

UROP : Assembly Line

Introduction

Des élèves du centre d'innovation (DVIC) du Pôle Universitaire Léonard de Vinci, à la Défense, ont pour objectif de développer un essaim de drones volants. Le développement de leur projet avançant à grands pas, ils ont désormais la nécessité de produire massivement leurs appareils.

L'option industrielle classique n'était pas viable. La quantité de drones à produire étant ridicule à l'échelle d'une usine spécialisée, et même si la quantité n'avait pas été un problème, cela aurait coûté très cher. Ils misent alors sur le développement d'une petite chaîne de production, interne au centre, qui réduirait les coûts en plus d'être adaptée à leurs problématiques.

C'est pour répondre à ce besoin que nous avons pour objectif de développer une chaîne d'assemblage capable de produire massivement et sans efforts des drones roulants simples. Une fois la technologie développée pour ces drones simples, la chaîne d'assemblage servira de base à une seconde version, modifiée pour produire les drones volants de l'essaim.

Partie 1 : Le Drone

Conception

Avant de se lancer dans la conception de la chaîne d'assemblage, nous devons savoir quoi assembler. Le cahier des charges du drone était relativement court. Il nous fallait un drone roulant autonome dont la surface au sol ne dépassait pas quinze par quinze centimètres.

Les contraintes supplémentaires, nous nous les sommes imposés en pensant à ce que nous imposerait notre chaîne d'assemblage.

La complexité d'une chaîne d'assemblage varie grandement. En effet, encastrier une pièce dans une autre est simple. Mais si notre système est plus complexe que deux pièces de Lego®, la complexité de la chaîne d'assemblage croît de façon exponentielle. Par exemple, l'ajout d'une simple vis peut impliquer la création d'un bras capable de s'aligner sur le pas de vis, poser la vis, puis changer d'outil pour la visser.

Bien que ce soit quelque chose qui est développable, notre mission est de produire des drones simples dans leur conception pour limiter au maximum la complexité de notre chaîne d'assemblage. Nos travaux serviront de base pour développer une chaîne plus complexe, adaptée aux drones volants.

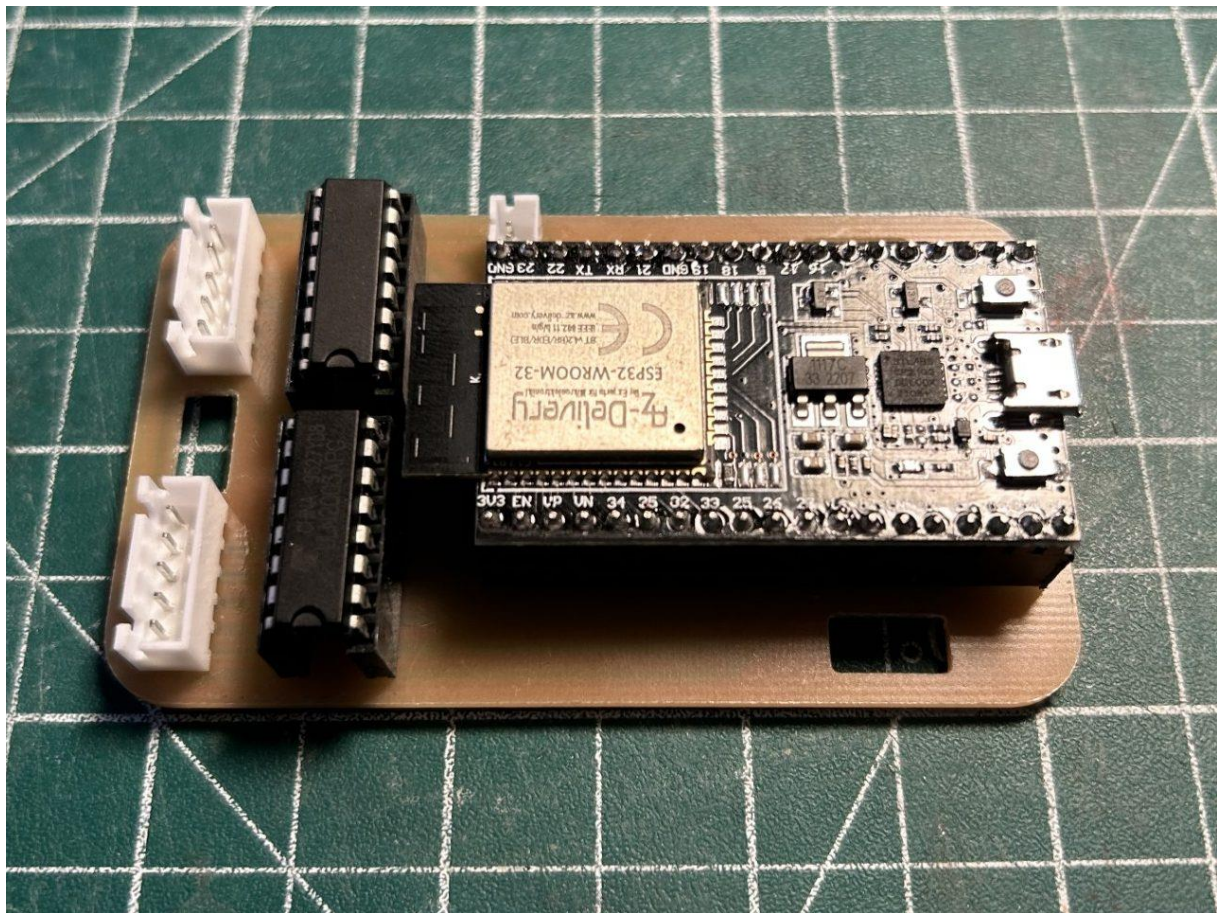
Ainsi, nous avons fait le choix de limiter à deux le nombre de roues, car moins de roues c'est moins de moteurs, d'électronique et d'encastres sur la chaîne d'assemblage. La stabilité de notre drone sera assurée différemment. De plus, nous avons décidé de nous débarrasser des vis et d'assurer l'intégralité des liaisons par des systèmes de fixation rapide (clips).

