RAPPORT DE STAGE SEMESTRE 8

Présenté par **Paul WALCZAK**

« Titre du stage»

Rapport présenté et soutenu à « Lieu », le « date»

Composition du Jury :

Tuteur ENIB :	Alexis MICHEL	Directeur de l'ENIB
Accesseur :	Prénom NOM	ENIB
Tuteur Entreprise :	Vincent KLYVERTS TOFTERUP	Responsable du développement des drones

Informations Administratives :

Entreprise :	Upteko ApS
Établissement :	École Nationale d'Ingénieurs de Brest (ENIB)
Dates :	du 06 Mars 2023 au 04 Aôut 2023

ACKNOWLEDGEMENT

Je tiens à remercier

I would like to thank. my parents..

J'adresse également toute ma reconnaissance à ....

....

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LISTE OF CODES

COMPANY PRESENTATION

1.1 Field of activity

1.1.1 Wind energy

Upteko has been manually inspecting wind-turbines since 2018. Today, they are using this data and knowledge to build a system to automatically inspect wind turbines Upteko's autonomous drone system will provide high-quality aerial data for all inspections and maintenance requirements, in a safer, more efficient manner, without causing operational downtime. Their drones are pre-programmed with geo-referenced 3D trajectories for the inspection of wind turbines. While flying the planned routes, several images are acquired. The actual number depends on the camera specification, flight altitude and post-processing performance (overlap between images). All images are georeferenced using the UAV position at the acquisition time and these images are afterwards stitched together and in one picture on which the locations of the damages are shown. They deliver these data as per the clients' requirements in raw files or provide a damage report that explains the severity of damages from type 1 to type 5.

1.1.2 Oil and Gas

Upteko is working on a solution for detecting oil spills on the surface of the water. They are developing payloads for their drone system that can help detect the signature of oil floating on water surfaces. This payload can be installed on Lärke and can be used to automatically or manually detect oil spills. Upteko have developed an application to detect pirate attacks on off-shore oil rigs and large commercial ships. Pirates use small ships to attack these oil rigs and large commercial ships at night. They take the crew and contents hostage. There are no mobile long-range detection tools with aerial insight to give the crew information about nearby pirates. This results not only in huge loss to the

oil industry but also endangers the crew at huge life risks. The total cost of piracy to the shipping industry, from insurance and time lost, is estimated at over €6B/year. Upteko's multipurpose drone system has a feature called perimeter control. This feature allows any member of the crew to investigate surrounding vessels approaching the premises. A built-in feature within the drone system is added to allow a crew member to operate and control the aerial camera in any direction with no prior drone experience. The geothermic cameras and the infrared sensors help to detect the approach of any pirates faster even during night-time.

1.1.3   Offshore and Maritime

Upteko has been consistently involved with the maritime industry since its inception. They are developing a drone system solution to live on the ship, to connect port operations on the ground and at the sea, with insight from the sky. This system includes a drone and a charging station for the drone that will be installed on the ship. The drone can autonomously perform a variety of tasks and then fly back and charge its batteries while staying protected against the weather. With an in-depth understanding of the challenges that the maritime industries face in attempting long and expensive inspections and other operational tasks, Upteko's software and hardware drone system allows a 100% automatic inspection of a ship in less than 2 hours. Having a permanent drone on a ship will be extremely helpful in a number of cases that can range from Search and Rescue (SAR) operations, vessel docking, dry dock inspections, fire hazard detection and situational awareness among other functionalities.

1.2   History

Upteko™ was founded in 2018 by Mads Joergensen, Benjamin Mejnertz, and Sebastian Duus to pursue the opportunity of developing drone applications for the maritime sector. For Benjamin and Sebastian, it started as a hobby, competing in RC Helicopter competitions around the world and later became a business worth pursuing. Mads came onboard, and together they created Upteko and built a great team of experienced drone pilots, software- and hardware engineers, and business developers. Today Upteko has offices located in Copenhagen, Odense, and Skanderborg. Upteko is about using their creativity and initiative to become leaders in the drone industry. To develop fitting so-

lutions to your needs, they value collaboration and co-creation highly among external parties. Through several years of collaborating with their customers, complementary assets providers, and even competitors, they have achieved priceless partnership within the maritime and other sectors. They are continuously on the hunt for new collaborations.

1.3   Legal and social model

In Denmark, most companies operate under a legal structure of a private limited company (Anpartsselskab, ApS) as Upteko. The country has strong social welfare systems and labor laws, which mandate fair treatment and protection of employees. Danish companies usually have a strong focus on maintaining a healthy work-life balance and uphold high ethical standards, both socially and environmentally.

1.4   Economic model

Denmark follows a mixed-market capitalist system, combining free market principles with a strong regulatory oversight. Upteko operates under a model of sustainable growth, focusing on long-term stability rather than short-term profits. This includes social responsibility, ethical business practices, and environmental sustainability.

