



ÉCOLE POLYTECHNIQUE DE L'UNIVERSITÉ DE TOURS
64, Avenue Jean Portalis
37200 TOURS, FRANCE
Tél. (33)2-47-36-14-14
Fax (33)2-47-36-14-22
www.polytech.univ-tours.fr

Parcours des écoles d'ingénieurs Polytech

Rapport de projet S4

Main robotique imprimée en 3D avec contrôle Bluetooth

Auteur(s)

Colin Troisemaine

[colin.troisemaine@etu.univ-tours.fr]

Quentin Levieux

[quentin.levieux@etu.univ-tours.fr]

Encadrant(s)

Guénaël Le Quilliec

[guenael.lequilliec@univ-tours.fr]

Polytech Tours

Département Peip

Table des matières

Introduction	2
1 Présentation du matériel et des logiciels utilisés	3
1.1 Catia	3
1.2 Imprimante 3D	3
1.3 Découpeuse laser	4
1.4 Capteur bluetooth et MIT App Inventor	5
1.5 Servomoteurs	6
1.6 Arduino	6
2 Les différentes pièces créées	8
2.1 La paume	8
2.2 Les cinq doigts	8
2.3 Les poulies	11
2.4 Divers	13
3 Le circuit et les logiciels	16
3.1 Le circuit	16
3.2 Le code version terminal	17
3.3 Le bluetooth et MIT App Inventor 2	18
Conclusion	20
Annexes	20
A Liens utiles	21
B Fiche de suivi de projet PeiP	22

Table des figures

1	Photo du projet	2
1.1	Logo de Catia	3
1.2	L'imprimante 3D utilisée	4
1.3	La découpeuse laser	4
1.4	Les deux fenêtres d'App Inventor	5
1.5	La capteur bluetooth HC-05	6
1.6	Le servomoteur Futaba	6
1.7	La carte Arduino Uno	7
2.1	La paume	8
2.2	La main d'InMoov	9
2.3	Le doigt assemblé	9
2.4	Les deux versions d'une phalange	10
2.5	Les différentes pièces du doigt	10
2.6	Le pouce assemblé	11
2.7	Poulie défectueuse	12
2.8	Relation de distance	12
2.9	Le modèle de poulie final	13
2.10	Le support des servomoteurs	13
2.11	L'attache de la paume	14
2.12	Un simple guide pour les fils	14
2.13	proportions	15
3.1	Diagramme du circuit	16
3.2	Fonction de décomposition	17
3.3	Autres fonctions	17
3.4	Un exemple de bloc pour l'ascenseur 2 dans MIT App Inventor	18
3.5	Un extrait du code Arduino	19

Remerciements

Nous tenons à remercier toutes les personnes qui nous ont permis d'avancer dans notre projet et qui ont pris de leur temps pour nous fournir des explications ou du matériel : Guénhaël Le Quilliec, Emmanuel Penaud ainsi que toute l'équipe des professeurs que nous avons eu l'occasion de rencontrer.

Et nous voulons notamment remercier Léo Legrand du DI pour avoir imprimé plusieurs pièces dont nous avions besoin, et ce toujours rapidement !

Introduction

L'objectif premier de ce projet était de concevoir une main, de l'imprimer en 3D et d'utiliser 5 servomoteurs afin de contrôler chacun des doigts individuellement. Nous nous sommes naturellement orientés vers le logiciel de DAO "Catia" pour modéliser la main, puisque l'un de nous avait déjà eu des cours de dispensés sur ce programme. Étant déjà familiers avec l'environnement arduino, nous l'avons logiquement choisi pour contrôler nos servomoteurs. Si le temps nous le permettait, nous avions aussi pour objectif de développer une application bluetooth sur smartphone pour contrôler cette main.

Une démonstration du projet est disponible à l'adresse suivante : <https://youtu.be/I3zrs9ICd6Y>

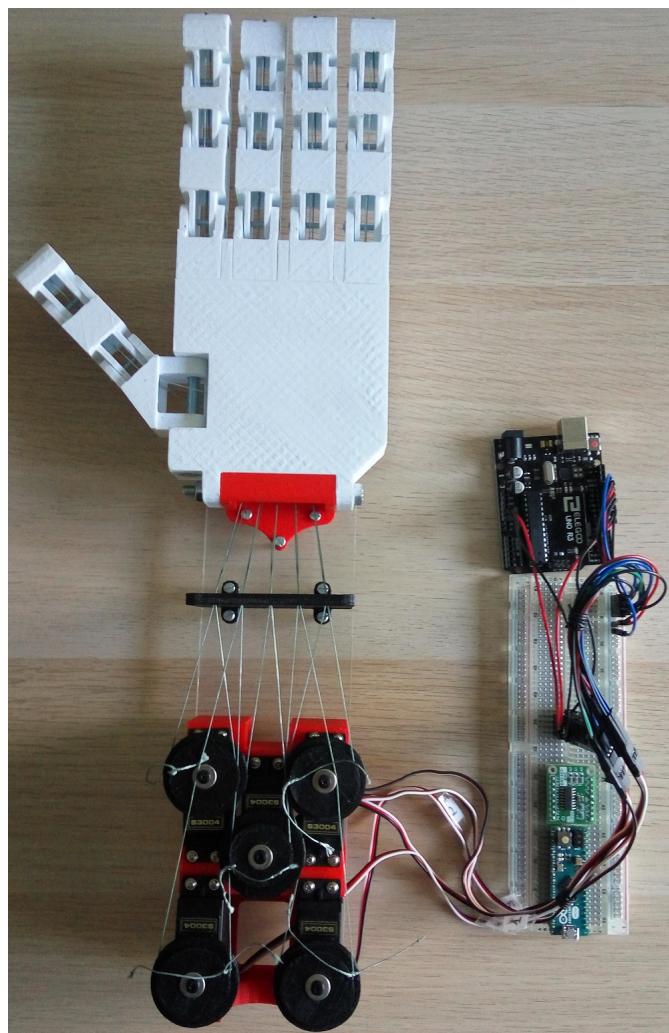


FIGURE 1 – Photo du projet

Chapitre 1

Présentation du matériel et des logiciels utilisés

1.1 Catia



Catia (Conception Assistée Tridimensionnelle Interactive Appliquée) est un logiciel de conception assistée par ordinateur, créé par la société Dassault Systèmes. L'intégralité de la main robot a été désignée par nos soins sur le logiciel Catia sur la version V5R20. Catia possède de nombreuses fonctionnalités, mais nous n'en avons utilisé que 3 : part design, assembly design et drawing.

FIGURE 1.1 – Logo de Catia

- Part design nous permet de réaliser une pièce en 3D. Pour réaliser une pièce nous faisons un assemblage de sketchs, les interfaces utilisateur sont assez intuitives. En nous avons réalisés 14 pièces au total.
- Assembly design nous permet de manipuler plusieurs pièces entre elles selon une contrainte imposée (contact, axe de rotation, ...), de les assembler et de voir comment elles réagissent entre elles.
- Drawing permet à partir d'une pièce issue de part design d'en faire une mise en plan selon la vue souhaitée.

