



Licence Fondamentale  
Sciences de la Matière Physique  
Parcours : Electronique

Projet de Fin d'Etude

# «RÉALISATION ET COMMANDE D'UNE MACHINE CNC A BASE DES MOTEURS PAS À PAS»

Réalisé Par :

Ezouagh Youness

El Maatouk Chaimae

Encadré Par :  
Pr. Setti Larbi

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

«وَمَا أَتَيْتُهُم مِّنَ الْعِلْمِ إِلَّا قَلِيلًا»

صَدَقَ اللَّهُ الْعَظِيمُ

# Remerciement

On tient à remercier dans un premier temps, notre Dieu pour les innombrables bénédictions.

Puis nous remercions nos chers parents, frères et sœurs pour leurs soutiens moraux et financiers.

Ainsi que toute l'équipe pédagogique de la Faculté Polidisciplinaire à Larache, pour l'expérience enrichissante et pleine d'intérêt qu'elles nous ont fait vivre durant ces trois années au sein de la faculté.

Nos remerciements en particulier :

*À notre Encadreur*

*Mr SETTI LARBI*

- ✓ Pour tout le temps qu'il nous a consacré, pour ces directives pratiques, son encouragement et confiance qu'il nous a donnée et son aide à la rédaction de ce rapport.
- ✓ Puis, nos profonds remerciement pour les membres de jury qui ont accepté d'évaluer ce travail.
- ✓ Et n'oublient pas :

Tout nous chères amis qu'on a rencontrées à ce bel établissement. On n'oubliera en aucun cas le beau temps qu'on a passé avec vous.

# Dédicace

Nous dédions ce rapport avec un grand amour, sincérité et fierté à :

*Nos chers parents,*

Source de tendresse et affection pour leur soutien, leur patience, leur sacrifice et leur amour ;

*Nos frères et sœurs,*

Pour leur affection et leur encouragement qui ont toujours été pour nous des plus précieux et à tous membre de nos famille ;

*Tous nos amis,*

A qui je souhaite un avenir radieux plein de réussite

A tous *nos professeurs* de la faculté Polydisciplinaire de Larache et spécialement du département de physique.

*Et*

À tous qui est contribué de près et de loin à réaliser ce modeste travail.

## Table des matières

Résumé .....	1
INTRODUCTION GENERALE .....	2
I. Chapitre d'Etude Théorique Des Moteurs Pas à Pas .....	3
INTRODUCTION : .....	3
A. GENERALITES : .....	6
B. TECHNOLOGIE DES MOTEURS PAS A PAS : .....	7
1. Moteurs pas à pas rotatifs : .....	7
a) Moteur pas à pas rotatif à reluctance variable : .....	7
b) Moteur pas à pas rotatif à aimant permanent : .....	9
(1) Les moteurs unipolaires : .....	10
(2) Les moteurs bipolaires : .....	12
c) Moteur pas à pas rotatif hybrides : .....	14
2. Moteurs pas à pas linéaires : .....	16
a) Moteur pas à pas linéaire à reluctance variable : .....	16
b) Moteur pas à pas linéaire polarisés : .....	16
(1) Les moteurs à aimant permanent : .....	16
(2) Les moteurs hybrides: .....	16
C. Comparaison entre les différents types des moteurs pas à pas.....	17
D. Etude du courant dans un enroulement (phase) du moteur .....	18
E. Commande d'un moteur pas-à-pas : .....	19
F. Fonctionnement d'un moteur pas-à-pas 4 phases :.....	19
CONCLUSION : .....	22
II. Chapitre d'Etude Technique du commande numérique par calculateur (CNC).....	23
INTRODUCTION : .....	24
A. Historique .....	25
B. Principe de fonctionnement d'une machine numérique .....	25
1. La commande numérique par calculateur(CNC) : .....	27
2. La Conception Assistée par Ordinateur (CAO) .....	27
3. La Fabrication Assisté par Ordinateur (FAO).....	27
4. Le post-processeur .....	28
C. Les éléments de la partie commande .....	29
1. La carte de commande ARDUINO : .....	29
a) Mise en œuvre de la carte ARDUINO UNO :.....	29
b) Présentations et caractéristiques du module ARDUINO UNO : .....	29

2. Stepper motor driver A4988 :	31
a) Définition :	31
b) Fonctionnement :	31
3. La carte SHIELD CNC:	33
a) Définition:	33
<b>D. Programmation Et Interface graphique</b>	34
1. Langue de contrôleurs numériques « G-code » .....	34
a) Description .....	34
b) Format d'une ligne .....	35
2. L'éditeur du texte ou image compilateur de langage CNC : .....	35
a) Définition Inkscape .....	35
3. Logiciel de pilotage de la machine CNC (Universal G-code Sender) .....	36
a) Modes de commande .....	37
b) Mode Configuration.....	39
c) Mode Manuel.....	39
d) Mode automatique .....	41
4. Langage de programmation la machine CNC.....	41
a) Programme Grbl.....	41
(1) La communication avec la carte ARDUINO .....	42
(2) code de l'opération .....	42
(3) Bibliothèques de la commande des moteurs.....	43
5. Organigramme de la commande de la machine CNC. ....	43
Conclusion .....	45
<b>III. Chapitre de Simulation et Réalisation de la machine CNC</b> .....	46
<b>INTRODUCTION :</b> .....	47
<b>A. Simulation électrique sous Porteuse Professionnel</b> .....	48
1. Le logiciel PROTEUS .....	48
2. ISIS.....	48
3. Simulation et test du programme .....	49
a) Le programme de commande.....	50
b) Le logiciel de commande « Universel G-code Sender » .....	51
c) Virtual Serial Port Emulateur .....	51
d) Lancement de la simulation.....	52
<b>B. MISE EN MARCHE DE LA MACHINE</b> .....	53
1. Les composantes .....	53

2. La Réalisation .....	54
3. Difficultés et problèmes .....	60
Conclusion : .....	61
Conclusion générale .....	62
Bibliographie.....	63

## **ABREVIATIONS**

DSP	Digital Signal Processor
UAL	Unité Arithmétique et Logique
CNC	Commande Numérique par Calculateur
Om	Origine machine
Xm	abscisse outil machine
Ym	ordonnée outil machine
Zm	hauteur outil machine
Op	Origine pièce
OP	Origine Programme
CAO	Conception Assisté par Ordinateur
FAO	Fabrication Assisté par Ordinateur
DFN	Définition de Formes Numérisées
CFAO	Conception, Fabrication Assisté par Ordinateur
MOCN	Machine-Outil à Commande Numérique
RAM	Mémoire vive ou Mémoire PC (Random Access Memory)
ROM	Mémoire morte (Read Only Memory)
EEPROM	Mémoire morte effaçable électriquement et programmable (Electrically Erasable Read Only Memory)
SCI	Interface de communication série (serial communication interface)
STEP	Standard pour l'échange de données de produit (STandard for the Exchange of Product model data en anglais)
ISO	Organisation internationale de normalisation (International Organisation for Standardization)
DXF	Drawing eXchange Format
IGES	Initial Graphics Exchange Specification

## Table Des Figures

Figure 1 . Imprimante 3D .....	4
Figure 2. Vue d'ensemble d'un moteur pas à pas.....	6
Figure 3. Structure classique d'un moteur à reluctance variable rotatif à double saillance.....	8
Figure 4. Structure interne d'un moteur à aimant permanent .....	9
Figure 5. Fonctionnement schématique d'un moteur pas à pas .....	10
Figure 6. Représentation schématique d'un moteur unipolaire .....	11
Figure 7. Séquence de rotation simple .....	11
Figure 8. Représentation schématique d'un moteur bipolaire .....	12
Figure 9. Séquences de commande d'un moteur bipolaire.....	13
Figure 10. Moteur pas à pas hybride .....	14
Figure 11. Structure interne d'un moteur hybride .....	14
Figure 12. Principe d'un moteur pas-à-pas .....	19
Figure 13. Flux induit de moteur pas à pas .....	19
Figure 14. Position de moteur pas-à-pas .....	20
Figure 15. Chronogramme des étapes de moteur pas-à-pas .....	21
Figure 16. Décomposition d'une machine numérique. ....	25
Figure 17. Machine à trois axes.....	26
Figure 18. Cycle d'une pièce fabriquée par une commande numérique (CFAO) .....	28
Figure 19. Carte ARDUINO UNO.....	30
Figure 20. Circuit de contrôle d'un moteur a l'aide du A4988 .....	32
Figure 21. Les différentes commandes pour le mode pas complet.....	33
Figure 22. SHIELD CNC avec quatre A4988.....	33
Figure 23. L'interface principale d'Inkscape.....	36
Figure 24. Universal G-code Sender.....	37
Figure 25. l'interface de "G-code Sender " on mode manuel.....	37
Figure 26. l'interface de "G-code Sender " choisir le fichier G-code.....	38
Figure 27. envoyé LE fichée G-code .....	39
Figure 28 . mode manuel. ....	40
Figure 29. GRBL code.....	42
Figure 30 . ProcessCommand() fonction .....	43
Figure 31. Organigramme de la commande de la machine CNC. ....	44
Figure 32. Logiciel ISIS. ....	49
Figure 33.Montage électronique dans Proteus. ....	49
Figure 34. Lancement du GrblUpload. ....	50
Figure 35. Chargement du programme dans l'ARDUINO. ....	50
Figure 36. Commande en mode manuel.....	51
Figure 37. Interface de VSPE. ....	51
Figure 38. Lancement de la simulation on ISIS .....	52
Figure 39. Visualisation du processus d'envoi du G-code.....	52
Figure 40. Arduino UNO .....	54
Figure 41. Moteur 28BYJ-48 .....	54
Figure 42. Servomoteur.....	54
Figure 43. CNC Shield .....	54
Figure 44. Le pilot A4988 du moteur.....	54

Figure 45. Source d'alimentation .....	54
Figure 46. Réalisation sur Plaque d'essaie .....	55
Figure 47. Première essaie de réalisation .....	55
Figure 48. Premier résultat de dessin .....	55
Figure 49. Les nouveaux chemins des axe (X, Y) .....	56
Figure 50. Le montage final du machine CNC .....	56
Figure 51. Les buttons du test des axes X, Y et Z .....	58
Figure 52. L'envoi du fichier G-code .....	58
Figure 53. Exemple dessin cnc machine.....	59
Figure 54. Les résultats de dessin de la machine .....	59

## Table Des Tableaux

Tableau 1. Ordre des différentes commandes pour moteur à reluctance variable .....	9
Tableau 2. Ordre des différentes commandes pour moteur unipolaire .....	11
Tableau 3. Ordre des différentes commandes pour moteur bipolaire .....	12
Tableau 4. Tableau de comparaison entre les différents types des moteurs pas à pas.....	17
Tableau 5. Table de vérité des étapes de moteur pas à pas.....	21

## Résumé

Les machines à commande numérique par calculateur (CNC) permettent une production économique et rentable. Le contrôle des coûts reste une préoccupation importante, La diminution des quantités des séries et le raccourcissement de la longévité des produits finaux demandent des déroulements de production de plus en plus flexibles ; l'élément principal de ces machines est le moteur pas à pas.

L'application de ces moteurs est très répandue dans les systèmes industriels et automatiques. Le moteur pas à pas est l'interface idéale entre l'électronique numérique et la mécanique, il permet de convertir directement un signal électrique en un positionnement angulaire à caractère incremental. Notre objectif est de réaliser une machine à commande numérique par un calculateur (CNC) à base des moteurs pas à pas.

# INTRODUCTION GENERALE

Ce travail représentera notre projet de fin d'étude pour l'obtention de licence d'étude fondamentale (LEF), à l'université Polydisciplinaire de Larache.

L'objectif de ce projet est de réaliser une machine à commande numérique par un calculateur (CNC) capable de dessiner des objets.

En effet, actuellement, les systèmes à commande numérique représentent le moyen de production le plus important.

Concevoir la partie commande de cette machine fait appel à des algorithmes d'interpolation, commande des moteurs pas à pas, asservissement, interfaçage Homme-Machine...

Notre objectif est d'avoir l'initiative de proposer une conception ainsi que la réalisation d'un prototype basé sur les moteurs pas à pas.

Nous nous intéressons dans la première partie de notre mémoire à l'étude théorique des moteurs pas à pas.

Dans la seconde partie, nous allons élaborer l'étude technique de notre commande numérique puis nous allons expliquer le choix des différents éléments utilisés (microcontrôleur et carte de commande des moteurs pas-à-pas...).

Ensuite, nous allons expliquer le fonctionnement de la machine en expliquant le programme implémenté dans L'ARDUINO et la partie de supervision sur l'ordinateur.

Enfin, nous allons entamer La partie Simulation et Réalisation de la machine CNC.

# I. Chapitre d'Etude Théorique

## Des Moteurs Pas à Pas



# INTRODUCTION :

Les moteurs pas à pas sont des transducteurs électromécaniques qui assurent la conversion des signaux électriques digitaux ou impulsionnels en mouvement de rotation ou de translation de type incrémental.

Les moteurs pas à pas, utilisés pratiquement dans tous les composants d'un système informatique, sont très mal connus, surtout en ce qui concerne leur mode de commande. Ils sont utilisés aussi bien dans les lecteurs de disquettes, les disques durs, que dans les scanners et les imprimantes 3D conçues pour le prototypage rapide (figure 1).

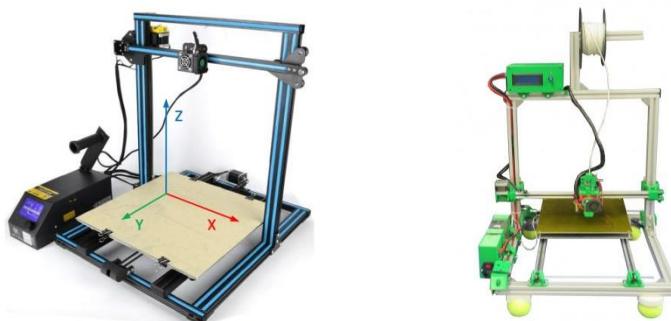


Figure 1 . Imprimante 3D

Dans le domaine de la robotique, ils sont la clef de voûte du système. Ils permettent d'obtenir une précision extraordinaire grâce à leur conception électrique et mécanique.

Le moteur pas à pas a été conçu à partir de deux démarches logiques très différentes:

- d'une part, on a cherché un moteur capable de développer un couple important à faible vitesse, voire même à l'arrêt ;
- d'autre part, on a étudié un dispositif capable de convertir des informations de caractère discret.

Le convertisseur d'énergie à basse vitesse et le transmetteur de l'information sont devenu un moteur pas à pas moderne vers les années 1970, grâce au développement conjugué de l'électronique de puissance et, surtout, grâce à l'apparition de l'électronique numérique à forte intégration.

Le moteur pas à pas est actuellement le principal élément intermédiaire entre les dispositifs de traitement d'information et le monde électromécanique extérieur. Par ailleurs, ses capacités à contrôler la position et la vitesse, par un train d'impulsions de commande, assurent à ce

convertisseur des applications comme :

- la traction des robots mobiles,
- le fonctionnement en moteur couple de grande puissance,
- l'indexage rotatif ou linéaire.

Dans sa version classique, le moteur pas à pas est alimenté à partir d'une source (de courant ou de tension) continue et le contrôle de la vitesse ou/et de la position s'effectue en boucle ouverte.

Le pilotage en boucle ouverte, qui constitue un des principaux avantages du moteur pas à pas, aussi bien du point de vue économique (coût et fiabilité d'installation) que fonctionnel (commande naturelle par "tout ou rien" à partir d'une horloge séparée ou intégrée dans un microprocesseur), présente un certain nombre d'inconvénients comme :

- la limitation du couple de démarrage,
- des instabilités de fonctionnement à certaines fréquences,
- des accélérations relativement modestes,

Maintenant, ce type de moteurs est utilisé de plus en plus dans les machines à commande numérique (CNC), leurs tailles ainsi que leur puissance ont augmenté.

Dans ce chapitre nous aborderons les différents types de moteurs pas à pas, ainsi que leur mode de commande.

## A. GENERALITES :

Ils existent trois types de moteurs pas à pas : les moteurs à aimant permanent, les moteurs à réductance variable ainsi que les moteurs hybrides. Nous verrons plus tard que les moteurs à aimants permanents se subdivisent en deux catégories.

Malgré les différences existant entre les moteurs, le résultat recherché est l'avance d'un seul pas, c'est-à-dire la rotation de leur axe suivant un angle déterminé à chaque impulsion que l'une ou l'autre de leurs bobines recevra. Cet angle qui varie selon la constitution interne du moteur, est en général compris entre  $0.9^\circ$  et  $90^\circ$ .

Les moteurs les plus couramment rencontrés présentent des pas de :

- $0.9^\circ$  soit 400 pas par tour.
- $1.8^\circ$  soit 200 pas par tour.
- $3.6^\circ$  soit 100 pas par tour.
- $7.5^\circ$  soit 46 pas par tour.
- $15^\circ$  soit 24 pas par tour.

Il est évident que les moteurs pas à pas, de par leurs technologies, présentent une très grande précision et une durée de vie quasi illimitée, l'usure mécanique étant pratiquement inexisteante (absence de frottement). La figure 2 en représente l'aspect externe. Leur domaine de prédilection sera donc ceux où la précision est de rigueur : les constituants mécaniques de l'informatique et la robotique.

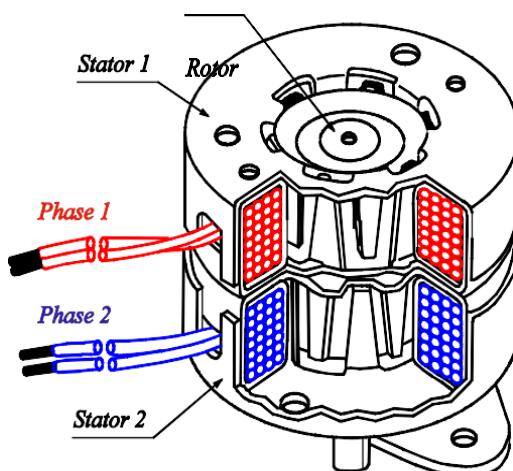


Figure 2. Vue d'ensemble d'un moteur pas à pas

Les moteurs pas à pas existent entre différentes tailles qui varient entre 1 cm et plus d'une dizaine de centimètres. Tout dépendra des applications dans lesquelles ils seront utilisés. Le plus petit moteur, par exemple, sera destiné au déplacement des têtes de lectures dans les disques durs ou un couple très faible est requis. Par contre le déplacement du bras d'un robot ou un chariot (machine CNC) demandera un couple nettement plus important, donc un moteur de diamètre élevé.

Signalons que le couple est exprimé en kilogramme par centimètre ( $\text{kg.cm}^{-1}$ ), ce qui définit le poids en kilogrammes que pourra soulever l'axe d'un moteur pourvu d'un bras de longueur exprimée en centimètre.

La valeur de leur tension d'alimentation varie dans de grandes proportions, elle peut être comprise entre 3V et plusieurs dizaine de volts. De même, selon la résistance ohmique de leurs bobinages, le courant consommé s'étendra dans une gamme allant de quelques dizaines de milliampères à plusieurs ampères.

En simplifiant, nous pourrons dire que plus le courant sera élevé, plus le couple sera important.

## B. TECHNOLOGIE DES MOTEURS PAS A PAS :

Un actionneur électrique peut créer deux types de mouvements : un mouvement de rotation ou un mouvement de translation.

### 1. Moteurs pas à pas rotatifs :

Les moteurs pas à pas peuvent être classés en fonction du phénomène physique qui est à l'origine de leur mouvement. On distingue principalement, deux catégories de moteurs pas à pas : les moteurs à reluctance variable et les moteurs polarisés.

#### a) *Moteur pas à pas rotatif à reluctance variable :*

La rotation d'un moteur à reluctance est engendrée par la réaction, entre un champ magnétique statorique et un rotor saillant, qui conduit à une disposition alignée de la partie saillante avec le pôle créé par le champ magnétique. Dans ces conditions le flux est maximum et la réluctance du circuit magnétique est minimum.

