



KoNaR

KOŁO NAUKOWE ROBOTYKÓW

LINEFOLLOWER MACZETA

ALEKSANDER BOJDA
PIOTR STOPYRA

KOŁO NAUKOWE ROBOTYKÓW KONAR

WWW.KONAR.PWR.EDU.PL

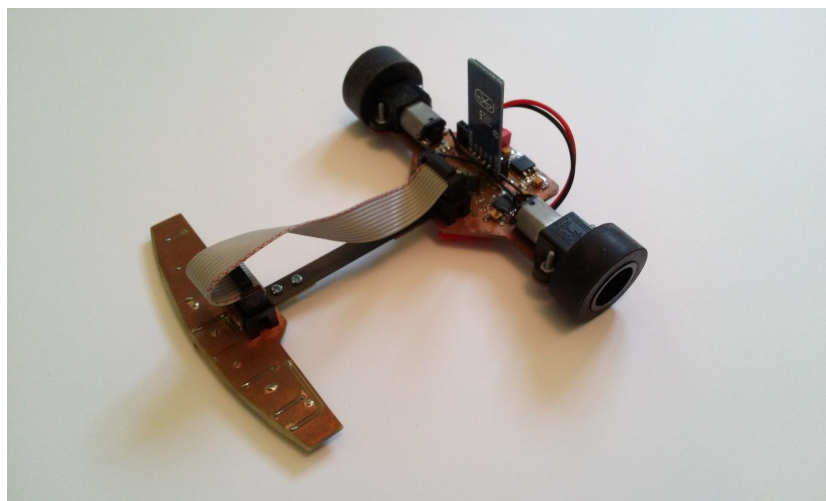
6 STYCZNIA 2016

Spis treści

1	Wstęp	2
2	Rozwinięcie	2
2.1	Mechanika	2
2.2	Elektronika	3
2.2.1	Zasilanie	3
2.2.2	Mikrokontroler	4
2.2.3	Czujniki	4
2.2.4	Sterowanie silnikami	5
2.2.5	Interfejs komunikacyjny	5
2.3	Program	5
2.3.1	Konfiguracja peryferii	6
2.3.2	Algorytm sterowania	6
3	Podsumowanie	6
4	Materiały źródłowe	7

1 Wstęp

Niniejszy dokument zawiera informacje dotyczące budowy robota mobilnego klasy Linefollower tworzego w ramach warsztatów robotycznych organizowanych przez KNR KoNaR. Pierwszym etapem pracy było zaprojektowanie schematu i płytki PCB w programie *KiCad*. Następnie płytkę naświetlono i wytrawiono. Potem wywiercono otwory i wyszlifowano kształt płytek przy pomocy multiszliferki i pilników. Następnie przylutowano elementy i zaprogramowano robota korzystając z programów *STM32CubeMX* oraz *System Workbench for STM32*. Opiekunem projektu był Adam Prochownik.



Rysunek 1: Linefollower Maczeta

Celem naszej pracy było zbudowanie stosunkowo niewielkiego i zwrotnego robota klasy Linefollower i wystartowanie nim w zawodach Robotic Arena.

2 Rozwinięcie

2.1 Mechanika

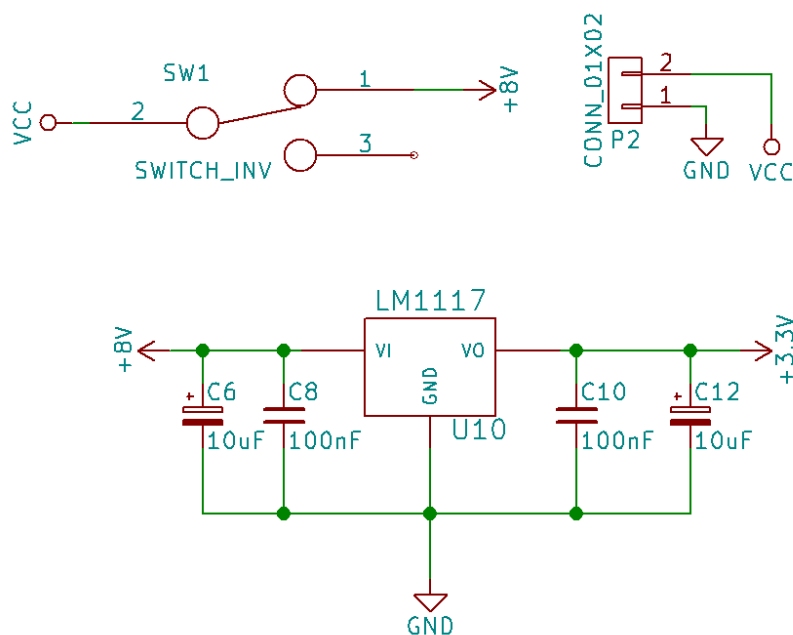
Podwozie składa się z dwóch części wykonanych z PCB połączonych listwą z włókna węglowego. Na module głównym zamocowane są silniki Pololu z przekładnią 50:1. Koła mają wymiary 31,2 x 13,2 mm. Silniki posiadają za małą prędkość obrotową i planowana jest ich wymiana na silniki z przekładnią 30:1 lub 10:1.

