Open Drone - FLA400 - Project in Dependable System

E. Beckman, R. Hamrén, J. Harborn, L. Harenius, D. Nordvall, A. Rad and D. Ramic

Abstract - The purpose of the project is to create an autonomous Unmanned Aerial Vehicle that could gather data mid-air from different ground stations depending on the mission. The UAV is ordered by AFarCloud and created by final year students at Mälardalen University from the Master of Science in Engineering - Dependable Systems program in the course FLA400. Autonomous UAVs are used more and more to gather data from the air, supporting other systems with data such as images, video, and sensor data. This can then later be used to complete certain tasks. When developing a UAV, dependability and safety become high focus since there is no operator. One simple way to increase the safety was to separate the flight controller from the data gathering unit. Data gathering and the flight controller is implemented as a separate system to make it easier to develop. Another requirement was to create an open source system, this results in that the UAV is totally modifiable to any task as long as the hardware and software is compatible with the specific mission task, some missions requires a camera some does not. The goal of the project was to gather data from a ground-sensor through Bluetooth. This was made possible by an onboard Odroid on the UAV, which could collect data via the support of Bluetooth. Each task was created after specific requirements, that was designed following ARP4761/ARP4754A. After the task requirements were created and fulfilled, validation and verification methods were performed to see if all requirements were fulfilled and to see that errors would not occur again.

1ST ERIK BECKMAN, EBN15013, EBN15013@STUDENT.MDH.SE
2ND RASMUS HAMRÉN, RHN15001, RHN15001@STUDENT.MDH.SE
3RD JAKOB HARBORN, JHN15009, JHN15009@STUDENT.MDH.SE
4TH LINUS HARENIUS, LHS15001, LHS15001@STUDENT.MDH.SE
5TH DANIEL NORDVALL, DNL15001, DNL15001@STUDENT.MDH.SE
6TH ARMIN RAD, MRD15002, MRD15002@STUDENT.MDH.SE
7TH DINO RAMIC, DRC15001, DRC15001@STUDENT.MDH.SE

IDT. MÄLARDALEN UNIVERSITY, DEPENDABLE SYSTEMS, VÄSTERÅS, SWEDEN

Table of Content

I. Introduction 5
II. Glossary 5
III. Background 5
A. Hardware 5
1) Design Decisions Hardware 5
B. Software 6
1) Design Decisions Software 6
C. Planning 6
1) Initial Development plan 6
2) Model AlphaX 7
3) Construct Hardware 7
4) Software Setup 7
5) V&V 8
D. Quality and process assurance 8
1) Process assurance Strategy 8
2) Main goals for each drone version 8
3) Deviations 8
4) Quality Assurance 8
5) Problem report 9
E. Design Decisions 9
1) Alpha0.5 9
2) Alpha1 9
3) Alpha2 9
4) Alpha3 10
5) Beta V1, 2, 3 11
F. Project administration 11
IV. Reliability work 11
A. Safety 11
B. Validation & Verification 12
1) Validation 12
2) Verification 13
V. The UAV 13
A. Software 13
1) QGroundControl 13
2) Python 14
3) Bosch XDK 14
4) Odroid XU4 14
5) MAVLink 14
6) DroneKit 14
7) Apache Thrift 14

8) JMAVSim 14
B. Hardware Electronics 15
1) Odroid-XU4 15
2) Motors 15
3) Pixhawk 4 15
GPS 15
Barometer 15
Accelerometer and Gyroscope 15
Magnetometer 15
Radio antennas 16
PMB 16
4) ESC 16
5) PPM encoder 16
6) Propellers 16
7) Battery 16
8) BOSCH-sensor 16
9) Lidar 17
C. Hardware Structures 17
1) CAD 17
2) 3D Printing 17
3) Materials 17
Polylactic acid (3D printed) 17
Nylon Carbon fiber filament (3D printed) 17
Carbon fiber 400HV 17
4) Materials Strain- & pressure test 17
V. Conclusion 19
A. Some common mistakes 19
B. Further Development Discussion 19
C. Materials Strain- & pressure test 19
VI. References 20
VII. Appendices 21
A. Rad, A. (2019). Process Assurance. 1st ed. Västerås: OpenDrone Project, p.9-13. 21
B. Rad, A. (2019). Process Assurance. 1st ed. Västerås: OpenDrone Project, p.14-16. 26
C. Harborn, J. and Nordvall, D. (2019). Requirements. 1st ed. Västerås: OpenDrone Project. 29
D. Harborn, J. (2019). Functional Hazard Assessment. 1st ed. Västerås: OpenDrone Project. 43
E. Harborn, J. (2019). Preliminary System Safety Assessment. 1st ed. Västerås: OpenDrone Project. 43
F. Harborn, J. (2019). System Safety Assessment. 1st ed. Västerås: OpenDrone Project. 56
G. Documents Description. (2019). 1st ed. Västerås: OpenDrone Project. 56
H. Nordvall, D. (2019). Validation 1st ed. Västerås: OpenDrone Project, p.1-3. 61
I. Nordvall, D. Verification Plan 1st ed. Västerås: OpenDrone Project, p.1-2. 66
J. Nordvall, D. Verification Summary 1st ed. Västerås: OpenDrone Project. 71
K. Nordvall, D and Harenius, L. Test Plan and Test Protocol 1st ed. Västerås: OpenDrone Project. 72

L.	Nordvall, D. DVM. 1st ed. Västerås: OpenDrone Project.	90
M.	Ramic, D. (2019), Flowcharts 1st ed. Västerås: OpenDrone Project, Data Collection & Server Flow charts	96
N.	(2019), AT2820-motor-specification, OpenDrone Project.	98
O.	Harenius, L. Figures 1st ed. Västerås: OpenDrone Project. Enlarged architecture figures.	100

I. INTRODUCTION

The main purpose of this project is to construct a fault tolerant open source drone, the Open Drone, for tasks such as detecting wildfires, tracking livestock, or collecting other data. The project will gather data with a sensor provided by the customer as a first task.

The aim is to compare the drone to a DJI Mavic Pro and to make the projects drone open source, and therefore create an open software for development and improvement in the future. The Mission Management Tool that will control mission specifics has already been developed by a previous project group, this tool will be used in this project, and further used by the customer AFarCloud.

The drone will be optimized with developed hardware and material to get maximum testing and developing during the project and by this analyze, lower the weight, and strengthen the chassis. Material gets tested at MDH Eskilstuna Sweden.

The focus of the project is to make an open system for future development for specific tasks from the customer AFarCloud and to make a moldable drone for other missions. Software such as mission planner and path planners will be developed and tested.

Requirements and safety-assessment will be done in the project and will be developed according to aerospace industry regulations as ARP4761/ARP4754A.

The project is at master thesis level and will be carried out by fifth year students at the Dependable Systems Engineering program at Mälardalen University.

II. GLOSSARY

MDH - Mälardalen University
QGC - QGroundControl
MMT - Mission Management Tool
UAV - Unmanned Aerial Vehicle
Drone - Unmanned Vehicle
Quadcopter - 4-armed UAV
Hexacopter - 6-armed UAV
Octocopter - 8-armed UAV
FLCC - Flight Control Computer
PMB- Power Management Board
BLE - Bluetooth Low Energy
LIDAR - Light Detection and Ranging
DAL - Development Assurance Level
FHA - Functional Hazard Assessment
PSSA - Preliminary System Safety Assessment
SSA - System Safety Assessment
PX4 – Pixhawk 4
IMU – Inertial Measurement Unit

III. BACKGROUND

A. *Hardware*

When it comes to the hardware part of the project the group has the ambitions of challenging one of the biggest competitors on the market, the DJI Mavic Pro. All hardware should at least be as good as the specifications of the DJI, maybe not the Alpha prototype drone but the goal of the project is to challenge the rival and get longer airtime in one charge. Some specifications will be harder to compare, such as mission accuracy, but the project group is aware of this.

For the project, an open source platform, is of most importance, so development and further future modifications of the drone is possible. The group chose an open source platform with the Pixhawk 4 (PX4) as Flight Control Computer (FLCC) due to its lightweight, size, modifiable aspects, and hardware specifications. The PX4 has a wide range of different sensors that could be easily installed allowing the user to customize the drone for whatever needs they have. most sensors on the market are compliant with the PX4. The group is planning to build three drones in total, Alpha1, Alpha2 and the final version of the drone. The Alpha1 drone will have a Pixhawk but not necessarily all the final sensors, motor or even the final range capabilities. The strategy of the Alpha1 drone is to just get it flying and then on Alpha2 start working with safety-critical sensors and make it autonomous. The group feels that alpha testing the flight capabilities of the UAV with expensive equipment is an unnecessary risk to take. In between Alpha1 and the final version of the drone, its skeleton shall be built out of carbon fiber.

1) *Design Decisions Hardware*

The PX4 was chosen as the FLCC because of its characteristics. It is built with redundancy in mind, while developed on an open source platform, which is exactly what the project is all about – an Open (source) Drone.

Open source is very important since drones, available for purchase today, are not easily modifiable. For example, if the user would want to change the behavior of the drone, it would have to buy another type of drone. With the PX4, the user could just mount it on another frame, re-calibrate it, and fly with different configurations and components.

The most important aspect about the redundancy of the PX4 is that it is developed with considerations to safety and reliability. It has redundant Inertial Measurement Units (IMU), which ensure that the probability of, for example, gyroscope failure, is decreased, since it has two gyroscopes.

A schematic overview of the hardware can be seen in Figure 1. An enlarged version can be found in Appendix 0. The arrows show how the different peripherals and sensors are planned to connect in a system-like environment to the PX4.

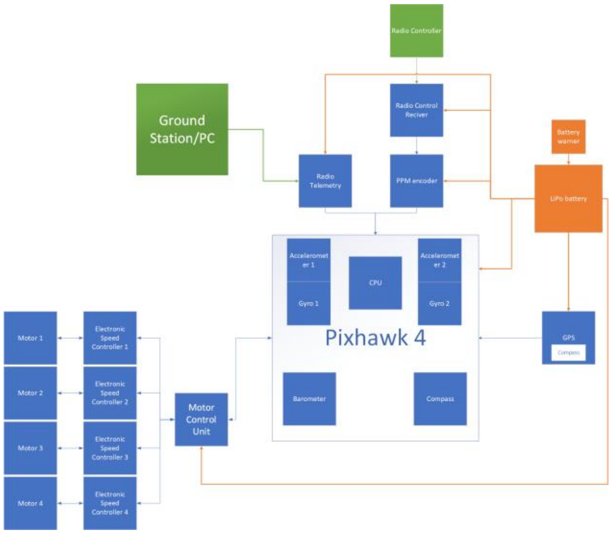


Figure 1 Initial hardware plan

B. Software

The project is called Open Drone as the purpose of the project is to build a platform where every parameter is changeable, unlike closed solutions on the market today. This means that researchers and future students can easily access the software and change different parameters to make it fit for their specific design and application. Thus, the goal is to make the drone platform easily accessible and understandable for people, so it can be further developed in future projects.

The software for the project will be developed through different applications and tools. The Px4 is an open platform flight controller, developed for drones and similar applications. The developer can simulate it through jMAVSim and QGC. In those applications, the developer can see how the drone, and consequently the PX4, will behave. Every parameter can also be tested through simulations, including sensor value outputs and flight modes.

For the first two prototypes control of the drone is performed through a radio controller or QGC. QGC is used to configure the PX4, set up missions, an interface to show sensor outputs and show its location.

Since QGC can directly communicate with jMAVSim, it is a good starting point for developing an autonomous drone.

The Mission Management Tool (MMT) will be developed/integrated separately during the project. The software for the MMT is already developed, with a working interface. The plan is to integrate it with the Open Drone during the project. Thrift will be used to translate the communication between the MMT and the PX4. This will be done if the group has the required time, but the goal is to make the MMT to work with the developed drone.

Throughout the project, the named platforms and different coding languages must be further investigated. This means that a considerate time allocated for the SW team, will be reserved for reading about and testing of languages/platforms.

A high-level overview of the different software components and their interaction is shown in Figure 2.

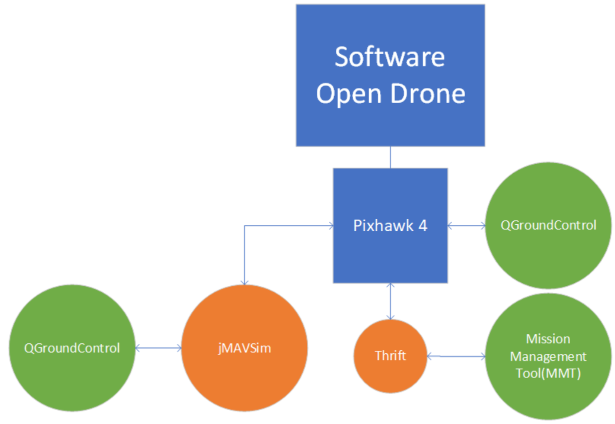


Figure 2 High-level software interaction

1) Design Decisions Software

In the previous section, QGC and jMAVSim were mentioned. Since the PX4 is used, both of those interfaces/tools will make it easier to develop a fully functional drone. An idea of a different configuration may pop up, then it can easily be tested in a simulated environment through jMAVSim. If the simulation was successful, it could then be transferred to the physical drone and then tested in a real environment. This will ensure that different configurations are at least tested in a simulated environment before applied to the real drone, which is both safe, and can save a lot of time and resources.

There are two alternatives when it comes to ground station/mission planner interface, Ardupilot and QGC.

QGC was chosen because of its more user-friendly interface. QGC is also much clearer when it comes to its settings, setup, and parameter interface.

C. Planning

1) Initial Development plan

Project Main Requirements and ambitions:

A quadcopter will be designed based on the customer requirements. To meet these requirements, in this project, the UAV should be able to:

1. Fly autonomously through the communication with the MMT.
2. Send data in real-time to the MMT if the bandwidth requirements are met.
3. Allow the human operator to, partially or fully, take control of its functions.
4. Allow collaboration with other UAV and other autonomous systems, such as Unmanned Ground Vehicles (UGV).

5. Demonstrate certain degree of dependability in line with the requirements from the application domain and experimental settings.
6. Long flight time for longer missions.
7. Controlling drone/drones using the MMT.
8. Collecting data via Bluetooth from placed sensors.
9. Long range control of the drone.
10. Stable control of the drone (smooth landing, hovering, tolerate winds, etc.).

The ambition is to use:

1. An open flight controller and/or computing unit onboard the drone.
2. Allow several UAVs to communicate with the MMT.
3. Increase the collaboration between the human operator and the fleet of UAVs.

Not all requirements will be met due to the limited time. Requirements not fulfilled will be possible areas of improvements for further development for the next generation of students.

The frame will be designed using Solidworks following an iterative process, where a new version will have improvements over the prior. Modeling the frame for the drone instead of buying an off-the-shelf frame has its benefits, such as customization around components, i.e. batteries, sensors, etc., to ensure perfect fit.

Initially four models were planned, where drone version Alpha3 was going to be the final version, as shown in Figure 3. Deviations during the development process led to a change, where an additional version was added, Beta. Each version will be tested to see if it meets the requirements. If requirements are not met, improvements must be made.

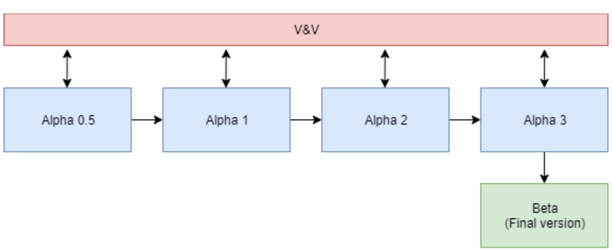


Figure 3 Prototype development process

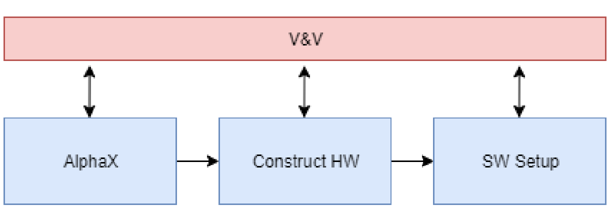


Figure 4 Development process of each Alpha prototype

2) Model AlphaX

The development process for the different Alpha versions of the drone can be seen in Figure 4. The first version, Alpha0.5, will be modeled using Solidworks and thereafter 3D printed in plastic. Its main purpose is to test the FLCC with motors, to get a basic understanding of its functionality. Alpha 0.5 is hence, a small and basic frame, designed to fit only the necessary components, i.e. PX4, motors, Power Management Board (PMB) and batteries, and flight will probably not be possible. This version also serves to test the 3D printing process and identify potential problems.

The Alpha1 will be a larger and more suitable design for flight. Alpha1 will be 3D printed in plastic.

When the first frame is used, different design flaws are discovered. With Alpha2 improvements such as the size of the drone, shape changes for more room for more components etc., are to be implemented to get a completer and more flyable drone. Alpha2 will be 3D printed in plastic.

Alpha3 is planned to be the final alpha model where the shape and size are perfectly formed to fit all the components. Sensors are also to be easily attached/removed. In case some sensors are not needed for a mission, they can be removed to save weight. The frame will be made of carbon fiber to reduce the weight as much as possible.

Deviations lead to a Beta version to be added, which will be the last and final version were all the requirements within the scope of this project are to be met. The Alpha3 goals are moved over and included in the Beta version.

3) Construct Hardware

Alpha0.5 will test the FLCC/PX4 with four non-specific motors.

Alpha1 features the PX4, a radio telemetry module for manual/autonomous control of the drone, four motors and a battery. Optimal motors and batteries are not important for this version.

For Alpha2, an Arduino board, or microcontroller, will be added to the drone to receive data from external Bosch sensors, placed at known locations within a test area, via Wi-Fi or Bluetooth. Receiving data from the Bosch sensor is the primary task to complete. Other optional sensors such as lidar or IR-sensors can be added if time allows. Using these sensors for obstacle detection is also optional since it is not a requirement from the customer.

For Alpha3 the non-specific batteries and motors will be replaced with new ones, suitable for optimal flight time. The frame shall also be made of carbon fiber to minimize weight.

4) Software Setup

Throughout the drone versions QGC will be used, to learn and test the software. QGC is compatible for PX4 and very easy to use. QGC is used to easily tune PID parameters and mission control, e.g. set autonomous flight controls through waypoints, auto landing and hovering. Documentation of QGC can be found on their homepage [2].

When developing Alpha2 and Alpha3, QGC will be used but for later version replaced by a MMT developed within a MDH research project. The communication between the MMT and PX4 is based on Thrift and to handle this a Python-based application will be developed. The same software will be used for the final drone version.

The MMT will let the user plan missions by selecting a flight area and setting waypoints. The MMT will then select suitable drone/drones for the given mission. In the future a large swarm of drones will be controlled by the MMT.

5) V&V

Throughout the process, the product and its components will be validated to verify that they are developed based on the requirements. Tests, analysis, and simulations are carried out to verify that the product has been correctly developed and that the requirements have been met.

D. Quality and process assurance

1) Process assurance Strategy

To assure that the plan has been followed and that requirements and goals have been met, a table is created to check if the plan has been followed for each drone version created and possible deviations from the development plan, see Appendix A. A description of how the version is modelled, HW development and SW development is described. Lastly the development is being validated and verified to check if the correct product is being developed according to the requirement goals that has been stated for each alpha version. Based on the results from the V&V, improvements must be made for the next drone version so that the requirement goals for the next version can be met.

2) Main goals for each drone version

Alpha0.5 goals

- Get to know PX4.
- Control motors using RC.
- Control motors using QGC.

Alpha1 goals

- Basic manual flying using RC.
- Basic autonomous flying using QGC.

Alpha2 goals

- Controlled manual and autonomous flying using QGC.
- Mission planning using MMT combined with Thrift.
- Receive data from Bosch sensor.

Alpha3 goals

- Maximize flight time (reducing weight, increasing battery capacity, choosing optimal motors).
- Long range connection.
- Meet most of the requirements within the scope of this project.
- Finished product (initially decided)

Beta goals (not initially planned)

- Finished product
- Reduced weight (carbon fiber frame)
- Meet most of the requirements within the scope of this project.

3) Deviations

All deviations that occurred during the development process has been tracked and corrective actions have been taken. Almost all versions had some type of deviation from the development process.

For the first version, Alpha0.5, the frame was too small and no components for wireless connection had been purchased. The flight controller could still be connected to the motors for testing and to a computer using a USB cable. No wireless control using RC or QGC could be performed due to the lack of transmitter/receiver. Transmitter/receiver module was ordered to be used for the next version.

The development of the next frame model was delayed so Alpha1 was never printed, instead an old frame made of carbon fiber with two floors was used as a replacement.

The Alpha2 model consisting of two levels was still not large enough, an extra floor had to be added for more room for components. The Alpha2 model was therefore scrapped and model development of Alpha3 was started. A decision was made that Alpha3 was to meet the requirements of both Alpha 2 and 3.

The frame for the Beta version was constructed but not used due to limited time. The decision was made to stop development at Alpha3 since further development would not be continued on the Beta version, as this would most likely require a new hexacopter frame to be modeled. Furthermore, the only difference between Alpha3 and Beta is the frame material (weight).

4) Quality Assurance

To ensure that the quality standards are high, requirements are defined and set to a standard compared to similar products, such as DJI Mavic Pro. All requirements have been assigned either a test case or on-site test which will serve as evidence to prove that the requirements are met. There are several ways of performing these tests depending on what type of requirement that needs testing, generally three methods are used: physical tests, analysis, and modelling. These three methods will be applied throughout the testing process to provide sufficient evidence that the requirements are fulfilled and that the system is dependable.

Every requirement will have to be evaluated independently using an appropriate method. Primary responsibility that these tests are performed is placed on the Verification & Validation team lead. However, the tests can be performed by any available member. Generally, the team states the requirement to be tested, followed by relevant circumstantial information about the tests such temperature, UAV setup and other parameters that could be important to the case.

Testing will be separated into two phases: Alpha Testing & Acceptance Testing. The Alpha Testing will be performed to ensure that the product meets the expected functionality, in every stage of the development. The Acceptance Testing will be performed to provide evidence that the UAV satisfies the requirements set by the project.

5) Problem report

A problem report has been created and filled out whenever problems occurred during the development process. Problems have been discussed and preventive solutions have been made. Future students, working on this project, can read this report and learn from the mistakes that have been made. The report is available in Appendix B.

E. Design Decisions

1) Alpha0.5

Alpha0.5 was thought to be the introductory concept, where getting to know the basics of QGC and PX4 was the main goal. The frame of Alpha0.5 was 3D printed without much thought put into the design. Both because it was thought to be "the introductory concept", but also because of the lack of knowledge of the amount of cabling needed and the size of the components. Alpha0.5 can be seen in Figure 5. It can be noted that there was not enough space for putting the battery, PMB and the PX4, and a new design was sketched on.

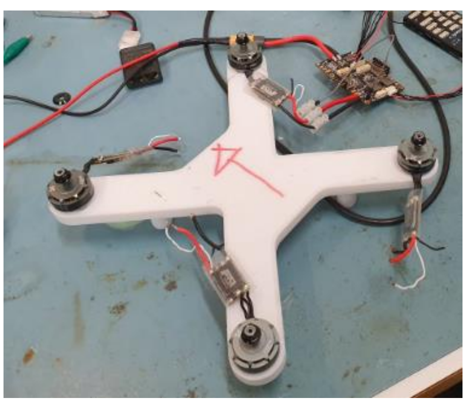


Figure 5 Alpha0.5

2) Alpha1

The "first" version of the drone was used to test the system architecture and design, while integrating the PX4 with the other peripherals. This version is referred to as Alpha1 and can be seen in Figure 6. The body of the drone was not designed to the needs and requirements of the components since it was created in a previous year project. The structure of the body was able to fit the components needed for the initial testing of the software/hardware, and hence able to fulfill the shortcomings of Alpha0.5.

Figure 6 Alpha1

From the tests of and flights with Alpha1, requirements and design decisions for further version could be derived. Autonomous flight was achieved with Alpha1, were it could execute simple missions, for example: Fly to point A, then to point B and following by a return to the launch position. When the tests were conducted, the test team was ready to take over manual control, in the blink of an eye, in the case if something were to go wrong. The biggest problem with Alpha1 was its battery time and it had to be ensured that a bigger battery and longer flight time would be possible for subsequent versions. Alpha 1 served its purpose as a test prototype, where settings could be evaluated to ensure that correct values were set for Alpha2.

3) Alpha2

The Alpha2, which was planned to be able to fit all the components required for flying autonomous, can be seen in Figure 7. The measurements for the design was thought to be correct, but it was too optimistic.

The PX4 must be placed as close to the center of gravity as possible to ensure a stable flight. If it is not centered, the user must manually input the offsets in every direction, to make the PX4 understand its placement on the drone. Additionally, due to the short length of available cables, the PMB had to be placed close to the PX4, rendering it impossible to free up necessary space for the PX4.

Alongside the PX4 and the PMB, other peripherals such as telemetry and the LiPo battery must fit to the design of the drone. Hence the failure of the design of Alpha2, Alpha3 were designed with considerations of the mistakes which were made during Alpha2 design phase.

Figure 7 Alpha2

4) *Alpha3*

Because of the limitations to the design which were found during the design of Alpha1 and Alpha2, Alpha3 had to be a conceptual re-design. The design approach was drastically changed. Since the structure of the drone was supposed to be 3D printed, the design of the structure followed a "modular approach". With the modular approach each individual part can be replaced. As an example, an arm, could break and be replaced, without having to disassemble any other part of the drone. This meant that if a crash would occur, all the structural parts that got damaged, could easily be re-printed and then replaced in a short amount of time. The only requirement for the newly printed parts was that the screw holes had to be at the same position, to correctly align the new part with the rest of the structure. This means, that an arm for example, could be alternated in design very easily.

