graphicbackgroundwhite codebackgroundwhite codeonlybackgroundwhite numbackgroundwhite textcodecolorblack numcolorgray

/tkzexampleexecute code/.is if = code@execute, show num/.is if = num@show, overhang/.is if = code@overhang, code only/.code = , width/.code = 1!, code/.code = codebackground1, graphic/.code = graphicbackground1, overhang/.code = , num/.code = , numcolor/.code = numcolor1, numbkgcolor/.code = numbackground1, global num/.code = , code style/.code = , normal/.code = , small/.code = , very small/.code = , right margin/.code = , left margin/.code = , inner left margin/.code = , inner right margin/.code = , inner top margin/.code = , inner bottom margin/.code = , line frame width/.code = , frame code/.code = , frame tex/.code = , above skip/.code = , below skip/.code = , hsep/.code = 1!, pre/.code = , post/.code = , latex/.code = 1!, vbox/.code = , every tkzexample/.style=width=6cm,

/tkzltxexamplewidth/.code = 1!, code/.code = codeonlybackground1, show num/.is if = num@show, num/.code = , global num/.code = , numcolor/.code = numcolor1, numbkgcolor/.code = numbackground1, normal/.code = , small/.code = , very small/.code = , right margin/.code = , left margin/.code = , inner left margin/.code = , inner right margin/.code = , inner top margin/.code = , inner bottom margin/.code = , line frame width/.code = , hsep/.code = , typeset listing/.code = 1, frame code/.code = , ignorespaces/.code = , leave comments/.code = , every tkzltxexample/.style = width=8cm, vbox/.code =

/tkzsavedcode/.cd, prefix/.code =

UNIVERSITY OF SOUTHERN DENMARK

Faculty of engineering

BEng Mechatronics

Mechatronics Semester Project 4

Ladybug

Supervisor:
Lars Duggen
Davi Goncalves Accioli

Group: 7
Tobias Blaabjerg Karlsen
Artis Fils
Thor Uerkvitz
Choyon Mainul Hasan
Johan Paul
Ruta Miglava

Year 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. [?] 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 [?], 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 [?]. 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 [?]. 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 Figure ?? [?].

Self-built drones weighing up to 250 g, as described in Figure ??, 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 [?]. In this paper, self-built rotary drones with four wings or propellers are the objective, making weight-based classification suitable.

UAS	Operation		Drone operator/pilot		
Max weight	**Subcategory**	**Operational restrictions**	**Drone operator registration**	**Remote pilot competence**	**Remote pilot minimum age**
< 250 g	A1 (can also fly in subcategory A3)	— No flight expected over uninvolved people (if it happens, overflight should be minimised) — No flight over assemblies of people	No, unless camera / sensor on board **and** the drone is not a toy	— No training required	No minimum age
< 500 g			Yes	— Read carefully the user manual — Complete the training and pass the exam defined by your national competent authority or have a 'Proof of completion for online training' for A1/A3 'open' subcategory	16*

Figure 1: Classification and restrictions for non-EASA class drones [?]

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. [?]

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. [?]

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/s2.
- 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 the tested trough 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 trough 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 fullfilments 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 Design and manufacturing of quadrotor

5 Software prototyping

5.1 Embedded development environment

Software shoud 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." [?]

After research, the list was narrowed to two top condenders – FreeRTOS and Zephyr. Both solutions are open source, widely used and support the Microcontroller board we have chosen. [?] According to 2018 IoT Developer Survey [?], FreeRTOS is one of the most popular OS used and while Zephyr only received a 2.8 % rating in 2018, it is often described as one of the fastest growing RTOS and in 2022 has become the largest open-source RTOS project by the number of commits and developers. (See Figure ??)

