

Politechnika Wrocławska  
AiR ARR  
Projekt zespołowy

# SENSGLOVE

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

Prowadzący: dr inż. Krzysztof Arent

20 czerwca 2017

# Spis treści

<b>1</b>	<b>Kryteria ewaluacji</b>	<b>2</b>
<b>2</b>	<b>Przebieg prac dla poszczególnych komponentów</b>	<b>2</b>
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 układu . . . . .	4
2.2.2	Lista elementów . . . . .	6
2.2.3	Połączenie z płytą STM32F3Discovery . . . . .	6
2.3	Kod programu . . . . .	6
2.3.1	Kryteria ewaluacji . . . . .	7
2.4	Program do akwizycji danych . . . . .	7
2.5	Opis klas programu . . . . .	7
2.5.1	Opis działania programu . . . . .	7
2.5.2	Format danych . . . . .	8
2.5.3	Raport z ewaluacji . . . . .	10
2.6	Baza danych . . . . .	10
2.6.1	Wprowadzone zmiany . . . . .	10
2.6.2	Opis działania skryptu . . . . .	11
2.6.3	Kryteria ewaluacji . . . . .	11
2.7	Wizualizacja danych . . . . .	11
2.7.1	Ogólny diagram klas . . . . .	11
2.7.2	Diagram klas zakładek . . . . .	12
2.7.3	Interfejs użytkownika . . . . .	13
2.7.4	Kryteria ewaluacji . . . . .	14
2.8	Ewaluacja końcowa projektu . . . . .	14

# 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.

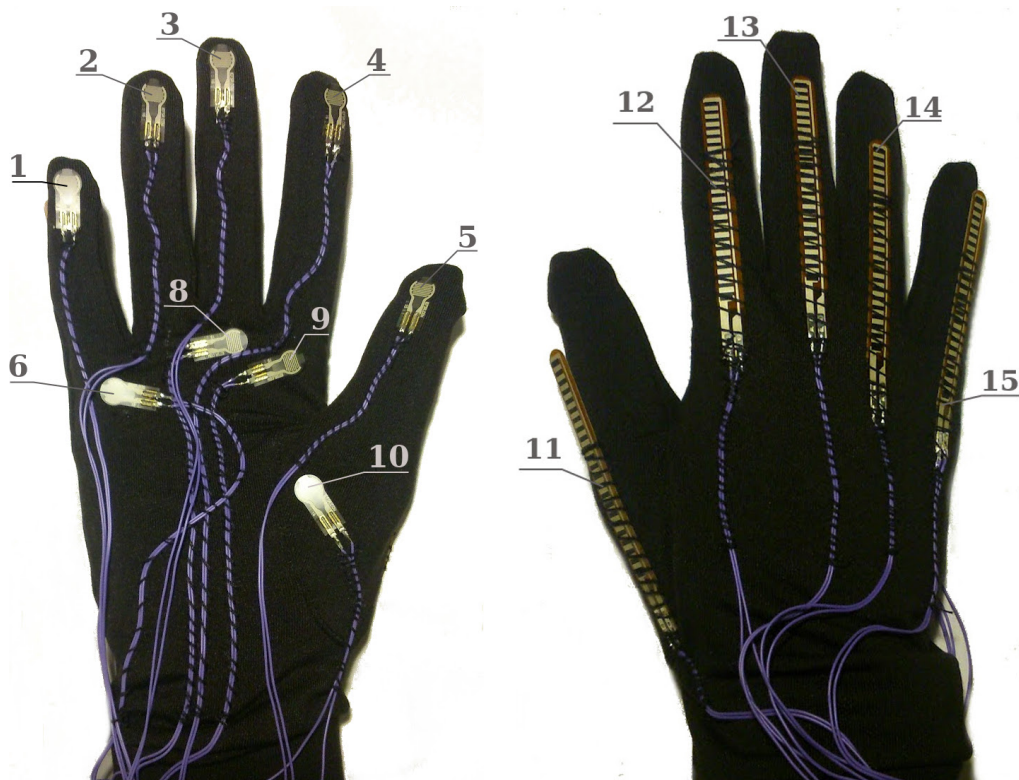
## 2 Przebieg prac dla poszczególnych komponentów

### 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 rękawiczce 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 rękawiczce o rozmiarze S.

Ze względu uszkodzenia się czujników, rękawiczka posiada 9 czujników nacisku. Jak widać na zdjęciu 1 czujnik 7 został pominięty, a czujnik 8 lekko przesunięty w lewą stronę by to zrekompensować. Jednakże dalej istnieje możliwość przymocowania brakującego czujnika 7.

