

République Algérienne Démocratique et Populaire
Ministère de L'Enseignement Supérieur et de la Recherche Scientifique
Université Mouloud Mammeri de Tizi-Ouzou



Faculté de Génie Electrique et Informatique
Département d'Electronique

Mémoire de Master en Electronique
Spécialité : Electronique Industrielle

Thème

Etude et conception d'une machine CNC
(Découpeuse laser)

Présenté par:

Chanez Guerrouabi

Roza Ait Rahmane

Dirigé par:

M.Bennamane. K

M.Kroubi.T

Mémoire soutenu publiquement le 19/09/2018 devant le jury composé de :

Président

M.Laghrouche

Examineur

M.Hamiche

Promotion : 2017/2018

Remerciements

Nous tenons tout d'abord à remercier Dieu le tout puissant et miséricordieux, qui nous a donné la force et la patience d'accomplir ce modeste travail.

En second lieu, nous tenons à remercier chaleureusement notre encadreur Monsieur BENAMANE pour ces précieux conseils et ses aides toute au long de notre travail ainsi que monsieur KERUBI pour son aide précieuse pour la réussite de ce modeste travail.

Nous remercions les membres du jury de leur présence et d'avoir accepté d'évaluer et d'examiné notre modeste travail

Nous tenons à exprimer nos sincères remerciements à tous les professeurs qui nous ont enseigné et qui par leurs compétences nous ont soutenu dans la poursuite de nos études.

Enfin, nous tenons également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail ainsi que nous respectable familles.

Merci à tous et à toutes.

Dédicaces

Je dédie ce modeste travail à mes cher parents qui m'ont toujours soutenues durant mon parcours.

Je le dédie aussi à mon frère Tahar et ma sœur Sandra

A ma grande mère Denise

A ma belle sœur Katia et sa famille

A mes oncles, tantes, cousins et cousines

A toutes mes copines en particulier Fariza, Thiziri, Rezika, Wassila, Djazia...

A tous mes amis en particulier Boa, Ghiles.z, Fiqu, Ghiles.g...

A ma deuxième famille croissant rouge CL Mekla

A tous les étudiant de master électronique toutes les spécialités.

2017/2018

Guerrouabi Chanez

Dédicaces

*Je dédie ce modeste travaille àmes chers parents qui m'ont
toujours soutenues durant mon parcours.*

Je le dédie aussi à mon frère yacine et mes sœurs

A mes oncles, tantes, cousins et cousines

A mes deux copines celia, dihia et tous mes amis

A tous les étudiant de master électronique toutes les spécialités.

2017/2018

A97 RAHMANE Roza

Sommaire

Introduction	1
Chapitre I : Généralités sur les machines CNC	
I.1 Historique.....	2
I.2. définition de la commande numérique	3
I.3. Définition d'une machine CNC	3
I.4. Principe de fonctionnement d'une machine CNC	3
I.5 Domaine d'utilisation	5
I.6 les différents types de machines CNC	5
I.7. Classification des MOCN	6
I.7.1. Classification des MOCN selon le mode de fonctionnement	6
I.7.1.1. Fonctionnement en boucle ouvert.....	6
I.7.1.2. Fonctionnement avec commande adaptative.....	6
I.7.1.3. Fonctionnement en boucle fermé.....	7
I.7.2. Classification des MOCN selon le mode d'usinage.....	7
I.7.2.1. Commande numérique point à point	8
I.7.2.2. Commande numérique par axiale	8
I.7.2.3. Commande numérique de contournage	9
I.7.3. Classification des MOCN selon le nombre d'axe	9
I.9. la structure d'une machine CNC	9
I.9.1. le système électronique	10
I.9.2. le système mécanique.....	10
I.10. Avantages des machines CNC :	11
I.11. inconvénients des machines CNC	11

Chapitre II : Conception d'une machine CNC

II.1 Architecture de la graveuse laser :	12
II.2 Caractéristiques de la graveuse laser :	12
II.2.1 Partie logicielle	12
II.2.1.1 Arduino :	12
II.2.1.1.1 Présentation	12
II.2.1.1.2 Langage de programmation :	14
II.2.1.1.3 Exemple de programme sur Arduino :	16
II.2.1.1.4 La structure d'un programme	16
II.2.1.1.5 Coloration syntaxique.....	17
II.2.1.1.6. Ponctuation.....	18
II.3 G-code	18
II.3.1 Présentation.....	18
II.3.2 L'éditeur de texte ou image compilateur de langage CNC.....	18
II.3.3 Définition de l'Inkscape.....	19 ²
II.3.4 Logiciel de pilotage de la machine CNC	19
II.4 GRBL	20
II.4.1 Présentation :	20
II.4.2 Fonctionnalités	20
II.2.2 Partie matériels.....	22
II.2.2.1 Présentation de la carte Arduino uno.....	22
II.2.2.2 Description matérielle	23
II.2.2.2.1 Le Microcontrôleur ATmega328	23
II.2.2.2.2. Les sources de l'alimentation de la carte	24

II.2.2.2.3 Les entrées & sorties.....	25
II.2.2.2.4 Les ports de communications	26
II.5 Les avantages et les inconvénients de la carte.....	27
II.5.1 Avantages.....	27
II.5.2 Inconvénients	27
II.6 moteur pas a pas :	28
II.6.1 Généralités	28
II.6.2 Les différents types de moteur pas à pas	28
II.6.2.1 Les moteurs à réluctance variable	28
II.6.2.2 Moteur à aimants permanents.....	29
II.6.2.2.1 Moteur à aimant permanent bipolaire.....	29
II.6.2.2.2.Moteur à aimant permanent unipolaire :.....	30
II.6.2.3 Moteur pas-à-pas hybride MH	31
II.6.2.3.1 Technologie des moteurs pas à pas hybrides.....	31
II.6.3 Comparaison entre les différents types des moteurs pas à pas	32
II.7 Driver L293D.....	32
II.7.1 Présentation :.....	32
II.7.2 Brochage	33
II.7.3 Caractéristiques	34
II.8 Diode laser.....	35
II.8.1 Présentation.....	35
II.8.2 Fonctionnement électrique	36
II.8.3 Courant de seuil d'une diode laser	36

Chapitre 03 : Réalisation d'une machine cnc

III.1 Les différentes étapes à suivre pour construire la machine CNC :	37
Etape 01 : les éléments nécessaires pour le fonctionnement de notre machine CNC :	37
Etape 02 : démonter le lecteur DVD	37
Etape 3 : Soudage des pattes du moteur pas à pas	38
Etape 4 : Fabrication de la structure en contreplaqué :	38
II.2 Branchement du L293D avec la carte Arduino et le moteur pas à pas	38
III.3 Pilotage de deux moteurs pas à pas via la carte Arduino :	40
III.4 Programmation :	42
III.5 le montage de la machine CNC	42
III.6 Traçage du premier dessin	44

Liste des figure

Figure I.1 Décomposition d'une machine numérique.	4
FigureI.2:machine à trois axes.....	4
Figure. I. 3.Domaine d'utilisation des M.O.C.N.....	5
Figure I.4. Fonctionnement en boucle ouvert.	6
Figure I.5. Commande adaptative.....	7
Figure I.6. Commande en boucle fermée.	8
Figure I.7. Commande Numérique point à point.	9
Figure I.8. Commande par axiale.....	9
Figure I.9. Commande numérique de contournage.	10
Figure II.1 : différentes parties d'une interface Arduino.....	13
Figure II.2 : Programme pour clignoter une LED.	16
Figure II.3: structure d'un programme.....	17
Figure II.5: Universal Gcode Sender.....	19
Figure II.6 : carte Arduino Uno.....	22
Figure II.7 : ATmega 328.....	23
Figure II.8 : image illustrant la carte Arduino Uno avec ces différents blocs.....	27
Figure II.9 : moteur à réluctance variable.	29
Figure II.10 : moteur bipolaire.	29
Figure II.11 : moteur unipolaire.....	30
Figure II.12 : moteur hybride (MH).....	31
Figure II.13 : différentes broches du L293D.....	33
FigureIII.1: Système de déplacement linéaire récupéré sur un lecteur de DVD.....	37

Figure III.2: Soudage des fils au niveau du moteur.....	38
Figure. III.3 : support de la machine	38
Figure III.4 : Branchement de deux drivers L293D	40
Figure III.5 : montage des deux moteurs sur isisproteus.	41
Figure III.6 : l'assemblage du support avec les lecteurs.....	43
Figure II.7 : exemple d'une forme réalisée par la machine.....	44

LISTES DES TABLEAUX

Tableau II.1 : Fonctions des icônes de la barre des boutons de l'IDE	14
TableauII.2 : codes de bases du G-code.....	21
TableauII.3 comparissant entre les différents types des moteurs pas à pas.....	32
Tableau II.4 : descriptif des broches du L293D	34
Tableau III.1 : branchement du l293D.....	39

INTRODUCTION

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 industrielle, ce qui a permis le développement de solutions numériques efficaces avec la possibilité d'implanter des algorithmes plus complexes. Après une première génération de commandes numériques à logique câblée, sont apparues les commandes numériques par ordinateur (CNC), ou par ordinateur.

Une machine CNC est tout d'abord une machine-outil : elle permet selon ses caractéristiques d'effectuer diverses opérations (percer, scier, rectifier, découper, fraiser, plier, graver, etc.) nécessitant des gestes précis et répétitifs, sur des matériaux divers. La machine est équipée d'un ou plusieurs outils qui se déplacent le long d'axes (habituellement X, Y et Z), et l'ensemble de ces opérations est programmé à l'avance à l'aide d'un logiciel, que la machine exécute ensuite.

Pour la bonne exécution de notre projet, on a opté dans notre rédaction pour le plan suivant :

- Le premier chapitre est consacré à une étude générale sur la machine CNC.
- Le deuxième chapitre présente la conception de notre propre machine CNC
- Le troisième chapitre est dédié à la réalisation de la machine CNC.

Et on terminera notre travail par une conclusion et des perspectives futures.

CHAPITRE I

GÉNÉRALITÉS SUR LES MACHINE CNC

I.1 Historique

Il était en 1947 lorsque le contrôle numérique est né. Il a commencé quand John C. Parsons de la Parsons Corporation, Traverse City, Michigan, un fabricant de pales de rotor d'hélicoptère, ne pouvait pas faire ses modèles assez vite. Ainsi, il a inventé un moyen de couplage matériel informatique avec un foret de gabarit. M. Parsons a utilisé des cartes perforées pour faire fonctionner son chiffre système Ron. 1949 a été l'année d'un autre « besoin urgent ». La commande Matériel US Air a réalisé que les pièces pour ses avions et missiles ont été de plus en plus complexes. En outre, comme les dessins ont été constamment améliorés, les changements du dessin ont souvent été faits. Ainsi, dans leur recherche de méthodes de production plus rapide, un contrat d'étude Air Force a été attribué à la Parsons Corporation. Les mécanismes d'asservissement laboratoire de l'Institut de technologie du Massachusetts (MIT) étaient le sous-traitant. En 1951, le MIT a repris le travail complet, et en 1952, le prototype de la machine à commande numérique d'aujourd'hui, une machine Cincinnati HydrogelMilling modifié, a été démontrée avec succès. Le terme contrôle numérique a été l'origine du MIT.

I.2. définition de la commande numérique

La commande numérique est un mode de commande dans lequel les valeurs désirées d'une variable commandée sont définies selon un code numérique (la machine-outil constitue le principal domaine d'application de la commande numérique).

