



KoNaR

KOŁO NAUKOWE ROBOTYKÓW

MICROMOUSE RAPID
RAPORT KOŃCOWY

MARCIN NIESTRÓJ

KOŁO NAUKOWE ROBOTYKÓW KoNaR

WWW.KONAR.PWR.WROC.PL

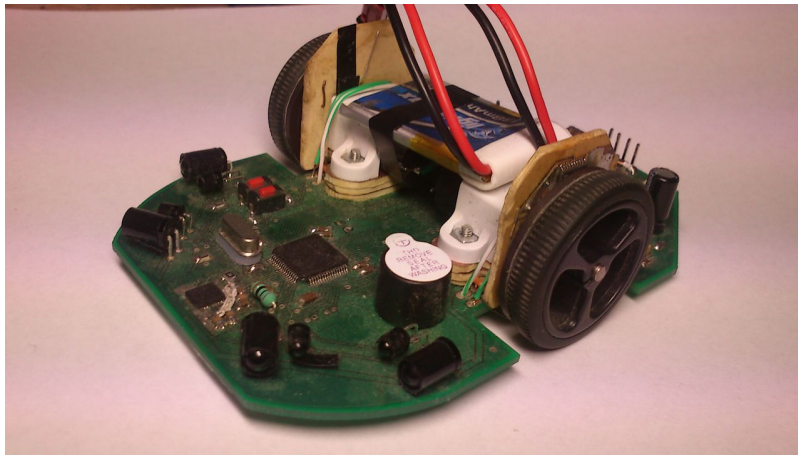
22 STYCZNIA 2013

Spis treści

| | | |
|----------|---|-----------|
| 1 | Wstęp | 2 |
| 2 | Konstrukcja mechaniczna | 2 |
| 2.1 | Podwozie | 2 |
| 2.2 | Napęd | 3 |
| 3 | Elektronika | 3 |
| 3.1 | Mikrokontroler | 3 |
| 3.2 | Czujniki odległości | 4 |
| 3.3 | Enkodery | 4 |
| 3.4 | Żyroskop | 5 |
| 3.5 | Mostki H | 5 |
| 3.6 | Zasilanie | 5 |
| 4 | Oprogramowanie | 10 |
| 4.1 | Interfejs użytkownika | 10 |
| 4.2 | Akwizycja danych z czujników analogowych | 10 |
| 4.3 | Korygowanie pozycji względem ścian | 11 |
| 4.4 | Zmiana wartości ADC z czujników IR na dystans | 12 |
| 4.5 | Profil jazdy robota | 12 |
| 4.6 | Sterowanie silnikami | 14 |
| 4.7 | Algorytm przeszukiwania | 15 |
| 4.8 | Algorytm wyznaczania najszybszej trasy | 15 |

1 Wstęp

Celem projektu było wykonanie robota klasy micromouse. Jego zadaniem jest przeszukiwanie labiryntu oraz optymalny przejazd od miejsca startowego do wyznaczonego pola końcowego. Projekt ten korzystał w dużej mierze na poprzedniej konstrukcji, robocie Wariat. Wyeliminowane zostały wszystkie podstawowe problemy występujące w poprzednim robocie oraz zastosowano większość pomysłów powstałych podczas budowy nowej konstrukcji. Ostatnim zadaniem projektu było wzięcie udziału w zawodach Robotic Arena 2012, co zaowocowało zajęciem pierwszego miejsca w kategorii micromouse.



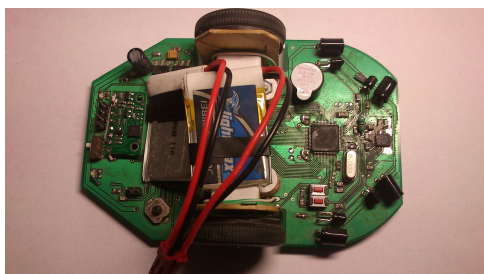
Rysunek 1: Rapid

2 Konstrukcja mechaniczna

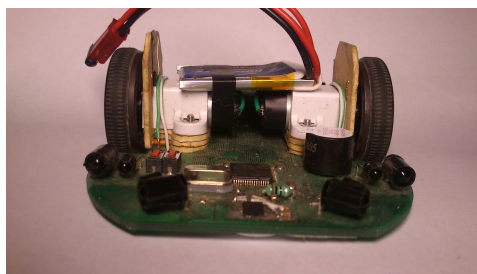
2.1 Podwozie

Robot zbudowany jest na płycie PCB, która stanowi jego podwozie. Założeniem projektu było uzyskanie nisko położonego środka ciężkości oraz jak najmniejszej masy. Dodatkowym atutem robota miała być jego mała szerokość, która pozwalałaby na poruszanie się po "skosie". Kształt robota jest zaokrąglony z przodu. Wynikiem tego zabiegu jest lepsze zachowanie się robota podczas ewentualnych zderzeń ze ścianami lub słupkami.