### **2.1.3 Kryteria ewaluacji**

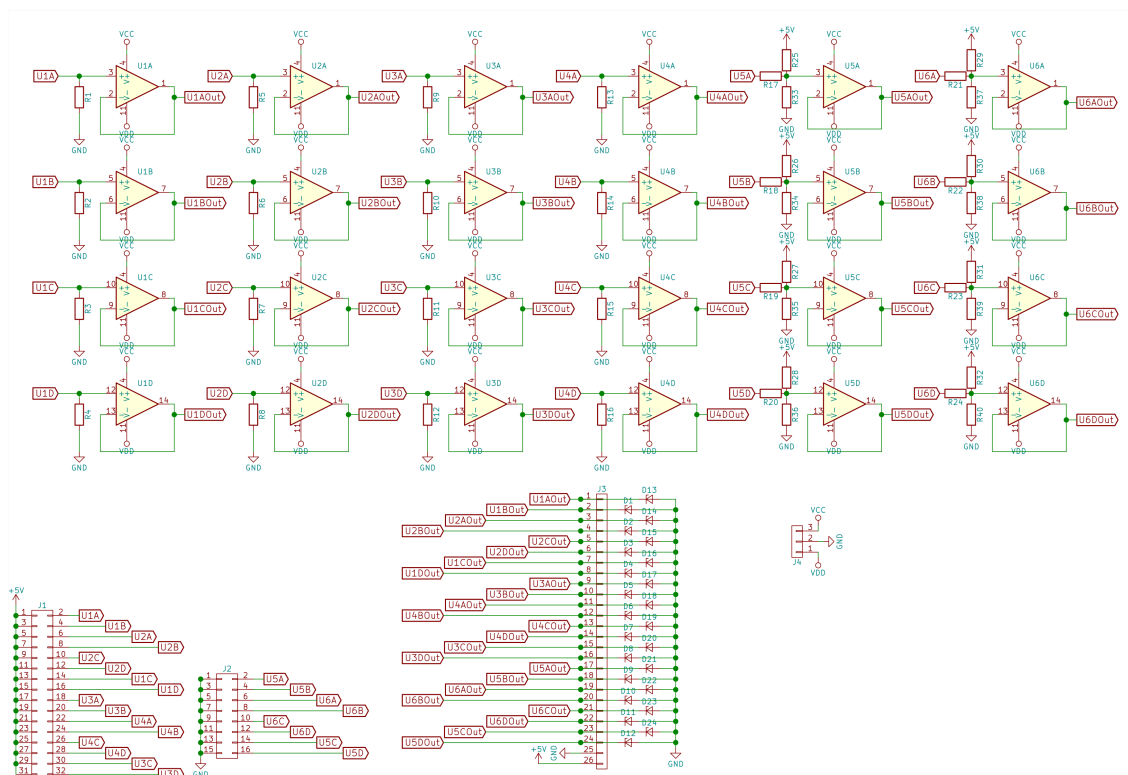
Jeden czujnik z wewnętrznej części dłoni został uszkodzony. Kilka czujników nacisku zostało odczepionych, ponieważ zostały przyklejone za nisko na placu, co powodowało złe odczyty przy chwytaniu przedmiotów. Rękawiczka spełnia następujące kryteria:

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 mierzono napięcie, które jak oczekiwano zmieniało odpowiednio swoją wartość. Sprawdzone zostały zarówno czujniki zgięcia jak i nacisku.

