



KoNaR

KOŁO NAUKOWE ROBOTYKÓW

Linefollower "Pantera"

Paweł Jakubowski
Michał Zasłona

KOŁO NAUKOWE ROBOTYKÓW KoNaR
WWW.KONAR.PWR.WROC.PL
22 LUTEGO 2013

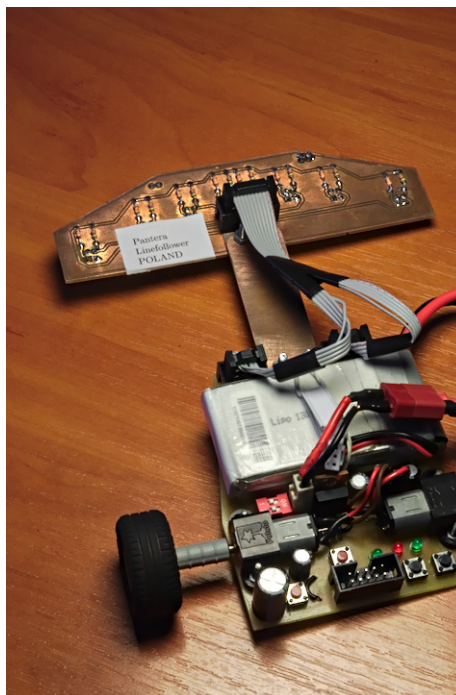
Spis treści

1	Wstęp	2
2	Założenia	2
3	Projektowanie	3
4	Układ zasilania	3
5	Układ napędowy	4
6	Czujniki	4
7	Komunikacja z robotem	5
8	Oprogramowanie	6
9	Wnioski	6

1 Wstęp

Linefollower Pantera to robot skonstruowany na potrzeby warsztatów organizowanych przez koło naukowe KoNaR i jest wynikiem współpracy dwóch studentów drugiego roku z wydziału Elektroniki.

Naszym głównym celem było stworzenie własnego, w pełni działającego robota oraz spróbowanie swoich sił na zawodach Robotic Arena 2012. Od pomysłu do realizacji minęły dwa miesiące, jednak przez ten cały okres mogliśmy korzystać z dużej wiedzy i doświadczenia starszych kolegów prowadzących warsztaty oraz liczyć na pomoc ze strony znajomych. Nasze zamierzenia zostały osiągnięte z nawiązką – pomimo braku doświadczenia w tworzeniu robotów, zdołaliśmy wystartować w zawodach, a nasza Pantera „wskoczyła” na 11. pozycję w kategorii linefollower.



Rysunek 1: LF Pantera

2 Założenia

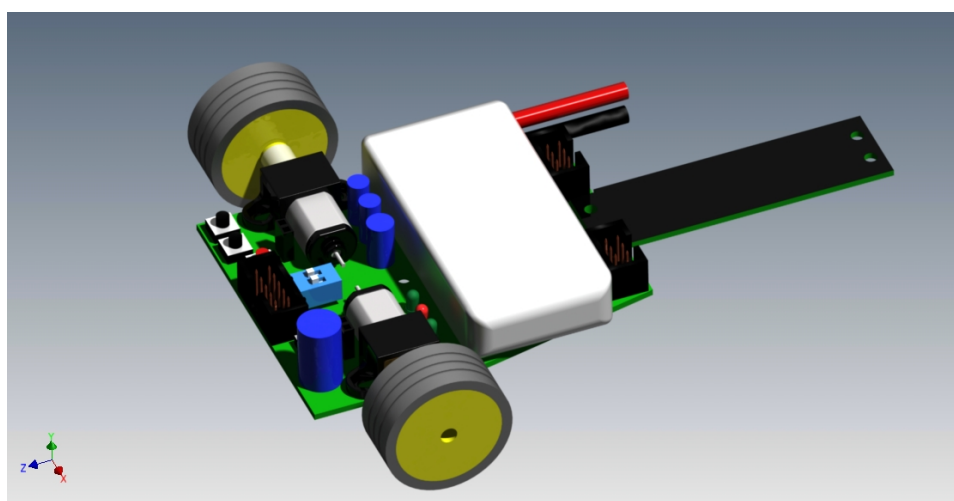
Ponieważ zdecydowaliśmy się na budowę robota typu linefollower, jedynym ograniczeniem ze strony regulaminu zawodów był wymiar robota - miał on zmieścić się na kartce formatu A4. Oprócz tego zdecydowaliśmy się uściślić nasze założenia, aby łatwiej było wykonać projekt.