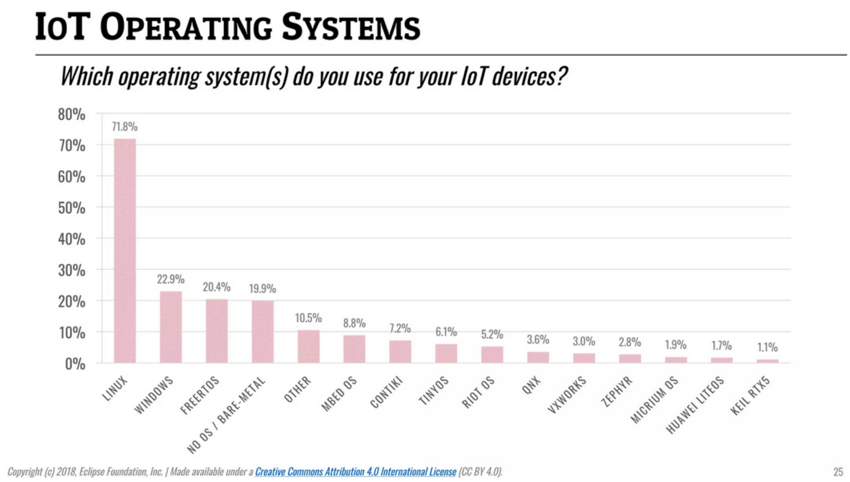

Figure 2: IoT Developer Survey 2018 results

To decide between the RTOS choice, a pros and cons table was created and evaluated. [?] (See Table ??)

RTOS	Advantages	Disadvantages
FreeRTOS	• Suitable for begginers • Open source – online comunity support available • Libraries for Seeed XIAO Sense available • Constantly improved • Fast code execution • Low memory consuption	• Not event-driven (scheduler will only be called once in a certain period of time) • Less flexible
ZephyrRTOS	• Open source – online comunity support available • Libraries for Seeed XIAO Sense available • Constantly improved • Designed to ensure energy efficiency • Highly configurable • Event-driven • Kernel can create additional system threads • Possible to exclude multi-threading • Additional debugging features • Supported by Nordic Semiconductor	• More difficult to set-up • Potentially complicated to use with no experience

Table 1: Comparison between ZephyrRTOS and FreeRTOS

It was evaluated, that overall FreeRTOS seems to be a better established more simple solution to be used in case of no experience, whereas Zephyr offers more flexibility and is a fast growing popular solution well suited for our application and would be a worthwile investment to build our skillset for future projects. [?]

PlatformIO is an open-source extension of Visual studio code, which supports hundreds of boards with different frameworks. [?] Seeed XIAO nRF52840 is not natively suported. Seeed made library for Arduino IDE, which supports Arduino and Mbed framework, that was implemented through Github to PlatformIO and allows non-problematic implementation. The XIAO has library in the Zephyr Github repository, but the PlatformIO does not use most recent version. The version of Zephyr itself is altered in PlatformIO and therefore does not work even after corssreferencing with other boards already implemented in PlatformIO, such as Seeeduino (as a reference for the formfactor) or nRF52840 DK (as a reference for the chip). The PlatformIO does not have inbuild serial USB communication, which results in inability to reprogram the board through the VS Code and need to go through bootloader first. This poses inconvinienice, which can be avoided by use of a Arduino IDE, therefore if route of MBed or Arduino framework is chosen, Arduino IDE should be chosen. If the route of Zephyr is chosen, SDK by nordic semi is the optimal route, because event though it also does not have serial USB, it uses barebone Zephyr.

Opon further research, the Seeeduino XIAO nRF52840 Sense uses the nRF52840 microcontroler, 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. [?]

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

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 [?] 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 propmt 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.

A issue accured 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, mentiones 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. (See Figure ??)

