Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра автоматики та управління в технічних системах

Лабораторна робота №1

з дисципліни «Інтелектуальні вбудовані системи»

Виконав:

студент групи ІА-02

Гузь Богдан

Перевірив:

Нікольський С. С.

Київ 2024

Постановка задачі

Відбувається розробка програмного забезпечення, яке є частиною великої системи Розумного Міста. А саме розробка Agent частини, яка збирає дані отримані з датчиків вбудованих в сучасні машини для моніторингу стану дорожнього покриття. Проте не всі розробники маю доступ до датчиків для роботи з ними.

Я як розробник хочу мати змогу розробляти систему без фізичного пристою, датчиків (акселерометер) які видають інформацію.Потрібнно абстрагуватися від конкретного джерела інформації або емулювати роботу датчиків.

Для рішення цієї задачі потрібно реалізувати програмну частину яка буде читати дані датчиків з файлу (до цього записаний csv file під назвою data.csv) та надсилати їх на Edge. В нашому випадку Agent та Edge будуть комунікувати через MQTT.

**Хід роботи:**

1.Створення віртуального середовища. Пакет virtualenv повинен бути встановлений в системі.

python -m venv venv

2. Активація середовища

source ./venv/bin/activate # linux

.\venv\Scripts\activate.bat # windows

3. Встановимо необхідні залежності(бібліотеки)

pip install paho-mqtt==1.6.1 # mqtt client

pip install marshmallow # for schema and converting object to json

4. Збережемо залежності в requirements.txt

pip freeze > requirements.txt

5.Основну логіку, а саме file\_datasource.py потрібно реалізувати

самостійно. Перед читанням даних відкрити файли, при читанні

зчитувати відповідний набір даних з файлів та повертати його.

В \_\_init\_\_ ми ініціалізуємо змінні шляхів для наших файлів з данними.

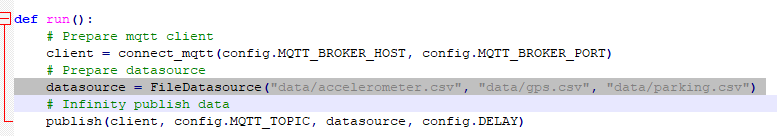
initialize\_reader - додатковий клас для відкриття і читання файлів.

startReading - ""Метод повинен викликатись перед початком читання даних""", з initialize\_reader отримуємо відкриті файли і об\*єкти з данними для читання і зберігаємо їх в змінні

stopReading - закриває файли для читання

read - читає данні з файлів за допомогою \_\_next\_\_(),що отримані з startReading , створює об\*єкти AggregatedData який містить прочитані дані

В main.py було добавлено шлях до parking\_filename у методі run()

