



---

## ROBOT MOBILNY KLASY LINE FOLLOWER MAVERICK

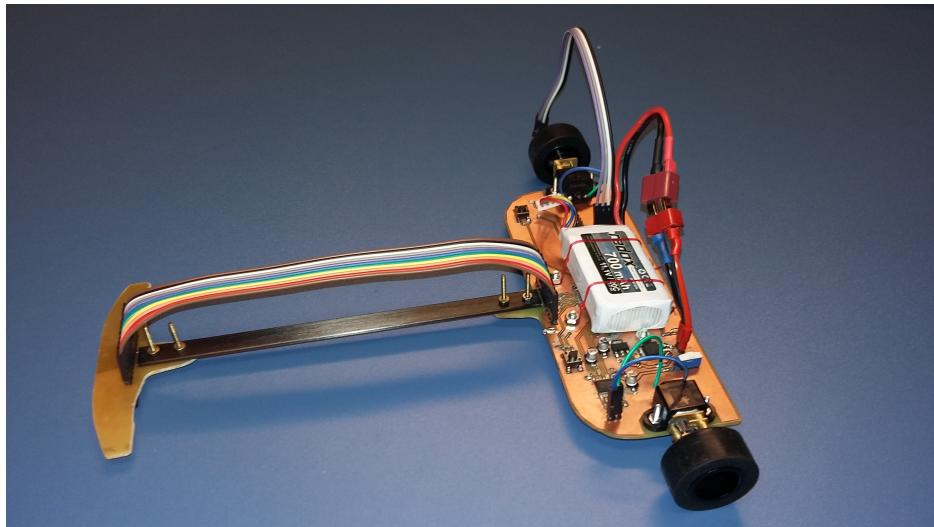
---

ŁUKASZ MICHALCZAK  
ANNA POSTAWKA

KOŁO NAUKOWE ROBOTYKÓW KoNAR  
[WWW.KONAR.PWR.EDU.PL](http://WWW.KONAR.PWR.EDU.PL)  
6 STYCZNIA 2016

## **Spis treści**

<b>1</b>	<b>Wstęp</b>	<b>2</b>
<b>2</b>	<b>Maverick</b>	<b>2</b>
2.1	Mechanika . . . . .	2
2.1.1	Listewka węglowa . . . . .	3
2.1.2	Koła . . . . .	3
2.1.3	Akumulator . . . . .	3
2.2	Elektronika . . . . .	3
2.2.1	Zasilanie . . . . .	4
2.2.2	Mikrokontroler . . . . .	4
2.2.3	Czujniki . . . . .	5
2.2.4	Sterowanie silnikami . . . . .	7
2.2.5	Interfejs komunikacyjny . . . . .	7
2.3	Program . . . . .	9
2.3.1	Konfiguracja peryferii . . . . .	9
2.3.2	Algorytm sterowania . . . . .	9
<b>3</b>	<b>Podsumowanie</b>	<b>10</b>
<b>4</b>	<b>Materiały źródłowe</b>	<b>10</b>



Rysunek 1: Robot mobilny Maverick

## 1 Wstęp

Maverick to robot klasy Line Follower Light. Został stworzony w ramach warsztatów robotycznych organizowanych w 2015 r. przez koło naukowe KoNaR działające na Politechnice Wrocławskiej. Celem było wystartowanie w zawodach Robotic Arena 2015. Opiekunem projektu został Bartek Kurosz. Był on nieocenioną pomocą i znacznie przyczynił się do sukcesu tej konstrukcji.

Głównym założeniem było po prostu stworzenie działającej maszyny, jako że był to pierwszy robot jego autorów. Ale na początku pracy, podczas intensywnego zbierania inspiracji i informacji na temat tego typu konstrukcji wykryształowała się wizja, aby stworzyć robota jak najlżejszego, bez zbędnych elementów, ale takiego, który pozwalałby na osiąganie dobrych wyników i konkurowanie z najlepszymi.

Dokument ten przedstawia poszczególne elementy budowy Mavericka oraz opisuje koncepcję zaimplementowanego programu.

## 2 Maverick

### 2.1 Mechanika

Podstawą konstrukcji robota (rysunek 1) są dwie płytki PCB. Zostały one połączone listewką węglową. Na przedniej płytce znajdują się tylko czujniki, na tylnej cała reszta elektroniki. Silniki i akumulator są przymocowane bezpośrednio do płytki.

