

République Algérienne populaire et Démocratique  
Ministère de l'enseignement supérieur et de la recherche scientifique  
Université de Abderrahmane mira de Bejaia



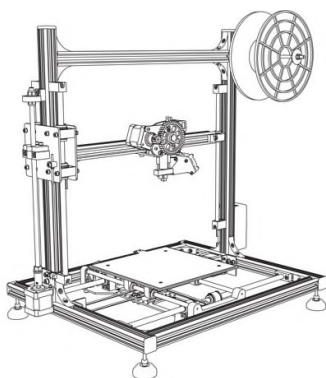
**Faculté de Technologie**  
**Département de Génie Electrique**

**En vue de l'obtention du diplôme de master en Electromécanique**  
**Option Electromécanique**

---

*Etude et réalisation d'une machine outils à trois axes*

---



Elaboré par  
**MR Benamor Abd elkrim**

**Mr Soualmi Badaoui**

Encadré  
**H.Ouatah**

**2018/2019**



## **Remercîment :**

On remercie tous d'abord notre dieu qui nous a permis de réaliser ce modeste travail, de nous avoir donné la patience et le courage nécessaire pour continuer ce travail après les contraintes rencontré

Ensuite on tient à remercier :

- second lieu notre promoteur Mr H.Ouateh, qui nous aidé.
- les membres des jurys pour lire et examiner notre travail.
- tous les enseignants qui nous ont aidés dans notre cursus.
- nos parents qui ont pris notre réussite pour priorité.
- Les personnes qui nous ont aidés à réussir dans ce projet

## Dédicace :

Je dédie ce modeste travail :

- **Mes deux parents qui se sont sacrifié pour ma réussite.**
- **Mes frères et ma petite sœur qui ont contribué à ma réussite.**
- **Mes amis sans exception.**
- **Toute personne ayant contribué a la réussite de ce projet de prêt ou de loin.**
- **A toute la section électromécanique (promotion 2018/2019).**

# **Sommaire**

---

## Sommaire

<b>Introduction générale</b> .....	1
<b>Chapitre I : Généralités sur les machines à commande numérique</b>	
<b>I.1. Introduction</b> .....	2
<b>I.2. Définition</b> .....	3
<b>I.3. Historique</b> .....	3
<b>I.4. Commande numérique par calculateur</b> .....	4
<b>I.5. Principe de fonctionnement d'une machine à commande numérique</b> .....	4
<b>I.6. Structure d'une machine à commande numérique</b> .....	5
<b>I.7. Axes normalisés</b> .....	6
<b>I.7.1. Axes principaux linéaires</b> .....	6
<b>I.7.2. Axes circulaires primaires</b> .....	8
<b>I.7.3. Axes additionnels</b> .....	8
<b>I.8. Classification des MOCN</b> .....	8
<b>I.8.1. MOCN classées selon la nature de déplacement</b> .....	8
<b>I.8.2. MOCN classées selon le mode de fonctionnement</b> .....	9
<b>I.8.3. Classification des MOCN selon le nombre d'axe</b> .....	9
<b>I.8.4. Classification des MOCN selon le mode d'usage</b> .....	11
<b>I.9. Commande des MOCN</b> .....	12
<b>I.9.1. Conception 3D</b> .....	12
<b>I.9.2. Génération des chemins de l'outil, création du G-code</b> .....	12
<b>I.9.3. Interprétation du G-code et transformation en mouvement sur la machine</b> .....	13
<b>I.10. Domaines d'utilisation des MOCN</b> .....	14
<b>I.11. Conception assistée par ordinateur (CAO)</b> .....	14
<b>I.11.1. Définition</b> .....	14
<b>I.11.2. Les outils de la CAO (logiciels)</b> .....	14
<b>I.12. Fabrication assistée par ordinateur</b> .....	14
<b>I.12.1. Définition</b> .....	14
<b>I.12.2. Les outils logiciels de la FAO</b> .....	15
<b>I.13. Conclusion</b> .....	15

**Chapitre II : vue générale sur la partie électrique**

<b>II.1. Introduction</b> .....	16
<b>II.2. Eléments de la partie commande</b> .....	17
<b>II.2.1. Contrôleur</b> .....	17
<b>II.2.2. Arduino</b> .....	17
<b>II.2.2.1. Caractéristiques techniques de l'Arduino</b> .....	17
<b>II.1.2.2. Détails techniques</b> .....	18
<b>II.1.2.3. Mémoire</b> .....	19
<b>II.2.2.4. Entrées et sorties</b> .....	19
<b>II.2.3. CNC-Shield</b> .....	20
<b>II.2.3.1. Description</b> .....	20
<b>II.2.3.2. Caractéristiques techniques</b> .....	20
<b>II.2.4. Pilotes A4988 et DRV8825</b> .....	21
<b>II.2.4.1. Descriptions</b> .....	21
<b>II.2.4.2. Rôle et réglage des pilotes moteurs pas à pas</b> .....	21
<b>II.2.5. Fins de course</b> .....	24
<b>II.3. Eléments de la partie opérative « Moteurs pas à pas »</b> .....	24
<b>II.3.1. Différents types de moteurs pas à pas</b> .....	24
<b>II.3.1.1. Moteurs à aimant permanent</b> .....	25
<b>II.3.1.2. Moteurs à réluctance variable</b> .....	25
<b>II.3.1.3. Moteurs hybrides</b> .....	25
<b>II.3.2. Moteurs Nema 17</b> .....	26
<b>II.3. Principe de fonctionnement du moteur pas à pas et de sa commande</b> .....	27
<b>II.3.1. Moteur à réluctance variable et moteur hybride</b> .....	27
<b>II.3.2. Excitation des phases</b> .....	29
<b>II.4. Pont en H à transistor bipolaire</b> .....	31
<b>II.4. Conclusion</b> .....	32

### Chapitre III : conception et réalisation de la partie mécanique

<b>III.1. Introduction</b> .....	33
<b>III.2. Caractéristiques visées de la machine</b> .....	33
<b>III.2.1. Zone d'opération</b> .....	33
<b>III.2.2. Dimensions</b> .....	34
<b>III.2.3. Matériaux à utiliser</b> .....	34
<b>III.3. Réalisation du châssis de la machine</b> .....	34
<b>III.4. Réalisation des mécanismes de mouvement sur les trois axes</b> .....	36

<b>III.4.1. Axe X</b> .....	37
<b>III.4.2. Axe Y</b> .....	38
<b>III.4.3. Axe Z</b> .....	38
<b>III.5. Moteurs d'entraînement</b> .....	39
<b>III.6. Mécanisme de transmission de mouvement</b> .....	40
<b>III.7. Système de guidage</b> .....	40
<b>III.8. Assemblage de la machine</b> .....	40
<b>III.8. Conclusion</b> .....	41

#### **Chapitre IV : programmation et commande de la machine**

<b>IV.1. Introduction</b> .....	41
<b>IV.2. Branchement des moteurs</b> .....	41
<b>IV.3. Arduino IDE</b> .....	42
<b>IV.4. Concept du programme</b> .....	46
<b>IV.5 Sous-programme de la remise aux points d'origine des moteurs</b> .....	47
<b>IV.6. Sous-programme de la commande manuelle des moteurs</b> .....	48
<b>IV.7. Inkscape et l'extension G-codeTools</b> .....	49
<b>IV.8. G-code Sender</b> .....	50
<b>IV.9. Conclusion</b> .....	51
<b>Conclusion générale</b> .....	52



# **Introduction Générale**

## **Introduction :**

L'évolution de la technologie mondiale a pris des progrès remarquable dans cette décennie, ce qui donne naissance à une nouvelle architecture de création soit dans le domaine du génie électrique, médicinale, informatique ou dans le domaine de la construction mécanique, ce qui implique une précision dans d'exécution qui dépasse la capacité humaine.

Alors la machine-outil à commande numérique et grâce à sa vitesse d'exécution et sa précision. Elle est aujourd'hui le moyen de production le plus important des pièces mécaniques, électronique...etc. De par l'avancée des techniques, de cette dernière a subi des modifications et des améliorations, et le couple utile de la machine-outil et s'est adapté aux exigences de la productivité moderne et à la nécessité visé par les sociétés.

Nous avons réalisé une machine outils a trois axes dans notre projet de fin d'étude pour l'obtention du diplôme de Master, le but essentiel de notre projet est de développer un prototype qui peut être utilisé soit comme imprimante 2D, 3D et fraiseuse à la fois tout et voir les problèmes et les contrainte qui peuvent subvenir dans cette réalisation.

Le projet est devisé en quatre chapitres :

- Généralités sur les machines à commandes numériques..
- Vue Générale sur la partie électrique.
- Conception et réalisation de la partie mécanique.
- Programmation et commande de la machine

# Chapitre I

## **I.1. Introduction**

Les machines-outils à commande numérique (MOCN) sont devenues le premier recours des géants de la fabrication mécanique et même électronique, elles permettent la fabrication et la production de formes complexes à usiner manuellement (courbées ou arrondies.).

Dans ce chapitre, nous faisons un aperçu général sur les MOCN, en présentant leur historique, leur structure, leur technologie et leurs domaines d'application.

## **I.2. Définition**

La MOCN est une machine totalement ou partiellement automatique à laquelle les ordres sont communiqués grâce à des codes qui sont portés sur un support matériel (disquette, cassette, USB, ...). Lorsque la machine-outil est équipée d'une commande numérique capable de réaliser les calculs des coordonnées des points définissant une trajectoire (interpolation), on dit qu'elle est à calculateur. Elle est appelée CNC (Commande Numérique par Calculateur). La plupart des MOCN sont des CNC [1].

## **I.3. Historique**

L'évolution de l'industrie dans notre époque a évoqué une nouvelle architecture dans la construction mécanique (avions, voitures ...etc.), ce qui a engendré la création de pièces très compliquées à usiner manuellement

Les travaux menés par Falcon et Jacquard à la fin du XVIII<sup>ème</sup> siècle ont montré qu'il était possible de commander les mouvements d'une machine à partir d'informations transmises par un carton perforé. Leur métier à tisser de 1805 fut le premier équipement à être doté de cette technique et, de ce point de vue, il peut être considéré comme l'ancêtre de la commande numérique. Il faut cependant rattacher l'exploitation industrielle de la CN au développement de l'électronique.

En 1947, dans l'État du Michigan, John Parsons fabrique pour le compte de l'US Air Force des pales d'hélicoptère par reproduction. Pour façonner ses gabarits, il utilise une méthode consistant à percer plusieurs centaines de trous faiblement espacés de manière à approcher le profil théorique. Mais, lorsque l'US Air Force confie à ce même Parsons la réalisation de pièces de formes encore plus complexes pour ses futurs avions, celui-ci réalise que sa méthode est trop approximative et que seul un usinage continu en 3 dimensions sera en mesure de donner satisfaction. [13]

Au printemps de 1949, il confie alors au Massachusetts Institute of Technology (MIT) le soin de développer des asservissements capables de piloter une machine qui recevra des instructions intermittentes à partir d'un lecteur de cartes.

Cette machine, une *fraiseuse prototype Cincinnati à broche verticale*, conçue pour exécuter des déplacements simultanés suivant 3 axes, est officiellement présentée en septembre 1952 dans le Servomechanisms Laboratory du MIT. L'information mathématique étant la base du concept, on lui donne le nom de *numerical control*.

Il faut encore attendre quelques années de vastes fonds de l'US Air Force et l'appui des chercheurs du MIT pour rendre la première Machine-outil à commande numérique (MOCN) réellement opérationnelle.

General Electric sera historiquement le premier fabricant industriel de machines à commandes numériques, rejoint en 1962 par son voisin japonais, FANUC. En 1964, les États-Unis comptaient 35 000 machines à commande numérique [2].

#### **I.4. Commande numérique par ordinateur**

C'est en 1942 aux États-Unis qu'on a commencé l'exploitation de CN, afin de produire des pompes à injection pour moteurs d'avions. Plus exactement des cames, dont le profil complexe était irréalisable avec des outils à commande manuelle.

La commande numérique a permis de générer des mouvements d'axes. En se basant sur la position angulaire du rotor d'un moteur pas à pas. Ces instructions viennent d'un programme pièce, ce programme divise la pièce en une infinité de points pour les transformer en coordonnées. Le processeur de la commande numérique va alors générer une consigne afin qu'on puisse commander les moteurs d'axes. Elle aura également pour tâche, lors du déplacement des axes, de vérifier la position de ces derniers et dans certain cas, la vitesse de déplacement.

Les commandes numériques actuelles sont capables d'effectuer des mouvements en combinant simultanément les positions sur les axes X, Y et Z, Elles s'appellent commande numérique par ordinateur. De plus, elles sont capables de gérer plusieurs axes simultanément. On parle alors d'interpolation. Cette dernière peut être linéaire ou circulaire selon la consigne générée par le microcontrôleur de la partie commande. Le rôle de ce processeur est d'interpréter un code machine que l'on appelle un code G, puis de générer les signaux de commande des axes et de la broche [3].

## I.5. Principe de fonctionnement d'une machine à commande numérique

Les programmes d'usinage sont réalisés à partir d'une origine appelée « origine programme » (OP) positionnée par le programmeur. Le programme commande les déplacements relatifs entre le brut et les outils dans le but de réaliser l'usinage de la pièce finale. Ces déplacements sont réalisés dans un repère orthonormé normalisé (O, X, Y, Z) basé sur la structure de la machine.

L'axe (Z) de ce repère est un axe confondu avec celui de la broche de la machine (axe de rotation de la fraise en fraisage, axe de rotation de la pièce en tournage). Le sens positif de cet axe est donné par le sens d'éloignement de l'outil par rapport à la pièce. La détermination de l'axe (X) entre les 2 axes restants se fait en identifiant celui qui permet le plus grand déplacement. Le sens positif de (X) est déterminé par le sens logique d'éloignement de l'outil par rapport à la pièce. L'axe(Y) est déterminé à partir de (X) et (Z) grâce à la règle du trièdre direct [2].

## I.6. Structure d'une machine à commande numérique

Ce type de machine se constitue de deux parties complémentaires :

- La partie opérative.
- La partie commande.

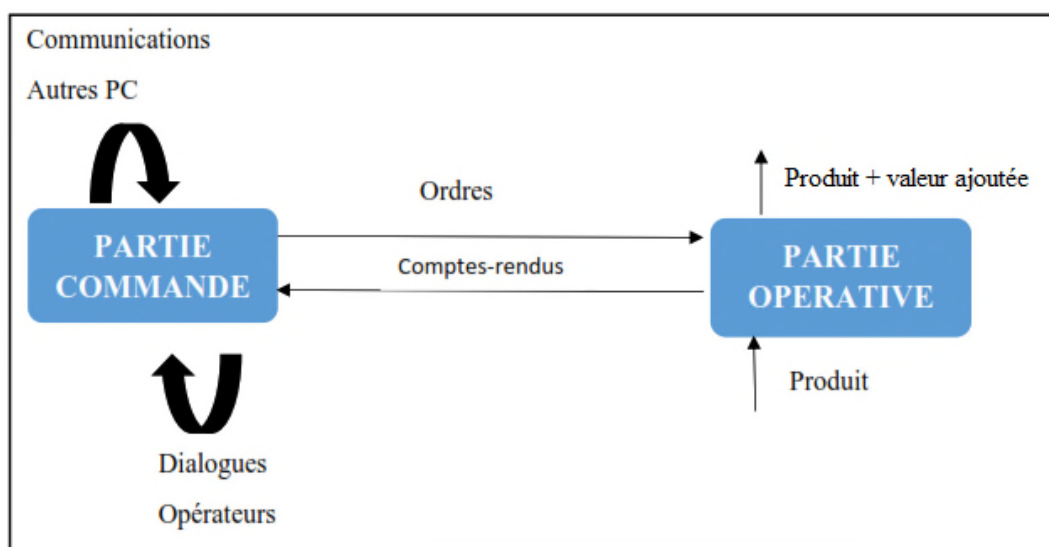


Figure I.1: Structure d'une MOCN

La partie opérative comporte les axes de déplacement et la tête et l'outil qui sert à usiner la pièce. La partie commande permet de piloter la partie opérative. Elle est composée d'un calculateur et d'éléments électroniques capables de piloter les moteurs.

Des ordres vont être générés vers la commande par le biais d'un code machine ou par action manuelle de l'opérateur. La commande va traiter ces informations et générer des consignes afin d'obtenir les déplacements voulus par le biais des moteurs d'axes. Des contrôles de vitesse et de position seront alors effectués de manière continue par la machine.

### **I.7. Axes normalisés**

Un degré de liberté d'un organe de machine est appelé *AXE* si l'actionneur est asservi en vitesse et position et s'il peut être synchronisé avec un autre degré de liberté pour obtenir un déplacement qui n'est pas parallèle à une direction principale du système de coordonnées.

Il permet d'obtenir une position, par une instruction numérique, à la résolution du moyen de mesure près. Un axe est constitué de :

- ✓ Un chariot mobile sur glissières,
- ✓ Un système de transmission vis-écrou,
- ✓ Un moteur ,
- ✓ Un dispositif de mesure de vitesse,
- ✓ Un dispositif de mesure de position.

« *Chaque axe de déplacement est asservi en position (Figure I.2), dont le principe consiste à mesurer continuellement la position réelle et à la comparer avec la grandeur d'entrée ou position de consigne, que délivre la CN pour atteindre la nouvelle position programmée. Dès que l'écart entre les deux mesures s'annule, le mobile s'arrête* ».

Le déplacement de la table ou de l'outil d'un point à un autre implique la connaissance :

- de l'axe (X,Y,Z,...) sur lequel le déplacement doit s'effectuer;
- des coordonnées du point à atteindre;
- Le sens de déplacement (+ ou -);
- de la vitesse déplacement (d'avance) de la table ou de l'outil.

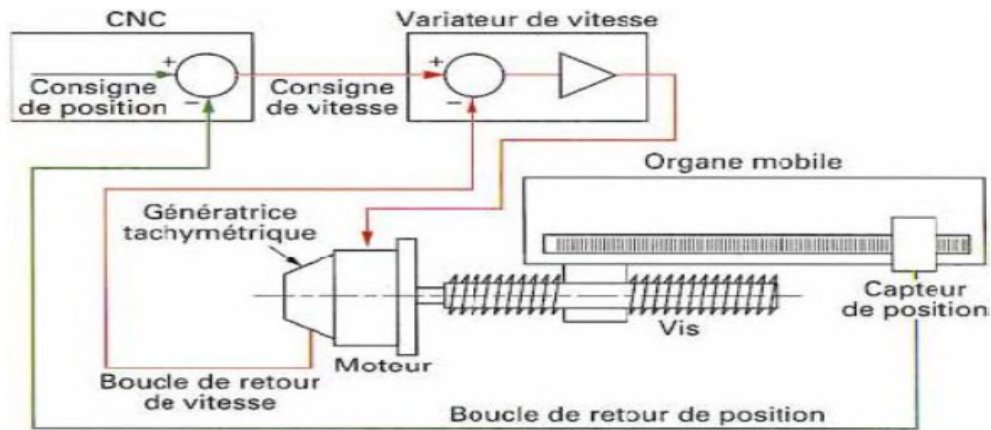


Figure I.2 : Asservissement d'un axe en boucle fermée

### I.7.1. Axes principaux linéaires

Ils sont au nombre de trois ayant pour symbolisation [4] :

- X Y Z quand l'outil se déplace,
- X' Y' Z' quand la pièce se déplace.

