

**Politechnika Rzeszowska
im. Ignacego Łukasiewicza
Wydział Budowy Maszyn i Lotnictwa
Katedra Awioniki i Sterowania**

Autorzy:

Dorota Chudzik

Mikołaj Kamiński

Aleksander Dykman

Temat pracy:

**Proces prototypowania urządzeń
mikrokomputerowych na przykładzie autopilota**

Kierunek kształcenia: LOTNICTWO I KOSMONAUTYKA

Specjalność: Awionika

Rok akademicki: 2017/2018

V MLD-S

Projektował: D. Chudzik, A. Dykman, M. Kamiński	Data:	Podpis:	Wydanie 1
Sprawdził:	Data:	Podpis:	Strona 1

Spis treści

1. Wstęp.....	3
1.1. Tematyka projektu.....	3
1.2. Harmonogram prac.....	3
1.3. Założenia do budowy urządzenia.....	4
1.4. Ograniczenia formalne.....	5
1.5. Przegląd istniejących rozwiązań.....	6
2. Projekt koncepcyjny.....	10
2.1. Sposób wykonania urządzenia.....	10
2.2. Schemat blokowy urządzenia wraz z opisem.....	10
2.3. Koncepcja oprogramowania.....	11
2.4. Dobór elementów.....	13
3. Projekt konstrukcyjny.....	15
3.1. Hardware.....	15
3.2. Software.....	17
3.2.1. Struktura oprogramowania.....	17
3.2.2. Model matematyczny.....	17
3.2.3. Regulator.....	21
4. Wykonanie.....	24
5. System kontroli wersji.....	26
5.1. Metodyka pracy.....	26
6. Podsumowanie prac.....	28

Projektował: D. Chudzik, A. Dykman, M. Kamiński	Data:	Podpis:	Wydanie	1
Sprawdził:	Data:	Podpis:	Strona	2

1. Wstęp

1.1. Tematyka projektu

Wybór tematu projektu jest uzasadniony chęcią poznania metod prototypowania oraz testowania oprogramowania przeznaczonego dla urządzeń wbudowanych.

W zakres prac wchodzi:

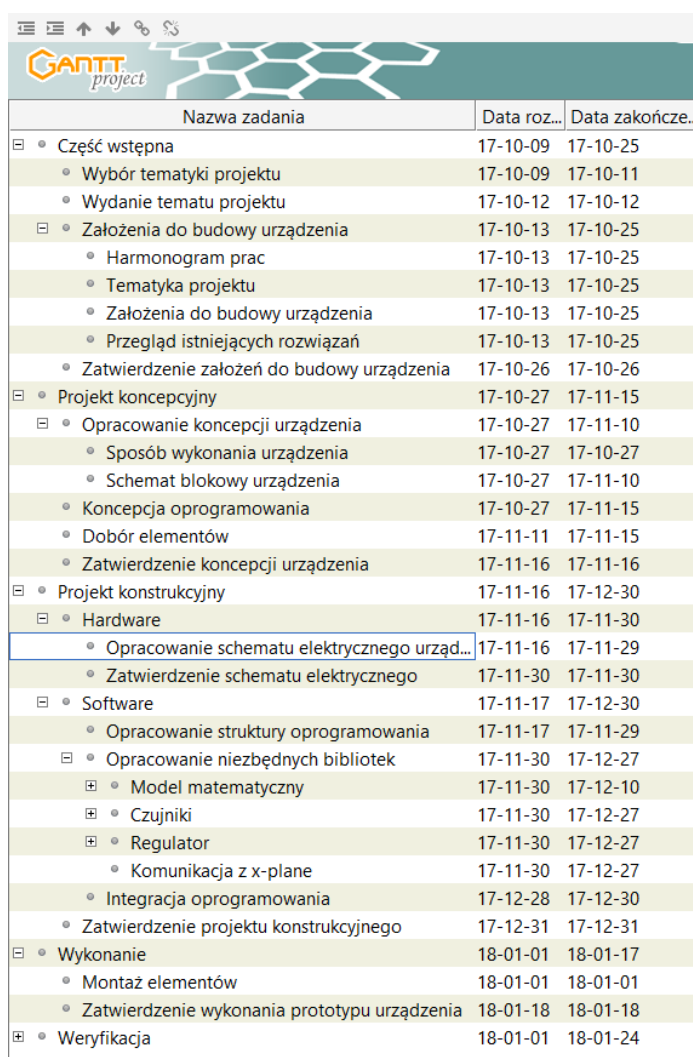
- wybór platformy sprzętowej (płytki prototypowej),
- projektowanie oprogramowania w zespole (z wykorzystaniem systemu kontroli wersji Git),
- testowanie oprogramowania (symulacje HIL oraz SIL).

Projekt pomoże w opracowaniu metodyki prototypowania urządzeń elektronicznych opartych o mikroprocesory.

1.2. Harmonogram prac

Zakres prac został zaplanowany w programie GanttProject. Główne zadania zostały przedstawione na planie widocznym na ilustracji 1. Cały harmonogram został załączony w repozytorium dostępnym pod adresem <https://github.com/AvionicsTeam/MAD> jako plik „Harmonogram prac.gan”.

Projektował: D. Chudzik, A. Dykman, M. Kamiński	Data:	Podpis:	Wydanie 1
Sprawdził:	Data:	Podpis:	Strona 3



Nazwa zadania	Data roz...	Data zakończe...
• Część wstępna	17-10-09	17-10-25
• Wybór tematyki projektu	17-10-09	17-10-11
• Wydanie tematu projektu	17-10-12	17-10-12
• Założenia do budowy urządzenia	17-10-13	17-10-25
• Harmonogram prac	17-10-13	17-10-25
• Tematyka projektu	17-10-13	17-10-25
• Założenia do budowy urządzenia	17-10-13	17-10-25
• Przegląd istniejących rozwiązań	17-10-13	17-10-25
• Zatwierdzenie założeń do budowy urządzenia	17-10-26	17-10-26
• Projekt koncepcyjny	17-10-27	17-11-15
• Opracowanie koncepcji urządzenia	17-10-27	17-11-10
• Sposób wykonania urządzenia	17-10-27	17-10-27
• Schemat blokowy urządzenia	17-10-27	17-11-10
• Koncepcja oprogramowania	17-10-27	17-11-15
• Dobór elementów	17-11-11	17-11-15
• Zatwierdzenie koncepcji urządzenia	17-11-16	17-11-16
• Projekt konstrukcyjny	17-11-16	17-12-30
• Hardware	17-11-16	17-11-30
• Opracowanie schematu elektrycznego urząd...	17-11-16	17-11-29
• Zatwierdzenie schematu elektrycznego	17-11-30	17-11-30
• Software	17-11-17	17-12-30
• Opracowanie struktury oprogramowania	17-11-17	17-11-29
• Opracowanie niezbędnych bibliotek	17-11-30	17-12-27
• Model matematyczny	17-11-30	17-12-10
• Czujniki	17-11-30	17-12-27
• Regulator	17-11-30	17-12-27
• Komunikacja z x-plane	17-11-30	17-12-27
• Integracja oprogramowania	17-12-28	17-12-30
• Zatwierdzenie projektu konstrukcyjnego	17-12-31	17-12-31
• Wykonanie	18-01-01	18-01-17
• Montaż elementów	18-01-01	18-01-01
• Zatwierdzenie wykonania prototypu urządzenia	18-01-18	18-01-18
• Weryfikacja	18-01-01	18-01-24

Ilustracja 1: Plan prac

1.3. Założenia do budowy urządzenia

Prototyp czterokanałowego autopilota do bezzałogowego statku powietrznego zostanie wykonany przy użyciu gotowego zestawu uruchomieniowego. Z tego powodu masa otrzymanego prototypu będzie różna od masy urządzenia, które będzie instalowane na obiekcie latającym.

Zakłada się, że wynikowe oprogramowanie będzie możliwe do implementacji na urządzeniach o różnej architekturze. Ponad to oprogramowanie będzie miało formę modułową, co ułatwi ewentualne zmiany.

