

**FABRICATION DE LA TETE D'UN ROBOT
HUMANOIDE A TAILLE HUMAINE EN
IMPRESSION 3D AVEC MISES EN OEUVRE
ELECTRONIQUE ET LOGICIELLE**



Etudiants :

Alexia BELHAIRE Perrine CEZERA
Colin LOCOGE Léana PERDRIAU
Arnaud QUILLENT Auguste RAMS

Enseignant-responsable du projet :
Fabrice DELAMARE

Date de remise du rapport : 19/06/2017

Référence du projet : STPI/P6/2017 – 25

Intitulé du projet : *Fabrication de la tête d'un robot humanoïde à taille humaine en impression 3D avec mise en œuvre électronique et logicielle*

Type de projet : *Expérimental*

Objectifs du projet: Le but de ce projet est d'imprimer des pièces à l'aide de l'imprimante 3D Ultimaker 2+, puis de les assembler afin de fabriquer une tête de robot humanoïde. A terme le robot doit pouvoir dialoguer, hocher la tête ainsi que bouger les yeux en utilisant des servomoteurs. Pour se faire, nous avons à disposition le site inMoov, ainsi que le matériel présent dans la salle de l'association de robotique de l'INSA.

Mots-clefs du projet :

- Robotique
- Programmation
- Impression 3D

TABLE DES MATIÈRES

Introduction	6
Méthodologie / Organisation du travail	6
Travail réalisé et résultats	7
Impression	7
Cura	7
Ultimaker 2+	7
Problèmes rencontrés	9
Electronique	10
Synthèse vocale : Arduino, Acapela Speech	10
Servomoteurs	11
Problèmes rencontrés	12
Assemblage	13
Bibliographie	17
Annexes	18
Documentation technique	18
6.1.1 Tutoriel synthèse vocale	18
6.1.2 Tutoriel des servomoteurs manuels	21
Listings des programmes réalisés	22
6.2.1 Programme MrlCommModifié	22
Schémas de montages	25
6.3.1 Branchements des servomoteurs	25
Propositions de sujets de projets (en lien ou pas avec le projet réalisé)	26

NOTATIONS, ACRONYMES

Tout au long de ce rapport, les termes suivants seront marqués par un astérisque :

STPL: Sciences et Techniques Pour l'Ingénieur

ASI: Architecture des Systèmes d'Information

ARI: Association de Robotique de l'INSA

CTL: Culture Technique de l'Ingénieur

1. INTRODUCTION

Dans le cadre de notre projet de P6 en STPI2*, nous avons décidé de nous pencher sur le projet « Fabrication de la tête d'un robot humanoïde à taille humaine en impression 3D avec mise en œuvre électronique et logicielle ». Ce projet comporte une partie informatique et une autre mécanique correspondant ainsi aux thématiques ASI* et MÉCA. Ce projet correspondait donc à notre cursus puisque nous avons tous choisis au moins une de ces thématiques. De plus, nous avons été attiré par son originalité puisque nous n'avons jamais fait de robotique auparavant.

Lors de la première séance, Fabrice Delamare, notre responsable de projet nous a présenté le sujet et les objectifs à atteindre ; il ne s'agit pas seulement d'imprimer une tête de robot humanoïde mais également d'arriver à la faire parler et bouger. Pour cela, nous avons travaillé dans la salle d'ARI* afin de pouvoir imprimer nos pièces et utiliser l'atelier pour l'assemblage de ces dernières.

2. MÉTHODOLOGIE / ORGANISATION DU TRAVAIL

Les premiers créneaux de P6 ont été des séances de découverte. Nous nous sommes dans un premier temps familiarisés avec le site inMoov et nous avons appris à utiliser l'imprimante 3D.

De plus, il s'agissait d'organiser notre travail pour la suite des semaines, avec par exemple le choix des pièces à imprimer en priorité (que nous avons numérotées dans l'ordre) , pour pouvoir commencer le montage au plus vite. Nous avons également pris en compte le temps d'impression des pièces les plus longues, car certains jours, l'imprimante était utilisée après nous et nous devons donc la libérer.

Nous nous sommes tous occupés de l'impression des pièces et nous passions en dehors des heures de P6 à tour de rôle afin d'en imprimer un maximum. Sur la fin de notre projet et suite à des problèmes d'impression rencontrés sur l'Ultimaker 2+, nous avons utilisé l'imprimante 3D disponible dans le bâtiment Dumont d'Urville pour accélérer l'impression des dernières pièces.

Au long des séances, nos rôles étaient polyvalents, nous nous répartissions par groupe de 2 ou 3 sur différents ateliers :

- Impression et organisation des futures pièces à imprimer
- Travail sur les pièces (numérotation, ponçage, perçage)
- Assemblage
- Travail sur les servomoteurs
- Synthèse vocale
- Rédaction du compte rendu de chaque séance

Au fil du temps la répartition s'est faite naturellement avec généralement, à la partie

technique/manuelle Léana, Alexia et Auguste et à la partie électronique Perrine, Colin et Arnaud.

3. TRAVAIL RÉALISÉ ET RÉSULTATS

3.1. Impression

3.1.1. Cura

Nous avons récupéré les structures des pièces à imprimer sur le site inMoov, en format stl, puis nous les avons converties en format GCODE, compatible avec l'imprimante, à l'aide du logiciel Cura.

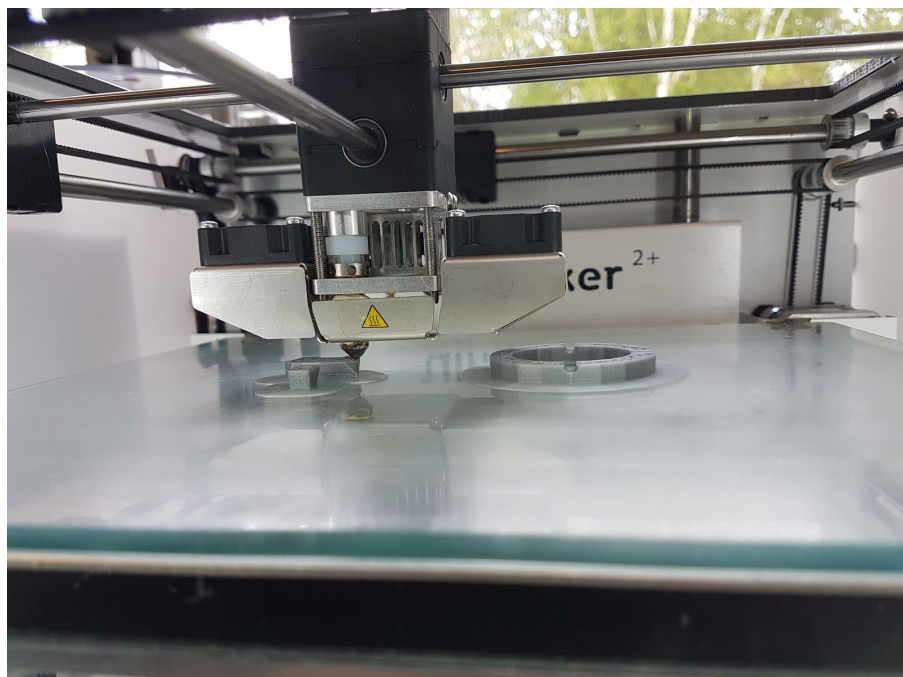
Cura est un logiciel open-source créé et maintenu par la société Ultimaker, fabricant d'imprimantes 3D. Il permet de convertir des fichiers 3D sous divers formats (.3MF, .obj et .stl) en fichier GCODE. Le GCODE est le langage implémenté dans les machines-outil à commande numérique utilisées entre autres en CTI* 1.2 (STPI* 1) pour graver un jeton de caddie.