Taka konstrukcja ma wiele zalet. Przede wszystkim jest lekka, bo nie trzeba wprowadzać dodatkowych elementów stanowiących podwozie czy ramę pojazdu. Dodatkowo można zmieniać odległość płytka od siebie. Problemem mogłoby być to, że bezpośrednie przykręcenie układu napędowego do płytki PCB może narażać elektronikę na potencjalnie szkodliwe wstrząsy, ale jak na razie nie zostały zauważone żadne negatywne skutki takiej budowy.

### 2.1.1 Listewka węglowa

Zostały przetestowane dwie długości listewki: 12 cm i 16 cm. Dłuższa sprawdziła się lepiej. W przyszłości warto by było przetestować jeszcze inne długości (max. 26 cm, aby robot mieścił się na kartce A4), aby dobrać najlepszą optymalną dla tej konstrukcji. Można by też spróbować zastosować lżejszy materiał.

### 2.1.2 Koła

Zamontowane zostały koła Solarbotics RW2i razem z oponami. Zapewniły one wystarczającą przyczepność.

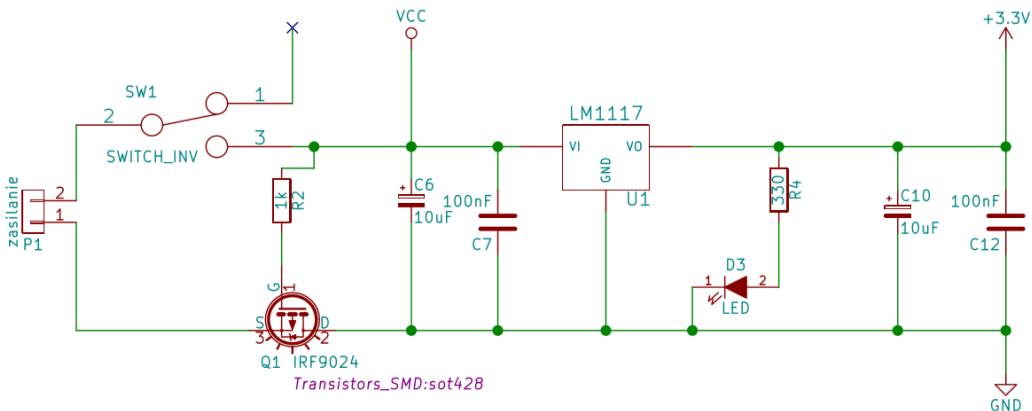
### 2.1.3 Akumulator

Akumulator został umieszczony na tylnej osi, aby dociążyć koła i zwiększyć przyczepność. Został przymocowany gumkami zawiązanymi do śrubek przykręconych do płytki PCB. Na przyszłość warto by pomyśleć nad innym mocowaniem - po pierwsze śrubki są niepotrzebnym ciężarem, po drugie aktualnie nie ma możliwości przesunięcia akumulatora, a jest to jeden z cięższych elementów konstrukcji i znacznie zmienia on wyważenie robota, co może wpływać na osiągi.

Aktualnie akumulator leży na podwyższeniu, co ma uchronić delikatne elementy znajdujące się pod nim od uszkodzenia mechanicznego. Nie jest to najlepsze rozwiązanie - dla zwiększenia przyczepności lepiej byłoby umieścić akumulator jak najniżej. Jeśli zostałaby wykonana nowsza wersja płytki, trzeba by było najpierw eksperymentalnie wyznaczyć najlepsze ustawienie akumulatora względem osi kół, po czym przeprojektować płytka tak, aby przesunąć elementy spod akumulatora w inne miejsce i może nawet wyciąć dziurę w płytce, aby ten ciężar znalazł się jak najbliżej podłoża.

## 2.2 Elektronika

Elektronika została zaprojektowana w programie KiCad. Jest to oprogramowanie typu open source przeznaczone do edycji schematów oraz obwodów



Rysunek 2: Zasilanie

drukowanych.

Płytki były przygotowywane samodzielnie, więc w projekcie unikano umiejscawiania elementów czy ścieżek zbyt blisko siebie oraz w pobliżu krawędzi. Ograniczono również ilość przelotek, gdyż ich wykonanie jest czasochłonne i wymaga dużej precyzji. Użyto głównie elementów powierzchniowo montowanych (ang. SMD, Surface Mounted Devices). Zajmują one mniej miejsca i nie potrzebują dodatkowych otworów. Zastosowano technikę fotochemiczną.

