



UV 3.4 - Projet IQX

—

Rapport d' avancement

Nicolas VEYLON
Sophie TUTON
Joris TILLET
Mathieu M.
Mohamed Amine OUADRHIRI
Yacine BENHNINI

—

Promotion 2018

9 janvier 2017

Table des matières

Table des matières.....	1
Introduction.....	2
Lexique / Abréviations utilisées.....	3
Table des figures.....	4
I. Analyse fonctionnelle.....	5
I.1. Présentation du projet	5
I.2 Détails des fonctionnalités et des flux.....	6
I.2.1 Détails des fonctionnalités et exigences.....	6
I.2.2 Caractérisation des fonctions.....	10
I.3. La routine de vol	11
I.4 La routine de pilotage automatique.....	12
II. Etat de l'art et choix	13
II.1. Bloc propulsion	13
II.1.1 Choix d'architecture	13
II.1.2 Moteurs, hélices et batterie.....	14
II.2. Contrôleur de vol	17
II.2.1 Explications	17
II.2.2 Contrôleur de vol choisi	18
II.3. Carte mère	19
II.4 Communications	20
II.5. Interface graphique.....	21
II.6. Exploration automatique.....	21
II.7. Algorithmes de SLAM	24
III. Architecture physique	25
IV. La conduite du projet.....	0
IV.1. Identification du travail à effectuer	0
IV.2. Fonctionnement du groupe	0
Conclusion	2
Bibliographie	3
Annexe	5

Introduction

Le projet IQX, IQX pour Indoor Quadrirotor X-ploration, est un projet industriel de robotique. Il consiste à concevoir un drone pouvant évoluer à l'intérieur d'un bâtiment inconnu en se repérant et en fournissant la carte à l'utilisateur. Il est la suite directe du projet QX de l'année 2015-2016 [1].

L'objectif est donc, en partant du drone conçu l'année dernière pour le projet QX, d'arriver à réaliser un robot conciliant tous ces objectifs en tenant compte des nombreuses contraintes liées au fait de voler en intérieur : la plupart des signaux (GPS, wifi, ...) ne passent pas, l'appareil doit être suffisamment petit pour passer des portes standards, le moindre choc peut provoquer un crash irréversible, etc.

Ce projet s'inscrit également dans l'optique de participer au concours euRathlon 2017. Ce concours fait partie de l'ERL (European Robotics League) et s'intitule l'ERL Emergency Robots 2017. Il aura lieu en septembre 2017 et requiert la conception de plusieurs robots (drones, sous-marins, chars) devant réaliser des tâches assez variées : exploration, cartographie, repérage, actions, etc [2]. Le projet IQX n'est donc qu'une partie des requis pour ce concours. Mais, d'autre part, cela rajoute des exigences pour le projet puisque le drone conçu doit répondre aux contraintes du règlement du concours.

Ce rapport d'avancement présente donc en détail toutes les exigences et contraintes que nous avons définies pour répondre aux nombreux besoins du projet. Ainsi, nous présentons dans ce rapport les résultats de nos études d'ingénierie système, à savoir l'analyse fonctionnelle, les différents états de l'art réalisés et enfin l'architecture physique élaborée pour remplir tous les objectifs.

Lexique / Abréviations utilisées

- ESC : Electronic Speed Control
- I2C : Inter-Integrated Circuit
- IQX : Indoor Quadri-rotor X-ploration
- KV : Constante de vélocité. $1KV = 1 \text{ tour.min}^{-1}.V^{-1}$
- LIDAR = LIght Detection And Ranging
- MAJ = Mise à jour
- PC : Personal Computer
- RAM : Random Access Memory (Mémoire vive)
- ROS : Robot Operating System
- SLAM : Simultaneous Localization And Mapping

Table des figures

Diagramme 1 : Bête à corne IQX	5
Diagramme 2 : Diagramme pieuvre IQX	6
Diagramme 3 : Analyse fonctionnelle avec flux	9
Diagramme 4 : Grafcet de la routine de vol	11
Diagramme 5 : Grafcet de la routine du pilotage automatique	12
Diagramme 6 : Description de l'architecture physique	25
Diagramme 7 : Visualisation des travaux à effectuer sur l'architecture physique	0
Tableau 1 : Détails de la fonctionnalité "Cartographier un bâtiment"	6
Tableau 2 : Détails de la fonctionnalité "Pouvoir voler"	7
Tableau 3 : Détails de la fonctionnalité "Visualiser les données"	7
Tableau 4 : Détails des contraintes	8
Tableau 5 : Description des performances attendues	10
Tableau 6 : Comparatif des contrôleurs de vol	18
Tableau 7 : Comparatif pour la carte mère	19
Tableau 8 : Choix du langage de programmation pour l'interface graphique	21
Tableau 9 : Détail des solutions envisagées	1
Tableau 10 : Les travaux à effectuer pour « Pouvoir Voler » (PV)	2
Tableau 11 : Les travaux à effectuer pour "Etablir une cartographie" (CB)	2
Tableau 12 : Les travaux effectuer pour "Visualiser les données" (VD)	2
Figure 1 : Hexacopter	13
Figure 2 : Octocopter	13
Figure 3 : Tricopter	13
Figure 4 : Quadricopter	13
Figure 5 : Explication Angle d'attaque et pas	14
Figure 6 : eCalc pour la première configuration de propulsion [7]	16
Figure 7 : eCalc pour la deuxième configuration de propulsion [7]	16
Figure 8 : Rendu visuel LIDAR + SLAM	22
Figure 9: Simulation en cours	23
Figure 10: Simulation terminée	23
Graphique 1 : Relation entre poussée et courant	16

I. Analyse fonctionnelle

I.1. Présentation du projet

Dans le cadre du concours Eurathlon, le système IQX doit cartographier l'intérieur d'un bâtiment. Voici un premier diagramme bête à corne décrivant sommairement notre système :

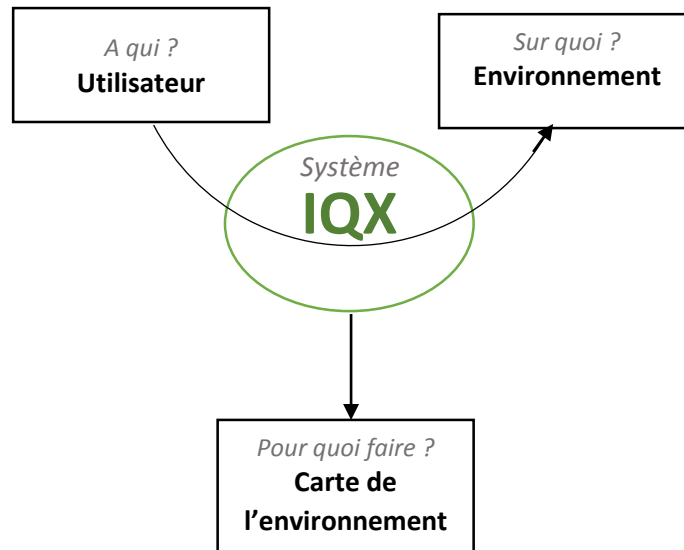
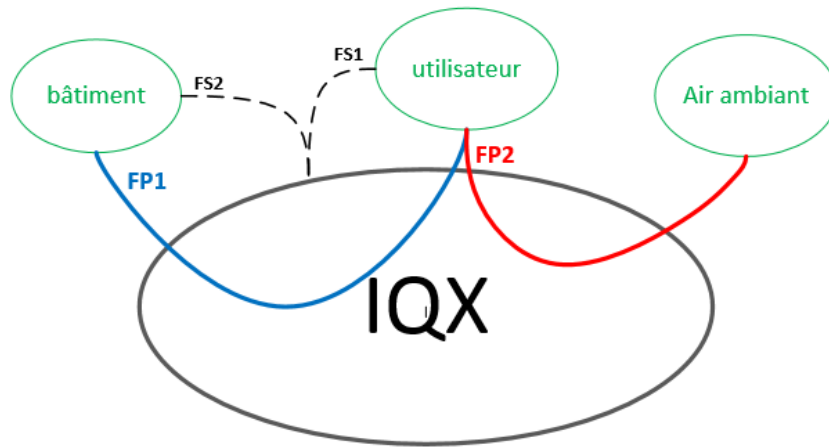


Diagramme 1 : Bête à corne IQX

Ainsi, il doit pouvoir à la fois établir la cartographie et la fournir à l'utilisateur. Mais la présence du drone dans le bâtiment peut engendrer des problèmes de communication entre le drone et l'utilisateur.

C'est pourquoi le drone doit être capable d'être autonome ; et lorsque le drone est piloté par l'utilisateur, ce dernier doit pouvoir apprécier l'état du drone pour prendre des décisions convenables. Il est donc nécessaire d'avoir une visualisation des données du drone lorsque c'est possible. Pour mieux comprendre les fonctions de notre système, voici le diagramme pieuvre associé :



FP1 : Etablir et fournir la cartographie de l'intérieur d'un bâtiment.

FP2 : voler dans l'air ambiant comme dicté par l'utilisateur ou de manière autonome.

FS1 : Permettre la visualisation des données du drone et de la cartographie.

FS2 : Evoluer dans le bâtiment en toute sécurité.

Diagramme 2 : Diagramme pieuvre IQX

Il ne faudra pas oublier que notre drone doit vérifier le règlement du concours Eurathlon.

I.2 Détails des fonctionnalités et des flux

I.2.1 Détails des fonctionnalités et exigences

FP1 : Pour cartographier le bâtiment dans lequel se trouve le drone, il faut acquérir des mesures de distance des murs par rapport au drone. Ces données doivent ensuite être traitées pour permettre d'établir la carte du bâtiment (CB-F402), et se localiser à partir de cette carte (CB-U401). Ces données doivent être enregistrées (CB-F201) et fournies (VD-E301) à une interface pour être visualisées par l'utilisateur (VD-F202 et VD-F103).

Pour plus d'informations sur la méthode d'identification des fonctions, cf. Annexes : logique des identifiants.

Fonction principale/ de service	Identifiant exigences	Type/ Nature	Importance	Détail exigence			
				Sous exigence 1	Sous exigence 2	Sous exigence 3	Sous exigence 4
Cartographier un bâtiment	CB-F201	Fonctionnalité	2	Produire une carte lisible d'un bâtiment inconnu	Stocker la carte en mémoire		
	CB-F402	Fonctionnalité	4		Produire une carte précise		
	CB-U401	Utilisabilité	4	Se localiser dans une pièce inconnue			

Tableau 1 : Détails de la fonctionnalité "Cartographier un bâtiment"

Ici, nous attendons une carte où sont représentés les objets et les murs détectés.

FP2 : Pouvoir voler signifie ici de pouvoir effectuer un vol stationnaire i.e. sans déplacement (PV-F005), d'atterrir (PV-F404), de pouvoir porter l'équipement (PV-403) et de pouvoir se déplacer en sécurité (PV-F401, PV-F102, PV-U301/2).

Ici on entend par « Se déplacer en sécurité » le fait de faire attention aux obstacles, et de les éviter pendant que le drone se déplace, et cela même pour une procédure de vol manuel. La notion de sécurité est aussi prise en compte à l'aide du réglage du failsafe (réaction automatique du drone quand la télécommande s'éteint).

« Pouvoir porter l'équipement » signifie que le drone doit pouvoir exécuter un vol stationnaire en prenant en charge tous les instruments intégrés dessus (capteurs, châssis, bloc propulsion...). Comme nous le verrons dans les parties suivantes, le poids total du drone (équipement compris) serait au maximum de 4kg.