Cura propose de nombreuses options afin de paramétrer au mieux l'impression : taille de la buse (0,4 et 0,6 mm pour notre projet), matériau utilisé, qualité d'impression (rapide ou optimale), remplissage de la pièce (creuse, structure clairesmée ou dense) ainsi que l'utilisation ou non de supports pour les pièces aux géométries spéciales. Nous avons également à plusieurs reprises ouvert deux pièces simultanément dans le le logiciel afin de les regrouper et de limiter le nombre d'impressions.

Le logiciel procède alors à la découpe de la pièce en tranches dont l'épaisseur dépend de la qualité d'impression. Le fichier ainsi créé est alors prêt à être enregistré sur une carte SD et transféré sur l'imprimante.

3.1.2. Ultimaker 2+

*Impression de pièces, effectuée par l'imprimante ultimaker 2+ de l'ARI**



L'imprimante 3D que nous avons utilisée tout au long du semestre était le modèle Ultimaker 2+.

Son principe de fonctionnement est le suivant : le fichier de la pièce à imprimer est importé via une carte SD. Ensuite, lorsqu'on lance l'impression, le fil est chauffé et des couches successives de matériau sont déposées sur le plateau afin de créer la pièce. La buse est déplacée à l'aide de deux rails horizontaux afin d'ajuster les coordonnées horizontales du fil, et le plateau monte et descend afin de se trouver à bonne hauteur d'impression de la pièce.

Lors de la toute première utilisation, il est important de paramétrer par défaut l'imprimante. Ainsi, nous avons défini l'épaisseur de la buse (qui a été changée en cours de projet, passant d'une buse de 0.4mm à 0.6mm). Nous avons également déterminé la hauteur entre le plateau et la buse lors des impressions, que nous avons mis à 1mm d'épaisseur, à l'aide d'une feuille de papier.

Pour lancer une impression, il suffit de déposer le fichier de la pièce en format GCODE (préalablement converti avec le logiciel Cura, comme expliqué précédemment) sur la carte SD de l'imprimante, puis de lancer l'impression à l'aide de la fonction "*print*" (imprimer).

Cependant, il faut s'être préalablement assuré que la bobine de matériau est bien insérée et qu'elle dispose de suffisamment de matière pour réaliser la pièce, sans oublier de vérifier qu'il s'agit bien du matériau (et éventuellement de la couleur) souhaité. Si le matériau n'est pas celui désiré, il suffit de demander à l'imprimante de l'éjecter (fonction "*remove material*"), puis de le remplacer par une nouvelle bobine en plaçant l'extrémité du fil à l'arrière de la machine, dans un interstice prévu à cet usage, et de demander à la machine de l'installer (fonction "*set up*").

Une fois l'impression lancée, il est possible de modifier certains paramètres, tels que la température de la buse ou celle du plateau, ou encore la vitesse de réalisation de la pièce, qui dépendent principalement du matériau utilisé. En effet, la température idéale pour le PLA est estimée à environ 180°C, bien que nous réglions généralement celle-ci à 200°C afin d'obtenir une sortie de matériau plus fluide.

De plus, il est possible d'utiliser certaines fonctionnalités ou de régler certains paramètres sans avoir lancé d'impression. En effet, on peut par exemple monter ou descendre le plateau, chauffer la buse ou le plateau...

L'adhérence de la pièce sur le plateau lors de l'impression 3D est censée être automatique, cependant nous avons remarqué qu'il était préférable d'ajouter une matière adhésive sur le plateau, comme de la colle, ou encore plus efficace : de la laque. Nous avons plusieurs fois utilisé un cutter afin de décoller la pièce après son impression, pour ensuite la récupérer. Nous ne risquons pas de l'abîmer car celle-ci comporte un support à décoller prévu à cet effet.

Cependant, comme nous n'avons pas le temps d'imprimer toutes les pièces nécessaires à la réalisation de notre robot, nous avons également utilisé les imprimantes accessibles aux STPI, se situant en Dumont Durville lors des trois dernières semaines. Néanmoins, nous n'étions pas autorisés à lancer directement l'impression, c'est pourquoi nous avons uniquement envoyé les fichiers à imprimer, et nous passons récupérer les pièces plus tard.

3.1.3. *Problèmes rencontrés*

Tout au long de ce semestre, nous nous sommes relayés au laboratoire de mécanique, en dehors des heures destinées au cours de P6, afin de lancer les impressions des différentes pièces. Cependant, nous avons été confrontés à de nombreux soucis techniques ou autres.

En effet, le problème le plus fréquemment rencontré était un soucis de déroulement de la bobine, qui provoquait la rupture du fil, et donc stoppait l'impression. Il était alors nécessaire de relancer la pièce depuis le début. De plus, cet arrêt bouchait généralement la buse, et il était alors nécessaire de la déboucher avant de relancer une impression. Il est aussi arrivé une fois lors de ce genre d'incident que des bouts de matériau viennent boucher la zone d'insertion du fil, nécessitant l'intervention d'un technicien pour nous aider à démonter la partie concernée.

De plus, bien que nous n'ayons pas de réelle explication, certaines pièces étaient fissurées, et donc se cassaient lorsque nous les manipulions. Ainsi, nous avons été contraints de les réimprimer ou bien de les coller, lorsque l'esthétique du robot n'était pas en jeu. Ce problème pouvait également être dû au matériau utilisé. En effet, certaines bobines étaient plus résistantes que d'autres, et nous n'étions qu'autorisés à utiliser du PLA, qui est plus fragile que l'ABS. Ainsi les contraintes mécaniques étaient parfois trop importantes pour ce matériau.

Défaut d'impression sur une pièce et recollage de celle-ci



Nous avons également été confrontés à des soucis d'adhérence de la pièce sur le plateau. En effet, lors de certaines impressions, le matériau n'arrivait pas à bien adhérer, et donc la buse venait arracher la pièce lorsqu'elle repassait par dessus. Nous avons donc essayé de mettre de la colle sur le plateau, sans grand succès. Nous avons finalement réalisé que le plus efficace était de déposer une fine pellicule de laque sur celui-ci avant d'imprimer une pièce.

Un autre souci rencontré fréquemment a été un problème d'emploi du temps. En effet, la salle était parfois fermée et nous ne pouvions donc pas y avoir accès, bien que nous pouvions souvent compter sur les techniciens pour l'ouvrir. De plus, nous n'étions pas les seuls à utiliser l'imprimante, et par conséquent, nous ne pouvions pas toujours lancer l'impression de nos pièces lorsque nous le voulions.

Enfin, nous avons également rencontré une fois un problème avec le bouton de commande de l'imprimante. En effet, celui-ci ne voulait plus s'enfoncer afin de valider les choix. Notre enseignant, Fabrice Delamare, a donc démonté cette partie de la machine afin de la réparer.

3.2. Electronique

3.2.1. Synthèse vocale : Arduino, Acapela Speech

Afin d'effectuer une synthèse vocale, nous avons utilisé une carte Arduino (permettant entre autre de connecter les servomoteurs à l'ordinateur) et le logiciel MyRobotLab. Ce dernier utilise Google Chrome et son micro, permettant la reconnaissance vocale de l'utilisateur.

