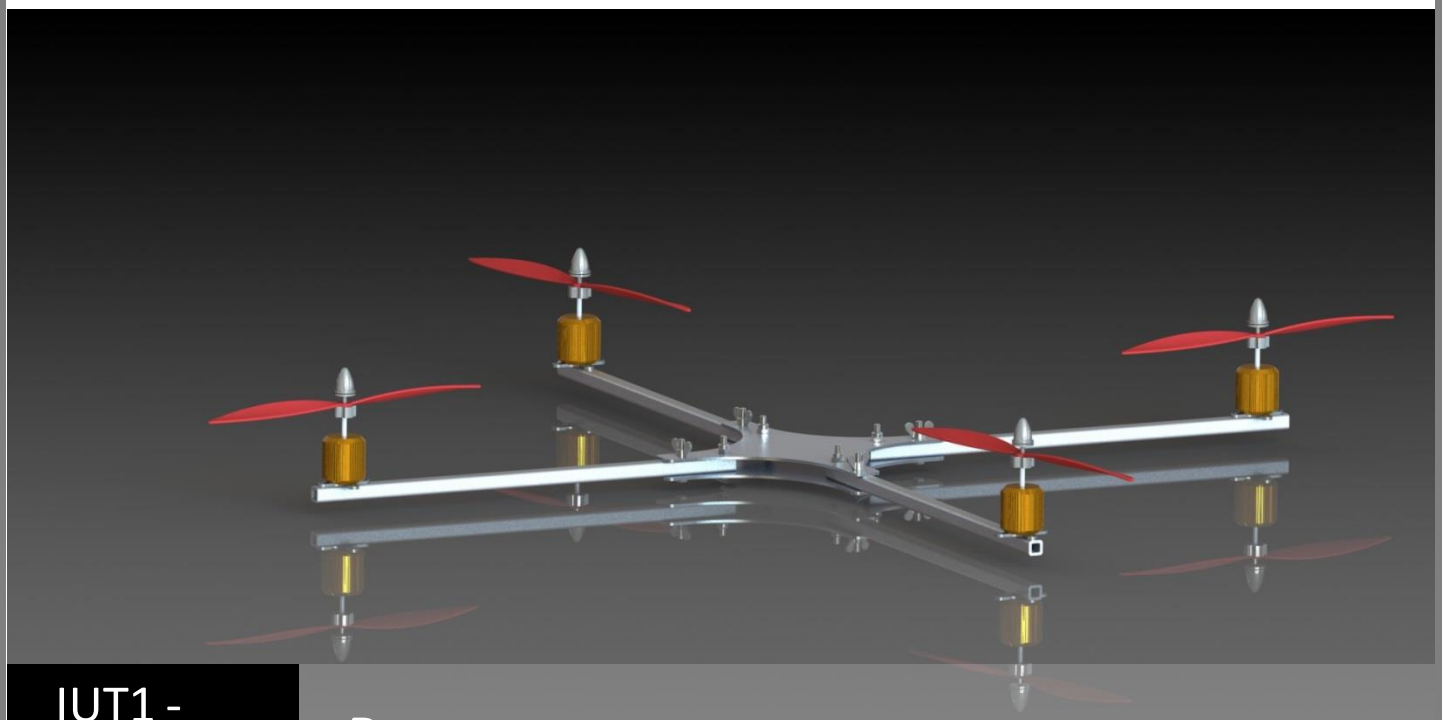
2013 - 2014



IUT1 - GRENOBLE

PROJETS TUTEURES

Réalisation d'un drone | Harold Ducept

Introduction

Description générale

De nos jours, la présence de véhicules volant sans pilotes, (plus connus sous le nom de drones) se fait de plus en plus grande. Une des raisons à ce phénomène est le fait que certains fabricants tendent à rendre facile d'accès ce matériel qui, il y a quelques années encore coutait extrêmement cher. De plus, avec les technologies actuelles, le pilotage est rendu des plus aisés. Ainsi, du matériel qui était destiné à des professionnels aguerris est mis entre les mains de jeunes, voir d'enfants. Une autre des raisons qui peut pousser les gens à vouloir se procurer un drone est le fait de voir fleurir sur le web un grand nombre de vidéos prises depuis ces systèmes. En effet, celles-ci offrent des points de vue incomparable et offrent également une impression de liberté, et donnent le sentiment de pouvoir voler.

Cependant, tous les possesseurs de tels engins n'en font pas la même utilisation. Cela va du domaine du loisir avec la simple envie de piloter jusqu'au domaine militaire. Il faut donc que ces appareils soient adaptés à l'utilisateur final, ce qui explique la très grande variété de produits proposés et les différences de prix associées.

Environnement

La première des contraintes liées à l'environnement est la gravité. En effet, il me faudra une configuration capable de soulever le poids de l'engin et de l'élever un la hauteur désirée.

Le drone que je souhaite réaliser doit être capable de voler en extérieur. Il n'est pas particulièrement destiné aux vols d'intérieurs qui seraient certainement trop dangereux que ce soit pour les objets avoisinants que pour les personnes présentes ou bien pour le drone lui-même.

Pour pouvoir voler en extérieur, le drone devra être capable de réagir à son environnement. En effet, il faudra qu'il puisse se repérer, en particulier connaître son altitude et savoir s'il s'approche d'un obstacle. Tous ces paramètres requièrent le fait de prendre en compte que le système évoluera dans un milieu ouvert et non fermé, dans lequel nous aurions pu nous appuyer sur les murs et le sol pour nous repérer. Il faudra également prendre en compte un élément naturel : le vent. En effet, bien qu'il ne sera pas conçu pour voler par mauvais temps, une simple petite brise pourrait suffire à déstabiliser le système.

Un autre point important à prendre en compte concerne les obstacles qu'il pourrait y avoir concernant la transmission entre la partie commande et le drone lui-même. En effet, les transmissions pourront être perturbées par toute sorte de parasites. Ainsi, une communication ne sera pas aussi bonne en ville (présence de beaucoup de perturbations et de bâtiments) qu'en campagne au milieu d'un champ.

Contraintes

Les contraintes sur ce projet, outre celles liées à l'environnement, sont de 2 ordres : Celles liés au système lui-même, c'est-à-dire les limites que je souhaite pouvoir atteindre avec ; et celles liées au déroulement du projet.

