



KoNaR

KOŁO NAUKOWE ROBOTYKÓW

LINEFOLLOWER „TORPEDA”

MAGDALENA KACZOROWSKA

KOŁO NAUKOWE ROBOTYKÓW KoNaR

WWW.KONAR.PWR.EDU.PL

6 STYCZNIA 2016

Spis treści

1	Wstęp	2
2	Rozwinięcie	2
2.1	Mechanika	3
2.2	Elektronika	4
2.2.1	Zasilanie	5
2.2.2	Mikrokontroler	5
2.2.3	Czujniki	7
2.2.4	Sterowanie silnikami	8
2.2.5	Interfejs komunikacyjny	8
2.3	Program	8
2.3.1	Konfiguracja peryferii	9
2.3.2	Algorytm sterowania	9
3	Podsumowanie	9
4	Materiały źródłowe	10



Rysunek 1: LF Torpeda

1 Wstęp

Torpeda to robot mobilny klasy linefollower, który powstał w ramach warsztatów rekrutacyjnych do Koła Naukowego Robotyków KoNaR. Jego twórcą jest Magdalena Kaczorowska, a opiekunem projektu Dawid Śliwa. Ze względu na brak doświadczenia celem było skonstruowanie działającego robota, który weźmie udział w zawodach Robotic Arena 2015. Tworzenie robota trwało teoretycznie 2 miesiące, jednak pierwsze praktyczne prace, jak schemat, zaczęły powstawać miesiąc przed zawodami.

2 Rozwinięcie

2.1 **Mechanika**

Jednym z założeń był podział na moduły. Konstrukcja miała być jak najprostsza - moduł zawierający czujniki połączony listwą węglową z modułem zawierającym całą resztę. Silniki, akumulator oraz elektronika znajdują się na górze płytki głównej, natomiast na płytce z czujnikami cała elektronika znajduje się na spodzie. Ze względu na prostotę konstrukcji nie projektowano mechaniki robota w żadnym programie. Do napędu robota wykorzystano dwa silniki Pololu HPCB z przekładnią 30:1, ponieważ są wytrzymałe oraz łatwo dostępne. Uznano, że przekładnia 30:1 będzie wystarczająca. Zastosowano standardowe mocowania tej samej firmy oraz koła Solarbotics, ponieważ mają one dobrą przyczepność.

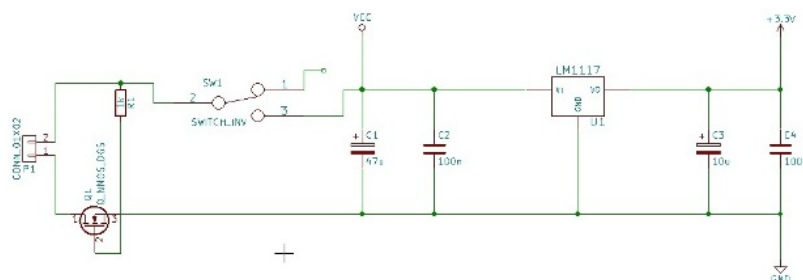
Podsumowanie Mimo braku projektowania mechaniki robot zachowuje się całkiem dobrze, jednak można znaleźć parę poprawek. Akumulator znajduje się bardziej z lewej strony, a lepiej by było gdyby znajdował się na środku. Błędem przy projektowaniu PCB było to, że nie zaznaczono otworów na śruby. Na skutek tego wywiercone otwory nie są idealnie symetryczne i przykładowo jeden silnik jest ok. 2mm bardziej wysunięty niż drugi, jednak nie przeszkadza to w poprawnym funkcjonowaniu robota. Użyta listwa ma około 4mm grubości i jest to za dużo. Podsumowując - zgodnie z założeniami konstrukcja mechaniczna jest bardzo prosta - można zrobić parę poprawek, jednak są one drobne.



