

Rapport du TP ANAD

BRAS ROBOTIQUE JOUEUR D'ECHECS

ANAD S2 2CS

Rédigé par :

BAZINE Tarek

CHIKH AISSA Mahfoud

2CS _SIL

Encadré par :

M. AIT ALI YAHIA Y

Table des matières

Introduction	2
Conception Mécanique :	3
Les mouvements :	5
La main (gripper) :	5
Montage électrique :	6
Principes de fonctionnement	7
Construction de base de connaissance :	7
Commander la carte-ARDUINO :	7
Commander les SERVO-moteurs :	9
Conclusion	10

Introduction

Le monde de robotique et d'intelligence artificiel est un monde très intéressant, pour cela on a pensé de faire nos premières expérimentations en réalisant un bras robotique qui puisse jouer les échecs.

L'idée ici n'est pas de construire un bras manipulateur de qualité industrielle, mais plutôt de bricoler un premier prototype simple qui nous permettra d'explorer le concept.

Bien sûr, l'expérience acquise lors de la création de ce dispositif modeste nous aidera probablement à concevoir de bras plus ambitieux dans l'avenir.

Conception Mécanique :

En inspirant du projet de M. Uladz qui a fait un bras robotique similaire à ce qu'on a valu réaliser, on a mis un premier plan contient les pièces de base qui forment notre bras (lien : <http://www.instructables.com/id/CARDBIRD-the-Cardboard-Robotic-Arm/>)

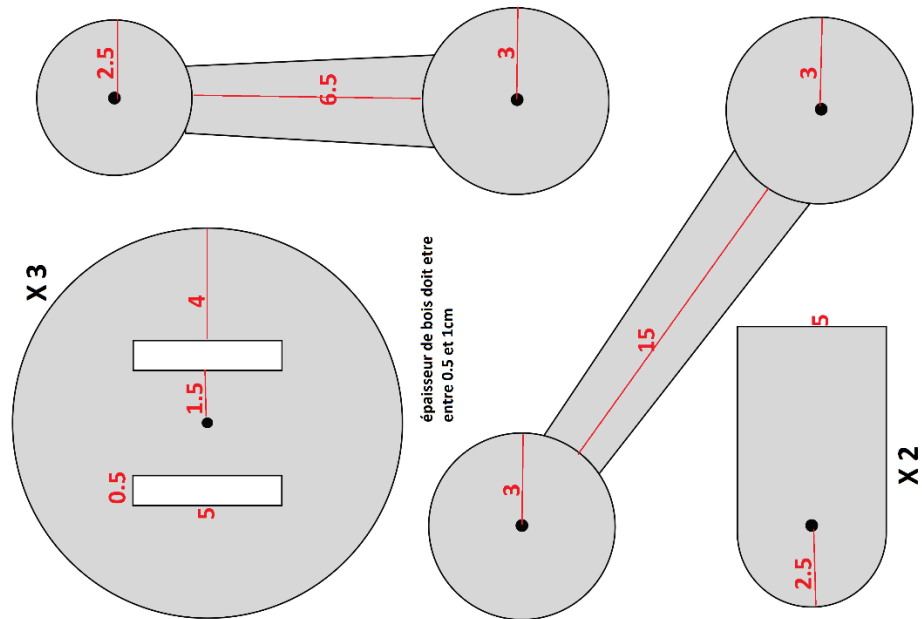


Figure 1 pièces de base du premier modèle

Ces pièces seront assemblées pour construire le modèle suivant :