Ne visant pas la performance sur ces petits drones, nous n'avons aucunement besoin qu'ils soient rapide. Ainsi, pour les contrôler plus facilement, nous faisons le choix de moteurs pas-à-pas pour actionner les roues, plutôt que les moteurs à courant continu standard. Les moteurs pas-à-pas permettent un contrôle très précis en échange d'une vitesse plus lente.

Electronique et PCB

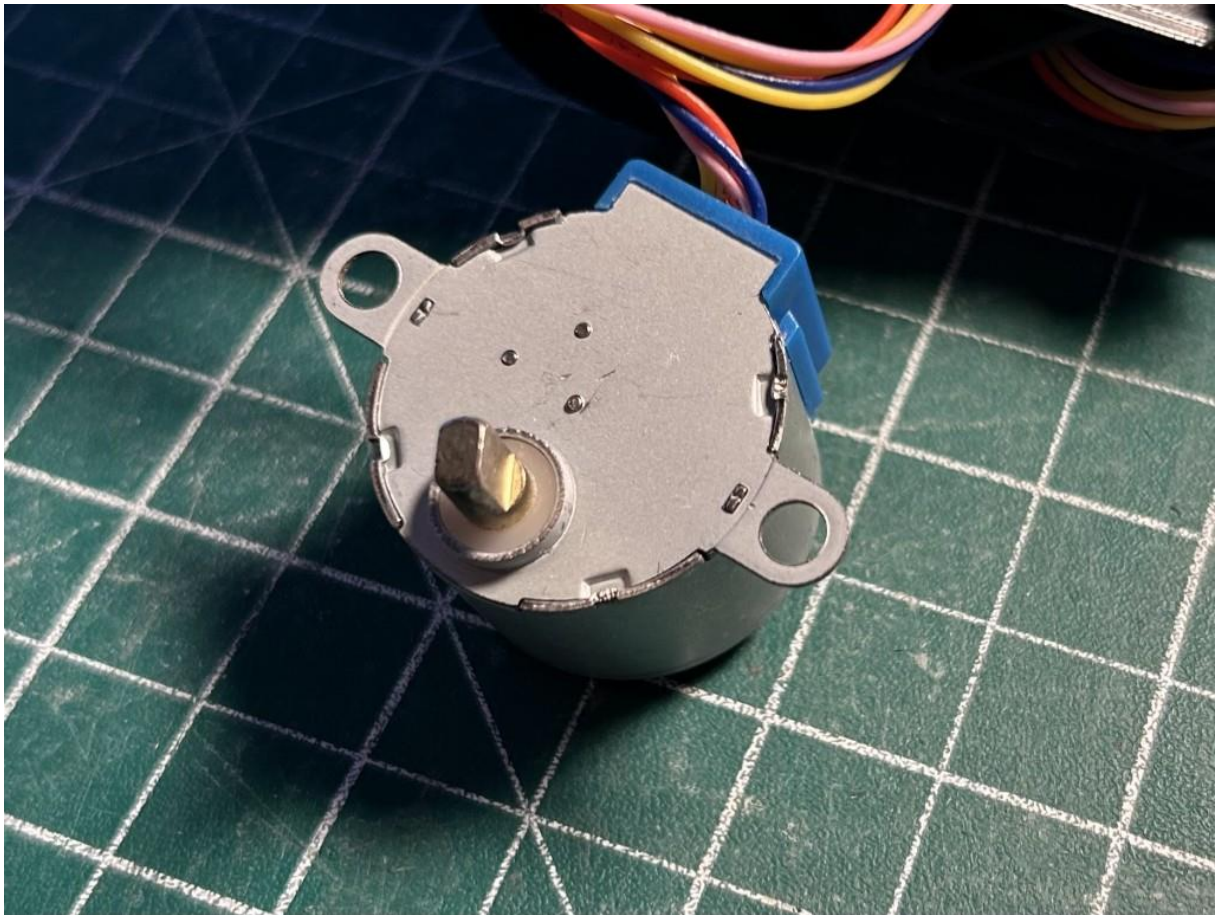
Sachant dans quelle direction nous allons, nous décidons que le drone serait construit autour de son PCB (Printed Circuit Board | Circuit Imprimé). Ainsi, nous commençons par cette étape avec comme objectif de le rendre le plus petit possible.

