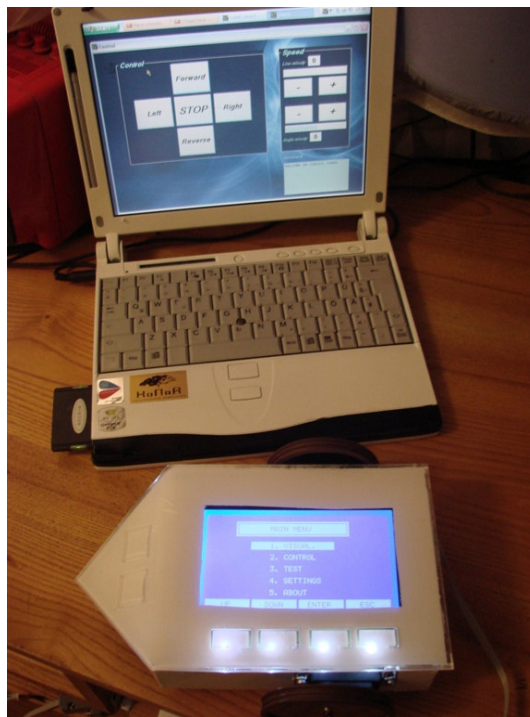




*Rozpoznawanie zdarzeń za pomocą
czujnika optycznego z myszki w robocie mobilnym*

**Jan Kędziński
Edgar Ostrowski**



Wrocław, 28.01.2007

Spis Treści

1	Wstęp	2
2	Platforma	2
3	System napędowy	4
4	Płyta główna	7
5	Klawiatura	12
6	Wyświetlacz LCD	13
7	Sterowanie	14
	7.1 Odometria	14
	7.2 Regulator PID	16
	7.3 PWM	16
	7.4 Wykrywanie błędu ruchu	17
8	Oprogramowanie robota	19
9	Sterowanie za pomocą komputera PC	20
10	Literatu	22

1 Wstęp

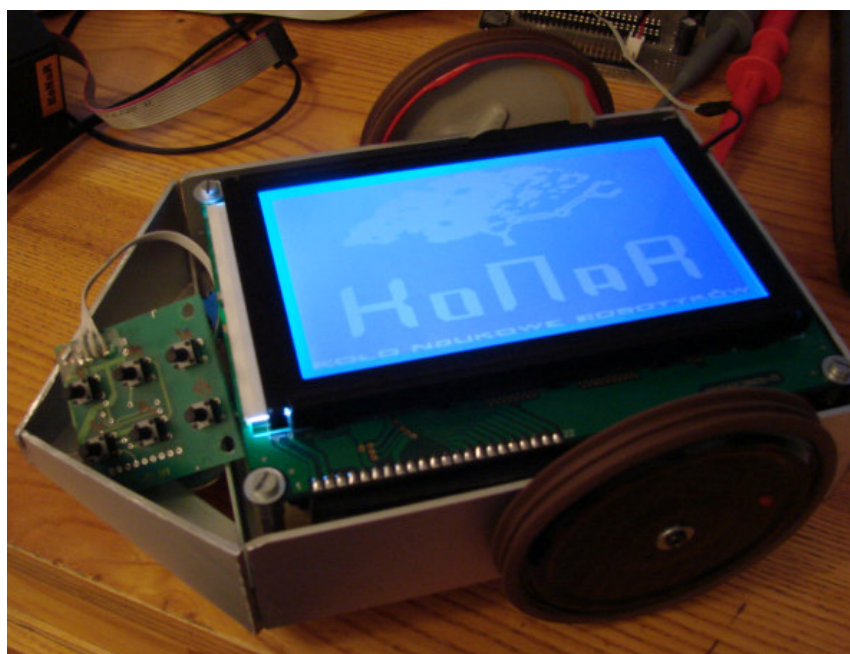
Niniejszy raport jest sprawozdaniem z zajęć projektowych „Systemy Autonomiczne i Roboty Mobilne” prowadzone przez dr Andrzeja Wołczowskiego w semestrze zimowym, roku akademickiego 2006/07 na Wydziale Elektroniki, Politechniki Wrocławskiej.

Dokument zawiera opis platformy mobilnej, na której zamocowano czujnik z myszki optycznej oraz inne urządzenia, które są konieczne do przeprowadzenia badań dotyczących systemów wspomagania odometrii.

2 Platforma

Do celów projektu przygotowano dwukołową platformę mobilną typu unicycle, która jest napędzana dwoma kołami. Dodatkowy punkt podparcia stanowi koło kulowe, które może poruszać się swobodnie w każdą stronę. Na platformie umieszczono następujące urządzenia:

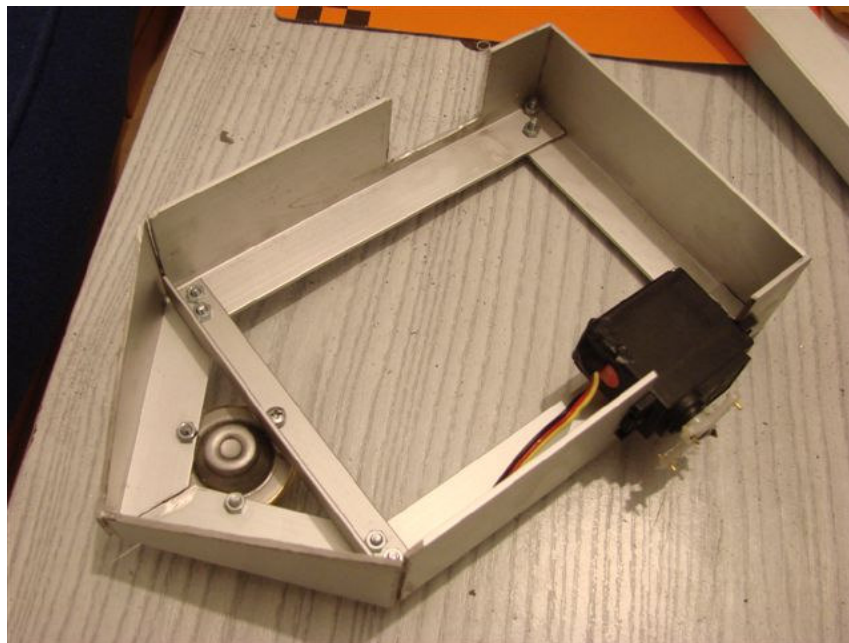
- Czujnik z myszki optycznej.
- Serwo mechanizmy do napędzania kół.
- Enkodery magnetyczne do mierzenia prędkości obrotowych kół.
- Graficzny wyświetlacz LCD
- Klawiaturę do obsługi robota.
- Płytę główną zawierającą elementy elektroniczne.



Fot. 1 – Platforma „BlueScreen”- prototyp.

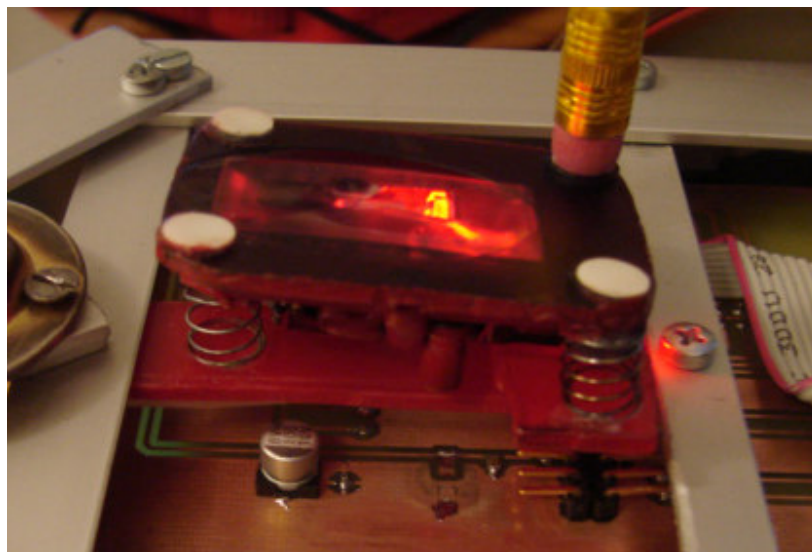
Ze względu na duży niebieski wyświetlacz, platformie została przyznana nazwa *BlueScreen*.

