

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

Wstęp	3
1 Opis specyfikacji	3
2 Implementacja systemu	4
2.1 Aplikacje serwera	4
2.2 Mobilna aplikacja klienta	5
2.3 Webowa aplikacja klienta	6
2.4 Desktopowa aplikacja klienta	6
3 Wyniki testów i integracji systemu	7
4 Wnioski i podsumowanie	8
Bibliografia	9

WSTĘP

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 wymagania (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 (doxygen)
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 aplikacje 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 (listing 1) 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.

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

Rys. 1. Plik zawierający dane odczytane z akcelerometru

Listing 1. Przykładowa instrukcja pobrania oraz zapisania danych z czujnika wilgotości

```
01. dict_data['name'] = 'humidity'
02. dict_data['value'] = round(sense.get_humidity(), 2)
03. dict_data['unit'] = '%'
04. dict_data['sensor'] = 'humidity_sensor'
05. tmp = dict_data
06. tmp_json = json.dumps(tmp)
07. # print(tmp_json)
08. # save to file
09. try:
10.     file = open("OneByOne/hum_p.json", "w")
11.     file.write(tmp_json)
12. except:
13.     print("Write_Error_RPY")
14. finally:
15.     file.close()
```

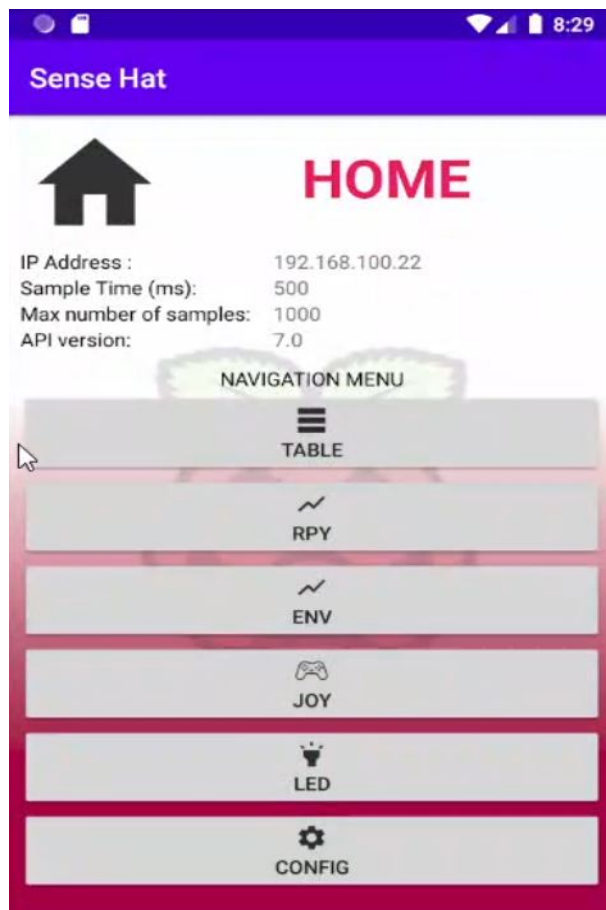
Do uruchomienia naszego serwera posłużyliśmy się serwisem, który startuje po każdym ponownym uruchomieniu Raspberry (możliwe dzięki komendzie **enable**). Serwis może być także uruchamiany bezpośrednio z terminala za pomocą komendy (listing 2):

Listing 2. Instrukcja uruchamiająca serwis

```
01. sudo systemctl start onStart.service
```

2.2 MOBILNA APLIKACJA KLIENTA

Implementację naszej aplikacji mobilnej rozpoczęliśmy od stworzenia strony startowej (rysunek 2), dzięki której możliwa będzie nawigacja pomiędzy poszczególnymi funkcjonalnościami. Po kliknięciu w odpowiedni przycisk przenosi on nas na odpowiadającą mu stronę. Każda kolejna strona została wykonana zgodnie z jej przeznaczeniem. Menu jest intuicyjne, a wszystkie z przycisków klarownie opisane.



Rys. 2. Menu nawigacyjne aplikacji mobilnej

Przejdźmy teraz do omówienia poszczególnych stron w aplikacji. Pierwsza z nich zawiera tabelę, na której użytkownik ma możliwość wyboru jakie dane chce wyświetlić oraz z jakim czasem próbkowania. Użytkownik może także zmienić wyświetlane jednostki za pomocą suwaka. Na samym dole strony widnieją przyciski:

- START - uruchamia proces pobierania danych i wyświetla je odświeżając co czas próbkowania
- STOP - zatrzymuje proces pobierania danych
- REFRESH - jednokrotnie pobiera i wyświetla dane w tabeli
- CLEAR - czyści tabelę z wszystkich wartości

Druga z kolei strona prezentuje wykresy położenia (raw, pitch i roll). Tutaj także użytkownik może zdecydować, które z przebiegów chce wyświetlić, po czym klikając START przebiegi będą się wyświetlać prezentując wartości odświeżane co ustawiony okres próbkowania (użytkownik może szybko przenieść się do menu konfiguracyjnego za pośrednictwem przycisku CONFIG gdzie możliwa jest zmiana okresu próbkowania).

Strona trzecia działa analogicznie jak ta opisana powyżej.

Strona z joystickiem zawiera wykres płaszczyzny x oraz y, które to odpowiadają wartością zmiennych odpowiednio Counter_X i Counter_Y. Poniżej widnieje zmienna zliczająca ile razy joystick został wciśnięty. Przyciski na samym dole strony odpowiadają za uruchomienie lub zatrzymanie timera, który umożliwia pobieranie wspomnianych wartości na bieżąco. Znajdziemy tam także przycisk REFRESH, dzięki któremu trzy wartości odpowiadające za bieżący stan joysticka są pobierane i wyświetlane jednokrotnie.

Obsługa matrycy LED została zaimplementowana jako graficzna wizualizacja bieżącego stanu matrycy 8x8 z nakładki SenseHat. Pod nią natomiast znajdują się suwaki ustawiające kolor w postaci RGB. Następnie mając wybrany kolor klikamy na diody jakie chcemy zapalić. Wysłanie tego żądania odbywa się poprzez naciśnięcie przycisku SEND. Przycisk CLEAR natomiast wyłącza nam wszystkie diody.

Strona odpowiedzialna za konfigurację zawiera cztery pola, w które użytkownik ma możliwość ustawienia:

- adresu IP
- Czasu próbkowania
- Maksymalna liczba próbek
- Wersja API

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>.