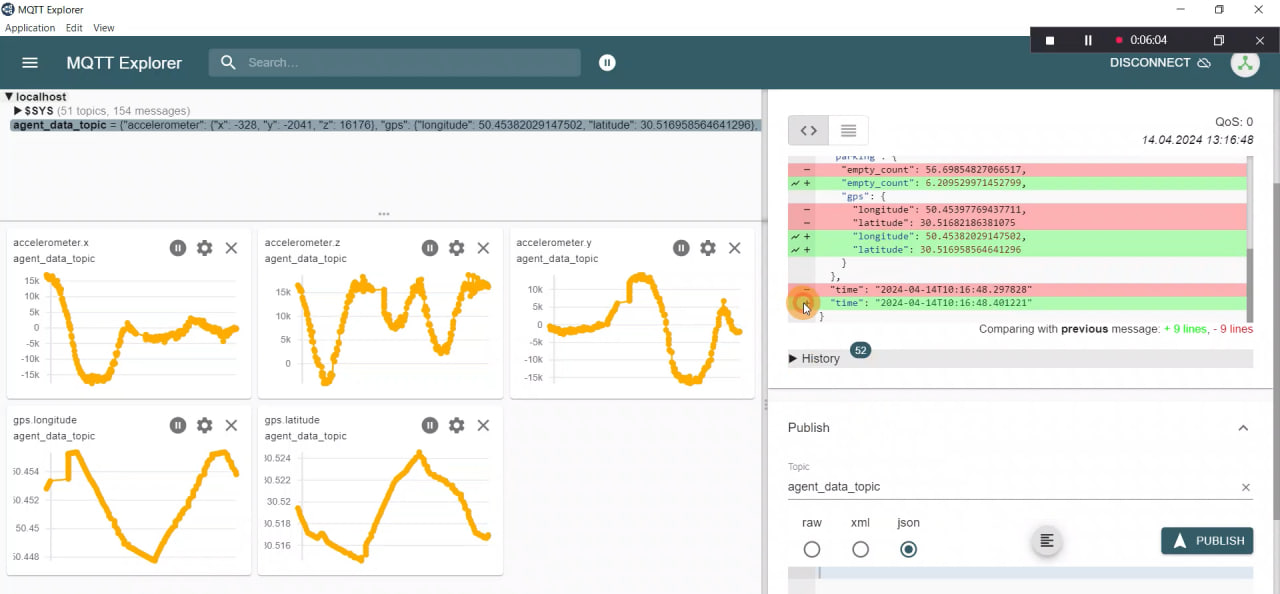


6.Запуск програму та mqtt в docker

cd docker

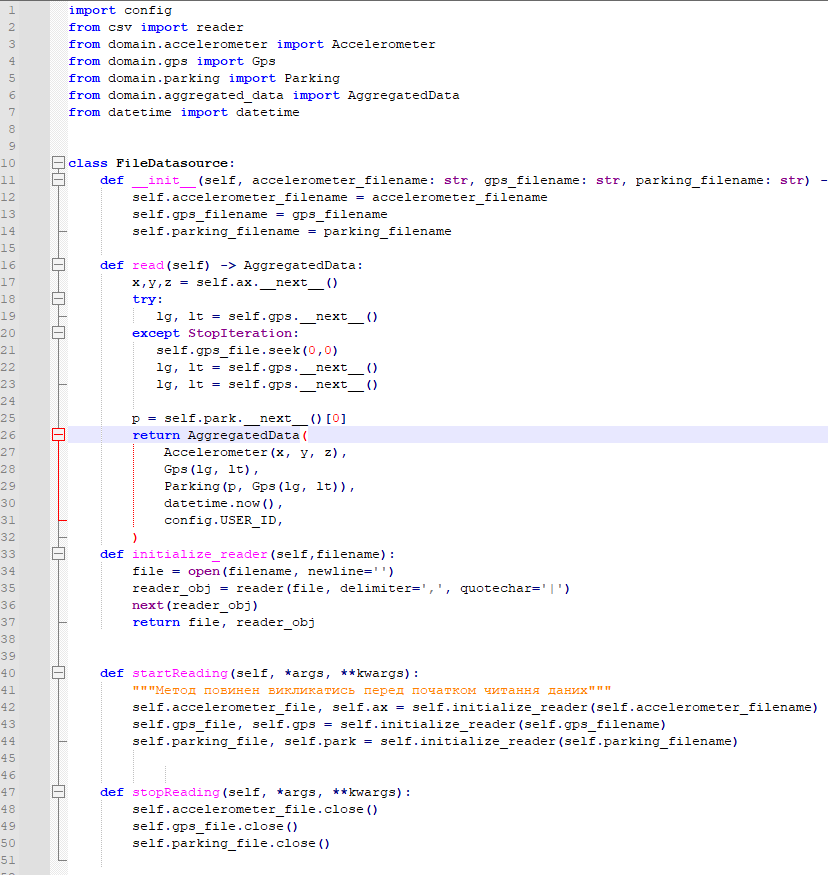
docker-compose up --build

1. Перевірка працездатності в MQTT Explorer

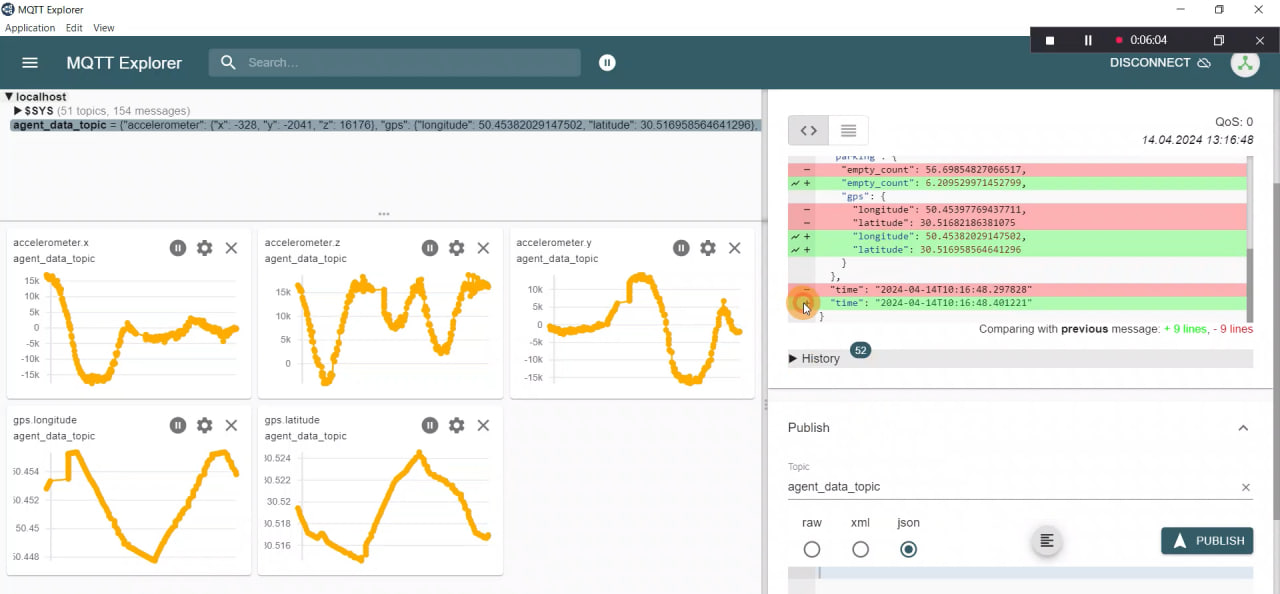


1. Очікувані результати в протоколі

* Посилання на git репозиторій з проектом
* Реалізація FileDatasource



* Показ результатів роботи в MQTT Explorer



* Висновки

В даній лабораторній роботі було потрібно реалізувати програмну частину яка буде читати дані датчиків з файлу (до цього записаний csv file під назвою data.csv) та надсилати їх на Edge.

Було реалізовано FileDatasource,добавлена таблиця даних parking.csv

розширений клас FileDatasource для читання даних з файлу

parking\_filename та відповідно розширити методи read, startReading,

stopReading,розширили логіку в main.py для відправки даних в відповідні топіки