

# DRONE VOLANT

Projet IMA3/IMA4

## Résumé

Ce projet basé sur deux ans, a pour but de réaliser les parties régulation et communication d'un drone.

## Table des matières

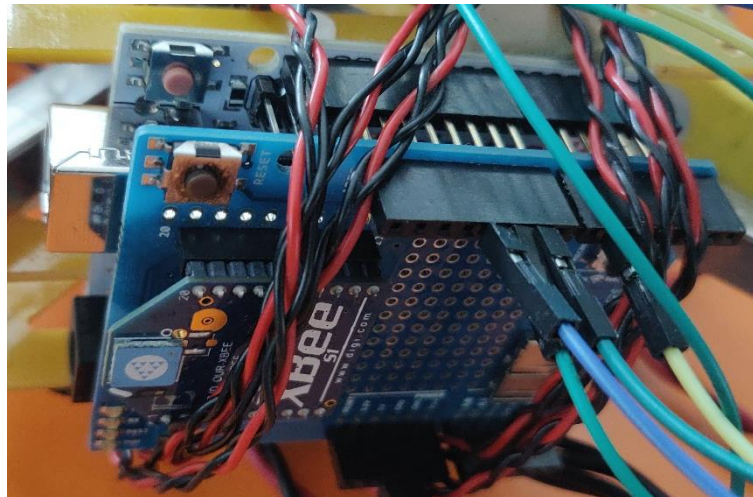
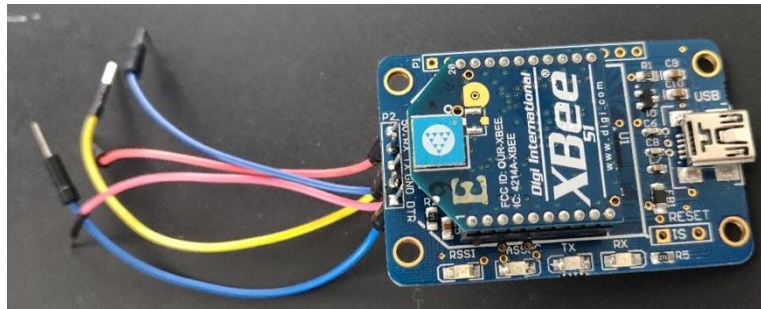
Introduction.....	2
1 – Gestion de la commande du drone .....	3
1.1 – Emission .....	3
1.1.1 – Initialisation de la XBee .....	4
1.1.2 – Création de trames .....	4
1.2 – Réception .....	5
1.3 – Commande moteur.....	5
2 – Contrôle de vol .....	6
2.1 – Choix de la carte de contrôle de vol et du logiciel.....	6
2.2 – Calibration de l'Ardupilot .....	6
2.2.1 – Boussole .....	6
2.2.2 - Accéléromètres.....	7
2.3 – Alimentation du complexe ESC/Ardupilot/Arduino .....	7
3 – Problèmes rencontrés .....	7
Conclusion .....	8

## Introduction

Notre groupe se compose de Maxime CLAUDEL, Lukas FAUCHOIS, Evan GURY et Richard Simonin. Nous avons dans un premier temps décidé de réaliser notre projet autour d'un drone servant à la livraison de petit colis, cependant dans un souci concernant la matériel mis à notre disposition et de budget, nous avons finalement opté pour un drone "loisir". Ce dernier sera destiné à un public amateur, dans un but d'apprentissage et/ou d'amusement et sera ainsi plus en adéquation avec nos contraintes et compétences.

## 1 – Gestion de la commande du drone

Dans cette partie, nous avons choisi de tout faire nous-mêmes. La création de la télécommande ainsi que la réception des trames. Nous avons séparé cette partie en deux sous-parties : l'émission et la réception. Nous avons choisi d'utiliser des modules XBee pour la communication.



### 1.1 – Emission

Afin de pouvoir diriger le drone, nous devons communiquer au contrôleur de vol les informations relatives aux joysticks. Afin de réaliser la télécommande, nous avons besoin de deux joysticks, une carte Arduino et un module XBee.

