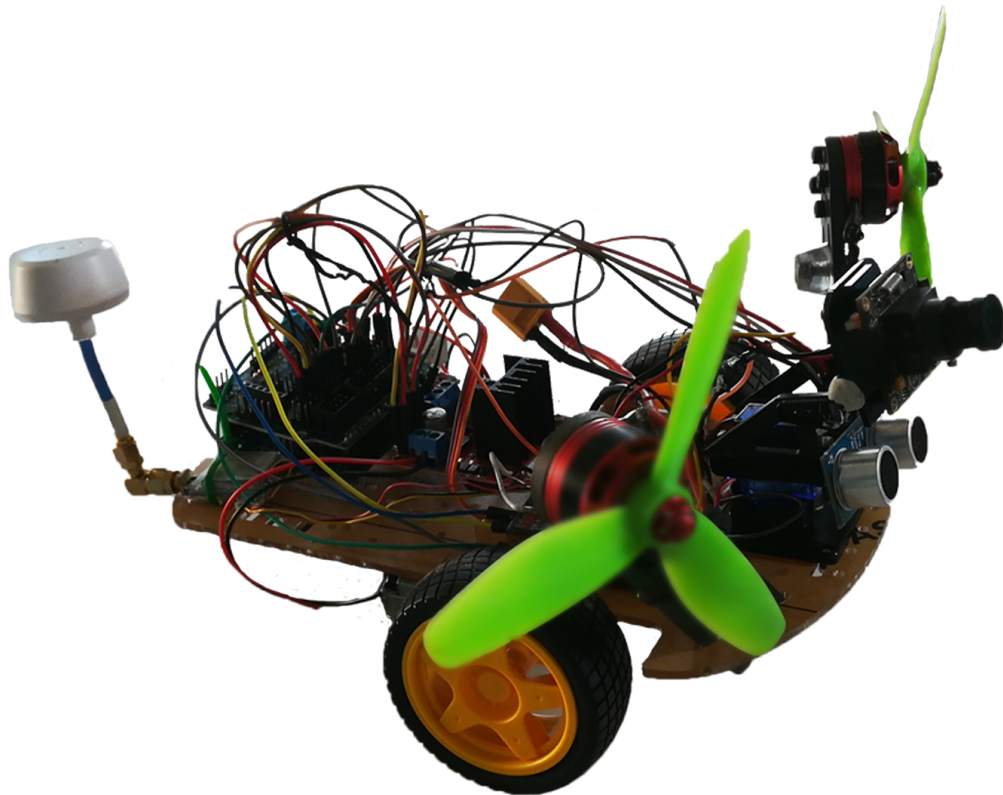


---

# Rapport de conception et technique de la “Flying cable car”

Equipe :  
Nicolas - Benjamin - Mathias - Kévin - Matthieu



# Table des matières

<b>1 - Cheminement création de la “Flying cable car”</b>	<b>3</b>
1.1 - Introduction	3
1.2 - Descriptif de l'équipement	3
1.3 - idées secondaires	4
1.4 - Problèmes rencontrés	5
1.5 - Bilan / présentation du produit final	6
1.6 - Conclusion / limite du projet / améliorations possibles	7
<b>2 - Documentation technique</b>	<b>8</b>
2.1 - Châssis motorisé:	8
2.2 - Caméra / casque FP0V	9
2.3 - Rotors	10
2.4 - Télécommande	13
2.5 - Vue d'ensemble	15

# 1 - Cheminement création de la “Flying cable car”

## 1.1 - Introduction

À l'issue de notre formation en IOT nous avons pour projet de créer notre propre objet connecté. Pour ce faire nous avons été initiés à l'utilisation de différentes cartes électroniques ainsi que différents types de protocoles de communications.

## 1.2 - Descriptif de l'équipement

Nous avons eu la possibilité de créer notre appareil à partir de plusieurs éléments. Nous avons récupéré la structure d'une voiture équipée d'une carte Arduino ainsi que des pièces glanées sur un ancien drone. Ainsi notre projet c'est orienté vers une voiture autonome aux diverses fonctionnalités :

Dans un premier temps nous avons assemblé la structure de la voiture, celle-ci comprend deux roues motorisées et une roue libre pour assurer la stabilité de la structure. Une fois la structure assemblée, nous avons testé puis ajouté la carte arduino et le système d'alimentation au véhicule.

