

# Rapport de projet : Dirigeable Publicitaire

*Élèves*

Loïc DELECROIX & Julien JOIGNAUX

*Encadrant*

Xavier REDON

# Sommaire

Remerciement .....	Page 1
Introduction .....	<i>Page 2</i>
Avant le projet .....	<i>Page 3</i>
Conception du dirigeable .....	<i>Page 5</i>
Solution testée : les moteurs d'AR Drone .....	<i>Page 8</i>
Solution finale : le contrôle par ventilateurs .....	<i>Page 11</i>
Conclusion .....	<i>Page 15</i>
Annexes .....	<i>Page 16</i>

# Remerciement

Nous tenons tout d'abord à remercier Monsieur Xavier Redon pour nous avoir accompagnés et encouragés tout au long de notre projet.

Nous remercions également tout le personnel de Polytech pour nous avoir apporté leur soutien dans tous les domaines abordés.

Un grand merci également à nos familles et nos amis pour leur contribution à la réalisation du projet.

Un merci tout particulier à Monsieur Laurent Engels pour sa patience et son investissement lors du tournage de notre vidéo finale.

# Introduction

Dans le cadre de notre 4ème année d'étude en Informatique, Micro-électronique et Automatique à Polytech'Lille, nous avons réalisé un projet, choisi à la fin du semestre 7 et à réaliser tout au long du semestre 8.

Nous avons choisi de prendre le projet n°39 : Le dirigeable publicitaire. Ce projet, proposé par Mr Xavier Redon, nous paraissait très intéressant de part sa variété de thèmes sur lesquels nous pouvions nous pencher durant ce projet (informatique, modélisation, système embarqué, mécanique, ...).

Dans un premier temps, nous détaillerons toutes les démarches antérieures au début du projet.

Pour suivre, nous parlerons de la conception matérielle de notre dirigeable.

Ensuite, nous expliquerons les différents chemins que nous avons suivis, et notamment les problèmes rencontrés.

Enfin, nous parlerons de notre rendu final, expliquerons nos choix vis-à-vis de celui-ci et exposerons les solutions envisageables pour un futur projet.

# Avant le projet

Comme nous l'avons indiqué dans l'introduction, le projet commence réellement lors du semestre 7. Un des points les plus importants du travail est de constituer un cahier des charges. Nous devons également savoir placer le projet dans un contexte précis, définir les objectifs finaux du projet ainsi que prévoir le matériel nécessaire.

## 1) Contexte du projet

Lors de conférences et animations en intérieur, il peut être utile de promouvoir sa marque ou son enseigne, ou tout simplement faire de la publicité pour une cause. Dans ce contexte, il peut être intéressant de concevoir un dirigeable publicitaire capable de voguer librement, tout en évitant les obstacles se présentant sur sa route. Ce serait donc également réalisable dans le hall de Polytech Lille, qui accueille régulièrement des personnes extérieures et organise des conférences.



*Hall de l'école*

## 2) Objectifs du projet

Le but du projet sera de créer un dirigeable publicitaire (structure et logiciel) capable de voguer en autonomie. Celui-ci devra éviter les obstacles présents sur sa route.

Nous devons donc d'abord réaliser la structure tout en prenant en compte le poids maximal que pourra soulever le ballon du dirigeable. Dans le même temps, nous devons concevoir la partie logicielle qui devra assurer le vol en toute autonomie du dirigeable.

## 3) Choix logiciel et matériel

Après plusieurs échanges avec Mr Redon et concertation entre nous, nous avons choisi d'adopter les éléments suivants pour notre dirigeable :

- Le contrôle se fera à l'aide d'un Arduino.
- Le ballon sera un ballon en chloroprène, imperméable 4 jours à l'hélium.
- Une bouteille (de  $7\text{m}^3$ ) d'hélium pour soulever environ 770g.
- Des hélices couplées à des moteurs pour faire avancer le dirigeable.
- Des capteurs ultrasons pour détecter les obstacles présents sur la route du dirigeable.
- Une batterie pour alimenter l'Arduino et les moteurs.

# Conception du dirigeable

## 1) L'enveloppe du dirigeable

Comme indiqué dans la partie précédente, nous devons utiliser un ballon capable de contenir environ  $1 \text{ m}^3$  d'hélium, ce qui correspondait à 2 bouteilles d'hélium pouvant être vendues dans le commerce.

Monsieur Redon nous a fourni une enveloppe de ballon sonde ayant déjà été utilisée. Notre premier test a donc été de vérifier l'étanchéité de ce ballon, tout à d'abord en le gonflant à l'air, puis par la suite en le remplissant à l'hélium (en effet, l'hélium est plus fin que l'air, ce qui implique que les ballons ne peuvent le contenir pour une durée aussi longue).

Malheureusement, ce ballon s'est très vite dégonflé, et n'était donc pas utilisable pour notre projet. C'est pourquoi nous avons commandé un autre ballon, d'une contenance de  $1.2\text{m}^3$ , qui s'est révélé être plus imperméable à l'hélium: une durée de 4 jours, comme indiqué sur le site du vendeur.

Une fois ce ballon reçu, nous avons pu faire nos tests. Il fallait tout d'abord réussir à le gonfler, sans dépasser le volume maximum que celui-ci pouvait contenir, tout en ayant un maximum d'efficacité. De plus, le détendeur à notre disposition ne permettait pas de connaître le volume injecté dans le ballon.



