

# CONCEPTION, SIMULATION, ET RÉALISATION

**Date :** 19 décembre 2022

**Auteur(s) :** Sébastien Charles

**Copyright :** S. Charles, UVSQ - IUT de Mantes en Yvelines

**Licence :** CC 4.0 BY-NC-SA [<https://creativecommons.org/licenses/by-nc-sa/4.0/deed.fr>] + licence commerciale ET-LIOS [<https://et-lios.s-mart.fr/licencecommerciale/>]

# Table des matières

|  |    |
|--|----|
| Introduction .....   | 5  |
| 1. Programmer et développer des systèmes avec Arduino, initiation .....          | 6  |
| 1.1. Introduction  | 6  |
| 1.2. L'environnement de développement  | 6  |
| 1.3. La carte Arduino Uno  | 13 |
| 1.4. Réaliser son premier montage : contrôleur de LED                            | 17 |
| 1.4.1. Introduction  | 17 |
| 1.4.2. Comprendre le principe de MLI   | 17 |
| 1.4.3. Déclaration d'une broche comme entrée                                     | 18 |
| 1.4.4. Délai entre deux exécutions de boucles                                    | 18 |
| 1.4.5. Réalisation du montage contrôleur de LED                                  | 20 |
| 1.5. Contrôleur de LED RGB   | 24 |
| 1.5.1. Introduction  | 24 |
| 1.5.2. Principe du RVB ou RGB  | 24 |
| 1.5.3. Montage et exécution du contrôleur de LED RGB                             | 25 |
| 1.6. Pilotage de LED par des interrupteurs                                       | 27 |
| 1.6.1. Introduction  | 27 |
| 1.6.2. Montage et pilotage de LED par interrupteurs                              | 27 |
| 1.7. Le moniteur série   | 29 |
| 1.7.1. Introduction  | 29 |
| 1.7.2. Afficher dans le moniteur série le caractère entré par l'utilisateur      | 29 |
| 1.7.3. Afficher dans le moniteur série des variables                             | 31 |
| 1.8. Multimètre à 2 voies  | 32 |
| 1.8.1. Introduction  | 32 |
| 1.8.2. Montage, mesure de variations de tensions et traçage de courbes           | 32 |
| 1.9. Buzzer actif  | 36 |
| 1.9.1. Introduction  | 36 |
| 1.9.2. Montage et pilotage d'un buzzer actif                                     | 36 |
| 1.10. Buzzer passif  | 37 |
| 1.10.1. Introduction   | 37 |
| 1.10.2. Montage et pilotage d'un buzzer passif                                   | 38 |
| 1.11. Tilt   | 40 |
| 1.11.1. Introduction   | 40 |
| 1.11.2. Montage et allumage d'une LED par tilt                                   | 41 |
| 2. Programmer et développer des systèmes avec Arduino, perfectionnement .....    | 43 |
| 2.1. Introduction  | 43 |
| 2.2. Contrôleur d'afficheur 7 segments   | 43 |
| 2.2.1. Introduction  | 43 |
| 2.2.2. Afficheur à 7 segments  | 44 |
| 2.3. Utilisation d'un registre pour contrôler un afficheur 4 digits à 7 segments | 46 |
| 2.3.1. Introduction  | 46 |
| 2.3.2. Pilotage d'un afficheur 7 segments à l'aide d'un registre à décalage.     | 47 |

|   |     |
|---|-----|
| 2.3.3. Pilotage d'un afficheur 7 segments à l'aide d'un registre à décalage, bibliothèque SevSeg.h. | 49  |
| 2.4. Utilisation d'un photoresistor   | 50  |
| 2.4.1. Introduction   | 50  |
| 2.4.2. Utilisation d'un photoresistor   | 51  |
| 2.5. Mesure de distance par ultrason  | 53  |
| 2.5.1. Introduction   | 53  |
| 2.5.2. Mesure de distance par ultrason  | 54  |
| 2.6. Mesure de distance par télémètre laser   | 56  |
| 2.6.1. Introduction   | 57  |
| 2.6.2. Mesure de distance par télémètre laser   | 57  |
| 2.7. Les moteurs électriques  | 59  |
| 2.7.1. Introduction   | 59  |
| 2.7.2. Classification des moteurs électriques   | 60  |
| 2.8. Contrôle de servomoteur  | 69  |
| 2.8.1. Introduction   | 69  |
| 2.8.2. Contrôle de servomoteur avec la bibliothèque Servo.h   | 69  |
| 2.9. Contrôle de 2 moteurs à courant continu  | 71  |
| 2.9.1. Introduction   | 71  |
| 2.9.2. Contrôle de 2 moteurs à courant continu  | 72  |
| 2.10. Contrôle de moteur à courant continu avec les drivers L293D et L298N                          | 74  |
| 2.10.1. Introduction  | 74  |
| 2.10.2. Contrôle de moteur à courant continu avec les drivers L293D et L298N                        | 75  |
| 2.10.3.   | 76  |
| 2.11. Contrôle de moteur pas-à-pas  | 79  |
| 2.11.1. Introduction  | 79  |
| 2.11.2. Contrôle de moteur pas-à-pas par commande simplifiée  | 80  |
| 2.11.3. Contrôle de moteur pas-à-pas par commande permettant de développer le couple maximal        | 83  |
| 2.11.4. Contrôle de moteur pas-à-pas par commande demi-pas  | 84  |
| 2.11.5. Contrôle de moteur pas-à-pas avec bibliothèque stepper Arduino                              | 86  |
| 2.12. Contrôle de moteur pas-à-pas avec driver TB6600   | 87  |
| 2.12.1. Introduction  | 87  |
| 2.12.2. Contrôle de moteur pas-à-pas avec driver TB6600   | 88  |
| 3. Le prototypage rapide en mécanique .....   | 91  |
| 3.1. Introduction   | 91  |
| 3.2. Présentation générale de l'impression 3D   | 91  |
| 3.3. Les secteurs et acteurs sur le marché des imprimantes 3D                                       | 92  |
| 3.4. Les technologies de l'impression 3D  | 97  |
| 3.4.1. Introduction   | 97  |
| 3.4.2. La photopolymérisation   | 98  |
| 3.4.3. L'extrusion  | 102 |
| 3.4.4. La projection  | 104 |
| 3.4.5. La fusion de poudre  | 106 |
| 3.4.6. La stratoconception  | 111 |
| 3.4.7. L'impression 4D  | 112 |

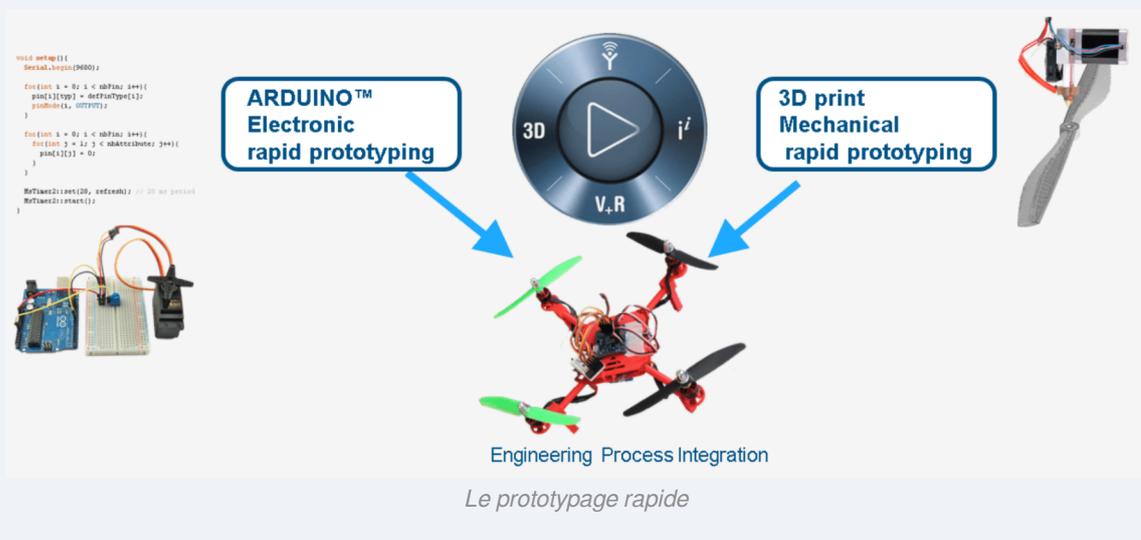
|   |     |
|---|-----|
| 3.4.8. Comparaison des procédés   | 114 |
| 3.5. Les matériaux utilisés en impression 3D  | 114 |
| 3.5.1. Introduction   | 114 |
| 3.5.2. Les plastiques   | 114 |
| 3.5.3. Les métaux utilisés en SLS, EBM et DED   | 117 |
| 3.5.4. Les céramiques   | 117 |
| 3.5.5. Les matériaux organiques   | 118 |
| 3.5.6. Les pigments   | 118 |
| 3.5.7. Les matériaux hybrides   | 119 |
| 3.5.8. Compatibilité entre les technologies d'impression 3D et les matériaux                  | 119 |
| 3.5.9. Comment choisir un matériau ?  | 120 |
| 3.6. Bonnes pratiques de l'impression 3D  | 121 |
| 3.7. La découpe laser   | 124 |
| 3.8. Les scanners 3D  | 128 |
| 4. Réalisation un jumeau numérique d'un système à base d'Arduino dans la 3DExperience .....   | 132 |
| 4.1. Introduction   | 132 |
| 4.2. Jumeau Numérique d'un Servomécanisme piloté par Arduino et modélisé dans la 3DExperience | 132 |
| Glossaire .....   | 241 |

# Introduction

**Objectifs pédagogiques :** L'objectif de cette ressource est de vous donner les concepts et outils nécessaires à la conception et à la réalisation de prototypes qui intègrent des composants électroniques et mécaniques.

L'approche se concentre sur les solutions de prototypage rapide, tant électronique que mécanique, à base d'Arduino et d'impression 3D, qui permettent de réaliser à moindre coût et dans des temps très courts des preuves de concept qui intègrent des composants mécatroniques interactifs.

La création de jumeaux numériques de prototypes à base d'Arduino par le biais de la 3DExperience sera également abordée dans ce module.



# 1. Programmer et développer des systèmes avec Arduino, initiation

## 1.1. Introduction

### Objectifs pédagogiques

L'objectif de cette ressource est de vous initier à l'utilisation des solutions Arduino pour développer des systèmes interactifs simples. Des notions d'électronique seront également abordées dans ce contenu.

### ↳ Déroulement

Durée : 12 heures

Arduino propose un environnement de développement logiciel et une carte électronique multifonction permettant de réaliser en open source des prototypes électroniques interactifs rapidement et à moindre coût.

L'environnement de développement permet de programmer des fonctions électroniques par le biais d'un langage de haut niveau simplifié, proche du langage C. En complément de cet environnement, il existe de nombreuses bibliothèques facilitant l'usage de composants électroniques actifs et passifs.

La carte principale utilise un microcontrôleur pour traiter l'information et commander les composants et est équipée de connecteurs d'entrée / sortie numériques et analogiques pouvant être reliés à d'autres composants par des fils de type Dupont.

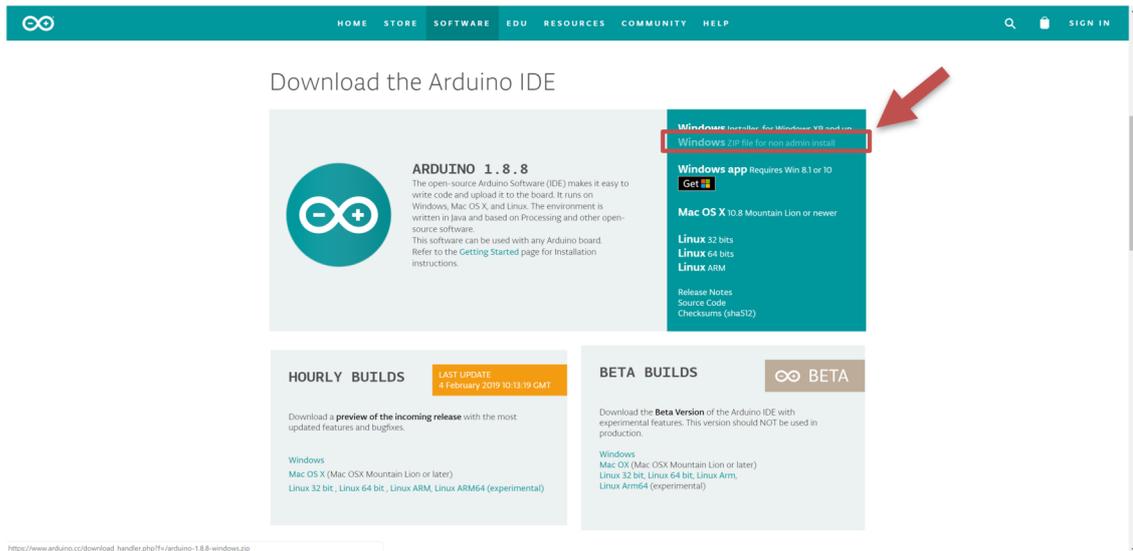
Site officiel Arduino <sup>[<https://www.arduino.cc/>]</sup>

## 1.2. L'environnement de développement

En premier lieu, il faut soit télécharger l'environnement de développement Arduino, soit utiliser la version web de l'IDE qui ne nécessite aucune installation.

Lien : Télécharger l'IDE <sup>[<https://www.arduino.cc/en/Main/Software>]</sup>.

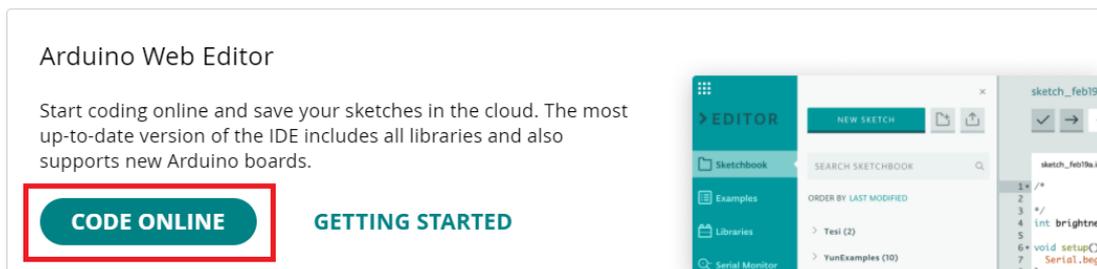
Si vous n'avez pas les droits d'administrateur sur vos sessions Windows, il vous faudra télécharger la version `Windows Zip file for non admin`.



IDE pouvant être installé sans disposer des droits d'administrateur

Ceci est possible car l'IDE est codé en java, qui est un langage interprété par une machine virtuelle (non compilé) et donc compatible avec tous les systèmes d'exploitation et exécutable sans installation.

Pour utiliser la version web, cliquer ici [\[https://www.arduino.cc/en/Main/Software\]](https://www.arduino.cc/en/Main/Software), puis choisir **CODE ONLINE**.

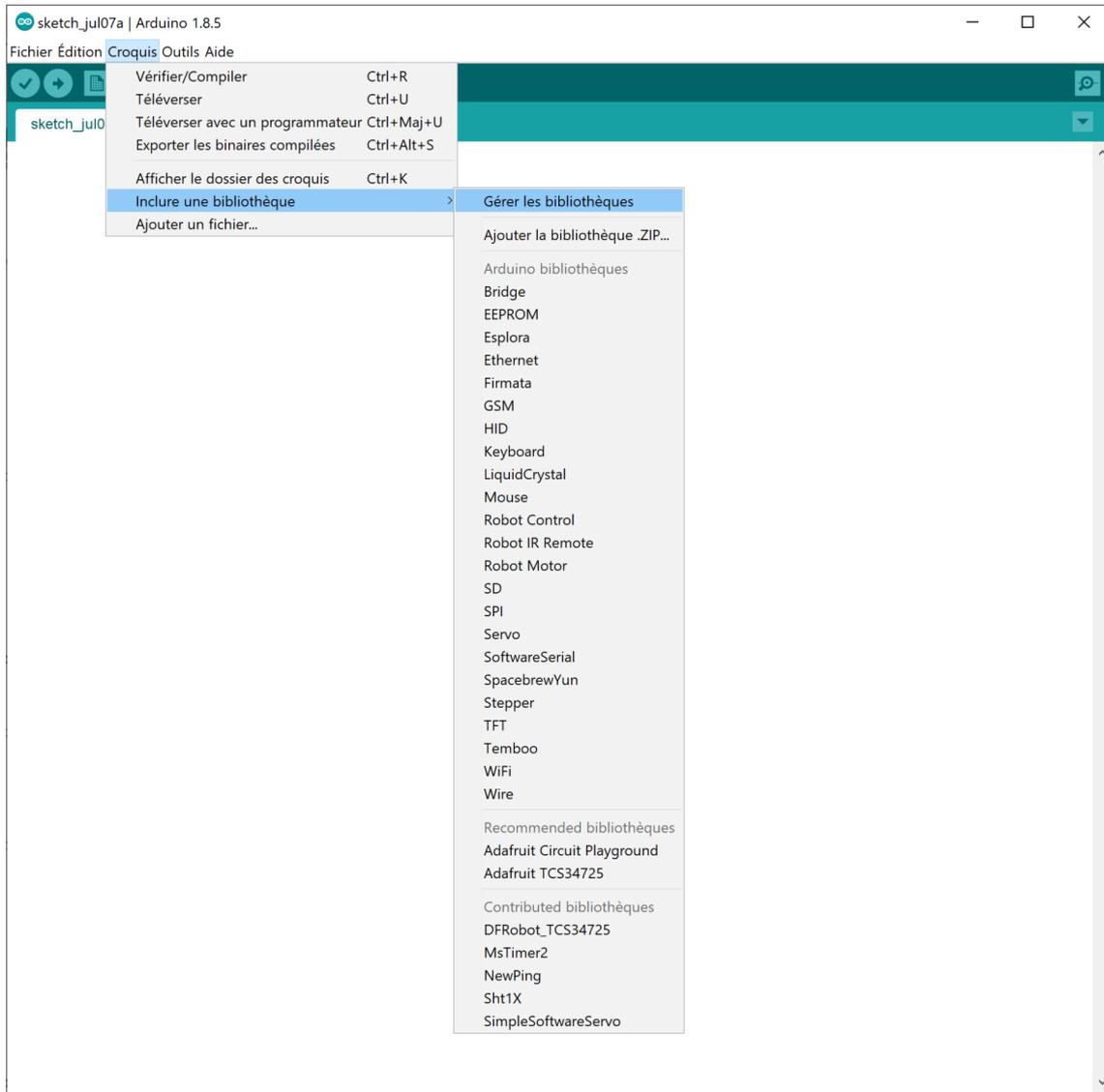


IDE version web

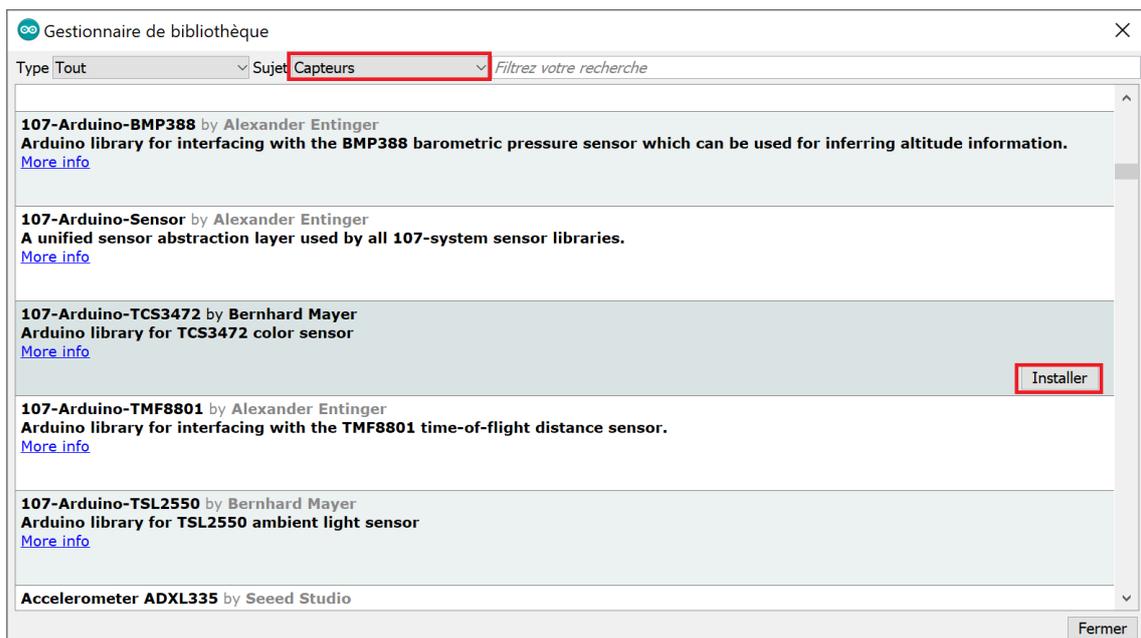
L'IDE est un environnement de développement qui permet d'implémenter un algorithme de contrôle / commande dans le micro-contrôleur de la carte Arduino (généralement un ATMEGA 328) par le biais d'un connecteur USB qui émule le port série traditionnel.

Certains programmes nécessitent des bibliothèques de fonctions qui simplifient la programmation.

Pour installer une nouvelle bibliothèque, le plus simple est d'utiliser le gestionnaire de bibliothèque en cliquant sur le menu **Croquis**, puis **Inclure une bibliothèque/Gérer une bibliothèque** :



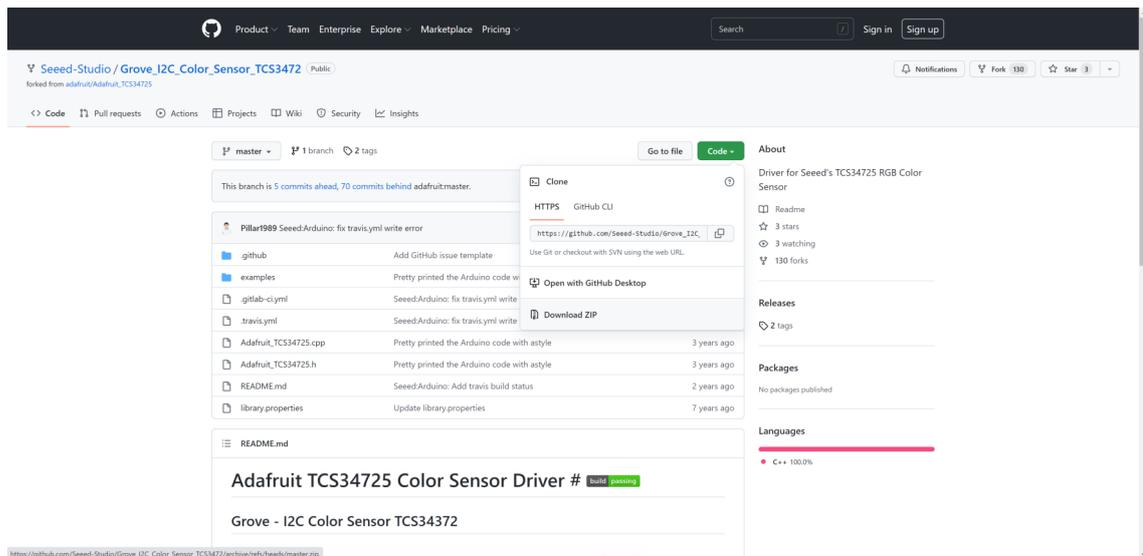
Ensuite, il vous suffit de rechercher la bibliothèque souhaitée, par mot clé ou par catégorie, par exemple dans la capture d'écran ci-dessous, nous ajoutons la bibliothèque d'un capteur de couleur TCS3472 en cliquant sur **Installer** :



Cette méthode nécessite toutefois l'utilisation de ports réseaux particuliers qui ne seront peut-être pas ouverts au niveau du proxy de votre établissement. Depuis votre connexion personnelle, il ne devrait pas y avoir de blocage. Si vous souhaitez installer des bibliothèques depuis votre établissement seule la méthode manuelle fonctionnera.

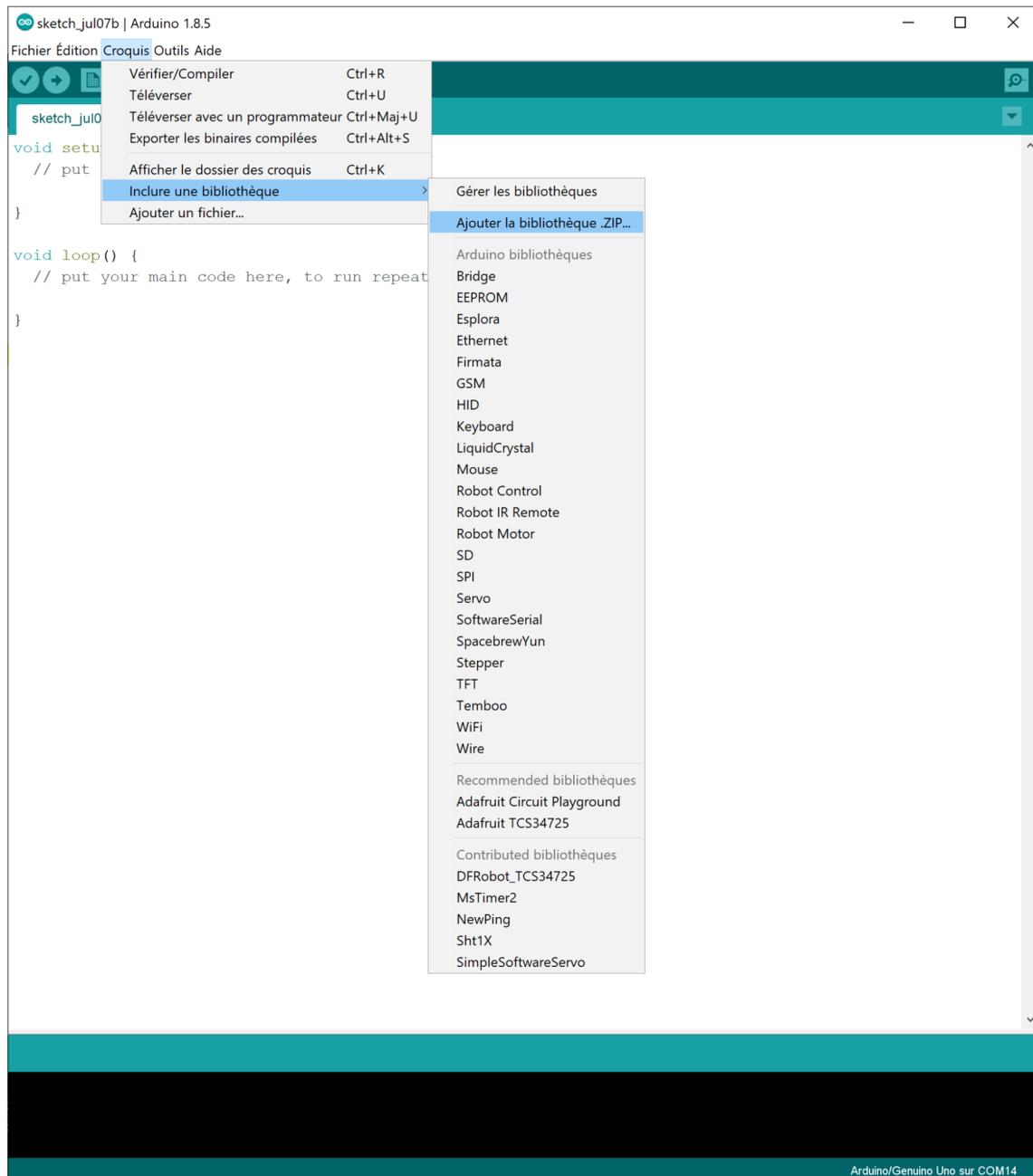
Pour ajouter manuellement des bibliothèques dans l'IDE Arduino, vous pouvez la télécharger sur un site, puis cliquer sur le menu **Croquis**, puis **Inclure une bibliothèque/Ajouter une bibliothèque .ZIP**.

Dans l'exemple ci-dessous, nous téléchargeons le zip de la bibliothèque du capteur en allant sur GitHub, puis en cliquant sur **Code**, puis sur **Download ZIP** :

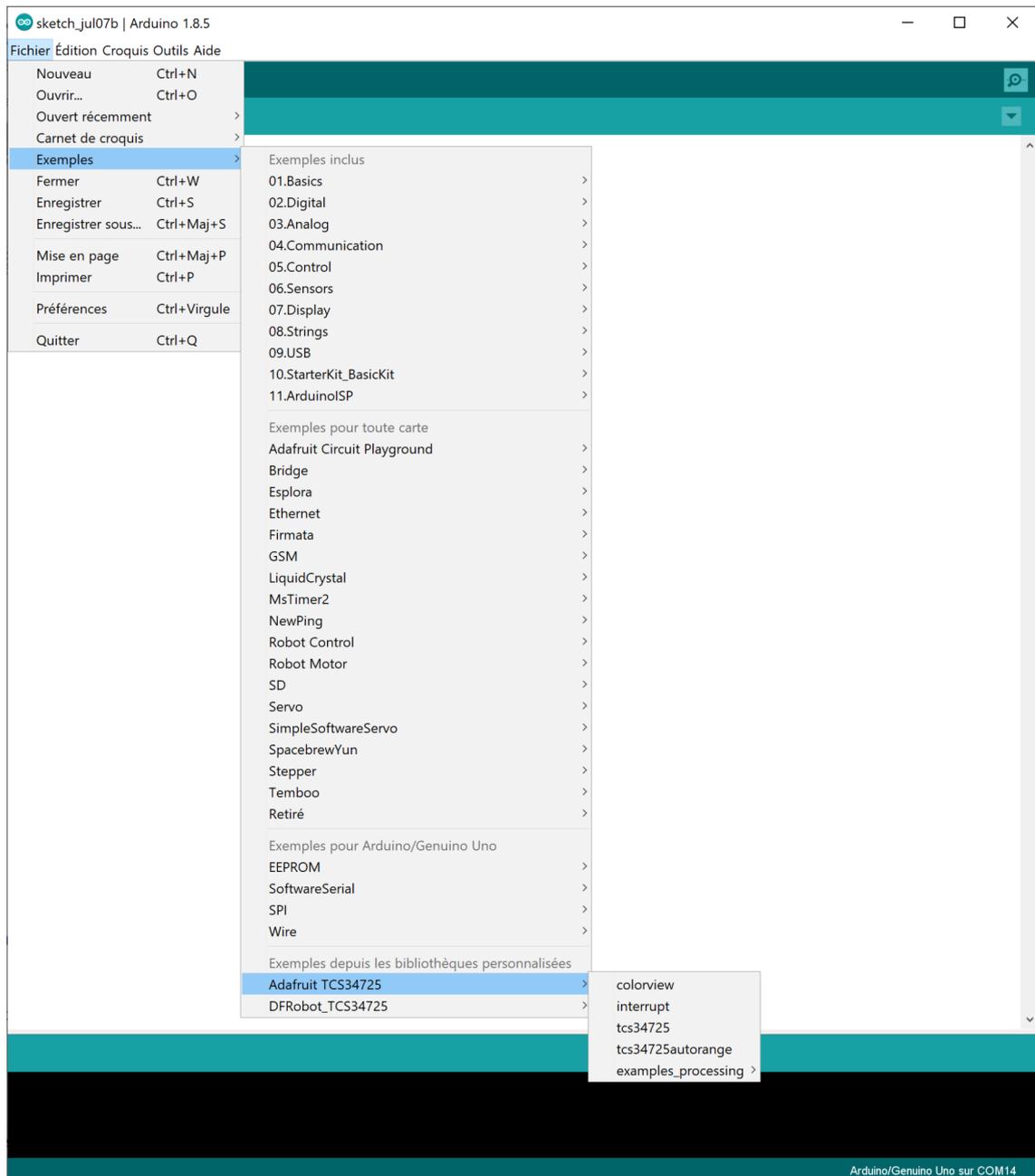


Le lien vers la page web ci-dessus : [GitHub \[https://github.com/Seeed-Studio/Grove\\_I2C\\_Color\\_Sensor\\_TCS3472\]](https://github.com/Seeed-Studio/Grove_I2C_Color_Sensor_TCS3472)

Ensuite, il suffit de cliquer sur le menu **Croquis**, puis **Inclure une bibliothèque/Ajouter une bibliothèque .ZIP** et de choisir le fichier téléchargé.



Une fois une bibliothèque téléchargée, vous obtenez de nouveaux exemples de code Arduino « Clés en main » en cliquant sur le menu **Fichiers/Exemples**, les bibliothèques téléchargées sont présentées en fin de liste.



### ⚠ Attention

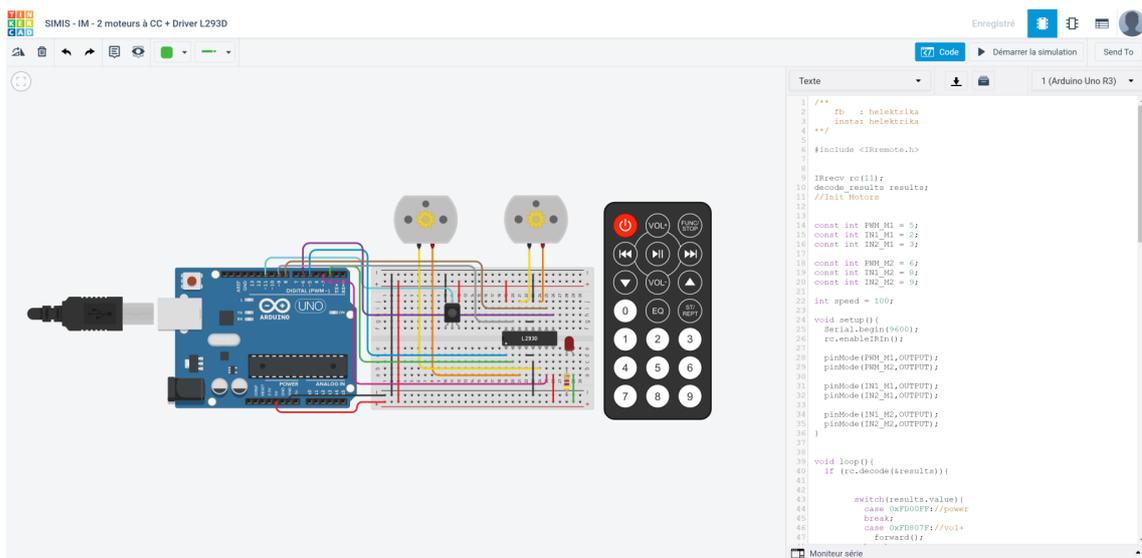
Les exemples de programmes présents dans l'IDE Arduino sont très utiles pour développer rapidement un code permettant de faire fonctionner un composant électronique.

Vous pouvez aussi installer une bibliothèque en l'incluant dans le dossier dédié à l'IDE. Pour ce faire, il suffit de fermer l'IDE, de coller les nouvelles bibliothèques dans le dossier intitulé `libraries` situé à la racine du dossier de votre IDE et de le relancer.

Cette ressource a pour objet de vous initier à l'Arduino et de vous apprendre à utiliser des capteurs et actionneurs courants pour prototyper un système mécatronique. Pour compléter vos compétences en Arduino, consultez le cours OpenClassrooms qui est bien plus complet : Programmez vos premiers montages avec Arduino [<https://openclassrooms.com/fr/courses/2778161-programmez-vos-premiers-montages-avec-arduino>]

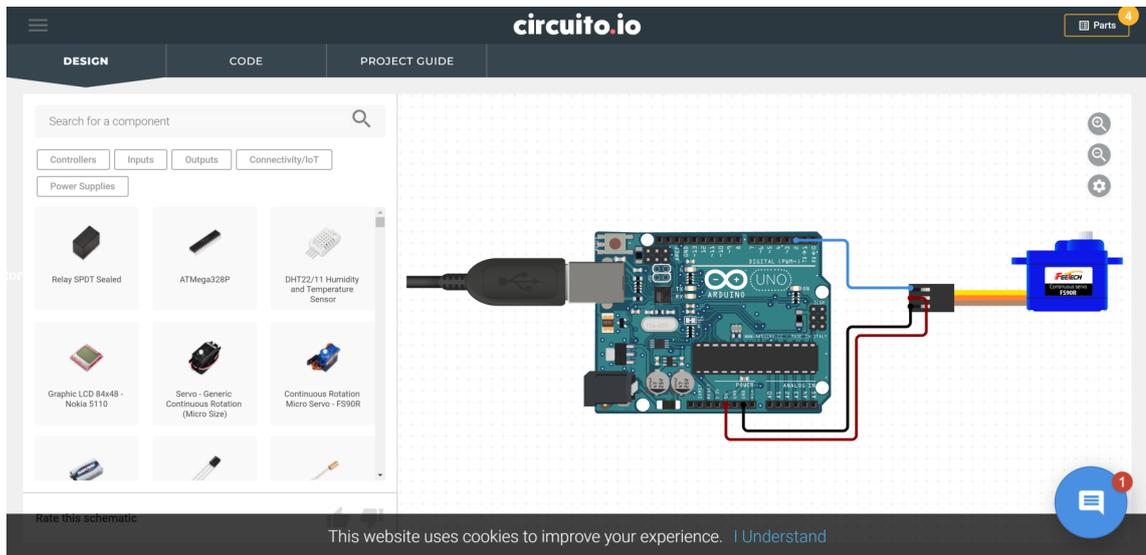
## ✦ Pour aller plus loin Si vous ne disposez pas de matériel électronique

Si vous ne disposez pas de carte Arduino, vous pouvez toujours réaliser le TP en allant sur le site Tinkercad [<https://www.tinkercad.com/>] qui, de manière collaborative, permet de simuler le fonctionnement d'une carte Arduino de manière réaliste (il s'agit d'une émulation). Cette solution entièrement web permet aussi de créer des modèles 3D et de les exporter au format STL pour les imprimer par exemple par fabrication additive. L'environnement web propose également de partager de manière communautaire les maquette 3D, les montages électroniques et les programmes réalisés.



## ✦ Pour aller plus loin Pour facilement composer vos propres montages

Vous pouvez utiliser le site Circuito.io [<https://www.circuito.io/>] qui propose un environnement web permettant de réaliser des montages électroniques avec Arduino et un large choix de composants. Cet environnement intelligent génère automatiquement les montages et programmes associés par simple glisser/déplacer des composants dans la fenêtre de création.

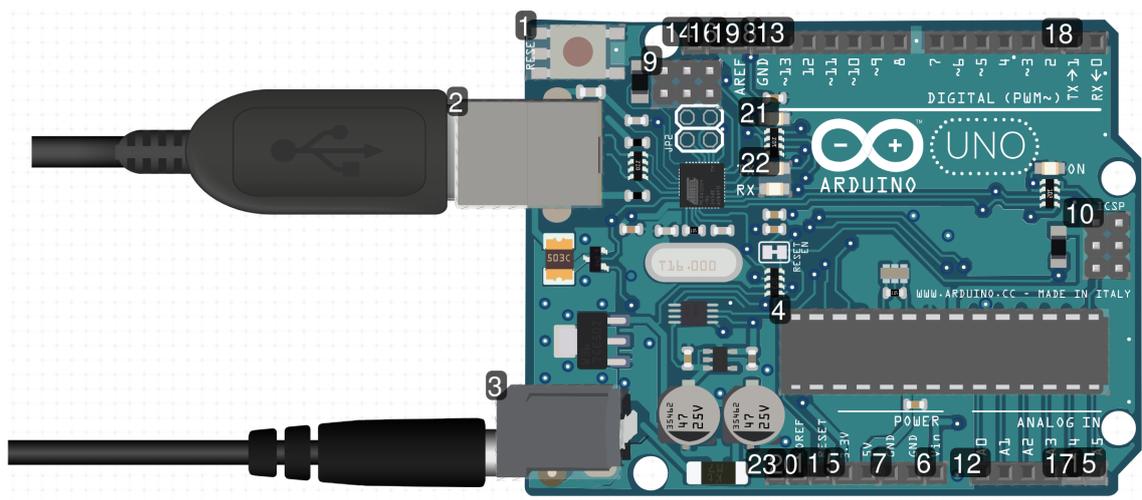


Pour concevoir des montages Arduino et plus généralement des circuits électroniques, la solution Fritzing [<https://fritzing.org/>] peut aussi vous être utile.

### 1.3. La carte Arduino Uno

La carte Arduino est une carte électronique permettant de programmer en langage C de manière simplifié. Si nous utilisons une carte de programmation, c'est qu'elle nous permet de faire des branchements électroniques et donc de relier l'hardware au software.

#### Constitution d'une carte Arduino Uno

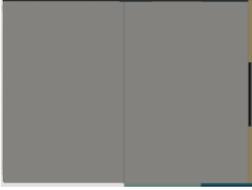


**1** Bouton de remise à zéro du programme « RESET »



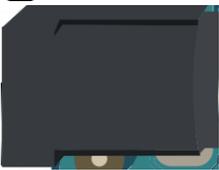
Ce bouton sert à réinitialiser le programme pour qu'il s'exécute depuis la première ligne de code.

## 2 Connecteur USB Type B (5V - 500mA max)



Ce connecteur permet d'alimenter la carte et de communiquer avec le microcontrôleur via un port COM virtuel (car la liaison physique USB fonctionne en série alors qu'un COM est utilisé pour des liaisons en parallèle).

## 3 Connecteur d'alimentation externe (7-12V)



Connecteur Rayon 2.1mm + au milieu

## 4 Microcontrôleur ATMEL de type ATMEGA 328



## 5 Broches d'alimentation de composants externes (3.3V et 5V - 500mA max)



## 6 Broche d'alimentation de la carte



## 7 Broche de masse (0V)



## 8 Broche de masse (0V)



## 9 Broches ICSP



Permet de brancher une sonde de programmation ICSP pour programmer le microcontrôleur sans passer par le connecteur USB.

Cela permet de reprogrammer le microcontrôleur sans passer par l'USB et pour exécuter son programme en mode debug pour suivre en temps réel l'exécution des instructions et vérifier l'état de la mémoire.

Ces pins ICSP sont aussi mappés sur un bus SPI.

## 10 Broches ICSP



Permet de brancher une sonde de programmation ICSP pour programmer le microcontrôleur sans passer par le connecteur USB.

Cela permet de reprogrammer le microcontrôleur sans passer par l'USB et pour exécuter son programme en mode debug pour suivre en temps réel l'exécution des instructions et vérifier l'état de la mémoire.

Ces pins ICSP sont aussi mappés sur un bus SPI.

## 11 Broches pour piloter le reset de la carte



## 12 Broches Analogiques A0 à A5



## 13 Broches Numériques D2 à D13



Certaines de ces broches peuvent être pilotées de manière numérique ou analogique (à préciser lors de l'initialisation des entrées / sorties) : 3,5,6,9,10,11,13 pour piloter des sorties en PWM.

Les broches 2 et 3 peuvent être utilisées pour gérer les interruptions, respectivement 0 et 1.

## 14 Broche I2C (SCL)



## 15 Broche I2C (SCL)



## 16 Broche I2C (SDA)



## 17 Broche I2C (SDA)



## 18 Broches de communication en série asynchrone reliées au connecteur USB



D0 : Rx : entrée

D1 : Tx : sortie

## 19 Broche AREF



La broche AREF permet de fixer la valeur haute de la tension de référence pour les entrées analogiques. Si cette broche n'est pas reliée à une source de tension, la tension de référence est alors de 5V pour une UNO.

### 20 Broche IOREF

Cette broche fournit une tension similaire à celle utilisée par les entrée / sortie de la carte, par exemple 5V pour une UNO ou 3.3V pour une DUE.

### 21 LED intégrée reliée au connecteur numérique D13

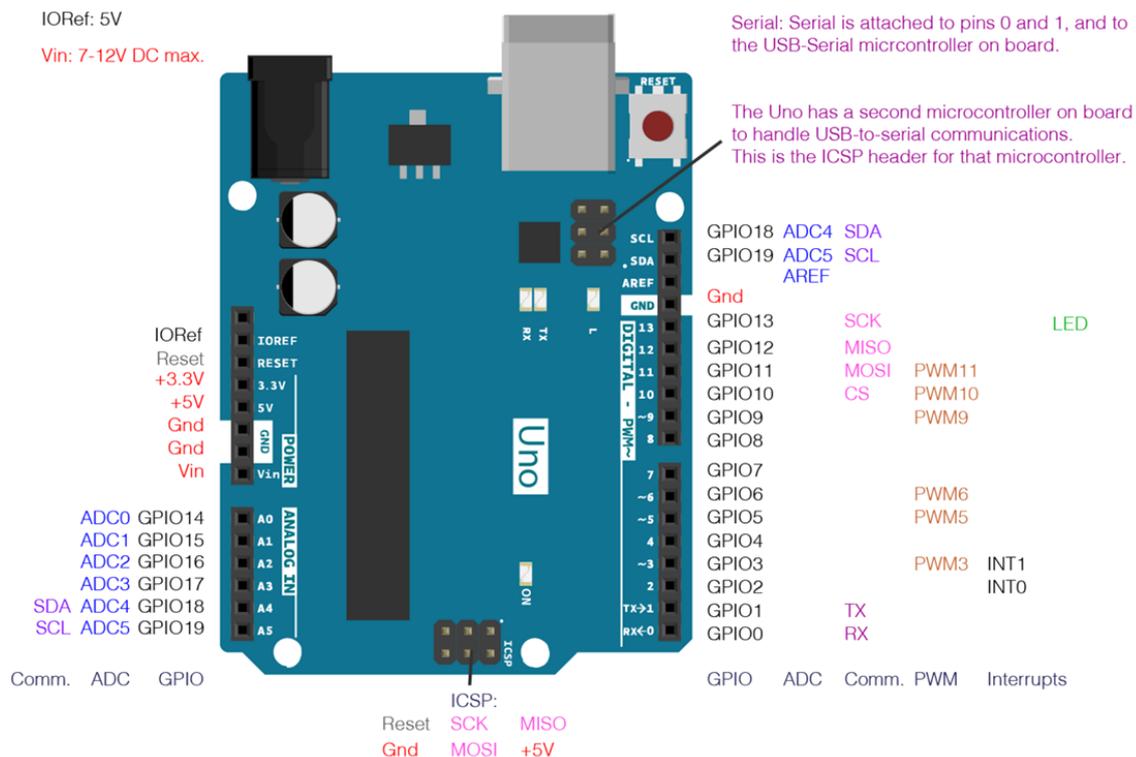
Permet de tester le fonctionnement de la carte sans connecter de composants externes.

### 22 2 LED indiquant l'usage de la liaison série TX et de RX

### 23 Broche inutile

Cette broche n'est pas reliée à la carte.

Sa présence s'explique sans doute pour éviter une découpe coûteuse du connecteur 8 broches lors de la fabrication en série



Détails des connecteurs de la carte

Une carte Arduino, bien que pratique, est limitée à l'usage de composants qui fonctionnent en basse tension.

Ses limites sont les suivantes :

- La carte peut supporter une tension d'entrée de 20 Volts au grand maximum.
- Intensité maximale disponible par connecteur d'entrée/sortie sous une tension de 5V est de 40 mA (avec un total maximal de 200 mA sur l'ensemble de ses connecteurs).
- Intensité maximale disponible pour la sortie 3,3V : 50 mA.
- Intensité maximale disponible pour la sortie 5V : 500 mA en cas d'alimentation par le port USB seul, sinon c'est en fonction de l'alimentation utilisée.

Si vous souhaitez contrôler des composants nécessitant des tensions ou des intensités plus importantes, il faut utiliser des cartes de puissance telles que des motor shield, driver ou relais. La carte Arduino assure dans ce cas uniquement le rôle de contrôle/commande.

## 1.4. Réaliser son premier montage : contrôleur de LED

---

### 1.4.1. Introduction

---

Le premier exercice va consister à réaliser un contrôleur de LED par Modulation de Largeur d'Impulsions (MLI ; en anglais : Pulse Width Modulation, soit PWM) qui va faire accroître puis décroître son intensité lumineuse.

Pour réaliser ce montage, il faudra vous munir des composants suivants :

- une carte arduino
- un câble USB
- une LED
- un résistor de 220 Ohms
- des fils Dupont Mâle/Mâle
- une platine de prototypage rapide à trous de type Labdec

La plupart des montages et programmes présentés dans le module d'initiation sont basés sur le kit de développement de UCTRONICS, que vous pouvez vous procurer sur : UCTRONICS [<https://www.uctronics.com/maker-kits/arduino-kits/uctronics-advanced-starter-kit-for-arduino-with-instruction-booklet-uno-r3-uno-r3-protoshield-v3-relay-breadboard-power-supply-sg90-9g-servo-remote-controller-and-ir-receiver.html>]

### 1.4.2. Comprendre le principe de MLI

---

Le principe de MLI est très utilisé pour piloter des composants électroniques analogiques, c'est pourquoi il convient de le comprendre avant de l'utiliser dans le premier montage.

Consultez l'article MLI sur le site Wikipédia [\[https://fr.wikipedia.org/wiki/Modulation\\_de\\_largeur\\_d%27impulsion\]](https://fr.wikipedia.org/wiki/Modulation_de_largeur_d%27impulsion) et la vidéo Qu'est-ce que la PWM ? [\[https://www.youtube.com/watch?v=CSReyYwbGRY\]](https://www.youtube.com/watch?v=CSReyYwbGRY)

## Question

Expliquez le principe de MLI en quelques lignes.

### 1.4.3. Déclaration d'une broche comme entrée

Le premier bloc d'un code Arduino contient les déclarations des variables du programme :

```
1 int ledpin=11; //précise que la broche numérique 11 sera utilisée par le programme
```

Le deuxième bloc contient une fonction qui est appelée une seule fois à chaque exécution du programme :

```
1 void setup () // ce bloc du code n'est exécutée qu'une seule fois à chaque
  initialisation
2 {
3   pinMode(ledpin,OUTPUT); // déclare que la broche 11 sera uniquement utilisée
  comme sortie
4 }
```

## ⚠ Attention

Pour comprendre et connaître la liste des instructions disponibles en Arduino, veuillez consulter le site de références Arduino [\[https://www.arduino.cc/reference/en/#page-title\]](https://www.arduino.cc/reference/en/#page-title).

## Question

Quelle est l'instruction pour déclarer qu'une broche est utilisée comme entrée ?

### 1.4.4. Délai entre deux exécutions de boucles

Le troisième bloc contient une fonction exécutée en boucle jusqu'à l'appui du bouton reset ou l'arrêt de l'alimentation de la carte. Le micro-contrôleur tourne naturellement à 16 MHz, et les boucles tournent généralement à environ 100 000 instructions par seconde (variable en fonction de la complexité des instructions) si aucun minuteur (delay) ne vient ralentir son exécution.

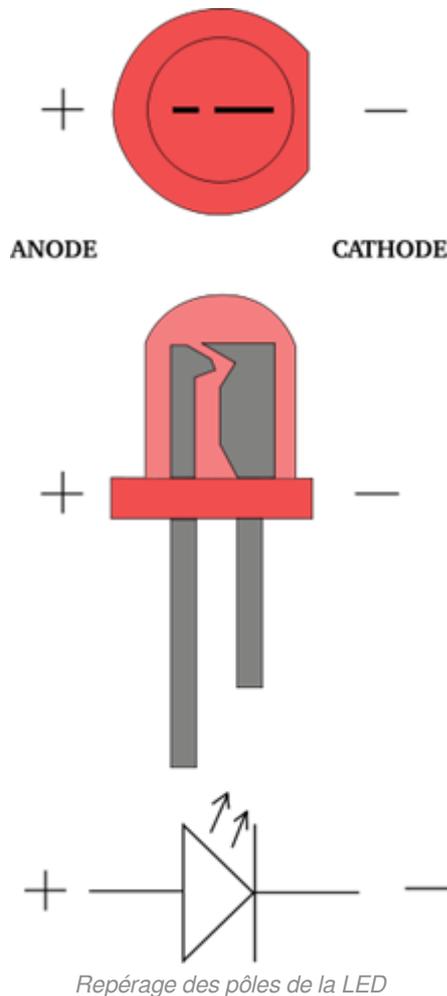
```
1 void loop() // ce bloc est exécuté en boucle jusqu'au débranchement de la carte ou
  appui sur le bouton reset
2 {
3   for (int a=0; a<=255;a++) // cette boucle permet d'incrémenter l'éclairage de
  la led par le biais d'un contrôle de type PWM (analogique), ceci est possible car le
  connecteur ou pin 11 a la particularité de pouvoir être utilisé en analogique ou
  numérique
4   {
5     analogWrite(ledpin,a); // cette boucle envoie la valeur de PWM à la sortie 11
  qui est déclarée comme analogique, 255 correspond à une valeur de 100 % en PWM
```

```

6     delay(15); // ajoute une temporisation de 15 ms entre deux incréments de
niveau de luminosité afin que le changement reste perceptible
7     }
8     for (int a=255; a>=0;a--) // cette boucle décrémente la valeur de PWM jusqu'à l'
extinction (valeur = 0)
9     {
10        analogWrite(ledpin,a);
11        delay(15);
12    }
13    delay(100); // temps de pause entre deux cycles d'augmentation et réduction de
la luminosité
14 }
15

```

Une LED est un composant polarisé qui ne laisse passer le courant que dans un sens, comme le ferait une diode. Le courant circule, et donc la LED s'allume, quand sa patte la plus longue est reliée au 5V (ou au 3.3V) et quand la plus courte est reliée au GND sur une carte Arduino :



### Question n°1

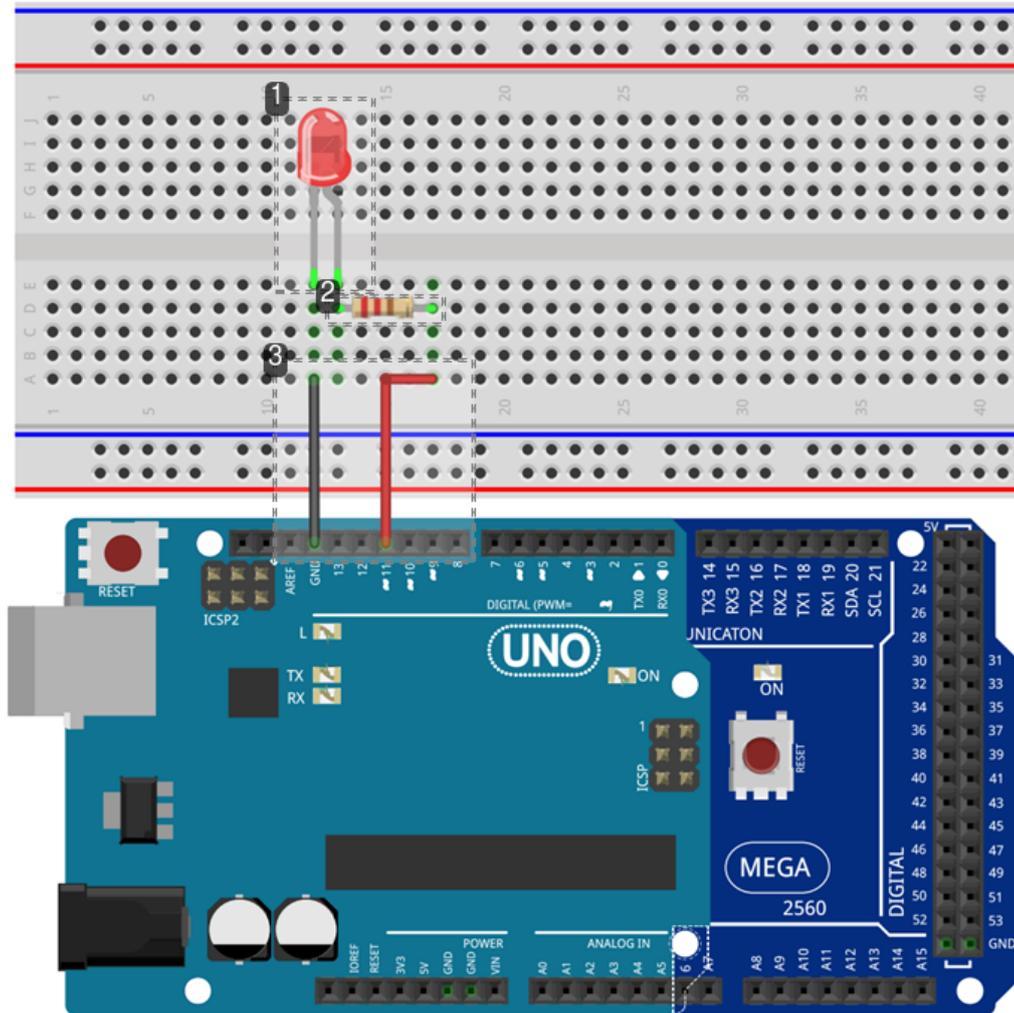
Comment fonctionne une boucle FOR en langage Arduino ?

### Question n°2

Pourquoi imposer un délai de 15 ms entre deux exécutions successives de boucles ? Il faut savoir que l'œil humain peut percevoir des changements allant jusqu'à 2 000 images par seconde.

## 1.4.5. Réalisation du montage contrôleur de LED

Réalisez le montage ci-dessous, puis copiez/collez les 3 blocs du code présentés auparavant dans l'IDE en respectant l'ordre indiqué et téléversez-le dans la carte Arduino.



### 1 LED



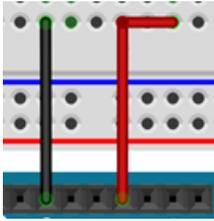
Diode électroluminescente

### 2 Résistor 220 ohm



Permet de baisser la tension dans un circuit.

### 3 Fils Dupont



Permet de relier les composants entre eux.

### Platine Labdec

~ Breadboard

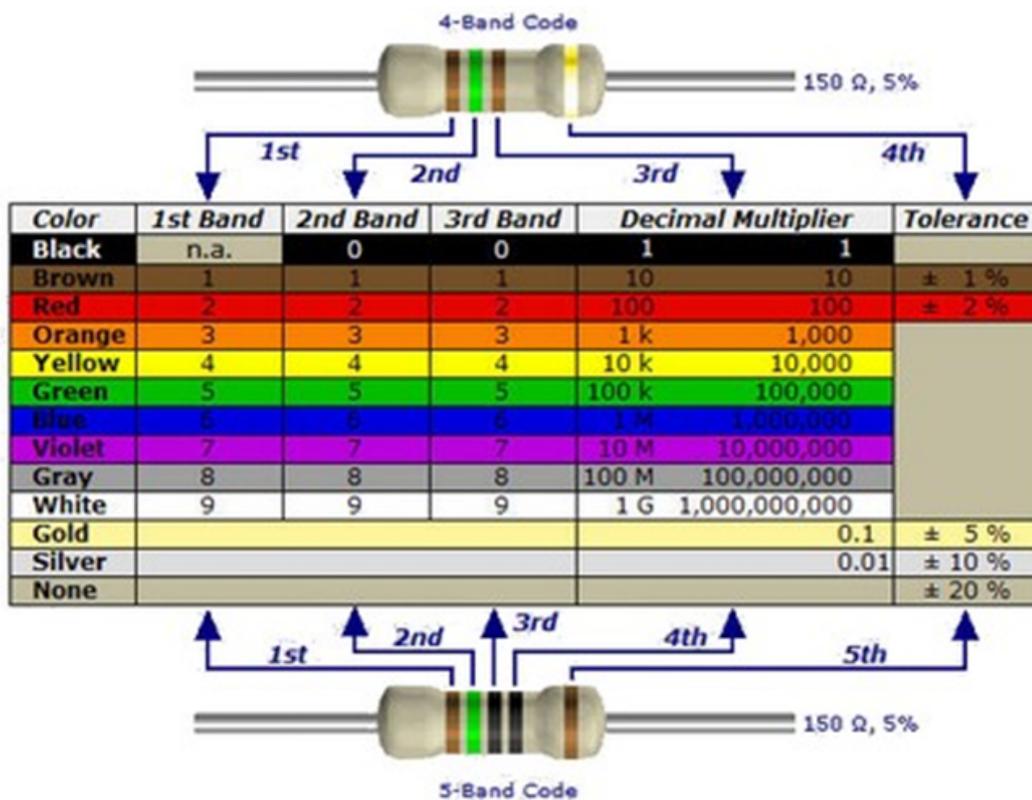
La « breadboard » ou « platine Labdec » simplifie le montage de composants. Cette platine est très utilisée pour le prototypage électronique afin d'éviter d'avoir recours à des cartes électroniques et des soudures.

Tous les connecteurs positifs et négatifs latéraux (entourés des lignes rouges et bleues) sont reliés entre eux. Tous les connecteurs des lignes A à E et des lignes F à J sont reliés entre eux (mais il n'y a aucune liaison entre A-E et F-J).

Le schéma ci-dessous illustre les liaisons internes de la platine.

La résistance d'un résistor et la tolérance associée à cette valeur sont clairement identifiables à travers un code couleur normalisé.

Le code couleur des résistances est présenté ci-dessous.



Les lignes colorées précisent la valeur de la résistance et la tolérance associée

## Résistance

Une résistance est un composant électronique ou électrique dont la principale caractéristique est d'opposer une plus ou moins grande résistance (*mesurée en ohms*) à la circulation du courant électrique.

La résistance électronique est l'un des composants primordiale dans le domaine de l'électricité.

### Information technique

Lorsqu'une résistance est placée dans un circuit électrique on obtient deux types de phénomènes :

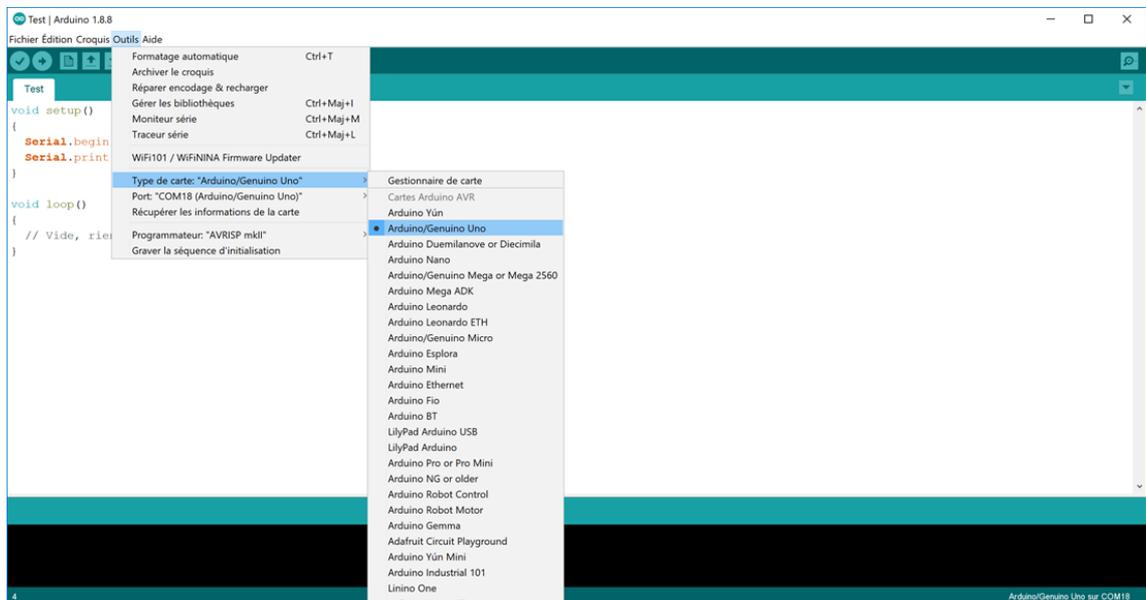
– La résistance a une influence sur l'intensité du courant continu, plus la résistance est élevée et plus l'intensité est faible.

Cette influence sur le courant est mise à profit dans de nombreux appareils électriques et électronique pour modifier l'intensité du courant. Elles permettent de protéger les dipôles qui ne supportent pas des intensités trop élevées. Dans ce cas les résistance servent à réguler l'intensité du courant continu.

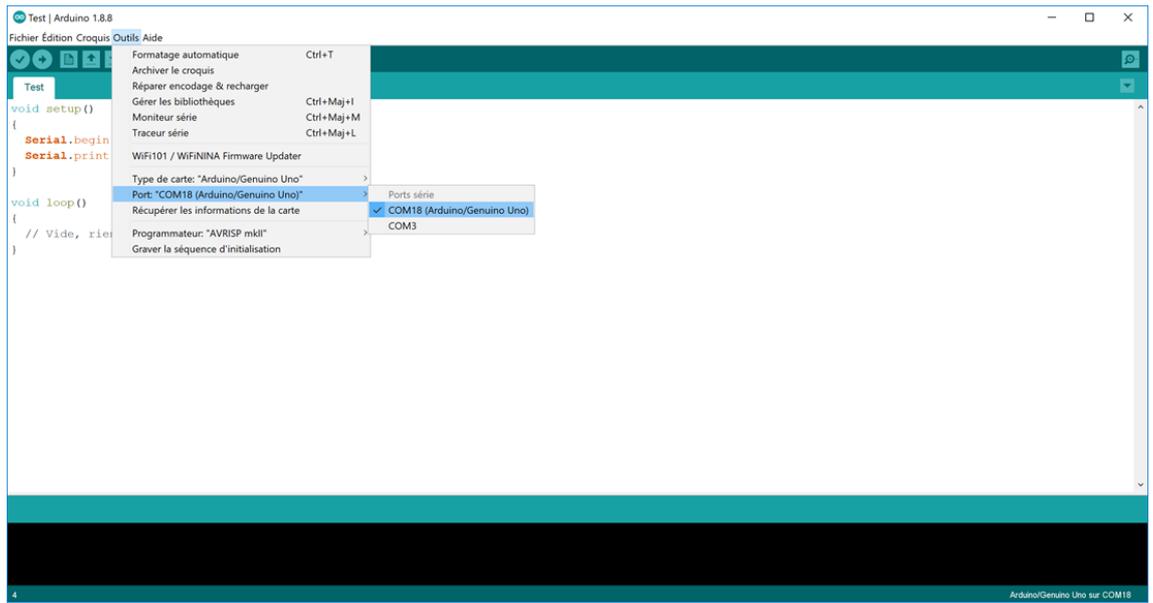
– La résistance parcourue par un courant électrique donne naissance à l'effet joule<sup>[p.241]</sup>.

## Procédure : Téléverser un programme Arduino par le port USB

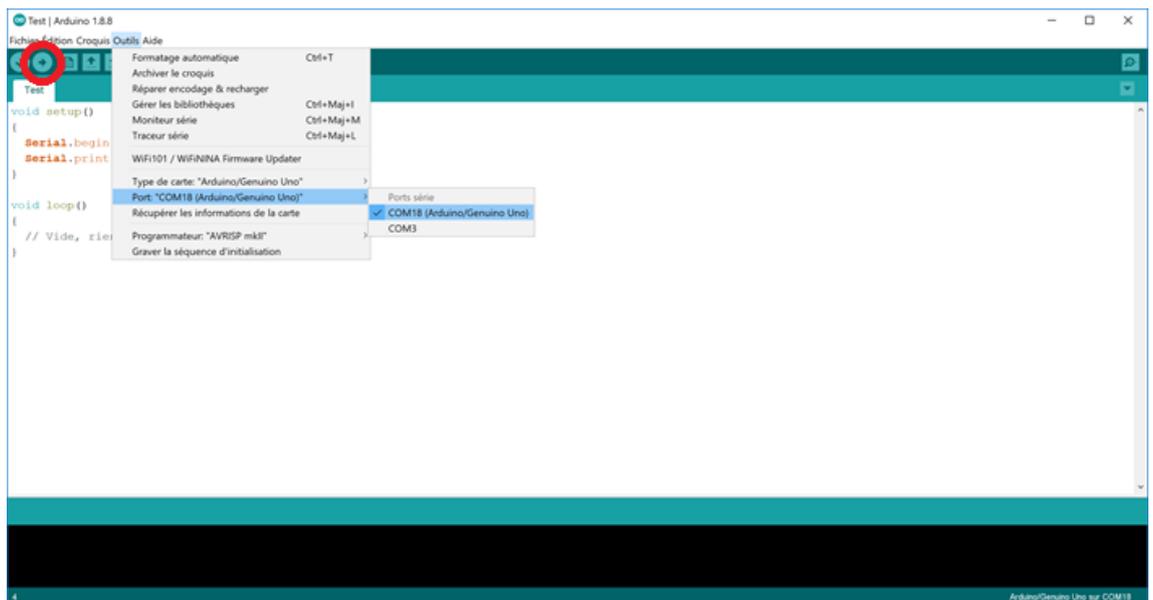
**1** S'assurer que  Outils / Type de carte est bien configuré sur  Arduino /Genuino Uno.



**2** S'assurer que le port COM est bien celui qui est connecté à votre carte Arduino. Si la carte est bien détectée, son nom est généralement écrit à côté du port COM associé.



**3** Cliquer sur la flèche dirigée vers la droite.



Si le texte **Téléversement terminé** n'apparaît pas dans le bandeau du bas, c'est qu'il y a eu un problème de communication.

### Question n°1

Réalisez le montage, téléversez le programme puis constatez son fonctionnement.

### Question n°2

A quoi sert la résistance de 220 Ohm branché sur l'anode (+) de la LED (pate la plus longue) ?

### Question n°3

Pourquoi appelle-t-on anode la partie reliée au 5V alors que sur une batterie, la borne + s'appelle une cathode ?

### Question n°4

Modifiez le programme afin que la LED s'allume 1s, s'éteigne 1s, s'allume 3s et s'éteigne 1s et constatez le fonctionnement.

## 1.5. Contrôleur de LED RGB

### 1.5.1. Introduction

#### Objectifs pédagogiques

L'objectif de cet exercice est de piloter une LED RGB de façon à faire varier sa couleur.

Pour réaliser ce montage, il faudra vous munir des composants suivants :

- une carte arduino
- un câble USB
- une LED RGB
- 3 résistors de 220 Ohms
- des fils Dupont Mâle/Mâle
- une platine de prototypage rapide à trou de type Labdec

### 1.5.2. Principe du RVB ou RGB

#### RVB

≈ RGB

Rouge, vert, bleu, abrégé en RVB ou en RGB (de l'anglais « red, green, blue ») est un système de codage informatique des couleurs, le plus proche du matériel.

Les écrans d'ordinateurs reconstituent une couleur par synthèse additive à partir de trois couleurs primaires : rouge, vert et bleu, formant sur l'écran une mosaïque trop petite pour être aperçue.

Le codage RVB indique une valeur pour chacune de ces couleurs primaires.

Des paramètres plus intuitifs tels que la teinte, la saturation et la luminosité exigent du système informatique qu'il calcule ces valeurs.

Pour chacune des couleurs primaires, la valeur s'exprime dans un intervalle compris entre 0 et le

maximum, qui est soit 1 ou 100 %, exprimé en codage informatique par 255 ou 0xFF (hexadécimal). Les trois primaires en quantité égale codent du gris, et au maximum donnent du blanc.

### Exemple Codage d'un ton saumon

Le code RVB indique rouge = 100 %, vert = 80 %, bleu = 60 %.

## 1.5.3. Montage et exécution du contrôleur de LED RGB

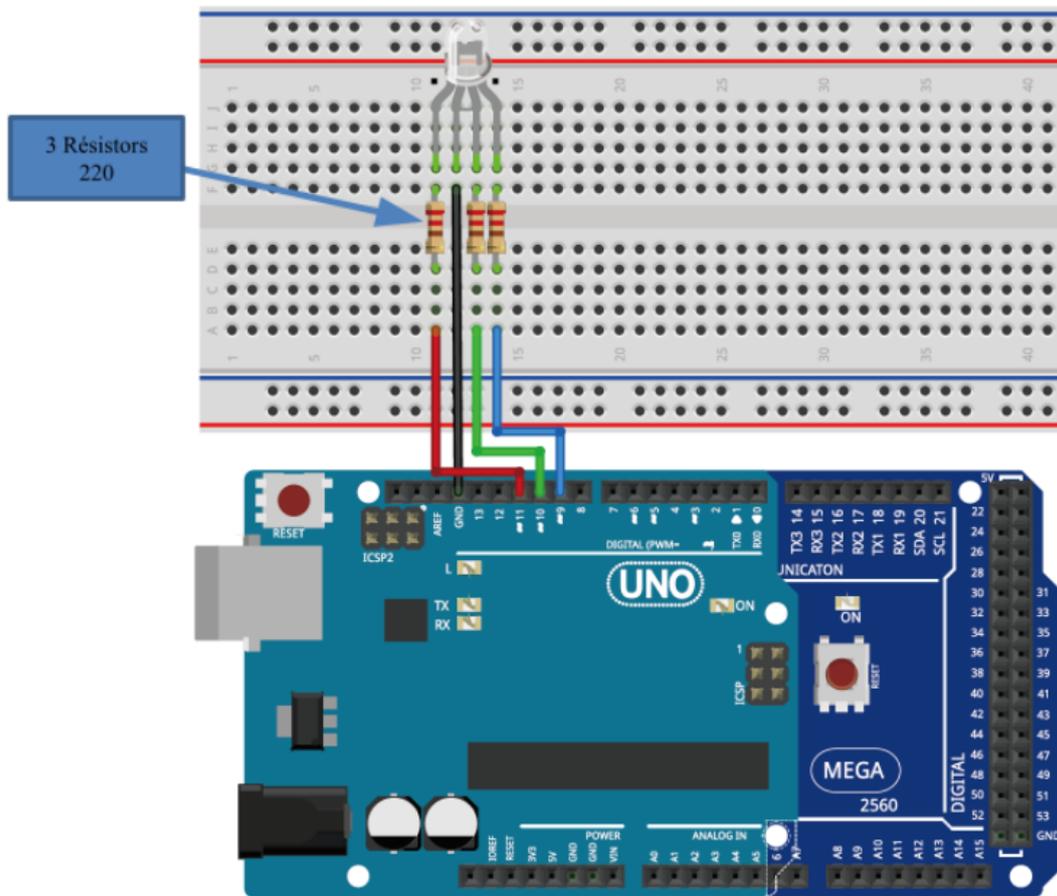
Le programme ci-dessous permet de contrôler une LED RGB :

```

1 int redPin = 11; // Rouge
2 int greenPin = 10; // Vert
3 int bluePin = 9; // Bleu
4 void setup()
5 {
6   pinMode(redPin, OUTPUT);
7   pinMode(greenPin, OUTPUT);
8   pinMode(bluePin, OUTPUT);
9 }
10 void loop()
11 {
12   // Couleurs brutes
13   color(255, 0, 0); // Active la couleur rouge de la LED RGB
14   delay(1000);
15   color(0,255, 0); // Active la couleur verte de la LED RGB
16   delay(1000);
17   color(0, 0, 255); // Active la couleur bleue de la LED RGB
18   delay(1000);
19
20   // Couleurs variées
21   color(255,255,0); // Active la couleur jaune de la LED RGB
22   delay(1000);
23   color(255,255,255); // Active la couleur blanche de la LED RGB
24   delay(1000);
25   color(128,0,255); // Active la couleur mauve de la LED RGB
26   delay(1000);
27   color(0,0,0); // Eteint la LED RGB
28   delay(1000);
29 }
30
31 void color (unsigned char red, unsigned char green, unsigned char blue)
32 {
33   analogWrite(redPin, red); // sortie MLI
34   analogWrite(greenPin, green); // sortie MLI
35   analogWrite(bluePin, blue); // sortie MLI
36 }
37

```

Le montage à réaliser est le suivant :



Correspondance des différentes pattes d'une LED RGB

## Lumen

En physique, un lumen est la quantité de lumière interceptée par 1 m<sup>2</sup> de surface interne d'une sphère creuse de 1 m de rayon, au centre de laquelle on a placé une bougie. Pour définir l'éclairement d'un objet, on utilise le lux qui correspond à un flux d'un lumen tombant sur une surface de 1 m<sup>2</sup> de l'objet.

## Question n°1

Réalisez le montage, téléversez le code et vérifiez son fonctionnement.

### Question n°2

L'intensité lumineuse de la couleur rouge est plus faible que les autres, pour quelle raison ? Que faudrait-il changer pour augmenter sa production de lumens ?

### Question n°3

Modifiez le programme de façon à produire d'autres couleurs.

## 1.6. Pilotage de LED par des interrupteurs

### 1.6.1. Introduction

#### Objectifs pédagogiques

L'objectif est d'apprendre à utiliser des interrupteurs pour piloter le programme, dans le cas présent pour piloter l'allumage d'une LED.

Pour réaliser ce montage, il faudra vous munir des composants suivants :

- une carte arduino
- un câble USB
- une LED RGB
- 1 résistor de 220 Ohms
- 2 interrupteurs
- des fils Dupont Mâle/Mâle
- une platine de prototypage rapide à trou de type Labdec

### 1.6.2. Montage et pilotage de LED par interrupteurs

Le programme ci-dessous permet de contrôler une LED en utilisant deux interrupteurs.

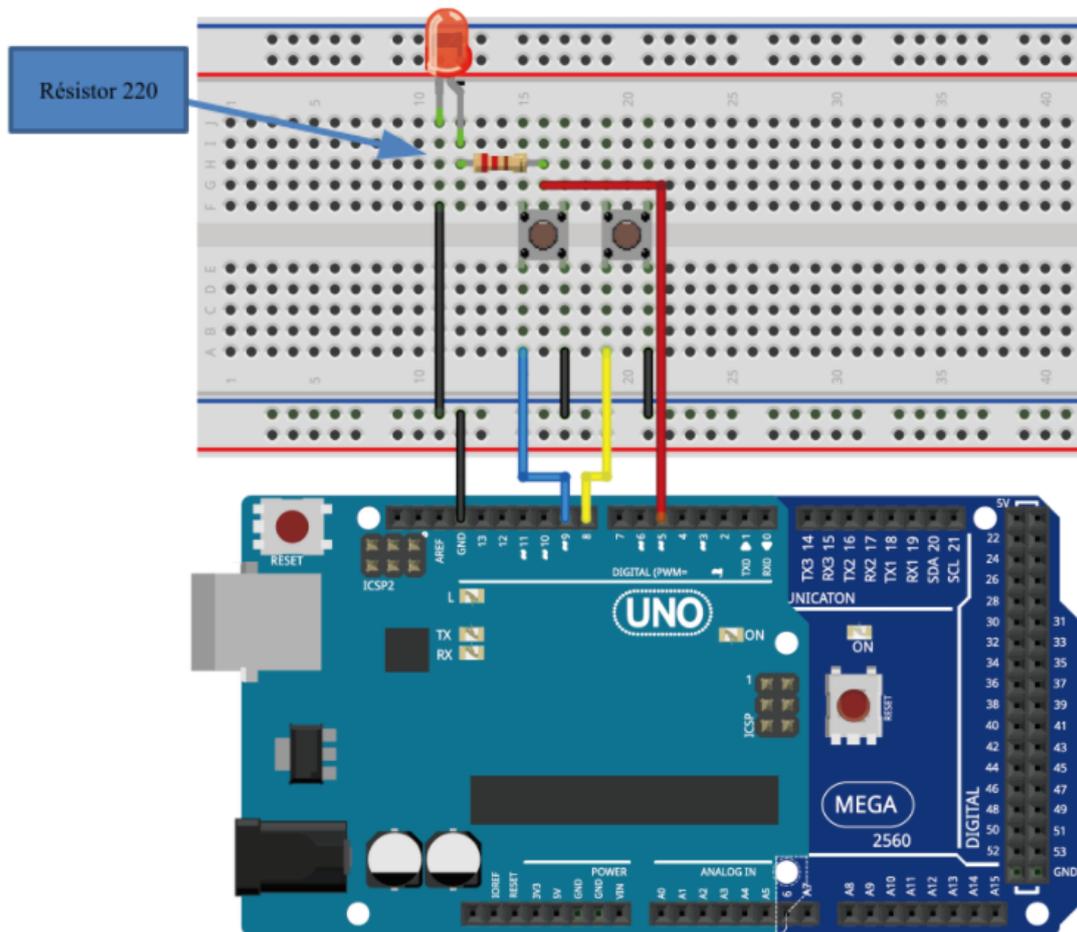
```
1 int ledPin = 5;
2 int buttonApin = 9;
3 int buttonBpin = 8;
4
5 byte leds = 0;
6
7 void setup()
8 {
9   pinMode(ledPin, OUTPUT);
10  pinMode(buttonApin, INPUT_PULLUP);
```

```

11  pinMode(buttonBpin, INPUT_PULLUP);
12 }
13
14 void loop()
15 {
16  if (digitalRead(buttonApin) == LOW)
17  {
18    digitalWrite(ledPin, HIGH);
19  }
20  if (digitalRead(buttonBpin) == LOW)
21  {
22    digitalWrite(ledPin, LOW);
23  }
24 }

```

Le montage à réaliser est le suivant :



### Question n°1

Pourquoi faut-il préciser que les entrées 8 et 9 sont en INPUT\_PULLUP ?

### Question n°2

Expliquez la différence entre un montage PULL-UP et un PULL-DOWN.

### Question n°3

Réalisez le montage, téléversez le code puis constatez le fonctionnement.

## Question n°4

Modifier le programme pour que la LED s'allume uniquement quand les deux interrupteurs sont activés puis testez.

## 1.7. Le moniteur série

### 1.7.1. Introduction

La liaison série de type RS232 peut être utilisée pour transmettre des informations depuis la carte vers l'ordinateur ou inversement. Ces informations peuvent être visualisées via le moniteur série.

Dans le cas de la carte Arduino Uno, elle est "simulée" et passe par la liaison USB.

Elle permet de transmettre une information sur 8 bits de manière asynchrone.

Sur les systèmes d'exploitation MS-DOS et Windows, les ports RS-232 sont désignés par les noms COM1, COM2... qui sont utilisés pour communiquer avec les cartes Arduino.

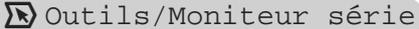
Le moniteur série a plusieurs usages intéressants, il permet par exemple d'afficher des messages de débogage depuis la carte vers l'ordinateur ou de créer une Interface Homme Machine entre l'utilisateur, le programme et la carte Arduino pour piloter son fonctionnement au moyen de variables entrées au clavier.

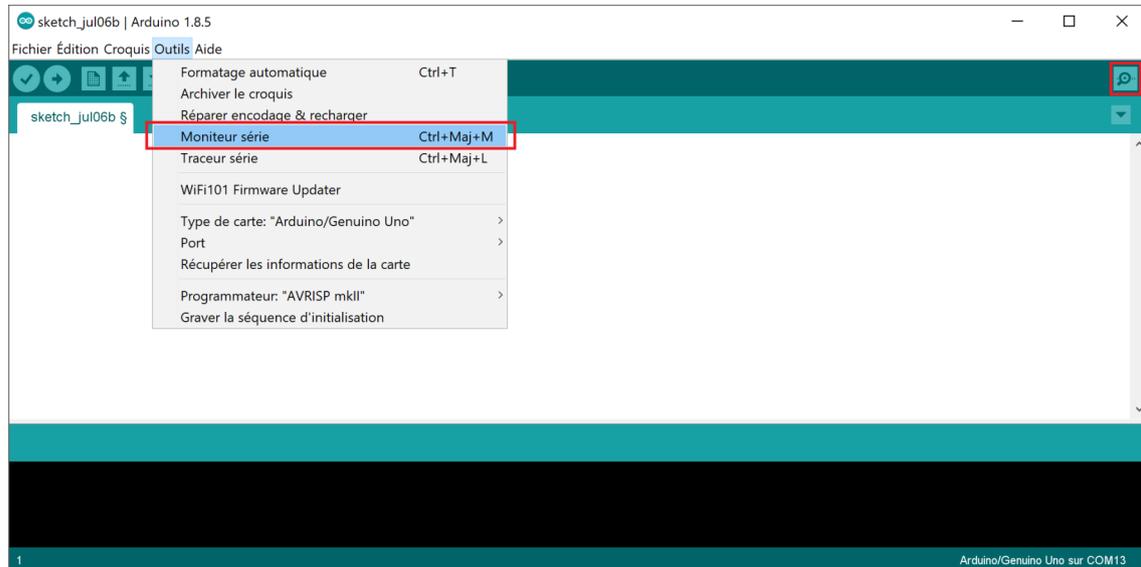
### 1.7.2. Afficher dans le moniteur série le caractère entré par l'utilisateur

Le programme ci-dessous permet de faire communiquer le programme et l'utilisateur :

```

1 void setup() {
2   Serial.begin(9600); // Ouvre le port série et fixe le débit de données à 9600
   bauds
3 }
4
5 void loop() {
6   if (Serial.available() > 0) { // Vérifie le fonctionnement de la liaison série
7     char c = Serial.read(); // Attribue à la variable c le caractère entré au
   clavier par l'utilisateur du moniteur
8     Serial.print("Caractere reçu : ");
9     Serial.println(c); // Affiche dans le moniteur le caractère entré par
   l'utilisateur
10  }
11 }
```

Le moniteur série peut être affiché dans Windows depuis le menu  ou en appuyant sur  ou en cliquant sur l'icone  en haut à droite.



La fenêtre du moniteur série est la suivante :



### ⚠ Attention

Veillez à bien régler le nombre de baud du moniteur série à celui que vous avez fixé dans le setup du programme, dans le cas contraire les informations affichées seront illisibles. Ce réglage est disponible en bas à droite de la fenêtre. Dans le cas présent, c'est 9600 bauds.

Dans cette fenêtre il est possible de cocher le défilement automatique pour que les valeurs affichées s'actualisent au cours du temps

### Question n°1

Qu'est-ce-que le nombre de bauds ?

### Question n°2

Dans le programme ci-dessus, combien de temps s'écoule entre deux changements d'état ?

## Question n°3

Copiez le programme dans l'IDE Arduino et testez-le.

### 1.7.3. Afficher dans le moniteur série des variables

Le programme ci-dessous permet de créer un menu à choix multiple :

```

1 int variable=0;
2
3 void setup() {
4   Serial.begin(9600); // Ouvre le port série et fixe le débit de données à 9600
   bauds
5   menu();
6 }
7
8 void loop() {
9   if (Serial.available() > 0) { // Vérifie le fonctionnement de la liaison série
10    char c = Serial.read();
11    if (c == '1') {
12      variable++;
13      menu();
14    }
15    else if (c == '2') {
16      variable--;
17      menu();
18    }
19    else if ((c!='1') && (c!='2') && (c!='\n')){
20      Serial.println("Pas la bonne touche !"); Serial.println();
21    }
22  }
23 }
24
25 void menu(){
26   Serial.print("La variable vaut : ");
27   Serial.println(variable); Serial.println();
28   Serial.println("Menu à choix multiple");
29   Serial.println("1: +");
30   Serial.println("2: -");
31 }

```

Dans ce programme, le menu est appelé sous la forme d'une fonction.

Une fonction s'écrit s'ainsi : variablesRenvoyées NomDeLaFonction(variablesReçues).

Dans le cas présent :

- void signifie que la fonction ne renvoie pas de valeur de variable
- menu est le nom de la fonction
- ( ) que la fonction ne reçoit aucune valeur

## Question n°1

Copiez le programme dans l'IDE Arduino et testez-le.

## Question n°2

Dans le code ci-dessus, le programme vérifie que la réponse de l'utilisateur n'est pas '\n'. A quoi sert cette vérification ?

### Question n°3

Rajoutez une troisième option dans le menu pour remettre à 0 la variable.

## 1.8. Multimètre à 2 voies

---

### 1.8.1. Introduction

---

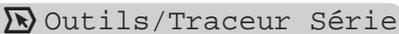
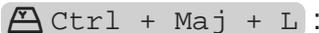
L'objectif de cette ressource est de mesurer deux variations de tension simultanées et de tracer leurs courbes en temps réel dans l'IDE.

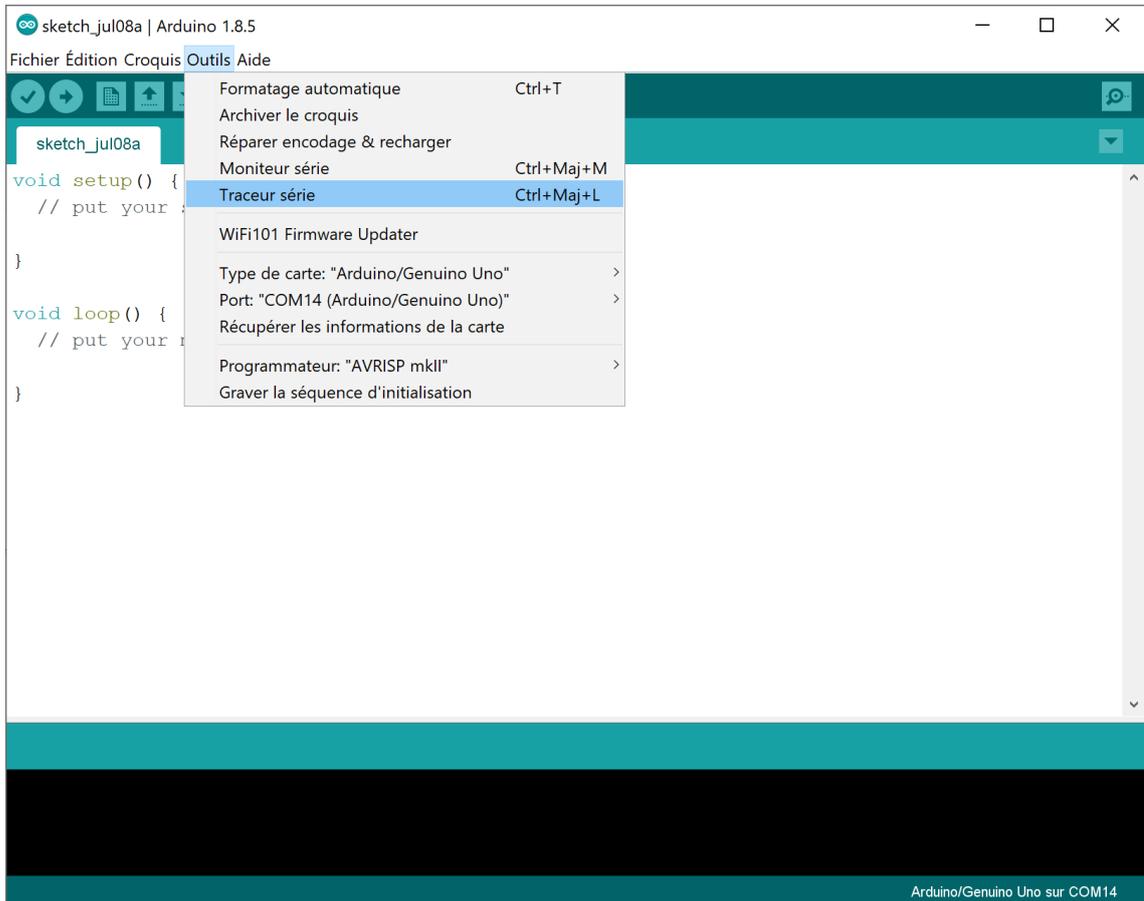
Pour réaliser ce montage, il faudra vous munir des composants suivants :

- une carte Arduino
- un câble USB
- deux potentiomètres
- des fils Dupont Mâle/Mâle
- une platine de prototypage rapide à trou de type Labdec

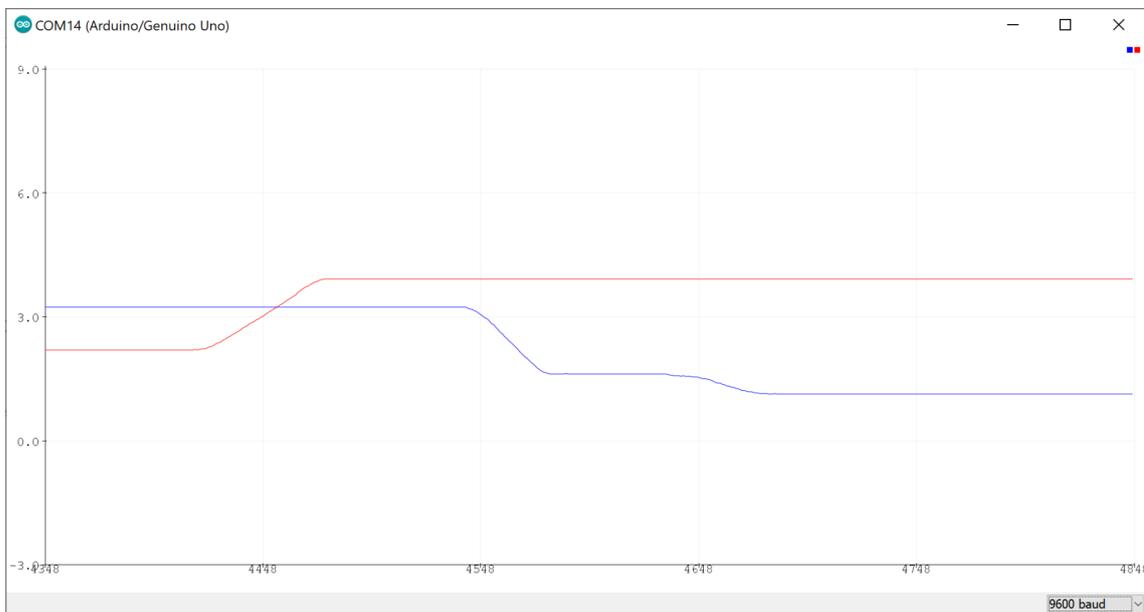
### 1.8.2. Montage, mesure de variations de tensions et traçage de courbes

---

L'IDE Arduino dispose d'une fonction pour tracer une ou plusieurs courbes simultanément, au moyen du traceur série disponible dans le menu  ou en appuyant simultanément sur .



Le traceur série se présente ainsi :



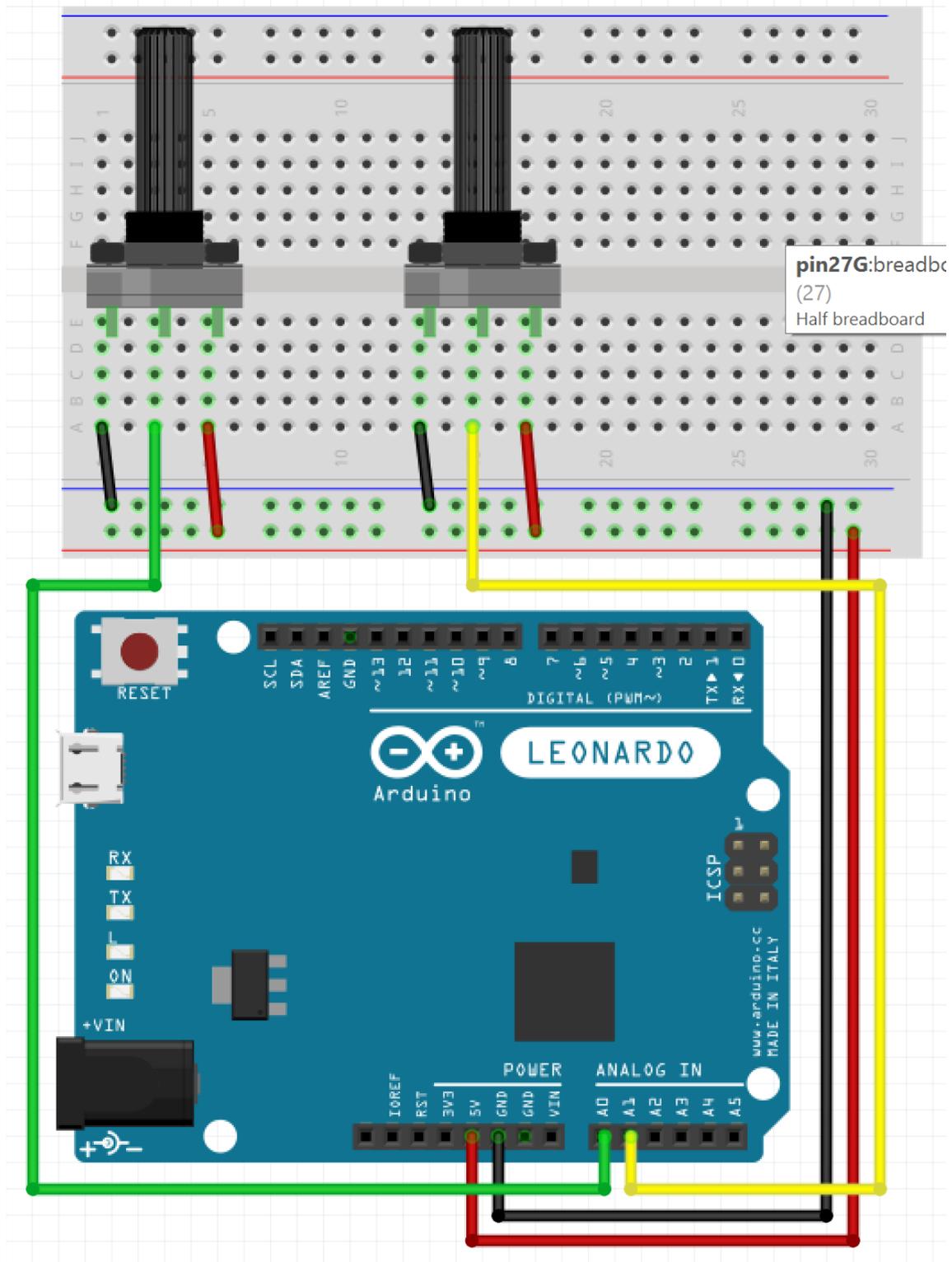
Le code ci-dessous permet de créer un multimètre à 2 voies :

```

1 // préciser les connecteurs reliés aux potentiomètres, le premier est branché sur
  le connecteur analogique 0 et le deuxième sur le 1
2 int potentiometre1 = 0;
3 int potentiometre2 = 1;
4
5 // déclaration des variables utilisées pour stocker la valeur des tensions mesurées
  analogiquement sur les entrées de la carte Arduino
6 int valeur1;
    
```

```
7 int valeur2;
8
9 // déclaration des variables qui vont accueillir les valeurs de tension sous forme
  de nombres réels
10 float tension1;
11 float tension2;
12
13 void setup() {
14   Serial.begin(9600);
15   while(!Serial);
16 }
17
18 void loop() {
19   valeur1=analogRead(potentiometre1);
20   valeur2=analogRead(potentiometre2);
21
22   // transformation des valeurs entières (5v=1023) en valeurs réelles en volts
23   tension1 = valeur1 * 5.0 / 1023;
24   tension2 = valeur2 * 5.0 / 1023;
25
26   // Affichage de la valeur mesure sur le traceur série
27   Serial.print(tension1);
28   Serial.print(",");
29   Serial.print(tension2);
30   Serial.println();
31 }
```

Le montage associé est le suivant :



### Question n°1

Quelle est l'instruction pour déclarer qu'un connecteur est utilisé comme entrée ?