C'est une somme d'automatismes dans laquelle les ordres de mouvement ou de déplacement, la vitesse de ces déplacements et leur précision, sont donnés à partir d'informations numériques. Ces informations sont codées sur des supports tels que :
Rubans perforés, cassettes ou disquettes magnétiques ou simplement sauvegardés en « mémoires » dans le cas des dernières générations de commandes numériques à calculateur intégré (CNC).

I.3. Définition d'une machine CNC

C'est une machine totalement ou partiellement automatique à laquelle les ordres sont communiqués grâce à des codes qui sont portés sur un support matériel (USB). Lorsque la machine-outil est équipée d'une commande numérique capable de réaliser les calculs des coordonnées des points définissant une trajectoire, elle est appelée CNC.

Le premier rôle d'une machine CNC est de générer des mouvements, elle recevra des valeurs de positionnement, de vitesse et d'accélération et générera suite à un traitement des consignes numériques en sortie.

I.4. Principe de fonctionnement d'une machine CNC

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 I.1):

- La partie opérative.

- La partie commande.

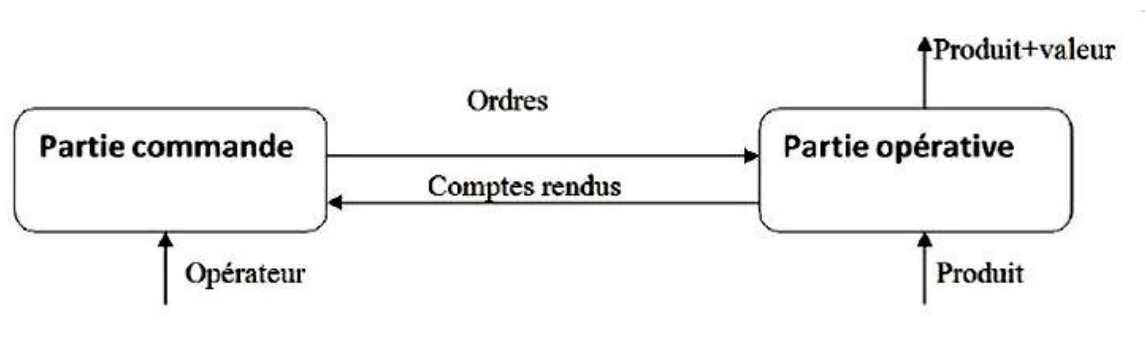


Figure I.1 Décomposition d'une machine numérique.

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 I.2) :

- 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.

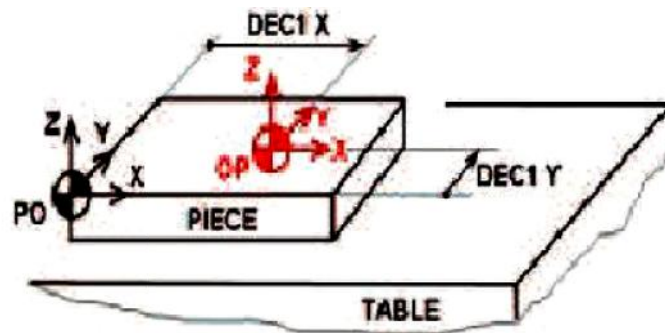


Figure I.2 : machine à trois axes

I.5 Domaine d'utilisation

Les MOCN sont employées dans de nombreux secteurs industriels (Métallurgie, Bois, Textile). Elle est aussi associée à de nouvelles technologies de façonnage (Laser, Électroérosion, Jet d'eau). Les principaux procédés de fabrication sont concernés :

- Perçage, taraudage ;
- Tournage, alésage ;
- Fraisage ;
- Rectification ;
- Oxicoupage, soudure en continu, par points ;
- Poinçonnage, cisailage.

I.6. les différents types de machines CNC

On distingue plusieurs types de machines :

Les machines à enlèvement de copeaux : les perceuses, les tours 2 et 4 axes, les centres de tournages 5 axes, les fraiseuses 2 axes, 3 axes, les centres d'usinage ,3 à 5 axes, les rectifieuses, les affuteuses, les machines d'usinage à très grande vitesse.

- Les électroérosions : les machines à enfonçages, les machines à fil.
- Les machines de coupes : oxycoupage, laser, jet d'eau
- Les presses : métal, injection plastique.
- Les machines à bois : à portique ou col de cygne.
- Les machines spéciales : à panneaux, à têtes multiples, de conditionnement (pour l'agroalimentaire)...

I.7. Classification des MOCN

Les machines-outils à commande numérique (MOCN) sont classées suivant le :

- Mode de fonctionnement de la machine.
- Nombre d'axes de la machine.
- Mode d'usinage.

I.7.1. Classification des MOCN selon le mode de fonctionnement

I.7.1.1. Fonctionnement en boucle ouvert

En boucle ouverte, comme l'illustre la (**Figure I.12**), le système assure le déplacement Du chariot mais ne le contrôle pas.

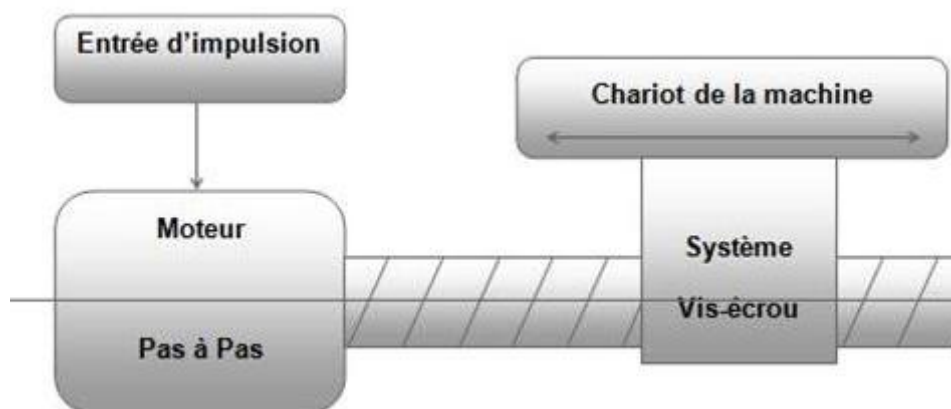


Figure I.12. Fonctionnement en boucle ouvert.

I.7.1.2. Fonctionnement avec commande adaptative

La commande adaptative réalise d'une façon continue et automatique l'adaptation des Conditions de coupe. Des capteurs relèvent les valeurs de couple de la broche, l'amplitude de vibration de la broche, la température au point de coupe. Ces informations sont transmises à une Unité spéciale qui les envoie vers le directeur de commande numérique qui agit selon l'analyse

Des informations sur les conditions de coupe pour permettre une meilleure qualité de travail, une meilleure productivité et une plus grande sécurité.

La (**Figure I.13**) illustre le fonctionnement de la commande adaptative.

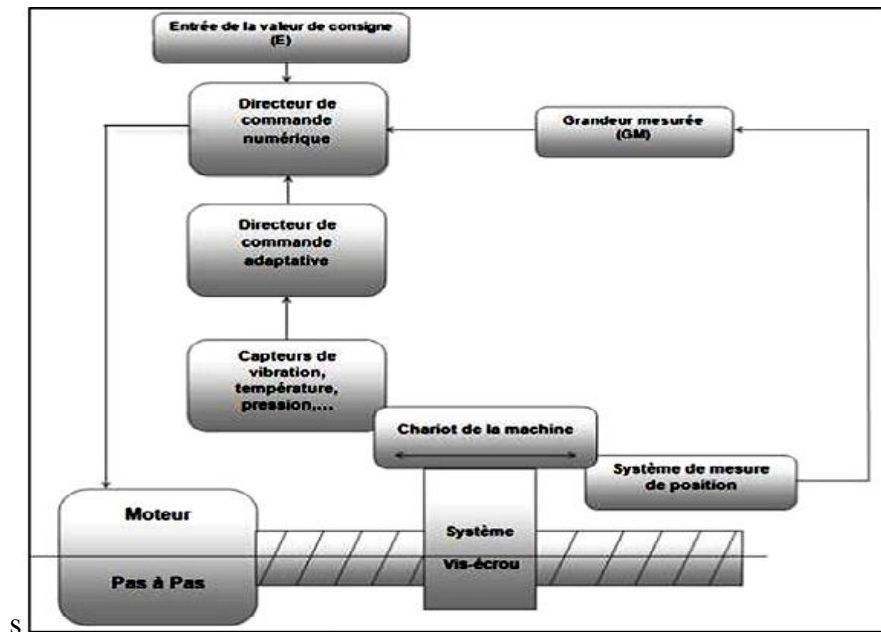


Figure I.13. Commande adaptative.

I.7.1.3. Fonctionnement en boucle fermé

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).comme illustre la (Figure I.14)

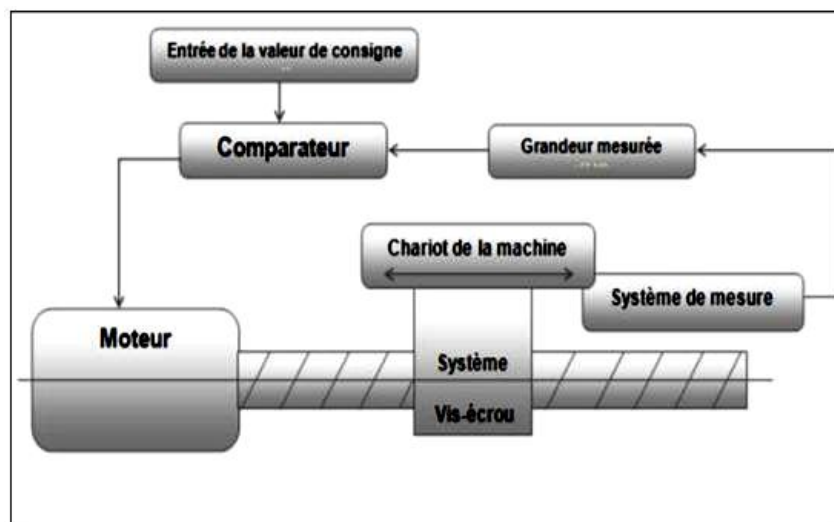


Figure I.14. Commande en boucle fermée.

I.7.2. Classification des MOCN selon le mode d'usinage

Selon le mode d'usinage on peut classer les MOCN en trois catégories :

- Commande numérique point à point.

- Commande numérique par axiale.
- Commande numérique de contournage.

I.7.2.1. Commande numérique point à point

C'est la mise position de l'outil ou de la pièce Par déplacements non synchronisés. Le Mouvement de coupe (usinage) n'est possible que lorsque le mouvement de positionnement. Exemples d'opération d'usinage : perçage, alésage, lamage taraudage, petit fraisage.

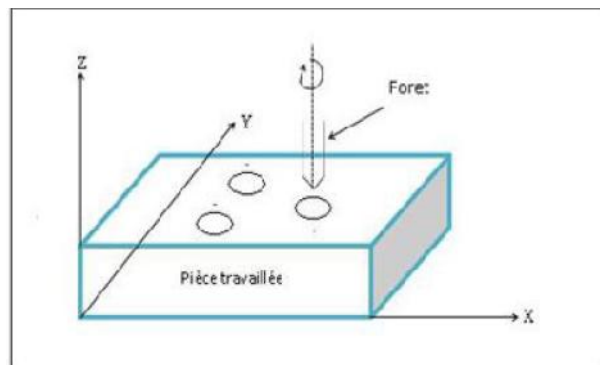


Figure I.18. Commande Numérique point à point.

