

Autor:
Teodor Ros

Anyway it's I'am i mean I 😊

Robot Line Follower –

opis utworzonego projektu!

Zapraszam do lektury! 😊

**Jak coś nie jasne to pytaj, prace tą pewnie zdobyłeś z
mojego profilu GitHub lub You Tube więc kontakt do
mnie masz 😊**

Spis treści

1. Wstęp.....	3
2. Konstrukcje autonomicznych robotów pojazdów.....	5
2.1. Popularne zawody robotyczne	5
2.1.1. Micromouse	5
2.1.2. Line Follower	7
2.1.3. Walki Robotów Sumo	8
2.2. Zaawansowane autonomiczne pojazdy	9
3. Projekt i realizacja robota śledzącego.....	10
3.1. Wnioski z przeglądu konstrukcji autonomicznych robotów pojazdów	10
3.2. Ograniczenia oraz wymagania konstrukcyjne wynikające z regulaminu zawodów	10
3.3. Podstawowe parametry robota	11
3.3.1. Położenie względem linii.....	11
3.3.2. Sterowanie ruchem	11
3.4. Napęd oraz konstrukcja mechaniczna.....	12
3.5. Ogólna koncepcja budowy robota.....	12
3.6. Projekt części elektronicznej.....	13
3.7. Orientacja robota na torze	16
3.8. Zasilanie robota.....	16
3.9. Moduł główny sterowania.....	17
3.10. Dodatkowa pamięć operacyjna	18
3.11. Sterowanie napędem robota	19
3.12. Algorytm sterowania.....	20
3.13. Bezprzewodowa komunikacja	22
3.13.1. Aplikacja mobilna.....	23
3.14. Projekt i realizacja obudowy robota.....	24
4. Testy i uruchomienie robota	26
4.1. Budowa tras testowych	26
4.2. Testy robota.....	27
5. Podsumowanie i wnioski końcowe.....	31
Literatura.....	32
Dodatek A – Schemat płytki sterującej.....	Błąd! Nie zdefiniowano zakładki.
Dodatek B – Schemat płytki z czujnikami	Błąd! Nie zdefiniowano zakładki.

1. Wstęp

Rozwój nauki i techniki w czasie nieco dłuższym niż ostatnie 50 lat przyczynia się do stopniowego, choć szybkiego rozwoju elektroniki cyfrowej i techniki mikroprocesorowej. Dobrze ilustruje to ilość tranzystorów upakowanych w jednej krzemowej obudowie. W roku 1964 wynosiła ona 10. Zaledwie 5 lat później już około tysiąca, natomiast po kolejnych 6 latach ponad 32 tysiące. Pierwszy procesor komercyjny powstał w 1971. Był on zaprojektowany i produkowany przez firmę Intel. Firma ta zastosowała całkowicie nowe podejście do problemu. Zamiast projektować zestaw układów scalonych, zdecydowano się zaprojektować zestaw uniwersalnych bloków, które można by zaprogramować do wykonywania różnych funkcji np. kalkulatora. Ten zestaw bloków składał się z jednostki przetwarzającej (CPU), pamięci (RAM), pamięci stałej (ROM) i rejestru przesuwne, umożliwiającego zewnętrzne połączenia systemu (I/O, wejścia-wyjścia). Jego częstotliwość taktowania wynosiła wtedy zaledwie 108kHz. Produkowane obecnie procesory osiągają graniczne możliwości technologii krzemowej tj. ponad 3GHz. Pokazuje to, że prędkość procesorów zwiększyła się od momentu ich powstania ponad trzydzieści tysięcy razy [1].

Należy zauważyć, że sam procesor (CPU) bez układów peryferyjnych okazuje się bezużyteczny, bo musi się on komunikować ze światem zewnętrznym, wysyłać wynik obliczeń itp. wymaga więc użycia układów peryferyjnych, które to zrealizują. Jednak w świecie elektroniki cyfrowej funkcjonuje też mikrokontroler, będący całkowicie autonomicznym systemem mikroprocesorowym, niewymagającym użycia dodatkowych elementów, których wymagałby do pracy tradycyjny procesor. Jest to możliwe dzięki temu, że wszystkie niezbędne do tego celu układy peryferyjne są wbudowane w strukturę krzemową. Dlatego też mikrokontrolery przystosowane są do bezpośredniej współpracy z różnymi urządzeniami zewnętrznymi. Obecnie mikrokontrolery i elementy elektroniczne są tanie i łatwo dostępne. Wiedza na temat ich budowy i zastosowania dzięki Internetowi jest szeroko rozpowszechniona. W sieci Internet oraz w literaturze można znaleźć wiele praktycznych przykładów umożliwiających rozpoczęcie prac związanych z budową urządzeń oraz robotów zarówno profesjonalnych, jak też amatorskich.

Niniejsza praca powstała w wyniku wykorzystania dostępnych technologii do stworzenia konstrukcji autonomicznego pojazdu śledzącego sterowanego cyfrowo. Inspiracją autora była chęć wykonania w pełni autonomicznego pojazdu śledzącego działającego bezawaryjnie zgodnie z własnoręcznie opracowanym algorytmem. Dodatkowym czynnikiem motywującym była chęć opanowania technologii

programowania mikrokontrolerów oraz konstruowania nowoczesnych urządzeń elektronicznych.

Głównym celem niniejszej pracy było opracowanie koncepcji i wykonanie modelu autonomicznego pojazdu śledzącego sterowanego cyfrowo. Aby zrealizować ten cel dokonano przeglądu istniejących obecnie śledzących pojazdów autonomicznych i zaprojektowano własną wersję robota klasy Line Follower.

Zakres pracy w pierwszej kolejności obejmował przegląd obecnie stosowanych rozwiązań w zakresie autonomicznych pojazdów, przedstawienie wymagań stawianych robotowi, pomysł na realizację praktyczną robota z wykorzystaniem obecnie dostępnych technologii. W kolejnym etapie pracy opanowano program Altium Designer, który umożliwił projektowanie obwodów drukowanych oraz symulacje obwodów elektronicznych. Następnym krokiem było wykorzystanie programu SolidWorks do zamodelowania obudowy robota, którą następnie wydrukowano na drukarce 3D.

W zakresie pracy znajdowało się także opanowanie wiedzy z programowania mikrokontrolerów w stopniu wystarczającym do zaimplementowania przygotowanego algorytmu pracy oraz zagwarantowanie odpowiedniej szybkości i bezawaryjności jego realizacji. Dodatkowo wykonano aplikację mobilną z wykorzystaniem środowiska graficznego AppInventor, które stanowi dobrą alternatywę dla klasycznych języków wysokiego poziomu.

2. Konstrukcje autonomicznych robotów pojazdów

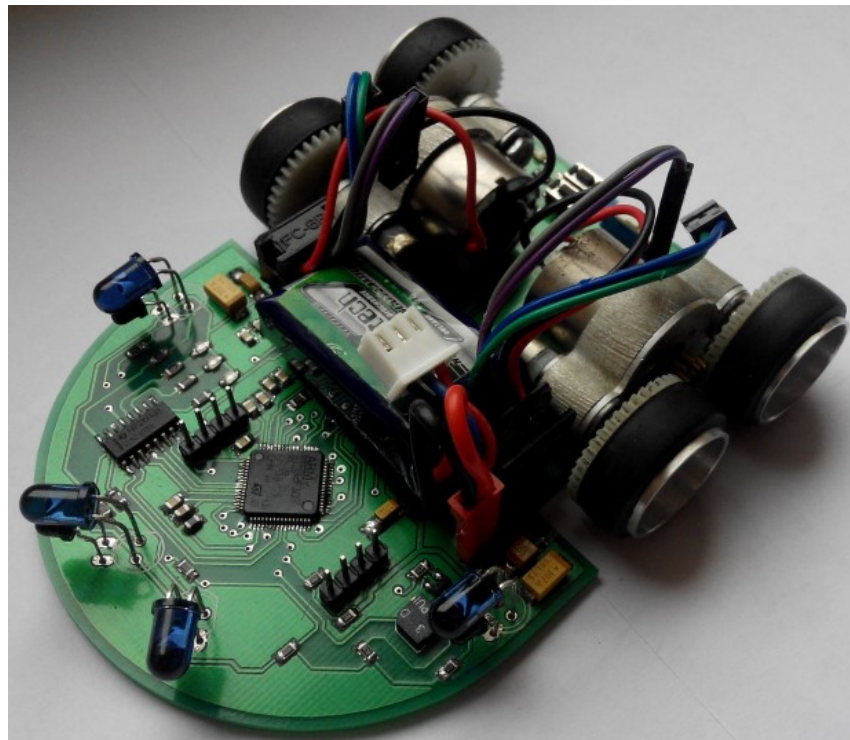
2.1. Popularne zawody robotyczne

W ostatnich latach również w Polsce, przed nastaniem kryzysu wywołanego pandemią Covid-19 w 2020 roku, popularność zyskiwały odbywające się cyklicznie zawody robotów. Podczas zawodów roboty rywalizują ze sobą w różnych konkurencjach. O zwycięstwie decyduje przede wszystkim, odpowiednia konstrukcja i strategia w postaci algorytmów sterujących.

2.1.1. Micromouse

Jednym z typów robotów biorących udział w takich zawodach są roboty mające za zadanie dostać się do środka labiryntu, a następnie możliwie szybko powrócić. Roboty te popularnie nazywa się Micromouse.

Labirynt zgodnie z obecnie panującymi zasadami składa się z 16 x16 komórek o wymiarach 18 x18 każda. Na Rys 2.1 została przedstawiona konstrukcja robota tego typu.



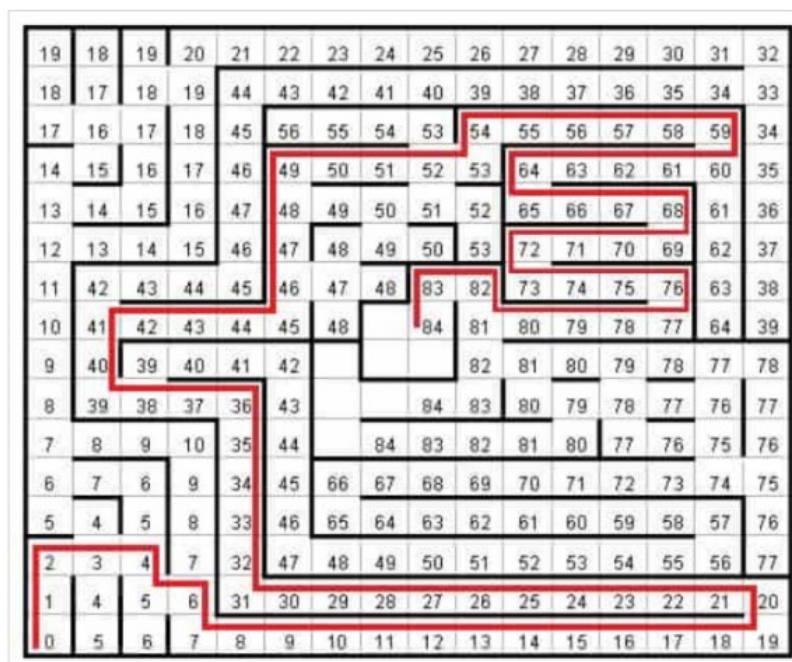
Rys 2.1 Robot typu MicroMouse [2]

Konstruowane roboty, aby dotrzeć do środka labiryntu i wrócić, muszą uzyskiwać informacje o labiryncie oraz posiadać odpowiednie algorytmy. W celu zebrania informacji o labiryncie wykorzystuje się czujniki takie jak:

- czujniki odległości – lokalizacja względem ścian, najczęściej jest to kilka diod IR z fototranzystorem,
- żyroskop - określenia orientacji robota,
- enkodery – pomiar przejechanej drogi przez poszczególne koła.