Le drone serait dans un premier temps contrôlé par une carte de développement de type ESP32. Elles ne sont pas très cher, simple à programmer et s'adaptent largement à nos besoins tout en laissant place à d'éventuelles améliorations du drone par la suite.



L'ESP32 (à droite), fixé sur le circuit imprimé de prototypage.

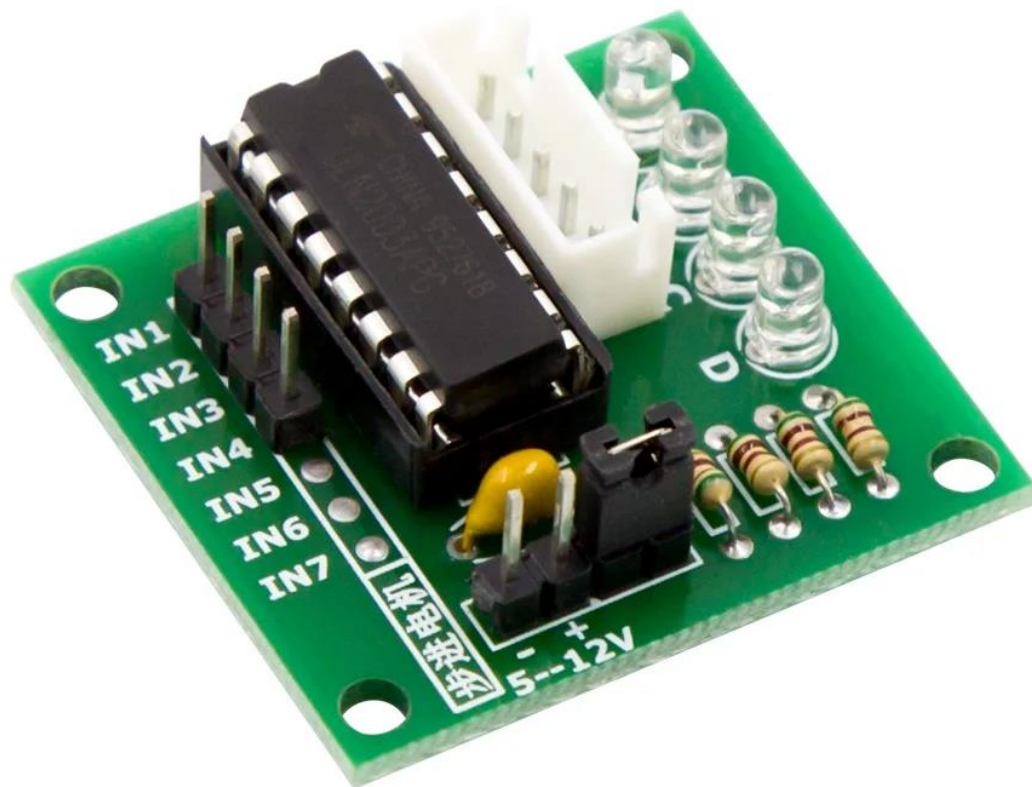
Dans le but de faire le drone le moins cher possible à la production, nous choisissons des moteurs pas-à-pas 28BYJ-48 en 5V. Ce sont les moteurs pas-à-pas de base, ils sont massivement produit, ne coûtent pas cher et sont simple à contrôler. De plus, après quelques tests, nous réalisons qu'ils tournent toujours avec une alimentation de 3.7V.