1.5   Type of work organization

Denmark has a distinctive workplace culture, often characterized by a flat organizational structure. This implies low power distance, high levels of trust, and extensive collaboration between different levels of the organization. Danish companies typically have strong communication and decision-making practices, promoting employee empowerment and autonomy. Upteko is organised in three main sectors : `Technical Development` , `Finance` and `Product management` . Here is the actual organization chart of the company :
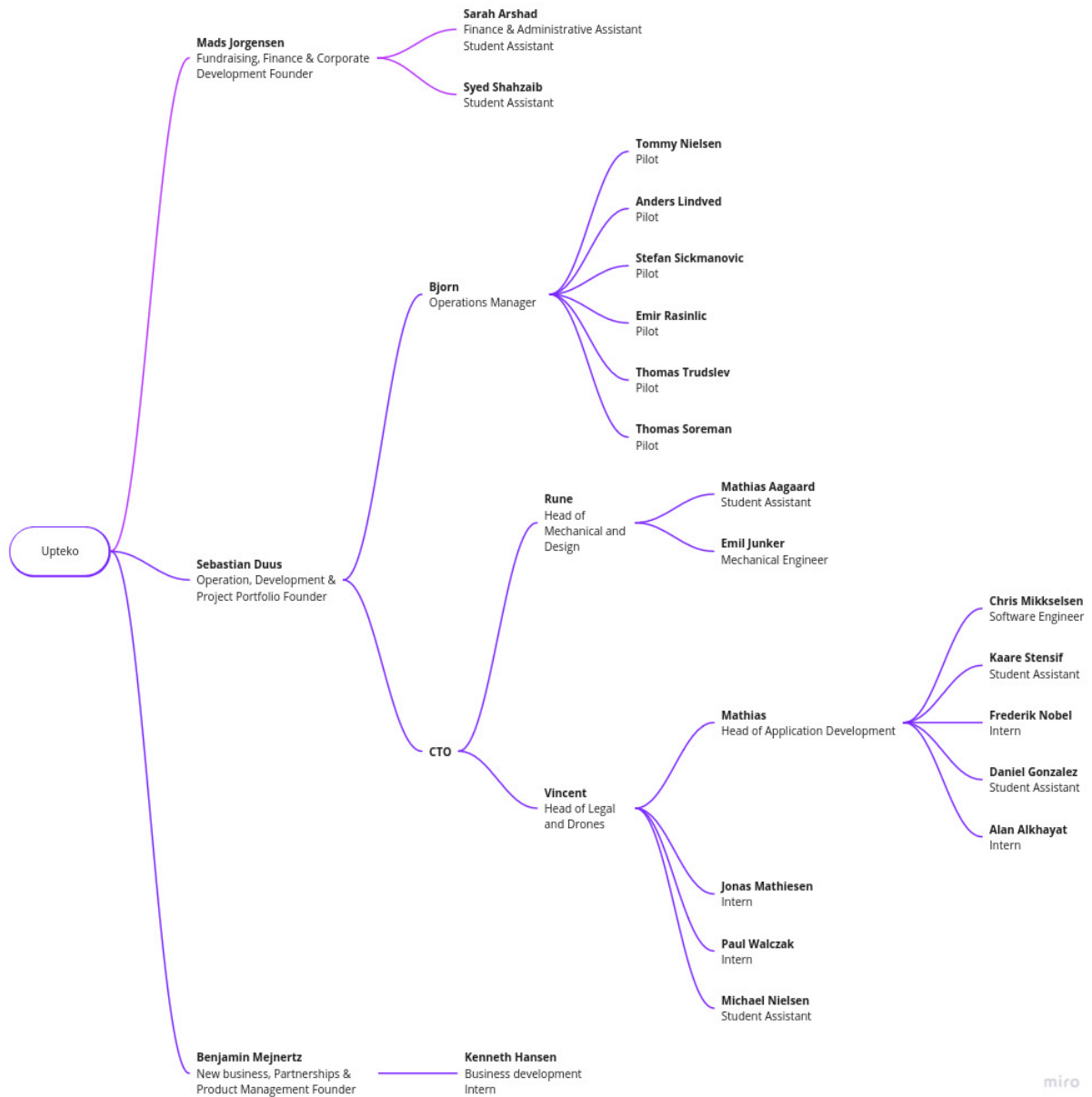
Figure 1.1 – Upteko company organization chart

PROJECTS

2.1 STM32 IMU

2.1.1 PModNav

The purpose is to get the `roll`, `pitch` and `yaw` from the PModNav to plot a 3D visualization of the IMU. The final purpose of it is to create log files from the orientation and attitude of the drone in real-time. By combining data from accelerometers, gyroscopes and magnetometers, an IMU can provide information about the object's position, orientation and angular velocity. This is crucial for tasks such as safety deployement or drone tracking. The STM32-L4796ZG recovers the value of the sensors from the `PModNav` via `SPI Bus` and transmits them to the HMI via `UART` connection.
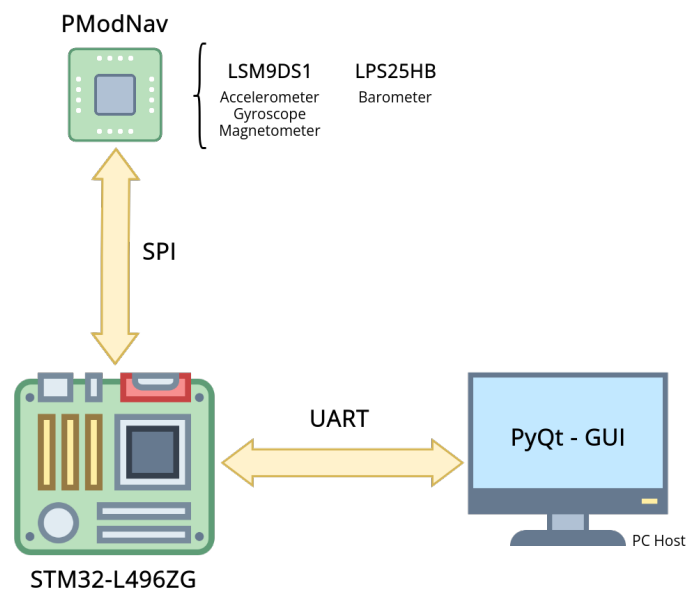
Figure 2.1 – Project communication overview

2.1.1.1 Hardware

The PModNav module is equipped with the `LSM9DS1` [1] sensor, offering 10-degrees-of-freedom (10-DOF) functionality. It integrates a 3-axis accelerometer, 3-axis gyroscope, 3-axis magnetometer, and an LPS25HB digital barometer. This comprehensive sensor suite allows users to obtain orientation-related data and determine the precise position and heading of the module. The module supports various full-scale options for linear acceleration, angular rate, and magnetic field measurements. It follows the Digilent Interface Specification Type 2A and utilizes a 12-pin Pmod connector with an SPI interface.