Ce type de moteur est caractérisé par une structure dentée au niveau du rotor et du stator. Le rotor est fabriqué en acier doux non magnétique. Le nombre de bobines dans le stator et le type de connexion déterminent le nombre de phases du moteur.

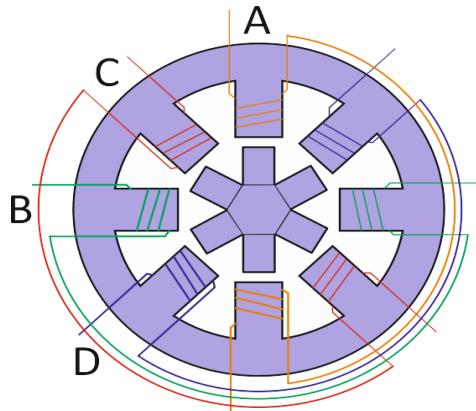


Figure 3. Structure classique d'un moteur à relucance variable rotatif à double saillance

La figure 3 présente un moteur pas à pas réluctant comportant huit plots au stator et six dents au rotor ce qui donne un moteur à quatre phases.

Chaque couple de deux bobines statoriques diamétralement opposées et connectées en série, constitue une des quatre phases de la machine considérée. L'alimentation d'une phase crée un couple permettant le déplacement du rotor vers une position d'équilibre qu'il garde tant que l'alimentation est maintenue. Cet état d'équilibre correspond à une position alignée entre les dents statoriques porteuses de la phase alimentée et les dents rotoriques. Chaque nouvelle séquence d'alimentation établit un nouvel équilibre. Le moteur se déplace donc avec un pas angulaire de 15°.

Le mode de commande peut dans ce cas, de la même façon que pour les autres moteurs, être monophasé, biphasé ou demi-pas.

Les séquences de commandes sont présentées dans le tableau 1.

Mode monophasé	Mode biphasé	Mode demi-pas
A	AC	A
B	CB	AC
C	BD	C
D	DA	BC
		B
		BD
		D
		DA

Tableau 1. Ordre des différentes commandes pour moteur à reluctance variable

Pour augmenter la résolution angulaire de ces moteurs, des constructions à circuits magnétiques multiples, peuvent être envisagées.

Ces moteurs se présentent principalement sous deux formes :

- Un empilage de plusieurs machines monophasées identiques assemblées mécaniquement sur le même arbre.
- Un stator unique, possédant plusieurs rotors magnétiquement indépendants, mais couplés mécaniquement.

Dans toutes ces machines, le rotor et les stators présentent le même nombre de dents, le mouvement étant obtenu par le décalage mécanique des rotors ou des stators, les uns par rapport aux autres.

### b) Moteur pas à pas rotatif à aimant permanent :

Les moteurs à aimant permanent sont constitués d'un stator supportant les bobinages et d'un rotor magnétique (aimant bipolaire). Cette catégorie de moteurs se subdivise en deux types : le moteur unipolaire et le moteur bipolaire.

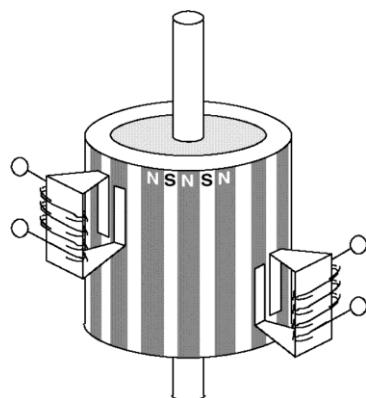


Figure 4. Structure interne d'un moteur à aimant permanent

La figure 5 représente le schéma simplifié d'un moteur à aimant. Le premier dessin de la figure le représente dans sa position de repos, lorsque les bobinages ne sont traversés par aucun courant. Chacun des pôles du rotor se place devant une paire des plots du stator.

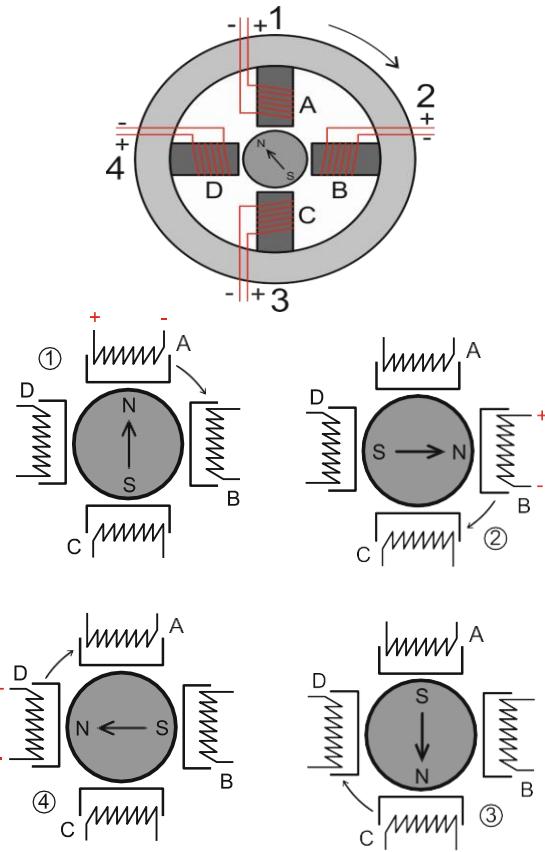


Figure 5. Fonctionnement schématique d'un moteur pas à pas

Les quatre dessins du bas de la figure illustrent ce qui se passe lorsque les bobinages sont alimentés à tour de rôle : d'abord A, puis B, puis C et enfin D. Le pôle nord du rotor sera attiré par le pôle sud du stator, pôle créé par la circulation d'un courant dans le bobinage.

Cet exemple permet de comprendre la progression pas par pas du moteur. Ici, il effectuera quatre pas par tours.

### (1) Les moteurs unipolaires :

Une représentation schématisée d'un moteur d'un moteur unipolaire est donnée en figure 6. Afin d'inverser le sens du courant, les enroulements sont réalisés au moyen de deux fils dont l'une des extrémités est reliée au + ou au - de l'alimentation. La commande de ce type moteur est la plus simple de tous les moteurs pas à pas puisqu'il suffira d'alimenter les bobinages à tour de rôle pour faire tourner l'axe d'un pas. Le schéma de la figure 7 résume la séquence la plus simple.

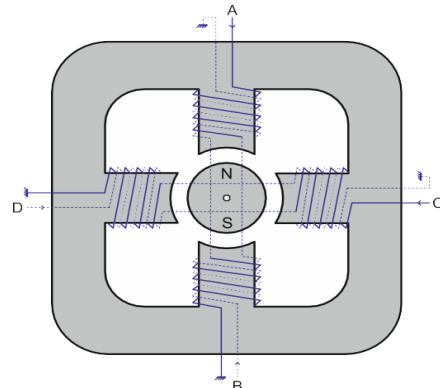


Figure 6. Représentation schématique d'un moteur unipolaire

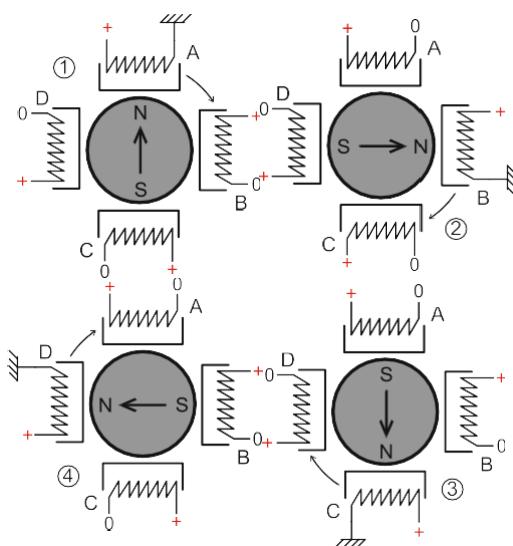


Figure 7. Séquence de rotation simple

Comme pour chaque type de moteur, le modèle unipolaire peut être commandé en mode monophasé, biphasé ou demi-pas. Le tableau 2 donne l'ordre des différentes commandes.

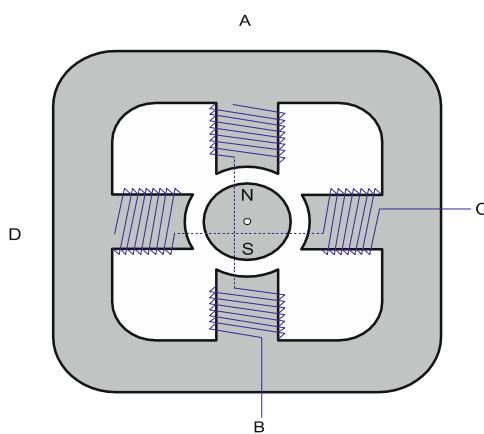
Mode monophasé	Mode biphasé	Mode demi-pas
<b>0001</b>	0101	0001
<b>0100</b>	0110	0101
<b>0010</b>	1010	0100
<b>1000</b>	1001	0110
		0010
		1010
		1000
		1001

Tableau 2. Ordre des différentes commandes pour moteur unipolaire

Signalons que le moteur unipolaire présentera, à volume égal, un couple moins important que le moteur bipolaire.

## (2) Les moteurs bipolaires :

La figure 8 représente la constitution interne d'un moteur de type bipolaire. Ce type de moteur nécessite une commande plus complexe que celle du moteur unipolaire, le courant devant changer de sens dans les enroulements à chaque pas effectué.



*Figure 8. Représentation schématique d'un moteur bipolaire*

Comme pour le modèle précédent, ce moteur peut être alimenté sous trois séquences différentes (tableau 3), représentées par ailleurs sur la figure 9.

Mode monophasé	Mode biphasé	Mode demi-pas
<b>AB</b>	AB-CD	AB
<b>CD</b>	BA-CD	AB-CD
<b>BA</b>	BA-DC	CD
<b>DC</b>	AB-DC	BA-CD
<b>AB</b>	AB-CD	BA
<b>Etc ..</b>	etc..	BA-DC
		DC
		AB-DC
		AB
		Etc ..

*Tableau 3. Ordre des différentes commandes pour moteur bipolaire*

Dans le mode monophasé, le couple n'est pas très important puisqu'un seul enroulement est alimenté pour effectuer un pas.

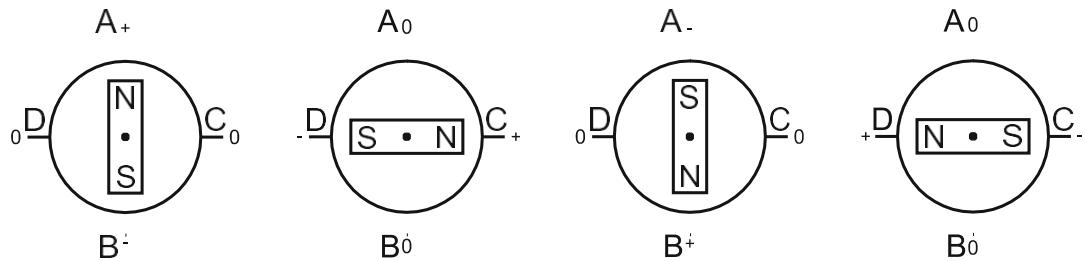
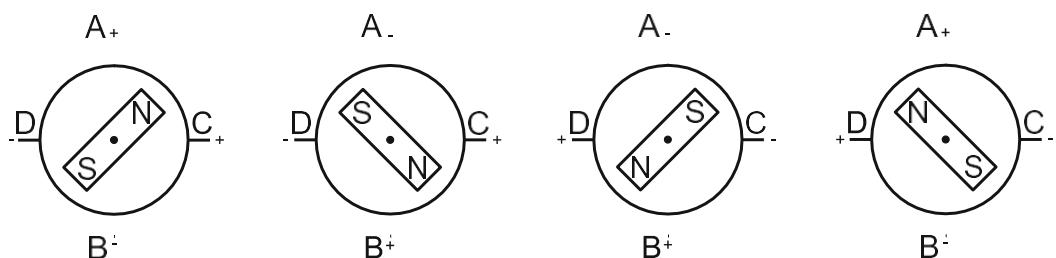
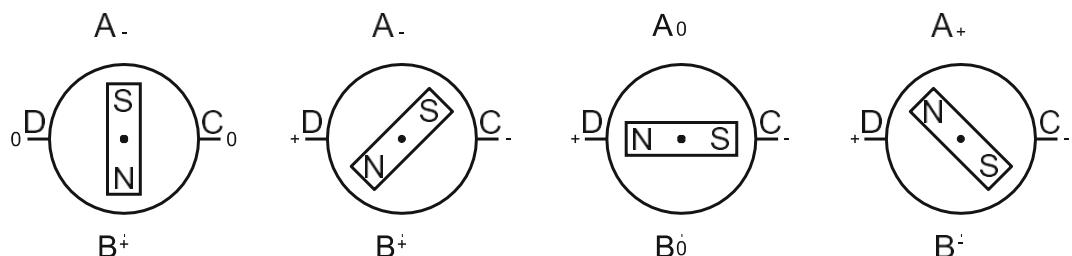
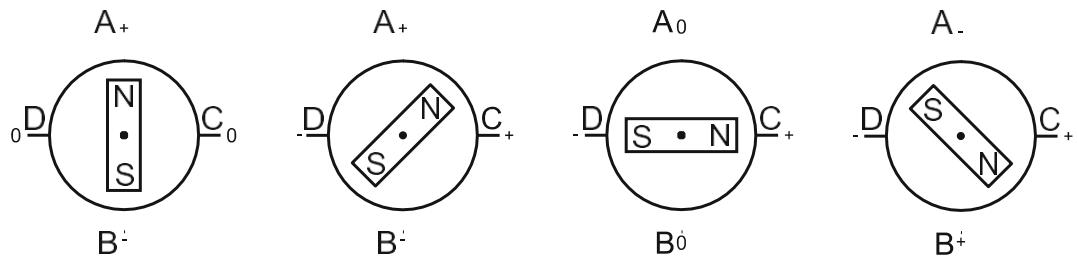
*Mode monophasé**Mode biphasé**Mode demi-pas*

Figure 9. Séquences de commande d'un moteur bipolaire

C'est dans le mode biphasé que le moteur développera la plus grande puissance (couple élevé) car les deux phases seront alimentées en même temps.

Le mode demi-pas permet de doubler le nombre de pas qu'un moteur peut effectuer par tour, mais cette précision supplémentaire engendre un couple irrégulier. En effet, dans ce mode, la commande du moteur est un mélange de mode biphasé et monophasé. Dans ce cas, si la charge est importante, des pas risquent de « sauter », ce qui n'est pas le but recherché.

### c) Moteur pas à pas rotatif hybrides :

Les moteurs pas à pas hybrides sont généralement constitués d'un rotor denté muni d'aimants permanents. Le rotor possède généralement deux disques polaires décalés d'un angle électrique  $\pi$ .

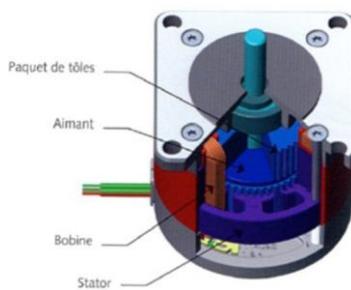


Figure 10. Moteur pas à pas hybride

Ce type de moteur présente à la fois les avantages du moteur à aimant permanent qui possède un couple élevé et ceux du moteur à réductance variable qui permet d'obtenir un nombre importants de pas par tour. Toutefois l'inertie d'un tel rotor ainsi que les pertes fer sont relativement importantes et pénalisent donc cette structure.

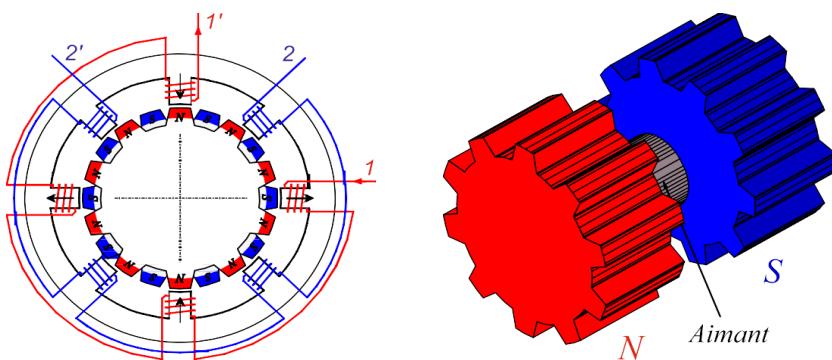


Figure 11. Structure interne d'un moteur hybride

Les moteurs pas à pas hybrides réunissent, au moins en partie, les avantages des moteurs pas à pas à réductance variable et à aimants permanents, à savoir :

- Un grand nombre de pas par tour.

- Une fréquence propre mécanique importante.
- Un couple massique élevé.
- Un amortissement interne important.
- Une mémoire de position.

Dans sa configuration de base le moteur pas à pas hybride comporte un stator en fer feuilleté à plots saillants et deux couronnes rotoriques dentées en matériau ferromagnétique, géométriquement identiques et réunies par un aimant permanent cylindrique magnétisé axialement. Les lignes de champs de l'aimant se ferment à travers les dents du rotor. Vu du stator, le rotor présente autant de pôles magnétiques actifs qu'il possède de dents. Les dents sur une des couronnes sont décalées par rapport aux dents de

l'autre d'un demi-pas dentaire  $1/2\tau$ .

$\frac{dr}{d\theta}$

Le nombre de pôles vu au stator est lié au nombre de dents d'une couronne rotorique par la relation

$$p = N_{dr} \quad 1.1$$

Le nombre de pas par tour prend donc la forme :

$$N_{pt} = 2N \cdot N_{dr} = 2Np \quad 1.2$$

Avec

$p$  : Nombre de pôles magnétiques vu par le stator.

$N_{dr}$  : Nombre de dents sur une couronne rotorique.

$N$  : Nombre de phases (doit être pair).

L'augmentation du nombre de plots statoriques alimentés simultanément permet d'augmenter le nombre de dents du rotor, et donc de diminuer le pas angulaire du rotor. Le même résultat s'obtient par la subdivision des plots en plusieurs dents.

Les moteurs pas à pas hybrides comptent parmi les moteurs pas à pas les plus fabriqués. Ils existent aussi bien en structure à circuits simples (single stack), qu'en structure multiple (multi stack) Dans presque tous les cas, les plots dentés du stator présentent le même pas que les dents aimantées du rotor.

Le couple est constitué par la variation des perméances mutuelles bobinages –dents aimantées du rotor (couple électromagnétique) et par un couple réluctant créé principalement par la variation de la perméance propre vu par les aimants (couple de détente) Les dents

aimantées de chaque couronne rotorique sont perçues par le stator comme autant d'aimants permanents, l'interaction de ces aimants avec les courants statoriques engendre un couple électromagnétique identique à celui du moteur pas à pas à aimants permanents.

## 2. Moteurs pas à pas linéaires :

Tout comme pour les versions rotatives, les moteurs pas à pas linéaires se composent d'un stator portant les bobinages et d'une partie mobile se déplaçant cette fois-ci linéairement.

Cette partie mobile peut être réalisée à partir d'une pièce ferromagnétique dentée, dans ce cas la structure est dite à reluctance ou passive ; ou alors la partie mobile est munie d'aimants permanents et dans ce cas la structure est dite polarisée ou active.

### a) Moteur pas à pas linéaire à reluctance variable :

Ce type de moteurs est caractérisé par une structure dentée au niveau du stator et de la partie mobile. Le circuit magnétique est généralement assemblé à partie de tôles magnétiques de forte perméabilité. Toutefois, les machines destinées au positionnement ou à une translation lente, peuvent être réalisées avec des pièces massives. Les enroulements du circuit électrique sont généralement concentrés autour des plots du stator et donc faciles à réaliser.

### b) Moteur pas à pas linéaire polarisés :

#### (1) Les moteurs à aimant permanent :

Ce sont des moteurs dont la partie mobile est généralement lisse et formée d'une succession d'aimants permanents montés en surface.

