



Cam' On 1.0

Projekt platformy omnikierunkowej

Fryderyk Melka
Mateusz Żarkowski

Wrocław, 29 grudnia 2010 roku

Spis treści

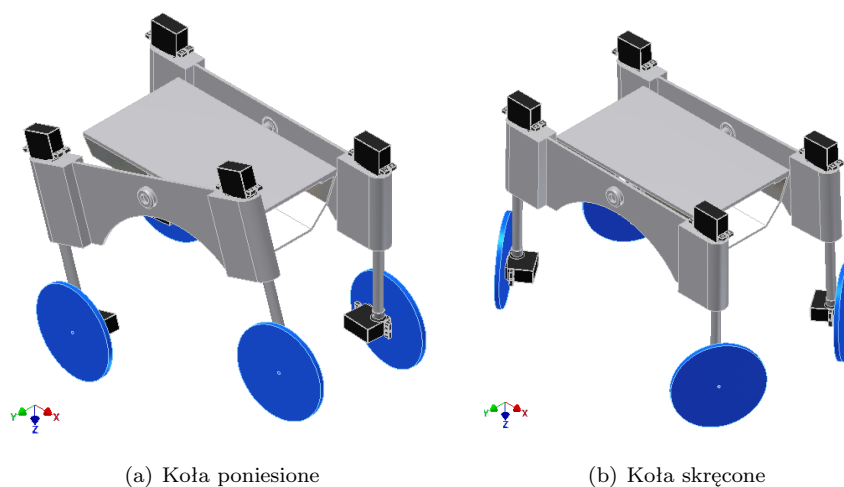
1	Konstrukcja mechaniczna	2
1.1	Holonomiczny, czy nieholonomiczny? — oto jest pytanie	2
1.2	Problem styku kół z powierzchnią	2
1.3	A co z samą platformą?	2
2	Elektronika	3
2.1	Sterownik napędów	3
2.2	RFM-12B	3
2.3	Nadajnik	4
2.4	Odbiornik	4
2.5	Ogólny schemat przepływu danych	5
2.6	Kształty ramek	5
3	Algorytmy sterowania	5
3.1	Instant Center of Rotation	6
3.2	Dwa problemy i trochę matematyki	6
4	Oprogramowanie sterujące	6
5	Cam' On 2.0	7
5.1	Wstęp	7
5.2	Linux vs router	7
5.3	Mechanika 2.0	8
6	Dodatki	10
A	Schemat sterownika serwomechanizmów	10
B	Schemat nadajnika	11
C	Schemat odbiornika	12

1 Konstrukcja mechaniczna

Powszechnie wiadomo, że dobra konstrukcja mechaniczna jest w stanie zaoszczędzić później wiele pracy na etapie sterowania i analogicznie. Aby określić funkcjonalność robota trzeba było odpowiedzieć sobie na parę ważnych pytań

1.1 Holonomiczny, czy nieholonomiczny? — oto jest pytanie

Sam robot w założeniu powinien być na tyle uniwersalny, żeby umożliwiał testowanie sterowania naprawdę znacznie różniących się od siebie modeli. Robot musiał być więc holonomiczny – aby mógł poruszać się bez żadnych ograniczeń, oraz tak wykonany konstrukcyjnie, aby w prosty sposób można było te ograniczenia nałożyć, np. poprzez blokadę orientacji kół na jednej osi (samochód Ackermanna), czy sprzężenie kół parami (sterownie różnicowe).



Rysunek 1: Model wykonany w programie Autodesk Inventor

1.2 Problem styku kół z powierzchnią

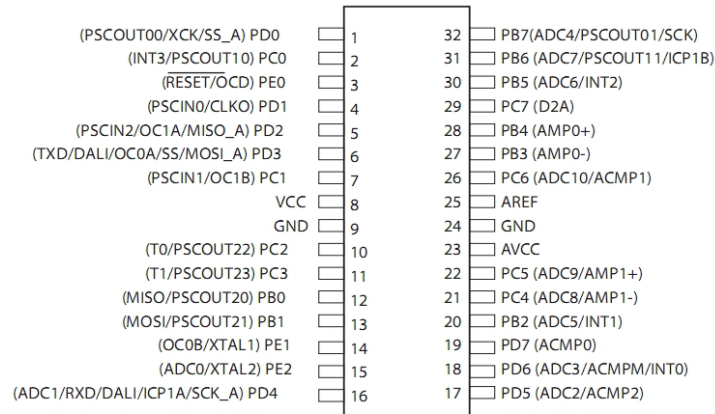
Wiadomo jest, że trzy punkty zawsze leżą na jednej płaszczyźnie, podczas gdy cztery koła nie zawsze stykają się z podłożem. W samochodach osobowych problem ten rozwiązywany jest przez zawieszenie.

