Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
(повна назва)
Кафедра Автоматизації проектування обчислювальної техніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)
(рівень вищої освіти)

Система дистанційного моніторингу мікроклімату

(тема)

Виконав: студент 2 курсу, групи СКСм-20-1
Попов Д.І.
(прізвище, ініціали)
Спеціальність *123 Комп'ютерна інженерія*
(код і повна назва спеціальності)
Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма *Спеціалізовані комп'ютерні системи*
(повна назва освітньої програми)
Керівник роботи доц. Хаханова Г.В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ Чумаченко С.В.
(підпис) (прізвище, ініціали)

2021 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
Кафедра Автоматизації проектування обчислювальної техніки
Рівень вищої освіти другий (магістерський)
Спеціальність 123 Комп'ютерна інженерія
Тип програми Освітньо-професійна
Освітня програма Спеціалізовані комп'ютерні системи
(шифр і назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 2021 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

Студентові Попову Денису Ігоревичу
(прізвище, ім'я, по батькові)

1. Тема роботи (проекту) Система дистанційного моніторингу мікроклімату

затверджена наказом по університету від "04" 11 2021 р. № 1635 Ст.
2. Термін подання студентом роботи (проекту) 24.12.2021
3. Вихідні дані до роботи (проекту)
Центральний пристрій на основі Raspberry Pi 3B
Розташована у хмарі система
Показники температури та вологості

4. Перелік питань, що потрібно опрацювати в роботі
Створення загальної структури системи
Вибір апаратних та програмних продуктів для створення системи
Встановлення програмного забезпечення у віртуальне оточення Python
Прошивка ZigBee модуля
Налаштування протоколів ZigBee2Mqtt та MqttBroker
Перевірка працездатності системи та точності вимірювання даних

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) 18 слайдів

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

7. Дата видачі завдання 20.10.2021

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання	20.10.2021 - 21.10.2021	
2	Аналіз предметної області	22.10.2021 - 23.10.2021	
3	Аналіз джерел з проблемної галузі	24.10.2021 - 06.10.2021	
4	Розробка підходу для реалізації поставленої	07.11.2021 - 10.11.2021	
5	Реалізація поставленої задачі	11.11.2021 - 20.11.2021	
6	Оформлення пояснювальної записки	21.11.2021 - 08.12.2021	
7	Оформлення графічного матеріалу	09.12.2021 - 12.12.2021	
8	Перевірка виконаного проекту керівником	13.12.2021 - 14.12.2021	
9	Захист проекту	23.12.2021 – 28.12.2021	
10			

Студент _____
(підпис)

Керівник роботи _____ _____ доц. каф. АПОТ Хаханова Г.В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка містить: 79 сторінок, 25 рисунків, 17 джерел за переліком посилань.

КОМП’ЮТЕРНА СИСТЕМА, МІКРОКЛІМАТ, ПЛАТФОРМА, СЕНСОР, СХЕМА, ІНТЕРФЕЙС, ІНФОРМАЦІЯ, ТЕХНОЛОГІЯ, БРОКЕР, ВРАЗЛИВІСТЬ

Метою кваліфікаційної роботи є створення моделі системи дистанційного моніторингу мікроклаімату.

Для досягнення поставленої мети були з’ясовані важливість вимірювання мікроклімату у різних сферах діяльності, розглянуто концепцію Інтернету речей, історію її створення, архітектуру, вразливості та складові, стан сучасного розвитку, технології безпровідного зв’язку, апартних та програмних продуктів для створення подібних систем, протоколів передачі даних, використання хмарних сервісів а також хмарних обчислень, таких як Fog та Edge. Були розглянуті вже існуючі системи, які використовуються у домашній та виробничій сфері.

На основі цього було обрано технології безпровідного зв’язку, пристрої з технологією Zigbee, мікроконтролер як основу центрального пристрою, протоколи передачі даних, операційну систему та мову програмування, а також хмарні сервіси. Задяки цим складовим система вийшла гнучкою та дешевою, і майже не відрізняється за надійністю та функціоналом від вже існуючих, більш дорогих систем.

ABSTRACT

The explanatory note of the qualification work: 79 pages, 25 figures, 17 sources.

COMPUTER SYSTEM, MICROCLIMATE, PLATFORM, SENSOR, SHEME, INTERFACE, INFORMATION, TECHNOLOGY, VULNERABILITY

The purpose of the certification work is to create a model of remote monitoring of microclimate.

To achieve this goal, the importance of measuring the microclimate in various fields was clarified, the concept of the Internet of Things, the history of its creation, architecture, vulnerabilities and components, the state of modern development, wireless technology, hardware and software products to create such systems. data transfer protocols, the use of cloud services, and cloud computing such as Fog and Edge. Existing systems used in the domestic and industrial spheres were considered.

Based on this, wireless technologies, Zigbee devices, a microcontroller as the basis of the central device, data protocols, operating system and programming language, as well as cloud services were chosen. Thanks to these components, the system turned out to be flexible and cheap, and almost does not differ in reliability and functionality from existing, more expensive systems.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

АЦП – аналогово-цифровий перетворювач;

ЦАП – цифро-аналоговий перетворювач;

IoT – Internet of Things (Інтернет речей);

M2M – Machine-to-Machine (спілкування типу Машина-Машина);

RFID – Radio Frequency IDentification (радіочастотна ідентифікація);

CC – Cloud Computing (хмарні обчислення);

FC – Fog Computing (туманні обчислення);

EG – Edge Computing (граничні обчислення);

MCC – Mobile Cloud Computing (мобільні хмарні обчислення);

IaaS – Infrastructure-as-a-Service (інфраструктура як сервіс);

PaaS – Platform-as-a-Service платформа як сервіс (платформа як сервіс);

SaaS – Software-as-a-Service програмне забезпечення як сервіс (програмне забезпечення як сервіс);

AES – Advanced Encryption Standard (розширений стандарт шифрування);

MITM – Man In The Middle (атака типу «людина посередині»);

Wi-Fi – Wireless Fidelity (бездротова мережа);

MQTT – Message Queuing Telemetry Transport (телеметрична передача повідомлень з чергою).

ВСТУП

Мікроклімат – поняття, яке включає в себе такі характеристики як температура, тиск, вологість наволишнього середовища та швидкість повітря. Потребність визначення температури була завжди актуальною, і майже у кожному будинку є термометр, але якщо необхідно дізнатися температуру на віддаленому об’єкті, або у небезпечному для житті людини місці, використовуються спеціалізовані комп’ютерні системи.

Ще п’ятдесят років тому такі системи мали великі розміри, а для з’єднання сенсорів та обчислювалюної техніки використовувалось велика кількість дротів, які могли простягатися на десятки і навіть сотні кілометрів. Такий принцип мав дуже складний процесс встановлення системи та пошуку несправностей – якщо якийсь дріт був перебитий, для пошуку цього місця треба було витрати дуже багату часу, або прокладати новий, що супроводжувалося великою витратою сил та ресурсів.

Окрім того такі системи не мали змогу зберігати великі обсяги відстежених даних, адже комп’ютерні системи того часу займали дуже багато місця, тож для фіксування показників їх просто записували у журнул. З того часу минуло багато років, і технології які є зараз дають змогу переосмислити побудову таких систем, при цьому заощаджуючи час та витрати.

Останніми роками відбувається стрімкий розвиток різноманітних технологій для використання бездротового зв’язку, і одна з них – це Інтернет речей, або IoT, основною задачею якого є об'єднання різноманітних приладів в одну єдину систему, зрозумілу для людини. Основною перевагою є те, що пристрої у такій системі можуть взаємодіяти між собою за допомогою

різноманітних алгоритмів, при цьому не потребуючи втручання самої людини. Такий принцип називається M2M, або Machine-to-machine.

Багато виробників у наш час розробляють та випускають спеціальзовані системи Інтернету речей, націлені на автоматизацію конкретних задач, відповідно до потреб замовника. Яскравим прикладом таких систем є розумний будинок. Такі системи можуть бути встановлені у будь-якому будинку, і потребують лише наявності джерела струму та безпровідної мережі, наприклад Wi-Fi.

Встановлюються різноманітні сенсори та виконавчі механізми по всьму будинку, а токож центральній пристрій для комунікації між приладами. Усе це з’єднується з хмарним сервером через мержу Інтернет, а користувач має змогу дивитися необхідну йому інформацію та керувати пристроями, знаходячись у будь-якій точці планети.

Окрім розумних будинків, системи, побудовані на концепції IoT, використовуються й у виробничих цілях. Такий підхід дозволяє копаніям заощаджувати велику кількість коштів, а завдяки інтеграції з хмарними сховищами забезпечити конфіденційність та безпеку даних. Окрім цього на виробництвах, яке має небезпечні умови для життя людини, такі системи дуже важливі. Але якщо системи для розумних будинків усі між собою дуже схожі, великим компаніям необхідно звертатися до розробників, які створять унікальну спеціалізовану систему для конкретних цілей.

Оскільки дана технологія все більше привертає до себе уваги, а її використання за різними прогнозами торкнеться кожну людину вже через десять років, проводяться багато досліджень у цій області, і з кожним роком виходять нові продукти, кращі за попередні. І з таким попитом та успіхом, одним з найважливіших питань залишається безпека. Існує дуже багато шкідливого програмного забезпечення, які використовують вразливості таких систем для отримання персональних даних, або даних конкретного

виробництва.

В даній роботі розглядається розробка системи для дистанційного моніторингу мікроклімату на базі мікроконтролеру, з використанням спеціалізованих приладів, програмного забезпечення а також хмарних сервісів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВА ЗАДАЧІ

1.1 Мікроклімат

Мікроклімат – будь-які кліматичні умови на відносно невеликій території, в межах кількох метрів або менше над і під поверхнею Землі та в межах рослинних покривів.

Кожен компонент мікрокліматичного середовища демонструє унікальні просторові та часові реакції на зміни прибережних структурних елементів. Крім того, взаємозв'язки між мікрокліматом і біологічними процесами є складними і часто нелінійними. Ці особливості легко уявити, якщо врахувати, що температура, сонячне випромінювання та вологість впливають на ріст рослин, впливаючи на фізіологічні процеси, такі як фотосинтез, дихання, проростання насіння, смертність та активність ферментів.

Найсильніші градієнти температури і вологості спостерігаються трохи вище і нижче земної поверхні. Складність мікроклімату необхідна для існування різноманітних форм життя, хоча будь-який окремий вид може переносити лише обмежений діапазон клімату, сильно контрастний мікроклімат у безпосередній близькості створює повне середовище, в якому багато видів флори та фауни можуть співіснувати та взаємодіяти.

Якщо ж роздивлятися міський мікроклімат, то тут все буде інакшим. Завдяки життєдіяльності людини температура в міському мікрокліматі вища, ніж в навколишніх територіях. Міські райони називаються міськими острівцями тепла, оскільки в спокійних умовах температура найвища в забудованому центрі міста і знижується до передмістя та сільської місцевості.

Існує кілька причин, чому виникає така закономірність. У міських районах будівельні матеріали не відбивають світло і тому поглинають тепло. Крім того,

дорожні покриття, такі як асфальт і бетон, мають високу теплоємність, тому також поглинають велику кількість тепла через свій темний колір. Це тепло поглинається протягом дня, а потім повільно виділяється вночі, підвищуючи температуру.

Подальше тепло виділяється наявністю заводів і збільшенням використання автомобілів у місті, що спричиняє забруднення, через що з’являється смог і утворюється купол забруднення. Цей купол забруднення дозволяє проникати короткохвильовій інсоляції, але затримує вихідне земне випромінювання завдяки своїй більшій довжині хвилі, таким чином збільшуючи кількість отриманого тепла. Усе це негативно впливає на людину.

Потрібність у вимірюванні мікроклімату була завжди, і сфери у яких це необхідно можуть бути дуже різними. Наприклад це може бути теплиця, у якій необхідно знати та слідкувати за температурою та вологістю для вирощення рослин, або велике підприємство, де відстежування температури у якомусь технологічному процесі може запобігти катастрофічним наслідкам, або ж звичайне вимірювання температури вдома.

1.2 Визначення поняття Інтернету речей

Для перенесення фізичних об’єктів в режим онлайн та можливості зробити так щоб вони між собою спілкувалися та співпрацювали між собою без участі людини, використовується платформи IoT (Internet of Things), або Інтернет речей, який забезпечує автоматизацію розумних пристроїв у мережах конкретної інфраструктури.

Можна сказати, що кожне середовище IoT – це комбінація технологій різних постачальників, які в об’єднанні створюють складну та різноманітну за своєю природою системою, зі спільною для їх інтеграції базою. Таким чином можна зробити висновок, що платформа інтернету речей являє собою точку

зустрічі для усіх підключених пристроїв та служить для збирання та обробки інформації, яка передається мережею.

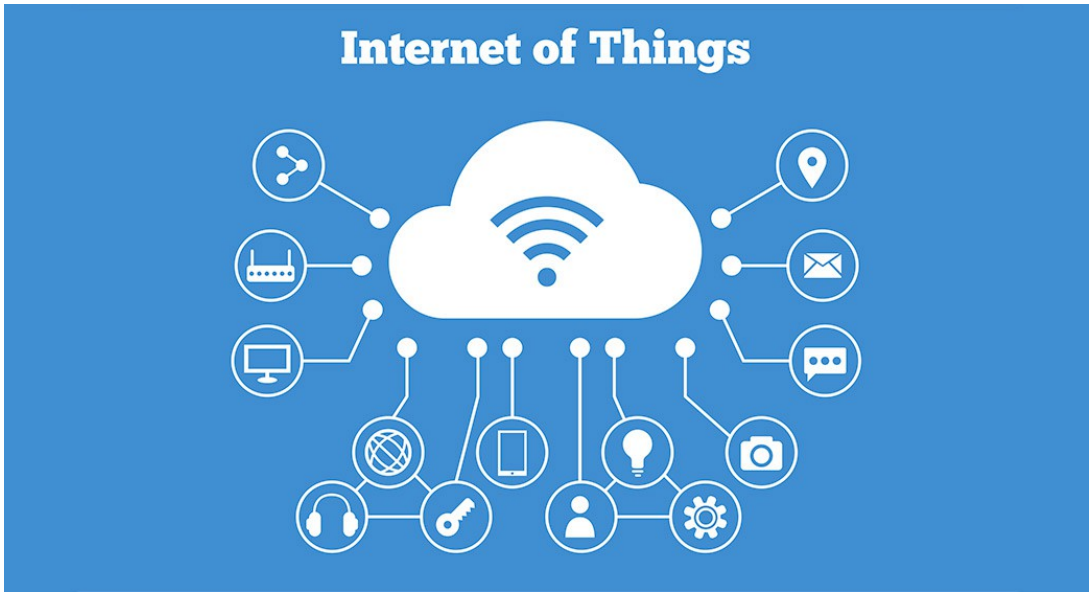

Рисунок 1.1 – Інтернет речей

Унікального визначення Інтернету речей, яке б було прийнято світовою спільнотою не має, хоч насправді існує багато різноманітних груп, які включають академіків, дослідників, практиків та розробників, які б могли визначити конкретний термін IoT. Але усі ці визначення об’єднує спільна ідея – перша версія IoT була про дані, створені людьми, наступні версії про дані, створені пристроями.

Найкращим визначенням Інтернету речей напевно є: «Відкрита та всеосяжна мережа інтелектуальних об’єктів, які мають здатність автоматично організовувати, обмінюватися інформацією, даними та ресурсами, реагуючи та діявши в умовах ситуацій та змін у навколишньому середовищі».

Також, концепцію Інтернету можна описати наступним чином:

Інтернет речей – це єдина мережа, що з'єднує пристрої з віртуальними об'єктами, що має на увазі тісну інтеграцію людей і різних предметів, підключених до мережі, а надалі практично повну їхню взаємодію.

Інтернет речей – це мережа мереж: безліч спеціальних датчиків і сенсорів

з'єднуються між собою, утворюючи мережі, які, у свою чергу, також з'єднані між собою, створюючи світову мережу мереж.

Інтернет речей – не лише основний тренд світу технологій, а й новий етап розвитку інтернету, що відкриває перед нами величезні можливості, з перспективою повної інтеграції віртуального та реального світів.

Дана концепція продовжує розвиватися та привертає все більше уваги до себе. Протягом останніх десятиліть термін інтернету речей привернув увагу, проектуючи бачення глобальної інфраструктури мережевих фізичних об’єктів, що забезпечують підключення в будь-який час і в будь-якому місці для будь-чого, а не тільки для когось [1].

Інтернет речей також прийнято розглядати як глобальну мережу, яка дозволяє спілкуватися між людиною та речами, які можуть бути чим завгодно у світі, забезпечивши унікальну ідентичність кожному об’єкту.

Інтернет очей описує світ, у якому майже все можна під’єднати у мережу та спілкуватися в розумній манері. Багато людей вважають, що термін з’єднання, з точки зору електронних пристроїв, це йдеться про комп’ютери, телефони, планшети або іншу портативну техніку.

Насправді, Інтернет речей означає підключення різноманітних датчиків або виконавчих механізмів, вбудованих у фізичні об’єкти, за допомогою дротових або бездротових мереж, часто використовуючи таку ж саму IP-адресу, яку надає Інтернет. Ці мережі виробляють та обробляють величезні обсяги інформації, які надходять для аналізу.

Коли об’єкти можуть відчувати навколишнє середовище та спілкуватися, вони стають інструментами для розуміння складності та швидкого реагування на неї. Революційним у всьому цьому є те, що ці фізичні інформаційні системи зараз починають розгортатися, а деякі з них навіть працюють переважно без участі людини.

Принцип роботи Інтернету речей часто описують як ABCDE:[2]

A – Analytics – аналітика. Об’єднує в єдину структуру фізичні пристрої, отримані з них дані та бізнес процеси. Без такої структури неможливо досягти

окупності інвестицій. Тобто IoT рішення не повинні коштувати більше ніж самі дані.

B – BigData – великі данні. Оскільки система може бути великою та складатися з великої кількості приладів, необхідно подбати про ефективний спосіб зберігання даних, а також про наповнення цих даних змістом. Фактично BigData – це великий файл, в якому зібрана вся пряма та непряма інформація з приладів. Усі дані, отримані від підключених датчиків, зберігаються у хмарному просторі. Вони дозволяють виявляти закономірності та автоматизувати існуючі процеси або вибудовувати нові. При цьому інформацію можна отримувати у режимі реального часу у двох варіантах – у графічній візуалізації або у вигляді історичної аналітики.

