

WSTĘP DO ROBOTYKI

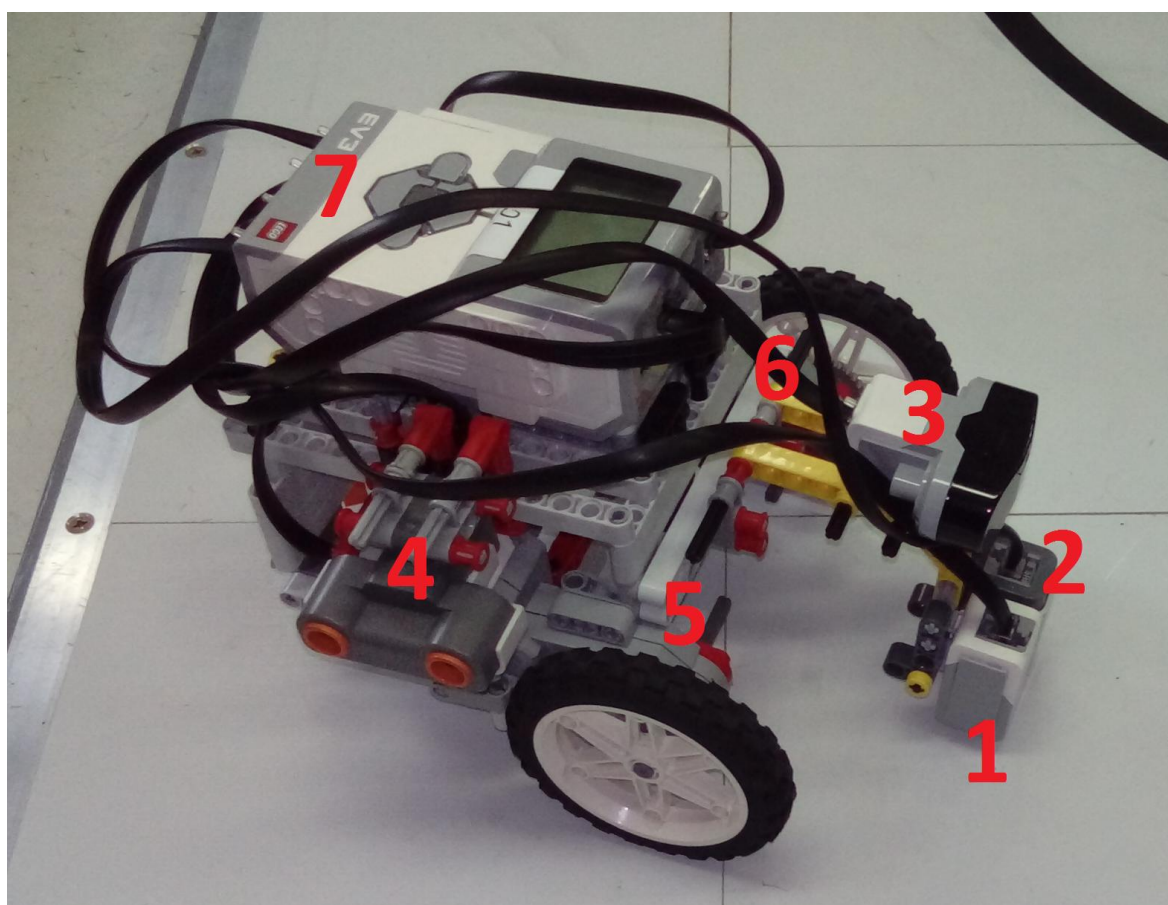
Sprawozdanie z laboratorium

Skład zespołu:

Adam Kozłowski

Marcin Słoka

1. Budowa robota



Zdjęcie 1: Robot wykonany w ramach laboratorium z przedmiotu Wstęp do Robotyki

Widoczny na *Zdjęciu 1* robot został zbudowany z klocków LEGO z wykorzystaniem zestawu MINDSTORMS EV3. Możemy wyróżnić następujące elementy:

1. Czujnik koloru.
2. Czujnik światła.
3. Czujnik podczerwieni

4. Sonar.
5. Silnik duży (prawy)
6. Silnik duży (lewy)
7. Kłoczek EV3

Czujniki 1 i 2 umożliwiają podążanie za linią, zaś czujniki 3 i 4 pozwalają na ominięcie przeszkody. Silniki 5 i 6 odpowiadają za sterowanie odpowiednio prawym i lewym kołem. Wszystkie elementy są podłączone do klocka EV3 – komputera.