Après quelques rapides tests, nous avons connecté la carte Arduino et les moteurs via une carte de gestion d'alimentation.

Notre kit de montage nous apportait deux autres éléments que nous avons décidé d'exploiter : un capteur de proximité ultrasonique et un servomoteur.

Nos esprits éveillés à l'IOT domestique, nous avons presque immédiatement pensé puis mis en place un système de détection d'obstacle.

Ainsi en fixant le capteur sur le bras du servomoteur, nous avons ajouté un mouvement de balayage afin de sécuriser un angle intéressant sur l'avant de véhicule.

Il s'en suit l'élaboration, via quelques tests, d'un petit programme pour traiter les informations et fournir une réaction appropriée. Ainsi notre petit robot peut maintenant avancer et, lorsqu'un obstacle est détecté sur la route, il exécute une habile manœuvre d'esquive avant de reprendre son chemin.

Enfin nous avons exploité le module de télécommande du drone pour pouvoir contrôler les différents moteurs du robot.

## 1.3 - Idées secondaires

Une fois notre premier prototype en libre circulation dans notre espace de test, nous avons reçu de nouvelles pièces que nous avons récupérées dû au démantèlement d'un drone de racing. Grâce à l'acquisition de ces pièces nous avons décidé de pousser plus loin le développement de notre voiture intelligente.

Une des premières pistes que nous avons étudié était de pouvoir faire voler notre petit robot. Cependant, il nous manquait pas mal de temps pour analyser la carte du drone afin d'exploiter l'accéléromètre intégré. De plus, pour coder un mode de vol viable et sûr pour l'équipement il nous aurait fallu beaucoup plus de temps.

La solution que nous avons décidée d'implémenter est alors un "turbo boost" en exploitant la puissance déployée par les deux pales du drone. Pour ce faire il nous a fallu ajouter un deuxième circuit d'alimentation adapté à la puissance de ces nouveaux moteurs. Ces moteurs étant équipés l'électro-interrupteur, nous avons pu, sans trop de complications, ajouter et connecter les contrôleurs à notre carte arduino.

Après un peu de casse et de nombreux tests afin d'étalonner nos équipements, nous sommes parvenus à un sous système de ventilateurs performant. Le résultat est un système de propulsion efficace mais malheureusement bloqué par le mode de fonctionnements des roues. En effet les roues sont motrices et donc ne sont pas "libre". De ce fait la vitesse max des roues du support est fixe malgré la force de traction proposées par les rotors.

Enfin nous nous sommes occupé du capteur vidéo anciennement utilisé pour le pilotage du drone en FPV. Fonctionnant avec la puissance des batteries du drone et un couple d'un émetteur / récepteur, nous avons donc connecté, assez simplement, l'émetteur à notre circuit et nous bénéficions donc d'un mode de conduite à la "première personne". Cet ajout nous a permis de pouvoir contrôler le véhicule sans avoir directement la vue sur lui puisque les ondes provenant du casque et de la télécommande ont une bonne portée de fonctionnement.