*Ballon original*



*Ballon final*

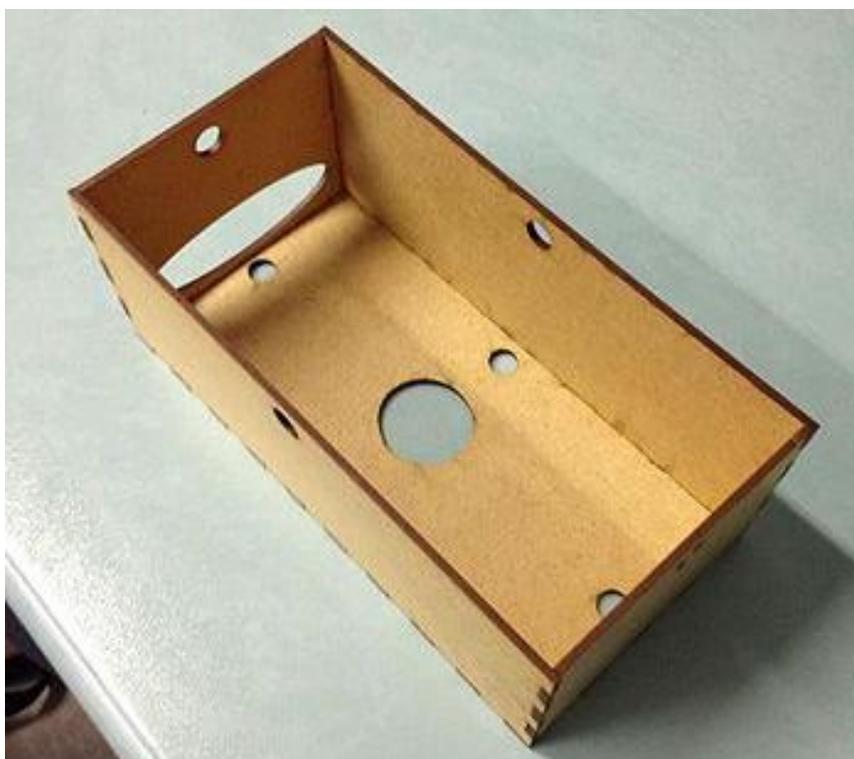
Nous sommes donc partis sur une base de  $1\text{m}^3$ , pour ne pas risquer de faire exploser le ballon. Ainsi, en considérant le ballon comme une sphère parfaite, nous devons avoir un rayon de 62cm, et donc une circonférence de 3.90m pour avoir un ballon rempli d' $1 \text{ m}^3$  d'hélium.

## 2) Conception de la nacelle

Nous avons également dû concevoir une nacelle. Celle-ci devait être capable de contenir l'ensemble des éléments permettant de contrôler le dirigeable : Arduino, batterie, circuits de contrôle, capteurs ultrasons ...

Notre première idée était de concevoir cette nacelle à l'aide de l'imprimante 3D disponible à l'école. Cependant, le temps de réalisation était conséquent, et nous avons trouvé un moyen plus simple et aussi efficace de réaliser cette nacelle. Nous nous sommes en effet servi de la découpeuse laser disponible à l'école.

A l'aide d'un site générant des schémas de boîte pouvant être imprimés à la découpeuse laser, nous avons imprimé une première boîte (voir Annexe 1). Cette version incluait des trous centrés sur chaque face afin de pouvoir faire passer une ficelle, pour relier la nacelle au ballon. De plus, le trou sur la face arrière était prévu afin de faire dépasser les moteurs permettant de pousser le dirigeable. Le trou sous la nacelle était censé permettre de passer un moteur sous le dirigeable, afin de réguler l'altitude du vol.



*Nacelle originale*

Cependant, cette nacelle est devenue obsolète pendant l'avancement de notre projet, pour plusieurs raisons. Tout d'abord, l'idée des moteurs d'AR Drone a été remplacée par les ventilateurs (voir partie suivante). Les trous à l'arrière et sous la nacelle devenaient donc inutiles.

De plus, un autre problème est apparu, concernant le poids de la nacelle. En effet, lors de nos différents essais, nous avons vu que notre ballon ne pouvait pas soulever autant de poids que nous l'espérions. Nous avons donc dû réduire le poids de l'ensemble, or,

le seul élément que nous pouvions modifier (sans changer la technique des ventilateurs) était la nacelle.

Ainsi, nous avons changé la structure de celle-ci pour nous adapter à ce nouveau choix. Nous avons donc gardé les trous pour maintenir la nacelle, mais avons supprimé ceux permettant de faire dépasser les moteurs. A la place, nous avons mis des trous sur la face inférieure, permettant de faire passer les câbles d'alimentation des ventilateurs (voir Annexe 2).

Enfin, nous avons changé la nature du bois : nous sommes passés du MDF au balsa, qui représente une importante diminution du poids : 138g pour le MDF, contre 50g pour le balsa. Pour finir, nous avons placé une plaque d'aluminium coudée à 90° aux extrémités afin de pouvoir maintenir 2 ventilateurs, tout en leur laissant l'appel d'air nécessaire à leur fonctionnement. Nous avons ainsi notre version finale de la nacelle.



