

<https://doi.org/10.31891/2307-5732-2025-357-69>

УДК 004.7+004.5+621.382+621.391+621.396+621.398

ГЕРМАН ЮРІЙ

Чернівецький Національний університет імені Юрія Федьковича

<https://orcid.org/0009-0003-2473-7365>

e-mail: herman.yurii@chnu.edu.ua

САМІЛА АНДРІЙ

Чернівецький Національний університет імені Юрія Федьковича

<https://orcid.org/0000-0001-8279-9116>

e-mail: a.samila@chnu.edu.ua

МОДУЛЬНА EDGE-СИСТЕМА ЗБОРУ ТА АНАЛІЗУ ДАНИХ НА БАЗІ RASPBERRY PI ТА SOC FPGA

У статті представлена модульна телеметрична платформа для збору, обробки та відображення даних у реальному часі, що базується на принципах кордонних (edge) обчислень. Архітектура інтегрує датчики як джерела даних, контролери (Raspberry PI та SoC) для обробки даних та веб-інтерфейс, що використовується для візуалізації цих даних, в єдину архітектуру з меншою залежністю від зовнішніх мереж. Дані передаються за протоколом MQTT з локальним брокером, що забезпечує мінімальну затримку і робить весь транспортний рівень локальним. Передача даних між компонентами відбувається в локальній мережі Wi-Fi або mesh. SQLite, що працює в режимі попереднього запису, підвищує надійність зберігання даних і полегшує відновлення після збоїв. Веб-інтерфейс на основі Leaflet полегшує візуалізацію і підтримує потокову оновлення даних через Server-Sent Events (SSE). Система спілкується з модулями SoC FPGA та периферійними пристроями за допомогою UART та SPI, що забезпечує масштабованість для різних додатків. Описана архітектура забезпечує простоту, автономність і базується на промислових стандартах IoT, що дозволяє реалізувати платформу в областях, які не мають доступу до мережі. Пропонована система може бути модернізована в різних напрямках, забезпечуючи при цьому критично важливі функції як основа.

Ключові слова: edge computing, IoT, IoE, SoC FPGA, телеметрія, модульні телекомунікаційні системи.

HERMAN YURIY, SAMILA ANDRIY

Yuri Fedkovych Chernivtsi National University

MODULAR EDGE SYSTEM FOR DATA COLLECTION AND ANALYSIS BASED ON RASPBERRY PI AND SOC FPGA

Article presents a modular telemetry platform for real-time collecting, processing, and presenting data that is relying on edge computing principles. The architecture integrates sensors as data producers, controllers (Raspberry PI and SoC) for processing data, and a web interface used to present and operate that data, into a unified architecture with less reliance on external networks. This method lets the platform function without reliance on global network therefore it can do its job in environments that have no reliable network coverage. Local data transfer is built around usage of MQTT broker, making sure that messages are sent and received, while keeping whole process asynchronous. All the parts function together in a localized Wi-Fi network, which makes it possible for data to be shared without needing additional transport layers. This setup does away with the requirement for cloud servers while still keeping data safe and making sure the system can keep running. SQLite, when set to write-ahead logging mode, is a strong way to save data locally and lets you quickly get back to work after an unforeseen error. The visualization layer, which is built on the Leaflet library, allows real-time map rendering and dynamic data updates through Server-Sent Events (SSE). This gives operators a clear, constantly updated view of the system's state. Connecting SoC FPGA modules and other devices through UART and SPI interfaces makes it possible to sync data at the hardware level and opens up new ways to scale the system. The architecture described provides a simple, modular, standalone solution. It meets the specifications for modern industrial IoT, so it can be used for research, field deployment, or laboratory automation. It is possible to easily add more sensors, data processing units, or communication interfaces to the system. Also, the whole approach allows possible extension to actually work with global network and distant compute power if needed in centralized structure arises. This makes it a flexible base for future development while keeping its essential functions and stability.

Keywords: edge computing, IoT, IoE, SoC FPGA, telemetry, modular telecommunications systems.

Стаття надійшла до редакції / Received 29.08.2025

Прийнята до друку / Accepted 15.09.2025

Постановка проблеми у загальному вигляді

та її зв'язок із важливими науковими чи практичними завданнями

З поширенням технологій у більшості активних галузей промисловості багато завдань вирішуються в тому ж місці, де вони виникають, що робить міграцію комп'ютерних процесів на периферію мережі поширеним підходом у польових пристроях. Постійне збільшення кількості датчиків, різке зростання обсягу поточкових даних та необхідність зменшення затримки ставлять нові виклики перед традиційними централізованими архітектурами. Хоча звичайні хмарні або серверні рішення надають користувачам велику обчислювальну потужність, місцеві оператори змушені мати справу зі зростаючою пропускнуою здатністю, відсутністю мереж або загальною заміною швидкості обробки на надійність і потужність делегованих обчислювальних одиниць десь інде.