Projektował: D. Chudzik, A. Dykman, M. Kamiński	Data:	Podpis:	Wydanie	1
Sprawdził:	Data:	Podpis:	Strona	4

Urządzenie zbudowane w oparciu o zaproponowaną metodykę:

1. zasilanie z portu USB,
2. komunikacja z czujnikami i mechanizmami wykonawczymi zainstalowanymi na pokładzie bezzałogowca,
3. możliwość zdalnego sterowania obiektem,
4. wybór pomiędzy danymi z zewnętrznego i wewnętrznego AHRS,
5. moc obliczeniowa wystarczająca do realizacji algorytmów: sterowania, filtracji sygnałów i obliczania kątów orientacji przestrzennej,
6. możliwość łatwego przeprowadzania symulacji hardware-in-the-loop.

1.4.Ograniczenia formalne

W celu określenia formalnych wymagań stawianych oprogramowaniu należy przeprowadzić analizę przepisów:

DO-178C

Projektowana metodyka prototypowania jest przeznaczona do użytku w przemyśle lotniczym dlatego musi spełniać wymagania stawiane oprogramowaniu zgodnie z normą DO-178C. Błąd oprogramowania projektowanego systemu może spowodować wiele ofiar oraz zniszczenie statku powietrznego. Ponad to utrata krytycznych funkcji może uniemożliwić kluczowe fazy lotu (start i lądowanie). Dlatego powinien on być projektowany i testowany jako najwyższy poziom DAL (Design Assurance Level) równy A co bezpośrednio implikuje możliwość jednej awarii na 109 godzin pracy. Bazując na tej normie możliwym będzie określenie WCET (Worst Case Execution Time) czyli wymaganego czasu wykonywania się programu dla najgorszych możliwych warunków pracy.

Projektował: D. Chudzik, A. Dykman, M. Kamiński	Data:	Podpis:	Wydanie 1
Sprawdził:	Data:	Podpis:	Strona 5

IEEE STD-1228

Norma opisana wyżej bazuje na założeniu spełnieniu warunków niższego poziomu. Te warunki zostały opisane w standardzie IEEE STD-1228, który opisuje minimalne wymagania dla oprogramowania mogącego mieć krytyczny wpływ na bezpieczeństwo. Dzięki temu standardowi możliwe jest określenie pewnych warunków pracy, które nie zostały zawarte w normie DO-178C.

ARP4754

Czasem, dzięki zastosowaniu pewnych mechanizmów możliwe jest obniżenie poziomu DAL. Takie mechanizmy zostały opisane we wskazówkach ARP4754. Jest to dokument dedykowany dla lotnictwa i jego głównym zadaniem jest ułatwienie doboru DAL.

1.5.Przegląd istniejących rozwiązań

Projekt ma być pomysłem na konkurencyjne urządzenie dla rozwiązań dostępnych na rynku.

Popularny **Ardupilot** to kontroler lotu dedykowany do helikopterów, samolotów, dronów oraz samochodów. Projekt zarówno hardware-u jak i software-u jest w pełni OpenSource. Oprogramowanie pozwala na stabilizację lotu oraz realizowanie autonomicznych misji w trybie autopilota. Po podłączeniu GPSa z kompasem istnieje możliwość planowania trasy lotu na komputerze. Układ posiada 6-osiowy akcelerometr i żyroskop, czujnik ciśnienia atmosferycznego oraz czujniki zewnętrzne.

Rozbudowanym następcą ardupilota jest **Pixhawk** o jeszcze większej prędkości i stabilności pracy. Zawiera 32-bitowy procesor ARM CortexM4, możliwość przejścia między trybami automatyczny/ ręczny, zapisanie lotu na karcie micro SD, interfejsy: UART, SPI, I2C, CAN. Zestaw czujników składa się z żyroskopu, akcelerometrów, magnetometru i barometru. Podobnie jak jego poprzednik jest uniwersalny do zastosowania dla wielu statków latających.

Projektował: D. Chudzik, A. Dykman, M. Kamiński	Data:	Podpis:	Wydanie 1
Sprawdził:	Data:	Podpis:	Strona 6



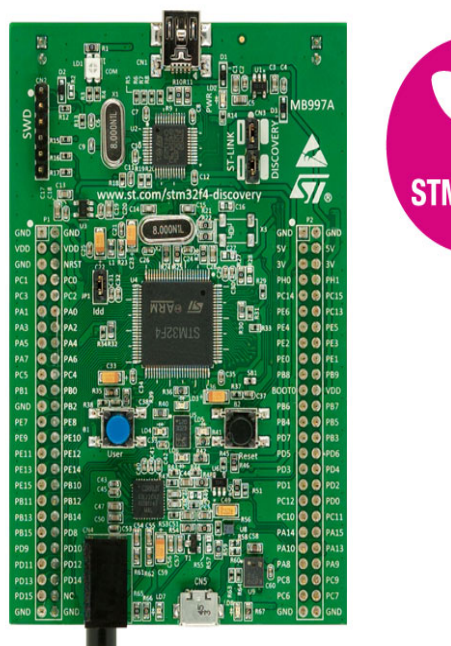
Ilustracja 2: Kontroler lotu Pixhawk PX4

Do prototypowania urządzeń mikrokomputerowych można wykorzystać gotowe zestawy uruchomieniowe zawierające mikroprocesor wraz z niezbędnymi modułami. Takie rozwiązanie pozwala oszczędzić czas na projektowanie i testowanie obwodów elektrycznych. Najpopularniejsze z nich to FPGA Altera oraz produkty firmy ST Microelectronics: STM32 Nucleo, STM32 Discovery.

STM32 Nucleo to seria płytek z mikrokontrolerami STM32. Integralną częścią jest programator ST-Link, co zwalnia z konieczności zakupu dodatkowych narzędzi. Układ wyprowadzeń zestawów uruchomieniowych NUCLEO jest kompatybilny ze standardem wyprowadzeń ARDUINO oraz ST Morpho. Podstawowa różnica pomiędzy wersjami zestawów jest w rodzajach mikroprocesorów oraz zainstalowanych komponentach.

STM32 Discovery to kolejna seria z mikrokontrolerami STM32 wyposażona w programator ST-LINK. Rozbudowany blok złącz wejść/wyjść pozwala na podłączenie dodatkowego sprzętu. Jest to seria dobrze wyposażona w komponenty. Zawierają między innymi czujniki MEMS (akcelerometr, żyroskop, magnetometr, mikrofon).

Projektował: D. Chudzik, A. Dykman, M. Kamiński	Data:	Podpis:	Wydanie 1
Sprawdził:	Data:	Podpis:	Strona 7



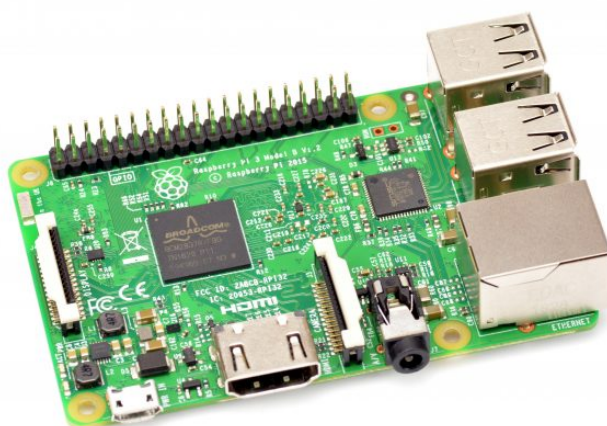
Ilustracja 3: STM Discovery F4

FPGA Altera czyli układy programowalne FPGA firmy Altera. Zaletą układów FPGA jest wydajniejsza praca od standardowych mikrokontrolerów dzięki możliwości konfiguracji wbudowanego oprogramowania bez fizycznego demontażu. Układy te mogą wykonywać równocześnie wiele zadań w krótkim czasie. Jednak skonfigurowanie tych układów jest dość skomplikowane. Ich wadą jest również wysoka cena.

