

# DRONE STATIONNAIRE

## PROJET ELECTRONIQUE PEIP 2 LEHYAN FLOURIOT G3

### Rapport de séance 2 :

Dans cette séance, je vais calibrer les ESCs ; ces composants électroniques sont une « passerelle » entre l'Arduino et le moteur. Je vais envoyer un signal sous forme de PWM à un ESC et celui-ci s'occupera de faire tourner le moteur avec une vitesse proportionnelle à la valeur du PWM.

Je vais donner donc la plage de puissance des moteurs comme indiqué :

Ceux-ci vont varier entre une valeur de PWM = 0 (moteur à l'arrêt) et une valeur de PWM = 180 (moteur plein régime). J'ai choisi une valeur de 180 pour plusieurs raisons : premièrement, les documentations sur les calibrages d'ESCs utilise cette plage ; deuxièmement, cette valeur est une valeur « angulaire » ce qui simplifiera la visualisation et les calculs lors de l'étape de stabilisation avec l'accéléromètre 6 axes.

D'après mes recherches, je suis dans l'obligation de recalibrer les ESCs à chaque mise sous tension du drone. Je vais donc pour le moment les calibrer manuellement en exécutant les actions les unes à la suite des autres, puis je rédigerai un code me permettant de le faire automatiquement à chaque démarrage.

### Calibrage :

```
/**
 * Usage, according to documentation(https://www.firediy.fr/files/drone/HW-01-V4.pdf) :
 * 1. Plug your Arduino to your computer with USB cable, open terminal, then type 1 to
send max throttle to every ESC to enter programming mode
 * 2. Power up your ESCs. You must hear "beep1 beep2 beep3" tones meaning the power supply
is OK
 * 3. After 2sec, "beep beep" tone emits, meaning the throttle highest point has been
correctly confirmed
 * 4. Type 0 to send 0 throttle
 * 5. Several "beep" tones emits, wich means the quantity of the lithium battery cells (3
beeps for a 3 cells LiPo)
 * 6. A long beep tone emits meaning the throttle lowest point has been correctly
confirmed
 * 7. Type 2 to launch test function. This will send 0 to 180 throttle to ESCs to test
them
 */

#include <Servo.h>

Servo motA, motB, motC, motD;
char data;

/**
 * Initialisation routine
 */
void setup() {
    Serial.begin(9600);
```

```

    motA.attach(3, 1000, 2000); // 1000 and 2000 are very important ! Values can be different
    with other ESCs.
    motB.attach(5, 1000, 2000);
    motC.attach(6, 1000, 2000);
    motD.attach(9, 1000, 2000);

    displayInstructions();
}

/**
 * Main function
 */
void loop() {
    if (Serial.available()) {
        data = Serial.read();

        switch (data) {
            // 0
            case 48 : Serial.println("Sending 0 throttle");
                      motA.write(0);
                      motB.write(0);
                      motC.write(0);
                      motD.write(0);

                      break;

            // 1
            case 49 : Serial.println("Sending 180 throttle");
                      motA.write(180);
                      motB.write(180);
                      motC.write(180);
                      motD.write(180);

                      break;

            // 2
            case 50 : Serial.print("Running test in 3");
                      delay(1000);
                      Serial.print(" 2");
                      delay(1000);
                      Serial.println(" 1...");
                      delay(1000);
                      test();

                      break;

            //3
            case 51 : Serial.print("power 20");
                      motA.write(20);
                      motB.write(20);
                      motC.write(20);
                      motD.write(20);

                      break;

            //4
            case 52 : Serial.print("power 40");
                      motA.write(40);
                      motB.write(40);
                      motC.write(40);
                      motD.write(40);

                      break;

            case 53 : Serial.print("Indpendance test");
                      motA.write(0);
                      motB.write(0);
                      motC.write(0);
                      motD.write(0);

                      delay(1000);

                      motA.write(20);
                      delay(2000);
                      motA.write(0);
                      delay(2000);

                      motB.write(20);
                      delay(2000);
                      motB.write(0);
                      delay(2000);

```

```

        motC.write(20);
        delay(2000);
        motC.write(0);
        delay(2000);

        motD.write(20);
        delay(2000);
        motD.write(0);
        delay(2000);

        delay(4000);

        motA.write(20);
        delay(1000);
        motB.write(20);
        delay(1000);
        motC.write(20);
        delay(1000);
        motD.write(20);
        delay(1000);

        delay(4000);

        motA.write(0);
        motB.write(0);
        motC.write(0);
        motD.write(0);

        break;

        case 54: Serial.println("Test de vol");
                vol();

        break;

        case 55: Serial.println("Calibrage esc");
                calibrage();

        break;
    }
}

/**
 * Test function sending angle to the ESCs from 0 to 180 degrees
 */
void test()
{
    for (int i=0; i<=180; i++) {
        Serial.print("Speed = ");
        Serial.println(i);

        motA.write(i);
        motB.write(i);
        motC.write(i);
        motD.write(i);

        delay(200);
    }

    Serial.println("STOP");
    motA.write(0);
    motB.write(0);
    motC.write(0);
    motD.write(0);
}

void vol()
{
    Serial.println("Bienvenu dans le module de test de vol, entrer 1 pour augmenter la puissance
ou 0 pour tout stopper");
    int puissance=0;
    Serial.println("La puissance actuelle est de 0");
    char saisi=0;