## 1.4 - Problèmes rencontrés

- Le plus compliqué dans ce projet fut de comprendre le fonctionnement de la télécommande. En effet, la télécommande envoie les informations par ondes radio or nous n'avions pas encore d'expérience dans ce domaine. De plus, la télécommande disponible dispose d'une multitude d'options ce qui est venu complexifier sa configuration et en expérimentant avec celle-ci. Nous avons toutefois réussi à la faire fonctionner en consultant sur internet sa documentation. Pour complexifier le tout, nous nous sommes heurté à un problème de *bruit* sur nos communications entre la télécommande et son récepteur. Il nous a donc fallu prendre en compte les petites fluctuations du signal alors qu'aucune commande n'est envoyée. Nous avons donc mis en place un système de seuil à atteindre pour que la carte arduino puisse éviter les mouvements/actions parasites.
- Nous avons passé plus de temps que prévu pour faire fonctionner les rotors des hélices. Après études des schémas électriques trouvés sur internet, nous avons essayé de connecter les rotors à une carte arduino dans le but de lui envoyer les informations de fonctionnement. C'est sur ce point que nous avons passé le plus de temps. En fonction des valeurs envoyées le rotor était censé tourner plus ou moins vite. Alors que le rotor ne fonctionnait pas nous avons essayé de lui envoyer une valeur très grande. C'est alors que le rotor s'est mis en fonctionnement très rapidement et que celui-ci nous a échappé engendrant quelques dégâts dans le montage. Nous avons dès lors décidé de faire les tests de mise en marche sans les pales des rotors.
- Un autre problème, mineur celui-ci, est le problème de batterie. Pour le fonctionnement nous devons déjà composer avec quatre piles pour l'alimentation de l'arduino, du moteur et du servo de la tourelle, mais aussi avec une batterie 11,1 V qui s'occupait d'alimenter les rotors, la caméra ainsi que l'émetteur. En plus de prendre de la place sur le montage, ces batteries apportent des contraintes d'autonomie, étant donnée la consommation plutôt élevée de certains montages. De plus les batteries utilisées étant un peu vieilles ne distribuent plus autant de tension et ont une durée de fonctionnement réduite.

## 1.5 - Bilan / présentation du produit final

Le drone que nous vous avons présenté au travers des parties précédentes de ce rapport a fini par voir. Il fonctionne très bien et peut réaliser plusieurs actions que voici :

- Le véhicule se mouvoir sur le sol et peut se déplacer en autonomie. Cette navigation est possible grâce au système de détection de proximité couplé à notre intelligence artificielle d'exploration.
- La télécommande peut contrôler entièrement l'appareil. Cinq boutons sont utilisables. Les actions possibles sont la marche avant, la marche arrière, tourner à gauche et droite, détecter les objets à proximité sur la gauche et la droite. De plus, il est possible de contrôler la puissance des rotors, de l'allumage de ceux-ci à puissance maximale. À travers la télécommande nous pouvons aussi diriger l'orientation de la caméra et l'activation/désactivation du mode de déplacement autonome.
- Le module caméra fonctionne parfaitement et retransmet en direct une vue à la "première personne" depuis le véhicule, qui permet de contrôler le drone sans le voir directement.
- Les ventilateurs apportent un vrai boost, mais il serait certainement bien plus efficace de mettre à jour le système de motorisation pour profiter pleinement de la puissance qu'ils dégagent.
- Cependant, le véhicule présente une autonomie malgré tout limitée. La transition vidéo est très énergivore et supportant mal le "bruit" produit par la consommation d'énergie des moteurs des ventilateurs. De plus, tous les moteurs consomment beaucoup à l'utilisation. Un système de recharge régulier est de ce fait à prévoir pour ce genre de véhicule.

(Photo de la vision depuis le casque FPV)



## 1.6 - Conclusion / limite du projet / améliorations possibles

Lors de ce projet nous avons atteint notre objectif principal qui était d'avoir une voiture télécommandée. La télécommande radio, les hélices et le retour caméra sont des ajouts dont nous sommes fier. La vitesse de la voiture nous à un peu déçu, nous pensions pouvoir compenser ce manque par les hélices, mais le boost qu'elles procurent est limité par la motorisation principale. Le contrôle de la voiture par onde radio permet de la commander depuis de longue distance et l'ajout du système de vidéo permet de la suivre même si on la perd de vue. Nous sommes très content du rendu final et celui-ci est viable.

Nous avons été limité par le kit de la voiture, la planche de la voiture manque de place et les moteurs sont peu puissant. De plus le socle à piles est un peu trop grand pour des piles AAA, il nous est arrivé d'en perdre.

Si nous avions eu plus de temps nous aurions aimé pouvoir ajouter certaines fonctionnalités à notre voiture. En premier lieu et comme dis précédemment, nous aurions voulu lui ajouter un mode de vol, aidé par les pièces récupérées sur le drone. En second lieu, si nous avions eu le matériel nous aurions aimé changer les roues pour un modèle permettant la roue libre, ou simplement plus rapide afin d'utiliser pleinement la puissance offerte par les rotors.