Szczytem finezji przy poruszaniu się po trudnym terenie wykazują się lądowiki kosmiczne. Wykorzystana w projekcie modyfikacja zawieszenia typu *rocker-bogey* pozwoliła robotowi na pokonywanie znacznych trudności terenu, przy zachowaniu styczności wszystkich kół z ziemią.

Zastosowanie taśmy samowulkanizującej zapewniło robotowi na tyle dobrą przyczepność, że był on w stanie przejechać przeszkody większe od swoich kół, a nawet podjechać pod pionową ścianę.

1.3 A co z samą platformą?

Zastosowanie ruchomego, niezależnego od siebie zawieszenia pozwoliło na dowolność w ustawieniu głównej platformy robota. Przetestowano i zastosowano eksperymentalne zamocowanie platformy z wykorzystaniem tylko jednej poziomej rurki metalowej jako osi podparcia. Zamocowanie elementów ciężkich u podstawy tej platformy nadaje jej cechy stabilnego wahadła i zapewnia, że sama platforma zawsze będzie umieszczona równolegle do podłoża, co może bardzo się przydać kiedy na podstawie obrazu będziemy chcieli poznać położenie robota.



Rysunek 2: Atmel AT90PWM3B – schemat wyprowadzeń.

2 Elektronika

2.1 Sterownik napędów

Sterownik robota miał zagwarantować poprawny odbiór rozkazów z jednostki nadrzędnej i synchronizację 8 serwomechanizmów, po 2 na każde koło – jedno odpowiedzialne za orientację, a drugie (zmodyfikowane) za napęd. Z uwagi na postawione przez nas wymagania, mikrokontroler spełniać musiał następujące kryteria

- Moc obliczeniowa wystarczająca do wystarczająco szybkiego wyliczenia orientacji i prędkości 4 kół, na co składają się funkcje trygonometryczne
- Pamięć, nie koniecznie dużą, wystarczająca do umieszczenia bibliotek
- PWM, interesowało nas przynajmniej 8 PWM'ów do niezależnego sterowania serwami. Można było to zrobić programowo, ale zrezygnowano uwagi na możliwe pomyłki, oraz chęć, by program działał niezależnie od sprzętu
- Komunikacja miała umożliwiać debugowanie programu, ale w miarę rozwoju projektu zdecydowano się wykorzystać transmisję szeregową do sterowania robotem. Najpierw z wykorzystaniem modułów radiowych, a aktualnie instrukcje wysyłane są przez interface WWW routera wifi

Zastosowano więc mikrokontroler *AT90PWM3b* firmy Atmel (Rys. 2), który spełnia wszystkie požądane przez nas kryteria, warto dodać, że posiada tak zwany *Power Stage Controller* – jest to system do sterowania silnikami, który z powodzeniem można wykorzystać jako PWM. Schemat sterownika został przedstawiony na rysunku rys. 8

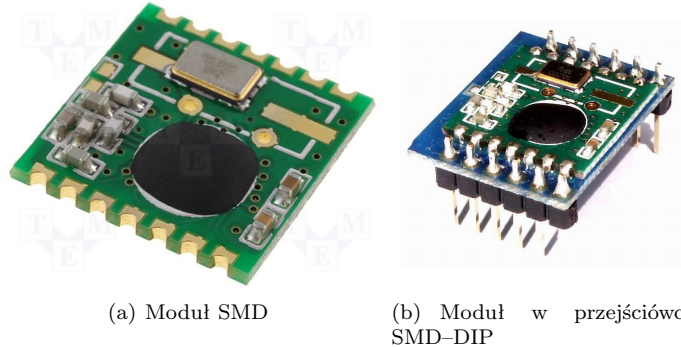
2.2 RFM-12B

Do komunikacji pomiędzy robotem, a komputerem został wybrany układ *RFM-12B*. Jest to transceiver pracujący w pasmie radiowym. Krótka charakterystyka:

- transceiver,
- praca w pasmie 868MHz,
- duży zasięg,
- napięcie zasilania V_{CC} 3.3V, ale...
- komunikacja z układem poprzez interfejs SPI,

- łatwa konfiguracja.

