

Politechnika Wrocławska
AiR ARR
Projekt zespołowy

SENSGLOVE

RAPORT Z PRZEBIEGU PROJEKTU

Autorzy:
Beata Berajter
Dawid Brząkała
Dorota Gidel
Katarzyna Wądrzyk
Ada Weiss
Małgorzata Witka-Jeżewska

Prowadzący: dr inż. Krzysztof Arent

5 czerwca 2017

Spis treści

| | | |
|----------|---|----------|
| 1 | Kryteria ewaluacji | 2 |
| 2 | Przebieg prac dla poszczególnych komponentów | 2 |
| 2.1 | Rękawiczka sensoryczna | 2 |
| 2.1.1 | Wykonanie rękawiczki | 2 |
| 2.1.2 | Wprowadzone zmiany | 3 |
| 2.1.3 | Kryteria ewaluacji | 3 |
| 2.2 | Interfejs sprzętowy | 3 |
| 2.2.1 | Schemat płytki pcb i wyprowadzenie pinów układu | 3 |
| 2.2.2 | Kryteria ewaluacji | 3 |
| 2.3 | Program do akwizycji danych | 5 |
| 2.4 | Opis klas programu | 5 |
| 2.4.1 | Opis działania programu | 5 |
| 2.4.2 | Format danych | 7 |
| 2.4.3 | Raport z ewaluacji | 7 |
| 2.5 | Baza danych | 8 |
| 2.5.1 | Wprowadzone zmiany | 8 |
| 2.5.2 | Opis działania skryptu | 8 |
| 2.5.3 | Kryteria ewaluacji | 8 |
| 2.6 | Wizualizacja danych | 8 |
| 2.6.1 | Ogólny diagram klas | 9 |
| 2.6.2 | Diagram klas zakładek | 9 |
| 2.6.3 | Interfejs użytkownika | 10 |
| 2.6.4 | Kryteria ewaluacji | 11 |
| 2.6.5 | Brakujące elementy | 11 |

1 Kryteria ewaluacji

Komponenty powinny działać każde z osobna oraz współdziałać jako kompletne stanowisko do pobierania sygnałów i biosygnałów. Użytkownik powinien mieć możliwość wykonywania ruchów w obrębie przynajmniej jednego metra od stanowiska. Rękawiczka pomiarowa powinna mieć umieszczone czujniki w taki sposób, aby nie ograniczała ruchów ani tym bardziej nie kępowała ich. Pomiary przesyłane mają być w czasie rzeczywistym, a ich podgląd zapewnić ma program wizualizujący je.

1. Założenie rękawiczki.
2. Poruszanie palcami.
3. Obserwowanie przebiegów pomiarów.
4. Rozpoczęcie pomiaru.
5. Zapis pomiaru.
6. Sprawdzenie katalogu, w którym dokonano zapisu.

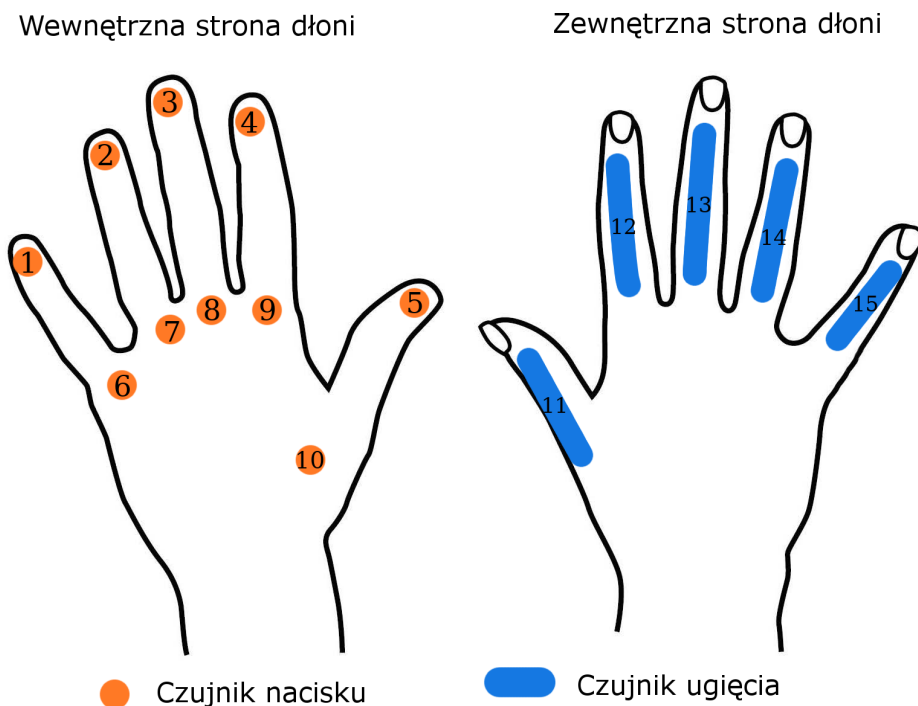
2 Przebieg prac dla poszczególnych komponentów

2.1 Rękawiczka sensoryczna

Osoby przydzielone do zadania: Dorota Gidel, Katarzyna Wądrzyk.