C – Connection – зв’язок. Для того, щоб пристрої у системі мали змогу обмінюватися даними, вони повинні буті з’єднані. І в залежності від потреб проекту з’єднання може бути провідним або безпровідним. Таким чином створюється єдина мережа. У зв'язку з цим зростає і вимога до самої мережі: вона повинна витримувати навантаження 24 години на добу 7 днів на тиждень та забезпечувати доступ до даних із пристроїв із ймовірністю 99,99% із мінімальними витратами. Також, необхідно звернути увагу до з’єднання сенсорів та хмарних сервісів, адже без підключення дані будуть заблоковані всередині датчиків.

D – Device – пристрій. Щоб вся система працювала злагоджено та без втрат, необхідно правильно підібрати пристрої, щоб вони мали змогу працювати один з одним. Усі пристрої повинні мати відповідну частоту повідомлень, причому залежно від об'єкта вона може бути різною (може змінюватись поріг виявлення, точність та інші параметри). Крім того, важливими є простота встановлення пристрою та термін його придатності, щоб він збігався з циклом життя об'єкта, на якому він буде розміщений. У такому разі можна уникнути додаткових витрат технічне обслуговування.

E – Experience – досвід. Оскільки системи будуються конкретно під задану проблему, нестача досвіду може привести до провалу проекту або додаткових витрат, тому перед проектуванням необхідно переглянути інші подібні роботи для закріплення знань та формування початкового уявлення про роботу.

Для успішного запровадження інтернету речей зіграли роль певні умови:

– динамічний попит на ресурси;

– потреби у реальному часу;

– експоненціальне зростання попиту;

– доступність додатків;

– захист даних та конфіденційність для користувачів;

– ефективне споживання енергії;

– виконання програм поблизу кінцевих користувачів;

– доступ до відкритої та взаємодіючої хмарної системи.

Також існує твердження, що існують ще три компоненти, які необхідні для створення та безперебійного обчислення даних інтернету речей:

– апаратне забезпечення – сенсори, виконавчі механізми, IP-камери, відеоспостереження та вбудоване комунікаційне апаратне забезпечення(центральні пристрої);

– проміжне програмне забезпечення – зберігання та інструменти для обчислень для аналізу даних за допомогою хмари(Cloud) та аналітики великих даних(Big Data);

– презентація – візуалізація, яка повинна бути простою для розуміння та інструменти для інтерпретації, які можуть бути розроблені для різних напрямків застосування.

1.3 Історія розвитку

Інтернет речей – це технологічна революція, яка представляє майбутнє обчислень і комунікацій, і його розвиток залежить від динамічних технічних інновацій у ряді важливих галузей, від бездротових датчиків до нанотехнологій Першим Інтернет-пристроєм прийнято вважати машину для напоїв Coca-Cola в Університеті Карнегі-Мелон на початку 1980-х років.

Її розробники, які працювали над торговим автоматом, написали серверну програму, яка відстежувала, скільки часу минуло з моменту, коли стовпець зберігання в автоматі не був заповнений. Розробники могли підключитися до машини через Інтернет, перевірити стан машини та визначити, чи буде на них чекати холодний напій, якщо вони вирішать спуститися до машини.

У 1990 році Джону Ромкі вперше вдалось підключити тостер до мережі Інтернет за протоколом TCP/IP. Вже через рік науковці з Кембриджського університету приступили до розробки першого прототипу веб-камери для контролю кількості кави в кав'ярнику комп'ютерної лабораторії. Ідеєю програми було знімання кавника тричі за хвилину, пересилаючі ці знимки на локальні комп'ютери.

Згодом, в 1999 році, в Массачусуцькому технологічному інституті, виконачий директор Auto-ID Labs Кевін Ештон придумав термін «Інтернет речей». Того ж року він виступив з презинтацією, у якій описав Інтернет речей як технологію, яка з'єднує декілька пристроїв за допомогою міток RFID для управління ланцюгом передачі повідомлень.

Інтернет речей з того часу набув популярності в спільноті RFID(Radio Frequency Identification) користувачів, які посилалися на можливість виявлення інформації про позначений об'єкт шляхом перегляду інтернет-адреси або запису бази даних, що відповідає певній мітці або комунікації ближнього поля.

З того часу проводилося багато дослідницьких робіт, які твердили що ключові технології IoT включають RFID, сенсорну технологію, нанотехнологію та технологію вбудованого інтелекту. Серед них RFID є основою та мережевим ядром побудови Інтернету речей. IoT дозволив користувачам перенести фізичні об'єкти в сферу кіберсвіту [3].

Це стало можливим завдяки різноманітним технологіям маркування, таким як NFC, RFID і 2D штрих-кодам, які дозволяли ідентифікувати фізичні об’єкти та передавати їх через Інтернет. IoT, який інтегровано з сенсорною технологією та радіочастотною технологією, є повсюдною мережею, заснованою на розповсюджених апаратних ресурсах Інтернету, є об’єктами вмісту Інтернету. Це також нова хвиля IT-індустрії, оскільки почали застосовувати обчислювальні поля, комунікаційну мережу та технологію глобального роумінгу.

Наприкінці 2000-х – початку 2010-х років корпорації в усьому світі почали дуже захоплюватися Інтернетом. Приблизно в цей період IBM почала роботу над компанією Smarter Planet. У 2011 році Cisco заявила, що IoT «народився» між 2008 та 2009 роками – моментом, коли до Інтернету було підключено більше речей або об’єктів, ніж живих людей на планеті. У тому ж році Gartner вперше додала нове явище до свого знаменитого списку Hype-cycle for Emerging Technologies. На даний момент у всьому світі нараховується понад 27 мільярдів пристроїв Інтернету речей, і з кожним роком це число тільки збільшується.

1.4 Архітектура Інтернету речей

На додаток до складних технологій комп’ютерної та комунікаційної мережі за межами, Інтернет речей, як і раніше, включає багато нових допоміжних технологій, таких як збір інформаційних технологій, технології віддаленого зв’язку, технології віддаленої передачі інформації, технології аналізу та контролю інформації морських заходів тощо.

На даний момент загальновизнаної архітектури не існує, але найпоширеніші версії – це поділ IoT на 3 та 5 рівнів. Трьох-рівнева архітектура складається з рівня сприйняття, мережевого рівня та рівня прикладного програмного забезпечення. П’яти-рівнева архітектура складається з бізнес рівня, прикладного програмного забезпечення, проміжного програмного забезпечення,

мережі та сприйняття, і кожний з них так чи інакше відповідальний за дані, які мандрують мережею [4].

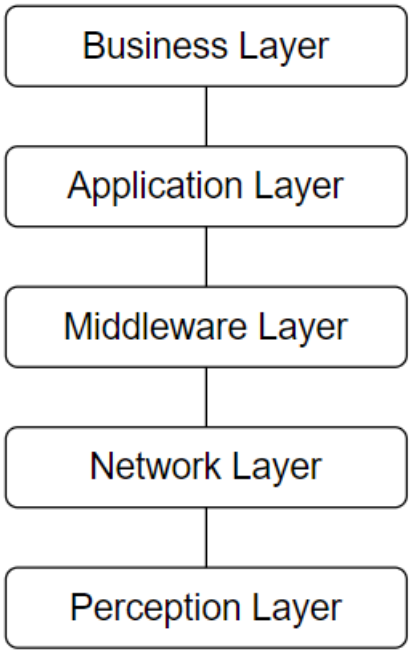

Рисунок 1.2 – П’яти-рівнева архітектура IoT

Рівень сприйняття(Perception Layer) складається з фізичних об'єктів, які контролюються та керуються пристроями датчиків і виконавчих механізмів, має в якості основної мети збір даних датчиків і спрацювання команди.

Далі, за допомогою мережевого рівня(Network Layer), який може характеризуватися декількома мережевими протоколами, дані датчиків передаються на рівень проміжного програмного забезпечення(Middleware Layer), в якому відбувається обробка отриманої інформації.

Тут інформацію можна не лише обробляти, використовуючи передові методології аналізу даних, а й зберігати отриману інформацію Головна мета цього рівня – досягти автономного процесу прийняття рішень, який буде обробляти та відправити команди для виконання назад фізичним об’єктам у порядку виконання дій, які впливатимуть на загальні умови фізичного середовище.

Зібрана та проаналізована інформація може бути представлена кінцевому користувачеві за допомогою прикладного рівня(Application Layer), який також може використовувати його для управління загальною системою. Останній, бізнес-рівень(Busines Layer) дозволяє системним адміністраторам керувати системою та контролювати загальну функціональність платформи Інтернету речей.

1.5 Використання хмарних сервісів у системах IoT

IT Cloud (IT-хмара) - набір з певної кількості IT-ресурсів, розміщених в інфраструктурі хмарного провайдера. Сама хмара може складатися з мережевих пристроїв, серверів, дискових накопичувачів, каналів зв'язку, різноманітного програмного забезпечення тощо. Основне завдання хмари – надання користувачеві доступу до інформації, яка знаходиться в хмарному сховищі, а також надання віртуальних апаратних потужностей.

Хмара може бути приватною, громадською або гібридною. Основна перевага застосування хмари полягає в можливості систематизувати дані, забезпечити їм надійний захист, збереження (завдяки автоматичному резервному копіюванню), а також швидкий доступ і масштабування IT-інфраструктури.

Оскільки пристрої IoT генерують великі обсяги даних і включають численні обчислювальні додатки (наприклад процеси обробки в реальному часі та аналітика), інтеграція з інфраструктурою хмарних обчислень може значно заощадити.

Основна концепція такого підходу – відмова від стаціонарних обчислювальних потужностей та інтеграція даних у віддалене сховище. Це дозволяє заощаджувати на додаткових апаратних засобах, а оскільки мережа Інтернету покриває майже увесь світ, є можливість безперервного контролю за даними.

Рисунок 1.3 – Система IoT з використанням хмари

Як чудовий приклад можна роздивитися наступний випадок: у малому та середньому підприємстві, яке виробляє пристрій керування, який використовується в розумних будинках і будівлях, амбіції щодо розширення можуть бути неприємними та дорогими. А з використанням хмарних технологій ім не треба задумуватися про додавання інших модулів, на розробку яких може піти багато часу. Крім того, хмарна інтеграція дозволяє підприємствам зберігати та обробляти величезні набори даних, зібраних з кількох джерел.

У всьому світі починають створювати цілі розумні міста, і використання в них хмарних технологій має певну вигоду. Очікується, що функціями розумного міста стануть інтелектуальні програми керування живленням, розумне керування подачею води, розумне керування транспортом, міська мобільність та інші. Крім того, вони будуть виробляти дуже великі обсяги даних. Маючи інтеграцію з хмарою, розумне місто тепер може обробляти ці записи та

програми [5].

Крім того, хмара може допомогти пришвидшити розширення будь-яких пакетів IoT та їх розгортання, які раніше викликали значні занепокоєння щодо забезпечення необхідними комп'ютерними ресурсами. Постачальник загальнодоступних хмарних обчислень може розширити екосистему Інтернету речей за запитом, надавши стороннім доступ до своєї інфраструктури, дозволяючи їм поєднувати дані Інтернету речей і комп'ютерні ресурси, що працюють на пристроях IoT.

Також компанія може надавати дані IoT для доступу та обслуговування. Це свідчить про застосовність і бажання модифікувати інфраструктуру IoT і хмарні обчислення. Але через різність архутектур хмари та IoT, їх інтеграція завжди була проблемою. Пристрої Інтернету речей, як правило, обмежені на регіональному рівні, з обмеженою підтримкою, дорогими оцінками (залежно від вартості оновлення/доставки) і часто нестабільними (відповідно до ресурсів і доступу).

З іншого боку, ресурси хмарних обчислень зазвичай розміщуються в розумному та ефективному місці, що забезпечує швидкість та гнучкість. Датчики та пристрої встановлюються перед інтеграцією даних та їхніх пропозицій у хмару, що дозволяє їм розподіляти дані між будь-якими хмарними ресурсами та зменшувати невідповідності.

Крім того, реалізація послуг і алгоритми отримання інформації з датчиків розміщуються в хмарі, щоб послуги та датчики були доступні в режимі реального часу. Інтеграція Інтернету речей та хмари може передавати сенсорну інформацію в хмару.

Ця широко поширена інфраструктура була однією з найперших інновацій (широко використовувалася для виявлення радіації та радіаційних карт під час землетрусів в Японії). На цей час існують десятки добре відомих хмар, включаючи ThingsWorx, ThingSpeak, CloudSensor і хмарні служби реального часу.

Споживачі, які бажають зберігати дані своїх IoT систем у хмарі можуть

платити постачальникам публічних хмар за послуги, якщо це необхідно. Більшість постачальників надають своїм клієнтам передові інструменти розробки, щоб вони мали змогу самі налаштувати свою архітектуру [6]. Крім того, інфраструктура хмарних обчислень та Інтернету речей, а також пов'язані з ними послуги можуть бути позначені таким чином:

Інфраструктура як послуга (IaaS) – хмари дозволяють користувачам підключатися до датчиків і виконавчих механізмів у хмарі. IaaS — це значне обчислення, яке зберігає та передає рішення за запитом. Вона надає послугу хмарних обчислень, відповідно до потреб замовника. IaaS пропонує управління IoT для керування об'єктами як обов'язкову умову надання відповідних послуг.

Платформа як послуга (PaaS) – високопродуктивна модель хмарної публічної інфраструктури для IoT та хмарних сервісів. PaaS — це повноцінна хмарна архітектура для розробки та розгортання власної платформи, що включає можливості надання послуг, які варіюються від найпростіших хмарних послуг до складних хмарних рішень.

Програмне забезпечення як послуга (SaaS) – це продукти, які дозволяють користувачам отримувати повні пакети програмного забезпечення, спеціально засновані на хмарі та призначені для Інтернету речей. Пакети SaaS подібні до стандартних хмарних пакетів із використанням сенсорів і пристроїв IoT.

Твердження полягає в тому, що пакети IoT SaaS зазвичай розробляються на основі інфраструктури PaaS і дозволяють використання бізнес-моделей, які залежать від програмного забезпечення та послуг IoT. Це дає широкий спектр розуміння взаємодій Інтернету речей і хмар а токож чому вони такі важливі та вигідні.

Наразі все більше датчиків та виконавчих механізмів Інтернету речей базуються на хмарі, що дозволяє користувачам отримувати вигоду від загальної продуктивності, досвіду бізнесу та наданих ними функцій оплати. IoT-хмари забезпечуючи сумісність даних пристроїв Інтернету речай і їх внесків у хмару,

оптимізовані алгоритми та рішення, тому вони забезпечують високі записи для аналітичних цілей у регіонах, що включають розумну енергетику, розумний транспорт, розумні міста та комунікації. Більше того, компоненти IoT з портативними комп'ютерами на цій основі можуть отримати ще більшу вигоду від інтеграції з хмарою.

IoT – це механізм для підключення комп'ютерних пристроїв, сенсорів та виконавчих механізмів, віртуальних об'єктів, яким було надано вказівки та можливість змінювати записи в мережі без необхідності контакту з людьми чи комп'ютером.

IoT охоплює все, що створюється людьми або пристроями, та можуть бути визначені IP-адресами й мають можливість надсилати дані через мережу. З розвитком інформаційних технологій Інтернет речей значно розширився і продовжує зростати. Керуючі пристрої IoT забезпечують зв'язок між датчиками, а мільярди підключених пристроїв, ймовірно, стануть частиною людського життя в майбутньому, а отже необхідно подбати про стабільне зберігання та обробку великих обсягів даних.

Багато підприємств по всьому місту, включаючи сільськогосподарський сектор, охорону здоров'я, енергетику, транспорт та управління будівлями, вже майже повністю захоплені Інтернетом речей.

Експерти та передові розробники намагаються визначити інші способи підключення до Інтернету речей через хмарну мережу. Для цього розробляються нові додатки IoT, які стають надзвичайним підходом до майбутнього розвитку. Люди більше перемагають не від збільшення підключеності пристроїв, а від соціально значущих пристроїв, дані яких вони збирають із мережі IoT. Машини можуть надавати корисну інформацію про продуктивність і зовнішній вигляд у польових умовах для зв'язку, що забезпечується хмарними рішеннями.

Пов'язані пристрої не обмежуються сертифікацією пристроїв компаній, але

вони також можуть відійти від персональних пристроїв для всіх за допомогою мережевих хмарних рішень. Через закрите сховище, потужність обробки, енергію та інші чинники, швидкий час з'єднання контролюється за допомогою реальності. Завдяки широкому розмаїттю датчиків і кількості даних, які вони створюють, збір, зберігання, обробка та ефективність IoT залишаються складними.

IoT також зв'язує пристрої та людей, виробляючи величезні обсяги даних. Через складність систем, угоди про підключення та відповідність застарілих програм, отримання доступу до інформації через підприємства може бути складним завданням. Інфраструктура Інтернету речей (наприклад, датчики, хмара та RFID) є або унікальною, або розповсюдженою, а ресурси для створення та передачі доступу обмежені та зазвичай дорогі.

Пристрої Інтернету речей часто страждають від дефіциту ресурсів обробки та зберігання, а також обмеженого бюджету через їх продуктивність. Хмарні обчислення забезпечують безмежне зберігання та є енергоефективним рішенням. Хмара з'єднує користувачів з інформацією та ресурсами через інтернет-посилання.

Хмарна інфраструктура – це автономний або повсюдно поширений регіон (ресурси, доступ до яких можна отримати з будь-якого місця), що забезпечує простий доступ до недорогих ресурсів. Віртуалізація в хмарі є результатом автономних зусиль ресурсної бази.

Хмарне з'єднання Інтернету речей – це спосіб отримати доступ до ресурсів і мати постійний доступ до них під час виконання хмарних обчислень. Попит на системи IoT для хмарного стиснення інформації, а також доступність та продуктивность сприяє інтеграції Інтернету речей і технології хмарних обчислень з кожним днем. Таке з'єднання дозволяє зберігати та обробляти накопичені факти, ідентичні дані в різних додатках, інтегрувати точки з кількох пристроїв і користувачів, які можуть бути розташовані у різних місцях, а також їх

мобільність.

Здатність передавати дані в хмару завдяки такому поєднанню потужності є надзвичайною з точки зору роботи системи, керування, відстежування та контролю розподілу даних. Для пакетів IoT хмара може використовувати надійне обладнання для відновлення у разі аварії або якоїсь помилки, обробки інформації, а також її пошуку. Великі системи IoT за своєю суттю можуть бути небезпечними. Вони містять різноманітний набір дифузних датчиків, які створюють дані, і на які потрібно постійно звертатися.