Za wyborem tego układu przemówił prosty interfejs komunikacyjny poprzez SPI, niska cena oraz dobry zasięg. Pomimo tego, że producent deklaruje, że układ może być zasilany tylko napięciem 3.3V z sukcesem udało się go podpiąć pod wyższe 5V. Oba układy w nadajniku i odbiorniku działają bezproblemowo. Istnieje wersja oznaczona symbolem *RFM-12*, która jest przeznaczona do pracy z wyższym napięciem zasilania, jednak, w momencie kupowania elementów elektronicznych do projektu, nie była dostępna.



Rysunek 3: Moduł RFM12B

2.3 Nadajnik

W module nadajnika (rys. 9), jako jednostkę centralną, wybrano mikrokontroler *Atmega16* firmy *Atmel*. W celu poznania dokładniejszych parametrów tego układu odsyłam do jego dokumentacji [5].

Jako układy peryferyjne zostały zastosowane:

- złącza Lewy i Prawy służące do podłączenia modułu *RFM-12B*,
- złącza JP1 i JP2 służące do podłączenia klawiatury matrycowej 4x4,
- złącze POTY służące do podłączenia do trzech potencjometrów,
- dioda LED3 oraz dwa przyciski BUTTON1 i BUTTON2 ogólnego zastosowania,
- standardowa aplikacja układu *MAX232* – konwertera poziomów TTL/RS232.

Nadajnik posiada możliwość podłączenia klawiatury matrycowej oraz potencjometrów po to, żeby w przyszłości móc sterować robotem bez pośrednictwa komputera tylko z wykorzystaniem joysticka analogowego podpiętego bezpośrednio pod moduł nadawczy.

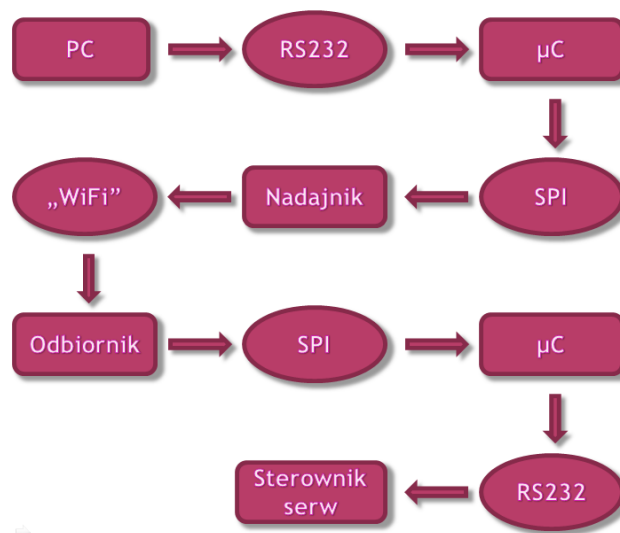
2.4 Odbiornik

W module odbiornika (rys. 10), jako jednostkę centralną, wybrano mikrokontroler *Atmega8* firmy *Atmel*. W celu poznania dokładniejszych parametrów tego układu odsyłam do jego dokumentacji [4].

Jako układy peryferyjne zostały zastosowane:

- złącza LEWY i PRAWY służące do podłączenia modułu *RFM-12B*,
- diody LED1–LED3 oraz przycisk BUTTON ogólnego zastosowania,
- standardowa aplikacja układu *MAX232* – konwertera poziomów TTL/RS232.

Odbiornik jest tylko medium pośredniczącym w przepływie danych pomiędzy nadajnikiem, a elementem wykonawczym (w naszym przypadku jest to driver serwomechanizmów).



Rysunek 4: Schemat przepływu danych.

2.5 Ogólny schemat przepływu danych

Przepływ danych został zobrazowany na rysunku 4. Sterowanie robota Cam' On odbywa się z wykorzystaniem nadrzędnego komputera PC, z podłączonym gamepadem, za pomocą aplikacji napisanej w języku C#. Program czytuje nastawy joysticków analogowych oraz stany przycisków i wysyła je do nadajnika poprzez interfejs RS232. Mikrokontroler w module nadawczym odbiera dane, konwertuje je formatu akceptowalnego przez nadajnik, a następnie, poprzez interfejs SPI przesyła dane do ww. nadajnika. Następuje przekaz danych drogą radiową do odbiornika. W module odbiorczym dane przebywają odwrotną drogą, niż w module nadawczym. Odbiornik odebrał dane, wysyła je po SPI do mikrokontrolera. Następnie dane są konwertowane do formatu, który jest akceptowalny przez sterownik serwomechanizmów oraz są wysyłane do niego. Efektem tych wszystkich zabiegów jest ruch robota w zadanym kierunku.

