

PROJEKT  
WIZUALIZACJA DANYCH SENSORYCZNYCH

---

**RoboVision**

---

**Marcin Bober, 249426**

---

---

---



*Prowadzący:*  
dr inż. Bogdan Kreczmer

Katedra Cybernetyki i Robotyki  
Wydziału Elektroniki  
Politechniki Wrocławskiej

10 maja 2021

# Spis treści

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Charakterystyka tematu projektu</b>   | <b>1</b> |
| <b>2</b> | <b>Podcele i etapy realizacji projektu</b>   | <b>1</b> |
| <b>3</b> | <b>Specyfikacja finalnego produktu</b>   | <b>1</b> |
| <b>4</b> | <b>Terminarz realizacji poszczególnych podcelów (z dokładnością do 1 tygodnia)</b> | <b>2</b> |
| <b>5</b> | <b>Projekt interfejsu graficznego</b>  | <b>3</b> |
| <b>6</b> | <b>Komunikacja z robotem</b>   | <b>4</b> |
| 6.1      | Ping . . . . .   | 4        |
| 6.2      | Dystans przeszkody . . . . .   | 4        |
| 6.3      | Bateria . . . . .  | 5        |
| 6.4      | Prędkość . . . . .   | 5        |
| 6.5      | Moc silników . . . . .   | 5        |
| 6.6      | Akcelometr . . . . .   | 5        |
| 6.7      | Żyroskop . . . . .   | 5        |
| <b>7</b> | <b>Wykres akcelometru</b>  | <b>5</b> |
| <b>8</b> | <b>Aktualne rezultaty</b>  | <b>6</b> |

# 1 Charakterystyka tematu projektu

Projekt ma na celu stworzenie aplikacji okienkowej, która poprzez połączenie internetowe będzie w stanie wydawać polecenia do robota mobilnego, sterować nim, a także pobierać informacje z czujników i wizualizować je.

## 2 Podcele i etapy realizacji projektu

Projekt powdzielony będzie na kilka pomniejszych celów tak, aby każdy z nich mógłbyć osobno rozwijany.

Lista podelów:

- Zapoznanie się z dostępną literaturą związaną z tematem oraz zdobycie informacji niezbędnych do zrealizowania zadania.
- Przygotowanie graficznego szkicu aplikacji wraz z rozplanowaniem funkcjonalności.
- Zdefiniowanie protokołu komunikacyjnego, struktury ramek przesyłanych danych i implementacja interfejsu sieciowego.
- Parsowanie danych odbieranych z robota.
- Przygotowanie wizualizacji zebranych danych.
- Obsługa klawiatury i joysticka.
- Implementacja algorytmu sterowania i przesyłanie wyników do urządzenia.