#### (2) Les moteurs hybrides:

Le mouvement des moteurs hybrides résulte de la superposition de la force développé par l'effet réluctant des dents et de la force créée par l'aimant. La contribution des amplitudes et des périodes géométrique de ces forces permet de réaliser des caractéristiques statiques très diversifiées. En effet, l'aimant, placé dans la structure hybride, assure une certaine distribution des lignes de champ. L'alimentation des bobines produit un phénomène d'aiguillage des lignes de champs plus au moins important suivant l'intensité du courant d'alimentation. En jouant sur l'orientation des lignes de champs, il est possible de maîtriser la variation de la force résultante.

## C. Comparaison entre les différents types des moteurs pas à pas

### Etude comparative :

Nous effectuons une étude comparative entre les moteurs à aimant et le moteur à réductance variable (les moteurs les plus utilisés).

Le tableau ci-dessous présente les avantages de l'un par rapport à l'autre :

Type de moteur	Moteur à aimant permanent	Moteur à réductance variable	Moteur hybride
Résolution (nb de pas/tour)	Moyenne	Bonne	Elevée
Couple moteur	Elevé	Faible	Elevé
Sens de rotation	Il dépend : — du sens du courant pour les moteurs bipolaires  — L'ordre d'alimentation des bobines	Il dépend uniquement de l'ordre d'alimentation des bobines	Il dépend : — du sens du courant pour les moteurs bipolaires  — L'ordre d'alimentation des bobines
Fréquence de travail	Faible	Grande	Grande

Tableau 4. Tableau de comparaison entre les différents types des moteurs pas à pas

## D. Etude du courant dans un enroulement (phase) du moteur

$$E = L * \frac{di}{dt} + r * i$$

L : L'inductance d'un enroulement du moteur ,

r : Résistance du moteur.

Solution générale de l'équation sans second membre :

$$L * \frac{di}{dt} + r * i = 0 \quad \frac{di}{i} = -\frac{r}{L} dt \quad \text{Log } i = -\frac{r}{L} * t + K_1 \quad i = K_2 * e^{-\frac{r}{L} * t}$$

Solution particulière de l'équation avec second membre :

$$i = \frac{E}{r}$$

Solution générale de l'équation avec second membre :

$$i = K_2 * e^{-\frac{r}{L} * t} + \frac{E}{r}$$

Détermination de la constante K2 :

$$\text{A } t = 0, i = 0 \quad K_2 = \frac{E}{r}$$

Equation finale :

$$i = \frac{E}{r} - \frac{E}{r} * e^{-\frac{r}{L} * t} = \frac{E}{r} (1 - e^{-\frac{r}{L} * t})$$

## E. Commande d'un moteur pas-à-pas :

On constate que le système est beaucoup plus simple Figure 12.

A chaque impulsion du signal de commande correspond au niveau du rotor un déplacement angulaire défini appelé « pas » ou « incrément mécanique ».

La vitesse de rotation est en fonction de la fréquence des impulsions.

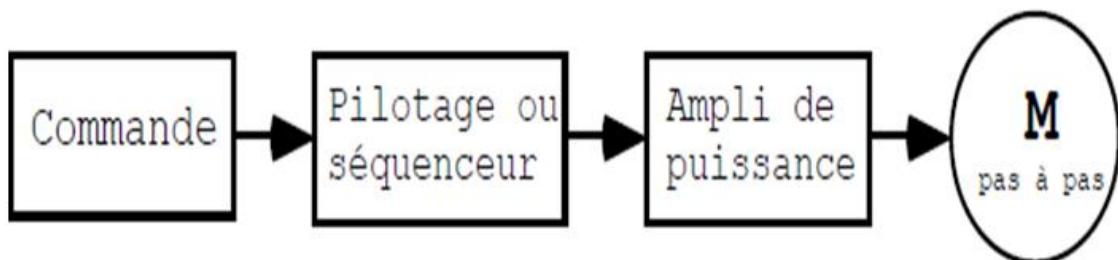


Figure 12. Principe d'un moteur pas-à-pas

## F. Fonctionnement d'un moteur pas-à-pas 4 phases :

La circulation d'un courant électrique dans un bobinage entraîne l'apparition d'un champ magnétique, comme le détaille la figure de gauche dans le cas du solénoïde, et donc la présence de pôles Nord et Sud (deux pôles de même nature se repoussent, deux pôles Nord et Sud s'attirent) ; c'est sur ce principe de base que repose le fonctionnement de tout moteur électrique, et, de manière plus générale, de bon nombre de dispositifs électromécaniques : relais, compteurs, galvanomètres, certains hautparleurs ou microphones . Figure 13.



Figure 13. Flux induit de moteur pas à pas

Le moteur pas à pas, est constitué d'un rotor aimanté (en gris) avec deux pôles, Nord et Sud, ainsi que d'un double-stator (une partie en bleu, l'autre en vert) : à chacune de ces deux parties, est associé un bobinage avec un point milieu et deux phases ; en alimentant l'une ou l'autre des phases, on peut ainsi inverser l'aimantation au niveau du stator correspondant. La flèche noire représente l'aiguille d'une boussole qui serait disposée en place et lieu du rotor ; elle indique l'orientation du champ magnétique (elle pointe vers le nord, qui attire donc le pôle Sud du rotor) et se décale alors d'un quart de tour à chaque étape : Figure 14.

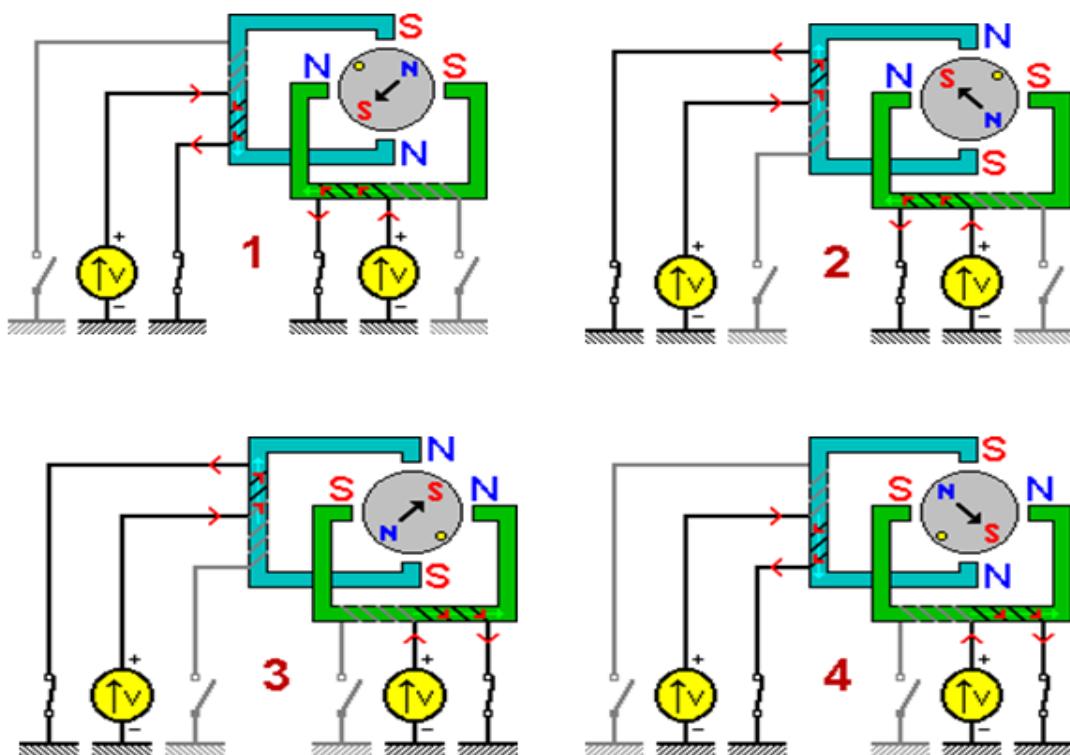


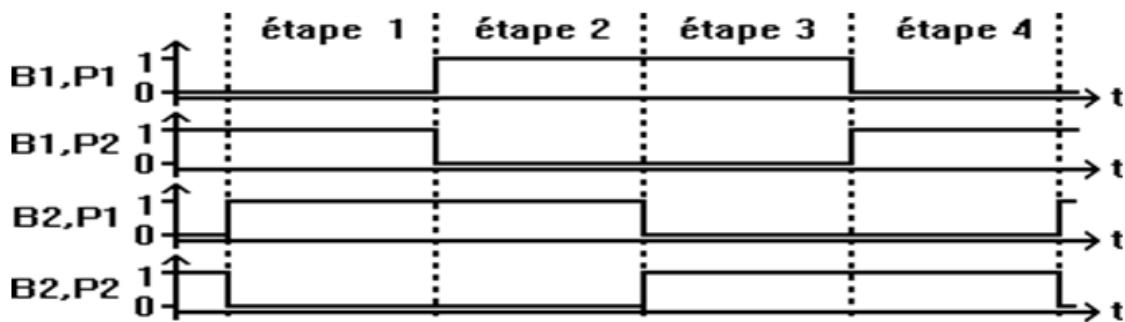
Figure 14. Position de moteur pas-à-pas

La table de vérité ci-dessous résume les états successifs des différentes phases ; l'état logique indique si la phase est alimentée (« 1 ») ou non (« 0 »).

Numéro de l'étape :	1	2	3	4
Bobinage 1, phase 1	0	1	1	0
Bobinage 1, phase 2	1	0	0	1
Bobinage 2, phase 1	1	1	0	0
Bobinage 2, phase 2	0	0	1	1

*Tableau 5. Table de vérité des étapes de moteur pas à pas*

Et le chronogramme correspondant :

*Figure 15. Chronogramme des étapes de moteur pas-à-pas*

Le chronogramme laisse apparaître que pour le bobinage 1, les signaux de contrôle de la phase 1 (B1,P1) et de la phase 2 (B1,P2) sont complémentaires ; il en va de même pour le bobinage 2, concernant (B2,P1) et (B2,P2) ; comme nous le verrons dans l'étude du schéma électrique, la génération des signaux de commande est une opération assez simple... **Figure 15.**

# CONCLUSION :

Dans ce chapitre, nous avons donné un aperçu général sur les différents types de moteurs pas à pas, ainsi que leur mode de fonctionnement.

*Voici leurs avantages :*

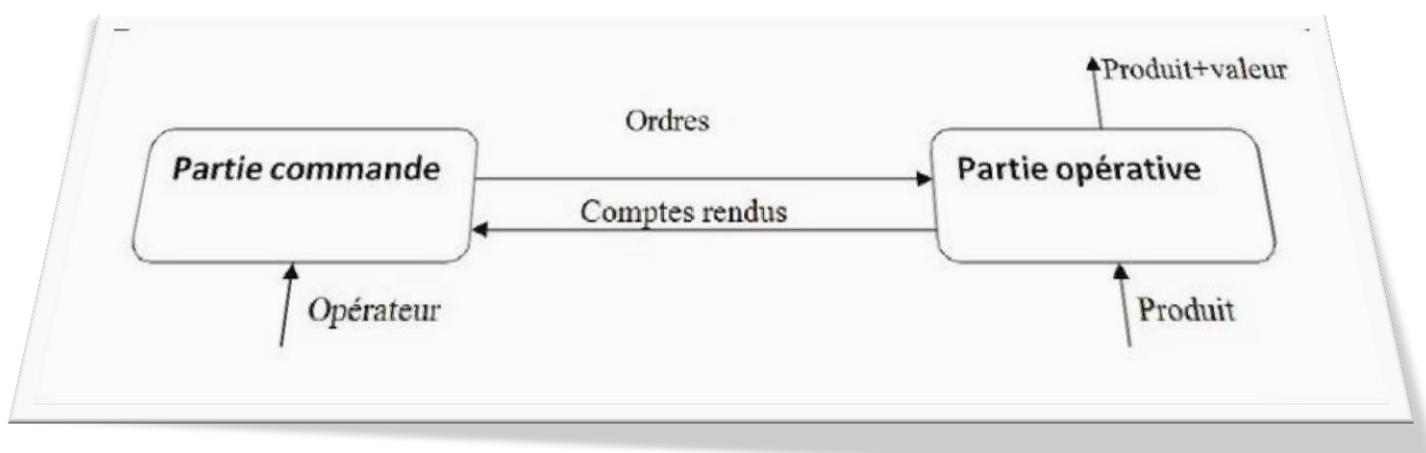
- Rotation constante pour chaque commande (précision meilleure que 5% d'un pas).
- Existence de couple à l'arrêt.
- Contrôle de la position, de la vitesse et synchronisation de plusieurs moteurs (pas besoin de contre-réaction).
- Moteur sans balai.

*Et les inconvénients :*

- Plus difficile à faire fonctionner qu'un moteur à courant continu.
- Vitesse et couple relativement faible.
- Couple décroissant rapidement lorsque la vitesse augmente.
- Résonance mécanique.

En ce qui concerne notre projet, nous avons utilisés des moteurs pas à pas de type bipolaires à aimant permanent, leurs résolution est de 5,62° par pas (64 pas), avec une alimentation électrique de 5V.

# *III. Chapitre d'Etude Technique du commande numérique par calculateur (CNC)*



# INTRODUCTION :

Après avoir décrit le principe et la systématique du moteur pas à pas, nous allons maintenant entamer la partie électronique qui va commander tout le système. Commençons tout d'abord par une description théorique des différents composants qu'on a utilisés pour pouvoir réaliser notre commande.

Durant ces dernières années, la commande des machines électriques a subi des progrès significatifs. Ces progrès sont essentiellement dus à la révolution technologique en informatique industriel, ce qui a permis le développement de solutions numériques efficaces avec une possibilité d'implanter des algorithmes plus complexes.

Après une première génération de commandes numériques à logique câblée sont apparues les commandes numériques par calculateur (CNC), ou par ordinateur, qui intègrent un ou plusieurs ordinateurs spécifiques pour réaliser tout ou partie des fonctions de commande. Tous les systèmes de commande numérique commercialisés actuellement contenant au moins un microprocesseur, les termes CN et CNC peuvent être considérés comme des synonymes.

Une CNC c'est une machine pilotée par des moteurs pas à pas. Ces derniers permettent de convertir directement un signal électrique numérique en un positionnement angulaire de caractère incrémental, par exemple :

- Imprimant 3d.
- La machine de découplage.
- Mini *traceur*.

Ce chapitre présente le contexte d'étude à travers une description des machines CNC et les moteurs pas à pas.

## A. Historique

Les travaux menés par Falcon et Jacquard à la fin du XVIII<sup>e</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, à Traverse City 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. L'emplacement et la profondeur de chaque trou sont calculés avec précision par un ordinateur IBM à cartes perforées. La finition de la surface est obtenue par des opérations manuelles de polissage.

## B. Principe de fonctionnement d'une machine numérique

Les machines à commande numérique sont devenues des moyens de production incontournables dans l'industrie. Elles permettent des cadences de production importantes et facilitent l'obtention de surfaces complexes (formes arrondies ...).

Ce type de machine se compose ainsi de deux parties complémentaires (figure 16):

- La partie opérative.
- La partie commande.

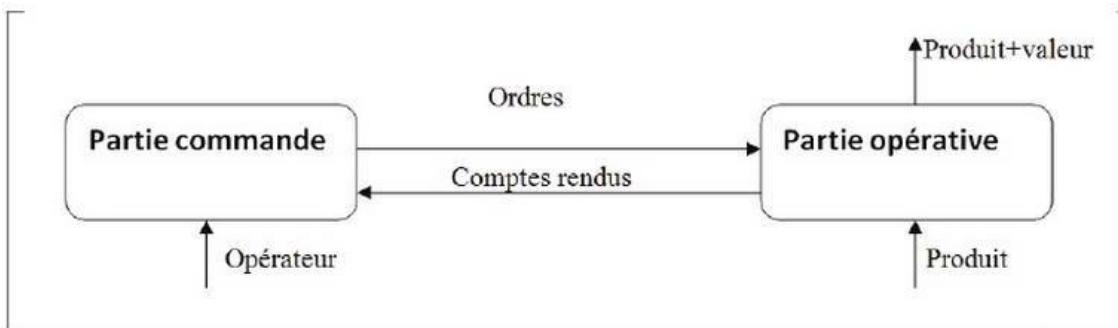


Figure 16. Décomposition d'une machine numérique.

La partie opérative comporte les axes de déplacement et la tête. La partie commande permet de piloter la partie opérative. Elle est composée d'un calculateur (CNC) et d'éléments électroniques capables de piloter les moteurs : les cartes d'axes.

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. Il existe 3 principaux types de référentiels à prendre en compte (figure 17):

- L'origine machine (Om) qui correspond à la position de référence de la machine où  $X_m = 0$ ,  $Y_m = 0$  et  $Z_m = 0$  (s'il y en a).
- L'origine pièce (Op) qui peut être décalée par rapport à l'origine machine.
- L'origine Programme (OP) qui est généralement confondu avec Op pour faciliter l'usinage.

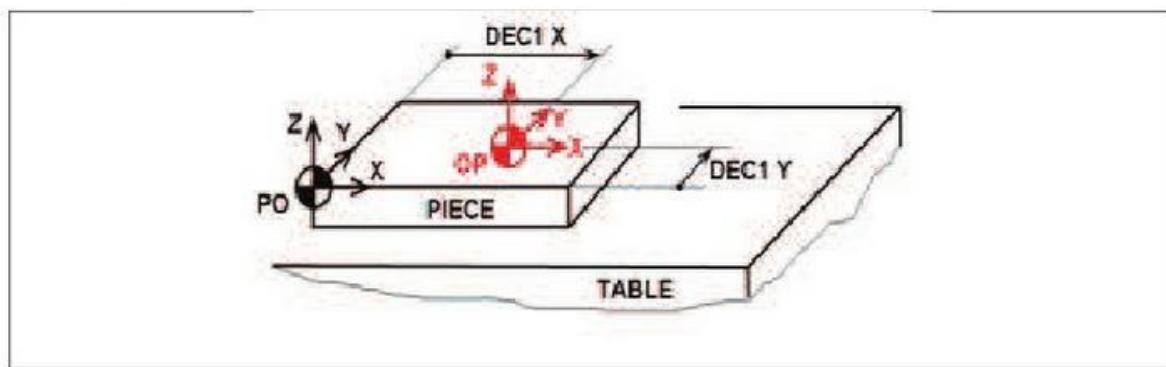


Figure 17. Machine à trois axes

## 1. La commande numérique par calculateur(CNC) :

C'est en 1942 aux États-Unis que la CN a commencé à être exploitée, pour permettre l'usinage de pompes à injection pour moteurs d'avions. Il s'agissait en fait de cames, dont le profil complexe était irréalisable au moyen d'une machine traditionnelle. La commande numérique a pour tâche de générer des mouvements d'axes.

Ces instructions qui viennent du programme pièce ou de l'opérateur machine consisteront en une position et une vitesse de déplacement. 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 calculateur. 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.

## 2. La Conception Assistée par Ordinateur (CAO)

« La CAO permet de concevoir des systèmes dont la complexité dépasse la capacité de l'être humain comme par exemple en micro-électronique». Dans notre cas, on s'intéresse à la conception des formes de deux dimensions.

Un progiciel tel que inkscape ou blinder et Solide Works génère un fichier qui simule la surface, et qui est compréhensible par d'autre progiciel de fabrication assisté par ordinateur.

## 3. La Fabrication Assisté par Ordinateur (FAO)

La conception de la surface à obtenir est donc réalisée à l'aide d'un progiciel de conception assistée par ordinateur (CAO): on nomme le fichier ainsi obtenu "DFN" pour Définition de Formes Numérisées.

Cette modélisation est ensuite « exportée » dans un fichier intermédiaire en utilisant un standard d'échange comme IGES, STEP, DXF ou autre.

Certains outils de FAO sont capables de relire directement les fichiers des grands fournisseurs de CAO. Dans d'autres cas, la CAO et la FAO sont complètement intégrées et ne nécessitent pas de transfert. Pour ces progiciels, on parle de CFAO.

#### 4. Le post-processeur

C'est un petit programme utilitaire qui transforme des trajectoires en format neutre en trajectoires en repère MOCN (machine-outil à commande numérique).

