

## ARAGOG Rapport #4

La séance d'aujourd'hui m'a servi à finir de prendre en main le module SSC-32U pour lui transférer des données de mon arduino.

Pendant les vacances, j'avais essayé de téléverser des informations de mon arduino vers le module en m'inspirant du code donné sur le site : <http://marc-tetrapod.blogspot.com/2012/10/arduino-ssc-32-servo.html> , sans succès. Il y avait 2 principaux problèmes :

1. Le code donné était fait de telle sorte à envoyer des informations directement au module par le câble fourni avec, seulement, je ne peux pas faire ça car Mr Masson nous a « interdit » de nous servir des ports 0 et 1 utilisés dans ce cas pour communiquer directement avec le module. J'ai donc utilisé un programme réalisé lors du TD sur la communication RF en restant quand même dans le flou puisque ce module n'est pas destiné à nous répondre, et mon code n'avait pas l'air de fonctionner puisque rien ne bougeait.
2. Dans le code donné, l'initialisation : `Serial.begin` était à une fréquence de téléversement de 115200 , j'ai donc pensé que je devais initialiser mon code avec une fréquence de 115200, mais comme j'ai créé un objet de type `SoftwareSerial`, que je devais également initialiser, je ne savais pas le quel mettre à combien.

Première chose que j'ai faite en arrivant : j'ai scotché mes deux plaques (ma plaque arduino et ma plaque SSC-32U) à une couche d'isolant, puisque mon araignée étant en fer, cela risquait de créer des courts-circuits.

Puis, comme après avoir tenté beaucoup de choses pour communiquer avec le module (en y ayant passé bien 6h un jour de vacances) les servo-moteurs ne bougeaient toujours pas, j'ai dû demandé de l'aide à Mr Masson, qui a tout d'abord constaté avec moi que lorsque je téléversait, aucune tension n'était donnée aux servo-moteurs, et pour cause, nous avons pu constater que deux des 4 piles (de provenance douteuse) fournissait une tension négative, ce qui ne laissait qu'un volt pour alimenter les quelques 12 servo-moteurs. Après avoir changé de piles, pour qu'elles ne fournissent pas trop de courant, j'ai débranché tous les servo-moteurs en en laissant un pour pouvoir effectuer des tests plus facilement.

J'avais un doute, le code qui doit être envoyé au module doit être en ASCII 13 et doit retourner à la ligne une fois la commande passée. Or, j'utilisais la commande `SSC.write()`, et je ne savais pas si elle faisait les deux requêtes précédentes. Grâce à l'aide de Mr Masson, je sais maintenant qu'il faut utiliser la commande `SSC.println()` (je ne savais pas que ça se faisait). Depuis le début des vacances, je me familiarise avec le langage utilisé pour envoyer des commandes vers le module, j'ai donc pu, une fois la communication opérationnelle, les appliquer. J'ai eu le plaisir de constater que les commandes envoyées fonctionnaient parfaitement (sur un servo-moteur).

J'ai donc essayé d'initialiser le mouvement d'une patte selon le modèle :

MOTEUR VERTICAL	MOTEUR HORIZONTAL
Down → Mid	Back
Mid → Up	Back → Mid
Up	Mid → Front
Up → Mid	Front
Mid → Down	Front
Down	Front → Mid
Down	Mid → Back
Down	Back

Ce qui n'a pas été très concluant...

En effet, je peux effectuer tous les mouvements que je demande sur un servo-moteur, voire les deux servo-moteurs d'une même patte, mais lorsque je commence à demander le mouvement de deux pattes, il y a quelques petites erreurs de coordination. De plus j'ai l'impression que la masse du haut de la patte fait tomber toute la patte (qu'elle fait donc tourner le servo-moteur du milieu), ce qui fait forcer sur ce servo-moteur, et je ne peux pas régler ce problème car j'effectue mes tests en surélevant l'araignée pour ne pas faire forcer sur les servo-moteur du bas pour la lever.

Je vais donc étudier le mouvement plus en détail et le perfectionner, et écrire un code à cet effet. C'est pour cela que je ne joins pas encore de code car il est actuellement en écriture.

**Mon But pour la prochaine séance :**

Réussir à faire faire ses premiers pas à ARAGOG (du 4 pattes bien-sûr), ce qui s'avère compliqué et finir le cahier de charges.

**THE END**