

PROJEKT

ROBOTY MOBILNE

Finalny rezultat

Robot mobilny LineFollower
"Kablostwór"

Autor

Janusz Domaradzki, 249024

Termin: środa TN

Prowadzący: Arkadiusz Mielczarek

Wrocław, 2021

1 Wstęp

Niniejszy raport zawiera podsumowanie projektu na Roboty Mobilne, będącym robotem mobilnym klasy LineFollower. Zawarte tutaj zostały konstrukcja robota oraz konfiguracja mikrokontrolera w CubeMX wraz z wgranym programem. Pojawi się też informacja o napotkanych problemach w trakcie realizacji projektu i ich wpływu na projekt jako całość.

2 Opis robota

Robot za swój wygląd oraz ilość użytych do niego kabli został nazwany "Kablostwór".

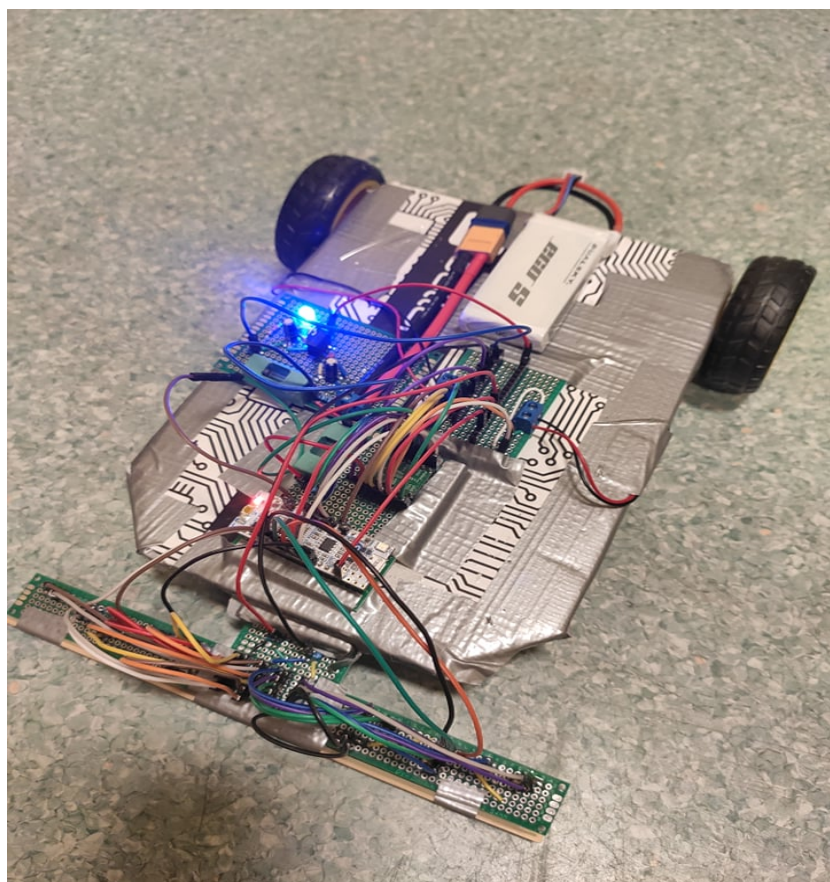
2.1 Konstrukcja robota

Konstrukcja robota została zroiona z trzech warstw tektury połączonych srebrną taśmą.

Na robocie zostały umieszczone płytki uniwersalne z polutowanymi na nich układach elektronicznych. Na jednej z płytek znajduje się układ zasilający wraz z gniazdem na baterię. Na drugiej gniazda dla obu silników oraz gniazda na kable łączące zasilania oraz uziemnienia (oddzielnie od siebie). Na trzeciej znajduje się gniazdo na mikrokontroler oraz sterownik silników. Robot też posiada z przodu połączone ze sobą płytki stykowe zawierające transoptory odbiciowe, mające za zadanie wykrywać linię.

Wszystkie płytki stykowe połączone są z konstrukcją robota srebrną taśmą. To samo dotyczy jego silników, o czym zostanie jeszcze wspomniane w innej sekcji.