Ce traducteur tien compte de la cinématique de la machine, de ses courses et de ses capacités.

Il signale les erreurs et exprime les trajectoires dans le langage spécifique à la machine.

Chaque machine possède donc son propre post-processeur (figure 18).

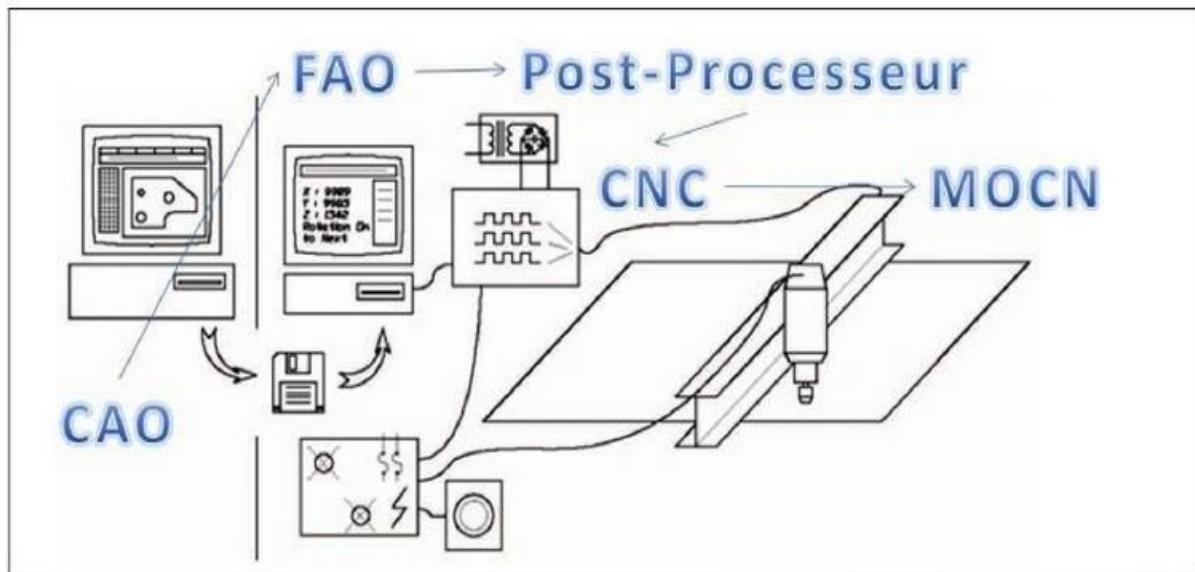


Figure 18. Cycle d'une pièce fabriquée par une commande numérique (CFAO)

## C. *Les éléments de la partie commande*

### 1. La carte de commande ARDUINO :

#### a) *Mise en œuvre de la carte ARDUINO UNO :*

La carte ARDUINO est une carte électronique basée autour d'un microcontrôleur et de composants minimum pour réaliser des fonctions plus ou moins évoluées à bas cout.

Elle possède une interface USB pour la programmer. C'est une plateforme open-source qui est basée sur une simple carte a microcontrôleur (de la famille AVR), et un logiciel, véritable environnement de développement intégré, pour écrire, compiler et transférer le programme vers la carte à microcontrôleur.

ARDUINO est peut être utilise pour développer des applications matérielles industrielles légères ou des objets interactifs, et il peut recevoir en entrées une très grande variété de capteurs. Les projets ARDUINO peuvent être autonomes, ou communiquer avec des logiciels sur un ordinateur (Flash, Processing ou MaxMSP, Labview).

Les cartes électroniques peuvent être fabriquées manuellement ou bien être achetées pré assemblées, le logiciel de développement open-source est téléchargé gratuitement. La programmation de la carte ARDUINO présente les principales fonctionnalités de l'interface de l'application ARDUINO. L'application ARDUINO vous permet de créer et éditer un programme (appelé sketch) qui sera compilé puis téléchargé sur la carte ARDUINO. Ainsi, lorsque vous apportez des changements sur le code, ces changements ne seront effectifs qu'une fois le programme téléchargé sur la carte.

#### b) *Présentations et caractéristiques du module ARDUINO UNO :*

La carte ARDUINO UNO, prêtée est une carte a microcontrôleur basée sur un Atmega328p.

Cette carte dispose :

- ✓ Broches numériques d'entrées/sorties,
- ✓ Entrées analogiques,
- ✓ Quatre UART (port série matériel),
- ✓ Quartz 16Mhz,
- ✓ Connexion USB,
- ✓ Connecteur d'alimentation jack,
- ✓ Connecteur ICSP,
- ✓ Bouton de réinitialisation.

Elle contient tout ce qui est nécessaire pour le fonctionnement du microcontrôleur.

Pour pouvoir l'utiliser et se lancer, il suffit simplement de la connecter à un ordinateur à l'aide d'un câble USB (ou de l'alimenter avec un adaptateur secteur ou une pile, mais ceci n'est pas indispensable, l'alimentation est fournie par le port USB).

La carte ARDUINO UNO dispose de toute une série de facilités pour communiquer avec un ordinateur, une autre carte ARDUINO, ou avec d'autres microcontrôleurs.

L'ARDUINO UNO se dispose de quatre UARTs (Universal Asynchrones Receiver Transmitter ou émetteur- récepteur asynchrone universel en français) pour une communication en série de niveau TTL (5V) et qui est disponible sur les broches 0 (RX) et 1 (TX). Un circuit intégré Atmega16U2 sur la carte assure la connexion entre cette communication série de l'un des ports série de l'Atmega UNO vers le port USB de l'ordinateur qui apparaît comme un port COM virtuel pour les logiciels de l'ordinateur.

Le code utilisé pour programmer l'Atmega8U2 utilise le driver standard USB COM, et aucun autre driver externe n'est pas nécessaire.

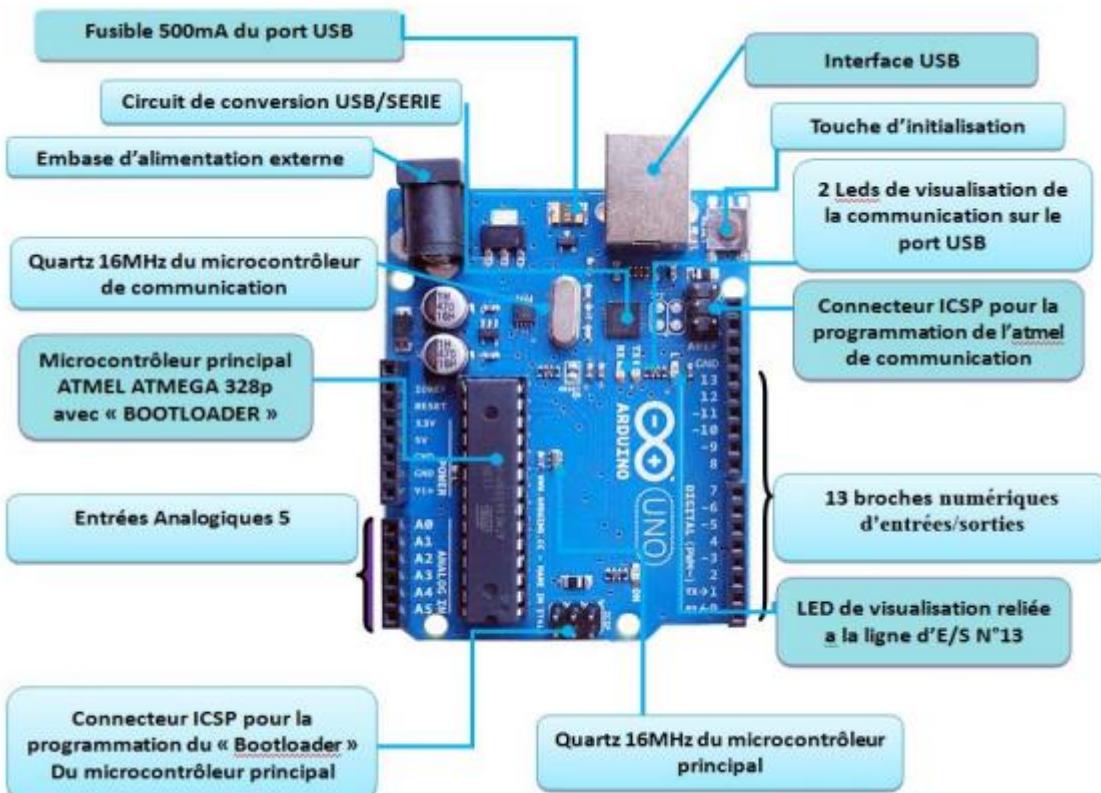


Figure 19. Carte ARDUINO UNO.

## 2. Stepper motor driver A4988 :

### a) Définition :

Le A4988 est un circuit électronique permettant de contrôler un moteur pas a pas avec différents modes.

Son interface de puissance est assurée par des pentes en H réalisé par des MosFets N-Channel.

### b) Fonctionnement :

le driver l'A4988 permet non seulement de fournir la puissance nécessaire a la commande des bobines (hacheur intégré) mais également de simplifier le pilotage et choisir facilement le mode de pas a pas.

Il offre ainsi non seulement la possibilité de travailler en pas complet et demi-pas mais également deux autres modes :

1/4 de pas et 1 /8 de pas en autorisant des positions intermediaires dans un pas.

Cela est rendu possible en modulant intelligemment la quantité de courant dans les bobines du moteur pas-a-pas.

Par exemple, piloter un moteur en mode « 1/4 de pas » permet d'obtenir 800 microsteps (micro-pas) sur un moteur prévu pour 200 pas par révolution et cela en imposant 4 niveau de courants différents pour chacun des micro-pas.

Le driver est donc équipé d'un asservissement de courant pour gérer correctement celui-ci dans les bobines.

Un potentiomètre permet de définir la limite de courant (ou consigne de courant maximale) qui doit être choisie en fonction de la charge entraînée par le moteur et de la vitesse souhaitée

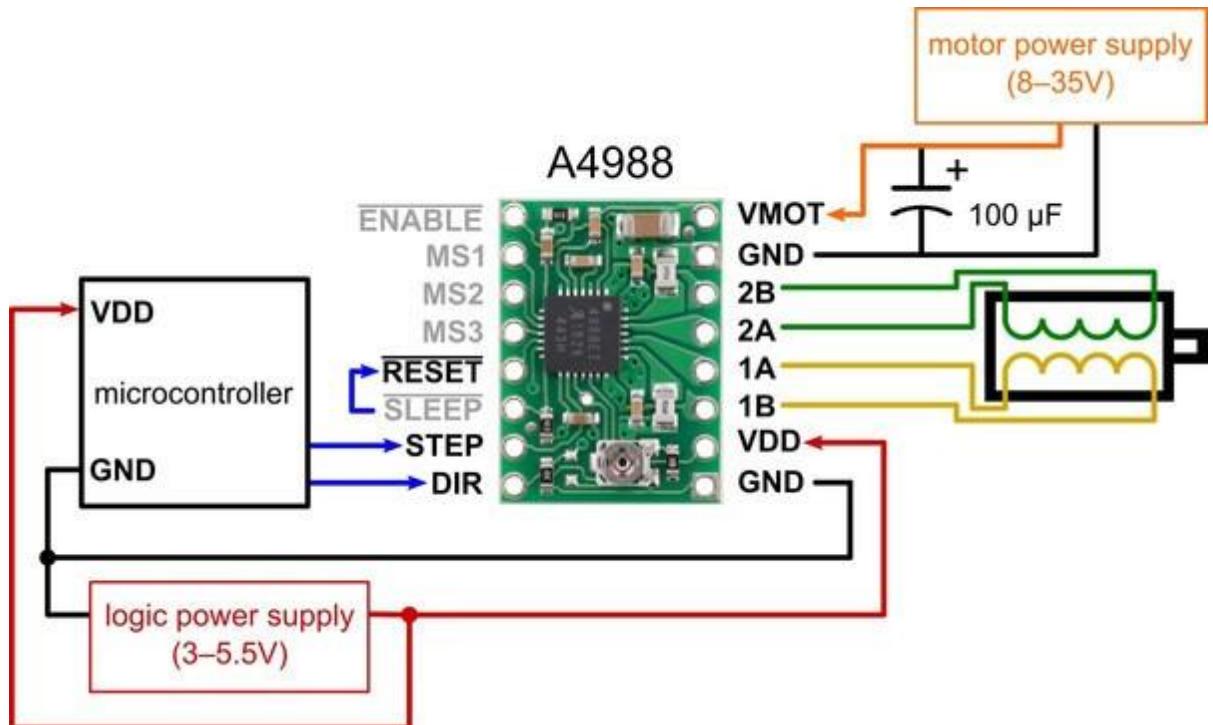


Figure 20. Circuit de contrôle d'un moteur a l'aide du A4988

La configuration du driver se fait a l'aide des broches MS1, MS2 et MS3.

Le câblage du driver est fourni ci-après.

Outre l'alimentation de la carte en 5V, le raccordement des 4 fils du moteur pas a pas et l'alimentation de puissance, on trouve plusieurs broches a raccorder au microcontrôleur :

- ENABLE : entrée de validation, active a l'état logique haut permettant de commander l'alimentation de la carte (sinon le moteur reste maintenu dans un pas donne par les bobines et consomme du courant)
- DIR : permet de spécifier le sens de rotation du moteur
- STEP : permet de spécifier le nombre de pas ou micro-pas a effectuer (le déplacement effectif du moteur dépendra du mode choisi pas entier, demi-pas etc.)
- MS1 a MS3 permettent de choisir le mode de fonctionnement du moteur. Ces entrées disposent de résistance de tirage a l'état bas (pull-down) internes, ce qui signifie que si l'on ne place pas le potentiel de ces broches au niveau logique haut = High = VDD (avec VDD la tension choisie pour la logique de commande) alors elles seront automatiquement ramenée au niveau logique bas = Low.

Le tableau suivant permet de comprendre le mode de pas choisi en fonction des broches MS1 et MS2.

MS1	MS2	MS3	Resolution Microstepping
Low	Low	Low	Pas complet (full step)
High	Low	Low	1 /2 pas
Low	High	Low	1 /4 de pas
High	High	Low	1 /8 <sup>eme</sup> de pas
High	High	High	1/16 <sup>eme</sup> de pas

Figure 21. Les différentes commandes pour le mode pas complet

### 3. La carte SHIELD CNC:

#### a) Définition:

Le SHIELD CNC s'imbrique directement sur l'ARDUINO UNO.

Il y a un emplacement prévu pour 4 contrôleurs moteurs du même type que le pilote du moteur pas à pas A4988

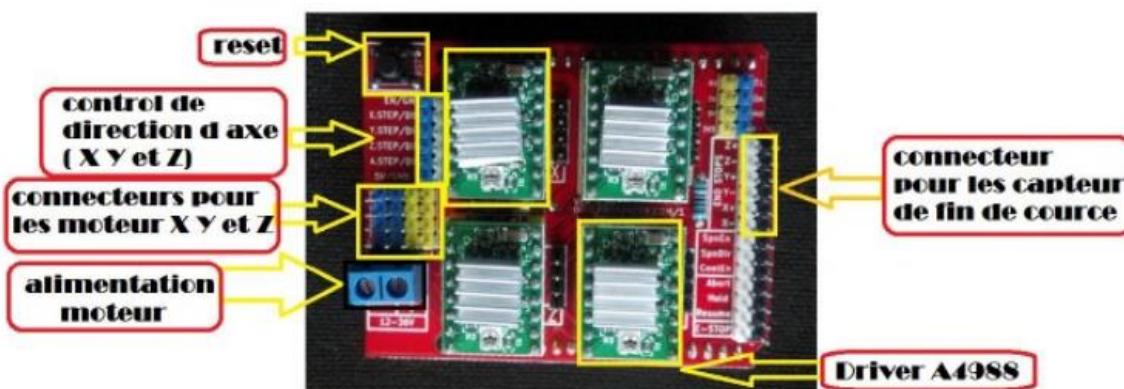


Figure 22. SHIELD CNC avec quatre A4988

Il suffit donc de clipper le contrôleur moteur et de connecter le moteur grâce aux pins situés à côté du contrôleur pour simplifier le câblage est d'utiliser un SHIELD CNC .

Il est très pratique car il se fixe directement sur l'ARDUINO.

## D. Programmation Et Interface graphique

Pour que la machine comprenne les instructions données par l'ordinateur de contrôle, il faut que ces instructions soient transmises dans un langage de programmation spécial, qui est le G-code, ou code-G.

Les outils informatiques actuels de la chaîne soft CAO permettent de se passer de la connaissance pratique du langage, mais pour ceux qui voudraient éventuellement faire des programmes simples directement avec un éditeur pour les envoyer dans le contrôleur ARDUINO, il est important de connaître quelques bases de ce langage.

Inkscape est un logiciel de dessin vectoriel qui offre une interface graphique de gestion des images vectorielles.

### 1. Langue de contrôleurs numériques « G-code »

#### a) Description

Le G-code est le langage de programmation des machines numériques est basé sur des lignes de code.

Chaque ligne (également appelée un bloc) peut inclure des commandes pour faire produire diverses actions à la machine.

Plusieurs lignes de code peuvent être regroupées dans un fichier pour créer un programme G-code.

Une ligne de code typique commence par un numéro de ligne optionnel suivi par un ou plusieurs mots.

Un mot commence par une lettre suivie d'un nombre (ou quelque chose qui permet d'évaluer un nombre).

Un mot peut, soit donner une commande, soit fournir un argument à une commande.

Par exemple, G1 X3 est une ligne de code valide avec deux mots.

G1 est une commande qui signifie déplaces toi en ligne droite à la vitesse programmée et X3 fournit la valeur d'argument (la valeur de X doit être 3 à la fin du mouvement).

La plupart des commandes G-code commencent avec une lettre G ou M.

Les termes pour ces commandes sont G-code ou M-codes.

Le langage G-code n'a pas d'indicateur de début et de fin de programme.

L'interpréteur cependant traite les fichiers.

Un programme simple peut être en un seul fichier, mais il peut aussi être partagé sur plusieurs fichiers.

Un fichier peut être délimité par le signe pour-cent de la manière suivante.

La première ligne non vide d'un fichier peut contenir un signe % seul, éventuellement encadré d'espaces blancs, ensuite, à la fin du fichier on doit trouver une ligne similaire.

Délimiter un fichier avec des % est facultatif si le fichier comporte un M2 où un M30, mais est requis sinon.

Une erreur sera signalée si un fichier a une ligne pour-cent au début, mais pas à la fin.

Le contenu utile d'un fichier délimité par pour -cent s'arrête après la seconde ligne pour-cent.

Tout le reste est ignoré. Le langage G-code prévoit les deux commandes ( M2 ou M30) pour finir un programme.

Le programme peut se terminer avant la fin du fichier.

Les lignes placées après la fin d'un programme ne seront pas exécutées.

L'interpréteur ne les lit pas.

### b) Format d'une ligne

Une ligne de G-code typique est construite de la façon suivante, dans l'ordre avec la restriction à un maximum de 256 caractères sur la même ligne.

1. Un caractère optionnel d'effacement de bloc, qui est la barre oblique.
2. Un numéro de ligne optionnel.
3. Un nombre quelconque de mots, valeurs de paramètres et commentaires.
4. Un caractère de fin de ligne (retour chariot ou saut de ligne ou les deux).

## 2. L'éditeur du texte ou image compilateur de langage CNC :

L'éditeur de texte est une interface graphique sur l'ordinateur où l'opérateur peut traiter le G-code à exécuter s'appelé « Inkscape ».

Ensuite, ce texte doit être compilé, s'il y a des erreurs ; l'opérateur est prévu afin de les corriger. Une fois compilé, le code est prêt à envoyer à la carte ARDUINO UNO ou à simuler sur l'ordinateur

### a) Définition Inkscape

Inkscape est un puissant logiciel de dessin vectoriel à vocation "artistique", le dessin y est

enregistré sous forme d'équations de courbes (chemins) et non par des pixels comme les images bitmap.