### Question n°2

Réalisez le montage puis téléversez le code et observez dans le traceur série la variation de tension en tournant les potentiomètres.

## Question n°3

Comment fonctionne un potentiomètre ?

# 1.9. Buzzer actif

## 1.9.1. Introduction

### Objectifs pédagogiques

L'objectif de cet exercice est de piloter un buzzer en utilisant un transistor.

Pour réaliser ce montage, il faudra vous munir des composants suivants :

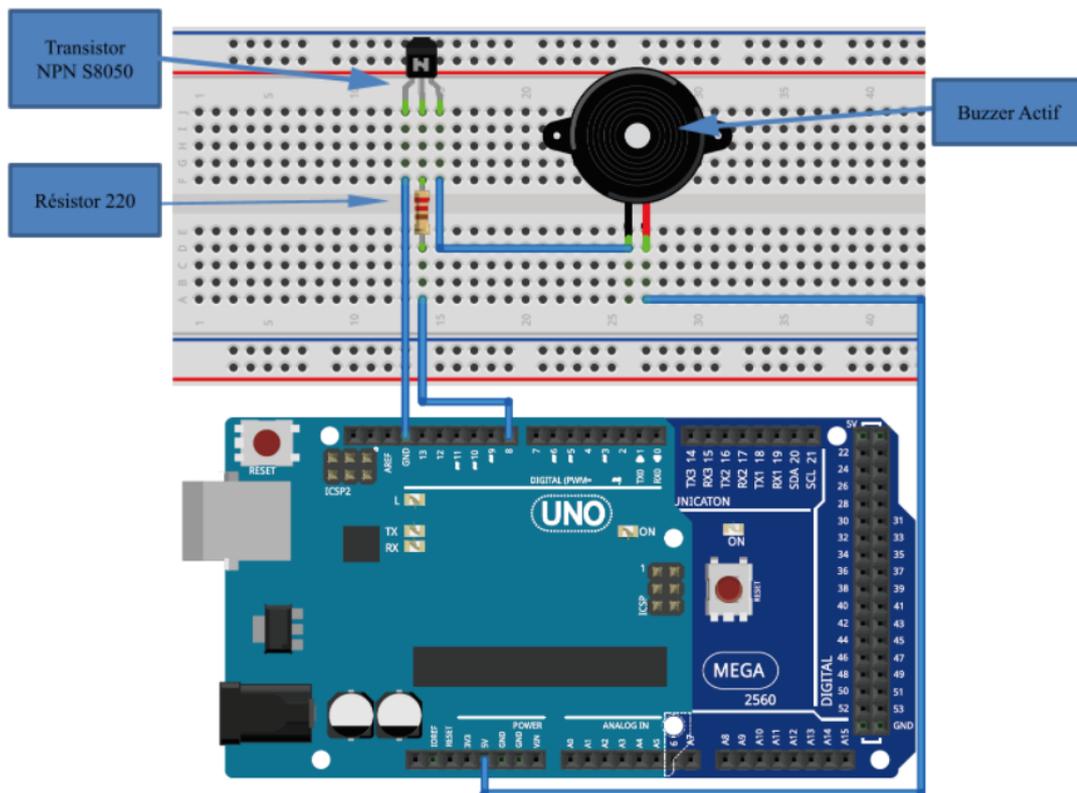
- une carte Arduino
- un câble USB
- un transistor NPN S8050
- un buzzer actif HYDZ
- un résistor 220 Ohms
- des fils Dupont Mâle/Mâle
- une platine de prototypage rapide à trou de type Labdec

## 1.9.2. Montage et pilotage d'un buzzer actif

Le programme ci-dessous permet de contrôler un buzzer actif :

```
1 int buzzerPin=8;
2 void setup()
3 {
4     pinMode(buzzerPin,OUTPUT);
5 }
6 void loop()
7 {
8     digitalWrite(buzzerPin,HIGH); //Force le connecteur à la valeur HIGH = 5 v
9     delay(2000);
10    digitalWrite(buzzerPin,LOW); //Force le connecteur à la valeur LOW = 0 v
11    delay(2000);
12 }
```

Le montage à réaliser est le suivant :



### Question n°1

Quelle est la différence entre un buzzer ACTIF et un buzzer PASSIF ?

### Question n°2

A quoi sert un transistor et comment fonctionne-t-il ?

### Question n°3

Quelles sont les différences entre les transistors NPN et PNP ?

### Question n°4

Réalisez le montage, téléversez le programme puis constatez son fonctionnement.

### Question n°5

Réalisez un montage et un programme permettant d'allumer le buzzer en appuyant sur un bouton, puis vérifiez son fonctionnement

## 1.10. Buzzer passif

### 1.10.1. Introduction

## Objectifs pédagogiques

L'objectif de cet exercice est d'apprendre à utiliser un buzzer passif pour réaliser une mélodie.

Pour réaliser ce montage, il faudra vous munir des composants suivants :

- une carte Arduino
- un câble USB
- un transistor NPN S8050
- un buzzer passif HX
- une LED
- deux résistors 220 Ohms
- des fils Dupont Mâle/Mâle
- une platine de prototypage rapide à trou de type Labdec

### 1.10.2. Montage et pilotage d'un buzzer passif

Le programme ci-dessous permet de contrôler un buzzer passif pour lui faire jouer une mélodie :

```

1 #define NTD0 -1 // Basse 1# 2# 3# 4# 5# 6# 7#
2 #define NTD1 294 // A 221 248 278 294 330 371 416
3 #define NTD2 330 // B 248 278 294 330 371 416 467
4 #define NTD3 350 // C 131 147 165 175 196 221 248
5 #define NTD4 393 // D 147 165 175 196 221 248 278
6 #define NTD5 441 // E 165 175 196 221 248 278 312
7 #define NTD6 495 // F 175 196 221 234 262 294 330
8 #define NTD7 556 // G 196 221 234 262 294 330 371
9
10 #define NTDL1 147 // Alto 1 2 3 4 5 6 7
11 #define NTDL2 165 // A 441 495 556 589 661 742 833
12 #define NTDL3 175 // B 495 556 624 661 742 833 935
13 #define NTDL4 196 // C 262 294 330 350 393 441 495
14 #define NTDL5 221 // D 294 330 350 393 441 495 556
15 #define NTDL6 248 // E 330 350 393 441 495 556 624
16 #define NTDL7 278 // F 350 393 441 495 556 624 661
17 // G 393 441 495 556 624 661 742
18 #define NTDH1 589
19 #define NTDH2 661 // Aigu 1# 2# 3# 4# 5# 6# 7#
20 #define NTDH3 700 // A 882 990 1112 1178 1322 1484
    1665
21 #define NTDH4 786 // B 990 1112 1178 1322 1484 1665 1869
22 #define NTDH5 882 // C 525 589 661 700 786 882 990
23 #define NTDH6 990 // D 589 661 700 786 882 990 1112
24 #define NTDH7 112 // E 661 700 786 882 990 1112 1248
25 // F 700 786 882 935 1049 1178 1322
26 #define WHOLE 1 // G 786 882 990 1049 1178 1322 1484
27 #define HALF 0.5
28 #define QUARTER 0.25
29 #define EIGHTH 0.25
30 #define SIXTEENTH 0.625

```

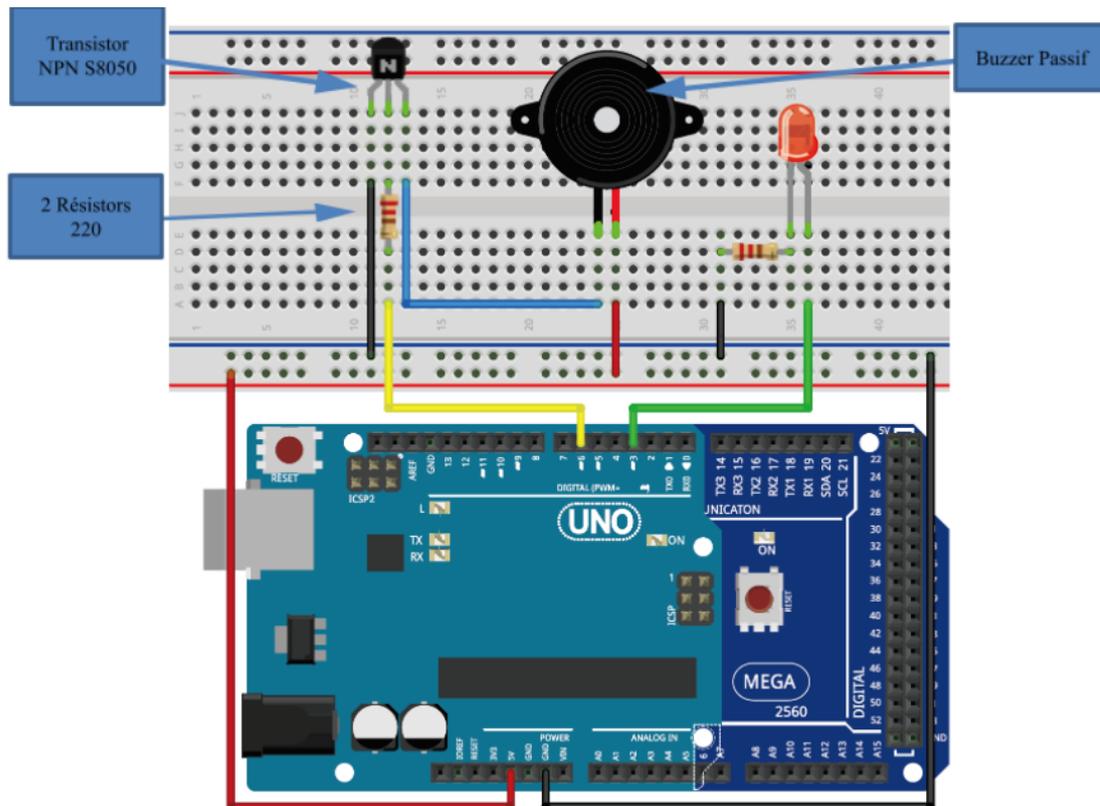
```

31 int tune[]=
32 {
33   NTD3,NTD3,NTD4,NTD5,
34   NTD5,NTD4,NTD3,NTD2,
35   NTD1,NTD1,NTD2,NTD3,
36   NTD3,NTD2,NTD2,
37   NTD3,NTD3,NTD4,NTD5,
38   NTD5,NTD4,NTD3,NTD2,
39   NTD1,NTD1,NTD2,NTD3,
40   NTD2,NTD1,NTD1,
41   NTD2,NTD2,NTD3,NTD1,
42   NTD2,NTD3,NTD4,NTD3,NTD1,
43   NTD2,NTD3,NTD4,NTD3,NTD2,
44   NTD1,NTD2,NTDL5,NTD0,
45   NTD3,NTD3,NTD4,NTD5,
46   NTD5,NTD4,NTD3,NTD4,NTD2,
47   NTD1,NTD1,NTD2,NTD3,
48   NTD2,NTD1,NTD1
49 };
50 float durt[]= // Durée de chaque note
51 {
52   1,1,1,1,
53   1,1,1,1,
54   1,1,1,1,
55   1+0.5,0.5,1+1,
56   1,1,1,1,
57   1,1,1,1,
58   1,1,1,1,
59   1+0.5,0.5,1+1,
60   1,1,1,1,
61   1,0.5,0.5,1,1,
62   1,0.5,0.5,1,1,
63   1,1,1,1,
64   1,1,1,1,
65   1,1,1,0.5,0.5,
66   1,1,1,1,
67   1+0.5,0.5,1+1,
68 };
69 int length;
70 int tonepin=6; // Connecteur du buzzer
71 int ledp=3;
72 void setup()
73 {
74   pinMode(tonepin,OUTPUT);
75   pinMode(ledp,OUTPUT);
76   length=sizeof(tune)/sizeof(tune[0]); // Calcule le nombre total de notes
77 }
78 void loop()
79 {
80   for(int x=0;x<length;x++)
81   {
82     tone(tonepin,tune[x]); // Active le buzzer
83     digitalWrite(ledp, HIGH);
84     delay(400*durt[x]);
85     digitalWrite(ledp, LOW);
86     delay(100*durt[x]);
87     noTone(tonepin); // Eteint le buzzer
88   }
89   delay(2000);

```

90 }

Le montage à réaliser est le suivant :



### Question n°1

Réalisez le montage, téléversez le code et constatez le résultat.

### Question n°2

Accélérez la vitesse à laquelle est jouée la mélodie et modifiez quelques notes et constatez le fonctionnement.

## 1.11. Tilt

### 1.11.1. Introduction

#### Objectifs pédagogiques

Réaliser un montage utilisant un capteur de type Tilt pour allumer une LED si une inclinaison donnée est atteinte.

Pour réaliser ce montage, il faudra vous munir des composants suivants :

- une carte Arduino
- un câble USB
- un Tilt
- une LED
- des fils Dupont Mâle/Mâle
- une platine de prototypage rapide à trou de type Labdec

## 1.11.2. Montage et allumage d'une LED par tilt

### Tilt

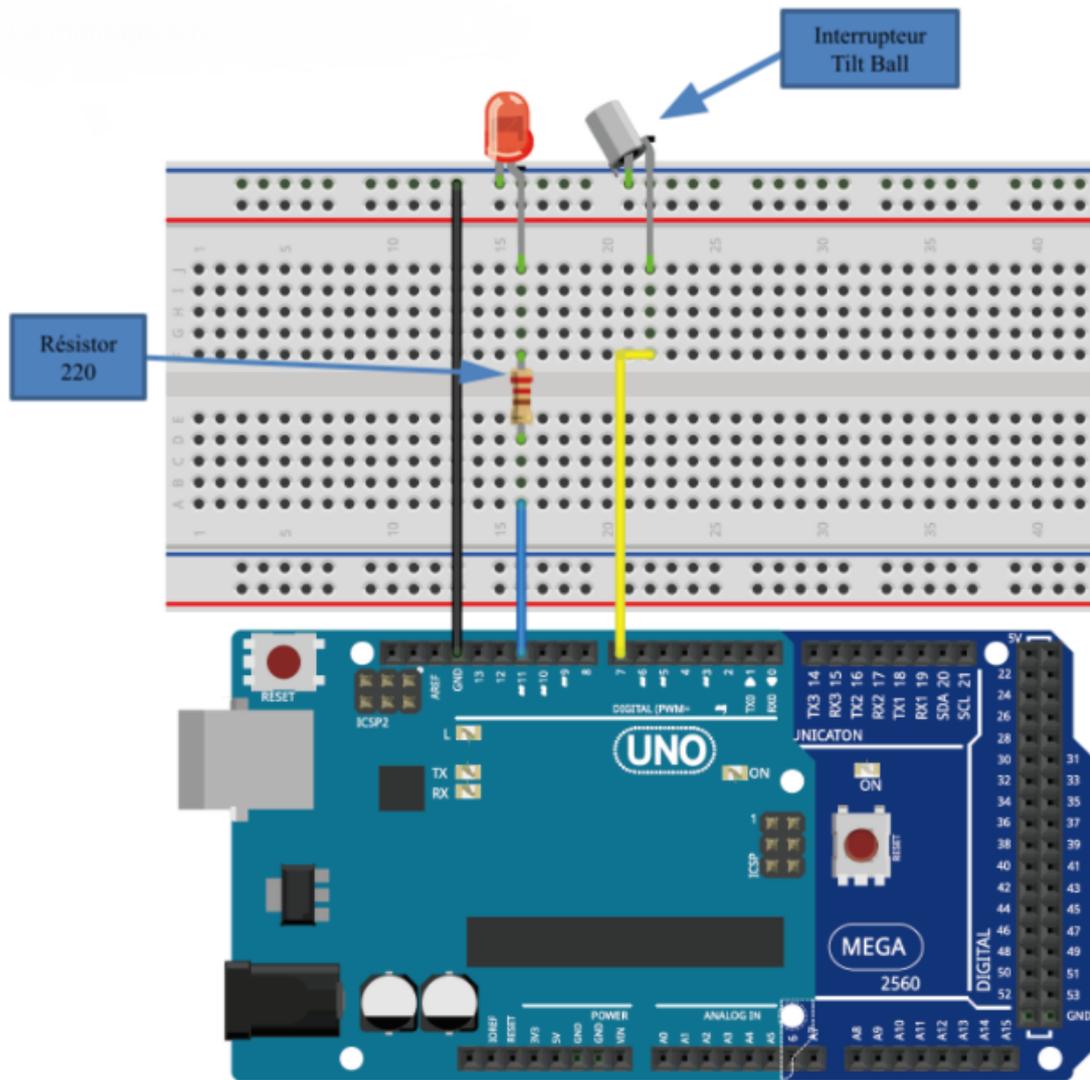
Un interrupteur Tilt permet de détecter l'orientation ou l'inclinaison d'un système. Il est souvent utilisé pour indiquer si un système (comme un véhicule agricole) dépasse sa plage d'inclinaison de fonctionnement ou pour détecter l'orientation d'un écran et, ainsi, modifier sa mise en page. Il ne donne pas autant d'information qu'un accéléromètre mais est plus robuste et ne nécessite pas de programme particulier pour être traité.

Le programme ci-dessous permet d'activer une LED par le biais d'un capteur d'accélération et/ou d'inclinaison appelé tilt :

```

1 int ledpin=11;
2 int tiltSwitchpin=7;
3 int val; // Déclaration d'une variable de type entière
4
5 void setup()
6 {
7   pinMode(ledpin,OUTPUT);
8   pinMode(tiltSwitchpin,INPUT_PULLUP);
9 }
10
11 void loop()
12 {
13   val=digitalRead(tiltSwitchpin);
14   if(val==LOW // Détecte si le tilt est inactif
15   { digitalWrite(ledpin,LOW);}
16   else // Si le tilt est déclenché, la LED est activée
17   { digitalWrite(ledpin,HIGH);}
18 }
```

Le montage à réaliser est le suivant :



### Question n°1

Comment est constitué l'intérieur du Tilt utilisé ?

### Question n°2

Réalisez le montage, téléversez le code et vérifiez son fonctionnement.

## 2. Programmer et développer des systèmes avec Arduino, perfectionnement

---

### 2.1. Introduction

---

L'objectif de cette ressource est de vous former à l'utilisation d'Arduino pour développer des systèmes interactifs qui intègrent des interfaces, des capteurs et des actionneurs. Le fonctionnement des effecteurs utilisés sera détaillé.

#### Déroutement

Durée : 12 heures

### 2.2. Contrôleur d'afficheur 7 segments

---

#### 2.2.1. Introduction

---

##### Objectifs pédagogiques

L'objectif de cette ressource est de réaliser un afficheur à 7 segments.



*Afficheur à LEDs 1 digit 7 segments*

Pour réaliser ce montage, il faudra vous munir des composants suivants :

- une carte Arduino

- un câble USB
- un afficheur à 7 segments
- un résistor de 220 Ohms
- des fils Dupont Mâle/Mâle
- une platine de prototypage rapide à trous de type Labdec

## 2.2.2. Afficheur à 7 segments

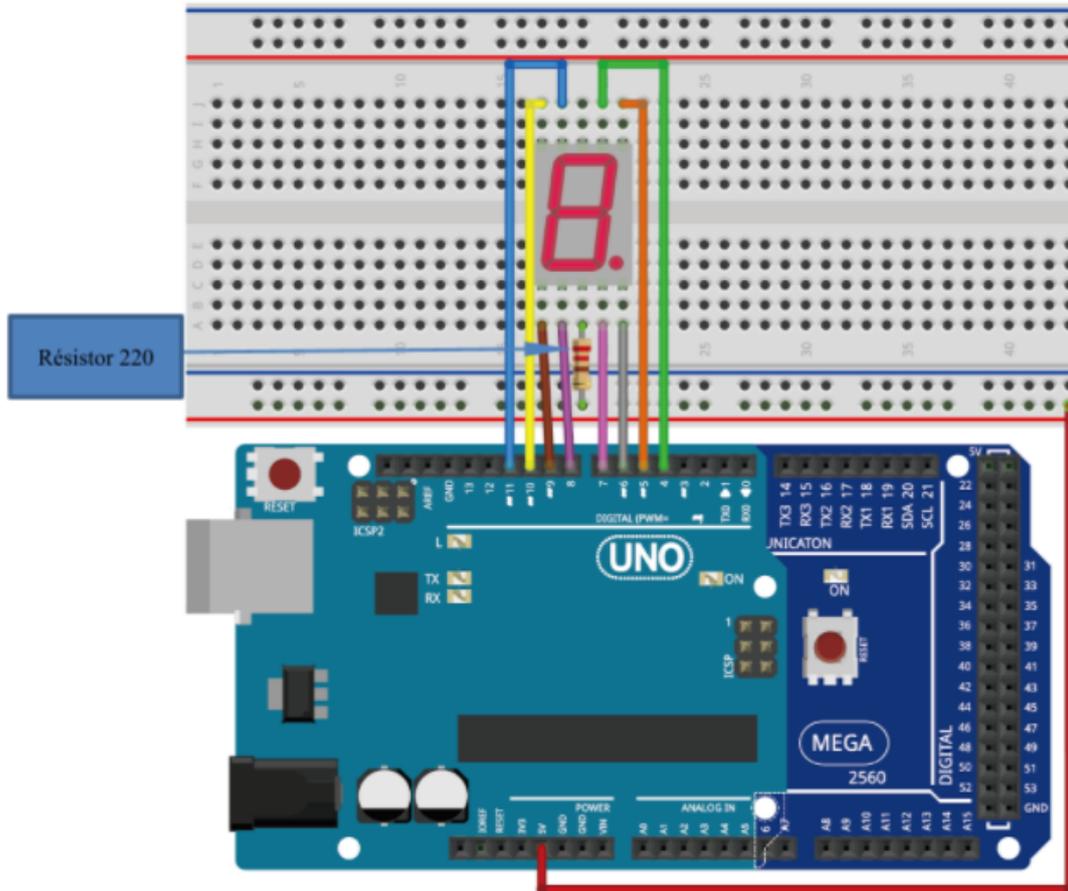
Le programme ci-dessous permet de contrôler un afficheur 7 segments pour lui faire afficher le chiffre 0 :

```

1 int a=4;
2 int b=5;
3 int c=7;
4 int d=8;
5 int e=9;
6 int f=11;
7 int g=10;
8 int dp=6;
9
10 void digital_0(void) // Fonction qui affiche '0' sur l'afficheur 7 segments
11 {
12   digitalWrite(a,LOW);
13   digitalWrite(b,LOW);
14   digitalWrite(c,LOW);
15   digitalWrite(d,LOW);
16   digitalWrite(e,LOW);
17   digitalWrite(f,LOW);
18   digitalWrite(g, HIGH);
19   digitalWrite(dp,HIGH);
20 }
21
22 void setup()
23 {
24   int i;
25   for(i=4;i<=11;i++)
26     pinMode(i,OUTPUT); // Règle les connecteurs 4 à 11 en mode « sortie »
27 }
28
29 void loop()
30 {
31   digital_0(); //Segment display digital 0
32 }

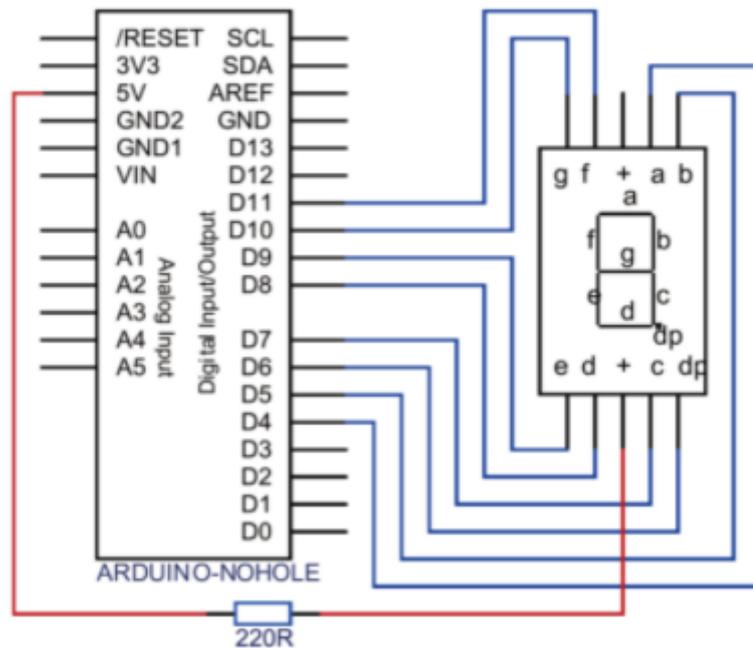
```

Le montage à réaliser est le suivant :



Le schéma ci-dessous présente la correspondance entre les broches de la carte Arduino et celles de l'afficheur à 7 segments.

## UNO R3/ Mega 2560



### Question n°1

Dans le programme ci-dessus, comment faut-il piloter les sorties de la carte Arduino pour allumer un segment ?

### Question n°2

Réalisez le montage, téléversez le code et constatez son fonctionnement.

### Question n°3

Modifiez le programme pour lui faire afficher des chiffres de 1 à 3 puis de 3 à 1 en opérant une pause d'une seconde entre 2 chiffres puis constatez le fonctionnement.

## 2.3. Utilisation d'un registre pour contrôler un afficheur 4 digits à 7 segments

### 2.3.1. Introduction

#### Objectifs pédagogiques

L'objectif de cette ressource est de piloter un afficheur à 7 segments à l'aide d'un registre à décalage.

Pour réaliser ce montage, il faudra vous munir des composants suivants :

- une carte Arduino
- un câble USB
- un afficheurs à 4 digits de 7 segments SMA420564L
- un registre à décalage de 8 bits 74HC595
- 4 résistors de 220 Ohms
- des fils Dupont Mâle/Mâle
- une platine de prototypage rapide à trous de type Labdec

## 2.3.2. Pilotage d'un afficheur 7 segments à l'aide d'un registre à décalage.

Le programme ci-dessous permet d'afficher les caractères 1 à 9 puis de a à f sur chaque afficheur en changeant d'afficheur toutes les 0.5 s :

```

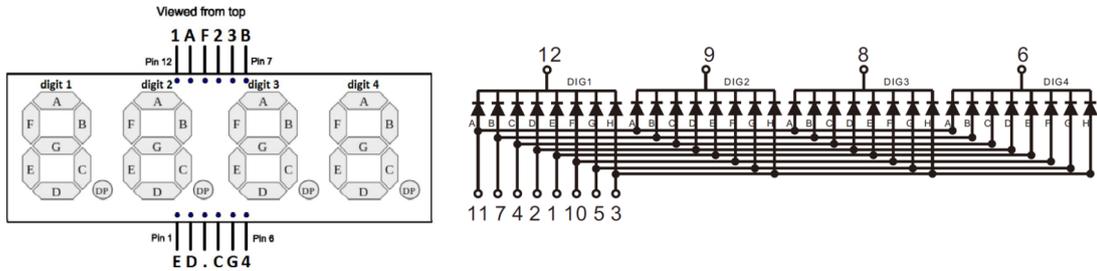
1 // broche ST_CP du registre, quand elle passe à 1 (5v) : copie les valeurs
  présentes dans le registre vers les sorties du registres de Q0 à Q7
2 int latch=11;
3 //broche SH_CP du registre, quand elle passe à 1 (5v) : décalce toutes les valeurs
  des bits vers la droite et copie la valeur de DS dans le bit de gauche du regsitre
4 int clock=12;
5 //broche DS du registre, accueille la valeur du bit : 0 (0V) ou 1 (5V) qui sera
  recopiée lors de la prochaine activation de la broche SH_CP
6 int data=8;
7
8 // tableau qui contient des octets qui seront traités par le registre à décalage,
  "unsigned char" peut être remplacé par "byte"
9 unsigned char table[]= {0b11111111,0b01100000,0b11011010,0b11110010};
10
11 void setup()
12 {
13   pinMode(latch,OUTPUT);
14   pinMode(clock,OUTPUT);
15   pinMode(data,OUTPUT);
16 }
17
18 void Display(unsigned char num)
19 {
20   digitalWrite(latch,LOW);
21   // la fonction shiftOut écrit vers la broche data (=8), les bits d'une variable
  un par un et active la broche clock (=12) à chaque envoi de bit
22   // MSBFIRST or LSBFIRST : (Most Significant Bit First = the leftmost or Least
  Significant Bit First = rightmost)
23   // ici c'est LSBFIRST qui est choisi, soit le bit le plus à droite en premier
24   // l'octet lu bit par bit est celui du tableau table[num]
25   shiftOut (data,clock,LSBFIRST,table[num]);
26   digitalWrite(latch,HIGH);
27 }
28
29 void loop()
30 {
31   Display(0); //active tous les segments des 4 digits
32   delay(1000);
33   Display(1); //active 2 segments pour afficher le chiffre 1 sur chaque digit

```

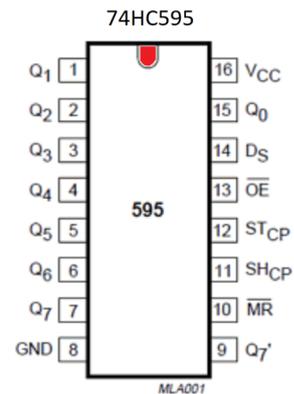
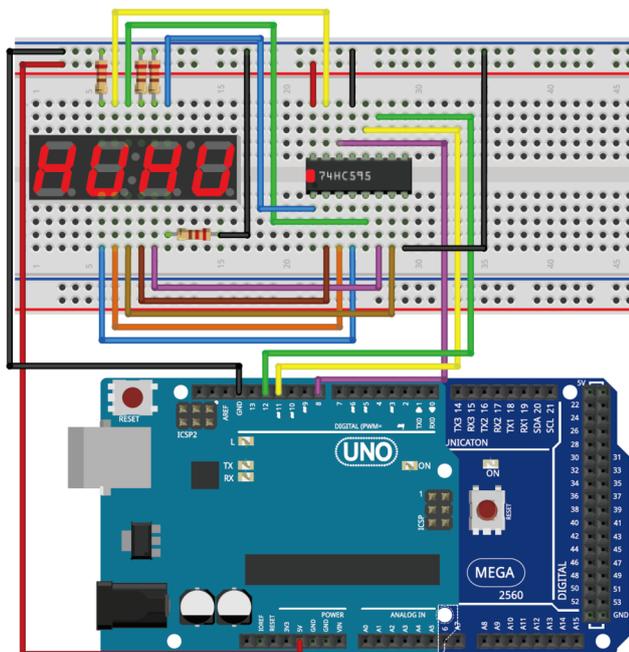
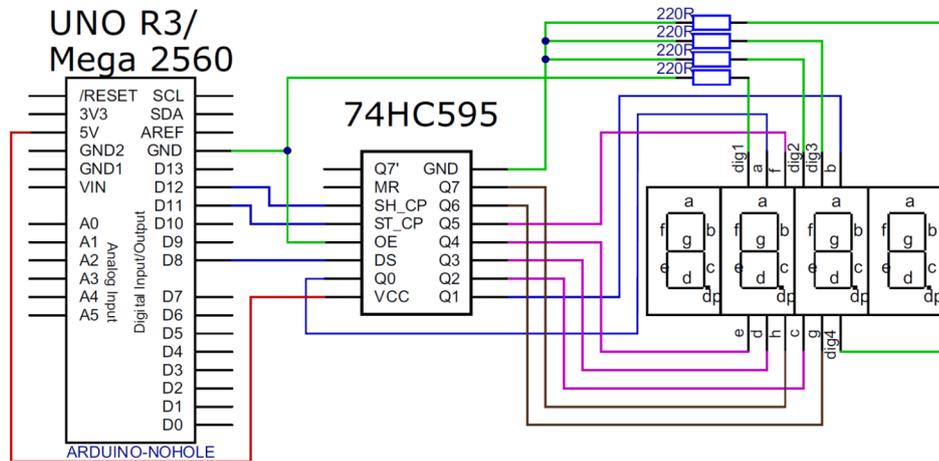
```

34 delay(1000);
35 Display(2); //active 5 segments pour afficher un chiffre 2 sur chaque digit
36 delay(1000);
37 Display(3); //active 5 segments pour afficher un chiffre 3 sur chaque digit
38 delay(1000);
39 }
    
```

Le montage à réaliser est le suivant :



Connection Schematic



**Question n°1**

Qu'est ce qu'un registre à décalage de 8 bits ?

**Question n°2**

En quoi est-il intéressant d'utiliser un registre de ce type dans ce montage ?

**Question n°3**

Réalisez le montage, téléversez le code et constatez le résultat.

**Question n°4**

Ecrire un programme qui affiche 9999 sur les 4 digits puis constatez le fonctionnement.

## 2.3.3. Pilotage d'un afficheur 7 segments à l'aide d'un registre à décalage, bibliothèque SevSeg.h.

Il est possible de piloter plus facilement les 4 digits d'un afficheur par le biais de la bibliothèque SevSeg.h, le programme est le suivant :

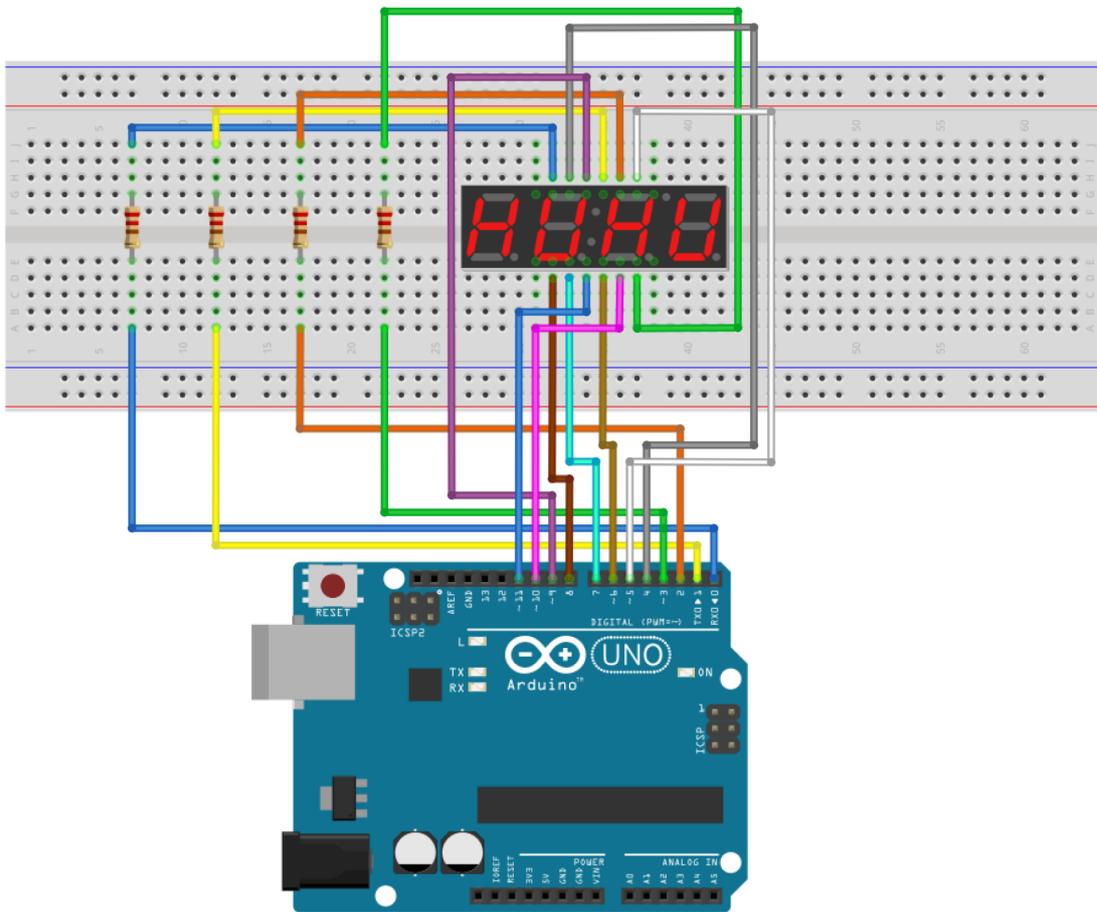
```

1 #include "SevSeg.h"
2
3 SevSeg sevseg;
4
5 void setup() {
6   // Pour le montage il faut connecter les broches 0 à 11 de l'Arduino sur les
7   // broches 1,2,3,4,A,B,C,D,E,F,G,DP de l'afficheur
8   // mettez une résistance 220Ohm entre les broches de l'Arduino et de l'afficheur
9   // 0 et 1, 1 et 2, 2 et 3, 3 et 4
10  sevseg.Begin(0,0,1,2,3,4,5,6,7,8,9,10,11); // Le premier 0 indique qu'il s'agit
11  // d'une cathode commune
12 }
13
14 void loop() {
15   // les 4 chiffres de la fonction NewNum seront affichés,
16   // le dernier chiffre après la virgule donne la position du "." de 0 à 3, un
17   // chiffre supérieur à 3 indique qu'on ne veut pas de virgule
18   sevseg.NewNum(1234,0);
19   sevseg.PrintOutput();
20 }

```

La bibliothèque est disponible ici : Bibliothèque SevSeg<sup>[<https://docs.google.com/file/d/0Bwrp4uluZCpNdE9oWTY0M3BncTA/edit?resourcekey=0-94SXXrPEy4vsIERsFfiFzQJ>]</sup>, téléchargez le fichier Zip puis dans l'IDE Arduino cliquez sur  Croquis/Inclure une bibliothèque/Ajouter la bibliothèque .ZIP ...

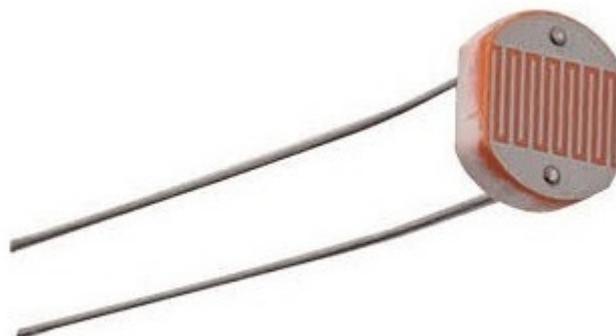
Le montage est le suivant :



## 2.4. Utilisation d'un photoresistor

### 2.4.1. Introduction

L'objectif est d'apprendre à contrôler un bandeau de LED qui indique la luminosité mesurée à l'aide d'un photo-résistor en utilisant un registre à décalage.



*Photoresistor*

Pour réaliser ce montage, il faudra vous munir des composants suivants :

- une carte Arduino
- un câble USB
- un photo-résistor
- un registre à décalage de 8 bits 74HC595
- 8 LED rouges
- 8 résistors de 220 Ohms
- 1 résistor de 10kOhm
- des fils Dupont Mâle/Mâle
- une platine de prototypage rapide à trous de type Labdec

## 2.4.2. Utilisation d'un photoresistor

Le programme ci-dessous permet de contrôler des LED avec un photoresistor :

```

1 // Broche reliée au photo-résistor
2 int lightPin = 0;
3 // Broche reliée au ST_CP du 74HC595
4 int latchPin = 11;
5 // Broche reliée au SH_CP du 74HC595
6 int clockPin = 12;
7 // Broche reliée au DS du 74HC595
8 int dataPin = 8;
9
10 int leds = 0;
11
12 void setup()
13 {
14   pinMode(latchPin, OUTPUT);
15   pinMode(dataPin, OUTPUT);
16   pinMode(clockPin, OUTPUT);
17   pinMode(lightPin, INPUT);
18 }
19
20 void updateShiftRegister()
21 {
22   digitalWrite(latchPin, LOW);
23   shiftOut(dataPin, clockPin, LSBFIRST, leds);
24   digitalWrite(latchPin, HIGH);
25 }
26
27 void loop()
28 {
29   int reading = analogRead(lightPin);
30   int numLEDSLit = reading / 57; // pour calibrer la sensibilité
31   if (numLEDSLit > 8) numLEDSLit = 8;
32   leds = 0;
33   for (int i = 0; i < numLEDSLit; i++)
34   {
35     leds = leds + (1 << i);

```

```

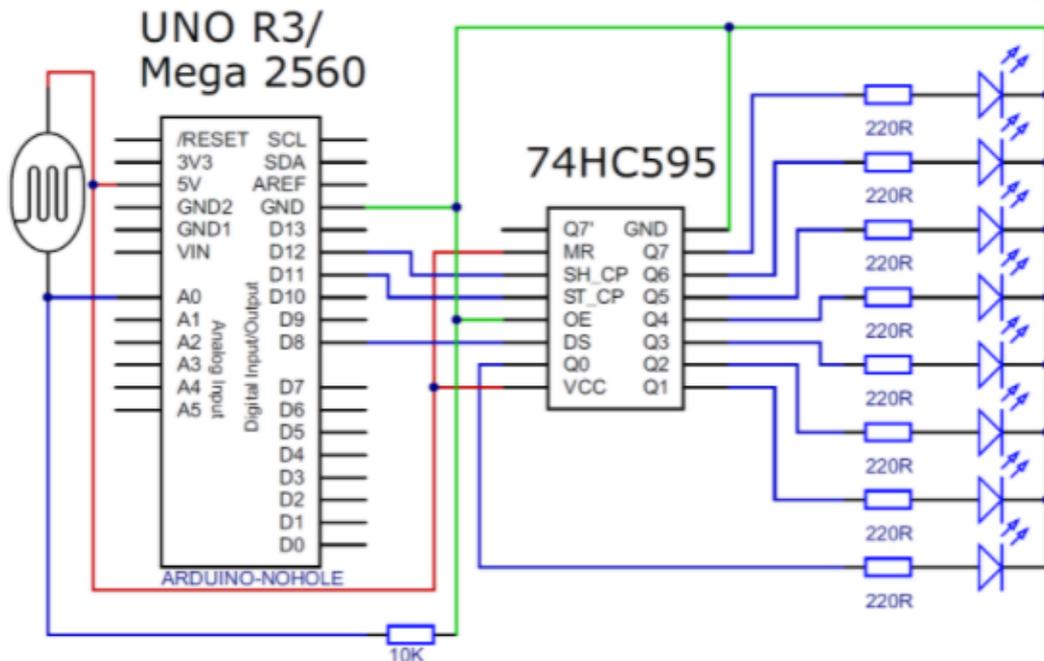
36 }
37 updateShiftRegister();
38 delay (500);
39 }
    
```

## Photorésistance

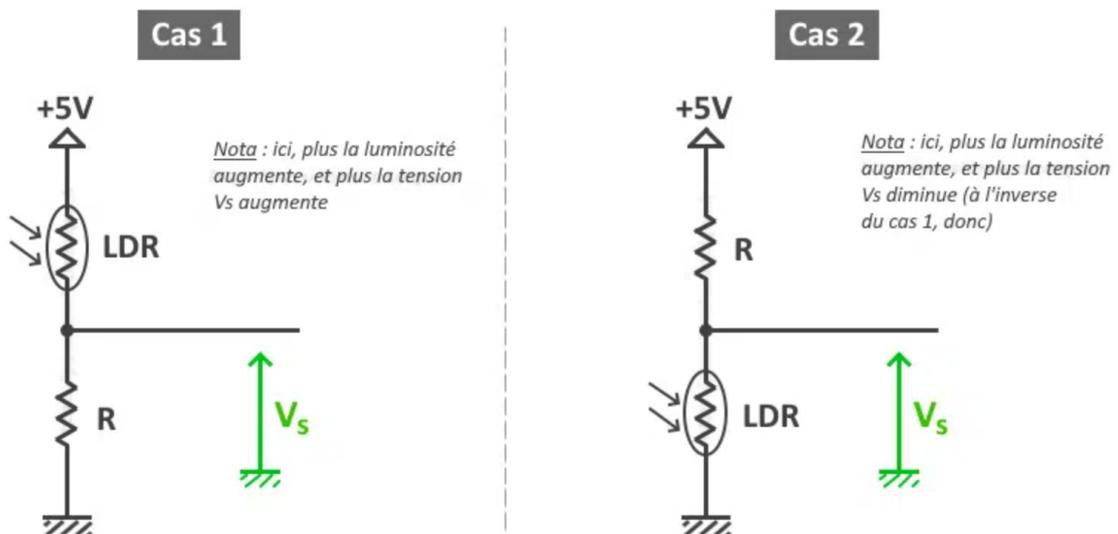
≈ résistance photogénique, cellule photoconductrice ou cellule photoélectrique

Une photorésistance est un composant électronique dont la résistance varie en fonction de la quantité de lumière incidente : plus elle est éclairée, plus sa résistance baisse et réciproquement.

Le montage à réaliser est le suivant :



Les photo-résistors sont généralement implantés dans les circuits au moyen d'un pont diviseur de tension selon un de ces deux schémas :

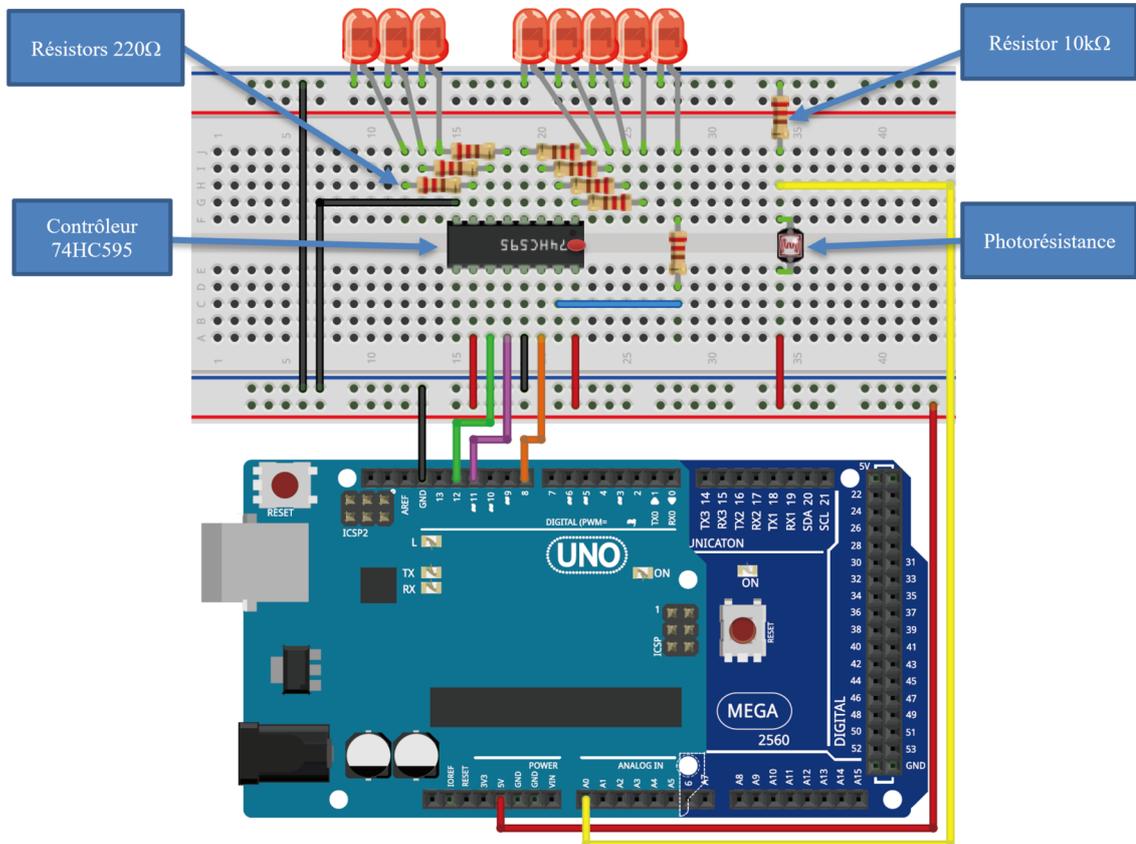


Dans le montage nous avons choisi le cas 1 avec une résistance R de 10kOhm et une tension Vs

mesurée sur l'entrée analogique A0. Dans ce cas, plus la luminosité augmente plus la tension mesurée sur l'entrée analogique se rapproche de 5V.

Pour plus d'information : Photo-résistor [<https://passionelectronique.fr/photoresistance/>]

Dans ce montage, nous utilisons un registre à décalage pour limiter l'utilisation des broches de la carte Arduino.



### Question n°1

Dans le programme, expliquez ce qu'apporte la boucle FOR.

### Question n°2

Réalisez le montage, téléversez le code et constatez le fonctionnement.

### Question n°3

Modifiez le programme et le montage de façon à afficher l'intensité lumineuse sur un afficheur à 7 segments (0 : aucune luminosité, 8 : luminosité maximale) puis constatez son fonctionnement.

## 2.5. Mesure de distance par ultrason

### 2.5.1. Introduction

L'objectif du montage est de mesurer une distance à l'aide d'un capteur à ultrason et de l'afficher sur un écran LCD.

Pour réaliser ce montage, il faudra vous munir des composants suivants :

- une carte Arduino
- un câble USB
- un capteur à ultrason HC-SR04
- un potentiomètre
- un écran LCD
- des fils Dupont Mâle/Mâle
- une platine de prototypage rapide à trous de type Labdec

## 2.5.2. Mesure de distance par ultrason

Le programme ci-dessous permet d'afficher la distance séparant le capteur à ultrason de sa cible :

```

1 #include <LiquidCrystal.h>
2 const int echoPin = 5;
3 const int trigPin = 6;
4
5 LiquidCrystal lcd(7, 8, 9, 10, 11, 12);
6 /*
7  * LCD RS au connecteur 7
8  * LCD Enable au connecteur 8
9  * LCD D4 au connecteur 9
10 * LCD D5 au connecteur 10
11 * LCD D6 au connecteur 11
12 * LCD D7 au connecteur 12
13 */
14
15 void setup()
16 {
17   pinMode(echoPin, INPUT);
18   pinMode(trigPin, OUTPUT);
19   lcd.begin(16, 2); // Règle le nombre de colonnes et de lignes du LCD
20   lcd.clear(); // Efface l'écran du LCD et positionne le curseur en haut à gauche
21   delay(1000);
22 }
23
24 void loop()
25 {
26   int cm = ping(echoPin) ;
27   lcd.setCursor(0, 0); // Place le curseur à la colonne 0 et ligne 0
28   lcd.print("Distance: "); // Affiche le Texte « Distance » sur le LCD
29   lcd.print(cm); // Affiche la distance mesurée par le capteur à ultra son
30   lcd.print(" cm  ");
31   //les espaces après " cm___" servent à effacer une éventuelle mesure résiduelle
32   // précédente, ceci permet d'éviter d'avoir à effacer l'écran entre deux
   affichages
33   delay(500);

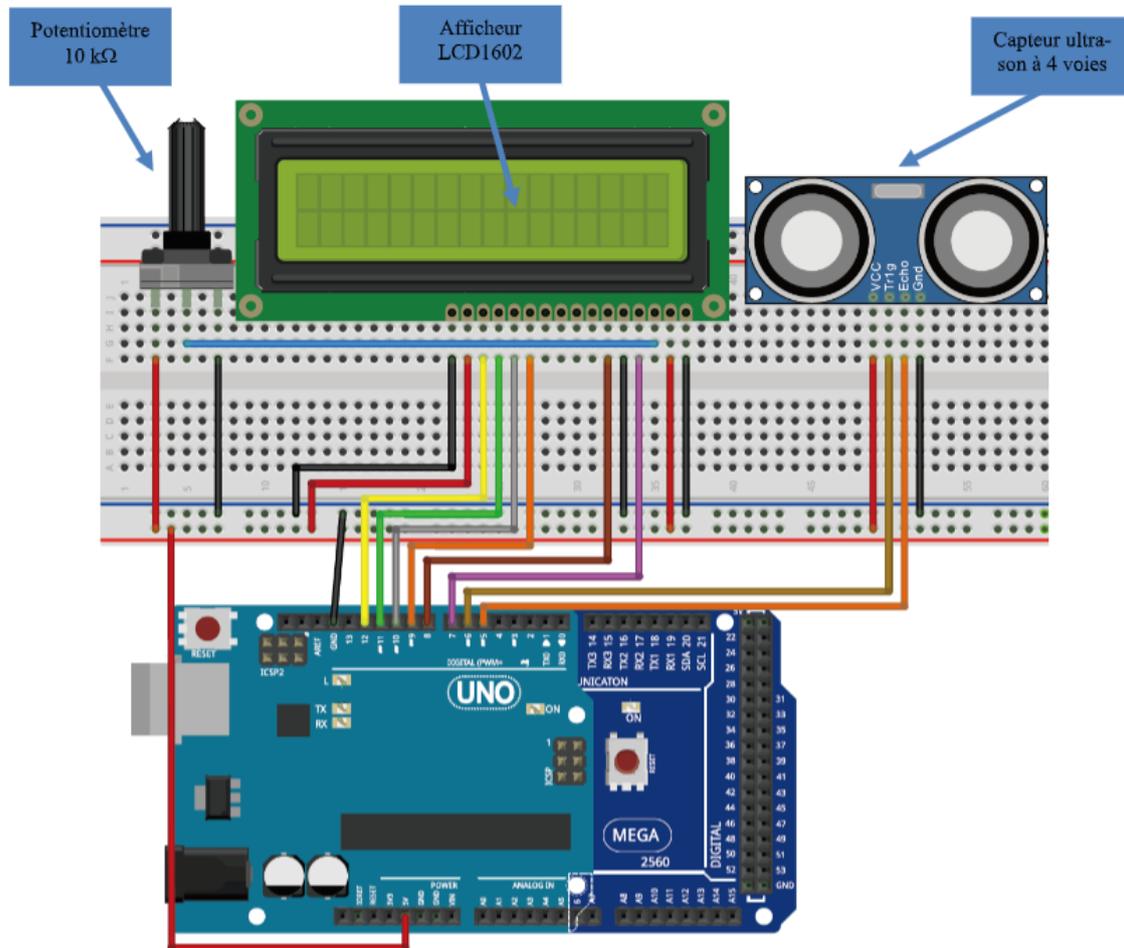
```

```

34 }
35
36 int ping(int echoPin)
37 {
38     // Cette fonction mesure la distance à la cible à partir du temps de réponse
    mesuré
39     long duration, cm;
40     // Un ping sonore est généré quand une pulsation HIGH est générée pendant 2
    microsecondes ou plus.
41     // Une courte pulsation LOW est envoyée avant la commande du trigger pour
    nettoyer le signal
42     pinMode(trigPin, OUTPUT);
43     digitalWrite(trigPin, LOW);
44     delayMicroseconds(2);
45     digitalWrite(trigPin, HIGH);
46     delayMicroseconds(5);
47     digitalWrite(trigPin, LOW);
48
49     pinMode (echoPin, INPUT);
50     // la fonction pulseIn (soit HIGH ou LOW) lit une impulsion sur une broche
51     // par exemple, si la valeur est HIGH, pulseIn() attends que la broche passe de
    LOW à HIGH,
52     // lance un timer, attend que la broche revienne à LOW et arrête le timer,
53     // la fonction renvoie ensuite le temps écoulé en microsecondes ou retourne 0 si
    aucune impulsion n'est mesurée avant le timeout.
54     duration = pulseIn(echoPin, HIGH);
55
56     // convertit le temps en distance en cm
57     cm = microsecondsToCentimeters(duration);
58     return cm ;
59 }
60
61 long microsecondsToCentimeters(long microseconds)
62 {
63     // La vitesse du son est de 340 m/s ou 29 microsecondes par centimètre.
64     // Le signal du ping part et revient suite à l'impact sur l'objet pendant la
    mesure ce qui fait qu'il
65     // faut diviser la distance mesurée par 2.
66     return microseconds / 29 / 2;
67 }

```

Le montage à réaliser est le suivant :



### Question n°1

Comment fonctionne le capteur à ultrason ?

### Question n°2

Comment fonctionne un écran LCD ?

### Question n°3

A quoi sert le potentiomètre sur le montage ?

### Question n°4

Réalisez le montage, téléversez le code et constatez le fonctionnement.

### Question n°5

Modifiez le programme pour afficher la distance sur la première ligne et le temps de réponse sur la deuxième.

## 2.6. Mesure de distance par télémètre laser

## 2.6.1. Introduction

### Objectifs pédagogiques

L'objectif du montage est de mesurer une distance à l'aide d'un télémètre laser et de l'afficher sur un écran LCD.

Pour réaliser ce montage, il faudra vous munir des composants suivants :

- une carte Arduino
- un câble USB
- un télémètre laser VL53L0X QT
- un potentiomètre
- un écran LCD
- des fils Dupont Mâle/Mâle
- une platine de prototypage rapide à trous de type Labdec

## 2.6.2. Mesure de distance par télémètre laser

Le programme ci-dessous permet d'afficher la distance séparant le télémètre laser de sa cible :

```
1 #include <LiquidCrystal.h>
2 #include "Adafruit_VL53L0X.h"
3
4 LiquidCrystal lcd(7, 8, 9, 10, 11, 12);
5 /*
6  * LCD RS au connecteur 7
7  * LCD Enable au connecteur 8
8  * LCD D4 au connecteur 9
9  * LCD D5 au connecteur 10
10 * LCD D6 au connecteur 11
11 * LCD D7 au connecteur 12
12 */
13
14 Adafruit_VL53L0X lox = Adafruit_VL53L0X();
15
16 void setup() {
17   Serial.begin(115200);
18   // Pour attendre que le port soit prêt pour communiquer en série avec le capteur
19   while (! Serial) {
20     delay(1);
21   }
22   lcd.begin(16, 2);
23   lcd.clear();
24   delay(1000);
25   lcd.setCursor(0, 0);
```

```

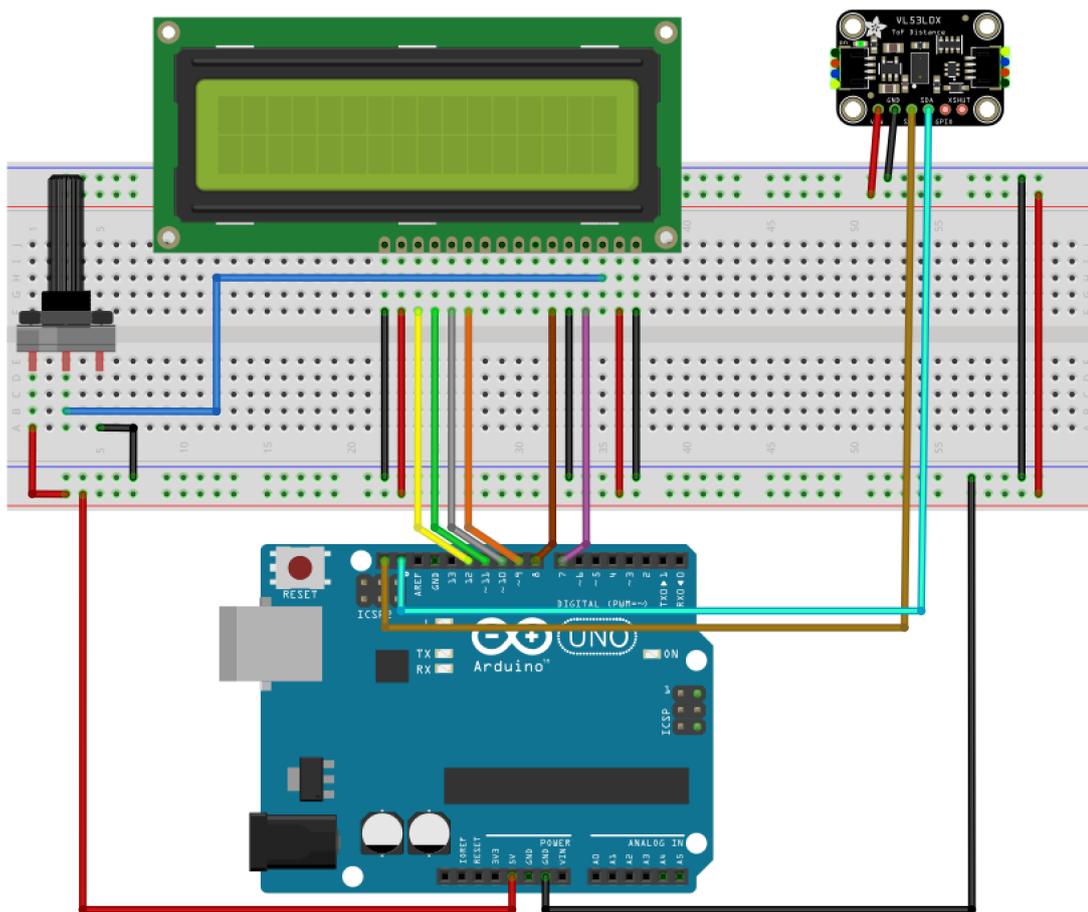
26 lcd.print("Adafruit VL53L0X test");
27 // Vérification du fonctionnement du capteur
28 if (!lox.begin()) {
29     lcd.print("Failed to boot VL53L0X");
30     while(1);
31 }
32 lcd.setCursor(0, 1);
33 lcd.print("VL53L0X Ready");
34 delay(1000);
35 lcd.clear();
36 }
37
38 void loop() {
39     VL53L0X_RangingMeasurementData_t measure;
40     lcd.setCursor(0, 0);
41     // Mesure et vérification qu'elle n'est pas en dehors de la zone de mesure
42     lox.rangingTest(&measure, false);
43     if (measure.RangeStatus != 4) {
44         lcd.print("Dist : "); lcd.print(measure.RangeMilliMeter); lcd.print(" mm ");
45     } else {
46         lcd.print("Out of range ! ");
47     }
48     delay(100);
49 }

```

La bibliothèque Adafruit\_VL53L0X est téléchargeable sur le site : [Adafruit\\_VL53L0X<sup>\[https://github.com/adafruit/Adafruit\\_VL53L0X\]</sup>](https://github.com/adafruit/Adafruit_VL53L0X) en cliquant sur  Code/Download ZIP.

Ensuite, pour intégrer cette bibliothèque dans l'IDE Arduino, il faut aller dans :  Croquis/Inclure une bibliothèque/Ajouter la bibliothèque .ZIP ...

Le montage à réaliser est le suivant :



### Question n°1

Comment fonctionne ce télémètre laser ?

### Question n°2

Réalisez le montage, téléversez le code et constatez le fonctionnement.

### Question n°3

Modifiez le programme pour que la distance soit affichée dans le moniteur série.

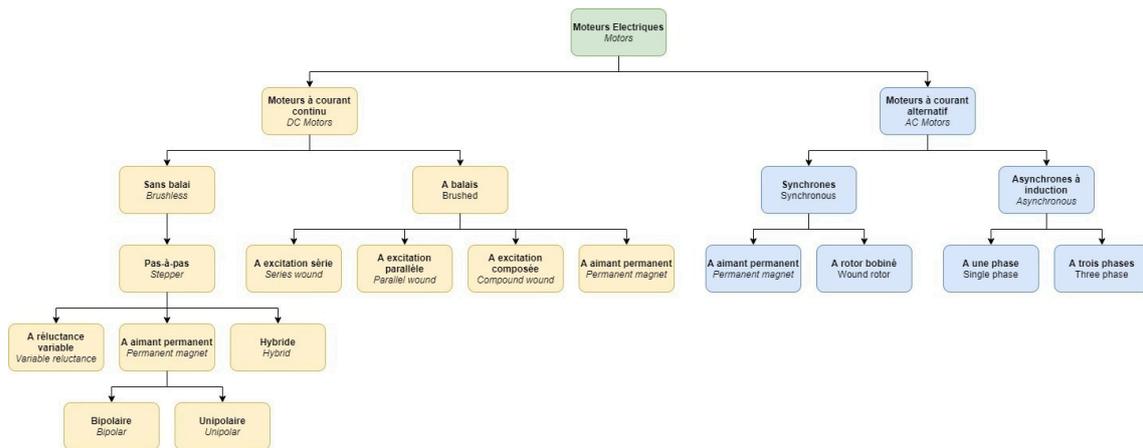
## 2.7. Les moteurs électriques

### 2.7.1. Introduction

#### Objectifs pédagogiques

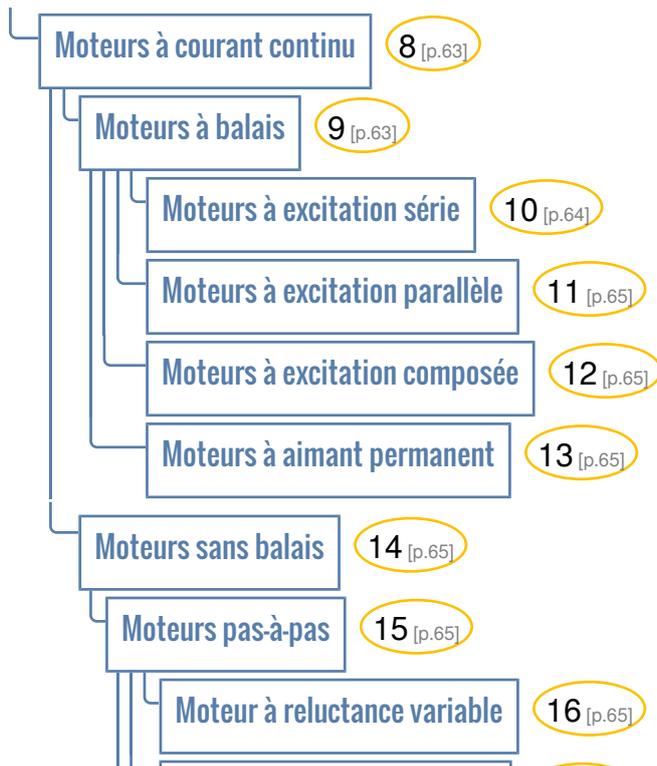
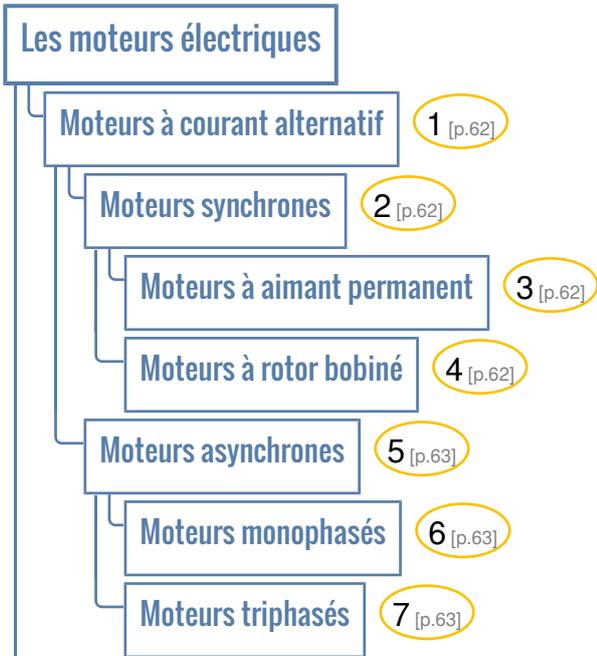
L'objectif de cette ressource est de présenter de manière synthétique les principales familles de moteurs et leurs spécificités.

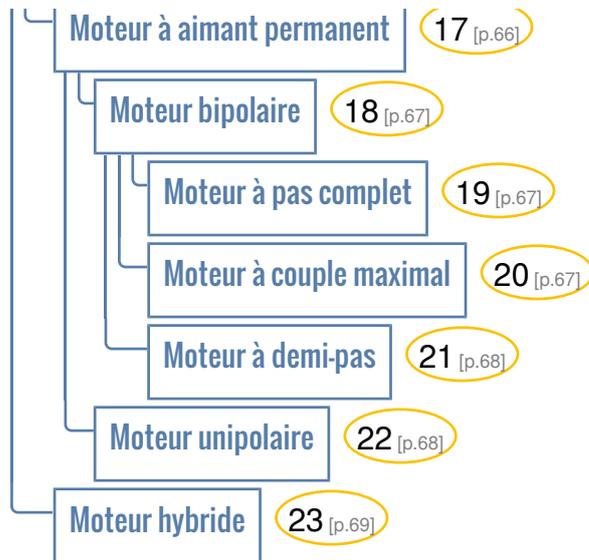
Les moteurs électriques se divisent en 2 grandes familles, les moteurs à courant continu et les moteurs à courant alternatifs.



Les familles de moteurs électriques

## 2.7.2. Classification des moteurs électriques





### Moteurs à courant alternatif

Ces moteurs sont les plus polyvalents et trouvent de nombreux usages, notamment pour les fortes charges.

Par rapport aux moteurs à courant continu, ils présentent quelques avantages :

- Meilleur rapport poids/encombrement/puissance
- Faible consommation au démarrage
- Accélération, courant de démarrage, vitesse de fonctionnement, limite de couple contrôlables
- Perturbations réduites de la ligne de puissance

### Moteurs synchrones

La rotation du rotor de ces moteurs est synchronisée avec la fréquence du courant d'alimentation, et la vitesse reste constante en cas de variation de la charge. Ce type de moteur est donc idéal pour les systèmes qui nécessitent une vitesse constante.

Applications habituelles : appareils de positionnement de grande précision comme les robots, l'instrumentation, les machines spéciales et le contrôle de processus

Pour plus d'informations

Les moteurs synchrones [\[https://energieplus-lesite.be/techniques/ascenseurs7/moteur-synchrone/\]](https://energieplus-lesite.be/techniques/ascenseurs7/moteur-synchrone/)

### Moteurs à aimant permanent

Dans cette configuration, le stator est bobiné et le rotor est à aimants permanents.

Ce sont des moteurs qui peuvent accepter des courants de surcharge importants pour démarrer rapidement.

Applications habituelles : motorisation d'ascenseurs compactes ayant une accélération rapide (immeuble de grande hauteur par exemple).

### Moteurs à rotor bobiné

Dans cette configuration, le stator et le rotor sont bobinés et combinés à un système de balais et de

collecteurs.

Ce type de machines est réversible car elles peuvent fonctionner en régime moteur comme en régime alternateur.

Applications habituelles : pour des machines performantes en moyennes et fortes puissances.

### Moteurs asynchrones

Ces moteurs utilisent un champ magnétique tournant qui induit un courant dans des conducteurs en court circuit situés sur le rotor. La combinaison du champ magnétique tournant et du courant passant dans les conducteurs du rotor produit une force électromagnétique qui se traduit mécaniquement par un couple et une vitesse de rotation. Ce type de moteur est intéressant pour sa capacité de charge et pour son couple relativement constant quelque soit sa vitesse.

Applications habituelles : systèmes industriels telles que les compresseurs, pompes, systèmes de convoyeurs et le matériel de levage.

Pour plus d'informations

Les moteurs asynchrones [<https://energieplus-lesite.be/techniques/ascenseurs7/moteur-asynchrone/#:~:text=Dans%20un%20moteur%20asynchrone%2C%20c,%C3%A0%20rattraper%20le%20champ%20tournant.>]

### Moteurs monophasés

Ils sont utilisés pour les faibles charges.

### Moteurs triphasés

Ils sont utilisés pour les fortes charges.

### Moteurs à courant continu

Ces moteurs sont très utilisés pour des applications à faibles charges.

Par rapport aux moteurs alternatifs, ils présentent quelques avantages :

- Installation facile
- Commande de vitesse simple
- Démarrage, arrêt, marche arrière et accélération rapides
- Couple de démarrage élevé
- Courbe couple-vitesse linéaire

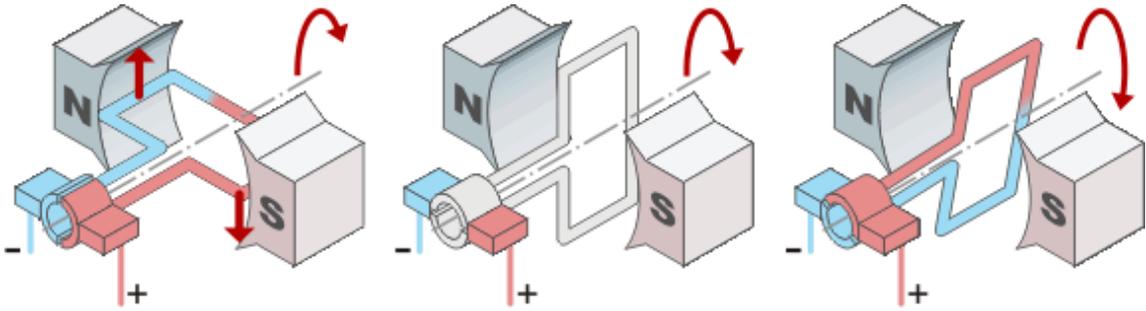
Pour plus d'informations

Les moteurs à courant continu [<https://energieplus-lesite.be/techniques/ascenseurs7/moteur-a-courant-continu/>]

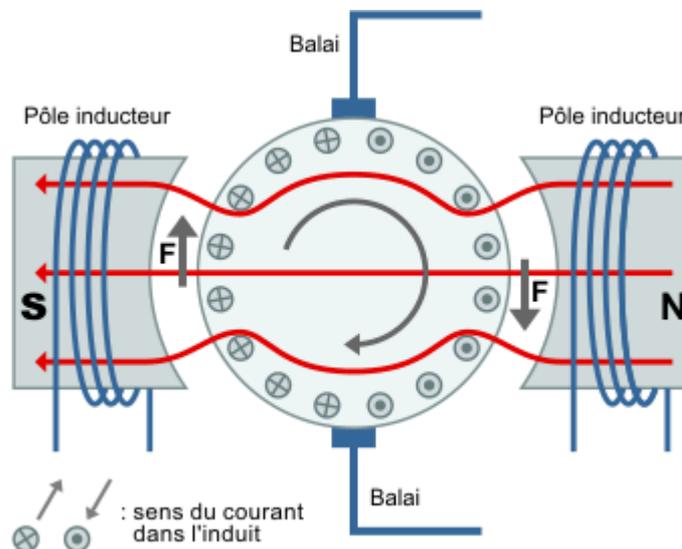
### Moteurs à balais

Famille de moteur à courant continu la plus répandue car offrant l'avantage d'être peu coûteuse à l'achat et simple à commander puisqu'il suffit de faire varier leur tension d'alimentation pour contrôler leur vitesse de rotation.

Ces moteurs utilisent un stator inducteur (bobiné ou à aimant permanent) et un rotor induit et des collecteurs et des balais qui servent à inverser alternativement le sens de circulation du courant dans le rotor afin de le faire tourner en fonction de son orientation par rapport aux pôles du stator.



Si le système balais-collecteurs n'était pas présent (simple spire alimentée en courant continu), la spire s'arrêterait de tourner en position verticale sur un axe appelé communément "ligne neutre". Le système balais-collecteurs a pour rôle de faire commuter le sens du courant dans les deux conducteurs au passage de la ligne neutre. Le courant étant inversé, les forces motrices sur les conducteurs le sont aussi permettant ainsi de poursuivre la rotation de la spire.



Dans la pratique, la spire est remplacée par un induit (rotor) de conception très complexe sur lequel sont montés des enroulements (composés d'un grand nombre de spires) raccordés à un collecteur "calé" en bout d'arbre. Dans cette configuration, l'induit peut être considéré comme un seul et même enroulement semblable à une spire unique.

Pour plus d'informations

Les moteurs à courant continu <https://energieplus-lesite.be/techniques/ascenseurs7/moteur-a-courant-continu/>

### Moteurs à excitation série

Dans cette configuration, le bobinage du stator est connecté en série au bobinage du rotor. La contrôle de la vitesse est réalisé en faisant varier la tension d'alimentation. Ce type de moteur offre un contrôle peu efficace de la vitesse puisque lorsque le couple résistant augmente, la vitesse du moteur chute.

Applications habituelles : systèmes exigeant un couple de démarrage élevé comme les automobiles,

les palans, ascenseurs et grues.

### Moteurs à excitation parallèle

Dans cette configuration le bobinage du stator est connecté en parallèle au bobinage du rotor afin de fournir un couple plus élevé, sans réduction de vitesse lors d'une augmentation du courant délivré au moteur. Son couple de démarrage est moyen avec une vitesse constante.

Applications habituelles : tours d'usinage, aspirateurs, convoyeurs et meuleuses.

### Moteurs à excitation composée

Dans cette configuration, le bobinage du stator est connecté en parallèle et en série au bobinage du rotor. Ainsi la polarité du bobinage parallèle s'ajoute aux champs en série. Ce type de moteur possède un couple de démarrage élevé et offre un large variation de vitesse.

Applications habituelles : compresseurs, pompes centrifuges à tête variable, presses rotatives, scies circulaires, machines de cisaillement, ascenseurs et carrousels à bagages.

### Moteurs à aimant permanent

Dans cette configuration, l'électroaimant du stator est remplacé par un aimant permanent. Ces moteurs peuvent être commandés précisément mais le couple fourni est faible.

Applications habituelles : robotique et servo-systèmes.

### Moteurs sans balais

Les moteurs sans balais réduisent les problèmes d'usure liés à l'utilisation de balais et leur conception mécanique est beaucoup plus simple. Les contrôleurs de ce type de moteur utilisent des capteurs à effet Hall pour détecter la position du rotor et ainsi mieux réguler sa vitesse. Les avantages de cette technologie sont une longue durée de vie, peu d'entretien et un haut rendement (85 à 90%), tandis que les inconvénients sont des coûts élevés liés à des contrôleurs plus compliqués.

Applications habituelles : contrôle de positionnement et de vitesse avec des applications telles que les ventilateurs, propulsion de drone, pompes et compresseurs, qui nécessitent fiabilité et robustesse.

### Moteurs pas-à-pas

Ces moteurs sont principalement utilisés dans le contrôle de position et se prêtent bien à l'asservissement en boucle ouverte (pas d'adaptation de la commande à la position réelle du système). Généralement lents, leur vitesse maximale est de l'ordre de 3000 tr/min. Leur durée de vie est grande.

Applications habituelles : imprimantes 3D, équipements de positionnement.

### Moteur à réluctance variable

Les moteurs à réluctance variable (autrement dit à résistance magnétique variable) doivent leur nom au fait que le circuit magnétique qui les compose s'oppose de façon variable à sa pénétration par un champ magnétique.

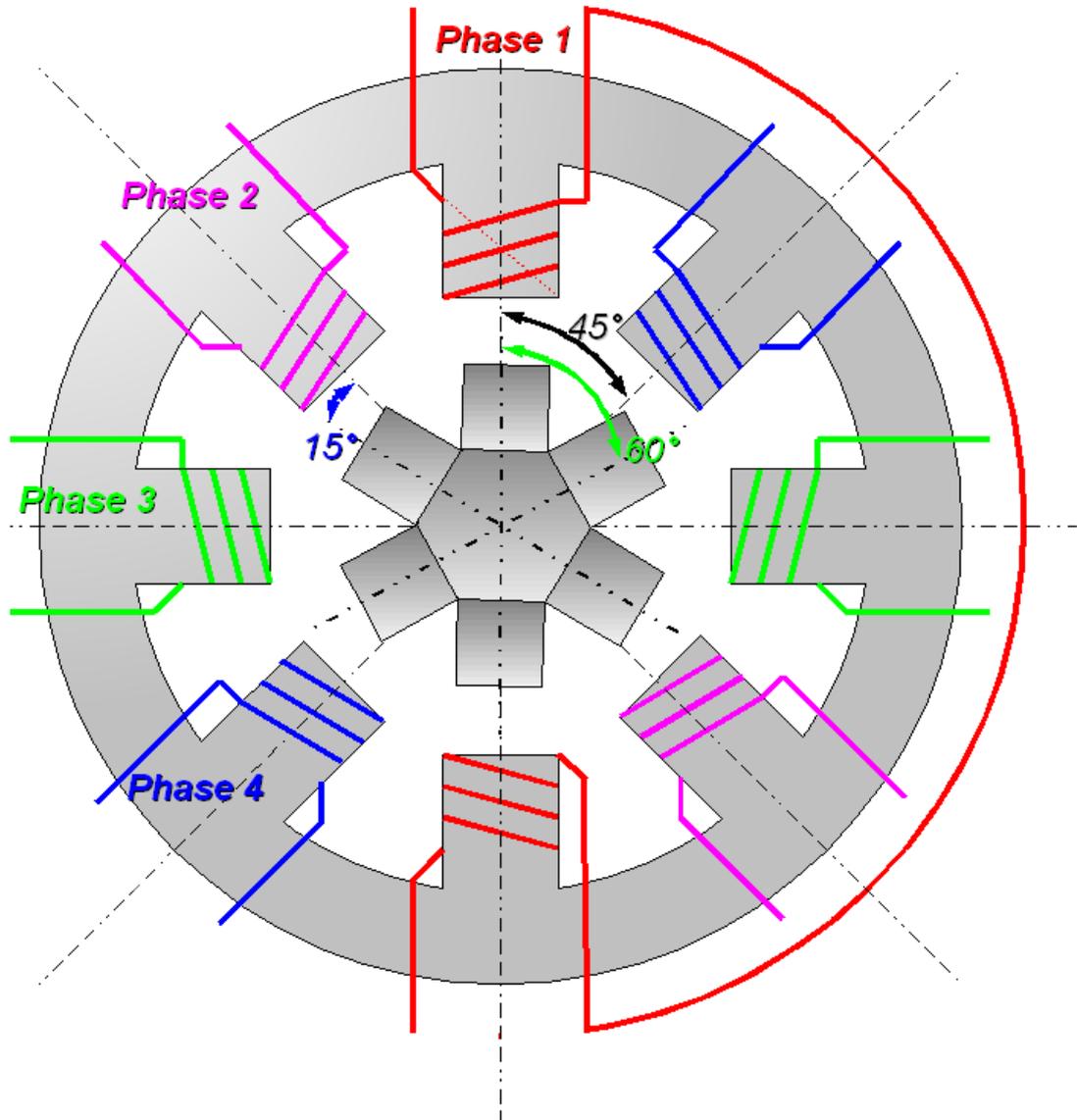
Ces moteurs sont composés d'un barreau de fer doux et d'un certain nombre de bobines. Lorsqu'on alimente une bobine, le champ magnétique cherche à minimiser le passage dans l'air. Ainsi

l'entrefer entre la bobine et le barreau se réduit. Le barreau s'aligne avec le champ magnétique pour obtenir une réluctance minimale. Pour faire tourner le moteur on alimente successivement les phases qui se suivent, dans un sens ou dans l'autre. Une phase alimente généralement deux bobines diamétralement opposées afin de générer un couple sur le barreau de ferrite.

Dans la pratique, le barreau de ferrite a plusieurs dents.

Pour connaître le nombre de pas par tour de ce type de moteur, il faut compte le nombre de bobine X, le nombre de dents de la ferrite Y. L'angle de rotation entre deux alimentations successives de phase est alors  $(360/Y) - (360/X) = Z$  si  $Y < X$ . Le nombre de pas est alors  $360/Z$ .

Par exemple dans le cas d'un moteur avec 8 bobines (soit 4 phases), 6 dents pour la ferrite, le moteur tourne de  $60^\circ - 45^\circ = 15^\circ$  par changement de phase, soit  $360/15 = 24$  pas par tour.



*Moteur à réluctance variable à 4 phases et ferrite à 6 dents*

Les inconvénients de ce moteur est qu'il nécessite au moins trois bobinages, qu'il n'a pas de couple résiduel (hors tension, le rotor est libre) et que sa fabrication est difficile puisque les entrefers doivent être très faibles. Ses avantages sont qu'il est peu coûteux, d'une bonne précision (par exemple, avec 4 bobinages on obtient 24 pas, il peut monter jusqu'à 360 pas) et que le sens du courant dans la bobine n'a aucune importance.

### Moteur à aimant permanent

Les moteurs à aimants permanents le rotor possède des pôles NORD et SUD. En raison des aimants permanents, le rotor reste freiné à sa dernière position lorsque le bloc d'alimentation cesse de fournir des impulsions.

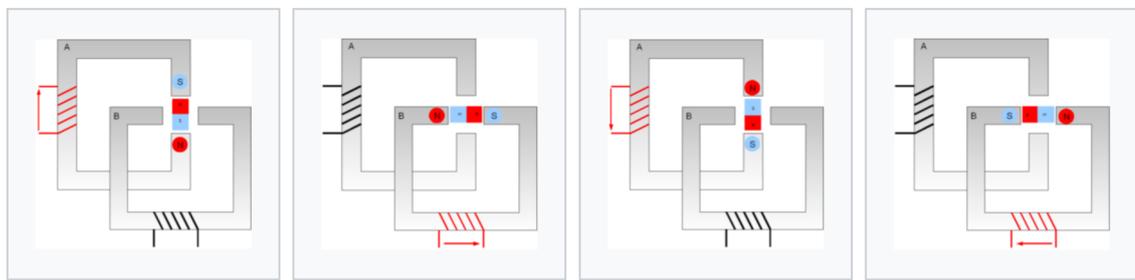
Ces moteurs peuvent être bipolaire ou unipolaire.

### Moteur bipolaire

Dans cette configuration le courant circule dans les deux sens dans les enroulements des bobines.

### Moteur à pas complet

Dans cette configuration, une seule bobine est pilotée à la fois et le courant change de sens de circulation dans la même bobine à chaque demi-révolution.



Pas n° 1

Pas n° 2

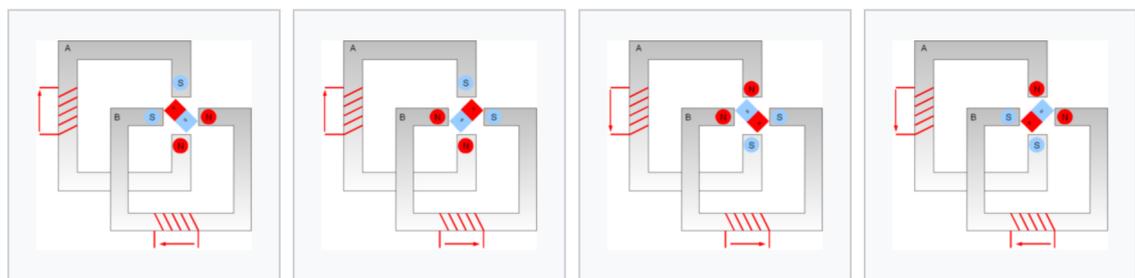
Pas n° 3

Pas n° 4

*Configuration à pas complet*

### Moteur à couple maximal

Dans cette configuration, deux bobines sont alimentées en même temps, ce qui fait que le couple est deux fois plus important que dans le cas précédent.



Pas n° 1

Pas n° 2

Pas n° 3

Pas n° 4

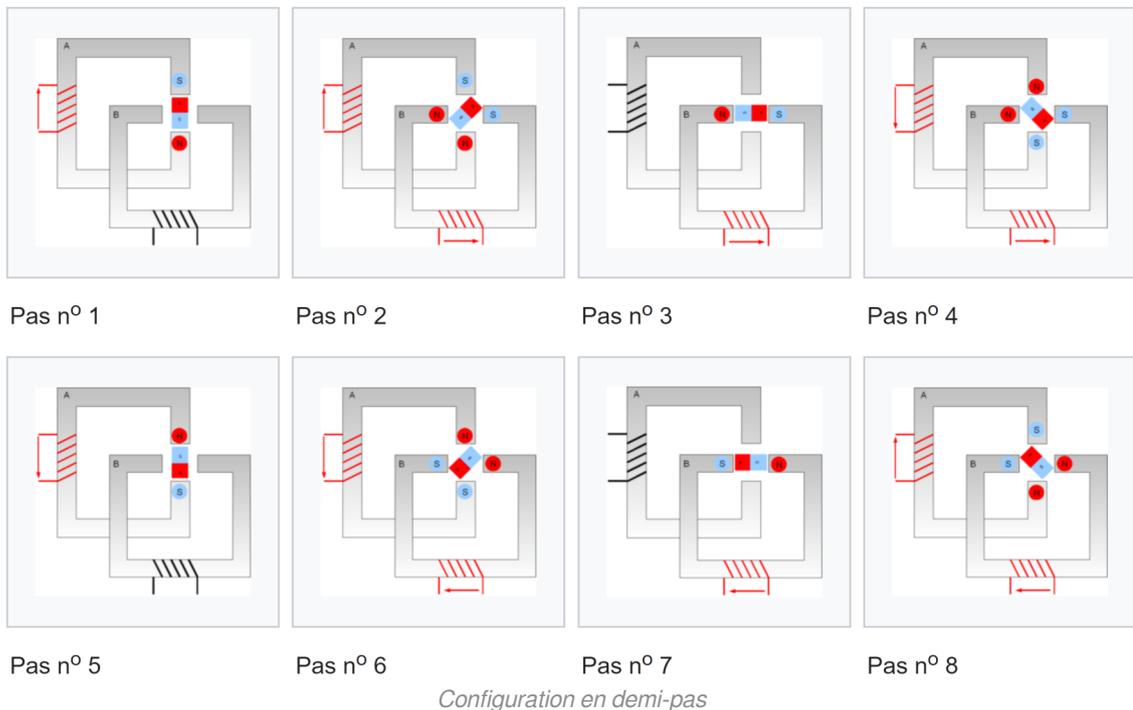
*Configuration à fort couple*

| Impulsion | Bobine A | Bobine A | Bobine B | Bobine B |
|-----------|----------|----------|----------|----------|
| T1        | +        | -        | +        | -        |
| T2        | +        | -        | -        | +        |
| T3        | -        | +        | -        | +        |
| T4        | -        | +        | +        | -        |

Alimentation des bobines au cours du cycle

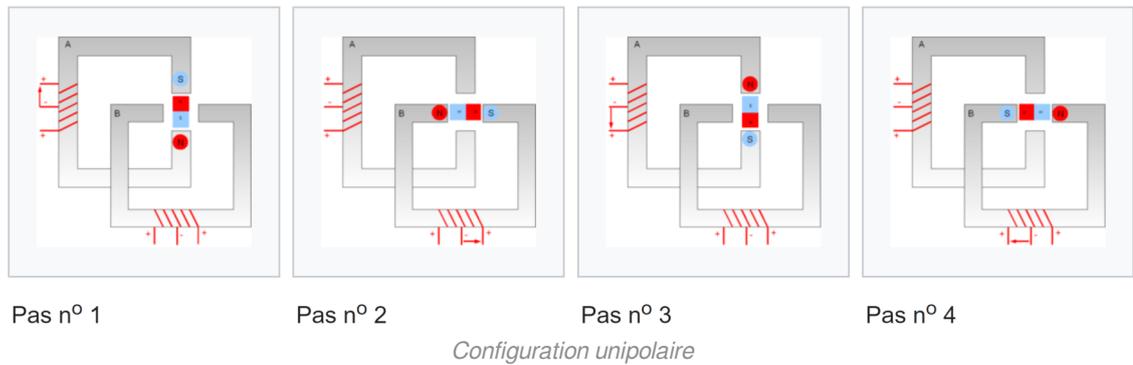
### Moteur à demi-pas

Dans cette configuration, le pas complet et le couple maximal sont combinés, ce qui augmente le nombre de pas par tour mais génère un couple variable.



### Moteur unipolaire

Dans cette configuration, on utilise des demi-bobines (bobines avec un point milieu) afin de ne jamais inverser le sens du courant, ce qui simplifie la commande. Cette configuration nécessite de doubler le nombre d'enroulements, ce qui fait que le moteur est plus coûteux et encombrant.



### Moteur hybride

Le moteur pas à pas hybride emprunte du moteur à aimant permanent et de la machine à réluctance variable. Il est donc à réluctance variable mais avec un rotor à aimants permanents. L'avantage est un nombre de pas très élevé.

## 2.8. Contrôle de servomoteur

### 2.8.1. Introduction

#### Objectifs pédagogiques

L'objectif de cette ressource est de contrôler un servomoteur en utilisant la bibliothèque Servo.h.

Pour réaliser ce montage, il faudra vous munir des composants suivants :

- une carte Arduino
- un câble USB
- un servomoteur
- des fils Dupont Mâle/Mâle
- une platine de prototypage rapide à trous de type Labdec

### 2.8.2. Contrôle de servomoteur avec la bibliothèque Servo.h

Le programme ci-dessous permet d'afficher les caractères 1 à 9 puis de a à f sur chaque afficheur en changeant d'afficheur toutes les 0.5s.

```

1 #include <Servo.h> // Inclut la bibliothèque Servo.h dans le programme
2 Servo myservo; // Produit un objet Servo à partir de la bibliothèque
3
4 void setup()
5 {

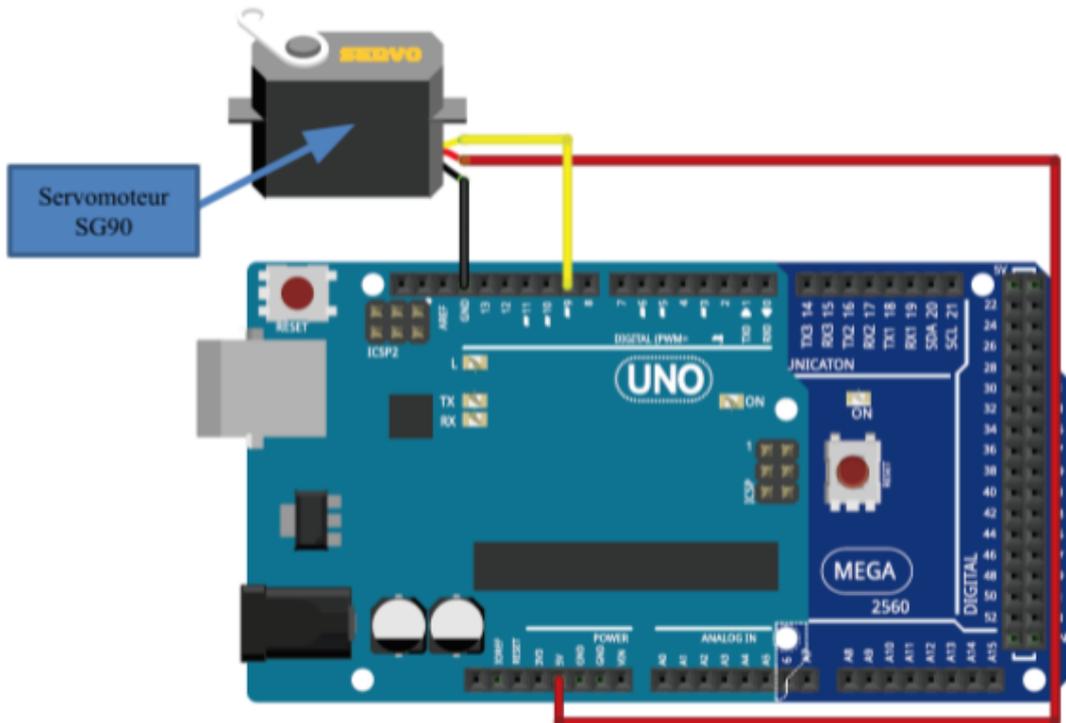
```

```

6  myservo.attach(9); // Attache le servomoteur au connecteur 9
7  myservo.write(0); // Positionne le palonnier à 0 degré
8  delay(1000);
9  }
10
11 void loop()
12 {
13  myservo.write(15); // Tourne de 15 degrés
14  delay(1000);
15  myservo.write(30);
16  delay(1000);
17  myservo.write(45);
18  delay(1000);
19  myservo.write(60);
20  delay(1000);
21  myservo.write(75);
22  delay(1000);
23  myservo.write(90);
24  delay(1000);
25  myservo.write(75);
26  delay(1000);
27  myservo.write(60); // Retourne à 60 degrés
28  delay(1000);
29  myservo.write(45);
30  delay(1000);
31  myservo.write(30);
32  delay(1000);
33  myservo.write(15);
34  delay(1000);
35  myservo.write(0);
36  delay(1000);
37  for(int num=0;num<=180;num++)
38  {
39      myservo.write(num); // Fait tourner le servomoteur jusqu'à la position
40      delay(10); // Contrôle la vitesse de rotation
41  }
42  for(int num=180;num>=0;num--)
43  {
44      myservo.write(num);
45      delay(10);
46  }
47 }

```

Le montage à réaliser est le suivant :



### Question n°1

Qu'est ce qu'un servo-moteur ?

### Question n°2

Réalisez le montage, téléversez le code et constatez le fonctionnement.

### Question n°3

Modifiez le montage et le programme de façon à ce que le servo-moteur soit contrôlé par 2 boutons : un qui le fait tourner avec un pas de  $10^\circ$  dans le sens horaire et un autre dans l'autre sens.