## **2.2 Interfejs sprzętowy**

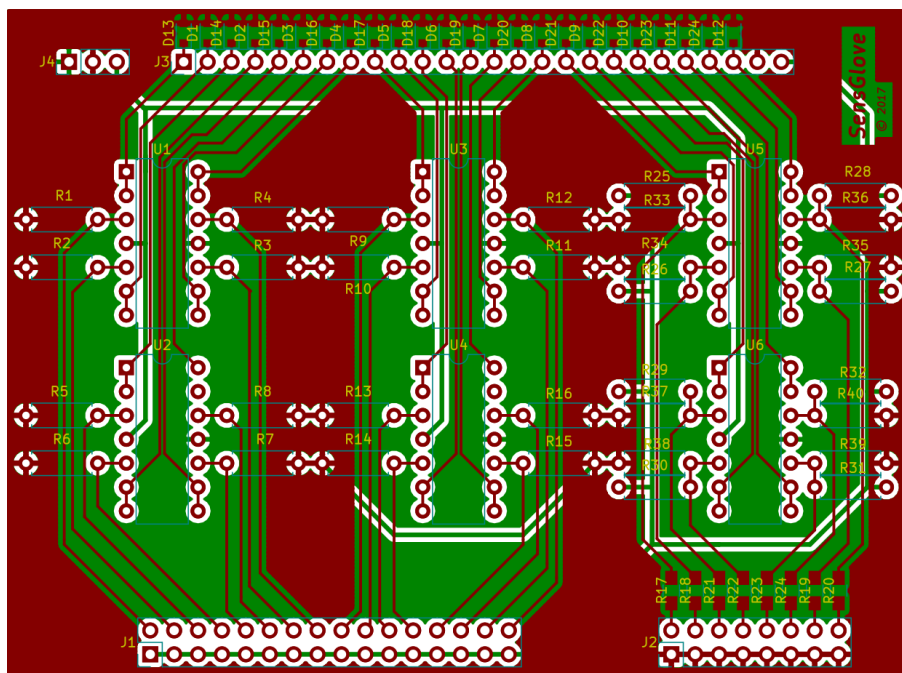
Osoby przydzielone do zadania: Beata Berajter i Dawid Brząkała.



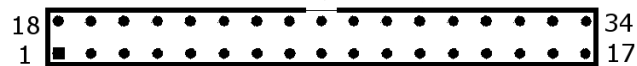
Rysunek 2: schemat układu

### 2.2.1 Schemat układu

Projekt płytki pcb i wyprowadzenie pinów układu przedstawiono na rysunku 3.

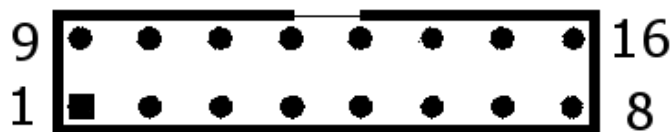


Rysunek 3: Schemat płytki pcb



Rysunek 4: J1 - pinout

1 - 16	zasilanie czujników - +5v
17 - 27	złącza czujników nacisku 1 - 10
28 - 33	złącza czujników ugięcia 1 - 6
2, 34	NC



Rysunek 5: J2 - pinout

1 - 7	GND
8 - 16	złącza czujników biosygnatów 1 - 8



Rysunek 6: J3 - pinout

1 - 10	sygnał czujników nacisku 1 - 10
11 - 16	sygnał czujników ugięcia 1 - 6
17 - 24	sygnał czujników biosygnatów 1 - 8
25	+5V
26	GND



Rysunek 7: J4 - pinout

1	-9V
2	GND
3	+9V

### 2.2.2 Lista elementów

U1-U6	TLC084CN	THT
R1-R10	10kOhm	THT
R11-R16	33kOhm	THT
R17-R24	3.3kOhm	SMD
R25-R32	3.3kOhm	THT
R33-R40	2.2kOhm	THT
D1-D24	3.3V Zenera	SMD
J1	IDC-34 żeńskie	THT
J2	IDC-16 żeńskie	THT
J3	listwa kołkowa 1x26 2.56mm	THT
J4	listwa kołkowa 1x3 2.56mm	THT