2.1.1 Wykonanie rękawiczki

Na rękawiczce wykonanej z poliestru przymocowano 10 czujników nasicku oraz 5 czujników ugięcia. Umieszczenie czujników zostało przedstawione na rysunku 1.



Rysunek 1: Umieszczenie i numeracja czujników na rękawiczce

Proces wykonania rękawiczki polegał na przylutowaniu przewodów do czujników, następnie przyklejeniu punktowo klejem czujników w odpowiednich miejscach na rękawicze oraz przyszycie przewodów do rękawiczki, w przypadku czujników zgięcia przyszyto również część czujnika by mógł się poruszać.

Przyszycie przewodów pozwala na zachowanie porządku i sprzyja ruchom osoby używającej rękawiczkę a przytwierdzenie klejem zapewnia trwałość. W celu zapobiegania uszkodzenia czujników (przede wszystkim haczeniu) uszyte zostały "kapturki" na palce. Dzięki nim sensory nacisku jak i także ugięcia są zabezpieczone przed uszkodzeniami.

Na końcach przewodów odprowadzających sygnały z czujników zamontowane zostały żeńskie końce, które umożliwiają proste podłączenie ich do płytki wzmacniającej. Przymocowane czujniki numerowane są w kolejności przedstawionej na rysunku 1.

2.1.2 Wprowadzone zmiany

Planowane było kupienie rękawiczki w rozmiarze S, ze względu na skład grupy, jednakże ostatecznie wybrano rozmiar M. Rękawiczka nadal dobrze dopasowuje się na dłoniach małych ale dodatkowo może z niej korzystać większe grono osób. Dzięki czemu do tworzenia bazy danych nie będzie potrzeby specjalnie wybierania i szukania osób o danym rozmiarze dłoni.

Zaletą większego rozmiaru jest też większa powierzchnia do mocowania czujników. Dużo łatwiej jest rozmieścić wszystkie sensory, po ich przymocowaniu nie ograniczają one ruchów w tak dużym stopniu jakby miało to miejsce przy rękawicze o rozmiarze S.

2.1.3 Kryteria ewaluacji

Na obecnym etapie projektu rękawiczka jest już po wstępnej ewaluacji i wprowadzane zostają poprawki. Jeden czujnik z wewnętrznej części dłoni został uszkodzony. Kilka czujników nacisku zostało odczepionych, ponieważ zostały przyklejone za nisko na placu, co powodowało złe odczyty przy chwytaniu przedmiotów. W tej chwili rękawiczka spełnia następujące kryteria:

1. Rękawiczka założona na rękę pozwala na możliwie swobodne poruszanie palcami.
2. Rezystancja czujników nacisku wraz ze wzrostem siły maleje, a wraz ze stopniem zgięcia rezystancja czujników zgięcia rośnie.
3. Przy chwytaniu przedmiotów czujniki ugięcia działają sprawnie.
4. Podczas wykonywania ruchów czujniki oraz przewody nie haczą się o nic dzięki dołączonym kapturkom.
5. Po nałożeniu kapturków sensory na wewnętrznej części dłoni sprawnie odczytują siłę nacisku.
6. Sygnały z czujników prowadzone przez przewody mogą być bezpośrednio podłączone do styków na wzmacniaczu dzięki ich żeńskim końcówkom.

Pomyślnie przeprowadzono próbę integracji rękawiczki z prototypem płytki wzmacniającej. Miernikiem mierzone napięcie, które jak oczekiwano zmieniało odpowiednio swoją wartość. Sprawdzone zostały zarówno czujniki zgięcia jak i nacisku.

