

POLITECHNIKA POZNAŃSKA

WYDZIAŁ AUTOMATYKI, ROBOTYKI I ELEKTROTECHNIKI

INSTYTUT ROBOTYKI I INTELIGENCJI MASZYNOWEJ

ZAKŁAD STEROWANIA I ELEKTRONIKI PRZEMYSŁOWEJ



PROGRAMOWALNE UKŁADY LICZNIKOWE

SYSTEMY MIKROPROCESOROWE

RAPORT LABORATORYJNY

KACPER KRASIŃSKI, 151234

KACPER.KRASINSKI@STUDENT.PUT.POZNAN.PL

BARTOSZ KUREK, 151188

BARTOSZ.KUREK@STUDENT.PUT.POZNAN.PL

PROWADZĄCY:

MGR INŻ. ADRIAN WÓJCIK

ADRIAN.WOJCIK@PUT.POZNAN.PL

7 STYCZNIA 2024



Spis treści

1	Opis Projektu	3
	Opis Projektu	3
1.1	Github	3
1.2	Doxygen	3
1.3	Aplikacja Desktopowa	3
1.4	MATLAB - symulacje/ skrypty	3
1.5	Wyswietlacz LCD	3
1.6	Dodatkowe wejście potencjometr/enkoder	3
1.7	System plików na karcie SD	3
1.8	Komunikacja sieciowa poprzez gniazdo ETHERNET	4
1.9	System sum kontrolnych	4
2	Schemat blokowy połączeń	4

OPIS PROJEKTU

Obiektem sterowania - elementem wykonawczym układu regulacji będzie ramię działające jako wskaźnik kąta obrotu. Sterowanie tego ramienia będzie zaimplementowane poprzez sterownik TB6560 który sygnał będzie wysyłał do silnika Nema 17. Urządzenie pomiarowe to w naszym przypadku IMU (ang. Inertial measurement unit) dzięki któremu uzyskamy informację o aktualnym położeniu końca wahadła, jego prędkości i przyspieszeniu. Najważniejsze dane będą widoczne na wyświetlaczu LCD. Dane dokładniejsze, w tym parametry regulatora i wskaźniki jakościowe będą dostępne w aplikacji desktopowej z którą będzie istniała możliwość strojenia regulatora i obserwacji przebiegów. Tryb manualny - sterowanie impulsatorem jak również wysyłanie danych wejściowych z komputera będą dostępne. Karta SD będzie zapewniała logi - zbiór najważniejszych danych z układu.

1.1 GITHUB

Projekt w całości będzie realizowany w dwie osoby, a więc użyjemy systemu kontroli wersji Github do przechowywania i zapisywania postępów. Projekt znajduje się pod poniższym linkiem. <https://github.com/Krasa35/inverted-pendulum> Dokumentacja będzie również prowadzona w miarę możliwości jednak z powodu ograniczeń wersji darmowej w pliku README udostępniony zostanie osobny link do projektu w Overleaf.

1.2 DOXYGEN

System komentarzy i struktury plików będzie bazował na standardzie Doxygen tak aby w przyszłości istniała możliwość wygenerowania pełnej dokumentacji do kodu w tym systemie.

1.3 APLIKACJA DESKTOPOWA

Aplikacja Desktopowa zostanie wykonana w środowisku programistycznym i będzie pozwalała na zintegrowanie danych otrzymanych z symulacji MATLAB, pomiarów z jednostki sterującej STM32 oraz wszystkich innych wejść do mikrokontrolera. Będzie ona pokazywała aktualny stan programu i pozwalała na zmianę wszystkich sterowalnych parametrów.

1.4 MATLAB - SYMULACJE/ SKRYPTY

Jeżeli uda nam się wykonać w pełni działającą aplikację desktopową sygnały sterujące będą generowane z zadanymi parametrami przez skrypt/symulację w środowisku Simulink. Natomiast Sygnały generowane będą zapisywane i przekazywane do głównego modułu sterowania - STM32 który będzie wysyłał sygnały sterujące.

1.5 WYŚWIETLACZ LCD

Na wyświetlaczu LCD będzie podgląd aktualnego stanu ramienia, oraz najważniejsze parametry sterowania. Tak aby system był w pełni sterowalny bez podłączonego komputera.

1.6 DODATKOWE WEJŚCIE POTENCJOMETR/ENKODER

Dodatkowe wejście dzięki potencjometrowi i przyciskowi będzie pozwalało nam zmienić pozycję ramienia za pomocą zmiany pozycji potencjometru i wciśnięcia przycisku.

1.7 SYSTEM PLIKÓW NA KARCIE SD

Na karcie SD będą zapisywane logi z działania programu, jego wartości parametrów a także wskaźniki jakościowe obiektu sterowania.