Хмара пропонує негабаритний вихідний сервіс і може використовувати технології IoT для створення складної системи без ускладнень. Крім того, багато послуг Інтернету речей можуть отримати вигоду від системи доставки повідомлень та даних, яка зосереджена на створенні та доставці систем IoT і в основному заснована на хмарній інфраструктурі.

Платформа для IoT в основному базується на хмарі з можливістю проектувати, розгортати, запускати та керувати мережами хмарних підключених пристроїв. Це показує початкові можливості хмарної платформи та архітектури IoT, на додаток до їх взаємодії з трьома модифікаціями хмарних обчислень (тобто Iaas, PaaS і SaaS).

Усі пристрої Інтернету речей будуть підключатися до єдиного хмарного пулу повсюдних ресурсів. Пристрої можуть отримати доступ без труднощів, накопичувати, систематизувати, візуалізувати, архівувати, пропорціонувати та шукати повнорозмірні обсяги готової інформації з багатьох програм, які використовують цю платформу, а найважливіше, без участі людини.

Для створення, аналізу та збереження великого обсягу інформації з датчиків можуть використовуватись обчислювальні ресурси та ресурси зберігання хмари. Крім того, хмарна платформа IoT надає клієнтам і програмам доступ до даних датчиків у гнучких ситуаціях використання, що дозволяє сенсорним пристроям виконувати спеціалізовані обов’язки обробки даних.

Такі платформи являють собою довготривале рішення для хмарних обчислень для сенсорного керування, яке включає сенсорні пристрої в якості постачання для клієнтів. Вони забезпечують відстежування та контроль за датчиками і маніпулює пропозиціями для клієнтів через веб-браузер. Крім того, хмарне рішення спрощує зберігання даних в усіх компонентах збору та обробки інформації IoT, беручи до уваги просте налаштування та взаємодію усіх проблем, зберігаючи при цьому низьку вартість на розгортання платформи та складну обробку даних.

За допомогою хмарної інфраструктури такої платформи клієнти можуть розгортати та користуватися будь-яким додатоком на хмарному обладнанні. Платформа спрощує процесс вдосконалення додатків, усуває потребу в інфраструктурі, робить проблеми менш складними для їх вирішення та знижує витрати на відновлення. Вона надає клієнтам унікальні можливості керування інструментами, прямий зв’язок з пристроями, зберігання для накопичених даних.

Хмарні обчислення та послуги можуть використовуватися для збереження, обробки та аналізу значної кількості даних різноманітних пристроєв. Пакет для розробників базується на складних і швидких хмарних пристроях для розробки додатків IoT. Ця технологія охоплює програмні інтерфейси прикладного програмного забезпечення з відкритим та доступним сервісом, надаючи розробникам можливості високого ступеня вдосконалення та розгортання [7].

Управління підписками, координація мережі, підключення суб’єктів, виявлення питань, аналіз статистичних даних і структура речей – усе це частина системи, пакета хмарних пропозицій, які допомагають у розгортанні та спеціалізації послуг обробки.

Сенсори та виконавчі механізми зазвичай групуються в мережі Інтернету речей у хмарних структурах IoT, включаючи домашнє співтовариство. Ці мережі

підключаються до хмари через спеціальний шлюз, підключений до мережі Інтернет, як правило, це домашній маршрутизатор. Роутер пересилає зібрану інформацію з мереж у хмару. Хмара безперервно отримує, обробляє та зберігає дані і гарантує, що вони завжди під рукою.

Постачальник хмарного сервісу може надавати певні послуги з дозволом на доступ до інформації та керування нею за допомогою спеціальних інструментів хмарної обробки. У цьому випадку хмара служить проміжним шаром між проблемами та додатками IoT, приховуючи від клієнта всі складності та функціональні можливості реалізації. Така платформа може значно покращити програмне забезпечення, оскільки збір даних і передача записів у цьому шарі будуть дуже гарним рішенням. Дизайн повністю заснованої на хмарі платформи IoT спрямований на максимальне надання даних і послуг.

На рисунку 1.4 зображена хмарна платформа IoT, підключена до кінцевих пристроїв через шлюзи. Хмарні програми можна зберігати та візуалізувати, щоб клієнт міг також отримувати доступ, контролювати й керувати звідусіль і в будь-який час за допомогою веб-браузера чи програми. Хмара містить у собі усі необхідні модулі та компоненти для роботи з даними, наприклад велике сховище, розмір якого може бути змінений за потреби, програмні та обчислювальні потужності для обробки даних, модулі для підключення та спілкування з пристроями.

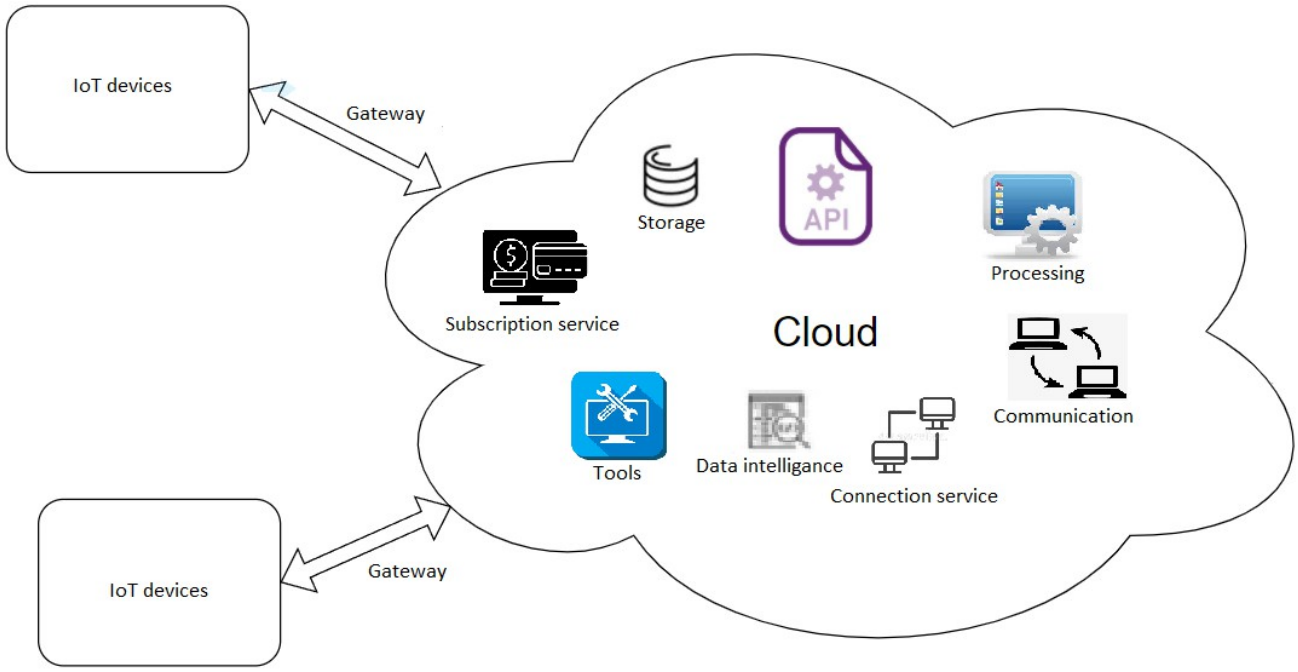

Рис 1.4 – Структура IoT системи, основаної на використанні хмари

Основні переваги Iot у Cloud:

Масштабованість – у разі розробки складних мережевих інфраструктур розширення масштабу вимагає придбання додаткового обладнання, витрати більше часу та зусиль для налаштування системи, щоб забезпечити його належну роботу. У хмарній системі Інтернету речей замовнику не потрібно про все це турбуватися, а додавання нових ресурсів зазвичай зводиться до оренди іншого віртуального сервера або ще більшого хмарного простору, які зазвичай мають додаткову перевагу в тому, що вони швидко реалізуються. Крім того, послуги хмарної платформи IoT пропонують більшу гнучкість, якщо замовник хочете збільнити або обмежити свої вимоги до сховища, або зменшити кількість пристроїв з підтримкою IoT, усе це робиться доволі легко та швидко.

Мобільність даних – якщо дані зберігаються та обробляються на хмарному сервері, не треба підключатися до певної мережі, до якої підключені датчики, до них можна отримати доступ майже з будь-якої точки світу, а це також означає, що вони не будуть обмежені будь-якими інфраструктурними або мережевими

границями. Мобільність даних є важливим фактором, особливо коли йдеться про проекти IoT, які передбачають відстеження та керування підключеними пристроями у реальному часі. На відміну від зберігання даних на локальних серверах, хмарний сервіс для Інтернету речей надає користувачеві інструменти для керування та оновлення пристроїв і датчиків, а також обробки отриманих даних віддалено та в реальному часі.

Економія часу та сил – з хмарними рішеннями IoT зазвичай потрібно менше часу та зусиль для їх реалізації а також значно знижує загальну вартість робіт, але це досягається за рахунок налаштування платформи. Хоча це і правда, що системи Інтернету речей, встановлені локально(тобто у тій же мережі), можуть бути більш пристосовані до цілей проекту, вони також передбачають тривале розгортання можливостей управління даними та аналізу та оновлення існуючої структури мережі компанії через збільшення трафіку даних. Загалом, хмарна інфраструктура IoT виявляється більш прибутковою, коли час виходу на ринок є вирішальним фактором бізнесу.

Безпека – проблеми безпеки, які турбували світ IoT з моменту його заснування, можуть бути складними. В інтеграції хмарної платформи та локальної інфраструктури Інтернету речей все залежить від відповідальності її проектувальників. Що стосується локальних серверів, то відповідальність лежить на плечах компанії, і це залежить лише від практик безпеки в організації. Тому цілком зрозуміло, що деякі організації можуть відчувати себе незручно відмовляючись від контролю над своїми конфіденційними даними та передавати їх сторонній стороні. Тим не менш, як постачальники послуг, так і клієнти погоджуються з тим, що зберігання та обробка даних Інтернету речей у хмарі є більш безпечним, ніж збереження їх на локальних серверах. Завдяки можливості регулярних оновлень програмного забезпечення, а також цілодобового моніторингу, які пропонують деякі постачальники хмарниих сервісів, можна уникнути серйозних порушень безпеки.

Ціна – великі початкові інвестиції та підвищений ризик впровадження у випадку локальної системи Інтернету речей можуть стати негативним чинником. Крім того, існує проблема постійних витрат на технічне обслуговування обладнання та IT-персонал. З точки зору хмари все виглядає значно краще. Знижені початкові витрати та гнучка схема ціноутворення на основі фактичного використання спонукають підприємства на основі IoT перейти на хмарне розташування. У цій бізнес-моделі витрати легше передбачити, і компаніям не доведеться турбуватися про збій обладнання та втрату даних, який у разі внутрішніх систем Інтернету речей може призвести до величезних додаткових витрат, не кажучи вже про збитки бізнесу через несправність системи.

1.6 Cloud Computing в IoT

На іншому кінці хмарного рішення Інтернету речей є Cloud Computing, або хмарні обчислення. Воно забезпечує програмне забезпечення як послуга (SaaS), Платформа як послуга (PaaS) та Інфраструктура як послуга (IaaS). Інтернет речей радикально змінює спосіб роботи бізнесу та взаємодію людей із фізичним світом. Хоча речі, Інтернет та підключення є трьома основними компонентами IoT, цінність полягає в тому, щоб закрити розрив між фізичним та цифровим світами у системах, що самозміцнюються та самовдосконалюються. Поєднання хмарних обчислень і Інтернету речей також відоме як Cloud IoT.

Вдихаючи нове життя в IT-сервіси, хмарні обчислення – це найновіші технології, які перемістили споживчі та бізнес-додатки до Інтернету, таким чином дозволяючи підприємствам оптимізувати свою IT-продуктивність та зменшити витрати, які в іншому випадку були б більш великими через потребу створення та підтримки IT-архітектури сайту для зберігання даних і запуску програм.

Хмарні рішення не тільки більш вигідні в довгостроковій перспективі, вони також забезпечують кращу безпеку, мобільність корпоративних даних,

розширення співпраці з колегами та клієнтами, більш розвинені рішення для аварійного відновлення, і це лише деякі переваги. Більше того, хмарні обчислення пропонують більшу гнучкість, що допомагає змістити фокус компанії з питань, пов’язаних з хостингом, на аспекти, які безпосередньо впливають на прибуток її бізнесу.

У наші дні, коли хмарні обчислення в популярності, вони поступово стають незамінним компонентом практично будь-якої бізнес-організації, незалежно від її масштабу. Але до того, як це здійснило революцію в галузі, у компаній, які прагнули перейти на цифрові технології, не було іншого вибору, окрім як створити власну внутрішню IT-інфраструктуру з усіма серверами, апаратним та програмним забезпеченням, які необхідно було налаштувати, підтримувати, оновлювати та захищати. Для будь-якої компанії все це означало, перш за все, величезні інвестиції та великий ризик.

Сьогодні, у випадку багатьох підприємств, що керуються Інтернетом речей, вибір є очевидним: створити дорогу, вразливу та важкомасштабовану серверну інфраструктуру на місці або обрати хмарне рішення IoT, яке дозволяє уникнути непотрібних витрат і не представляти основні компроміси щодо функціональності платформи. У ряді випадків локальна платформа Інтернету речей може генерувати витрати, які підштовхують весь проект до межі прибутковості. З розгортанням хмари все стає зовсім іншим.

Граничні обчислення або практика обробки даних біля краю мережі(Edge Computing), де дані генеруються, зазвичай використовуються в рішеннях на основі Cloud IoT, щоб скоротити час відповіді та прискорити обробку даних. У розгортанні Інтернету речей часто використовується комбінація хмарних і граничних обчислень, щоб отримати найкраще з обох світів. Центри обробки даних Edge в Індії прискорюють обробку даних в режимі реального часу, що дозволяє особам, які приймають рішення, діяти швидше. Але підхід лише на периферії не дає цілісної картини бізнес-операцій. За відсутності хмарного рішення фабрика може відстежувати кожне обладнання окремо, але не зможе оцінити, як ці пристрої працюють по відношенню один до одного. Лише

оптимальне поєднання хмари та периферії може допомогти компаніям отримати максимальну користь від своїх ініціатив IoT.

Обробка даних на периферії мережі або периферійні обчислення використовується з рішеннями IoT і забезпечує швидшу обробку та час відповіді. Щоб краще зрозуміти, як це працює, розглянемо велику фабрику з багатьма впровадженими датчиками IoT. У цій ситуації має сенс, перш ніж надсилати дані в хмару для обробки, об'єднати їх поблизу кордону, щоб запобігти перевантаженню хмари за рахунок зменшення прямих з'єднань.

Дата-центри з таким підходом роблять обробку даних набагато швидше. Проте підхід, який базується лише на периферії, ніколи не дасть повного уявлення про бізнес-операції. Якщо хмарного рішення немає, фабрика контролює лише кожен блок окремо. Крім того, вона не може уявити, як ці одиниці працюють по відношенню один до одного. Ось чому лише поєднання периферії та хмари дозволить підприємствам отримати вигоду від розробок IoT.

Роль хмарних обчислень в Інтернеті речей полягає у підвищенні ефективності щоденних завдань у поєднанні з Інтернетом речей. Хмарні обчислення – це забезпечення шляху для даних до місця призначення, в той час як Інтернет речей генерує величезну кількість даних. Таким чином, роль хмарних обчислень в IoT полягає в тому, щоб працювати разом для зберігання даних IoT, забезпечуючи легкий доступ, коли це необхідно. Важливо зазначити, що хмарні обчислення — це простий спосіб переміщення великих пакетів даних через Інтернет, створених Інтернетом речей.

Основні переваги використання хмарних обчислень в IoT:

Хмарні обчислення Інтернету речей надають багато варіантів підключення, що передбачає великий доступ до мережі. Люди використовують широкий спектр пристроїв, щоб отримати доступ до ресурсів хмарних обчислень: мобільні пристрої, планшети, ноутбуки. Це зручно для користувачів, але створює проблему необхідності точок доступу до мережі.

Розробники можуть використовувати хмарні обчислення Інтернету речей на вимогу. Іншими словами, це веб-сервіс, доступ до якого здійснюється без

спеціального дозволу чи будь-якої допомоги. Єдина умова – доступ до Інтернету.

На основі запиту користувачі можуть масштабувати сервіс відповідно до своїх потреб. Швидкість і гнучкість означає, що ви можете розширювати простір для зберігання, редагувати налаштування програмного забезпечення та працювати з кількістю користувачів. Завдяки цій характеристикі можна забезпечити глибокі обчислювальні потужності та зберігання.

Хмарні обчислення мають на увазі об’єднання ресурсів. Це впливає на посилення співпраці та налагоджує тісні зв’язки між користувачами.

Із зростанням кількості використовуваних пристроїв IoT і автоматизації виникають проблеми з безпекою. Хмарні рішення надають компаніям надійні протоколи аутентифікації та шифрування.

Хмарні обчислення Інтернету речей зручні, оскільки користувач отримуєте від послуги рівно стільки, скільки платитить. Це означає, що вартість залежить від використання – постачальник вимірює статистику використання. Для підключення до Інтернету та обміну даними між компонентами мережі потрібна зростаюча мережа об’єктів з IP-адресами.

1.7 Fog Computing в IoT

Переважаючими хмарними сервісами, які пропонує Cloud Computing, є інфраструктура як послуга (IaaS), платформа як послуга (PaaS) і програмне забезпечення як послуга (SaaS). Усі ці хмарні сервіси рухаються в напрямку «Все як послуга» (XaaS). Тим не менш, інформація, отримана з цих мільйонів датчиків, визначена як великі дані, не може бути повністю оброблена та перенесена в хмару, оскільки це може спричинити затримку.

Крім того, лише деякі програми Інтернету речей потребують швидкої обробки, ніж наявні можливості Cloud Computing. Цю проблему можна вирішити

за допомогою Fog Computing, який об’єднує потужність обробки інтелектуальних пристроїв, розташовану поблизу клієнта, щоб допомогти з використанням мережі, обробки та зберігання поблизу межі. Функціональність туману з IoT полягає в тому, щоб скоротити передачу інформації в Cloud Computing для зберігання, аналізу, обробки, ефективності та покращення продуктивності.