2.2 Elektronika

PCB zaprojektowano w programie *KiCad*. Są one dwustronne i tworzone metodą fototransferu. Zakładano minimalizację wymiarów robota, więc elementy są umieszczone blisko siebie.

2.2.1 Zasilanie

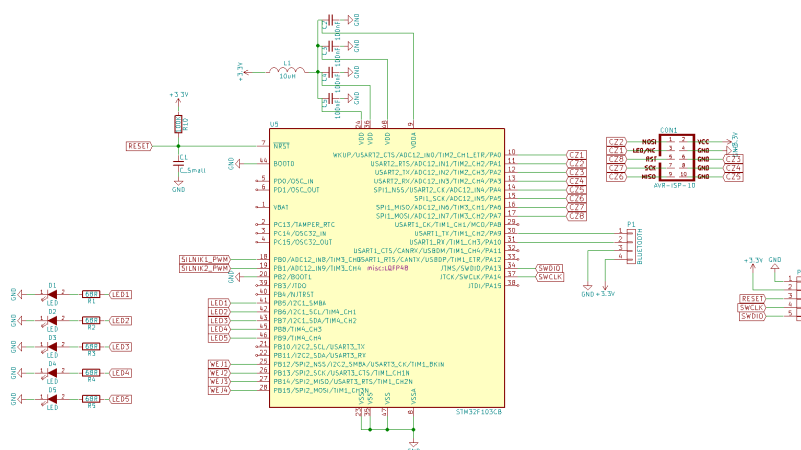
Robot jest zasilany za pomocą pakietu Li-Pol składającego się z dwóch ogniw o łącznym napięciu 7.4V Wybrano ten sposób zasilania ponieważ baterie Li-Pol charakteryzują się stosunkowo małym rozmiarem i dużą pojemnością oraz wydajnością prądową. Zasilanie można wyłączyć za pomocą przełącznika. Aby zasilić mikrokontroler, czujniki oraz moduł Bluetooth napięcie jest stabilizowane za pomocą stabilizatora LM1117 do napięcia 3.3V. Zabezpieczeniem przed odwrotną polaryzacją jest wtyczka którą można włożyć tylko w jeden, uniemożliwiający błąd sposób. Napięcie jest filtrowane za pomocą kondensatorów ceramicznych 100 nF oraz elektrolitycznych 10 μ F.



Rysunek 2: Schemat zasilania

2.2.2 Mikrokontroler

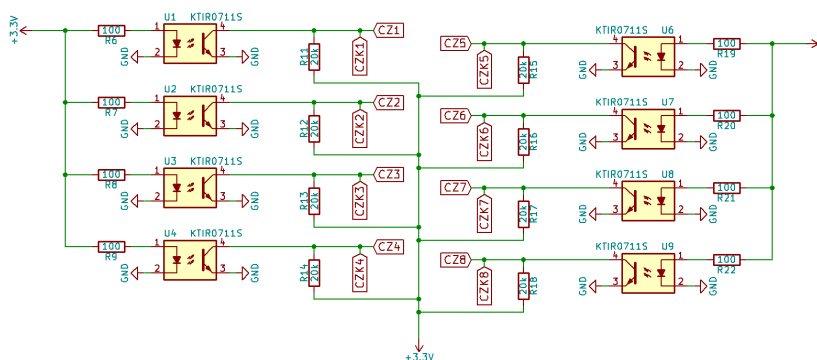
Wybrano mikrokontroler STM32F103CBT6 w obudowie LQFP48 o parametrach: maksymalna częstotliwość taktowania 72MHz, napięcie pracy 2.0-3.6V, pamięć Flash 128kB, pamięć RAM 20kB, 16 kanałów przetwornika ADC, 4 timery. Zdecydowano się na niego ze względu na wysoką częstotliwość taktowania oraz dużą ilość peryferiów. Posiada on możliwość łatwego debugowania za pomocą SWD.



