

SPRO4 PROBLEM FORMULATION

Group 7

March 2023

1 Background

Possessing the ability of flight and minimising effort and casualties has always been desirable for the utility flight can provide. The first unmanned aircrafts can be dated back to 1849, where Austria seemingly had utilised unmanned air balloons with stuffed explosives to attack Venice. [11] Ever since an unmanned aircraft vehicle (UAV), is one that is flown by technological means or as a pre-programmed flight without pilot control, as defined by the ECAA Transport Agency [4], nowadays called drones, have risen in popularity.

Because of this, UAVs come in a wide range of sizes and weights. UAVs often include multirotor, radio-controlled miniature helicopters, and aeroplanes [3]. As a result, there are several methods to categorize drones. The performance parameters of UAVs, such as weight, wingspan, wing load, flight range, maximum flying altitude, speed, and production cost, are typically used to categorize them [5]. According to how the lift is produced, drones may also be divided into fixed-wing and rotating-wing types. According to the drone code category, the European Aviation Safety Agency (EASA) categorizes unmanned aircraft by weight. The EASA regulations for open categories, or drones without an EASA class designation, are summarized succinctly and simply in Table I [2].

Table I Classification and restrictions for non-EASA class drones [2]

Self-built drones weighing up to 250 g, as described in Table 1, may be used without registration if the drone is a toy or the drone is not equipped with a camera, the remaining drones must be registered, and the pilot must pass examinations [2]. In this paper, self-built rotary drones with four wings or propellers are the objective, making weight-based classification suitable.

When it comes to the state-of-the-art project, PULP-DroNet is a deep learning-powered visual navigation engine that enables autonomous navigation of a pocket-size quadrotor in a previously unseen environment. Thanks to PULP-DroNet the nano-drone can explore the environment, avoiding collisions also with dynamic obstacles, in complete autonomy – no human operator, no ad-hoc external signals, and no remote laptop! This means

that all the complex computations are done directly aboard the vehicle and very fast. The visual navigation engine is composed of both a software and a hardware part. [7]

When it comes to the future, the simulated pollination of agricultural plants by means of nano copter can provide collecting and delivering pollen in the mode of automatic control. A design of nano copter for pollination can be made on the basis of innovative modification of existing model by its reprogramming with regard to its flight controller that is to be fully adapted to computer interface. The robotic system is offered specially for artificial pollination in conditions of greenhouses and minor agricultural enterprises. [1]

2 Problem statement

The utility of smaller drones are immense, where it can be used in surveillance, toys and potentially to also be part of a swarm of drones. Although, there are smaller drones existing in the current market, we would like to challenge ourselves to build one ourselves, where certain goals ranging from functionality to budget are listed below.

2.1 Primary goals

- Net maximum weight of the drone is 250 grams. Weight under 250 grams ensures it falls under A1 category in EU regulations. 1
- Flight time of 20 seconds.
- Stress of the structural system should not exceed rupture point. System does not experience fracture.

2.2 Secondary goals

- Flight time of minimum one minute.
- Can land with acceleration less than 9.8 m/s^2 .
- Stress of the drone system should not exceed the yield point. System does not experience plastic deformation.
- Drone is remote controllable.
- Drone can fly in formation with another identical drone.
- Total production cost of the drone is under 500 DKK (Not including remote controller).
- Drone can play audio.

2.3 Constraints

- Budget for entirety of project is 2000 DKK.
- Time available to finish the project is 4 months.
- Drone should have a minimum hover time of 5 seconds.
- Drone should be fully functional and able to take off again after landing.
- No use of flight controller software or unmanned vehicle Autopilot software Suite, capable of controlling autonomous vehicles.

3 Test Specifications

3.1 Primary goals:

- To test this, the drone will be weighed with a scale of a precision on 0,1 grams.
- In order to test the flight time, a stopwatch will be started from the moment the drone leaves the ground and is stopped as soon as it lands.
- This goal will be tested through FEM, ensuring that the chosen material for the drones body, will not rupture.

3.2 Secondary goals

- This will be tested with the same method as primary goals tests point 2.
- This will be tested with a mobile phone, recording the drones landing, using the drones position compared to the timestamp of the video.
- This will be tested with the same goals as primary goals test point 3.
- This will be tested by the possibility of sending wireless signals to the drone, with the drone reacting to those send signals.
- This will be tested purely by ear, listening to the drones output.
- This will be tested by mobile phone video, looking at the drones positions at given timestamps.
- This will be tested through summing the price for each single part, ensuring that it doesn't exceed 500 DKK.