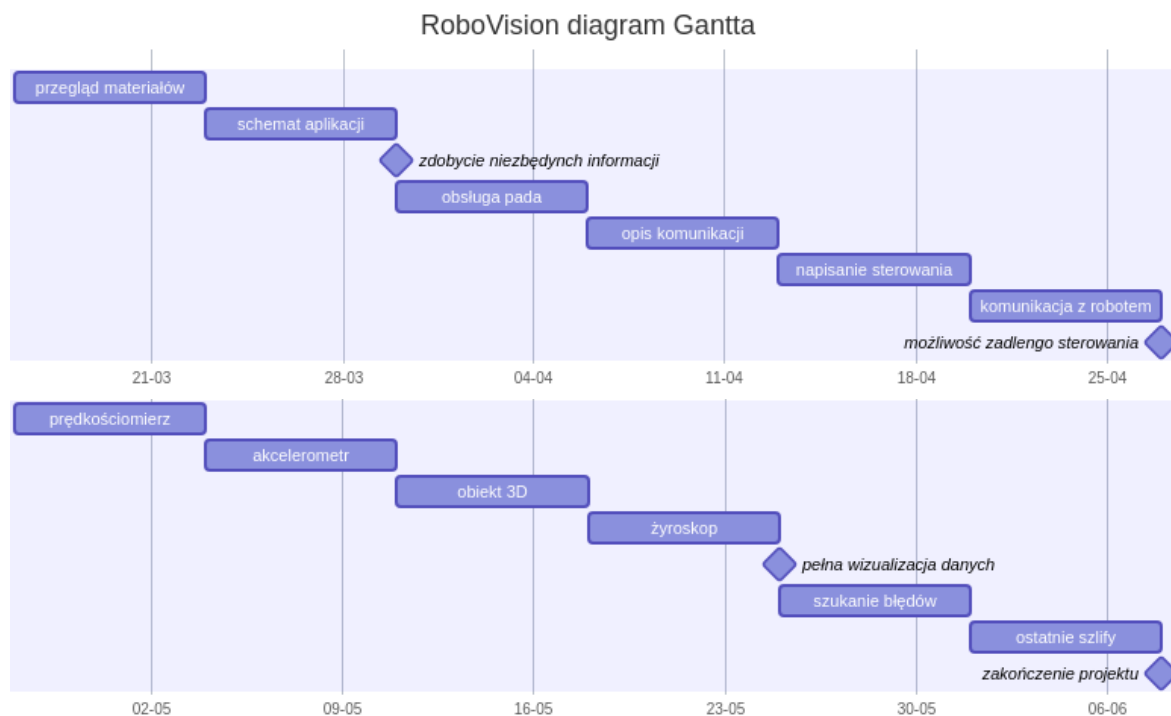
## 3 Specyfikacja finalnego produktu

Aplikacja będzie w stanie wizualizować dane odbierane z czujników robota. Będą to między innymi:

- wskazania akcelometru,
- wskazania żyroskopu,
- aproksymacja poziomu baterii,
- odlegość przeszkody zczytanej z przedniego czujnika ultradźwiękowego,
- prędkość rzeczywista pojazdu z enkoderów.

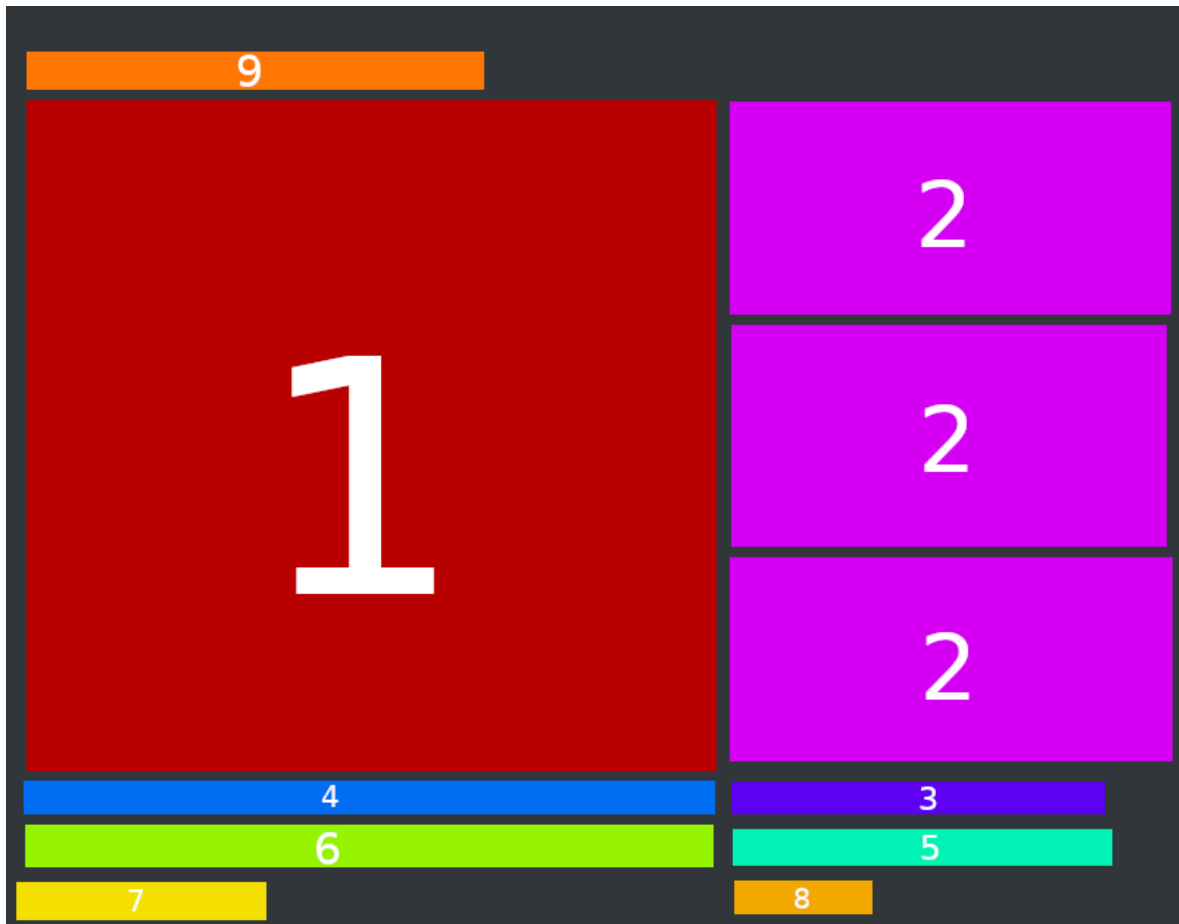
## 4 Terminarz realizacji poszczególnych podcelów (z dokładnością do 1 tygodnia)