Sur le logiciel, il est tout d'abord nécessaire de créer différents programmes, à savoir :

- la bouche, qui utilise le programme "*Acapela Speech*", permet la synthèse vocale d'un texte;
- les oreilles, qui utilisent le programme "*WebkitRecognition*", sont reliées au micro et permettent la reconnaissance vocale;
- une session de chat avec l'ordinateur, qui utilise le programme "*programAB*".

Il faut ensuite relier ces différents éléments entre eux à l'aide d'un code écrit en langage Python (utilisant l'instruction "*addTextListener*"). Ainsi, lorsque l'utilisateur parle, le son est détecté et retranscrit à l'écrit par les oreilles, qui vont ensuite transmettre ces données à la session de chat, qui va alors répondre et envoyer le texte à synthétiser vocalement à la bouche. De cette manière, le robot pourra répondre à l'utilisateur.

Cependant, régulièrement, le robot répondait à l'écrit mais pas à l'oral. A force d'effectuer des tests, nous avons remarqué que la probabilité d'une réponse orale augmentait considérablement lorsque nous tapions le code python directement dans la fenêtre du logiciel myrobotlab, plutôt qu'en passant par chrome. Nous ne sommes néanmoins pas capables d'expliquer ce phénomène car les deux fenêtres sont censées être interactives, et nous espérons ne pas rencontrer de soucis lors de la soutenance orale.

Toute cette démarche, est expliquée plus en détail dans l'annexe "tutoriel synthèse vocale",

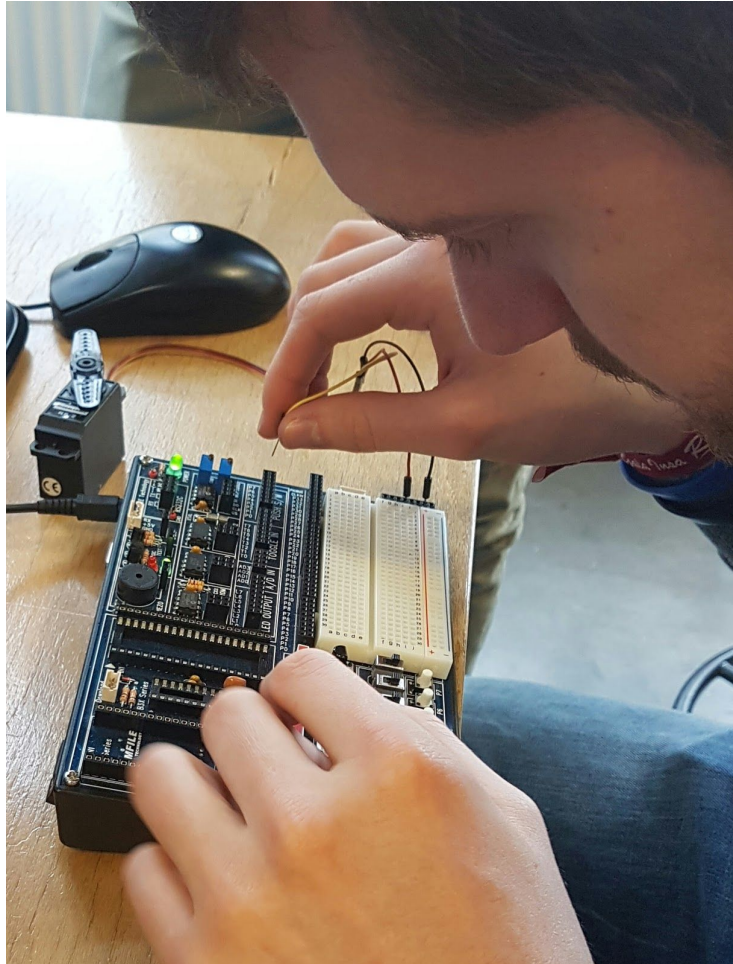
et nous l'avons synthétisée à partir de vidéos, principalement trouvées sur le site inMoov.

3.2.2. Servomoteurs

Lors de ce projet, nous avons pour objectif de faire bouger la mâchoire pendant que le robot parlait. Nous voulions aussi faire bouger aléatoirement les yeux. Pour réaliser ces actions, nous avons dû utiliser des servomoteurs branchés sur une carte Arduino.

Nous avons commencé par essayer de comprendre comment fonctionnaient les servomoteurs dans le logiciel Arduino, les différents ports de la carte, le code pour la vitesse des servos, les différentes positions... Nous l'avons utilisé pour les servomoteurs dont les mouvements sont programmés à l'avance, alors que nous avons directement utilisé le logiciel MyRobotLab pour ceux à bouger manuellement. MyRobotLab inclut aussi une bibliothèque adaptée à inMoov, cependant, nous ne l'avons pas utilisée car elle comportait les dossiers pour tout le corps du robot (les bras, le torse...), et complexifiait ainsi considérablement le code. De plus cette bibliothèque n'était pas conçue pour notre carte (Arduino Uno). Il nous a donc fallu apprendre par nous même les commandes pour servomoteurs sur MyRobotLab ou arduino, ce qui était difficile notamment à cause des problèmes de connexion entre l'ordinateur, la carte et le servomoteur. En effet, dès qu'une modification était effectuée dans MyRobotLab, il fallait débrancher la carte, la reconnecter, la reflasher et refaire toutes les manipulations sous MyRobotLab, nous faisant perdre de la sorte un temps considérable.

Afin d'apprendre à utiliser les servomoteurs, nous avons tout d'abord commencé par les faire bouger à l'aide de la souris, car il s'agit de l'utilisation la plus simple. Nous avons ensuite programmé de petites séquences (comme nous l'avons fait pour le mouvement des yeux dont le programme est donné en annexe "MrCommModifié") en faisant bouger les servomoteurs d'une position à une autre avant de les arrêter pour un temps donné, puis de recommencer. Cependant, nous n'avons pas trouvé comment indiquer au robot de bouger la mâchoire lorsqu'il répond. En effet, nous pensons qu'une commande doit exister, indiquant que lorsque l'on obtient une sortie de texte ou de son pour le programme, il faut bouger le servomoteur. Cependant, nous avons testé beaucoup de commandes indiquées comme compatibles avec notre programme et la bouche (en les triant en fonction des données utilisées en entrée ou restituées en sortie), sans grand succès. Il aurait sans doute été possible d'ajouter une entrée micro sur la carte arduino afin de bouger la mâchoire lorsqu'une certaine fréquence (à savoir la voix du robot) est détectée, mais nous ne l'avons pas fait par manque de moyen et de temps. Ainsi, nous avons décidé de bouger nous-même le servomoteur de la mâchoire manuellement, en changeant les angles de rotation du servomoteur sur l'ordinateur. (Le fonctionnement est détaillé en annexe: "tutoriel des servomoteurs manuels").



Essai de fonctionnement d'un servomoteur

3.2.3. Problèmes rencontrés

Même si globalement, nous avons réussi à utiliser les servomoteurs de façon satisfaisante, nous avons néanmoins été confrontés à quelques difficultés.

Tout d'abord, les différents logiciels, que ce soit ceux pour les servomoteurs ou pour la synthèse vocale, ont été très longs à installer et nécessitaient de nombreuses étapes préliminaires, comme par exemple l'installation de Java. Nous n'avons donc pas pu nous pencher sur les servomoteurs dès les premières séances.

