

|  |
| --- |
|  |
| Aéroglisseur radiocommandé |
| Projet GE II 2020 |



2020

NUNEZ ANGO Walter

SOUDIER Jean

GE4

SOMMAIRE

[Introduction 2](#_Toc34985457)

[I. Généralitées 3](#_Toc34985458)

[1. Le cahier des charges 3](#_Toc34985459)

[2. Analyse fonctionnelle 3](#_Toc34985460)

[3. Gantt 4](#_Toc34985461)

[II. Partie mécanique 5](#_Toc34985462)

[1. La structure 5](#_Toc34985463)

[III. Partie puissance 5](#_Toc34985464)

[3.1 Les composants imposés pour la motorisation 5](#_Toc34985465)

[3.1.1 Batterie BRAINERGY 5.200mAh 3S 45C 5](#_Toc34985466)

[3.1.2 Moteur Brushless NTM Prop Drive Series 35-30 1100kv / 380w 6](#_Toc34985467)

[3.1.4 Driver de demi-point MIC4104YM 8](#_Toc34985468)

[3.1.5 Condensateur Aluminium Polymère 330uF-16V 8](#_Toc34985469)

[3.2 Tests et mesure 9](#_Toc34985470)

[3.2.1 L’essai à vide 10](#_Toc34985471)

[3.2.2 l’essai en court-circuit 12](#_Toc34985472)

[3.2.3 L’essai de lâcher 13](#_Toc34985473)

[3.2.4 L’essai en DC 14](#_Toc34985474)

[IV. Partie commande 15](#_Toc34985475)

[4.1 Le schéma de principe 15](#_Toc34985476)

[4.2 Les composants 15](#_Toc34985477)

[4.2.1 Les microcontrôleurs : 15](#_Toc34985478)

[4.3 Les fonctionnalitées à remplir 17](#_Toc34985479)

[4.4 Les algorigrammes 17](#_Toc34985480)

[V. Partie application 18](#_Toc34985481)

[5.1. La vue utilisateur 18](#_Toc34985482)

[5.2. L’algorigramme 18](#_Toc34985483)

[Conclusion 20](#_Toc34985484)

[Table des illustrations 21](#_Toc34985485)

# Introduction

Le but de ce projet est de mettre en application les connaissances acquises dans le module d’électronique de puissance et électronique numérique. Le thème de cette année est la conception d’un aéroglisseur radiocommandé.

Cet aéroglisseur fonctionnera à l'aide d'un moteur brushless triphasé qui, entraîné par un onduleur, fera tourner une hélice afin de faire avancer l'aéroglisseur en faisant gonfler un coussin d'air. De plus, nous utiliserons un servomoteur commandé par un microcontrôleur pour pouvoir diriger l'aéroglisseur vers la droite ou la gauche. L'alimentation du système sera réalisée à l'aide d'une batterie Lipo 3S (3 cellules en série).

Dans ce projet, nous avons pour objectif l'étude et la réalisation d'un aéroglisseur radiocommandé, La conception de ce projet comporte une partie mécanique, partie motorisation et une partie électronique.

En effet, les différents étages de notre système ont été d’abord étudiés de façon théorique et mathématique, puis en simulation avec les logiciels Proteus, PSIM et Matlab. Par la suite, nous allons réaliser des tests sur une plaquette à essais pour observer les différents signaux obtenus et ainsi pouvoir ajuster nos montages. Pour finir nous allons concevoir le système grâce à des circuits imprimés.

Dans la suite de ce rapport, nous allons annoncer dans une première partie l'analyse fonctionnelle de notre système ainsi que le cahier des charges que nous devions respecter pour chaque partie. Ensuite, nous allons expliquer le fonctionnement de la partie commande du système, composé d'un module Bluetooth permettant l'interface entre l'utilisateur et l'aéroglisseur et d'un microcontrôleur de type PIC.