Ce programme est vaste (mais très bien fait), les fonctions qui nous intéressent dans le cadre de ce tutoriel c'est qu'il permet d'ouvrir un nombre très important de type d'image (PS, DXF, PDF, SVG, PNG, JPG, BMP, XCF,...) et qu'il est capable de transformer des images bitmap en image vectorielle

"G-code tools" est une extension d'Inkscape permettant (entre autre) de générer du G-code à partir des chemins d'une image vectorielle.

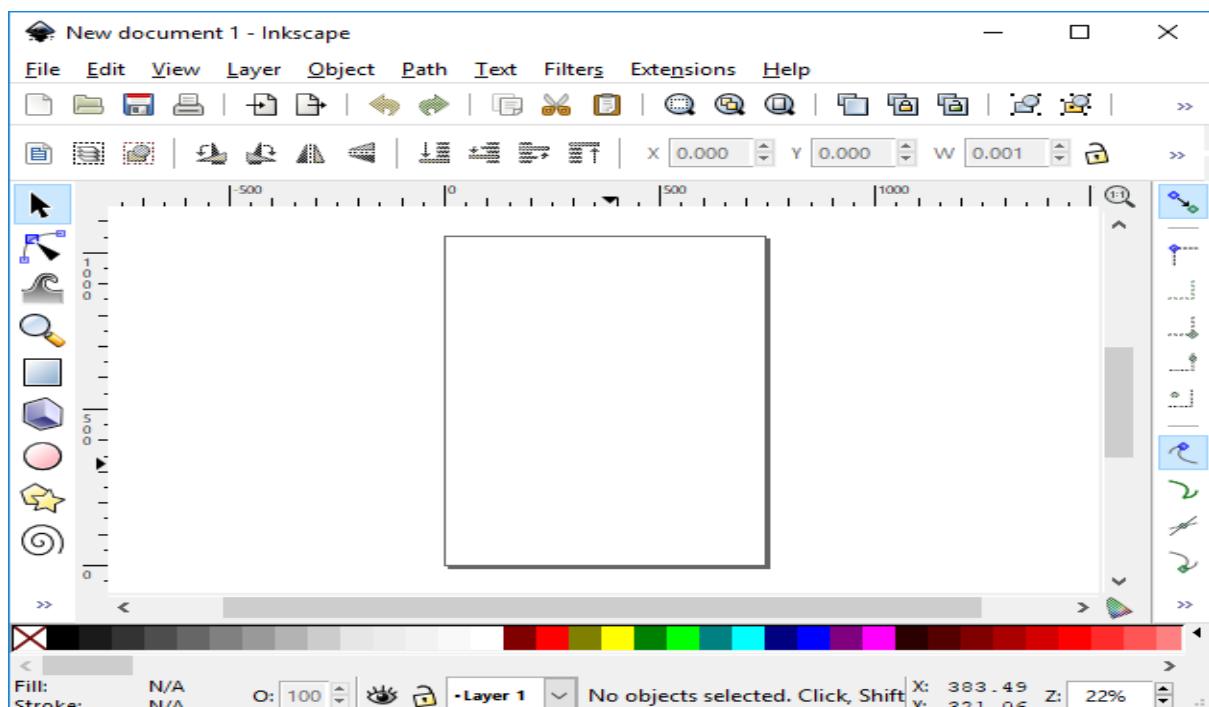


Figure 23. L'interface principale d'Inkscape

### 3. Logiciel de pilotage de la machine CNC (Universal G-code Sender)

La solution la plus fonctionnelle que j'ai trouvée est une évolution d'Universal G-code Sender (développé initialement par l'auteur de GRBL), qui est cours de développement par Winder.

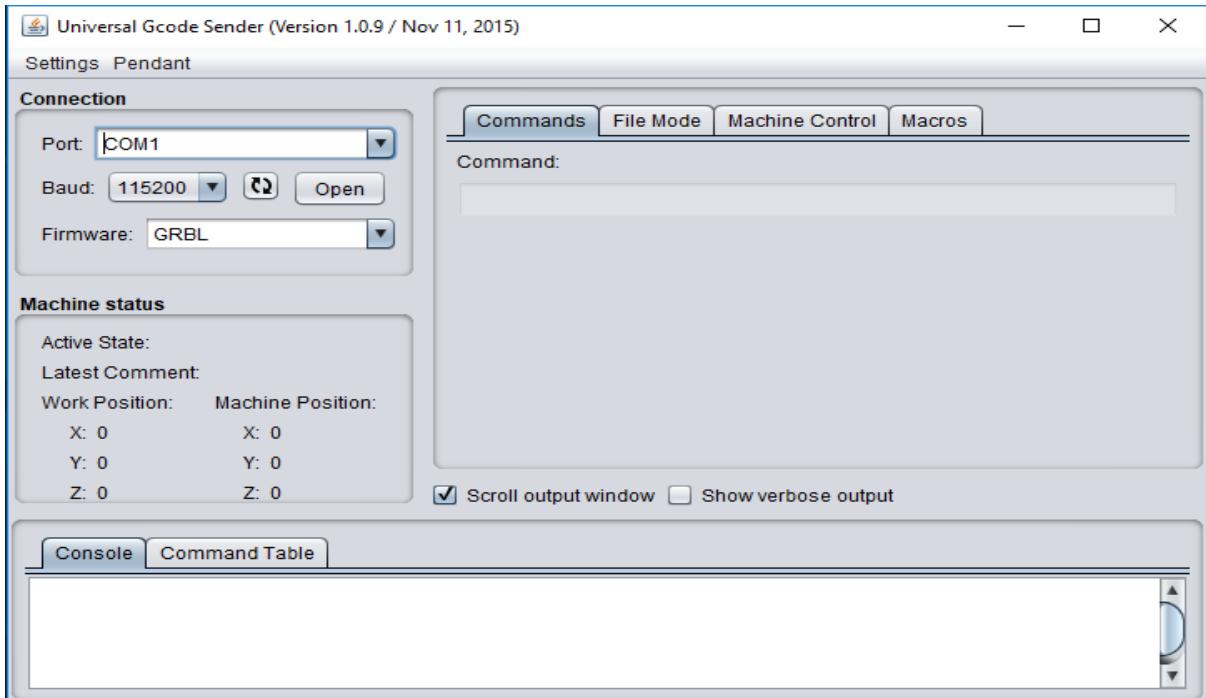


Figure 24. Universal G-code Sender.

### a) Modes de commande

Il y'a deux mode utilisé dans ce programme soit mode manuel ou mode automatique pour commande le machine CNC, Le programme fournit spéciale boutons dans l'interface de base qui contrôle chaque moteur séparément, Ce qui fonctionne ces boutons après choisi le mode manuel.

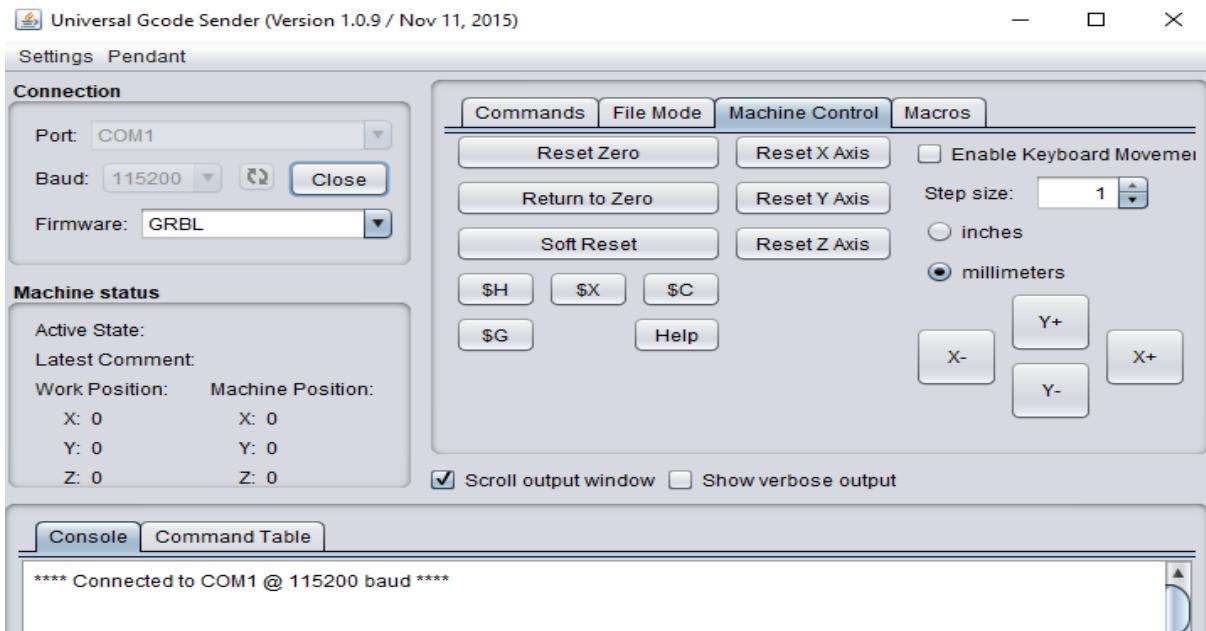


Figure 25. l'interface de "G-code Sender " on mode manuel.

Pour l'utilisation graphique : le mode automatique, Charge notre fichier G-code: dans l'onglet "file mode" faites "browse" et sélectionnez fichier G-code précédemment créé. En faisant "visualize" nous pourrons vérifier notre tracé et le chemin emprunté par l'outil: en rouge les zone de travail ( $Z < 0$ ); en bleu les déplacements ( $Z > 0$ ).

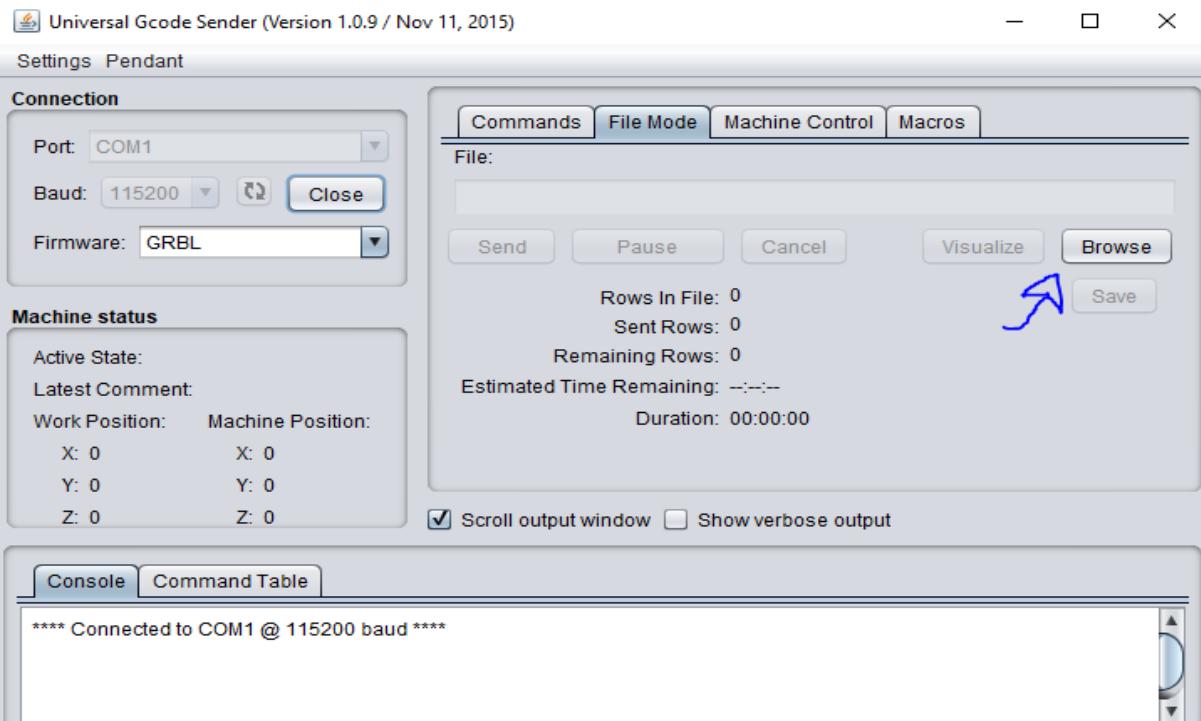


Figure 26. l'interface de "G-code Sender " choisir le fichier G-code.

Faire le zéro (le "home"): dans l'onglet "machine control" déplacez l'outil avec les boutons X, Y et Z.

Nous pouvons modifier le pas de déplacement en modifiant le "step size".

Une fois l'outil positionner là où vous souhaitez établir le zéro (en contact avec notre support) cliquez sur "reset zéro". Pour lancer le travail: retournez dans l'onglet file mode et cliquez sur "Send".

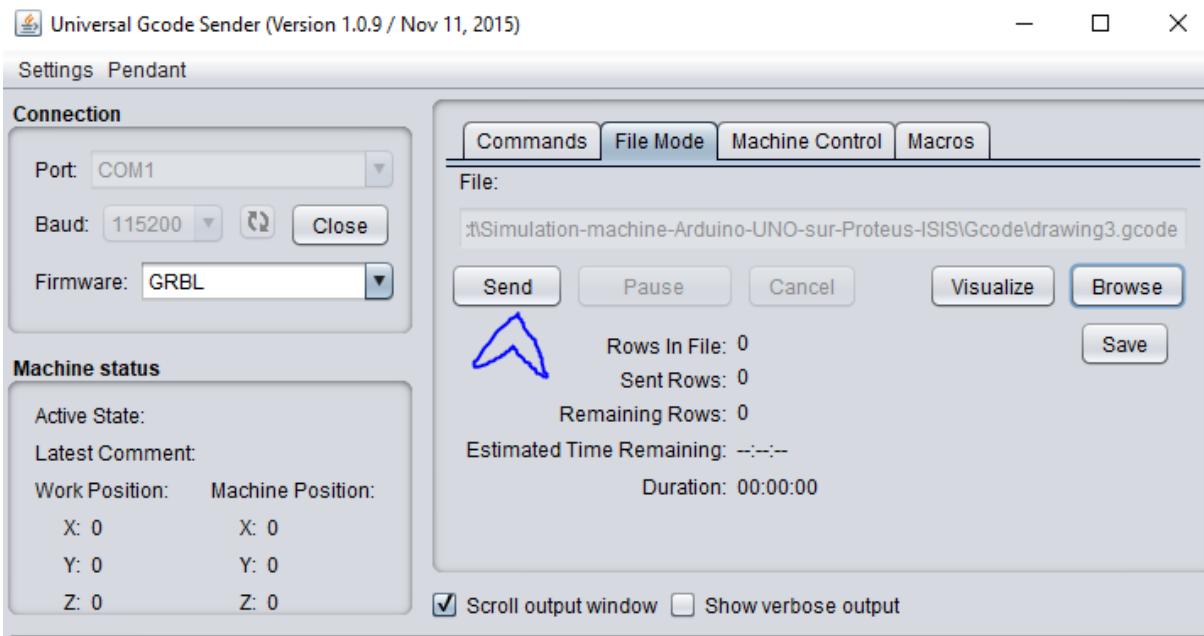


Figure 27. envoyé LE fichée G-code

### b) Mode Configuration

Autre que le mode manuel et le mode automatique, il y a le mode configuration. Dans ce mode, nous pouvons configurer les dimensions de la table.

Afin de configurer les dimensions, l'opérateur est appelé à mettre la tête aux coins inverses de l'origine table après le lancement d'une prise d'origine.

Cela est nécessaire pour mémoriser les distances maximales des axes X et Y. et la vitesse et nombre de pas par unité de tel moteur. Ainsi, ces paramètres sont enregistrés dans la mémoire EEPROM.

### c) Mode Manuel

En mode manuel, l'opérateur peut commander les moteurs directement par les boutons (X+, X-, Y+, Y-, Z+, Z-, ..) dans la programme de transfert G-code , En cliquant sur ses boutons nous générerons un instruction de code G qui s'envoie à la carte de commande pour commander le moteur.

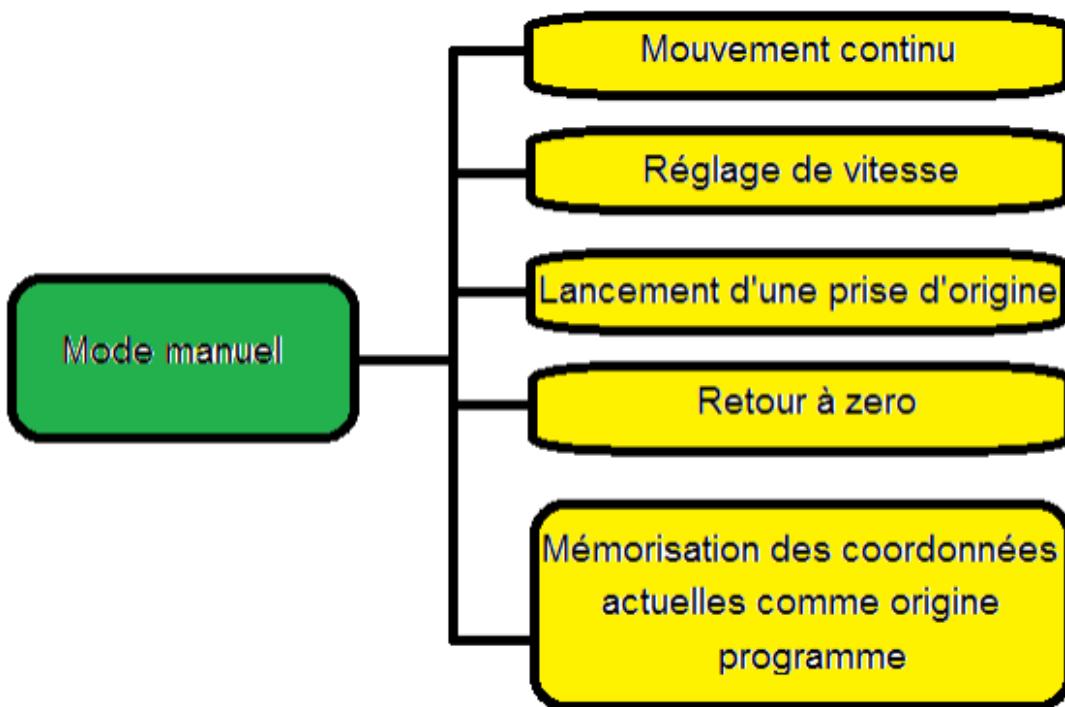


Figure 28 . mode manuel.

Mouvement continu : lors de déplacement des moteurs, si l'opérateur appui sur un bouton de déplacement, la tête se déplace d'une manière continue avec une vitesse réglable.

Incrémentation par pas : pour un réglage fin, plusieurs distances de déplacement sont proposées. Dans ce mode, un seul appui génère un déplacement d'une seule unité choisie par l'opérateur.

Réglage de la vitesse : la vitesse de la tête en mode manuel est réglable. Le choix de cette vitesse est limité par la vitesse minimale et maximale indiquées au manuel des moteurs ( $V_{min}$ ,  $V_{max}$ ).

Retour à zéro: il s'agit de déplacer la tête à l'origine absolue. Ce qui diffère de la prise d'origine décrite précédemment c'est que la tête retourne à l'origine sans passer par les butées, donc il est basé sur l'estimation de la position calculé par l'ARDUINO.

Retour à l'origine programme : l'origine programme est par défaut l'origine absolue de la table, en fait, c'est l'origine du texte ou image. Cette origine peut être changée par le code en G en mode automatique ou même manuellement. Une fois changé, l'ARDUINO le mémorise dans son EEPROM. Mémoriser l'origine programme : si l'opérateur veut changer cette origine, il doit déplacer la tête manuellement à l'aide du déplacement continu et l'incrémentation par pas, puis il exécute la mémorisation

#### d) Mode automatique

En mode automatique, la carte ARDUINO UNO reçoit les instructions en G-code et les exécute l'une après l'autre. Il y a plusieurs étapes pour achever cette tâche. Commençant par la réception de l'instruction jusqu'à la commande des moteurs et de la tête. Si l'opérateur clique sur le bouton « send » après le chargement de fichier G-code (soit texte ou image) dans le programme de transfert, l'ARDUINO fonctionne en mode automatique, le ARDUINO entre dans une boucle et prêt pour recevoir les données. L'ARDUINO fait l'enregistrement dans la mémoire RAM, Puis fait la compilation pour savoir l'opcode (code de l'opération) et évidemment ses paramètres s'ils existent. Chaque opcode de code G a une fonction spécifique, s'il s'agit d'un paramétrage, l'instruction affecte une variable en EEPROM tels que l'origine programme, origine pièce, vitesse de déplacement....