#### a) Propriétés

Axe	Particularités
Z ou Z'	Dans la majorité des cas, l'axe Z ou Z' est l'axe de la broche. Le sens positif de ces axes correspond à un accroissement des distances relatives porte pièce-outil.
X ou X'	Il est perpendiculaire à l'axe Z ou Z' et correspond à l'axe ayant le plus grand déplacement. <ul style="list-style-type: none"> <li>- Sur une fraiseuse : c'est généralement un axe // à la glissière longitudinale,</li> <li>- Sur un tour : cet axe est radial.</li> </ul>
Y ou Y'	L'axe Y ou Y' forme avec les axes (X et Z) ou (X' et Z') un trièdre direct.

Les figure I.3 et I.4 illustrent respectivement les axes linéaires d'une à 2 axes et une machine à 3 axes.

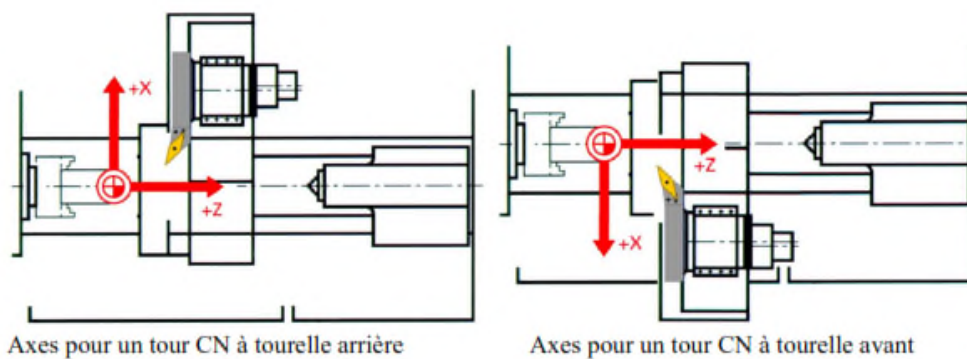


Figure I.3 : Axes linéaires d'une machine à 2 axes « tour »



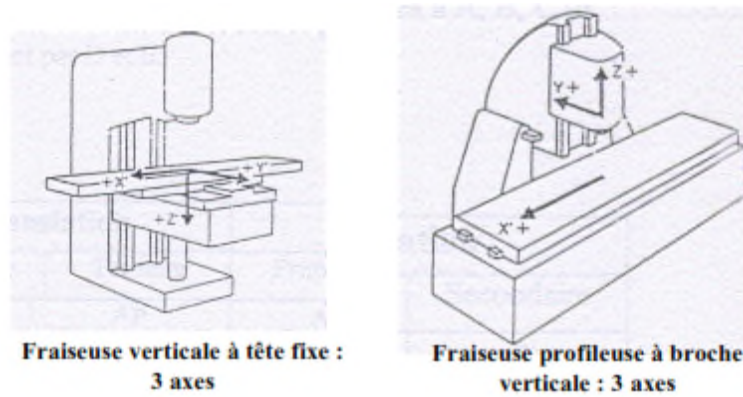


Figure I.4 : Axes linéaires normalisés d'une machine à 3 axes « fraiseuse »

### I.7.2. Axes circulaires primaires

A, B, C désignent les axes circulaires, ils tournent respectivement autour des axes X, Y, Z. Le sens positif est donné en positionnant la main droite tel que le pouce placé dans le sens du mouvement de translation positif, les doigts repliés donnent le sens positif de rotation [4].

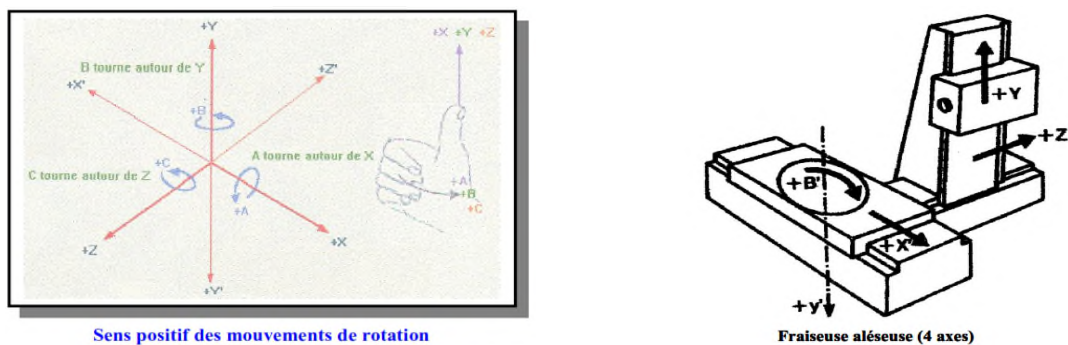


Figure I.5 : Axes circulaires primaires

### I.7.3. Axes additionnels

Si en plus des mouvements de translation rectiligne primaire X, Y et Z il existe des mouvements de translation secondaires parallèles à ceux-ci, ils seront désignés par U, V, W. la figure I.5 montre un exemple d'axes additionnels dans le cas d'un tour à tourelles [4].

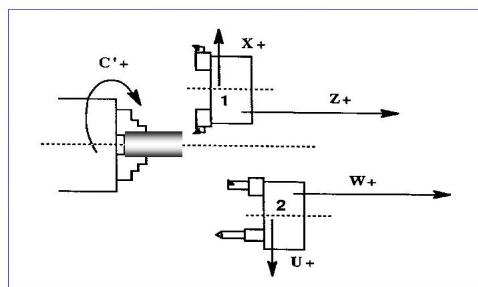


Figure I.6 : Exemple d'axes additionnels « Tour à 2 tourelles »

## I.8. Classification des MOCN

Le classement des machines est nécessaire car il aide au choix de machines, lors d'étude de gammes de fabrication. Les machines-outils à commande numérique (MOCN) sont classées selon la nature de déplacement, le mode de fonctionnement de la machine, le nombre d'axes de la machine ou le mode d'usinage.

### I.8.1. MOCN classées selon la nature de déplacement

Traditionnellement, on a classé les machines en fonction des formes de surfaces à réaliser : cylindriques / parallélépipédiques, tournage / fraisage. Cette classification est remise en cause, car la commande numérique et l'adaptation des structures de machine cassent le lien entre les deux couples [5].

### I.8.2. MOCN classées selon le mode de fonctionnement

#### a. Fonctionnement en boucle ouvert

En boucle ouverte, le système assure le déplacement du chariot mais ne le contrôle pas (Figure I.7).

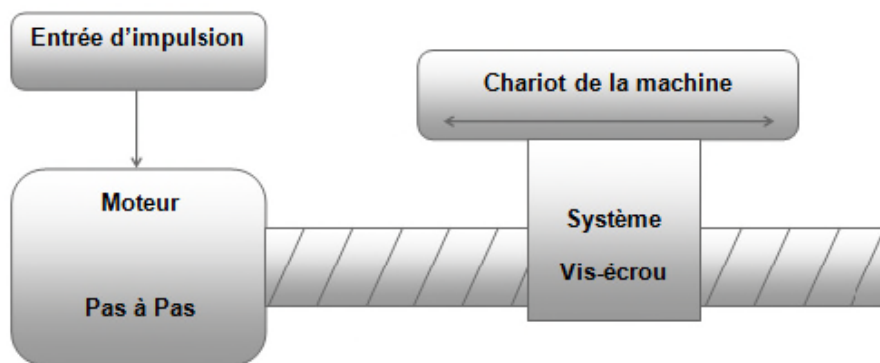


Figure I.7 : Fonctionnement des MOCN en boucle ouvert [6]

#### c. Fonctionnement en boucle fermé

Comme le montre la figure I.8, en boucle fermée le système contrôle le déplacement ou la position jusqu'à égalité des grandeurs entrée (E) dans le programme et celui mesuré (Gm).

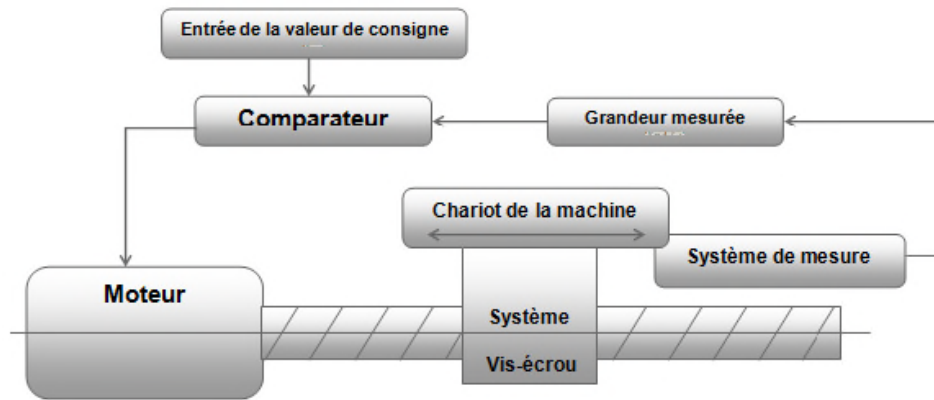


Figure I.8 : Commande des MOCN en boucle fermée [6]

### I.8.3. Classification des MOCN selon le nombre d'axe

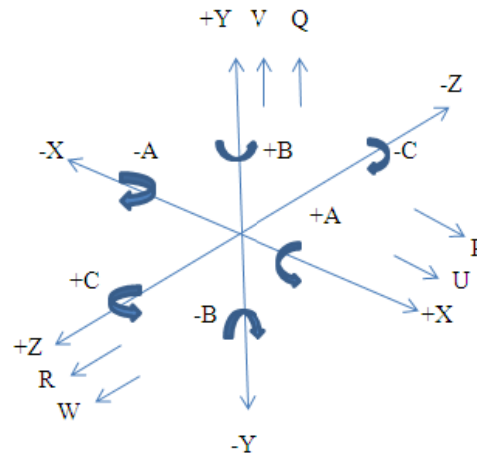
Les machines-outils peuvent être classées par le nombre de mouvements élémentaires qu'elles peuvent mettre en œuvre lors du déplacement de l'outil par rapport à la pièce (tableau I.1). La mise en œuvre simultanée de plusieurs outils entraîne l'augmentation du nombre d'axes. Cette classification ne permet pas d'associer directement un type de forme usinable à une classe de machine, car elle ne reflète pas la cinématique de l'outil.

Par exemple un tour à cinq axes ne permet pas de faire des pièces différentes par rapport à un tour à trois axes [5] ,[11].

Tableau I.1 : Classification des MOCN selon le nombre d'axe

Nbre d'axe	Mouvements	Désignation du type d'usinage et des opérations possibles
1	Z	Brochage, presse
2	X, Z	Tournage : toutes les formes obtenues sont le même axe des y métrie
3	X, Y, Z	Fraisage : surfacage, perçage, fraisage de poches, de rainures et de surfaces gauches. L'axe outil reste parallèle à une direction fixe par rapport à la pièce.
3	X, Z, C	Tournage, avec asservissement de la broche, permet le fraisage sur tour : tournage général, fraisage à l'outil tournant, perçage hors axe. La broche est asservie en position.
4	X, Y, Z, B	Fraisage : surfacage, perçage, fraisage de poches de rainures et de surfaces gauches. L'axe outil reste contenu dans un plan fixe par rapport à la pièce.
4	X, Y, Z, C	Fraisage (cf X, Y, Z, C)
4	2(X, Z)	Tournage à deux tourelles
5	2(X, Z), C	Tournage à deux tourelles et asservissement de la broche, (cf. X, Z, C)
5	X, Y, Z, A, C	Fraisage de formes gauches (cf. X, Y, Z, A, C)
5	X, Y, Z, B, C	Fraisage de formes gauches (cf. X, Y, Z, B, C)
5	X, Y, Z, A, B	Fraisage de formes gauches (cf. X, Y, Z, A, B)

Chaque mouvement de translation ou de rotation est donc représenté par un axe défini une lettre affectée de signe + ou -. La figure I.9 montre l'ensemble des axes qu'un DCN peut contrôler.



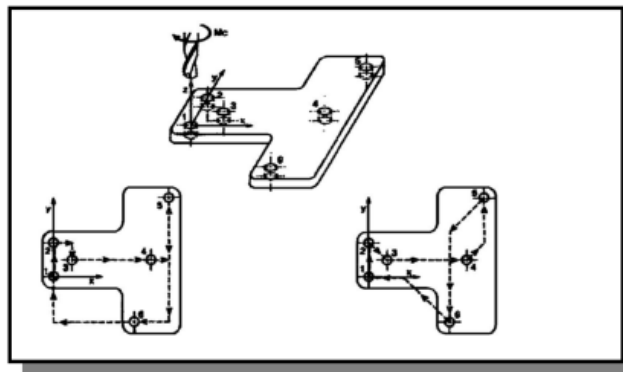
**Figure I.9 : Axes Primaires et Axes Additionnels**

**I.8.4. Classification des MOCN selon le mode d'usinage**

On peut classer les MOCN en trois catégories selon le mode d'usinage : Commande numérique point à point, commande numérique paraxiale, commande numérique de contournage [5].

**a. Commande numérique point à point**

Amène le mobile à un point ou successivement à plusieurs points définis par coordonnées numériques, dans un plan (Figure I.10) [5].



**Figure I.10 : Trajectoires en positionnement point à point**

**b. Commande numérique par axiale**

C'est une machine à commande numérique Point à Point permettant d'usiner durant les déplacements parallèles aux axes. Ce type de déplacement permet par exemple des fraisages précis à vitesses imposées (Figure I.11) [5].

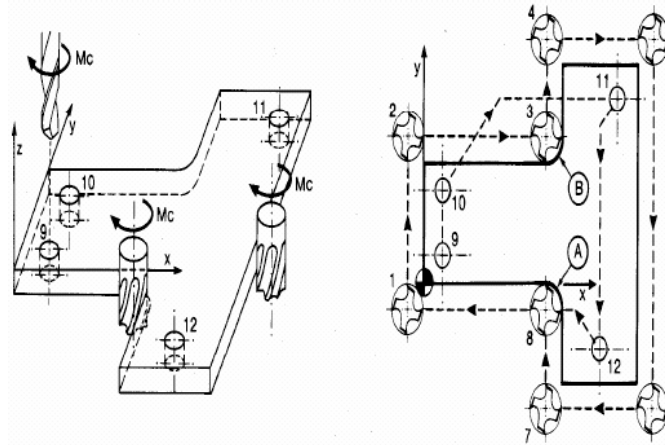


Figure I.11 : Fraisage par axiale

**c. Commande numérique de contournage**

Des interpolateurs linéaires et circulaires permettent de réaliser des trajectoires linéaires et circulaires dans le plan ou dans l'espace (Certains D.C.N. n'autorisent pas l'interpolation circulaire dans l'espace) (Figure I.12). Les différents axes exécutant la trajectoire sont contrôlés en vitesse et en position pour assurer une synchronisation permanente des mouvements [5].

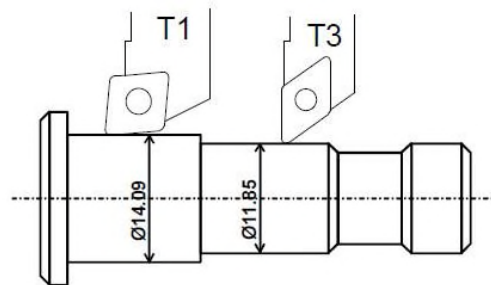


Figure I.12 : Opération de contournage en tournage

**I.9. Commande des MOCN**

De l'idée de départ jusqu'à la fabrication de la pièce sur la machine CNC, tout va se passer sur l'ordinateur, utilisant différents logiciels selon l'étape du processus à effectuer, constituant ainsi une chaîne logicielle.

Les étapes de création :

- La conception 3D.
- La génération des chemins de l'outil, création du G-code.
- Interprétation du G-code et transformation en mouvement sur la machine.

**I.9.1. Conception 3D**

La conception permet d'élaborer, dessiner, visualiser un objet en trois (3) dimensions. La modélisation va plus loin, permettant de modéliser des mouvements des objets en relation les uns avec les autres, de simuler des efforts, etc. Autrement dit, tous les outils de conception 3D ne sont pas nécessairement aussi des outils de modélisation. Il existe plusieurs logiciels de conception et modélisation 3D, tel que le logiciel Solid Works qui est le plus populaire.

### **I.9.2. Génération des chemins de l'outil, création du G-code**

Après la conception en 3D de la pièce, par exemple avec le logiciel SolidWorks, on obtient un fichier de format STL, OBJ, Image ou autre. Pour la génération d'un fichier G-code il existe des logiciels spécialement conçus pour cela, tel que le logiciel Slic3r, Inkscape, Charlygraal et bien d'autre. Aussi, Il est possible de créer un fichier d'usinage directement à partir d'un site spécialement conçu pour cela, tel que Makercam.com.

Le G-code est le langage de programmation pour contrôler une machine à commande numérique. Il est basé sur des lignes de code, plusieurs de ces lignes peuvent être regroupées dans un fichier pour créer un programme G-code, qui sert à programmer les mouvements que la machine va effectuer (droite, arc de cercle, ect). Il est à noter que le G-code est utilisé pour des fraiseuses à commande numérique, des tours, des imprimantes 3D et des lasers de découpe.

Les principales fonctions du G-code sont résumées dans le tableau I.2.

**Tableau I.2 : Principales fonctions du G-code**

<b>Code</b>	<b>Signification</b>
G00	Déplacement rapide
G01	Interpolation linéaire
G02	Interpolation circulaire (sens horaire)
G03	Interpolation circulaire (sens anti-horaire)
G04	Arrêt programme et ouverture carter (pour nettoyer) (temporisation - suivi de l'argument F ou X en secondes)
G10/G11	Écriture de données/Effacement de données
G17	Sélection du plan X-Y
G18	Sélection du plan X-Z
G19	Sélection du plan Y-Z
G20	Programmation en pouces
G21	Programmation en mm
G28	Retour à la position d'origine
G31	Saute la fonction (utilisé pour les capteurs et les mesures de longueur d'outil)
G33	Filetage à pas constant
G34	Filetage à pas variable
G40	Pas de compensation de rayon d'outil
G41	Compensation de rayon d'outil à gauche
G42	Compensation de rayon d'outil à droite
G90	Déplacements en coordonnées absolues

G91	Déplacements en coordonnées relatives
G94/G95	Déplacement en Pouces par minute/Pouce par tour
G96/G97	Vitesse de coupe constante (vitesse de surface constante) /Vitesse de rotation constante ou annulation de G96
M00/M01	Arrêt du programme, arrêt optionnel ou avec condition
M06	Changement d'outil
M30	Fin du programme, réinitialisation, rembobinage

### **I.9.3. Interprétation du G-code et transformation en mouvement sur la machine**

Après la génération d'un fichier G-code, il reste à l'interpréter de façon à pouvoir contrôler les moteurs avec un microcontrôleur plus précisément avec une carte Arduino. Pour cela il existe des logiciels et des micro-logiciels spécialement conçu pour cela, tel que le Grbl.