The main changes for the hardware of Alpha3 are new motors and different propellers. Alpha3 can be seen in Figure 8. The motors are more powerful than the ones on Alpha1, and the propellers are bigger and two-bladed instead of triple-bladed. During the test phase of Alpha1 and mostly Alpha3 (higher lift force from the propellers/motors), it was discovered that landing the drone with the help of only the barometer for altitude measurement, could not be considered reliable. A barometer measures the air pressure and then calculates the altitude of the drone. The main drawback with the barometer is that it has been calibrated with respect to a certain air pressure. If the pressure is different, there will be a slight error in its altitude estimation, and hence weather conditions have an effect on landing precision. Furthermore, when the drone gets close to the drone, the propellers will induce "downwash" [1][2] which will further decrease the accuracy of the barometer. This phenomenon reduces the precision of landings even further, to a degree where some landings were deemed unacceptable. To circumvent this problem, a LiDAR was added. The LiDAR use IR light to measure the distance to an object, or in this application, the ground. The LiDAR is combined with the barometer to find the correct altitude while the drone is at most four meters above ground. The added sensor increased the accuracy of the landings substantially.

Figure 8 Alpha3

One objective of the project was to be able to gather data from a sensor which was placed on the ground at a random location. The lack of space on Alpha1, made this task infeasible, therefore, the required components was added on Alpha3. The sensor which would gather data and transmit it to the drone is the Bosch XDK1100. The sensor can perform different kinds of measurements, such as temperature and humidity, and then transmit the gathered data via Bluetooth. To be able to receive and save the measurements, a single board computer, the Odroid XU4, was added to the drone. On startup, the it runs a python script which tries to connect to the Bosch sensor via Bluetooth. If a connection is established, the Bosch sensor transmits sensor data via Bluetooth, which is then saved to the memory of the Odroid. This process is always running when the drone is flying, which means that the operator just has to tell the drone to fly to the location of the Bosch sensor, hover for a couple of seconds, and then return to base.

Figure 9 shows an updated view of the system architecture, including all peripherals for Alpha3. An enlarged version can be found in [O]. In the figure, it can be seen which components communicate with each other, and which components that depend on another component.

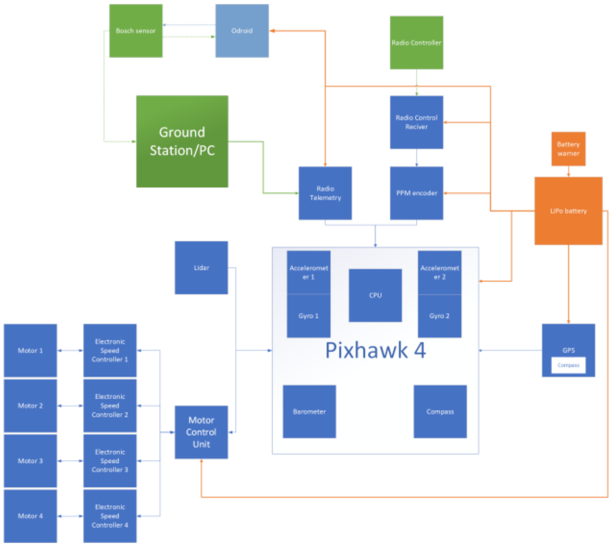


Figure 9 Overview of the final system architecture

Alpha3 can perform all moments of autonomous flight completely on its own, except collision avoidance (further development). Extensive testing has been conducted to ensure that the autopilot is reliable, and that safety features are in place and working. Alpha3 is the last design version where hardware components and software algorithms are evaluated. Some notes on parts which was not feasible to include in Alpha3 due to lack of resources is discussed in the “further development” section of the report.

5) Beta V1, 2, 3

The Beta versions are in a conceptual phase of development, where different kinds of materials for the frame of the drone will be evaluated, with respect to stiffness and weight, and how this will affect increase/decrease in flight time.

a) Beta V1 (carbon fiber and fiberglass)
Beta V1 was one concept which would have been further investigated in if time was available. The arms are made of fiberglass, and the middle platform are made of molded carbon fiber. Beta V1 can be seen in Figure 10.

b) Beta V2 (carbon fiber mold)
Since the group did not receive the molded carbon fiber frame, Beta V2 was not produced.

c) Beta V3 (3D printed carbon fiber)
The printed carbon fiber did not live up to the expectations of the materials properties. Due to the lack of resources to further investigate in this material option, it was instead scrapped completely.

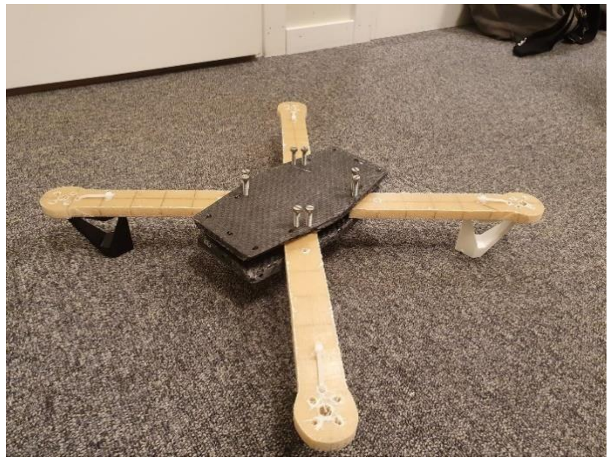


Figure 10 BetaV1

F. Project administration

Initially the project held the time plan, but towards the end of the development phase, the plan could no longer be followed, since problem occurred from both Vaxholm and other development phases such as crashes etc. Week 38-47 was within the time schedule, except assignments from the course examiners. Week 48-51 did not follow the time schedule, but the project ended at 18 of December, still with good results.

The budget set by MDH was 15 000 SEK. The budget was not met since parts of the project were more expensive than anticipated. Some components were prioritized to fulfill the project requirements.

IV. Reliability work

A. Safety

The safety process has identified the strengths and flaws and served as a base for UAV design decisions. The process that has been used is from the standards ARP-4761 and ARP-4754A but have been adapted for the mission criticality classifications of the UAV, instead of the DAL classifications, provided by the standards. The process consists of four documents in the following order: Requirements (Appendix C), Functional Hazard Assessment (Appendix D), Preliminary System Safety Assessment (Appendix 0), and System Safety Assessment (Appendix 0). The documents are further described in the Documents Description document (Appendix 0).

Most of the safety design decisions taken are based on software architecture where a watchdog has been implemented to serve as a fail-safe precaution to ensure that when no signals are sent or executed, action is taken. As the UAV is very limited in regards to space and battery capacity, which immediately affect the flight time and overall functionality of the UAV, design decisions opting for software solutions were deemed advantageous, because of the limits in space and power consumption, but hardware redundancy is also included to some degree.

Graceful degradation was implemented with regards to flight modes. A total of five flight modes were implemented: Manual, Altitude, Mission, Return to Base and Emergency flight mode. Altitude flight mode is regarded as an idle, safe state and is activated when the watchdog is triggered. When in altitude flight mode the drone holds the current altitude awaiting control commands. Return to Base is a mode in which the UAV returns to its launch location. This mode is triggered when signal is lost, i.e. connection to RC or base station. Emergency mode is a last resort where the UAV emergency lands in its current location [3].

Redundancy was implemented for sensors needed for flight controls. Two accelerometers and gyroscopes are present in the PX4, which alerts in case of inter-sensor inconsistency. A compass and GPS are combined to enhance the UAV orientation. The barometer is used for altitude measurements together with the LIDAR to achieve precision landings, which make the landings safer as the barometer is sensitive to changes in weather.

Hazards which cause loss of functionality have been reduced by the design decisions made. Single points of failure identified are ensured to have very low failure rates. Erroneous behavior of components can however not be reduced to an acceptable degree. There are comparisons and assertion checks, which can identify the erroneous events during startup of the UAV. This cannot be dealt with in-flight in a safe manner. The action taken, in the event of such a condition, would be to swap to a safer flight mode, assuming it is a non-crucial part that is affected by the erroneous condition. This can however not be verified.

Designing the UAV from a safety perspective is challenging since the UAV is very limited with respect to space and battery. Implementing more components to increase the safety levels of the UAV means that power consumption will increase and requires more space. This means some tradeoffs must be made and some implementations that could increase the safety levels have to be skipped.

B. Validation & Verification

The Validation & Verification (V&V) process strives to prove the produced products compliance with the needs of the customer. This is firstly done through validation, proving the requirements being in-line with customer demand, and that they conform with standard praxis in the industry such as EARS [10]. Secondly, a verification of product shall be performed to prove the systems compliance with the set requirements.

Since no clear regulation standard exists for the focus area, the project follows the guidelines on how to develop a safety-critical system from the aviation industry: ARP4754A & ARP4761. From these regulatory documents the V&V process were structured, adapted to the project regarding manpower and time.

1) Validation

The validation process, as shown in Figure 11, begins by taking the customer requirements as initial inputs. With the requirements of the customer clear, their knowledge and assumptions can be applied, to determine what is necessary to fulfill their needs. These inputs are processed through validation methods. Primarily, through a requirement review, where the reviewer asks themselves a series of questions regarding the requirement and its rationale. A traceability tree was also created to supplement the validation report, showcasing the connection between higher-level and lower-level requirement.

The validation methods will produce validation evidence that can prove that the requirement, set by the team, is in fact in line with the needs of the customer.

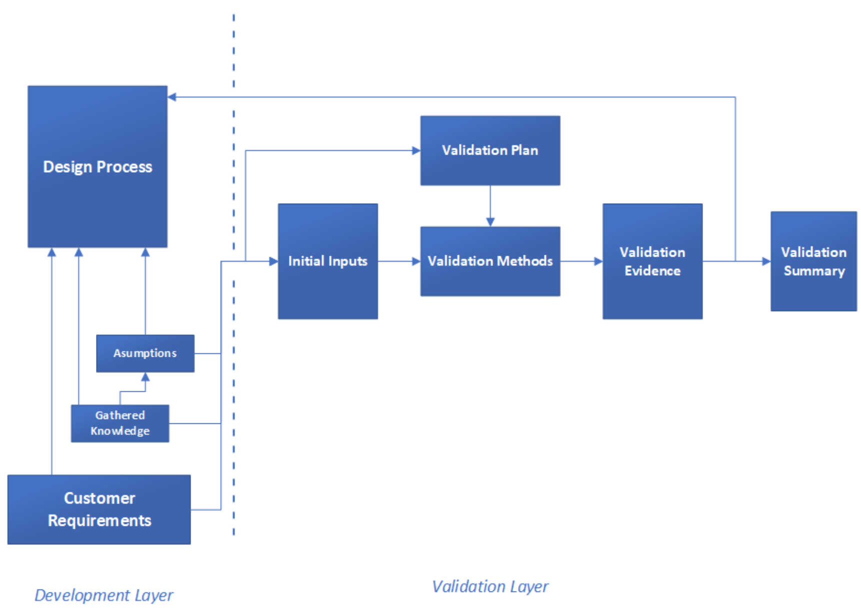


Figure 11 Validation Process.

This is then finalized, providing a verdict of what the evidence proves, see Appendix 0 and C.

In this project the validation process ended with a verdict approving the requirements, as the evidence was pointing towards the requirements being in line with expectations and needs of the customer, user and maintainer.

2) Verification

The verification process, which can be seen in Figure 12, takes requirements as input. Three different methods have been chosen to determine how well a requirement has been met: Analysis, Tests & Coverage Analysis. From the initial requirements the verification methods are formed, tests are arranged, and analyses are performed. These methods produce evidence which is tracked in the Design Verification Matrix (DVM). Every test and analysis get one of three classifications: Failed, Partial Approval, or Approved. Based on the evidence produced from the verification methods and its classification, a final verdict for process can be reached, see Appendix I-L.

In this project the verification process gave the implementation of our system an approval, even though some partial approvals of requirement were present. This could be argued to be acceptable due to time and resource constraints. In further development of the project some of these partially approved requirements should be resolved.

V. The UAV

A. *Software*

1) *QGroundControl*

QGC provides mission management for, and control over PX4 [4]. QGC is used to set up the drone with the firmware. The firmwares that can be chosen are PX4/native or Ardupilot. We chose to use the native firmware because it is supported for the PX4 and is still actively being developed.

QGC works with many different airframes such as fixed wing, quadcopter, VTOL, and many more. The airframe that is used in this project is a quadcopter.

The calibration of the sensors is performed in QGC by following a step-by-step guide. During the calibration the drone is rotated at all the possible angles. The ESC calibration is simply done in QGC by pressing a button and toggling the power.

QGC has a graphical user interface with a map that shows the drone location, orientation, speed, and other information about the mission. The missions can be created, uploaded, and executed with QGC. Figure 13 illustrates what the operator sees in QGC. The operator can send commands to the drone during missions, from the leftmost menu, including land, return to the launch position (RTL) and pause the current mission. If the pause command is sent, the drone will stop and hover at the current position. Land will issue a direct landing and RTL will tell the drone to fly to the launch position and land.

QGC saves the flight logs with all the sensor information.

Figure 13 QGC interface during a mission

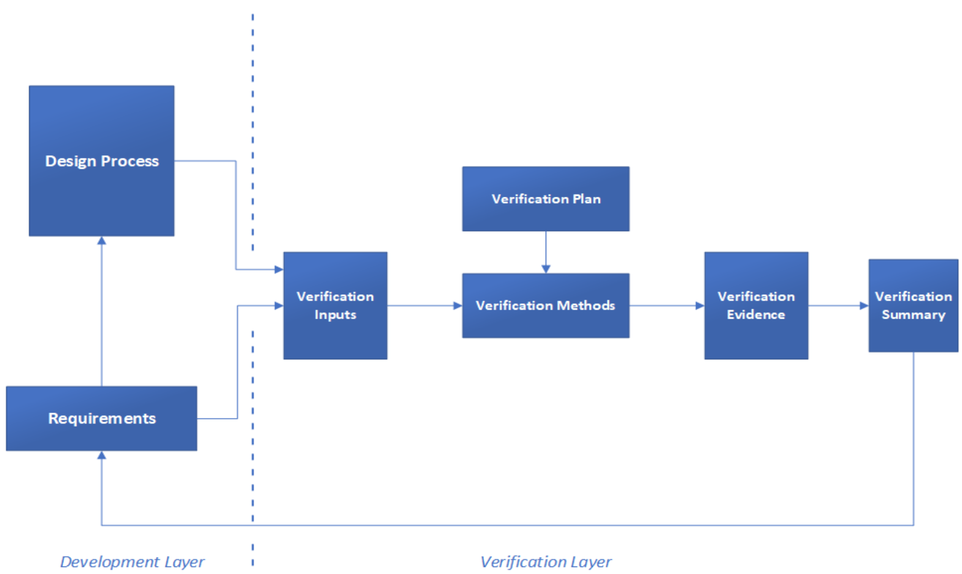


Figure 12 Verification Process

2) *Python*

The development of the software was mainly done in Python. The reason why Python was chosen is because there already exists libraries in Python for sending messages to the PX4. Python was also used to communicate with the Bosch XDK sensor via Bluetooth Low Energy (BLE).

3) *Bosch XDK*

The Bosch XDK is an all-in-one programmable sensor. It uses a 32-bit ARM Cortex M3 microcontroller. The Bosch XDK has an accelerometer, gyroscope, magnetometer, humidity sensor, pressure sensor, thermometer, acoustic sensor, and a light sensor. The values that are collected are pressure, temperature, and humidity. The Bosch XDK is programmed in the C programming language. It uses the Free Real-time Operating System [8]. It also has BLE, which is used for communication between the sensor and the drone on-board computer (Odroid XU4). The Bosch XDK is developed with the XDK Workbench [5].

The Bosch XDK was programmed to always have BLE enabled. When a device connects to the Bosch XDK then it will start collecting humidity, pressure and temperature readings and send them over BLE to the connected device.

4) *Odroid XU4*

Odroid XU4 was used as the onboard computer for the drone. Its main task is to collect data from the Bosch sensor whenever it is in range. To accomplish this task a Python script was developed. The script tried to connect to the Bosch XDK all the time, once it is connected it will collect the data and save it to a text file, with time and date stamp.

The data collection script is started on reboot and runs all the time, so it does not require to be manually started for every mission.

5) *MAVLink*

Micro Air Link Vehicle Link (MAVLink) is the communication protocol that is used to communicate with the PX4. MAVLink can also be used to communicate with onboard drone components. The MAVLink message definitions are stored in an XML documents in the MAVLink source. Custom MAVLink messages can be created but were not required for this project. All the messages for the PX4 are already defined. Most of the communication with the drone is done with MAVLink messages, e.g. uploading missions, change or parameters.

MAVLink libraries can be generated for C, C#, C++, Python and more. No MAVLink libraries were generated in this project because DroneKit was used and no custom MAVLink messages were necessary.

MAVlink can support up to 255 concurrent systems, (e.g. vehicles & ground stations).

6) *DroneKit*

DroneKit is an open source high-level Python library that is used to develop the software that will run on the ground control station. DroneKit is used to communicate with the drone via MAVLink. Dronekit can be used to read/set drone parameters, e.g. reading the battery level or setting the RTL altitude. It is also possible to take direct control over the drone movements and operations [6].

Dronekit is used to create, upload, execute and monitor missions and to set or get parameters such as altitude, speed or battery level.

7) *Apache Thrift*

Thrift is a code generation tool that is used to enable cross-language communication between the Python scripts and the MMT. Thrift supports several languages and among them are Python and C#. The Thrift architecture consists of a server and a client. The code is generated, for each specified programming language, by defining the service interfaces and data types in a Thrift configuration file.

Thrift supports several different protocols for serialization and deserialization. The server/client uses the TBinaryProtocol, which encodes numeric values as binary. The TBufferedTransport provides buffering of input/output data at the transport layer and was used for both server and client.

Thrift supports basic data types such as booleans, bytes, doubles, strings, integers and structs. Structs are equivalent to classes in Python. The data flowcharts for the Thrift server are shown in [7].

8) *JMAVSim*

In order to test the code without risking crashing the drone jMAVSim was used. jMAVSim simulates a PX4 Quadcopter or multirotor. It has MAVLink support, so MAVLink messages can be tested. Another advantage of using jMAVSim is that there is no need to wait for the batteries to recharge between tests [8].

JMAVSim works with QGC and DroneKit. Parameters such as starting position, battery drain, etc., can be changed before starting the simulation. Missions are uploaded via MAVLink with QGC or the developed Python application.

B. Hardware Electronics

1) Odroid-XU4

Max input voltage:	5V
Max input current:	4A
Main Processor:	Samsung Exynos5 Octa ARM Cortex™-A15 Quad 2Ghz
IO Processor:	Cortex™-A7 Quad 1.3GHz CPUs
RAM	2GB 933MHz
Dimensions:	83 x 58 x 20mm
Weight:	60g
Operating temp:	-40-90°C

The Odroid XU4 is an embedded computer onto which a Linux OS environment was installed. The XU4 has 3x USB port, is an open source platform, and it is easy to plug in different sensors to the it, depending on the mission. The current setup is to, via Bluetooth, collect data from the BOSCH ground sensor [11].

The Odroid is isolated, in terms of communication, from the PX4 so it can never interfere or harm the flight control system. Thanks to the setup developers can create programs or try out new sensors for the Odroid separately, as a stand-alone unit, without having to use the actual drone.

The Odroid is powered by the main battery of the drone.

2) Motors

4x T-Motor MS2820 830 kv

Voltage:	2-4 cells LiPo (7.4-14.8V)
Dimensions:	35x42mm
Weight:	125g
kV:	830rpm/v
Effect:	668w

The motors have a typical low speed design with 830kv. With lower speed larger propellers are needed to generate enough lift for the UAV. Lower speed means less energy consumption and larger propellers contribute to the stability of the flight. Important for non-ideal weather conditions or carrying a payload. Small motors and smaller propellers would make the UAV faster, but this did not suit the requirements. Motor specifications are available in Appendix N.

3) Pixhawk 4

Power module output:	4.9-5.5V
Max input voltage:	6V
Max current sensing:	120A
Main Processor:	STM32F765 - 32 Bit Arm ® Cortex®-M7, 216MHz, 2MB memory, 512KB RAM
IO Processor:	32 Bit Arm ® Cortex®-M3, 24MHz, 8KB SRAM
Weight:	15.8g
Operating temp:	-40-85°C
Storage temp:	-40-85°C
Dimensions:	44x84x12mm

The PX4 FLCC is the heart of the quadcopter. This device provides drone control all I/O connections for peripherals such as motors and external sensors. The group has chosen to go with the PX4 since it is an open source platform, and has many built in functions such as gyro, blackbox, CAN-network, I^2C, WIFI, etc. [12].

GPS

The PX4 uses an external ublox Neo-M8N GPS/GLONASS receiver. This module compares its heading with the integrated magnetometer IST8310 for increased accuracy [12].

Barometer

The barometer integrated within the PX4 is a MS5611 unit. The barometer primary task is to measure the air pressure [12].

Accelerometer and Gyroscope

The accelerometer, BMI055/ICM-20689, and the gyroscope, ICM-20689, inside the PX4 work together to determine the tilt acceleration. The ICM-20689 is actually a 6-axis gyroscope/accelerometer with 3 axes for each sensor. The BMI055 works as a redundant accelerometer. The main difference between the accelerometer and gyroscope in general is that the gyroscope can sense rotation. The accelerometer does not take the earth gravitational pull into consideration, and hence it can only determine linear acceleration [12].

Magnetometer

The magnetometer IST8310 is the compass of the drone. It measures the external force from magnetic field of the earth to determine heading. Inside the magnetometer is an inductor that is constantly changing its magnetic field by flipping the polarity. This creates a stable PWM signal. When an external magnitude is affecting the signal, strength and direction can be determined by measuring the deviation from the original signal. With 3 different inductors inside the magnetometer it can determine X, Y and Z direction of the earth magnitude [12].

Radio antennas

There are two different radio antennas attached to the UAV. One is for connecting the manual mode RC and the other is data connection of the mission control computer. The data-link radio antenna is a V3 3DR Radio Telemetry 433MHz [13] and the RC antenna is i8A 2.408-2.475GHz [12].

PMB

The Power Management Board (PMB) is a power distribution board, to which the main power source is connected, the Li-Po battery. The battery itself feeds the circuit with 11.1V, at full charge, decreasing during discharge. The PMB converts the voltage to a constant 5V and the input voltage. The PMB distributes power to the components on the quadcopter, e.g. PX4, motors, Odroid, future payload, etc. [12].

4) *ESC*

EMAX BLHeli 20A

The Electric Speed Controllers (ESC) are directly connected to the motors and each controls the speed for the specific motor. The ESCs get their control signals and power from the PMB [13].

Voltage:	2-4s LiPo (7.4V-14.8V)
Dimensions:	52x26x7mm
Weight:	28g
Max Current:	25A
Max Speed:	35k-210k

5) *PPM encoder*

Holybro PPM Encoder

Dimensions:	22x19x5.5mm
Weight:	1.45g

The PPM encoder translates PWM signals for the PX4. The manual radio controller sends PWM signals. For the PX4 to understand the commands the signal must be converted to PPM signal.

The PPM and PWM signals do not match up, the PWM signals sends more continuously, which will cause a deadline miss. A deadline miss from the controller could have terrible consequences as the manual control is to provide a dependable backup in case the autonomous control fails [15].

6) *Propellers*

Dimensions:	254x11x2 mm
Weight:	15g

HobbyKing Slowfly Propeller 10x4.5

The propellers are Slowfly 10x4.5”, which as indicated by the name, are low speed. With larger propellers the drone will be less affected by winds and gushes, at the cost of max speed. This suits the project purpose as high speed is not a requirement. With smaller propellers and higher speed also comes a higher power consumption.

Capacity:	8000mAh
Type:	Li-Po
Voltage:	3S1P / 3 Cell / 11.1V
Discharge:	30C Constant / 60C Burst
Weight:	565g (including wire, plug and shrink wrap)
Dimensions:	167x69x24mm
Balance Plug:	JST-XH
Discharge plug:	XT60

7) *Battery*

ZIPPY Compact 8000mAh 11,1V Li-Po

The battery delivers power to the whole drone, including the Odroid and other separate sensors. It is the only power source on the drone [16].

8) *BOSCH-sensor*

BOSCH xdk110

Max input voltage:	5V
Max input current:	500 mA
Main Processor:	ARM Cortex M3
RAM	128 kB
Dimensions:	83 x 58 x 20

Weight:	60g
Operating temp:	-20-60°C
Internal battery	Li-Ion 560 mAh

The Bosch XDK110 is a universal programmable sensor that can collect data from a number of sensors, such as: humidity, pressure, and temperature (BME280), accelerometer (BMA280), light (MAX44009), acoustic (AKU340), gyroscope (BMG160), magnetometer (BMM150), and a barometer (BMI160). The Bosch XDK is an open source platform and can be tailored to fit the application. The BOSCH sensor serves as a ground station for the project. When the drone is in range the drone can connect to the sensor via Bluetooth [4].

9) Lidar

Input Voltage	5V
Max current	800 mA
Weight:	6.1g
Operating temp:	-20-60°C
Operating range:	0,3-12m
Dimensions:	42x15x16mm

The Benewake TFmini is a Light Detection And Ranging (LIDAR) sensor. This sensor measures the distance between the quadrocopter and the ground. Its range is 0-12m and it is automatically activated during landing, when the barometer indicates an altitude below 4m. This creates smoother landing since the barometer is inconsistent [17].