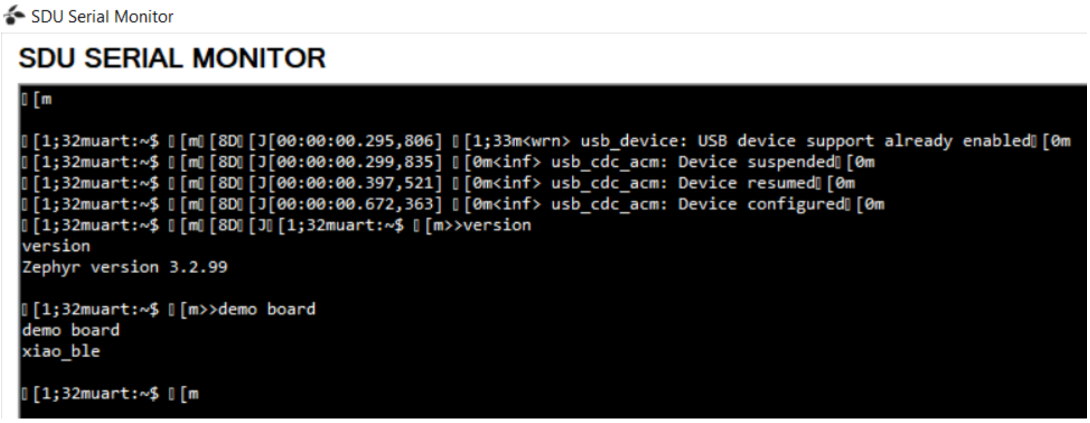

Figure 3: Serial output of the XIAO MCU

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 caviet was found, which was GDB stub, which Zephyr supports, this would save budget. [?]

After achieving a succesful 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 guidence of Davi, it was concluded, that acquiring such skills would be outside the scope of the project, as the main focus is control engineering.

5.2 QuickPID

For implementing the PID controller on the MCU, the QuickPID library was chosen. QuickPID is an updated implementation of the Arduino PID library with additional features for PID control. [?] It was chosen due to the following reasons:

- Advanced Anti-Windup Mode: When compared to the normal PID implementation, the library features a more robust anti-windup mode. Integral windup can occur when the controller's integral term accumulates error during periods of saturation or when the actuator cannot keep up with the needed control effort, and anti-windup approaches help prevent this. [?] The QuickPID library can lessen integral windup, reduce overshoot, and improve stability during transient situations by adding an advanced anti-windup mode.

- Timer-Based Control: QuickPID is a timer-based control option that lets you use external timers or Interrupt Service Routines (ISRs) for precise timing control. [?] This functionality is especially beneficial in systems with strict timing requirements or when employing external devices or sensors that are synced with the control loop. In this situation, if issues with the IMU or the controller develop, the timer-based technique can simply remedy them.

- Compatibility with Arduino IDE: updated implementation of the Arduino PID library. As the Arduino ecosystem was chosen over Zephyr RTOS using the QuickPID library provides a seamless integration.

When utilizing the library, the PID compute sample time, default = 100000 µs, was changed to 2500 µs to match the IMU 400 Hz measuring frequency. Matching the PID sample frequency with the IMU measuring frequency ensures synchronization, consistent data availability, and accurate control in quadrotor systems. It enables the PID controller to operate based on up-to-date sensor measurements, respond effectively to dynamic changes, simplify system identification and tuning, and optimize computational efficiency. Additionally, the output limits of the PID (0-255 by default), were changed to (-255) – (255), as a negative error can be achieved in the quadrotor system.

5.3 NanoBLEFlashPrefs

Data logging has evolved as a significant approach for overcoming the issues of the PID tuning process. [?] During quadrotor operation, data logging entails recording numerous flying characteristics, sensor measurements, control inputs, and system responses. Engineers and researchers can obtain useful insights into the quadrotor's behavior and make informed judgments to fine-tune PID controller parameters by collecting this comprehensive set of data. [?] Data logging also allows for the comparison and evaluation of various PID tuning strategies or algorithms. Thru analysis, the influence of each adjustment on the system's behavior by methodically changing PID controller parameters can be observed in the quadrotor's response. This iterative method makes it easier to identify ideal PID parameter combinations that produce the required flight characteristics. The Seeduino XIAO nRF52840 lacks an inbuilt EEPROM (Electrically Erasable Programmable Read-Only Memory). The nRF52840 SoC has 1 MB of inbuilt Flash memory, and the XIAO board has 2MB of onboard flash memory for storing data. The choice was based on the following benefits of adopting flash memory:

- Faster Read and Write Operations: Flash memory generally offers faster read and write speeds compared to EEPROM. This will be advantageous in the iteration process when there is a need to access data quickly or update it frequently.
- Endurance and Lifetime: Flash memory typically has a higher endurance level than EEPROM. It can withstand a larger number of read and write cycles before it starts to degrade. This endurance is particularly important when you need to update or modify the contents of the memory frequently.
- Cost and Space Efficiency: Since the nRF52840 chip already integrates Flash memory, using it eliminates the need for an additional EEPROM chip. This reduces the component count, simplifies the PCB layout, and potentially lowers the overall cost and space requirements of your design.