2.6 Kształty ramek

W systemie bezprzewodowej komunikacji istnieją dwa rodzaje ramek. Pierwsza z nich jest przesyłana poprzez RS232, druga natomiast jest przesyłana zdalnie pomiędzy nadajnikiem, a odbiornikiem ("WiFi"). Wyglądają one następująco:

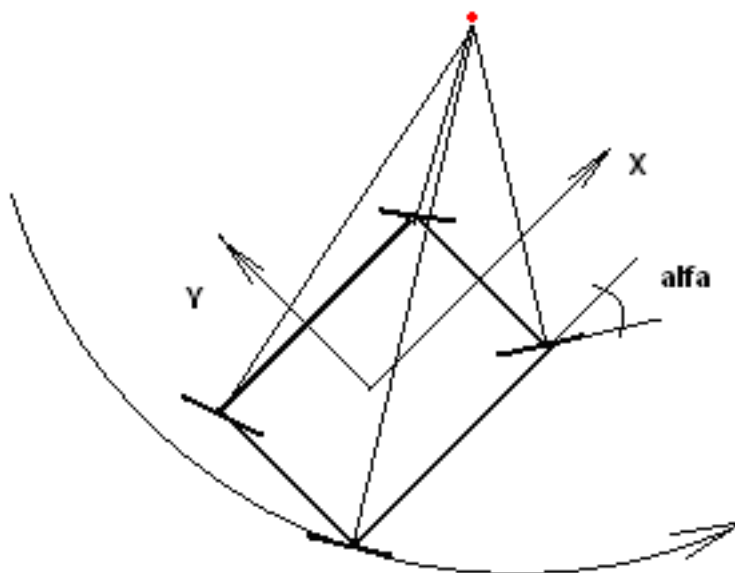
- RS232
 - 8 bitów danych,
 - brak kontroli parzystości,
 - 1 bit stopu.
- "WiFi"
 - 3 bajty preambuły (0xAA),
 - 2 bajty synchronizacyjne (0x2D oraz 0xD4),
 - 8 bajtów danych,
 - 1 bajt sumy kontrolnej.

3 Algorytmy sterowania

Wysterowanie 8 serwomechanizmów jest problemem złożonym i z początku były obawy, że może to nas przerosnąć. Po burzy mózgów zdecydowano się wykorzystać autorski pomysł ICR.

3.1 Instant Center of Rotation

Aby spełnić założenia o braku poślizg bocznego przyjęto, że każdy kierunek jazdy danego koła musi być prostopadły do promienia koła po jakim porusza się robot. Oznacza to, że dla każdego koła można znaleźć wektor normalny do jego powierzchni (inaczej zgodny z kierunkiem jego prędkości kątowej). Wektory te, pochodzące ze wszystkich kół, muszą przecinać się w jednym punkcie, który jest środkiem obrotu całej platformy. Idea została przedstawiona na rys. 5.



Rysunek 5:

Poprzez zadawanie punktów *ICR*, w układzie współrzędnych związanym z robotem, jesteśmy w stanie realizować pożądane trajektorie. Ponadto warto zauważyć, że dla *ICR* leżącego bardzo daleko wynikiem jazdy jest linia prosta (koło o nieskończenie dużym promieniu).

3.2 Dwa problemy i trochę matematyki

Pozostał jedynie problem wyliczania prędkości i orientacji kół, który rozwiązujemy z pomocą matematyki.

Dysponujemy następującą wiedzą - znamy położenie *ICR* oraz każdego koła w układzie robota. Na podstawie tego wyznaczamy wektor *ICR*-koło i korzystając z funkcji *Atan2* (biblioteka *math.h*) wyliczamy pożądaną orientację kółka.