I.7.2.2. Commande numérique par axiale

Ce sont des déplacements parallèles aux axes avec les vitesses d'avance programmées. Le mouvement de coupe et de positionnement sont synchronisés de façon à avoir un usinage Selon des trajectoires parallèles aux axes de déplacement.

Exemples d'opération d'usinage : tournage, fraisage, alésage.

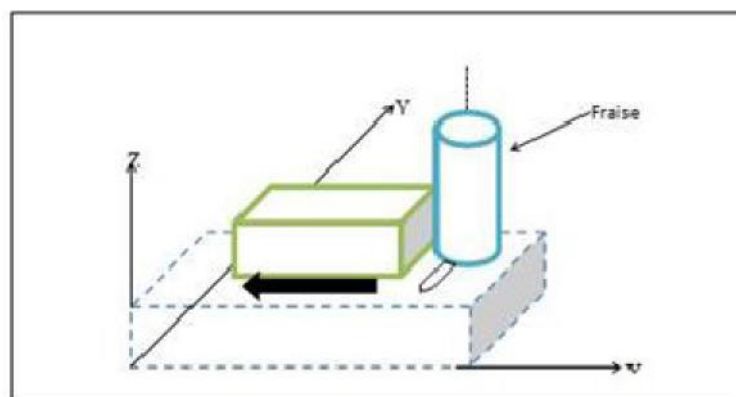


Figure I.19. Commande par axiale.

I.7.2.3. Commande numérique de contournage

Ce sont des déplacements synchronisés des divers axes avec la vitesse d'avance programmée. Les trajectoires sont décomposées en éléments de droites ou de cercles dans un ou plusieurs plans.

Exemples d'opération d'usinage : toute opération possible sur un centre de tournage ou Centre d'usinage.

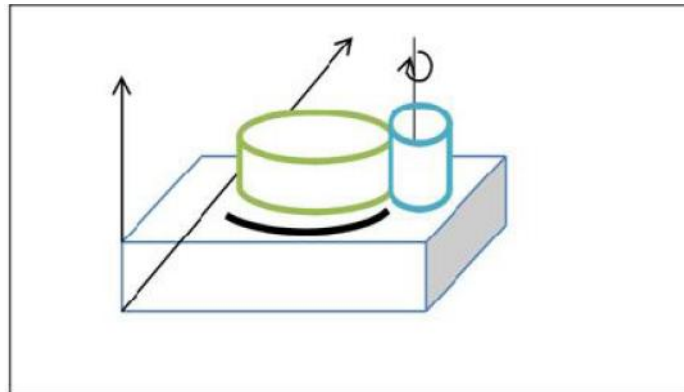


Figure I.20. Commande numérique de contournage.

1.7.3. Classification des MOCN selon le nombre d'axe

Les possibilités de travail des MOCN s'expriment en nombre d'axes de travail.

Un axe définit toute direction principale suivant laquelle le mouvement relatif de l'outil et de la pièce a lieu lorsqu'un seul des moteurs de déplacement fonctionne avec contrôle numérique continu.

Un demi-axe définit la direction dans laquelle l'avance n'est pas contrôlable numériquement mais contrôlé par pistes, cames ou plateaux diviseurs.

I.8.1a structure d'une machine CNC

Ce système peut être divisé en trois parties. Le système mécanique qui reçoit des signaux de commande nécessaire au système électronique qui aboutit finalement à l'actionnement souhaité des moteurs. Le système électronique obtient une commande ou un ensemble de commandes du système logiciel et génère des commandes pour le système mécanique.

I.8.1. le système électronique

Le système électronique est responsable de la génération du signal de commande pour les moteurs qui guide le mouvement de la trajectoire de l'outil dans chaque direction ou axe. Le système électronique est composé de :

- L'alimentation
- La carte microcontrôleur
- La carte de commande des moteurs pas à pas

I.8.2.le système mécanique

Le système mécanique est assemblé de telle manière que le mouvement des 3 axes soit réalisé en utilisant des rails linéaires assemblés avec des roulements linéaires, les moteurs sont montés chacun sur un axe qui est source de mouvement et agit en fonction du signal de commande généré à partir du circuit électronique.

Chaque moteur est relié à une vis sans fin de chaque axe qui est chargé de transformer le mouvement du moteur de rotation en mouvement linéaire.

Le mouvement contrôlé de chaque axe est obtenu directement par la commande de la rotation du moteur. La vitesse du mouvement de chaque axe peut également être contrôlée par le contrôle direct de la vitesse du moteur, en donnant des signaux de commande nécessaires. Ainsi, la trajectoire de l'outil fixée à l'organe terminal est contrôlée dans chaque axe pour une action sur la pièce à usiner.

Pour éviter que les axes sortent de la plage de fonctionnement, on utilise des fins de course pour chaque axe. Une commande manuelle de bouton d'arrêt d'urgence peut également être utilisée pour la machine.

I.8.3.le système de programmation

La pièce à usiner sera conçue dans une conception assistée par ordinateur (CAO), dont la sortie est un dessin dans un des nombreux formats acceptables, le format le plus préférable est le format .stl. Ce dessin est ensuite introduit dans le logiciel de fabrication assistée par ordinateur (FAO), dont la sortie est le code lisible par une machine utilisée pour une machine utilisée pour la commande numérique de la machine.

Depuis la mise en œuvre du Gcode , il est nécessaire de tester de différents choix pour un interprète open source Gcode pour l'arduino, de sorte que les mouvements corrects seront obtenus par les axes de la machine par l'intermédiaire du pilote de moteur

I.9. Avantages des machines CNC

- Amélioration de la précision dimensionnelle.
- Amélioration de l'état de surface.
- Possibilité de produire un grand nombre de pièces identiques en un temps réduit.
- Usinage de forme quelconque.
- Changement de production plus facile.

I.10. Inconvénients des machines CNC

- Prix de la machine.
- Coût horaire plus élevé.
- Moins de tolérance aux erreurs.
- Environnement plus contrôlé à cause des systèmes électroniques et de la précision supérieure

CHAPITRE II

CONCEPTION DE LA MACHINE CNC

II.1 Architecture de la graveuse laser

Pour la réalisation de notre mini machine CNC, nous avons opté pour l'utilisation des éléments suivants :

- Deux moteurs pas à pas (extraits) de deux lecteurs DVD.
- Une carte Arduino pour la commande.

II.2 Caractéristiques de la graveuse laser :

Une graveuse laser possède plusieurs caractéristiques. Et ce pour objectif de graver (graver un logo) ou découper des formes (sur du bois, de l'acier...).

Pour aboutir à cela il nous faut du matériel et un logiciel adapté.

II.2.1 Partie logicielle

II.2.1.1 Arduino

II.2.1.1.1 Présentation

Le logiciel de programmation des modules Arduino est une application Java, libre et multiplateformes, serve d'éditeur de code et de compilateur, et qui peut transférer le firmware et le programme à travers la liaison série (RS232, Bluetooth ou USB selon le module). Il est également possible de se passer de l'interface Arduino, et de compiler les programmes en ligne de commande. Le langage de programmation utilisé est le C++. La mise en place de ce langage standard rend aisé le développement de programmes sur les plates-formes Arduino, à toute personne qui maîtrise le C ou le C++. Afin de pouvoir programmer la carte Arduino, il faudra le logiciel Arduino pour charger le code en question. Le logiciel Arduino a pour fonctions principales :

- Pouvoir écrire et compiler des programmes pour la carte Arduino.
- Se connecter avec la carte Arduino pour y transférer les programmes.
- Communiquer avec la carte Arduino.

Le logiciel Arduino intègre également un TERMINAL SERIE (fenêtre séparée) qui permet d'afficher des messages textes reçus de la carte Arduino et d'envoyer des caractères vers la carte Arduino. Cette fonctionnalité permet une mise au point facile des programmes, permet d'afficher sur l'ordinateur l'état des variables, des résultats de calculs ou de conversions

analogique-numérique : un élément essentiel pour améliorer, tester et corriger ses programmes.

Comme la montre la Figure II.1, l'environnement de développement comporte quatre

Parties :

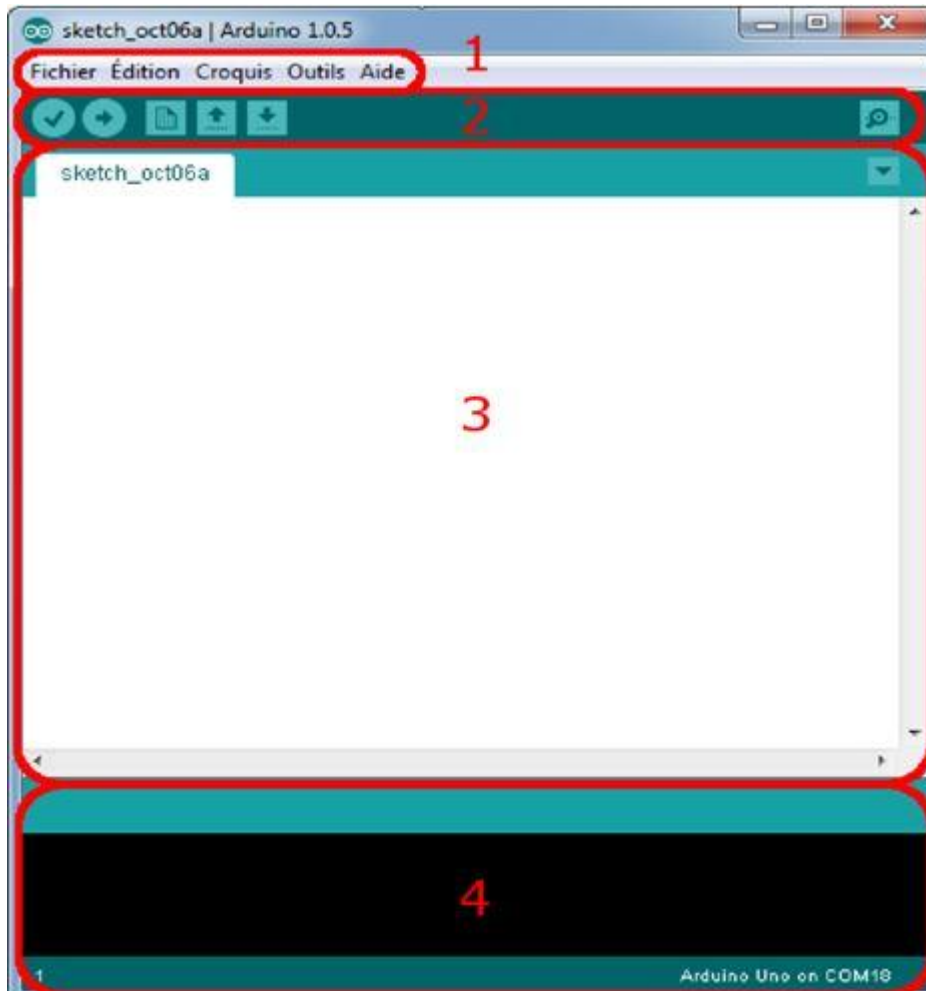


Figure II.1 : différentes parties d'une interface Arduino

Partie 1 : Représente les options de configuration du logiciel.

Partie 2 : Barre des boutons, leur rôle est illustré dans le Tableau.

Partie 3 : Editeur de texte permettant d'écrire le programme source en langage évolué.

Partie 4 : Console de texte qui affiche les messages concernant les résultats de la compilation du programme.