Nous aurions par exemple pu utiliser un autre châssis appartenant à un membre de l'équipe qui est une voiture radiocommandée. En utilisant la radio déjà en place et en la branchant sur une carte arduino, nous aurions pu ajouter une mode autonome ainsi que la gestion de la vidéo tout en gardant le contrôle sur la direction de l'engin.

Si nous avions suivi cette voie, l'utilisation des rotors n'aurait pas été la même, le châssis étant beaucoup rapide. Nous aurions certainement chercher un moyen de faire voler le châssis sur de très courte distance afin par exemple d'enjamber des obstacles.



## 2 - Documentation technique

L'un des éléments intéressants dans la création de ce drone vient du fait que nous avons utilisé des éléments récupérés à d'autres projets/drones. Nous avons donc dû, sans toujours avoir de documentation constructeur à disposition, trouver et utiliser des périphériques pour arriver à la création de cet élément.

Malgré l'intérêt que cela ajoute au projet, il faut noter que ça crée aussi des problèmes. En effet, les éléments venant de différents projets, il n'est pas forcément simple de les faire cohabiter dans un espace restreint, ici le châssis fourni pour le projet.

### 2.1 - Châssis motorisé:

Dans le cadre de ce cours, notre intervenant a participé à notre projet final en fournissant un kit de démarrage Arduino permettant de créer un drone roulant fonctionnant à l'aide de 4 piles AAA.

Ce [kit](#) est arrivé sans documentation ni référence, nous avons donc utilisé de notre navigateur favori afin de l'identifier et donc de chercher de la documentation. Nous avons trouvé le [document suivant](#) qui donnait des explications assez claires sur la marche à suivre pour le montage et le câblage des éléments du kit.

Il fournissait de plus un code à déployer sur la carte Arduino afin de faire fonctionner le drone de manière autonome.





Les principaux éléments permettant de recréer le châssis sont donc disponible dans la documentation liée.

## 2.2 - Caméra / casque FPV

A ce niveau là nous avons donc un drone roulant capable de se déplacer mais il n'est pas possible de le suivre à distance. C'est à ce problème que devait répondre le système suivant.

En effet grâce au casque Eachine l'idée était d'avoir une vue à la première personne de l'environnement dans lequel se trouve le drone. Cela permet d'avoir une idée de l'endroit ou il se trouve, pour aller le récupérer si il est parti de manière autonome.

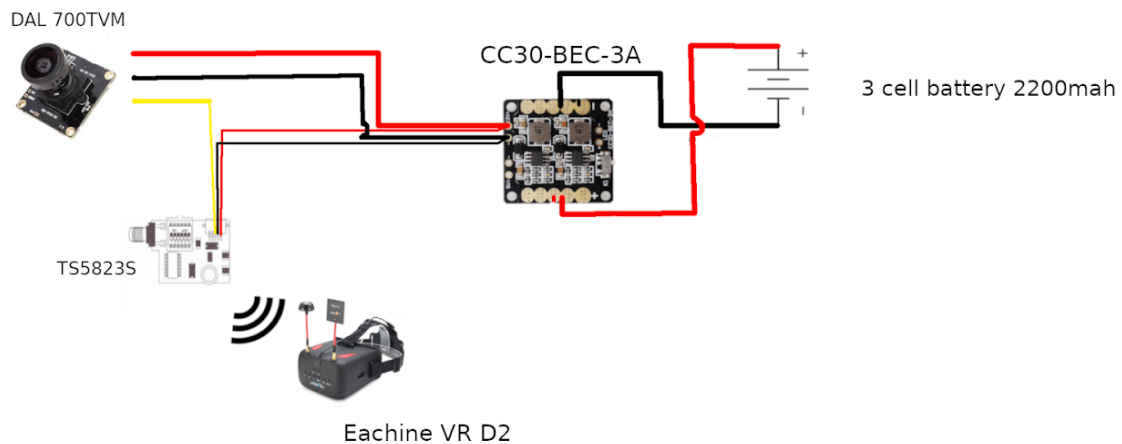
Le casque utilisé est un Eachine VR D2 Pro 5 Pouces 800\*480. Il permet la récupération d'images via une fréquence de 5,8 Ghz.

Il fallait donc une caméra pour envoyer les images et celle ci était lié au drone volant que nous avons a disposition.