- 22 marca 2020 – zakończenie przeglądu materiałów związanych z danym tematem
- 29 marca 2020 – przygotowanie schematu widoku aplikacji
- 12 kwietnia 2020 – oprogramowanie obsługi joysticka
- 19 kwietnia 2020 – zdefiniowanie protokołu komunikacji i budowy przesyłanych ramek
- 26 kwietnia 2020 – przygotowanie logiki sterownia
- 4 maja 2020 – implementacja dwustronnej komunikacji z robotem
- 10 maja 2020 – wizualizacja wskazań prędkości i naładowania baterii
- 17 maja 2020 – wizualizacja wskazań akcelometru
- 24 maja 2020 – przygotowanie wizualizacji obiektu 3D
- 31 maja 2020 – implementacja obracania obiektu 3D przy użyciu żyroskopu
- 7 czerwca 2020 – szukanie błędów i testowanie wszystkich funkcji
- 14 czerwca 2020 – ostateczne testy działania aplikacji



Rysunek 1: Diagram Gantta

## 5 Projekt interfejsu graficznego



Rysunek 2: Szablon interfejsu graficznego

Największy wycinek okna przeznaczony jest na prezentowanie modelu robota w trójwymiarze (1). Obiekt ten będzie obracał się zgodnie z wskazaniami akcelometru zamontowanego na realnym pojeździe. Będzie więc to wizualizacja ustawienia robota w przestrzeni.

Po prawej stronie widniejąc trzy wykresy prezentujące pomiary akcelometru w 3 osiach (2). Poniżej znajdują się kolejno wskaźniki opóźnienia komunikacji (3), prędkości liniowej pojazdu (4) i odległości od przeszkody (5), a także poziom naładowania baterii (6).

Na dolnej belce umieszczona jest informacja o podłączonym kontrolerze (7), i słownym statusie komunikacji z robotem (8).

Na szczycie aplikacji znajduje się belka narzędziowa (9), która zawiera opcję nawiązania/zerwania połączenia, wyjście z programu i informację o autorze aplikacji. Po wybraniu funkcji połączenia z robotem, wyświetli się dodatkowe okienko z możliwością wprowadzenia adresu sterowanego obiektu i przycisk umożliwiający inicjację połączenia.

## 6 Komunikacja z robotem

Połączenie z robotem odbywa się poprzez sieć WiFi. W pierwszej kolejności nawiązywane jest połączenie przy użyciu protokołu TCP. Jeśli się ono powiedzie to uruchamiana jest dodatkowa transmisja z wykorzystaniem UDP. Dzięki takiej koncepcji mamy dwa niezależne kanały komunikacji. Pierwszy służy do przesyłania danych które mają niski priorytet czasowy, potrzebują potwierdzenia odebrania i ewentualnej retransmisji danych. Druga droga komunikacji powstała, aby przesyłać ciągły strumień nowych danych. Zależy nam na jaknajniższym opóźnieniu, a ewentualny błąd transmisji nie jest krytyczny, bo informacje te szybko się przedawniają i są zastępowane przez nowe, świeższe.