Fonction principale/ de service				Détail exigence			
Identifiant exigences	Type/ Nature	Importance	Sous exigence 1	Sous exigence 2	Sous exigence 3	Sous exigence 4	
Pouvoir voler	PV-F401	Fonctionnalité	4	Se déplacer en sécurité	Etre Télé-opérable	Pilotage semi-automatique	Via télécommande
	PV-F102	Fonctionnalité	1				Via pc
	PV-U301	Utilisabilité	3			Pilotage automatique	Choisir la nouvelle destination
	PV-U302	Utilisabilité	3				Se localiser dans une pièce inconnue
				Eviter les obstacles	Repérer les obstacles		
	PV-F403	Fonctionnalité	4	Pouvoir porter l'équipement			
	PV-F404	Fonctionnalité	4	Atterrir			
	PV-F405	Fonctionnalité	4	Effectuer un vol stationnaire			

Tableau 2 : Détails de la fonctionnalité "Pouvoir voler"

Les données de vol et la cartographie doivent être acquises et visualisées sur une interface graphique. Les détails se trouvent dans le tableau ci-dessous.

Fonction principale/ de service	Identifiant exigences	Type/ Nature	Importance	Détail exigence			
				Sous exigence 1	Sous exigence 2	Sous exigence 3	Sous exigence 4
Visualiser les données	VD-E301	Ergonomie	3	Acquérir les données	En temps réel		
	VD-F201	Fonctionnalité	2		En temps décalé		
	VD-E202	Ergonomie	2	Avoir une IHM	Avoir une interface claire		
	VD-E103	Ergonomie	1		Présenter l'état du drone	Afficher l'état de la batterie	Prendre en compte le PNR
	VD-E104	Ergonomie	1		Afficher qualité du signal	Wifi	
	VD-E105	Ergonomie	1			Télécommande	
	VD-E106	Ergonomie	1		Afficher le mode courant	Mode de vol	
	VD-E107	Ergonomie	1			Etat de la fonction cartographique	
	VD-E108	Ergonomie	1		Afficher les données des capteurs	Altitude/ Assiette etc.	
	VD-E109	Ergonomie	1			Vitesse	
	VD-E110	Ergonomie	1		Afficher les événements	Afficher les pannes	
	VD-E111	Ergonomie	1			Afficher les manœuvres d'évitement	
	VD-F302	Fonctionnalité	3		Présenter la cartographie	Afficher les murs	
	VD-F303	Fonctionnalité	3			Afficher la position du drone	

Tableau 3 : Détails de la fonctionnalité "Visualiser les données"

Mais il ne faut pas oublier de prendre en compte les contraintes imposées, notamment par les règles du concours Eurathlon.

Fonction principale/ de service	Identifiant exigences	Type/ Nature	Importance	Détail exigence			
				Sous exigence 1	Sous exigence 2	Sous exigence 3	Sous exigence 4
Fonction de contraintes	FC-E201	Ergonomie	2	Etre robuste			
	FC-E402	Ergonomie	4	Etre stable			
	FC-E303	Ergonomie	3	Etre facilement pilotable	Avoir un mode semi-automatique		
	FC-F301	Fonctionnalité	3	Etre capable de passer des portes standards	Ne pas heurter		
	FC-E304	Ergonomie	3		Etre stable		
	FC-F402	Fonctionnalité	4	Avoir une procédure de sécurité			
	FC-F303	Fonctionnalité	3	Séparer les alimentations	Avoir deux batteries	Moteur	
	FC-F304	Fonctionnalité	3			Système embarqué	
	FC-C401	Contrainte	4	Etre en accord avec la législation			
	FC-C402	Contrainte	4	Etre en accord avec le règlement d'Eurathlon 2017			

Tableau 4 : Détails des contraintes

Ici, le mode semi-automatique fait référence à un pilotage manuel accompagné d'un évitement d'obstacle.

La procédure de sécurité se traduit par une télécommande, différente de celle servant pour le mode semi-automatique, qui permet à tout moment d'arrêter le drone à distance en cas de danger imminent. Elle est indispensable pour le concours Eurathlon.

Nous avons ensuite détaillé ces différentes fonctions. Pour se faire, nous nous sommes servi de ce qui existait déjà sur le drone, nous avons étudié chaque partie, et enfin, nous avons réalisé des états de l'art pour faire les bons choix de composants qui réaliseront les différentes fonctions trouvées. Le détail se trouve sur le diagramme suivant avec - **en bleu** - les fonctions permettant d'effectuer et de fournir une cartographie à l'utilisateur, et - **en rouge** - les fonctions permettant de voler. Ainsi on voit différents blocs apparaître communiquant par des flux.

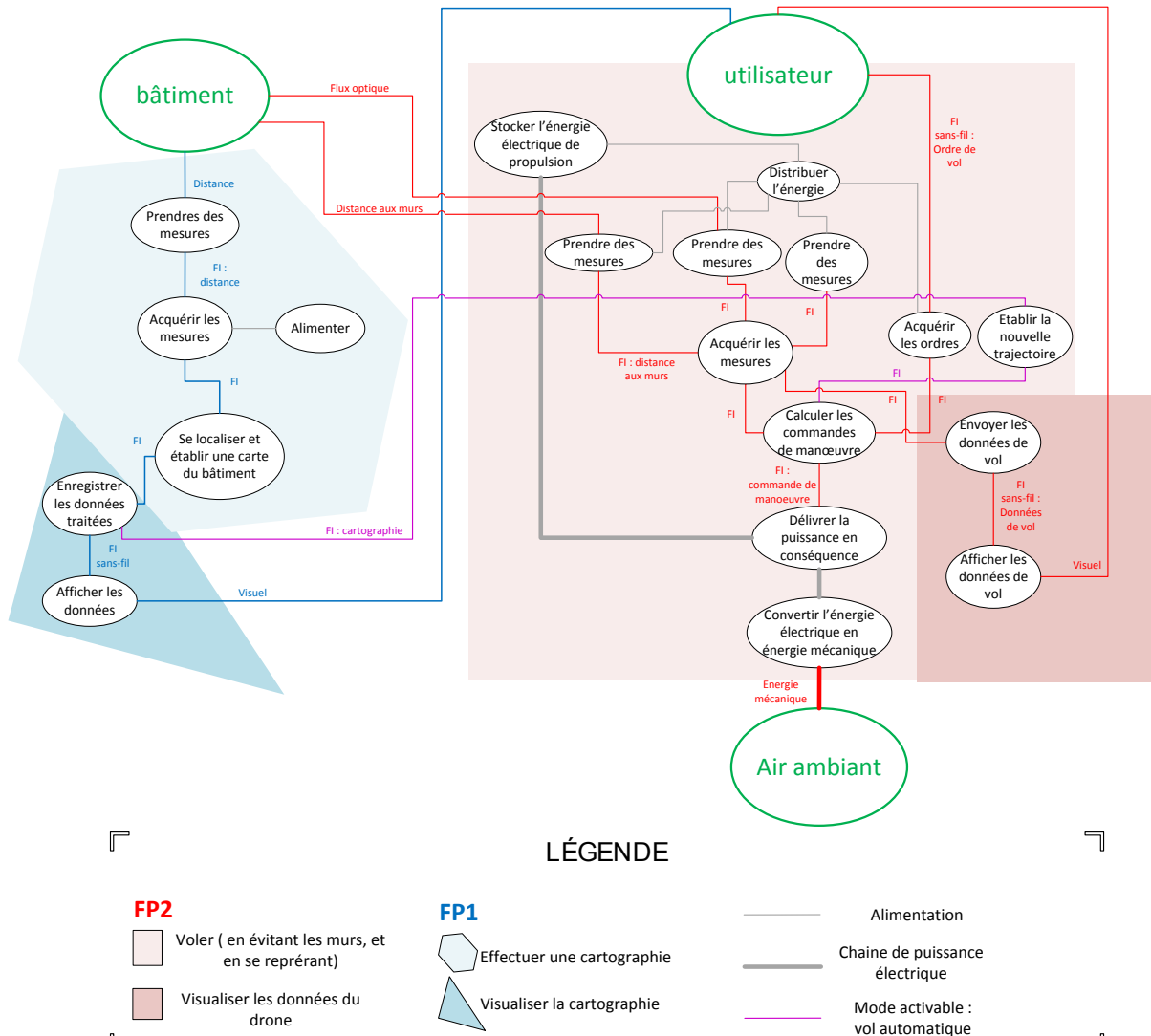


Diagramme 3 : Analyse fonctionnelle avec flux

I.2.2 Caractérisation des fonctions

Nous avons détaillé les performances nécessaires dans le tableau ci-dessous.

Fonction	Sous fonction		Performance nécessaire
Cartographier un bâtiment (CB)	Produire une carte lisible	Stocker la carte en mémoire	Capacité de 4 GB
		Produire une carte précise	Portée de 5m, précision <1°
	Se localiser dans une pièce inconnu		
Pouvoir Voler (PV)	Se déplacer	Etre télé-opérable	Portée 40m
	Pouvoir porter les équipements et rester manœuvrable		Poussée > 4kg
	Effectuer un vol stationnaire		
	Atterrir		
Visualiser les données (VD)	Acquérir les données	En temps réel	Transmission wifi Ping<250ms
		En temps décalé	Données récupérables
	Avoir une IHM	Une interface claire	
		Afficher la qualité des signaux	
		Afficher les modes de vol	
		Afficher les données des capteurs	
		Afficher les événements	
		Présenter la cartographie	
Fonctions de contrainte (F)	Etre robuste		
	Etre stable		
	Etre capable de passer les portes standards		Largeur de 70 cm et 2m de hauteur
	Pouvoir agir en autonomie		Durée < durée de vol
	Durée de vol suffisante		Vol mixte >10min

Tableau 5 : Description des performances attendues

I.3. La routine de vol

Notre drone, initialement à l'arrêt au sol, doit décoller lorsque l'utilisateur lui donne l'ordre correspondant. Par défaut, il maintiendra un vol stationnaire. A partir de ce moment : s'il reçoit l'ordre d'atterrir, alors il lance la procédure d'atterrissage et revient donc dans l'état initial à l'arrêt au sol, s'il reçoit un ordre de déplacement il exécute l'ordre de déplacement puis ce remet en vol stationnaire, et pour finir, s'il reçoit l'ordre d'acquies il lance la procédure d'acquisition puis ce remet en vol stationnaire.

Donc le drone suit la routine de vol correspondant au grafct suivant, inspiré de celui de nos prédécesseurs [1].

