МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ СИСТЕМ ТА МЕРЕЖ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач випускової кафедри

_____ Ігор ЖУКОВ

«____»_____2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ "МАГІСТР"
ЗА СПЕЦІАЛЬНІСТЮ 123 "КОМП'ЮТЕРНА ІНЖЕНЕРІЯ"

Тема: Засоби моніторингу та управління елементами *IoT*

Виконавець: Антон ВОЛОШИН

Керівник: Станіслава КУДРЕНКО

Нормоконтролер: Василь МАЛЯРЧУК

Київ 2023

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет Комп'ютерних наук та технологій

Кафедра Комп'ютерних систем та мереж

Спеціальність 123 «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри КСМ

_____ Ігор ЖУКОВ

"_____"_____ 2023 р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи

Волошина Антона Вікторовича

(прізвище, ім'я, по батькові)

1. Тема роботи: "Засоби моніторингу та управління елементами *IoT*", затверджена наказом ректора від «29» серпня 2023 р. № 1521/ст.
2. Термін виконання роботи: з 02.10.2023 до 31.12.2023
3. Вихідні дані до роботи: розробити систему моніторинку та управління Інтернету речей, яка включає в себе зчитування даних з модулів *ESP8266-12 ESP8266-12E* та їх подальше виведення у канал *ThingSpeak.*
4. Зміст пояснювальної записки: титульний аркуш, завдання, реферат, зміст, перелік умовних скорочень, вступ, перший розділ, висновки до першого розділу, другий розділ, висновки до другого розділу, третій розділ, висновки до третього розділу, загальні висновки до роботи, список джерел, додаток.
5. Перелік обов'язкового графічного (ілюстративного) матеріалу: Матеріали дипломного проекту мають бути представлені у вигляді презентацій *Power Point* у форматі *.ppt* та універсальному форматі *pdf.*

6. Календарний план-графік

№ п/п	Етапи виконання кваліфікаційної роботи	Термін виконання Етапів	Примітка
1	Ознайомитись з постановкою задачі дипломної роботи	02.10.23 – 04.10.23	
2	Вивчити спеціальну літературу і технічну документацію	04.10.23 – 06.10.23	
3	Проаналізувати системи моніторингу елементів IoT	07.10.23 – 09.10.23	
4	Написати розділ 1.	11.10.23 – 21.10.23	
5	Проаналізувати принципи аналізу активності користувачів і стану обладнання	23.10.23 – 30.10.23	
6	Написати розділ 2.	31.10.23 – 07.11.23	
7	Провести опис роботи розробленої системи	07.11.23 – 18.11.23	
8	Написати розділ 3.	20.11.23 – 30.11.23	
9	Оформити пояснювальну записку та пройти нормоконтроль	09.12.23 – 18.12.23	
10	Підготувати презентаційний матеріал та захистити дипломну роботу	20.12.23 – 31.12.23	

7. Дата видачі завдання: “02” жовтня 2023 р.

Керівник кваліфікаційної роботи _____ Кудренко С.О.
(підпис керівника)

Завдання прийняв до виконання _____ Волошин А.В.
(підпис випускника)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи «Засоби моніторингу та управління елементами IoT», 90 сторінок, 44 рисунків, 11 літературних джерел.

МОНІТОРИНГ, УПРАВЛІННЯ, БЕЗПЕКА ІНФОРМАЦІЇ, ЗАГРОЗИ КОНФІДЕНЦІЙНОСТІ, КОНТРОЛЬ, ОБМІН ІНФОРМАЦІЄЮ.

Мета роботи: здійснити системний аналіз вибраного об’єкту комп’ютерної інженерії на предмет його вдосконалення, використавши як наявний дослідницький інструментарій організації – бази практики, так і дослідницький інструментарій.

Головна ціль роботи: проаналізувати та дослідити засоби моніторингу та управління елементами IoT на функціональність та ефективність. Провести порівняльну характеристику та визначити найбільш ефективні засоби моніторингу та управління елементами IoT .

Об'єкт дослідження: Засоби моніторингу та управління елементами IoT.

Предмет дослідження: Характеристика засобів моніторингу та управління елементами IoT.

Практична значущість дослідження: написання готового звіту з детальним аналізом предмету, його проблематики та ймовірності його вдосконалення шляхом дослідницького інструментарію.

Основні показники та результати дослідження: Засоби моніторингу та управління елементами Інтернету речей, розбір актуальних проблем та сучасного їх вирішення.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ЗНАКІВ, СКОРОЧЕНЬ, ТЕРМІНІВ

IoT – Internet of Things

AI - Artificial intelligence

ШІ - Штучний інтелект

MITM - Man-in-the-Middle

NFC - Near field communication

MQTT - message queuing telemetry transport

HTTP - HyperText Transfer Protocol

ML - Machine learning

ASW - Amazon Web Services

API - Application Programming Interface

SDK - software development kit

AMQP - Advanced Message Queuing Protocol

IDE - Integrated development environment

АЦП - Аналого-цифровий перетворювач

GDPR – General Data Protection Regulation

ЄС - Європе́йський Сою́з

BI - Enterprise business intelligence

IEEE - Institute of Electrical and Electronic Engineers

IP - Internet Protocol

ADR - Adaptive Data Rate

LTE - Long-Term Evolution

QoS - Quality of service

URL - Uniform Resource Locator

ВСТУП

У сучасному світі, де технологічний прогрес стрімко розвивається, концепція Інтернету речей (*IoT*) набула величезного значення та стала основною силою в еволюції технологій. *IoT* відкриває безмежні можливості для підвищення ефективності, зручності та автоматизації управління різними аспектами нашого повсякденного життя. Ця концепція дозволяє об'єднувати фізичні пристрої, датчики та об'єкти навколишнього середовища в єдину мережу, створюючи таким чином "розумне" оточення, в якому пристрої можуть спілкуватися, обмінюватися даними та взаємодіяти.

Актуальність Інтернету речей проявляється в тому, як ця технологія змінює наше повсякденне життя та виробничі процеси. Від "розумного" дому, де пристрої автоматизують наше життя, до індустріальних підприємств, де *IoT* оптимізує виробничі процеси, ця концепція стає невід'ємною частиною нашої реальності. Значення Інтернету речей полягає в ефективному використанні ресурсів, покращенні якості життя та створенні нових можливостей для розвитку бізнесу. Впровадження *IoT* дозволяє збирати, обробляти та використовувати величезний обсяг даних для прийняття обґрунтованих рішень.

IoT революціонізує такі сфери, як медицина, транспорт, енергетика та сільське господарство. У медицині він допомагає в реальному часі відстежувати стан пацієнтів, у транспорті оптимізує рух автотранспорту, а в енергетиці забезпечує ефективне використання ресурсів. Зростання кількості підключених пристроїв свідчить про те, що *IoT* - це не просто технологія, а цілісна екосистема, яка формує новий спосіб взаємодії з технікою та середовищем. Забезпечення безпеки та приватності в цьому контексті стає важливим завданням, оскільки зростає кількість обміну важливою інформацією між пристроями. Інтернет речей визначає майбутнє, де взаємодія між об'єктами навколишнього світу стає безперервною та невидимою, а *IoT* сам стає невід'ємною частиною нашого щоденного існування. Важливо розуміти та впроваджувати ці технології з урахуванням їх впливу на наше суспільство та

інфраструктуру, забезпечуючи збалансований розвиток цього інноваційного напрямку.

Виникнення Інтернету речей (*IoT*) представляє собою важливий етап у технологічному розвитку, що визначає новий спосіб сприйняття та взаємодії з технікою. Історія *IoT* має свої корені в прагненні зробити фізичні об'єкти більш "розумними" та взаємодіючими, щоб полегшити та поліпшити наше щоденне життя. Зародження Інтернету речей пов'язане з розвитком мініатюрних сенсорів, бездротових технологій та зростанням обчислювальної потужності пристроїв. Початково використанням *IoT* була автоматизація промислових процесів, але з часом ця концепція перейшла в повсякденний простір, трансформуючи наше оточення в повністю підключений екосистему.

Роль Інтернету речей полягає в тому, щоб забезпечити інтеграцію та взаємодію між фізичними об'єктами та цифровим середовищем. Ця технологія стала каталізатором для змін у багатьох сферах, від приватного життя до великих промислових систем. *IoT* дозволяє збирати величезні обсяги даних з різних джерел, роблячи їх доступними для аналізу та прийняття рішень. У сфері медицини IoT допомагає в реальному часі моніторити стан пацієнтів, у сфері транспорту оптимізує рух транспорту, а в енергетиці забезпечує ефективне використання енергоресурсів.

Розвиток Інтернету речей відбувається в умовах постійної зміни та вдосконалення технологій. Роль *IoT* у формуванні майбутнього полягає не лише в його існуючих можливостях, а й у потенційних перспективах для розвитку сучасного суспільства та економіки. *IoT* стає інтегральною частиною нашого повсякденного життя, забезпечуючи нам нові можливості та підвищуючи рівень комфорту. Розуміння ролі *IoT* у визначенні нашого технологічного майбутнього дозволяє належним чином використовувати цей інноваційний потенціал для досягнення покращень у різних галузях та формування нового етапу технологічного розвитку.

Безпека в Інтернеті речей (*IoT*) стає невід'ємною складовою його впровадження в різні сфери життя та промисловості. З тим як *IoT* входить в наше повсякденне життя, питання безпеки стає ще більш актуальним та складним. Однією з ключових аспектів безпеки є захист особистої інформації. Оскільки *IoT* використовується для збору та

обробки великої кількості даних, важливо забезпечити їхню конфіденційність та недоступність для несанкціонованого доступу. Заходи шифрування, захист від зловмисного програмного забезпечення та ефективна аутентифікація стають ключовими аспектами в забезпеченні безпеки особистої інформації користувачів.

Іншим важливим аспектом є захист від кібератак. З зростанням кількості підключених пристроїв збільшується ймовірність кіберзлочинців використовувати їх для здійснення атак на системи та мережі. Безпека в *IoT* повинна враховувати можливість виявлення та запобігання таким атакам, забезпечуючи стійкість та відновлюваність систем. У сфері промисловості та критичних інфраструктур, де *IoT* також використовується широко, важливо забезпечити надійність та безпеку пристроїв. Вразливості в пристроях, які використовуються в енергетичних системах, транспорті чи виробництві, можуть мати серйозні наслідки.

Розуміння та реалізація високих стандартів безпеки в *IoT* стає важливою умовою для його успішного впровадження. Тільки з врахуванням цих аспектів можна забезпечити розвиток Інтернету речей як технології, яка дійсно приносить користь суспільству та не становить загрози для його безпеки.

Виклики та перспективи Інтернету речей (*IoT*) визначають його динамічний шлях розвитку та впровадження в різні сфери сучасного життя. З одного боку, *IoT* відкриває необмежені можливості для оптимізації процесів, автоматизації, збільшення продуктивності та покращення якості життя. З іншого боку, існують численні виклики, які потрібно подолати для повноцінного впровадження та розвитку *IoT*. Одним із важливих викликів є питання стандартизації та сумісності. У зв'язку з тим, що ринок *IoT* насичується різноманітними пристроями, протоколами та платформами, стає важливим встановлення загальноприйнятих стандартів, які дозволять різним пристроям спілкуватися та взаємодіяти ефективно.

Ще однією суттєвою трудністю є питання приватності та безпеки даних. Збільшення кількості підключених пристроїв призводить до збільшення обсягу збираємих та оброблюваних даних. Важливо розробляти та впроваджувати механізми захисту даних, щоб уникнути несанкціонованого доступу та зловживань. Іншою проблемою є питання енергоефективності та продовження терміну служби пристроїв

IoT. Багато з них використовуються в умовах, де зміна акумулятора або зарядка може бути проблематичною. Вирішення цього виклику вимагає розробки енергоефективних технологій та оптимізації використання енергії.

Незважаючи на ці виклики, перспективи *IoT* залишаються вельми захоплюючими. Зростання кількості підключених пристроїв, розвиток хмарових технологій та штучного інтелекту надає нові можливості для створення інноваційних продуктів та послуг. Інтернет речей має потенціал перетворити наше оточення, забезпечуючи зв'язок між фізичним та цифровим світом та відкриваючи нові горизонти для розвитку технологій у майбутньому.

Сучасний ландшафт Інтернету речей (*IoT*) визначається різноманітністю платформ, які надають інфраструктуру для підключення та управління підключеними пристроями. Ці платформи відіграють ключову роль у впровадженні та розвитку *IoT*, надаючи інструменти для збору, обробки та аналізу даних. Прості та безкоштовні платформи стають важливим елементом цього екосистеми, особливо для початківців та тих, хто має обмежений бюджет. Такі платформи забезпечують доступність технологій *IoT*, дозволяючи ентузіастам та малим компаніям швидше розпочати свої проекти.

Важливість простих та безкоштовних платформ *IoT* полягає в тому, що вони знижують бар'єри для входження в цей області. Вони часто мають інтуїтивний інтерфейс, що полегшує налаштування та використання. Також, завдяки їх безкоштовній або демократичній вартості, вони дозволяють експериментувати та навчатися, не стикаючись з великими витратами. Ці платформи часто мають широкий набір функцій, таких як візуалізація даних, аналітика та засоби безпеки. Такий комплекс можливостей спрощує розгортання та використання *IoT*-проектів.

Проте важливо також розуміти, що простота та безкоштовність можуть мати свої обмеження. У випадку більших та бізнес-орієнтованих проектів може знадобитися більш потужна та гнучка платформа, яка пропонує розширені можливості та підтримку для складних випадків використання.

Отже, прості та безкоштовні платформи *IoT* відіграють роль важливого каталізатора для широкого впровадження Інтернету речей, дозволяючи більшій кількості людей та компаній залучитися до цього захоплюючого інноваційного світу.

Інтернет речей перетворює світ навколо нас, надаючи безліч можливостей для покращення ефективності та якості життя. Зрозуміння та використання *IoT* стає ключовими для компаній, урядів та суспільства в цілому. Інтернет речей розширює горизонти інновацій та впливає на всі сфери нашого життя, від дому та офісу до медицини та промисловості. Однак з великими можливостями *IoT* приходять і великі відповідальності, такі як забезпечення безпеки та приватності даних. Зміцнення цих аспектів стане ключовим фактором для сталого та успішного розвитку Інтернету речей в майбутньому.

РОЗДІЛ 1
КОНЦЕПЦІЯ ТА ПРИНЦИПИ ІНТЕРНЕТУ РЕЧЕЙ

1.1. Технологія "Інтернет речей" та його елементи

1.1.1. Визначення поняття “Інтернет речей”

Інтернет речей (*IoT*) - це концепція, що поєднує фізичні пристрої (речі) з мережевими технологіями та забезпечує їх спроможність збирати та обмінюватися даними. В основі *IoT* лежить ідея підключення різноманітних об'єктів - від побутових пристроїв (наприклад, побутової техніки та електроніки) до промислового обладнання, транспорту, інфраструктури та інших об'єктів - до глобальної мережі Інтернет.

Ці підключені речі мають можливість взаємодіяти між собою та з людьми, передавати дані, отримувати команди і виконувати різноманітні функції. Вони здатні сприяти автоматизації, моніторингу, контролю, оптимізації процесів та покращенню різних аспектів життя.

IoT базується на використанні датчиків, актуаторів, мережевих з'єднань та програмного забезпечення, що дозволяє збирати, обробляти та аналізувати великі обсяги даних.

Ця технологія розширює можливості збору інформації, управління процесами та прийняття рішень, що впливають на ефективність, безпеку та комфорт у різних сферах, включаючи побутове використання, промисловість, транспорт, охорону здоров'я, сільське господарство та багато інших.

Завдяки Інтернету речей створюються інтелектуальні середовища, де реальний світ зливається з цифровим простором, що відкриває безліч нових можливостей для оптимізації процесів, покращення якості життя та розвитку нових інноваційних продуктів та послуг.

1.1.2. Елементи *IoT*

Інтернет речей складається з різних елементів, які взаємодіють між собою для забезпечення підключеності, обміну даними та управління пристроями та системами. Такими елементами є прискорювачі, що призначені для збору різних типів даних з навколишнього середовища. Прискорювачі можуть вимірювати рух, температуру, вологості, освітленість та інші параметри. Завдяки сенсорам, *IoT* може отримувати реальний час інформації про стан об'єктів та середовища.

Засоби комунікації, які дозволяють підключеним пристроям обмінюватися даними та інформацією. Це можуть бути бездротові технології, такі як *Wi-Fi, Bluetooth, Zigbee, NFC*, або провідні мережі, такі як *Ethernet*. Засоби комунікації забезпечують зв'язок між різними пристроями, включаючи сенсори, контролери та хмарні платформи.

Хмарні платформи, де збираються, зберігаються та обробляються дані з підключених пристроїв. Вони надають розширені функції аналітики, обробки даних, забезпечення безпеки та керування. Хмарні платформи дозволяють використовувати високопродуктивні обчислювальні ресурси та масштабувати систему *IoT*.

IoT використовує аналітику та штучний інтелект (ШІ) для обробки та аналізу великого обсягу даних, зібраних з підключених пристроїв. ШІ дозволяє виявляти патерни, робити передбачення та забезпечувати автоматизоване управління системами *IoT*. Аналітика та ШІ допомагають здійснювати розумну обробку даних та використовувати їх для прийняття рішень [1].

Програмне забезпечення використовується для управління підключеними пристроями та системами *IoT*. Це може бути спеціалізоване програмне забезпечення для збору даних, керування, аналітики та взаємодії з користувачами. Програмне забезпечення грає важливу роль у забезпеченні функціональності та ефективності системи *IoT*.

Для відстеження стану оточуючого середовища об’єкти оснащуються датчиками, що забезпечують збирання та опрацювання даних у реальному часі.

Датчик вимірює фізичні параметри контрольованого об’єкта і перетворює їх у цифровий сигнал, який може бути переданий і прийнятий відповідним пристроєм.

Для опрацювання та накопичення даних з датчиків використовують мікрочипи, до яких датчики приєднано через контролери (рис. 1.1).

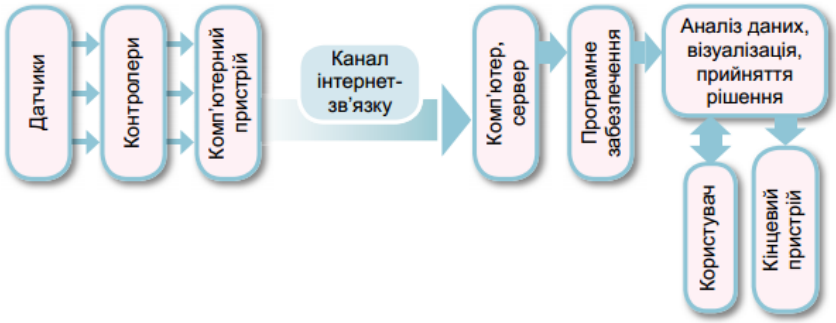

Рис 1.1. Схема функціонування мережі Інтернет речей

1.1.3. Принципи *IoT*

Інтернет речей (*IoT*) базується на кількох загальних принципах та має свої особливості, які визначають його функціональність і потенціал. Ось опис загальних принципів та особливостей *IoT*:

1. *IoT* передбачає підключення різних фізичних пристроїв до мережі Інтернет. Це можуть бути датчики, пристрої збору даних, контролери, актуатори тощо. Ця підключеність дозволяє обмінюватися даними та командами між різними пристроями та системами.

2. *IoT* використовує різні типи сенсорів та датчиків, які забезпечують збір реальних даних з навколишнього середовища. Ці дані можуть стосуватись температури, вологості, руху, освітленості та багатьох інших параметрів. Сенсори дозволяють отримувати актуальну інформацію про стан об'єктів та навколишнього середовища.

3. *IoT* забезпечує збір великого обсягу даних з підключених пристроїв та сенсорів. Ці дані можуть бути зібрані та оброблені в режимі реального часу, або зберігатись для подальшого аналізу. Аналіз даних в *IoT* дозволяє отримати корисну інформацію, виявити закономірності та тренди, та здійснити прийняття рішень на основі цих даних.

4. *IoT* забезпечує взаємодію між підключеними пристроями, системами та людьми. Це може включати передачу даних, команд та повідомлень між пристроями, а також інтерфейси для взаємодії з користувачами. Комунікація може відбуватись

через різні протоколи та мережеві технології, такі як *Wi-Fi, Bluetooth, Zigbee, LoRaWAN* тощо.

5. *IoT* дозволяє використовувати аналітичні методи та інтелектуальні алгоритми для обробки та аналізу зібраних даних. Це може включати машинне навчання, штучний інтелект, аналіз великих даних (*Big Data*) тощо. Інтелектуальність допомагає виявляти патерни, здійснювати передбачення, приймати автоматизовані рішення та оптимізувати процеси.

6. *IoT* вимагає особливої уваги до питань безпеки, оскільки підключені пристрої та дані можуть бути підвержені ризикам кібератак та несанкціонованому доступу. Забезпечення безпеки включає захист з'єднань, шифрування даних, аутентифікацію пристроїв та впровадження механізмів контролю доступу.

Ці загальні принципи та особливості *IoT* визначають основу для розробки і впровадження підключених пристроїв та систем, що сприяють автоматизації, оптимізації та покращенню різних аспектів нашого життя та діяльності.

1.2. Історія розвитку технології *IoT*

Інтернет речей (*IoT*) має свої корені в далекому минулому, але термін "Інтернет речей" став широко відомим у 1999 році завдяки Кевіну Ештону, співзасновнику *Auto-ID Center*, під час презентації проекту автоматизації обігу товарів у сфері логістики.

Основна ідея полягала у тому, щоб об'єднати фізичний світ речей зі світом Інтернету, надаючи речам здатність збирати, обробляти і обмінювати даними через Інтернет.

У 2000-х роках *IoT* почав здобувати популярність та став предметом інтенсивних досліджень і розробок.

Великі технологічні компанії, такі як *IBM*, *Intel*, *Cisco*, розпочали активно працювати над розробкою та впровадженням *IoT*-систем. Перші застосування *IoT* були пов'язані зі збіром даних в промисловості, логістиці та енергетиці.

Прорив у розвитку *IoT* відбувся в середині 2010-х років, коли було досягнуто значного зростання швидкості передачі даних, зниження вартості сенсорів та пристроїв зв'язку, а також розвиток хмарних технологій. Ці фактори забезпечили сприятливі умови для розширення *IoT* та введення його в побутове використання.