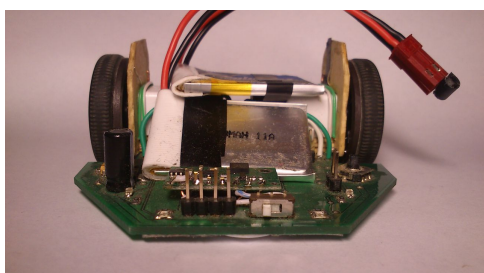
Wymiary robota to 112mm długości i 75mm szerokości, natomiast jego waga to $82 - 87\text{g}$ w zależności od użytych baterii. Zdjęcia przedstawiające robota pokazano na rysunkach 2 - 5.



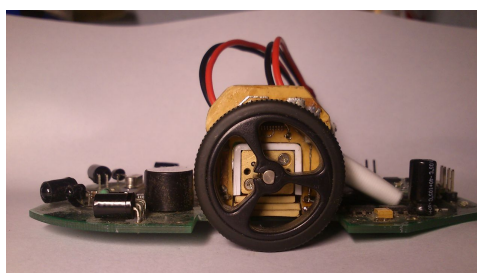
Rysunek 2: Widok z góry



Rysunek 3: Widok z przodu



Rysunek 4: Widok z tyłu



Rysunek 5: Widok z boku

2.2 Napęd

Napędem robota są dwa silniki Pololu 10:1 HP. O ich wyborze zdecydowała cena, dostępność, zintegrowane przekładnie oraz małe wymiary i waga. Silniki jednocześnie pozwalają uzyskać bardzo dobre prędkości przejazdu robota. Maksymalnymi parametrami jakie udało się osiągnąć są: przyspieszenie liniowe równe $a = 9 \frac{m}{s^2}$ oraz prędkość maksymalna $v_{max} = 3 \frac{m}{s}$. Silniki posiadają jednak spory zapas mocy, który będzie można wykorzystać po zaimplementowaniu lepszego algorytmu sterowania robotem.

Negatywną cechą stosowanego napędu jest jego trudne wysterowanie dla małych prędkości, do $30 \frac{cm}{s}$. Własność ta przeszkadza szczególnie w początkowych fazach rozwoju oprogramowania.

3 Elektronika

3.1 Mikrokontroler

Sercem robota jest mikrokontroler STM32F405RGT6. O wyborze zdecydowała chęć nauki programowania mikrokontrolerów STM32 z najpotężniejszym rdzeniem Cortex-M4F oraz najmniejsza obudowa spośród wszystkich procesorów rodziny STM32F4. Najważniejszymi cechami mikrokontrolera są:

taktowanie 168MHz, 192kB RAM, 1MB FLASH, bardzo duża liczba timerów (m.in. ze sprzętową obsługą PWM oraz wyjść kwadraturowych enkoderów), wiele kanałów przetwornika ADC, DMA, sprzętowa obsługa liczb zmiennoprzecinkowych i wiele dostępnych interfejsów komunikacyjnych.

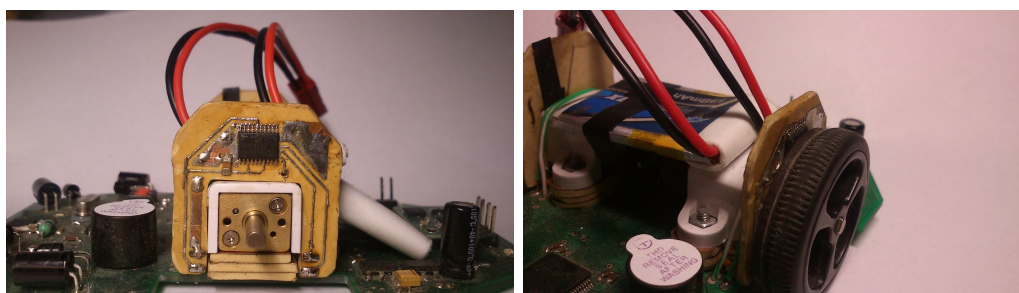
Dostępność poszczególnych peryferii mikrokontrolera na wielu pinach wyjściowych pozwoliła na łatwe zaprojektowanie płytki drukowanej PCB. Duża pojemność pamięci RAM oraz FLASH pozwala natomiast na łatwiejsze i szybsze rozwijanie oprogramowania, bez konieczności ciągłej optymalizacji kodu programu.

3.2 Czujniki odległości

Jako czujniki odległości wykorzystano diody SFH4511 oraz fototranzystory TEFT4300. Schemat podłączenia został pokazany na rysunku 8. Małe wartości rezystorów w szeregowym połączeniu z diodami podczerwieni pozwalają na uzyskanie większej intensywności emitowanego światła, czego wynikiem są dokładniejsze pomiary odległości. Wadą takiego rozwiązania jest konieczność impulsowego sterowania diodami LED (ciągłe świecenie diody powodowałoby jej trwałe uszkodzenie).