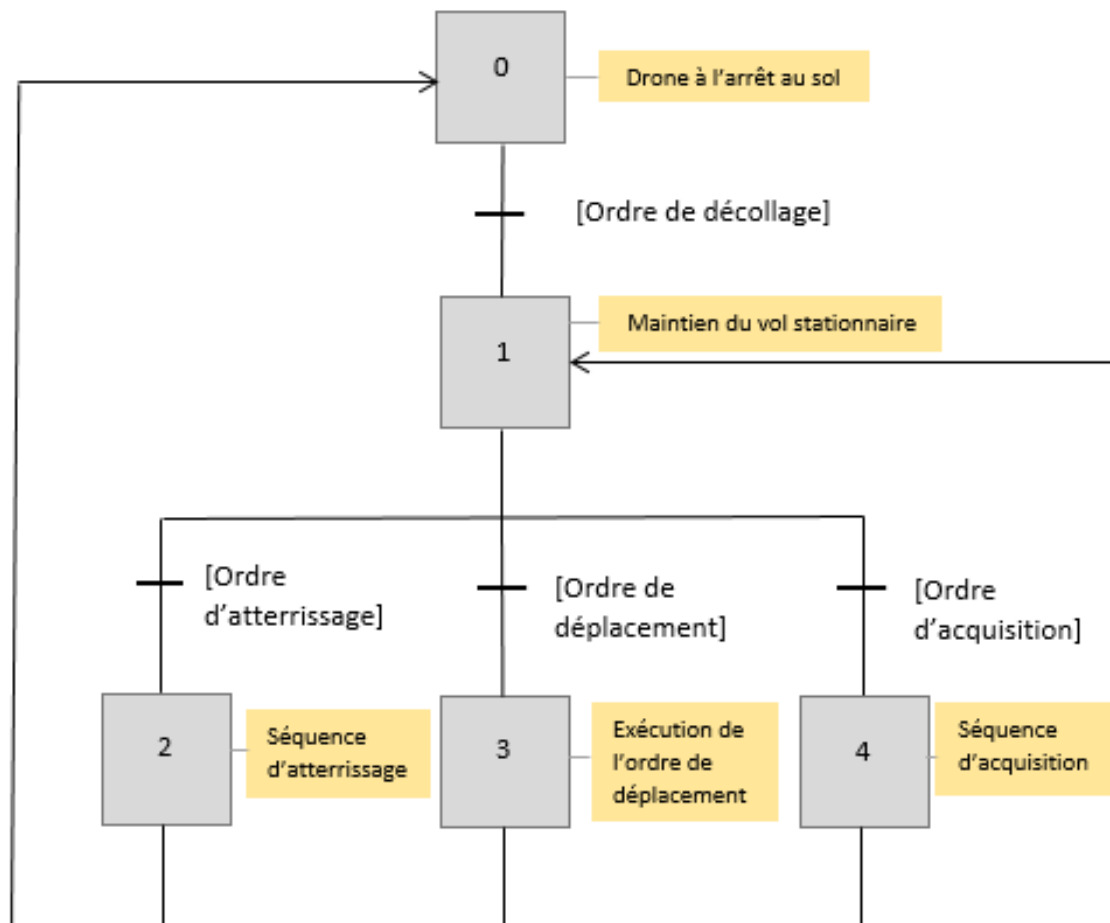


Diagramme 4 : Grafct de la routine de vol

I.4 La routine de pilotage automatique

Au démarrage, le drone se place à l'altitude de vol que l'on veut puis commence à effectuer sa routine automatique de vol. Premièrement, la carte actuelle est mise à jour avec le reçu du lidar, puis l'algorithme d'exploration met à jour les deux cartes qu'il utilise : la carte normalisée pour ses calculs de distances, et la carte des obstacles. Ensuite, il choisit la prochaine destination, s'y déplace et recommence jusqu'à ce que la carte soit complète. A ce moment, il n'y aura plus de destination éligible et le drone revient à son point de départ. L'algorithme est résumé plus en détail en II.6.. En cas d'interruption quelconque par l'utilisateur, le drone repasse en mode manuel.

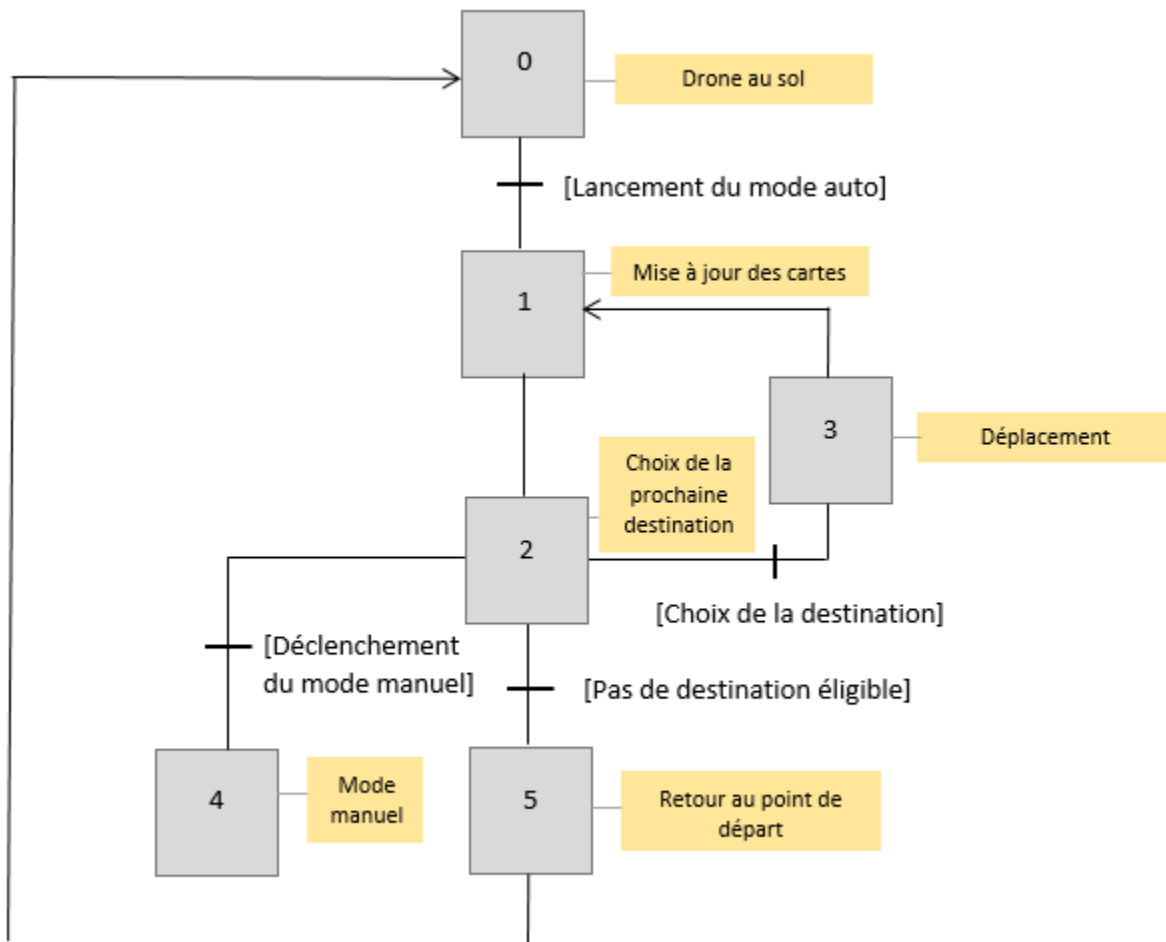


Diagramme 5 : Grfcet de la routine du pilotage automatique

II. Etat de l'art et choix

II.1. Bloc propulsion

II.1.1 Choix d'architecture

Un drone multirotor possède en pratique entre 3 et 8 hélices. Le choix de la solution ne reposera pas sur leur complexité car le contrôle est assuré par un équipement électronique appelé contrôleur de vol. Le choix se fera donc sur la charge utile, sur la taille et la stabilité qu'offre chacune des solutions.

Notre drone devant pouvoir franchir des portes, nous excluons les hexacopters et octocopters qui sont bien trop larges malgré l'avantage qu'ils présentent d'avoir une redondance (= doublage des systèmes électroniques embarqués) qui permet de réduire les risques de pannes[3].



Figure 1 : Hexacopter



Figure 2 : Octocopter

Ensuite, un quadricopter offre plus de charge utile qu'un tricopter. Mais nous avons choisi de nous orienter sur un quadricopter puisqu'il est bien plus stable, qu'un châssis de bonne qualité a déjà été acheté par les membres du projet QX, et que ce type de structure est pris en charge par tous les contrôleurs.



Figure 3 : Tricopter



Figure 4 : Quadricopter

Visuel : MissionPlanner

II.1.2 Moteurs, hélices et batterie

II.1.2.1 Explications [4]–[6]

- Moteurs :

La caractéristique principale d'un moteur est le KV. C'est le nombre de tours par minute par volt du rotor. Plus le moteur possède un KV important, plus il tourne vite, donc plus il est nerveux, mais moins son couple est grand. On privilégie toujours un moteur avec un faible KV (inférieur à 1000) lorsqu'on cherche un drone stable, ce qui est le cas pour notre drone.

Les moteurs ont également une puissance indiquée. Le poids du moteur augmente avec la puissance du moteur. Ce poids s'ajoute à la masse à soulever par la propulsion. Il faut donc plus de poussée : c'est pour cela que les dimensions des hélices (diamètre et pas) conseillées augmentent avec la puissance du moteur.

- Hélices :

Une hélice est caractérisée par son diamètre ainsi que par son pas, donnés en pouce (hélice : *diamètre x pas*). Plus ses dimensions sont grandes, plus la poussée est grande, en supposant que le moteur soit assez puissant. En effet, de grandes hélices engendrent beaucoup de frottement avec l'air. Comme le moteur est tenu de tourner à vitesse constante pour une tension donnée (du fait du KV), et comme des hélices plus grandes demandent au moteur plus de puissance (pour vaincre les frottements), d'après $Puissance = Tension * Intensité$, les moteurs consommeront plus de courant. Il est donc nécessaire d'avoir un moteur puissant lorsqu'on utilise de grandes hélices.

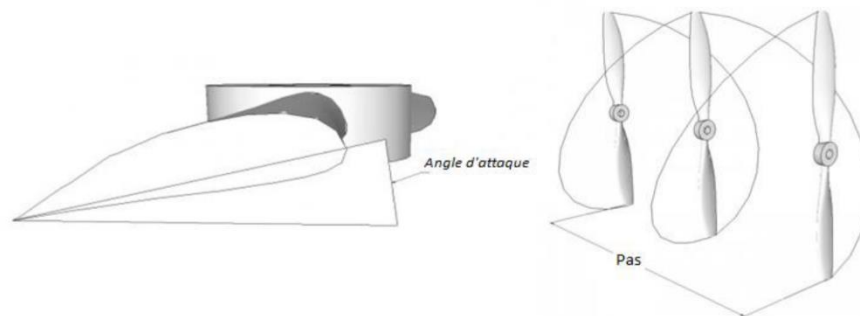


Figure 5 : Explication Angle d'attaque et pas

Les hélices peuvent avoir en pratique entre 2 et 4 pâles. Augmenter le nombre de pâles est utile si on veut des hélices plus compactes, ce qui est le cas pour notre projet car notre drone doit pouvoir franchir des portes standards [6].

- Batterie LiPo :

Les batteries LiPo (Lithium Polymère) sont largement utilisées avec les drones. C'est de loin le type de batterie qui offre le meilleur compromis capacité-poids. Elles sont notamment à l'origine de la rapide démocratisation des drones hélicoptères depuis une dizaine d'années car il fut enfin possible d'avoir suffisamment de puissance en un poids réduit. Une LiPo est caractérisée par :

- Sa capacité (en mAh) : c'est la charge maximale de la batterie. Une batterie de 5000mAh = 5Ah peut délivrer un courant de 5 ampères pendant une heure.
- Son nombre de S : c'est le nombre de cellules 3.7V mises en série constituant la batterie. Chaque cellule délivre du 3.7V. Une batterie 3S délivre donc une tension continue de 11.1V.
- Son nombre de C : c'est le pouvoir de décharge d'une batterie. Deux valeurs de C sont indiquées, le premier est le pouvoir de décharge en continue, le deuxième en *burst* (pic d'ampérage, par exemple lors de l'allumage des moteurs). Une batterie 5000mAh 10-20C peut délivrer $5\,000 \times 10 = 50\,000\text{mA} = 50\text{A}$ en continue, et 100A en *burst*.