Описана вище ситуація ускладнює експлуатацію цих систем і порушує такі критичні питання, як відсутність гарантованого з'єднання з іншими серверами, неможливість швидкої адаптації до місцевих умов та загальне збільшення часу обробки. За таких обставин впровадження передових технологій, що забезпечують первинну обробку, фільтрування та аналіз даних безпосередньо на рівні

локальних пристроїв, стає перспективним напрямком. Це полегшує навантаження на транспортні маршрути, скорочує час відгуку системи та підвищує її стійкість у реальних умовах.

Практична користь дослідження полягає у розробці архітектури, що поєднує простоту конструкції, гнучкість масштабування та підвищену відмовостійкість. З точки зору сучасної інженерії, нагальною потребою є розробка систем, здатних працювати в автономному режимі, підтримуючи безперебійний обмін даними між датчиками, вузлами обробки та інтерфейсами користувача. Пропонована архітектура має на меті вирішити ці проблеми шляхом створення модульної телеметричної платформи з використанням Raspberry Pi та технології SoC FPGA. Основною метою є продемонструвати можливість створення надійної системи для збору, обробки та візуалізації даних у реальному часі без залежності від централізованих або хмарних систем використовуючи підходи поширені в edge систем.

Аналіз досліджень та публікацій

У межах розвитку розподілених систем обробки даних особливу увагу в сучасних дослідженнях приділено концепції edge computing, яка передбачає перенесення частини обчислювальних процесів безпосередньо до рівня сенсорів або локальних вузлів. У роботах [1] та [2] наголошено, що саме така архітектура забезпечує зменшення затримок під час передавання інформації та дозволяє знизити навантаження на центральні сервери. Для побудови edge-вузлів активно застосовуються платформи Raspberry Pi [3] завдяки низькому енергоспоживанню, гнучкій конфігурації периферії та можливості інтеграції з FPGA-модулями Використання SoC FPGA дає змогу апаратно реалізувати частину функцій збору або синхронізації даних, зменшуючи навантаження на центральний процесор. Проблеми телеметрії та керування розподіленими пристроями у режимі реального часу розглянуто у працях [4], присвячених використанню протоколу MQTT [5] який довів свою ефективність у середовищах із високою затримкою або нестабільним каналом зв'язку. Його модель публікаційно-підписної взаємодії дає змогу організувати обмін даними між великою кількістю вузлів без складних механізмів маршрутизації. Особливу роль у розробленні сучасних систем відіграють легкі бази даних, що забезпечують швидкий доступ до даних і високу надійність транзакцій. SQLite у режимі write-ahead logging (WAL) [6] є прикладом рішення, яке поєднує компактність, відмовостійкість і підтримку одночасного доступу до даних без втрати продуктивності. Таким чином, сучасні дослідження демонструють наявність окремих технічних рішень, однак недостатньо уваги приділяється побудові цілісних архітектур з малою кількістю компонентів, та з пониженою залежністю від зовнішньої мережі. Загальний підхід орієнтується на створення компонентних систем, що взаємодіють між собою та, за потреба, із зовнішніми елементами, чим самим підвищують складність всієї інфраструктури.

Формулювання цілей статті

Метою роботи є створення архітектури модульної edge-системи, яка здатна збирати, обробляти та демонструвати телеметричні дані в режимі реального часу. **Основне завдання** полягає у розробці платформи, яка може забезпечити надійну роботу в локальній мережі без залежності від зовнішньої підтримки, підтримувати низькі затримки та бути стійкою до втрат з'єднання. **Для досягнення поставленої мети** передбачено послідовне виконання таких етапів:

- розробка єдиної архітектури для системи,
- визначити, як її елементи транспортуватимуть дані,
- створити місцевий центр збору та обробки даних,
- інтерактивної панелі для показу даних.

Особливу увагу приділено забезпеченню автономності роботи, надійності передавання даних та можливості подальшого розширення функціональних можливостей системи.

Результатом дослідження є формування практичної архітектури телеметричної платформи, орієнтованої на автономність, модульність і простоту інтеграції в уже існуючі системи.

Виклад основного матеріалу

Незважаючи на те, що система складається з багатьох рівнів, обмін даними є її загальним та найкритичнішим елементом. Оскільки кожен компонент працює в мережі, зберігання та передача даних у правильному форматі й у потрібний момент є критично важливими. Це ключове питання для функціонування системи, оскільки всі її елементи побудовані з метою спрощення прийняття швидких рішень та збереження інформації для подальшого аналізу

Джерелами даних є додаткові пристрої, такі як GPS-приймачі та зовнішній міношукач, котрі надсилають дані у двійковому форматі. Система одразу обробляє кожен частину потоку, в даному випадку на апаратному забезпеченні Raspberry Pi. На цьому рівні система читає та змінює дані в єдиний формат, з яким вона може в подальшому працювати.