Voici la liste des limites que je souhaite pouvoir atteindre :

- Autonomie : 15min minimum. En effet, un drone est extrêmement couteux en ressources. Les moteurs nécessaires à la propulsion demandent une puissance relativement grande alors que la capacité des batteries reste relativement limitée. Il est bien entendu possible d'augmenter cette capacité, mais pour cela, il faut une batterie plus importante, donc plus lourde, donc des moteurs plus puissants qui consomment d'avantages. On arrive donc rapidement dans des prix qui peuvent monter très haut.
- Portée du signal : 1 km grand minimum (1.5km serait correct). En effet, il est important que le pilote garde toujours un lien avec l'appareil. Il faut donc prévoir un système de transmission qui sera suffisamment puissant. Il faut donc également prévoir un système pour que le pilote puisse continuer à observer comment évolue le drone même lorsque celui-ci sera trop loin pour être à portée de vue.

Ces 2 contraintes, en plus d'être liées, en apportent d'autres. En effet, l'autonomie sera fonction du poids de l'appareil. C'est pourquoi j'ai choisis de ne pas dépasser les 2kg. La structure du drone pourra quant à elle jouer sur la qualité d'émission et de réception du signal ainsi que sur les vibrations du système. Celles-ci pourraient erroner les données provenant des capteurs.

Le 2ème type de contraintes est lié au déroulement du projet. En effet, étant donné l'importance du projet, il est primordial d'avoir une idée assez précise de comment atteindre l'objectif. Pour cela, 2 points importants sont à étudier :

- Le premier est au niveau financier. Pour un projet avec les spécifications données, il faut compter environ 400€. Ce qui représente une importante somme d'argent.
- Le second est au niveau du temps à y consacrer. Les 100 heures prévues pour les projets tuteurés ne seront en aucun cas suffisantes pour le terminer. Il faut donc être prêt à passer une partie de son temps personnel dessus.

Présentation des différents sous-ensembles et des différentes possibilités envisagées:

Choix de la forme du drone :

Il existe un certain nombre de type de drone. Chacun ayant ses particularités. Voici un bref aperçut des plus courant d'entre eux :

Drone de type avion :

Ces drones sont tout particulièrement faits pour faire de longues distances et des pointes de vitesse. Ils peuvent également servir à faire des figures. Ils peuvent être motorisé ou non. Si celui-ci ne dispose pas de motorisation, il s'agit alors d'un planeur.

Drone de type aile volante

Ce type de drone se rapproche beaucoup de l'avion sur le fait qu'il utilise également une surface de portance, sauf qu'à la différence de l'avion qui dispose d'une aile de chaque côté, l'aile volante, comme son nom l'indique n'en a qu'une seule qui représente l'ensemble de sa surface. Ils sont faits essentiellement pour la vitesse.

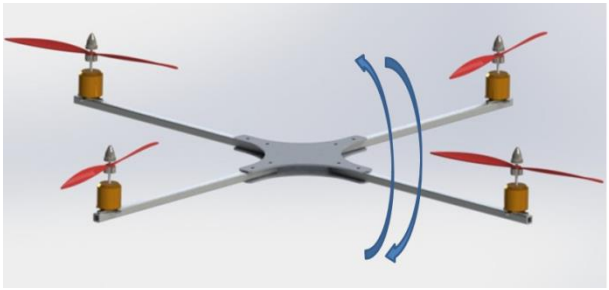
Drone de type hélicoptère/multicoptère :

Ceux-ci utilisent le même principe que l'hélicoptère. C'est-à-dire une voir plusieurs hélices qui tournent suffisamment vite pour permettre à l'appareil de s'élever. La particularité de ces appareils est qu'ils permettent de faire du surplace. Pour mon projet, j'ai choisis de réaliser un multicoptère, plus précisément un quadricoptère. J'ai choisi ce type d'appareils car la gestion des mouvements est plus simple et plus intuitive qu'un hélicoptère (cf. description des C.U.s). De plus, l'armature de celui-ci sera en aluminium tubulaire carré. J'ai choisis ce matériel pour sa légèreté et le fait d'utiliser des tubes carrés augmente sa rigidité et me permettra de faire passer les câbles à l'intérieur de ceux-ci.

Gestion du tangage

Présentation :

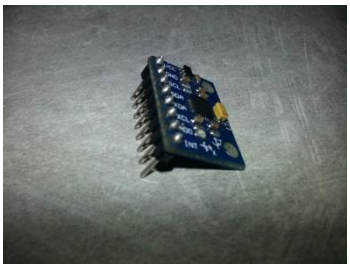
Le tangage est le mouvement d'inclinaison vers l'avant ou l'arrière de l'appareil. Sur un drone de type avion, le tangage permettra à celui-ci de se diriger vers le sol si il tangue vers l'avant ou de monter dans le cas contraire. Sur un drone de type quadricoptère, le tangage permet à celui-ci de se diriger vers l'avant ou l'arrière.



Solution envisagée :

Utilisation du gyroscope :

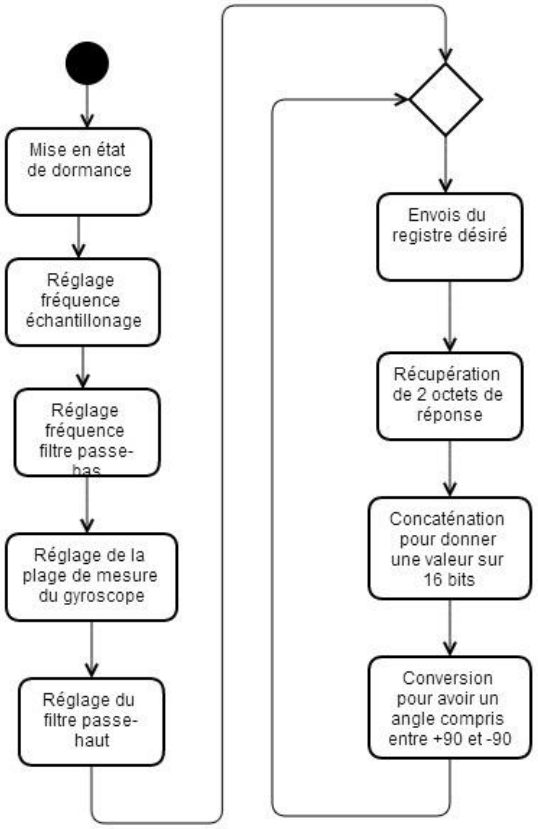
La solution qui m'est venu de suite est l'utilisation d'un gyroscope. Grâce à ce module, je peux connaître l'angle d'inclinaison du système aussi bien pour le roulis que pour le tangage.