Сфери застосування *IoT* значно розширилися. Він знайшов застосування у смарт-будинках, де розумні пристрої дозволяють контролювати освітлення, опалення, безпеку та інші системи з віддаленої точки.

У смарт-містах *IoT* використовується для управління транспортною системою, збору даних про якість повітря, утилізацію відходів та багато іншого [2].

Промисловість також впроваджує *IoT* для вдосконалення моніторингу та управління виробництвом.

Впровадження технології "промислової Інтернету речей" (*IoT*) допомагає підвищити ефективність, знизити витрати та покращити безпеку на підприємствах.

Застосування *IoT* також поширилося на галузі охорони здоров'я, сільське господарство, роздрібну торгівлю та багато інших сфер. *IoT* є основою для розвитку концепції "розумних міст" та "розумних фабрик", де використання сучасних технологій допомагає зробити життя людей більш комфортним, ефективним та безпечним.

Прогнози розвитку *IoT* є дуже оптимістичними. За даними дослідників, кількість підключених до Інтернету речей може досягти декількох мільярдів до 2025 року. *IoT* продовжує рухатися вперед, вдосконалюючи світ навколо нас і роблячи його більш підключеним та інтелектуальним.

1.3. Технології та стандарти *IoT*

Бездротові мережі зв'язку грають важливу роль у розвитку технології Інтернету речей (*IoT*). Вони дозволяють підключати мільйони пристроїв до Інтернету без необхідності проводового підключення. Оскільки *IoT* базується на зборі та обміні данних між пристроями, бездротові мережі зв'язку є ключовим фактором успіху цієї технології.

1.3.1. Бездротові мережі зв'язку для *IoT*

Для *IoT* використовуються різні бездротові технології, залежно від вимог до зв'язку, дальності передачі даних, енергоефективності та інших факторів. Основні технології бездротового зв'язку для *IoT* включають:

1. *Wi-Fi* є однією з найпоширеніших технологій бездротового зв'язку, яка забезпечує широкосмуговий доступ до Інтернету для пристроїв у межах певної зони покриття. Вона дозволяє швидкий обмін даними, що робить її привабливою для домашнього та офісного використання *IoT* пристроїв (рис. 1.2).

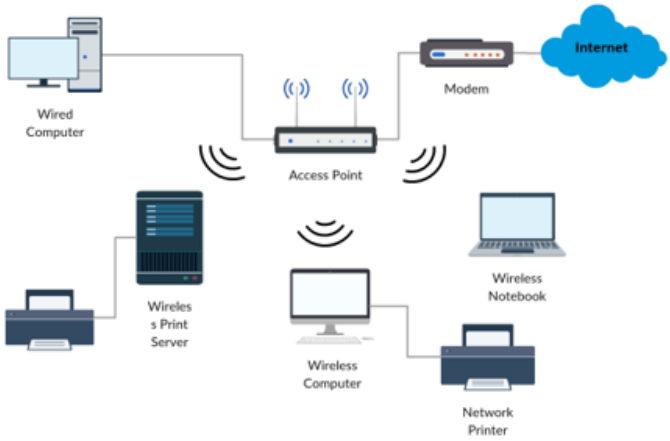

Рис. 1.2. Схема принципу роботи *Wi-Fi*

Це високошвидкісне підключення до Інтернету та підключення до мережі без використання будь-яких кабелів чи проводів. Бездротова мережа керує трьома основними елементами: радіосигналами, антеною та маршрутизатором. Радіохвилі - це ключі, які роблять можливим мережу *Wi-Fi*. Карти *Wi-Fi* є приймачем радіосигналів. Комп’ютери та мобільні телефони вже мають карти *Wi-Fi*. Радіосигнали передаються з антен і маршрутизаторів, які сприймаються приймачами *Wi-Fi*, як-от комп’ютери та стільникові телефони з картами *Wi-Fi*. Щоразу, коли комп’ютер отримує сигнали в діапазоні 100-150 футів для маршрутизатора, він негайно підключає пристрій.

Діапазон *Wi-Fi* залежить від навколишнього середовища, у приміщенні чи на вулиці. Карти *Wi-Fi* зчитують сигнали, і в результаті створюється інтернет-з’єднання між користувачем і мережею. Швидкість пристрою, який використовує з’єднання *Wi-*

Fi, збільшується, коли комп’ютер наближається до основного джерела, і швидкість зменшується, коли комп’ютер віддаляється.

Багато нових ноутбуків, мобільних телефонів мають вбудовану карту *Wi-Fi*, тому вам не потрібно нічого робити, що є одним із найкращих. Якщо це безкоштовний тип підключення до мережі, користувач отримає ідентифікатор користувача та пароль. Безкоштовне підключення до базової мережі також добре в деяких областях. Підключення до мережі *Wi-Fi* створює гарячі точки в містах. Гарячі точки — це точки підключення мережі *Wi-Fi*. Це невелика коробка, яка підключена до Інтернету. У громадських місцях, таких як ресторани, аеропорти, офіси готелів, університети тощо, є багато точок доступу *Wi-Fi.*

2. *Bluetooth* є технологією короткодіючого бездротового зв'язку, яка використовується для з'єднання пристроїв на невеликій відстані. Вона зручна для сполучення мобільних пристроїв, сенсорів та актуаторів у невеликих *IoT*-проектах (рис. 1.3).

Пристрої *Bluetooth Smart* — це пристрої на основі технології *Bluetooth Low Energy*, які можуть з'єднуватися з пристроями *Bluetooth Smart Ready*.

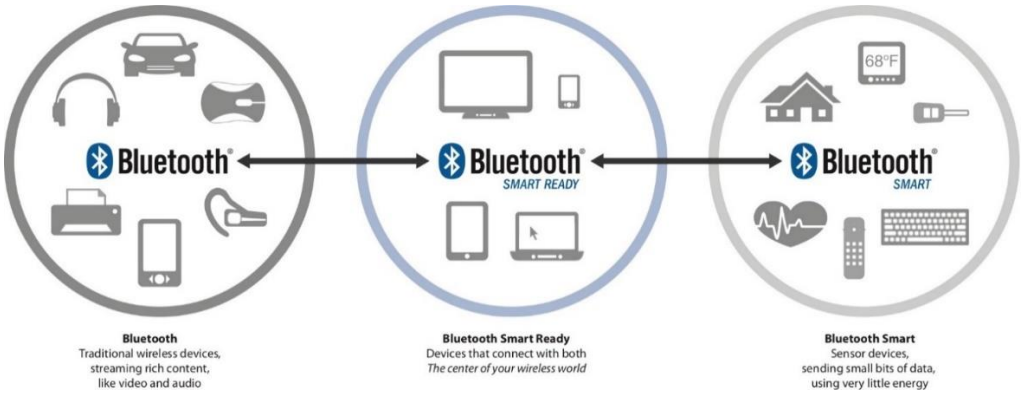

Рис. 1.3. Схема взаємозв'язку між пристроями *Bluetooth Smart* та *Bluetooth Smart Ready*

Для зручності споживача *Bluetooth*-пристрою, що підтримують технологію *Bluetooth Low Energy*, мають стікери *Bluetooth Smart ma Bluetooth Smart Ready*. Ці стікери необов'язково прямо відповідають технічним категоріям, які будуть описані нижче, але вони уособлюють визначення цих технологій, зручне для сприйняття споживача.

Пристрої *Bluetooth Smart Ready* — це «вузлові» пристрої, наприклад комп'ютери, планшети, мобільні телефони та подібні продукти, які підтримують стандарт *Bluetooth Smart (Bluetooth Low Energy)* і дозволяють надалі додавати додаткові профілі за рахунок завантаження програм або драйверів, або іншими методами [3].

Пристрої *Bluetooth Smart* — це пристрої на основі технології *Bluetooth Low Energy*, які можуть з'єднуватися з пристроями *Bluetooth Smart Ready*. Слід зазначити одну важливу річ: пристрої на основі *Bluetooth Smart* за визначенням не можуть з'єднуватися з класичними *Bluetooth*-пристроями, а тільки з пристроями *Bluetooth Smart Ready*. Такий підхід спрямований на те, щоб споживач при покупці пристрою Bluetooth Smart знав, що він працюватиме з іншим пристроєм із логотипом *Bluetooth Smart Ready*, але не з пристроями із звичайним логотипом *Bluetooth*.

Zigbee - це бездротовий стандарт для мереж з низьким споживанням енергії, який забезпечує довгий термін служби батарей у малопотужних пристроях. Він часто використовується в "розумних будинках" для з'єднання різних пристроїв, таких як термостати, датчики руху та освітлення (рис. 1.4).

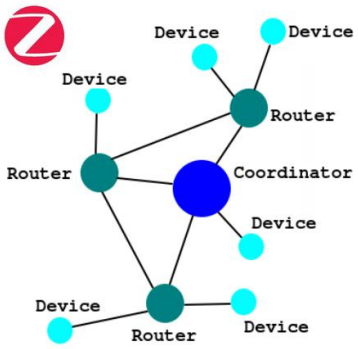

Рис. 1.4. Схема *Zigbee*

Протокол *ZigBee* 3.0 призначений для передачі даних через радіочастотне середовище з високим рівнем завад, яке є поширеним у комерційних і промислових додатках. Версія 3.0 базується на наявному стандарті зв’язку *ZigBee*, але уніфікує профілі додатків для певного ринку, щоб дозволити всім пристроям бути бездротовим підключенням до однієї мережі, незалежно від їх ринкового призначення та функції. Крім того, схема сертифікації *ZigBee 3.0* забезпечує взаємодію продуктів від різних виробників пристроїв. Підключення мереж *ZigBee 3* до *IP*-домену відкриває

можливість бездротового моніторингу та керування за допомогою радіопристроїв, таких як смартфони та планшети в *LAN* або *WAN*, включаючи Інтернет, і втілює справжній Інтернет речей.

Переваги використання протоколу *ZigBee*:

1) Чудово підходить для низького енергоспоживання.

2) Надає мережеву безпеку та послуги підтримки для програм на основі *IEEE*.

3) *ZigBee* дозволяє повністю об'єднати дім користувача у мережу, де всі пристрої можуть обмінюватися даними.

4) Підходить для використання у розумному будинку.

5) Низька вартість, а саме мікросхеми та модулі *ZigBee* недорогі порівнянно з аналогами, що робить їх економічно вигідним вибором для програм *IoT*.

ZigBee використовує топологію мережі mesh, яка дозволяє пристроям обмінюватися даними один з одним без необхідності центрального координатора чи маршрутизатора. Це робить його ідеальним для використання в додатках розумного дому, де пристрої повинні спілкуватися один з одним і з центральним центром управління. протокол *ZigBee* розроблено з надійними механізмами, які гарантують надійну доставку даних навіть у несприятливих умовах

Недоліки використання протоколу *ZigBee*:

1. *ZigBee* має короткий радіус дії порівняно з іншими протоколами бездротового зв'язку, що у свою чергу впливає на його придатність для певних типів програм або для використання у великих будівлях.

2. *ZigBee* розроблено для додатків із низькою швидкістю передачі даних, тому додатки, які потребують високої швидкості передачі даних, можуть бути не сумісні з *ZigBee*.

3. Функції безпеки *ZigBee* не такого високого рівня як інші протоколи Інтернету речей, що робить його більш вразливим до хакерських атак та інших загроз безпеці.

LoRaWAN (Long Range Wide Area Network) - це технологія довгого діапазону передачі даних, яка дозволяє створювати великі мережі *IoT* пристроїв на великій відстані від базової станції. Вона підходить для використання в промислових та

міських умовах, де потрібно охопити великі площі з низькою споживанням енергії (рис.1.5).

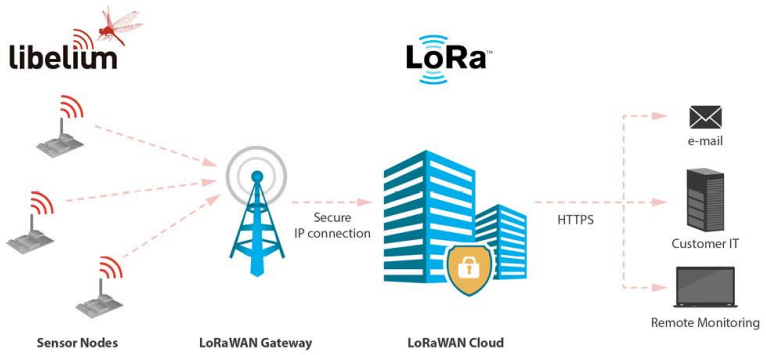

Рис. 1.5. Схема мережі *LoRaWAN*

Цей стандарт забезпечить безперебійну взаємодію між розумними речами без необхідності складних локальних інсталяцій і повертає свободу користувачам, розробникам, підприємствам, надаючи роль поза Інтернетом речей.

Мережна архітектура *LoRaWAN* зазвичай побудована за топологією «зірка зірок», у якій шлюзи є прозорим мостом, що передає повідомлення між кінцевими пристроями та центральним мережевим сервером на сервері. Шлюзи підключаються до мережевого сервера через стандартні *IP*-з’єднання, тоді як кінцеві пристрої використовують бездротовий зв’язок з одним або кількома шлюзами.

Зв’язок між кінцевими пристроями та шлюзами розподіляється на різних частотних каналах і швидкостях передачі даних. Вибір швидкості передачі даних є компромісом між діапазоном зв’язку та тривалістю повідомлення. Завдяки технології розширеного спектру зв'язок з різними швидкостями передачі даних не заважає один одному і створює набір «віртуальних» каналів, збільшуючи пропускну здатність шлюзу. Щоб максимізувати термін служби батареї кінцевих пристроїв і загальну ємність мережі, мережевий сервер *LoRaWAN* керує швидкістю передачі даних і радіочастотним виходом для кожного кінцевого пристрою окремо за допомогою схеми адаптивної швидкості передачі даних (*ADR*).

Національні мережі, націлені на Інтернет речей, такі як критична інфраструктура, конфіденційні особисті дані або критичні функції для суспільства, мають особливу потребу в безпечному зв’язку. Це було вирішено за допомогою кількох рівнів шифрування.

NB-IoT (*Narrowband Internet of Things*) та *LTE-M* - Це технології спеціально розроблені для IoT та мають підтримку мобільних операторів. Вони забезпечують довгий термін служби батарей, гарантуючи надійне підключення вдалеких та важкодоступних місцях (рис. 1.6).

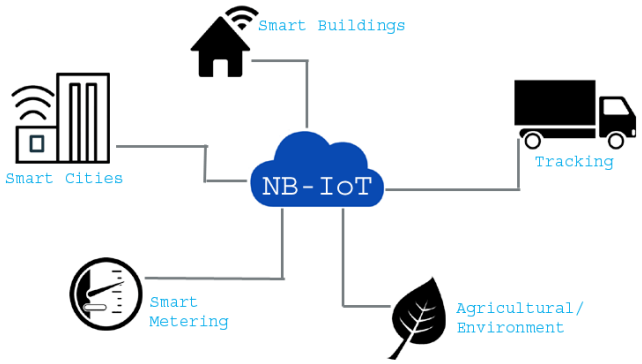

Рис. 1.6. Схема структури *NB-IoT*

Це мережа *LPWA* (*Low Power Wide Area*), відома як вузька смуга, спеціально розроблена для пристроїв *IoT*, які працюватимуть на низьких частотах і споживатимуть менше енергії.

NarrowBand-IoT - це мережева технологія, розроблена компанією *3GPP*, яка співпрацює з такими великими телекомунікаційними гігантами, як Huawei, *Qualcomm, Ericsson* і *Vodafone*.

Усі пристрої *IoT*, які вже розроблені або будуть розроблені, повинні працювати 24/7, щоб надсилати дані та підтримувати користувача в курсі. Ці довгі години роботи означають споживання більше енергії. Крім того, споживаючи більше енергії, дані повинні надсилатися на високій смузі пропускання для мільярдів підключених пристроїв. Все це призведе до використання великої смуги пропускання, меншого терміну служби батареї та вищої ціни з поточними мережами *LTE*

Таким чином, щоб підключити всі мільярди пристроїв і досягти більшої надійності та якості обслуговування, *NarrowBand* з'явився на світ, що призводить до переваг *NB-IoT*.

Пристрої *NB-IoT* споживатимуть менше електроенергії, оскільки вони передадуть невелику кількість даних із низькою швидкістю передачі даних приблизно 100–150 Кбіт/с. Підключені пристрої матимуть петлю для надсилання даних після кожного певного проміжку часу, що означає, що енергія буде збережена

весь час, за винятком випадків, коли дані надсилаються протягом кількох секунд. Лише потужність, споживана джерелами живлення та Інтернет-пристроєм. В результаті споживається набагато менше енергії, ніж пристрій *LTE IoT*.

Оскільки пристрій споживатиме низьку енергію та надсилатиме дані на низькій частоті, вартість електроніки продукту буде низькою для пристрою. Схеми також будуть менш завершеними, і, як наслідок, чіпи, які будуть створені для *NB-IoT*, будуть дешевшими, а загальна вартість продукту також зменшиться.

Використання смугового спектру для *NB-IoT* ліцензовано *3GPP*, і це завжди забезпечить якісний зв'язок *M2M* (машина-машина). В результаті користувачі *NB-IoT* отримають надійний сервіс.

Програми *IoT* добре відомі людям, які цікавляться IoT, але з використанням *NarrowBand* кількість програм *NB-IoT* значно зросте. Пристрій також можна використовувати для зв'язку між машинами, які знаходяться під землею або там, де нормальне покриття мобільної мережі неможливо отримати. Таким чином, його можна використовувати для домашнього, промислового та комерційного застосування. Крім того, додаток *Smart City* отримає чудову користь від технології *NB-IoT*.

Мережа *NB-IoT* є ліцензованою мережею, а це означає, що передані дані безпечні. Він забезпечить усі функції безпеки, як у мережі *LTE*, яка вже є перевіреною та надійною безпекою для користувачів, забезпечуючи автентифікацію даних користувача, цілісність і конфіденційність у всьому світі.

Електроніка в пристрої *IoT* споживала б менше енергії, ніж споживала б мобільний пристрій *LTE*. Таким чином, порівняно час автономної роботи пристрою буде більше. Кажуть, що термін служби батареї може навіть тривати до 10 років або більше для будь-якого окремого пристрою *IoT*.

Передача даних для мережі *NB-IoT* відбуватиметься двома способами, тобто дані можуть надсилатися та отримуватись, а мережа має смугу радіочастот 200 кГц.

Мережа *NB-IoT* може бути легко розгорнута в існуючій архітектурі стільникової мережі, що означає, що мобільні пристрої можуть легко підключатися до пристроїв *NB-IoT*, з приблизно 1 мільйоном пристроїв *IoT* на один мобільний

телефон. Таким чином, доступ до даних 1 мільйона пристроїв *IoT* можна отримати лише за допомогою 1 мобільного телефону, що збільшить кількість підключень.

Дуже відомий *Shanghai Disney Resort* у Китаї має практичну реалізацію програми *Smart Parking NB-IoT*, розробленої *Huawei* та *China Unicom*. Інтелектуальна система паркування допомагає визначити наявність вільного паркувального місця для клієнтів, підрахувати час паркування та розрахувати вартість, яку повинен сплатити клієнт. При цьому система працює надійно, споживаючи невелику енергію.

Багато будинків/суспільств мають підвал, і оскільки він знаходиться під землею, йому постійно потрібна електрика. Але використовуючи технологію *NB-IoT*, світло підвалу вмикатиметься лише тоді, коли хтось присутній, а не вручну вмикати його протягом довгих годин.

Великі телекомунікаційні компанії Індії конкурують одна з одною за технологію *4G*, і всі вони планують незабаром запровадити мережу *NB-IoT* в Індії разом із мережею *5G*. Такі телекомунікаційні гіганти, як JIO, *Airtel* і *Vodafone*, планують незабаром ознайомитися з технологією *NB-IoT* і розширити свою мережу *5G* по всій Індії. Хоча вони вже запустили свої плани чи продукти для пристроїв *IoT*, у 2020 році всі вони також керуватимуть технологією NB-*IoT*.

1.3.2. Протоколи комунікації та передачі даних

Протоколи комунікації та передачі даних в Інтернеті речей (*IoT*) є суттєвими компонентами, які забезпечують ефективний обмін інформацією між різними пристроями. Вони дозволяють забезпечити надійність, безпеку та ефективність передачі даних в різних умовах.

MQTT (*Message Queuing Telemetry Transport*) - це легкий та ефективний протокол передачі повідомлень для мереж Інтернету речей (*IoT*). Він був розроблений з метою забезпечити надійний та ефективний обмін даними між різними пристроями, особливо в умовах обмеженого ресурсу та низької швидкості передачі даних. *MQTT* - це протокол з мінімальними накладними витратами. Він використовує *TCP/IP* протокол для передачі даних та забезпечує мінімальну кількість додаткових даних у повідомленнях. *MQTT* підтримує асинхронний обмін повідомленнями, що дозволяє пристроям спілкуватися без очікування відповіді.

MQTT підтримує систему з відмовостійкістю, що означає, що пристрої можуть продовжувати обмін даними, навіть якщо деякі з них вийшли з ладу. *MQTT* підтримує три рівні якості обслуговування (*QoS*) для передачі повідомлень: *QoS* 0 (найнижча гарантія), *QoS* 1 (гарантована доставка) і *QoS* 2 (подвійна гарантія) (рис. 1.7).

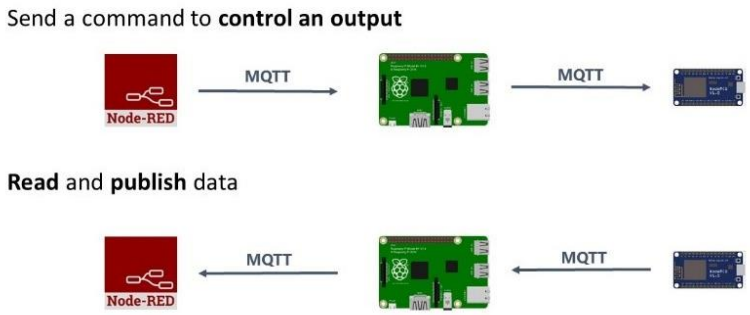

Рис. 1.7. Схема принципу роботу *MQTT*

Протокол працює на основі публікації-підписки, де пристрої відправляють повідомлення на певні теми, і інші пристрої (підписники) отримують ці повідомлення на підставі своїх підписок на теми.