C. Hardware Structures

1) CAD

The 3D design of the quadcopter has been made in SolidWorks 2019©. The goal of the design was to make a modular quadcopter, where it is easy to change any part, depending on the mission, or if anything is broken.

The design is based around a middle platform, or body, onto which everything is attached.

2) 3D Printing

The 3D printing program and slicer, Cura 4.3.0©, was used throughout the project. Since the goal was to make quadcopter, that allowed for further development, the 3D printing was perfect to test and apply different parts, with updates of the design.

PolyLactic Acid (PLA) 3D printing comes with a weight penalty compared to carbon fiber. However, carbon fiber was too difficult to apply to the UAV. Both the ordered carbon from Vaxholm and the 3D printed carbon fiber was unsuccessful. Hence, PLA was used instead, with different Tri-Hexagon infills, as tradeoff between strength and weight. For the quadcopter arms, which need to be strong, infill was 15-30%. As the arm was updated during the project, the infill changed. For other parts the infills could be around 8-15%.

The 3D-printer was an Ender 3 brand from Creality.

3) Materials

Polylactic acid (3D printed)

PLA is a thermoplastic material with a higher strength and stiffness than nylon. PLA has a low melting temperature and minimal warping (twisted/out of shape material) which makes it one of the easiest materials to 3D print. PLA is brittle which leads to parts with poor durability and impact resistance [18].

Nylon Carbon fiber filament (3D printed)

Nylon is a flexible, durable plastic with less strength and stiffness than PLA. Its flexibility makes it much tougher than PLA. Nylon requires extra care to print. It needs to be extruded at high temperatures and, due to its tendency to soak up moisture from the air, must be kept in a dry box [19].

Carbon fiber 400HV

Carbon fiber that was supposed to be used was 400HV, with carbon fibers in two directions, for high strength and tension. The 400HV needed 400gr of plastic per m2. Carbon fiber needs to have either iso-vinylester or epoxy to bind the fibers with the distance material [20].

4) Materials Strain- & pressure test

All tests on structure and materials were performed in the IDT workshop in Eskilstuna.

The strain and pressure tests were done with two different materials, biaxial-fiberglass and chopped fiberglass. All the parts had the same height, length, and width so the experiment would become more reliable and repeatable. Multiple strain tests were conducted with similar pieces and then a mean value could be calculated for each material.

The initial plan was to perform strain and pressure test on carbon fiber, biaxial fiberglass, chopped fiberglass fiber and 3D PLC plastic. However, this was not possible since Vaxholm Komposit never delivered the parts.

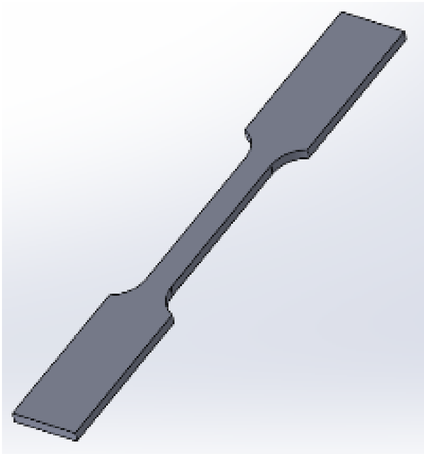


Figure 14 Strain test part.

Three identical strain tests parts, as illustrated in Figure 14, were tested. The result of the strain tests for chopped fiberglass are shown in Figure 15. As can be seen, the parts do not withstand the same amount of force before reaching the breaking point. Hence, the mean result calculated is: $\frac{1485+1410+1738}{3} = 1544{,}33N$

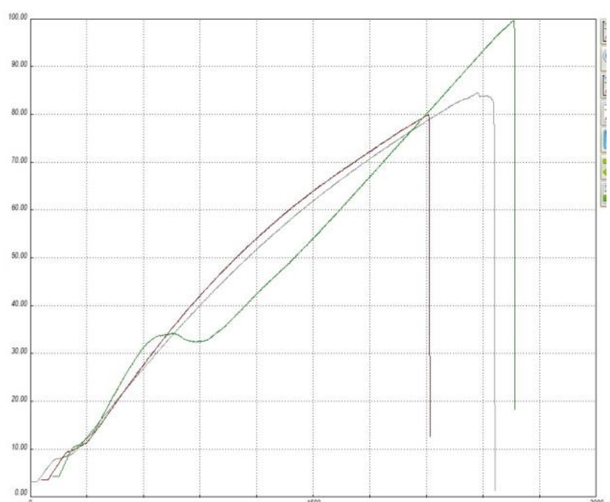


Figure 15 Strain test result, chopped fiberglass

The procedure was repeated for the biaxial fiberglass and the results are shown in Figure 16. The mean result is: $\frac{927+1293+1497}{3} = 1239N$

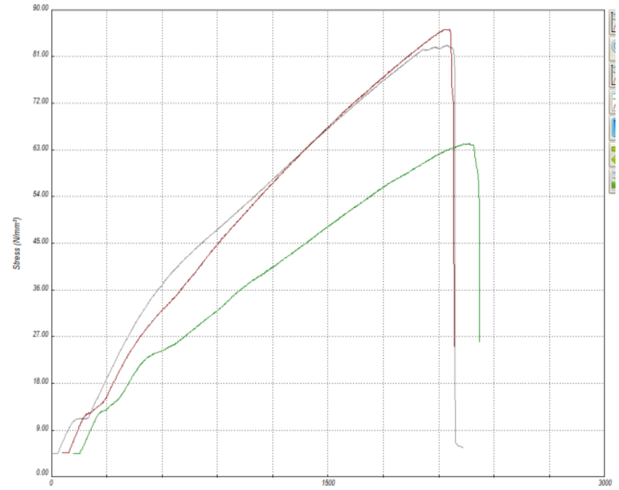


Figure 16 Strain test result, biaxial fiberglass

The final test was the pressure test. The test setup can be seen in Figure 17. Two identical pieces were used, one from chopped fiberglass and one biaxial. The results are shown in Figure 18. The chopped (red) withstood 1085N and the biaxial (green) withstood 692N before breaking.

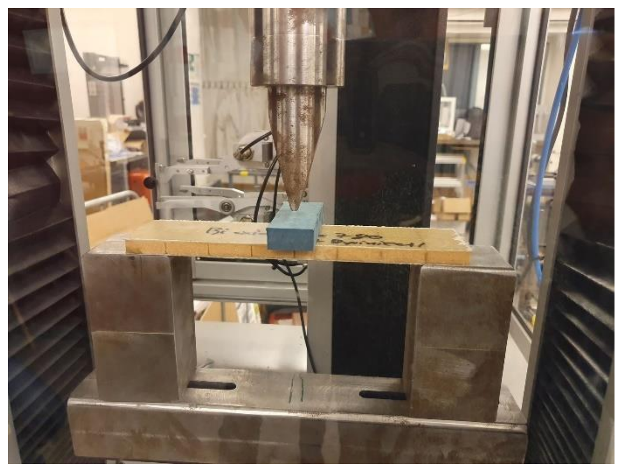


Figure 17 Pressure test setup

Both pieces had the exact same dimensions but there was a weight difference between the materials. The chopped piece was 56.6g and the biaxial 51.6g, so the biaxial was a bit lighter.

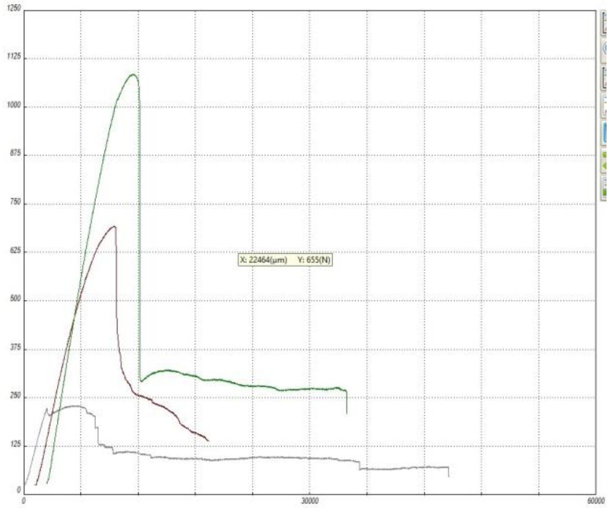


Figure 18 Pressure test results, chopped (red) and biaxial (green) fiberglass.

As a conclusion the chopped material was better for both strain and pressure force. For the drone this means that chopped should be preferred when making the final version. The weight difference was too small to include in the decision making as the chopped was so much stronger.

V. Conclusion

The final drone fulfilled the objectives of the project. The drone can fly missions autonomously, where the operator defines the missions. The drone will fulfill every mission if there are no objects in the way, and no critical failures occur, or the battery runs out. The operator can take full control of the drone in any given flight phase.

To further increase the functionality of the delivered UAV, obstacle and collision avoidance could be integrated.

Investigating further optimization of materials and components could be done to increase the dependability and functionality of the UAV.

Overall, the project was a success and both the project customer and the project group are satisfied with the achieved result, given the limited time.

A. Some common mistakes

During the development of OpenDrone numerous mistakes were made.

Tests were performed multiple times without any pre-flight checks. Establishment of check lists and routines could have prevented crashes.

Before performing a test, the drone must be analyzed, and potential problems that could occur during flight must be identified before takeoff.

Another mistake was that the group did not learn from some of the crashes. Instead of changing the parameters, which were derived as faulty or plain out wrong, the development went on without taking a step back and finding the actual source of the problem.

B. Further Development Discussion

As the first group working on this project, we have accomplished a lot on limited time. The focus was to lay the foundation for the Open Drone project and develop the most important features, such as autonomous flight and data collection. During the development process many areas of future development have been identified.

Obstacle avoidance is a possible feature for future works. Depending on the environment and mission given to the drone, obstacle avoidance could be required for a safe and reliable flight mission. A 360-degree rotating lidar would be suitable for this application.

Object detection can be implemented using a camera combined with AI software. Object detection may be used for missions where specific objects are to be found, tracked and/or followed. For farming applications animals (e.g. a herd of cows) may be followed and tracked to make sure that they are within a specified area.

Adding additional motors to the current quadcopter (hexa-, octocopter) gives the possibility to develop a larger drone that could fit more components such as a camera or a lidar sensor.

As of now, control and mission planning for one drone has been developed. Multiple drones can be controlled and communicate with each other through the MMT, but the Python software needs to be developed to handle multiple drones.

For missions in harsh environments, improvements in robustness can be implemented to tolerate different types of weather, e.g. rain, snow, heat and cold.

The frame can always be improved. As of now, some components are easy to add/remove while others are not. If more components are to be added, such as a lidar, the frame must be redesigned for these to fit.

Erroneous flight mode control is something that is very relevant since the UAV does not achieve this level of dependability. This is an area that should be analyzed further and implemented to ensure this level of dependability.

C. Materials Strain- & pressure test

The test result was made by the project at MDH Eskilstuna, where the result was that the chopped fiberglass was better than biaxial. The information from our sponsors, Vaxholm Komposit, is that directed fiber such as biaxial is stronger than chopped fiber.

The test showed that chopped is stronger both in strain and pressure, but the conclusion is that the area of the drone is so small and chopped has small fibers in all directions and that could help the strain and strength in all directions.

Weight-wise biaxial would require 900g of plastic, such as iso-polyester, per m2, while the similar number for chopped is 450g. This could not be confirmed during the weight test of the material. Faulty implementation of the iso-plastic during the creation of the material could be a reason for the discrepancy.

VI. References

[1] Yeo, D. Shrestha, E. Paley, D. Atkins, E. *An Empirical Model of Rotorcraft UAV Downwash for Disturbance Localization and Avoidance,* AIAA Atmospheric Flight Mechanics Conference, *5-9 January 2015*

[2] Hall, N. NASA (2015). Downwash Effects on Lift. [online] Available at: https://www.grc.nasa.gov/WWW/k-12/airplane/downwash.html [Accessed 16 Dec. 2019].

[3] Docs.px4.io. (2019). *Flight Modes · PX4 v1.9.0 User Guide*. [online] Available at: https://docs.px4.io/v1.9.0/en/flight_modes/ [Accessed 12 Dec. 2019].

[4] Gagne, D. (2019). Overview · QGroundControl User Guide. [online] Docs.qgroundcontrol.com. Available at: https://docs.qgroundcontrol.com/en/ [Accessed 12 Dec. 2019].

[5] Xdk.bosch-connectivity.com. (2019). Hardware - XDK. [online] Available at: https://xdk.bosch-connectivity.com/hardware [Accessed 12 Dec. 2019].

[6] GitHub. (2019). dronekit/dronekit-python. [online] Available at: https://github.com/dronekit/dronekit-python [Accessed 12 Dec. 2019].

[7] Thrift-tutorial.readthedocs.io. (2019). *Thrift protocol stack — Thrift Tutorial 1.0 documentation*. [online] Available at: https://thrift-tutorial.readthedocs.io/en/latest/thrift-stack.html#runtime-library [Accessed 12 Dec. 2019].

[8] Dev.px4.io. (2019). *jMAVSim Simulation · PX4 v1.9.0 Developer Guide*. [online] Available at: https://dev.px4.io/v1.9.0/en/simulation/jmavsim.html [Accessed 12 Dec. 2019].

[9] FreeRTOS. (2019). FreeRTOS - Market leading RTOS (Real Time Operating System) for embedded systems with Internet of Things extensions. [online] Available at: https://www.freertos.org/ [Accessed 16 Dec. 2019].

[10] Mavin, A. (2019). EARS – The Easy Approach to Requirements Syntax: The Definitive Guide - QRA Corp. [online] QRA Corp. Available at: https://qracorp.com/easy-approach-to-requirements-syntax-ears-guide/ [Accessed 16 Dec. 2019].

[11] Odroid Wiki Specifications (2019)[Online] Available at: https://wiki.odroid.com/odroid-xu4/hardware/hardware#specifications [Accessed 2 Dec. 2019]

[12] Holybro.com. (2019). PPM ENCODER [online] Available at: http://www.holybro.com/manual/Pixhawk4-DataSheet.pdf [Accessed 16 Dec. 2019].

[13] www.banggood.com. (2019). *New V3 3DR Radio Telemetry 433MHz 1000mW with OTG Cable for APM Pixhawk Flight Controller RC Drone RC Parts from Toys Hobbies and Robot on banggood.com*. [online] Available at: https://tinyurl.com/rcw4wte [Accessed 16 Dec. 2019].

[14] Imgaz.staticbg.com. (2019). Electronic Speed Controller. [online] Available at: https://imgaz.staticbg.com/images/upload/2014/12/Emax%20BLHeli%20ESC%20Manual.pdf [Accessed 16 Dec. 2019].

[15] Download.ardupilot.org(2019) [online] Available at: https://download.ardupilot.org/downloads/wiki/pdf_guides/PPM-Encoder-V3-Manual.pdf

[16] ZIPPY Compact 8000mAh 3S1P 30C Lipo Pack w/XT90. (2019) Hobbyking Available at: https://hobbyking.com/en_us/zippy-compact-8000mah-3s1p-30c-lipo-pack-with-xt90.html

[17] Odroid Specification [Online] Available at: https://wiki.odroid.com/odroid-xu4/hardware/hardware

[18] https://cdn.instructables.com/ORIG/FX6/XPEZ/JCKUE79S/FX6XPEZJCKUE79S.pdf [Accessed 2 Dec. 2019]

[19] Creality3dofficial.com. (2019). PLA Filament – Creality3D Store® Official Store for Creality 3D Printers and Accessories. [online] Available at: https://www.creality3dofficial.com/products/pla-filament?variant=16021777481777 [Accessed 16 Dec. 2019].

[20] Black, P. (2019). PrimaSelect NylonPower Carbon Fibre - 1.75mm - 500g - Natural - Black. [online] 3D-Prima. Available at: https://www.3dprima.com/filaments/primaselect-nylonpower-carbon-fibre-1-75mm-500g-natural/a-23423/ [Accessed 16 Dec. 2019].

[21] ThoughtCo. (2019). How Is Carbon Fiber Made?. [online] Available at: https://www.thoughtco.com/how-is-carbon-fiber-made-820391 [Accessed 16 Dec. 2019].

VII. Appendices

A. Rad, A. (2019). Process Assurance. 1st ed. Västerås: OpenDrone Project, p.9-13.

Alpha 0.5			
Model Frame	HW	SW	V&V
The Alpha 0.5 model was designed using Solidworks to make a basic model and then 3D printed. **Deviations:** The frame was too small to fit the necessary components for test. **Corrective actions:** The flight controller and motors were connected together without a frame. This solution worked since flying was not necessary for this version. The goal was just to test the connection between the flight controller and motors but also to have control using RC and QG.	Four, old motors, pixhawk 4 and a small battery. **Deviations:** No wireless control using RC or QG could be done since we had no transmitter/receiver. **Corrective Actions:**. Transmitter/receiver module was ordered to be used for the next version.	Connection to Pixhawk was only possible through USB cable. **Deviations:** Same as for HW. **Corrective Actions:**. Same as for HW	Connection between the flight controller and motors was possible but not possible to control wirelessly using RC nor QG. **Improvements for next model:** • A larger frame to fit more components. • Transmitter/receiver module for wireless control.

Alpha 1			
Model Frame	HW	SW	V&V
Instead of designing and printing a new frame for Alpha 1, Alpha 0.5 has been replaced with an old drone frame (not 3D printed) with more room for components and suitable to achieved the goals of Alpha 1. **Deviations:** The frame was delayed so no frame was 3D Printed. **Corrective actions:** Replacement frame was used.	Old motors, pixhawk 4, receiver/transmitter radio antennas and a small battery was used on the replacement frame **Deviations:** The flight controller was mounted in the opposite direction compared to the motors which made it impossible to fly. **Corrective Actions:**. Switching the motors positioning fixed the problem.	Calibration was done using QGroundcontrol. **Deviations:** The calibration did not help since the flight controller and motors were wrongly mounted. **Corrective Actions:**. After switching the motors positioning, the drone could be correctly calibrated.	When testing Alpha1, it was able to fly manually and autonomously in a controlled manner using both RC and QGroundcontrol. **Improvements for next model:** • Create a 3D model that fits the necessary components but also has more room for additional components such as possible sensors. • Components needs to be protected as well. • Longer range for manual/autonomous control • Sensor data collection • Components for optimal flight time.

Alpha 2			
Model Frame	HW	SW	V&V
The Alpha 2 model was modeled with a frame with the shape of a box. The box consists of two floors. One floor for the battery and one for the other components. The box shape will protect the components from rain and collisions. **Deviations:** The model was still not large enough, a third floor has to be added and the frame needs to be increased in overall size for more room for components. The Alpha 2 model is therefore scrapped. **Corrective Actions:**. A new larger frame will be developed with an extra floor for more room. The new model will be called Alpha 3 and meet the requirements for both Alpha 2 and 3.			The Alpha 2 frame has been concluded to be too small for use and is therefore scrapped. **Improvements for Alpha 3 version:** ● Increase overall frame size and add another floor. ● Meet the requirement goals for Alpha 2.

Alpha 3			
Model Frame	HW	SW	V&V
The Alpha 3 model is improved with a larger frame with the shape of a box. The box consists of two floors for microcontrollers and modules etc and one floor for the battery.. The box shape will protect the components from rain and collisions.	New motors with low Kv-values has been added for increased flight time. Propellers with larger wing area has been added for better lift force. First two batteries was used instead of one which increased the flight time. A larger more powerful battery with larger capacity was then bought which resulted in a even longer flight time than than the use of 2 older batteries. For longer range, a new radio antenna was bought with longer range. An Odroid board has been added with a Bluetooth module for wireless connection to the Bosch sensor for sensor data collection. A Wi-Fi module has also been added for using TeamViewer between the ground station and Odroid. A Lidar sensor has been added for precision landing.	QGroundcontrol is still available for use for autonomous flight. SW has been developed in Python for autonomous flight. The SW can communicate with the MMT through the communication protocol Thrift. Code for the Bosch sensor and the Odroid has been developed for bluetooth communication for sending sensor data between the Bosch sensor and the Odroid microcontroller.	After the frame improvement components had more room and could more easily fit the box shaped frame where they are protected. Alpha 3 was also able to fly manually and autonomously in a controlled manner using both QGroundcontrol and using the Thrift protocol. Sensor data collection was tested and worked as planned. The drone can fly to a given waypoint where a Bosch sensor has been placed and collect the data wirelessly, save the data on the microcontroller and return to ground station. An autonomous mission was successfully done using the Thrift protocol creating a host and a client. The Thrift protocol allows a client (MMT) to run the mission. Longer flight time was achieved using the new battery which resulted in a hovering flight time of about 17 min. The autonomous landing was improved significantly using the Lidar sensor. **Improvements for final version:** • For the last and final version flight time needs to be maximized by reducing weight, creating a frame made of carbon fiber and removing unnecessary parts/shapes of the frame.

Beta			
Model Frame	HW	SW	V&V
The final version (Beta) has been improved with a carbon fiber/glass fiber frame which will reduce the weight by an estimate of 35%. **Deviations:** The frame for the Beta version was constructed but not used. A decision was made to stop the development at Alpha3 since further development won't be continued on the Beta version if constructed anyways. Further development will most likely require a new hexacopter frame to be modeled. **Corrective Actions:** Further development will be continued by developing a larger drone. A hexacopter is suggested.	Same	Same	

B. *Rad, A. (2019). Process Assurance. 1st ed. Västerås: OpenDrone Project, p.14-16.*

Problem report

GPS module mounting:

The GPS module was mounted too close to the flight controller which cause interference between the two. Since the flight controller uses GPS for positioning this had a large negative effect on the flying characteristics.

Corrective action:

Mounting the GPS module on a stick, positioning the module further away form the flight controller solved the problem.

Choosing correct sized screws when attaching motors:

A mistake of using too long screws when attaching motors lead to screws going in too far in to the motors copper coils and damaging the coil when the motor were in use. Luckily this was done on old reused motors.

Corrective actions:
Make sure the screws aren't too long hitting the coils before screwing the motors.

Flight controller setup:

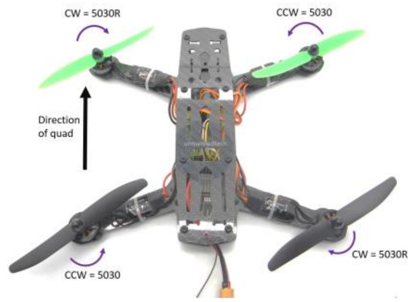
The flight controller was mounted in the opposite direction compared to the motors which made it impossible to fly. Switching the motors positioning fixed the problem. After switching the motors positioning, the drone could be correctly calibrated and flying was made possible.

Corrective actions:
Make sure the flight controller is in the correct direction compared to what QG is indicating.

Propeller blades Mounting:

Mistakes were made were propellers were mounted in the wrong direction.

Corrective actions:
To prevent this propellers have been marked with arrows so that the wrong direction won't be mounted. The propellers high angle shall be in the rotation direction.



Battery warner:
Forgot to disconnect battery warner from the battery over the weekend which can drain the batteries and allow the cells to become dangerously low.

Corrective actions:
Always disconnect battery warner when not in use.

Crash 1: Low battery

First time testing the battery time, landing was not done immediately after low battery warning. The battery was quickly drained and the drone crashed.

Corrective actions:
Increasing the battery warning limit from 10% to 30% battery remaining so that more time remains for landing.

Crash 2: Propeller test crash

Two propellers were merged together to create propellers with four blades to test if flight time or thrust is increased. The propellers were merged together using super glue. When the test was performed, one of the modified propellers broke loose and the drone crashed.

Corrective actions:
The idea of using quad propellers was scrapped since we after the test can conclude that thrust was increased but flight time remained the same. Using quad propellers won't benefit the purpose of this product.