Le choix du gyroscope s'est porté sur un MPU-6050 pour plusieurs raisons. La première étant son prix. En effet, celui-ci ne m'a coûté que 2.98€. Ce qui est très peu lorsque nous regardons les prix que proposent certains fabricants. La 2ème raison de ce choix fut sa connectivité. Etant donné que je souhaite utiliser le moins de ports possible sur le microcontrôleur pour en faciliter la configuration et réaliser une petite économie d'énergie, j'ai choisis de brancher tous mes périphériques sur un même bus I²C. Fonction que proposait ce capteur. De plus, ce capteur permet également de connaître l'accélération sur les 3 axes. Ce qui sera une information fort utile lorsque je devrais calculer la poussée que devront procurer les moteurs.

Fonctionnement détaillé de la solution :

N'ayant pas pu trouver d'informations intéressantes dans la documentation (pourtant très fournie) du MPU-6050, je me suis basé sur un exemple de code fourni sur internet. Les commentaires de celui-ci étant en chinois, je n'ai pas pu vraiment comprendre tout le sens de chaque commande. J'ai donc préféré utiliser les valeurs fournies dans ce code qui semblent fonctionner mais qui ne sont peut-être pas les meilleurs pour mon système. Voici ce que j'ai pu comprendre : ->



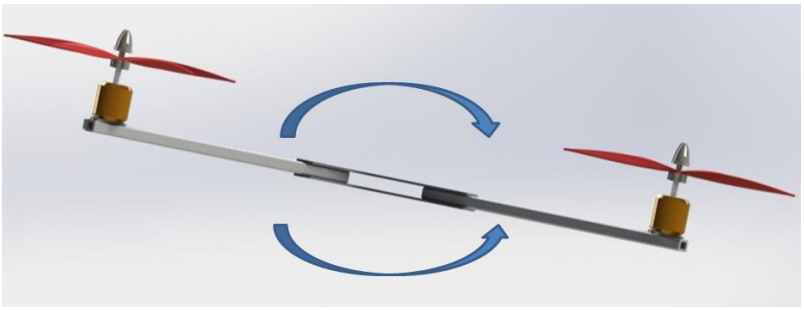
Pour connaître l'angle ou l'accélération sur un axe, il suffit de donner le registre correspondant au début de la boucle. Le contenu de la réponse sera alors la réponse désirée. Il y a donc 6 registres à interroger si nous souhaitons avoir l'angle et l'accélération sur les 3 axes.

L'inconvénient de ce système est qu'il ne détectera pas si l'appareil se retourne. Donc au lieu de se remettre en bonne position, celui-ci accélèrera jusqu'à toucher le sol. Je pense qu'il est possible de contrer ce problème mais je n'ai pas encore réalisé le morceau de code qui s'en chargera.

Gestion du roulis

Présentation :

Le roulis est le mouvement d'inclinaison sur la droite ou la gauche du drone. C'est grâce au roulis que le drone pourra se diriger sur les côtés. Cependant, il faut faire attention à ne pas adopter un angle de roulis trop important au risque de voir l'appareil décrocher, ce qui résulterait en une perte de contrôle pour le pilote.

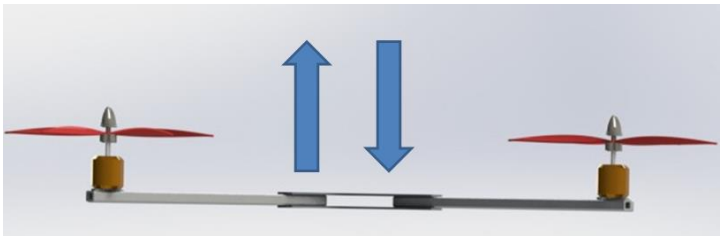


La solution utilisée pour répondre à ce problème est la même que celle pour le tangage. S'y référer pour le fonctionnement détaillé.

Gestion de l'altitude

Présentation :

L'altitude correspond la hauteur de vol de l'objet par rapport à un référentiel terrestre préétabli. Celui-ci peut être basé sur le niveau de la mer ou sur la distance réelle par rapport au sol ou bien encore la distance au sol à l'emplacement du décollage.

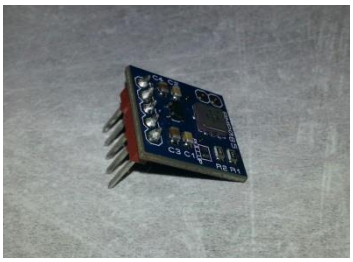


Solution envisagée :

Utilisation de l'ultra son :

Une idée relativement intuitive pour répondre à ce problème, serait l'utilisation de capteurs à ultra-sons qui seraient disposés sous le drone, pointant en direction du sol. Cette solution à l'avantage d'être peu couteuse et m'aurait permis d'utiliser une technologie que je connaissais déjà. Cependant, l'utilisation des ultra-sons ne permet pas de monter à une altitude très élevée. En effet, la distance de détection de ces capteurs n'est que d'environ 2 mètres. Ce qui est très nettement inférieur à ce que je souhaite atteindre. J'ai donc finis par proscrire cette solution.

Utilisation d'un capteur de pression :



La deuxième solution que j'ai eue m'est venue en parcourant les sites dédiés au modélisme. En effet, lorsque les appareils sont destinés à monter relativement haut, il est alors conseillé de se baser sur la pression atmosphérique pour connaître l'altitude. Cette méthode est certes moins précise que les capteurs à ultra-sons (entre 15 et 40cm selon le capteur) mais permet d'atteindre des altitudes bien plus élevées. Dans le cas du capteur que j'ai choisis (le BMP085), l'altitude maximale est de 9km (ce qui est très nettement au-dessus de ce qui sera demandé au drone) et la précision est d'environ 25cm. Celui-ci utilise également le protocole I^2C pour communiquer et j'ai pu trouver ce capteur pour 9.64€ sur internet.