*Nacelle finale*

# Solution testée : les moteurs d'AR Drone

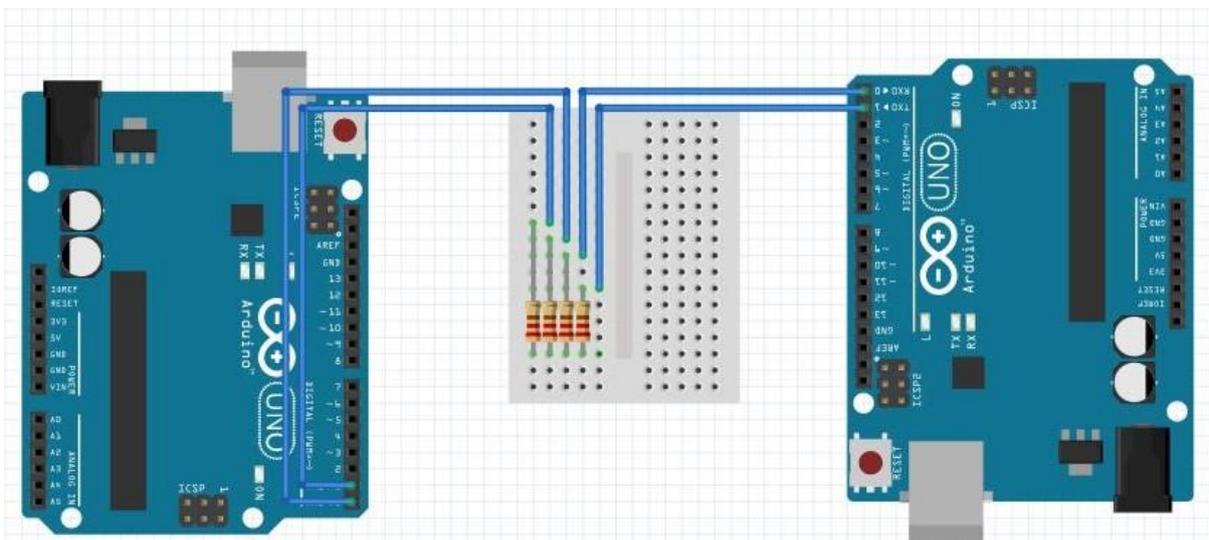
## 1) Contexte de la solution

L'école possédait un AR Drone défectueux, un de ses quatre moteurs étant HS. C'est pourquoi Monsieur Redon nous a donc demandé de récupérer cet AR Drone, d'en extraire les moteurs et de réussir à les contrôler, afin de les mettre dans la nacelle et de diriger notre structure en utilisant ces moteurs. Cette idée nous paraissait être, de loin, la meilleure afin de diriger de manière fiable le dirigeable.

## 2) Première étape : communication série

Après avoir récupéré les moteurs, nous avons fait des recherches qui nous ont permis de découvrir que les moteurs communiquaient avec la carte de contrôle par liaison série.

Cependant, cette liaison série s'effectuait sur un seul et unique fil, ce qui signifiait que les branches TX/RX étaient confondues. Or, un Arduino est construit, de base, pour avoir un port d'émission (TX) et un port de réception (RX). La première étape de notre solution était donc de réaliser un programme permettant à deux Arduino de communiquer entre eux via une liaison série unifilaire. Ce programme fut réalisé avec l'aide de Monsieur Redon (voir Annexe 3 pour le code). Nos recherches sur internet ont également abouti sur le fait de rajouter des résistances en série le long de la ligne de communication. Nous avons ainsi réalisé le montage final suivant, qui permet de communiquer via une liaison série sur un seul fil entre deux Arduino.

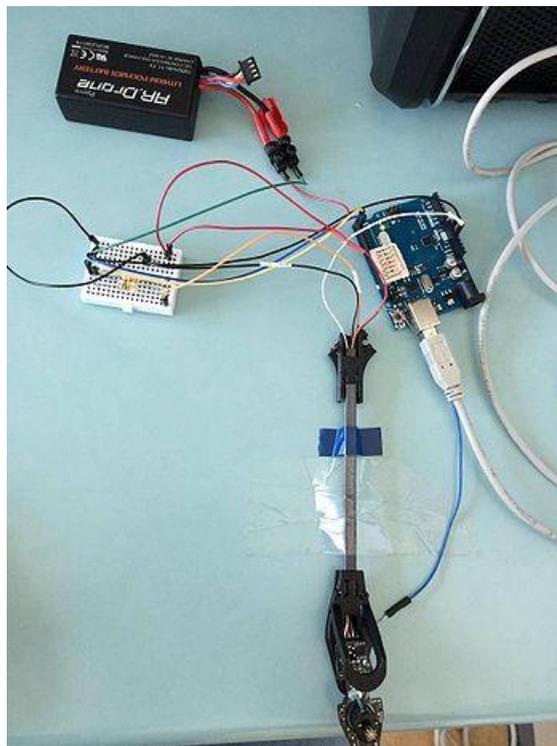


Communication série via un seul fil entre deux Arduino

Le programme de test de la liaison série unifilaire était basé sur un échange de “balle” entre les 2 entités. En effet, les 2 Arduino comptaient sur le principe du “Ping-pong” en se renvoyant la balle.