Bardziej zaawansowanym rozwiązaniem jest płytką **Raspberry PI**. Jest to w pełni funkcjonalny komputer z systemem operacyjnym. Może obsługiwać zaawansowane czujniki i peryferia oraz interfejsy: SPI, I2C, UART, CSI, DSI. Urządzenie posiada dużą moc obliczeniową, a procesor składa się z czterech rdzeni. Raspberry PI ma wyraźną przewagę w rozwiązaniach wymagających rozbudowanego oprogramowania, dlatego też jest dedykowane do zaawansowanych projektów.

Projektował: D. Chudzik, A. Dykman, M. Kamiński	Data:	Podpis:	Wydanie	1
Sprawdził:	Data:	Podpis:	Strona	8

Projekt: *Proces prototypowania urządzeń mikrokomputerowych na przykładzie autopilota*



Ilustracja 4: Raspberry PI model 3

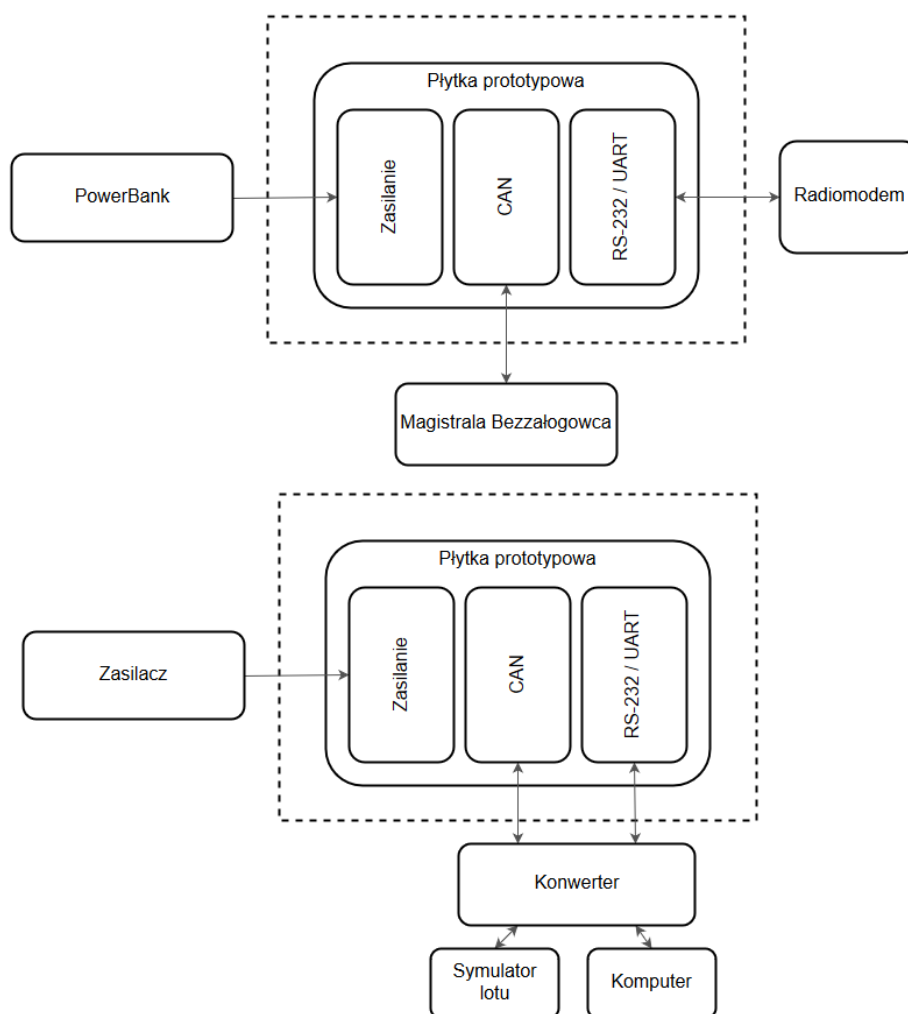
Projektował: D. Chudzik, A. Dykman, M. Kamiński	Data:	Podpis:	Wydanie 1
Sprawdził:	Data:	Podpis:	Strona 9

2. Projekt koncepcyjny

2.1.Sposób wykonania urządzenia

Prototyp autopilota zostanie wykonany z wykorzystaniem płytki prototypowej, która musi mieć możliwość implementacji założonych funkcjonalności. Aby założenia zostały spełnione płytka musi zawierać niezbędne podzespoły i mieć możliwość obsługi właściwych magistrali przesyłu danych.

2.2.Schemat blokowy urządzenia wraz z opisem



Ilustracja 5: Schemat blokowy urządzenia

Projektował: D. Chudzik, A. Dykman, M. Kamiński	Data:	Podpis:	Wydanie	1
Sprawdził:	Data:	Podpis:	Strona	10

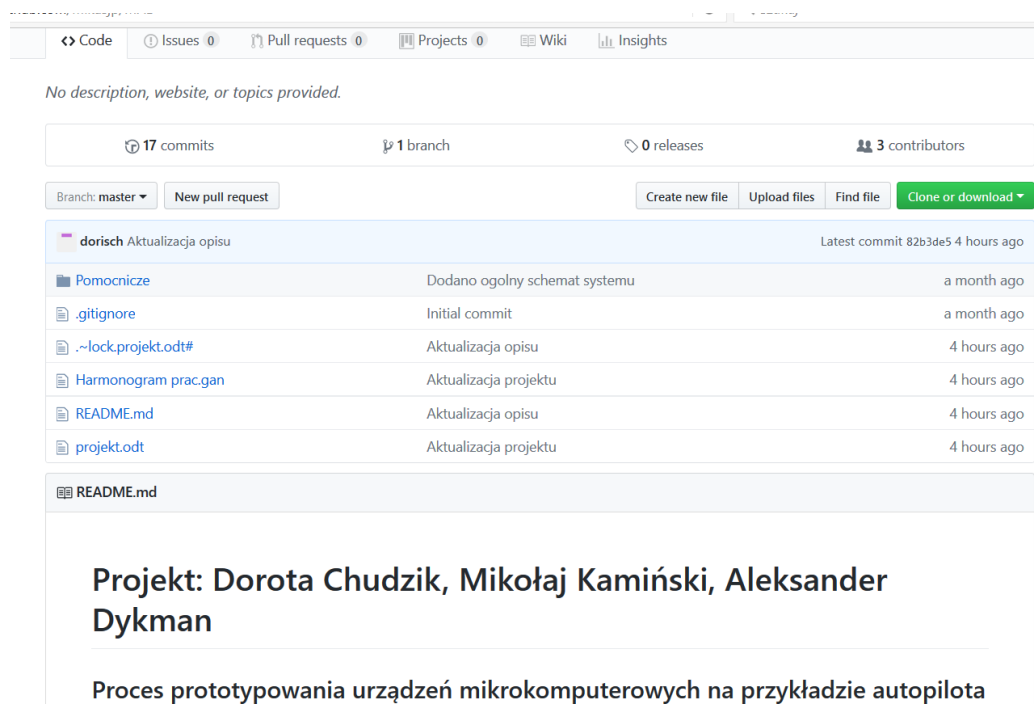
lp	Założenie	Realizacja
1.	Zasilanie z portu USB	Płytką prototypową wyposażoną w złącze USB.
2.	Komunikacja z czujnikami i mechanizmami wykonawczymi zainstalowanymi na pokładzie bezzałogowca	Wykorzystanie płytki prototypowej obsługującej magistralę CAN.
3.	Możliwość zdalnego sterowania obiektem	Wykorzystanie płytki prototypowej obsługującej magistralę UART / RS-232.
4.	Wybór pomiędzy danymi z zewnętrznego i wewnętrznego AHRS	Wykorzystanie płytki prototypowej posiadającej trójosiowe: akcelerometry, żyroskopy i magnetometry.
5.	Moc obliczeniowa wystarczająca do realizacji algorytmów: sterowania, filtracji sygnałów i obliczania kątów orientacji przestrzennej	Wykorzystanie płytki prototypowej z dostatecznie szybkim procesorem.
6.	Możliwość łatwego przeprowadzania symulacji hardware-in-the-loop	Stworzenie dodatkowej płytki będącej konwerterem UART/RS-232 oraz zawierającej driver magistrali CAN.