Konstrukcję mechaniczną zbudowano z aluminiowych kątowników nierównoramiennych. Dzięki temu uzyskano stabilny korpus o niewielkiej wadze. Całość skręcono za pomocą śrub i nakrętek M3 [Fot 2].



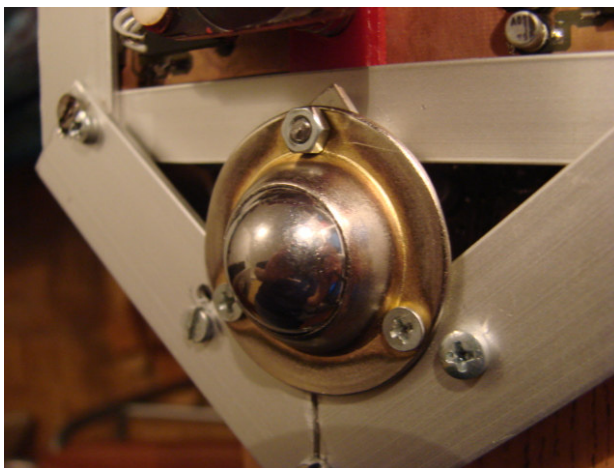
Fot. 2 – Korpus platformy.

Moduł z czujnikiem z myszki optycznej [1], zainstalowano na sprężynkach [Fot 3]. Sprężynki przyklejono do płaskowników z odpowiednio wygiętej pleksi. Całość pokryto czerwonym lakierem. Dokładny opis czujnika znajduje się w załączniku (wersja drukowana)



Fot. 3 – Mocowanie czujnika optycznego.

Dobrym pomysłem okazało się zainstalowanie w punkcie podparcia łożyska kulowego [Fot 4]. Doskonale radzi sobie z pokonywaniem niewielkich nierówności zapewniając przy tym małe tarcie elementu toczącego.



Fot. 4 – Mocowanie łożyska kulowego.

3 System napędowy

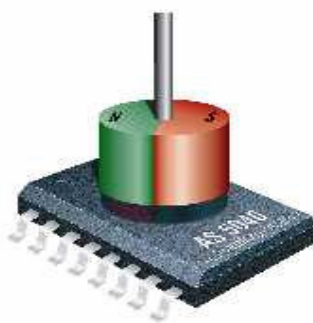
Platforma jest napędzana dwoma serwomechanizmami firmy Hi-TECH, które zostały odpowiednio zmodyfikowane [Fot 5]. Usunięto blokadę mechaniczną oraz elektroniczne elementy. Dzięki temu uzyskano wieloobrotowe silniki z przekładnią.



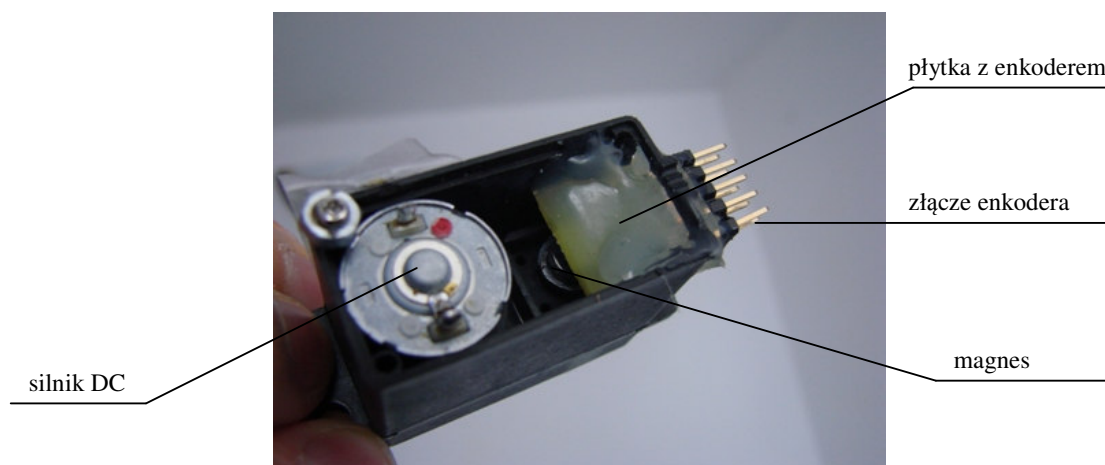
Fot. 5 – Widok przerobionego serwomechanizmu.

Silniki dodatkowo zostały wyposażone w rocesor magnetyczny firmy astriamicrosystems [3,4]. Zapewniają bardzo precyzyjny bezkontaktowy pomiar obrotów. Na

elemente ruchomy montuje się magnes [Rys 1], czujnik reaguje na zmiany natężenia pola magnetycznego i wyznacza kąt obrotu. Dzięki niewielkim rozmiarom układu oraz magnesu całość można zamontować wewnątrz obudowy serwomechanizmu. Dokładny opis enkodera znajduje się w załączniku.

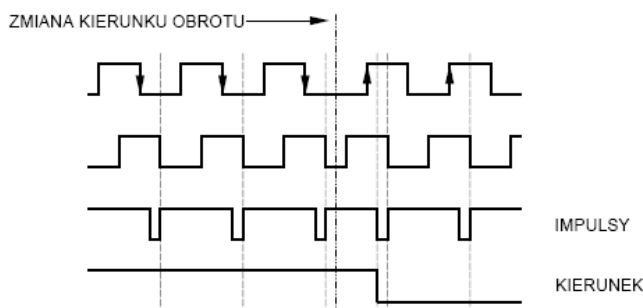


Rys. 1 – Enkoder magnetyczny.



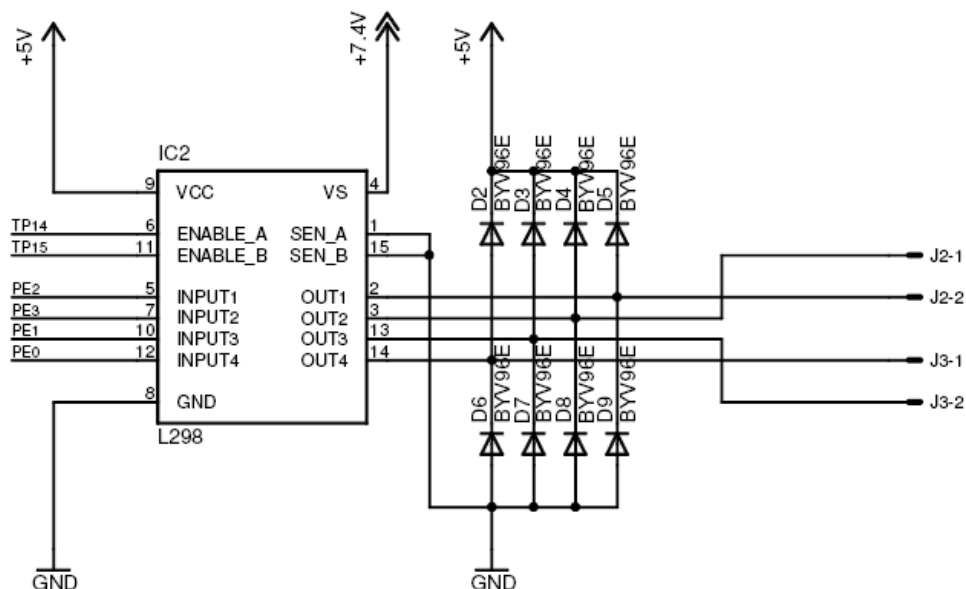
Fot 6 – Sposób mocowania enkodera.

Enkodery magnetyczne wykorzystano jak tradycyjne czujniki optyczne. Sygnały pochodzące z układów to dwa przesunięte względem siebie przebiegi prostokątne. Przesunięcie to wynosi $r. \frac{1}{4}$ całego okresu generowanego na wyjściu układu. Dzięki temu możliwe jest odczytanie kierunku oraz prędkości. Zasada pomiaru polega za zliczaniu w równych odstępach czasowych impulsów jednego z wyjść. Kierunek natomiast odczytujemy badając stan drugiego wyjścia w trakcie zbocza opadającego pierwszego wyjścia [Rys. 2].