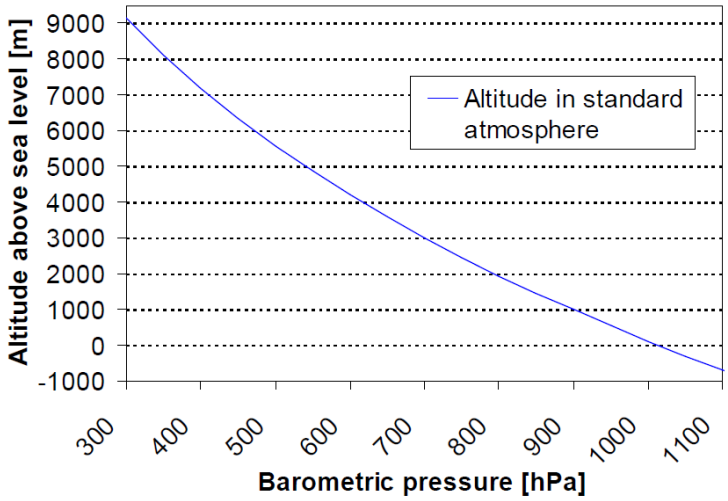
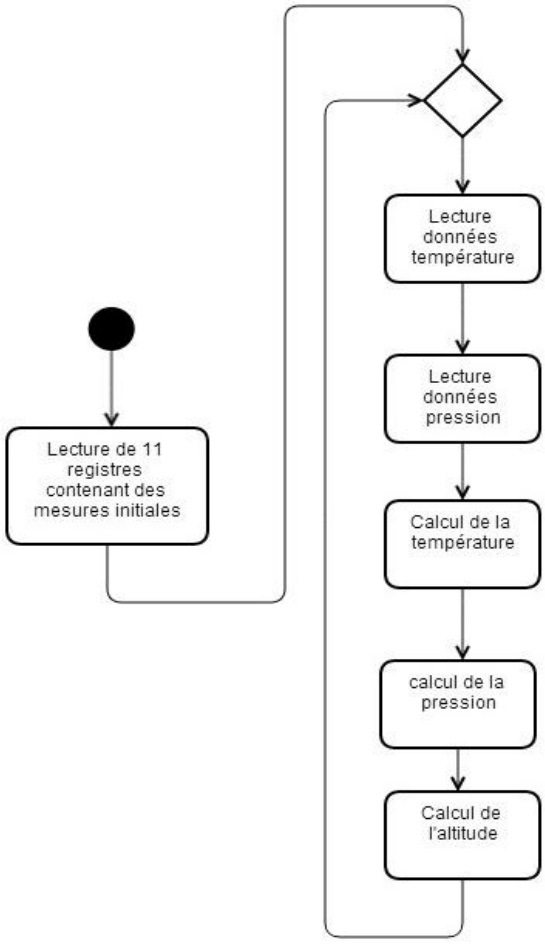


Figure 1: Rapport altitude / pression atmosphérique

Fonctionnement détaillé de la solution :

Le plus gros du travail avec cette solution, est la conversion de la température (également relevée par le capteur) et de la pression atmosphérique en une altitude. Cependant, tous les calculs sont bien expliqués dans la documentation de celui-ci. Voici un aperçu des différentes étapes de l'algorithme :

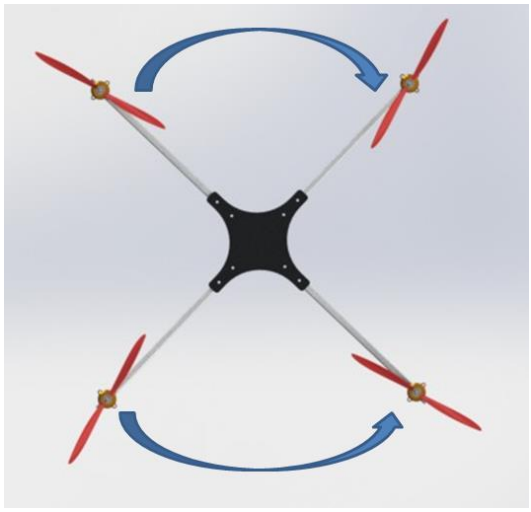


Cependant, un gros problème persiste sur cette partie du programme. En effet, les lignes de code relatives au calcul de l'altitude font planter le bus I²C avant même qu'elles ne soient exécutées. Aucune solution n'a été trouvée pour le moment.

Gestion du lacet

Présentation :

En aéronautique, le lacet correspond à la rotation autour de l'axe de la verticale. Il est important que cette rotation soit maîtrisée. Dans le cas contraire, il serait compliqué pour le pilote de maîtriser la direction de l'engin.



Solutions envisagées :

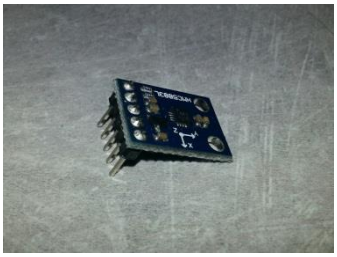
Utilisation du gyroscope :

La première solution envisagée fut d'utiliser l'axe correspondant qu'offrait le gyroscope choisit pour la solution concernant le tangage et le roulis. Cette solution me permettait de ne pas avoir à acheter de matériel supplémentaire et également de n'avoir que très peu de code à rajouter et donc une occupation système limitée. Cette solution permettait également de limiter la place occupée sur le PCB final. En effet, il est important de veiller à avoir un PCB minimal pour plusieurs raisons. La première est qu'un PCB plus large est systématiquement plus lourd. Et il faut garder à l'esprit que le poids est l'ennemi numéro 1 de tout bon matériel volant. Le deuxième avantage à avoir un PCB minimal est le prix qu'il coutera à réaliser. En effet, les fabricants calculs le prix de celui-ci en fonction de sa surface. Un PCB plus grand entrainera donc un coût plus important.

Cependant, pour le moment cette solution n'a pas été retenue car il m'a été impossible de récupérer les données sur l'axe correspondant. Je reviendrais peut-être sur cette solution plus tard, mais il m'a semblé important d'avancer sur des points plus complexes.