Un moteur pas-à-pas 28BYJ-48 utilisé sur le drone.

Nous faisons le choix d'alimenter notre drone à l'aide d'une batterie LiPo 3.7V. Nous économisons ainsi sur le plan financier et énergétique. En effet, pour une consommation de x ampères, notre drone sera 25% moins puissant avec une batterie en 3.7V plutôt que 5V, et par conséquent 25% moins énergivore.

Pour les contrôler, ils sont généralement livrés avec des drivers "UNL-2003". Ces derniers mesurent 3 centimètres par 3, il en faut un par moteur et par conséquent il serait bon de pouvoir s'en passer pour gagner de la place.



Driver "UNL-2003" pour contrôler les moteurs pas-à-pas 28BYJ-48

Nous étudions donc leur conception. Leur nom "UNL-2003" vient du circuit intégré éponyme, pièce principale du driver. Ce circuit intégré n'est en réalité qu'un simple inverseur logique. On fait entrer 1 par une broche et on récupère 0 sur la broche d'en face.

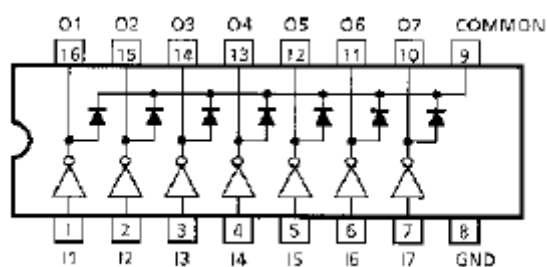


Schéma du circuit intégré UNL-2003

Voyant un circuit intégré si simple, nous nous demandons pourquoi avoir besoin de tout le reste pour que le driver fonctionne. Par une simple étude du circuit imprimé du driver, nous concluons que hors du circuit intégré, le reste de le PCB du driver n'est utile qu'à l'allumage de LEDs pour avoir un retour visuel des ordres de contrôle envoyés au moteur. Nous décidons donc de ne garder de ce driver superflu que le circuit intégré.

Nous n'aurons ainsi qu'à commander les circuits intégrés à quelques centimes l'unité plutôt que le PCB dans son intégralité. Nous simplifions ainsi le circuit de notre drone et une nouvelle fois, nous réalisons des économies financières, de place et énergétiques (suppression des LEDs implique suppression de la consommation liées à ces dernières) !

Notre circuit, désormais simplifié au maximum, ne se résume qu'à une carte de développement ESP32, quelques connecteurs et deux circuits intégrés UNL-2003. Ces derniers ne disposant que de 7 couples de broches entrées/sorties, sachant qu'il en faut 4 par moteur, nous sommes contraints d'en utiliser 2. Remarque nous pourrions à l'avenir, chercher un circuit intégré similaire intégrant plus de broches, mais le drone n'étant pas l'objet principal de notre projet, pour gagner du temps sur le développement de la chaîne de production à proprement parler, nous n'avons pour l'heure pas cherché à l'optimiser à ce point.