2.1.1.2 Software and Development Environment

The development environment for the PModNav project is STM32 CubeIDE. The documentation references project sources, including code snippets and libraries, such as the PModNav driver and Madgwick's filter implementation.

2.1.1.3 Data Processing

The project outlines two approaches for deriving object attitudes: Euler angles and quaternions. Euler angles are obtained through the integration of angular velocity and provide information about the roll, pitch, and yaw of an object. However, a challenge known as `Gimbal Lock` [2] arises when using Euler angles directly, resulting in a loss of a degree of freedom when two axes of rotation overlap. To overcome this, quaternions are introduced as a mathematical representation of displacement and rotation. They effectively resolve the Gimbal Lock issue and provide a more robust solution for determining attitudes.
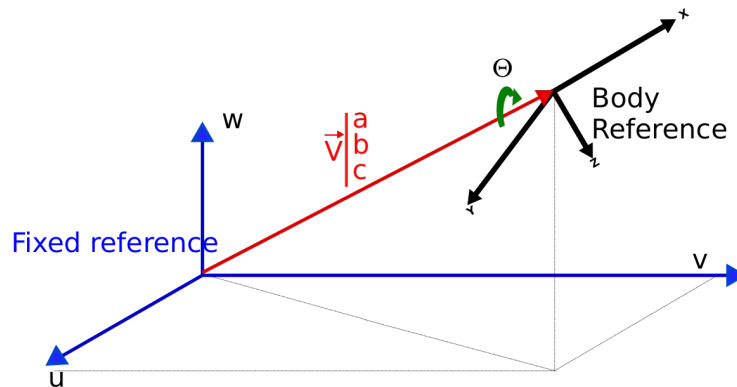
Figure 2.2 – Quaternion representation

13

$$Q = [q_0, q_1, q_2, q_3]$$

$$Q = \left[cos(\frac{\theta}{2}), a.sin(\frac{\theta}{2}), b.sin(\frac{\theta}{2}), c.sin(\frac{\theta}{2}) \right]$$

At each `Te`, we can calculate the new value of the quaternion vector from the velocity :

$$Q_{k+1} = Q_k + \frac{1}{2}.T_e.\omega_k.Q_k$$

For cheap IMUs, it is unavoidable to perform a data fusion to make the accelerometer compensate for the gyroscope defect. If using a Kalmann filter is possible, there are other (faster) algorithms like Madgwick's. The idea is to compensate for the gyroscope measurement error by modulating its values by the result of a comparison between an estimate of the gravity field and the measured gravity field (with the accelerometer).
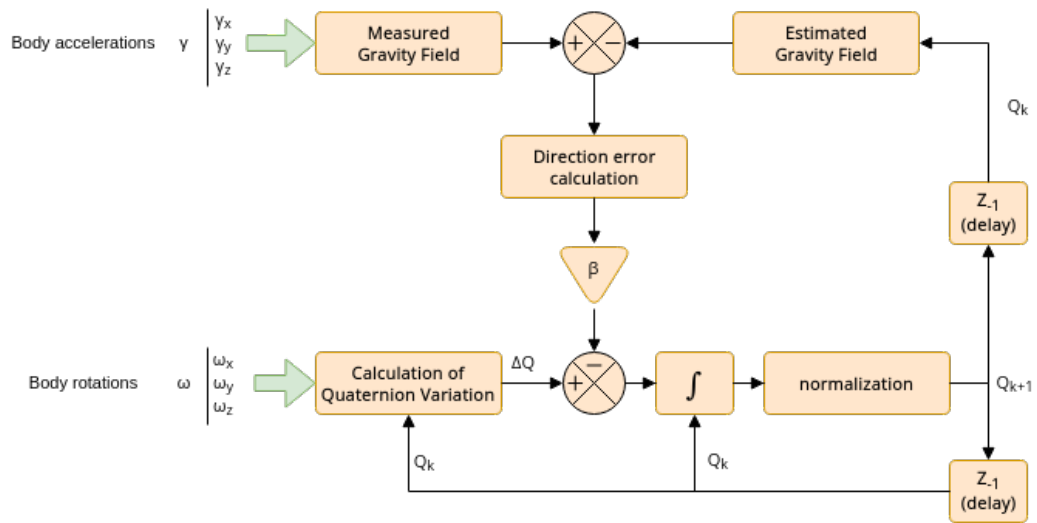
Figure 2.3 – Madgwick filter schematic

I decided to use the most popular open source algorithm to compute rotations, the `Madgwick's algorithm` [3]. This calculation updates the quaternion, from which the attitudes (Euler angles) can be calculated.
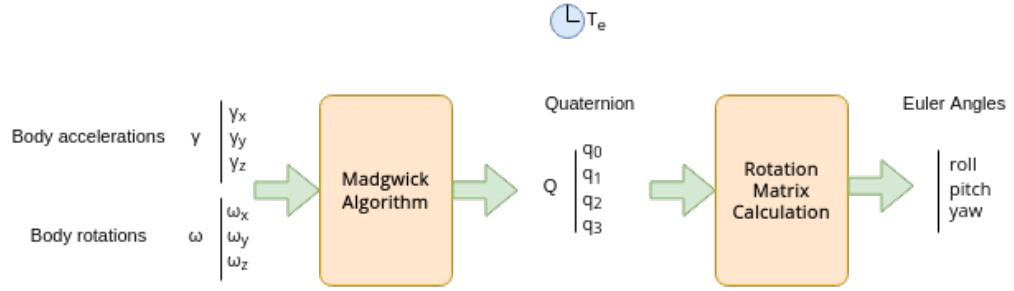
Figure 2.4 – Data transform schematic