Robot musi posiadać także odpowiedni mikrokontroler zdolny odpowiednio przetworzyć uzyskane pomiary z czujników, a następnie odpowiednio sterować silnikami odpowiedzialnymi za ruch robota. Najpopularniejszymi silnikami w tych konstrukcjach są silniki Faulhaber 1717T006SR, natomiast najczęściej wybierane mikrokontrolery to STM32 [2].

Najprostszym algorytmem sterującym jest podążanie obok ściany, co w przypadku prostych labiryntów umożliwi dotarcie do celu. Kolejną metodą jest tzw. Brute Force polegające na losowym wybieraniu kierunku jazdy na rozwidleniach. W przypadku ślepej uliczki robot zawraca do rozwidlenia i podąża w innym kierunku. Cykl się powtarza, aż robot dojedzie do celu, powrót również odbywa się na tych zasadach. Zatem robot nie zapamiętał trasy, więc jego przejazdy nie są efektywne. Lepszym podejściem jest zatem zapamiętywanie trasy, co umożliwiają wspomniane enkodery oraz żyroskop. Dlatego też najpopularniejszą metodą wykorzystywaną przez zawodników jest tzw. metoda propagacji fali stanowiąca uproszczenie algorytmu BELLMANA-FORDA. Na Rys 2.2 została przedstawiona najkrótsza trasa opracowana na podstawie omawianego algorytmu.



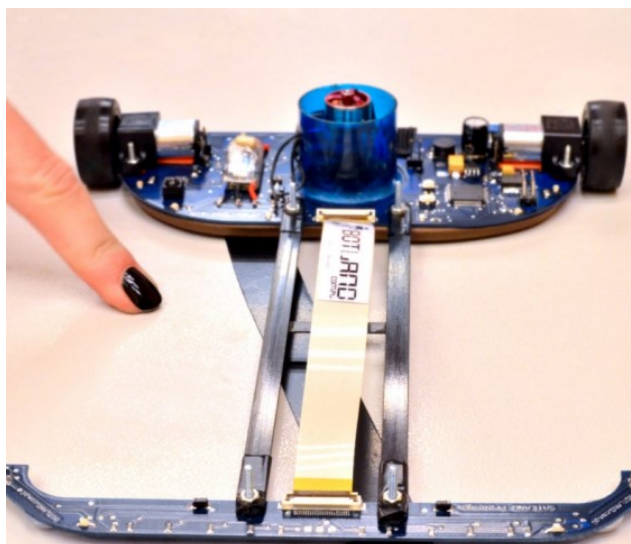
Rys 2.2 Wyznaczanie najkrótszej trasy metodą propagacji fali [3]

W dużym skrócie metoda ta polega na tym, że robot zwiedza cały labirynt, oznaczając kolejne komórki cyframi, te znajdujące się najbliżej punktu początkowego najmniejszymi,

a kolejne większymi. Wyznaczenie najkrótszej drogi odbywa się poprzez odczytanie wartości największej znajdującej się u celu labiryntu do najmniejszej na jego początku [3].

2.1.2. Line Follower

Kolejnym rodzajem robotów śledzących biorącym udział w takich zawodach są roboty typu Line Follower. Czyli roboty śledzące linie, przeważnie czarna na białym tle. Na Rys 2.3 został przedstawiony takowy robot wykonany przez zespół internetowego sklepu Bot land.



Rys 2.3 Line Follower zespołu Botland [4]

Roboty tego typu konkurują ze sobą o to, który z nich w najkrótszym czasie pokona wyznaczoną trasę. Na trasie w zależności od konkurencji mogą się pojawiać utrudnienia w postaci ostrych kątów, krótkich przerw linii prowadzącej, a nawet fizyczne przeszkody w postaci ukształtowania podłoża i ukośnych najazdów [4].

Najważniejsze elementy konstrukcji robotów tego typu to:

- czujniki odbiciowe pozwalające na określenie położenia robota względem linii prowadzącej,
- silniki gwarantujące osiągnięcie dużych prędkości,
- koła zapewniające dobrą przyczepność do podłoża.

Do sterowania oraz obliczenia położenia robota względem linii konieczny jest także odpowiedni mikrokontroler. Przeważnie wybierane są mikrokontrolery Atmel lub STM32. Na szybkość przejazdu robota wpływa kilka czynników. Najważniejsze to zastosowanie odpowiednich silników oraz kół gwarantujących dobrą przyczepność z podłożem. Popularnie wykorzystywanymi silnikami są Micro Silniki Pololu z przekładnią 1:10 pozwalające uzyskać maksymalną prędkość obrotową na poziomie 3 tysięcy na minutę,

natomiast popularnie wykorzystywane koła to Solarbotics RW2i zapewniające wysoką przyczepność [5].

Do określenia położenia względem linii niemal w każdej konstrukcji używane są czujniki KTIR0711S, zgodnie z dokumentacją, przy odległości 1mm od badanego obiektu, oraz prądzie diody 4mA. Pozwalają na pomiary z dużą częstotliwością w zależności od konfiguracji. Działanie czujnika opiera się na emisji światła podczerwonego przez fotodiodę oraz jego detekcji od odbitej powierzchni przez fotokomórkę. W zakresie aktywnym przewodność fotokomórki zależy od ilości światła, które pada na jej powierzchnię – większa ilość światła padającego na fotokomórkę, tym jej przewodność jest większa. Wykorzystując tę właściwość za pomocą odpowiedniego podłączenia czujnika możliwe jest sprawdzenie czy przed nim znajduje się powierzchnia czarna czy biała poprzez pomiar napięcia na kolektorze fotokomórki. Przeważnie moduł z czujnikami z modulem sterującym jest połączony za pomocą taśm IDC lub FFC, takie rozwiązanie pozwala szybką korektę odległości czujników od osi kół prowadzących oraz ewentualną modyfikację modułu z czujnikami bez konieczności zmian w module sterującym [6].

Wyposażenie robota w dodatkowe enkodery na silnikach oraz żyroskop pozwala na efektywniejsze sterowanie poprzez zwiększanie prędkości na prostych odcinkach lub redukcję krzywizn występujących na trasie. Najprostsza implementacja enkoderów do wspomnianych silników Pololu jest możliwa za pomocą dedykowanych w tym celu modułów. Ich montaż odbywa się bezpośrednio na wale. Na Rys 1.3 w środkowej części robota widoczny jest również napęd tunelowy EDF27 firmy HobbyKing. Stosuje się go, aby zapewnić lepszą przyczepność robota do podłoża wskutek zasysania powietrza spod robota [5].

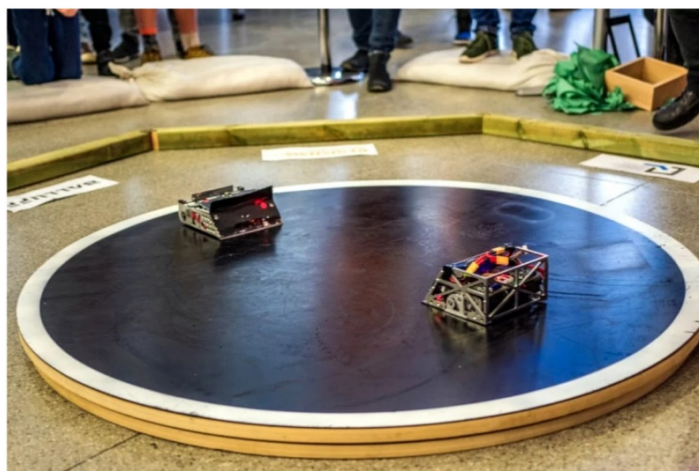
2.1.3. Walki Robotów Sumo

Ciekawą konkurencją są także walczące roboty Sumo. Konkurencja robotów Sumo polega na tym, aby wypchnąć robota przeciwnika z ringu zwanym dohyo jakim jest czarny okrąg, na którego obrzeżach znajdują się biała linia. Na Rys 2.4 zostały przedstawione roboty podczas walki na zawodach Robotic Arena Wrocław z 2020 r. Walka przeciętnie trwa 5-6 s, jednak potrafi w tym czasie dostarczyć wielu emocji konstruktorom, a także widzom.

Najważniejsze elementy robotów typu Sumo to:

- koła gwarantujące wysoką przyczepność,
- odpowiednie silniki,
- czujniki odległości, lokalizujące przeciwnika, na przykład GP2Y0D340K,

- czujniki odbiciowe lokalizujące krawędź ringu, KTIR0711S lub CNY70.



Rys 2.4 Walka robotów Sumo podczas zawodów Robotnic Arena Wrocław 25.01.2020 źródło: <https://forbot.pl/blog/zawody-robotow-xii-robotic-arena-wroclaw-25-01-2020-id39198>

Najlepsze konstrukcje posiadają klin i zaostrenie z jednej strony, naprzeciw której znajdują się koła. W przypadku robotów Sumo konstruktorzy często wykonują koła samodzielnie z wykorzystaniem drukarek 3D i silikonu formierskiego. Do napędu robota wybierane są najczęściej silniki Pololu z przekładnia, 1:30 pozwalające uzyskać prędkość na poziomie 1000obr./min z momentem 0,65Nm przy zasilaniu 6V. Oczywiście do odpowiedniej interpretacji pomiarów z czujników oraz implementacji algorytmu wymagany jest odpowiedni mikrokontroler [7].

Obecnie na większości zawodów walka robotów rozpoczyna się poprzez wysłanie sygnału przez sędziego za pomocą pilota na podczerwień działającego w standardzie RC5. Eliminuje to konieczność synchronizacji ruchów zawodników, gwarantując start robotów w niemal identycznych momentach [7].

2.2. Zaawansowane autonomiczne pojazdy

Oprócz wspomnianych konstrukcji pojazdów występują również komercyjne roboty ułatwiające życie codzienne. Konstrukcje takie jak automatyczne odkurzacze i kosiarki pozwalają zaoszczędzić czas jako pomoc domowa.

Trwają również intensywne prace nad stworzeniem w pełni autonomicznego samochodu. Obecnie nie są one dostępne, ale producenci przedstawiają swoje rozwiązania koncepcyjne w tym zakresie. Producent Tesla wprowadził do sprzedaży komercyjnej pojazdy z autopilotem potrafiące dobrać prędkość oraz podążać za torem jazdy, jednak system wymaga stałego nadzoru człowieka. Projekt Waymo zdaje się mieć także ciekawą ofertę. W jednym z miast w Stanach Zjednoczonych przy pomocy aplikacji mobilnej

można zamówić autonomiczny pojazd bez kierowcy, który dowiezie pasażera do wskazanego celu.

3. Projekt i realizacja robota śledzącego

3.1. Wnioski z przeglądu konstrukcji autonomicznych robotów pojazdów

Przedstawione w poprzednim rozdziale przykłady konstrukcji autonomicznych pojazdów śledzących umożliwiają zapoznanie się z metodami konstruowania oraz sterowania robotami tego typu. Stwierdzenie, że wiedza i doświadczenie zdobyte w skutek konstruowania prostych konstrukcji będzie użyteczna podczas budowy wspomnianych zaawansowanych pojazdów, jest z pewnością bliskie prawdy.