1.2 Imprimante 3D

L'imprimante 3D que nous avons utilisée pour créer la majorité des pièces qui composent notre projet est notre imprimante personnelle. Il s'agit de la Mecreator 2 fabriquée par Giantarm. C'est une imprimante open source très facile à configurer et à mettre en marche avec un prix abordable pour une première imprimante 3D (environ 350€).

Pour réaliser nos modèles, nous avons utilisé le logiciel fourni par le fabricant : Repetier-Host. Le fichier 3D que l'on lui transmet sous format .stl directement depuis Catia est traité par le très connu "Cura engine" qui organise le découpage en tranches nécessaire à la réalisation de la pièce.

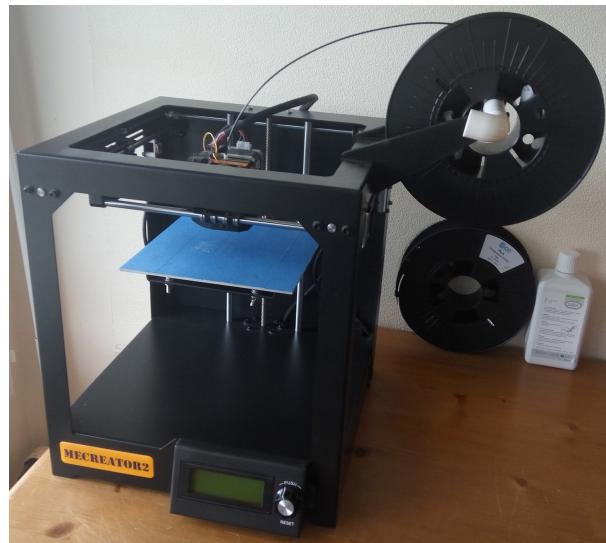


FIGURE 1.2 – L'imprimante 3D utilisée

1.3 Découpeuse laser

Pendant ce projet, nous n'avons pas utilisé uniquement l'impression 3D, nous avons également eu l'occasion d'utiliser une machine de découpe laser.

Le Plug And Fab dispose de la graveuse laser Speedy 400 de la marque autrichienne « Trotec » d'un coût d'environ 35 000 €. Elle à comme avantage de posséder une grande surface de travail de 1.00m x 0.61m ce qui est évidemment largement supérieur à ce que nous voulions faire découper. De plus elle permet de découper divers matériaux tel que le bois, le plastique, le plexiglas ou même le métal. La découpe laser est rapide et propre (aucun ponçage à effectuer). Les fichiers pour la découpe doivent être de type vectoriel, nous avons donc utilisé la fonction drawing de Catia pour enregistrer nos pièces en format SVG (Scalable Vector Graphics).



FIGURE 1.3 – La découpeuse laser

source : <http://celene.univ-tours.fr/course/view.php?id=5953§ion=5>

1.4 Capteur bluetooth et MIT App Inventor

MIT App Inventor est une application développée par Google du 15 décembre 2010 au 31 décembre 2011. C'est le Massachusetts Institute of Technology (MIT) qui s'occupe dorénavant de son développement. MIT App Inventor permet de concevoir une application pouvant s'exécuter sur les appareils Android. Son but premier est de simplifier le développement d'une application, en effet le langage de programmation est graphique. C'est à dire que le programme est écrit par assemblage d'éléments graphiques. Ce qui est bien plus intuitif qu'un langage de programmation textuel classique tel que Java et ses nombreuses lignes de code.

MIT App Inventor se décompose en deux interfaces. La première est pour le design de l'application et une seconde pour le codage en blocks.

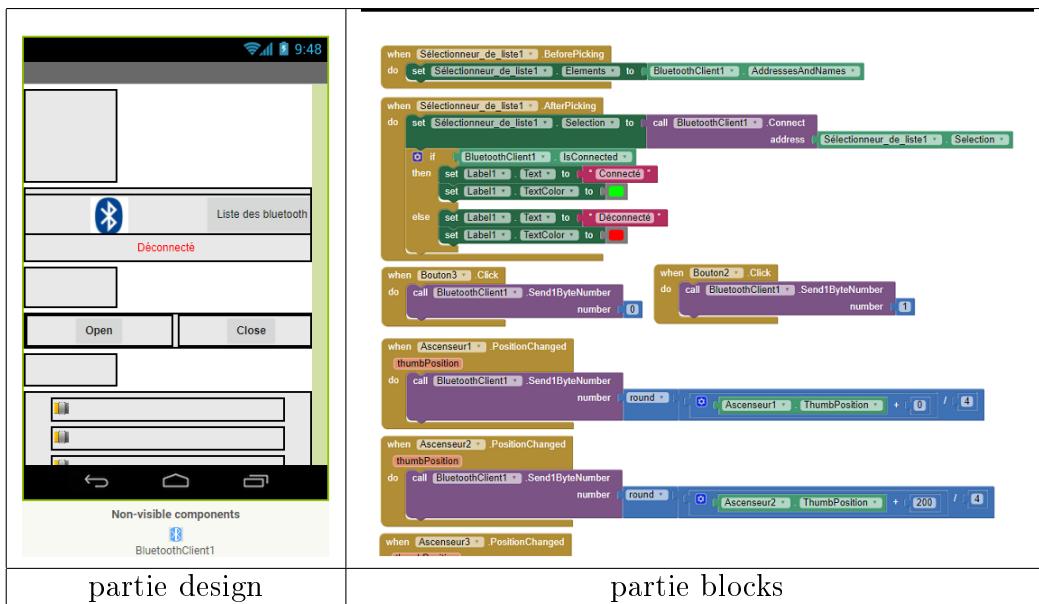


FIGURE 1.4 – Les deux fenêtres d'App Inventor

Pour utiliser l'application nous devions pouvoir interagir entre notre carte Arduino et MIT App Inventor, pour ce faire nous avons utilisé un module bluetooth HC-05 : C'est un petit module que vous pouvez trouver pour 8,50€ sur Amazon. Le module HC-05 fonctionne via un port série avec une carte Arduino. Le récepteur possède 6 broches : VCC, GND, RXD, TXD, En et State. Nous n'utiliserons pas les deux dernières. VCC et GND servent pour l'alimentation, VCC relié à du 5V et GND à la masse. TXD et RXD servent pour la transmission et la réception des informations, eux même reliés sur des pins de la carte.

