

Juin 2018

# RAPPORT DE PROJET

## DRODUINO



Encadrants : Pascal MASSO – Fabien FERRERO

Camille BAUCE - ERRIADI Yasmin - Rizlene EL BOUCHIKHI  
PEIP2 – GROUPE 4

# DESCRIPTION DU PROJET

---

## FRANÇAIS

Notre projet consiste en la réalisation d'un drone (nommé Flash McDrone) muni d'une caméra reliée à un casque FPV afin de pouvoir guider le drone sans nécessairement avoir un visuel. De plus, avoir une image directe de ce que "voit" le drone facilite le contournement d'obstacles en vol. Le pilote du drone est comme à l'intérieur du drone : c'est une expérience unique !

Le drone est également muni d'éléments ajoutés à la fois amusants et utiles pour le vol à savoir : des leds de position tout autour du châssis dont les couleurs peuvent être changées, un buzzer déclenchable en cas de perte du drone, des phares dont l'intensité est réglable et une led indicatrice de la batterie. Tous les éléments sont gérés à distance par la télécommande - exceptée la led de batterie que la télécommande n'a pas besoin de commander. Le code qui gère toutes les informations envoyées par la télécommande pour changer les états de chaque élément est stocké sur une carte Arduino pro-micro. Le drone en lui-même est également contrôlé par la télécommande à travers une carte de vol SPRacingF3.

## ENGLISH

Our Droduino's project consists of realizing a drone (named Flash McDrone) with a camera connected to FPV helmet without seeing the drone flying from the outside but instead having a live visual of what the drone actually sees making it easier to avoid obstacles. The pilot is inside the drone: it's a unique experience!

The drone has also added elements both funny and useful for the flight including: position leds all around the frame with changeable colors, a buzzer that is triggered if we lose the drone, headlights with adjustable intensity and a led that indicates the battery level. All those elements are controlled remotely with a remote control - except the led battery that does not need to be controlled by the remote. The computer code that manages all the informations sent from the remote to change the state of each element is stored on an Arduino pro-micro card. The drone itself is also controlled with the remote through an SPRacingF3 flight card.

# SOMMAIRE

---

- I. INTRODUCTION
  - a. Motivations
  - b. Objectifs
- II. MATÉRIEL UTILISÉ
  - a. Matériel principal
  - b. Matériel secondaire
- III. FONCTIONNEMENT
  - a. Théorie
  - b. Pratique
- IV. PROBLEMES RENCONTRES ET SOLUTIONS
  - a. Problèmes d'électronique
  - b. Problèmes d'informatique
  - c. Problèmes d'ergonomie
- V. CONCLUSION
  - a. Synthèse
  - b. Perspectives
- VI. ANNEXE

# I. INTRODUCTION

---

## a) MOTIVATIONS

Dans le cadre de notre projet d'électronique pour le cours d'Arduino nous avons la volonté de mettre au point un objet électronique à la fois télécommandé et avec lequel nous pourrions nous amuser. Après discussion avec Mr Masson, un de nos professeur encadrant, nous avons décidé de réaliser un drone de vitesse avec une caméra connectée à un casque FPV (First View Person).

## b) OBJECTIFS

### ❖ Objectif principale

Le principal objectif de notre projet fut évidemment de réaliser un drone FPV fonctionnel.

### ❖ Objectifs secondaires

Nos objectifs secondaires étaient d'ajouter à notre drone des composants utiles au pilote :

- Une led batterie, placée dans le champ de vision de la caméra FPV qui clignote lorsque la batterie est faible pour indiquer qu'il faut faire atterrir le drone avant qu'il n'y ait plus assez d'énergie (pour le fonctionnement normal du drone).
- Des leds de position placées tout autour du châssis pour que le drone soit visible dans l'obscurité et dont la couleur peut être modifiée par la télécommande.
- Des phares à l'avant du drone pour éclairer l'avant du drone pour le vol de nuit dont l'intensité peut être modifiée à partir de la télécommande.
- Un buzzer que la télécommande peut enclencher à distance en cas de perte du drone.

Et également de terminer le projet à temps pour pouvoir le faire voler.