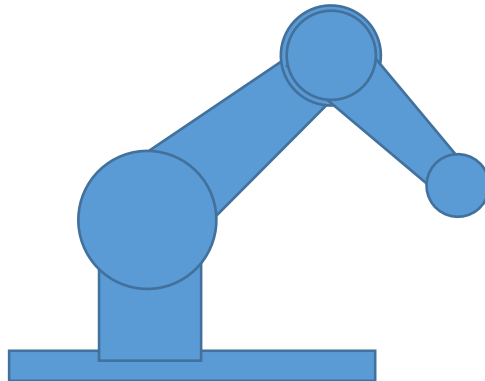
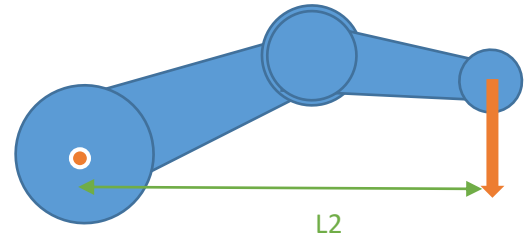
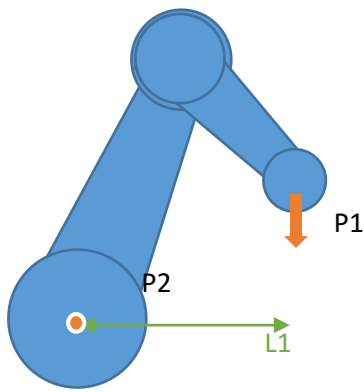


Figure 2 premier modèle sans main (gripper)

Contrairement à M. Uladz qui a utilisé le carton, on a opté au bois comme matière première. Pour construire un bras plus sophistiqué. Le modèle précédent est très sensible au poids, et même si qu'on a mesuré le poids des pièces qui était largement inférieur au poids max supporté par les servomoteur SG90 et MG90s, on n'a pas tenu compte que le poids levé augmente proportionnellement avec l'augmentation de la longueur (distance entre le moteur et l'extrémité de morceau levé) du bras –principe des leviers- (figure 3).



$$L1 < L2 \text{ (distance)} \implies P1 < P2 \text{ (poids)}$$

Figure 3 Principe des leviers

Un autre défaut rencontré dans la conception initiale, c'est que le servomoteur situé au point A lève un poids très important par rapport aux autres servos.

Les contraintes précédentes nous ont fait améliorer le plan d'une façon supportable par les servo utilisés.

La nouvelle conception :

pour minimiser le poids on a utilisé un plastique brusque au lieu du bois et on a reparti le poids levé sur les deux moteurs m1, m2 (figure 4)

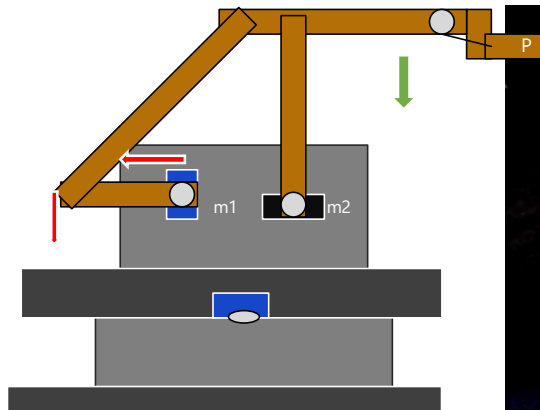


Figure 4 Nouveau Plan

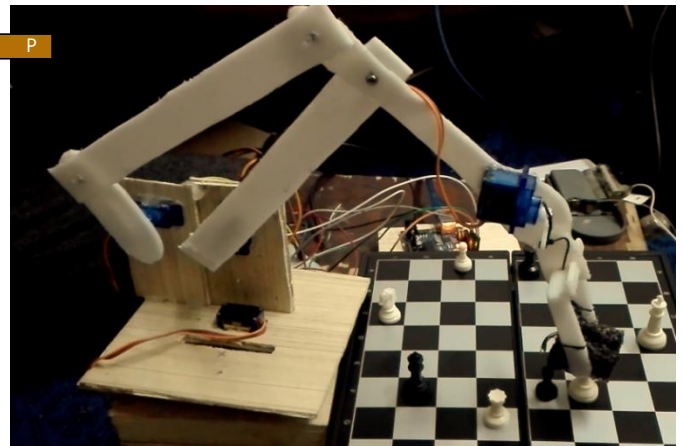


Figure 5 le bras réel

Les mouvements :

Le bras est composé de 4 servos moteurs qui tournent en angles de rotation $[0^\circ, 180^\circ]$ pour faire des mouvements appropriés au jeu d'échecs, les quatre angles de rotation sont illustrés dans la figure 5.