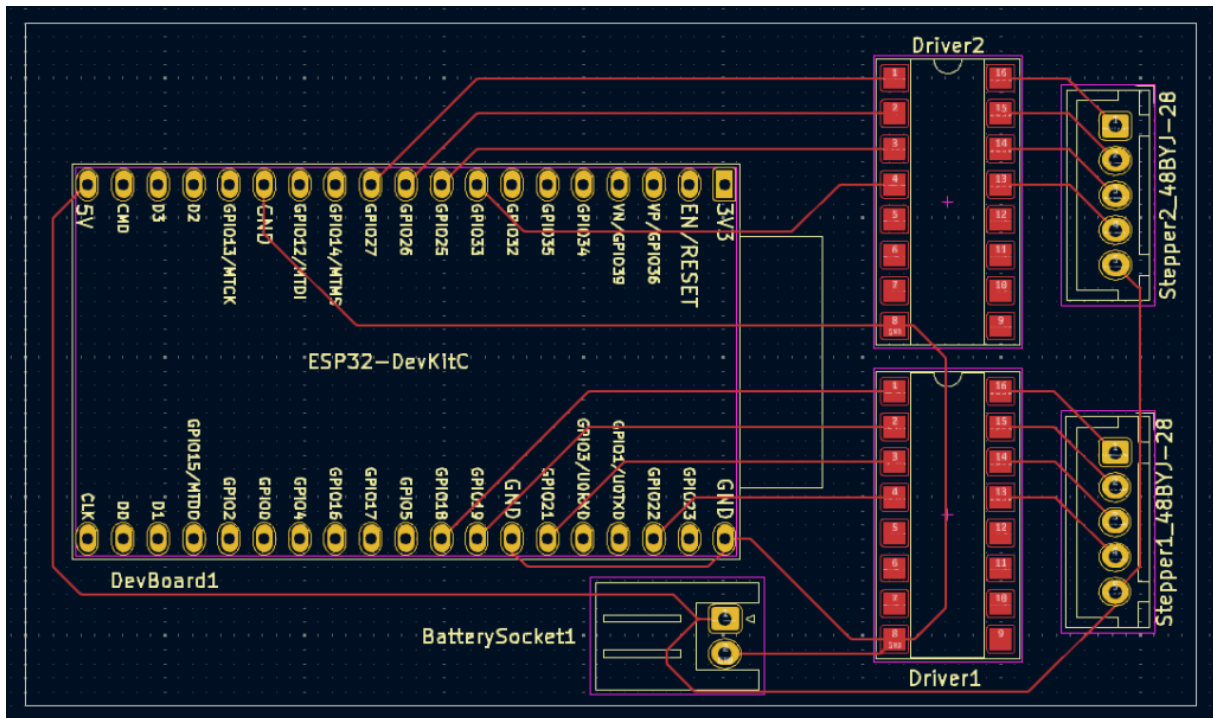
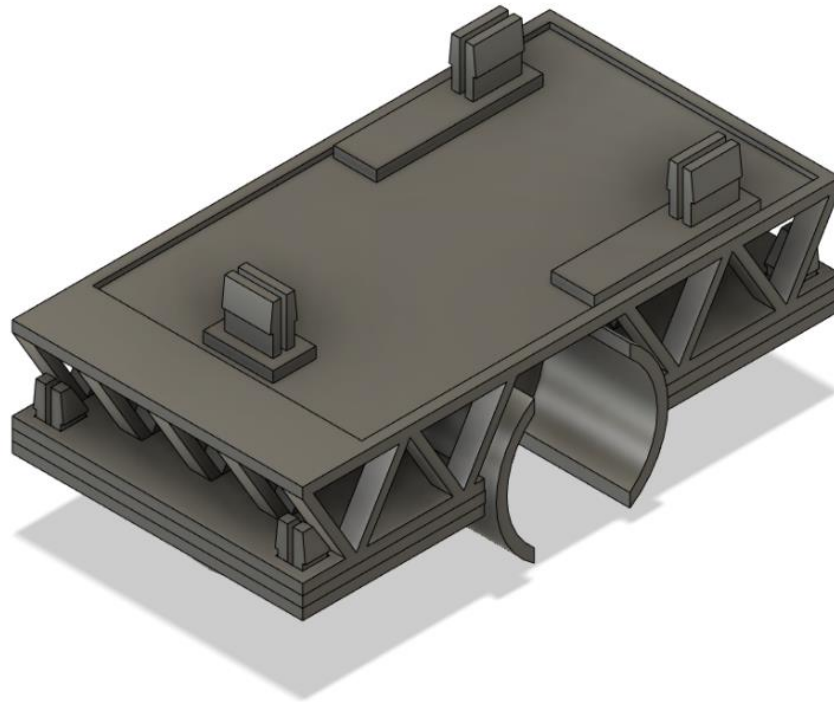


