AZZOPARDI

Thomas

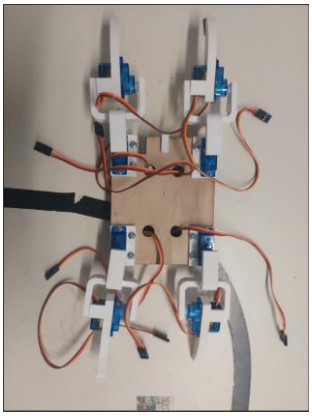
PeiP2 Groupe 3

RAPPORT

1/ L'assemblage.

Durant cette séance, mon partenaire et moi avons pu enfin terminer de récolter toutes les pièces nécessaires à l'assemblage. Nous avons pu assembler les pièces pour avoir un réel quadrupède à manipuler.

Il a fallu rajouter et visser toutes les pates et emboiter les servo-moteurs et notre version finale a pu être assez compacte.



2/ Difficultés :

En faisant des tests sur les servo-moteur (position de base,...) Nous nous sommes rendu compte que notre carte Nano ne fournirai pas assez de courant pour tout les servos-Moteurs. En effet ceux-ci s'arrêtaient dès la 3ème rotation demandée. Néanmoins sur un Arduino Uno, ceux-ci marchaient correctement. Il faudra donc : soit utiliser un Arduino Uno mais celui-ci sera sur élever car il ne rentre pas sur notre « plateforme » centrale, soit rajouter une alimentation sous notre Arduino Nano pour donner assez de courant.

3/ Les tests

J'ai enfin pour faire les 1er test sur un vrai quadrupède. Après avoir finaliser tout les branchement sur l'avant, j'ai pensé a crée la fonction moveF qui permettra d'avancer. Cette fonction à été améliorée et optimisé. J'ai aussi pu utiliser des fonction tels que DeplacementAr et DeplacementAv qui permettront de faire bouger un servo-Moteur mais avec un « temps de rotation » defini afin que la rotation faite ne soit pas trop brutale.

A l’aide de l’algorithme suivant, j’ai pu créer la fonction moveF :

idée pour pousser les membres avant :

droite monte -> decale -> se pose

gauche monte -> décale

droite -> pousse

gauche -> se pose

droite monte -> angle ini

gauche pousse

droite pose

gauche monte -> angle ini -> se pose

Cet algorithme sera certainement changé plus tard.

Enfin j’ai aussi créé la fonction coucou() qui reponds au cahier des charges avec « possibilité de faire un coucou ».

Il a aussi fallu tout calibré : lorsque le servo-Moteur (SM) fait une rotation de +45° par rapport à notre angle initial (90°) il va vers la droit ou vers la gauche selon ou il est placé :

pour le avant droit Bas:+ = vas vers le bas

pour le avant droit Haut:+ = vas vers la gauche

pour le avant gauche Bas :+ = vas vers le bas

pour le avant gauche Haut:+ = vas vers la gauche

De plus il faudra réfléchir si les pates accrochent assez.

Bonus : voici le code écrit fonctionnel écrit:

```
void DeplacementAv(Servo moteur,int time,int Ini,int angle){
  for (int i = Ini; i<Ini+angle;i++){
    moteur.write(i);
    delay(time);
  }
}
```

```
void DeplacementAr(Servo moteur,int time,int Ini,int angle){
  for (int i = Ini; i>Ini-angle;i--){
    moteur.write(i);
    delay(time);
  }
}
```

```
void moveF(Servo moteur1, Servo moteur2 ,Servo moteur3, Servo moteur4){            // fonction pour aller vers l'avant
  DeplacementAr(moteur1,15, angle_Ini,45);
  DeplacementAv(moteur2,15, angle_Ini,45);
  DeplacementAv(moteur1,15,angle_Ini-45,45);
  delay(15);    // droite monte -> decale -> se pose
  DeplacementAr(moteur3,15, angle_Ini,45);
  DeplacementAr(moteur4,15, angle_Ini,45);
  delay(15);//gauche monte -> décale
  DeplacementAr(moteur2,15,angle_Ini+45,75);
  delay(15);//droite pousse
  DeplacementAv(moteur3,15,angle_Ini+45,45);
  delay(15); //gauche pose
  DeplacementAr(moteur1,15, angle_Ini,45);
  moteur2.write(angle_Ini);
  delay(15); //droite monte -> angle ini
  DeplacementAv(moteur4,15,angle_Ini+45,75);
  delay(15); //gauche pousse
  DeplacementAv(moteur3,15,angle_Ini+45,45);
  delay(15); //droite pose
  DeplacementAr(moteur3,15, angle_Ini,45);
  DeplacementAr(moteur4,15, angle_Ini,30);
  DeplacementAv(moteur3,15,angle_Ini+45,45);
  delay(15); //gauche monte -> angle ini -> se pose
  moteur1.write(angle_Ini);
  moteur2.write(angle_Ini);
  moteur3.write(angle_Ini);
  moteur4.write(angle_Ini);
}
```