Ensuite, comme évoqué précédemment, nous n'avons pas réussi à faire bouger la mâchoire autrement que manuellement, en faisant varier les positions du servomoteur à l'aide de l'ordinateur.

Nous avons également eu une petite frayeur la dernière semaine, car l'un des moteurs subissait des contraintes trop élevées et donc ne fonctionnait pas. Nous l'avons donc débranché, repositionné et il a refonctionné. Cependant, nous évitons dorénavant de trop l'utiliser afin de ne pas l'abîmer avant la démonstration lors de la soutenance, et en vue d'une éventuelle utilisation future.

3.3. Assemblage

Dans cette partie, nous allons expliquer chronologiquement le déroulement de l'assemblage et les principaux problèmes rencontrés.

Tout d'abord, nous avons numéroté les pièces à imprimer en fonction de leur nécessité, afin d'optimiser notre temps. Cela fut quelque peu difficile car il fallait comprendre le mécanisme et se projeter dans le temps afin de savoir quelles pièces nous seraient utiles ou non. Cela nous a donc permis de pouvoir assembler à chaque cours de nouvelles parties du robot, excepté lorsqu'il y avait quelques erreurs d'impression ou que la pièce ne pouvait être imprimée faute de temps.

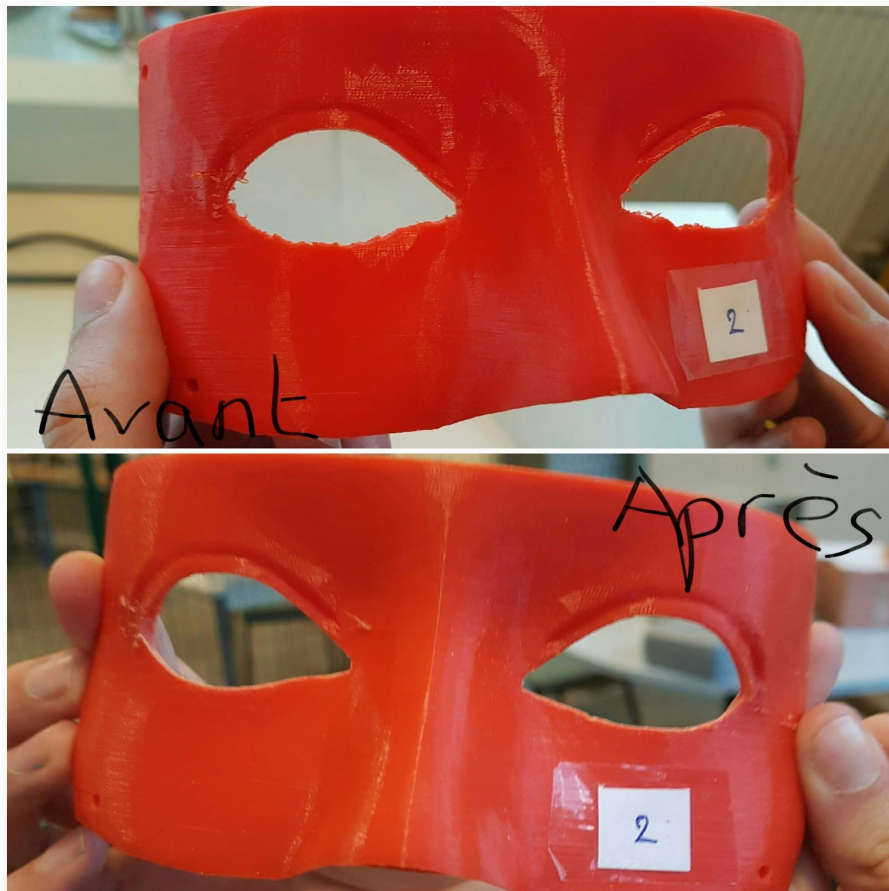
Nous avons imprimé en premier le masque du robot, puisque le mécanisme des yeux est le premier que nous avons souhaité créer. A chaque pièce imprimée, nous nous occupons de poncer légèrement la pièce pour faire disparaître les rugosités, et quand cela était nécessaire, nous reperçons les trous déjà présents. Nous avons également dû créer les yeux du robot afin de leur donner un aspect vivant. Pour cela nous avons utilisé des billes rappelant l'oeil d'un chat afin de recréer l'idée d'une pupille. Nous avons donc divergé des instructions du site après plusieurs essais infructueux car nous n'avions pas à notre disposition le matériel nécessaire. En effet le créateur d'inMoov a utilisé un film plastique chauffé à haute température afin de le mouler et de lui donner la forme d'un oeil. Il a ensuite appliqué ce film sur les yeux qu'il avait coloriés pour donner l'aspect final. Nous avons essayé cette méthode mais nous n'avions pas de plastique qui se déformait convenablement ni de quoi le faire chauffer uniformément (nous utilisons un fer à souder).

Reperçage d'une pièce



Lorsque les yeux furent enfin créés, nous avons suivi précisément les consignes du site (en anglais) et regardé de nombreuses vidéos afin de bien comprendre le mécanisme et les manipulations les plus précises. Nous avons ensuite ajouté au mécanisme les servomoteurs, ce qui fut une opération plus délicate car ils devaient être réglés suivant le bon angle et correctement placés pour que les yeux puissent bouger correctement.

Nous avons ensuite commencé à créer la base du crâne et le mécanisme du cou. Le plus difficile fut le manque de précision des pièces. En effet, un engrenage devait pouvoir tourner à l'intérieur d'une pièce assez librement mais après impression, les pièces ne s'emboîtaient pas. Il a donc fallu retravailler la pièce en la ponçant plus profondément pour que le mécanisme puisse fonctionner. Après cette étape, l'assemblage était simple techniquement jusqu'à la mâchoire.



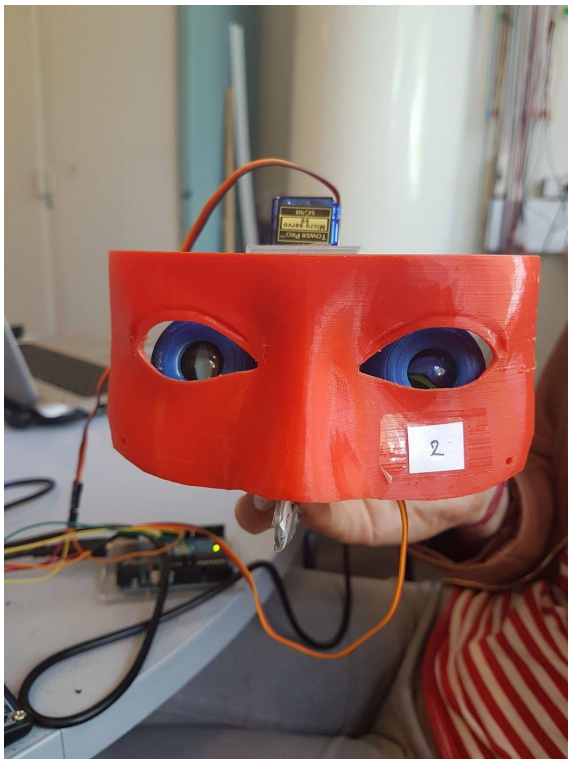
Exemple de ponçage sur le masque

La suite s'est révélée plus difficile. En effet, le site inMoov ne décrivait plus par écrit comment assembler la fin du crâne et ce fut grâce à un PDF interactif que nous avons pu accéder à l'assemblage et que nous l'avons réalisé. Dans ce PDF, chaque pièce avait sa propre place et c'était à nous de savoir quelle vis utiliser et quelle était la pièce

correspondante.

