WOJSKOWA AKADEMIA TECHNICZNA

**im. Jarosława Dąbrowskiego**

**WYDZIAŁ CYBERNETYKI**



STUDIA I°

Temat: **Sprawozdanie z realizacji ćwiczenia laboratoryjnego**

**INFORMATYKA**

…………………………………………………………………………

(kierunek studiów)

**INŻYNIERIA SYSTEMÓW**

…………………………………………………………………………

(specjalność)

|  |  |
| --- | --- |
| Wykonał: | Prowadzący: |
| **Radosław Relidzyński** | **mgr inż. Tomasz Gumowski** |

**Warszawa 2023**

Spis treści

[Rozdział I. Treść zadań 4](#_Toc149478913)

[Rozdział II. Rozwiązanie zadania 1 4](#_Toc149478914)

[II.1. Implementacja zadania 1 4](#_Toc149478915)

[II.2. Opis działania zadania 1 4](#_Toc149478916)

[II.3. Prezentacja działania zadania 1 4](#_Toc149478917)

[Rozdział III. Rozwiązanie zadania 2 4](#_Toc149478918)

[III.1. Implementacja zadania 2 4](#_Toc149478919)

[III.2. Opis działania zadania 2 4](#_Toc149478920)

[III.3. Prezentacja działania zadania 2 4](#_Toc149478921)

[Rozdział IV. Podsumowanie 4](#_Toc149478922)

1. Treść zadań

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

1. Rozwiązanie zadania 1
   1. Implementacja zadania 1
      1. Publisher.py:

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

* + 1. Subscriber1.py:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

* + 1. Subscriber2.py:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

* + 1. Subscriber3.py:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

* 1. Opis działania zadania 1

Są 3 sensory:

* Vibration: mało istotny, sprawdzający wibracje reaktora
* Pressure: średnio istotny, sprawdzający ciśnienie w reaktorze
* Radiation: bardzo istotny, sprawdzający poziom promieniowania radioaktywnego

Każdy subscriber podłącza się do danego sensora (każdy przy pomocy innego qos) i w nieskończonej pętli nasłuchuje nadchodzących wartości.

Publisher wysyła 3 sygnały (po 1 na sensor) w taki sposób, żeby każdy subscriber otrzymał odpowiednią dla niego wartość.

Subscriber1 działa przy użyciu qos 0 (Zero Assurance), wiadomość jest przesyłana bez upewniania się, że dotrze. Nie sprawdza on dostępności odbiorcy.

Mechanizm qos 0 sprawdza się przy mało krytycznych sensorach.

Subscriber2 działa przy użyciu qos 1 (At Least One), gwarantuje, że wiadomość zostanie dostarczona co najmniej raz. Wysyła tą samą wiadomość do momentu potwierdzenia odbioru.

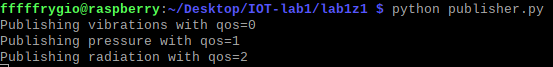
Mechanizm qos 0 sprawdza się przy średnio krytycznych sensorach, przy których dostarczenie danych jest niezbędne.

Subscriber3 działa przy użyciu qos 2 (Exactly One), gwarantuje, że wiadomość zostanie dostarczona dokładnie raz, bez duplikacji. Nawiązuje trwałe połączenie, a wiadomość oznacza unikatowym identyfikatorem. Przy pomocy połączenia upewnia się, że wiadomość została dostarczona.

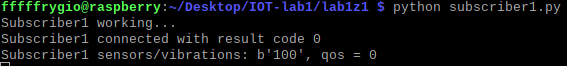
Mechanizm qos 0 sprawdza się przy krytycznych sensorach, przy których dostarczenie danych jest niezbędne oraz to w jaki sposób są przekazywane (pod względem chociażby ilości przesyłanych wiadomości).

* 1. Prezentacja działania zadania 1

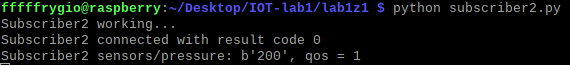
Publikowanie wiadomości:



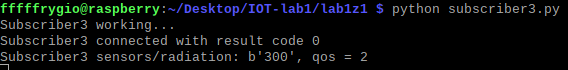
Nasłuchiwanie 1:



Nasłuchiwanie 2:



Nasłuchiwanie 3:



1. Rozwiązanie zadania 2
   1. Implementacja zadania 2
      1. fastapi\_app.py:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

* + 1. fastapi\_database\_app.py

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Opis wygenerowany automatycznie Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

* + 1. fastap\_user.py

Obraz zawierający tekst, zrzut ekranu, oprogramowanie

Opis wygenerowany automatycznie

* 1. Opis działania zadania 2

Serwer fastapi posiada mechanizm przechwytywania zapytań http i na ich podstawie obsługuje system (w tym przypadku bazę danych) w odpowiedni dla danego zapytania sposób.

Serwer wystawiany jest pod adresem <http://0.0.0.0:8000/>

Przyjmuje on różne zapytania http (get, post, delete) i na ich podstawie wykonuje odpowiednią metodę.

Zapytania get zwracają odpowiednią zawartość z bazy danych

Zapytania post dodają elementy do bazy danych poprzez generowanie wartości lub przejmowanie przekazywanych danych w zapytaniu.

Zapytanie delete usuwa zawartość bazy danych

Zaimplementowany jest skrypt symulujący działanie użytkownika, który:

1. Dodawanie elementów.
2. Pobieranie elementów do wypisania.
3. Usunięcie elementów.
4. Sprawdzenie, czy elementy są usunięte.
   1. Prezentacja działania zadania 2

Uruchomienie serwera:

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Uruchomienie użytkownika:

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Przegląd przechwyconych zapytań przez serwer:

Obraz zawierający tekst, zrzut ekranu

Opis wygenerowany automatycznie

1. Podsumowanie

Oba zadania dostarczyły praktycznej wiedzy na temat implementacji mechanizmów technologii Internetu rzeczy. Celem ćwiczenia było utworzenie dwóch małych systemów wprowadzających oraz obsługujących wiadomości wysyłane pomiędzy nadawcą (sensorem lub użytkownikiem), a odbiorcą (subscriberem lub serwerem fastapi). Każdy z systemów udało się stworzyć i jest w pełni działający.

Ćwiczenie pozwoliło zrozumieć jak działają mechanizmy obsługi sensorów przy pomocy usługi QoS oraz jak działają zapytania http.