Таким чином, інформація, зібрана сенсорними пристроями, передається граничним мережевим пристроям, для тимчасового зберігання та обробки, а не передає їх у хмару, таким чином зменшуючи затримку та мережевий трафік. Об’єднання IoT з Fog Computing створює унікальну перспективу для послуг, які називаються Fog as a Service (FaaS), де кілька вузлів туману створюються постачальником послуг у різних географічних місцях і працюють як власник для різних мешканців з різних вертикальних місць. Кожен вузол у тумані керує можливостями зберігання, обчислень і мереж [8].

Fog – це повністю розподілений обчислювальний підхід, він не повністю залежить від будь-якого інтегрованого компонента, такого як Cloud Computing. Проблему затримки CC можна подолати за допомогою туману, використовуючи невикористані ресурси різних пристроїв поблизу клієнта. Тим не менш, від Cloud Computing залежить виконання основних завдань.

На відміну від Cloud Computing, Fog – це розподілений обчислювальний підхід, коли різні пристрої поблизу клієнтів використовують обчислювальні можливості, які мають менше функцій, але хорошу обчислювальну потужність з кількома ядрами. Таким чином, кілька розумних пристроїв, таких як керування мережевими пристроями, комутатори, базові станції, маршрутизатори, смартфони та інші, встановлені з накопичувачем і обчислювальною потужністю, які можуть працювати як туманні обчислювальні пристрої.

Через різноманітну організацію та глобальну взаємодію розвиваються різні дослідницькі проблеми, пов’язані з туманними обчисленнями.

Середовище розгортання Fog Computing та його вимоги є ключовими питаннями в принципі туманних обчислень. Це причина того, чому обчислювальні схеми, які присутні в домені Fog Computing, різноманітні.

Численні постачальники послуг, а саме Google, IBM, Amazon, Microsoft тощо, користуються послугами хмари. З еволюцією хмарних обчислень технологія перетворилася на нове покоління з різними хмарними сервісами, такими як інфраструктура як послуга (IaaS), платформа як послуга (PaaS), програмне забезпечення як послуга (SaaS) тощо. водночас пов'язані з підприємством, програмним забезпеченням та освітою.

MCC, або мобільні хмарні обчислення – це ще одна парадигма обчислень, де споживачі хочуть запускати всі свої програми на своїх мобільних пристроях, крім комп’ютерів старого типу. MCC орієнтований на час виконання програми, зберігання, енергетичні та ресурсні бар’єри. Однак, коли виконуються критичні програми, краще запускати програми за межами портативних пристроїв. У таких сценаріях MCC розробляє різноманітні обчислення, щоб допомогти таким сценаріям розвантажити мобільні додатки та виконати їх поблизу кінцевих споживачів.

MCC містить дзеркальні сервери, які мають невелику вагу, які називаються хмарами. Вони розташовані на краю мережі. Трирівнева архітектура формується з мобільних пристроїв, хмарних серверів і хмарних програм, яка розробляє ієрархічний додаток для підвищення якості досвіду (QoE) мобільних користувачів і надає багато бізнес-можливостей для постачальників хмарних послуг і мережевих операторів.

Подібно до MEC і MCC, Fog computing також виконує обчислення на краю мережі та на стороні основного компонента. Головними компонентами є основні маршрутизатори, комутатори wan, регіональні сервери тощо, які використовуються в інфраструктурі для обчислень у середовищі Fog. У результаті велика кількість пристроїв IoT можна легко інтегрувати з Fog. Крім того,

компоненти туманних обчислень на межі мережі можна розташувати ближче до датчиків або пристроїв IoT порівняно з мобільними периферійними серверами та хмарами.

Оскільки датчики або пристрої IoT щільно розкидані і потребують негайної відповіді на запити послуг, цей метод дозволяє обчислювати та зберігати інформацію IoT в навколишньому районі датчиків або пристроїв IoT. Як результат, час очікування передачі послуг для запитів IoT в режимі реального часу може бути надмірно скорочений.

На відміну від Edge Computing, послуги Cloud Computing, такі як SaaS, IaaS, PaaS тощо, можуть бути розширені за допомогою Fog Computing до границі мережі. Через ці заявлені особливості туманні обчислення вважаються добре організованими та більш перспективними для IoT на відміну від інших пов’язаних обчислювальних парадигм.

Туманні обчислення визначаються як надзвичайно віртуалізоване середовище, яке забезпечує мережу, зберігання та обчислювальні ресурси між застарілими інформаційними центрами Cloud Computing, зазвичай, але не повністю розташованими на межі мережі. Структура туману містить різні граничні вузли з невеликою кількістю компетенцій обробки, які часто називають вузлами туману. Ці вузли туману мають менше потужностей для обробки та зберігання. У туманній мережі іноді граничні сервери називають хмарами , які беруть участь у спільному обчислювальному оточенні, а не за межами мережі. Використовуючи ці пристрої туману, клієнти можуть отримати відповідь у режимі реального часу для чутливих додатків із затримкою. Незважаючи на те, що ця фраза спочатку була розроблена Cisco, різні дослідники та галузі визначили туманні обчислення з багатьох різних точок зору.

Туманні обчислення — це розширення Cloud Computing, але ближче до речей, які працюють з інформацією IoT. Fog Computing виконує роль арбітра між кінцевими пристроями та Cloud Computing, що наближає служби зберігання,

мережеві послуги та обчислювальні послуги до граничних пристроїв. Це проілюстровано на рисунку 1.5. Ці кінцеві пристрої відомі як туманні вузли, і їх можна розташувати з підключенням до мережі в будь-якому місці. Пристрій, який має обчислення, обчислення, підключення до мережі та зберігання, можна назвати туманним вузлом. Такими туманними вузлами можуть бути комутатори, сервери, камери спостереження та маршрутизатори.

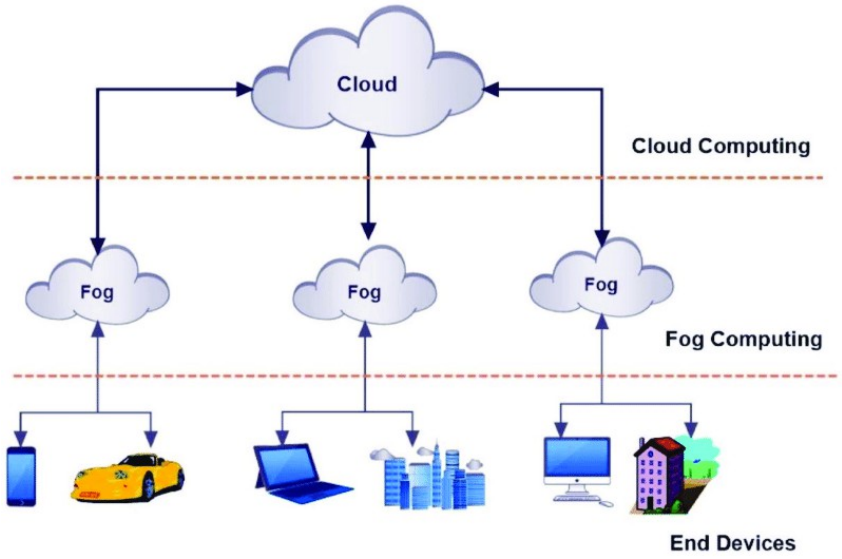

Рисунок 1.5 – Fog Computing, як проміжний рівень

Головні особливості Fog Computing:

Адаптивність – є великі мережеві датчики, які відстежують навколишнє середовище. Туман надає ресурси для зберігання даних і поширені обчислення, які можуть працювати з такими великими кінцевими пристроями.

Зв’язок у режимі реального часу – запити на обчислення туману забезпечують одночасний зв’язок серед туманних вузлів щодо пакетного аналізу, який використовується в хмарі.

Фізичний розподіл – на відміну від інтегрованої хмари, fog надає програми та послуги, які є децентралізованими та можуть розміщуватися в будь-якому

місці.

Менша затримка та усвідомлення позиції – туман знаходиться поблизу граничних пристроїв, він забезпечує менше часу очікування під час обчислення інформації граничних пристроїв. Крім того, це допомагає реагувати на положення, при якому туманні вузли можуть бути розміщені в різних місцях.

Сумісність – модулі Fog можуть адаптуватися та взаємодіяти з різними платформами через різноманітних постачальників послуг.

Положення для веб-аналітики та інтеграції з хмарою – туман розташований серед граничних пристроїв і хмари, щоб відігравати важливу роль у швидкості та обчисленні інформації поблизу граничних пристроїв.

Неоднорідність – крайові пристрої або туманні вузли розробляються різними компаніями і, таким чином, походять із різноманітними умовами та потребують розміщення відповідно до місця їх показу. Тому туман може адаптуватися на різних платформах.

Забезпечення гнучкості – однією з важливих особливостей запитів про туман є здатність безпосередньо підключатися до таких пристроїв, як мобільні телефони, і, отже, включати методи гнучкості, наприклад, протокол розділення ідентифікатора локатора (LISP), який вимагає розпорошеної індексованої системи.

Планування роботи – хоча туманні обчислення надають додаткові обчислювальні можливості на межі мережі, ключове запитання, яке вони підносять, полягає в тому, яким чином потрібно впоратися при виконанні роботи. У цій структурі вони мають намір визначити графік між роботами та серверами, що зменшує ймовірність зупинки роботи. Вони також досліджують питання планування роботи в мобільній мережі, створеній на шарі туману, де дозволено використання крихітних осередків з обчислювальними можливостями, які утворюють вузли туману.

Завантаження та вивантаження розподілу – багато алгоритмів планування

роботи було запропоновано стосовно систем туману. Незважаючи на те, що вони дозволяють розподіляти обчислювальні завдання на обчислювальні туманні вузли через різні шари, вони не обговорювали ймовірну дестабілізацію між вузлами туману щодо обсягу роботи.

Виділення ресурсів – основна особливість, яка була досліджена в обчислювальних системах туману, є співпраця між туманними вузлами та оцінка спільного використання ресурсів. У шарі туману ці функції були оброблені з метою виконання обчислювальних вимог. Ці осередки утворюють крихітні групи, кожна з яких позначає групу крихітних осередків, які обмінюються ресурсами для скидання навантаження на стільниковий гаджет від їх обсягу роботи.

Таким чином, ці підходи спрямовані на оптимізацію ЦП, пропускної здатності та загального сховища, щоб допомогти обчислювальним потужностям. Завдяки різноманітності цих ресурсів вони розбивають їх на різні інтервали ресурсів, щоб дозволити перерахувати їх у подібну одиницю.

1.8 Edge Computing в IoT

Обчислювальні можливості в Edge Computing надаються граничними серверами або кінцевими пристроями. Загальні граничні обчислення ніколи не мають імпульсивного зв’язку через будь-які види хмарних сервісів і зосереджуються на стороні пристроїв IoT. У певному дослідженні стверджується, що периферійні обчислення як обробка ресурсу або мережі, які знаходяться посередині серед центрів обробки даних хмари та джерел даних.

Будь-який інтелектуальний датчик або пристрої можуть мати джерела даних, проте граничні обчислення зазвичай різноманітні. Наприклад, невеликий центр обробки даних або хмара – це край хмарних обчислень і мобільних додатків, а шлюз IoT – це край серед хмарних обчислень і датчиків Інтернету речей.

Аналогічно, якщо на мобільному телефоні працює програма хмарних обчислень, то мобільний телефон виступає як край серед хмари та програми. Ключовим імпульсом обчислень у ЕС є те, що виконання має виконуватися в ближчому місці, звідки генерується інформація. З огляду на сприйняття периферійних обчислень, пристрої не тільки зберігають інформацію, але й генерують інформацію, беручи участь у переході. Ці пристрої Edge можуть виконувати роботу з центральної хмари, що додатково вимагає надання послуг. Зберігання інформації, розподіл вивантаження та виконання даних – все буде завершено граничним вузлом.

Пристрої Edge також зручні для визначення вимог і надання клієнтам послуг як представника хмарних обчислень. У таких ситуаціях пристрої периферії повинні бути належним чином розроблені, щоб відповідати вимогам конфіденційності, політики надійності та проблем безпеки. Туманні та граничні обчислення пропонують ідентичні функції з точки зору надсилання як розвідувальних даних, так і інформації в аналітичне середовище, розташоване поблизу, звідки генерується інформація, наприклад, датчики Інтернету речей, аудіо, відео тощо.

В основному хмарні центри обробки даних розташовані географічно централізовано, а також далеко від клієнтських пристроїв. Це призводить до затримок і затримок в режимі реального часу хмарними центрами обробки даних, а наслідки цього призводять до погіршення якості обслуговування, перевантаження мережі та затримок в обидва боки. Саме для вирішення таких проблем було запропоновано нову обчислювальну машину, а саме «Edge Computing» [9].

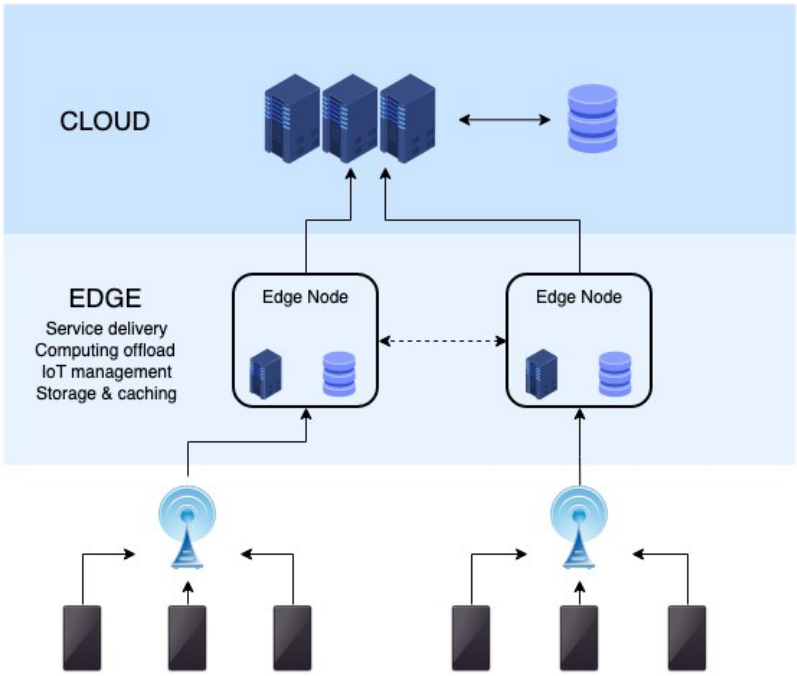

Рисунок 1.6 – Edge Computing

Основна ідея Edge Computing полягає у створенні гібридної архітектури, яка поєднує одноранговий і хмарні сервери з мобільними терміналами. Основною метою є розбіжність трафіку шляхом виконання обчислень ближче до краю мережі. Граничні обчислення підтримуються кінцевими пристроями (мобільні телефони, смарт-об'єкти), граничними пристроями (коммутаторами, мостами, точками доступу), периферійними серверами тощо. Усі ці компоненти мають необхідні можливості для підтримки обчислень на межі.

Однак парадигма Edge Computing зосереджена на периферії і не пов’язується з послугами Cloud Computing, такими як IaaS, PaaS, SaaS тощо. З концепцією Edge Computing та Cloud Computing були введені інші обчислювальні парадигми, як-от Mobile Edge Computing(MEC), мобільні хмарні обчислення (MCC). MEC зосереджується на 2- або 3-рівневому додатку в мережі разом з мобільними пристроями з сучасними базовими станціями стільникового зв’язку. Це підвищує ефективність мережі разом із розподілом вмісту та розробкою

програми.

Туманні та граничні обчислення значно схожі. Обидва турбуються щодо роботи з обчислювальними можливостями всередині обмеженої мережі, щоб виконувати обчислювальні роботи, які зазвичай виконувались у хмарі. І туман, і Edge Computing можуть допомогти галузям зменшити їхню залежність від хмарних платформ для перевірки своїх даних, що регулярно призводить до затримок, і швидше приймати рішення на основі даних.

1.9 Вразливості IoT

Розумні та підключені пристрої IoT пропонують кілька способів покращення процесів і продуктивності, покращення роботи користувачів і зниження витрат у різних галузях і середовищах. Хоча переваги IoT-пристроїв можна спостерігати на фабриках, лікарнях, автомобілях, будинках і містах, притаманні їм вразливості справді створюють нові ризики та виклики безпеки. Ці вразливості залишають мережі відкритими для кібератак, які можуть небезпечним чином порушити розвиток промисловості та економіки.

Пристрої Інтернету речей вразливі здебільшого тому, що в них відсутні необхідні вбудовані засоби контролю безпеки для захисту від загроз. Основною причиною є обмежене середовище та обмежені обчислювальні можливості цих пристроїв. Пристрої IoT зазвичай є пристроями з низьким енергоспоживанням, і їх можливості дозволяють виконувати лише певні функції. В результаті вони не витримують включення засобів контролю та механізмів безпеки та схем захисту даних.

Уразливості пристроїв Інтернету речей можуть дозволити кіберзлочинцям захопити їх і продовжувати атаки на інші критичні системи. Критичність підключених пристроїв і вплив атаки на них визначає важливість виробництва цих пристроїв з підходом «Безпека за проектом».

Кіберзлочинці прагнуть використати відомі вразливості пристроїв IoT і перетворити їх на зомбі або ботнети IoT. У 2016 році атака ботнету Mirai знищила популярні сайти та служби (після атак DDoS), захопивши тисячі скомпрометованих домашніх пристроїв IoT. Крім порушень безпеки, вразливості Інтернету речей є основною причиною багатьох порушень конфіденційності, що тягне за собою величезні законодавчі санкції за порушення таких правил, як GDPR, CCPA, HIPAA та PCI DSS.

Не всі пристрої Інтернету речей мають обчислювальну потужність для інтеграції складних брандмауерів або антивірусного програмного забезпечення, деякі ледве мають можливість підключатися до інших пристроїв. Безпека IoT є критичною значною мірою через розширену поверхню атак загроз, які вже розповсюджені по мережі. До цих загроз додається небезпечна практика серед користувачів і організацій, які, можливо, не мають ресурсів або знань для найкращого захисту своїх екосистем IoT.

Закон про покращення кібербезпеки Інтернету речей 2020 року – щоб подолати розширений ландшафт загроз і обмежити вразливість пристроїв IoT для федеральних агентств і служб, уряд США підписав закон про поліпшення кібербезпеки IoT від 2020 року. Закон зобов’язує NIST створювати стандарти кібербезпеки для підключених пристроїв, придбаних і використовується федеральними агентствами.