### 3) Deuxième étape : communication avec les moteurs AR Drone

Une fois le programme de communication terminé, nous sommes donc passés à l'étape suivante : réussir à communiquer avec les moteurs du drone. Pour cela, nous avons trouvé [un site internet](#) expliquant le protocole de communication. Tout d'abord, nous devons alimenter le moteur sur la batterie 11.4V fournie avec le drone, et mettre une masse commune. Un 3ème fil était connecté à une broche 5V de l'Arduino, tandis que le 4e fil était relié à la broche TX/RX (monté en série comme précédemment). Le dernier fil n'était pas relié, car nous n'avions pas besoin de gérer les interruptions. Le montage réalisé était donc le suivant.



Montage de contrôle d'un moteur AR Drone

Il fallait ensuite envoyer l'octet E0 au moteur afin d'initialiser la connexion. Celui-ci était censé nous répondre, et si sa réponse était correcte, cela signifiait que nous étions capables de contrôler le moteur.

Cependant, nous n'avons jamais réussi à recevoir une quelconque réponse de ce moteur. En effet, le programme effectuait bien le `send_serial` puis bloquait ensuite dans le `get_serial` qui lui ne pouvait donc acheminer aucun octet de réponse.

#### 4) Troisième étape : Vérification à l'analyseur de signaux

Suite à l'échec lors de l'essai de communication série unifilaire avec le moteur de l'AR Drone, nous avons décidé de vérifier si les signaux que l'on envoyait passaient bien lors de la liaison série.

Dès le premier test, nous n'observions aucune trace des signaux envoyés, ni de ceux reçus. Nous avons donc essayé de modifier notre code afin d'envoyer quelque chose de bien visible (0xFF puis 0x00). Aucun de ces tests ne s'est avéré concluant, c'est à ce moment que nous avons décidé de changer de solution afin d'avoir quelque chose à présenter en fin de projet à la soutenance ainsi que lors de la vidéo.

Avec un peu de recul, il est vrai que nous aurions dû essayer de débrancher le moteur du montage, afin de vérifier qu'au moins le *send\_serial* se voyait à l'analyseur. En effet, peut-être que le moteur était la source du problème et que ce dernier bloquait tous les octets entrants et sortants.

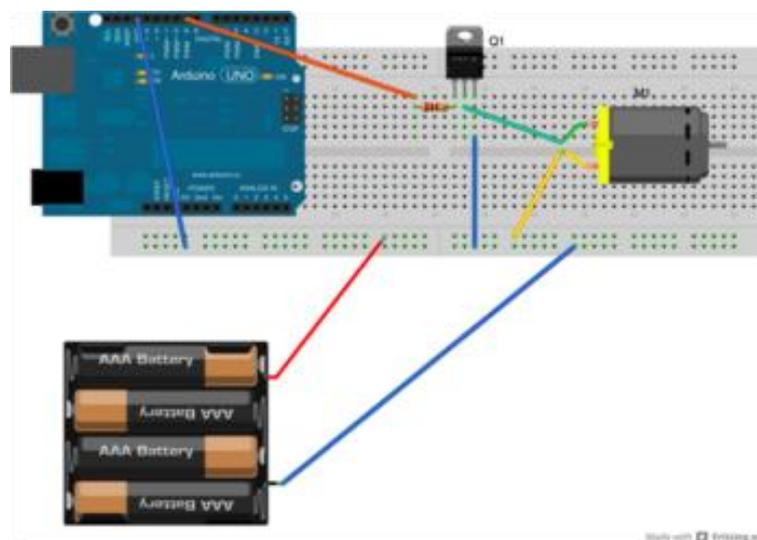
## Solution finale : le contrôle par ventilateurs

### 1) Contexte

Après l'échec de la solution de contrôle des moteurs AR Drone, nous nous sommes dirigés vers la solution proposée par un autre professeur : contrôler le ballon à l'aide de ventilateurs fixés à la nacelle. Cette solution paraissait plus simple dans le sens où il n'y avait pas de communication série avec les ventilateurs, et qu'un système de contrôle en tout ou rien suffirait à manœuvrer le ballon.

### 2) Premier essai : contrôle de 1 ventilateur

Pour commencer, nous avons cherché le moyen de contrôler un ventilateur, qu'on allumerait et éteindrait lors de la détection d'un obstacle. Pour réaliser ce montage, nous avons acheté des transistors MOSFET IRF-540-N, permettant de laisser passer ou non le courant en fonction de la tension présente sur la broche BASE du transistor. Le montage permettant le contrôle d'un ventilateur est le suivant (le code de contrôle correspondant se trouve en Annexe 4).

