POLITECHNIKA POZNAŃSKA

Wydział Automatyki, Robotyki i Elektrotechniki Instytut Robotyki i Inteligencji Maszynowej Zakład Sterowania i Elektroniki Przemysłowej



ZADANIE ZALICZENIOWE - SERWER IOT

APLIKACJE MOBILNE I WBUDOWANE DLA INTERNETU RZECZY

RAPORT LABORATORYJNY

SZYMON KWASIBORSKI 140270, MIŁOSZ PLUTOWSKI 140299 SZYMON.KWASIBORSKI@PUT.POZNAN.PL, MIŁOSZ.PLUTOWSKI@PUT.POZNAN.PL

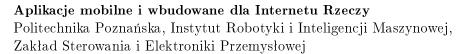
JAKUB GĄTARSKI 140241, FILIP KAŁUŻNY 140 252 JAKUB.GATARSKI@PUT.POZNAN.PL, FILIP.KALUZNY@PUT.POZNAN.PL

Prowadzący:

Mgr inż. Adrian Wójcik

Adrian.Wojcik@put.poznan.pl

01-07-2021





Spis treści

W	$V_{ m step}$	3
1	Opis specyfikacji	3
2	Implementacja systemu2.1 Aplikacje serwera2.2 Mobilna aplikacja klienta2.3 Webowa aplikacja klienta2.4 Desktopowa aplikacja klienta	4 4 4
3	Wyniki testów i integracji systemu	5
4	Wnioski i podsumowanie	6
Ri	i bliografia	7



WSTEP

Raport wykonanego projektu wykonano w oparciu o szablon pobrany z platformy uczelnianej eKursy [1]. Do jego wykonania posłużyliśmy się wiedzą nabytą podczas kursu: Aplikacje mobilne i wbudowane dla Internetu Rzeczy - laboratorium.

Poniższe sprawozdanie podzielono na cztery główne sekcje, podobne do tych używanych podczas wykonywania sprawozdań z poszczególnych instrukcji laboratoryjnych, w trakcie roku akademickiego.

Opis specyfikacji

Na samym wstępie opiszemy jakie wymogi (rzecz jasna poza podstawowymi) chcemy zrealizować w naszym projekcie. W późniejszych sekcjach raportu dotyczących opisu implementacji poszczególnych aplikacji, zaznaczymy czy zamierzona funkcjonalność (dany wymóg) została spełniona.

- 1. Stworzony system wykorzystywać będzie architekturę REST
- 2. Wszystkie trzy środowiska zachowają analogiczną architekturę oraz nazewnictwo metod
- 3. Kod źródłowy będzie zawierał komentarze według wspólnego standardu
- 4. Po uruchomieniu serwera odbędzie się automatyczne uruchomienie skryptów serwera
- 5. Każda z aplikacji umożliwiać będzie próbkowanie danych z okresem maksymalnie 100ms
- 6. Aplikacje serwera pozwalać będą na podgląd wszystkich wielkości fizycznych odczytanych z czujników
- 7. Podczas realizacji, implementacji wykorzystamy system kontroli wersji GitHub
- 8. Aplikacja mobilna wykorzysta wzorzec architektoniczny zapewniający separację interfejsu użytkownika od logiki aplikacji
- 9. Aplikacja desktopowa wykorzysta wzorzec architektoniczny zapewniający separację interfejsu użytkownika od logiki aplikacji
- 10. Wszystkie stworzone aplikacji posiadać będą jednolitą szatę graficzną



Implementacja systemu

2.1 Aplikacje serwera

Nasz projekt wykonaliśmy na fizycznym urządzeniu Raspberry Pi z dołączoną nakładką SenseHat. Nasz serwer składa się z nieskończonej pętli (plik serverAPP_INF.py), w której przeprowadzany jest odczyt ze wszystkich czujników dostępnych dzięki nakładce SenseHat, a następnie zapis odczytanych danych do plików (przykładowe dane zapisane do pliku ukazuje rysunek 1). W pętli obsługiwany jest także joystick. Na serwerze umieszczono także skrypty w językach python oraz php odpowiedzialne za wyświetlacz LED oraz zliczanie kliknięć joysticka.

```
{"name": "accelerometer_pitch", "value": 359.0, "unit": "deg", "sensor": "accelerometer"}
```

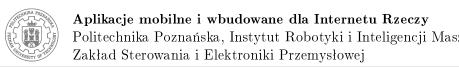
Rys. 1. Plik zawierający dane odczytane z akcelerometru

Do uruchomienia naszego serwera posłużyliśmy się serwisem, który startuje po każdym ponownym uruchomieniu Raspberry (możliwe dzięki komendzie enable). Serwis może być także uruchamiany bezpośrednio z terminala za pomocą komendy (listing 1):

Listing 1. Instrukcja uruchamiająca serwis

```
01. sudo systemct1 start onStart.service
```

- 2.2 Mobilna aplikacja klienta
- 2.3 Webowa aplikacja klienta
- 2.4 Desktopowa aplikacja klienta



Wyniki testów i integracji systemu



Wnioski i podsumowanie



BIBLIOGRAFIA

1. Szablon sprawozdania. Dostępne także z: https://ekursy.put.poznan.pl.