3.3 Enkodery

Zamontowanie enkoderów obrotowych z magnesami na osiach silników powodowałoby poszerzenie całej konstrukcji, więc zdecydowano się na użycie enkoderów liniowych wraz z odpowiednimi magnesami. Wykorzystane enkodery AS5304 oraz zamocowane magnesy na kołach robota przedstawiono na rysunku 6.



Rysunek 6: Sposób zamontowania enkoderów i magnesów

Użyte czujniki pozwalają uzyskać rozdzielczość 3520 impulsów na obrót koła, co daje w wyniku 0.028 mm/impuls.

3.4 Żyroskop

Wybrany żyroskopem jest analogowy czujnik ISZ-650 firmy Invensense. Jego zakres pomiarowy to $\pm 2000^\circ/s$. O jego wyborze zdecydowała głównie dostępność i cena w czasie zakupu. Sposób podłączenia czujnika przedstawiono na rysunku 8. Błędem okazał się brak wyprowadzenia pinu pozwalającego na uzyskanie informacji o temperaturze sensora. Konsekwencją tego był szybko narastający dryf czujnika.

Podczas testów robota uszkodzono sensor, dlatego konieczna była jego wymiana. Zdecydowano się na moduł z żyroskopem L3GD20. Do podłączenia wykorzystano miejsce na wyświetlacz N3310 wraz z wyprowadzonym interfejsem SPI. Ważną cechą charakterystyczną tego sensora jest bardzo mały dryf wartości wyjściowej.

3.5 Mostki H

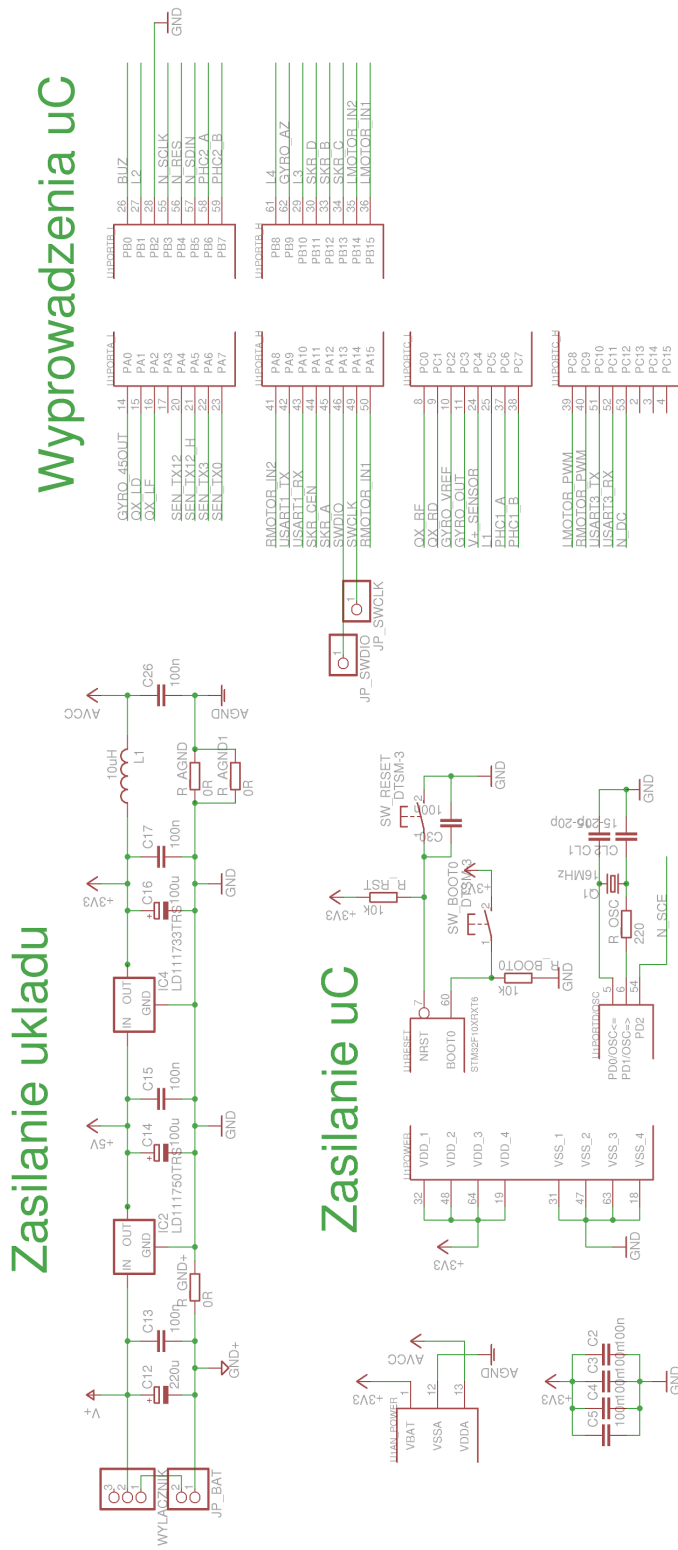
Mostki H pełnią rolę sterownika silników. Sygnałem wejściowym i wyjściowym jest modulacja PWM. W projekcie użyto dwóch układów TB6612, po jednym na każdy silnik. Wynikiem tego jest spory zapas mocy wydzielanej przez sterowniki. Schemat podłączenia układów przedstawiono na rysunku 9.