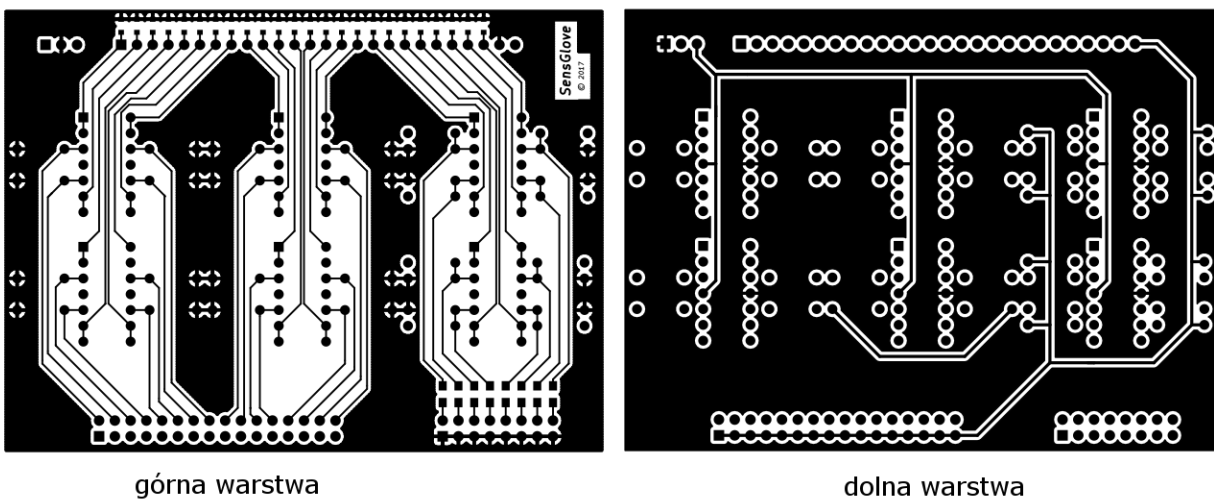
2.2 Interfejs sprzętowy

Osoby przydzielone do zadania: Beata Berajter i Dawid Brząkała.

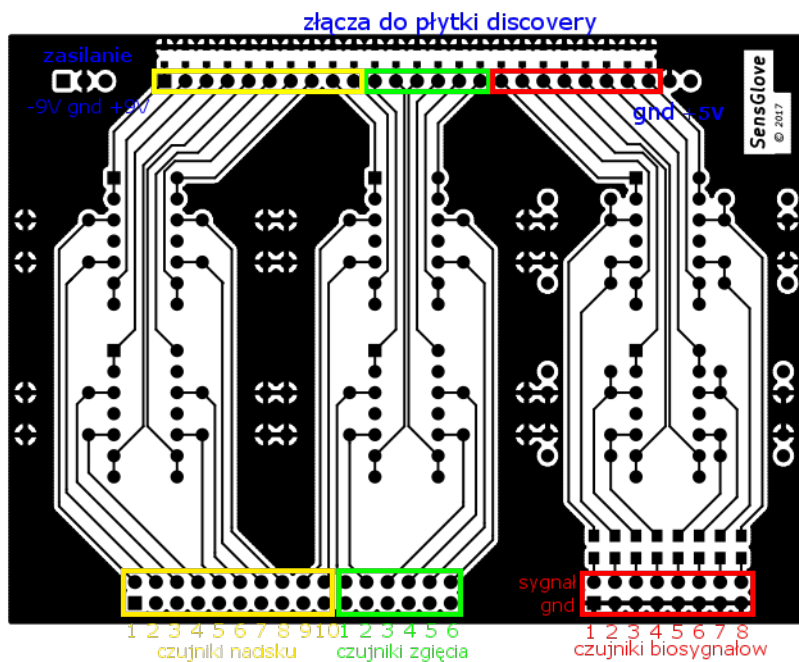
2.2.1 Schemat płytki pcb i wyprowadzenie pinów układu

2.2.2 Kryteria ewaluacji

Poprawność odczytu sygnałów przez układ wzmacniaczy sygnałów sprawdzono poprzez podłączenie potencjometrów w miejsce czujników i pomiar napięć na wyjściu. Sprawdzone również współpracę z rękawiczką.



Rysunek 2: Schemat płytki pcb



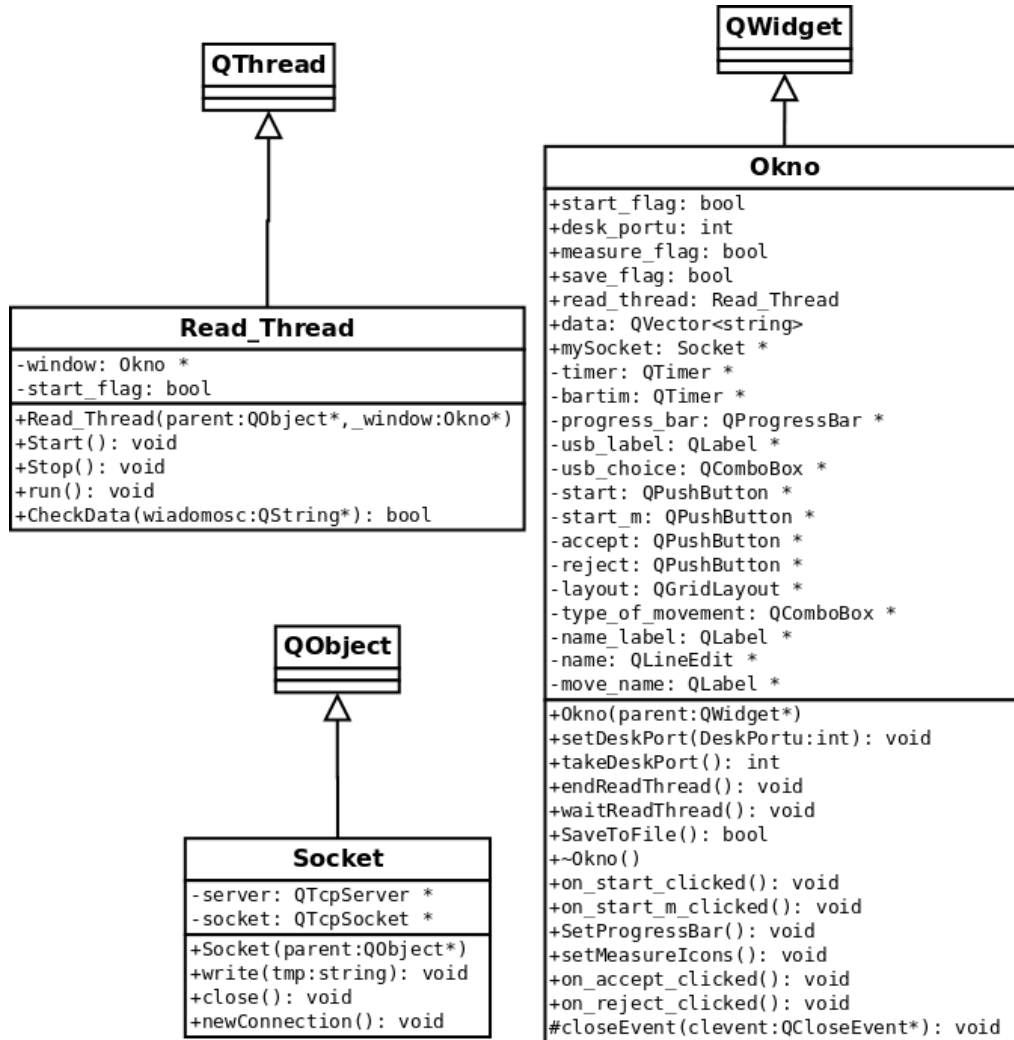
4. Podłączenie wzmacniaczy do płytki Discovery i powtórzenie testu z użyciem potencjometrów.