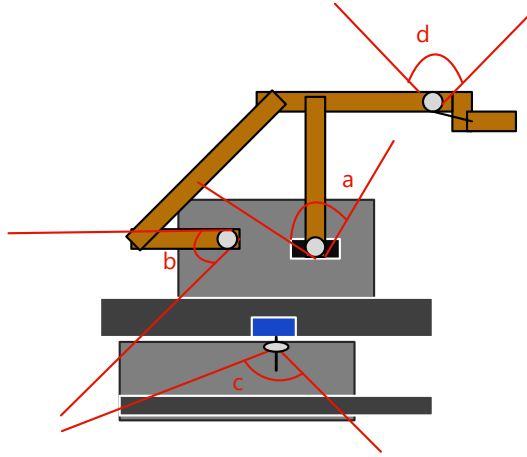


Figure 6 Angles de rotation

La main (gripper) :

Pour la main(gripper) l'aspect poids est très important vu qu'elle se trouve à l'extrémité du bras, d'où la simplicité est demandée (figure7).

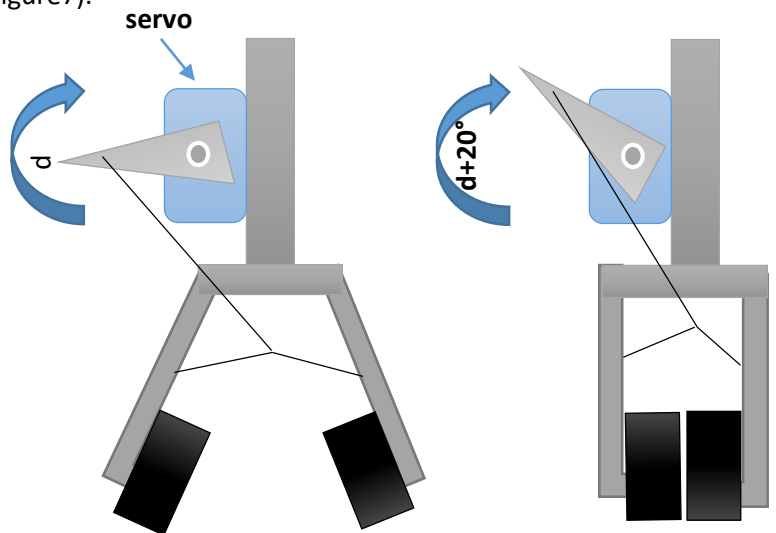


Figure 7 Trigger

Les pièces en noire ce sont des morceaux d'éponge pour que la main capte bien les pièces.

Montage électrique :

Pour éviter la surcharge en terme d'alimentation électrique on a opté à alimenter les quatre moteurs par un chargeur de téléphone portable qui donne en sortie 1000Ma et 5v, en effet les servos moteurs supportent de 4.5v à 6v, donc on n'a pas trouvé des problèmes de manque ou de débordement d'alimentation.

