

Gizmo

v2

Arkadiusz Materek

24 czerwca 2007

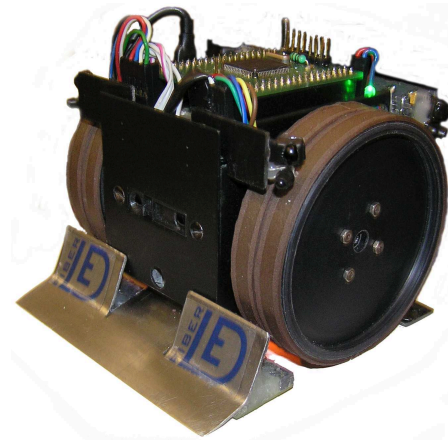
Spis treści

1	Wstęp	3
2	Konstrukcja	5
2.1	Mechanika	5
2.2	Sensory i elektronika	7
2.2.1	Mikrokontroler	7
2.2.2	Schemat elektroniki	9
2.2.3	Czujniki białej linii	11
2.2.4	Czujnik GP2D12 firmy Sharp	12
2.2.5	Czujniki zbliżeniowe IR	14
2.2.6	Enkodery	16
2.2.7	Końcówki mocy	18
3	Implementacja	19
3.1	Nawigacja przyrostowa	19
3.1.1	Kinematyka robota	19
3.1.2	Implementacja na mikrokontrolerze	20
3.2	Sterowanie ruchem	20
3.3	Zachowania	21
3.3.1	Wprowadzenie	21
3.3.2	Szukanie przeciwnika	23
3.3.3	Reakcja robota na wykrycie białej linii	23
3.3.4	Reakcja robota na wykrycie przeciwnika	25
3.3.5	Reakcja robota na wykrycie przeciwnika i białej linii	26
3.3.6	Reakcja robota na zgubienie przeciwnika	26
3.3.7	Wykrywanie kontaktu z robotem przeciwnika	26
3.3.8	Zachowania defensywne podczas kontaktu z robotem	26

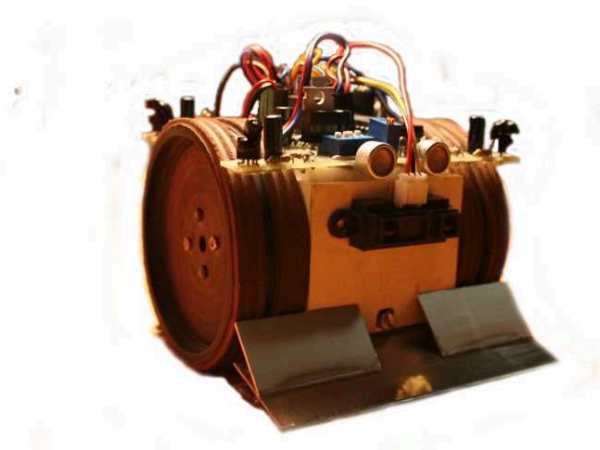
3.4 Środowisko programistyczne	27
3.5 Program diagnostyczny	27
Spis rysunków	28
Spis tabel	30
Literatura	30

1 Wstęp

Robot Gizmo V2 został skonstruowany pod kątem startu w zawodach robotów mini-sumo, co narzuciło pewne ograniczenia i wymuszenia konstrukcyjne. Roboty mini-sumo, zgodnie z regulaminem mają ograniczoną szerokość i długość do 100mm oraz masę do 500g. Walka odbywa się na dohyo, ma ono średnicę 77cm, jest czarne i ma na brzegach namalowaną białą linię o szerokości 2cm.



Rysunek 1: Zdjęcie robota Gizmo V2



Rysunek 2: Zdjęcie robota Gizmo

Gizmo V2 jest ulepszoną wersją mojego poprzedniego robota mini-sumo Gizmo¹. Nowy robot jest wyposażony w wydajny 16bit procesor, co pozwala na implementację dość złożonych obliczeniowo algorytmów. Ze starej wersji robota

¹Zwycięzca III Ogólnopolskiego Turnieju Robotów Mini-sumo we Wrocławiu

zaczepnięto sprawdzone rozwiązania tj. podstawowe wymiary, dobrze sprawujący się napęd (serwomechanizmy), ruchomy klin, zabezpieczenie przed podważeniem przez przeciwnika oraz nawigację przyrostową. Wylimitowano również wady poprzedniej konstrukcji: źle położony środek ciężkości, zwiększony zakres podważenia bez utraty przyczepności kół, wygładzony (bez wystających elementów) przód robota. Gizmo V2 początkowo był wyposażony w myszkę optyczną umieszczoną pod robotem z przesunięciem względem osi obrotu robota, jednak takie rozwiązanie nie dawało zadowalających rezultatów (duże błędy przy obliczaniu orientacji). Powrócono więc do poprzedniego rozwiązania - dwóch toczących się po podłożu enkoderów, z tą różnicą, iż zastosowano magnetyczne enkodery o rozdzielczości 1024, a nie optyczne z myszki od komputera, które mają ok. 5 razy mniejszą rozdzielczość. W oprogramowaniu zostały dopracowane algorytmy, które były testowane już w walce z prawdziwym przeciwnikiem (starym robotem). Ulepszone zostały również metody zbierania i interpretacji pomiarów z czujników oraz został napisany program diagnostyczny na PC łączący się z robotem przez port RS232.

Robot został zaprojektowany według założeń:

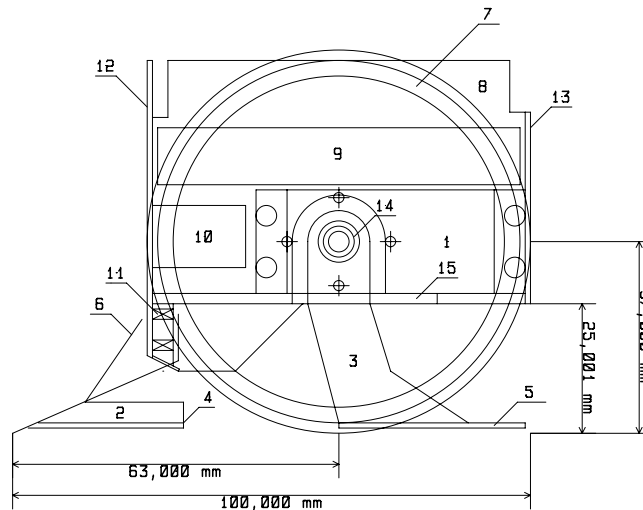
- Konstrukcja:
 - wymiary i masa zgodne z wymaganiami zawartymi w regulaminie Minisumo,
 - podwozie klasy (2,0) na dwóch kołach z oponami o dobrej przyczepności,
 - napęd: dwa niezależnie sterowane silniki elektryczne prądu stałego (serwomechanizmy modelarskie),
 - klin na przegubie zapewniający przyleganie go do podłoża,
 - zabezpieczenie przed podważeniem robota przez przeciwnika

- Czujniki
 - cztery optyczne czujniki białej linii w postaci diody na podczerwień i fototranzystora,
 - dalmierz Sharp do wykrywania przeciwnika z przodu robota,
 - czujniki na podczerwień z modulowaną częstotliwością do wykrywania przeciwnika z boków lub z tyłu robota,

- enkodery toczące się pod robotem, aby wyeliminować błędy spowodowane poślizgami kół oraz określić czy robot jest pchany czy pcha robota przeciwnika.
- Zachowanie robota:
 - reakcja na wykrycie białej linii
 - szukanie przeciwnika
 - atak przodem na wykryty cel
 - próba uniku gdy robot przegrywa w zwarciu
 - próby przeważenia sytuacji na swoją korzyść w sytuacjach remisowych
 - możliwość wyłączenia przez użytkownika zachowań, które mogą być zagrożeniem w walce z niektórymi robotami np. uniku (w przypadku gdy robot jest szybszy).