Po wykonaniu powyższego przeglądu zdecydowano się wykonać pojazd podążający za czarną linią stanowiącą graficzną drogę prowadzącą, czyli wspomnianego wcześniej Line Followera. Robot tego typu wydaje się być ciekawy oraz możliwy do wykonania bez dużych nakładów finansowych. Podobne roboty tego typu wykorzystywane są również w przemyśle w celu automatyzacji transportu [8].

3.2. Ograniczenia oraz wymagania konstrukcyjne wynikające z regulaminu zawodów

Aby umożliwić udział robota w najpopularniejszych w Polsce zawodach, kategorii LineFollower Light [9] należy w pierwszej kolejności zapoznać z ich regulaminem i dostosować do niego konstrukcje oraz możliwości pojazdu.

Regulamin ten co do konstrukcji robota wprowadza w zasadzie dwa ograniczenia. Po pierwsze, jego obrys powinien mieścić się na kartce papieru formatu A4 z tolerancją 5%. Po drugie, nie może być on wyposażony w napęd tunelowy lub inne urządzenie aktywnie zwiększające przyczepność. Ponadto, regulamin mówi o tym, że w trakcie działania robot powinien być niewrażliwy na warunki oświetlenia, zadymienia, efektów dźwiękowych. Działanie pojazdu powinno odbywać się w sposób autonomiczny, komunikacja z robotem podczas przejazdu jest zabroniona. Wyjątkiem od tej reguły jest zdalne startowanie i zatrzymywanie pojazdu.

W regulaminie określona jest także następująca specyfika trasy:

- trasa przejazdu wyznaczana jest przez czarną linię o szerokości do 2 cm umieszczoną na jasnym tle,

- na trasie nie występują uskoki, przerwy, wzniesienia, rozwidlenia i inne przeszkody,
- w odległości 210 mm od linii nie znajduje się inna linia,
- na trasie mogą pojawiać się skrzyżowania, które należy przejechać na wprost oraz kąty proste,
- obszar trasy określony jest przez prostokąt.

Występowanie kątów ostrów na trasie wymaga, żeby robot był zwrotny. Środek ciężkości powinien znajdować się możliwie nisko, blisko osi silników, co pozwoli na szybszą zmianę kierunku jazdy.

3.3. Podstawowe parametry robota

3.3.1. Położenie względem linii

Podstawowym parametrem robota LineFollower jest jego położenie względem linii wyznaczającej jego trasę. Moduł do pomiaru położenia robota względem linii powinien być mało wrażliwy na zakłócenia spowodowane warunkami oświetlenia oraz pomiary powinny być wykonywane z dużą częstotliwością przy zachowaniu niezbędnej precyzji. Moduł powinien także umożliwiać możliwie płynne określenie położenia linii prowadzącej. Obecnie jest to realizowane niemal w każdej konstrukcji poprzez zastosowanie od około 8 do 20 czujników obciowych KTIR0711S, które spełniają stawiane wymagania modułu z czujnikami.

3.3.2. Sterowanie ruchem

Utrzymanie w linii toru robota wymaga sprawnego i dokładnego sterowania. Jest to możliwe poprzez umieszczenie enkoderów na silnikach poruszających jego koła. Dzięki temu można wyznaczyć ich chwilową rzeczywistą prędkość i dopasować ją do aktualnej sytuacji na torze jednocześnie redukując poślizg. Takie sterowanie jest również mniej wrażliwe na zmiany zasilania, na przykład w skutek spadku napięcia zasilania [5].

W pracach [5,10-11] opisano ulepszenia, wspomnianego powyżej sterowania w postaci nieco bardziej zaawansowanych algorytmów poprawiających czas przejazdu. Aby umożliwić ich implementację, należy zastosować dodatkowo moduł do nawigacji inercyjnej. Gdy w konstrukcji robota zawarte są wspomniane elementy, pozwala to na:

- sporządzenie mapy trasy w ramach przejazdu testowego,
- optymalizację kształtu trasy,
- projekt profilu prędkości,
- finalny przejazd.

Wykonanie tych czynności sprawia, że sterowanie robotem jest zdecydowanie bardziej efektywne. W związku tym należy zastosować mikrokontroler z nieco większą ilością pamięci ram oraz flash, ze względu na prawdopodobieństwo potrzeby większych zasobów sprzętowych.

3.4. Napęd oraz konstrukcja mechaniczna

Chcąc osiągać jak najlepsze wyniki, należy również wykorzystać silniki zapewniające wysoki moment oraz dużą prędkość obrotową, a także koła o wysokiej przyczepności, aby robot nie wpadał w poślizg. Ważne jest także, aby całkowita masa robota nie była zbyt duża. Nadmiar masy wprowadza bezwładność, co sprawia, że robotem trudniej sterować, zwłaszcza w momentach, gdy na trasie pojawiają się kąty ostre. Z tego powodu popularna jest konstrukcja zbudowana zasadniczo z dwóch części:

- płytki z elektroniką sterującą oraz dwoma silnikami na jednej osi oddalonymi od siebie o około 10 -15 cm,
- możliwie lekkiej listwy z czujnikami.

W najszybszych konstrukcjach części te połączone są ze sobą lekką listewką oraz taśmą FFC lub IDC. Takie konstrukcje są lekkie oraz znajdują się w kategorii robota mobilnego klasy 2.0. Umożliwia to dużą manewrowość oraz łatwość sterowania.

Kolejne wymagania stawiane konstrukcji:

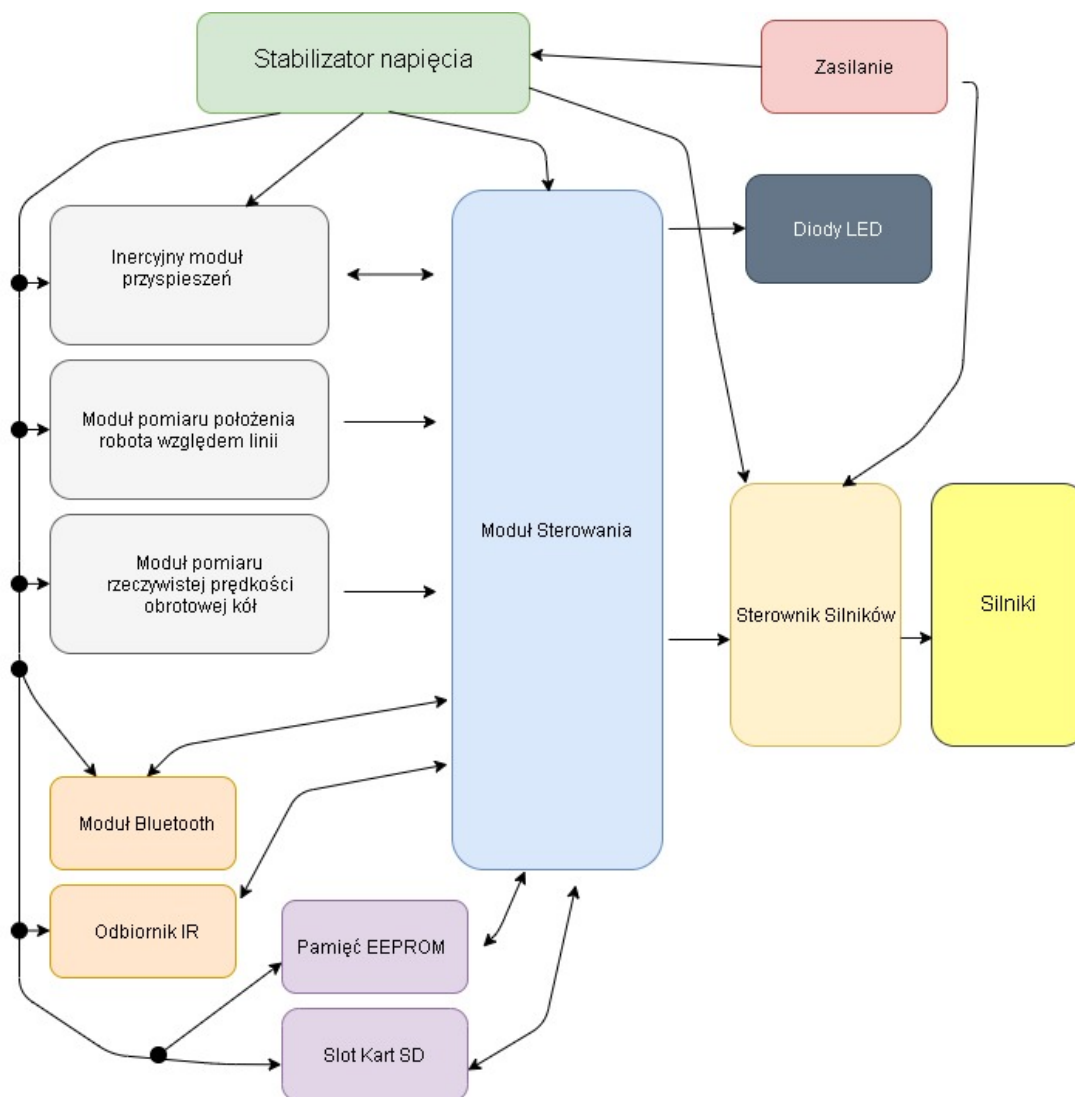
- wysoka odporność na ewentualne zderzenia z otoczeniem, co może się zdarzyć w przypadku utraty przyczepności,
- zabezpieczenie przed zwarcieniem, wskutek przypadkowego kontaktu z metalowymi przedmiotami,
- estetyczny wygląd, co poprawi pozytywny odbiór całości konstrukcji przez obserwatorów.

Do realizacji przedstawionych powyżej wymagań odpowiednia będzie dedykowana lekka obudowa, która nie powinna znacząco negatywnie wpływać na czasy przejazdu i ogólne działanie całej konstrukcji.

3.5. Ogólna koncepcja budowy robota

Przechodząc do realizacji projektu w pierwszej kolejności opracowano ogólny własny schemat budowy robota. Został on przedstawiony na Rys 3.5. Zaznaczono tu moduły wejść, które wykonują pomiary niezbędne do sterowania mikrokontrolera, który z kolei steruje pośrednio elementami wykonawczymi w postaci silników. Ze względów

praktycznych w ramach projektu zdecydowano również umieścić moduł Bluetooth oraz odbiornik IR, by mieć szerokie możliwości modyfikacji parametrów algorytmu sterującego.



Rys 3.5 Blokowy schemat budowy robota

Zastosowanie stabilizatora napięcia zagwarantuje odpowiedni poziom napięcia oraz jego stabilność. Diody LED poprawiają widoczność robota, zwłaszcza w warunkach słabego oświetlenia.

