

ДОДАТОК А

Графічний матеріал кваліфікаційної роботи

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

ФАКУЛЬТЕТ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА УПРАВЛІННЯ
КАФЕДРА КІТС

«Розробка мікросервісної системи з підтримкою паралельної обробки запитів для аналізу телеметричних даних в реальному часі»

Виконав ст. гр. КІУКІ-21-10
Маслов Є. О.

Керівник ст. викл. кафедри КІТС
М.В. Кушнар'юв

Харків, 2025

Мета та завдання

Метою даної бакалаврської роботи є розробка мікросервісної програмної системи для збору, обробки, аналізу та візуалізації телеметричних даних, що надходять з віддалених пристроїв, із підтримкою паралельної обробки запитів.

Для досягнення мети необхідно вирішити такі основні задачі:

- Проаналізувати предметну область та існуючі підходи до обробки телеметричних даних у режимі реального часу, з урахуванням вимог до масштабованості та швидкодії.
- Спроекувати архітектуру системи на основі мікросервісного підходу з виділенням основних сервісів: реєстрації пристроїв, збору даних, обробки потоків, виявлення аномалій, сповіщення та візуалізації.
- Реалізувати підтримку паралельної обробки за допомогою відповідних технологій потокової обробки даних (Apache Kafka, Apache Flink).
- Забезпечити взаємодію між мікросервісами через брокер повідомлень або REST/gRPC API.
- Оцінити продуктивність системи, її масштабованість та здатність до виявлення критичних ситуацій з мінімальною затримкою.

Архітектура мікросервісної системи для аналізу телеметричних даних у реальному часі



3

Порівняння характеристик ESP32 та Arduino (Uno/Nano)

Критерій	Arduino (Uno/Nano)	ESP32 (DevKit)
Процесор	8-бітний (ATmega328P)	32-бітний (Xtensa LX6, 2 ядра)
Тактова частота	16 МГц	80–240 МГц
Оперативна пам'ять	2 КБ (<u>Uno</u>), 2 КБ (<u>Nano</u>)	520 КБ
Flash-пам'ять	32 КБ (<u>Uno</u>), 32 КБ (<u>Nano</u>)	4–16 МБ
Живлення	5 В	3.3 В (може працювати від 5 В)
Вбудований зв'язок	Немає	Wi-Fi, Bluetooth (BLE)
Зв'язок через модулі	<u>Wi-Fi/GSM/LoRa</u> через модулі	Не потрібні додаткові модулі
Енергоспоживання	Низьке	Дуже низьке в режимі сну
Підтримка ШІ	Немає	Є (<u>TensorFlow Lite</u>)
ОТА-оновлення	Немає	Підтримується
Складність програмування	Дуже проста	Вища (багатозадачність, ОТА)
Ціна	~\$5–10	~\$10–20
Приклади застосування	Простий полив, метеостанція	Розумна теплиця, хмарна аналітика

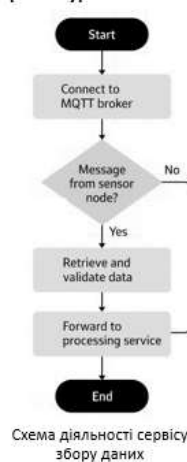
4

Методика проведення експериментів

Етап 1. Вибір і калібрування датчиків

- Обрано датчики для моніторингу: вологість ґрунту, температура, вологість повітря, освітленість, pH.
- Використано сенсори DHT22, ВН1750, pH-метри, інтеграція з ESP32 (Wi-Fi/Bluetooth) та Arduino Nano (LoRa/Serial).
- Проведено калібрування й тестування у лабораторних і реальних умовах.
- Збір даних — через MQTT/HTTP у мікросервісну систему, подальша візуалізація в Grafana.
- Налаштовано автоматичний збір, перевірку, передачу та зберігання даних для подальшого аналізу агропараметрів.

