Oprogramowanie sterujące wirtualnym modelem robota klasy Micromouse

Agnieszka Dawid
Promotor: dr hab. inż. Wojciech Skarka, prof. Pol. Śl.
Opiekun: dr inż. Wawrzyniec Panfil

Instytut Podstaw Konstrukcji Maszyn
Politechnika Śląska

Gliwice, 2018

Plan prezentacji

- Cel projektu i zakres prac
- Zawody Micromouse - studium literatury
- Specyfikacja robota
- Wirtualne modele robota i labiryntu
- Oprogramowanie
- Weryfikacja działania oprogramowania
- Podsumowanie i wnioski

Cel projektu i zakres prac

Cel projektu

Opracowanie oprogramowania sterującego wirtualnym modelem robota klasy *Micromouse* w środowisku symulacyjnym V-Rep.

Cel projektu i zakres prac

Projekt obejmuje następujące zadania:

- studium literatury w zakresie zawodów Micromouse,
- zdefiniowanie założeń,
- opracowanie wirtualnych modeli robota oraz labiryntu w wybranym środowisku symulacyjnym,
- opracowanie oprogramowania do obsługi układów sensorycznych i wykonawczych wirtualnego modelu robota,
- weryfikację działania poprawności programów.

Cel projektu i zakres prac

Projekt obejmuje następujące zadania:

- studium literatury w zakresie zawodów Micromouse,
- zdefiniowanie założeń,
- opracowanie wirtualnych modeli robota oraz labiryntu w wybranym środowisku symulacyjnym,
- opracowanie oprogramowania do obsługi układów sensorycznych i wykonawczych wirtualnego modelu robota,
- weryfikację działania poprawności programów.

Cel projektu i zakres prac

Projekt obejmuje następujące zadania:

- studium literatury w zakresie zawodów Micromouse,
- zdefiniowanie założeń,
- opracowanie wirtualnych modeli robota oraz labiryntu w wybranym środowisku symulacyjnym,
- opracowanie oprogramowania do obsługi układów sensorycznych i wykonawczych wirtualnego modelu robota,
- weryfikację działania poprawności programów.

Cel projektu i zakres prac

Projekt obejmuje następujące zadania:

- studium literatury w zakresie zawodów Micromouse,
- zdefiniowanie założeń,
- opracowanie wirtualnych modeli robota oraz labiryntu w wybranym środowisku symulacyjnym,
- opracowanie oprogramowania do obsługi układów sensorycznych i wykonawczych wirtualnego modelu robota,
- weryfikację działania poprawności programów.

Cel projektu i zakres prac

Projekt obejmuje następujące zadania:

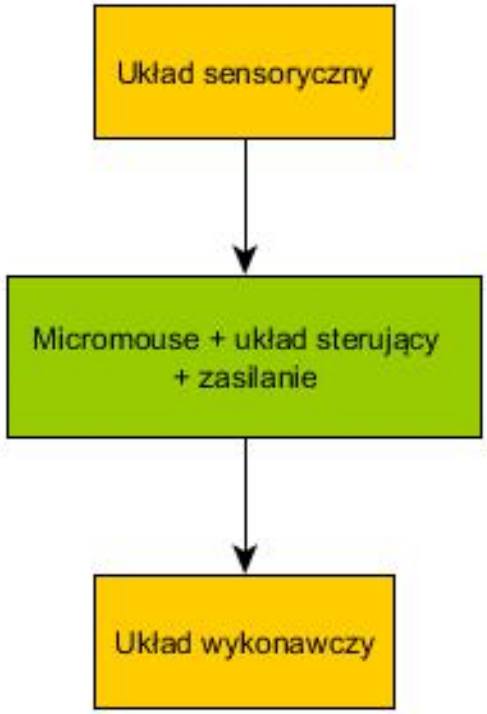
- studium literatury w zakresie zawodów Micromouse,
- zdefiniowanie założeń,
- opracowanie wirtualnych modeli robota oraz labiryntu w wybranym środowisku symulacyjnym,
- opracowanie oprogramowania do obsługi układów sensorycznych i wykonawczych wirtualnego modelu robota,
- weryfikację działania poprawności programów.

Zawody Micromouse - studium literatury

	Rybnik	Wrocław	Poznań	Warszawa	Kraków	Łódź
Wymiary labiryntu [mm]	2000x2000	1440x1440 2880x2880	-	2880x2880	2880x2880	1620x1620
Ilość pól	11x11	8x8 16x16	Podane na miesiąc przed zawodami	16x16	16x16	9x9
Max wymiary robota [mm]	160x160x400 [dłxszerxh]	168x168xh	250x250xh	168x168xh	250x250xh	250x250xh