Crash 3: Manual control failure

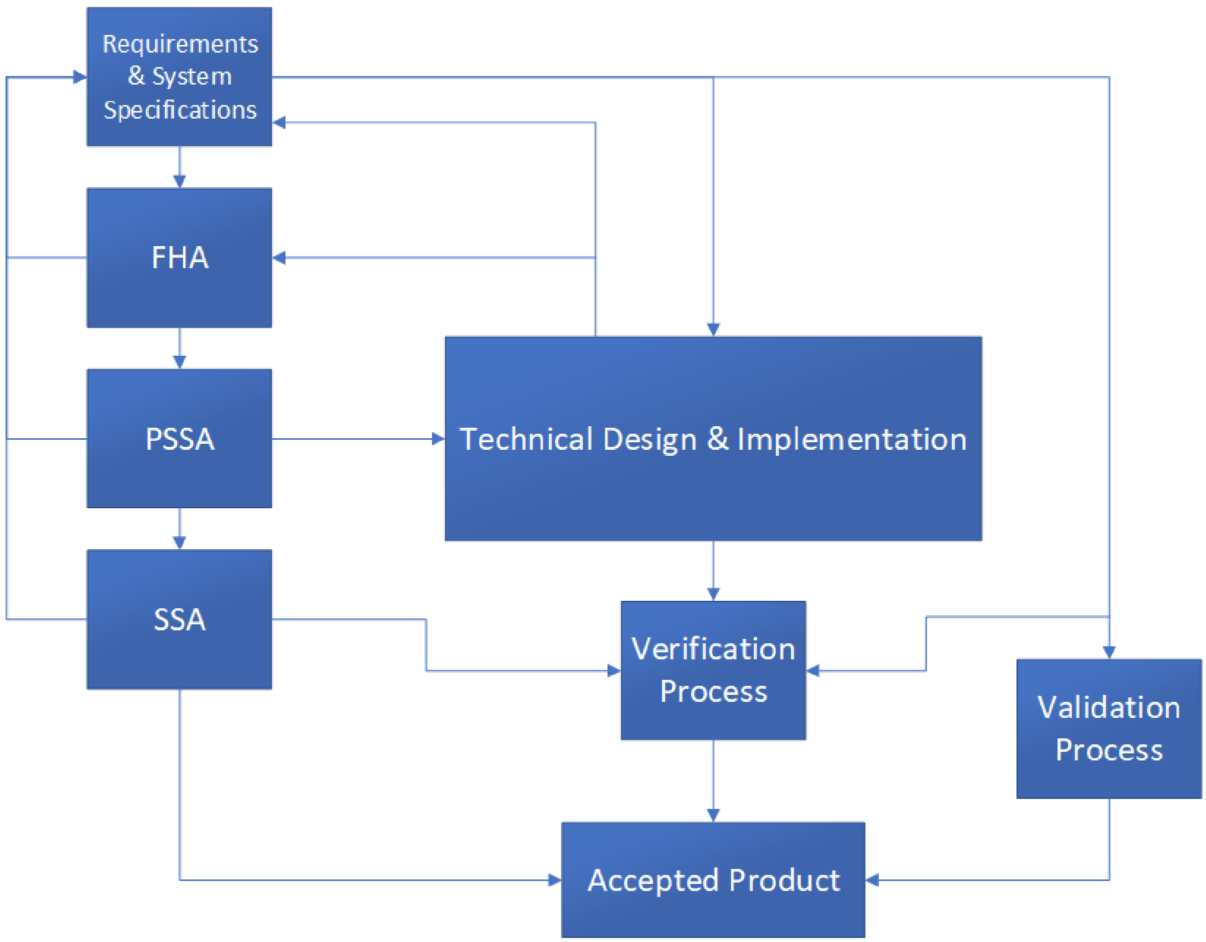
A test was performed. During the autoland part of the test mission, manual control was activated since autoland using barometric pressure sensor is not accurate enough. When manual mode was activated, the drone crashed because stabilized mode was activated and the throttle was accidentally set to 0, which resulted in the drone plummeting to the ground.

Corrective actions:
Before stabilized mode was the default mode when manual RC was activated, which does not support autoland/automatic hover and instead causes the drone to crash if the throttle is accidentally set to 0.

The default mode is now position mode to prevent this. If the throttle is set to 0 when starting the RC manual control, the drone will autoland since position mode is the default mode.

C. Harborn, J. and Nordvall, D. (2019). Requirements. 1st ed. Västerås: OpenDrone Project.

Requirements

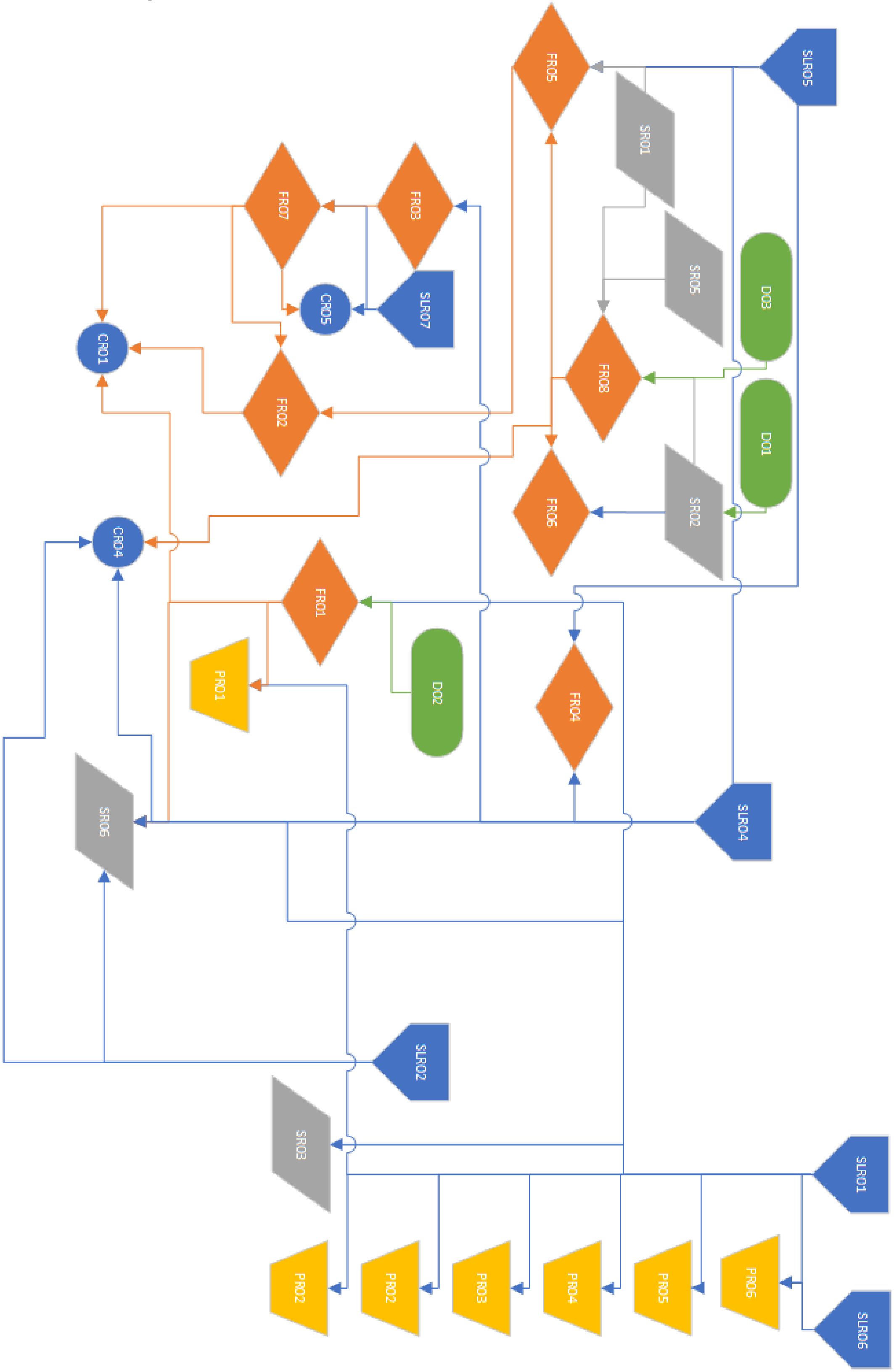


1. The Safety and V&V Process in Unison.

Definitions

ID	***Word***	***Definition***
D01	**Essential equipment**	Equipment that is needed for the UAV to fly. (All equipment except mission specific sensors).
D02	**Real-Time**	Describes the process of event to system response. A real-time system would control its environment by receiving data, processing said data and acting upon the result quickly enough to affect the environment at that time.
D03	**Flight modes**	Flight modes define how the autopilot responds to user input and controls vehicle movement.

Traceability Tree



2. Traceability Tree showcasing interconnection between requirements.

Functions

- Manual mode
  - Steering
- Mission mode
  - Automatic takeoff
  - Automatic waypoint-following
  - Automatic object/target tracking
  - Automatic landing
- Semi-autonomous mode
  - Automatic collision avoidance
- Sensor data gathering
- Data dump to base
- Heartbeat signal - System Status Signal
  - Position in X, Y, Z
  - Ground speed
  - Airspeed
  - Inner status (battery)
  - Task status
  - (Pitch, Roll, Yaw/Heading, Camera feed)
- Return to base

Flight phases

- Takeoff
- Mission
- Hover (Idle)
- Landing

Failure Classifications

- Critical
  - Failure conditions that causes the loss of ability to control the UAV resulting in a crash or complete loss of functionality.
- Major
  - Failure Conditions which would reduce the capability of the UAV for example, a significant reduction in safety margins or functional capabilities.
- Minor
  - Failure Conditions which would not significantly reduce UAV safety.
- No safety effect
  - Failure Conditions that would have no effect on safety; for example, Failure Conditions that would not affect the operational capability of the UAV.

System Level Requirements:

ID	***Requirement***	***Rationale***	***MoC***	***Notes***	***Traceability***
SLR01	The drone shall fly autonomously with inputs from the Mission Management Tool (MMT). (*)	One of the initial goals of the project.	On site tests		
SLR02	The system shall allow the human operator to take manual control of the UAV during any given flight phase. (*****)	Primarily a safety feature, but regarded as a System Level requirement.	On site tests		
SLR03	The system shall allow collaboration with other UAV and other autonomous systems, such as Unmanned Ground Vehicles (*****)	Necessary to deliver a fully functional product to customer, must cooperate with other autonomous farming vehicles.	No V&V as it is part of the larger scope of Afarcloud .	This is outside the scope of the project and is a area of further possible development .	
SLR04	The system shall demonstrate a degree of dependability by comparing application domain standards with the experimental settings. (*)	The UAV shall work in a environment were multiple other systems work autonomously, the system must be trusted to perform its task.	Proved in the SSA.	By working accordingly to ARP4761 & ARP4754A a degree of dependabilit y can be achieved.	
SLR05	The UAV shall be built upon a open source platform to ease further development.	By leaving the design open, the project can be further developed and specialised for	Analysis		

		other tasks.			
SLR06	The UAV positioning solution shall provide the accuracy necessary to complete basic farming tasks.	Innate requirement on the UAV to fulfill basic agricultural tasks. Such as measuring humidity levels.	Analysis and tests.		
SLR07	The UAV shall be able to perform sensor based tasks during missions.	A basic task for the autonomous UAV is to fly to a waypoint and collect sensor information such as humidity data.	On site tests.		
SLR08	The UAV shall use the same date/time formats of other vehicles and nodes in the chain.	To prevent complications with communications between components.	Not tested.	This is outside the scope of the project and is a area of further possible development.	

Functional Requirements:

ID	Requirement	Rationale	MoC	Notes	Traceability
FR01	If the range requirements are met, the system shall send data in real-time to the MMT. (*****)	This will act as the primary communication method between the UAV and MMT.	On site tests.		SLR01, D02
FR02	The UAV shall post-mission transmit mission relevant sensor data to the MMT.	Gathered sensor data needs to be analyzed by the operator and therefore needs to be presented in a user-friendly manner.	On site tests.		SLR07, FR07
FR03	The UAV shall record safety critical faults.	If faults should occur these shall be recorded to allow patches to prevent the faults from reoccurring.	Analyis	The Pixhawk saves logs for each flight, these can be found on the SD card.	SLR04
FR04	The UAV shall be designed in a modular fashion to allow easy maintenance and replacement of parts.	Parts will degrade differently and should therefore be easy to replace and maintain by the operator.	Analysis of the structure.	Require ment fulfilled.	SLR04, SLR05
FR05	If prompted the UAV shall perform an emergency landing	To prevent the drone from taking	On site tests, Analysis.		SLR01, SLR04, SLR05, SR01, FR08

	with a low descend speed.	damage.			
FR06	The UAV shall have a “Return to Base” function, when triggered, signalling the UAV to immediately return to base.	In order to have a way to cancel a mission in progress.	On site tests.	“Return to launch”, built into many failsafe features.	SR02, FR08
FR07	The UAV shall store mission relevant data in the following ways: 1. Store relevant mission data onboard until post-mission transmission. 2. Send stored data about the mission to the MMT.	To allow analysis of measurements that data needs to be stored both during them mission and after its completion.	On sites test.		FR03
FR08	The UAV shall have flight modes that allow degraded operation as Return to Base and Emergency Landing.	To make it possible to swap between flight modes depending on desired functionality.	On site tests	Built into the Pixhawk system from the start	SR01, SR02, SR05, D03

Safety requirements:

ID	Requirement	Rationale	MoC	Notes	Traceability
SR01	If any critical failure occurs the UAV shall be able to perform a emergency landing.	If any critical failure occurs the UAV shall be able to land, not crash and damage the UAV.	Analysis of the system safety.	See classificati on list for definition of *Critical Failure*	Failure Classification list
SR02	The drone shall enter a low power mode turning off non-essential equipment and return to base when battery power is not sufficient to complete the mission.	To prevent the UAV from continuing the missions without the required battery time and failing to complete it.	Design analysis.	See definition table for definition on essential equipment.	D01
SR03	The operating altitude shall be no less than 20 meters above the ground.	To ensure the UAV can perform its tasks without falling the risk of disturbing potential cattle or other farm animals and avoiding terrain.	On site tests.		SLR01
SR04	The UAV should include collision detection and avoidance functionality.	For further development and full autonomy the UAV should be able to detect and avoid collisions, the scope of this project may however not cover this. entirely.	Not tested.	This is outside the scope of the project and is a area of further possible developme nt.	

SR05	The drone shall have a fail-safe mode with degraded operation.	If a failure occurs that compromises the safety of the UAV it should enter a degraded mode and return to base.	On site tests.		
SR06	The UAV shall transmit a system status signal with the following information: 1. Three dimensional positioning data. 2. Velocity 3. Airspeed 4. Remaining battery 5. Task status	Certain parameters are necessary to be recorded for scheduled maintenance of the UAV. These needs to be stored and analyzed by the MMT. The system status signal will act as a assurance for the MMT of the status of the UAV.	On site tests.		SLR01, SLR04, FR01

Performance requirements:

ID	Requirement	Rationale	MoC	Notes	Traceability
PR01	The drone **shall** be able to hover for 24 min minimum.	In order for longer missions to be possible, the drone must be able to fly for longer times.	On site tests.		SLR01, FR01
PR02	The drone **shall** have a telemetry range of 1 km.	Needs to be specified for testing engineers to know what to test.	On site tests.		SLR01
PR03	The drone **shall** have an obstacle detection range of 15 m or more of its surrounding.	A safe flight mission will require that the drone has well awareness of its surrounding to assure safe distance from obstacles.	On site tests.	This is outside the scope of the project and is a area of further possible develop ment.	SLR01
PR04	The drone **shall** have a ground detection distance of at least 30 m or more.	To assure a safe distance from terrain.	On site tests.	This is outside the scope of the project and is a area of further possible develop ment.	SLR01
PR05	The drone **shall** be able to operate in	In order to fly the drone in both cold and	On site tests.	This is outside the	SLR01

	temperatures between -20°C - 40°C.	hot environments.		scope of the project and is a area of further possible develop ment.	
PR06	The drone **shall** be able to withstand winds and fly in a controlled manner.	The drone shall not deviate from it's flight path when it's windy outside.	On site tests.		SLR01, SLR06

Communication requirements:

ID	Requirement	Rationale	MoC	Notes	Traceability
CR01	Sensor data **Shall** be sent to a cloud server.	Saving the data in a cloud server so that the data is not lost.	On site tests.	This is outside the scope of the project and is a area of further possible develop ment.	FR01, FR02, FR07
CR02	The drone **shall** be able to communicate with other drones. Send and receive data.	Enables a network of multiple drones to communicate with each other during missions.	On site tests.	This is outside the scope of the project and is a area of further possible develop ment.	SLR03, SLR08
CR03	There **shall** be one control station controlling multiple drone on the same wifi.	In order to be able to control multiple drones at the same time.	On site tests.	This is outside the scope of the project and is a area of further possible develop ment.	SLR03, SLR08
CR04	Switch time under 1s from autonomous	We want to be able to control the drone manually as	On site tests.		SLR02, SLR04, FR08,

	mode to manual control from distances of **X** km.	fast as possible for safety reasons.			
CR05	Sensors **shall** be able to connect via Bluetooth to send data.	Collect data from Bosch sensor.	On site tests.		SLR07, FR02, FR07

D. *Harborn, J. (2019). Functional Hazard Assessment. 1st ed. Västerås: OpenDrone Project*.

Functional Hazard Assessment

The FHA defines the main functions of the UAV for the intended usage of the system and describes the hazards associated with each function based on what phase the function is used in and the failure effect of each function.

Functions:

- Manual mode
  - Steering
- Mission mode
  - Automatic takeoff
  - Automatic waypoint-following
  - Automatic object/target tracking
  - Automatic landing
- Semi-autonomous mode
  - Automatic collision avoidance
- Sensor data gathering
- Data dump to base
- Heartbeat signal - System Status Signal
  - Position in X, Y, Z
  - Ground speed
  - Airspeed
  - Inner status (battery)
  - Task status
  - (Pitch, Roll, Yaw/Heading, Camera feed)
- Return to base

Flight phases:

- Takeoff
- Mission
- Hover (Idle)
- Landing

Classifications:

- Critical
  - Failure conditions that causes the loss of ability to control the UAV resulting in a crash or complete loss of functionality.
- Major

    - Failure Conditions which would reduce the capability of the UAV for example, a significant reduction in safety margins or functional capabilities.
- Minor
    - Failure Conditions which would not significantly reduce UAV functionality or safety.
- No safety effect
    - Failure Conditions that would have no effect on safety; for example, Failure Conditions that would not affect the operational capability of the UAV.

Notes regarding the safety design:

The manual flight mode is regarded as an fail-safe mechanism if mission mode steering fails.

Functional Hazard Assessment

1.Function	2. Failure Condition (Hazard Description)	3. Phase	4. Effect of Failure Condition on UAV	5. Classification	6. Reference to Supporting Material	7. Verification
Manual flight mode	Loss of manual flight mode.					
	a. Announced loss of manual flight mode	Takeoff, Mission, Hover, Landing	The UAV is unable to be manually controlled resulting in the loss of a steering option which could cause a potential system failure.	Major		Loss of system functionality FTA
	b. Unannounced loss of manual flight mode	Takeoff, Mission, Hover, Landing	The manual flight mode is unable to be activated making the UAV unable to respond to manual flight commands resulting in a potential crash.	Critical		Loss of system functionality FTA
	Erroneous manual flight mode.					
	a. Announced erroneous manual flight mode	Takeoff, Mission, Hover, Landing	The UAV controls are erroneous causing the manual control to malfunction resulting in a crash.	Critical	Due to limited resources, this area could be improved upon with further development.	
	b. Unannounced erroneous manual flight mode	Takeoff, Mission, Hover, Landing	Erroneous control in manual flight mode results in an immediate crash.	Critical	Due to limited resources, this area could be improved upon with further development.	
Mission flight mode	Loss of mission flight mode					
	a. Announced loss of mission flight mode	Takeoff, Mission, Hover, Landing	Loss of mission flight control during any phase causes the loss of the primary flight control which results in a crash.	Critical		Loss of system functionality FTA

	b. Unannounced loss of mission flight mode	Takeoff, Mission, Hover, Landing	Loss of mission flight control during any phase causes the loss of the primary flight control which results in a crash.	Critical		Loss of system functionality FTA
	Erroneous mission flight mode					
	a. Announced erroneous mission flight mode	Takeoff, Mission, Hover, Landing	Erroneous mission flight mode causes the UAV to become unable to maintain the flight controls safely.	Major	Due to limited resources, this area could be improved upon with further development.	
	b. Unannounced erroneous mission flight mode	Takeoff, Mission, Hover, Landing	Unannounced erroneous mission flight mode control causes the UAV to behave unstable which results in a crash.	Critical	Due to limited resources, this area could be improved upon with further development.	
Semi autonomous flight mode	Loss of semi autonomous flight mode					
	a. Announced loss of semi autonomous flight mode	Mission	Announced loss of semi autonomous flight mode causes the UAV to lose the ability to detect and avoid obstacles resulting in a potential collision unless mission is aborted.	Minor	Area of further investigation and development.	
	b. Unannounced loss of semi autonomous flight mode	Mission	Unannounced loss of semi autonomous flight mode causes the UAV to lose the ability to detect and avoid obstacles resulting in a potential collision.	Critical	Area of further investigation and development.	
	Erroneous semi autonomous flight control					
	a. Announced erroneous semi autonomous flight control	Mission	Causes the UAV to shut down the semi autonomous flight control resulting in the loss collision avoidance.	Minor	Area of further investigation and development.	
	b. Unannounced erroneous semi autonomous flight mode	Mission	Unannounced erroneous semi autonomous flight mode results in the UAV performing avoidance maneuvers when no object is close to the UAV.	Major	Area of further investigation and development.	
Mission sensor data gathering	Loss of mission sensor data gathering					
	a. Announced loss of mission sensor data gathering	Mission	Announced loss of sensor data gathering results in inability to complete the mission given by the MMT.	Major	Area of further investigation and development.	
	b. Unannounced loss of mission sensor data gathering	Mission	Unannounced loss of sensor data gathering causes the UAV to try to perform the mission with faulty equipment resulting in a mission failure.	Major	Area of further investigation and development.	
Data transmission to base/MMT	Loss of ability to dump collected data to base/MMT					
	a. Announced loss of ability to	Landing	UAV fails to transmit collected mission	Minor		No design

	transmit collected data to base/MMT		data to base/MMT which results in the system alerting the MMT the data could not be transmitted.			decisions.
	b. Unannounced loss of ability to transmit collected data to base/MMT	Landing	UAV fails to transmit collected mission data to base/MMT which results in the sensor data not being transmit to base, losing the data.	Major		No design decisions.
System Status Signal	Loss of system status signal					
	a. Announced loss of system status signal	Takeoff, Mission, Hover, Landing	The system fails to send a status signal resulting in a the UAV returning to base.	Critical		A3PT013
	b. Unannounced loss of system status signal	Takeoff, Mission, Hover, Landing	The MMT does not receive any status signal resulting in the UAV returning to base.	Critical		A3PT013
	Erroneous system status signal					
	a. Announced erroneous system status signal	Takeoff, Mission, Hover, Landing	The system sends erroneous values to the MMT which results in the MMT triggering the return to base function.	Critical	Area of further investigation and development.	
	b. Unannounced erroneous system status signal	Takeoff, Mission, Hover, Landing	The MMT receives erroneous values which results in the MMT triggering the return to base function.	Critical	Area of further investigation and development.	
Return to base	Loss of Return to Base					
	a. Announced loss of Return to Base	Takeoff, Mission, Hover, Landing	The system is unable to trigger the Return to Base function which results in the inability to manually override the UAV controls.	Major		Loss of system functionality FTA
	b. Unannounced loss of Return to Base	Takeoff, Mission, Hover, Landing	The system is unable to receive or execute the Return to Base command which results in the inability to manually override UAV controls.	Major		Loss of system functionality FTA
	Erroneous Return to Base					
	a. Announced erroneous Return to Base	Takeoff, Mission, Hover, Landing	The Return to Base function is triggered from faulty reasons resulting in the UAV returning to base cancelling the mission.	Major	Due to limited resources, this area could be improved upon with further development.	
	b. Unannounced erroneous Return to Base	Takeoff, Mission, Hover, Landing	The system can erroneously trigger the Return to Base function without sufficient reason resulting in a mission failure.	Critical	Due to limited resources, this area could be improved upon with further development.	

E. Harborn, J. (2019). *Preliminary System Safety Assessment*. 1st ed. Västerås: OpenDrone Project.

Preliminary System Safety Assessment

The PSSA is used to complete the failure conditions list and demonstrate the requirements set on the system for each of the hazards. The process identifies the fail-safe concepts and design choices made. Displaying the requirements and derived requirements generated.

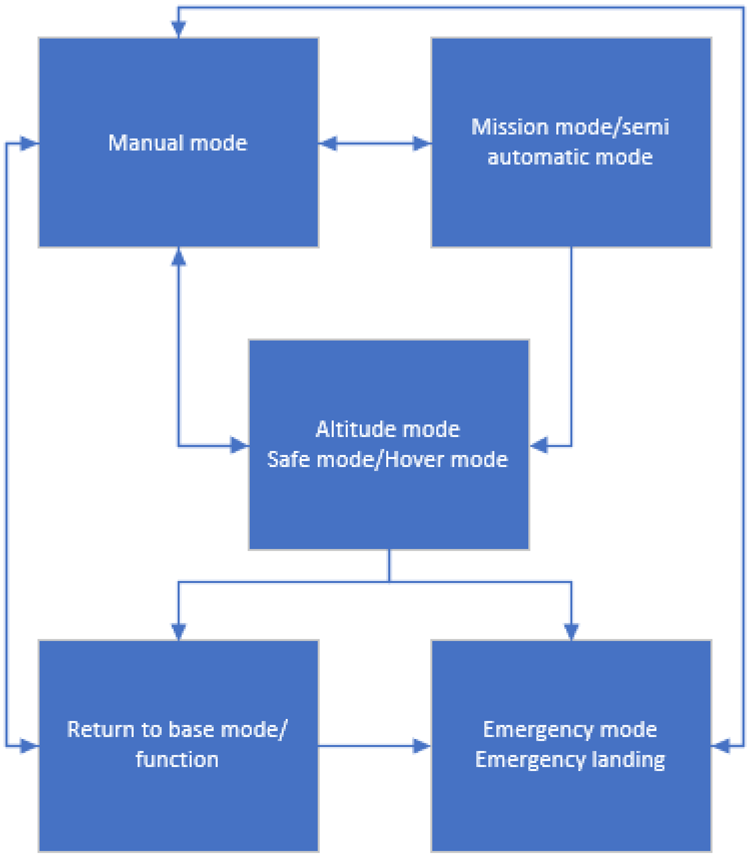
Inputs:

- Announced loss of manual flight mode - Major
- Unannounced loss of manual flight mode - Critical
- Announced erroneous manual flight mode - Critical
- Unannounced erroneous manual flight mode - Critical
- Announced loss of mission flight mode - Critical
- Unannounced loss of mission flight mode - Critical
- Announced erroneous mission flight mode - Major
- Unannounced erroneous mission flight mode - Critical
- Unannounced loss of semi autonomous flight mode - Critical
- Unannounced erroneous semi autonomous flight mode - Major
- Announced loss of sensor data gathering - Major
- Unannounced loss of sensor data gathering - Major
- Unannounced loss of ability to transmit collected data to base/MMT - Major
- Announced loss of system status signal - Critical
- Unannounced loss of system status signal - Critical
- Announced erroneous system status signal - Critical
- Unannounced erroneous system status signal - Critical
- Announced loss of return to base - Major
- Unannounced loss of return to base - Major
- Announced erroneous return to base - Major
- Unannounced erroneous return to base - Critical

Design decisions flight modes:

1. Manual mode: highest priority - the user shall be able to always override the other modes
2. Mission mode: the drone controls all actions
3. Altitude mode: the drone is only able to hold the altitude “hover” on its location
4. Return mode: the drone will fly back to the launch area cancelling any other actions.
5. Emergency mode: the drone will commence an emergency landing at the current location

Graceful Degradation Block Diagram



Manual mode can overtake any other mode if it is desired to. Mission mode will degrade down to altitude mode if it fails. The drone will stay in altitude mode when mission mode has failed until the controller decides the next action (continue in manual mode, return to base mode, emergency mode or retry mission mode). Return to base mode can be degraded down to emergency mode if it fails. Only the manual mode can cancel either return to base or emergency mode. [1]

PSSA Table

Safety Requirement	Design Decisions	Remarks
1. Announced loss of manual flight mode in any phase shall be less than X per mission duration.	A watchdog will be implemented to detect if inputs stop from the manual control mode.	Manual flight mode is designed as an alternate flight mode independent from the other flight modes with all inputs from the radio controller.
2. Unannounced loss of manual flight mode in any phase shall be less than X per mission duration. - Critical	A watchdog will be implemented to detect if inputs stop from the manual control mode.	If this occurs the UAV will swap to altitude mode.