### 2.2.3 Połączenie z płytą STM32F3Discovery

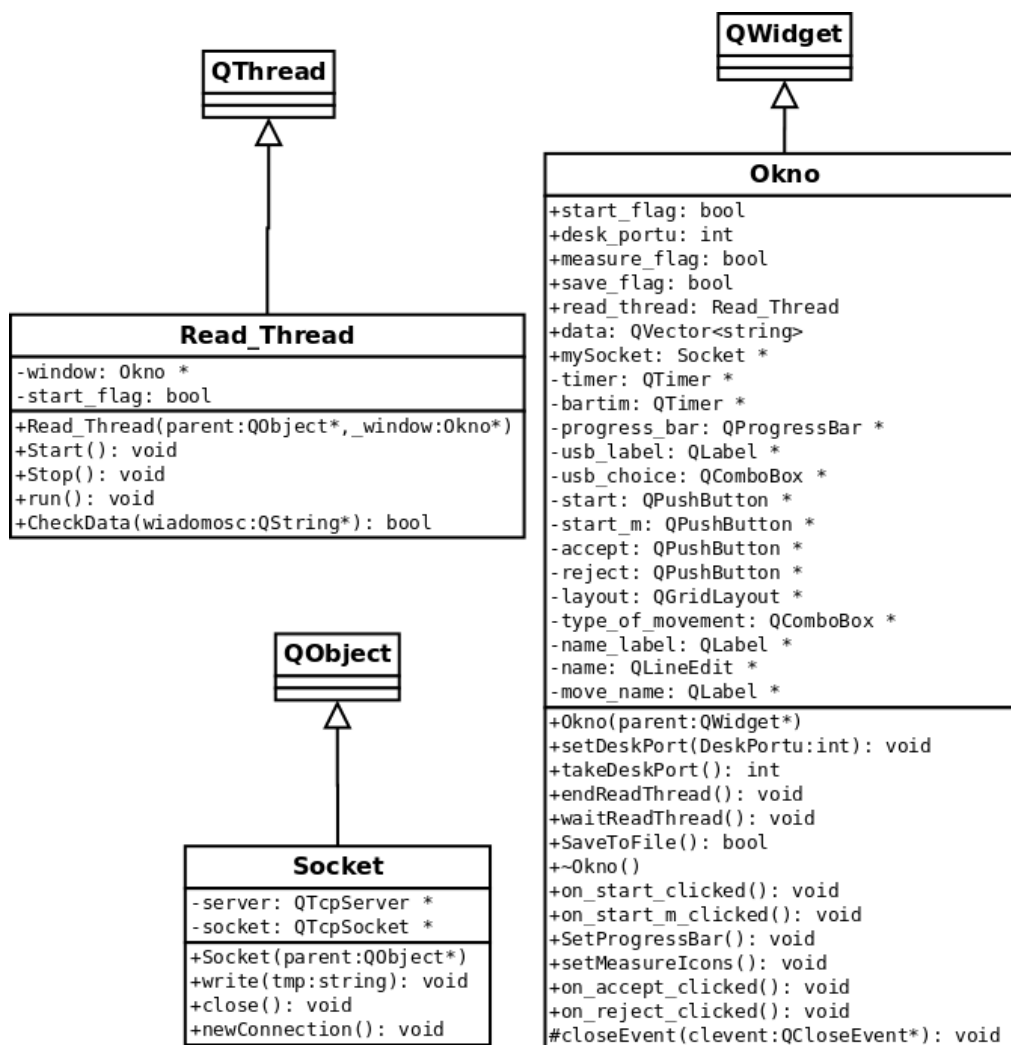
J1	ST3M32Discovery
1	PB12
2	PB14
3	PB15
4	PD8
5	PD9
6	PD10
7	PD11
8	PD12
9	PD13
10	PD14
11	PB0
12	PB1
13	PE7
14	PE8
15	PE9
16	PE10
17	PA0
18	PA3
19	PA2
20	PF4
21	PA4
22	PA5
23	PA6
24	PA7
25	GND
26	5V

## 2.3 Kod programu

```
//struktura danych do zapisu odczytu sygnałów
typedef union
{
    struct
    {
        uint16_t forceSensors[10];
        uint16_t flexSensors[6];
        uint16_t biosignals[8];
    }
};
```







Rysunek 8: Diagram klas programu

Zasada działania została przedstawiona na rys. 9. 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. 11. 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 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, który również został zaprogramowany z wykorzystaniem bibliotek Qt5 (klasa Socket).