Ścieżki, przez które prąd z akumulatora płynie bezpośrednio, są szersze (mają 32 milsy). Dla najważniejszych elementów, takich jak mikrokontroler czy sterownik silników, poprowadzono osobne ścieżki zasilania. Zakłóczenia są filtrowane za pomocą kondensatorów.

### 2.2.1 Zasilanie

Do zasilania użyto akumulatora Li-pol. Podzespoły pracują pod napięciem 3,3V. Wyjątkiem są silniki, do których prąd dostarczany jest bezpośrednio z zasilacza. Do uzyskania pożądanego napięcia użyto stabilizatora LM1117S 3,3V/1A. Zastosowano zabezpieczenie przed odwrotną polaryzacją układu w postaci tranzystora typu P. Wykorzystano suwakowy przełącznik zasilania. Poprawne zasilanie sygnalizowane jest poprzez diodę LED. Na rysunku 2 przedstawiono schemat zasilania.

### 2.2.2 Mikrokontroler

Użyto mikrokontrolera STM32F100C4T6B.

Specyfikacja:

- Maksymalne taktowanie: 24MHz

- Napięcie pracy: 2,0 - 3,6V
- Pamięć Flash: 16kB
- Pamięć RAM: 4kB
- Przetwornik ADC: 10 kanałowy, 12 bitowy
- Interfejs: I2C, SPI, USART
- Liczba timerów: 6 (pięć 16 bitowych)
- Liczba wejść/wyjść: 37
- Obudowa: LQFP48

Debugowanie ułatwiają dwie diody LED (rys. 7).

Początkowo użyto mikrokontrolera STM32F100C8T6B, jednak uległ on uszkodzeniu. Powyższego mikrokontrolera użyto jako zamiennika, gdyż poprzedni był już niedostępny. Należało połączyć wyprowadzenia generujące sygnał PWM ze sterownikiem silników za pomocą izolowanych przewodów, ponieważ występowała niezgodność timerów w stosunku do pierwotnego układu.

Schemat mikrokontrolera (rys.3) oraz jego zasilanie (rys.4).

### 2.2.3 Czujniki

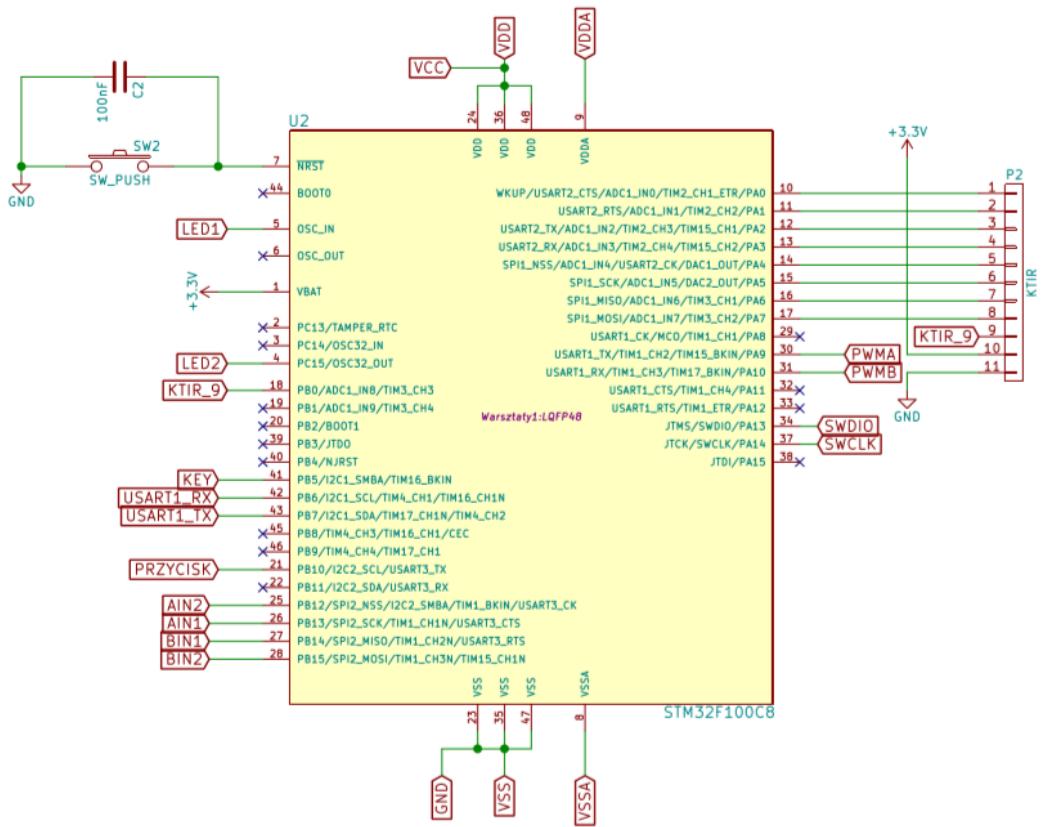
Do wykrywania czarnej linii zastosowano 9 czujników optycznych KTIR0711S.