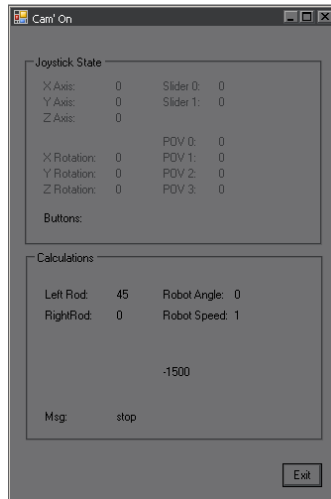
Każde koło ma poruszać się po okręgu o środku w punkcie *ICR* i promieniu *R*. Jego prędkość postępową w trakcie jednego okresu wynosi.

$$v_i = \frac{2\pi R_i}{T}$$

Oznacza to, że prędkość każdego z kół jest proporcjonalna do jego odległości od *ICR* (długość wcześniej wyznaczonego wektora *ICR*-koło). Ostatnim krokiem jest normalizacja uzyskanych prędkości. Przyjmuje się maksymalną prędkość, z jaką będzie się kręcić najszybsze (najdalsze od *ICR*) koło, a resztę dostosowuje się do niego.

4 Oprogramowanie sterujące

Algorytm opisany w rozdziale 3 jest realizowany poprzez oprogramowanie na komputerze PC. Jego interfejs jest przedstawiony na rysunku 6. Umożliwia on odczyt stanu joysticka oraz wyświetla aktualnie obliczone nastawy.



Rysunek 6: Interfejs programu.

Manualne sterowanie robotem odbywa się poprzez gamepada. Z kierunkiem jazdy skojarzono lewy dżączek analogowy, natomiast do ustalenia prędkości wykorzystywane są przyciski od 1 do 4. Po ustawieniu pożądanego położenia, należy je zatwierdzić przyciskiem 8. Skrećanie odbywa się poprzez przesuwanie chwilowego środka skrętu wzdłuż kierunku prostopadłego do chwilowego ustawienia kół. Przyciski lewo i prawo pozwalają na dokładne ustawienie pozycji, natomiast góra i dół na manipulacje zgrubne.

W projekcie wykorzystano gotowe biblioteki [8] do obsługi joysticka w języku C#.

5 Cam' On 2.0

5.1 Wstęp

Kolejna wersja robota będzie posiadać zmienioną elektronikę sterującą. Jego jednostką centralną ma się stać router Linksys WRT54G z zainstalowanym zmodyfikowanym firmware'm (Linux). Wstępne testy pokazały, że można uzyskać dostęp do wbudowanych portów szeregowych (patrz rozdział 5.2). Niestety fizyczne uszkodzenie routera uniemożliwiło dalsze badania.

Zmianie ulegnie także mechanika robota na bardziej *profesjonalną* (patrz rozdział 5.3).

5.2 Linux vs router

Zastosowanie routera jako sterownika nadrzędnego jest przewidywanym kierunkiem rozwoju tej platformy. Z uwagi na komplikacje związane z konfiguracją portu szeregowego przy użyciu dystrybucji *White Russian* i wymagającym skomplikowanej kompilacji źródeł *Kamikaze*, prace zostały znacznie utrudnione. Rozwiązanie pojawiło się w postaci nowej dystrybucji *OpenWrt – Backfire 10.3*.

Instalacja, podobnie jak wcześniej, odbywa się za pomocą przeglądarki i jest bardzo prosta. *Backfire* umożliwia dodawanie pakietów rozszerzeń zarówno z poziomu przeglądarki, jak i okna konsoli. W naszym wypadku, do konfiguracji portu szeregowego, konieczna była instalacja pakietu `setserial`, co w zupełności wystarczyło do inicjalizacji obu portów.

```
setserial -a /dev/tts/0
setserial -a /dev/tts/1
cat /proc/tty/driver/serial/
```

Po wykonaniu powyższych instrukcji, zainicjalizowane zostaną porty oraz wyświetlą się informacje na temat ich konfiguracji. Domyślne prędkości transmisji to:

- 115200 baudów dla `tts0`

- 9600 dla tts1

Prędkości te można dowolnie ustawiać korzystając z poniższego wywołania.

```
setserial -a /dev/tts/1 baud_base <baudbase>
```

Należy pamiętać, że ustawiamy jedynie wartość bazową, która jest następnie dzielona (szczegóły w specyfikacji `setserial`). W każdej chwili można sprawdzić aktualną wartość prędkości poprzez poniższe wywołanie.

```
cat /proc/tty/driver/serial/
```

Uruchomienie portu szeregowego już umożliwia bezprzewodową komunikację z robotem i wysyłanie nastawień do sterownika. Przewidywany kierunek rozwoju projektu zawiera dodanie interfejsu `www`, umożliwiającego intuicyjne sterowanie robotem poprzez przeglądarkę. Interfejs ten ma zostać wykonany z użyciem skryptów `cgi`.

