

Projek kontrolera lotu

Krzysztof Jamroz

Spis treści

1	Wprowadzenie	5
1.1	Wstęp	5
1.2	Przeznaczenie	5
1.3	Założenia konstrukcyjne	6
2	Realizacja sprzętowa - hardware	9
2.1	Wymagania sprzętowe	9
2.2	Mikrokontroler	9
2.3	Akcelerometr	9
3	Realizacja programowa - software	11
3.1	Narzędzia programistyczne	11
A	Kod źródłowy	17

Rozdział 1

Wprowadzenie

1.1 Wstęp

Celem projektu jest wykonanie działającego modelu kontrolera lotu spełniającego wymagania (rozdział ??), wykonującego zadania niezbędne do prowadzenia skutecznego wypracowywania komend oraz procedur związanych ze wznoszeniem i utrzymaniem maszyn latających w powietrzu. Projekt ten będzie złożony z 3 podstawowych modułów: teoria lotu i sterowania, rozwiązanie elektroniczne (z ang. *hardware*) oraz rozwiązanie programowe (z ang. *software*). Wykorzystane komponenty oraz podzespoły zostaną dostatecznie opisane tak aby niniejszy dokument był istotnym źródłem niezbędnej wiedzy oraz instrukcji (samouczka) w próbach odtworzenia procesu lub skonstruowania podobnej konstrukcji na bazie własnych wymagań. Jednocześnie uprasza się o ostrożność w podejmowaniu wszelkich decyzji oraz działań w procesie twórczym oraz nie traktowania tego skryptu jako wyroczni (autor nie ponosi odpowiedzialności za odniesione szkody na zdrowiu oraz straty materialne). Wskazówki określone w tej pracy wynikają z doświadczeń prowadzenia podobnego projektu na przestrzeni lat. Wszelkie kroki podjęte w dokumencie są konieczne w celu stworzenia oraz kontynuowania projektu na tak szeroką skalę z jednoczesnym rutynowym uczeniem się na bieżąco wszelkich technik lub umiejętności koniecznych (a nie posiadanych na etapie pracy). Dodatkowo wszystkie niezbędne materiały będą umieszczone we spisach na końcu pracy. Autor pracy zachęca do wnikliwego podjęcia wyzwania i życzy sukcesów na polu realizowania inżynierskich projektów.

1.2 Przeznaczenie

Głównym zadaniem projektu jest wykonanie działającego modelu kontrolera lotu, którego przeznaczeniem będzie obsługa podespołów pokładowych bezzałogowych statków powietrznych. W tym celu należy przypomnieć wcześniej wykonany projekt o nazwie POLAR (Polowy Obserwacyjno Latający Aparat Rozpoznawczy) zaprezentowany na konkursie III edycji konkursu bezzałogowych statków powietrznych Ministerstwa Obrony Narodowej. Projekt ten polegał na fuzji pocisku mózdzierzowego z klasycznym układem quadrokoptera. Zadanie takiej platformy polegało na wykonaniu rozpoznania przy użyciu mózdzierzy LM60 w ramach szybkiego i skrytego rozpoznania i zasilenia systemu dowodzenia informacjami z pierwszej ręki prosto z pola walki. Konstrukcja skorupy zmuszona była do wytrzymania ciśnienia panującego w lufie mózdzierza oraz przeciżenia

wynikające z nagłego skoku przyspieszeń w wyniku wybuchu ładunku miotającego. Poza trwałością części mechanicznych należało pamiętać o trwałości komponentów elektronicznych oraz o ich sztywności umieszczenia (potencjalnych dyslokacji czy przegrzewania). Rozmieszczenie wszystkich niezbędnych elementów na niewielkiej przestrzeni (pojemność pocisku) oraz stabilność prowadzenia lotu (obserwacji). Jedną z najważniejszych decyzji było wybranie platformy odpowiadającej za prowadzenie obliczeń i wypracowywania sygnałów do sterowania i kontroli jednostki latającej.