2.3 Program do akwizycji danych

Osoby przedzielone do zadania: Ada Weiss, Małgorzata Witka-Jeżewska

2.4 Opis klas programu

Dokładny opis klas i poszczególnych parametrów tworzony jest za pomocą programu Doxygen.



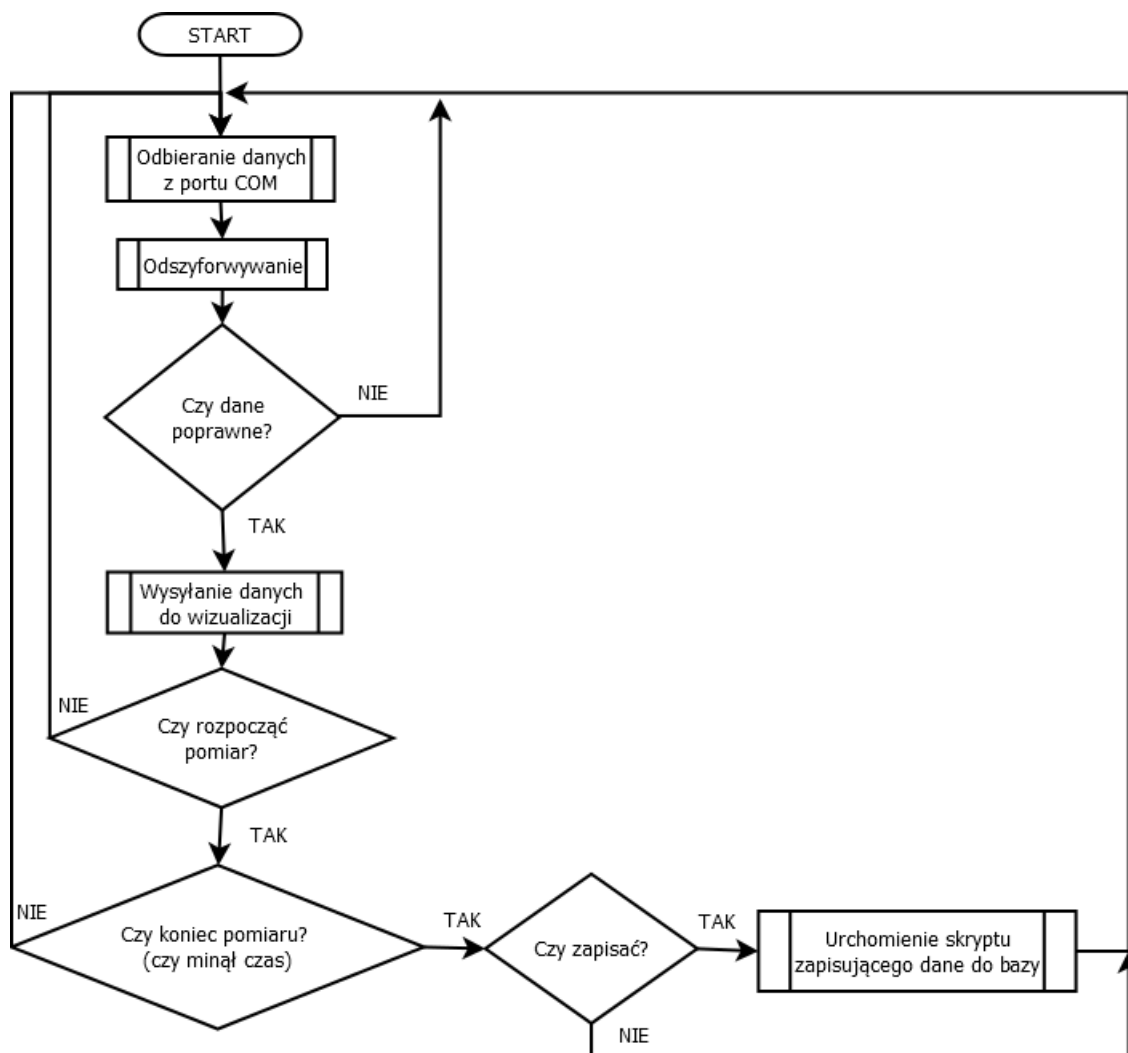
Rysunek 4: Diagram klas programu

2.4.1 Opis działania programu

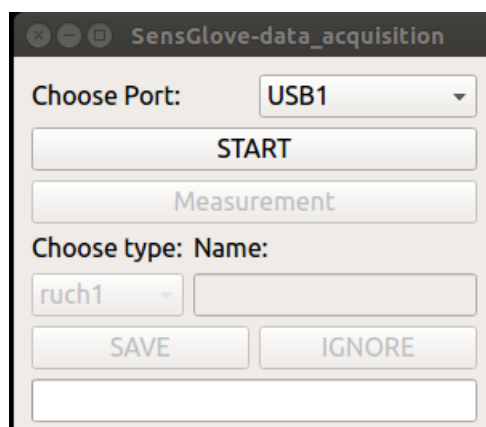
Interfejs graficzny programu został napisany z wykorzystaniem klas biblioteki Qt5 i został zaprezentowany na rys. 6.

Zasada działania została przedstawiona na rys. 5. Pierwszym krokiem jest wybór portu USB i wciśnięcie przycisku start, co uruchamia konfigurację wybranego portu USB. W przypadku braku podłączonego urządzenia na danym porcie wyświetlany jest komunikat przedstawiony na rys. 7. Kliknięcie "Yes" umożliwia ponowny wybór, a "No" kończy pracę programu.

Dane są odczytywane z poru USB w oddzielnym wątku Read_thread, dziedziczącego po QThread. Zapewnia

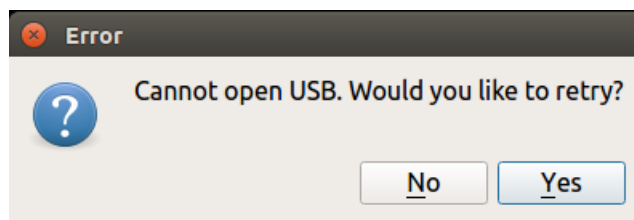


Rysunek 5: Schemat ideowy działania programu



Rysunek 6: Okno główne programu

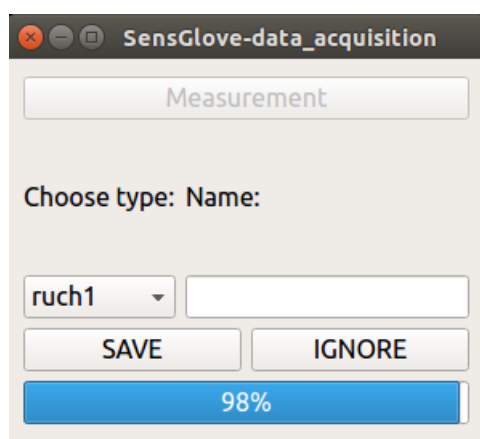
to odpowiednio szybką pracę programu. W tym samym wątku sprawdzana jest poprawność otrzymanej wiadomości (obliczanie sumy kontrolnej, sprawdzanie długości ciągu znaków) oraz wysyłanie danych do Socketu,