Вся платформа базується на обробці потоків даних, що надходять із зовнішніх джерел. Це означає, що ви не можете переміщати ці потоки по системі на свій розсуд. Саме тому було обрано протокол MQTT: він може гарантувати, що дані можна надсилати та отримувати в різний час [7]. Для цього існує модель «публікація-підписка», де кожне повідомлення має тему, і клієнти отримують тільки ті повідомлення, на які вони підписані. Місцевий брокер зберігає повідомлення і може отримати їх з журналів, якщо вони потрібні. Система також робить копію отриманих даних і зберігає її в

окремому файлі перед відправкою. Існує кілька точок відновлення (WAL, файл резервної копії, журнал брокера), що зменшує ймовірність втрати важливих даних.

Антену на щоглі висотою кілька метрів покриває робочу зону. Це дозволяє локальним користувачам використовувати планшети чи ноутбуки без додаткових пристроїв для зв'язку. Система недоступна поза зоною та без доступу в мережу, у полі Wi-Fi залишається основним засобом передачі даних [8]. Даний підхід гарантує, що система буде працювати навіть у разі великого потоку даних або виникнення проблем. Зберігання відбувається локально, тому доступ до даних та їх відновлення можливе з декількох точок відмовостійкості.

На рис. 1 зображено загальну процесну діаграму функціонування системи та взаємодії користувача із нею. Візуалізація та управління доступні оператору в локальній мережі. Панель керування відображає карти у веб-інтерфейсі, доступному через браузер. Користувачі можуть бачити активність на самій карті в моменті надходження даних, змінювати пороги для отримання даних та, за потреби, експортувати результат для подальшої роботи у GIS-середовищах.

Архітектура підтримує протоколи NMEA, MQTT і HTTP/SSE. Це дає змогу системі взаємодіяти з іншими платформами та не обмежуватися конкретним обладнанням. Важливо, щоб кожен рівень мав способи роботи без доступу до зовнішньої мережі Internet. Така характеристика дозволяє масштабувати систему задля підтримки різних рівнів зв'язності, за потреби [9].

Проте телекомунікаційний рівень визначає стабільність телеметричної платформи, яка базується на обміні даними. Усі інші частини, будь то база, алгоритми чи карта, не матимуть значення, якщо не буде гарантованої доставки повідомлень. Ця залежність є очевидною у випадках, де оператор повинен отримувати дані від сенсора відразу після їх появи, а не через кілька хвилин. Кожне відкладення або втрата повідомлення означає ймовірність помилки під час виконання завдань.

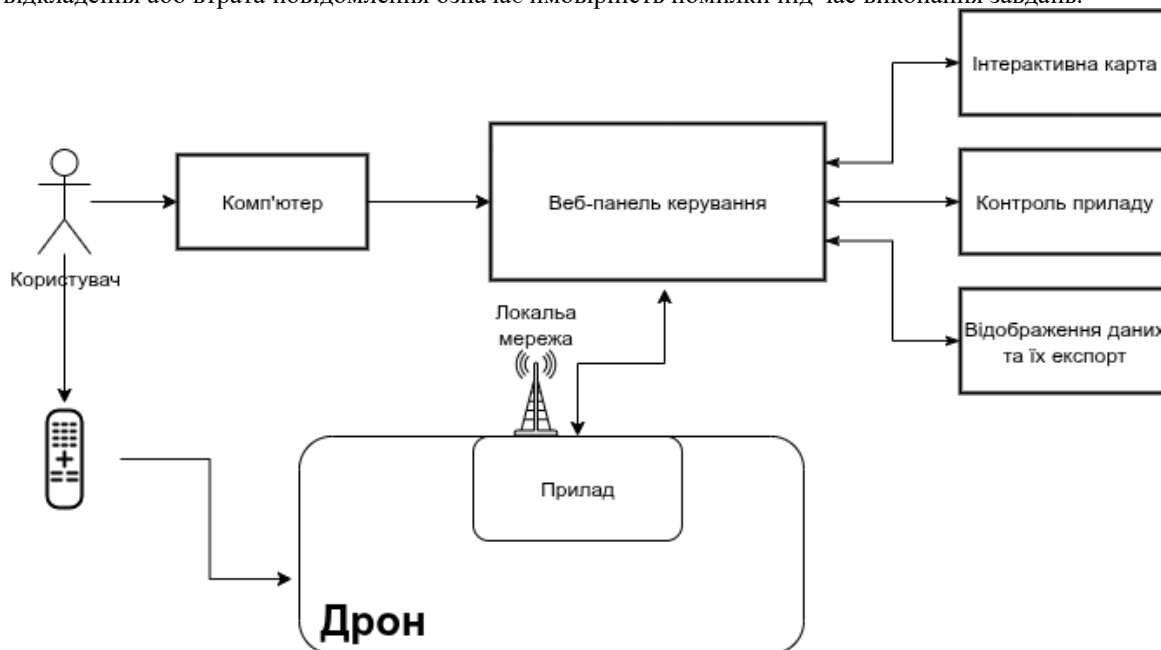


Рис. 1. Взаємодія користувача з системою

Уся система функціонує в закритому середовищі. Для забезпечення зв'язку використовується точка доступу Wi-Fi, яка розташована на щоглі висотою кількох метрів. Це просте рішення дозволяє покрити робочу зону, дозволяючи групам саперів використовувати ноутбуки та планшети без додаткових пристроїв. У полі Wi-Fi часто є лише єдиним каналом, який можна використовувати без доступу до глобальних мереж [10]. Мобільна мережа може бути відсутньою або нестабільною, а супутниковий зв'язок занадто дорогий і повільний. Таким чином, все починається з роботи в повністю офлайн-режимі.