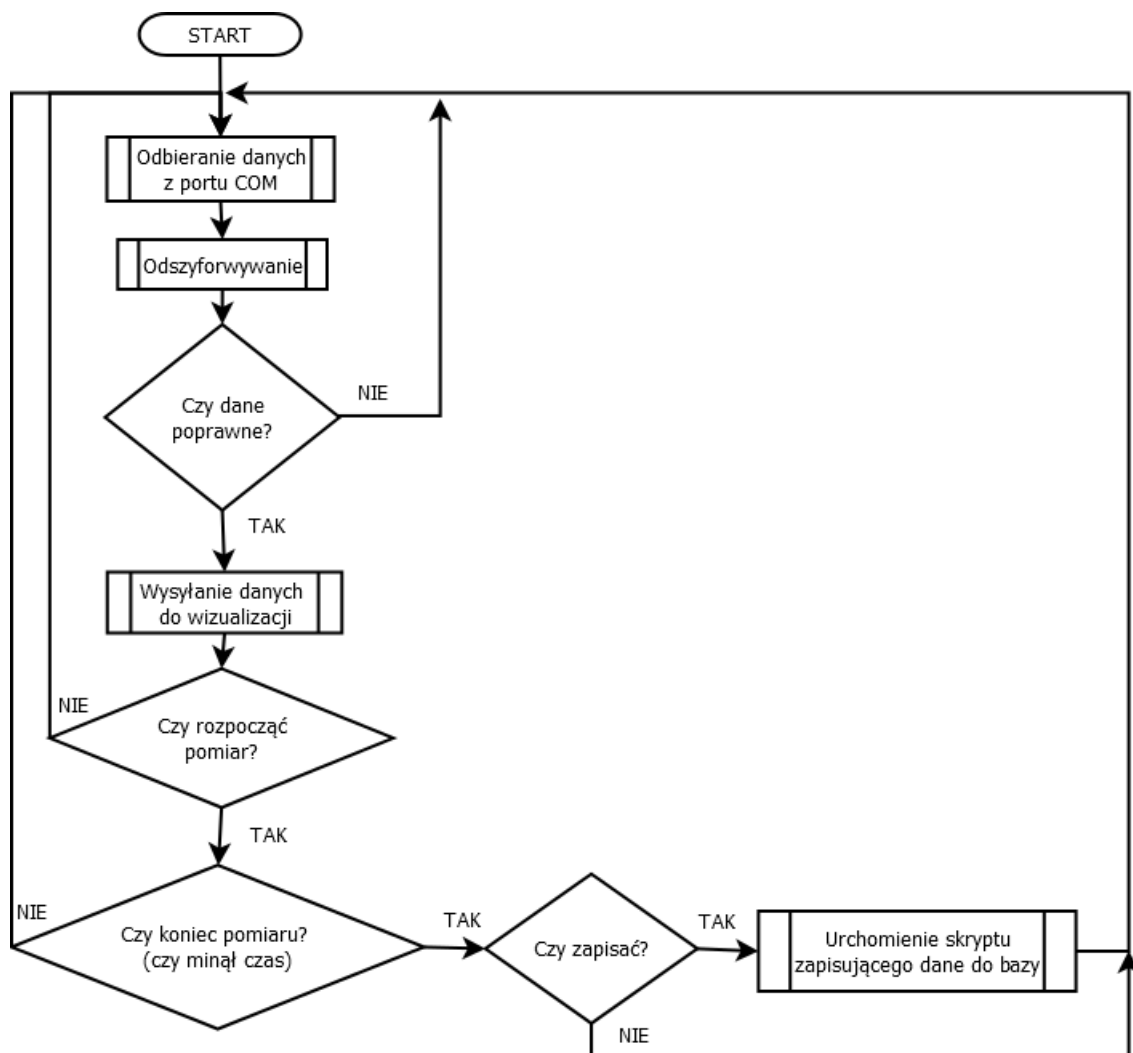
W przypadku aktywowania przycisku "Measurment" aktywowany jest timer (na 2s) i dane te są zapisywane w wektorze 2000x23, z którego to jest możliwość zapisu danych do pliku w przypadku akceptacji pomiaru po jego ukończeniu. Widok okna w trakcie pomiaru przedstawia 12.

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.

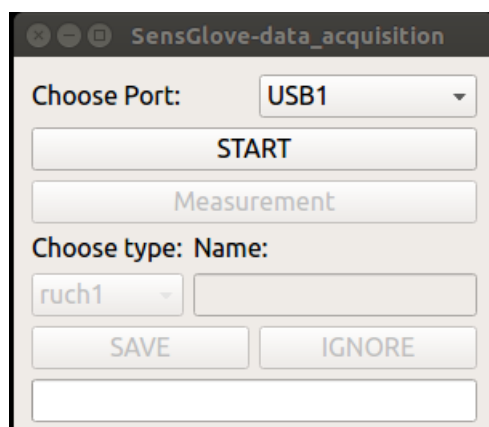
Można również dany pomiar zignorować, np. w przypadku stwierdzenia, że został on błędnie wykonany.