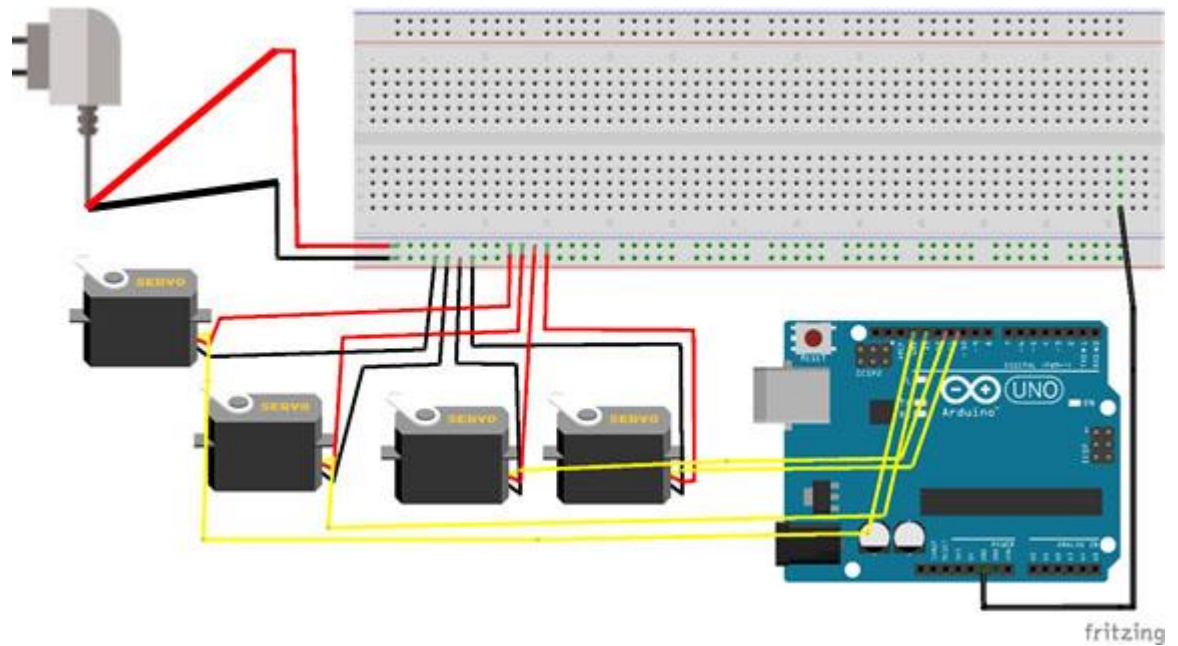


Figure 8 Montage électrique

Principes de fonctionnement

Afin que notre bras puisse fonctionner, on a été obligé de faire une architecture logicielle et hardware qui s'adapte à notre cas moyennant utilisation d'un ordinateur pour contrôler la carte-ARDUINO et l'utilisation d'une base de connaissance dans laquelle on sauvegarde les angles de rotation de bras qui seront transmis à la carte-ARDUINO sous forme de signaux.

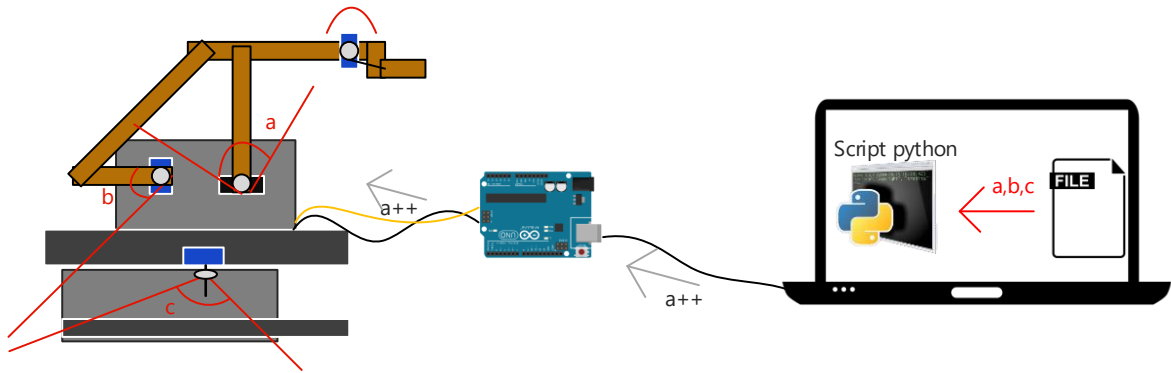


Figure 9 Architecture Logiciel et Hardware

On a décomposé notre architecture en trois parties :

Construction de base de connaissance :