Utilisation de la boussole

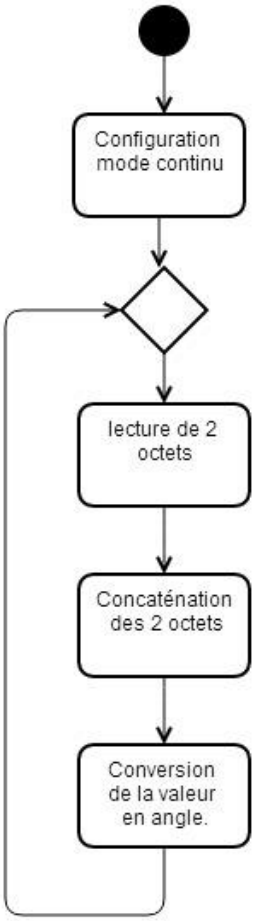
La solution que j'ai finalement adoptée est l'utilisation d'une boussole électronique. Avec cet élément, je connaîtrais en permanence mon angle en fonction du Nord. Il me suffira donc de me baser sur celui-ci. De plus, ayant déjà eu l'occasion de travailler un peu sur ce type de matériel, je savais que celui-ci pourrait répondre à mes attentes. Le capteur que j'ai choisis d'utiliser est le HMC5883L. J'ai pu trouver celui-ci pour environ 8€. De plus, celui-ci communique via le bus I²C, ce qui correspond à mes attentes.



Fonctionnement détaillé de la solution :

Le fonctionnement de cette boussole est extrêmement simple. Il me suffit de lire régulièrement sur le bus I²C 2 octets provenant de ce capteur qui a été préalablement placé en mode continu (envoi de données en permanence). Une fois la lecture terminée, je concatène les 2 valeurs pour former une variable sur 16 bits. Il me faut ensuite convertir cette donnée pour avoir un angle.

Pour le moment, certaines valeurs provenant de la boussole sont lues. Cependant celles-ci ne correspondent pas exactement à ce que j'espérais. Il faut donc que je continu à creuser de ce côté-ci.



Gestion de la communication

Présentation :

Pour permettre à l'utilisateur de piloter le drone, une communication sûre doit être établie. Celle-ci doit permettre un protocole permettant le contrôle de l'engin sur tous les axes et éviter au maximum les erreurs de communication. Il faut également que celle-ci soit fonctionnelle sur la distance demandée. C'est-à-dire 1.5km.

Solutions envisagées :

Radio fréquence:

La radio fréquence est le système de base utilisé dans le modélisme. Ce système à la particularité d'être très robuste et de permettre l'émission sur plusieurs canaux. Ce qui permet d'envoyer plusieurs informations à la fois.

Wi-Fi:

Le Wi-Fi est un mode de communication sans fil. On retrouve beaucoup celui-ci dans les box d'accès à internet. Celui-ci à une portée pouvant aller jusqu'à 150m. Ce qui est très nettement insuffisant dans mon cas. Je n'ai donc pas choisi ce module.

XBee Pro Série 2:

Le XBee Pro Série 2 est un petit module permettant d'avoir une liaison série avec un autre module du même type. Ce module a été conçu pour être aussi simple à utiliser qu'une liaison filaire. De plus, sa portée est de 1.5km en champ libre. Il pourrait donc convenir à l'usage que j'ai à en faire. Cependant, ce produit à un coût. En effet, il faut compter environ 40€ par module. Sachant qu'il m'en faut 2 (un sur la partie embarquée et un sur la partie commande) et qu'il faut également un appareil permettant de les configurer (environ 10€), il faudrait que je compte environ 90€ uniquement pour la transmission.

Etat actuel du CU

Mon choix serait actuellement d'avantages axé sur les modules XBee mais de récentes recherches m'ont permises de m'apercevoir qu'un module de radio fréquence pourrait être également utilisable. Aucun choix n'est donc fixé pour le moment et la transmission reste filaire.

Choix des moteurs

Présentation :

Le choix des moteurs est à faire en fonction du poids qu'ils devront porter. J'ai pu réaliser ce calcul à l'aide d'une feuille Excel que j'ai pu trouver sur internet et qui permettait de donner une idée des moteurs, des hélices, des variateurs et de la batterie à choisir en fonction du poids et de l'autonomie de l'appareil.

Solution envisagée :

Il en a donc résulté que le bon choix serait des moteurs brushless proposant une puissance de 1000kv (1000 tours par minute pour un Volt). Le voltage de ceux-ci est à choisir en fonction de la batterie utilisée. Mais il faudrait que je m'oriente vers 4 moteurs alimentés entre 7.4 et 14.8V et alimentés en 20 Ampères.

Choix des ESC

Présentation

Les ESC (pour Electronic Speed Control) permettent de faire varier la vitesse des moteurs. Ils reçoivent les informations en provenance du microcontrôleur et adaptent les 3 phases du moteur brushless connecté dessus pour en faire varier la vitesse. On les trouve aussi sous le nom de variateur ou contrôleur.

Il existe plusieurs types de variateurs. Certains sont commandés par une PWM. Plus le rapport cyclique sera grand, plus la vitesse de rotation sera élevée. D'autre sont connectés sur un bus I^2C. Le microcontrôleur envois donc la vitesse sur ce bus et si l'adresse correspond à un variateur, celui-ci s'adaptera en fonction de ces données.

Les variateurs I^2C proposent une vitesse de rafraichissement plus élevée et permettent de n'occuper que 2 fils pour connecter tous les ESC au microcontrôleur. Cependant, je ne sais pas encore si ces avantages seront réellement bénéfiques comparé au prix auquel ils sont proposés.

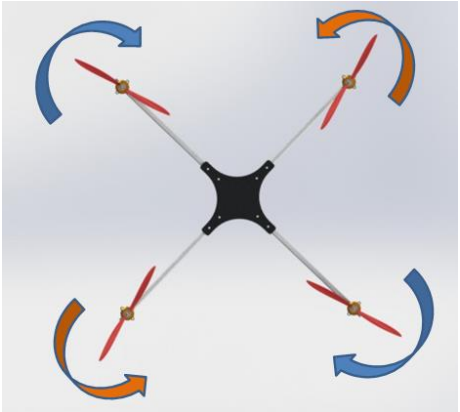
Etat actuel du CU

Pour le moment aucun variateur n'a vraiment été sélectionné. Ceux-ci le seront au moment du choix définitif des moteurs.