3.3 Constraints

- This will be done with the same method as the secondary goals test, though ensuring the project cost is over 2000 DKK.
- To evaluate the time constraint point of the project, the goal fulfillments will be evaluated in the end of the project period. In the case that all primary goals are fulfilled, the constraint is succeeded.
- This will be tested with a stopwatch, ensuring that the hover time is atleast 5 seconds.
- This will be tested with making the drone take off right after a landing, making sure that the drone is fully operational at the second take-off.
- This will fulfilled by not employing any of the aforementioned in the drone.

4 Prototyping

4.1 Embedded development environment

Software should take care of the motor control, IMU output readings and remote control, this could create complications, as essentially they would interrupt each other. A way of multitasking should be introduced. "An RTOS (Real-Time Operating System) is a software component that lets you rapidly switch between different running sections of your code. Think of it as having several loop() functions in an Arduino sketch where they all run at the same time." [6]

As the Seeeduino XIAO nRF52840 Sense, uses the nRF52840 microcontroller, so looking to Nordic Semiconductor nRF Connect SDK for development, which uses a RTOS called ZephyrRTOS. Additionally, the SDK provides useful tools for development, such as build, flashing and debugging actions. [10]

Seeed XIAO nRF52840 is not natively supported on nRF Connect SDK, but as ZephyrRTOS supports it. [8]

There is a caviat around this, we can fetch profiles from GitHub of the current version of Zephyr via `Zephyr/boards/arm/xiao_ble` at main branch [9] and import the needed files into the SDK version of Zephyr we have. In our case the path of the files would be in `~\ncs\v2.3.0-rc1\zephyr\boards\arm`. After doing so, following the steps in the DevAcademy, nRF Connect SDK Fundamentals, Lesson 1 can be followed for setup. Then a blinky application can be created, and built via nRF Connect. After a build of the application is created, a files for flashing will be created in the "build" subdirectory of the application. To flash the Seeed XIAO nRF52840 Sense chip, entering a bootloader is needed, as it ships with the Adafruit nRF52

Bootloader, which supports UF2 flashing. Further more, a `zephyr.ut2` file can be found in the "build" subdirectory of the application, after building the application, as we have done. To access the bootloader, connect the Seeed XIAO to a PC. Now to enter and flash an application, the reset button on the Seeed board should be clicked two times in quick succession, this will prompt the memory of the Seeed board on your PC. Now find the previously mentioned `zephyr.ut2` and drag it in to the memory of the Seeed board, and this will flash the board with the new application.

An issue occurred when trying to test the "Hello World" application, when connecting to serial monitor no output is given. After further investigation, it was noticed that the Seeed documentation, mentions no debugging interface, hence no USB serial exists. To solve the issue, an application called "console", from `zephyr/samples/subsys/usb/console` was cloned, in order to test a virtual USB serial connection, with the use of CDC ACM UART. After building and flashing, results were achieved as seen in the picture. */*Insert picture of serial monitor*/* Another issue that arose was debugging. The Seeed XIAO nRF52840 Sense does not have any sort of built in debugging tools. One can use a J-link debugging tool, but a caveat was found, which was GDB stub, which Zephyr supports, this would save budget.

After achieving a successful development cycle, it was conducted, that the use of ZephyrRTOS is possible with our current setup. After further consultation with the supervisors, it was conducted, that the writers of the project have no skills in RTOS, more specifically in threading, and with the guidance of Davi, it was concluded, that acquiring such skills would be outside the scope of the project, as the main focus is control engineering.

References

- [1] Renat N. Abutalipov, Yuriy V. Bolgov, and Hamisha M. Senov. Flowering plants pollination robotic system for greenhouses by means of nano copter (drone aircraft). *2016 IEEE Conference on Quality Management, Transport and Information Security, Information Technologies, IT and MQ and IS 2016*, pages 7–9, 11 2016.
- [2] European Union Aviation Safety Agency. Open category - civil drones — easa, 2023.
- [3] Cavoukian Ann. Privacy and drones: Unmanned aerial vehicles. 2012.
- [4] Trafikstyrelsen Droner. Flying drones in denmark, 2022.
- [5] M. Hassanalilian and A. Abdelkefi. Classifications, applications, and design challenges of drones: A review. *Progress in Aerospace Sciences*, 91:99–131, 5 2017.

- [6] Joe Jaworski. A real-time operating system for the arduino — nuts and volts magazine, 2019.
- [7] Vlad Niculescu, Lorenzo Lamberti, Francesco Conti, Luca Benini, and Daniele Palossi. Improving autonomous nano-drones performance via automated end-to-end optimization and deployment of dnns. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 11:548–562, 12 2021.
- [8] Zephyr Project. Introduction — zephyr project documentation, 2022.
- [9] Zephyr Project. Zephyr project github repository, 2023.
- [10] Nordic Semiconductor. nrf connect for vs code - nrf connect for vs code, 2022.
- [11] Kashyap Vyas. A brief history of drones: The remote controlled unmanned aerial vehicles (uavs), 2020.