Bouton	Fonction
	Vérifier : compile le programme en cours d'édition et, de ce fait, vérifier sa syntaxe
	Téléverser : compile le programme en cours d'édition et le télécharger en mémoire de l'Arduino
	Nouveau : ouvrir une nouvelle fenêtre vierge pour écrire un nouveau programme.
	Ouvrir : ouvre la liste de tous les sketches contenus dans skethcbook.
	Enregistrer : sauvegarde le sketch en cours d'édition.
	Moniteur série : ouvre l'écran du moniteur de port série intégré à l'environnement.

Tableau II.1 : Fonctions des icônes de la barre des boutons de l'IDE

II.2.1.1.2 Langage de programmation

Un programme, ou sketch, destiné à Arduino, comporte toujours deux fonctions distinctes. La première est appelée `setup`, n'est exécutée qu'une seule fois après le lancement du programme. Elle contient généralement des instructions de certaines ressources de la carte telle que, entrée/sortie (output/input), définition de la vitesse de fonctionnement du port série...etc. Elle s'écrit de la façon suivante:

```
void setup() {
```

```
  Instruction à n'exécuter qu'une seule fois ;
```

```
}
```

La seconde fonction s'appelle `loop` (boucle en anglais), elle contient les instructions qui seront indéfiniment répétées tant que l'Arduino restera sous tension. En d'autres termes, suite à un reset au moyen de bouton poussoir ou suite à une mise sous tension qui a pour effet de provoquer un reset automatique, l'Arduino exécute une seule fois les instructions contenues dans la fonction `setup` puis exécute ensuite indéfiniment les

instructions contenues dans la fonction loop de ce même programme. La fonction loop s'écrit de la fonction suivante :

```
voidloop() {
```

Instructions qui seront répétées indéfiniment ;

```
}
```

Selon le cas, une troisième partie peut être présente mais ne contient pas des instructions exécutables. Il s'agit de la zone de définition des constantes au moyen de l'instruction `define` ou de l'instruction `const`, ou bien encore d'inclusion de bibliothèques utilisées par le programme au moyen de `include`. Elle se place avant la fonction `setup`. La structure générale d'un programme destiné à Arduino est la suivante :

Définition éventuelle des constantes et bibliothèques :

```
voidsetup() {
```

Instructions à n'exécutent qu'une seul fois ;

```
}
```

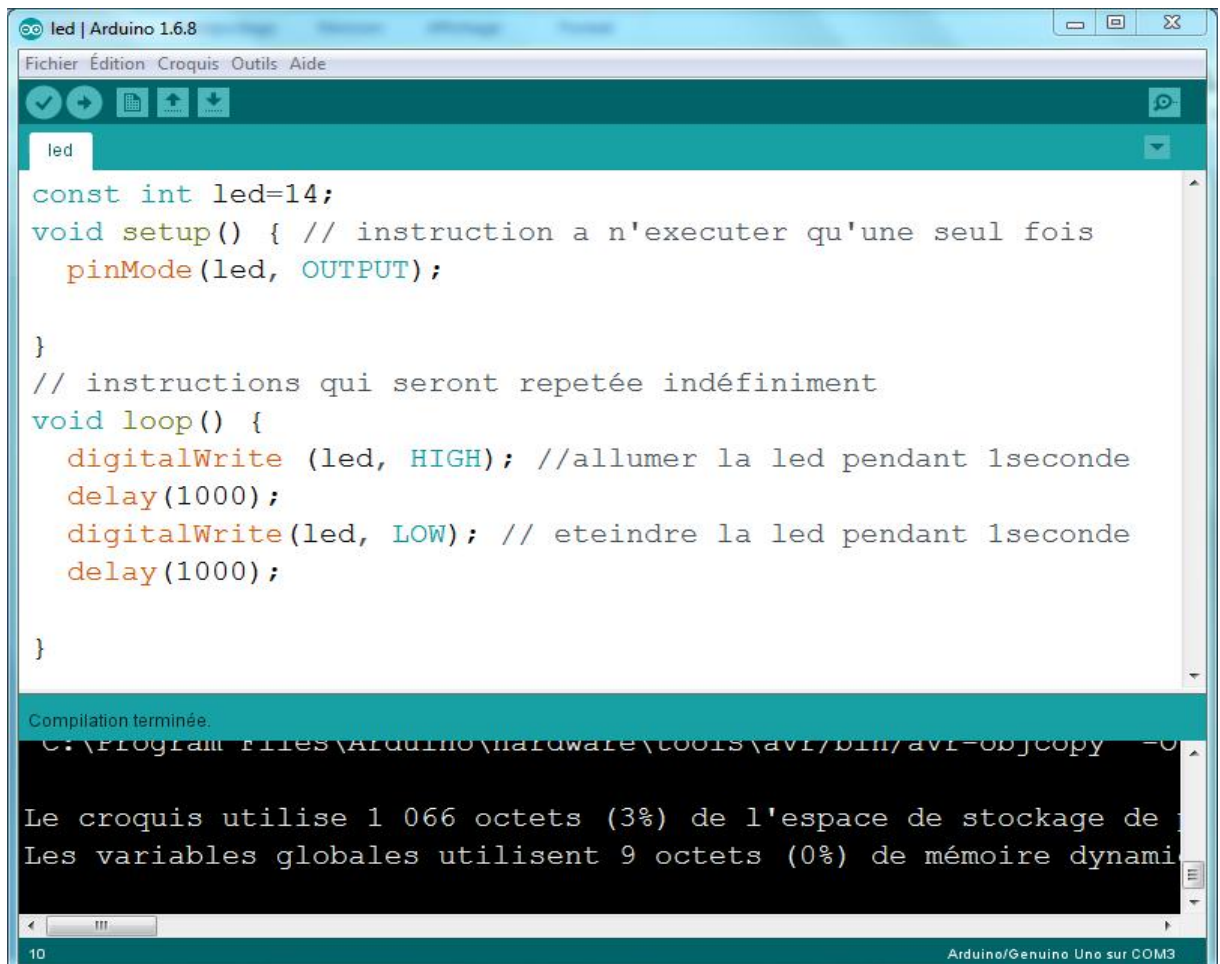
```
voidloop() {
```

Instructions qui seront répétées indéfiniment ;

```
}
```

II.2.1.1.3 Exemple de programme sur Arduino

La Figure (II.2) illustre un programme Arduino qui permet de faire clignoter une LED, avec la structure décrite précédemment.



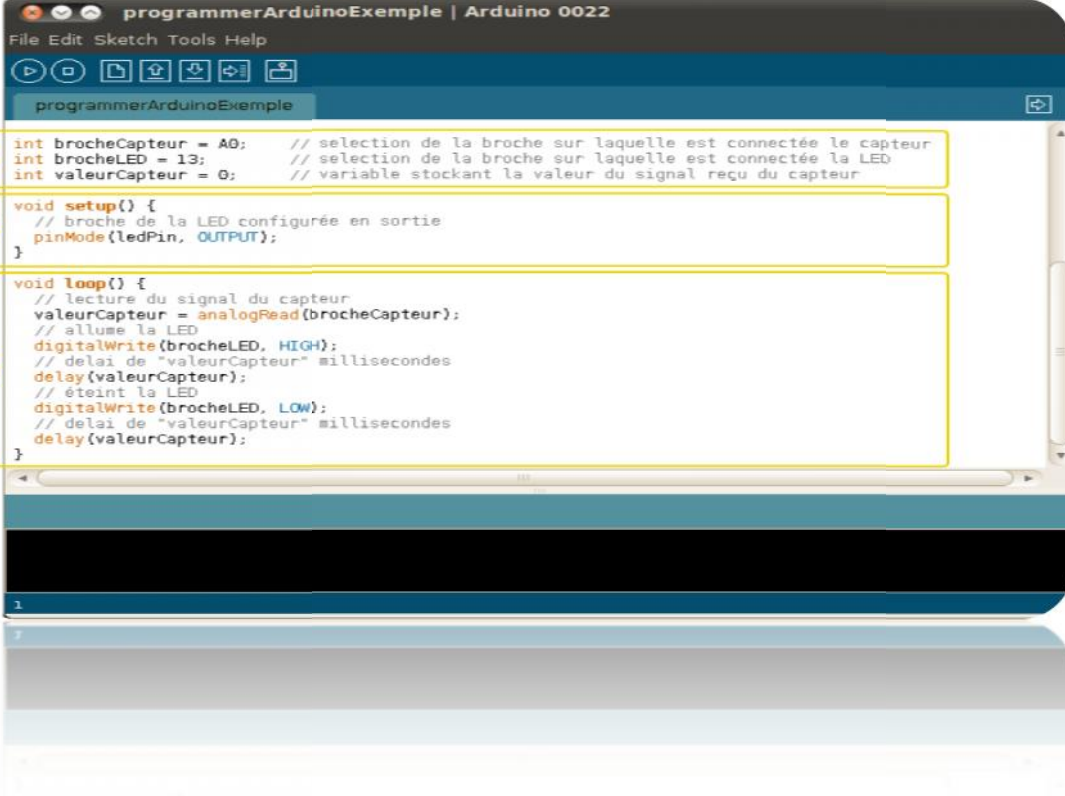
```
led | Arduino 1.6.8
Fichier Édition Croquis Outils Aide
led
const int led=14;
void setup() { // instruction a n'executer qu'une seul fois
  pinMode(led, OUTPUT);
}
// instructions qui seront repetée indéfiniment
void loop() {
  digitalWrite (led, HIGH); //allumer la led pendant 1seconde
  delay(1000);
  digitalWrite(led, LOW); // eteindre la led pendant 1seconde
  delay(1000);
}
Compilation terminée.
C:\Program Files\Arduino\hardware\tools\avr\bin/avr-objcopy -O
Le croquis utilise 1 066 octets (3%) de l'espace de stockage de
Les variables globales utilisent 9 octets (0%) de mémoire dynami
10 Arduino/Genuino Uno sur COM3
```

Figure II.2 : Programme pour faire clignoter une LED.

II.2.1.1.4 La structure d'un programme

Un programme Arduino comporte trois parties :

1. la partie déclaration des variables (optionnelle)
2. la partie initialisation et configuration des entrées/sorties : la fonction **setup ()**
3. la partie principale qui s'exécute en boucle : la fonction **loop ()**



```
programmerArduinoExemple | Arduino 0022
File Edit Sketch Tools Help
programmerArduinoExemple
1 int brocheCapteur = A0; // selection de la broche sur laquelle est connectée le capteur
  int brocheLED = 13; // selection de la broche sur laquelle est connectée la LED
  int valeurCapteur = 0; // variable stockant la valeur du signal reçu du capteur
2 void setup() {
  // broche de la LED configurée en sortie
  pinMode(ledPin, OUTPUT);
}
3 void loop() {
  // lecture du signal du capteur
  valeurCapteur = analogRead(brocheCapteur);
  // allume la LED
  digitalWrite(brocheLED, HIGH);
  // delai de "valeurCapteur" millisecondes
  delay(valeurCapteur);
  // éteint la LED
  digitalWrite(brocheLED, LOW);
  // delai de "valeurCapteur" millisecondes
  delay(valeurCapteur);
}
```

Figure II.3: structure d'un programme

Dans chaque partie d'un programme sont utilisées différentes instructions issues de la syntaxe du langage Arduino.

II.2.1.1.5 Coloration syntaxique

Lorsque le code est écrit dans l'interface de programmation, certains mots apparaissent en différentes couleurs qui clarifient le statut des différents éléments :

- En **orange**, apparaissent les mots-clés reconnus par le langage Arduino comme des **fonctions** existantes. Lorsqu'on sélectionne un mot coloré en orange et qu'on effectue un clic avec le bouton droit de la souris, on a la possibilité de choisir « Find in référence » : cette commande ouvre directement la documentation de la fonction sélectionnée.
- En **bleu**, apparaissent les mots-clés reconnus par le langage Arduino comme des **constantes**.