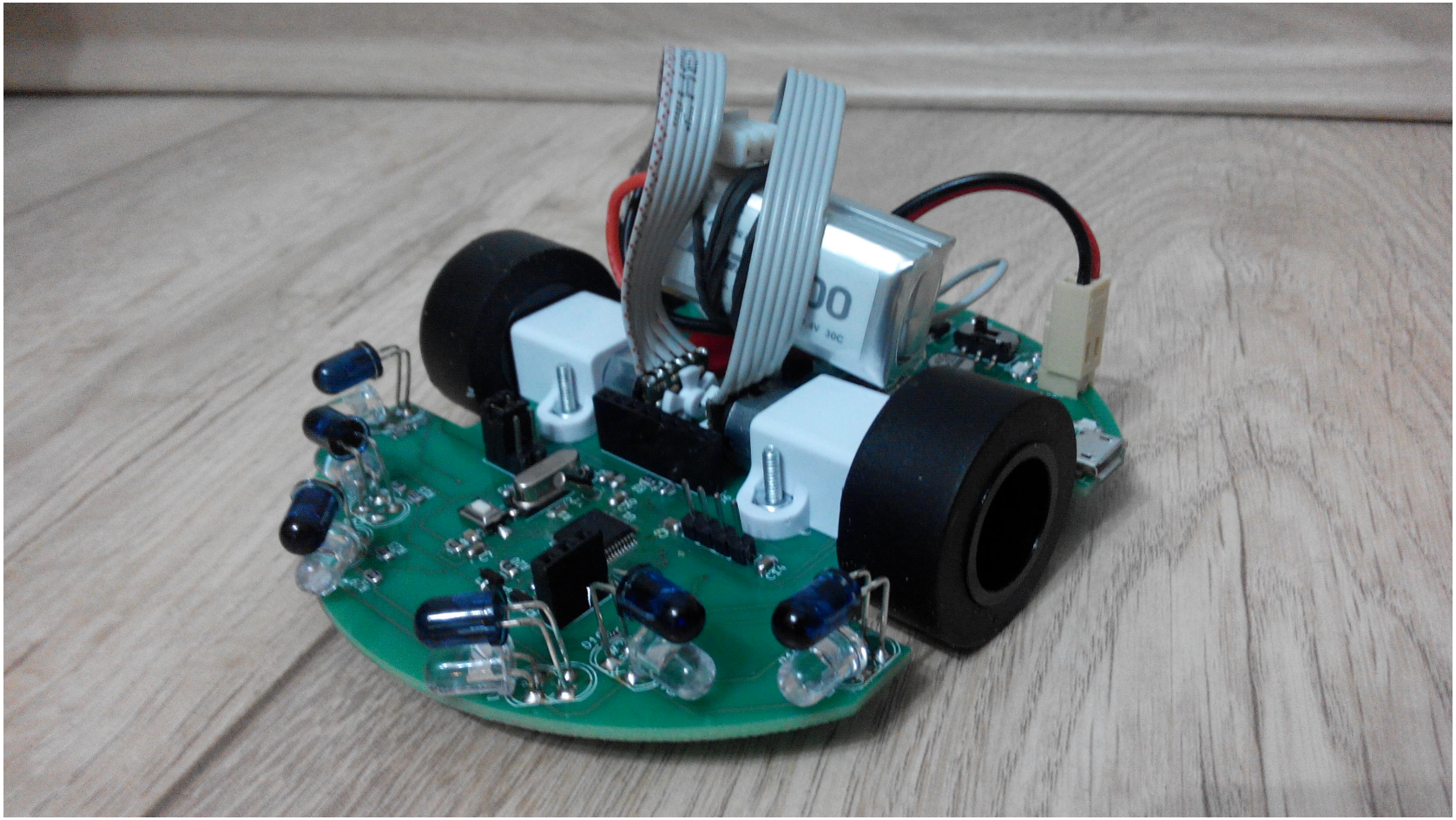
Rysunek: Porównanie regulaminów zawodów Micromouse[1,2,3,4,5,6,7]