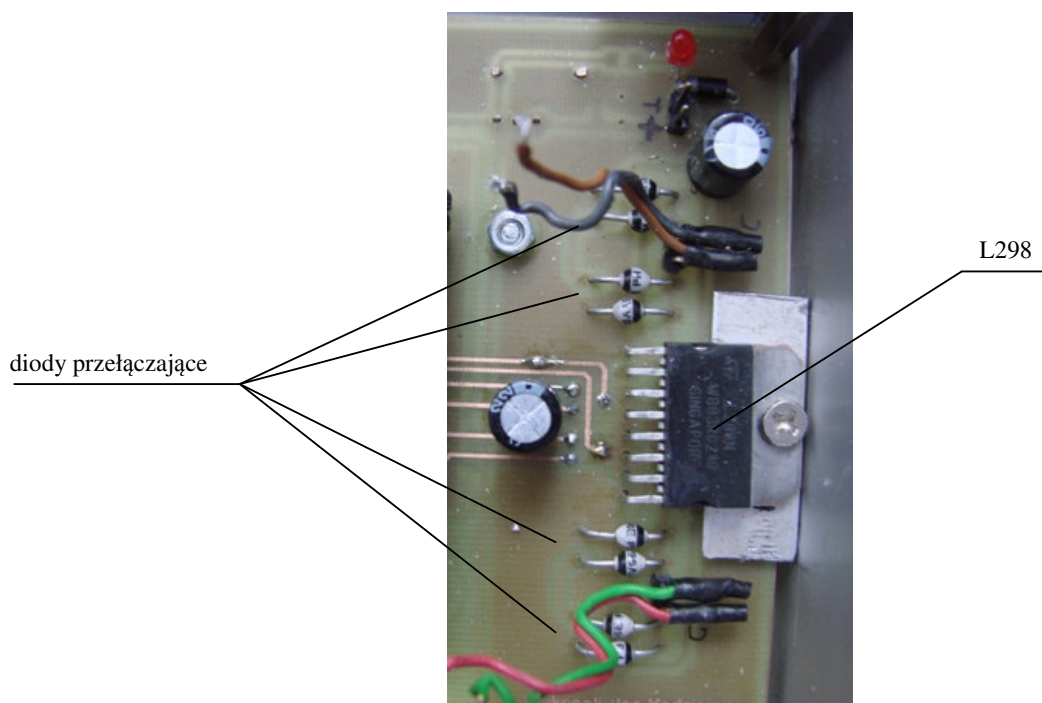


Rys. 2 – Interpretacja kierunku i prędkości.

Do sterowania silnikami wykorzystano gotowy układ scalony L298, który zawiera w swojej strukturze dwa mostki H. Zapewnia sterowanie przy maksymalnym prądzie 2 A na każdy silnik, zupełnie wystarczająco do celów naszego projektu. Pełny opis układu można znaleźć w dokumentacji fabrycznej [5][6]. Wyjścia układu wyposażono w diody przełączające, które zapewniają odprowadzenie prądów upływowych pochodzących od silników. Jest to konieczne aby zabezpieczyć układ. Poniższy rysunek przedstawia w sposób podłączenia układu.



Rys. 3 – Układ mostka L298.

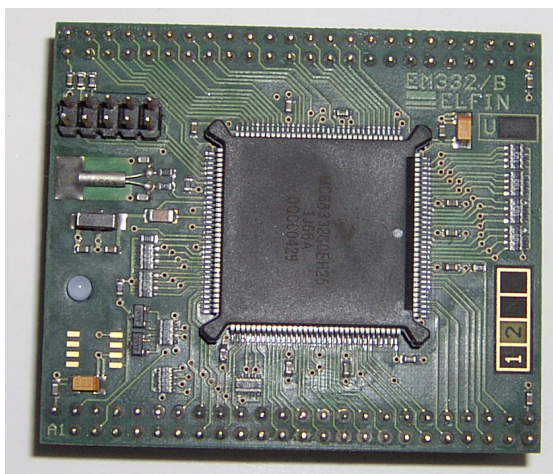


Fot 7 – Widok gotowego mostka.

4 Płyta główna

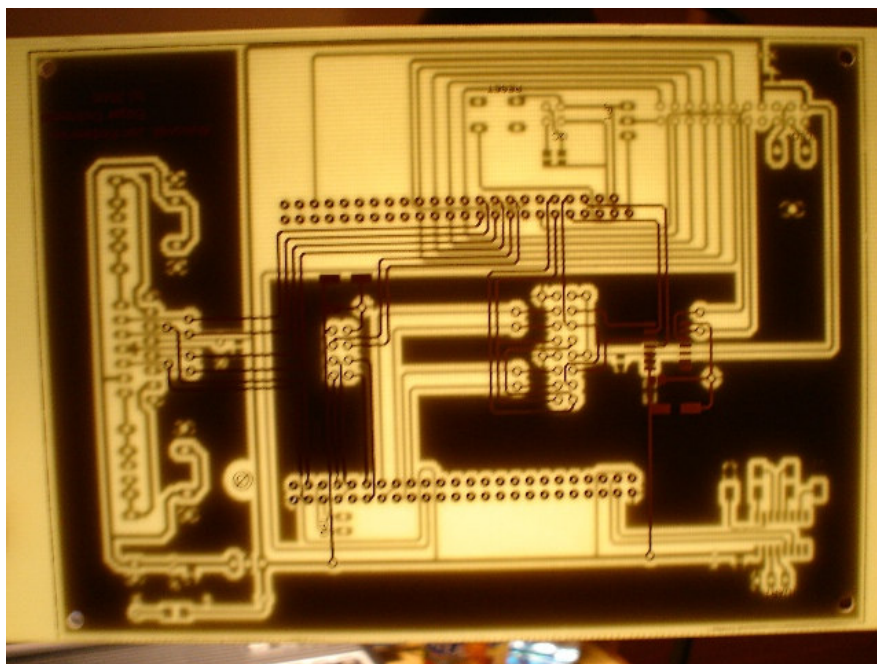
Do sterowania robota użyto modułu z mikrokontrolerem firmy MOTOROLA [Fot. 8]. Moduł ten został zaprojektowany przez Instytut Cybernetyki Technicznej Politechniki Wrocławskiej z przeznaczeniem do projektów badawczych i dydaktycznych.

Rodzina M683xx mikrokontrolerów firmy Motorola jest oparta na bardzo rozpowszechnionej rodzinie 16/32-bitowych mikroprocesorów M68K. Mikrokontrolery M683xx zostały wyposażone w 32-bitów , a jednostkę centralną (CPU32) opartą na procesorze 68020 oraz w zestaw układów wejścia/wyjścia (porty równoległe i szeregowy). Niemal ortogonalna architektura, bogata lista instrukcji i rozbudowane tryby adresowania czynią z CPU32 idealną jednostkę do programowania w językach wyższego rzędu. Z urządzeń peryferyjnych na szczególną uwagę zasługuje programowalny timer (*TPU – Time rocesor Unit*), który zawiera własny procesor i pozwala odciążyć CPU32 od częstych, powtarzalnych operacji związanych z funkcjami czasowo-licznikowymi. Funkcje TPU mogą być programowane przez użytkownika w mikrokodzie i wprowadzane do wbudowanej pamięci RAM na miejsce fabrycznie przygotowanych funkcji czasowych. Inne ułatwienie to kolejgowany interfejs synchroniczny urządzeń zewnętrznych (*QSPI . Queued Serial Peripheral Interface*). Zasoby pamięciowe mikrokontrolera są skromne (2kB RAM przeznaczone dla TPU), ale wbudowany blok programowalnych dekodów adresowych (*Chip Selects*) umożliwia bezpośrednie przyłączenie standardowych pamięci RAM i FLASH bez używania jakichkolwiek układów dodatkowych. Wszystkie funkcje mikrokontrolera są w szerokim zakresie programowalne, zapewniając wielką elastyczność i w rezultacie, łatwość dopasowywania konfiguracji do potrzeb. Bardzo ważną zaletą rodziny M683xx jest wbudowany emulator z interfejsem BDM (*Background Debug Mode*), który pozwala uruchamiać, testować i modyfikować oprogramowanie bezpośrednio na systemie docelowym (*tekst zaczerpnięty z raportu serii SPR nr 7/2004*) [7].