Nous nous sommes également beaucoup aidé du robot Lucie déjà créé qui nous a permis d'observer au mieux le mécanisme et le positionnement des pièces afin de le reproduire. Nous avons dû affronter un réel problème de temps pour l'impression des pièces et finir l'assemblage du crâne lors de la dernière séance, et de la semaine précédant l'oral.

De plus, nous avons réalisé que nous avons oublié le support permettant de poser la tête et de faire fonctionner la mâchoire en même temps. En effet, au début de l'assemblage, la tête semble pouvoir tenir sans soucis sur une surface plane, cependant, en ajoutant la mâchoire, cela crée un déséquilibre et surélève légèrement le mécanisme. Nous l'avons donc imprimé rapidement.



*Masque, structure des yeux et servomoteurs
assemblés*



Armature du crâne

Par ailleurs, nous avons été confrontés à un manque de matériel. En effet, pour certaines pièces, il fallait des vis spéciales et nous n'avons qu'un choix limité. Cela nous a obligé à prendre des décisions en prévoyant les futures pièces pour savoir si une vis dépassant beaucoup serait gênante ou non, ou bien si un mécanisme mal retenu serait moins dérangeant. De plus la tête de notre robot est de taille moyenne et toutes les pièces sont petites et fragiles donc il était difficile de correctement visser sans casser la pièce elle-même. Nous en avons d'ailleurs brisé plusieurs.

Cet assemblage nous a permis de développer notre esprit d'initiative en trouvant des solutions aux différents problèmes techniques que nous avons rencontrés par nous-même.

4. CONCLUSIONS ET PERSPECTIVES

Conclusions sur le travail réalisé

Lors de ce projet, nous avons réussi à imprimer toutes les pièces de la tête de notre robot, et à les assembler. Nous avons également fait fonctionner les servomoteurs pour permettre le mouvement des yeux et de la mâchoire. Enfin nous avons fait fonctionner la reconnaissance et la synthèse vocale pour permettre au robot d'interagir avec l'utilisateur.

Toutefois, nous avons rencontré de nombreux problèmes tout au long de ce projet, rendant impossible la réalisation de certains points prévus initialement, tels que le hochement de tête ou la synchronisation de la voix et de la mâchoire.

Conclusions sur l'apport personnel de cet E.C. projet

Nous sommes satisfaits de notre progression dans ce projet, car malgré notre manque de connaissance au départ, nous avons tout de même réussi à rendre une tête de robot relativement complète, capable d'effectuer des tâches simples. En effet, ce travail de groupe nous a poussé à obtenir diverses connaissances en robotique, en impression 3D et en programmation. De plus, il nous a permis de développer le travail de groupe, que ce soit au niveau de la répartition du travail ou de l'écoute portée à chacun. En effet, dans notre vie professionnelle (ou non), il sera important de savoir s'intégrer au sein d'une équipe non choisie.

Perspectives pour la poursuite de ce projet

Si ce projet était à poursuivre, nous commencerions par achever nos premiers objectifs, à savoir l'impression de la nuque pour permettre le mouvement vertical de la tête et la synchronisation de la mâchoire avec la synthèse vocale.

Par la suite nous ajouterions également une caméra dans l'un des yeux et travaillerions sur une reconnaissance du mouvement et pourquoi pas une reconnaissance faciale.

Remerciements

Toute l'équipe remercie sincèrement Monsieur Delamare, l'enseignant responsable de notre projet, pour son encadrement et son aide tout au long des séances. Nous remercions aussi les techniciens de la salle ARI et de l'imprimante STPI et en particulier Victor pour avoir passé de nombreuses heures à nous aider avec tous les soucis techniques de l'imprimante.

5. BIBLIOGRAPHIE

<http://inmoov.fr/project/> (valide à la date du 15/06/17)

6. ANNEXES

6.1. Documentation technique

6.1.1 *Tutoriel synthèse vocale*

Flasher la carte :

Brancher la carte

ouvrir arduino, fichier → ouvrir → MRLcomm

vérifier dans outil : port et type de carte

téléverser (flèche vers la droite en haut à gauche)

Vérifier que chrome est par défaut

Lors de la première utilisation : copier coller le code en python :

<https://github.com/MyRobotLab/pyrobotlab/blob/master/home/hairygael/InMoov3.DeepAB.V7.py>

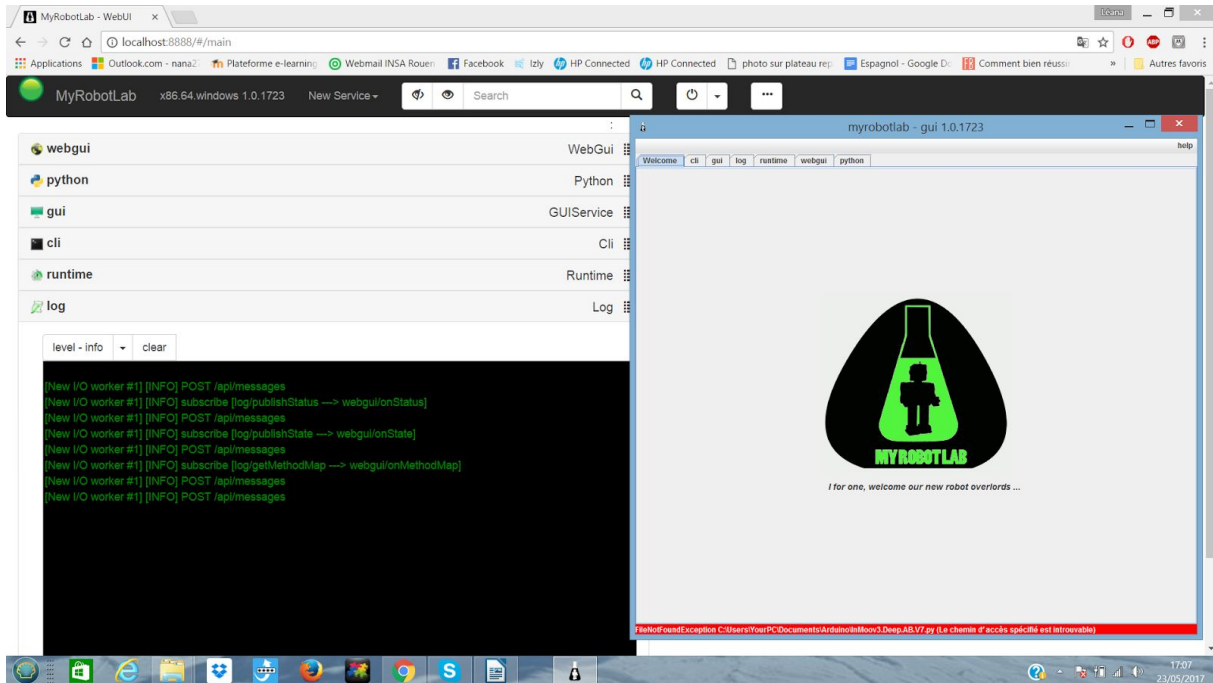
Modifier ce code python dans :