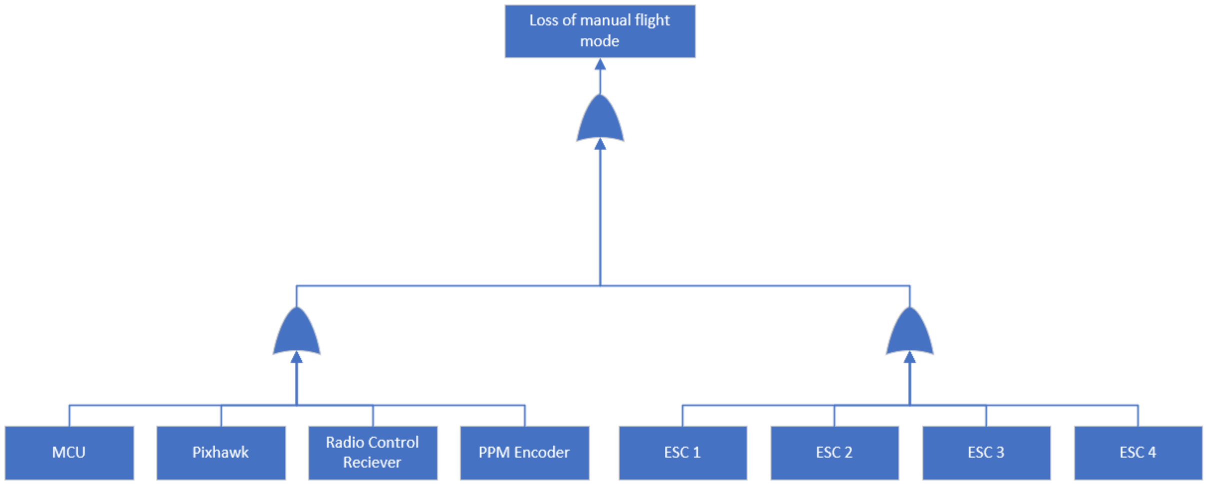
3. Announced erroneous manual flight mode in any phase shall be less than X per mission duration. - Critical	Due to limited resources, this area could be improved upon with further development.	**See Erroneous flight mode decisions**
4. Unannounced erroneous manual flight mode in any phase shall be less than X per mission duration. - Critical	Due to limited resources, this area could be improved upon with further development.	**See Erroneous flight mode decisions**
5. Announced loss of mission flight mode in any phase shall be less than X per mission duration. - Critical	A watchdog will be implemented to detect when control/steering signals are no longer sent and then swap to altitude mode and hover in place.	If this occurs the UAV will swap to altitude mode.
6. Unannounced loss of mission flight mode in any phase shall be less than X per mission duration. - Critical	A watchdog will be implemented to detect when control/steering signals are no longer sent and then swap to altitude mode and hover in place.	If this occurs the UAV will swap to altitude mode.
7. Announced erroneous mission flight mode in any phase shall be less than X per mission duration. - Major	Due to limited resources, this area could be improved upon with further development.	**See Erroneous flight mode decisions**
8. Unannounced erroneous mission flight mode in any phase shall be less than X per mission duration. - Critical	Due to limited resources, this area could be improved upon with further development.	**See Erroneous flight mode decisions**
9. Unannounced loss of semi autonomous flight mode in any phase shall be less than X per mission duration. - Critical	This is outside the scope of the project and is a area of further possible development.	Area of further investigation and development.
10. Unannounced erroneous semi autonomous flight mode in any phase shall be less than X per mission duration. - Major	This is outside the scope of the project and is a area of further possible development.	Area of further investigation and development.
11. Announced loss of mission sensor data gathering in any phase shall be less than X per mission duration. - Major	Due to costs it is deemed that no design decisions will be implemented.	Area of further investigation and development.
12. Unannounced loss of sensor data gathering in any phase shall be less than X per mission duration. - Major	Due to costs it is deemed that no design decisions will be implemented.	Area of further investigation and development.
13. Unannounced loss of ability to transmit collected data to base/MMT in any phase shall be less than X per mission duration. - Major	To prevent data loss the UAV shall save current mission on local storage.	

14. Announced loss of system status signal in any phase shall be less than X per mission duration. - Critical	A watchdog will be implemented, if the signal is not regained after 10 seconds the system triggers the return to base command.	
15. Unannounced loss of system status signal in any phase shall be less than X per mission duration. - Critical	Same as above in requirement 14.	
16. Announced erroneous system status signal in any phase shall be less than X per mission duration. - Critical	Very important but outside the scope of the project.	Area of further investigation and development.
17. Unannounced erroneous system status signal in any phase shall be less than X per mission duration. - Critical	Very important but outside the scope of the project.	Area of further investigation and development.
18. Announced loss of return to base function in any phase shall be less than X per mission duration. - Major	A watchdog will be implemented to detect if inputs stop from the return to base mode.	
19. Unannounced loss of return to base function in any phase shall be less than X per mission duration. - Major	A watchdog will be implemented to detect if inputs stop from the return to base mode.	
20. Announced erroneous return to base function in any phase shall be less than X per mission duration. - Major	Due to limited resources, this area could be improved upon with further development.	**See Erroneous flight mode decisions**
21. Unannounced erroneous return to base function in any phase shall be less than X per mission duration. - Critical	Due to limited resources, this area could be improved upon with further development.	**See Erroneous flight mode decisions**

As we have no information regarding failure rates of the items and adapted the classifications to fit the UAVs mission criticality rather than the DAL from the ARP4754 and ARP4761, no failure rates will be set in the safety requirements.

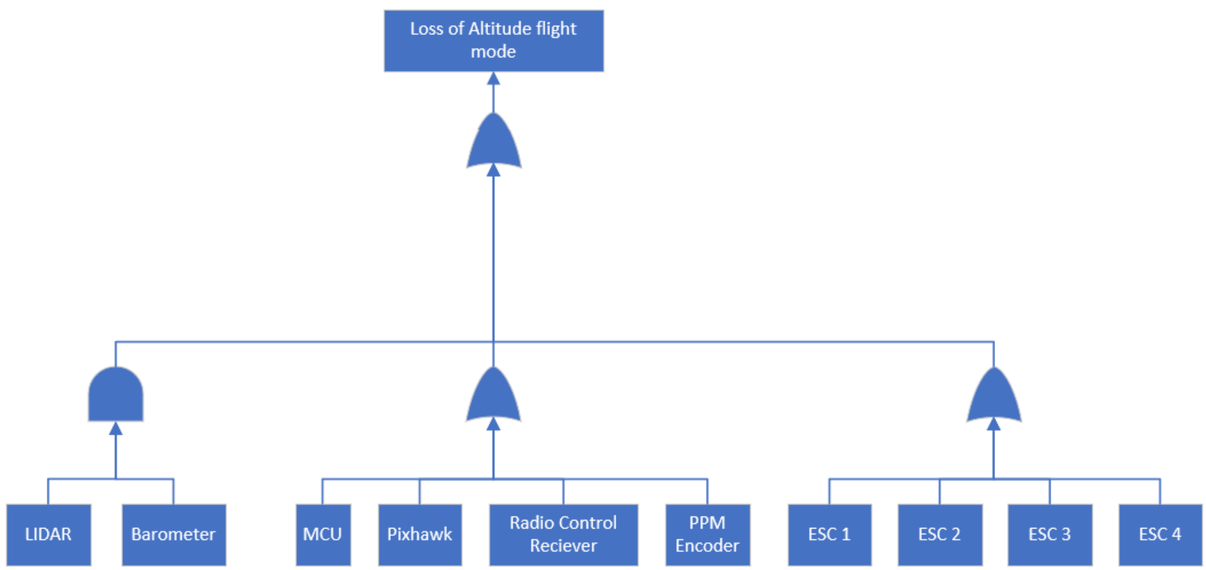
Fault Tree Analysis:

Loss of Manual flight mode



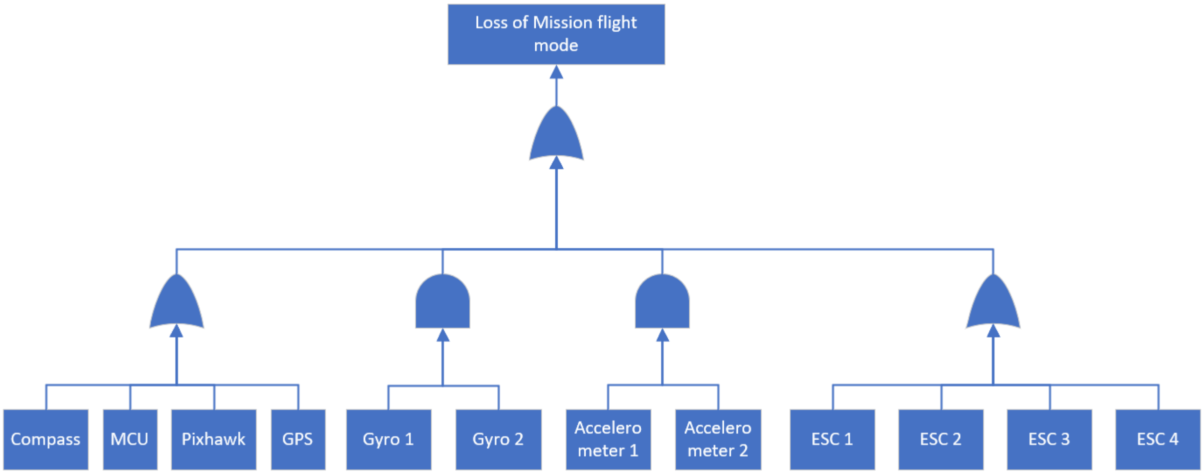
This fault tree displays the Manual flight mode.

Loss of Altitude flight mode



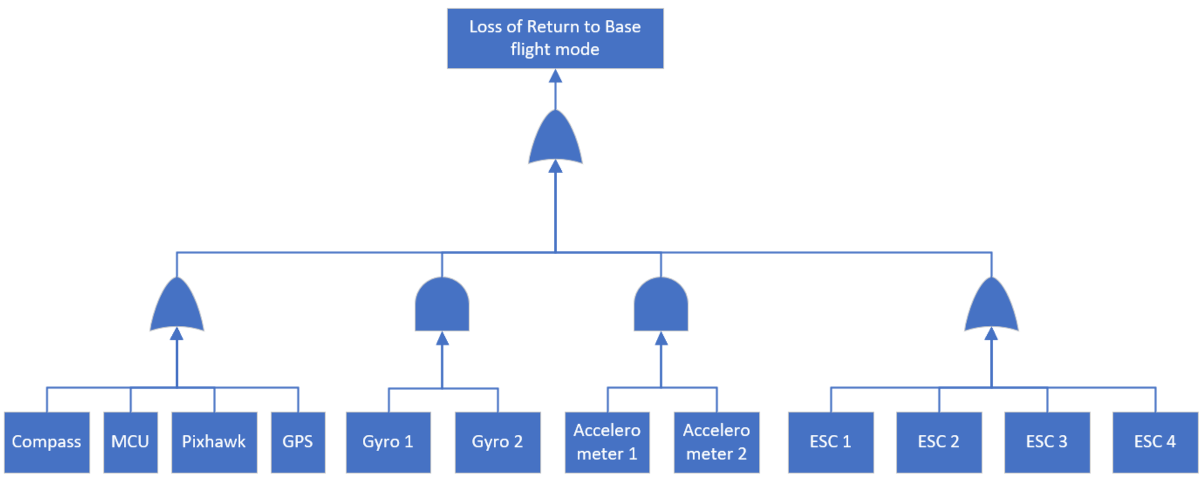
This fault tree displays the Altitude flight mode.

Loss of Mission flight mode



This fault tree displays the Mission flight mode.

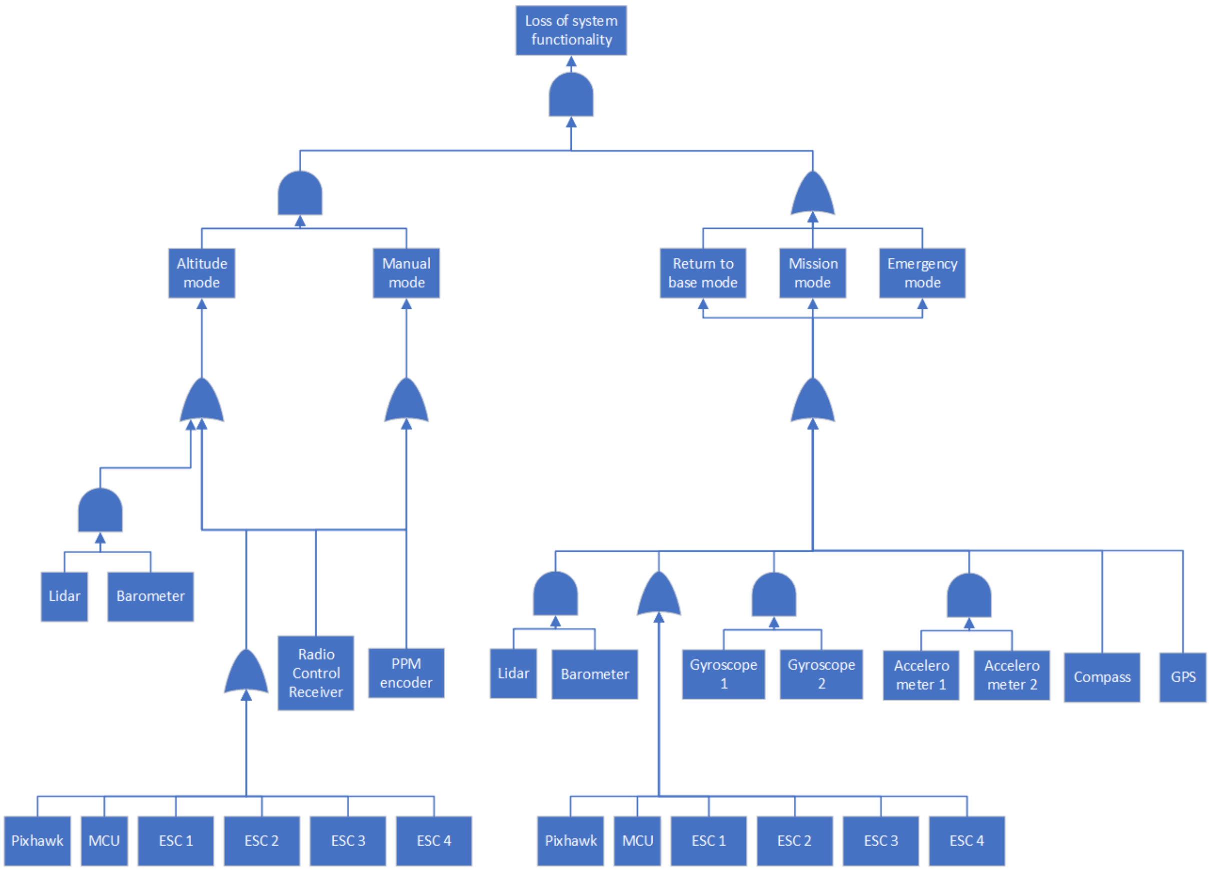
Loss of Return to Base Mode flight mode



This fault tree displays the Return to Base flight mode.

Loss of System Functionality

It is important to notice that the lowest part of the tree including the Pixhawk, MCU and the ESCs are duplicated and only one of each of these items exists, they are duplicated in two parts to make reading this fault tree easier.



This fault tree shows all of the flight modes with the fail-safe design between manual and autonomous flight modes.

Erroneous flight mode decisions:

The fault tree analysis reveals that the UAV is extremely vulnerable to erroneous conditions regarding the different flight modes. Almost every part covered in the fault tree analysis contributes enough to instantly cause the entire system to act erroneously. The UAV is built without any redundancy of hardware components as adding more hardware on the UAV would be impractical from a mission viewpoint, it would add more weight, power consumption and would require more space to mount the equipment, increasing the overall size. Adding redundancy would make it possible to design fail-safe computations methods (e.g COM-MON architecture) or even fault tolerant designs (e.g voting system) but the budget and time required for design choices like these are not sufficient for this project and can be hard to achieve in an UAV project as space and weight are key parameters for functionality.

Adding redundancy for hardware does not fulfill all the requirements for erroneous or even loss of cases as the UAV is built as a quadcopter (four arms) and the loss of/erroneous function of a motor or accelerometer leading to a motor will cause a crash since all four arms are essential for flying. There are algorithms that can compensate for a non functional arm but for erroneous operation its hard to verify that an algorithm is a verifiable solution. A redesign of the UAV from a quadcopter to a hexacopter or octocopter might be required to compensate for erroneous behavior of different parts. This would also require a bigger budget and more time which this project does not have.

There are pre-flight checks that the Pixhawk runs before arming the quadcopter. These checks compare the values of different sensors to make sure the sensor data is correct, this acts as a fail safe to identify that all sensors are correct and no sensor is erroneous, this is a way of reducing the chance that you fly with an erroneous sensor. However there is still a possibility of sensor data turning erroneous in-flight and this is something we cannot prevent.[2]

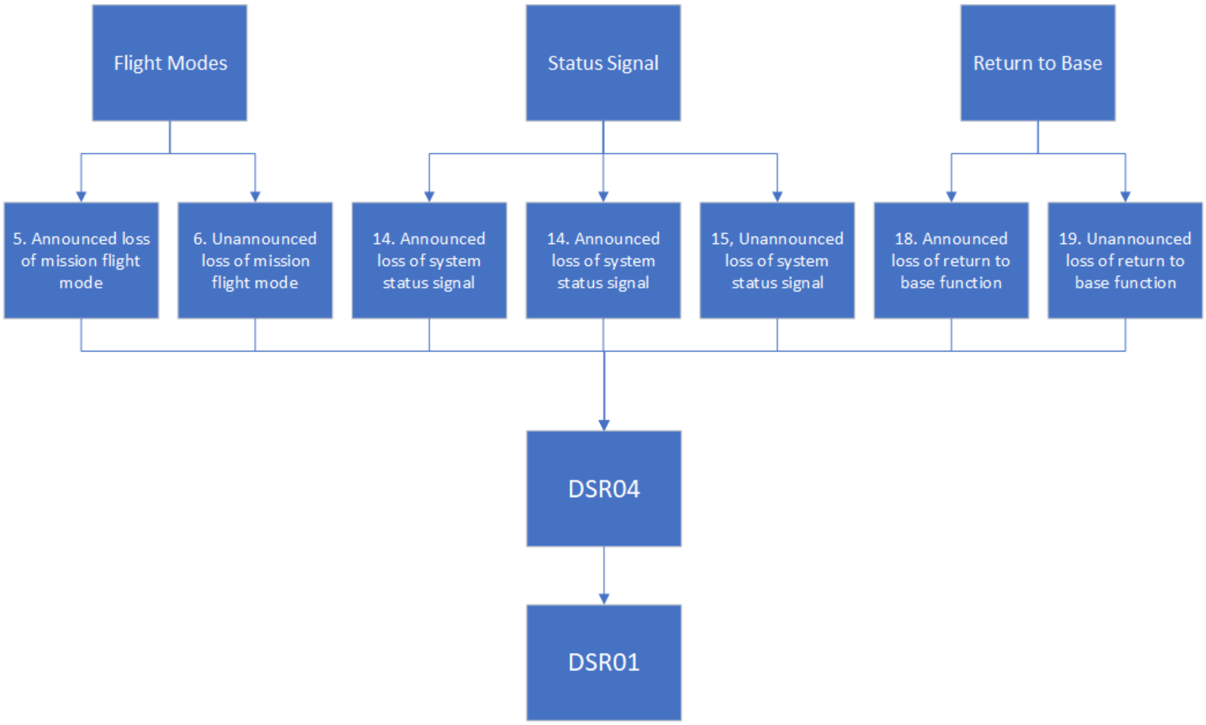
The fail-safe design choices between the manual flight modes and autonomous flight modes gives the user a choice to manually swap to manual flight mode but no automatic action is taken to swap to altitude or manual flight mode. If the erroneous behavior could be identified in flight this method could work to prevent a crash.

Derived safety requirements

ID	Requirement	Rationale	MoC	Notes
DSR01	Altitude flight mode shall be implemented to serve as a degraded flight mode.	Altitude flight mode acts as an automatic fail-safe when signal is lost.	Design analysis.	
DSR02	Manual flight modes shall be independent from the other flight modes.	Manual flight mode acts as a fail safe for the other flight modes with the only inputs coming from the radio controller.	Design analysis.	
DSR03	It shall be possible to swap between all flight modes in all phases.	The UAV needs to be able to swap between flight modes to be able to complete the missions with safety margins.	Flight tests.	

DSR04	A watchdog shall be implemented to determine if any flight mode stops transmitting commands to the Pixhawk.	A design requirement developed to achieve a safe way of detecting loss of events.	Design analysis.	

Derived Safety Requirements Traceability Tree



References:

[1] Docs.px4.io. (2019). *Flight Modes · PX4 v1.9.0 User Guide*. [online] Available at: https://docs.px4.io/v1.9.0/en/flight_modes/ [Accessed 12 Dec. 2019].
[2] Docs.px4.io. (2019). *Preflight Checks · PX4 v1.9.0 User Guide*. [online] Available at: https://docs.px4.io/v1.9.0/en/flying/pre_flight_checks.html [Accessed 12 Dec. 2019].

F. Harborn, J. (2019). System Safety Assessment. 1st ed. Västerås: OpenDrone Project.

System Safety Assessment

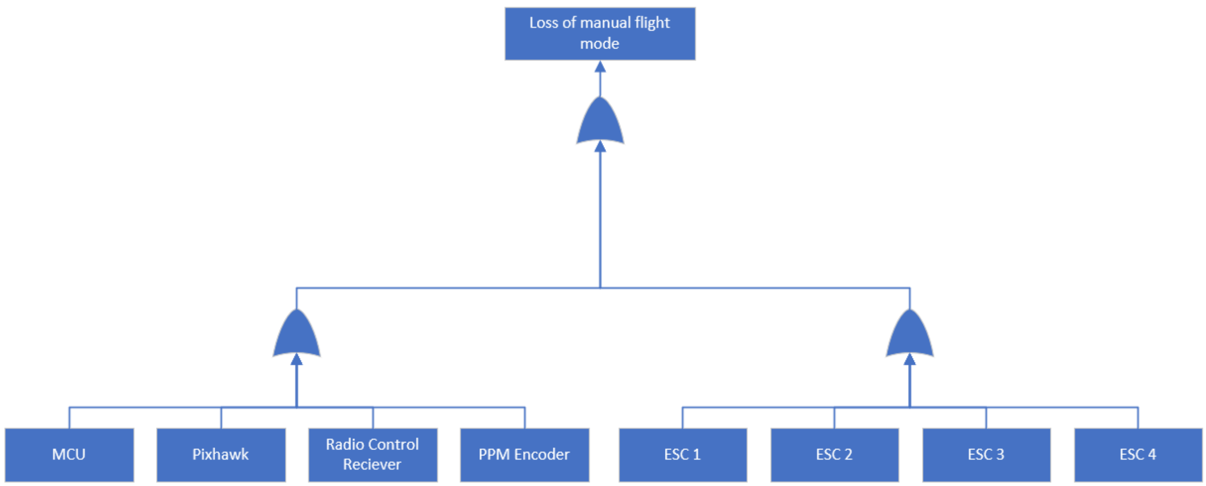
The System Safety Assessment is complemented by the FHA, PSSA, FTAs and FMEA, this document takes all the previous work into account when making a final assessment of the safety of the system.