Enfin, l'ARDUINO envoi le mot ‘ ’ pour indiquer au PC qu'il est prêt à recevoir l'instruction suivante. L'appui sur le bouton 'cancel' termine le dialogue.

### 4. Langage de programmation la machine CNC

Un langage de programmation est un langage permettant à un être humain d'écrire un ensemble d'instructions (code source) qui seront directement converties en langage machine grâce à un compilateur (c'est la compilation). L'exécution d'un programme ARDUINO s'effectue de manière séquentielle, c'est-à-dire que les instructions sont exécutées les unes à la suite des autres. Voyons plus en détail la structure d'un programme écrit en ARDUINO. Le programme utilise l'équipe de Grbl est une le programme de commande dans la carte ARDUINO.

#### a) Programme Grbl

Grbl est un micro-logiciel libre développé sur ARDUINO pour contrôler des graveuses CNC (Computer Numerical Control), i.e. des fraiseuses munis d'une tête mobile contrôlée en X, Y et Z par un ordinateur. Grbl interprète du G-code (cf. plus bas) et déplace en conséquence un outil sur 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...

L'organigramme qui assure ce micro-logiciel:

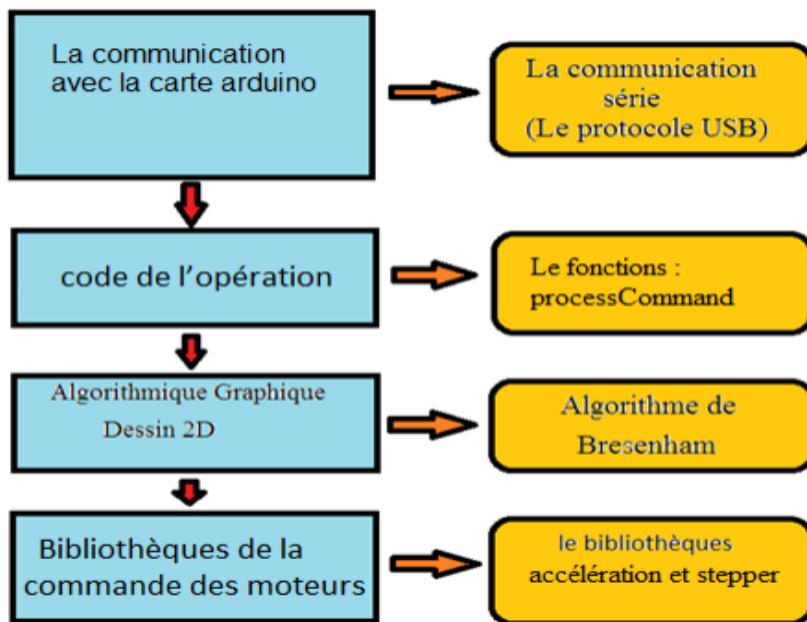


Figure 29. GRBL code

### (1) La communication avec la carte ARDUINO

La communication série est indispensable pour dialoguer avec ARDUINO puisque c'est le mode de transmission utilisé pour communiquer avec la carte.

Les protocoles de communication série le plus connus sont : Le protocole USB.

- Serial.begin(speed) (configuration de la vitesse de communication Série).
- Serial.available() (donne combien de caractères disponibles dans la zone tampon Série) .
- Serial.read() (lit les données Série) .
- Serial.print(data) (envoie des données Série) .
- Serial.println(data) (envoie des données Série suivies de caractères spécifiques).

L'avantage de la communication série est qu'elle nécessite moins de lignes, donc moins de broches, donc moins de composants. Son coût est donc plus faible

### (2) code de l'opération

Cette partie est le cœur du programme, qui définit les actions à effectuer à partir le traitement de code-G, qui contient plusieurs fonctions.

L'ARDUINO fait l'enregistrement dans la mémoire RAM, Apres la compilation nous obtenons l'opcode (code de l'opération) et évidement ses paramètres s'ils existent. Chaque opcode du code-G a une fonction spécifique.

La fonction principale qui traduit le G code s'appelle : « processCommand() » qui contient

l'opcode se la forme suivent :

```

void processCommand() {
    int cmd = parsenumber('G', -1);
    switch(cmd) {
        case 0:
        case 1: { // line
            feedrate(parsenumber('F', fr));
            line( parsenumber('X', (mode_abs?px:0)) +
                (mode_abs?0:px),
                parsenumber('Y', (mode_abs?py:0)) + (mode_abs?0:py),
                parsenumber('Z', (mode_abs?pz:0)) + (mode_abs?0:pz),
                break;
        }
        case 2:
        case 4: pause(parsenumber('P', 0)*1000); break; // dwell
        case 90: mode_abs=1; break; // absolute mode
        case 91: mode_abs=0; break; // relative mode
        case 92: // set logical position
            position( parsenumber('X', 0),
            parsenumber('Y', 0),
            parsenumber('Z', 0));
            break;
        default: break;
    }
    cmd = parsenumber('M', -1);
    switch(cmd) {
        case 17: motor_enable(); break;
        case 18: motor_disable(); break;
        case 100: help(); break;
        case 114: where(); break;
        default: break;
    }
}

```

Figure 30 . ProcessCommand() fonction

### (3) Bibliothèques de la commande des moteurs

Cet partie contient les fonctions qui transfère les calculs de l'interpolation de Bressenham vers un signal de commande pour contrôle de la carte de commande. La fonction principale utilise est la fonction accélération et la fonction stepper.

## 5. Organigramme de la commande de la machine CNC.

Finalement en peut résumée le pilotage et la commande de la machine CNC par un microordinateur, l'organigramme suivant :

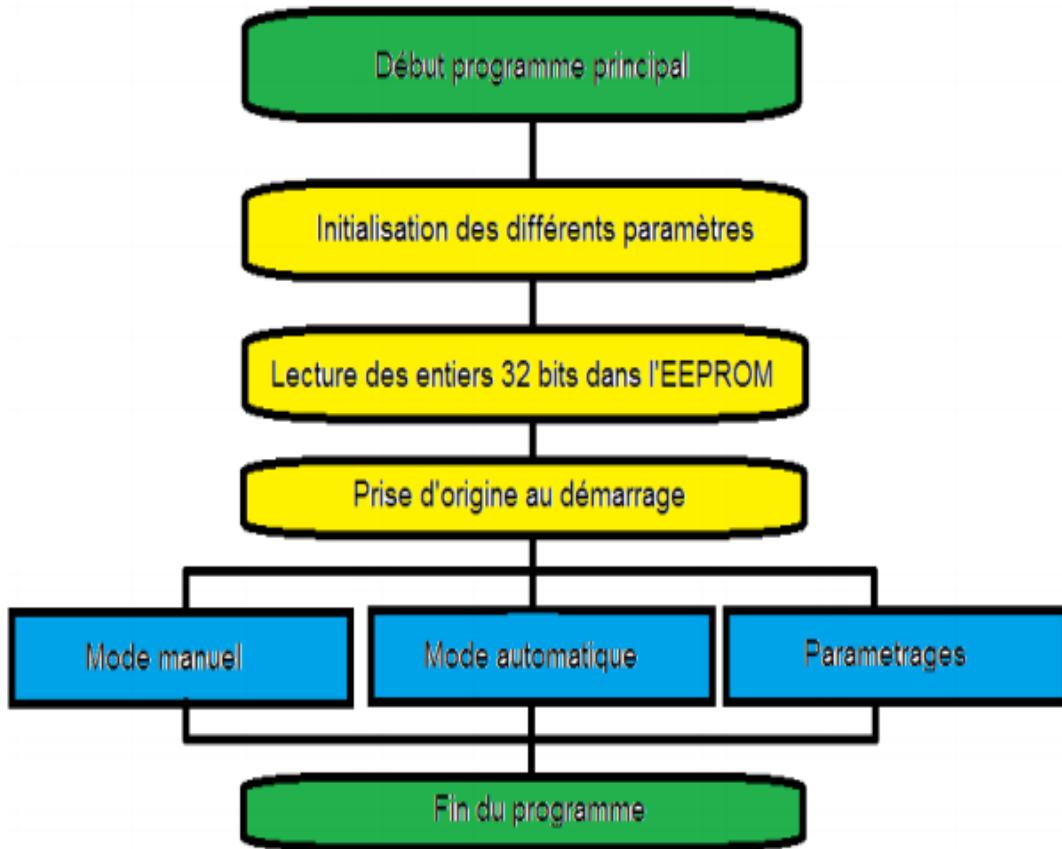


Figure 31. Organigramme de la commande de la machine CNC.

Une fois la machine CNC fonctionne, il existe trois différents modes majeurs pour entrer des commandes. Les modes Manuel, Auto et Paramétrage. Passer d'un mode à une autre marque une grande différence dans le comportement de la machine CNC. Des choses spécifiques à un mode ne peuvent pas être faites dans un autre.

L'opérateur peut faire une prise d'origine sur un axe en mode manuel mais pas en mode auto. L'opérateur peut lancer l'exécution complète d'un programme de G-codes en mode auto mais pas en mode manuel. En mode manuel, chaque commande est entrée séparément. En termes humains une commande manuelle pourrait être active un moteur pas à pas. Ces commandes sont normalement contrôlées en pressant un bouton de l'interface graphique avec la souris ou en maintenant appuyée une touche du clavier.

En mode auto, un bouton similaire ou l'appui d'une touche peuvent être utilisés pour charger ou lancer l'exécution complète d'un programme de G-codes stocké dans un fichier.

# Conclusion

Ce chapitre est un portail pour notre réalisation et conception matérielle. Ce travail nous a permis de maîtriser les options de notre carte ARDUINO UNO avec circuit de commande et ses caractéristiques afin de les exploiter d'une manière correcte. Nous avons présenté également des principes et des fonctions différentes de la programmation dédiée à la carte ARDUINO UNO, le plus populaire et le plus célèbre de la gamme. Une brève description, ainsi qu'une figure montrant l'interconnexion des autres composants qui permettent de mener et apprendre plus traiter de multiples techniques et périphériques entrées numériques et analogiques.

Nous avons sélectionné la carte ARDUINO UNO pour son aspect économique, sa popularité et sa puissance de traitement. Ensuite nous avons présenté aussi une étude sur les circuits intégrés de commande des moteurs pas à pas utilisées dans notre carte électronique.

Nous avons fait une étude générale du logiciel de commande graphique et le programme de commande de la machine CNC. L'ensemble des outils que nous avons mis au point ont pour la plupart été développés grâce au logiciel Inkscape, pour son utilisation simple et les nombreuses possibilités qu'il offre. La première étape de cette partie consiste en la lecture d'image pour la tracer et la génération fichier G code.

Puis nous avons choisi logiciel G-code Sender pour transfert fichier G-code vers la carte ARDUINO UNO.

Ce chapitre relève ainsi une utilité majeure pour ce qui suit puisqu'il détaille des notions exploitées au sein de la partie réalisation de notre projet

# *III. Chapitre de Simulation et Réalisation de la machine CNC*



# INTRODUCTION :

Ce chapitre est consacré à la simulation et la fabrication d'une machine CNC Pour cette raison, plusieurs étapes ont été envisagées :

Simulation sur logiciel Isis Proteus : pour valider la faisabilité et le bon fonctionnement des circuits déjà développées théoriquement.

Implémentation des circuit simulés sur la plaque à essais (plaquette à trous) afin de s'assurer de son fonctionnement.

Une fois la simulation est bien vérifiée, on passe à l'étape Réalisation de la carte électronique et la soudure des composants sur une carte principale (carte de commande) dont le rôle principal est de gérer les moteurs pas à pas.

Essayé d'employer des matériaux que l'on peut se procurer facilement et des composants les moins chers possibles pour la fabrication de la machine CNC. Pour piloter les moteurs pas à pas des axes X et Y, nous avons utilisé la carte ARDUINO UNO.

Faire un test avec la machine CNC.

## A. *Simulation électrique sous Porteuse Professionnel*

La simulation est un outil utilisé par le chercheur, l'ingénieur, le militaire, etc. pour étudier les résultats d'une action sur un élément sans réaliser l'expérience sur l'élément réel.

Lorsque l'outil de simulation utilise un ordinateur on parle de simulation numérique.

Les chercheurs, les ingénieurs, les militaires et bien d'autres professionnels se posent souvent la question : quel est le résultat que j'obtiens si j'exerce telle action sur un élément ?

Le moyen le plus simple serait de tenter l'expérience, c'est-à-dire d'exercer l'action souhaitée sur l'élément en cause pour pouvoir observer ou mesurer le résultat.

Dans de nombreux cas l'expérience est irréalisable, trop chère ou contraire à l'éthique. On a alors recours à la simulation : rechercher un élément qui réagit d'une manière semblable à celui que l'on veut étudier et qui permettra de déduire les résultats.

Dans notre cas, nous allons utiliser un logiciel très connu des électroniciens « PROTEUS »

### 1. Le logiciel PROTEUS

Proteus est une suite logicielle destinée à l'électronique. Développé par la société Labcenter Electronics, les logiciels incluent dans Proteus permettent la CAO (conception assisté par ordinateur) dans le domaine électronique.

Deux logiciels principaux composent cette suite logicielle :

ISIS principalement connu pour éditer des schémas électriques,  
ARES dédié à la création de circuits imprimés.

Pour la simulation de la commande numérique sur les moteurs responsables des mouvements axiales de la machine CNC, nous allons utiliser le logiciel ISIS de PROTEUS.

### 2. ISIS

Le logiciel ISIS de PROTEUS est principalement connu pour éditer des schémas électriques et de les simuler de manière très facile.

Cette simulation permet, entre autre, de déceler certaines erreurs de conception et de fabrication.

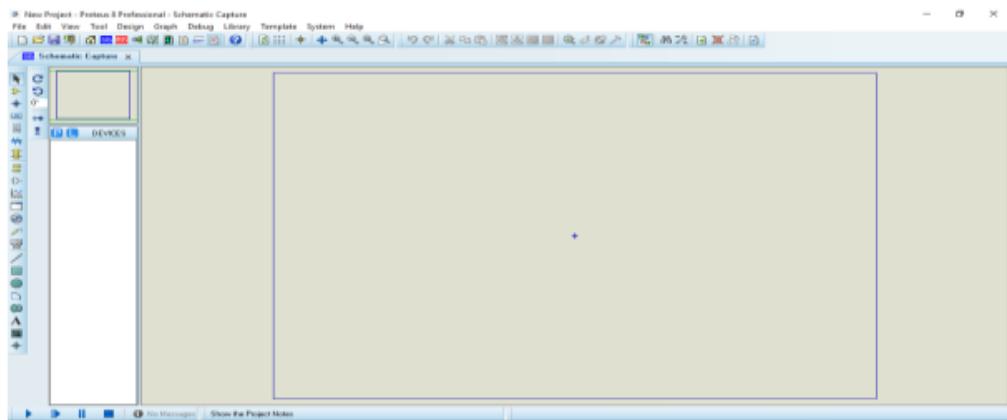


Figure 32. Logiciel ISIS.

### 3. Simulation et test du programme

Plusieurs types de commandes peuvent être envisagés quand il s'agit d'un moteur pas à pas.

Parmi ces commandes nous citons : ) La commande directe avec un amplificateur de la famille L298. ) La commande dir/step avec le couple L297 et L298. Les circuits intégrés L297 et L298 sont les composants les plus utilisés lorsqu'il s'agit de piloter des moteurs pas à pas. Ce sont les composants standards par excellence car ils ont été conçus afin d'être utilisé ensemble mais ils peuvent également être inclus dans un quelconque montage.

Pour saisir le schéma, il faut créer un nouveau projet puis placer les composants qui doivent être sélectionnés à partir de la bibliothèque des composants :

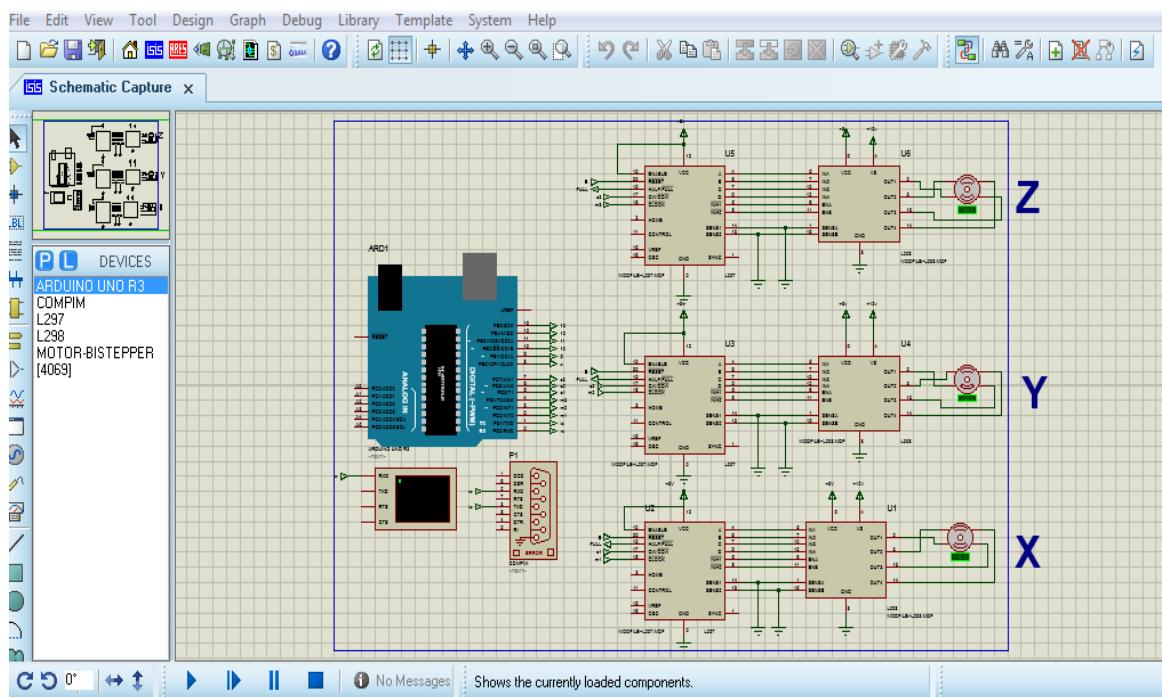


Figure 33. Montage électronique dans Proteus.

### a) Le programme de commande

Le programme de commande des moteurs pas à pas « Grbl » est téléchargé dans l'ARDUINO et pour cela, le programme « GrblUpload » doit être exécuté dans l'application ARDUINO IDE.