Sam "Kablostwór" prezentuje się następująco:

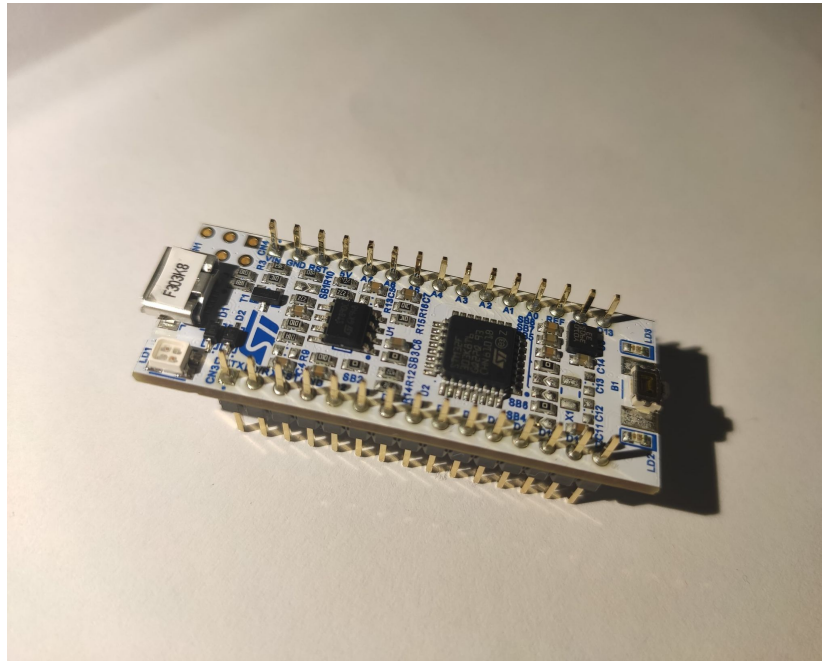


Rysunek 1: Kablostwór z podłączonym zasilaniem

2.2 Użyte komponenty

- Mikrokontroler

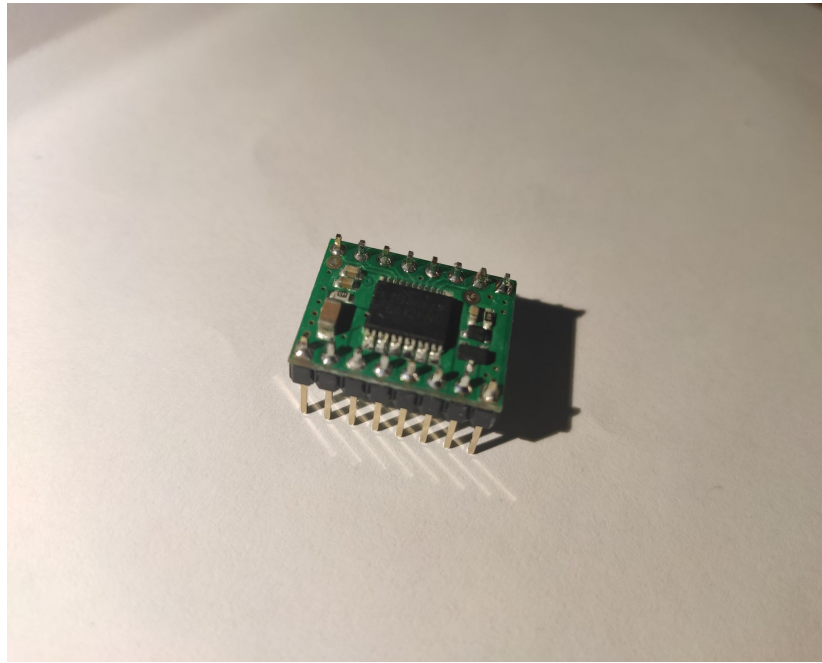
Wykorzystany mikrokontroler to STM32F303K8 z 32-bitowym procesorem Arm Cortex M4.



Rysunek 2: Mikrokontroler STM32F303K8

- Sterownik silników

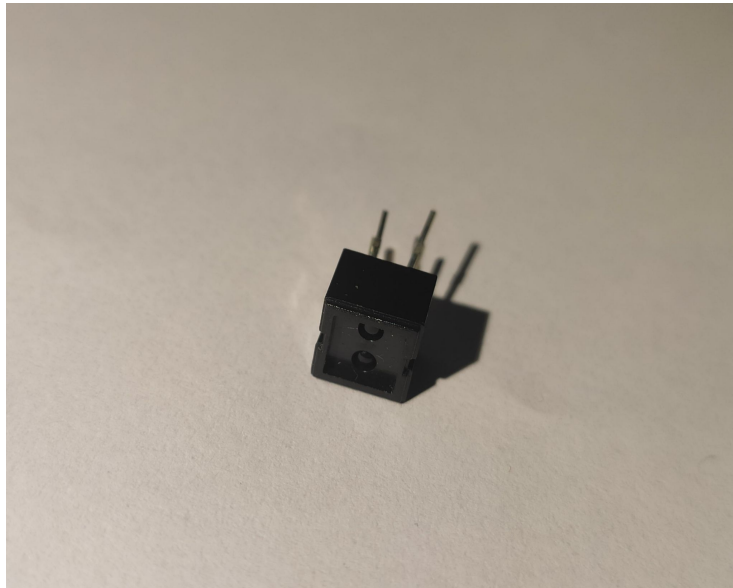
Silniki są obsługiwane przy pomocy dwukanałowego sterownika silników Pololu 713, zawierającego w sobie mostek H TB6612FNG. Zostają mu przekazane sygnały PWM z odpowiednim wypełnieniem, nadające silnikom prędkość.



