

The Drone Project

Présentation	1
Schéma électrique	1
Algorithme de fonctionnement	2
Coût du projet	3
Comparaison du planning	3
Problèmes rencontrés	3
Conclusion	4

Présentation

L'objectif initial de notre projet était de concevoir un drone à voilure fixe, imprimé entièrement en 3D, et capable de voler et de maintenir une trajectoire rectiligne en autonomie. Ce projet avait pour but de développer une plateforme nous permettant de tester et de mettre au point des prototypes de plus en plus opérationnels ultérieurement.

Nous avons choisi de diviser le projet en deux majeures parties:

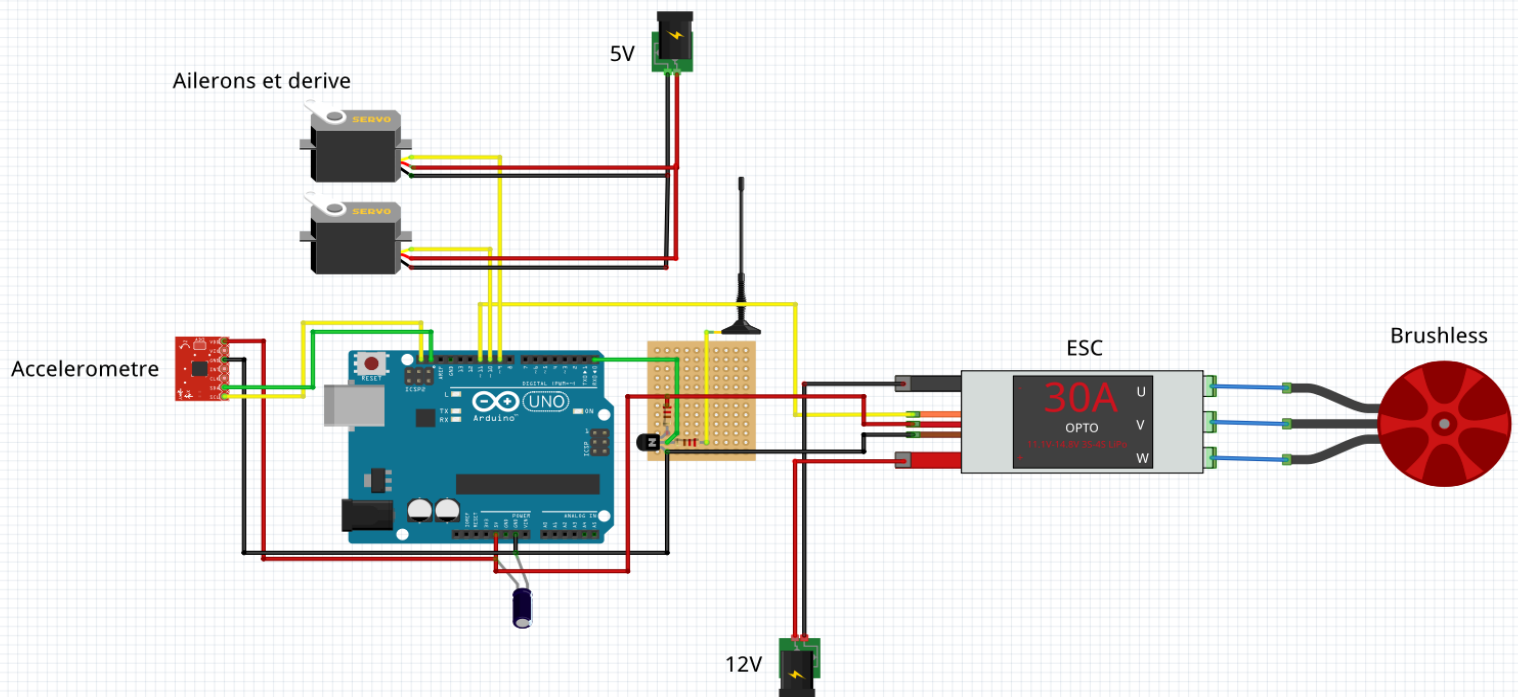
- La partie "low level" qui consiste en le développement d'un hardware fiable, d'un logiciel simple et adaptable, et de modes de contrôles basiques.
- La partie autonomie, qui consiste en le développement d'un logiciel de télémétrie et de guidage, l'ajout de nombreux capteurs pour implémenter la navigation autonome, et la navigation par balises.

Schéma électrique

Le schéma électrique mentionne deux sources de tensions (12v et 5v), mais elles peuvent être simplifiée a une seule (11.6V) provenant d'une batterie LiPo et passant par un regulateur lineaire type L7805 ou un buck converter pour fournir le 5v de l'Arduino et des Servo.