La caméra est une DAL 700 TVL, caméra HD grand angle avec ouverture à  $\frac{1}{4}$ . Celle ci, couplé à un émetteur Micro VTX FPV nous a permis d'envoyer les images de la caméra au casque.



Voici une idée du montage électrique mis en place pour obtenir ce rendu:



Dans celui ci on note les éléments présentés ci dessus mais aussi l'apparition d'une batterie 3 cellules 2200mah qui va permettre l'alimentation de la caméra mais aussi du module émetteur, et ce par l'intermédiaire d'une carte d'alimentation CC30-BEC-3A.

Celle ci permet via une source d'énergie d'avoir des sortie lissées en 12v et 5v. Cela signifie qu'à terme nous aurons pu baser notre système uniquement sur cette carte d'alimentation afin de n'avoir qu'une batterie à embarquer.

## 2.3 - Rotors

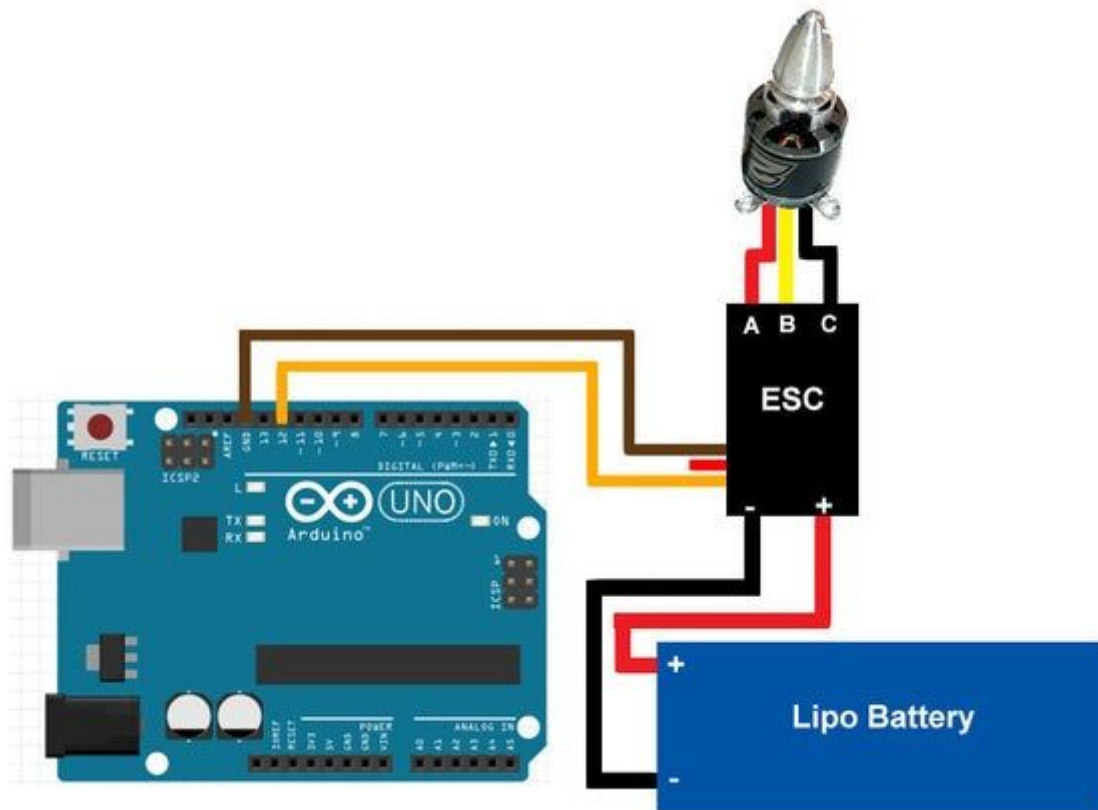
Composants :

- Arduino Uno
- Batterie Lipo 11.1V et de capacité de 1500mAh à 2200mAh
- Moteur sans balais (ESC)
- Rotor

Pour développer le rotor nous avons commencé par développer un schéma basique afin de valider son fonctionnement. Le rotor contrôlé par le moteur sans balais (ESC), qui est la partie intelligente de l'appareil. Le moteur est alimenté par ses deux gros cable (rouge et noir) à l'aide d'une batterie externe de type Lipo. Pour ce qui est de contrôler et modifier la

vitesse de rotation du rotor nous avons développé un programme arduino qui nécessite de connecter le moteur sans balais sur le “ground” ainsi que sur un port numérique.