Rysunek 3: Sterownik silników TB6612FNG

- Czujniki

Do wykrycia linii zostały wykorzystane transoptory odbiciowe CNY70. Składają się one z nadajnika i odbiornika. Nadajnik wysyła wiązkę światła, zaś odbiornik mierzy natężenie światła odbitego od powierzchni. Sygnał wyjściowy jest zależny od zmierzonego natężenia. Im mniej światła dotrze ponownie, tym większy sygnał zostanie nadany, a więc gdy zostanie wykryta linia, tym większa wartość jest odczytywana przez znajdujący się na mikrokontrolerze przetwornik ADC.



Rysunek 4: Czujnik CNY70

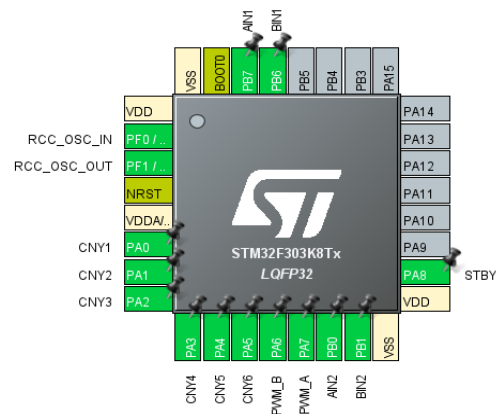
- Bateria zasilająca
Do zasilania wykorzystana została bateria Li-Pol Dualsky o pojemności 1300 mAh.



Rysunek 5: Bateria Li-Pol Dualsky

2.3 Konfiguracja mikrokontrolera w CubeMX

Na poniższym zdjęciu oraz tabelach została zaprezentowana konfiguracja peryferii mikrokontrolera



Rysunek 6: Konfiguracja peryferii mikrokontrolera w CubeMX

2.3.1 ADC1

ADC_Settings	Wartość
Clock Prescaler	Synchronous clock mode divided by 1
Resolution	8 bits
Data Alignment	Right alignment
Scan Conversion Mode	Enabled
Continuous Conversion Mode	Disabled
Discontinuous Conversion Mode	Disabled
DMA Continuous Requests Mode	Disabled
End Of Conversion Selection	End of single conversion
ADC_Regular_ConversionMode	Wartość
Number Of Conversion	4
External Trigger Conversion Source	Regular Conversion launched by software
External Trigger Conversion Edge	None
Rank	1
Channel	Channel 1
Sampling Time	4.5 Cycles
Rank	2
Channel	Channel 2
Sampling Time	4.5 Cycles
Rank	3
Channel	Channel 3
Sampling Time	4.5 Cycles
Rank	4
Channel	Channel 4
Sampling Time	4.5 Cycles
ADC_Injected_ConversionMode	
Number Of Conversions	0
WatchDog	
Enable Analog WatchDog Mode	false

Tabela 1: Konfiguracja peryferium ADC1

2.3.2 ADC2

ADC_Settings	Wartość
Clock Prescaler	Synchronous clock mode divided by 1
Resolution	8 bits
Data Alignment	Right alignment
Scan Conversion Mode	Enabled
Continuous Conversion Mode	Disabled
Discontinuous Conversion Mode	Disabled
DMA Continuous Requests Mode	Disabled
End Of Conversion Selection	End of single conversion
ADC_Regular_ConversionMode	Wartość
Number Of Conversion	2
External Trigger Conversion Source	Regular Conversion launched by software
External Trigger Conversion Edge	None
Rank	1
Channel	Channel 1
Sampling Time	4.5 Cycles
Rank	2
Channel	Channel 2
Sampling Time	4.5 Cycles
ADC_Injected_ConversionMode	
Number Of Conversions	0
WatchDog	
Enable Analog WatchDog Mode	false

Tabela 2: Konfiguracja peryferium ADC2

2.3.3 TIM3

TIM_Settings	Wartość
Counter Settings	
Prescaler	1599
Counter Mode	Up
Counter Period (ARR)	99
Internal Clock Division (CKD)	No Division
auto-reload preload	Disable
PWM Generation Channel 1 Mode	PWM mode 1
Pulse (16 bits value)	0
Output compare preload	Enable
Fast mode	Disable
CH Polarity	High
PWM Generation Channel 2 Mode	PWM mode 1
Pulse (16 bits value)	0
Output compare preload	Enable
Fast mode	Disable
CH Polarity	High

Tabela 3: Konfiguracja Timera generującego sygnał PWM dla obu silników

2.4 Zaimplementowany program

Do sterowania robotem podjęta została próba implementacji regulatora PID. Przed jego działaniem zainicjowane zostały wszystkie zmienne potrzebne do obliczeń. Skonfigurowane zostały również PINy odpowiadające za kierunek obrotu silników – oba silniki mają za zadanie obracać się do przodu. Zainicjowane zostało również DMA dla obu przetworników ADC.

