

# Rapport n°6

## Définition des références et appellations :

Les jambes sont numérotées de 0 à 3 (*Figure 1*) et sont composées d'une épaule et d'un coude (*Figure 2*). Pour la valeur des angles, on prend comme référence les lignes en pointillé.

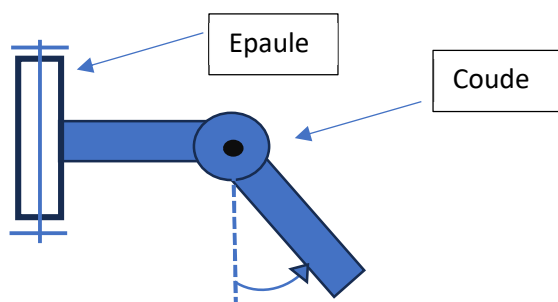


Figure 2: Nommage des parties d'une jambe

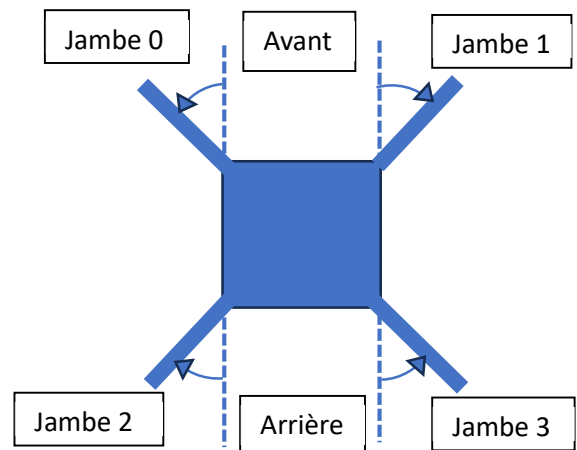


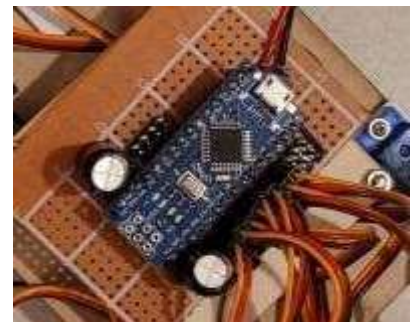
Figure 1: vue de dessus

## Problèmes d'alimentation :

Lors de séance précédente, nous n'arrivions pas à faire fonctionner correctement les servomoteurs avec l'alimentation de l'ordinateur. La raison était que l'ordinateur ne fournissait pas assez de courant, on a donc utilisé une alimentation secteur ce qui nous a réglé le problème.

Néanmoins, pour des raisons obscures lors de cette séance c'était l'inverse. Avec le professeur nous avons donc rajouté des condensateurs en parallèle de l'alimentation pour supprimer d'éventuels perturbations. On a constaté une amélioration mais les bugs subsistés.

Ce problème s'est résolu lors de l'utilisation des batteries 18650 que nous voulions utiliser la séance dernière mais qui n'était pas en stock. Finalement, nous ne connaissons pas l'origine du problème. Mais il n'a pas de conséquence sur le projet puisque nous comptons le faire fonctionner sur batterie.



### Réalisation du support pour le capteur de distance :

Afin de fixer notre capteur de distance laser sur son servomoteur il nous fallait un support. J'ai réalisé le support à partir d'une impression ratée d'un « U » (*Figure 4*) que j'ai découpé pour obtenir un « L » (*Figure 3*).

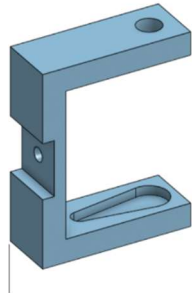


Figure 3: Pièce en "U"



Figure 4: Support capteur

Ensuite, pour fixer le capteur j'ai utilisé le trou fait pour la fixation mi-bois et une vis (*Figure 5*).

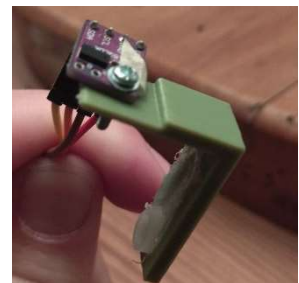


Figure 5: support capteur 2

### Familiarisation avec le capteur de distance laser :

Pour mieux comprendre le fonctionnement du capteur de distance laser j'ai lu une partie de sa doc associée.

J'ai appris qu'il communiquait avec la carte par une liaison I2C. Et j'ai trouvé une librairie me permettant de facilement me connecter avec et de prélever une mesure. Il me reste encore à vérifier si les valeurs retournées sont correctes. Dans le cas contraire, je vais devoir me pencher sur une calibration du capteur.

### Réflexion sur la détection d'obstacle :

L'un des objectifs du projet est que le quadrupède puisse passer des obstacles. Pour cela, il doit les détecter et mesurer leur hauteur pour juger s'il est infranchissable ou non.

La première étape est de le détecter :

L'idée que j'ai eu est de comparer une valeur retournée par le capteur et une valeur théorique. Comme nous connaissons la hauteur du robot, le déport

du capteur (constante  $h$  et  $l$ ) et l'angle que prend le capteur (angle  $\phi$ ), il devient facile de calculer une valeur théorique en utilisant de la trigonométrie.

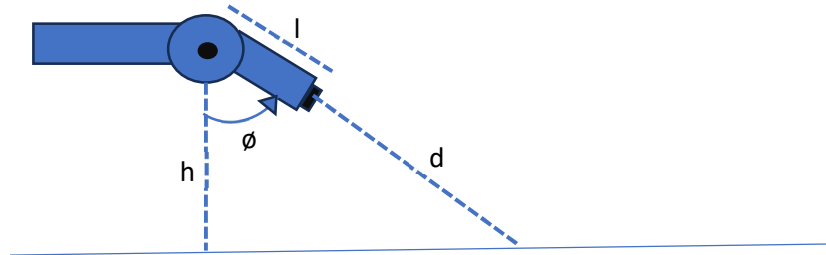


Figure 6: Valeur théorique

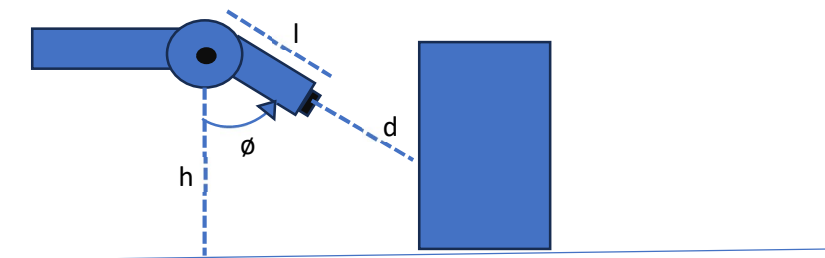


Figure 7: Mesure de la valeur

Dans un monde idéal, si la valeur mesurée n'est pas égale à la valeur théorique alors il y a un obstacle.

Mais dans notre monde, il faut prendre en compte que le capteur peut avoir des erreurs de mesures et que notre quadrupède a de petite différence de position.

Pour palier à ça, on va rajouter une tolérance  $\epsilon$  :  $|d_{\text{mesure}} - d_{\text{théorie}}| < \epsilon$ .

Avec cette méthode on perd la détection d'obstacle de petite taille, nous devront donc faire attention que cette tolérance ne soit pas trop grande pour toujours détecter des obstacles de tailles non négligeables.

Pour déterminer la hauteur d'un obstacle, je me pencherai sur la question lors de la prochaine séance.