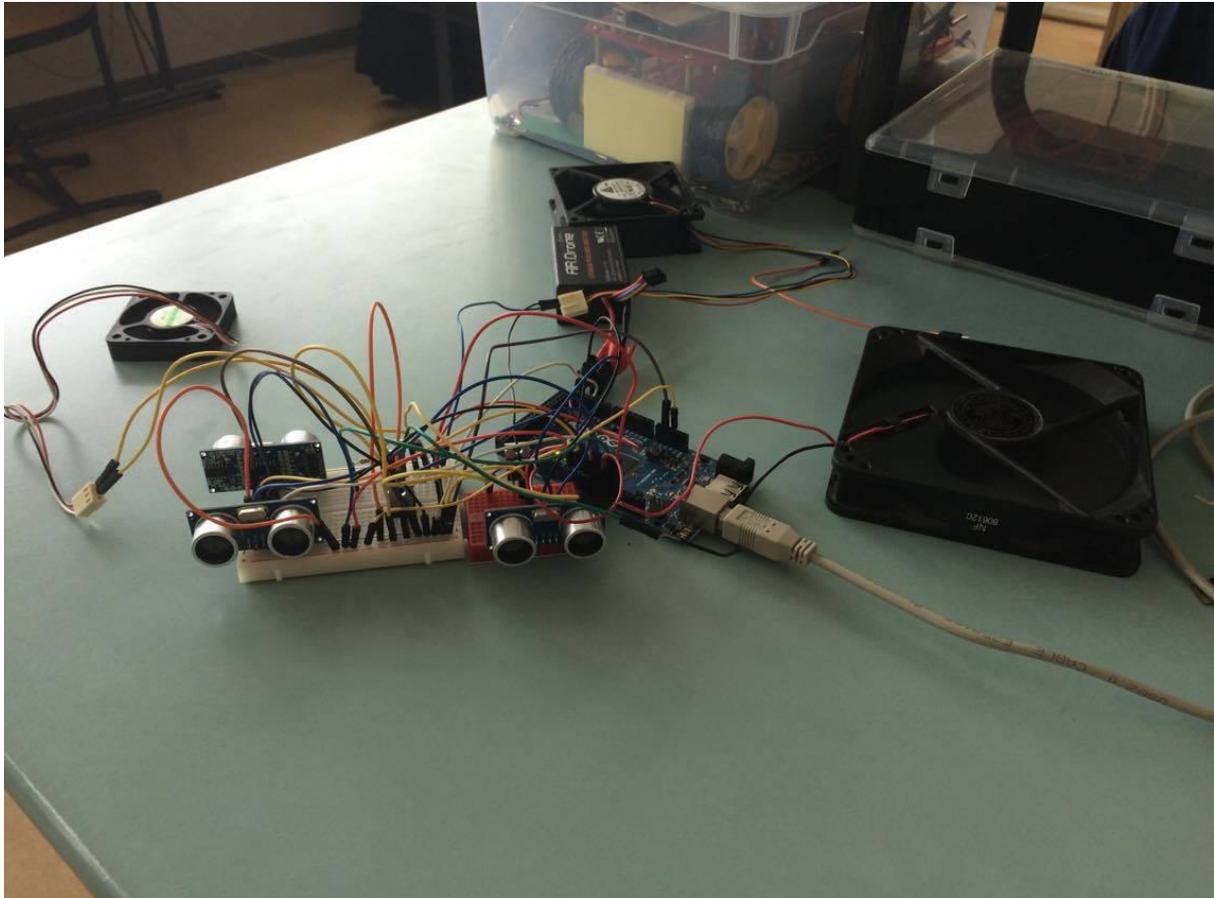


Contrôle d'un ventilateur via MOSFET

### 3) Essai de contrôle à 3 ventilateurs

Lorsque nous avons réussi à contrôler un ventilateur avec détecteur d'obstacle (le ventilateur s'arrête lorsque le capteur ultrasons détecte un obstacle à 2 mètres par exemple), nous avons tout de suite eu notre première idée applicable au dirigeable : utiliser 3 ventilateurs.

Le principe étant de fixer un ventilateur derrière la nacelle, un à gauche ainsi qu'un autre à droite. Le ventilateur fixé derrière la nacelle tournerait en permanence jusqu'à la détection d'un obstacle par le capteur devant la nacelle. Le ventilateur de droite se mettrait en route lors de la détection d'un obstacle à droite afin d'éviter ce dernier, et de même pour celui de gauche. Le montage suivant (hors de la nacelle) permettait de contrôler les ventilateurs de la manière décrite précédemment.

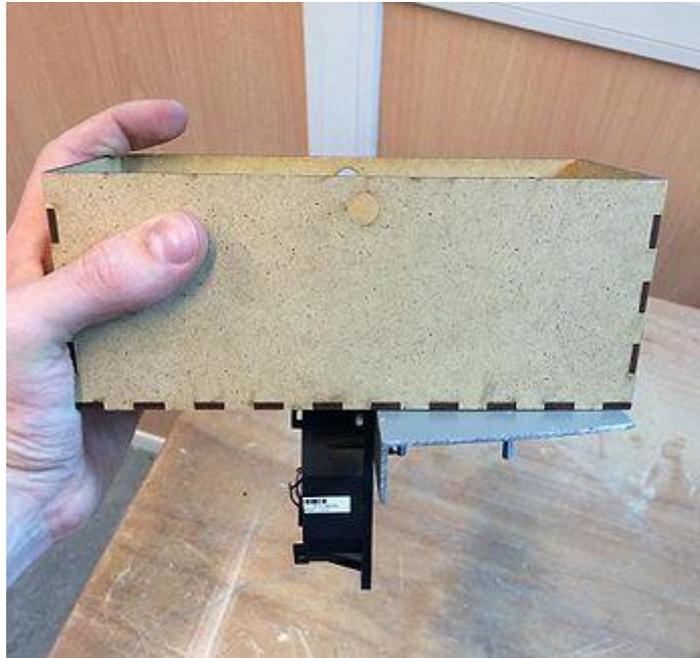


*Contrôle de 3 ventilateurs*

Malheureusement, lors de la réception du ballon et après l'avoir gonflé, nous avons constaté qu'il ne pouvait porter qu'un seul ventilateur en plus de la nacelle et des composants nécessaires au vol de notre dirigeable.