Rysunek 7: Komunikat o błędnym otwarciu USB.

który również został zaprogramowany z wykorzystaniem bibliotek Qt5 (klasa Socket).

W przypadku aktywowania przycisku "Measurement" jest timer (na 2s) i dane te są zapisywane w wektorze 2000x23, z którego to jest możliwość zapisu danych do pliku w przypadku akceptacji pomiaru po jego ukończeniu. Widok okna w trakcie pomiaru przedstawia 8.



Rysunek 8: Okno główne programu w trakcie pomiaru

Akceptacja pomiaru wymaga również podania imienia użytkownika oraz wyboru ruchu który jest wykonywany. Program wywołuje po zapisie do pliku skrypt który umieszcza plik w bazie danych.

Można również dany pomiar zignorować, np. w przypadku stwierdzenia, że został on błędnie wykonany.

2.4.2 Format danych

Ramka danych:

- znak X,
- 23 sygnały z czujników w postaci 4 znaków w systemie szesnastkowym,
- suma kontrolna w postaci dwóch znaków w systemie szesnastkowym,
- znak końca linii \r \n;

2.4.3 Raport z ewaluacji

Do symulowania danych wejściowych zaprogramowano płytkę STM32F476G Discovery tak, aby nadawała przykładowe dane w ustalonym przez nas formacie poprzez odpowiednią konfigurację USB.

