

Міністерство освіти і науки України
Національний університет “Львівська політехніка”
Інститут комп’ютерних наук та інформаційних технологій

Кафедра САПР



Лабораторна робота №5
з дисципліни: “Технології та стандарти інтернету речей”
на тему:
“Передача та обробка даних інтернету речей ”

Виконав:
Ст. групи ПП-44
Верещак Б. О.
Прийняв:
асис. Гавран В. Б.

Мета роботи

Ознайомитись з базовим принципами передачі даних IoT, вивчити методи збору, передачі та обробки даних з IoT-пристроїв, а також набуті практичних навичок використання відповідних технологій та протоколів у мережах IoT.

Теоретичні відомості

Основою роботи IoT-систем є процес збору, передачі та аналізу даних. Принцип роботи IoT об'єднує чотири різні етапи:

- Зчитування інформації за допомогою сенсорів;
- Передача даних від сенсорів до хмарних сховищ;
- Обробка даних отриманих за допомогою сенсорів;
- Передача інформації на інтерфейс користувача.

У найзагальнішому вигляді з інфокомунікаційної точки зору Інтернет речей можна записати у вигляді наступної символічної формули:

$$\text{IoT} = \text{Сенсори (датчики)} + \text{Дані} + \text{Мережі} + \text{Послуги}.$$

Сенсори, розташовані на пристроях, збирають інформацію про стан об'єкта або середовища. Далі ця інформація передається до серверів чи хмарних сервісів, де відбувається обробка та аналіз даних. В результаті аналізу формуються відповідні команди, які направляються до виконавчих пристроїв або систем для забезпечення потрібної реакції.

Роль сенсорів, виконавчих пристроїв та вбудованих систем у IoT-пристроях. Сенсори відіграють ключову роль у IoT-пристроях, оскільки вони відповідають за збір даних. Виконавчі пристрої (актуатори) забезпечують фізичну реакцію на команди, отримані від серверів чи хмарних сервісів. Вбудовані системи відповідають за обробку, зберігання та передачу інформації між різними компонентами IoT-пристрою.

Різноманітні способи передачі даних, включаючи дротові та бездротові мережі. Передача даних у IoT-системах може здійснюватися за допомогою дротових та бездротових мереж. До дротових мереж належать Ethernet, оптоволоконні кабелі та інші види кабельних з'єднань. Бездротові мережі включають Wi-Fi, Bluetooth, Zigbee, LoRaWAN, мобільні мережі (3G, 4G, 5G) та інші технології передачі даних бездротовим способом. Вибір способу передачі даних залежить від специфіки IoT-пристрою, відстані між пристроями, що взаємодіють, та вимог до швидкості та надійності передачі інформації. Крім того, передача даних може відбуватися через мережі, що використовують різні протоколи та стандарти, такі як MQTT, CoAP, HTTP та інші. Ці протоколи розроблені для специфіки IoT-систем, враховуючи особливості їх архітектури, ресурсів та вимог до енергоспоживання.

Лабораторне завдання

- Ознайомитися з теоретичними відомостями.
- Описати архітектуру передачі та обробки даних інтернету речей згідно з варіантом.
 - Розумні парковки: Система моніторингу та резервування місць;
 - Інтелектуальна система керування шторами;
 - Інтелектуальна система керування електроспоживанням будинку;
 - Інтелектуальна система керування та оптимізації домашніх генераторів;
 - Застосунок для керування LED-стрічкою та освітленням;
 - Розумний інкубатор;
 - Система моніторингу периметру і виявлення загрози;
 - Інтелектуальна система догляду за садом;
 - Розумний термостат для опалення;
 - Розумні двері для домашніх тварин.
- Оформити звіт до лабораторної роботи.

Результати виконання завдання:

У межах цієї лабораторної роботи було розроблено архітектуру передачі та обробки даних для системи Інтернету речей, що здійснює моніторинг погодних умов у реальному часі. Основою системи є мікроконтролер ESP32, який зчитує дані з підключених сенсорів (BME280, ENS160 та HALL-сенсор) і передає їх на сервер для подальшої обробки та збереження.

Теперішня система

Передача даних реалізована через HTTP-протокол поверх Wi-Fi. Метеостанція формує JSON-пакет із поточними показниками температури, вологості, тиску, якості повітря (CO₂, TVOC, AQI) та кількості опадів. Потім цей пакет надсилається HTTP POST-запитом на сервер Flask, який виступає центральним вузлом обробки даних.

Серверна частина системи побудована на Python Flask, що забезпечує прийом, валідацію та обробку вхідних запитів. Після отримання даних сервер зберігає їх у базі даних MongoDB у форматі JSON-документів. Крім зберігання, сервер виконує аналітичні операції: визначає середні, мінімальні й максимальні значення параметрів за певний період, виявляє перевищення порогових показників (наприклад, температури чи рівня CO₂) та надсилає сповіщення користувачам через WebSocket.