$$R_{12} = 2.(q_1.q_2 + q_0.q_3)$$

$$R_{22} = q_0^2 + q_1^2 - q_2^2 - q_3^2$$

$$R_{31} = 2.(q_0.q_1 + q_2.q_3)$$

$$R_{32} = 2.(q_1.q_3 - q_0.q_2)$$

$$R_{33} = q_0^2 - q_1^2 - q_2^2 + q_3^2$$

Calculation of the Euler Angles from the rotation matrix :

$$roll = atan2(R_{31}, R_{33})$$

$$pitch = asin(R_{32})$$

$$yaw = atan2(R_{12}, R_{22})$$

2.1.2   Graphical User Interface

Additionally, a graphical user interface (GUI) is provided, leveraging `PyQt5` [4] and `PyOpenGL` [5] modules. The GUI manages the main window and handles OpenGL object management. It offers features like port selection and serial communication. The received data is displayed in a textbox within the GUI, facilitating real-time monitoring and analysis. Here is how the GUI is threaded :
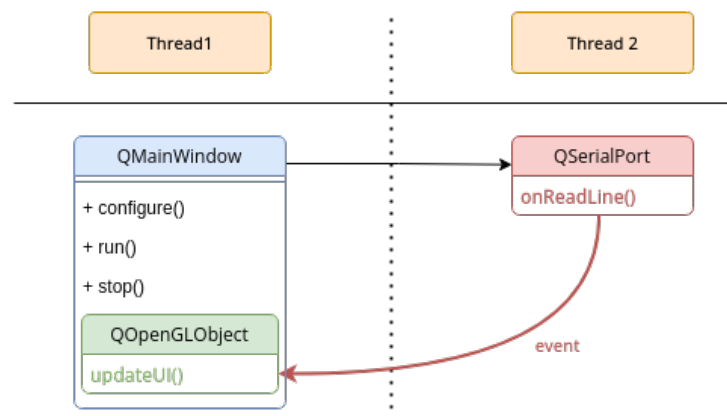
Figure 2.5 – Python GUI threads schematic

The use is rather simple for the communication configuration (cyan box) :

— the port
— the baud rate
— the number of bits per frame
— the number of stop bits
— the parity
— the flow control
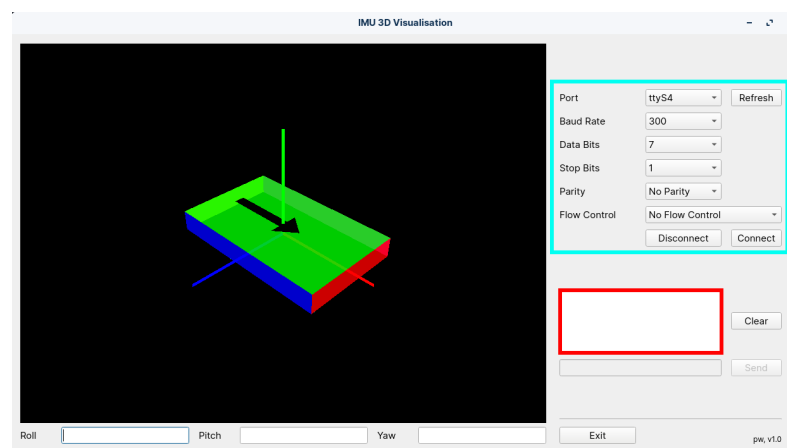
Figure 2.6 – Python GUI window screenshot

Then click on `Connect` to start the serial communication. Every received line appeared in the textBox (red box).

2.1.2.1 Final result
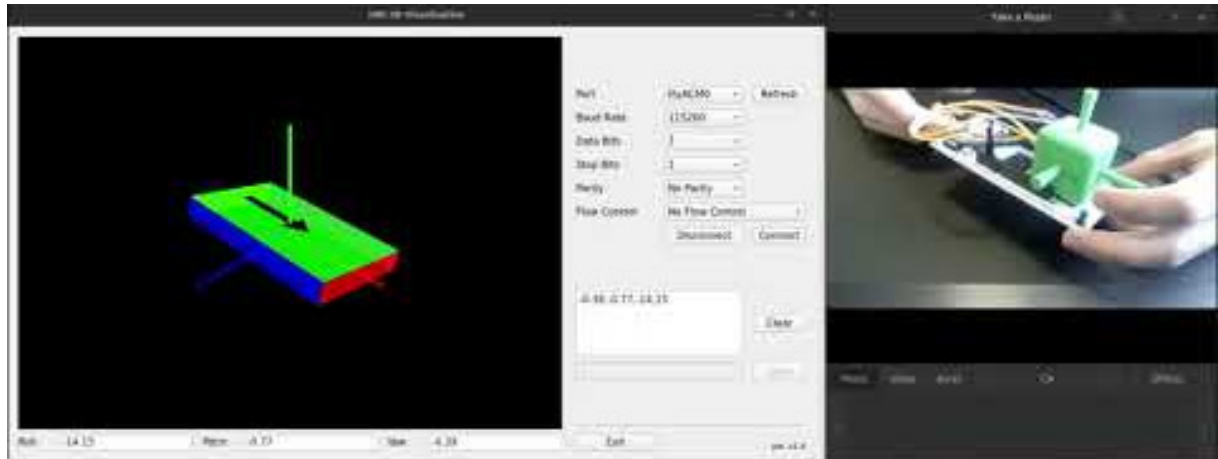
Figure 2.7 – 3D IMU Visualizer demo

2.1.3 Telemetry data

2.2 LogViewer

The objective of a log viewer application is to facilitate the efficient viewing and analysis of log data generated by systems or applications. This scientific report highlights the key goals and functionality of a log viewer, which includes log visualization, search and filtering capabilities, log analysis and insights, integration with other tools, and customization options.