Do sprawdzenia poprawności przesyłu danych wykorzystany został program telnet. Dane nadawane są na porcie 2666 (planowane jest dodanie możliwości wyboru portu, na którym dane będą nadawane). Przesył danych był poprawny.

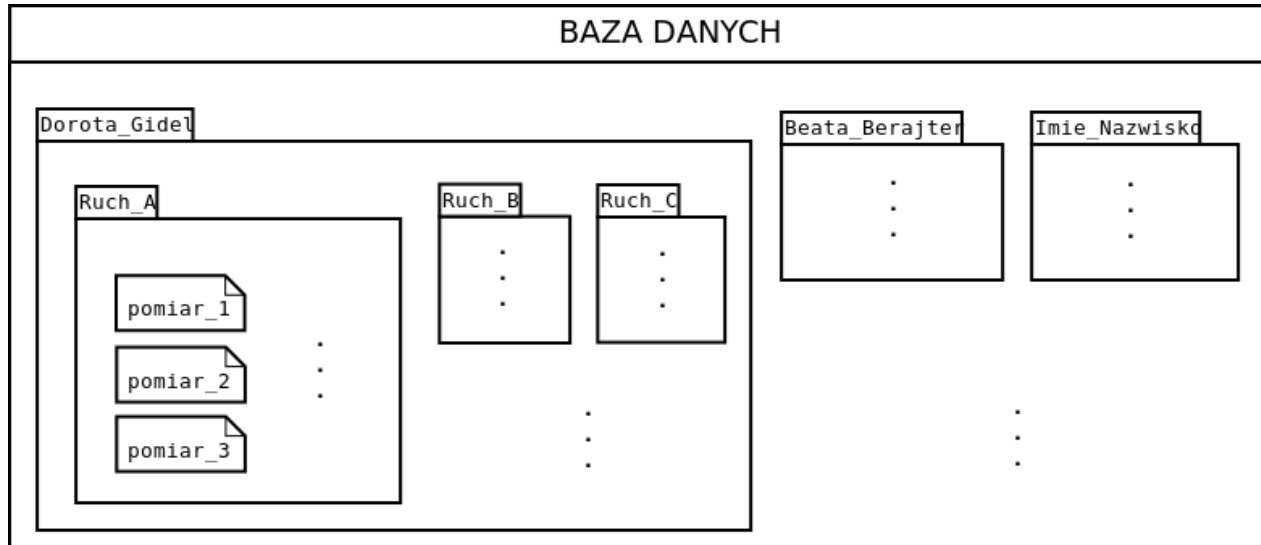
Dane były również wysyłane do programu do wizualizacji rękawiczki poprzez wspólny router Wi-Fi do drugiego komputera. Przesyłanie odbywało się bez zarzutów.

2.5 Baza danych

Osoby przydzielone do zadania: Beata Berajter, Dorota Gidel.

2.5.1 Wprowadzone zmiany

Zdecydowano się na zmianę struktury bazy danych. Zamiast umieszczania trzech osobnych plików zawierających dane odpowiednio z czujników nacisku, zgięcia i elektrod, tworzony będzie tylko jeden plik zawierający wszystkie te dane w takiej kolejności. Plik taki zawierać będzie 2000 linii i 23 kolumny odczytów. Dodatkowo, nazwa pliku nie będzie zawierać daty gdyż można ją uzyskać z daty tworzenia pliku. Nowy schemat struktury bazy danych przedstawiono na rysunku 9.



Rysunek 9: Schemat struktury bazy danych

2.5.2 Opis działania skryptu

Stworzony skrypt do umieszczania w odpowiednim miejscu plików z odczytami napisany jest w powłoce bash. Przykładowe wywołanie skryptu wygląda następująco:

```
$ bash przeniesPlik sciezka/do/pliku/z/danymi Imie NazwaRuchu
```

Skrypt przenosi podany plik (*sciezka/do/pliku/z/danymi*) do głównego folderu bazy danych który zdefiniowany jest jako katalog o nazwie "baza_danych" w katalogu domowym. Dzięki podanym argumentom umieszcza plik w odpowiednim miejscu. Numeracja pliku po kolei jest zapewniona przez prowadzenie spisu w osobnym pliku tekstowym należącym do bazy danych. Dla bezpieczeństwa sprawdzane jest też czy w wywołaniu skryptu została podana wystarczająca liczba argumentów.