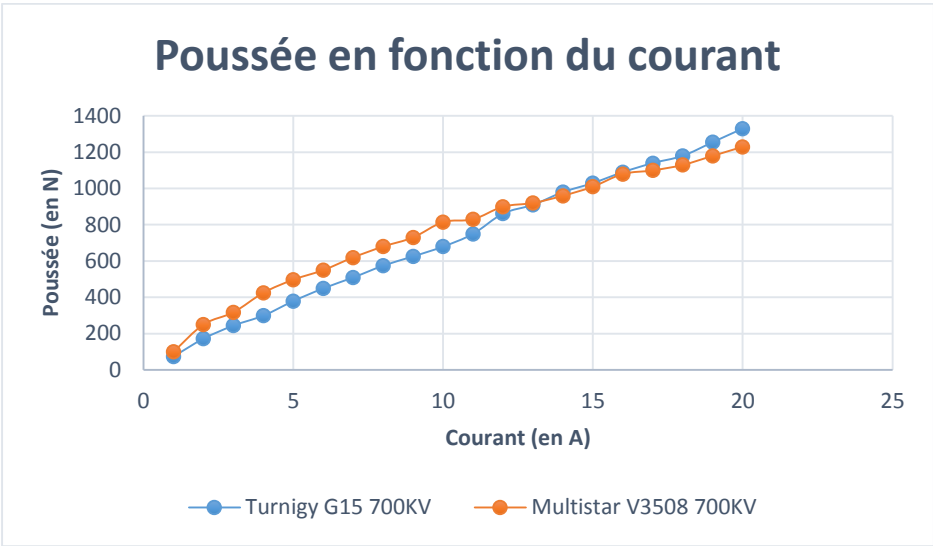
Plus la capacité d'une batterie est grande, plus celle-ci est lourde (700g pour 8000mAh, 1.5kg pour une 16000mAh). Comme le drone doit soulever cette batterie, il consomme plus (d'ampère) avec une batterie lourde, mais comme les batteries lourdes offrent plus de capacité, on note que la durée de vol augmente avec la capacité de la batterie (malgré la consommation supérieure). Cependant, une batterie plus lourde fait diminuer le rapport $\frac{\text{Poussée}_{\text{max}}}{\text{Poids}}$. En effet, si chaque moteur possède une poussée max de 2kg, soit une poussée totale maximale de 8kg, on estime que le drone sera facilement manœuvrable s'il pèse moins de 4kg (soit un rapport $\frac{\text{Poussée}_{\text{max}}}{\text{Poids}}$ supérieur à 2:1) [5]. Pour conclure, il faut trouver la batterie qui offre le meilleur compromis entre manœuvrabilité et autonomie.

Enfin, il est possible de mettre plusieurs batteries en parallèle pour augmenter l'autonomie (mais diminuer encore la manœuvrabilité). Une batterie 3S2P est en fait deux mêmes batteries 3S mises en parallèle.

II.1.2.2. Etude des choix QX.

Les moteurs du projet QX sont les Turnigy G15 710KV. Ces moteurs ont une puissance élevée et sont donc lourds. Cependant, la structure du drone empêche d'avoir de grandes hélices (10x4.5 au lieu de 12x6 conseillée pour ces moteurs) et empêche donc d'utiliser le maximum du potentiel des moteurs. Les moteurs sont donc surdimensionnés par rapport aux hélices, ils ne pouvaient offrir un bon rapport poussée/poids au drone QX.

Des mesures ont été faites sur le banc d'essai du club robotique. On constate qu'un moteur moins puissant offre autant de poussée quelque soit le courant. Les mesures n'ont pu être effectuées que sur la plage [0A,20A].



Graphique 1 : Relation entre poussée et courant

Configurations retenues

Drone QX de base	
Moteurs	Turnigy G15 710KV
Hélices	10x4.5 tripale
Batterie	16000mAh 4S 10C

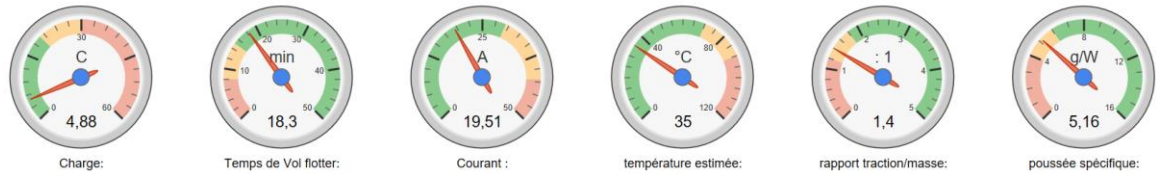


Figure 6 : eCalc pour la première configuration de propulsion [7]

Drone IQX	
Moteurs	Turnigy G15 710KV
Hélices	11x7 tripale
Batterie	16000mAh 4S 10C

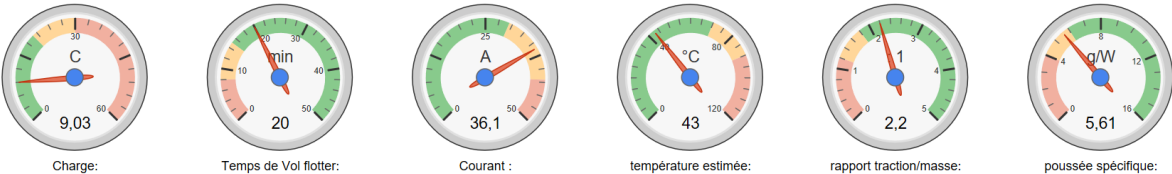


Figure 7 : eCalc pour la deuxième configuration de propulsion [7]

Comme notre drone doit avoir une bonne autonomie malgré son poids, nous avons choisi d'utiliser les mêmes moteurs que nos prédécesseurs, le Turnigy G15 700KV. Cependant, faisant suite à l'étude plus haut, nous utiliserons des hélices plus grandes et avec un plus grand pas, permettant de libérer toute la puissance de ces imposants moteurs. Cette nouvelle configuration offre plus d'autonomie (20 minutes

contre 18 pour le drone QX) et un bien meilleur rapport poussée/poids qui maintenant dépasse 2:1, supprimant ainsi le problème de surcharge du drone QX.

On constate que la batterie est très sollicitée, mais est théoriquement bien dimensionnée pour tenir. Il faudra attendre l'arrivée des hélices pour conclure. Dans tous les cas, il est possible d'utiliser des batteries en parallèle, plus petites, mais offrant bien plus de décharge.

II.2. Contrôleur de vol

II.2.1 Explications

Le contrôleur de vol sera alimenté par une batterie et les capteurs utilisés y seront branchés avec les ESC, eux même branchés aux moteurs. Il sera paramétré à l'aide d'un logiciel sur ordinateur (MissionPlanner ou QGroundControl).

On peut trouver des contrôleurs de vol à différents prix avec différentes utilités en fonction des options déjà présentes dessus tel qu'un gyroscope, un baromètre, un GPS (, etc), ou que l'on peut greffer en plus comme un affichage sur écran ou du Bluetooth [8]. Mais nous n'avons pas la nécessité d'avoir de nombreuses options spécifiques. Donc plusieurs choix s'ouvrent à nous.

Voici un premier aperçu de certains contrôleurs de vol très utilisés :

- L'ArduPilotMega, doté d'un processeur de 8 bits, contient un gyroscope, un accéléromètre, un magnétomètre et un baromètre et permet ainsi l'autonomie du drone [9].
- Le PixHawk correspondant à une version améliorée de l'ArduPilotMega avec un processeur en 32 bits et est configurable sous Unix/Linux [10].
- La MultiWii, utilisant la technologie de contrôle des manettes de Wii, contenant des gyroscopes et des accéléromètres, et compatible Linux. Ce contrôleur de vol est souvent utilisé pour des vols acrobatiques [11].
- L'OpenPilot, spécial multirotor acceptant au maximum 6 moteurs [12].
- Le Pxfmini, correspondant à un shield pour Raspberry Pi [13].

Les avantages et inconvénients de ces contrôleurs de vol sont répertoriés dans le tableau suivant [1] :






Contrôleur de vol	Avantages	Inconvénients
<i>ArduPilotMega</i> 	<ul style="list-style-type: none"> - Polyvalent - fonctionnalités diverses - logiciel open source 	<ul style="list-style-type: none"> - prix relativement élevé
<i>PixHawk</i> 	<ul style="list-style-type: none"> - Polyvalent - fonctionnalités diverses - logiciel open source - documentation complète - possibilité de mettre un GPS 	<ul style="list-style-type: none"> - prix élevé
<i>MultiWii</i> 	<ul style="list-style-type: none"> - Bas prix - Compatible à n'importe quel type de drone 	<ul style="list-style-type: none"> - prise en main difficile
<i>OpenPilot</i> 	<ul style="list-style-type: none"> - Bas prix - Open source - Polyvalent 	<ul style="list-style-type: none"> - pas de mode automatique
<i>Pxfmini</i> 	<ul style="list-style-type: none"> - Bas prix - Adapté aux cartes Raspberry Pi préexistantes sur le drone - Compatible sur différent types de drones 	<ul style="list-style-type: none"> - peu de documentation

Tableau 6 : Comparatif des contrôleurs de vol

II.2.2 Contrôleur de vol choisi

Nous avons choisi de prendre un Pixhawk, conseillé par nos encadrants qui nous en ont fourni un, pour tous les avantages qu'il nous procure. Nous allons paramétrer ce contrôleur de vol à l'aide du logiciel Mission Planner. Nous devons également nous munir d'un bus I2C car nous utilisons plusieurs composants qui se branche en I2C sur le Pixhawk mais ce dernier ne comporte qu'une seule prise I2C.

Pour valider dans ce choix nous avons travaillé de manière plus approfondie sur l'utilisation d'un Pixhawk, du logiciel Mission Planner [14], et de la calibration des différents composants nécessaires au vol via ce contrôleur de vol, en particulier à l'aide de l'atelier technique « conception d'un drone ».

II.3. Carte mère

Le projet requiert de faire des calculs depuis le drone en temps réel. C'est pourquoi il est nécessaire d'embarquer du matériel électronique de calcul. De nombreuses possibilités existent, mais les plus adaptées à un drone sont les mini et nano PC, puisque le poids maximum que nous pouvons embarquer est limité. Les facteurs déterminants pour faire le choix sont :

- Le poids ;
- La puissance de calcul ;
- Les systèmes d'exploitation compatibles ;
- Les ports d'interface disponibles.

En ce qui concerne les nano PC, il existe notamment les Raspberry Pi [15], [16], utilisées l'année dernière par nos camarades sur le projet QX. Sinon, il est possible de faire un nano PC sur mesure, ce qui serait certainement la meilleure solution si nous avons les connaissances et l'argent nécessaire. Enfin, les autres cartes électroniques de ce genre ne sont pas suffisantes pour installer un système d'exploitation tel que Linux, nécessaire pour utiliser des outils indispensables comme ROS. La seule solution envisageable parmi les nano PC est donc la Raspberry Pi. Cependant, une seule de ces cartes-là n'est pas suffisante par rapport aux performances dont nous avons besoin. C'est pourquoi les membres du projets QX ont utilisé deux de ces cartes. Une paire de Raspberry Pi est donc la solution retenue parmi les nano PC.

Les mini PC sont nombreux aujourd'hui sur le marché. Ils offrent une puissance comparable aux ordinateurs bas de gamme du commerce, et sont suffisamment peu encombrants et légers pour pouvoir être embarqués. L'état de l'art a été mené avec un ZOTAC ZBOX-PI332-W2B, que Fabrice Lebars nous a prêté pour pouvoir l'étudier, et qui correspond à un mini PC classique.

Le tableau ci-dessous compare donc les principales différences entre l'utilisation de deux Raspberry Pi et l'utilisation d'un ZOTAC [17].

	2 Raspberry Pi	ZOTAC
Fréquence du processeur	2 x 1.2GHz = 2.4GHz	1.44GHz jusqu'à 2.24GHz
Système d'exploitation	Linux	Windows ou Linux
RAM disponible	2 x 1GB = 2GB	✓ 4 GB
Espace de stockage disponible	✓ Mini carte SD (de 256 MB à 64 GB) x 2	32 GB
Alimentation électrique	5V – 2 x 2.5A = 5A maximum	✓ 5V – 3A
Ports d'interface	✓ 2 x : <ul style="list-style-type: none">• 4 ports USB• 40 pins GPIO• Port Ethernet• Wifi 802.11n• Camera interface (CSI)	<ul style="list-style-type: none">• 1 x USB 3.0 Type-C• 2 x USB 3.0• Port Ethernet• Antenne interne• Wifi 802.11ac
Poids	✓ 2 x 45g = 90g	165g
Autres avantages	Permet de séparer physiquement le traitement de la cartographie et celui du vol	Possibilité d'installer une version graphique de Linux

Tableau 7 : Comparatif pour la carte mère

Finalement, il nous a paru que l'utilisation de deux cartes Raspberry Pi offrait plus d'avantages que le mini PC. En effet, nous gagnons en poids, ce qui est un des facteurs les plus importants concernant le matériel embarqué sur un drone, et nous ne perdons pas en puissance (juste en mémoire vive mais il se trouve que 1GB par carte est suffisant). L'interface de la Raspberry Pi est beaucoup plus intéressante et si le traitement de la cartographie provoque une erreur pendant le vol, le drone a peu de chance de se crasher pour autant, puisque les deux traitements sont complètement séparés. Cette solution consomme peut-être en revanche un peu plus d'énergie, mais la différence est de l'ordre de 2A maximum, ce qui est assez faible, et les cartes ne devraient pas consommer les 5A en continu.