5.3 Mechanika 2.0



Rysunek 7: Obecnie rozwijana wersja robota

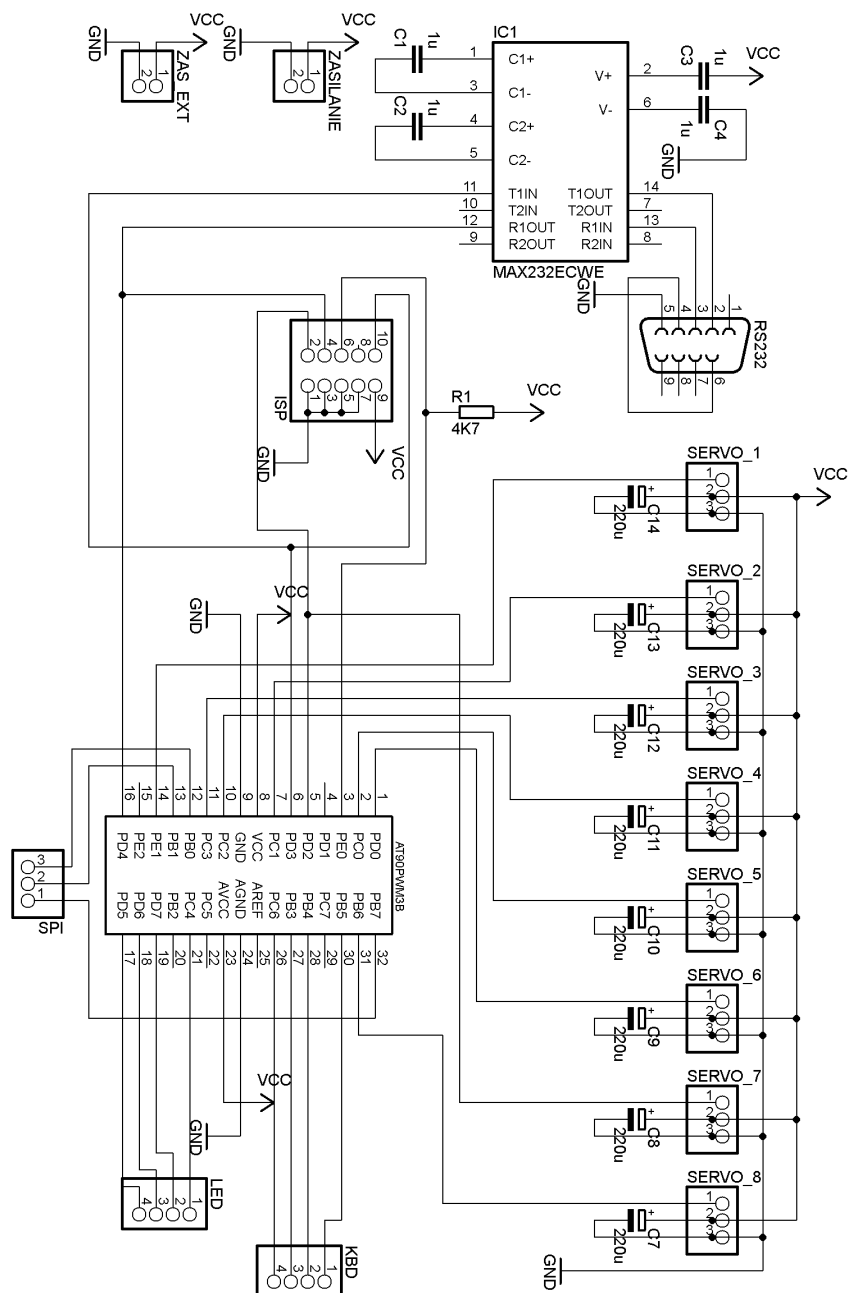
Nowa wersja robota będzie posiadała zmienioną mechanikę. Drewniane elementy zostały zastąpione nowymi częściami wyciętymi z białej pleksi o grubości 10mm. Zmieniony zostanie także sposób mocowania kół. Stary (oba serwa bezpośrednio połączone ze sobą) nie zapewniał wystarczającej sztywności. W nowym rozwiązaniu zostanie wykorzystana łożyskowana tuleja pomiędzy serwami. Testy dowiodły znaczne zmniejszenie się elastyczności mocowania.