## II. MATÉRIEL UTILISÉ

---

### a) MATÉRIEL PRINCIPALE

- **Le châssis** : armature générale du drone, squelette
- **Carte de vol (FC)** : Reçoit les informations des différents capteurs internes (récepteurs RC, accéléromètres, gyroscopes) et les traite pour stabiliser l'appareil et la direction du drone. Transmet les informations de commandes aux ESC.
- **ESC** : Contrôleurs de vitesse électronique (x4). Composants électroniques permettant de transmettre les commandes venant de la carte de vol au moteur (en transformant ces ordres en signal électrique). Permettent réguler la vitesse des moteurs.
- **Moteurs (brushless)** : moteurs électriques avec un rotor fait d'un ou plusieurs aimants permettant la rotation du moteur sans système mécanique. Permettent d'entraîner les hélices.
- **Hélices** : dispositif mécanique composés de plusieurs pales disposées autour d'un axe.
- **Batterie** : Fournit l'énergie à tous les autres composants grâce à une plaque de distribution de puissance.
- **Radio Émetteur/Récepteur** : L'émetteur est la télécommande. Il envoie les ordres de directions (en signal radio). Le récepteur est placé sur le drone. Il reçoit les ordres à distance et les transmet à la carte de vol.

### b) MATÉRIEL SECONDAIRE

- **Arduino pro-micro** : Carte possédant un microcontrôleur (architecture Atmel AVR - ATmega32u4 pour la pro micro) sur laquelle on code le comportement (les différents états) des phares, leds strip et buzzer en fonction de ce qu'envoie la télécommande.
- **Buzzer 5V** : Branché à la carte Arduino pro-micro et alimenté en 5V directement par l'Arduino. Emet un son.
- **Phares 5V** : Branchés à la carte Arduino pro-micro et alimentés en 5V directement par l'Arduino aussi. S'allument avec une intensité variable.
- **Leds strip (leds de position)** : Branchées à l'Arduino (où est codé leur comportement) mais alimentées directement par la plaque de distribution de l'énergie.

### III. FONCTIONNEMENT

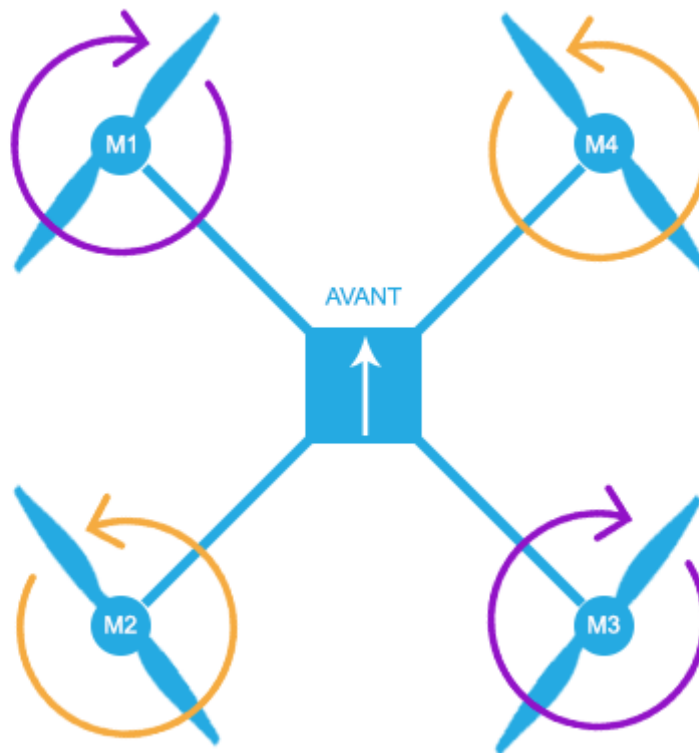
---

#### a) THÉORIE

##### ❖ LA PROPULSION

L'envol du drone repose sur sa capacité à se propulser horizontalement dans les airs. Pour cela un quadcopter, ou drone à quatre hélices, utilise ses quatre moteurs en même temps. Ainsi entraînée, chaque hélice va exercer une force vers le haut sur l'appareil, mais aussi un moment de rotation dans son sens de rotation.

Un quadcopter composé de quatre hélices qui possèdent le même sens de rotation tournerais sans cesse sur lui-même. C'est pourquoi le sens de rotation des hélices doit être inversé d'une hélice à l'autre (les hélices sur une même diagonale ont donc le même sens de rotation), comme ci-dessous :



##### ❖ LA DIRECTION

Pour contrôler le drone, le pilote peut agir sur le throttle, pitch, roll et yaw.