De plus, nous parlerons de différents tests effectués pour déterminer les caractéristiques du moteur.

1. Partie mécanique

## La structure

Nous avons choisi de nous rapprocher de la structure proposée en suivant les plans. Nous avons dû adapter les côtes sur le plan pour respecter les dimensions du cahier des charges. En effet, il est imposé de réaliser une structure de taille maximale 350x250x300 mm. Le schéma proposé comportait les dimensions 430x200x200 mm. Le choix a été fait de conserver au maximum les côtes de la partie avant qui permet de garder l’aéroglisseur plaqué au sol.

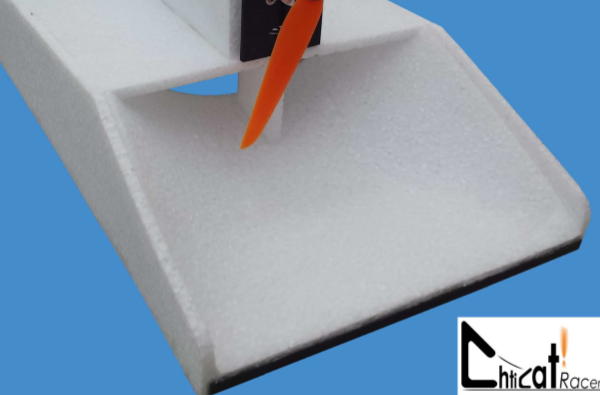


Figure 1 : Partie avant de l'aéroglisseur

Pour pallier à la différence de longueur nous avons alors choisi de réduire le corps de l’aéroglisseur, tout en conservant les dimensions du gouvernail. Ces 2 choix ont été motivé pour conserver la stabilité mécanique qui a dû être pensé pour ces plans. Ainsi, notre aéroglisseur ne devrait pas se soulever facilement et pouvoir changer de cap sans problèmes.

Nous avons aussi dû agrandir la largeur de la structure pour pouvoir contenir l’hélice. En effet, l’hélice fait 230 mm de diamètre et nous devons ajouter une protection autour de celle-ci. Pour éviter tout dégâts, il a été décidé d’utiliser la largeur maximale autorisé de 250 mm. De même, pour la hauteur de 300 mm maximum car nous avions conservé la structure avant qui surélève de quelques centimètres le montage.

Enfin, pour la protection de l’hélice, il a été choisi d’utiliser des montant en métal ou du PCB avec un grillage métal collé dessus et du carton collé sur les côtés. Ci-dessous, un aperçu CAO de la structure finale.

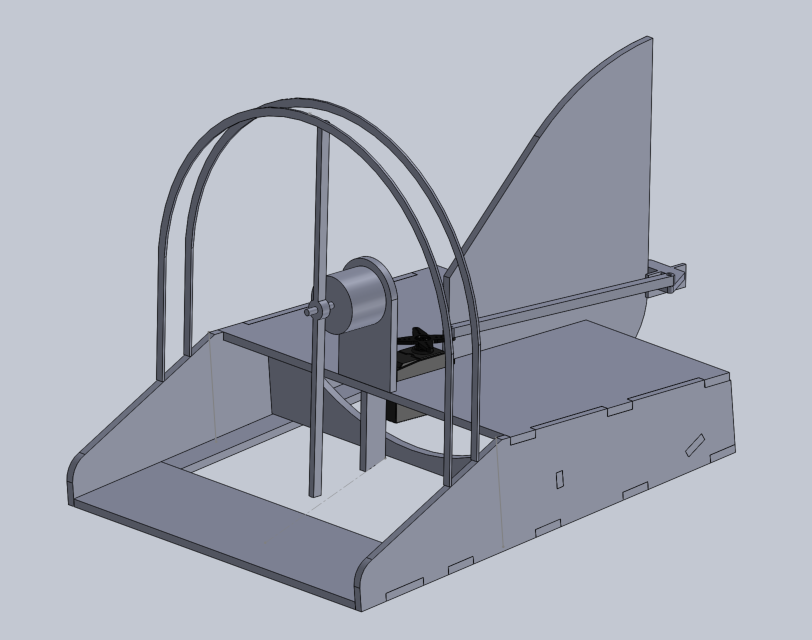


Figure 2 : Structure finale de l'aéroglisseur

1. Partie puissance

# Partie commande

## Le choix des modules

### 3.1.1 Le Module Bluetooth

Le Module Bluetooth est réglé pour communiquer en UART à 9600 baud avec des clé BLE qui sont :

