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 repo: <https://github.com/faaant/iot>
2. Реалізація класу FileDatasource та його нащадків для нескінченого 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(

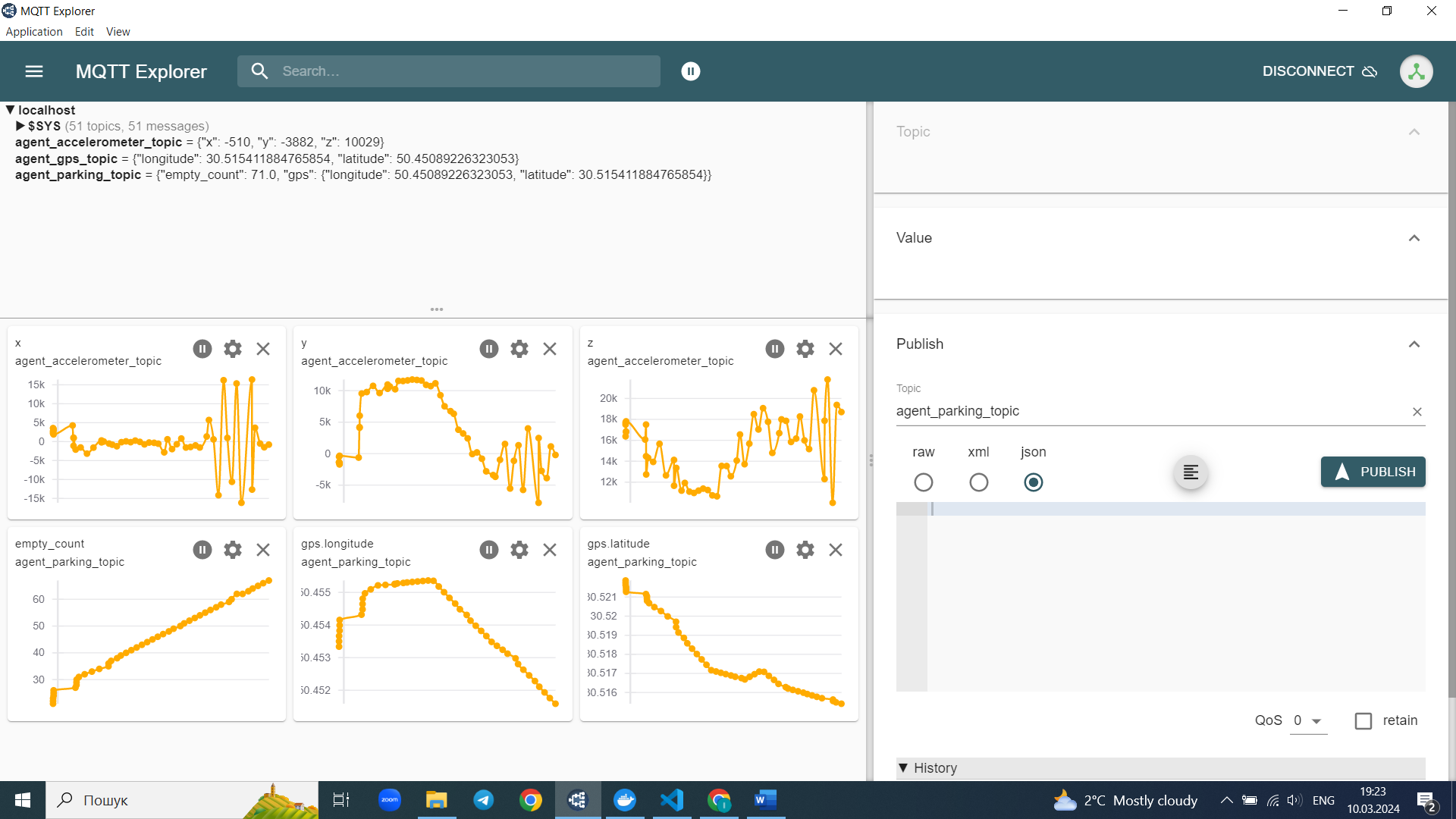
                longitude=float(data[1]),

                latitude=float(data[2])

            )

        )

1. 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).