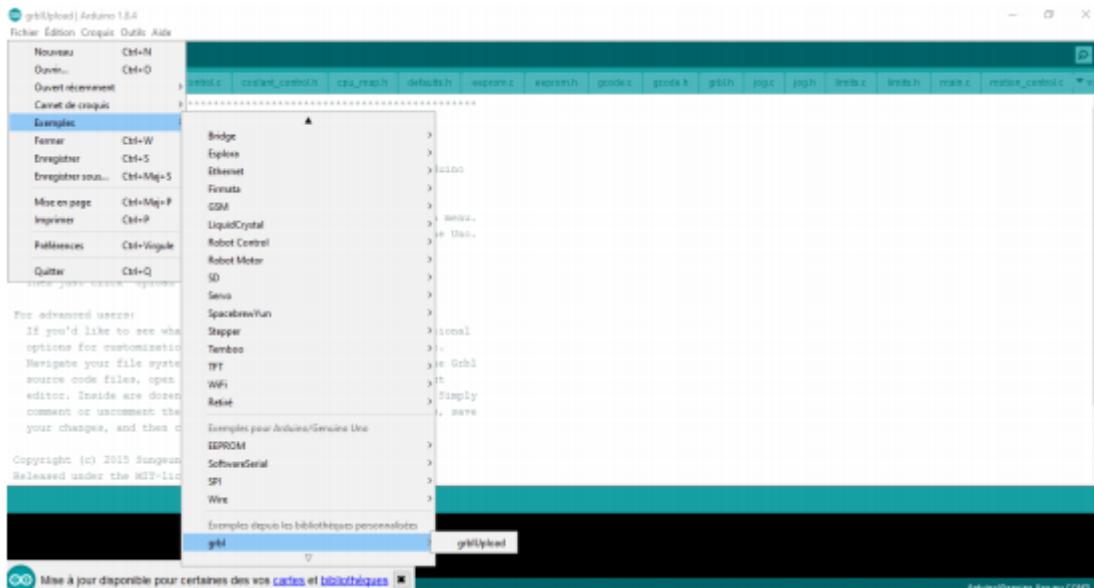


Figure 34. Lancement du GrblUpload.

Pour associer le programme au processeur (ARDUINO), il suffit de charger le fichier HEX généré, en suivant son adresse d'enregistrement, dans l'ARDUINO dans ISIS.

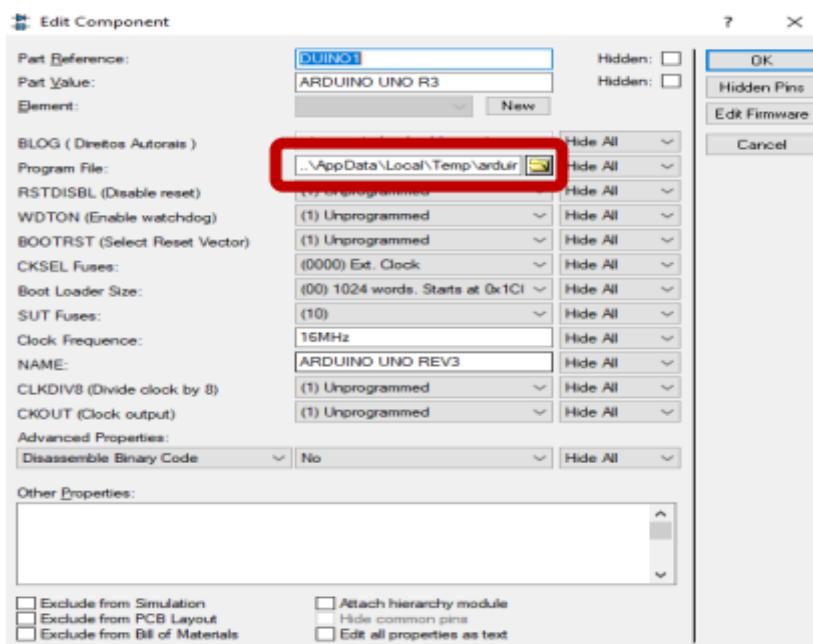


Figure 35. Chargement du programme dans l'ARDUINO.

### b) Le logiciel de commande « Universel G-code Sender »

Universel G-code Sender est un lecteur de G-code libre pour piloter les moteurs des machines à commande numérique qui fonctionne en Grbl.

UGS propose plusieurs façons de commander la machine, soit en écrivant la commande (G-code), en important directement un fichier G-code ou bien en contrôlant les moteurs



Figure 36. Commande en mode manuel

### c) Virtual Serial Port Emulateur

VSPE est un émulateur de port série virtuel. Le port série virtuel COM1 est créé pour connecter le logiciel de commande à l'ARDUINO.

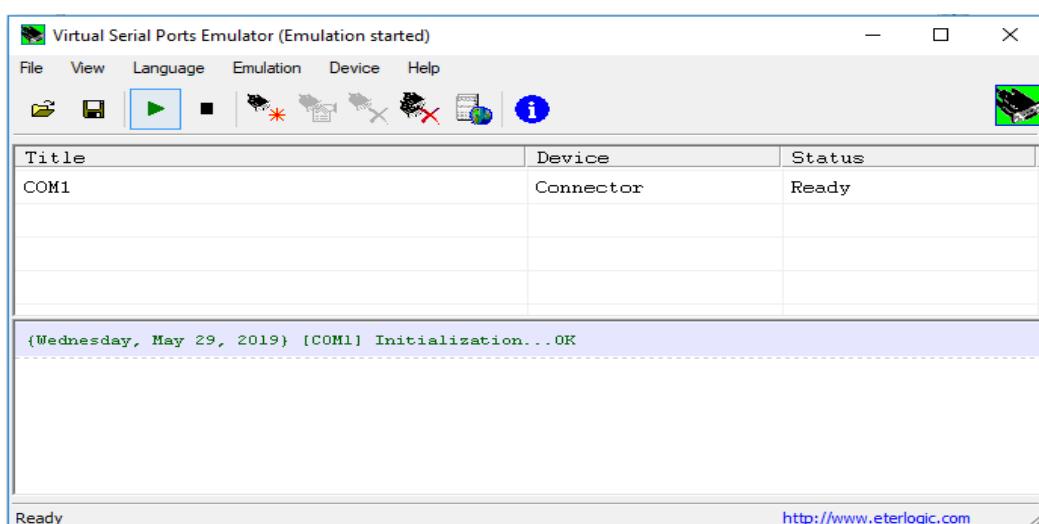


Figure 37. Interface de VSPE.

### d) Lancement de la simulation

Au lancement de la simulation, on obtient les résultats suivants :

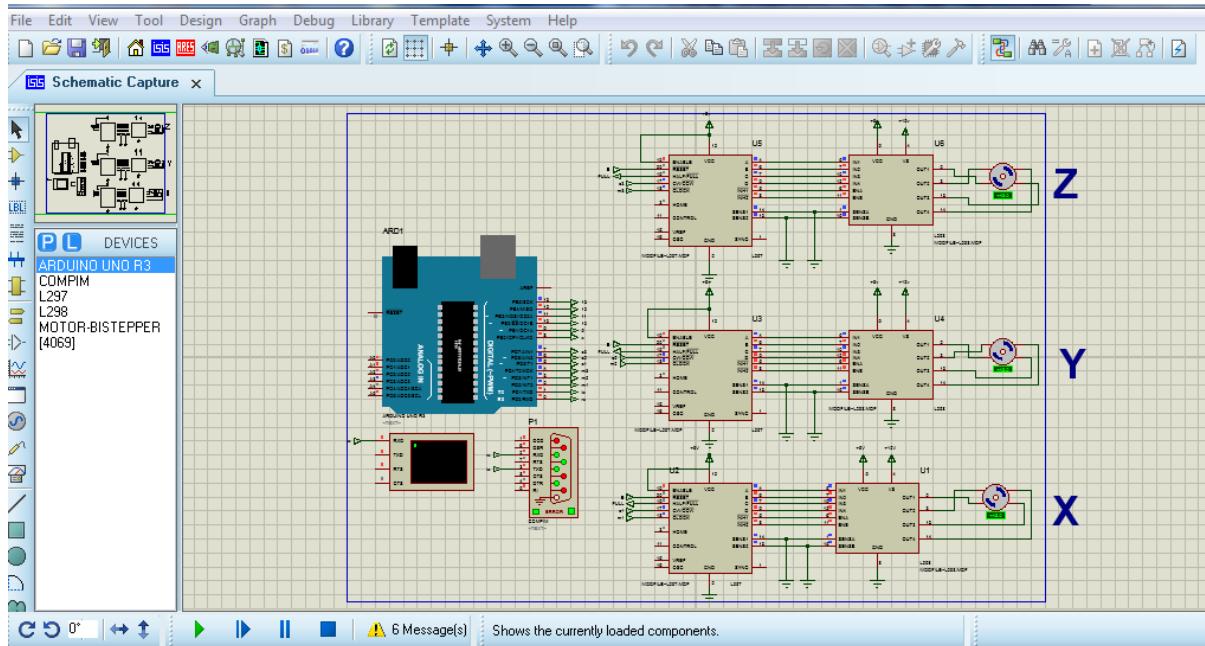


Figure 38. Lancement de la simulation on ISIS

Après l'importation du fichier g-code, préalablement créé, on appuie sur le bouton 'Visualiser' après sur 'Envoi' pour lancer la fabrication automatique.

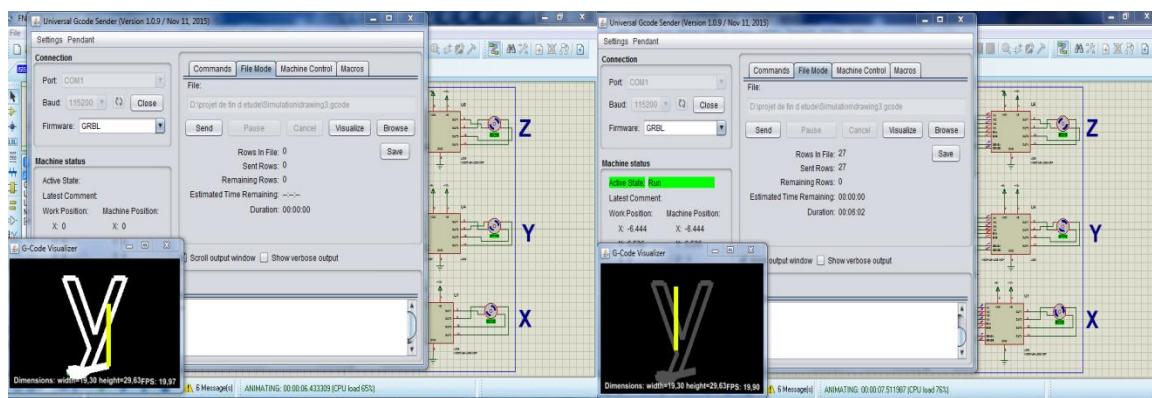


Figure 39. Visualisation du processus d'envoi du G-code

## B. MISE EN MARCHE DE LA MACHINE

Ce chapitre concerne la réalisation d'une machine CNC avec la carte Arduino qu'on a déjà étudié dans la partie de simulation sur ISIS.

### 1. Les composantes

Les composantes utilisées pour la réalisation de la machine CNC sont:

- Arduino UNO (Figure 40): est une carte électronique basée autour d'un microcontrôleur et de composants minimum pour réaliser des fonctions plus ou moins évoluées à bas cout.
- Source D'alimentation « 12V - 1000 mA » (Figure 45).
- Moteur pas à pas 28BYJ-48 (Figure 41): Moteur pas-à-pas unipolaire à 4 enroulements avec réducteur et driver basé sur un ULN2003A permettant de contrôler directement le moteur avec un microcontrôleur (ex: Arduino) sans interface de puissance.

#### Caractéristiques:

- *Alimentation : 5 Vcc*
- *Résistance: 21 ohms*
- *Intensité: 25 mA*
- *Réduction: 1/64*
- *Nombre de pas par tour: 64 (réduction de 4096 en sortie d'axe)*
- *Entraxe de fixation: 35 mm*
- *Axe: 5 mm avec double méplat (épaisseur 3 mm)*
- Moteur Servo (Figure 42): est un système qui a pour but de produire un mouvement précis en réponse à une commande externe, C'est un actionneur (système produisant une action) qui mélange l'électronique, la mécanique et l'automatique.
- CNC Shield (Figure 43): s'imbrique directement sur l'Arduino Uno. Il y a un emplacement prévu pour 4 contrôleurs moteurs du même type que le pilot du moteur pas à pas A4988.
- Le pilot A4988 du moteur pas à pas (Figure 44): est un circuit électronique permettant de contrôler un moteur pas à pas avec différents modes. Son interface de puissance est assurée par des pentes en H réalisée par des MosFets N-channel.



Figure 40. Arduino UNO



Figure 41. Moteur 28BYJ-48



Figure 42. Servomoteur

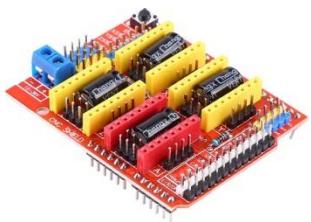


Figure 43. CNC Shield



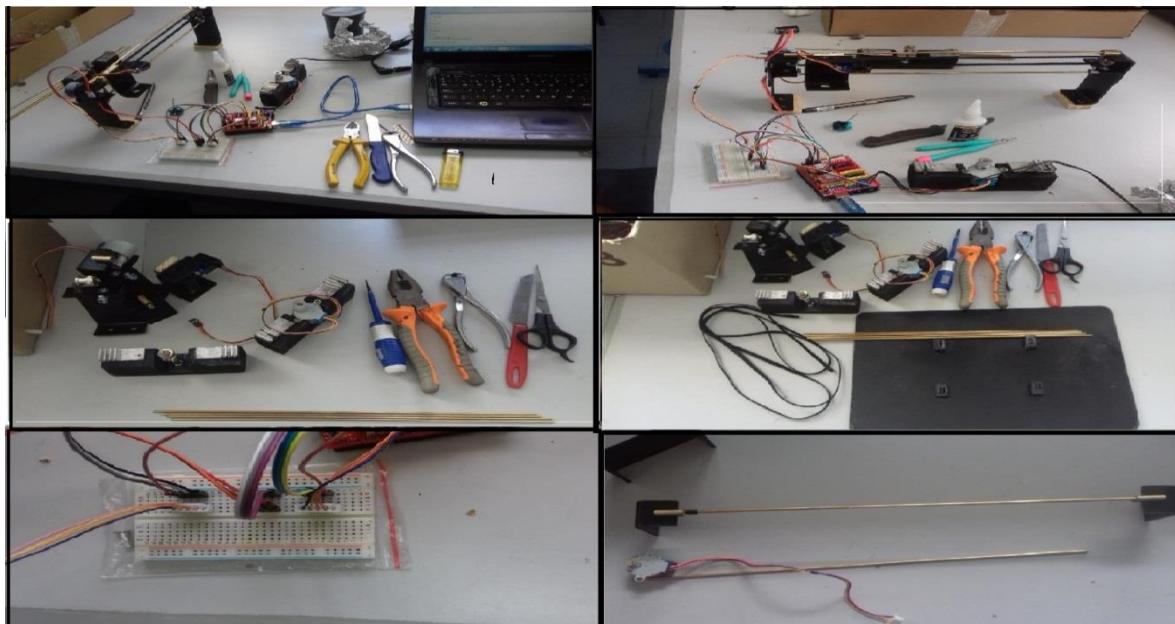
Figure 44. Le pilote A4988 du moteur



Figure 45. Source d'alimentation

## 2. La Réalisation

Ce que nous verrons ci-dessous sont les premières étapes de préparation de la machine CNC.



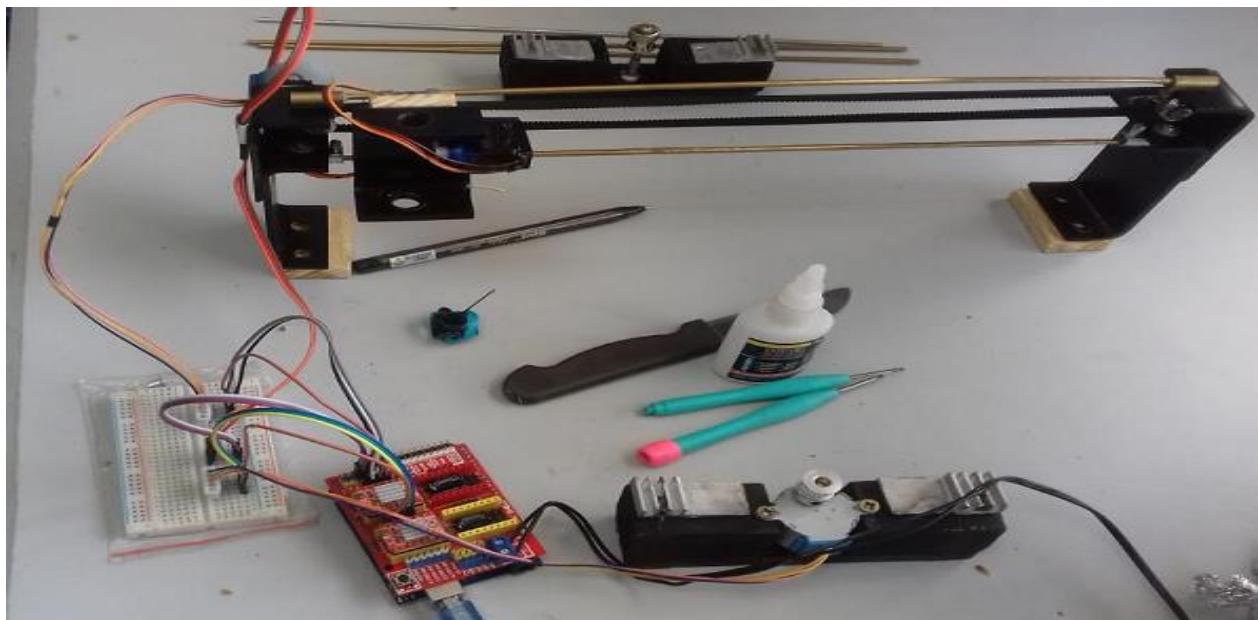


Figure 46. Réalisation sur Plaque d'essai

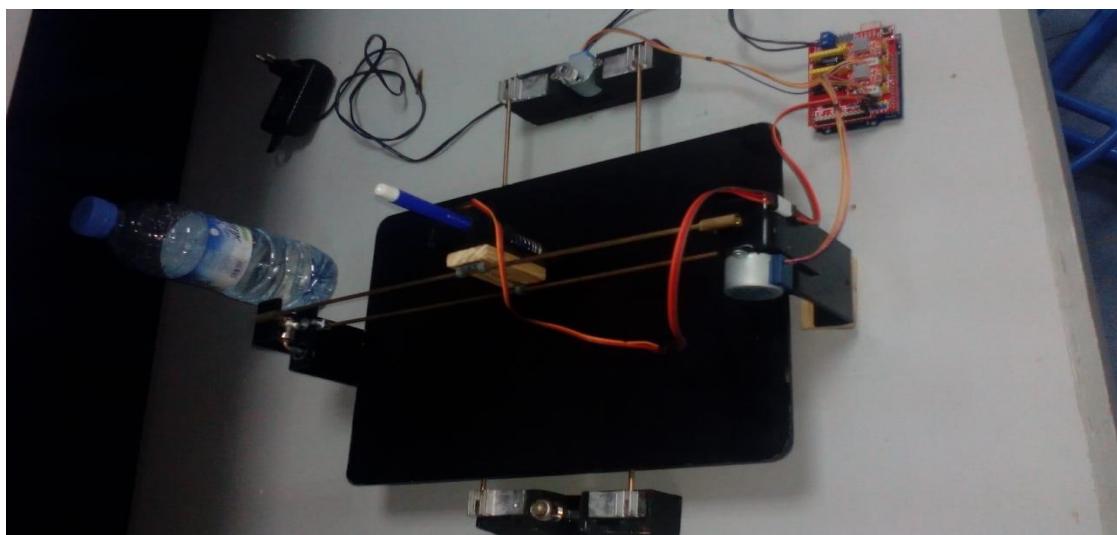


Figure 47. Première essai de réalisation

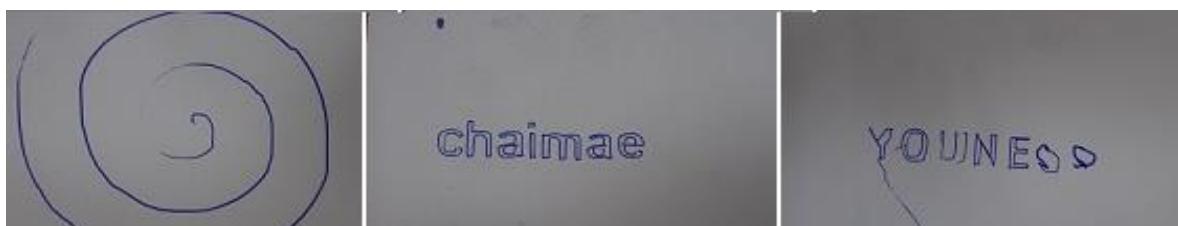


Figure 48. Premier résultat de dessin

Après la construction de la machine nous avons rencontré des problèmes au niveau des chemins de mouvement suivant les axes (X, Y), c'est pour cela nous avons changé les chemins précédents de la figure 47 par celle de la figure suivante:

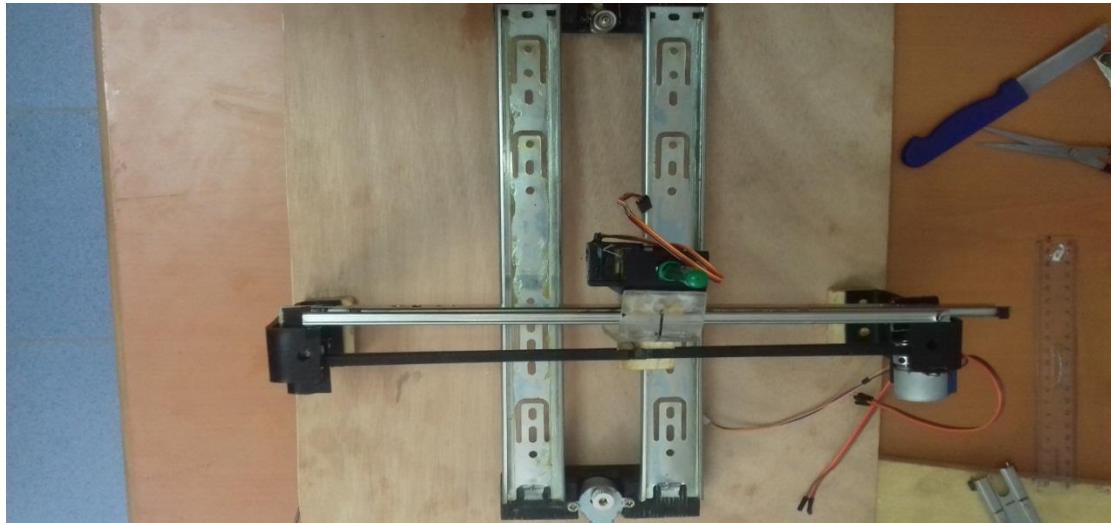


Figure 49. Les nouveaux chemins des axe (X, Y)

❖ Assemblage des Axes (X et Y et Z) de la machine

Commençons par fixer l'axe X et l'axe Y à l'aide d'un adhésif fort.