Robot napierw pobiera dane z czujników. Jeśli odczytana wartość przekroczy wyznaczony próg wykrycia linii, to zapamiętuje ten czujnik i wartość jego wagi dodaje do sumy kontrolnej oraz inkrementuje ilość sensorów, które wykryły linię. Potem dzieli sumę kontrolną przez ilość sensorów, które wykryły linię. Następnie oblicza różniczkę na podstawie sumy kontrolnej oraz jej poprzedniej wartości (która na początku ma wartość 0). Na podstawie różniczki oraz sumy kontrolnej oblicza się wartość regulatora PID, która to po obliczeniu, wraz z wartością startową, jest wpisywana jako wypełnienie sygnału PWM sterującego silnikami. Na koniec ponownie zostają pobierane wartości z sensorów poprzez DMA dla obu przetworników ADC.

3 Problemy

W trakcie realizacji projektu wystąpiły pewne problemy, które wymagały rozwiązania. Niestety ale niektóre z nich znacząco wpłynęły na działanie robota i nie wszystkie z nich udało się rozwiązać na czas.

3.1 Rozwiązane

- Początkowo po wgraniu programu Robot nie był w stanie jednocześnie sterować silnikami oraz odczytywać danych z sensorów, pomimo ustawienia trybu DMA w ADC na Circular. Aby to rozwiązać, należało wyłączyć cykliczne pobieranie informacji przez DMA w konfiguracji, a zamiast tego wykonywać je programowo.
- Zdarzało się, że po zmianie orientacji robota, tzn. jego obrocie zasilanie zwierało się z masą. Okazało się, że układ zasilający nie został poprawnie zlutowany. Zby to rozwiązać, należało poprawić lutowanie po uprzedniej dekonstrukcji robota.
- Jeden z przewodów silnika nie trzymał się poprawnie w gnieździe. Aby to naprawić pogrubiono końcówkę przewodu za pomocą cyny.

3.2 Nerozwiazane

- Problem z konstrukcją mechaniczną – W trakcie testów okazało się, że łączenie ze sobą elementów konstrukcji robota srebrną taśmą nie jest najlepszym rozwiązaniem. W szczególności zostało to odczute na silnikach, które przez nacisk robota na jego tylną oś, zostają przekrzywione. Może to być jedna z przyczyn opisanego poniżej ostatniego z problemów.
- Problem z PIDem – Niestety ale robot sterowany regulatorem PID był bardziej skłonny uciekać z trasy niż na niej pozostać. Jeździł płynnie, ale nie spełniał podstawowego założenia, a mianowicie podążania po trasie. Problem nie został rozwiązany, jedynie algorytm został zastąpiony innym programem, który w większości przypadków powoduje poprawne działanie robota.
- Zacina się robot – Nie do końca wiadomo, czy wynika to bardziej z problemu z konstrukcją, czy z programem zastępującym PIDa, czy też z układem elektronicznym. Nie został on rozwiązany przez brak wystarczającej ilości czasu.

4 Podsumowanie

Główną ideą w założeniach było zbudowanie czegoś działającego. Można teoretycznie uznać, iż została ona spełniona, gdyż robot sam w sobie się porusza, więc jest robotem mobilnym, czyli działającym. Nie porusza się on niestety

płynnie. Pomimo to, jest w stanie przebyć wyznaczony dla niego tor. Czasami z niego ucieka, ale to jest zdecydowaną rzadkością.

Występowanie nierozwiązanych problemów wynikało głównie ze źle podejmowanych decyzji na początku konstruowania robota (wspomniane wcześniej mocowania na srebrną taśmę). Spowodowały one pogorszenie działania robota w finalnym rezultacie.

Można wyciągnąć sporo wniosków z tego projektu, dotyczących każdego aspektu pracy nad nim, poczynając od konstrukcji robota, poprzez elektronikę, a kończąc na jego oprogramowaniu.

- Konstrukcja robota powinna być najbardziej sztywna, jak to możliwe.
- Układ elektroniczny powinien być jak najdokładniej polutowany.
- Lepszym rozwiązaniem powinno być zaprojektowanie płytki PCB, niż lutowanie elementów na płytce uniwersalnej (Na płytce uniwersalnej ryzyko błędu w lutowaniu jest bardzo wysokie).

Te oraz inne wnioski pomogą w pracy przy przyszłych projektach.