II.4 Communications

Nous avons vu dans l'étude des fonctions du drone que celui-ci doit pouvoir envoyer des informations à l'utilisateur. Pour ce faire, il faut établir un protocole de communication entre le robot et le terminal qu'utilisera l'utilisateur. De plus, étant donné que nous avons choisi l'utilisation de deux cartes de calcul dans le robot, ces deux cartes doivent également communiquer entre elles. En effet, lorsque le drone volera de manière autonome, il aura besoin d'avoir accès au plan dessiné par la carte de cartographie pour savoir où il lui est intéressant d'aller ensuite (voir II.6. Exploration automatique), et également de la carte de vol pour réaliser concrètement les déplacements prévus.

Il était donc nécessaire de choisir une communication sans fil, avec suffisamment de portée, et avec une consommation d'énergie raisonnable. Ce qu'il se fait de mieux aujourd'hui dans ce domaine et qui nous était accessible est ce qu'on appelle une « bulle wifi ». C'est-à-dire qu'on branche un routeur wifi auquel vont se connecter les deux cartes du drone ainsi que les terminaux des personnes souhaitant communiquer avec l'appareil. Puis la communication se fait de manière très simple grâce aux adresses IP distribuées par le routeur.

Il existe plusieurs normes wifi, la plus performante aujourd'hui s'appelle 802.11n et offre jusqu'à 250m de portée (dans des conditions parfaites) [18]. Le débit de ces protocoles de communication est largement suffisant pour la quantité d'information que nous souhaitons transmettre. De plus, nous pourrions toujours utiliser du matériel réseau Ubiquiti pour étendre cette portée [19].

II.5. Interface graphique

Pour l'interface graphique, nous avons étudié le choix du langage de programmation en se basant sur les langages que nous avons déjà aperçu auparavant pour des raisons d'efficacité. Voici trois des langages répertoriés avec leurs avantages et inconvénients.

	Avantages	Inconvénients
<i>Python</i>	<ul style="list-style-type: none">- Facile d'utilisation- Langage de haut niveau- maintenu et documenté- rapidité de développement grâce à des bibliothèque très complète- L'équipe a des bases assez solide dans ce langage- langage interprété (=> possibilité de tester directement)	<ul style="list-style-type: none">- Lent d'exécution- packaging compliqué
<i>Java</i>	<ul style="list-style-type: none">- langage de haut niveau- rapidité de développement grâce à des bibliothèque très complète- langage orienté objet (=> code plus facilement réutilisable)- langage interprété (=> possibilité de tester directement)	<ul style="list-style-type: none">- Lent d'exécution- L'équipe a été récemment initié à ce langage
<i>C++</i>	<ul style="list-style-type: none">- rapide d'exécution	<ul style="list-style-type: none">- L'équipe n'as pas de base dans ce langage- langage de bas niveau, nécessitant une gestion de la mémoire.- langage non interprété

Tableau 8 : Choix du langage de programmation pour l'interface graphique

Notre choix s'est porté sur l'outil Python avec sa bibliothèque PyQt (à l'origine Qt en C++) et le logiciel Qt Designer permettant la création très rapide d'un squelette d'interface graphique.

II.6. Exploration automatique

L'exploration autonome d'un environnement inconnu fait appel à plusieurs tâches que l'on peut séparer comme suit :

- La localisation et la génération d'une carte (SLAM = Simultaneous Localization And Mapping)
- Le choix du parcours d'exploration

Les capteurs utilisés sont les RPLIDAR A1/A2 pour la détection des objets, murs ou autres, et des capteurs ultrasons comme détecteurs de proximité pour éviter les collisions.

Concernant l'exploration, nous optons pour l'algorithme d'exploration mise au point par l'université de Koblenz-Landau [20].

On part de notre point de départ qui sera retenu dans la carte au fur et à mesure de ses mises à jour. Ainsi, une fois la mission terminée, le drone pourra y revenir. Pour se repérer, on découpe la carte en cases auxquelles on associe un état parmi les états suivants : occupé, libre ou inconnu. Cet état est déterminé par le lidar : soit on voit qu'il n'y a pas d'obstacle (blanc), soit on voit un obstacle (noir), soit on

ne voit pas l'endroit (gris). De cette façon, on obtient les frontières de notre carte : c'est l'ensemble des cases libres adjacentes à une case inconnue.

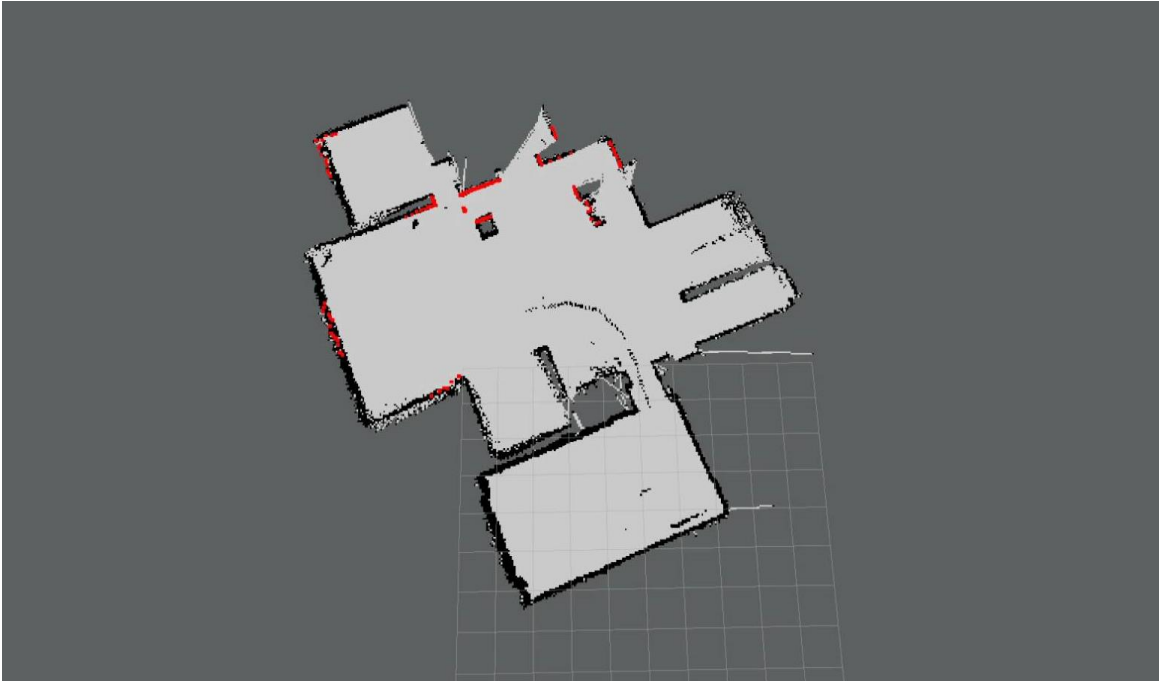


Figure 8 : Rendu visuel LIDAR + SLAM

Il faut maintenant choisir une prochaine destination à atteindre parmi ces frontières. Plusieurs possibilités sont envisageables selon la façon dont on veut explorer la zone, par exemple si on veut être plus rapide, ou plus précis.

Pour faire un survol rapide, on peut choisir le prochain point qui offrirait une mise à jour de la carte la plus grande, c'est-à-dire qui donnerait accès à une zone inconnue la plus large possible.

L'autre cas plus simple, est de choisir comme destination le prochain point frontière le plus proche. L'inconvénient de cette version est que si pour une raison quelconque le drone doit arrêter son exploration avant d'avoir fini, il est possible d'obtenir une carte dont on ne peut rien extrapoler.

Prenons l'exemple d'une pièce carrée, où l'on fait démarrer le drone dans un coin. Après une étape, le premier choix nous ferait aller vers le centre de la pièce d'où on verrait les quatre murs mais peut-être pas les coins. Le second choix longerait un des deux murs. Ainsi, après une étape, on sait déjà dans le premier cas qu'on a affaire à une pièce, alors que dans le second cas on pourrait très bien croire longer un mur extérieur. Evidemment, si on suppose qu'on a beaucoup de temps devant nous pour la cartographie, les deux méthodes donneront le même résultat, à savoir une carte complète des lieux.

Enfin, il faut ajouter à cela une sécurité. Choisir le chemin le plus court amène certainement à longer des murs, passer trop près d'obstacles en tout genre, ce que l'on veut absolument éviter. Pour remédier à cela, on ajoute une contrainte de distance aux obstacles. A chaque case de notre carte est attribuée sa distance à la case obstacle la plus proche. On peut ensuite rendre inaccessible certaines de

ces cases, par exemple toutes celles dont la valeur est inférieure à 2. De cette manière, on épaissit les murs virtuellement et le drone restera à distance des murs réels.

Nous avons programmé une simulation de l'exploration du robot, voici un exemple de résultat :