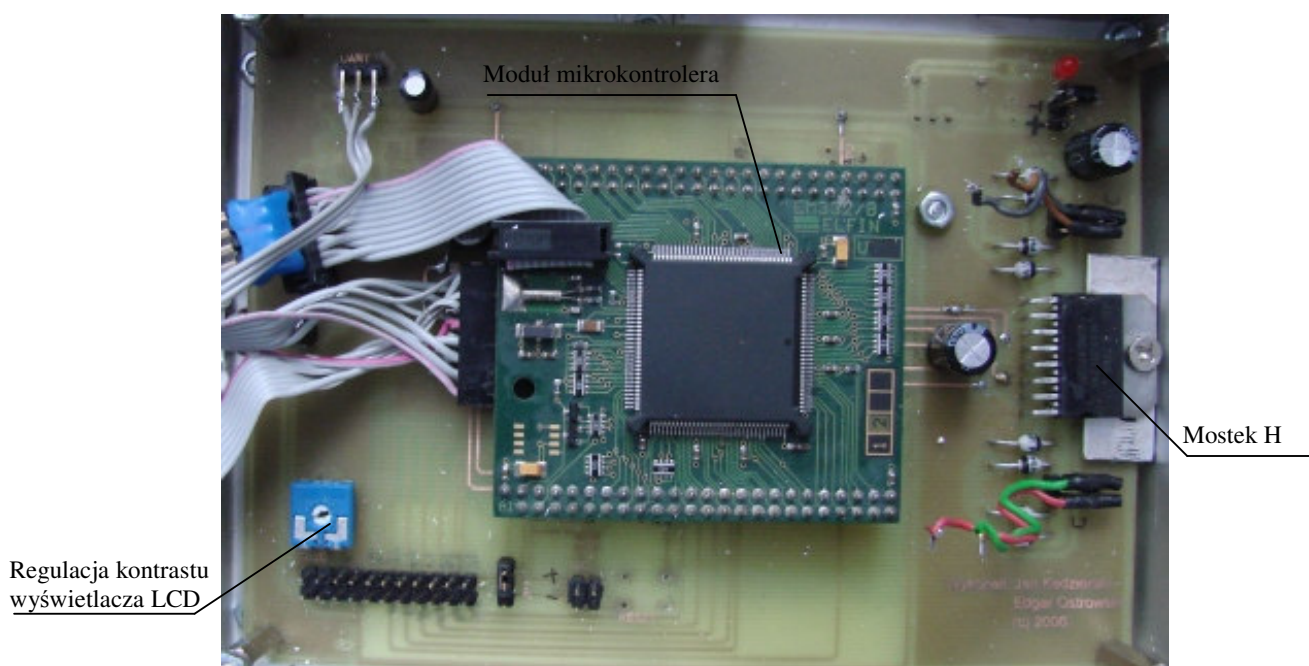


Fot. 8 – Widok gotowego mostka.

Płytę główną wykonano na dwustronnym laminacie [Fot 9]. Niemal wszystkie użyte elementy są w obudowie do montażu powierzchniowego. Sygnały sterujące doprowadzono przy użyciu taśm wielożyłowych oraz złączy [Fot 10]. Dzięki temu cały system poddawał się łatwemu serwisowi.



Fot. 9 – Widok płyty głównej.

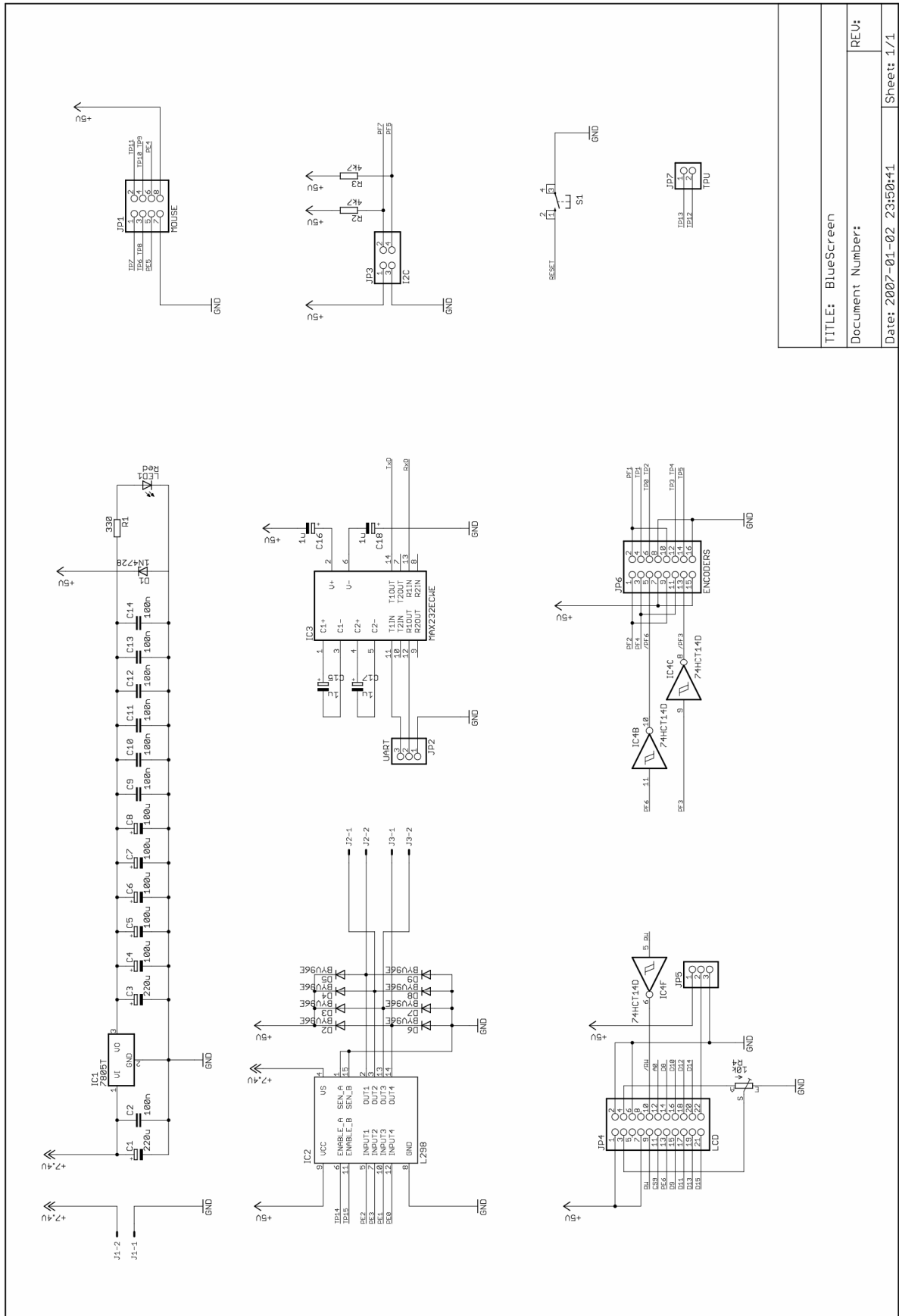


Fot. 10 – Widok płyty głównej z elementami.

W poniższej tabeli zestawiono wszystkie elementy, które znajdują się na płycie. [Fot 10] przedstawiają rozmieszczenie elementów.

Płytką została wykonana metodą domową jednak efekt końcowy jest bardzo imponujący.

Ilość	Wartość	Obudowa	Oznaczenie na schemacie
Rezystory			
1	330	0805	R1
2	4k7	0805	R2, R3
1	10k		R4
Kondensatory			
7	100n	0805	C2, C9, C10, C11, C12, C13, C14
4	1u	Ø4 x 5.3mm	C15, C16, C17, C18
5	100u	Ø6.3 x 5.3mm	C4, C5, C6, C7, C8
2	220u		C1, C3
Diody			
1	BZX85C5V1	DO41	D1
4	BYV96E	SOD64	D2, D3, D4, D5, D6, D7, D8, D9
Układy scalone			
1	LM2940CT-5	TO-220	IC1
1	L298	MUTIWATT15	IC2
1	MAX232D	SO-16	IC3
1	74HC14D	SO-14	IC4