Choix des hélices

Présentation :

Les hélices seront fixées directement aux moteurs. C'est elles qui porteront le quadricoptère. Il faut donc qu'elles soient suffisamment grandes pour avoir une surface de portance assez élevée mais pas trop pour pouvoir être entrainées par les moteurs sans que ceux-ci ne forcent. Il faut également choisir un pas d'hélice adapté. En effet, un pas trop petit contraindra les moteurs à faire plus de tour pour que le système s'élève. Un pas trop grand risque de demander un couple trop élevé aux moteurs.



Solution envisagée

Le choix des hélices est à faire en liens avec le poids du quadricoptère et des moteurs choisis. Dans mon cas, d'après la même feuille de calcul qui m'a permis de déterminer le type de moteurs à utiliser, il faudrait que j'installe des hélices de 10 pouces et un pas de 4.5 pouces.

Lors de l'achat des hélices, il est important de prendre 2 hélices avec un pas dans le sens horaire et 2 hélices avec un pas dans le sens trigonométrique. En effet, étant donné que 2 moteurs tourneront dans le sens horaire et 2 dans le sens trigonométrique pour éviter que le quadricoptère ne tourne en permanence sur lui-même, il est important d'avoir des hélices adaptées.

Choix de la batterie

Le choix de la batterie est encore assez flou. Toujours est-il que celle-ci devrait être suffisamment puissante pour pouvoir alimenter les 4 moteurs ainsi que le reste du système. Cette batterie sera de technologie LiPo (Lithium Potassium) car malgré son instabilité (inflammabilité) celle-ci est celle qui offre le meilleure ration poids/puissance. J'opterais également pour une batterie 4 cellules qui d'après mes recherches sur internet offre une bien meilleure autonomie que les 3 cellules.

Il faudra bien entendu que j'achète un chargeur pouvant recharger cette batterie sans l'endommager.

Gestion de la vidéo

Présentation

La dernière étape de ce projet serait de mettre un système vidéo pour pouvoir piloter le drone en FPV. Le FPV (First Person View) est le fait de piloter le drone en se basant sur les images provenant d'une caméra embarquée. Ce système permet de piloter sur de plus longue distance, ce qui ne pouvait être fait au-par-avant. En effet un visuel est nécessaire pour se diriger et prévenir d'une éventuelle perte de contrôle du système.

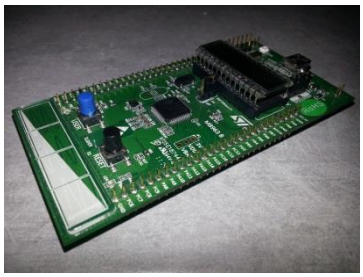
Solution envisagée

Concernant la vidéo, je pense utiliser une caméra de type GoPro ou un modèle équivalent mais moins cher. Concernant la transmission, ma première idée fut d'utiliser le même module que celui qui me permet de guider l'appareil. Cependant, il semblerait que la vitesse de celui-ci ne soit pas suffisante. J'ai donc décidé d'utiliser un module spécialement prévu à cet effet et qui permet d'atteindre 1km. Afin d'étendre sa portée à 1.5km, il me faudra changer l'antenne par une PinWheel qui de par sa forme permet d'émettre plus loin.

Figure 2: Antenne PinWheel

Choix du microcontrôleur

Pour réaliser ce projet, je me suis naturellement lancé sur un microcontrôleur STM32 de chez ST. Si j'ai choisis ce composant, c'est uniquement car j'ai déjà eu l'occasion de travailler dessus lors de projets durant mon DUT informatique. Sachant que celui-ci pourrait répondre à mes besoins (bus I²C et fréquence en particulier), j'ai choisis d'acheter une plaque de démonstration vendue par ST incluant le microcontrôleur, un debugger et un certain nombre de périphériques pour 10€. De plus, cette plaque peut être installée sur une breadboard (plaque de prototypage), ce qui me facilite considérablement toute ma phase de tests.