Відповідно до Закону, NIST має розробляти та публікувати «стандарти та рекомендації щодо належного використання та управління» пристроями IoT, які належать або контролюються федеральними агентствами, які підключені до федеральних мереж. Ці рекомендації також мають містити «мінімальні вимоги безпеки для управління ризиками кібербезпеки», властиві цим пристроям.

Крім того, закон передбачає, що федеральні агенції утримуються від закупівлі або отримання, поновлення контракту на закупівлю чи отримання або використання пристрою IoT, якщо пристрій не відповідає інструкціям NIST. У

відповідь на Закон про покращення кібербезпеки IoT NIST випустив чотири нові публікації:

– SP 800-213 – інструкції з кібербезпеки пристроїв IoT для федерального уряду: встановлення вимог щодо кібербезпеки пристроїв IoT;

– NISTIR 8259B – базовий базовий рівень нетехнічної підтримки IoT;

– NISTIR 8259C – створення профілю з використанням базового та нетехнічного базового рівня IoT;

– NISTIR 8259D – профіль із використанням базового та нетехнічного базового рівня IoT для федерального уряду.

Метою цих чотирьох документів є створення спільної системи кібербезпеки між урядом і виробниками IoT-пристроїв для пристроїв IoT, які закуповуються та використовуються федеральними агентствами.

Проект Open Web Application Security Project (OWASP), некомерційний фонд для вдосконалення програмного забезпечення, опублікував 10 найпопулярніших уразливостей Інтернету речей, які є чудовим ресурсом як для виробників, так і для користувачів [10].

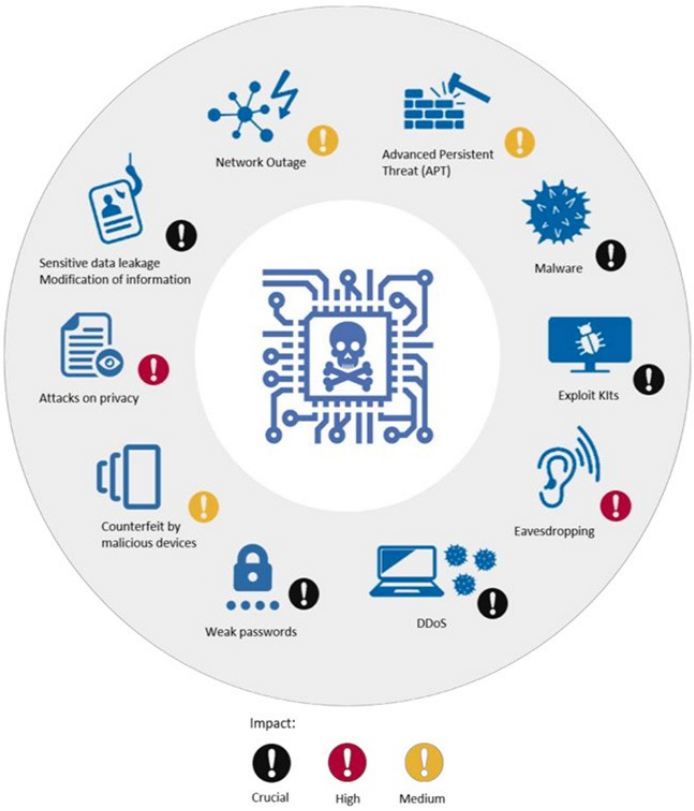

Рисунок 1.7 – Способи атак на системи IoT

Загалом, основними вразливостями систем Інтернету вважають:

Слабкі паролі, які можна вгадати або жорстко закодовані – використання легко примусових, загальнодоступних або незмінних облікових даних, включаючи backdoor у мікропрограмі чи клієнтському програмному забезпеченні, надають несанкціонований доступ до розгорнутих систем. Слабкі паролі, паролі за замовчуванням і жорстко закодовані – це найпростіший спосіб для зловмисників зламати пристрої Інтернету речей і далі запускати великомасштабні ботнети та інше шкідливе програмне забезпечення. Управління паролями в розподіленій екосистемі Інтернету речей – це трудомісткий і важкий обов’язок, тим більше, що пристроями IoT керують по повітрю.

Небезпечні мережеві послуги – непотрібні або незахищені мережеві послуги, що працюють на самому пристрої, особливо ті, що мають доступ до

Інтернету, які ставлять під загрозу конфіденційність, цілісність/автентичність або доступність інформації або дозволяють несанкціоноване дистанційне керування. Зловмисники намагаються використати слабкі місця в протоколі зв'язку та службах, які працюють на пристроях Інтернету речей, щоб зламати та порушити конфіденційну інформацію, якою обмінюються пристрій і сервер. Атаки типу «Людина посередині» (MITM) спрямовані на використання цих вразливостей для захоплення облікових даних, які використовуються для аутентифікації цих кінцевих точок, і подальшого використання цих облікових даних для запуску більш масштабних шкідливих атак. Тому вкрай важливо забезпечити зв'язок IoT за допомогою найкращих практик галузі.

Небезпечні інтерфейси екосистеми – небезпечні веб-інтерфейси, серверний API, хмарні або мобільні інтерфейси в екосистемі за межами пристрою, що дозволяє компрометувати пристрій або пов'язані з ним компоненти. Поширені проблеми включають відсутність аутентифікації/авторизації, відсутність або слабке шифрування, а також відсутність фільтрації введення та виведення. Потрібен надійний механізм аутентифікації та авторизації, щоб пом'якшити небезпечні веб-інтерфейси, серверні API, хмарні або мобільні інтерфейси в екосистемі Інтернету речей.

Для захисту ідентичності пристроїв IoT було розроблено кілька рішень, які враховують характер обмежень цих кінцевих точок. Використовуючи ефективний механізм ідентифікації пристрою, щоразу, коли сервер спілкується з пристроєм IoT, сервер зможе відрізнити дійсну кінцеву точку від шахрайської, змусивши кінцеву точку пройти автентифікацію.

Відсутність безпечного механізму оновлення – немає перевірки мікропрограмного забезпечення на пристрої, відсутність безпечної доставки повідомлень (відсутність шифрування під час доставки), відсутність механізмів запобігання відкату та відсутність повідомлень про зміни безпеки через оновлення. Несанкціоноване оновлення програмного та мікропрограмного

забезпечення є основним вектором загроз для запуску атак на пристрої IoT. Пошкоджене оновлення може порушити роботу критичних пристроїв Інтернету речей і мати фізичні наслідки в таких секторах, як охорона здоров’я або енергетика. Щоб захистити оновлення мікропрограми та програмного забезпечення, нам потрібно забезпечити доступ до оновлень і перевірити джерело та цілісність оновлень.

Використання незахищених або застарілих компонентів – може призвести до скомпрометації пристрою. Це включає небезпечне налаштування платформ операційної системи та використання сторонніх програмних або апаратних компонентів із скомпрометованого ланцюга поставок. Безпека екосистеми IoT може бути порушена через вразливості залежностей від програмного забезпечення або застарілих систем. Використання виробниками застарілого або незахищеного програмного забезпечення, включаючи компоненти з відкритим кодом, для створення своїх пристроїв IoT створює складний ланцюжок поставок, який важко відстежити. Ці компоненти можуть успадкувати вразливості, відомі зловмисникам, створюючи розширений ландшафт загроз, які очікують на використання.

Недостатній захист конфіденційності – особиста інформація користувача, що зберігається на пристрої або в екосистемі, яка використовується небезпечно, неналежним чином або без дозволу. Багато розгорнутих пристроїв IoT збирають персональні дані, які необхідно безпечно зберігати та обробляти для забезпечення відповідності різним нормам конфіденційності, таким як GDPR або CCPA. Ці особисті дані можуть бути будь-якими: від медичної інформації до споживання електроенергії та поведінки за кермом. Відсутність належного контролю поставить під загрозу конфіденційність користувачів і матиме правові наслідки.

Небезпечна передача та зберігання даних – відсутність шифрування чи контролю доступу до конфіденційних даних будь-де в екосистемі, у тому числі в

стані спокою, передачі або під час обробки. Захист даних IoT – як у стані спокою, так і під час передачі – має велике значення для надійності та цілісності додатків IoT. Ці дані використовуються в автоматизованих процесах прийняття рішень і контролю, які можуть мати серйозні фізичні наслідки. Дуже важливо, щоб ми ефективно захищали ці дані. Використання надійного шифрування протягом усього життєвого циклу даних Інтернету речей і адаптивного контролю ідентифікації та доступу допоможуть захистити дані Інтернету речей від компрометації та злому.

Відсутність керування пристроями – відсутність підтримки безпеки на пристроях, розгорнутих у виробництві, включаючи управління активами, оновлення, безпечне виведення з експлуатації, моніторинг систем та можливості реагування. Однією з найважливіших завдань і однією з найбільш значущих проблем безпеки в екосистемі IoT є керування всіма пристроями протягом їхнього життєвого циклу. Якщо в екосистему IoT будуть введені неавторизовані пристрої, вони зможуть отримувати доступ і стежити за корпоративними мережами, а також перехоплювати трафік та інформацію. Основними проблемами керування пристроями IoT є надання, експлуатація та оновлення пристроїв. Виявлення та ідентифікація пристроїв IoT є необхідним першим кроком у моніторингу та захисту цих пристроїв.

Небезпечні налаштування за замовчуванням – пристрої або системи, які поставляються з небезпечними налаштуваннями за замовчуванням або не мають можливості зробити систему більш безпечною, обмежуючи операторів у зміні конфігурацій. Пристрої Інтернету речей постачаються з жорстко запрограмованими налаштуваннями за замовчуванням, які легко не захищають і їх легко зламати зловмисники. Як тільки ці налаштування будуть скомпрометовані, зловмисники можуть або шукати жорстко закодовані паролі за замовчуванням, приховані бекдори та вразливості у мікропрограмі пристрою. У той же час ці налаштування користувачеві важко змінити. Глибоке розуміння

цих налаштувань і пробілів безпеки, які вони створюють, є першим кроком до впровадження відповідних елементів керування для посилення цих пристроїв.

Відсутність фізичного загартування – дозволяє потенційним зловмисникам отримувати конфіденційну інформацію, яка може допомогти у майбутній віддаленій атаці або взяти локальний контроль над пристроєм. Пристрої IoT розгортаються в розсіяних і віддалених середовищах – не зберігаються в жодному контрольованому середовищі, але надаються в поле для виконання своїх операцій. Зловмисник може порушити послуги, що пропонуються пристроями IoT, отримавши доступ і втрутившись у фізичний рівень. Такі дії можуть перешкодити, наприклад, датчикам виявити такі ризики, як пожежа, повінь та несподіваний рух. Ми повинні переконатися, що обладнання захищене від втручання, фізичного доступу, маніпуляцій та саботажу.

Налагоджені цифрові сертифікати, керовані PKI, можуть допомогти організаціям усунути багато з вищезгаданих вразливостей. Ключом до забезпечення поширення IoT-пристроїв є можливість їх ідентифікувати. Цифрові сертифікати чудово підходять для надання ідентифікаційних даних машин і для аутентифікації розподіленої екосистеми IoT. Багато виробників та організацій IoT вже використовують переваги цифрових сертифікатів для ідентифікації пристрою, аутентифікації та шифрування. Однак випуск і керування тисячами цифрових сертифікатів у всій корпоративній екосистемі IoT може бути складним, якщо рішення для керування сертифікатами не передбачає автоматизації та масштабованості.

Рішення для управління ідентифікаторами машини допоможе організаціям захистити свою екосистему Інтернету речей, забезпечуючи унікальні надійні ідентифікатори, визначаючи та забезпечуючи політику та стандарти безпеки, масштабуючи безпеку та підтримуючи надійну й ефективну безпеку, не ставлячи під загрозу ефективність та роботу обмежених пристроїв IoT.

Оскільки IoT розширюється, жодна компанія не може скинути з уваги величезні ризики безпеки, пов’язані з безліччю можливих недоліків інфраструктури. Цифрові сертифікати PKI з автоматизованим керуванням не вирішать усіх проблем із безпекою, але вони є важливою частиною рівняння, яку вам потрібно оцінити та адаптувати до потреб вашої організації.

1.10 Аналіз існуючих систем

Системи дистанційного моніторингу мікроклімату можна розділити на побутові та виробничі. Побутові системи зазвичай мають невисоку вартість та не складну структуру, часто має обмеженний функціонал. Такі системи будуються за концепцією IoT (Internet of things), тобто усі прилади у приміщенні з’єднуються між собою за допомогою безпровідних технологій, таких як WiFi або Bluetooth, а інформація передається на пристрій користувача.

Тут найчастіше використовуються стандарти Zigbee та Z-wave, завдя яким усі прилади об’єднуюь в одну єдину систему. Ці стандарти використовують не лише сенсори та виконавчі пристрої, а і базові станції, які розташовуються в певних місцях між іншими приладами для створення необхідного покриття сигналом. Але незважаючи на те, що з одного боку такі системи не складні в об’єднанні фізичних пристроєв, принцип обробки даних з сенсорів є складним завданням. Інформація, яку відправляють пристрої є постійним потоком, і необхідний алгоритм, який буде обробляти цю інформацію в реальному часі.

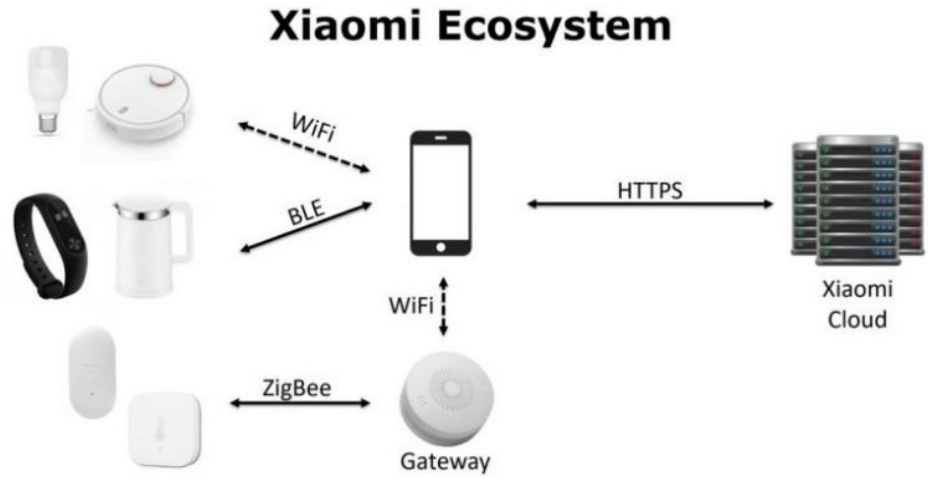

Рисунок 1.8 – Система домашньої автоматизації компанії Xiaomi

Так, компанія Xiaomi випустила лінійку смарт-приладів для тих, кто хоче власноруч зробити систему розумного будинку. До цих приладів відносяться різноманітні сенсори, розумні розетки, побутова електроніка і навіть робототехніка. Для об’єднання у єдину систему використовується спеціальні шлюзи та технології безпровідного зв’язку, а застосунок користувача дає змогу керувати та відстежувати показання приладів у режимі online.

У виробничій сфері зазвичай використвоються дорогі та складні системи, які були розроблені спеціально під замовлення. Залежно від специфікації середовища, в якому будуть розташовані прилади, системи можуть відрізнятися за розміром, принципом з’єднання приладів та способом зберігання даних. Розмір системи визначається за кількістю приладів у системі та відстанюю їх розташування, за принципом з’єднання виділяють провідні та безпровідні, а за спосібом зберігання даних є локальні(стаціонарний пристрій для зберігання даних) та видалені(Cloud-сервера). Використання видалених серверів є найпопулярнішим способом, адже за невелику плату можна орендувати сховище, при цьому гарантовано мати стабільність та безпеку інформації.

Яскравим прикладом таких систем є продукція компанії Acrel[1]. Дана фірма спеціалізується на створенні систем дистанційного моніторингу на

замовлення. Залежно від специфікації приміщень, система може бути змінена під конкретні потреби. Своїм замовникам компанія пропонує багатий вибір різноманітних сенсорів та виконавчих механізмів, які відрізняються розмірами та типами кріплень, центральний пристрій-станцію, власний Cloud сервер та унікальне програмне забезпечення. Основними перевагами є видалений моніторинг, сповіщення персоналу про зміну показників та майже відсутність проводів у системі, адже усі прилади з'єднуються між собою за допомогою безпровідних технологій зв'язку.

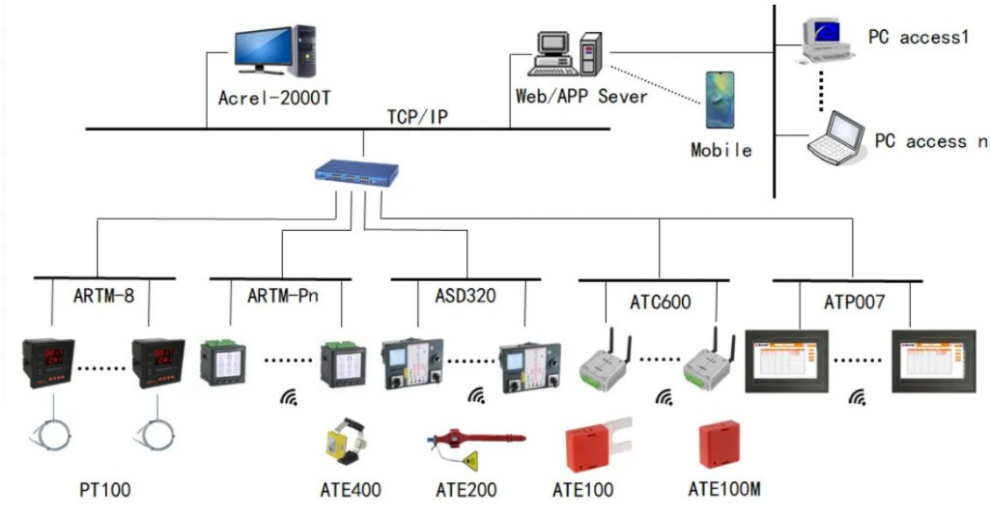

Рисунок 1.9 – Система моніторингу компанії Acrel

1.11 Постанова задачі

Предметом даного дослідження є технології Інтернету речей та їх взаємозв'язок з хмарними сервісами. Об'єктом дослідження виступає система дистанційного моніторингу мікроклімату.