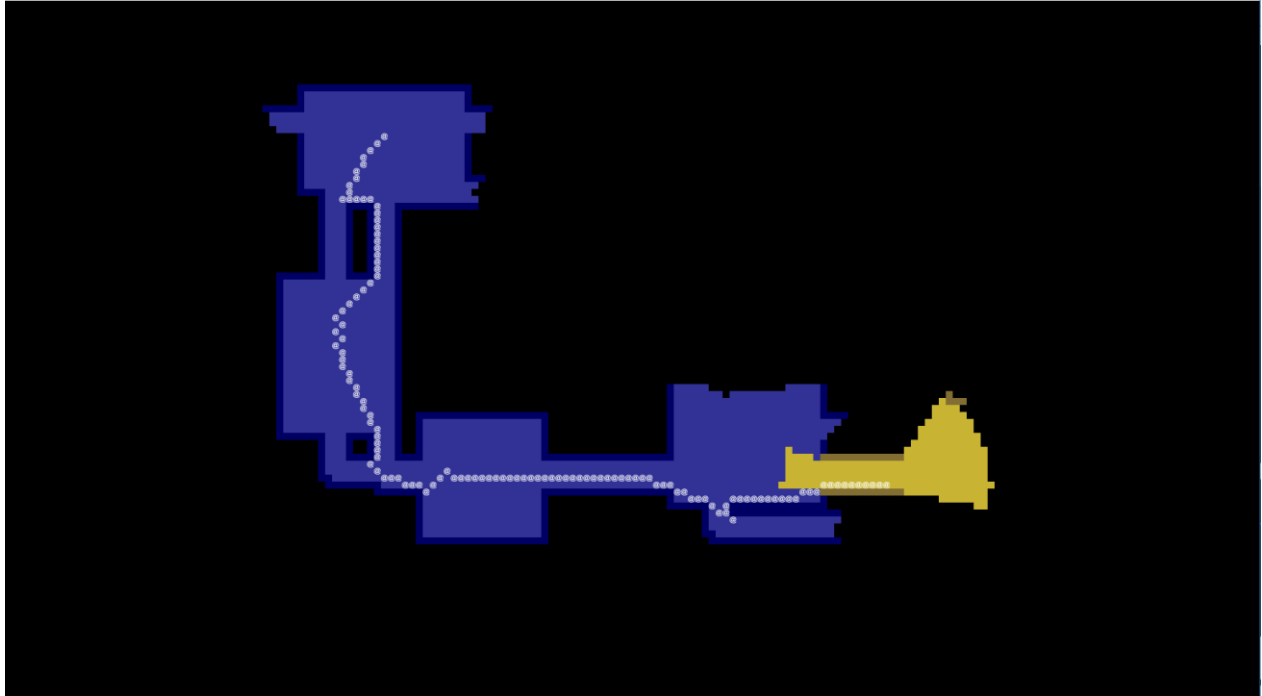


Figure 9: Simulation en cours

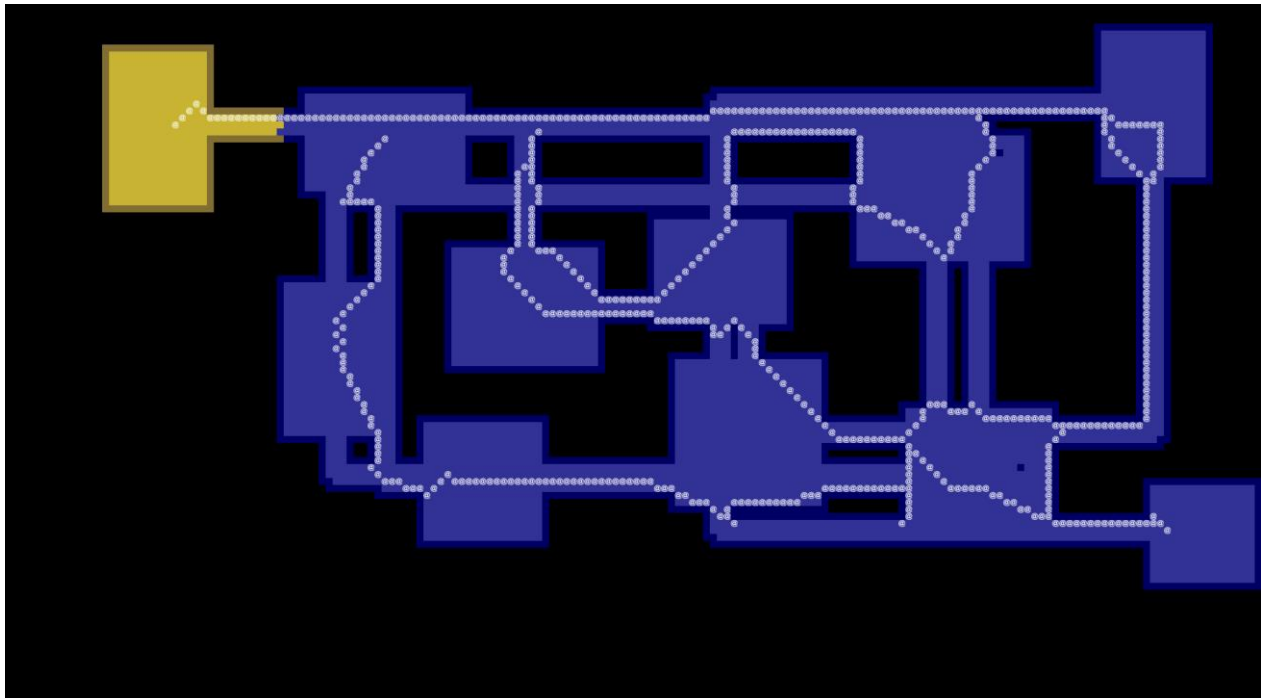


Figure 10: Simulation terminée

II.7. Algorithmes de SLAM

Le SLAM est un problème où un robot doit à la fois fournir une carte d'un lieu inconnu et se localiser dans cette même carte. Les différents algorithmes proposant une solution au problème fournissent des approximations de la position du robot et des obstacles et utilisent le plus couramment des variations du filtre de Kalman.

Chaque algorithme est dépendant du type d'environnement exploré, car on peut souvent les optimiser en ajoutant des hypothèses sur le type de terrain à explorer, par exemple si on sait déjà que le terrain est étroit (couloir, pièce encombrée, ...), dégagé (plaines, ...), dense (forêt, ...).

De plus, les types de capteurs sont aussi à prendre en compte. On peut utiliser des capteurs optiques comme les lidars (1D, 2D ou 3D), les caméras (classiques ou 3D), ou des sonars (pour un drone sous-marin par exemple). Globalement, n'importe quel capteur duquel on peut déduire une distance à un point fait l'affaire, puis les hypothèses du terrain et les prérequis de l'utilisateur nous font choisir le capteur le plus adapté selon la distance de vue et la précision que l'on estime convenable par rapport au problème.

Pour finir, la différenciation entre objets statiques et dynamiques dans l'environnement présente des challenges encore non totalement résolu pour l'instant. Dans notre cadre de cartographie d'un lieu, on espère que rien ne bouge durant l'acquisition car les résultats seraient faussés. Supposons qu'un objet tourne en rond quelque part, il apparaîtrait sur la carte comme un cercle. Or un cercle peut très bien être interprété comme un pilier du bâtiment par exemple. pour l'instant.

Nous utilisons les algorithmes déjà présents sur le robot car ils sont fonctionnels. Il s'agit des « nodes ROS » propres à l'utilisation du RPLIDAR [21]–[23]. L'avantage du RPLIDAR par rapport à d'autres lidars est qu'il offre de base une vision de son environnement à 360° par rapport à sa position. On n'a donc pas besoin de s'occuper de l'asservissement des servomoteurs ni de l'ajustement des mesures prises comme cela aurait été le cas si on faisait notre propre tourelle avec un lidar fixé dessus. De plus, l'installation ROS pour son utilisation est assez directe [24].

III. Architecture physique

L'architecture physique de notre projet est assez complexe, puisque pour répondre à tous les besoins et contraintes du projet, beaucoup de matériel est nécessaire. Le diagramme ci-dessous présente donc cette architecture composée des différents composants du drone, des milieux extérieurs avec lesquels il échange, et des différents types de flux qui existent entre toutes ces entités.

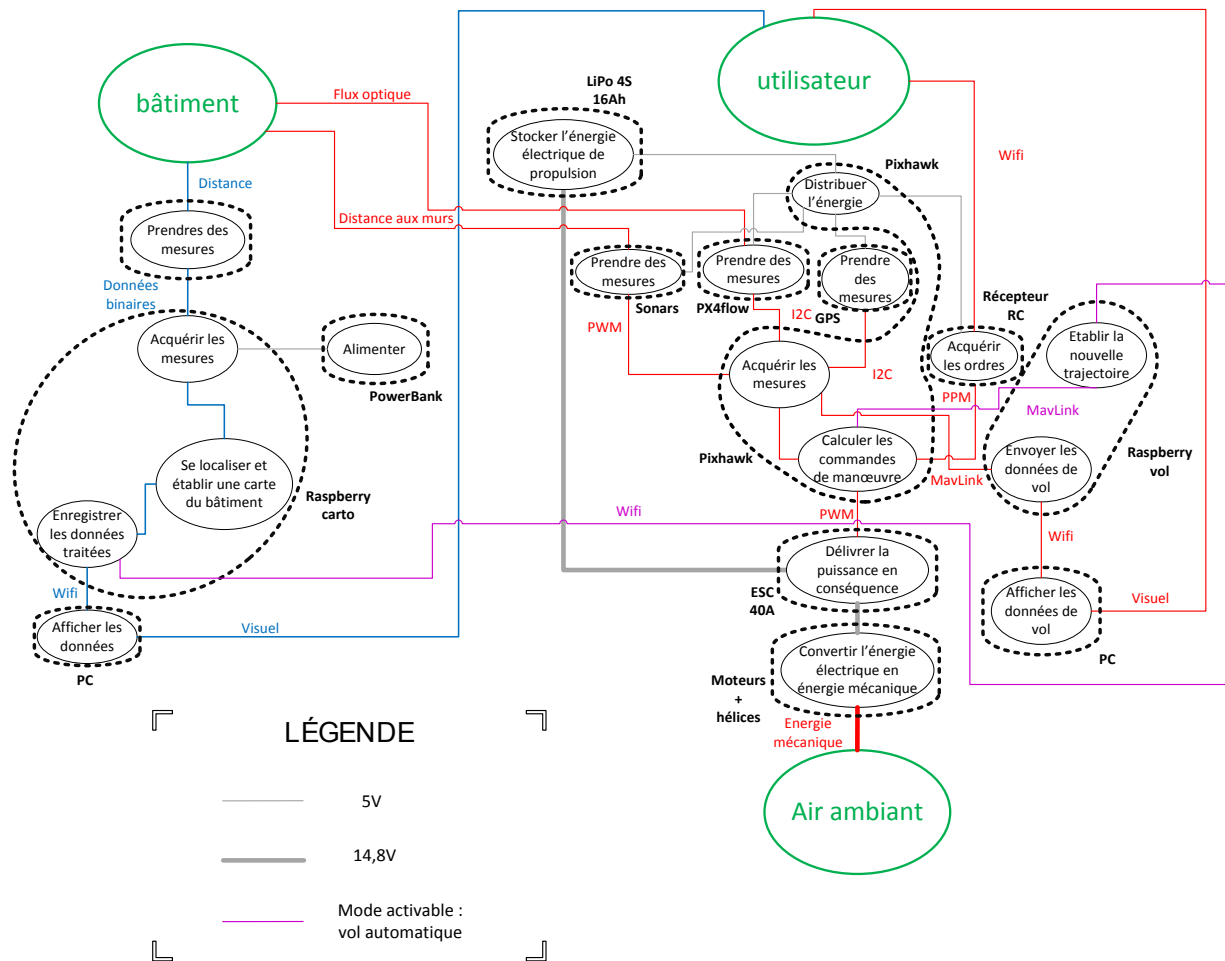


Diagramme 6 : Description de l'architecture physique

→ A gauche du diagramme :

Sur le drone il existait déjà un LIDAR (LIght Detection And Ranging) permettant de prendre les mesures de distance entre le drone et les murs. Ce LIDAR renvoie des données binaires à une carte mère déjà présente et réservée pour le travail de cartographie. Cette carte mère traite les mesures obtenues pour en faire une cartographie 2D et localise le drone dans le bâtiment grâce à un algorithme de SLAM (Simultaneous Localization And Mapping). Elle enregistre également les données obtenues qui sont ensuite envoyées à un PC pour que l'utilisateur puisse visualiser la cartographie.

→ A droite du diagramme :

Pour voler à l'intérieur d'un bâtiment en toute sécurité, il faut prendre différentes mesures et les acquérir tel que des mesures de hauteur, de distance aux murs, de cap, ... Dans un même temps il faut acquérir les ordres de l'utilisateur. Une fois ces données obtenues, deux choses restent à faire :

- Les commandes de manœuvre doivent être calculées, puis la puissance à délivrer doit être envoyée aux blocs propulsion qui doivent être capables de porter le poids du drone.
- Les données doivent être fournies à une interface pour être affichées.

Donc le drone doit être équipé d'un certain nombre de capteurs pour pouvoir éviter les obstacles et analyser ses mouvements. Sur ce diagramme nous avons placé les capteurs à ultrasons (déjà présent sur le drone pour obtenir les distances du drone par rapport aux obstacles), un capteur à flux optique de type px4-flow (déjà choisi par nos prédécesseur) et un GPS pour utiliser sa boussole. Le contrôleur de vol, ici un Pixhawk, récupère les informations des capteurs et des commandes de l'utilisateur. Puis il envoie les données à un PC pour que l'utilisateur puisse les visualiser, et envoie une commande aux ESC (Electronic Speed Control) qui permettent ensuite – à l'aide d'une batterie, d'un power module et d'un PDB (Power Distribution Board) - de délivrer la puissance nécessaire au bloc propulsion.

Nous en avons résumé les solutions techniques dans le tableau page suivante.