1.3 Założenia konstrukcyjne

Po określeniu celu oraz przeznaczenia projektu należy określić solidne ramy założeń co do konstrukcji kontrolera. W pierwszej kolejności należy przedstawić hierarchię założeń według której odbędzie się rozdzielanie szarych komórek i topienie czasu. Oto zgrubny podział na charakter założeń:

1. skuteczność prowadzenia zadania
2. funkcjonalność
3. powtarzalność
4. prostota działania
5. ograniczenie gabarytu i masy

Zgodnie z tą listą należy przedstawić co kryje się pod tymi pojęciami. Do tego posłuży wykaz zadań jakie konstrukcja jako całość będzie musiała spełniać, aby wykonać swoje zadanie. W punkcie 1 narzucono, aby sprzęt wykonywał swoje zadanie prawidłowo i nie stał się nieskutecznym działaniem marnującym zasoby (czas, amunicję, itp.). W tym celu należy rozpatrzyć czym jest skuteczność działania:

- rejestrowanie obrazu z kamery video
- uzyskiwanie prawidłowego obrazu
- intuicyjne sterowanie w celu wprowadzania poprawek położenia
- możliwość prowadzenia śledzenia

Funkcjonalność ma za zadanie zapewnić działanie na wymuszenia związane z charakterem działań wojskowych. Tutaj zostaje wykorzystany pomysł z prowadzenia obserwacji przy pomocy ognia prowadzonego z moździerza. Działanie to powinno być zapewnione dla najmniejszego moździerza dostępnego w działaniach zbrojnych (LM60), ponieważ dla coraz większych kalibrów znika ograniczenie wynikające z ograniczenia miejsca (punkt 5). Zatem założenia wyglądają następująco:

- możliwość szybkiego kalibrowania przyrządów
- trwałość mechaniczną na nagłe przyspieszenia
- zapewnienie trwałego przymocowania elementów nawigacyjnych

- kompaktowe ułożenie elementów
- przystosowanie obwodu do kształtu koła

Zadaniem powtarzalność jest nauczanie (lub zapewnienie) użytkownika o przewidywalnym działaniu platformy latającej i zapewnienie prawidłowej reakcji na zaistniałe zachowania występujące podczas eksploatacji sprzętu. Jednocześnie zapewnia to schemat zachowań, który jest nioceniony (i niedoceniany) zwłaszcza w sytuacji stresowej (jaką jest prowadzenie działań wojennych w bliskim kontakcie z przeciwnikiem). Dlatego funkcjonalność jest tak wysoko w hierarchii. Od funkcjonalności będzie zależeć między innymi:

- uruchomienie (lub wybudzenie) systemu
- autonomiczne i manualne otwarcie ramion
- stabilizacja lotu
- wprowadzenie układu odniesienia dla sterowania
- klawiszologia¹

Prostota działania jest elementem związanym bardziej z prowadzenie sztuki konstruowania urządzeń niż z wymaganiami działania. Istotą prostoty jest możliwość rozwiązywania skomplikowanych problemów za pomocą prostych mechanizmów. Pozwala to na szybką analizę układu oraz wprowadzania poprawek (eliminacji błędów). Brzytwa Okhama pozwala również na realizowanie zadań i procesów w sposób wystarczający (nie angażujący pełnej uwagi czy skupienia) co jest kluczowe podczas pracy w terenie. Należy również pamiętać o kosztach ponoszonych w czasie walk. Jeżeli bezzałogowiec jest traktowany jak amunicja nie należy przejmować się problemami takimi jak odzyskiwanie materiałów, ponowne użytkowanie, oszczędzanie na amunicji. Oczywiście prostota rozwiązania powinna równać się z opłacalnością. Jeżeli koszt jednostkowy będzie niski, to produkcja masowa będzie jeszcze tańsza niż detaliczna. Zatem celem tego punktu jest zachowanie prostoty działania przy jednoczesnym nie przekraczaniu budżetu na produkcję amunicji obserwacyjnej.

- stosowanie tylko wymaganych komponentów i podzespołów
- optymalizacja czynności pilota i strzelca
- optymalizacja działania mikrokontrolera
- prosta struktura działania programu
- możliwość poszerzenia możliwości o dodatkowe funkcje

Ostatni punkt jest wystarczająco jasny. Polega jedynie na tym iż całość musi zostać umieszczona na niewielkiej przestrzeni i dodatkowo dobrze gdyby za ograniczeniem wymiarów, ograniczałoby to również masę pocisku. Żołnierze transportujący tego rodzaju amunicję będą bardzo wdzięczni za szanowanie ich kręgosłupów.

