**Fixed wing drone stabilization and control system**



**Fixed wing drone stabilization and control system**

**Tro- og loveerklæring**

*Det erklæres herved på tro og love, at undertegnede egenhændigt og selvstændigt har udformet denne opgave. Alle citater i teksten er markeret som sådanne, og opgaven eller væsentlige dele af den har ikke tidligere været og er ikke aktuelt fremlagt i anden bedømmelsesammenhæng.*

*Undertegnede er gjort bekendt med, at overtrædelse af reglerne om videnskabelig redelighed behandles i henhold til §19 i Bekendtgørelse om prøver og eksamen i erhvervsrettede uddannelser nr. 1016 af 24/08/2010.*

**Solemn Declaration**

*I solemnly declare that I have personally and independently created this report. I have clearly marked any and all quotes in the text as such, and neither the report nor any essential parts of it are at present or have previously been submitted for any other examination.*

*I am aware that any violation of the rules on academic integrity shall be treated in accordance with Article 19 of the Danish Order No 1016 of 24 August 2010 on Tests and Examinations in vocational educations.*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Miroslav Lakota 1.6.2018

# Synopsis

This report describes research, development and the process of making a fixed wing drone stabilisation system, which I made as a Final project in the 4th semester of IT technology programme. It is supposed to give a reader an overview of what it takes to write a drone control and stabilisation system, going through a basic physical theories involved in such design.

# 

**Tabl****e of Contents**

[Synopsis 3](#__RefHeading___Toc227_1667555828)

[Preface 5](#__RefHeading___Toc229_1667555828)

[Introduction 6](#__RefHeading___Toc231_1667555828)

[Problem formulation and project scope 7](#__RefHeading___Toc233_1667555828)

[The Project 8](#__RefHeading___Toc235_1667555828)

[The theory of airplane control 8](#__RefHeading___Toc2482_2229226070)

[ChibiOS RTOS 9](#__RefHeading___Toc2484_2229226070)

[Hardware build 10](#__RefHeading___Toc2486_2229226070)

[Control algorithm 12](#__RefHeading___Toc2488_2229226070)

[Writing software 13](#__RefHeading___Toc2490_2229226070)

[Testing and debugging 14](#__RefHeading___Toc2492_2229226070)

[Conclusion 15](#__RefHeading___Toc237_1667555828)

[List of references 16](#__RefHeading___Toc239_1667555828)

[Bibliography 17](#__RefHeading___Toc241_1667555828)

[Appendices 18](#__RefHeading___Toc243_1667555828)

# Preface

Making a project combining hardware and software together is neither cheap, nor easy to make. It requires material, time, tools, and lots of patience. Therefore I would like to thank the company Danish Aviation Systems ApS which provided me with work space, all hardware parts, my colleagues Riccardo Miccini and Allan Hein for their advices, help with programming and patience while explaining, Steven Friberg for help with construction of the airplane frame and transportation to the test field.

# Introduction

Unmanned Aviation has a relatively short history. First unmanned hydrogen filled airships started to appear in the 19th century. During the 2nd World War they found their use mainly as dummy targets for artillery. With improvements to electronics and a decrease in price in 2000s, the number of amateur pilots grew. Flying an airplane was relatively difficult and required hours of practice. A half decade later, manufacturers started to produce small MEMS (Microelectromechanical systems) gyroscopes and IMUs (Inertial measurement units) for first smartphones. Technology came to the point, when hardware equipment is light enough (few grams), cheap and at the same time powerful and accurate enough to estimate the orientation of a flying object. From this point, it was just a small step to first attempts to control airplanes autonomously.

Currently, companies producing commercially available drones e.g. DJI or SenseFly are trying to decrease the level of difficulty and skill needed to fly a drone and at the same time make them as autonomous as possible. Other companies are aiming for commercial usage of drones in transport, monitoring, farming, and many other fields. Electronics in all drones has to have one common function – it has to be able to control the orientation and position of the drone in the air.

And this is the moment when the purpose of this project comes in, and that is to create a system which can stabilise the airplane and allow the user to control its position, using still relatively cheap and accessible components. Of course, this system would not be the first one of its kind. There are more of them and they are using multiple algorithms. During my internship at the company Danish Aviation Systems I prepared a fixed wing drone – flying wing eBee and equipped it with electronics necessary for this system. Honestly, I thought that making one myself will be easy. I was terribly wrong.

# Problem formulation and project scope

Making drones autopilot is a difficult and complex task and it cannot be accomplished by a single developer in a bearable amount of time, especially if the developer needs to first study the subject. Therefore, it was not possible to do all the work in the Final project‘s predefined period. It turned out to be important to define, what work is being done as a part of the Final project and what is excluded from it.