## 2.9. Contrôle de 2 moteurs à courant continu

### 2.9.1. Introduction

#### Objectifs pédagogiques

L'objectif de cette ressource est d'apprendre à piloter deux moteurs à courant continu avec un motor shield Arduino

Pour réaliser ce montage, il faudra vous munir des composants suivants :

- une carte Arduino

- un câble USB
- deux moteurs à courant continu fonctionnant sous 9V, 18W max
- un Arduino Motor Shield Rev 3
- une alimentation externe de 9V

## 2.9.2. Contrôle de 2 moteurs à courant continu

Le programme ci-dessous permet de contrôler deux moteurs à courant continu :

```

1 const int vitesseMotA=3; // Constante pour la broche 3
2 const int sensMotA=12; // Constante pour la broche 12
3 const int freinMotA=9; // Constante pour la broche 9
4 const int intensiteMotA=A0; // intensité du moteur 0
5
6 const int vitesseMotB=11; // Constante pour la broche 11
7 const int sensMotB=13; // Constante pour la broche 13
8 const int freinMotB=8; // Constante pour la broche 8
9 const int intensiteMotB=A1; // intensité du moteur 1
10
11 void setup()
12 {
13     Serial.begin(115200);
14     pinMode (vitesseMotA, OUTPUT); // Broche vitesseMotA configurée en sortie
15     pinMode (freinMotA, OUTPUT); // Broche freinMotA configurée en sortie
16     pinMode (vitesseMotB, OUTPUT); // Broche vitesseMotB configurée en sortie
17     pinMode (sensMotA, OUTPUT); // Broche sensMotA configurée en sortie
18     pinMode (sensMotB, OUTPUT); // Broche sensMotB configurée en sortie
19
20     digitalWrite(vitesseMotA, LOW); // a l'arret
21     digitalWrite(sensMotA, LOW);
22     digitalWrite(freinMotA, LOW); // frein off
23
24     digitalWrite(vitesseMotB, LOW); // à l'arret
25     digitalWrite(sensMotB, LOW);
26     digitalWrite(freinMotB, LOW); // frein off
27 }
28
29 void loop()
30 {
31     //----- test initial du moteur A ----
32     //- sens 1
33     digitalWrite(sensMotA, LOW); // sens 1
34     digitalWrite(vitesseMotA, HIGH); // vitesse maximale
35     delay(2000); // 2 secondes
36     digitalWrite(vitesseMotA, LOW); // vitesse maximale
37     //- sens 2
38     digitalWrite(sensMotA, HIGH); // sens 2
39     digitalWrite(vitesseMotA, HIGH); // vitesse maximale
40     delay(1000); // 2 secondes
41     Serial.println(analogRead(intensiteMotA));
42     delay(1000); // 2 secondes
43     digitalWrite(vitesseMotA, LOW); // vitesse maximale
44
45     //----- test initial du moteur B ----
46     //- sens 1

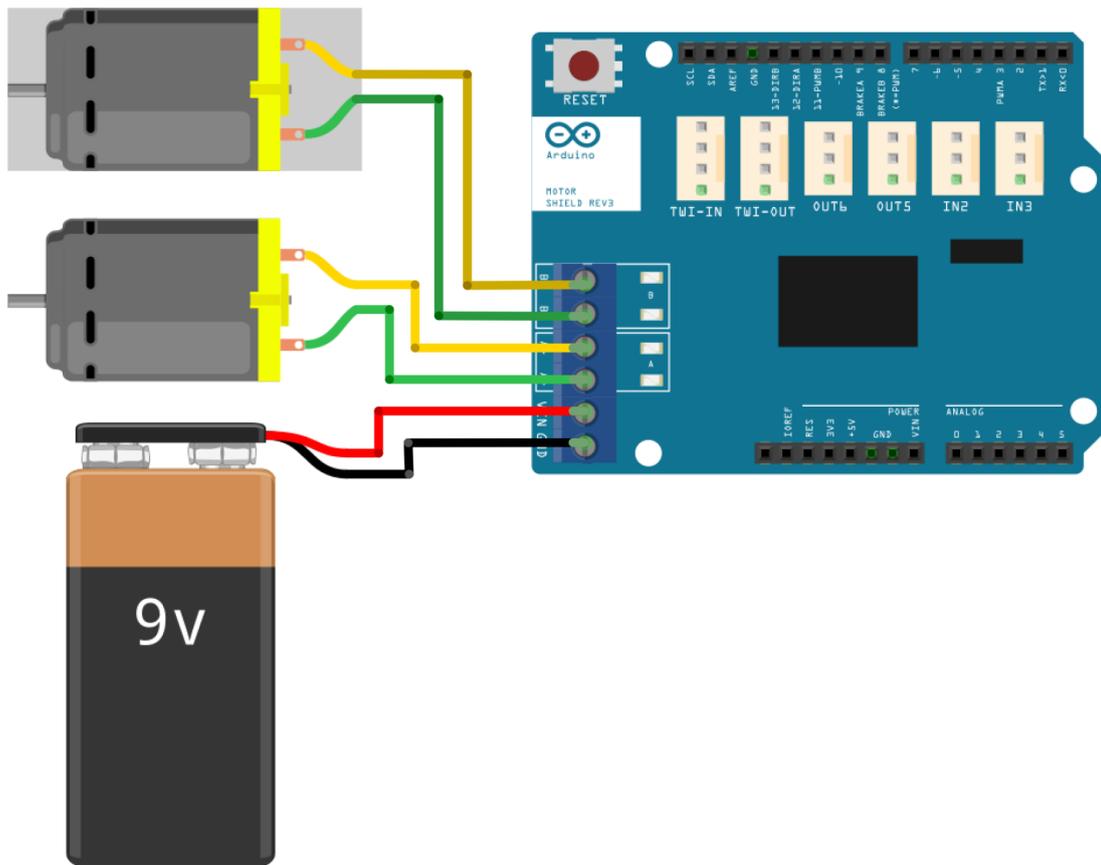
```

```

47  digitalWrite(sensMotB,LOW); // sens 1
48  digitalWrite(vitesseMotB, HIGH); // vitesse maximale
49  delay(2000); // 2 secondes
50
51  digitalWrite(vitesseMotB, LOW); // vitesse maximale
52  //- sens 2
53  digitalWrite(sensMotB,HIGH); // sens 2
54  digitalWrite(vitesseMotB, HIGH); // vitesse maximale
55  delay(1000); // 2 secondes
56  Serial.println(analogRead(intensiteMotB));
57  delay(1000); // 2 secondes
58  digitalWrite(vitesseMotB, LOW); // vitesse maximale
59
60  //---- test vitesse variable moteur A ---
61  for (int i=0; i<=255; i++)
62  {
63    analogWrite(vitesseMotA,i); // PWM croissant
64    delay(50); // pause
65    Serial.println(analogRead(intensiteMotA));
66  }
67  for (int i=0; i<=255; i++)
68  {
69    analogWrite(vitesseMotA,255-i); // PWM décroissant
70    delay(50); // pause
71    Serial.println(analogRead(intensiteMotA));
72  }
73
74  //---- test vitesse variable moteur B ---
75  for (int i=0; i<=255; i++)
76  {
77    analogWrite(vitesseMotB,i); // PWM croissant
78    delay(50); // pause
79    Serial.println(analogRead(intensiteMotB));
80  }
81    for (int i=0; i<=255; i++)
82  {
83    analogWrite(vitesseMotB,255-i); // PWM décroissant
84    delay(50); // pause
85    Serial.println(analogRead(intensiteMotB));
86  }
87  while(1); // stop loop
88 }

```

Le montage à réaliser est le suivant. Ce montage utilise un motor shield Arduino qui vient s'emboîter sur le dessus de la carte Uno (branchez une alimentation extérieure sur les bornes Vin et Ground) :



### Question n°1

A quoi sert un motor shield ?

### Question n°2

Réalisez le montage, téléversez le code, et constatez le fonctionnement.

### Question n°3

En imaginant que chaque moteur est en prise avec une des deux roues motrices droite et gauche d'un modèle réduit de voiture, écrivez le code permettant à la voiture d'avancer en ligne droite, de tourner à droite, de tourner à gauche puis de reculer en ligne droite.

## 2.10. Contrôle de moteur à courant continu avec les drivers L293D et L298N

### 2.10.1. Introduction

#### Objectifs pédagogiques

L'objectif de cette ressource est de piloter des moteurs à courant continu avec des drivers utilisant des ponts H, plus compacts et beaucoup moins coûteux que des Motor Shield Arduino.

Pour réaliser ces deux montages, il faudra vous munir des composants suivants :

- une carte Arduino
- un câble USB
- deux moteurs à courant continu fonctionnant sous 9V, 18W max
- un driver L293D
- un driver L298N
- une alimentation externe de 9V
- des fils Dupont Mâle/Mâle
- une platine de prototypage rapide à trous de type Labdec

## 2.10.2. Contrôle de moteur à courant continu avec les drivers L293D et L298N

---

Le programme ci-dessous permet de contrôler deux moteurs à courant continu avec un contrôleur L293D :

```

1 //initialisation des broches : GA & GB pour le moteur de gauche, DA & DB pour le
  moteur de droite
2 int GA=11,GB=10,DA=5,DB=6;
3
4 void setup() {
5   Serial.begin(9600);
6   pinMode(DA,OUTPUT);
7   pinMode(DB,OUTPUT);
8   pinMode(GA,OUTPUT);
9   pinMode(GB,OUTPUT);
10 }
11
12 void arriere() {
13   digitalWrite(DA,HIGH);
14   digitalWrite(DB,LOW);
15   digitalWrite(GA,HIGH);
16   digitalWrite(GB,LOW);
17 }
18 void avant() {
19   digitalWrite(DA,LOW);
20   digitalWrite(DB,HIGH);
21   digitalWrite(GA,LOW);
22   digitalWrite(GB,HIGH);
23 }
24 void gauche() {
25   digitalWrite(DA,LOW);
26   digitalWrite(DB,HIGH);
27   digitalWrite(GA,HIGH);
28   digitalWrite(GB,LOW);
29 }
30 void droite() {
31   digitalWrite(DA,HIGH);
32   digitalWrite(DB,LOW);
33   digitalWrite(GA,LOW);
34   digitalWrite(GB,HIGH);

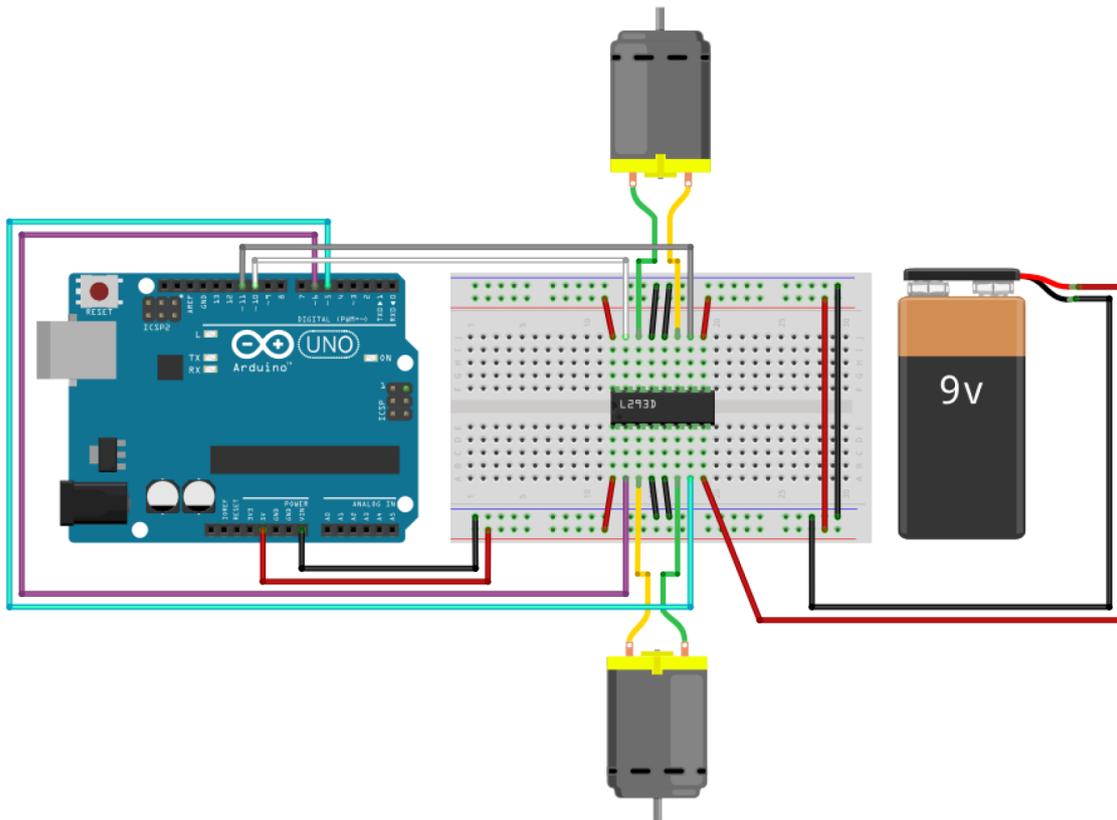
```

```

35 }
36
37 void loop() {
38   avant(); delay(1000);
39   arriere(); delay(1000);
40   gauche(); delay(1000);
41   droite(); delay(1000);
42 }
43

```

Le montage à réaliser est le suivant :



### Question n°1

Comment fonctionne le driver L293D ?

### Question n°2

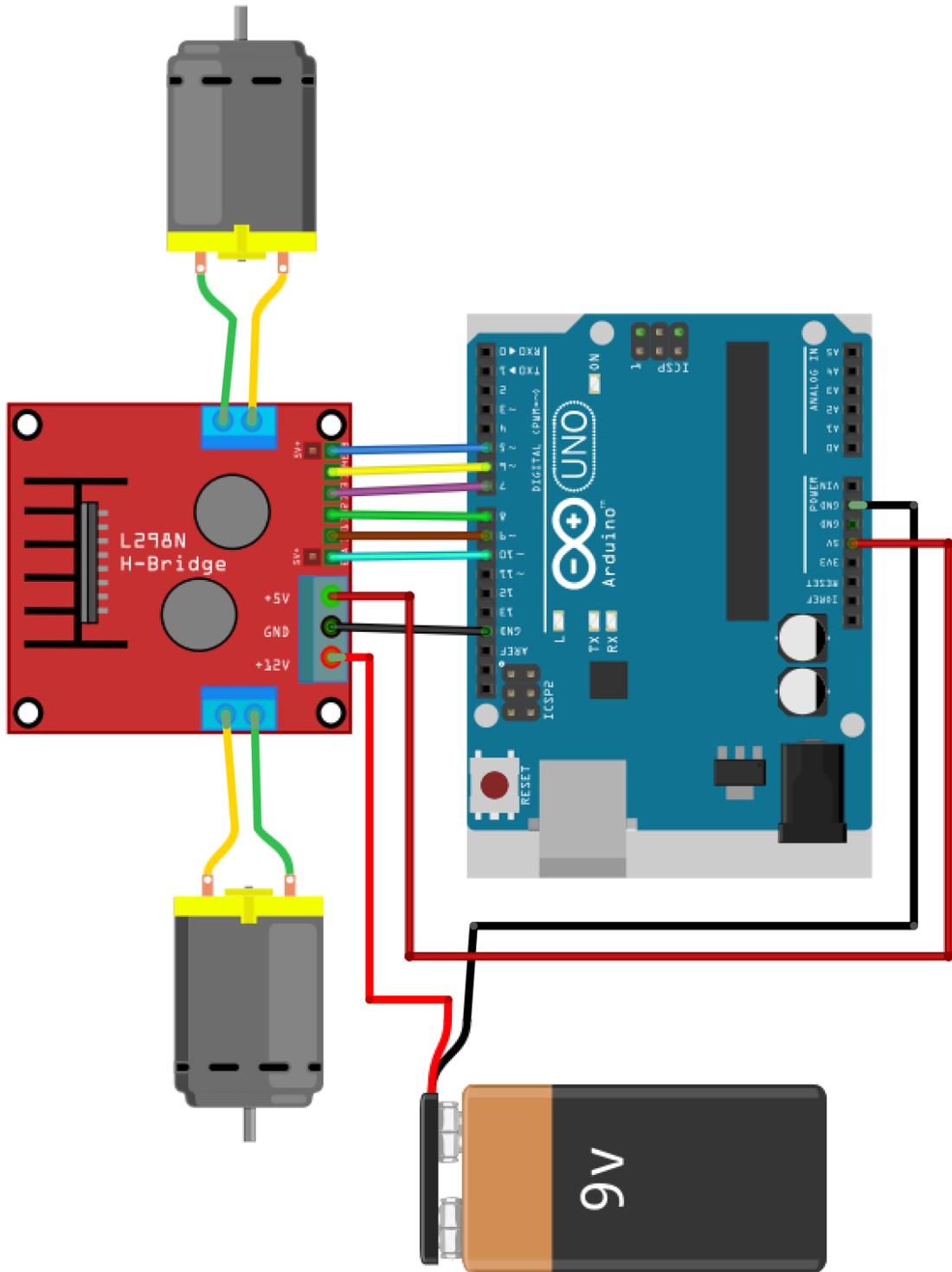
Réalisez le montage, téléversez le code, et constatez le fonctionnement.

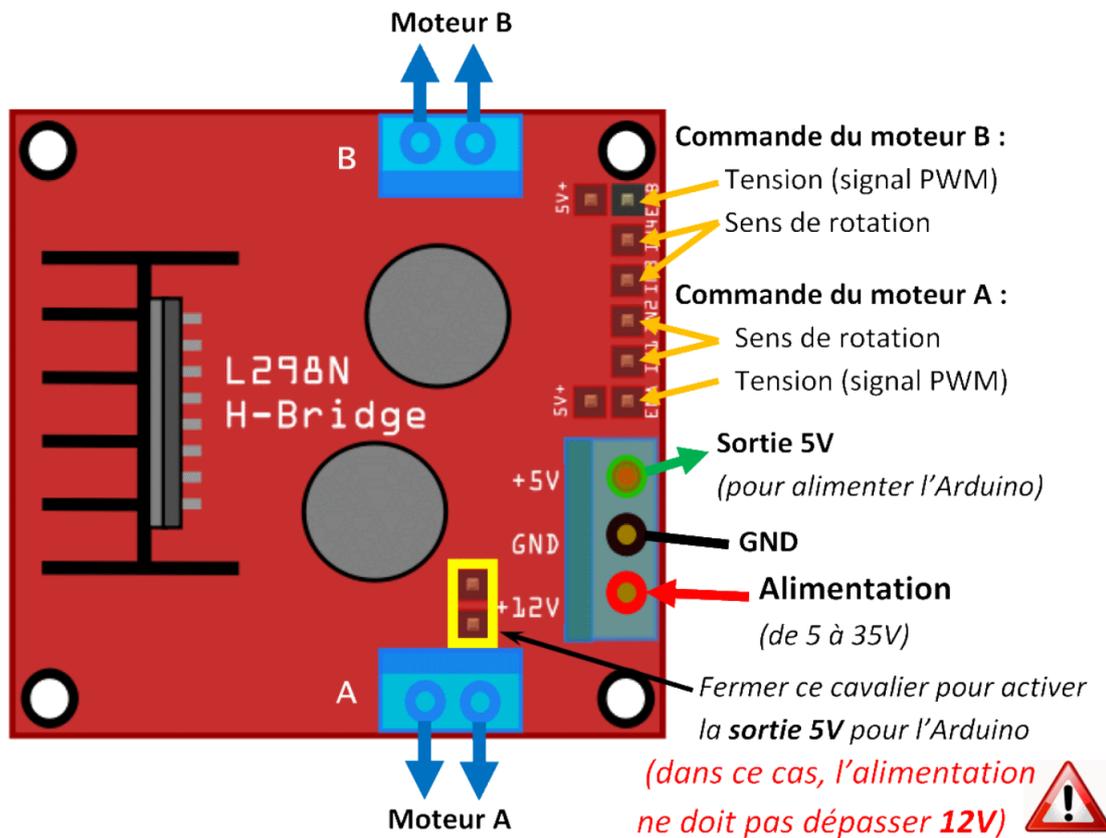
### Question n°3

Modifiez le code et le montage pour qu'un moteur soit piloté par deux boutons poussoir, un pour la rotation dans un sens et l'autre pour la rotation dans le sens contraire.

## 2.10.3.

Vous pouvez également utiliser un driver L298N pour piloter un moteur à courant continu (ce driver peut également contrôler un moteur pas-à-pas), le montage est alors le suivant :





Le programme est alors le suivant :

```

1 //Ports de commande du moteur A
2 int motorA1 = 9; //In2
3 int motorA2 = 8; //In1
4 int enableA = 10; //ENA
5
6 //Ports de commande du moteur B
7 int motorB1 = 7; //In3
8 int motorB2 = 6; //In4
9 int enableB = 5; //ENB
10
11 // Vitesse du moteur
12 int state = 0;
13
14 void setup() {
15
16     pinMode(motorA1, OUTPUT);
17     pinMode(motorA2, OUTPUT);
18     pinMode(enableA, OUTPUT);
19     pinMode(motorB1, OUTPUT);
20     pinMode(motorB2, OUTPUT);
21     pinMode(enableB, OUTPUT);
22
23     Serial.begin(9600);
24 }
25
26 void loop() {
27     if (Serial.available() > 0)
28     {
29         // Lecture de l'entier passé au port série
30         state = Serial.parseInt();

```

```

31
32     if (state > 0) // avant
33     {
34         digitalWrite(motorA1, HIGH);
35         digitalWrite(motorA2, LOW);
36         digitalWrite(motorB1, HIGH);
37         digitalWrite(motorB2, LOW);
38         Serial.print("Avant ");
39         Serial.println(state);
40     }
41     else if (state < 0) // arrière
42     {
43         digitalWrite(motorA1, LOW);
44         digitalWrite(motorA2, HIGH);
45         digitalWrite(motorB1, LOW);
46         digitalWrite(motorB2, HIGH);
47         Serial.print("Arrière ");
48         Serial.println(state);
49     }
50     else // Stop (freinage)
51     {
52         digitalWrite(motorA1, HIGH);
53         digitalWrite(motorA2, HIGH);
54         digitalWrite(motorB1, HIGH);
55         digitalWrite(motorB2, HIGH);
56         Serial.println("Stop");
57     }
58
59     // Vitesse du mouvement
60     analogWrite(enablePin, abs(state));
61 }
62 delay(100);
63 }

```

### Question n°1

En vous aidant d'internet, déterminez la puissance maximale des moteurs pouvant être piloté par ce driver

### Question n°2

A quoi sert la pièce noire dotée d'ailettes directement vissée sur le pont H L298N ?

### Question n°3

Réalisez le montage, téléversez le code et constatez le fonctionnement

## 2.11. Contrôle de moteur pas-à-pas

### 2.11.1. Introduction

## Objectifs pédagogiques

Le but de cette ressource est d'apprendre à piloter un petit moteur pas à pas, pratique et bon marché avec un driver ULN2003.

Pour réaliser ces deux montages, il faudra vous munir des composants suivants :

- une carte Arduino
- un câble USB
- un moteur pas à pas unipolaire 28BYJ-48
- un driver ULN2003
- des fils Dupont Mâle/Mâle

### 2.11.2. Contrôle de moteur pas-à-pas par commande simplifiée

Le programme ci-dessous permet de contrôler un moteur pas-à-pas 28BYJ-48 avec un contrôleur ULN2003 de la manière la plus simple possible mais dans ce cas le moteur ne développera pas tout son couple :

```

1 int IN1 = 8;      // pin digital 8 relié à IN1
2 int IN2 = 9;      // pin digital 9 à IN2
3 int IN3 = 10;     // pin digital 10 à IN3
4 int IN4 = 11;     // pin digital 11 à IN4
5 int temps = 20;  // temps entre les pas, minimum 10 ms.
6
7 void setup(){
8   // tous les pins se configurent comme sorties
9   pinMode(IN1, OUTPUT);
10  pinMode(IN2, OUTPUT);
11  pinMode(IN3, OUTPUT);
12  pinMode(IN4, OUTPUT);
13 }
14
15 void loop(){
16   // 512*4 = 2048 pas pour un tour complet
17   for (int i = 0; i < 512; i++) {
18     digitalWrite(IN1, HIGH); // pas 1
19     digitalWrite(IN2, LOW);
20     digitalWrite(IN3, LOW);
21     digitalWrite(IN4, LOW);
22     delay(temps);
23
24     digitalWrite(IN1, LOW); // pas 2
25     digitalWrite(IN2, HIGH);
26     digitalWrite(IN3, LOW);
27     digitalWrite(IN4, LOW);
28     delay(temps);
29
30     digitalWrite(IN1, LOW); // pas 3
31     digitalWrite(IN2, LOW);
32     digitalWrite(IN3, HIGH);

```

```

33  digitalWrite(IN4, LOW);
34  delay(temps);
35
36  digitalWrite(IN1, LOW); // pas 4
37  digitalWrite(IN2, LOW);
38  digitalWrite(IN3, LOW);
39  digitalWrite(IN4, HIGH);
40  delay(temps);
41  }
42  // pause de 5 secondes
43  digitalWrite(IN1, LOW);
44  digitalWrite(IN2, LOW);
45  digitalWrite(IN3, LOW);
46  digitalWrite(IN4, LOW);
47  delay(5000);
48  }

```

Ce programme peut être réduit en utilisant un tableau pour piloter les sorties du driver :

```

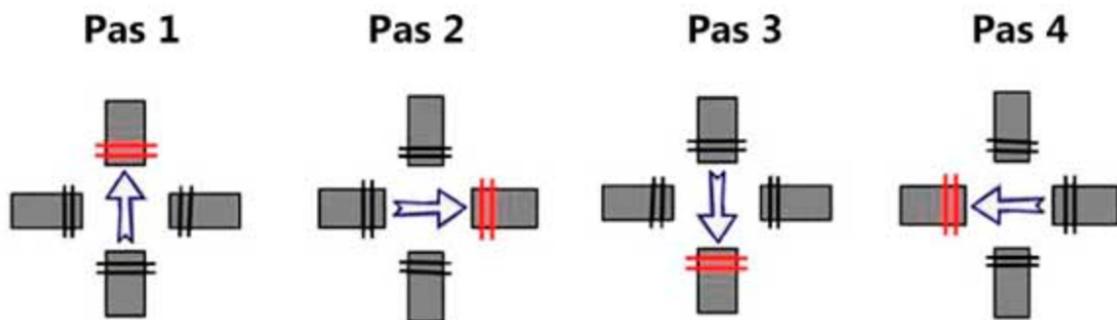
1  int IN1 = 8; // pin digital 8 relié à IN1
2  int IN2 = 9; // pin digital 9 à IN2
3  int IN3 = 10; // pin digital 10 à IN3
4  int IN4 = 11; // pin digital 11 à IN4
5  int temps = 20; // temps entre les pas, minimum 10 ms.
6
7  // Tableau bidimensionnel avec la séquence des pas
8  int pas [4][4] = {
9    {1, 0, 0, 0},
10   {0, 1, 0, 0},
11   {0, 0, 1, 0},
12   {0, 0, 0, 1}
13 };
14
15 void setup(){
16   // tous les pins se configurent comme sorties
17   pinMode(IN1, OUTPUT);
18   pinMode(IN2, OUTPUT);
19   pinMode(IN3, OUTPUT);
20   pinMode(IN4, OUTPUT);
21 }
22
23 void loop(){
24   // 512 cycles * 4 pas par cycle = 2048 pas par tour d'axe en sortie de réducteur
25   for (int i = 0; i < 512; i++){
26     // boucle qui lit le tableau ligne par ligne
27     for (int i = 0; i < 4; i++){
28       digitalWrite(IN1, pas[i][0]);
29       digitalWrite(IN2, pas[i][1]);
30       digitalWrite(IN3, pas[i][2]);
31       digitalWrite(IN4, pas[i][3]);
32       delay(temps);
33     }
34   }
35   // pause
36   digitalWrite(IN1, LOW);
37   digitalWrite(IN2, LOW);
38   digitalWrite(IN3, LOW);
39   digitalWrite(IN4, LOW);
40   delay(temps);
41 }

```

Dans ce cas, les bobines sont pilotées ainsi :

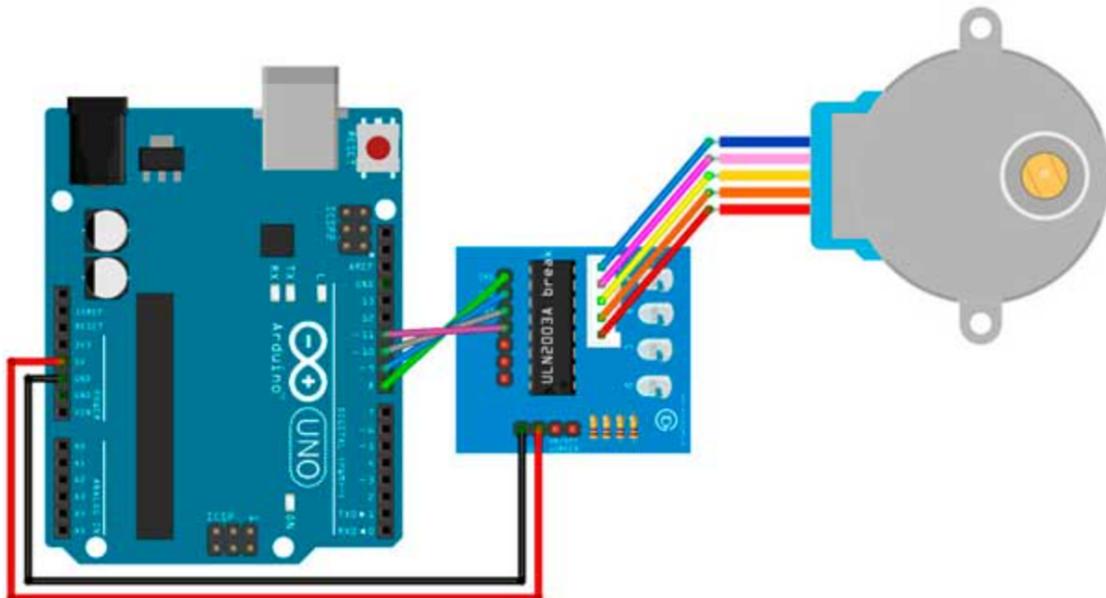
### Séquence pas à pas Unipolaire

| Pas   | A | B | C | D |
|-------|---|---|---|---|
| pas 1 | 1 | 0 | 0 | 0 |
| pas 2 | 0 | 1 | 0 | 0 |
| pas 3 | 0 | 0 | 1 | 0 |
| pas 4 | 0 | 0 | 0 | 1 |



Pour plus de détails : Commander un moteur pas-à-pas [\[https://ledisrupteurdimensionnel.com/arduino/connexion-du-moteur-pas-a-pas-28byj-48-et-le-module-unl2003-a-la-carte-darduino/\]](https://ledisrupteurdimensionnel.com/arduino/connexion-du-moteur-pas-a-pas-28byj-48-et-le-module-unl2003-a-la-carte-darduino/)

Le montage à réaliser est le suivant :



Ce moteur nécessite en théorie une alimentation de 12V pour fonctionner correctement mais vous pouvez l'alimenter avec la carte Arduino si vous le faites tourner à vide. Si vous possédez une alimentation 12V, branchez le + sur le Vin du driver et reliez les masses de l'alimentation et de la carte Arduino pour éviter les phénomènes de masse flottante.

#### Question n°1

Réalisez le montage, téléversez le code et constatez son fonctionnement.

## Question n°2

Modifiez le programme pour faire faire au moteur 3 tours dans un sens et 1 tour dans l'autre.

### 2.11.3. Contrôle de moteur pas-à-pas par commande permettant de développer le couple maximal

Le programme ci-dessous permet de contrôler un moteur pas-à-pas 28BYJ-48 avec un contrôleur ULN2003 de la manière à ce qu'il développe tout son couple :

```

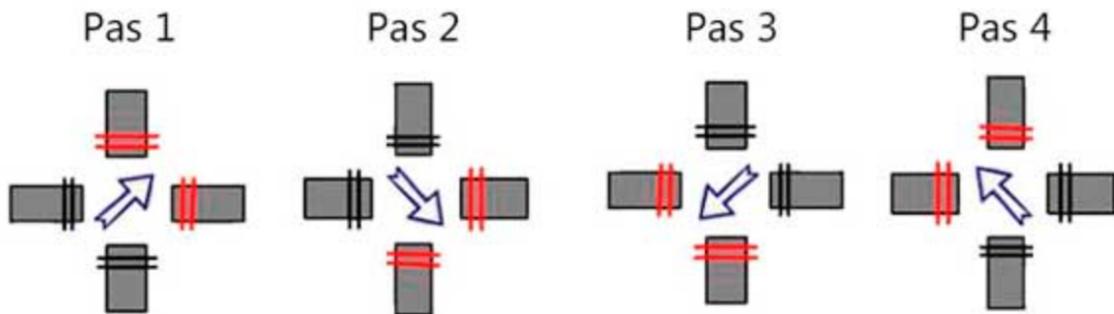
1 int IN1 = 8;      // pin digital 8 relié à IN1
2 int IN2 = 9;      // pin digital 9 à IN2
3 int IN3 = 10;     // pin digital 10 à IN3
4 int IN4 = 11;     // pin digital 11 à IN4
5 int temps = 20;  // temps entre les pas, minimum 10 ms.
6 // Array bidimensionnel indiquant la séquence des pas
7 int pas [4][4] ={
8   {1, 1, 0, 0},
9   {0, 1, 1, 0},
10  {0, 0, 1, 1},
11  {1, 0, 0, 1}
12 };
13
14 void setup(){
15   // tous les pins se configurent comme sorties
16   pinMode(IN1, OUTPUT);
17   pinMode(IN2, OUTPUT);
18   pinMode(IN3, OUTPUT);
19   pinMode(IN4, OUTPUT);
20 }
21
22 void loop(){
23   // 512*4 = 2048 pas
24   for (int i = 0; i < 512; i++){
25     // boucle repasse le array ligne à ligne
26     for (int i = 0; i < 4; i++){
27       // valeurs qu'on va appliquer
28       digitalWrite(IN1, pas[i][0]);
29       digitalWrite(IN2, pas[i][1]);
30       digitalWrite(IN3, pas[i][2]);
31       digitalWrite(IN4, pas[i][3]);
32       delay(temps);
33     }
34   }
35   // pause de 5 secondes
36   digitalWrite(IN1, LOW);
37   digitalWrite(IN2, LOW);
38   digitalWrite(IN3, LOW);
39   digitalWrite(IN4, LOW);
40   delay(temps);
41 }

```

Dans ce cas, les bobines sont pilotées ainsi :

## Séquence pas à pas unipolaire avec deux bobines

| Pas   | A | B | C | D |
|-------|---|---|---|---|
| Pas 1 | 1 | 1 | 0 | 0 |
| Pas 2 | 0 | 1 | 1 | 0 |
| Pas 3 | 0 | 0 | 1 | 1 |
| Pas 4 | 1 | 0 | 0 | 1 |



## Question

Téléversez le code, et constatez le fonctionnement

## 2.11.4. Contrôle de moteur pas-à-pas par commande demi-pas

Le programme ci-dessous permet de contrôler un moteur pas-à-pas 28BYJ-48 avec un contrôleur ULN2003 en étant plus précis (commande par demi-pas) :

```

1 int IN1 = 8;      // pin digital 8 relié à IN1
2 int IN2 = 9;      // pin digital 9 à IN2
3 int IN3 = 10;     // pin digital 10 à IN3
4 int IN4 = 11;     // pin digital 11 à IN4
5 int temps = 20;  // temps entre les pas, minimum 10 ms.
6 // Array bidimensionnel qu'indique la séquence des pas
7 int pas [8][4] = {
8   {1, 0, 0, 0},
9   {1, 1, 0, 0},
10  {0, 1, 0, 0},
11  {0, 1, 1, 0},
12  {0, 0, 1, 0},
13  {0, 0, 1, 1},
14  {0, 0, 0, 1},
15  {1, 0, 0, 1}
16 };
17
18 void setup() {
19   // tous les pins se configurent comme sorties
20   pinMode(IN1, OUTPUT);
21   pinMode(IN2, OUTPUT);
22   pinMode(IN3, OUTPUT);
23   pinMode(IN4, OUTPUT);

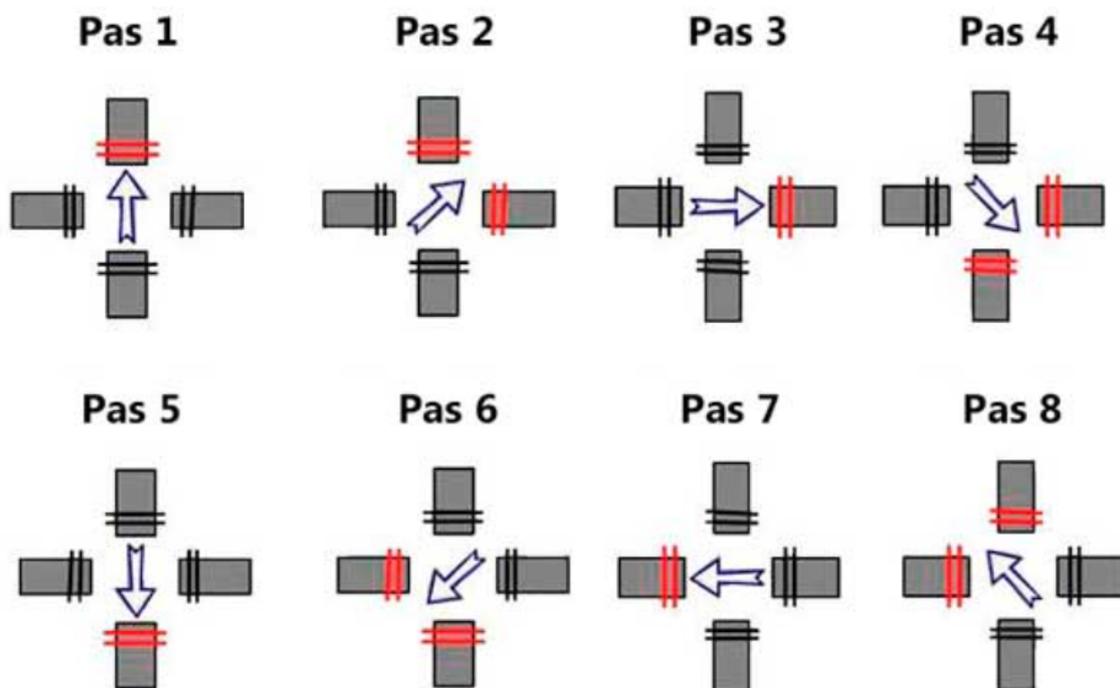
```

```
24 }
25
26 void loop(){
27     // 512*8 = 4096 pas
28     for (int i = 0; i < 512; i++){
29         // boucle repasse le array ligne à ligne
30         for (int i = 0; i < 8; i++){
31             // valeurs qu'on va appliquer
32             digitalWrite(IN1, pas[i][0]);
33             digitalWrite(IN2, pas[i][1]);
34             digitalWrite(IN3, pas[i][2]);
35             digitalWrite(IN4, pas[i][3]);
36             delay(temps);
37         }
38     }
39     // pause de 5 secondes
40     digitalWrite(IN1, LOW);
41     digitalWrite(IN2, LOW);
42     digitalWrite(IN3, LOW);
43     digitalWrite(IN4, LOW);
44     delay(temps);
45 }
```

Dans ce cas, les bobines sont pilotées ainsi :

## Séquence pas à pas avec deux bobines, mi-pas

| Pas   | A | B | C | D |
|-------|---|---|---|---|
| Pas 1 | 1 | 0 | 0 | 0 |
| Pas 2 | 1 | 1 | 0 | 0 |
| Pas 3 | 0 | 1 | 0 | 0 |
| Pas 4 | 0 | 1 | 1 | 0 |
| Pas 5 | 0 | 0 | 1 | 0 |
| Pas 6 | 0 | 0 | 1 | 1 |
| Pas 7 | 0 | 0 | 0 | 1 |
| Pas 8 | 1 | 0 | 0 | 1 |

**Question**

Téléversez le code, et constatez le fonctionnement.

**2.11.5. Contrôle de moteur pas-à-pas avec bibliothèque stepper Arduino**

Le programme précédent peut être considérablement simplifié en utilisant la bibliothèque stepper d'Arduino :

```
1 // charge une bibliothèque qui permet de piloter des moteurs pas à pas simplement
2 #include <Stepper.h>
3
4 // fixe le nombre de pas du moteur à 513 par tour
```

```

5 const int stepsPerRevolution = 513;
6
7 // fixe la vitesse du moteur : 0 = min, 10 = max
8 int motorSpeed=10;
9
10 // initialise la bibliothèque en fonction des broches du driver
11 // dans le cas présent : (step,1N2,1N4,1N3,1N1)
12 Stepper myStepper(stepsPerRevolution, 9, 11, 10, 8);
13
14 void setup()
15 {
16   Serial.begin(9600);
17 }
18
19 void loop()
20 {
21   // définit de la vitesse de révolution du moteur
22   myStepper.setSpeed(motorSpeed);
23   Serial.print("Vitesse : ");
24   Serial.println(motorSpeed);
25
26   // fait tourner le moteur de 513 pas dans le sens trigonométrique
27   Serial.println("Rotation en sens trigonométrique");
28   myStepper.step(stepsPerRevolution);
29   delay(1500);
30
31   // fait tourner le moteur de 513 pas dans le sens horaire
32   Serial.println("Rotation en sens horaire");
33   myStepper.step(-stepsPerRevolution);
34   delay(1500);
35 }

```

### Question n°1

Téléversez le code, et constatez le fonctionnement.

### Question n°2

Modifiez le programme pour demander à l'utilisateur le nombre de pas qu'il souhaite faire faire au moteur et à quelle vitesse pendant 2s. Limitez le programme à un seul sens de rotation.

## 2.12. Contrôle de moteur pas-à-pas avec driver TB6600

### 2.12.1. Introduction

#### Objectifs pédagogiques

L'objectif de cette ressource est de piloter un moteur pas-à-pas qu'on trouve généralement dans les imprimantes 3D avec un driver de qualité

Pour réaliser ces deux montages, il faudra vous munir des composants suivants :

- une carte Arduino
- un câble USB
- un moteur pas à pas bipolaire Nema 17
- un driver TB6600
- une alimentation externe de 12V
- des fils Dupont Mâle/Mâle
- une platine de prototypage rapide à trous de type Labdec

## 2.12.2. Contrôle de moteur pas-à-pas avec driver TB6600

Le programme ci-dessous permet de contrôler un moteur pas-à-pas Nema 17 avec un contrôleur TB6600 :

```

1 const int ena = 2;
2 const int dir = 3;
3 const int pul = 4;
4
5 void setup()
6 {
7   pinMode(ena, OUTPUT);
8   pinMode(dir, OUTPUT);
9   pinMode(pul, OUTPUT);
10  // Pour activer le driver, si HIGH : le moteur passe en rotation libre et le
    driver ne consomme plus de courant
11  digitalWrite(ena, LOW);
12 }
13
14 void loop()
15 {
16  digitalWrite(dir,HIGH); // Définit le sens de rotation
17
18  // Envoie 6400 pulsations pour faire tourner le moteur
19  for(int x = 0; x < 6400; x++)
20  {
21    digitalWrite(pul,HIGH);
22    delayMicroseconds(300);
23    digitalWrite(pul,LOW);
24    delayMicroseconds(300);
25  }
26  delay(1000);
27
28  // Change le sens de rotation
29  digitalWrite(dir,LOW);
30  for(int x = 0; x < 6400; x++)
31  {
32    digitalWrite(pul,HIGH);
33    delayMicroseconds(300);
34    digitalWrite(pul,LOW);
35    delayMicroseconds(300);
36  }

```

```
37 delay(1000);
38 }
```

Pour rappel, la séquence de pilotage des bobines d'un moteur bipolaire est :

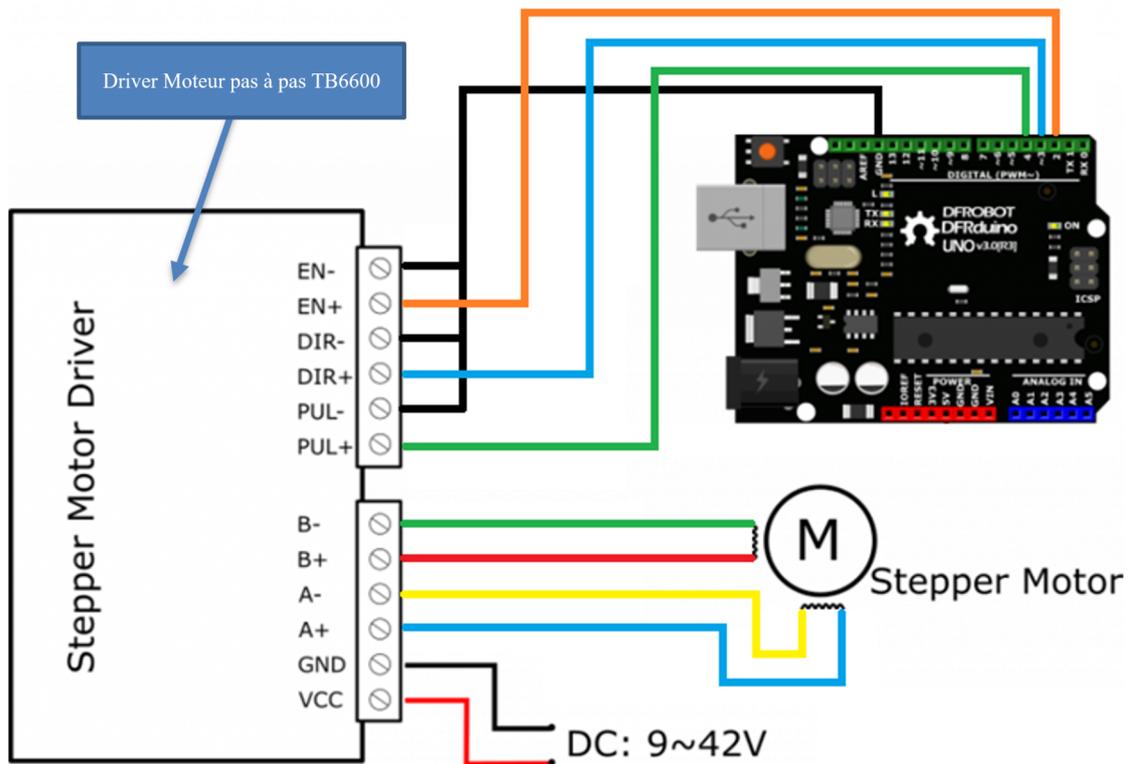
| Impulsion | Bobine A | Bobine A | Bobine B | Bobine B |
|-----------|----------|----------|----------|----------|
| T1        | +        | -        | +        | -        |
| T2        | +        | -        | -        | +        |
| T3        | -        | +        | -        | +        |
| T4        | -        | +        | +        | -        |

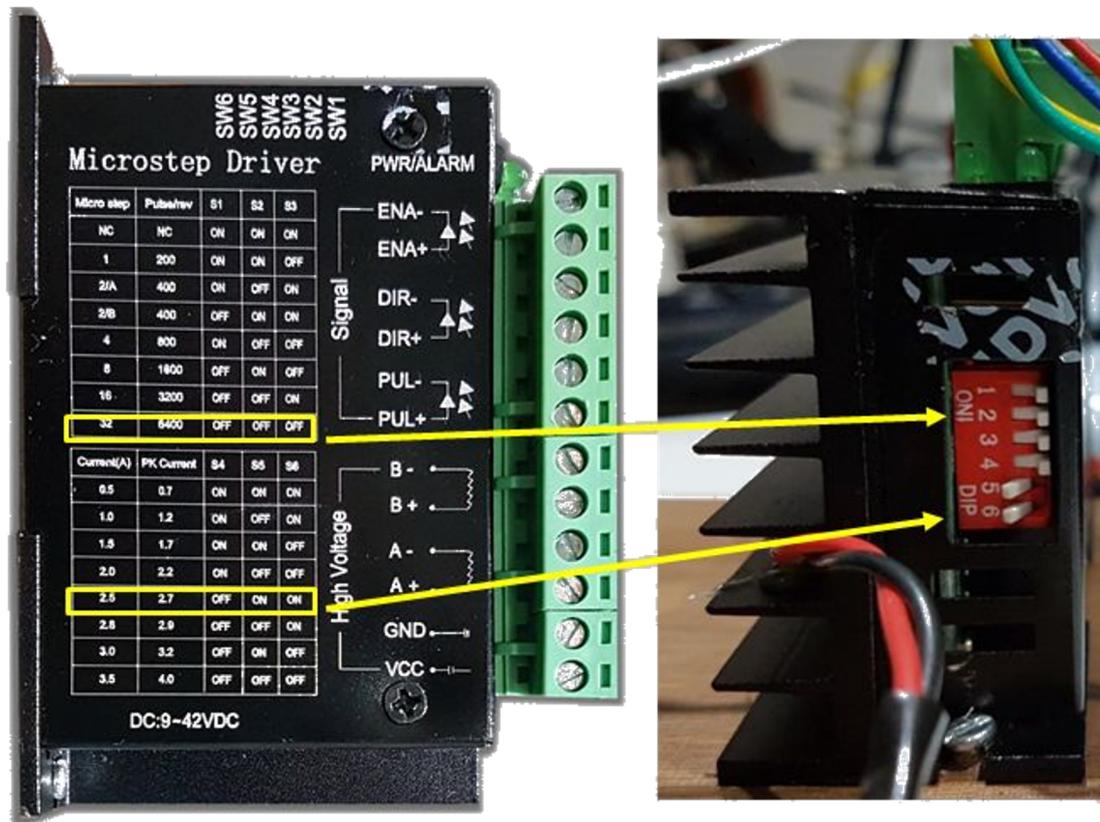
*Alimentation des bobines au cours du cycle*

Le montage à réaliser est le suivant (réglez les micro interrupteurs du driver de sorte à ce que le moteur reçoive un courant de 2A et 6400 pulsations par tour).

Branchez le Vin et le Ground sur une alimentation 12V.

Si vous ne disposez pas d'alimentation 12V, vous pouvez recycler une vieille alimentation de PC : Transformer une alimentation de PC en alimentation d'atelier électronique <https://www.latelierdugeek.fr/2013/05/11/transformer-une-alimentation-de-pc-en-alimentation-datelier/>





### 💡 Truc & astuce

Pour identifier sans la datasheet les paires de bobines sur votre moteur pour savoir comment les relier au A+A- ou B+B-, débranchez le moteur, faites tourner son axe pour ressentir sa résistance naturelle, prenez un des fils, reliez-le à un autre, faites tourner l'axe, si vous sentez une résistance plus grande c'est qu'il s'agit de deux fils de la même bobine (car elle est alors en court-circuit et s'oppose donc à la rotation du moteur), sinon essayez avec un autre fil jusqu'à sentir une résistance. Une fois les paires identifiées, mettez une paire sur les broches A-A+ et une autre sur les broches B-B+.

### Question n°1

Quels sont les avantages de ce driver ?

### Question n°2

Réalisez le montage, téléversez le code, et constatez le fonctionnement

## 3. Le prototypage rapide en mécanique

---

### 3.1. Introduction

---

La fabrication additive, désignée plus communément par les termes « impression 3D », prend une place grandissante dans les process de production industrielle. Cette technologie de production est notamment appréciée pour la rapidité avec laquelle on obtient un objet fonctionnel et pour la simplicité de gestion et de la disponibilité des stocks des matières premières.

### 3.2. Présentation générale de l'impression 3D

---

Il existe deux grandes familles de production mécanique : la fabrication soustractive et additive.



#### Fabrication soustractive

La fabrication soustractive consiste à retirer de la matière comme par exemple avec le tournage et le fraisage.



#### Fabrication additive

La fabrication additive consiste à ajouter de la matière (impression 3D, injection plastique, rotomoulage, etc.). En ce qui concerne l'impression 3D, l'ajout de matière est très souvent réalisée en couche par couche d'épaisseurs constantes.

L'impression 3D vient compléter les procédés de productions traditionnels que sont l'usinage, le moulage, le formage ou encore l'assemblage (soudage, collage, clipsage, clinchage, etc.).

Dans le domaine de l'impression 3D, il existe différentes techniques d'ajout de matière dont le choix dépend des propriétés physiques de l'objet produit (mécaniques, chimiques, magnétiques, etc.) et de ses contraintes géométriques et dimensionnelles.

L'impression 3D est également appelée impression tridimensionnelle, fabrication rapide, fabrication additive (ce qui prête à confusion car d'autres procédés plus anciens fonctionnent par ajout de matière) ou encore prototypage rapide.

Cette technologie englobe différents procédés :

- l'extrusion de matière (le plus démocratisé),
- la photopolymérisation (le plus ancien),
- la projection de matière,
- la projection de liant,
- la lamination de papier,

- la fusion de lit de poudre,
- le dépôt d'énergie dirigée.

Cette méthode de fabrication répond très bien à des besoins :

- de prototypage rapide,
- de fabrication d'objets hautement personnalisés,
- de production de géométries complexes impossibles à produire autrement,
- de plus en plus à la production à grande échelle.

Un autre atout de cette technologie réside dans le fait qu'elle simplifie considérablement la gestion des stocks puisqu'il n'est plus nécessaire de stocker une grande variété de pièces mais simplement de la matière première et une base de données des modèles 3D des pièces à produire.

### Complément

Pour plus d'information : L'impression\_3D<sup>[[https://fr.wikipedia.org/wiki/Fabrication\\_additive](https://fr.wikipedia.org/wiki/Fabrication_additive)]</sup>

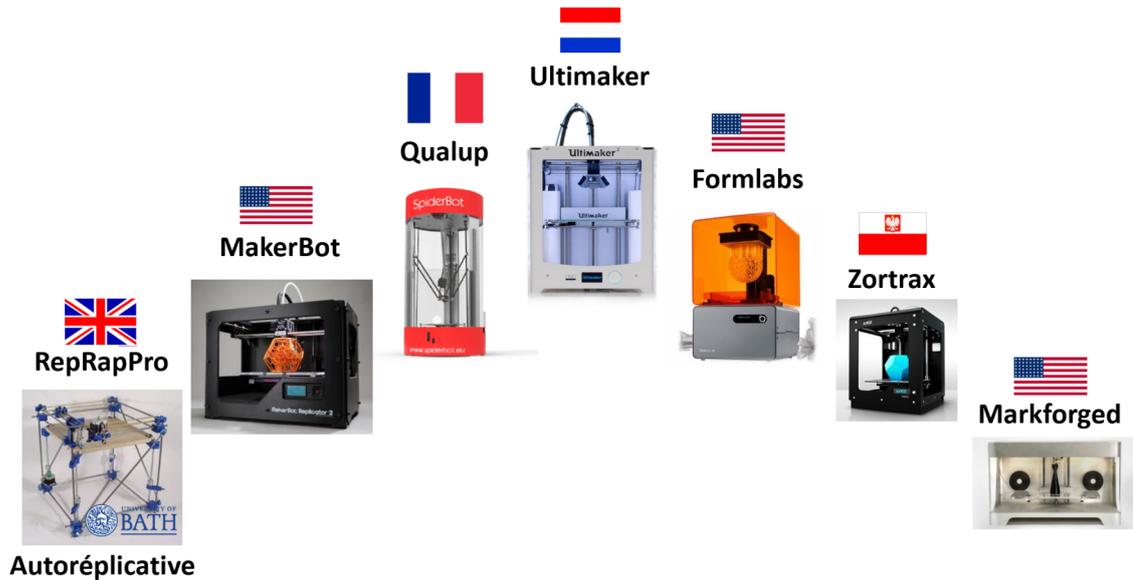
## 3.3. Les secteurs et acteurs sur le marché des imprimantes 3D

---

Dans le domaine des imprimantes 3D, il existe de nombreuses entreprises et profils de clients. Depuis plus de quarante ans, le secteur d'activités a pris de l'ampleur et il devient de plus en plus courant d'avoir une imprimante 3D chez soi ou dans divers entreprises.

Les acteurs des imprimantes 3D sont très majoritairement des entreprises américaines. On peut facilement citer Stratasys et 3DSystems qui ont de très grandes parts de marché pour - respectivement - le dépôt de matière (FDM) et la stéréographie (SLA).

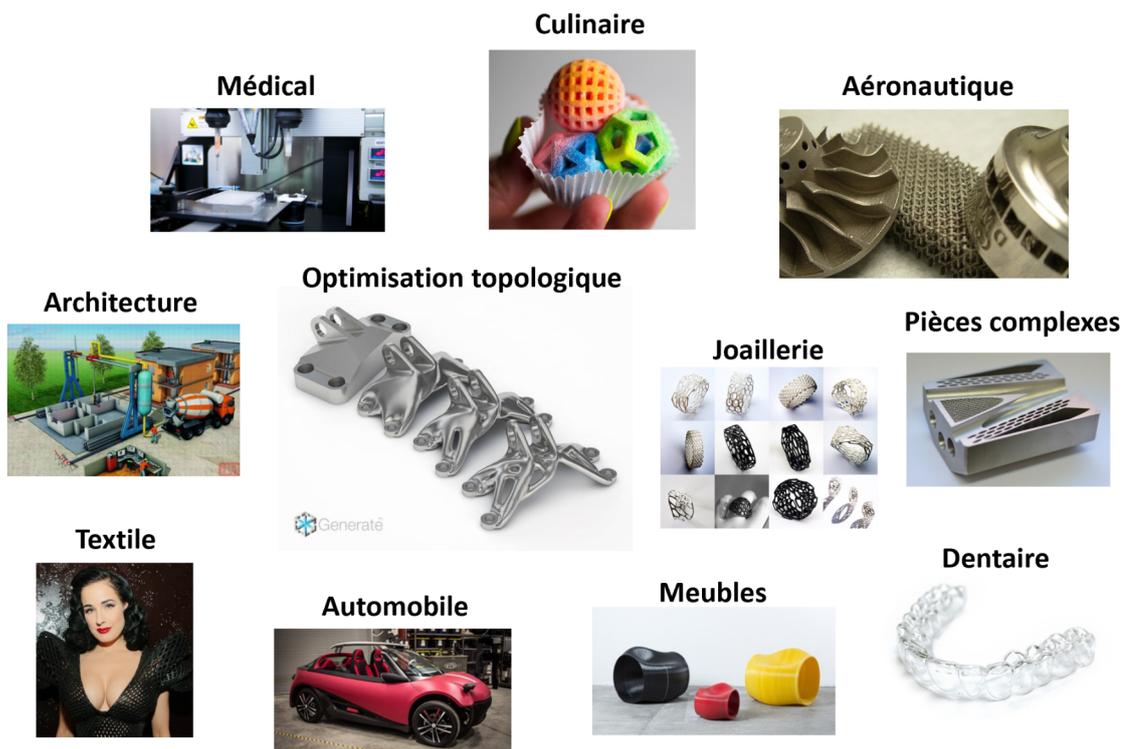
Dans le domaine grand public, les entreprises viennent des 4 coins du globe, il en existe plus d'une centaine. Par exemple, l'entreprise néerlandaise Ultimaker propose des imprimantes 3D fiables dans différents domaines comme l'éducation, l'architecture ou les emballages alimentaires. Actuellement il existe plus de 9 000 brevets sur l'impression 3D, ce chiffre a explosé à partir des années 2000 avec un pic en 2012.



Entreprises de l'impression 3D

## Les applications de l'impression 3D

Le secteur de l'impression 3D touche un large éventail de filières, telles que la mode, le monde culinaire, le médical, l'architecture, l'ameublement, l'automobile, l'aéronautique, la joaillerie, etc.



Les applications les plus courantes sont :

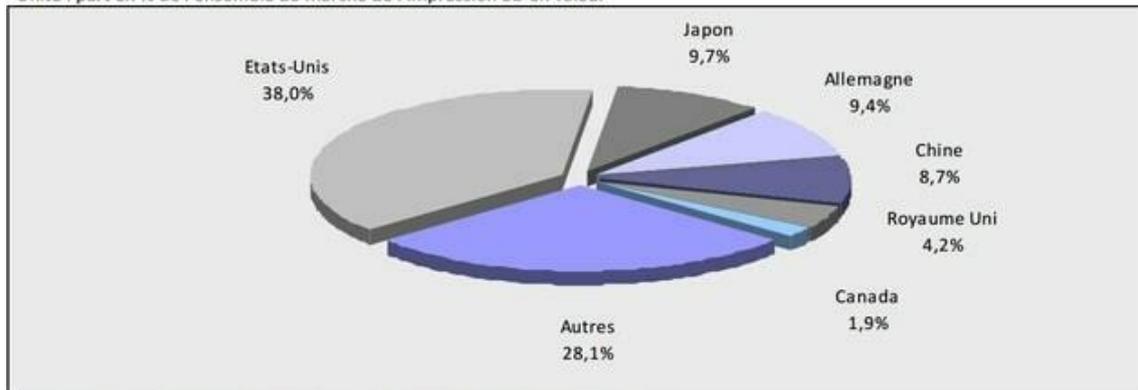
- la visualisation et la validation expérimentale de la conception de produits (prototypage),
- la fabrication d'outillage industriel sur mesure,
- la création de produits hautement personnalisés selon les besoins,
- la réalisation de géométries complexes impossibles à réaliser autrement,

- la fabrication de systèmes polyarticulés ne nécessitant pas de montage,
- la production de composants en petites séries selon les besoins de la production pour optimiser les stocks.

Certaines applications actuellement en développement concernent la création d'organes et de tissus humains, la construction d'édifices architecturaux, la confection de vêtements et la préparation de produits alimentaires.

#### La répartition du marché mondial de l'impression 3D par zone géographique

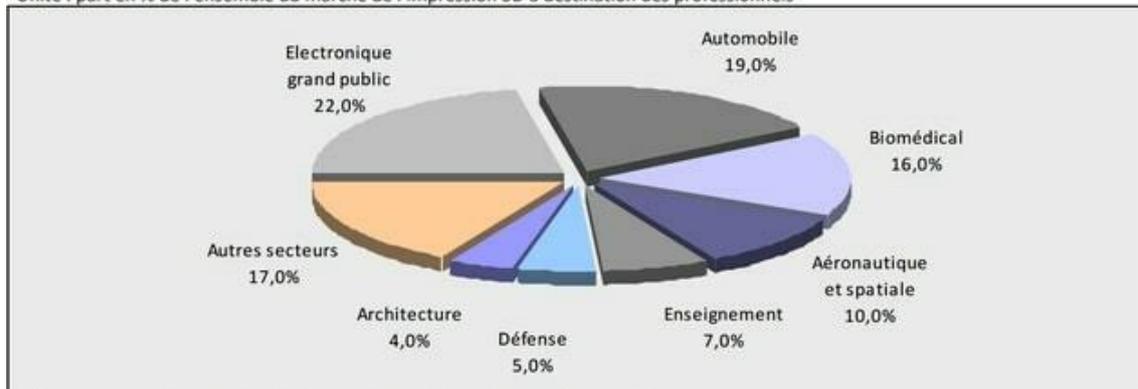
Unité : part en % de l'ensemble du marché de l'impression 3D en valeur



Traitement Xerfi / Source : Crédit Suisse via MonUnivers3D.com, données 2014

#### La répartition du marché mondial de l'impression 3D par débouchés professionnels

Unité : part en % de l'ensemble du marché de l'impression 3D à destination des professionnels



Traitement Xerfi / Source : Crédit Suisse via MonUnivers3D.com, données 2014

## Les avantages de l'impression 3D

Si beaucoup de personnes optent pour l'impression 3D au lieu des techniques artisanales, c'est qu'au fil des années, l'impression 3D propose de plus en plus d'avantages.

Voici les points positifs de l'impression 3D :

- Un temps de fabrication réduit au minimum grâce au nombre de machines et de moules existant qui permettent de réaliser plusieurs impressions en même temps
- La possibilité de faire des designs détaillés et complexes à un prix abordable car il suffit juste de réaliser une seule fois la modélisation
- La possibilité de réaliser des personnalisations sur chaque produit en modifiant le modèle en très peu de temps
- L'autonomie de fabrication car il n'y a pas besoin de superviser l'impression
- Des temps de fabrication réduits au minimum (pas de préparation préalable de la machine, peu de pré-traitement requis, pas de moule, ...)

- La suppression des contraintes liées à la fabrication par usinage
- L'autonomie de fabrication : des machines qui fonctionnent sans intervention humaine et qui ne nécessitent que peu de connaissance pour être utilisées
- Le recyclage des matières en utilisant des machines spécifiques qui reprennent des matériaux déjà utilisés en impression pour les transformer en filament, poudre ou liquide
- La fabrication à base de nouveaux matériaux comme des mélanges de matériaux (polymère + poudre de bois, ou + céramique, ou + poudre métallique, etc.)
- La possibilité de faire une impression 3D à distance sans intervention humaine (ex : sur d'autres planètes !)
- Toujours plus de nouvelles opportunités de conception : constructions cinématiques complexes pré-assemblées, matériaux allégés par le biais d'un maillage complexe 3D, fabrication de pièces non usinables (ex : moules d'injection plastique avec circuit de refroidissement complexe intégré dans la masse), conception de carters en plastique plus fonctionnels, par exemple en apporter des additifs les rendant plus isolant thermiquement ou d'un point de vue électro-magnétique.

Exemple d'objet poly-articulé réalisé sans assemblage :



Exemple d'objet très résistant et très léger : la micro-lattice, très utile pour créer des ailes d'avions :



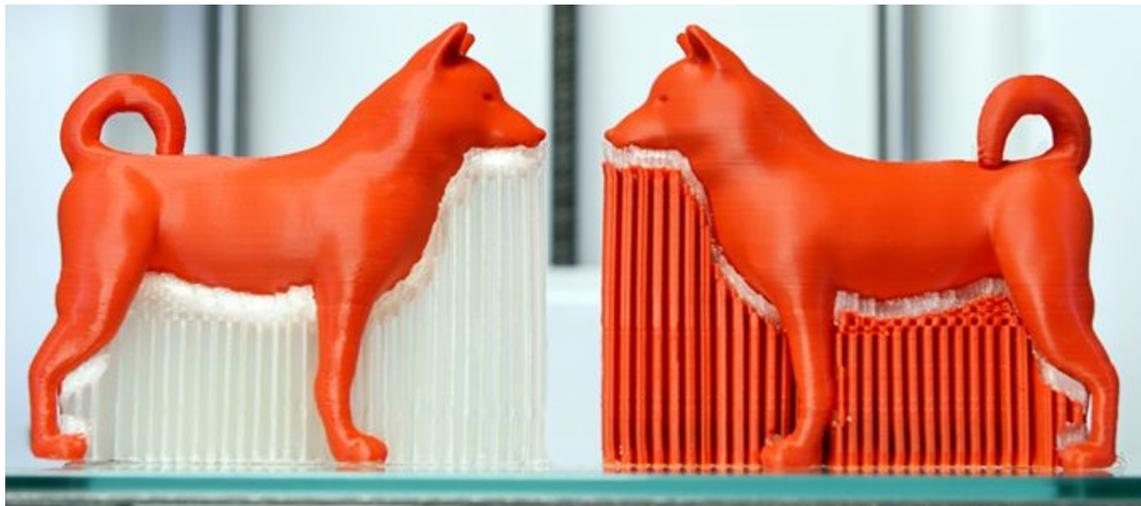
## Les inconvénients de l'impression 3D

Malgré cette énumération de points positifs, il existe des points négatifs à l'impression 3D.

Voici les points négatifs à l'impression 3D :

- Ce n'est pas adapté à la production en série d'un grand nombre de pièces,
- La précision géométrique et dimensionnelle des pièces obtenues (de l'ordre du dixième de mm) est nettement inférieure à celle de la fabrication soustractive (de l'ordre du micron : soit 100 fois plus petit),
- L'état de la surface des pièces est généralement rugueux,
- Les propriétés mécaniques sont souvent non isotropes et non homogènes,
- Les pièces produites à base de polymères ont une moindre grande résistance mécanique que leurs équivalents réalisés par des process de plasturgie conventionnels en raison notamment de la fragilité inter-filaire et inter-couche qui sont des sources d'amorce de rupture,
- Pour certains procédés, il est nécessaire de s'appuyer sur un support d'impression pour les géométries en porte à faux qui sont généralement à l'origine de défauts de surface

Exemple de pièces réalisées avec support :



Pour celle-ci, il s'agit d'un support soluble, qui est plus coûteux que la matière traditionnelle et qui génère aussi des défauts du fait du manque d'adhérence entre cette matière et la matière de la pièce. À droite, le support soluble est uniquement déposé à proximité de la pièce, pour faire des économies.



Pour celle-ci, il s'agit de la même matière que la pièce, celle-ci génère des imperfections de surface et peut être parfois très difficile à retirer.

## 3.4. Les technologies de l'impression 3D

---

### 3.4.1. Introduction

---

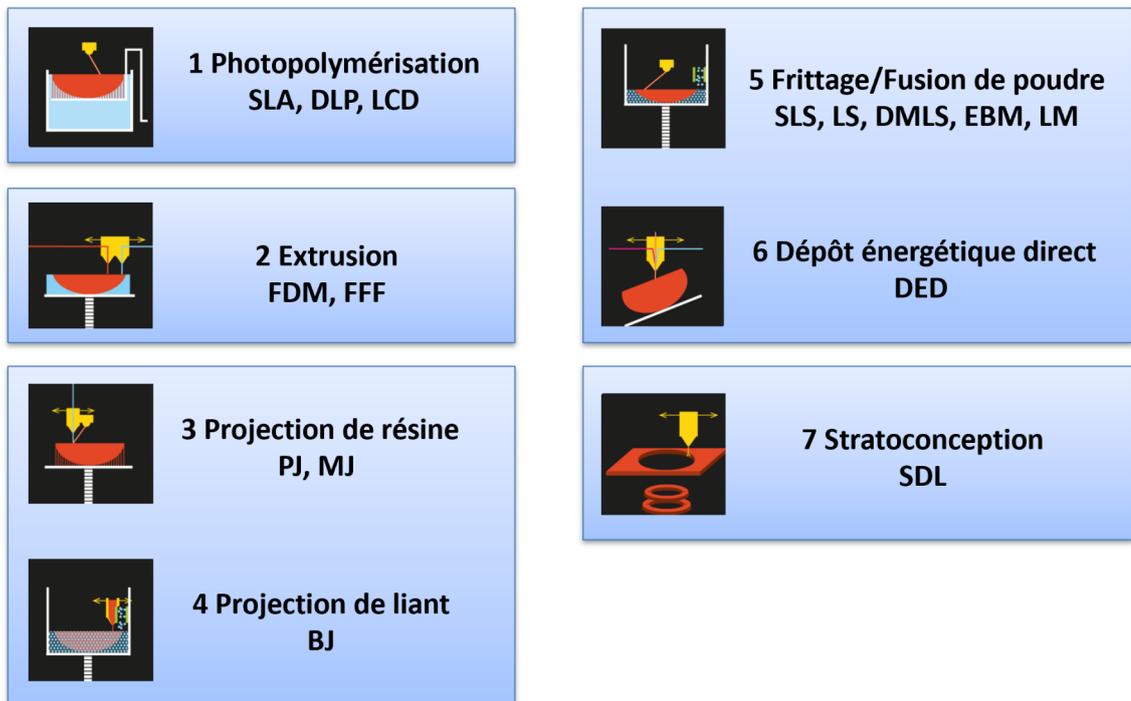
Il existe 7 procédés de fabrication différents, répartis en 5 grandes familles :

1. La photopolymérisation (SLA, DLP, LCD)
2. L'extrusion (FDM, FFF)
3. La projection :
  - La projection de résine (PJ, MJ)
  - La projection de liant (BJ)

#### 4. La fusion de poudres :

- La fusion d'un lit de poudre (SLS, LS, DMLS, EBM, LM)
- Le dépôt énergétique direct (DMP)

#### 5. La stratoconception



*Les catégories d'impression 3D*

Pour plus d'informations sur les procédés : Les processus [\[https://3dprintingindustry.com/3d-printing-basics-free-beginners-guide#04-processes\]](https://3dprintingindustry.com/3d-printing-basics-free-beginners-guide#04-processes) & Les procédés d'impression 3D [\[http://www.lesnumeriques.com/imprimante-3d/impression-3d-differents-procedes-a1876.html\]](http://www.lesnumeriques.com/imprimante-3d/impression-3d-differents-procedes-a1876.html)

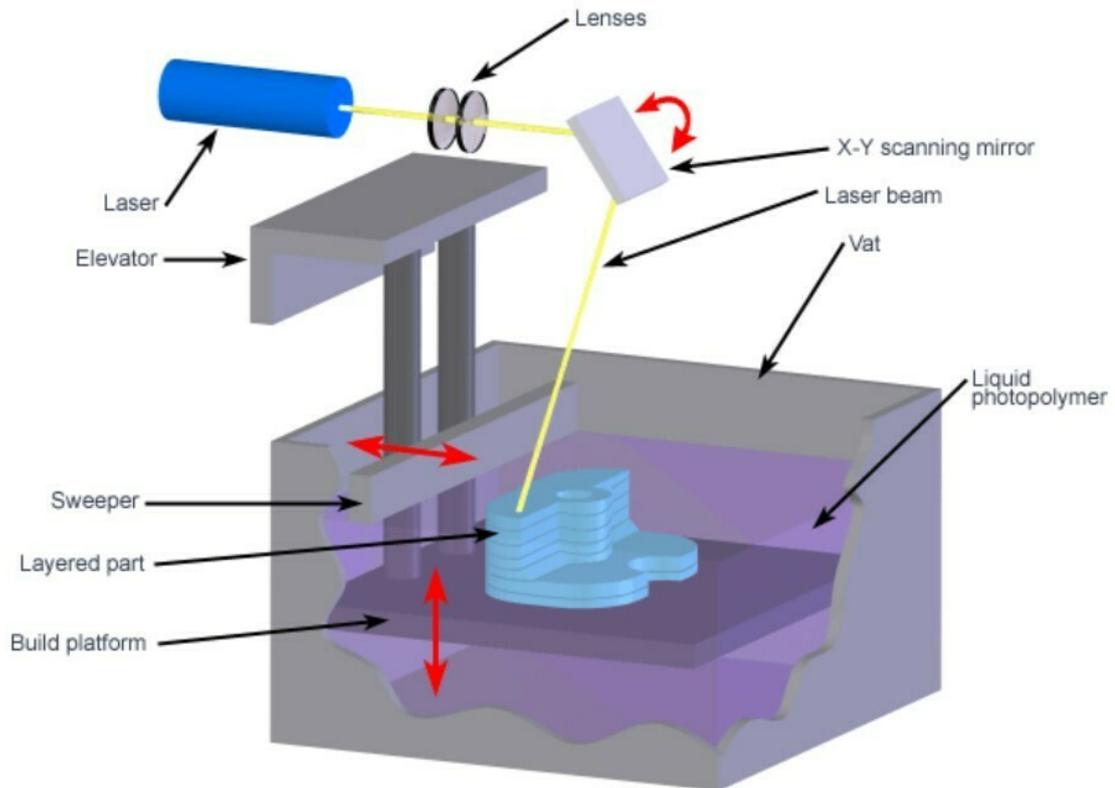
### 3.4.2. La photopolymérisation

La photopolymérisation est une technique venant de la stéréolithographie (aussi appelée écriture en relief). La technique consiste à durcir de la résine liquide photosensible au moyen de lumière UV, d'un laser ou de chaleur. L'impression est réalisée couche par couche.

#### La stéréolithographie ou SLA

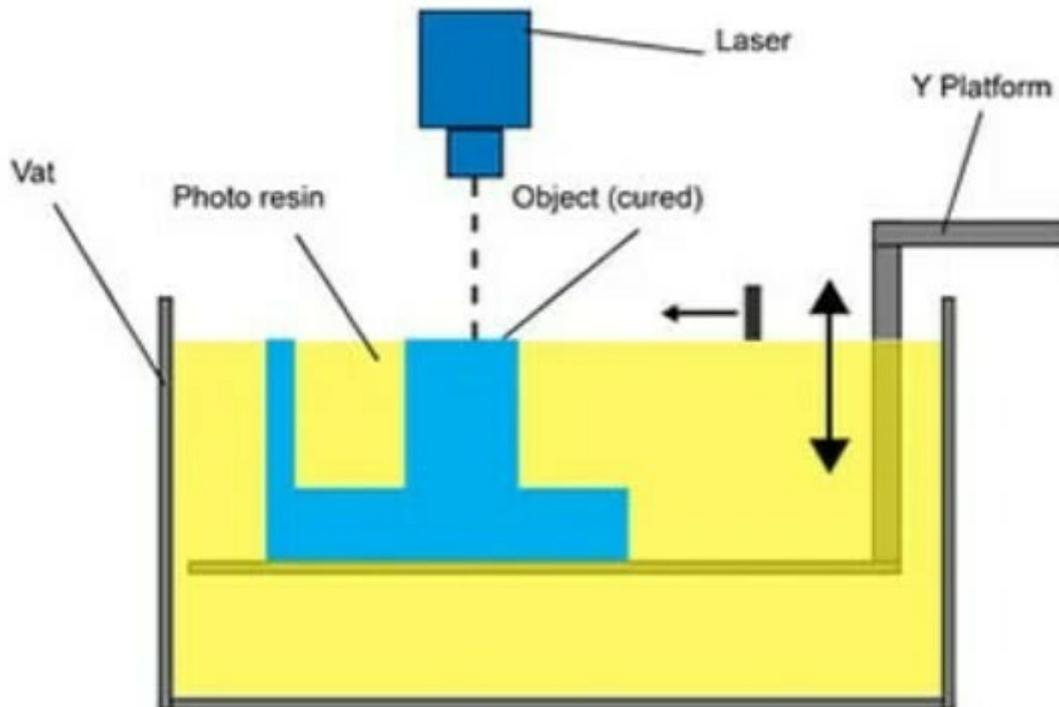
Le procédé SLA est apprécié pour sa capacité à produire des pièces et prototypes isotropes et étanches de haute précision avec des détails fins et une finition de surface lisse

La machine de stéréolithographie (SLA) contient une cuve de résine époxy liquide qui est durcie couche par couche (de 0,10 à 0,15 millimètre) par un faisceau d'ultraviolet ou d'un laser. Une fois terminé, l'objet est sorti de la cuve et la résine non-durcie en surface de la pièce est nettoyée par vibration dans un liquide ou dissoute par un solvant. Selon le type de résine, il faut réaliser une cuisson dans un four UV pour achever de durcir l'objet.



*Principe de la Stéréolithographie*

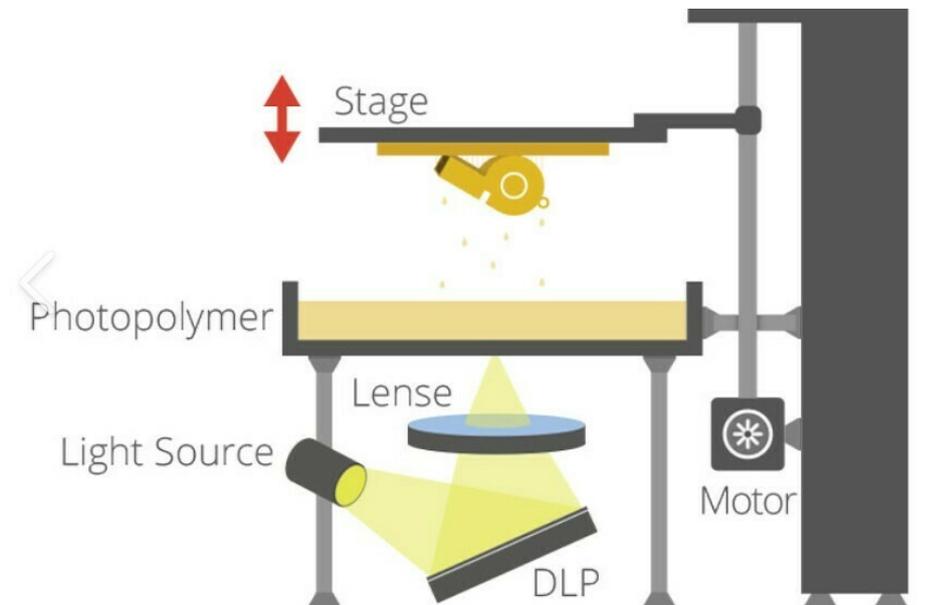
Ce procédé durcit la résine liquide située en surface du bain par le biais d'un laser de grande précision qui balaie la surface en suivant un parcours qui couvre entièrement la surface correspondant à la strate de pièce à produire. Le plateau sur lequel repose la pièce produite descend progressivement au fur et à mesure que les couches sont solidifiées.



*Principe de la photopolymérisation par laser direct*

## La photopolymérisation par DLP

Le traitement numérique de la lumière ou le Digital Light Processing (DLP) est une technique de photopolymérisation qui utilise comme source lumineuse un projecteur à écran numérique qui par un jeu de miroirs et de concentrateur par lentille va diriger la lumière vers le fond du bac où la résine liquide sera polymérisée couche par couche. Le DLP durcit une surface de couche plus importante en un passage que le laser qui ne durcit la matière que point par point. Dans ce procédé, la première couche se situe dans le fond du bac et le plateau qui maintient la pièce monte d'une épaisseur de couche à chaque fois qu'une couche est solidifiée.



Principe de photopolymérisation par projection numérique DLP

Ce procédé est plus économique dans la mesure où il ne faut que très peu de résine pour travailler, seul le fond du bac doit être rempli alors que dans le procédé SLA, l'ensemble du volume du bac doit être rempli de résine. Généralement, dans les machines DLP, le bac se remplit au cours du temps à partir d'un réservoir de résine, ce qui évite à la résine d'être inutilement gaspillée. En effet, la résine étant sensible à la lumière et à la chaleur, le simple fait de la faire sortir de son contenant la dégrade très rapidement.

## La photopolymérisation par LCD

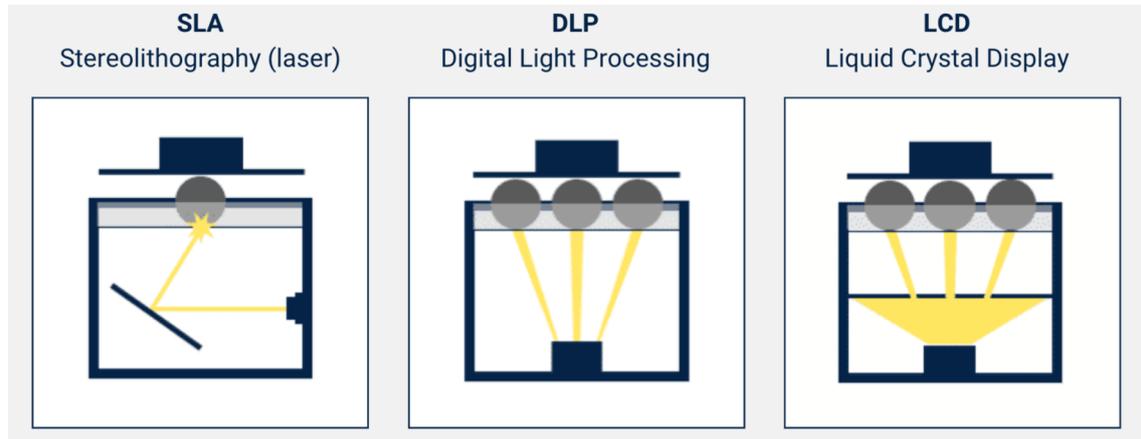
La source lumineuse est dans ce cas un écran Liquid Crystal Display (LCD), la projection d'une image très lumineuse pour durcir entièrement une surface, donc une couche, en un seul passage contrairement au laser de la stéréolithographie qui durcit la couche en suivant un parcours.

Les cristaux liquides, orientés par champs magnétiques, laissent passer ou non une lumière située en rétro éclairage à la manière d'ombres chinoises. La taille des pixels de l'écran influence directement la finesse des pièces produites.

Cette technique est très rapide mais le choix des résines est plus limité que le SLA traditionnel. Cette source peut aussi fonctionner avec des LEDs pour une plus grande précision.

## Synthèse

Les trois procédés de photopolymérisation :



Principes des trois procédés

Les avantages de la photopolymérisation sont :

- la très grande précision des pièces produites,
- la capacité à produire des pièces isotropes,
- la rapidité d'exécution en comparaison aux techniques de FDM.

Les inconvénients sont :

- la faible variété de résines disponibles pour ces machines,
- les pièces qui en résulte sont fragiles,
- le nettoyage du bac et des composants de la machine est compliqué et nécessite des produits nocifs (le bac est généralement jeté au bout de plusieurs réalisations),
- les pièces produites doivent être systématiquement nettoyées en sortant de la machine et passée au four à UV, ce qui demande des opérations manuelles supplémentaires,
- le procédé est relativement lent, même en utilisant le DLP ou le LCD.

### Complément

Pour plus d'informations : Les procédés d'impression 3D [<http://www.lesnumeriques.com/imprimante-3d/impression-3d-differents-procedes-a1876.html>]

## 3.4.3. L'extrusion

### Le FDM ou FFF

L'extrusion de filament, plus connue sous l'acronyme FDM (Fused Deposition Modeling), a vu le jour il y a une trentaine d'années grâce à S. Scott Crump, cofondateur de la société Stratasys.

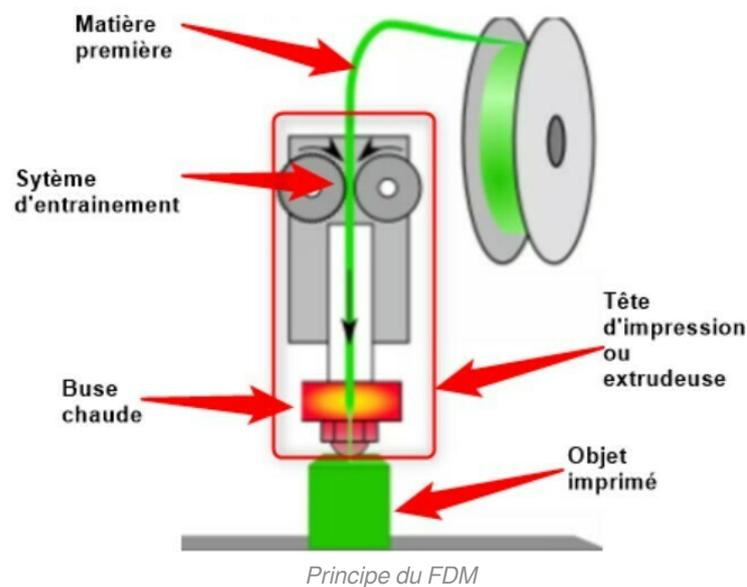
FDM est une marque déposée par Stratasys, c'est pourquoi d'autres appellations sont été créées par des entreprises concurrentes. C'est notamment le cas du FFF (Fused Filament Fabrication), utilisé dans la communauté RepRap sous licence libre.

Il s'agit du procédé utilisé par une écrasante majorité d'imprimantes 3D dites personnelles. Son principe de fonctionnement est simple : un matériau, souvent présenté sous forme de bobine, passe à travers

une buse d'extrusion chauffée entre 170 et 260 °C selon la nature du matériau ou de l'environnement de travail. Le matériau fond et est déposé couche par couche par une buse effectuant des translations selon 2 axes sur un support chauffé ou adhérent. La finesse des couches varie en fonction du matériel et des réglages : 0,02 mm étant souvent la valeur la plus utilisée pour son bon ratio qualité/temps de réalisation/coût. Une fois qu'une couche est terminée, le plateau d'impression descend ou la tête d'impression remonte pour produire la couche suivante, et ainsi de suite.

Le plateau d'impression peut être chauffé pour pallier la déformation due au choc thermique subi par le plastique, en effet, celui-ci passe de plus de 200 °C à la température ambiante quasi instantanément. On peut également trouver des imprimantes FDM dotées de 2 buses d'extrusion. Cela permet entre autres d'imprimer un objet dans deux couleurs différentes ou deux matériaux différents : par exemple pour déposer du support soluble à l'eau en complément de la matière de la pièce.

En plus d'être très abordable, facile d'accès, relativement propre, en comparaison aux autres procédés, le FDM permet d'utiliser une grande variété de matériaux et de couleurs.



Il existe des imprimantes 3D FDM professionnelles qui peuvent produire des pièces plus résistantes mécaniquement en insérant des fibres de carbone, de verre, ou d'aramide dans le matériau lors de l'impression. C'est le cas des imprimantes Markforged.

## Le stylo 3D

Le stylo 3D est avant un outil de retouche ou de création artistique. C'est un outil adapté aux enfants comme aux adultes. La pointe du stylo contient une buse qui permet de faire fondre et d'extruder du plastique par une motorisation intégrée. L'admission de fil se trouve à l'arrière du stylo.



Le stylo 3D

Les avantages sont :

- utile pour retoucher des défauts d'impression ou pour réaliser des premiers concepts sans passer par la modélisation 3D,
- permet de recycler les chute de filaments (les fins de bobine notamment).

L'inconvénient de ce produit est qu'il est un outil d'artisanat et non un moyen de production.

### 3.4.4. La projection

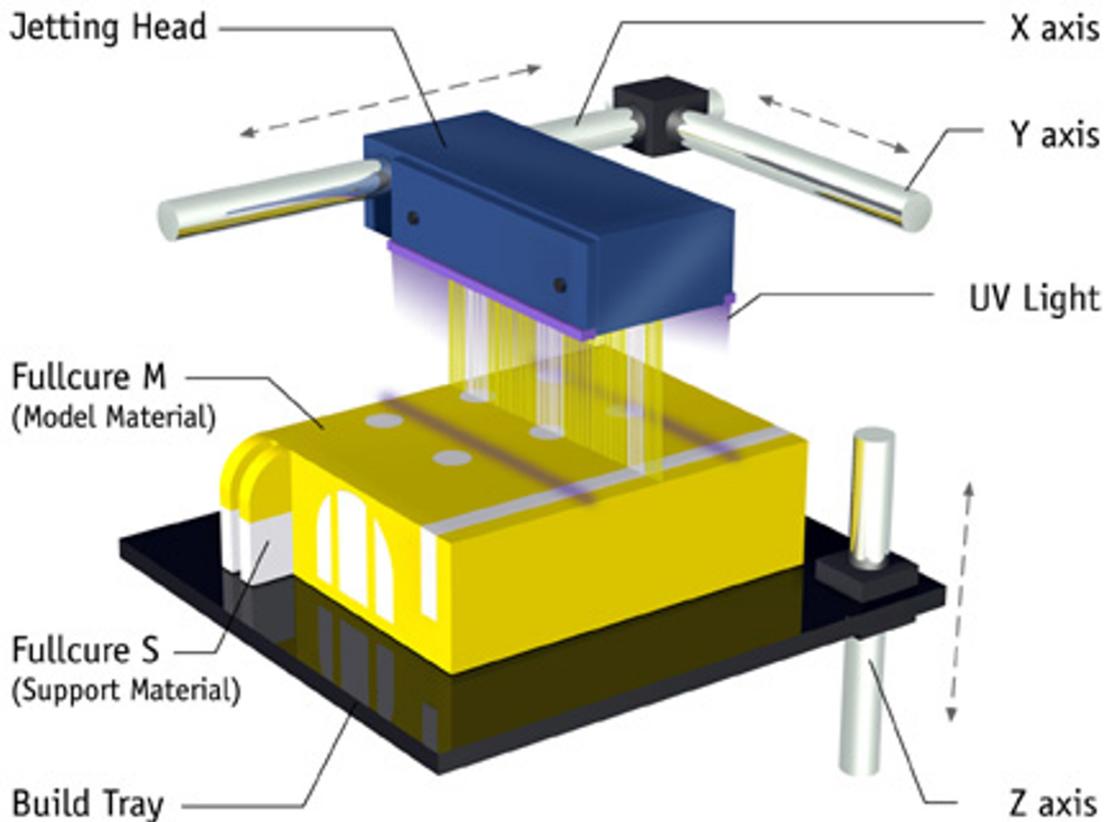
#### La projection de résine (PJ, MJ)

Le Photopolymer Jetting (PJ), ou Material Jetting (MJ), s'appuie sur la photopolymérisation, tout comme le SLA.

Dans ce procédé, des gouttelettes de matériau sont projetées sur le plateau d'impression selon un motif qui correspond à la tranche de la pièce réalisée. Après chaque jet, le polymère est solidifié grâce à une lumière UV. Les gouttelettes étant liquides, un autre matériau liquide durcissant rapidement en refroidissant est également projeté autour du motif que forme la résine photodurcissable pour la contenir le temps qu'elle durcisse (ce qui peut prendre un temps important). Cet autre matériau sert aussi de

support pour les géométries en porte à faux. Ce matériau peut ensuite être fondu dans un four à 70° ou dissout dans de l'eau selon sa nature, pour être évacué et libérer la pièce produite en résine photodurcissable.

L'avantage de cette technologie, est qu'elle ne nécessite pas d'autre post-traitement que la dissolution du matériau de support, contrairement au SLA qui nécessite du nettoyage, du durcissement au four UV et de l'ébavurage.

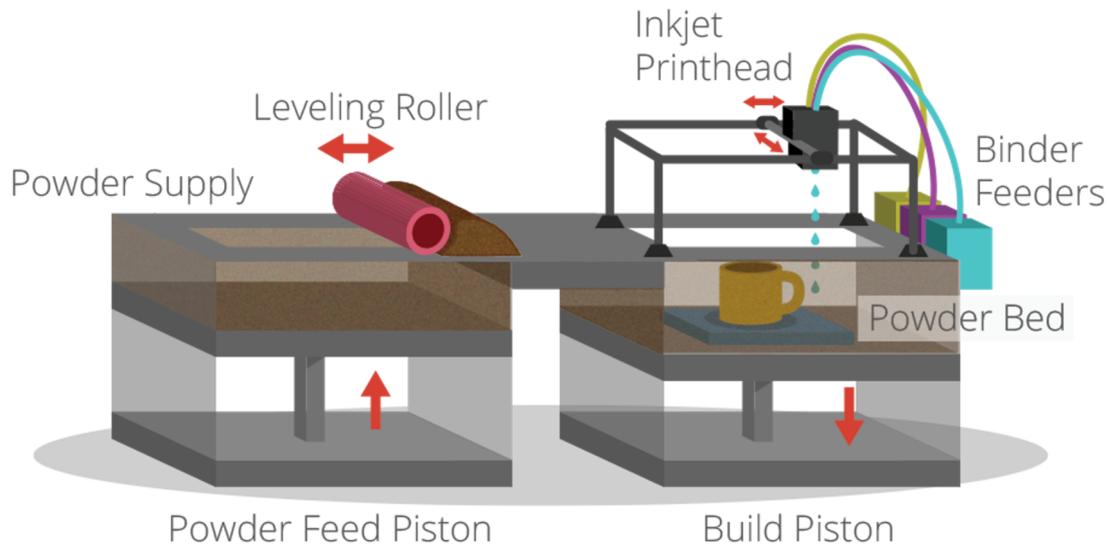


Principe du PJ MJ

## La projection de liant (BJ)

La projection de liant (ou Binder Jetting) est une technique qui utilise la projection d'un liant sur un bain de poudre pour l'agglomérer et former les couches de la pièce. Une fois une couche durcie, le plateau descend d'une hauteur de couche, un chariot vient déposer une nouvelle couche de poudre non agglomérée sur la précédente, du liant est à nouveau projeté et ainsi de suite jusqu'à obtention de la pièce finale.

## Inkjet: Binder Jetting



Principe du BJ

Les avantages de ce procédé sont :

- il ne nécessite aucun support puisque la poudre assure cette fonction,
- il peut utiliser un grand nombre de matériaux dont la céramique qui peut ensuite être recuite à très haute température pour obtenir une pièce très dure,
- il permet de colorer les pièces avec une palette complète de couleurs en intégrant les pigments dans le liant,
- la poudre non utilisée est parfaitement réutilisable dans un autre cycle.

Le principal inconvénient est que les pièces qui sortent de la machine sont très fragiles et nécessitent un post traitement, soit un recuit, soit un apport de vernis durcissant pour améliorer leurs performances mécaniques.

### 3.4.5. La fusion de poudre

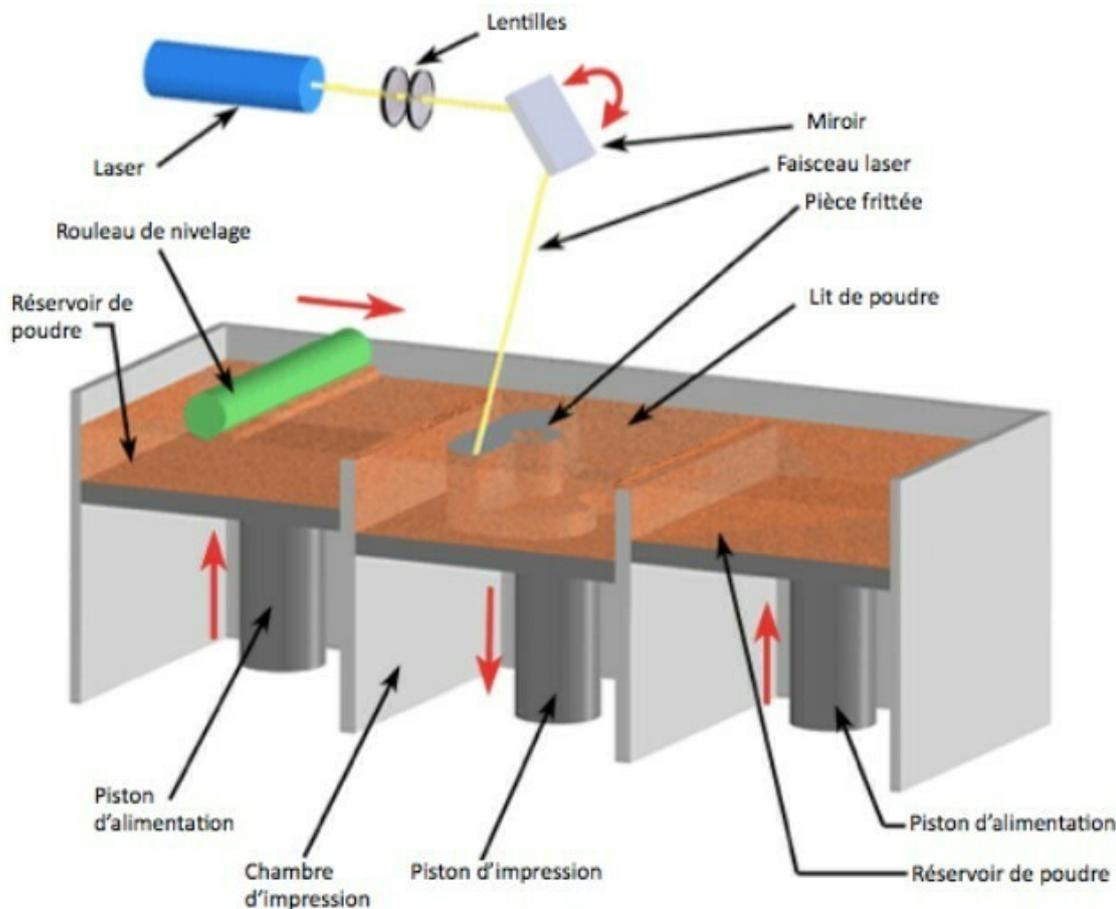
#### Le frittage par laser (SLS, LS, LM, DMLS, LM)

En impression 3D, le frittage consiste à fusionner des particules de poudre par apport de chaleur concentrée. Il existe plusieurs procédés : le Laser Sintering (LS), le Selective Laser Sintering (SLS), le Laser Melting (LM), le Direct Metal Laser Sintering (DMLS) qui utilisent tous des lasers.

Il s'agit de poudre de plastique, de céramique, de verre ou de métal.

Tout comme pour la projection de liant, ces procédés utilisent un bac qui contient le matériau sous forme de poudre, dont les grains mesurent de 50 à 200 microns de diamètre, mais la différence réside dans le fait que l'agglomération des particules se fait par fusion grâce à une source de chaleur concentrée en un point donné (par laser ou flux d'électrons) qui se déplace pour réaliser le motif de la couche. Une fois la couche durcie, un rouleau vient déposer une fine couche (0,1 mm) de poudre sur la précédente et alors la source de chaleur la durcie. L'opération se répète pour chaque couche jusqu'à obtention de la pièce. Une fois le processus terminé, on retire l'objet puis on le débarrasse des restes de poudre non fusionnée.

par air pulsé et un sablage est ensuite envisageable pour lisser l'état de surface.