2.2.1 Sensors tests

Before using any sensors, make sure they work properly, that communication is correct and that measurements are consistent with the reality of the environment. To check all these prerequisites, it's often easiest to use the manufacturer's proprietary software, which is robust, reliable and easy to use.

A few parameters need to be modified to enable the 8 distance sensors and their data to be taken into account by the collision avoidance built-in algorithm.

Here is the final configuration for a minimum effective distance of 3 meters between the drone and obstacles and a maximum effective distance of 5 meters :

	TowerEvo	Description
PRX_ORIENT	0	Default
PRX_TYPE	6	TeraRangerTowerEvo
SERIAL_BAUD	921600	Serial baud
SERIAL_PROTOCOLE	11	Lidar 360 deg
AVOID_ANGLE	1300	Max lean angle
AVOID_BEHAVE	1	Stop when obstacle detected
AVOID_DIST_MAX	5	Max distance avoidance
AVOID_ENABLE	3	Enable drone reaction
AVOID_MARGIN	3	Min distance avoidance

(a) Ardupilot flight controller configuration   (b) Data acquisition acknowledgement

Figure 2.8 – Ardupilot configuration & data acknowledgement

18

2.2.2 Sensors Integration and drone flight

(a) Front left view

(b) Left view
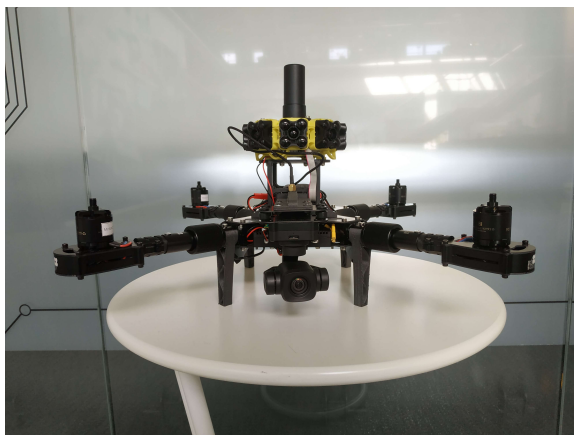
(c) Front view
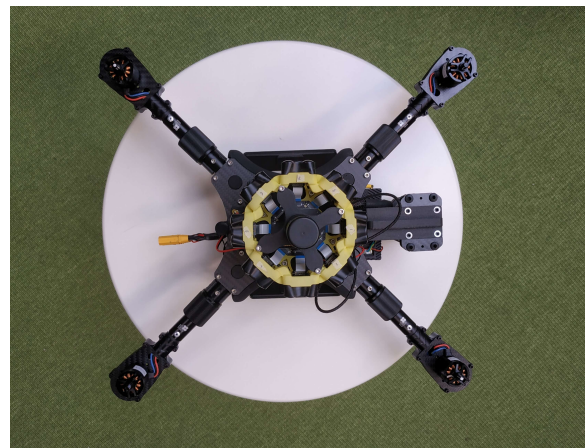
(d) Top view

Figure 2.9 – Larke Mini drone pictures with 360° degrees Lidar mounted

The sensors are not integrated into the hull, requiring a large, bulky tower on top of the Larke Mini drone. The integration with the drone is not aesthetic but functional.

Nevertheless the drone successfully achieved its intended mission objectives. Whether it was aerial mapping or surveying, the drone captured distance data effectively. My colleague Sebastian Duus ensured that the drone was in the optimal position for data acquisition, adjusting altitude and camera settings accordingly.

2.2.3 Python Plotly Visualizer

A graphical user interface is provided, leveraging `Plotly` [6] library. The Logviewer import the `.csv` file as data. Here is the `logfile.csv` header :

timestamp(ms)	PRX.D0	PRX.D45	PRX.D90	PRX.D135	PRX.D180
PRX.D225	PRX.D270	PRX.D315	ATT.Roll	ATT.Pitch	ATT.Yaw

Table 2.1 – Plotly logviewer file header

The application is pretty simple : it use the timestamp as main variable to create animated data. I created 3 main plots in th same window :
— the distance sensors plot
— the roll and pitch attitude
— the yaw plot

The most valuable part of this application is the time slider that can be used to go back in time, pause and restart the view. This way the data analysis is easy-to-use.

2.2.3.1 Final result
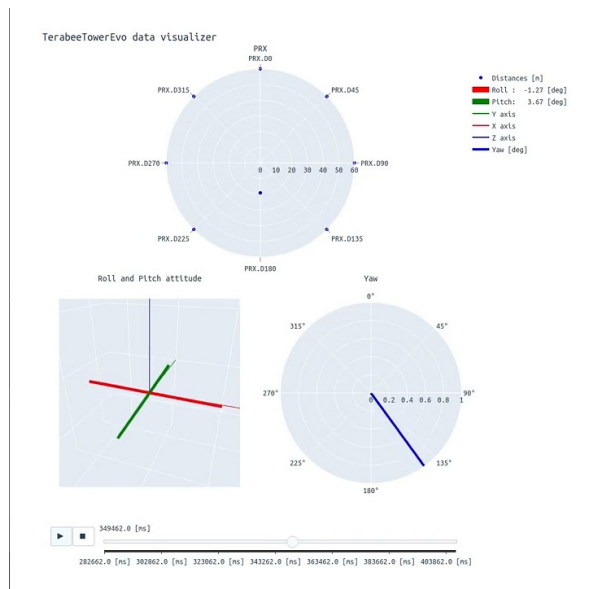
Figure 2.10 – Python LogViewer demo

2.3 Collision avoidance SITL

2.3.1 Environement description

2.3.1.1 Robot Operating System

`ROS2` is an open-source framework for building robotic systems. It supports distributed systems, allowing communication between different components and nodes running on separate devices. It provides a communication infrastructure for exchanging messages and supports various protocols. ROS 2 introduces a quality of service system for controlling communication behavior. It includes real-time capabilities for time-critical applications. It supports multiple programming languages and emphasizes modularity and extensibility. ROS 2 comes with development tools and has an active community and ecosystem. Overall, ROS 2 provides a powerful and versatile platform for developing robotic systems. Its distributed architecture, flexible communication infrastructure, real-time capabilities, language support, and extensibility make it a valuable tool for building complex and advanced robots.