MQTT здатний зберігати повідомлення для підписників, які тимчасово відсутні або недоступні.

Коли підписник з'являється, він отримує збережені повідомлення. *MQTT* може використовуватись в масштабних мережах *IoT*, де можна мати багато публікаторів та підписників.

MQTT підтримує захист даних за допомогою *TLS/SSL* протоколу, що забезпечує захищену передачу даних.

Протокол *MQTT* застосовується в різних областях, таких як моніторинг промислового обладнання, смарт-доми, автономні транспортні засоби, телеметрія, охорона здоров'я та багато інших.

Він є одним із ключових протоколів для обміну даними в *IoT* та допомагає створювати зв'язані, "розумні" системи.

CoAP (Constrained Application Protocol) - це протокол, спеціально розроблений для обмежених пристроїв та мереж Інтернету речей (*IoT*), які мають обмежені ресурси, такі як обсяг пам'яті, потужність та пропускна здатність.

Основна ідея застосування *CoAP* полягає в забезпеченні зручності та ефективності обміну даними у контексті мереж *IoT* (рис. 1.8).

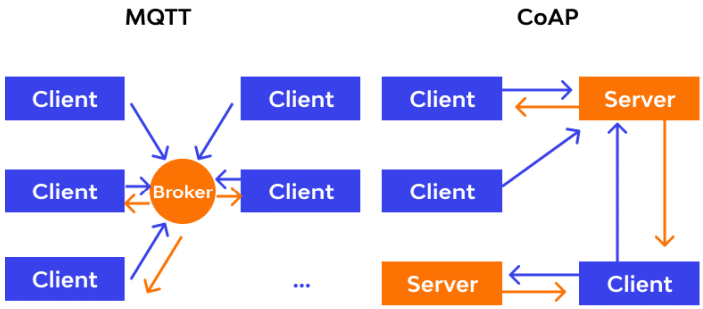

Рис. 1.8. Схема порівняння *CoAP* та *MQTT*

CoAP спеціально призначений для обмежених пристроїв, тому він має мінімальний наклад на ресурси пристроїв. Це дозволяє йому працювати на пристроях з низькою потужністю та обсягом пам'яті.

CoAP використовує архітектуру *REST* (*Representational State Transfer*), що дозволяє звертатися до ресурсів за допомогою стандартних *HTTP*-подібних методів, таких як *GET*, *POST*, *PUT* та *DELETE*.

Протокол підтримує асинхронну передачу даних, а також можливість гарантованої доставки повідомлень за допомогою вбудованого підтвердження отримання.

CoAP також підтримує рівні якості обслуговування, подібно до *MQTT*: *QoS* 0 (найнижча гарантія), *QoS 1* (гарантована доставка) і *QoS 2* (подвійна гарантія).

Протокол працює з ресурсами, на які можна відсилати запити та з яких можна отримувати відповіді. Це дозволяє створювати структуровані *API* для обміну даними. *CoAP* підтримує можливість кешування, що дозволяє прискорити доступ до ресурсів та знизити навантаження на мережу.

Протокол підтримує захист даних за допомогою *DTLS* (*Datagram Transport Layer Security*), що дозволяє забезпечити захищену передачу даних.

CoAP використовується в сценаріях, де важливий низький споживання енергії, обмежені ресурси та обмін даними в режимі реального часу. Він дуже популярний у застосуваннях *IoT*, таких як "розумний дім", моніторинг обладнання, вимірювання

середовища та багато інших випадків, де важливо мінімізувати споживання ресурсів та мережеве навантаження.

HTTP (*Hypertext Transfer Protocol*) та *HTTPS* (*Hypertext Transfer Protocol Secure*) - це стандартні протоколи комунікації в Інтернеті, що використовуються для передачі даних між клієнтами та серверами. Вони можуть бути також використовані в контексті Інтернету речей (*IoT*) для обміну даними між пристроями та серверами.

HTTP передбачає взаємодію між клієнтом та сервером за допомогою запитів та відповідей. Клієнт відправляє запит на сервер, а сервер відповідає на цей запит, надсилаючи додаткові дані.

HTTP підтримує різні методи запитів, такі як *GET* (отримання даних), *POST* (створення нових даних), *PUT* (оновлення даних) та *DELETE* (видалення даних). Це дозволяє виконувати різні дії з ресурсами на сервері.

Кожен ресурс на сервері має унікальний *URL*-адресу, за допомогою якої його можна ідентифікувати та звертатися до нього.

HTTP передає метадані у заголовках запиту та відповіді. Це може включати інформацію про тип контенту, дату, джерело тощо. *HTTP* дозволяє встановлювати сесії між клієнтом та сервером, що дозволяє відстежувати користувачів та зберігати стан між різними запитами. Кукі (*cookies*) використовуються для збереження інформації на боці клієнта.

HTTPS - це захищена версія *HTTP*, де дані шифруються за допомогою *SSL/TLS* протоколу. Це забезпечує конфіденційність та захист від перехоплення даних.

HTTP включає в себе роль серверів (які надають ресурси) та клієнтів (які звертаються до цих ресурсів). У контексті IoT це можуть бути, наприклад, сенсори, актуатори, мобільні додатки та сервери хмарних служб.

HTTP/HTTPS використовуються в різних сценаріях *IoT*, таких як збір та відображення даних з датчиків, управління віддаленими пристроями, моніторинг стану систем та багато інших випадків. *HTTPS* особливо важливий, коли безпека даних є пріоритетом, забезпечуючи шифрування даних та захист від несанкціонованого доступу.

Advanced Message Queuing Protocol (*AMQP*) - це відкритий та стандартизований протокол передачі повідомлень між різними пристроями та системами в мережах Інтернету речей (*IoT*). *AMQP* був спроектований для забезпечення надійної та ефективної передачі повідомлень у розподілених середовищах.

AMQP дозволяє структурувати повідомлення за допомогою різних типів даних, таких як рядки, числа, масиви та об'єкти. Це сприяє однорідності даних та полегшує їх обробку.

AMQP має рівні взаємодії, які дозволяють відокремити рівні забезпечення, маршрутизації та обробки повідомлень. Це дозволяє підтримувати різні архітектури та сценарії взаємодії. *AMQP* забезпечує гарантовану доставку повідомлень, що означає, що повідомлення будуть доставлені до призначеної точки прийому навіть у випадку відмови чи збою.

AMQP підтримує підтвердження отримання повідомлень, дозволяючи впевнитись у тому, що повідомлення були успішно доставлені та оброблені.

AMQP дозволяє налаштовувати маршрутизацію повідомлень на основі різних критеріїв, таких як тип повідомлення чи властивості.

Протокол розроблений з урахуванням масштабованості та здатності обробляти великі обсяги повідомлень у розподіленому середовищі. *AMQP* підтримується багатьма мовами програмування, що дозволяє реалізовувати рішення для *IoT* на різних платформах.

AMQP використовується в різних сферах *IoT*, де важлива надійність та гарантована доставка повідомлень, таких як моніторинг промислового обладнання, фінансові транзакції, медична телеметрія та багато інших. Протокол допомагає створювати розподілені та надійні системи обміну даними в умовах масштабованих мереж *IoT*.

5G - це п'яте покоління мобільних комунікацій, що представляє собою значне покращення порівняно з попереднім *4G*. Ця технологія відкриває нові можливості для швидкої та надійної передачі даних у мережах з більш великим обсягом пристроїв та високою швидкістю передачі.

5G надає значно вищу швидкість передачі даних порівняно з *4G*. Це дозволяє завантажувати та відтворювати великі файли, відео високої якості та інші важливі завдання швидше.

Однією з головних особливостей *5G* є низька затримка в передачі даних. Це важливо для реалізації таких сценаріїв, як віддалене управління обладнанням, автономні автомобілі та інші додатки, де кожна мілісекунда має значення.

5G забезпечує велику пропускну здатність мережі, що дозволяє підключати велику кількість пристроїв до мережі одночасно без втрати якості зв'язку. *5G* розроблено з огляду на підтримку масштабних мереж Інтернету речей (*IoT)*, де можуть бути підключені різні типи пристроїв - від датчиків до великих індустріальних обладнань.

5G розроблено таким чином, щоб воно могло працювати в мережах з великою кількістю підключених пристроїв, які можуть бути розташовані навіть на обмеженій площі. *5G* використовує більше частотних діапазонів, включаючи вищі частоти, що дозволяє забезпечити більше пропускної здатності та швидкості передачі.

5G дозволяє створювати віртуальні мережі для різних застосувань, що полегшує управління мережею та надає більше гнучкості. *5G* включає покращені механізми захисту даних, що стає особливо важливим у зв'язку зі зростаючою кількістю підключених пристроїв.

5G використовується в різних галузях, від мобільних зв'язків та розумних міст до індустріальних автоматизованих систем та медичних додатків. Ця технологія відкриває нові горизонти для розвитку Інтернету речей та дозволяє реалізувати інноваційні та високотехнологічні рішення.

Успішне використання *IoT* залежить від правильного вибору бездротової технології, яка відповідає специфікаціям та потребам конкретного проекту. З введенням *5G* та подальшим розвитком бездротових технологій, *IoT* отримає ще більше можливостей для свого росту і розширення в різних сферах застосування.

1.3.3. Стандарти безпеки та приватності в *IoT*

Стандарти безпеки та приватності в Інтернеті речей (*IoT*) є критично важливими для забезпечення надійності, захищеності та конфіденційності даних, які

обмінюються між підключеними пристроями. Оскільки *IoT* включає в себе різноманітні пристрої з різними можливостями та функціями, створення спільних стандартів є важливою умовою для розвитку безпечних та надійних рішень.

Стандарт *IEEE 802.15.4* є одним з ключових стандартів для бездротової комунікації в мережах Інтернету речей (*IoT*) та інших промислових бездротових мережах. Він визначає фізичний та доступовий контроль до мереж низької потужності та великого розповсюдження, які використовуються для збирання даних з різних датчиків та передачі їх до центральних вузлів для подальшої обробки.

Стандарт визначає різні фізичні шари для роботи на різних частотних діапазонах, таких як 2.4 ГГц, 868 МГц та 915 МГц. Це дозволяє пристроям працювати в різних регіонах світу, де доступні різні бездротові діапазони.

Стандарт використовує різні методи доступу, такі як *CSMA-CA* (*Carrier Sense Multiple Access with Collision Avoidance*) та *TDMA* (*Time Division Multiple Access*), для уникнення колізій та ефективного використання доступового каналу.

IEEE 802.15.4 визначає структуру мережі зі спеціальними типами вузлів: *PAN Coordinator* (координатор мережі), *Full-Function Device* (*FFD* - повноправний пристрій) та *Reduced-Function Device* (*RFD* - пристрій з обмеженими можливостями).

Стандарт надає базові механізми безпеки, такі як шифрування та аутентифікація, для захисту передачі даних у мережі. За допомогою шифрування та ключів, можливо захистити дані від несанкціонованого доступу та перехоплення.

Однією з ключових переваг стандарту є його спроможність працювати на низькій споживаній потужності. Це дозволяє пристроям працювати довше від батарей та забезпечує довгий термін служби вузлів *IoT*.

IEEE 802.15.4 дозволяє створювати великі мережі, включаючи тисячі пристроїв, які можуть співпрацювати для обміну даними та взаємодії.

Хоча стандарт визначає фізичний та доступовий контроль, він не надає специфікації для протоколів верхніх шарів. Проте, багато розробників використовують стеки протоколів, такі як *Zigbee* та *6LoWPAN*, щоб забезпечити повні функціональність мереж *IoT*.

Стандарт *IEEE 802.15.4* є однією з основних основ для бездротової комунікації в *IoT*. Він забезпечує надійну та низькопотужну комунікацію між пристроями та вузлами мережі, дозволяючи створювати різноманітні застосування в різних галузях, від домашнього використання до промислової автоматизації.

Open Connectivity Foundation (*OCF)* - це стандарт та фреймворк для забезпечення інтероперабельності між різними пристроями та системами Інтернету речей (*IoT*). *OCF* була створена з метою розв'язання проблеми різноманітності пристроїв, протоколів та платформ, яка ускладнює взаємодію між пристроями в мережі *IoT*.

Основна мета *OCF* - створити стандарт, який дозволить пристроям з різних виробників та платформ взаємодіяти між собою. Це важливо, оскільки на ринку присутні різноманітні платформи та протоколи, і можливість спільної роботи є важливою для розвитку *IoT*.

OCF намагається зробити розробку та використання пристроїв *IoT* максимально простим. Це осягається шляхом створення стандартних *API* та інтерфейсів для різних типів пристроїв, що спрощує їх інтеграцію та взаємодію. *OCF* надає увагу безпеці та приватності. Він використовує шифрування та аутентифікацію для захисту передачі даних між пристроями.

OCF підтримує різні типи пристроїв - від датчиків та актуаторів до розумних домашніх пристроїв та медичних систем. Це робить його придатним для застосування в різних галузях. *OCF* використовує стек протоколів *CoAP* (*Constrained Application Protocol*) для обміну даними між пристроями. Цей протокол є оптимізованим для праці в обмежених ресурсах.

OCF - відкрита організація, і стандарти та документація доступні для спільного використання. Це дозволяє розробникам використовувати їх у своїх проектах та продуктах. *OCF* прагне забезпечити сумісність з іншими стандартами та фреймворками, такими як *Zigbee, Thread* та інші, щоб дозволити інтеграцію між різними системами.

OCF використовується в різних галузях, включаючи смарт-дом, промисловість, охорону здоров'я тощо. Він допомагає спростити розробку та впровадження IoT рішень, забезпечуючи їхню сумісність та безпеку.

ISO/IEC 27001 є міжнародним стандартом, який визначає вимоги до системи управління інформаційною безпекою (*ISMS - Information Security Management System*). Цей стандарт надає методологію та підходи для встановлення, впровадження, утримання та постійного вдосконалення системи управління безпекою інформації в організації.

Стандарт охоплює всі аспекти, пов'язані з інформаційною безпекою, включаючи захист інформації, управління ризиками, фізичну безпеку, захист від зловживань тощо.

ISO/IEC 27001 базується на підході до управління ризиками. Організації повинні ідентифікувати потенційні загрози та ризики для інформації, оцінювати їх вплив та ймовірність виникнення, та впроваджувати заходи для їх зниження.

Стандарт надає структурований підхід до створення та впровадження *ISMS*. Це включає визначення політики безпеки, встановлення контрольних заходів, оцінку ефективності та постійне вдосконалення.

ISO/IEC 27001 може бути використаний для підтвердження відповідності організації вимогам щодо інформаційної безпеки. Аудитори здійснюють перевірку відповідності та ефективності заходів забезпечення безпеки.

ISO/IEC 27001 може бути впроваджений в різних типах організацій, від корпорацій до урядових установ та неприбуткових організацій.

З урахуванням зростаючого використання *IoT*, стандарт *ISO/IEC 27001* може бути застосований для забезпечення безпеки даних в мережах *IoT*. Він допомагає ідентифікувати та управляти ризиками, пов'язаними зі збиранням, передачею та зберіганням даних в *IoT* системах.

Застосування *ISO/IEC 27001* допомагає організаціям забезпечити високий рівень безпеки інформації, підвищити довіру споживачів та стейкхолдерів, а також відповідати регуляторним вимогам щодо безпеки даних.

NIST (*National Institute of Standards and Technology*) *Cybersecurity Framework* - це комплексна система рекомендацій та практик для покращення кібербезпеки в організаціях. Розроблений Національним інститутом стандартів і технологій США, цей фреймворк визначає ключові елементи, необхідні для захисту від кіберзагроз.

Розроблений у 2014 році, фреймворк базується на кращих практиках, стандартах та підходах до кібербезпеки.

Основна мета фреймворку – допомогти організаціям підвищити рівень захисту від кіберзагроз шляхом впровадження рекомендацій та методів безпеки.

Фреймворк складається з п'яти ключових компонентів:

– *Framework core,* що включає кібербезпекові об'єктиви та практики, що сприяють досягненню кібербезпеки.

– *Implementation tiers,* що вказує на рівень кібербезпеки організації - від базового до передового.

– *Profiles,* що допомагає організаціям створювати індивідуальний план дій на основі конкретних вимог та ризиків.

– *Function categorie*, що визначає функції, необхідні для досягнення кібербезпеки, такі як ідентифікація, захист, виявлення, відповідь та відновлення.

– *Informative* references, що містить вказівки на додаткові ресурси та стандарти для покращення кібербезпеки.

Організації можуть використовувати фреймворк для створення індивідуального плану дій з покращення кібербезпеки.

Фреймворк може бути адаптований до різних галузей та розмірів організацій, зокрема до великих корпорацій, малого бізнесу та урядових установ.

Фреймворк застосовується в різних галузях, включаючи бізнес, уряд, критичну інфраструктуру та багато інших.

Допомагає організаціям ідентифікувати ризики, розробляти політику безпеки, впроваджувати та здійснювати контроль за заходами безпеки.

Фреймворк пропонує ітеративний підхід, включаючи оцінку ефективності, аналіз результатів та коригування стратегії для покращення кібербезпеки.

Цей фреймворк є важливим інструментом для розробки та покращення стратегій кібербезпеки, допомагаючи організаціям ефективно впроваджувати заходи безпеки, виявляти та реагувати на загрози кібербезпеці.

GDPR (Загальний регламент з захисту даних) - це регуляторний акт Європейського Союзу, прийнятий з метою забезпечення захисту особистих даних громадян ЄС та встановлення однорідних правил для обробки та передачі цих даних. *GDPR* набув чинності 25 травня 2018 року і впливає на всі організації, які обробляють особисті дані громадян ЄС, незалежно від їхнього місцезнаходження.

Регламент застосовується до будь-якої організації, незалежно від місцезнаходження, яка обробляє особисті дані громадян ЄС. Організації повинні обробляти дані законно та чесно.

Дані повинні збиратися для певних законних цілей та не можуть використовуватися інакше. Обмеження обсягу та видів зібраних даних до необхідного мінімуму.

Дані повинні бути точними та оновлюватися при необхідності. Дані повинні зберігатися лише протягом необхідного терміну. Дані повинні бути захищені від несанкціонованого доступу та обробки.

GDPR надає громадянам ЄС ряд прав, зокрема право на доступ до своїх даних, право на забуття, право на перенос даних, право на виправлення невірних даних та інші.

Організації, які обробляють дані, повинні встановити внутрішні процедури для забезпечення дотримання положень *GDPR*, включаючи впровадження заходів безпеки та призначення уповноважених осіб з питань захисту даних.

Організації повинні повідомляти відповідні регуляторні органи та суб'єктів даних про будь-яке порушення безпеки, яке може призвести до ризику для прав та свобод суб'єктів даних.

Для передачі особистих даних за межі ЄС організаціям потрібно забезпечити адекватний рівень захисту даних у країні-одержувачі. За порушення положень *GDPR* передбачені великі штрафи, які можуть становити значну частку обороту організації.

Цей регламент впливає на спосіб, яким організації збирають, обробляють та зберігають дані громадян ЄС, і вимагає забезпечення високого рівня захисту особистої інформації та конфіденційності.

Ці стандарти, серед багатьох інших, грають ключову роль у забезпеченні безпеки та приватності в системах. Вони сприяють відкритості, стандартизації та підвищенню довіри до інтернету речей.

1.4. Архітектура Інтернету речей

1.4.1. Концепція класичної архітектури *IoT*

Класична архітектура Інтернету речей (*IoT*) є складною системою, що включає в себе різноманітні компоненти та елементи, які співпрацюють для забезпечення збору, обробки та обміну даними між пристроями та хмарними ресурсами.

Сенсори та датчики - це елементи, що є "розумними очима і вухами" *IoT*. Вони здатні збирати дані з фізичного світу, такі як температура, вологість, рух, освітленість тощо. Сенсори можуть бути різних типів, від температурних до біометричних, і генерують дані, які будуть подальше оброблятися.

Пристрої та вузли - це різноманітні пристрої, які можуть включати в себе сенсори, контролери, мікропроцесори та інші електронні компоненти. Вони можуть обробляти дані на місці та приймати рішення, відправляти дані до хмари або до інших пристроїв.

Хмарні сервіси та інфраструктура включають в себе хмарні ресурси, сервери та інфраструктуру для зберігання, обробки та аналізу даних. Важливою частиною є забезпечення масштабованості та високої доступності.

Комунікаційні протоколи та мережі - це основа взаємодії між різними компонентами *IoT*. Різні протоколи (наприклад, *MQTT, CoAP, HTTP*) використовуються для передачі даних між сенсорами, пристроями та хмарними ресурсами через бездротові або провідні мережі.

Додатки можуть бути розроблені для моніторингу, управління та аналізу даних *IoT*. Це можуть бути мобільні додатки, веб-інтерфейси, системи управління тощо.

Актуатори та виконавчі механізми - це компоненти, що дозволяють впливати на фізичний світ на основі даних, що були зібрані та оброблені. Актуатори можуть бути різних типів, наприклад, пристрої для регулювання температури, віддаленого включення/виключення пристроїв тощо.

Забезпечення безпеки та захисту даних є критичним аспектом архітектури *IoT*. Це включає в себе захист від несанкціонованого доступу, шифрування даних та ідентифікацію пристроїв.

Менеджмент та моніторинг - це частина архітектури, що включає рішення для управління та моніторингу пристроїв, збору даних про їх стан, віддаленого керування та налагодження (рис. 1.9).

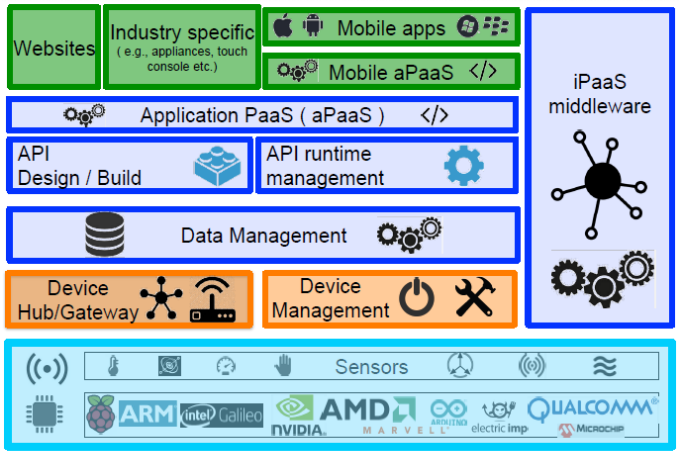

Рис. 1.9. Схема архітектури *IoT*

Ця класична архітектура *IoT* може варіюватися в залежності від конкретного випадку використання, але загальні принципи взаємодії та функціонування залишаються подібними. Вона надає засади для створення комплексних систем *IoT*, які відповідають вимогам підключених та автоматизованих середовищ.