Naszym głównym celem była możliwie najprostsza konstrukcja robota. O ile strona mechaniczna nie sprawiała kłopotów, o tyle często zastanawialiśmy się nad dodatkowymi układami (typu czytywanie parametrów za pomocą USART czy startowanie robota za pomocą pilota). Jako, że nasza wiedza z elektroniki była wtedy na podstawowym poziomie, dobrze, że uniknęliśmy niepotrzebnego skomplikowania, które prawdopodobnie uniemożliwiłoby ukończenie robota na czas. Ważnym czynnikiem, uwzględnionym przez nas przy projekcie była masa Pantery. Stwierdziliśmy, że niska masa zapewni nam lepsze osiągi na torach, a umieszczenie jej możliwie nisko i blisko osi kół zapobiegnie nadmiernej bezwładności. Dodatkowo koszt robota miał być relatywnie niski.

Cały układ był sterowany za pomocą popularnego mikrokontrolera ATmega8A w obudowie TQFP.

3 Projektowanie

Do projektowania wykorzystaliśmy oprogramowanie Autodesk Inventor oraz Altium Designer. Projekt rozpoczęliśmy od stworzenia strony mechanicznej układu, a więc jego modelu w Inventorze. Dopiero gdy ustaliliśmy zakładane wymiary rozpoczęliśmy pracę nad stworzeniem schematu, a następnie projektu płytki PCB.



Rysunek 2: Model robota w programie Autodesk Inventor 2013

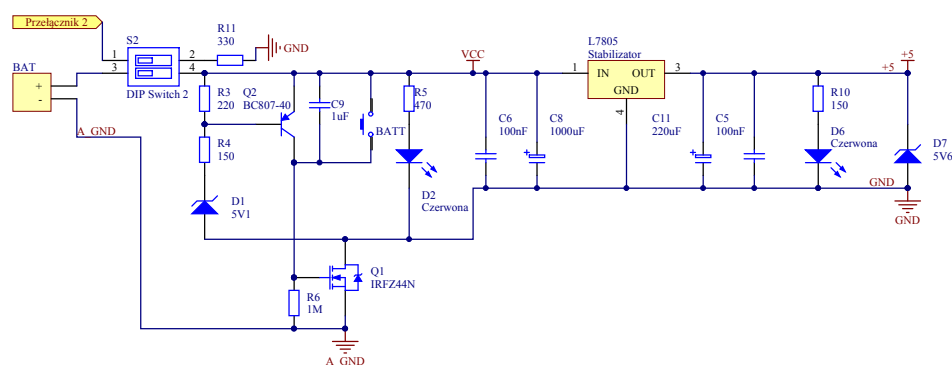
W Panterze zastosowaliśmy nieczęsto spotykane rozwiązanie w postaci umieszczenia elementów elektronicznych na spodzie robota. To, razem z wyborem większości części w obudowie SMD, pozwoliło na zmniejszenie rozmiarów płytki. Dzięki temu rozwiązaniu na górnej części robota oprócz kondensatorów elektrolitycznych pozostały jedynie elementy mechaniczne, bateria i gniazdo do podłączenia baterii, oraz elementy niezbędne do sterowania robotem.

4 Układ zasilania

Jako źródło zasilania zdecydowaliśmy się wykorzystać akumulator litowo-polimerowy o pojemności 1350mAh. Wybór ten wynikał z możliwości pozyczenia go od naszego kolegi z KoNaRu (a więc znacząco zmniejszył koszt konstrukcji). Zdawaliśmy sobie przy tym sprawę z nadmiernej wagi (ok. 80g) i wymiarów akumulatora, zdecydowały jednak ograniczone fundusze. Jednak

podczas zawodów udało nam się pożyczyć mniejszą i lżejszą (ok. 26g) baterię o pojemności 300mAh, która w znaczący sposób przyczyniła się do poprawienia czasów na torach (na bardziej skomplikowanym torze różnica wynosiła nawet 5s!).