Main goals of this project are:

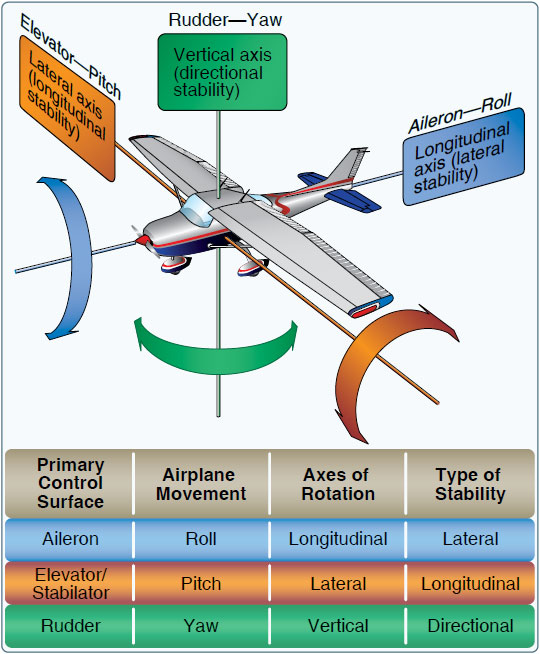
* Develop the logic hidden behind a control system, find out how the drone shall be controlled
* Write and application (single or multiple threads) controlling the drone in ChibiOS RTOS on STM32F745 microprocessor, including communication in between individual threads
* Test and debug the code in lab and on the field (in the air) on a fixed wing drone
* Make the control software open source and publish it on GitHub

Parts which are excluded from the final project:

* Building the drone, making and installing servos, making and installing a speed controller, preparing electronics with the microprocessor, gyroscope and accelerometer
* Setting up ChibiOS Kernel functions for STM32F745 (basically getting the clean ChibiOS running)
* Writing orientation sensor fusion (mixing data from gyroscopes, accelerometers, magnetometer and outputting the estimated orientation of the object in space)

Despite the fact that these parts were excluded from the project, it was necessary for me to understand them and learn how they works. Therefore they are to some extend an inevitable part of this report.

# The Project

  
Figure 1: Airplane's axis of movement

## The theory of airplane control

In order to design a control stabilisation system, it is important to understand how airplanes are controlled. Airplanes these days have many control surfaces . For a purpose of this project, we can take into account only main control surfaces which are:

* Ailerons (controlling Roll)
* Elevator (controlling Pitch)
* Rudder (controlling Yaw)

These surfaces allow us to control the airplane‘s rotation around 3 axes of rotation. As an example – increasing the angle of elevator by 5° can give us pitch angular velocity of 20° per second. In an ideal airplane, in still air at constant speed, the pirch angular velocity would be linearly depend on the elevators position. Unfortunately, this is not the case in real life. The airplane’s reactions to the input change

## ChibiOS RTOS

## Hardware build

## 

## Control algorithm

## Writing software

## Testing and debugging



# Conclusion

# List of references

<https://www.flightliteracy.com/wp-content/uploads/2017/11/6-4.jpg>

# Bibliography

Miroslav Lakota (1.6.2018) Fixed Wing Stabilization. Retrieved from <https://github.com/lakotamm/FixedWingStabilization> (visited 1.6.2018)

Erik Bærentsen (16.3.2017) Acquiring Accurate State Estimates For Use In Autonomous Flight. Retrieved from <https://discoverycenter.nbi.ku.dk/teaching/thesis_page/Erik-MasterThesis_FinalAsHandedIn.pdf/> (visited 31.5.2018)

Giovanni Di Sirio (2006 - 2017) ChibiOS documentation. Retrieved from <http://www.chibios.org/dokuwiki/doku.php> (visited 31.5. 2018)

iNavFlight. INAV - navigation capable flight controller. Retrieved from <https://github.com/iNavFlight/inav> (visited 31.5. 2018)

John W. Eaton (1998-2018). GNU Octave. Retrieved from <https://www.gnu.org/software/octave/> (visited 31.5. 2018)

Paweł Spychalski (2.2.2017). INAV 1.6: Fixed Wing PIFF Controller Retrieved from <https://quadmeup.com/inav-1-6-fixed-wing-piff-controller/> (visited 31.5. 2018)

Quanser Inc. (8.29.2017) Controller. Retrieved from <http://quanser-update.azurewebsites.net/quarc/documentation/controller_block.html> (visited 31.5. 2018)

flightliteracy.com (2018-2018) Flight Control Systems – Primary Flight Controls (Part One). Retrieved from <https://www.flightliteracy.com/flight-control-systems-primary-flight-controls-part-one/> (visited 1.6. 2018)

# Appendices