

Karta projektu zaliczeniowego

Systemy mikroprocesorowe - 2019

Temat projektu: **Robot samostabilizujący (segway)**

Imię i nazwisko:**Szymon Kwiatkowski**

Politechnika Poznańska

kierunek:**AiR**, grupa:**A2**, nr albumu: **132090**

1. Opis projektu

Tematem projektu jest robot samostabilizujący dwukołowy. Działa on na zasadzie odwrotnego wahadła. Utrzymuje pionową pozycję dzięki wykorzystaniu układu z akcelerometrem oraz żyroskopem. Wspomaga go cyfrowy sterownik PID zrealizowany na mikrokontrolerze STM32f103RC8T6. Wykorzystałem dany mikrokontroler aby poszerzyć moją wiedzę na temat programowania mikrokontrolerów innych niż atmega.



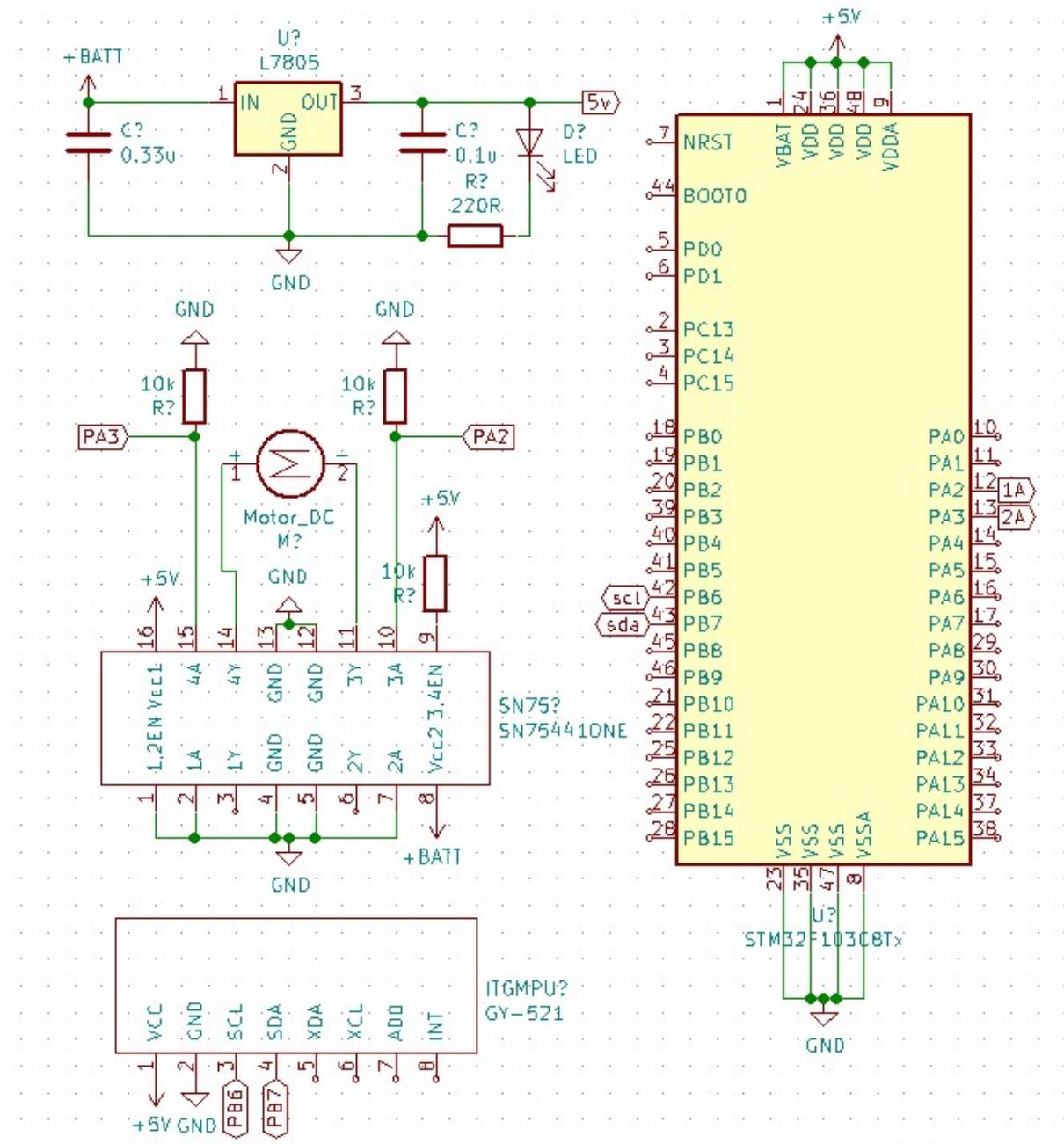
Rysunek 1: Prezentacja projektu

2. Budowa układu

Spis elementów elektronicznych:

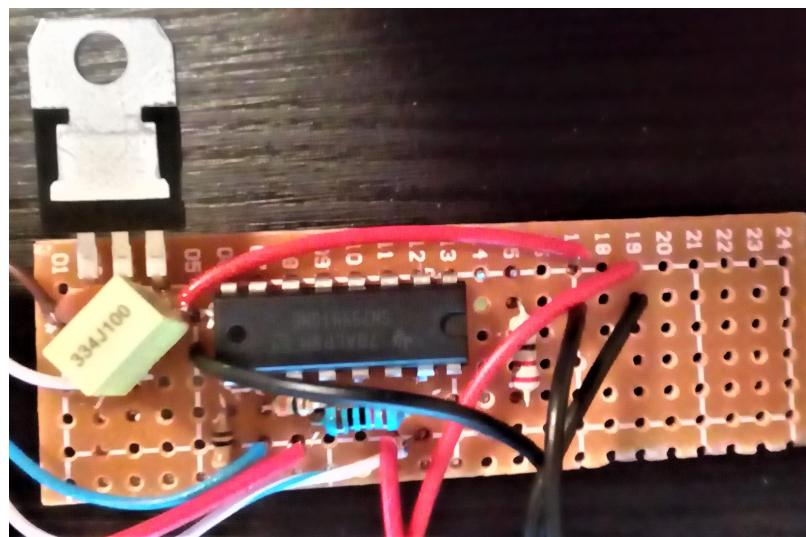
- Mikrokontroler STM32f103RC8T6 (bluepill)
- Moduł MPU-6050
- Sterownik silnika SN75441ONE
- Silnik DC znaleziony z odzysku
- Stabilizator napięcia 5V L7805CV
- Elementy pasywne

Schemat połączeń układu:



Rysunek 2: Schemat elektryczny układu

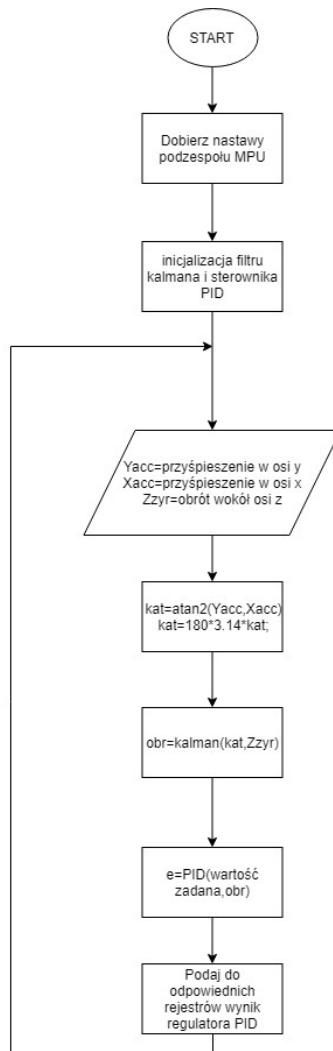
Połączenia wykonane na płytce prototypowej:



Rysunek 3: Wykonanie płytki

3. Elementy oprogramowania

Uproszczony schemat blokowy oprogramowania:



Rysunek 4: Uproszczony schemat blokowy programu

W projekcie zostały wykorzystane biblioteki z prostym regulatorem PID i filtrem Kalmana. Reszta oprogramowania została zrobiona przy użyciu biblioteki HAL. Biblioteka z regulatorem PID jest złożona z dwóch prostych funkcji: inicjalizacji regulatora PID, oraz obliczenie kolejnej próbki uchybu[5]. Biblioteka z filtrem Kalmana również zawiera funkcję inicjalizującą filtr, oraz aktualizującą odpowiedź[7]. Zostały one lekko przerobione względem oryginałów aby mogły działać w programie pisany w języku C.

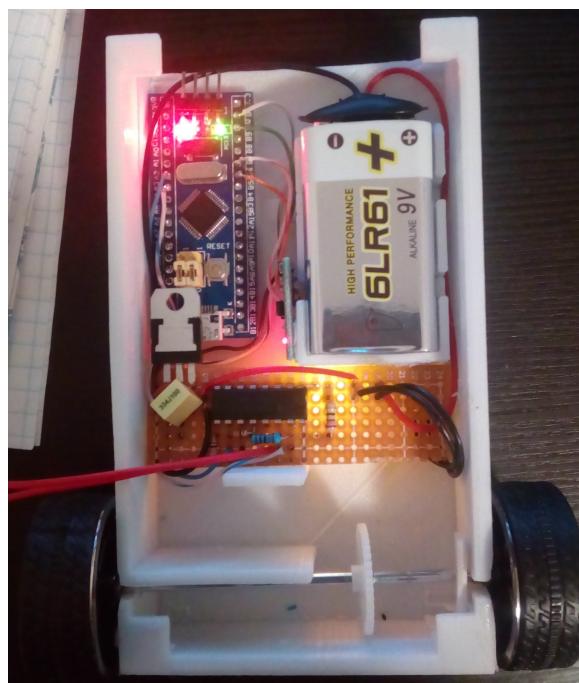
4. Wykorzystane narzędzia projektowe

W projekcie zostało użyte środowisko atollctrueSTUDIO, jest to środowisko oparte na Eclipse. Praktycznie nie ma możliwości znalezienia darmowego środowiska nie opartego na Eclipse. Mikrokontroler nie był programowany w ArduinoIDE, ponieważ dzięki zostałem zapoznany z nowym sprzętem i nowymi środowiskami. Do projektowania obudowy posłużył program SolidWorks, a schematu elektrycznego środowisko KiCad. Do załadowania programu posłużył STLink v2. Jest to dobry i wygodny w obsłudze programator do mikrokontrolerów STM. Do podglądu zmiennych w czasie rzeczywistym został użyty program STM Studio. Pomogło to np. wyznaczyć czas oscylacji robota.

5. Weryfikacja poprawności działania układu

Układ był testowany w trzech różnych obudowach. W pierwszej nie przewidziałem niskiego momentu silnika i nie zastosowałem żadnej przekładni oraz nie optymalizowałem odległości masy od wałów. Robot nie miał wystarczającej mocy aby z pozycji poziomej (trzymany w rękach) obrócić się do pionu. Druga została zrobiona w celach testu silnika z zastosowanymi przekładniami. Robot miał wystarczający moment, ale był źle zbalansowany i brakowało miejsca na podzespoły. Ostateczna obudowa została wydrukowana tak, aby zmieściły się 2 baterie 9V – daje to dłuższy czas działania robota i rozkładu wagę równomiernie. Układy elektroniczne zostały umieszczone od góry na lekko dłuższych kablach aby były dobrze widoczne i dawały możliwość wymiany baterii. Układ został przetestowany używając regulatora P, PI i PID. Najlepsze wyniki zostały osiągnięte dla regulatora PID. Układ wracał do pionu mimo nawet mocnych zakłóceń wywołanych pchnięciem od boku.