1.8 KOMUNIKACJA SIECIOWA POPRZECZ GNIAZDO ETHERNET

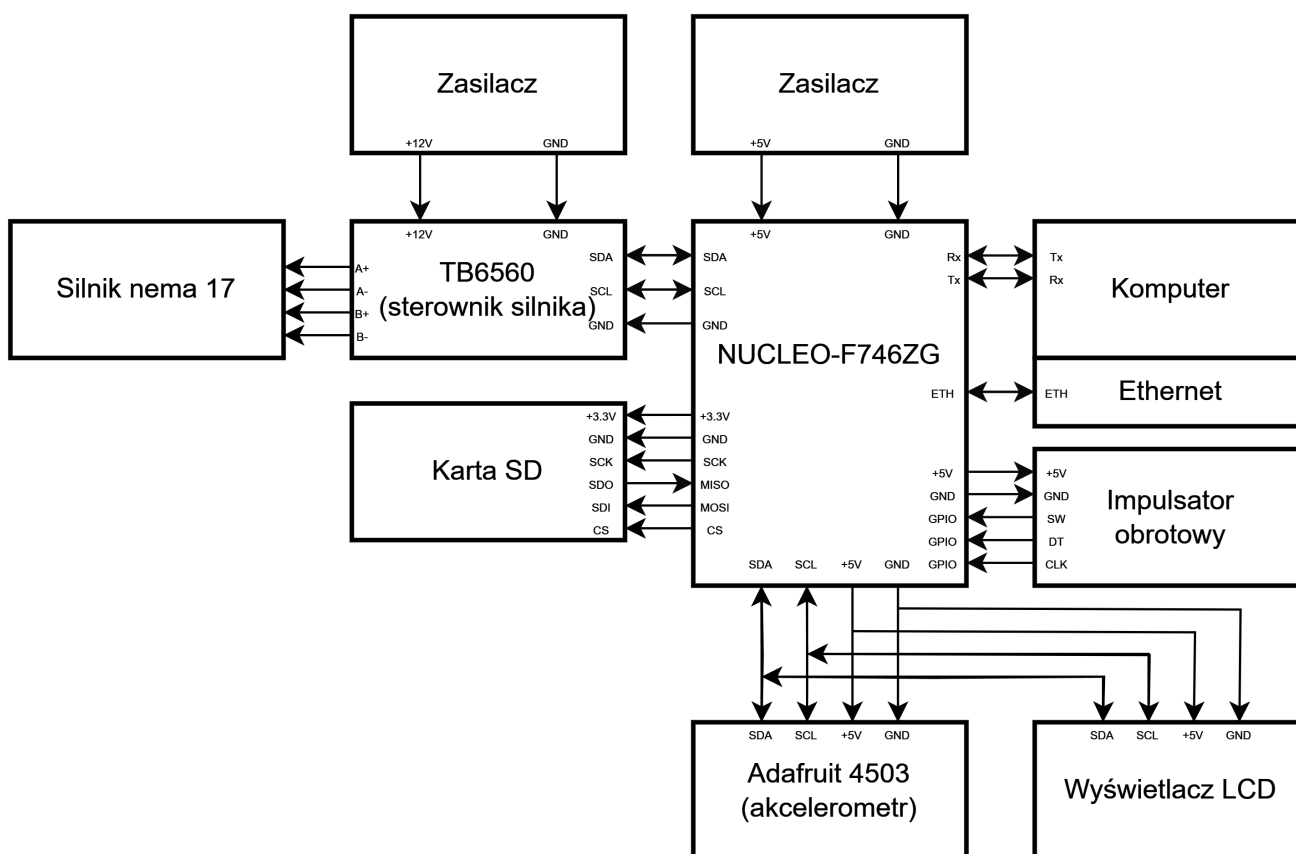
STM32 będzie komunikowało się z urządzeniem sterującym poprzez interfejs sieciowy. Protokół zostanie sprecyzowany podczas tworzenia projektu.

1.9 SYSTEM SUM KONTROLNYCH

Za pomocą systemu sum kontrolnych, najprawdopodobniej CRC, będziemy weryfikować łączność modułu sterującego (komputera) z jednostką sterującą. Tak aby zminimalizować problemy łączności i ewentualne przerwy nie wpłynęły na pracę obiektu.

SCHEMAT BLOKOWY POŁĄCZEŃ

Poniżej przedstawiono schemat blokowy połączeń niezbędnych peryferii wraz z wyszczególnionymi portami komunikacyjnymi które będziemy używać w zadaniu zaliczeniowym.



Rys. 1. Schemat blokowy połączeń obiektu sterowania