Principe du SLS

Les avantages de ce procédé sont :

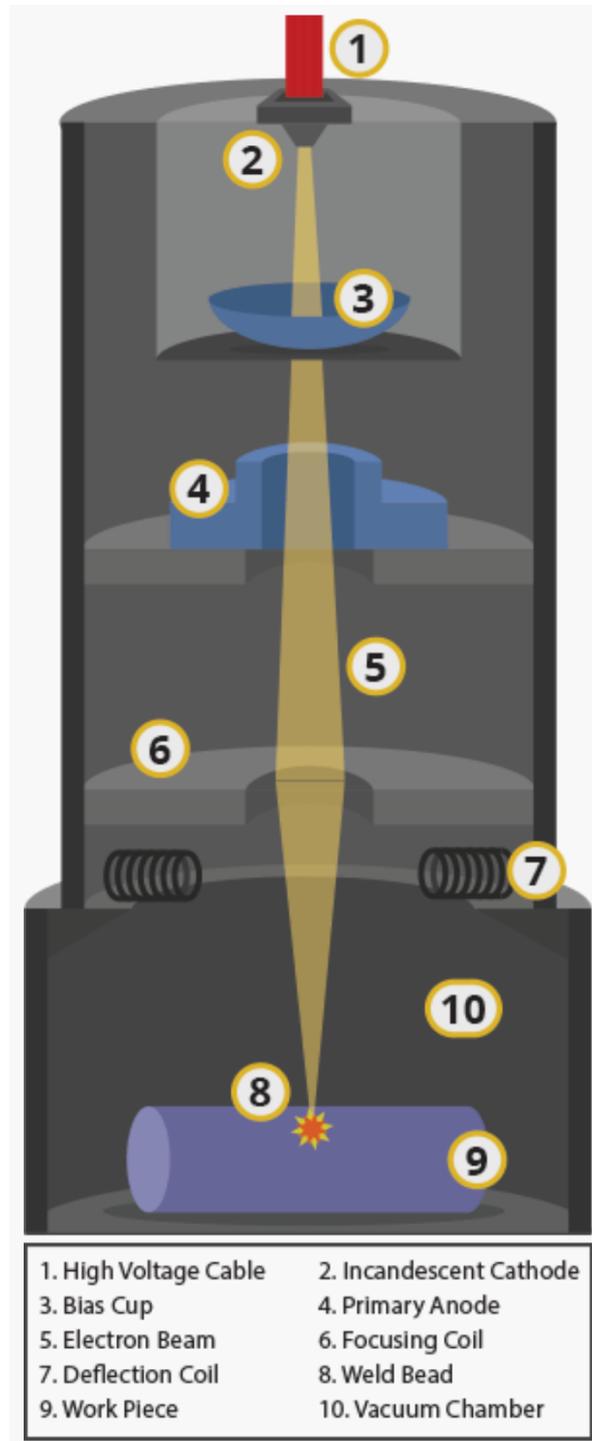
- la possibilité de produire des pièces très solides,
- une grande variété de matériaux pouvant être mise en forme,
- il ne nécessite aucun support puisque la poudre assure cette fonction,
- la poudre non utilisée est parfaitement réutilisable dans un autre cycle.

Les inconvénients sont :

- les pièces ont un état de surface rugueux,
- les poudres sont très volatiles et extrêmement dangereuses pour l'homme compte tenu de leur finesse, notamment en cas d'inhalation, il faut donc placer la machine en salle blanche et doter les opérateurs de combinaisons complètes.

## Le frittage par flux d'électron (EBM)

Le Electron Beam Material (EBM) qui diffère des autres procédés à base de laser est plus astreignant puisqu'il nécessite des conditions de vide particulières pour être exécuté. Il est toutefois bien plus efficace pour produire des pièces en alliance métallique très dense et trouve son utilité dans les domaines du médical et de l'aéronautique.

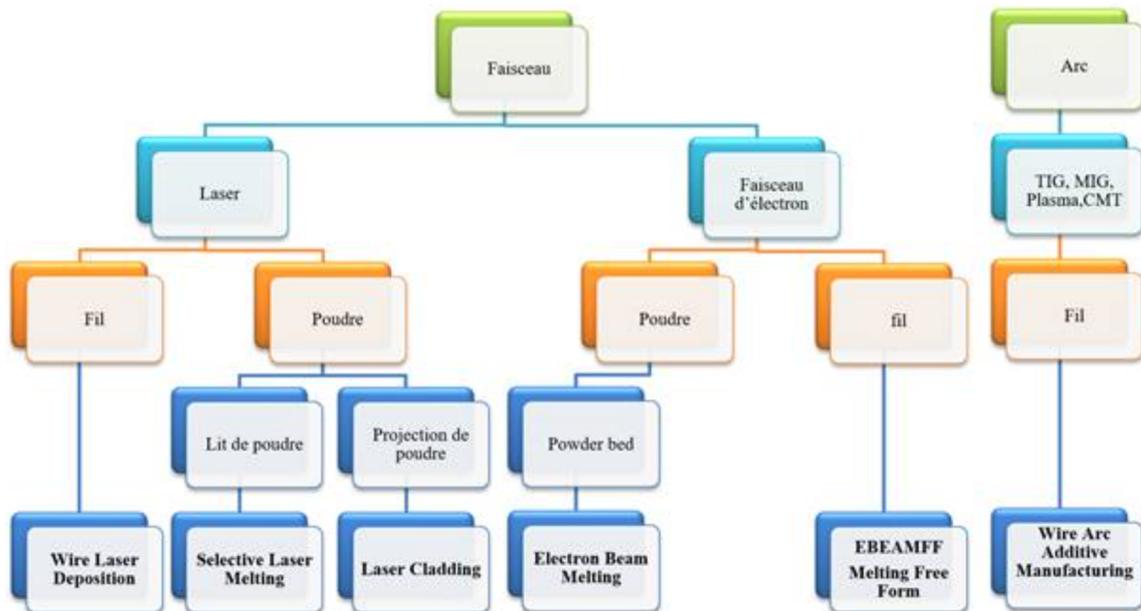


Principe de l'EBM

## Le dépôt d'énergie directe (DED)

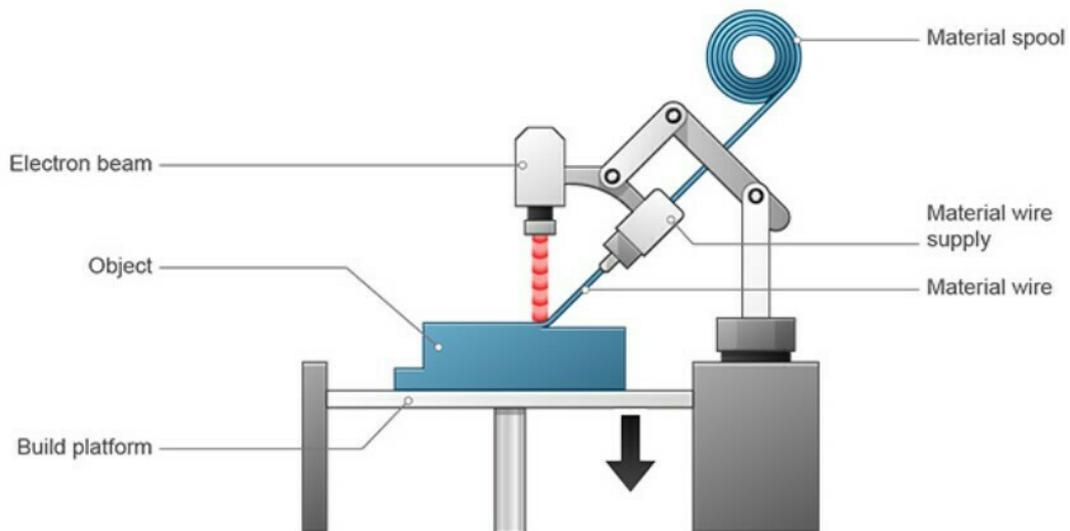
L'impression 3D par dépôt de matière sous énergie concentrée ou DED en anglais (Directed Energy Deposition) est une technique qui consiste à déposer du matériau (poudre ou fil) sur le plateau à l'aide d'une buse montée sur un bras polyarticulé (généralement 4 axes). Lors du dépôt, une source de chaleur fait fondre le matériau simultanément à l'aide d'un laser, un plasma ou d'un faisceau d'électron. Cette procédure est répétée jusqu'à la fin de la fabrication de l'objet. Presque tous les matériaux soudables sont imprimables. Le diamètre des fils est généralement compris entre 1 et 3mm et la taille des particules entre 50 et 150 microns.

Les différentes catégories de DED sont :



Les familles du DED

Elle est souvent référencée sous d'autres noms tels que Laser Engineered Net Shaping (LENS), Direct Metal Deposition (DMD), Electron Beam Additive Manufacturing (EBAM), CLAD (Construction Laser Additive Directe) en fonction de l'application ou de la méthode.



Principe du DED

Les avantages de cette technique sont :

- les pièces produites sont presque aussi résistantes que des pièces forgées,
- elle offre la possibilité de réparer des objets endommagés par regarnissage,
- elle nécessite moins de matière que les procédés d'usinage traditionnels, notamment pour produire des formes creuses telles que des tuyères de réacteur par exemple.

Les inconvénients sont :

- le nombre limité d'alliages compatibles,
- le prix des machines est exorbitant,

- la surface des pièces produite est généralement granuleuse et la dispersion dimensionnelle est grande, il est donc nécessaire de les réusinier pour maîtriser les états de surface et les cotes.

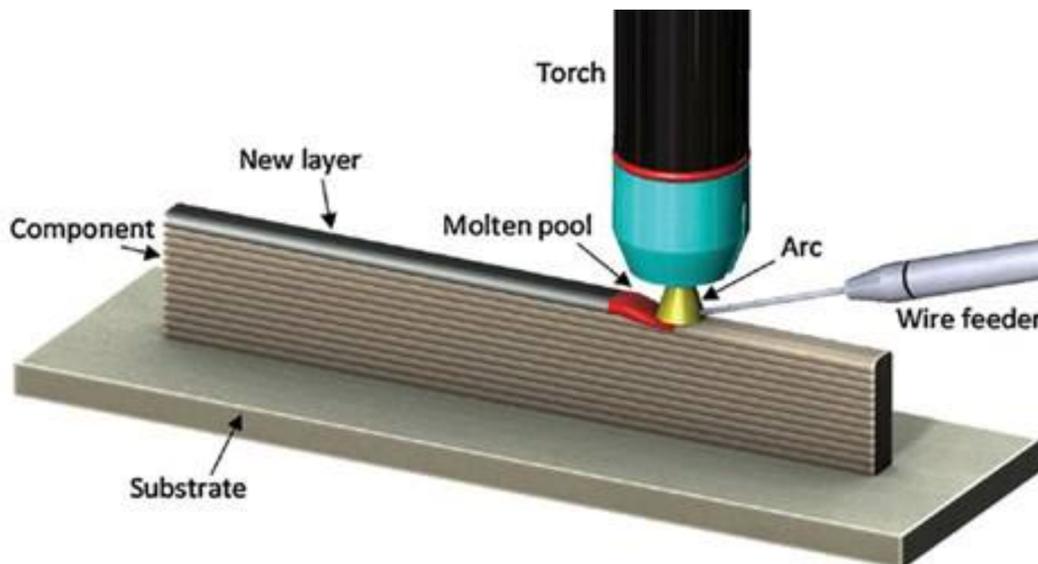
## ❖ Complément

Pour plus d'information : DED [<https://www.3dnatives.com/depot-de-matiere-sous-energie-concentree-10092019/>]

### Le WAAM (Wire Arc Additive Manufacturing)

Le WAAM est une technique d'impression 3D de métal qui appartient à la famille de la déposition direct d'énergie (DED) mais sort du lot dans la mesure où la technique s'appuie sur les concepts fondamentaux des procédés de soudage automatisés. En effet, le procédé reprend les concepts de soudage à l'arc sous atmosphère gazeuse (inerte ou active) avec fil fusible Metal Inert Gaz (MIG), le Metal Active Gaz (MAG) ou encore le Gas metal arc welding (GMAW), le soudage à l'arc au plasma (PAW) ou le soudage à l'arc au tungstène gazeux (GTAW).

La machine construit les pièces en empilant couche par couche des fils de métal fondus sortant d'une torche de soudure attachée à un bras de robot comme le ferait un procédé FMD à base de fil polymère.



Principe du WAAM

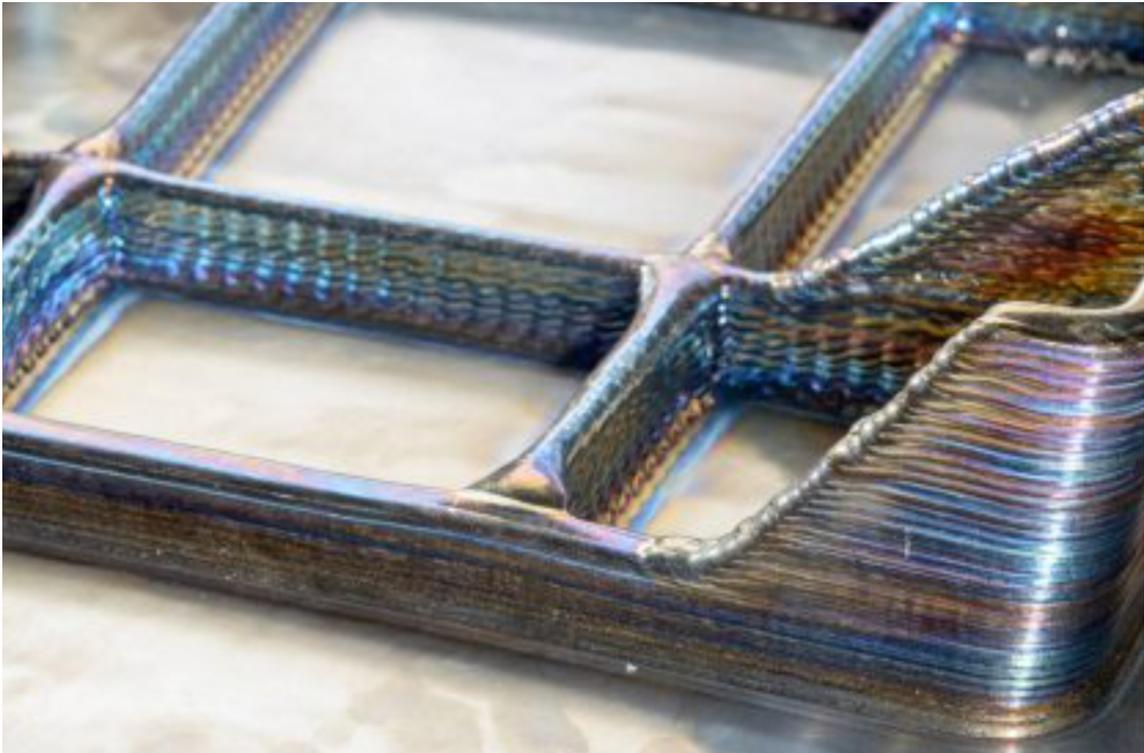
Le WAAM peut aussi être utilisé pour réparer des pièces en métal usée en edommagée comme des rails, des puces électroniques ou alors des rotors par regarnissage et réusinage.

Les avantages de ce procédé sont :

- sa capacité de réaliser des impressions 3D en très grande taille,
- un coût de production relativement bas par rapport aux autres techniques d'impression 3D métallique,
- un large choix de matériaux compatibles,
- une forte réduction des déchets car l'impression utilise la quantité de matière juste nécessaire contrairement à la fabrication soustractive.

Les inconvénients sont :

- les contraintes résiduelles qui conduisent souvent à des déformations de pièce pendant le refroidissement,
- l'état de surface obtenu est déformé par les bourrelets de matière et nécessite donc généralement une reprise en usinage.



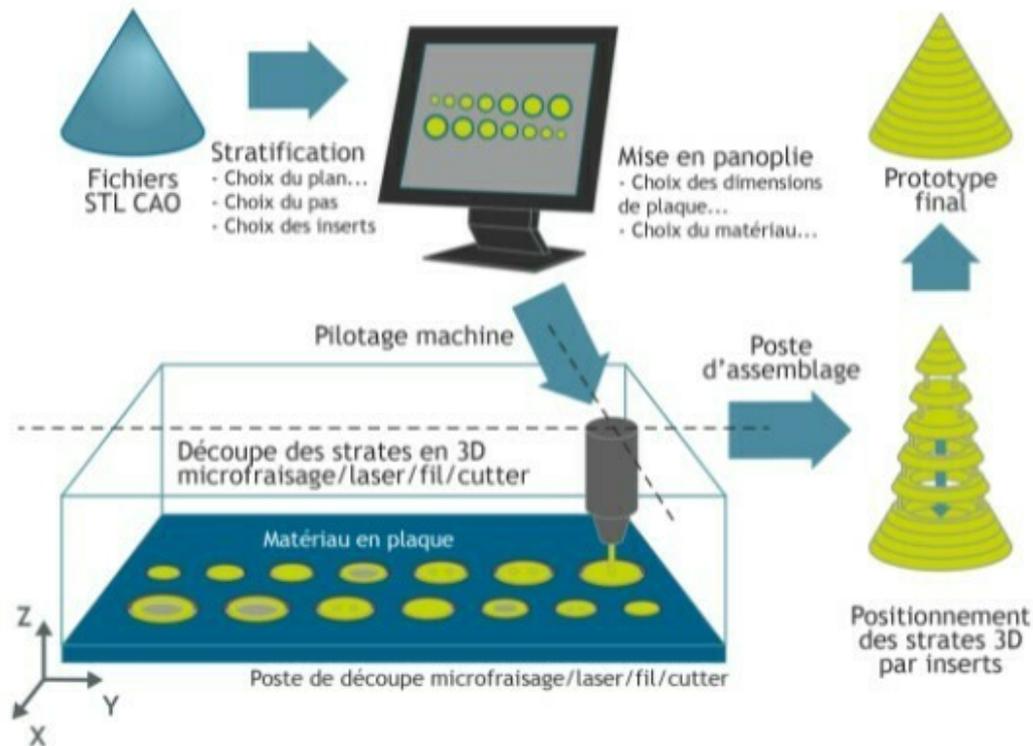
*Les défauts de surface du WAAM*

### 3.4.6. La stratoconception

La stratoconception est une méthode de fabrication additive qui consiste à produire des volumes en empilant des plaques ou des feuilles préalablement découpées.

La découpe du modèle 3D est réalisée en fonction de l'épaisseur des feuilles ou plaques découpées. La machine découpe la matière au moyen d'un laser ou d'une microfraisage ou d'un fil ou encore d'un cutter. Les couches sont empilées et liées les unes aux autres au fur et à mesure de la découpe.

Ce procédé est compatible avec un large choix de matériaux tels que les métaux, les polymères ou encore le bois.



La stratoconception

Les avantages de cette technique sont :

- la possibilité de faire des grandes pièces,
- une grande diversité de matériaux,
- une grande précision notamment pour les courbures,
- la possibilité de teinter les feuilles ou plaques pour obtenir des pièces polychromatiques.

Les inconvénients sont :

- la lourdeur de l'étape d'assemblage des sections surtout si elle n'est pas automatisée (notamment pour le collage),
- les déchets en raison des importants résidus de découpe.

### 3.4.7. L'impression 4D

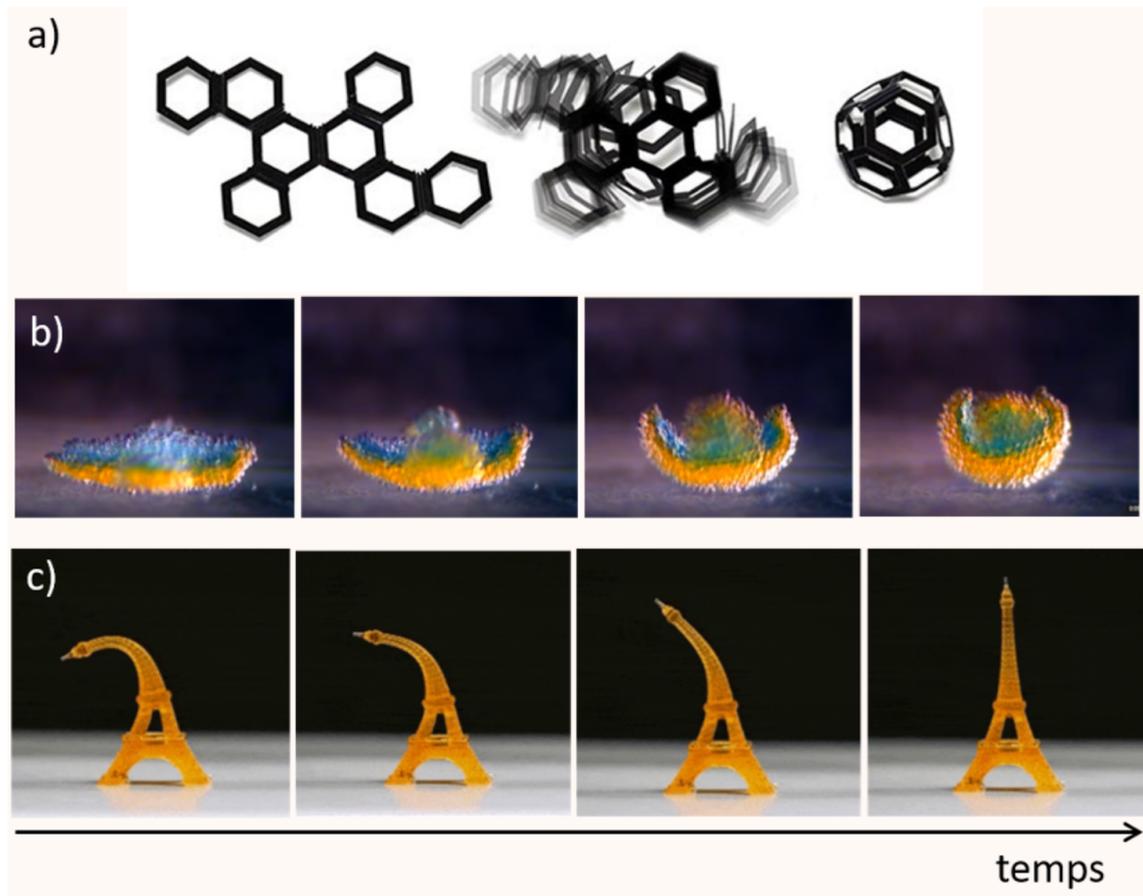
L'impression 4D, désigne la création de pièces 3D dont la forme évolue au cours du temps en fonction de stimuli externes. Les stimuli peuvent être de différente nature : physiques (champ électrique, champ magnétique, lumière, température, vibrations), chimiques (PH, photochimie) ou biologiques (glucose, enzymes, biomolécules)

Cette technologie permet de créer des objets ayant des comportements dynamiques bien qu'ils soient dépourvus d'effecteurs intégrés.

Les procédés de mise en forme restent traditionnels, la différence réside dans les matériaux utilisés :

- polymères à mémoire de forme (PMF) qui sont des matériaux capables de mémoriser une forme macroscopique, de la conserver pendant un certain temps, et de reprendre leur forme d'origine,

- élastomères à cristaux liquides (LCE en anglais) qui, comme leur nom l'indique, contiennent des cristaux liquides sensibles à la chaleur. En contrôlant leur orientation, on peut programmer la forme souhaitée,
- hydrogels. Ce sont des chaînes polymères principalement constituées d'eau, particulièrement utilisés dans les procédés de photopolymérisation,
- des multi-matériaux : ce sont principalement des composites (bois, carbone, etc.) qui sont ajoutés aux PMF ou aux hydrogels.



Exemple d'impression 4D : a) structure avec auto-assemblage, b) : tissu synthétique à comportement organique, c) : mémoire de forme piloté par température

Les applications potentielles sont nombreuses :

- composants capables de s'auto-assembler sans intervention humaine,
- composants auto-adaptables capables de combiner détection et actionnement au sein d'un même matériau,
- composants auto-réparables pour compenser des défauts (d'usure, de fabrication),
- robots miniatures souples capables de travailler dans des environnements dangereux ou de se déplacer en milieu confiné, comme dans le corps humain, pour livrer un médicament ou pour effectuer des opérations micro-invasives,
- endoprothèses, organes et tissus intelligents,
- cellules solaires auto-orientables en fonction de l'angle d'incidence la lumière,
- textiles biomimétiques auto-adaptatifs,
- etc.

## Complément

Pour plus d'information : L'impression 4D <https://www.polytechnique-insights.com/dossiers/industrie/comment-les-nouveaux-materiaux-transforment-lindustrie/impression-4d-les-materiaux-intelligents-du-futur/#:~:text=L'impression%204D%2C%20%C3%B9%20la,r%C3%A9pondent%20activement%20aux%20stimuli%20externes.>

### 3.4.8. Comparaison des procédés

Il est souvent délicat de choisir un procédé au regard des besoins, ce tableau donne quelques pistes de choix :

| Comparaison des attributs de prototypes |    |     |      |     |     |      |     |    |
|---|----|-----|------|-----|-----|------|-----|----|
| ATTRIBUT                                | SL | SLS | DMLS | FDM | 3DP | PJET | CNC | IM |
| Quantité                                | ⊖  | ⊖   | ⊖    | ⊖   | ⊖   | ⊖    | ⊕   | ⊕  |
| Complexité                              | ⊕  | ⊕   | ⊕    | ⊕   | ⊕   | ⊕    | ⊕   | ⊕  |
| Finition de surface                     | ⊕  | ⊖   | ⊖    | ⊖   | ⊖   | ⊕    | ⊕   | ⊕  |
| Choix du matériau                       | ⊕  | ⊖   | ⊕    | ⊖   | ⊖   | ⊖    | ⊕   | ⊕  |
| Stabilité du matériau                   | ⊖  | ⊕   | ⊕    | ⊕   | ⊖   | ⊖    | ⊕   | ⊕  |
| Couleur                                 | ⊕  | ⊖   | ⊖    | ⊕   | ⊕   | ⊖    | ⊖   | ⊕  |
| Tolérance                               | ⊖  | ⊕   | ⊕    | ⊖   | ⊖   | ⊖    | ⊕   | ⊕  |
| Rapidité                                | ⊕  | ⊕   | ⊕    | ⊕   | ⊕   | ⊕    | ⊕   | ⊕  |
| Prix - faible volume                    | ⊕  | ⊕   | ⊕    | ⊕   | ⊕   | ⊕    | ⊕   | ⊖  |
| Prix - volume élevé                     | ⊖  | ⊖   | ⊖    | ⊖   | ⊖   | ⊖    | ⊖   | ⊕  |

**Définitions** Les définitions varient et peuvent différer d'une entreprise à l'autre, mais on pourra utiliser les définitions ci-après comme point de départ.

**Maquette conceptuelle** - Maquette physique numériquement réalisée pour illustrer une idée. Les maquettes conceptuelles permettent aux personnes travaillant dans divers domaines, lors d'étapes de voir / concevoir / améliorer, elles stimulent la réflexion et la discussion et motivent l'acceptation ou le rejet.

**ATTRIBUTS DE PROTOTYPAGE IMPORTANTS :**

**Rapidité** - Délai nécessaire pour convertir un fichier informatique en prototype physique.

**Aspect** - tout attribut visuel - couleur, texture, dimensions, forme, etc.

**Test d'assemblage/ajustement** - Consiste à fabriquer tout ou partie des pièces d'un ensemble, les assembler et vérifier qu'elles s'assemblent bien. À un niveau rudimentaire, ce test permet d'identifier les erreurs de conception telles que deux languettes séparées par un intervalle de 2 cm alors que les fentes correspondantes sont séparées par un intervalle de 1 cm. À un niveau plus fin, il permet de contrôler les différences dimensionnelles mineures et les tolérances. Il est évident que tout test concernant des tolérances doit utiliser le processus de fabrication réel ou un processus ayant des tolérances similaires.

**Forme** - la forme de la pièce - ses caractéristiques et ses dimensions.

**Ajustement** - comment la pièce s'assemble avec d'autres pièces.

**Tests fonctionnels** - Servent à voir comment une pièce ou un ensemble fonctionneront en les soumettant à des contraintes préfigurant celles qu'ils rencontreront dans leur utilisation réelle.

**Résistance chimique** - résistance aux produits chimiques, notamment aux acides, aux bases, aux hydrocarbures, aux colorants, etc.

**Propriétés mécaniques** - robustesse de la pièce mesurée en résistance à la traction, résistance à la compression, résistance à la flexion, résistance aux chocs, résistance à la déchirure, etc.

**Propriétés électriques** - interaction entre les charges électriques et la pièce. Elles peuvent concerner la constante diélectrique, la force diélectrique, le facteur de dissipation, la résistivité superficielle et volumique, la dissipation de l'électricité statique, etc.

**Propriétés thermiques** - Modification des propriétés mécaniques se produisant sous l'effet de changements de température. Elles pourront concerner le coefficient d'expansion thermique, la température de distortion à chaud, le point de ramollissement Vicat, etc.

**Propriétés optiques** - aptitude à transmettre la lumière. Peut comprendre l'indice de réfraction, la transmission et la diffusion.

**Propriétés aux vieillissement (UV, fluage ou déformation)** - capacité à résister à une exposition aux ultraviolets avec un niveau de dégradation acceptable, capacité de la pièce à résister à des contraintes prolongées avec un niveau de déformation acceptable.

**Tests réglementaires** - Tests spécifiques réalisés par une autorité de régulation ou une organisation de normalisation afin de garantir que les pièces seront adaptées à une utilisation particulière telle qu'une utilisation médicale, un service alimentaire ou une utilisation grand public. Ces organismes sont par exemple l'ULI (Underwriters Laboratory), la CSA (Canadian Standards Association), la FDA (US Food and Drug Agency), la FCC (US Federal Communications Commission), l'ISO (International Standard Organization) et la CE (Commission européenne).

**Résistance au feu** - degré d'inflammabilité d'un matériau ou d'une pièce en présence de feu.

**Propriétés EMI/RFI (interférences électromagnétiques/radioélectriques)** - aptitude d'un matériau, d'une pièce ou d'un ensemble à filtrer ou à bloquer les interférences électromagnétiques ou radioélectriques.

**Qualité alimentaire** - agrément pour l'utilisation d'un matériau ou d'une pièce pour les applications où ils se trouveront en contact avec des aliments en cours de préparation, servis ou consommés.

**Biocompatibilité** - aptitude d'un matériau ou d'une pièce à pouvoir être en contact avec le corps humain ou animal, à l'extérieur ou à l'intérieur du corps, sans causer aucun effet nocif tox., irritations, réaction avec le sang, toxicité, etc... La biocompatibilité est importante pour les instruments chirurgicaux et pour beaucoup d'autres appareils médicaux.

Comparaison des procédés de fabrication

## 3.5. Les matériaux utilisés en impression 3D

### 3.5.1. Introduction

Les matériaux utilisés en impression 3D sont très variés et leur choix dépend à la fois du procédé et des fonctionnalités recherchées.

### 3.5.2. Les plastiques

## Le PLA (acide polylactique)

C'est une matière très utilisée lors des impressions 3D en FDM et se présente sous forme de filament. Il est fabriqué à partir d'amidon de maïs (ou de racine de manioc ou de betterave), ce qui lui permet d'être au contact d'aliment. En revanche, c'est une matière sensible à l'eau, aux UV et à la chaleur. Son utilisation répétée à l'une ou l'autre peut fortement endommager la matière.

En impression 3D, le PLA fond aux alentours de 160-220 °C et ne nécessite pas de plateau chauffant. Il peut être recouvert de peinture mais le ponçage est à éviter, les frottements favorisent la chaleur qui peut modifier la forme de l'objet. L'avantage du PLA est qu'il se présente sous une multitude de bobines variables, notamment au niveau des couleurs

## L'ABS (acrylonitrile butadiène styrène)

C'est aussi une matière très utilisée en impression 3D et fabriquée à base de pétrole.

Il est notamment utilisé pour la fabrication des pièces LEGO. Sa résistance à l'eau et à la chaleur est meilleure que celle du PLA et son rendu plus lisse. Sa température de fusion est de 200-250 °C mais il lui faut un plateau chauffant pour éviter qu'il se rétracte en refroidissant. Il se présente sous forme de bobine de filament ou de polymère liquide pour le procédé SLA. Il est soluble dans l'acétone, ce qui est pratique quand on doit souder des pièces métalliques ou pour lisser la matière par broyage, par trempage ou par mise en présence de vapeur.

## Le PETG (polytéréphtalate d'éthylène modifié au glycol)

C'est un matériau plus solide que l'ABS, très utilisé pour des pièces qui demandent de la flexibilité et robustesse. Il est adapté au contact alimentaire. Sa température de fusion est de l'ordre de 220 °C et il ne dégage aucune odeur.

## Le PC (Polycarbonate)

C'est un matériau résistant aux chocs, à la chaleur (jusqu'à 120 °C), il présente de bonnes performances mécaniques. Il est en revanche difficile à mettre en œuvre puisque sa température de fusion est de 260 °C, il requiert également de chauffer le plateau à 90 °C et une enceinte fermée est préconisée.

## Le PEEK et PEKK (polyétheréthercétone)

Ce sont des matières difficilement inflammables, résistantes aux hautes températures, aux agents chimiques et aux rayonnements à haute énergie. Disponibles en poudres pour le SLS ou en filaments pour le FDM. Ce sont des matériaux coûteux qui nécessitent des imprimantes professionnelles, puisque la température d'impression doit être comprise entre 380 et 440 °C, la température de lit supérieure à 100 °C et une enceinte chauffée est nécessaire, à une température d'environ 140 °C.

## Le PEI (polyétherimide) ou ULTEM®

Ce sont des matériaux haute performance de part leur résistance chimique, leur rapport poids/résistance mécanique et leur tolérance au feu, à la fumée et à la toxicité. Ils nécessitent des imprimantes 3D plus

industrielles puisque leur température d'extrusion est de 370-400°C et que la température du plateau d'impression doit être de 150°C et la chambre doit être idéalement à 80 °C.

## Les TPU et TPE (élastomères thermoplastiques)

Ce sont des matériaux très flexibles, possédant une forte élongation à la rupture, disponibles filament pour le FDM. Le TPE est plus tendre et moins résistant à l'abrasion que le TPU.

## Les polyamides (Nylon)

Utilisés sous forme de poudre pour le frittage laser (SLS), on compte notamment parmi les polyamides, le nylon qui est une matière élastique et résistante qui est disponible également sous forme de bobine de filament pour le FDM. Sa température de fusion est d'environ 235-260 °C et ne nécessite pas forcément de plateau chauffant. Il est aussi compatible avec les aliments.

## Les CFR (chargé fibre)

Ce sont des polymères chargés par des fibres composites courtes (Verre, Carbone, Kevlar) ou longues (Exclusivité Markforged) qui ont de meilleures résistances mécaniques que leurs homologues non chargés. Disponibles sous forme de filament. Leur extrusion nécessite des buses plus résistantes à l'abrasion.

## Les résines

Très souvent utilisées pour des impressions très détaillées mais peu solides, elles sont utiles pour réaliser des formes complexes en SLA. Ce sont généralement des polymères photosensibles disponibles sous forme liquide.

## Le PVA (alcool polyvinylique)

C'est un matériau de support utilisé dans les impressions 3D utilisant le dépôt de filament fondu, notamment avec les machines à plusieurs buses d'extrusion. Il est biodégradable, et se dissout dans de l'eau mais son coût est important.

## Le HIPS (high impact polystyrène)

C'est un matériau de support utilisé de la même manière que le PVA, à la différence qu'il se dissout dans du limonène. Sa dissolution demande plus de temps que le PVA et le limonène peut être agressif pour certains polymères, il est préconisé de l'utiliser comme support de pièces en ABS qui résiste à ce solvant. Sa température de fusion est comprise entre 230 et 245 degrés. Un plateau chauffant atteignant des températures entre 90 et 110 degrés est conseillé car le matériau a tendance à se déformer. Il est également recommandé d'utiliser une imprimante 3D avec une chambre fermée.

### 3.5.3. Les métaux utilisés en SLS, EBM et DED

---

#### L'aluminium

Très souvent utilisé dans des alliages comme AISi10Mg mélangé avec le silicium et le magnésium, l'aluminium est un alliage léger et résistant à la fois. Il est très utilisé dans la fabrication de moteur, dans l'aérospatial ou pour des moules à parois fines et aux géométries complexes.

#### L'alumide

Alliage fait à partir d'un polymère et d'aluminium trouvable sous forme de poudre, il reste très léger, et a un aspect proche de celui du métal. En revanche, il n'est pas aussi solide que du véritable métal. C'est une matière non-étanche car poreuse. Il est utilisé par la technique de frittage laser avec des particules d'alumine de 60 µm.

#### L'acier

L'acier est l'un des matériaux les plus utilisés dans l'industrie. C'est le premier métal à être commercialisé pour la fabrication additive. Il possède des propriétés mécaniques de haute résistance à la corrosion.

#### Le cobalt-chrome

C'est un matériau très résistant et solide malgré l'usure. On le retrouve notamment dans les prothèses médicales et l'industrie dentaire. Il est relativement résistant à des températures de plusieurs centaines de degrés, idéales pour la fabrication de moules. Compatible avec les imprimantes EBM (technique voisine du SLS).

#### Le titane

Très utilisé dans l'industrie pour son excellent rapport poids/solidité et sa résistance à la corrosion. Mélangé avec certains métaux, il peut être biocompatible et utilisable dans la fabrication de prothèses mais aussi pour l'automobile et l'aéronautique. Les inconvénients du titane sont son prix assez élevé et le fait qu'il refroidit rapidement, ce qui nécessite d'avoir des outils spécifiques pour le maintenir plus longtemps à la chaleur.

### 3.5.4. Les céramiques

---

L'avantage des céramiques est qu'elles sont pratiques pour des réalisations complexes.

On peut l'utiliser avec les imprimantes SLS, Si on utilise de la poudre de céramique, SLA, lorsque la céramique est couplée à de la résine, BJ ou encore FDM. Il est biocompatible, pratique pour les objets en contact des aliments ou pour la médecine.

La plupart du temps les pièces réalisées en céramique nécessitent un passage en four à haute température (supérieure à 1000°C) pour obtenir un émailage séché (cela sert à durcir l'objet et à le rendre lisse et brillant). C'est durant cette phase que l'émail obtient sa couleur.

## 3.5.5. Les matériaux organiques

### Les cires

La cire est généralement imprimée au moyen de la technique de projection de résine (PJ, MJ) . La cire imprimée en 3D est utilisée pour créer des moules grâce à la technique de la fonte à la cire perdue. Le moulage en 3D à la cire perdue (ou fonte à la cire perdue ) constitue un procédé de production qui permet principalement de produire des pièces de joaillerie personnalisables de très haute qualité et cela dans différents métaux.

Le processus consiste à imprimer en 3D une pièce en cire de l'objet désiré, puis à ajouter ensuite un matériau réfractaire autour de la pièce, on obtient alors un moule. Le passage du métal fondu dans le moule fait fondre de la cire se trouvant à l'intérieur et crée alors un espace vide qui est rempli de métal. Il ne reste plus qu'à briser le moule pour obtenir l'objet final. De nombreuses options de finition sont alors disponibles.

### Les matières alimentaires

Le procédé d'extrusion se prête bien à la production d'objets en chocolat, en sucre, ou en fromage. De nombreuses matières alimentaires peuvent être utilisées. Le résultat reste encore toutefois controversé par rapport à l'hygiène.

### Le bois

Il est possible de réaliser des pièces en bois en stratoconception, en projection de liant (BJ) sur de la poudre de bois, ou en le mélangeant sous forme de poudre ou de fibres à d'autres matériaux comme du polymère, afin qu'il puisse être manipulé sous forme de filament, la température d'extrusion est de 185-230°C dans ce cas.

### Les tissus biologiques

La bio-impression de tissus biologiques utilisent généralement les techniques d'impression spécifiques par laser, micro extrusion ou jet d'encre biologique. Leur utilisation est médicale notamment dans le cadre de greffes d'organe. On utilise ce procédé car il est plus efficace que les xéno greffes.

## 3.5.6. Les pigments

Il est possible lors des impressions 3D de mélanger des pigments liquides aux résines de photopolymérisation opaques ou transparentes, soit directement dans le bac pour obtenir une pièce de

couleur unie soit dans les gouttelettes projetées avec les procédés PJ et MJ afin d'obtenir des pièces polychromatiques.

Il est aussi envisageable d'ajouter des pigments sous forme de poudre (mica, cuivre, ...), en les mélangeant avec la résine avant de la mettre dans le bac. Cependant, la poudre peut avoir une masse volumique plus élevée que la résine, ce qui lui donne une tendance à couler dans le fond du bac. Dans ce cas, Il faut prévoir des supports adaptés lors du pré-traitement pour éviter que l'objet ne se détache lors de la fabrication.

### 3.5.7. Les matériaux hybrides

Les matériaux composites sont une alternative aux polymères bruts afin d'accroître leur fonctionnalité. Dans ce domaine, le champs des possibles est très grand et de nombreuses recherches sont menées. Pour ne citer que quelques exemples :

- des filaments à base de marc de café recyclé mélangé à du PLA pour obtenir une teinte particulière,
- du chanvre mélangé avec du PLA pour éviter les phénomènes de retassure,
- du béton fibré en utilisant 3 buses, une pour le ciment, une autre pour le granulats et la dernière pour la fibre de verre afin de pouvoir imprimer des murs,
- des filaments à base de poudre de grès pour avoir un texture proche de la roche,
- des filaments magnétiques à partir de poudre de fer, etc.

### 3.5.8. Compatibilité entre les technologies d'impression 3D et les matériaux

Tous les matériaux ne sont pas compatibles avec tous les procédés et les performances d'un même matériau peuvent différer selon le procédé de mise en oeuvre.

Ci-dessous, une comparaison entre les procédés et compatibilité des matériaux :

| PROCÉDÉ                                   | DESCRIPTION                                    | ROBUSTESSE       | FINITION                                     | EXEMPLES DE MATERIAUX                                       |
|---|--|------------------|--|---|
| Stéréolithographie - SL                   | Photopolymérisation au laser                   | 17,2-68,9 (mpa)  | Couches additives 0,051-0,152 mm en moyenne  | Photopolymères ressemblant à des thermoplastiques           |
| Frittage sélectif par laser - SLS         | Frittage laser de poudres                      | 36,5-77,9 (mpa)  | Couches additives 0,102 mm en moyenne        | Nylon, métaux   |
| Frittage direct de métal par laser - DMLS | Frittage laser de poudre de métal              | 260-1310 (mpa)   | Couches additives 0,020-0,030 mm en moyenne  | Acier inoxydable, titane, chrome, aluminiums, Inconel       |
| Dépôt de fil fondu - FDM                  | Extrusions fondues                             | 35,9-67,6 (mpa)  | Couches additives 0,127-0,330 mm en moyenne  | ABS, PC, PC/ABS, PPSU                                       |
| BJET                                      | Liant liquide imprimé par jet sur de la poudre | Faible           | Couches additives 0,089-0,203 mm en moyenne  | Poudre à base de plâtre/ liant liquide                      |
| Poly-Jet - PJET                           | UJet de résine photopolymère durcie par UV     | 49,6-60,3 (mpa)  | Couches additives 0,015-0,30 mm en moyenne   | Photopolymères à base acrylique, Photopolymères élastomères |
| Usinage CNC                               | Usinage sur machine CNC                        | 20,7-137,9 (mpa) | Procédé d'usinage soustractif (lisse)        | La plupart des thermoplastiques techniques métaux           |
| Moulage par injection - IM                | Moulage par injection - moule aluminium        | 21,4-137,9 (mpa) | Moulage lisse (ou avec texture sélectionnée) | La plupart des thermoplastiques techniques                  |

Comparaison des procédés en terme de finesse et de robustesse

Ci-dessous, une matrice qui présente les atouts des matériaux et leur compatibilité avec les machines de FDM :

| Matériau            | À imprimer dans un caisson | Boite sèche recommandée | Buse renforcée requise | Température de la buse. | Température du plateau | Imprimable sur la feuille à revêtement poudré | Imprimable sur feuille lisse PEI | Imprimable sur une feuille satinée | Soluble avec la plupart des solvants | Température de déflexion thermique | Résistance à l'impact Charpy | Résistance à la traction | Prix |
|---------------------|----------------------------|-------------------------|------------------------|-------------------------|------------------------|---|----------------------------------|------------------------------------|--------------------------------------|------------------------------------|------------------------------|--------------------------|------|
| PLA                 | Non                        | Non                     | Non                    | 210 - 215 °C            | 60 °C                  | ✓   | ✓                                | ✓                                  | ✗                                    |                                    |                              |                          |      |
| PETG                | Non                        | Non                     | Non                    | 230 - 270 °C            | 90 °C                  | ✓   | avec de la colle en bâtonnet     | ✓                                  | ✗                                    |                                    |                              |                          |      |
| PETG HT             | Non                        | Non                     | Non                    | 270 °C                  | 110 °C                 | ✓   | avec de la colle en bâtonnet     | ✓                                  | ✗                                    |                                    |                              |                          |      |
| ASA                 | Oui recommandé             | Non                     | Non                    | 260 - 265 °C            | 95 - 110 °C            | avec de la colle en bâtonnet                  | avec de la colle en bâtonnet     | ✓                                  | ✓                                    |                                    |                              |                          |      |
| ABS                 | Oui recommandé             | Non                     | Non                    | 240 - 255 °C            | 110 °C                 | avec de la colle en bâtonnet                  | avec de la colle en bâtonnet     | ✓                                  | ✓                                    |                                    |                              |                          |      |
| PC (Polycarbonate)  | Oui recommandé             | Non                     | Non                    | 270 - 275 °C            | 115 °C                 | avec de la colle en bâtonnet                  | avec de la colle en bâtonnet     | ✓                                  | ✗                                    |                                    |                              |                          |      |
| CPE                 | Non                        | Oui                     | Non                    | 275 °C                  | 90 °C                  | ✓   | avec de la colle en bâtonnet     | ✓                                  | ✗                                    |                                    |                              |                          |      |
| PVA / BVOH          | Non                        | Oui                     | Non                    | 195 - 215 °C            | 60 °C                  | ✓   | ✓                                | ✓                                  | ✓                                    |                                    |                              |                          |      |
| HIPS                | Non                        | Non                     | Non                    | 220 °C                  | 110 °C                 | ✓   | ✓                                | ✓                                  | ✓                                    |                                    |                              |                          |      |
| PP (Polypropylene)  | Oui                        | Non                     | Non                    | 220 °C                  | 100 °C                 | ✗ non recommandé                              | avec du bâtonnet polypropylé     | ✓                                  | ✗                                    |                                    |                              |                          |      |
| Flex                | Non                        | Oui                     | Non                    | 230 - 260 °C            | 50 - 85 °C             | ✓   | avec de la colle en bâtonnet     | avec de la colle en bâtonnet       | ✗                                    |                                    |                              |                          |      |
| nGen                | Non                        | Non                     | Non                    | 240 °C                  | 90 °C                  | ✓   | avec de la colle en bâtonnet     | ✓                                  | ✗                                    |                                    |                              |                          |      |
| Nylon               | Oui recommandé             | Oui                     | Non                    | 250 - 270 °C            | 75 - 90 °C             | avec de la colle en bâtonnet                  | non recommandé                   | avec de la colle en bâtonnet       | ✗                                    |                                    |                              |                          |      |
| Composite materials | -                          | -                       | Oui                    | 240 - 285 °C            | 75 - 110 °C            | -   | -                                | -                                  | ✗                                    |                                    |                              |                          |      |
| Wood / metal filled | Non                        | Non                     | -                      | 190 - 220 °C            | 60 °C                  | ✓   | ✓                                | ✓                                  | ✗                                    |                                    |                              |                          |      |
| PVB                 | Non                        | Non                     | Non                    | 215 °C                  | 75 °C                  | ✗ non recommandé                              | ✓                                | ✓                                  | ✓                                    |                                    |                              |                          |      |

Comparaison des matériaux

### 3.5.9. Comment choisir un matériau ?

Pour choisir un matériau, en dehors des critères esthétiques, il y a plusieurs caractéristiques à prendre en compte, les plus importantes étant souvent celles-ci :

- ses propriétés mécaniques, notamment sa limite élastique (limite à partir de laquelle le matériau subira une déformation irréversible) et sa limite à la rupture (limite à partir de laquelle la matériau se casse),
- sa résistance thermique,
- la résistance aux produits chimiques,
- la résistance aux conditions extérieures (UV, hygrométrie...),
- sa compatibilité alimentaire.

Ce tableau recense les principaux critères de choix d'un matériau :

### Comment choisir un procédé

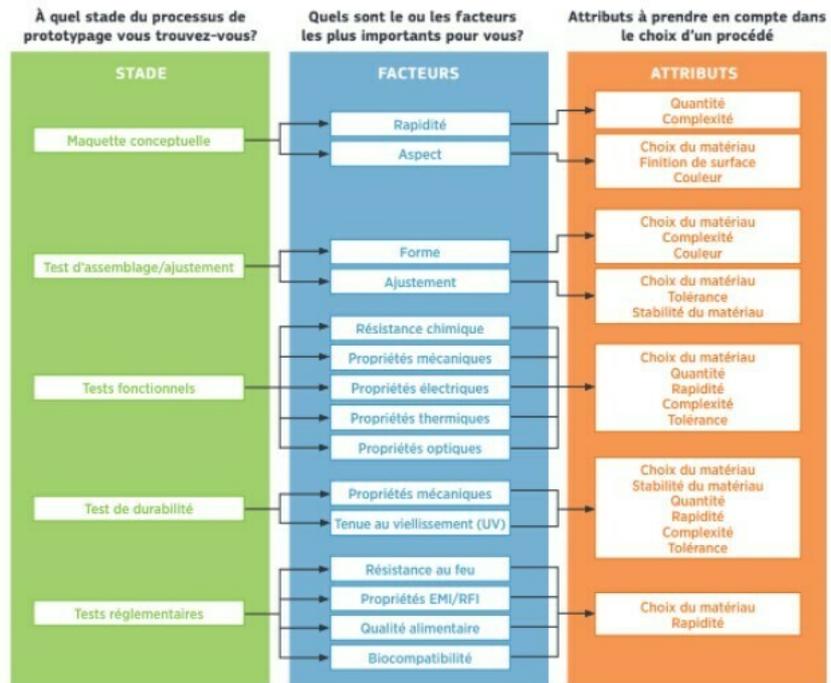
Afin de déterminer le procédé qui convient le mieux à votre projet, utilisez les outils ci-après.

#### Étape 1:

Utilisez d'abord l'arbre de décision à droite pour déterminer les facteurs présentant le plus d'importance pour vous en fonction du stade du processus de prototypage auquel vous vous trouvez, et en vous référant aux définitions de la page 8 le cas échéant.

#### Étape 2:

Sur la base des attributs recommandés pour votre ou vos facteur(s) les plus importants identifiés à l'étape 1, comparez les procédés à l'aide de la matrice de la page 8 afin de déterminer le processus le mieux adapté à votre projet.



Critères de choix des matériaux

## 3.6. Bonnes pratiques de l'impression 3D

### Procédure : Réaliser une impression 3D

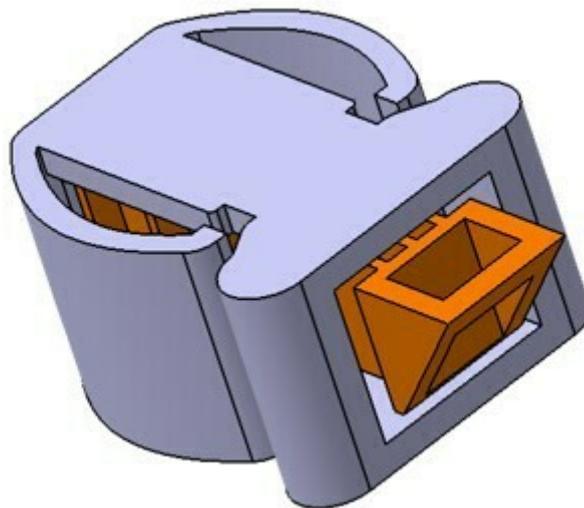
- 1 Il faut dans un premier temps posséder un modèle 3D des pièces à fabriquer qui peut être obtenu ainsi :
  - Modéliser la pièce dans un logiciel de CAO (Conception Assistée par l'Ordinateur),
  - Numériser une pièce existante avec un scanner 3D et reconstruire la pièce à l'aide d'un logiciel spécialisé,
  - Récupérer la pièce sur un site de partage de modèles 3D tel que Thingiverse, YouMagine, Shapeways ou Sculpteo.
- 2 Transformer la pièce en format STL (format inventé pour la STereo-Lithography qui historiquement était le premier procédé d'impression 3D industriel).
- 3 L'ouvrir dans un logiciel de pré-traitement appelé slicer qui va découper le modèle en couche et construire les supports nécessaires à son impression dans de bonnes conditions.
- 4 Régler et optimiser les paramètres d'impression.
- 5 Générer le fichier G-code qui sera lu par la machine.

- 6** Lancer l'impression après avoir éventuellement réglé la machine.

## Bonnes pratiques de conception 3D

Pour obtenir de bons résultats d'impression 3D, il faut respecter quelques règles de bonnes pratiques :

- Evitez d'avoir recours à des supports (pas d'angle à moins de  $40^\circ$  ou de surface en porte à faux), construisez vous-même vos supports si nécessaire pour gagner du temps en post production (ébavurage/ponçages pénibles et imparfaits, à éviter !). De manière générale, concevez vos modèles 3D pour éviter à tout prix d'avoir recours à des supports car ils altèrent la qualité des surfaces et sont parfois difficiles à retirer, une bonne pièce est une pièce utilisable directement en sortie de machine sans avoir à effectuer de retouche manuelle,
- Par exemple dans la pièce dessous, le support a été modélisé en 3D (pièce orange) pour facilement l'extraire après impression sans avoir à gratter au cutter, la pièce orange est en « lévitation » afin d'éviter qu'elle ne « colle » trop à la pièce, vous pouvez laisser trois épaisseurs de couche vide entre les surfaces, la matière s'appuiera dessus mais ne collera pas :

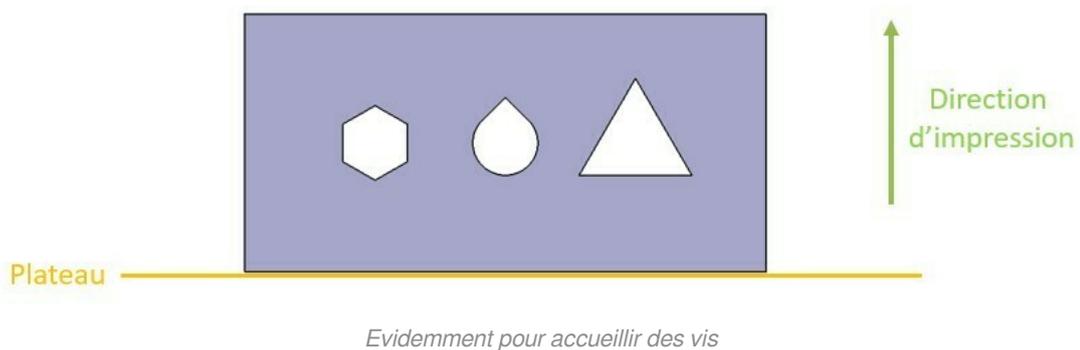


*Construisez vos propres supports d'impression extractibles*

- Imprimez des parois dont l'épaisseur est un multiple de la largeur de buse,
- Imprimez des hauteurs égales à un multiple de l'épaisseur des couches pour une bonne précision dimensionnelle et un bon état de surface,
- Utilisez une épaisseur minimale égale à 2 fois l'épaisseur des couches,
- N'exportez jamais (au format « stl » par exemple) un assemblage de pièces car l'orientation 3D de chaque pièce ne pourra pas être traitée indépendamment ce qui vous empêchera d'optimiser le positionnement et l'orientation des pièces pour éviter les supports,
- Préférer un taux de remplissage supérieur à 20% pour une meilleure qualité des couches supérieures
- Si le plastique utilisé à un fort « warping » (distorsion), évitez une base de surface faible ou utilisez du scotch au Kapton (polymère) ou déposez de la colle lavable en stick ou en bombe ou

placez vous-même dans le modèle 3D des « pastilles » de 2 couches d'épaisseur aux extrémités de vos pièces pour évitez leur décollement (les logiciels de pré-traitement peuvent rajouter de la matière sur le côté de vis pièces pour éviter ce phénomène mais en place souvent trop et il faudra ensuite les retirer au cutter, ce qui altère la qualité du résultat),

- Faites des essais en faisant varier la température de la matière que vous utilisez. Chaque fabricant et chaque matière ont des caractéristiques qui peuvent différer pour un même matériau,
- Pour mieux contrôler la qualité d'impression, placez la machine dans une salle à température et hygrométrie contrôlées. Utilisez des imprimantes avec des enceintes fermées ou fermez les vous-même, un simple carton suffit !,
- Evitez de faire passer des efforts cisailant entre deux couches : cela provoque un risque d'amorce de rupture,
- Evitez les filetages et taraudages de petites dimensions, les matériaux d'impression 3D sont fragiles et peu précis en faible dimension,
- Les perçages accueillant des vis n'ont pas besoin d'être cylindriques, vous n'utilisez pas de foret ou de mèche pour les réaliser, pensez « impression 3D » ! Les solutions ci-dessous donnent de meilleurs résultats car aucune surface ne nécessite de support (ces trois trous peuvent accueillir le même axe) :



- Pour réaliser des montages par vissage : insérer des écrous dans vos pièces ou des inserts métalliques ou utilisez des vis à bois montées dans des trous cylindriques de diamètre légèrement supérieur au diamètre au fond de filet :



Inserts métalliques

*Inserts métalliques : la solution pour visser efficacement dans des polymères*

## Les problèmes et les solutions lors des impressions 3D

Une première impression 3D se passe rarement bien, il est fréquent que la pièce ait des imperfections, ne serait-ce que mineures qui peuvent être corrigées par de bons réglages lors du pré-traitement.

Ce site, véritable mine d'or, recense la majorité des solutions aux problèmes les plus fréquents : Les solutions à vos problèmes d'impression [\[https://www.grossiste3d.com/content/27-identifier-et-corriger-les-defauts-d-impression?](https://www.grossiste3d.com/content/27-identifier-et-corriger-les-defauts-d-impression?gclid=CjwKCAiAs92MBhAXEiwAXTi257i1QkahryJsRnCYXa19VJelDd5AKEKds_Ui5ylCD18WgheqLJdmSBoCdOEQAvD_BwEJ)

[gclid=CjwKCAiAs92MBhAXEiwAXTi257i1QkahryJsRnCYXa19VJelDd5AKEKds\\_Ui5ylCD18WgheqLJdmSBoCdOEQAvD\\_BwEJ](https://www.grossiste3d.com/content/27-identifier-et-corriger-les-defauts-d-impression?gclid=CjwKCAiAs92MBhAXEiwAXTi257i1QkahryJsRnCYXa19VJelDd5AKEKds_Ui5ylCD18WgheqLJdmSBoCdOEQAvD_BwEJ)

## 3.7. La découpe laser

Le découpage laser est une technique soustractive qui consiste à graver de nombreux matériaux et de découper des plaques d'épaisseur constante grâce à un laser.

Cette technique peut découper et graver différents types de matériaux comme le bois, le plastique, le carton et graver le verre.

Le laser utilisé est précis et ne nécessite pas de source de forte puissance car la lumière est concentrée sur une toute petite surface du matériau. La focalisation du faisceau du laser, permet de chauffer la matière jusqu'à sa fusion ou sa vaporisation. La taille du rayon laser est approximativement de 0,5 mm.

Très souvent la découpe laser est guidée par ordinateur pour tracer le chemin de découpe.

## Un peu d'histoire

Le laser est l'abréviation des mots anglais Light Amplification by Stimulated Emission of Radiation. Le principe est de dévier un électron de son orbite vers une orbite plus élevée. L'électron tendant à revenir naturellement à son niveau initial, il restitue l'énergie absorbée sous forme d'un rayonnement lumineux appelé photon. Le laser utilise une propriété de l'atome qui est l'inversion de population. Cette propriété

fût découverte en 1917 par le Physicien Albert Einstein. Elle reste alors Théorique. Il faudra attendre 50 ans pour trouver une application à cette Théorie. C'est en 1960 que le physicien Théodore Maiman obtient le premier LASER avec un rubis. En 1967, Peter Holcroft découpe une plaque d'acier inoxydable de 2,5 mm d'épaisseur à une vitesse de 1 m/min, sous dioxygène avec un laser CO<sub>2</sub> de 300W et conçoit la première tête de découpe.

## Les types de découpe laser

Il existe différents types de découpe laser, en voici les principales :

### Le laser au CO<sub>2</sub>

Le laser CO<sub>2</sub> est issu d'un mélange gazeux de CO<sub>2</sub>, d'azote et d'hélium qui est ionisé par un courant électrique permettant l'excitation de celui-ci. Le CO<sub>2</sub> est le gaz qui crée le phénomène. L'azote est le gaz permettant cette réaction, l'hélium lui est utilisée pour refroidir le mélange. La puissance d'un LASER CO<sub>2</sub> est proportionnelle à la longueur d'ionisation et à la vitesse de circulation des GAZ.

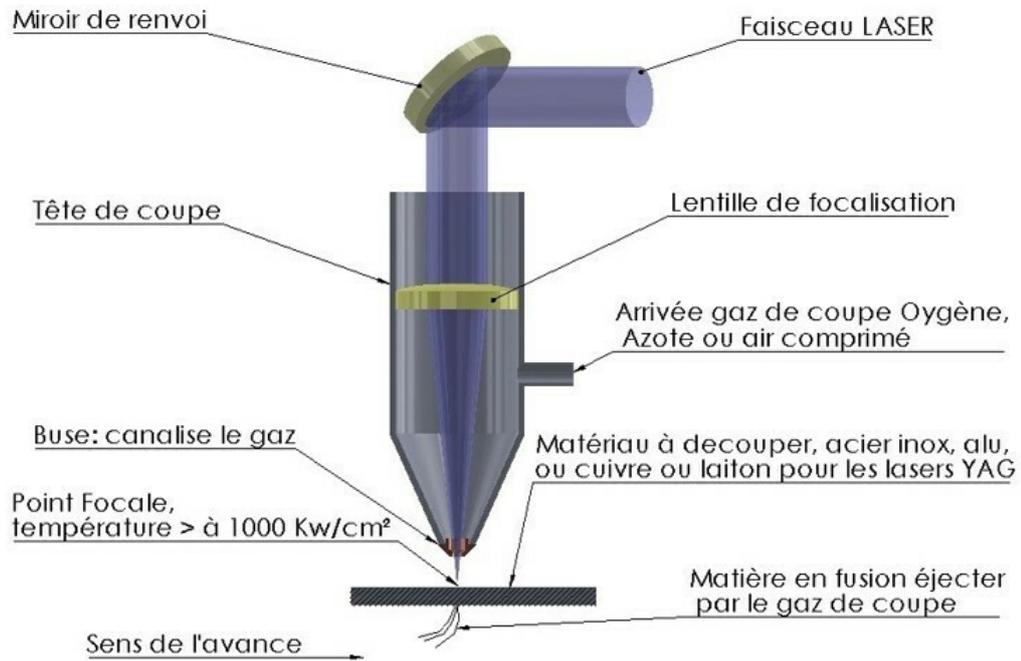
### Le laser YAG

Le laser YAG ou Grenat d'Yttrium et d'Aluminium est issu de l'excitation d'un cristal de synthèse par un courant électrique ou une source lumineuse. L'avantage du YAG est qu'il est conductible par fibre optique. Il est idéal pour les applications de soudure robotisée.

### Le laser à fibre optique

Le laser à fibre est issu de l'excitation d'une fibre de verre dopée par le biais de terres rares (l'ytterbium, l'erbium, le néodyme, le praséodyme, l'holmium, le dysprosium et le thulium). Lorsque des photons sont émis, ils sont confinés à l'intérieur de ce noyau de fibre dopée, ce qui amplifie la puissance du laser émis.

## Principe



Principe de la découpe laser

### Complément

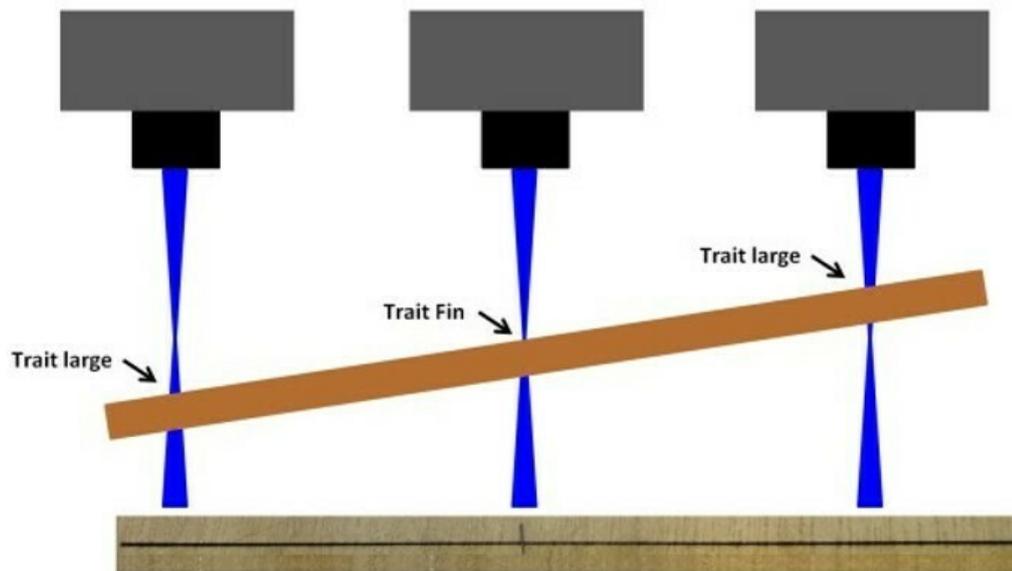
Pour plus d'informations sur ces trois techniques : La découpe laser [\[https://www.decoupe-laser-plier.com/decoupe-laser.html#:~:text=t%C3%AAt%20de%20d%C3%A9coupe-,Laser%20CO2,utilis%C3%A9%20pour%20refroidir%20le%20m%C3%A9lange.\]](https://www.decoupe-laser-plier.com/decoupe-laser.html#:~:text=t%C3%AAt%20de%20d%C3%A9coupe-,Laser%20CO2,utilis%C3%A9%20pour%20refroidir%20le%20m%C3%A9lange.)

Ces techniques ne permettent pas de découper le verre avec un laser, mais d'autres technologies plus récentes à base de faisceaux de BESSEL le peuvent : Découpe laser du verre [\[https://amplitude-laser.com/fr/news/decoupe-de-verre-structure-spatiale-et-temporelle-avancee-pour-la-decoupe-de-verre#:~:text=Les%20impulsions%20ultracourtes%20combinant%20des,toute%20la%20largeur%20du%20verre.\]](https://amplitude-laser.com/fr/news/decoupe-de-verre-structure-spatiale-et-temporelle-avancee-pour-la-decoupe-de-verre#:~:text=Les%20impulsions%20ultracourtes%20combinant%20des,toute%20la%20largeur%20du%20verre.)

### La focale de la découpe laser

La focale est la distance qui sépare la lentille convergente du foyer. Ce dernier correspond à un point unique où l'ensemble des rayons convergent. Les rayons lasers concentrés chauffent le point à tel point que la matière s'évapore. La découpe est plus performante quand ce point de foyer arrive au milieu de l'épaisseur de la plaque, c'est pourquoi il est toujours nécessaire de régler cette hauteur de focale avant de lancer la découpe. Si la focale est trop haute ou trop basse par rapport à la plaque la largeur de coupe sera d'autant plus importante que l'écart est grand par rapport à la focale idéale. De plus, les bords de coupe seront également fortement biseautés.

Principe de focale :



Réglage de la focale

## Les matières organiques compatibles

Les matériaux compatibles avec ce procédé sont les suivants :

- Les plastiques : ABS (acrylonitrile butadiène styrène), Acrylique ou PMMA (polyméthacrylate de méthyle), caoutchouc, PA (polyamide), PBT (polytéréphtalate de butylène), PC (polycarbonate), PE (polyéthylène), PET (polyéthylène téréphtalate), PI (polymide), POM (polyoxyméthylène), PP (polypropylène), PPS (sulfure de polyphénylène), PS (polystyrène), PUR (polyuréthane) et mousses sans PVC.
- les autres matériaux : bois, papier (blanc ou coloré), aliments, cuir, tissu, carton et liège.

Attention : il ne faut jamais découper au laser des matériaux à base de chlore tel que le PVC car cela peut libérer des gaz très toxiques et peut aussi endommager la machine car c'est une molécule très oxydante.

## Réaliser un projet avec une découpe-laser

- Première méthode : à partir d'un logiciel de CAO

Pour commencer, il faut réaliser un modèle 3D avec n'importe quel logiciel de modélisation 3D puis réaliser une mise en plan. Quand cela est fait, il faut enregistrer le fichier sous format DXF. Après cette première partie, il faut importer le fichier dans un logiciel de pré-traitement et définir les paramètres de coupe. Puis il faut charger le fichier dans la machine, placer la plaque, placer la tête de coupe à hauteur de focale de la surface, définir le point de départ et lancer la découpe.

- Deuxième méthode : avec un outil de dessin

Il suffit juste de dessiner directement le contour et de reprendre les étapes de la première méthode à

partir de la définition des paramètres de coupe.

A noter qu'il faut laisser le film protecteur des plaques de plastique sur les deux faces : le dessus permet d'éviter le voile blanc lié au dépôt des vapeurs de matière et le dessous permet de protéger des éventuelles flammes.

## 3.8. Les scanners 3D

---

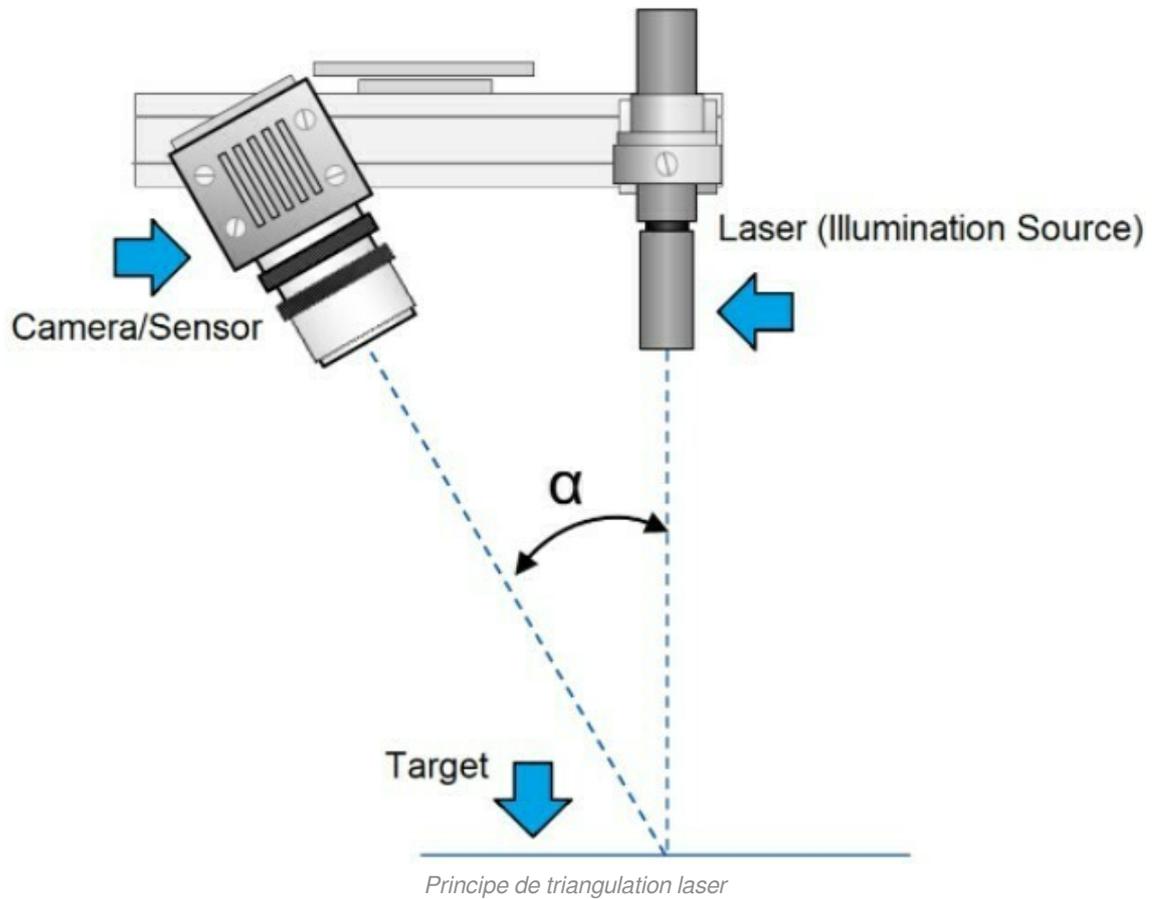
La numérisation 3D est utile pour réaliser des modèles 3D à partir de pièces de géométries complexes pouvant difficilement être relevées par des moyens de métrologie standards.

Les scanners 3D fonctionnent avec des technologies optiques combinées à des lasers pour obtenir des images 3D sous forme de nuage de points qu'il faut ensuite exploiter pour reconstruire les surfaces du modèle 3D dans des logiciels spécialisés. Ces appareils utilisent généralement une caméra pour relever les couleurs et textures de l'objet et un laser pour obtenir les coordonnées en 3D de chaque point géométrique de la pièce.

### Les principes de numérisation 3D

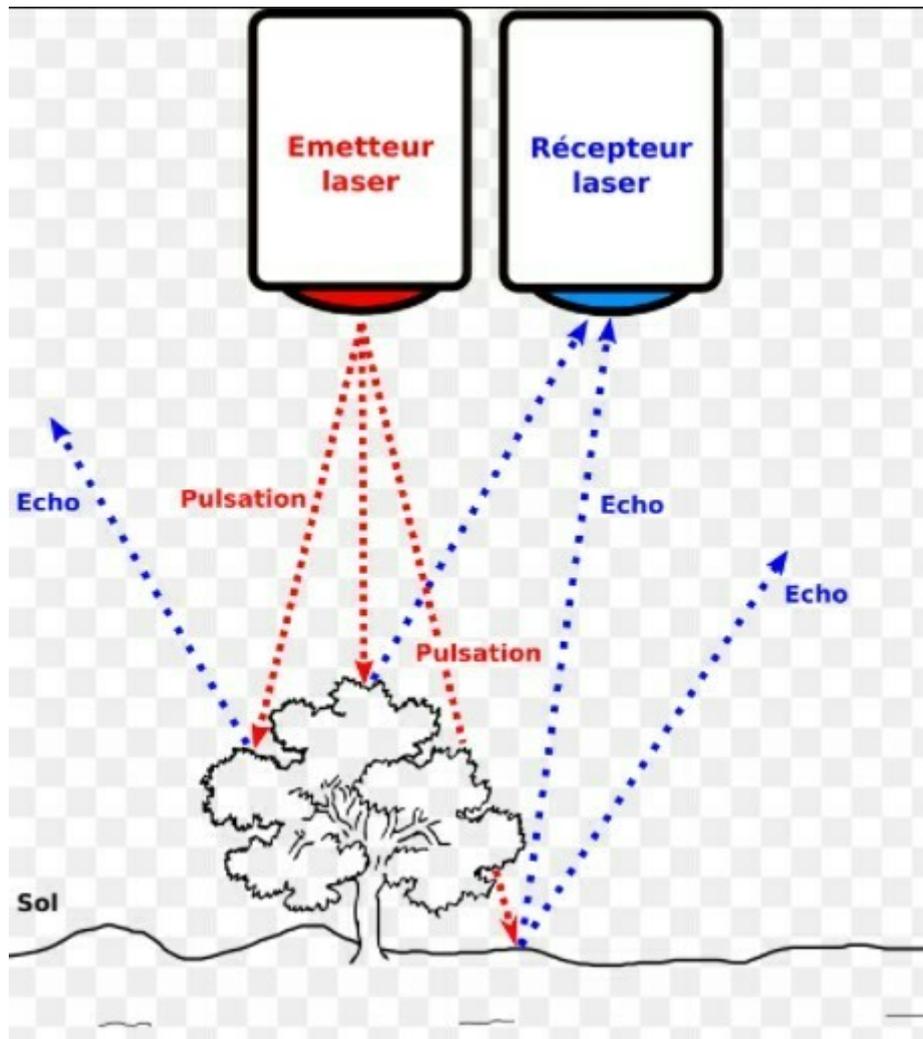
Le principe de triangulation laser :

La triangulation combine une mesure de distance et un calcul angulaire. Un émetteur projette un point laser sur un objet à mesurer. La lumière réfléchie atteint sous un certain angle un récepteur et la distance est calculée à partir du temps de vol (temps d'aller/retour) et de la vitesse de la lumière. C'est une technique très précise mais qui ne fonctionne pas pour les surfaces brillantes ou transparentes.



### L'impulsion laser ou la numérisation 3D par temps de vol

L'impulsion laser est une technique de mesure à distance fondée sur l'analyse des propriétés d'un faisceau de lumière renvoyé vers son émetteur. Cela est basé sur le calcul de la durée mise par le laser pour toucher une surface et revenir. Elle peut être aussi nommée LiDaR pour Light Detection and Ranging.



*Principe de la mesure par temps de vol*

### La photogrammétrie

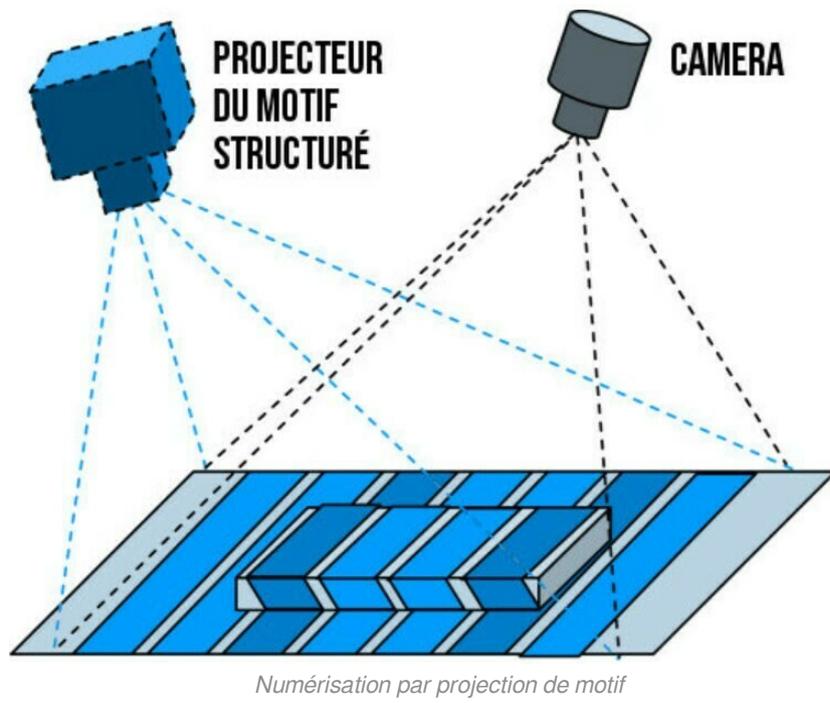
La photogrammétrie consiste à prendre plusieurs angles vue d'un objet pour le reconstruire en 3D. Les photographies peuvent prendre les proportions correctement. Elle est très utilisée pour les grands objets ou scènes. Cette technique est rapide, mais peu précise.

### Numérisation 3D par contact

Les mesures par contact se font à l'aide d'un palpeur ou d'une sonde. Ce procédé est d'une grande précision et peut relever des surfaces réfléchissantes ou transparentes. Par contre c'est un procédé assez lent et inadapté aux formes très complexes ou aux matériaux souples.

### Numérisation par lumière structurée

Cette technique projette un motif lumineux structuré en lumière infrarouge ou visible et relève sa déformation par le biais d'un capteur photo. Le calcul de la déformée permet au logiciel de reconstituer les volumes 3D. Ce procédé est rapide et plus précis que la photogrammétrie mais moins que la triangulation laser ou la mesure par contact.



## 4. Réalisation un jumeau numérique d'un système à base d'Arduino dans la 3DExperience

### 4.1. Introduction

#### Objectifs pédagogiques

Dans ce module, nous allons vous expliquer comment interconnecter un modèle virtuel d'un système mécatronique constitué d'une carte Arduino et d'un servomécanisme avec son équivalent réel dans une approche de jumeau numérique s'incrivant plus globalement dans le contexte de l'Internet des Objets.

Nous verrons comment simuler le comportement de ce système dans la 3DExperience puis comment le contrôler physiquement depuis le logiciel et enfin comment les relier intimement de sorte à ce qu'ils s'influencent mutuellement.

### 4.2. Jumeau Numérique d'un Servomécanisme piloté par Arduino et modélisé dans la 3DExperience

#### A propos de l'loT :

loT signifie en anglais Internet of Things, il est également appelé Internet of Everything ou Network of Everything. Ce concept regroupe intimement en réseau des objets réels ou virtuels qui intègrent des composants électroniques, logiciels et des éléments de connectivité qui leur permettent d'échanger des données, des mesures ou de contrôler d'autres systèmes similaires distants.

Selon l'Union Internationale des télécommunications, l'Internet des objets en français (IdO) est une « **infrastructure mondiale pour la société de l'information, qui permet de disposer de services évolués en interconnectant des objets** ([\[https://www.futura-sciences.com/sciences/definitions/physique-physique-15839/\]](https://www.futura-sciences.com/sciences/definitions/physique-physique-15839/)) **physiques ou virtuels) grâce aux technologies de l'information et de la communication interopérables existantes ou en évolution** ». En réalité, la définition de ce qu'est l'Internet des objets n'est pas figée. Elle recoupe des dimensions d'ordres conceptuel et technique.

D'un point de vue conceptuel, l'Internet des objets caractérise des objets physiques connectés ayant leur propre identité numérique et capables de communiquer les uns avec les autres. Ce réseau crée en quelque sorte une passerelle entre le monde physique et le monde virtuel.

D'un point de vue technique, l'IdO consiste en l'identification numérique directe et normalisée (adresse MAC, IP, protocoles SMTP...) d'un objet physique grâce à un système de communication sans fil qui peut être une puce RFID, Bluetooth ou WiFi ([\[https://www.futura-sciences.com/tech/definitions/internet-wi-fi-1648/\]](https://www.futura-sciences.com/tech/definitions/internet-wi-fi-1648/)).

L'Internet des Objets permet à des systèmes de percevoir des signaux ou d'être contrôlé à distance au travers d'un réseau et de créer des opportunités nouvelles d'intégration entre le monde physique réel et le monde virtuel pour améliorer leur performance et générer des plus grands bénéfices économiques.

En 2020, des experts ont estimé que la population des IoT était de 50 billions ( $10^{12}$ ) d'objets.

L'interfaçage d'une carte Arduino et de son jumeau numérique implémenté dans la 3DExperience s'inscrit dans ce concept de l'Internet des Objets.

Pour plus d'informations : IoT [\[https://www.futura-sciences.com/tech/definitions/internet-internet-objets-15158/\]](https://www.futura-sciences.com/tech/definitions/internet-internet-objets-15158/)

## A propos des Jumeaux Numériques

IBM définit le jumeau numérique comme une représentation virtuelle d'un objet ou d'un système qui couvre son cycle de vie. Il est mis à jour à partir de données en temps réel et utilise la simulation, l'apprentissage automatique et le raisonnement pour aider à la prise de décision.

Par exemple, pour compenser une faiblesse, un défaut de fabrication, une usure, un encrassement ou un vieillissement du système réel par un meilleur réglage en temps réel de ses paramètres de fonctionnement au regard d'un objectif de performance à atteindre.

Pour plus d'informations : Le Jumeau Numérique [\[https://www.ibm.com/fr-fr/topics/what-is-a-digital-twin\]](https://www.ibm.com/fr-fr/topics/what-is-a-digital-twin)

### Procédure : Réalisation du jumeau numérique

Dans ce tutoriel, nous allons réaliser le jumeau numérique d'un servomécanisme et piloter son équivalent réelle avec une carte Arduino.

Dans cette procédure nous utiliserons 4 applications de la 3D Experience :

- Part Design : pour réaliser 2 composants
- Assembly Design : pour assembler les deux composants
- Mechanical System Design : pour créer le modèle de simulation dynamique
- Functional and Logical Design : pour y adjoindre le modèle logique qui permettra son pilotage
- Behavior Modeling : pour créer le modèle physique associé et simuler son comportement réel

#### **1** Réglages préalables

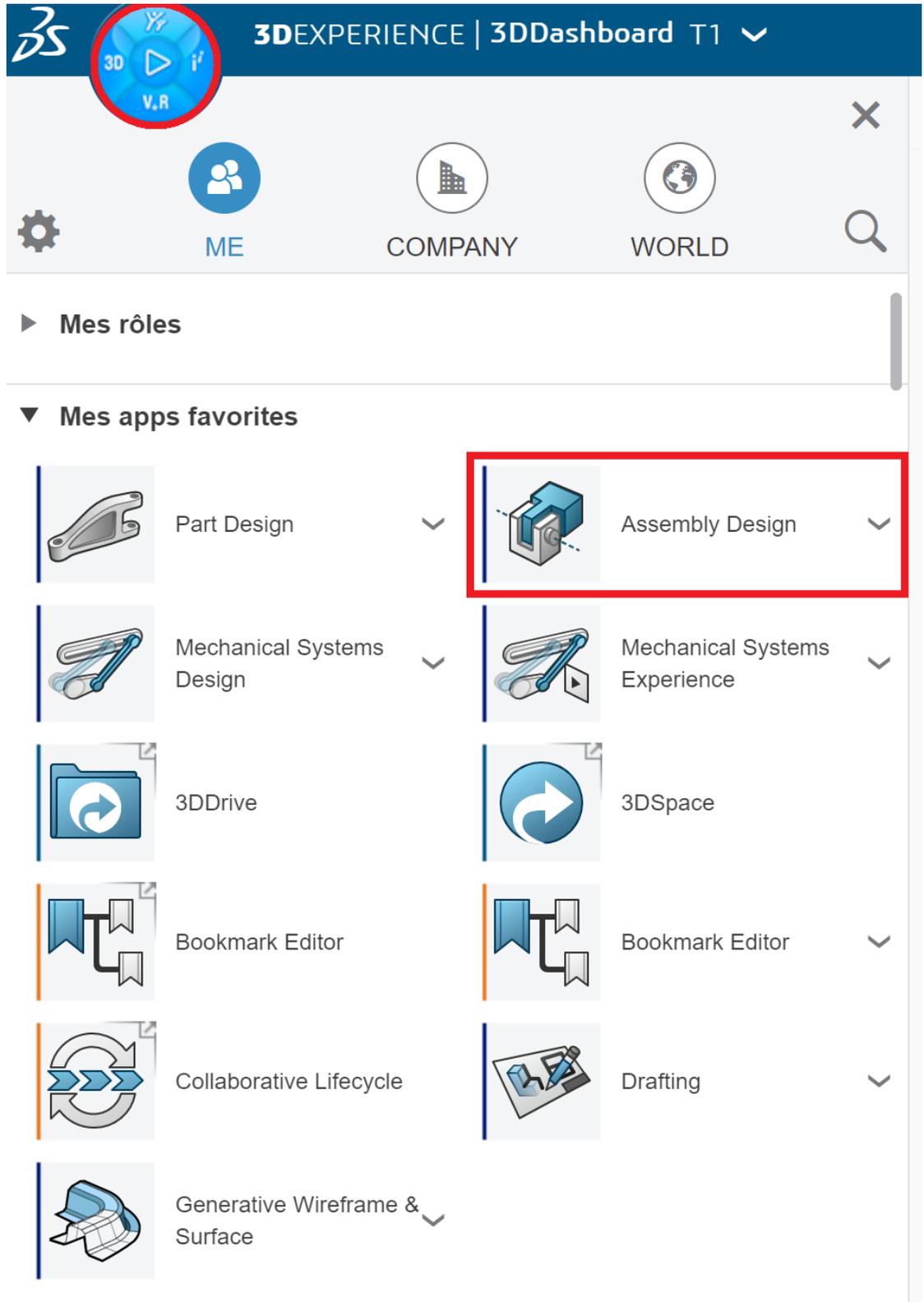
Connectez vous à la 3DExperience dans le client web, puis Veillez à sélectionner la bonne base de données :

The screenshot shows the 3DEXPERIENCE dashboard with the following components:

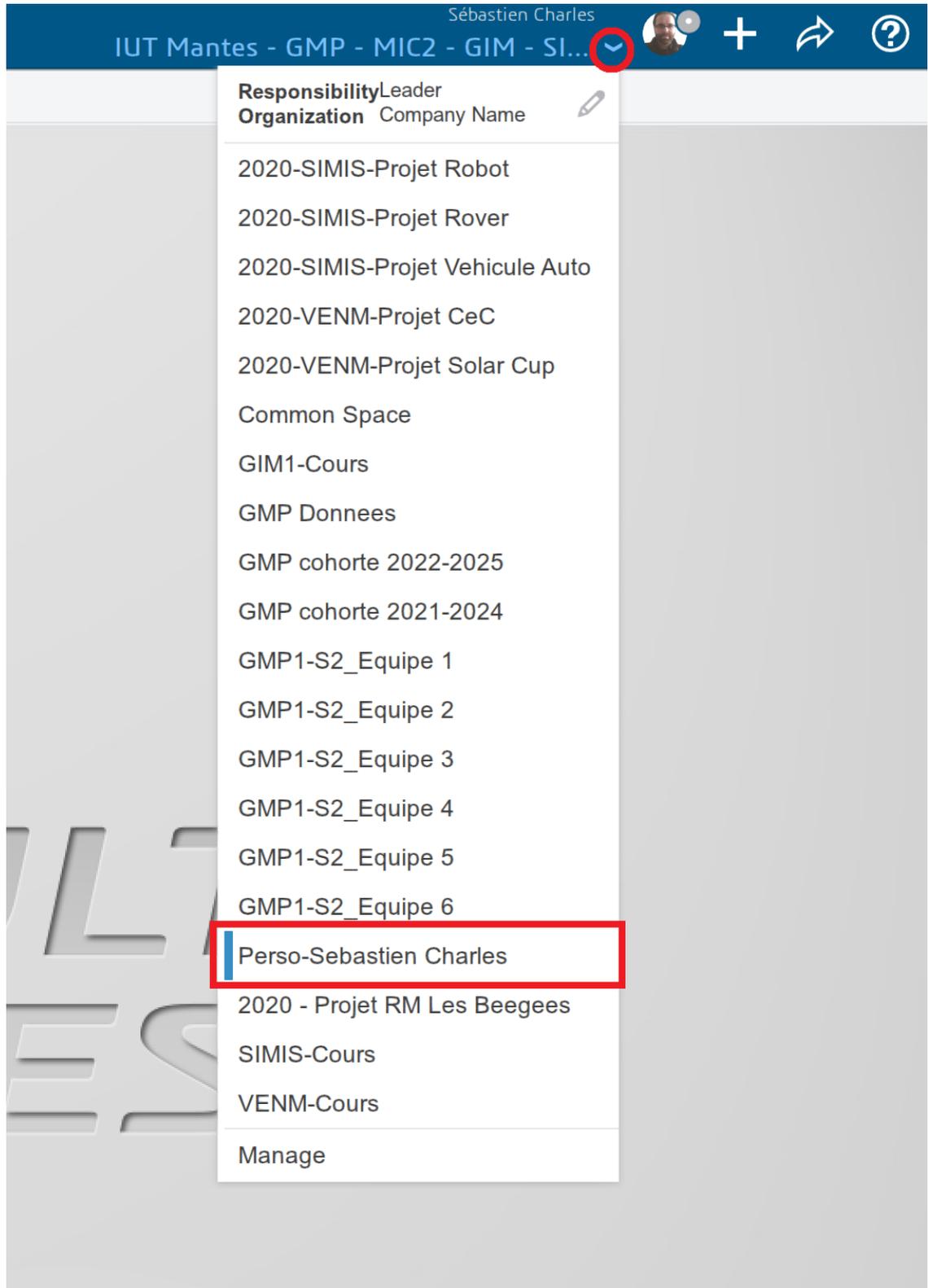
- Top Bar:** 3DEXPERIENCE | 3DDashboard T1, Recherche IUT Mantes - GMP - MIC2 - GIM, and user profile 'Sebastien Charles'.
- Left Panel:** 'Mes rôles' and 'Mes apps favorites' (Part Design, Assembly Design, Mechanical Systems Design, 3D Drive, 3D Space, Bookmark Editor, Collaborative Lifecycle, Generative Wireframe & Surface).
- Center Panel:** 'Mes espaces de collaboration' listing various workspaces like '2020 - Projet RM Les Beeeegs', '2020-SIMIS-Projet Robot', etc.
- Right Panel:** A table of collaboration spaces with the following data:
 

| Titre                           | Visibilité    | Famille |
|---------------------------------|---------------|---------|
| Common Space                    | 18 mai 2019   | Privée  |
| SIMIS-Cours                     | 03 nov. 2020  | Privée  |
| 2020-SIMIS-Projet Vehicule Auto | 03 nov. 2020  | Privée  |
| 2020-SIMIS-Projet Robot         | 03 nov. 2020  | Privée  |
| 2020-SIMIS-Projet Rover         | 03 nov. 2020  | Privée  |
| VENM-Cours                      | 05 nov. 2020  | Privée  |
| 2020-VENM-Projet CeC            | 05 nov. 2020  | Privée  |
| 2020-VENM-Projet Solar Cup      | 05 nov. 2020  | Privée  |
| GIM GIM1-Cours                  | 06 nov. 2020  | Privée  |
| 2020 - Projet RM Les Beeeegs    | 30 nov. 2020  | Privée  |
| Perso-Sebastien Charles         | 05 juil. 2021 | Privée  |
| GMP1-S2_Equipe 1                | 07 Mvr. 2022  | Privée  |
| GMP1-S2_Equipe 2                | 07 Mvr. 2022  | Privée  |

Lancez ensuite l'application Assembly Design :

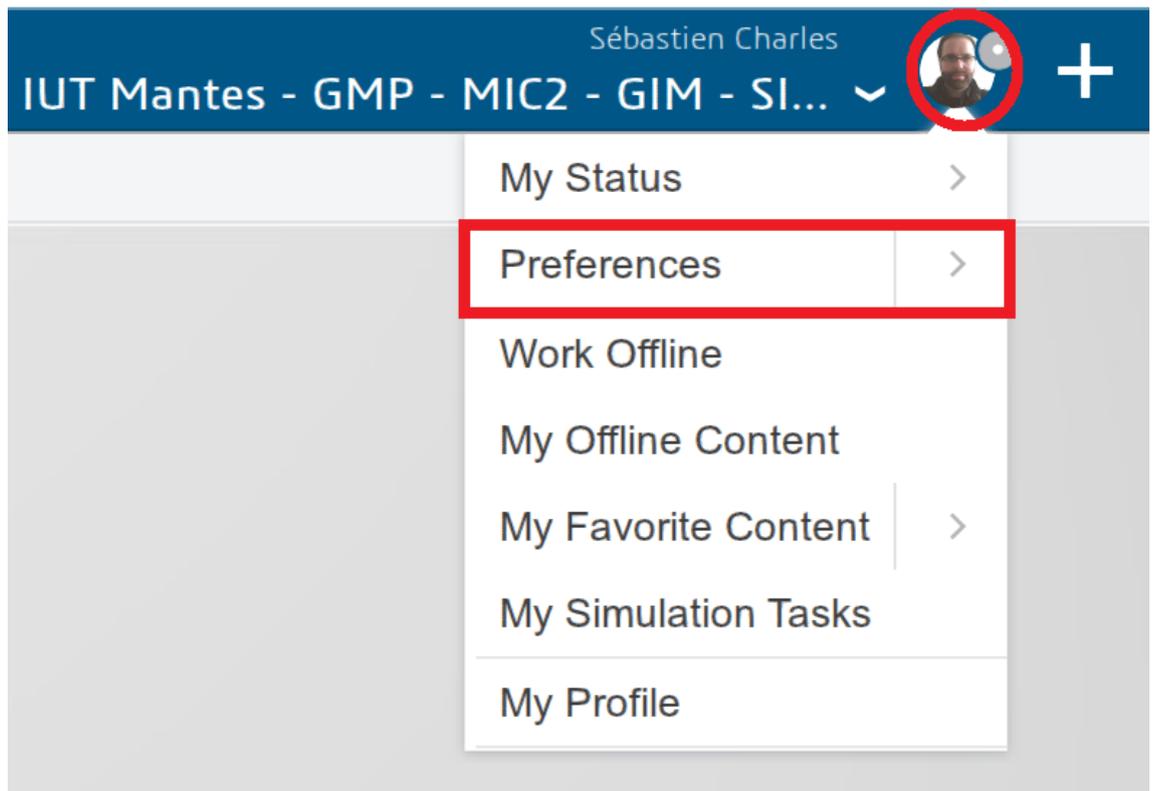


Une fois, le client lourd ouvert, veillez à sélectionner le bon collaborative space :

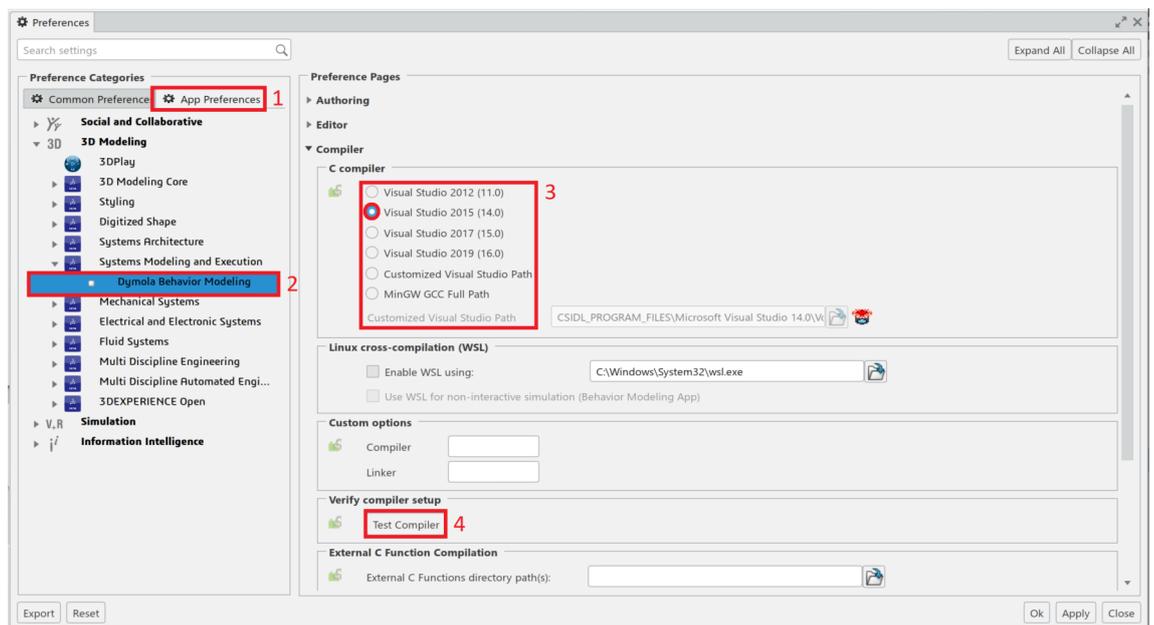


Si vous devez changer de collaborative space, il faut fermer tous les documents ouverts non enregistrés et recommencer.