2.3. Koncepcja oprogramowania

Oprogramowanie zostanie udostępnione na platformie <https://github.com/> wraz z dokumentacją projektu. W tym celu założono repozytorium i udostępniono następujące pliki: harmonogram prac (plan pracy w programie GanttProject) oraz projekt.odt (niniejszy dokument). Wraz z rozwojem oprogramowania będą dodawane i aktualizowane kolejne biblioteki.

Projektował: D. Chudzik, A. Dykman, M. Kamiński	Data:	Podpis:	Wydanie 1
Sprawdził:	Data:	Podpis:	Strona 11

Projekt: Proces prototypowania urządzeń mikrokomputerowych na przykładzie autopilota



Ilustracja 6 Screen repozytorium na github.com

Oprogramowanie tworzone na potrzeby projektu będzie podzielone na trzy moduły:

- moduł zawierający funkcje służące do obsługi regulatorów PID,
- moduł zawierający funkcje służące do odczytywania wielkości fizycznych mierzonych przez czujniki,
- moduł umożliwiający komunikację systemu ze sterowanym obiektem.

Taka architektura zapewni szereg możliwości:

- możliwość łatwego testowania poszczególnych funkcji oprogramowania,
- możliwość łatwej implementacji modułów na innych urządzeniach,
- wyraźny podział obowiązków na uczestników projektu,
- możliwość bezkonfliktowej pracy z wykorzystaniem systemu kontroli wersji.

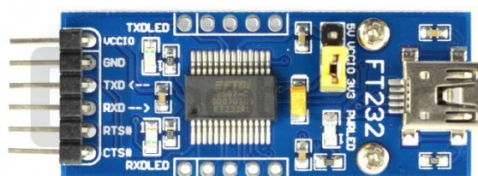
Projektował: D. Chudzik, A. Dykman, M. Kamiński	Data:	Podpis:	Wydanie	1
Sprawdził:	Data:	Podpis:	Strona	12

2.4. Dobór elementów

W celu realizacji projektu została wybrana płytki prototypowa **STM32F303** Discovery.

Główną jednostką obliczeniową jest STM32F303VCT6 czyli 32 – bitowy procesor z rdzeniem Cortex M4F, który jest przeznaczony do zaawansowanych systemów wbudowanych gdzie pojawia się potrzeba wykonywania wielu obliczeń w czasie rzeczywistym. Ten układ integruje w sobie dużo różnych peryferii, między innymi jednostkę zmiennoprzecinkową oraz wysoce wydajny rdzeń. Procesor taktuje z częstotliwością 72 MHz i ma 48 kB pamięci RAM oraz 256 kB pamięci Flash. Na płytce znajduje się akcelerometr z magnetometrem, moduł **LSM303DLHC**, który do komunikacji z jednostką centralną używa magistrali I2C. Ponadto na płytce jest również żyroskop **L3GD20**, który pracuje w technologii MEMS. Moduł ten może komunikować się za pośrednictwem magistrali I2C lub SPI. Ponadto dobrany zestaw ma możliwość komunikacji za pośrednictwem magistrali USART oraz CAN. Na płytce został umieszczony programator ST-Link co umożliwi programowanie i zasilanie układu przez USB.

W celu konwersji sygnału UART na USB zostanie wykorzystany układ **FT232**.



Ilustracja 7: FT 232

Sygnał wyjściowy CAN (z płytki prototypowej) zostanie przetworzony przy pomocy sterownika **SN65HVD230**.

Projektował: D. Chudzik, A. Dykman, M. Kamiński	Data:	Podpis:	Wydanie 1
Sprawdził:	Data:	Podpis:	Strona 13



Ilustracja 8: SN65HVD230

Ponad to podczas realizacji projektu zostaną użyte drobne elementy, takie jak gniazdo RS232 i przewody połączeniowe różnego typu.



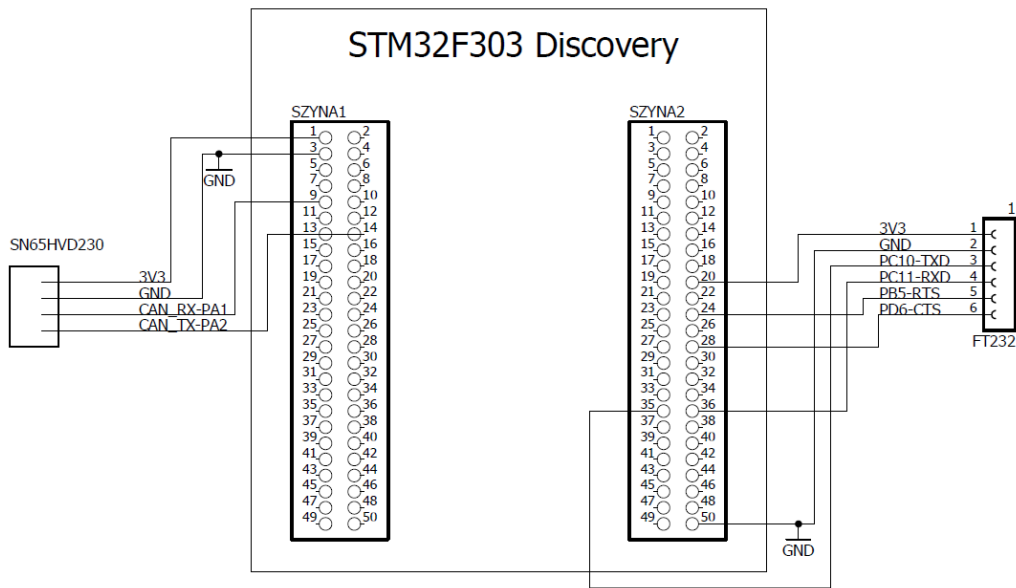
Ilustracja 10: Gniazdo RS232



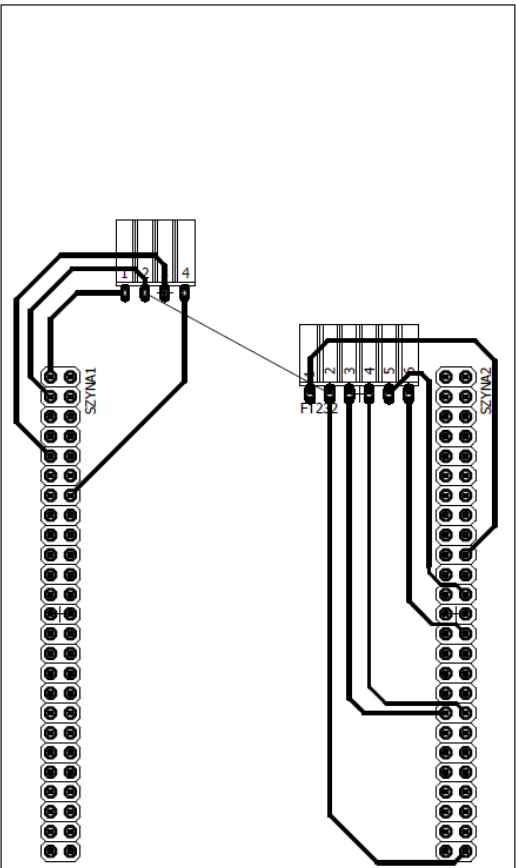
Ilustracja 9: Przewody połączeniowe

Element	Cena [zł]
STM32F303	95
FT232	23
SN65HVD230	19
Gniazdo RS232	0,50
Przewody połączeniowe	15
SUMA	152,50

Projektował: D. Chudzik, A. Dykman, M. Kamiński	Data:	Podpis:	Wydanie 1
Sprawdził:	Data:	Podpis:	Strona 14



