

Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний
інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

З лабораторної роботи № 1 з дисципліни
«Технології розробки вбудованих IoT системи»

„MQTT Explorer”

Перевірив

(прізвище, ім'я, по батькові)

Київ 2023

- 1) Посилання на git репо: <https://github.com/faaant/iot>
- 2) Реалізація класу FileDatasource та його нащадків для нескінченного послідовного читання даних

```
import config
from csv import reader
from datetime import datetime
from typing import Dict, Tuple
from domain.accelerometer import Accelerometer
from domain.parking import Parking
from domain.gps import Gps

class FileDatasource:
    def __init__(self, filename: str) -> None:
        self.filename = filename
        self.row_counter = 0
        self.file_content = []

    def read(self):
        # infinite read
        self.row_counter %= len(self.file_content)
        data = self.file_content[self.row_counter]
        # read next row
        self.row_counter += 1

        print(data)

        return data

    def startReading(self, *args, **kwargs):
        with open(self.filename, "r") as file:
            self.file_content = list(reader(file))[1:]

    def stopReading(self, *args, **kwargs):
        self.row_counter = 0

class AccelerometerFileDatasource(FileDatasource):
    def __init__(self, accelerometer_filename: str) -> None:
        super().__init__(accelerometer_filename)

    def read(self) -> Accelerometer:
        data = super().read()
        return Accelerometer(
            x=int(data[0]),
            y=int(data[1]),
            z=int(data[2]),
        )
```

```

class GpsFileDatasource(FileDatasource):
    def __init__(self, gps_filename: str) -> None:
        super().__init__(gps_filename)

    def read(self) -> Gps:
        data = super().read()
        return Gps(
            latitude=float(data[0]),
            longitude=float(data[1]),
        )

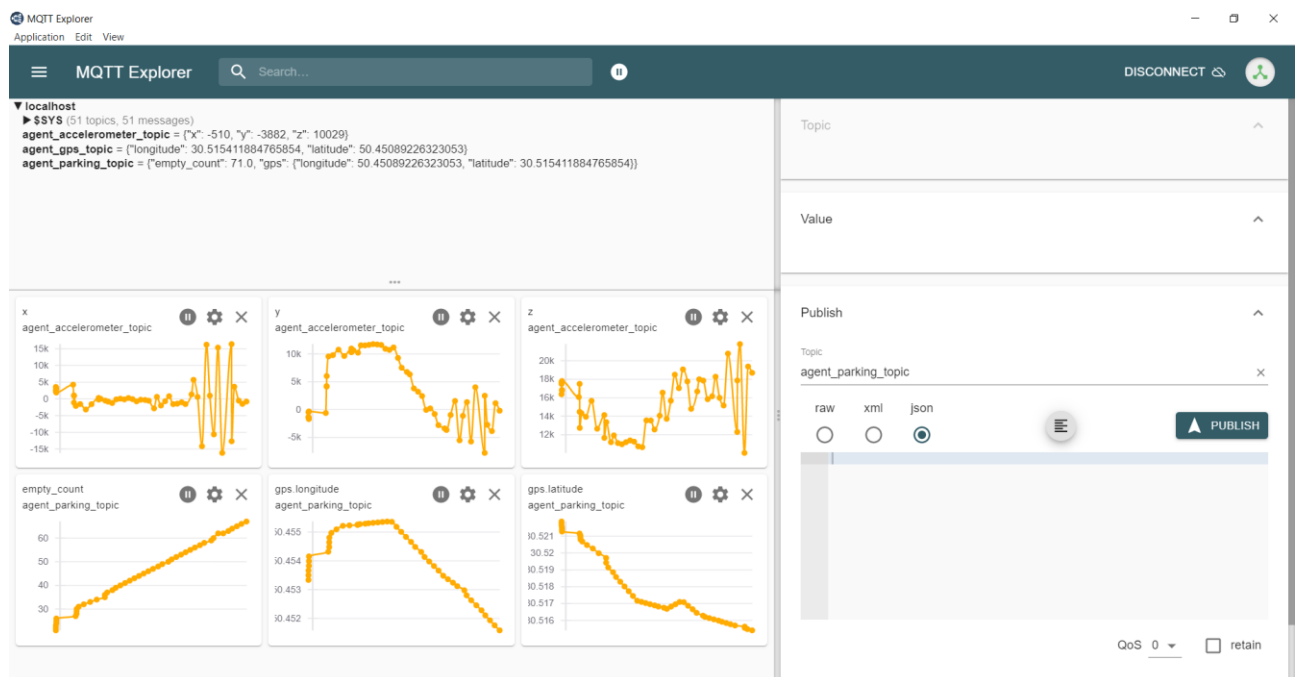
class ParkingFileDatasource(FileDatasource):
    def __init__(self, parking_filename: str) -> None:
        super().__init__(parking_filename)

    def read(self) -> Parking:
        data = super().read()

        return Parking(
            empty_count=int(data[0]),
            gps=Gps(
                longitude=float(data[1]),
                latitude=float(data[2])
            )
        )

```

3) Результати роботи в MQTT Explorer



Висновок: під час виконання даної лабораторної роботи ми ознайомились з MQTT Explorer та його застосуванням, для відображення історії змін даних. З

допомогою мови Python реалізували зчитування даних з файлів та надсилання їх на mqtt клієнт під різними ключами(topic).