Allez dans les préférences d'environnement :

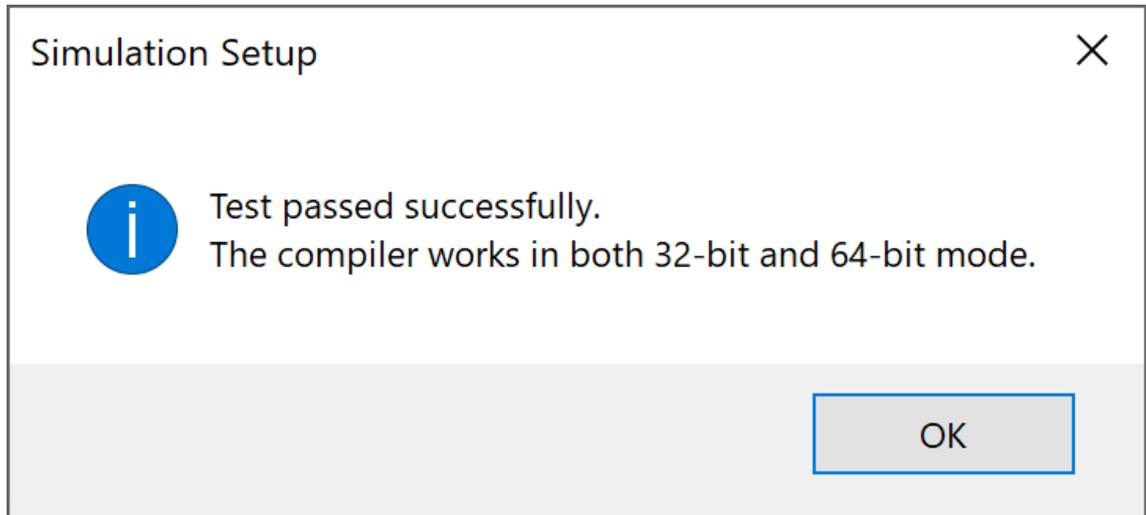


Vérifiez qu'un des compilateurs requis est bien installé, bien sélectionné dans la liste :

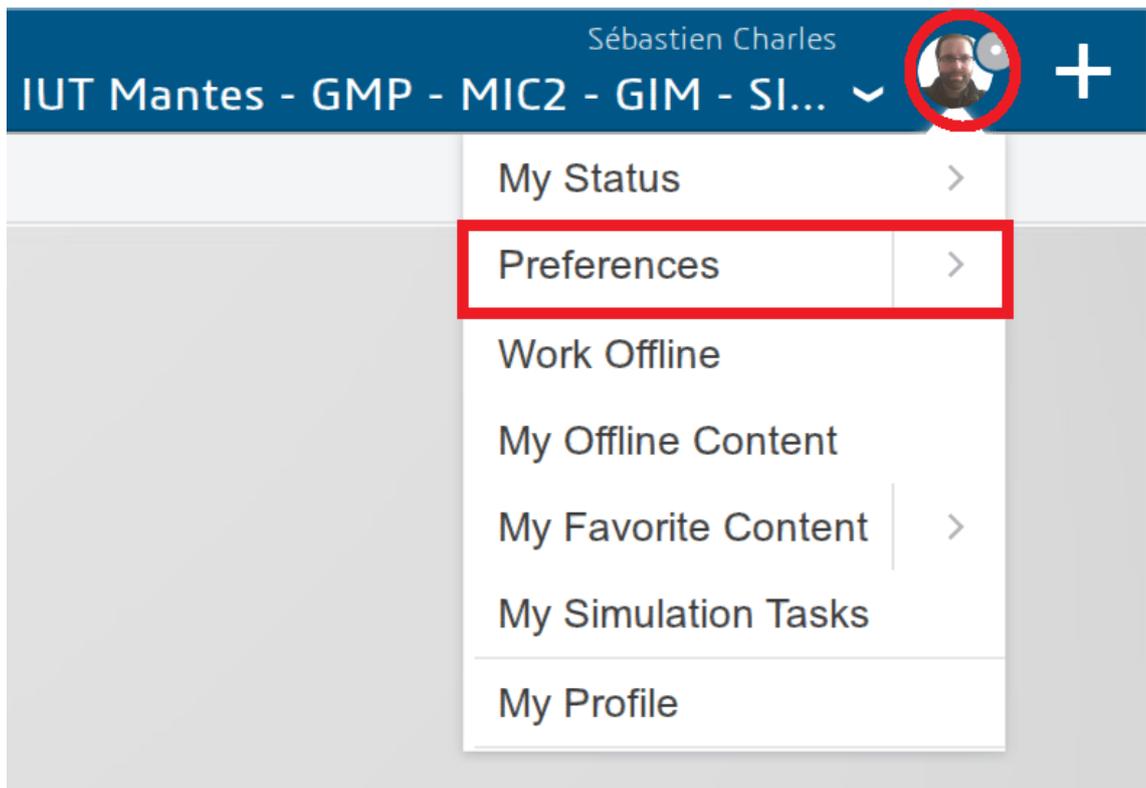


Si aucun compilateur n'est présent, installez le, relancez le client lourd et sélectionnez le dans la liste.

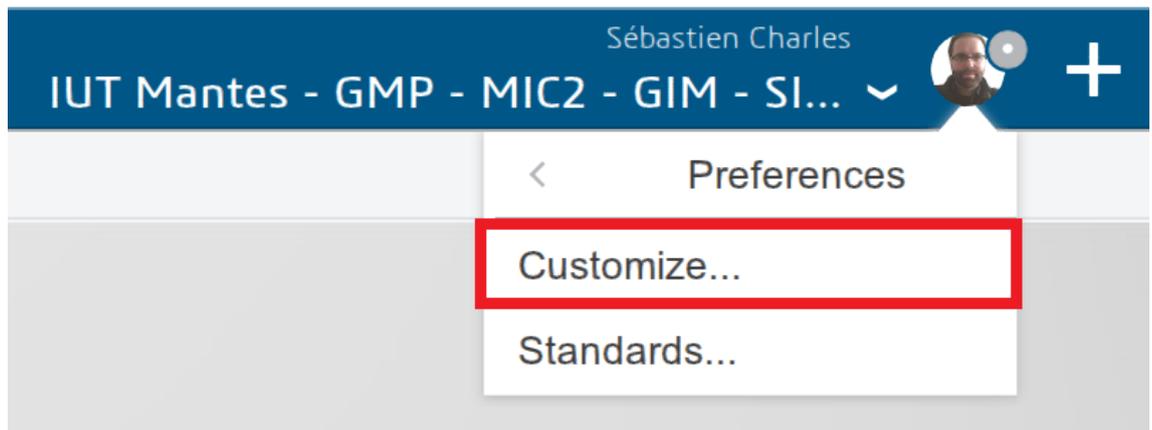
Testez le compilateur. Le message suivant doit apparaître :



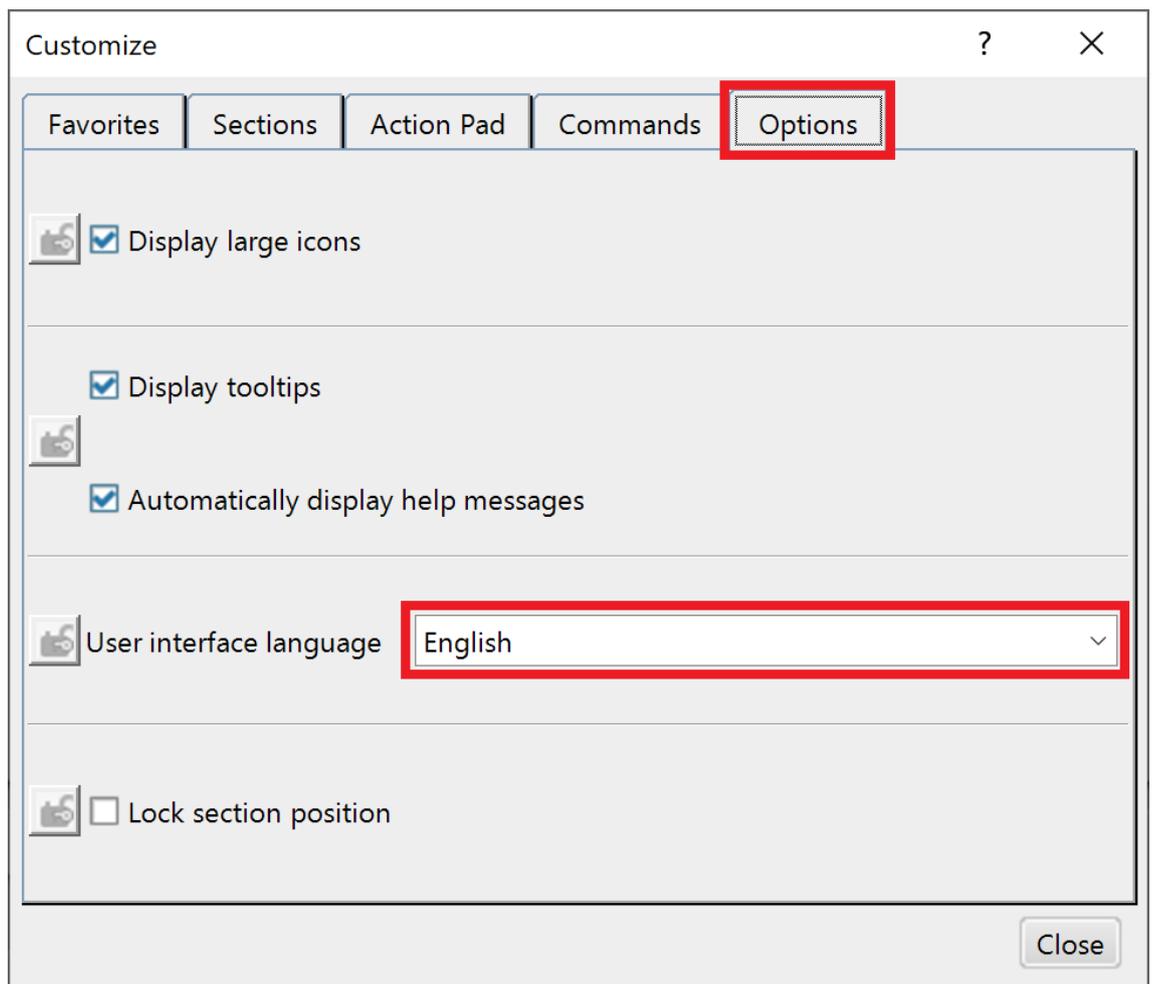
Ensuite, cliquez sur la flèche située à droite de Preferences :



Puis sur Customize :



Puis passez la langue de l'environnement en anglais :

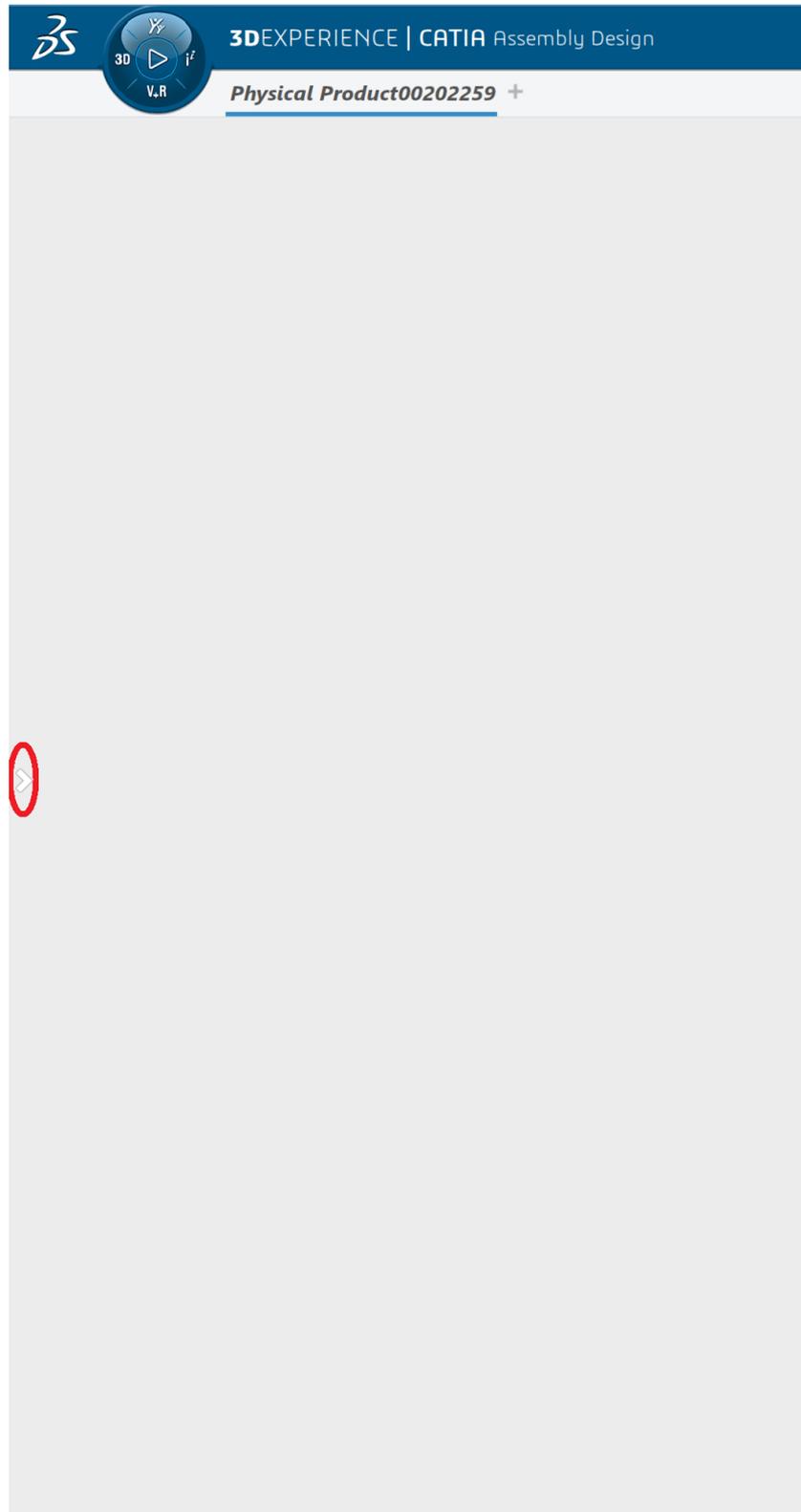


Cette modification est nécessaire en raison de l'utilisation de modèles de simulation au format Modelica et du standard d'échange FMI (tous deux développés en anglais) pour dialoguer avec Arduino.

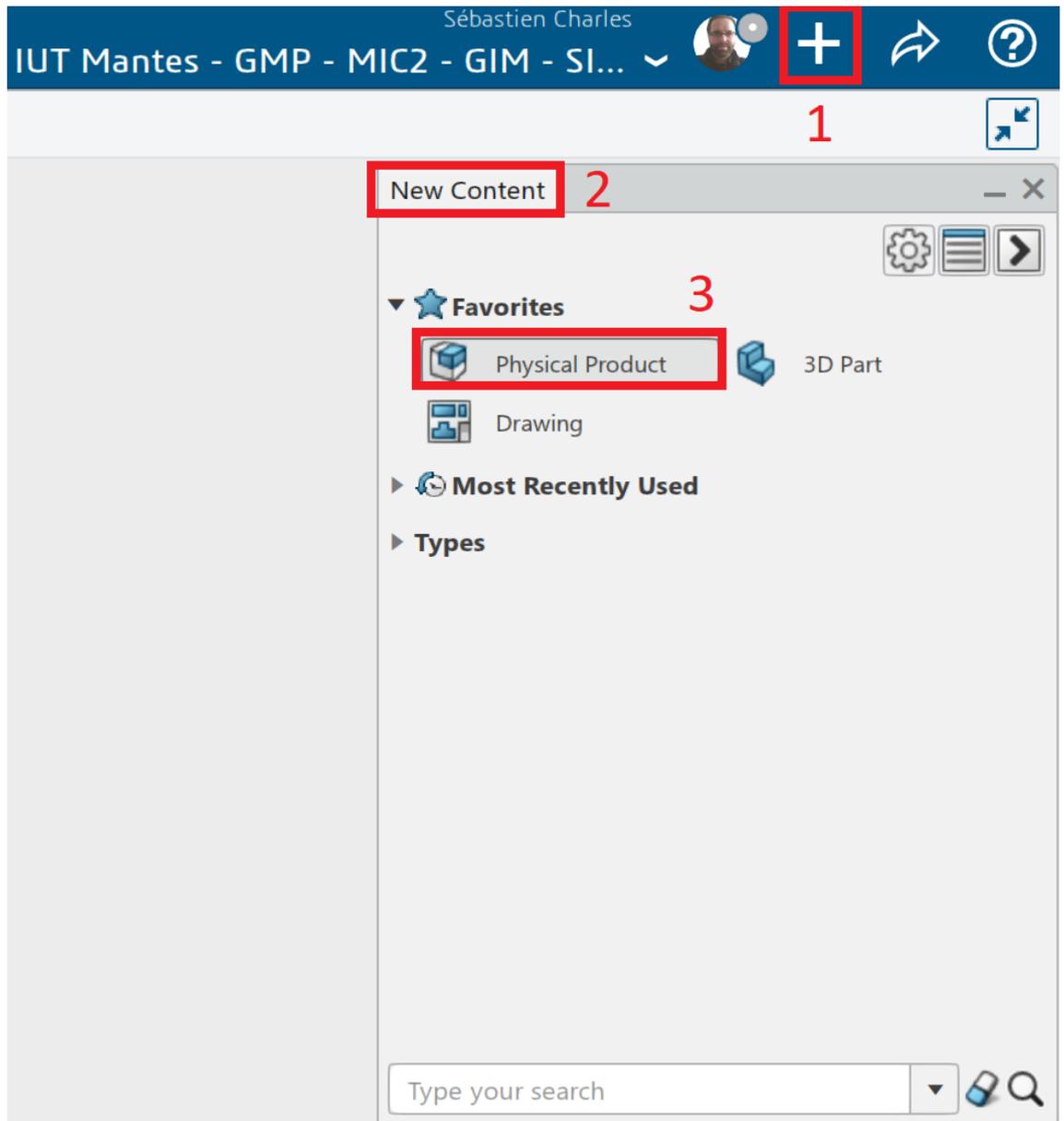
## 2 Création du modèle 3D du servomécanisme

Structuration de l'arbre du modèle 3D de l'assemblage

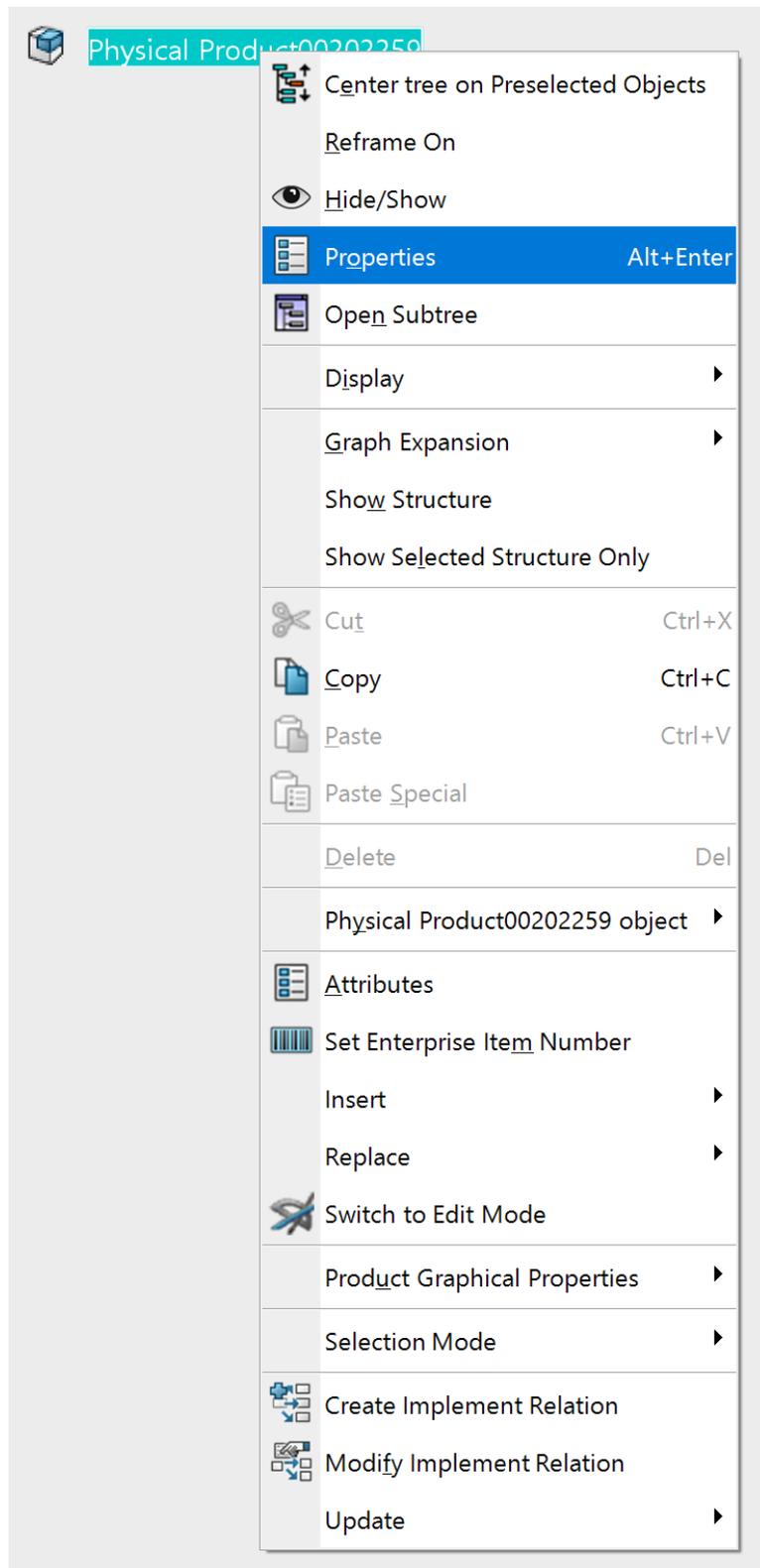
Si un produit est bien créé, cliquez sur la flèche > située à gauche de l'écran pour visualiser l'arbre du modèle :



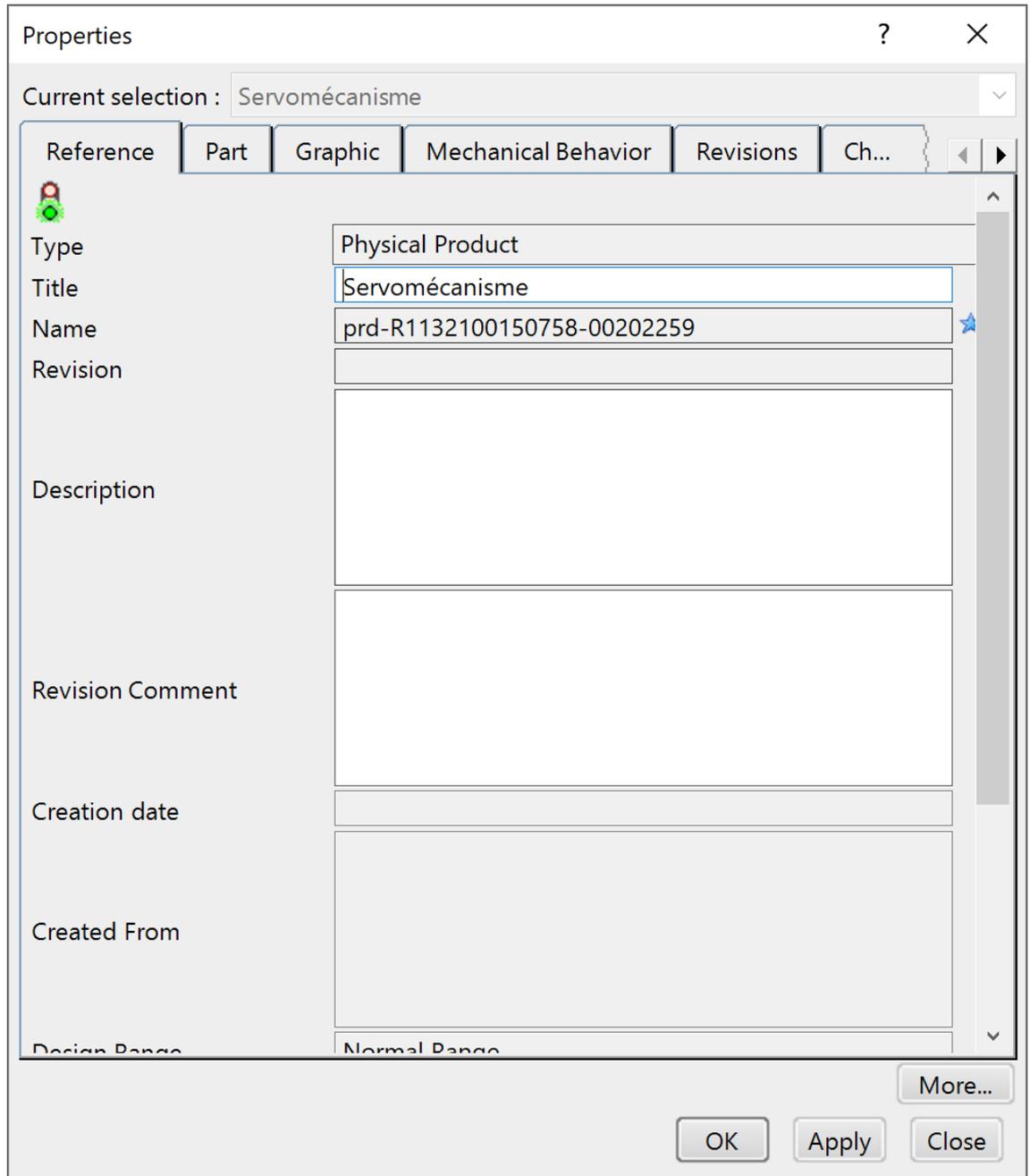
Si aucun produit n'apparaissait dans la fenêtre, créez en un préalablement en cliquant sur + en haut à droite puis New Content puis Physical Product :



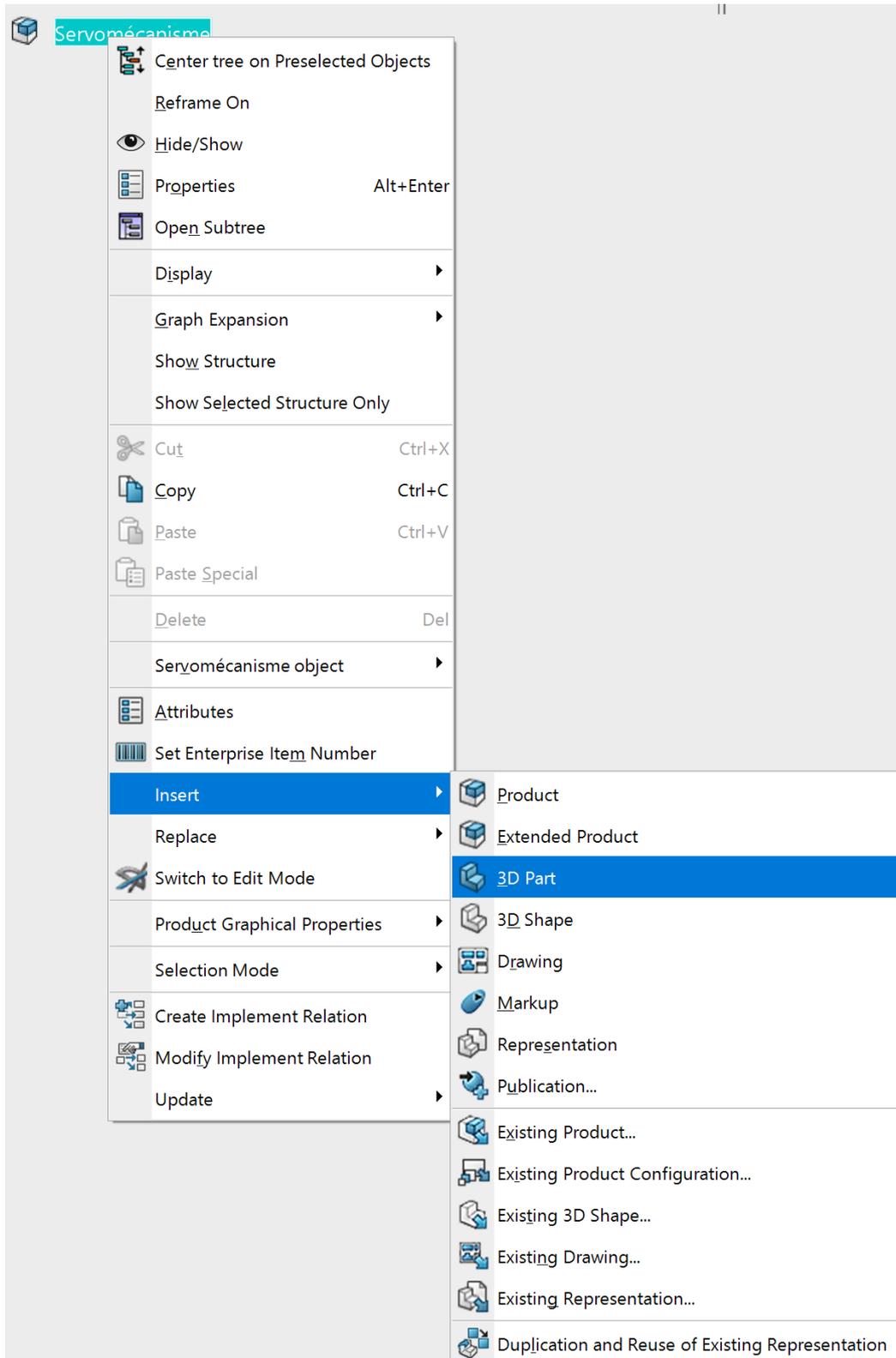
Renommez votre assemblage en faisant un clic droit sur le plus haut niveau de l'arbre du modèle puis Properties



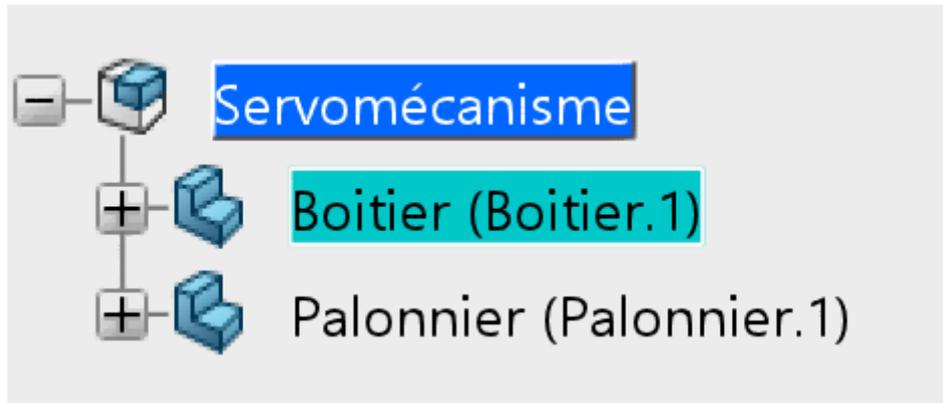
Renommez en « Servomécanisme » dans la case Title :



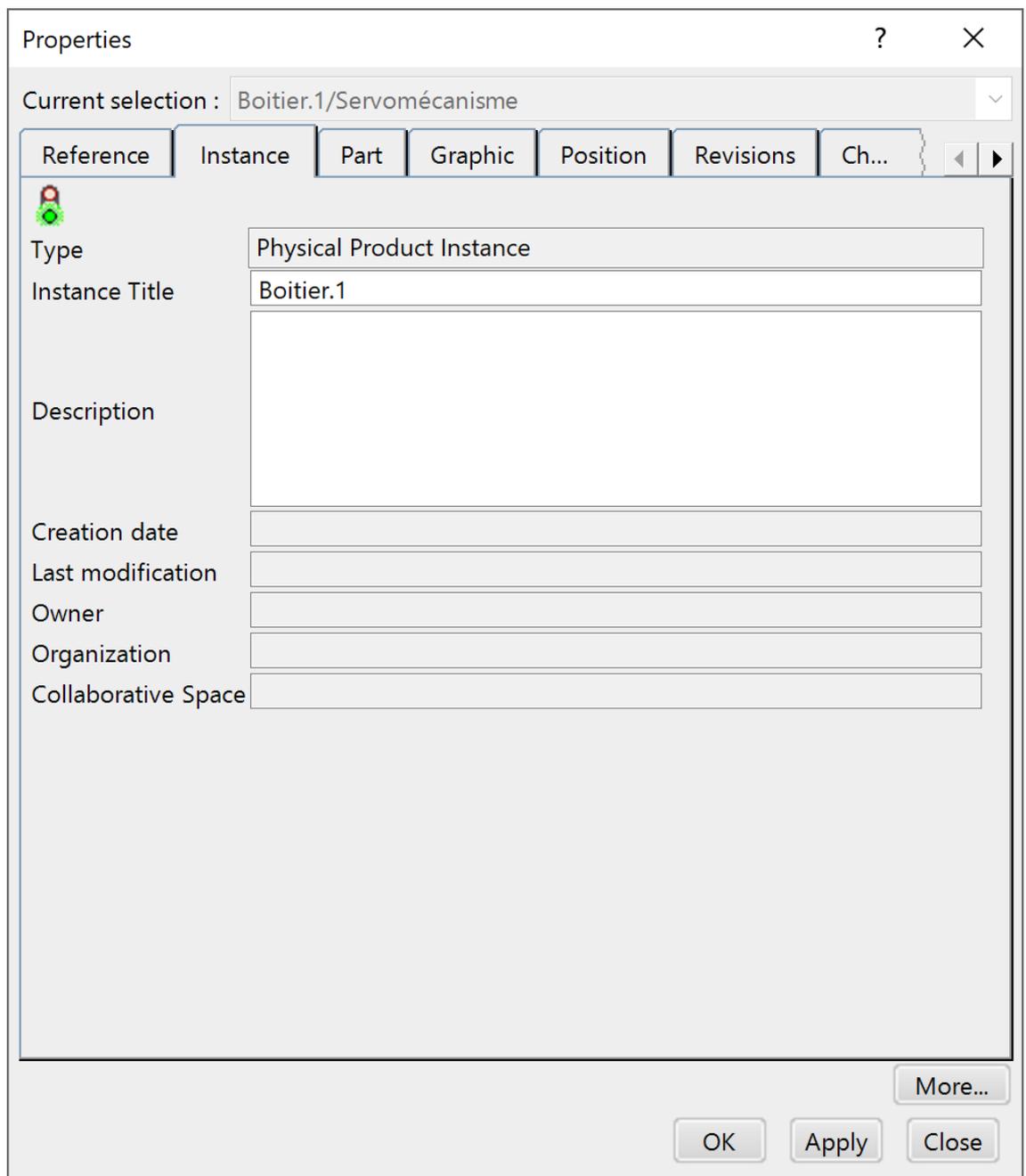
Faites un clic droit sur le plus haut niveau de l'arbre du modèle puis Insert D Part :



Recommencez l'opération une deuxième fois puis renommez les deux pièces en Boitier et Palonnier :



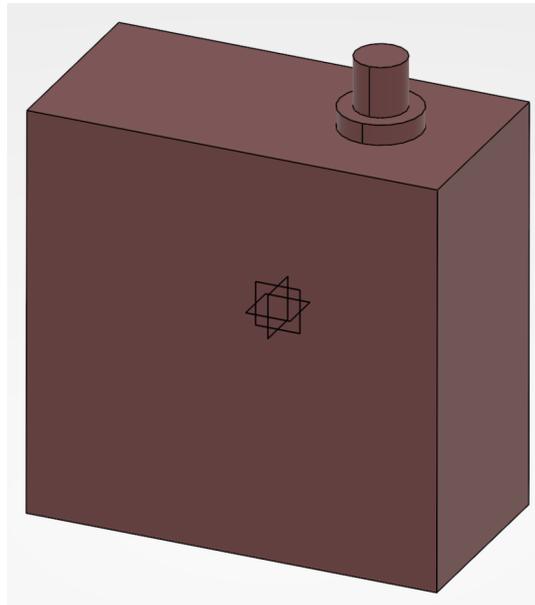
Pour les 3D Part, il vous faudra aussi renommer les instances, dans l'onglet Instance de Properties :



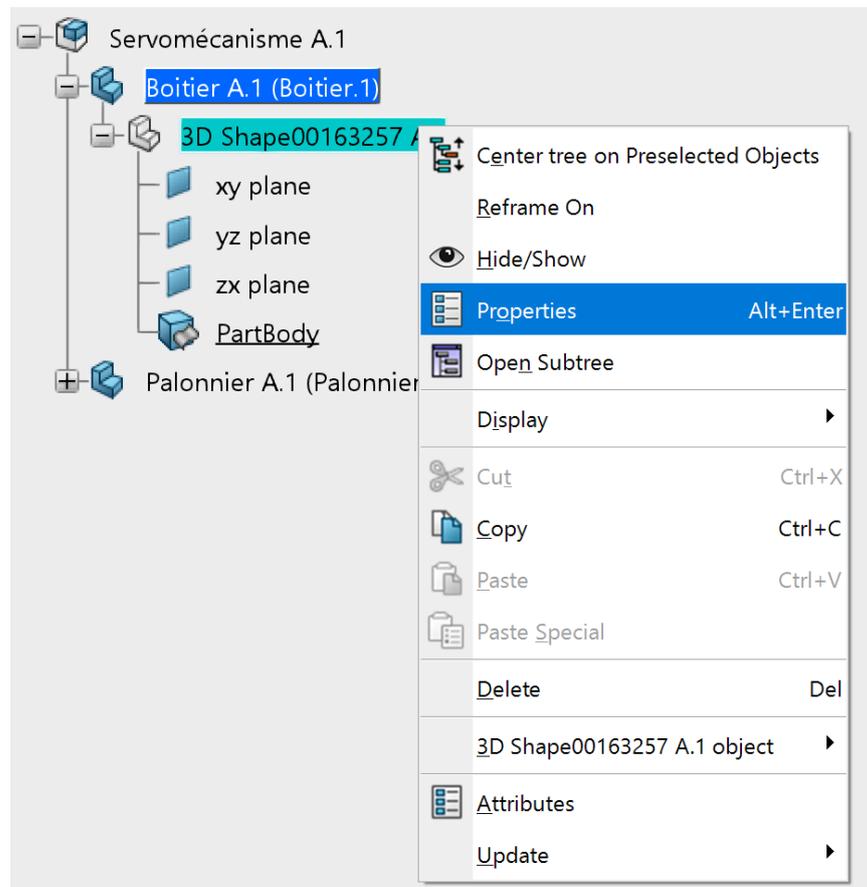
Remarque : contrairement à CATIA v5 les noms des pièces peuvent être changés à tout

moment sans manipulation compliquée car elles ne sont que des titres et non plus les références des pièces dans la base de données. En effet, chaque objet créé dans la 3DExperience possède un identifiant unique attribué automatiquement par le logiciel.

Réalisation du modèle 3D du boîtier.

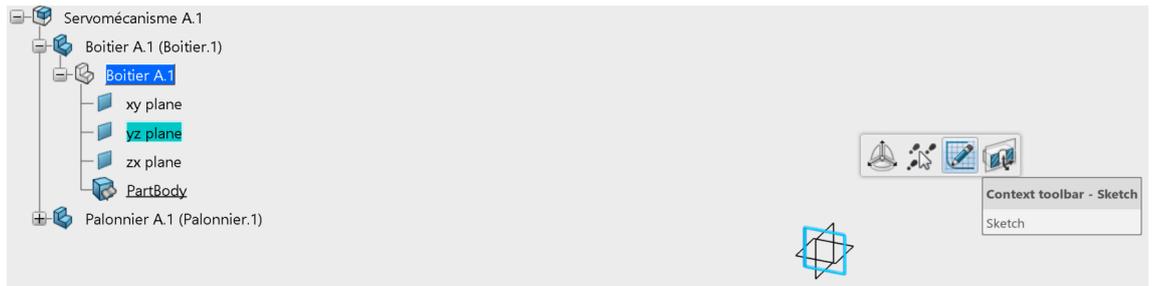


Double-cliquez sur le 3D Part du boîtier dans l'arbre du modèle puis changez au passage le nom du Shape associé en Boîtier en cliquant dessus avec le bouton droit puis sur Properties :



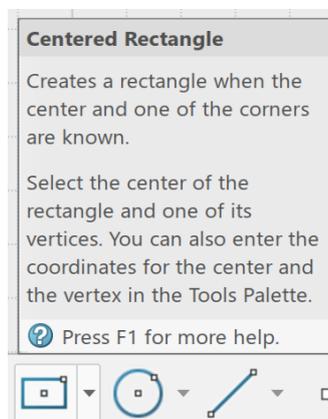
Double-cliquez ensuite sur le 3D Shape pour commencer à modéliser le boîtier, le 3D Shape doit être surligné en bleu, puis cliquez sur un des plans du repère puis sur Sketch dans la

barre d'outils contextuelle Context Toolbar :

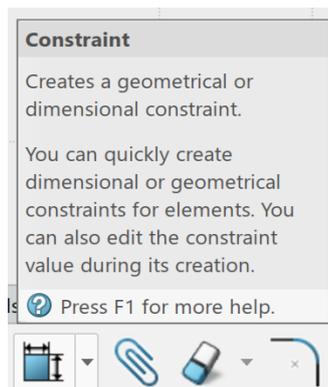


Remarque : travailler directement dans un assemblage peut être pratique pour créer des pièces dont les dimensions sont dépendantes les unes des autres, comme par exemple les deux parties d'un même boitier. Vous pourrez dans ce cas utiliser les géométries de la pièce complémentaire comme références des cotation de la pièce que vous modélisez.

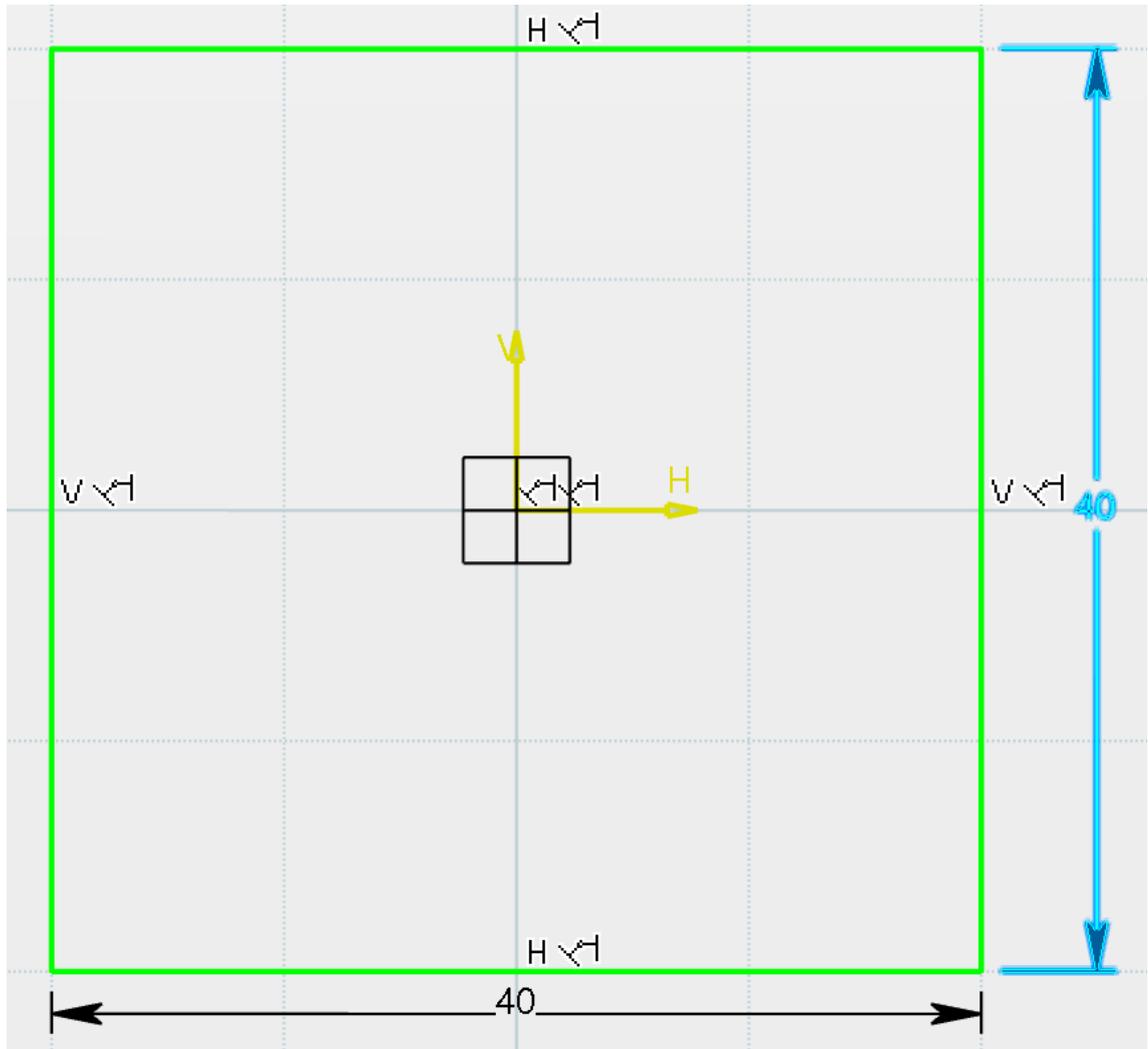
Créez ensuite un carré de 40mmx40mm centré sur le centre du repère avec la fonction Centered Rectangle :



Pour coter le carré, cliquer sur l'icône Constraint ou sur une de droite du rectangle et sur l'icône Constraint qui apparait dans le Context toolbar :

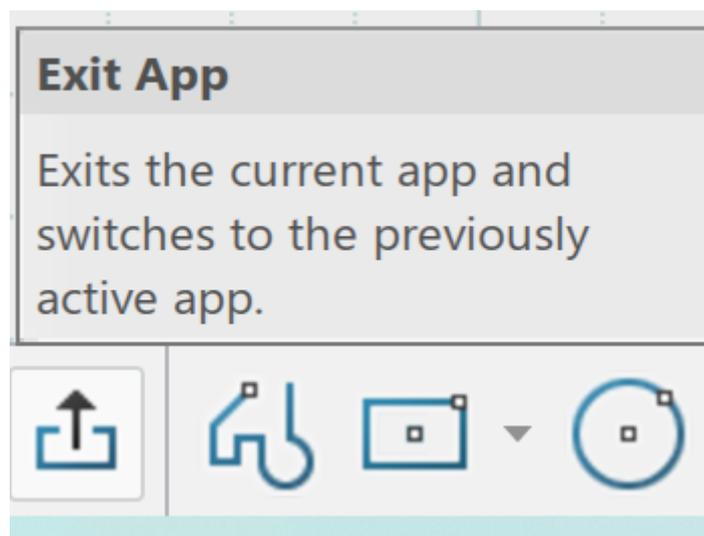


Le résultat doit être le suivant :

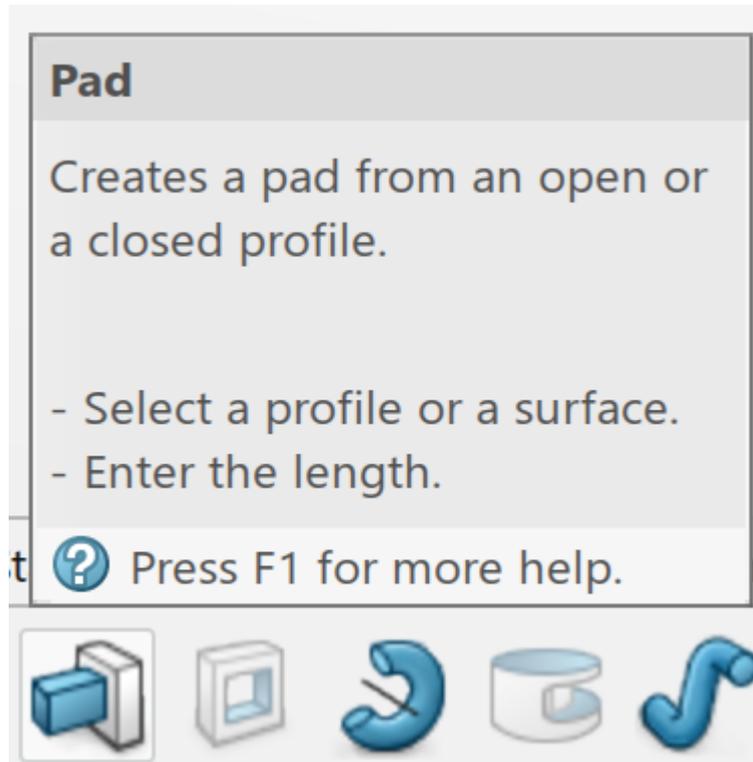


Remarque : prenez l'habitude de travailler avec le Context Toolbar, cette nouvelle fonctionnalité de 3DExperience permet de gagner beaucoup de temps en évitant de nombreux aller/retour de la souris entre le centre de l'écran et les barres d'outils.

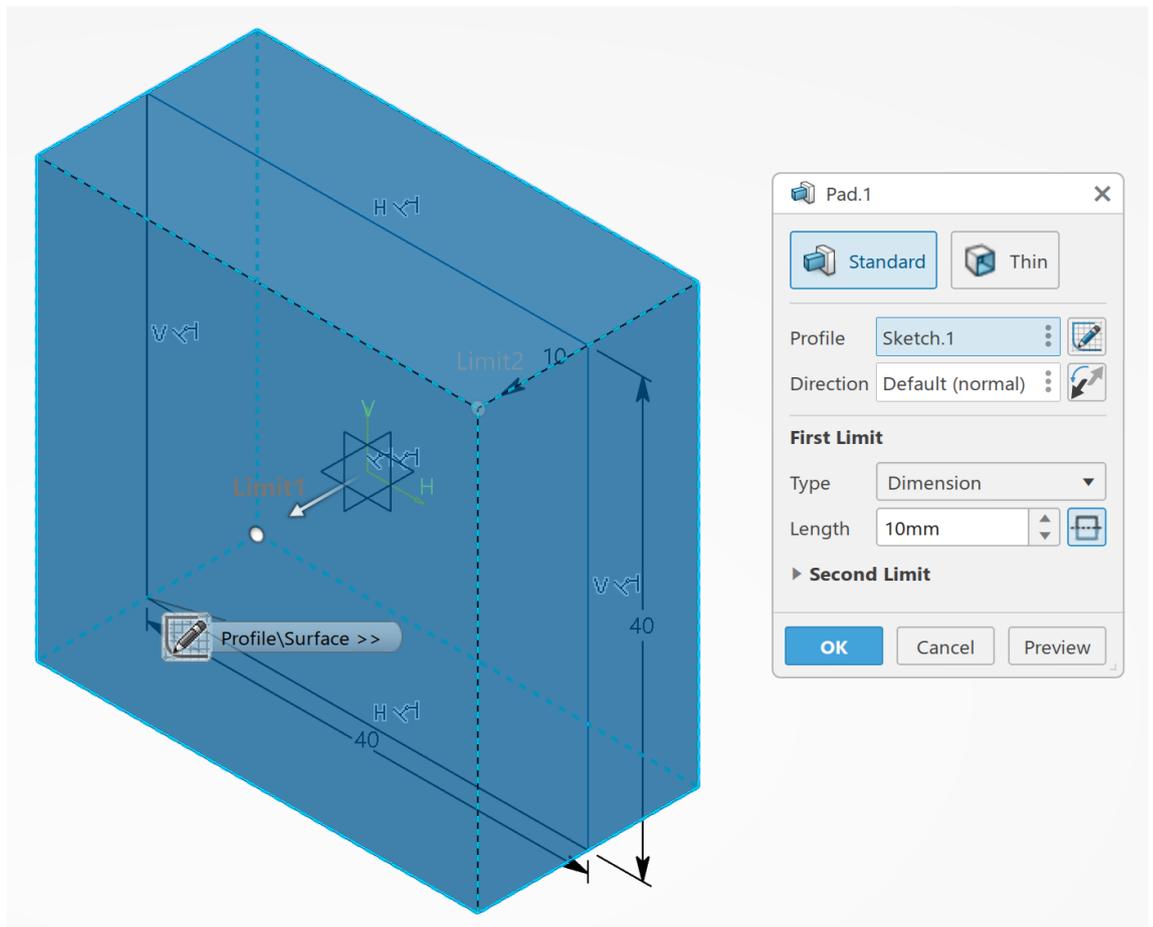
Sortez de l'esquisse en cliquant sur Exit App :



Créer un parallélépipède de 20mm en cliquant sur Pad :



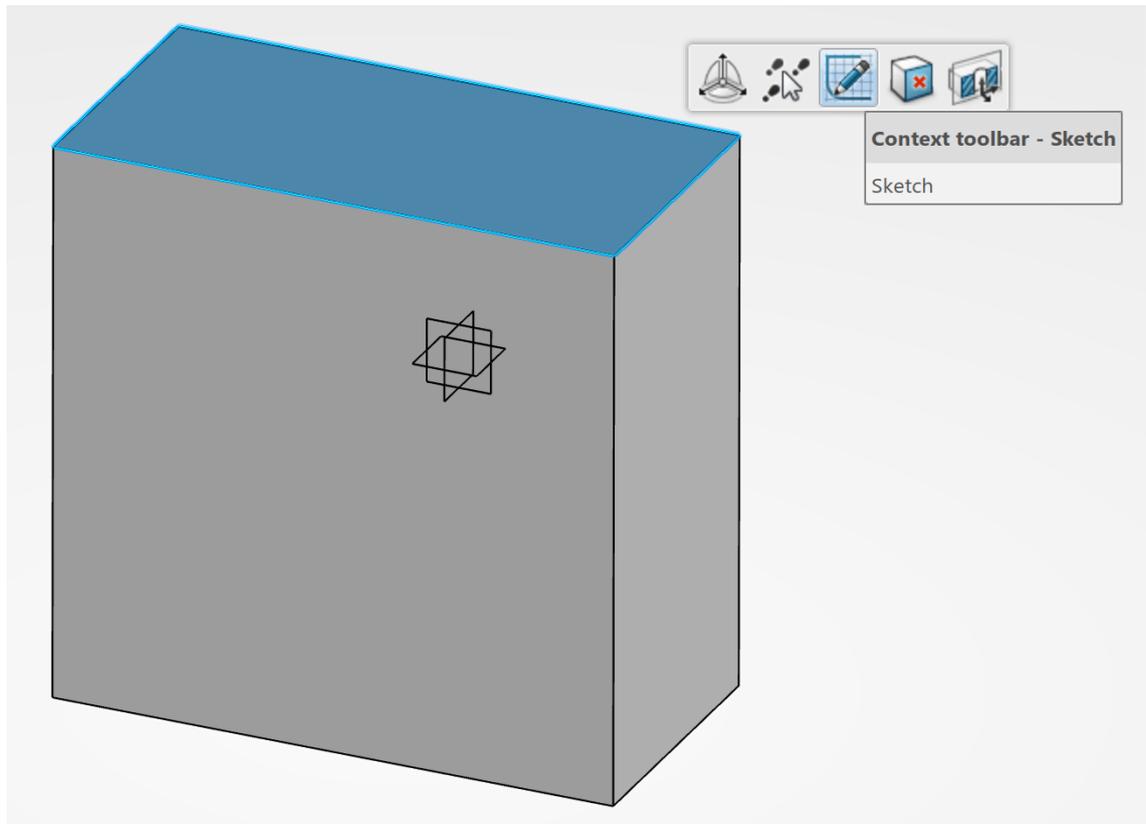
Puis entrez 10mm dans le paramètre Length et cliquez sur l'icone d'extension symétrique située à droite de la case Length de sorte que l'épaisseur de l'extrusion fasse 20mm et que le plan d'esquisse devienne le plan de symétrie du volume :



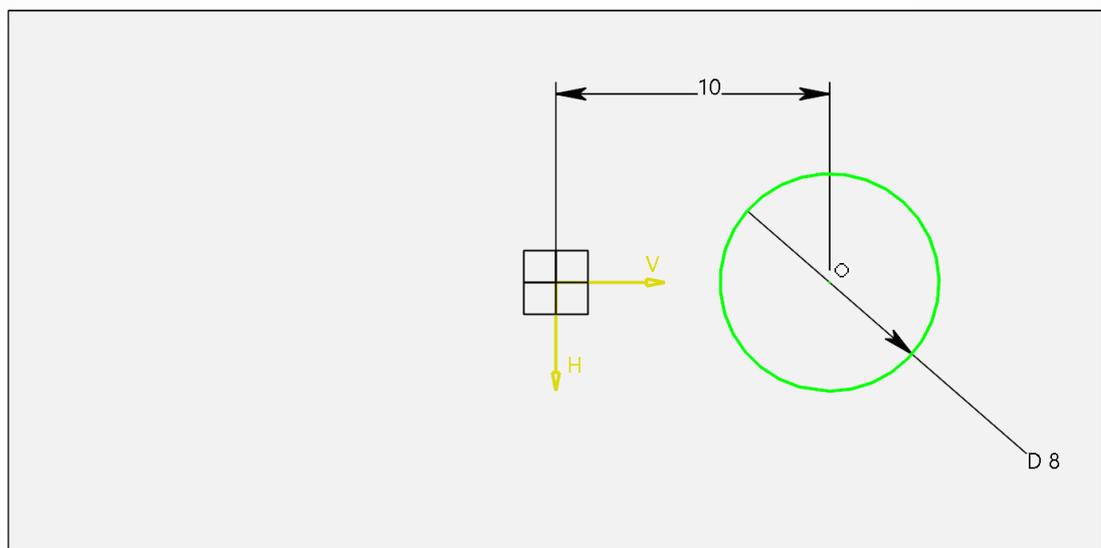
Remarque : veillez à toujours disposer les plans de symétrie de vos modèles sur les plans de

symétrie des volumes, ceci vous fera gagner beaucoup de temps sur les placements des géométries et sur les fonctions de symétrie ou de répétition.

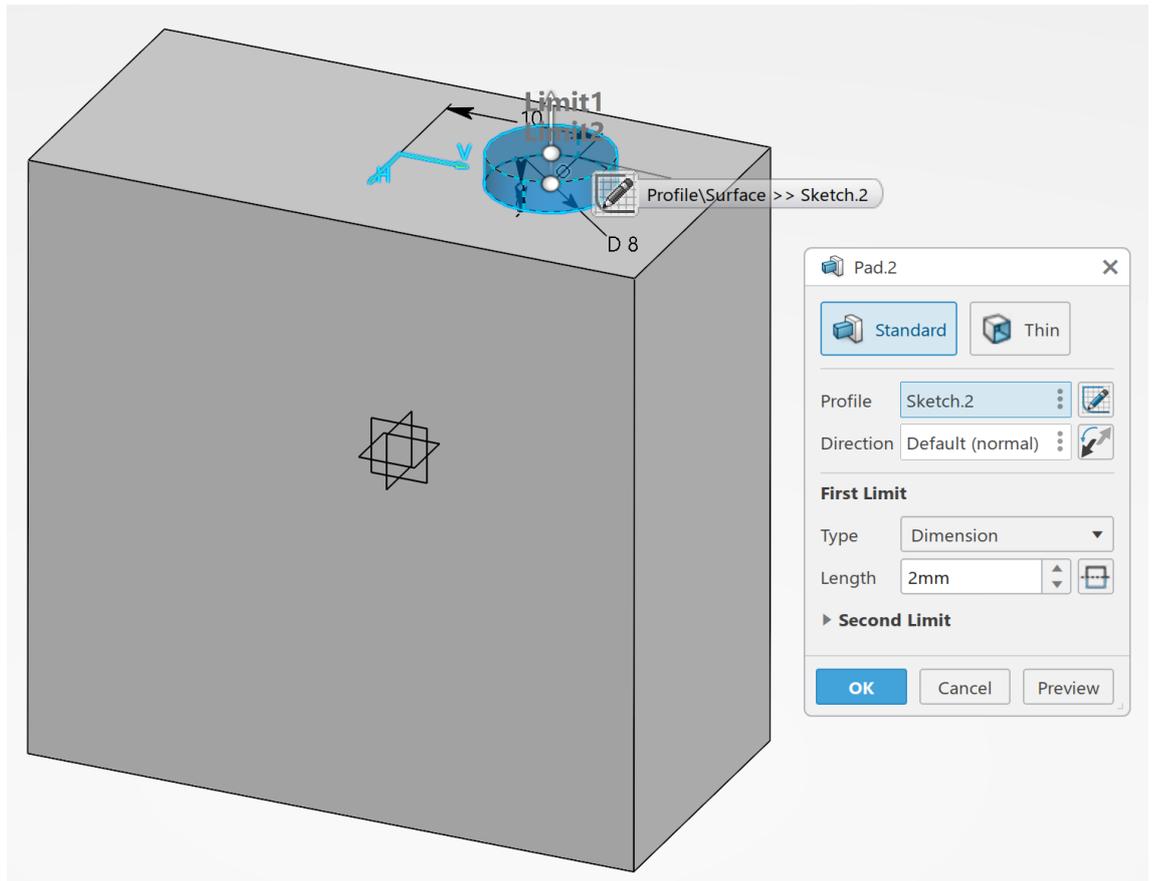
Sélectionnez la face supérieure du parallélépipède et cliquez sur Sketch dans le Context Toolbar :



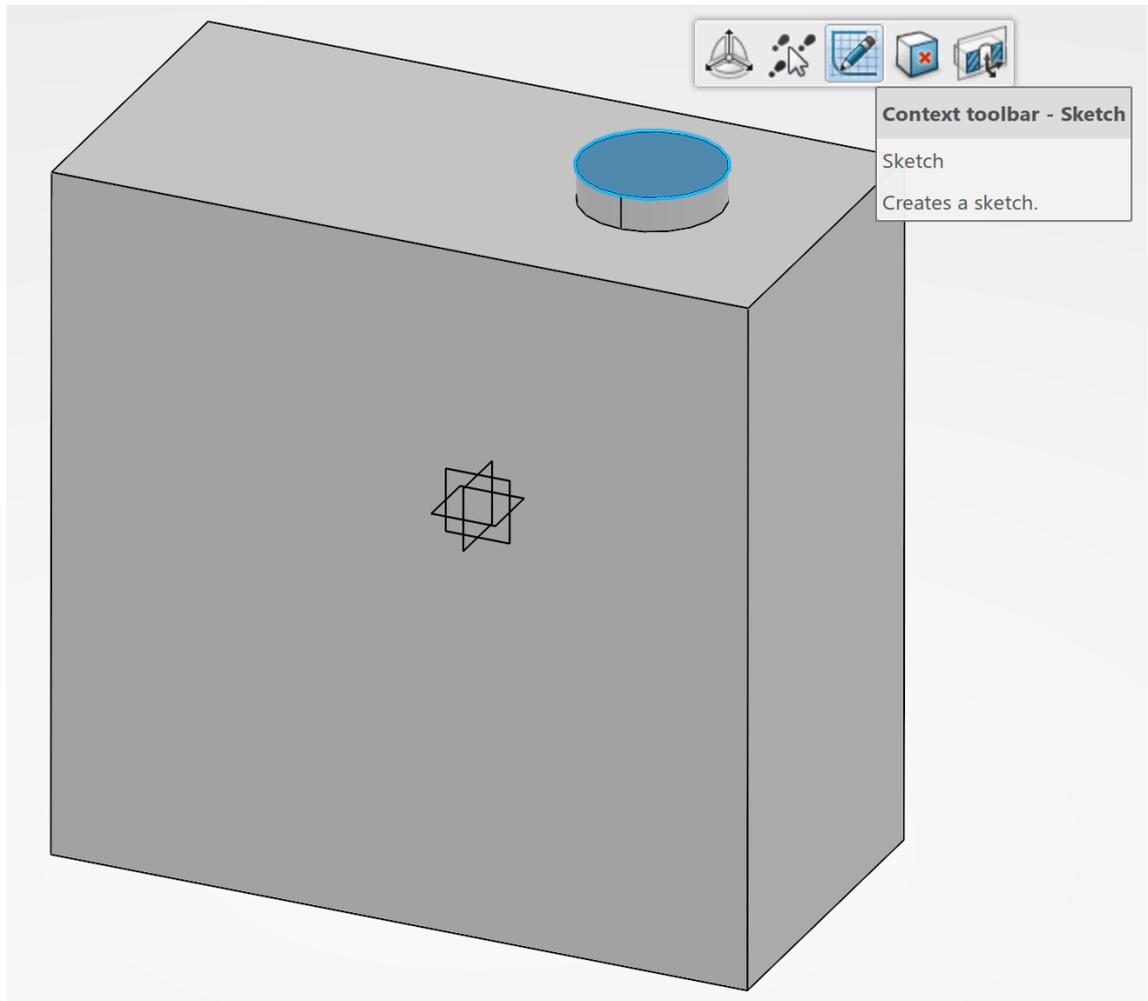
Réalisez l'esquisse suivante puis sortez de l'esquisse :



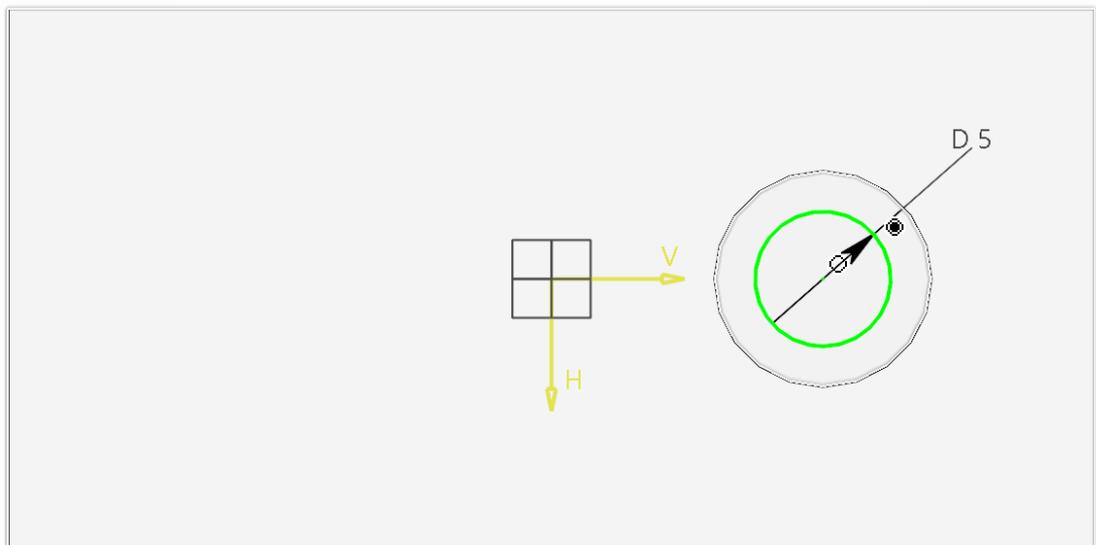
Réalisez une extrusion de 2mm de longueur en cliquant sur l'esquisse puis sur Pad :



Cliquez sur la face supérieur du cylindre créé précédemment puis cliquez sur Sketch.



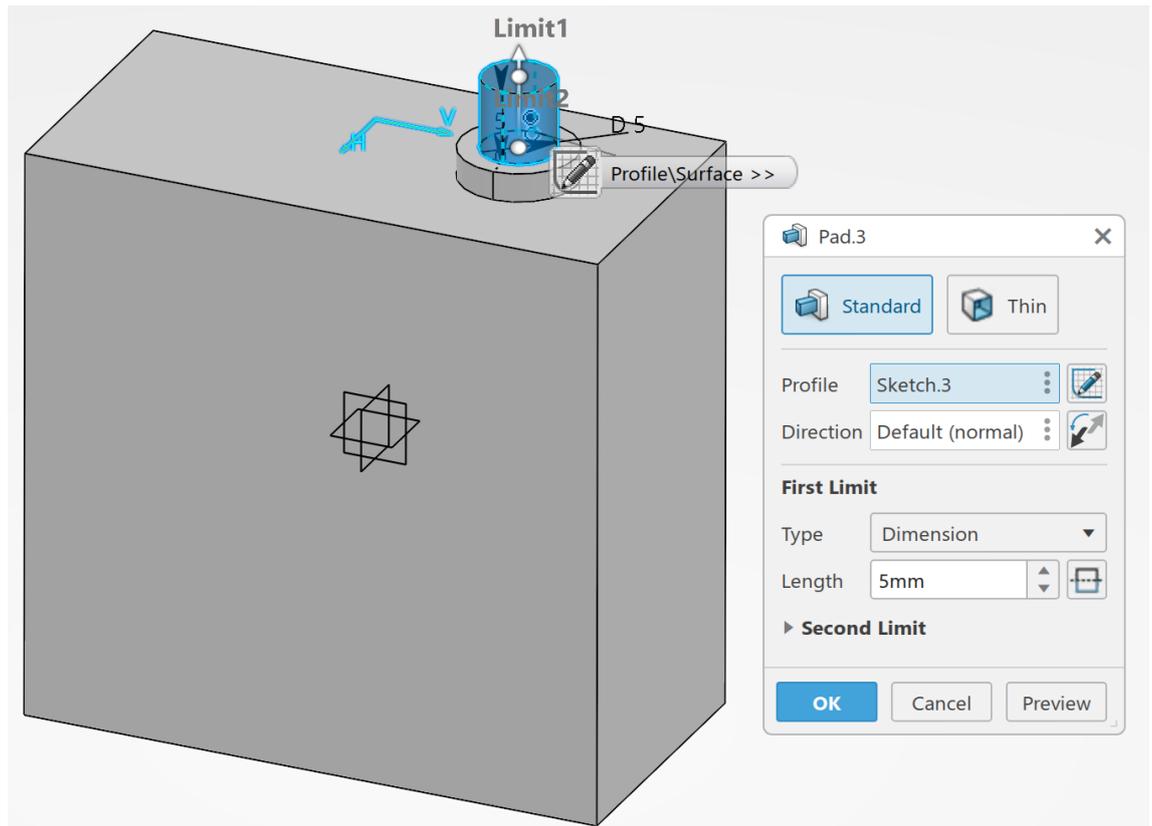
Réalisez l'esquisse suivante :



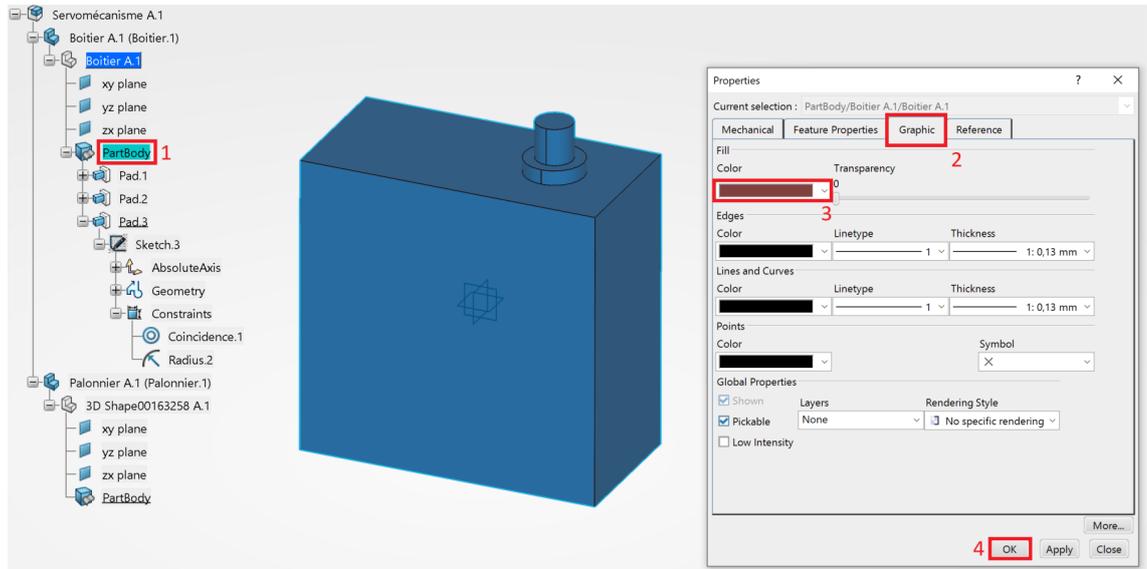
Utilisez la contrainte de concentricité en sélectionnant **Constraint** puis en cliquant sur les deux cercles puis choisir **Concentricity** (vous pouvez aussi sélectionner les deux cercles en maintenant la touche **Ctrl** puis cliquer sur **Constraint** soit dans la barre d'icône, soit dans le menu contextuel puis sélectionnez **Concentricity**). Sortez de l'esquisse :



Réalisez une extrusion de 5mm de longueur en cliquant sur l'esquisse puis sur Pad :



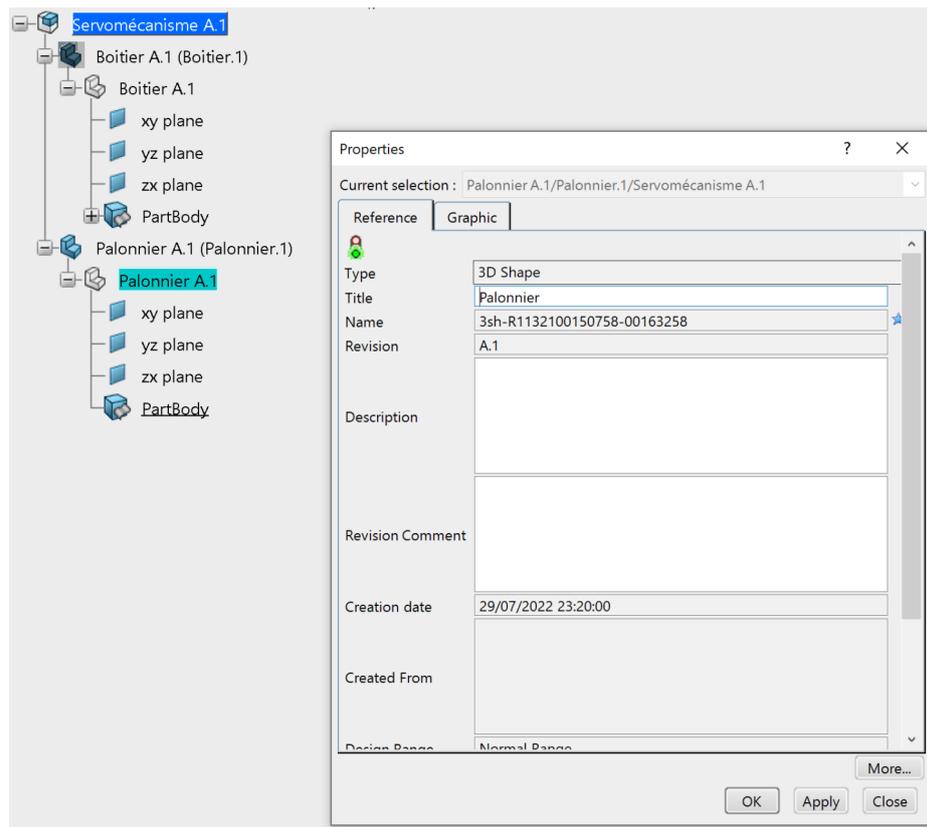
Cliquez avec le bouton droit Propriété sur PartBody dans l'arbre du modèle puis sur l'onglet Graphic puis changez la couleur de la catégorie Fill, en marron par exemple.



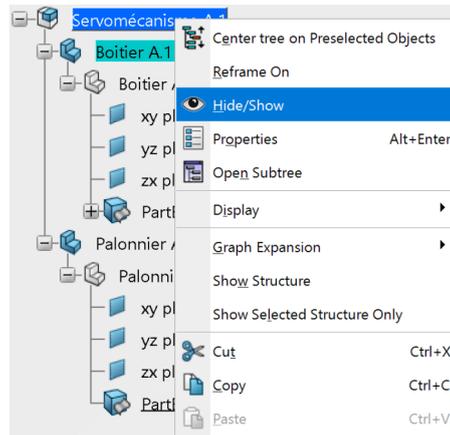
Le boîtier est fini.

### Réalisation du palonnier

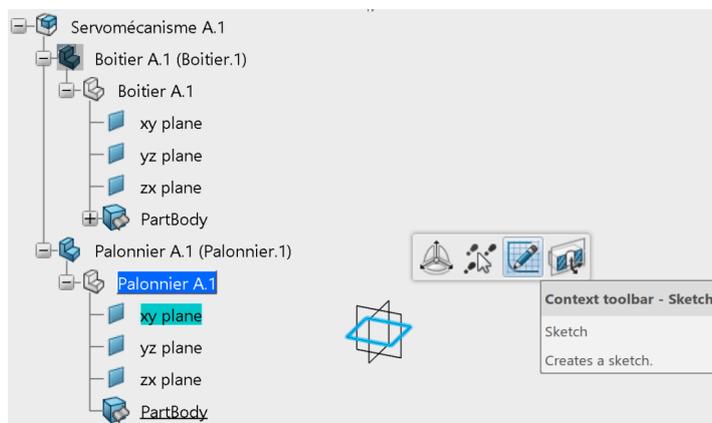
Double-cliquez sur le 3D Part du palonnier dans l'arbre du modèle puis changez au passage le nom du Shape associé en Boîtier en cliquant dessus avec le bouton droit puis sur Properties :



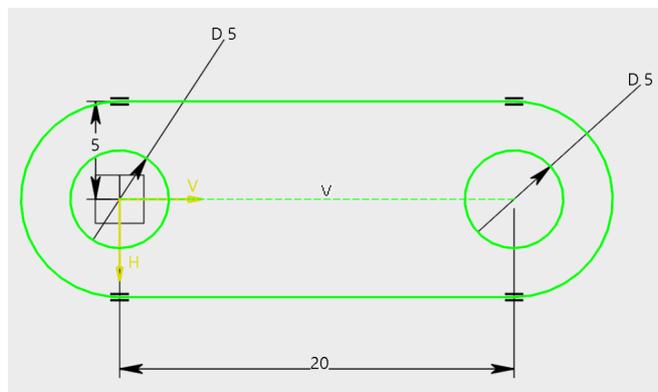
Cliquez avec le bouton droit sur sur le 3D Part ou le 3D Shape du boîtier et cliquez sur Hide /Show pour le masquer :



Double-cliquez ensuite sur le 3D Shape pour commencer à modéliser le palonnier, le 3D Shape doit être surligné en bleu, puis cliquez sur un des plans du repère puis sur Sketch dans la barre d'outils contextuelle Context Toolbar :



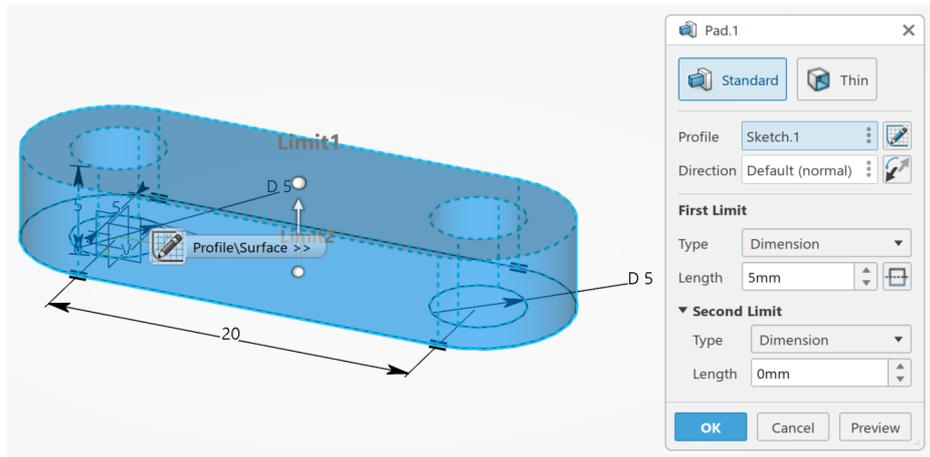
Réalisez l'esquisse suivante :



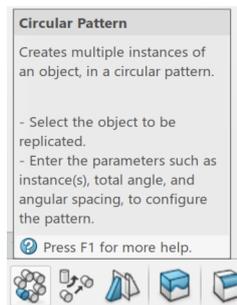
Pour ce faire vous pouvez utiliser la fonction Elongated Hole :



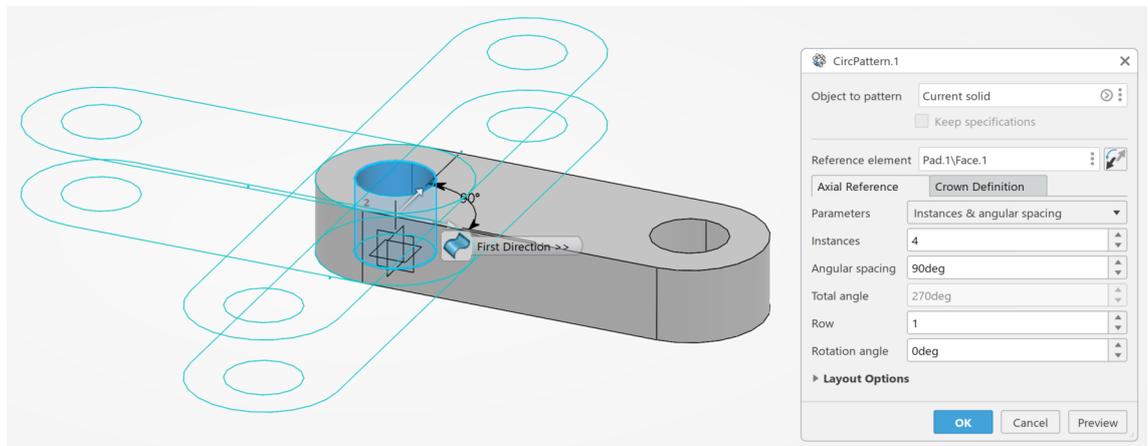
Sortez de l'esquisse puis créez un Pad de 5mm avec l'esquisse créée :



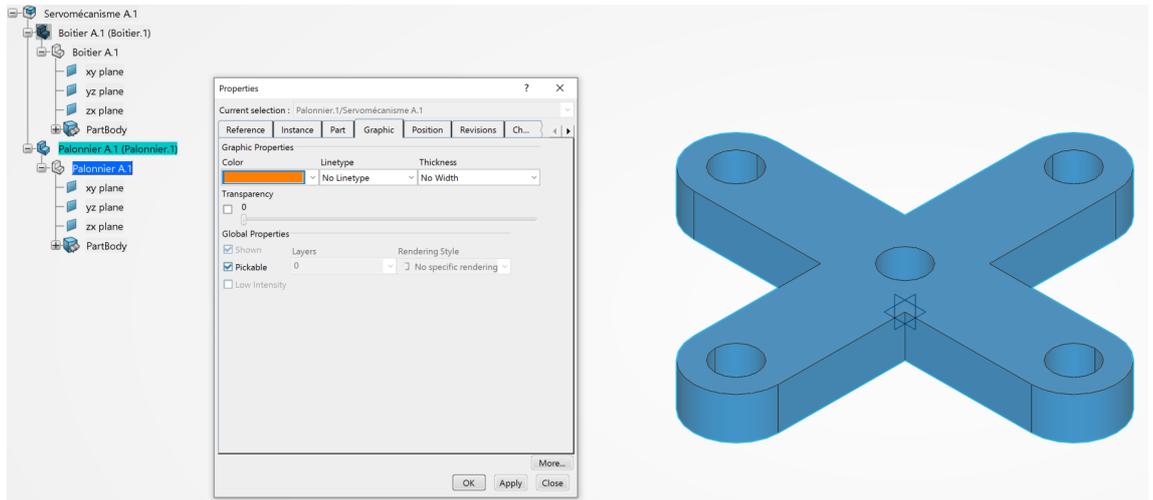
Ensuite sélectionnez la fonction Circular Pattern :



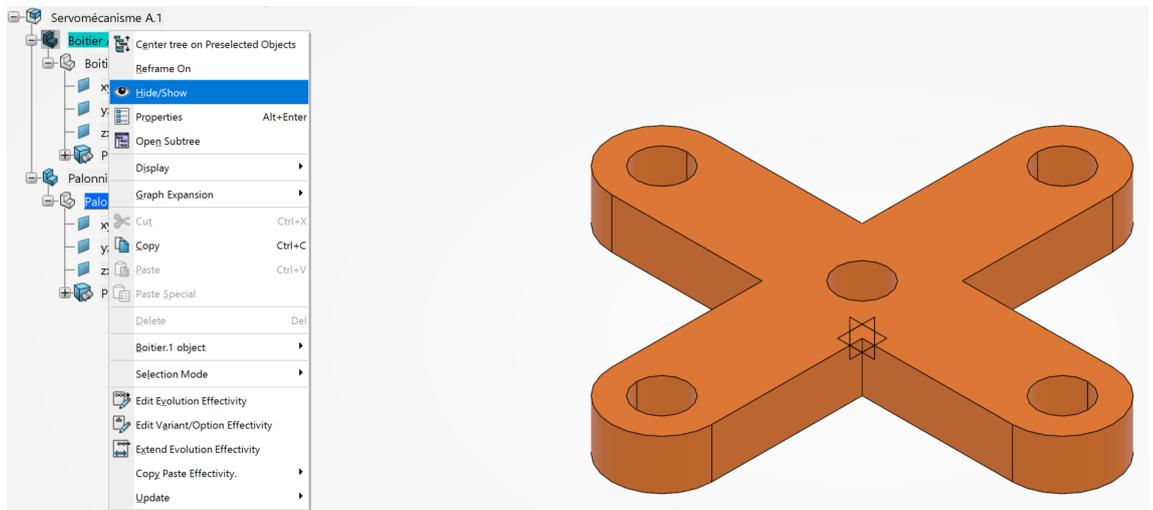
Puis sélectionnez le face intérieure du cylindre de gauche comme Reference Element, puis choisissez 4 Instances espacées de 90° dans Angular Spacing :



Ensuite cliquez avec le bouton droit sur le Palonnier puis choisissez Properties, Graphic puis une couleur et OK :

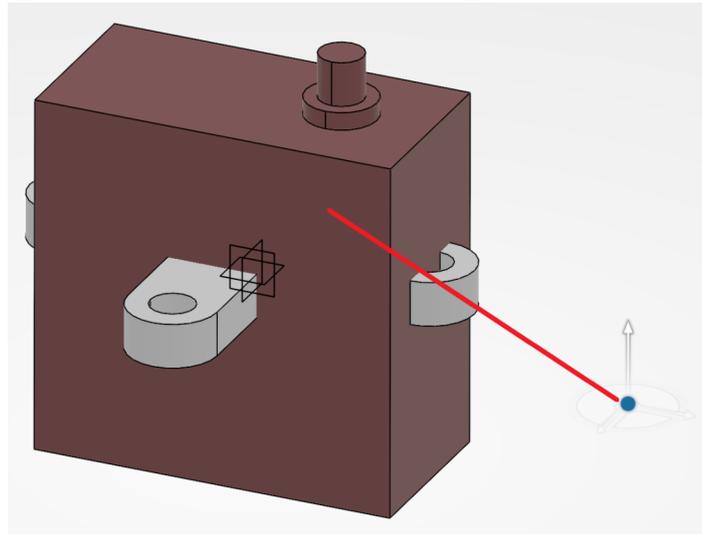


Cliquez avec le bouton droit sur le Boitier puis sur Hide/Show pour l'afficher à nouveau

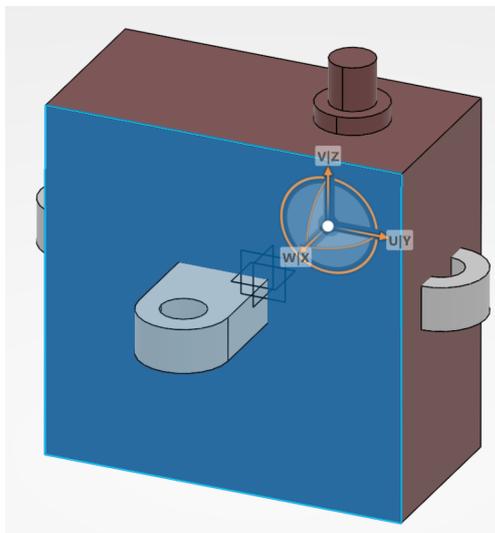


### Création du mécanisme

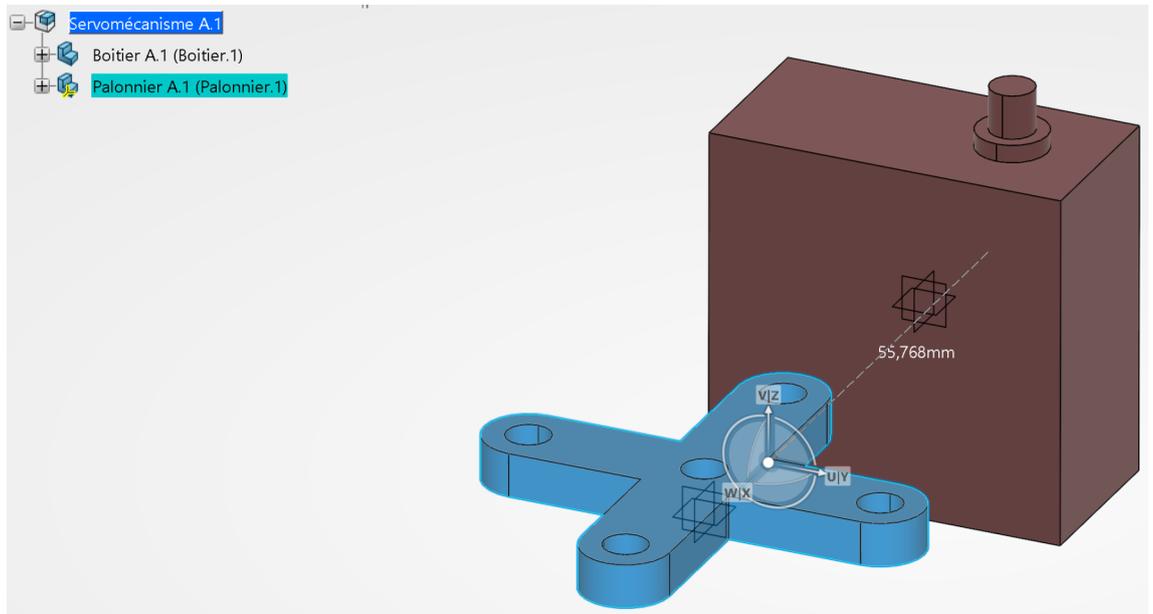
Réalisez une liaison pivot dans Assembly Design entre le boitier et le palonnier réalisée. Pour faciliter la création de cette liaison vous pouvez dans un premier temps déplacer le palonnier de sorte à ce qu'il ne soit plus en contact avec le boitier. Pour ce faire, il vous faut faire un glisser/déplacer de la boussole sur un des plans d'une des pièces. Pour réaliser cette opération il faut cliquer sur le centre de la boussole située en bas à droite puis déplacer la boussole en maintenant le bouton de la souris enfoncé et la déposer sur un des plans d'une des pièces et ensuite relâcher la souris :



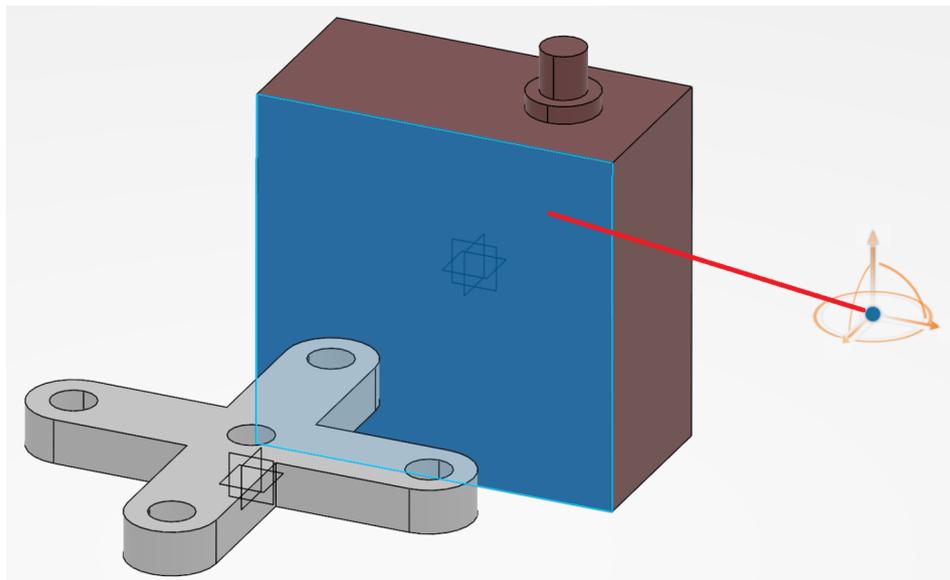
Le plan qui reçoit la boussole devient alors bleu :



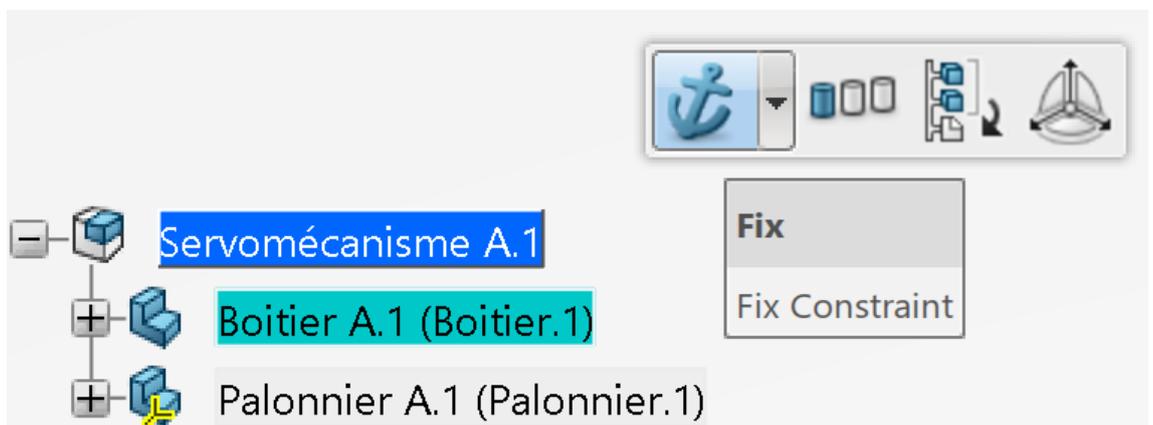
Pour déplacer une des deux pièces par rapport au repère d'assemblage : sélectionnez la pièce dans l'arbre du modèle en faisant un seul clic gauche dessus (la pièce devient alors bleue dans la 3D et dans l'arbre), puis réalisez un clic gauche sur un des axes du repère de la boussole tout en maintenant le bouton enfoncé pour faire translater la pièce sélectionnée selon un axe (si vous sélectionnez un arc du repère une rotation est effectuée et un plan, une translation selon le plan) :



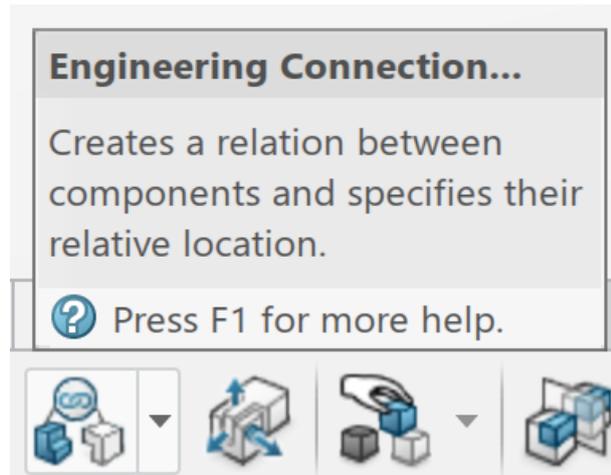
Pour remettre la boussole à sa position initial, il suffit de sélectionner son centre avec le clic gauche, maintenir le bouton enfoncé et déplacer ce point sur un espace libre :