Ilustracja 12: Schemat płytki

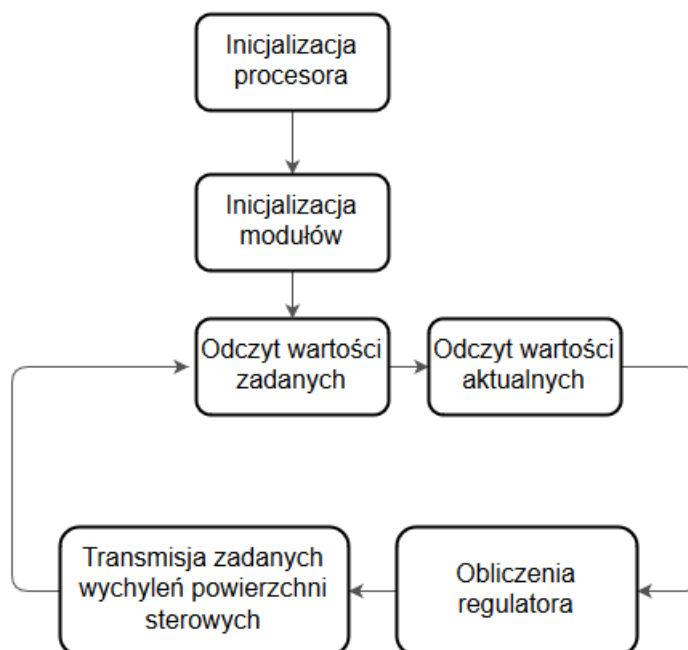


Ilustracja 13: Płytki PCB

Projektował: D. Chudzik, A. Dykman, M. Kamiński	Data:	Podpis:	Wydanie 1
Sprawdził:	Data:	Podpis:	Strona 16

3.2. Software

3.2.1. Struktura oprogramowania



Ilustracja 14: Ogólna struktura oprogramowania

3.2.2. Model matematyczny

W celu przetestowania prototypu w symulacji HIL niezbędne jest stworzenie modelu matematycznego sterowanego obiektu. Sterowanym obiektem będzie samolot PZL M28 Skytruck. Model matematyczny samolotu zostanie stworzony na podstawie danych pozyskanych podczas prób w symulowanym locie. W tym celu zostanie wykorzystane oprogramowanie X-Plane 10.

Planowany program prób:

Dla kanałów przechylenie i pochylania:

1. Wymuszenie w postaci skoku jednostkowego w prawo/ujemny kąt pochylenia,
2. Powrót do neutrum,

Projektował: D. Chudzik, A. Dykman, M. Kamiński	Data:	Podpis:	Wydanie 1
Sprawdził:	Data:	Podpis:	Strona 17

3. Wymuszenie w postaci skoku jednostkowego w lewo/dodatni kąt pochylenia,
4. Powrót do neutrum,
5. Wymuszenie trapezowe w prawo/ujemny kąt pochylenia,
6. Powrót do neutrum,
7. Wymuszenie trapezowe w lewo/dodatni kąt pochylenia,
8. Powrót do neutrum
9. Wymuszenie 3, 2, 1, 1 w prawo/ujemny kąt pochylenia,
10. Powrót do neutrum
11. Wymuszenie 3, 2, 1, 1, w lewo/dodatni kąt pochylenia
12. Powrót do neutrum.

Dla kanału odchyłania:

1. Ślizg w prawo,
2. Powrót do neutrum,
3. Ślizg w lewo,
4. Powrót do neutrum.

Wszystkie powyższe kroki powinny być wykonane dla trzech prędkości lotu samolotu w celu stworzenia modelu matematycznego prawdziwego dla większej ilości punktów pracy.

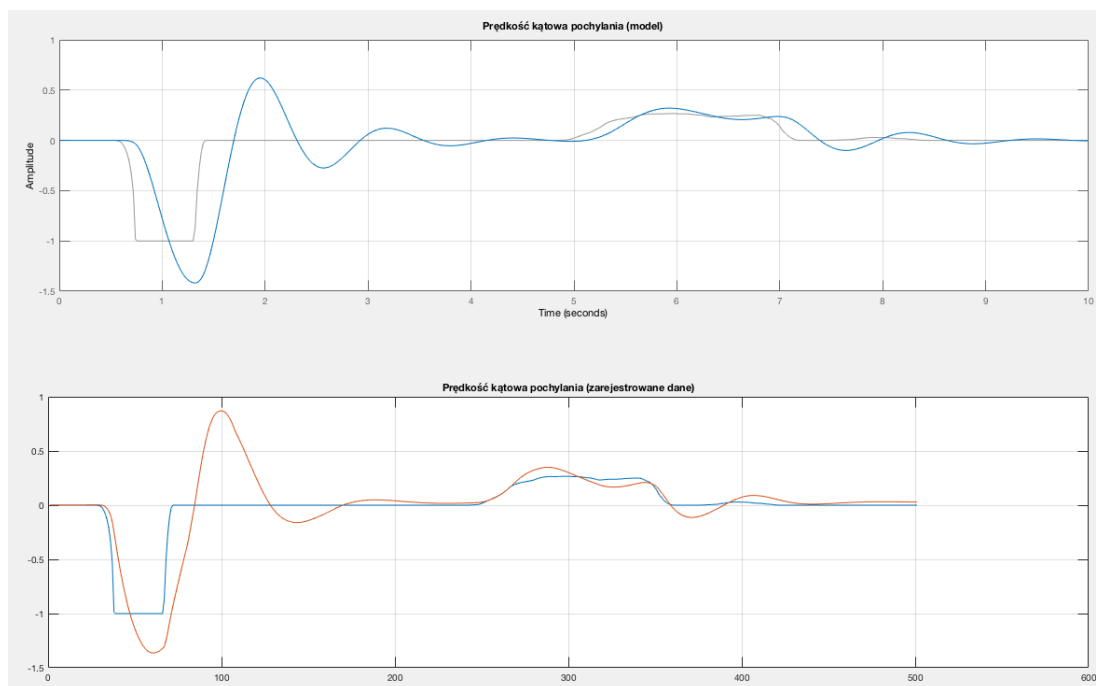
3.2.2.1 Zarejestrowane dane

Dane zarejestrowane podczas prób w locie są zapisywane przez oprogramowanie X-Plane 10 w nietypowym formacie. Uniemożliwia to łatwe skorzystanie z nich w innym oprogramowaniu. W celu umożliwienia odczytu zarejestrowanych danych w oprogramowaniu MATLAB, powstało narzędzie konwertujące pliki w nietypowym formacie X-Plane 10 do plików CSV (comma separated values), który jest dużo bardziej powszechny. Konwerter „xplane2csv” został napisany w języku programowania Python.

Projektował: D. Chudzik, A. Dykman, M. Kamiński	Data:	Podpis:	Wydanie 1
Sprawdził:	Data:	Podpis:	Strona 18

3.2.2.2 Wyznaczanie modelu

Model matematyczny samolotu został stworzony przy pomocy oprogramowania MATLAB. W tym celu powstał skrypt wyznaczający transmitancje poszczególnych kanałów samolotu na podstawie zarejestrowanych sygnałów wymuszających oraz sygnałów odpowiedzi.



Ilustracja 15: Porównanie zachowania się (od góry) modelu i rzeczywistego obiektu na to samo wymuszenie

Powyższe wykresy przedstawiają nam odpowiednio odpowiedzi (prędkość kątowna pochylenia) na to samo wymuszenie (od góry): modelu i prawdziwego samolotu. Jak widać, model wystarczająco wiernie odwzorowuje rzeczywisty samolot aby uznać go za prawidłowy. Operacja identyfikacji została powtórzona dla pozostałych kanałów.