Незважаючи на те, що покриття Wi-Fi обмежене десятками метрів, це достатньо для ситуацій, коли оператори працюють у межах одного місця. У разі потреби радіус дії можна збільшити за допомогою підсилювачів або кількох точок доступу, але основна логіка залишається тією ж: усі компоненти — сенсори, клієнтські пристрої та Raspberry Pi як edge-сервер — знаходяться в одній локальній мережі [11].

Протокол MQTT дозволяє елементам системи надсилати дані з використанням брокера повідомлень. Це простий протокол, створений для пристроїв з обмеженими ресурсами та мереж із великою затримкою. Оскільки він використовує асинхронну модель «публікація-підписка», пристрій надсилає повідомлення в певну чергу, а клієнти отримують лише ті повідомлення, які є для них актуальними, використовуючи механізм підписки. Такий підхід має багато переваг. По-перше, трафік не надто великий, оскільки ніхто не бачить нічого, крім того, на що підписався. По-друге, можливість

адаптуватися до змін у ситуації. MQTT має різні рівні якості обслуговування (QoS). У нашому випадку QoS1 означає, що повідомлення надсилається принаймні один раз. Це гарантує, що навіть у випадку короткого збою мережі інформація не загубиться. Хоча QoS2 більш надійний теоретично, він вимагає більшої кількості підтверджень і, як наслідок, збільшує навантаження на канал, що недоцільно в цьому контексті [12].

Локальний брокер зберігає збережені повідомлення, що дозволяє пристрою відразу відновити поточний стан системи після перезавантаження. Якщо edge-сервер перезапустився, оператор отримує останні дані замість порожньої карти.

Для передачі даних на сторону було використано механізм SSE (Server-Sent Events). Це спрощений протокол, котрий дозволяє серверу послідовно надсилати повідомлення браузеру. Цього достатньо для карти, котра відображає точки у міру надходження без складних систем синхронізації. SSE краще підходить для односпрямованого потоку даних і потребує менше ресурсів, ніж імплементація WebSocket [13].

SSE передає повідомлення у форматі текстових подій і підтримує тривале HTTP-з'єднання. Браузер ж автоматично намагається відновити зв'язок, якщо він втрачає його. Це означає, що модель можна використовувати в польових умовах, де відключення відбуваються часто. Для користувача це виглядає як стандартна карта, яка сама себе оновлює.

При цьому, варто зазначити, що всі операції що потребують ресурсів:

- підвантаження карти (або ж, у випадку роботи повністю оффлайн, інтерактивної сітки)
- відображення точок та їх фільтрування за параметрами
- зміна масштабу
- Делегуються на клієнт та проходять на пристрої самого користувача.

Результат можна побачити на рис. 2 де зображена панель керування під час активного функціонування системи.