1.4.2. Розподілені та хмарні архітектури *IoT*

Розподілені та хмарні архітектури Інтернету речей (*IoT*) є двома ключовими підходами до організації та управління великими мережами підключених пристроїв. Обидва підходи мають свої переваги та обмеження, і їх вибір залежить від конкретних потреб та сценаріїв використання.

У розподіленій архітектурі, обробка даних відбувається в основному на самому пристрої або на локальному вузлі в мережі.

Дані збираються та обробляються на самому пристрої або на вузлі в локальній мережі. Це дозволяє швидку відповідь та відсутність потреби в постійному підключенні до хмари. Такий підхід дозволяє зменшити затримку в передачі даних, оскільки обробка відбувається на місці збору.

Продукти на основі розподіленої архітектури можуть функціонувати навіть у випадку відсутності доступу до хмарних ресурсів.

У хмарній архітектурі, дані збираються та передаються до хмарного сервісу для подальшої обробки та аналізу. Основні риси хмарної архітектури *IoT* включають:

Дані збираються на пристроях та вузлах, але основна обробка та аналіз відбуваються на хмарних серверах. Хмарні сервіси можуть надавати велику обчислювальну потужність для складних обчислень та аналітики даних. Хмарні сервіси дозволяють легко масштабувати інфраструктуру в залежності від обсягу даних та потреб. Всі дані та операції можуть бути керовані з одного центрального місця.

Обидва підходи мають свої переваги та використовуються в різних випадках використання.

Наприклад, розподілена архітектура може бути вигідною для важливих додатків, де затримка та локальний контроль є критичними.

З іншого боку, хмарна архітектура дозволяє використовувати потужні обчислювальні ресурси для складних аналітичних завдань та масштабованих рішень.

У багатьох випадках, компанії використовують гібридні підходи, поєднуючи розподілені та хмарні архітектури, щоб забезпечити оптимальний баланс між локальною обробкою та централізованою аналітикою.

1.4.3. Роль розумних датчиків та актуаторів в архітектурі *IoT*

Розумні датчики та актуатори є ключовими компонентами в архітектурі Інтернету речей (*IoT*), оскільки вони дозволяють збирати дані з фізичного середовища, а також впливати на нього на основі зібраних даних. Ці компоненти відіграють критичну роль у створенні автоматизованих та підключених систем в різних сферах, від промисловості до домашнього використання.

Розумні датчики здатні вимірювати різні фізичні параметри, такі як температура, вологість, тиск, освітленість, рух, звук та багато інших. Вони перетворюють ці параметри на цифрові дані, які можуть бути оброблені та аналізовані. Розумні датчики дозволяють реальному часу моніторити стан різних об'єктів та процесів. Наприклад, вони можуть слідкувати за температурою в промисловому обладнанні або рівнем вологості в ґрунті в сільському господарстві.

Розумні датчики можуть бути розташовані важкодоступних або небезпечних місцях, де людям складно або небезпечно проникнути. Вони можуть легко інтегруватися з іншими системами та пристроями, надаючи їм інформацію для прийняття рішень або взаємодії.

Розумні актуатори здатні змінювати стан фізичних об'єктів або середовища на основі отриманих даних. Наприклад, вони можуть регулювати температуру приміщення, включати/виключати світло, відкривати/закривати двері тощо. Розумні актуатори можуть бути керовані віддалено через мережу. Це дає можливість взаємодіяти з пристроями та середовищем, навіть якщо користувач не знаходиться фізично поруч.

За допомогою розумних актуаторів можна створювати автоматизовані сценарії. Наприклад, вимкнення підсвічування при виході з приміщення або автоматичне поливання рослин залежно від вологості ґрунту.

Розумні актуатори можуть реагувати на отримані дані навіть без прямого втручання користувача. Наприклад, автоматично включити систему опалення при зниженні температури.

Розумні датчики та актуатори взаємодіють ускладнюють архітектуру *IoT*, дозволяючи створювати розумні та автоматизовані системи, які реагують на зміни в оточенні та взаємодіють з ним.

Висновки за розділом

В ході дослідження було проведено аналіз наукових публікацій, наукових статей та джерел, що стосуються засобів моніторингу та управління в контексті

Інтернету речей. Визначено, що *IoT* є важливим напрямком розвитку сучасних технологій, де підключені пристрої сприяють автоматизації, оптимізації та підвищенню продуктивності у різних галузях.

Предметом дослідження є засоби моніторингу та управління, що створюють платформу для обміну даними між підключеними пристроями, аналізу цих даних та впливу на навколишнє середовище через актуатори. Розкрито основні аспекти функціонування таких засобів, включаючи сенсори, актуатори, протоколи комунікації, платформи та програмне забезпечення.

В цілому, аналіз першого розділу надав поглиблене розуміння концепції моніторингу та управління в контексті Інтернету речей, розкрив важливі аспекти цієї теми та вказав на актуальні проблеми, які потребують подальшого дослідження та вирішення.

В подальших розділах магістерської роботи будуть розглянуті більш докладні аспекти, а також розроблені можливі рішення для вирішення ідентифікованих викликів.

РОЗДІЛ 2
АНАЛІЗ ЗАСОБІВ МОНІТОРИНГУ ТА УПРАВЛІННЯ ЕЛЕМЕНТАМИ ІНТЕРНЕТУ РЕЧЕЙ

2.1. Означення засобів моніторингу та управління в *IoT*

Зважаючи на контекст Інтернету речей (*IoT*), моніторинг та управління є ключовими аспектами функціонування систем, побудованих на базі *IoT*.

Моніторинг означає систематичний процес збору, спостереження, аналізу та відстеження даних, що стосуються підключених пристроїв, сенсорів або систем, які належать до *IoT*.

2.1.1. Визначення понять "моніторинг" та "управління" в *IoT*

Цілью моніторингу є отримання інформації про стан, функціонування та ефективність цих пристроїв або систем.

Моніторинг в *IoT* включає в себе багато важливих аспектів.

Збір даних - це процес збору даних з підключених пристроїв або сенсорів, які можуть вимірювати різні параметри, наприклад, температуру, вологість, рівень розходження, віддаленість тощо.

Спостереження - це процес спостереження за отриманими даними, перевірки їхньої достовірності та виявлення будь-яких аномалій, помилок або відхилень від заданих параметрів.

Аналіз - це процес обробки та аналізу зібраних даних з метою отримання корисної інформації, виявлення трендів, залежностей або шаблонів, а також прогнозування майбутніх станів або подій.

Відстеження - це процес постійного контролю за даними та їхньої актуалізації з урахуванням змін в стані або поведінці підключених пристроїв.

Це дозволяє оперативно реагувати на відхилення та приймати необхідні рішення.

Моніторинг в *IoT* допомагає забезпечувати ефективне функціонування підключених пристроїв, виявляти проблеми або відхилення в роботі та забезпечувати своєчасну реакцію на них.

Управління в *IoT* означає процес контролю, координації та керування підключеними пристроями, системами або мережами *IoT* з метою досягнення певних цілей або оптимізації їхньої роботи.

Керування пристроями - це процес керування функціями, режимами роботи, конфігурацією або поведінкою підключених пристроїв через централізовану або розподілену систему управління.

Координація - це процес забезпечення взаємодії та синхронізації різних підключених пристроїв або систем для досягнення спільних цілей або розв'язання взаємозалежних завдань.

Керування мережею - це процес управління комунікаційною інфраструктурою *IoT*, включаючи маршрутизацію, протоколування, безпеку, контроль за пропускною здатністю та енергоспоживанням.

Адміністрування та безпека - це процес забезпечення безпеки, ідентифікації, аутентифікації, авторизації та аудиту підключених пристроїв, а також встановлення правил, політик та процедур для ефективного управління *IoT*.

Управління в *IoT* дозволяє забезпечувати цілеспрямовану роботу підключених пристроїв, оптимізувати використання ресурсів, реагувати на зміни в оточенні та вдосконалювати функціональні можливості систем *IoT*.

В цілому, моніторинг і управління в контексті *IoT* взаємопов'язані процеси, що забезпечують ефективне та безперебійне функціонування систем *IoT*, забезпечують збір та аналіз даних, контроль за пристроями та управління мережами, сприяють підвищенню продуктивності, надійності та безпеки в *IoT* -системах.

Моніторинг означає систематичний процес збору, спостереження, аналізу та відстеження даних, що стосуються підключених пристроїв, сенсорів або систем, які належать до *IoT*.

2.1.2. Опис характеристик засобів моніторингу та управління

Засоби моніторингу та управління в Інтернеті речей (*IoT*) включають різноманітні компоненти, які сприяють збору даних, передачі інформації та керуванню підключеними пристроями. Основні характеристики таких засобів включають:

1. Сенсори, які є ключовими компонентами в *IoT*-системах. Вони призначені для збору фізичних або хімічних параметрів з оточуючого середовища. Основні характеристики сенсорів включають тип збирання даних (температура, вологість, рух тощо), точність вимірювання, швидкість оновлення даних, енергоспоживання та зв'язок з мережею передачі даних.

2. Актуатори, які виконують функцію керування фізичними пристроями або системами на основі отриманих від сенсорів даних. Вони можуть включати соленоїди, мотори, клапани, світлодіоди тощо. Основні характеристики актуаторів включають тип керування, швидкість реакції, точність позиціювання, витрата енергії та зв'язок з мережею управління.

3. Протоколи комунікації, що визначають правила передачі даних між підключеними пристроями в *IoT* -системах.

Вони можуть включати бездротові протоколи, такі як *Wi-Fi*, *Bluetooth*, *Zigbee*, *LoRaWAN*, або проводові протоколи, такі як *Ethernet*. Основні характеристики протоколів комунікації включають швидкість передачі даних, дальність зв'язку, стабільність підключення, енергоефективність та сумісність з іншими пристроями.

4. Платформи *IoT*, що надають інфраструктуру та середовище для розробки, управління та моніторингу *IoT*-рішень.

Вони можуть включати хмарні платформи, локальні сервери або розподілені системи обробки даних. Основні характеристики платформ включають масштабованість, надійність, безпеку, можливості аналітики даних, інтеграцію з іншими системами та підтримку розробки додатків.

5. Програмне забезпечення, яке включає програми, драйвери, бібліотеки та інструменти, які дозволяють налагоджувати, керувати та аналізувати *IoT*-системи. Воно може включати операційні системи для пристроїв *IoT*, платформи для розробки

додатків, аналітичні інструменти, засоби безпеки тощо. Основні характеристики програмного забезпечення включають простоту використання, функціональність, сумісність зі стандартами *IoT* та розширюваність.

Ці характеристики засобів моніторингу та управління в *IoT*-системах є важливими при проектуванні, виборі та інтеграції компонентів для успішної реалізації IoT-рішень. Залежно від конкретного сценарію використання, можуть вимагатися різні характеристики для досягнення максимальної ефективності, безпеки та надійності систем *IoT*.

2.2. Переваги та виклики використання засобів моніторингу та управління в *IoT*

Засоби моніторингу та управління в контексті Інтернету речей (*IoT*) надають численні переваги, які сприяють покращенню ефективності, надійності та безпеки систем.

Засоби моніторингу та управління пропонують широкі можливості для оптимізації та управління різними системами, підвищення ефективності та забезпечення безпеки. Ці переваги роблять їх невід'ємною складовою сучасних інноваційних рішень у багатьох галузях, включаючи промисловість, транспорт, охорону довкілля, охорону здоров'я та будівництво.

2.2.1. Аналіз переваг для забезпечення засобів моніторингу та управління в контексті *IoT*

Засоби моніторингу та управління дозволяють збирати великі обсяги даних з різних джерел, включаючи сенсори, пристрої та мережі. Це дозволяє здійснювати глибокий аналіз даних і виявляти корисні залежності, тренди та патерни, що допомагають в прийнятті обґрунтованих рішень та оптимізації процесів.

IoT-системи дозволяють віддалено моніторити та керувати підключеними пристроями та процесами з будь-якого місця, що забезпечує зручність та гнучкість управління. Це може бути особливо корисно в сферах, де доступність та швидкість реакції є критичними, наприклад, в промисловому управлінні або системах безпеки.

Засоби моніторингу та управління *IoT* дозволяють автоматизувати рутинні задачі, оптимізувати робочі процеси та забезпечувати реалізацію систем, що самоналаштовуються. Це сприяє зниженню людського втручання, підвищує продуктивність та ефективність роботи систем.

Засоби моніторингу та управління *IoT* дозволяють отримувати дані в режимі реального часу та забезпечувати швидку реакцію на зміни. Це особливо важливо в ситуаціях, коли потрібна негайна реакція на події або аварійні ситуації.

Засоби моніторингу та управління *IoT* включають механізми захисту даних, шифрування, автентифікації та контролю доступу, що забезпечує підвищену безпеку систем. Вони також дозволяють виявляти та реагувати на відхилення, несправності або аварійні ситуації, що сприяє забезпеченню високої надійності роботи систем.

Засоби моніторингу та управління *IoT* можуть допомогти знизити витрати на експлуатацію, підтримку та ремонт обладнання. Вони дозволяють здійснювати превентивне обслуговування, прогнозування витрат та раціональне використання ресурсів, що призводить до економічної вигоди.

2.2.2. Аналіз викликів та проблем, пов'язаних з використанням засобів моніторингу та управління

Використання засобів моніторингу та управління в сучасних системах Інтернету речей (*IoT*) також стикається з рядом викликів та проблем, які необхідно врахувати для успішної реалізації проектів *IoT*.

У *IoT* -системах використовуються різні протоколи комунікації, структури даних та архітектури, що ускладнює інтеграцію та взаємодію між різними компонентами. Необхідність стандартизації в *IoT*-галузі стає великим викликом для забезпечення сумісності, інтероперабельності та масштабованості систем.

Забезпечення безпеки в *IoT*-системах є надзвичайно важливою задачею. Зростання кількості підключених пристроїв та обміну даними створює нові вектори атак та загрози для конфіденційності, цілісності та доступності даних.

Недостатня захищеність може призвести до ускладнень, крадіжок даних або витоку інформації. Забезпечення безпеки в *IoT*-системах є складним завданням і

вимагає впровадження різних заходів, включаючи шифрування, аутентифікацію, контроль доступу та захист мережі.

Збір великого обсягу даних в *IoT*-системах породжує проблеми з приватністю та захистом особистої інформації.

Важливо розробляти та використовувати механізми анонімізації, контролю зберігання та передачі даних, щоб забезпечити конфіденційність інформації та дотримання вимог законодавства про захист персональних даних.

IoT-системи мають потенціал зростати до величезних розмірів з великою кількістю підключених пристроїв і обсягами даних. Забезпечення масштабованості системи, що здатна ефективно обробляти, аналізувати та керувати такими обсягами даних, може бути складним завданням, включаючи оптимізацію алгоритмів, інфраструктури та ресурсів.

Багато пристроїв *IoT* працюють на батарейках або обмежених джерелах живлення.

Тому енергоефективність є важливим фактором для продовження тривалого функціонування таких пристроїв без потреби в частій зарядці або заміні батарейок. Використання енергозберігаючих технологій, розумного керування споживанням енергії та оптимізація роботи пристроїв можуть допомогти вирішити цю проблему.

Збір інформації з багатьох джерел та обробка даних в режимі реального часу вимагають потужних аналітичних інструментів та інфраструктури.

Розробка алгоритмів, моделей машинного навчання та систем аналізу даних, які можуть працювати з великими обсягами даних в реальному часі, є важливим завданням для успішної реалізації *IoT*-систем.

Застосування *IoT*-систем пов'язане зі змінами у роботі та підходах користувачів до технологій. Необхідно брати до уваги людський фактор, забезпечувати навчання та підтримку користувачів, щоб забезпечити успішне впровадження та прийняття *IoT*-систем.

Використання засобів моніторингу та управління в системах Інтернету речей (*IoT*) може стикатися з різними проблемами, які варто врахувати.

Різні компоненти *IoT*, такі як сенсори, пристрої збору даних, мережі та платформи, можуть використовувати різні протоколи комунікації та стандарти. Виникнення проблем з сумісністю може ускладнити інтеграцію та взаємодію між різними компонентами системи.

IoT-системи містять велику кількість підключених пристроїв та обмін даними через мережу.

Це створює нові ризики для безпеки, такі як несанкціонований доступ до пристроїв, крадіжка даних, атаки на мережу тощо. Забезпечення безпеки в *IoT*-системах вимагає використання шифрування, аутентифікації, контролю доступу та інших заходів.

Збір великого обсягу даних в *IoT*-системах може створювати проблеми з приватністю та захистом особистої інформації. Недостатні заходи щодо анонімізації, контролю зберігання та передачі даних можуть порушувати приватність користувачів.

У *IoT*-системах, де критично важливо, щоб пристрої та мережі функціонували безперебійно, проблеми надійності можуть стати серйозними. Відмовка пристроїв, перебої в мережі або збої в програмному забезпеченні можуть призвести до відмови системи або неправильного функціонування.

Інфраструктура та системи повинні бути здатні обробляти великі обсяги даних та запитів, забезпечуючи ефективну роботу.

Багато пристроїв *IoT* працюють на обмежених джерелах енергії, таких як батарейки. Забезпечення енергоефективності пристроїв, оптимізація споживання енергії та продовження терміну служби батарейок є важливими факторами для успішного впровадження *IoT*-систем.

Відсутність стандартів у галузі *IoT* може призвести до фрагментації ринку, складнощів у взаємодії між різними системами та обмеження розвитку інновацій.

Загалом, використання засобів моніторингу та управління в *IoT*-системах стикається зі своїми викликами та проблемами. Проте, шляхом вирішення цих проблем та прийняттям належних заходів безпеки та захисту, можна забезпечити успішне функціонування та реалізацію потенціалу Інтернету речей.

2.3. Засоби моніторингу та управління в реальних сценаріях

2.3.1. Аналіз існуючих систем моніторингу та управління *IoT*

Аналіз існуючих систем моніторингу та управління в контексті Інтернету речей (*IoT*) включає дослідження та огляд різних реалізацій та платформ, які використовуються для збору даних, моніторингу стану пристроїв та управління ними. *IBM Watson IoT Platform* є однією з провідних платформ для розробки та впровадження рішень Інтернету речей (*IoT*). Вона надає комплексні можливості збору, аналізу та управління даними в *IoT*-системах, дозволяючи організаціям ефективно моніторити та керувати підключеними пристроями (рис. 2.1).

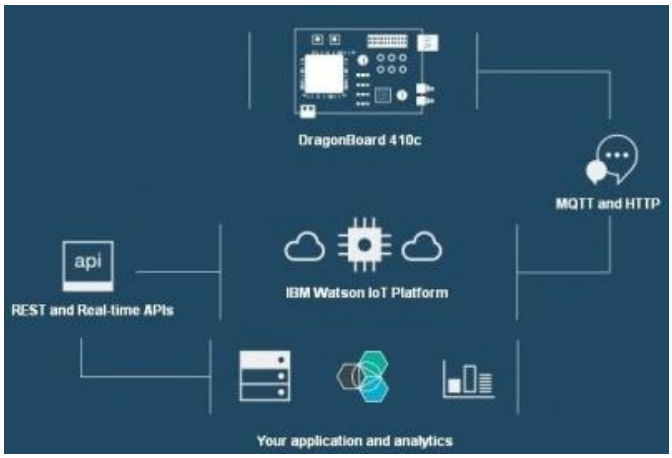

Рис. 2.1. Схема представлення взаємодії та позиції платформи *Watson Internet of Things (IoT)*

IBM Watson виник у результаті досліджень та розробок в області штучного інтелекту та аналізу великих обсягів даних. Система здатна аналізувати неймовірну кількість інформації та надавати контекстно-залежні відповіді на запитання.

Зростання інтересу до Інтернету речей призвело до того, що компанії почали розробляти платформи для управління підключеними пристроями та аналізу вироблених даних. *IBM Watson* був розширений у цю область.

Створення *IBM Watson IoT* Platform відбулось у контексті потреб ринку в інструментах для обробки та аналізу даних з підключених пристроїв. Платформа надає засоби для збору, аналізу та використання даних з *IoT*-пристроїв.

Протягом часу *IBM Watson IoT Platform* зростала функціонально і отримувала різноманітні можливості. Це включає інструменти для реального часу аналітики, безпеки та інтеграції з іншими сервісами.

BM активно розвиває стратегічні партнерства з іншими компаніями в галузі Інтернету речей. Це може включати спільні проекти, стандартизацію та інші ініціативи.

Як і будь-яка технологія, *IBM Watson IoT Platform* постійно оновлюється, додаючи нові функції, оптимізуючи роботу та враховуючи зміни в сфері Інтернету речей.

Історія *IBM Watson IoT Platform* тісно пов'язана з розвитком технологій штучного інтелекту та Інтернету речей загалом. Ця платформа є результатом вдосконалення та адаптації технологій для відповіді на виклики, що ставляться в галузі підключених пристроїв.

Платформа забезпечує широкі можливості для підключення різних типів пристроїв, включаючи датчики, девайси, машини тощо. Вона підтримує різні протоколи комунікації, такі як *MQTT, HTTP* та *LoRaWAN*, що дозволяє інтегрувати різні пристрої та платформи.

IBM Watson IoT Platform дозволяє збирати дані з підключених пристроїв та зберігати їх у хмарі. Платформа надає можливості для аналізу даних, використовуючи інструменти штучного інтелекту (*AI*) та машинного навчання (*ML*). Це дозволяє отримувати цінні інсайти з даних та робити прогнозування для поліпшення ефективності системи [4].

Платформа надає можливості для реал-тайм моніторингу стану пристроїв, збирання даних про їх роботу та статус.

Це дозволяє операторам системи вчасно виявляти проблеми, проводити діагностику та приймати рішення щодо ремонту або обслуговування пристроїв.

IBM Watson IoT Platform надає інструменти для керування підключеними пристроями з централізованої панелі керування. Це включає можливості дистанційного керування, налаштування параметрів пристроїв, оновлення програмного забезпечення та встановлення правил автоматизації.

IBM Watson IoT Platform надає *API* та інструменти для інтеграції з іншими системами та додатками.

