

KoNaR

KOŁO NAUKOWE ROBOTYKÓW

TIGER

Autonomiczny robot mobilny typu Line Follower

Sebastian Sadurski
Marcin Stolarek

Koło Naukowe Robotyków „KoNaR”
Wrocław, 2012r.

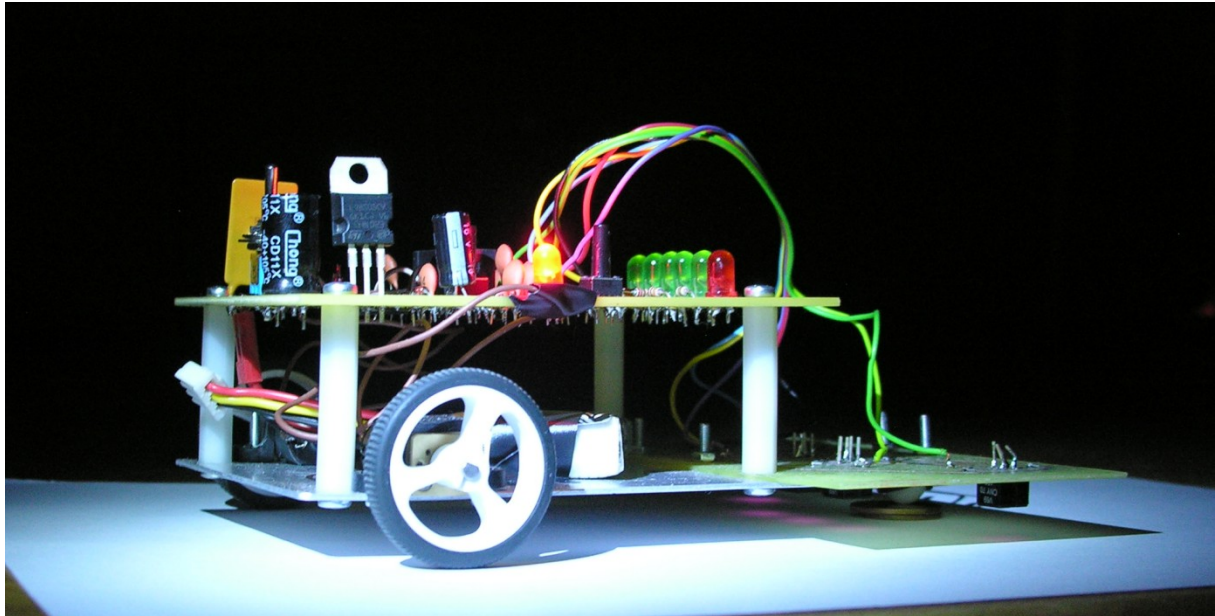
1.Wstęp.....	3
2.Konstrukcja robota.....	3
3.Zasilanie.....	5
4.Mikrokontroler i elementy elektroniczne.....	5
5.Czujniki czarnej linii.....	6
6.Napęd.....	7
7.Schemat elektroniczny i płytki PCB.....	7
8.Algorytm sterujący robotem.....	8
9.Zawody Robotic Arena 2011.....	9
10.Uwagi końcowe.....	10

1. Wstęp

Robot został wykonany podczas warsztatów Koła Naukowego Robotyków „KoNaR”.

Jest to nasza pierwsza konstrukcja i naszym celem było, aby wzięła udział w zawodach Robotic Arena 2011 i bezproblemowo przejechała wyznaczone trasy. Choć „Tiger” powstawał przez około dwa miesiące to jednak większość pracy wykonaliśmy w ciągu ostatnich dwóch tygodni, a umiejętność podążania robota za linią została dopracowana dopiero dzień przed zawodami.

