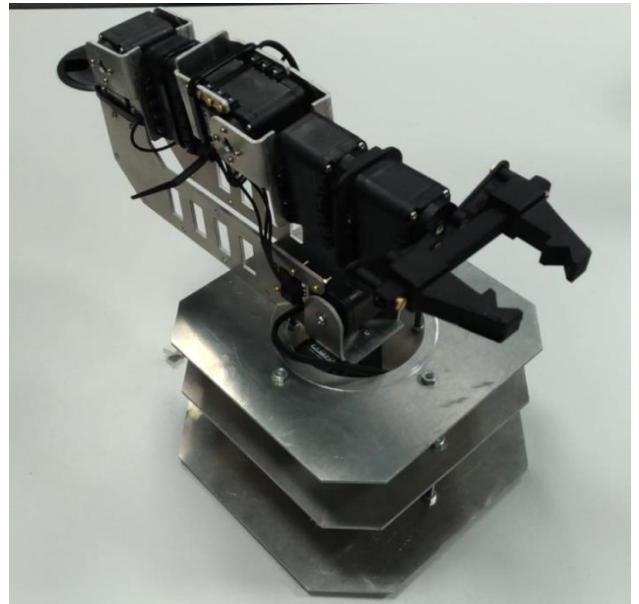
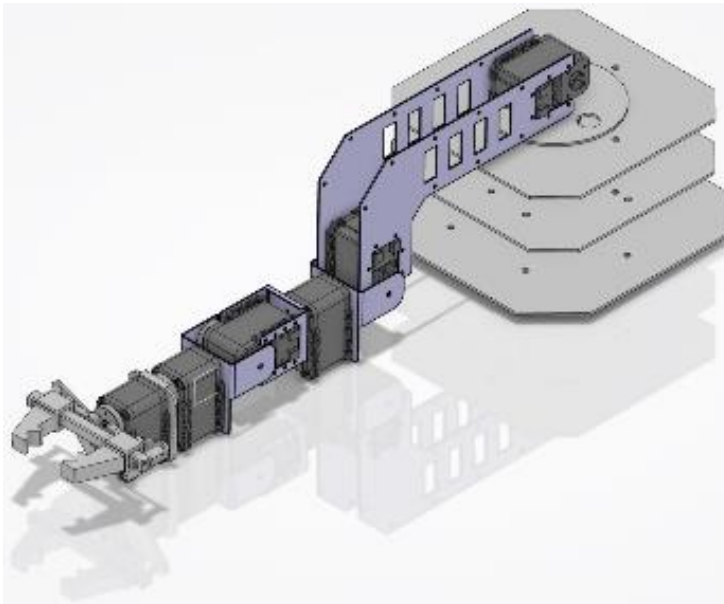


Ménélik Epinat, Gaëtan Parisot,
Maxime Lamétrie et Matthéo Marchand

Projet B.R.A.V.O: Conception et réalisation d'un bras robots 6 axes d'aide à la personne

31 janvier 2019



I. Table des matières

I.	Remerciements	4
II.	Introduction	4
A.	Présentation de l'association My Human Kit	4
B.	Présentation du projet Bras Robotisé Auxiliaire de Vie Opensource (B.R.A.V.O)	4
III.	État de l'art	5
IV.	Méthodologie et organisations.....	6
V.	Cahier des charges.....	9
VI.	Partie Mécanique	10
A.	Dimensionnement mécanique	10
B.	Conception	17
C.	Prototypage	20
D.	Assemblage	23
E.	Création du mécanisme	26
F.	Simulation dynamique	28
G.	Simulation - Dymola :.....	29
VII.	Partie Électronique.....	35
A.	Arduino Uno	35
1.	Présentation.....	35
2.	Structure de la carte.....	35
3.	Auto-formation : OpenClassroom	36
4.	Programmation d'un servomoteur	37

5.	Programme N°2 : servomoteur + potentiomètre.....	38
6.	Programme N°3 : servomoteurs + joystick.....	40
B.	Arbotix-M.....	41
1.	Présentation.....	41
2.	Structure de la carte.....	42
3.	Programmation Arbotix-M.....	43
4.	Problème rencontré	44
C.	OpenCM 9	45
VIII.	Coûts du projet	46
IX.	Conclusion.....	46
X.	Table des figures.....	48

I. Remerciements

Nous tenons premièrement à remercier Nicolas POUSSET pour la confiance qu'il nous a accordé pour la réalisation de son projet, Hugues AUBIN pour toute son aide autour de l'écriture du code, ainsi que tous les collaborateurs de l'association My Human Kit pour avoir apporté ce projet.

Nous tenons à remercier toute l'équipe du Pôle mécanique et technologique de l'université de Rennes 1, plus particulièrement Bruno LERAY pour nous avoir mis à disposition le matériel nécessaire à la conception de notre projet et Fabienne BLANCHET pour l'utilisation de la découpe jet d'eau. Nous remercions également Erwan BOUGUENNEC pour ses conseils sur 3DExperience et Nicolas VIGNERON pour sa disponibilité et ses conseils.

II. Introduction

A. Présentation de l'association My Human Kit

Fondée par Nicolas HUCHET, My Human Kit (MHK) est une association visant à développer la santé pour tous, à travers le partage et la fabrication d'aides techniques aux handicaps. Les projets sont réalisés pour et avec les personnes concernées. Elle invite les personnes en situation de handicap à s'impliquer dans la facilitation de leur quotidien à travers un projet dont elles sont acteurs principales.



Figure 1 : Logo My Human Kit

B. Présentation du projet Bras Robotisé Auxiliaire de Vie OpenSource (B.R.A.V.O)

Dans le cadre de notre master 2 mécanique mention mécatronique, nous avons réalisé un projet en lien avec cette spécialité. Un de nos camarades faisant partie de l'association My Human Kit, Gaëtan PARISOT, nous a fait part d'un besoin. L'objectif de ce projet est de fabriquer un bras robot adapté pour une personne handicapée. Le système s'adaptera sur un fauteuil roulant électrique. Nicolas POUSSET, porteur du projet et bénévole de l'association MHK, souhaiterait utiliser un bras robot pour effectuer diverses tâches du quotidien; ouvrir une porte, prendre une bouteille d'eau ou appuyer un bouton.



III. État de l'art

Le marché de la robotique est en pleine croissance en raison de son utilisation dans le domaine militaire, du médical et industriel. L'emploi des robots dans le domaine du handicap est devenu un sujet important dans notre société actuelle. C'est pour cette raison que nous nous sommes intéressés à la société Kinova, une entreprise canadienne, spécialisée dans la fabrication de bras robot destinés aux personnes à mobilité réduite.



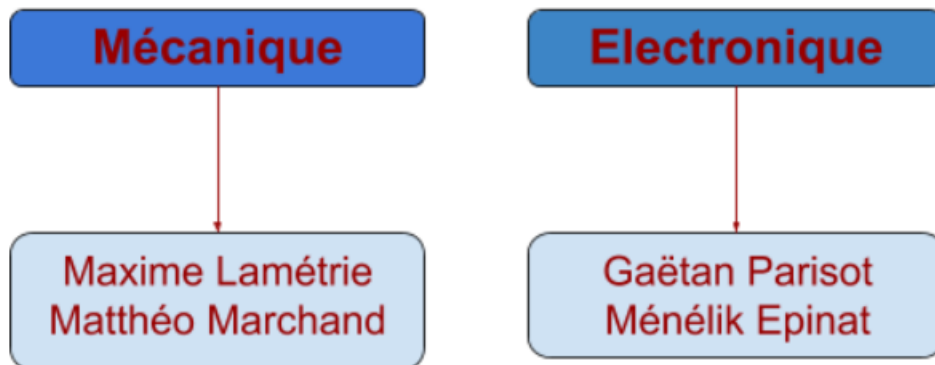
Figure 2 : Bras robotisé "Jaco"

Cette société propose un bras robot 6 axes, appelé « Jaco », pour un montant de 50 000€. Nous nous sommes intéressés à ses caractéristiques techniques ; sa portée, son poids, son outil de préhension, sa structure et les matériaux utilisés.

Outils de préhension		
Résumé des spécifications	3 doigts	2 doigts
Poids total	5,2 kg	5,0 kg
Charge utile (mi-portée)	1,6 kg	1,8 kg
Charge utile (temporaire / pleine portée)	1,3 kg	1,8 kg
Portée	90 cm	
Matériaux	Liaison : Fibre de carbone	Actionneurs : Aluminium
Vitesse linéaire maximale du bras	20 cm/s	
Tension d'alimentation	18 à 29 VCC	
Puissance moyenne	Mode fonctionnement 25 W Mode veille 5 W	
Puissance de pointe	100 W	
Résistance à l'eau	PX2 – Résistance à une fine pluie	
Température d'opération	-10°C à 40°C Peut être utilisé dans des conditions plus extrêmes, mais pour une durée limitée	

IV. Méthodologie et organisations

Afin de mieux répartir les différentes tâches de ce projet, nous avons décidé de scinder notre groupe en deux équipes.



Tout d'abord une équipe mécanique s'est orientée sur la conception et le prototypage. Ils avaient pour mission de trouver de nouvelles solutions afin de rendre le projet le moins coûteux possible et le plus simple à concevoir. Durant toute la durée du projet, ils ont pu se former aux différents ateliers de 3DExperience. Grâce à ce nouveau logiciel, ils ont pu gérer les différentes phases de conception et de simulation en restant toujours sur la même plateforme.

En collaboration avec My Human Kit, l'équipe électronique s'est concentrée sur les différents choix de composants, la formation Arduino et le codage. Ils ont donc pu se former au langage C et travailler sur des cartes Arduino qui se développent beaucoup depuis quelques années et ce, même en entreprise.

Pour gérer ce projet, nous avons tenu à rester sur la plateforme 3DExperience. De ce fait, nous nous sommes donc servi de l'application "Project Management" disponible dans le rôle "Governance and Project Management" de ce logiciel. Nous avons notamment pu créer un diagramme de Gantt, ce qui nous a permis d'appliquer nos notions de gestion de projets vus avec Nicolas VIGNERON et de mieux comprendre l'esprit de collaboration de 3DExperience.

Espace collaboratif



Figure 3 : Management sur 3DExperience

Nous avons choisi de séparer ce Gantt en trois parties : la partie Mécanique avec les différentes tâches pour l'équipe mécanique et la partie Électronique avec les différentes tâches de l'équipe électronique. Enfin, une partie se consacrera sur la documentation et réunion avec les différentes tâches communes aux deux équipes : les réunions, l'écriture du rapport et les rendez-vous chez My Human Kit.

Name	Type	M	Dependency	State	% Comp	Progress	Duration	Start Date	End Date	Duration	Start Date	End Date	Owner	Level
Project S...	Project S...	0	0	In Work	79.3	104.0 Days	Sep 10, 2018	Jan 31, 2019	0.0	0.0	Oct 11, 2018		Médecine Eperlat	
Bras Robot	Task	1	0	In Work	71.4	90.0 Days	Sep 10, 2018	Jan 21, 2019	0.0	0.0	Oct 11, 2018		Médecine Eperlat	1.1
Etude du bras Jaco	Task	2	0	Complete	100.0	7.0 Days	Sep 10, 2018	Sep 18, 2018	1.0	1.0	Oct 11, 2018	Oct 15, 2018	Médecine Eperlat	1.1
Conception Bras proto	Task	3	0	In Review	100.0	30.0 Days	Sep 17, 2018	Nov 2, 2018	16.0	16.0	Oct 11, 2018	Nov 5, 2018	Médecine Eperlat	1.2
Modification 3D	Task	4	0	Complete	100.0	21.0 Days	Sep 17, 2018	Oct 15, 2018	16.0	16.0	Oct 11, 2018	Nov 5, 2018	Médecine Eperlat	1.2.1
Prototypage	Task	5	4FS+0.0 d	Complete	100.0	14.0 Days	Oct 16, 2018	Nov 2, 2018	1.0	1.0	Nov 5, 2018	Nov 5, 2018	Médecine Eperlat	1.2.2
Conception Bras robot final	Task	6	2FS+0.0 d	In Work	50.0	96.0 Days	Nov 5, 2018	Jan 21, 2019	0.0	0.0	Jan 28, 2019		Médecine Eperlat	1.3
Mécatronique Général	Task	7	0	Complete	100.0	28.0 Days	Nov 5, 2018	Dec 12, 2018	1.0	1.0	Jan 28, 2019	Jan 28, 2019	Médecine Eperlat	1.3.1
Modélisation 3D	Task	8	7FS+0.0 d	Ready	0.0	14.0 Days	Dec 15, 2018	Jan 1, 2019	0.0	0.0			Médecine Eperlat	1.3.2
Prototypage	Task	9	0FS+0.0 d	Ready	0.0	14.0 Days	Jan 2, 2019	Jan 21, 2019	0.0	0.0			Médecine Eperlat	1.3.3
Electronique	Task	10	0	In Work	96.1	91.0 Days	Sep 10, 2018	Jan 14, 2019	0.0	0.0	Oct 11, 2018		Médecine Eperlat	2
Etudes des composants	Task	11	0	Complete	100.0	14.0 Days	Sep 10, 2018	Sep 22, 2018	21.0	21.0	Nov 12, 2018	Nov 12, 2018	Médecine Eperlat	2.1
Firmware Arduino	Task	12	11FS+0.0 d	Complete	100.0	20.0 Days	Sep 28, 2018	Nov 5, 2018	21.0	21.0	Oct 11, 2018	Nov 12, 2018	Médecine Eperlat	2.2
Recherches Code Sources et test sur le prototype	Task	13	12FS+0.0 d	Complete	100.0	20.0 Days	Nov 7, 2018	Dec 14, 2018	36.0	36.0	Oct 11, 2018	Dec 3, 2018	Médecine Eperlat	2.3
Amélioration/adaptation du code	Task	14	13FS+0.0 d	Complete	100.0	14.0 Days	Nov 17, 2018	Jan 3, 2019	1.0	1.0	Jan 28, 2019	Jan 28, 2019	Médecine Eperlat	2.4
Tests	Task	15	14FS+0.0 d	In Work	50.0	7.0 Days	Jan 4, 2019	Jan 14, 2019	0.0	0.0	Jan 28, 2019		Médecine Eperlat	2.5
Documentation Réunion	Task	16	0	In Work	75.8	104.0 Days	Sep 10, 2018	Jan 31, 2019	0.0	0.0	Oct 11, 2018		Médecine Eperlat	3
Rapport	Task	17	0	In Work	88.0	104.0 Days	Sep 10, 2018	Jan 31, 2019	0.0	0.0	Jan 28, 2019		Médecine Eperlat	3.1
WikiLab	Task	18	0	In Work	70.0	104.0 Days	Sep 10, 2018	Jan 31, 2019	0.0	0.0	Oct 11, 2018		Médecine Eperlat	3.2
Réunion	Task	19	0	Complete	100.0	7.0 Days	Oct 15, 2018	Oct 23, 2018	1.0	1.0	Nov 5, 2018	Nov 5, 2018	Médecine Eperlat	3.3
Soutenance	Milestone	20	0	Preliminary	0.0	0.0 Days	Jan 31, 2019	Jan 31, 2019	0.0	0.0			Médecine Eperlat	4