Це дозволяє підключати різні сервіси, аналітичні інструменти та розробляти власні рішення на основі платформи.

Платформа забезпечує високий рівень безпеки для передачі, зберігання та обробки даних. Вона використовує шифрування, аутентифікацію та авторизацію для забезпечення конфіденційності та цілісності даних.

До переваг платформи *IBM Watson IoT* можна віднести такі пункти, як широкий спектр функціональності для аналізу та управління даними *IoT*, можливості машинного навчання та штучного інтелекту для виявлення патернів та прогнозування та розширена інтеграція з різними системами та протоколами.

Недоліками *IBM Watson IoT* можна назвати високу складність використання для новачків без попереднього досвіду в роботі з платформою та високі затрати порівняно з іншими рішеннями.

IBM Watson IoT Platform має широкий спектр застосувань в різних галузях, включаючи промисловість, транспорт, охорону здоров'я, сільське господарство та багато інших.

Amazon Web Services (AWS) IoT є однією з найпопулярніших та надійних платформ для розробки та впровадження рішень Інтернету речей (*IoT*). Її відмінності від інших платформ *IoT* полягають у ряді унікальних функцій та можливостей.

AWS IoT може працювати з великим обсягом пристроїв *IoT*. Вона надає можливість керувати мільйонами пристроїв одночасно, забезпечуючи стабільну та ефективну роботу системи. Платформа також дозволяє легко масштабувати рішення *IoT* від прототипу до продуктивного застосування, забезпечуючи підтримку зростаючого обсягу пристроїв.

AWS IoT підтримує різноманітні типи пристроїв, включаючи мікроконтролери, смартфони, сенсори та інші. Вона також підтримує різні протоколи комунікації, такі як *MQTT*, *HTTP* і *WebSocket*, що дозволяє легко підключати різні пристрої до платформи.

Крім того, *AWS IoT* пропонує *SDK* для різних платформ розробки, що дозволяє розробникам створювати додатки для пристроїв на основі їх вибору (рис. 2.2).

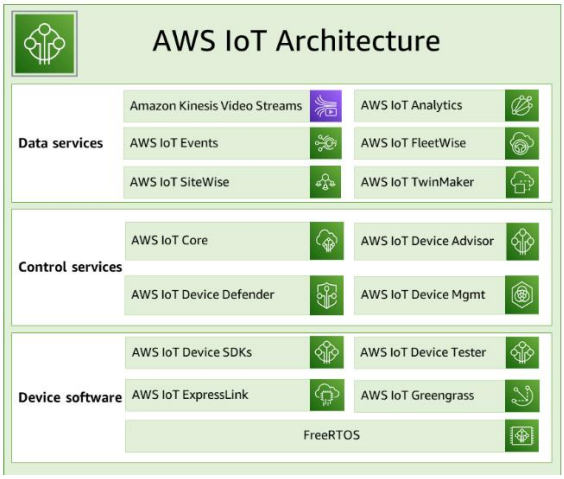

Рис. 2.2. Схема сервісів *AWS IoT*

Amazon Web Services був запущений в 2002 році та спочатку надавав послуги зберігання та обчислень в хмарному середовищі. Протягом наступних років платформа розширювалась, додаючи нові сервіси та функціонал.

З появою концепції *IoT* та зростанням популярності підключених пристроїв, бізнеси стали шукати рішення для ефективного управління та обробки даних великого обсягу, які генеруються *IoT*-пристроями.

У 2015 році Amazon анонсувала *AWS IoT* як рішення для підтримки Інтернету речей. Платформа була спроектована для вирішення проблем обробки великого обсягу даних, забезпечуючи безпеку та швидкість обміну інформацією між підключеними пристроями та хмарним сервісом.

AWS IoT постійно оновлюється та розширює свої можливості. Додавання нових сервісів та інтеграція з іншими сервісами *AWS* дозволяють користувачам створювати складні та потужні системи *IoT*.

AWS IoT використовується у багатьох галузях, таких як промисловість, охорона здоров'я, сільське господарство та інші. Його гнучкість дозволяє адаптувати рішення до різних вимог та випадків використання.

AWS IoT співпрацює з розробниками та партнерами для створення нових рішень та надання підтримки для розширених сценаріїв використання.

Історія створення *AWS IoT* тісно пов'язана з еволюцією *Amazon Web Services* та попитом на розвиток ефективних інструментів для роботи з Інтернетом речей.

AWS *IoT* надає інструменти для збору, зберігання та аналізу даних з пристроїв *IoT*. Вона інтегрується з іншими сервісами *AWS*, такими як *AWS Lambda*, *Amazon Kinesis*, *Amazon S3* та *Amazon DynamoDB*, що дозволяє проводити потужну аналітику, машинне навчання та створювати складні сценарії обробки даних.

AWS IoT також підтримує можливості зібрання, аналізу та використання даних у реальному часі для прийняття швидких та ефективних рішень [5].

AWS IoT надає широкий спектр інструментів та сервісів для забезпечення безпеки пристроїв, комунікації та збереження даних.

Вона використовує механізми шифрування для захисту даних від несанкціонованого доступу та надає можливість керування правами доступу до пристроїв та ресурсів.

AWS IoT також відповідає високим стандартам безпеки, включаючи відповідні сертифікації та аудити, що дозволяє користувачам максимально захистити свої *IoT*-рішення.

AWS IoT безпосередньо інтегрується з іншими сервісами й екосистемою *AWS*, такими як *Amazon S3, Amazon Machine Learning*, *AWS Lambda* та інші. Це дозволяє розробникам ефективно використовувати існуючі рішення та сервіси *AWS* для розширення можливостей своїх *IoT*-додатків.

З переваг *AWS IoT* можна відмітити широкі можливості для збору, аналізу та управління даними *IoT*, високу маштабованість та продуктивність системи та гнучкість інтеграції з іншими сервісами та інфраструктурою *AWS*.

До недоліків *AWS IoT* можна віднести складність налаштування та використання для новачків без попереднього досвіду з *AWS*, а ще не виключені додаткові витрати, які не включені у базові плани підписки.

Загалом, *Amazon Web Services (AWS) IoT* є потужною та гнучкою платформою для розробки та впровадження рішень Інтернету речей.

Вона пропонує широкий набір функцій, масштабованість та високу безпеку, що робить її привабливим вибором для побудови *IoT*-рішень в різних галузях та застосуваннях.

Microsoft Azure IoT Suite - це інтегрована платформа для розробки та управління рішеннями *IoT*. Вона надає засоби для збору даних, аналітики, безпеки та масштабування системи.

Azure IoT Suite надає багатий набір сервісів, які допомагають в розробці, впровадженні та управлінні *IoT*-рішеннями.

Azure IoT Suite також надає можливість використання інших сервісів *Azure*, таких як *Azure Functions, Azure Machine Learning і Azure Stream Analytics* для розширення функціональності *IoT*-рішень.

Azure IoT Suite підтримує різні протоколи комунікації, такі як *MQTT, HTTP* і *AMQP*, що дозволяє підключати різноманітні типи пристроїв до платформи.

Вона також підтримує різні платформи розробки, включаючи *Windows, Linux* та мікроконтролери, що забезпечує гнучкість вибору для розробників (рис. 2.3).

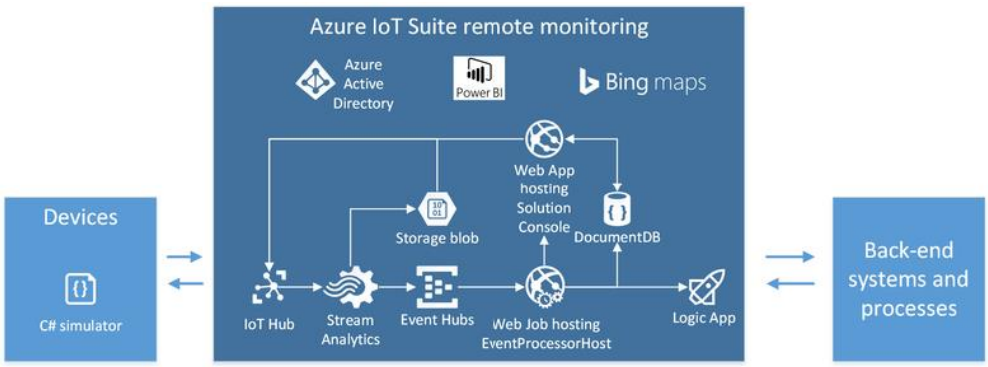

Рис. 2.3. Схема архітектури Azure IoT Suite

Azure IoT Suite пропонує широкі можливості для захисту даних та забезпечення конфіденційності. Вона використовує механізми шифрування для захисту даних під час передачі та зберігання.

Крім того, платформа пропонує інтеграцію з *Azure Active Directory* для керування правами доступу до пристроїв та ресурсів.

Microsoft також активно займається забезпеченням відповідності з регуляторними стандартами та надає сертифікації, які доводять відповідність платформи вимогам безпеки [6].

Azure IoT Suite надає інструменти для аналізу та використання даних з *IoT*-пристроїв. Вона використовує вбудовані аналітичні сервіси, такі як Azure *Stream Analytics* та *Azure Machine Learning*, для виявлення трендів, прогнозування та прийняття рішень на основі даних. З

а допомогою штучного інтелекту, *Azure IoT Suite* може автоматично аналізувати великі обсяги даних та здійснювати прогнозування для покращення ефективності та прийняття рішень.

Azure IoT Suite безпосередньо інтегрується з іншими сервісами та рішеннями *Microsoft*, такими як *Azure Cloud Services, Power BI, Dynamics* 365 та багатьма іншими.

Це забезпечує максимальну сумісність з існуючими системами та дозволяє використовувати потужність інших рішень *Microsoft* для розширення можливостей *IoT*-рішень.

До переваг платформи *Azure IoT Suite* можна включити інтегровану екосистему хмарних послуг, що дозволяє зручно розробляти та масштабувати системи *IoT*, високий рівень безпеки та захищеності даних та широкі можливості для інтеграції з іншими продуктами та послугами Microsoft.

Серед недоліків *Azure IoT Suite* можна виділити обмеженість деяких функцій для певних планів підписки та потреба у глибокому розумінні хмарних технологій *Microsoft*.

Загалом, Microsoft *Azure IoT Suite* є потужною платформою для розробки та впровадження Інтернету речей.

Вона пропонує широкий спектр сервісів, гнучкість вибору протоколів та платформ розробки, високу безпеку та інтеграцію з екосистемою *Microsoft*. Це робить її привабливим вибором для створення інноваційних *IoT*-рішень у різних галузях та застосуваннях.

Google Cloud IoT Core - це платформа *Google* надає засоби для підключення, керування та моніторингу пристроїв *IoT*. Вона використовує потужність хмарних послуг *Google* для збору та аналізу даних *IoT*.

Зародження *Google Cloud* припадає на 2008 рік, коли *Google* анонсувала свою хмарну платформу. У перші роки це був набір різних послуг, спрямованих на розробників та підприємства.

Поступово *Google* розширювала свої хмарні послуги, включаючи інфраструктуру, обчислювальні ресурси, бази даних та аналітичні інструменти.

З появою Інтернету речей та ростом популярності підключених пристроїв *Google* визнавала потребу в ефективному та масштабованому рішенні для *IoT*.

У 2017 році *Google* оголосила про запуск *Google Cloud IoT Core*. Це рішення призначене для керування пристроями *IoT*, забезпечення безпеки та швидкого обміну даними.

Google Cloud IoT Core надає можливості для реєстрації пристроїв, керування ними, обмін даними та інтеграцію з іншими сервісами *Google Cloud* (рис. 2.4).

Ключовою особливістю є інтеграція з іншими сервісами *Google Cloud*, такими як *BigQuery, Dataflow, Pub/Sub*, що дозволяє розробникам створювати повноцінні та розширені рішення для *IoT*.

Google Cloud IoT Core використовується в різних галузях, включаючи промисловість, логістику, сільське господарство та інші.

Історія *Google Cloud IoT Core* тісно пов'язана з загальним розвитком екосистеми *Google Cloud* та потребами ринку в ефективних інструментах для роботи з Інтернетом речей.

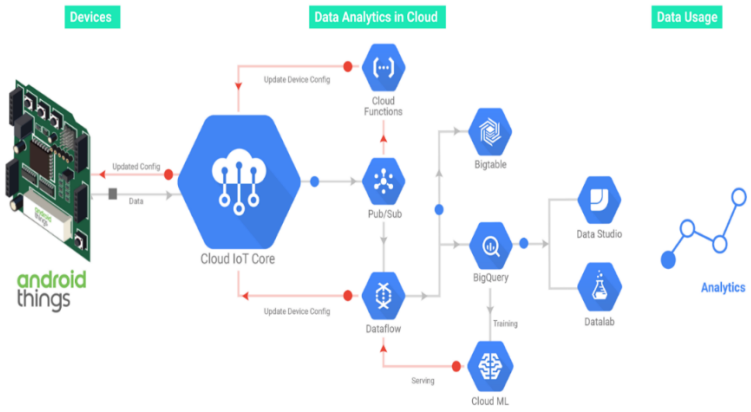

Рис 2.4. Схема *Google Cloud IoT Core*

Google Cloud IoT надає розширений набір сервісів для побудови *IoT*-рішень. Це включає інструменти для реєстрації та керування пристроями, збору, аналізу та використання даних, а також інструменти для візуалізації та моніторингу.

Google Cloud IoT також надає можливість використовувати інші сервіси *Google Cloud*, такі як *Google Cloud Pub/Sub, Google BigQuery* та інші, для розширення функціональності *IoT*-рішень.

Google Cloud IoT забезпечує масштабованість для підключення та обробки великого обсягу *IoT*-даних. Вона дозволяє легко масштабувати рішення в залежності від потреб, а також підтримує різні типи IoT-пристроїв і протоколів комунікації. Це дає розробникам велику гнучкість у виборі технологій та пристроїв для їх проектів.

Google Cloud IoT глибоко інтегрована з іншими сервісами та інструментами *Google*, такими як *Google Cloud Storage*, *Google Cloud Machine Learning* та *Google Cloud AI*. Це дозволяє використовувати потужність штучного інтелекту та машинного навчання для аналізу та використання *IoT*-даних.

Крім того, *Google Cloud IoT* пропонує інтеграцію з іншими платформами *Google*, такими як *Google Assistant, Google Maps* та *Google Cloud Functions*, для розширення функціональності *IoT*-рішень [7].

Google Cloud IoT забезпечує високий рівень безпеки для *IoT*-рішень. Вона використовує різні заходи безпеки, такі як шифрування даних, ідентифікацію та автентифікацію пристроїв, контроль доступу та моніторинг безпеки. *Google Cloud IoT* також відповідає найвищим стандартам безпеки та конфіденційності даних.

Google Cloud IoT побудована на глобальній масштабованій інфраструктурі *Google*, що забезпечує високу доступність та надійність сервісу. Це означає, що *IoT*-рішення, побудовані на *Google Cloud IoT*, можуть працювати в будь-якій точці світу з високою продуктивністю та надійністю.

Загалом, *Google Cloud IoT* є потужною платформою Інтернету речей, яка надає широкі можливості для розробки та управління *IoT*-рішеннями. Вона відрізняється своїм розширеним набором сервісів та масштабованістю.

Blynk - це платформа для розробки мобільних додатків для *Internet of Things* (*IoT*). Вона дозволяє створювати і керувати власними мобільними додатками, які

взаємодіють з мікроконтролерами та одноплатними комп'ютерами для віддаленого моніторингу та управління різними пристроями.

Blynk пропонує простий та зручний інтерфейс для створення додатків без необхідності програмування з нуля.

Платформа підтримує велику кількість мікроконтролерів та одноплатних комп'ютерів, таких як *Arduino, Raspberry Pi, ESP8266, ESP32*, і багато інших.

Користувачі можуть налаштовувати зв'язки та умови використовуючи візуальний інтерфейс *Blynk.*

Платформа має готові шаблони для швидкого старту та створення проектів. *Blynk* має активну спільноту користувачів, яка допомагає вирішувати питання та обмінюватися досвідом.

Blynk може інтегруватися з різними хмарними сервісами для зберігання даних та отримання сповіщень.

Blynk є популярним вибором для розробників, що працюють з *IoT*, які шукають ефективний спосіб взаємодії з своїми пристроями через мобільні додатки.

Blynk пропонує дуже зручний та легкий у використанні інтерфейс для створення мобільних додатків для *IoT*-пристроїв. Платформа підтримує широкий спектр мікроконтролерів та одноплатних комп'ютерів, що дозволяє вибрати пристрій, який найкраще відповідає конкретному проекту. *Blynk* надає готові шаблони для різних проектів, що значно прискорює процес розробки та розгортання.

Використання концепції "енергії" дозволяє керувати обсягом ресурсів на безкоштовному тарифі, а автентифікація гарантує безпеку обміну даними.

Blynk має велику та активну спільноту користувачів, яка сприяє обміну досвідом та рішенням проблем. Користувачі можуть використовувати власні бібліотеки для розширення можливостей *Blynk*.

Проте безкоштовний тариф має обмеження за кількістю енергії, що може бути використана. Це може бути обмеженням для складних проектів.

Платформа вимагає постійного доступу до Інтернету, що може бути обмеженням для деяких застосувань. Деякі функції та розширені можливості вимагають плату за енергію, що може збільшити витрати для великих проектів.

Blynk часто використовується для створення систем домашньої автоматизації, де користувачі можуть віддалено керувати своїми пристроями. *Blynk* ідеально підходить для проектів, які використовують мікроконтролери *Arduino* або одноплатні комп'ютери, такі як Raspberry Pi.

Завдяки своєму інтуїтивному інтерфейсу, *Blynk* добре підходить для навчання та прототипування нових *IoT*-проектів.

Blynk використовується для створення систем віддаленого моніторингу різних параметрів, таких як температура чи вологість. Платформа застосовується в освітніх цілях для навчання студентів основам *IoT* та програмування.

Thinger.io - це хмарна платформа для розробки проектів *Internet of Things*. Вона дозволяє створювати, моніторити та керувати різними пристроями *IoT* за допомогою веб-інтерфейсу. Thinger.io надає інструменти для збору, візуалізації та аналізу даних, а також взаємодії з пристроями через *API*.

Thinger.io підтримує різні платформи та пристрої, такі як *Arduino, ESP8266, Raspberry Pi*, та інші. Вона забезпечує легку інтеграцію та підключення пристроїв до хмари.

Платформа дозволяє користувачам налаштовувати параметри пристроїв та проекти відповідно до їхніх потреб.

Thinger.io надає засоби для створення графіків, діаграм та інших візуальних елементів для відображення даних в реальному часі. *Thinger.io* має активну спільноту користувачів, яка допомагає вирішувати питання та обмінюватися досвідом.

Платформа надає можливості для забезпечення безпеки підключення та обміну даними між пристроями та хмарою.

Thinger.io може інтегруватися з іншими хмарними сервісами та додатками для розширення можливостей проекту.

Thinger.io відомий своєю простотою в налаштуванні та використанні. Він призначений для швидкої розробки та простого впровадження проектів *IoT*.

Платформа сумісна з різними мікроконтролерами та пристроями, такими як *Arduino, ESP8266, Raspberry Pi* та інші. Thinger.io надає потужні засоби для

візуалізації даних, дозволяючи користувачам створювати графіки та діаграми для аналізу зібраних інформацій.

Користувачі можуть легко налаштовувати параметри пристроїв та визначати правила для обробки даних. Thinger.io може інтегруватися з різними хмарними сервісами, розширюючи функціональність проектів.

Проте безкоштовний тариф може бути обмеженим у функціональності та обсязі даних. Оскільки *Thinger.io* є хмарною платформою, вона потребує стабільного інтернет-з'єднання для роботи.

Платні тарифи можуть бути високими для великих обсягів даних та інтенсивного використання ресурсів.

Thinger.io використовується для створення систем управління освітленням, опаленням, безпекою та іншими аспектами домашнього середовища.

Користувачі використовують *Thinger.io* для створення проектів з моніторингу різних параметрів навколишнього середовища, таких як температура, вологість та якість повітря.

У промислових застосуваннях Thinger.io використовується для моніторингу та управління різними процесами та обладнанням.

Thinger.io використовується в освітніх закладах для навчання студентів розробці проектів *IoT* та моніторингу даних.

Розробники використовують *Thinger.io* для швидкої розробки та тестування прототипів *IoT*-пристроїв та додатків.

Thinger.io є популярним вибором серед розробників, які шукають простий та потужний інструмент для роботи з *IoT*-проектами.

Cayenne - це хмарна платформа для Інтернету речей, розроблена спеціально для спрощення розробки та управління проектами *IoT*.

Вона надає інструменти для створення з'єднань між різними пристроями та додатками, візуалізації даних, а також дистанційного моніторингу та управління підключеними пристроями через Інтернет.

Cayenne підтримує широкий спектр пристроїв, таких як мікроконтролери *Arduino, Raspberry Pi, ESP8266* та інші.

Платформа працює з різними протоколами зв'язку, такими як *MQTT, MQTT API, REST API*, що робить її гнучкою для різних вимог проектів.

Cayenne дозволяє користувачам створювати власні панелі та графіки для візуалізації зібраних даних.

За допомогою *Cayenne* можна створювати правила для автоматизації взаємодії між пристроями в залежності від умов. Користувачі можуть в реальному часі відслідковувати стан своїх підключених пристроїв і взаємодіяти з ними віддалено.

Cayenne надає можливість спільної роботи над проектами, що робить її ідеальним інструментом для команд розробників.

Cayenne відома своєю простотою в налаштуванні та використанні, що робить її ідеальним вибором для початківців в області Інтернету речей. *Cayenne* надає користувачам інтуїтивно зрозумілі засоби для візуалізації даних та створення графіків без необхідності використання коду.

Платформа дозволяє легко ділитися даними та станом підключених пристроїв через соціальні мережі.

Cayenne дозволяє користувачам створювати правила для автоматизації взаємодії між пристроями в залежності від умов. Платформа дозволяє розробникам спільно працювати над проектами, обмінюючись доступом та управлінням.

Проте для деяких складних *IoT*-проектів, які вимагають глибокого програмування та специфічних функцій, може бути обмеженою. Використання *Cayenne* передбачає залежність від хмарних сервісів, що може бути недоцільним для деяких користувачів у зв'язку з питаннями конфіденційності та безпеки.

Деякі розширені функції та можливості доступні лише на платних планах, що може становити обмеження для бюджетних проектів.