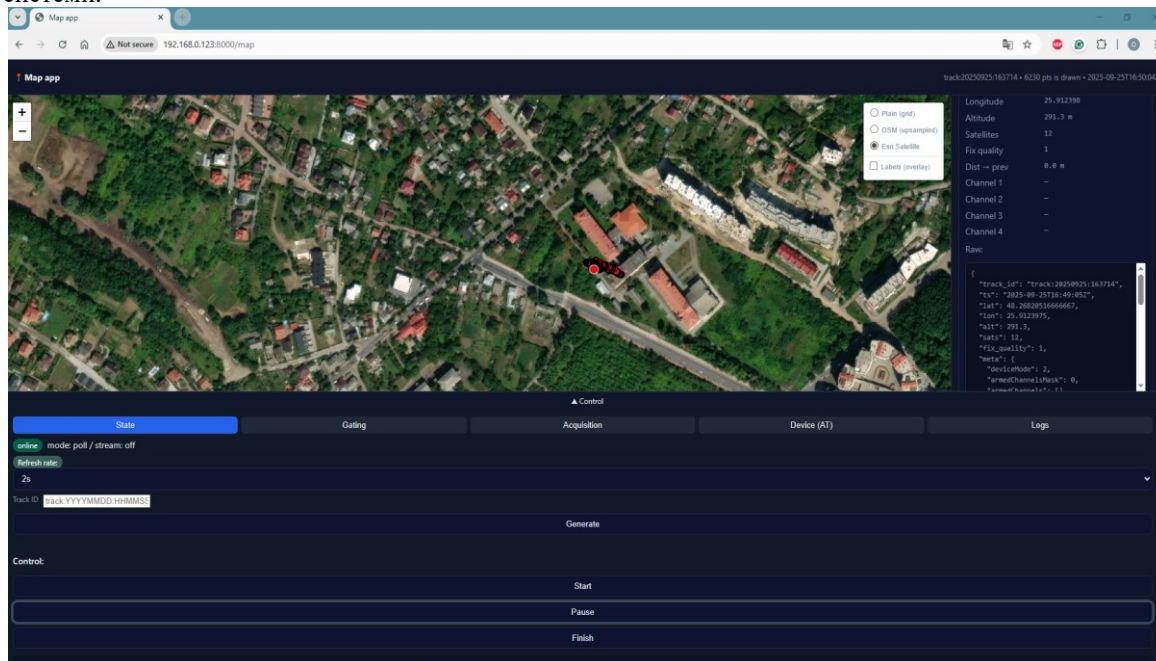


Рис. 2. Супутникова мапа та контрольна панель

Основним фоном обрано Супутниковий шар (Esri Satellite), а панель керування дозволяє запускати або зупиняти збір даних та контролювати технічні параметри у реальному часі.

Надійність є головною метою телекомунікаційного рівня системи. Дані дублюються в кількох точках, а використані протоколи розроблені таким чином, щоб мінімізувати ризик втрати інформації. Файлова система, WAL і локальний брокер працюють разом. Дані можна відновити навіть у випадку, якщо один компонент виходить з ладу.

Зважаючи на те, що мережа локальна і не має доступу до Інтернету, питання безпеки не є головним у базовій конфігурації. З іншого боку, при масштабуванні можливо включити TLS для MQTT, аутентифікацію клієнтів і доступ до VPN [14]. Це робить платформу сумісною з кращими практиками безпеки Інтернету речей.

Три основні компоненти складають телекомунікаційний рівень: Wi-Fi як середовище, протокол MQTT як транспортний протокол і технологія SSE як інтерфейсна технологія. Вони створюють контур разом, який дозволяє збирати дані сенсорів, зберігати їх у кількох копіях і надсилати їх операторам у зрозумілому вигляді. Цей рівень підходить для edge систем, оскільки він простий для налаштування, здатний відновлюватися після збоїв і має низьке навантаження на канал. Запропонований підхід відображено процесною діаграмою на рис. 3.

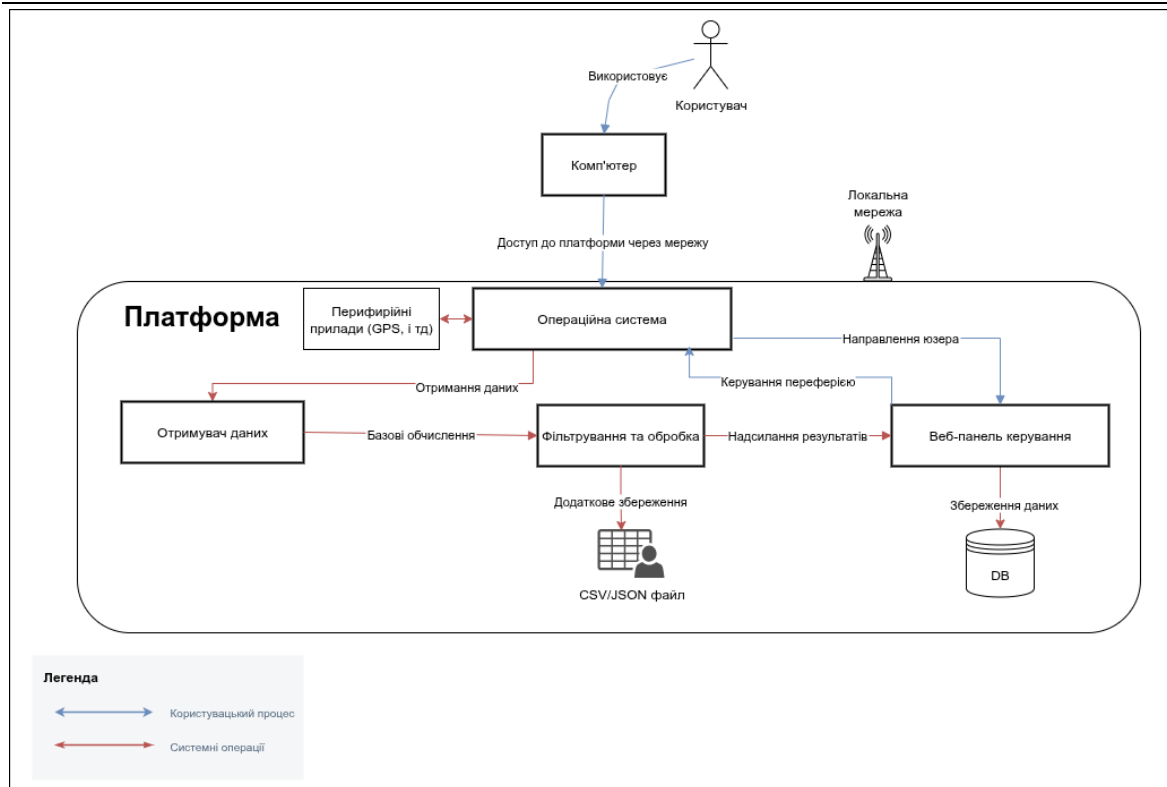


Рис. 3. Архітектура платформи — логічні шари та системні взаємодії

Використання єдиного носія дозволяє зменшити вагу, вартість і кількість деталей, що робить його більш надійним в експлуатації.