Fonction principale/ de service		Identifiant exigences	Type/ Nature	Détail exigence				Solution
			Sous exigence 1	Sous exigence 2	Sous exigence 3	Sous exigence 4		
Cartographier un bâtiment	CB-F201	Fonctionnalité	Produire une carte lisible d'un bâtiment inconnu	Stocker la carte en mémoire			Carte SD de 8 GB au sein de la Raspberry Pi	
	CB-F402	Fonctionnalité		Produire une carte précise			LIDAR	
	CB-U401	Utilisabilité	Se localiser dans une pièce inconnue				SLAM dans la Raspberry Pi	
Pouvoir voler	PV-F401	Fonctionnalité	Se déplacer en sécurité	Etre Télé-opérable	Pilotage semi-automatique	Via télécommande	Emetteur récepteur (télécommande pour drone)	
	PV-F102	Fonctionnalité				Via pc	Emetteur récepteur (manette)	
	PV-U301	Utilisabilité			Pilotage automatique		Programme implémenté dans la Raspberry Pi	
	PV-U302	Utilisabilité				Se localiser dans une pièce inconnue	SLAM dans la Raspberry Pi	
				Eviter les obstacles	Repérer les obstacles		Capteurs à ultrasons et SLAM	
	PV-F403	Fonctionnalité	Pouvoir porter l'équipement				Moteurs, batteries et hélices dimensionnés pour le transport de la charge du drone	
	PV-F404	Fonctionnalité	Atterrir				Emetteur/ récepteur via télécommande	
	PV-F405	Fonctionnalité	Effectuer un vol stationnaire				Capteur à flux optique	

Fonction principale/ de service			Identifiant exigences	Type/ Nature	Détail exigence				Solution
					Sous exigence 1	Sous exigence 2	Sous exigence 3	Sous exigence 4	
Visualiser les données	VD-E301	Ergonomie	Acquérir les données	En temps réel				Communication Wifi avec pc	
	VD-F201	Fonctionnalité		En temps décalé				Dépouillement de la carte SD	
	VD-E202	Ergonomie	Avoir une IHM	Avoir une interface claire				Interface graphique à implémenter	
	VD-E103	Ergonomie		Présenter l'état du drone	Afficher l'état de la batterie	Prendre en compte le PNR	Interface graphique à implémenter		
	VD-E104	Ergonomie		Afficher qualité du signal	Wifi		Interface graphique à implémenter		
	VD-E105	Ergonomie			Télécommande		Interface graphique à implémenter		
	VD-E106	Ergonomie		Afficher le mode courant	Mode de vol		Interface graphique à implémenter		
	VD-E107	Ergonomie			Etat de la fonction cartographique		Interface graphique à implémenter		
	VD-E108	Ergonomie		Afficher les données des capteurs	Altitude/ Assiette etc		PixHawk		
	VD-E109	Ergonomie			Vitesse		PixHawk		
	VD-E110	Ergonomie		Afficher les événements	Afficher les pannes		Interface graphique à implémenter		
	VD-E111	Ergonomie			Afficher les manœuvres d'évitement		Interface graphique à implémenter		
	VD-F302	Fonctionnalité		Présenter la cartographie	Afficher les murs		A faire en télé opérable		
	VD-F303	Fonctionnalité			Afficher la position du drone		A faire en télé opérable		
Fonction de contraintes	FC-E201	Ergonomie	Etre robuste						Ajouter un carénage
	FC-E402	Ergonomie	Etre stable						Capteur à flux optique
	FC-E303	Ergonomie	Etre facilement pilotable	Avoir un mode semi-automatique				PixHawk	
	FC-F301	Fonctionnalité	Etre capable de passer des portes standards	Ne pas heurter				Taille 80cm pour 2m	
	FC-E304	Ergonomie		Etre stable					
	FC-F402	Fonctionnalité	Avoir une procédure de sécurité						
	FC-F303	Fonctionnalité	Séparer les alimentations	Avoir deux batteries	Moteur		Batterie à grand pouvoir de décharge. Choix d'une batterie LIPO 4S		
	FC-F304	Fonctionnalité			Système embarqué		Batterie de portable		
	FC-C401	Contrainte	Etre en accord avec la législation						
	FC-C402	Contrainte	Etre en accord avec le règlement d'Eurathlon 2017						

Tableau 9 : Détail des solutions envisagées

IV. La conduite du projet

IV.1. Identification du travail à effectuer

Après avoir étudié une grande partie du système préexistant (cf. Annexes : drone QX récupéré), nous avons pu identifier les travaux nécessaires au bon fonctionnement de notre système en vérifiant si les choix effectués pendant l'ancien projet nous satisfaisaient. Les fonctions à réaliser depuis le début sont des bulles **rouges** sur le diagramme ci-dessous, **orange** pour les travaux à modifier, et **vert** pour ce qui est repris sur le drone préexistant.

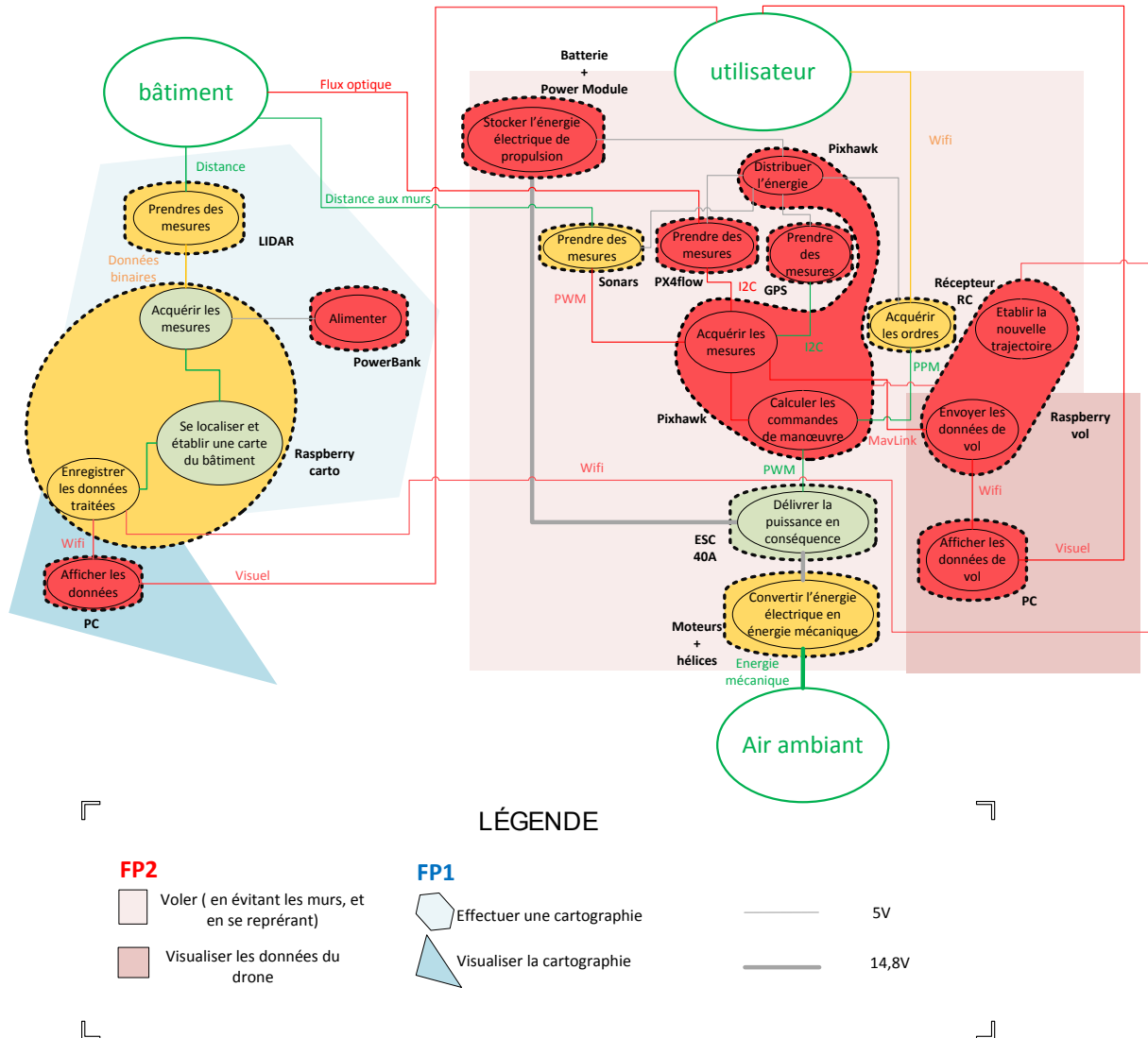


Diagramme 7 : Visualisation des travaux à effectuer sur l'architecture physique

Nous avons ensuite détaillé les différentes tâches à réaliser pour chacune de ses fonctions, comme ci-dessous. Naturellement, les différents choix à faire après les états des arts engendreront d'autres travaux à réaliser. Mais ce travail nous permet tout de même une bonne estimation pour une meilleure répartition des tâches. Ici ne sont traitées que les fonctions principales.

Le détail des tâches pour « le drone doit pouvoir voler » est le suivant :

Prendre des mesures (pour éviter les murs)	
<i>Composant</i>	Capteurs à ultrasons
<i>Identifiants concernés</i>	<i>PV-U012</i>
Travaux préexistants	- Existence de 4 capteurs à ultrasons sur le drone, mais non branchés
Travaux prévus	- Branchement à la carte mère vol, - Intégrer à un software d'évitement d'obstacles, - Lui-même à intégrer dans la logique de vol.

Convertir l'énergie électrique en énergie mécanique	
<i>Composant</i>	Moteurs – hélices
<i>Identifiants concernés</i>	<i>PV-F003</i>
Travaux préexistants	- Bloc propulsion préexistant mais insatisfaisant.
Travaux prévus	- Choix d'un autre bloc de propulsion (cf Etat de l'art et choix), - Achat des composants non présents, - Installation du nouveau bloc.

Etablir une nouvelle trajectoire (vol automatique)	
<i>Composant</i>	Raspberry Pi
<i>Identifiants concernés</i>	<i>PV-U101, PV-U102</i>
Travaux préexistants	- Inexploitable
Travaux prévus	- Mise en place d'un algorithme de génération de trajectoire - Mise en place d'une communication entre les cartes Raspberry Pi.

Autres fonctions dans « voler »	
<i>Composant</i>	Contrôleur de vol, batterie, ESC, GPS, capteur à flux optique, PDB, Power Module, récepteur, émetteur, convertisseur...
<i>Identifiants concernés</i>	<i>PV</i>
Travaux préexistants	- Présence d'un contrôleur de vol mais insatisfaisant, - Présence d'ESC satisfaisants, - Choix d'un capteur à flux optique, - Présence d'un PDB satisfaisant, - Présence d'un récepteur et d'un convertisseur adapté.

Travaux prévus	<ul style="list-style-type: none"> - Choix d'un contrôleur de vol (cf. Etat de l'art et choix) et intégration, - Choix d'une batterie avec le bloc propulsion (cf. Etat de l'art et choix) et intégration, - Intégration du capteur à flux optique, - Choix du Power Module et intégration, - Etude sur le récepteur-émetteur-convertisseur, - Création de la logique de vol et du software correspondant.
----------------	--

Tableau 10 : Les travaux à effectuer pour « Pouvoir Voler » (PV)

Celui pour « le drone doit établir une cartographie » est le suivant :

Effectuer la cartographie	
Composant	LIDAR, carte mère cartographie
Identifiants concernés	CB
Travaux préexistants	<ul style="list-style-type: none"> - Carte mère présente, branchée et contenant des softwares permettant : <ul style="list-style-type: none"> ▪ Le traitement des mesures du LIDAR, ▪ Se localiser dans le bâtiment. - LIDAR présent, branché et intégré dans le software.
Travaux prévus	<ul style="list-style-type: none"> - Se pencher sur l'enregistrement de données, - Vérifier le choix de la carte mère (cf. Etat de l'art et choix), - Intégrer une batterie d'alimentations.

Tableau 11 : Les travaux à effectuer pour "Etablir une cartographie" (CB)

Le détail des travaux pour l'affichage des données pour le vol et la cartographie se trouve ci-dessous :

Afficher les données (vol et cartographie)	
Composant	PC, Raspberry Pi
Identifiants concernés	VD
Travaux préexistants	<ul style="list-style-type: none"> - Inexploitable.
Travaux prévus	<ul style="list-style-type: none"> - Création d'un réseau de communication Wi-fi entre PC et Pi, - Intégration des cartes mères dans la communication, - Création d'une interface graphique.