Projektował: D. Chudzik, A. Dykman, M. Kamiński	Data:	Podpis:	Wydanie 1
Sprawdził:	Data:	Podpis:	Strona 19

Uzyskano następujące transmitancje:

```
Command Window
q =
      27.61
-----
s^2 + 2.659 s + 27.92
Continuous-time transfer function.

|
theta =
      27.61
-----
s^3 + 2.659 s^2 + 27.92 s
Continuous-time transfer function.

w =
      585.1
-----
s^3 + 2.822 s^2 + 11.07 s
Continuous-time transfer function.

u =
      -122.2
-----
s^3 + 3.83 s^2 + 11.64 s
Continuous-time transfer function.
```

Ilustracja 16: Wyznaczone transmitancje ruchu podłużnego

Wszystkie skrypty zostały umieszczone w repozytorium w folderze ModelMatematyczny:

<https://github.com/AvionicsTeam/MAD/ModelMatematyczny/> .

Projektował: D. Chudzik, A. Dykman, M. Kamiński	Data:	Podpis:	Wydanie	1
Sprawdził:	Data:	Podpis:	Strona	20

3.2.3. Regulator

W projekcie zostały zaimplementowane regulatory PID typowe dla autopilotów lotniczych. Wartościami wejściowymi regulatora będą wartości zadane przez pilota oraz zmierzone wartości rzeczywiste, na podstawie których zostanie wyliczona ich różnica (uchyb). Regulator ten ma za zadanie wygenerować sygnał sterujący, który w sposób sposób stabilny i kontrolowany zmieni parametry lotu samolotu do zadanych.

Etapy pracy nad implementacją regulatora:

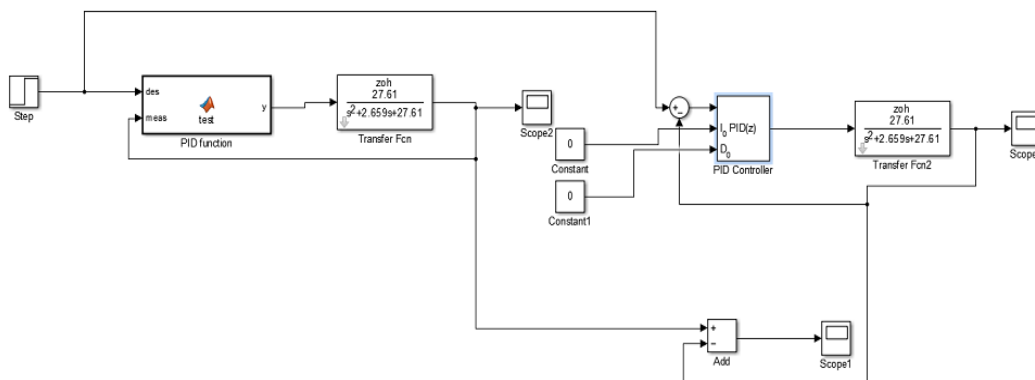
1. napisanie biblioteki realizującej algorytm regulatora PID w języku C,
2. testy stworzonych algorytmów w symulacji SIL przy pomocy oprogramowania MATLAB Simulink,
3. implementacja biblioteki do oprogramowania autopilota.

Biblioteka realizująca algorytm regulatora PID została przygotowana dla czterech regulatorów o różnych parametrach. W funkcji został dołączony system antywinupowy dla członu całkującego. Wartości graniczne należy podać przy inicjalizacji konkretnego regulatora. Cały kod realizujący algorytm został umieszczony w repozytorium w folderze **PID controller/PID** jako plik PID.c.

Inicjalizacja regulatorów, czyli podanie parametrów wejściowych oraz wywołanie funkcji PID znajduje się w bibliotece test.c, która została również załączona do folderu PID controller/PID.

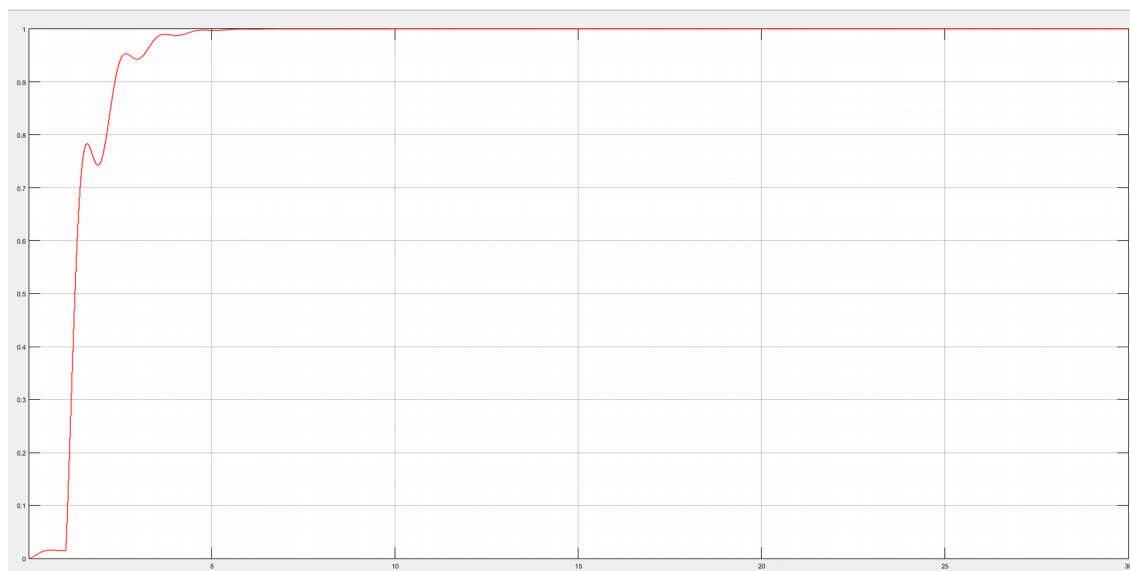
Poprawność działania funkcji została przetestowana w programie Matlab Simulink za pomocą bloku „Matlab function”. Pozwolił on na implementację algorytmu w języku C do modelu w Simulinku.

Projektował: D. Chudzik, A. Dykman, M. Kamiński	Data:	Podpis:	Wydanie 1
Sprawdził:	Data:	Podpis:	Strona 21



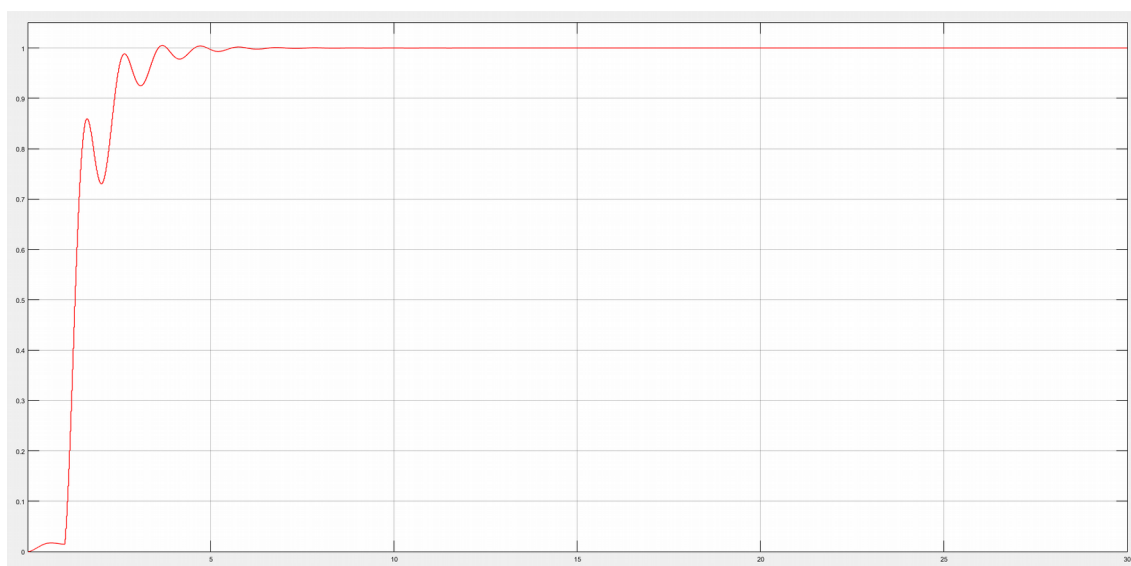
Ilustracja 17: Model w programie Matlab Simulink do testowania biblioteki PID

Dla transmitancji $q(s)$ wyznaczonej w poprzednim etapie pracy został zbudowany model z regulatorem PID i wymuszeniem skokowym. Następnie zdyskretyzowano model i dobrano wartości regulatora, które zainicjalizowano w bibliotece test.c dla regulatora PID o numerze 0. Wyniki symulacji porównano z wynikami symulacji dla takiego samego modelu, w którym zastąpiono bloczek PID bloczkiem „Matlab function” jak na ilustracji 12. Wyniki symulacji przedstawiono na ilustracjach 13-15.