## 2.5.2 Format danych

Ramka danych:

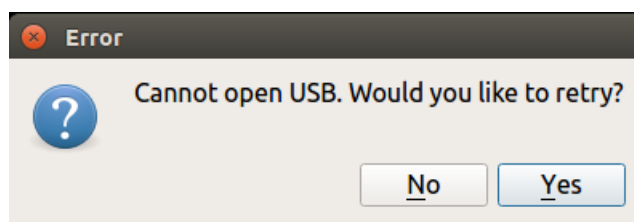


Rysunek 9: Schemat ideowy działania programu

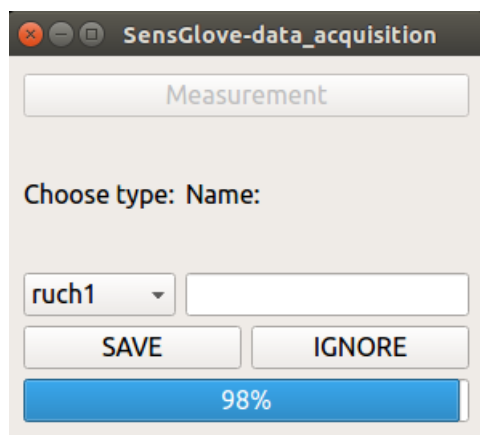


Rysunek 10: Okno główne programu

- znak X,



Rysunek 11: Komunikat o błędnym otwarciu USB.



Rysunek 12: Okno główne programu w trakcie pomiaru

- 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;

### 2.5.3 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.

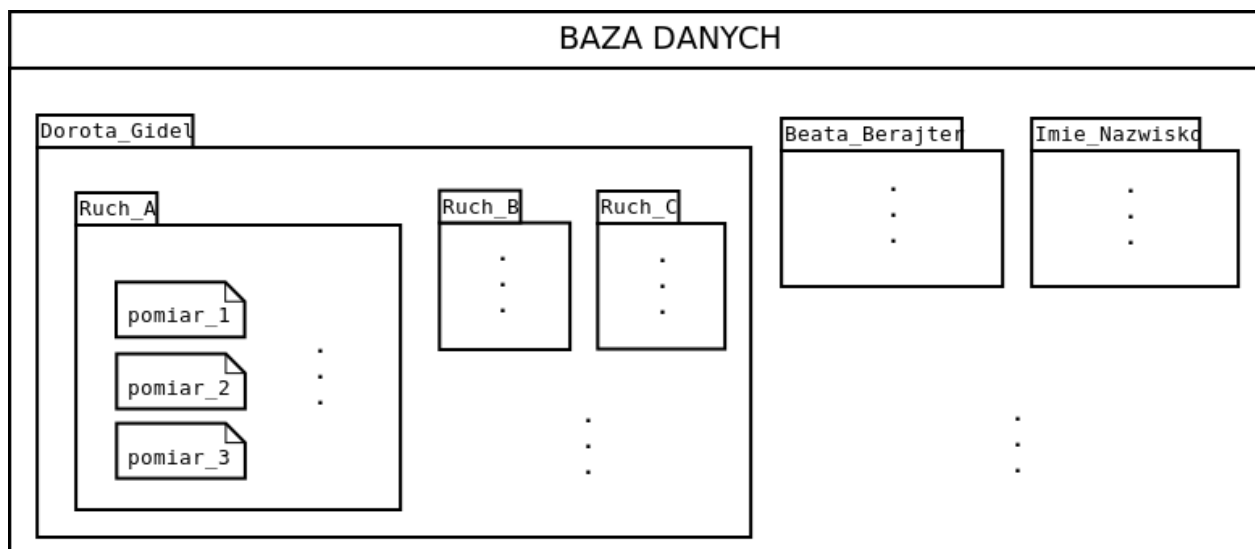
Dane były również wysyłane do programu do wizualizacji rękawiczki poprzez wspólny router Wi-Fi do drugiego komputera. Przesyłanie odbywało się bez zarzutów.

## 2.6 Baza danych

Osoby przydzielone do zadania: Beata Berajter, Dorota Gidel.

### 2.6.1 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 13.



Rysunek 13: Schemat struktury bazy danych

### 2.6.2 Opis działania skryptu

Stworzony skrypt do umieszczania w odpowiednim miejscu plików z odczytami napisany jest w powłocie 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.