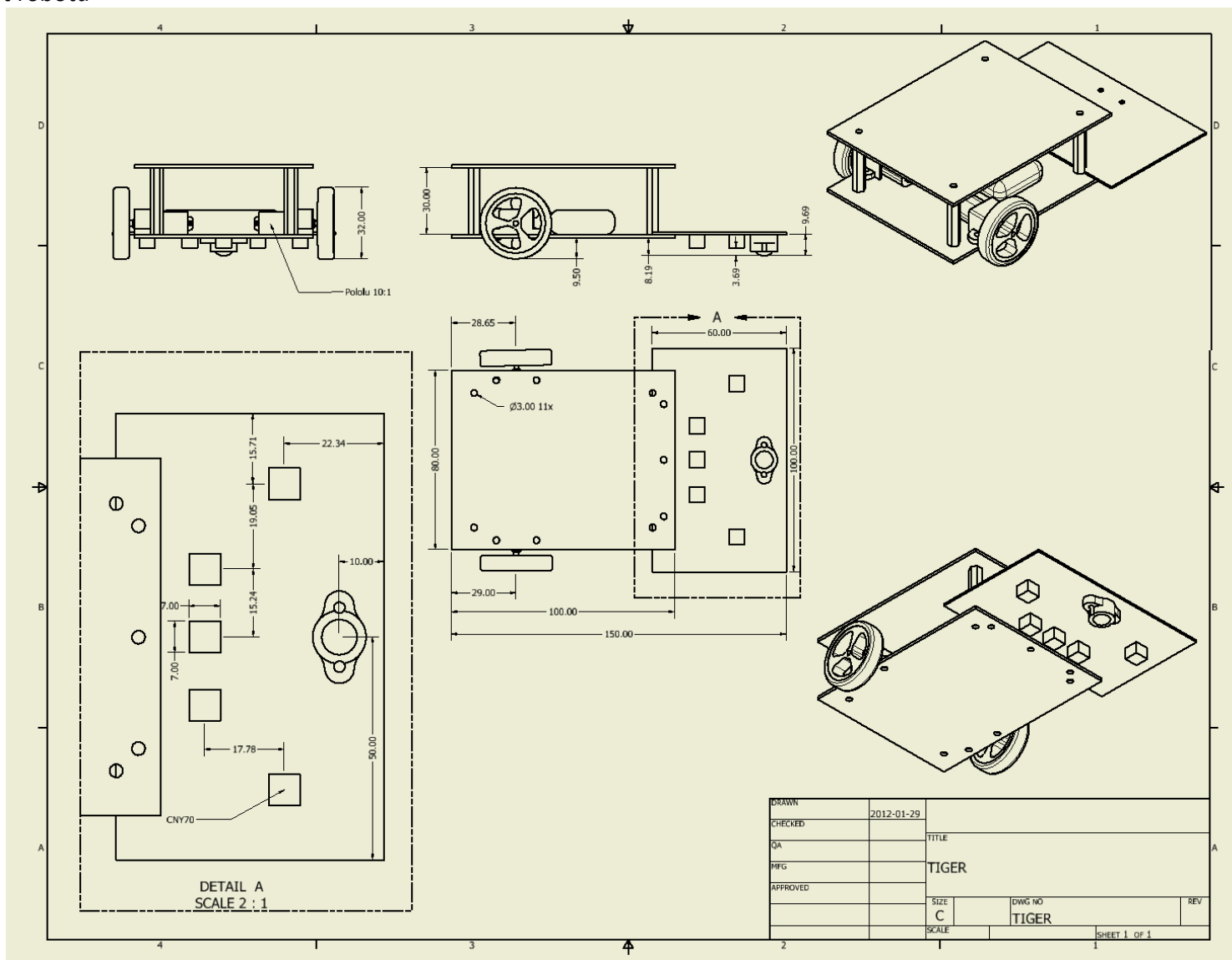
Zdjęcie 1. Robot „Tiger”



2. Konstrukcja robota

Robot jest złożony z płytki aluminiowej (na dole) o wymiarach około 8x10 cm. Równolegle do niej zamocowaliśmy płytkę PCB z elektroniką o takich samych wymiarach. Utrzymujemy ją na czterech plastikowych słupkach dystansowych mocowanych na śruby. Z przodu do aluminium przytwierdziliśmy płytkę z czujnikami. Połączyliśmy ją z główną płytką luźnymi kabelkami, co okazało się bardzo dużym błędem, bo przewody te są bardzo wrażliwe i często się urywają.