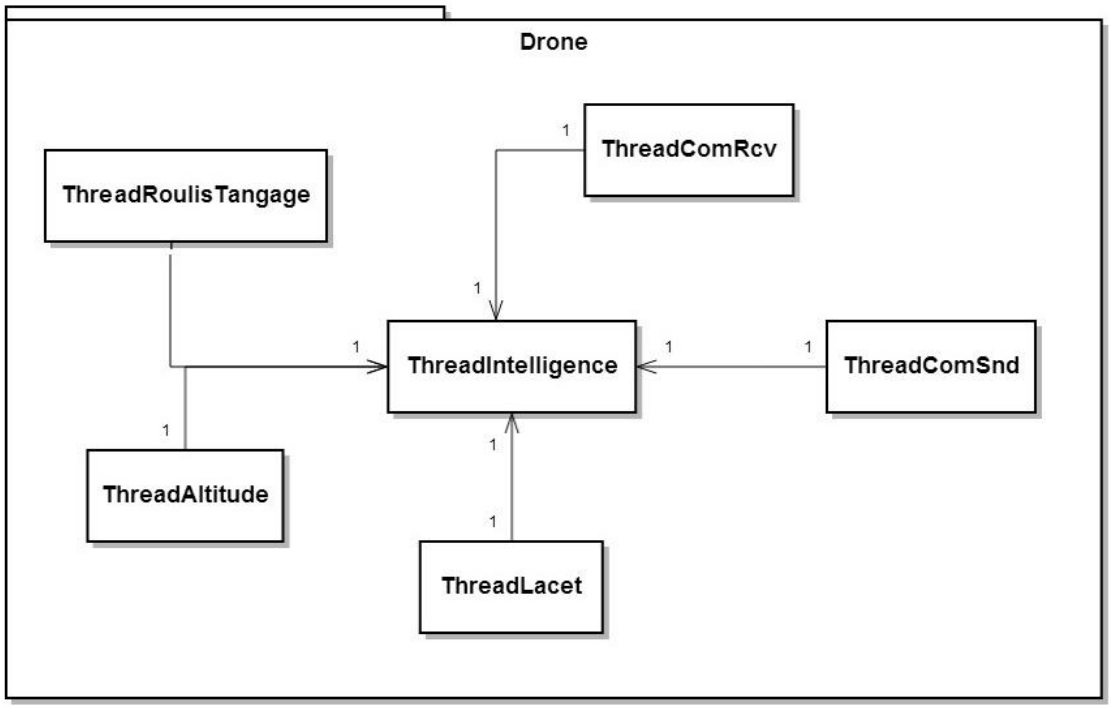


Choix du système d'exploitation

Il m'a semblé judicieux pour un projet de cette importance d'utiliser un système d'exploitation qui me permettrait d'avoir plusieurs tâches qui tourneraient en même temps. Mon premier choix ses basé sur FreeRTOS. En effet, j'ai eu l'occasion durant mon DUT Informatique de travailler sur cet OS. Cependant, j'ai dû abandonner cette voie car même si celui-ci existait pour mon microcontrôleur, il n'avait été porté qu'à l'aide de l'environnement de développement iAR qui est une suite payante et extrêmement chère. J'ai donc choisi de m'orienter vers un autre O.S.

Mon 2ème choix fut chibiOS. Cet O.S. est fourni sur un grand nombre de microcontrôleurs différents et qui plus est, sur la suite de développement que j'ai pris l'habitude d'utiliser durant mon DUT : Keil. Cet O.S. offre une bonne couche d'abstraction avec les différents périphériques de mon microcontrôleur. Cela me permet donc par exemple d'utiliser le bus I²C sans avoir à manipuler directement les registres de celui-ci.

Voici comment sont organisées les tâches à l'aide de ChibiOS :



La tache ThreadIntelligence permet le calcul de la poussée à mettre sur chaque moteur. Pour le moment, étant donné que je n'ai pas encore commencé à travailler sur ceux-ci, aucune tâche ne leur est destinée mais peut-être en sera-t-il autrement dans les prochaines versions.

Calcul de stabilisation

Une très grosse partie du travail consistera en la mise en place de l'ensemble des calculs qui permettront au drone de se stabiliser. Malgré les quelques recherches que j'ai commencé à faire et avec les documents fournis par Mr Gomez, je n'ai, pour l'instant, pas suffisamment avancé sur le sujet pour présenter quoi que ce soit.

Partie commande

Présentation

Concernant la partie commande, j'ai choisis d'utiliser mon ordinateur plutôt que d'utiliser les télécommandes traditionnelles. Mon choix s'est porté sur cette technologie au vue du prix de ces télécommandes. En effet, il faut compter minimum 150€ pour ce type de matériel et certainement plus pour atteindre les 1500 mètres. De plus, le but de ce projet était d'en réaliser la plus grande partie possible moi-même.

J'ai donc réalisé une interface graphique en C fonctionnant sur l'ordinateur à l'aide de la bibliothèque SDL. J'utilise également un port com. de mon ordinateur pour communiquer avec la maquette. Pour le moment, la transmission n'est que filaire étant donné que je ne me suis pas encore réellement décidé sur quel matériel je vais travailler pour la communication.

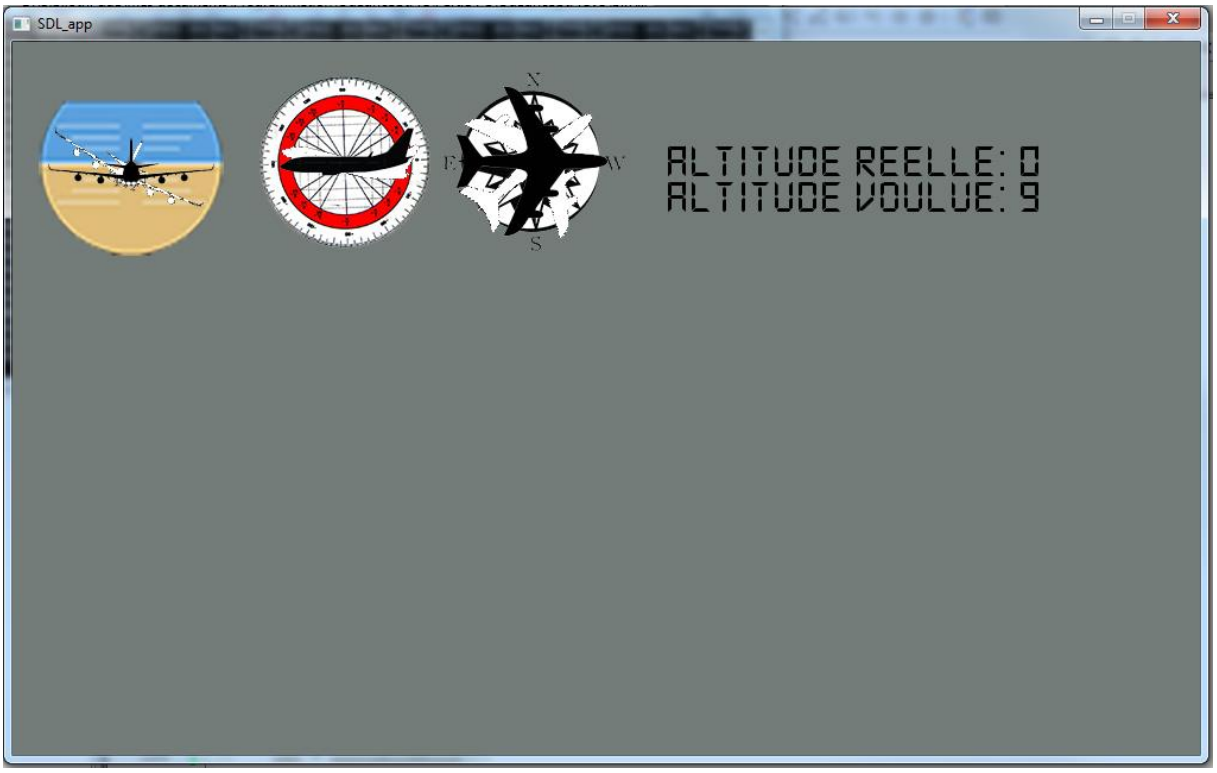


Figure 3: Etat actuel de l'interface graphique

Description de l'interface graphique :

De gauche à droite, nous avons les éléments suivants : Aperçut du roulis, aperçut du tangage, aperçut du lacet, altitude.