Tableau 12 : Les travaux à effectuer pour "Visualiser les données" (VD)

IV.2. Fonctionnement du groupe

Pour le fonctionnement du groupe, nous avons tout d'abord évolué ensemble afin de déterminer les grandes lignes de travail. Une fois sûrs que chacun savait ce qu'il avait à faire, il ne nous restait plus qu'à former des sous-groupes pour travailler efficacement. Les sous-groupes étaient d'abord fixes à priori puis en fonction des centres d'intérêts ils pouvaient être modifiés. Le travail en sous-groupes a été très efficace pour au moins deux raisons. Premièrement, sur un projet d'une telle étendue avoir une répartition du travail est nécessaire à sa réussite. En effet personne ne peut rentrer dans les détails techniques de toutes les structures du projet. Deuxièmement, le travail en sous-groupes n'a pas été un travail cloisonné humainement. Nous travaillions au club robotique où l'espace de travail est propice à la division du travail. Chaque sous-groupe travaillant aux mêmes horaires et dans la même pièce, il était alors plus simple de communiquer et de se poser des questions entre nous.

A) Identification des tâches

Afin d'effectuer une identification des tâches la plus exhaustive possible nous avons d'abord fait le travail global ensemble avant de nous diviser en sous-groupes. Cela permettait de n'oublier quasiment aucune tâche et surtout de n'oublier aucune tâche principale ou cruciale. Nous avons donc produit une liste de tâches à accomplir relativement exhaustive et qui est pour nous un appui solide pour le bon déroulement de la suite du projet.

B) Répartition du travail entre les membres

La répartition du travail dans le groupe s'est faite de manière assez naturelle. En effet nous étions tous intéressés par le projet et les affinités personnelles étaient plutôt bonnes. Nous avons donc créé les sous-groupes par intérêt et curiosité sur certains points techniques que nous voulions approfondir. Cela a permis d'avoir des sous-groupes à la fois efficaces et extrêmement dynamiques. La répartition du travail lors des ateliers techniques a également été la preuve de notre efficacité et de notre dévouement au projet.

C) Investissement dans le travail

L'investissement dans le travail a été relativement facile à suivre puisque nous travaillions toujours tous ensemble sur les créneaux dédiés au projet. Ces deux matinées par semaine nous ont permis de travailler tous ensemble au club robotique. Lors de ces séances, nous arrivions en énonçant le travail fait à la maison. Un point a été amélioré, c'est le débriefing à l'issue de chaque séance. Celui-ci permet d'obtenir un point de situation sur chaque état d'avancement des différents sous-groupes. En effet nous ne le faisons pas au début et il nous manquait parfois de la coordination. Cependant, l'assiduité de tous les membres et la proximité dans le travail nous ont permis de ne pas rencontrer de difficulté majeure. La correction a donc été préventive et non curative.

D) Collaboration/Interaction

Comme cela a été développé dans le paragraphe précédent, la collaboration au sein des groupes a été bonne puisque nous avons formé des sous-groupes à l'unanimité. De plus nous travaillons tous ensemble dans un espace de grande proximité, ceci facilite les échanges entre les sous-groupes. Enfin, des comptes rendus de chaque séance nous permettaient de bien suivre l'évolution globale du projet.

E) Respect des délais

Au niveau du respect des délais, nous sommes dans les temps même s'il faut reconnaître que les échéances principales sont à venir et non déjà passées. Il nous faudra être vigilant sur le travail à accomplir mais nous pouvons déjà être fiers de ce qui a été fait.

F) Régularité du « reporting » avec les encadrants

Les réunions avec le consultant IS ont été d'abord nombreuses et régulières afin d'avoir un travail efficace. En effet le risque à travailler sans encadrement régulier était de produire un travail qui ne répondait pas aux attentes. Une fois le travail bien défini nous avons pu prendre un peu plus de recul et espacer un peu plus dans le temps nos réunions.

Les réunions avec les encadrants du projet se faisaient sous deux formes. Soit des rendez-vous afin de poser des questions précises. Par exemple lors de la phase de commande de pièces, ou d'interview sur un sujet particulier. Soit des réunions moins formelles pour discuter ensemble de l'avancement du projet et pour mettre en évidence les prochaines difficultés approchant.

G) Force de proposition du groupe

Notre force de proposition en tant que groupe s'est déclinée de deux manières :

- d'une part nous avons fait preuve de proactivité dans la recherche de solutions notamment au niveau de l'IS.

- d'autre part, dans la réalisation de la commande de pièces nouvelles, nous avons, grâce à l'expérience de certains membres du club robotique, pu faire des choix judicieux et correspondant à un budget limité.

H) Degré d'agilité du groupe

Pour le degré d'agilité du groupe, bien que nous ayons ancré les valeurs fondamentales de cette méthode que sont l'équipe, l'application, la collaboration et l'acceptation du changement, nous n'avons pas pu, au cours de cette partie propre à l'analyse fonctionnelle, réellement l'appliquer au groupe. En effet cette méthode bien qu'applicable à tout travail semble plus à même d'être utilisée pour des travaux techniques. C'est pourquoi nous essayerons de la mettre en pratique au second semestre dès que des versions opérationnelles de notre drone pourront être livrées.

Ce projet qui est de grande envergure, est également notre premier projet industriel. Nous avons bien entendu rencontré quelques difficultés au niveau humain qui ont été surpassées par une communication honnête dans l'équipe, des discussions sans tabous et surtout grâce à la réunion de mi-parcours où l'équipe s'autoévaluait. Lors de cette réunion, nous avons identifiés plusieurs difficultés mineures au sein de l'équipe. La principale était le manque de connaissance de la part du groupe entier sur l'état d'avancement de chaque partie. Pour être plus précis, chaque groupe travaillait dans son coin, demandait des informations aux autres groupes en cas de besoin, mais personne n'allait de lui-même expliquer aux autres groupes où il en était. Afin d'éviter tous soucis à l'avenir, et surtout d'éviter un manque de lisibilité à long terme, nous avons décidé de faire un point récapitulatif à la fin de chaque séance sur l'état d'avancement des travaux de chacun. Cette difficulté déjà surmontée à l'heure où nous écrivons ce rapport a été bien prévenue grâce à la réunion de mi-parcours. Elle aurait pu avoir des conséquences peut-être plus graves si le problème n'avait pas été détecté si tôt.

Conclusion

A mi-parcours, nous avons posé les bases nécessaires au bon déroulement de notre projet. En effet, tout ce travail en amont permet de gagner du temps en aval et de rendre notre travail plus efficace. Nous avons déjà appris beaucoup sur le travail en équipe qui n'est pas si facile à mettre en place sur un groupe d'une taille encore jamais expérimentée pour nous. Les retours d'expérience sont eux aussi déjà nombreux et nous n'attendons plus qu'avec hâte l'arrivée des pièces pour vérifier la véracité de notre dimensionnement. Grâce à une analyse fonctionnelle poussée, nous avons maintenant un recul important sur les objectifs finaux du projet et nous pourrions ainsi commencer sereinement la partie plus technique et opérationnelle du projet qui est ce qui nous attirait avant tout dans ce projet. Nous sommes donc plus motivés que jamais à relever le défi et le challenge que représente le concours Eurathlon car la fin, comme les moyens à mettre en place, nous seront très instructifs.

Bibliographie

- [1] H. BOENNINH, C. BADIER, P. MARTIN, G. NEAU, S. DO MONTE, et G. LE ROCH, « Projet QX Rapport. » 2016.
- [2] « The European Robotics League | SPARC Robotics. » [En ligne]. Disponible sur: <http://sparc-robotics.eu/the-european-robotics-league/>.
- [3] M. Gortolev, « 2016 Quadcopter vs Hexacopter vs Octocopter: Pros & Cons [EXPOSED] », *Dronebly*, 24-nov-2014. .
- [4] « How to choose the right propeller/motor combination for a quadcopter? », *Robotics Stack Exchange*. [En ligne]. Disponible sur: <http://robotics.stackexchange.com/questions/25/how-to-choose-the-right-propeller-motor-combination-for-a-quadcopter>.
- [5] @unmannedtech Alex et @Sam Sam, « How to choose the right motor for your multicopter drone », *DroneTrest*. [En ligne]. Disponible sur: <http://www.dronetrest.com/t/how-to-choose-the-right-motor-for-your-multicopter-drone/568>.
- [6] Oscar, « How to choose Motor and Propeller for Quadcopter », *Oscar Liang*, 10-oct-2013. .
- [7] M. MUELLER, « eCalc - xcopterCalc. » [En ligne]. Disponible sur: <http://www.ecalc.ch/xcopter.calc.php>.
- [8] Ardupilot, « Introducing Copter — Copter documentation », *ardupilot.org*. [En ligne]. Disponible sur: <http://ardupilot.org/copter/docs/introduction.html>.
- [9] Pascal, « Contrôleur de vol APM 2.6 pour multicoptère », *Drone About*. .
- [10] « Pixhawk Overview — Copter documentation. » [En ligne]. Disponible sur: <http://ardupilot.org/copter/docs/common-pixhawk-overview.html>.
- [11] « MultiWii », *MultiWii*. [En ligne]. Disponible sur: <http://www.multiwii.com/>.
- [12] « Contrôleur de vol OpenPilot CC3D ATOM avec boîtier », *La DroneShop*. [En ligne]. Disponible sur: <http://ladroneshop.com/fr/multiwii-drone-racing/1547--controleur-de-vol-openpilot-cc3d-atom-avec-boitier.html>.
- [13] « PXFmini | Erle Robotics. » .
- [14] « Mission Planner Overview — Mission Planner documentation. » [En ligne]. Disponible sur: <http://ardupilot.org/planner/docs/mission-planner-overview.html>.
- [15] « Framboise 314, le Raspberry Pi à la sauce française. », *Framboise 314*. .
- [16] « Raspberry Pi - Teach, Learn, and Make with Raspberry Pi », *Raspberry Pi*. [En ligne]. Disponible sur: <https://www.raspberrypi.org/>.
- [17] « Mini PC | ZOTAC », *zotac.com*. [En ligne]. Disponible sur: https://www.zotac.com/tn/product/mini_pcs/all.
- [18] « WiFi - Portée et débit », *CommentCaMarche*. [En ligne]. Disponible sur: <http://www.commentcamarche.net/contents/1280-wifi-portee-et-debit>.
- [19] « Ubiquiti Networks - Wireless networking products for broadband and enterprise. » [En ligne]. Disponible sur: <https://www.ubnt.com/>.
- [20] S. Wirth et J. Pellenz, « Exploration Transform: A stable exploring algorithm for robots in rescue environments. »
- [21] « robopeak/rplidar_ros », *GitHub*. [En ligne]. Disponible sur: https://github.com/robopeak/rplidar_ros.
- [22] « rplidar - ROS Wiki. » [En ligne]. Disponible sur: <http://wiki.ros.org/rplidar>.
- [23] « hector_mapping - ROS Wiki. » [En ligne]. Disponible sur: http://wiki.ros.org/hector_mapping.
- [24] M. Quigley, B. Gerkey, et W. D. Smart, *Programming Robots with ROS: A Practical Introduction to the Robot Operating System*. O'Reilly Media, Inc., 2015.

Annexe

Logique des identifiants :

Exemple : VD-F203

Les deux premières lettres correspondent à l'identification de la fonction principale ou exigences que l'élément détaille. Ici VD signifie Visualisation de Donnée.

La lettre après le tiret correspond au type d'exigence : F = fonctionnalité, E = Ergonomie, U = Utilisabilité. Ici VD-F203 correspond à « Présenter la cartographie », il s'agit donc d'une fonctionnalité.

Le chiffre suivant la dernière lettre correspond à l'importance. L'importance croît avec la valeur.

Les deux derniers chiffres correspondent au numéro qui rend unique cet identifiant. Ici il s'agit donc de la 3^{ème} exigence de type fonctionnalité d'importance 2 détaillant l'exigence de visualisation de Donnée.

Drone QX récupéré :