Schéma du PCB de notre drone : Le schéma du circuit est disponible sur le [GitHub](#)

CAD

Une fois le PCB déterminé et dont le prototype ait été fabriqué au sein même du DVIC, nous avons pu nous lancer sur le développement du drone. Pour simplifier l'assemblage nous avons supprimé d'éventuelles vis au profit d'un système d'accroche rapide pour le drone et le PCB et d'encastrement pour les moteurs.

Le drone sera constitué de trois couches. Une base sur laquelle siègent les accroches de notre système de fixation, qui traverseront tout le drone ; un étage intermédiaire destiné au maintien des moteurs latéraux; enfin, un étage supérieur, soutenu par une forêt de supports, sur lequel viendra se fixer notre PCB. La batterie viendra se glisser à l'arrière, dans un emplacement dédié, caché dans la forêt de supports.



Structure du drone, les 3 parties assemblées, modélisée sur Fusion360, fichiers disponibles sur le [GitHub](#)

Partie 2 : La chaîne d'assemblage

Conception

Le drone à assembler prêt, nous pouvions nous pencher sur la chaîne d'assemblage en elle-même. Dans un premier temps nous avons pensé à un système de tapis roulant, bacs à pièces et presses pour tout assembler. Cependant l'idée est de toujours penser au plus simple. Le but est de pouvoir adapter notre chaîne à différents types de drones facilement. Ainsi nous nous sommes demandé comment réduire le nombre d'éléments pour faciliter l'adaptabilité.

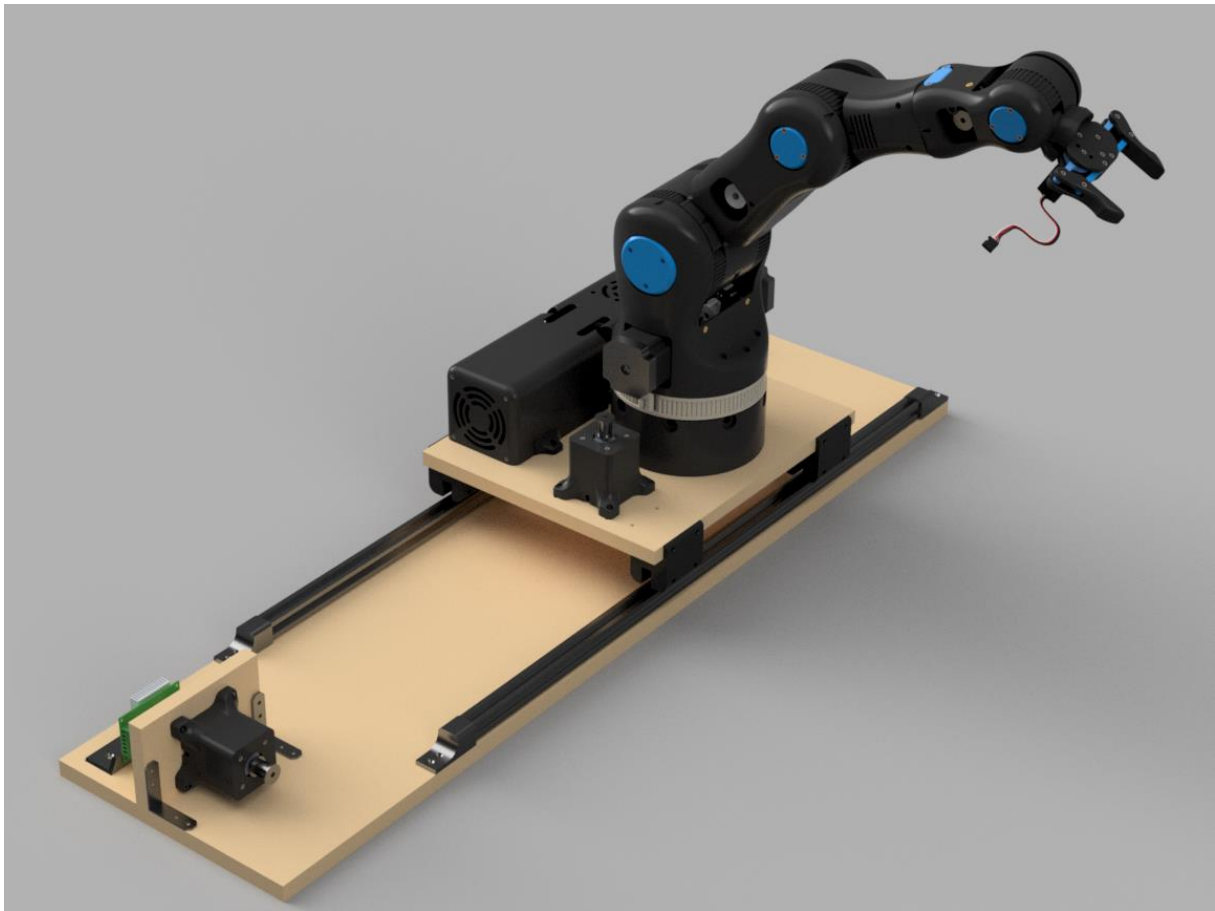
C'est pourquoi nous nous sommes orientés vers un bras robotisé. Très versatile, il nous permettra de réaliser une flopée de tâches de différentes natures : attraper, poser, enfoncer, visser (soyons fou !).