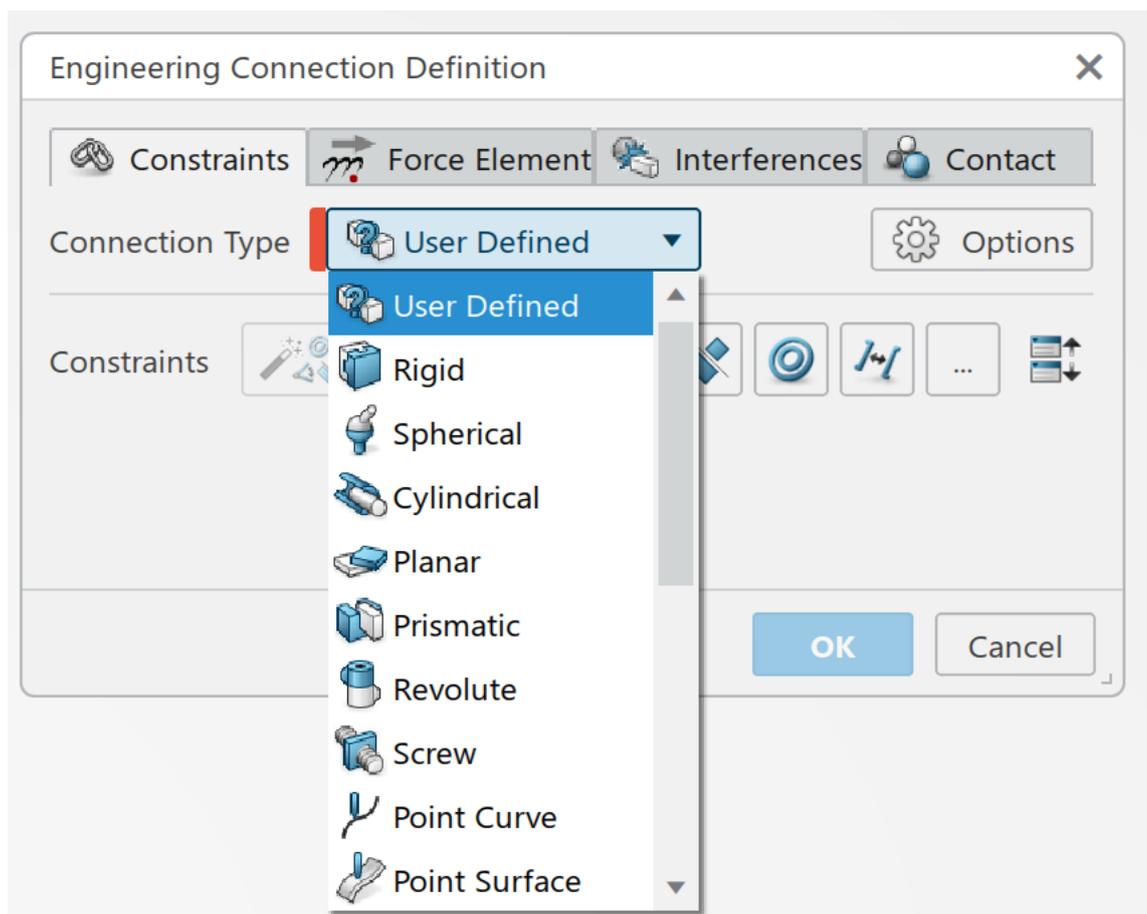
Avant de réaliser la liaison, nous allons définir la pièce qui sera la référence d'immobilité lors du positionnement des pièces entre elles. Pour ce faire, faites un clic gauche sur le boitier dans l'arbre du modèle puis cliquez sur Fix dans le menu contextuel qui apparaît :



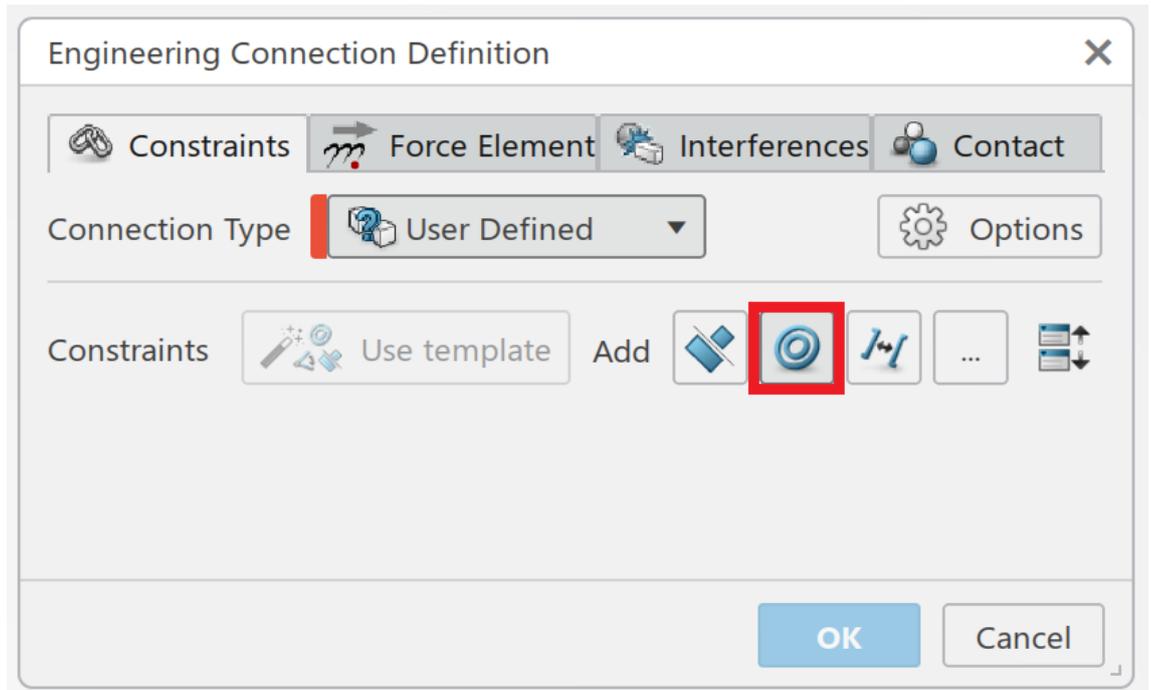
Pour ensuite réaliser la liaison pivot, il faut cliquer sur l'icône Engineering Connection de l'onglet Assembly :



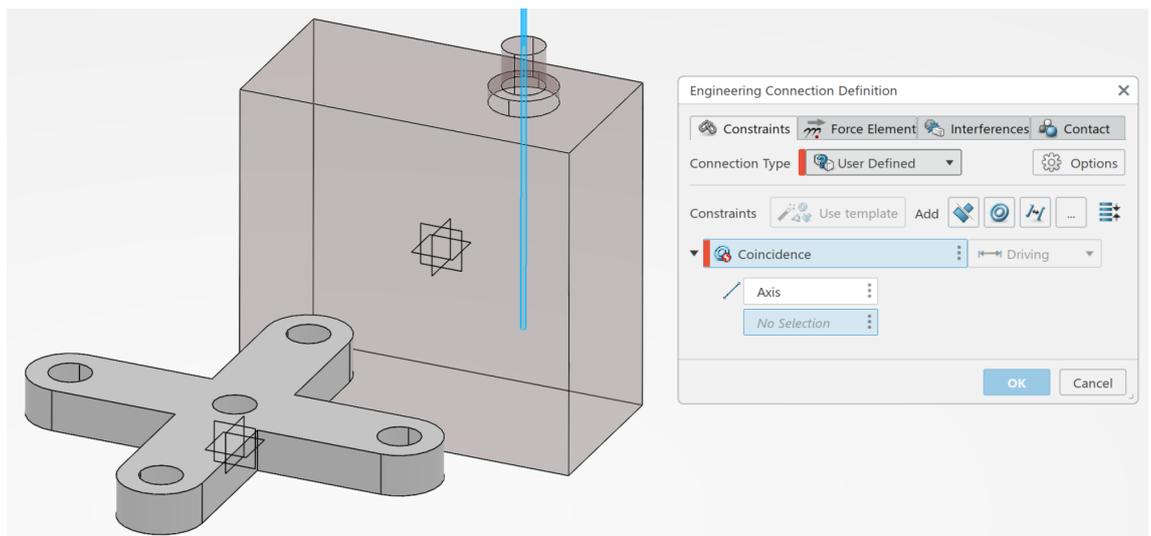
Ensuite, deux choix s'offrent à vous, soit sélectionner Revolute dans le menu déroulant Connection Type qui correspond au mode guidé, soit cliquer sur User Defined et définir vous mêmes les géométries qui serviront à définir la liaison. Ce mode reconnaît automatiquement le type de liaison cinématique selon les choix opérés par l'utilisateur. Nous choisirons ce mode car il est moins astreignant que l'autre mode qui impose de sélectionner les géométries dans un ordre préétabli :



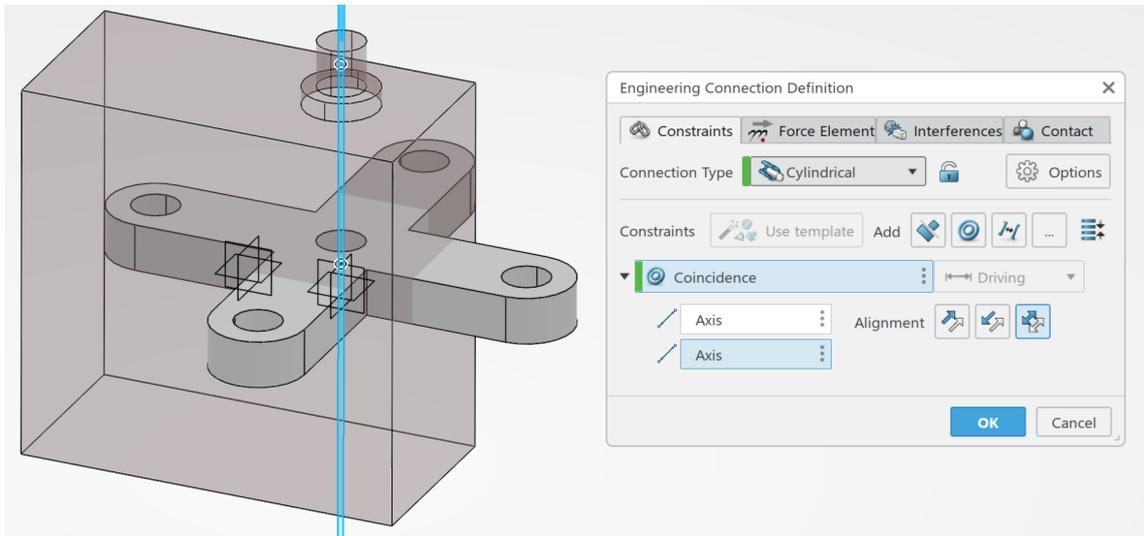
Cliquez sur Coïncidence :



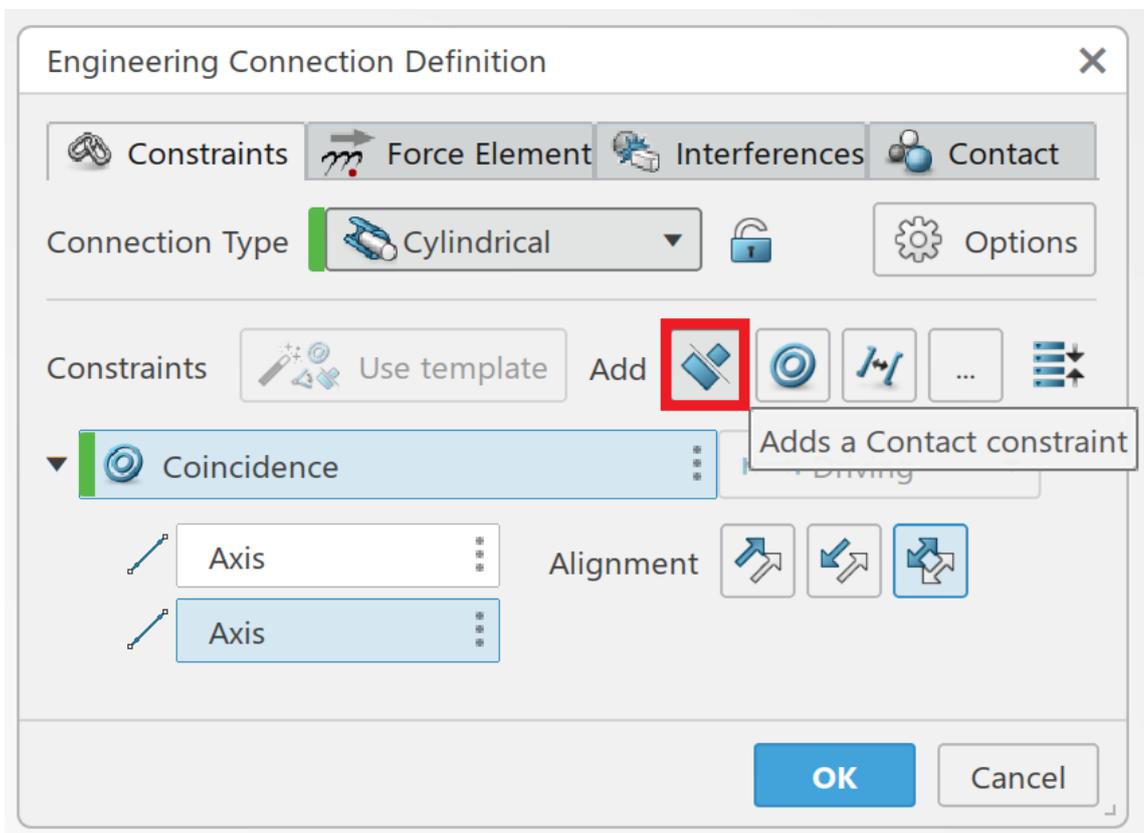
Puis cliquez sur le flan du cylindre supérieur du boîtier pour sélectionner son axe de révolution :



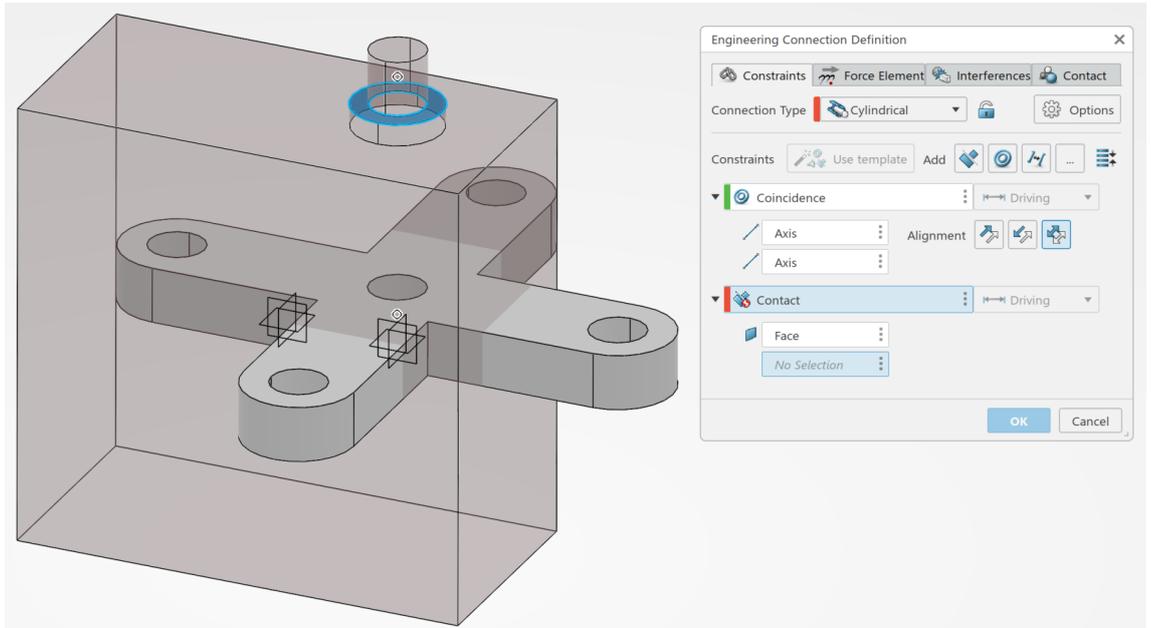
Puis cliquez sur le flan du cylindre central du palonnier pour sélectionner son axe central :



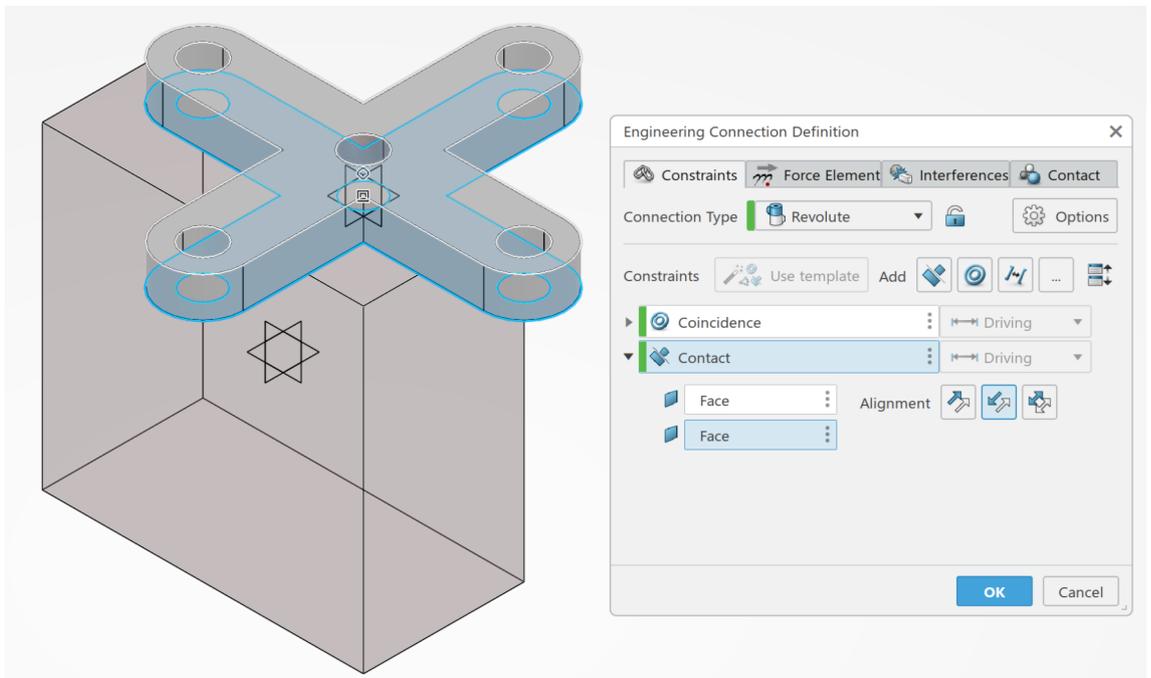
Puis sélectionnez Contact Constraint dans les Constraints :



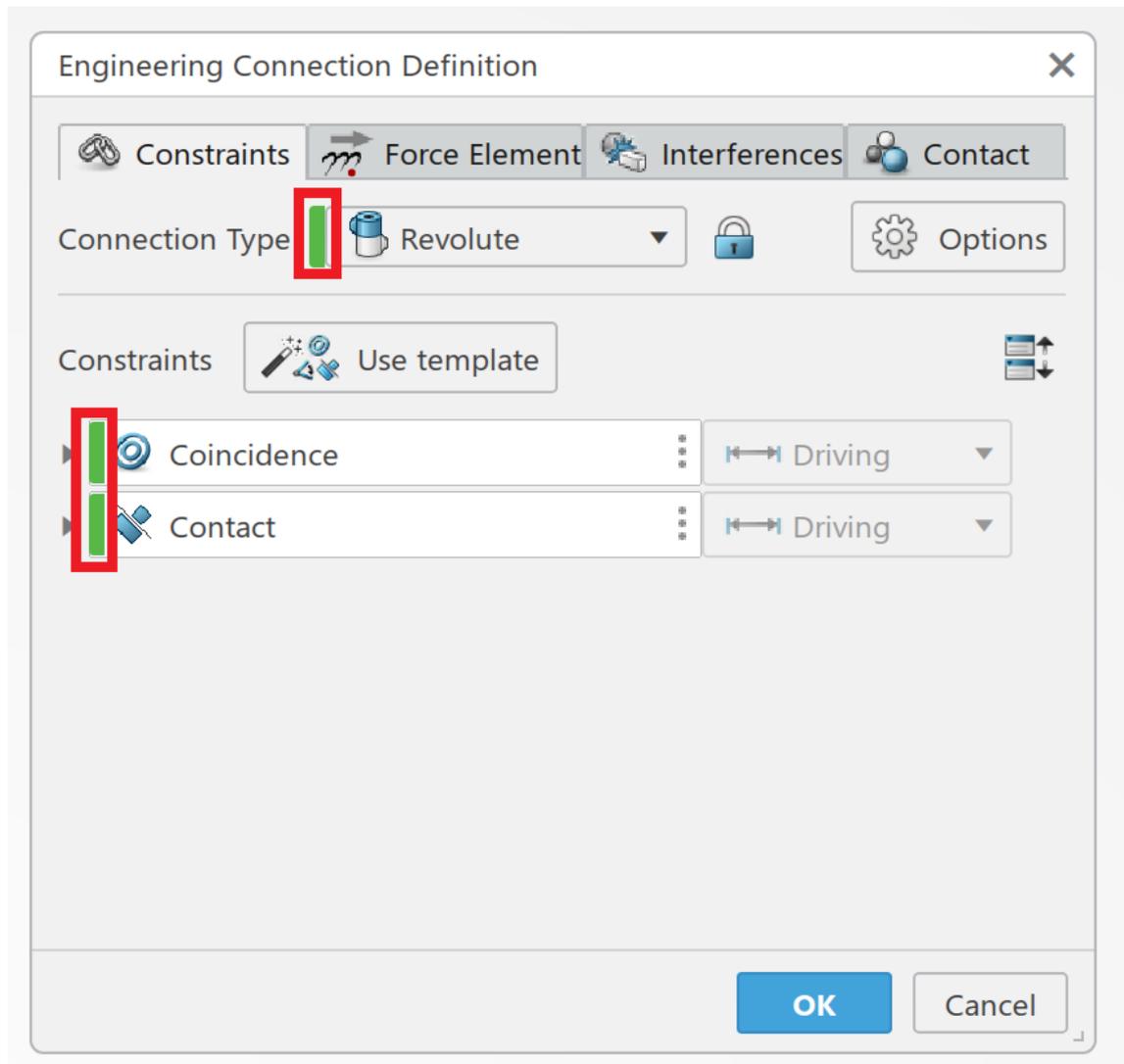
Puis sélectionnez l'épaulement du haut du boîtier :



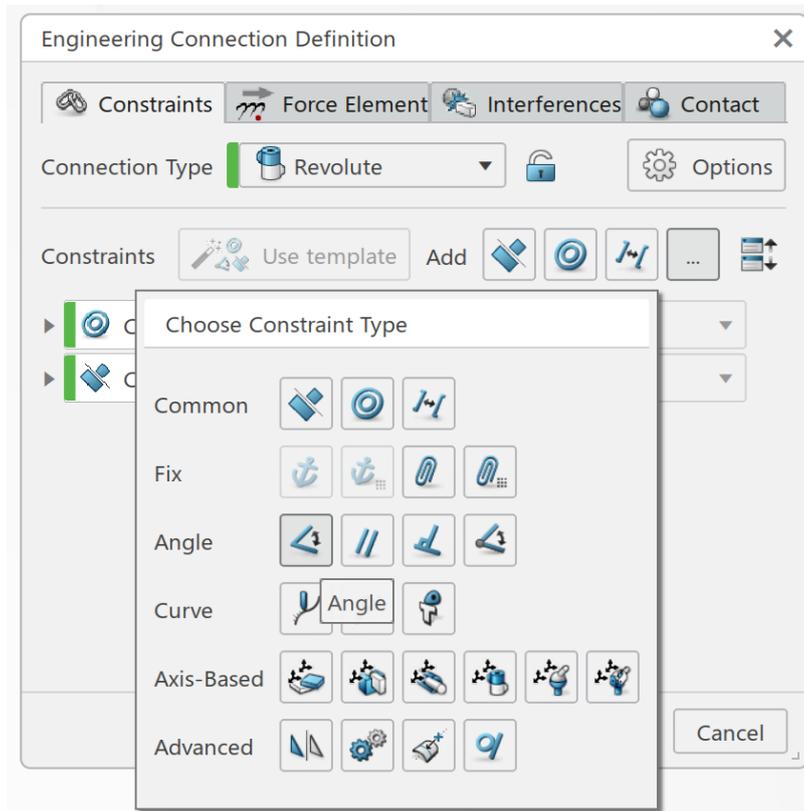
Puis sélectionnez la face inférieure du palonnier :



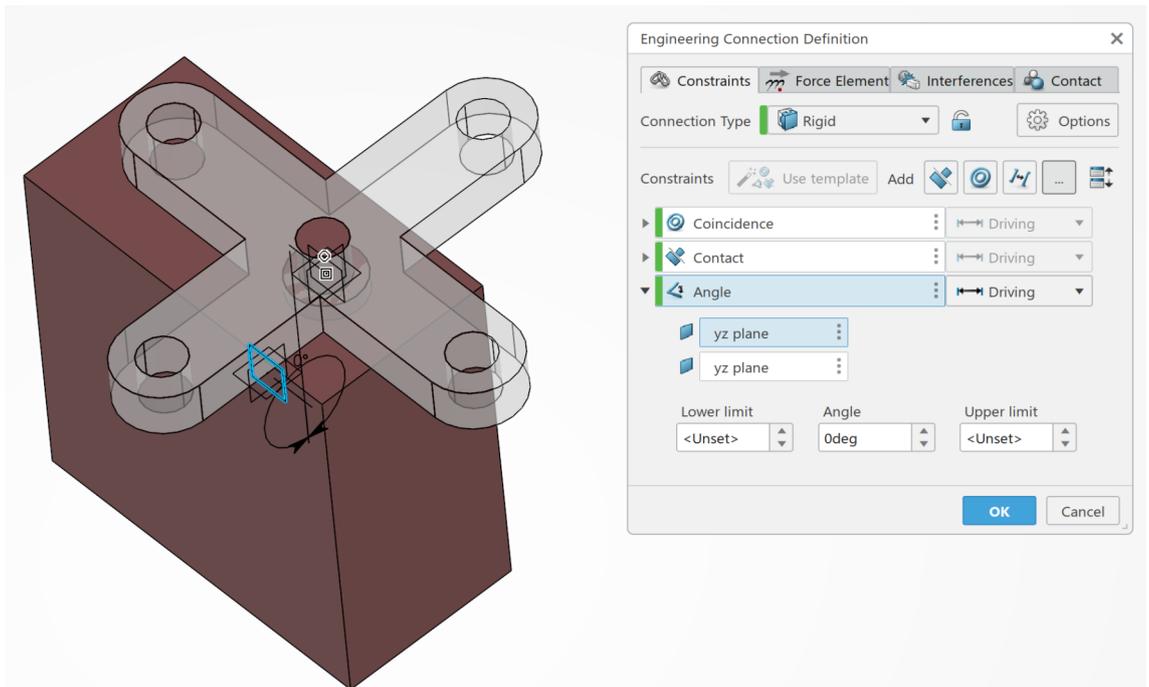
La liaison pivot est terminée puisque l'axe commun de rotation est défini ainsi que l'arrêt en translation, vous pouvez donc cliquer sur OK. Si tout est correctement défini, tous les indicateurs doivent être verts :

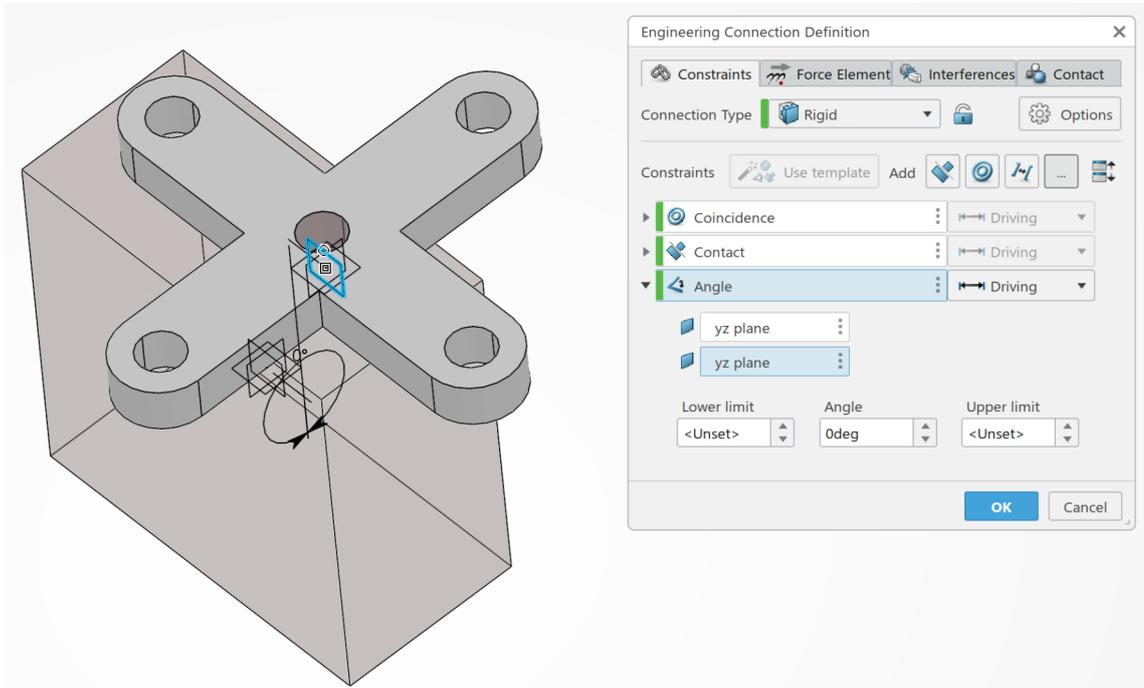


Mais afin que la rotation du palonnier puisse être contrôlée par la simulation il nous faut rajouter un angle contrôlé entre un plan du palonnier et un plan du boîtier. Pour ce faire, cliquez sur l'icône permettant de voir toutes les contraintes possibles dans Constraints, puis sur Angle :

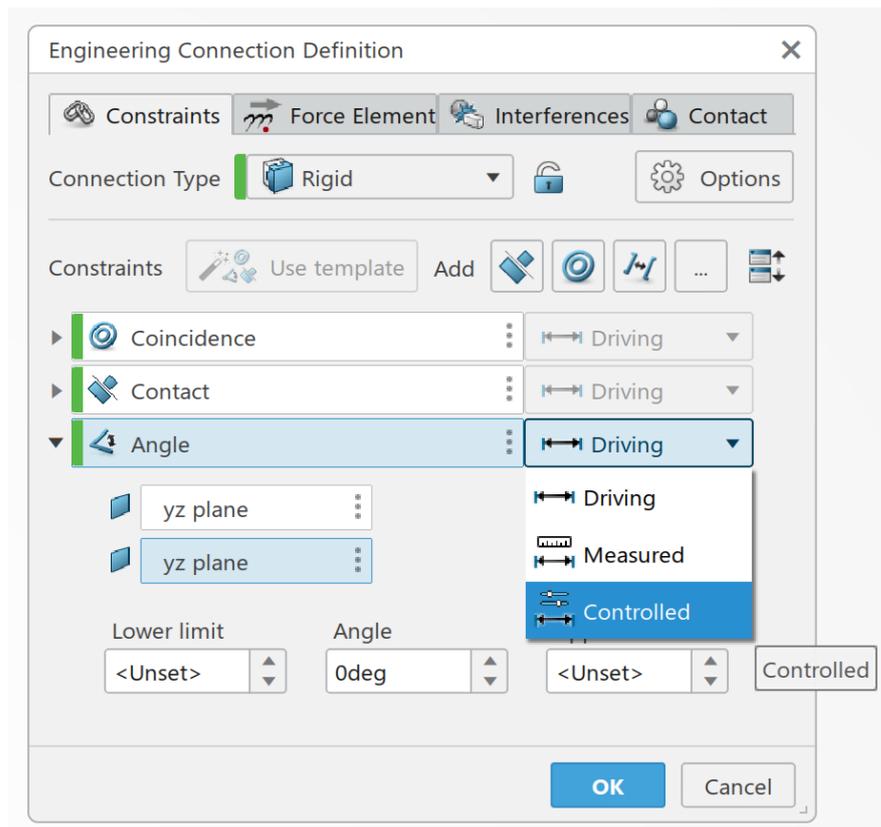


Ensuite, sélectionnez ces deux plans parallèles (ceux des repères de chaque pièce) :

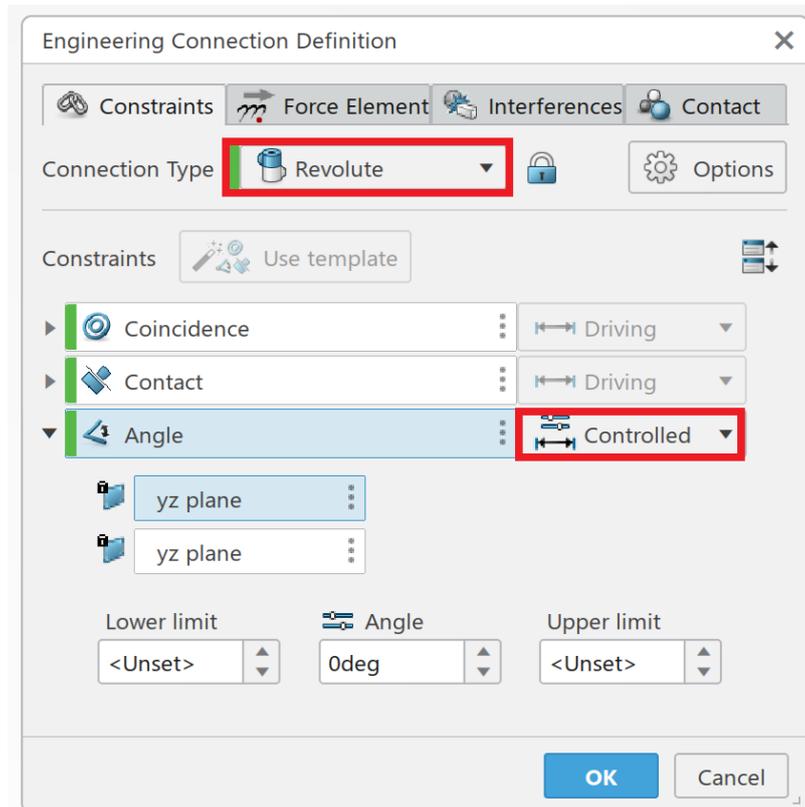




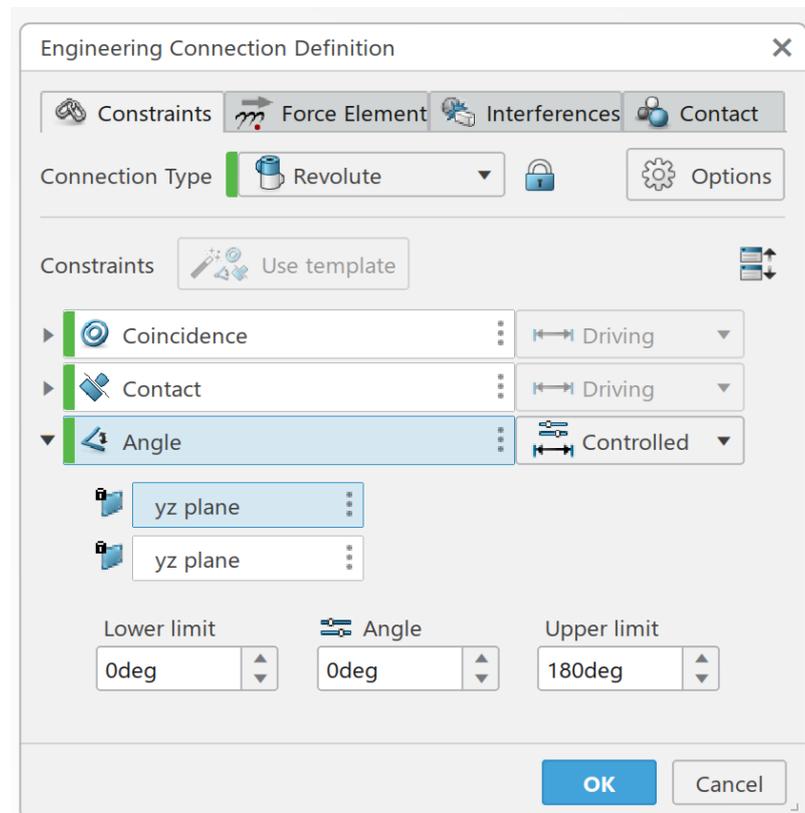
Puis changez le mode de contrainte en Controlled de façon à pouvoir piloter cet angle dans la simulation :



Vérifiez bien que la liaison est toujours affichée en Revolute et que tous les indicateurs sont verts :



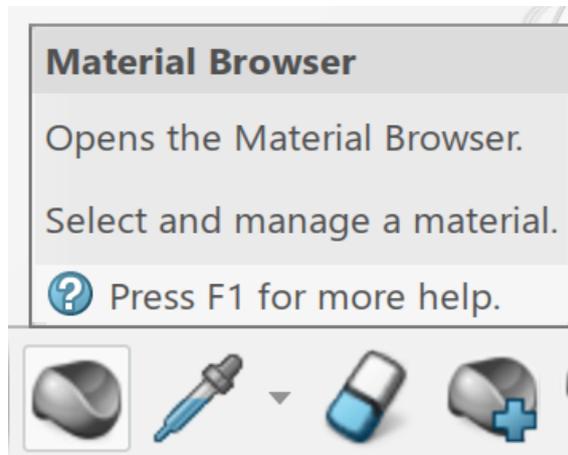
Ensuite, définissez la Lower limit de l'angle à 0° et la Upper Limit à 180°, puis cliquez sur OK et encore sur Ok pour choisir le type d'interférence par défaut :



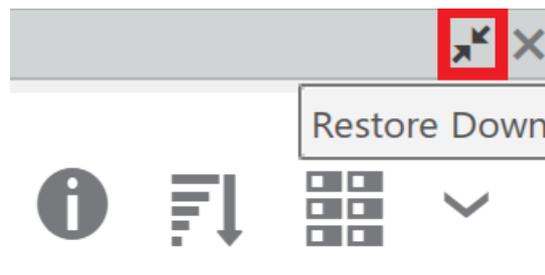
Remarque : il aurait également été possible de réaliser cette liaison dans Mechanical System Design qui possède les mêmes outils d'assemblage. Cette App est dédiée à la réalisation de systèmes cinématiques et dynamiques qui sont destinés à être simulés ensuite, elle est donc

plus riche et donc plus complexe que Assembly Design.

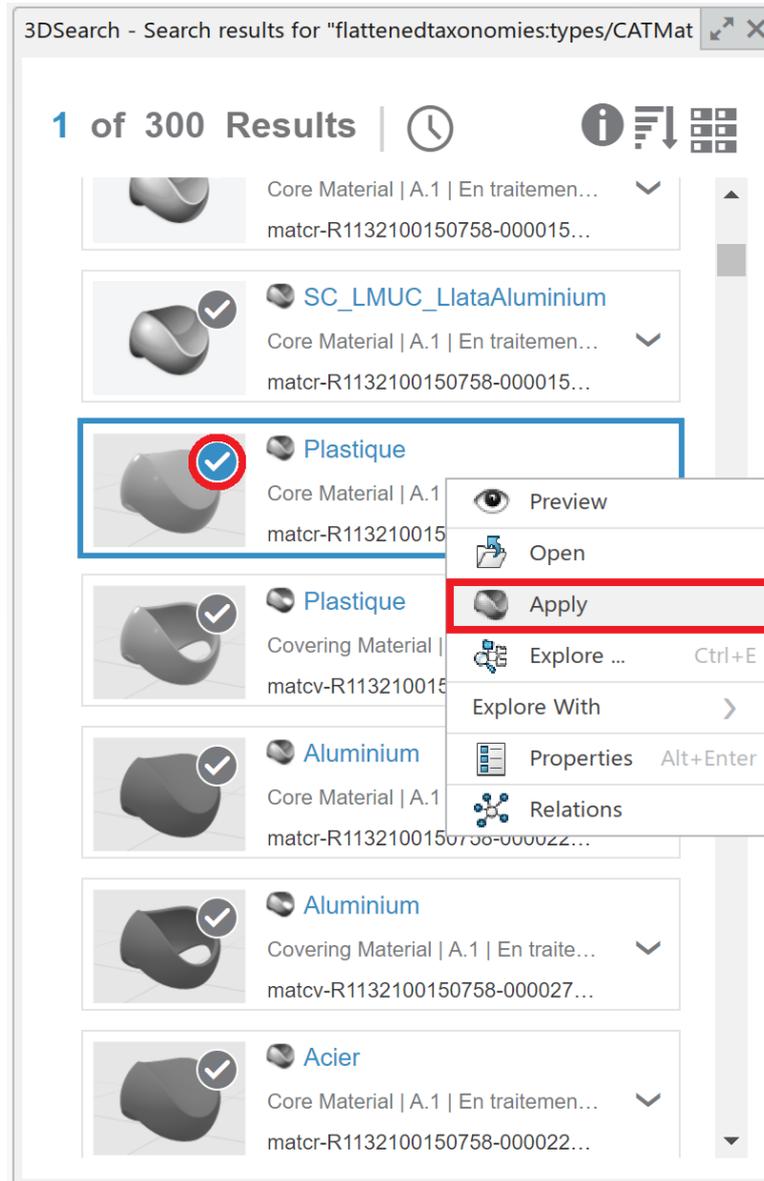
L'étape suivante consiste à appliquer des matériaux. Cette étape peut être réalisée à n'importe quel moment, par exemple lors de la création des modèles des pièces. Pour ce faire, dans l'app Assembly Design ou Mechanical System Design, allez dans l'onglet Tools puis cliquez sur Material Browser :



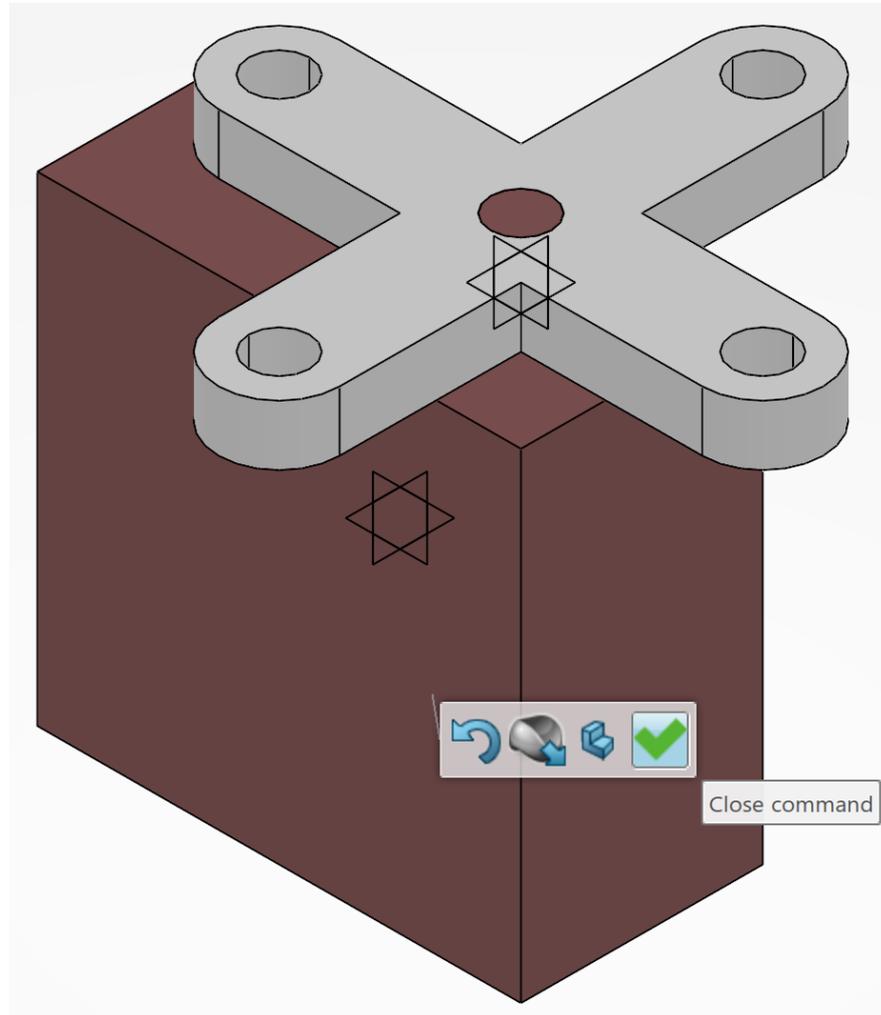
Pour visualiser simultanément votre modèle et la fenêtre des matériaux, vous pouvez réduire la fenêtre en cliquant sur les deux flèches opposées situées en haut à droite :



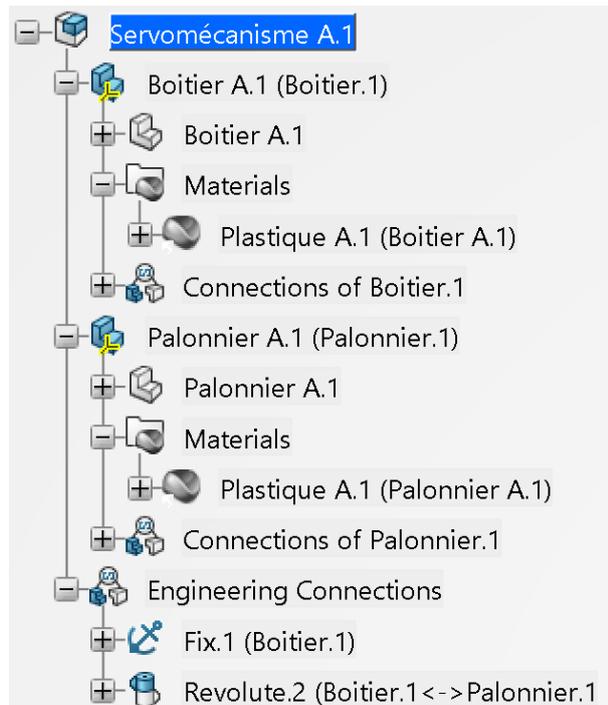
Cherchez un matériau (ce choix n'a pas d'importance car nous n'allons pas nous intéresser aux résultats de la simulation physique), cochez le puis cliquez avec le bouton droit de la souris et ensuite sur Apply :



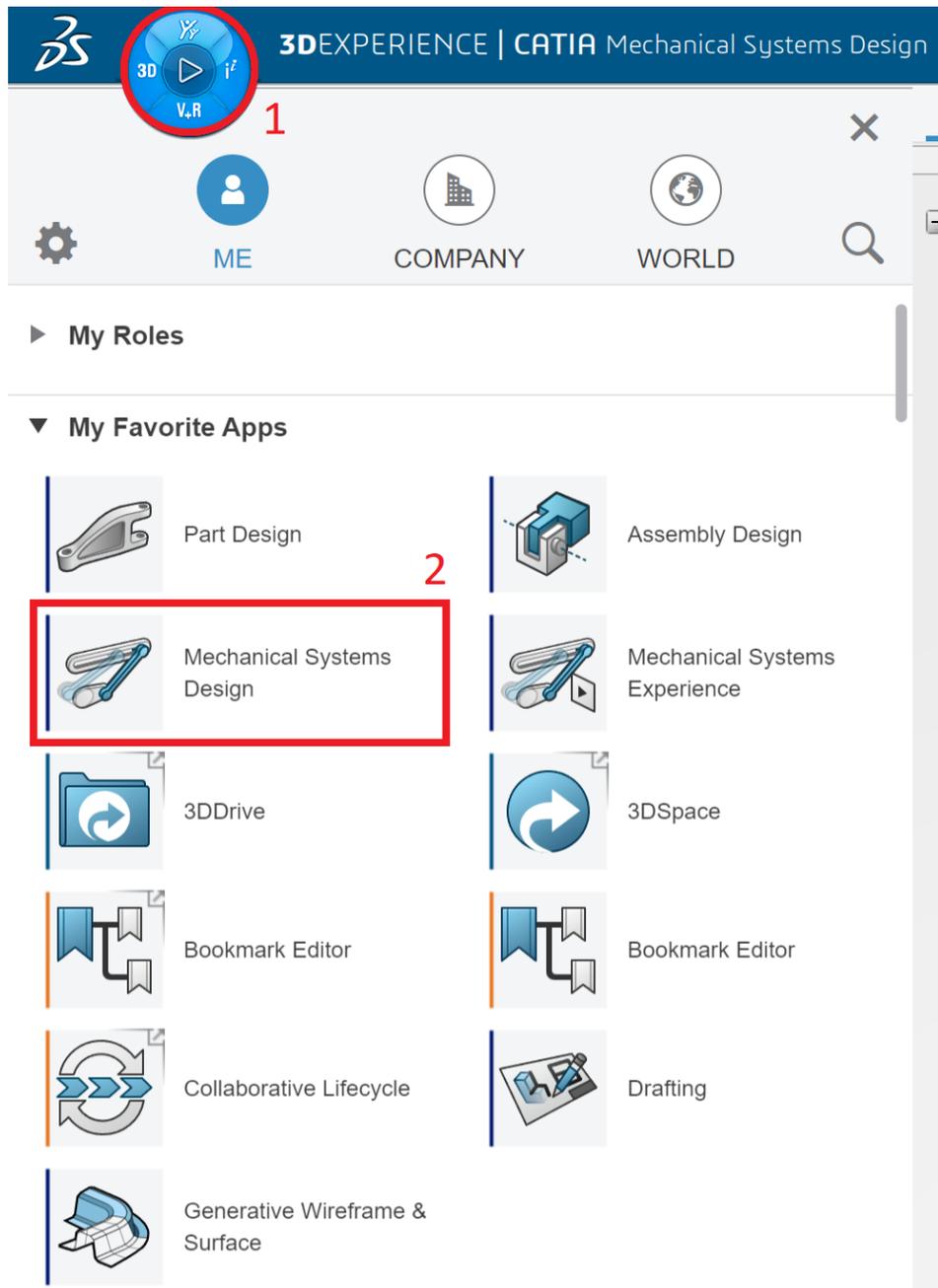
Puis cliquez sur le palonnier et le boîtier avec le bouton gauche de la souris et validez ensuite en cliquant sur le coche vert du menu contextuel :



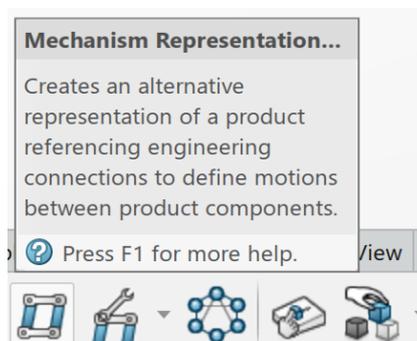
Pour vérifier que les matériaux sont bien appliqués, vous pouvez vérifier qu'ils apparaissent bien dans l'arbre du modèle (dans la catégorie de composants Materials) :



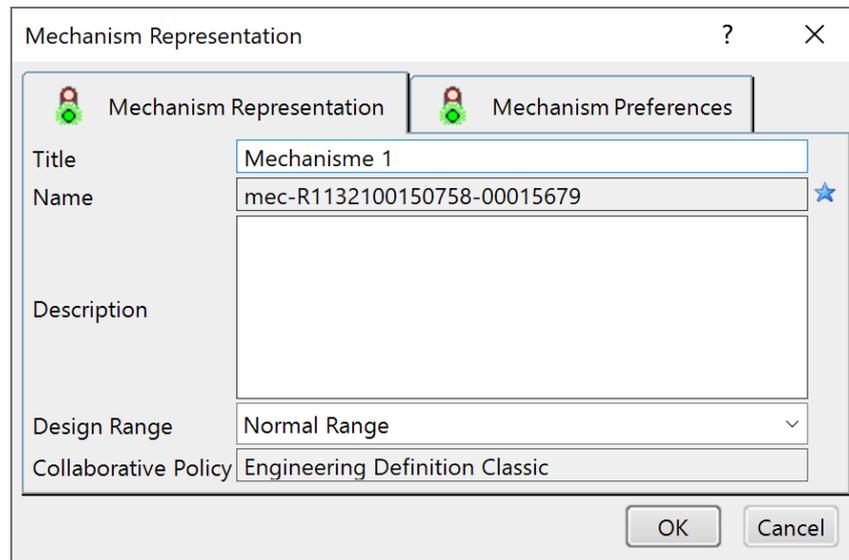
Cliquez sur l'app Mechanical System Design :



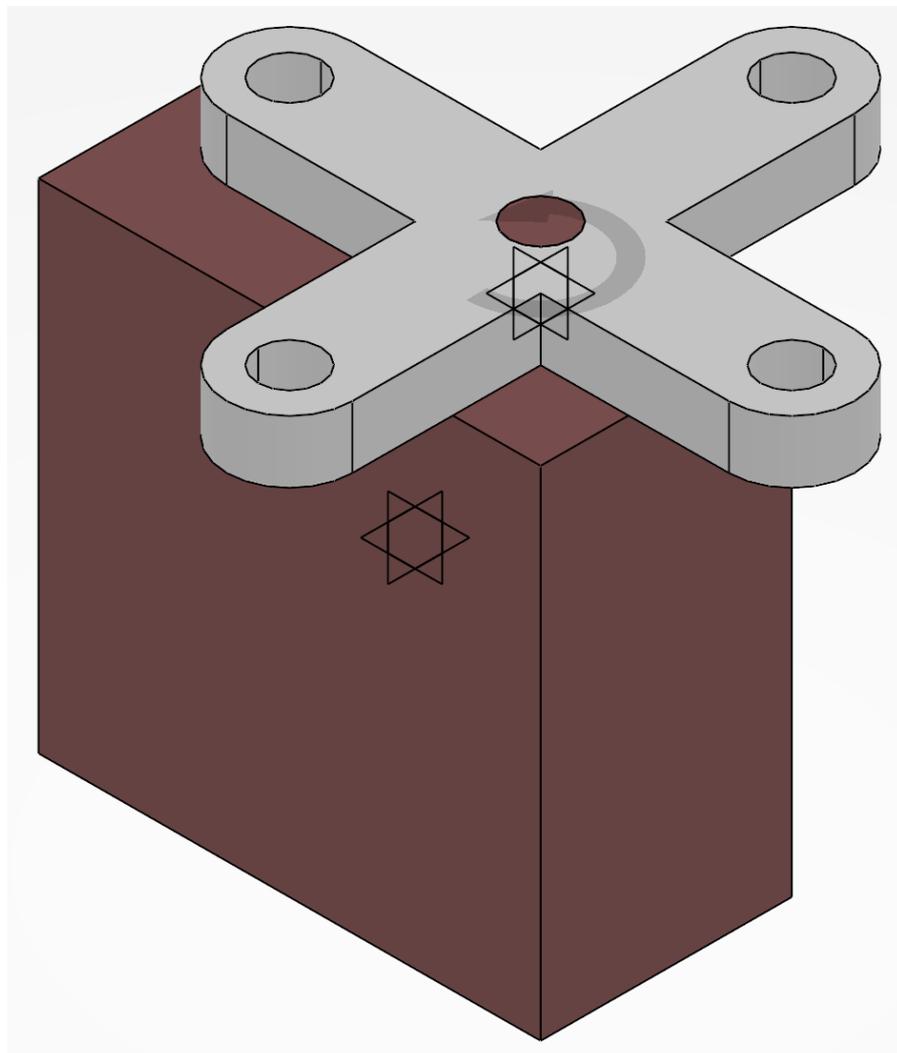
Réaliser un mécanisme dans Mechanical System Design en cliquant sur Mechanism Representation :



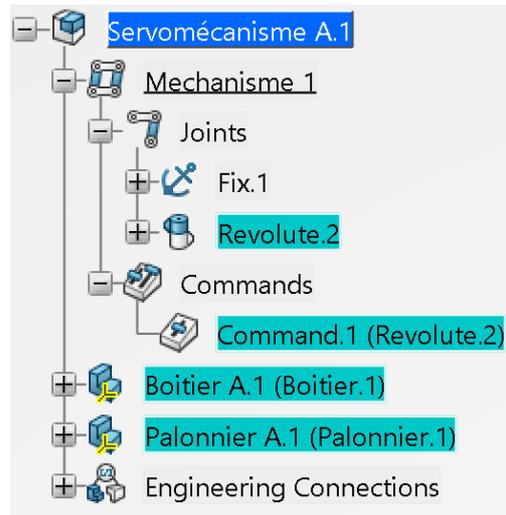
Créer un mécanisme et renommez le en « Mécanisme 1 » puis cliquez sur OK :



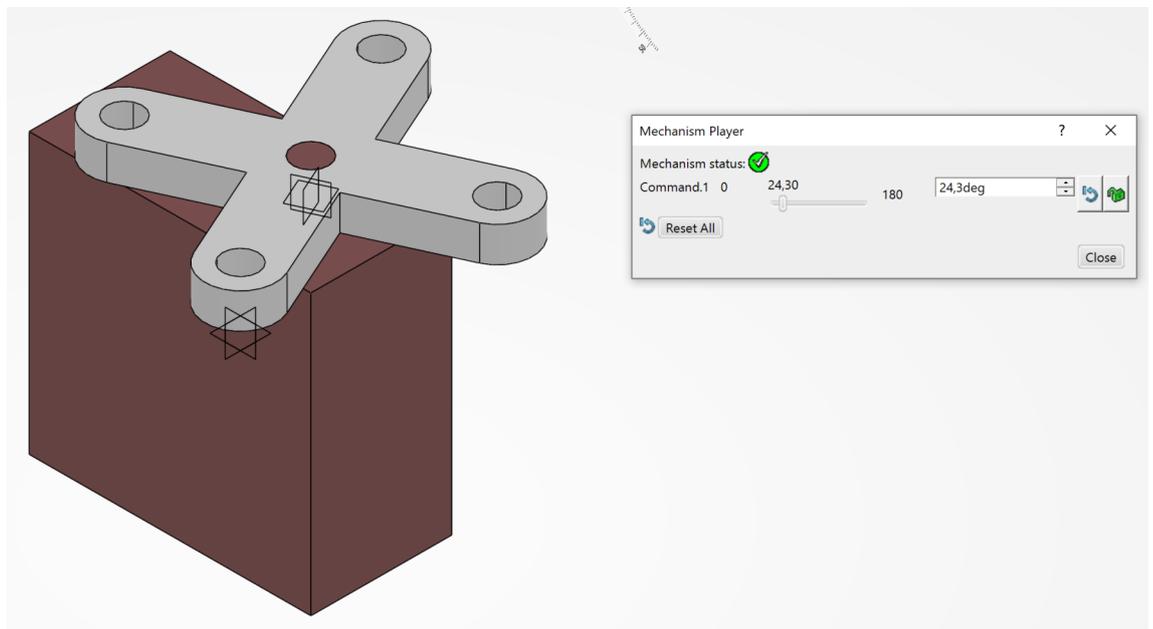
Une flèche symbolisant la mobilité du mécanisme apparaît alors :



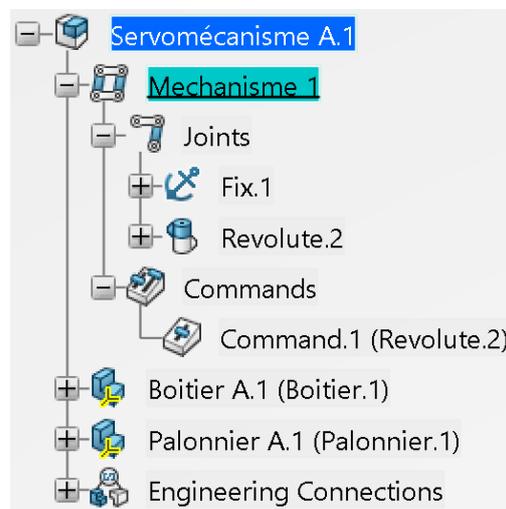
Pour vérifier le fonctionnement du mécanisme, faite un double clic sur la commande « Command 1 » dans l'arbre du modèle :



Puis déplacez le curseur de gauche à droite pour opérer des rotations dans un sens puis dans l'autre :

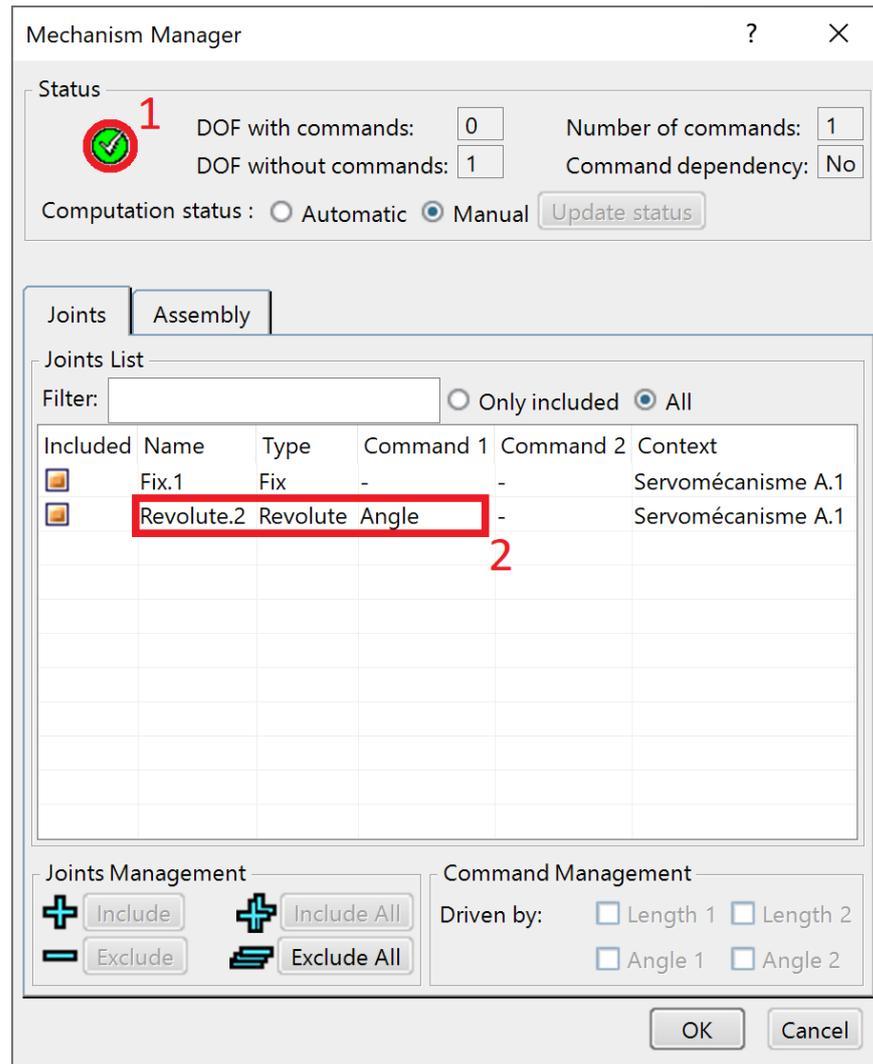


Si rien ne se passe faire un double clic sur Mécanisme 1 dans l'arbre du modèle :

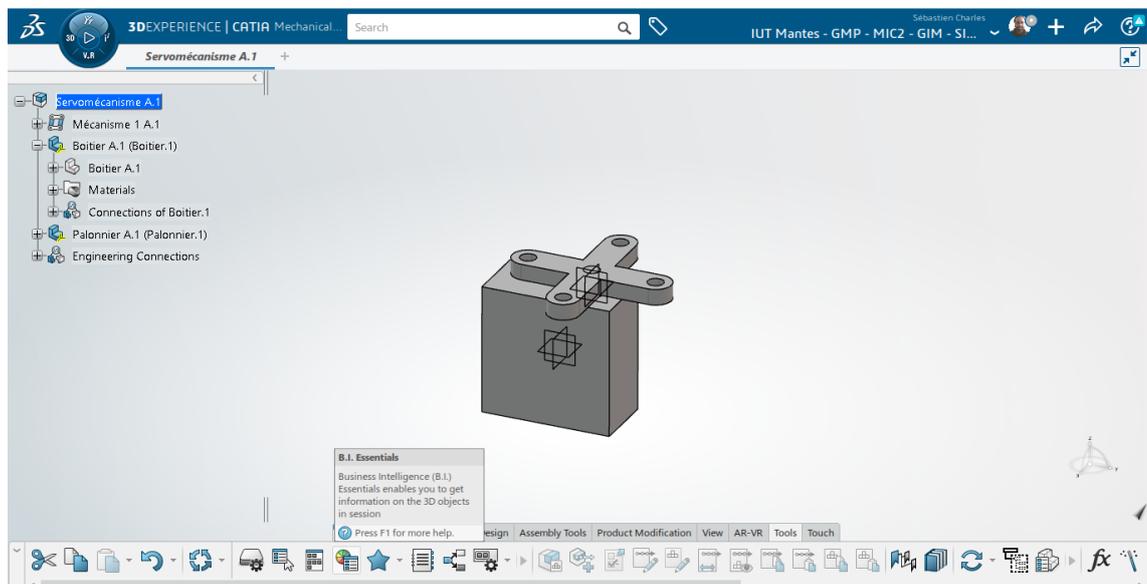


Puis vérifiez que le mécanisme est bien valide et que la liaison pivot possède bien une

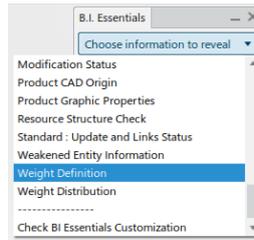
commande d'angle associée, si e n'est pas le cas vérifiez le étapes décrites précédemment :



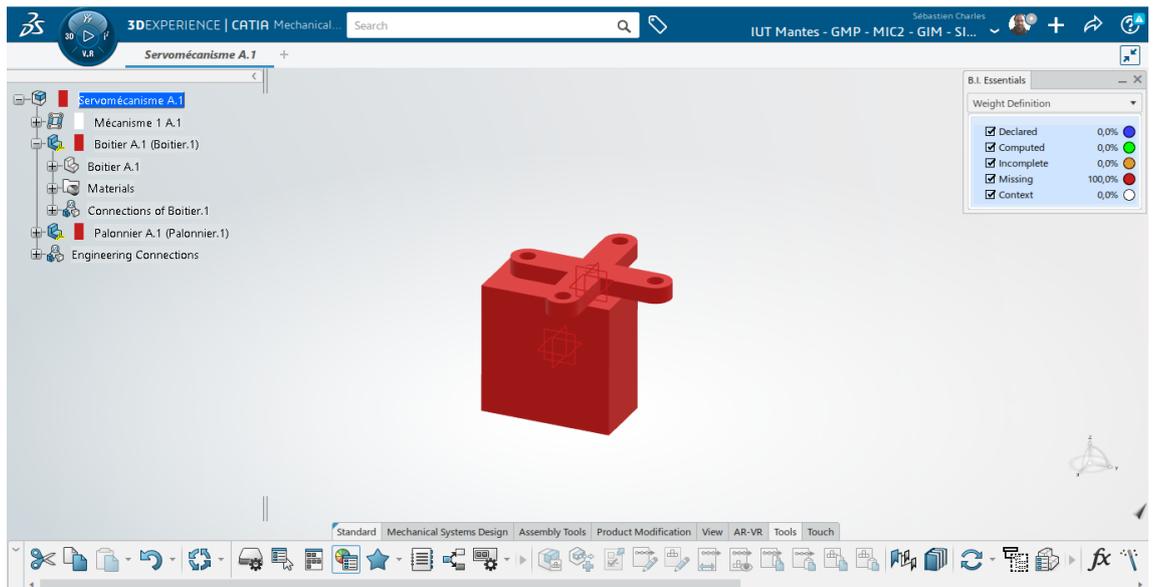
Pour que le modèle dynamique du système puisse être simulé, il est nécessaire que les masses des composants soient calculées par le logiciel. Pour ce faire, il faut aller dans l'application Mechanical Systems Design, puis dans l'onglet Tools et cliquer sur B.I. Essentials :



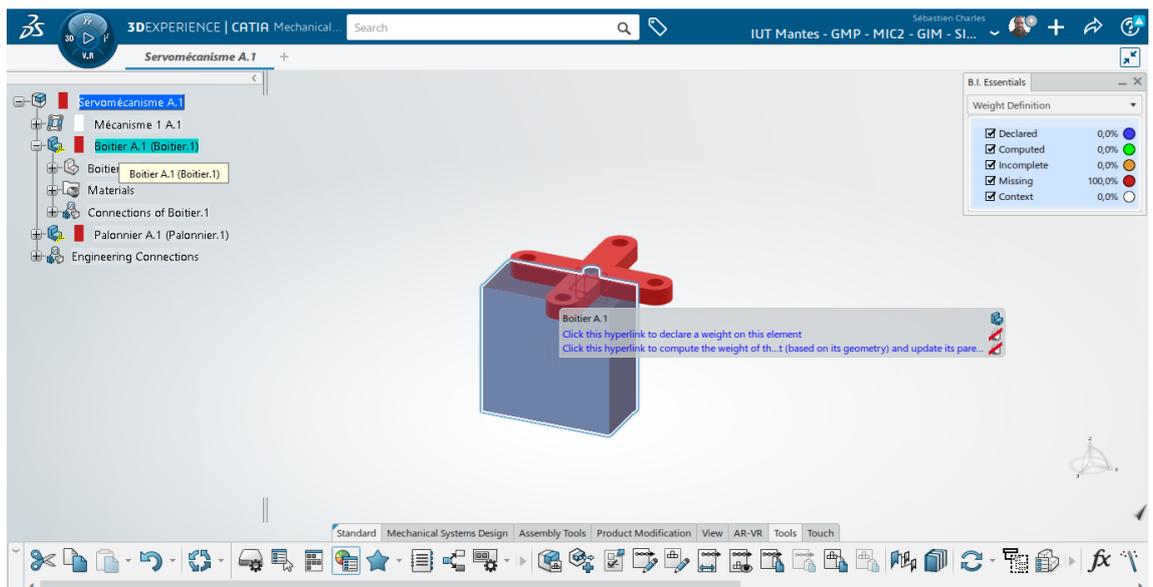
Puis trouver « Weight Definition » dans le menu déroulant :



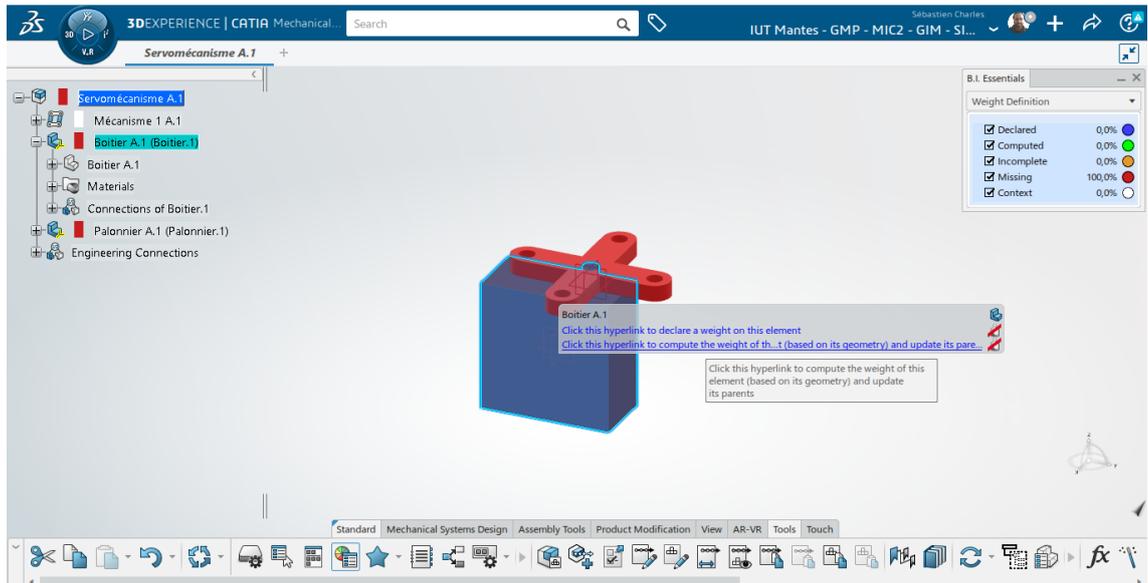
La fenêtre suivante apparaît alors. Il est important que toutes les masses soient calculées. La couleur rouge indique les pièces pour lesquelles la masse n'a pas été calculée. Les couleurs verte ou bleu (souhaitées) correspondent respectivement à calculées ou déclarées. La couleur orange indique qu'elles sont partiellement calculées :



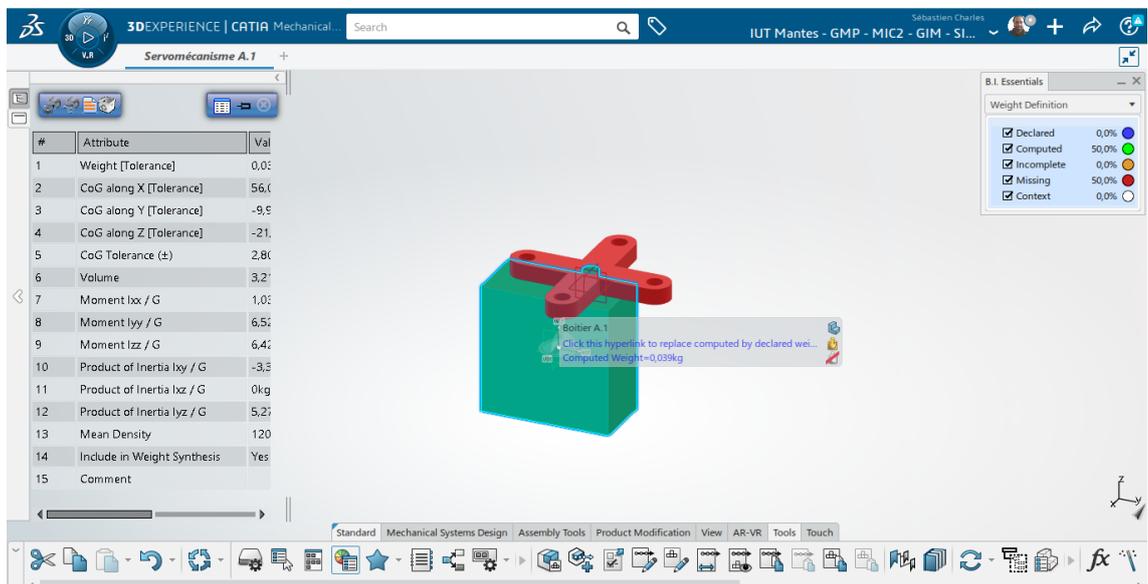
Pour forcer le calcul du boîtier du servomécanisme, cliquez sur l'objet dans le modèle 3D ou dans l'arbre du modèle :



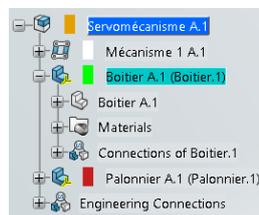
Puis cliquez sur le lien hypertexte qui apparaît sur la pièce : « Click this hyperlink to compute the weight... » :



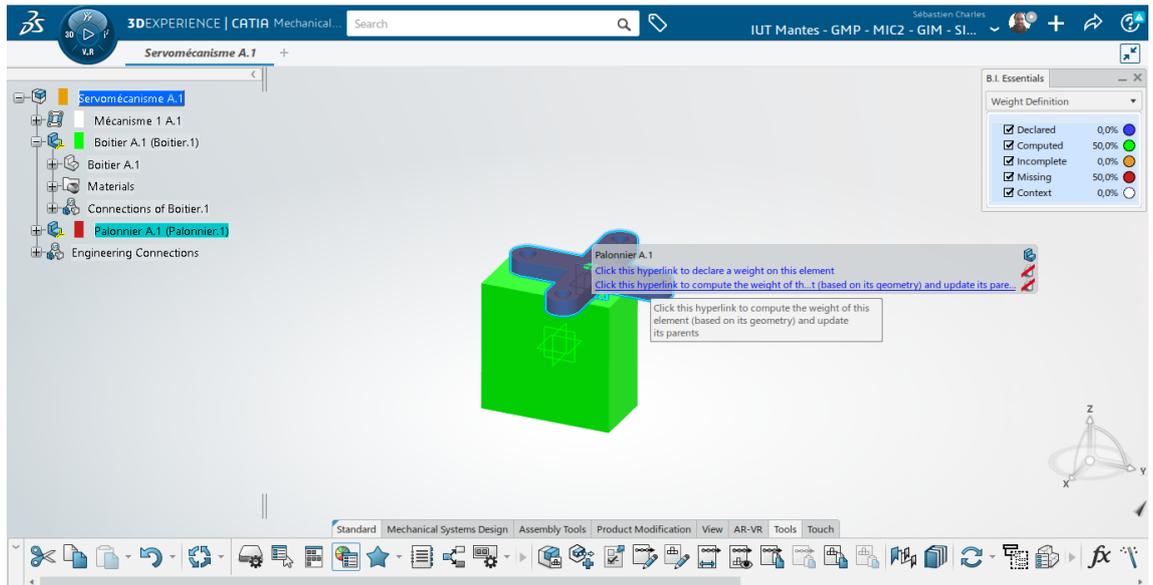
La pièce devient alors verte et son poids est indiqué dans le lien hypertexte :



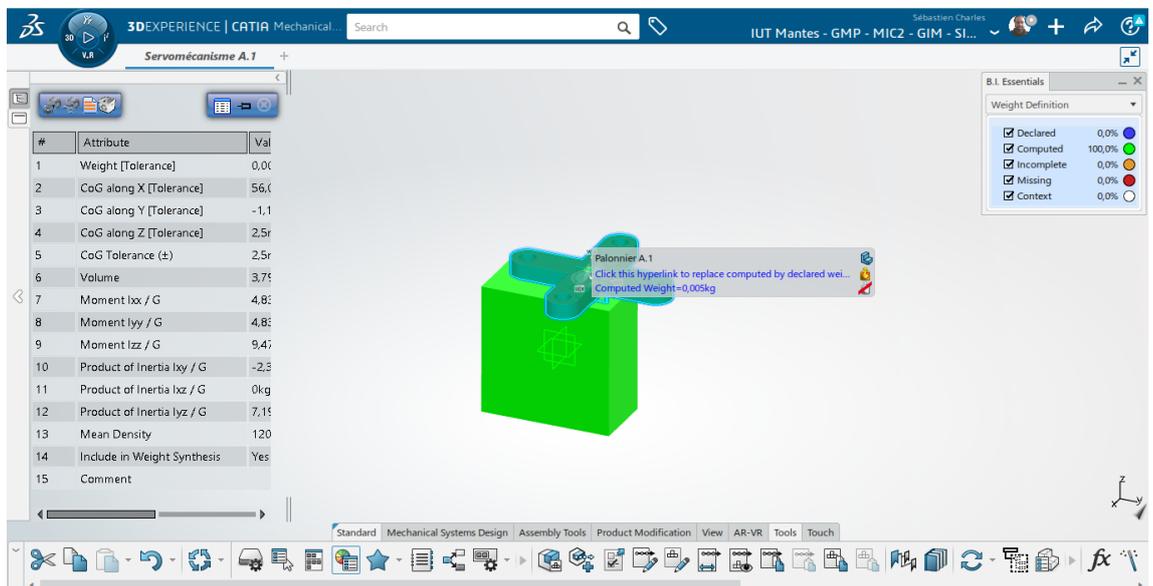
Cliquez sur la croix blanche entourée d'un rond blanc dans la fenêtre de gauche et constatez que le rectangle qui précède le boîtier dans l'arbre du modèle est désormais vert :



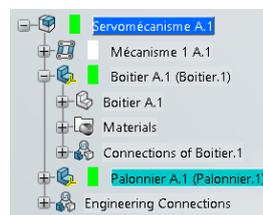
La fenêtre de droite indique le poids total est calculé à 50% car il faut en effet calculer celui du palonnier. Cliquez sur le modèle 3D du palonnier ou dans l'arbre du modèle :



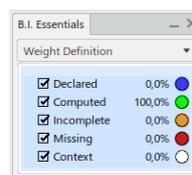
Cliquez ensuite sur le lien hypertexte qui permet de réaliser le calcul de poids :



Fermez la fenêtre de gauche et constatez que tous les éléments de l'arbre du modèle apparaissent en vert. Si l'assemblage n'apparaît pas en vert, cliquez dessus et forcez le calcul de poids de l'ensemble :



Vous pouvez constater que dans la fenêtre de synthèse de droite, le modèle est bien calculé à 100% :



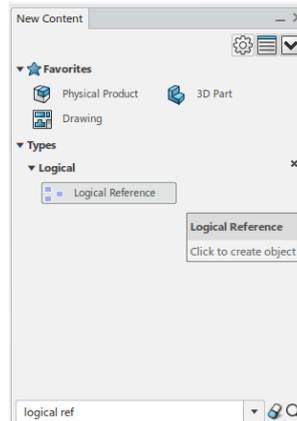
## Création du modèle logique de référence et contrôle cinématique en position

La prochaine étape consiste à créer un modèle logique du servomécanisme pour le lier à la carte Arduino par le biais des FMI.

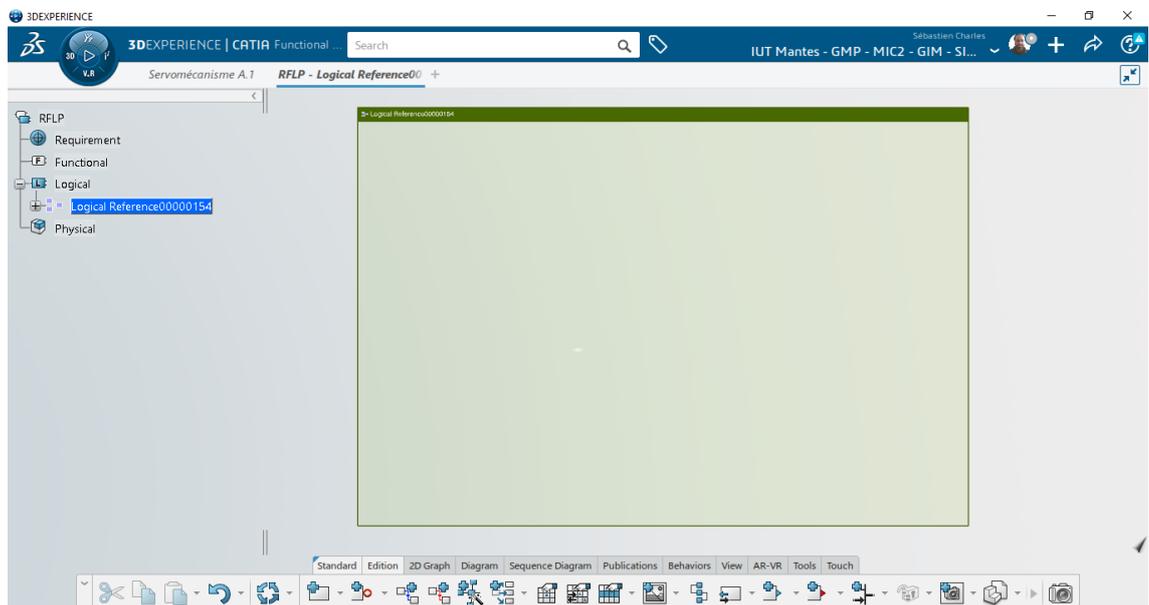
Pour commencer, appuyez sur l'icone « + » puis sur « New Content » :



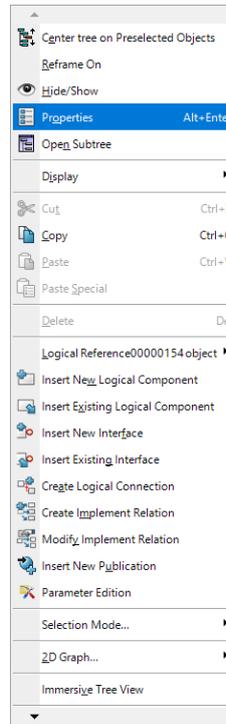
Puis recherchez « Logical Reference » dans les types en utilisant le moteur de recherche situé en bas de la fenêtrre et cliquez dessus :



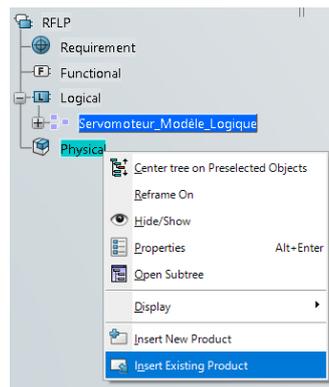
Cliquez sur la fenêtre verte qui correspond au modèle logique créé puis cliquez sur le bouton droit de la souris :



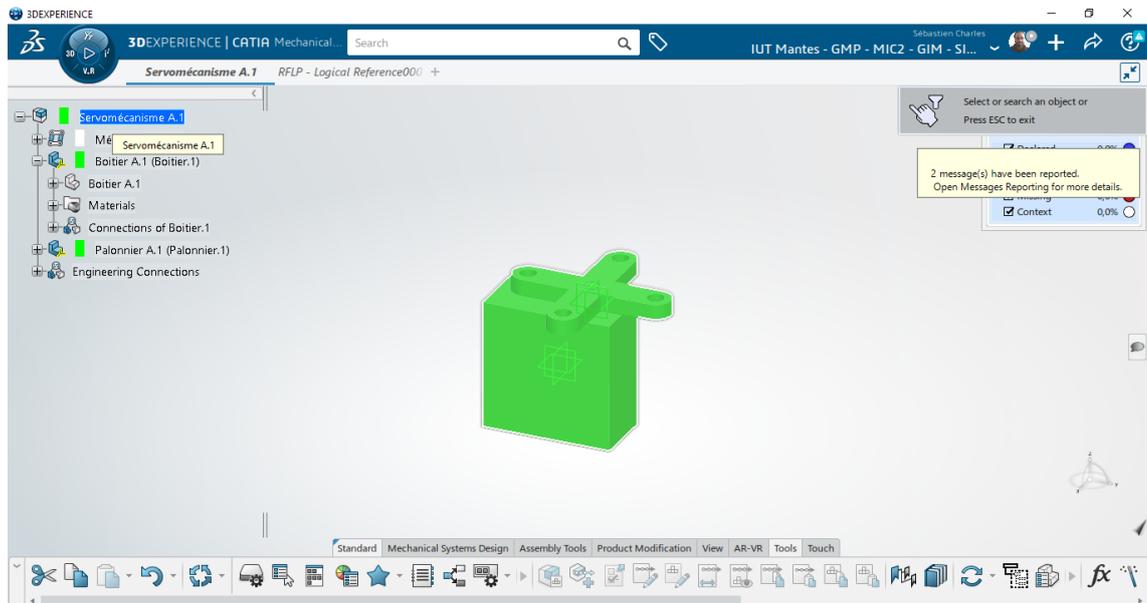
Et choisissez ensuite « Properties » :



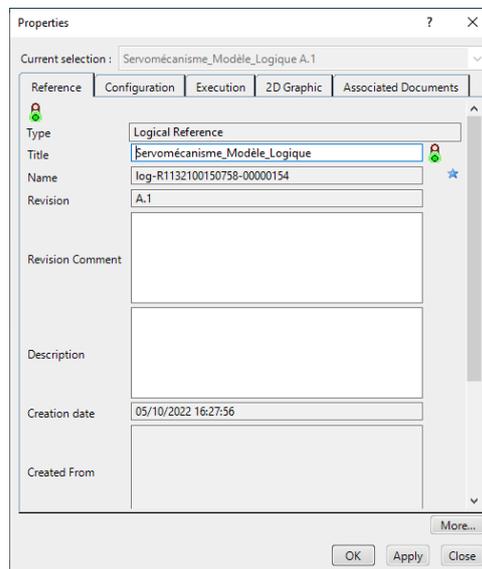
Dans l'arbre du modèle, sélectionnez « Physical » puis cliquez sur le bouton droit de la souris puis sur « Insert Existing Product » :



Puis dans l'onglet du servomécanisme, cliquez sur le noeud le plus haut de l'arbre du modèle pour intégrer l'ensemble du système au modèle logique :



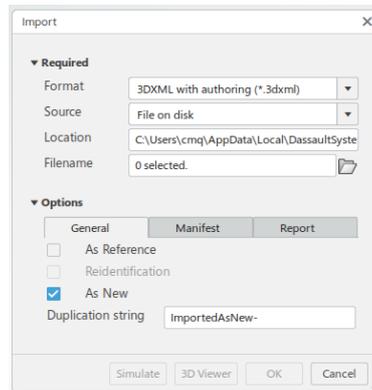
Cliquez avec le bouton droit de la souris sur le modèle logique puis sur propriétés et renommez le :



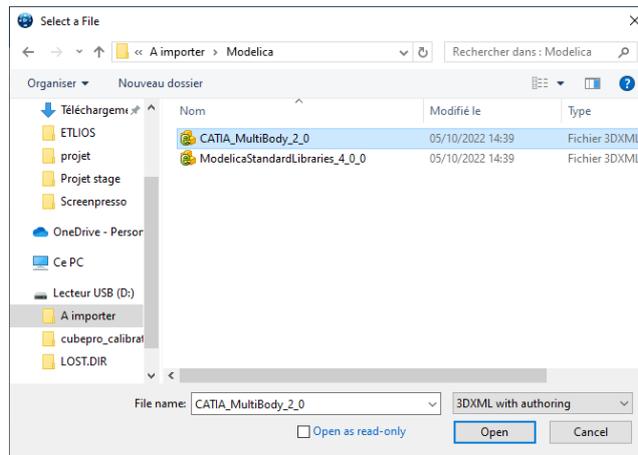
Ensuite, cliquez sur « + » en haut à droite de la fenêtre principale puis sur « Import » :



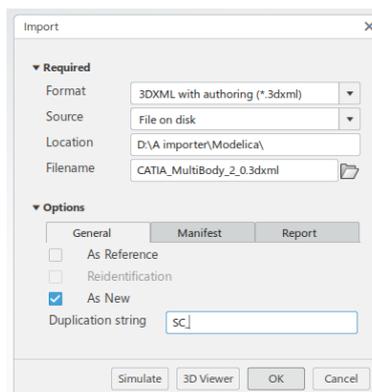
La fenêtre suivante s'ouvre :



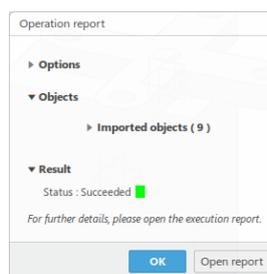
Cliquez sur l'icône de dossier et aller chercher les fichiers XML « CATIA\_MultiBody\_2\_0 », cochez la case AS New puis précisez éventuellement un Duplication String et cliquez enfin sur OK :



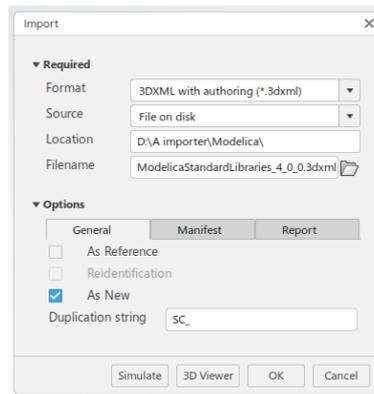
NB : Ces fichiers XML peuvent être téléchargés sur Ressources numériques à télécharger <sup>[http://edu.3ds.com/en/hub/arduino-robot]</sup> <sup>[https://edu.3ds.com/en/hub/arduino-robot]</sup> <sup>[https://edu.3ds.com/en/hub/arduino-robot]</sup>



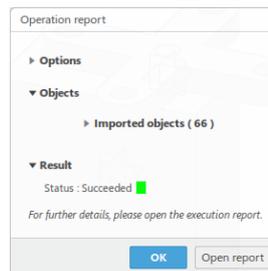
La fenêtre suivante doit apparaître :



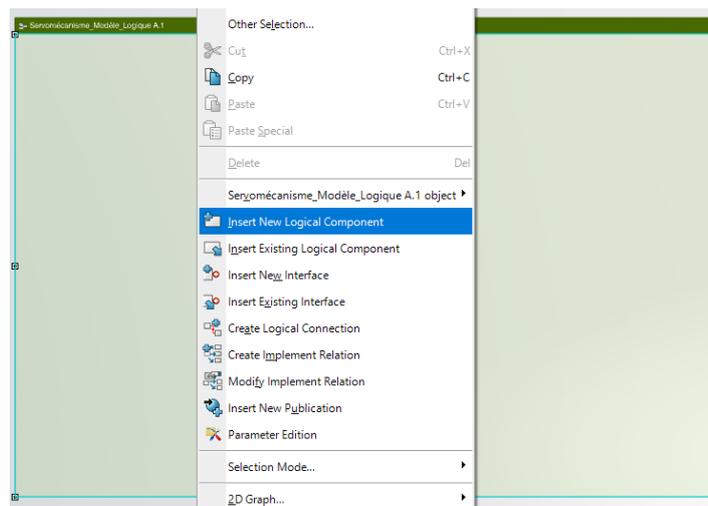
Recommencez l'opération avec « ModelicaStandardLibraries\_4\_0\_0 ».