Nous avons dû découvrir XBee, qui nous était inconnue et ensuite programmer une carte Arduino afin d'envoyer les informations. Le logiciel nous a permis de comprendre le fonctionnement et de programmer les XBee

## 1.1.1 – Initialisation de la XBee

```

// Xbee configuration
void setDestination() {
  Serial.print("+++");
  char thisByte = 0;
  while (thisByte != '\r') { // wait for xbee response
    if (Serial.available() > 0) {
      thisByte = Serial.read();
    }
  }
  Serial.print("ATRE\r");
  Serial.print("ATDL0\r"); // sent to xbee 0
  Serial.print("ATMY1\r"); // this xbee is 1
  Serial.print("ATID1111\r");
  Serial.print("ATCN\r");
}

```

Cette fonction permet l'initialisation la XBee. L'envoi de « +++ » permet de passer la XBEE en mode configuration et ensuite la paramétrer pour créer notre réseau

Dans un premier temps nous avons pensé à envoyer un caractère afin de simplifier l'envoi d'informations. Mais nous nous sommes rendu compte que sur un joystick entre 0 et 1024 nous perdions trop d'informations et la commande se trouvait être beaucoup trop imprécise.

Nous avons donc dû réfléchir à une nouvelle solution. Une idée nous est venue, créer une trame composée de chaque chiffre composant le nombre renvoyé par le joystick. Ainsi nous avons créé une fonction qui génère une trame.

## 1.1.2 – Création de trames

```

char* convertir_vers_char(int val){
  char* buf=malloc(sizeof(char)*6);
  buf[3]=val%10 +48;
  buf[2]=(val%100-val%10)/10+48;
  buf[1]=(val%1000-val%100)/100+48; //memcpy
  buf[0]=(val-val%1000)/1000+48;
  buf[4]=0;
  buf[5]=0;
  return buf;
}

```

Nous calculons à l'aide de modulus les différents chiffres indiquant la position du potentiomètre du joystick. Nous laissons le dernier octet à 0 afin d'envoyer un caractère de fin de chaîne. L'octet 5 permet de choisir l'information à envoyer. Nous envoyons 4 valeurs, 2 pour chaque joystick, et il y a 2 potentiomètres par joystick.

Afin d'envoyer nos trames, nous devons dans un premier temps, lire les valeurs des potentiomètres. Nous devons ensuite convertir les valeurs en une trame lisible par le récepteur. Nous personnalisons un octet par trame, le 5<sup>ème</sup> afin de savoir à quel potentiomètre correspond la valeur. Et nous envoyons les trames dans l'ordre.

### 1.2 – Réception

Nous recevons des trames en continu, et à l'aide de la valeur du 5<sup>ème</sup> octet, nous sommes capables de définir à quelles informations correspond la trame. Nous identifions la valeur correspondante à l'aide de la fonction *Serial.readbytes(buffer,taille)*, nous connaissons ainsi les trames réceptionnées et leur taille.

Nous sommes ainsi capables d'identifier les valeurs reçues et de les attribuer à chaque potentiomètre.

### 1.3 – Commande moteur

Dans un premier temps nous avons utilisé une commande PWM pour envoyer les informations à notre contrôleur de vol afin de commander nos ESC, mais nous nous sommes rendu compte que les ESC se contrôlent à l'aide de la bibliothèque *Servo.h*. Nous convertissons notre valeur comprise entre 0 et 1023 en une valeur entre 0 et 180 afin de pouvoir utiliser *Servo.Write(val)*. Lors de nos différents tests nous avons remarqué que la commande des ESC était entre 35 et 150, c'est pourquoi nous utilisons un *map(tram,0,1023,35,150)* afin d'obtenir les valeurs correctes.

## 2 – Contrôle de vol

### 2.1 – Choix de la carte de contrôle de vol et du logiciel

Nous cherchions donc une carte facilement connectable avec un Arduino Uno/Micro. Nous avons opté pour l'Ardupilot 2.8. Il comprend tout le nécessaire pour diriger un drone quadcopter, boussole, centrale gyroskopique, accéléromètres, interface de radiocommande.