Rysunek 1. Schemat robota



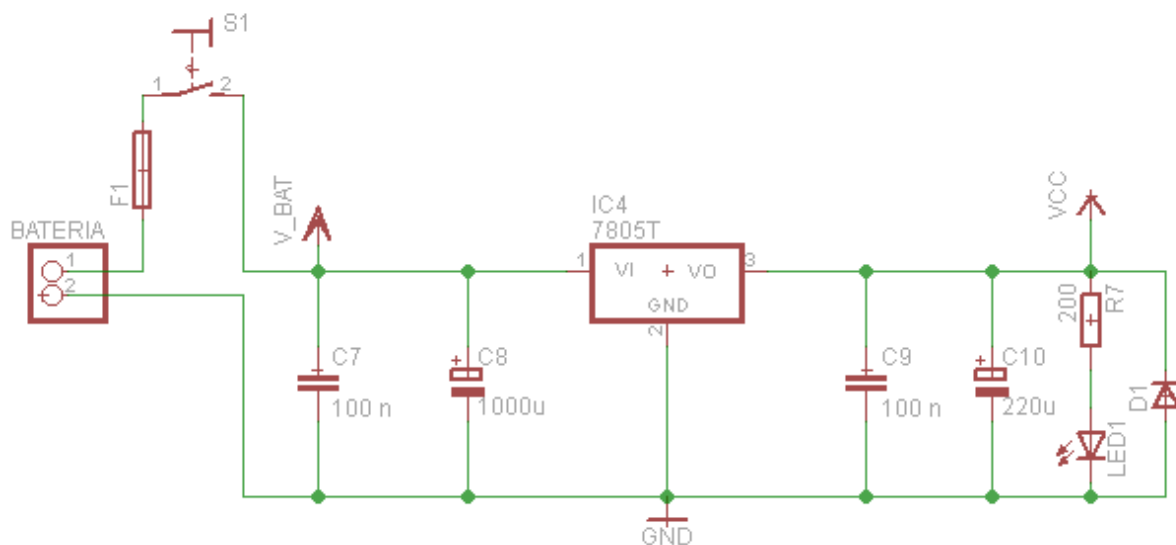
3. Zasilanie

Źródłem zasilania robota jest dwu-ogniowa bateria litowo-polimerowa o napięciu 7,4V i pojemności 500 mAh. Mimo niewielkiej pojemności robot po jej naładowaniu może nieprzerwanie jeździć przez ponad 20 minut. Bateria jest podłączana przez złącze goldpin. Do zasilania elektroniki użyliśmy stabilizatora obniżającego napięcie do 5V, natomiast silniki zasilamy pełnym napięciem z baterii. Zastosowaliśmy czerwoną diodę LED wraz z rezystorem 220Ω ograniczającym płynący prąd, która świeci się przy włączonym zasilaniu.

Kondensatory filtrujące napięcie to:

- Elektrolit 1000μF i kondensator 100nF przed stabilizatorem,
- Elektrolit 220μF i kondensator 100nF za stabilizatorem.

Rysunek 2. Schemat zasilania



4. Mikrokontroler i elementy elektroniczne

Mózgiem robota jest 8-bitowy mikrokontroler ATmega8AU w obudowie TQFP. Pracuje przy zasilaniu 5V. Programujemy go szeregowo programatorem ISP, do czego potrzebne jest złącze IDC10 (KANDA). Posiada wbudowany ośmio-kanalowy 10-bitowy przetwornik analogowo-cyfrowy, który wykorzystujemy do zbierania informacji z czujników czarnej linii.

Do obniżenia napięcia z baterii do 5V użyliśmy stabilizatora 78S05 w obudowie TO-220 pozwalającego uzyskać na wyjściu prąd do 2A. Niewielki spadek napięcia powoduje, iż układ się nie grzeje. Użycie radiatora nie było więc konieczne.

Silnikami sterują dwa dwukanałowe mostki H typu TB6612 – po jednym na silnik, co pozwala uzyskać większą wydajność prądową (do 2A). Napięcie do części sterującej wynosi 5V. Na precyzyjne sterowanie prędkością obrotową silników pozwalają kanały PWM działające w przypadku naszego robota z częstotliwością 1kHz. Bardzo niski spadek napięcia na mostku sprawia, że jest to obecnie jeden z najlepszych mostków H do tego typu zastosowań.

Na płytce umieściliśmy również 6 diod LED szeregowo z rezystorem 220Ω. Dioda czerwona zapala się tuż przed startem robota, natomiast zielone diody sygnalizują, który czujnik „widzi” linię.

Dwa przyciski typu tact switch miały służyć do startowania robota oraz włączania trybu serwisowego. Jednak przyciski przepaliły się podczas lutowania, co powoduje, że nie zawsze działały zgodnie z założeniem i zaburzały prawidłowe działanie robota. Z tego też powodu obsługa przycisków została wyłączona i obecnie nie spełniają one żadnej roli.