Failure Mode Effects Analysis

Function Name	Function Code	Failure Mode	Mode Failure Rate	Flight Phase	Failure Effect	Detections Methods	Comments
Pixhawk	F001	Loss of Pixhawk function.	N/A	Takeoff, Mission, Hover, Landing	The flight computer is lost causing the UAV to crash.	None	As we only have one flight computer we have no detection methods or ways to prevent this from happening.
Pixhawk	F002	Erroneous Pixhawk function.	N/A	Takeoff, Mission, Hover, Landing	The flight computer issues erroneous commands causing unpredictable behavior of the UAV possibly resulting in a crash.	None	As we only have one flight computer we have no detection methods or ways to prevent this from happening.
Motor Control Unit	F003	Loss of Motor Controls.	N/A	Takeoff, Mission, Hover, Landing	Power to the motors are lost causing the UAV to crash	None	As we only have one Motor Control Unit no detection methods are implemented.
Motor Control Unit	F004	Erroneous Motor Controls.	N/A	Takeoff, Mission, Hover, Landing	The MCU sends erroneous signals to the motors causing erroneous behavior possibly resulting in a crash.	None	As we only have one Motor Control Unit no detection methods are implemented.
Electric Speed Controller	F005	Loss of Electric Speed Controller function.	N/A	Takeoff, Mission, Hover, Landing	Losing an Electric Speed Controller results in the UAV crashing due to the loss motor power signal.	None	There are a total of four Electric Speed Controllers and losing any of them equals losing a motor which will result in a crash.
Electric Speed Controller	F006	Erroneous Electric Speed Controller function.	N/A	Takeoff, Mission, Hover, Landing	The UAV will be uncontrollable since the Electric Speed Controller feeds erroneous data to the motor resulting in a potential crash.	None	There are a total of four Electric Speed Controllers and if any of them act erroneously a motor will do the same which can result in a crash.

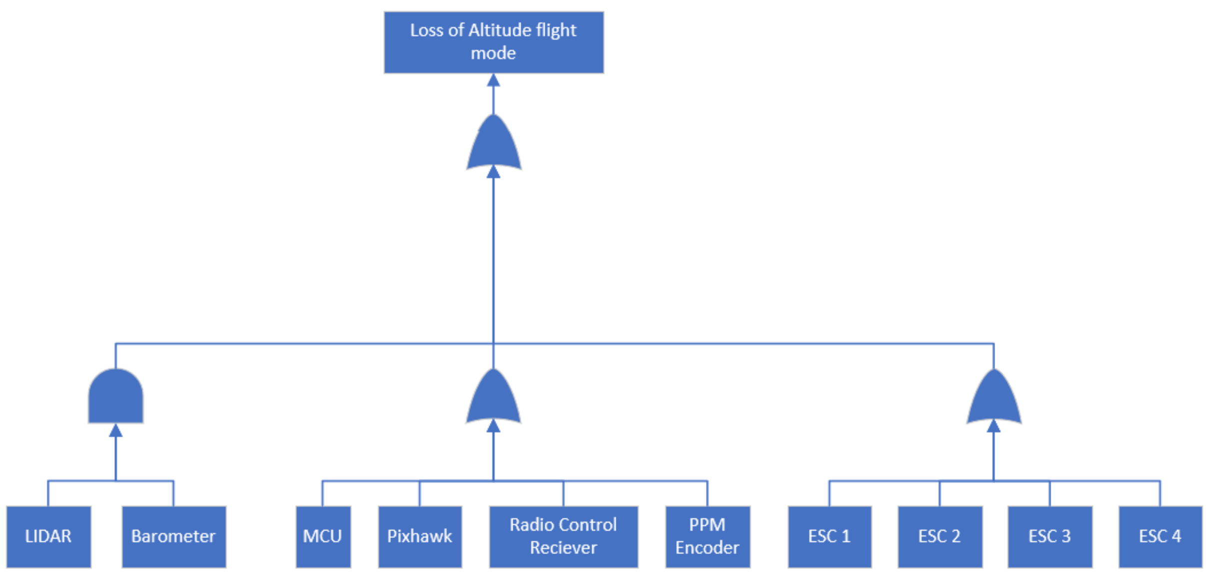
Fault Tree Analysis

Loss of Manual flight mode



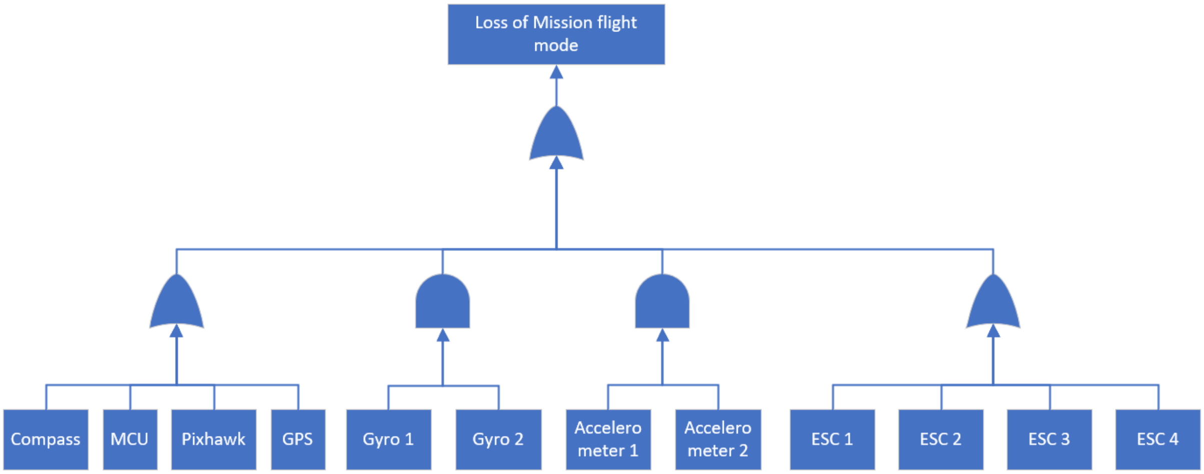
This fault tree displays the Manual flight mode.

Loss of Altitude flight mode



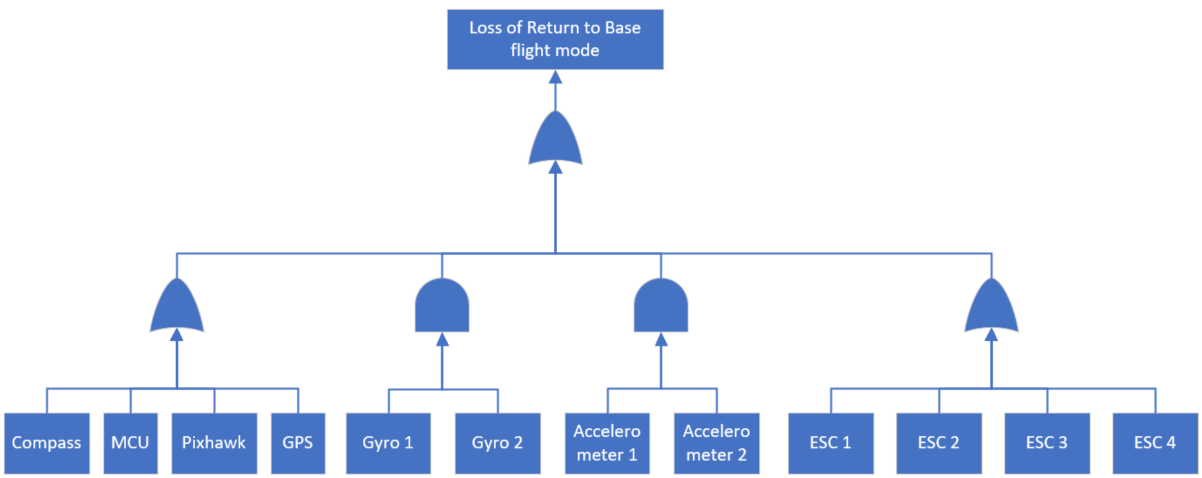
This fault tree displays the Altitude flight mode.

Loss of Mission flight mode



This fault tree displays the Mission flight mode.

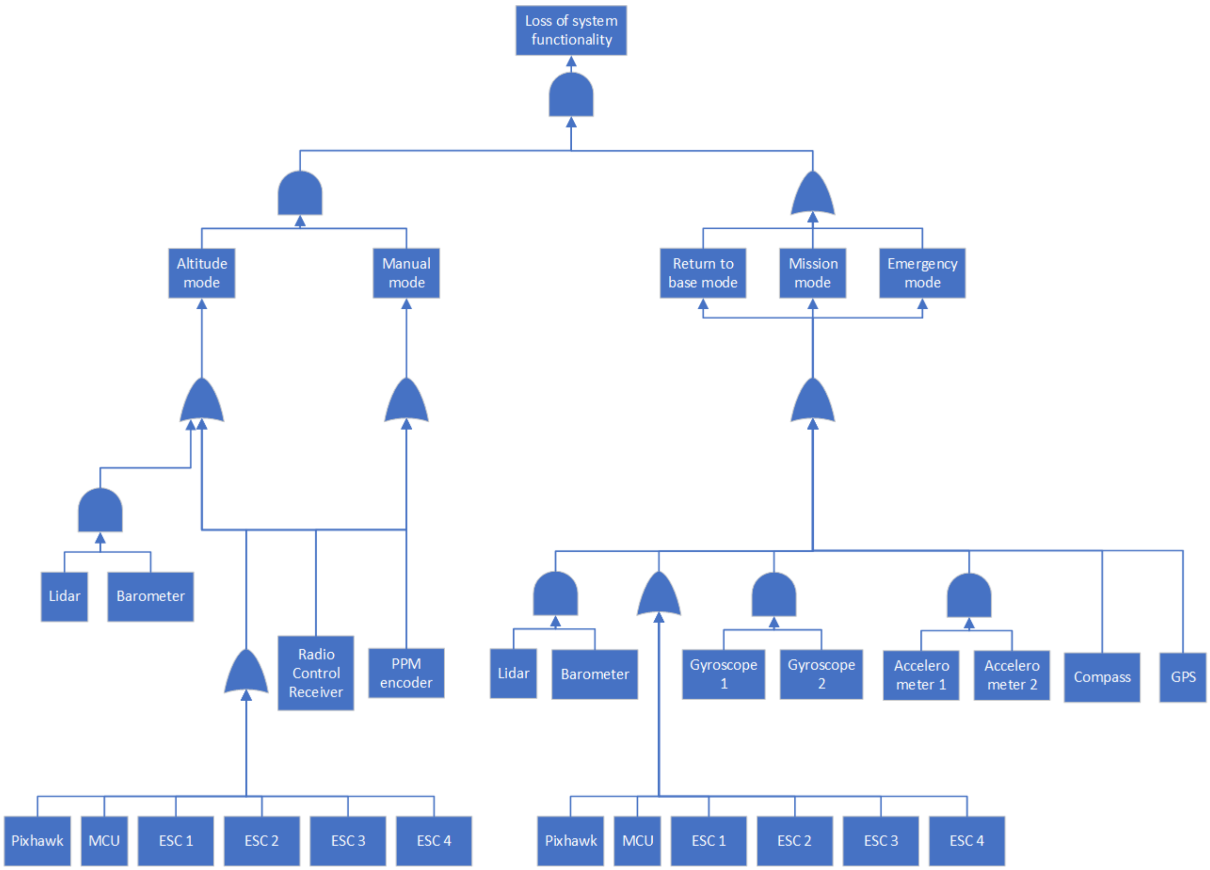
Loss of Return to Base flight mode



This fault tree displays the Return to Base flight mode.

Loss of system functionality

It is important to notice that the lowest part of the tree including the Pixhawk, MCU and the ESCs are duplicated and only one of each of these items exists, they are duplicated in two parts to make reading this fault tree easier.



This fault tree shows all of the flight modes with the fail-safe design between manual and autonomous flight modes.

Single points of failure

The system has proven to have multiple single points of failure which greatly reduces the safety of the UAV. As shown in the Loss of system functionality FTA and further developed in the FMEA the single points of failure are the Pixhawk, Motor Control Unit and the four Electronic Speed Controllers. There are two more single points of failure which are not taken in to consideration which are the four motors and four propellers these are considered to have a very low failure rate and are therefore not taken in to consideration (they would only make the FTA harder to read and have the same failure outcomes as the ESCs).

The system is designed with a fail safe architecture as can be seen in the Loss of system functionality FTA where one part is the automatic/autonomous flight modes which are the Mission, Return to Base and Emergency flight modes. The other half is the Manual and Altitude flight modes which act as an alternative in case of the other automatic/autonomous modes fail. This solves the Loss of scenarios but Erroneous scenarios are unable to be covered while in-flight. Erroneous scenarios are however detected in the pre-flight checks conducted by the Pixhawk, which compares the sensor values to make sure they are showing the same result. If not this is detected and warns the user arming the UAV that there is an inconsistency in sensor value, not allowing the UAV to arm. Because of this we can partially detect Erroneous flight mode scenarios but not fully.

G. *Documents Description. (2019). 1st ed. Västerås: OpenDrone Project.*

Small description of all documents in open drone project

1 Intro 62

2 Documents 62

2.1 3D print 62
2.2 HW diagram Linus 62
2.3 Budget 62
2.4 Project directive (SV & Eng) 62
2.5 Requirements 62
2.6 FHA 62
2.7 PSSA 62
2.8 Fault-Trees 63
2.9 SSA 63
2.10 DVM 63
2.11 Test plan & protocol 63
2.12 Traceability 63
2.13 Validation 63
2.14 Verification 63
2.15 Process Assurance Deknas 63
2.16 SW Tutorials and notes Dino (Add the other files) 63
2.17 Drone Version Description - Text 64
2.18 Drone Version Description - Powerpoint 64
2.19 Logs **Error! Bookmark not defined.**
2.20 Report (Finishing in the end) 64

3 Practical work 64

3.1 Vaxholm 64
3.2 Testing 64
3.3 Supporting 64
3.4 3D Printing 65
3.5 Codes 65
3.5.1 collectData.py 65
3.5.2 findAddress.py 65
3.5.3 findCharacteristics.py 65
3.5.2 server.py 65
3.5.3 client.py 65

1 Intro

By using this page of document description, we can analyze the workflow and workload in OpenDrone group at MDH in courtse FLA400 HT19. We will exclude assignments from the course FLA400 and seminars. This document also include information about every document so the reader easier can find the correct information.

2 Documents

2.1 3D print

Hours laid tot: 1h

Rasmus: 1h
Discussion about the best infills to 3D print and define some parts of the printing process.

2.2 HW diagram Linus

100% Linus

2.3 Budget

Shows the budget of the project with income statement and project budget. The income statement is always updating.

2.4 Project directive (SV & Eng)

Shows the purpose of the project with decided roles and smaller task description. The project directive is both in english and swedish, and include some project description and plans.

2.5 Requirements

Here you'll find all requirements set upon the project, their traceability to each other as well a tree portraying the traceability in a more illustrative fashion.

Can be found in: OpenDroneFolder/System Safety Case

2.6 FHA

The Functional Hazard Assessment shows the functions used by the UAV and the hazards associated with each of the functions, classifications and effects of the failures.

Can be found in: OpenDroneFolder/System Safety Case

2.7 PSSA

The Preliminary System Safety Assessment is built from the inputs of the FHA where safety requirements and goals are included. It also includes the Graceful Degradation block diagram, Fault Tree Analysis, Erroneous flight mode decisions, Derived Safety Requirements and Derived Safety Requirement traceability tree.
Can be found in: OpenDroneFolder/System Safety Case

2.8 Fault-Trees

The Fault Trees are included in the PSSA and SSA.

Can be found in: OpenDroneFolder/System Safety Case/Felträd

2.9 SSA

The System Safety Assessment is the conclusion of the safety case of the system summing all the previous assessments displaying the final safety assessment of the system.

Can be found in: OpenDroneFolder/System Safety Case

2.10 DVM

The Design Verification Matrix tracks the traceability of the requirement testing.

2.11 Test plan & protocol

Includes a test plan describing the testing process of the project, this also includes examples of how the tests shall be documented. In this document you'll find all the tests and their results.

Linus : 60%
Daniel: 20%
Armin: 20%

2.12 Traceability

Traceability is located in the Requirement document, see "Requirements" section.

2.13 Validation

In this document you will find the validation plan, description of the validation methods, the requirement reviews performed on a few requirements and the validation summary.

2.14 Verification

Here you will find the verification plan, description of the verification methods and a verification summary.

2.15 Process Assurance Deknas

Describes the Process development plan. An assurance strategy to check that the plan has been followed according to the plan. Deviations from the plan and corrective actions for the deviations. Problem report discussing problems occurred during the development process and corrective actions for solving the problems.

2.16 SW Tutorials and notes Dino (Add the other files)

asdasd

2.17 Drone Version Description - Text

Daniel 10%

Describes Alpha drone and examples of development phases in the case of physical development, such as PID. Also shows some hardware development.

2.18 Drone Version Description - Powerpoint

Description for every drone and every version the project has occurred. Some drones were created and then disposed because of lack of space or unplanned creation.

2.19 Logs

The folders contain some logs which could help understanding how the drone functions but also how QGC works. It also contains a file which describes how to use the logs available for the Pixhawk.

Linus 100%

2.20 Report (Finishing in the end)

Main report to the course of FLA400.

3 Practical work

3.1 Vaxholm

Hours laid tot: 16h (2 work days)

Rasmus: 16h
Linus: 16h
Erik: 16h

Worked att Vaxholm for two days with creation of Epoxi-carbonfiber.

3.2 Testing

Practical testing out in the fields or inside of C2

3.3 Supporting

Supporting the project in different criterias; objects, places and other field where the project needs support during daily activities. This also includes go and shopping product that is needed and other administrative assignments (documentation for nextweeks workload).

3.4 3D Printing

3D printing is mostly during the night and days, where no one is around but prepearement is being done by Rasmus and the 3D printer that is in use is project leaders Rasmus own printer (Ender 3X).

3.5 Codes

3.5.1 collectData.py

This is the script that collects sensor data from the Bosch XDK. It is autostarted on the Odroid XU4 and always running in the background. It saves sensor logs to a .txt file called sensorLogs in the Open Drone folder on the Odroid.

3.5.2 findAddress.py

This script is used to find the bluetooth device address that is used in collectData.py. If you want to add more sensors or change the sensor you can copy and paste the address into collectData.py.

3.5.3 findCharacteristics.py

This script is used to find the GATT characteristics of the bluetooth device.

3.5.2 server.py

This is the Thrift server script that is used to communicate with the Mission Management Tool. It contains most of the drone functionalities such as creating missions, aborting missions, executing and monitoring missions. The most important library here is Dronekit. Dronekit is used to connect to the drone and send Mavlink messages. Commands such as reading/writing to parameters, uploading missions, executing missions, and much more can be sent to the drone with Dronekit.

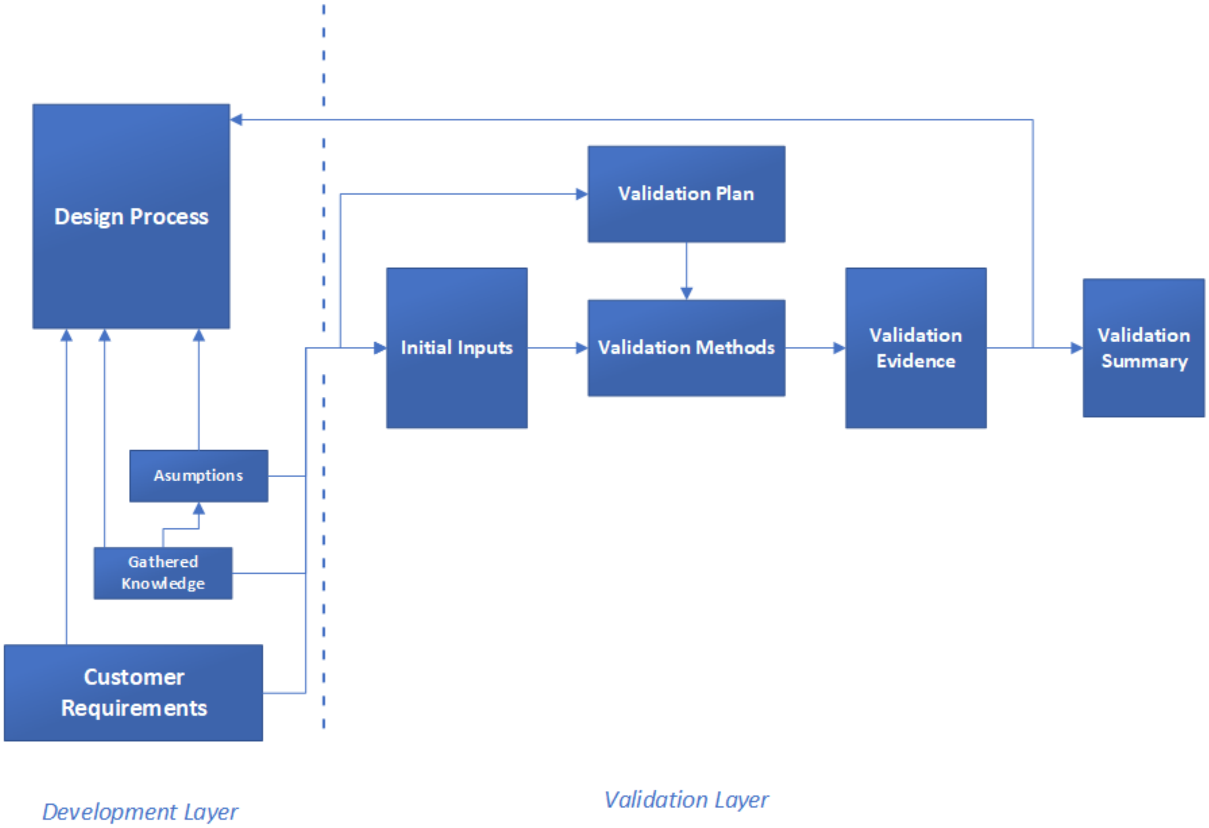
3.5.3 client.py

This is a test script for Thrift. It’s a Thrift client written in Python. We used it to test the Thrift communication. It can be used to test the Thrift server.

H. Nordvall, D. (2019). Validation 1st ed. Västerås: OpenDrone Project, p.1-3.

Validation Plan

The validation process will follow the described process in ARP4754A, adapted to the limitations of this project. Unfortunately, due to lack of manpower, independence between the validation team and design team is not achievable as most members have participated in both processes.



1. *Validation Process Diagram*

Method

The validation process will utilize traceability, analysis and requirement reviews to prove the validity of the requirement; e.g. the correctness and completeness of the requirement.

Traceability will be shown through tables and a tree representing the connection that lower-level requirements have to higher-level requirements. Every requirement has a validation column which shows what higher-level requirements they derive from, this is then tracked in a traceability tree. With this analysis every requirements completeness will be demonstrated.

Some requirements may need further analysis of their completeness. This is performed by the use of a rationale for every requirement. As the requirement may not be self-evident to its existence, this rationale will prove the reasoning and thereby prove the completeness of the requirement. These rationales shall be included in the Requirement Review.

A Requirement Review will be performed to further prove both the completeness and correctness of the requirement. These in a realistic project would have to be performed on all requirements, however, in this project this will be performed on only a handful as they are quite time consuming and only proof-of-concept is necessary to achieve the project goals.

The Requirement Review will answer the following questions:

- Is the requirement unambiguous?
- Is the requirement contradicting any other requirement?
- Is the requirement in-line with the needs of the customer, user or maintainer or does the requirement have traceability to any high-level function, system design or safety analysis?
- Does the requirement follow correct, appropriate and consistent grammar?
- Is it identifiable as a requirement?
- Is the requirement redundant?
- Is it physically possible to meet the requirement?
- Is the requirement verifiable?
- Is the requirement supported by a rationale?
- Safety analysis specific questions
    - Are all system failure conditions identified and classified correctly?
    - Is the requirement derived from an identified hazard?
    - Does the requirement mitigate the identified hazard?

A validation summary shall be provided where the evidence gathered during the validation process shall be presented. This should give an overview of the evidence from the Traceability tree and the Requirement Reviews in regards to the customers demands. It shall also include, if any, deviations from the Validation plan that were necessary to complete the validation process.

Completeness & Correctness

Completeness is the degree of which a set of requirements meets the needs of customer, maintainer, user, regulators, system and item developers.

Correctness is the degree to which the individual requirement is unambiguous, verifiable, consistent, concise, feasible and traceable. The requirements follow The Easy Approach to Requirements Syntax (EARS).

Validation Summary

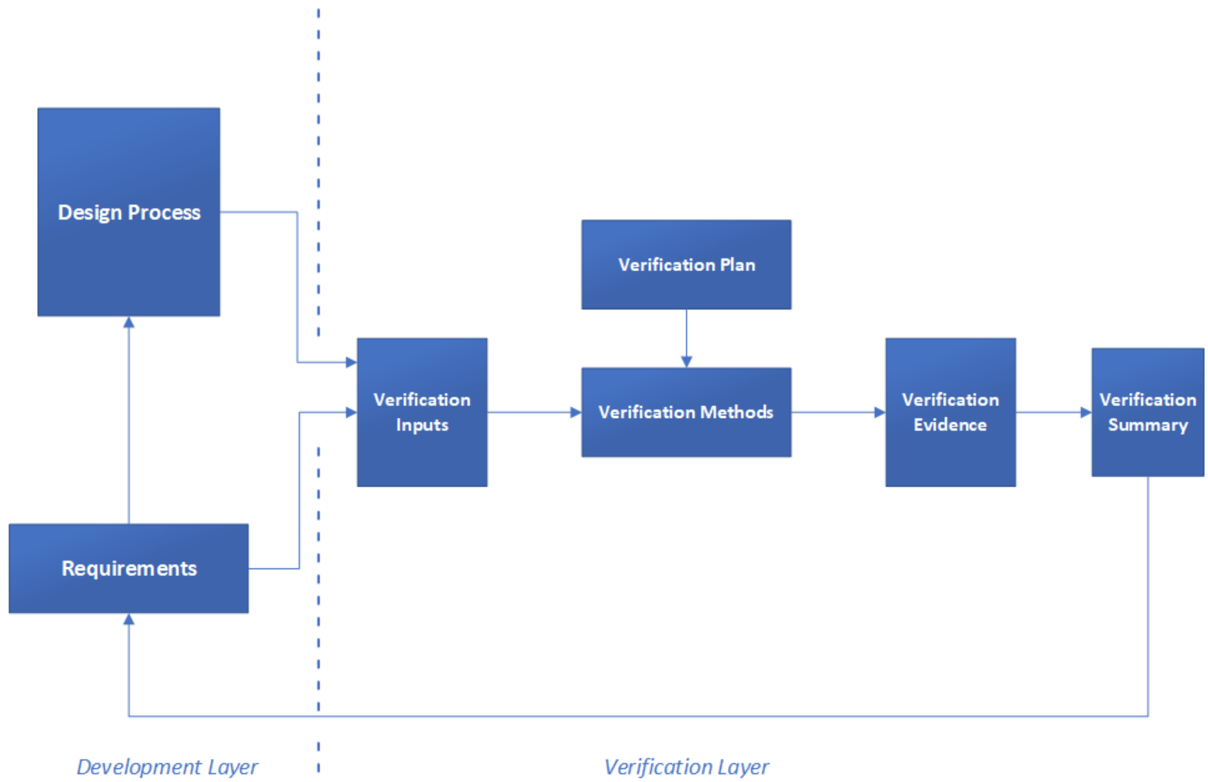
Based upon given information from the Traceability Tree and the Requirement Reviews the requirements are deemed sufficient in completeness and correctness to provide the customer with the product that they ordered.