Figure 4 : Définition des tâches sur 3DExperience

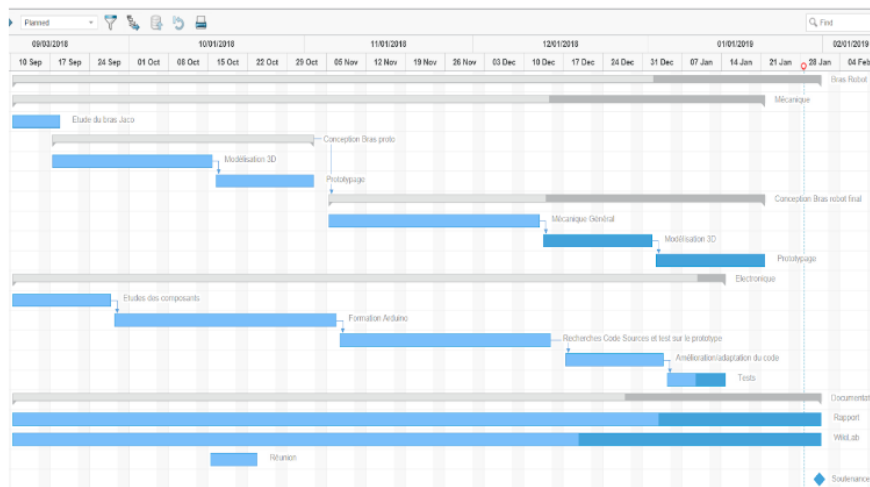


Figure 5 : Gantt sur 3DExperience

Name	Type	State
Filter		
Bras Robot	Project Space	In Work
Mécanique	Task	In Work
Etude du bras J...	Task	Comple
Conception Br...	Task	In Review
Modélisation...	Task	Comple
Prototypage	Task	Comple
Conception Br...	Task	In Work
Mécanique...	Task	Comple
Modélisation...	Task	Ready
Prototypage	Task	Ready
Electronique	Task	In Work
Etudes des co...	Task	Comple
Formation Ardu...	Task	Comple
Recherches Co...	Task	Comple
Amélioration/ad...	Task	Comple
Tests	Task	In Work
Documentation -...	Task	In Work
Rapport	Task	In Work
WikiLab	Task	In Work
Réunion	Task	Comple
Soutenance	Milestone	Prelimina

Figure 6 : Évolution des différentes tâches

Nous avons séparé la partie Mécanique en trois grandes parties.

Tout d'abord, une étude des solutions déjà existantes. Actuellement le leader du marché est le bras Jaco, c'est donc sur ce produit que nous avons fait un état de l'art. Ensuite la conception d'un prototype avec deux sous-tâches, sa CAO et son fabrication. Et enfin la conception du bras robot final, subdivisé en trois, les calculs de dynamique du robot, la CAO du modèle final et son prototypage.

La partie Électronique a été séparé en cinq grandes parties. Premièrement, l'étude des composants fournies par My Human Kit de manière à transmettre les données nécessaires à l'équipe Mécanique pour qu'ils optimisent leur CAO. Ensuite une formation pour apprendre les bases du code Arduino. Après cela, une recherche de codes sources pour notre bras robot en faisant appel à la large communauté d'Arduino. Des recherches d'amélioration du code seront également réalisées afin d'optimiser le

contrôle du bras robot. Et enfin une phase de test pour finaliser au mieux le code.

V. Cahier des charges

Suite à notre première visite au sein de My Human Kit et de notre entretien avec le porteur de projet Nicolas POUSSET, nous avons réalisé un cahier des charges.

Ce cahier des charges comporte de nombreux points regroupés dans différentes parties. La contrainte première de ce projet est de sécuriser l'utilisateur. Le bras devra donc posséder des bridages logiciels et des butées mécaniques afin de ne pas heurter et blesser l'utilisateur. Un arrêt d'urgence devra être facilement actionnable en cas de problème. Des limiteurs de couple seront adjoints afin d'éviter de détériorer le système et le fauteuil. La gestion du câblage sera de même prise en compte.

Le bras robot aura pour but de saisir des objets de faible poids (verre, bouteille), ouvrir des portes et actionner des interrupteurs. Pour cette partie fonctionnelle, plusieurs contraintes sont alors discernables. L'angle d'attaque de la pince manipulatrice devra être relativement ample afin de ne pas entrer en collision avec les éléments extérieurs comme du mobilier. On estime aussi qu'un rayon d'action de 1,50 mètre du bras permettra d'être relativement à l'aise pour la préhension de la plupart des objets.

Une autre catégorie de contrainte traitera de l'environnement lié à l'évolution du système. La résistance à une collision avec l'environnement (mobilier) du bras prendra une part importante lors du développement. Le robot sera tenu de résister à tout genre d'intempéries tel que de fortes et faibles températures/taux d'humidité.

Une partie examinera les contraintes budgétaires de ce projet. Le coût total d'un tel système devra être accessible au plus grand nombre financièrement. Tous les composants (matériaux, matériels électronique, outillages.....) devront être peu onéreux.

Enfin nous présenterons une partie sur le caractère open-source du projet. La conception ainsi que la fabrication seront explicitement détaillées afin de reproduire au mieux le système, voir l'améliorer. Les matériaux devront être travaillés dans la plupart des FabLabs ou par des amateurs dits makers. La programmation devra être claire et concise durant la présentation du projet afin de guider au mieux les novices. Chaque fichier, process, procédure devront être accessibles à tous et gratuitement.

VI. Partie Mécanique

A. Dimensionnement mécanique

Dans le cadre du projet, l'association My Human Kit nous a fourni des servomoteurs de la marque Dynamixel modèle AX-12. Ce sont des actuateurs qui comportent un réducteur, un contrôleur et un pilote. Ces servomoteurs possèdent un identifiant propre et permettent de communiquer en série avec d'autres servomoteurs.



Figure 7 : Servomoteur Dynamixel AX-12A



Figure 8 : Fonctionnement du servomoteur

Les caractéristiques techniques de la gamme Dynamixel sont les suivantes :

- Une LED d'état permettant de visualiser l'état du servomoteur.
- Communication en série avec plusieurs servomoteurs.
- Grande précision de position (dans le cas de notre projet, les servomoteurs AX-12 disposent de 1024 positions).
- Retour d'information (température, position actuelle, vitesse ...).
- Une alarme en cas d'un dépassement d'une limite (température, tension ...).

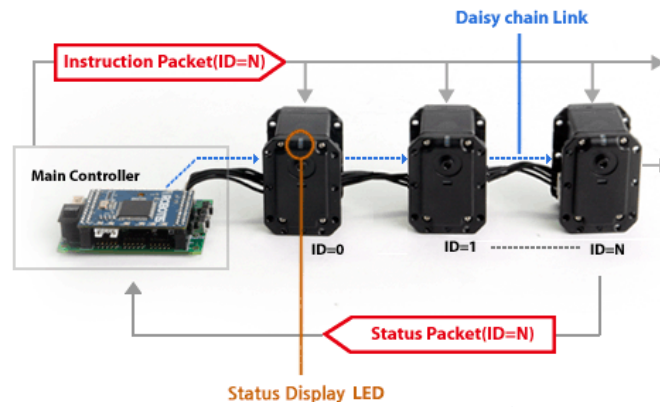


Figure 9 : Fonctionnement en série

Caractéristiques Techniques du servomoteur Dynamixel AX-12

- Poids : 53.5 grammes
- Dimension : 32mm * 50mm * 40mm
- Ration du réducteur : 254 : 1
- Couple statique : 1.5 N.m à 12.0 Volts et 1.5 Ampères
- Vitesse en charge à vide : 59 tr/min à 12 Volts.
- Feedback : Position, Température, Charge, Tension, etc.
- Matériaux : ABS
- Température : -5°C à +70°C
- Tension : 9 à 12 Volts, 11,1 Volts recommandé
- Commande par paquet
- Protocole : communication série asynchrone en semi-duplex. Sur 8 bits avec un bit de stop et sans bit de parité
- Connecteur 3 broches

L'intérêt premier de ce genre de servomoteurs est qu'ils sont asservis par un système interne facilitant la programmation de ceux-ci. De plus ils offrent de nombreux retours d'informations tels que la charge et la position moteur. Ils peuvent être câblés en série ce qui facilite la gestion du câblage en supprimant de nombreux fils. Il s'agit d'un facteur de fiabilité. La programmation ne consistera alors qu'à attribuer un ordre à ces moteurs, sans se soucier de leur asservissement, c'est-à-dire une position.

La conception de notre prototype de bras robot s'est alors articulé autour des servomoteurs. Nous avons effectué un rapide calcul de couple afin d'optimiser les dimensions de nos pièces et bien sûr confirmé la faisabilité d'un prototype.

Données					
Désignations	Abréviations	Valeurs	Unités		
Pesanteur	g	9,81	m/s ²		
Masse Moteur 1	Mas Mot 1	0,0503	Kilogrammes	0,493443	Newtons
Masse Moteur 2	Mas Mot 2	0,0503	Kilogrammes	0,493443	Newtons
Masse Moteur 3	Mas Mot 3	0,0503	Kilogrammes	0,493443	Newtons
Masse Moteur 4	Mas Mot 4	0,0503	Kilogrammes	0,493443	Newtons
Masse Moteur 5	Mas Mot 5	0,0503	Kilogrammes	0,493443	Newtons
Masse Moteur 6	Mas Mot 6	0,0503	Kilogrammes	0,493443	Newtons
Masse Moteur 7	Mas Mot 7	0,0503	Kilogrammes	0,493443	Newtons
Masse Plaque1	MasPlq 1	0,05	Kilogrammes	0,4905	Newtons
Masse U1	MasU1	0,017	Kilogrammes	0,16677	Newtons
Masse U2	MasU2	0,016	Kilogrammes	0,15696	Newtons
Masse U3	MasU3	0,017	Kilogrammes	0,16677	Newtons
Masse U4	MasU4	0,01	Kilogrammes	0,0981	Newtons
Masse Support Moteur	MasSupMot	0,01	Kilogrammes	0,0981	Newtons
Masse Pince	MasO	0,01	Kilogrammes	0,0981	Newtons
Distance entre axes Moteur 2 et Moteur 3 selon x	M2M3	0,15	Mètres		
Distance entre axes Moteur 3 et Moteur 4 selon x	M3M4	0,04	Mètres		
Distance entre axes Moteur 4 et Moteur 5 selon x	M4M5	0,05	Mètres		
Distance entre axes Moteur 5 et Moteur 6 selon x	M5M6	0,05	Mètres		
Distance entre axes Moteur 6 et Moteur 7 selon x	M6M7	0,03	Mètres		
Distance entre l'axe Moteur 2 et le centre de gravité de la plaque	M2/P	0,10	Mètres		
Distance entre l'axe Moteur 3 et le centre de gravité de la U2	M3/U2	0,01	Mètres		
Distance entre l'axe Moteur 4 et le centre de gravité de la U3	M4/U3	0,03	Mètres		
Distance entre l'axe Moteur 5 et le centre de gravité de la U4	M5/U4	0,03	Mètres		
Distance entre l'axe Moteur 6 et le centre de gravité du support moteur pince	M6/SupMPince	0,03	Mètres		
Distance entre l'axe Moteur 7 et le centre de gravité de la pince	M7/Pince	0,05	Mètres		
Distance approximative entre M2/M6		0,372	Mètres		
Masse approximative du système		0,462	Kilogrammes		
Resultat					
Designation	Maintien statique				
Couple statique admissible Moteur 1 (Rotation système)	CsMot1	0,112	N/m		
Couple statique admissible Moteur 2	CsMot2	0,801	N/m		
Couple statique admissible Moteur 3	CsMot3	0,275	N/m		
Couple statique admissible Moteur 4	CsMot4	0,181	N/m		
Couple statique admissible Moteur 5	CsMot5	0,088	N/m		
Couple statique admissible Moteur 6	CsMot6	0,026	N/m		
Couple statique admissible Moteur 7 (Serrage pièce)	CsMot7S	0,005	N/m		

Figure 10 : Table paramétrée des couples moteurs

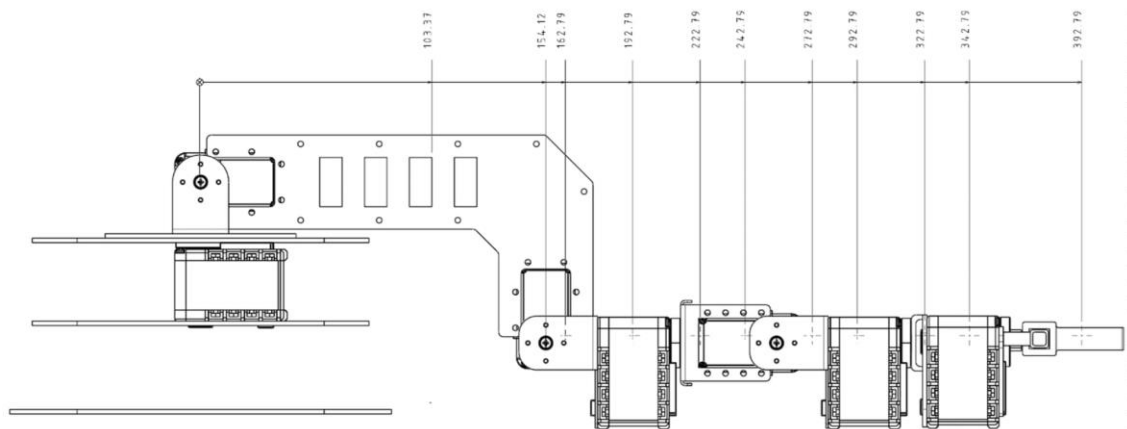


Figure 11 : Dimension du bras robot

Calcul du couple de maintien nécessaire :

$$C_{mot2} = \overrightarrow{M2M3} * Masse_{mot} + \overrightarrow{M2M4} * Masse_{mot} + \overrightarrow{M2M5} * Masse_{mot} + \overrightarrow{M2M6} * Masse_{mot} \\ + \overrightarrow{M2M7} * Masse_{mot} + \overrightarrow{M2M_{plaque}} * Masse_{plaque} * 2 + \overrightarrow{M2M_{u2}} * Masse_{u2} + \overrightarrow{M2M_{u3}} * Masse_{u3} \\ + \overrightarrow{M2M_{u4}} * Masse_{u4} + \overrightarrow{M2M_{pince}} * Masse_{pince}$$

Les différentes valeurs utilisées lors de notre calcul se trouvent en annexe figure ().

Le moteur 2 étant capable de supporter l'ensemble de la masse du bras robot à l'arrêt, nous pouvons en conclure que les autres moteurs seront en mesure de supporter le couple qu'il leur est appliqué.

Après avoir calculé le couple de maintien, nous voulions savoir si les moteurs seraient en mesure de soulever le bras et de le faire bouger. Nous avons pour cela réalisé un calcul de dynamique simplifié. Nous avons fait l'hypothèse que le bras était entièrement. Si le moteur 2 est en mesure de soulever entièrement le bras, les autres moteurs pourront également faire bouger la partie du bras qui leur est associée.