Pour le logiciel de contrôle de vol, L'objectif était d'en trouver un compatible avec l'Ardupilot 2.8 et avec suffisamment de documentation et tutoriels d'utilisation. Les deux logiciels qui sont ressortis le plus dans nos recherches sont Mission Planner et APM Planner 2. Ces deux logiciels sont très similaires au niveau de l'interface et des options de calibration de la carte de vol à laquelle ils sont connectés.

### 2.2 – Calibration de l'Ardupilot

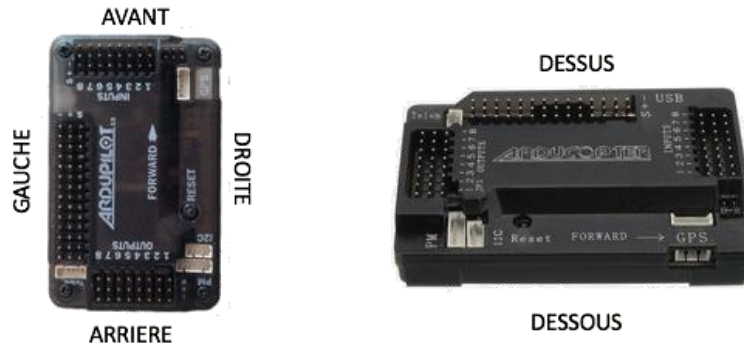
#### 2.2.1 – Boussole

La calibration de la boussole s'effectue en bougeant l'Ardupilot selon tous ces axes pendant une période de temps donnée jusqu'à ce que les informations nécessaires soient récoltées.



### 2.2.2 - Accéléromètres

La calibration des accéléromètres s'effectue en positionnant l'Ardupilot dans des positions définies (sur le côté gauche, sur la face avant, etc.)



### 2.3 – Alimentation du complexe ESC/Ardupilot/Arduino

En se documentant sur le site servant de manuel d'utilisation d'Ardupilot et des logiciels Mission Planner et APM Planner 2 (<http://ardupilot.org/copter/index.html>), nous avons trouvé que l'Ardupilot pouvait être alimenté via les ESC. Une batterie LiPo alimente les ESC qui elles-mêmes alimentent l'Ardupilot via les broches Output. L'Ardupilot alimente ensuite l'Arduino qui réceptionne les trames reçues par le récepteur Xbee. Sur les sorties (Output) de l'Ardupilot sont aussi branchés les fils permettant la réception du signal envoyé aux moteurs. Ainsi, tous les composants nécessaires au drone sont alimentés.

## 3 – Problèmes rencontrés

Lors de la calibration des ESC, nous avons toujours eu un peu de mal à cerner le protocole à suivre pour armer le drone le plus rapidement possible. Les premiers essais de vol venaient souvent après plusieurs minutes passées à essayer d'armer le drone correctement.

Concernant le vol en lui-même, lors de nos essais concluants, plusieurs problèmes se sont posés. Etant donné la qualité des joysticks, un petit déplacement de l'un d'eux provoque souvent un gros changement au niveau de la vitesse de rotation des hélices. Ainsi, la commande du drone est très nerveuse et rend le pilotage assez difficile. De plus, le drone a parfois tendance à tourner sur lui-même s'il n'est pas parfaitement calibré au niveau des accéléromètres ou de la boussole. Il faut donc être très pointilleux lors de ces étapes.



### Conclusion

L'objectif de ce projet est la réalisation d'un drone. Ce sujet étant large, nous avons choisi pour cette année, la création d'une télécommande ainsi que l'utilisation d'un contrôleur de vol. Nous avons réussi nos objectifs malgré quelques problèmes rencontrés. Nous sommes capables de faire "voler" le drone à l'aide de notre télécommande.

Lors de la réalisation de notre projet, nous avons eu quelques idées d'améliorations pour l'année prochaine ainsi que des tâches à effectuer. Nous pensons à la réalisation d'un contrôleur de vol ainsi qu'à une amélioration de la commande en modifiant la programmation et quelques composants.