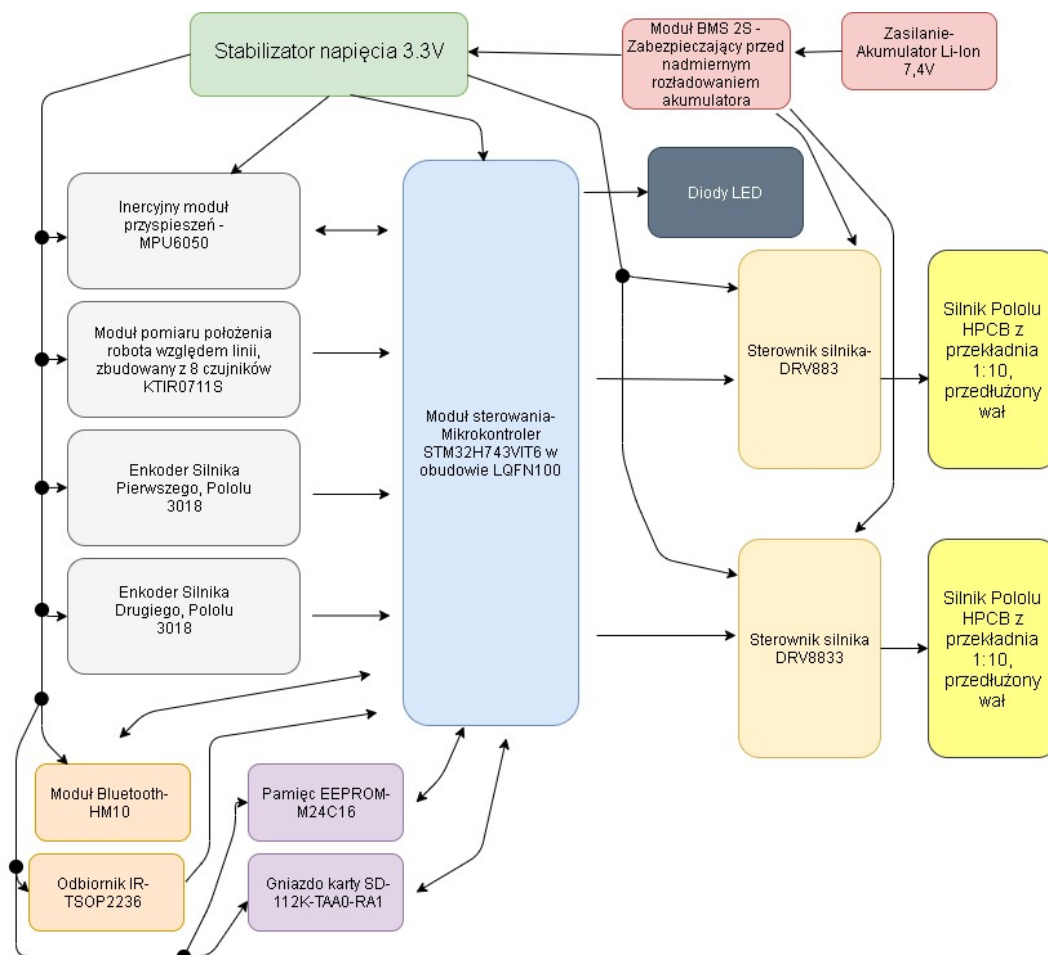
Zdecydowano umieścić także dodatkowy moduł pamięci EEPROM, który umożliwi zapamiętanie wielokrotnych zmian parametrów algorytmu za pomocą wspomnianego wcześniej modułu Bluetooth. Pojazd ma być również wyposażony w slot dla kart SD. W celu gromadzenia dużej ilości danych potrzebnych do jego sterowania.

3.6. Projekt części elektronicznej

Na podstawie opracowanego ogólnego schematu budowy robota (Rys 3.5) dokonano wyboru elementów pozwalających na realizację przedstawionej koncepcji

zgodnie z wymaganiami projektowymi. Rysunek 3.6 przedstawia schemat blokowy ilustrujący ogólnie sposób połączenia wybranych elementów elektronicznych. Aby zrealizować

w sposób praktyczny przedstawioną koncepcję zaprojektowano dwie płytki obwodów drukowanych. Do przygotowania obwodów drukowanych wykorzystano program Altium Designer. Jest to bardzo rozbudowane narzędzie umożliwiające tworzenie profesjonalnych obwodów drukowanych. Na tle konkurencji wyróżnia się przede wszystkim tym, że jest zintegrowany z ofertą producentów elektroniki, co pozwala na szybkie wyszukiwanie elementów elektronicznych i ich bibliotek. Istnieje także możliwość wyeksportowania plików trójwymiarowych zaprojektowanego obwodu drukowanego w standardowym formacie wymiany danych produktów STEP. Pozwala to na łatwiejsze projektowanie obudowy i jej dopasowanie w innym programie do modelowania przestrzennego.

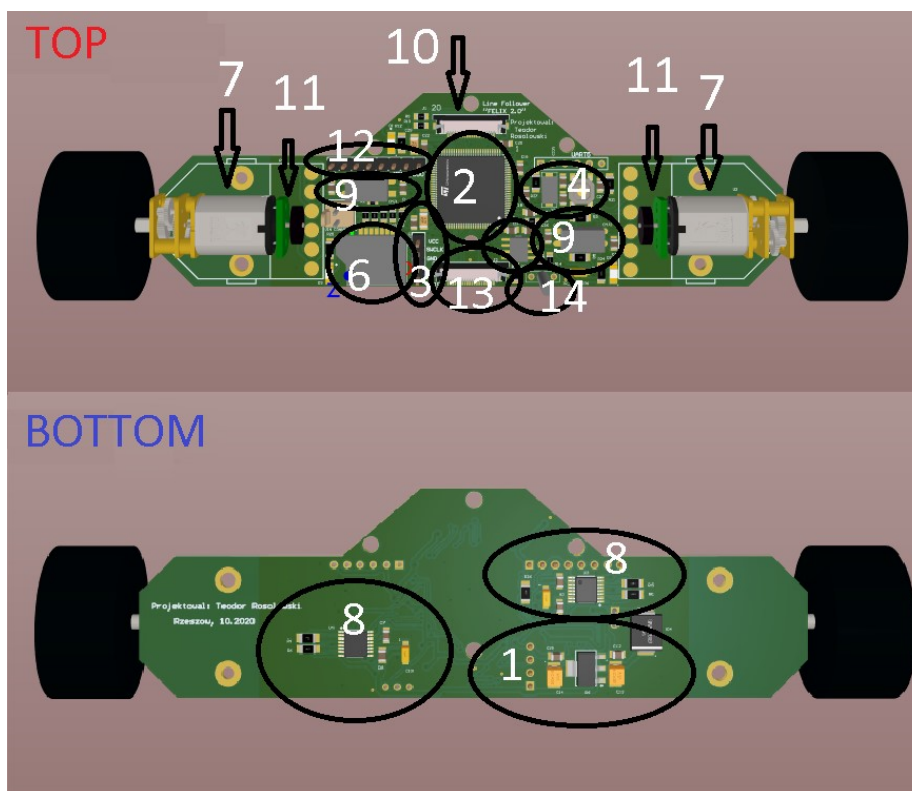


Rys 3.6 Schemat ideowo-koncepcyjny projektowanego układu elektronicznego robota

Na pierwszej z przygotowanych płytek umieszczono większość z zaplanowanych modułów elektronicznych, uwzględniono także otwory do montażu silników i obudowy. Podczas projektu tej płytki umieszczono na niej modele 3D silników i enkoderów, aby

zredukować możliwość popełnienia ewentualnego błędu w dalszej części prac projektowych. Wizualizacja 3D tej płytki została przedstawiona na Rys 3.7, natomiast cały schemat elektryczny tej płytki został umieszczony w dodatku A. Na wizualizacji zaznaczono również kolejnymi cyframi elementy płytki zgodnie z poniższą listą:

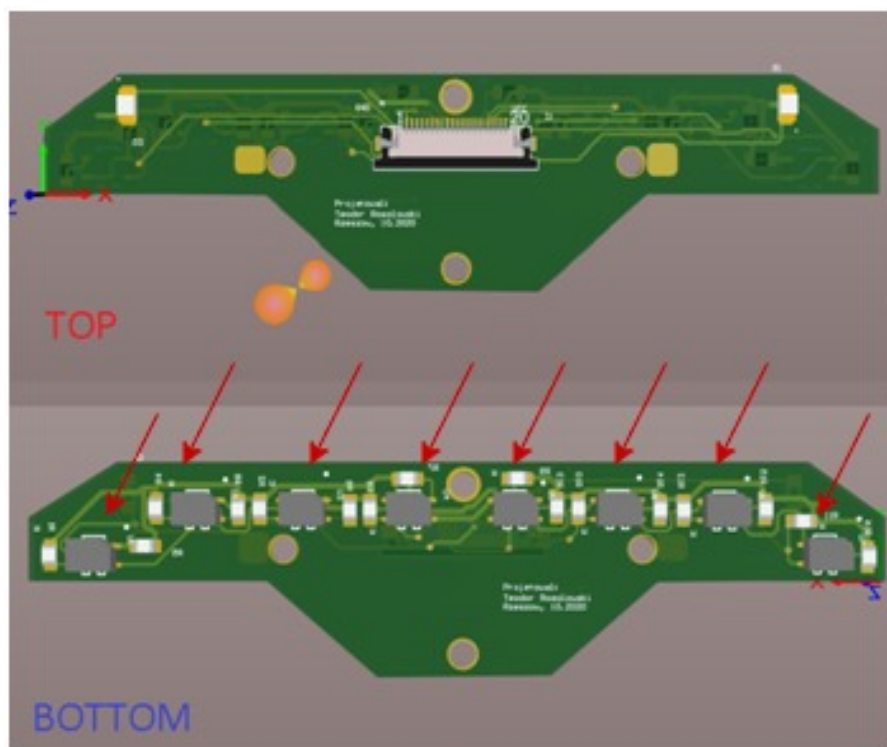
- 1) Stabilizator napięcia 3,3V,
- 2) Moduł Sterowania- Mikrokontroler STM32H743VIT6,
- 3) Wyprowadzenia programatora SWD,
- 4) Rezonator 16MHz oraz Przycisk Reset,
- 5) Pamięć EEPROM- M24C16R,
- 6) Gniazdo Kart SD,
- 7) Micro Silniki Pololu z przekładnia 1:10 z przedłużonym wałem,
- 8) Sterowniki silników DRV8833,
- 9) Filtry Przeciw zakłóceń CMJ2101 Talema,
- 10) Konektor modułu z czujnikami,
- 11) Enkodery inkrementalne Pololu 3081,
- 12) Konektor Modułu Akcelometru MPU-6050,
- 13) Konektor modułu Bluetooth – HM-10,
- 14) Scalony odbiornik sygnału IR- TSOP2236.



Rys 3.7 Wizualizacja 3D płytki z modułem sterującym wraz z rozmieszczeniem elementów

3.7. Orientacja robota na torze

Druga z przygotowanych płytek w praktyczny sposób realizuje moduł pomiaru położenia pojazdu względem linii będącej torem jazdy. Jej wizualizacja została przedstawiona na Rys 3.8. Na tej wizualizacji czerwonymi strzałkami wskazano położenie czujników odbiciowych KTIR0711S odpowiedzialnych za detekcję linii. Cały schemat elektryczny tej płytki został umieszczony w dodatku B.



Rys 3.8 Płytki modułu pomiaru położenia pojazdu na torze.

Połączenie z płytką zawierającą moduł sterujący odbywa się za pomocą konektora (patrz konektor zaznaczony cyfrą 10 na rys.2.7) przy użyciu 20 pinowej taśmy FFC, o rastrze 0,5mm. Na omawianej płytce przygotowano otwory, za pomocą których możliwy będzie montaż obudowy.

3.8. Zasilanie robota

Zasilanie robota w całości zostało zrealizowane za pomocą akumulatora z ogniwami Li-ion firmy DualSky, o pojemności 250mA o nominalnym napięciu wyjściowym 7.4V. Wybrany akumulator jest lekki oraz powinien zapewnić niezbędny czas pracy.

W celu zabezpieczenia akumulatora przed rozładowaniem na linii zasilania pomiędzy płytką z modułem sterującym a baterią umieszczono moduł BMS 2S. Pozwoli to

na długotrwałe korzystanie z baterii dzięki zapobieganiu, jej przypadkowemu nadmiernemu rozładowaniu.