Основною метою кваліфікаційної роботи є розробка системи дистанційного моніторингу мікроклімату. Для виконання поставленої задачі необхідно з'ясувати принцип роботи, скласти структурну схему системи,

проаналізувати сучасний ринок мікроконтролерів і сенсорів та обрати необхідні складові. Також необхідно обрати протоколи передачі даних для взаємодії компонентів. Необхідно переконатися у коректності роботи функцій системи, дізнатися про її недоліки, якщо вони існують, розміркувати щодо покращення роботи системи та додавання нового функціоналу.

Загалом, система повинна мати центральний пристрій з приймачем, сенсор з підтримкою обраної безпровідної технології, програмне забезпечення для обробки даних та хмарний сервіс для зберігання та отримання доступу до цих даних. При цьому система має бути легкою у використанні.

2 ВИБІР ТЕХНОЛОГІЙ ДЛЯ ВИРІШЕННЯ ЗАДАЧ

2.1 Апаратна платформа

Мікроконтролер – це комп’ютер на одному чіпі, який включає ядро процесора, пам’ять і периферійні пристрої введення/виводу, і зазвичай він використовується для виконання певних функцій у вбудованій системі. Пам’ять мікроконтролера, порти вводу/виводу та ПЗУ/ОЗУ розташовані всередині мікросхеми. Периферійні пристрої мікропроцесора є зовнішніми для нього. Таким чином, пам’ять, послідовний інтерфейс, таймери та порти вводу/виводу вбудовані до мікропроцесора.

Мікроконтролери дуже сфокусовані. Мікроконтролери призначені для певних програм і використовуються для виконання заздалегідь визначених завдань. Не слід плутати мікроконтролер та мікропроцесор. На відміну від першого, мікропроцесори використовуються для програм, які вимагають інтенсивної обробки, наприклад, для запуску великих графічних програм на комп'ютерах та ноутбуках. Тобто не має потреби використовувати комп'ютерний

процесор для вирішення подібних задач.

Мікроконтролери можуть зробити багато за відносно невелику вартість. Вони не пропонують високу швидкість і велику пам'ять, як мікропроцесор. Мікроконтролер має керувати специфічним завданням, і, таким чином, не вимагає захоплюючої швидкості та великої кількості пам’яті, тоді як мікропроцесор буде задіяний для виконання складних, ресурсомістких завдань. Тактова частота мікроконтролера може становити 300 МГц, у той час як швидкість процесора може досяшати 4 ГГц та більше.

Універсальність мікроконтролерів і простота їхнього застосування для вирішення широкого кола задач спричинили справжню технічну революцію. Попри величезну кількість персональних й інших типів потужних комп’ютерів на сьогодні більша частка всіх комп’ютерних систем належить мікроконтролерам [11].

На сьогоднішній день існують дуже багато подібних апаратних платформ. Найбільш популярні – це мікроконтролери Arduino, STM32, Raspberry Pi. Усі вони відрізняються між собою вартістю, обчислювальною потужністю, кількістю портів та різними протоколами передачі даних.

Raspberry Pi – одноплатний комп’ютер малого розміру, який набув популярності у всьому світі завдяки відносно невеликій вартості та обчислювальної потужності. В основному цей комп’ютер працює під операційними системами на Linux-ядрі, але за бажанням можна поставити систему Windows 10 IoT. Взагалі на даний час існують вже понад 20 операційних систем, і продовжуються розроблятися нові. Розглянемо мікрокомп’ютер Raspberry Pi 3 Model B+.

Рисунок 2.1 – Мікроконтролер Raspberry Pi 3 Model B+

Комп’ютер має розміри банківської картки, але при цьому має при собі усі необхідні інтерфейси та технології. Встановлений 64-бітний процесор ARM Cortex-A53 має частоту 1,2ГГц, графічний процесор VideoCore4, який здатен кодувати, декодувати та відображувати відео у Full-HD форматі. Розмір оперативної пам’яті дорівнює 1Гб, але ця пам’ять ділиться з графічною системою. Серед безпровідних інтерфейсів є Wi-Fi та Bluetooth версії 4.1, для доступу в інтернет також є роз’єм Etherneth. Завдяки роз’єму HDMI є змога підключення монітору або телевізору, а наявність чотирьох портів USB дає змогу підключати різну переферію, таку як клавіатуру або комп’ютерну миш, а також є можливість підключення SD карт. Звук можна виводити по каналу HDMI або через роз’єм 3,5мм для підключення навушників або колонок [12].

Окрім усього цього на платі є окремі спеціальні слоти:

- CSI-S;
- DSI;
- UART;
- I2C/TWI;
- SPI;

– піни для живлення;

– 40 портів для підключення різних пристроїв для вводу-виводу.

2.2 Програмна платформа

Для того, щоб комп'ютер почав працювати, необхідно обрати оперційну систему. Raspbian – операційна система, розроблена для роботи з мікрокомп’ютерами Raspberry Pi з процесорами ARM, в якості основної середи робочого стола використовується PIXEL(Pi Improved Xwindows Environmemt Lightweight).

Оскільки мікрокомп’ютер не має своєї внутрішньої пам’яті, операційну систему записують на SD карту, яку потім підключають до пристрою та встановлюють. Дана програмна платформа також має вбудовані середовища створення програм, такі як Scratch, Sonic PI, Python, а також додаткові програми та компоненти, такі як GPIO, Terminal, Wordpress, Mathematica та інші. Окрім цього є можливість розгортання власного серверу на пристрої за допомогою інтегрованих модулів Python [13].

Рисунок 2.2 – Операційна система Raspberry Pi OS

Python – високорівнева мова програмування загального призначення з автоматичним керуванням пам’яттю, яка вже встановлена в дистрибутив Raspbian. Підтримує процедурне, структурне, функціональне та об’єктноорієнтоване програмування. Цей інтерпретатор переводить початковий код програми в високорівневий байт-код, який потім виконується у стековій віртуальній машині.

Стандартна бібліотека має багатий функціонал, включаючи як і простий функціонал роботи з текстами, так і складний для роботи з мережевними застосунками. Окрім стандартної бібліотеки є можливість використання сторонніх, що розширює можливості написання програм. Сама мова Python використовується у різних напрямах, наприклад у створенні ігр, аналізі великих об’ємів даних, веб-застосунки та навіть у машинному навчанні, до того ж завдяки своїй структурі та синтаксису, мова є легкою для вивчення. Окрім цього модулі Python дають змогу встановлювати віртуальне оточення з подальшим встановленням програмного забезпечення у ньому.

Для того щоб система почала працювати та виконувати поставлені задачі, необхідно встановити спеціалізовану систему. Homeassistant – операційна система для побудування систем Інтернету речей. Вона безкоштовна та має підтримку для встановлення на мікроконтролери сімейства Raspberry Pi та виступає у ролі центральної керуючої системи. Графічний інтерфейс самої системи(Рисунок 2.3), яка встановлена на мікроконтролері виконана у вигляді командного терміналу, та підтримує команди, які зазвичай використовує операційна система Linux.

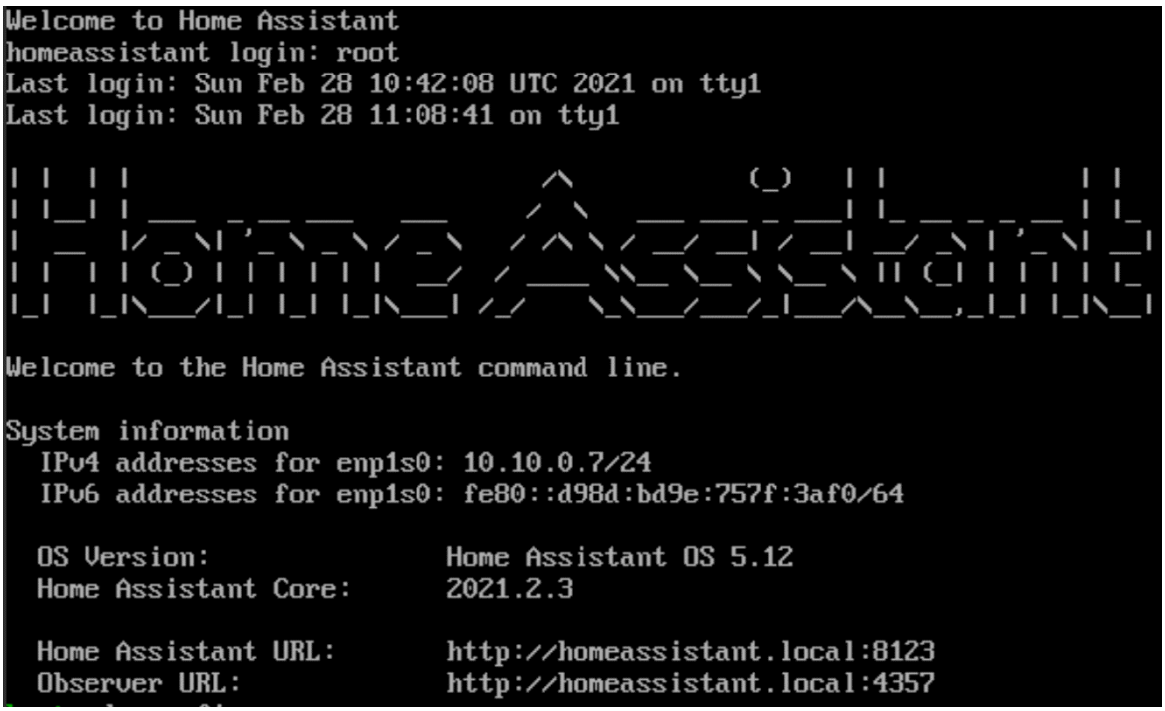

Рисунок 2.3 – Графічний інтерфейс системи Homeassistant

Не зважаючи на такий простий інтерфейс на пристрої, коростувацький інтерфейс серверу дуже зручний та має велике різноманіття налаштувань а також безкоштовних додатків, які можуть бути встановлені. Окрім цього даний сервіс має можливість інтегрувати свою систему з різними застосунками, такими як Google Assistant та Amazon Web Services. Але найважливіше це те, що дана система може бути розташована в Cloud, а отже доступ до неї може бути отриманий з будь-якої точки планети, а не лише з тієї ж мережі у якій росташована сама система.

2.3 Wi-Fi

Технологія Wi-Fi була заснована за рішенням Федеральної комісії зв’язку США ще у 1985 році, яке випустило діапазони радіочастот на частотах 900 МГц, 2,4 ГГц і 5,8 ГГц для неліцензованого використання ким завгодно. Після цього

різні фірми почали будувати бездротові мережі та пристрої, щоб скористатися перевагами нещодавно впроваджених радіочастот, але без загального стандарту рух залишався фрагментованим, оскільки пристрої різних виробників рідко були сумісні. Згодом комітет науковців та розробників у цій галузі розробив загальний стандарт під назвою 802.11, який був затверджений Інститутом інженерів з електротехніки та електроніки (IEEE) у 1997 році.

Через два роки група великих компаній сформувала Альянс сумісності бездротового Ethernet (WECA), яка явяла собою глобальну некомерційну організацію, створену для просування нового стандарту бездротового зв’язку. WECA назвала нову технологію Wi-Fi. Наступні стандарти IEEE для Wi-Fi були введені, щоб забезпечити більшу пропускну здатність. На даний час існують такі стандарти Wi-Fi: 802.11a, 802.11b, 802.11g, 802.11n, 802.11ac і 802.11ax тощо.

Оригінальний стандарт 802.11 дозволяв максимальну швидкість передачі даних лише 2 мегабіти в секунду (Мбіт/с), 802.11n, представлений у 2007 році, має максимальну швидкість 600 Мбіт/с. Як правило, типовий діапазон для з’єднання Wi-Fi – 100 метрів, але найбільш поширена дальність обмежена 10-35 метрами. Навіть потужність антени та частотне мовлення можуть істотно вплинути на дальність дії мережі. Радіус дії і швидкість Wi-Fi підключення до Інтернету залежать від навколишнього середовища, а також від того, чи надається воно всередині або зовні приміщення.

Таким чином, швидкість різних пристроїв, що використовують підключення до Інтернету Wi-Fi, збільшується, оскільки пристрій також стає ближче до основного джерела, а швидкість знижується в міру того, як комп'ютер віддаляється від джерела [14].

2.4 Технологія Zigbee

ZigBee – це протокол бездротового зв'язку, який використовується для

з'єднання між машинами (M2M) і керування ними. Це малопотужна бездротова мережа, яка дозволяє пристроям підключатися та передавати команди в обох напрямках. Таким чином, це дозволяє бездротовій технології вийти за межі того, що ми зараз відчуваємо, наближаючи нас до автоматизації розумних систем [15].

ZigBee в певній мірі схожий на Bluetooth, вони обидва є протоколами безтротової мережі та працюють в одному діапазані частот, але вони все ж таки достатньо різні, та не можуть утворювати одну мережу один з одним. Хоча оскільки ці протоколи мають різне призначення, потребності в їх об’єднанні немає. В той час коли Bluetooth слушить для з’єднання певної електроніки в зручному для користувача способі, ZigBee служить для з’єднання спеціалізованих пристроїв для усунення участі людини у процессі взагалі [16].

Стандарт Zigbee був впроваджений ще у 2003 році, і продовжує набувати попит, адже все більш популярними становляться системи автоматизації розумних пристроїв. Цей протокол використовується в автономних пристроях з невеликим споживанням енергії, а також в різних додатках, які вимагають низьку швидкість передачі даних.

Основна мета Zigbee – створення самоорганізованих мереж з комірчастою топологією. У більшості випадків такі мережі є сукупченням інших мереж. Такаж особливістб даного стандарту є можливість пудключення дуже великої кількості пристроїв, при цьому зберігая стабільність мережі та відносту дешевизну.

У мережах з відключеними маячками(пристроями) використовується механізм доступу до каналів. У цьому типі мережі приймальні пристрої Zigbee зазвичай підтримують свої приймачі включеними тривало, що вимагає більш потужної енергопідтримки. Однак це дозволяє різнорідним мережам розподілити режими роботи пристроїв – деякі пристрої тривало приймають повідомлення, поки інші тільки передають. Пристрої ZigBee у системах можна поділити на три групи.

Координатор – це найбільш потужні пристрої, які утворюють корінь мережі та можуть підключатися до інших мереж. Для кожної мережі необхідний координатор, оскільки це пристрій, який запускає мережу. Він зберігає мережеву інформацію, у тому числі є сховищем для ключів безпеки.

Маршрутизатор – маршрутизатори виконують дві функції в мережі ZigBee. Вони виконують звичайну функцію маршрутизатора для підключення пристроїв до мережі. Крім того, вони можуть запускати програми в мережі.

Кінцевий пристрій – це окремі пристрої, підключені до мережі. Вони не мають можливості передавати дані з інших пристроїв. Кінцевий пристрій містить достатньо функціональних можливостей, щоб спілкуватися з мережею, зокрема спілкуватися з координатором або маршрутизатором. Коли їм не потрібно спілкуватися, вони переходять у «сплячий» режим, щоб заощадити батареї.

Вузол Zigbee в пристрої може приймати постійно, з того часу, як він підключений до загального живлення та мережі. Потім, при отримані сигналу, пристрій виконує команду та знову переходить у режим очікування. У таких мережах вузол пристроя повинен бути, щонайменше, маршрутизатором Zigbee, якщо не координатором, вузол ключа, зазвичай, це кінцеве пристрій Zigbee.

Програмне забезпечення, розроблене з метою спрощення процесу побудови невеликих недорогих мікропроцесорів. Радіорозробки, використовувані в Zigbee, ретельно оптимізовані, щоб досягти низької ціни серед великого числа продукції цієї лінійки. Є кілька аналогових каскадів, де, можливо, використовуються цифрові контури.

Є ще один схожий стандарт, який називається Z-Wave. Як і у випадку з Zigbee, він являє собою мережу мереж та використовується у системах автоматизацій. Але не зважаючи на таку схожість, Zigbee та Z-Wave відрізняються один від одного:

– кількість срибків у мержі Z-Wave між контролером та пристроєм обмежена чотирма, у Zigbee вона необмежена;

– Z-Wave використовує одну частоту роботи, Zigbee використовує два;

– на відміну від Z-Wave, Zigbee – відкритий стандарт.

У даній роботі для з’єднання приладів у системі використовується приймач Zigbee CC2531 Sniffer.

Рисунок 2.4 – Пристрій Zigbee CC2531 Sniffer

Даний приймач дозволяє підключати різноманітні пристрої до стороннього програмного забезпечення. Комплектація даного пристрою складається з самого приймача у вигляді плати з USB роз’ємом, програматор CC-Debugger, антена, перехідники та шлейфи. Загалом, має наступні характеристики:

– стандарти: ZigBee, 802.15.4;

– алгоритм шифрування: AES-128;

– кількість портів введення-виводу: 8;

– частота: 2,4 ГГц;

– мікроконтролер (SOC): CC2531;

- пропускна спроможність: 250 кбіт/с;
- робоча частота: 2,405-2,485 ГГц;
- потужність: 20 мА для прийому; 25 мА для передачі;
- габаритні розміри: 41 х 16 х 2 мм;
- маса: 30 гр.

2.5 Протокол передачі даних MQTT

MQTT, або Message Queue Telemetry Transport - це легкий, компактний і відкритий протокол обміну даними, створений для передачі даних на віддалених локаціях, де потрібен невеликий розмір коду і є обмеження по пропускній здатності каналу. Обмін повідомленнями в протоколі MQTT здійснюється між клієнтом, який може бути видавцем або підписником повідомлень, і брокером повідомлень.