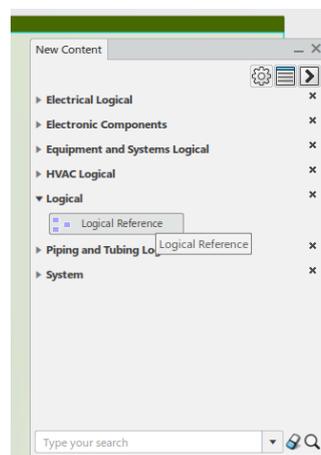
La fenêtre suivante doit apparaître :



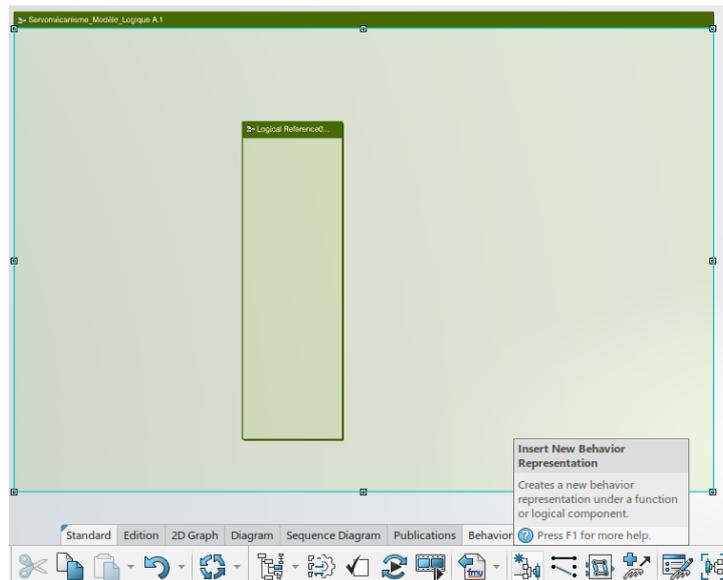
Sélectionnez ensuite le modèle logique puis réalisez un clic droit, puis Insert « New Logical Component » :



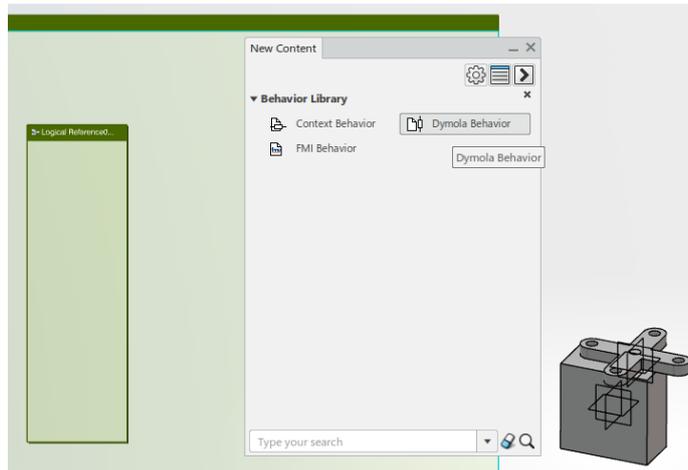
Puis, ajoutez un nouveau « Logical Reference » :



Sélectionnez le nouveau modèle logique puis cliquez sur « Insert New Behavior Representation » dans la barre d'icône du bas, dans l'onglet Behavior :



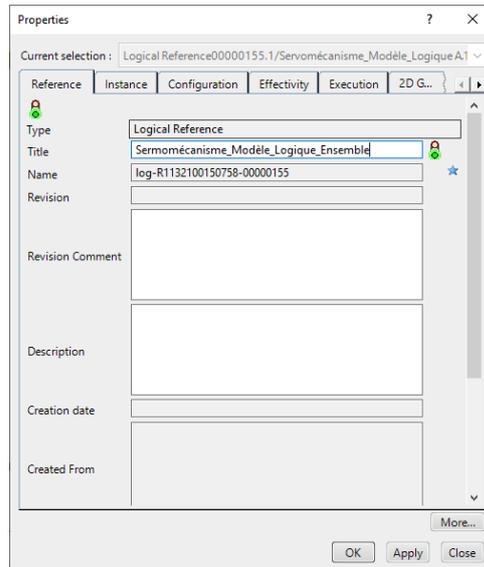
Puis choisissez « Dymola Behavior » :



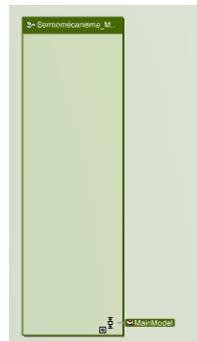
Un icône apparait alors en bas à droite de la fenêtre :



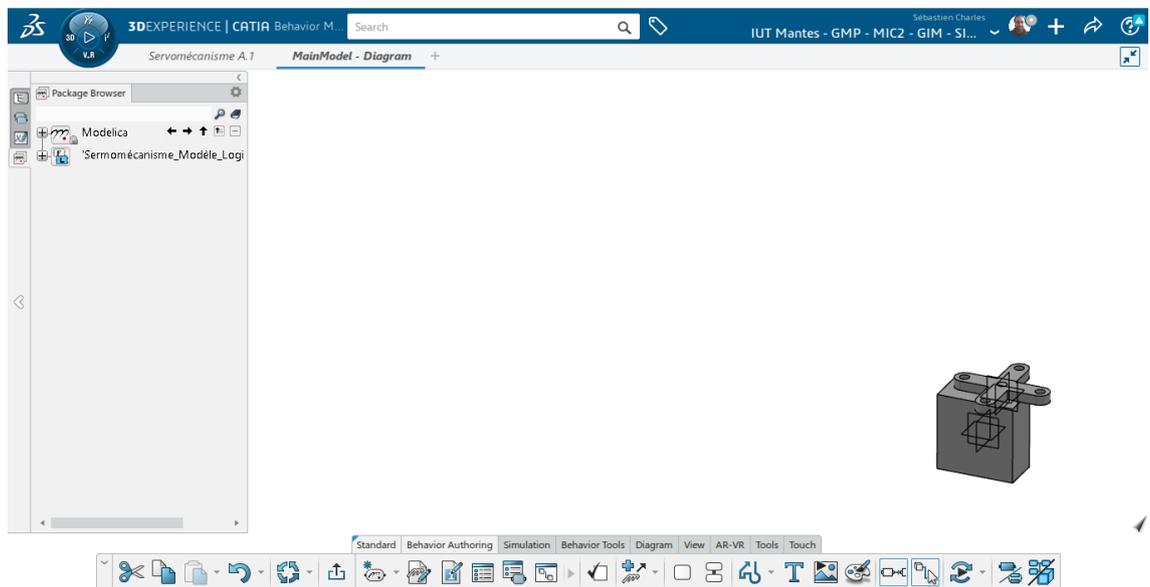
Faites un clic droit sur le modèle logique, puis cliquez sur Properties pour le renommer :



Cliquez ensuite sur l'icône situé en bas à droite de la fenêtre, puis double cliquer sur « Main Model » :



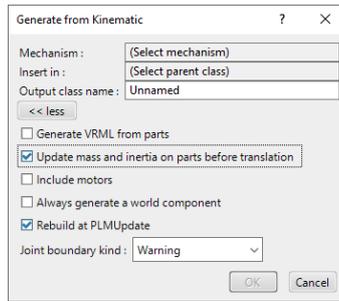
La fenêtre suivante apparaît :



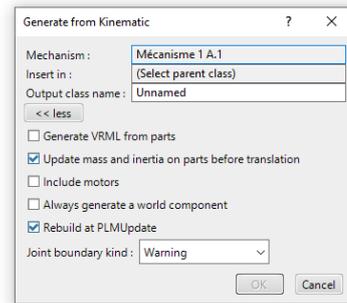
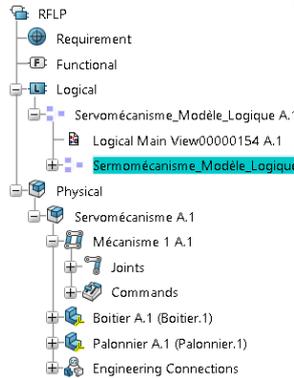
Dans Behavior Tools, cliquez sur « Generate from Kinematic » :



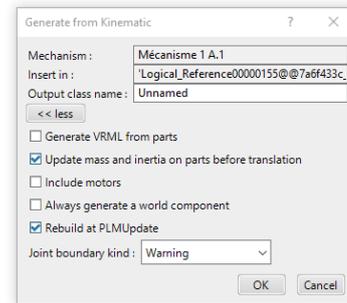
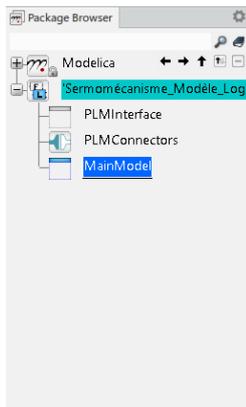
Cochez « Update mass an inertia on part before translation » :



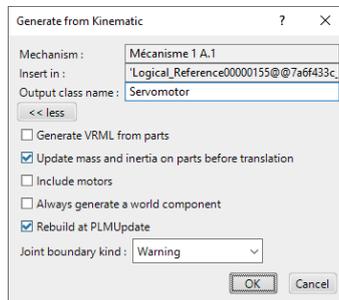
Cliquez la case « Mechanism » puis cliquez sur le Mécanisme dans l'arbre du modèle :



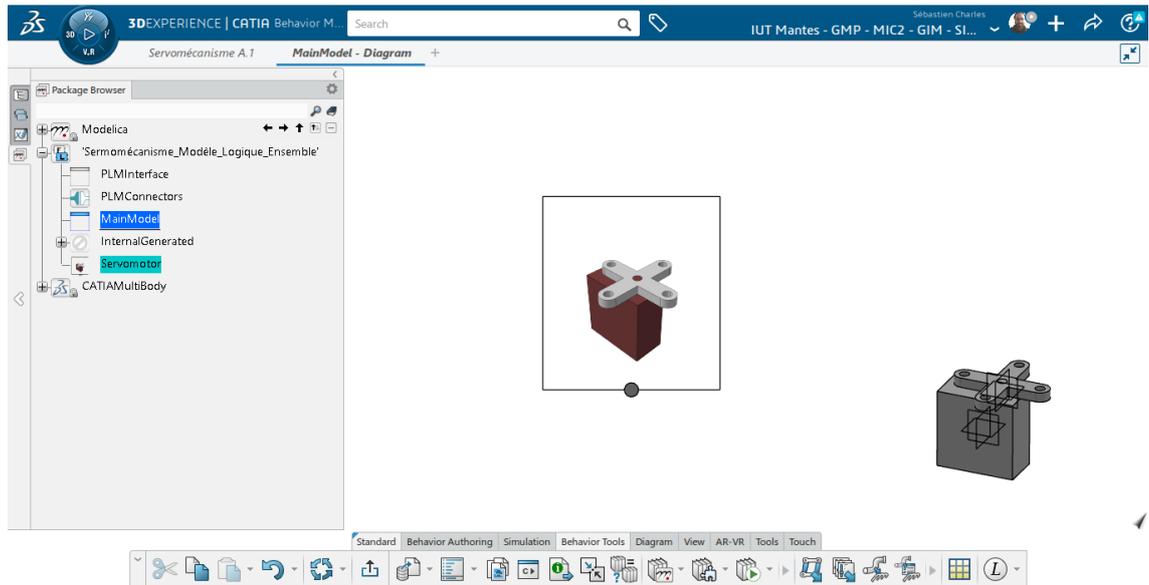
Cliquez ensuite sur la cas « Insert In » puis cliquez sur le modèle logique dans l'arbre du modèle :



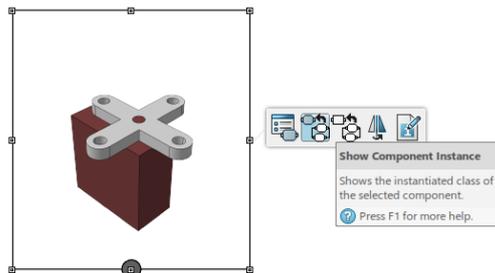
Puis renommez l' « Output Class Name » et cliquez sur OK :



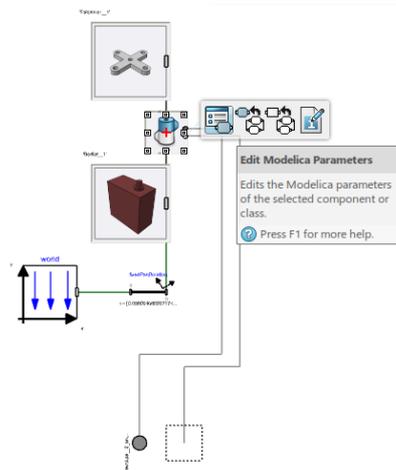
La fenêtre suivante apparaît :



Cliquez dessus puis choisissez « Show Component Instance » dans le menu contextuel :

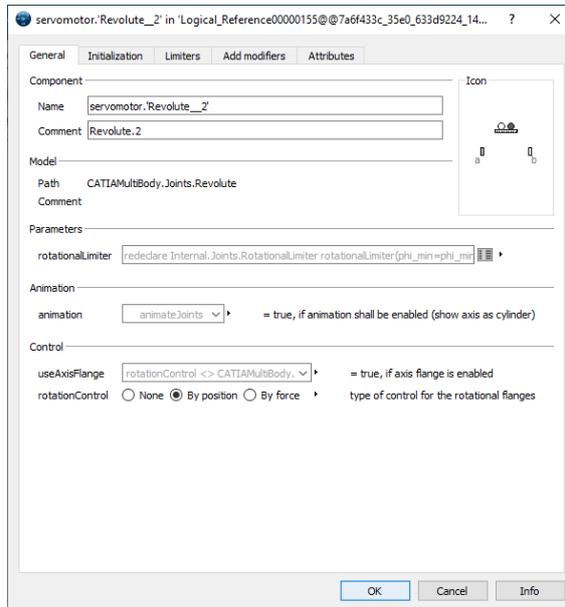


Puis cliquez sur le symbole de liaison pivot et ensuite sur « Edit Modelica Parameters » :

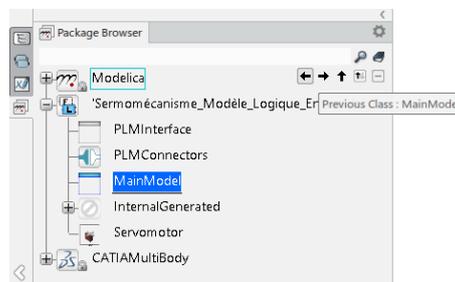


NB : vous constaterez que dans ce modèle, nous voyons apparaître les deux pièces, la liaison pivot et le référentiel de gravité de l'ensemble.

Puis cochez « RotationControl » « By Position » puis cliquez sur OK :



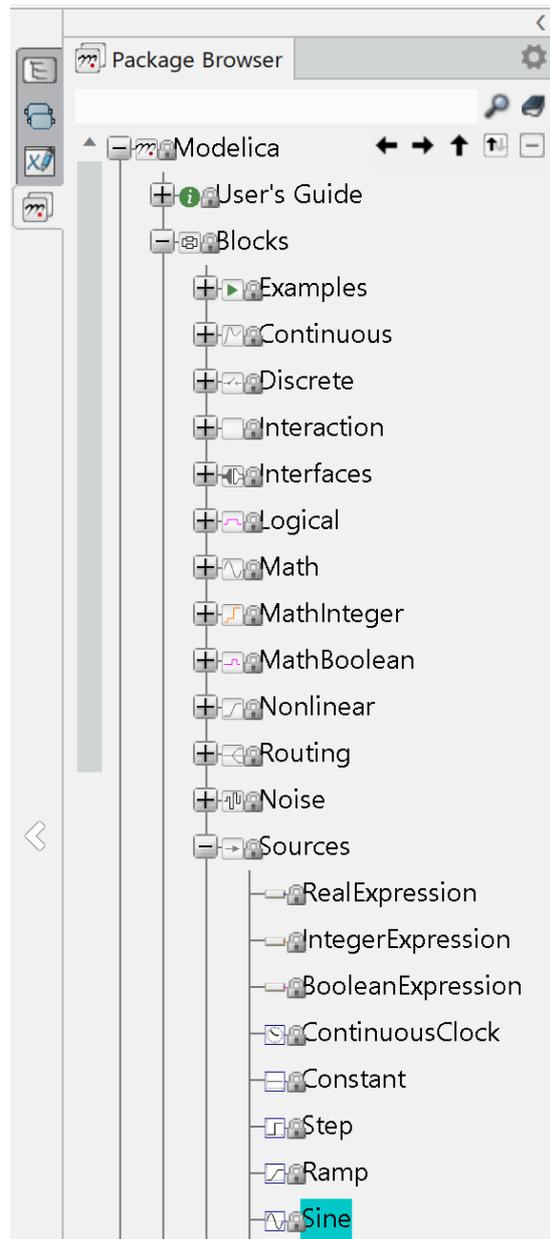
Puis cliquez sur la flèche noire pointée vers la gauche pour revenir au modèle précédent :



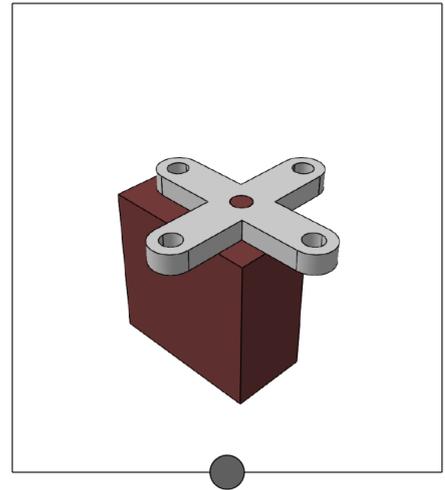
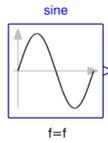
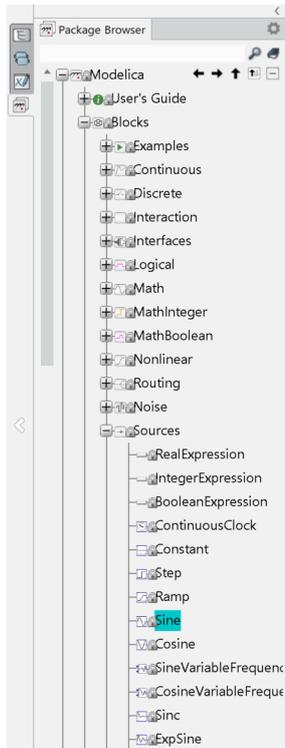
Ce niveau de modèle apparaît alors :



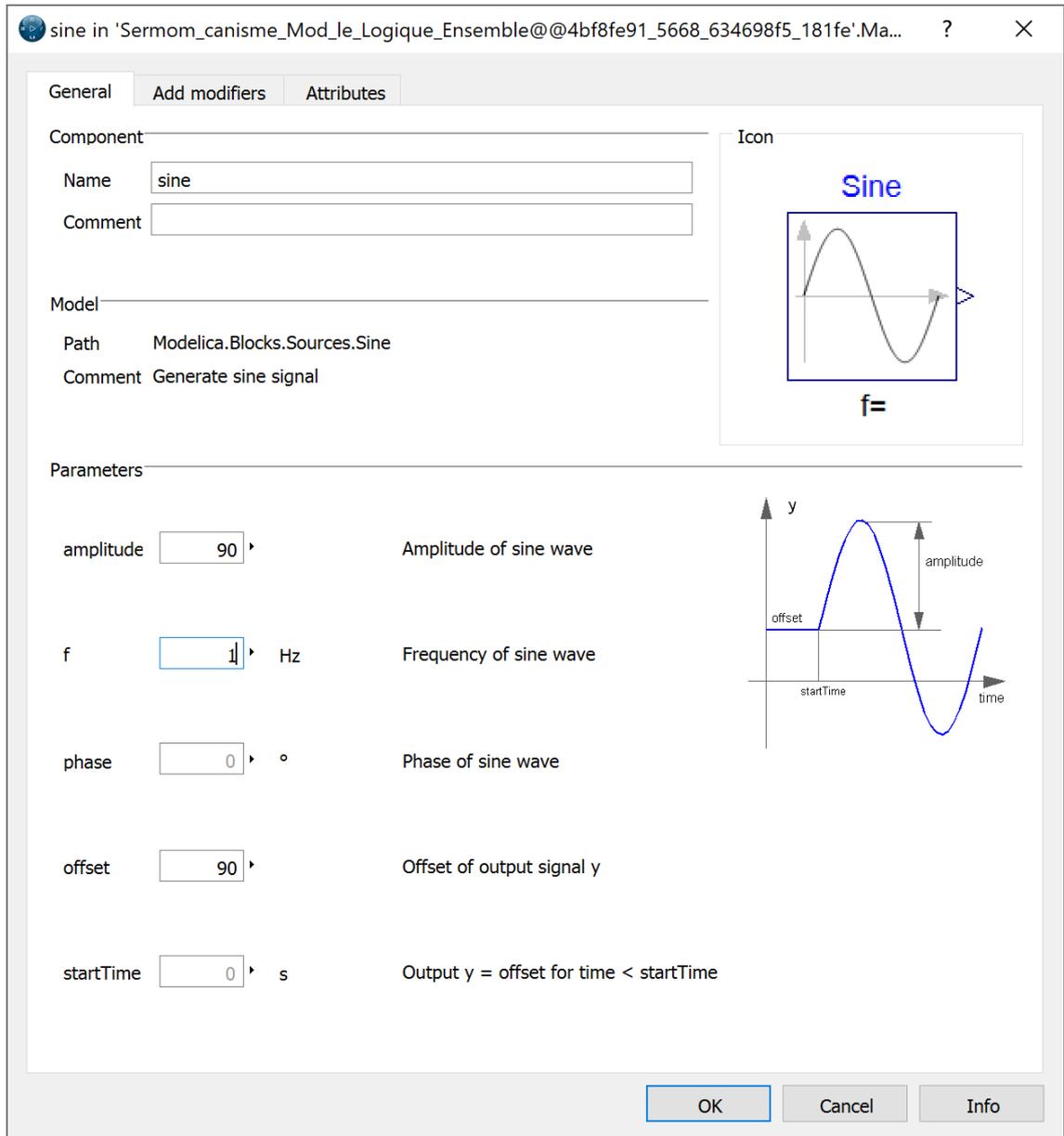
Ensuite, dans l'arbre de gauche cherchez le bloc « Sine » situé dans la catégorie « Modelica /Blocks/Sources/ ». Cet objet produit un signal de commande de type sinusoïdal



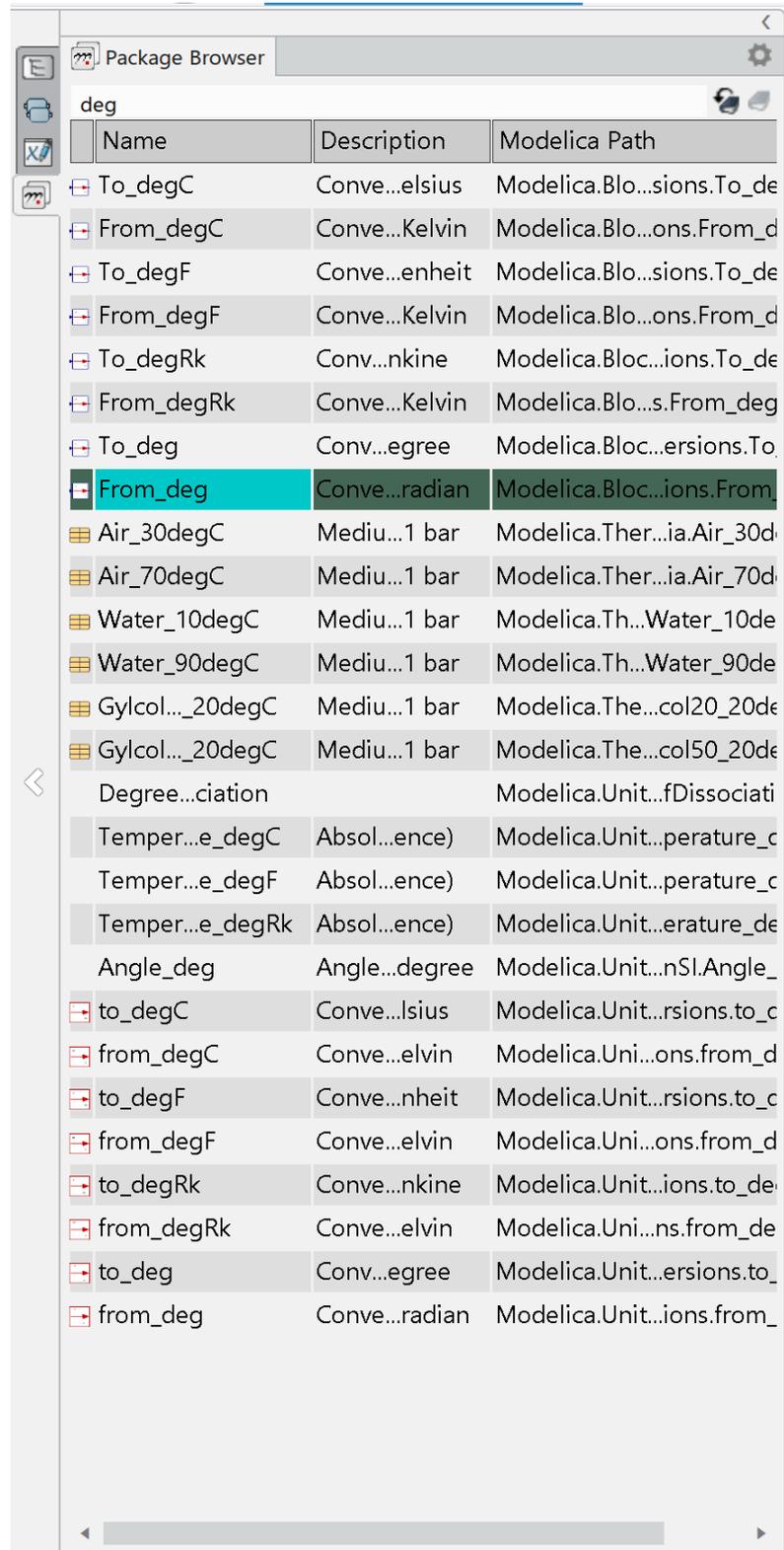
Puis faites un glisser/déplacer pour positionner l'objet sur l'écran :



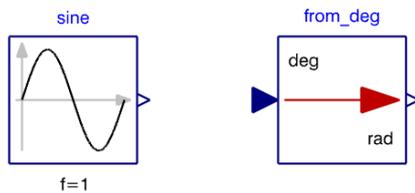
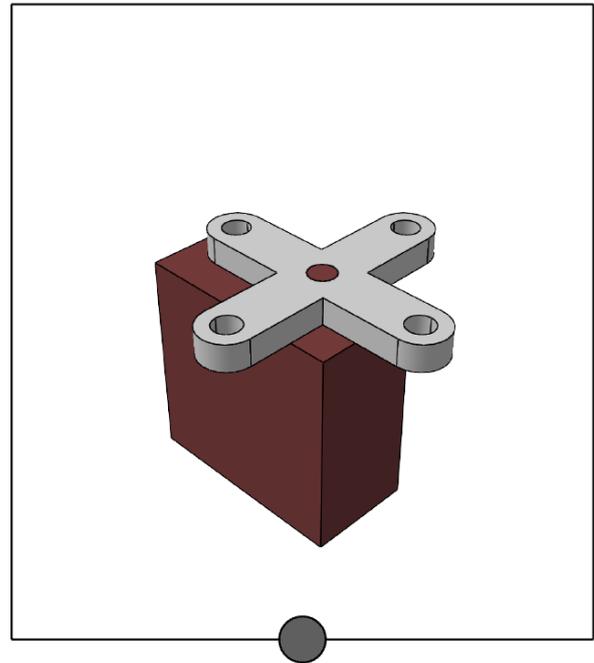
Puis double cliquez sur le bloc Sine pour changer son « amplitude » à 90, sa fréquence « f » à 1Hz, son « offset » à 90 puis cliquez sur OK.



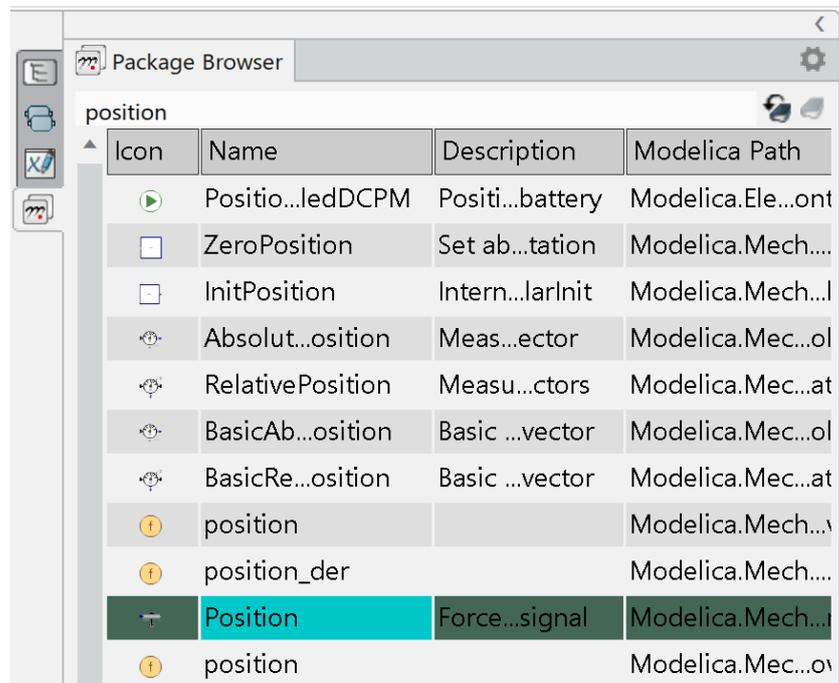
Cherchez ensuite le convertisseur de format « From\_deg » qui convertit les angles exprimés en degrés en radians :



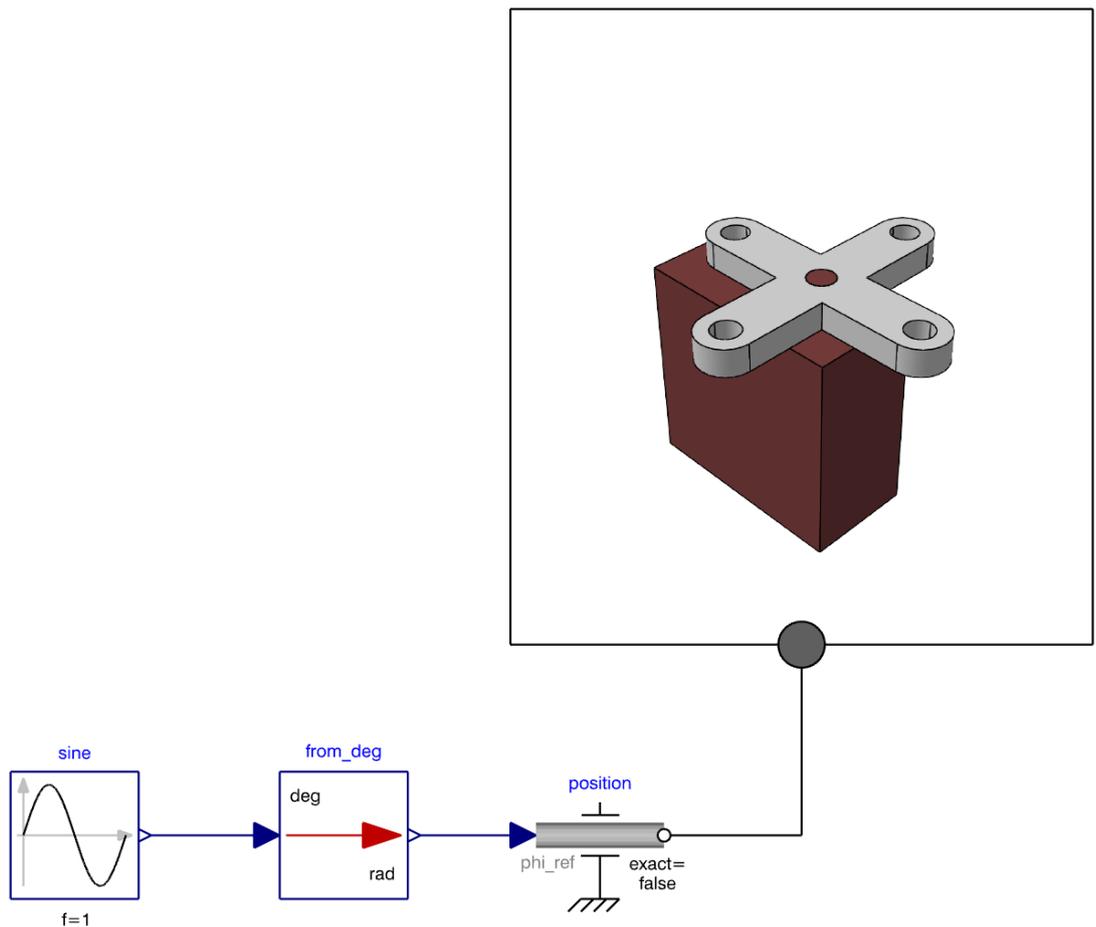
Faites un glisser/déplacer de ce bloc pour le placer à droite du signal sinusoïdal :



Cherchez ensuite le bloc « Position » :

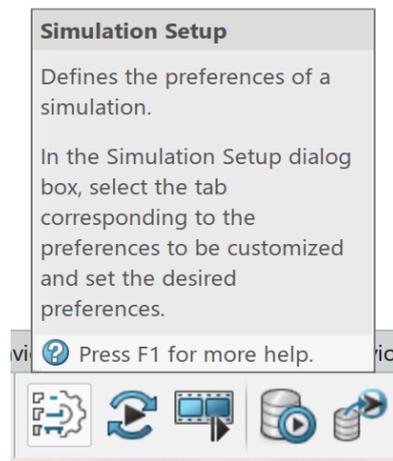


Puis réalisez un glisser/déplacer pour le positionner à droite du précédent bloc et reliez tous les blocs entre eux :

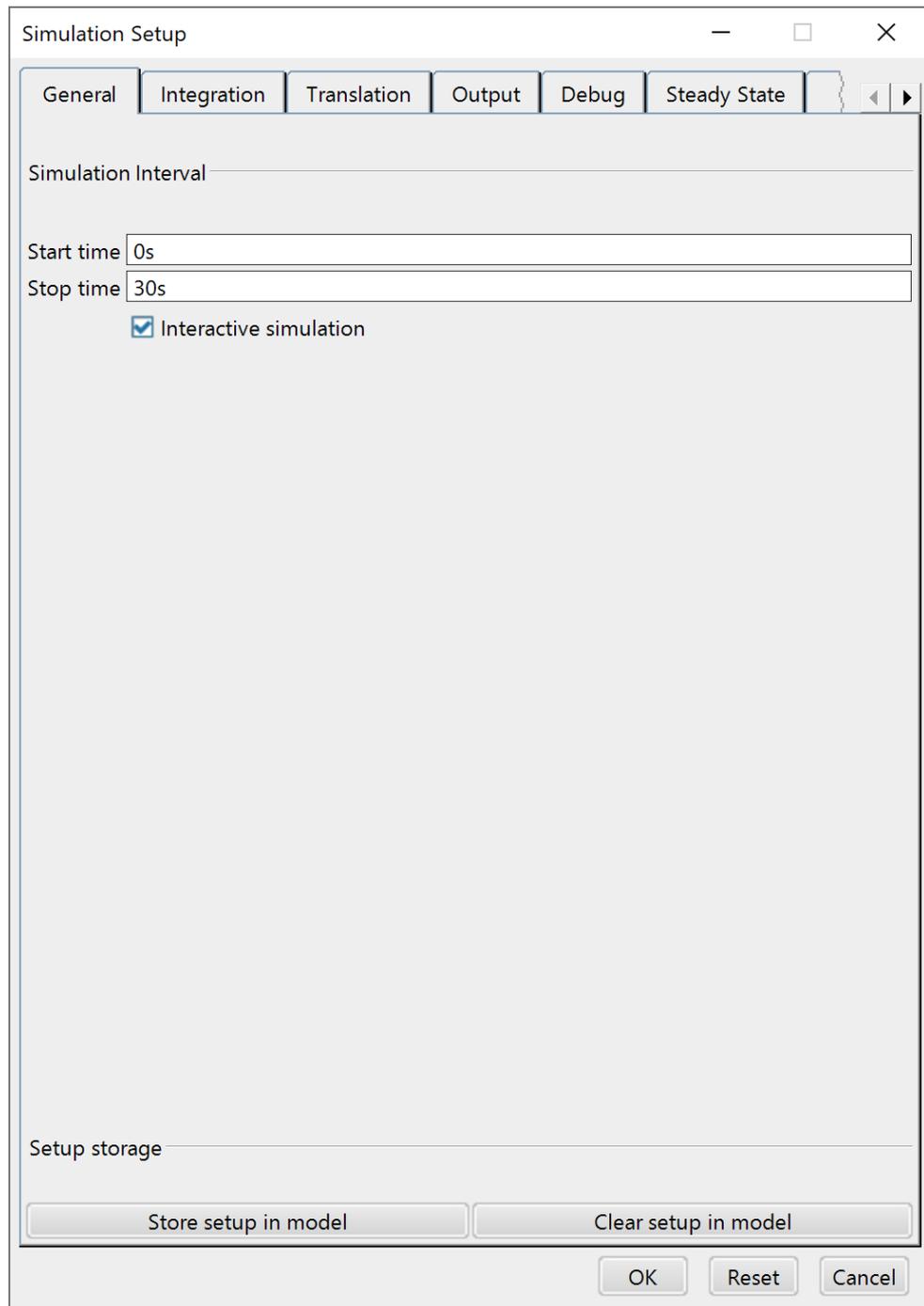


NB : Il était nécessaire de convertir les données ainsi pour que leur format soit compatible avec le connecteur d'entrée du bloc du servomécanisme.

Cliquez sur « Simulation Setup » :



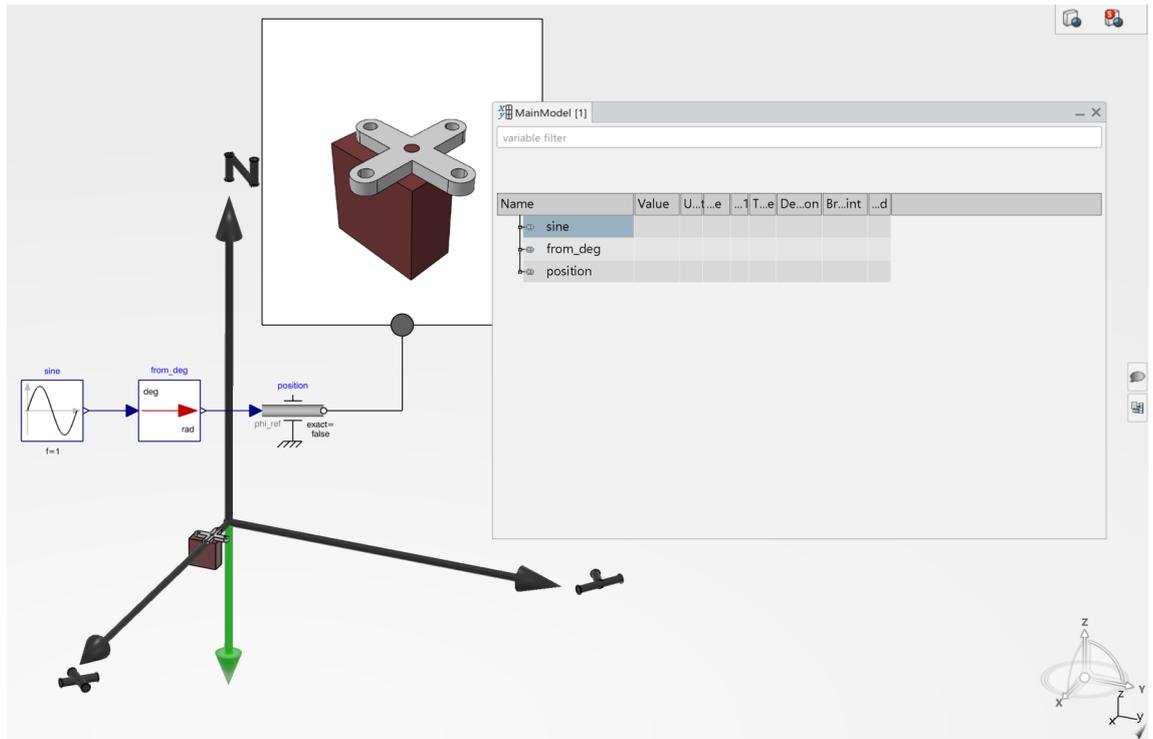
Reglez les paramètres : « Start time » à 0s, « Stop time » à 30s et cochez « Interactive simulation ». Notez que le temps de départ du calcul n'est pas forcément 0s, on peut en effet choisir de commencer une simulation à tout moment d'un cycle cinématique :



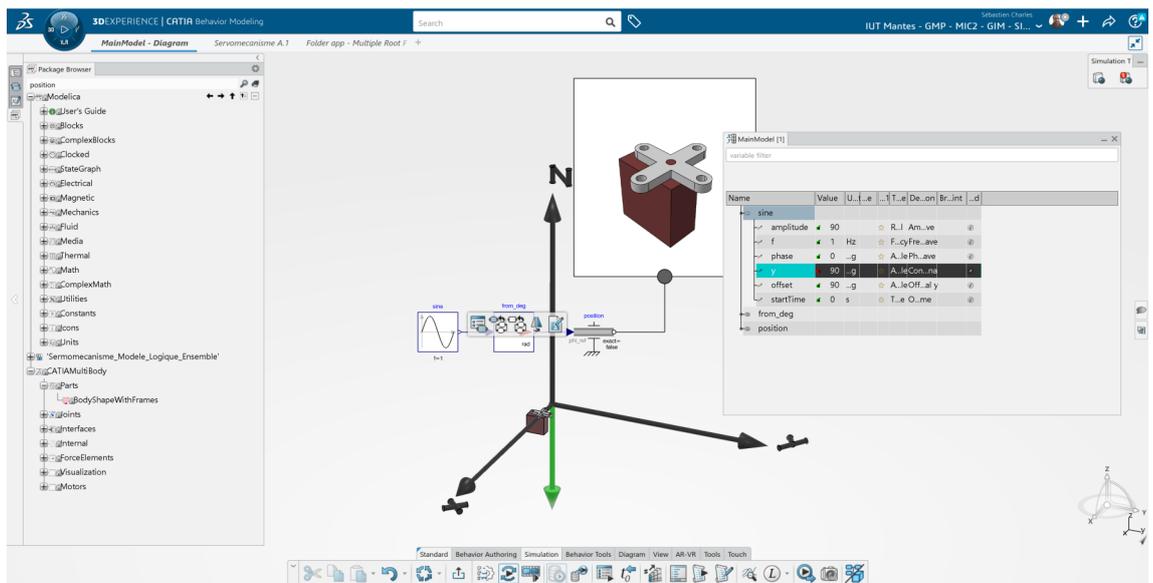
Lancez la simulation en cliquant sur « Simulate » :



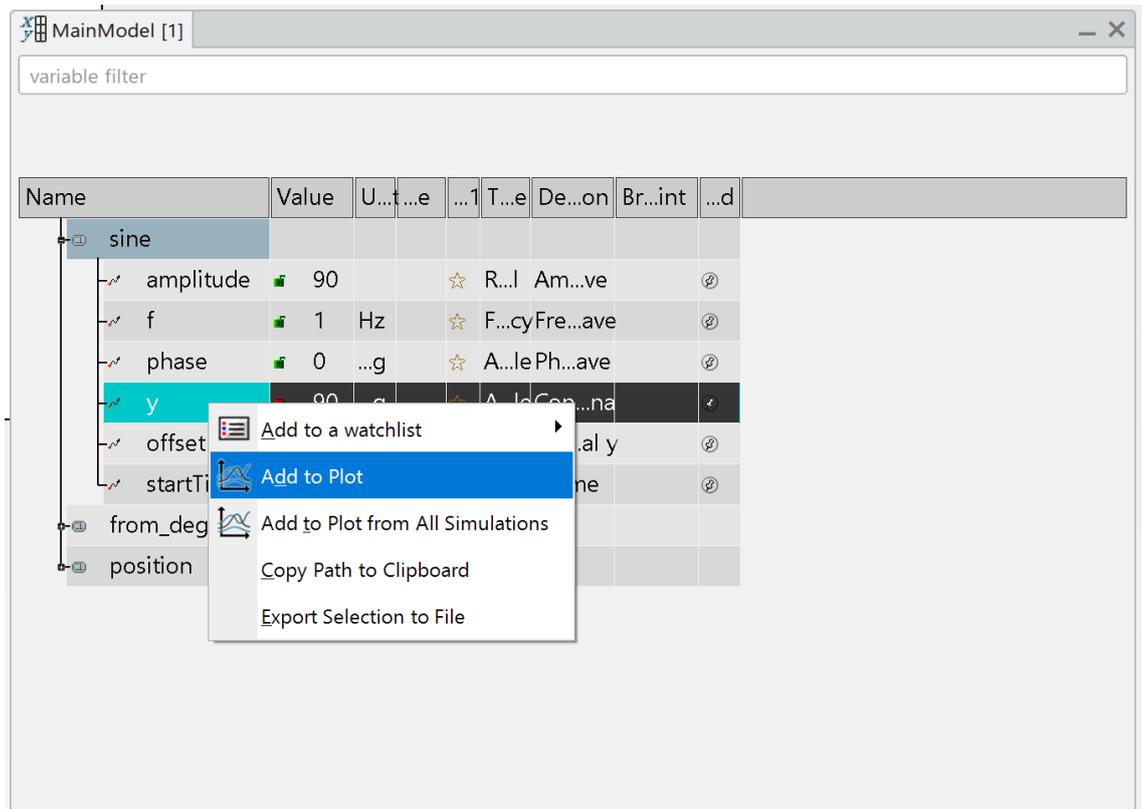
L'écran suivant apparaît :



Dans la fenêtre « Main Model », cherchez le paramètre « y » de la commande de pilotage « Sine » :



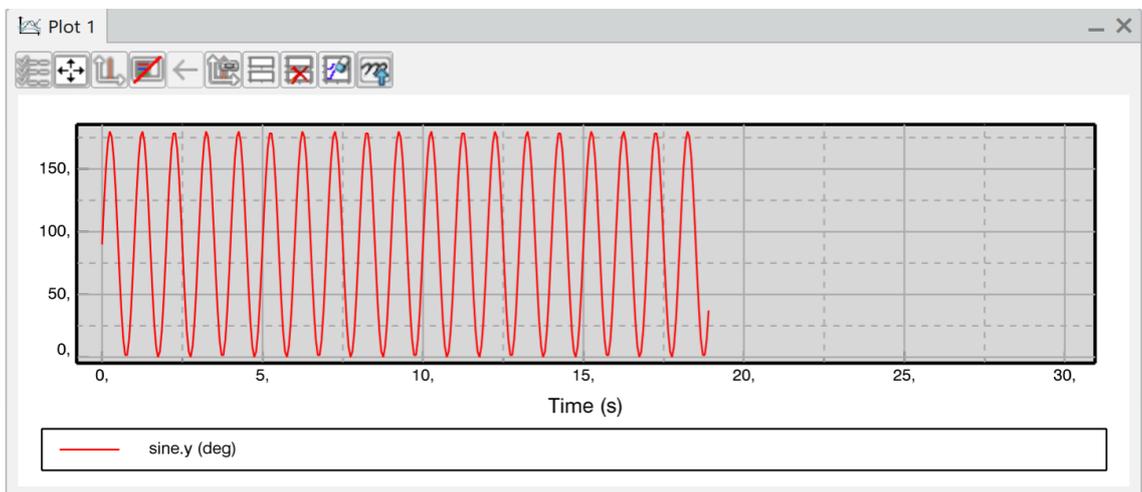
Faites ensuite un clic droit sur cette ligne et cliquez sur « Add to plot » pour afficher la courbe de commande :



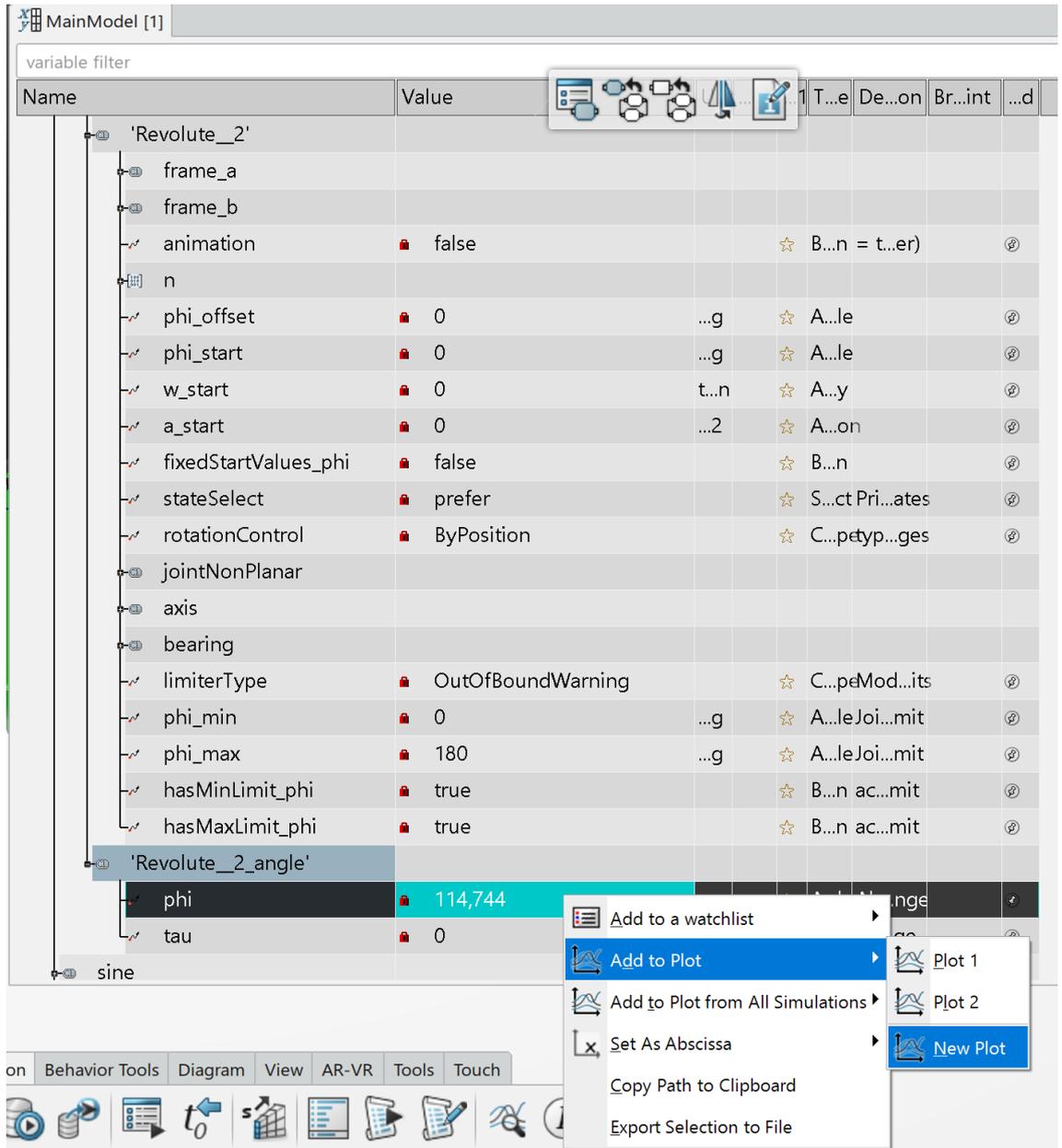
Lancez l'affichage de la simulation en cliquant sur le symbole play de cette barre située en bas à gauche de la fenêtre :



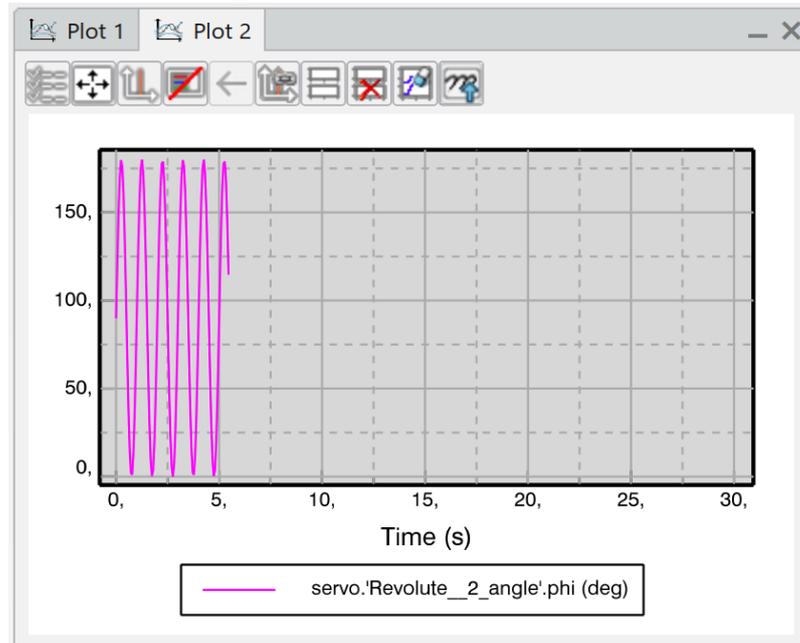
Constatez le tracé de la courbe de commande au cours du temps et la rotation du palonnier dans le modèle 3D :



Cherchez dans la fenêtre « Main model » le paramètre « Phi » de la liaison pivot « Revolute\_\_2\_Angle » :

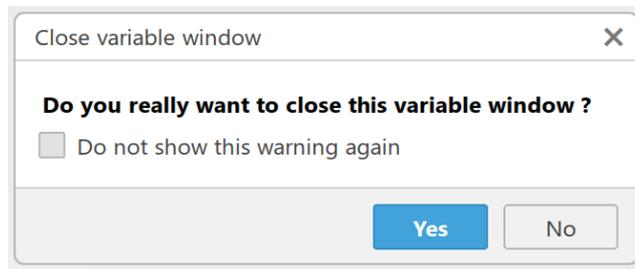


Cliquez sur le bouton droit de la souris et sur « Add to plot » et constatez le résultat suivant :



NB : étant donné que nous avons choisi de réaliser une commande de position, il n'y aura pas d'écart entre la commande et le résultat car l'inertie des pièces n'impacte pas la position angulaire du palonnier au cours du temps. Le couple fourni par le moteur doit cependant évoluer au cours du temps pour compenser les inerties, mais nous le mesurerons pas dans cette exercice car l'objet est avant tout de se focaliser sur le jumeau numérique.

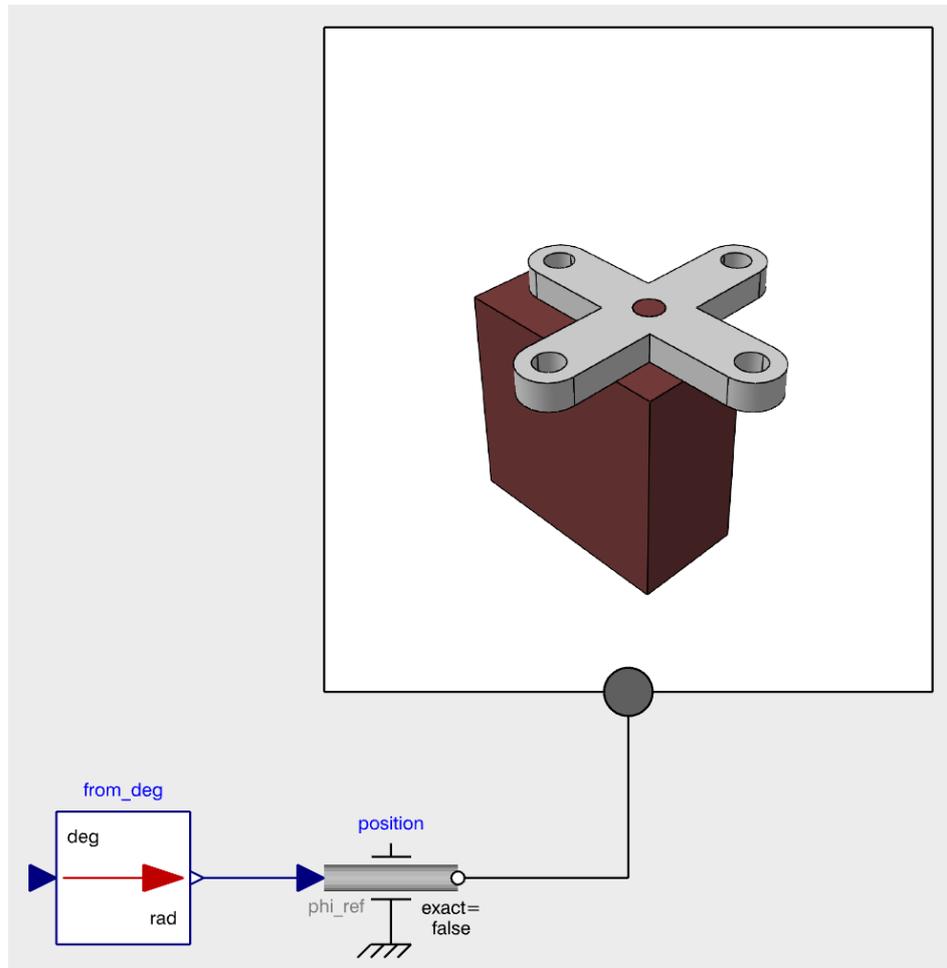
Fermez toutes les fenêtres de la simulation pour passer aux étapes suivantes :



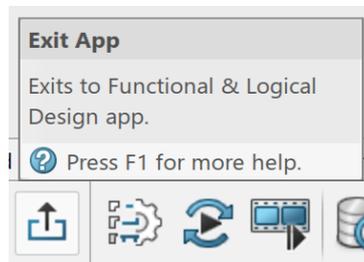
### Création du jumeau numérique de la carte Arduino

Afin de séparer la commande du modèle physique, nous allons restructurer le modèle logique de l'ensemble.

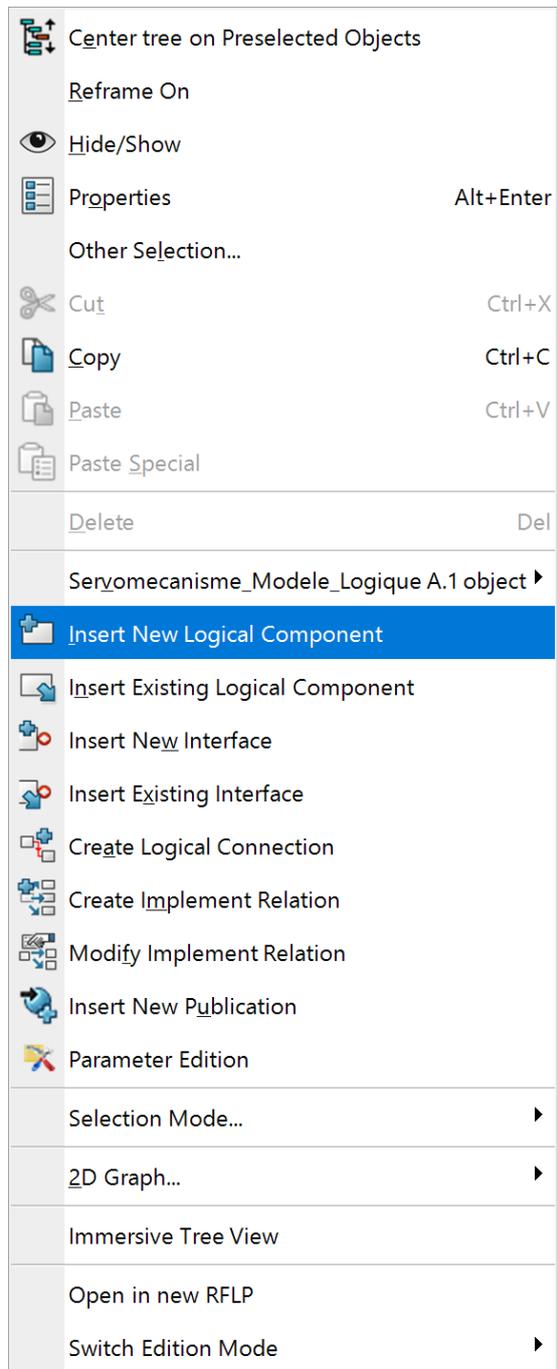
Dans le « Main model du servomécanisme, supprimez la commande « Sine » :



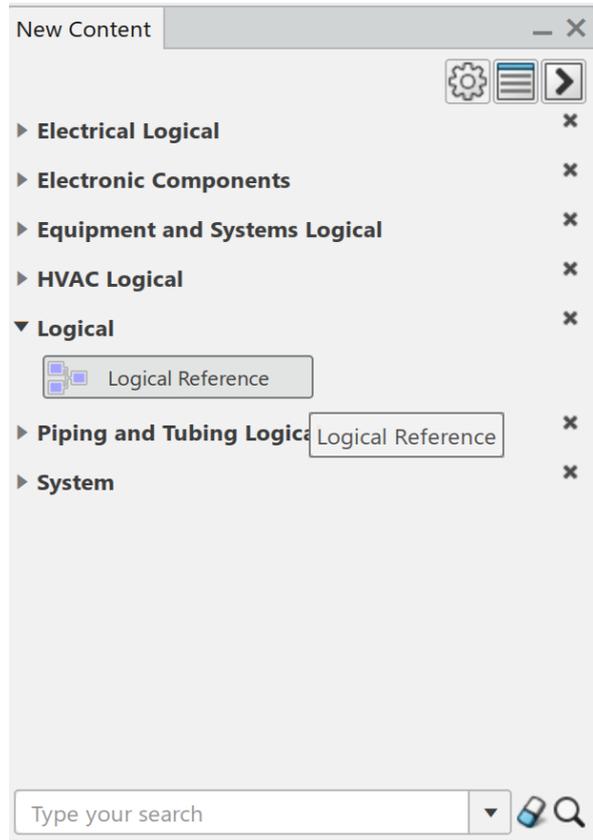
Sortez ensuite de l'application :



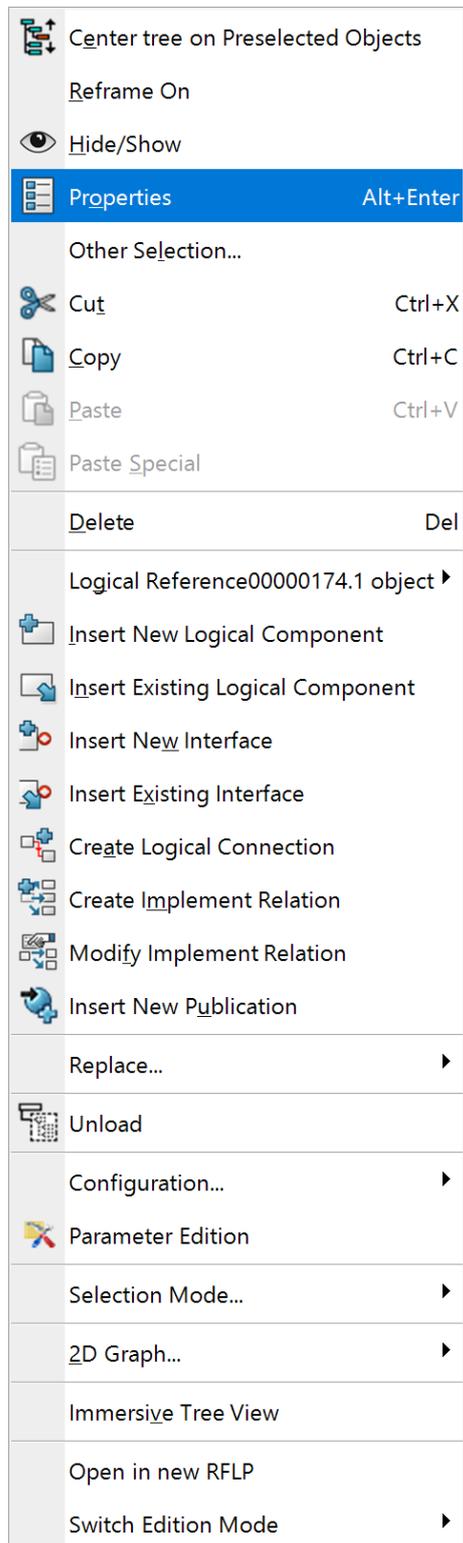
Dans le modèle logique global, cliquez sur le bouton droit et sur « Insert New Logical Component » :



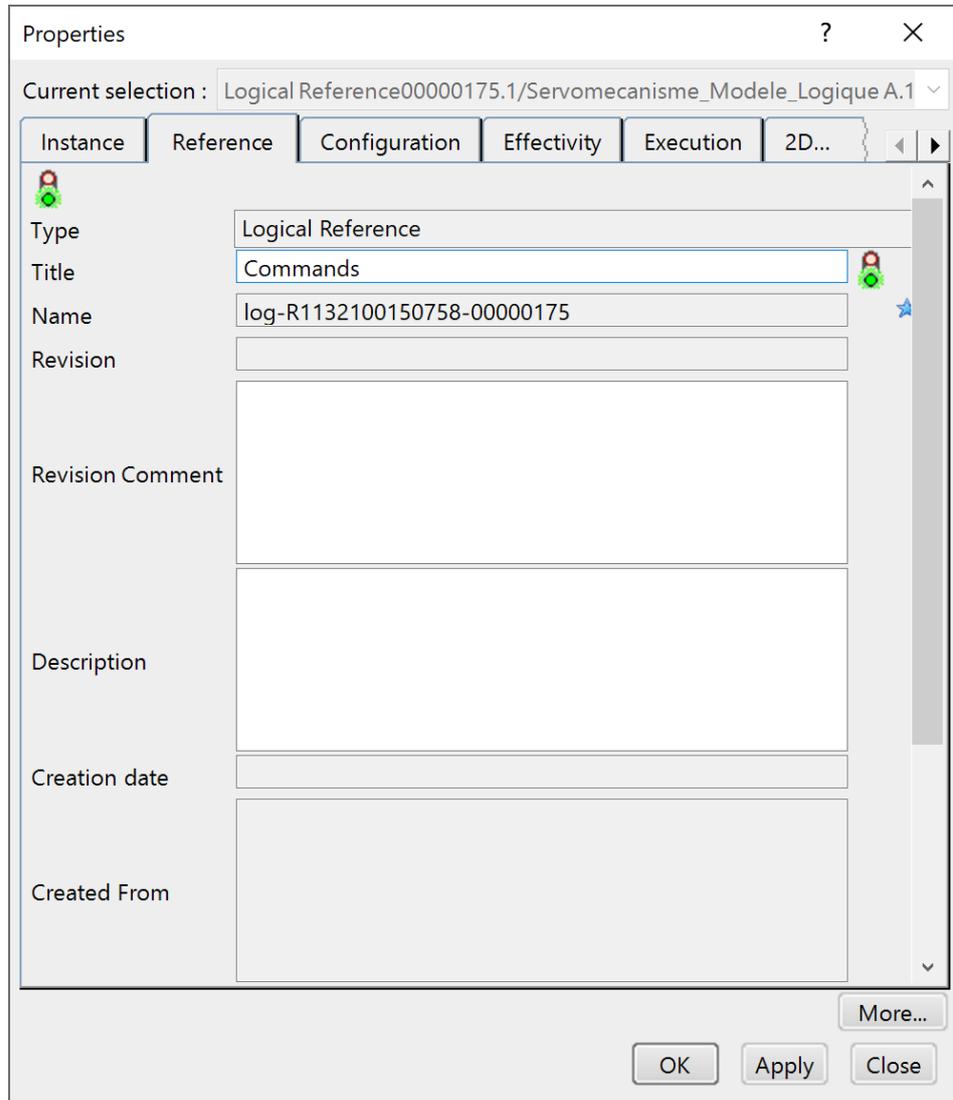
Et cliquez sur Logical Reference :



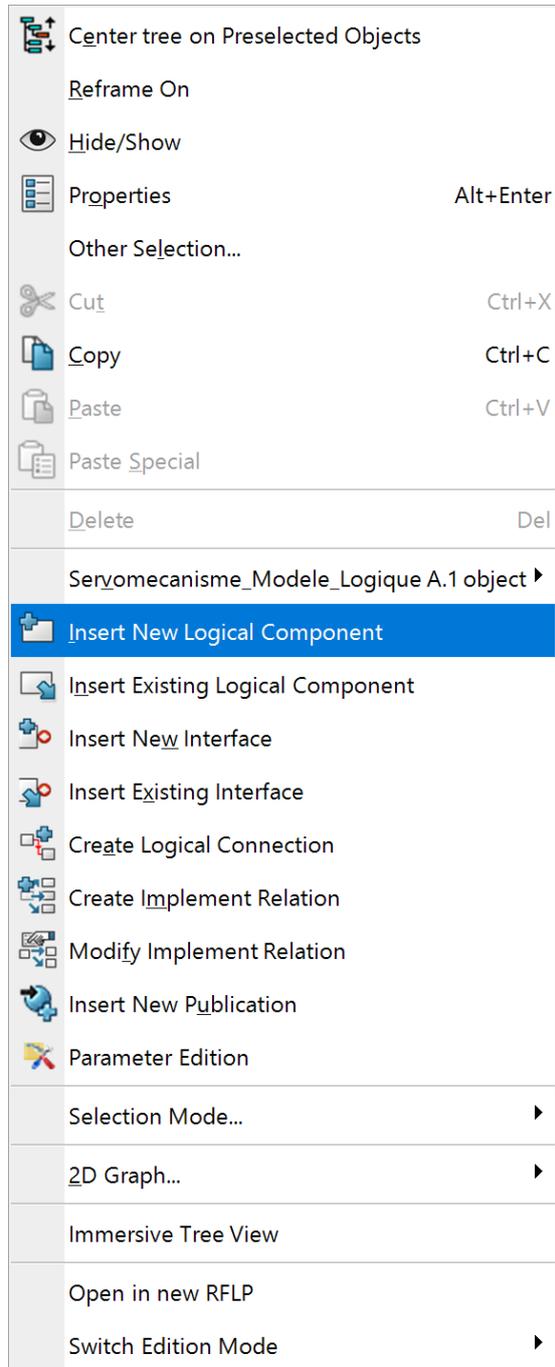
Faites un clic droit sur la nouvelle référence logique créée :



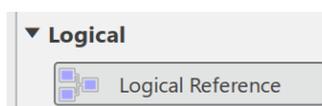
Puis cliquez sur « Properties » pour changer son nom en « Commands », c'est à partir de cette référence logique que le modèle sera piloté :



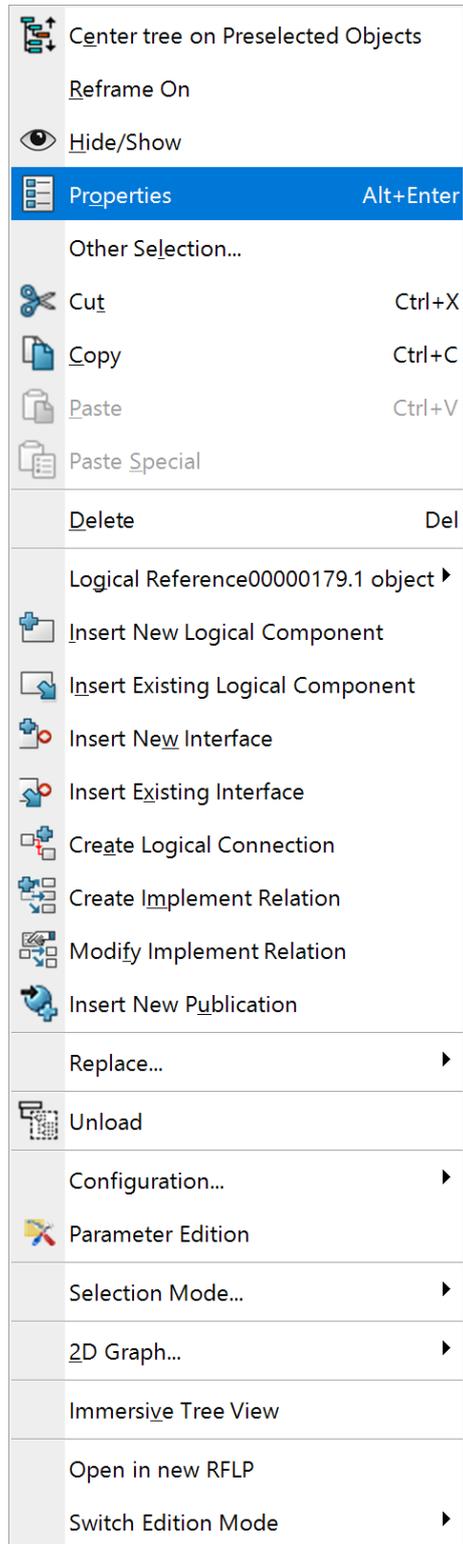
Dans le modèle logique global, cliquez à nouveau sur le bouton droit et sur « Insert New Logical Component » :



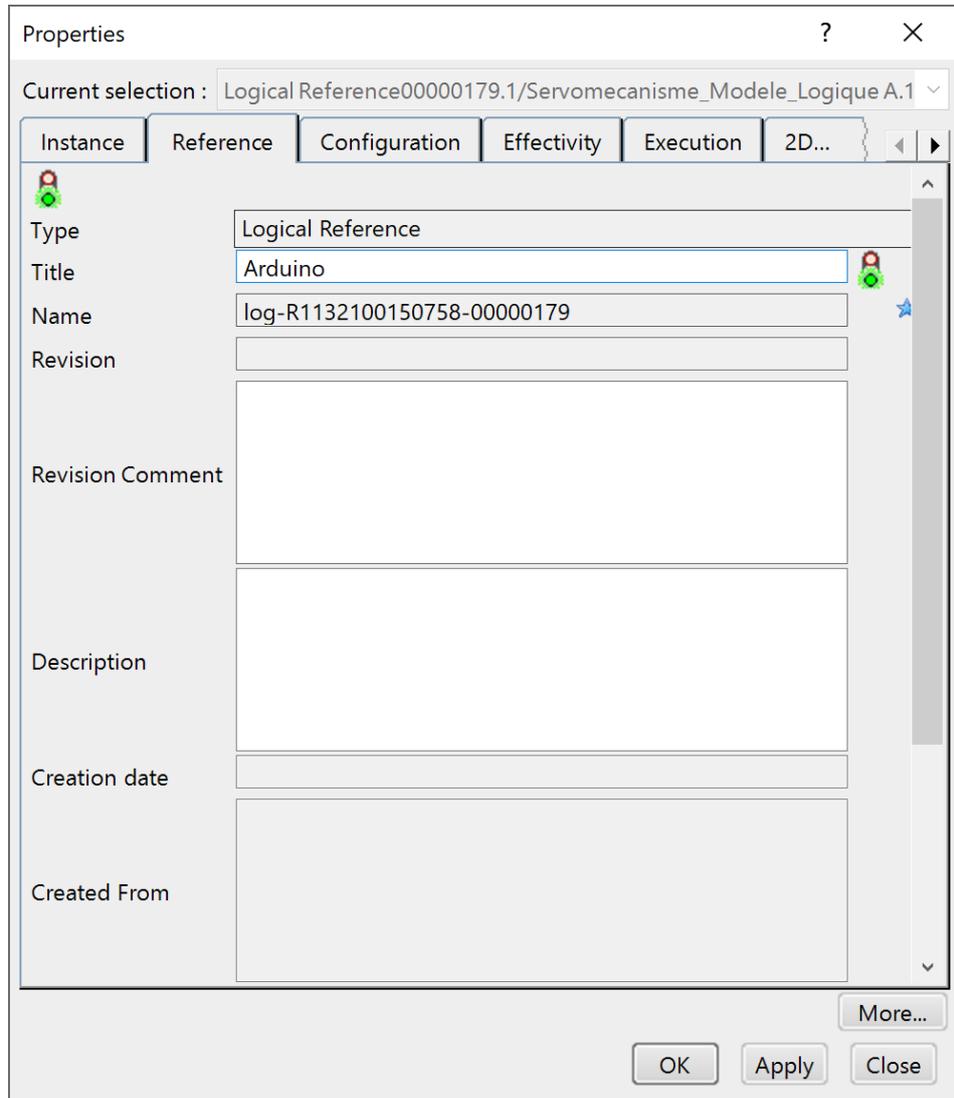
Choisissez Logical Reference :



Faites un clic droit sur la nouvelle référence logique créée :



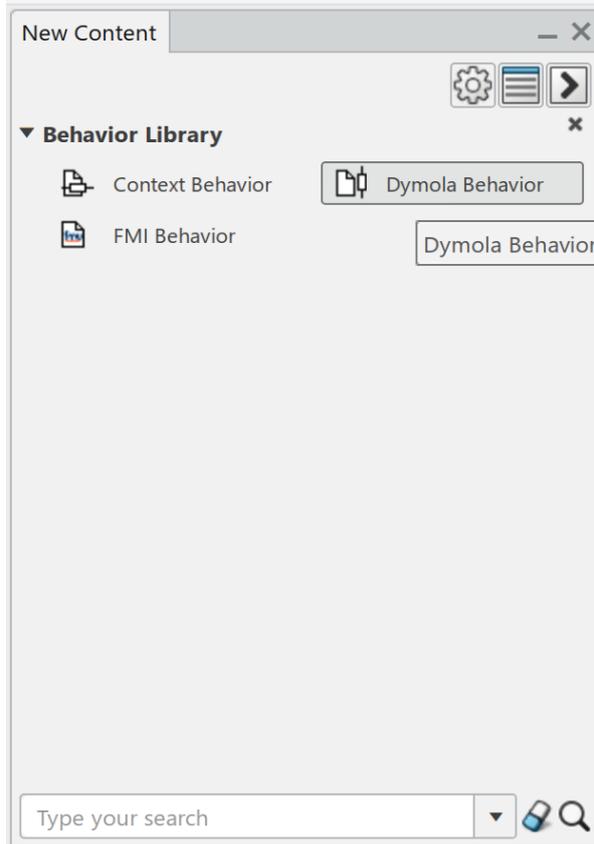
Puis cliquez sur « Properties » pour changer son nom en « Arduino », c'est cette référence logique qui servira d'interface avec la carte Arduino :



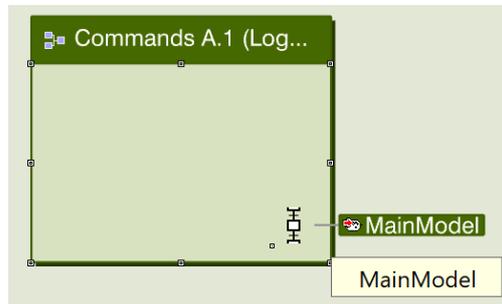
Cliquez sur la référence « logique Commands » puis sur « Insert New Behavior Representation » :



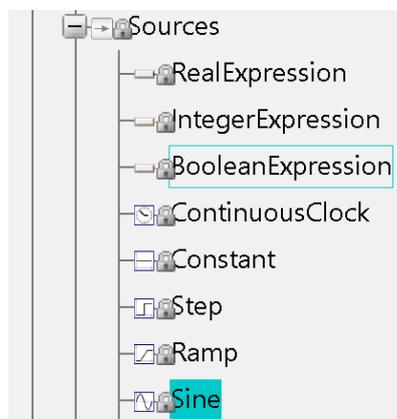
Cliquez sur « Dymola Behavior » :



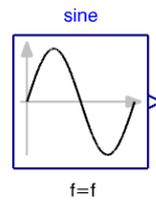
Puis cliquez sur l'icone qui est apparue en bas à droite de la référence logique « Commands », puis double cliquez sur « MainModel » :



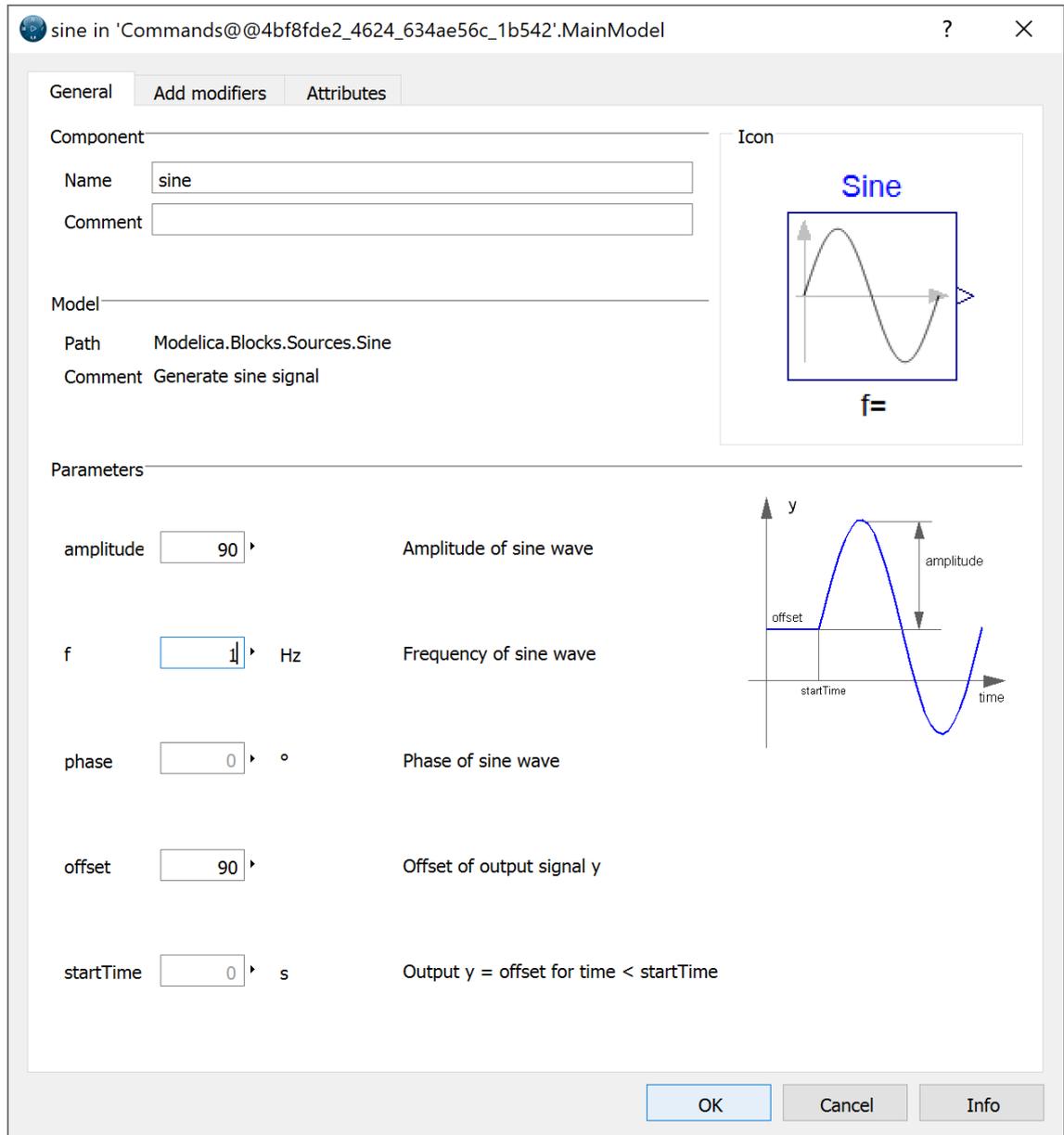
Recherchez ensuite « Sine » dans l'arbre du modèle :



Glissez/Déplacez le bloc Sine dans la fenêtre centrale :



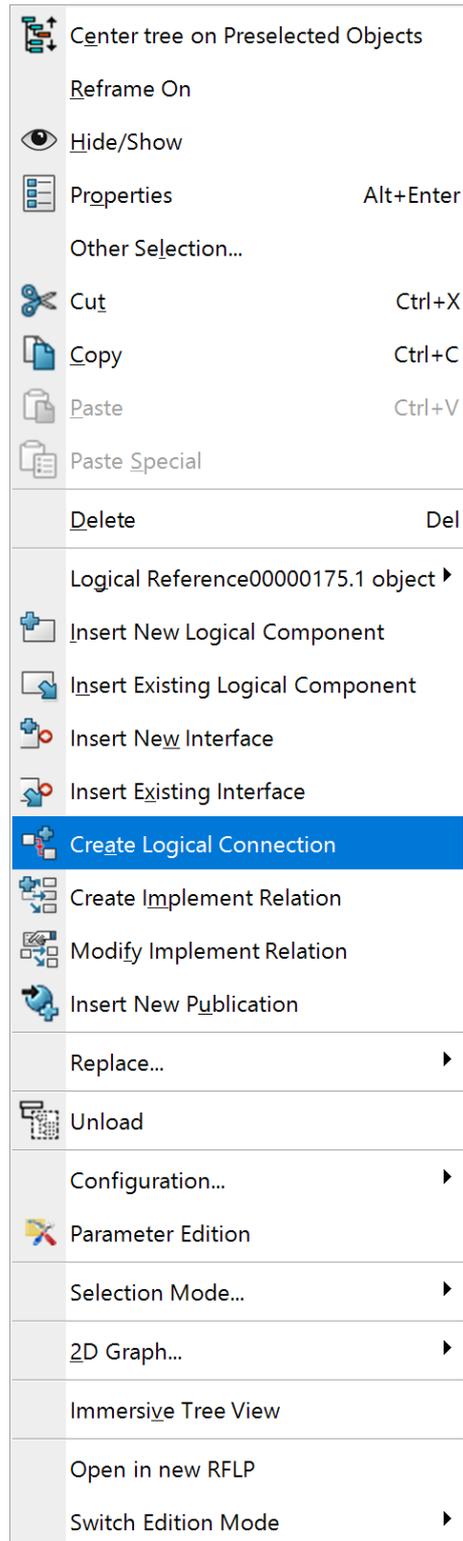
Double-cliquez sur le bloc « Sine » pour entrer 90° dans « amplitude », 1 dans « f » et 90° dans offset :



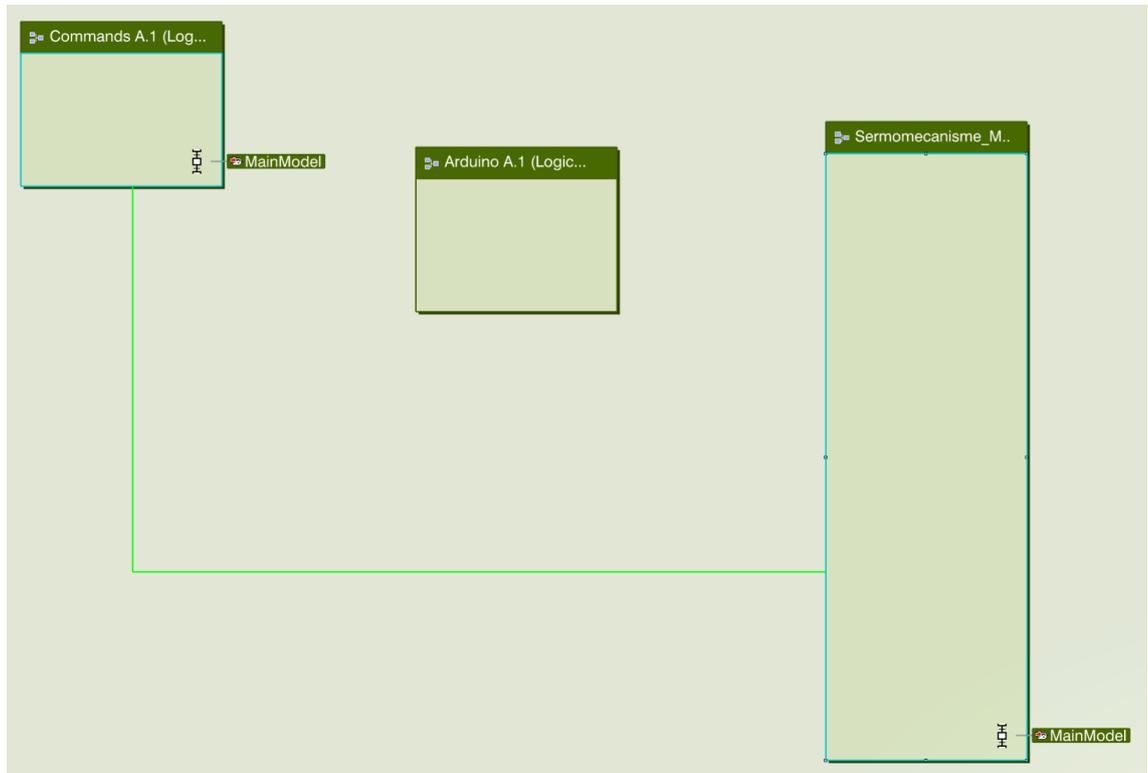
Cliquez ensuite sur « Exit App » :



Cliquez avec le bouton droit sur la référence logique « Commands » puis sur « Create Logical Connection » :



Créez cette connexion entre les références logiques « Commands » et « Servormecanisme » :



Renommez la en « System\_Type\_Reference » puis cliquez sur « Next » :

System Type Association

No System Type  
 Associate Existing System Type No System Type ...  
 Associate New System Type System Type Reference

System Type Reference     System Type Interface Instance

Title System\_Type\_Reference

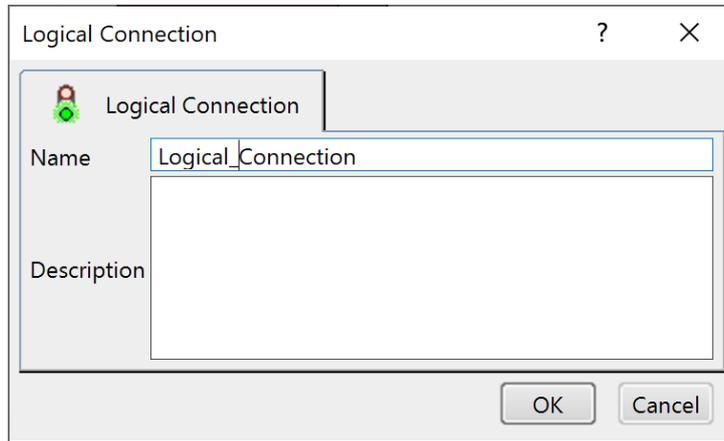
Name typ-R1132100150758-00000169 ★

Description

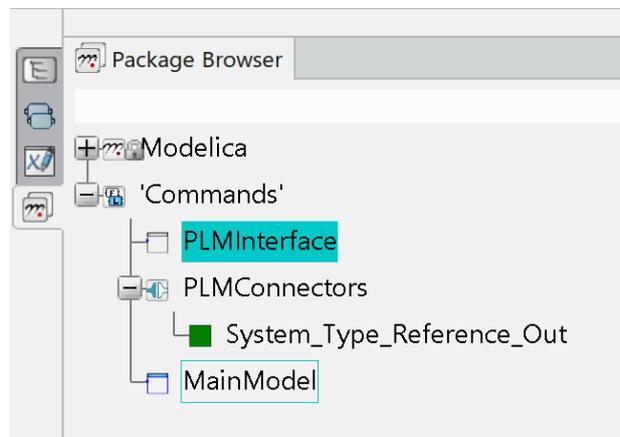
Collaborative Policy Engineering Definition Classic

Previous Next Finish Cancel

Ensuite, renommez la en « Logical Connection » :



Dans la référence logique « Commands », cliquez sur l'icone en bas à droite puis double cliquez sur « MainModel » :

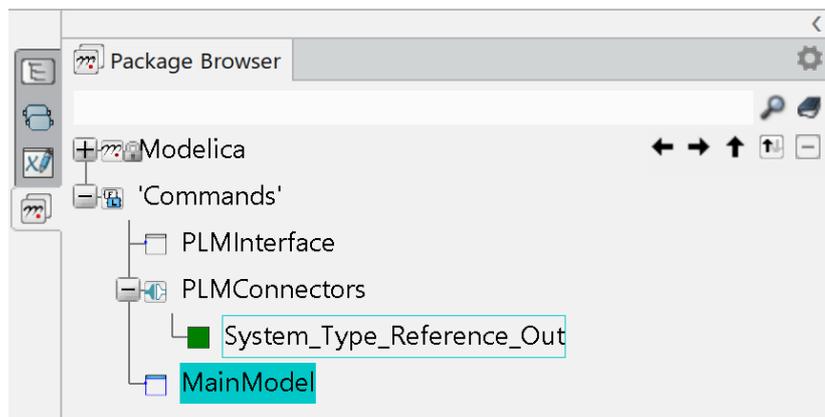


Cliquez sur « PLMInterface » dans l'arbre du modèle, puis déplacez le connecteur « System\_Type\_Reference » pour le disposer à droite de la fenêtre, à peu près à la même hauteur que le bloc « Sine » :

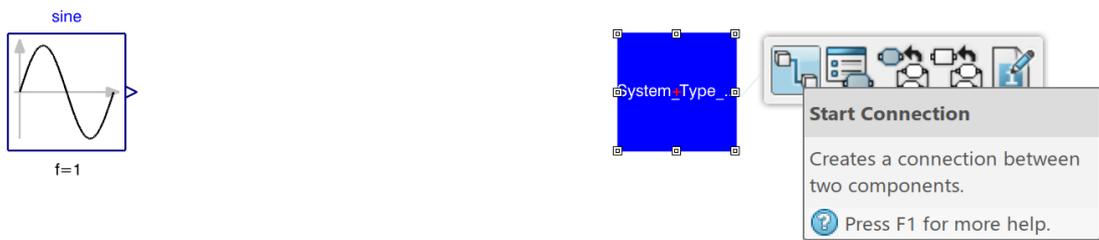


NB : Remarquez qu'il faut être dans « PLMInterface » pour pouvoir déplacer les connecteurs (blocs bleus).