Specyfikacja robota



Rysunek: Schemat robota

Specyfikacja robota



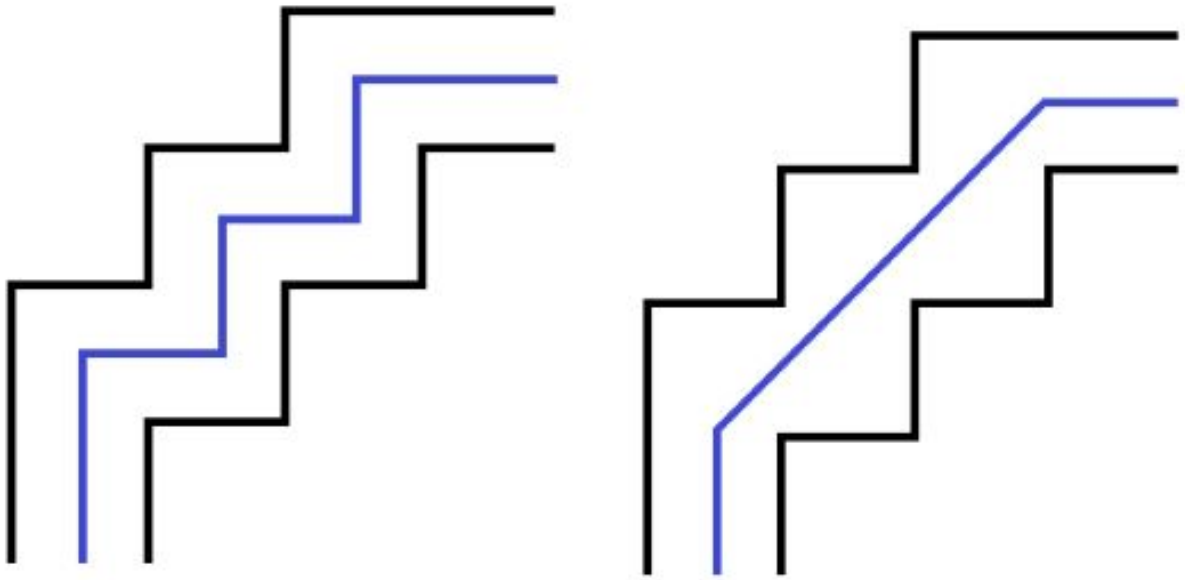
Rysunek: Rzeczywisty robot

Specyfikacja robota

Tablica: Dane techniczne robota

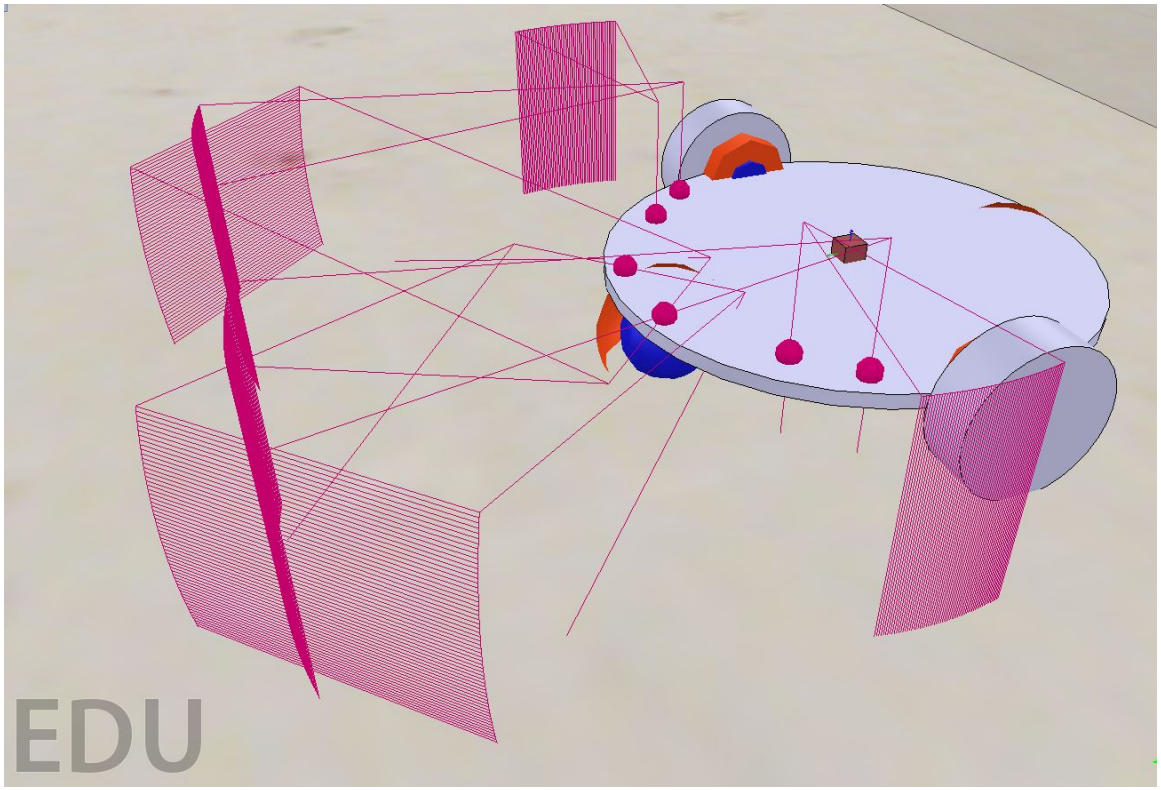
Wymiary	106x90[mm]
Waga	-
Procesor	STM32F103C8T6
Bateria	Pakiet Li-Pol Dualsky 300mAh 30C 2S 7.4V
Silniki	Micro Pololu HP z przekładnią 30:1
Enkodery	Pololu
Dioda IR	TSAL-6400
Fototranzystor	LIRT5b
Żyroskop z akcelerometrem	L3GD20
Prędkość na prostej	2 m/s

Specyfikacja robota



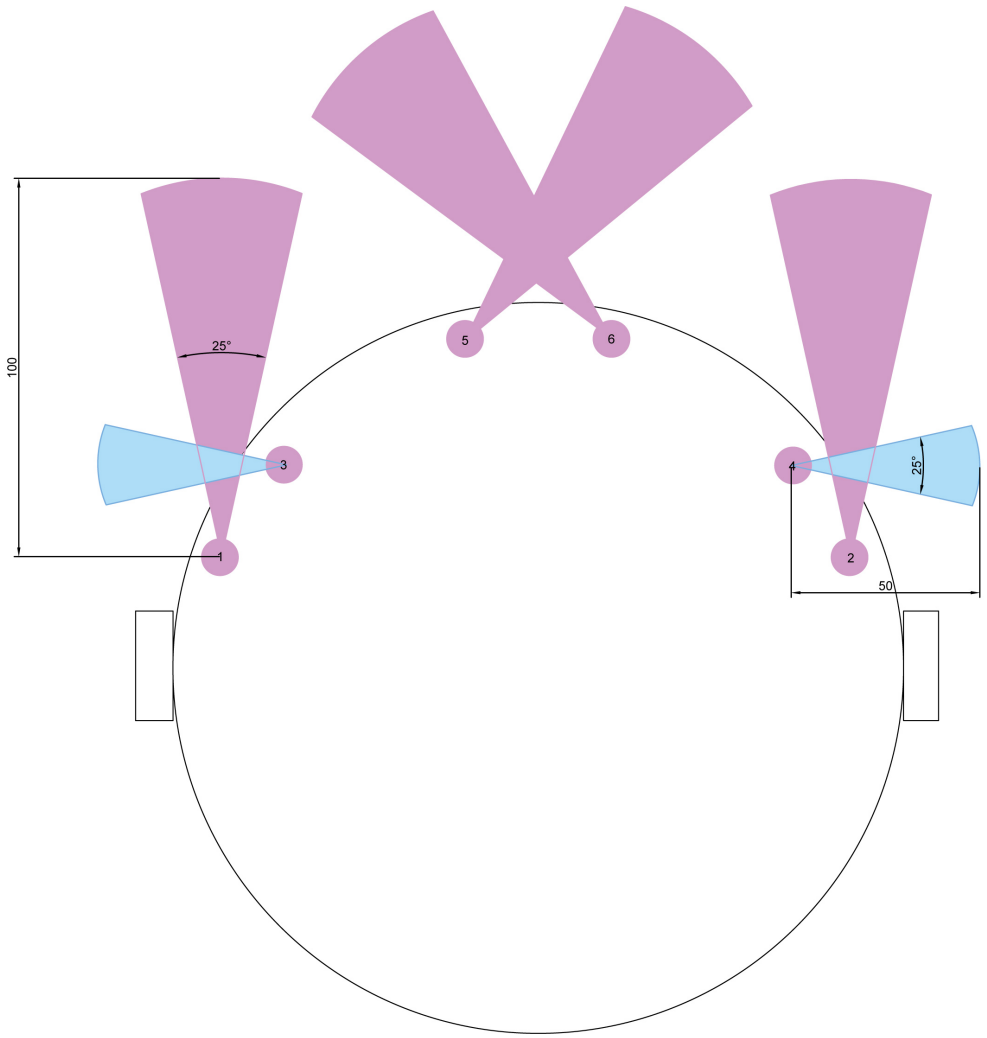
Rysunek: Sposoby pokonywania zakrętów[9]