Le Grbl est un micro-logiciel libre développé sur Arduino pour contrôler des graveuses CNC (Computer Numerical Control), des fraiseuses munis d'une tête mobile contrôlée en X, Y et Z par un ordinateur. Grbl interprète du G-code et déplace en conséquence un outil sur trois (3) axes (X, Y et Z). Il comprend de multiples optimisations sur l'usage et le déplacement des moteurs afin de gérer correctement les accélérations, les trajectoires, etc [7].

### **I.10. Domaines d'utilisation des MOCN**

Les MOCN sont employées dans de nombreux secteurs industriels (Métallurgie, Bois, Textile). Elle est aussi associée à de nouvelles technologies (Laser, Electro-érosion).

- ✓ Perçage, taraudage,
- ✓ Tournage, alésage,
- ✓ Fraisage,
- ✓ Rectification,
- ✓ Soudure continue, par points,
- ✓ Poinçonnage, cisailage.

### **I.11. Conception assistée par ordinateur (CAO)**

#### **I.11.1. Définition**

La CAO (Conception Assistée par Ordinateur) est un ensemble d'outils et de programmes informatiques permettant d'assister l'ingénieur dans la conception et la mise au

point d'un produit. Un système de CAO permet de représenter et d'étudier le fonctionnement d'un objet sans l'avoir fabriqué réellement, c'est-à-dire en virtuel [8].

### **I.11.2. Les outils de la CAO (logiciels)**

Un grand nombre de logiciels de CAO sont disponibles en open-source :

- ✓ Art of Illusion,
- ✓ Blender,
- ✓ FreeCAD,
- ✓ ImplicitCAD,
- ✓ QCAD,
- ✓ pythonOCC,
- ✓ OpenCASCADE,

## **I.12. Fabrication assistée par ordinateur**

### **I.12.1. Définition**

La fabrication assistée par ordinateur ou FAO est d'écrire le fichier contenant le programme de pilotage d'une machine-outil à commande numérique. Ce fichier va décrire précisément les mouvements que doit exécuter la machine-outil pour réaliser la pièce demandée. On appelle également ce type de fichiers : programme ISO ou blocs ISO.

Les logiciels de FAO utilisent les modèles et les ensembles créés dans les logiciels de CAO pour générer des trajectoires d'usinage sur lesquelles s'appuient les machines qui convertissent les conceptions en pièces physiques [9].

### **I.12.2. Les outils logiciels de la FAO**

Les outils de la FAO sont des modeleurs, nous citons :

- ✓ Catia,
- ✓ RhinosCAM,
- ✓ SolidCAM,
- ✓ Power MILL.
- ✓ Hyper MILL.
- ✓ Camworks.

## **I.13. Conclusion**



Dans ce chapitre nous avons présenté une recherche bibliographique sur Les machines –outils à commande numérique (MOCN), les commandes les plus utilisées et les Classifications des MOCN. Ces notions sont nécessaires sont très importantes dans la réalisation d'une MOCN.

Aujourd'hui, l'utilisation des MOCN présente un grand intérêt pour la fabrication en petite et moyenne série ainsi que pour les formes complexes des pièces à usiner.

# Chapitre II

## II.1. Introduction

Comme toute autre machine-outil à commande numérique, la MOCN à trois axes est formée par un ensemble de composants électriques et mécaniques de deux parties complémentaires, une partie commande et une partie opérative. [15]

## II.2. Eléments de la partie commande

### II.2.1. Contrôleur

Le contrôleur est le cerveau de la machine-outil à trois axes. Dans notre projet nous allons utiliser une carte électronique qui est le microcontrôleur Arduino. Bien que de nombreuses variations existent, elles sont interchangeables et basiquement font la même chose. Parfois, le contrôleur est un circuit autonome avec des puces dessus. D'autres fois le contrôleur est un Arduino Méga avec une carte ajoutée dessus appelée 'shield'[10] .

### II.2.2. Arduino

La carte Arduino UNO est un microcontrôleur ATmega328 programmable permettant de faire fonctionner des composants (moteurs, LED...). Elle possède des « ports » permettant par exemple de se connecter à un ordinateur ou de s'alimenter. La figure II.1 illustre un exemple d'une carte Arduino (UNO). [10]

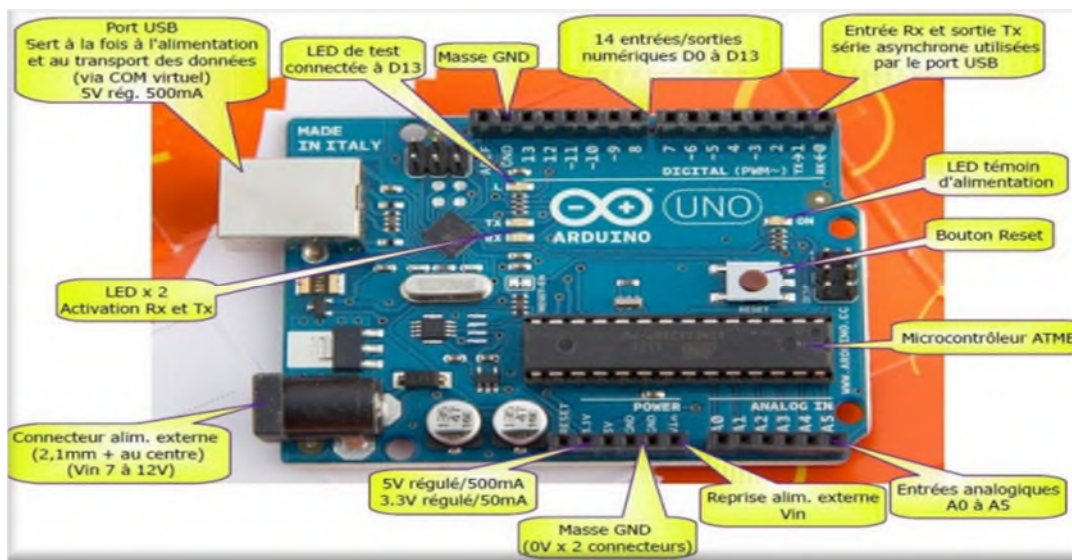


Figure II.1 : Carte Arduino UNO [10]

#### II.2.2.1. Caractéristiques techniques de l'Arduino

Le tableau II.1 montre les caractéristiques techniques de la carte Arduino utilisée dans notre réalisation.

Tableau II.1 : Caractéristiques techniques de la carte Arduino

Microcontrôleur	Atmega328p
Tension de fonctionnement	5 V
Tension d'entrée (limite)	6-20 V
E/S numériques	14 (dont 6 pins fournissent la sortie PWM)
Tension d'entrée (recommandé)	7-20 V
PWM numérique E/S pins	6
DC courant par I O/pin	20 mA
Pins d'entrée analogiques	6
Courant DC pour pins 3.3v	50 mA
Mémoire flash	32 kB (Atmega 328p ) dont 0.5 kB utilisé est pour le boot loader
Sram	2 kB (Atmega 328p )
EEPROM	1 kB (Atmega 328p )
Vitesse de l'horloge	16 Mhz
Longueur	68.6 mm
Largeur	53.4 mm
Poids	25 g

### II.1.2.2. Détails techniques

La carte Arduino UNO peut être alimentée via la connexion USB ou avec une alimentation externe. La source d'alimentation est automatiquement sélectionnée. Une alimentation externe peut provenir soit d'un adaptateur AC-DC ou d'une batterie. L'adaptateur peut être connecté en branchant une prise dans la prise d'alimentation de la carte ou à partir d'une batterie connectée dans le pin (ou broche) GND et V-in (alimentation externe).

Le processeur peut fonctionner sur une alimentation externe de 6 à 20 V. Cependant, si la tension est inférieure à 7 V, le pin 5 V peut fournir moins de cinq volts et le processeur peut devenir instable. Si la tension est supérieure à 12 V, le régulateur de tension peut surchauffer et endommager la carte. La plage recommandée est de 7 à 12 V.

Les pins (ou broches) d'alimentation sont les suivants :

- **V-in** : Tension d'entrée à la carte Arduino à l'aide d'une source d'alimentation externe (par opposition à 5 V de la connexion USB ou une autre source d'alimentation régulée). Si

l'alimentation en tension est faite par l'intermédiaire de la prise d'alimentation, on pourra y accéder via ce pin,

- **5 V** : Ce pin délivre une tension de 5 V régulée par la carte. Le processeur peut être alimenté soit à partir de la prise d'alimentation DC (7-12 V), le connecteur USB (5 V), ou le pin V-in de la carte (7-12 V). La fourniture d'une tension via les 5 V ou 3,3 V contourne le régulateur, et peut endommager votre processeur,
- **3,3 V** : Une alimentation de 3,3 V est générée par le régulateur. La consommation de courant maximale est de 50 mA,
- **GND** : Masse,
- **IOREF** : Ce pin sur la carte Arduino fournit la référence de tension avec laquelle le microcontrôleur fonctionne.

### II.1.2.3. Mémoire

L'ATmega328 a 32 ko (avec 0,5 kB occupées par le bootloader). Il a également 2 ko de SRAM et 1 ko de mémoire EEPROM (qui peut être lue et écrite avec la bibliothèque de l'EEPROM).

### II.2.2.4. Entrées et sorties

Chacune des 14 broches numériques sur la carte UNO peut être utilisée comme une entrée ou une sortie, en utilisant les fonctions `pinMode()`, `digitalWrite()`, et `digitalRead()`. Ils fonctionnent à 5 V. Chaque broche peut fournir ou recevoir 20 mA en état de fonctionnement recommandé et a une résistance de pull-up interne (déconnecté par défaut) de 20-50 kΩ. Un maximum de 40 mA est la valeur qui ne doit pas être dépassée sur toutes les broches d'Entrées/Sorties pour éviter des dommages permanents au microcontrôleur.

Certaines broches ont des fonctions spécialisées :

- **Série** : 0 (RX) et 1 (TX), permettent de recevoir (RX) et transmettre (TX) TTL données série. Ces pins sont connectés aux pins correspondants de l'USB-TTL puce Serial ATmega8U2.
- **LED** : 13, elle est équipée d'un conduit par la broche numérique 13. Lorsque la broche est à la valeur HIGH, la LED est allumée et elle est éteinte lorsqu'elle est hors tension.

L'UNO dispose de 6 entrées analogiques, A0 à A5, dont chacune fournit 10K bits de résolution (ou 1024 valeurs différentes). Par défaut, la tension est de 5 V. Il est cependant possible de changer la limite supérieure de la gamme en utilisant la broche AREF et la fonction `analogReference()`.

Autres broches de la carte :

- AREF. Tension de référence pour les entrées analogiques. Pin utilisé avec analogReference(),
- Réinitialiser,
- **PWM**

En numérique, les signaux ont toujours une valeur LOW ou HIGH (0 ou 1). Le principe est de construire un signal alternativement LOW ou HIGH et de répéter cette alternance. La LED est donc alternativement allumée et éteinte mais le cycle est tellement rapide que la persistance rétinienne nous donne l'illusion d'une LED allumée en permanence.

- **Bootloader**

Il s'agit de la racine du système, à la manière du bios sous Windows, où l'on peut installer un recovery alternatif, un kernel, une image d'Android, parmi d'autres choses... après l'avoir ouvert.

- **EEPROM**

Mémoire dont les valeurs sont conservées lorsque la carte est éteinte (comme un petit disque dur) [10]

### **II.2.3. CNC-Shield**

#### **II.2.3.1. Description**

La carte CNC V3 (Figure II.2) est une carte "shields" qui se connecte à la carte Arduino UNO ou équivalente. Elle fut créée spécialement pour les machines à découpe numérique (CNC). C'est la carte d'interface qui distribue la puissance à l'aide des pilotes (driver A4988 ou DVR8825) enfichable. Elle va commander les moteurs pas à pas dans notre machine outils à 3 axes, sur les axe X,Y,Z ainsi que la commande de rotation. Elle permet également de traiter les signaux du refroidissement, ainsi que des fins de courses mini et maxi. [10]

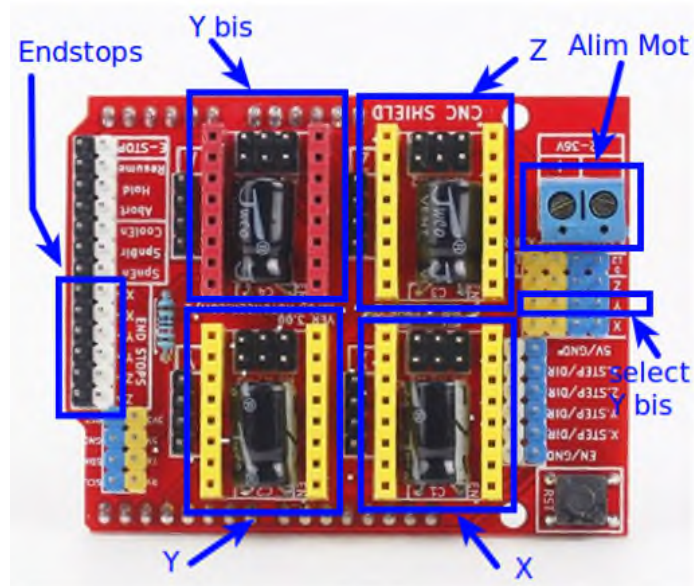


Figure II.2 : CNC shield (La rampe) [10]

### II.2.3.2. Caractéristiques techniques

- Compatible avec Arduino UNO.
- Compatible avec GRBL 0.8c.
- Tension d'entrée : 12 – 36 V.
- Axes à piloter : X, Y, Z.
- Commande pour le liquide de refroidissement : NON.
- Entrées de fin de course : NON.

### II.2.4. Pilotes A4988 et DRV8825

#### II.2.4.1. Descriptions

Les pilotes de moteurs pas à pas sont des petits modules qui permettent de contrôler la rotation d'un moteur en fonction des instructions reçues de la carte de pilotage. Ils sont connectés à cette dernière, soit directement, soit par l'intermédiaire d'une interface.

Pour les imprimantes 3D, les modèles couramment utilisés sont le A4988 et le DRV8825 (Figure II.3).

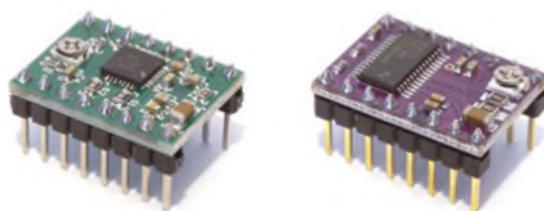


Figure II.3 : Pilotes A4988 et DRV8825[11]

### II.2.4.2. Rôle et réglage des pilotes moteurs pas à pas

Ces modules gèrent des moteurs pas à pas bipolaires ou unipolaires. La majorité des cartes de pilotage et leurs adaptateurs dédiés aux machines à commande numériques, sont conçus pour utiliser des moteurs bipolaires, dotés de 4 fils (2 par bobine). Ces moteurs pas à pas ont généralement une résolution angulaire de 200 pas / tour ( $1,8^\circ$ ), ou de 400 ( $0,9^\circ$ ). [12]

La résolution angulaire a un impact direct sur la précision des mouvements de nos machines. L'avantage de ces pilotes est d'avoir la capacité d'augmenter le nombre de micros pas des moteurs pour en améliorer la précision. Pour y parvenir, ils font varier les champs magnétiques des 2 bobines pour créer des positions intermédiaires. [12]

#### a. DRV8825

Il offre une résolution supplémentaire de  $1/32^{\text{ème}}$  de pas. Un moteur qui a 200 pas, passe à 6400 pas / tour ( $0,05625^\circ$ ) pour  $1/32$ . Pour des raisons évidentes de qualité, comme pour les **A4988**, il suffit de déplacer des cavaliers ou des switches situés par groupes de 3 sous les emplacements des pilotes. Un circuit fermé  $M = 1$  et ouvert  $M = 0$ . Le tableau II.2 représente les entrées

**Tableau II.2 :** entrée de signaux des trois switch M1,M2 et M3

M0	M1	M2	Résolution du micro pas
0	0	0	Pas complet (valeur moteur)
1	0	0	1/2 de pas
0	1	0	1/4 de pas
1	1	0	1/8 de pas
0	0	1	1/16 de pas
1	0	1	1/32 de pas
0	1	1	1/32 de pas
1	1	1	1/32 de pas

Le DRV8825 a une capacité de débiter 1,5 A par phase en courant continu, et de 2,2 A à 2,5 A maximum (nécessite aussi une ventilation en plus du dissipateur). Il gère des tensions de 8,2 à 45 V (par sécurité ils disposent tous d'un dispositif d'arrêt en cas de surchauffe). Il est donc important de choisir le pilote en fonction des spécifications du moteur pour éviter tout problème. Il est aussi important de ne pas utiliser un moteur nécessitant une puissance



supérieure à ce que peut délivrer un pilote au risque de devoir limiter sa puissance. L'intensité par phase nécessaire au moteur est clairement indiquée dans sa documentation.

Dans notre projet on a utilisé le microcontrôleur A4988. [6].

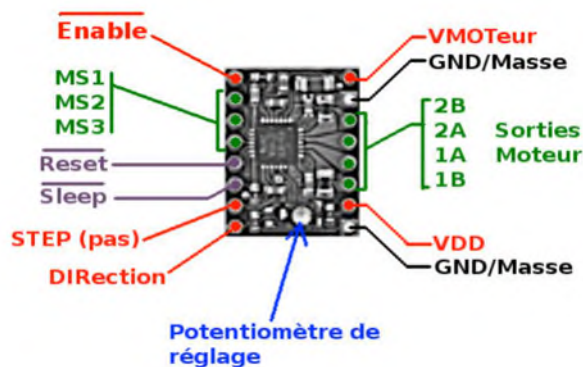
**b. Microcontrôleur A4988**

Il offre la possibilité d'augmenter la finesse du pas dans les proportions suivantes : 1, 1/2, 1/4, 1/8 et 1/16<sup>ème</sup> de pas. Un moteur qui a 200 pas, passe respectivement à 3200 pas / tour (0,1125°) pour le réglage de 1/16. Pour paramétrer ces micros pas, il suffit de déplacer des cavaliers ou des switches situés par groupes de 3 sous les emplacements des pilotes. Un circuit fermé M = 1 et ouvert M = 0.