3.6 Zasilanie

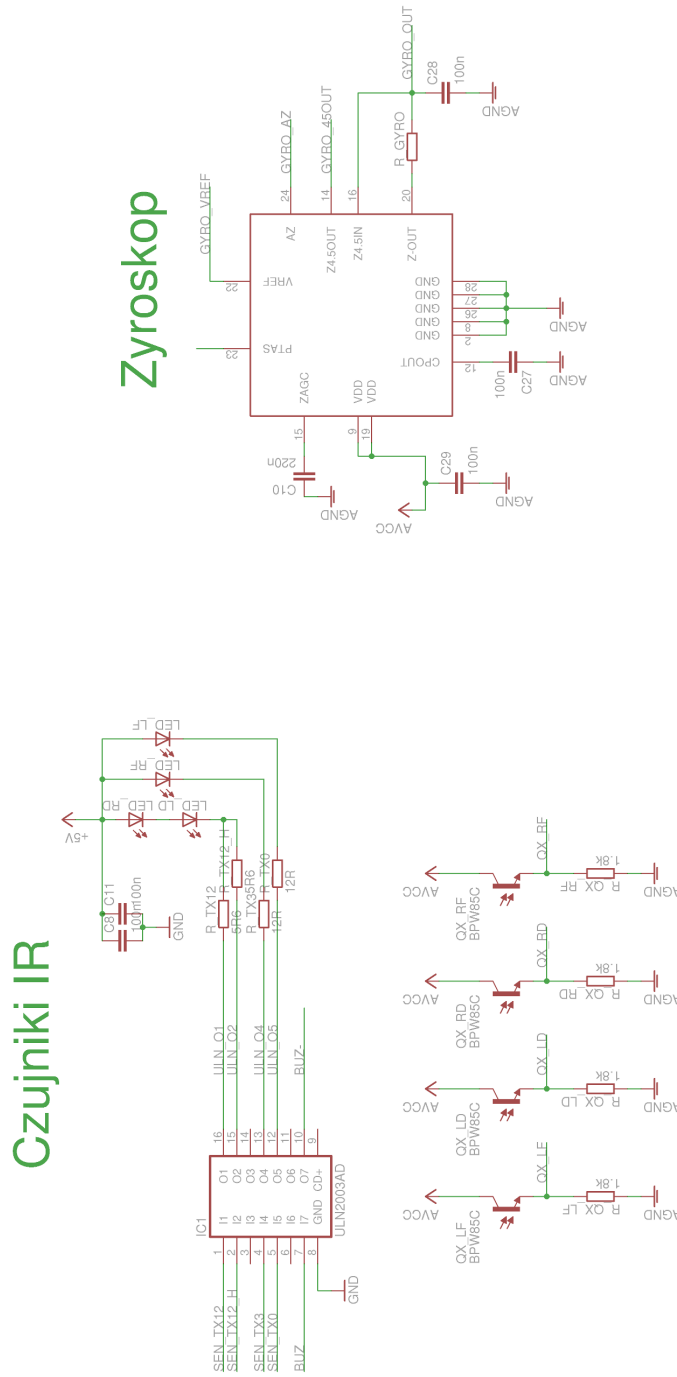
Jako zasilanie wykorzystywane są akumulatory litowo-polimerowe o bardzo małym rozmiarze i niewielkiej masie. Z powodu małych pojemności konieczna jest ich częsta wymiana i ładowanie. W robocie używane są zamiennie dwa zestawy pakietów: 2x Lightmax 138mAh 3.7V oraz 2x 3E Model Max Force 230mAh 3.7V. Baterie są podłączane szeregowo, tak by uzyskać napięcie wyjściowe równe 7.4V.

W konstrukcji użyte zostały dwa stabilizatory, dające na wyjściu 5V oraz 3.3V. Napięcie równe 5V wykorzystywane jest do zasilania enkoderów oraz diod podczerwieni, natomiast napięciem 3.3V zasilana jest pozostała część elektroniki w robocie. Silniki są sterowane natomiast napięciem bezpośrednio z akumulatorów.

Schemat elektryczny zasilania robota przedstawiony jest na rysunku 7.