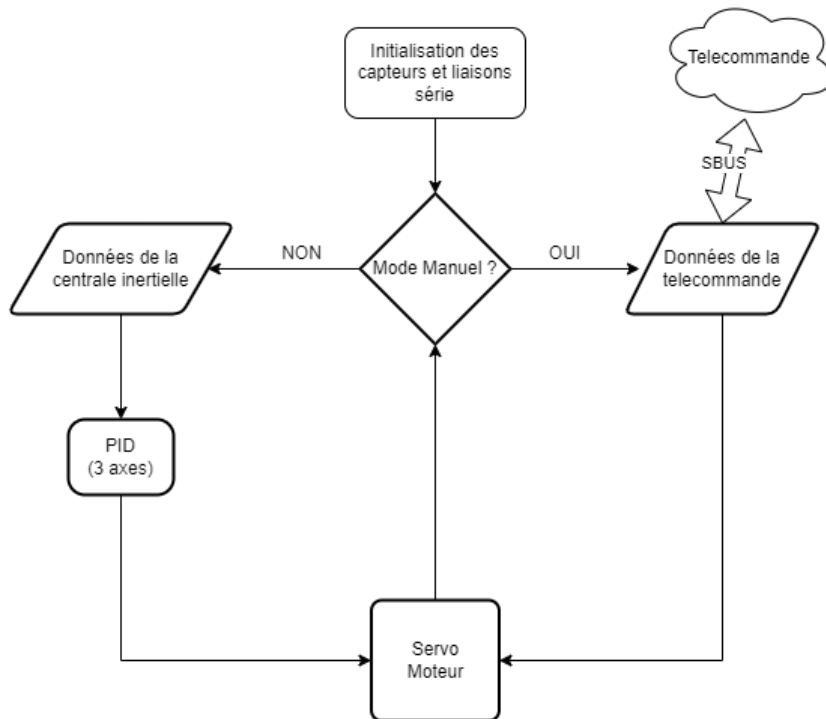
D'un point de vue hardware, tous les composants sont connectés au shield arduino sur lequel nous avons soudé l'inverseur logique, les bus I2C, des connecteurs 3 pin pour les servo...

Cela permet de minimiser le poids et le volume des circuits pour optimiser le vol.



Algorithme de fonctionnement

Le programme tel qu'il était à la fin du projet et durant la présentation est très simple et montre les fonctionnalités implémentées à la fois en Hardware et Software, bien que beaucoup d'autres codes et classes ont été développées pour contrôler les autres capteurs (Baromètre, Altimètre...) ainsi qu'une classe globale Drone pour facilement interfacer d'autres programmes en gardant une syntaxe simple et claire. Le code utilise des library déjà existantes mais modifiées pour notre programme.



Coût du projet

Cout du materiel :

Drone 3D : 8€

Moteurs/Servo : 40€

Electronique : 20€

Télécommande : 150€

Nombre d'heures :

Romain : 26 (heures de td + heures de développement et impression 3D) 624€

Robin : 75 (TD + développement) 1800€

Comparaison du planning

Le planning initial était pour le moins flou, car nous ne savions pas encore à quoi nous attendre pour le temps de développement de chaque partie du projet. Nous avons commencé au tout début de l'année, pour prendre de l'avance, et implémenter plus de fonctionnalités. Nous avons ensuite travaillé chaque séance sur le hardware, avec les soudures, l'impression 3D et le montage du drone, et le temps libre pour coder le software du drone.

Problèmes rencontrés

Tout d'abord, nous avons eu des difficultés à imprimer le drone en 3D puisqu'il représente ~600 g de filament, nous avons dû recommencer deux impressions. Ensuite, interfacer la télécommande avec l'arduino de manière instinctive (utiliser PulseIn pour lire la valeur de chaque canal) était loin d'être suffisamment rapide, donc nous avons dû utiliser le protocole SBUS, assez complexe et manquant cruellement de documentation en ligne, qu'il a fallu déboguer sans Serial puisque la Uno dont nous disposions alors n'en a qu'un. Ensuite il a fallu trouver la datasheet de l'ESC en chinois et la comprendre pour implémenter la procédure d'initialisation. Finalement, le modèle du drone n'était pas suffisamment grand pour accueillir l'électronique et l'arduino; il a donc fallu construire un deuxième drone de secours en polystyrène extrudé. Aussi, les commandes des surfaces de contrôles sont presque inutilisables pour un vrai vol car le fablab ne disposait pas de tiges en métal suffisamment rigides pour que les servo fassent leur travail.

Conclusion

A la fin du projet, qu'avons nous réussi à faire marcher ? Et bien nous avons avec succès réalisé l'objectif de base : une plateforme de test qui puisse servir à développer la partie autonome que nous continuerons de développer après le projet. Nous avons un modèle d'avion radiocommandé qui peut être commandé soit en manuel comme on l'attendrait d'un drone rc, soit en automatique avec des boucles de rétroaction PID permettant de stabiliser le drone

en ligne droite. Nous n'avons pas pu tester le drone autrement qu'en l'attachant, puisqu'on voulait s'assurer d'avoir quelque chose à présenter à la fin. Le logiciel de télémétrie est aussi opérationnel, entièrement fait en QT C++, le code est disponible sur le github. Nous sommes contents de ce que nous avons pu réaliser et en même temps nous avons la sensation de ne pas avoir réellement fini puisque le drone n'a jamais volé.