Odczyt wykonywany jest z użyciem przetwornika ADC mikrokontrolera.

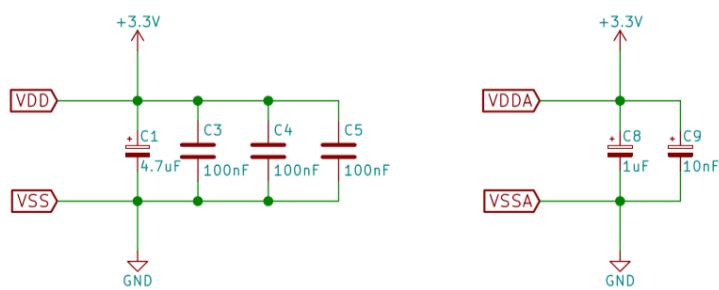
Specyfikacja:

- Napięcie diody: 5V
- Maksymalny prąd diody: 50mA
- Maksymalne napięcie kolektor-emiter: 30V
- Maksymalny prąd kolektora: 20mA
- Odległość pomiaru: max 0,5cm

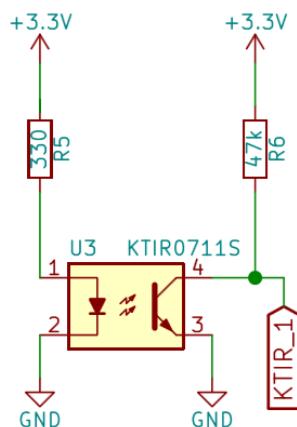
Na rysunku 5 przedstawiono schemat podłączenia czujnika optycznego.



Rysunek 3: Mikrokontroler



Rysunek 4: Zasilanie mikrokontrolera



Rysunek 5: Czujnik optyczny

#### 2.2.4 Sterowanie silnikami

Do sterowania silnikami użyto podwójnego mostka H TB6612.

Układ pozwala na proste i niezależne sterowanie dwoma silnikami DC.

Specyfikacja:

- Napięcie logika: 2,7V - 5,5V
- Napięcie silnik: 4,5V - 13,5V
- Ciągły prąd wyjścia na kanał: 1,2A
- Chwilowy prąd wyjścia na kanał: 3,2A
- Obudowa: SSOP24-szeroki - 8,3x7,6x1,6mm

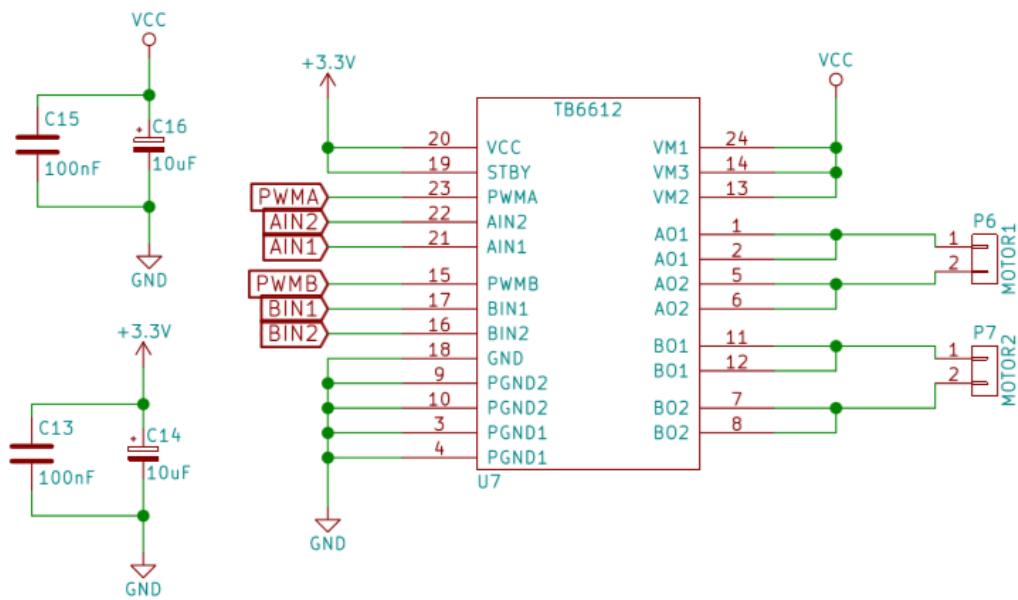
Na rysunku 6 przedstawiono schemat podłączenia sterownika silników.