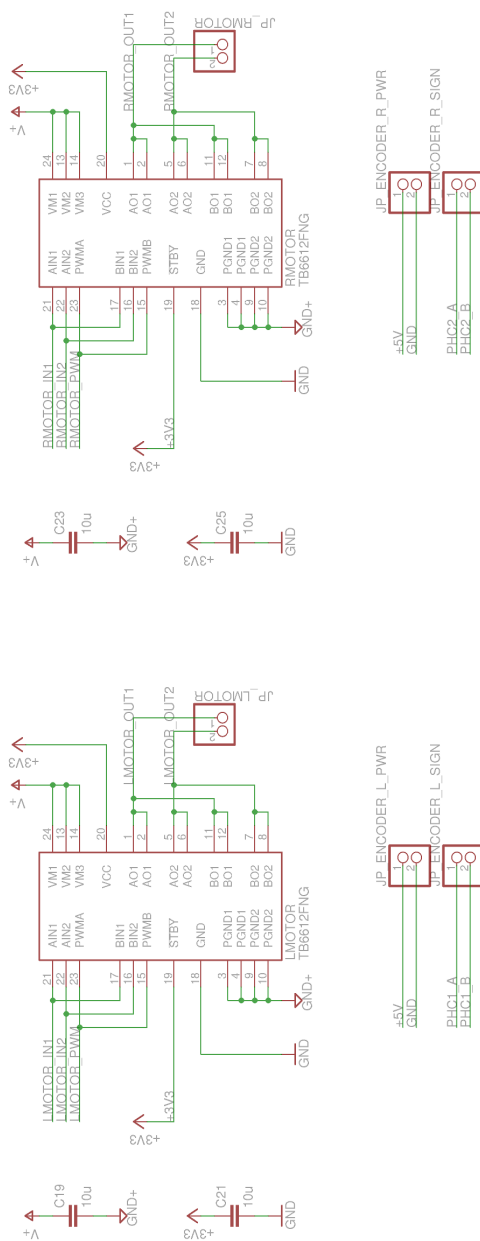


Rysunek 7: Schemat elektryczny - zasilanie oraz wyprowadzenia mikrokontrolera



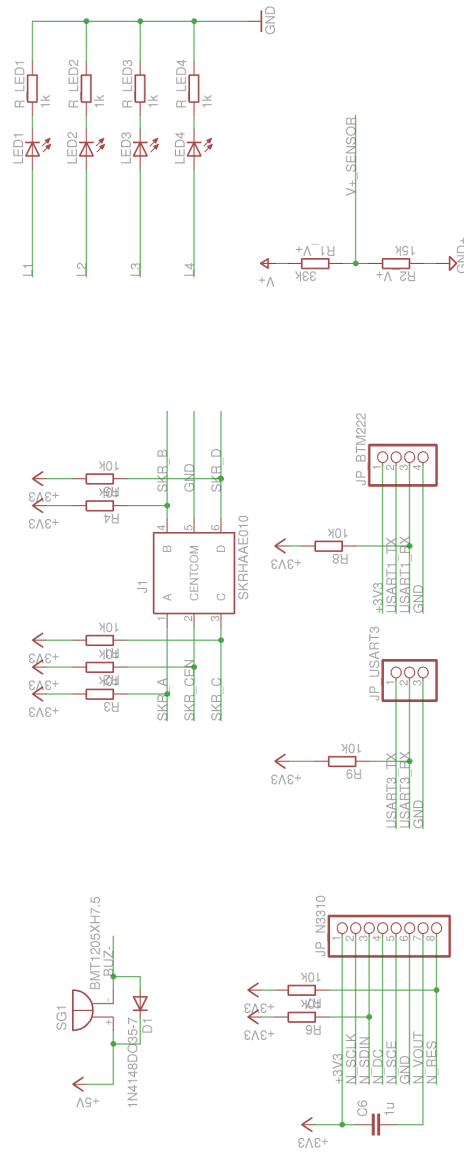
Rysunek 8: Schemat elektryczny - czujniki odległości oraz żyroskop

Silniki + enkodery



Rysunek 9: Schemat elektryczny - silniki oraz enkodery

Komunikacja



Rysunek 10: Schemat elektryczny - interfejs komunikacyjny

4 Oprogramowanie

4.1 Interfejs użytkownika

Założeniem projektu było wykorzystanie joysticka oraz wyświetlacza z telefonu Nokia 3310. Obecnie oprogramowanie nie zawiera obsługi wyświetlacza, dlatego nie jest on zamontowany. Do wyboru prostych komend służy joystick oraz zamontowany buzzer. Buzzer służy także jako powiadomienie o charakterystycznych sytuacjach w labiryncie, np. wykrycie ścianki, dojechanie do środka labiryntu. Jest on niezastąpiony podczas debugowania programu, ponieważ znacznie łatwiej jest usłyszeć sygnał dźwiękowy niż zobaczyć świecąca się diodę LED podczas szybkich przejazdów robota.

W projekcie zastosowano moduł Bluetooth BTM-222 do komunikacji z komputerem oraz analizowania danych sensorycznych. Dzięki niemu możliwe jest wypisanie stanu programu podczas jazdy robota, bez konieczności wyjmowania go z labiryntu.

4.2 Akwizycja danych z czujników analogowych

Oprogramowanie mikrokontrolera służące do zbierania informacji z żyroskopu i dalmierzy optycznych zostało zrealizowane z użyciem następujących peryferii mikrokontrolera: SysTick, ADC oraz DMA. Opis sposobu ich wykorzystania został zamieszczony w tabeli 1.