**Tableau II.3 :** montrant les entrées des switch M1,M2 et M3 du A4988

M0	M1	M2	Résolution du micro pas
0	0	0	Pas complet (valeur moteur)
1	0	0	1/2 pas
0	1	0	1/4 de pas
1	1	0	1/8 de pas
1	1	1	1/16 de pas

L'utilité d'une résolution élevée n'est pas l'unique critère de choix à prendre en compte. La puissance délivrée par le composant est très importante. Bien qu'il y ait plusieurs versions de ces pilotes, le module A4988 a majoritairement a une capacité de délivrer 1 A par phase (bobine moteur) en courant continu, et de 2 A maximum (nécessite une ventilation en plus du dissipateur). Il gère les tensions de 8 à 35 V. La figure II.4 illustre les entres et les sorties du microcontrôleur A4988 [6]



**Figure II.4 :** Entres/Sorties du microcontrôleur A4988

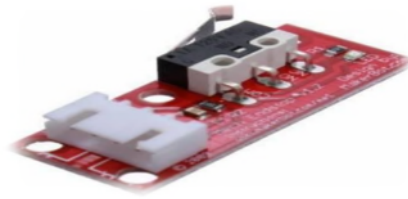
- **Enable** : Logique inversée, permet d'activer ou de désactiver le moteur.  
Etat *Haut* = *High* = *Moteur actif* et axe bloqué entre les pas. Etat *bas* = *Low* = *Axe totalement libre*.
- **MS1, MS2, MS3** : permettent de sélectionner la configuration Step/MicroStep. Ces broches disposent de résistances Pull-Down ramenant le potentiel à 0 V lorsque rien n'est connecté sur ces broches.
- **Reset** : Logique inversée, permet de faire une réinitialisation du module généralement connecté sur la broche "sleep".
- **Sleep** : Logique inversée, généralement connecté sur la broche "Reset" du module.
- **Step** : Envoyer un signal d'horloge (Niveau Haut puis Niveau bas, High puis Low) pour faire avancer le moteur d'un pas.
- **DIR** : Permet d'indiquer la direction de rotation du moteur. Etat *Haut* = *High* pour tourner dans un sens, Etat *bas* = *Low* pour tourner dans l'autre sens.
- **VMot** : Tension d'alimentation du moteur. Habituellement 12 V pour les moteurs pas à pas. Tension entre 8 et 12 V.
- **GND** : Sous "VMOT", masse pour l'alimentation moteur. Habituellement mise en commun avec la masse de logique de commande (celle sous "VDD").
- **2B 2A** : Première bobine du moteur pas à pas bipolaire.
- **1A 1B** : Deuxième bobine du moteur pas à pas bipolaire.
- **VDD** : Alimentation de la logique de commande entre 3 et 5.5 V (habituellement 5 V).
- **GND** : Sous "VDD", masse de la logique de commande. Souvent mise en commun avec la masse d'alimentation du moteur.

### II.2.5. Fins de course

Afin de bien protéger notre matériel il faut utiliser des appareils qui s'appellent End\_ (Stop fins de cours). C'est un capteur de position qui indique la fin de course des axes, il y a 3 groupes de deux connecteurs dans l'ordre X, Y, Z. Chaque groupe de connecteur commence par End\_Stop Min, suivit de End\_Stop Max du même axe.

La figure II.5 montre la photo du fin de course.

**Remarque** : dans ce travail on a pas utilisé les fins de course puisque notre logiciel nous permet de limiter l'espace du travail.



**Figure II.5 :** Photo du fin de course

### **II.3. Eléments de la partie opérative « Moteurs pas à pas »**

Le moteur pas à pas est un convertisseur électromécanique qui assure la transformation d'un signal électrique impulsionnel en un déplacement mécanique (angulaire ou linéaire). Sa structure de base se présente sous la forme de deux pièces séparées mécaniquement, le stator et le rotor. L'interaction électromagnétique entre ces deux parties assure la rotation. [12]

#### **II.3.1. Différents types de moteurs pas à pas**

Les moteurs pas à pas sont très utilisés dans toutes les applications mécaniques où l'on doit contrôler simplement la position ou la vitesse d'un système en boucle ouverte. Ces moteurs sont par exemple utilisés dans les imprimantes jet d'encre ou laser, pour positionner les têtes d'impression ou pour l'avancée du papier. Il existe 3 types de moteurs pas à pas, à aimants permanents, à réluctance variable ou hybrides. [12]

##### **II.3.1.1. Moteurs à aimant permanent**

C'est le modèle dont le concept de fonctionnement est le plus simple. Le rotor est constitué d'un aimant permanent, et le stator comporte deux paires de bobines. En agissant sur les bobines alimentées, et le sens des courants, on fait varier le champ créé par le stator. A chaque pas, la direction du champ induit par le stator tourne de  $90^\circ$ . L'aimant permanent suit le déplacement du champ magnétique créé par les bobines et s'oriente selon une de ses quatre positions stables. Comme le rotor est aimanté, lorsque le moteur n'est pas alimenté le flux magnétique dû à l'aimant permanent crée un couple résiduel en se plaçant dans l'axe de l'une des bobines.

Pour augmenter le nombre de positions stables et donc de pas du moteur à aimant permanent, on peut alimenter successivement une puis deux paires de bobines : c'est le mode "demi-pas". A chaque pas, la direction du champ induit par le stator tourne de  $45^\circ$ . Dans ce mode, le couple est différent pour les pas pairs et impairs. Les moteurs pas à pas à aimant permanent ont un couple moteur important, mais un nombre de pas par tour faible, et une

fréquence de rotation maximale faible. La commande de ces moteurs pas à pas nécessite de contrôler le sens du courant dans chaque bobine.

### II.3.1.2. Moteurs à réluctance variable

Le rotor est en fer doux et comporte un nombre de pôles différent du stator. Le rotor tourne de façon est ce que le flux le traversant soit maximum. Ces moteurs n'ont aucun couple de maintien si aucune bobine n'est alimentée. [12]

### II.3.1.3. Moteurs hybrides

Le rotor est constitué par deux pièces en fer doux ayant chacune  $n$  pôles séparées par un aimant permanent magnétisé dans le sens de l'axe du rotor. Le nombre  $m$  de pôles du stator est différent de celui du rotor. Le rotor se déplace pour que le flux qui le traverse soit maximum. En mode pas entier, les bobines sont alimentées paire par paire alternativement avec inversion à chaque pas. Il est nécessaire d'avoir un rotor polarisé pour imposer le sens de rotation à chaque commutation.

Pour les moteurs pas a pas bipolaires hybrides , (stator avec deux paires de bobines et rotor avec rotor à deux fois trois pôles) à chaque pas, la direction du champ induit par le stator tourne de  $30^\circ$  en mode "pas entier" et de  $15^\circ$  en mode "demi-pas". Un modèle classique comporte un stator avec 8 pôles ayant chacun 5 dents et un rotor avec 2 pièces de 50 dents ce qui donne 200 pas par tour en mode "pas entier".

On peut aussi classer les moteurs pas à pas en deux types selon l'alimentation de bobinage, à savoir les moteurs bipolaires et les moteurs unipolaires.

#### 🚦 Les moteurs bipolaires

Ils ont en général quatre (4) fils et se commandent en inversant le sens du courant dans une des deux bobines, tel que c'est représenté dans la figure II.6, qui sont alimentées une fois dans un sens et une fois dans l'autre sens. Ils créent une fois un pôle nord et une fois un pôle sud d'où le nom de bipolaire.

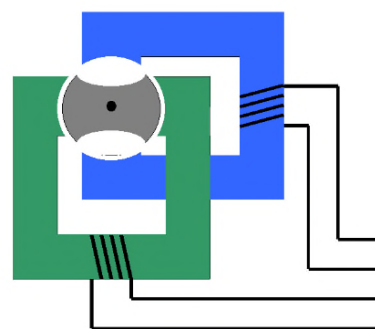


Figure II.6 : Moteur bipolaire

### 🚦 Moteurs unipolaires

Ils ont en général six (6) fils, dont deux sont reliés au milieu des bobines, tel que c'est représenté dans la figure II.7. Ils se commandent en les alimentant tour à tour, mais les bobinages sont alimentés toujours dans le même sens par une tension unique d'où le nom d'unipolaire.

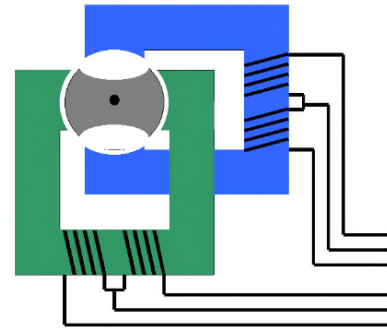


Figure II.7 : Moteur unipolaire

### II.3.2. Moteurs Nema 17

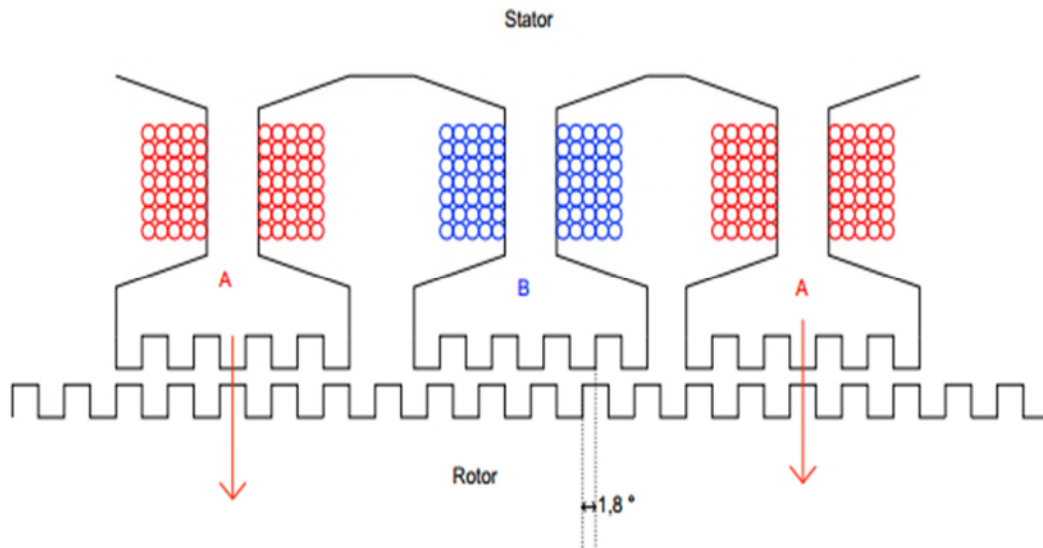
Dans notre projet nous allons utiliser les moteurs Nema 17 (moteurs hybride bipolaires). Les caractéristiques techniques de ce moteur sont données ci-dessous :

- ✓ 200 pas par tour ( $1,8^\circ/\text{pas}$ ),
- ✓ 2 phases (bipolaire),
- ✓ Cordon de 4 fils,
- ✓ Voltage : 2 V DC,
- ✓ Courant 1.2 A,
- ✓ Résistance de la phase :  $1.7 \Omega \pm 10 \%$  ( $20^\circ\text{C}$ ),
- ✓ Inductance de la phase :  $4.5 \text{ mH} \pm 20 \%$  ( $1 \text{ kHz } 1 \text{ Vrms}$ ),
- ✓ Couple de maintien : 0.4 N.m min,
- ✓ Diamètre de l'axe : 5 mm/0.188" (3/16"),
- ✓ Longueur de l'axe : 22 mm,
- ✓ Hauteur du corps du moteur : 40 mm.

## II.3. Principe de fonctionnement du moteur pas à pas et de sa commande

### II.3.1. Moteur à réluctance variable et moteur hybride

Dans un moteur pas à pas à réluctance variable, les bobines d'excitation sont sur le stator. Le rotor est en acier de forte perméabilité, et n'a pas d'aimantation permanente. Le stator et le rotor sont munis d'encoches. La figure suivante montre un détail avec quelques encoches. La représentation est rectiligne (au lieu de circulaire) . [12]



**Figure II.8 :** Dentures du stator et du rotor d'un moteurs pas à pas

Dans la position représentée, les dents des pôles A du stator sont en vis-à-vis des dents du rotor, alors que celle des pôles B sont décalées. Les pôles A sont excités par un bobinage formant la phase A (en rouge sur la figure). Dans la configuration représentée, un courant circule dans la phase A mais pas dans la phase B. Le champ magnétique généré dans les pôles A induit une aimantation dans le rotor en acier. La configuration stable est celle où les dents du rotor sont alignées avec celles du stator. Si l'on tente d'écarter le rotor de cette position, un couple de rappel s'exerce. Lorsqu'on coupe le courant dans la phase A et qu'on alimente le bobinage de la phase B, le rotor tourne pour obtenir l'alignement avec les dents des pôles B. Si le rotor comporte 100 dents, cela fait une rotation de 1,8 degrés (moteur 200 pas par tour).

Les moteurs pas à pas sont souvent de type hybride, avec une aimantation permanente du rotor. Le rotor d'un moteur hybride est constitué de deux roues dentées décalées. La première roue est un pôle sud magnétique et se trouve en vis-à-vis des pôles A du stator. La seconde roue est un pôle nord magnétique et se trouve en vis-à-vis des pôles B du stator. La figure suivante représente un moteur hybride dans la configuration où la phase A est alimentée. Les dents du pôle sud du rotor sont alors alignées avec celle des pôles A du stator.

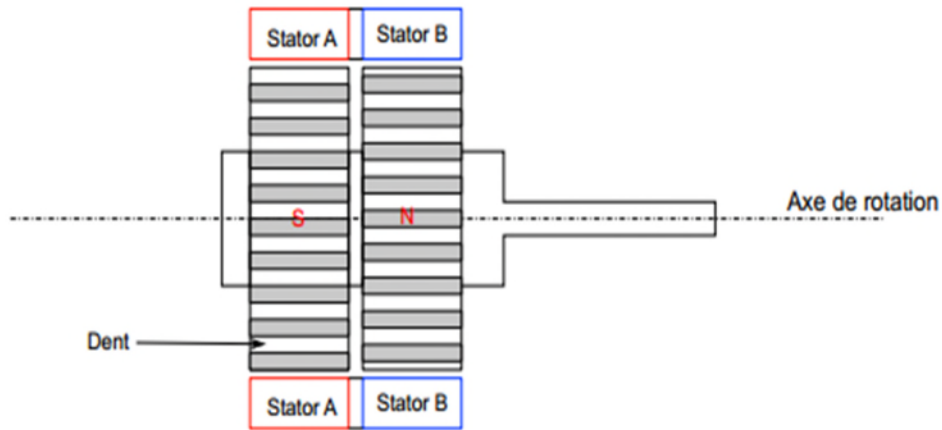


Figure II.9 : axe de rotor et phase statorique d'un moteur pas a pas hybride [12]

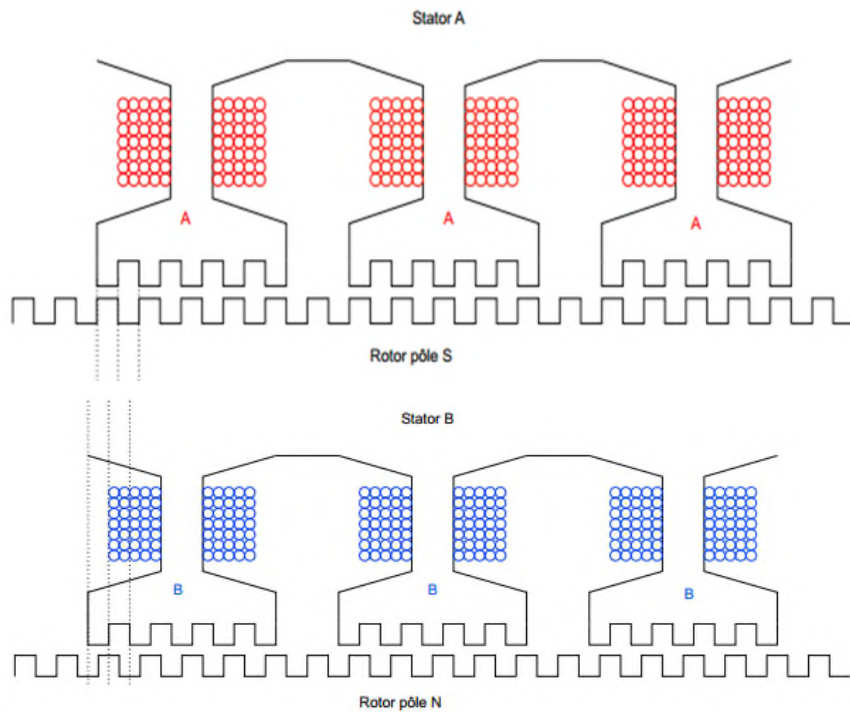


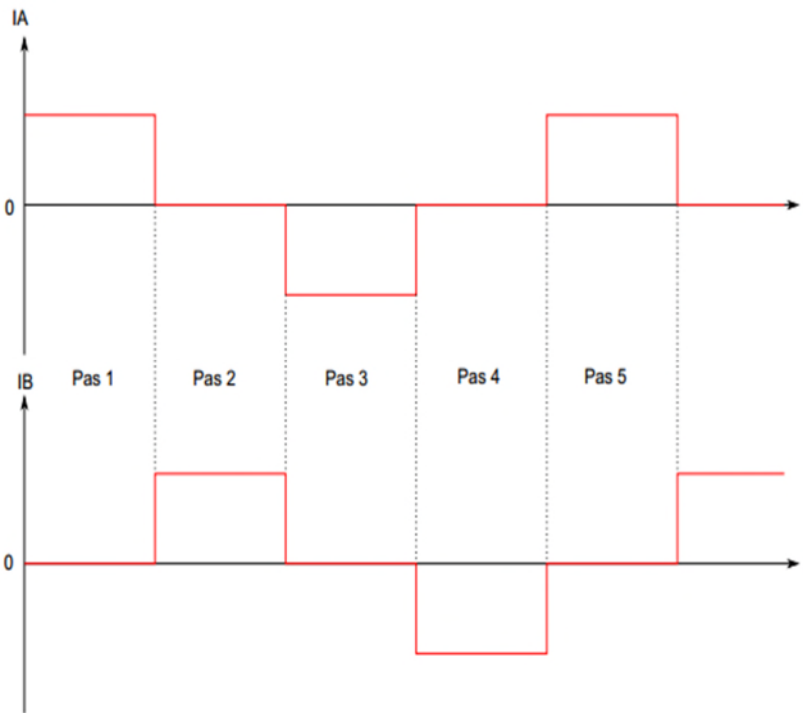
Figure II.10 : Bobinage d'un moteur hybrides (phase A alimenté ) [12]

Dans un moteur hybride, il y a un faible couple même en l'absence de courant, qui bloque le rotor dans une position d'alignement, soit avec les pôles A soit avec les pôles B. Ce couple permet souvent de bloquer le moteur au repos sans lui fournir du courant (à condition que la charge au repos soit faible).

Un moteur pas à pas bipolaire comporte 4 fils d'alimentation, deux pour la phase A, deux pour la phase B. Un ohmmètre permet facilement de repérer les deux paires. Pour les moteurs unipolaires, qui comportent deux bobines en sens inverse par pôle (deux pour le pôle A, deux pour le pôle B).

