



ROBOT MOBILNY KLASY LINE FOLLOWER QUBA

JAKUB SZCZYRK

KOŁO NAUKOWE ROBOTYKÓW KoNaR
WWW.KONAR.PWR.EDU.PL

13 STYCZNIA 2018

Spis treści

1	Wstęp	2
2	Quba	2
2.1	Mechanika	2
2.1.1	Koła	2
2.1.2	Silniki	2
2.1.3	Podsumowanie	2
2.2	Elektronika	2
2.2.1	Zasilanie	3
2.2.2	Mikrokontroler	4
2.2.3	Czujniki	5
2.2.4	Sterowanie silnikami	6
2.2.5	Interfejs komunikacyjny	6
2.2.6	Zdjęcia Robota	7
2.3	Program	8
2.3.1	Konfiguracja	8
2.3.2	Algorytm sterowania	8
3	Podsumowanie	9
4	Materiały źródłowe	9

1 Wstęp

Quba to robot klasy Line-Follower, który tworzyłem w wolnych chwilach od lipca 2017r. Na początku nie miał większego celu prócz edukacyjnego, z czasem zacząłem myśleć nad wzięciem udziału w Robotic Arena 2017. Inspiracje oraz teorie czerpałem z Internetu

2 Quba

2.1 Mechanika

Robot składa się z dwóch elementów płytki PCB, w której znajduje się najważniejsza elektronika oraz oddzielnych czujników, umieszczonych w platformie, stworzonej ze starego urządzenia. Moduły są połączone ze sobą za pomocą plastikowej części. Zakrzywienie łącznika miało na celu obciążyć koła znajdujące się w tylnej części robota.

2.1.1 Koła

Koła zostały wybrane ze względu na bieżnik, który niweluje wpływ podłoża na tarcie i przyczepność.

2.1.2 Silniki

Na module głównym znajdują się silniki Polulu 30:1. Niewielkie, lekkie a zarazem mocne silniki wykorzystane są jako jednostka napędowa. Napięcie, przy którym stosunek mocy do wytrzymałości jest najkorzystniejszy, wynosi 6 V- zastosowałem napięcie z baterii 7,4V-8,4V.

2.1.3 Podsumowanie

Jeżeli chodzi o błędy to na pewno zastosowanie tak małego rozstawu między kołami wpłynęło na zmniejszenie zwinności robota. Ponadto użycie kabli, które zapewniają połączenie między modułami, nie było korzystnym rozwiązaniem - można było stworzyć połączenie z płytki PCB z wytworzonymi ścieżkami.

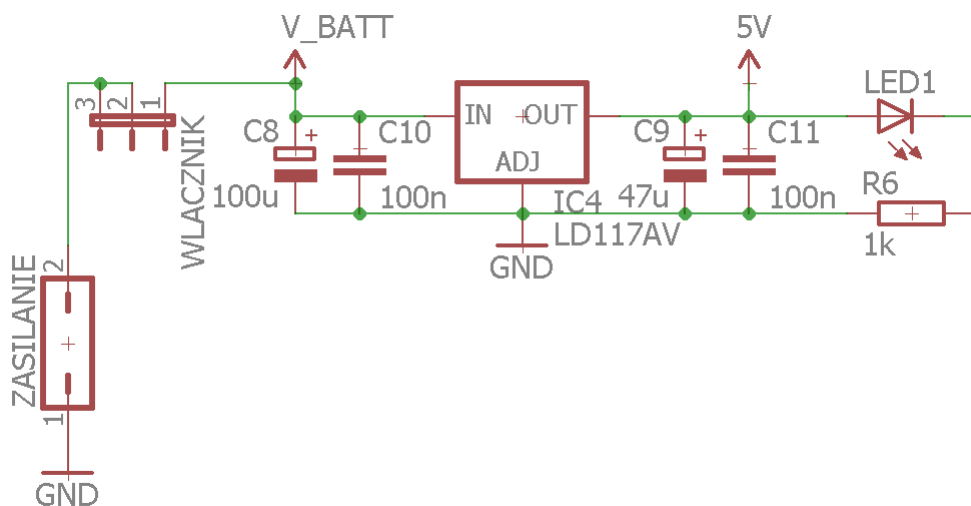
2.2 Elektronika

PCB zostało stworzone w programie Eagle. Płytką jest dwustronna i tworzone za pomocą metody termotransferu. Umożliwia ona łatwą zmianę mostków

H, microporocedora i silników. Możliwe są własne ingerencje, dzięki zwiększonemu obszarowi uziemienia.