Wirtualne modele robota i labiryntu



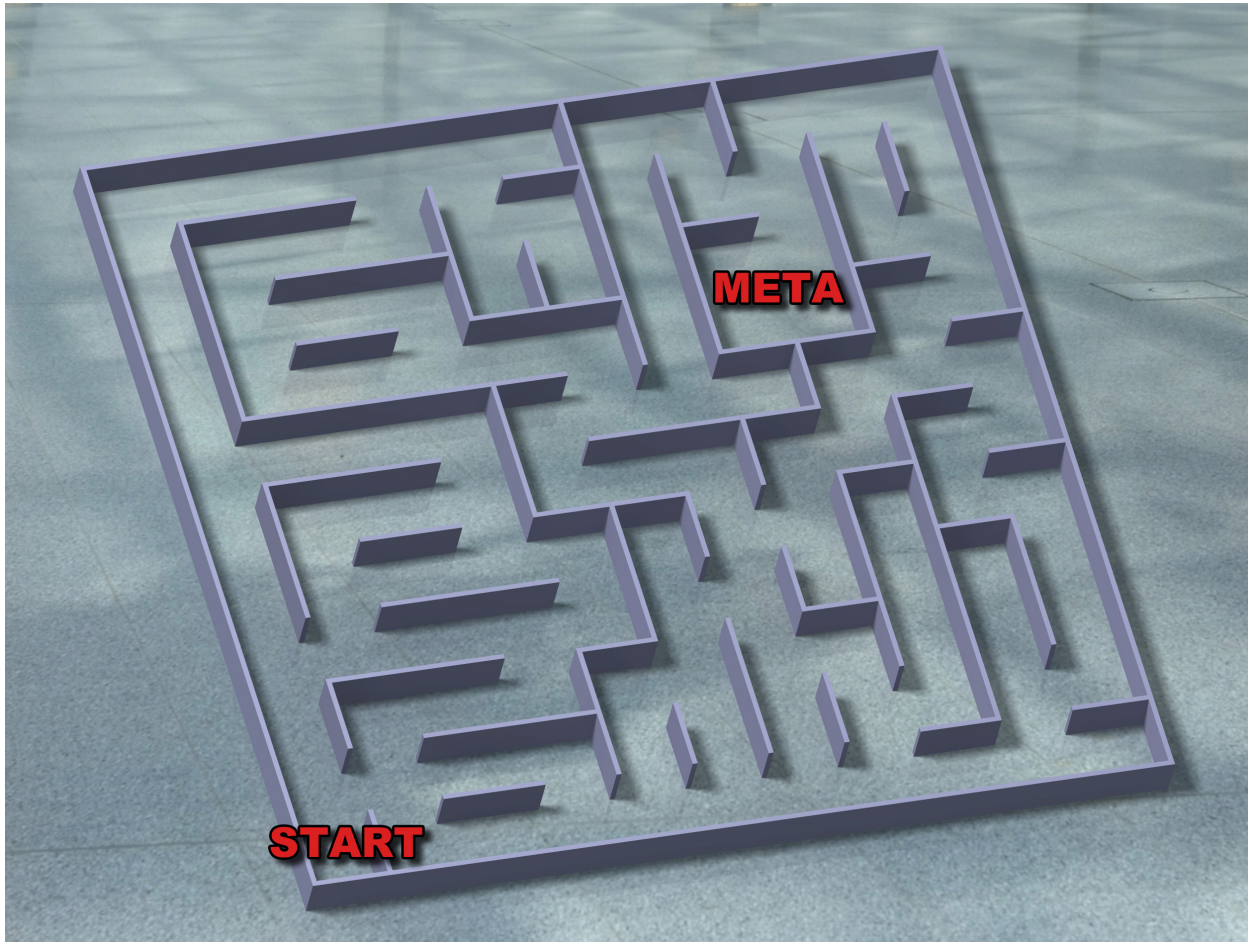
Rysunek: Model robota w środowisku symulacyjnym