- En **gris**, apparaissent les **commentaires** qui ne seront **pas exécutés dans le programme**. Il est utile de bien commenter son code pour s'y retrouver facilement ou pour le transmettre à d'autres personnes. On peut déclarer un commentaire de deux manières différentes :

dans une ligne de code, tout ce qui se trouve après « // » sera un commentaire.

On peut encadrer des commentaires sur plusieurs lignes entre « /* » et « */ ».

II.2.1.1.6 Ponctuation

Le code est structuré par une ponctuation stricte :

- **toute ligne** de code se termine par un point-virgule « ; »
- le contenu d'une **fonction** est délimité par des accolades « { » et « } »
- les **paramètres** d'une fonction sont contenus pas des parenthèses « (» et «) ».

II.3 G-code

II.3.1 Présentation

Le G-Code est un langage de programmation pour contrôler une machine à Commande Numérique. Il sert donc à programmer les mouvements que la machine va effectuer, Il s'agit de simple fichier texte au même titre le code en C, Pascal...

Il se compose d'un certain nombre de "commandes" spécifiques, indiquant à la machine quel type de mouvement elle doit exécuter (droite, arc de cercle, etc.), et d'indications de coordonnées sur les axes X, Y et Z.

Ce langage 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.

II.3.2 L'éditeur de 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 ArduinoUno ou à simuler sur l'ordinateur.

II.3.3 Définition de l'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.

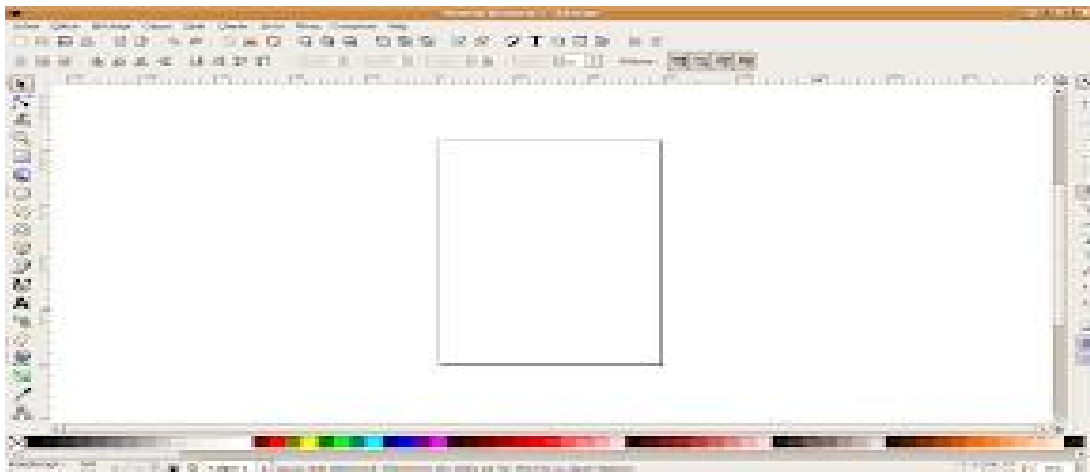


Figure II.4 L'interface principale de Inkscape

II.3.4 Logiciel de pilotage de la machine CNC

La solution la plus fonctionnelle qu'on a trouvée est une évolution de UniversalGcode Sender (développé initialement par l'auteur de GRBL), qui est cours de développement

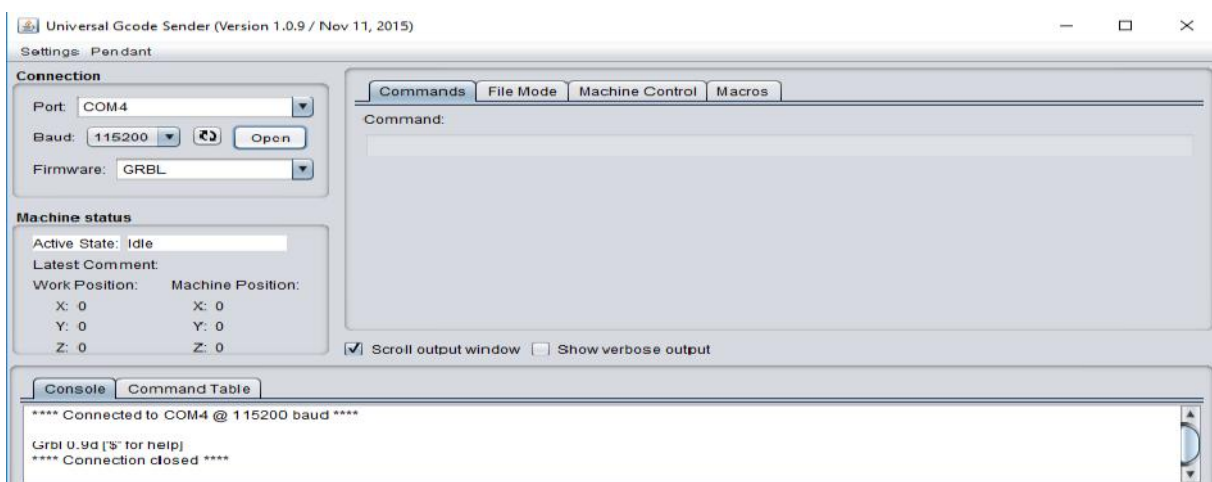


Figure II.5: UniversalGcode Sender.

II.4 GRBL

II.4.1 Présentation

GRBL est un logiciel pour contrôler le mouvement de machines qui font des gravures. La plupart des imprimantes 3D Open-Source sont basées sur GRBL. Il a été adapté pour être utilisé dans des centaines de projets, comprenant des machines à découpe laser, des écritoires manuel automatisés, perceuses, peintre de graffiti et machines à dessins bizarroïdes... En raison de ses performances, de sa simplicité et de sa frugalité en besoins matériels, GRBL a grossi en un vrai petit phénomène Open Source.

II.4.2 Fonctionnalités

GRBL est destiné pour la production non intensive. Nous l'utilisons pour tous nos fraisages, depuis Laptop où PCs, utilisant d'excellentes interfaces utilisateur. Il est écrit en C optimisé pour utiliser les fonctions intelligentes des puces Atmega328, pour obtenir un timing précis et des fonctions asynchrones. Il est capable de maintenir un taux de pas supérieur à 30kHz, et délivre un courant propre de pulsations de contrôle.

GRBL est pour l'usinage 3 axes. Pas d'axe de rotation Juste X, Y et Z.

L'interpréteur **G-Code** implémente un sous-ensemble du standard NIST rs274/ngc et est testé au travers d'un grand nombre d'outils sans problèmes. Mouvements linéaires, circulaires et hélicoïdaux sont supportés sans problèmes.

Voici un petit tableau récapitulatif des codes de base :

G00	Déplacement rapide
G01	Interpolation linéaire
G02	Interpolation circulaire (sens horaire, anti-trigo)
G03	Interpolation circulaire (sens anti-horaire, trigo)
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 (suivi de l'argument L suivant le type de données à écrire)
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 (mode <i>Interrupt</i> utilisé pour les capteurs et les mesures pièces et 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
G54 à G59	Activation du décalage d'origine pièce (<i>Offset</i>)
G68 / G68.1	Activation du mode "Plan incliné" (<i>Tilted plane working</i>) pour les centres d'usinage 5 axes
G71 / G71.7	Cycle d'ébauche suivant l'axe Z (appel de profil balisé entre les arguments P et Q)
G76 / G76.7	Cycle de filetage
G69	Annulation du mode <i>Tilted plane working</i> (Plan incliné)
G84	Cycle de taraudage rigide
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

Tableau II.2 : codes de bases du G-code.

La plupart des options de configuration peuvent être réglées en fonctionnement et sauvées en mémoire (EEPROM) entre les sessions et même conservées entre différentes versions de GRB lorsque vous mettez à jour le firmware.

En général, le G-code est généré à partir de fichiers 3D ou 2D, à l'aide des programmes générateurs de G-code.

II.2.2 Partie matériels

II.2.2.1 présentation de la carte Arduino Uno

La carte Arduino Uno, présentée est une carte à microcontrôleur basée sur un Atmega328p. 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). 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. 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.

La carte Arduino Uno dont le prix est très abordable convient bien pour le récepteur.



Figure II.6 : carte Arduino Uno

II.2.2 Description matérielle

Généralement tout module électronique qui possède une interface de programmation est basé toujours dans sa construction sur un circuit programmable ou plus.

II.2.2.1 Le Microcontrôleur ATmega328

Un microcontrôleur ATmega328 est un circuit intégré qui rassemble sur une puce plusieurs éléments complexes dans un espace réduit au temps des pionniers de l'électronique. Aujourd'hui, en soudant un grand nombre de composants encombrants ; tels que les transistors; les résistances et les condensateurs tout peut être logé dans un petit boîtier en plastique noir muni d'un certain nombre de broches dont la programmation peut être réalisée en langage C. la figure II.7 montre un microcontrôleur ATmega 328, qu'on trouve sur la carte Arduino.

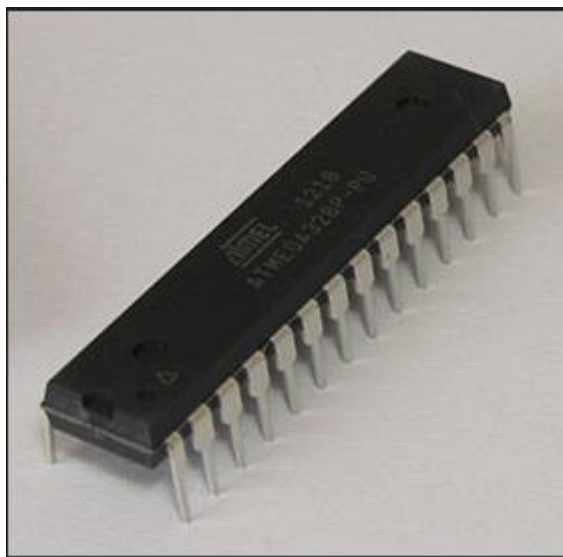


Figure II.7 : ATmega 328.

Le microcontrôleur ATmega328 est constitué par un ensemble d'éléments qui ont chacun une fonction bien déterminée. Il est en fait constitué des mêmes éléments que sur la carte mère d'un ordinateur. Globalement, l'architecture interne de ce circuit programmable se compose essentiellement sur :

- **La mémoire Flash:** C'est celle qui contiendra le programme à exécuter. Cette mémoire est effaçable et réinscriptible mémoire programme de 32Ko (dont bootloader de 0.5 ko).

- **RAM** : c'est la mémoire dite "vive", elle va contenir les variables du programme. Elle est dite "volatile" car elle s'efface si on coupe l'alimentation du microcontrôleur. Sa capacité est 2 ko.
- **EEPROM** : C'est le disque dur du microcontrôleur. On y enregistre des infos qui ont besoin de survivre dans le temps, même si la carte doit être arrêtée. Cette mémoire ne s'efface pas lorsque l'on éteint le microcontrôleur ou lorsqu'on le reprogramme.