Żyroskop posiada funkcję automatycznej kalibracji, wyzerowania różnicy pomiędzy napięciem referencyjnym oraz wyjściem żyroskopu. Zauważono jednak, że funkcja ta jest niewystarczająca, dlatego zdecydowano się na programową kalibrację żyroskopu. Poniżej przedstawiono wynik pracy:

```
void CalibrateGyro(int16_t msec) {
    int32_t kat=0;
    int16_t i;

    gyro_calib_time = msec; // Zapamiętanie czasu kalibracji

    for (i=0; i<msec; i++) {
        DelaymSec(1);
        kat += (adc3_buf[1] { adc3_buf[0]); // (GYRO_OUT - GYRO_VREF)
    }

    gyro_calib = kat;
}
```

Tabela 1: Opis wykorzystanych peryferii mikrokontrolera

| Peryferia | Funkcja |
|-----------|--|
| SysTick | Służy on jako przerwanie cykliczne mikrokontrolera co 1ms. W funkcji obsługi przerwania wyzwalane jest przetwarzanie ADC oraz całkowana jest prędkość kątowna żyroskopu. |
| ADC | Przetwornik analogowo-cyfrowy został wykorzystany w celu zamiany sygnału analogowego na prędkość kątową, wykorzystywaną w programie mikrokontrolera. Po zakończeniu konwersji wyzwalany jest transfer DMA. |
| DMA | Służy do skopiowania wyniku konwersji ADC do pamięci RAM. Jest to bardzo ważne przy obsłudze wielu kanałów przetwornika, gdzie wynik przetwarzania jednego kanału jest zapisywany w rejestrze, a następnie po zakończeniu konwersji wyniku kolejnego kanału, do rejestru zapisywana jest nowa wartość. DMA służy głównie do odciążenia procesora oraz łatwiejszego sczytywania pomiarów. |

- *gyro_calib* - wartość odejmowana przy całkowaniu prędkości kątowej z żyroskopu
- *gyro_calib_time* - czas kalibracji

4.3 Korygowanie pozycji względem ścian

Podczas jazdy po labiryncie ważne jest korygowanie pozycji względem ścian po bokach jak i z przodu robota. Sumowanie się małych błędów, wynikających z obliczania rzeczywistej pozycji w labiryncie, prowadzi do błędu rzędu centymetrów. W celu utrzymania robota w środku toru jazdy zastosowano prosty regulator PD. Wejściem dla regulatora jest sygnał z czujników IR, natomiast wyjście regulatora wchodzi bezpośrednio do zmiennej *curPos*, czyli do pozycji zadanej. Czujniki po bokach odpowiadają tylko za pozycję dla ruchu obrotowego robota, natomiast czujniki skierowane na wprost odpowiadają za pozycję dla ruchu postępowego i obrotowego. Dla czujników skierowanych na ściany po bokach robota można wyróżnić trzy sytuacje. Pierwsza i najprostszą sytuacją ma miejsce wtedy, gdy po obu stronach są ściany. Wejściem dla regulatora jest wtedy różnica między odległością od lewej i od prawej ściany. Druga sytuacja ma miejsce wtedy, gdy po jednej stronie jest ściana.

Wejściem regulatora jest w tym wypadku różnica między odległością od tej ściany i odległością od ściany, gdy robot jest na środku toru jazdy. Trzecia sytuacja to taka, gdy nie ma ścian z żadnej ze stron. W tym wypadku niemożliwe jest określenie błędu pozycji robota, więc regulacja jest wyłączana. Regulacja przy pomocy czujników skierowanych na wprost włączana jest jedynie podczas dojeżdżania do ściany. Wejściem dla regulatora pozycji dla ruchu postępowego jest suma odległości do ściany z obydwu czujników, natomiast wejściem dla regulatora pozycji dla ruchu obrotowego jest różnica odległości od ściany.

4.4 Zmiana wartości ADC z czujników IR na dystans

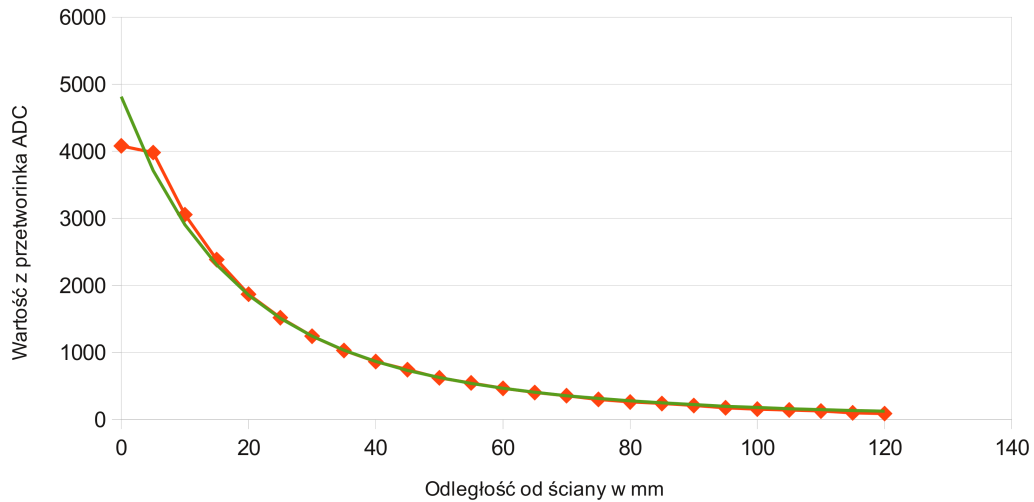
Przejazd w labiryncie, jak i korygowanie pozycji robota wymaga dokładnego określenia odległości od ściany. W tym celu konieczna jest zamiana sygnału z czujników na dystans w milimetrach. Na podstawie odczytu wartości ADC z czujników, wybrano równanie (1) [2], które dobrze określa zależność odległości od ściany od wartości pomiarowej z przetwornika ADC.