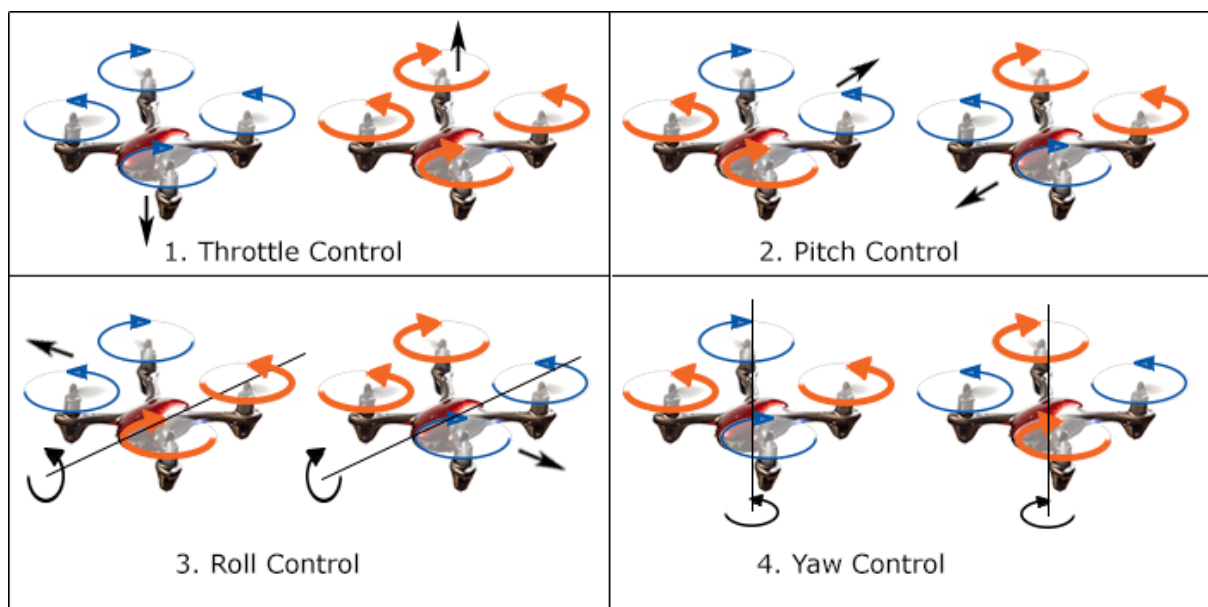
Le **throttle** permet de monter ou descendre en respectivement augmentant et réduisant la vitesse des moteurs (et donc des hélices).

Le **pitch** permet d'avancer ou de reculer en augmentant les deux moteurs avant ou les deux moteurs arrière, et en réduisant aussi les deux autres moteurs. Ainsi la poussée globale vers le haut reste constante.

Le **roll** permet de pencher le drone sur le côté, se mouvant ainsi directement sur le côté. Le roll utilise le même principe que le pitch mais avec les deux moteurs sur le même côté.

Le **yaw** permet de faire tourner le drone sur lui-même en augmentant la vitesse des deux moteurs opposés (le moment de rotation est alors plus fort dans un sens que dans l'autre) et en réduisant celle des deux autres.

Schéma représentant le fonctionnement des différentes directions (la flèche bleue indique que la vitesse de rotation est réduite et inversement les orange indique que la vitesse de rotation augmente) :



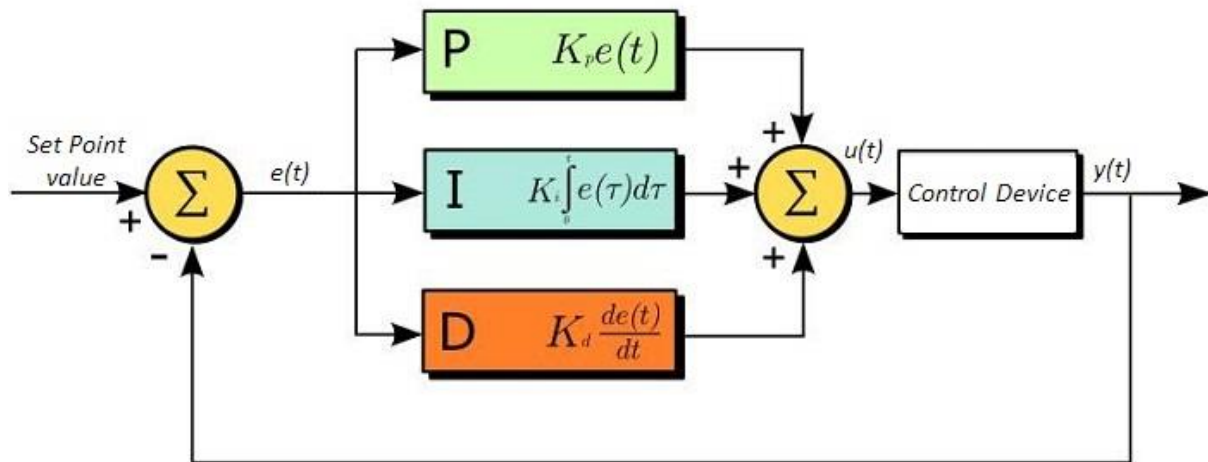
### ❖ STABILITÉ EN VOL

Le drone est stable en vol, c'est-à-dire qu'il ne change de position sur aucun de ses axes lorsque les commandes restent les mêmes. Pour un drone FPV de course, la stabilité de hauteur (donc sur le throttle) n'est pas voulue pour laisser plus de liberté au pilote et n'est donc pas implémenté.

Pour atteindre cette stabilité, la carte de vol recalcule en permanence l'erreur de position par rapport à ce qui est demandé et lui applique alors une correction avant de la transmettre au couple ESC/moteur. Ce calcul est réalisé et optimisé à l'aide de PID, c'est-à-dire qu'il prend en compte les éléments qu'il connaît du passé et du présent pour prévoir la future erreur et éviter que le système soit instable. La subtilité est dans le fait d'arriver à prévoir la future erreur, en effet :

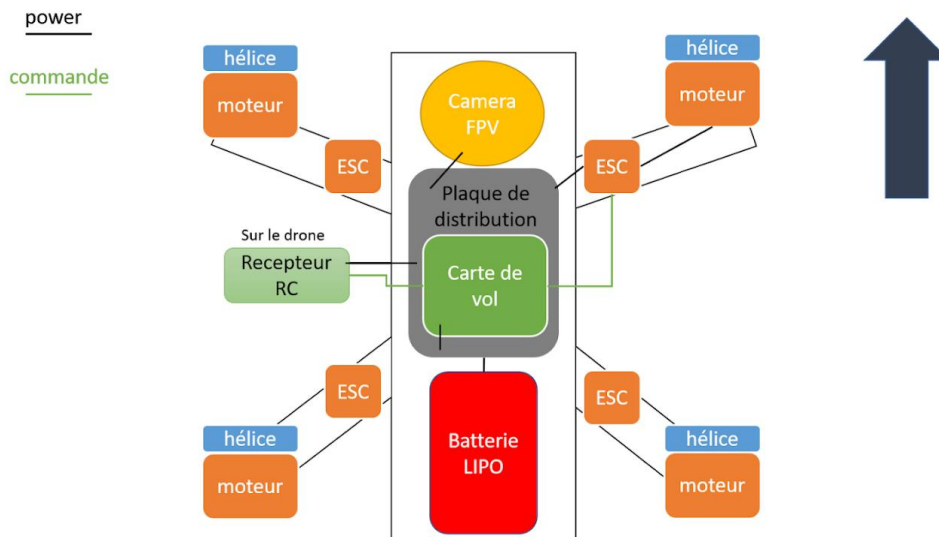
- Si la correction est trop grande, alors l'erreur suivante sera très grande aussi ce qui fera que le système devra appliquer une correction encore plus grande et ainsi de suite. Le drone serait alors de plus en plus instable jusqu'au crash.

- Si la correction est trop faible alors elle ne suffit pas à corriger l'erreur et les erreurs s'accumulent. Le drone ne sera donc plus stable dès que l'environnement ne l'est pas. C'est le cas lorsqu'il y a du vent (même très faible) et au décollage (car les hélices en tournant proche du sol créent des courants)



## b) PRATIQUE

### ❖ SCHÉMA SIMPLIFIÉ D'UN QUADCOPTER



Un quadcopter est donc un drone qui possède quatre hélices et quatre moteurs.

*Le contrôle de l'appareil est ainsi fait :*

L'utilisateur a une télécommande depuis laquelle il peut envoyer les diverses commandes au drone. La transmission des données entre la télécommande et le drone se fait par le biais d'un récepteur radio installé sur le drone que l'on peut aussi appeler *receiver*.

L'information est alors récupérée par la carte de vol, qui est directement branchée au récepteur. La



carte va ensuite traiter la commande et transmettre l'ordre correspondant à sa réalisation aux 4 ESC. Les ESC, à leur tour, vont transformer cet ordre en signal PWM pour leur moteur respectif.

*L'alimentation est séparée en deux parties :*

La batterie est branchée à une carte de distribution d'énergie, transmettant ainsi du 11.3V et du 5V séparément. La carte de vol est alors alimentée par la sortie 5V tandis que tous les ESC sont branchés sur les 11.3V. Le courant nécessaire au fonctionnement des moteurs ne passe donc pas par la carte.

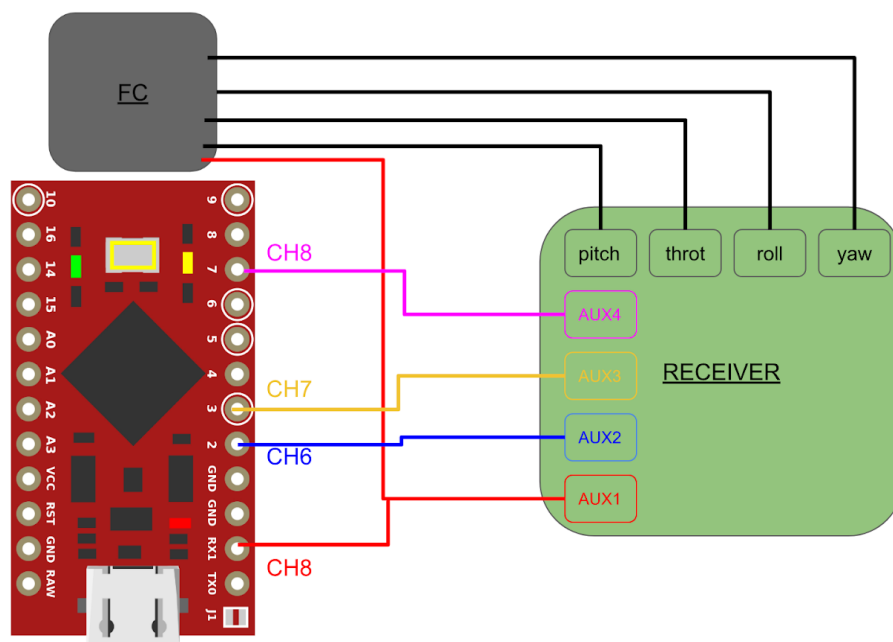
La carte de vol alimente alors le récepteur radio.

La camera FPV et sa batterie forme un système séparé.

## ❖ LA RECEPTION DES CANAUX AUXILIAIRES

Afin d'ajouter nos fonctionnalités supplémentaires au vol qui sont un buzzer, des leds et des phares nous avons besoin de les contrôler. Ce contrôle se fait depuis la télécommande grâce aux switches et boutons non utilisés pour le vol. Sur le *receiver* (et configurable sur la télécommande elle-même), seulement quatre de ces boutons sont associés à quatre canaux différents.

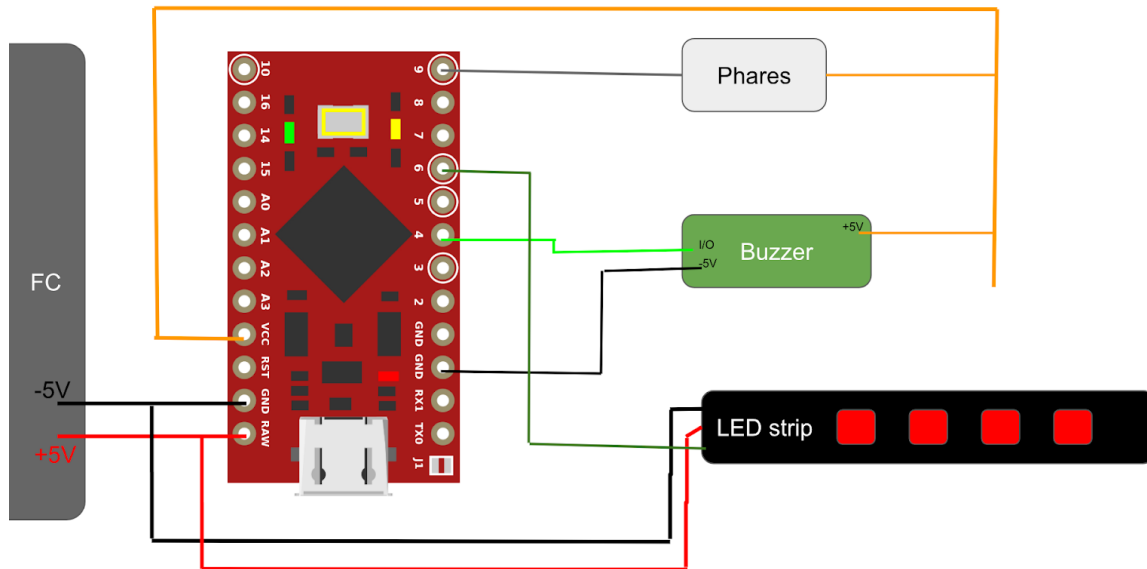
Afin de pouvoir récupérer, lire et traiter ces canaux, nous avons utilisé une Arduino pro-micro que nous avons branché avec le *receiver* de la façon suivante :



Les signaux envoyés sont des signaux PWM. Pour les lire nous utilisons des fonctions d'interruptions disponible uniquement sur les pin 0,1,2,3,7 (voir annexe pour le code). Ainsi la carte Arduino récupère les valeurs des quatre canaux auxiliaires.

L'auxiliaire 1 qui correspond à la sécurité moteur et est aussi branchée à la carte de vol pour qu'elle puisse continuer à désarmer et armer les moteurs correctement.

## ❖ TRAITEMENT ET BRANCHEMENT DES AJOUTS



Le traitement des commandes des canaux auxiliaires reçues est fait par l'Arduino. La carte contrôle le buzzer, les phares et la bande de leds. (Voir code en annexe)

Le buzzer et les phares sont alimentés au travers de la carte tandis que la bande de leds est alimentée directement depuis la plaque de distribution d'énergie, et donc de la batterie car elle nécessite trop de courant pour fonctionner.

L'Arduino est, quant à elle, alimentée depuis la carte de vol.

## IV. PROBLÈMES RENCONTRÉS ET SOLUTIONS

---

Tout au long du projet, nous avons rencontrés un certain nombre de problèmes. Des problèmes d'électronique, de code, d'ergonomie...

### a) PROBLÈMES D'ÉLECTRONIQUE :

Après le montage du drone (montage du châssis, intégration des moteurs, ESC, contrôleur de vol...) le premier problème auquel nous nous sommes retrouvées confrontées a été la configuration des ESC. En effet, par défaut, quand nous les avons branchés, les moteurs tournaient tous dans le même sens, rendant le vol impossible. La solution la plus simple aurait été de changer leur sens de rotation en reconfigurant les esc grâce à BLHeli (logiciel permettant de gérer les ESC c'est à dire les calibrer, les mettre à jour ou encore régler le sens de rotation des hélices). Mais la communication avec les ESC depuis la carte de vol était impossible (aucun pin visible) car nos modèles d'ESC n'étaient pas programmables. Il a donc fallu inverser les soudures pour inverser le sens de rotation (inversement des + et -).

Lors de la configuration de la carte de vol avec CleanFlight (un firmware embarqué possédant une interface pc permettant qui permet de configurer la carte de vol et gère la stabilité et la réception des commandes), après avoir réussi à armer les moteurs (armer les moteurs leur permet de pouvoir se mettre en fonctionnement, quand ils sont désarmés les moteurs sont bloqués), il y a eu un bug, un des ESC a pris feu. Cette combustion peut s'expliquer par le fait que l'ESC était déjà défectueux, que les soudures faites se sont touchées et l'on court-circuité ou encore que les données envoyées à l'ESC par la carte l'aient fait brûler, mais impossible de vérifier après coup. Nous avons donc été obligé de le remplacer car celui-ci était complètement brûlé.

Pour l'assemblage, nous avons aussi certains pins qui devaient être raccordés à plusieurs éléments comme par exemple le VCC de la carte Arduino qui devait être branché à la fois au buzzer et aux deux phares. On a de ce fait, "construit" de nouveaux fils à plusieurs sorties en soudant des fils entre eux.