Ключовим елементом виступає модуль SoC FPGA. Він виконує одразу кілька завдань: керування сенсорами, синхронізацію імпульсів, попередню обробку даних і координацію польотного контролера. Завдяки цьому вдалося відмовитися від кількох зовнішніх процесорних модулів, спростивши інтерфейси (SPI, соах) і знизивши енергоспоживання. Обробка ж та візуалізація даних проходить на платформі Raspberry Pi [15].

Передача даних у польоті не прив'язана до супутникового інтернету.

На рис. 4 приведено панель керування, з обраною опцією прямої передачі команд на саму платформу, що дозволяє надавати більш детальний контроль користувачу. У якості інтерактивного фону було обрано звичний формат мапи. У нижній панелі виводу можна побачити відповідь платформи на команду відновлення до базових налаштувань.

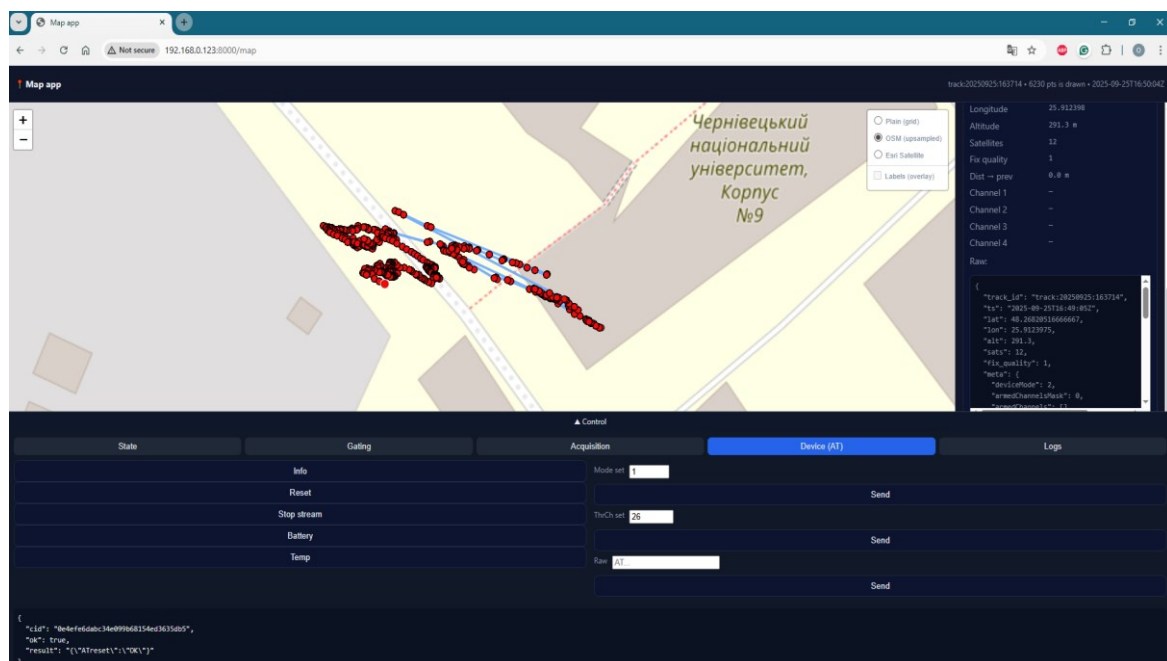


Рис. 4. Локальна карта OSM з панеллю команд X1

Варто зазначити що, при умові функціонування хоча би одного елементу системи, всі дані зберігаються на пристрої і можуть надсилатися до веб-інтерфейсу оператора в режимі реального часу. Даний підхід дає низку переваг, таких як:

- один носій замість двох;
- стійкість до втрат каналів зв'язку завдяки автономному режиму;
- менша складність і вартість;
- можливість роботи в складних умовах місцевості.

Оскільки обробка даних відбувається якомога ближче до їх джерела - це дозволяє їй працювати навіть за повної відсутності зовнішнього зв'язку. Локальний вузол виконує всі важливі завдання, такі як збір даних, їх синхронізація, реєстрація та виконання базової аналітики. Для операторів це означає можливість одразу отримати інформацію з датчиків без залежності від супутникових чи мобільних мереж, проте з втратою географічної деталізації на даний момент. На карті оператор бачить результат як точки з усією зібраною інформацією, це вже спростує прийняття рішень в моменті.