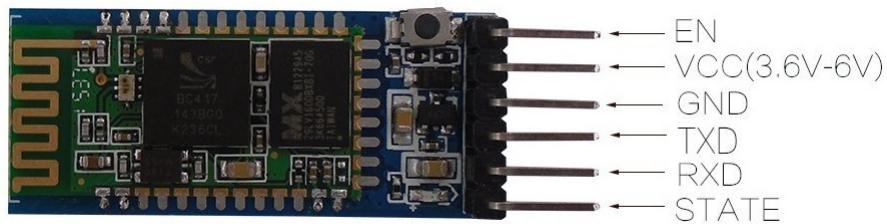


FIGURE 1.5 – La capteur bluetooth HC-05

source :

https://images-na.ssl-images-amazon.com/images/I/61TQG6ds5mL._SL1001_.jpg

1.5 Servomoteurs

À l'instar d'un moteur classique, le servomoteur ajuste et corrige sa position continuellement pour tenir la position qui lui a été commandée. C'est un système qui mélange l'électronique, la mécanique et l'automatique.

Pour actionner chacun de nos doigts, il nous fallait 5 servomoteurs. Nous avons donc utilisé le site du Plug And Fab sur lequel nous avons pu remplir une fiche de demande de consommables. Le modèle de servomoteur dont nous nous sommes servis est donc le Futaba S3004, il est léger, possède un roulement à billes et un angle de rotation de 180 degrés.



FIGURE 1.6 – Le servomoteur Futaba

source : www.dbsportandscale.com/ekmps/shops/290035/images/futaba-s3004-standard-bb-servo-7678-p.jpg

1.6 Arduino

L'idée d'Arduino est d'utiliser un circuit imprimé pour pouvoir contrôler tout ce que l'on veut de façon très simple. Il suffit de brancher la carte à son ordinateur et de lancer le logiciel gratuit d'Arduino pour commencer à développer son programme que l'on pourra transférer par

le câble USB fourni.

Afin de contrôler nos servomoteurs en toute simplicité, nous avons naturellement choisi d'utiliser cet environnement. En effet nous étions déjà familiers avec celui-ci grâce à nos projets de Peip 1 ou d'autres travaux personnels.

Possédant déjà une carte, nous avons utilisé celle ci. Il s'agit du modèle Arduino Uno R3 fabriquée par Elegoo et coûte environ 11€ (voir : www.amazon.fr/gp/product/B01N91PVIS).



FIGURE 1.7 – La carte Arduino Uno

source : images-na.ssl-images-amazon.com/images/I/81xSV1TGCLL._SL1500_.jpg

Chapitre 2

Les différentes pièces créées

2.1 La paume

La paume de la main fut notre première réalisation. Elle est la pièce centrale autour de laquelle toutes les autres seront articulées et organisées. Avant de commencer une modélisation sur ordinateur nous avons commencé par un travail en amont sur un brouillon. Nous savions également que nous voulions faire bouger un doigt à l'aide de 2 fils. La forme de la paume est tout simplement inspirée d'une paume humaine avec une représentation très carrée et épurée de celle ci. La faible complexité de la pièce a permis à Colin de s'approprier les bases de Catia, car il n'a pas reçu de cours d'initiation au DAO. Le sketch de la pièce fut extrudé, et nous avons ensuite retiré la matière à l'intérieur afin de la rendre creuse. Premièrement pour une économie de plastique au moment de l'impression 3D et secondement pour nous faciliter le passage des fils dans chaque doigts.

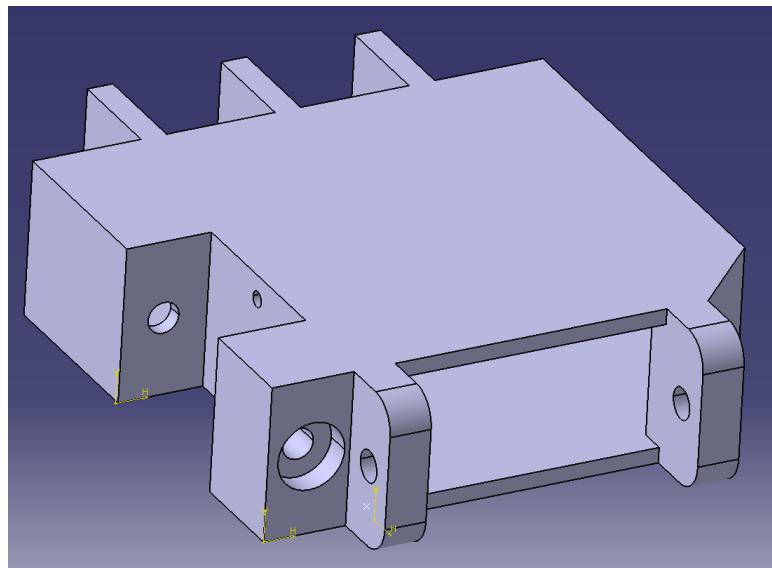


FIGURE 2.1 – La paume

2.2 Les cinq doigts

Une fois la paume modélisée, nous avons commencé à travailler sur les doigts : La quantité de travail à fournir pour créer un seul doigt étant déjà très grande, nous avons choisi de n'utiliser qu'un seul modèle pour l'index, le majeur, l'annulaire et l'auriculaire. Seul le pouce aura un design qui diffère.

La première étape de création de notre doigt a été la recherche de projets similaires sur internet. Le site www.thingiverse.com permet à ses utilisateurs de partager les modèles 3D qu'ils ont créés et il possède une importante communauté, il nous a donc été d'une grande aide lors de nos recherches. Le modèle de main qui est le plus souvent revenu sur ce site est la main du robot "InMoov". Il s'agit d'un robot Open Source que n'importe qui peut télécharger et imprimer chez lui. Il a notamment été créé par Gael Langevin, un sculpteur et designer Français.

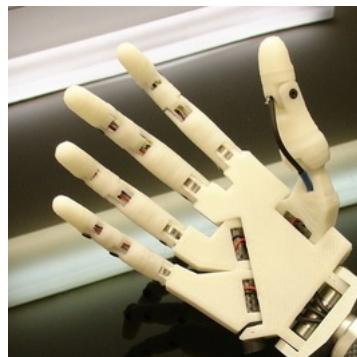


FIGURE 2.2 – La main d'InMoov
source : www.thingiverse.com/thing:17773

Mais cette main ayant été réalisée par un professionnel et étant tellement avancée, elle ne pouvait que peu nous servir de modèle. Nous nous sommes donc rabattus sur des mains plus simplistes et à notre portée. Nous avons également choisi de ne pas prendre comme source d'inspiration un unique modèle, mais de plutôt en prendre une multitude pour éviter le plagia. Ainsi nous avons pu comparer de nombreux systèmes de pliage-dépliage et noter de multiples détails cruciaux.

Lors des séances suivant la création de la paume, nous avons donc modélisé notre doigt. Mais le modèle de doigt que nous avons à présent n'a pas été obtenu du premier coup, nous avons procédé par tâtonnement :

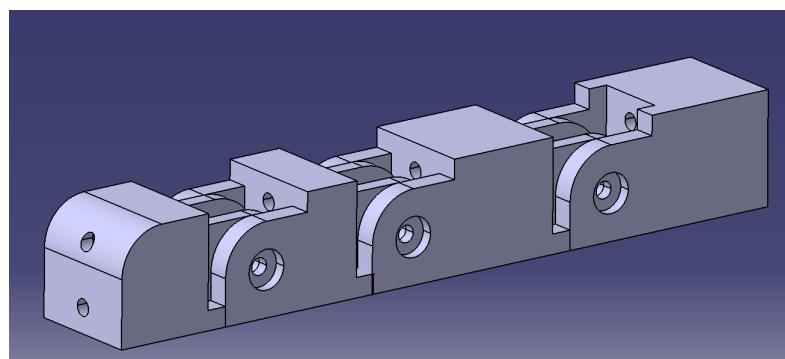


FIGURE 2.3 – Le doigt assemblé

La première version que nous avions imaginée était plutôt simpliste et carrée, mais une fois imprimée elle nous a permis de remarquer de nombreuses imperfections. En effet les deux fils

cheminant à l'intérieur du doigt n'étaient pas guidés du tout, ainsi il était impossible d'ouvrir le doigt une fois qu'il avait été fermé. Pour résoudre ce problème, nous avons ajouté une cloison avec deux trous les plus espacés possibles et un axe à l'extérieur de la jointure de chaque phalange (voir figure 2.4).

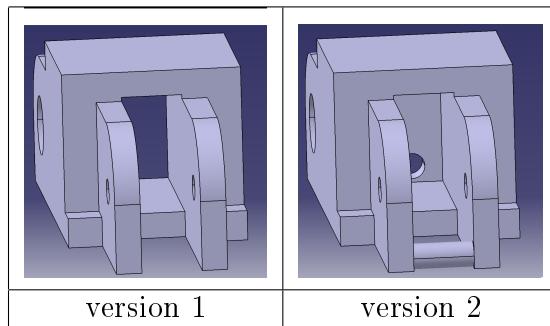


FIGURE 2.4 – Les deux versions d'une phalange

Enfin avant d'imprimer la version finale de notre doigt, nous avons arrondi la dernière phalange afin de lui permettre de se refermer aisément et améliorer l'esthétique et nous avons corrigé quelques erreurs comme la taille des renflements où se rangent les écrous.

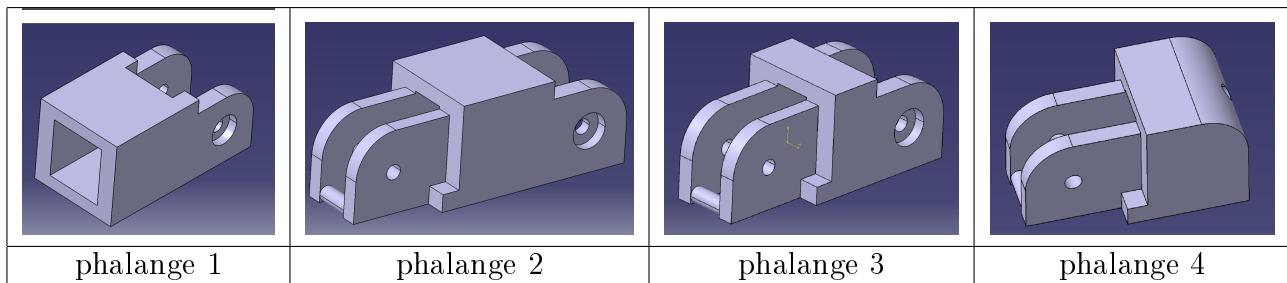


FIGURE 2.5 – Les différentes pièces du doigt

Au final, nous sommes vraiment satisfaits du résultat. Le doigt est composé de 4 phalanges différentes qui s'articulent grâce à deux fils qui circulent le plus possible à l'extérieur pour l'ouverture et à l'intérieur pour la fermeture. La première phalange a quand à elle été pensée de façon à ce qu'elle puisse s'emboiter et être collée dans la paume, c'est de là que vient sa base plate et carrée.

Créer le pouce après avoir réalisé le doigt a été assez rapide puisque nous avons pu réutiliser de nombreuses formes et mesures déjà existantes. Le seul challenge a été d'imaginer une jointure qui permette au pouce d'être d'une part à 45 degrés de la paume et d'autre part qu'il n'y ait aucune intersection avec cette dernière lors la fermeture du pouce. La fonction d'analyse des intersections de Catia nous donc a été ici d'une grande aide !

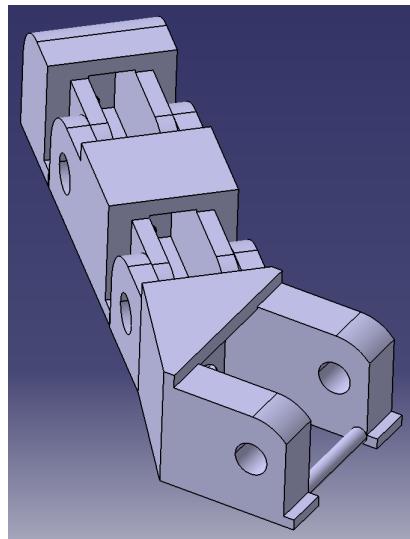


FIGURE 2.6 – Le pouce assemblé

2.3 Les poulies

Nous avons décidé d'articuler un doigt avec deux fils : un fil passant dans la partie intérieur qui ferme le doigt lorsqu'il est tiré et un second fil passant dans la partie extérieur permettant au doigt de s'ouvrir. Ces deux actions peuvent alors être interprétées par un servomoteur surmonté d'une poulie. Lorsque la poulie tourne dans un sens, elle tire sur un fil et donne du mou au second, et inversement dans l'autre sens.

En partant de cette idée et possédant déjà un premier modèle de doigt imprimé, nous avons pu calculer la distance parcourue par les fils à l'intérieur du doigt : Nous avons donc immobilisé le doigt et pour le plier entièrement il fallait tirer de 4,9 cm sur le fil intérieur tandis que le fil extérieur rentrait de 4 cm.

Notre servomoteur pouvant effectuer une rotation de 180 degrés, nous avons simplement appliquée la formule :

$$\pi \times r = 4,9 \text{ cm} \iff r = 1,56 \text{ cm}$$

$$\pi \times r = 4,0 \text{ cm} \iff r = 1,27 \text{ cm}$$

Connaissant les deux rayons, nous avons utilisé la fonction Drawing de Catia pour ébaucher une première poulie ovale :

Nous nous sommes vite rendus compte que ce type de poulie n'était absolument pas adapté car le fil n'était guidé lors de son trajet et ne parcourait pas 4,9 ou 4 cm. Nous avons alors imaginé une poulie possédant deux rigoles. Mais avant de la créer il fallait nous assurer que le trajet d'un fil par rapport à l'autre était bien linéaire. En effet si ce n'était pas le cas, il aurait fallu créer une poulie avec une rigole ronde surmontée d'une rigole d'une forme non circulaire. Il existe des simulations sur internet permettant de trouver une poulie à partir d'un chemin (par exemple : <http://demonstrations.wolfram.com/ShapingARoadAndFindingTheCorrespondingWheel/>). Pour vérifier cela, nous avons alors pris de nouvelles mesures à différentes positions du doigt en relevant la position du premier fil tout en tirant le second.