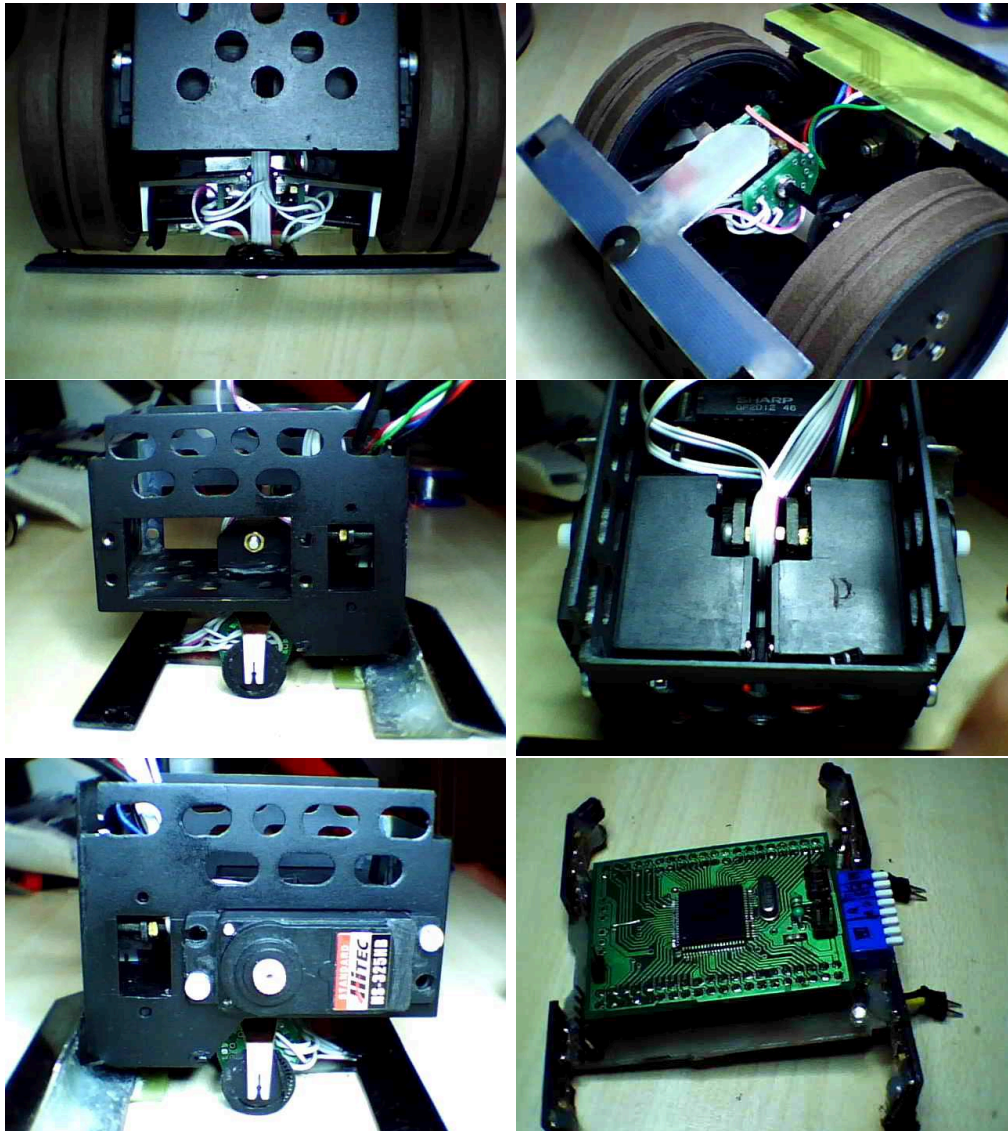
2 Konstrukcja

2.1 Mechanika



Rysunek 3: Schemat konstrukcji robota Gizmo V2

Poglądowy schemat konstrukcji robota przedstawiony jest na rysunku 3. Konstrukcję wykonano z płytek laminatowych ażurowanych w celu zmniejszenia masy,

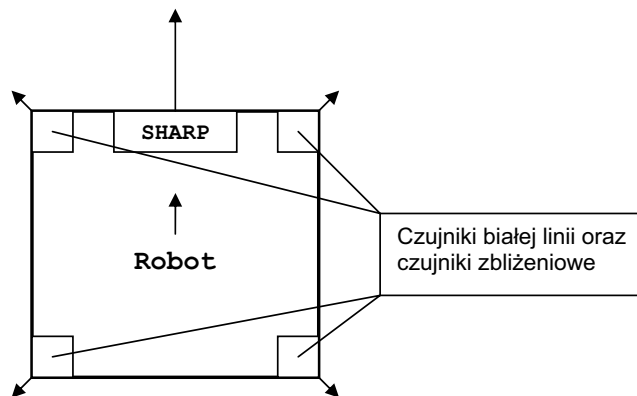


Rysunek 4: Zdjęcia konstrukcji robota Gizmo V2

boki (8) oraz podłoga (15) są grubości 1.5mm natomiast przód (12) i tył (13) grubości 1mm. Klin (6) wykonano ze stali nierdzewnej grubości 1mm mocowany jest do konstrukcji przez łożysko (11), co zapewnia możliwość ruchu klinu na boki. Od spodu klinu przyklejona jest płytko laminatowa (4) grubości 1mm, na której zamocowane są dwa przednie czujniki białej linii, dodatkowo klin jest dociążony ołowiem (2) w celu przeniesienia środka ciężkości do przodu. Płyta laminatowa (5) grubości 1mm z tylnymi czujnikami białej linii, przyklejona jest do wspornika (3), wykonanego z laminatu grubości 3mm, który przez łożysko kulkowe (14) montowany jest do podłogi (15). Dodatkowo do wspornika (3) na zawiasach montowane

są enkodery (widoczne na rysunku 4). Docisk enkoderów do podłoża zapewnia gumka ściąająca je do siebie. Rozwiązanie takie zapewnia możliwość podważenia robota o ok. 30° bez utraty przyczepności kół oraz zapewnia, że enkodery cały czas się toczą po podłożu. Kółka (7) o średnicy 70mm i grubości 20mm wytoczone z plastyku mocowane są bezpośrednio do przerobionych serwomechanizmów (1), serwomechanizmy przykręcone są do boków (8).

Jako opony wykorzystano uszczelki do okien, ze względu na ich dobrą przyczepność. Wewnątrz zamocowany jest dalmierz (10). Na serwomechanizmach leży zasilanie (9) - pakiet siedmiu akumulatorów rozmiaru AAA, nad akumulatorami znajduje się miejsce na elektronikę. Czujniki zbliżeniowe są zamocowane na płytce z elektroniką widoczną na rysunku 4. Napęd robota stanowią przerobione serwomechanizmy firmy HiTec HS-325HB wyposażone w łożysko kulkowe. Z serwomechanizmów usunięto ogranicznik obrotu oraz całą elektronikę. Schemat roz-



Rysunek 5: Schemat rozmieszczenia sensorów w robocie Gizmo V2

mieszczenia czujników w robocie przedstawiony jest na rysunku [2]. Robot posiada cztery czujniki białej linii, cztery czujniki zbliżeniowe na podczerwień z modulowaną częstotliwością dające sygnał binarny oraz dalmierz firmy Sharp GP2D12 o zakresie odległości 10-80cm.