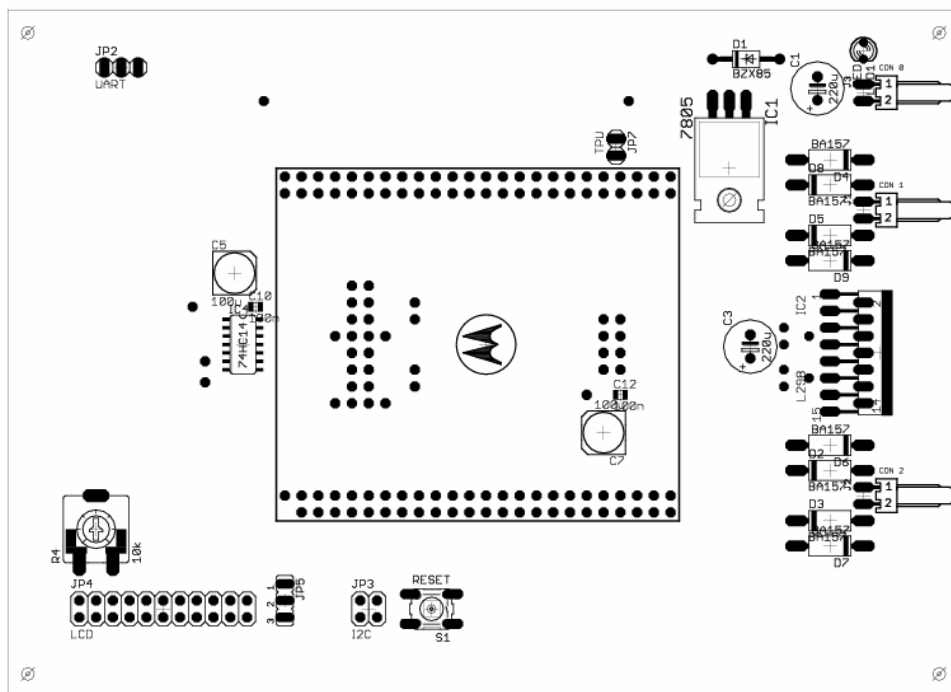
TITLE: BlueScreen

Document Number:

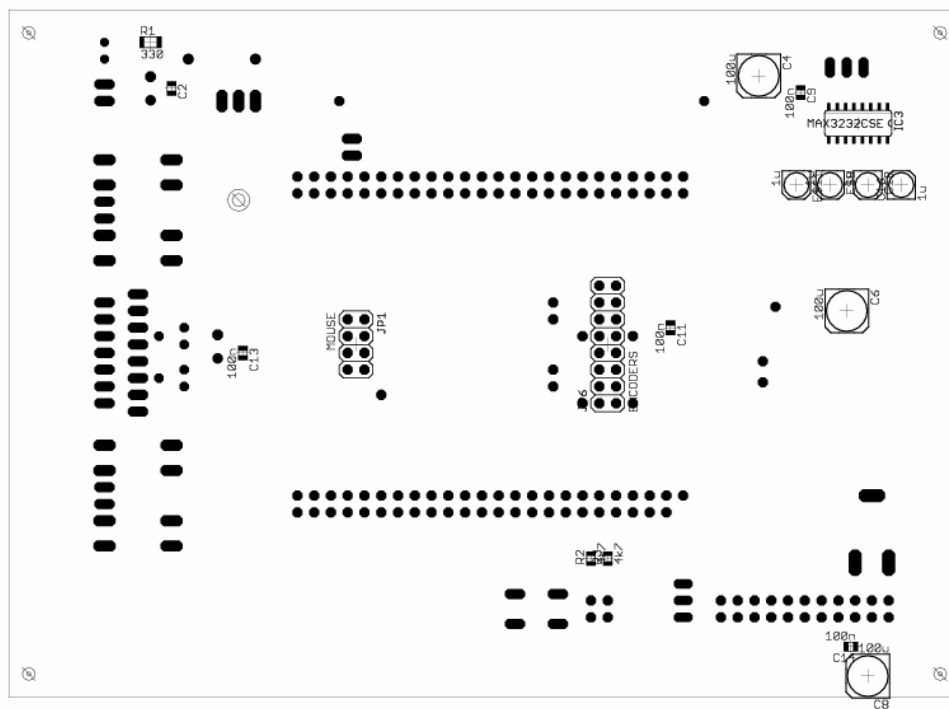
Date: 2007-01-02 23:50:41

REU:

Sheet: 1/1



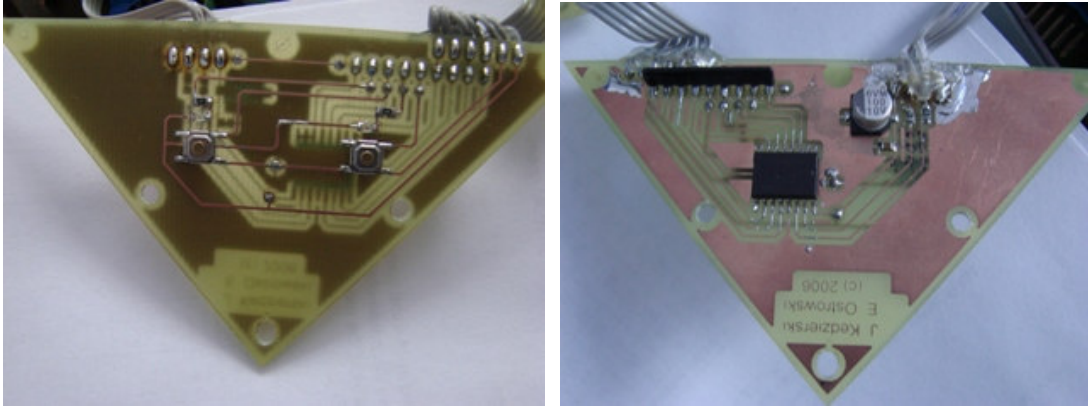
Rysunek 5 – Rozmieszczenie elementów na górnej warstwie.



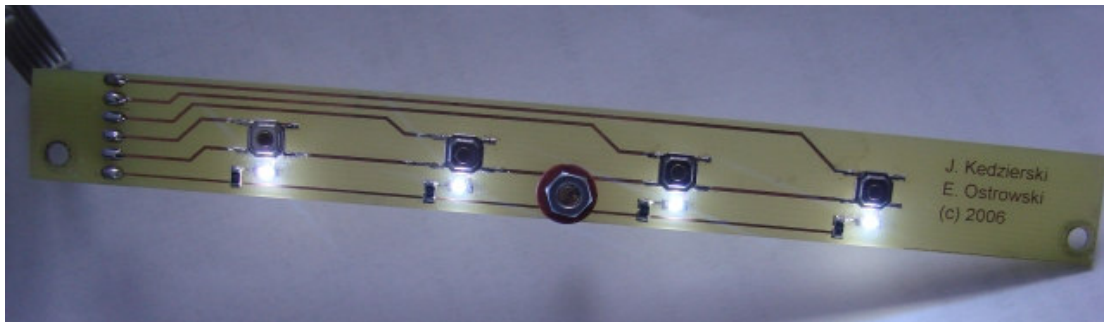
Rysunek 6 – Rozmieszczenie elementów na dolnej warstwie.

5 Klawiatura

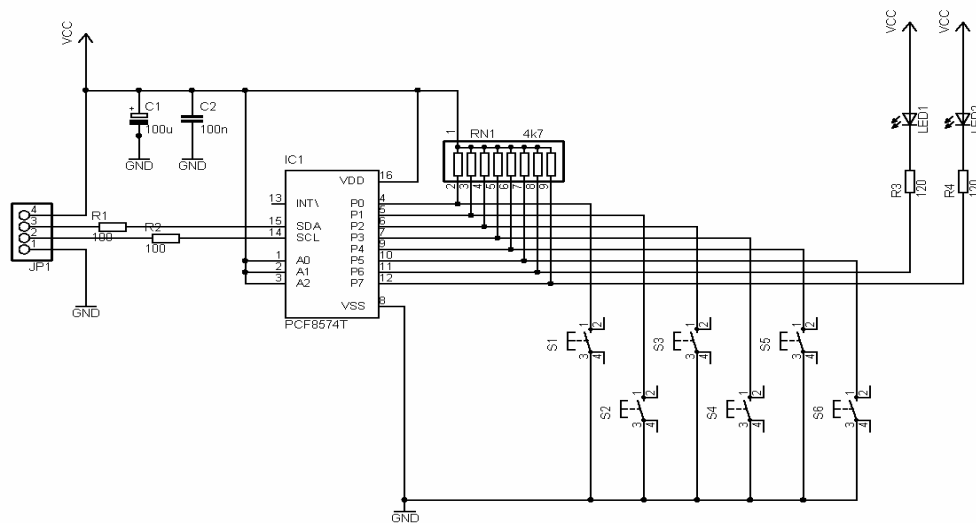
Dodatkowo w robocie przewidziano 6-cio klawiszową klawiaturę [Fot. 11,12]. Użyto w niej układ PCF8574 komunikujący się po magistrali I²C. Dzięki niej możliwa jest zmiana parametrów pracy robota takich jak: profile prędkości czy też nastawy regulatorów PID. Niezbędna okazała się również przy poruszaniu się po menu w oprogramowaniu robota.