Pod względem konstrukcyjnym robot jest sztywny i stabilny. Z tyłu posiada samonastawne koło, dlatego tor jego ruchu zależy tylko od sterowania dwóch przednich kół. Jego relatywnie krótka konstrukcja pozwala na obrót o 90 stopni bez wyjechania poza obszar jednego kafelka stołu.

2. Algorytm oraz jego implementacja

Podążanie za linią

Podążanie za linią zostało zrealizowane w oparciu o maszynę stanów. Robot dąży do takiego ustawienia, aby jego prawy czujnik (koloru) znajdował się nad linią, zaś lewy czujnik (światła) był z lewej strony linii. Czujniki mogą się znajdować w następujących stanach:

- lewy na białym, prawy na czarnym: robot znajduje się we właściwej pozycji; reakcja: jedź prosto
- lewy na białym, prawy na białym: robot zjeżdża w lewo; reakcja: kręć szybciej lewym kołem
- lewy na czarnym, prawy na białym: robot zjeżdża w prawo; reakcja: kręć szybciej prawym kołem
- lewy na czarnym, prawy na czarnym: niejednoznaczność; reakcja: przejedź kawałek dalej i sprawdź ponownie czujniki
 - jeśli lewy na białym, prawy na białym: ostry zakręt w lewo; reakcja: kręć szybciej prawym kołem, a lewym kołem kręć wstecz – obrót w miejscu, dopóki nie trafisz na linię
 - w przeciwnym wypadku: to było skrzyżowanie; reakcja: nie zmieniaj toru jazdy

Ponadto, program zlicza przez ile obiegów pętli robot znajduje się nieprzerwanie w danym stanie. Jeśli licznik osiągnie pewną ustaloną wartość dla stanu biały/biały lub czarny/biały, oznacza to, że robot dojechał do ostrego zakrętu (niekoniecznie 90 stopni, ale nie jest to też łagodny łuk). Robot zaczyna skręcać w miejscu, odpowiednio w lewo lub w prawo, dopóki nie natrafi z powrotem na linię.

Omijanie przeszkody

Wielkość i ustawienie przeszkody były dobrze zdefiniowane, dlatego część algorytmu (kroki 1, 2, 3, 6) mają przypisane na stałe wartości, które nie zależą od odczytu czujników. Algorytm prezentuje się następująco:

Jeśli czujnik przedni (podczerwieni) znajdzie przeszkodę, to:

1. Skręć w miejscu w lewo o ok. 60 stopni.
2. Jedź prosto, aż dojedziesz na sąsiedni kafelek.
3. Skręć w miejscu w prawo o ok. 60 stopni.
4. Jedź prosto, dopóki czujnik boczny (sonar) nie odnajdzie przeszkody.
5. Jedź prosto, dopóki czujnik boczny nie przestanie wykrywać przeszkody.
6. Skręć w miejscu w prawo o ok. 60 stopni.
7. Jedź prosto, dopóki czujnik koloru nie natrafi na linię.
8. Powrót do głównej pętli programu.

Testy pokazały, że pomimo predefiniowanych wartości, robot zachowywał się przewidywalnie. Dojeżdżał do wyznaczonego punktu z dokładnością wystarczającą, aby bezproblemowo ominąć przeszkodę.

Implementacja

Program sterowania robotem został napisany strukturalnie w języku Python. Nie licząc kalibracji (patrz pkt. 3. *Dobieranie parametrów*), ciało programu istnieje w obrębie jednej, nieskończonej pętli. Przy każdym obiegu sprawdzane jest, czy czujnik podczerwieni nie wykrył przeszkody. Jeśli tak, uruchamia się fragment kodu odpowiedzialny za omijanie przeszkody. W przeciwnym wypadku, wykonuje się maszyna stanów, która realizuje podążanie za linią.