2.2 Sensory i elektronika

2.2.1 Mikrokontroler

Mózgiem robota jest mikrokontroler MC9S12A64 firmy Freescale. Rodzina 16bit mikrokontrolerów S12 jest następcą 8bit mikrokontrolera serii HC11. Seria S12

cehuje się wysoką wydajnością oraz dużą ilością zintegrowanych układów wejścia/wyjścia (timerów, przetworników AC, portów, interfejsów szeregowych). W stosunku do poprzednika rozszerzono zestaw instrukcji procesora (m.in. interpolację liniową, fuzzy logic) oraz wprowadzono interfejs uruchomieniowy BDM, który pozwala na "debugowanie" programu w układzie docelowym. Dzięki ww zaletom mikrokontrolery S12 są często stosowane w robotyce jak i w przemyśle samochodowym. W robocie wykorzystano moduł z wymienionym wyżej mikrokontrolerem opisanym w [4].

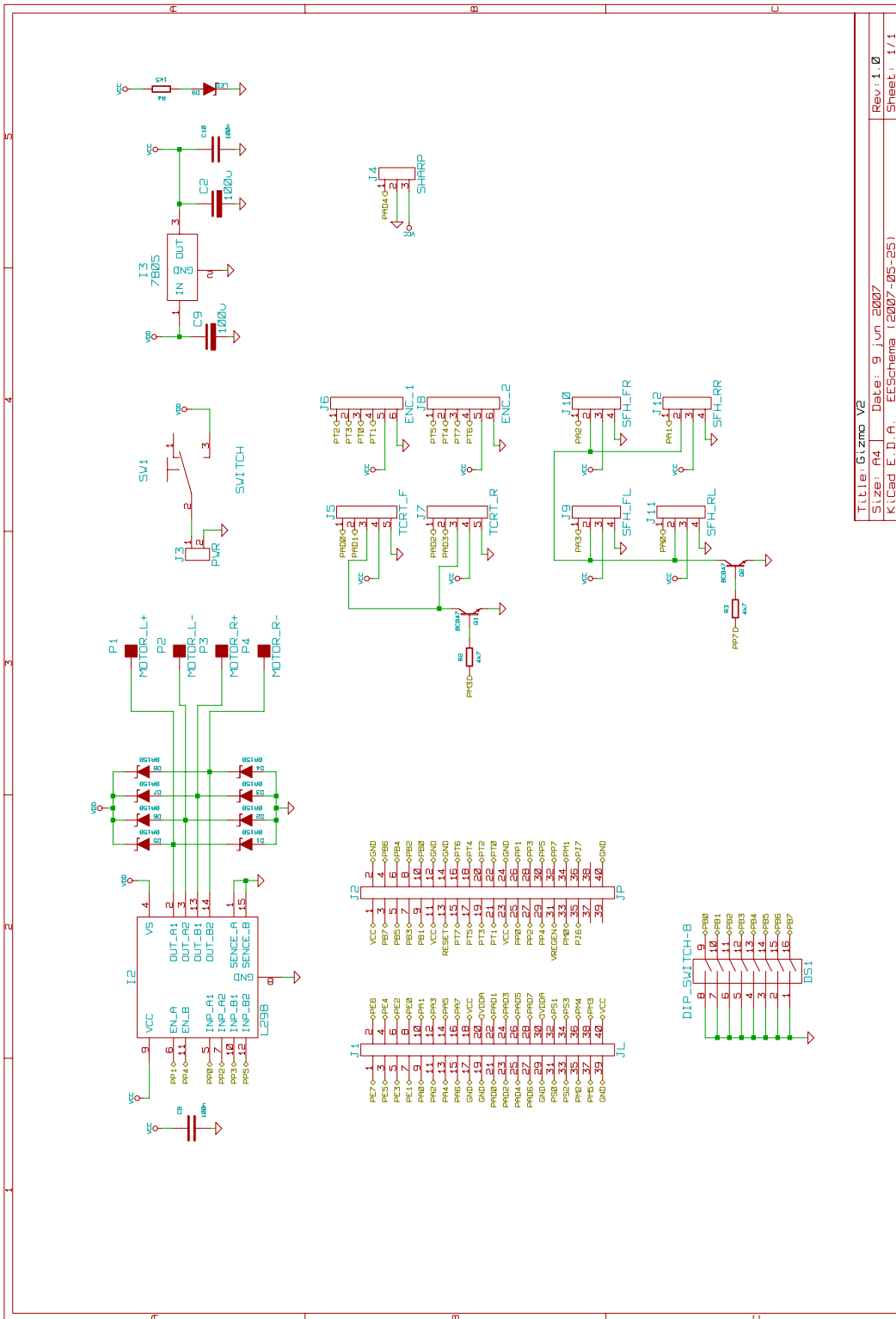
Podstawowe własności mikrokontrolera:

- 16 bit jądro HCS12
 - kompatybilność z MC68HC11
 - Kolejka instrukcji
 - Rozbudowane tryby adresowania
 - 20 bit jednostka arytmetyczno logiczna (ALU)
 - instrukcje Fuzzy Logic
- Moduł CRG
 - Pętla fazowa (PLL)
 - Generator zegara systemowego i monitorowanie jego stanu
 - Watchdog (COP)
 - System resetu wyzwalany przez wyjątki podczas działania programu
 - Przerwanie czasu rzeczywistego (RTI)
- 8 bit i 4bit port przerwań zewnętrznych z cyfrową filtracją oraz konfiguracją zbocza wyzwalania
- Pamięć
 - 64KB pamięci FLASH
 - 1KB pamięci EEPROM
 - 4KB pamięci SRAM
- 10bit przetwornik analogowo cyfrowy

- 1Mbps CAN
- 16bit ośmio kanałowy timer ETC z funkcjami Input Capture i Output Compare, oraz cztery 8bit lub 2 16bit liczniki impulsów
- Puls wide modulation PWM
 - Ośiem 8bit lub cztery 16bit kanały
 - Osobna konfiguracja (okres i wypełnienie dla każdego kanału)
 - Tryb symetryczny *Center align* i asymetryczny *Left align*
 - Wejście wyłączania awaryjnego
- Interfejsy szeregowo
 - Dwa moduły asynchronicznej transmisji SCI
 - Moduł transmisji synchronicznej SPI
 - Moduł I2C
- Wbudowany regulator napięcia 2.5V
- Interfejs uruchomieniowy BDM
- Maksymalna częstotliwość pracy procesora 50MHz, magistrali 25MHz
- Obudowa 80 pin oraz 112 pin

2.2.2 Schemat elektroniki

Przy opracowywaniu schematu elektroniki dążono do zredukowania liczby elementów, ze względu na ograniczoną dostępność miejsca w robocie. Wzięto też pod uwagę dostępność podzespołów. W rezultacie otrzymano schemat przedstawiony na rysunku 3. Znajdują się tam złącza do dołączenia modułu mikrokontrolera, czujników, stabilizator napięcia i podwójny mostek-H do sterowania silnikami.