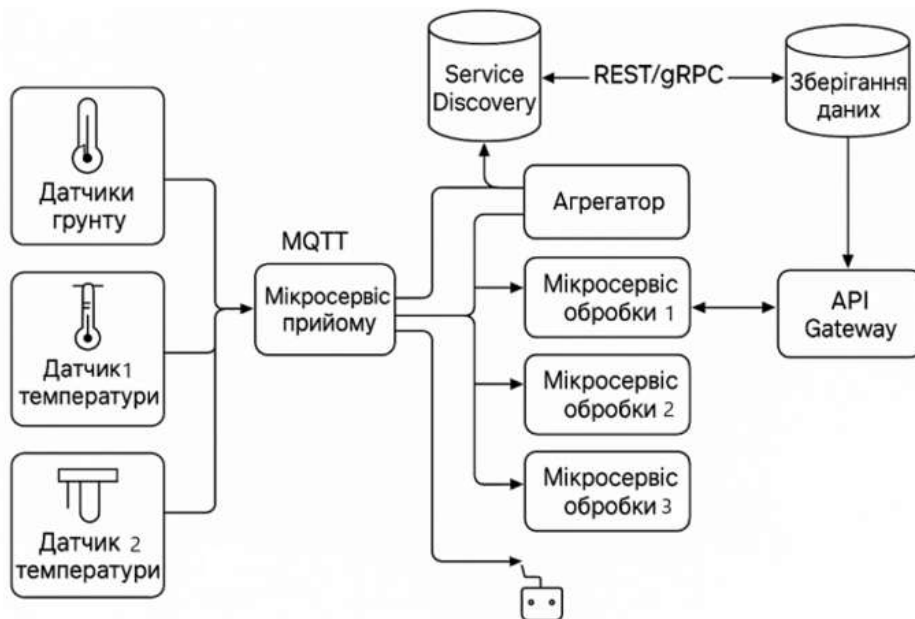
Етап 2. Формування модульної архітектури



Етап 3. Тестування системи

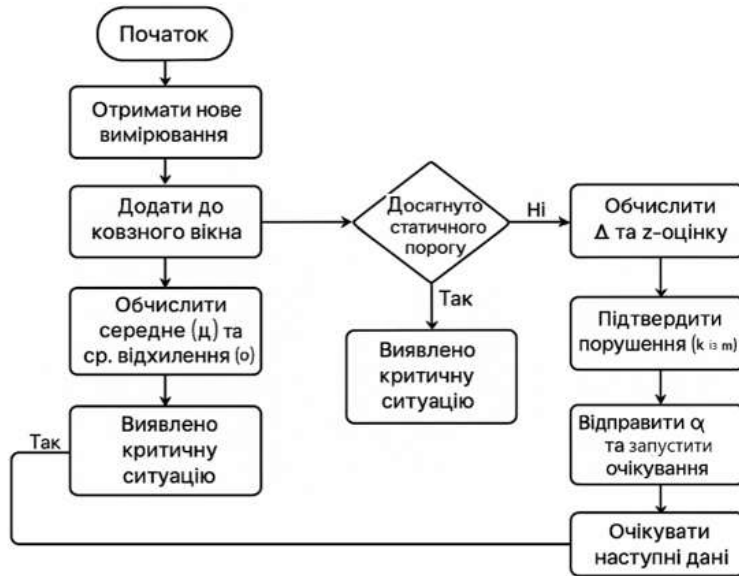
5

Шлях даних від датчиків через мікросервіс прийому MQTT до агрегатора й обробних сервісів



