

# **Liste de l'ensemble des fonctions que nous devons coder pour faire fonctionner l'aéroglesseur**

On établit dans ce document la liste de nos principales fonctions et leur prototypage, afin d'avoir les idées claires avant de commencer la programmation.

Pour commencer, voici un bilan de comment le projet est perçu.

On suppose que le système est constitué de :

- 4 entrées
  - un bouton agissant directement sur l'alimentation comme interrupteur :
  - un bouton switch on/off qui met le système sous tension
  - deux entrées digitales :
    - un bouton poussoir servant à sélectionner le mode souhaité (Cf : cahier des charges)
    - un bouton servant à activer le mode sélectionné
  - une entrée analogique
    - un capteur tachymétrique dont les données serviront à corriger la trajectoire ou la vitesse de notre hovercraft (ce qui constitue la part d'asservissement).
- 3 sorties analogiques qui commandent :
  - un moteur qui fait tourner une hélice orientée vers le bas (dont le rôle est d'assurer la suspension du véhicule, pour réduire les frottements avec le sol)
  - un moteur qui fait tourner une hélice orientée vers l'arrière (dont le rôle est de propulser l'aéroglesseur vers l'avant)
  - un servomoteur qui contrôle les ailerons et permet de maintenir la trajectoire de l'aéroglesseur.

Pour répondre au cahier des charges, nous devons établir 3 fonctions, permettant de coder les 3 modes : avancer droit, avancer sur une longueur de 10 mètres, ou avancer le plus rapidement possible.

Nous ajoutons à cette liste un mode plus simple "avancer 5 secondes" (sans nous préoccuper de la trajectoire) pour d'abord nous familiariser avec notre système.

Dans l'idéal, nous aimerions pouvoir aussi sélectionner le mode que l'on souhaite, à l'aide du bouton d'entrée prévu à cet effet.

Notre système aurait finalement 5 états différents (sélection, avancer, avancer droit, avancer de 10m, avancer le plus vite possible)

## 1) Fonction “main” :

Cette fonction permet le fonctionnement global de l’aéroglesseur. Elle réalise une boucle infinie qui exécute successivement SelectMode, puis la fonction associée au mode sélectionné.

En langage C, elle devrait ressembler à :

```
int main(void){  
    mode = 0;  
    while(1) {  
        mode = SelectMode()  
  
        switch(mode){ //on exécute la bonne fonction selon le mode choisi  
            case 0:  
                Forward();  
                break;  
            case 1:  
                ForwardFront();  
                break;  
            case 2:  
                Forward10();  
                break;  
            case 3:  
                ForwardFast();  
                break;  
        }  
        return 0; //jamais atteint car on ne sort pas du while précédent  
    }  
}
```

Elle fera appel aux fonctions suivantes ...

- **Forward (void)** : “avancer 5 secondes” => doit donner la consigne pour faire tourner correctement les moteur 1 et moteur 2
- **ForwardFront (void)** : avancer droit => rapport entre moteur 1 et moteur 2 + servomoteur
- **Forward10 (void)** : avancer droit sur 10m => prendre en compte la vitesse (ou l’accélération) de l’hovercraft pour qu’il s’arrête au bout de 10m
- **ForwardFast (void,)** : avancer le plus vite possible => prendre en compte la vitesse de l’hovercraft afin de déterminer la vitesse optimale pour aller le plus vite possible

Nom de la fonction	Paramètres en entrée	donnée(s) en sortie	Utilité de la fonction
SelectMode	void	int	Renvoie un int “selecteur” qui déterminera quelle fonction appeler lorsque “run” est exécuté
Forward	void	void	
FastForward	void	void	
Forward10	void	void	
ForwardFast	void	void	

**2) Mode 0 : “avancer 5 secondes” → Forward() :**

Ce mode nécessitera les fonctions suivantes...

[illegible]

### 3) Mode 1 : “avancer droit” → ForwardFront():

Ce mode nécessitera les fonctions suivantes...

[illegible]

#### 4) Mode 2 : “avancer droit sur 10 mètre” → Forward10()

Ce mode nécessitera les fonctions suivantes...

[illegible]

5) Mode 3 : “parcourir 10m le plus vite possible” → ForwardFast()

Ce mode nécessitera les fonctions suivantes...

[illegible]

## 6) Fonction de sélection

Le prototype de la fonction est (void → void).

La fonction lit en boucle les fronts montants du bouton de sélection, **tant que le bouton run n'est pas appuyé**, et retourne en sortie le nombre, compris entre 0 et 3, correspondant au mode souhaité.

Note : la fonction doit faire une boucle pour sélectionner à nouveau le 1er mode après avoir dépassé le 4ème (dernier) mode. On utilisera certainement l'opérateur modulo “%4”.

• **SelectMode (void)** : permet choix du mode =>

*selecteur = 0*

*Tant que (“le bouton “run” n’est pas appuyé) :*

*lecture du bouton poussoir*

*si “front montant” :*

*// Il faudra élaborer une fonction pour détecter un front montant du bouton de sélection*

*// Pour cela, on peut garder en mémoire l'état actuel du bouton ainsi que son état à*

*// l'instant précédent, pour déduire quand le bouton est appuyé*

*selecteur = (selecteur + 1) % 4*

*// permet de choisir entre les 4 premiers modes*

*renvoyer selecteur*

En langage C, elle devrait ressembler à :

```
int SelectMode() {
    int selecteur = 0;
    int btn_run = 0;
    int btn_sel_current = 0;
    int btn_sel_previous = 0;
    while (btn_run != 1) :
        btn_sel_previous = btn_sel_current;
        btn_sel_current = (*lecture du bouton selecteur*);
        if (btn_sel_current & ((btn_sel_prev+1)%2)) {
            // on s'assure que le bouton est appuyé et qu'il ne l'était pas avant
            selecteur = (selecteur + 1) % 4 ;
            btn_run = (*lecture du bouton run*);
        }
    }
    return selecteur; // permettra dans le main() de choisir entre les 4 premiers modes
}
```