2.2.1 Zasilanie

Robot jest zasilany za pomocą pakietu Li-Pol, składającego się z dwóch ogniw z napięciem po 3,7V. Wszystkie podzespoły pracują pod napięciem 5V, prócz silników, które są zasilane bezpośrednio z akumulatora. Zdecydowałem się na akumulator Li-Pol, ponieważ stabilizator LDO LM1117 na 5V wymaga napięcia co najmniej 6,5V, zastosowanie baterii nie miało by tu racji bytu - wybór takiego stabilizatora był związany z zastosowaniem Atmega8A. Do włączania zasilania użyłem przełącznika suwakowego. Poprawne zasilanie sygnalizuje żółty LED.

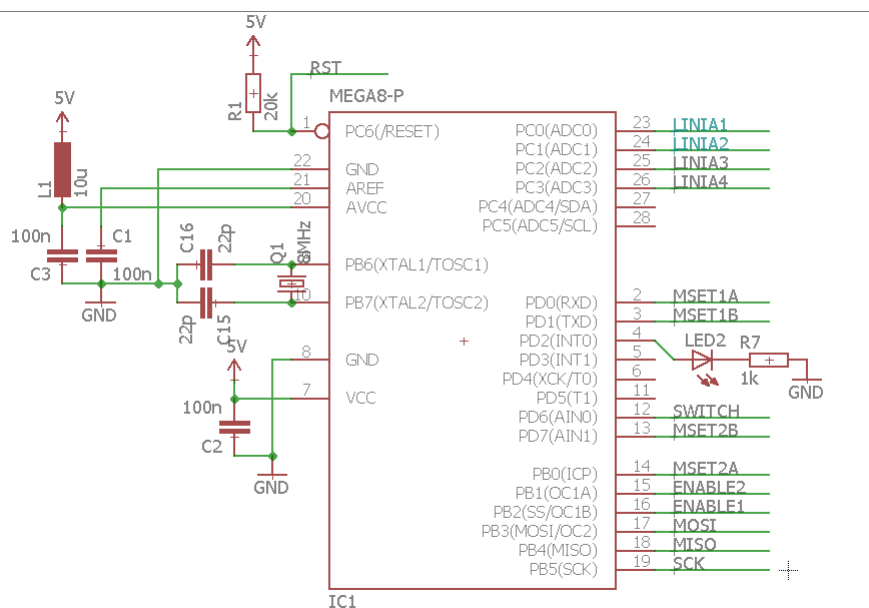


2.2.2 Mikrokontroler

Mikrokontroler AVR - ATmega8A-PU DIP

Specyfikacja:

- Zasilanie: 2,7 V - 5,5 V
- Taktowanie: do 16 MHz
- Pamięć Flash: 8 kB
- Pamięć RAM: 1 kB
- Pamięć EEPROM: 512 B
- 23 linie wyjścia/wejścia
- Dwa 8-bitowe Liczniki
- Jeden 16-bitowy licznik
- Trzy kanały PWM
- 6 kanałów 10-bitowego przetwornika analogowo-cyfrowego
- Sprzętowe interfejsy komunikacyjne: USART, SPI
- Pięć trybów uśpienia
- Obudowa przewlekana DIP28

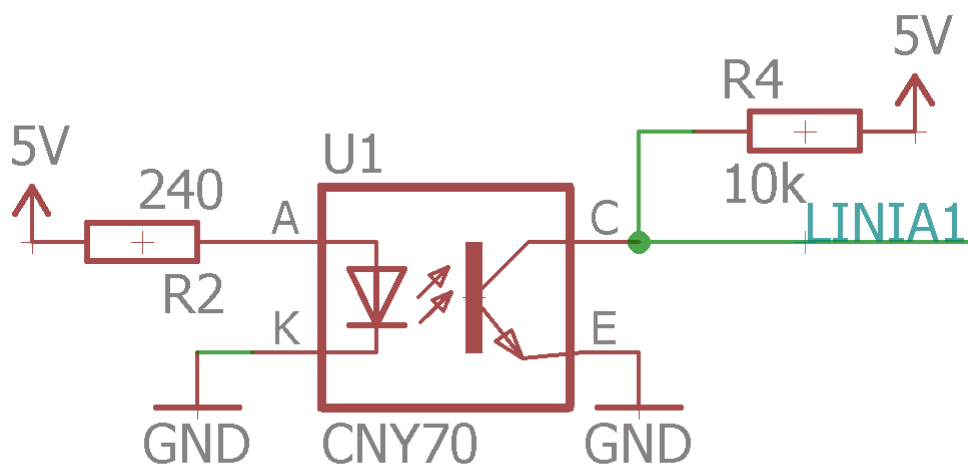


2.2.3 Czujniki

Wybrałem czujnik transoptor odbiciowy CNY70

Specyfikacja:

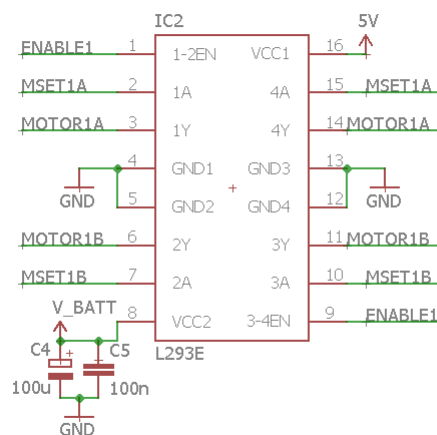
- Napięcie zasilania diody: 5 V
- Maksymalny prąd diody: 50 mA
- Maksymalne napięcie kolektor-emiter: 32 V
- Maksymalny prąd kolektora: 50 mA



2.2.4 Sterowanie silnikami

Zdecydowałem się na L293D - dwukanałowy sterownik silników 36V/0.6A
Specyfikacja:

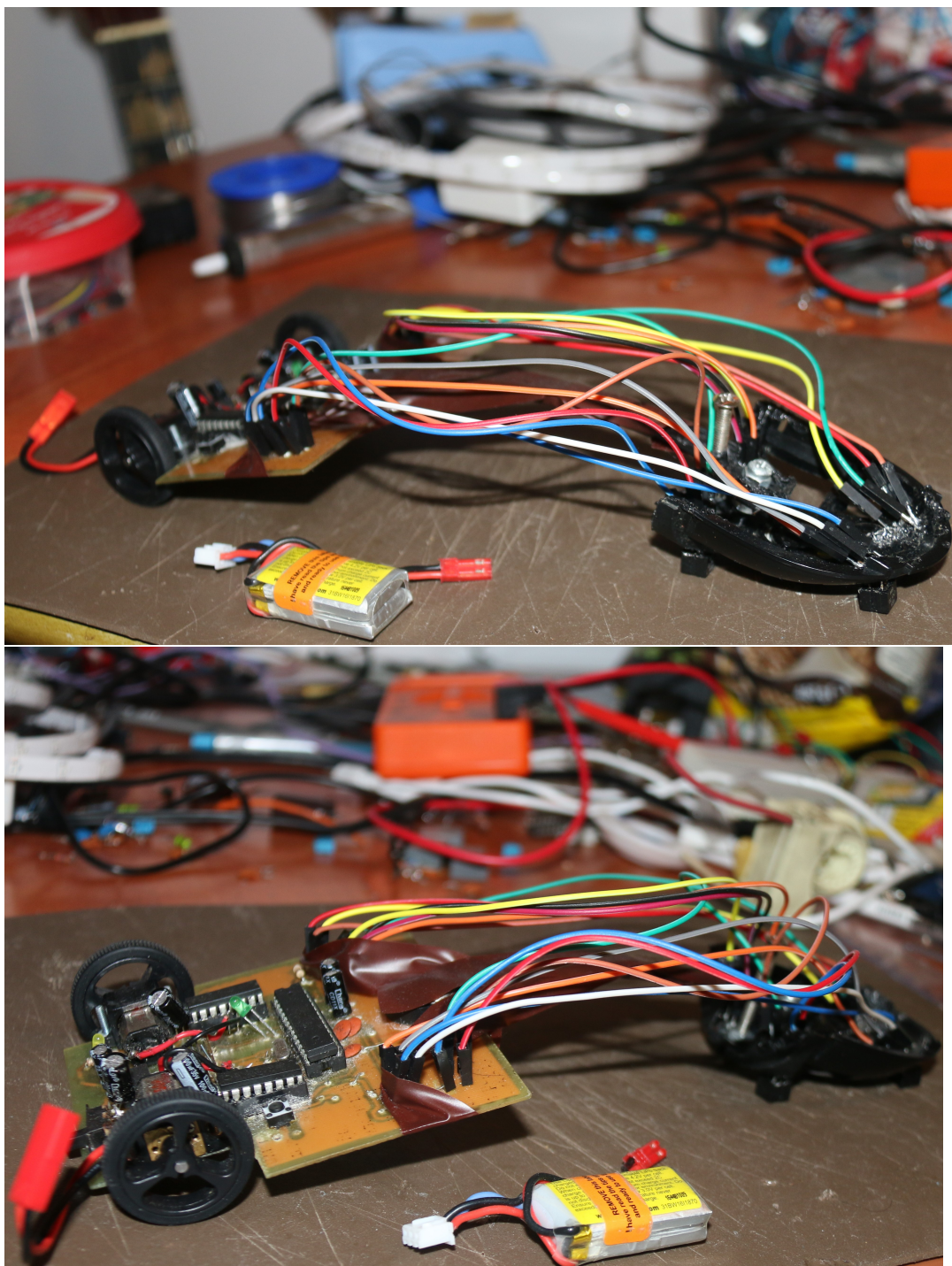
- Liczba kanałów: 2
- Maksymalne napięcie zasilania silników: 36 V
- Średni prąd kanał: 0,6 A
- Szczytowy prąd na kanał: 1,2 A
- Obudowa: DIP 16 (przewlekana)
- Wbudowane diody zabezpieczające