Literatura

- [1] http://www.societyofrobots.com/robot_taurus2.shtml, Taurus 2 Robot
- [2] <http://en.wikipedia.org/wiki/Rocker-bogie>, Wikipedia
- [3] http://www.atmel.com/dyn/resources/prod_documents/doc4317.pdf, AAtmel AT90PWM3B
- [4] http://www.atmel.com/dyn/resources/prod_documents/doc2486.pdf, ATMEGA8 datasheet, Atmel, Jun 2009.
- [5] http://www.atmel.com/dyn/resources/prod_documents/doc2466.pdf, ATMEGA16 datasheet, Atmel, May 2009.
- [6] <http://datasheets.maxim-ic.com/en/ds/MAX220-MAX249.pdf>, MAX232 datasheet, Maxim
- [7] <http://www.hoperf.com/upfile/RFM12B.pdf>, RFM12B datasheet, Hoperf
- [8] <http://www.codeproject.com/KB/directx/joystick.aspx>, C# joystick library, Mark Harris
- [9] http://en.wikipedia.org/wiki/Instant_centre_of_rotation, Wikipedia

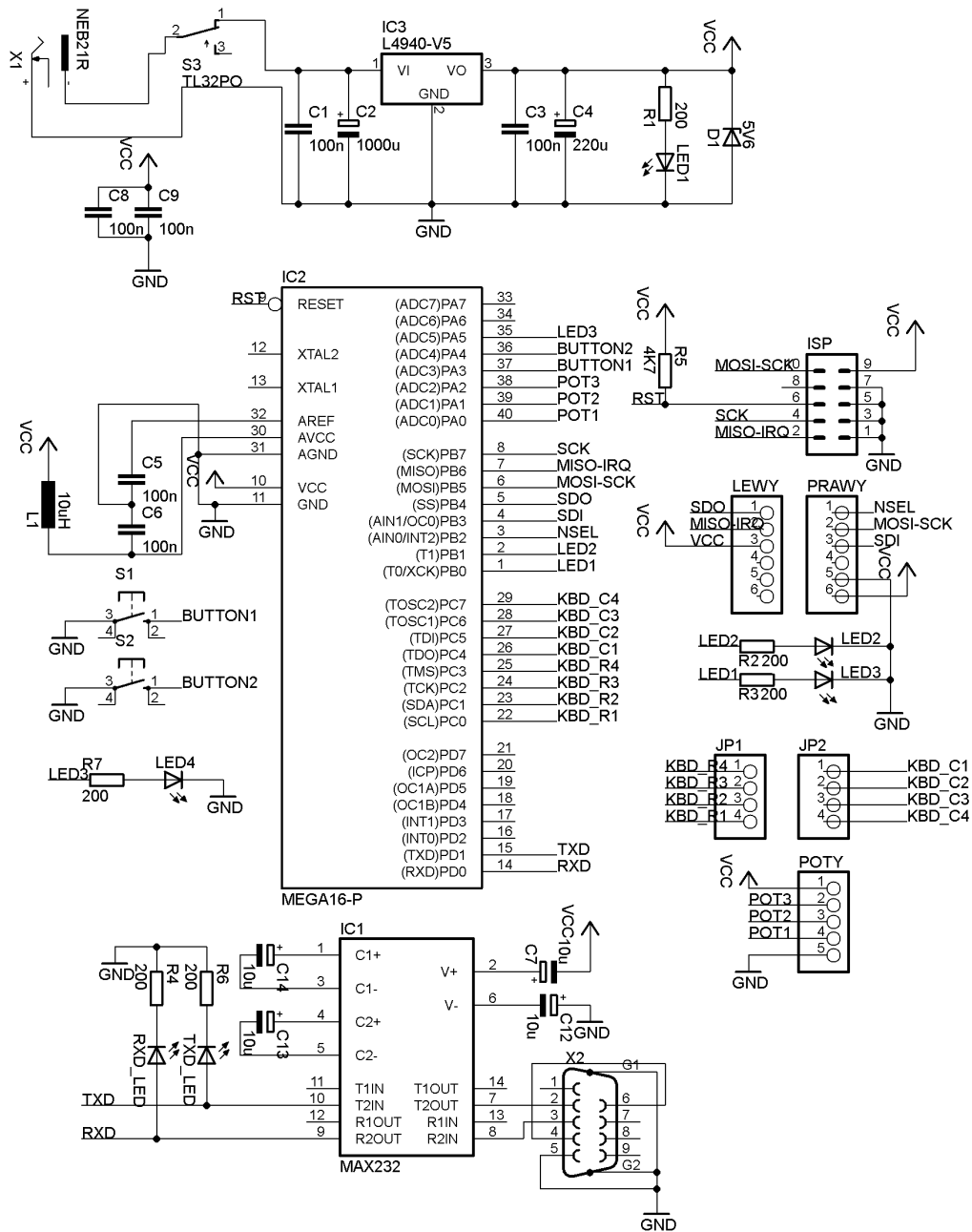
6 Dodatki

A Schemat sterownika serwomechanizmów



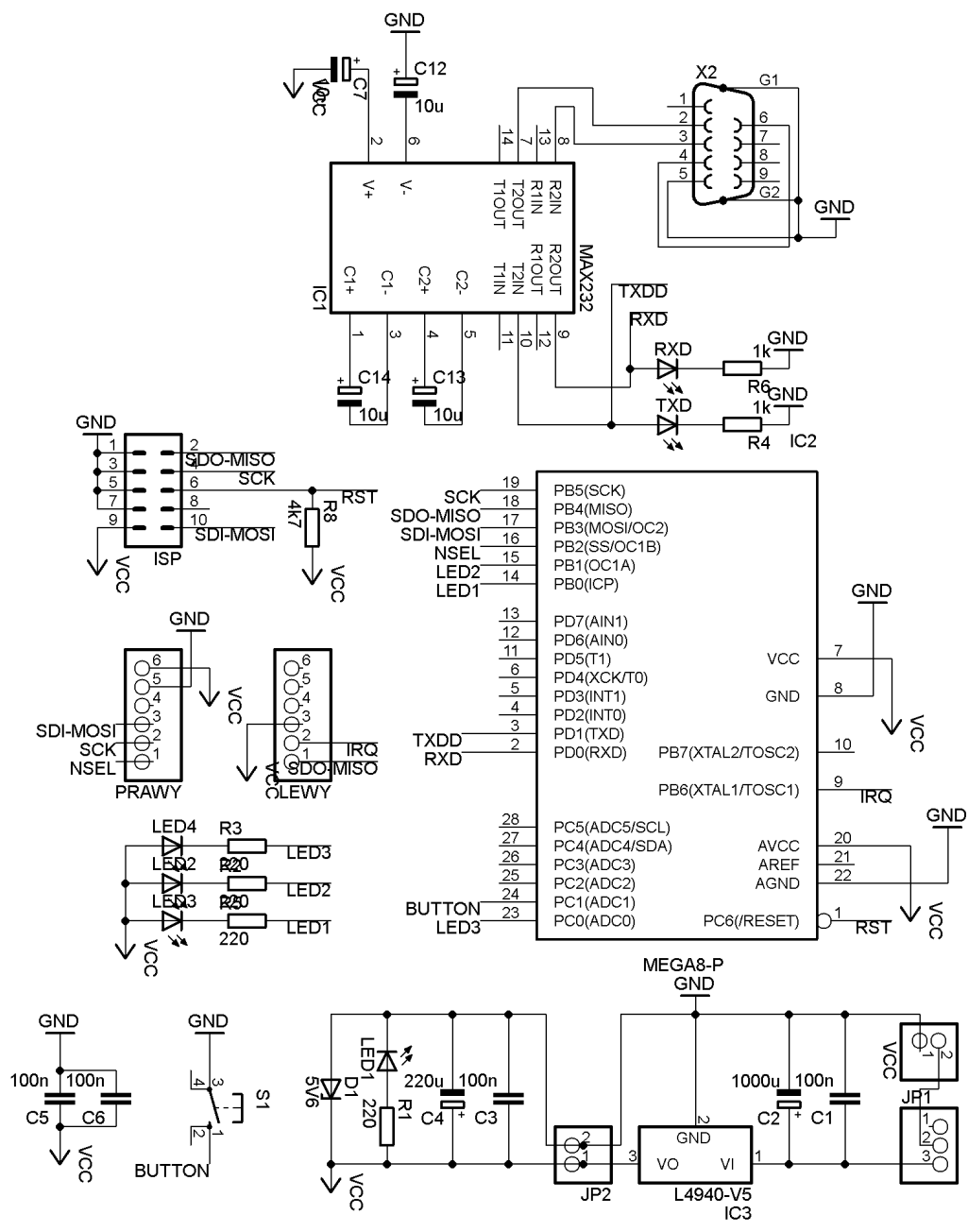
Rysunek 8: Driver serwomechanizmów – schemat.

B Schemat nadajnika



Rysunek 9: Nadajnik – schemat.

C Schemat odbiornika



Rysunek 10: Odbiornik – schemat.