Wirtualne modele robota i labiryntu



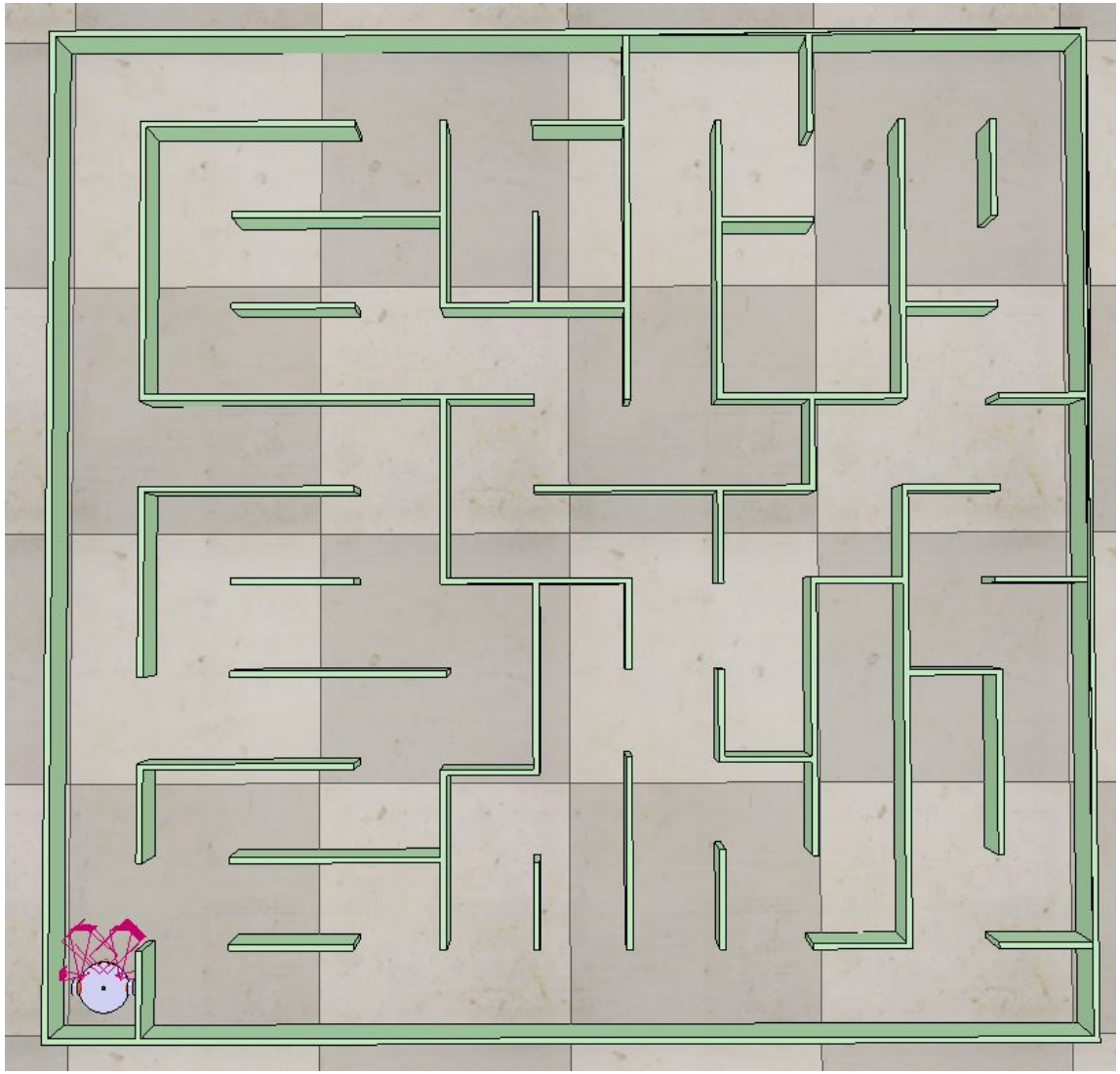
Rysunek: Zakres działania czujników

Wirtualne modele robota i labiryntu



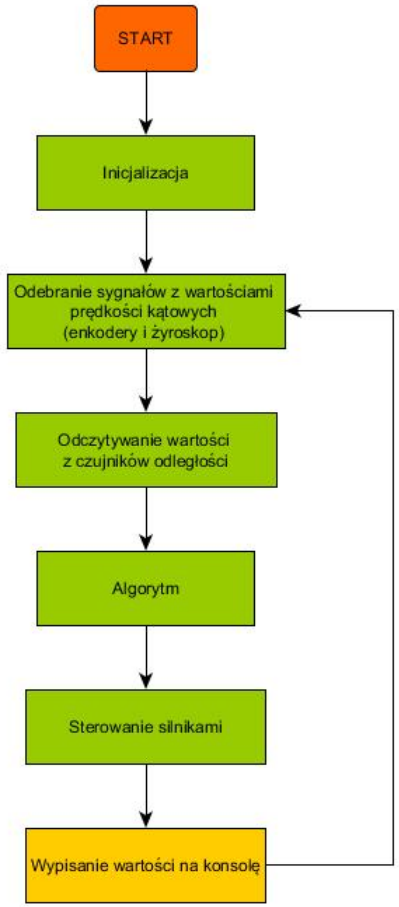
Rysunek: Model labiryntu

Wirtualne modele robota i labiryntu



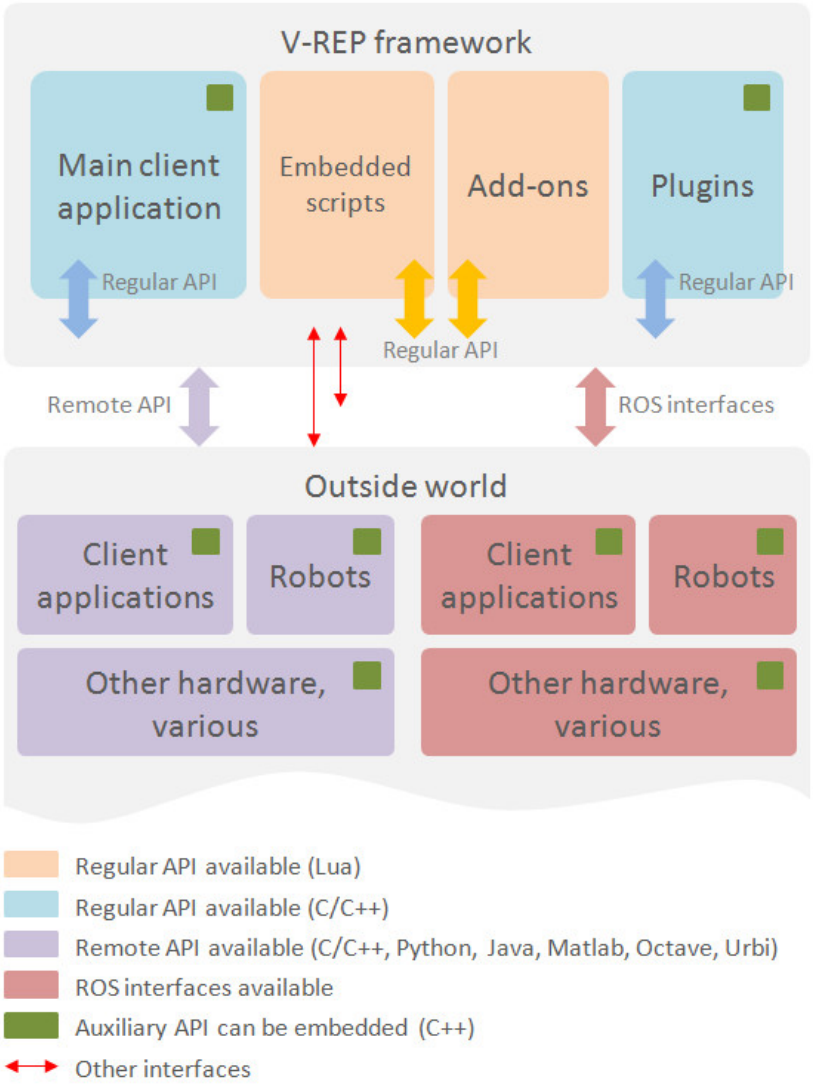
Rysunek: Robot w labiryncie w środowisku symulacyjnym V-Rep