Rysunek 2: Robot podczas składania

2.2 Elektronika

Zostały stworzone 2 płytki PCB - obie zaprojektowane w KiCadzie ze względu na pracowanie z tym programem podczas warsztatów rekrutacyjnych. Wykonane zostały metodą fototransferu. Ze względu na problemy z wytrawianiem lub za późne zdiagnozowanie problemu były naświetlane oraz wytrawiane 2 razy. Obydwie są dwustronne, jednak nie posiadają wielu przelotek.



Rysunek 3: Schemat zasilania

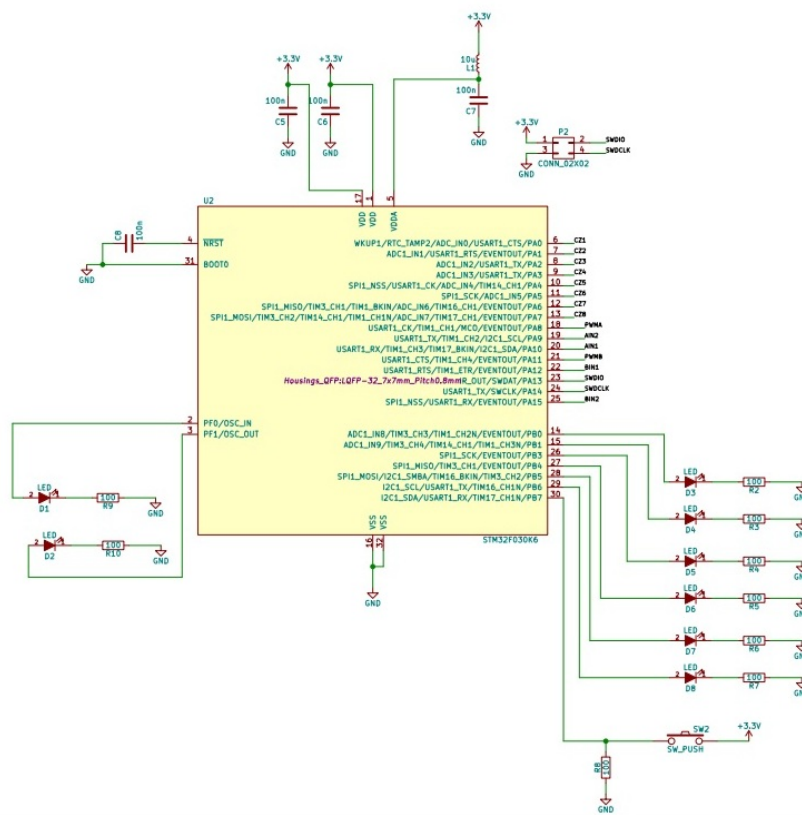
2.2.1 Zasilanie

Robot jest zasilany przez akumulator Li-Pol o pojemności 300mAh i napięciu 7,4V. Ma on niewielką masę, co jest ważne, aby uzyskać większą prędkość, a starczy na kilka-kilkanaście przejazdów. Silniki zasilane są bezpośrednio z baterii, natomiast elektronika robota zasilana jest napięciem stabilizowanym 3.3V. Całość posiada podwójne zabezpieczenie przed odwrotną polaryzacją baterii w postaci tranzystora MOSFET typu N oraz mechanicznego zabezpieczenia - gniazda z kluczem. Układ posiada włącznik zasilania w postaci zworki. O podłączonym zasilaniu informuje świecąca się dioda.

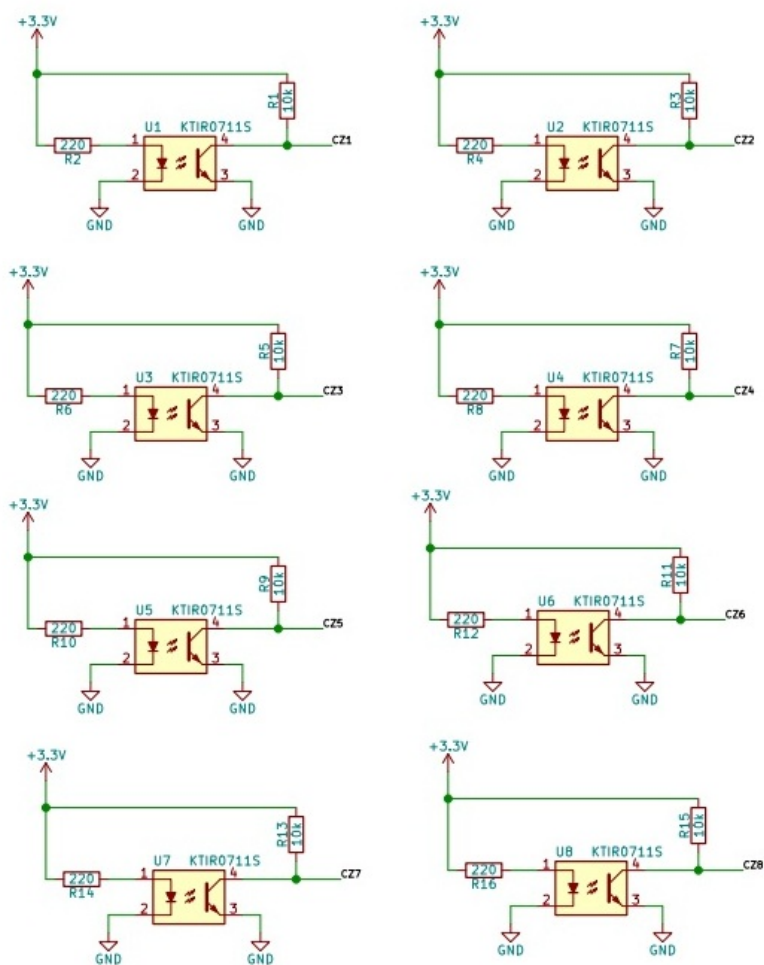
2.2.2 Mikrokontroler

Początkowo robot miał być sterowany przez jeden z mikrokontrolerów z rodziny AVR ze względu na dużą ilość materiałów w internecie na ich temat. Jednak na warsztatach omawiane było programowanie STM32, które są mocniejszymi układami. Ponadto posiadano płytkę STM32F3DISCOVERY zawierającą programator ST-Link. Więc finalnie wybrano mikrokontroler STM32F030K6T6 (maksymalne taktowanie 48MHz, 1 dwunastobitowy przetwornik ADC, 5 szesnastobitowych timerów). Był to bardzo dobry wybór, ponieważ spełnia wszystkie wymagania, posiada wystarczającą ilość wejść przetwornika analogowo-cyfrowego (dokładnie 10, a użyte zostało 8) i do tego jest tani (ok. 6zł).

Dużą zaletą przy programowaniu były programy pomocnicze - *STM32CubeMX* przy konfiguracji oraz *STMStudio* przy debugowaniu.



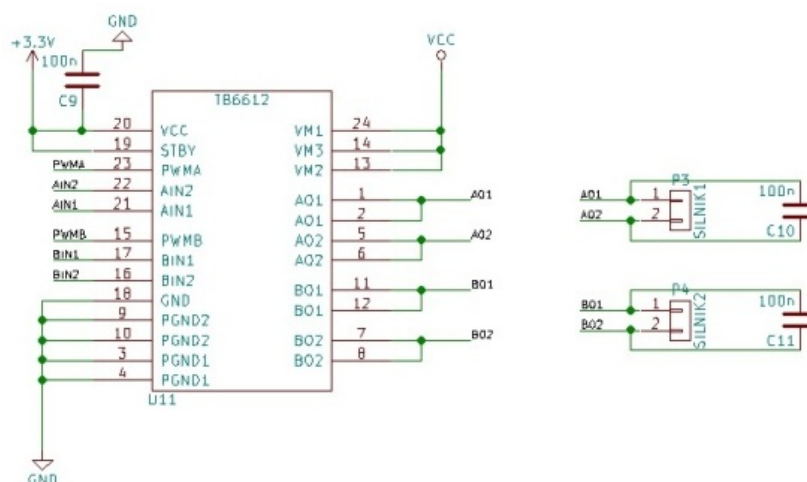
Rysunek 4: Schemat podłączenia mikrokontrolera