Pour chaque donnée, nous avons l'état réel de l'appareil (en noir sur les jauges) et l'état désiré (en blanc). J'ai choisis de faire la différence entre les 2 états pour 2 raisons : La première raison concerne le temps de réaction de l'appareil. Avec cette méthode, on a un aperçut pratique de si l'appareil a atteint ou non la position demandée. La 2ème raison prend son sens lorsque l'appareil est déplacé par un élément extérieur au système, par exemple le vent ou si celui-ci a touché le sol.

Pour le moment, pour commander l'appareil, l'utilisateur a besoin des touches Z, S, Q, D ainsi que de la souris. Voici la répartition des commandes :

- Z et S : faire monter et descendre le drone (commande de l'altitude)
- Q et D : faire tourner sur lui-même le drone (commande du lacet)
- Souris vers la gauche et vers la droite : Faire pencher le drone sur la droite ou la gauche (commande du roulis)
- Souris vers le haut et vers le bas : Faire penchez le drone vers l'avant ou vers l'arrière (commande du tangage)

Dans une version future, j'envisage de remplacer la souris par un joystick ou voir même remplacer tous les éléments de pilotage par un système de leap motion (contrôle à l'aide des gestes de la main).

Communication

La communication entre le système embarqué et la partie PC est faite par une liaison série. Sur cette liaison, j'ai choisi d'envoyer une trame qui me permet de stocker toute les données relatives à la position du quadricoptère. Voici comment la trame se forme :

Txxx	Rxxx	Lxxx	Axxxxx	S
Tangage	Roulis	Lacet	Altitude	Stop

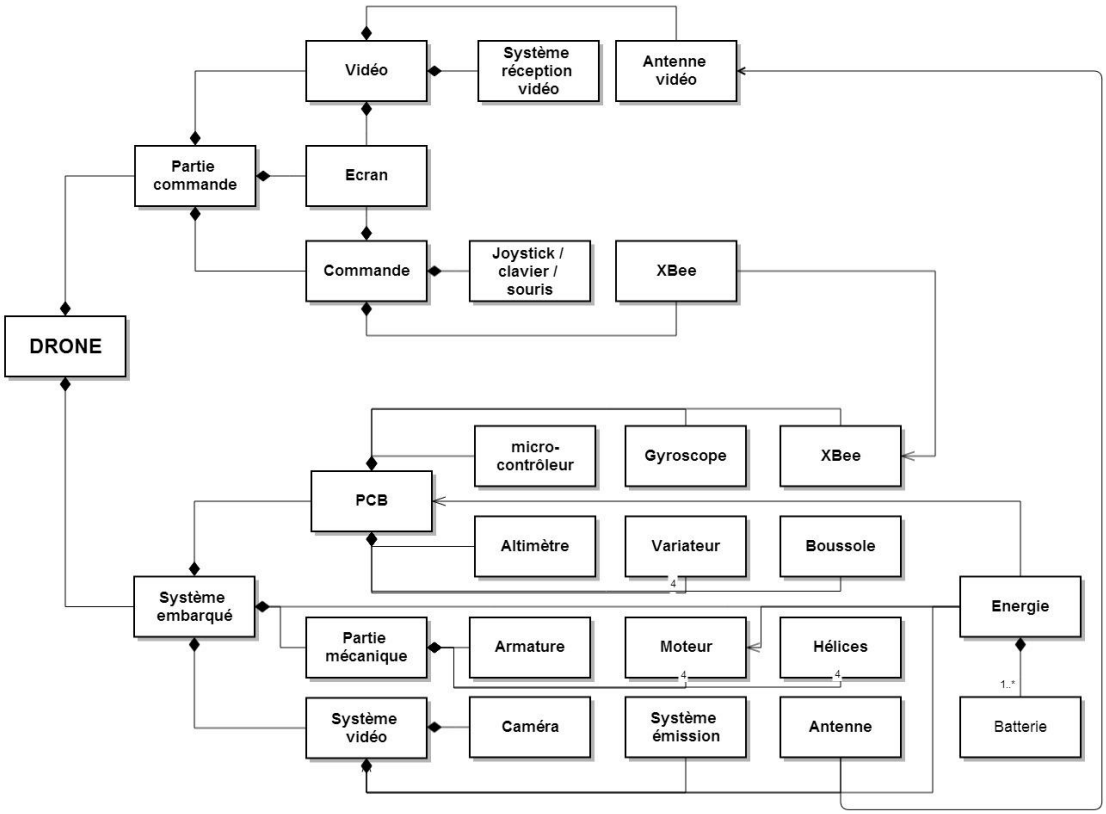
Un exemple de trame pourrait donc être : T123R012L023A12345S

Dans cet exemple, nous aurions un tangage de 123° et un roulis de 12° par rapport à l'horizontal, un Lacet de 23° par rapport au Nord et une altitude de 12345 cm par rapport au point de départ.

Bilan du CU

Concernant le bilan de cette interface graphique, je pense la reprendre de 0 étant donné que le code est assez brouillon. De plus, je pense que le langage que j'ai utilisé n'est pas forcément le bon. En effet, j'ai décidé de faire celle-ci en C uniquement car il s'agissait du langage que je maîtrisais le mieux. Cependant, je pense la refaire en C++ pour améliorer l'allure général de l'interface et avoir un code plus fonctionnel. De plus, si j'intègre le leap motion comme j'aimerais le faire, le constructeur propose une librairie C++ et non C.

Résumé de l'organisation du drone



Conclusion

Cette première moitié de projet à, selon moi plutôt bien avancé. Surtout en ce qui concerne le gyroscope. Cependant, certains problèmes restent présents et pour le moment, je n'ai toujours pas la moindre idée de comment résoudre ceux-ci. Le principal étant le problème lié à l'I²C et les calculs de l'altimètre.

Je ne pense pas terminer ce projet dans les 100 heures imparties. Cependant, je commence à y voir plus claire sur comment parvenir à atteindre mon objectifs et les difficultés auxquelles je vais faire face.