Oprogramowanie



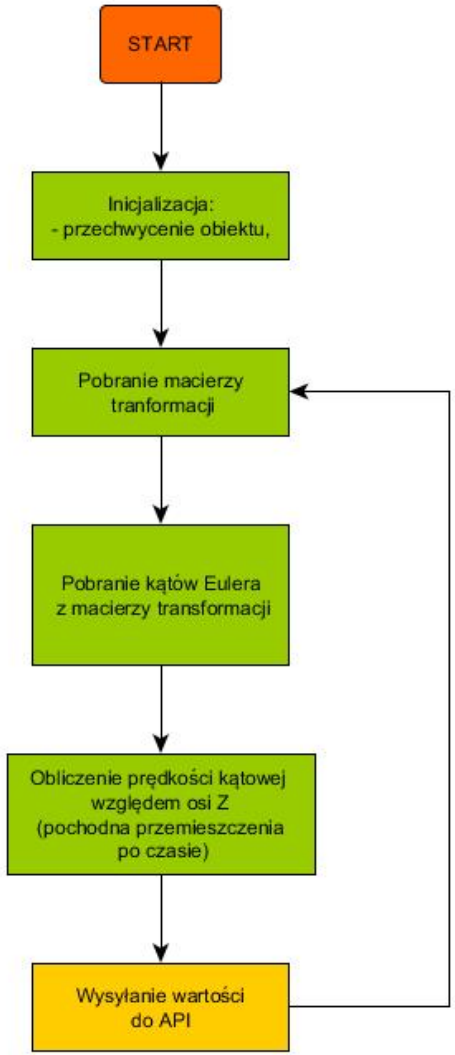
Rysunek: Schemat kodu głównego

Oprogramowanie



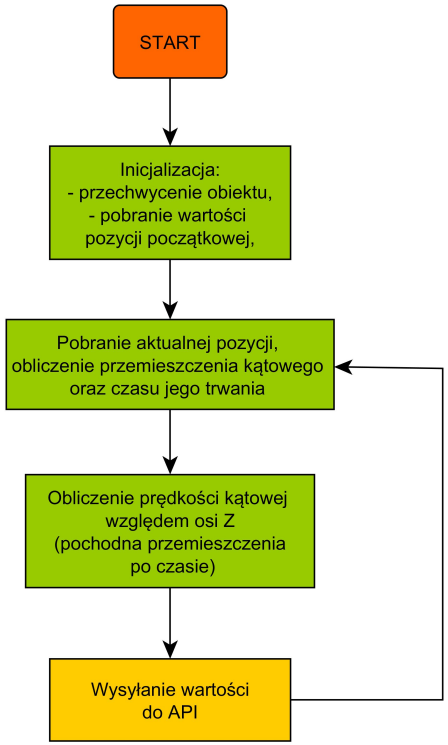
Rysunek: Schemat działania API[8]

Oprogramowanie



Rysunek: Schemat działania kodu dla żyroskopu

Oprogramowanie



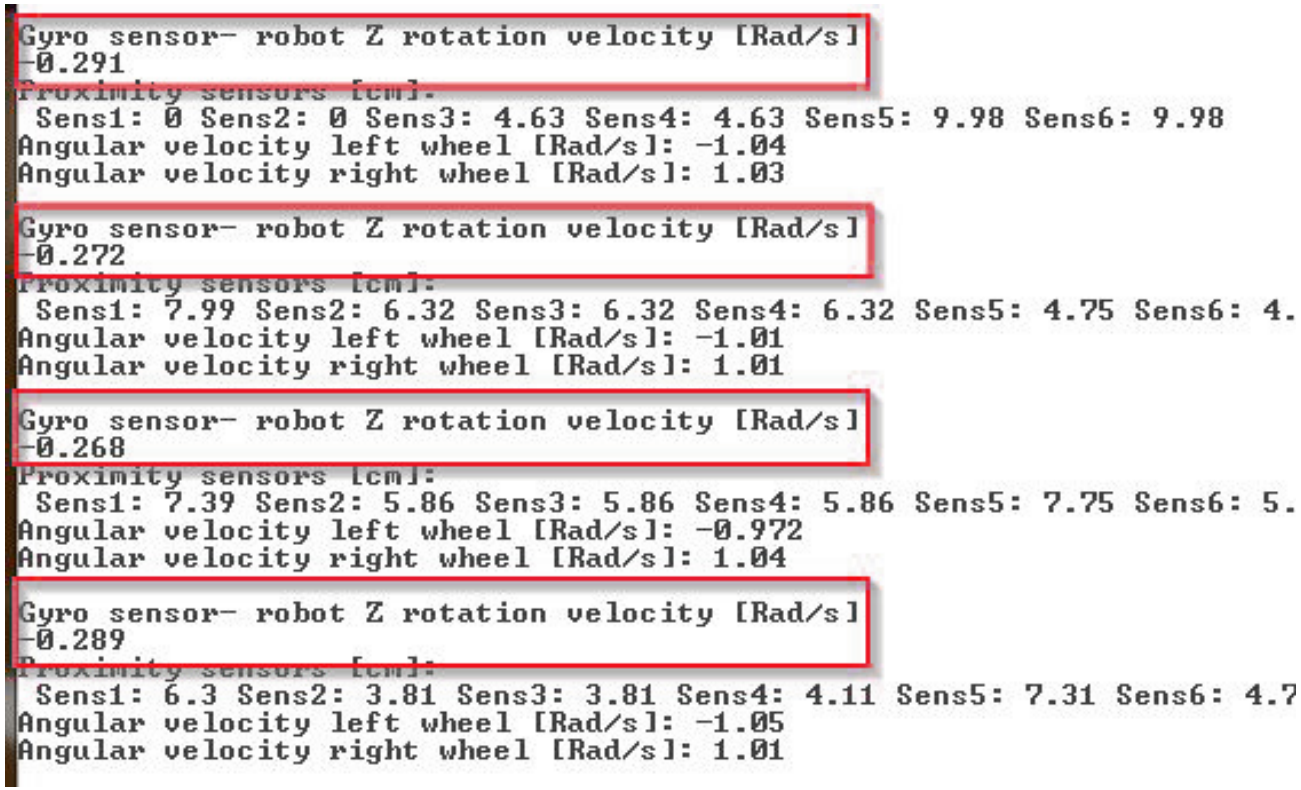
Rysunek: Schemat działania kodu dla enkoderów