No deviations from the Validation Plan was necessary to complete the validation process.

I. *Nordvall, D. Verification Plan 1st ed. Västerås: OpenDrone Project.*

Verification Plan

The verification process is designed and monitored by the V&V teamlead who has final say and responsibility for the process. Every method described below can be performed by any member of the project, but the V&V teamlead is responsible to ensure the viability of the tests and that the documentation is formalized in an appropriate manner. Unfortunately, due to the small size of the group, the independence between the design and verification activities can't be guaranteed as most members have been part of both processes.



1. Verification Process Diagram

The verification process begins with taking inputs from the requirements and design process, these inputs are the requirements that need verification, and the produced product. This is then examined through the Verification Methods, described in the Verification Plan and the method for Means of Compliance is described in the "Requirement" document that is a input to the process. These evaluations will produce Verification Evidence that will be used in the Verification Summary to determine to compliance of the produced system to the requirements.

The verification process will utilize three different methods for verification; Analysis, Tests & a Coverage Analysis.

Every verification method performed will result in evidence, this evidence shall be classified according to the following parameters; Failed, Partial Approval & Approved.

- Failed: The system fails to prove that the targeted requirement(s) are met by the system.

- Partial Approval: The system proves compliance with parts of the requirement, but not the complete requirement.
- Approved: The system proves to fulfill the requirement.

The evidence produced by the verification methods shall be tracked in the Design Verification Matrix (DVM) provide traceability over the verification process.

Method

Analysis

An analysis provides evidence of compliance through detailed examination of the system or item. This should include functionality, performance & safety. Also an evaluation of the systems expected performance in normal and abnormal conditions.

Tests

Testing consists of two potential paths; physical tests or modelling. Every test will provide the following information: Description of test, test operative(s), relevant circumstantial information, requirements traceability & test results

Coverage Analysis

Coverage analysis will prove the traceability of the requirements from both a design and verification perspective. The requirements shall have a traceability tree proving the link between lower-level and higher-level requirements and the verification evidence shall have traceability to requirements through the DVM.

Verification Summary

a. A reference to the verification plan and a description of any significant deviations from the plan.
b. The verification matrix
c. a description of any open problems reports and an assessment of the related impact on safety.
d. Verification Summary?

J. Nordvall, D. Verification Summary 1st ed. Västerås: OpenDrone Project.

Verification Summary

No major deviations from the verification plan was necessary to complete the verification process (See “Verification Plan” document). The Design Verification Matrix (DVM) can be found in its own separate document labeled “DVM”.

Two Performance requirements and one communication requirements has tests as verification that was never performed due to time constraints and the necessary materials never arriving. However, these requirements could still be approved as other tests proved their compliance.

All tested Safety Requirements but one achieved the classification of Partial Approval. This is not desirable. However, the safety of the design is further argued in SSA and proved to be compliant to a reasonable and plausible degree. (See “SSA” document).

The verification process, even though it has a few partial approvals can be deemed approved as a whole as the greater majority of all requirements are both tested and approved by said testing. The Partial Approved requirements should be for further development looked into to remove any doubt of the systems dependability

K. *Nordvall, D and Harenius, L. Test Plan and Test Protocol 1st ed. Västerås: OpenDrone Project*.

Testing Plan

The testing process is part of the overarching verification project that will serve as evidence to prove the requirements met. The testing is necessary to provide tangible evidence that the requirements are fulfilled, there are multiple ways of performing these tests depending on what type of requirement that needs testing, generally two methods are used; *physical tests* and *modelling*. These three methods will be applied throughout the testing process to provide sufficient evidence that the requirements are fulfilled and that the system is dependable.

Every requirement will have to be evaluated independently using an appropriate method. Primary responsibility that these tests are performed is placed on the Verification & Validation teamlead, however, the tests can be performed by any available member. The tests will follow a specific structure based on the type of test method (see example tests). Generally they'll state the requirement to be tested, followed by relevant circumstantial information about the tests such temperature, UAV setup and other parameters that could be important to the case.

Testing will be separated into two phases; Alpha testing & Acceptance Testing. The Alpha Testing will be performed to ensure that the product meets the functionality that is expected of the drone in every stage of the development. The Validation Testing will be performed to provide the evidence that the UAV satisfies the requirements set by the project.

Testing Protocols

Example Test: Physical Tests ID PT00X

- Verify data packages being sent to and from the UAV to the MMT.

  Test Operative: Daniel Nordvall

  Relevant circumstantial information:
  The tests will be performed indoor, in a secure area with little to no electromagnetic interference. The tests will be performed in room temperature. The will be run on a Lenovo laptop, owned by one of project group members. The tests will be run on the BetaX model of the UAV with the Bosch sensor as the data gathering unit.

  Requirement Traceability: SLRXX

  Test Results:
  During autonomous flight a extract request was sent from the MMT to the UAV to gather relevant parameters assigned pre-mission. A data package was returned containing the aforementioned data, supporting manual requests for data.

  Another test where the UAV was allowed to autonomously gather data and send this to the MMT when memory capacity reached near max levels. The UAV continued to gather sensor data until 90% of max memory space, this was transmitted to the MMT and once a confirmation was sent to the UAV it dumped the memory and began gathering data once more. Supporting autonomous gathering of data.

*****Example Test: Analysis AN00X

- Examine the structure of the UAV, where potential breakpoints could be and how the UAV is constructed.

  Test Operative: Daniel Nordvall

  Relevant circumstantial information:
  The analysis will review model BetaX of the UAV. This will be done through structural analysis of potential breakpoints and the general structure.

  Requirement Traceability: FR12

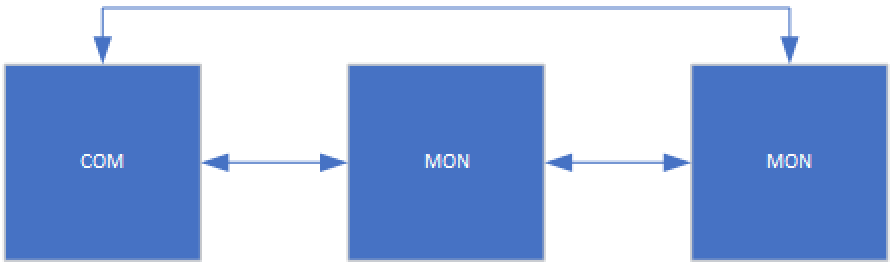
  Test Results:
  Examining the structure of the UAV we find that all arms are detachable from the main body. The inner levels are separated by plastic layers, every layer being detachable from the body of the UAV. The layers are mounted with bolts and nuts to the main body.

  Upon further analysis we find that by bending the arms there are potential break points for the arms where they are inserted into the body, these areas will need reinforcement to prevent any breaking if the UAV falls to the ground.

Example test: Modelling M00X

- Examine the chosen computer architecture, verify that it will be sufficient for the project. Requirement ID SR0X, FR0X, SLR0X

  As we can see in the modelled flowchart of the system, the project relies on a COM-MON-MON architecture to cover potential errors. A voter inside the COM will handle all inputs, decide based on majority what the correct value is.



Alpha1 Testing

Version Description

The function of Alpha 1 is/will be very basic. The desired functionality is that the drone is able to do very basic maneuvers while airborne, e.g. descend/ascend.
The Pixhawk 4 interface shall be explored, where configuration of sensors shall be done.
The different sensors that are configured, for example, the gyro, shall be tested in basic ways. First off while the drone is on the ground, and then be tested in flight.

If the different tests are approved, upgrades and experiences will be applied to the next version of the drone.
If the tests are not approved, the tests will either be tweaked, or re-configurations must be done.

Physical Tests

Test A1PT001

- Calibrate drone according to the Pixhawk 4 manual/guidelines.

  Test Operative:
  Dino & Linus

  Relevant circumstantial information:
  The drone was calibrated in QGroundControl. The sensors were tested by putting the drone in some different orientations and then verifying that the sensor output is correct.

Motors were incorrectly mounted at first which caused some problems and delays during the PID tuning. After correcting the mistake default PID values were set. The PID values were gradually changed until the oscillations were gone and the drone was stable.

Test Results:
The test was approved, the drone flies steadily.

Test A1PT002

- Test the PID settings by moving the quad without propellers to determine if the FCC is adjusting properly.

Test Operative: Dino & Linus

Relevant circumstantial information:

Test Results:
The test was approved.

Test A1PT003

- The drone shall be able to take flight and hover at low heights(0-2m).

Test Operative: Dino & Linus

Relevant circumstantial information:
The test was conducted outdoors with the manual stabilized flight mode and auto take-off.

Test Results:
The test was approved. It was able to hover at a minimum altitude of around 1.5-2m.

Test A1PT004

- The different sensors shall be tested to see if they are configured and calibrated.

Test Operative: Dino & Linus

Relevant circumstantial information:

Test Results:
The test was approved. The drone was tilted in some different orientations and the sensor output was correct. The GPS was only working outdoors.

Test A1PT005

- Motor RPM shall correspond to different angles, e.g. pitch up should make different motors to have different RPM.

  Test Operative: Dino & Linus

  Relevant circumstantial information:
  Drone is able to fly steadily.

  Test Results:
  Test was approved.

Test A1PT006

- Test sensors and verify their validity.

  Test Operative: Dino & Linus

  Relevant circumstantial information:

  Test Results:
  Test approved.

Test A1PT007

- Communicate sensor data to a device with any given medium available.

  Test Operative: Dino & Linus

  Relevant circumstantial information:
  QGroundControl was used as software and a WiFi module was used to connect the computer to the Pixhawk.

  Test Results:
  Test approved.

Test A1PT008

- Test the waypoint system in the Pixhawk using Qgroundcontrol.

  Test Operative: Dino & Linus & Armin Rad

  Relevant circumstantial information:
  Test will be performed outdoors with the QGroundcontrol software.

Test Results:
The drone was able to fly to the given waypoints accurately without any problems.

Alpha2 Testing

Version Description

Alpha 2, is designed to be bigger and more space-effective, compared to Alpha 1. This will be evaluated in tests. In software terms, it’s the same as Alpha 1.

Physical Tests

Test A2PT001

- Mount all the components.

  Test Operative: Linus, Dino, Armin

  Relevant circumstantial information:
  Try to get all the components to fit the structure of Alpha 2

  Test Results: Failed, the drone is to small, so this version of the structure will be scrapped, and a new design, Alpha 3, will be used instead.

  Evidence Classification: Failed

Alpha3 Acceptance Testing

Version Description

The function of Alpha 2 has been improved compare to the previous version Alpha1 . The desired functionalities for this version are stable hover mode, autonomous flying using waypoints and autoland.
The Pixhawk 4 shall be recalibrated for the new Alpha2 version. Sensors such as Borch-, Lidar- and IR-sensors shall be attached and tested.

If the different tests are approved, upgrades and experiences will be applied to the next version of the drone.
If the tests are not approved, the tests will either be tweaked, or re-configurations must be done.

Physical Tests

Test A3PT001

- Calibrate the drone according to the Pixhawk 4 manual/guidelines and adjust PID settings for stable flight.

  Test Operative: Linus, Dino, Armin

  Relevant circumstantial information:
  The PID tuning and calibrating shall be done in QGroundControl. The sensors shall be tested by putting the drone in some different orientations and then verifying that the sensor output is correct.

  Requirement traceability: SLR06, PR06

  Test Results: Calibrations was easily done using QGC. The drone has now good PID values for a stable/controlled flight and can easily be tuned again if needed.

  Evidence Classification: SLR06 Approved, PR06 Approved

Test A3PT002

- Test flight hovering time.

  Test Operative: Armin Rad, Dino Ramic, Linus Harenius

  Relevant circumstantial information:
  The hovering time was tested outdoors with small winds. Autonomous hovering using QG at a height of 6m until the battery warner is activated.

  Requirement traceability: PR01

  Test Results:Failed. Flight time: 16m 40s < 24m, but sufficient for Alpha 3.

  Evidence Classification: Partial Approval

Test A3PT003

- Test sensors for obstacle detection in both horizontal and vertical directions.

  Test Operative: Armin Rad, Dino Ramic, Linus Harenius

Relevant circumstantial information:
Test shall be done both indoors and outdoors.

Obstacle detection in horizontal detection using a Lidar- and IR- sensors will be tested by placing an object at a distance and moving it away until the obstacle cannot be detected anymore. The sensor will be tested alone but also when mounted on the drone flying.

In the vertical direction, ground detection will be tested using an IR-sensor. The sensor will be tested alone but also when mounted on the drone flying. A Lidar sensor will be used to test obstacle detection in the vertical direction.

Requirement traceability:
None.

Test Results: Out of scope for this project and it’s an area for further possible development.

Evidence Classification:

Test A3PT004

- Test drone temperature tolerance.

Test Operative: Armin Rad, Dino Ramic, Linus Harenius

Relevant circumstantial information:
A infrared thermometer will be used to measure temperature. Drone will be placed in a hot and cold environment.

Requirement traceability:
None.

Test Results: Not feasible as there is no access to an environmental chamber. Area for further possible development.

Evidence Classification:

Test A3PT005

- Check voltages across the motors & MCUs.

Test Operative:Erik Beckman

Relevant circumstantial information:
All tests were performed indoor in a electronics lab. The operative used a multimeter tool to examine voltages across the motors and MCUs.

Requirement traceability:

None

Test Results:
All voltages correspond with those mentioned in the datasheets for the MCUs and motors.

Evidence Classification: Approved

Test A3PT006

- Check different mode functionalities (Autonomous flight-, hover- and safe landing mode).

Test Operative: Dino & Linus & Armin Rad

Relevant circumstantial information:
Using QGroundcontrol, waypoints will be set to test the autonomous flight mode. Other flight modes shall also be tested using QGroundcontrol.

Requirement traceability:
None.

Test Results: All flight modes works using QGroundcontrol.

Evidence Classification: Approved

Test A3PT007

- Out of battery safe landing test

Test Operative: Linus, Armin, Erik, Rasmus

Relevant circumstantial information:
Check if drone autolands when it runs out of battery.

Requirement traceability:
None.

Test Results: When the battery runs out, the drone crashes. Safety margins for the battery has been added.

Evidence Classification: Failed

Test A3PT008

- Switch time for manual control

Test Operative: Armin Rad, Linus Harenius, Dino Ramic

Relevant circumstantial information:
Check the time it takes to take manual control of the drone when it is flying autonomous. The time shall be recorded for short range and long range distances.

Requirement traceability:
None.

Test Results: 100 meters away, there was no delay for manual control at all.

Evidence Classification: Approved.

Test A3PT009

- Test running a plan through the Thrift client

  Test Operative: Armin Rad, Dino Ramic, Linus Harenius.

  Relevant circumstantial information:
  Through the thrift client, the drone is able to use functions which are used by the MMT. With thrift, a mission can be programmed, and call the specific functions needed and used by the MMT.

  Requirement traceability: SLR01, SLR07, FR01, FR02, FR07, SR06, CR05

  Test Results: Successful, the mission was uploaded and the drone flew to the waypoints. This test was done without QGroundControl or RC involvement.

  Evidence Classification: Approved.

Test A3PT010

- Test if operator is able to take control during any flightphase

  Test Operative: Armin Rad, Dino Ramic, Linus Harenius.

  Relevant circumstantial information:
  If the drone fly erroneous or completely wrong, the operator should be able to cancel a missions or similar and be able to take fully manual control and/or land the drone.

  Requirement traceability:SLR02, FR05

  Test Results: The operator is able to take control through RC or through the data link in every mode.

  Evidence Classification: Approved.

Test A3PT011

- Test range control of drone.

  Test Operative: Armin Rad, Dino Ramic, Linus Harenius.

  Relevant circumstantial information:
  Plan a autonomous mission in a open field. Set waypoints with a long range, when the data link is lost, the drone will return to launch.

  Requirement traceability: FR01, PR02, CR04

  Test Results: The test was performed on a sunny, windstill winter day. A 1.7km long range mission was planned. The data link was lost at 1.3km and the drone returned to launch.

  Evidence Classification: Approved.

Test A3PT012

- Test geofence violation for Altitude.

  Test Operative: Armin Rad, Dino Ramic, Linus Harenius.

  Relevant circumstantial information:

  Ascend above the altitude set as the geofence violation.

  Requirement traceability: SR03

  Test Results: The geofence violation for altitude was tested with flying up to and above the entered maximum allowed altitude. When the drone was above this height, the Return to Launch mode was entered automatically.

  Evidence Classification: Approved.

Test A3PT013

- Test all the failsafe features/modes

  Test Operative: Armin Rad, Dino Ramic, Linus Harenius.

  Relevant circumstantial information:

  Check QGroundControl for the parameters.

  Requirement traceability: SR03, SR05

  Test Results: The failsafe modes can be changed however the user wants, for an example, wait time until the failsafe is triggered in case of data link lost. All the available failsafes has been tested as follows:

- Datalink lost → Return to launch after 7 seconds of no communication
- Battery low → Return to launch, criticality levels have to be configured to the specific battery which is being used.
- RC lost → Lands in the manual mode used before it lost its RC connection(i.e no inputs so it lands as standard if Position/Altitude mode is used) if it’s in manual flight mode, it would just have crashed, so as of now it does not detect RC loss.
  Manual flight mode should never be used since it will severely increase the risk of crashing the drone.
- Geofence violation, both Altitude and radius → See test A3PT012

Evidence Classification: Partial approved. The only function not working completely is the RC loss failsafe.

Test A3PT014

- Test the return to base functionality, integrated with the failsafe mode “Low Battery”.

Test Operative: Armin Rad, Dino Ramic, Linus Harenius.

Relevant circumstantial information:
Fly the drone until the battery is low enough where the mission cannot be completed. The drone shall then abort the mission and return to the ground station.

Requirement traceability: FR06, FR08

Test Results: The critically low battery threshold was set to a high level so it would trigger the return to launch failsafe(Which it did). If the battery runs below the emergency threshold it lands immediately.

Evidence Classification: Approved.

Test A3PT015

- Test the lidar sensor for precision landing.

Test Operative: Armin Rad, Dino Ramic, Linus Harenius.

Relevant circumstantial information:
Test outdoors where GPS signal i available. Auto takeoff 5m and then activate autoland to check if a smooth landing is performed.

Requirement traceability: FR05.

Test Results:
Successful. The autoland function was improved significantly compared to using only barometric altitude sensor.

Evidence Classification: Approved.

Test A3PT016

- High speed test.

  Test Operative: Armin Rad, Dino Ramic, Linus Harenius.

  Relevant circumstantial information:
  Fly the drone at high speeds.

  Requirement traceability: Noone.

  Test Results:
  Successful. The drone can easily fly faster than 15 m/s. It’s not recommended though because of safety and increased wear on materials such as propellers.
  Recommended ground speed is below 15 m/s ≈ 54 km/h.

  Evidence Classification: Approved.

Analysis Tests

Test A3AN001

- Analysis of the platform the drone is built. Does the platform satisfy the “Open Source requirements”?

  Test Operative: Linus Harenius & Daniel Nordvall

  Relevant circumstantial information:
  The project should allow the developer to change every parameter to the users needs. This will allow changing the whole behaviour of a drone between different modes by just uploading another configuration file to the Pixhawk.

  Requirement traceability: SLR05

  Test Results:
  The platform allows the user to switch whatever parameter he likes. A huge variety of different peripheral hardware can also be installed and switched out whenever the user wants. This provides the UAV with the possibility to be adapted with any kind of goal in mind,

such as improving the safety by adding redundancy or improving the design with multiple batteries.

Normal and abnormal conditions are irrelevant for the requirement in question.

Evidence Classification: Approved

Test A3AN002

Analysis of the log if a crash has happened and search for information regarding faults that caused the crash.

Test Operative: Armin Rad, Dino Ramic, Linus Harenius & Daniel Nordvall.

Relevant circumstantial information:
The Pixhawk logs data from relevant sensors and inputs during flight on the SD card. QGroundControl logs the telemetry and flighmode/battery on the operating computer which is connected via the data link.

The safety log gives the user a comfort knowing that the UAV tracks its vital information for analysis later.

Performance is unchanged.

Under normal conditions the UAV could use the data to present to MMT, during a crash this data would be used as a crash log to understand what went wrong.

Requirement traceability: FR03

Test Results:
The flight log can easily be accessed through USB or through the SD-card directly. The telemetry can be replayed through QGroundControl on the computer who was connected.

Evidence Classification: Approved

Test A3AN003

- Test the emergency landing functionality.

Test Operative: Armin Rad, Dino Ramic, Linus Harenius.

Relevant circumstantial information:
Check if any critical failure occurs the UAV shall be able to perform an emergency landing.

Requirement traceability: SR01, FR05,

Test Results:

If a critical failure occurs, the drone will either land immediately(GPS loss) or crash if multiple critical failures occur(more than 1 IMUs lost).

Evidence Classification: Approved.

BetaV1-3 Acceptance Testing

Version Description

Beta has been further improved compared to Alpha 3 and Alpha 1. The weight has been reduced because of carbon fiber and fiberglass has been added as the main structure material. The battery of the Beta versions will be the LiPo 3S 8000mAh battery, which is the one used for Alpha 3. The battery has the maximum flight time of all the available batteries.
No essential features has been added to Beta V1-3 compared to Alpha 3, except for new materials which is ment to help evaluate which material configuration is the best in terms of weight reduction and increased flight time.

The main features of the drone is tested for Alpha 3, so the main test for Beta V1-3 is just to verify that the flight time is increased due to a lower weight, and verify that all the functionalities from Alpha 3 (Autonomous flight etc) is still functional.
When the features and functionalities are verified and validated for Beta V1-3, the tests and development of the drones are complete.

Physical Tests

Test BXPT001

- Test flight, both autonomous and manual

  Test Operative: Armin Rad, Dino Ramic, Linus Harenius.

  Relevant circumstantial information:
  Test if Beta Vx performance is the same or better than Alpha 3.
  Autonomous flight and Manual flight

  Requirement traceability: PR06, CR04

  Test Results:

  Evidence Classification:

Test BXPT002

- Measure realistic hover time.

Test Operative: Armin Rad, Dino Ramic, Linus Harenius.

Relevant circumstantial information:
Test if Beta Vx performance is the same or better than Alpha 3 in terms of hover time

Requirement traceability: PR01

Test Results:

Evidence Classification:

Analysis Tests

Test BXAN001

- Analysis of the which structure/material combination is the best one in terms of performance.

  Test Operative: Linus Harenius, Armin Rad, Erik Beckman.

  Relevant circumstantial information:

  Requirement traceability:

  Test Results:

  Evidence Classification:

Test BXAN002

- Analyze the UAV chassi structure to determine how modular it is.

  Test Operative: Daniel Nordvall

  Relevant circumstantial information:
  The analysis will be performed on the final version of the UAV produced by this project. It will focus on examining the structural construction of the UAV.

  Requirement traceability: FR04

  Test Results:
  Examining the structure of the UAV we find that all arms are detachable from the main body. The inner layers consists of two separate plastic layers, these are mounted through screw holes in the side of the UAV. The two layers press upon the arms locking them in place together with bolts and nuts. This provides the UAV the capability of removing any body part easily to replace if needed.

The modular design doesn’t affect the performance, but it does affect the time required to maintain or fix a damaged part, meaning downtime of the UAV outside of the recharge time of the battery is drastically reduced. If launched as a commercial product, this would also allow the user to replace damage parts without the need to send the UAV to service operator, or buy an entirely new drone.

The safety is aspect is unaffected by the modular design.

Evidence Classification: Approved

Test BXAN003

- Analyze the positioning solution that it provides the required accuracy.

  Test Operative: Daniel Nordvall

  Relevant circumstantial information:
  The analysis will be performed on the final beta version of the product. The autonomous positioning solution utilizes a combination of GPS, Barometer and the Pixhawks PID settings to control the engines.

  Requirement traceability: SLR06

  Test Results:
  Through careful calibration of the Pixhawk PID settings, the project could achieve stable behaviour of the drone in manual flight mode, alongside the GPS and Barometer the system is able to stay in its current position relatively stable, drifting only a little bit.

  A stable behaviour is key for the mission safety, without it, the drone could drift too far off position or crash.

  During normal conditions, weak winds, the drone can hold its position relatively tight, however, during heavier winds, abnormal conditions, the drone would have a much harder time being as precise.