Варто ще зазначити, що процес отримання даних конфігурується з панелі керування оператором, а отже дозволяє створювати поріг для потужності сигналу, лише подолання якого є причиною обробки даної точки та збереження даних. Додатковий функціонал фільтрування що працює напямку на клієнті дозволяє динамічно виділяти точки, на яких сигнал пройшов встановлений поріг та додатковий поріг відображення що встановлюється в реальному часі.

Елементи налаштування дозволяють впливати на алгоритм обробки даних, проте загальні його кроки залишаються незмінними. Загалом, покроковий процес обробки даних відображено на рис. 5 нижче:

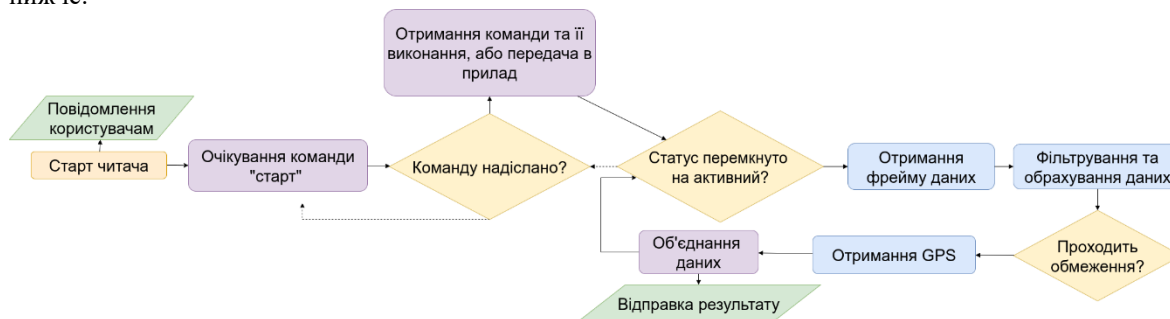


Рис. 5. Алгоритм обробки даних

Поєднання локальної обробки, офлайн-режиму та багатосенсорного підходу робить цей інструмент гнучким для роботи на місцях, де інші рішення не працюють. А здатність до експорту даних у форматі GeoJson [16] дозволяє використовувати функціонал GIS систем у випадках, де аналіз виходить за рамки можливі у даній форматі описаної платформи.

Висновки з даного дослідження і перспективи подальших розвідок у даному напрямі

Продемонстрована система функціонує як edge платформа, має характеристики мультисенсорного комплексу та гнучкої телекомунікаційної архітектури. Одним з найважливіших аспектів є те, що всі основні функції виконуються локально, від збору та синхронізації даних до їх зберігання та відображення в польовому інтерфейсі. Це робить рішення стійким до відсутності інтернету, перебоїв у каналах чи помилок оператора. Така автономність безпосередньо відповідає потребам задач у полі, де швидкий доступ до інформації часто є одним із критичних чинників роботи.

Поєднання Wi-Fi, MQTT і SSE створює надійну телекомунікаційну основу для роботи з декількома клієнтами у реальному часі. Водночас дані дублюються у кількох точках (WAL, файли, журнали брокера), що знижує ризик втрат навіть у разі відмови окремих вузлів. Особливу цінність система має завдяки простоті розгортання. Вона не потребує дорогого обладнання або складних процедур обслуговування. Усе рішення може бути відновлено чи перенесено у новий район з мінімальними витратами часу й ресурсів.

Майбутнє системи залежить від конфігурації її елементів. Наприклад, додавання тепловізійних або гіперспектральних модулів може збільшити кількість каналів датчиків. По-друге, архітектура дозволяє масштабувати рішення від одного дрона до мережі з декількох пристроїв, які всі підключені до спільно мережі. По-третє, використання відкритих протоколів дозволяє підключатися до більших аналітичних платформ і систем GIS.

Отже, описана система здатна виконувати поставлені задачі в обмежених умовах, надавати доступ користувачам у реальному часі, контролювати власні елементи відповідно до команд користувача та зберігати дані у декількох точках їх обробки. А описані вище способи масштабування та оновлення дозволяють будувати як глобально пов'язані системи, так і локально працюючі мультифункціональні edge системи що складаються як з одного так і багатьох елементів.

Подяка

Автори висловлюють подяку за підтримку проекту НФДУ № 2023.04/0150 «Розробка комплексу для визначення положення та відносної потужності джерел радіовипромінювання та їх візуалізації».