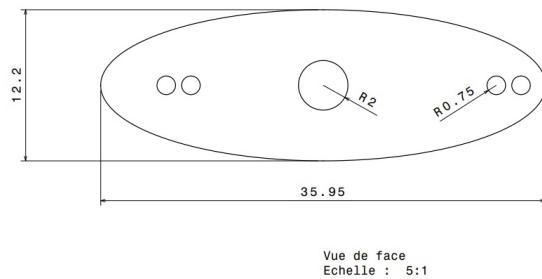


FIGURE 2.7 – Poulie défectueuse

Voici le graphique que l'on a ainsi obtenu :

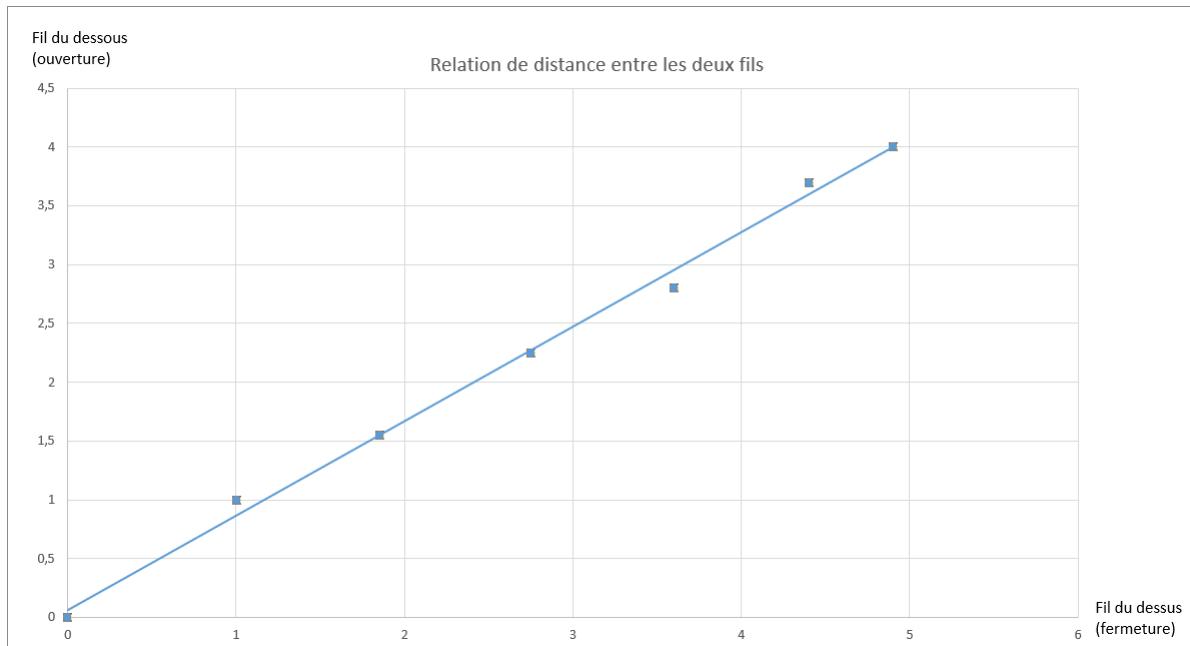


FIGURE 2.8 – Relation de distance

Il apparaît clairement que les points sont répartis le long d'une droite, ce qui nous permet bien de créer deux poulies de forme circulaire. Au final nous avons même pu conserver nos calculs de rayons qui étaient corrects.

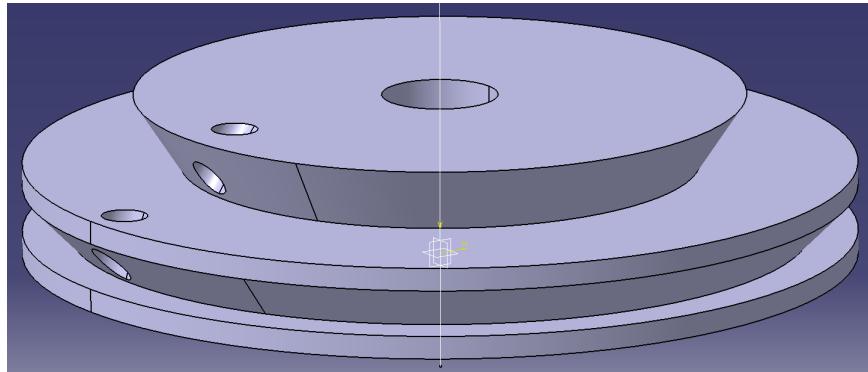


FIGURE 2.9 – Le modèle de poulie final

2.4 Divers

Outre les pièces dont nous venons de parler et que nous pourrions qualifier de principales, nous avons du modéliser quatre autres pièces pour assurer le bon fonctionnement de la main :

- Afin de solidement fixer les cinq servomoteurs qui actionnent les doigts, nous avons créé un support adapté spécifiquement au modèle de servomoteur dont nous disposions. Lors sa modélisation sous Catia, nous avions également comme objectif de créer un support le plus léger possible afin de réduire le temps et le coût lors de l'impression, tout cela sans perdre en solidité.

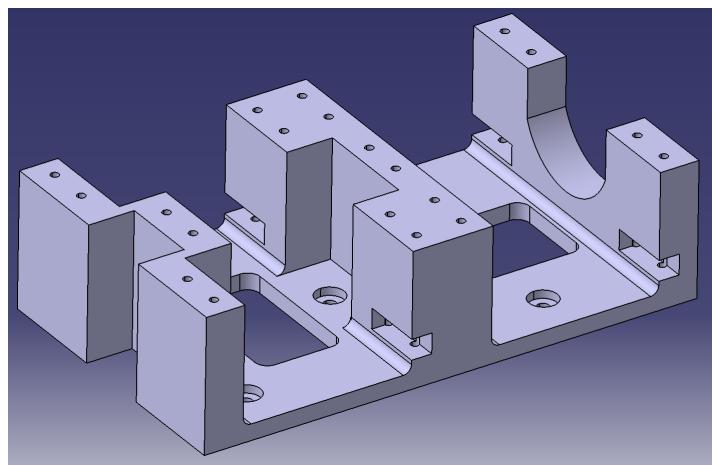


FIGURE 2.10 – Le support des servomoteurs

Une des contraintes majeures à prendre en compte était l'espacement minimum qu'il fallait respecter entre chaque poulie. Pour éviter de prendre trop de place, la solution a été d'alterner le sens des servomoteurs puisque leur axe de rotation n'était pas centré.

- Un autre élément auquel il a fallut réfléchir est l'attache qui fixait la paume. Ici rien de bien compliqué puisqu'il a suffit de copier la forme de la paume et d'ajouter une base avec

3 trous espacés pour la solidité et une cloison avec dix trous pour guider chacun des fils.