#### 2.2.5 Interfejs komunikacyjny

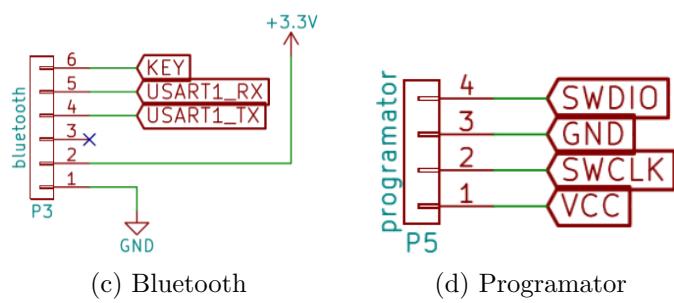
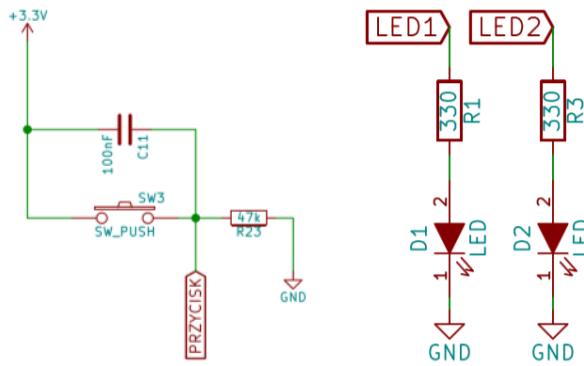
Zastosowano przycisk generujący przerwanie oraz drugi, wywołujący reset mikrokontrolera.

Programowanie dokonywane jest poprzez czteropinowe złącze SWD programatora. Wyprowadzono również złącze na moduł bluetooth. Zamontowano diodę sygnalizującą poprawne zasilanie oraz dwie diody programowalne. Ostatecznie moduł bluetooth nie został użyty ze względu na brak czasu oraz niezgodność z wyprowadzeniami złącza.

Schematy interfejsów komunikacyjnych przedstawia rysunek 7.



Rysunek 6: Sterownik silników



Rysunek 7: Interfejsy komunikacyjne

## 2.3 Program

Program napisany jest w języku C. Korzystano z darmowego, zintegrowanego środowiska programistycznego dla mikrokontrolerów STM32 i związanych z nimi płytek.

STM32CubeMX jest graficznym narzędziem konfiguracyjnym, pozwalającym na generowanie gotowego kodu w języku C dla wybranego mikrokontrolera. System Workbench for STM32 to środowisko programistyczne oparte na oprogramowaniu Eclipse, które wspiera pełną gamę mikrokontrolerów STM32. Pozwala na programowanie i debugowanie urządzenia.

Pomocne okazały się również darmowe programy STM32 ST-LINK Utility, jako alternatywne narzędzie do programowania i łączenia się z mikrokontrolerem oraz STMStudio, pozwalający na monitorowanie wartości zmiennych podczas pracy robota, który w szczególności użyty został do testu czujników linii.

### 2.3.1 Konfiguracja peryferii

Do wykrywania czarnej linii wykorzystano przetwornik ADC. Skonfigurowany został do pracy w trybie DMA (Direct Memory Access, z ang. bezpośredni dostęp do pamięci). Technologia ta pozwala na transfer danych pomiędzy pamięcią a peryferiami z pominięciem procesora. Wykorzystana została do zapisu odczytów z czujników optycznych.

Sterowanie silnikami oparte zostało na generacji sygnału PWM przez timer. Prescaler ustawiono na wartość 23, a AutoReload Register na 999. Umożliwia to regulacje mocy silnika na przedziale [0;1000]. Jest to w wystarczającej rozdzielcości w zastosowaniu do line followera. Przy taktowaniu 24MHz częstotliwość przerwań wynosi 1000Hz.