Montage associé à notre conception :



Code associé à la variation de la vitesse de rotation du rotor dans notre cas :

```
projet_rotor $
#include <Servo.h>
Servo esc_signal;
Servo esc_signal2;

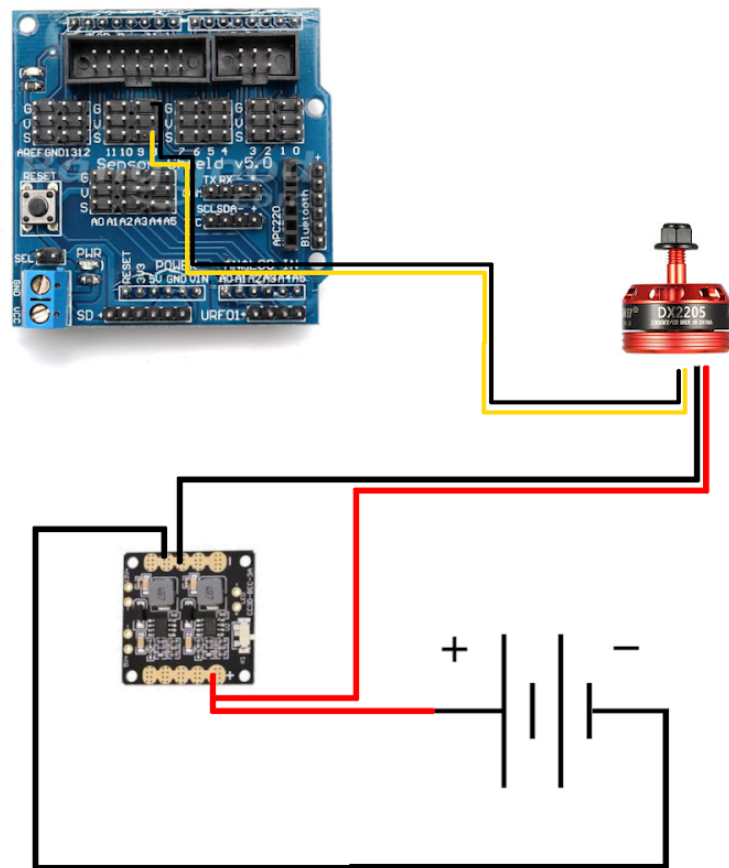
void setup()
{
  esc_signal.attach(12); //Specify here the pin number on which the signal pin of ESC is connected.
  esc_signal2.attach(11); //Specify here the pin number on which the signal pin of ESC is connected.
  delay(5000);
  esc_signal.writeMicroseconds(500); //ESC arm command. ESCs won't start unless input speed is less during initialization.
  esc_signal2.writeMicroseconds(500); //ESC arm command. ESCs won't start unless input speed is less during initialization.
  delay(2000); //ESC initialization delay.
  Serial.begin(9600);
}

int speed2 = 1400;
int pas = 20;

void loop()
{
  if (speed2 < 1300 ) {
    pas = 20;
  }
  if (speed2 > 2000 ) {
    pas = -20;
  }

  speed2 = speed2 + pas;
  Serial.print(speed2);
  Serial.print('-');
  esc_signal.writeMicroseconds(speed2);
  esc_signal2.writeMicroseconds(speed2);
  delay(200);
}
```

Une fois le fonctionnement de ce prototype validé, nous avons inséré les rotors\* dans le montage de la manière suivante:



Il nous a fallu ensuite trouver un moyen de faire tenir ces rotors sur le châssis. Les rotors étaient fixés sur un morceau de carbone, vestige du drone sur lequel ils étaient avant. En limant un peu ces morceaux de carbone nous avons pu les faire rentrer dans les fentes à l'avant. Une vis/écrou en dessous nous permet de s'assurer que les hélices ne vont pas s'échapper de leur place.