Rysunek 3: Schemat podłączenia mikrokontrolera

2.2.3 Czujniki

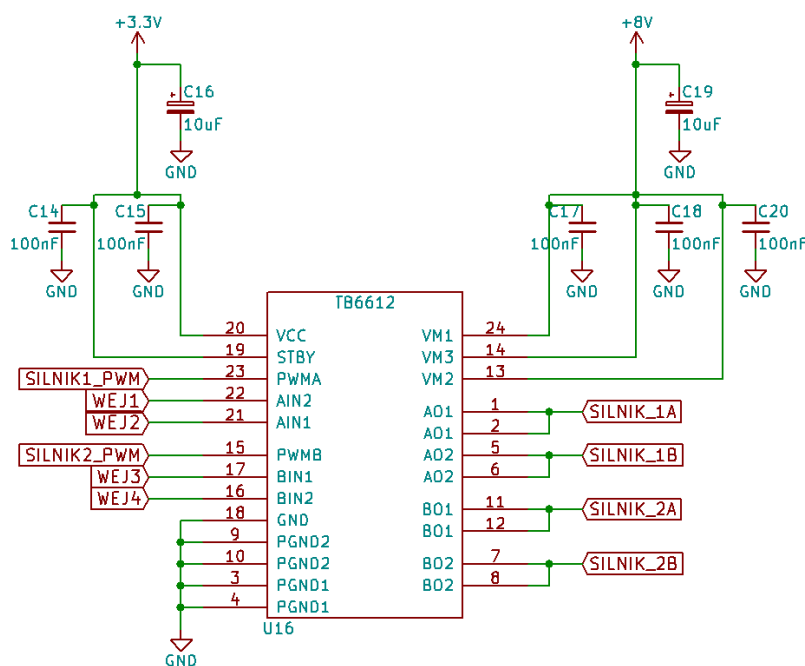
Robot jest wyposażony w 8 czujników KTIR0711S służących do wykrywania linii. Są one bezpośrednio połączone z wejściami przetwornika ADC za pomocą przewodu IDC-10.



Rysunek 4: Schemat podłączenia czujników

2.2.4 Sterowanie silnikami

Do sterowania silnikami został użyty podwójny mostek H TB6612 w obudowie SSOP24. Jego napięcie pracy wynosi 2.7V-5.5V, napięcie na silnik 4.5V-13.5V, ciągły prąd na kanał 1.2A, chwilowy prąd na kanał 3.2A. Zdecydowano się na niego ze względu na dużą wartość prądu na kanał oraz mały spadek napięcia. Silniki zasilane są napięciem 7.4V aby uzyskać jak największą moc.



Rysunek 5: Schemat sterowania silnikami

2.2.5 Interfejs komunikacyjny

Komunikacja z robotem może zachodzić w dwojaki sposób: za pomocą programatora lub modułu bluetooth. Wyposażenie w moduł bluetooth daje możliwość zdalnej komunikacji z robotem co ułatwia jego obsługę. Robot komunikuje się z otoczeniem za pomocą pięciu diod LED. Trzy znajdują się na górnej stronie modułu głównego, a dwie na dolnej.

2.3 Program

Program jest napisany w języku C z użyciem bibliotek HALa. Pisany był w darmowym środowisku *Eclipse*. Do wygenerowania konfiguracji peryferiów

użyto darmowego programu *STM32CubeMX*. Do sprawdzenia działania programu został użyty darmowy *STMStudio*.

2.3.1 Konfiguracja peryferii

Użyte zostało ADC korzystające z DMA. Do regulacji obrotów silników użyto sprzętowej generacji PWM o częstotliwości taktowania 1000 Hz.

2.3.2 Algorytm sterowania

Do sterowania robota użyto regulatora PD. Człon proporcjonalny odpowiedzialny jest za natychmiastową reakcję przy zmianie pozycji robota względem linii, zaś człon różniczkujący tłumi oscylacje robota. Pierwszym etapem jest odebranie i przetworzenie wartości z czujników. Do każdego czujnika przypisana jest waga, im dalej od środka, tym większa jej wartość. Czujniki posiadają wagi z zakresu od -4 do 4. Kolejnym etapem jest obliczenie błędu, od którego wartości zależy prędkość silników. Dokonuje się to przez dodanie wag i podzielenie przez ich ilość, a następnie pomnożenie przez zadane wzmocnienie. Przy zerowym błędzie oba silniki kręcą się z tą samą prędkością, a przy różnym od zera jeden silnik posiada większą prędkość, a drugi mniejszą lub na odwrót w zależności od wartości błędu.

3 Podsumowanie

W dokumencie opisano proces budowy robota klasy Linefollower. Końcowym efektem pracy jest działający robot. Udało się zrealizować założenie jak najmniejszej wielkości robota. Elementem, który mógłby być zrealizowany lepiej jest złącze IDC-10, które zajmuje dużo miejsca na płytce, zastosowanie złącza o mniejszym rastrze pozwoliłoby na zmniejszenie konstrukcji. Planowane jest rozszerzenie aplikacji sterującej startowaniem i zatrzymywaniem się robota o kolejne funkcjonalności. Planowane są również testy części mechanicznych robota - zastosowanie silnika o mniejszej przekładni, kół odlewanych z silikonu oraz znalezienie optymalnej długości listewki łączącej czujniki z główną płytką.

Podczas pracy nad projektem nauczyliśmy się podstawowych zagadnień dotyczących pracy z mikrokontrolerami STM32 z wykorzystaniem bibliotek HALa, poznaliśmy zasady stosowane przy projektowaniu układów elektronicznych i robotycznych oraz zapoznaliśmy się z programami ułatwiającymi pracę nad przyszłymi projektami.

4 Materiały źródłowe

Literatura

- [1] Warsztaty robotyczne KoNaR 2015. <http://konar.pwr.wroc.pl/index.php/warsztaty-2015/>.
- [2] Forbot. <http://forbot.pl/blog/>.
- [3] STM32F103CBT6 Datasheet. <http://st.com/web/en/resource/technical/document/datasheet/CD00161566.pdf>.