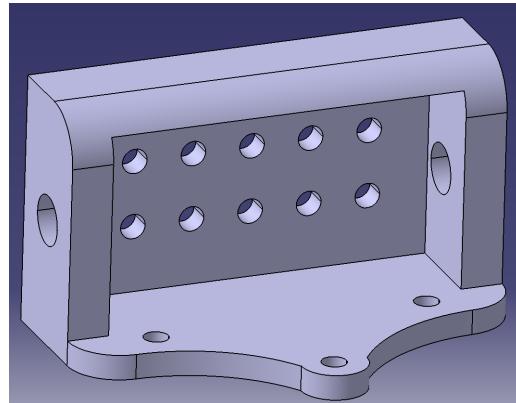


FIGURE 2.11 – L'attache de la paume

- Ensuite pour que le support des servomoteurs reste à une distance fixée par rapport à la main, nous avons choisi de découper au laser une simple planche de plexiglas de 5 mm d'épaisseur. Ici aucune contrainte, il fallait simplement que les trous du support des servomoteurs et de l'attache de la paume soient bien positionnés. Étant une pièce découpée au laser, nous avons eu l'occasion de nous initier à la fonction dessin de Catia.
- Enfin la dernière pièce que nous avons créée est un guide pour les fils. En effet lorsque les fils sortaient de la paume, ils étaient plus bas que les poulies, faisant sortir les fils des poulies lorsque les servomoteurs tournaient.

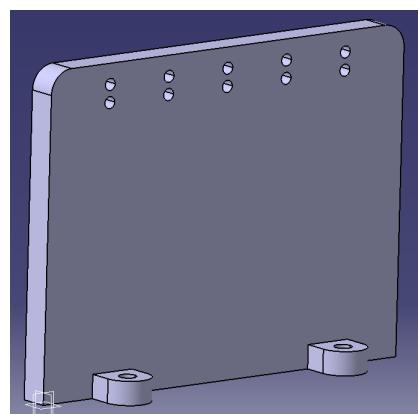


FIGURE 2.12 – Un simple guide pour les fils

Lors de la conception de la main, nous n'en avons pas négligé l'aspect général. En effet pour que cette dernière ait un aspect relativement esthétique, nous avons essayé de conserver des proportions "humaines". Nous nous sommes alors inspiré des dessinateurs pour les proportions générales qu'une main doit avoir.

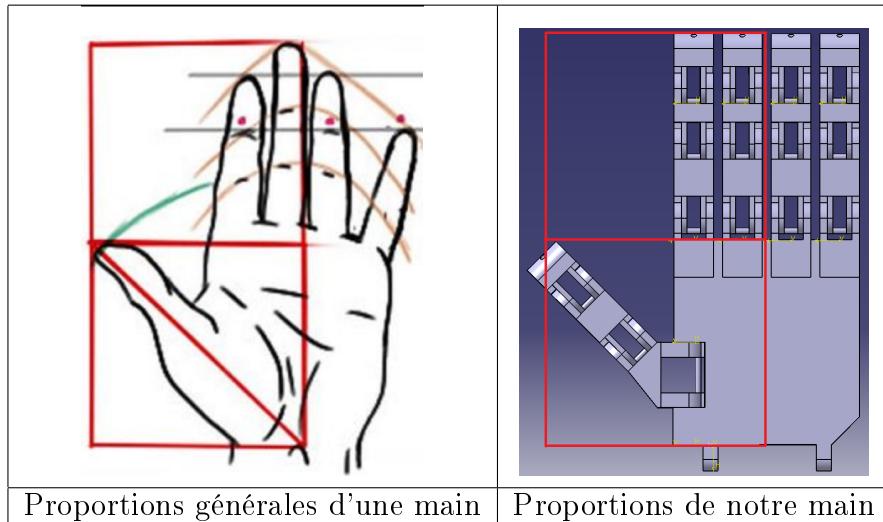


FIGURE 2.13 – Les proportions

Le trait rouge sur la figure 2.13 fait environ 9 cm pour une main normale, tandis que sur notre main il fait 11 cm. Nous avons donc créé une grande main, mais aux proportions correctes.

Nous avons notamment bien pris en compte le problème de la visserie, et avant de modéliser quoi que ce soit, nous avons soigneusement choisi quel type de boulons et d'écrous nous voulions utiliser. Chaque phalange est ainsi tenue par un boulon de 3 x 20 mm, le pouce est traversé par un boulon de 6 x 60 mm et la paume est fixée par un boulon de 6 x 80 mm.

Au final cette partie de conception de toutes les pièces composant la main aura été de loin la plus chronophage avec plus d'une centaine d'heures de modélisation pendant notre temps libre (vacances, week-end, entre les cours, ...). Sans cela il aurait été impossible de mener notre projet jusqu'au bout.

Chapitre 3

Le circuit et les logiciels

3.1 Le circuit

L'un de nous ayant eu l'année dernière un projet comportant également une carte arduino et des servomoteurs, le montage du circuit ne nous a causé aucun problème : Il a suffit de connecter en dérivation grâce à une breadboard les 5 servomoteurs et la batterie. Ensuite il a fallut connecter pour chaque servomoteur le fil restant (qui correspond au signal) aux pins 9 à 13 de la carte arduino. Et enfin, comme nous l'avons décrit dans la partie 1.4, nous avons connecté le capteur bluetooth directement à la carte arduino avec les deux pins RX et TX ainsi que l'alimentation au pins *ground* et *5V*. Sans oublier de connecter un câble du côté (-) de la breadboard jusqu'au *ground* de notre carte.