C'est donc pour cela que nous avons pensé à tenter le contrôle du ballon avec un seul ventilateur, cette solution nous permettait alors de gérer une seule direction, la ligne droite.

Afin de mettre en place cette méthode, nous avons gonflé le ballon à son volume maximal, puis nous avons trouvé expérimentalement le poids précis afin de garder une altitude constante. Une fois cette altitude constante atteinte, nous avons pu remplacer les lestes par les constituants de notre projet (Arduino, breadboard, batterie 12V). Le ventilateur est fixé sous la nacelle grâce à une plaque d'aluminium coudée à 90°.



*Nacelle allié à un seul ventilateur*

Ce test s'est avéré concluant, la nacelle et le ballon se déplaçaient bien en ligne droite à altitude constante, cependant la détection d'obstacle n'était pas encore mise en place.

#### 4) Solution de contrôle avec 2 ventilateurs

Une fois la solution à 1 ventilateur mise en place, le temps restant avant la fin du projet a été consacré au test du vol grâce à 2 ventilateurs. Afin que cette solution soit testable, nous devons réduire le poids total que le ballon devait soulever. Notre seul degré de liberté était bien entendu notre nacelle, c'est pourquoi nous avons décidé d'en recréer une, cette fois-ci en balsa. Cette nacelle en balsa pèse environ 100 grammes de moins que la précédente, ce qui nous permet alors de rajouter un ventilateur (voir partie Conception de la nacelle).

Nous avons donc coudé la plaque d'aluminium de la même manière de l'autre côté afin de venir y fixer le deuxième ventilateur. Cette plaque est précisément centrée en dessous de la nacelle (voir photo de la nacelle finale).

Cette méthode nous permettait alors une direction en double sens: allers/retours du dirigeable entre 2 obstacles grâce aux capteurs ultrasons situés aux opposés des ventilateurs.

Nous avons directement testé notre code qui déclenchait le ventilateur n°1 jusqu'à détection d'un obstacle. Lors de la détection, nous arrêtons ce ventilateur n°1 puis nous mettons en marche le deuxième afin de compenser la force d'entraînement et de parcourir le chemin retour, jusqu'à une nouvelle détection. Le programme était fonctionnel, seulement une fois embarqué, nous avons pu constater un problème non négligeable: la dérive de l'ensemble ballon/nacelle. Lorsque les ventilateurs inversent leurs rôles, cela crée une petite dérive de la nacelle, qui s'amplifie avec le temps. Très vite, environ au bout de 3 ou 4 secondes, le dirigeable prenait une direction totalement aléatoire.

Nous aurions aimé corriger ce problème de dérive, car nous possédions quelques solutions à essayer, mais par manque de temps nous avons préféré "perfectionner" la solution à 1 seul ventilateur.

## 5) Solution finale présentée

Afin d'être en mesure de présenter une solution viable lors du tournage de notre vidéo ainsi qu'à la soutenance, nous avons repris notre méthode qui fonctionnait déjà avec un ventilateur, puis nous y avons ajouté le capteur ultrasons.

Grâce à ce capteur, le ballon se déplace en ligne droite à altitude constante via l'équilibre parfait des forces puis détecte l'obstacle à 2,5 mètres et envoie l'ordre au ventilateur de cesser de fonctionner.

Une fois tous les composants embarqués, la solution se montre concluante. Toutefois, nous sommes obligés de détecter l'obstacle à cette distance afin d'éviter au ballon de rentrer en collision avec lui. En effet, la force d'entraînement créée par le ventilateur lors de la trajectoire continue de faire avancer l'ensemble même si le ventilateur est hors-tension.

## 6) Quelques idées d'améliorations

- Forme du ballon : essayer de le rendre plus ovale et donc plus "dirigeable". En effet, un ballon rond est beaucoup plus difficilement manœuvrable qu'un ballon ovale. Nous avons eu l'idée de gonfler le ballon en l'entourant de ficelle afin qu'il change de forme.
- Continuer de travailler sur les moteurs de l'AR Drone pour avoir un ensemble plus léger et donc plus d'agents de contrôles (facilement 4 moteurs et 4 hélices).
- Installer un accéléromètre pour calculer la dérive de la nacelle et la réguler grâce à d'autres ventilateurs (si plus de poids portable possible).
- Installer une régulation par PWM pour le contrôle des ventilateurs, ce qui permettrait un mouvement plus fluide et plus précis qu'un contrôle par "tout ou rien".