Do zasilania silników wykorzystaliśmy bezpośredni prąd z baterii, natomiast część cyfrową poprzedziliśmy stabilizatorem L7805. Całość dodatkowo miała być zabezpieczona układem opracowanym przez Adama Pykę, jednak przez błędy w trakcie projektowania płytki zabezpieczenie nie działało poprawnie. Ponieważ na początku nie zdawaliśmy sobie z tego sprawy udało nam się spalić dwa mostki TB6612. Po tych problemach zdecydowaliśmy się na mechaniczne zabezpieczenie przed odwrotną polaryzacją w postaci zwykłego podwójnego goldpina z kluczem.



Rysunek 3: Schemat zasilania Pantery

5 Układ napędowy

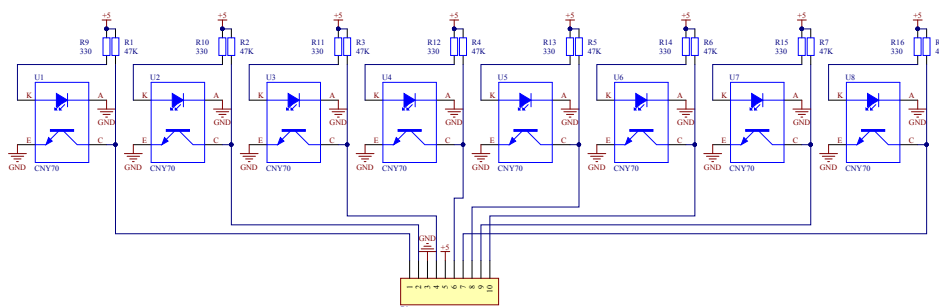
Panterę napędzały dwa silniki Pololu 30:1 z obustronną osią. Podobnie jak akumulator wybraliśmy te silniki ze względu na możliwość pożyczania ich i tym samym odciążenia naszych portfeli. Jako koła wykorzystaliśmy stare koła z LEGO Technic o średnicy 30mm. Niezbędne również były mocowania kół na osiach silników, które wykonaliśmy z części LEGO. Stworzyliśmy je z podwójnego gniazda i dwóch "krzyżaków". Jeden z nich został przyklejony, przecięty na pół i przewiercony 3mm wiertłem. W tak stworzony otwór oś od silnika weszła na wcisk, bez żadnych luzów. Silniki sterowane były pojedynczym mostkiem TB6612.

6 Czujniki

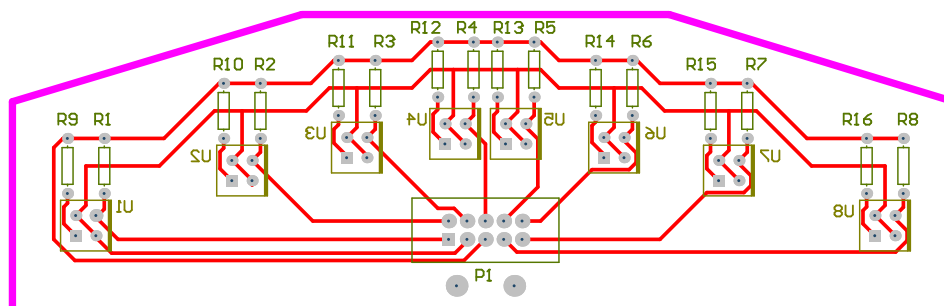
Jako czujniki wykorzystaliśmy popularne transoptory CNY70. W naszym robocie znalazło się ich osiem sztuk, ze względu na taką właśnie ilość

portów przetwornika analogowo-cyfrowego w mikrokontrolerze. Sama płytką została stworzona na wzór płytki z "Bizonka".

Jeśli chodzi zaś o rozmieszczenie czujników, to zdecydowaliśmy, aby dwa środkowe czujniki mogły widzieć linię jednocześnie, żeby szybciej wychwytywać odchyłki. Skrajne czujniki były za to wyraźniej dalej i niżej od innych, ponieważ w założeniu miały służyć do wykrywania kątów prostych.