Cayenne потребує Інтернет-з'єднання для нормальної роботи, що може створювати проблеми в умовах обмеженої або відсутньої мережі.

Для досвідчених розробників, які шукають повний контроль та гнучкість, інші більш програмістично орієнтовані платформи можуть бути більш підходящими.

Cayenne використовується для створення різноманітних систем управління освітленням, опаленням, системами безпеки в домашньому середовищі.

Користувачі можуть використовувати Cayenne для створення пристроїв для моніторингу погодних умов та відображення цих даних в реальному часі. У промислових застосуваннях Cayenne використовується для моніторингу та управління різними процесами та обладнанням.

Cayenne використовується в навчальних закладах для вивчення основ IoT та розробки простих проектів.

Розробники використовують Cayenne для швидкої розробки та тестування прототипів IoT-пристроїв та додатків.

2.3.2. Функції, можливості та результати впровадження різних засобів управління та моніторингу *IoT*

Огляд конкретних прикладів застосування засобів моніторингу та управління в різних галузях Інтернету речей (*IoT*) показує, як ця технологія може впливати на різні сфери життя та покращувати ефективність та комфорт у різних сегментах. В сфері смарт-дому засоби моніторингу та управління *IoT* можуть бути використані для автоматизації та контролю різних пристроїв у будинку.

Наприклад, з допомогою *IoT*-рішень можна віддалено керувати освітленням, опаленням, кондиціонуванням повітря, системами безпеки та іншими пристроями в будинку.

Це дозволяє забезпечити зручність, енергоефективність та безпеку у житлових приміщеннях.

У контексті смарт-міста засоби моніторингу та управління *IoT* можуть бути використані для оптимізації управління різними інфраструктурними системами.

Наприклад, впровадження *IoT* може допомогти відстежувати рух транспорту, контролювати освітлення на вулицях, оптимізувати сміттєві системи, управляти енергозберігаючими системами в будівлях та іншими аспектами, що покращують якість життя мешканців.

У промисловості 4.0 засоби моніторингу та управління *IoT* можуть використовуватися для збору та аналізу даних з обладнання, виробничих ліній та процесів.

Це дозволяє забезпечити прогнозування відмов, оптимізувати виробничі процеси, зменшити витрати на обслуговування та підвищити ефективність.

Наприклад, *IoT*-пристрої можуть моніторити стан обладнання, вимірювати параметри виробництва та автоматично вживати заходи у разі необхідності.

В галузі здоров'я та медицини засоби моніторингу та управління *IoT* відкривають широкі можливості для поліпшення надання медичних послуг та дистанційного моніторингу пацієнтів.

IoT-пристрої можуть бути використані для вимірювання показників здоров'я, передачі цих даних до медичних центрів та в реальному часі спостерігати за станом пацієнтів.

В сільському господарстві *IoT* може бути використано для моніторингу та управління різними аспектами сільськогосподарського виробництва. Наприклад, засоби моніторингу *IoT* можуть контролювати рівень вологості ґрунту, температуру, якість ґрунту, стан рослин тощо.

Це дозволяє аграрним виробникам точніше визначати потреби рослин у воді та добривах, ефективно керувати поливом та збільшувати врожайність.

Загалом, засоби моніторингу та управління в *IoT*-рішеннях знайшли своє застосування у багатьох галузях, що відкриває нові можливості для покращення продуктивності, ефективності та комфорту у різних сферах життя.

IoT-технології продовжують розвиватись, привносячи інновації та перетворюючи наш спосіб життя [8].

Висновки за розділом

Було проведено детальний аналіз засобів моніторингу та управління в контексті Інтернету речей (*IoT*). Були розглянуті основні характеристики таких засобів, включаючи сенсори, актуатори, протоколи комунікації, платформи та програмне забезпечення.

У результаті аналізу було виявлено, що сенсори є ключовими складовими засобів моніторингу *IoT*, оскільки вони здатні збирати різноманітні дані з фізичних

об'єктів. Актуатори, з свого боку, відповідають за виконання дій на основі отриманих даних, що робить їх необхідними для здійснення управління елементами IoT.

Протоколи комунікації відіграють важливу роль у забезпеченні передачі даних між різними пристроями в мережі *IoT*. Вони забезпечують надійну та безпечну комунікацію, дозволяючи засобам моніторингу та управління взаємодіяти між собою.

Платформи *IoT* та програмне забезпечення грають важливу роль у спрощенні розробки, впровадженні та управлінні *IoT*-рішеннями. Вони надають інструменти для збору, зберігання та аналізу даних, а також можливості управління та взаємодії з пристроями *IoT*.

Аналізуючи переваги використання засобів моніторингу та управління в контексті *IoT*, було виявлено, що вони сприяють автоматизації, оптимізації та покращенню різних процесів.

Вони забезпечують збір та аналіз великого обсягу даних, що дозволяє здійснювати обґрунтоване прийняття рішень. Засоби моніторингу та управління також сприяють виявленню та усуненню неполадок, що покращує надійність та ефективність систем.

Однак, використання засобів моніторингу та управління *IoT* не звільняє від викликів та проблем.

Зокрема, виникають питання щодо приватності та безпеки зібраних даних, необхідності стандартизації та інтеграції різних пристроїв та платформ, а також потреби у високій швидкості передачі даних та надійному з'єднанні.

Дослідження існуючих систем моніторингу та управління *IoT* показало, що на ринку існує ряд платформ, які надають розширені можливості для створення та управління *IoT*-рішеннями. При цьому, кожна платформа має свої особливості та переваги.

Також було проведено аналіз платних і безкоштовних платформ *IoT*. Вибір між безкоштовними та платними платформами Інтернету речей залежить від конкретних потреб і завдань, а також від ресурсів, доступних для проекту.

Безкоштовні платформи часто обмежують кількість підключених пристроїв та обсяг передачі даних.

Якщо проект невеликий і не потребує багато ресурсів, можливо, безкоштовна платформа буде достатньою. Для великих та амбіційних проектів, де потрібно більше ресурсів, платна платформа може бути оптимальним вибором.

У платних платформах часто є великі команди для надання підтримки, що може бути критичним у випадках проблем або запитань. Безкоштовні платформи можуть надавати обмежену підтримку або покладатися на спільноту для допомоги.

Застосування засобів моніторингу та управління в різних галузях відкриває нові можливості для покращення якості життя людей, підвищення продуктивності та ефективності процесів.

Враховуючи особливості кожної галузі, необхідно вибрати відповідні засоби моніторингу та управління, які найкраще відповідають потребам та вимогам конкретного сценарію застосування.

РОЗДІЛ 3
РОЗРОБКА СИСТЕМИ МОНІТОРИНГУ ТА УПРАВЛІННЯ ІНТЕРНЕТУ РЕЧЕЙ

3.1. Платформа *IoT ThingSpeak*

ThingSpeak - це веб-платформа для Інтернету речей (*IoT*), яка входить до екосистеми *MathWorks*, що відома своїм математичним програмним забезпеченням, зокрема *MATLAB*. ThingSpeak спрощує процес збору, аналізу та візуалізації даних з пристроїв *IoT*, роблячи його доступним для широкого кола користувачів, включаючи початківців та професіоналів (рис. 3.1).

Рис. 3.1. Платформа *ThingSpeak*

Історія ThingSpeak розпочалася в 2010 році, коли експерти з галузі розробки програмного забезпечення та Інтернету речей вирішили створити відкритий та доступний інструмент для збору та обробки даних. Ціль була в тому, щоб зробити IoT доступним для широкого кола користувачів, незалежно від їхнього рівня технічних знань.

Платформа розроблялася як відкритий проект, що надавало можливість спільноті розробників вносити свій внесок у її розвиток. Перші версії ThingSpeak були

спрямовані на створення інтерфейсу для спрощеного збору даних з IoT-пристроїв і подальшої їхньої обробки.

З часом ThingSpeak отримала популярність серед розробників та інженерів, які шукали ефективні засоби для роботи з даними в галузі IoT. Постійні оновлення та розширення функціоналу робили платформу більш гнучкою та потужною.

Сьогодні ThingSpeak використовується в різних галузях, включаючи моніторинг погоди, відстеження даних з сенсорів, розумний дім, промисловість та інші області, де важлива обробка та аналіз даних з IoT-пристроїв. Історія ThingSpeak - це історія про те, як створення доступних та простих у використанні інструментів може значно полегшити розвиток Інтернету речей.

Користувачі можуть легко створювати "канали" для кожного зі своїх проектів чи пристроїв. Кожен канал служить як місце для зберігання, візуалізації та обробки даних.

ThingSpeak надає зручні інструменти для візуалізації даних, такі як графіки, графіки та карти. Це дозволяє користувачам швидко спостерігати за змінами в їхніх даних. Платформа дозволяє користувачам виконувати базовий аналіз та обробку даних безпосередньо в системі. Це може включати у себе середні значення, медіани, фільтрацію та інші операції. Оскільки *ThingSpeak* належить *MathWorks*, він інтегрований з *MATLAB*. Це дає можливість використовувати складні алгоритми аналізу та моделювання для обробки даних [9].

ThingSpeak має активну спільноту користувачів, яка обмінюється проектами, ідеями та вирішенням проблем. Платформа підтримує різні протоколи зв'язку, такі як *HTTP, MQTT, ThingSpeak API*, що дозволяє інтегрувати її з різними пристроями та сервісами.

Також *ThingSpeak* може взаємодіяти з геолокаційними даними, що дозволяє створювати *IoT*-проекти, пов'язані із місцезнаходженням.

ThingSpeak може використовуватися для моніторингу та візуалізації погодних умов, збираючи дані з різних сенсорів. *IoT*-пристрої в смарт-домах можуть відправляти дані на *ThingSpeak* для аналізу та візуалізації. Використання для

вимірювання та аналізу споживаної енергії. Відстеження показників здоров'я та медичних даних через *IoT*-пристрої.

Звісно у такої платформи є свої недоліки та переваги. До переваг можна віднести безкоштовний рівень обслуговування. *ThingSpeak* має безкоштовний рівень обслуговування для невеликих проектів та експериментів. *ThingSpeak* призначений для широкого кола користувачів, включаючи початківців, і має інтуїтивний інтерфейс. *ThingSpeak* підтримує різні протоколи зв'язку, включаючи *HTTP*, *MQTT* та *ThingSpeak API*, що дозволяє легко інтегрувати різноманітні пристрої.

До недоліків платформи відносяться такі фактори як безкоштовний рівень обслуговування.

Безкоштовний рівень обслуговування обмежений, і для більших проектів може знадобитися перехід на платний план. У порівнянні з деякими іншими платформами *ThingSpeak* може мати менше функціоналу та гнучкості для складних сценаріїв [10].

Загалом, *ThingSpeak* - це добрий вибір для швидкого старту та простих проектів Інтернету речей. Якщо у користувача є конкретні потреби або ви шукаєте більше функціоналу, вам може бути корисно розглянути також інші платформи.

3.2. Архітектура та пристрої

ThingSpeak працює на архітектурі клієнт-сервер. Це означає, що *ThingSpeak* має центральний сервер, який обробляє дані, надсилані клієнтами. Клієнти, у цьому випадку, це пристрої або додатки, які відправляють дані на сервер *ThingSpeak* для зберігання, обробки та візуалізації.

ThingSpeak є гнучкою платформою та може працювати з різними пристроями, які можуть надсилати дані через Інтернет.

Наприклад *Arduino,* а саме, використання *Wi-Fi* модуля *ESP8266* або *ESP32* для передачі даних на ThingSpeak. Використання *Raspberry Pi* з модулем *Wi-Fi* для надсилання данних.

Датчики температури, вологості, освітлення тощо, які підключаються до мікроконтролерів і відправляють дані на *ThingSpeak*. Розробка мобільних додатків,

які можуть відправляти дані на *ThingSpeak*. Це може бути корисно для моніторингу чого-небудь за допомогою смартфона.

Та інші пристрої з можливістю підключення до Інтернету, які можуть використовувати *HTTP* або *MQTT* для надсилання даних на *ThingSpeak*.

ThingSpeak надає *API* та інструменти для спрощення інтеграції з різноманітними пристроями та додатками, що робить її універсальною для застосувань в Інтернеті речей.

Інтернет речей (*IoT*) стає дедалі більш важливим аспектом нашого технологічного ландшафту, а мікропроцесори відіграють ключову роль у забезпеченні функціональності *IoT*-пристроїв. *ESP8266* та його розвинена версія *ESP32* вирізняються вбудованим *Wi-Fi*, що дозволяє *IoT*-пристроям спілкуватися з Інтернетом безпосередньо. Їх простота використання та низьке енергоспоживання роблять їх популярними серед розробників. Платформа *Arduino*, особливо на основі мікроконтролера *ATmega*, надає просте та доступне середовище для розробки *IoT*-пристроїв. Її широкий вибір бібліотек та дружня спільнота сприяють швидкому розгортанню проектів. *Raspberry Pi*, хоч і не є мікроконтролером, але скоріше невеликим комп'ютером, також використовується в *IoT* завданнях через свій потужний процесор та здатність обробляти великий обсяг даних. Він часто використовується в проектах, де потрібно обробляти складні алгоритми або запускати серверні застосунки.

ARM Cortex-M серія мікропроцесорів, таких як *Cortex-M0*, *Cortex-M3*, та *Cortex-M4*, визначаються своєю енергоефективністю та високою продуктивністю. Вони часто використовуються в *IoT*-пристроях, де критично важливе споживання енергії.

Particle Photon використовується для підключення пристроїв до Інтернету через *Wi-Fi* та має вбудований мікроконтролер. Він спрощує розробку *IoT*-проектів через свою інтеграцію з платформою *Particle Cloud*. Мікропроцесори *Texas Instruments*, такі як *MSP430* та *CC3200*, роблять акцент на низькому споживанні енергії, що робить їх ідеальними для вбудованих систем, які працюють в умовах обмеженого живлення. *Nordic Semiconductor nRF* серія мікроконтролерів відзначається високою

продуктивністю та вбудованою підтримкою *Bluetooth*, що робить їх популярними для *IoT*-пристроїв, які вимагають безпровідного зв'язку.

Ці мікропроцесори відіграють ключову роль у реалізації *IoT*, забезпечуючи зв'язок, обробку даних та взаємодію з оточуючим середовищем. Розробники вибирають конкретний мікропроцесор в залежності від потреб свого проекту, забезпечуючи оптимальну комбінацію продуктивності та функціональності.

Arduino - це відкрита платформа для розробки електроніки та програмування мікроконтролерів. Вона складається з аппаратної та програмної частин, які співпрацюють для створення простих або складних електронних проектів. Основний компонент - мікроконтролер, який виконує програми, написані на мові програмування *Arduino*. *Arduino* застосовується в різних областях, таких як робототехніка, автоматизація, мистецтво, *Internet of Things (IoT)* та багато інших. Для прикладу практичного завдання були обрані мікроконтролери *ESP8266-12* та *ESP8266-12E* [11].

ESP8266-12 - це модуль *Wi-Fi*, який використовується для забезпечення можливостей бездротового зв'язку між електронічними пристроями. Основою цього модуля є мікроконтролер *ESP8266*, який виготовляється компанією *Espressif Systems* (рис. 3.2).

Рис. 3.2. Модуль *ESP8266-12*

До основних характеристик *ESP8266-12* входять мікроконтролер *ESP8266*, який має достатньо потужності для виконання завдань зі збору і обробки даних та вбудований *Wi-Fi*, що робить його ідеальним для проектів Інтернету речей. Модуль має виводи GPIO, які можна використовувати для підключення до різних сенсорів, пристроїв введення-виведення та інших периферійних пристроїв. Деякі версії ESP8266-12 мають вбудовану антену для забезпечення кращого зв'язку по Wi-Fi. Відома як "NodeMCU," версія ESP8266-12 має розширену кількість GPIO портів

порівняно з базовим модулем ESP8266. ESP8266-12 використовується для бездротового підключення різних пристроїв до Інтернету, і він часто використовується в різних проектах IoT, де зручність бездротового зв'язку є ключовою.

За допомогою датчиків температури, вологості та барометра може збирати дані та відправляти їх на сервер для аналізу. Використовується для моніторингу якості повітря чи рівня шуму в приміщенні. Взаємодіє з різними пристроями в смарт-домі, керує освітленням, опаленням тощо. Інтегрується в системи безпеки для спостереження та виявлення небезпек. Застосовується для надсилання стану пристроїв та отримання команд від користувача [12].

ESP8266-12 - це відмінний вибір для тих, хто шукає доступний, потужний та легко програмований мікроконтролер для своїх проектів Інтернету речей. Його надійність та гнучкість роблять його ідеальним для великої кількості застосувань, від простих пристроїв збору даних до складних *IoT*-систем.

ESP8266-12E - це покращена версія модуля *Wi-Fi ESP8266*, виробленого компанією Espressif Systems.

Цей модуль включає в себе мікроконтролер ESP8266, але він має ряд покращень порівняно з базовою версією. Відомий своєю надійністю та потужністю, *ESP8266-12E* знаходить широке застосування серед розробників та ентузіастів електроніки.

До основних характеристик *ESP8266-12* входять вбудований мікроконтролер *ESP8266*, який забезпечує потужність для виконання завдань зі збору, обробки та передачі даних, вбудований *Wi-Fi*, як і в базовій версії, що робить його ідеальним для проектів Інтернету речей. *ESP8266-12E* має більше *GPIO* портів порівняно з базовою версією.

Це розширює можливості для підключення до різних пристроїв та сенсорів. Модуль підтримує різні протоколи зв'язку, такі як *SPI (Serial Peripheral Interface)*, *I2C (Inter-Integrated Circuit)*, та *UART (Universal Asynchronous Receiver-Transmitter),* що робить його більш гнучким для підключення до різних пристроїв. *ESP8266-12E* може мати вбудовану антену або підтримувати зовнішню антену для забезпечення кращого зв'язку.

ESP8266-12E може працювати в широкому діапазоні напруги, що дозволяє ефективно використовувати його в різних сценаріях.

Використовується для створення пристроїв, що моніторять та оптимізують споживання електроенергії та створення інтерактивних пристроїв, що взаємодіють із користувачем.

Застосовується в смарт-будинках для автоматизації освітлення, термостатів та інших пристроїв.

Збирає дані про якість повітря за допомогою відповідних сенсорів (рис. 3.3).

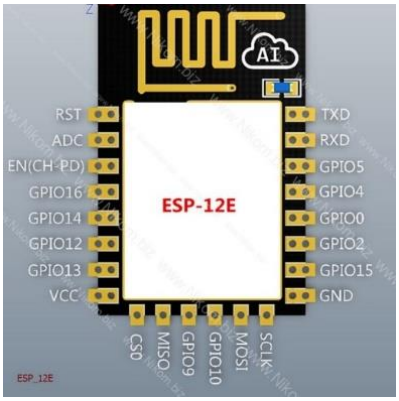

Рис. 3.3. Модуль *ESP8266-12E*

ESP8266-12E також легко програмується з використанням *Arduino IDE*, існують багато готових бібліотек та прикладів коду, що полегшує розробку.

ESP8266-12E залишається однією з найпопулярніших та потужних опцій для *IoT*-проектів. Його висока функціональність, зручність у використанні та надійність роблять його ідеальним вибором для широкого спектру застосувань в електроніці та *IoT*.

3.3. Використання *ThingSpeak* для зчитування та аналізу даних

В цій задача буде розглянуто процес збору та аналізу даних в новому каналі *ThingSpeak*.

Додатково, буде надано приклади коду для відправлення даних з різних джерел.

Користувач читає дані з загальнодоступного каналу *ThingSpeak 12397 - Weather Station* і записує їх у свій новий канал.

Створити канал:

1. Увійти у *ThingSpeak*, використовуючи облікові дані облікового запису *MathWorks*, або створіть новий обліковий запис.

2. Натиснути *Channels* > *My Channels* (рис. 3.4).

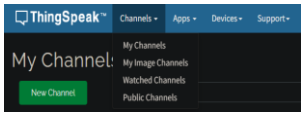

Рис. 3.4. Шлях до розділу "*My Channels*"

3. На сторінці каналів треба натиснути "Новий канал».

4. Поставити прапорці поруч із полями 1–3. Ввести ці значення налаштувань каналу (рис. 3.5):

Ім'я : *Dew Point Measurement.*

Поле 1: *Temperature (F).*

Поле 2: *Humidity.*

Поле 3: *Dew Point.*

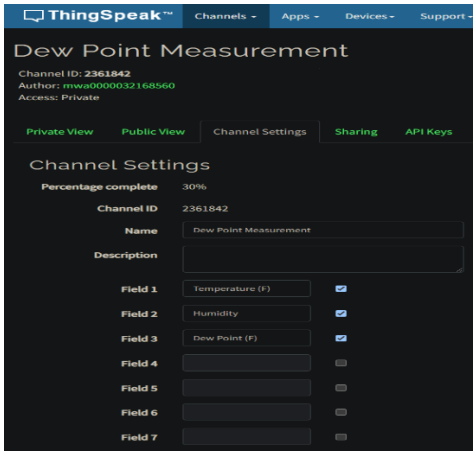

Рис. 3.5. Розділ "*Channel Settings*"

5. Натиснути «Зберегти канал» внизу налаштувань.

Private view – це вкладка, на якій відображається інформація про ваш канал, яку можете бачити лише ви.

Public view – це вкладка, яку можна використати, якщо користувач вирішить зробити свій канал загальнодоступним, щоб відобразити вибрані поля та візуалізацію каналу.

Channel settings - це вкладка, на якій відображаються всі параметри каналу, які ви встановили під час створення. На цій вкладці можна редагувати, очистити або видалити канал.

На вкладці *Sharing* показано параметри спільного доступу до каналу. Користувач може встановити канал як приватний, спільний для всіх (публічний) або спільний для певних користувачів.

API keys – це вкладка на якій відображаються ключі *API* каналу. Потрібно використовуввати клавіші для читання та запису на канал користувача.

Вкладка *Data Import/Export* дозволяє імпортувати та експортувати дані каналу (рис. 3.6).