### II.3.2. Excitation des phases

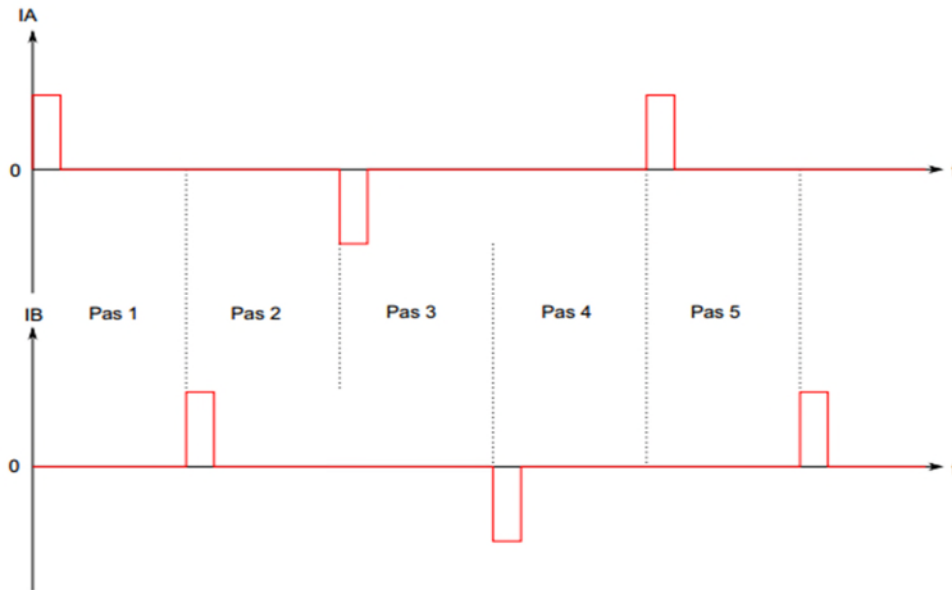
- ❖ Dans le premier mode d'excitation, la rotation du rotor se fait lorsqu'on coupe le courant dans une phase tout en déclenchant l'alimentation de l'autre phase. Dans ce mode, le courant ne circule que dans une phase à la fois (mode One Phase ON, full step). Le sens du courant est inversé à chaque alternance (Figure II.11). Ce mode est aussi appelé excitation ondulée. [12]



**Figure II.11 :** Courants des phases A et B en fonction des pas

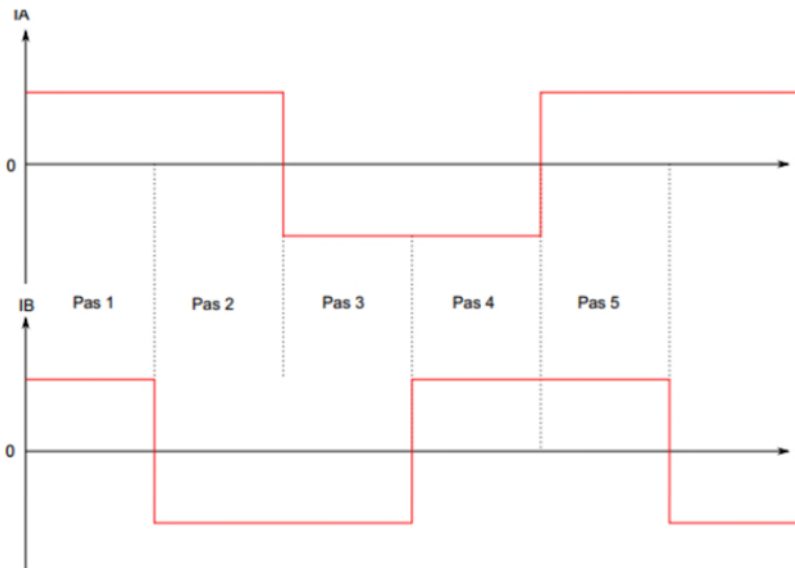
La rotation se fait très rapidement, avec un temps de l'ordre de la milliseconde, juste après le changement des courants (il faudra tout de même tenir compte du temps d'établissement du courant). La durée des pas est généralement de plusieurs dizaines de millisecondes. Le courant est maintenu de manière à bloquer le rotor dans sa position. Si la charge est faible, on peut aussi couper le courant entre les pas, après avoir appliqué une impulsion assez longue pour provoquer la rotation. Par exemple, si les pas sont espacés d'une seconde, on peut appliquer des impulsions de 50 millisecondes suivies de plages de courant nul (Figure II.12). [12]





**Figure II.12 :** Impulsions de dixième de millisecondes dans les phases A et B

- ❖ Le second mode (Two Phase ON, Full Step) consiste à alimenter les deux phases en même temps (Figure II.13), et à changer le sens du courant à chaque pas. Avec ce mode, le couple moteur est plus important, mais la dissipation est évidemment plus grande (dans le moteur est dans le circuit de commande). Ce mode est aussi appelé excitation standard.



**Figure II.13 :** Impulsion alimentant deux phases du moteur

- ❖ Le troisième mode permet de faire des déplacements par demi-pas (mode Half Step). La figure II.14 illustre les impulsions des courants injectés dans les phases A et B.

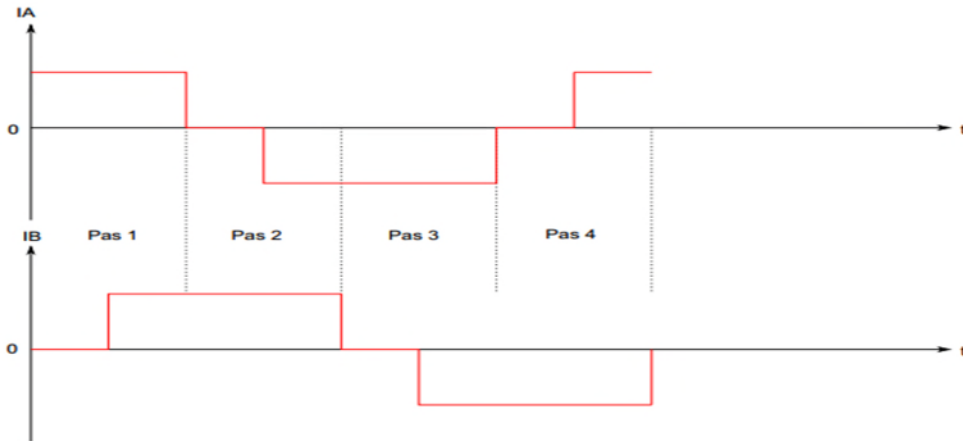


Figure II.14 : Impulsions alimentant un pas et demi

### II.4. Pont en H à transistor bipolaire

Un pont en H permet d'alimenter un bobinage du moteur (une phase) en permettant de changer le sens du courant. Voici le schéma de principe d'un pont en H :

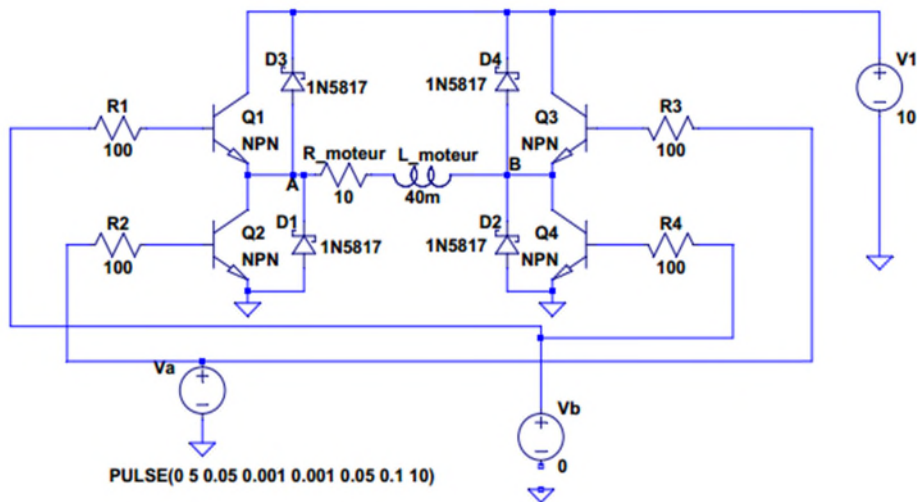


Figure II.15 : Pont en H de transistors

Le bobinage est représenté par une résistance et une inductance. Lorsque  $V_a = 5 V$  et  $V_b = 0$ , les transistors Q2 et Q3 sont passants et un courant positif circule dans le moteur de B vers A. Inversement, lorsque  $V_a = 0$  et  $V_b = 5 V$ , un courant positif circule de A vers B. Lorsque les deux tensions de commande sont à zéro, le courant dans le bobinage est nul. Les diodes permettent au courant de se réduire continûment lorsque les transistors sont bloqués (diodes de roue libre).[12]

## **II.4. Conclusion**

On a consacré ce chapitre à la partie électrique car c'est la partie qui contient les composants nécessaires pour faire fonctionner la commande et permettre d'exécuter notre programme on a :

- ✓ définis les éléments qui constituent cette partie.
- ✓ Illustré ces composants et donner leur caractéristiques techniques.
- ✓ Une vue sur les moteurs pas à pas et leurs commandes plus précisément les moteurs utilisés (NEMA17)

# Chapitre III

### **III.1. Introduction**

Ce chapitre est réservé à la conception assistée par ordinateur (CAO) et la réalisation de la partie mécanique de notre machine-outil à 3 axes. La conception de toutes les pièces de la machine est faite avec le logiciel SolidWorks.

Pour faciliter la réalisation de notre machine, nous avons décidé de réaliser tout d'abord le châssis puis les systèmes d'axes (X, Y, Z) séparément et enfin faire l'assemblage des différentes parties.

### **III.2. Caractéristiques visées de la machine**

L'objectif recherché est de réaliser une petite machine qui peut faire le déplacement sur les trois axes dans la zone d'opération. Cette dernière pourra être utilisée plus tard pour des applications 3D, telles que les imprimantes 3D, fraiseuse numérique à 3 axes...etc.

#### **III.2.1. Zone d'opération**

- Axe X : 200 mm.
- Axe Y : 200 mm.
- Axe Z : 100 mm.

#### **III.2.2. Dimensions**

- Longueur : 300 mm
- Largeur : 300 mm
- Hauteur : 400 mm

#### **III.2.3. Matériaux à utiliser**

Nous avons décidé de réaliser notre machine par le bois à cause de sa disponibilité, sa facilité d'usinage, son faible coût et sa légèreté. Pour les tiges filetées et les tiges de guidage, elles sont en acier inoxydable. Enfin, les supports de transmission sont réalisés avec l'acier.

### **III.3. Réalisation du châssis de la machine**

En premier lieu, et dans le but d'avoir une structure solide et résistante contre toute vibration et mouvement, nous avons au début conçu le châssis de la machine. Ce dernier aura pour rôle de supporter les trois axes de mouvement de la tête qui effectue les opérations. En plus du poids de la machine, le châssis doit supporter une charge supplémentaire (pièce à usinée par exemple) en cas d'utilisation en fraiseuse numérique ou bien imprimante 3D.

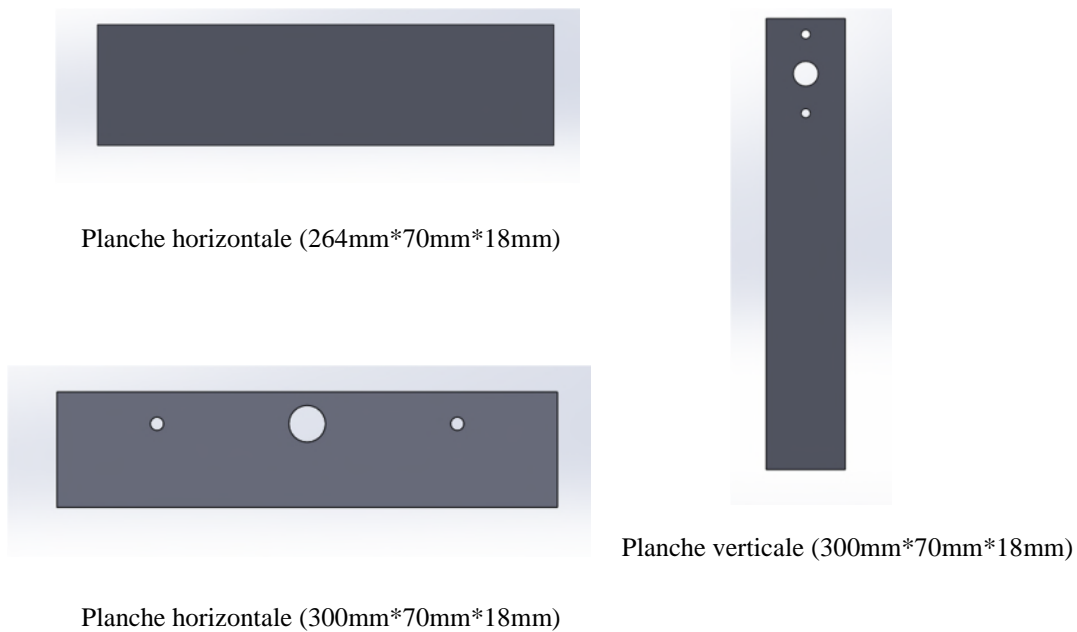
Le châssis de notre machine est réalisé par six planches (en bois), quatre pour la partie inférieure qui supporte la table et son système de transmission (axe Y) et deux planches

verticales sur lesquelles se fixent les systèmes de transmission de mouvement des deux axes X et Z.

Deux planches opposées de la partie inférieure et les deux planches verticales ont été percées dans trois endroits différents pour supporter les deux tiges de guidage et les deux roulements à billes (6000-2Z-C) pour chaque axe (X, Y).

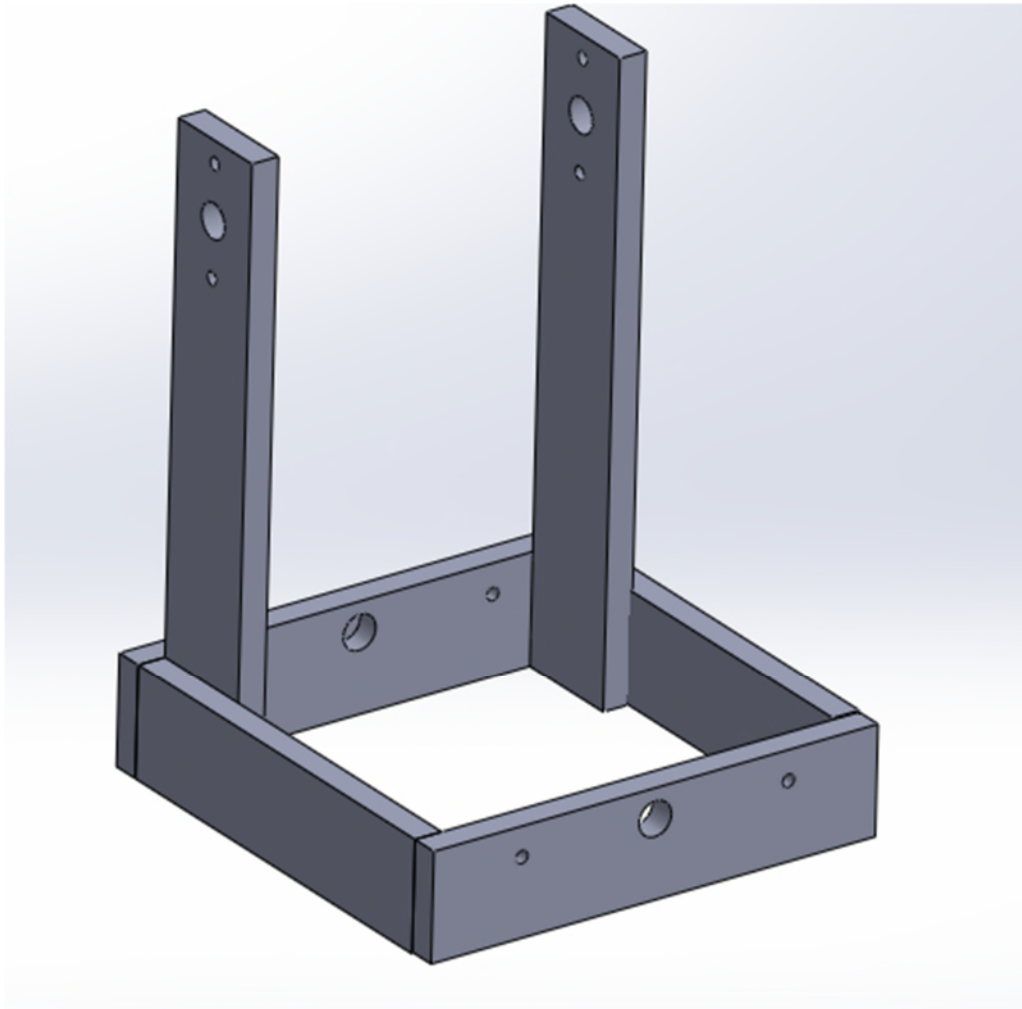
L'assemblage de la structure est effectué par une colle spéciale et des vis (M5). La partie inférieure du châssis est renforcées par des équerres dans ses quatre coins.

La figure III.1 illustre les planches et leurs dimensions utilisées dans la réalisation du châssis.



**Figure III.1** : Planches utilisées dans la réalisation du châssis

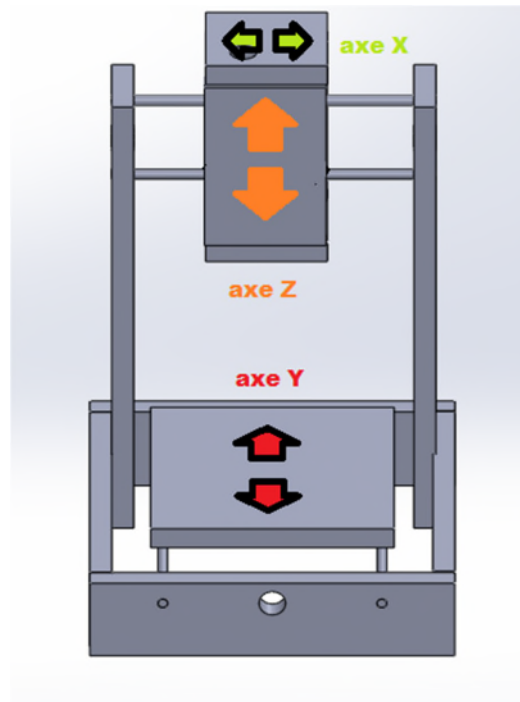
La figure III.2 montre la forme du châssis après assemblage des planches.



**Figure III.2 :** Châssis de la machine-outil à 3 axes

#### **III.4. Réalisation des mécanismes de mouvement sur les trois axes**

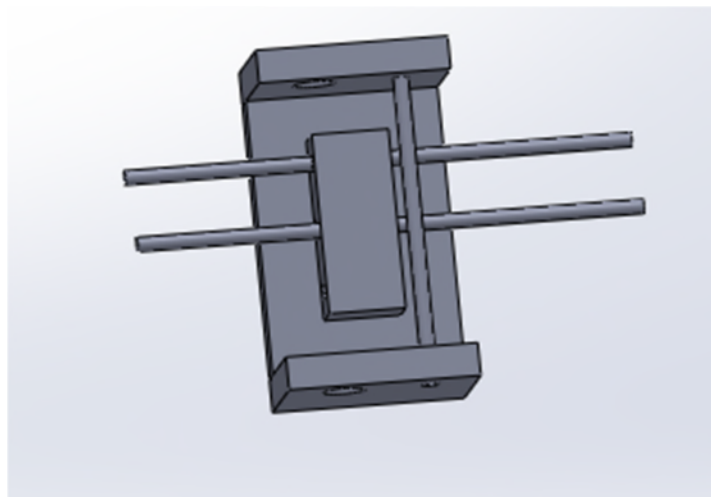
La figure III.3 résume le système d'axes de notre machine. L'axe *X* concerne le mouvement horizontal de la tête pilotée (d'usinage), son mouvement vertical est désigné par l'axe *Z* et enfin le mouvement de la table est l'axe *Y*.



**Figure III.3 :** Système d'axe d'une machine-outil à trois axes avec table mobile

### III.4.1. Axe X

L'axe X est une partie importante dans notre machine car il porte avec lui le mécanisme d'un autre axe, à savoir l'axe Z. Pour assurer un mouvement libre avec un minimum de frottements et sans vibrations, nous avons fixé une planche (100mm\*160mm\*18mm) verticalement sur un support en acier porté par des tiges lisses en acier inoxydable. Cet assemblage avec une tige filetée et un écrou soudé sur le support en acier permet de faire un déplacement linéaire qui définit l'axe X. La figure III.4 illustre le mécanisme de mouvement suivant l'axe X de la machine à réaliser.

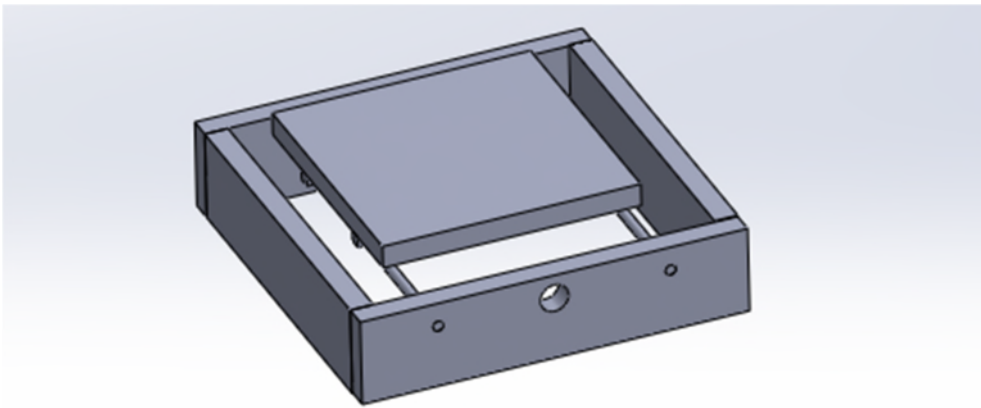




**Figure III.4 :** Mécanisme de déplacement linéaire de la tête d'usinage suivant l'axe X

### III.4.2. Axe Y

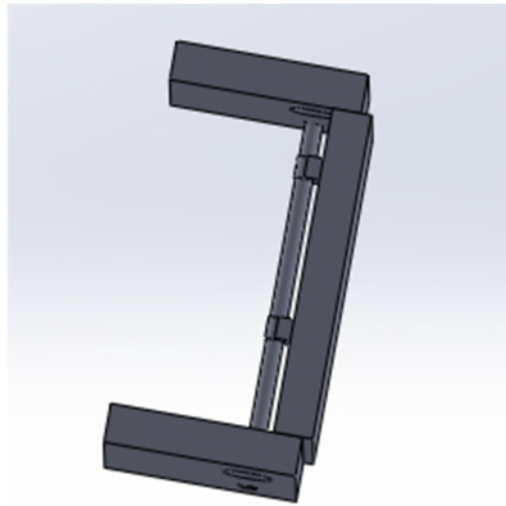
L'axe Y est aussi important qu'il porte le plateau ou s'effectue les opérations (c'est celui qui porte la pièce au cours de l'usinage). Dans notre cas, l'axe Y constitue la table (200\*mm\*200mm\*18mm) de la machine. Comme le cas de l'axe X, cette table est fixée sur un support métallique porté par deux tiges. Dans l'axe Y on utilise aussi la transmission tige filetée-écrou pour assurer le déplacement linéaire de la table. La figure III.5 montre la forme de la table et son mécanisme de déplacement linéaire.



**Figure III.5 :** Mécanisme de déplacement linéaire suivant l'axe Y

### III.4.3. Axe Z

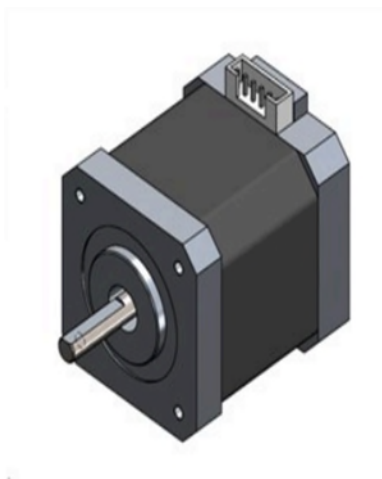
L'axe Z permet de porter le système de la tête qui effectue l'usinage de la pièce. Pour effectuer un mouvement libre sans frottement on a utilisé le même système que les deux axes X et Y : tige filetée-écrou assurant un déplacement linéaire de la table sur des tiges lisses. La figure III.6 présente le mécanisme de déplacement linéaire de la tête d'usinage suivant l'axe Z.



**Figure III.6 :** Mécanisme de déplacement linéaire de la tête d'usinage suivant l'axe Z

### III.5. Moteurs d'entraînement

Les trois axes doivent leurs mouvements au moteur pas à pas (Figure III.7.a) qui donne un mouvement de rotation à l'arbre de sortie, et le transmette directement au coupleur (Figure III.7.b). Les moteurs sont fixés sur les planches par des vis 0.8.



a. Moteur pas à pas



b. coupleur

### III.6. Mécanisme de transmission de mouvement

La fonction du mécanisme de transmission de mouvement dans notre machine est de transformer la rotation du coupleur en un déplacement linéaire (translation). Il nous faut donc un système qui transforme un mouvement de rotation en mouvement de translation. Pour les trois axes de la machine, on a choisi le système tige filetée-écrou pour son efficacité et sa simplicité de réalisation. La tige filetée est supportée par deux roulements aux extrémités et reliée au coupleur. L'écrou, quant à lui, est soudé le support métallique fixé sur la planche (la table par exemple pour l'axe Y). A la rotation du moteur, le coupleur tourne en transmettant le mouvement de rotation à la tige filetée. Cette dernière montée sur l'écrou, permet d'avoir un mouvement de translation en assurant le déplacement linéaire de la partie concernée de la machine.

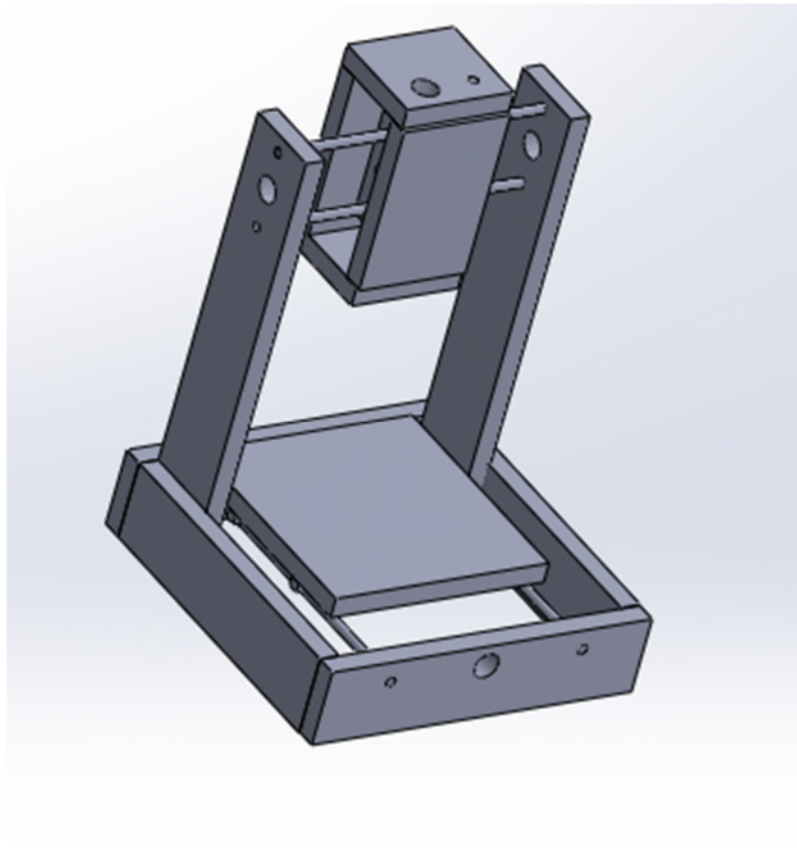
### **III.7. Système de guidage**

Pour les trois axes X Y et Z, nous avons utilisé le même système de guidage à savoir de tiges en acier inoxydable de dimensions :

- Axe X :  $\varnothing 8 \text{ mm}$ ,  $L = 264 \text{ mm}$ .
- Axe Y :  $\varnothing 10 \text{ mm}$ ,  $L = 300 \text{ mm}$ .
- Axe Z :  $\varnothing 6 \text{ mm}$ ,  $L = 168 \text{ mm}$ .

### **III.8. Assemblage de la machine**

Après avoir réalisé le châssis et les systèmes d'axes séparément, nous avons procédé à l'assemblage de la machine en insérant les systèmes d'axes dans le châssis. Ensuite, on monte les coupleurs sur les trois axes et enfin on fixe les moteurs sur les planches par des vis M5 de telle manière qu'ils entraînent les coupleurs. La figure III.8 est une photo de la machine réalisée.



**Figure III.8 :** Photo de la machine à trois axes réalisée

### **III.8. Conclusion**

La conception et le dimensionnement est une partie décisive dans notre projet, c'est pour cela qu'on a consacré ce chapitre a cette partie.

On a possédé à cette étude comme suit :

- Etude de la structure générale du prototype.
- Etude du système de guidage des axes.
- Présentation des différentes parties constituant la machine outils a 3 axes

Le commande de ces trois axes fera l'objet du 4 chapitre

# Chapitre IV

### IV.1. Introduction

Ce chapitre montre comment commander un petit moteur pas à pas avec un Arduino. On considère le cas du moteur pas à pas bipolaire. De nombreux modèles au format NEMA17 sont disponibles sur le marché, avec des pas de 1.8 ou 0.9 degrés (respectivement 200 et 400 pas par tour) et dans notre projet nous avons utilisé un moteur 1.8°.

### IV.2. Branchement des moteurs

La carte shield V 3.0 est directement placée sur la carte Arduino. Cette dernière est alimentée par une source de courant continu de 12.0 V. Les images suivantes montrent le branchement du moteur pas à pas et du driver sur la carte. [10]

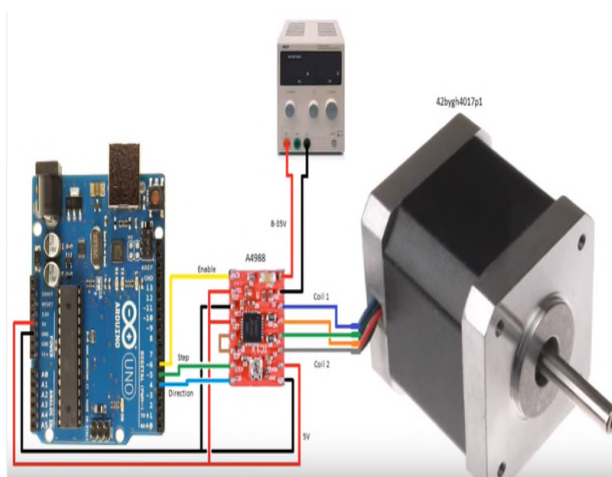


Figure IV.1 Branchement des moteurs

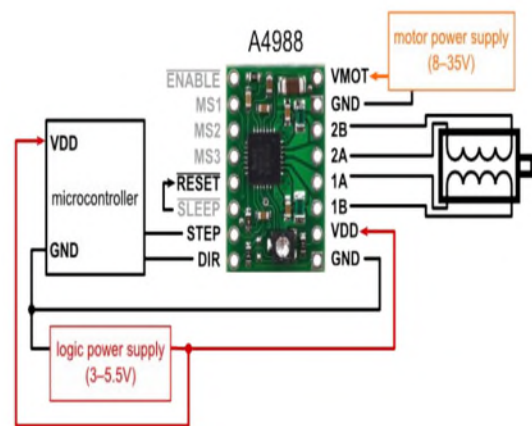


Figure IV.2 : Branchement du driver

### IV.3. Arduino IDE



Le logiciel Arduino est un environnement de développement (IDE) open source et gratuit, téléchargeable sur le site officiel Arduino. ([www.arduino.cc](http://www.arduino.cc)). C'est un logiciel développé en langage C. [17]

L'interface de l'Arduino IDE comme dans Figure IV.12 donne accès à :

- ❖ Fichier : ouvrir un programme déjà enregistré, enregistrer un programme...
- ❖ Editer : permet d'éditer l'interface du logiciel (copier, coller, taille de police).
- ❖ Croquis (sketch) : éditer un programme .
- ❖ Outils : mettre à jours la bibliothèque, choisir le type de carte et le programmeur .
- ❖ Aide : voir le catalogue du logiciel.

- ❖ Compiler : vérifier si le programme ne contient pas d'erreurs.
- ❖ Televerser : permet d'envoyer le programme à la carte Arduino .

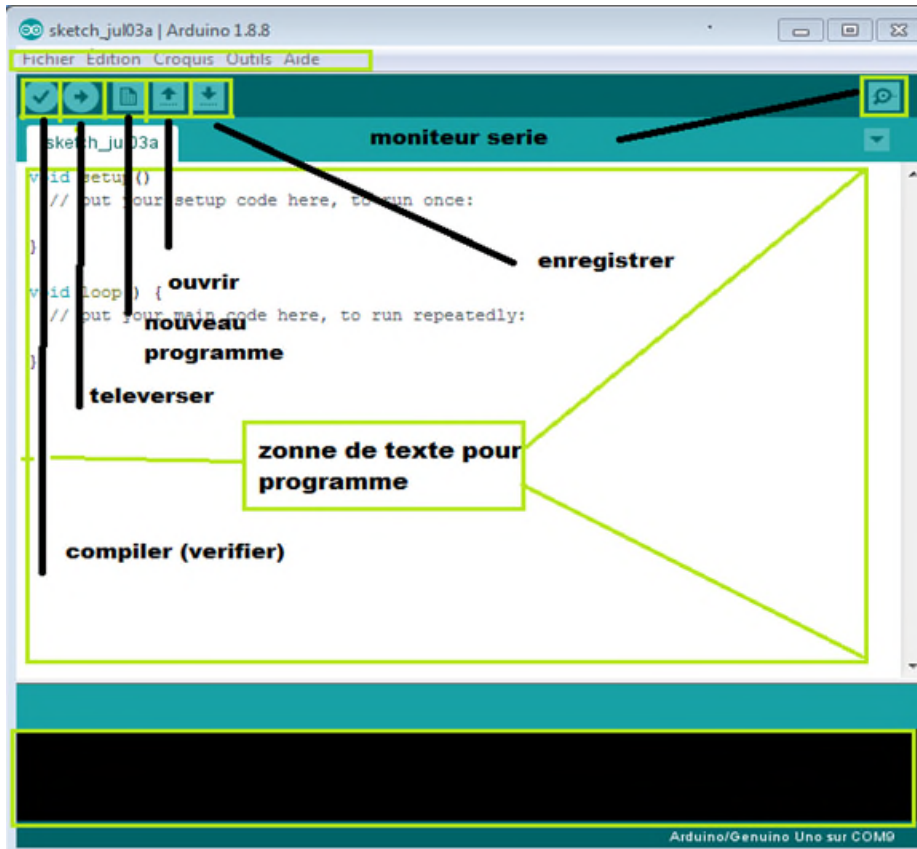
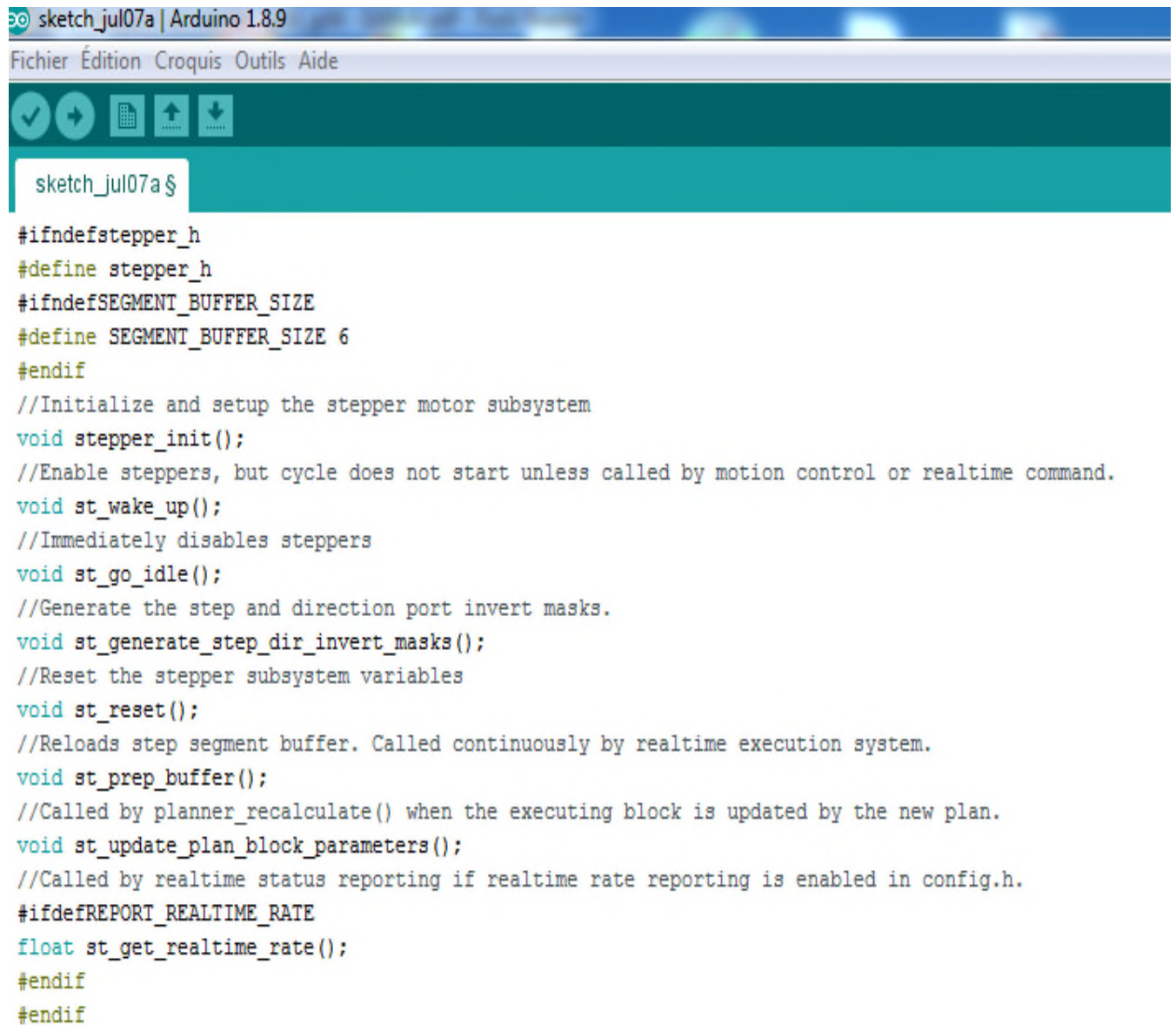