Fot. 11 – Widok płytki przedniej.



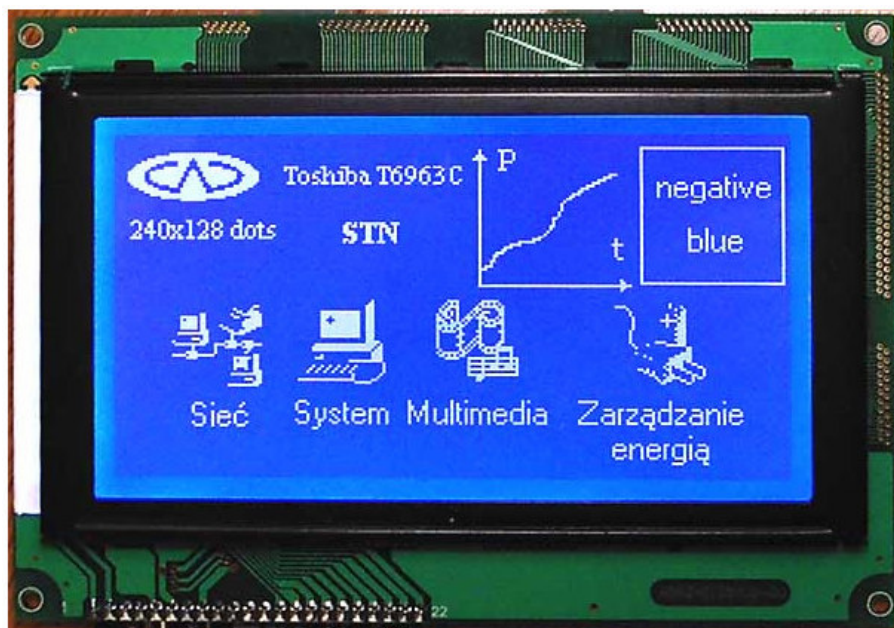
Fot. 12 – Widok płytki bocznej.



Rys. 7 – Schemat klawiatury.

6 Wyświetlacz LCD

Do wizualizacji danych użyto wyświetlacza graficznego o rozdzielczości 240x128 [Fot. 13]. Wyświetlacz był wyposażony kontroler firmy Toshiba T6963C [8], który posiada już wbudowany generator znaków. Podświetlenie zastosowane w tym modelu to kilka sztuk białych diod LED. Wyjścia podłączono tuż na magistralę danych mikrokontrolera co przedstawiają schematy płyty głównej. Dokładny opis sterowania wyświetlaczem znajduje się w załączniku.



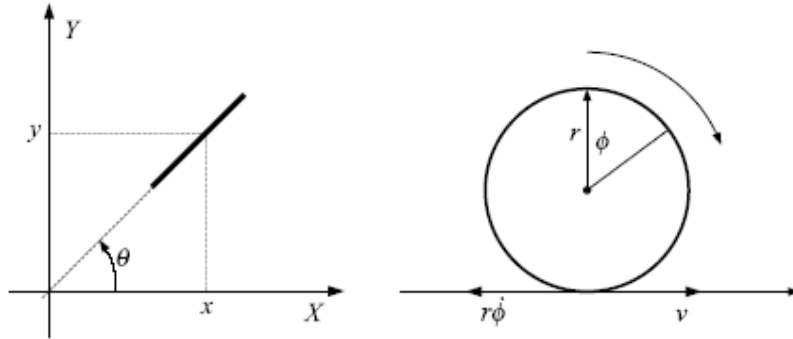
Fot. 13 – Widok wyświetlacza.

No.	Symbol	Function
1	A	Backlight supply voltage(+5V)
2	K	Backlight supply voltage(-)
3	VOUT	Negative voltage output
4	V0	Negative voltage input
5	PD	Control pin for oscillation
6	GND	VSS(0V)
7	VDD	Power supply for Logic(+5.0V)
8	VEE	LCD supply voltage
9	/WR	Write signal
10	/RD	Read signal
11	/CE	Chip selection signal
12	C/D	Data or Instruction select signal(H: data register L:instruction register)
13	/RST	Reset signal
14-21	DB0-DB7	Data bus line
22	FS	Font selection terminal

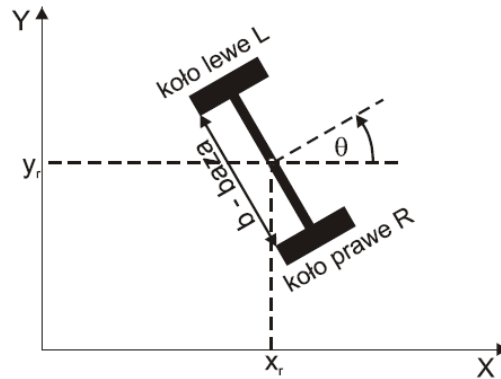
7 Sterowanie

7.1 Odometria

Zbudowana na potrzeby tego projektu platforma mobilna jest robotem unicycle (jednokołowy) [Rys. 8,9]. W rzeczywistości jest to robot dwukołowy jednakże sterowanie przypomina nieco sterowanie jednym kołem. Zadaje się prędkość liniową platformy oraz prędkość skrętu. Układ ten jest ciekawy z tego względu, że umożliwia robotowi zmianę orientacji w miejscu, co pozwala mu na łatwiejsze pokonywanie skomplikowanych przeszkód.



Rys. 8 – Opis prędkości sterujących.



Rys. 9 – Zmiana orientacji.

Na każdym kole zainstalowano enkoder który dostarczał do mikrokontrolera informacji o aktualnej prędkości koła. Dzięki temu możliwe było zrealizowanie regulacji PID. Wyliczenie położenia oraz orientacji na podstawie odometrii polega na doliczaniu przyrostów przebytej drogi. Enkodery zamontowane na osiach silników zliczają ilość impulsów N_P i N_L odpowiadającą katom obrotu kół. Konwersji na przyrosty drogi kół można dokonać za pomocą wzorów:

$$\Delta U_L = C_m N_L \quad \Delta U_P = C_m N_P$$

Współczynnik C_m odpowiada za przeliczenie wskazania enkodera na przemieszczenie liniowe:

$$C_m = \frac{\pi 2r}{nC_e}$$

gdzie

C_m - rozdzielczość enkodera

$2r$ - nominalna średnica kół [Rys. 8]

n - np. przekładnia

Przyrost drogi środka robota wynosi:

$$\Delta U = \frac{\Delta U_L + \Delta U_R}{2}$$

Przyrost orientacji wyznacza się przy pomocy wzoru:

$$\Delta \theta = \frac{\Delta U_L - \Delta U_R}{b}$$

nową orientację ze wzoru:

$$\theta_i = \theta_{i-1} + \Delta \theta$$

przyczym:

b - baza robota [Rys. 9]