### 2.6.3 Kryteria ewaluacji

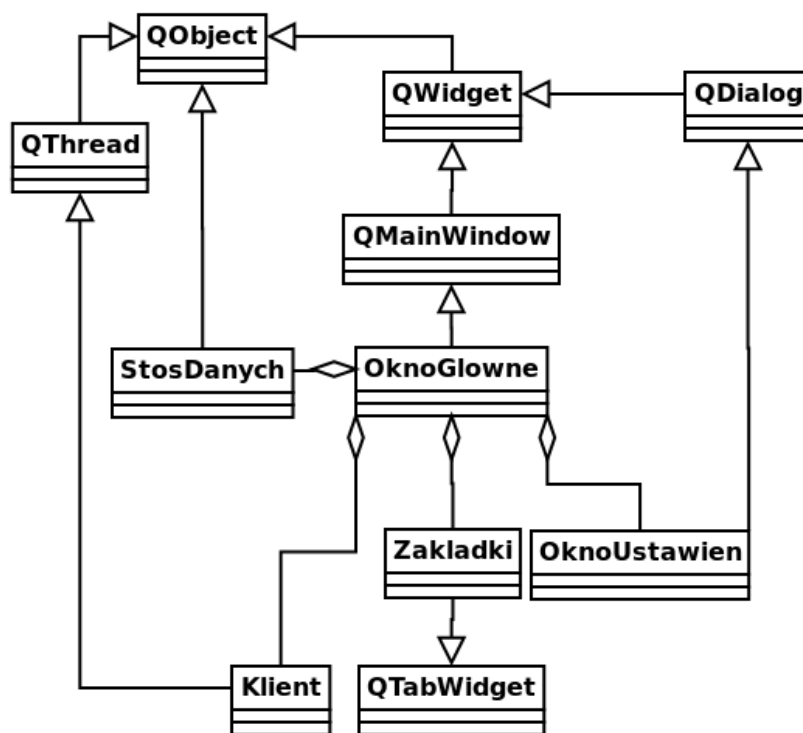
Do sprawdzania poprawności działania bazy danych wywołano skrypt z przykładowymi argumentami. Otrzymano oczekiwane efekty, tj. utworzenie w odpowiednim miejscu w strukturze katalogów pliku o pożądanej nazwie (indeksie zwiększającym swoją wartość przy kolejnych przenoszeniach pliku do tego samego katalogu).

## 2.7 Wizualizacja danych

Osoby przydzielone do zadania: Dorota Gidel, Katarzyna Wądrzyk.

### 2.7.1 Ogólny diagram klas

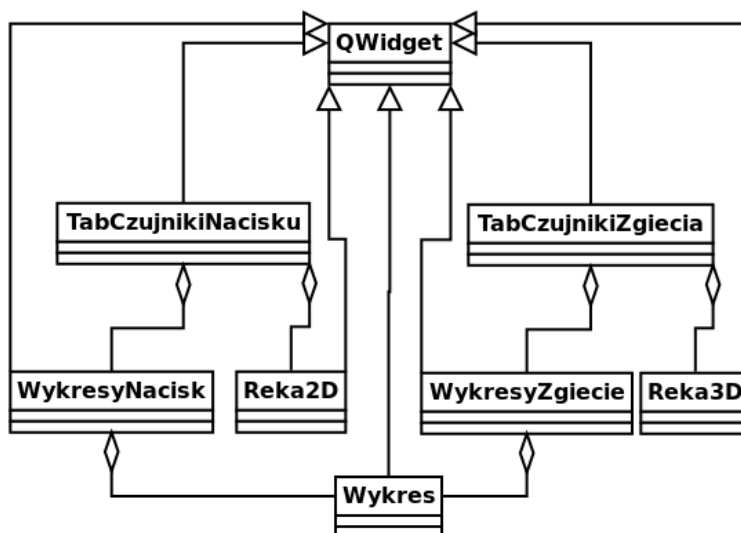
Na rysunku 14 zaprezentowano ogólny diagram klas w programie wizualizującym dane z czujników.



Rysunek 14: Ogólny diagram klas - wizualizacja

### 2.7.2 Diagram klas zakładki

Na obrazku 15 zaprezentowano ogólny diagram klas zakładki zawierających wykresy i prezentacje odpowiednio z czujników nacisku oraz zgięcia.



