

## Rapport séance 3 - Jade Morin

### Projet : Arbébéduino

**Objectif de séance :** Aujourd'hui je devais finir le programme des moteurs. En effet, le programme de la dernière séance ne permettait pas de faire tourner deux moteurs pas à pas simultanément. Mon objectif était donc d'utiliser la bibliothèque AccelStepper.

**Différence entre la bibliothèque AccelStepper et Stepper :** En lisant les forums, je me suis rendue compte que si je voulais faire tourner plusieurs moteurs avec un seul programme il fallait utiliser la bibliothèque AccelStepper qui est une bibliothèque non bloquante contrairement à Stepper.

En TD je suis donc arrivée aux programmes suivants :

#### Version 1 du voidloop():

```
Moteur_pas__pas3$
#include <AccelStepper.h>

AccelStepper moteur1(8, 9, 10, 11);
AccelStepper moteur2(2, 3, 4, 5);
/*AccelStepper moteur3{};*/

int i=200;

void setup() {
  Serial.begin(9600);
  /*on définit les vitesses des moteurs pour l'instant à vitesse égale*/
  moteur1.setSpeed(300);
  moteur2.setSpeed(400);
  /*moteur3.setMaxSpeed(300);*/

  /*Est ce qu'on a forcément besoin d'une accélération et d'une vitesse maximale ??*/
  moteur1.setAcceleration(1000);
  moteur2.setAcceleration(1000);

  moteur1.setCurrentPosition(0);
  moteur2.setCurrentPosition(0);
}

void loop() {
  while ((moteur1.currentPosition()<3000) && (moteur2.currentPosition()<3000))
  {
    Serial.println(moteur1.currentPosition());
    moteur1.moveTo(i);

    moteur2.moveTo(i);
    moteur1.run();
    moteur2.run();
    i+=200;
  }
}
```

## Version 2 du voidloop():

```
if (moteur1.distanceToGo() == 0){
  /* on veut faire remonter les objets*/
  position1=-position1;
  moteur1.moveTo(position1);
}
else{moteur1.run();}

if(moteur2.distanceToGo()==0){
  position2=-position2;
  moteur2.moveTo(position2);
}
else{moteur2.run();}

if(moteur3.distanceToGo()==0){
  position3=-position3;
  moteur3.moveTo(position3);
}
else{moteur3.run();}
```

1

Dans les deux cas l'aspect non bloquant de la bibliothèque AccelStepper devait permettre à mes 3 moteurs de fonctionner simultanément. Mais ce n'est pas le résultat que j'ai obtenu, en effet, un des moteurs tournait comme il fallait (en faisant le bon nombre de pas etc) alors que le deuxième moteur semblait vibrer sur place. Il n'arrivait pas à faire les pas que le programme lui demandait.

Après vérification des branchements et du bon fonctionnement des moteurs indépendamment l'un de l'autre, vous êtes arrivés à la conclusion que la bibliothèque AccelStepper n'était pas capable de supporter deux moteurs. Autrement dit, on ne peut pas fournir suffisamment d'alimentation pour que les deux moteurs tournent en même temps.

Vous vous êtes donc retournés vers la bibliothèque Stepper en faisant une boucle for, pour que les deux moteurs tournent ensembles. On a donc obtenu le programme suivant :

```
Moteur_pas__pas3_PM_bis $
#include <Stepper.h>
const int nombrePas= 32*64;
Stepper moteur1(nombrePas, 8, 9, 10, 11);
Stepper moteur2(nombrePas, 2, 3, 4, 5);

int i=200;

void setup() {
  Serial.begin(9600);
  /*on définit les vitesses des moteurs pour l'instant à vitesse égale*/
  moteur1.setSpeed(3);
  moteur2.setSpeed(3);
}

void loop() {

  for (int i = 0; i<10; i++)
    moteur1.step(10);
  delay(500);
  moteur2.step(10);
  delay(500);
  ...
}
```

Cependant même problème que précédemment, certes les deux moteurs tournent simultanément, mais un seul parvient à suivre le rythme, le deuxième reste bloqué en vibrant.

Après tous ces essais sans résultats concluant vous m'avez donc conseillé de reprendre le modèle du programme que vous aviez mis en ligne. Ainsi je ne devais plus être embêtée par une bibliothèque car votre programme commandait uniquement les moteurs à faire un pas puis l'autre grâce à une boucle for.

Donc en faisant faire à chacun de mes moteurs 1 pas dans ma boucle for, je devrais parvenir à les faire tourner "simultanément".

En réalité, l'enchaînement de la boucle for se fait si rapidement que l'on ne voit pas que les moteurs ne tournent pas parfaitement ensembles.

J'ai donc obtenu le début de programme Moteur\_pas\_\_pas3\_pm que je mets en ligne sur le Github. Je le terminerai pour l'oral. Pour l'instant la partie que j'ai codé permet de faire tourner les moteurs 1 et 2 dans des sens contraire et pour 10 pas.

**A faire avant la prochaine séance:** Je vais donc devoir coder les pas du troisième moteur, faire "descendre" 2 moteurs sur 3 puis les faire "remonter". Il me faut donc écrire une deuxième boucle for.