C:/Myrobotlab/myrobotlab.1,0,1723/inMoov3.deep.AB.V7

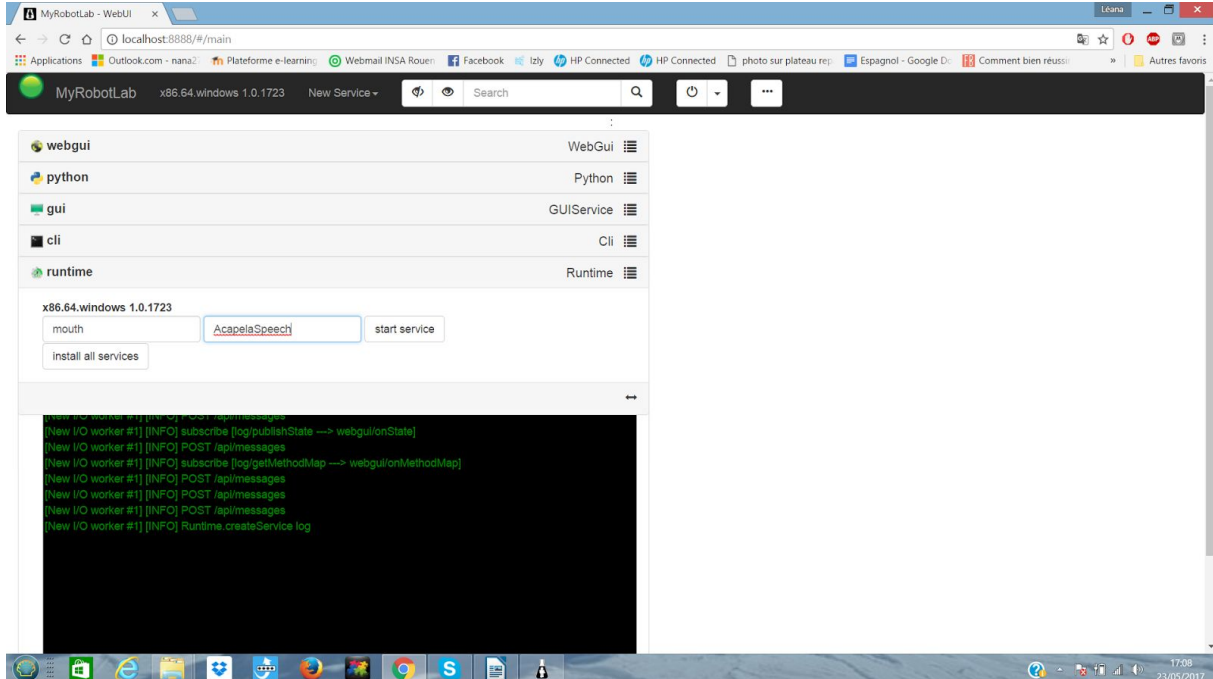
(en cas de besoin mais normalement déjà fait et enregistré)

Ouvrir mrl.bat

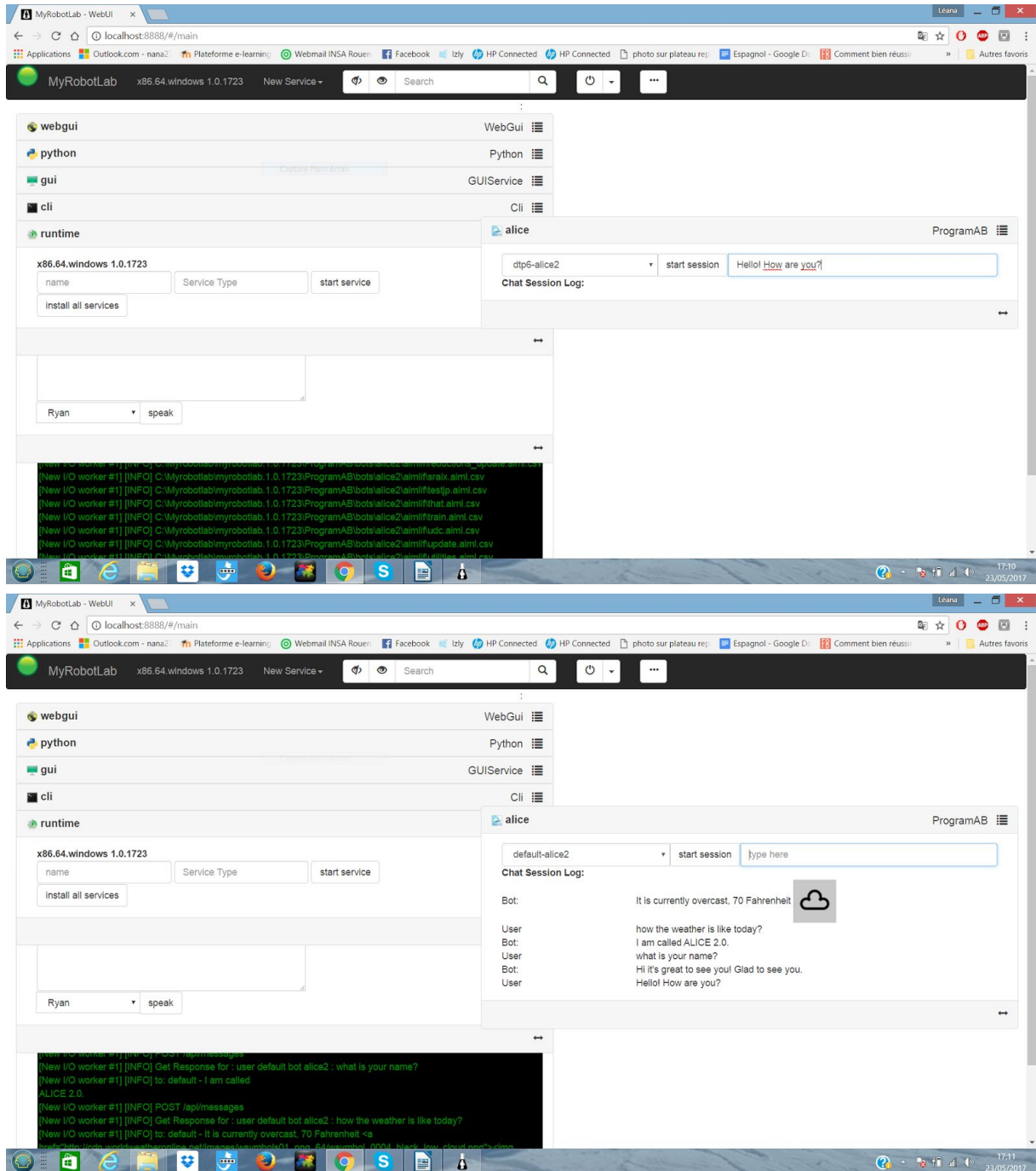
Dans myrobotlab sur chrome ou sur le logiciel directement