Figure IV.3 : Interface du logiciel Arduino IDE

GRBL est un programme “firmware”, exécuté sur un Arduino, qui analyse le G-Code reçu, construit un modèle mathématique des actions à effectuer et transforme le modelé en un résultat logique (Hight (1) or Low (0)), et exécute ces mouvements en envoyant un flot continu d’impulsions haute fréquences aux moteurs pas à pas. Le seul petit détail, c’est que GRBL n’a aucune interface utilisateur, Il faut alors disposer d’un autre logiciel exécuté sur une autre machine, en liaison avec Arduino, pour pouvoir faire la liaison entre l’Arduino et notre commande G-code avec la machine. Il existe plusieurs logiciels, par exemple GRBL Contrôleur. [18]

Le GRBL est un programme constitué de 37 sous-programmes qui permettent de contrôler la machine-outil à 3 trois axes. Le programme principal fait appelle à ces sous-programmes lors de l’utilisation de la commande. On donne exemple sur quelque sous-programme :

🚧 Programme de reconnaissance des moteurs pas à pas,

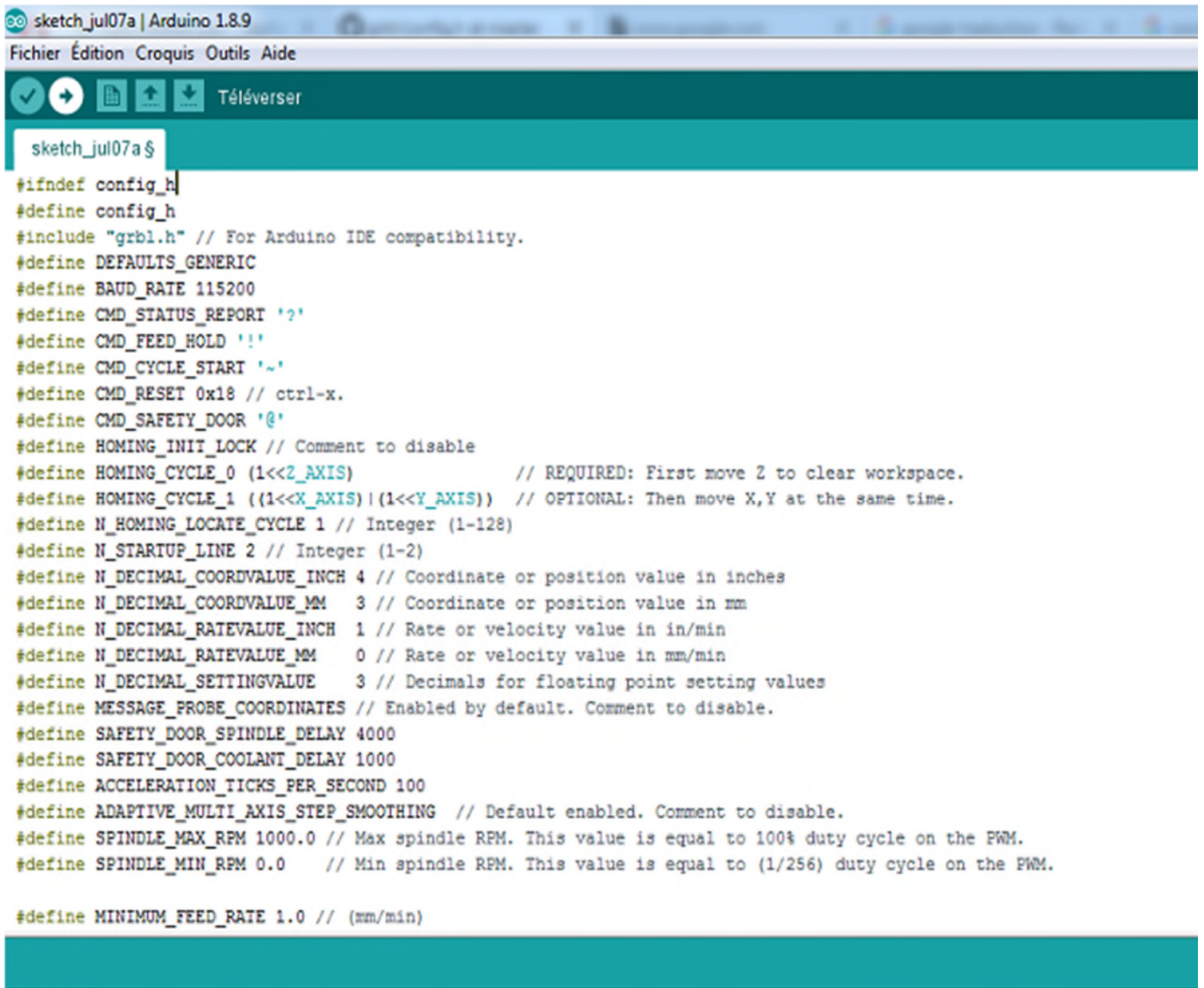


```
sketch_jul07a | Arduino 1.8.9
Fichier Édition Croquis Outils Aide
sketch_jul07a $
#ifndef stepper_h
#define stepper_h
#ifndef SEGMENT_BUFFER_SIZE
#define SEGMENT_BUFFER_SIZE 6
#endif
//Initialize and setup the stepper motor subsystem
void stepper_init();
//Enable steppers, but cycle does not start unless called by motion control or realtime command.
void st_wake_up();
//Immediately disables steppers
void st_go_idle();
//Generate the step and direction port invert masks.
void st_generate_step_dir_invert_masks();
//Reset the stepper subsystem variables
void st_reset();
//Reloads step segment buffer. Called continuously by realtime execution system.
void st_prep_buffer();
//Called by planner_recalculate() when the executing block is updated by the new plan.
void st_update_plan_block_parameters();
//Called by realtime status reporting if realtime rate reporting is enabled in config.h.
#ifdef REPORT_REALTIME_RATE
float st_get_realtime_rate();
#endif
#endif
```

**Figure IV.13** : Programme de reconnaissance du moteur pas à pas



✚ Programme de configuration des paramètres,

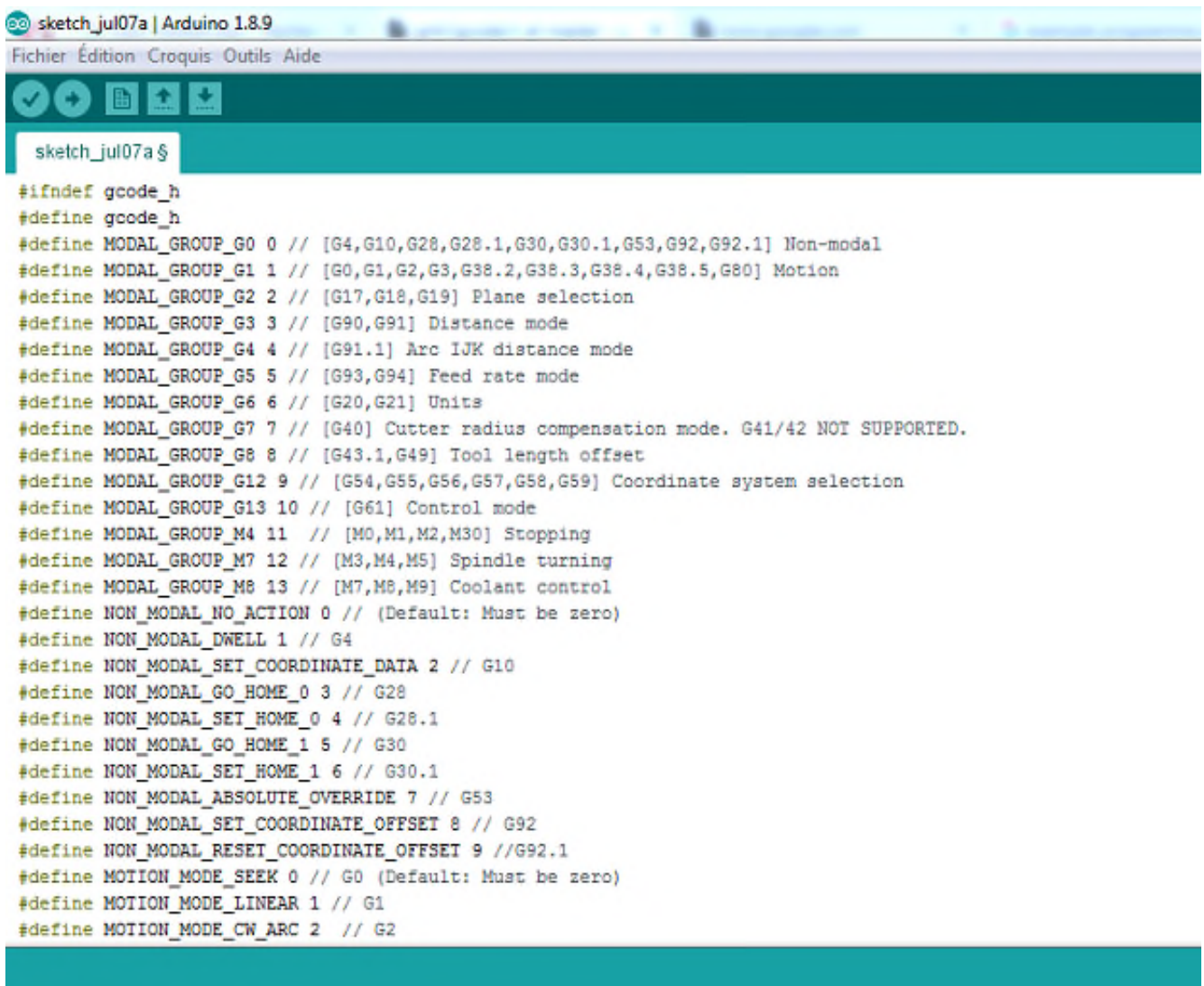


```
sketch_jul07a | Arduino 1.8.9
Fichier Édition Croquis Outils Aide
sketch_jul07a $
#ifndef config_h
#define config_h
#include "grbl.h" // For Arduino IDE compatibility.
#define DEFAULTS_GENERIC
#define BAUD_RATE 115200
#define CMD_STATUS_REPORT '?'
#define CMD_FEED_HOLD '!'
#define CMD_CYCLE_START '~'
#define CMD_RESET 0x18 // ctrl-x.
#define CMD_SAFETY_DOOR '@'
#define HOMING_INIT_LOCK // Comment to disable
#define HOMING_CYCLE_0 (1<<2_AXIS) // REQUIRED: First move Z to clear workspace.
#define HOMING_CYCLE_1 ((1<<X_AXIS)|(1<<Y_AXIS)) // OPTIONAL: Then move X,Y at the same time.
#define N_HOMING_LOCATE_CYCLE 1 // Integer (1-128)
#define N_STARTUP_LINE 2 // Integer (1-2)
#define N_DECIMAL_COORDVALUE_INCH 4 // Coordinate or position value in inches
#define N_DECIMAL_COORDVALUE_MM 3 // Coordinate or position value in mm
#define N_DECIMAL_RATEVALUE_INCH 1 // Rate or velocity value in in/min
#define N_DECIMAL_RATEVALUE_MM 0 // Rate or velocity value in mm/min
#define N_DECIMAL_SETTINGVALUE 3 // Decimals for floating point setting values
#define MESSAGE_PROBE_COORDINATES // Enabled by default. Comment to disable.
#define SAFETY_DOOR_SPINDLE_DELAY 4000
#define SAFETY_DOOR_COOLANT_DELAY 1000
#define ACCELERATION_TICKS_PER_SECOND 100
#define ADAPTIVE_MULTI_AXIS_STEP_SMOOTHING // Default enabled. Comment to disable.
#define SPINDLE_MAX_RPM 1000.0 // Max spindle RPM. This value is equal to 100% duty cycle on the PWM.
#define SPINDLE_MIN_RPM 0.0 // Min spindle RPM. This value is equal to (1/256) duty cycle on the PWM.

#define MINIMUM_FEED_RATE 1.0 // (mm/min)
```

Figure IV.14 : Programme de réglage des paramètres

Programme de définition du langage G-code dans que la carte Arduino déchiffre le code ,



```

sketch_jul07a | Arduino 1.8.9
Fichier Édition Croquis Outils Aide
sketch_jul07a $
#ifndef gcode_h
#define gcode_h
#define MODAL_GROUP_G0 0 // [G4,G10,G28,G28.1,G30,G30.1,G53,G92,G92.1] Non-modal
#define MODAL_GROUP_G1 1 // [G0,G1,G2,G3,G38.2,G38.3,G38.4,G38.5,G80] Motion
#define MODAL_GROUP_G2 2 // [G17,G18,G19] Plane selection
#define MODAL_GROUP_G3 3 // [G90,G91] Distance mode
#define MODAL_GROUP_G4 4 // [G91.1] Arc IJK distance mode
#define MODAL_GROUP_G5 5 // [G93,G94] Feed rate mode
#define MODAL_GROUP_G6 6 // [G20,G21] Units
#define MODAL_GROUP_G7 7 // [G40] Cutter radius compensation mode. G41/42 NOT SUPPORTED.
#define MODAL_GROUP_G8 8 // [G43.1,G49] Tool length offset
#define MODAL_GROUP_G12 9 // [G54,G55,G56,G57,G58,G59] Coordinate system selection
#define MODAL_GROUP_G13 10 // [G61] Control mode
#define MODAL_GROUP_M4 11 // [M0,M1,M2,M30] Stopping
#define MODAL_GROUP_M7 12 // [M3,M4,M5] Spindle turning
#define MODAL_GROUP_M8 13 // [M7,M8,M9] Coolant control
#define NON_MODAL_NO_ACTION 0 // (Default: Must be zero)
#define NON_MODAL_DWELL 1 // G4
#define NON_MODAL_SET_COORDINATE_DATA 2 // G10
#define NON_MODAL_GO_HOME_0 3 // G28
#define NON_MODAL_SET_HOME_0 4 // G28.1
#define NON_MODAL_GO_HOME_1 5 // G30
#define NON_MODAL_SET_HOME_1 6 // G30.1
#define NON_MODAL_ABSOLUTE_OVERRIDE 7 // G53
#define NON_MODAL_SET_COORDINATE_OFFSET 8 // G92
#define NON_MODAL_RESET_COORDINATE_OFFSET 9 //G92.1
#define MOTION_MODE_SEEK 0 // G0 (Default: Must be zero)
#define MOTION_MODE_LINEAR 1 // G1
#define MOTION_MODE_CW_ARC 2 // G2

```

Figure IV.15 : Programme de définition des fonctions G-code

#### IV.4. Concept du programme

Une opération typique de la machine requière le déplacement manuel par petits pas au point origine 0 pour la pièce considérée. Souvent cela demande une mise au point sur chacun des axes. C'est pour cette raison qu'il y a un bouton zéro près de chaque axe. Une fois qu'on exécute un fichier G-Code, on ne devrait plus être capable d'envoyer des commandes manuelles. Ici les deux (2) sont liés, une opération bloque l'autre. Lorsque qu'une condition d'alarme est détectée par GRBL, l'exécution du G-Code est stoppée. Les boutons de contrôle et de visualisation sont groupés en sous-panneaux qui sont activés/désactivés en fonction de ce

que le panneau de G-code Sender est en train de faire. Un onglet pour les réglages fournit un certain nombre de zones.

L'organigramme donné sur la figure IV.16 résume le mode de fonctionnement du firmware développé. Les paramètres de notre système sont : la commande de position des moteurs et le mode de récupération des données (depuis l'ordinateur ou directement de la carte SD), et l'exécution du G-Code.

Éditables qui modifient le fonctionnement du panneau, où ce qui est envoyé à GRBL. On peut aussi modifier directement les réglages internes à GRBL depuis ce même onglet.

Voilà si dessous des logigrammes qui expliquent d'une façon directe et simple le principe de fonctionnement du GRBL :

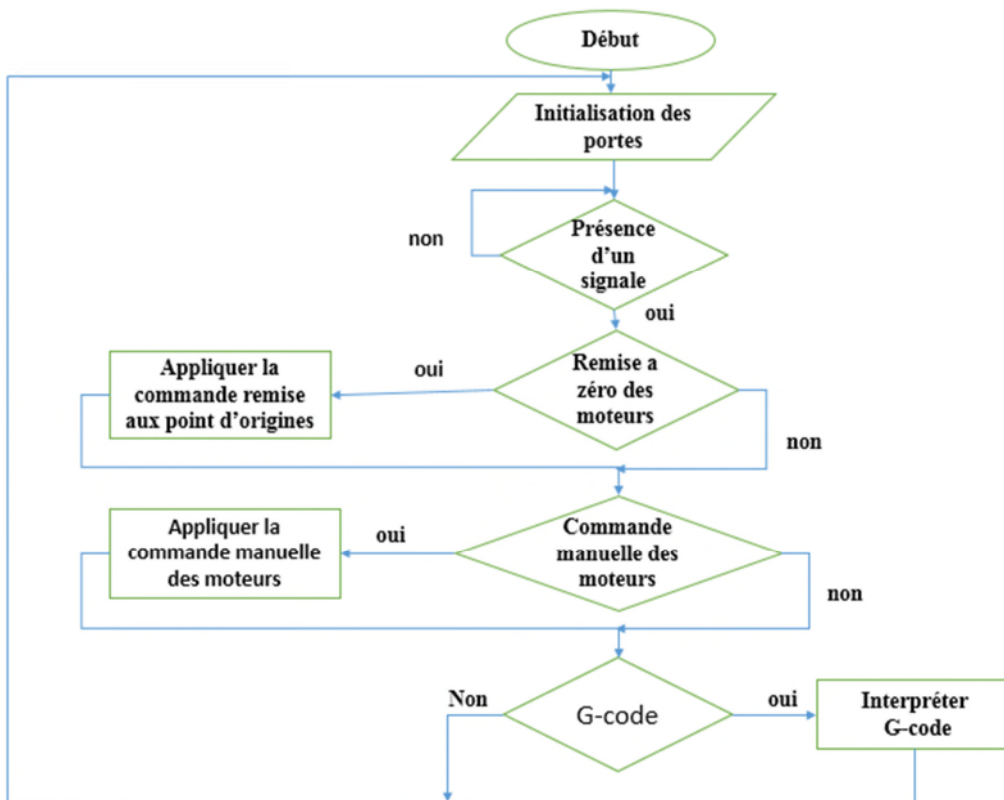


Figure IV.16 : Organigramme de fonctionnement du firmware.