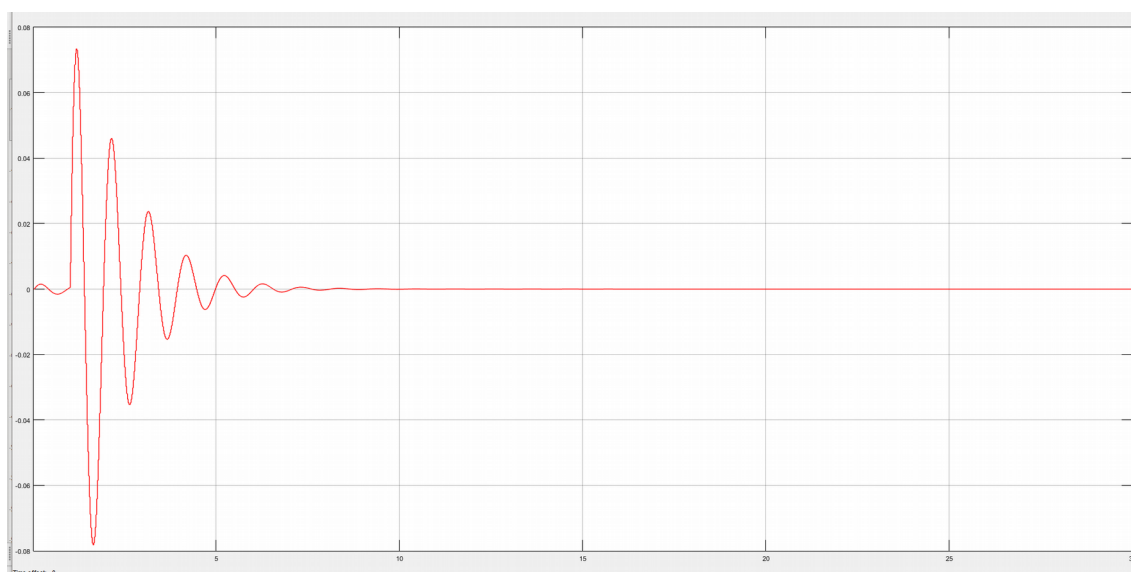


Ilustracja 18: Wyniki dla regulatora zaimplementowanego z biblioteki w języku C

Projektował: D. Chudzik, A. Dykman, M. Kamiński	Data:	Podpis:	Wydanie 1
Sprawdził:	Data:	Podpis:	Strona 22



Ilustracja 19: Wyniki dla regulatora PID wywołanego z z bločka PID



Ilustracja 20: Różnica działania regulatorów

Projektował: D. Chudzik, A. Dykman, M. Kamiński	Data:	Podpis:	Wydanie 1
Sprawdził:	Data:	Podpis:	Strona 23

4. Wykonanie

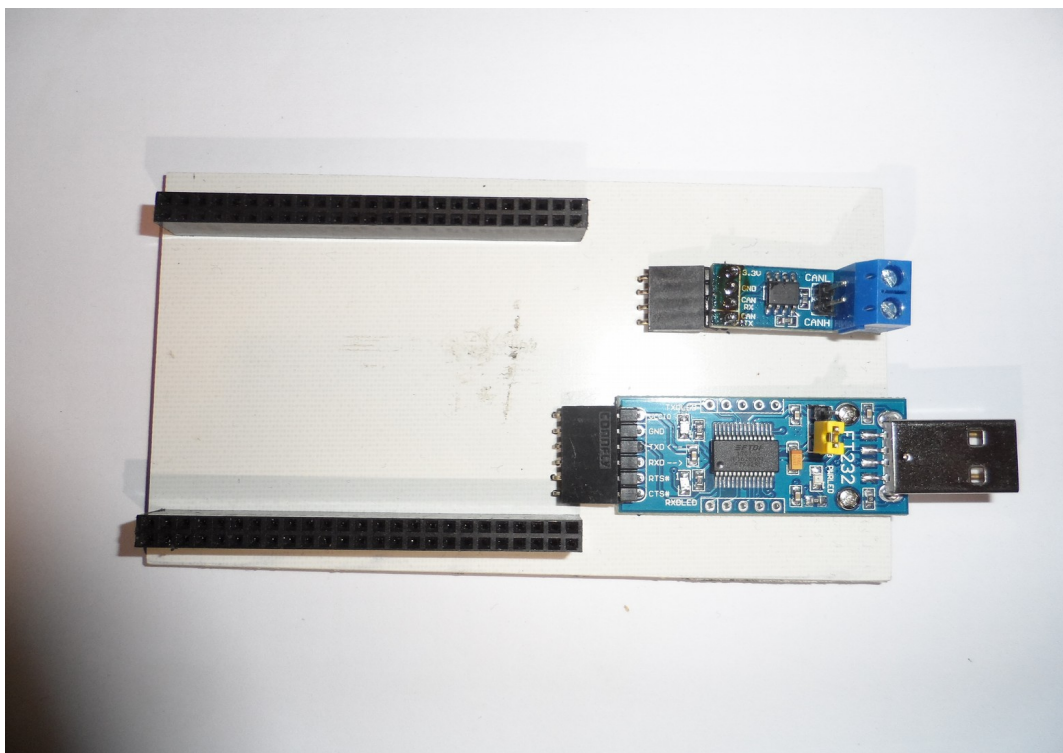
Płytką pcb została wykonana w następujących krokach:

- Przycięcie laminatu na pożądany wymiar
- Wydrukowanie schematu płytki z programu Eagle drukiem laserowym na papierze kredowym
- Przy pomocy laminarki naniesienie wydrukowanego schematu na laminatu
- Odklejenie druku i poprawienie połączeń
- Oczyszczenie laminatu
- Kąpiel płytki w wytrawiaczu
- Wywiercenie otworów (w tym punkcie popełniono niekrytyczny błąd: użyto wiertła 1mm, które było za duże i czasem zrywało miedź z padu, mimo to udało się zostawić minimum potrzebnej do zalutowania miedzi)
- Przylutowanie goldpinów do płytki