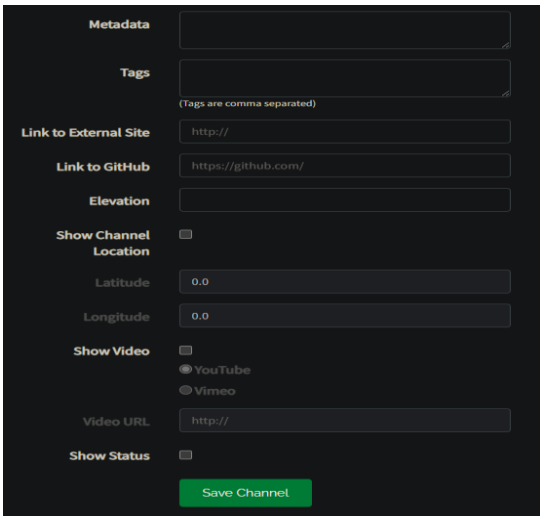

Рис. 3.6. Функція збереження каналу

У наступному етапі показано, як зчитувати дані про температуру та вологість з каналу *ThingSpeak 12397* , який збирає дані про погоду з пристрою *Arduino*. Користувач записує дані про температуру та вологість у свій канал вимірювання точки роси разом із розрахованими даними точки роси. Потім використовує *ThingSpeak*, щоб візуалізувати результати на своєму каналі.

Використати *MATLAB Analysis* для читання, обчислення та запису даних:

1. Перейти на вкладку «Програми» та натисніть «Аналіз *MATLAB*».

2. Натиснути “Створити”. Виберіть шаблон і натисніть «Створити» .

3. У полі “Ім'я” ввести “*Dew Point Calculation”*.

4. У полі “Код *MATLAB”* ввести наступні рядки коду.

Потрібно зберегти загальнодоступний ідентифікатор каналу метеостанції та ідентифікатор каналу вимірювання точки роси у змінних (рис. 3.7).

```
readChId = 12397;
writeChId = 2361842; %Your channel number
```

Рис. 3.7. Фрагмент коду збереження загальнодоступного ідентифікатора каналу

Зберегти ключ *Write API* у змінній (рис. 3.8).

```
writeKey = 'G8C9RKJGTNFLHT7G';
```

Рис. 3.8. Фрагмент коду збереження ключа *Write API* у змінній

Щоб знайти ідентифікатор каналу та написати ключ *API*, треба переглянути інформацію про канал на вкладці «Мої канали» (рис. 3.9).

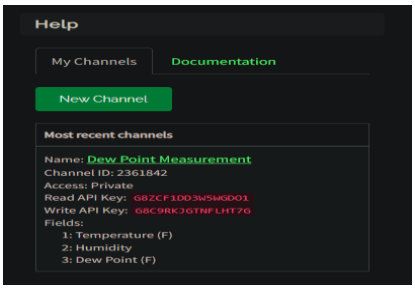

Рис. 3.9. Вкладка «Мої канали»

Зчитати останні 20 точок даних про температуру з часовими мітками та даними про вологість із загальнодоступного каналу метеостанції у змінні (рис. 3.10).

```
[temp,time] = thingSpeakRead(readChId,'Fields',4,'NumPoints',20);
humidity = thingSpeakRead(readChId,'Fields',3,'NumPoints',20);
```

Рис. 3.10. Фрагмент коду зчитування даних

Додати наступний код MATLAB, щоб обчислити точку роси за допомогою показників температури та вологості.

1. Перевести температуру з Фаренгейта в Цельсій.
2. Вказати константи для водяної пари (*b*) і барометричного тиску (*c*).
3. Обчислити точку роси в градусах Цельсія.
4. Перетворити результат назад у Фаренгейт (рис. 3.11).

```
tempC = (5/9)*(temp-32);
b = 17.62;
c = 243.5;
gamma = log(humidity/100) + b*tempC./(c+tempC);
dewPoint = c*gamma./(b-gamma)
dewPointF = (dewPoint*1.8) + 32;
```

Рис. 3.11. Код для обчислення точки роси за допомогою показників температури та вологості

5. Запис даних у канал вимірювання точки роси.

Цей код публікує всі доступні дані за одну операцію та містить правильні мітки часу (рис. 3.12).

```
thingSpeakWrite(writeChId,[temp,humidity,dewPointF],'Fields',[1,2,3],...
'TimeStamps',time,'Writekey',writeKey);
```

Рис. 3.12. Фрагмент коду з записом даних у канал

Зчитати вологість і температуру з полів 3 і 4 загальнодоступного каналу *WeatherStation* і записати ці дані в поля 2 і 1, відповідно, каналу вимірювання точки роси. Точка роси розраховується та записується в поле 3.

Повний блок коду тепер виглядає так (рис. 3.13):

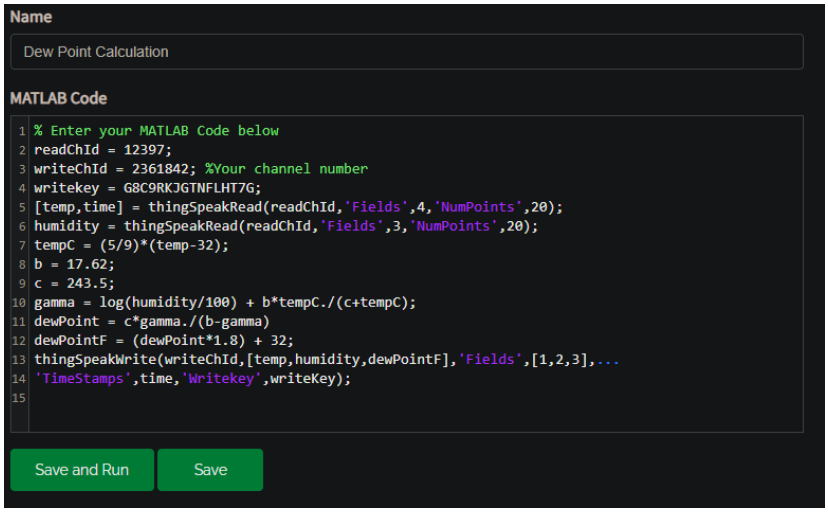

Рис. 3.13. Повний блок коду обчислення точки роси

6. Натиснути «*Save and Run*», щоб перевірити та обробити свій код. Будь-які помилки в коді вказуються в полі “Виведення”.

7. Щоб перевірити, чи успішно працював код, треба клацнути посилання каналу вимірювання точки роси на панелі інформації про канал (рис 3.14).

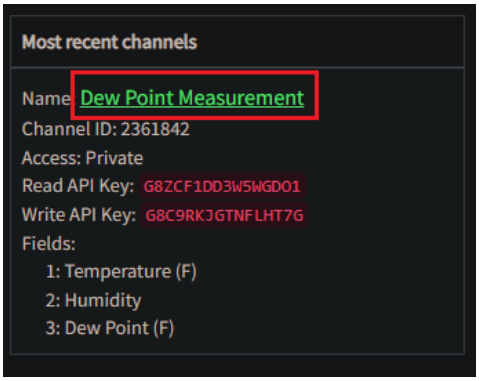

Рис. 3.14. Панель інформації про канал

Канал вимірювання точки роси тепер показує діаграми з даними каналу з кожного поля (рис. 3.15).

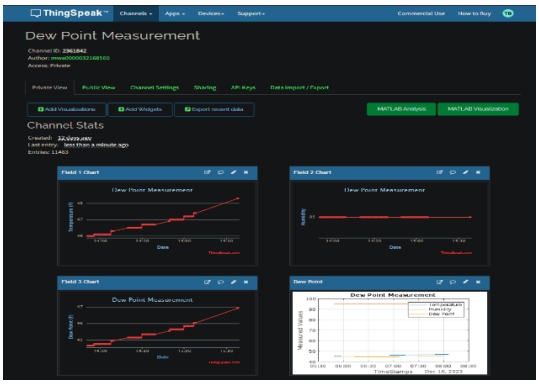

Рис. 3.15. Діаграми з даними каналу з кожного поля

Запланувати зчитування даних з метеостанції кожні 30 хвилин і обчислення точки роси.

1. Прокрутити донизу сторінки «Обчислення точки роси аналізу *MATLAB*». Клацніть *TimeControl*, щоб відкрити програму з попередньо вибраним аналізом *MATLAB* у полі «Дії» та обчисленням точки роси як кодом для виконання.

2. Назвати новий *TimeControl Dew Point TC*

3. Вибрати «Повторювані» в полі «Частота».

4. Вибрати «Хвилина» в полі «Повторення».

5. Вибрати 30 у полі “Кожні” — хвилин.

6. Залишити час початку за замовчуванням.

7. Переконатися, що дія є *MATLAB Analysis*, а код для виконання є *Dew Point Calculation*.

8. Клацнути Save *TimeControl* (рис. 3.16).

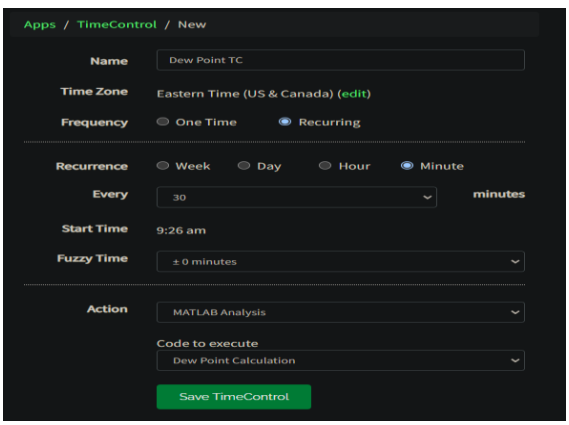

Рис. 3.16. Програма TimeControl

Використати програму *Visualizations MATLAB* для візуалізації виміряних даних точки роси, температури та вологості з каналу вимірювання точки роси. У цьому прикладі використовується функція *(MATLAB)* , щоб показати всі три точки даних в одній візуалізації.*plot*.

Перейти до «Програми» > «Візуалізації *MATLAB*» і натисніть «Створити» , щоб створити візуалізацію. Крім того, користувач може натиснути «Візуалізація *MATLAB*» у перегляді каналу вимірювання точки роси.

В цій задачі використовується функція *(MATLAB)* , щоб показати всі три точки даних в одній візуалізації.*plot* (рис. 3.17).

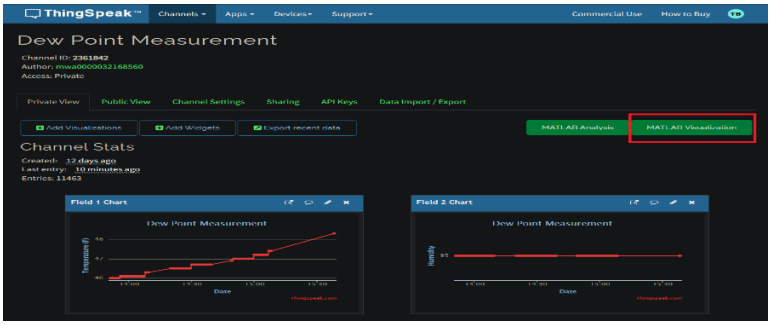

Рис. 3.17. Програма *Visualizations MATLAB*

1. Вибрати Custom шаблон і натисніть Створити.

2. Назвати візуалізацію "Точка роси"

3. Створити змінні для ідентифікатора каналу вимірювання точки роси та ключа *Read API*.

Замінити значення в коді ідентифікатором свого каналу та *Read API* ключ (рис. 3.18).

4. Прочитати дані з полів вашого каналу та отримайте останні 100 точок даних для:

1. Температура: від “Поля 1”.

2. Вологість: від “Поля 2”.

3. Точка роси: від “Поля 3”.

```
readChId = 2361842
readKey = 'G8ZCF1DD3W5WGDO1';
[dewPointData,timeStamps] = thingSpeakRead(readChId,'fields',[1,2,3],...
    'NumPoints',100,'ReadKey',readKey);
```

Рис. 3.18. Створення змінних для ідентифікатора каналу вимірювання

5. Побудувати дані з мітками x і y, заголовком і легендою (рис. 3.19).

```
plot(timeStamps,dewPointData);
xlabel('TimeStamps');
ylabel('Measured Values');
title('Dew Point Measurement');
legend({'Temperature','Humidity','Dew Point'});
grid on;
```

Рис. 3.19. Створення змінних для ідентифікатора каналу вимірювання

Код виглядатиме приблизно так (рис. 3.20):

MATLAB Code

```
readChId = 2361842
readKey = 'G8ZCF1DD3W5WGDO1';
[dewPointData,timeStamps] = thingSpeakRead(readChId,'fields',[1,2,3],...
    'NumPoints',100,'ReadKey',readKey);
plot(timeStamps,dewPointData);
xlabel('TimeStamps');
ylabel('Measured Values');
title('Dew Point Measurement');
legend({'Temperature','Humidity','Dew Point'});
grid on;
```

Save and Run Save*

Рис. 3.20. Повний блок коду візуалізації вимірювання точки роси

6. Натиснути Зберегти та запустити. Якщо код *MATLAB* не містить помилок, вихідний графік виглядатиме наступним чином (рис. 3.21):

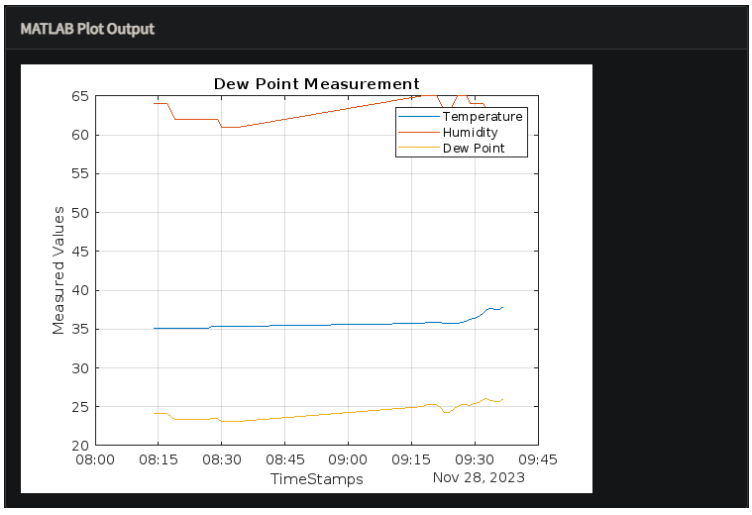

Рис. 3.21. Вихідний графік

На цьому етапі показано, як надіслати твіт, коли дані про точку роси перевищують 60°*F* за допомогою програми *ThingTweet* та *React*.

1. Перейти до “Програми” > *ThingTweet*.

2. На сторінці *ThingTweet* натиснути “Пов’язати обліковий запис *Twitter”*, щоб пов’язати свій обліковий запис *Twitter* з обліковим записом *ThingSpeak*.

3. Ввести ім’я користувача та пароль у *Twitter* і натисніть “Авторизувати Програму” (рис. 3.22).

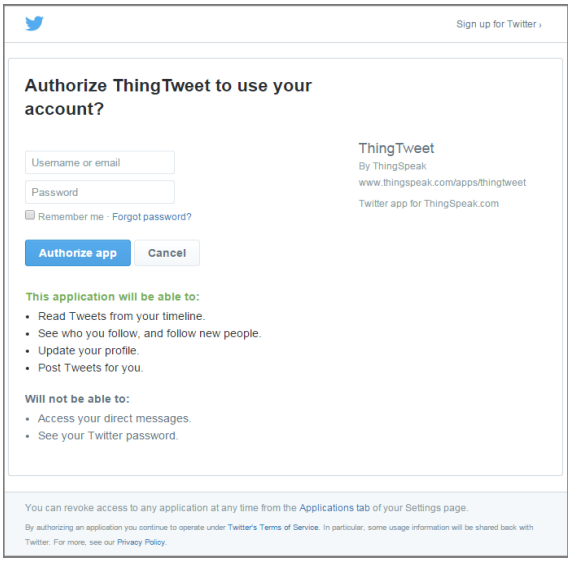

Рис. 3.22. Приєднання облікового запису *Twitter* до *ThingSpeak*

4. На сторінці Авторизація натисніть “Назад” до *ThingTweet.* Зверніть увагу, що ваш обліковий запис *Twitter* пов’язано з *ThingSpeak* (рис. 3.23).

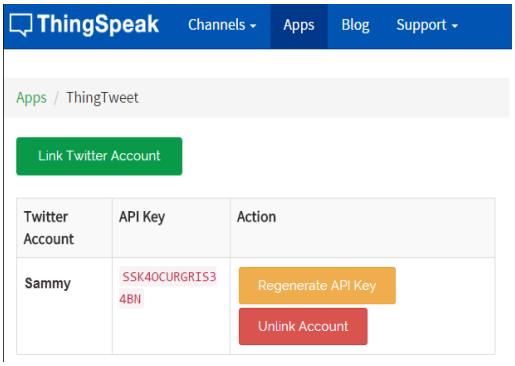

Рис. 3.23. Перевірка з’єднаності облікових записів

На цьому етапі показано, як надсилати твіт кожного разу, коли рівень точки роси перевищує 60°*F* перевіряючи канал кожні 10 хвилин.

1. Перейти до “Програми” > *React* і натисніть *New React*.

2. Назвати *React* "Твіт точки роси"

3. Установити тип умови на Числовий.

4. Установити частоту тестування на Кожні 10 хвилин.

5. Встановити умову для запуску, коли значення вологості у вашому каналі досягає або знижується вище 60:

1) Якщо канал, виберіть вимірювання точки роси канал.

2) Поле: вибрати 3 (Точка роси).

3) Для типу умови вибратиь більше або дорівнює.

4) Для значення умови ввести 60 (рис. 3.24).

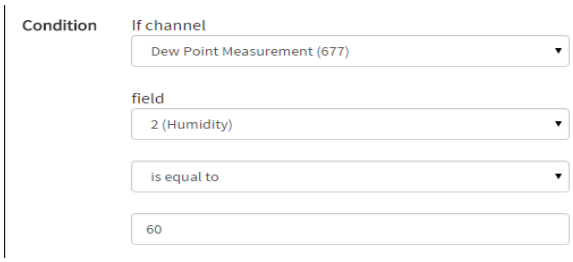

Рис. 3.24. Встановлення умов для запуску програми

5) Установити *Action* для *ThingTweet*.

6) Ввести цей рядок у твіт:

7) “Turn off that humidifier! It’s above 60*F!”*

8) Вибрати обліковий запис *Twitter* у використанні облікового запису *Twitter*.

9) У Параметрах вибрати “Виконувати дію кожного разу”, коли умова виконана.

10) Натиснути *Save React* (рис. 3.25).

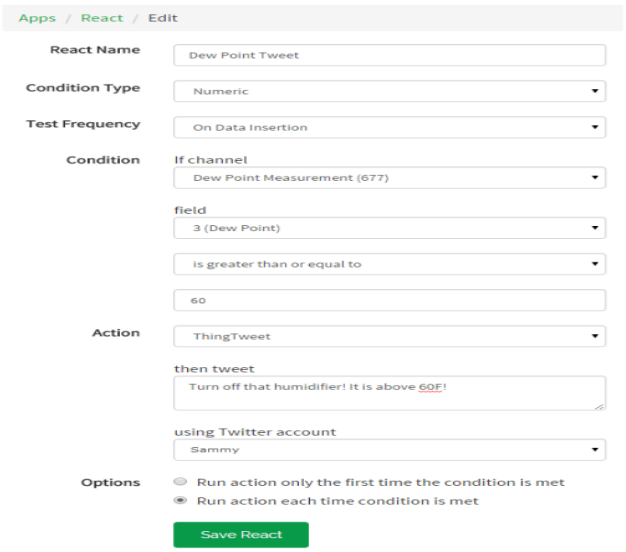

Рис. 3.25. Завершення налаштувань умов для запуску

Твіт надсилається кожного разу, коли рівень вологості підвищується вище 60°*F* (рис. 3.26).

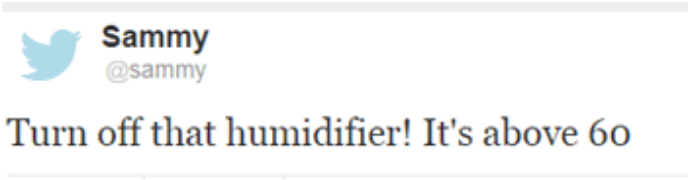

Рис. 3.26. Результат виконання програми

На цьому етапі показано, як збирати дані про навколишнє середовище з пристрою та публікувати їх із пристрою на канал *ThingSpeak*.

В даній задачі використовується аналоговий термістор і вбудований АЦП *ESP8266-12* для зчитування напруги та перетворення її в температуру (рис 3.27).

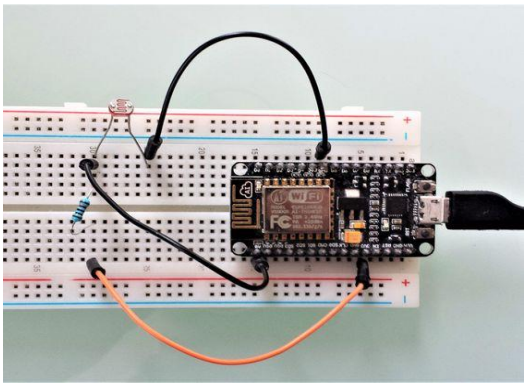

Рис. 3.27. Вбудований АЦП *ESP8266-12*

Термістор - це пристрій, опір якого обернено пропорційний температурі. Дані калібрування для термістора зберігаються у каналі користувача та зчитуються пристроєм.

Дані калібрування базуються на відповідності моделі Стейнхарта-Харта. Деякі виробники надають дані калібрування разом із пристроєм.

Користувач може самостійно підібрати параметри, вимірявши опір термістора при трьох різних температурах. Створити канал *ThingSpeak*, як показано в розділі Збір даних у новому каналі.

Увімкнути принаймні два поля, одне для читання та одне для запису. Дані записуються в поля 1, 2 і 3. Поля 6, 7 і 8 використовуються для зберігання коефіцієнтів калібрування температури.

На цьому зображенні показано вікно “Налаштування” каналу для налаштування одного каналу (рис. 3.28).

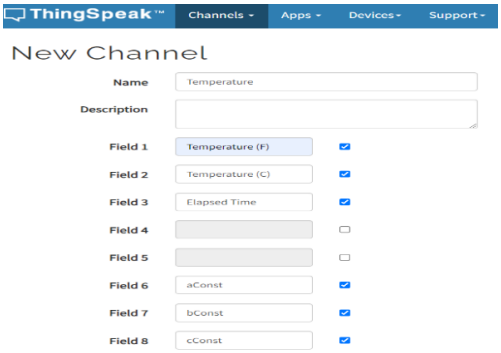

Рис. 3.28. Налаштування каналу

Записати ключ *API* для читання та запису ключа Заповнити поля калібрування за допомогою адресного рядка веб-переглядача. Наступні значення *a*, *b* та *c* є приблизними. Змінити наведений нижче текст за допомогою ключа *API* для запису та вставте його безпосередньо в адресний рядок свого браузера.