$$y = \ln\left(\frac{a}{x - b}\right) \quad (1)$$

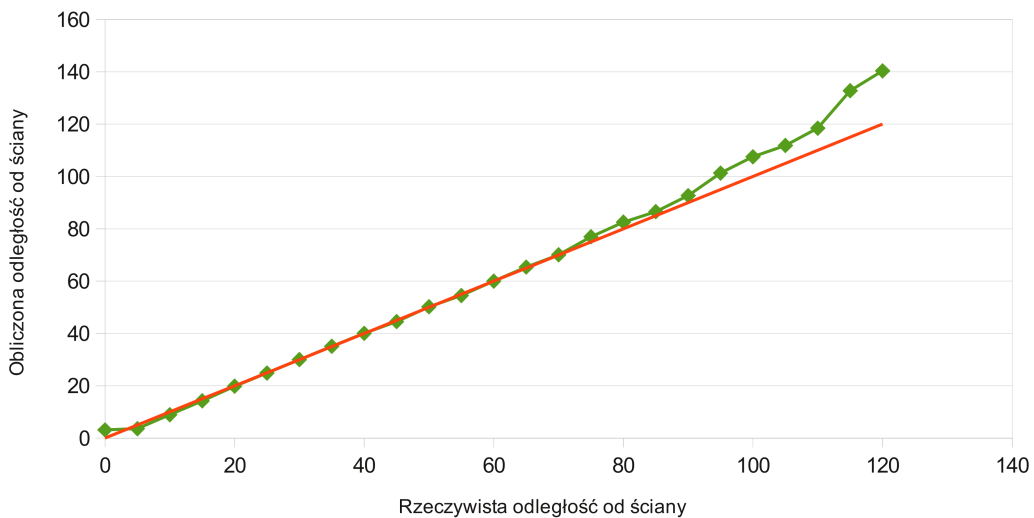
Symbol y oznacza wartość ADC, natomiast x oznacza odległość od ściany. Równanie ma dwa parametry (a oraz b), zależne od aktualnego oświetlenia oraz koloru i powierzchni ścian. W celu skalibrowania czujników do danej ściany, wymagany jest pomiar wartości dla dwóch, znanych odległości. Przykładowe pomiary wraz z krzywą aproksymującą zostały przedstawione na rysunku 11. Linia pomarańczowa oznacza wartości z przetwornika dla danej odległości od ściany, natomiast linia zielona jest określona równaniem podanym wcześniej, dla skalibrowanych już czujników (określonych wartości a oraz b). Zależność wyliczonej odległości od ściany od jej rzeczywistej wartości została przedstawiona na rysunku 12. Można zauważyć, że określenie odległości ma najmniejsze błędy (rzędu ułamków mm) dla przedziału 10-80 mm, czyli dla najczęściej występujących sytuacji w przejeździe robota po labiryncie.

4.5 Profil jazdy robota

Podczas jazdy robota po labiryncie, ważna jest dokładna znajomość pozycji. Pozycja robota jest możliwa do obliczenia z odczytu enkoderów, pod warunkiem, że koła robota się nie ślizgają. W tym celu należało zapewnić stopniowe rozpędzanie się robota oraz stopniowe hamowanie [2].



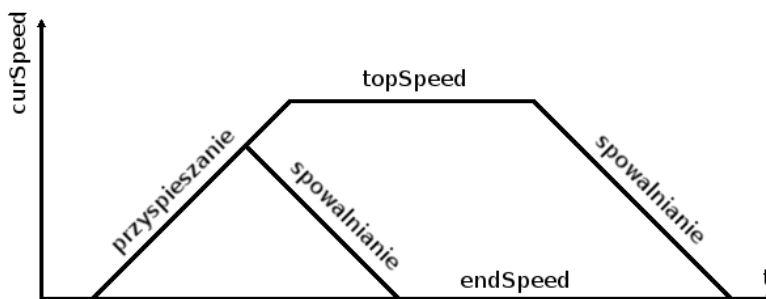
Rysunek 11: Pomarańczowe punkty - wartości z przetwornika ADC, zielona linia - równanie aproksymujące



Rysunek 12: Pomarańczowa linia - rzeczywiste wartości odległości, zielone punkty - wyliczone wartości