Zmiany położenia w układzie kartezjańskim możemy obliczać na podstawie wzorów:

$$\Delta x = \Delta U \cos \theta$$

$$\Delta y = \Delta U \sin \theta$$

nową pozycję ze wzorów:

$$x_i = x_{i-1} + \Delta x$$

$$y_i = y_{i-1} + \Delta y$$

7.2 Regulator PID. Algorytm sterowania wygląda następująco [9]:

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt}$$

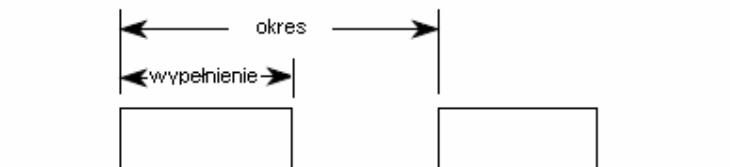
Regulator ten zrealizowano w następujący sposób:

```
//Zmienne pomocnicze:
signed int epsilon=0;
signed int czlon_P=0;
signed int czlon_I=0;
signed int czlon_D=0;
// badamy różnice między wartością zadana a rzeczywistą
epsilon=(ster.V_P-stan.V_W_P);
//zapisujemy to jako epsilon czyli błąd
czlon_P=epsilon;
// teraz korzystając ze zmiennej np. globalnej epsilon_suma dodajemy do niej wartość błędu
epsilon_suma+=epsilon;
// należy jednak ograniczyć całkowanie naszego błędu do np +/- 100
// tak aby pomnożone przez Ki nie dało więcej niż max. sterowanie
if (epsilon_suma>100) epsilon_suma=100;
if (epsilon_suma<-100) epsilon_suma=-100;
// I zapisujemy to jako czlon_I
czlon_I= epsilon_suma;
// odejmując od obecnej wartości błędu poprzednia wartość błędu uzyskujemy różniczkę
czlon_D=epsilon-epsilon_old;
// sumujemy wszystkie człony mnożąc je przez odpowiadające im wzmocnienia
sterowanie= Kp*czlon_P+ Kd*czlon_D+ Ki*czlon_I;
// ograniczamy sterowanie tak aby nie wykraczało po za możliwie największe
if (sterowanie >1000) sterowanie =1000;
if (sterowanie <-1000) sterowanie =-1000;
// zapamiętujemy wartość błędu
epsilon_old= epsilon;
Zadaj_PWM(sterowanie); //opis funkcji PWM znajduje się poniżej
```

Naturalnie zmienne *epsilon_suma*, *epsilon_old*, *Kp*, *Kd*, *Ki* powinny byc zmiennymi albo typu static, albo zmiennymi globalnymi. Wygodnie jest je również umieścić w jednej strukturze.

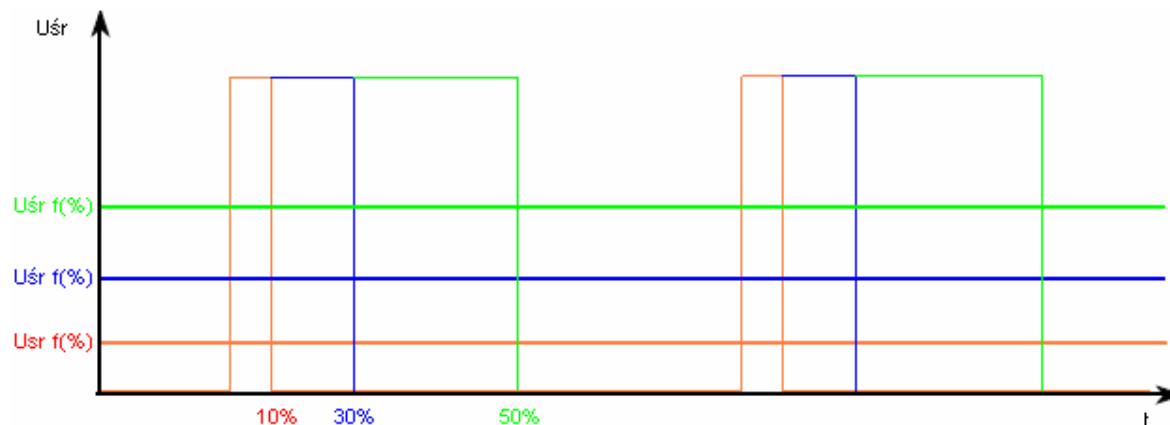
7.3 Sterowanie PWM

Jest to generator sygnału prostokątnego z możliwością programowej zmiany częstotliwości (okresu) i wypełnienia [Rys. 10]. Sygnał generowany wyprowadzany jest na zewnętrzne linie (piny) mikrokontrolera.



Rys. 10 – Generowany sygnał PWM.

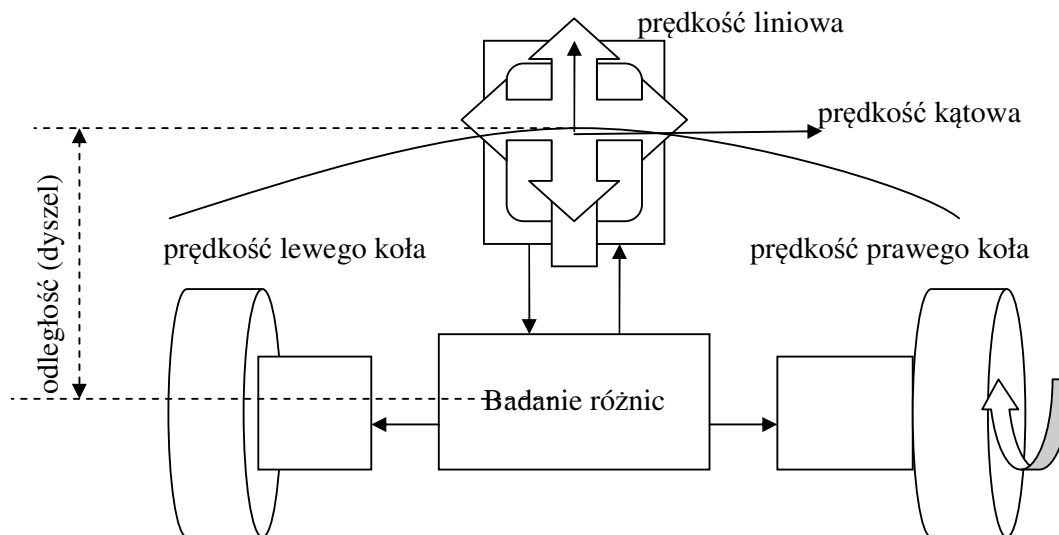
Sygnal ten można wykorzystać do sterowania mostków H czyli układami sterującymi napędy. Cewki silników wprawiających robota w ruch mają głównie charakter indukcyjny, a zatem całkują (uśredniają) napięcie zasilające. Dokładniej przedstawia to wykres poniżej [Rys. 11]. Częstotliwość PWM dla sterownia silnikami wynosiła 1kHz.



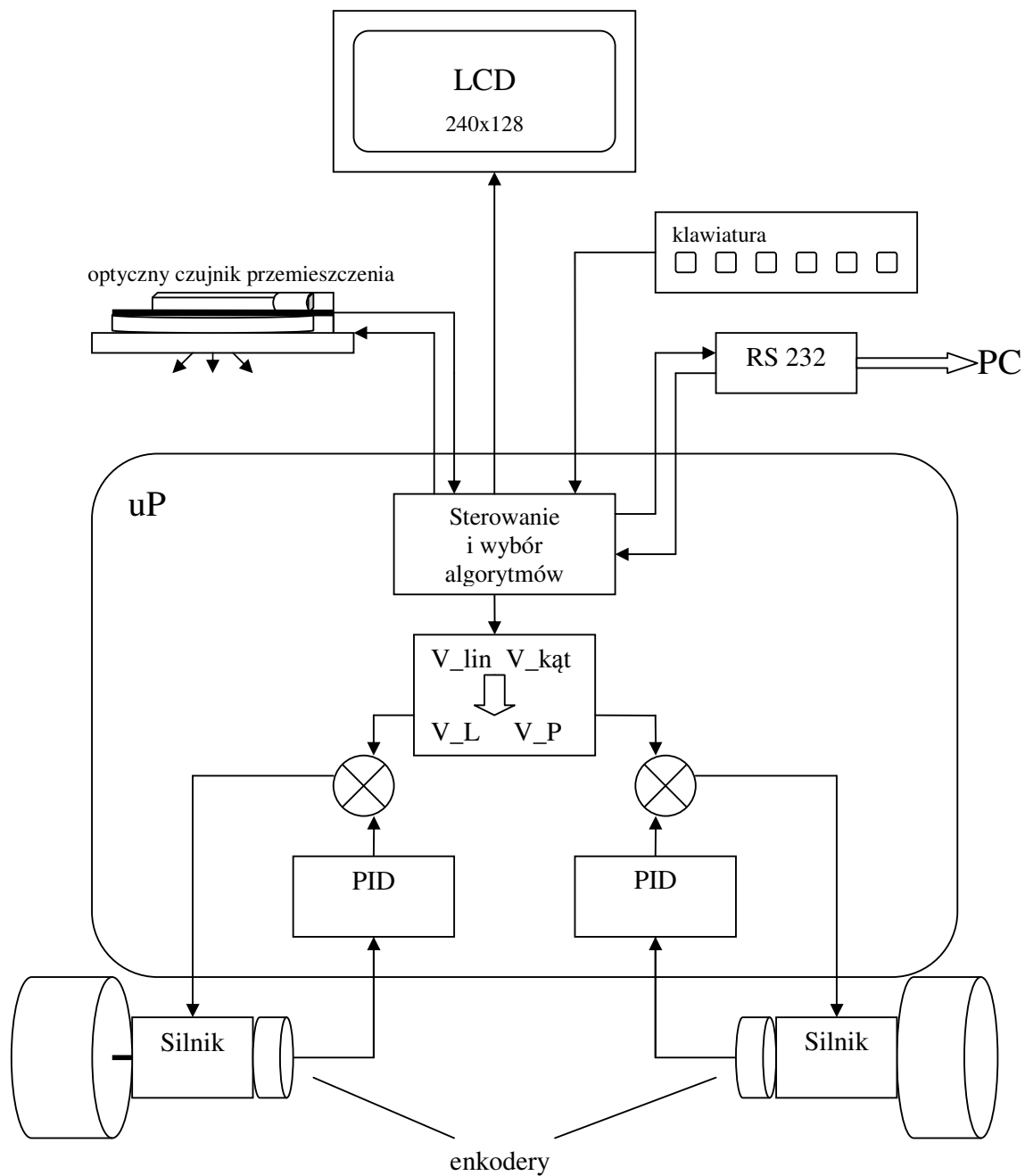
Rys. 11 – Wartości średnie napięcia w funkcji wypełnienia.