Weryfikacja działania oprogramowania

```
Connected to sensorIR6
Gyro sensor- robot Z rotation velocity [Rad/s]
0.00865
Proximity sensors [cm]:
 Sens1: 0 Sens2: 0 Sens3: 4.52 Sens4: 4.52 Sens5: 9.49 Sens6: 9.49
Angular velocity left wheel [Rad/s]: 1.01
Angular velocity right wheel [Rad/s]: 1.01

Gyro sensor- robot Z rotation velocity [Rad/s]
0.0108
Proximity sensors [cm]:
 Sens1: 0 Sens2: 0 Sens3: 4.53 Sens4: 4.53 Sens5: 9.63 Sens6: 9.63
Angular velocity left wheel [Rad/s]: 0.98
Angular velocity right wheel [Rad/s]: 1.02

Gyro sensor- robot Z rotation velocity [Rad/s]
-0.00255
Proximity sensors [cm]:
 Sens1: 0 Sens2: 0 Sens3: 4.55 Sens4: 4.55 Sens5: 9.71 Sens6: 9.71
Angular velocity left wheel [Rad/s]: 1.03
Angular velocity right wheel [Rad/s]: 1.02

Gyro sensor- robot Z rotation velocity [Rad/s]
-0.00916
Proximity sensors [cm]:
 Sens1: 0 Sens2: 0 Sens3: 4.6 Sens4: 4.6 Sens5: 9.78 Sens6: 9.78
Angular velocity left wheel [Rad/s]: 0.997
Angular velocity right wheel [Rad/s]: 1
```

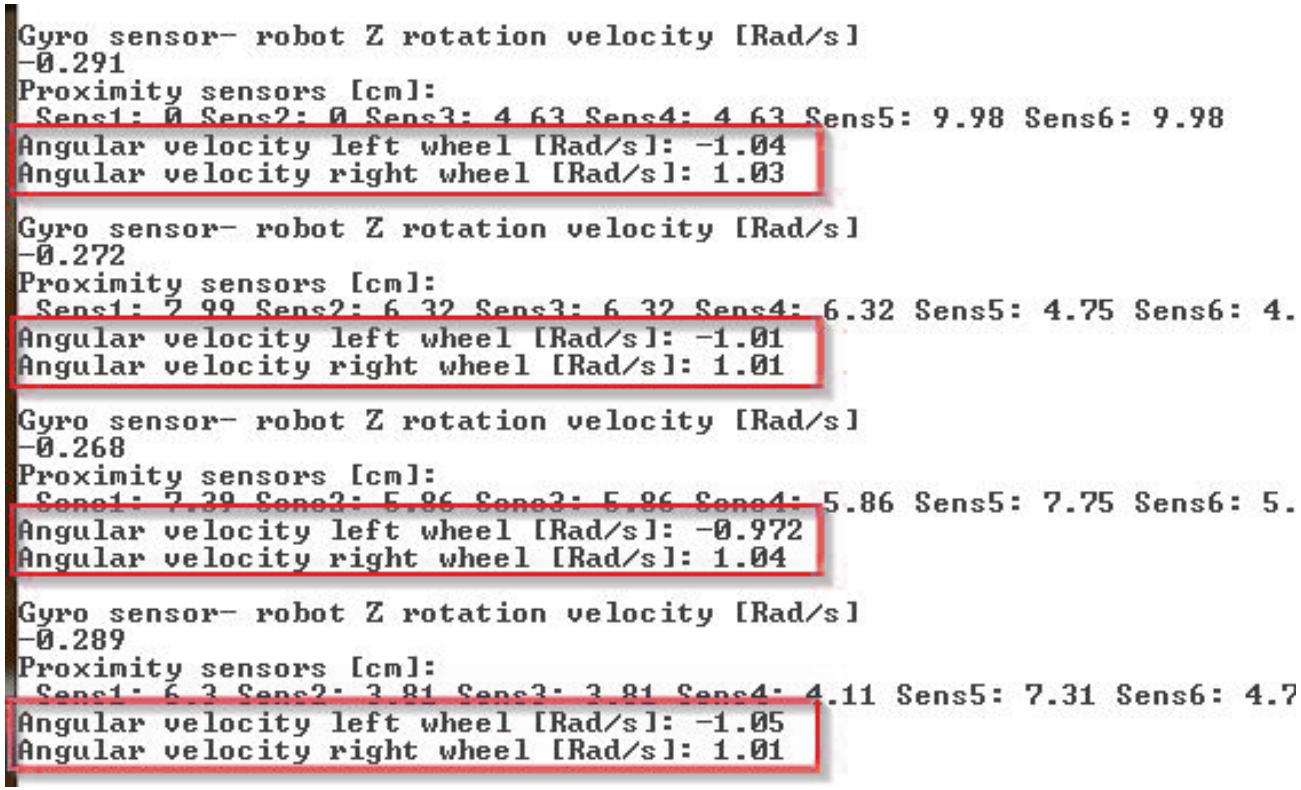
Rysunek: Konsola użytkownika

Weryfikacja działania oprogramowania



Rysunek: Wyświetlanie wartości żyroskopu

Weryfikacja działania oprogramowania



Rysunek: Wyświetlanie wartości dla enkoderów

Podsumowanie i wnioski

Aby zrealizować cel wykonano:

- model robota i labiryntu,
- oprogramowanie dla układu sensorycznego i wykonawczego robota,
- weryfikację w środowisku symulacyjnym V-Rep.