La hauteur n'est pas très importante, elle sera réglée ultérieurement en ajustant simplement la position du crayon dans son support.

Puis nous fixons une plaque plastique dans l'axe Y, pour fixer l'axe z à cette plaque à l'aide d'adhésif.

❖ Porte crayon (axe Z)

Fixons un support de crayon dans l'axe Z. Ensuite fixons le moteur servo sur le support.

❖ La carte électronique

L'Arduino et le cnc shield prend place à côté des deux moteur des axes (X, Y) .

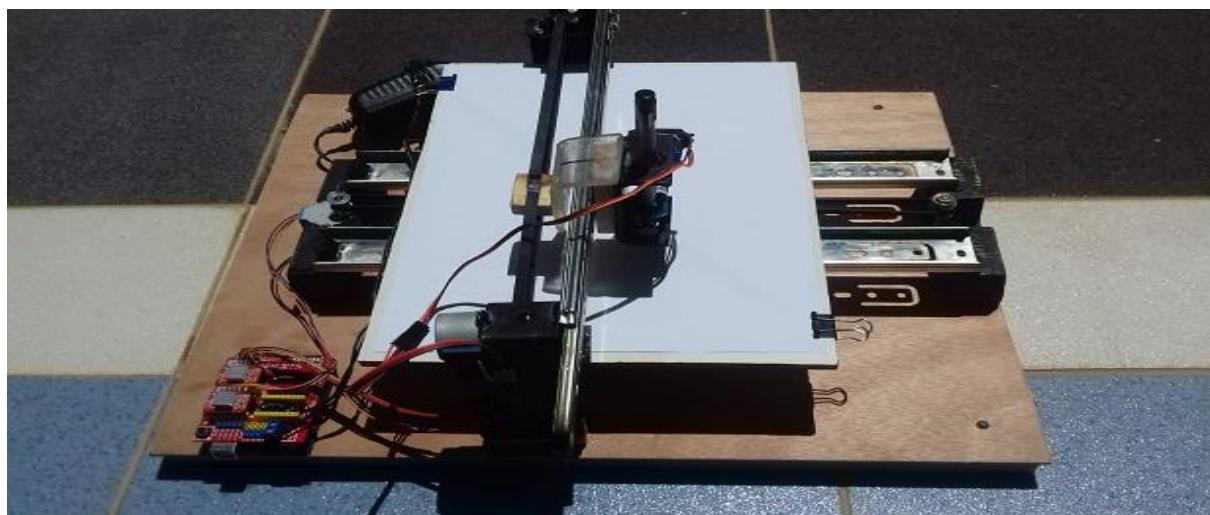


Figure 50. Le montage final du machine CNC

❖ Les paramètres de la machine cnc dans GRBL

Pour que la machine cnc fonctionne comme il faut, il fallait définire des paramètres qui contrôle la vitesse, le retard entre les actions, les distances maximales ( $X_m$ ,  $Y_m$ ) et le nombre des pas/mm.

\$0=10 (step pulse, usec)  
\$1=25 (step idle delay, msec)  
\$2=0 (step port invert mask:00000000)  
\$3=0 (dir port invert mask:00000000)  
\$4=0 (step enable invert, bool)  
\$5=0 (limit pins invert, bool)  
\$6=0 (probe pin invert, bool)  
\$10=3 (status report mask: 00000011)  
\$11=0.010 (junction deviation, mm)  
\$12=0.002 (arc tolerance, mm)  
\$13=0 (report inches, bool)  
\$20=0 (soft limits, bool)  
\$21=0 (hard limits, bool)  
\$22=0 (homming cycle, bool)  
\$23=0 (homming dir invert mask:00000000)  
\$24=25.000 (homming feed, mm/min)  
\$25=500.000 (homming seek, mm/min)  
\$26=250 (homming debounce, msec)  
\$27=1.000 (homming pull-off, mm)  
\$100=1040.000 (x, step/mm)  
\$101=1040.000 (y, step/mm)  
\$102=250.000 (z, step/mm)  
\$110=1010.000 (x max rate, mm/min)  
\$111=1010.000 (y max rate, mm/min)  
\$112=500.000 (z max rate, mm/min)  
\$120=20.000 (x accel, mm/sec<sup>2</sup>)  
\$121=20.000 (y accel, mm/sec<sup>2</sup>)  
\$122=10.000 (z accel, mm/sec<sup>2</sup>)  
\$130=185.000 (x max travel, mm)  
\$131=272.000 (y max travel, mm)  
\$132=200.000 (z max travel, mm).

❖ Le test des axes X / Y / Z

Après le câblage et l'assemblage, nous allons maintenant faire un test de mouvements pour les axes X et Y et Z

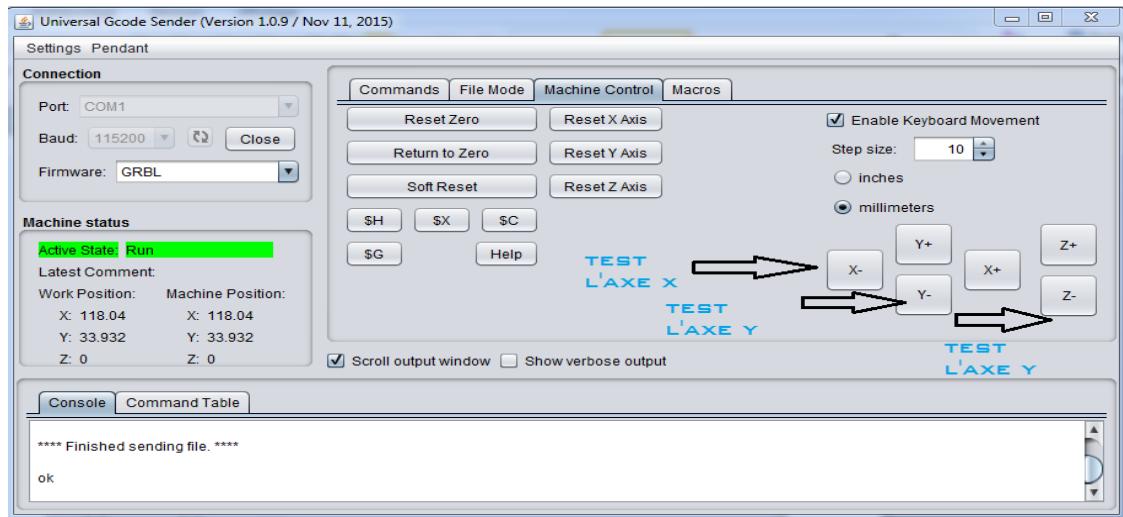


Figure 51. Les buttons du test des axes X, Y et Z

❖ Préparons notre ordinateur à envoyer des fichiers G code

C'est presque prêt ! Tous les axes fonctionnent, la calibration est faite, le programme GRBL est téléchargé dans l'Arduino. Il ne reste plus qu'à préparer notre ordinateur pour envoyer le fichier G-code à notre machin CNC.

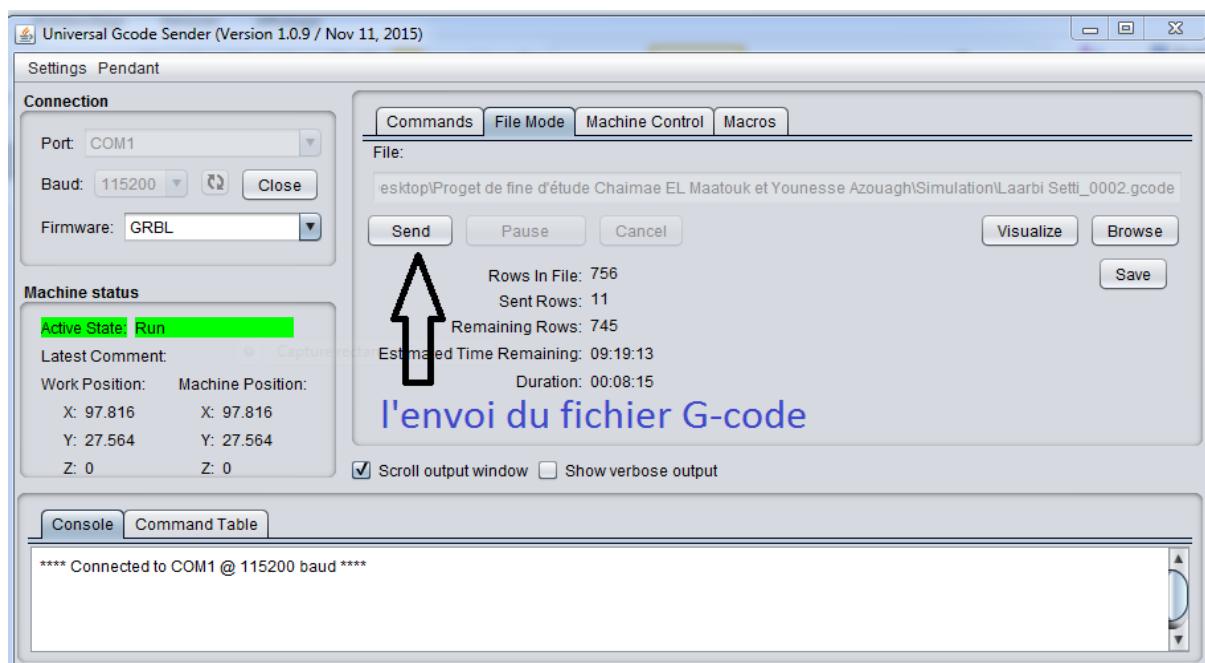


Figure 52. L'envoi du fichier G-code

❖ Traçons notre premier dessin

Avant de tracer notre premier dessin :

- Assurons que le crayon est à l'origine.
- Vérifions que rien n'entrave les mouvements des axes
- Plaçons une feuille sur la zone de dessin.

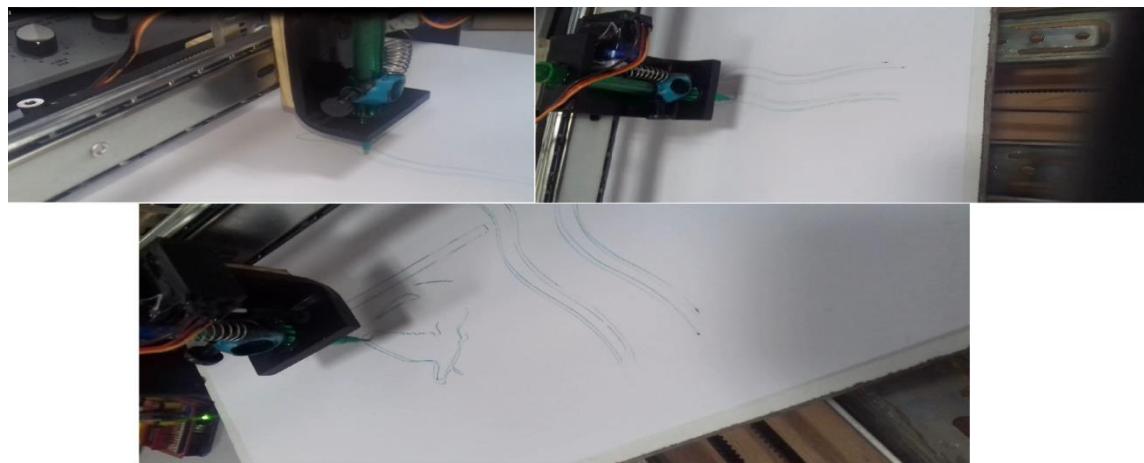


Figure 53. Exemple dessin cnc machine.



Figure 54. Les résultats de dessin de la machine

### 3. Difficultés et problématiques

Durant la processus de simulation et la fabrication de la machine CNC on a rencontré plusieurs problèmes et difficultés qui nous a ouvert les yeux sur la difficulté du projet, parmi ces problèmes on avait la difficulté de :

- Trouver un logiciel capable de simuler les connexion virtuel entre le program responsabl d'envoyer le code G et l'Arduino (Virtual Serial Port Emulateur).
- Générer le code G apartir des images et texts qui nécessite plusieurs taches à suivre (solution : l'integration d'une bibliothèque capable de simplifier la processus en queques clics).
- Trouver les moteurs les moins chères et les plus efficace (convertir le moteur pas à pas 28BYJ-48 unipolaire au bipolaire en enlevant le fil rouge), .
- Trouver un générateur du code G compatible à l'interpréteur Grbl (Inkscape avec la bibliothèque ⇔Grbl 0.9j).
- Calculer et mesurer les paramètres tel que : Le nombre de pas/mm, la vitesse des moteurs maximums, l'angle de rotation convenable du moteur servo pour descendre et ascender le crayon...
- Construire des Chemins capable de faire glisser le porteur de l'axe Z et la zone de dessin de façon lisse et ferme, dont on a fait plusieur essaies commençons par des baguettes de cuivre comme notre premier choi qui n'a pas marcher comme il faut à cause de la frottement et le poid qui faire courber le cuivre, finalement on a choisi d'aller avec des chemins spécial.
- Installer le moteur servo sur l'axe Z à cause d'espace insuffisant et la structure du moteur.
- Les fils du servomoteur ont été coupés de l'intérieur, nous avons donc dû les réparer
- Faire l'équilibre et la précision des mesures.

## Conclusion :

Comme nous avons vu dans ce chapitre, beaucoup des taches ont été réalisées pour construire la machine CNC, nous avons présenté plusieurs types des logiciels utilisés pour la simulation et la commande de la machine. Pour réaliser cette commande on a utilisé une carte microcontrôleur "Arduino Uno" qui possède un espace de programmation qui est très claire et simple et les circuits intégrés CNC Shield et A4988 qui sert à contrôler les moteurs pas à pas de manière efficace.

A l'aide du logiciel " Inkscape " nous avons générer les codes G qu'on fait envoyer à la carte arduino par le logiciel " G-code sender ".

Après plusieurs essais et expérimentations, la machine CNC fonctionne correctement et répond aux différentes commandes.

# Conclusion générale

Durant tout le temps qui nous a été alloué pour effectuer ce travail, nous avons essayé, du mieux qu'on pouvait, d'expliquer avec beaucoup de détail le fonctionnement d'une machine à commande numérique avec tout ce que ce nom englobe comme étapes à étudier.

Nous avons d'abord entamé la partie mouvement qui est basée sur les moteurs pas à pas que nous avons étudié en détail dans le chapitre 1, cette partie est essentielle au fonctionnement des machines à commande numériques.

Ensuite nous avons entamé une étude sur le processus de communication entre la carte de commande et l'ordinateur, ceci a été fait en détail dans le chapitre 2. Aucune machine à commande numérique ne peut fonctionner sans un système qui lui envoie les données à transformer en mouvements. C'est pour cela qu'on a opté pour les logiciels (Inkscape, G-code sender).

Enfin, dans le dernier chapitre nous avons essayé de rassembler ce qui a été étudié dans les chapitres précédent. Nous avons vu les différents type de programmation effectués au niveau du microcontrôleur ainsi qu'au niveau de l'ordinateur et ceci pour former un ensemble homogène qui permettra le bon fonctionnement de notre machine.

Pour conclure, nous souhaiterions qu'il y'ait une continuation pour cette étude par d'autres étudiants afin de concevoir d'autre utilisations possibles pour cette machine.

# Bibliographie

[1] Fonctionnement d'un moteur pas à pas :

<http://spt06.perso.libertysurf.fr/pas.htm>

[2] Wikipédia Moteur pas à pas :

[https://fr.wikipedia.org/wiki/Moteur\\_pas\\_%C3%A0\\_pas](https://fr.wikipedia.org/wiki/Moteur_pas_%C3%A0_pas)

[3] Cours moteurs pas a pas :

[http://c.divoux.free.fr/phyapp/moteur\\_pas\\_a\\_pas/cours\\_moteur\\_pas\\_a\\_pas.pdf](http://c.divoux.free.fr/phyapp/moteur_pas_a_pas/cours_moteur_pas_a_pas.pdf)

[4] Système de commande pas a pas :

[http://col2000.free.fr/pasapas/pap\\_indx.htm](http://col2000.free.fr/pasapas/pap_indx.htm)

[5] « Systèmes électromécaniques » ; Haute Ecole d'ingénierie et de Gestion Du Canton du Vaud, CD/SEM/Cours/Chap07 . M. Correvon.

[6] <http://www.ELECTRONIQUEPRATIQUE.com>

[7] <http://arduino.cc/en/Main/ArduinoBoardMega2560>

[8] [www.reality.be/elo/labos2/files/AtMega32DocFr.pdf](http://www.reality.be/elo/labos2/files/AtMega32DocFr.pdf)

[9] [www.jp79dsfr.free.fr/Elec%20\\_%20Arduino%20%20Memo%20et%20condense.pdf](http://jp79dsfr.free.fr/Elec%20_%20Arduino%20%20Memo%20et%20condense.pdf)

[10] <https://www.arduino.cc/en/Main/ArduinoBoardUno>

[11] PATRIQUE OGUIC « Moteur pas à pas et PC deuxième édition » dunod ,Paris,2004

[12] <http://webge.github.io/EasyDriverStepperMotor/>

[13] Steve Krar, Arthur Gill, «Computer numerical control programming basics»

[14] <http://inkscape.fr/>

[15] <http://www.civade.com/post/2014/01/02/Arduino-et-GRBL-l-incontournable-solutionpour-piloter-une-petite-CNC> ,2016.

[16] Torjus Spilling « Self-Improving CNC Milling Machine »,master ,university oslo,2014

[17] Arduino Lesson . The Serial Monitor Created by Simon Monk 2013-06-22

[18] John Kennedy «Bresenham's Integer Only Line Drawing Algorithm»

[19] Electronique et loisir, Février 2000,numéro 9 <http://www.cncloisirs.com/>