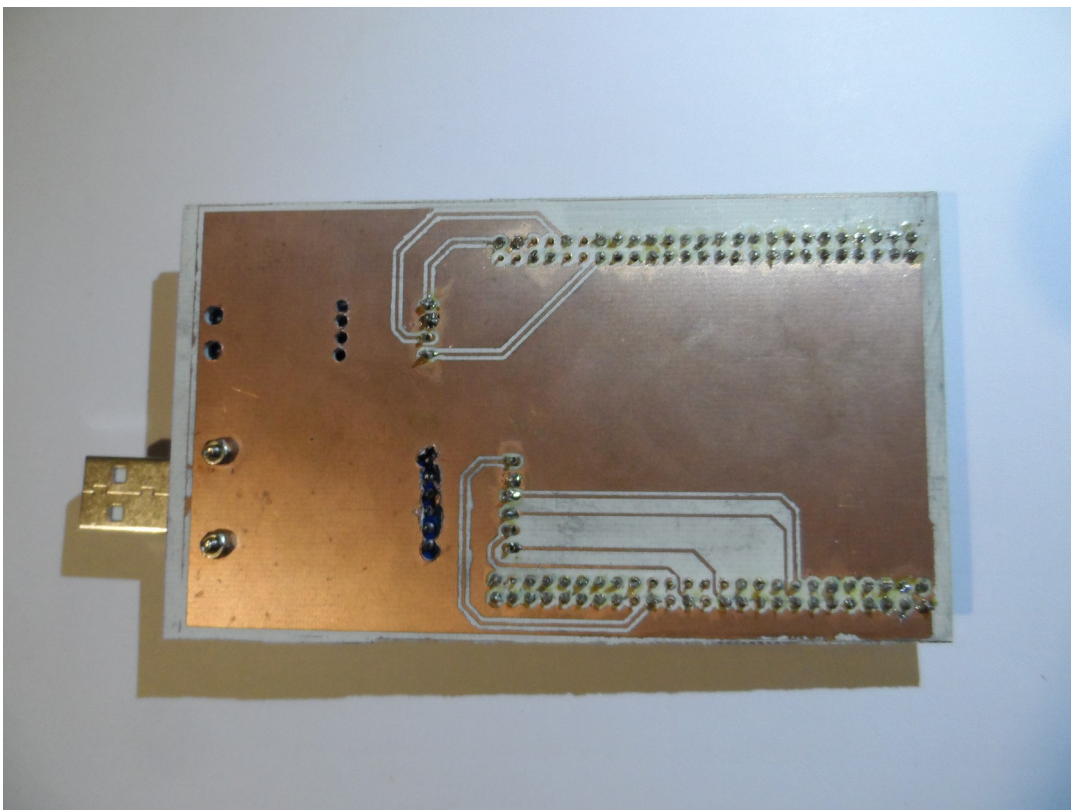
Ostatecznie wykonanie płytki kosztowało ok. 10 zł (nie licząc czasu pracy). Nie jest to jednak wielka oszczędność w porównaniu z masową produkcją tego typu płytek, które kosztują 35 zł (wraz z robocizną). Ponadto oddając płytkę do wydruku w profesjonalnej firmie nie trzeba się martwić wątpliwej jakości wierceniem i lutowaniem.

W celu sprawdzenia działania połączeń, wszystkie piny funkcyjne zostały sprawdzone przy pomocy multimetru (funkcja ciągłość obwodu). Wynik testu jest pozytywny i płytka spełnia przeznaczone zadanie.

Projektował: D. Chudzik, A. Dykman, M. Kamiński	Data:	Podpis:	Wydanie 1
Sprawdził:	Data:	Podpis:	Strona 24



Ilustracja 21: Płytką PCB - góra



Ilustracja 22: Płytką PCB - dół

Projektował: D. Chudzik, A. Dykman, M. Kamiński	Data:	Podpis:	Wydanie 1
Sprawdził:	Data:	Podpis:	Strona 25

5. System kontroli wersji

Jednym z narzędzi, które ułatwiło zespołowi pracę nad projektem był system kontroli wersji Git. Repozytorium projektu zostało umieszczone w serwisie internetowym GitHub, który jest przeznaczony do hostowania projektów wykorzystujących system kontroli wersji Git.

5.1. Metodyka pracy

Korzystając ze wsparcia Git-a dla rozgałęzionego procesu tworzenia oprogramowania, każdy z członków zespołu mógł bez przeszkód pracować nad swoją częścią projektu.

Active branches			
KontrolaWersji	Updated an hour ago by mikasjp	0 0	New pull request
ModelMatematyczny	Updated 18 hours ago by mikasjp	2 0	#6 Merged
PID	Updated 18 hours ago by dorisch	10 0	#5 Merged
Keil	Updated 3 days ago by greeback	22 1	New pull request

Ilustracja 23: Lista gałęzi w repozytorium umieszczonym w serwisie GitHub

Dzięki takiemu rozwiązaniu możliwe było również śledzenie postępów prac w czasie. Po zakończeniu prac nad daną funkcjonalnością możliwe jest zgłoszenie chęci włączenia dotychczasowych zmian do głównej gałęzi projektu poprzez funkcję „pull request” w serwisie GitHub.

Projektował: D. Chudzik, A. Dykman, M. Kamiński	Data:	Podpis:	Wydanie	1
Sprawdził:	Data:	Podpis:	Strona	26

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base: master + compare: Keil ✓ Able to merge. These branches can be automatically merged.

Utworzono projekt w Keil

Write

Preview

AA B i “ > < : : : : ↶ @

Leave a comment

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

Styling with Markdown is supported

Create pull request

Reviewers

No reviews—request one

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

< 1 commit

4 files changed

0 commit comments

1 contributor

Ilustracja 24: Tworzenie "pull request" w serwisie GitHub

Stworzenie nowego „pull request” umożliwiało dyskusję nad zmianami wprowadzonymi przez członka zespołu, ewentualne korekty błędów oraz rozwiązywanie konfliktów między plikami. Jeśli zespół zdecydował, że zmiany wprowadzone w gałęzi są zgodne z założeniami oraz nie zawierają błędów, następował tak zwany „merge” czyli włączenie zmian do głównej gałęzi projektu.

Projektował: D. Chudzik, A. Dykman, M. Kamiński	Data:	Podpis:	Wydanie 1
Sprawdził:	Data:	Podpis:	Strona 27

6. Podsumowanie prac

Konstrukcja urządzenia została wykonana przez zespół w pełni w zakresie hardware oraz częściowo software. Hardware obejmował zaprojektowanie i wykonanie płytki prototypowej. Płytką została wykonana poprawnie i jest przygotowana do zaprogramowania.

Software został podzielony na sekcje: czujniki, regulator, komunikacja z X-Plane oraz integracja oprogramowania. Dodatkowo należało wyznaczyć model matematyczny obiektu latającego.

W ramach stworzenia modelu matematycznego zostały przeprowadzone próby w locie na symulatorze lotu, obróbka uzyskanych w ten sposób danych oraz wyznaczenie transmitancji w kanale podłużnym. Prace zostały opóźnione ze względu na napotkane przeszkody: przetworzenie danych zarejestrowanych podczas prób w locie, konfiguracja symulatora, identyfikacja modelu ze względu na dobór parametrów wygenerowanych transmitancji a co za tym idzie wierności względem rzeczywistego modelu.

Ze względu na opóźnienia w uzyskaniu modelu matematycznego oraz problemy z konfiguracją środowiska Keil (wgranie odpowiednich bibliotek), nie udało się stworzyć pełnego oprogramowania (czujniki, komunikacja z X-Plane, integracja systemu). Zostały napisane biblioteki regulatora PID. Kod algorytmu regulatora został zweryfikowany poprzez testy w programie Matlab Simulink. Stwierdzono, że działa poprawnie. Należałoby go jednak udoskonalić poprzez dodanie filtra do członu różniczkującego.

Reasumując, zadania projektowe zrealizowane wg. planu gantta to:

- część wstępna w całości: wybór tematyki projektu, założenia do budowy urządzenia, harmonogram prac, przegląd istniejących rozwiązań;
- projekt koncepcyjny w całości: opracowanie koncepcji urządzenia (sposób wykonania, schemat blokowy), koncepcja oprogramowania, dobór elementów);

Projektował: D. Chudzik, A. Dykman, M. Kamiński	Data:	Podpis:	Wydanie 1
Sprawdził:	Data:	Podpis:	Strona 28

Projekt: *Proces prototypowania urządzeń mikrokomputerowych na przykładzie autopilota*

- projekt konstrukcyjny w zakresie: hardware (opracowanie schematu elektrycznego) , software (opracowanie struktury oprogramowania, opracowanie niezbędnych bibliotek: model matematyczny, regulator);
- wykonanie: montaż elementów.

Etapy, których nie udało się zrealizować:

- opracowanie niezbędnych bibliotek: czujniki, komunikacja z X-Plane, integracja oprogramowania;
- weryfikacja: testy HIL.

Projektował: D. Chudzik, A. Dykman, M. Kamiński	Data:	Podpis:	Wydanie 1
Sprawdził:	Data:	Podpis:	Strona 29