On pose J la matrice d'inertie qui sera de la forme :

$$J = \begin{pmatrix} A & 0 & 0 \\ 0 & B & 0 \\ 0 & 0 & C \end{pmatrix}$$

On a une matrice diagonale car nous avons une double symétrie suivant l'axe x et l'axe z .

$$A = \iiint (y^2 + z^2) dm$$

Avec

$$dm = \rho dx dy dz$$

$$= \rho \int_{-\frac{a}{2}}^{\frac{a}{2}} \int_{-\frac{b}{2}}^{\frac{b}{2}} \int_0^l (y^2 + z^2) dx dy dz$$

$$\begin{aligned}
&= \rho \int_{-\frac{a}{2}}^{\frac{a}{2}} \int_{-\frac{b}{2}}^{\frac{b}{2}} \int_0^l y^2 dx dy dz + \rho \int_{-\frac{a}{2}}^{\frac{a}{2}} \int_{-\frac{b}{2}}^{\frac{b}{2}} \int_0^l z^2 dx dy dz \\
&= \rho l b \frac{a^3}{12} + \rho l a \frac{b^3}{12} \\
&= m \frac{a^2}{12} + m \frac{b^2}{12} \\
&= \frac{m}{12} (a^2 + b^2)
\end{aligned}$$

$$\begin{aligned}
B &= \iiint (x^2 + z^2) dm \\
&= \frac{mb^2}{12} + \frac{ml^2}{3}
\end{aligned}$$

$$\begin{aligned}
C &= \iiint (y^2 + x^2) dm \\
&= \frac{ma^2}{12} + \frac{ml^2}{3}
\end{aligned}$$

$$\mathcal{E}_c(S_1/R_0) = \frac{1}{2} \{ \overrightarrow{\mathfrak{C}}_{S_1/R_0} \}_0 \{ \overrightarrow{\mathfrak{U}}_{S_1/R_0} \}$$

$$\frac{1}{2} \left\{ \frac{m \overrightarrow{V}_{G_1/R_0}}{\overrightarrow{\sigma}_0(S_1/R_0)} \right\}_0 \left\{ \frac{\overrightarrow{\Omega}(S_1/R_0)}{V_0 \in (S_1/R_0)} \right\}$$

$$= \frac{1}{2} \overrightarrow{\sigma}_{0(S_1/R_0)} \overrightarrow{\Omega}_{(S_1/R_0)}$$

$$\overrightarrow{\Omega}_{S_1/R_0} = \dot{q}_1 \overrightarrow{z}_0$$

$$\overrightarrow{\sigma}_{(0,S_1/R_0)} = J_{(0,S_1)} \overrightarrow{\Omega}_{S_1/R_0} + m \overrightarrow{0} \overrightarrow{G} \Delta \overrightarrow{V}_{(0,S_1/R_0)}$$

$$= \begin{pmatrix} A & 0 & 0 \\ 0 & B & 0 \\ 0 & 0 & C \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ \dot{q}_1 \end{pmatrix}$$

$$= C \dot{q}_1 \overrightarrow{z}_0$$

$$\mathcal{E}_c = \frac{1}{2} C \dot{q}_1^2 \vec{z}_0$$

$$\frac{\partial \mathcal{E}_c}{\partial \dot{q}_1} = 2C \dot{q}_1$$

$$\frac{d}{dt} \frac{\partial \mathcal{E}_c}{\partial \dot{q}_1} = 2C \ddot{q}_1$$

$$\frac{\partial \mathcal{E}_c}{\partial q_1} = 0$$

On obtient donc :

$$\Gamma = C \ddot{q}_1 + mgD$$

$$\ddot{q}_1 = \frac{\Gamma - mgD}{C}$$

Avec Γ le couple moteur égale à 1,5 N.m

- $M=0,460$ Kg
- $g=9,80$ m.s²
- $D=0,193$ m
- $C=0,028$

On obtient donc :

$$\ddot{q}_1 = 22,5 \text{ rad.s}^{-2}$$

Le robot aura donc une accélération de **22,5 rad.s⁻²** ce qui est largement suffisant pour déplacer notre bras. Nos différents calculs on permet de valider notre choix de moteur, le bras robot pourra donc bouger sans difficulté et supporter son propre poids, sans tombé au repos.

B. Conception

Nous avons choisi de réaliser les différentes pièces du prototype du bras robot 6 axes avec des tôles métalliques. En effet nous avons à notre disposition une machine de découpe par jet d'eau permettant la réalisation de pièces de façon relativement rapide. Des chutes de tôles d'aluminium 4 AUG était disponible en magasin. Certaines des tôles seront par la suite pliées.

On peut voir ci-dessous une représentation de nos différents composants sous l'atelier part design de 3DExpérience :

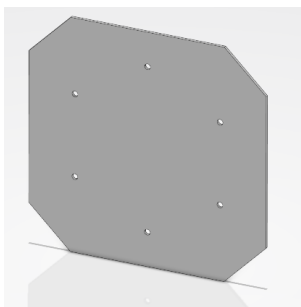


Figure 12 : Plaque 1

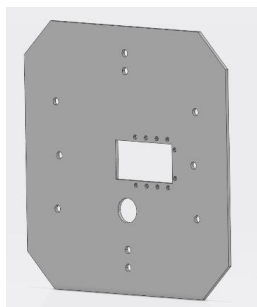


Figure 13 : Plaque 2

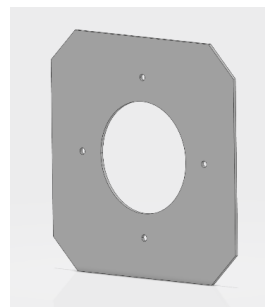


Figure 14 : Plaque 3

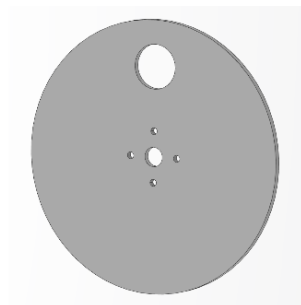


Figure 15 : Disque

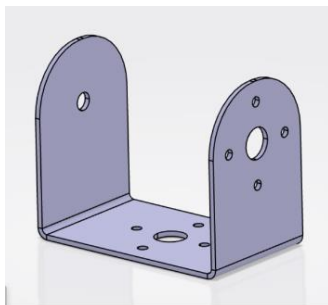


Figure 16 : U1

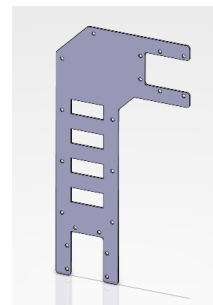


Figure 17 : Plaque bras

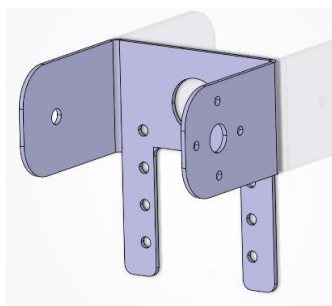


Figure 18 : U2

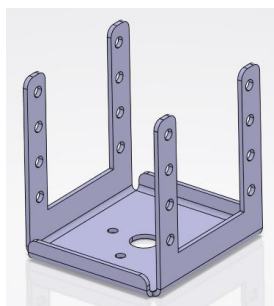


Figure 19 : U3

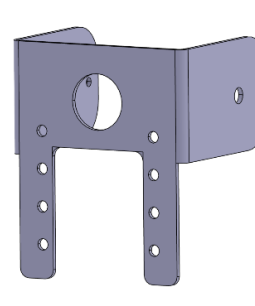


Figure 20 : U4

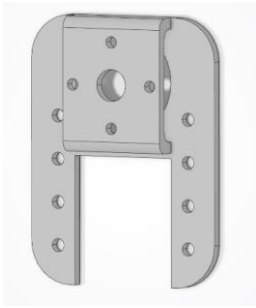


Figure 21 : Support moteur

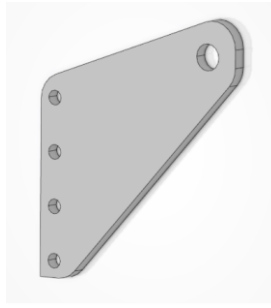


Figure 22 : Support moteur pince

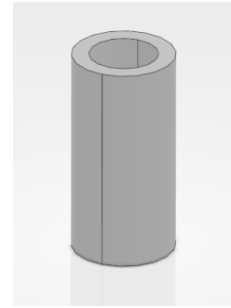


Figure 23 : Entretoise

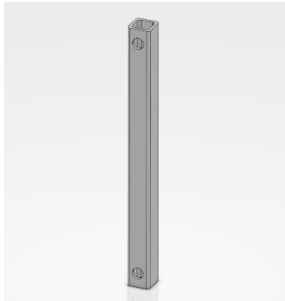


Figure 24 : Glissière

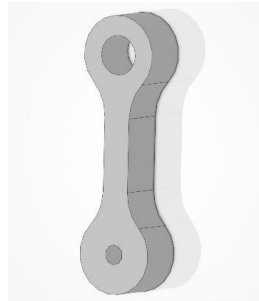


Figure 25 : Biellette

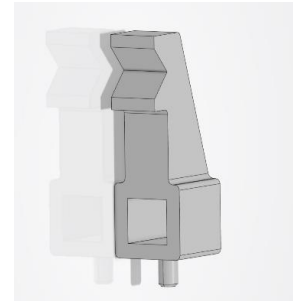


Figure 26 : Pince

Pour que le bras robot puisse accompagner son utilisateur dans son quotidien il a été décidé de l'installer directement sur son fauteuil roulant. C'est ce qui semblait le plus à même de correspondre à ses besoins et désirs. Pour cela nous sommes allés au plus simple, la base de notre robot se constitue de trois plaques en aluminium horizontales superposées.

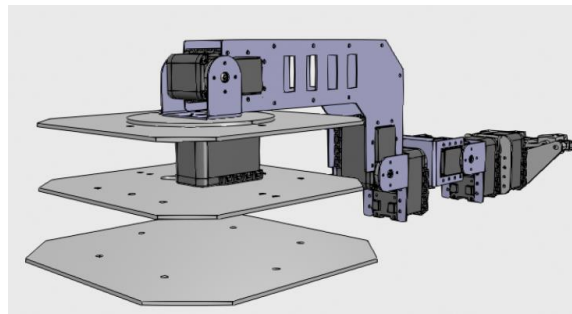


Figure 27 : Bras robot sous 3DExperience

L'espace entre la plaque intermédiaire et la plaque supérieur permettent de loger le premier servomoteur. Un disque est placé au dessus pour répartir les efforts sur la base et réduire la flexion de l'arbre du moteur 1. La plaque intermédiaire et la plaque inférieure peuvent se rapprocher à l'aide d'écrou et de vis autour de l'accoudoir. Ce système permet de s'adapter à de nombreux modèle de fauteuils.

Afin de réaliser la première moitié de notre bras, nous avons conçu deux plaques permettant la liaison entre les moteurs de la deuxième et la troisième articulations. La pièce a été évidée au maximum afin de réduire sa masse. Des perçages ont été disposés sur le pourtour de celle-ci afin de boulonner le tout pour améliorer la rigidité de l'ensemble. Il se révélera plus tard lors du montage que ceux-ci n'étaient pas utiles car la rigidité liée à la fixation des moteurs était suffisante.

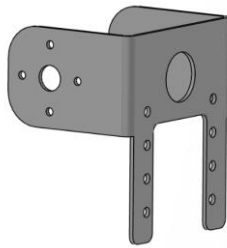


Figure 29 : U2

Les dernières pièces du robot seront conçues en fabrication additive par extrusion de fil chaud. C'est à dire les composants de l'outil.

Nous avons en effet choisi de tester ce procédé pour simplifier une partie de la conception ainsi que pour réaliser des tests sur l'efficacité des matériaux mis à disposition. Ce choix nous a permis, comme on peut le voir ci-contre, de réaliser une pièce complexe pour le support de la pince.

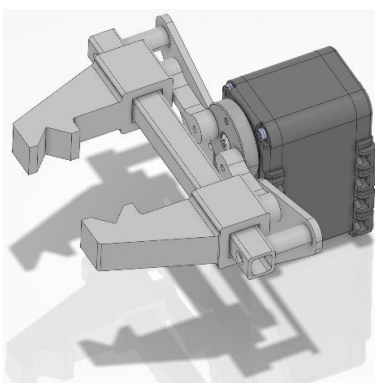


Figure 31 : Pince du bras robot

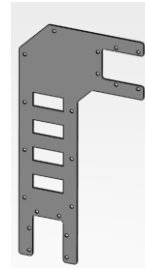


Figure 28 : Plaque bras

On observe ci-contre un support moteur que l'on retrouve fréquemment sur la structure du bras. De nombreux évidements sont présents sur cette pièce afin d'avoir un maximum d'accès au serrage des différentes vis de fixation. Chaque perçage dispose d'un jeu d'assemblage. Nous avons toutefois décidé d'ajouter un maximum de ceux-ci pour rattraper un éventuel jeu au niveau de la liaison.



Figure 30 : Support moteur

La pince est composée quant à elle d'une cinématique simple. Elle est composée de deux biellettes qui font translater des mâchoires sur une glissière. Les biellettes sont actionnées par la rotation du hauban du servomoteur.

Nous aurions pu choisir un système avec un train d'engrenage cependant nous doutions du bon glissement entre les roues de ce système. En effet l'imprimante 3D que nous possédons ne permet pas d'obtenir des états de surface assez lisse pour permettre un glissement sans trop d'effort.

C. Prototypage

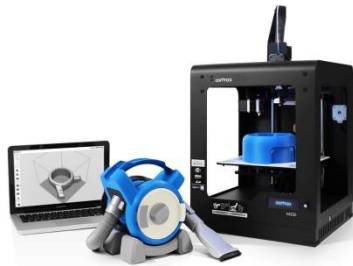


Figure 32 : Imprimante 3D "Zortrax"



Figure 33 : Découpeuse jet d'eau



Figure 34 : Plieuse mécanique

Lors de la réalisation de ce prototype, nous avons rencontré quelques problèmes. Nous n'avions pas prévu un délai d'une semaine pour recevoir les pièces de découpe.

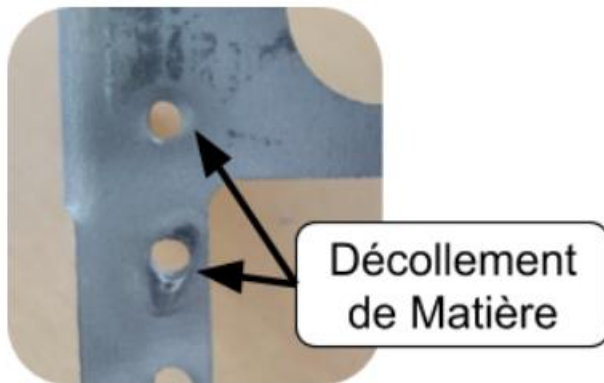


Figure 35 : Défaut N°1

Des problèmes liés à la découpe jet d'eau sont aussi apparus comme nous pouvons l'observer ci-contre. La surface de la tôle d'aluminium s'est désolidarisée de la matière. Il apparaît donc des cloques au niveau des perçages. Ceux-ci ont eu aussi tendance à s'ovaliser. Nous avons dû les reprendre par la suite à l'aide d'une perceuse à colonne.