2.3.1.2 Gazebo

`Gazebo` is an open-source 3D simulation environment for robotics. It allows developers to create virtual worlds where they can simulate and test robots. Gazebo provides realistic physics-based simulations, visualizes the environment in 3D, and simulates sensors like cameras and lidars. Users can model robots and their surroundings, integrate with the Robot Operating System (ROS), and extend its functionality using plugins. Gazebo has a supportive community and offers a wide range of resources for developers. In summary, Gazebo is a valuable tool for simulating and evaluating robotic systems before deploying them in real-world environments.

2.3.1.3 Ardupilot

`Ardupilot` is open-source software used for controlling autonomous vehicles like drones. It includes firmware that runs on the vehicle's autopilot hardware, as well as ground control station software for mission planning and monitoring. Ardupilot supports different flight modes, navigation based on waypoints, and telemetry for communication with the vehicle. It is compatible with various hardware platforms and can be customized

and extended. Ardupilot is widely used in drones and other vehicles, and it has a supportive community and extensive documentation. In summary, Ardupilot is a versatile and customizable software suite for autonomous vehicle control.

2.3.2   Built-In Ardupilot collision avoidance

The collision avoidance feature in Ardupilot is designed to enhance the safety and reliability of these vehicles by detecting and avoiding potential collisions with obstacles or other aircraft.

The collision avoidance system in Ardupilot relies on various sensors and algorithms to perceive the environment and make informed decisions about navigation. Some of the key components and techniques used in the Built-In Ardupilot collision avoidance include:

— Sensor Inputs: Ardupilot can interface with different types of sensors, including proximity sensors, lidar, sonar, and cameras. These sensors provide information about the surrounding environment and help in detecting obstacles.

— Obstacle Detection: The collision avoidance system processes the sensor data to identify potential obstacles in the flight path. This can include buildings, trees, other aircraft, or any other objects that may pose a risk of collision.

— Path Planning: Based on the environment map and the current position of the vehicle, Ardupilot generates a safe and collision-free path for the aircraft to follow. It considers factors such as obstacle proximity, vehicle speed, and maneuverability to calculate an optimal trajectory.

— Collision Avoidance Algorithms: Ardupilot employs sophisticated algorithms to predict the future movement of obstacles and determine the best course of action to avoid collisions. These algorithms take into account the dynamics of the vehicle and the obstacles, allowing the autopilot to adjust the flight path accordingly.
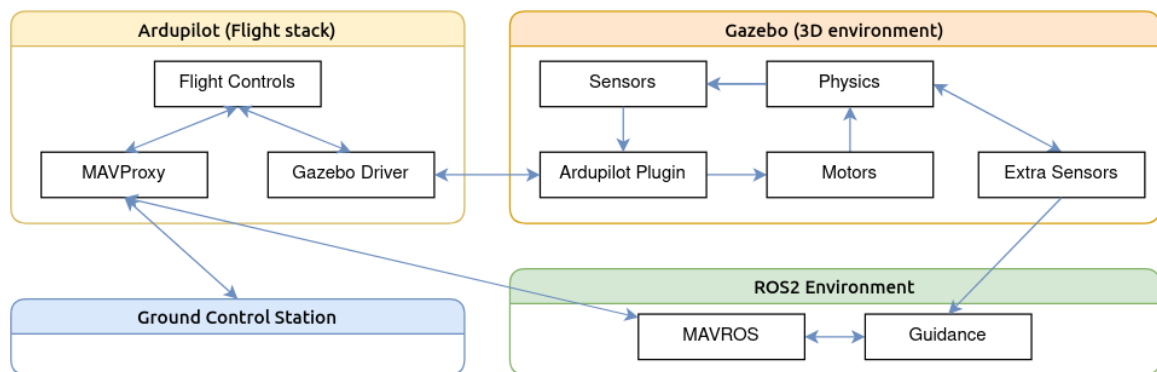
Figure 2.11 – Software-In-The-Loop overall chart[7]

2.3.3 Collision Avoidance ROS2 Migration

2.3.3.1 ROS1 collision-avoidance description

The aim of this task is to migrate a project from the old version of Robot Operating System (ROS) to the new one ROS2.

Oops, I just tried compiling it. Apparently I need to include a bit more. I could go on a bit and tell you more secrets of image placement. Did you know that you can use the `center` environment around the `includeimage` to make things look even prettier? It's true. The control which LATEXprovides with it is nice.

Figure 2.12 – Collision Avoidance flowchart

TEMPORAL ORGANIZATION OF TASKS - GANTT

FILENAME	UPTEKO - Assistant Engineer Intership	COMPANY	Upteko Aps
NAME	Paul Walczak	DATE	24/05/23