Rysunek 6: Schemat elektroniki Gizmo V2

odrzućcane, a z pozostałych obliczana jest średnia arytmetyczna. Następnie, jeśli otrzymana wartość jest większa od górnego progu napięcia to wartość wyjściowa czujnika jest równa 1, jeśli mniejsza od dolnego progu to 0, a w przeciwnym wypadku pozostawiana jest bez zmiany.

Powyższy sposób przetwarzania danych zapewnia histerezę, eliminację chwilowych zakłóceń (np. błysków fleszy) oraz eliminowana jest składowa stała zewnętrznego oświetlenia.

Opisywane czujniki sprawują się bardzo dobrze, odczyty z nich są pewne w zakresie odległości transoptora od obiektu 0.5-3mm. W zależności od podłoża należy dobrać eksperymentalnie progi histerezy.

Alternatywnym sposobem interpretacji danych może być zamiana bezwzględnych progów histerezy na względne, czyli o ile procent maleje napięcie przy włączonej diodzie IR w stosunku do wyłączonej. Taki sposób postępowania jeszcze bardziej uodporniłby czujniki na zewnętrzne źródła światła, jednak w przypadku zastosowania w robocie mini-sumo nie jest to niezbędne.

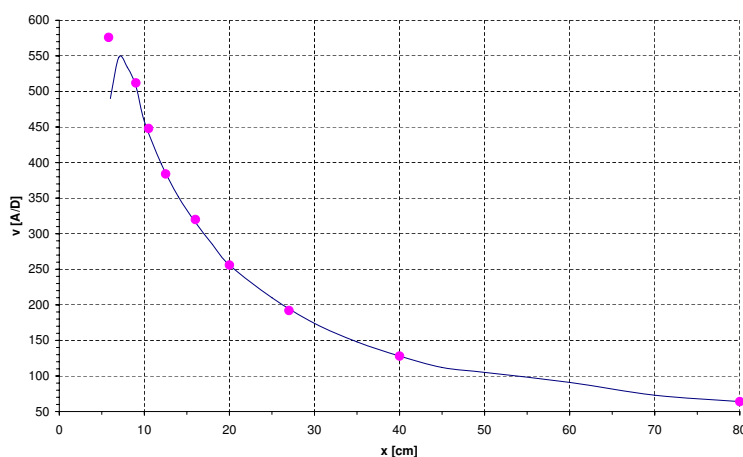
2.2.4 Czujnik GP2D12 firmy Sharp

Firma Sharp produkuje dalmierze serii GP2DXX o zakresie pomiarowym od 30cm do 150cm. Działają na zasadzie pomiaru kąta wyemitowanej wiązki światła podczerwonego po odbiciu od obiektu. Cechują się bardzo małym wpływem koloru oraz refleksyjności obiektu na wynik pomiaru. Wynik pomiaru jest podawany analogowo lub binarnie. Wadami tych czujników są: stosunkowo duża strefa martwa, mała częstotliwość pomiarów (ok. 20-25Hz), nieliniowa charakterystyka wyjścia. Parametry poszczególnych modeli przedstawiono w tabeli 1, a charakterystykę wyjścia na rysunku 8.

Obsługa przez μC : Do pomiaru napięcia wykorzystano wbudowany w mikrokontroler 10bit przetwornik A/C z napięciem referencyjnym 5V. W celu poprawnego przetwarzania sygnału analogowego zebrano charakterystykę wyjścia dalmierza, przedstawioną na rysunku 8. Wyznaczono odległości, dla których wartość z przetwornika A/C zmienia się co 64 jednostki. Wartości pośrednie odległości obliczane są przez interpolację liniową wartości w tabeli 2 (również zaznaczone na wykresie 8). Przy implementacji interpolacji liniowej wykorzystano gotową instrukcję mikrokontrolera ETBL.

Model	zasięg	Wyjście
GP2D150	15cm z histerezą 2.5cm	binarne
GP2D120	4-30cm	analogowe
GP2D15	24cm z histerezą +- 3cm	binarne
GP2D12	10-80cm	analogowe
GP2DY0A02	20-150cm	analogowe

Tabela 1: Porównanie parametrów dalmierzy firmy SHARP



Rysunek 8: Charakterystyka wyjścia czujnika GP2D12

v [A/C]	0	64	128	192	256	320	384	448	512	576
x [mm]	1600	800	400	270	200	160	125	105	90	58

Tabela 2: Tabela pomiarów napięcia wyjściowego czujnika GP2D12

Przykład implementacji funkcji zwracającej odległość na podstawie odczytu napięcia z czujnika GP2D12:

```
const int gp2d12[] = { 1600, 800, 400, 270, 200,
                      160, 125, 105, 90, 58 };
```

```
int sharp_gp2d12( int adc_value ){
    asm{
```

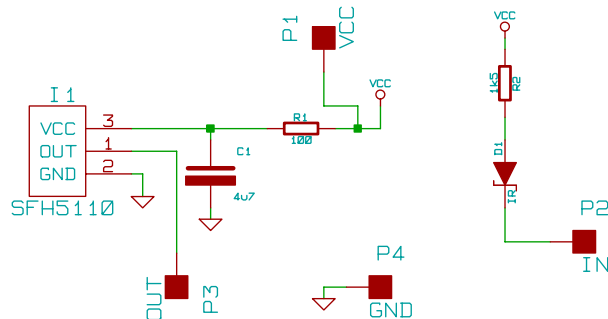
```

LDX      #64          ; X = 64
IDIV     ; X = D/64, D = D%64
LSLB     ; B=B*2
LSLB     ; B=B*2
XGDX     ; zamiana X<->D
LSLD     ; przesuniecie D w lewo
XGDX     ; zamiana X<->D
LEAX     gp2d12, X   ; do X adres elementu i
ETBL     0,X         ; przyblizenie liniowe
}
}

```

2.2.5 Czujniki zblizeniowe IR

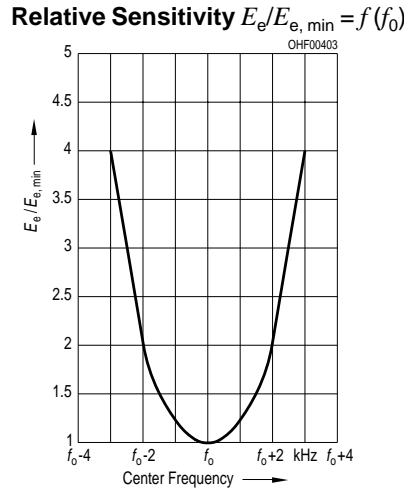
Zadaniem czujnika jest dostarczenie binarnej informacji czy obiekt znajduje się w zasięgu do ok 20cm, do tego celu wykorzystano odbiorniki podczerwieni od pilotów TV (SFH5110). Schemat został przedstawiony na rysunku 9. Układ



Rysunek 9: Schemat czujnika zblizeniowego IR

SFH5110-38 zasilany jest napięciem 5V, ma on wewnętrzny demodulator częstotliwości 38kHz, oraz wyjście w standardzie TTL. Konieczne jest zasilanie tego układu przez filtr dolnoprzepustowy R1-C1. Nadajnikiem jest dioda D1, której prąd został ograniczony przez rezystor R2 do ok. 2.5mA³. Sygnał 38kHz generuje mikrokontroler a sterowanie prądem wszystkich diod od wszystkich czujników tego

³ większy prąd sprawia, że układ jest zbyt czuły



Rysunek 10: Czulość układu SFH5110 w zależności od częstotliwości

typu odbywa się przez wspólny tranzystor. Izolację optyczną między nadajnikiem i odbiornikiem zapewnia czarna koszulka termokurczliwa założona na diodę oraz układ SFH5110 (z otworem w miejscu odbiornika)

Zasada działania: Dioda D1 miga z częstotliwością 38kHz, jeśli światło zostanie odbite i wróci do odbiornika to na wyjściu układu I1 pojawi się stan niski.