```

```

while (saisi != 48)
{
    motA.write(puissance);
    motB.write(puissance);
    motC.write(puissance);
    motD.write(puissance);

    if(Serial.available())
    {
        saisi = Serial.read();
        if (saisi == 49)
        {
            Serial.print("Vous avez saisi : ");
            Serial.println(saisi);
            puissance = puissance + 1;

            if( puissance == 180)
            {
                puissance = 0;
            }
            Serial.print("La nouvelle puissance des moteurs est : ");
            Serial.println(puissance);
        }
    }
    motA.write(0);
    motB.write(0);
    motC.write(0);
    motD.write(0);
    Serial.println("Moteur eteins");
}

void calibrage()
{
    Serial.println("Valeur maximal 180");
    motA.write(180);
    motB.write(180);
    motC.write(180);
    motD.write(180);

    delay(3000);

    Serial.println("Valeur minimale 0");
    motA.write(0);
    motA.write(0);
    motA.write(0);
    motA.write(0);

    delay(4000);
}

/**
 * Displays instructions to user
 */
void displayInstructions()
{
    Serial.println("READY - PLEASE SEND INSTRUCTIONS AS FOLLOWING :");
    Serial.println("\t0 : Sends 0 throttle");
    Serial.println("\t1 : Sends 180 throttle");
    Serial.println("\t2 : Runs test function");
    Serial.println("\t3 : Sends 20 throttle");
    Serial.println("\t4 : Sends 40 throttle");
    Serial.println("\t5 : Runs independant test");
    Serial.println("\t6 : Test de vol");
    Serial.println("\t7 : Calibrage\n");
}

```

Voila le code du fichier « Calibrage\_esc » disponible sur mon GitHub.

On y retrouve plusieurs fonctions bien distinctes :

- La fonction « setup » qui comme son nom l'indique va initialiser les moteurs avec la bibliothèque « servo.h ». (je ne l'ai pas inventé, je l'ai seulement adapté à mes moteurs).
- La fonction « test » qui va faire monter progressivement la puissance de TOUS les moteurs ensemble.
- La fonction « vol » que j'ai entièrement écrit (ce n'était pas non plus un exploit je l'admet) qui me permet de tester le drone en détails. Avec cette fonction j'ai pu constater notamment que les moteurs ne se mettent en route qu'à partir de valeurs supérieures approximativement à 20. Le code est facile à comprendre normalement.
- La fonction « calibrage » que j'ai également ajouté pour calibrer automatiquement les ESCs, je l'appellerai simplement lors du setup du drone et cela me permettra d'automatiser cette tâche. Le processus de calibrage est expliqué tout en haut du code en Anglais.

Chaque fonction est appelée via le moniteur série avec la fonction « displayInstructions ».

Voilà l'avancée pour le moment, je souhaiterai approfondir le code sur l'accéléromètre MPU6050 déjà disponible sur le GitHub, mais je manque cruellement de temps cette semaine. De plus je n'ai pas les pièces requises pour équilibrer physiquement le drone. Je le ferai donc lors de la prochaine séance. J'ai réalisé un test de vol à l'aide de la fonction « vol » qui s'est solvé par un échec relatif ; en effet, le drone vol à partir d'environ une puissance de 60-80 MAIS il part en vrille (réellement). Il faut donc absolument la fonction qui me permettra de compenser les defaults de poids et de puissance des moteurs à l'aide de l'accéléromètre pour faire un test de vol poussé.

J'ai d'ailleurs oublié de le préciser, j'utilise une batterie LiPo de 5500 mAh et de 11.1V que je connecte en parallèle à chaque moteur avec des connecteurs rapides. Je vais également créer une « Power Distribution Card » ou quelque chose qui s'en approche afin de pouvoir aisément mettre sous tension le drone, au lieu de tripoter des fils comme actuellement. La coque de protection rajoutera trop de poids sur le drone, je décide donc de ne pas l'implémenter. Je réaliserai plutôt mes tests avec le drone suspendu au plafond par un fil afin d'éviter toute chute.

Je n'ai pas eu de réel problème pour le moment, mise à part des petites erreurs de téléversement avec l'ordinateur portable que j'utilise.

Je filmerai un autre test dans la semaine afin de le mettre sur YouTube et ensuite sur GitHub.

Dans la prochaine séance j'expliquerai et essayerai de terminer la partie sur l'équilibrage avec accéléromètre.