Podsumowanie i wnioski

Aby zrealizować cel wykonano:

- model robota i labiryntu,
- oprogramowanie dla układu sensorycznego i wykonawczego robota,
- weryfikację w środowisku symulacyjnym V-Rep.

Podsumowanie i wnioski

Aby zrealizować cel wykonano:

- model robota i labiryntu,
- oprogramowanie dla układu sensorycznego i wykonawczego robota,
- weryfikację w środowisku symulacyjnym V-Rep.

Podsumowanie i wnioski

Wnioski, jakie nasunęły się podczas wykonywania projektu, są następujące:

- środowisko symulacyjne jest przyjazne dla użytkownika, ponieważ są dostępne modele poszczególnych podzespołów robota, w których należy ustawić parametry działania,
- bez wykonanego modelu rzeczywistego można testować napisane oprogramowanie na poszczególne elementy robota, co znacznie przyspiesza pracę,
- symulacja daje możliwość równoczesnej pracy nad robotem rzeczywistym i jego oprogramowaniem bez przeszkadzania sobie nawzajem,

Podsumowanie i wnioski

Wnioski, jakie nasunęły się podczas wykonywania projektu, są następujące:

- środowisko symulacyjne jest przyjazne dla użytkownika, ponieważ są dostępne modele poszczególnych podzespołów robota, w których należy ustawić parametry działania,
- bez wykonanego modelu rzeczywistego można testować napisane oprogramowanie na poszczególne elementy robota, co znacznie przyspiesza pracę,
- symulacja daje możliwość równoczesnej pracy nad robotem rzeczywistym i jego oprogramowaniem bez przeszkadzania sobie nawzajem,

Podsumowanie i wnioski

Wnioski, jakie nasunęły się podczas wykonywania projektu, są następujące:

- środowisko symulacyjne jest przyjazne dla użytkownika, ponieważ są dostępne modele poszczególnych podzespołów robota, w których należy ustawić parametry działania,
- bez wykonanego modelu rzeczywistego można testować napisane oprogramowanie na poszczególne elementy robota, co znacznie przyspiesza pracę,
- symulacja daje możliwość równoczesnej pracy nad robotem rzeczywistym i jego oprogramowaniem bez przeszkadzania sobie nawzajem,

Podsumowanie i wnioski

- można sprawdzać kilka koncepcji robotów bez potrzeby ich wykonywania, co oszczędza czas i ogranicza koszty,
- można sprawdzić w krótkim czasie zachowanie robota w różnych labiryntach, dzięki czemu praca nad algorytmem sterującym przebiega szybciej. Dzięki temu można uniknąć zbytniego dopasowania algorytmu pod konkretny labirynt, co może mieć miejsce podczas testów na torach rzeczywistych,
- wyświetlanie wartości parametrów na konsoli znacznie ułatwia weryfikację pisanego programu.

Podsumowanie i wnioski

- można sprawdzać kilka koncepcji robotów bez potrzeby ich wykonywania, co oszczędza czas i ogranicza koszty,
- można sprawdzić w krótkim czasie zachowanie robota w różnych labiryntach, dzięki czemu praca nad algorytmem sterującym przebiega szybciej. Dzięki temu można uniknąć zbytniego dopasowania algorytmu pod konkretny labirynt, co może mieć miejsce podczas testów na torach rzeczywistych,
- wyświetlanie wartości parametrów na konsoli znacznie ułatwia weryfikację pisanego programu.

Podsumowanie i wnioski

- można sprawdzać kilka koncepcji robotów bez potrzeby ich wykonywania, co oszczędza czas i ogranicza koszty,
- można sprawdzić w krótkim czasie zachowanie robota w różnych labiryntach, dzięki czemu praca nad algorytmem sterującym przebiega szybciej. Dzięki temu można uniknąć zbytniego dopasowania algorytmu pod konkretny labirynt, co może mieć miejsce podczas testów na torach rzeczywistych,
- wyświetlanie wartości parametrów na konsoli znacznie ułatwia weryfikację pisanego programu.

Bibliografia

1. Regulamin zawodów Robotic Tournament 2017 w kategorii Micromouse w Rybniku.
2. Regulamin zawodów Robotic Arena 2017 w kategorii Micromouse we Wrocławiu, 16x16 pól.
3. Regulamin zawodów Robotic Arena 2017 w kategorii Micromouse we Wrocławiu, 8x8 pól.
4. Regulamin zawodów w kategorii Micromouse w Poznaniu.
5. Regulamin zawodów w kategorii Micromouse w Warszawie.
6. Regulamin zawodów w kategorii Micromouse w Krakowie, str. 9-11.
7. Regulamin zawodów w kategorii Micromouse w Łodzi.
8. Wykonanie remote API: http://www.coppeliarobotics.com/helpFiles/en/apisOverview.htm, dostęp:grudzień 2017.
9. Chojnacki Ł., Praca dyplomowa inżynierska: Projekt optymalnego układu pomiarowego dla robota typu „micromouse", 2013, Wydział Mechaniczny, Politechnika Wrocławska.

Dziękuję za uwagę