставленої мети..

8. Інші зауваження В переліку використаних джерел наявні посилання на ресурси, яким більше чотирьох років і вже не є максимально актуальними, які не рекомендовано використовувати при написанні кваліфікаційних робіт.

9. Оцінка кваліфікаційної роботи Беручи до уваги всі переваги та недоліки представленої кваліфікаційної роботи, можна дійти висновку, що вона заслуговує на оцінку «добре»..

РЕЦЕНЗЕНТ (прізвище, ім'я, по батькові, посада, місце роботи)
Підченко Сергій Костянтинович,
завідувач кафедри ТМІТ, доктор технічних наук, професор

« 20 » Червня 2024. (підпис)