Rysunek 4: Schemat płytki z czujnikami



Rysunek 5: Projekt płytki z czujnikami

7 Komunikacja z robotem

Cała komunikacja z robotem odbywała się za pomocą jednego podwójnego przełącznika, oraz pięciu przycisków. Jeden przełącznik odpowiadał za załączenie zasilania, natomiast drugi za ustawienie trybu serwisowego. Dwa z pięciu przycisków były zamieszczone w celu innym niż komunikacja z linefollowerem (jeden do zabezpieczenia li-pola, natomiast drugi do resetu mikrokontrolera), a pozostałe, podłączone pod piny ATmegi, odpowiadały za ustawienie robota.

Naszym "wyświetlaczem" były trzy diody LED (zielona-czerwona-zielona), które informowały o stanie Pantery. I tak np. mruganie pierwszej (zielonej) diody informowało o gotowości do jazdy (wyłączony tryb serwisowy), mru-

ganie pierwszej diody i zapalenie się drugiej informowało o ustawieniu dokładnego trybu jazdy (maksymalna prędkość robota była teraz ustawiona na połowę jego maksymalnej prędkości) itd.

8 Oprogramowanie

Przy pisaniu programu do linefollowera opieraliśmy się głównie na jednym z artykułów na forbot.pl. Program napisaliśmy w języku C. Do sterowania silnikami wykorzystany został tryb fast PWM, a obsługa przycisków została zawarta w przerwaniach. Czujniki były czytywane bezpośrednio przez porty ADC i podczas włączenia robota została uruchomiona autokalibracja.

Jak zostało wspomniane, zdecydowaliśmy się na dwa tryby pracy robota - tryb jazdy i tryb serwisowy. Dostęp do tych dwóch trybów zrealizowaliśmy jako dwie osobne pętle, które program wybiera na początku swojego działania. Ponieważ poza jazdą prędkość przetwarzania danych nie była dla nas istotna nie zajmowaliśmy się optymalizacją. Rozwiązaliśmy problem ograniczenia ilości instrukcji podczas jazdy zastosowaniem osobnej pętli po starcie robota i w czasie jej działania nie są sprawdzane żadne przyciski, a jedynie tryby jazdy oraz dane niezbędne do sterowania. Zatrzymanie robota następowało jedynie poprzez reset mikrokontrolera.

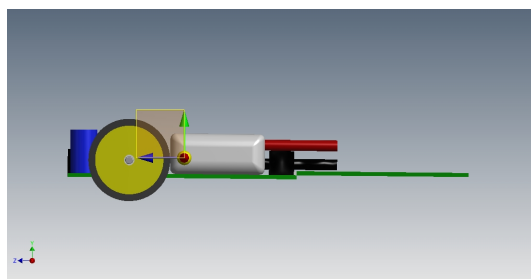
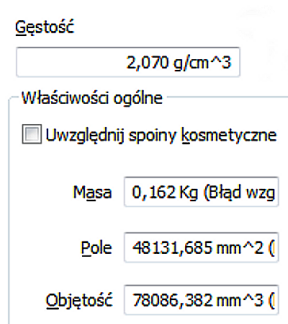
W programie zawarliśmy dwa tryby jazdy - jeden normalny, a drugi powolny (nazywany przez nas dokładnym), gdzie brana była połowa wyniku sterowania dla trybu normalnego. Przełączanie między nimi było możliwe przed jazdą za pomocą jednego z przycisków. Tryb serwisowy umożliwiał testowanie silników oraz każdego z czujników.

9 Wnioski

Podsumowując, tworzenie Pantery było prostsze, niż z początku sądziliśmy, jednak w trakcie konstrukcji natrafiliśmy również na parę problemów, których się nie spodziewaliśmy.

Nowością były dla nas programy do projektowania. I tak jak Inventor okazał się zaskakująco prosty, tak Altium sprawił parę niespodzianek. Tworzenie schematu było bardzo proste i intuicyjne, ale nie mogliśmy ruszyć z projektem płytki, dopóki nie znaleźliśmy kursu na myelectronics.pl. Mnożość opcji przytłoczyła nas i ciężko nam było zacząć pracę z tym programem. Z drugiej strony korzystanie z Inventora było samą przyjemnością. Przydatną i zaskakująco dokładną opcją była możliwość wyliczenia masy dla naszego modelu (program wyliczył 162g, bez płyty z czujnikami i przedłużek do kół, natomiast waga faktyczna wyniosła 183g). Warto wspomnieć, że nie przyłożyliśmy się do szczegółowego określania materiałów - wszystkie plastiki oznaczyliśmy jako plastik ABS, elementy metalowe jako stal - prawdopodobnie, gdybyśmy poświęcili temu więcej uwagi szacowane wyliczenia

byłyby jeszcze dokładniejsze.