NanoBLEFlashPrefs is a Arduino library, which is a substitute for missing EEPROM storage on Arduino Nano 33 BLE and 33 BLE Sense (not for Nano 33 IoT or other Nano boards). [?] As the Seeduino XIAO nRF52840 Sense uses the same SoC chip as the Arduino Nano 33 BLE Sense, it can be easily used for the project. [?] [?]

[] [10pt]article xcolor pgf,tikz tkzexample typedef struct flashStruct int time[DURATION / LOG$_P ERIOD]; signedcharphi[DURATION/LOG_P ERIOD]; signedchartheta[DURATION/LOG_P E$

int rc = myFlashPrefs.readPrefs(prefs, sizeof(prefs));

// Prepare preference record for writing prefs.time = millis(); prefs.someNumber = Complimentary[0]; prefs.anotherNumber = Complimentary[1];

// Write it to flash memory myFlashPrefs.writePrefs(prefs, sizeof(prefs));

References

[1] Arduino® nano 33 ble sense.

[2] Dirk-/nanobleflashprefs: Substitute for the missing eeprom storage on arduino nano 33 ble and ble sense.

[3] 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.

[4] European Union Aviation Safety Agency. Open category - civil drones | easa, 2023. `https://www.easa.europa.eu/en/domains/civil-drones/drones-regulatory-framework-background/open-category-civil-drones`.

[5] Cavoukian Ann. Privacy and drones: Unmanned aerial vehicles, 2012. `http://www.ipc.on.ca/English/Resources/Discussion-Papers/`.

[6] Benjamin Cabé. Iot developer survey 2018, 2018. `https://www.slideshare.net/kartben/iot-developer-survey-2018`.

[7] Dlloydev. Dlloydev/quickpid: A fast pid controller with multiple options. various integral anti-windup, proportional, derivative and timer control modes.

[8] Trafikstyrelsen Droner. Flying drones in denmark, 2022. `https://www.droneregler.dk/english`.

[9] M. Hassanalian and A. Abdelkefi. Classifications, applications, and design challenges of drones: A review. *Progress in Aerospace Sciences*, 91:99–131, 5 2017.

[10] Connected Industry. Freertos vs threadx vs zephyr: The fight for true open source rtos, 2023. `https://www.iiot-world.com/industrial-iot/connected-industry/freertos-vs-threadx-vs-zephyr-the-fight-for-true-open-source-rtos/`.

[11] Joe Jaworski. A real-time operating system for the arduino - nuts and volts magazine. *Copyright © 2023 T and L Publications*, 2019. `https://www.nutsvolts.com/magazine/article/a-real-time-operating-system-for-the-arduino`.

[12] PlatformIO Labs. A professional collaborative platform for embedded development · platformio. `https://platformio.org/`.

[13] micro ROS. Comparison between rtoses. `https://micro.ros.org/docs/concepts/rtos/comparison/`.

[14] Dennis Nash. How to access data for optimal controller tuning | rockwell automation.

[15] 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.

[16] Zephyr Project. Introduction — zephyr project documentation, 2022. https://docs.zephyrproject.org/latest/introduction/index.html.

[17] Zephyr Project. Zephyr project github repository, 2023. https://github.com/zephyrproject-rtos/zephyr/.

[18] Nordic Semiconductor. nrf connect for vs code - nrf connect for vs code, 2022. https://nrfconnect.github.io//vscode-nrf-connect/index.html.

[19] Seeed Studio. Getting started with seeed studio xiao nrf52840 (sense) | seeed studio wiki, 2023. https://wiki.seeedstudio.com/XIAO_BLE/.

[20] Jasper J. van Beers and Coen C. de Visser. Peaking into the black-box: Prediction intervals give insight into data-driven quadrotor model reliability. 1 2023.

[21] Kashyap Vyas. A brief history of drones: The remote controlled unmanned aerial vehicles (uavs), 2020. https://interestingengineering.com/innovation/a-brief-history-of-drones-the-remote-controlled-unmanned-aerial-vehicles-uavs.

[22] Oleksandr Zozulia. Freertos vs. zephyr project for embedded iot projects - lemberg solutions, 2022. https://lembergsolutions.com/blog/freertos-vs-zephyr-project-embedded-iot-projects.