UUID : 0000FFE0-0000-1000-8000-00805F9B34FB

CHAR : 0000FFE1-0000-1000-8000-00805F9B34FB

Nous avons choisi de laisser la configuration de base. Nous avons simplement changé le nom du module en « Jean-Walter ». La configuration UART de base étant assez rapide pour cette application (9600 messages/s pour un envoi toutes les 20 ms de 6 à 12 mots soit au maximum ).

### 3.1.2 Le PIC

La programmation du PIC a été réaliser avec MCC (MplabX code configurator), une interface graphique permettant de régler les registres de configuration des modules. Pour remplir le cahier des charges, nous avons choisi d’utiliser les modules suivants :

* Timer 2 pour envoyer périodiquement la vitesse au DsPIC et pour générer la fréquence de la PWM du servomoteur
* PWM3 pour gérer le servomoteur. Nous avons choisi d’utiliser une PWM interne plutôt qu’un timer émuler pour simplifier le code et pour avoir plus de précision
* EUSART pour la communication avec le module Bluetooth
* L’I2C pour la communication entre le PIC et le DsPIC. Nous avons la possibilité d’aller plus loin dans ce projet et d’asservir le servomoteur en angle, or le capteur réalisant cette tache fonctionne en I2C. Nous avons alors préféré l’I2C au SPI pour n’avoir qu’un seul bus.
* Une entrée numérique pour le courant pour gagner du temps sur les interruptions en placent un trigger physique. Néanmoins, des ponts sont prévus et le broche est analogique au cas ou le trigger poserais problème, nous pourrions utiliser l’ADC.
* L’accélérateur mathématique pour les calculs du PID car il réalise les calculs en 16 bits sur un microcontrôleur 8 bits et fonctionne en interruption, ce qui ne freine pas la boucle du timer 2. De plus, l’asservissement numérique est mieux géré car le temps de boucle est géré en interne.

### 3.1.3 Le DsPIC

La programmation du DsPIC s’est fait avec des registre car le module MCC ne prend pas en charge ce DsPIC. Ainsi, nous avons mis en place les modules suivants :

* L’ADC pour la mesure du courant qui limitera le courant dans le moteur. Il aura un seul pour couper la commande. Nous avons choisi de ne pas mettre de trigger ici car nous avons la possibilité à terme de réguler le courant.
* La PWM triple pour le fonctionnement des 3 bras de ponts
* Les interruptions sur les entrées numériques pour ne pas louper de front sur les sorties des comparateurs qui recréent la position du rotor via le FEM.
* L’I2C pour communiquer avec le PIC

### 3.1.4 L’application

Nous avons fait le choix de développer notre propre application mobile car cela ne prenait pas énormément de temps. De plus, il existe peu d’interface se pliant à notre besoin. Pour des raisons de souplesse nous avons alors développé notre application sur App Inventor avec les modules suivant :

* Un Timer pour assurer un envoi constant de données.
* Un module Bluetooth BLE (Low Energy) pour pouvoir communiquer avec notre module physique.
* Un centre de notification pour avoir un retour utilisateur.
* Des boîtes d’arrangement pour avoir une interface « responsive » (qui s’adapte à la taille de l’écran).
* Des boutons et des images.