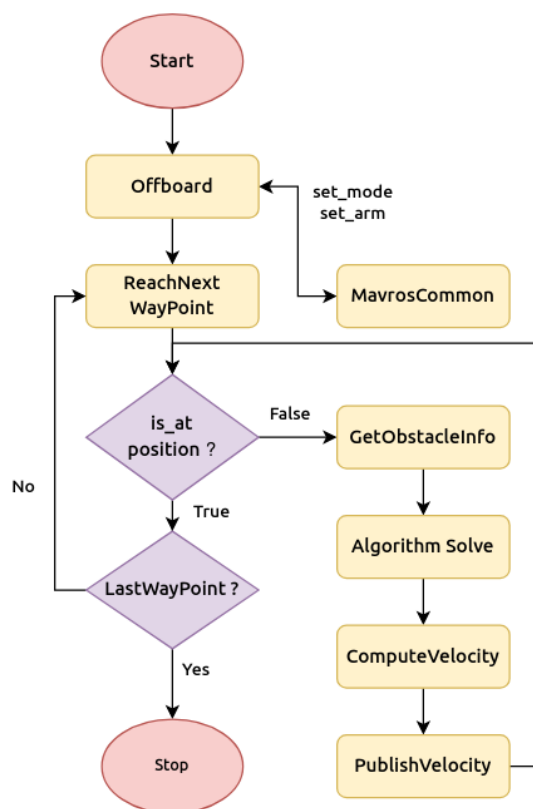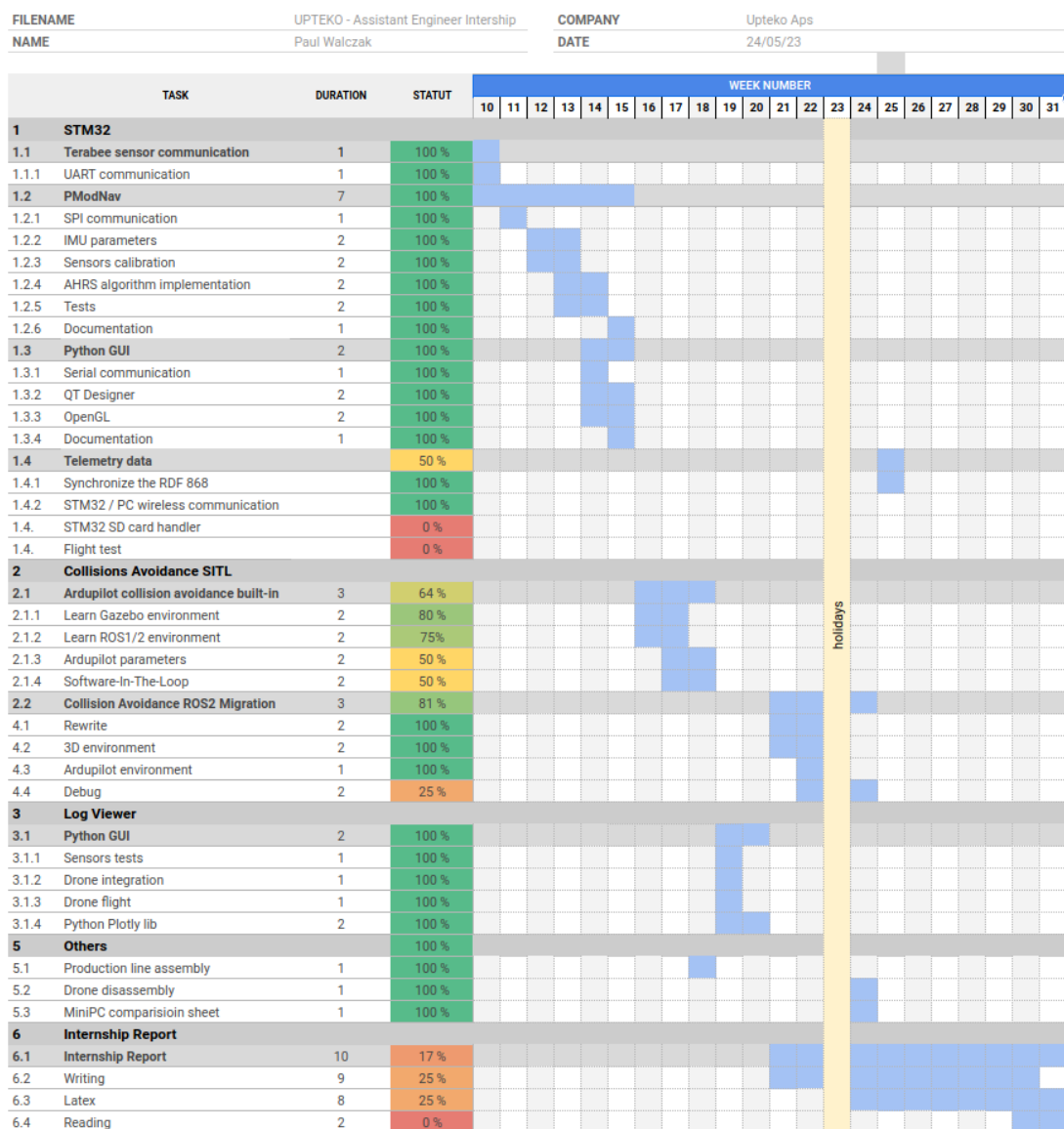|	TASK	DURATION	STATUT	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	**STM32**																								
1.1	**Terabee sensor communication**	1	100 %																						
1.1.1	UART communication	1	100 %																						
1.2	**PModNav**	7	100 %																						
1.2.1	SPI communication	1	100 %																						
1.2.2	IMU parameters	2	100 %																						
1.2.3	Sensors calibration	2	100 %																						
1.2.4	AHRS algorithm implementation	2	100 %																						
1.2.5	Tests	2	100 %																						
1.2.6	Documentation	1	100 %																						
1.3	**Python GUI**	2	100 %																						
1.3.1	Serial communication	1	100 %																						
1.3.2	QT Designer	2	100 %																						
1.3.3	OpenGL	2	100 %																						
1.3.4	Documentation	1	100 %																						
1.4	**Telemetry data**		50 %																						
1.4.1	Synchronize the RDF 868		100 %																						
1.4.2	STM32 / PC wireless communication		100 %																						
1.4.	STM32 SD card handler		0 %																						
1.4.	Flight test		0 %																						
2	**Collisions Avoidance SITL**																								
2.1	**Ardupilot collision avoidance built-in**	3	64 %																						
2.1.1	Learn Gazebo environment	2	80 %																						
2.1.2	Learn ROS1/2 environment	2	75 %																						
2.1.3	Ardupilot parameters	2	50 %																						
2.1.4	Software-In-The-Loop	2	50 %																						
2.2	**Collision Avoidance ROS2 Migration**	3	81 %																						
4.1	Rewrite	2	100 %																						
4.2	3D environment	2	100 %																						
4.3	Ardupilot environment	1	100 %																						
4.4	Debug	2	25 %																						
3	**Log Viewer**																									
3.1	**Python GUI**	2	100 %																						
3.1.1	Sensors tests	1	100 %																						
3.1.2	Drone integration	1	100 %																						
3.1.3	Drone flight	1	100 %																						
3.1.4	Python Plotly lib	2	100 %																						
5	**Others**		100 %																						
5.1	Production line assembly	1	100 %																						
5.2	Drone disassembly	1	100 %																						
5.3	MiniPC comparisioin sheet	1	100 %																						
6	**Internship Report**																									
6.1	**Internship Report**	10	17 %																						
6.2	Writing	9	25 %																						
6.3	Latex	8	25 %																						
6.4	Reading	2	0 %																						