Література

1. Shi W., Cao J., Zhang Q., Li Y., Xu L. Edge computing: vision and challenges // IEEE Internet of Things journal. – 2016. – Vol. 3, No. 5. – P. 637–646.
2. Satyanarayanan M. The emergence of edge computing // Computer. – 2017. – Vol. 50, No. 1. – P. 30–39.
3. European Commission. Humanitarian demining and dual-use robotics programmes [Електронний ресурс]. – Brussels, 2023. – Режим доступу: <https://ec.europa.eu/research>.
4. Raspberry Pi Foundation. Edge deployment guide for IoT systems [Електронний ресурс]. – Cambridge, 2022. – Режим доступу: <https://www.raspberrypi.com/documentation>.
5. OASIS. MQTT version 5.0 specification [Електронний ресурс]. – 2019. – Режим доступу: <https://docs.oasis-open.org/mqtt/mqtt/v5.0>.
6. SQLite Consortium. Write-ahead logging (WAL) mode [Електронний ресурс]. – 2021. – Режим доступу: <https://sqlite.org/wal.html>.
7. IBM Developer. MQTT QoS explained [Електронний ресурс]. – 2022. – Режим доступу: <https://developer.ibm.com/articles/mqtt-qos>.
8. ETSI EN 303 645. Cyber security for consumer IoT: baseline requirements [Електронний ресурс]. – 2020. – Режим доступу: <https://www.etsi.org/standards>.
9. Tiang H., Wei J., Chao K. FastAPI in edge data pipelines [Електронний ресурс]. – 2022. – Режим доступу: <https://fastapi.tiangolo.com>.
10. IEEE 802.11 Working Group. Wi-Fi standard [Електронний ресурс]. – 2021. – Режим доступу: https://standards.ieee.org/802_11.
11. Zhou L., Wang T., Chen J. Wireless mesh networks for rugged environments // Journal of wireless communications and networks. – 2020. – No. 15. – P. 112–128.
12. IBM. MQTT architecture and applications: white paper [Електронний ресурс]. – 2021. – Режим доступу: <https://www.ibm.com/docs/en/mq>.
13. Chen Y. Performance comparison of SSE and WebSocket protocols [Електронний ресурс]. – 2022. – Режим доступу: <https://arxiv.org/abs/2206.145>.
14. ETSI TS 103 645. Cyber security for IoT devices [Електронний ресурс]. – 2019. – Режим доступу: <https://www.etsi.org/standards>.
15. Raspberry Pi Docs. Serial (UART) communication and peripheral control [Електронний ресурс]. – 2022. – Режим доступу: https://www.raspberrypi.com/documentation/computers/config_uart.html.
16. Open Geospatial Consortium (OGC). GeoJSON and web feature service standards [Електронний ресурс]. – 2023. – Режим доступу: <https://www.ogc.org/standards>.

References

1. Shi W., Cao J., Zhang Q., Li Y., Xu L. Edge computing: vision and challenges // IEEE Internet of Things journal. - 2016. - Vol. 3, No. 5. – P. 637–646.
2. Satyanarayanan M. The emergence of edge computing // Computer. - 2017. - Vol. 50, No. 1. – P. 30–39.
3. European Commission. Humanitarian demining and dual-use robotics programs [Digital resource]. – Brussels, 2023. – Access mode: <https://ec.europa.eu/research>.
4. Raspberry Pi Foundation. Edge deployment guide for IoT systems [Digital resource]. – Cambridge, 2022. – Access mode: <https://www.raspberrypi.com/documentation>.
5. OASIS. MQTT version 5.0 specification [Digital resource]. – 2019. – Access mode: <https://docs.oasis-open.org/mqtt/mqtt/v5.0>.
6. SQLite Consortium. Write-ahead logging (WAL) mode [Digital resource]. – 2021. – Access mode: <https://sqlite.org/wal.html>.
7. IBM Developer. MQTT QoS explained [Digital resource]. – 2022. – Access mode: <https://developer.ibm.com/articles/mqtt-qos>.
8. ETSI EN 303 645. Cyber security for consumer IoT: baseline requirements [Digital resource]. – 2020. – Access mode: <https://www.etsi.org/standards>.
9. Tiang H., Wei J., Chao K. FastAPI in edge data pipelines [Digital resource]. – 2022. – Access mode: <https://fastapi.tiangolo.com>.
10. IEEE 802.11 Working Group. Wi-Fi standard [Digital resource]. – 2021. – Access mode: https://standards.ieee.org/802_11.
11. Zhou L., Wang T., Chen J. Wireless mesh networks for rugged environments // Journal of wireless communications and networks. – 2020. – No. 15. – P. 112–128.
12. IBM. MQTT architecture and applications: white paper [Digital resource]. – 2021. – Access mode: <https://www.ibm.com/docs/en/mq>.
13. Chen Y. Performance comparison of SSE and WebSocket protocols [Digital resource]. – 2022. – Access mode: <https://arxiv.org/abs/2206.145>.
14. ETSI TS 103 645. Cyber security for IoT devices [Digital resource]. – 2019. – Access mode: <https://www.etsi.org/standards>.
15. Raspberry Pi Docs. Serial (UART) communication and peripheral control [Digital resource]. – 2022. – Access mode: https://www.raspberrypi.com/documentation/computers/config_uart.html.
16. Open Geospatial Consortium (OGC). GeoJSON and web feature service standards [Digital resource]. – 2023. – Access mode: <https://www.ogc.org/standards>.