Obsługa przez μC : Mikrokontroler generuje sygnał 38kHz. Pomiar z każdego czujnika odczytywane są co 1ms, gdy na wyjściu czujnika jest stan niski to zwiększany jest licznik dla danego czujnika o 1. Po zebraniu dziesięciu pomiarów, jeśli wartość licznika dla danego czujnika przekracza górny próg (np. 9) to jako odczyt z czujnika ustawiana jest wartość 1, jeśli nie przekracza dolnego progu (np. 3) to 0, a w przeciwnym pozostaje bez zmian. Jest też możliwość regulacji czulości przez zmianę częstotliwości nadawanego sygnału⁴. Charakterystykę zależności czulości od częstotliwości dla układu SFH5110 przedstawiono na rysunku 10

Opisany powyżej detektor spisuje się bardzo dobrze. Duży wpływ na dobre działanie ma histereza, gdy obiekt znajduje się na granicy zasięgu. Wadą opisywanego sensora jest wpływ koloru oraz refleksyjności na zasięg (ok. 2-3 razy mniejszy dla ciemnych matowych obiektów w porównaniu z białymi)

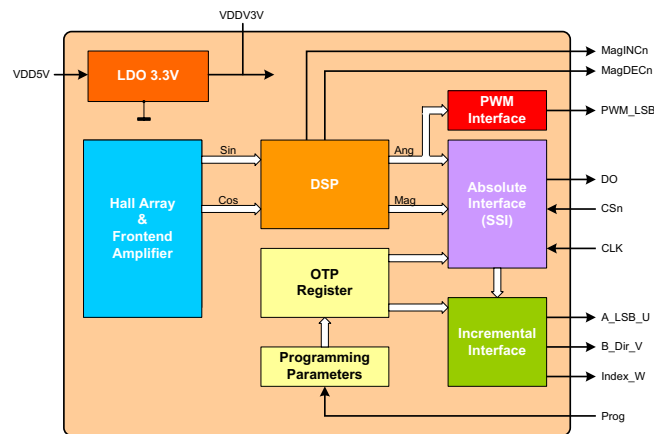
⁴ jeśli przeciwnik korzystał z podobnych czujników to była generowana cz. o 2kHz większa

2.2.6 Enkodery

Jako enkodery wykorzystano układy AS5040 firmy Austriamicrosystems. Aby zapewnić poprawną pracę układu niezbędny jest magnes o średnicy 4-6mm umieszczony nad (lub pod) obudową układu w odległości 0.5-1.8mm. Należy jeszcze zwrócić uwagę aby oś obrotu magnesu była dokładnie na osi układu (producent sugeruje tolerancję 0.25mm). Szczegółowe informacje znajdują się w nocie producenta [1].

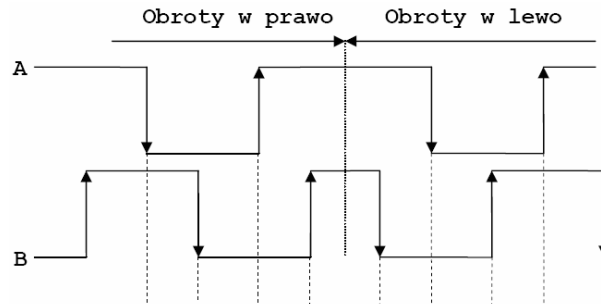
Podstawowe własności układu:

- Programowalna rozdzielczość 7, 8, 9, 10 bit
- Pomiar bezwzględny kąta (wyjście PWM oraz SPI)
- Wyjścia inkrementacyjne: kwadraturowe, Impuls-kierunek, 3 fazowa komunikacja z silnikami z BDLC
- Maksymalna prędkość obrotowa 10 000 obr/min



Rysunek 11: Schemat blokowy układu AS5040

Obsługa przez μC : Dekodowanie sygnału kwadraturowego zaimplementowane jest przy użyciu modułu ETC i funkcji Input Capture mikrokontrolera. Każdy pin portu T ma osobne zewnętrzne przerwanie z konfigurowalnym zboczem wyzwalającym. Wygląd oraz charakterystyczne punkty sygnału kwadraturowego pokazano na rysunku 12, zauważono że przy obrotach w prawo po zboczach na kanale A stan kanału B jest zawsze inny, a przy obrotach w lewo taki sam i odwrotnie dla zboczów kanału B. Więc wyzwalanie przerwań Input Capture skonfigurowano na



Rysunek 12: Dekodowanie sygnału kwadraturowego

dowolne zbocze (opadające lub narastające). W przerwaniu od linii A sprawdzany jest stan linii B (jeśli $A=B$ to zwiększ licznik, w przeciwnym wypadku zmniejsz), a w przerwaniu do linii B sprawdzany jest stan linii A (jeśli $A=B$ to zmniejsz licznik, w przeciwnym wypadku zwiększ). Przykład implementacji:

```
int volatile qdec[4];

/* KANAL 1A */
interrupt 8 void _qdec0ISR( void )
{
    if( PTIT_PTITO != PTIT_PTIT1 )
        qdec[0]++;
    else
        qdec[0]--;
    TFLG1 = TFLG1_COF_MASK; //wyczyszczenie flagi przerwania
}

/* KANAL 1B */
interrupt 9 void _qdec1ISR( void )
{
    if( PTIT_PTITO != PTIT_PTIT1 )
        qdec[0]--;
    else
        qdec[0]++;
    TFLG1 = TFLG1_C1F_MASK; //wyczyszczenie flagi przerwania
}
```

2.2.7 Końcówki mocy

Do sterowania silnikami DC wykorzystano układ L298. Jest to podwójny mostek-H oparty na tranzystorach bipolarnych wraz z układem ich sterowania. Układ posiada zabezpieczenie przed zwarcie (przez tranzystory) zasilania. Sterowanie pojedynczego mostka odbywa się za pomocą trzech wejść w standardzie TTL: IN1, IN2 i EN. Liniami IN1 i IN2 sterowany jest kierunek obrotu silnika natomiast na wejście EN podawany jest sygnał PWM o częstotliwości ok 1.5kHz, który reguluje prędkość obrotową, hamowanie silnika odbywa się przez zwarcie ze sobą jego wyprowadzeń (przez tranzystory z L298). W tabeli 3 przedstawiono możliwe sterowania na wejściach L298 i ich funkcje.

IN1	IN2	PWM	Funkcja
x	x	0	Bieg jałowy
1	0	1	Przód
0	1	1	Tył
0	0	1	Hamowanie
1	1	1	Hamowanie

Tabela 3: Sterowanie L298

Układ L298 jest bardzo prostym w użyciu układem, jedyne zewnętrzne elementy to diody zabezpieczające, a sterowanie odbywa się bezpośrednio z mikrokontrolera. Ma on jednak dużą wadę, a mianowicie spadki napięć na tranzystorach bipolarnych, dochodzą one nawet do ok. 3V przy prądzie 1A⁵. To niekorzystne zjawisko kompensowane jest przez podwyższone napięcie zasilania, jednak takie rozwiązanie nie jest energooszczędne.