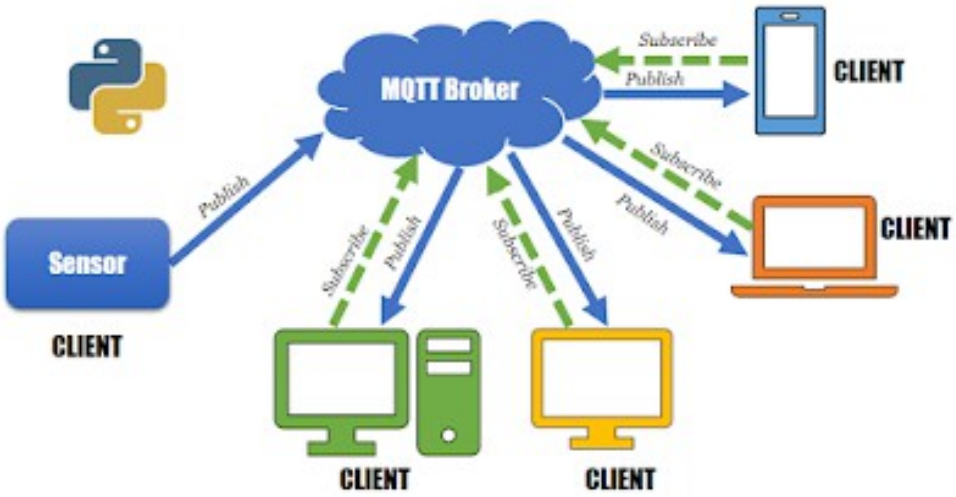

Рисунок 2.5 – Модель publisher/subscriber(підписник-видавник) протоколу MQTT

Видавець відправляє дані до MQTT брокеру, вказуючи в повідомленні

певну тему та топік. Підписники можуть отримувати різні дані від безлічі видавців залежно від підписки на відповідні топіки. Повідомлення проходять шлях від видавця, через брокера до одного або кількох передплатників, що використовують теми. Темами є ієрархічні рядки у форматі UTF-8. Кожен рівень теми розмежовується косою рискою. Кожне повідомлення видавця має містити тему.

Щоб отримати опубліковане повідомлення, організація, яка споживає повідомлення, повинна підписатися на ту саму тему, а брокер надсилає отримане повідомлення лише тим клієнтам, які підписалися на одну і ту ж тему. MQTT має підтримку постійних повідомлень, що зберігаються у брокера. Під час публікації повідомлень клієнти можуть вимагати, щоб брокер зберігав повідомлення. Зберігається лише останнє постійне повідомлення. Коли клієнт підписується на тему, будь-яке збережене повідомлення буде надіслане клієнту. На відміну від черги повідомлень, брокери MQTT не дозволяють зберігати збережені повідомлення всередині сервера [17].

2.6 Сенсор температури та вологості

Зробивши вибір основного пристрою та технології взаємодії між пристроями, необхідно обрати сенсор. Xiaomi aqara Temperature and Humidity Sensor – цифровий датчик температури та вологості, який підтримує технологію Zigbee. Пристрій має повну підтримку, й як і інші датчики на батарейках – це кінцевий пристрій, тобто він не може пересилати через себе команди інших учасників мережі, а лише спілкуватися з центральним пристроєм. Має наступні характеристики:

- діапазон робочих температур: -20 - 60 С;
- діапазон вологості: 0 - 100%;
- елемент живлення: CR2032;
- розмір: 36x36x11.5.

Рисунок 2.6 – Сенсор температури та вологості Xiaomi aqara Temperature and Humidity Sensor

Даний сенсор підходить для встановлення у житлових приміщенях та має можливість інтеграції до систем smart-house. Окрім цього має наступні особливості:

– відстежування температури та вологості в приміщенні;
– синхронізація показників температури і вологості в реальному часі з додатком;
– може запускати пристрої Wi-Fi в розумній сцені;
– можливість додавання камери;
– повідомлення про низький заряд батареї;
– швидке та легке встановлення на робоче місце.

3 РОЗРОБКА СИСТЕМИ

3.1 Структура системи

Перш ніж приступати до розробки системи, необхідно з’ясувати її структуру. Система складається з IoT датчиків або виконавчих механізмів, цетрального пристрою у вигляді мікроконтролеру Raspberry Pi Model 3B+ з підключеним до нього приймачем та встановленим на ньому відповідним програмним забезпеченням(операційна система, Homeassistant, додатки Zigbee2MQTT та MosquittoMqttBroker), роутер для з’єднання центрального пристрою до мережі Інтернет та Cloud сервер.

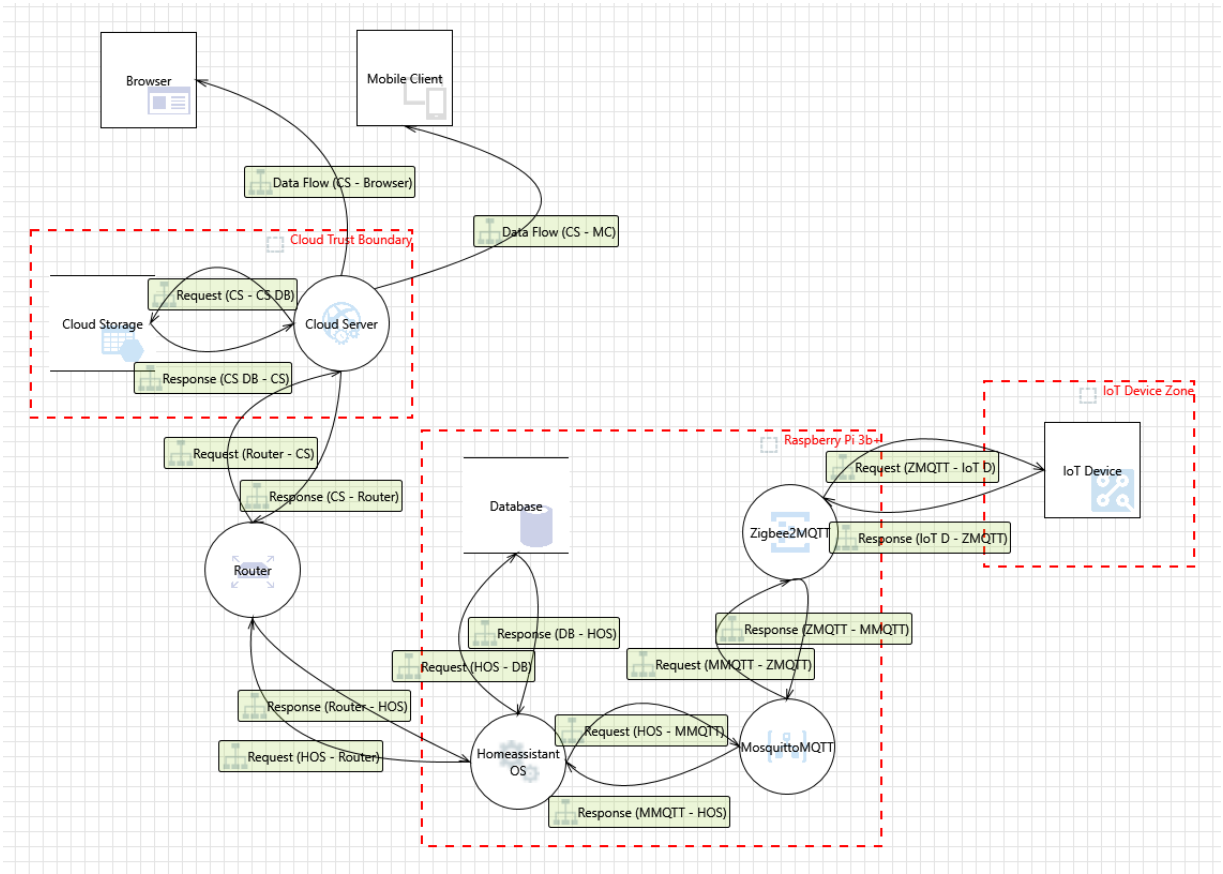

Рисунок 3.1 – загальна структура системи

Після запуску системи MQTTBroker відправить запит на отримання даних з сенсора. Цей запит спочатку надійде до Zigbee2mqtt, а потім, через приймач, безпосередньо на сенсор. В свою чергу сенсор надасть свої дані по тому ж маршруту, але в зворотньому напрямку. Ці запити будуть відправлятися з періодичністю, вказаною у конфігурації додатку. Отримані дані збережуться у сховищі Homeassistant на мікроконтролері та, синхронізувавшись з хмарою, відправить дані на видалений сервер. Коли користувач захоче зайти до хмарного серверу, він відправить запит на отримання цих даних зі сховища.

3.2 Встановлення Homeassistant

Система Homeassistant може бути встановлена на платформу Raspbery Pi у декілька способів. Перший і найлегший спосіб це запис образу дистрибутиву Hass.io на SD-карту або на флеш-накопичувач та встановлення його як основну систему мікроконтролера. Такий підхід потребує менше часу, але не дуже зручний у плані управління мікроконтролера, адже не має повного доступу до системи. Другий спосіб полягає у встановленні дистрибутиву у віртуальне оточення Python у системі Raspbian. Такий метод більш складний та потребує більше часу, але має переваги у вигляді повного доступу до системи, більш розширена підтримка різних пристроїв та можливість підключення та керування мікроконтролером з комп’ютера.

Для більш зручного процесу взаємодії з мікроконтролером, можна скористатися програмним забезпеченням VNC Connect. Данна система дозволяє дистанційно підключатися да здійснювати керування мікроконтролером з комп’ютера, які знаходяться в одній мережі.

Для здійснення підключення необхідно на стороні комп’ютера ввести IP адресу, яка вказана у клієнті на Raspbian, та вказати ім’я та пароль. Після

успішного підключення на комп'ютері відкриється вікно з системою Raspbery PI.

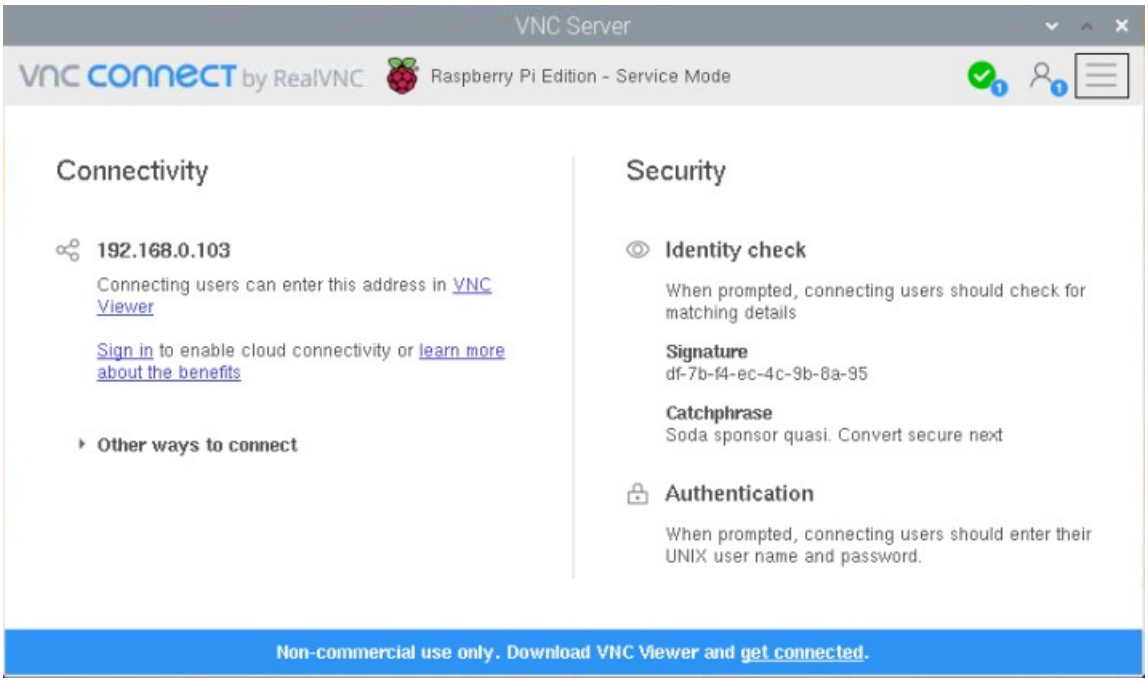

Рисунок 3.2 – Користувацький інтерфейс VNC server

Перш ніж встановити Homeassistant у віртуальне оточення, необхідно переконатися, що на системі встановлені усі необхідні компоненти а також їх останні версії. Для цього необхідно відкрити вікно терміналу та ввести наступні команди:

– sudo apt-get update та sudo apt-get dist-upgrade – оновлення усіх пакетів, встановлених на Raspbian;
– sudo apt-get install python3 python3-dev python3-venv python3-pip libffi-dev libssl-dev libjpeg-dev zlib1g-dev autoconf build-essential libopenjp2-7 libtiff5 – встановлення пакетів, необхідних для роботи Homeassistant.

Система Homeassistant потребує наявності версії Python-3.8.0 та вище, тож спочатку треба перевірити версію встановленої системи, і якщо вона нижче, встановити більш нову за допомогою команд:

– sudo apt-get install -y build-essential tk-dev libncurses5-dev libncursesw5-

dev libreadline6-dev libdb5.3-dev libgdbm-dev libsqlite3-dev libssl-dev libbz2-dev libexpat1-dev liblzma-dev zlib1g-dev libffi-dev – встановлення необхідних для Python-3.8.0 модулів;

- wget https://www.python.org/ftp/python/3.8.0/Python-3.8.0.tar.xz – завантаження дистрибутиву;
- tar xf Python-3.8.0.tar.xz – розпакування архіву;
- cd Python-3.8.0;
- ./configure --enable-optimizations --prefix=/usr – конфігурація;
- Make – компілювання розпакованих файлів;
- sudo make altinstall – встановлення компільованих файлів;
- sudo update-alternatives --install /usr/bin/python python /usr/bin/python3.8 1 – встановлення Python-3.8.0 за замовченням.

Наступним кроком буде створення ного користувача та запуск Homeassistant у віртуальному оточенні. Для цього необхідно виконати декілька команд у такій послідовності:

- sudo useradd -rm homeassistant -G dialout,gpio,i2c – створення нового користувача з вказанням необхідних аргументів;
- cd /srv – перехід до каталогу srv;
- sudo mkdir homeassistant – створення власного каталогу користувача
- sudo chown homeassistant:homeassistant homeassistant – передання створеного каталогу новому користувачеві;
- sudo -u homeassistant -H –s – переключення користувача homeassistant;
- cd /srv/homeassistant – перехід до каталогу homeassistant;
- python3 -m venv . – встановлення віртуального оточення у поточній директорії;
- source bin/activate – зчитування та виконання заданого файлу;
- python3 -m pip install wheel – встановлення wheel;
- pip3 install homeassistant – встановлення дистрибутиву homeassistant;

– hass – запуск Homeassistant у віртуальному оточенні.

Після виконання усіх команд, при першому запуску система Homeassistant створе базові конфігураційні файли та завантажить додаткові модулі. Після завершення цих операцій в інтерфейсі з’явиться IP-адреса серверу, яку необхідно ввести у браузер комп’ютера, який розташований у тій же мережі.

При першому вході користувачеві буде запропоновано створити новий обліковий запис, вказавши ім’я, логін та пароль. Після реєстрації користувач буде направлений на початкову сторінку(Dashboard).

3.3 ZigBee Sniffer CC2531

У даній системі використовується приймач USB ZigBee Sniffer CC2531. Для того щоб він працював, на нього необхідно встановити прошивку. Для цього в комплекті є програматор CC-Debugger, а на офіційній сторінці виробника є програмне забезпечення під назвою SmartRF Flash Programmer.

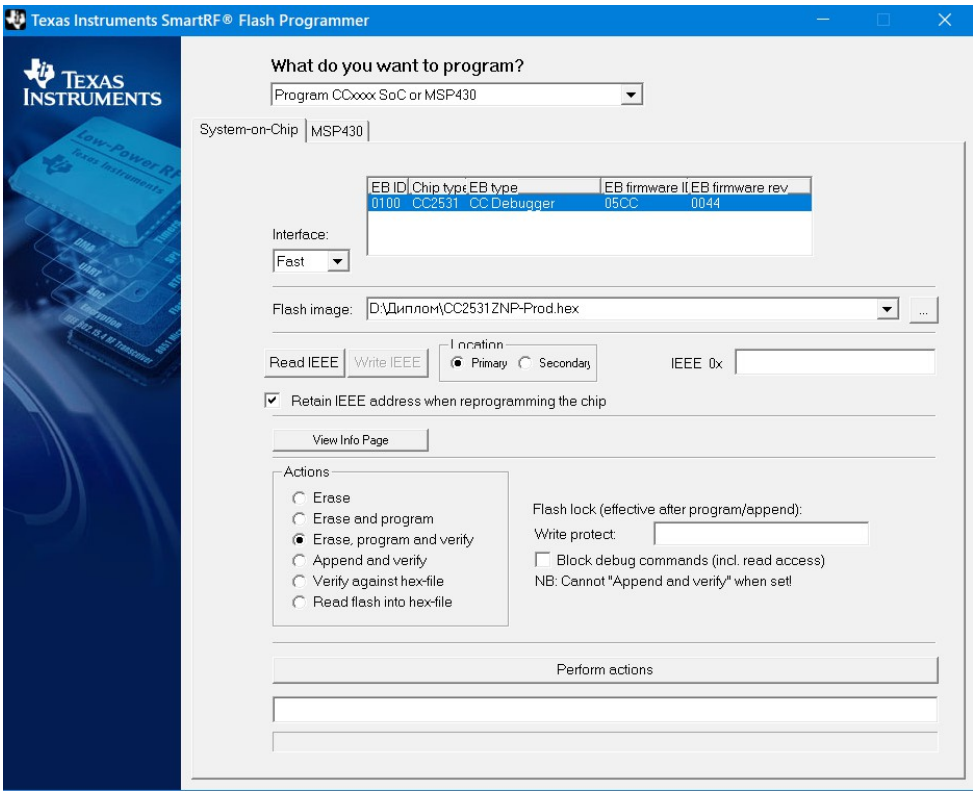

Рисунок 3.3 – SmartRF Flash Programmer

Підключаємо приймач до комп’ютера та програматора за допомогогою шлейфів та перехідникі з комплекту і чекаємо поки він з’явиться у вікні пристроїв. Далі обираємо файл з прошивкою та завантажуємо її на пристрій.

Рисунок 3.4 – Підключення приймача для прошивки

Після успішного завантаження прошивки, на приймачі повинен загорітися зелений світлодіод, який означає, що пристрій готовий до використання. Після цього необхідно підключити приймач до одного з портів мікроконтролера, вісля цього зайти на вкладку Superuser-System-Hardware та перевірити його наявність у списку.