Każda ramka zaczyna się od wybranej dużej litery alfabetu, określającej rodzaj przesyłanych danych. Dla protokołu TCP są to:

- P - ping,
- D - dystans przeszkody,
- B - bateria,
- S - realna predkość.

Natomiast dla protokołu UDP wyróżniamy:

- E - moc silników,
- A - akcelometr,
- G - żyroskop.

Wszystkie paczki danych zakończone są średnikiem, przed którym znajduje się ośmiobitowy cykliczny kod nadmiarowy. Niestety ze względu na różnorodność transmitowanych informacji, w tym miejscu kończą się cechy wspólne poszczególnych ramek.

### 6.1 Ping

Jest to najprostrza z obecnych tu ramek. Nie przenosi żadnych informacji. Oznacza jedynie konieczność odesłania do nadawcy identycznej wiadomości, aby można było wyznaczyć chwilę czasową, niezbędną do obliczenia opóźnienia transmisji.

Forma ramki to:  $P\#;$   
Gdzie  $\#$  oznacza CRC.

### 6.2 Dystans przeszkody

Przesyła informacje z robota o odległości odczytanej z czujnika ultradźwiękowego.

Forma ramki to:  $D < uint8\_t > \#;$

Gdzie  $\#$  oznacza CRC. Przed nim znajduje się wartość odległości wyrażonej w centymetrach, w zakresie 0-100cm.

## 6.3 Bateria

Przesyła informacje z robota o poziomie baterii.

Forma ramki to:  $B < uint8\_t > \#$ ;

Gdzie  $\#$  oznacza CRC. Przed nim znajduje się poziom baterii wyrażonej w procentach, w zakresie 0-100%.

## 6.4 Prędkość

Przesyła informacje z robota o prędkości na kołach.

Forma ramki to:  $S < uint8\_t > < uint8\_t > \#$ ;

Gdzie  $\#$  oznacza CRC. Przed nim znajdują się dwie wartości oddzielone spacją odnoszące się do prędkości poszczególnych kół wyrażonej w metrach na sekundę.

## 6.5 Moc silników

Przesyła informacje z aplikacji do robota o zadanej mocy silników.

Forma ramki to:  $E < uint8\_t > < uint8\_t > \#$ ;

Gdzie  $\#$  oznacza CRC. Przed nim znajdują się dwie wartości oddzielone spacją odnoszące się do zadanej mocy poszczególnych kół wyrażonej w procentach. Zakresy tych wartości muszą mieścić się od 0 do 100%

## 6.6 Akcelometr

Przesyła informacje z robota o aktualnych wskazaniach akcelometru.

Forma ramki to:  $A < uint8\_t > < uint8\_t > < uint8\_t > \#$ ;

Gdzie  $\#$  oznacza CRC. Przed nim znajdują się trzy wartości oddzielone spacją odnoszące się do aktualnych wskazań akcelometru.

## 6.7 Żyroskop

Przesyła informacje z robota o aktualnych wskazaniach żyroskopu.

Forma ramki to:  $G < uint8\_t > < uint8\_t > < uint8\_t > \#$ ;