2.5.3 Kryteria ewaluacji

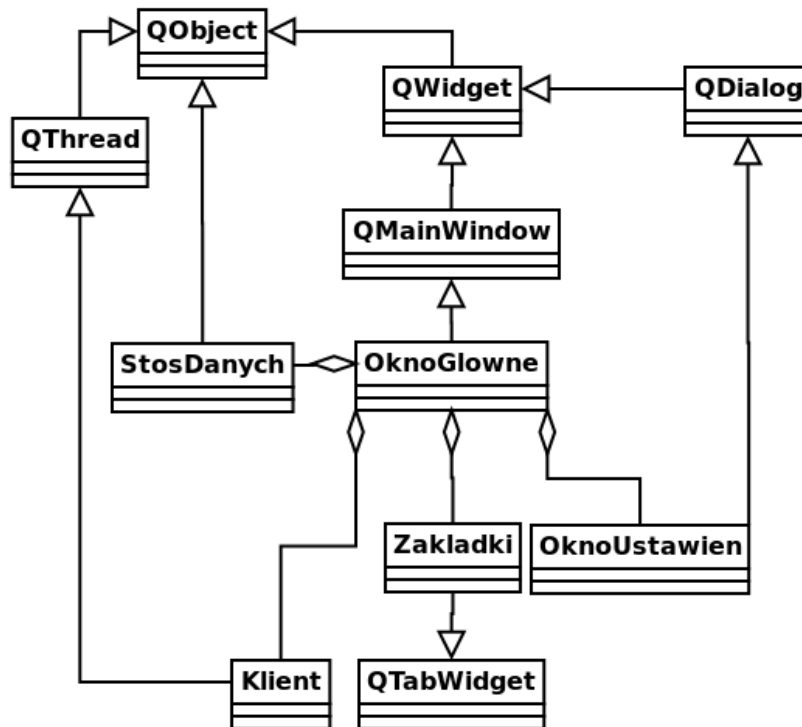
Do sprawdzania poprawności działania bazy danych wywołano skrypt z przykładowymi argumentami. Otrzymano oczekiwane efekty, tj. utworzenie w odpowiednim miejscu w strukturze katalogów pliku o pożądanej nazwie (indeksie zwiększającym swoją wartość przy kolejnych przenoszeniach pliku do tego samego katalogu).

2.6 Wizualizacja danych

Osoby przydzielone do zadania: Dorota Gidel, Katarzyna Wądryk.

2.6.1 Ogólny diagram klas

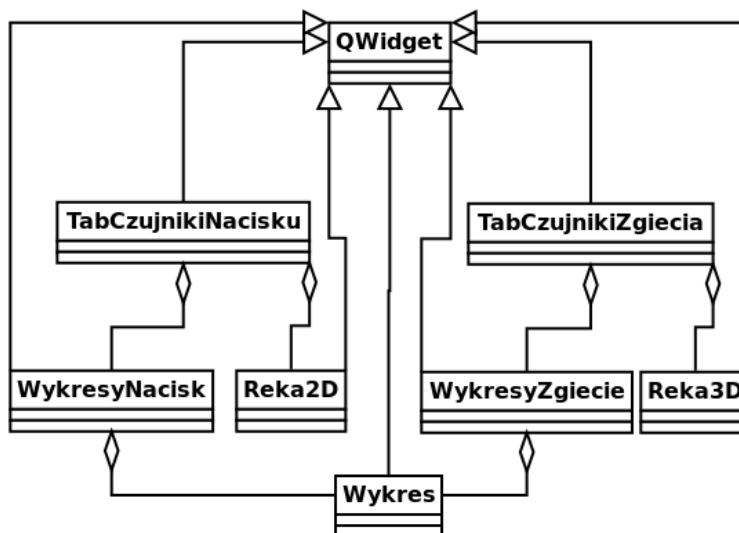
Na rysunku 10 zaprezentowano ogólny diagram klas w programie wizualizującym dane z czujników.



Rysunek 10: Ogólny diagram klas - wizualizacja

2.6.2 Diagram klas zakładek

Na obrazku 11 zaprezentowano ogólny diagram klas zakładek zawierających wykresy i prezentacje odpowiednio z czujników nacisku oraz zgięcia.