### IV.5 Sous-programme de la remise aux points d'origine des moteurs

Ce dernier a le rôle de déplacer les axes à leur position initiale.

L'organigramme de la figure IV.17 résume le mode de fonctionnement du sous-programme de la commande de remise aux points d'origine des moteurs.

Au début, on définit la commande à exécuter, ensuite le firmware donne l'ordre au moteur désiré à reculer d'un pas jusqu'à l'activation du capteur fin de course de ce dernier.

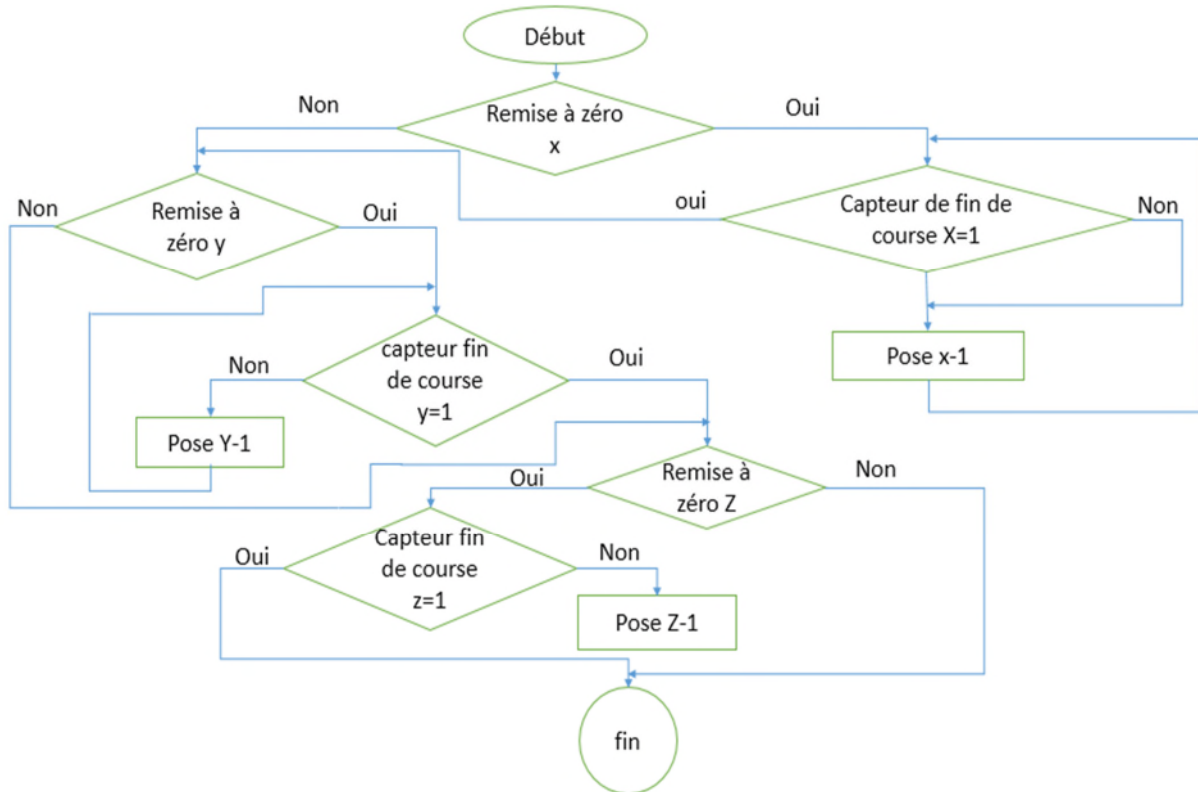


Figure V.17 : Sous-programme de la remise aux points d'origine des moteurs

### IV.6. Sous-programme de la commande manuelle des moteurs

Ce dernier a le rôle de contrôler le déplacement des moteurs des trois axes X, Y et Z manuellement.

L'organigramme de la figure IV.18 résume le mode de fonctionnement du sous-programme de la commande manuelle des moteurs. On oriente au début le moteur à la direction voulue avec sa commande et on maintient la commande jusqu'à ce qu'on atteigne le point voulu.

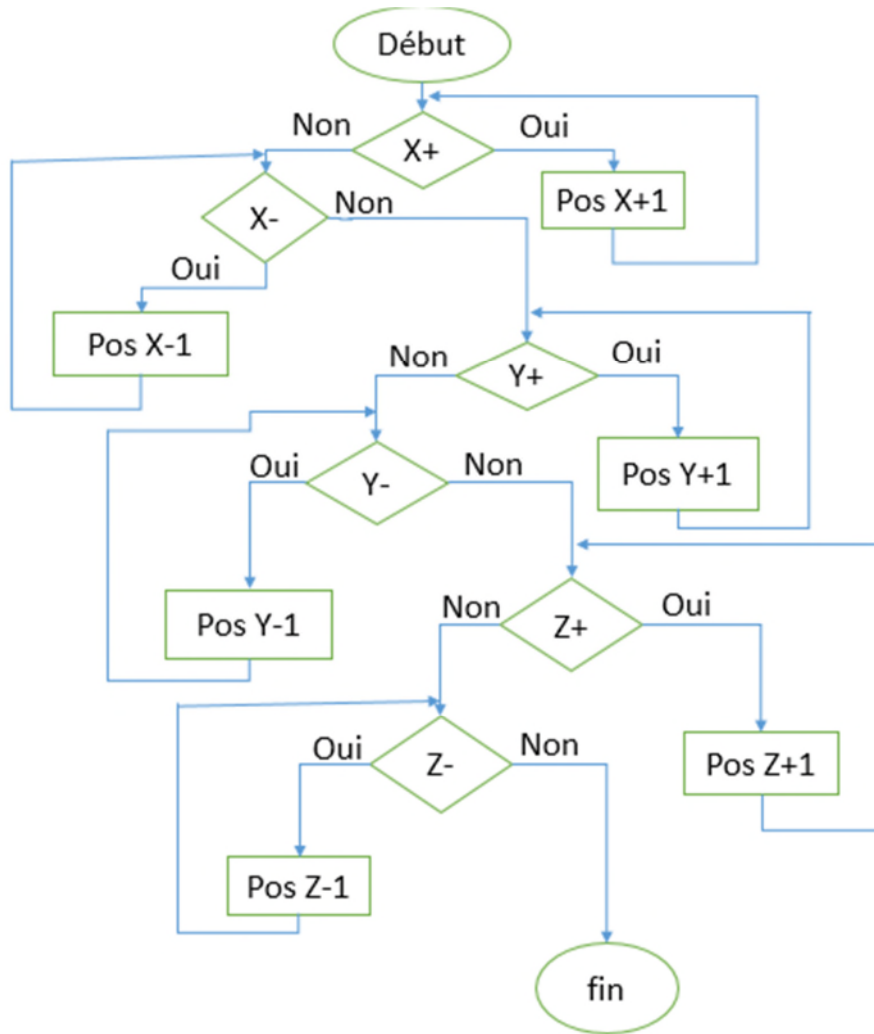


Figure IV.18 : Sous-programme de la commande manuelle des moteurs

### IV.7. Inkscape et l'extension G-code Tools



Inkscape est un logiciel libre de dessin vectoriel. C'est un outil extrêmement puissant permettant de créer des graphismes au format vectoriel SVG.

Récemment une extension créée nommée " G-codetools " a été intégrée de façon native à Inkscape. Cette extension permet de convertir des chemins obtenus dans Inkscape en langage G-code, langage permettant de piloter des machines à commandes numériques. Ces machines peuvent être des fraiseuses, des lasers. La figure IV.19 représente la fenêtre principale du logiciel Inkscape.

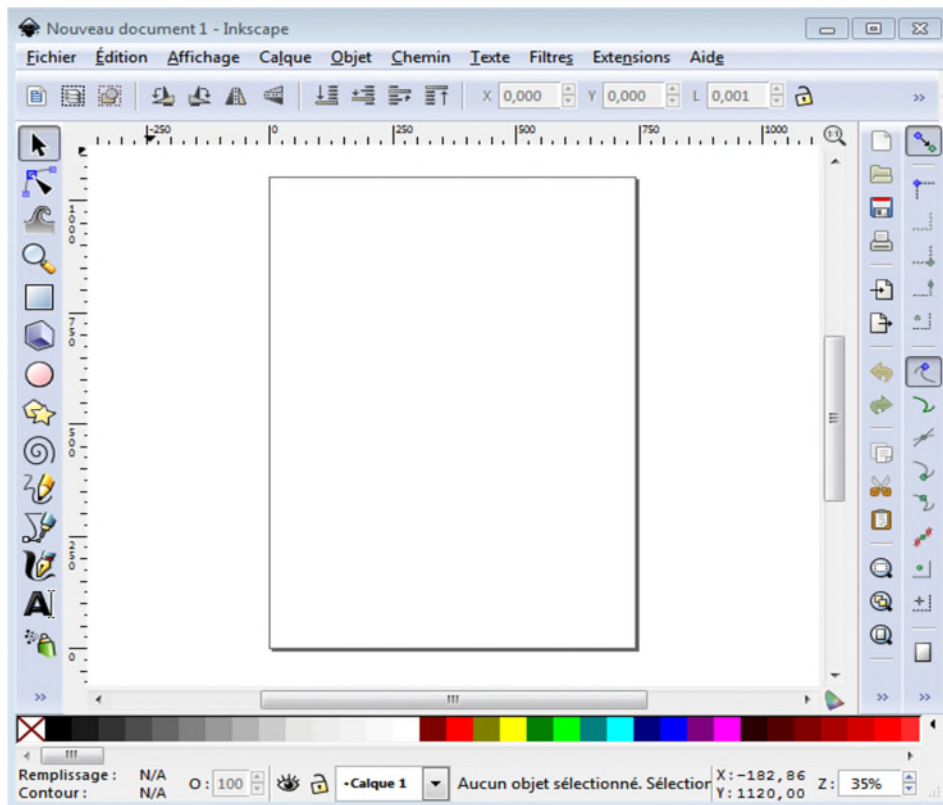


Figure IV.19 : Interface d'Inkscape

## IV.8. G-code Sender

Le Panneau de contrôle G-code Sender comme beaucoup d'autres panneaux professionnels (GRBL Contrôleur, Candel) intègre de nombreuses fonctions habituellement trouvées sur des machines industrielles. Il "clone" plus ou moins un panneau physique comme on pourrait le trouver sur une machine Fanuc ou Haas. Tous les outils nécessaires au réglage et au fraisage sont organisés de manière claire. La figure-III.20 représente l'interface de commande du logiciel G-code Sender.

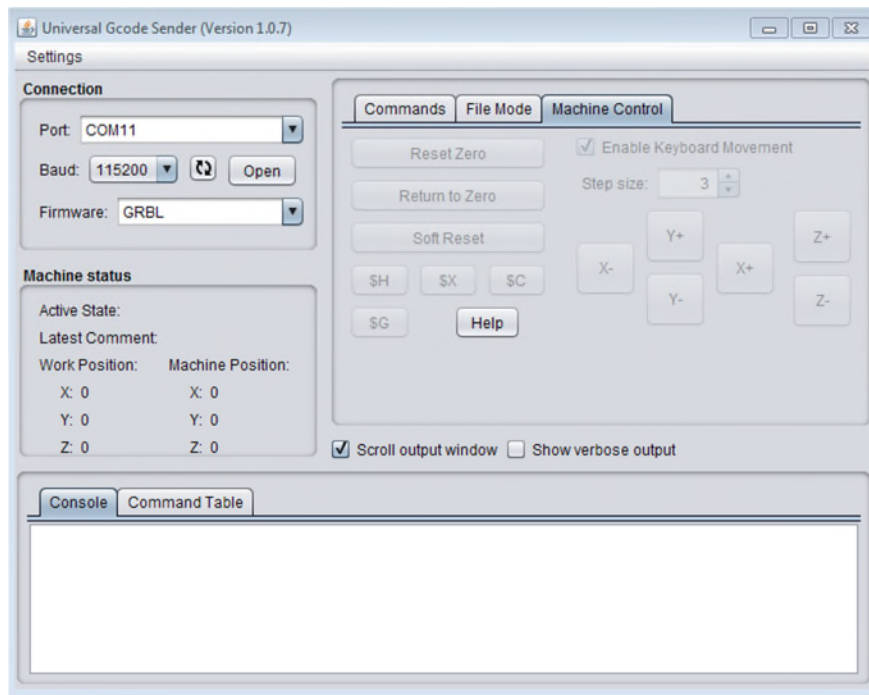


Figure IV.20 : Interface G-code Sender

## IV.9. Conclusion

La partie programmation est la partie qui met en pratique la partie électrique, elle permet de faire la communication entre la partie électrique et la partie mécanique, nous avons abordé alors dans ce chapitre :

- Le montage électrique des moteurs vers les driver
- Vue générale sur les logiciels à utiliser durant la manipulation ou de l'usage de la machine.
- Le programme qui utilise pour contrôler notre carte Arduino et les driver et la carte Shield

# **Conclusion Générale**



## Conclusion Générale :

Le projet présenté est une machine outils a trois axes on a conçu une machine outils a 3 axes. Il s'agit de l'étude et la conception des différentes parties de la machine et le développement de sa commande.

On a conçu un prototype qui s'adapte avec plusieurs fonctions (imprimante 2d, 3d fraiseuse numérique). On a utilisé un châssis en bois pour les différentes parties des châssis Et des tiges en acier inoxydable .des tige filetés en Fer (pas 0.8mm). des roulement (6000-2Z-c), des moteurs Nema 17 (1.8°, de puissance 30W) une carte Shield (v3.0) , une Arduino (UNO R3).

On a pu aussi étudier le programme GRBL qui est notre programme principal pour communiquer entre l'Arduino et notre logiciel de commande G-code Sender ainsi que le logiciel Arduino IDE qui est un logiciel avec lequel on a programmé notre carte Arduino, Inckape un logiciel de dessin qui contient l'extension qui peut commander notre machine en tant que une fraiseuse

A la fin de ce travail on a pu :

- ✓ Approfondir nos connaissances sur les Arduino et leur programmation.
- ✓ Connaitre le type des machines-outils a 3 axes et leur fonctionnement.
- ✓ Faire la conception et le dimensionnement d'une machine-outil a 3 axes
- ✓ Touché à l'aspect pratique et l'esprit de création et de travail en équipe.

Ce projet reste le premier prototype de ce type de machine pour nous qui nécessite une amélioration dans la structure esthétique et niveau programme .

# **Références bibliographiques**

- [1] Gilles prud'homme, 1995, « commande numérique des machines-outils », Technique de l'ingénieur, génie mécanique, usinage, B7130.
- [2] Gilles Patri, « Aérospatiale : une nouvelle ère industrielle. Des outils classiques à la commande numérique », dans Revue aérospatiale, N° hors-série 20 ans d'Aérospatiale, janvier 1990.
- [3] <https://bu.univ-ouargla.dz/master/pdf/Bentaleb-gasmi.pdf?idmemoire=55>
- [4] D. Gelin, M. Vincent « éléments des fabrications », Edition marketing, paris, Mars 1995.
- [5] OUANIK Fouad, « Simulation de l'usinage d'un guidage longitudinal sous le logiciel SINUTRAIN 828D ShopMill », Mémoire de Master, Université Badji-Mokhtar-Annaba, 2017.
- [6] BENBEKHTI Ahmed, « Etude de réalisation d'un support pour affutage des forêts sur une machine à commande numérique », Mémoire de Master, Université Abou Bekr Belkaid-Tlemcen, 2013. <http://dSPACE.univ-tlemcen.dz/bitstream/112/3808/1/msgm2.pdf>
- [7] <https://blog.bendinelli.net/index.php?post/2016/02/06transformer-une-d%C3%A9coupeuse-laser-avec-arduino-et-Grbl>
- [8] Patrick BLAIN, technique de l'ingénieur (CAO et méthodologie de conception). Document B 2 810
- [9] [https://fr.wikipedia.org/wiki/Fabrication\\_assist%C3%A9e\\_par\\_ordinateur](https://fr.wikipedia.org/wiki/Fabrication_assist%C3%A9e_par_ordinateur)
- [10] <http://reprap.org>
- [11] C. Barlier et A. Bernard, Fabrication additive: Du prototypage rapide à l'impression 3D. Dunod, 2015.
- [12] p.oguic, moteur pas-à-pas et pc. Dunod 2004
- [13] « Histoire de l'impression 3D | SUPINFO, École Supérieure d'Informatique ». [En ligne]. Disponible sur: <http://www.supinfo.com/articles/single/34-histoire-impression-3d>. [Consulté le: 06-janvier-2016].
- [14] ouvrage Arduino : Premiers pas en informatique embarquée [Auteurs :Simon Andrault(Eskimon) Hippolyte Weisslinger(olyte)] Édition du 01 juin 2014.

[15] BELLOUFI Abderrahim, «Machines-outils à commande numérique», Université Kasdi Merbah Ouargla, Algérie, 2010.

[16] [www.cnc-book.com/7077-les-langages-cn.html](http://www.cnc-book.com/7077-les-langages-cn.html)

[17] [www.arduino.cc](http://www.arduino.cc)

[18] [www.github.com](http://www.github.com)

## **RESUME**

Le travail présenté aborde les notions fondamentales des machines à commande numérique et leurs classifications. L'étude et la réalisation d'un prototype d'une machine à commande numérique par ordinateur et pour y parvenir on a évoqué, toutes les parties utilisées dans notre projet (moteurs utilisés, carte Shield, Arduino...Etc.). Les logiciels utilisés durant la manipulation et quelque partie du programme utilisé pour contrôler la carte Arduino

**Mots clés :** Machine à commande numérique, CNC,

## **ABSTRACT**

The work presented discusses the fundamentals of numerically controlled machines and their classifications. The study and the realization of a prototype of a numerical control machine by computer and to own it one evoked, all the parts used in our project (engines used, card Shield, Arduino ... Etc.). Software used during manipulation and any part of the program used to control the Arduino board

**Key words:** Numerically controlled machine, CNC,