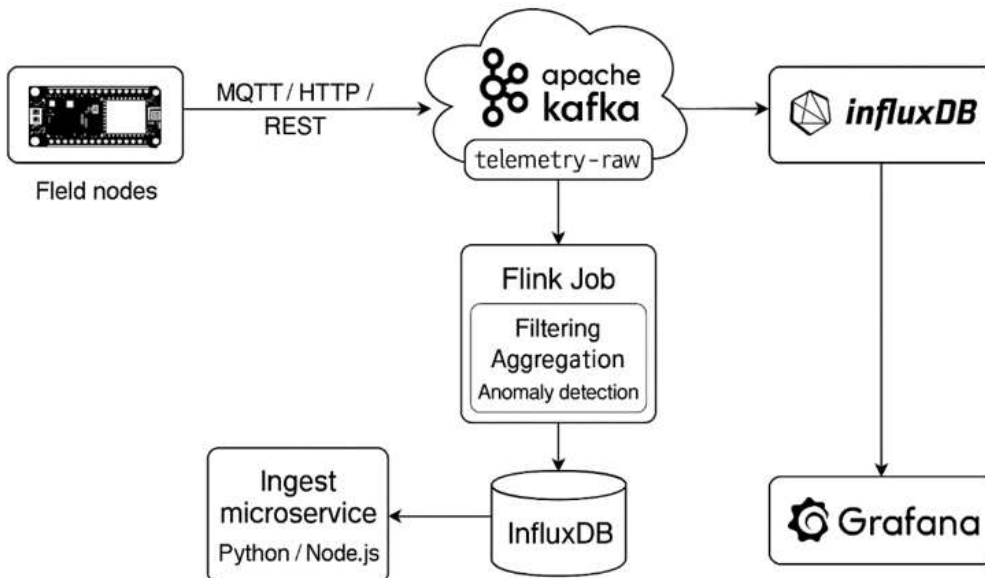
6

Схема алгоритму виявлення критичних ситуацій



7

Паралельна обробка потоків даних



8

Результати паралельної обробки

```
{ "sensor_id": "S01", "hum": 32, "ts": "2024-06-16T14:01:05" }
{ "sensor_id": "S02", "hum": 28, "ts": "2024-06-16T14:01:07" }
{ "sensor_id": "S03", "hum": 41, "ts": "2024-06-16T14:01:09" }
{ "sensor_id": "S01", "hum": 25, "ts": "2024-06-16T14:01:15" }
```

Вхідний потік із Kafka

```
Alert! Sensor S02: low humidity 28 at 2024-06-16T14:01:07
Alert! Sensor S01: low humidity 25 at 2024-06-16T14:01:15
```

Вивід у консолі Flink



Дашборд Grafana із двома часовими кривими

Порівняння потокових даних

Метрика	ESP32 (Wi-Fi)	Arduino (Nano + LoRa)	Коментар
Середнє відхилення вологості ґрунту(МАЕ, %, 24 год)	0.8 %	1.3 %	ESP32 має меншу похибку, імовірно через вищу роздільну здатність АЦП (12-бит проти 10-бит).
Середнє відхилення температури(МАЕ, °C, 24 год)	0.26 °C	0.24 °C	Різниця статистично незначуща; обидва вузли використовують однаковий цифровий сенсор (VME280).
Дисперсія шуму після фільтрації(σ^2 , % ²)	0.11	0.19	Нижчий шум у ESP32 пояснюється меншою кількістю перешкод на I ² C-шині (коротші з'єднання, відсутність LoRa-шилду).
Середня затримка доставки(мс, P50 / P95)	85 / 135	450 / 830	Wi-Fi-канал ESP32 швидший; LoRa-передача буферизується та пакетизується.
Втрати повідомлень(% ² , 24 год)	0.4 %	0.1 %	LoRa стабільніший на дальності 80 м із перешкодами (металеві стелажі теплиці).
Добове споживання енергії(мА-год)	620	210	Nano + LoRa працює з $duty-cycle \leq 1\%$; ESP32 активний 30 с кожні 5 хв, але Wi-Fi піки до 250 мА.
Середній розмір пакетів(байт)	160	42	Protocol Buffers (ESP32) проти LoRa-payload (ASCII-CSV).
Коефіцієнт успішного запису в InfluxDB	99.6 %	99.9 %	У Arduino-гілці більше повторних спроб → вища успішність, але за рахунок латентності.

9

Висновки

У результаті кваліфікаційної роботи:

- Розроблено прототип мікросервісної системи агротелеметрії з паралельною обробкою потоків даних у реальному часі (Kafka, Flink, InfluxDB, Grafana).
- Забезпечено масштабованість і гнучкість — додавання сенсорів та модулів не потребує зміни всієї системи.
- ESP32 довів свою ефективність для хмарної інтеграції й миттєвих алертів, Arduino Nano + LoRa — для автономних вузлів із низьким енергоспоживанням.
- Система автоматично виявляє критичні ситуації та генерує сповіщення, підтримує гібридну архітектуру й оптимізує енергоспоживання.
- Підтверджено стабільність, точність і можливість масштабування рішення для агро-ІoТ та промислового моніторингу.

10