Ce problème pourrait être corrigé à l'aide de différents réglages de la machine à découpe jet d'eau ou de l'augmentation des diamètres des perçages lors de la conception.

Des soucis sont aussi apparus lors du pliage. Les angles de retrait n'ont pas été respectés, de plus les mors n'ont pas été placés sur les traits de pliages. Le radius dû aux paramètres matières est donc incorrectes. Un meilleur positionnement des mors ainsi qu'un calcul des angles à atteindre lors du pliage peuvent en effet résoudre ce problème. Nous aurions pu également choisir des matériaux différents, et en modifiant les paramètres de conception, les finitions pourraient être améliorées.

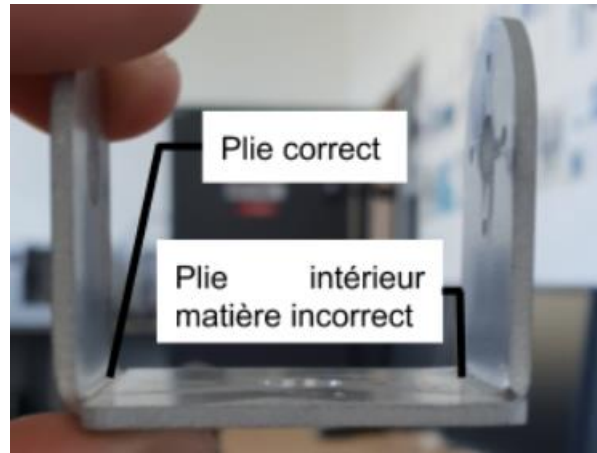


Figure 36 : Défaut N°2

Lors de la fabrication additive des pièces de la pince, nous avons aussi relevé quelques défaillances. Des fissures se sont déclarées au niveau des arrêtes du support moteur lors du retrait du support d'impression. Celles-ci sont dues au manque d'adhésion entre les couches de plastique. Un changement de température de buse pourrait régler ce problème cependant nous n'avons pas accès à ce paramètre sur le logiciel fourni par le constructeur.



Figure 37 : Défaut N°3

On observe, comme ci-contre, une ovalisation et une surépaisseur de tous les perçages lors de la fabrication par extrusion de fil chaud. Nous avons donc reperçé les trous par la suite. Des réglages de machine différents à l'aide d'un autre logiciel que celui fourni par le constructeur pourrait peut être pallier ces défauts. Des jeux supplémentaires ont pourtant été placés lors de la conception pour prévenir cela. Ces jeux ont cependant bien fonctionné sur les mâchoires de la pince au niveau des surfaces fonctionnelles utiles à la liaison glissière.



Figure 38 : Défaut N°4



Figure 39 : Assemblage du bras robot en tôle d'aluminium



Figure 40 : Vue rapproché de la pince réalisée en fabrication par extrusion de fil chaud

Les principales difficultés rencontrées lors de l'assemblage du prototype concerne les liaisons vis écrous au niveaux des haubans. Les écrous ont tendance à se déloger de leurs emplacements rendant difficile le montage. À l'avenir, nous concevrons nous même ces composants pour plus de praticité. Les tôles pliées en forme de "u" avaient aussi parfois tendances à frotter contre les têtes de vis. Après l'ajout des entretoises ou de rondelles aux niveaux des différents axes, le problème fut résolu. La pluparts des vis utilisées sont de diamètre 2 millimètres avec une tête bombée fendue.

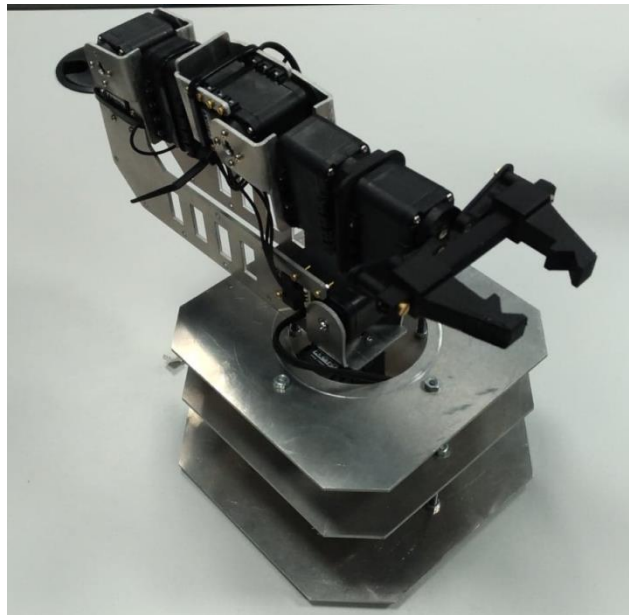


Figure 41 : Prototype final

On peut voir ci dessus l'assemblage finale du bras robot. des rails ont été disposé le long de la structure afin de maintenir les câbles et éviter qu'ils se coincent dans le mécanisme.

D. Assemblage

Nous allons maintenant passer à l'assemblage de notre robot à l'aide du logiciel 3D EXPERIENCE. Pour cela, nous avons à notre disposition les différentes pièces détaillées précédemment ainsi qu'un fichier STEP de nos moteurs dynamixel.

Nous avons dans un premier temps réalisé des sous-assemblages afin de faciliter la création du mécanisme ultérieurement.

Pour ce faire nous nous sommes rendus dans l'atelier « Assembly Design ».



Figure 42 : Atelier Assembly Design

Nous avons donc appliqué les liaisons entre nos différentes pièces afin d'obtenir nos sept sous-assemblages :

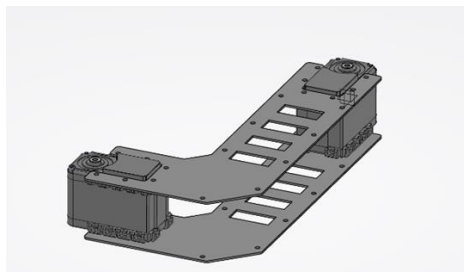


Figure 43 : Sous assemblage 1

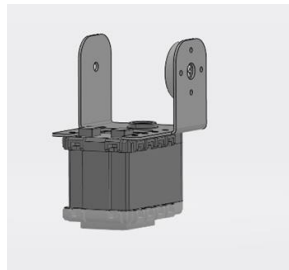


Figure 44 : Sous assemblage 2

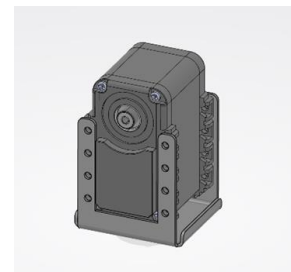


Figure 45 : Sous assemblage 3

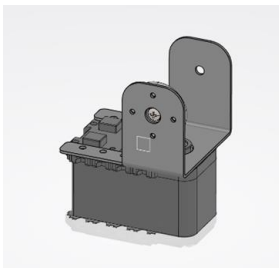


Figure 46 : Sous assemblage 4

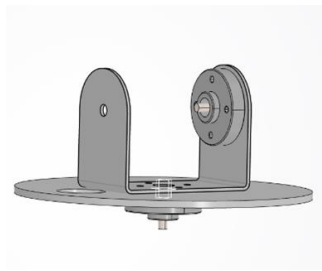


Figure 47 : Sous assemblage 5

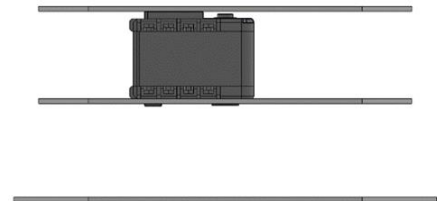


Figure 48 : Sous assemblage 6

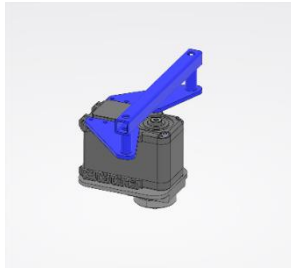


Figure 49 : Pince

Nous avons ensuite créé un nouveau « Product » dans lequel nous avons importé tous nos sous assemblage. Dans un premier temps, nous avons fixé notre base. En effet celle-ci ne devra pas bouger lorsque nous lancerons des simulations ultérieurement. Nous avons ensuite créé les liaisons pivot entre nos sept sous assemblages sans oublier de mettre un angle contrôlé dans chacune. Dans le but de pouvoir les faire bouger lors de la simulation mécanique.

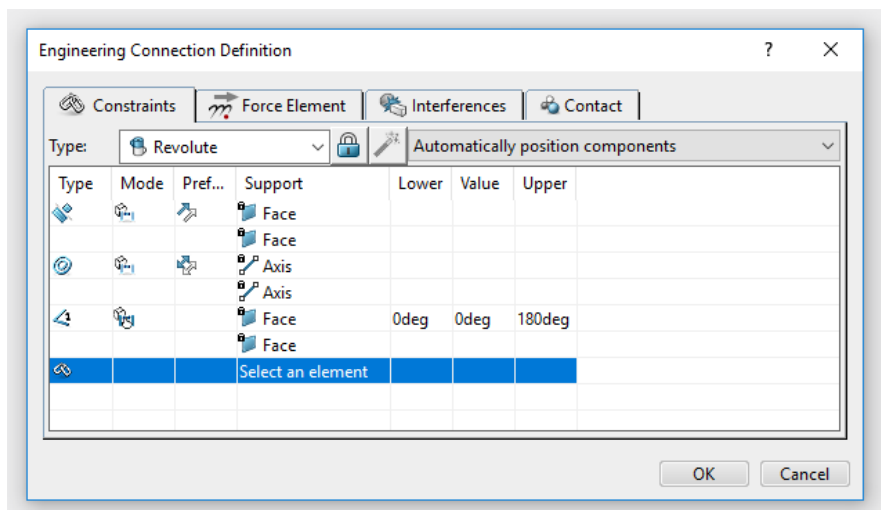


Figure 50 : Création d'une liaison pivot

Nous pouvons observer des angles maximum et minimum qui représentent l'amplitude de notre liaison. Nous avons ensuite répété la même procédure pour tous nos assemblages et avons inséré les pièces de notre pince. Nous obtenons l'assemblage suivant :



Figure 52 : Liste des liaisons de notre mécanisme

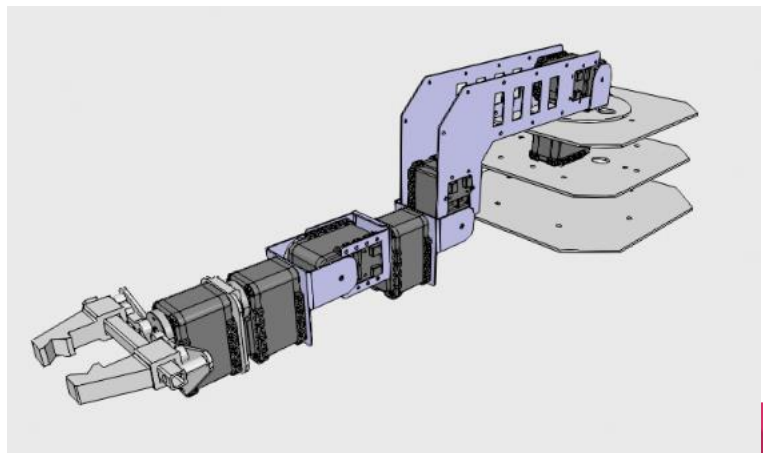


Figure 51 : Représentation de notre assemblage sous 3D Experience

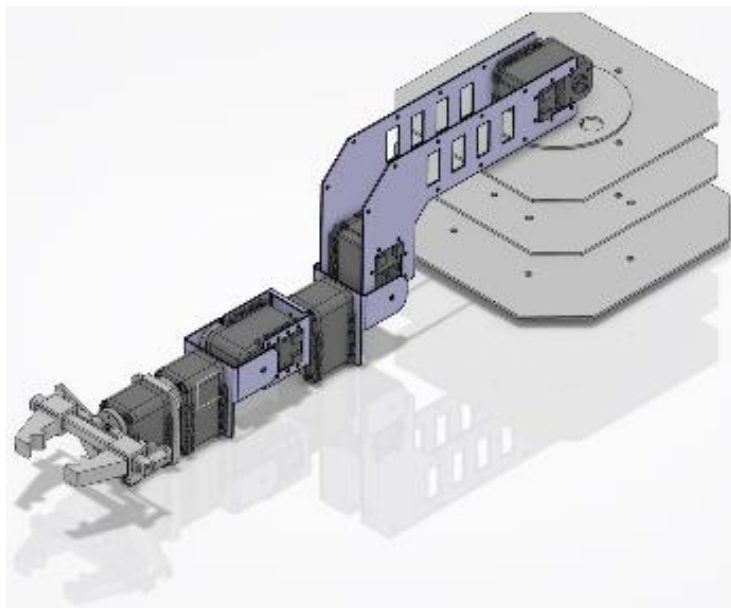


Figure 53 : Vue isométrique de l'assemblage

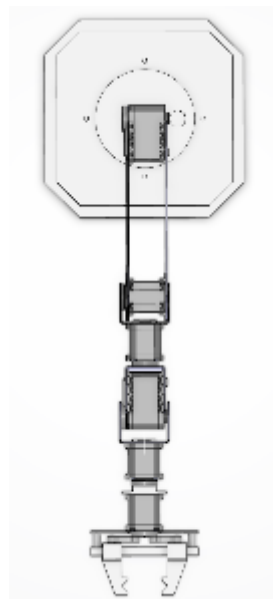


Figure 54 : Vue du dessus

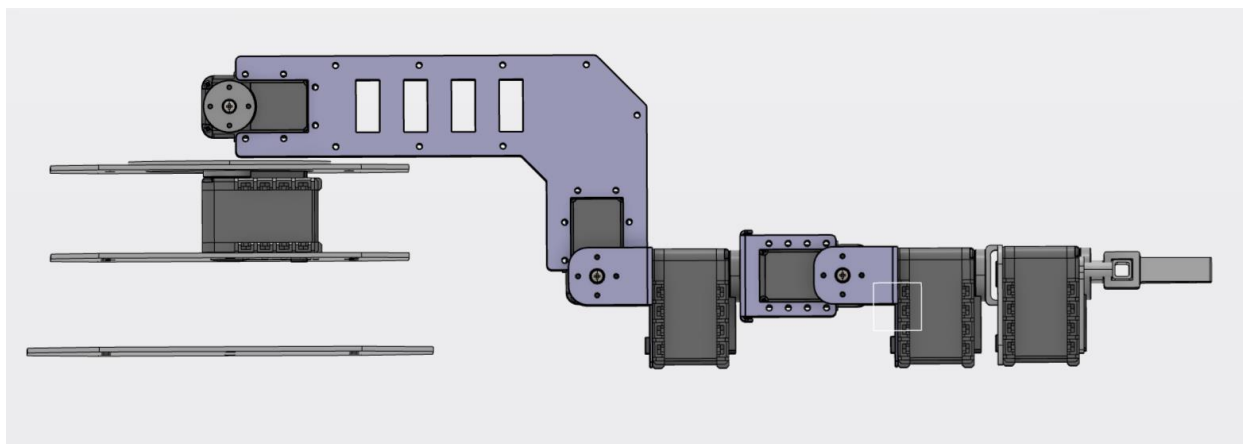


Figure 55 : Vue de côté de l'assemblage

Nous appliquons un matériau à toutes nos pièces, pour ce faire nous sommes allés chercher les matériaux désirés dans les bibliothèques de 3D EXPERIENCE et les appliquons à chacune d'elles (Aluminium et ABS). Il fallait ensuite mettre une masse à chaque pièce. Pour cela, il faut se rendre dans l'atelier « Mechanical System Design ».