Projekt w pierwszej obudowie:



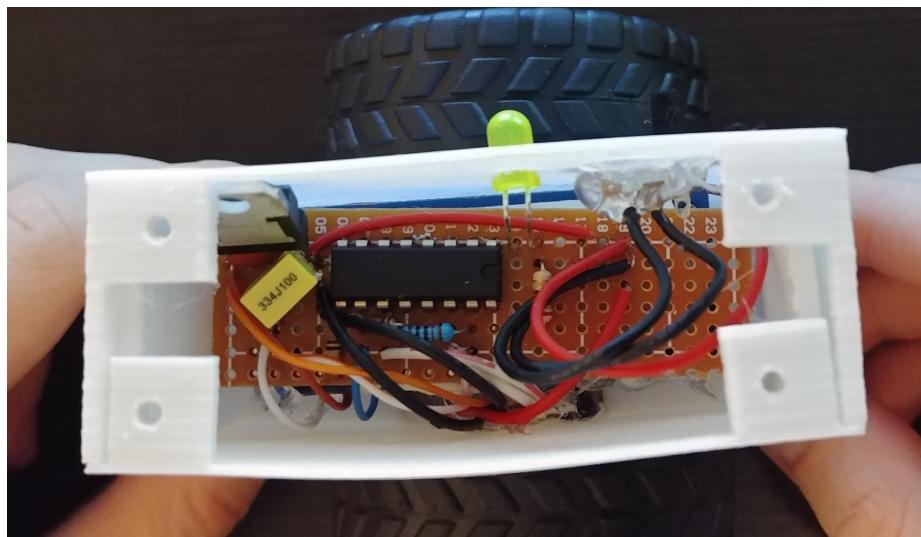
Rysunek 5: Obudowa 1

Obudowa nr 2:



Rysunek 6: Obudowa 2

Projekt w trzeciej obudowie:



Rysunek 7: Obudowa 3

Film z testami oprogramowania:

<https://www.youtube.com/watch?v=QrD4yn4eAcc&feature=youtu.be>

6. Obsługa układu

Aby uruchomić układ należy przełączyć przycisk zasilania i przytrzymać robota na ziemi w pozycji pionowej. Po około sekundzie robot sam zacznie utrzymywać równowagę. Żółta dioda świeci wraz z załączaniem robota i informuje nas o tym, że do robota podpięte jest napięcie.

Uruchomienie:



Rysunek 8: Włączenie układu

Wymiana baterii:

1. Odkręcić śruby:



Rysunek 9: Wymiana baterii 1

2. Zdjąć koło, następnie wysunąć pokrywę i wymienić baterie:



Rysunek 10: Wymiana baterii 2

7. Literatura

1. <https://forbot.pl/blog/stm32-praktyce-1-platforma-srodowisko-id2733>
2. <https://forbot.pl/blog/kurs-stm32-f1-migracja-na-hal-wstep-spis-tresci-id23580>
3. <https://forbot.pl/blog/kurs-stm32-f4-1-czas-poznac-hal-spis-tresci-kursu-id14114>
4. <https://forbot.pl/blog/filtr-kalmana-teorii-praktyki-1-id2855>
5. https://en.wikipedia.org/wiki/PID_controller#Pseudocode
6. <http://www.jarzebski.pl/arduino/rozwiazania-i-algorytmy/odczyty-pitch-roll-oraz-filtr-kalmana.html>
7. <https://github.com/jarzebski/Arduino-KalmanFilter>
8. <http://playground.arduino.cc/Main/MPU-6050>
9. https://en.wikipedia.org/wiki/Ziegler%E2%80%93Nichols_method