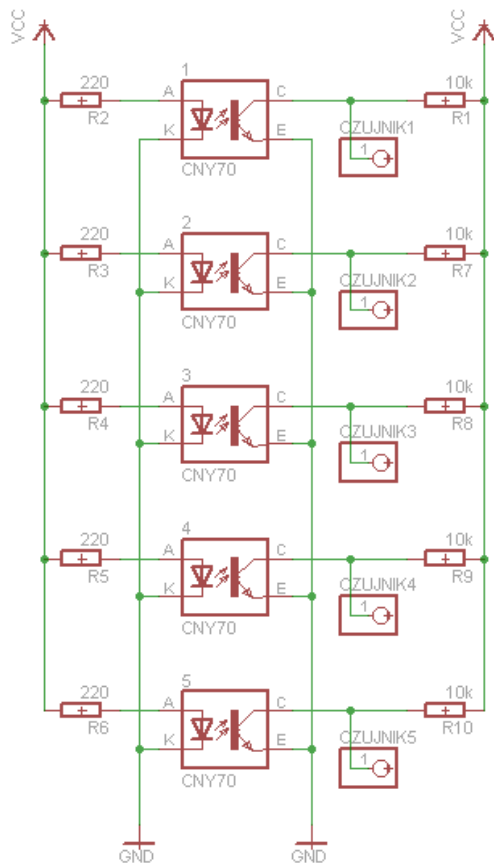
W robocie zamontowaliśmy prosty układ mierzący napięcie baterii na dzielniku napięcia. Połowa wartości napięcia jest odczytywana przez kanał ADC w mikrokontrolerze. Jednak z powodu braku trybu serwisowego nie korzystamy z tej możliwości.

5. Czujniki czarnej linii

Jako czujniki czarnej linii wykorzystaliśmy przewlekany transoptor odbiciowy CNY70 składający się z podczerwonej nadawczej diody LED i fototranzystora. Prąd diody został ograniczony rezystorem 220 Ω , a prąd tranzystora dla uzyskania wymaganej czułości rezystorem 10k Ω .

Płytki czujników została połączona z główną płytką robota luźnymi kabelkami, co nie jest najlepszym rozwiązaniem ze względu na nieestetyczny wygląd i niewygodę w użyciu – podczas łapania pojazdu bardzo łatwo można te kabelki oderwać.