Figure 56 : Atelier Mechanical System Design

Pour finir nous nous sommes rendus dans l'onglet « Weight Definition » et avons affecté ces dites masses.

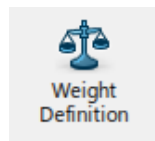


Figure 57 : Icône Weight Definition

E. Création du mécanisme

Afin de créer notre mécanisme, nous restons dans l'application « Mechanical System Design ». Comme nous avons défini des angles contrôlés dans chacune de nos liaisons, nous pouvons directement cliquer sur « Mechanism Representation » afin de créer celui-ci. Nous cliquons ensuite sur « Mechanism Manager » afin de vérifier s'il s'est créé sans soucis.

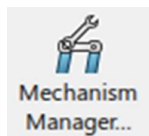


Figure 58 : Icône Mechanism Manager



Figure 59 : Icône Mechanism Representation



Figure 60 : Représentation des liaisons de notre mécanisme

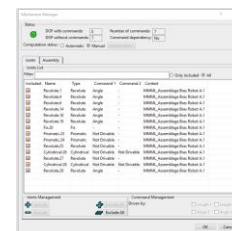


Figure 61 : Vérification de notre mécanisme

Nous appliquons la gravité à notre système et mettons des moteurs au niveau de chaque liaison afin de correspondre au mieux à la réalité.

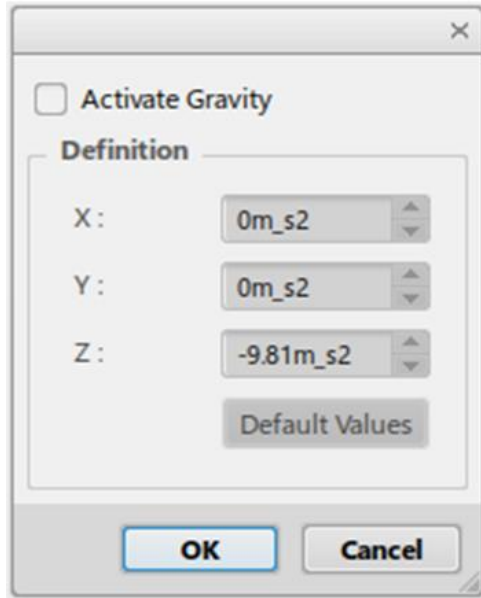


Figure 62 : Application de la gravité

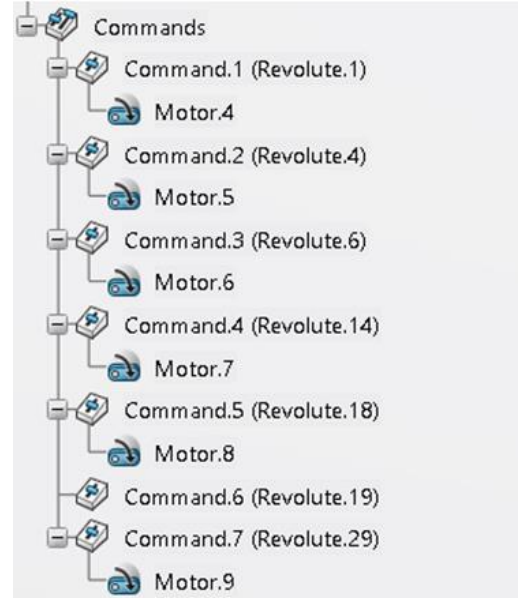
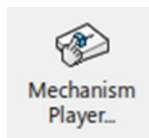


Figure 63 : Commande du mécanisme

Après avoir effectué toutes ces étapes, nous pouvons lancer la simulation en cliquant sur « Mechanism Player ». Nous obtenons une fenêtre avec des curseurs permettant de contrôler chaque liaison.



Mechanism Player...

Figure 64 : Icône de lancement du mécanisme

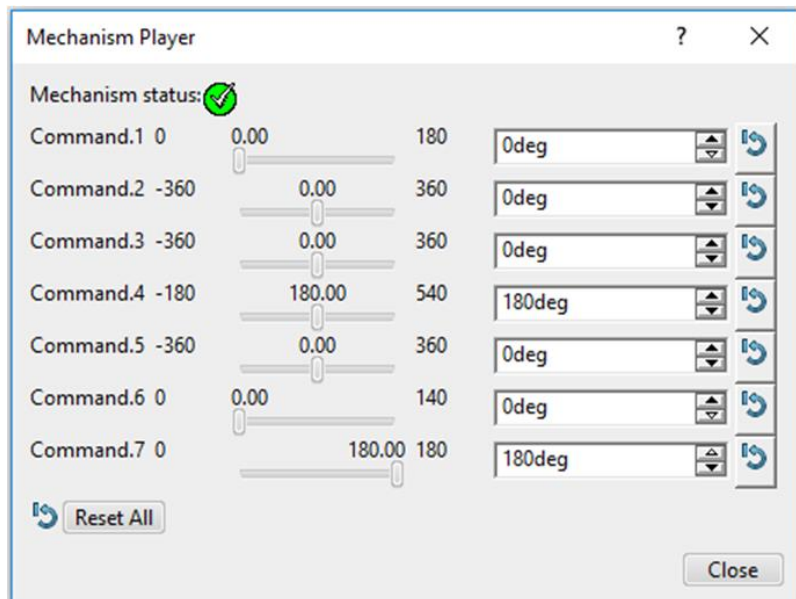


Figure 65 : Animation du mécanisme du système

En faisant varier les différents curseurs, nous faisons bouger le robot dans 3 positions différentes. Nous pouvons cependant observer que les interférences n'ont pas été prises en compte, ce qui sera un point à améliorer dans le futur.

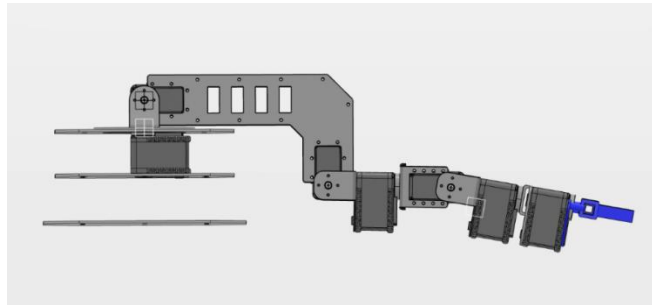


Figure 66 : Position N°1

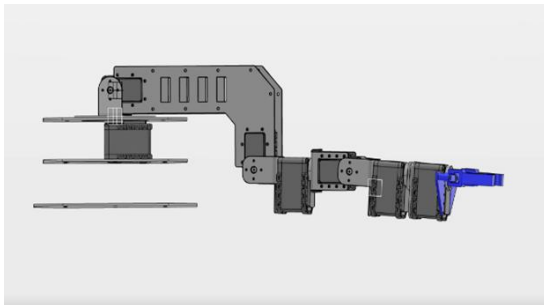


Figure 67 : Position N°2

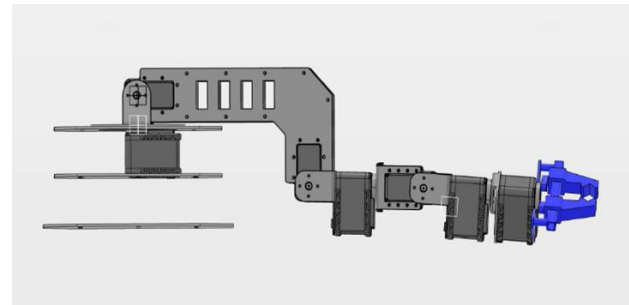


Figure 68 : Position N°3

F. Simulation dynamique

Nous allons maintenant effectuer une simulation dynamique en nous rendant dans l'atelier « Mechanical System Experience ».



Figure 69 : Atelier Mechanical System Experience

Nous créons ensuite un scénario de cinématique afin de simuler la gravité sur notre système sans l'action des moteurs. Pour ce faire il faut sélectionner « Gravity excitation ».



Figure 70 : Icône de la gravité

Puis nous cliquons sur « Dynamics scénario » afin de créer le scénario. Une fenêtre apparaît et nous choisissons « Gravity excitation » afin de simuler notre mécanisme.

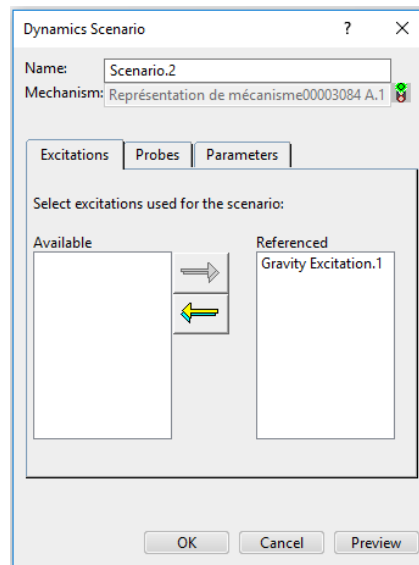


Figure 71 : Création du scénario

Nous obtenons notre simulation avec notre robot qui tombe sous son propre poids. Ceci est logique du fait que nous n'avons pas renseigné les paramètres de nos moteurs. Nous pourrions aller plus loin en rentrant des lois de déplacement pour faire bouger notre robot à des coordonnées souhaitées. Cette partie sera améliorée à l'aide de l'application « Dymola ».

G. Simulation - Dymola :

Nous allons maintenant représenter notre robot à l'aide de « Dymola ». Celui-ci sera affiché sous forme de blocks correspondant aux différentes parties du robot ainsi qu'aux différentes liaisons entre ces sous-assemblages. Pour ce faire nous lançons l'application « Behavior Modeling ».



Figure 72 : Atelier Behavior Modeling

Nous voulons dans un premier temps importer notre mécanisme créé dans la partie création du mécanisme. Cependant nous ne pouvons pas importer dans la bibliothèque dymola déjà présente. Il nous a donc fallu dans un premier temps créer notre propre bibliothèque. Puis nous avons importé notre mécanisme et obtenu la vue globale de notre robot. Celle-ci est composée de tous nos sous-assemblages, de nos différentes liaisons, de la pesanteur, ainsi que des commandes des différentes liaisons.

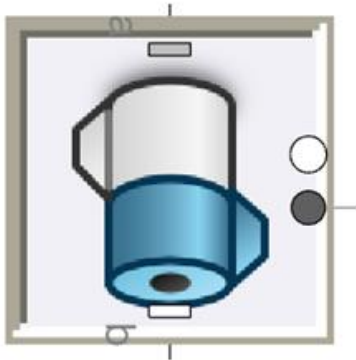


Figure 73 : Icône liaison pivot

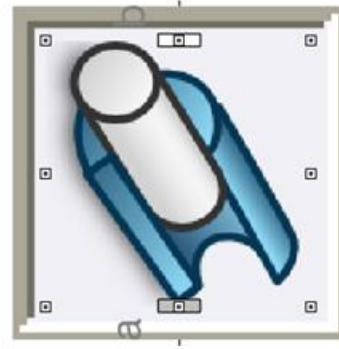


Figure 74 : Icône liaison pivot-glissant

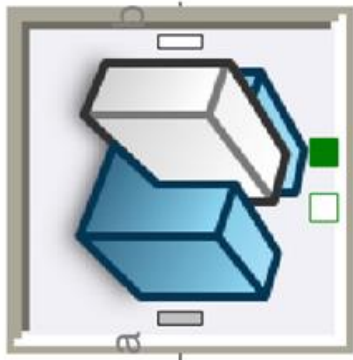


Figure 75 : Icône liaison prismatique

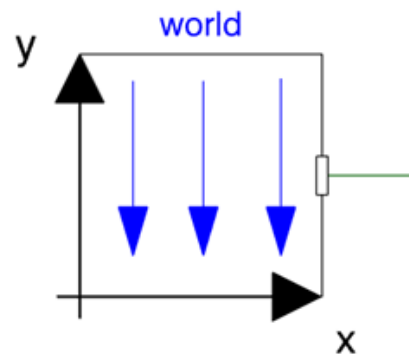


Figure 76 : Icône force de gravité

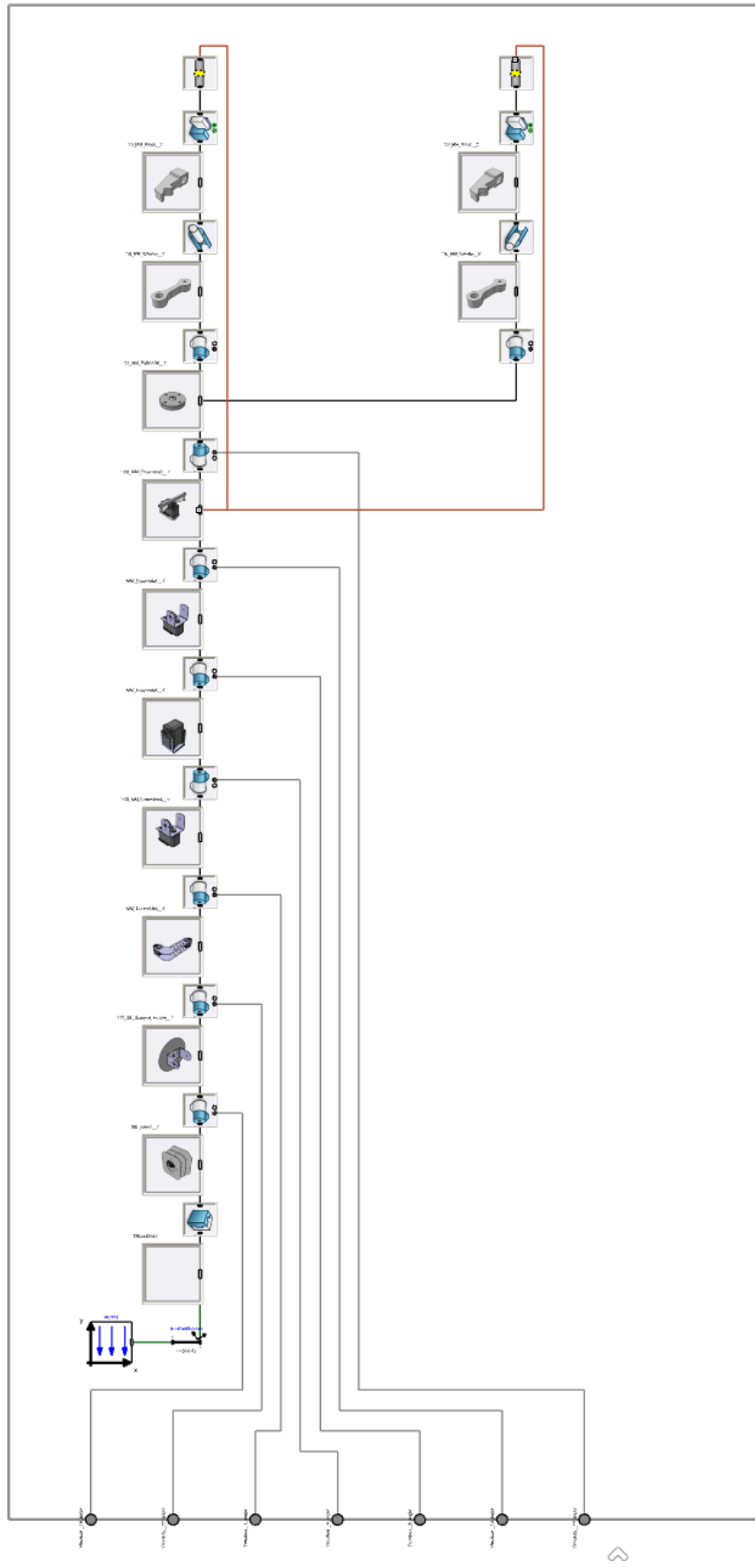


Figure 77 : Représentation du système sous Dymola

Nous avons ensuite réglé les paramètres de simulation afin d'obtenir les résultats souhaités.