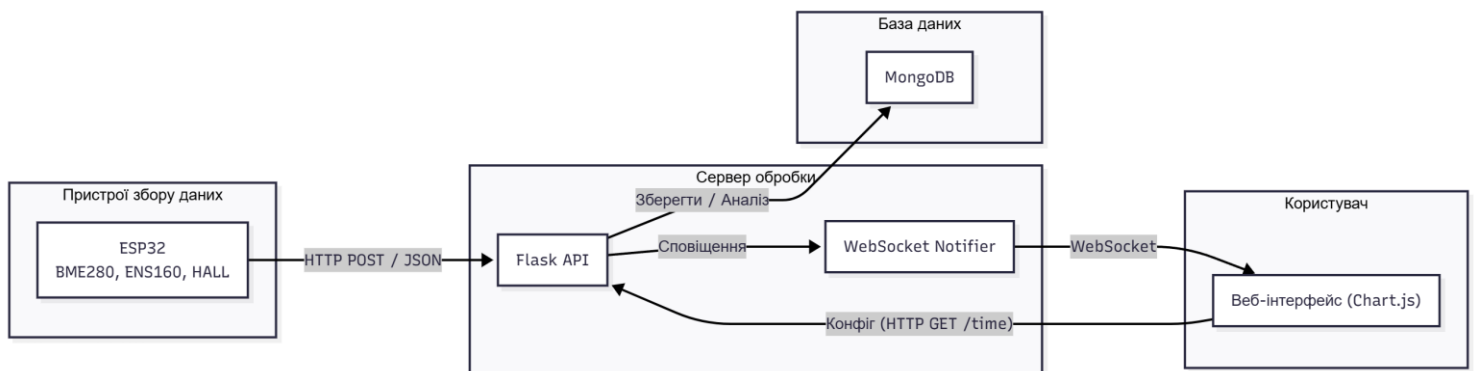
Для отримання конфігураційних параметрів (наприклад, інтервалу оновлення даних або Wi-Fi налаштувань) метеостанція здійснює HTTP GET-запит до сервера. Також реалізована синхронізація часу через API /time, що забезпечує коректне маркування вимірювань часовими мітками у форматі ISO 8601.

Користувач взаємодіє з системою через веб-інтерфейс, який отримує дані з Flask API та візуалізує їх за допомогою Chart.js. Це дозволяє переглядати графіки зміни температури, вологості, тиску та якості повітря за вибраний період, а також налаштовувати порогові значення для отримання сповіщень.

Таким чином, теперішня архітектура має класичну трирівневу структуру:

- пристрої збору даних (ESP32 + сенсори),
- сервер обробки (Flask + MongoDB),
- користувацький інтерфейс (веб-додаток).

Це забезпечує повний цикл обробки — від вимірювання параметрів до аналітики та відображення результатів у зручній формі.



Масштабування системи

Поточна реалізація чудово підходить для однієї метеостанції або кількох пристроїв у межах однієї локальної мережі. Проте при розширенні системи на десятки або сотні сенсорних вузлів доцільно перейти на архітектуру з використанням MQTT-брокера.

У масштабованому варіанті кожна метеостанція передаватиме свої дані не безпосередньо серверу Flask, а до MQTT-брокера (наприклад, Mosquitto або EMQX), який виступатиме центральним вузлом комунікації. Сервер Flask буде підписаний на відповідні топіки (station/+data) і збиратиме всі повідомлення, виконуючи ті самі функції аналізу та збереження.

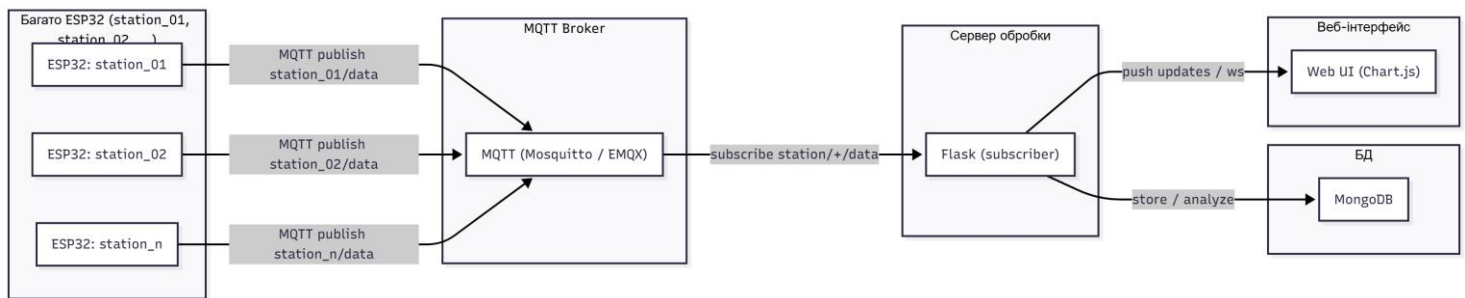
Кожна метеостанція при цьому повинна мати унікальний ідентифікатор `device_id`, який додається до JSON-пакета.

База даних MongoDB зберігатиме дані з усіх станцій, а веб-інтерфейс отримає можливість фільтрувати й порівнювати показники за різними локаціями. MQTT дозволить зменшити навантаження на сервер, забезпечить надійність доставки повідомлень завдяки механізму QoS і дозволить масштабувати систему без змін у логіці клієнтських пристроїв.

Отже, при масштабуванні архітектура набуває вигляду:

ESP32 → MQTT Broker → Flask Server → MongoDB → Web Interface

Така структура забезпечує централізований збір даних, мінімальні затримки передачі, підтримку великої кількості пристроїв і можливість легкого розширення системи.



Висновки

На даній лабораторній роботі я ознайомився з базовими принципами передачі даних IoT, вивчив методи збору, передачі та обробки даних з IoT-пристроїв, а також набув практичних навичок використання відповідних технологій та протоколів у мережах IoT.

У результаті було спроектовано архітектуру передачі та обробки даних для системи Інтернету речей. Розроблена структура забезпечує повний цикл обробки інформації — від збору показників із сенсорів до їх аналізу, зберігання та візуалізації у реальному часі.

Поточна архітектура на основі HTTP-протоколу є оптимальною для невеликої кількості пристроїв, забезпечує простоту реалізації та стабільну роботу. Водночас, для подальшого розширення системи передбачено можливість переходу на MQTT-протокол, що забезпечить гнучке масштабування, підвищену надійність передачі та ефективне керування великою кількістю IoT-вузлів.