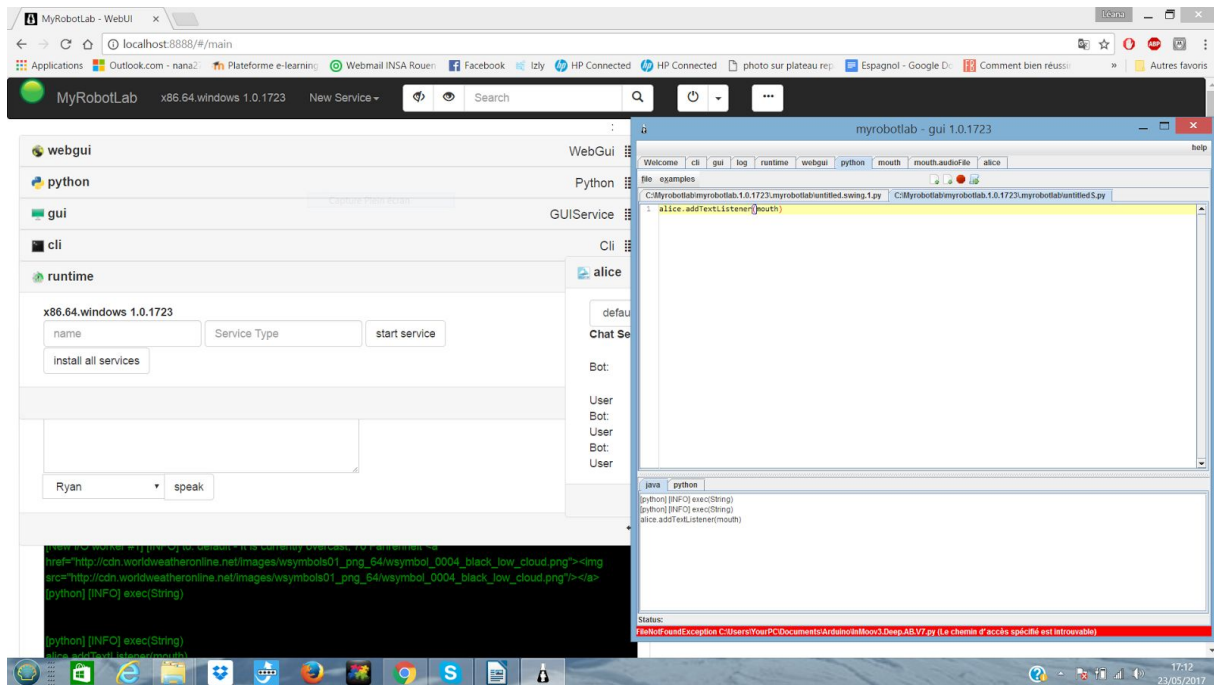
runtime → log / LOG puis mouth / accapelaspeech puis alice /programAB



Dans alice dans myrobotlab → startservice → remplacer default par notre nom → start (taper texte)



python → new script → `alice.addTextListener(mouth)`



runtime → ear/webkitrecognition

Aller dans python → changer le script pour : ear.addTextListener(alice) → execute

6.1.2 *Tutoriel des servomoteurs manuels*

Dans le logiciel myrobotlab:

runtime : arduino (appelé arduino)

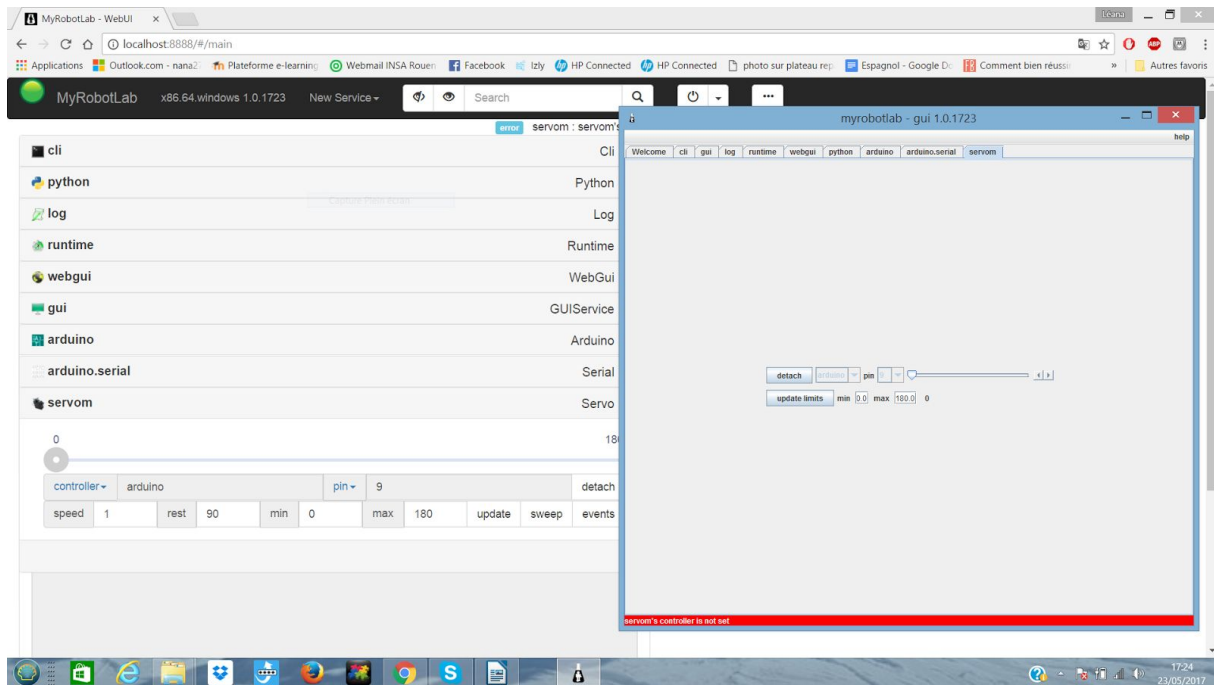
runtime : servo (appelé servom)

dans arduino sélectionner arduino uno, upload MRL comm

dans arduino serial sélectionner le port (COM4)

Dans servom rattacher au bon « pin » puis bouger le curseur : le moteur réagit et tourne de l'angle choisi.

Si cela ne fonctionne pas, rebrancher la carte arduino et retéléverser



6.2. Listings des programmes réalisés

6.2.1 Programme MrlCommModifié

```

/**
 * MRLComm.c
 * -----
 * This file is part of MyRobotLab.
 * (myrobotlab.org)
 *
 * Enjoy !
 * @authors
 * GroG
 * Kwatters
 * Mats
 * calamity
 * and many others...
 *
 * MRL Protocol definition
 * -----
 * MAGIC_NUMBER|NUM_BYTES|FUNCTION|DATA0|DATA1|...|DATA(N)
 *     NUM_BYTES - is the number of bytes after NUM_BYTES to the end
 *
 * more info - http://myrobotlab.org/content/myrobotlab-api
 *
 * General Concept
 * -----
 * Arduino is a slave process to MyRobotLab Arduino Service - this file receives
 * commands and sends back data.
 * Refactoring has made MRLComm.c far more general

```

```

* there are only 2 "types" of things - controllers and pins - or writers and readers
* each now will have sub-types
*
* Controllers
* -----
* digital pins, pwm, pwm/dir dc motors, pwm/pwm dc motors
*
* Sensors
* -----
* digital polling pins, analog polling pins, range pins, oscscope, trigger events
*
* Combination
* -----
* pingdar, non-blocking pulsIn
*
* Requirements: MyRobotLab running on a computer & a serial connection
*
* TODO - need a method to identify type of board http://forum.arduino.cc/index.php?topic=100557.0
* TODO - getBoardInfo() - returns board info !
* TODO - getPinInfo() - returns pin info !
* TODO - implement with std::vector vs linked list -
https://github.com/maniacbug/StandardCplusplus/blob/master/README.md
* TODO - make MRLComm a c++ library
*/