De plus, au début, lorsque les leds de position étaient alimentées par l'Arduino, si la luminosité était un peu trop élevée, la carte chauffait beaucoup trop car les leds consommaient trop de de courant. Il a donc fallu les alimenter directement avec la batterie.

### b) PROBLÈMES D'INFORMATIQUE

Le majeur problème niveau informatique a été de comprendre et d'apprendre à manipuler le logiciel CleanFlight, primordial à la configuration du drone. Nous avons dû faire face à beaucoup de fonctions compliquées à comprendre mais indispensables. En outre il a aussi fallu déchiffrer le code (disponible en open source du Github) de CleanFlight pour pouvoir récupérer et utiliser des valeurs envoyées

depuis la télécommande, aussi précises que celles de CleanFlight afin de coder le comportement des leds, phares, buzzer, et armements des moteurs sur l'Arduino.

De plus, pour le code concernant les leds, nous utilisons au début une fonction `-- delay()` -- qui arrête toute la loop Arduino et donc arrête l'acquisition d'autres données car l'acquisition des données est faite par des `Interrupt()`. Il a donc fallu utiliser une méthode avec laquelle le programme ne s'arrête pas. Nous avons donc remplacé cette fonction par des boucles temporelles et l'utilisation d'un chronomètre avec l'Arduino avec la fonction `millis()`.

Dernier problème, la fonction `show()` de Adafruit pour la bande de Led, dont l'appel permet d'en quelques sorte "mettre" la couleur demandée avec la fonction `setPixelColor()`, rendait notre acquisition de valeur totalement fausse. On a donc arrêté l'acquisition le temps de cette fonction puis redémarrer (voir code en annexe)

### c) PROBLÈMES D'ERGONOMIE

Devant être optimisé pour la course, la taille du drone est assez réduite. Il a donc fallu organiser au mieux le peu d'espace que l'on avait sur le châssis pour pouvoir tout y rentrer : les différentes cartes (de vol, de distribution de l'énergie, Arduino), les différents éléments (la caméra et sa batterie, le buzzer, les leds) et tous les fils.

Comme elles devaient être assez proche sans pour autant se gêner l'une l'autre, on a dû placer la carte de vol juste au-dessus de la plaque de distribution d'énergie en les séparant avec des entretoises et on les a fixés au châssis.

L'Arduino pro-micro et le buzzer ont dû être fixés au pistolet à colle sur une plaque elle-même fixée à l'aide de straps à la partie inférieure du châssis.

Les phares ont dû être fixés avec beaucoup de petits bouts de ruban adhésif car leur forme faisait qu'ils ne tenaient pas avec simplement de l'adhésif double face.

Nous avons dû placer la batterie au-dessus du châssis pour optimiser la place à l'intérieur sans pour autant qu'elle gêne au vol.

La quantité de fils étant très importantes, on a dû les attacher avec des straps dans le châssis et même ajouter de l'adhésif sur certains pour éviter qu'il ne se prennent dans les pâles de hélices quand celles-ci tournent.

Il a également fallu coller avec de l'adhésif double face la caméra et sa batterie pour qu'elles soient facilement enlevées et remises en place, avec ladite batterie en dessous du drone pour qu'elle soit facilement accessible et la caméra sur le devant du drone pour avoir une bonne visibilité.

# V. CONCLUSION

---

## a) SYNTHÈSE

En somme, ce projet a été très enrichissant et un des plus intéressant que l'on ait eu à faire. Il nous a permis d'entreprendre une véritable démarche de recherche et de documentation sur des problèmes concrets et nous as appris à gérer les contre-temps inhérents à tout projets. Nous avons pu réaliser la quasi-totalité de nos objectifs et réaliser un drone FPV fonctionnel tel qu'on le voulait.



## b) PERSPECTIVES

Avec plus de temps, on pourrait ajouter la led batterie que l'on voulait placer dans le champ de la caméra FPV. On pourrait également changer les hélices car une de nos pales a été accidentée. Il serait aussi intéressant de configurer les PIDs pour de l'optimisation de vol. Autant d'optimisation qui pourront être réalisées par d'autres étudiants après nous.

## VI. ANNEXE

---

**Code :** [https://github.com/Rizlensee/Droduino/blob/master/code/led\\_drone.ino](https://github.com/Rizlensee/Droduino/blob/master/code/led_drone.ino)