Cette partie consiste à faire un programme qui nous donne la main pour contrôler le bras avec des touches de clavier, et avec des touches spécifiques, on lance l'enregistrement de ces mouvements dans une base de données sous forme d'un fichier, pour cela, on a utilisé le langage JAVA avec toutes les bibliothèques nécessaires comme la bibliothèque Serial pour la communication par USB (le code source est dans l'Annexe prog.pde)

L'ingénierie : comment sauvegarder tous les mouvements possibles d'un échiquier dans un fichier ? est-ce qu'on s'amuse de sauvegarder tous les mouvements possibles de chaque case vers les autres ? Non, ce n'est pas une solution applicable à cause du grand nombre de possibilités des mouvements (chemins) d'une case, c'est dans la grandeur de 8^8 chemins !, donc on a bien conçu une technique pour remédier à ce problème. Cette technique est inspirée d'une machine à état, on précise un point initial (état 0) pour le bras, ensuite on sauvegarde dans le fichier tous les mouvements de ce point initial vers chaque case de l'échiquier donc juste $8*8=64$ mouvements, et on n'a même pas besoin de sauvegarder le mouvement de retour c'est juste on le déduit, selon l'état du bras en changeant la manière d'exploiter le fichier pour extraire le chemin du mouvement (exemple d'état : bras libre –état 0-, bras en mouvement pour prendre une pièce –état 1-, bras en mouvement pour jeter une pièce –état 2- ...).

Commander la carte-ARDUINO :

Après la génération de fichier de mouvements dans l'étape précédente, on a arrivé à l'étape d'exploitation pour cela on a utilisé le langage python vu que sa puissance dans le traitement des données et dans notre cas là lecture des angles d'après le fichier.

Ce programme s'exécute en ligne de commande, il attend l'entrée d'utilisateur sous forme de mouvement (par exemple : déplacer la pièce du case A1 vers la case A8 s'écrit **a1a8**, Enlever la pièce du case C5 s'écrit **c5**) ensuite il les valide, s'ils sont des mouvements valides le programme va prendre les informations d'après le fichier ensuite il envoie les signaux vers la carte-ARDUINO sur le port USB dans lequel la carte est branchée. (le code source est dans l'Annexe cmd.py)