Rysunek 3. Schemat płytki z czujnikami



6. Napęd

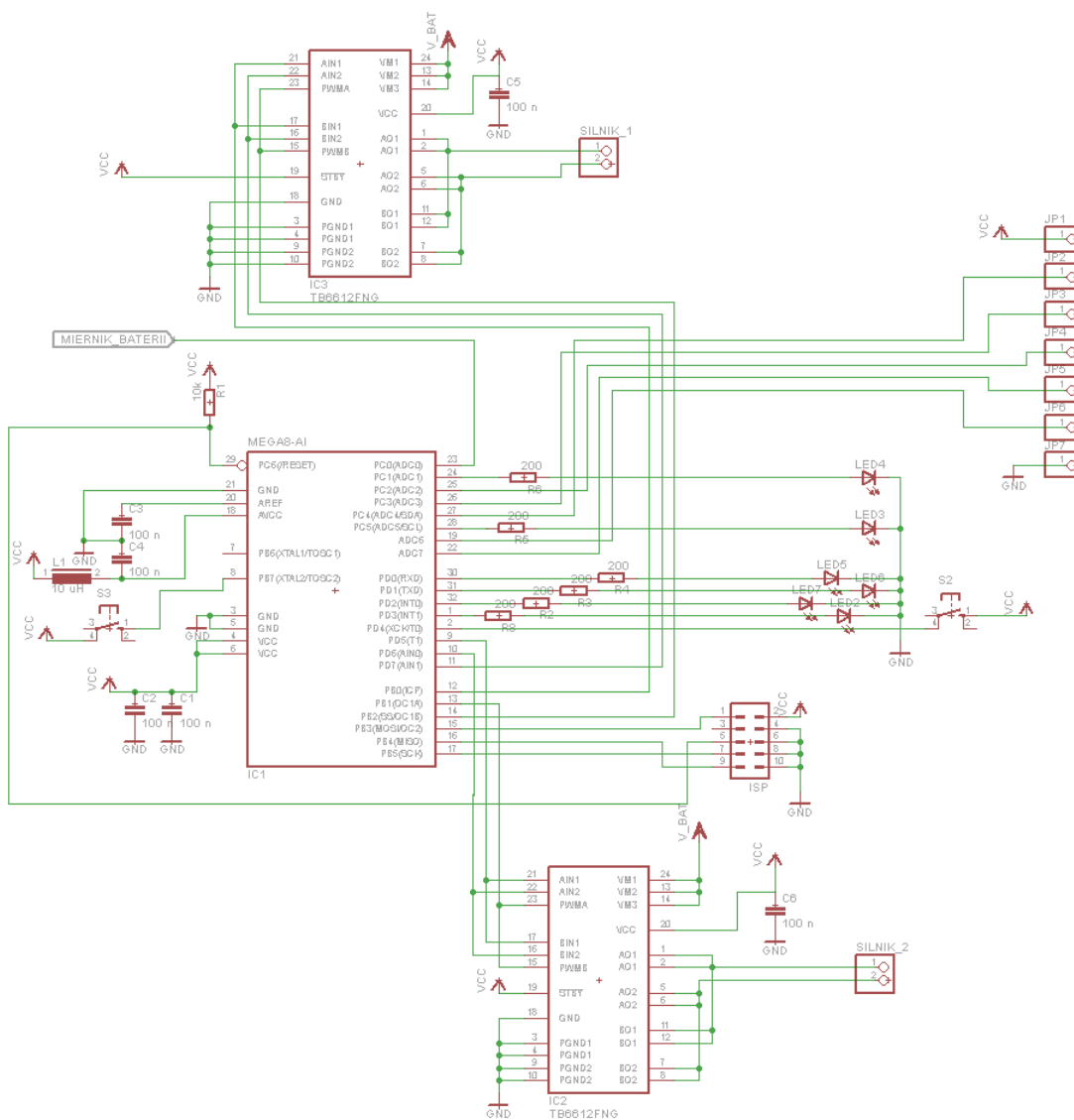
Jako silniki wykorzystaliśmy popularne i dobre napędy Pololu z wbudowaną przekładnią 10:1. Mają one wystarczająco dużą moc, żeby robot mógł ruszyć, więc nie musieliśmy stosować większych przełożeń. Wykorzystaliśmy koła z Pololu 32x7mm, które dobrze się spisują w naszej konstrukcji, choć przy wysokich prędkościach występują lekkie poślizgi.

Jako podporę wykorzystaliśmy plastikową obrotową kulkę zamocowaną przy płytce z czujnikami.

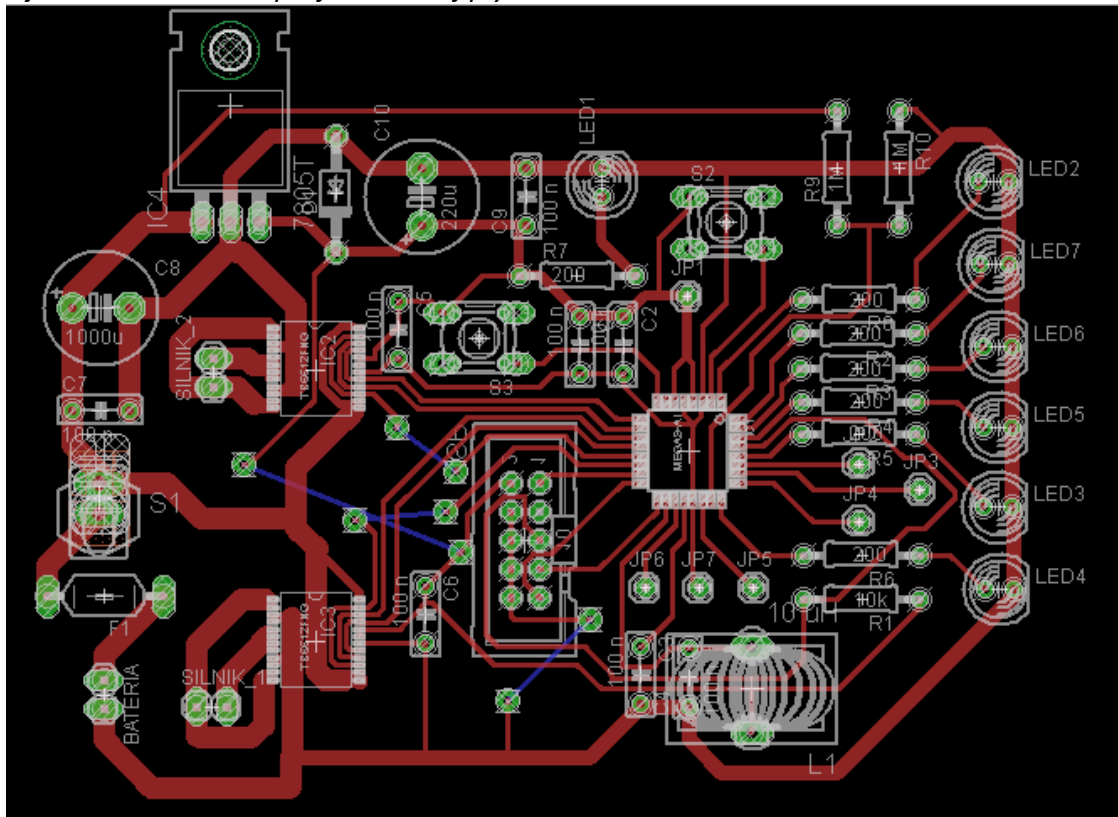
Prędkość obrotową regulujemy za pomocą mostków H i wypełnienia PWM generowanego w mikrokontrolerze.