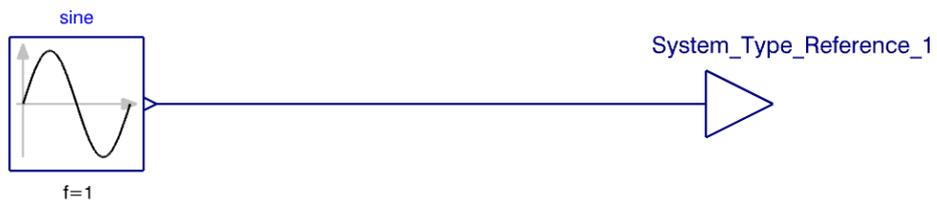
Cliquez ensuite sur MainModel dans l'arbre du modèle pour faire réapparaître le bloc « Sine » :



Cliquez sur le connecteur « System\_Type\_Reference » puis sur l'icone « Start Connection » qui apparait dans le menu contextuel :



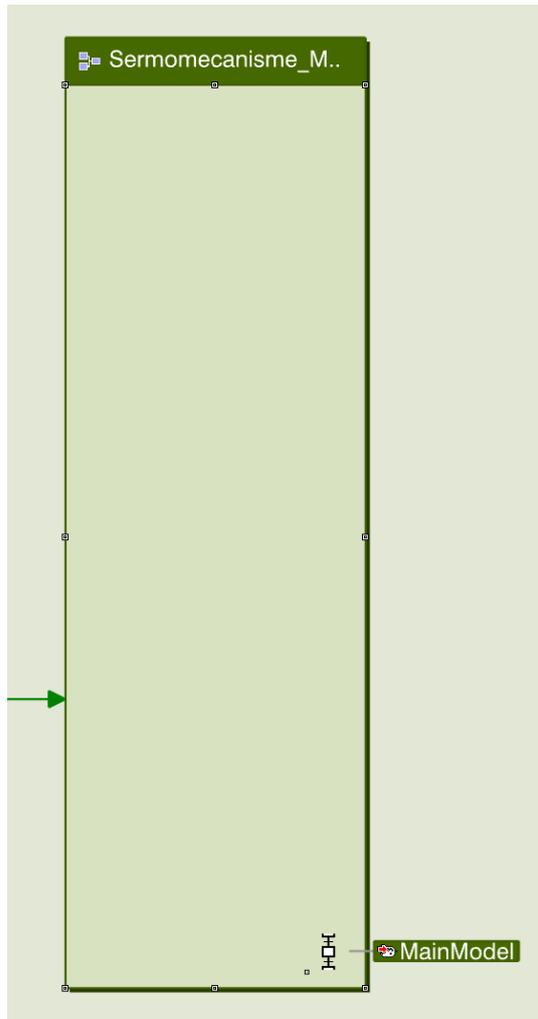
Reliez ensuite le connecteur au bloc « Sine » :



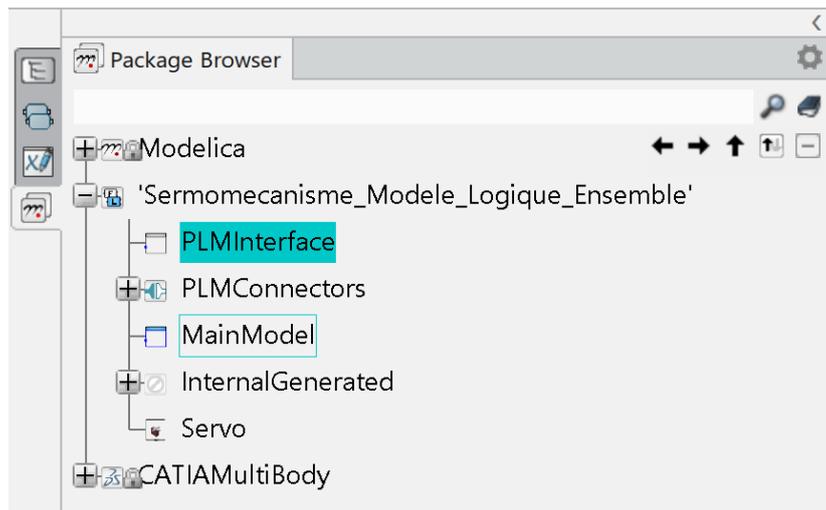
Puis cliquez sur « Exit App » dans la barre d'icônes du bas :



Cliquez sur la reference logique « Servomecanisme... », puis double cliquez sur « MainModel » :



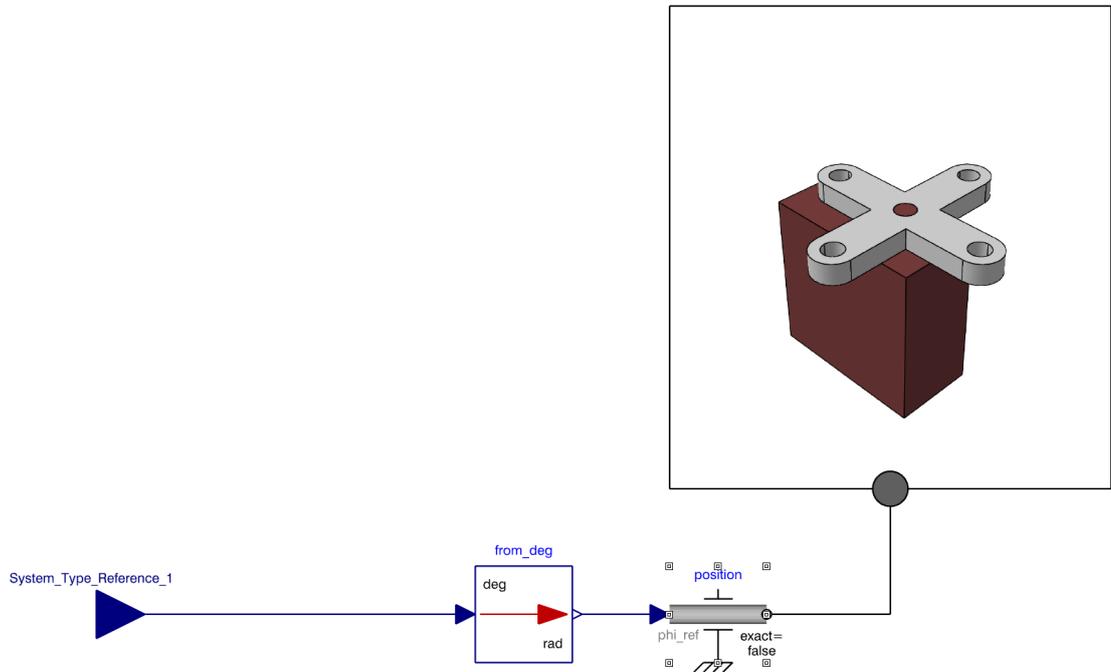
Cliquez sur « PLMinterface » et déplacez le connecteur à gauche de la fenêtre, à peu près au même niveau que le bloc « from\_deg » :



Cliquez sur le connecteur puis sur « Start Connection » et reliez le au bloc « from\_deg » :



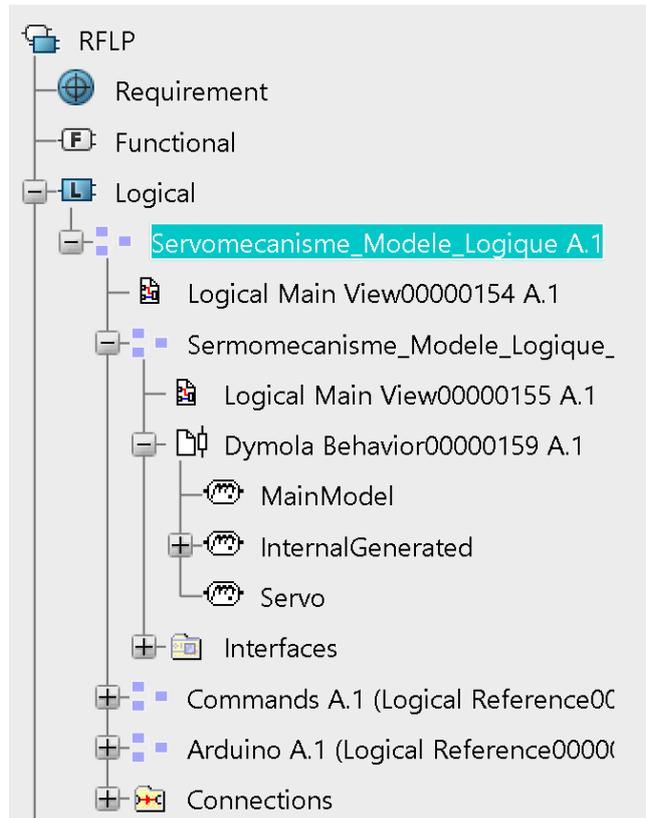
Le résultat doit être le suivant :



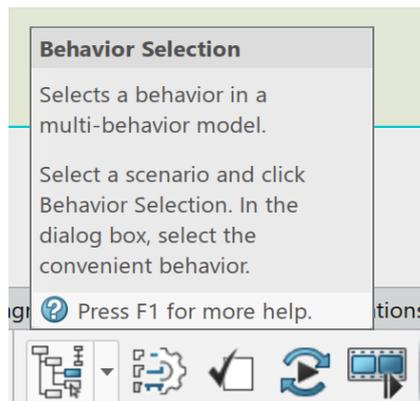
Cliquez sur « Exit App » :



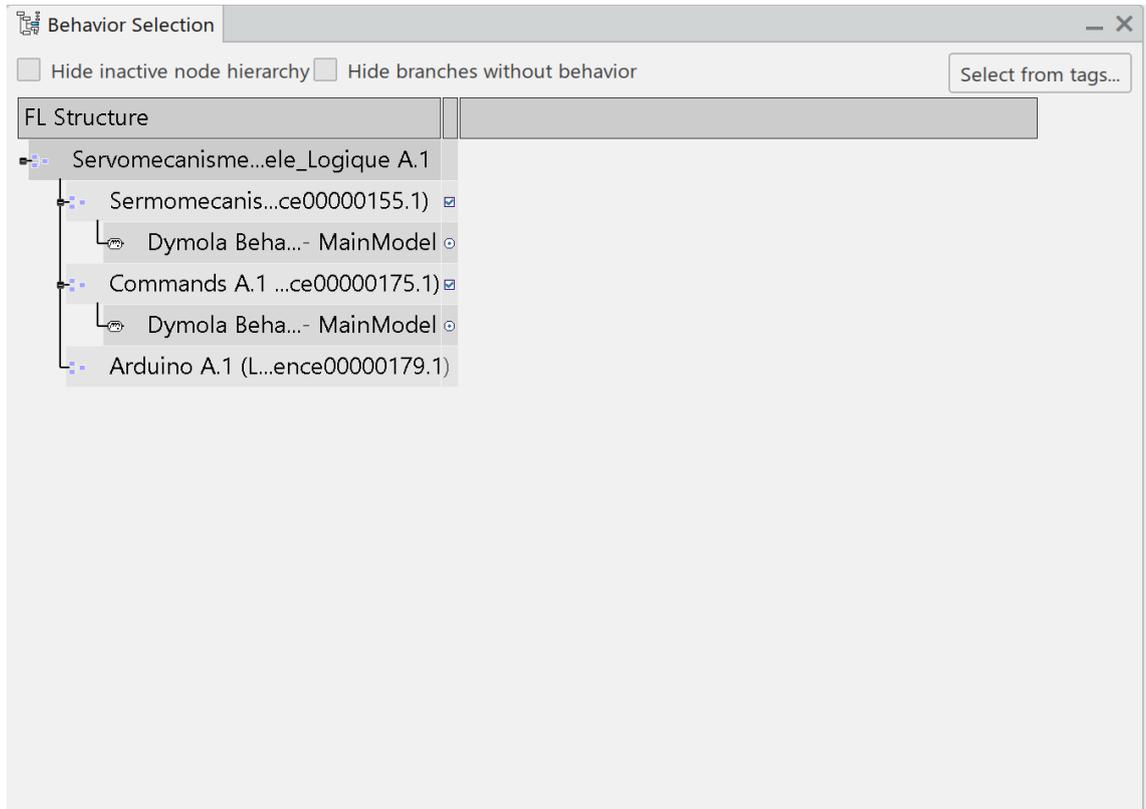
Cliquez sur la plus haute référence logique de l'arbre du modèle :



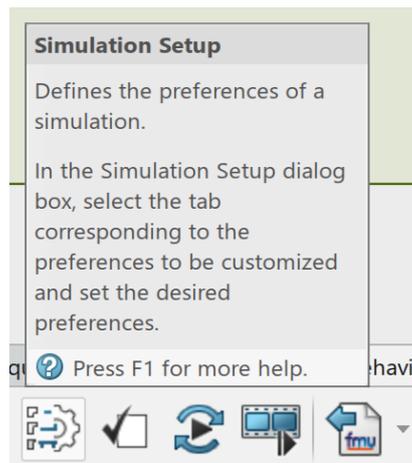
Puis cliquez sur « Behaviour Selection » :



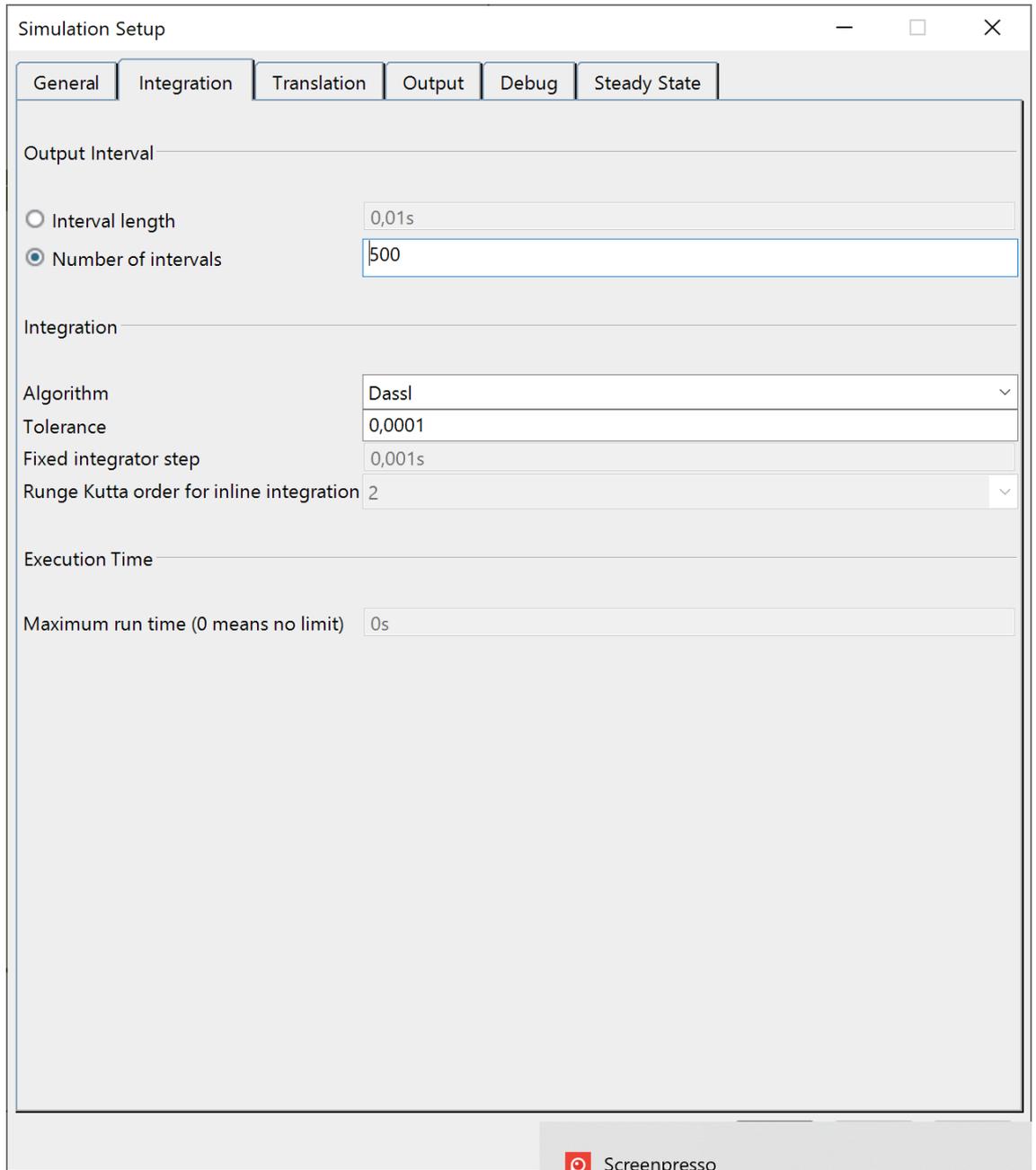
La fenêtre suivante apparait, vous pouvez constater que les deux modèles de simulation ont activés. Cette fenêtre permet de gérer des alternatives de simulation en cochant ou décochant les modèles qui seront simulés :



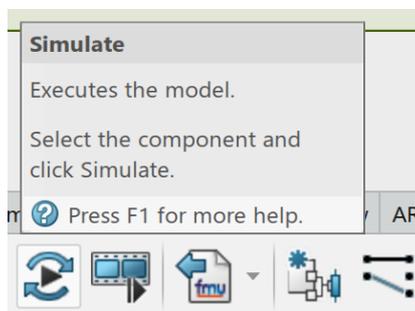
Fermez cette fenêtre puis cliquez sur Simulation Setup :



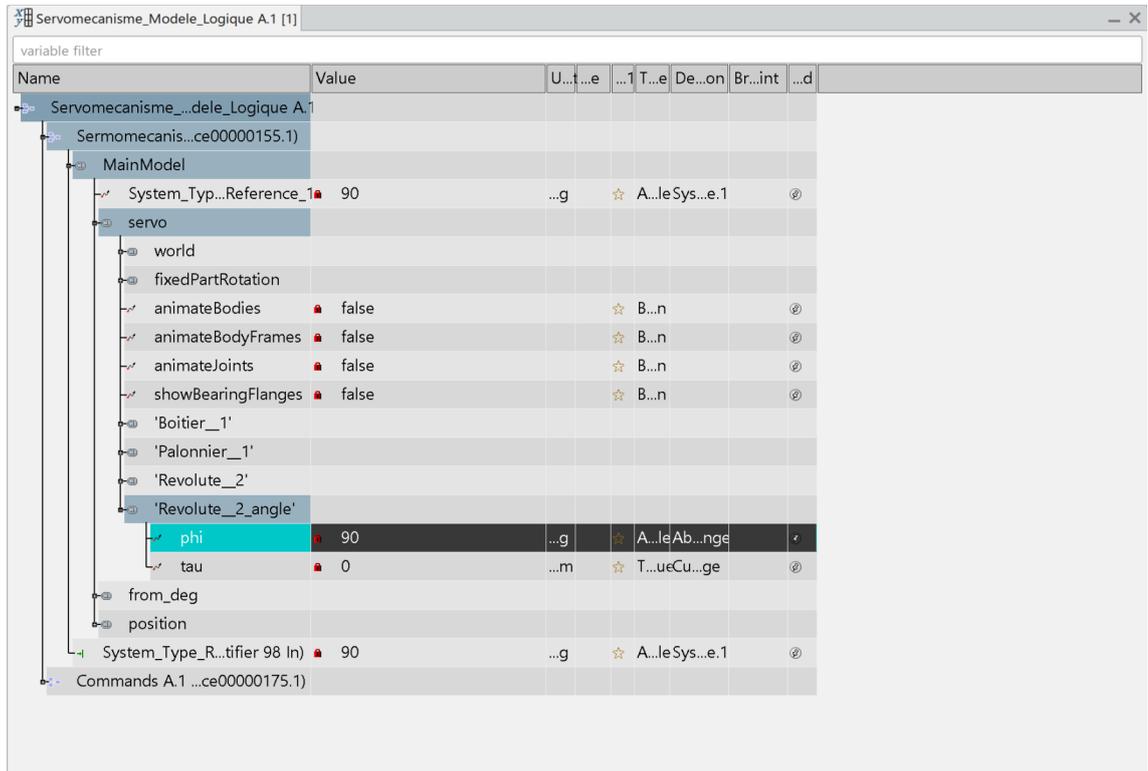
Ensuite réglez le « Start time » à 0, puis le « Stop time » à 30s, cochez « Interactive simulation » puis cliquez sur OK :



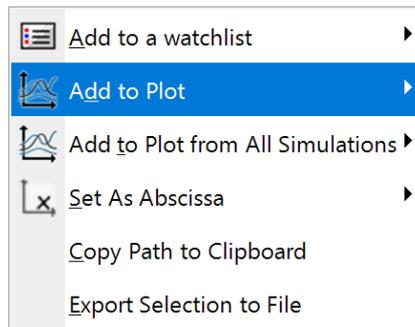
Cliquez ensuite sur Simulate :



Cherchez l'angle « Phi » dans la liaison pivot « Revolute\_2\_angle » :



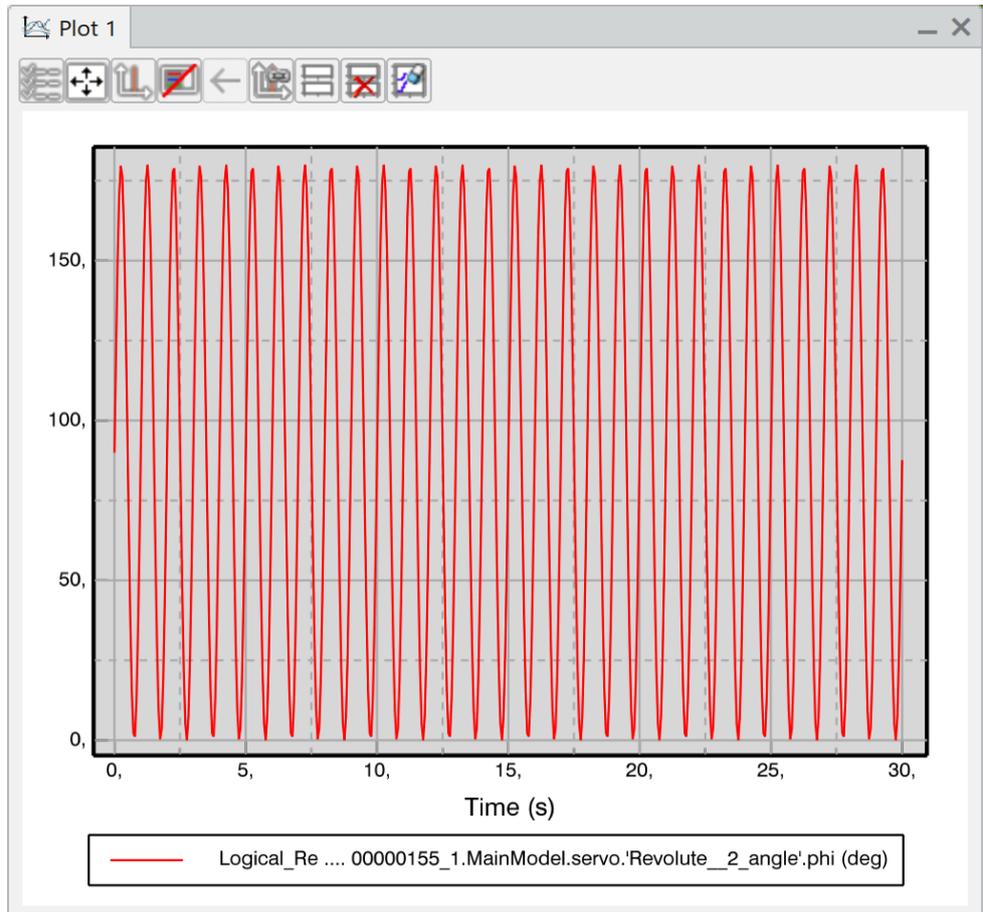
Puis effectuez un clic droit et choisissez ensuite « Add to plot » :



Cliquez sur play dans la fenêtre située en bas à gauche :



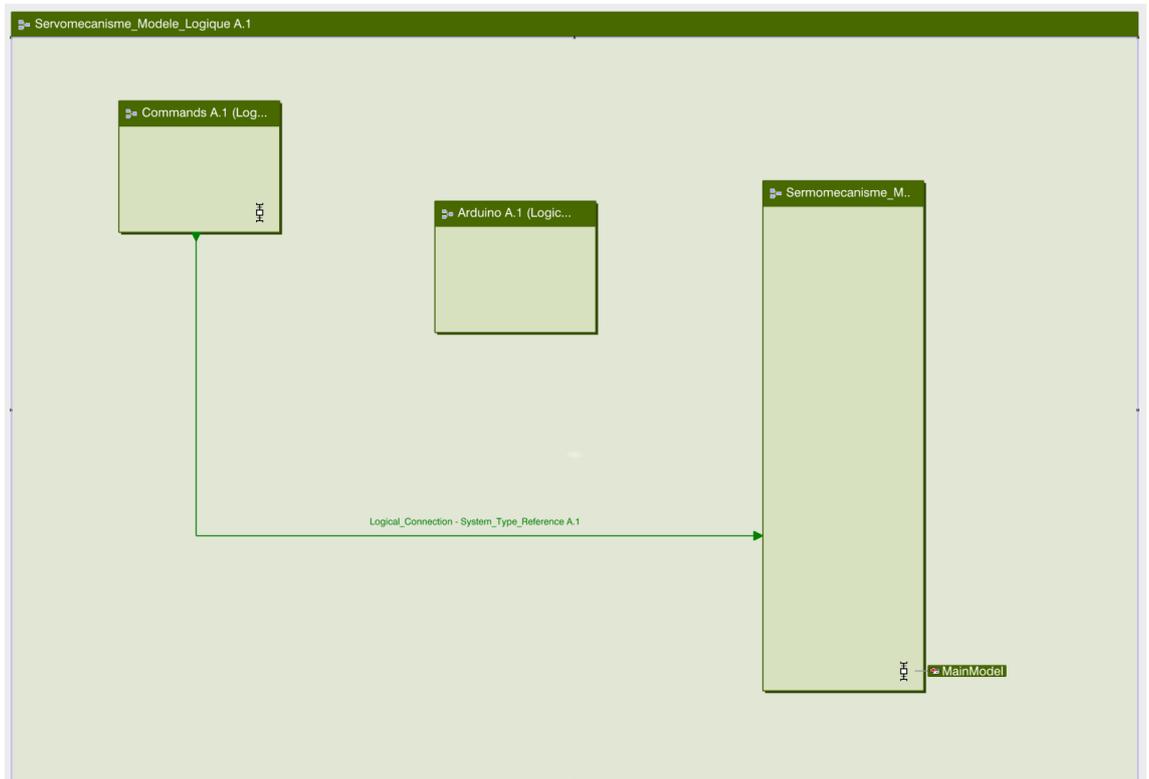
Puis observez le résultat :



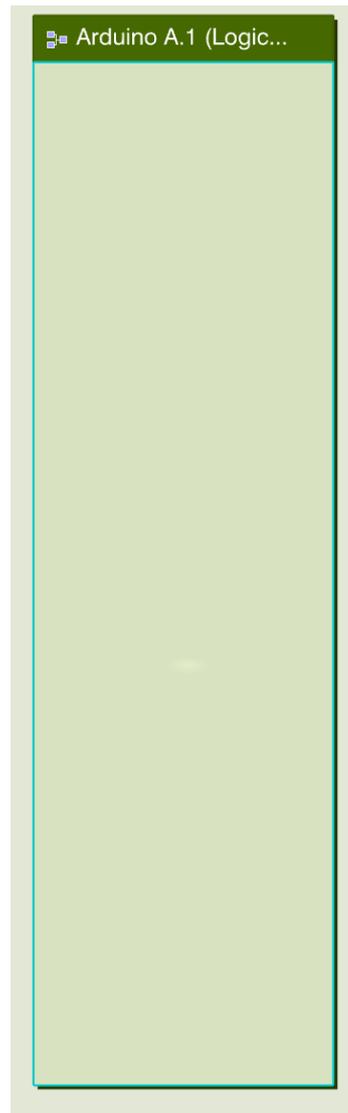
En procédant ainsi, nous avons pu tester que la commande fonctionnait correctement.

Il est désormais temps d'interfacer le modèle 3D avec la carte Arduino.

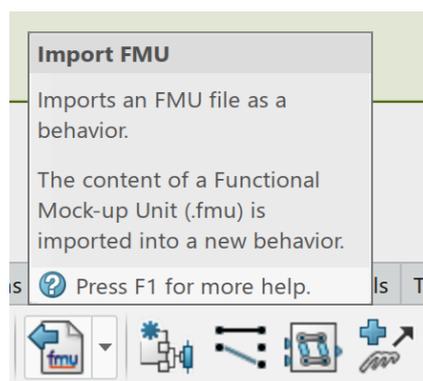
Pour ce faire, sortez de la simulation et effacez la connexion entre les références logiques « Commands » et « Servomecanisme... » et effacez les connecteurs résiduels.



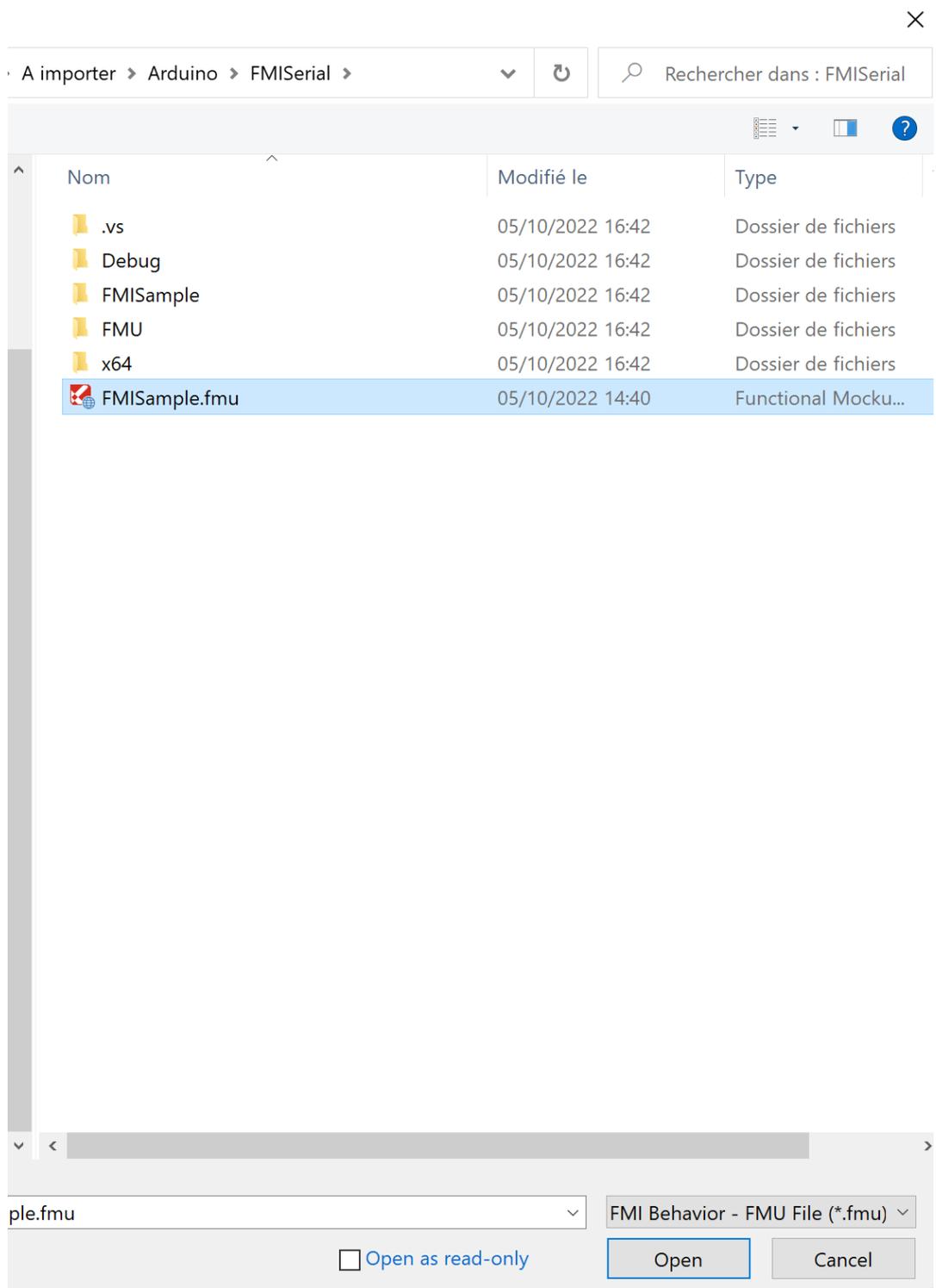
Cliquez sur la référence logique « Arduino » :



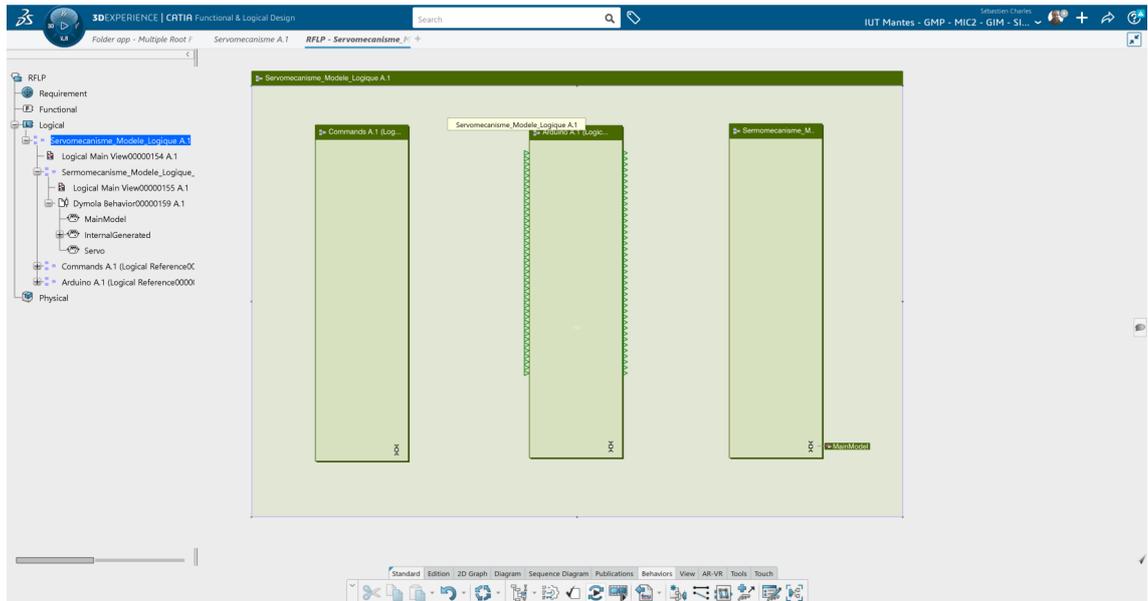
Puis sur « Import FMU » :



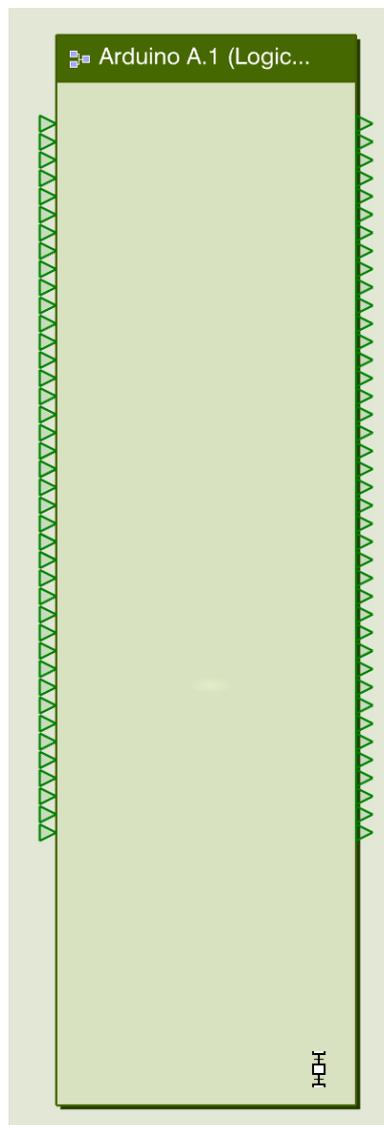
Puis cherchez « FMISample.fmu » dans les ressources téléchargées :



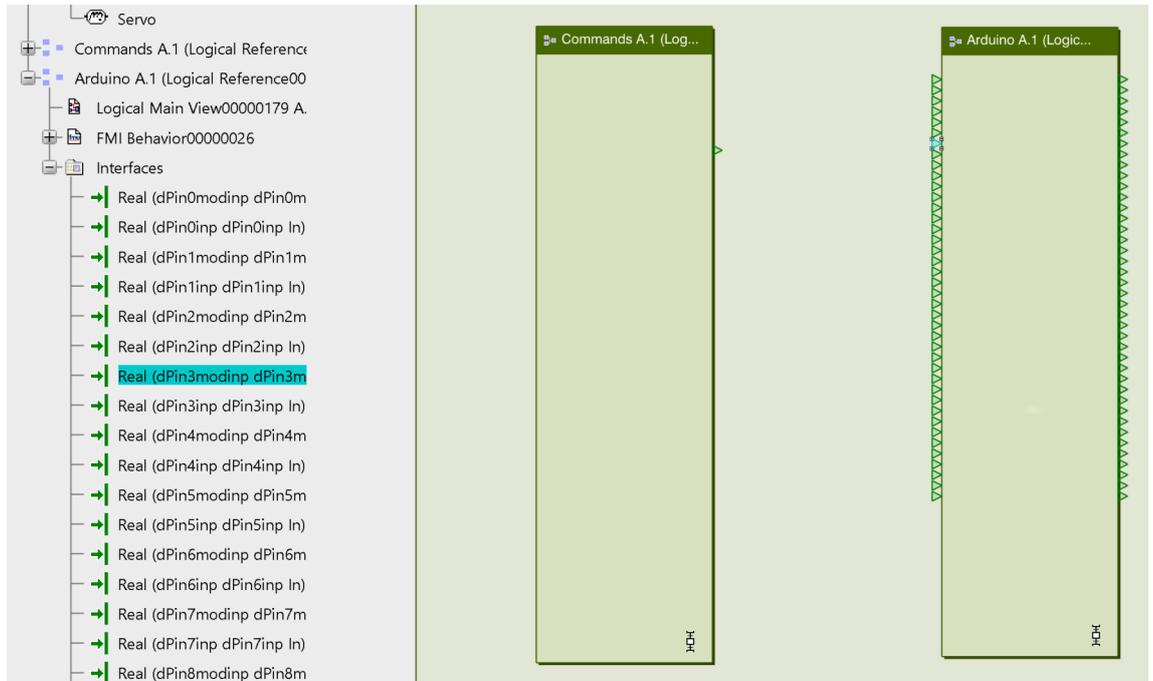
La référence logique « Arduino » se dote alors de nombreux connecteurs d'entrées / sorties qui correspondent aux entrées / sorties physiques de la carte Arduino :



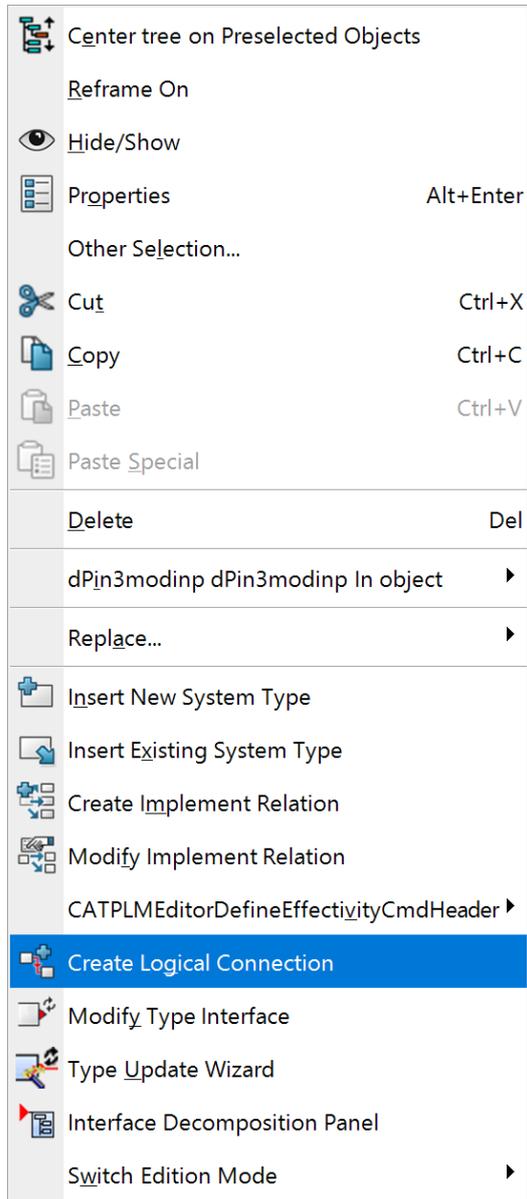
Chaque connecteur de gauche est une entrée, et chaque connecteur de droite une sortie :



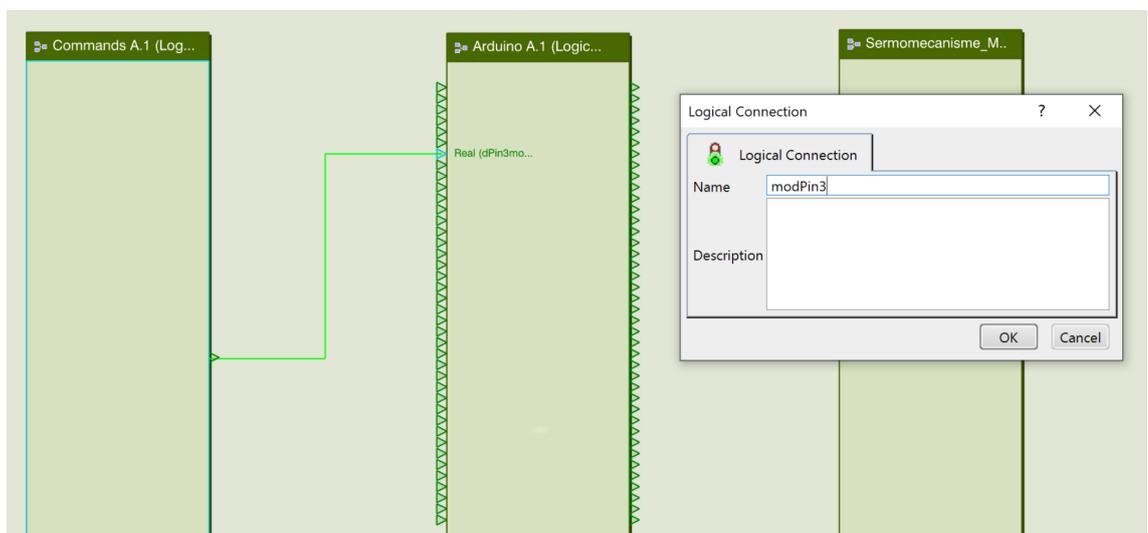
Sélectionnez le connecteur « dPin3modinp » dans la fenêtre principale ou dans l'arbre du modèle :



Puis, cliquez sur le bouton droit de la souris et ensuite sur « Create Logical Connection » :

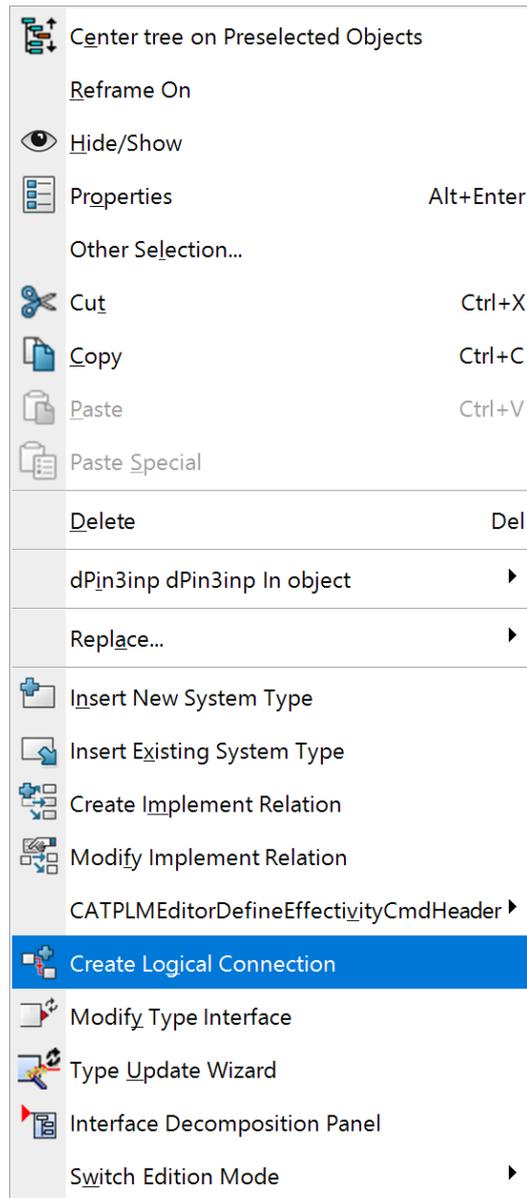


Puis donnez le nom « modPin3 » au connecteur :

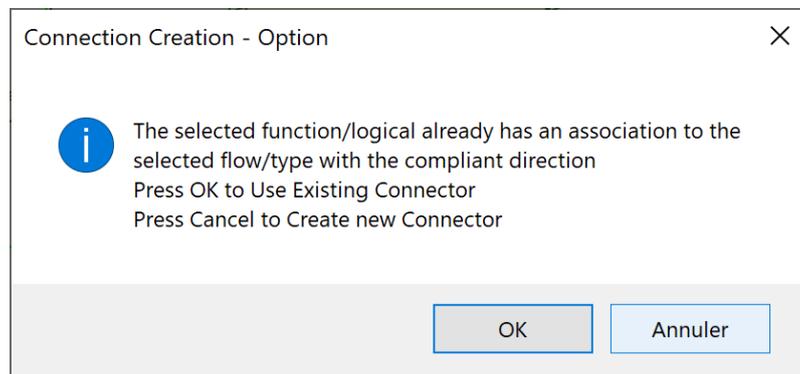


Sélectionnez le connecteur « dPin3inp » dans l'arbre du modèle, puis cliquez sur le bouton droit de la souris et ensuite sur « Create Logical Connection » et cliquez sur Commands pour

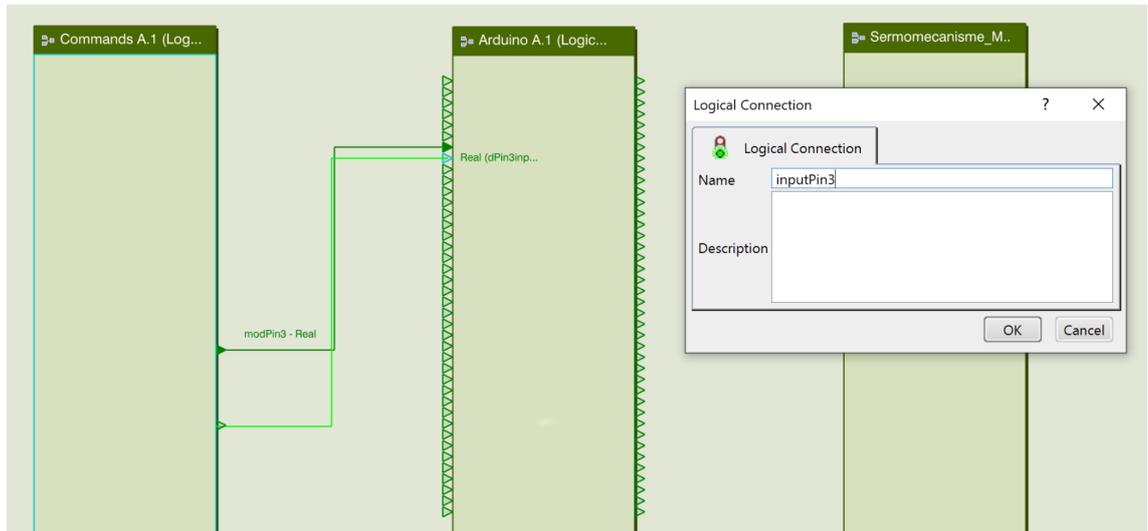
que le lien soit établi entre les deux modèles logiques :



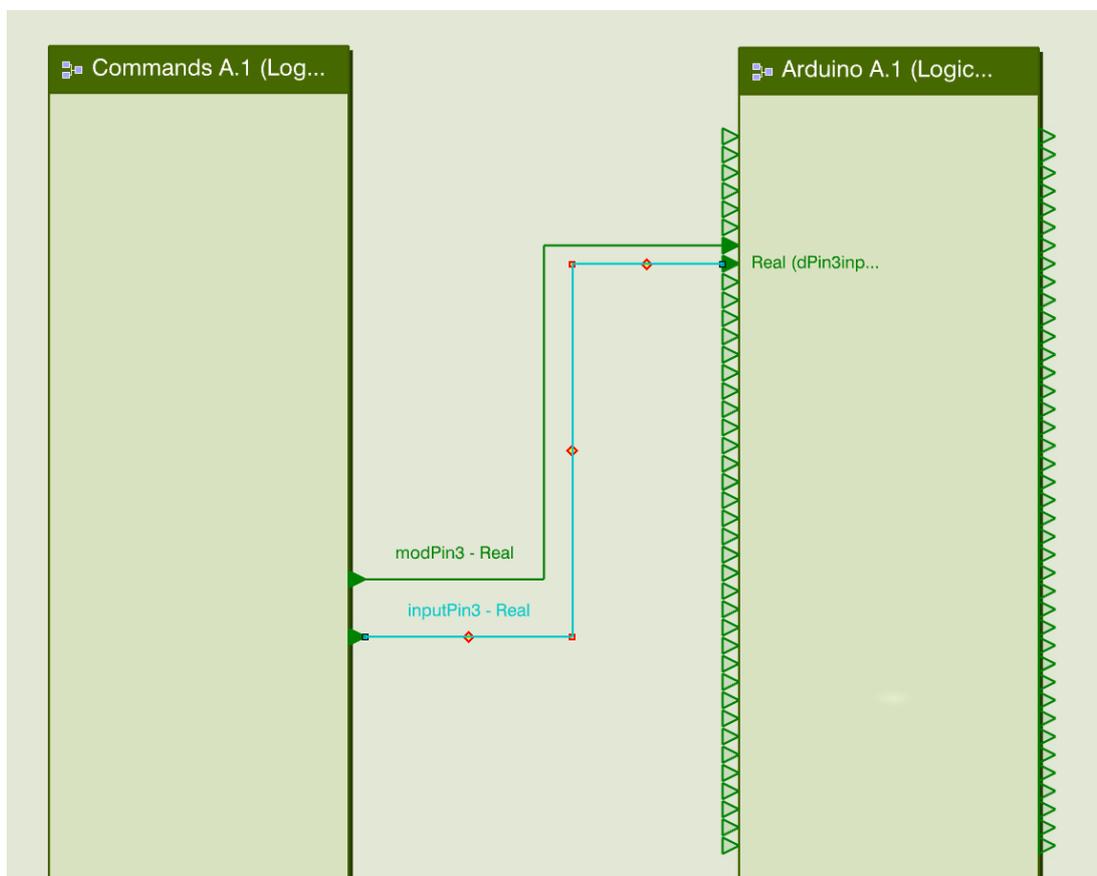
Dans la fenêtre de Connection Creation - Option, cliquez sur Cancel pour créer un nouveau connecteur :



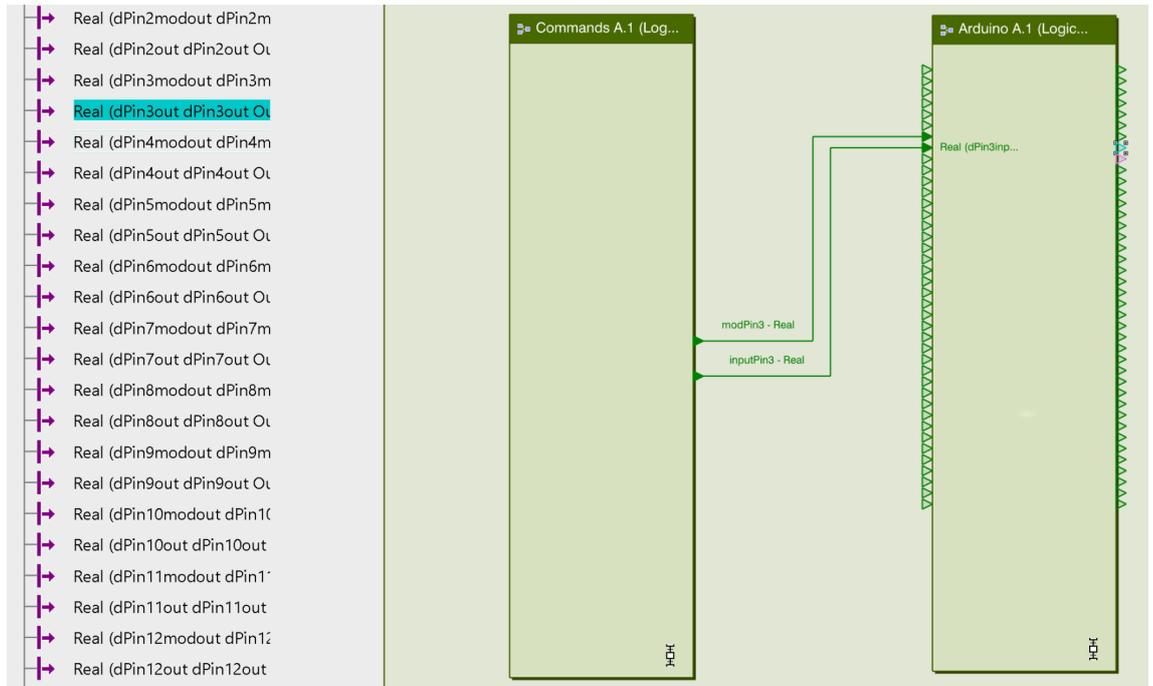
Renommez le connecteur en inputPin3 :



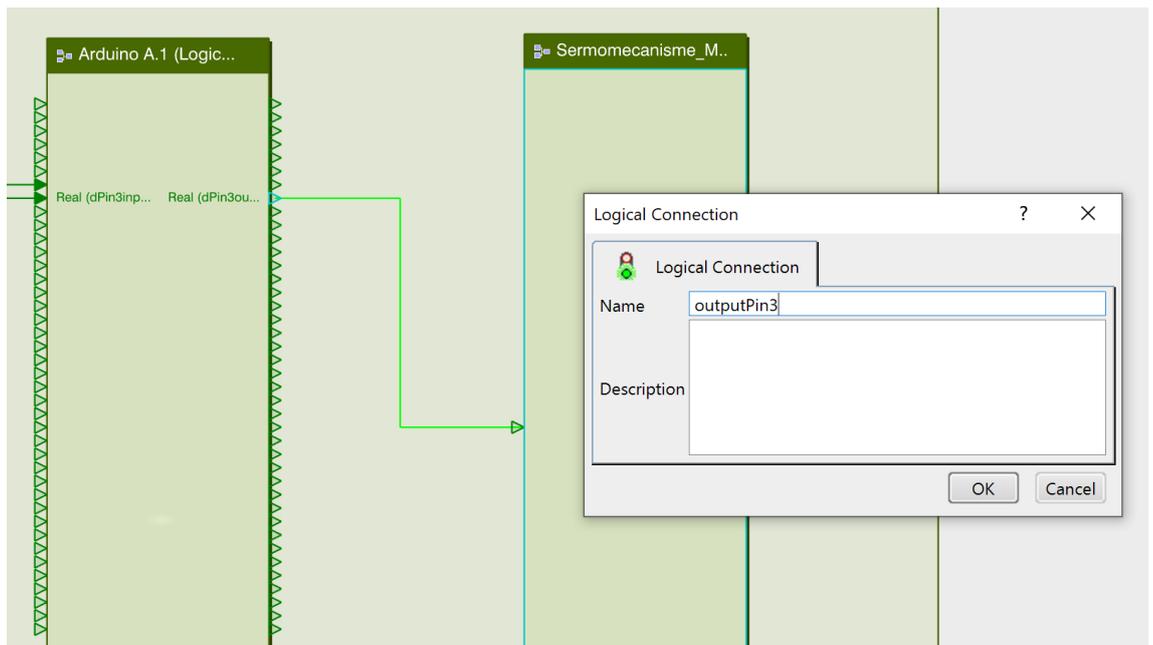
Remplacez les traits pour éviter leurs chevauchements :



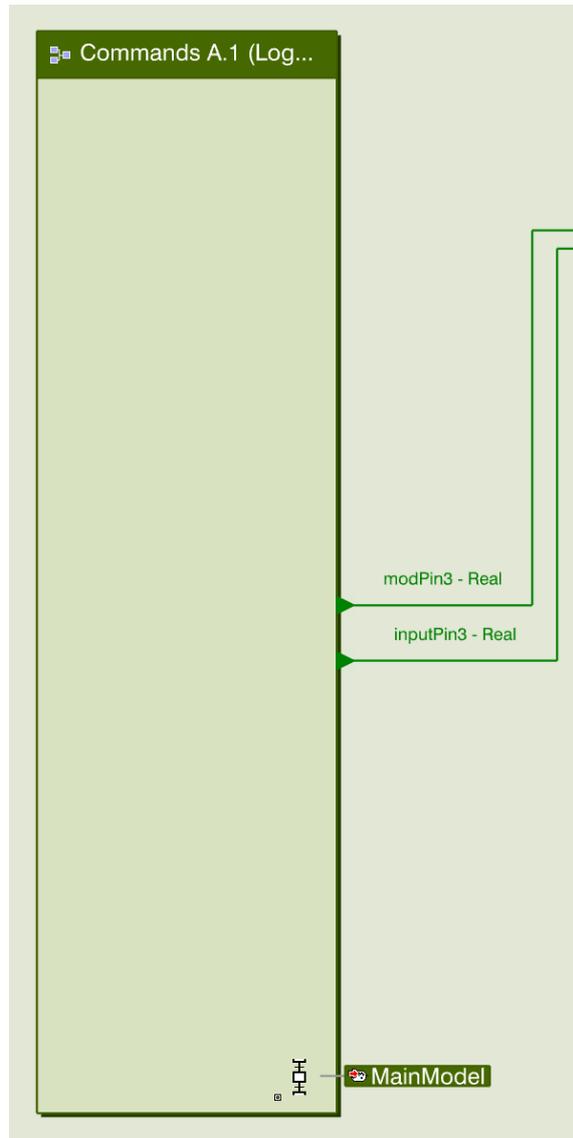
Sélectionnez le connecteur « dPin3out » dans l'arbre du modèle :



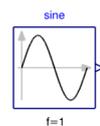
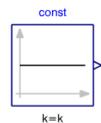
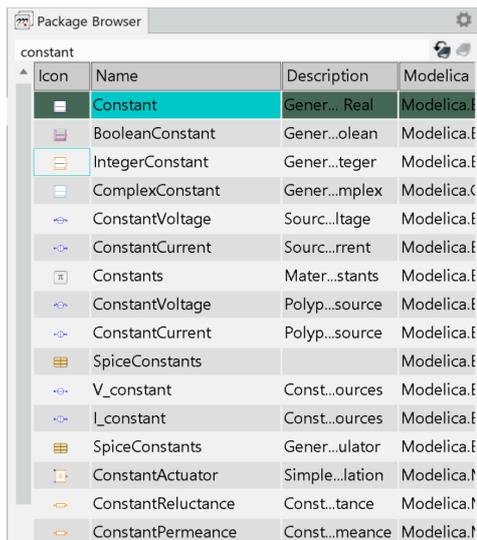
Cliquez sur le modèle logique Servomécanisme à droite et renommez le connecteur en outputPin3 :



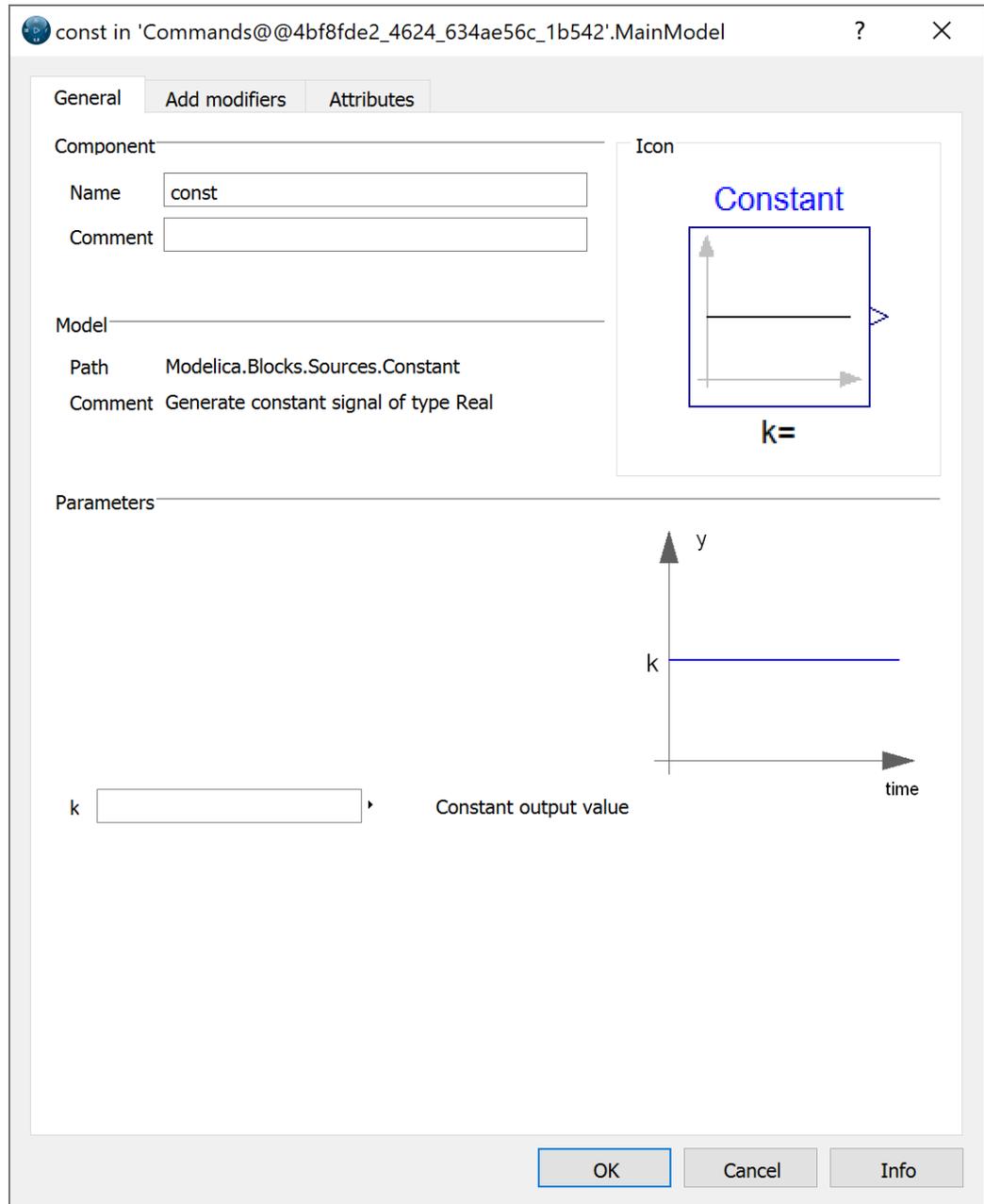
Double-Cliquez sur MainModel en bas du modèle logique « Commands » :



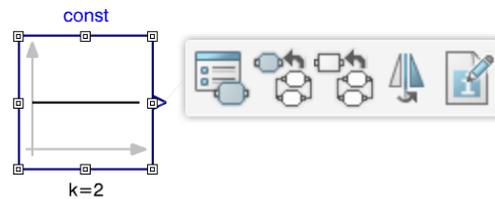
Placez un bloc Modelica « Constant » dans le modèle :



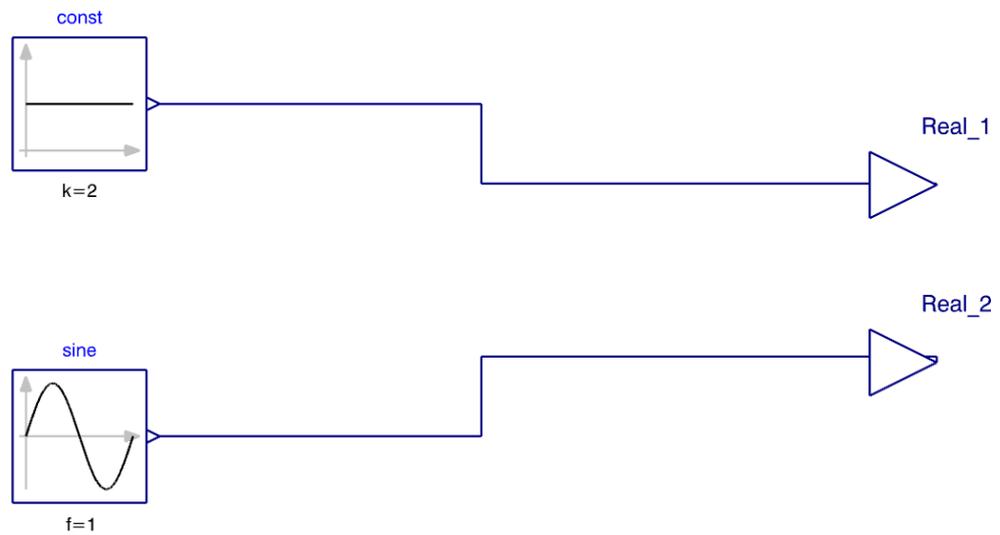
Attribuez la valeur 2 à la constante dans le champs « Constant Value » :



Le résultat est le suivant :



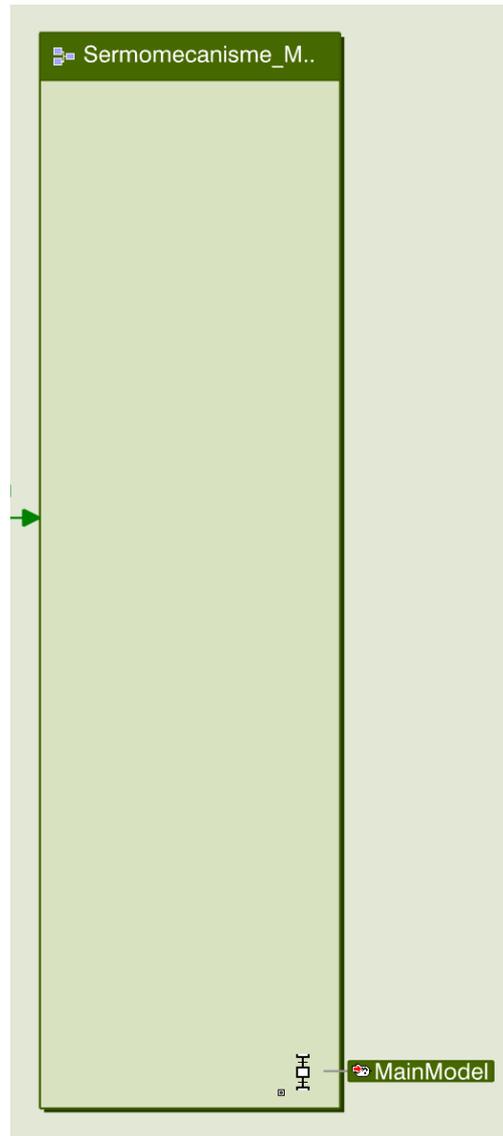
Reliez les deux blocs aux sorties tel qu'indiqué ci-dessous :



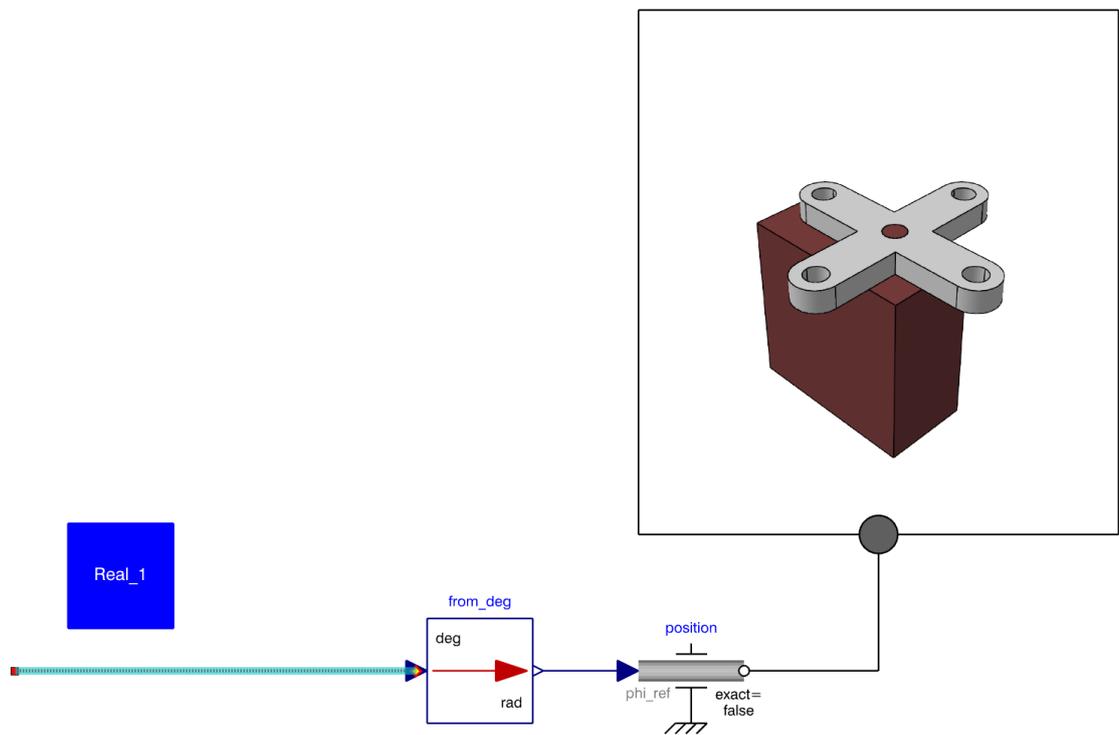
Sortez de l'App :



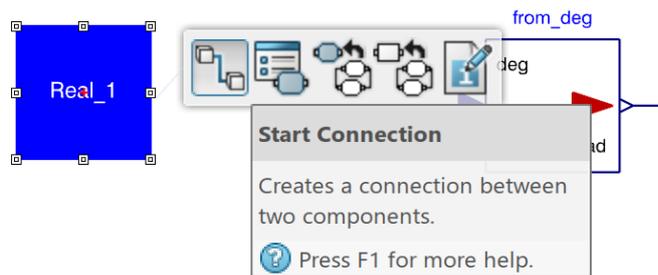
Double-cliquez sur « mainModel » dans le modèle logique « Servomecanisme » :



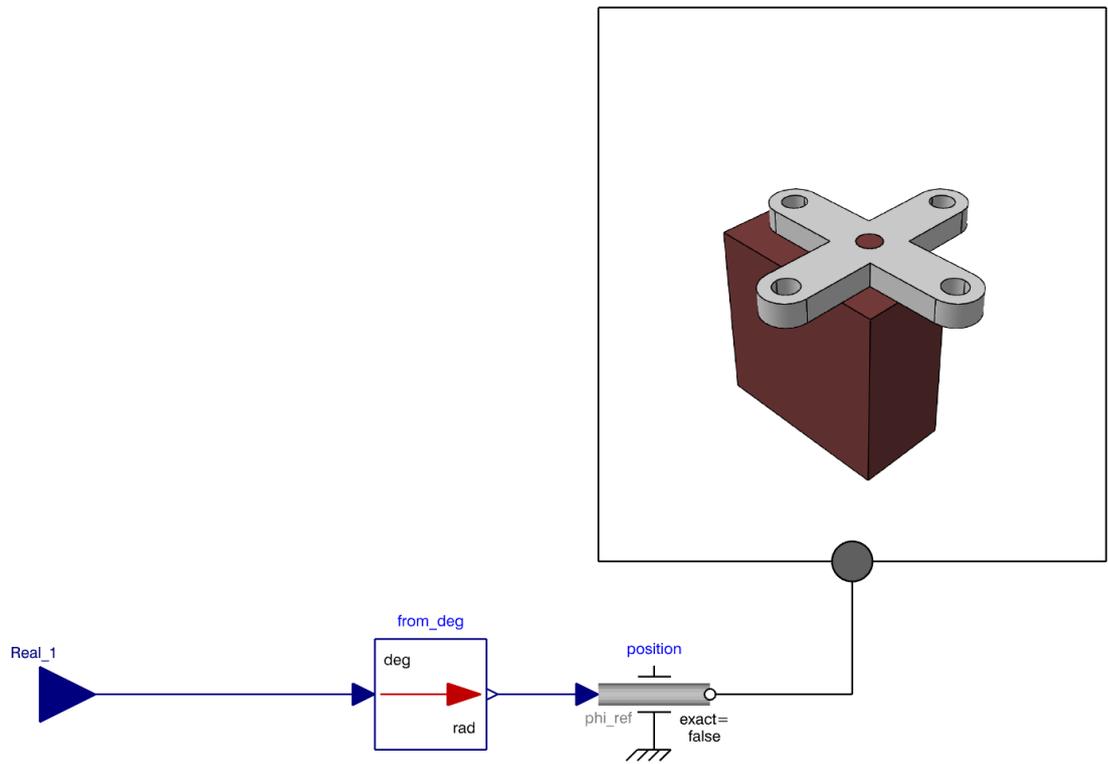
Supprimez l'éventuelle connexion résiduelle en entrée du blo « from\_deg » :



Placez le connecteur d'entrée « Real\_1 » au niveau des autres blocs puis cliquez sur le bloc « Real\_1 » puis sur « Start Connection » :



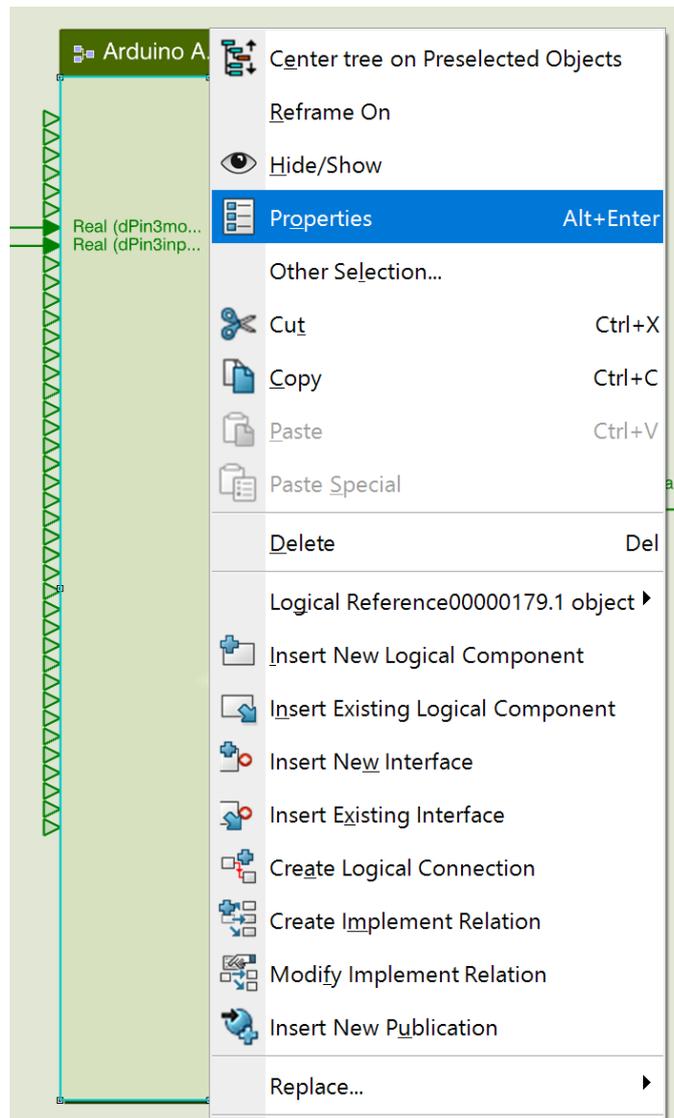
Reliez l'entrée « Real\_1 » au bloc « from\_deg » :



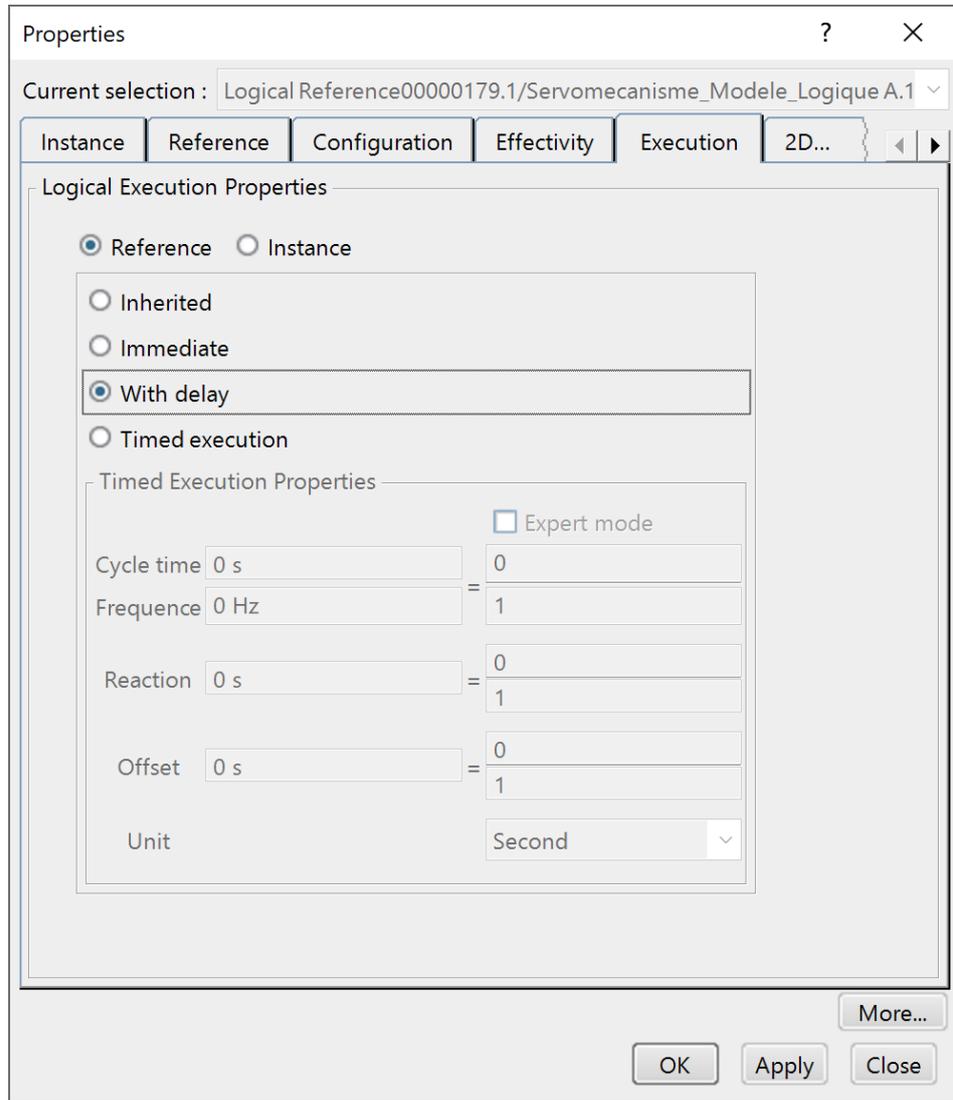
Cliquez sur « Exit App » :



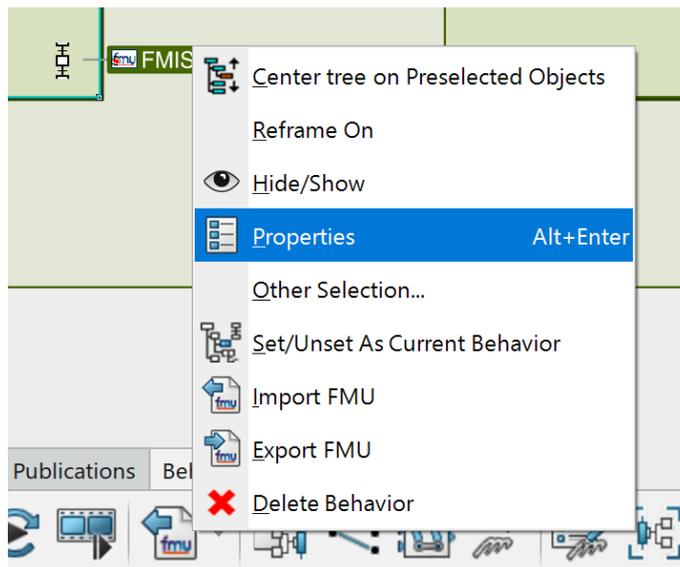
Cliquez sur le modèle logique « Arduino » avec le bouton droit puis sur « Properties » :



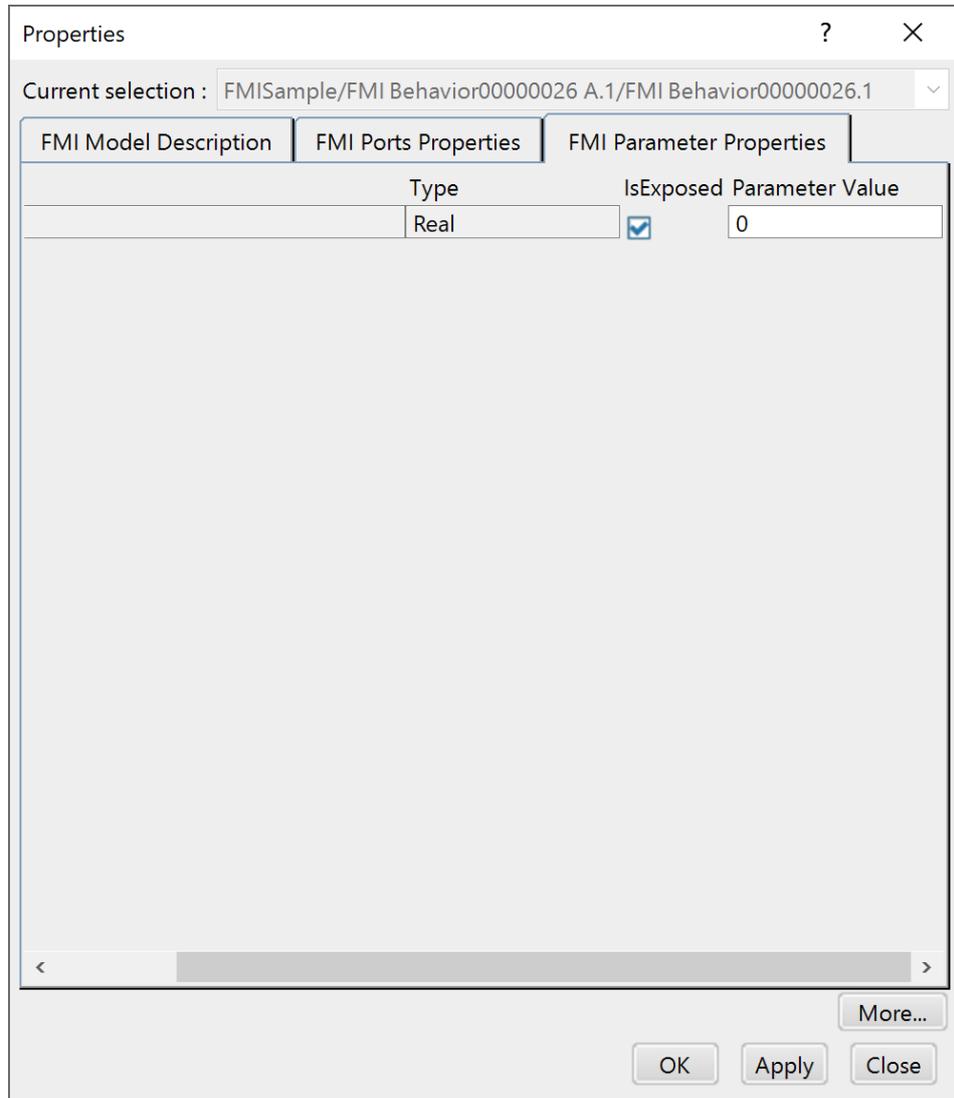
Réglez les paramètres d'exécution ainsi :



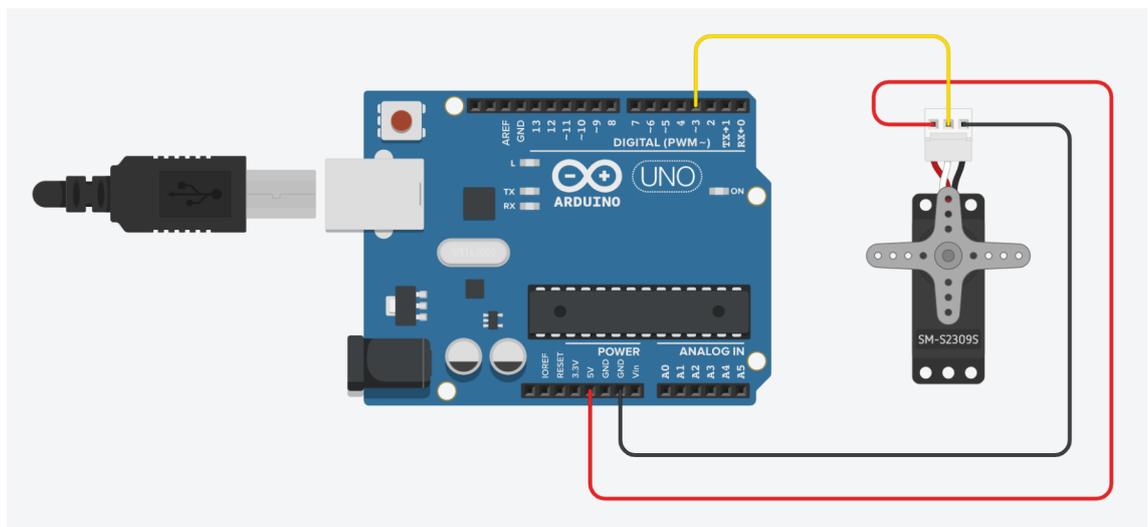
Cliquez sur le « mainModel » du modèle logique « Arduino » avec le bouton droit et cliquez sur « Properties » :



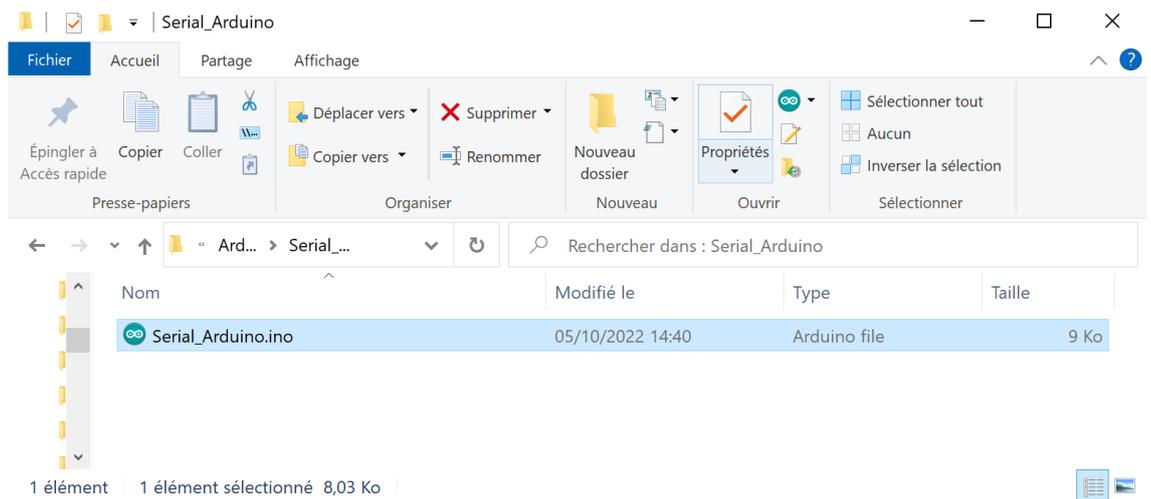
Réglez les paramètres « FMI Parameter Properties » ainsi :



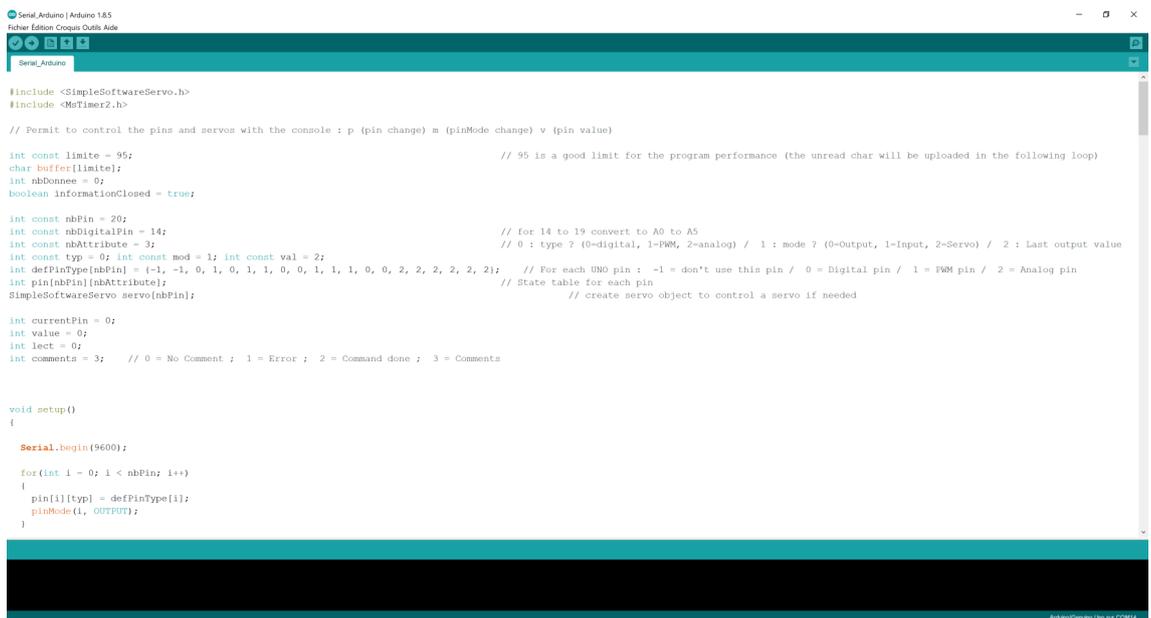
Connectez physiquement la carte Arduino à un servomécanisme selon le schéma de branchement ci-dessous, en veillant à relier la broche 3 au fil de commande du servomécanisme :



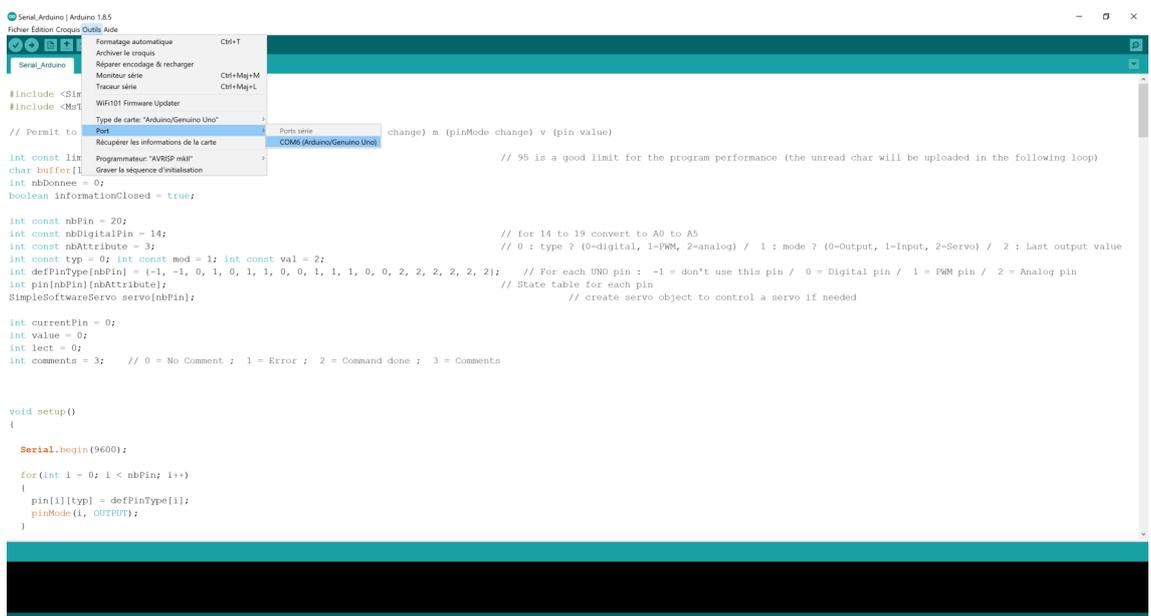
Double-cliquez sur le porgramme « Serial\_Arduino » fourni pour lancer l'IDE Arduino :



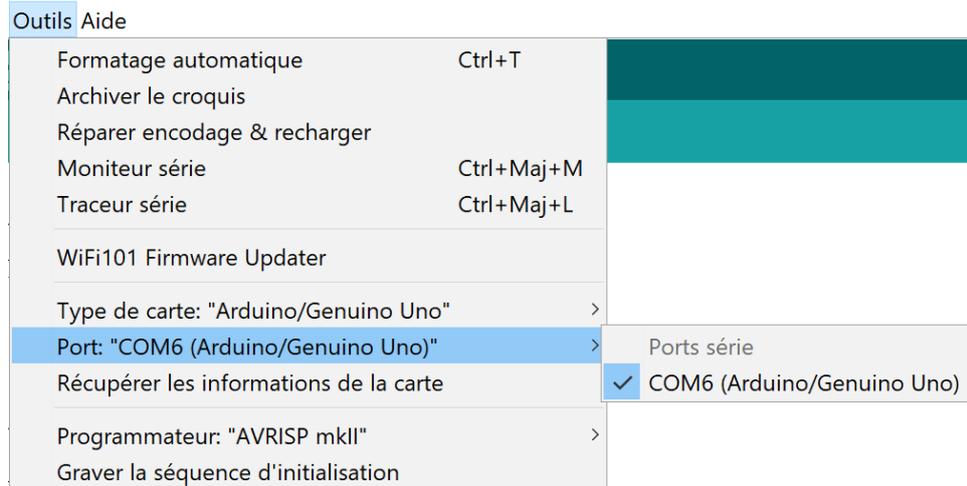
Le programme, ou sketch, suivant apparait :



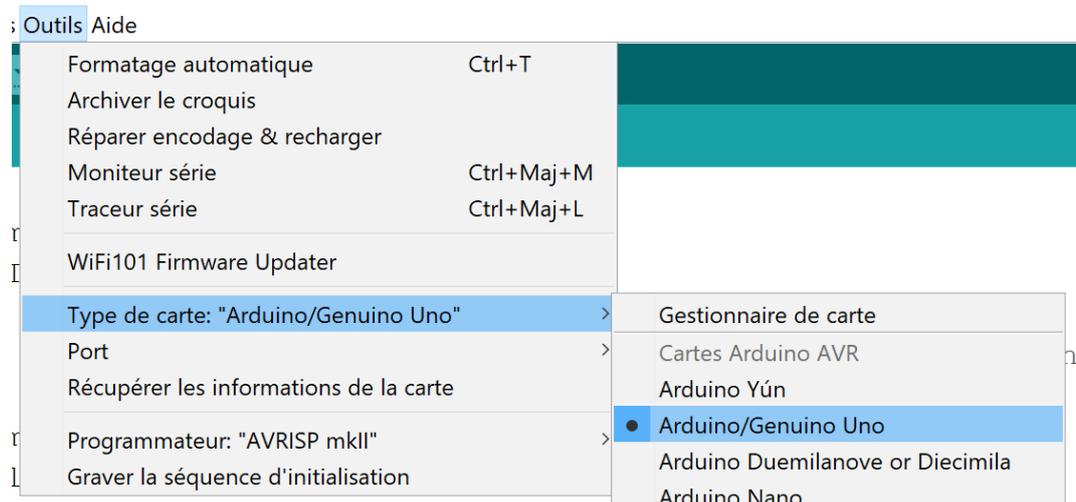
Branchez le cable USB de la carte Arduino au PC, puis cliquez sur « Outils/Port » et cochez le port COM qui indique la présence d'une carte Arduino, dans le cas présent c'est le COM6 :



Le port doit être coché ainsi :



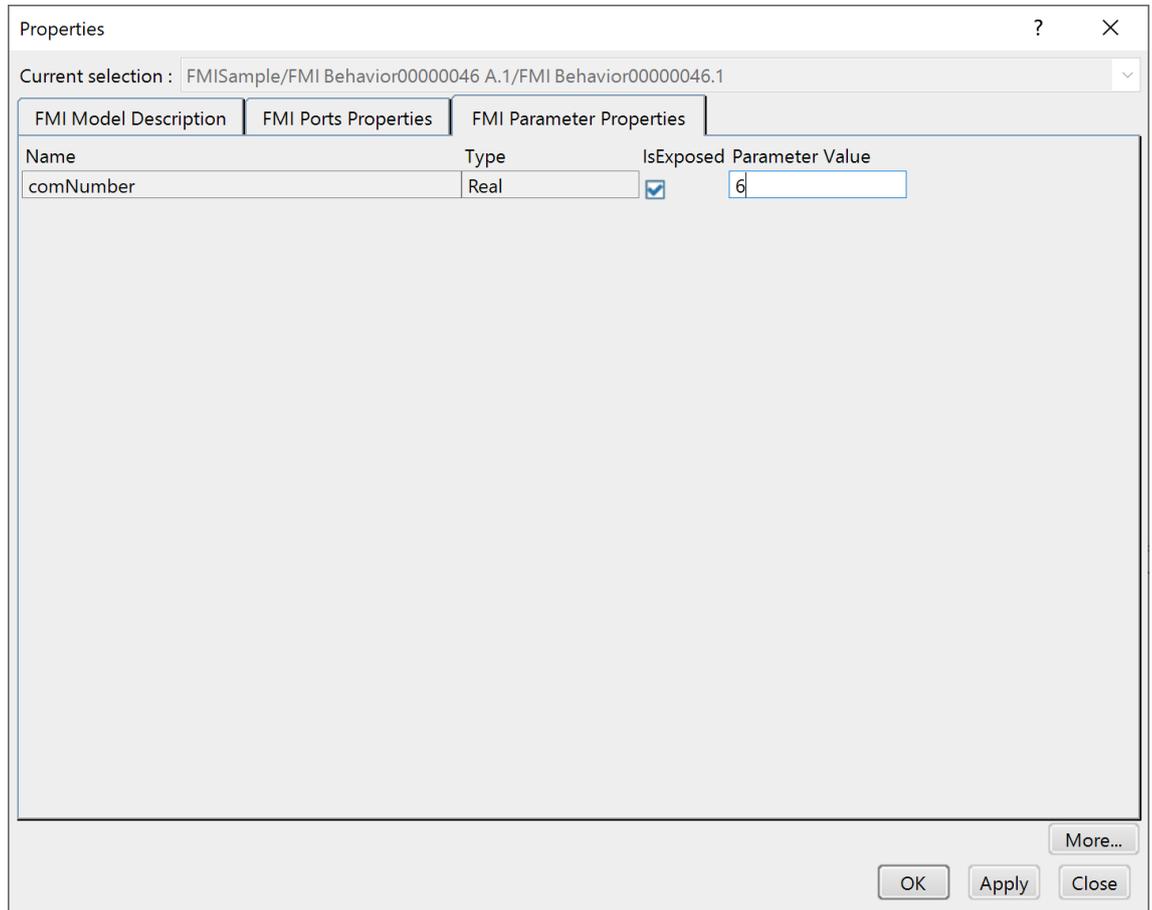
Cliquez ensuite sur « Outils/Type de carte » puis sélectionnez le modèle utilisé, dans notre cas, il s'agit d'une carte « Arduino/Genuino Uno » :



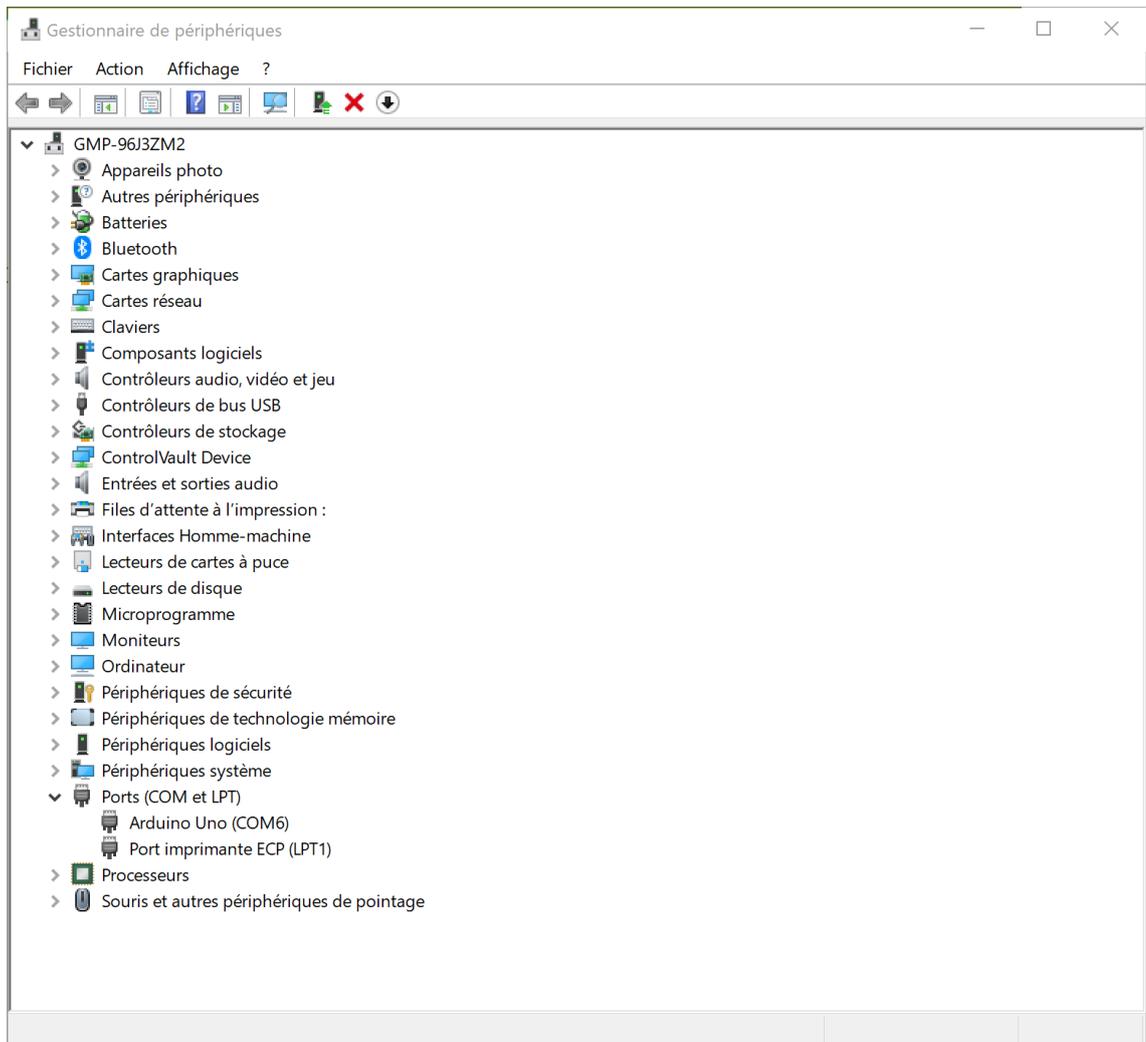
Cliquez sur téléverser le code : la flèche en haut à gauche de l'IDE :



Retournez dans la 3D Experience et changez la valeur du « FMI Parameter Properties / Parameter Value » des « Properties » du « mainModel » du modèle logique « Arduino » à celle du port utilisé COM utilisé par le PC pour communiquer avec la carte Arduino, dans notre exemple c'était le « 6 » :



NB : Pour être certain de bien renseigner le port COM, vous pouvez aussi l'identifier dans le « Gestionnaire de périphériques » de Windows accessible en cliquant avec le bouton droit de la souris sur l'icône Windows située en bas à gauche de votre écran. La carte Arduino apparaît dans les « Ports COM et LPT » et son numéro de port COM est rappelé :



Enfin, retournez dans la 3D Experience puis cliquez sur « Simulate » dans la fenêtre principale. Vous devriez observer des mouvements simultanés entre le servomécanisme relié à la carte Arduino et le jumeau numérique du servomécanisme dans la 3D Experience.

Dans le cas contraire, vérifiez que toutes les étapes précédentes ont bien été exécutées. Vérifiez également le fonctionnement du servomécanisme en utilisant dans l'IDE Arduino : « Fichiers/Exemples/Servo/Knob » puis en téléversant ce code dans la carte ; Si le servomécanisme ne bouge toujours pas, changez le, ou changez les fils électriques, ou changez la carte.

# Glossaire

|   |  |
|---|--|
| <b>Effet joule</b>  | <p>L'effet joule est la transformation de l'énergie électrique reçue par la résistance en énergie thermique. La résistance parcourue par un courant se met alors à chauffer. Ce phénomène peut être gênant et non souhaité dans certains cas mais il peut être mis à profit pour produire de la chaleur. On trouve ainsi des résistances dans les fours électriques, les radiateurs électriques, les sèche cheveux, les fers à repasser, etc.</p>  |
| <b>Fabrication additive</b>   | <p>La fabrication additive consiste à ajouter de la matière (impression 3D, injection plastique, roto-moulage, etc.). En ce qui concerne l'impression 3D, l'ajout de matière est très souvent réalisée en couche par couche d'épaisseurs constantes.</p>   |
| <b>Fabrication soustractive</b>   | <p>La fabrication soustractive consiste à retirer de la matière comme par exemple avec le tournage et le fraisage.</p>   |
| <b>Lumen</b>  | <p>En physique, un lumen est la quantité de lumière interceptée par 1 m<sup>2</sup> de surface interne d'une sphère creuse de 1 m de rayon, au centre de laquelle on a placé une bougie. Pour définir l'éclairement d'un objet, on utilise le lux qui correspond à un flux d'un lumen tombant sur une surface de 1 m<sup>2</sup> de l'objet.</p>   |
| <b>Photorésistance</b><br><i>≈ résistance photogénique, cellule photoconductrice ou cellule photoélectrique</i> | <p>Une photorésistance est un composant électronique dont la résistance varie en fonction de la quantité de lumière incidente : plus elle est éclairée, plus sa résistance baisse et réciproquement.</p>   |
| <b>Platine Labdec</b><br><i>≈ Breadboard</i>  | <p>La « breadboard » ou « platine Labdec » simplifie le montage de composants. Cette platine est très utilisée pour le prototypage électronique afin d'éviter d'avoir recours à des cartes électroniques et des soudures.</p> <p>Tous les connecteurs positifs et négatifs latéraux (entourés des lignes rouges et bleues) sont reliés entre eux. Tous les connecteurs des lignes A à E et des lignes F à J sont reliés entre eux (mais il n'y aucune liaison entre A-E et F-J).</p> <p>Le schéma ci-dessous illustre les liaisons internes de la platine.</p> |
| <b>Résistance</b>   | <p>Une résistance est un composant électronique ou électrique dont la principale caractéristique est d'opposer une plus ou moins grande résistance (<i>mesurée en ohms</i>) à la circulation du courant électrique. La résistance électronique est l'un des composants primordiale dans le domaine de l'électricité.</p>   |
|   | <p>Rouge, vert, bleu, abrégé en RVB ou en RGB (de l'anglais « red, green, blue ») est un système de codage informatique des couleurs,</p>  |

|                            |   |
|----------------------------|---|
| <b>RVB</b><br><i>≈ RGB</i> | <p>le plus proche du matériel.</p> <p>Les écrans d'ordinateurs reconstituent une couleur par synthèse additive à partir de trois couleurs primaires : rouge, vert et bleu, formant sur l'écran une mosaïque trop petite pour être aperçue.</p> <p>Le codage RVB indique une valeur pour chacune de ces couleurs primaires.</p>  |
| <b>Tilt</b>                | <p>Un interrupteur Tilt permet de détecter l'orientation ou l'inclinaison d'un système. Il est souvent utilisé pour indiquer si un système (comme un véhicule agricole) dépasse sa plage d'inclinaison de fonctionnement ou pour détecter l'orientation d'un écran et, ainsi, modifier sa mise en page. Il ne donne pas autant d'information qu'un accéléromètre mais est plus robuste et ne nécessite pas de programme particulier pour être traité.</p> |