7.4 Wykrywanie błędu ruchu

Dzięki zastosowaniu optycznego czujnika przemieszczenia można było wnioskować o prawidłowym kierunku jazdy robota. Jeżeli prędkości zadane znacznie różniły się od prędkości sczytanych przez czujnik stwierdzano błąd ruchu [Rys 12].



Rys. 12 odzworowaniem prędkości kątowej i liniowej



Rys. 13 – Schemat blokowy.

8 Oprogramowanie robota

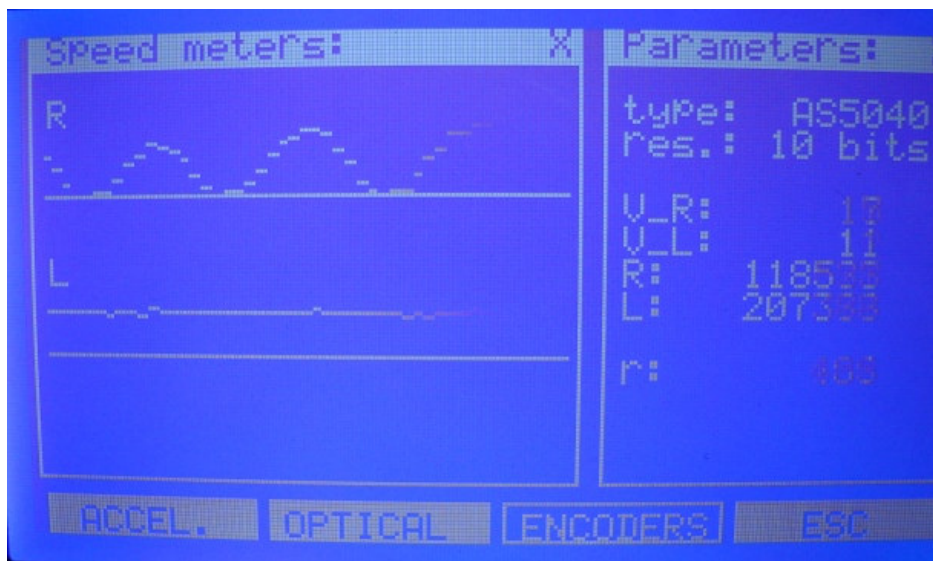
Oprogramowanie robota przygotowano tak, aby łatwo i bez wykonywania skomplikowanych obliczeń można było zaobserwować wyniki pomiarów. Pomiary zarówno z enkoderów jak i z czujnika przemieszczenia wizualizowano w postaci przebiegów i wektorów prędkości. Łatwe o obsłudze menu robota pozwala na szybkie przełączanie się pomiędzy ekranami wizualizacyjnymi oraz pozostałymi funkcjami robota. Przewidziano również interfejs pozwalający dokonywać ustawień trybów pracy robota oraz parametrów np regulatora obrotów. Tryb konsoli służy do sprawdzania poprawności otrzymywanych komunikatów z komputera PC.



Fot. 14 – Widok płyty górnej – ekran kontroli zadanych i rzeczywistych wartości.



Fot. 15 – Menu główne.



Fot. 16 – Wizualizacja prędkości kół – oscylacje prawego koła.

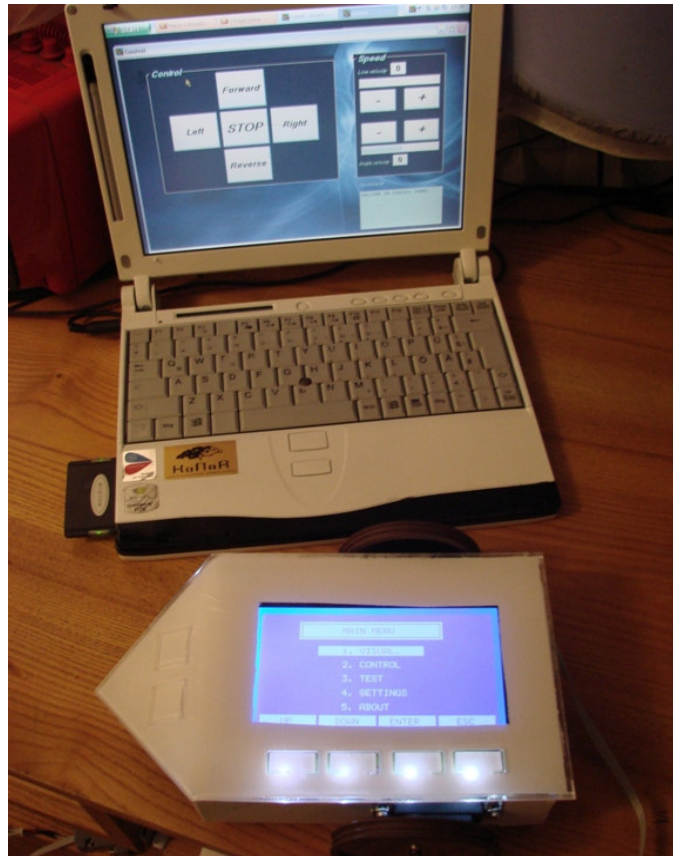


Fot. 17 – Wizualizacja wektora prędkości z czujnika przemieszczenia.

9 Sterowanie za pomocą komputera PC

Do sterowania robotem napisano wygodny i bardzo prosty w obsłudze interfejs. Program ten zainstalowano na komputerze wyposażonym w panel dotykowy. Dzięki temu sterowanie odbywało się w bardzo efektywny i wygodny sposób. Komunikacja odbywała się po

interfejsie RS232. Każdy komunikat wysłany z komputera jest potwierdzany przez mikrokontroler co daje stabilne połączenie.



Fot. 18 – Stanowisko do pracy z robotem.



Fot. 19 – Panel sterujący.

10 Literatura

- [1] KĘDZIERSKI J. OSTROWSKI E. "Optyczny czujnik przemieszczenia"
<http://konar.ict.pwr.wroc.pl/module.php?op=download&cmd=click&id=58>
- [2] KĘDZIERSKI J. OSTROWSKI E. "Opis enkodera magnetycznego AS5040"
<http://konar.ict.pwr.wroc.pl/module.php?op=download&cmd=click&id=69>
- [3] www.austriamicrosystems.com
- [4] AS4050 10 bit programmable magnetic rotary encoder, Data sheet, austriamicrosystems, 2004.
- [5] www.st.com
- [6] L298 Dual Full-Bridge motor driver, SGS-THOMSON Microelectronics, 1995
- [7] WNUK M. Moduł z Mikrokontrolerem MC68332, Raport ICT serii 07/2004, Wrocław, 2004
- [8] KĘDZIERSKI J. OSTROWSKI E. "Sterowanie wyświetlaczem graficznym z kontrolerem firmy Toshiba T6963C "
www.konar.ict.pwr.wroc.pl
- [9] ARTUR HONISCH Praca Magisterska Dyplomowa. „RoboMouse – mały robot mobilny do celów badawczych”