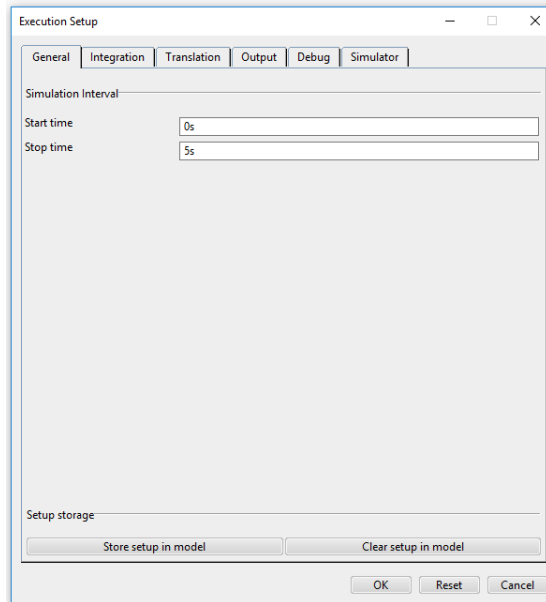


Figure 78 : Réglage de la simulation

Nous lançons ensuite la simulation et obtenons les mêmes résultats que pour la simulation dynamique. Cependant, dymola nous permet de récupérer les courbes de déplacement. Nous décidons donc de visualiser le déplacement des moteurs 2 et 3, les autres ne bougeant que très peu.

[-] CI	'Revolute_6'			
[+] CI	frame_a			
[+] CI	frame_b			
	animation	0		= true,...inder)
[+] [100/010/001]	n			
	phi_offset	0	deg	
	phi_start	0	deg	
	w_start	0	t...n	
	a_start	0	ra...s2	
	fixedSta...alues_phi	1		
	stateSelect	5		Priority... states
[+] CI	jointNonPlanar			
[-] CI	axis			
	tau	0	Nxm	Cut tor...flange
	phi	-96.0585	deg	Absolu...flange

Figure 79 : Récupération des résultats

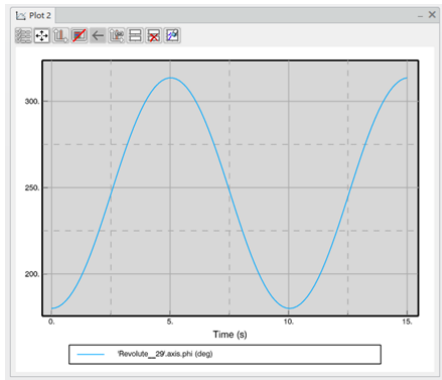


Figure 80 : Position du moteur 2 en fonction du temps

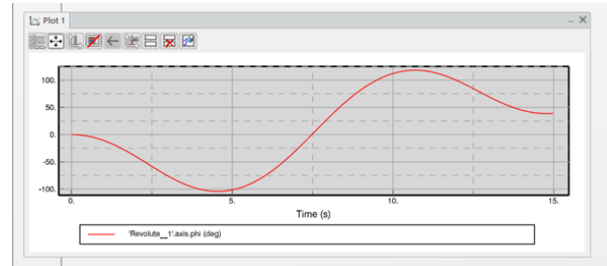


Figure 81 : Position du moteur 3 en fonction du temps

Nous pouvons observer que ces résultats correspondent à ceux d'un double pendule ce qui valide nos résultats, notre robot pouvant être assimilé à cela.

Nous décidons ensuite d'appliquer des vitesses à nos liaisons pivots pour faire bouger notre bras.

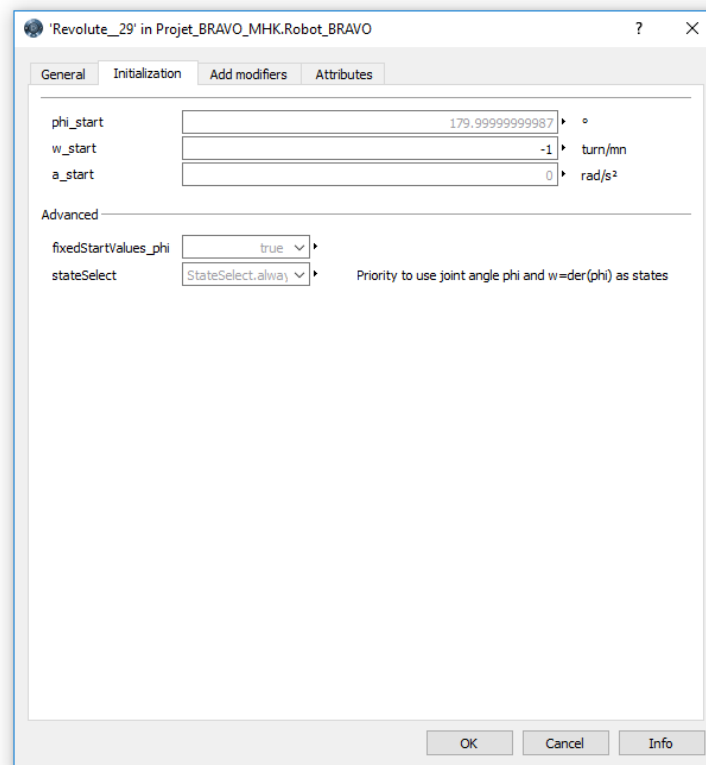


Figure 82 : Réglage des moteurs des liaisons pivot

Nous obtenons donc un déplacement dudit robot contrôlé par les vitesses renseignées aux différents moteurs.

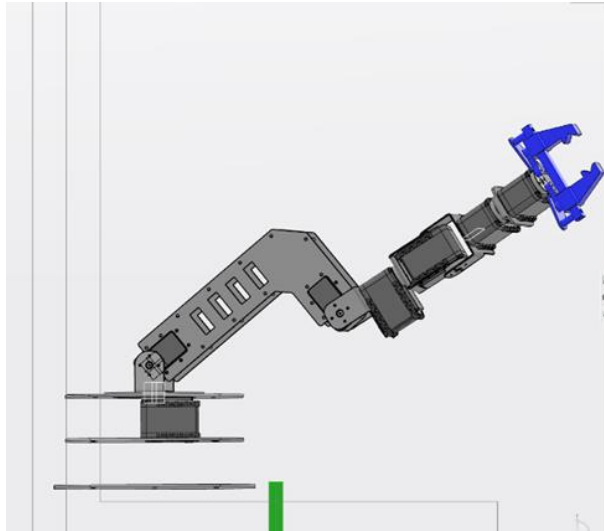


Figure 83 : Position N°1

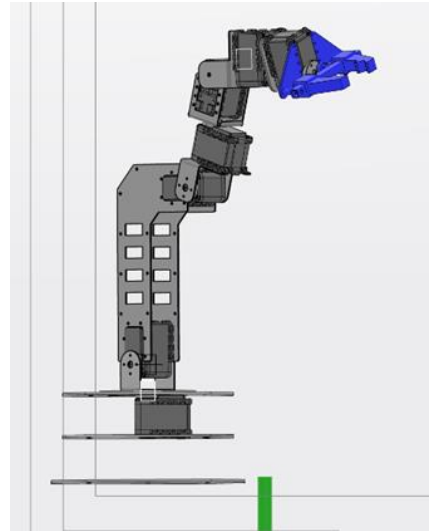


Figure 84 : Position N°2

Il serait intéressant de poursuivre cette étude en plaçant des contrôleurs et du régulateur dans notre simulation afin de contrôler plus précisément le robot. En couplant cette méthode avec celle décrite dans la partie simulation dynamique, il nous suffirait d'entrer des coordonnées pour déplacer le robot à l'endroit souhaité.

VII. Partie Électronique

A. Arduino Uno

1. Présentation

Arduino est un logiciel de codage en langage C/C++ permettant de programmer les cartes du même nom. Les principaux intérêts de ce logiciel sont sa simplicité d'utilisation, le fait qu'il soit totalement open source, sa large communauté et sa gratuité.

Ses différents points sont les principaux que nous recherchons pour correspondre aux besoins de notre projet. En effet, notre robot et tout ce qui l'entoure doit entièrement être open source. Dans l'optique de le rendre toujours plus accessible il fallait un logiciel avec une prise en main simple, avec une importante communauté d'aide et gratuit. Parmi les différents choix de cartes proposées par Arduino, encore une fois nous avons recherché la simplicité, nous avons donc opté pour la carte Arduino Uno.



Figure 85 : Logo Arduino Open-Source

2. Structure de la carte

L'Arduino se compose :

- D'une entrée USB pour y téléverser un programme
- D'une alimentation 9V pour fournir la tension nécessaire à la carte
- D'un bouton reset pour relancer un programme
- De sortie d'alimentation en 3.3 ou 5 V pour alimenter certains de nos montages
- D'une entrée et sortie logique pour envoyer ou recevoir des informations
- De 2 masses pour faire nos montages

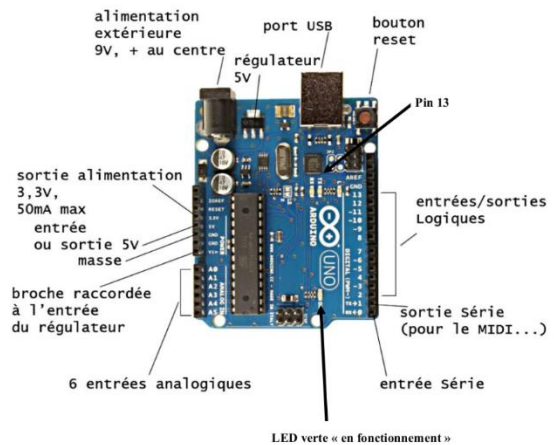


Figure 86 : Structure d'une carte Arduino Uno

3. Auto-formation : OpenClassroom

Pour l'étude des composants, l'association MHK nous mis à disposition des kits d'Arduino Uno (carte, LED, moteurs, résistances et etc.) pour nous former à la programmation en C ; une carte Arbotix-M et six moteurs dynamixel ax-12a.

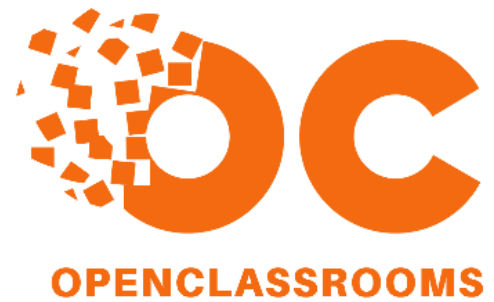


Figure 87 : Logo OpenClassrooms

Pour la formation Arduino, nous avons suivi la formation de 15h [« Programmez vos premiers montages avec Arduino »](#) du site OpenClassroom pour suivre des tutoriaux concernant la programmation et l'électronique. Cette formation nous a donc permis de nous former au langage C de manière à piloter les moteurs du robot.

Suite la formation, nous avons recherché des codes sources qui correspondent à nos servomoteurs. Nous avons par la suite adapté nos codes pour notre projet. En parallèle de ces améliorations, nous avons procédé à une phase de test pour continuer d'optimiser notre programme. Pour finir, nous avons documenté nos recherches et nos programmes afin de les publier sur le site « Wikilab » ; c'est un site qui référence tous les projets réalisés par l'association My Human Kit.

Afin de débiter à la programmation sous Arduino , nous avons eu besoin du matériel suivant :



Figure 88 : Arduino Uno



Figure 89 : Câble de connexion USB

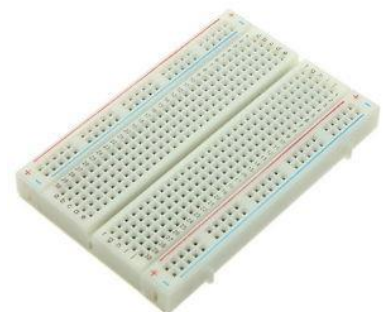


Figure 90 : Platine

4. Programmation d'un servomoteur

Notre première programme nous a permis de contrôler un servomoteur « TowerPro SG90 » suivant plusieurs instructions. Nous avons développé trois manières à commander le moteur :

- Programme N°1 : servomoteur + programme simple
 - Programme N°2 : servomoteur + potentiomètre
 - Programme N°3 : servomoteur + joystick
- Programme N°1 : servomoteur + programme



Figure 91 : Servomoteur

À l'aide du logiciel Arduino, nous avons pu programmer le servomoteur en lui faisant suivre deux instructions :

- Tout d'abord, le servomoteur tourne de 0° à 180°.
- Dans le sens contraire, le servomoteur tourne de 180° à 0°.

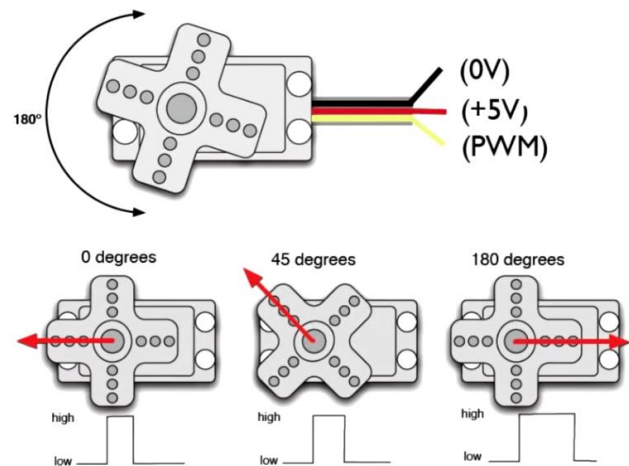


Figure 92 : Fonctionnement d'un servomoteur

ATTENTION : pour ce type de servo, il est limité en position de 0° à 180°. Si on applique un angle supérieur à 180°, le servomoteur va rentrer en buté et risque de chauffer ou se casser.

Câblage du montage :

- Reliez le câble de connexion USB de type A/B de la borne de la carte Arduino à l'ordinateur.
- Reliez un câble noir à la masse du servomoteur (marron) au pin GND (partie power) de la carte Arduino.
- Reliez un câble rouge à l'alimentation du servomoteur (rouge) au pin 5V (partie power) de la carte Arduino.