W skład modułu zasilania wchodzi stabilizator napięcia 3.3V zapewniający odpowiedni poziom napięcia do zasilania mikrokontrolera oraz pozostałych układów peryferyjnych.

3.9. Moduł główny sterowania

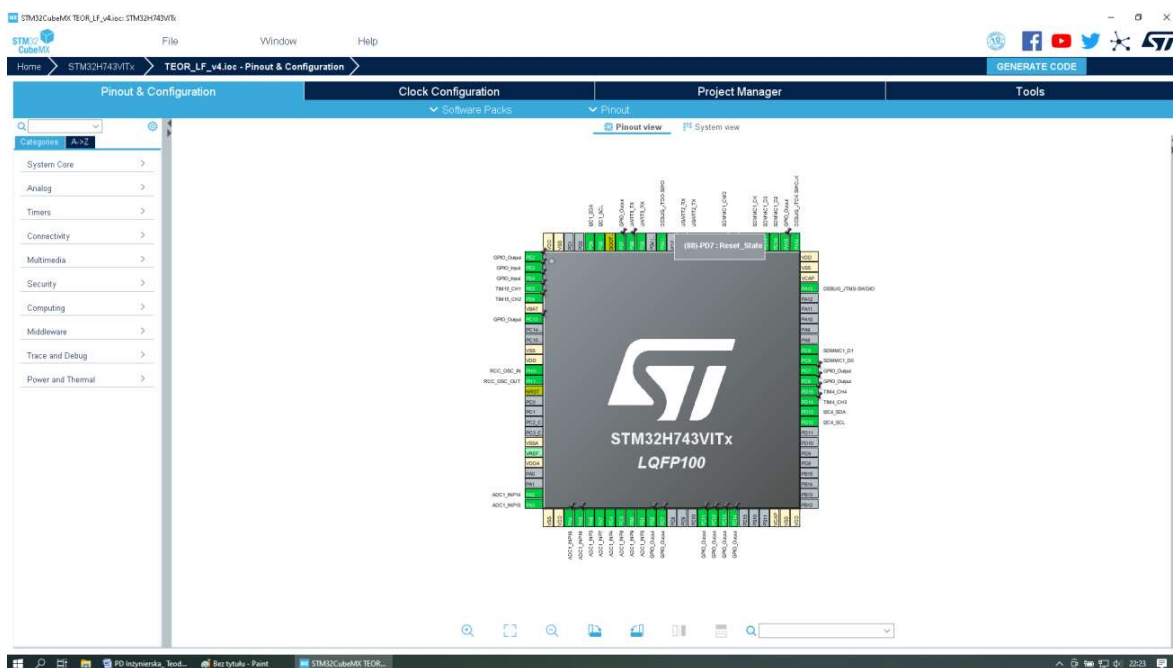
Do sterowania robotem wybrano Mikrokontroler STM32H743VIT6 w obudowie LQFN100 ze względu na łatwość montażu. Jest to jeden z nowszych produktów firmy ST produkowany od kwietnia 2019 roku. Główną jego zaletą jest bardzo duża częstotliwość taktowania rdzenia wynosząca 480Mhz. Mikrokontroler jest wyposażony także między innymi w:

- 2Mb pamięci flash,
- 1MB pamięci RAM,
- 16-bitowe przetworniki A/C – niezbędne do przetworzenia informacji od czujników odbiciowych,
- Rozbudowane 16-bitowe timery, wymagane do sterowania silnikami oraz do pomiaru czasu z różną dokładnością,
- Kanały bezpośredniego dostępu do pamięci DMA.

Obecnie osoby programujące mikrokontrolery STM32 mają do wyboru kilka środowisk, są to m.in. „System Workbench for STM32, Atollic True Studio for STM32 KeilVision oraz STM32CubeIDE. Ostatnie wymienionych wyróżnia się przede wszystkim tym, że jest wydane przez producenta mikrokontrolera.

Duża ilość układów peryferyjnych powoduje to, że przygotowawczy etap projektu polegający na ich konfiguracji jest czasem bardziej pracochłonny niż napisanie właściwej aplikacji sterującej. Obecnie w przypadku mikrokontrolerów STM32 można przyspieszyć ten proces za pomocą programu STM32CubeMX. Program ten pozwala szybko skonfigurować peryferia mikrokontrolera za pomocą myszy, a następnie na wygenerowanie odpowiedniego kodu w języku C na tej podstawie. Rysunek 3.9 przedstawia widok programu STM32CubeMX po wstępnej konfiguracji wykorzystywanych pinów wybranego mikrokontrolera. Z lewej strony programu jest menu, w którym wybiera się peryferia, które chcemy skonfigurować. W środkowej części widać rozmieszczenie nóżek wyjściowych/wejściowych, natomiast w zakładce

ClockConfiguration dostępne są ustawienia zegarów oraz pętli PLL. W zakładce Project Manager dostępny jest wybór środowiska wyjściowego i kilka zaawansowanych ustawień. Przycisk GENERATE CODE sprawia, że proces generacji kodu wyjściowego pod wybrane środowisko się rozpoczyna i po chwili w wybranej lokalizacji znajdują się nowy projekt z skonfigurowanymi peryferiami. Wygenerowany kod ma pewien stopień abstrakcji czyli ukrywa przed użytkownikiem działania na rejestrach konfiguracyjnych, stąd nazwa biblioteki z gotowymi funkcjami upraszczającymi proces programowania - HAL – *Hardware Abstraction Layer*. Dzięki temu do dyspozycji jest wiele gotowych funkcji i za ich pomocą można wydawać polecenia mikrokontrolerowi. Dokumentacja całej biblioteki HAL dla Dokumentacja całej biblioteki HAL dla mikrokontrolerów STM32H7xx jest dostępna na stronie producenta. Stanowi ona obszerny dokument pokazujący i objaśniający dostępne funkcje, które można wykorzystać w swoim projekcie[12].



Rys 3.9 Widok programu STM32CubeMX, po wstępnej konfiguracji pinów wybranego mikrokontrolera

3.10. Dodatkowa pamięć operacyjna

Zgodnie z przedstawioną koncepcją teoretyczną na obwodzie drukowanym umieszczono scalony układ pamięci EEPROM - M24C16 firmy STMicroelectronics. Zastosowana pamięć pozwoli zapisywać wartości zmiennych programowych podczas działania programu, aby po wyłączeniu zasilania ponownie je odczytać. Dzięki takiemu

podejściu istnieje możliwość prostej wielokrotnej i stałej zmiany parametrów wgranego algorytmu bez potrzeby ponownego wgrywania całego programu do mikrokontrolera [12].

Zastosowany Slot 112K-TAA0-RA1 pozwala umieścić kartę microSD. Slot jest mały i nie zajmuje dużo miejsca na obwodzie drukowanym. Wybrany mikrokontroler ma sprzętową obsługę kart SD, co uprości korzystanie z niej w razie takiej potrzeby.

3.11. Sterowanie napędem robota

Sterowanie silnikiem prądu stałego nie jest możliwe bezpośrednio z portu mikrokontrolera ze względu na jego ograniczenia prądowe. Stosuje się więc odpowiednie układy do tego celu, w elektronice umownym określeniem nazywającym taki układ sterujący jest mostek H (ang. H-bridge). W związku z tym wybrano scalony mostek DRV8833. Pozwala on na sterowanie jednocześnie dwoma silnikami prądu stałego. Posiada również zabezpieczenia nadprądowe, przepięciowe oraz temperaturowe. Zabezpieczenia te zapewniają wysoką niezawodność mostka. Mostek dostępny jest w kilku obudowach, w zależności od zastosowanej obudowy jego parametry są nieco inne. Zdecydowano się wykorzystać obudowę PWP16 ze względu na łatwość montażu [13].

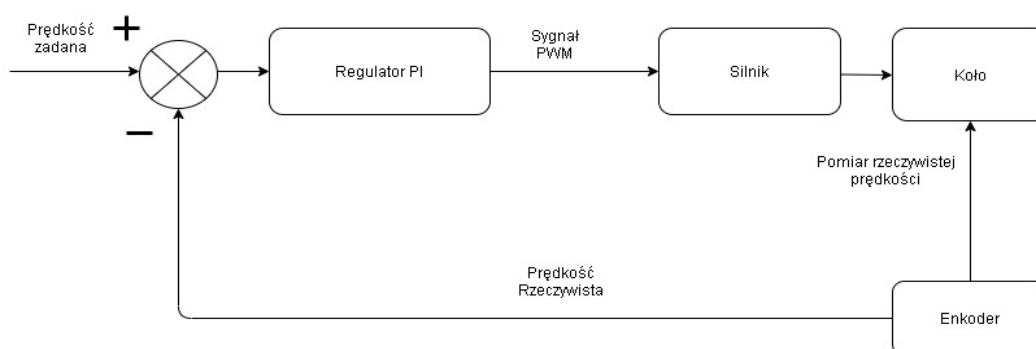
Wykorzystano dwa takowe układy, ponieważ wybrane silniki w przypadku zatrzymania wału osiągają prąd o wartości 1,6A przy napięciu 6V, a bateria zasilania ma znamionowe napięcie 7.4V, więc wartość prądu może być jeszcze większa. Pojedynczy mostek pozwala na maksymalny prąd wyjściowy na kanał wynoszący 1,5A, natomiast w trybie równoległym - 3A. Wykorzystanie trybu równoległego pozostawia sporą rezerwę wydajności prądowej mostka. Zastosowany mostek H, wspiera sterowanie prędkością obrotową silnika za pomocą sygnału PWM – (ang. *PulseWave*). Jest to metoda polegająca na zmianie wypełnienia sygnału sterującego przeważnie o przebiegu prostokątnym przy zachowaniu stałej amplitudy i częstotliwości. Taki sposób sterowania powoduje, że średnia wartość napięcia na wejściu silnika zależy od wartości wypełnienia - D , okresu - T , Amplitudy – A , b – szerokości impulsu podawanego na wejścia silnika zgodnie z wyrażeniem:

$$U_{sr} = A \cdot \frac{b}{T} \quad (3.11)$$

Ze względu na stosunkowo duży szum jaki generują silniki prądu stałego, który mógłby negatywnie wpływać na mikrokontroler, zdecydowano się go zredukować. W tym celu zastosowano dławik drutowy CMJ2101 firmy TALEMA stosowany jako element przeciwzakłóceńowy. Maksymalny prąd pracy według producenta wynosi 500mA. Jednak podczas wykonywanych testów nie udało się uszkodzić dławika podczas współpracy

z wybranym silnikiem przy zasilaniu napięciem do 8.4V. Na Rys 3.7 zostały one zaznaczone cyfrą 9.

Jak zostało to przedstawione w wymaganiach projektowych, aby efektywnie sterować napędem robota, należy uzyskiwać informacje o rzeczywistej prędkości obrotowej kół. Do skutecznego osiągnięcia rzeczywistej prędkości zadanej odpowiedni okazuje się regulator PI ze sprzężeniem zwrotnym w postaci wspomnianej informacji o rzeczywistej prędkości obrotowej kół. Idee takiego sterowania przedstawia Rys. 3.11. Dzięki temu, że zostały wybrane zostały silniki HPCB z dwustronnym wałem, przekładnią 1:10, możliwy jest prosty montaż enkoderów inkrementalnych Pololu 3018, dedykowanych przez producenta do tych silników.



Rys 3.11 Sterownik rzeczywistej prędkości obrotowej kół.