II.2.2.2. Les sources de l'alimentation de la carte

On peut distinguer deux genres de sources d'alimentation (Entrée Sortie) et cela comme suit :

- **VIN**. La tension d'entrée positive lorsque la carte Arduino est utilisée avec une source de tension externe (à distinguer du 5V de la connexion USB ou autre source 5V régulée). On peut alimenter la carte à l'aide de cette broche, ou, si l'alimentation est fournie par le jack d'alimentation, accéder à la tension d'alimentation sur cette broche
- **5V**. La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (pour info : les circuits électroniques numériques nécessitent une tension d'alimentation parfaitement stable dite "tension régulée" obtenue à l'aide d'un composant appelé un régulateur et qui est intégré à la carte Arduino). Le 5V régulé fourni par cette broche peut donc provenir soit de la tension d'alimentation VIN via le régulateur de la carte, ou bien de la connexion USB (qui fournit du 5V régulé) ou de tout autre source d'alimentation régulée.
- **3V3**. Une alimentation de 3.3V fournie par le circuit intégré FTDI (circuit intégré faisant l'adaptation du signal entre le port USB de votre ordinateur et le port série de l'ATmega) de la carte est disponible : ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5V. L'intensité maximale disponible sur cette broche est de 50mA

II.2.2.2.3 Les entrées & sorties

Cette carte possède 14 broches numériques (numérotée de 0 à 13) peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions `pinMode()`, `digitalWrite()` et `digitalRead()` du langage Arduino. Ces broches fonctionnent en 5V. Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité et dispose d'une résistance interne de "rappel au plus" (pull-up) (déconnectée par défaut) de 20-50 KOhms. Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction `digitalWrite(broche, HIGH)`.

En plus, certaines broches ont des fonctions spécialisées

- **Interruptions Externes:** Broches 2 et 3. Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur. -Impulsion PWM (largeur d'impulsion modulée):

Broches 3, 5, 6, 9, 10, et 11. Fournissent une impulsion PWM 8-bits à l'aide de l'instruction `analogWrite()`

- **SPI (Interface Série Périphérique):** Broches 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ces broches supportent la communication SPI (Interface Série Périphérique) disponible avec la librairie pour communication SPI. Les broches SPI sont également connectées sur le connecteur ICSP qui est mécaniquement compatible avec les cartes
- **I2C:** Broches 4 (SDA) et 5 (SCL). Supportent les communications de protocole I2C Mega. (ou interface TWI (TwoWire Interface - Interface "2 fils"), disponible en utilisant la librairie `Wire/I2C` (ou `TWI - Two-Wire interface - interface "2 fils"`).
- **LED:** Broche 13. Il y a une LED incluse dans la carte connectée à la broche 13. Lorsque la broche est au niveau HAUT, la LED est allumée, lorsque la broche est au niveau BAS, la LED est éteinte.

La carte UNO dispose 6 entrées analogiques (numérotées de 0 à 5), chacune pouvant fournir une mesure d'une résolution de 10 bits (c.à.d. sur 1024 niveaux soit de 0 à 1023) à l'aide de la très utile fonction `analogRead()` du langage Arduino. Par défaut, ces broches

mesurent entre le 0V (valeur 0) et le 5V (valeur 1023), mais il est possible de modifier la référence supérieure de la plage de mesure en utilisant la broche AREF et l'instruction `analogReference()` du langage Arduino.

La carte Arduino UNO intègre un fusible qui protège le port USB de l'ordinateur contre les surcharges en intensité (le port USB est généralement limité à 500mA en intensité). Bien que la plupart des ordinateurs aient leur propre protection interne, le fusible de la carte fournit une couche supplémentaire de protection. Si plus de 500mA sont appliqués au port USB, le fusible de la carte coupera automatiquement la connexion jusqu'à ce que le court-circuit ou la surcharge soit stoppé

II.2.2.2.4 Les ports de communications

La carte Arduino UNO a de nombreuses possibilités de communications avec l'extérieur. L'Atmega328 possède une communication série TTL (5V), grâce aux broches numériques 0 (RX) et 1 (TX).

On utilise (RX) pour recevoir et (TX) transmettre (les données séries de niveau TTL). Ces broches sont connectées aux broches correspondantes du circuit intégré ATmega328 programmé en convertisseur USB – vers – série de la carte, composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur. Comme un port de communication virtuel pour le logiciel sur l'ordinateur, La connexion série de l'Arduino est très pratique pour communiquer avec un PC, mais son inconvénient est le câble USB, pour éviter cela, il existe différentes méthodes pour utiliser ce dernier sans fil

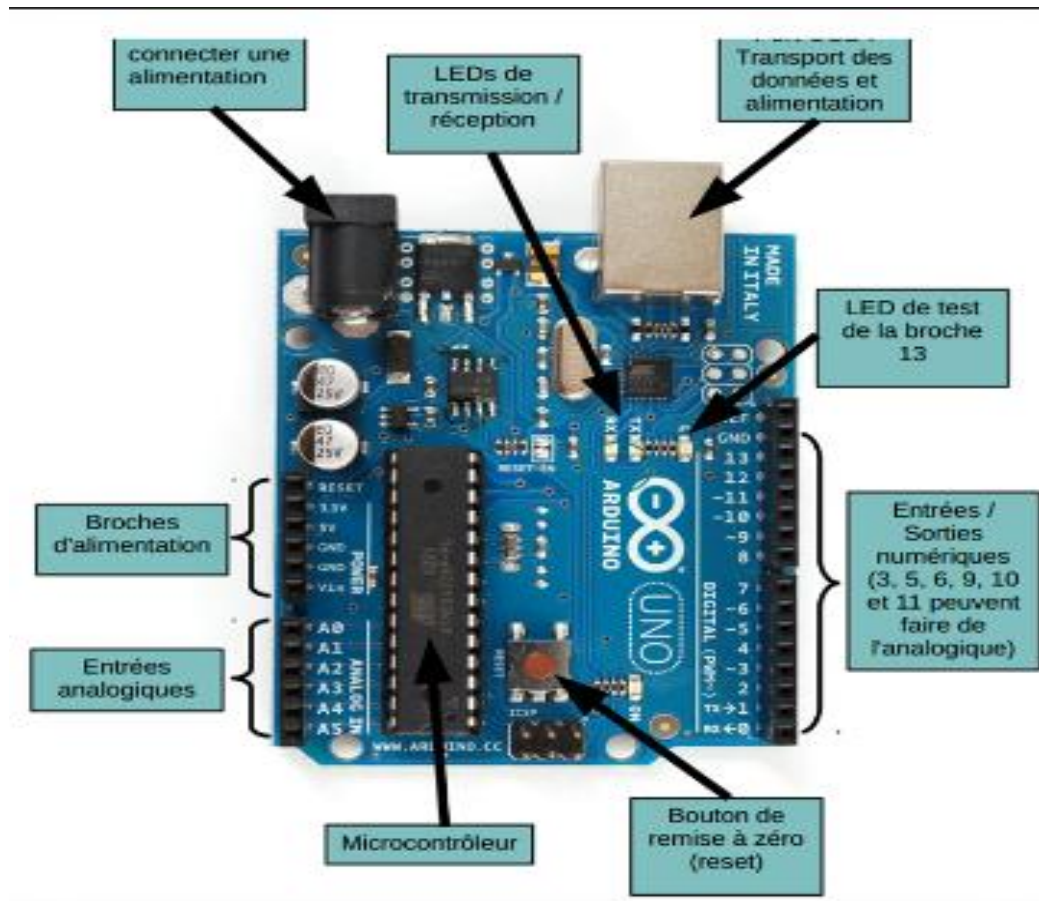


Figure II.8 : image illustrant la carte Arduino Uno avec ces différents blocs.

II.5 Les avantages et les inconvénients de la carte :

II.5.1 Avantages:

- Nombre suffisant de broches d'entrées-sorties pour les projets élémentaires.
- Vaste choix de *shields*.
- Bon marché.

II.5.2 Inconvénients:

- Nombre insuffisant de broches d'entrées-sorties pour les projet ambitieux.
- La mémoire disponible risque d'être un peu juste pour les gros projets.
- Ne peut pas être utilisée comme hôte USB pour simuler un clavier ou une souris, par exemple.

II.6 moteur pas a pas

II.6.1 Généralités

Les moteurs pas à pas sont des moteurs spéciaux, composés simplement d'un stator réunissent des pièces polaires et des bobinages, et utilisés pour commander avec grande précision le déplacement et la position d'un objet.

Comme leur nom l'indique, ces moteurs tournent par incrément discret. Chaque incrément de rotation est provoqué par une impulsion de courant fournie à l'un des enroulements du stator. Le moteur pas à pas est l'organe de positionnement et de vitesse travaillant généralement en boucle ouverte.

Le principe de base est donc toujours la création d'un champ tournant comme dans les moteurs triphasés industriels ou dans les petits moteurs équipant les programmeurs mécaniques :

II.6.2 Les différents types de moteur 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.

II.6.2.1 Les moteurs à réluctance variable

Les moteurs à réluctance variable fonctionnent selon un principe différent de celui des moteurs à aimant permanent. Ils possèdent bien évidemment un stator, mais ce dernier est fabriqué en acier doux non magnétique. Il n'est pas lisse et possède plusieurs dents. Ce type de moteur est représenté en (figure II.9). On peut voir, dans cet exemple, que le stator est composé de 8 plots sur lesquels enroulés les bobinages, ce qui donne 4 phases. Le rotor, quant à lui, ne comporte que 6 dents.

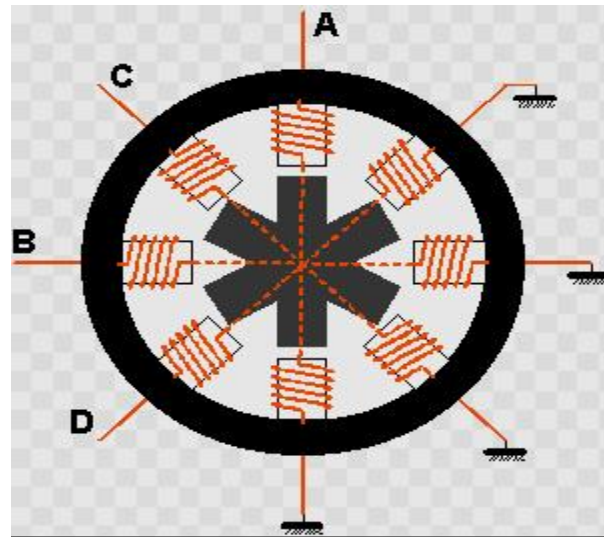


Figure II.9 : moteur à réluctance variable.

II.6.2.2 Moteur à aimants permanents

Les moteurs à aimants permanents sont semblables aux moteurs à réluctance variable, sauf que le rotor possède des pôles NORD et SUD. À cause des aimants permanents, le rotor reste freiné à sa dernière position lorsque le bloc d'alimentation cesse de fournir des impulsions. Une façon simple de voir le système, est de placer une boussole entre deux aimants. Suivant la bobine qui est alimentée et le sens du courant, l'aimant va s'aligner avec le champ.

II.6.2.2.1 Moteur à aimant permanent bipolaire

Le courant de commande est bidirectionnel et l'avance d'un pas s'effectue par une séquence de commutation des enroulements statoriques.

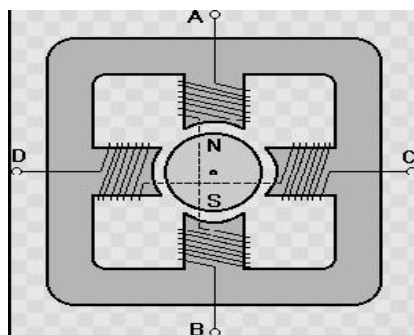


Figure II.10 : moteur bipolaire.