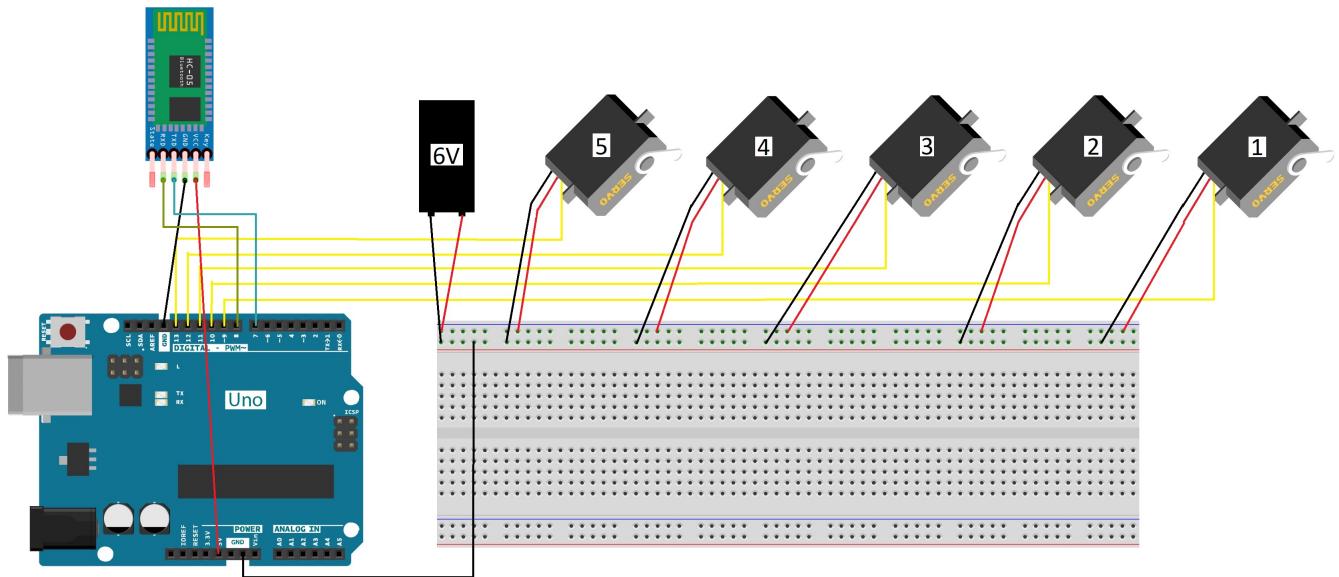


FIGURE 3.1 – Diagramme du circuit

Au final, nos servomoteurs n'ayant quasiment pas à exercer de force pour refermer les doigts, notre batterie composée simplement de 4 piles AA en série nous offre une autonomie tout à fait suffisante. Nous avons par ailleurs pris soin de mesurer à l'aide d'un multimètre le voltage délivré par cette batterie : Il est de 5,5V lorsque la charge est récente, ce qui convient parfaitement puisqu'il est recommandé de faire fonctionner les servomoteurs entre 4,8 et 6V.

3.2 Le code version terminal

Ayant terminé le montage de la main et du circuit, nous avons alors créé un premier code destiné principalement à nous permettre de vérifier le bon fonctionnement de tous nos doigts. Sans nous intéresser pour l'instant à l'aspect bluetooth, nous avons écrit un code utilisant le port *serial* de notre carte arduino. La principale caractéristique de ce code était qu'en écrivant dans la console du logiciel arduino, notre carte était capable de comprendre des commandes et de les interpréter. Par exemple si l'on envoie "set 1 90", notre servomoteur numéro 1 se positionne à 90 degrés.

Ci-dessous est la fonction qui permet à notre carte d'interpréter les commandes envoyées par l'ordinateur :

```
void decompose(String command) {
    found=0;
    value1="";
    value2="";
    value3="";
    for(int i=0; i<=command.length()-1; i++){
        if(command.charAt(i)==' '){
            found+=1;
        }
        else{
            if(found==0){value1+=command.charAt(i);}
            if(found==1){value2+=command.charAt(i);}
            if(found==2){value3+=command.charAt(i);}
        }
    }
}
```

FIGURE 3.2 – Fonction de décomposition

On découpe la commande reçue à chaque espace trouvé en un maximum de trois "values", laissant place à de nombreuses possibilités. Une fois la commande décomposée, on appelle la fonction correspondante. Voici deux exemples :

```
if(value1=="help"){
    lu=true;
    Serial.print("Command list :\n");
    Serial.print("open = Set all servos to 180 degrees\n");
    Serial.print("close = Set all servos to 0 degrees\n");
    Serial.print("set X Y = Sets servo n°X to Y degrees\n");
}

if(value1=="open"){
    lu=true;
    open();
}
```

FIGURE 3.3 – Autres fonctions

3.3 Le bluetooth et MIT App Inventor 2

Nous disposions à la dixième séance d'un projet fonctionnant à partir d'un terminal. Il nous restait donc 2 séances pour intégrer la fonctionnalité bluetooth afin de rendre l'utilisation de la main plus ludique et accessible. N'ayant jamais réalisé d'application pour smartphone par le passé, nous avons alors effectué des recherches sur internet à ce sujet. Le résultat le plus fréquent était MIT App Inventor 2.

Et en effet, MIT App Inventor 2 correspondait tout à fait à nos besoins puisqu'il est parfaitement compatible avec Arduino Uno et un module HC-05. Dès que nous avons disposés de ce capteur bluetooth HC-05, nous avons pu développer l'application sans trop de soucis :

Au final notre application se connecte au module bluetooth qui lui même est branché en série à la carte Arduino. Elle présente également deux boutons : Open et Close qui permettent logiquement d'ouvrir et de fermer la main, ainsi que 5 sliders aussi appelés « ascenseurs », qui sont associés à chaque doigts. On peut déplacer les ascenseurs le long d'une barre, ce qui permet à chacun des doigts d'adopter une position de 0 à 180 degrés.

La partie code nous a en revanche demandé une certaine réflexion : En effet nous sommes uniquement capables d'envoyer un byte (donc un entier compris entre 0 et 255) depuis notre application jusqu'à notre carte Arduino. Pour encoder les deux boutons c'était relativement simple, nous envoyons un octet de valeur 0 pour ouvrir, et un octet de valeur 1 pour fermer. Mais pour coder les ascenseurs, c'était bien plus délicat puisque chacun d'entre eux ne fait qu'envoyer un chiffre entre 10 et 170, il était donc impossible à la carte d'associer un chiffre reçu au doigt correspondant sans manipulation de notre part !

La solution a d'abord été d'ajouter 200 à l'ascenseur numéro 2, 400 au numéro 3, etc...

Ainsi :

- l'ascenseur 1 envoie une valeur comprise entre 10 et 170.
- l'ascenseur 2 envoie une valeur comprise entre 210 et 370.
- l'ascenseur 3 envoie une valeur comprise entre 410 et 570.
- l'ascenseur 4 envoie une valeur comprise entre 610 et 770.
- l'ascenseur 5 envoie une valeur comprise entre 810 et 970.

De cette manière, tous les intervalles sont disjoints. C'est-à-dire que deux ascenseurs différents ne peuvent émettre la même valeur. Enfin il ne nous restait plus qu'à diviser toutes les valeurs par 4 et à les arrondir avant de les envoyer à la carte puisque nous ne nous pouvions envoyer qu'un entier de 255 maximum ($970 \div 4 = 242,5 \leq 255$).