Rysunek 15: Diagram klas zakładki - wizualizacja

### 2.7.3 Interfejs użytkownika

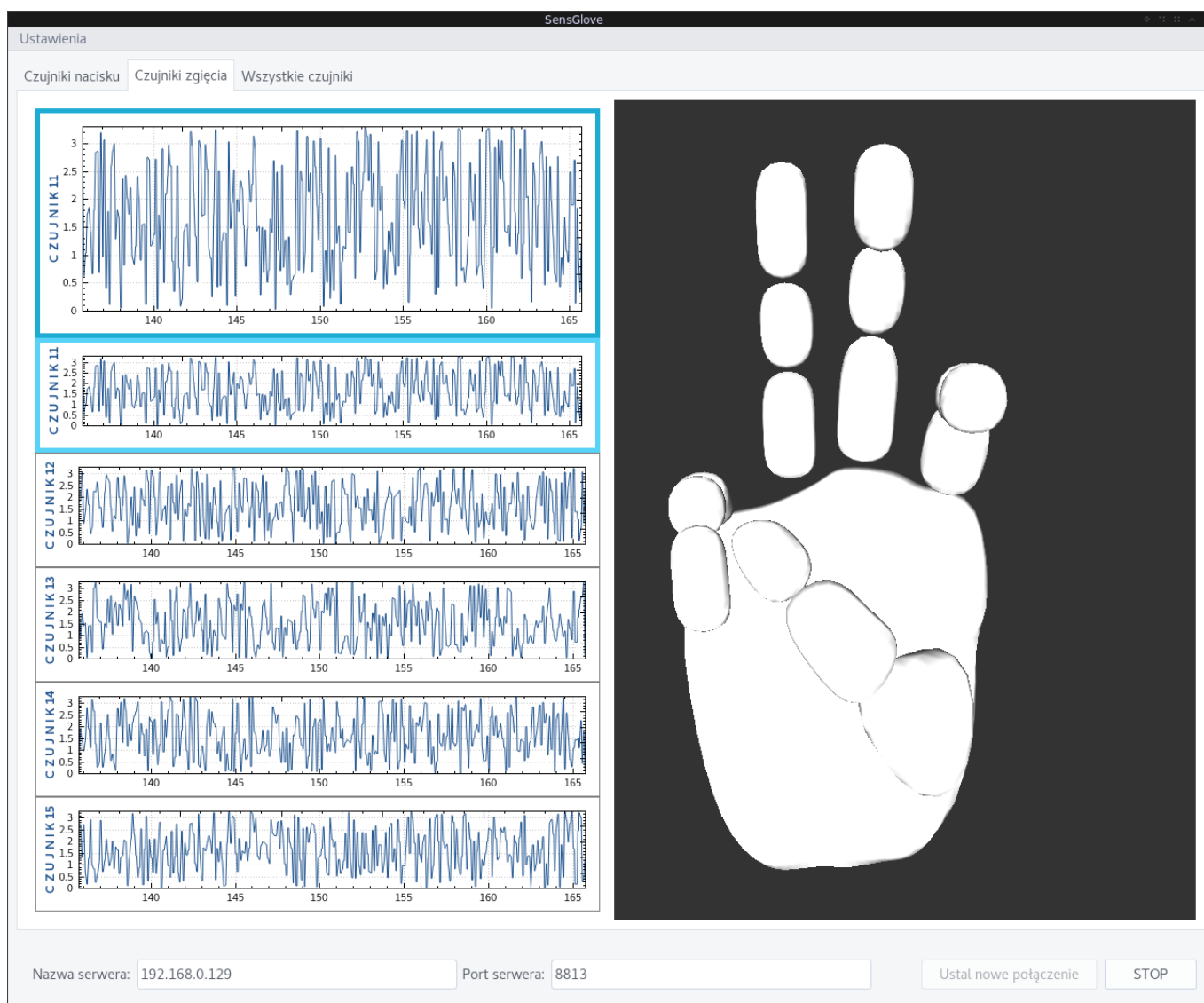
Użytkownik ma możliwość połączenia się z portem nadającym dane przez przyciśnięcie przycisku "Ustal nowe połączenie". W trakcie odbierania danych istnieje możliwość zatrzymania wykresów i wizualizacji dłoni w danym momencie poprzez przyciśnięcie przycisku "Stop". W lewym górnym rogu użytkownik ma dostęp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

Rysunek 16 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 16: Diagram klas zakładek - wizualizacja

Rysunek 17 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 17: Diagram klas zakładek - wizualizacja

#### 2.7.4 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.

### 2.8 Ewaluacja końcowa projektu

Po połączeniu wszystkich komponentów projektu przeprowadzono testy. Polegały one na wykonywaniu ruchów w rękawiczce sensorycznej i obserwowanie efektów w programie wizualizującym oraz sprawdzenie poprawnego zapisu pliku do bazy danych. Testy pokazały, że sygnały przesyłane są odbierane poprawnie, co pokazują zdjęcia i film umieszczony na stronie internetowej [sensglove.happyrobotics.com](http://sensglove.happyrobotics.com) w odpowiednich zakładkach.