Pour ce type de moteur nous avons trois possibilités de commande :

- Fonctionnement à demi-pas.
- Fonctionnement avec couple maximal.
- Fonctionnement à pas complet.

II.6.2.2. Moteur à aimant permanent unipolaire

Les moteurs unipolaires se différencient par le fait qu'ils sont à double bobinage. Le double bobinage est utilisé pour l'inversion du flux statorique et le moteur se commande de la même manière qu'un bipolaire excepté qu'un seul transistor pour chaque enroulement suffit dans l'étage de puissance.

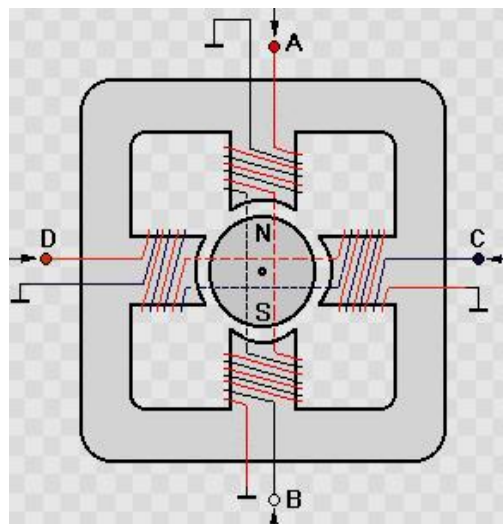


Figure II.11 : moteur unipolaire

Simplement, les moteurs unipolaires sont plus chers car leur fabrication réclame un double bobinage. De plus, pour une taille donnée, ce type de moteur a un couple plus faible à cause des enroulements qui sont plus fins. Il fut une époque où les moteurs unipolaires étaient intéressants pour les concepteurs parce qu'ils simplifiaient l'étage de commande électronique. Maintenant, grâce aux circuits de commande (push pull monolithique) du genre L293D, les moteurs bipolaires sont devenus populaires et d'une utilisation courante.

II.6.2.3 Moteur pas-à-pas hybride MH

En combinant les structures des deux moteurs précédents, c'est à dire en plaçant les aimants du moteur à aimants permanents dans un circuit ferromagnétique on crée un nouveau type de moteur appelé moteur réluctant polarisé ou moteur hybride (Hybride Motors : HB). Dans ce cas, il existe un couple réluctant provoqué par la variation de perméances propres associées à chaque aimant et à chaque bobine.

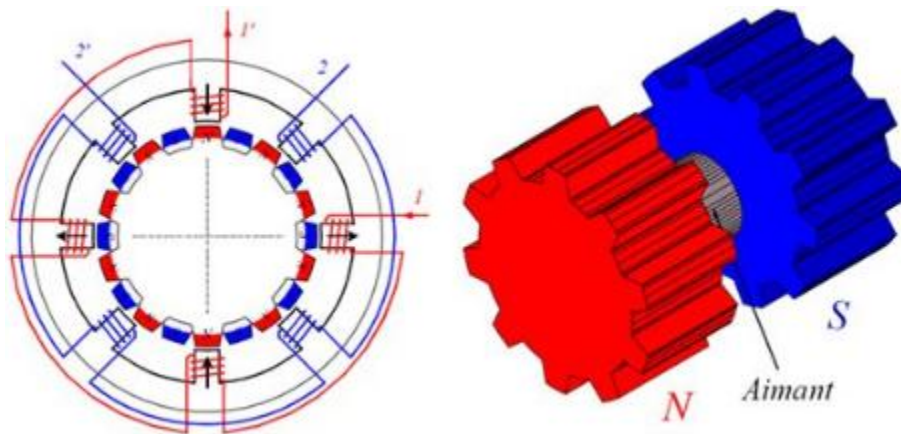


Figure II.12 : moteur hybride (MH)

II.6.2.3.1 Technologie des moteurs pas à pas hybrides

Les moteurs pas à pas hybrides réunissent, au moins en partie, les avantages des moteurs pas à pas à réluctance 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.

II.6.3 Comparaison entre les différents types des moteurs pas à pas

Cette comparaison peut se résumer selon le tableau suivant :

Type de moteur pas à pas	Aimant permanent	Hybride	Reluctance variable
Résolution (nb de pas par tour)	Moyenne	Elevée	Bonne
Couple moteur	Elevée	Elevée	Faible
Sens de rotation	Dépend : -de l'ordre d'alimentation des phases -du sens du courant dans les bobines	Dépend : -De l'ordre d'alimentation des phases -du sens du courant dans les bobines	Dépend : - de l'ordre d'alimentation des phases
Fréquence de travail	Faible	Grande	Grande
Puissance	Quelques dizaines de watts	Quelques watts	Quelques watts
Inconvénients			Pas de mémoire de position

Tableau II.3 : Comparaisant entre les différents types des moteurs pas à pas

II.7 Driver L293D

II.7.1 Présentation

Le composant L293D est un pont de puissance composé de plusieurs transistors et relais qui permet d'activer la rotation d'un moteur. Ce composant ne coûte pas très cher. Le L293D est un double pont-H, ce qui signifie qu'il est possible de l'utiliser pour commander quatre moteurs distincts (dans un seul sens) grâce à ses 4 canaux. Il est également possible de constituer deux pont-H afin de piloter deux moteurs distincts, dans

les deux sens et indépendamment l'un de l'autre. Il est important de noter que le L293D peut délivrer au maximum 600mA, veuillez donc choisir vos moteurs en conséquence.

II.7.2 Brochage

Ci-dessous la configuration des broches du L293D et la table de la logique de commande.

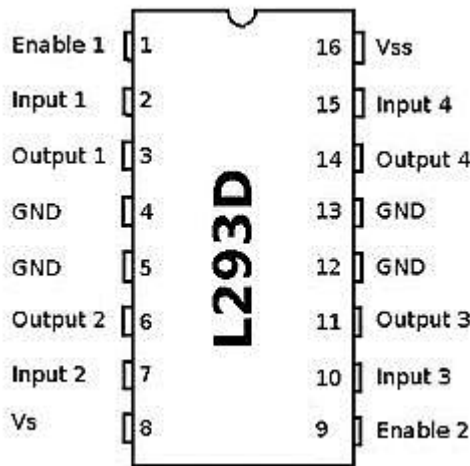


Figure II.13 : Différentes broches du L293D

Broche	Nom	Description
1	Enable 1	<p>permet d'envoyer (ou pas) la tension sur les sorties du moteur via OUTPUT1 & OUTPUT2.</p> <p>ENABLE1 commande l'activation/désactivation du premier Pont-H.</p> <ul style="list-style-type: none">) Si ENABLE1 = GND, le pont-H est déconnecté et le moteur ne fonctionne pas.) Si ENABLE1 = VSS, le pont-H est connecté aux sorties et le moteur fonctionne dans un sens ou l'autre ou pas en fonction des tensions appliquée sur INPUT1 & INPUT2.
2	Input 1	<p>Avec Input 2, sont les broches de commande du Pont-H Output1/Output2.</p> <p>Se raccorde a Arduino, permet de commander le sens du courant entre Output 1 et Output 2.</p>
3	Output 1	Avec Output 2, seront les broches à raccorder à la charge (le moteur).

4	GND	Doit être raccorder à la masse (GND) de la source d'alimentation de puissance VS (ex: la borne négative de l'accumulateur +9.2v) et à la masse de la source d'alimentation de la logique "VSS" (donc GND Arduino). Si vous n'avez qu'une source d'alimentation pour le tout, c'est forcément plus simple.
5	GND	
6	Output 2	Avec Output 1, seront les broches à raccorder à la charge (le moteur).
7	Input 2	Avec Input 1, sont les broches de commande du Pont-H Output1/Output2. Se raccorde a Arduino, permet de commander le sens du courant entre Output 1 et Output 2.
8	VS	Alimentation de puissance des moteurs. Par exemple, s'il s'agit d'une ancien véhicule téléguidé transformé, il s'agira de la borne positive de l'accumulateur (souvent +9.2v).
9	Enable 2	Commande l'activation du second pont-H constitué de Output3/Output4
10	Input 3	A utiliser conjointement avec Input 4 pour commander le pont-H Output3/Output4.
11	Output 3	Constitue une des deux sorties du second pont-H (Output3/Output4)
12	GND	
13	GND	
14	Output 4	Constitue une des deux sorties du second pont-H (Output3/Output4)
15	Input 4	A utiliser conjointement avec Input 3 pour commander le pont-H Output3/Output4.
16	VSS	Alimentation de la logique de commande (5V). A raccorder à la borne +5V d'Arduino (donc sur le régulateur d'Arduino).

Tableau II.4 : descriptif des broches du L293D

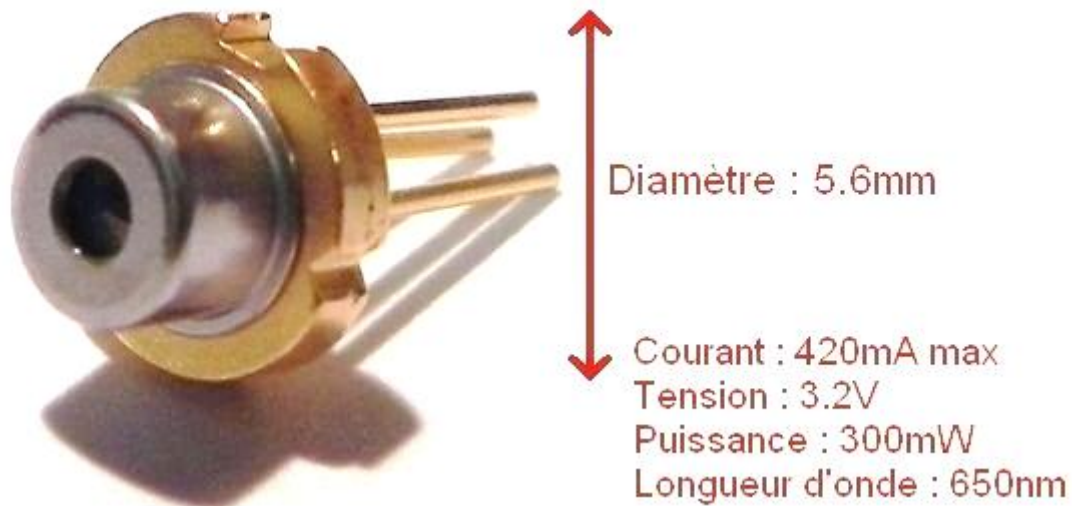
II.7.3 Caractéristiques

Caractéristique	Valeur
Nbre de pont-H	2
Courant Max Régime continu	600mA (x2)
Courant de pointeMax < 2ms	1200mA
VS Max Alim moteur	36v
VSS Max Alim logique	7v
Nbre de Broche	16 DIP
Perte de tension	1.3 à 1.4v (typical)

II.8 Diode laser

II.8.1 Présentation

Les diodes Laser reposent sur des technologies bien différentes en fonction de la couleur émise. En fonction de la couleur du Laser, le principe sera différent : diode laser simple ou diode laser de pompage et cristaux spécifiques.



II.8.2 Fonctionnement électrique

Electriquement, une diode laser s'alimente comme une LED. Comme toute diode, elle est polarisée. Le courant qui circule dans la diode laser doit être maîtrisé avec une beaucoup plus grande précision que pour une LED. En effet, une pointe de courant, même très brève, ou une décharge d'électricité statique sont fatales à une diode laser. En tension inverse, une diode laser ne supporte que 2 ou 3V.