Pętla algorytmu sterowania robotem jest wykonywana w przerwaniu timera z częstotliwością 100Hz. Długość przerw wpływa na efekt działania członu różniczkującego regulatora.

### 2.3.2 Algorytm sterowania

Algorytm sterowania robota oparty jest o regulator PD (proporcjonalno-różniczkujący) wykonywany w przerwaniu timera z częstotliwością 100Hz. Odczyt z czujników, na podstawie którego obliczany jest uchyb regulacji, wykonywany jest w trybie DMA. Następnie na silnik podawana jest nowa wartość mocy z dostępnego przedziału wyznaczona poprzez regulator.

Istotnym elementem algorytmu jest zapamiętywanie ostatniego błędu położenia względem linii. Wypadnięcie z trasy skutkuje zwiększonym uchybem.

Zastosowano stopniowe rozpędzanie przy startcie, gdyż unosiła się przednia część robota.

### **3 Podsumowanie**

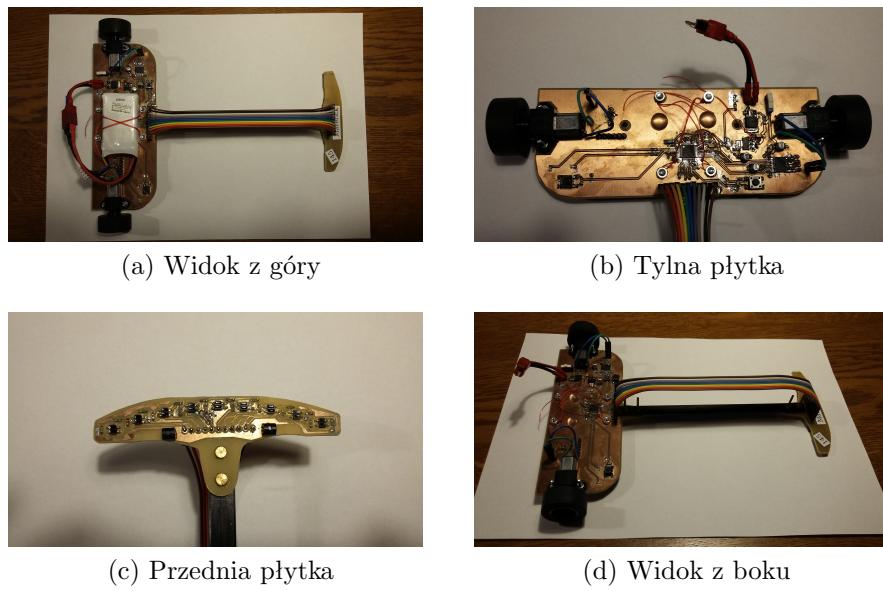
W dokumencie opisano proces budowy robota klasy Line Follower. Prace nad nim trwały od 8 listopada do 12 grudnia, czyli trochę ponad miesiąc. Cel został osiągnięty: udało się wystawić Mavericka na Robotic Arenie 2015. Podczas zawodów robot nie sprawiał żadnych problemów, więc można się było skupić na testowaniu różnych parametrów i ustawień. Maverick poradził sobie zadziwiająco dobrze - w klasie Line Follower Light uzyskał 6. miejsce, czyli niewiele brakowało, a wszedłby do finału.

Podsumowując, robot przeszedł najważniejszą próbę i sprawdził się na zawodach. Okazał się dobrą konstrukcją, w której jednak nadal jest sporo rzeczy, które można ulepszyć. Przy dalszych pracach nad Maverickiem warto byłoby skupić się na następujących kwestiach:

- Odchudzenie konstrukcji: wymiana listewki, zmiana sposobu mocowania akumulatora.
- Zwiększenie przyczepności: lepsze wyważenie dzięki przeniesieniu akumulatora, odlanie własnych opon.
- Zmniejszenie lub zlikwidowanie opóźnienia: prawdopodobnie dzięki lepszemu wyważeniu można by zniwelować podrywanie się przodu przy startcie.
- Ulepszenie algorytmu.

### **4 Materiały źródłowe**

Korzystano z wiedzy uzyskanej na warsztatach robotycznych oraz z wielu materiałów ogólnodostępnych w sieci. Najbardziej pomocna okazała się strona [www.forbot.pl](http://www.forbot.pl) oraz KoNaRowe archiwum: <http://lirec.iiar.pwr.edu.pl/~konarch/download/linefollower/>



Rysunek 8: Maverick w różnych ustawieniach



Rysunek 9: Bonus: Maverick wykonany z piernika