Установити a коефіцієнт:

https://api.thingspeak.com/update?api_key=AS3EGLRHPJBVY76P&field6=0.002039

Установити b коефіцієнт:

https://api.thingspeak.com/update?api_key=AS3EGLRHPJBVY76P&field7=0.0000672

Установити c коефіцієнт:

https://api.thingspeak.com/update?api_key=AS3EGLRHPJBVY76P&field8=0.0000008929

Відповідь - це кількість записів у каналі.

Якщо канал створено щойно, браузер повертає 1, 2 і 3 після створення першого, другого та третього полів калібрування відповідно (рис. 3.29).

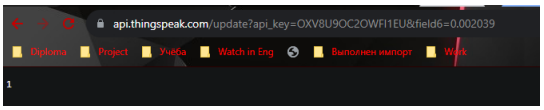

Рис. 3.29. Результат калібрування

Необхідне обладнання:

1. *ESP8266*. У цій задачі використовується *ESP8266-12E* на платі розробки *NodeMCU*. *ESP8266-01* не відкриває контакт АЦП. З *ESP8266-01* можна використовувати датчик температури з цифровим інтерфейсом.

2. Термістор 10 кОм, наприклад *Cantherm MF52A2103J3470*.

3. Резистор 10 кОм. Резистор з високим допуском, 1% або менше, дає кращі результати.

4. Макетна дошка.

5. Перемички (мінімум 3).

Підключити наступні контакти:

1. Один контакт термістора до контакту *A0* на *NodeMCU*.

2. Другий пін термістора до резистора 10 кОм.

3. Другий пін резистора на землю, який повинен бути загальним для *ESP8266*.

4. Плата *NodeMCU* має вбудований дільник напруги для перетворення вхідного діапазону 1 В АЦП *ESP8266* у діапазон 3,3 В (рис 3.30).

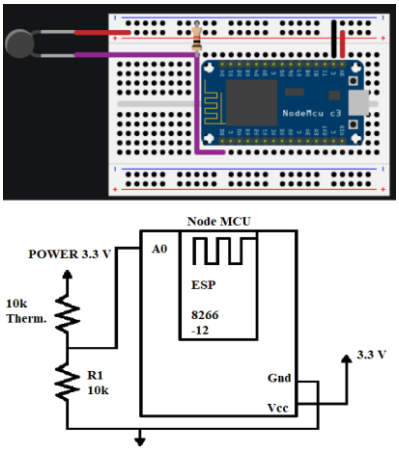

Рис. 3.30. Схема піключення контактів

Програмування *ESP8266*:

1. Завантажити останню версію *Arduino IDE*.

2. Додати *ThingSpeak Library for Arduino* and *ESP8266* (рис. 3.31):

Вибрати *Sketch* > *Include Library* > Керування бібліотеками.

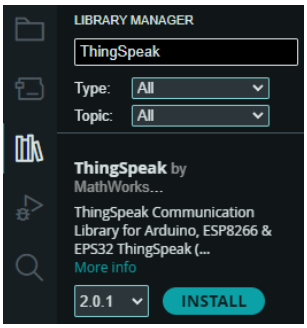

Рис. 3.31. Процес підключення бібліотек

Вибрати *ThingSpeak*, щоб додати його до ескізу.

3. Додати пакет плати *ESP8266* (рис. 3.32).

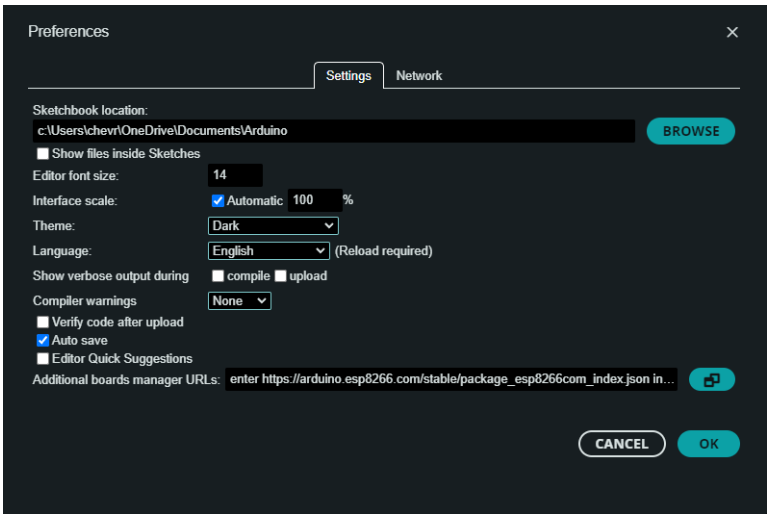

Рис. 3.32. Підключення пакету плати *ESP8266*

Вибрати “Інструменти” > “Дошки” > “Менеджер правління”. Ввести *ESP8266* у рядок пошуку та встановіть пакет. Але перед цим, під *File* > *Preferences*, ввести *https://arduino.esp8266.com/stable/package_esp8266com_index.json* у *Additional Board Manager URLs*.

4. Вибрати відповідний порт і плату в *Arduino IDE*. Апаратне забезпечення, використане для створення цього прикладу, використовувало параметр *Node MCU 1.0* (*ESP 8266–12E*).

5. Створити додаток. Відкрити нове вікно в *Arduino IDE* і зберегти файл. Додайти наданий у додатку А код (рис. 3.33). Обов’язково змінити інформацію про бездротову мережу, ідентифікатори каналів (можна використовувати один канал), ключ *API* читання та ключ *API* запису. Не потрібно змінювати коефіцієнти в коді, тому що програма зчитує їх з вашого каналу. Після підключення пристрій кожні дві хвилини вимірює напругу на термісторі.

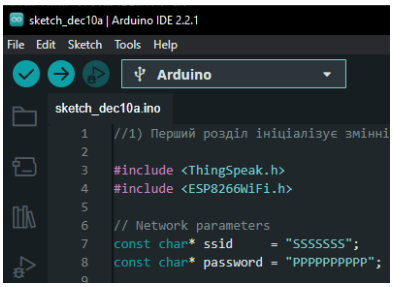

Рис. 3.33. Збереження файлу та внесення коду

Він розраховує температуру на основі калібрування, збереженого у вашому каналі, і надсилає температуру на ваш канал (рис. 3.34).

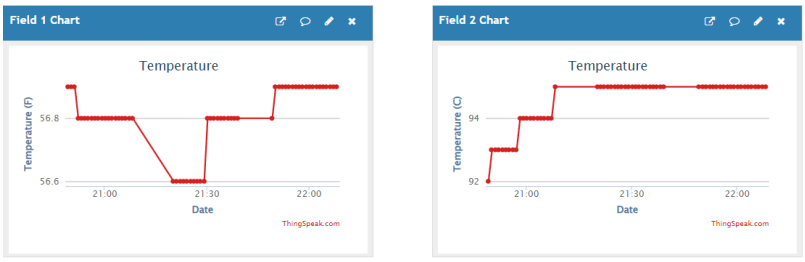

Рис. 3.34. Результат зчитування даних та розрахунку температури

Висновки за розділом

У цьому розділі було детально розібрано платформу Інтернету речей *ThingSpeak* та її можливості в зборі та аналізі даних. Розглянувши прилади *ESP8266* та *ESP32*, були розкриті її характеристики та використання у сценаріях *IoT*. Вивчено процес збору та аналізу даних в новому каналі *ThingSpeak*, від налаштування каналу до використання унікального ключа *API* для взаємодії із платформою.

Особливу увагу було приділено реальному використанню *ThingSpeak* через приклад зчитування даних про температуру та вологість з каналу *ThingSpeak* 12397. Використовуючи візуалізацію в *MATLAB Visualizations*, було не лише перевірена точність вимірювань, а й створено інтуїтивно зрозумілі графіки, які полегшують розуміння динаміки даних.

Покроково вивчено процес надсилання повідомлення на *Twitter* через *ThingTweet* та *React*, коли значення точки роси перевищують 60°*F*. Це показало, як *ThingSpeak* може використовуватися для автоматичної реакції на певні умови за допомогою реакційних дій.

На завершення, було досліджено, як збирати дані про навколишнє середовище з пристрою, використовуючи *ESP8266* та *ThingSpeak Communication Library*. Цей приклад чудово продемонстрував простоту публікації та отримання даних з каналу *ThingSpeak*.

ВИСНОВКИ

Інтернет речей (*IoT*) на сучасному етапі стає не лише ключовим напрямком технологій, але і каталізатором для трансформації різних сфер життя. Результати дослідження підтверджують, що *IoT* відкриває широкі перспективи для створення інтелектуальних середовищ, де об'єднуються фізичні об'єкти, пристрої та апаратні засоби в одну ефективну систему.

У першому розділі було виявлено, що *IoT* не просто забезпечує з'єднання пристроїв через мережу, але й створює інтелектуальні рішення, що дозволяють автоматизувати процеси, збирати та аналізувати великі обсяги даних, що в свою чергу приводить до більш ефективного використання ресурсів та раціонального управління.

Другий розділ розглянув ключові компоненти систем *IoT*, такі як сенсори, актуатори, протоколи та платформи. Виявлено, що сенсори відіграють рішучу роль у зборі різноманітних даних, що стає основою для подальшого аналізу та управління. Водночас, актуатори виконують крок за кроком дії на основі цих даних, забезпечуючи реакцію системи на зміни в навколишньому середовищі.

У третьому розділі проведено детальний аналіз платформи *ThingSpeak* та використання пристроїв *ESP8266/ESP32*. Цей розділ продемонстрував, як засоби моніторингу та управління можуть бути реалізовані в реальних проектах, а також як вони інтегруються з соціальними мережами, такими як *Twitter*, для реакції на певні події.

Отримані результати не тільки розкривають теоретичні аспекти *IoT*, але й надають практичні рекомендації для впровадження цих технологій у різні галузі. Зокрема, вони можуть служити основою для розробки нових систем моніторингу довкілля, енергоефективних рішень у будівництві, або навіть вдосконалення систем управління транспортом.

Незважаючи на досягнуті успіхи, виникають важливі питання, такі як забезпечення безпеки та конфіденційності в *IoT*, розробка єдиної стандартізованої архітектури для підключення різних пристроїв, і вдосконалення механізмів взаємодії

між платформами. Ці напрямки можуть стати основою для майбутніх досліджень в галузі Інтернету речей. Також необхідно підкреслити важливість доступності та безкоштовності платформ для *IoT*, особливо для студентів, які можуть зіткнутися з обмеженими ресурсами та бюджетними обмеженнями.

У сучасному світі *IoT* визначає нові можливості та розвиток технологій, а доступ до засобів моніторингу та управління цими системами стає ключовим елементом для освіти та практичного досвіду студентів. Забезпечення доступності безкоштовних платформ IoT стає необхідною передумовою для вивчення та розвитку студентських проектів, навіть в умовах обмежень та бідності.

Сприяючи доступності та безкоштовності платформ *IoT*, освітні установи і компанії можуть створити сприятливе середовище для навчання та досліджень, дозволяючи студентам реалізувати свої ідеї та вдосконалювати свої навички без фінансових обмежень.

Проектуючи або розробляючи систему Інтернету речей (*IoT*) з використанням безкоштовної платформи, такої як *ThingSpeak*, студент може зіткнутися з кількома викликами та обмеженнями.

Безкоштовні тарифи можуть обмежувати кількість датчиків або полів даних, які можна використовувати. Це може бути важливим, якщо ваш *IoT*-пристрій має багато параметрів для вимірювання.

Безкоштовні тарифи можуть обмежувати кількість даних, які можна зберігати. Це може призвести до необхідності видалення старих даних або обмежити тривалість моніторингу.

Деякі безкоштовні плани можуть мати обмеження на частоту відправки даних. Це може призвести до великих затримок у відображенні актуальних даних на платформі.

В залежності від величини користувацької бази *ThingSpeak*, обробка та відображення даних може вимагати часу, що може бути важливим при швидких змінах у ваших даних.

Безкоштовні тарифи можуть обмежувати доступ до деяких продвинутих інструментів візуалізації та аналізу даних. Це може збільшити важкість детального вивчення зібраних даних.

Безкоштовні плани можуть обмежувати можливості налаштування захисту доступу до даних. Це може створити проблеми з точки зору безпеки.

Безкоштовні плани можуть не включати високий рівень підтримки для вирішення можливих проблем. Спільнота користувачів може бути обмеженою, що ускладнює вирішення проблем.

Студенту важливо бути свідомим цих обмежень та використовувати їх як можливості для вдосконалення навичок управління обмеженими ресурсами та оптимізації роботи з *IoT*-системами.

Для невеликих проектів чи домашніх експериментів безкоштовні платформи можуть надавати всі необхідні можливості без додаткових витрат.

Проте важливо враховувати, що для великих підприємств та складних проектів, де необхідні великі ресурси та високий рівень безпеки, можуть вибирати платформи з платними тарифами.

У випадку росту проекту може виникнути потреба у більш розширених функціях, які надають платні плани.

Саме тому, розвивати безкоштовні платформи Інтернету речей (*IoT*), наприклад, *ThingSpeak*, має сенс з декількох причин.

Безкоштовні платформи надають доступ до базового функціоналу *IoT* без необхідності значних витрат. Це особливо важливо для студентів, початківців та тих, хто працює над невеликими проектами з обмеженими ресурсами.

Безкоштовні платформи чудово підходять для навчання та експериментів. Студентам легше вивчати принципи *IoT*, використовуючи доступні та прості платформи перед тим, як переходити до більш складних та коштовних.

Деякі безкоштовні платформи, такі як *ThingSpeak*, мають велику кількість користувачів та активну спільноту. Це створює можливості для обміну досвідом, підтримки та співпраці.

Безкоштовні платформи часто мають інтуїтивний і простий інтерфейс, що полегшує встановлення та конфігурацію *IoT*-проектів.

Для невеликих проектів чи домашніх експериментів безкоштовні платформи можуть надавати всі необхідні можливості без додаткових витрат.

Проте важливо враховувати, що для великих підприємств та складних проектів, де необхідні великі ресурси та високий рівень безпеки, можуть вибирати платформи з платними тарифами.

Узагальнюючи, дана кваліфікаційна робота висвітлює та аналізує ключові аспекти Інтернету речей та його систем моніторингу та управління.

Отримані результати демонструють перспективи *IoT* у покращенні якості життя, ефективності та екологічної стійкості, а також підкреслення важливості доступності та безкоштовності *IoT*-платформ для студентів відкриває можливості для ширшого впровадження та прийняття цих технологій, сприяючи розвитку наступного покоління інженерів та інноваторів у сфері Інтернету речей.

Отже, можна стверджувати, що Інтернет речей - це не просто технологічний тренд, але і ключовий елемент майбутнього сучасного суспільства.

СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. *Tripathy B. Internet of Things (IoT): TeChnologies, AppliCations, Challenges and Solutions* (англ.) / *B. Tripathy, J. Anuradha*. – *Florida: CRC Press*, 2017. – 334 с.

2. Наконечний А.Й., Верес З.Є. Інтернет речей і сучасні технології. Вісник Національного університету "Львівська політехніка". Автоматика, вимірювання та керування. 2016. № 852.С. 136-138.

3. *Sutaria, R., and Raghunath, G., "Making sense of interoperability: Protocols and Standardization initiatives in IoT," International Conference on Recent Trends in Communication and Computer Networks – ComNet 2013, 2013.*

4. *Android Developers Blog: Android Things client library for Google Cloud IoT Core* [Електронний ресурс]. – Режим доступу до ресурсу*:* *https://android-developers.googleblog.com/2018/06/android-things-client-library-for.html* (дата звернення 02.11.23р). – Назва з екрана.

5. *Azure IoT Development Cookbook* [Електронний ресурс]. – Режим доступу до ресурсу*:* *https://subscription.packtpub.com/book/virtualization-&-cloud/9781787283008/1/ch01lvl1sec06/understanding-the-azure-iot-suite* (дата звернення 04.11.23р). – Назва з екрана.

6. *How AWS IoT works - AWS IoT Core* [Електронний ресурс]. – Режим доступу до ресурсу*:* *https://docs.aws.amazon.com/iot/latest/developerguide/aws-iot-how-it-works.html* (дата звернення 05.11.23р). – Назва з екрана.

7. *Introduction to Google Cloud IoT* [Електронний ресурс]. – Режим доступу до ресурсу*:* *https://www.udemy.com/course/introduction-to-google-cloud-iot* (дата звернення 06.11.23р). – Назва з екрана.

8. Огляд технологій Індустрії 4.0 та основи Інтернету Речей [Електронний ресурс]. – Режим доступу до ресурсу*:* *https://pupenasan.github.io/intro.html* (дата звернення 06.11.23р). – Назва з екрана.

9. *ThingSpeak for IoT* [Електронний ресурс]. – Режим доступу до ресурсу*:* *https://thingspeak.com/pages/commercial_learn_more* (дата звернення 22.11.23р). – Назва з екрана.

10. *The IoT Platform with MATLAB Analytics* [Електронний ресурс]. – Режим доступу до ресурсу*:* *https://www.mathworks.com/help/thingspeak/* (дата звернення 22.11.23р). – Назва з екрана.

11. *Getting Started with Arduino products Analytics* [Електронний ресурс]. – Режим доступу до ресурсу*:* *https://www.arduino.cc/en/Guide* (дата звернення 27.11.23р). – Назва з екрана.

12. Сучасні мікроконтролери в електронній та інформаційно-вимірювальній техніці: навч. посіб. / О.В. Вовна та ін. Покровськ: ДВНЗ «ДонНТУ», 2020. 311с.

Код програмування *ESP8266*

//1) Перший розділ ініціалізує змінні. Обов’язково відредагуйте ключ *API* читання та запишіть ключ API, номери каналів і інформацію про з’єднання з бездротовою мережею.

```
#include <ThingSpeak.h>
#include <ESP8266WiFi.h>

// Network parameters
const char* ssid     = "NahIdWin";
const char* password = "STANDPROUDSUKUNAURSTRONG";

// ThingSpeak information
char thingSpeakAddress[] = "api.thingspeak.com";
unsigned long channelID = NNNNNN;
char* readAPIKey = "AS3EGLRHPJBVY76P";
char* writeAPIKey = "XEZ0Z6TQXOOE7S84";
const unsigned long postingInterval = 120L * 1000L;
unsigned int dataFieldOne = 1;                  // Field to write temperature data
unsigned int dataFieldTwo = 2;                  // Field to write temperature data
unsigned int dataFieldThree = 3;                // Field to write elapsed time data
unsigned int aField = 6;                        //Field to hold first constant of the thermistor calibration
unsigned int bField = 7;                        //Field to hold second constant of the thermistor calibration
unsigned int cField = 8;                        //Field to hold third constant of the thermistor calibration
```

```
// Global variables
// These constants are device specific.  You need to get them from the manufacturer or determine them yourself.
float aConst = 2.25E-02;
float bConst = -0.003422894649;
float cConst = 0.00001518485044;

unsigned long lastConnectionTime = 0;
long lastUpdateTime = 0;
WiFiClient client;
```

//2) У функції setup запустіть послідовний монітор і зчитайте константи калібрування.

```
void setup() {

Serial.begin(9600);
Serial.println("Start");
connectWiFi();

// Read the constants at startup.
aConst = readTSData( channelID, aField );
bConst = readTSData( channelID, bField );
cConst = readTSData( channelID, cField );
}
```

//3) Основний цикл постійно перевіряє, скільки часу минуло з моменту зчитування температури. Якщо встановлений період часу минув, пристрій зчитується, температура обчислюється, а вихід записується на ваш канал.

```
void loop() {
```

```
// Update only if the posting time is exceeded
  if (millis() - lastUpdateTime >=  postingInterval) {

    float fahrenheitTemperature, celsiusTemperature;

    lastUpdateTime = millis();

    float readValue = analogRead(A0);
    float logR = log( 10000 * ( 1024 / readValue - 1 ));          // Separate the calculation for simplicity and debugging

    celsiusTemperature = 1 / ( aConst + bConst * logR + cConst * pow(logR,3) ) - 273.15;  // Calculate the temperature in Celsius
    fahrenheitTemperature = celsiusTemperature * 9 / 5 + 32;
    Serial.println("ADC =  " + String( readValue )+ " Temp = "+String( fahrenheitTemperature ));
    write2TSData( channelID , dataFieldOne , fahrenheitTemperature , dataFieldTwo , celsiusTemperature , dataFieldThree , millis() );       // Write the temperature in F, C, and time since starting.
  }
}
```

//4) Підключіть свій пристрій до бездротової мережі за допомогою функції connectWiFi.

```
int connectWiFi(){

  while (WiFi.status() != WL_CONNECTED) {

    WiFi.begin( ssid, password );
    delay(2500);
```

```
        Serial.println("Connecting to WiFi");
      }

      Serial.println( "Connected" );
      ThingSpeak.begin( client );
    }
```

//5. Читайте дані з одного поля на каналі за допомогою readTSData. Ви можете записати одне значення в ThingSpeak за допомогою writeTSDatata й записати кілька значень одночасно за допомогою write2TSdata.

```
    float readTSData( long TSChannel,unsigned int TSField ){

      float data =  ThingSpeak.readFloatField( TSChannel, TSField, readAPIKey );
      Serial.println( " Data read from ThingSpeak: " + String( data, 9 ) );
      return data;

    }

    // Use this function if you want to write a single field.
    int writeTSData( long TSChannel, unsigned int TSField, float data ){
      int  writeSuccess = ThingSpeak.writeField( TSChannel, TSField, data, writeAPIKey ); // Write the data to the channel
      if ( writeSuccess ){

        Serial.println( String(data) + " written to Thingspeak." );
        }

        return writeSuccess;
    }
```

// Використовуйте цю функцію, якщо ви хочете записати кілька полів одночасно.

```
int write2TSData( long TSChannel, unsigned int TSField1, float field1Data, unsigned int TSField2, long field2Data, unsigned int TSField3, long field3Data ){

  ThingSpeak.setField( TSField1, field1Data );
  ThingSpeak.setField( TSField2, field2Data );
  ThingSpeak.setField( TSField3, field3Data );

  int writeSuccess = ThingSpeak.writeFields( TSChannel, writeAPIKey );
  return writeSuccess;
}
```