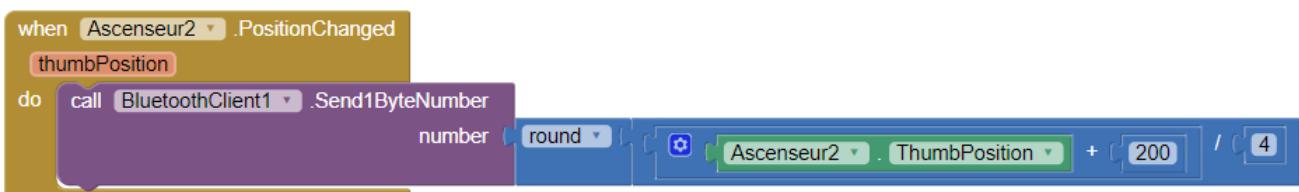


FIGURE 3.4 – Un exemple de bloc pour l'ascenseur 2 dans MIT App Inventor

Les valeurs sont donc transmises par bluetooth à notre carte Arduino qui interprète les valeurs pour définir les rotations des servomoteurs ou les fonctions à appeler. A la réception de celle

ci le programme Arduino les multipliera par 4 pour retrouver la valeur initialement envoyée à MIT App Inventor 2.

```
int command = bluetooth.read()*4;

if(command > 10 && command < 200) {
    angle=command-0;
    setServo(1, angle);
}
if(command > 200 && command < 400) {
    angle=command-200;
    setServo(2, angle);
}
```

FIGURE 3.5 – Un extrait du code Arduino

Conclusion

Nous pensons avoir réussi ce projet dans son ensemble malgré l'énorme quantité de travail qu'il a fallut fournir. En effet nous avons été surpris par l'intérêt que nous avons développé, puisque ce projet nous a tellement captivés que nous n'avons pas compté les heures de travail en dehors de Polytech, et sans cela il aurait été impossible de le mener à son terme !

Grâce à ce projet, nous avons également eu un aperçu de plusieurs domaines qui sont la mécanique avec l'impression 3D, la découpe laser et la modélisation sous Catia, l'électronique avec le circuit et enfin l'informatique avec l'utilisation du langage Arduino et le développement d'une application avec App Inventor.

Au final, pouvoir contrôler grâce à notre application la main que nous avons créée de toutes pièces nous a vraiment apporté satisfaction puisque cela marquait la réussite de notre projet.

Nous pensons que ce projet pourrait être amélioré avec la création d'un pouce qui ne serait pas fixé à 45 degrés de la paume, mais qui serait orientable ; ou encore l'ajout d'un poignet et la modélisation d'un avant bras pour masquer les servomoteurs et la carte Arduino.

Annexe A

Liens utiles

Voici une petite liste d'url intéressantes au sujet de ce projet :

- www.polytech.univ-tours.fr
- www.thingiverse.com
- inmoov.fr/hand-and-forarm/
- <https://www.amazon.fr/gp/product/B01G9KSAF6>
- <http://appinventor.mit.edu/explore/>
- <http://demonstrations.wolfram.com/ShapingARoadAndFindingTheCorrespondingWheel/>

Annexe B

Fiche de suivi de projet PeiP

Séance n° 1 du 19/01/2018	Nous avons effectué des recherches sur internet sur des projets similaires. Nous avons retenu des résultats intéressants, notamment le projet « Inmoov » qui utilise des servomoteurs et une carte Arduino.
Séance n° 2 du 26/01/2018	Nous avons commencé à représenter notre main à l'aide de dessins et avons travaillé plus en détail sur le schéma de la paume. Ensuite modélisation de celle ci sous Catia.
Séance n° 3 du 02/02/2018	Nous avons fait une représentation schématique d'un doigt. Au final, pour se simplifier le travail, le modèle de celui ci sera le même pour l'index, le majeur, l'annulaire et l'auriculaire. Seul le pouce aura un design différent.
Séance n° 4 du 09/02/2018	Pendant le week-end du 3 et 4 février, Colin a pu imprimer chez lui une première version (probablement non définitive) du doigt. Pendant cette séance, nous avons réalisé des modifications et des retouches que nous n'avions pas envisagées et que l'on a pu remarquer grâce au premier doigt imprimé. Il en a suivi une réflexion sur un modèle de poulie.
Séance n° 5 du 16/02/2018	A partir du doigt imprimé, nous avons fait un graphe de mesure de la longueur du chemin parcouru par chaque fil dans un doigt. Suite à cela, nous avons pu calculer le rayon de notre poulie et designer un premier modèle.
Séance n° 6 du 23/02/2018	Nous avons commencé notre séance par la découpe laser de la première poulie qui est un échec. Cela nous a permis d'imaginer une poulie circulaire à double rigole. Début de design de cette nouvelle poulie.
Séance n° 7 du 09/03/2018	Pendant les vacances de Février, Colin a imprimé 4 poulies identiques, modélisé le pouce et effectué des retouches sur le premier modèle du doigt. Suivi d'une impression de toutes les pièces de la main : La paume et 5 nouveaux doigts. Lors de cette 7ème séance, nous avons donc assemblé les différentes pièces de la main avec la visserie et les fils. Et avons amorcé le début de conception d'un support pour les servomoteurs.

Séance n°8 du 16/03/2018	Nous avons terminé le design d'un support pour les servomoteurs de façon à ce qu'ils prennent le moins de place possible. Et avons également fait les mesures et le design de la poulie du pouce qui est différente des 4 autres.
Séance n°9 du 23/03/2018	Dans la semaine, impression du support au DI. Nous avons modélisé une fixation en plexiglas afin de fixer les servomoteurs par rapport aux doigts.
Séance n° 10 du 30/03/2018	Nous avons pu découper au laser notre fixation et sommes ensuite allés à l'atelier du DMS pour modifier notre support des servomoteurs.
Séance n° 11 du 06/04/2018	Nous avons profité du week-end de Pâques pour concevoir et imprimer un guide pour les fils et écrire un code Arduino fonctionnant avec le terminal. Lors de cette 11ème séance, nous avons réalisé une application bluetooth sur MIT App Inventor 2, mais avons rencontré un problème de compatibilité entre le bluetooth low energy et MIT App Inventor 2. Nous avons donc acheté un capteur bluetooth classique et avons pu terminer l'application et modifier le code Arduino afin de le rendre compatible avec le capteur bluetooth.
Séance n° 12 du 13/04/2018	Le projet ayant aboutit, nous avons fait une vidéo de démonstration, et avons commencé notre rapport.

Main robotique imprimée en 3D avec contrôle Bluetooth

Rapport de projet S4

Résumé : Le projet consiste à concevoir et à construire une main robotique entièrement imprimée en 3D. Il est également entièrement actionné en utilisant 5 servomoteurs pour déplier chaque doigt individuel. Le mouvement peut être contrôlé par un smartphone en utilisant l'application bluetooth conçue pour le main.

Mots clé : main, CATIA, impression 3D, bluetooth, Arduino, application, servomoteurs

Abstract : The project involves designing and building a robotic hand entirely 3D printed. It is also fully driven using 5 servomotors to unfold each individual finger. The movement can be controlled by a smartphone using the bluetooth application designed for the hand.

Keywords : hand, CATIA, 3D printing, bluetooth, Arduino, application, servomotors

Auteur(s)

Colin Troisemaine

[colin.troisemaine@etu.univ-tours.fr]

Quentin Levieux

[quentin.levieux@etu.univ-tours.fr]

Encadrant(s)

Guénhaël Le Quilliec

[guenhael.lequilliec@univ-tours.fr]

Polytech Tours

Département Peip

Ce document a été formaté selon le format EPUProjetPeiP.cls (N. Monmarché)

École Polytechnique de l'Université de Tours
64 Avenue Jean Portalis, 37200 Tours, France
<http://www.polytech.univ-tours.fr>