7. Schemat elektroniczny i płytka PCB

Rysunek 4. Schemat elektroniczny robota



Rysunek 5. Widok zaprojektowanej płytki PCB

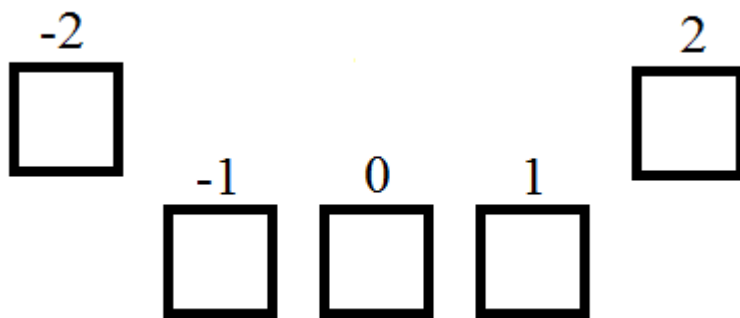


8. Algorytm sterujący robotem

Program do robota został napisany w całości w języku C.

Początkowo w robocie zaimplementowaliśmy algorytm z regulatorem PID, gdzie błąd stanowiły informacje z czujników. Początkowo główną rolę odgrywał człon proporcjonalny, a pozostałe miały przynajmniej dwukrotnie mniejszą wagę. W takiej wersji robot wystąpił na zawodach Robotic Arena 2011. Nie była to jeszcze dobra wersja i widać było, że należy coś w niej zmienić. Po eksperymentach odrzuciliśmy człon całkujący, który odpowiada za jak najmniejszy uchyb, co w tak krętych trasach jakie obecnie stosuje się na zawodach ma raczej marginalne znaczenie. Znaczne zwiększenie wagi członu różniczkującego (kilkukrotnie większa niż proporcjonalnego) pozwoliło na bardzo dokładne trzymanie się linii w szerokich i średnich zakrętach. Niestety robot z samego algorytmu PD nie potrafi pokonać zakrętów o małym promieniu oraz kątów prostych bez gubienia linii. W tym celu zastosowaliśmy dodatkową instrukcję dla sytuacji, gdy robot nie widzi linii – zapamiętuje który skrajny czujnik jako ostatni znajdował się nad linią. Jeżeli był to lewy – należy obrócić się w lewo, jeżeli prawy to w prawo. Sprawdza się to bardzo dobrze, ponieważ po wyjechaniu z trasy robot bardzo szybko znajduje ją ponownie.