Rysunek 6: Masa obliczona w programie

Rysunek 7: Wyznaczony środek ciężkości

Przy programowaniu musieliśmy się zmierzyć z różnymi wartościami optymalnych parametrów regulatora PD w różnych warunkach. Z tego względu, za każdym razem tworzyliśmy inne wersje programu, które testowaliśmy w dniu zawodów - tylko jedna okazała się być właściwa, inne zazwyczaj gubiły tor dla normalnego trybu jazdy. Bardzo przydatną opcją okazał się spowolniony tryb jazdy robota. Umożliwiło to szybkie dostosowanie robota, gdy trasa okazała się zbyt wymagająca.

Mieliśmy ciekawe spostrzeżenia dotyczące kół. Koła LEGO okazały się wyjątkowo przyczepne (choć nie robiliśmy dokładnych badań, widoczna była przewaga nad kołami Pololu). Ważną sprawą okazało się dokładne czyszczenie kół z brudu. Różnica między brudnymi a czystymi kołami wynosiła nawet do 20% (dla przykładu - robot na czystych kołach pokonał trasę z czasem 5,54s, natomiast na zabrudzonych 6,78s). Brud przyczepiający się do kół zwiększał boczny poślizg, a przez to częstsze wypadanie z toru.

Oprócz tego, było parę spraw projektowych, które zmienilibyśmy przy następnej konstrukcji. Zastosowane przez nas połączenie między główną płytą robota, a płytką z czujnikami nie do końca nas zadowalało. Powodem były zbyt duże luzy, przez co pomimo dokręcenia śrub płyta z czujnikami zmieniała swoje położenie względem płyty głównej (zwłaszcza po uderzeniu). Lepszym rozwiązaniem byłoby stworzenie jednej płyty bez żadnych łączeń. Poza tym mankamentem było również kiepskie ułożenie dipswitcha. Znajdowało się ono w miejscu, w którym ciężko go było dosięgnąć. Sam przełącznik też z pewnością byśmy wymienili, ponieważ jest on przystosowany do pracy z niższymi napięciami i do załączania zasilania lepiej zastosować pojedynczy przełącznik, który może pracować przy napięciu 12V. Z pewnością również zastosowalibyśmy od razu gniazdo, które mechanicznie zabezpiecza przed odwrotnym podłączeniem baterii.

Pozytywną rzeczą z jaką się spotkaliśmy były elementy w obudowie SMD. Kiedy decydowaliśmy, że większość z nich będzie w takiej właśnie obudowie

obawialiśmy się problemów przy lutowaniu. Okazało się, że mieliśmy więcej problemów z lutowaniem elementów THT (nadmienię tu jeszcze, że to była nasza pierwsza płytka, a mostek TB6612, który jest przestawiany jako koszmar do lutowania, udało nam się przylutować za pomocą zwykłej kolbówki). W następnym projekcie także będziemy dążyć, by jak najwięcej elementów było montowanych powierzchniowo. Prawdopodobnie zadamy również o odpowiednie zabezpieczenie ścieżek. Zauważyliśmy, że przez ułożenie ścieżek na spodzie płytki, a także przez nie zabezpieczenie ich soldermaską, często zdarzało się przypadkowe włączenie przycisku, przez zwarcie ścieżek palcem (powodowało to niechciane starty robota, czy nieplanowane tryby jazdy).

Podsumowując, robota tworzyliśmy przez mniej więcej dwa miesiące, z czego około miesiąc na projektowanie, trzy tygodnie na konstrukcję i tydzień na program. Udało nam się zminimalizować koszty do ok. 120zł (wliczając cynę, spalone mostki, wiertła, taśmy do tras i śrubki). *Pantera* spełniła z namiarą nasze oczekiwania, czyli wystartowanie w *Robotic Arena 2012*, zajmując przy tym 11. miejsce.