Wybrane enkodery pozwalają na uzyskanie maksymalnej rozdzielczości 12 impulsów na jeden obrót wału silnika. Przekładnia silnika wynosi dokładnie 1:9,96. Można więc uzyskać 119,6 impulsów na jeden pełen obrót koła. Aby dokonać konwersji pojedynczego impulsu d_{imp} na dystans S należy skorzystać z wyrażenia:

$$S = \frac{d_{imp}}{n} \quad (3.11)$$

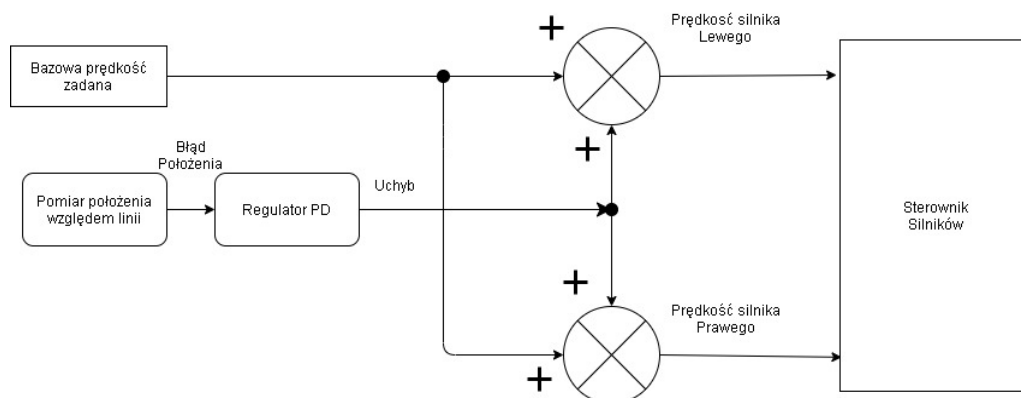
Gdzie n to ilość impulsów na pełen obrót, d_{imp} to średnica koła.

Na tym etapie prac zdecydowano, że zostaną wykorzystane koła Solarbotics RW2i gwarantujące dobrą przyczepność do podłoża. Umożliwiło to obliczenie, że na jeden impuls enkodera wartość przebytej drogi wyniesie około 0,79mm.

3.12. Algorytm sterowania

Algorytm, jaki zostanie zaimplementowany w pojeździe w celu szybkiego i stabilnego podążania za linią w postaci schematu blokowego, został przedstawiony na Rys. 3.12. Jego działanie opiera się o regulator PD, który to z kolei modyfikuje wartość

Aby zwiększyć efektywność takiego sterowania, zostanie wykorzystany tak zwany profil prędkości przedstawiony w pracy [5]. Metoda ta polega na zwiększaniu zadanej prędkości na prostych odcinkach i zmniejszaniu jej przed ostrymi zakrętami. Do realizacji tego celu niezbędna jest cyfrowa reprezentacja mapy trasy oraz przebyty dystans od położenia startowego.



We wspomnianej pracy przedstawiono wyrażenie, za pomocą którego można uzyskać informacje o orientacji i położeniu środka osi pojazdu w układzie współrzędnych kartezjańskich:

$$\left\{ \begin{array}{l} x_i = x_{i-1} + \frac{1}{2} \cos(\theta_{i-1}) \cdot (\Delta l_r + \Delta l_l) \\ y_i = y_{i-1} + \frac{1}{2} \sin(\theta_{i-1}) \cdot (\Delta l_r + \Delta l_l) \\ \theta_i = \theta_{i-1} + \frac{1}{b} \Delta l_r + \Delta l_l \end{array} \right\} \quad (3.12)$$

Aby uzyskać informacje o przebytej drodze pojazdu w i -tej iteracji [10] należy skorzystać z wyrażenia:

$$T(i) = \frac{d_{L,i} + d_{R,i}}{2} \quad (3.13)$$

Aby uzyskać informacje o przebytej drodze pojazdu w i -tej iteracji [11] należy skorzystać z wyrażenia:

$$T(i) = \frac{d_{L,i} + d_{R,i}}{2} \quad (3.14)$$

Gdzie T - translacja, $d_{L,i} + d_{R,i}$ - dystans pokonywany odpowiednio przez koło lewe i prawe. Natomiast aby uzyskiwać aktualną wartość przebytej drogi S w i -tej iteracji, należy tę wartość sumować :

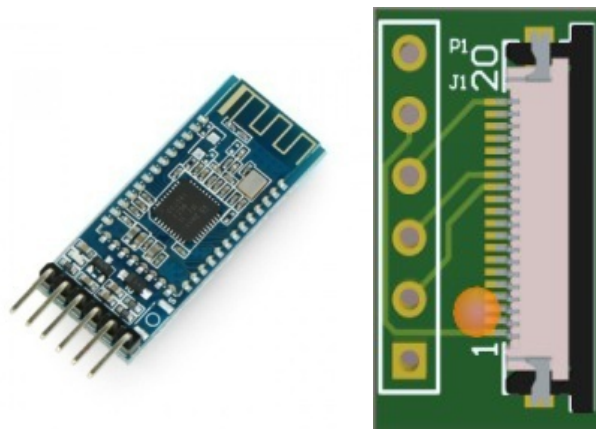
$$S(i) = S(i - 1) + T(i) \quad (3.15)$$

Przedstawione powyżej metody pomiarowe cechują się błędem kumulującym wraz z przebytą odległością. Mimo to, zebrane dane powinny pozwolić na szacowanie w jakiej odległości od startu należy spodziewać się zakrętów i odcinków prostych.

3.13. Bezprzewodowa komunikacja

Bezprzewodowa komunikacja z poruszającym się pojazdem została wprowadzona w celu wygodnej komunikacji z robotem. Takie podejście umożliwia wygodną zmianę parametrów działania algorytmu, diagnostykę poszczególnych peryferiów robota, pobranie danych do pliku.

Połączenie modułu Bluetooth z modułem sterującym zostanie zrealizowane za pomocą taśmy FFC oraz małej płytki pomocniczej, co pozwoli na prosty montaż modułu oraz wybór dogodnego położenia (Rys 3.13).



Rys 3.13 Moduł Bluetooth z dedykowanym obwodem drukowanym do jego podłączenia

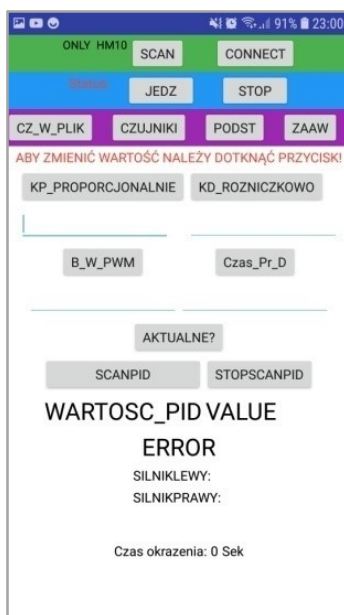
Po przeprowadzonych testach połączenie z modułem jest utrzymywane w odległości do około 15m. W praktyce jest to w zupełności wystarczająca wartość. Moduł Bluetooth skonfigurowano, aby komunikował się z maksymalną możliwą prędkością 115,2 kbps.

Zastosowano również odbiornik podczerwieni TSOP2236. Co zostanie wykorzystane do zmiany parametrów algorytmów, zatrzymywania i startowania robota. W praktyce w większości sytuacji korzystanie z pilota jest wygodniejsze, jednak nie ma wówczas możliwości odczytu żadnych parametrów.

3.13.1. Aplikacja mobilna

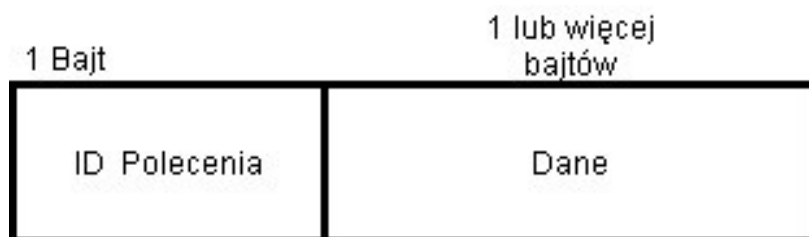
W celu łatwiejszej komunikacji z pojazdem przygotowano za pomocą środowiska AppInventor mobilną aplikację. Za jej pomocą odbywać się będzie połączenie modułem Bluetooth zainstalowanym w pojeździe. Na Rys 3.14 przedstawiony jest widok przygotowanej aplikacji, umożliwi ona m.in.:

- sprawdzenie i zmianę ustawień stosowanego algorytmu sterowania,
- sprawdzenie działania czujników,
- określenie wag błędów poszczególnych czujników,
- wybór wartości odczytu kiedy algorytm powinien uznać, że wykryta przez dany czujnik została linia czarna,
- zapis otrzymywanych danych do pliku tekstowego.



Rys. 3.14 Widok aplikacji mobilnej

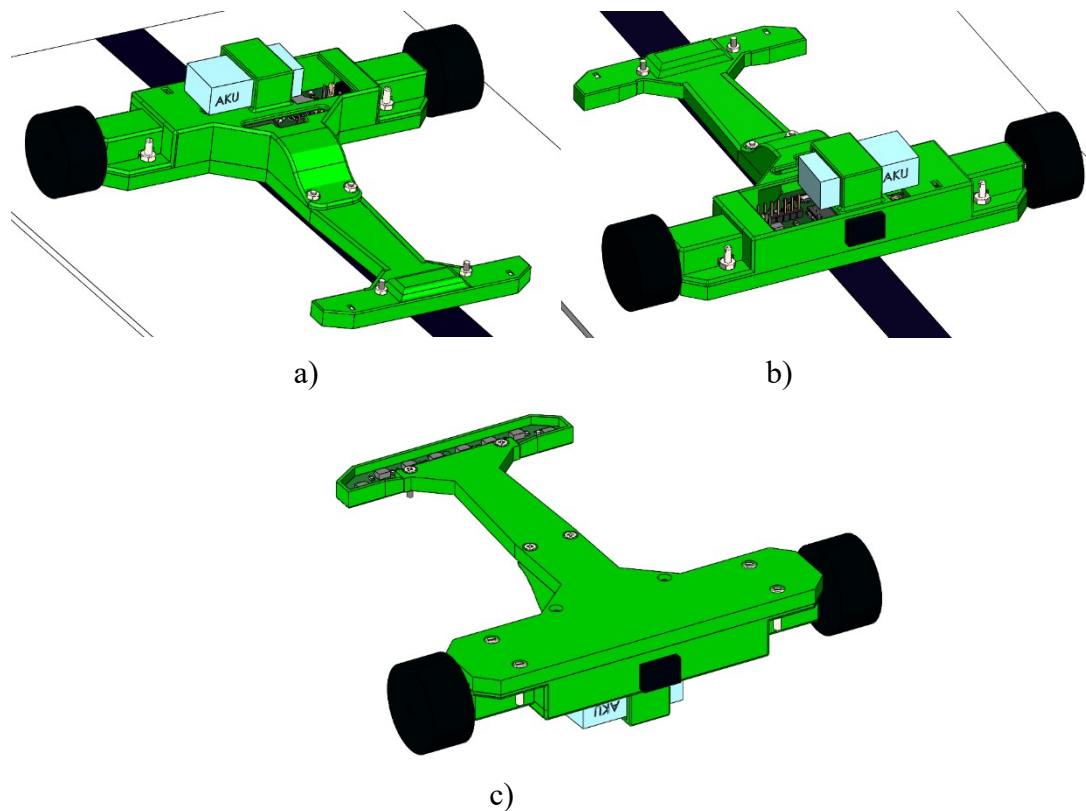
Podczas przygotowania aplikacji opracowano ramkę danych (Rys 3.15) umożliwiającą identyfikację polecenia oraz odbiór danych. Odbiór danych uznaje się za zakończony, jeśli przez co najmniej 5ms nie otrzymano żadnych danych.