Obsługa przez μC : Mikrokontroler generuje 8 bit sygnał PWM na wyjściach PP1 (silnik lewy) i PP4 (silnik prawy) przy użyciu modułu PWM. Pozostałe linie sterujące podłączone są również do portu P. Przy implementacji sterownika silników przewidziano: sterowanie prędkością z rozdzielczością 8 bit w zadanym kierunku, włączenie hamulca (przez ustawienie prędkości 0) oraz możliwość ustawienia przyspieszenia (zwiększenie/zmniejszenie współczynnika wypełnienia co stały okres czasu do uzyskania wartości zadanej). W przypadku przełączania z prędkości

⁵tyle pobiera serwomechanizm podczas zablokowania

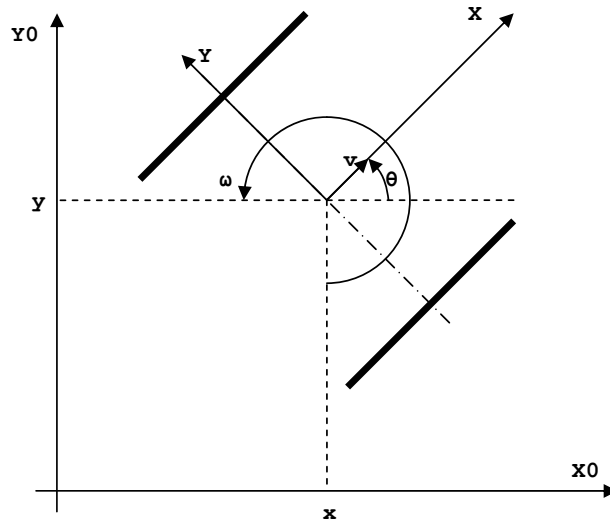
dotądniej na ujemną i na odwrót zawsze jest ustawiania prędkość 0.

3 Implementacja

3.1 Nawigacja przyrostowa

3.1.1 Kinematyka robota

Podwozie robota klasy (2,0) przedstawiono na rysunku 13. Równanie kinematyki przedstawia wzór 1. Sterowaniami są prędkość liniowa robota v oraz prędkość obrotowa ω .



Rysunek 13: Podwozie robota klasy (2,0)

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (1)$$

gdzie:

- (x, y) – położenie robota
- θ – orientacja robota
- v – prędkość liniowa robota
- ω – prędkość obrotowa robota

3.1.2 Implementacja na mikrokontrolerze

Prędkości obrotowe kół odczytywane są z enkoderów. Odczyt odbywa się z liczników dekodera sygnału kwadraturowego co 10ms, po zapamiętaniu wartości liczniki są zerowane. W czasie tej czynności należy zablokować przerwania. Następnie prędkość liniowa v obliczana jako suma prędkości ω_L i ω_P , a prędkość obrotowa ω jako różnica ω_L i ω_P . Wartości wynikowe nie są skalowane w celu uniknięcia błędów spowodowanych zaokrągleniami (wszystkie operacje wykonywane są na liczbach całkowitych). Z wartości v i ω obliczana jest pozycja robota zgodnie z kinematyką 1. Dodatkowo obliczane są takie zmienne jak przejechana droga s , oraz prędkości ds i $d\theta$ uśrednione z ostatnich 100ms. Następnie wszystkie wartości są skalowane do znormalizowanych jednostek (dostępne są zmienne bez skalowania i przeskalowane).

Do obliczania funkcji trygonometrycznych zostały napisane funkcje obliczające \sin i \cos działające na liczbach całkowitych (działają dużo szybciej niż na liczbach zmiennoprzecinkowych). Ich działanie polega na interpolacji tablicy z wybranymi wartościami funkcji \sin .

Kąt: 360° odpowiada 36000

Liczby: 1 odpowiada 10000

Błędy: mniejsze niż 0.0002 (tylko 32% wartości ma błąd większy niż 0.0001)

Szybkość działania: 103 takty procesora, co przy taktowaniu 50Mhz daje ok $2\mu s$

3.2 Sterowanie ruchem

W sterowniku została zaimplementowana kolejka ruchów. Każdy ruch ma takie wartości jak:

- priorytet ruchu
- PWM dla silnika lewego i prawego
- przyspieszenie dla silnika lewego i prawego
- czas, po którym ma zakończyć się ruch
- droga, po przejechaniu której ruch zostaje zakończony

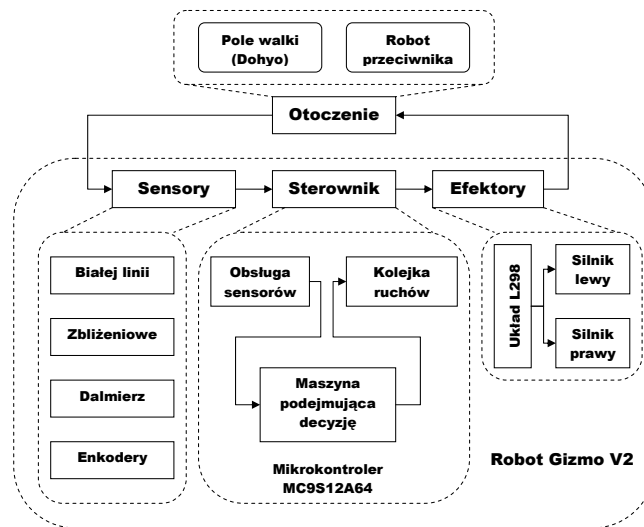
- kąt obrotu robota, po którym ruch zostaje zakończony

Dzięki takiemu rozwiązaniu możliwe jest składanie ruchów podstawowych w sekwencji. Gdy w kolejce nie ma żadnego ruchu domyślnym zachowaniem jest stop. Ułatwieniem w implementacji zachowań robota są priorytety. Aktualnie wykonywany ruch może być przerwany tylko przez ruch o wyższym priorytecie, co zapewnia wykonanie specjalnych ruchów końca.

Przykład zastosowania priorytetów: Gdy robot zostaje przepychany, atak robota jest przerwany przez sekwencje ruchu ucieczki. Robot nie może zaatakować dopóki sekwencja się nie zakończy, ale w przypadku wykrycia białej linii sekwencja ta zostanie przerwana na rzecz ucieczki od białej linii.