```

```

// Included as a 3rd party arduino library from here: https://github.com/ivanseidel/LinkedList/
#include "LinkedList.h"
#include "MrlComm.h"
#include <Servo.h>
/*****
* GLOBAL VARIABLES
* TODO - work on reducing globals and pass as parameters
*/
MrlComm mrlComm;
Servo monServomoteur;
Servo Monserv2;
/*****
* STANDARD ARDUINO BEGIN
* setup() is called when the serial port is opened unless you hack the
* serial port on your arduino
*
* Here we default out serial port to 115.2kbps.
*/
void setup() {
  // TODO: the arduino service might get a few garbage bytes before we're able
  // to run, we should consider some additional logic here like a "publishReset"
  // publish version on startup so it's immediately available for mrl.
  // TODO: see if we can purge the current serial port buffers
  mrlComm.publishVersion();
  // publish the board type (uno/mega)
  mrlComm.publishBoardInfo();
  monServomoteur.attach(9);
  Monserv2.attach(11);
}

```

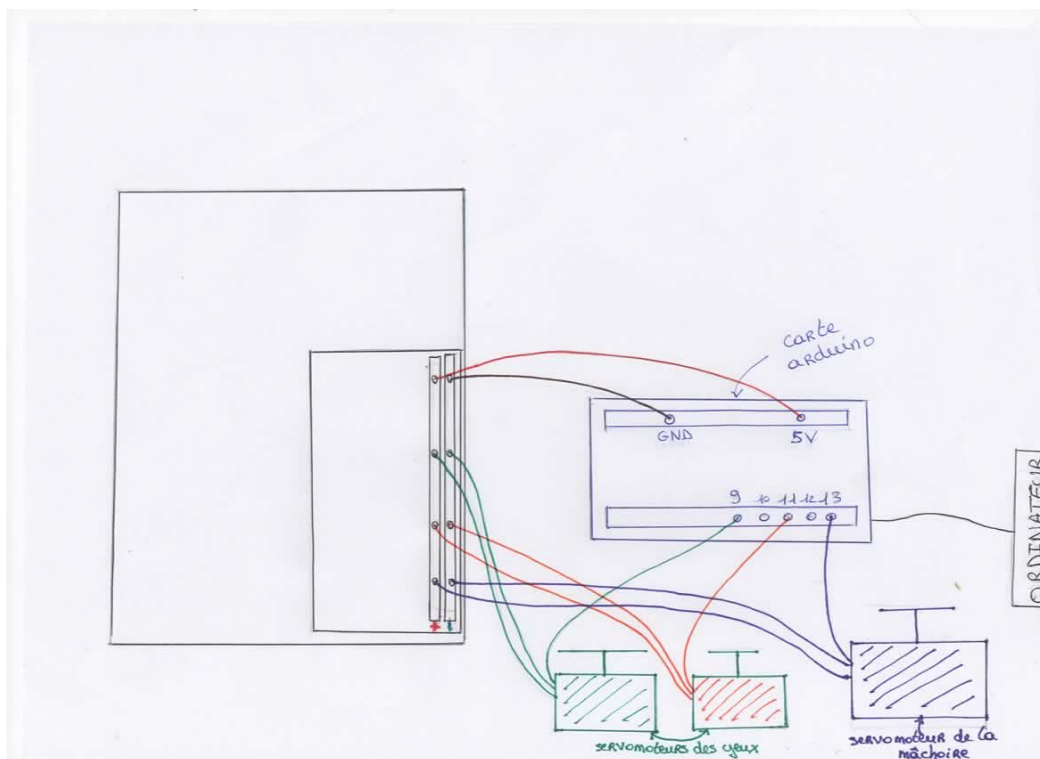
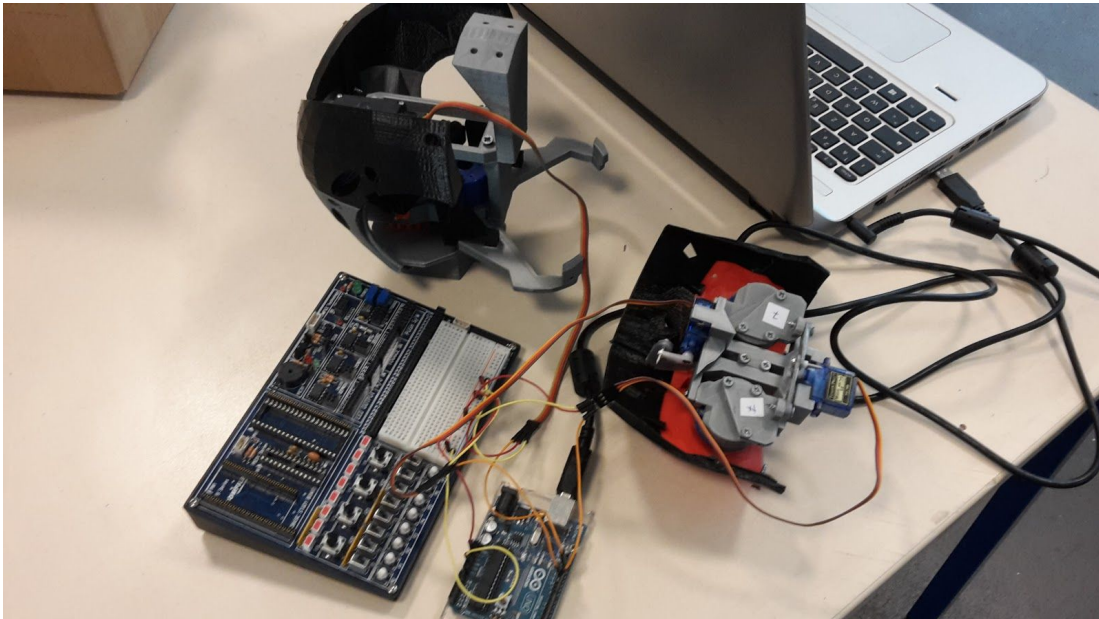
```
/**
 * STANDARD ARDUINO LOOP BEGIN
 * This method will be called over and over again by the arduino, it is the
 * main loop any arduino sketch runs
 */
void loop() {
  // increment how many times we've run
  // TODO: handle overflow here after 32k runs, i suspect this might blow up?
  mrlComm.loopCount++;
  // get a command and process it from the serial port (if available.)
  mrlComm.readCommand();
  // update devices
  mrlComm.updateDevices();
  // send back load time and memory
  mrlComm.publishBoardStatus();
  // Fait bouger le bras de 0° à 180°
  for (int position = 90; position <= 180; position++) {
    monServomoteur.write(position);
    delay(15);
  }

  // Fait bouger le bras de 180° à 10°
  for (int position = 180; position >= 90; position--) {
    monServomoteur.write(position);
    delay(15);
  }
  for (int position = 90; position <= 180; position++) {
    Monserv2.write(position);
    delay(15);
  }

  // Fait bouger le bras de 180° à 10°
  for (int position = 180; position >= 90; position--) {
    Monserv2.write(position);
    delay(15);
  }
} // end of big loop
```


6.3. Schémas de montages

6.3.1 Branchements des servomoteurs



6.4. Propositions de sujets de projets (en lien ou pas avec le projet réalisé)

Il pourrait être intéressant de créer la suite de notre robot, soit d'autres parties du corps afin de créer le robot en entier sur le long terme (par des élèves de STPI2).