Rysunek 5: Schemat podłączenia czujników

2.2.3 Czujniki

Robot zawiera 8 czujników KTIR0711s (maksymalne napięcie diody IR: 5 V, maksymalny prąd diody IR: 50 mA, maksymalne napięcie kolektor-emiter: 30 V, maksymalny prąd kolektora: 20 mA). Służą one do wykrywania czarnej linii. Zostały one rozmieszczone w półkolu, a skrajne czujniki znajdują się wyraźnie niżej niż inne, aby łatwiej wykrywać kąty proste. Odczyt z czujników odbywa się za pomocą przetwornika ADC.



Rysunek 6: Schemat podłączenia mostka

2.2.4 Sterowanie silnikami

Silniki sterowane były jednym mostkiem TB6612 (maksymalne napięcie zasilania silników: 4,5 - 15 V, prąd ciągły 1,2 A, prąd chwilowy: 3,2 A). Silniki zasilane są prosto z baterii, żeby miały większą moc.

2.2.5 Interfejs komunikacyjny

Do komunikacji z robotem wykorzystano diody, przycisk oraz włącznik zasilania w postaci zworki. Podłączenie zasilania sygnalizuje jedna dioda, następnie pierwsze użycie przycisku to kalibracja czujników, co sygnalizuje kolejną diodą. Drugie użycie przycisku to start robota, co sygnalizuje miganie kolejnej diody. Zatrzymanie robota następuje przy wyłączeniu zasilania.

Programowanie odbywa się poprzez połączenie z programatorem ST-Link za pomocą 4 pinów - 3.3V GND SWDIO SWDCLK.

Zrezygnowano ze zdalnej komunikacji ze względu na to, że projekt byłby wtedy bardziej skomplikowany i prawdopodobnie nie starczyłoby czasu na jego ukończenie do Robotic Areny.

2.3 Program

Program został napisany w języku C. Korzystano z darmowego środowiska *System Workbench for STM32* oraz *STM32CubeMX*.

2.3.1 Konfiguracja peryferii

Obsługa przycisku zawarta została w przerwaniach. Podstawą czasu jest przerwanie od timera wykonujące się z częstotliwością 1000Hz. Do korzystania z ADC wykorzystano DMA. Do sterowania silnikami wykorzystano sygnał PWM.

2.3.2 Algorytm sterowania

Algorytm jazdy oparty jest na regulatorze PD. Nastawy dobierane są doświadczalnie. Przy wypadnięciu poza trasę, robot wykorzystuje informację, który czujnik jako ostatni widział linię i wraca na trasę, co jest wykorzystywane przy pokonywaniu kątów prostych.

3 Podsumowanie

W dokumencie opisano proces budowy robota mobilnego „Torpeda” klasy linefollower. Końcowym efektem pracy był działający robot, który wystartował na zawodach Robotic Arena zajmując 13. miejsce, tak więc cel został zrealizowany. Planowane jest udoskonalenie programu, żeby robot jeździł jeszcze lepiej i szybciej oraz branie udziału w zawodach.

Z powodu braku doświadczenia tworzenie projektu było bardzo czasochłonne, jednak wyniesiono z niego bardzo dużo oraz zdobyto cenne doświadczenie.



Rysunek 7: LF Torpeda

4 Materiały źródłowe

- warsztaty rekrutacyjne do Koła Naukowego Robotyków KoNaR
- forum oraz artykuły - www.forbot.pl
- raporty oraz prezentacje - www.konar.pwr.wroc.pl