Dokładną implementację można zobaczyć w załączonym kodzie źródłowym, którego poszczególne elementy zostały dość dokładnie skomentowane.

3. Dobieranie parametrów

Czujniki koloru i światła są kalibrowane za każdym razem przed rozpoczęciem głównej pętli programu. Robota należy ustawić na linii, a następnie uruchomić kalibrację. Robot skręci w miejscu w lewo, następnie w prawo, a potem znowu w lewo, wracając do położenia wyjściowego. Podczas ruchu robota, czujniki cały czas odczytują wartości i zapamiętują pomiar maksymalny (najbardziej biały) oraz minimalny (najbardziej czarny), osobno dla każdego czujnika. Następnie dla każdego z czujników wyliczana jest średnia, którą można uznać za próg powyżej którego czujniki *widzą* biały oraz poniżej którego *widzą* czarny.

Za pomocą doświadczeń empirycznych zauważyliśmy, że robot zachowuje się lepiej, kiedy granica biały – czarny jest przesunięta tak, że kolor biały zajmuje szersze spektrum. Robot jeździ wtedy płynniej, nie dokonuje niepotrzebnych korekcji toru jazdy. W związku z tym, w naszym programie od wyliczonej średniej odejmujemy predefiniowaną wartość.

Przed zamontowaniem w robocie czujnika podczerwieni oraz sonaru, sprawdziliśmy jakie wartości zwracają one przy danej odległości od obiektu. Na podstawie tych pomiarów dobraliśmy takie

wartości, przy których robot ma uznać, że przeszkoda istnieje. Tak przyjęte parametry delikatnie modyfikowaliśmy podczas testów.

Prędkość obrotową silników empirycznie dobraliśmy tak, aby robot za wszelką cenę nie był w stanie zgubić linii. Wymagało to spowolnienia robota względem pierwotnych założeń oraz zaimplementowania skręcania w miejscu (uruchamiane w pewnych warunkach). Jest ono wolniejsze, ale szybciej naprowadza robota na pożądany tor.

Wartości predefiniowane (fragment algorytmu omijania przeszkody; ruch przy kalibracji czujników; czas jechania dalej przy niejednoznaczności czarne-czarne), dobieraliśmy w następujący sposób: najpierw na podstawie analizy budowy robota i zachowania jego silników, dobieraliśmy najbardziej prawdopodobne wartości, a następnie podczas testów modyfikowaliśmy je tak, aby robot zachowywał się idealnie zgodnie z naszymi oczekiwaniami.

4. Wady oraz zalety przyjętego rozwiązania

Zalety:

- maszyna stanów jest prosta w implementacji oraz jej działanie jest łatwe w obserwacji, co pozwala na unikanie błędów i łatwe modyfikacje,
- implementacja strukturalna pozwala na łatwą obserwację przebiegu programu,
- wykorzystanie dwóch czujników do śledzenia linii pozwala na precyzyjniejszą kontrolę nad torem jazdy robota oraz wykrywanie skrzyżowań (praktycznie niemożliwe przy jednym czujniku),
- umieszczenie czujnika podczerwieni z przodu i sonaru z boku, zapewnia większą przewidywalność odczytu, niż korzystanie z jednego ruchomego czujnika odległości, znajdującego się na obrotowym wysięgniku,
- sztywna, stabilna konstrukcja robota zwiększa deterministyczność jego ruchu.

Wady:

- maszyna stanów zmniejsza płynność ruchów; w przeciwieństwie do regulatora PID nie pozwala na łagodną zmianę toru jazdy,
- zastosowanie dwóch nieruchomych czujników odległości umożliwia ominięcie przeszkody tylko z jednej strony,
- sztywna, stabilna konstrukcja jest niestety ciężka, co negatywnie odbija się na prędkości jazdy robota oraz zużyciu energii przez silniki.