¹termin zaczepliędzy z gier video. Oznacza intuicyjne lub wyuczone rozumienie sterowania przez elementy sterowania manipulatora

Rozdział 2

Realizacja sprzętowa - hardware

2.1 Wymagania sprzętowe

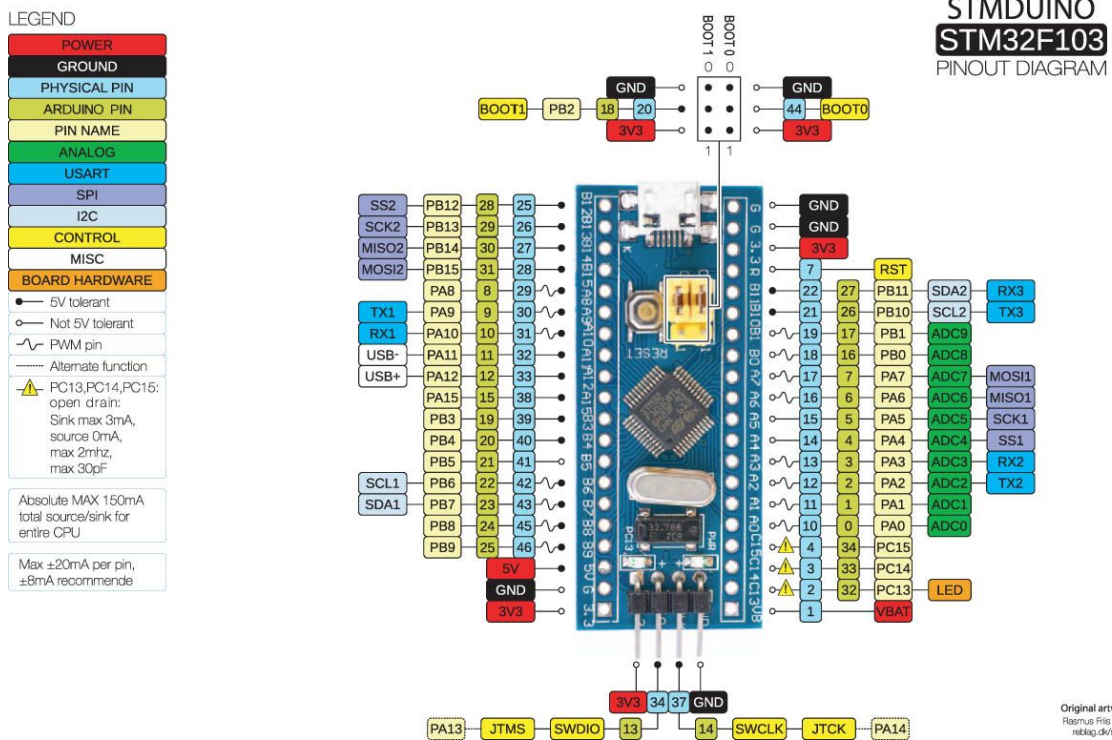
W wyprodukowania latającego podzespołu liczącego należy określić cel produkcji, założenia konstrukcyjne

2.2 Mikrokontroler

2.3 Akcelerometr

Tabela 2.1: Wykaz wymagań podzespołów sprzętowych

Lp.	Podzespół	Ilość	Pin
1.	ECS	4	
2.	akcelerometr	1	
3.	radio nadajnik	1	
4.	kamera video	1	
5.	serwomechanizm	4	



Rys. 2.1: Zestawienie pinów mikrokontrolera

Rozdział 3

Realizacja programowa - software

3.1 Narzędzia programistyczne

Wykaz rysunków

2.1	Zestawienie pinów mikrokontrolera	10
-----	---	----

Wykaz tabel

2.1	Wykaz wymagań podzespółów sprzętowych	9
-----	---	---

Dodatek A

Kod źródłowy

```
1 #include <libopencm3/stm32/rcc.h>
2 #include <libopencm3/stm32/gpio.h>
3
4 int main(void) {
5     rcc_periph_clock_enable(RCC_GPIOC);
6
7     gpio_set_mode(GPIOC,
8         GPIO_MODE_OUTPUT_2_MHZ,
9         GPIO_CNF_OUTPUT_PUSHPULL,
10        GPIO13);
11
12     while(1) {
13         for (int i = 0; i < 3000000; i++) {
14             __asm__("nop");
15         }
16
17         gpio_toggle(GPIOC, GPIO13);
18     }
19 }
```