## Conclusion

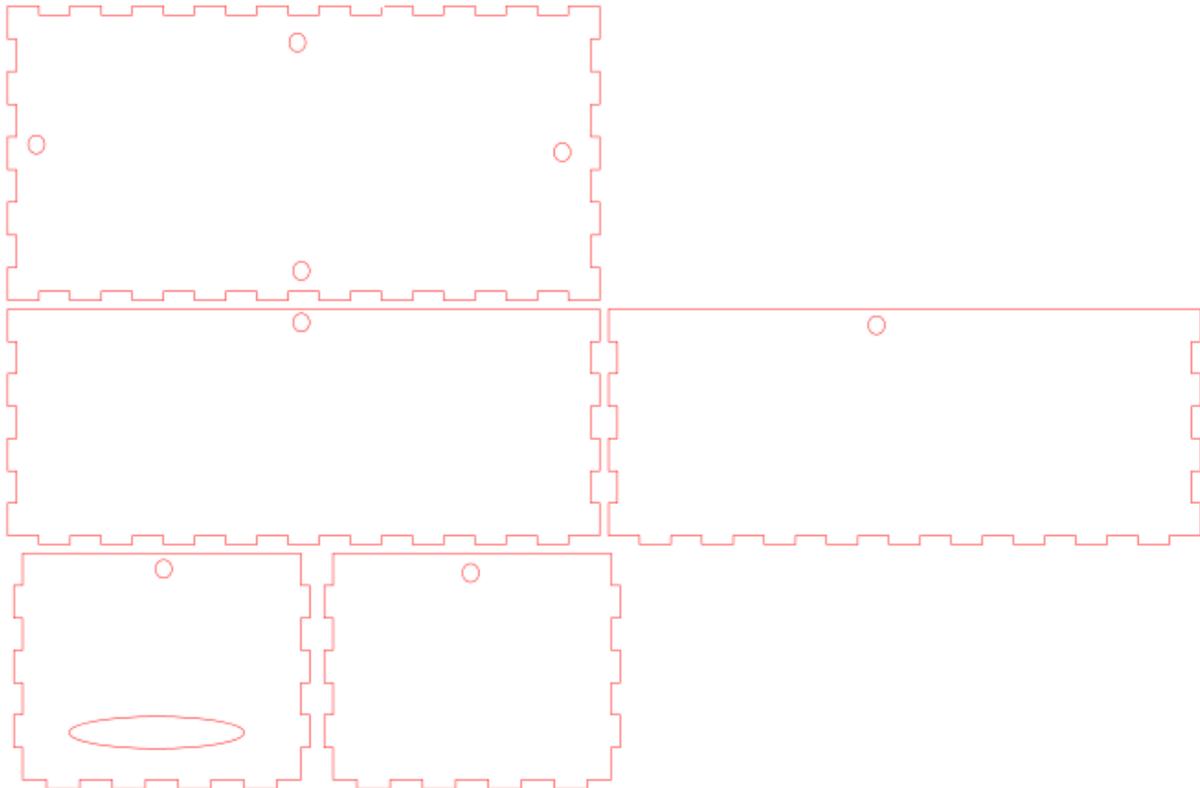
Tout au long de ce projet nous avons pu accroître nos compétences dans plusieurs domaines. Lors de la conception de la nacelle ainsi que des calculs primordiaux avant le gonflage du ballon, nous avons découvert quelques notions de mécanique que nous n'avions pas. Ensuite pendant l'établissement de codes pour gérer le déplacement de notre dirigeable, nous avons appris à programmer dans l'IDE Arduino même si les tests de communication avec le moteur se faisaient en C.

Il est également important de noter que nous avons du faire face à plusieurs problèmes durant ces six mois de travail. Certains d'entre eux ont été résolus, par nous-mêmes ou avec l'aide de notre tuteur, alors que pour d'autres nous n'avons pas encore trouvé la solution.

Enfin, nous avons tous les deux une notion de "projet" beaucoup mieux définie maintenant. Il est vrai que les précédents projets dont nous étions en charge étaient des projets sur le court terme, alors que celui-ci a duré plusieurs mois. C'est pour cela que nous avons du développer un sens de l'organisation ainsi que de la répartition du travail au sein de notre binôme afin d'être les plus efficaces possibles.

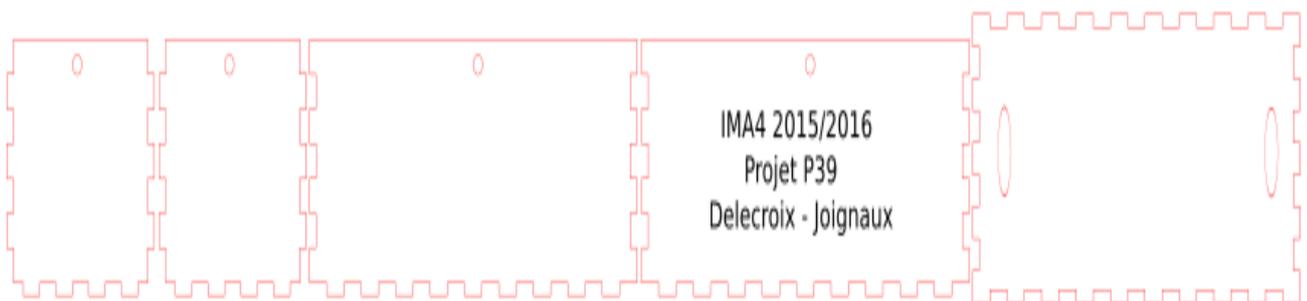
# Annexes

## Annexe 1



*Plan de notre première nacelle en MDF*

## Annexe 2



*Plan de la deuxième nacelle en Balsa*

## Annexe 3

### Fichier ping.c

```
#include <avr/io.h>
#include <util/delay.h>
#include <stdio.h>

#include "serial.h"

int main(void)
{
  DDRB=0x3f;
  unsigned char ball;
  init_serial(115200);
#ifdef PING
  ball=1;
  PORTB=ball&0x3f;
  send_serial(ball);
#endif
  while(1){
    ball=get_serial();
    PORTB=ball&0x3f;
    ball++; if(ball==0x40) ball=0;
    _delay_ms(100);
    send_serial(ball);
    _delay_ms(50);
  }
  return 0;
}
```