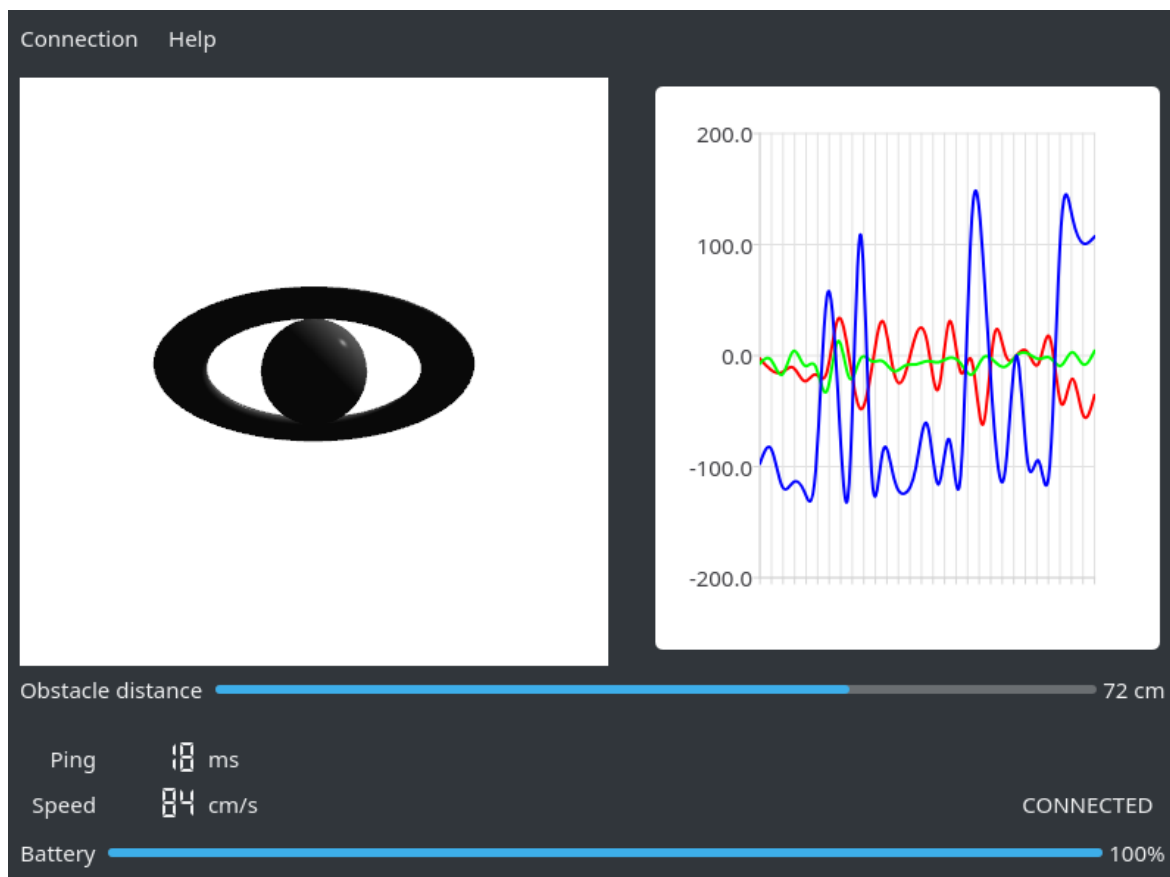
Gdzie  $\#$  oznacza CRC. Przed nim znajdują się trzy wartości oddzielone spacją odnoszące się do aktualnych wskazań żyroskopu.

# 7 Wykres akcelometru

Jednym z założeń projektu było umieszczenie wykresów przyspieszeń wszystkich trzech osi sterowanego pojazdu. Została podjęta decyzja scalenia tych wskazań do jednego dużego wykresu w celu zwiększenia czytelności. Przełożyło się to także na znaczną poprawę wydajności aplikacji. W celu dalszej optymalizacji zaimplementowałem algorytm, który pozbywa się wskazań wychodzących poza okno aplikacji. Wszystkie te działania

sprawiły że dodanie tej funkcjonalności nie wpłynęło znacząco na ogólną złożoność obliczeniową aplikacji.

Wykres wyskalowany został w przedziale od -2G do +2G. Każda z odczytywanych osi ma swój własny kolor. W przypadku osi Z jest to kolor niebieski. Dla osi X wybrany został kolor zielony, natomiast osi Y przypadł kolor czerwony.



Rysunek 3: Wykresy wskazań akcelometru

## 8 Aktualne rezultaty

Na dzień dzisiejszy, projekt nie jest obciążony żadnymi opóźnieniami. Wszystkie zaplanowane kamienie milowe zostały osiągnięte przed terminem. Aplikacja może poszczycić się już działającą obsługą joysticka, gotową komunikacją z robotem, wykresem przeciężeń, algorytmem różnicowego układu sterowania oraz działającym wskaźnikiem opóźnienia, prędkości liniowej i poziomu baterii. Pozostało więc jedynie zaimplementować wizualizację obiektu 3D. Ostatnia faza projektu zakłada również eliminację błędów.