Cela nous permet donc de faire fonctionner les rotors en les contrôlant via la carte arduino. Mais ce n'est pas suffisant. Il faudrait de plus contrôler finement la vitesse de rotation de ceux-ci.

## 2.4 - Télécommande

Composants :

- Radio AT9
- Récepteur R9D Radiolink

Nous avons, pour ce projet, utilisé une télécommande mise à disposition qui était la Radiolink AT9.

Afin de profiter un maximum des possibilités offertes par la télécommande, nous avons dû créer une connexion avec le receveur R9D.

Dans le mode normal, on peut retrouver sur les ports numérotés en blanc sur le récepteur et vérifier à quoi les canaux correspondent via l'écran de la télécommande.



Cette télécommande nous a permis de prendre le contrôle du drone mais aussi des hélices. Pour ce faire nous avons utilisé les commandes suivantes :



Après cela, nous sommes donc normalement capable de contrôler la direction/vitesse du drone, mais aussi de lui indiquer de passer en mode automatique ou manuel (exploration via script ou récupération des commandes utilisateur), déplacer la tourelle avant sur laquelle nous avons fixé le capteur de distance et la caméra du système de vue à la première personne.

Il est aussi possible d'utiliser les rotors et la puissance sera défini en fonction de la position du potentiomètre.

Pour faire cela, nous avons récupéré les signaux sur le récepteur R9D. Pour récupérer ces signaux, nous devons alimenter le module, évidemment, et connecter les ports qui nous intéressent au port analogique de la carte arduino.

Dans un premier temps, nous avons juste afficher les valeurs récupérées dans le moniteur série, afin de nous faire une meilleure idée des données avec lesquelles nous allons travailler.

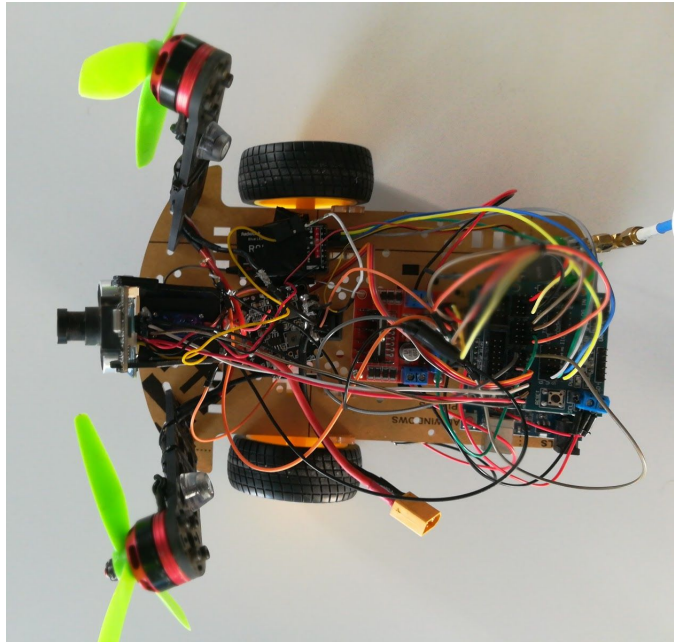
Nous avons donc travaillé à créer une fonction permettant de récupérer les valeurs et de les normaliser en fonction de l'utilisation qui leur seraient réservé.

Une bibliothèque nous a été particulièrement utile lors de la récupération des données c'est [EnableInterrupt](#). Celle-ci permet de détecter correctement les informations présentes sur les ports analogiques et d'avoir une représentation la plus fidèle possible, et avec le moins de latence de l'état des boutons utilisés sur la manette.



## 2.5 - Vue d'ensemble

Une fois toutes ces parties assemblées et fixées, le drone est prêt à fonctionner.



Vous pouvez retrouver un exemple de fonctionnement ici:

<https://drive.google.com/file/d/1lw5mBOZeDbYDxPk2abDU7plXrlhAVucw/view?usp=sharing>

Et un exemple de la vue à l'intérieur du casque de vue à la première personne:

<https://drive.google.com/open?id=1pQNiOjSqFqwFYYdFVP3gUdPHs9RR6Fzp>

Voici le schéma électrique complet du drone:

