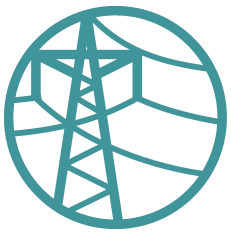


WYDZIAŁ
ELEKTROTECHNIKI
I INFORMATYKI
POLITECHNIKI RZESZOWSKIEJ

Damian Bielecki

Implementacja sterowania hierarchicznego mobilnym robotem kołowym z wykorzystaniem języka Python

Praca dyplomowa inżynierska

Opiekun pracy:
dr. Paweł Penar

Rzeszów, 2023

Spis treści

Przegląd literatury dotyczącej algorytmu A* Implementacja algorytmu A* w języku Python. Integracja implementacji algorytmu A* ze środowiskiem symulacyjnym. Testowanie przyjętych rozwiązań na obiekcie rzeczywistym.

1. Wprowadzenie

Ludzie od zawsze próbują latać, mój aStar im w tym nie pomoże

1.1. Cel pracy

Celem pracy jest zaimplementowanie, przetestowanie w środowisku symulacyjnym i rzeczywistym algorytmu najkrótszej ścieżki(A*)

1.2. Zakres pracy

Na zakres pracy składa się:

- Przegląd literatury
- Implementacja algorytmu A* w języku Python
- Symulacja działania
- Weryfikacja na obiekcie rzeczywistym

2. Przegląd literatury

3. Przegląd istniejących rozwiązań

4. Opis środowiska

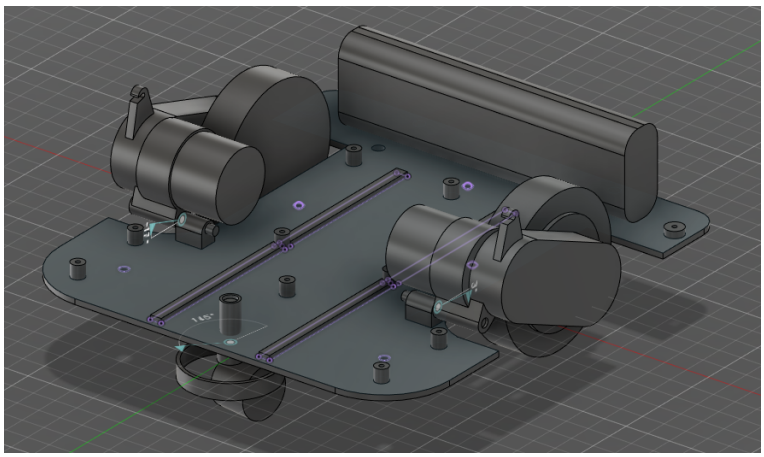
5. Projektowanie robota

5.1. Założenia projektowe

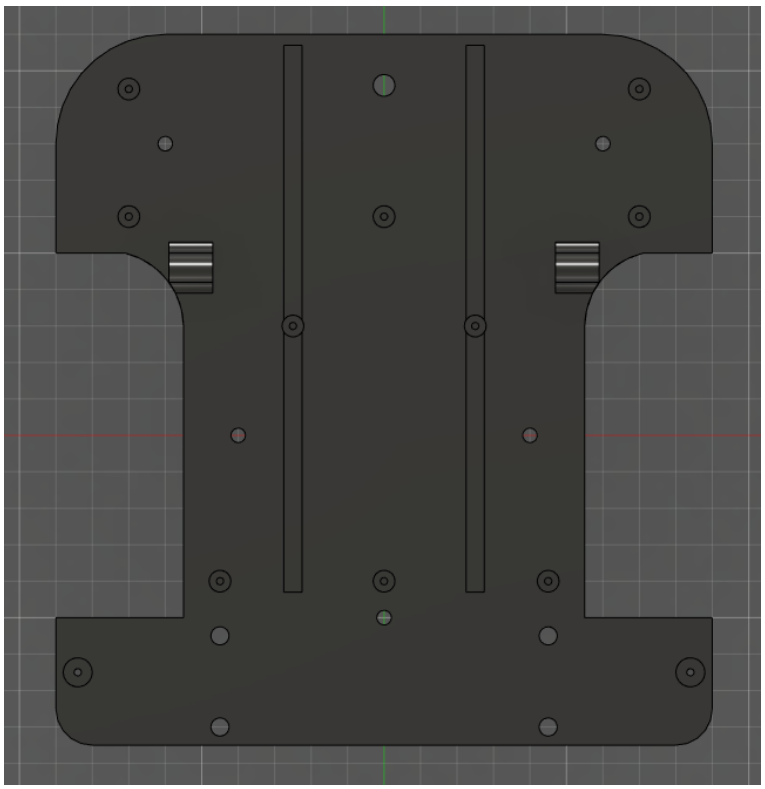
Wykonany robot powinien być jak najmniejszy i najprostszy w wykonaniu oraz sterowaniu tak aby móc przetesotwać przy jego pomocy działanie algorytmu A*. Robot będzie zbudowany z platformy, do której zostaną przyczepiony napęd, elektronika sterująca oraz bateria. Do platformy zostaną przymocowane dwa gotowe moduły napędowe składające się z silnika, przekładni oraz dużego koła. Aby pojazd stał stabilnie, doczepione zostanie trzecie koło obracające się swobodnie w każdym kierunku. Całość będzie sterowana przy pomocy mikrokontrolera ESP32 oraz dwukanałowym sterownikiem silników DC opartym na układzie L298n. Za zasilanie będzie odpowiadała litowo-jonowy akumulator 4S.

5.2. Projektowanie zarysu robota w środowisku CAD

Zbudowany ma być prosty w budowie i wykonaniu a więc założyłem że podstawa utrzymujące pozostałe komponenty zostanie wydrukowana na drukarce 3D. Model platformy i pozostałych posiadanych elementów został wykonany w programie Fusion 360. Modele modułów napędowych, przedniego kółka oraz baterii pozwoliły na optymalne wyznaczenie pod względem wielkości lokalizacji wszystkich elementów.

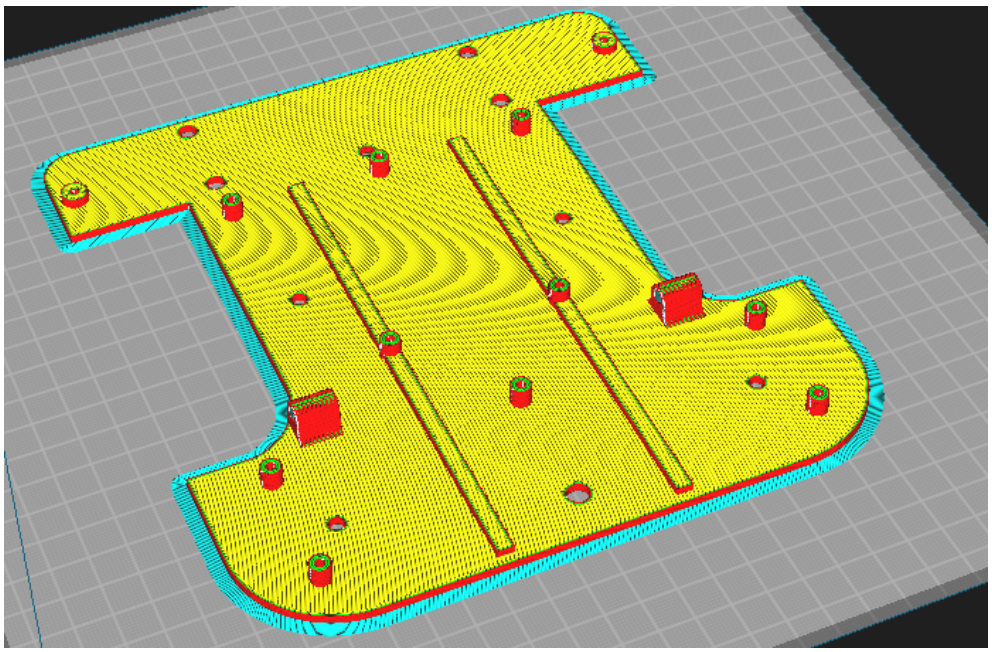


Rysunek 5.1: Zaprojektowane podwozie robota w programie Fusion360



Rysunek 5.2: Widok z góry na podwozie robota

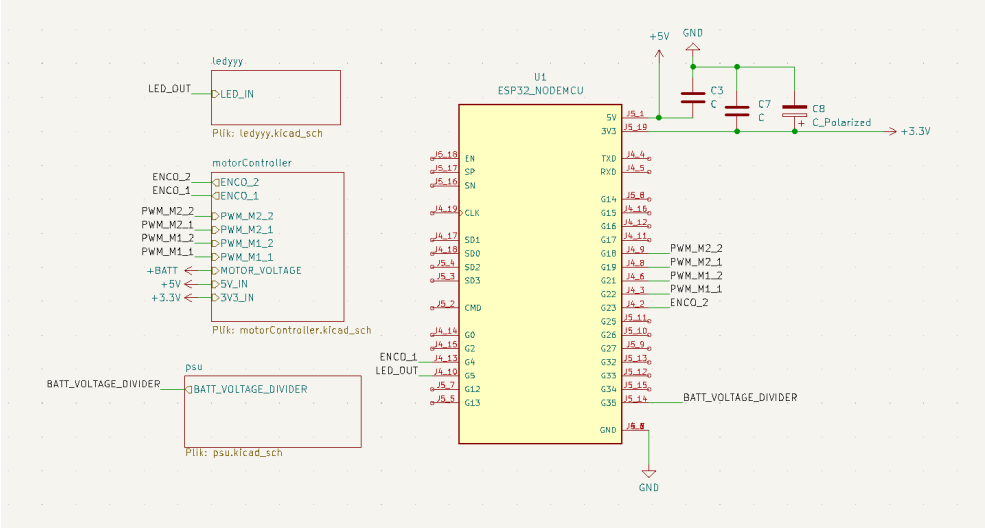
Model został przygotowany do druku w programie Ultimaker Cura. Parametry druku zostały dobrane eksperymentalnie na podstawie własnych doświadczeń. Podstawę wydrukowano z PLA w temperaturze 220*C i wysokości warstwy 0,2mm. Żeby zwiększyć wytrzymałość temperatura głowicy została lekko zawyżona względem wymagań producenta filamentu



Rysunek 5.3: Widok przygotowanego do druku modelu

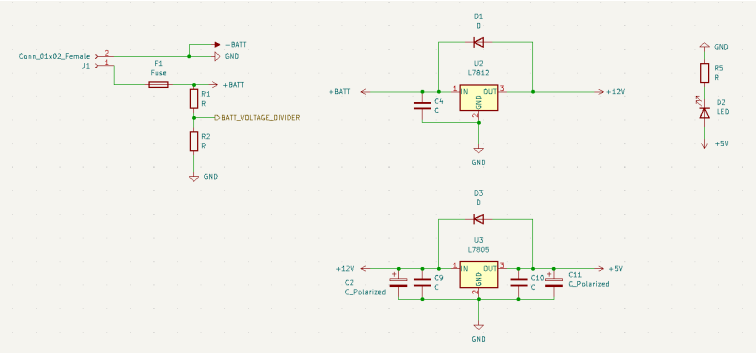
5.3. Projekt, wykonanie oraz podłączenie elektroniki robota

Ze względu na chęć bezprzewodowego sterowania robotem zostanie wykorzystany moduł ESP32, dla którego zostanie przygotowana odpowiednia płytka z wyprowadzeniami do enkoderów silnika oraz ich sterownika. Schemat elektroniczny i projekt pcb został wykonany w programie KiCad.



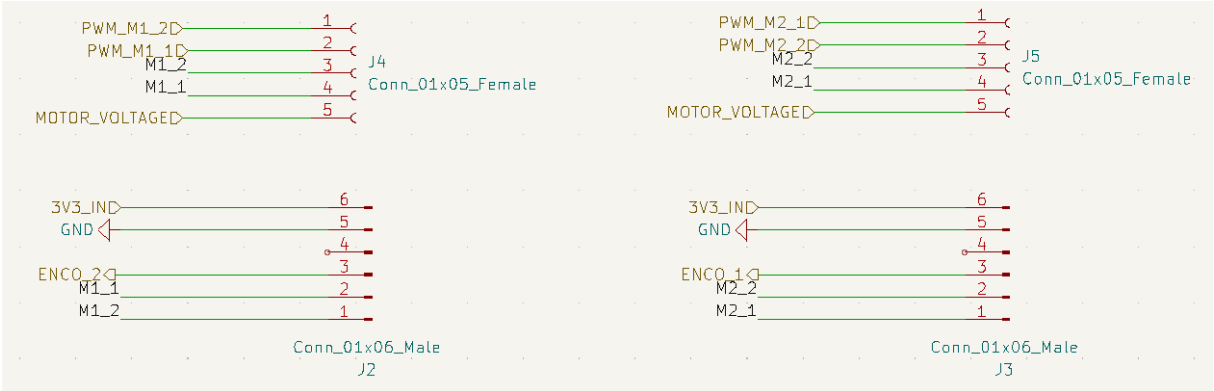
Rysunek 5.4: Ogólny schemat połączeń

Wszystkie potrzebne elementy podsystemy mikrokontrolera zostały już wlutowane w module deweloperskim a więc mój schemat zawiera jedynie odpowiednia połączenia z modułami roboczymi.



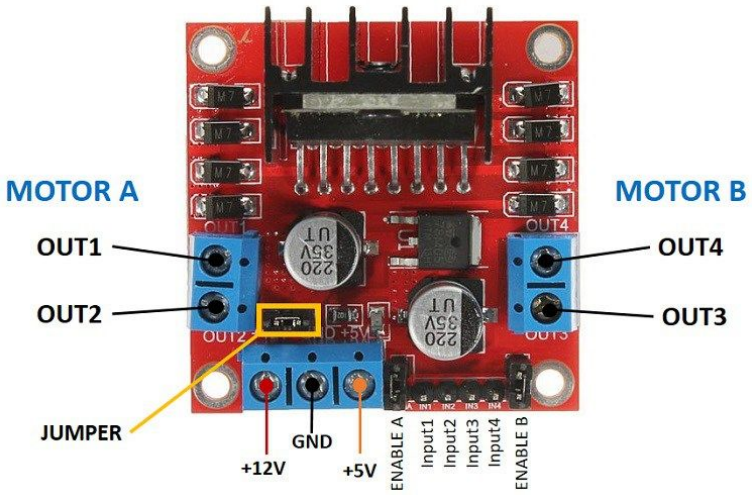
Rysunek 5.5: Ogólny schemat połączeń

Użyta bateria posiada napięcie maksymalne 16,8V a więc musi zostać odpowiednio zmniejszone przed podaniem go na piny zasilania. Ze względu na wykorzystywaną łączność bezprzewodową a więc zwiększone zużycie prądu zmniejszanie napięcia zostało podzielone na dwie sekcje. Najpierw zmniejszane jest poprzez stabilizator LM7812 z 16.8V do 12V a następnie poprzez LM7805 z 12V napięcie redukowane jest do 5V. ESP32 zasilane jest napięciem 3.3V tworzonym poprzez stabilizator AMS1117 w module deweloperskim. Aby zredukować spadki napięć powstałe przy nagłym poborze prądu dodałem dodatkowe kondensatory filtrujące. Dodatkowo na schemacie widoczny jest dzielnik napięcia pozwalający na poziom naładowania baterii przez sterownik jednak ostatecznie nie został on wykorzystany w projekcie.

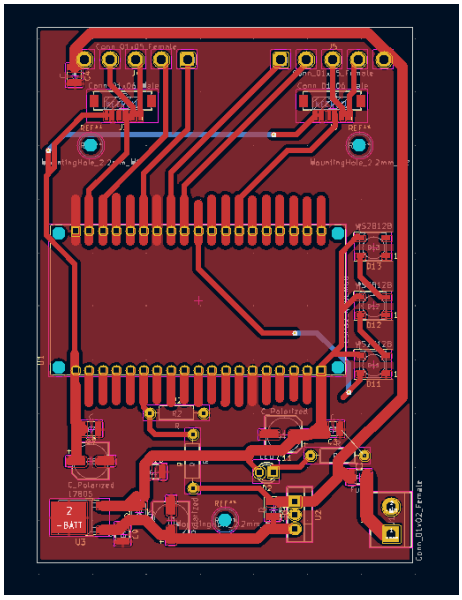


Rysunek 5.6: Ogólny schemat połączeń silników

Użyte moduły napędowe posiadają enkoder inkrementalny zbudowany z czujnika halla i tarczy magnetycznej zamocowanej na wale silnika. Czujnik halla jest zasilany napięciem 3.3V i na wyjściu otrzymujemy sygnał ze śzpilkami"proporcjonalny do aktualnych obrotów silnika. Silniki zasilane są poprzez moduł z układem L298n a prędkość ustalana jest poprzez odpowiednio generowany sygnał PWM.



Rysunek 5.7: Wykorzystany moduł do sterowania silnikami, L298n



Rysunek 5.8: Projekt PCB

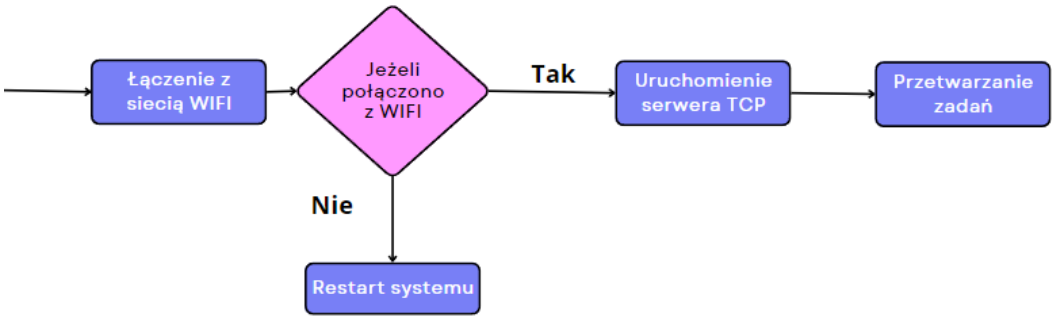
Na widocznym powyżej zdjęciu widać ostateczną wersje projektu pcb. Można zauważyć że widoczne są na niej trzy adresowalne diody świecące, które nie zostały

wykorzystywane przez napisane oprogramowanie.

5.4. Oprogramowanie robota

Program sterujący robotem został napisany w C++. Szkielet aplikacji bazuje na projekcie utworzonym przez framework IDF w wersji 4.4 udostępnionym przez producenta użytego procesora. Po za tym użyta została biblioteka implementująca system czasu rzeczywistego FreeRTOS i ASIO do obsługi połączenia TCP.

Rysunek 5.9: Schemat programu cz.1



Rysunek 5.10: Schemat programu cz.2

Działanie programu rozpoczyna się od wywołania funkcji app_main, inicjalizacji funkcji systemowych i sterowników silników. W dalszej kolejności nawiązywana jest łączność z siecią WiFi. W przypadku braku połączenia system resetuje się. Po poprawnym połączeniu uruchamiany jest kontekst biblioteki ASIO a następnie uruchomienie serwera TCP. Klienci połączeniu do serwera utowrzonego przez robota wysyłają komendy tekstowe wraz z odpowiednimi arguemntami, które robot odpowiednio przetwarza.

5.4.1. Sterowanie silnikami

Sterowanie silnikami odbywa się poprzez klasę MotorController, która dziedziczy po klasie PIDController implementującej regulator PID. Obiekt automatycznie

tworzy timer uruchamiający co 100ms metodę aktualizującą wyjścia sterujące silnikiem. Aktualna prędkość wyznaczana jest na podstawie przerwania wyzwalanego przez enkoder silnika. Przerwanie inkrementuje licznik, czyszczony przez wcześniej opisany timer. Nastawy regulatora PID zostały dobrane eksperymentalinie, człon proporcjonalny wynosi 8 a całkujący i różniczkujący 0,1.

5.4.2. Serwer TCP

Literatura

[1] http://weii.portal.prz.edu.pl/pl/materialy-do-pobrania. Dostęp 5.01.2015.

[2] Jakubczyk T., Klette A.: Pomiary w akustyce. WNT, Warszawa 1997.

[3] Barski S.: Modele transmitancji. Elektronika praktyczna, nr 7/2011, str. 15-18.

[4] Czujnik S200. Dokumentacja techniczno-ruchowa. Lumel, Zielona Góra, 2001.

[5] Pawluk K.: Jak pisać teksty techniczne poprawnie, Wiadomości Elektrotechniczne, Nr 12, 2001, str. 513-515.

POLITECHNIKA RZESZOWSKA im. I. Łukasiewicza Rzeszów, 2023
Wydział Elektrotechniki i Informatyki

STRESZCZENIE PRACY DYPLOMOWEJ INŻYNIERSKIEJ

IMPLEMENTACJA STEROWANIA HIERARCHICZNEGO MOBILNYM ROBOTEM KOŁOWYM Z WYKORZYSTANIEM JĘZYKA PYTHON

Autor: Damian Bielecki, nr albumu: EF-163461
Opiekun: dr. Paweł Penar
Słowa kluczowe: (max. 5 słów kluczowych w 2 wierszach, oddzielanych przecinkami)

Treść streszczenia po polsku

RZESZOW UNIVERSITY OF TECHNOLOGY Rzeszow, 2023
Faculty of Electrical and Computer Engineering

BSC THESIS ABSTRACT

TEMAT PRACY PO ANGIELSKU

Author: Damian Bielecki, nr albumu: EF-163461
Supervisor: (academic degree) Imię i nazwisko opiekuna
Key words: (max. 5 słów kluczowych w 2 wierszach, oddzielanych przecinkami)

Treść streszczenia po angielsku