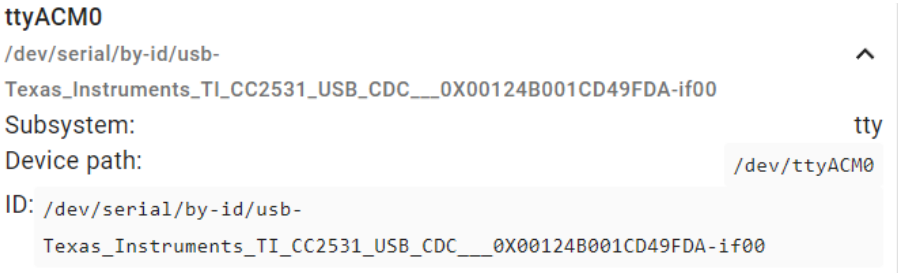

Рисунок 3.5 – порт приймача

3.4 Zigbee2mqtt

Для того щоб встановити додаток, треба зайти у вкладку Superuser, увійти у магазин додатків(ADD-ON-STORE) та додати репозиторій Github(https://github.com/danielwelch/hassio-zigbee2mqtt). Після цього натиснути кнопку Install для встановлення, та запустити додаток.

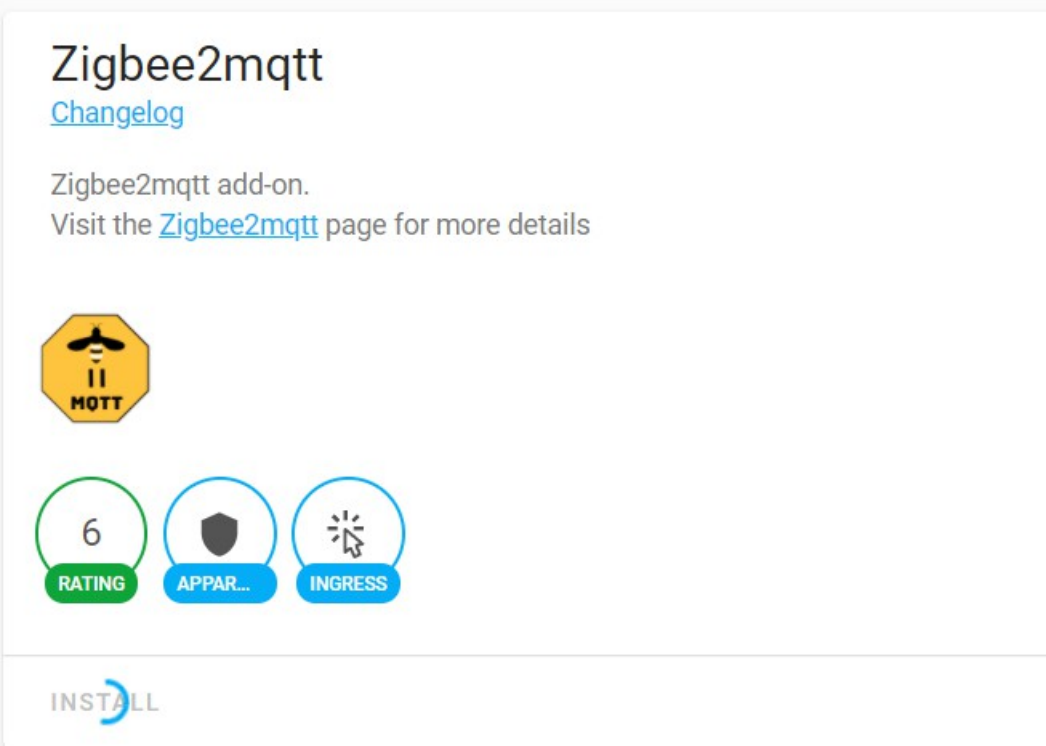

Рис 3.6 – встановлення додатка Zigbee2mqtt

Після встановлення додатку його необхідно налаштувати. Для цього треба зайти на сторінку керування додатком та змінити його файл конфігурації. Перед цим необхідно перевірити чи підключений приймач та у якому порту він знаходиться, для цього треба перейти на вкладку Superuser-System-Hardware.

Основні налаштування, які необхідно додати у файл configuration.yaml:

- data_path – розташування конфігураційного файлу;
- devices – підключення файлу devices.yaml;
- groups – підключення файлу groups.yaml;
- homeassistant – значення true, якщо встановлений у homeassistant;
- permit_join – якщо значення true, дозволити автоматичне підключення пристроїв;
- mqtt – містить декілька полів, base_topic, username, password;
- serial – вказання порту, до якого підключений приймач;
- advanced – містить поля log_level, pan_id, channel, network_key, availability_blocklist, availability_passlist;
- device_options – налаштування пристроїв;
- blocklist – список заблокованих пристроїв;
- passlist – список дозволених пристроїв;
- queue – черга;
- frontend – порт;
- socat – інструмент командної строки.

Загалом, файл конфігурації після налаштування виглядає наступним чином:

Лістинг 3.1 – Файл конфігурації Zigbee2mqtt

```
data_path: /config/zigbee2mqtt
external_converters: []
```

```
devices: devices.yaml
groups: groups.yaml
homeassistant: true
permit_join: true
mqtt:
  base_topic: zigbee2mqtt
  username: mqtt
  password: mqtt
serial:
  port: /dev/ttyACM0
advanced:
  log_level: warn
  pan_id: 6754
  channel: 11
  network_key:
    - 1
    - 3
    - 5
    - 7
    - 9
    - 11
    - 13
    - 15
    - 0
    - 2
    - 4
    - 6
    - 8
    - 10
    - 12
    - 13
  availability_blocklist: []
  availability_passlist: []
device_options: {}
blocklist: []
passlist: []
queue: {}
frontend:
  port: 8099
experimental: {}
socat:
  enabled: false
  master: pty,raw,echo=0,link=/tmp/ttyZ2M,mode=777
                                                slave:                    tcp-
listen:8485,keepalive,nodelay,reuseaddr,keepidle=1,keepintvl=1,keepcnt=5
  options: '-d -d'
  log: false
```

Після збереження конфігурації додаток необхідно перезапустити та перевірити вивід на присутність помилок.

Далі необхідно вказати автоматичне підключення пристроїв та налаштувати частоту оновлення показників сенсора, для цього додамо до файлу конфігурації наступні налаштування:

Лістинг 3.2 – Конфігурації підключень і таймеру Zigbee2mqtt

```
# Zigbee2mqtt
input_boolean:
  zigbee_permit_join:
    name: Allow devices to join
    initial: off
    icon: mdi:cellphone-wireless

timer:
  zigbee_permit_join:
    name: Time remaining
    duration: 100
```

3.5 MqttBroker

Для того щоб встановити додаток, треба зайти у вкладку Superuser та увійти у магазин додатків(ADD-ON-STORE) і обрати його зі списку. Після встановлення його необхідно запустити. Додаток MqttBroker має два файли конфігурації configuration.yaml та automations.yaml, які необхідно заповнити.

У файлі configuration.yaml необхідно записати IP-адреса серверу, порт, ім’я та пароль, які були вказані у конфігурації Zigbee2mqtt, а також налаштування пошуку пристроїв.

Лістинг 3.3 – Файл конфігурації configuration.yaml додатка MqttBroker

```
discovery:

mqtt:
  broker: 192.168.1.107
  port: 1883
  username: mqtt
  password: mqtt
  discovery: true
  discovery_prefix: homeassistant

sensor:
  - platform: mqtt
    name: Zigbee-Bridge state
    state_topic: "zigbee2mqtt/bridge/state"
    icon: mdi:router-wireless
```

Далі необхідно заповнити файл automations.yaml. До цих налаштувань

входять параметри та функції, які будуть автоматично виконуватися при запуску системи. Необхідно додати такі параметри як підключення ZigBee пристроїв, використання різних сервісів а також порядок обміну запитами між пристроями.

Лістинг 3.4 – Файл конфігурації automations.yaml додатка MqttBroker

```
- id: enable_zigbee_join
  alias: Enable Zigbee joining
  trigger:
    platform: state
    entity_id: input_boolean.zigbee_permit_join
    to: 'on'
  action:
  - service: mqtt.publish
    data:
      topic: zigbee2mqtt/bridge/config/permit_join
      payload: 'true'
  - service: timer.start
    data:
      entity_id: timer.zigbee_permit_join
- id: disable_zigbee_join
  alias: Disable Zigbee joining
  trigger:
  - entity_id: input_boolean.zigbee_permit_join
    platform: state
    to: 'off'
  action:
  - data:
      payload: 'false'
      topic: zigbee2mqtt/bridge/config/permit_join
    service: mqtt.publish
  - data:
      entity_id: timer.zigbee_permit_join
    service: timer.cancel
- id: disable_zigbee_join_timer
  alias: Disable Zigbee joining by timer
  trigger:
  - platform: event
    event_type: timer.finished
    event_data:
      entity_id: timer.zigbee_permit_join
  action:
  - service: mqtt.publish
    data:
      topic: zigbee2mqtt/bridge/config/permit_join
      payload: 'false'
  - service: input_boolean.turn_off
    data:
      entity_id: input_boolean.zigbee_permit_join
```

Після збереження налаштувань додаток необхідно перезапустити та переконатися що він працює з доданими конфігураціями без помилок.

3.6 Результати та тестування

Зібрана система відображена на рисунку 3.7.

Рисунок 3.7 – Зібрана система

Для початку необхідно перевірити роботу Cloud серверу, для цього необхідно в налаштуваннях підключень перейти на сторінку провайдера сервісу та перейти за посиланням. Для тестування зайдемо на сервер з телефону та з мобільної мережі, а не локальної, до якої підключений центральний пристрій.

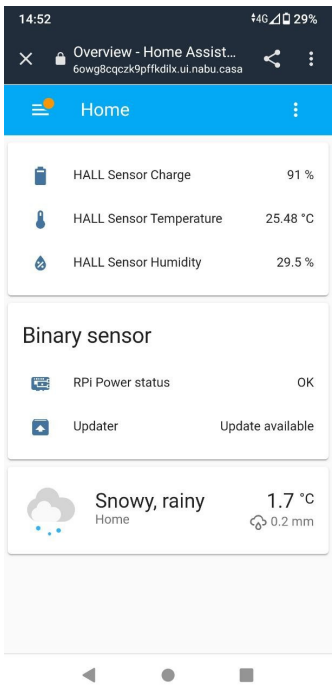

Рисунок 3.8 – Підключення до Cloud серверу з мобільного телефона та іншої мережі

Після успішного підключення буде завантажена основна сторінка Dashboard на якій буде відображена картка з підключеним сенсором. Відразу можна перейменувати назву об’єктів сенсора, дадаючи розташування сенсора або його призначення. Наприклад HALL Sensor Temperature, що означає що сенсор знаходиться у холі та відображує температуру у кімнаті. Загалом картка містить три об’єкти: рівень заряду батареї сенсора, температура та вологість (рисунок 3.9).

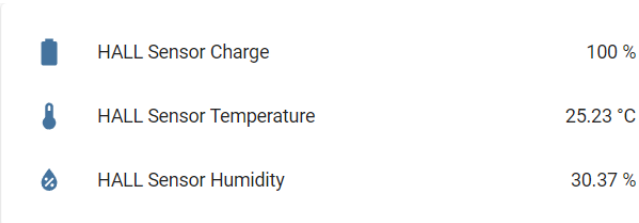

Рисунок 3.9 – Карточка з об’єктами датчика, розташованого в холі будинку

Якщо натиснути на один з об’єктів картки, наприклад показники температури, можна відкрити додаткове вікно, в якому буде відображений графік показників. У цьому вікні можна обрати проміжок часу, за який користувач хоче подивитися дані. Такі графіки можуть відображати будь-яку інформцію з будь якого сенсора з того моменту, коли він почав надсилати свої дані.

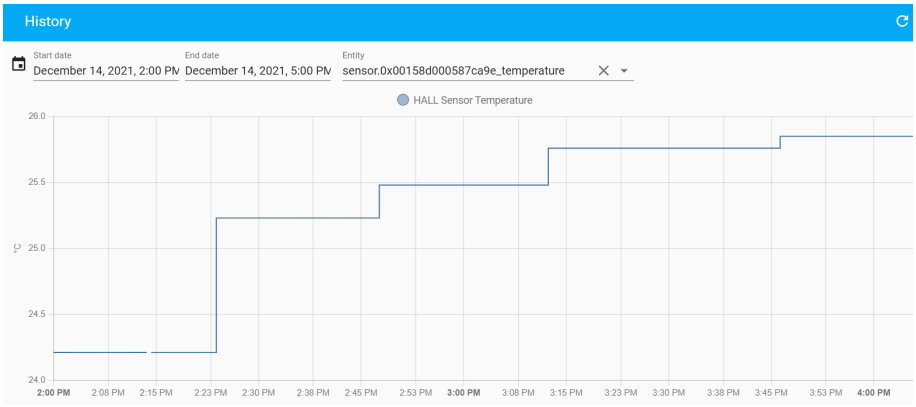

Рисунок 3.10 – Історія даних з сенсора, відображена на графіку

ВИСНОВКИ

Під час виконання кваліфікаційної роботи була розроблена система дистанційного моніторингу мікроклімату. Для вирішення поставленої задачі були розглянуті існуючі приклади, технології обміну інформацією та передачею даних між приладами у системі Інтернету речей, її основні вразливості, програмне забезпечення та cloud-сервіси для IoT систем. Завдяки обраним компонентам та технологіям система може відзначитися своєю дешивизною та простотою, при цьому за надійністю та функціоналом не буде відрізнятися від більш дорогих та складних систем.

Однією з переваг цієї системи є технологія Zigbee, прилади якої мають змогу працювати в автономному режимі дуже довго, без необхідності заміни

джерела живлення, до того ж на даний час на ринку існує дуже багато різноманітних датчиків та виконавчих механіхмів, які підтримують даний стандарт, а їх кількість зростає майже з кожним днем.

Використання Cloud-сервісів дає змогу зменшити витрати на компоненти для збереження та обробки інформації, а завдяки додатку користувача є змога цілодобово відстежувати температуру та вологість, переглядати попередні показники а також отримувати повідомлення при сильних змінах параметрів з будь-якої точки планети.

Завдяки існуючим програмним продуктам є змога встановлення віртуального оточення на операційну систему мікроконтролеру, що в свою чергу дозволяє повноцінно використовувати його функції. Дана система має кілька недоліків: при першому запуску їй потрібно близько 5 хвилин для автоматичного налаштування, за безпеку даних відповідальний лише постачальник хмарного сервісу, безкоштовний період користування хмарним сервісом – 1 місяць. Згодом дана система може бути доповнена іншими пристроями та перетворена у повноцінну систему розумного будинку або розвинена під конкрету область застосування.

ПЕРЕЛІК ПОСИЛАНЬ

1. Gershenfeld, N., Krikorian, R. and Cohen, D. (2004) The Internet of Things. Scientific American, 291, P 76-81 [Електронний ресурс] – Режим доступу: URL: http://dx.doi.org/10.1038/scientificamerican1004-76

2. Gigli, M. and Koo, S. (2011) Internet of Things, Services and Applications Categorization. Advances in Internet of Things, 1, 27-31. [Електронний ресурс] – Режим доступу: URL: http://dx.doi.org/10.4236/ait.2011.12004

3. Kosmatos, E.A., Tselikas, N.D. and Boucouvalas, A.C. (2011) Integrating RFIDs and Smart Ob-jects into a Unified Internet of Things Architecture. Advances in Internet of Things: Scientific Research, 1,P 5-12 [Електронний ресурс] – Режим доступу: URL: http://dx.doi.org/10.4236/ait.2011.11002.

4. Chen, X.-Y. and Jin, Z.-G. (2012) Research on Key Technology and Applications for the Internet of Things. Physics Procedia, 33, P 561-566. [Електронний ресурс] – Режим доступу: URL: http://dx.doi.org/10.1016/j.phpro.2012.05.104.

5. Suciu, G.; Vulpe, A.; Halunga, S.; Fratu, O.; Todoran, G.; Suciu, V. Smart cities built on resilient cloud computing and secure IoT. In Proceedings of the 2013 19th International Conference on Control Systems and Computer Science, IEEE, Bucharest, Romania, 29–31 May 2013; P. 513–518. [Електронний ресурс] – Режим доступу: URL: https://ieeexplore.ieee.org/abstract/document/6569312

6. What is the Cloud? How Does it Fit into the Internet of Things[Електронний ресурс] – Режим доступу: https://www.iotforall.com/what-is-the-cloud

7. Big-Data Analytics for Cloud, IoT and Cognitive Computing [Електронний ресурс] – Режим доступу: URL: https://books.google.com.ua/books?hl=uk&lr=&id=Kz1GDgAAQBAJ&oi=fnd&pg=PT10&dq=cloud+iot+computing&ots=b-

BjO0mzh-&sig=h_djuucYOQasqz4HOyVoxdkpDgk&redir_esc=y#v=onepage&q=cloud%20iot%20computing&f=false

8. M. Aazam, E.N. Huh, Fog computing and smart gateway based communication for cloud of things, in: Proceedings of the 2014 International Conference on Future Internet of Things Cloud, FiCloud 2014, Barcelona, Spain, 27–29 August 2014, P 464–470. [Електронний ресурс] – Режим доступу: URL: https://ieeexplore.ieee.org/abstract/document/6984239

9. Edge computing and IoT: How they fit together [Електронний ресурс] – Режим доступу: URL: https://enterprisersproject.com/article/2021/3/how-edge-computing-and-iot-fit-together

10. Iot under attack: Security is still not good enough on these edge devices. [Електронний ресурс] – Режим доступу: URL: https://www.zdnet.com/article/iot-under-attack-security-is-still-good-not-enough-on-these-edge-devices/

11. Основні типи мікроконтролерів та їх архітектура [Електронний ресурс] – Режим доступу: URL: https://ieeexplore.ieee.org/abstract/document/6984239

12. Raspberry Pi 3. Обзор и начало работы [Електронний ресурс] – Режим доступу: URL: https://dmitrysnotes.ru/raspberry-pi-3-obzor-i-nachalo-raboty

13. Raspberry Pi OS [Електронний ресурс] – Режим доступу: URL: https://www.raspberrypi.org/software/

14. WiFi [Електронний ресурс] – Режим доступу: URL: https://www.webopedia.com/definitions/wifi/

15. Zigbee [Електронний ресурс] – Режим доступу: URL: https://internetofthingsagenda.techtarget.com/definition/ZigBee

16. ZIGBEE WHAT IS IT, ZIGBEE VS BLUETOOTH [Електронний ресурс] – Режим доступу: URL: – https://www.ramelectronics.net/ZigBee.aspx

17. Протокол MQTT: концептуальное погружение [Електронний ресурс] – Режим доступу: URL: https://habr.com/ru/post/463669/