### 3.1.5 Le capteur gyroscopique (MPU9265)

Il sera paramétré à l’allumage par le PIC avec un filtre passe bas avec une fréquence de coupure à 92 Hz et une échelle de 8G maximum.

## Le choix des modules

### 3.2.1 Le PIC

Les modules du PIC sont réglés comme ci-dessous :

* Le système est réglé avec 4 MHz d’horloge principale, oscillateur interne, pas de chien de garde, pas de PLL et la broche de remise à zéro n’est pas une entrée.
* Les interruptions dans l’ordre sont (1) EUSART réception, (2) Timer 2 dépassement de comptage, (3) mathématique accélérateur.
* EUSART est réglé à 9600 baud, en mode de transmission et réception 8 bits, les données sont non inversées, les interruptions sont activées pour la réception. En effet, le code pourra être bloquant à l’envoi étant donné que le flux n’est pas énorme. Au contraire, la réception est critique pour le pilotage. La transmission à 9600 baud n’a pas pu être accéléré car avec la fréquence d’horloge de 4 MHz, on obtient une erreur trop grande. Or on ne pas accélérer l’horloge car le diviseur du timer 2 ne permet pas de générer un temps plus grand que 32 ms à 4 MHz, si on accélère, on ne pourra plus réaliser les 20 ms de période pour le servomoteur.
* L’I2C est réglé avec une vitesse de transmission de 100 kHz, en maitre sans interruptions. Nous avons décidé que le PIC serait le servo du système et le DsPIC ferait simplement l’interface entre la donné en vitesse et les commandes à appliquer. Aucune interruption n’est utilisée en raison du protocole. Encore une fois, l’envoi peut être bloquant dans le code, or la réception se fait avec l’envoi. Il faut envoyer un message pour recevoir une donnée (Maitre/Esclave).
* La PWM 3 est régler pour suivre le Timer2 avec une résolution de 9 bits qui lui confère une fréquence de 50.08 Hz (environ 20 ms).
* Le Timer 2 est réglé sur la fréquence interne divisé par 4 pour pouvoir être utilisé par la PWM avec un diviseur de fréquence de 1/128ième. Il est paramétré pour fonctionner en continu. Il n’y a pas de diviseur postérieur au résultat car il n’a pas d’utilité, il n’affecte pas la PWM. La période est réglée à 20 ms avec des interruptions.
* L’accélérateur mathématique est réglé pour fonctionner en mode PID. Ici, nous avons implémenté un simple PI pour garder de la simplicité et conserver une erreur statique nul. Le temps d’échantillonnage est réglé à 40 ms. Les interruptions sont activées pour ne pas bloquer le code du timer 2. Les paramètres PI serons déterminé expérimentalement dû fait de la complexité de modélisation du système. Une tentative est présentée ci-dessous. Les principaux problèmes de modélisation résident dans la prise en compte du couple résistant lié à la pousser du moteur brushless et de l’orientation du gouvernail. De plus, bon nombre de paramètres sont indisponibles comme les constantes électriques et mécaniques du moteur DC dans le servomoteur et du gouvernail.