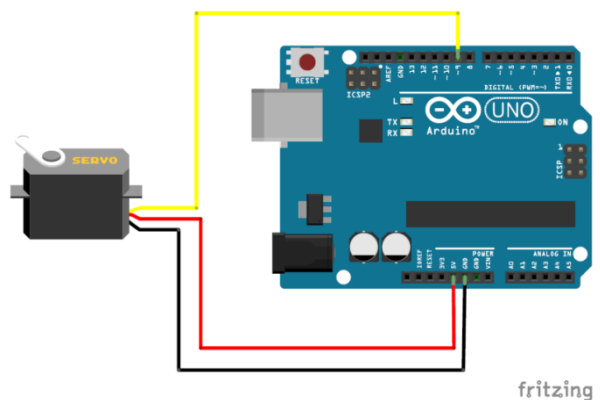


Figure 93 : Montage Arduino Uno + servomoteur

- Reliez un câble jaune au pin du servomoteur (jaune/orange) au pin 9 (partie digital) de la carte Arduino.

Programme N°1 : servomoteur

```
#include <Servo.h>

Servo Servo1; // Déclare l'utilisation d'un servomoteur sous le nom de "Servo1"
int positionDuServo1 = 0; // Variable contenant la position souhaitée pour le servo 1

/* Configuration des composants et de la carte Arduino*/
void setup()
{
  Servo1.attach(9); // Connexion du servo 1 sur le pin DIGITAL 9
  Servo1.write(positionDuServo1); // Position initiale du servo 1
}

/* Programme principal*/
void loop()
{
  // Rotation dans un sens
  for(positionDuServo1 = 0; positionDuServo1<=180;positionDuServo1++)
  {
    Servo1.write(positionDuServo1); // On affecte la position au servo 1
    delay(15);
  }

  // Rotation dans le sens opposé
  for(positionDuServo1 = 180; positionDuServo1>=0;positionDuServo1--)
  {
    Servo1.write(positionDuServo1);
    delay(15);
  }
}
```

5. Programme N°2 : servomoteur + potentiomètre

Pour ce programme, nous avons réalisé une commande permettant à l'utilisateur de contrôler le servomoteur à l'aide d'un potentiomètre.

Câblage du montage :

- Réalisez le même montage servomoteur+arduino que le précédent.
- Reliez deux câbles noir/rouge à l'alimentation du potentiomètre aux pins GND/5V (partie power) de la carte Arduino.
- Reliez un câble bleu au pin du potentiomètre au pin A0 (partie analog) de la carte arduino.

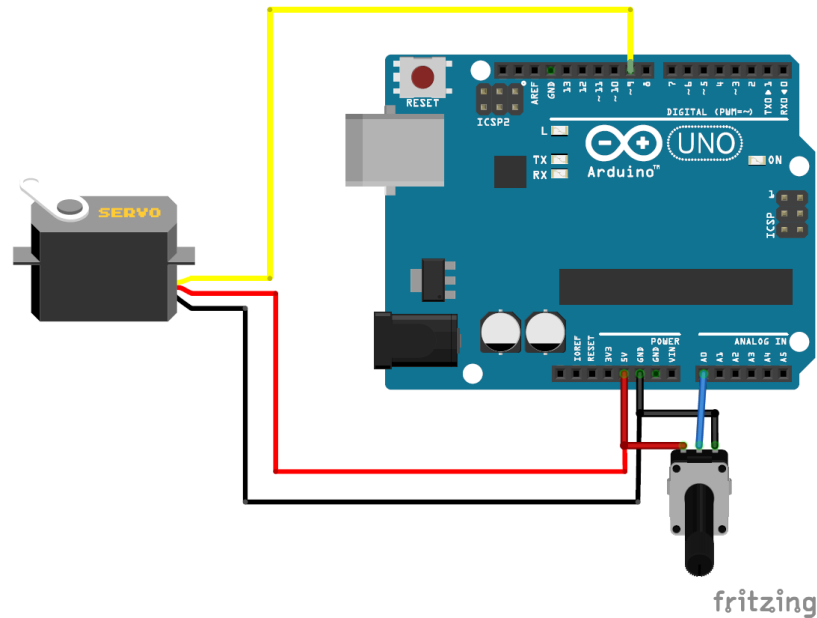


Figure 94 : Montage servomoteur + potentiomètre

Programme N°2 : servomoteur+potentiomètre

```
#include <Servo.h>
```

```
Servo Servo1; // Déclare l'utilisation d'un servomoteur sous le nom de "Servo1"
```

```
char potentiometre=0; // Variable pour le potentiomètre sur le pin A0
```

```
int positionDuServo1 = 0; // Variable contenant la position souhaitée pour le servo 1
```

```
/* Configuration des composants et de la carte arduino*/
```

```
void setup()
```

```
{
```

```
  Servo1.attach(9); // Connexion du servo 1 sur le pin DIGITAL 9
```

```
  Servo1.write(positionDuServo1); // Position initiale du servo 1
```

```
}
```

```
/* Programme principal*/
```

```
void loop()
```

```
{
```

```
  positionDuServo1 = analogRead(potentiometre); // Lecture de la position du potentiometre (A0)
```

```
  positionDuServo1=map(positionDuServo1,0,1023,0,180); // Effectue un produit en croix pour réaliser une conversion entre le servomoteur et le potentiomètre
```

```
  Servo1.write(positionDuServo1); // Mise en position du servomoteur à une position donnée
```

```
  delay(15);
```

```
}
```

6. Programme N°3 : servomoteurs + joystick

Pour ce programme, nous avons réalisé une commande permettant à l'utilisateur de contrôler deux servomoteurs à l'aide d'un joystick.

Câblage du montage :

- Effectuez la même méthode pour le câblage d'alimentation des servomoteurs. Connectez par la suite les servomoteurs sur les pins 9 et 10.
- Reliez respectivement les pins A0 et A1 sur les pins X et Y du joystick.

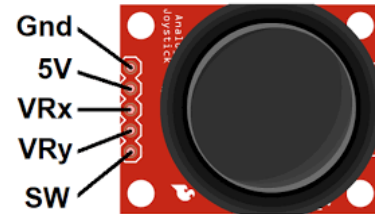


Figure 95 : Structure d'un module joystick

```

Programme N°3 : 2 servomoteurs+josytick
#include <Servo.h>

Servo Servo1; // Déclare l'utilisation d'un servomoteur sous le nom de "Servo1"
Servo Servo2; // Déclare l'utilisation d'un servomoteur sous le nom de "Servo2"

int joyX=0; // Variable pour l'axe X du joystick sur le pin A0
int joyY=1; // Variable pour l'axe Y du joystick sur le pin A1
int positionDuServo;

/* Configuration des composants et de la carte arduino*/
void setup()
{
  Servo1.attach(9); // Connexion du servo 1 sur le pin DIGITAL 9
  Servo2.attach(10); // Connexion du servo 2 sur la pin DIGITAL 10
}

/* Programme principal*/
void loop()
{
  /* Contrôle du servo 1 sur l'axe X du joystick*/
  positionDuServo=analogRead(joyX);
  positionDuServo=map(positionDuServo,0,1023,0,180);
  Servo1.write(positionDuServo);

  /* Contrôle du servo 2 sur l'axe Y du joystick*/
  positionDuServo=analogRead(joyY);
  positionDuServo=map(positionDuServo,0,1023,0,180);
  Servo2.write(positionDuServo);
  delay(15);
}
  
```

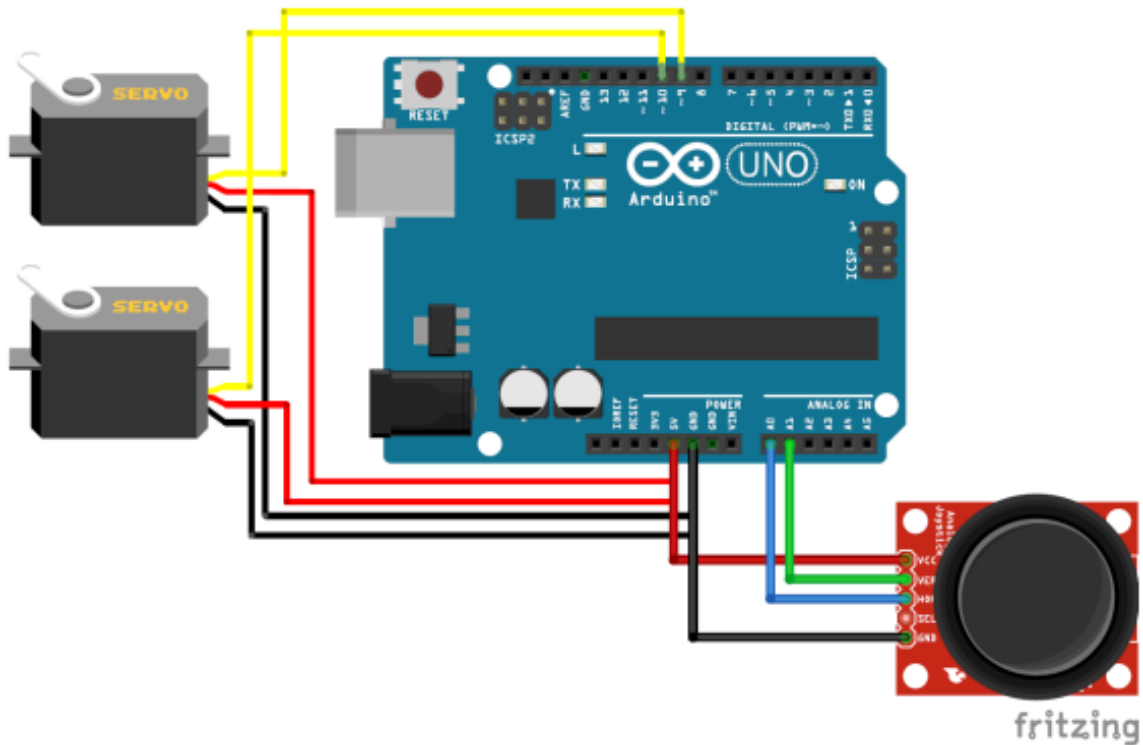



Figure 96 : Montage deux servomoteurs + joystick

B. Arbotix-M

1. Présentation

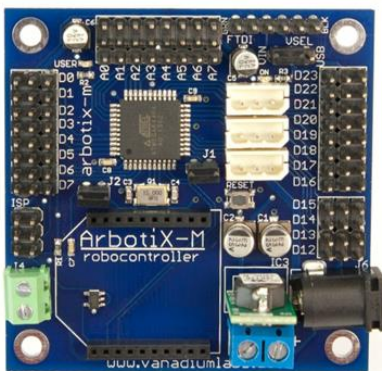


Figure 97 : Carte Arbotix-M

Pour la programmation des servomoteurs Dynamixel, nous avons utilisé la carte Arbotix-M. Elle a été spécialement conçue pour ce type de servomoteur et compatible avec les cartes Arduino. L'Arbotix-M dispose des « Headers 3-pin » de type servo (GND, VCC et Signal) sur huit entrées analogiques et huit IO numériques. Il est nécessaire d'avoir un câble FTDI pour communiquer entre l'ordinateur et la carte, d'avoir un chargeur de 12V et d'installer le logiciel Arduino 1.0.6.

La documentation détaillée de la carte contrôleur est disponible sur le site du constructeur « [Trossen Robotics](#) ».

2. Structure de la carte

La carte Arbotix-M propose plusieurs pins permettant de recevoir plusieurs composants électroniques :

- Des pins d'entrées/sorties numériques (D0 à D23).
- Des pins d'entrées analogiques (A0 à A7).
- Des ports DYNAMIXEL pour commander des servomoteurs (AX-12a).

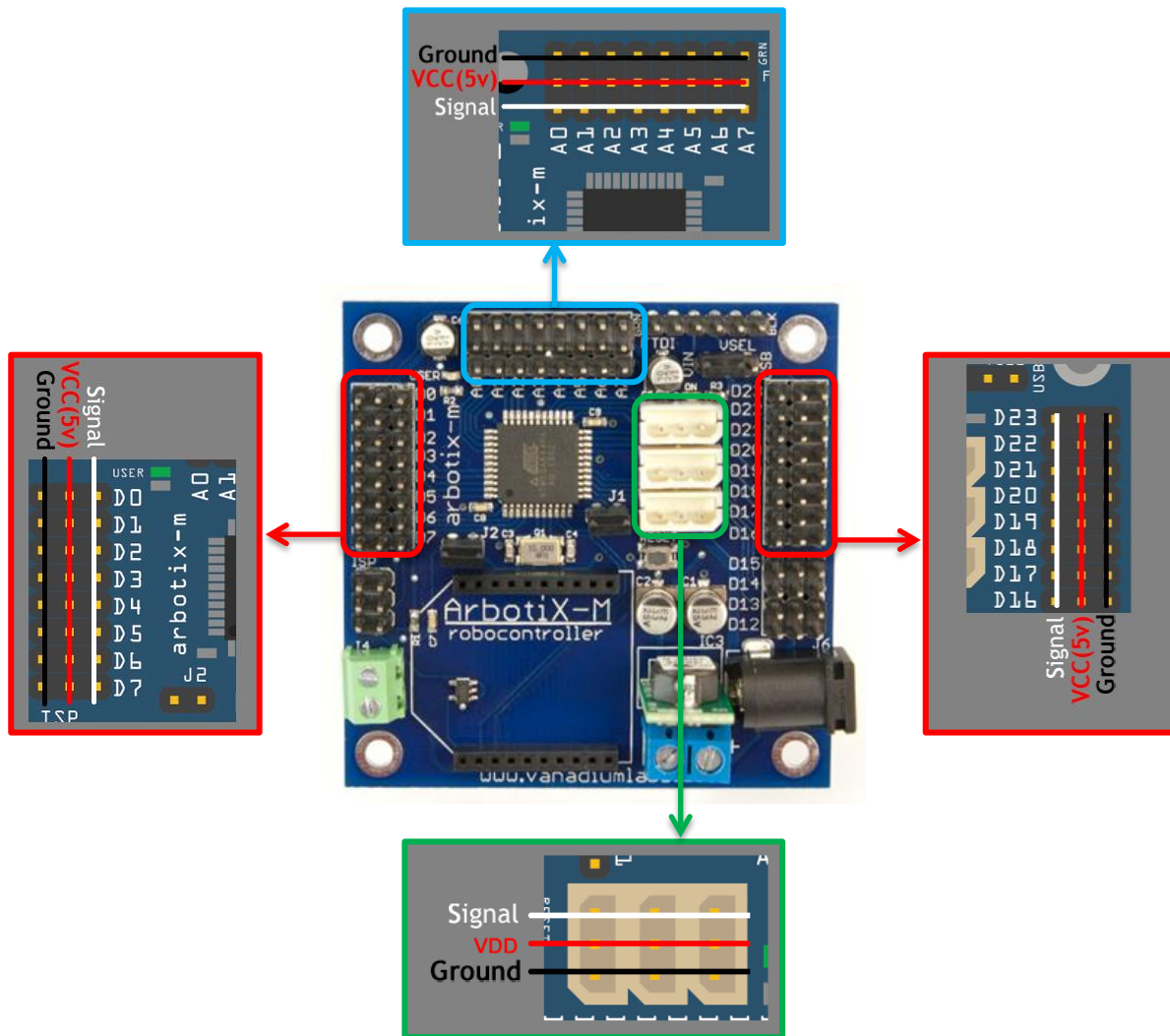


Figure 98 : Structure de la carte

3. Programmation Arbotix-M

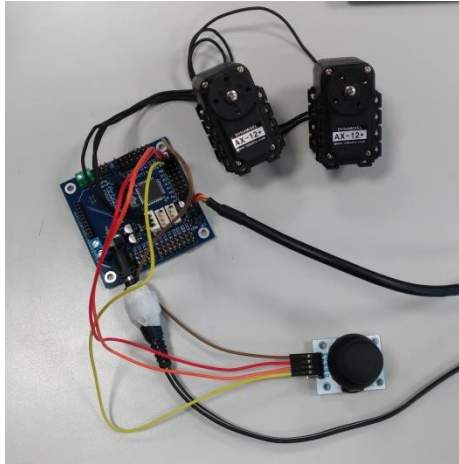


Figure 99 : Montage servomoteur + joystick

Avec la même méthode utilisée pour commander les servomoteurs « TowerPro SG90 » avec un joystick, nous avons adapté notre code pour contrôler deux servomoteurs Dynamixel AX-12A avec la carte Arbotix-M.