Rys. 3.15 Struktura ramki danych

3.14. Projekt i realizacja obudowy robota

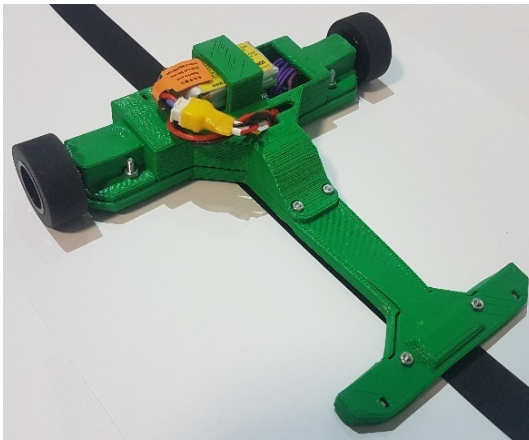
Zgodnie z założonymi wymaganiami projektowymi, w celu poprawy estetyki całej konstrukcji oraz zagwarantowania odporności mechanicznej, zaprojektowano dedykowaną obudowę robota. Według przygotowanego złożenia czujniki powinny znajdować się około 1mm nad powierzchnią, jest to optymalna odległość od obiektu pomiarowego według producenta. Prosta wymiana baterii oraz dostęp do karty SD, gniazd programatora powinien być także możliwy. Na Rys 3.16 została przedstawiona zaprojektowana obudowa wraz z ogólnym projektem mechaniki robota. Obudowa planowo zostanie wydrukowana na Drukarce 3D metoda przyrostową FDM, z wykorzystaniem filamentu PLA.



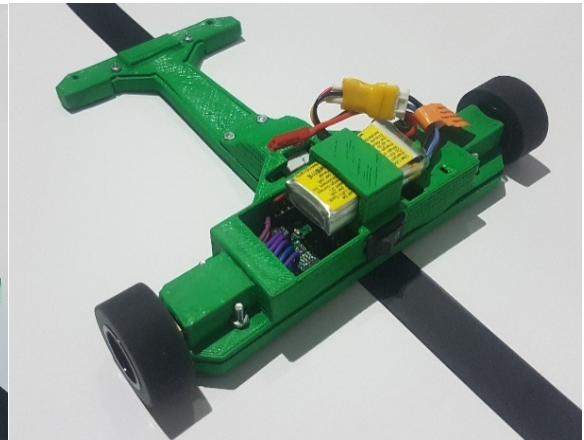
Rys 3.16 Model 3D konstrukcji robota widok a) przód b) tył c) dół

Na podstawie szczegółowo wykonanego projektu udało się wyeliminować ewentualne błędy konstrukcyjne. Na Rys 3.17 zostały przedstawione fotografie przygotowanego modelu robota. Obudowa poprawia estetykę całej konstrukcji oraz zapewnia wytrzymałość mechaniczną na ewentualne zderzenia. Wykonana obudowa umożliwia również prostą wymianę baterii, dostęp do karty SD, montaż modułu

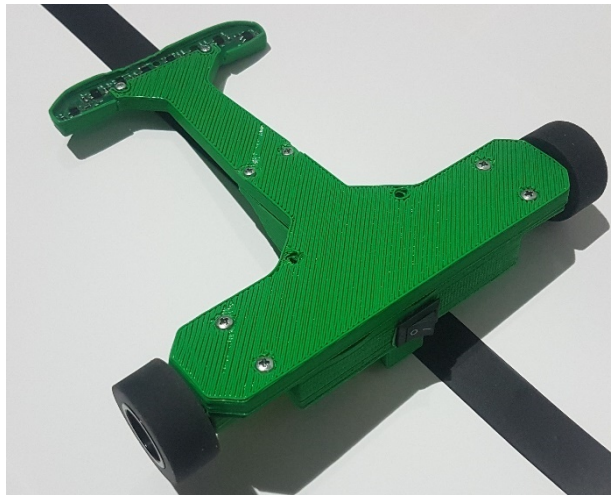
akcelerometru. Natomiast umieszczony w tylnej części przełącznik umożliwia proste i szybkie załączanie i wyłączanie zasilania



a)



b)



c)

Rys 3.17 Praktyczna realizacja przygotowanego modelu robota widok a) przodu b) tył c) dół

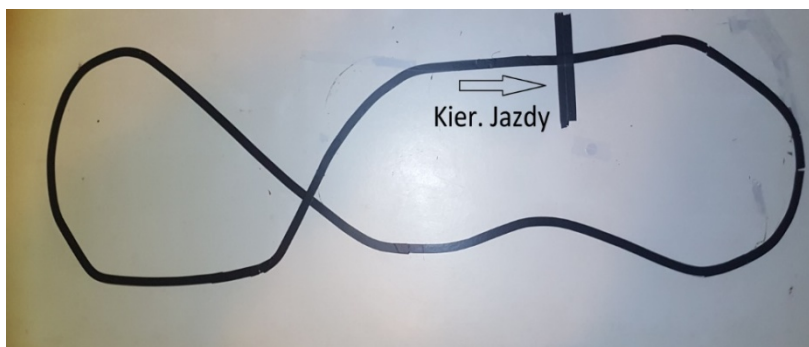
4. Testy i uruchomienie robota

Po zmontowaniu konstrukcji robota sprawdzono poprawność działania wszystkich zastosowanych elementów. Udało się uzyskać komunikację z kartą SD, modulem nawigacji inercyjnej, pamięcią EEPROM i modulem Bluetooth. Opracowany model robota został przygotowany do uruchomienia i testów.

4.1. Budowa tras testowych

Aby przetestować użyteczność pojazdu pod kątem startu w zawodach oraz w celu ewentualnego porównania możliwości przygotowanego pojazdu z możliwościami innych pojazdów tego typu, niezbędne jest przygotowanie odpowiednich tras oraz pomiar czasu przejazdu i średniej prędkości. Zgodnie z regulaminem zawodów na trasach mogą się pojawić łuki, kąty ostre oraz skrzyżowania na prostych odcinkach. W związku z tym należy umieścić takie elementy i sprawdzić jak pojazd sobie z nimi radzi. Trasa powinna być zbudowana z płyty MDF lub podobnego materiału w kolorze białym, natomiast do budowy toru trasy można wykorzystać czarną taśmę izolacyjną o szerokości 19mm wykonaną z PCV.

Do przygotowania testowych tras pozyskano dwustronnie białą płytę MDF o wymiarach 200x80mm, oraz taśmę izolacyjną. Fotografie tras ilustruje Rys 4.1.



a)



b)

Rys 4.1 Fotografie przygotowanych tras a) trasa A b) trasa B

Na trasie A przygotowano utrudnienia w postaci zakrętów oraz skrzyżowań, jej długość to około 4,1m. Natomiast na trasie B przygotowano utrudnienia w postaci kątów prostych, długość tej trasy to około 3,85m. Na obu trasach umieszczono czarny znacznik utworzony z podwójnej szerokości linii toru. Zastosowanie enkoderów pozwoli na jego wykrycie. Umożliwi to pomiar czasu okrążenia oraz średniej prędkości na trasie. Przy wykorzystaniu równania 3.15 możliwy będzie również przybliżony pomiar długości trasy za pomocą systemów odometrii robota.

4.2. Testy robota

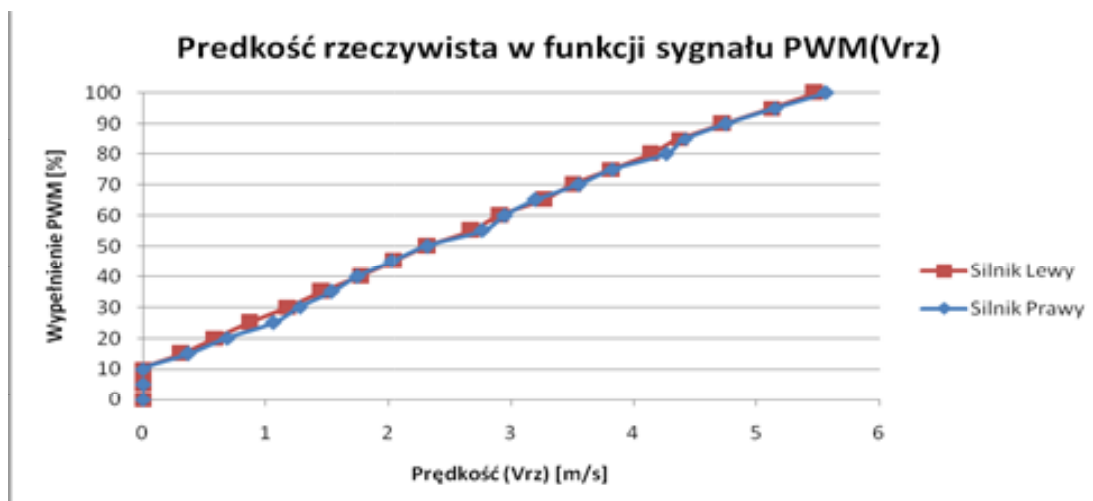
Przed rozpoczęciem testów konieczna była implementacja w mikrokontrolerze sterownika silników przedstawionego na Rys. 3.11. W celu poprawy jakości jego działania wykonano serię eksperymentów polegających na zadawaniu określonej wartości sygnału PWM. W momencie, gdy wartość prędkości się ustaliła, odczytywano wynik. Pozwoliło to utworzyć charakterystykę rzeczywistej prędkości poszczególnych kół w zależności od wypełnienia sygnału PWM (Rys 4.2). Ze względu na to, że otrzymane charakterystyki są bliskie linii prostej oraz są niemal takie same, postanowiono je aproksymować jedną linią prostą. W wyniku czego otrzymano przybliżoną funkcję sygnału PWM w zależności od rzeczywistej prędkości V_{rz} co można opisać równaniem:

$$PWM(V_{rz}) = 17,3 \cdot V_{rz} + 8,25 \quad (4.2)$$

Umożliwiło to przeliczanie wartości prędkości zadanej na wartość wypełnienia sygnału PWM. W następnym kroku zaimplementowano algorytm podążania za linią przedstawioną na Rys 3.12. Pozwoliło to rozpocząć testy, jak pojazd radzi sobie z przygotowanymi trasami. Pojazd bez problemu pokonuje skrzyżowania oraz kąty proste. Podczas przejazdów pojazd gubił trasę, jedynie w momentach gdy w skutek zbyt dużych szybkości tracił przyczepność i wypadał całkowicie poza nią. W tej sytuacji stawał się bezradny i wykonywał obroty wokół własnej osi. W ramach przejazdów po przygotowanych trasach, eksperymentalnie dobierano parametry regulatora PD (tabela 4.2), dzięki którym osiągnięto najlepsze rezultaty (tabela 4.3) przy których pojazd nie wypadał z trasy. Zadanie większych prędkości powodowało wypadanie poza trasę.