## Fichier serial.c

```
/** Functions for serial port **/  
  
#include <avr/io.h>  
#include <stdio.h>  
  
#include "serial.h"  
  
static int send_serial_printf(char c,FILE *stream);  
static FILE mystdout=FDEV_SETUP_STREAM(send_serial_printf,NULL,_FDEV_SETUP_WRITE);  
  
void init_serial(long int speed){  
/* Set baud rate */  
UBRR0 = F_CPU/(((unsigned long int)speed)<<4)-1;  
  
/* Enable only receiver */  
UCSR0B = (1<<RXEN0);  
  
/* Do not let TX as output be be able to receive */  
DDRD &= ~0x02;  
  
/* Set 8 bits character and 1 stop bit */  
UCSR0C = (1<<UCSZ01 | 1<<UCSZ00);  
  
/* Set off UART baud doubler */  
UCSR0A &= ~(1 << U2X0);  
}  
  
void send_serial(char c){  
UCSR0B &= ~(1<<RXEN0);  
UCSR0B |= (1<<TXEN0);  
DDRD |= 0x02;  
UDR0 = c;  
loop_until_bit_is_set(UCSR0A, TXC0);  
UCSR0A |= (1<<TXC0);  
DDRD &= ~0x02;  
UCSR0B &= ~(1<<TXEN0);  
UCSR0B |= (1<<RXEN0);  
}  
  
char get_serial(void){  
loop_until_bit_is_set(UCSR0A, RXC0);  
return UDR0;  
}  
  
void init_printf(void){  
init_serial(9600);  
stdout=&mystdout;  
}  
  
static int send_serial_printf(char c,FILE *stream){  
if(c=='\n') send_serial('\r');  
send_serial(c);  
return 0;  
}
```

## Annexe 4

### Fichier de contrôle avec 3 ventilateurs et 3 capteurs ultrasons

---

```
// définition des broches utilisées pour chacun des 3 ventilateurs

int echo1 = 2;
int trig1 = 3;

int echo2 = 4;
int trig2 = 5;

int echo3 = 6;
int trig3 = 7;

// 3 mesures (1/ventilateur) seront effectuées
long lecture_echo1;
long lecture_echo2;
long lecture_echo3;

long cm1;
long cm2;
long cm3;

// déclaration des 3 pins permettant le contrôle des marche/arrêt de chaque ventilateur
int cont1 = 8;
int cont2 = 9;
int cont3 = 10;

void setup()
{
  pinMode(trig1, OUTPUT);
  digitalWrite(trig1, LOW);
  pinMode(trig2, OUTPUT);
  digitalWrite(trig2, LOW);
  pinMode(trig3, OUTPUT);
  digitalWrite(trig3, LOW);
  pinMode(echo1, INPUT);
  pinMode(echo2, INPUT);
  pinMode(echo3, INPUT);
  pinMode(cont1, OUTPUT);
  pinMode(cont2, OUTPUT);
  pinMode(cont3, OUTPUT);
  Serial.begin(9600);
}
```

```

}

void loop()
{
// mesure de la distance du capteur1 (relié au ventilateur 1) en cm
digitalWrite(trig1, HIGH);
delayMicroseconds(10);
digitalWrite(trig1, LOW);
lecture_echo1 = pulseIn(echo1, HIGH);
cm1 = lecture_echo1 / 58;
Serial.print("Distance1 m : ");
Serial.println(cm1);
// mise en route du ventilateur si la distance mesurée est inférieure à 50cm
if (cm1<50){
    digitalWrite (cont1, HIGH);
}
else {
    digitalWrite (cont1, LOW);
}

digitalWrite(trig2, HIGH);
delayMicroseconds(10);
digitalWrite(trig2, LOW);
lecture_echo2 = pulseIn(echo2, HIGH);
cm2 = lecture_echo2 / 58;
Serial.print("Distance2 m : ");
Serial.println(cm2);
if (cm2<50){
    digitalWrite (cont2, HIGH);
}
else {
    digitalWrite (cont2, LOW);
}

digitalWrite(trig3, HIGH);
delayMicroseconds(10);
digitalWrite(trig3, LOW);
lecture_echo3 = pulseIn(echo3, HIGH);
cm3 = lecture_echo3 / 58;
Serial.print("Distance3 m : ");
Serial.println(cm3);
if (cm3<50){

Serial.println(cm3);
if (cm3<50){
    digitalWrite (cont3, HIGH);
}
else {
    digitalWrite (cont3, LOW);
}

delay(500);
}
}

```