3.3 Zachowania

3.3.1 Wprowadzenie

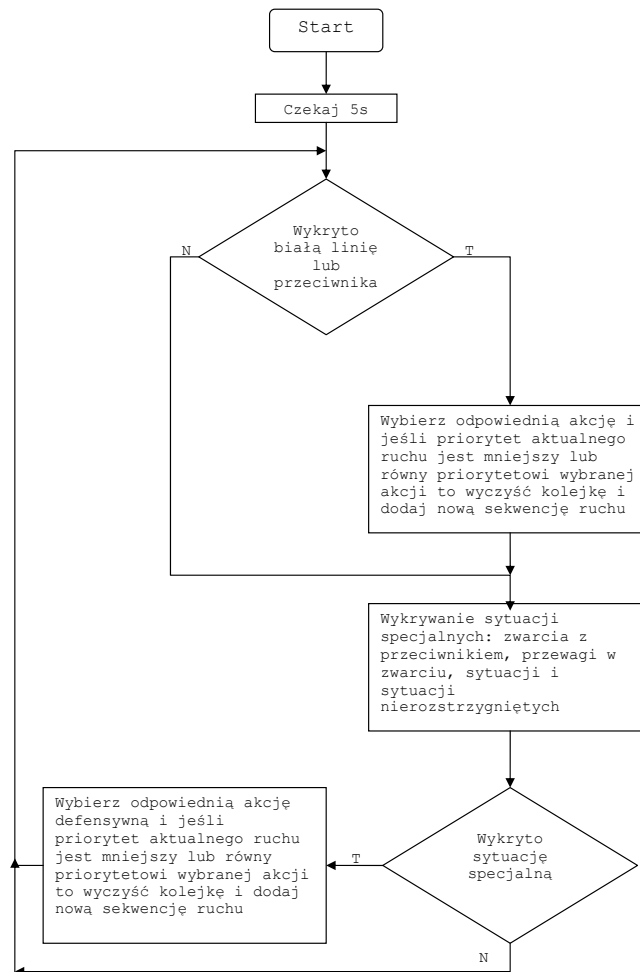


Rysunek 14: Schemat działania robota

W poprzedniej części pracy opisano możliwości robota. Przy implementacji zachowań starano się je wykorzystać. Nie wykorzystano nawigacji przyrostowej ze względu na jej niepewność, nawet toczące się enkodery pod robotem nie zapewniają pewności bezwzględnej pozycji podczas walki, oraz nie zawsze jest znana pozycja początkowa robota. Z enkoderów wykorzystano informacje takie jak: względny

obrót, względne przemieszczenie, prędkość liniową i prędkość obrotową. Odległość odczytywana z dalmierza nie może być brana jako pewna ze względu na strefę martwą, dlatego traktowana jest ona jako binarna (jest coś w odległości mniejszej niż 35cm lub nie ma wcale).

Ogólny algorytm zachowania robota przedstawiono na rysunku 15. Robot po włączeniu zasilania czeka 5s, następnie wchodzi w pętlę algorytmu. Szczegółowy opis zachowań znajduje się w dalszej części pracy. W sekcji 3.3.8 opisano zachowania w sytuacjach specjalnych.



Rysunek 15: Ogólny algorytm zachowań

3.3.2 Szukanie przeciwnika

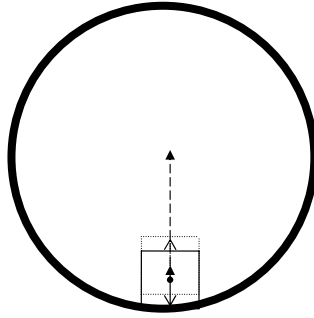
Robot obraca się powoli w lewo przez 10s następnie przemieszcza się do przodu o 15cm i zaczyna obracać powoli w prawo przez 10s. Następnie cykl jest powtarzany.

3.3.3 Reakcja robota na wykrycie białej linii

Opisane tutaj zachowanie jest wykonywane w momencie wykrycia przez robota białej linii, gdy żaden z czujników wykrywania przeciwnika nie wykrył go.

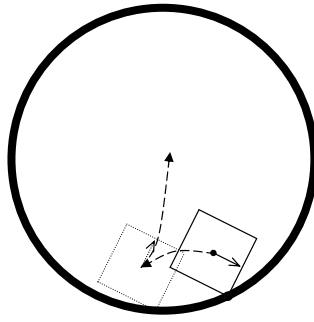
Wyróżnionych zostało osiem przypadków:

1. *Dwa czujniki przednie wykryły białą linię*: robot wycofuje się 4cm, następnie obraca o 180° i jedzie do środka dohyo. Patrz rysunek 16.

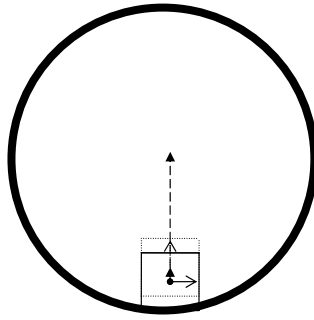


Rysunek 16: Reakcja na białą linię (1)

2. *Czujnik lewy przedni wykrył białą linię*: zapamiętywana jest orientacja robota. Robot wycofuje się po łuku do napotkania białej linii tylnym czujnikiem. Patrz rysunek 17.
3. *Czujnik prawy przedni wykrył białą linię*: zapamiętywana jest orientacja robota. Robot wycofuje się po łuku do napotkania białej linii tylnym czujnikiem. Patrz rysunek 17.
4. *Dwa czujniki lewe wykryły białą linię*: robot obraca się w prawo o 90° i jedzie do środka dohyo. Patrz rysunek 18.

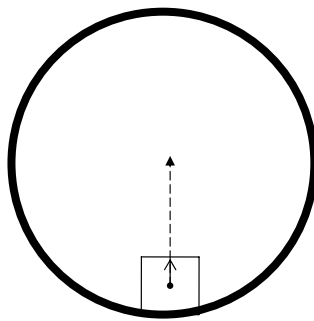


Rysunek 17: Reakcja na białą linię (2)



Rysunek 18: Reakcja na białą linię (3)

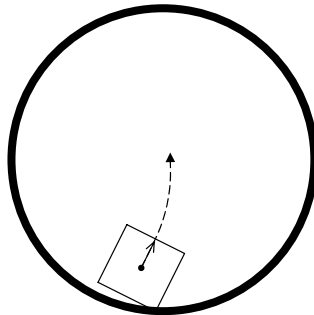
5. *Dwa czujniki prawe wykryły białą linię*: robot obraca się w lewo o 90° i jedzie do środka dohyo. Patrz rysunek 18.
6. *Dwa czujniki tylne wykryły białą linię*: jedzie do środka dohyo. Patrz rysunek 19.



Rysunek 19: Reakcja na białą linię (4)

7. *Czujnik lewy tylni wykrył białą linię*: jeśli w ciągu ostatnich 2s czujnik lewy przedni nie wykrył białej linii to robot jedzie po łuku mniej więcej do środ-

ka dohyo. Patrz rysunek 20. W przeciwnym wypadku obliczany jest kąt, o jaki ma się obrócić, aby dojechał do środka dohyo (na podstawie orientacji zapamiętanej w punkcie 2). Obraca się o ten kąt i jedzie do środka. Patrz rysunek 17.



Rysunek 20: Reakcja na białą linię (5)

8. *Czujnik prawy tylni wykrył białą linię*: jeśli w ciągu ostatnich 2s czujnik prawy przedni nie wykrył białej linii to robot jedzie po łuku mniej więcej do środka dohyo. Patrz rysunek 20. W przeciwnym wypadku obliczany jest kąt, o jaki ma się obrócić, aby dojechał do środka dohyo (na podstawie orientacji zapamiętanej w punkcie 2). Obraca się o ten kąt i jedzie do środka. Patrz rysunek 17.

Po wykonaniu któregoś z ww przypadków robot przełącza się w tryb poszukiwania przeciwnika.

3.3.4 Reakcja robota na wykrycie przeciwnika

Głównym zadaniem robota w takiej sytuacji jest ustawienie się do przeciwnika przodem, więc przyjęto następujące reguły zachowania.

1. *przeciwnik jest z tyłu lub z boku*: bardzo szybki obrót w kierunku przeciwnika.
2. *przeciwnik jest z przodu*: atak na przeciwnika z pełną prędkością. W przypadku gdy tylko dalmierz Sharp wykrył przeciwnika to jedź z pełną prędkością do miejsca, w którym ostatnio widziano przeciwnika.

3. przeciwnik jest z przodu z lewej lub prawej strony: atak na przeciwnika po łuku, którego promień zależy od pozycji przeciwnika (im bardziej z przodu tym mniejszy promień).

3.3.5 Reakcja robota na wykrycie przeciwnika i białej linii

W tym przypadku jest to pomieszczenie dwóch poprzednich zachowań. Gdy przeciwnik znajduje się z przodu robota reakcja na białą linię jest ignorowana i kontynuowany jest atak. W przeciwnym wypadku najprawdopodobniej jest to pozycja przegrana więc robot podejmuje akcję z podrozdziału 3.3.3

3.3.6 Reakcja robota na zgubienie przeciwnika

W przypadku gdy wdziano przeciwnika, a następnie przez 1s czujniki nic nie wykryły robot przełącza się w tryb szukania.

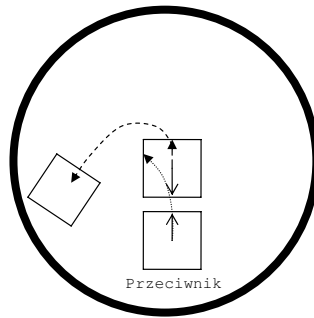
3.3.7 Wykrywanie kontaktu z robotem przeciwnika

Kontakt z robotem przeciwnika wykrywany jest na podstawie prędkości odczytanej z enkoderów toczących się pod robotem w stosunku zadanego sterowania na silnik. Jeśli rzeczywista prędkość robota stanowi maksymalnie 30% oczekiwanej wtedy uznawany jest kontakt z robotem.

3.3.8 Zachowania defensywne podczas kontaktu z robotem

Unik W czasie przepychania się robotów monitorowana jest rzeczywista prędkość robota. Jeżeli prędkość jest ujemna i robot przejechał 5cm w tył wtedy podejmowana jest akcja uniku polegająca na wycofaniu się następnie jazdy po łuku tyłem (rysunek 21). Taki unik może być jedynie skuteczny gdy robot przeciwnika jest wolniejszy. Dlatego alternatywnym zachowaniem w takim przypadku może być gwałtowne wycofanie się i ruszenie do przodu. Takie zachowanie może być stosowane do robotów szybszych.

Wibracje W przypadku gdy roboty nie mogą się przepchać, co 4s robot Gizmo wprowadzany jest w wibracje. Takie zachowanie ma na celu spowodowanie utraty przyczepności kół przeciwnika i przeważenie zwycięstwa na swoją korzyść.



Rysunek 21: Unik w zwarciu

Obrót W walkach robotów często się zdarza, że zaczynają one kręcić się w kółko. Taka sytuacja prowadzi zwykle do nierozstrzygniętej rundy. Robot Gizmo potrafi rozpoznać taką sytuację przez pomiar orientacji. Gdy orientacja zmieni się o 360° i czujniki wykryły przeciwnika podejmowane jest zachowanie polegające na chwilowej zmianie (na przeciwny) prędkości silników.

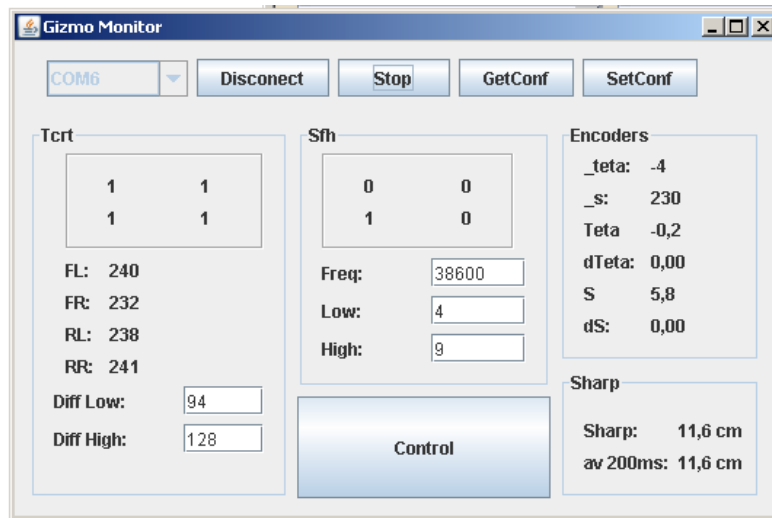
3.4 Środowisko programistyczne

Oprogramowanie mikrokontrolera MC9S12A64 napisano w zintegrowanym środowisku programistycznym *CodeWarrior 4.5 for HC12* firmy *FreeScale*. Jest to środowisko bardzo wygodne w użytkowaniu, dodatkowym atutem jest dopracowany Debugger i symulator układów HC/S12. Jako moduł ICD (In Circuit Debugger) został wykorzystany TBDML (Turbo BDM Light). Jest to projekt na licencji GPL. Autorem jest Daniel Malík.

3.5 Program diagnostyczny

Do celów sprawdzenia poprawności działania sensorów oraz monitorowania algorytmów walki został napisany program diagnostyczny w języku JAVA. Interfejs widoczny jest na rysunku 22. Komunikuje się on z robotem przez protokół RS232. Podczas walk testowych komunikował się z komputerem PC drogą radiową z wykorzystaniem modemu radiowego opisanego w [3].

Oprogramowanie pozwala na monitorowanie stanu czujników (białej linii, zbliżeniowych, Sharp, enkoderów). Pozwala na konfigurację progów histerezy dla czujników oraz na sterowanie robotem. Pozwala również na zapisywanie logów z walki.



Rysunek 22: Interfejs programu GizmoMonitor

Spis rysunków

1	Zdjęcie robota Gizmo V2	3
2	Zdjęcie robota Gizmo	3
3	Schemat konstrukcji robota Gizmo V2	5
4	Zdjęcia konstrukcji robota Gizmo V2	6
5	Schemat rozmieszczenia sensorów w robocie Gizmo V2	7
6	Schemat elektroniki Gizmo V2	10
7	Schemat czujnika białej linii	11
8	Charakterystyka wyjścia czujnika GP2D12	13
9	Schemat czujnika zbliżeniowego IR	14
10	Czułość układu SFH5110 w zależności od częstotliwości	15
11	Schemat blokowy układu AS5040	16
12	Dekodowanie sygnału kwadraturowego	17
13	Podwozie robota klasy (2,0)	19
14	Schemat działania robota	21
15	Ogólny algorytm zachowań	22
16	Reakcja na białą linię (1)	23

17	Reakcja na białą linię (2)	24
18	Reakcja na białą linię (3)	24
19	Reakcja na białą linię (4)	24
20	Reakcja na białą linię (5)	25
21	Unik w zwarciu	27
22	Interfejs programu GizmoMonitor	28

Spis tabel

1	Porównanie parametrów dalmierzy firmy SHARP	13
2	Tabela pomiarów napięcia wyjściowego czujnika GP2D12	13
3	Sterowanie L298	18

Literatura

- [1] Austriamicrosystems. „*AS5040 10 BIT 360° PROGRAMMABLE MAGNETIC ROTARY ENCODER*”, 2006. Datasheet.
- [2] Freescale. „*MC9S12DJ64 Device User Guide V01.19*”, 2004. Datasheet.
- [3] Materek A. „*Prosty modem radiowy*”, Raport instytutowy, ICT, Politechnika Wrocławska, 2007.
- [4] Wnuk M. „*Moduł z mikrokontrolerem MC9S12A64*”, Raport instytutowy, ICT, Politechnika Wrocławska, 2005.
- [5] STMicroelectronics. „*DUAL FULL-BRIDGE DRIVER*”, 2000, Datasheet.