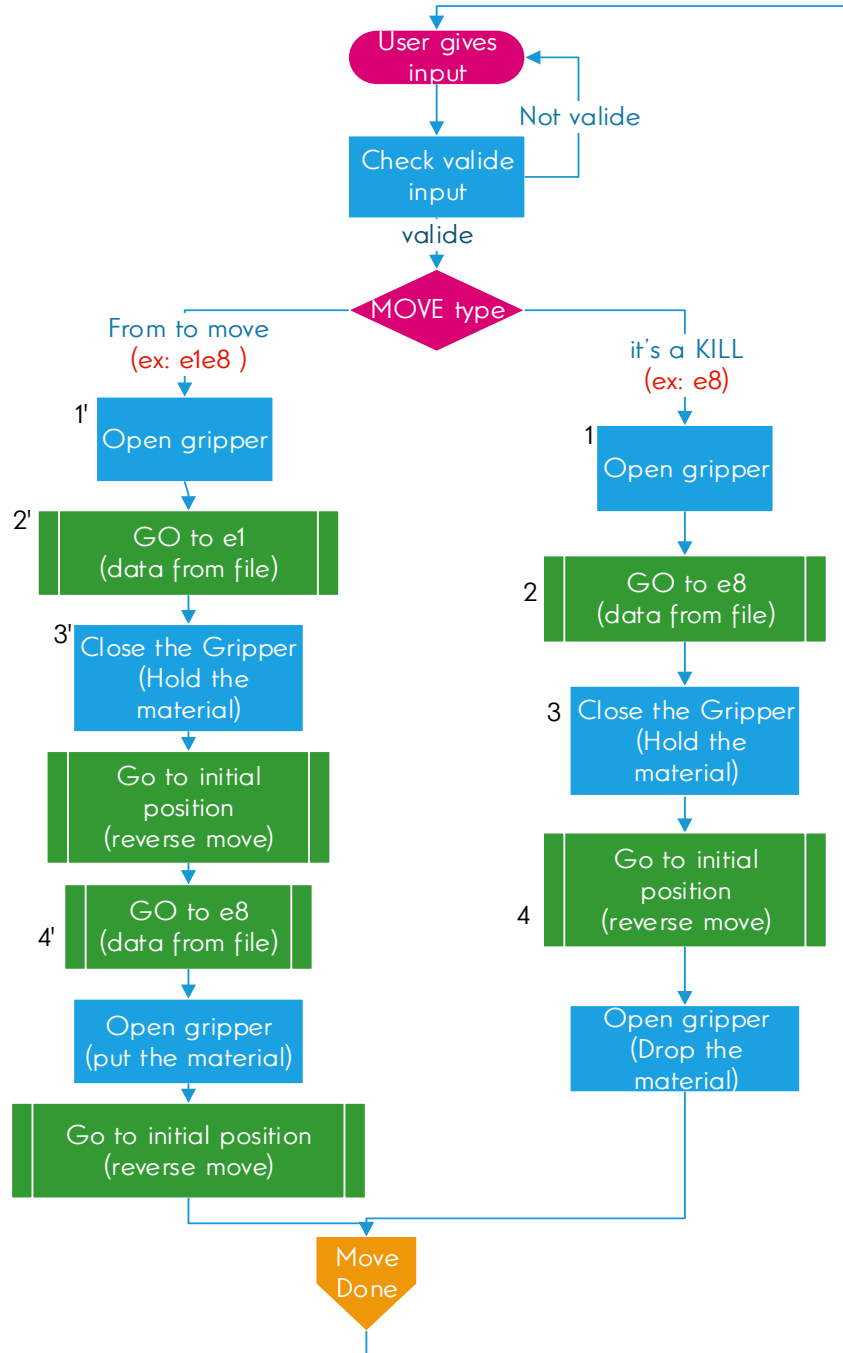


Figure 10 Digramme illustre le fonctionnement du programme

Commander les SERVO-moteurs :

Pour que la carte-ARDUINO puisse commander les SERVO-moteurs, il faut flasher un simple programme qui attend des signaux sur l'interface serial ensuite ils les transforment comme des commandes pour les moteurs.

```
// Inclusion des bibliotheques
#include <Servo.h>

// Declaration servo variables
Servo plattform;
Servo first_arm;
Servo second_arm;
Servo grip;

void setup()
{
  // Definition de port SERIAL
  Serial.begin(9600);

  // Attachement des variables servo a des pin d'ARDUINO
  plattform.attach(9);
  first_arm.attach(8);
  second_arm.attach(7);
  grip.attach(6);

  // Initialisation des angles pour les variables servo
  plattform.write(90);
  first_arm.write(90);
  second_arm.write(90);
  grip.write(120);
}

void loop()
{
  // This part is for computer control
  if(Serial.available()>0){

    // Lecture du signal sous form d'un nombre
    int com = Serial.read();

    switch(com){
      // Bouger la plattform (angle) d'un degre
      case 1: plattform.write(plattform.read()+1);
      break;
      case 2: plattform.write(plattform.read()-1);
      break;
      case 3: first_arm.write(first_arm.read()+1);
      break;
      case 4: first_arm.write(first_arm.read()-1);
      break;
      case 5: second_arm.write(second_arm.read()-1);
      break;
      case 6: second_arm.write(second_arm.read()+1);
      break;
      case 7: grip.write(grip.read()-20);
      break;
      case 8: grip.write(grip.read()+20);
      break;
    }
    delay(50);
  }
}
```

Conclusion

Cette expérience nous amenait à découvrir le monde de robotique dont on a pris des nouveaux langages de programmation et plein d'autres connaissances.

Notre bras a puis fait des mouvements complets, mais le chemin est assez long pour garantir l'exactitude et l'ajout de l'intelligence artificielle, en attendant celui qui va s'inspirer de notre travail et avancer dans l'objectif final, un bras performant et intelligent.