Rysunek 6. Przedstawia przykładowe wartości błędów dla poszczególnych czujników



Algorytm PD zaimplementowaliśmy w ten sposób, że każdy czujnik ma swoją wagę, która została dobrana eksperymentalnie. Rysunek powyżej przedstawia jedynie samą ideę. Czujniki po lewej stronie dostały wagi ujemne, po prawej dodatnie, a środkowy zerową. Program sprawdza, które czujniki widzą linię, sumuje ich wagi, a na koniec dzieli przez ilość tych, które wzięliśmy pod uwagę. Do członu proporcjonalnego wykorzystujemy po prostu obecną obliczoną wartość z czujników. Dla członu różniczkującego od obecnej wartości odejmujemy poprzednią. Poszczególne człony mnożymy przez eksperymentalnie dobrane wagi, które trzeba zmieniać wraz ze wzrostem/spadkiem prędkości. Ustalanie prędkości obrotowej silnika wygląda następująco:

- prawy silnik: prędkość bazowa - wyliczona wartość z PD
- lewy silnik: prędkość bazowa + wyliczona wartość z PD

Między skrajnymi i środkowymi czujnikami jest duża przestrzeń, w której bez problemu można zmieścić jeszcze jeden czujnik. Dlatego właśnie w tym miejscu zaimplementowaliśmy w programie dwa „wirtualne czujniki”.

Instrukcja ta uaktywnia się, gdy:

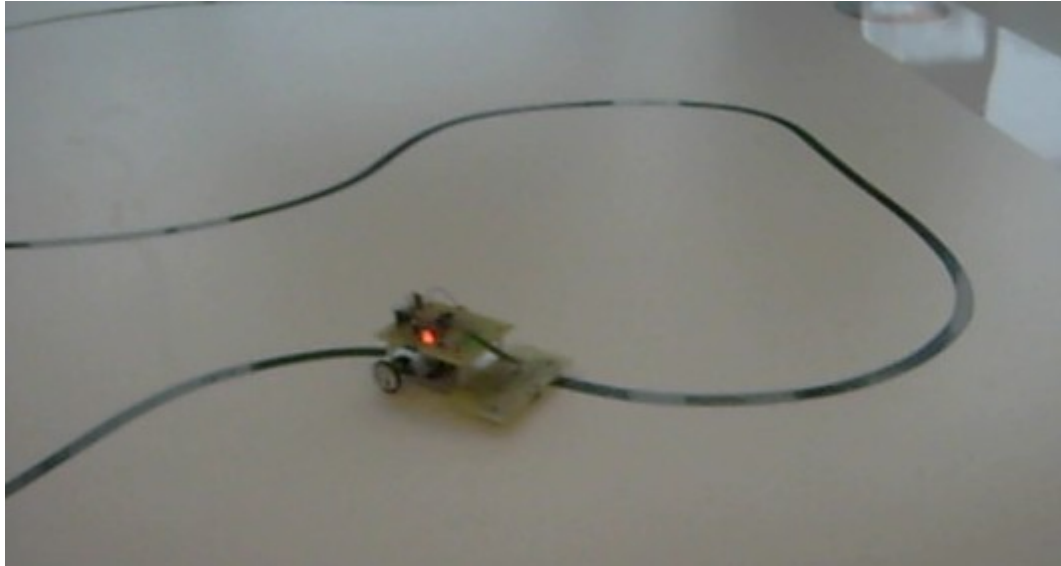
- podczas ostatniego pomiaru linia była widziana na czujniku drugim lub czwartym,
- w ciągu kilku ostatnich cykli skrajne czujniki jej nie widziały,
- w obecnym pomiarze robot w ogóle nie znajduje linii

Jeżeli powyższe warunki są spełnione, to znaczy, że linia znajduje się pod obszarem „wirtualnego czujnika”, któremu dodaliśmy odpowiednią wagę.

9. Zawody Robotic Arena 2011

„Tiger” na zawodach Robotic Arena 2011 zajął 16 miejsce spośród 31 robotów, które przejechały przynajmniej jedną z dwóch tras przygotowanych przez organizatorów. Uzyskał czas 34,50 s.

Zdjęcia 2. Przejazd pierwszej trasy



10. Uwagi końcowe

Robot „Tiger” daje możliwości do przetestowania algorytmów do pokonywania wyznaczonej trasy. Konstrukcja oczywiście nie jest doskonała i przede wszystkim posiada za mało czujników, które naszym zdaniem powinny być ułożone dokładnie w linii. Ponadto nie warto używać luźnych kabli do połączeń między płytkami – lepiej wykorzystać złącza IDC i przewody wstążkowe wielożyłowe.