Tabela 4.2 Parametry najlepszego bez profilu prędkości przejazdu

Parametr	Trasa A	Trasa B
Prędkość zadana[m/s]	1,5	1,3
Wartość K_p [-]	0,2	0,23
Wartość K_d [-]	0,4	1,5
Czas próbkowania członu różniczkującego[ms]	30	30



Rys. 4.2 Rzeczywista prędkość przy danym wypełnieniu sygnału PWM

Tabela 4.3 Osiągnięte najlepsze rezultaty

-	Trasa A	Trasa B
Najkrótszy czas przejazdu[s]	3,07	3,68
Prędkość średnia środka osi pojazdu przy najkrótszym osiągniętym czasie[m/s]	1,43	1,28

W celu wyznaczenia chwilowej prędkości jazdy skorzystano z wyrażenia:

$$V(i) = \frac{d_{L,i} + d_{Pi,i}}{t_i} \quad (4.3)$$

Gdzie $d_{L,i} + d_{Pi,i}$ dystans pokonywany odpowiednio przez koło lewe i prawe w i-tej iteracji, t_i to czas próbkowania. Do wyznaczenia średniej prędkości na trasie skorzystano z wyrażenia:

$$V_{sr} = \frac{\sum_{i=1}^{L_{pr}} V(i)}{L_{pr}} \quad (4.4)$$

Gdzie: L_{pr} - całkowita liczba uzyskanych próbek.

Do optymalizacji przejazdu z wykorzystaniem profilu prędkości, zaimplementowano algorytm budowy mapy trasy zgodnie z wyrażeniami 3.12, 3.15. Pozwoliło to rozpocząć przejazd zbierający informacje o trasie. Podczas tego przejazdu ustawiono małą prędkość zadaną na poziomie 0.7m/s, aby wierniej odwzorować trasę. Na Rys 4.3 została przedstawiona uzyskana w ten sposób mapa tras (pojazd wolny). Przesłanie

zebranych danych na smartphon było możliwe dzięki przygotowanej aplikacji mobilnej wspomnianej w poprzednim 2. rozdziale. W następnym kroku przesłano je do komputera, co umożliwiło ich prostą analizę.

Przyglądając się uzyskanym danym dla trasy A (Rys 4.3a, pojazd wolny) określono, że pojazd w okolicy punktu (0,281;0,0482), powinien zwolnić przed zbliżającym się zakrętem. Za pomocą wyrażenia 3.15 obliczono, że przebyty dystans odpowiadający temu punktowi wynosi w przybliżeniu 0,3m. Dokonując takiej analizy przed poszczególnymi zakrętami oraz momentami wyjścia z nich, przygotowano profil prędkości. Dla poszczególnych tras przedstawiony został on na Rysunku 4.4. Uzyskane różnice czasu względem przejazdów z wykorzystaniem profilu prędkości i bez niego przedstawia tabela 4.4.

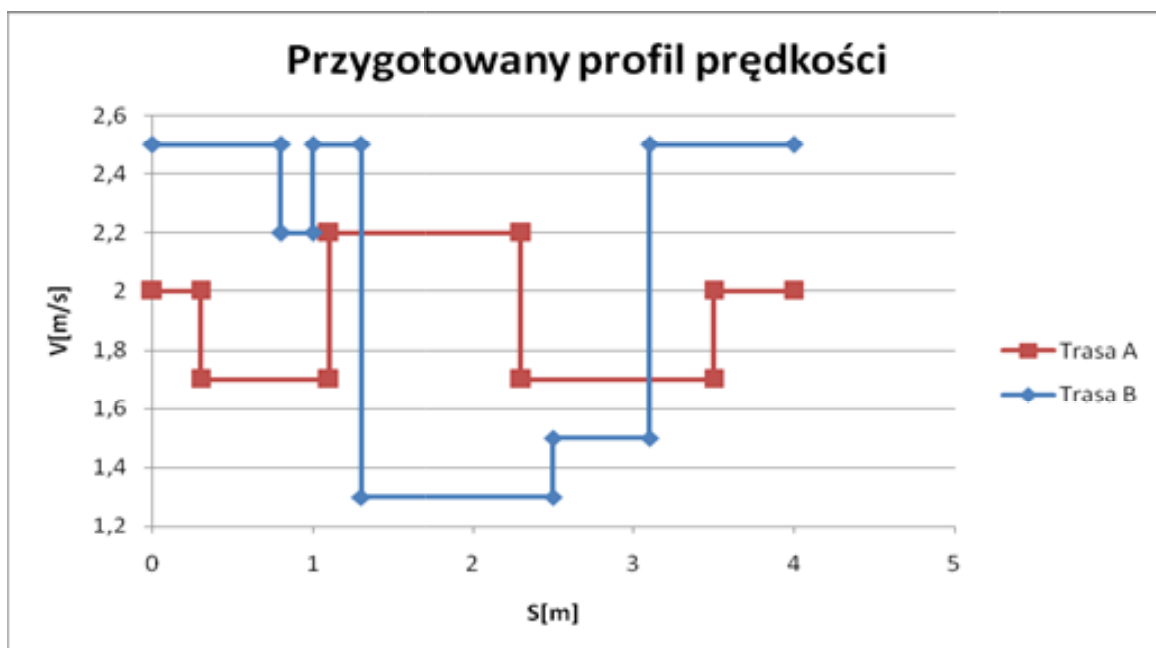


a)



b)

Rysunek 4.3 Ścieżka, po której poruszał się środek osi pojazdu uzyskana na podstawie systemów odometrii robota a) trasa A b) trasa B



Rys 4.4 Utworzony analitycznie profil prędkości

Tabela 4.4 Różnice czasu z wykorzystaniem profilu prędkości

Trasa	Najlepszy czas bez profilu prędkości[s]	Najlepszy czas z wykorzystaniem profilu prędkości[s]	Zmiana[s]
Trasa A	3,07	2,92	-0,15
Trasa B	3,68	3,33	-0,35

Jak zostało to przedstawione w tabeli 4.4, udało się nieco skrócić czasy przejazdów, dzięki zastosowanej metodzie. W przypadku trasy B metoda wykazała lepszy rezultat ze względu na występujące kąty ostre, przy których pojazd musiał poruszać się, ze znacząco mniejszą prędkością. Jak widać, na rysunku 4.3 podczas szybkiego przejazdu z wykorzystaniem profilu prędkości (pojazd szybki), odwzorowanie tras było bardzo nie dokładne, a uzyskana całkowita w ten sposób długość na obu trasach była większa o około 0,3m. Pozwala to wysnuć wniosek, że w przypadku dłuższych tras zastosowanie przedstawionej metody optymalizacji może być niemożliwe

5. Podsumowanie i wnioski końcowe

Podstawowym celem niniejszej pracy było opracowanie koncepcji i praktyczna realizacja modelu autonomicznego pojazdu śledzącego sterowanego cyfrowo. Cel ten został zrealizowany w całości.

Wykonano robota śledzącego, w którym zaimplementowano algorytm sterujący z optymalizacją przejazdów przy pomocy profilu prędkości zapewniający działanie zgodne z wstępnymi założeniami projektowymi. Do bezprzewodowej komunikacji z robotem został wykorzystany moduł Bluetooth oraz odbiornik podczerwieni, a mobilna aplikacja stworzona w graficznym środowisku AppInventor ułatwia sterowanie pojazdem. Wykonana przy pomocy drukarki 3D konstrukcja modelu pojazdu zapewnia jego stabilność i odpowiednią funkcjonalność.

W dalszej perspektywie dzięki temu, że w pojeździe przygotowano wyprowadzenia dla modułu nawigacji inercyjnej, jak również możliwy jest pomiar rzeczywistej prędkości kół, możliwe jest zastosowanie algorytmów korekcji położenia i optymalizacji ścieżki przejazdu. Powinno to umożliwić osiąganie lepszych rezultatów w zawodach robotów, nawet przy długich trasach. W pracach [5,10,11] przedstawiono takie rozwiązania oraz badania ich wpływu na czasy przejazdów.

Autor za wkład własny uważa wykonanie konstrukcji modelu autonomicznego pojazdu klasy LineFollower, zaprojektowanie obwodów drukowanych oraz obudowy, połączenie całości w jeden działający system. Za wkład własny autora można uznać także, napisanie programu sterującego mikrokontrolerem oraz utworzenie aplikacji mobilnej służącej do komunikacji z robotem.

Literatura

- [1] Shane https://elportal.pl/pdf/k03/08_04.pdf Dostęp 13.01.2021.
- [2] Ł. Chojnacki. Projekt optymalnego układu pomiarowego dla robota typu „micromouse”. Praca inżynierska. Politechnika Wrocławska 2015. https://www.researchgate.net/profile/Lukasz_Chojnacki2 Dostęp 13.01.2021
- [3] Marcin13021988. Roboty MicroMouse – 5 metod przeszukiwania labiryntu <https://forbot.pl/blog/roboty-micromouse-5-metod-przeszukiwania-labiryntu-id17354>.
- [4] O. Pacelt. Walki robotów - pasjonujące hobby elektronika <https://botland.com.pl/blog/walki-robotow-pasjonujace-hobby-dla-elektronika/> Dostęp 13.01.2021
- [5] W. Lipieta. Badania porównawcze algorytmów śledzenia trasy dla robotów klasy LineFollower. Praca Magisterska. Politechnika Wrocławska 2017. https://kcir.pwr.edu.pl/~much/Pracki/Witek_Lipieta_praca_magisterska.pdf Dostęp 13.01.2020.
- [6] Transoptor odbiciowy KTIR0711s. Dokumentacja techniczna. www.farnell.com/datasheets/2307824.pdf Dostęp 13.01.2021.
- [7] T. Cejrowski, M. Paczyński, D. Machala. Konstrukcja robota turniejowego Mirror startującego w zawodach Mini Sumo <https://yadda.icm.edu.pl/baztech/element/bwmeta1.element.baztech-d0615731-586c-4910-bcd4-fec63886f078/c/Cejrowski.pdf> Dostęp 13.01.2021.
- [8] Bezobsługowe nośniki pojazdów dla małych ładunków w przemyśle, Strona internetowa firmy SSI Schaefer. <https://www.ssi-schaefer.com/pl> Dostęp 13.01.2021.
- [9] Robotic Arena 2020 - Regulamin konkurencji LinefollowerLight https://www.roboticarena.pl/media/filer_public/1e/f4/1ef40f87-a142-45e0-b4a7-045850bb19f7/lf_light.pdf Dostęp 13.01.2021
- [10] J. Stolarz. Optymalizacja trajektorii przejazdu trasy robota klasy linefollower. Praca inżynierska. Politechnika Wrocławska 2015 https://kcir.pwr.edu.pl/~much/Pracki/Jedrzej_Stolarz_praca_inzynierska.pdf Dostęp 13.01.2021
- [11] R. Cymiński Optymalizacja toru ruchu robota klasy LineFollower. Praca Magisterska. Politechnika Wrocławska 2018 https://kcir.pwr.edu.pl/~much/Pracki/Rafal_Cyminski_praca_magisterska.pdf Dostęp 13.01.2021
- [12] Shane https://elportal.pl/pdf/k03/08_04.pdf Dostęp 13.01.2021.

- [13] Silnik Pololu HPCB 10:1 z przedłużonym wałem. Dokumentacja techniczna.
www.pololu.com/product/3071 Dostęp 13.01.2021.
- [14] Mostek H – DRV8833. Dokumentacja techniczna
<https://www.ti.com/lit/gpn/drv8833> Dostęp 13.01.2021.
- [15] Dokumentacja techniczna modułu Bluetooth HM-10
<http://esp32-server.de/wp-content/uploads/HM-10.pdf> Dostęp 13.01.2021
- [16] Baton. Algorytm linefollowera w C dla początkujących i nie tylko.
<https://forbot.pl/blog/algorytm-linefollowera-c-poczatkujacych-id2722> Dostęp 13.01.2021