Rysunek 11: Diagram klas zakładek - wizualizacja

2.6.3 Interfejs użytkownika

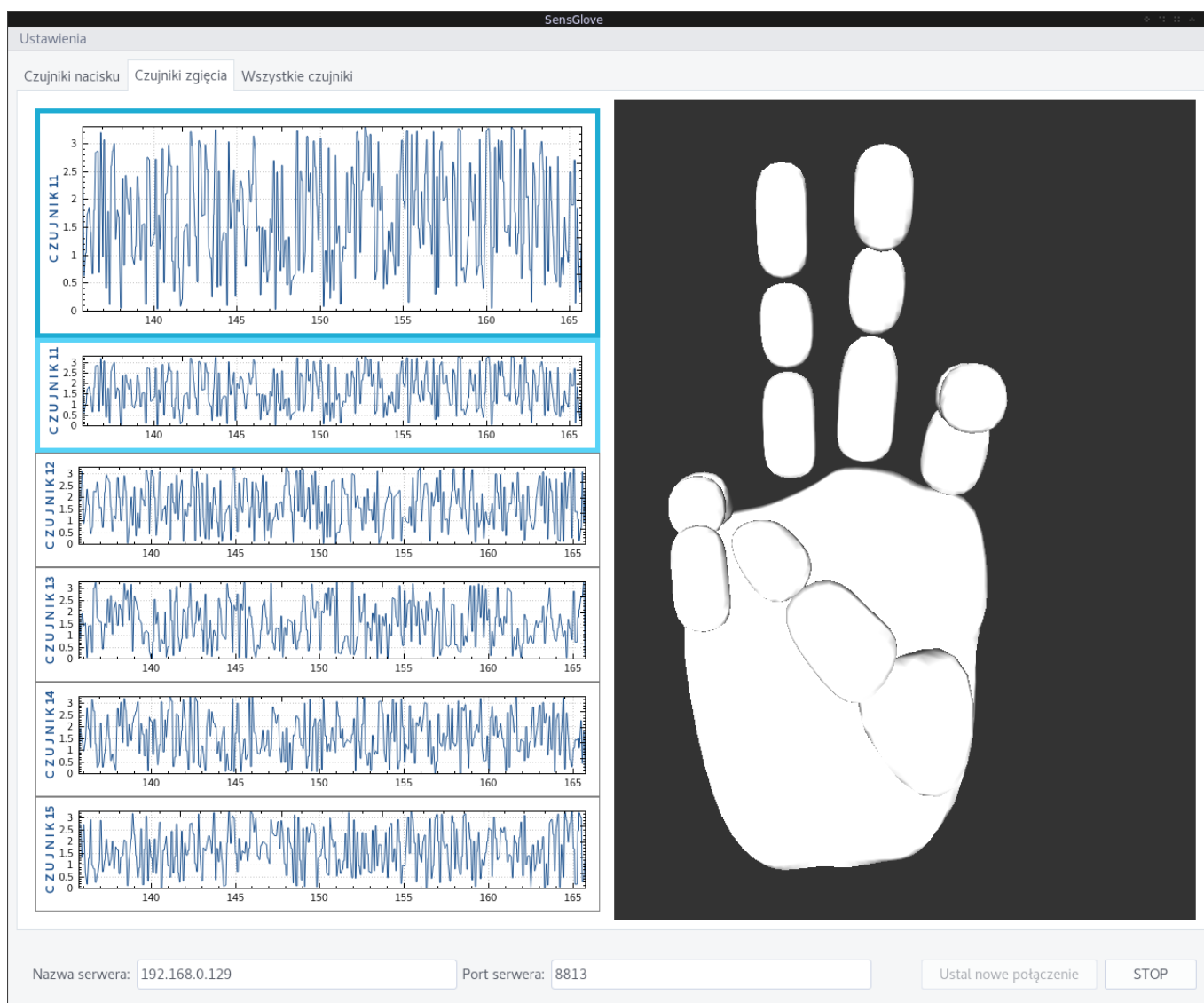
Na obecnym etapie użytkownik ma możliwość połączenia się z portem nadającym dane przez przyciśnięcie przycisku "Ustal nowe połączenie". W trakcie odbierania danych istnieje możliwość zatrzymania wykresów i wizualizacji dłoni w danym momencie poprzez przyciśnięcie przycisku "Stop". W lewym górnym rogu użytkownik ma dostęp do ustawień w których obecnie może zmienić zakres czasu prezentowanego na wykresach. Domyślnie wartość ta wynosi 30 s.

Rysunek 12 przedstawia pierwszą zakładkę aplikacji wizualizującej dane - czujniki nacisku. Po lewej stronie umieszczone są wykresy. Pierwszy największy wykres, można zmieniać przyciskając odpowiednio wykres który chcemy najbardziej śledzić. Wybrany wykres zaznaczony jest przez kolorową obwódkę. Po prawej stronie znajduje się widget zawierający rękę 2D prezentującą przez kolory odczytywane w danym momencie wartości napięcia.



Rysunek 12: Diagram klas zakładek - wizualizacja

Rysunek 13 przedstawia drugą zakładkę aplikacji wizualizującej dane - czujniki zgięcia. Po lewej stronie tak samo jak poprzednio znajdują się wykresy odczytywanego napięcia. Po prawej stronie znajduje się widget zawierający rękę 3D której palce zginają się odpowiednio do otrzymywanych wartości z czujników.



Rysunek 13: Diagram klas zakładek - wizualizacja

2.6.4 Kryteria ewaluacji

Do testowania poprawności działania aplikacji napisano dodatkowy program mający udawać serwer nadający dane. Program ten napisany jest w C. Dane które generuje są randomowe, a port, częstotliwość i ilość wysyłanych wiadomości podaje się jako jego argumenty.

Aplikacja odbiera pomyślnie zasymulowane dane z dodatkowego programu oraz wizualizuje zinterpretowane pomiary na wykresach i modelach dłoni. Przeprowadzono też integrację tej aplikacji z aplikacją to akwizycji danych, która przebiegła pomyślnie.

2.6.5 Brakujące elementy

Aplikacja nie jest jeszcze w pełni gotowa i wymaga dopracowania. Wizualizowane dane nie są jeszcze zsynchronizowane z rzeczywistymi wartościami danych z czujników. Brakuje też dodatkowych ustawień pozwalających na zmianę interfejsu w sposób wygodny dla użytkownika.