  Evidence Classification: Approved

Test BXAN004

- Analyze the system, what happens when a critical failure occurs.

  Test Operative: Daniel Nordvall

  Relevant circumstantial information:
  The analysis will be performed on the final beta version of the product.

Requirement traceability: SR01

Test Results:
The UAV, according to the System Fault Tree, the potential to emergency land if a mission critical failure occurs, however, depending functionality fails the system may or may not be able to perform said landing. If for an example the barometer fails, the system is still able to perform a emergency landing. However, if a motor is lost the system will crash.

The emergency landing is therefore possible, but only under specific circumstances.

Evidence Classification: Partial Approval

Test BXAN005

- Analyze the potential of entering a low power mode, turning off non-essential equipment and return to base when battery power is not sufficient to complete the mission.

Test Operative: Daniel Nordvall

Relevant circumstantial information:
The analysis will be performed on the final beta version of the product.

Requirement traceability: SR02

Test Results:
The UAV in its current form does not have the capability of turning off non-essential equipment, as there are very few components to the system that are not essential. Due to a lack of resources this functionality has not been implemented but could be with further development.

The drone is however able to Return to Base if the battery level is not sufficient to complete the mission.

Based on this analysis the UAV would receive a Partial Approval of the requirement.

Evidence Classification: Partial Approval

L. Nordvall, D. DVM. 1st ed. Västerås: OpenDrone Project.

Design Verification Matrix

Verification Status:

- Approved
- Partial Approval
- Failed
- Pending
- Not Started

Requirement ID	Requirement	Source of Requirement	Verification Status	MoC	Verification Reference
101	The drone shall fly autonomously with inputs from the Mission Management Tool (MMT). (*)	**SLR01** System Level Requirements	Approved	On site tests	A3PT009
102	The system shall allow the human operator to take manual control of the UAV during any given flight phase. (*****)	**SLR02** System Level Requirements	Approved	On site tests	A3PT010
103	The system shall allow collaboration with other UAV and other autonomous systems, such as Unmanned Ground Vehicles (*****)	**SLR03** System Level Requirements	Not started	No V&V as it is part of the larger scope of Afarcloud.	NaN
104	The system shall demonstrate a degree of dependability by comparing application domain standards with the experimental settings. (*)	**SLR04** System Level Requirements	Not started	Proved in the SSA.	NaN
107	The UAV shall be built upon a open source platform to ease further development.	**SLR05** System Level Requirements	Approved	Analysis	A3AN001

108	The UAV positioning solution shall provide the accuracy necessary to complete basic farming tasks.	**SLR06** System Level Requirements	Approved	Analysis and tests.	A3PT001, BXAN003
109	The UAV shall be able to perform sensor based tasks during missions.	**SLR07** System Level Requirements	Approved	On sites test..	A3PT009
110	The UAV shall use the same date/time formats of other vehicles and nodes in the chain.	**SLR08** System Level Requirements	Not started	This is outside the scope of the project and is a area of further possible developm ent	NaN
201	If the range requirements are met, the system shall send data in real-time to the MMT. (*****)	**FR01** Functional Requirements	Approved	On site tests.	A3PT009, A3PT011
202	The UAV shall post-mission transmit relevant sensor data to MMT.	**FR02** Functional Requirements	Approved	On site tests.	A3PT009
203	The UAV shall record safety critical faults	**FR03** Functional Requirements	Approved	Analysis	A3AN002
204	The UAV shall be designed in a modular fashion to allow easy maintenance and replacement of parts.	**FR04** Functional Requirements	Approved	Analysis of the structure.	BXAN002
205	If prompted the UAV shall perform an emergency landing with a low descend speed.	**FR05** Functional Requirements	Approved	On site tests and Analysis.	A3PT010, A3AN003, A3PT015

206	The UAV shall have a "Return to Base" function, when triggered, signalling the UAV to immediately return to base.	**FR06** Functional Requirements	Approved	On site tests.	A3PT014
207	The UAV shall store mission relevant data in the following ways: 1. Store relevant mission data onboard until post-mission transmission. 2. Send stored data about the mission to the MMT.	**FR07** Functional Requirements	Approved	On site tests.	A3PT009
208	The UAV shall have flight modes that allow degraded operation as Return to Base and Emergency Landing.	**FR08** Functional Requirements	Approved	On site tests	A3PT014
301	If any critical failure occurs the UAV shall be able to perform a emergency landing.	**SR01** Safety Requirements	Partial Approval	Analysis of the system safety.	A3AN003, BXAN004
302	The drone shall enter a low power mode turning off non-essential equipment and return to base when battery power is not sufficient to complete the mission.	**SR02** Safety Requirements	Partial Approval	Implementation of a degraded mode were less power drain is necessary to run the drone.	BXAN005
303	The operating altitude shall be no less than 20 meters above the ground.	**SR03** Safety Requirements	Partial Approval	On site tests.	A3PT012, A3PT013

304	The UAV should include collision detection and avoidance functionality.	**SR04** Safety Requirements	Not started	This is outside the scope of the project and is a area of further possible development..	NaN
305	The drone shall have a fail-safe mode with degraded operation.	**SR05** Safety Requirements	Partial Approval.	On site tests.	A3PT013
306	The UAV shall transmit if in range, alternatively store a backlog of the following information: 1. Three dimensional positioning data. 2. Velocity 3. Airspeed 4. Remaining battery 5. Task status 6. Heartbeat signal for proof of life	**SR06** Safety Requirements	Approved	On site tests.	A3PT009
401	The drone **shall** be able to hover for 24 min (no wind) minimum.	**PR01** Performance Requirements	Partial Approval	On site tests.	A3PT002 (Partial Approval), BXPT002 (Not Tested)
402	The drone **shall** have a telemetry range of 1 km.	**PR02** Performance Requirements	Approved	On site tests.	A3PT011
403	The drone **shall** have an obstacle detection range of 15 m or more of its surrounding.	**PR03** Performance Requirements	Not started	This is outside the scope of the project and is a area of further	NaN

				possible developm ent.	
404	The drone **shall** have a ground detection distance of at least 30 m or more.	**PR04** Performance Requirements	Not started	This is outside the scope of the project and is a area of further possible developm ent.	NaN
405	The drone **shall** be able to operate in temperatures between - 20°C - 40°C.	**PR05** Performance Requirements	Not started	This is outside the scope of the project and is a area of further possible developm ent.	NaN
406	The drone shall be able to operate in harsh weather. (snow/rain)	**PR06** Performance Requirements	Approved.	On site tests.	A3PT001 (Approved), BXPT001 (Not Tested)
601	Sensor data **Shall** be sent to a cloud server.	**CR01** Communication Requirements	Not started	This is outside the scope of the project and is a area of further possible developm ent	NaN
602	The drone **shall** be able to communicate with other drones. Send and receive data.	**CR02** Communication Requirements	Not started	This is outside the scope of the project	NaN

				and is a area of further possible developm ent.	
603	There **shall** be one control station controlling multiple drone on the same wifi.	**CR03** Communication Requirements	Not started	This is outside the scope of the project and is a area of further possible developm ent.	NaN
604	Minimal switch time from autonomous mode to manual control from far distances.	**CR04** Communication Requirements	Approved	On site tests.	A3PT011 (Approved), BXPT001 (Not Tested)
605	Sensors **shall** be able to connect via wifi to send data.	**CR05** Communication Requirements	Approved	On site tests.	A3PT009

M. Ramic, D. (2019), Flowcharts 1st ed. Västerås: OpenDrone Project, Data Collection & Server Flow charts

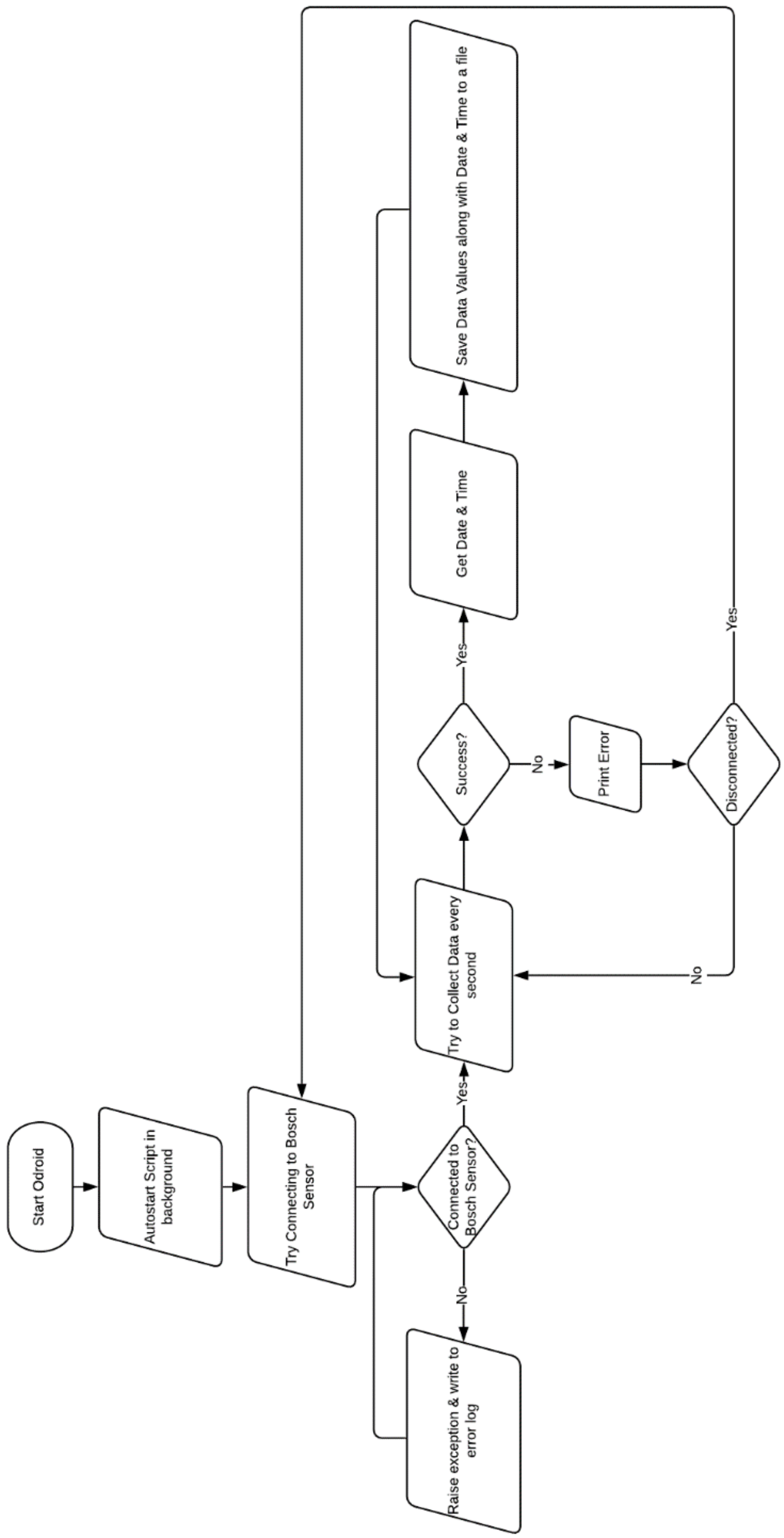


Figure 19 Data Collection flow chart

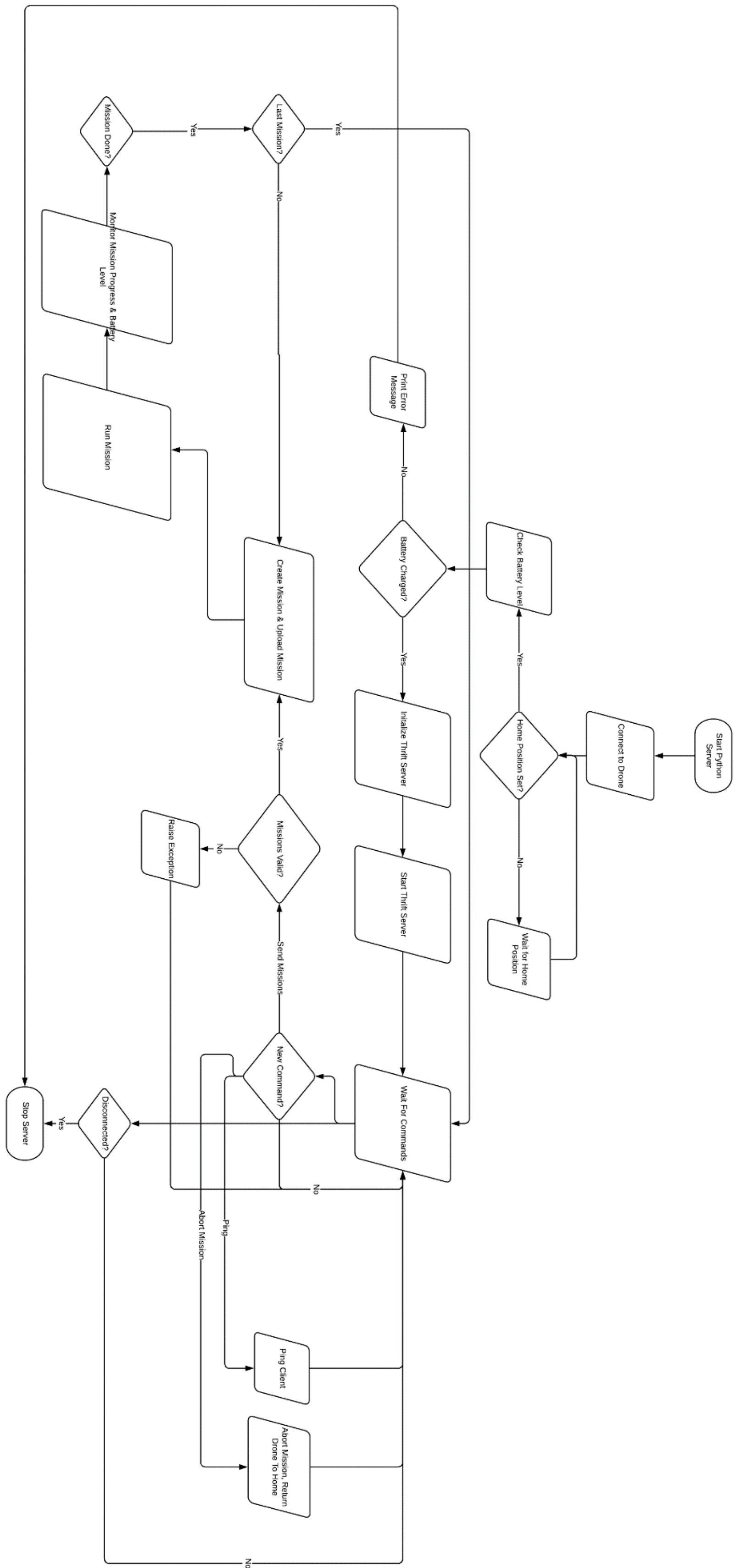


Figure 20 Server flow chart

N. (2019), AT2820-motor-specification, OpenDrone Project.

Motor: AT2820- 6 KV:870					
Technical Datas		**Recommended Prop(inch)**			
KV	870	standard	3S-1470/1480	Max thrust	3s-1470/1480
Configu-ration	12N14P		4S-1155/1260		4s-1360
Stator Diameter	28mm				
STator Length	20mm				
Shaft Diameter	5mm				
Motor Dimension(Dia. * Len)	Φ35×42mm				
Weight(g)	132				
Idle current(10)@10v(A)	1.2				
No.of Cells(Lipo)	3-6S				
Max Continuous current(A)180S	42A				
Max Continuous Power(W)180S	480W				
Max. efficiency current	(10-32A)>75%				
internal resistance	58mΩ				

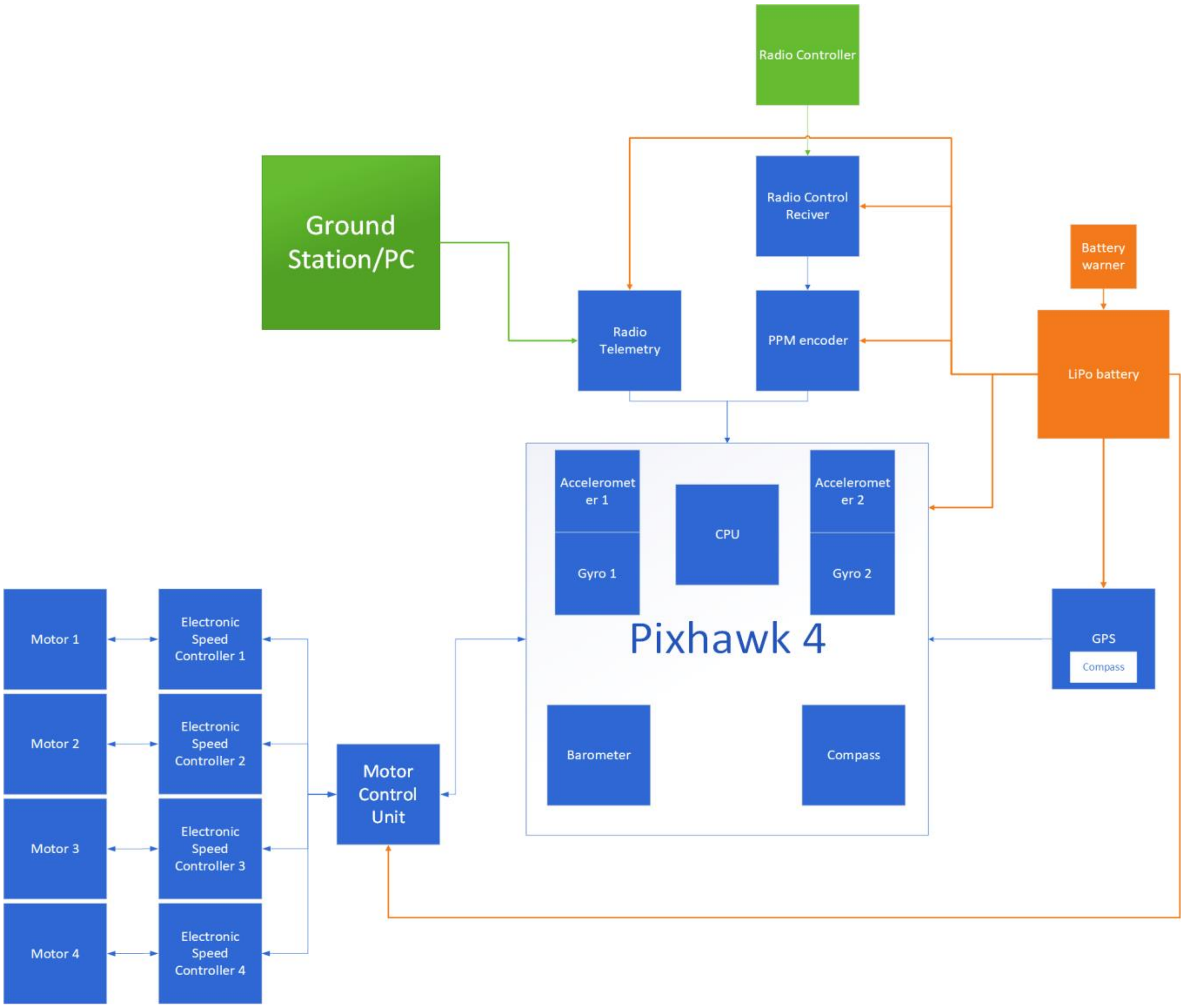
Tested with Tiger motor 60A ESC								
Prop	**Volts (V)**	**Amps (A)**	**Watts (W)**	**Thrust (g)**	**Thrust (oz)**	**Efficiency (g/W)**	**Efficiency (oz/W)**	**Remark**
12X3.8	10.5	24.3		1496				
12X6	10	26.4		1455				
	10.5	28.6		1603				
	11	30.9		1712				
	12	35.4		1931				
	13	40.5		2199				
	14	43.4		2367				
	14.8	42.7		2307				

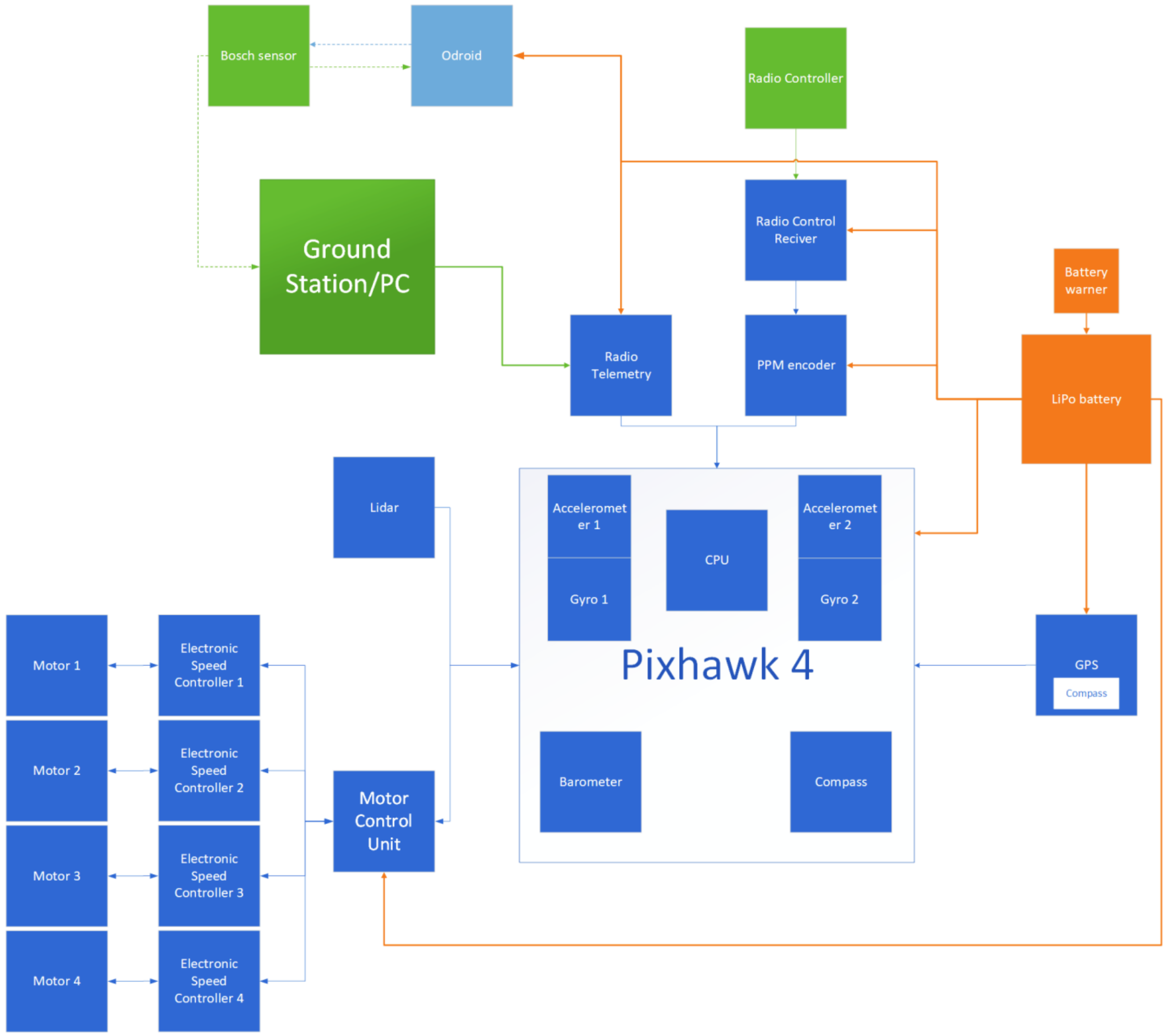
Motor: AT2820- 5 KV:970					
Technical Datas		**Recommended Prop(inch)**			
KV	970	standard		Max thrust	
Configu-ration	12N14P				
Stator Diameter	28mm				

STator Length	20mm	
Shaft Diameter	5mm	
Motor Dimension(Dia.＊Len)	Φ35×42mm	
Weight(g)	132	
Idle current(10)@10v(A)	1.4	
No.of Cells(Lipo)	3-6S	
Max Continuous current(A)180S	45A	
Max Continuous Power(W)180S	520W	
Max. efficiency current	(12-35A)>71%	
internal resistance	42mΩ	

Motor: AT2820-4 KV:1100					
Technical Datas		Recommended Prop(inch)			
KV	1100	Standard	3S-1260	Max thrust	3S-1365/1365
Configuration	12N14P		4S-1047/1060		4S-1155/1170
Stator Diameter	28mm				
Stator Length	20mm				
Shaft Diameter	5mm				
Motor Dimension(Dia.＊Len)	Φ35×42mm				
Weight(g)	132				
Idle current(10)@10v(A)	1.7				
No.of Cells(Lipo)	3-4S				
Max Continuous current(A)180S	50A				
Max Continuous Power(W)180S	580				
Max. efficiency current	(13-35A)>72%				
internal resistance	36mΩ				

O. Harenius, L. Figures 1st ed. Västerås: OpenDrone Project. Enlarged architecture figures.




Bosch sensor
Odroid
Radio Controller
Ground Station/PC
Radio Control Reciver
Battery warner
Radio Telemetry
PPM encoder
LiPo battery
Lidar
Accelerometer 1
Gyro 1
CPU
Accelerometer 2
Gyro 2
Pixhawk 4
GPS
Compass
Barometer
Compass
Motor 1
Electronic Speed Controller 1
Motor 2
Electronic Speed Controller 2
Motor 3
Electronic Speed Controller 3
Motor 4
Electronic Speed Controller 4
Motor Control Unit