Pour gagner du temps nous nous sommes basés sur le bras open source de BCN3D, lui-même basé sur un autre bras open source. Après lui avoir apporté quelques modifications pour qu'il réponde à nos besoins, nous l'avons mis sur rail pour ajouter un axe de mobilité, étendant le champ des possibles.

Le bras

Comme dit plus haut, la base de notre bras est le Moveo de BCN3D (bras open-source) dont voici le [GitHub](#). Après quelques calculs, nous décidons de changer les moteurs pas-à-pas pour des plus puissants, afin qu'il soit en mesure de réaliser les tâches qui lui incombent. Ce changement impliqua un recalcul de la mécanique du bras, ainsi que de la puissance totale nécessaire pour faire fonctionner la chaîne. (cf. document « *CalculsCouples* »)

Par la suite, nous avons conçu son rail, le bras se déplaçant latéralement sur environ quatre-vingts centimètres, tracté par un moteur pas-à-pas supplémentaire.



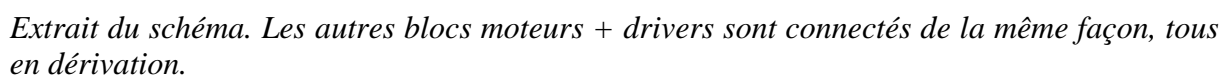
Le MOVEO modifié sur son rail.

Après avoir effectué les calculs mécaniques et électroniques mentionnés plus haut et disponibles dans des documents PDF sur le [GitHub](#), nous jetons nos dévolus sur les moteurs NEMA suivants :

Référence	Type	Couple	Quantité
14HS20-1504S	NEMA 14	40 Ncm	1
17HS13-0404S1	NEMA 17	26 Ncm	1
17HS19-1684S-PG5	NEMA 17 Engrenages 5:1	52 Ncm (moteur seul)	1
17HE24-2104S	NEMA 17	60 Ncm	2
23HE22-2804S	NEMA 23	126 Ncm	2

Concernant l'électronique, le bras sera contrôlé par une carte de développement fonctionnant avec une puce ATmega (un équivalent Arduino MEGA). La carte contrôlera les sept drivers TB6560, le tout étant alimenté par une alimentation de 24V pour 600 watts. Voici un extrait du schéma, le schéma complet et les évaluations de consommation étant disponibles sur le [GitHub](#).

Nous avons choisi les NEMA pour leur rapport qualité/prix et leur disponibilité sur le marché (en plus du fait qu'ils respectent nos contraintes).



GitHub du projet : [GitHub - Assembly Line](#)