2.2.5 Interfejs komunikacyjny

Początkowo robota miałem programować za pomocą programatora USBasp i złącza ISP. Jednakże to nie zadziałało - prawdopodobnie sterowniki z USBasp nie chciały się wgrać poprawnie. W związku z tym wykorzystałem zastosowanie mikrokontrolera jako AVR i podłączałem go do płytki stykowej wraz z połączonym Arduino Leonardo, który miał wgrany program do kompilowania. Robot posiada dwie diody. Pierwsza dioda świecąc wskazywała na płynący prąd w obwodzie. Druga służyła do testów oraz do sygnalizacji rozpoczęcia działania. Robota można było uruchomić z przycisku, który się na nim znajdował.

2.2.6 Zdjęcia Robota



2.3 Program

Program był pisany w języku Basic. Darmowe Środowisko programistyczne obsługującego wybrany język to BASCOM-AVR. Jedynym jego ograniczeniem jest limit rozmiaru pliku wynikowego równy 4kB – nigdy nie udało mi się go przekroczyć. Kompilator służy do tworzenia plików *.hex, które są tłumaczeniem naszego kodu napisanego w języku Basic na kod zrozumiały dla mikrokontrolera. Do wgrywania programu zastosowałem tak jak wspominałem Arduino Leonardo oraz program SinaProg.

2.3.1 Konfiguracja

Do tej pory nie udało mi się poprawnie skonfigurować programu do odpowiedniego działania. W celu wykrycia czarnej linii użyłem ADC. Przy regulacji prędkości oraz obrotów, zastosowałem PWM przez timer o częstotliwości taktowania 16MHz.

2.3.2 Algorytm sterowania

Zasada działania:

1

Po wciśnięciu przycisku program zaczyna działać, a robot startuje po pięcio-krotnym zapaleniu się diody zielonej po której przechodzi do pętli wykonującej się aż do ponownego wciśnięcia przycisku.

2

Następnie sczytywane są dane z czujników

3

Gdy któryś z czujników wykryje czarną linię, następuje zmiana w zasilaniu silników odpowiednio przeciwnie do wykrytej linii.

4

Proces powtarza się z częstotliwością 100Hz

Przy niewykryciu linii oba silniki mają taką samą prędkość.

3 Podsumowanie

W dokumencie opisano proces budowy robota klasy Line-Follower. Prace nad nim trwały 5 miesięcy, z dużymi przerwami. Robot próbował swoich sił na zawodach Robotic Arena 2017. Niestety podczas transportu wystąpił problem z silnikami, który udało mi się naprawić dzięki lutownicy Michała Burdki. Jednakże nie udało mi się odpowiednio skonfigurować programu, aby robot był w stanie przejechać zadaną trasę. Podsumowując, robot dostarczył mi wiele zabawy oraz zwiększył moje umiejętności w zakresie elektroniki. Jest jednak wiele rzeczy do udoskonalenia, którymi zajmę się w najbliższej przyszłości.

Cele na przyszłość:

- ulepszenie algorytmu
- zmodyfikowanie drugiego członu z czujnikami

4 Materiały źródłowe

- <https://forbot.pl/blog/>
- <https://botland.com.pl/>
- <https://www.youtube.com/user/ElektroPrzewodnik>
- <https://www.arduino.cc/>