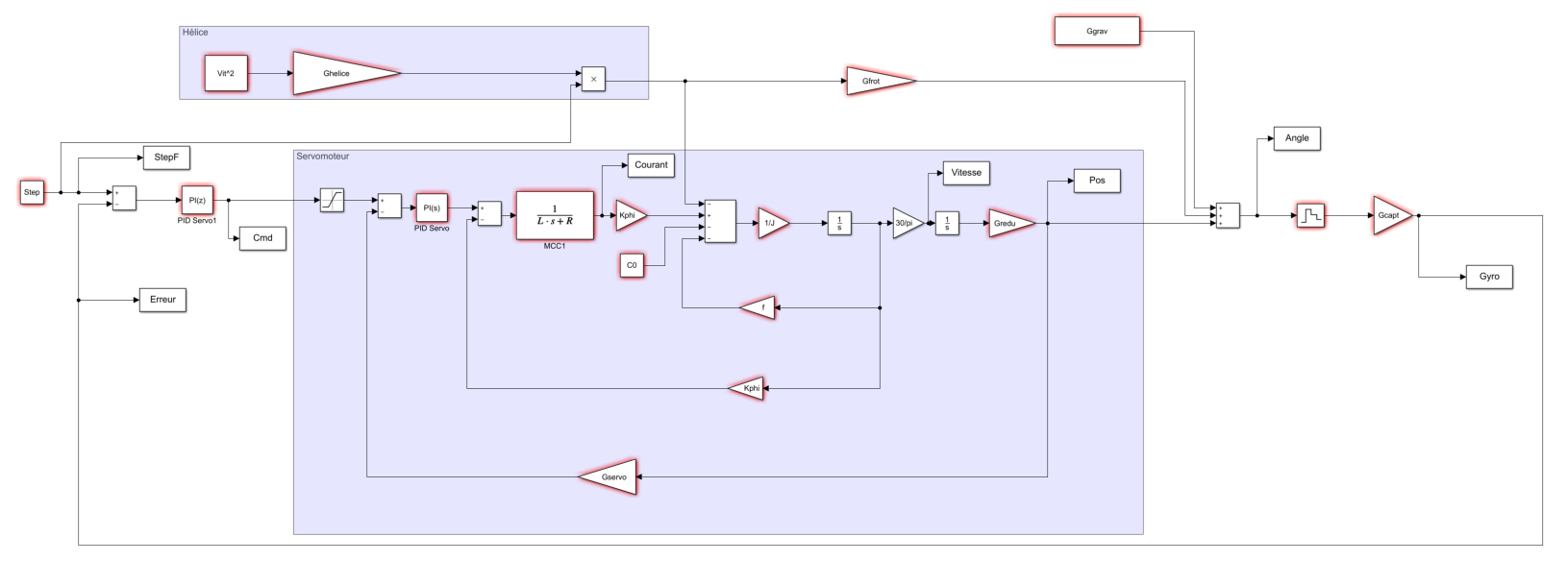


Figure 3 : Modèle d'asservissement du servomoteur en angle

### 3.2.2 Le DsPIC

Les modules du PIC sont réglés comme ci-dessous :

* Le système est réglé avec une horloge interne et une PLL (multiplicateur de fréquence) de 16, pas de chien de garde, la broche de remise à zéro en fonctionnement normale, les PWM active sur l’état haut et la programmation sur les broches habituelles.
* L’ADC est réglé pour fonctionner sur la broche RB2 uniquement, pour convertir au rythme de la PWM, pour fonctionner sur son oscillateur interne (un circuit RC) et fonctionner en interruption.
* La PWM est réglé pour fonctionner à environ 25 kHz pour ne pas créer un bruit gênant et limiter les pertes à la commutation. Elle est aussi réglée pour suivre les entrées d’interruption via OVDCON et ainsi appliquer le bon cycle sur les PWM. Elle fonctionne en continu et ne génère pas d’interruption.
* Le Timer 3 est réglé à 20 ms de période avec des interruptions pour traiter les informations ainsi que la protection de déconnexion de la télécommande. Il est réglé avec un diviseur de fréquence de 256 et fonctionne sur l’horloge interne.
* Les entrées d’interruption sont paramétrées pour fonctionner sur CN5, CN6 et CN7 avec des interruptions.
* L’I2C est réglé avec un diviseur d’horloge pour pouvoir recevoir une fréquence d’environ 100 kHz, en mode esclave, avec des interruptions de réception. En effet, il ne faut louper aucun message entrant.

### 3.2.3 L’application

Les modules de l’application sont réglés de la manière suivante :

* Le Timer est réglé en interruption avec une période de 40 ms, la transmission ne pouvant pas aller plus vite du fait du baud rate.
* Le module BLE est réglé en auto-reconnection avec un temps d’arrêt de 10 secondes. Les interruptions sont paramétrées pour la connexion, la réception de données, la déconnexion, quand une erreur de connexion survient et quand il découvre un nouveau périphérique.
* Les boutons sont tous paramétré en interruption au touché.