Ruch robota został rozbity na ruch postępowy oraz ruch obrotowy. Każdy z tych ruchów jest sterowany tym samym algorytmem, jednak wyjaśnienie zostanie przedstawione jedynie na przykładzie ruchu postępowego. Program zapisuje prędkość, z którą robot powinien się w danym momencie poruszać, w zmiennej *curSpeed*. Podczas jazdy po długiej prostej, robot zwiększa swoją

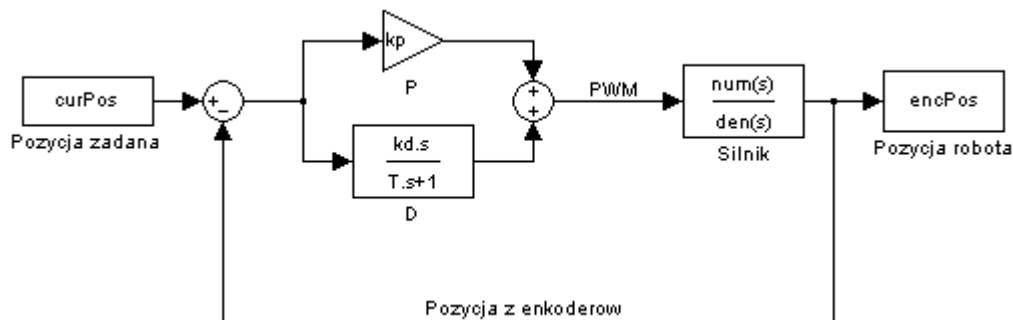
prędkość, aż do uzyskania maksymalnej zadanej prędkości $topSpeed$. W każdym kolejnym kroku (przerwaniu mikrokontrolera) obliczane jest opóźnienie (wartość przeciwna przyspieszenia) ze wzoru $a = \frac{curSpeed^2 - endSpeed^2}{2 \cdot dist}$, gdzie $dist$ oznacza dystans który pozostał do pokonania, natomiast $endSpeed$ jest zadaną prędkością końcową. Obliczona wartość porównywana jest z zadanym przyspieszeniem, z którym ma się poruszać robot. W momencie przekroczenia tej wartości, algorytm rozpoczyna fazę spowalniania. Wartość zmiennej $curSpeed$ podczas fazy przyspieszania i spowalniania została przedstawiona na rysunku 13.



Rysunek 13: Stan prędkości $curSpeed$ w fazach: przyspieszenia, prędkości maksymalnej i spowalniania

4.6 Sterowanie silnikami

Sterowanie prędkością silników zostało zrealizowane przez mostek H, którego sygnałem wejściowym jest sygnał PWM. Wartość sygnału obliczana jest przez regulator PD (proporcjonalno-różniczkujący) osobno dla ruchu postępowego i ruchu obrotowego [3]. W labiryncie najważniejsze jest położenie robota, dlatego jest ono wartością zadaną regulatora. Pozycja robota $curPos$ jest obliczana przez sumowanie (całkowanie) prędkości $curSpeed$. Uchyb regulatora jest obliczany przez odjęcie od zadanej pozycji przebytego dystansu, odczytanego z enkoderów oraz żyroskopu. Wartość odczytana z enkoderów zostaje przekształcona z prędkości dwóch kół na prędkość liniową (suma prędkości kół), wartość z żyroskopu na prędkość kątową. W końcowej fazie, sygnał wyjściowy z regulatora musi być zamieniony z prędkości liniowej i kątowej na prędkość (sygnał PWM) dla każdego koła z osobna. Zostaje to wykonane w sposób odwrotny do zamiany sygnału z enkoderów. Cały układ regulacji został przedstawiony na rysunku 14.



Rysunek 14: Schemat układu regulacji zadanej pozycji robota

4.7 Algorytm przeszukiwania

W robocie został zaimplementowany algorytm wlewania wody (floodfill). Służy on do wyznaczania najkrótszej trasy prowadzącej do środka labiryntu. Zakładając, że robot porusza się ze stałą prędkością, wyznaczona najkrótsza ścieżka jest także najszybszą ścieżką prowadzącą do celu. Algorytm floodfill wykorzystywany jest także do przeszukania pozostałej części labiryntu oraz do powrotu robota do pola startowego.

4.8 Algorytm wyznaczania najszybszej trasy

Algorytm wyznaczania najszybszej ścieżki różni się od poprzedniego tym, że wagami przejazdu nie jest jego odległość, natomiast jego czas. Pozwala to na wybranie przez robota trasy z mniejszą liczbą zakrętów i większą liczbą odcinków prostych, a w konsekwencji także lepszym wynikiem końcowym danego przejazdu.

Literatura

- [1] <http://invensense.com/jp/mems/gyro/documents/PS-ISZ-0650B-00-02.pdf>
- [2] Strona konstruktorów robotów micromouse, <http://www.micromouseonline.com/>
- [3] Strona konstruktora Ng Beng Kiat, <https://sites.google.com/site/ngbengkiat/>