Il est nécessaire d'ajouter les bibliothèques « #include <ax12.h>P » et « #include <BioidController.h> » afin d'utiliser les fonctionnalités nécessaires pour contrôler les servomoteurs.

Programme N°1 : Arbotix Dynamixel ax-12

```

/* Bibliothèque Dynamixel*/
#include <ax12.h>P
#include <BioidController.h>

int joyX=0;
int joyY=1;
int joyVal;

void setup()
{
  SetPosition(5,512); // Mise à position initiale des servomoteurs
}

void loop()
{
  joyVal=analogRead(joyX);
  joyVal=map(joyVal,0,1023,30,1000);
  SetPosition(5,joyVal);

  joyVal=analogRead(joyY);
  joyVal=map(joyVal,0,1023,30,1000);
  SetPosition(8,joyVal);

  delay(25);
}

```

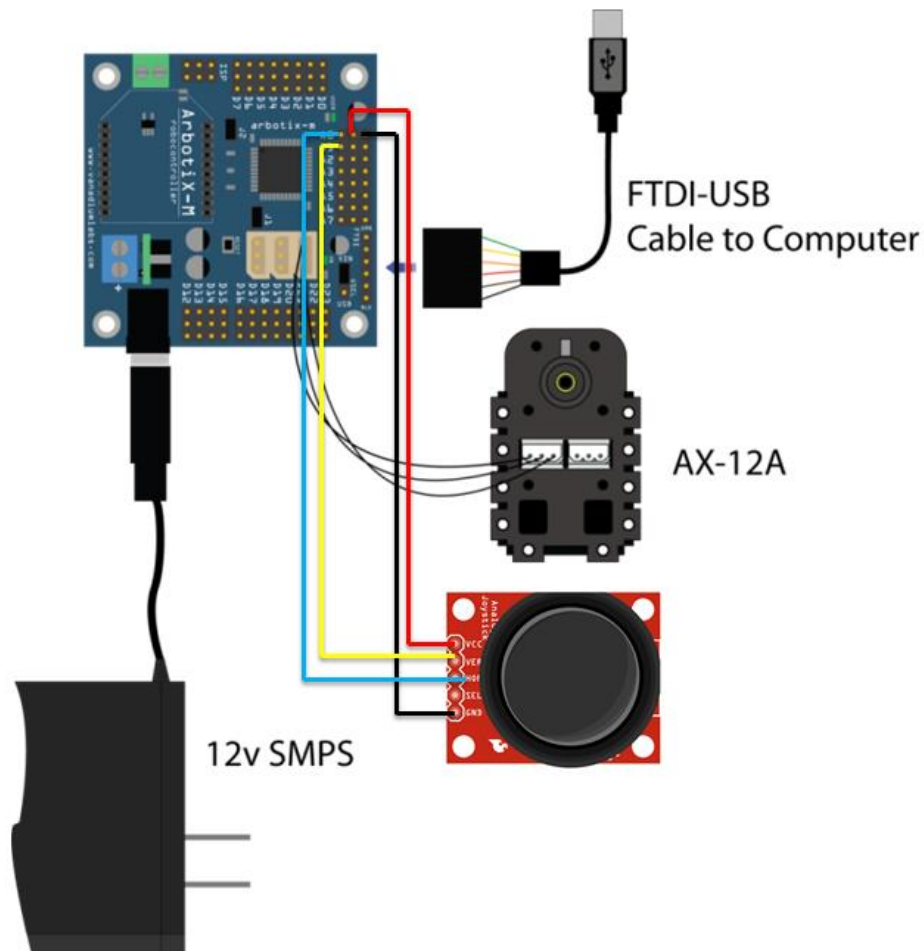


Figure 100 : Câblage des différents composants

4. Problème rencontré

Au cours de notre projet, nous avons constaté avec l'association My Human Kit que la carte Arbotix-M présentait des problèmes de connexion entre la carte et les servomoteurs Dynamixel. En effet, lors d'une connexion en série de servomoteurs, la carte Arbotix-M n'envoyait plus d'informations et par conséquent ne les commandait plus. Nous n'avons pas réussi à reconfigurer la carte et faire fonctionner le bras robot avec cette carte.

Nous avons donc décidé de changer avec une autre carte de la même marque pour continuer nos développements : la carte OpenCM 9.

C. OpenCM 9



La carte OpenCM 9 fonctionne de la même manière que le contrôleur Arbotix-M sur la commande des servomoteurs Dynamixel. En raison du manque de connaissances de cette carte, Hugues AUBIN, le responsable du projet au sein de l'association de My Human Kit, s'est chargé du développement du code afin de commander le bras robot. De plus, avec le porteur de projet Nicolas POUSSET, ils ont pu réaliser un programme permettant de communiquer entre la carte OpenCM 9 et un smartphone en bluetooth.

Figure 101 : Carte OpenCM 9

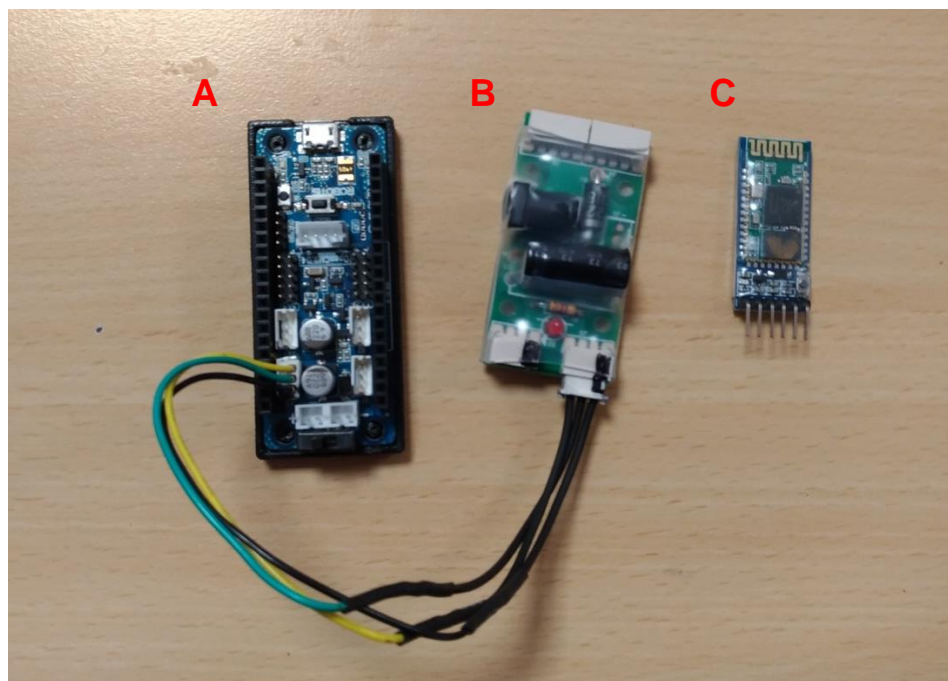


Figure 102 : (A) Carte OpenCM 9, (B) alimentation et (C) bluetooth

VIII. Coûts du projet

	Matériel	Coût unitaire (pc/€)	Quantité	Coût (€)
Composants électroniques	ArbotiX-M Robocontroller	40€	1	40€
	FTDI Cable 5V	15€	1	15€
	Câble DYNAMIXEL (lot de 10)	13€	1	13€
	Servomoteur ax-12a	45€	6	270€
Prototypage	Structure aluminium	200€	1	200€
	Découpe jet d'eau			
Coût du prototype bras robot				538€

IX. Conclusion

Ce projet nous a permis d'allier compétence professionnelle et scolaire. Passer de la théorie à la pratique fut une tâche difficile à surmonter pour nous. De plus travailler avec un chargé de projet n'étant pas présent nécessite la mise en place d'un cahier des charges très précis. Nous avons donc pu tout au long de notre projet améliorer nos compétences dans le domaine de la robotique, de la gestion de projet, ainsi qu'en électronique qui était un domaine complètement nouveau pour nous. Avoir travaillé avec le logiciel 3DExperience nous a également permis de pouvoir utiliser les logiciels appliqués dans l'usine 4.0. Nous pourrions également parler de la collaboration avec MY Human kit et du Pôle Mécanique et Technologique de l'université de Rennes 1 qui nous a apportés beaucoup tout au long de notre projet. En effet, leurs méthodes de travail n'étant pas les mêmes, nous avons donc dû nous adapter à chacun et à leurs spécificités propres. Dans ce projet nous avons donc abordé bon nombre d'aspect de la vie en entreprise que ce soit la gestion du travail, la répartition des tâches ou la collaboration d'autres organismes. Ce projet nous a permis de mettre en lumière les difficultés à travailler en groupe. Le travail que nous avons effectué sur ces compétences nous permettra de nous démarquer dans nos emplois à venir.

Face à la quantité de travail nécessaire, nous n'avons malheureusement pas réussi à produire le robot final que nous avons imaginé. Cependant, nous nourrissons l'espoir que ce projet soit repris par les promotions futures afin de le mener à bien à l'aide des bases que nous avons

fournies. Cela pourrait à terme amener un partenariat entre l'association My Human Kit et le Pôle Mécanique et Technologique de l'université Rennes 1.

X. Table des figures

Figure 1 : Logo My Human Kit	4
Figure 2 : Bras robotisé "Jaco"	5
Figure 3 : Management sur 3DExperience	7
Figure 4 : Définition des tâches sur 3DExperience	7
Figure 5 : Gantt sur 3DExperience.....	7
Figure 6 : Évolution des différentes tâches	8
Figure 7 : Servomoteur Dynamixel AX-12A	10
Figure 8 : Fonctionnement du servomoteur.....	10
Figure 9 : Fonctionnement en série	11
Figure 10 : Table paramétrée des couples moteurs	13
Figure 11 : Dimension du bras robot.....	13
Figure 12 : Plaque 1	17
Figure 13 : Plaque 2	17
Figure 14 : Plaque 3.....	17
Figure 15 : Disque	17
Figure 16 : U1	17
Figure 17 : Plaque bras	17
Figure 18 : U2.....	17
Figure 19 : U3.....	17
Figure 20 : U4.....	17
Figure 21 : Support moteur	18

Figure 22 : Support moteur pince.....	18
Figure 23 : Entretoise.....	18
Figure 24 : Glissière.....	18
Figure 25 : Bielle	18
Figure 26 : Pince.....	18
Figure 27 : Bras robot sous 3DExperience.....	18
Figure 28 : Plaque bras	19
Figure 29 : U2.....	19
Figure 30 : Support moteur	19
Figure 31 : Pince du bras robot	19
Figure 32 : Imprimante 3D "Zortrax".....	20
Figure 33 : Découpeuse jet d'eau	20
Figure 34 : Plieuse mécanique	20
Figure 35 : Défaut N°1.....	20
Figure 36 : Défaut N°2.....	21
Figure 37 : Défaut N°3.....	21
Figure 38 : Défaut N°4.....	21
Figure 39 : Assemblage du bras robot en tôle d'aluminium.....	22
Figure 40 : Vue rapproché de la pince réalisée en fabrication par extrusion de fil chaud.....	22
Figure 41 : Prototype final	22
Figure 42 : Atelier Assembly Design	23
Figure 43 : Sous assemblage 1	23

Figure 44 : Sous assemblage 2	23
Figure 45 : Sous assemblage 3	23
Figure 46 : Sous assemblage 4	23
Figure 47 : Sous assemblage 5	23
Figure 48 : Sous assemblage 6	23
Figure 49 : Pince	24
Figure 50 : Création d'une liaison pivot.....	24
Figure 51 : Représentation de notre assemblage sous 3D Experience.....	24
Figure 52 : Liste des liaisons de notre mécanisme	24
Figure 53 : Vue isométrique de l'assemblage	25
Figure 54 : Vue du dessus	25
Figure 55 : Vue de côté de l'assemblage.....	25
Figure 56 : Atelier Mechanical System Design	26
Figure 57 : Icône Weight Definition	26
Figure 58 : Icône Mechanism Manager.....	26
Figure 59 : Icône Mechanism Representation	26
Figure 60 : Représentation des liaisons de notre mécanisme	26
Figure 61 : Vérification de notre mécanisme	26
Figure 62 : Application de la gravité.....	27
Figure 63 : Commande du mécanisme	27
Figure 64 : Icône de lancement du mécanisme	27
Figure 65 : Animation du mécanisme du système.....	27

Figure 66 : Position N°1	28
Figure 67 : Position N°2	28
Figure 68 : Position N°3	28
Figure 69 : Atelier Mechanical System Experience	28
Figure 70 : Icône de la gravité	29
Figure 71 : Création du scénario	29
Figure 72 : Atelier Behavior Modeling	30
Figure 73 : Icône liaison pivot	30
Figure 74 : Icône liaison pivot	30
Figure 75 : Icône liaison prismatique	30
Figure 76 : Icône force de gravité	30
Figure 77 : Représentation du système sous Dymola	31
Figure 78 : Réglage de la simulation	32
Figure 79 : Récupération des résultats	32
Figure 80 : Position du moteur 2 en fonction du temps	33
Figure 81 : Position du moteur 3 en fonction du temps	33
Figure 82 : Réglage des moteurs des liaisons pivot	33
Figure 83 : Position N°1	34
Figure 84 : Position N°2	34
Figure 85 : Logo Arduino Open-Source	35
Figure 86 : Structure d'une carte Arduino Uno	Erreur ! Signet non défini.
Figure 87 : Logo OpenClassrooms	36

Figure 88 : Arduino Uno	36
Figure 89 : Câble de connexion USB	36
Figure 90 : Platine	36
Figure 91 : Servomoteur	37
Figure 92 : Fonctionnement d'un servomoteur	37
Figure 93 : Montage Arduino Uno + servomoteur	37
Figure 94 : Montage servomoteur + potentiomètre	39
Figure 95 : Structure d'un module joystick	40
Figure 96 : Montage deux servomoteurs + joystick.....	41
Figure 97 : Carte Arbotix-M	41
Figure 98 : Structure de la carte	42
Figure 99 : Montage servomoteur + joystick	43
Figure 100 : Câblage des différents composants	44
Figure 101 : Carte OpenCM 9.....	45
Figure 102 : (A) Carte OpenCM 9, (B) alimentation et (C) bluetooth	45