Figure 3.1 – Gantt chart

BIBLIOGRAPHY

[1]   Digilent. « Vivado pmodnav c library ». (), [Online]. Available: `https://github.com/Digilent/vivado-library/tree/master/ip/Pmods/PmodNAV_v1_0/drivers/PmodNAV_v1_0`.

[2]   OpenSource. « Gimbal lock explained ». (), [Online]. Available: `https://en.wikipedia.org/wiki/Gimbal_lock`.

[3]   x-ioTechnologies. « Open source imu and ahrs algorithms - x-io technologies ». (), [Online]. Available: `https://x-io.co.uk/open-source-imu-and-ahrs-algorithms/`.

[4]   QtCompany. « Pyqt5 documentation ». (), [Online]. Available: `https://www.riverbankcomputing.com/static/Docs/PyQt5/`.

[5]   OpenGLGroup. « Pyopengl documentation ». (), [Online]. Available: `https://mcfletch.github.io/pyopengl/index.html`.

[6]   PlotlyGroup. « Plotly official website ». (), [Online]. Available: `https://plotly.com/`.

[7]   IntelligentQuads. « Sitl explained ». (), [Online]. Available: `https://www.youtube.com/watch?v=W2ncr0DKWHE&ab_channel=IntelligentQuads`.

ANNEXES

bla bla bla

Stagiaire: Paul WALCZAK
Semestre: Semestre 8
Entreprise: Upteko ApS
Tuteur Entreprise:Vincent KLYVERTS TOFTERUP
Tuteur ENIB: Alexis MICHEL

Sujet : «Mon Sujet»

Mot clés : «de 3 à 6 mots clefs»

Entreprise : Eius populus ab incunabulis primis ad usque pueritiae tempus extremum, quod annis circumcluditur fere trecentis, circummurana pertulit bella, deinde aetatem ingressus adultam post multiplices bellorum aerumnas Alpes transcendit et fretum, in iuvenem erectus et virum ex omni plaga quam orbis ambit inmensus, reportavit laureas et triumphos, iamque vergens in senium et nomine solo aliquotiens vincens ad tranquilliora vitae discessit. Hoc inmaturo interitu ipse quoque sui pertaesus excessit e vita aetatis nono anno atque vicensimo cum quadriennio imperasset. natus apud Tuscos in Massa Veternensi, patre Constantio Constantini fratre imperatoris, matreque Galla. Thalassius vero ea tempestate praefectus praetorio praesens ipse quoque adrogantis ingenii.

Travaux Réalisés : Eius populus ab incunabulis primis ad usque pueritiae tempus extremum, quod annis circumcluditur fere trecentis, circummurana pertulit bella, deinde aetatem ingressus adultam post multiplices bellorum aerumnas Alpes transcendit et fretum, in iuvenem erectus et virum ex omni plaga quam orbis ambit inmensus, reportavit laureas et triumphos, iamque vergens in senium et nomine solo aliquotiens vincens ad tranquilliora vitae discessit. Hoc inmaturo interitu ipse quoque sui pertaesus excessit e vita aetatis nono anno atque vicensimo cum quadriennio imperasset. natus apud Tuscos in Massa Veternensi, patre Constantio Constantini fratre imperatoris, matreque Galla. Thalassius vero ea tempestate praefectus praetorio praesens ipse quoque adrogantis ingenii, considerans incitationem eius ad multorum augeri discrimina, non maturitate vel consiliis mitigabat, ut aliquotiens celsae potestates iras principum molliverunt, sed adversando iurgandoque cum parum congrueret, eum ad rabiem potius evibrabat, Augustum actus eius exaggerando creberrime docens, idque, incertum qua mente, ne lateret adfectans. quibus mox Caesar acrius efferatus, velut contumaciae quoddam vexillum altius erigens, sine respectu salutis alienae vel suae ad vertenda opposita instar rapidi fluminis irrevocabili impetu ferebatur. Hae duae provinciae bello quondam piratico catervis mixtae praedonum.

Conclusion : Eius populus ab incunabulis primis ad usque pueritiae tempus extremum, quod annis circumcluditur fere trecentis, circummurana pertulit bella, deinde aetatem ingressus adultam post multiplices bellorum aerumnas Alpes transcendit et fretum, in iuvenem erectus et virum ex omni plaga quam orbis ambit inmensus, reportavit laureas et triumphos, iamque vergens in senium et nomine solo aliquotiens vincens ad tranquilliora vitae discessit. Hoc inmaturo interitu ipse quoque sui pertaesus excessit e vita aetatis nono anno atque vicensimo cum quadriennio imperasset. natus apud Tuscos in Massa Veternensi, patre Constantio Constantini fratre imperatoris, matreque Galla.