# Les Circuits imprimés

Il a été décider de réaliser 4 circuit imprimé différents. A savoir un circuit pour les régulateurs, un pour le DsPIC, un pour les bras de pont (réalisé en 3 exemplaires) et un pour le PIC et la partie de commande.

# Conclusion

Ce projet nous permet de mettre directement en application nos connaissance apprissent au semestre précédent mais aussi celle acquît au long de la formation. Il y a une grande partie électronique de puissance et numérique mais aussi de l’automatique (régulation de courant) et de l’électronique analogique (circuit de mesure). Dans ce rapport, nous avons pu nous rendre compte de la densité et la diversité des domaines touchées par ce projet. Nous avons pu aborder les notions importantes comme la mesure de la FEM et la communication au BLE (Bluetooth Low Energie).

# Table des illustrations

[Figure 1 : Schéma de principe du projet 3](#_Toc34985255)

[Figure 2 : Diagramme de Gantt 4](#_Toc34985256)

[Figure 3 : Modèle mécanique d'aéroglisseur proche de celui qui sera réalisé 5](#_Toc34985257)

[Figure 4 : Batterie Lipo 5](#_Toc34985258)

[Figure 5 : Données de la batterie 6](#_Toc34985259)

[Figure 6 : Moteur brushless du projet 6](#_Toc34985260)

[Figure 7 : Caractéristiques du moteur 7](#_Toc34985261)

[Figure 8 : MOSFET du projet 7](#_Toc34985262)

[Figure 9 : Données du MOSFET 8](#_Toc34985263)

[Figure 10 : Image du driver du projet 8](#_Toc34985264)

[Figure 11 : Données du driver 8](#_Toc34985265)

[Figure 12 : Condensateur du projet 8](#_Toc34985266)

[Figure 13 : Données du condensateur 9](#_Toc34985267)

[Figure 14 : Graphique de la réponse résistive du condensateur en fonction de la fréquence 9](#_Toc34985268)

[Figure 15 : Schéma équivalent d'une phase du moteur 10](#_Toc34985269)

[Figure 16 : Photo du banc de test moteur 10](#_Toc34985270)

[Figure 17 : Schéma équivalent des différentes parties du moteur 11](#_Toc34985271)

[Figure 18 : Données expérimentales relevées sur le moteur 11](#_Toc34985272)

[Figure 19 : Chronogrammes des essais à vide 11](#_Toc34985273)

[Figure 20 : relevé expérimentale en court-circuit 12](#_Toc34985274)

[Figure 21 : Chronogrammes du moteur en court-circuit 12](#_Toc34985275)

[Figure 22 : relevés expériementale en court-circuit 13](#_Toc34985276)

[Figure 23 : relevé expérimentale de l'essai de lâcher 13](#_Toc34985277)

[Figure 24 : Chronogramme des essais de lâcher 13](#_Toc34985278)

[Figure 25 : relevé expérimentale de l'essai en DC 14](#_Toc34985279)

[Figure 26 : Chronogramme de l'essai en DC 14](#_Toc34985280)

[Figure 27 : Schéma de principe des différentes cartes de comande 15](#_Toc34985281)

[Figure 28 : Vue physique du PIC 16](#_Toc34985282)

[Figure 29 : Vue physique du DsPIC 16](#_Toc34985283)

[Fréquence d’émission 2,4 GHzFigure 30 : Image du module bluetooth 16](#_Toc34985284)

[Figure 31 : Algorigramme du PIC 17](#_Toc34985285)

[Figure 32 : Algorigramme du DsPIC 18](#_Toc34985286)

[Figure 33 : Algorigramme de l'application mobile 19](#_Toc34985287)