Comme une LED, la caractéristique courant-tension fait qu'il s'établit une tension aux bornes de la diode laser qui varie assez peu avec le courant. La diode laser doit donc être contrôlée en courant.

II.8.3 Courant de seuil d'une diode laser

La lumière "laser" ne commence à être émise qu'à partir d'un certain courant (10 à 25% du courant maximum). Par exemple, il faut au moins 80mA pour commencer à voir le faisceau laser rouge pour la diode laser 300mW. Ce phénomène n'existe pas sur les LED.

CHAPITRE III

RÉALISATION DE LA MACHINE CNC

III.1 Les différentes étapes à suivre pour construire la machine CNC

Pour confectionné notre mini machine CNC, on a eu recours à différentes étapes qui sont les suivantes :

Etape 01 : Les éléments nécessaires pour le fonctionnement de notre machine CNC

- ✓ Moteurs pas à pas.
- ✓ Carte Arduino Uno.
- ✓ Driver L293D (pilotage des moteurs pas à pas).
- ✓ Les fils.
- ✓ Support en bois.
- ✓ breadboard.
- ✓ Source laser(stylo).

Etape 02 : Démonteur le lecteur DVD

Nous avons retirés tous les éléments électroniques de la tête du lecteur. Il faut obtenir une surface plane pour pouvoir coller correctement le plateau de dessin et le porte crayon. Nous avons démonté le lecteur DVD pour ne conserver que le mécanisme de déplacement de la tête du lecteur. On obtient un système de déplacement linéaire qu'on a pu récupérer sur le lecteur DVD qui représente un porteur d'un axe de la CNC machine.

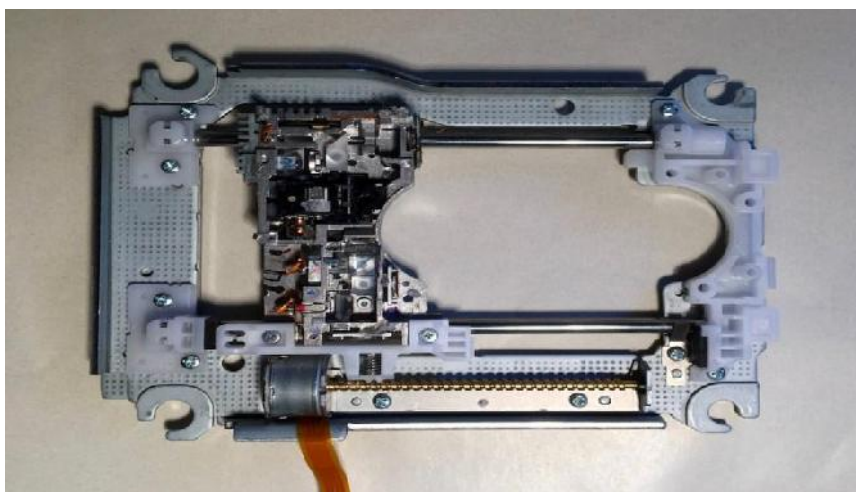


Figure III.1: Système de déplacement linéaire récupéré sur un lecteur de DVD.

Etape 3 : Soudage des pates du moteur pas à pas

Dans cette étape, nous allons réaliser les soudures sur les pates du moteur pas à pas pour avoir une bonne connexion entre les moteurs, driver, et la carte Arduino.



Figure III.2: Soudage des fils au niveau du moteur

Etape 4 : Fabrication de la structure en contreplaqué

Pour réduire le coût du projet, nous avons opté pour réaliser la structure en contreplaqué de 2cm d'épaisseur disponible dans n'importe quel magasin. La fabrication est très simple. Il n'y a que 3 plaques de contreplaqué à découper.



Figure. III.3 : Support de la machine

II.2 Branchement du L293D avec la carte Arduino et le moteur pas à pas

Composent	Pin composent	Arduino	Moteur pas à pas
L293D	Broche 1	+5V	
	Broche 8		
	Broche 9		
	Broche 16		
	Broche 4	GND	
	Broche 5		
	Broche 12		
	Broche 13		
	Broche 3		Pin 4
	Broche 6		Pin 3
	Broche 11		Pin 1
	Broche 14		Pin 2
	Broche 2	Pin 8	
	Broche 7	Pin9	
	Broche 10	Pin10	
Broche 15	Pin11		

Figure III.1 : Branchement du l293D

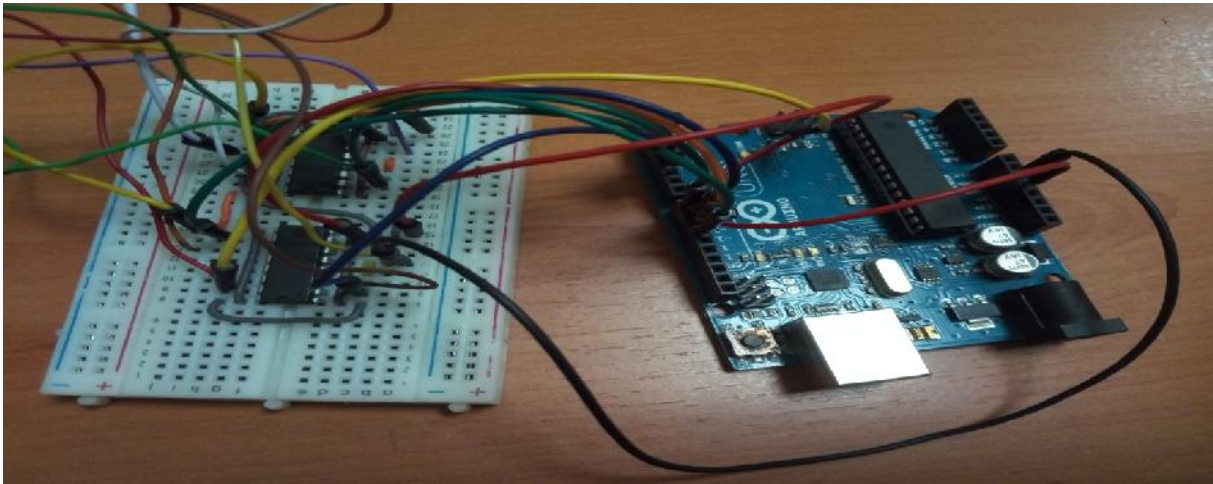


Figure III.4 : Branchement de deux drivers L293D

III.3 Pilotage de deux moteurs pas à pas via la carte Arduino

Le pilotage d'un seul moteur pas à pas ce fait avec une carte arduino et ceci en ajoutant le programme adequat, il est donc nécessaire d'appelé la bibliothèque<<stepper oneRevolution>> puis le transféré vers la carte.

Pour le pilotage de deux moteurs pas à pas on a écrit un programme Arduino avec une simulation sur isisproteus.

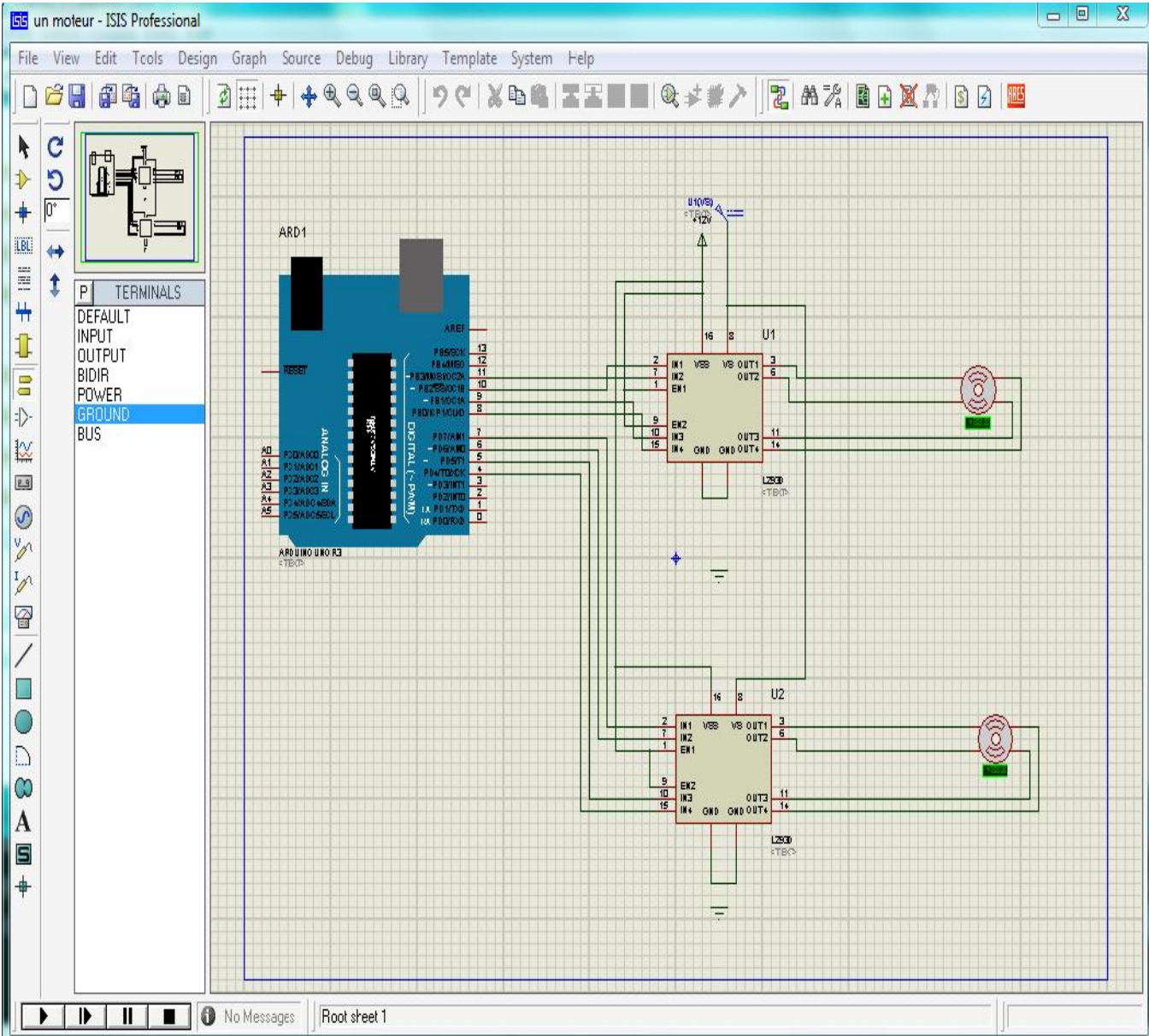


Figure III : Montage des deux moteurs sur isis proteus.

III.4 Programmation

```
#include <Stepper.h>

constintstepsPerRevolution = 200;

Stepper myStepper(stepsPerRevolution, 8, 9, 10, 11);

void setup() {

myStepper.setSpeed(60);

Serial.begin(9600);

}

void loop() {

Serial.println("clockwise");

myStepper.step(stepsPerRevolution);

delay(500);

Serial.println("counterclockwise");

myStepper.step(-stepsPerRevolution);

delay(500);

}
```

III.5. Le montage de la machine CNC

Après avoir cité les différentes étapes, nous allons assembler tous les éléments nécessaires. Pour ce faire on a fixé les deux lecteurs dvd de manière perpendiculaire, placer une petite plaque sur le lecteur du bas qui représente l'axe des X et un stylo sur l'autre dvd qui représente l'axe des Y.

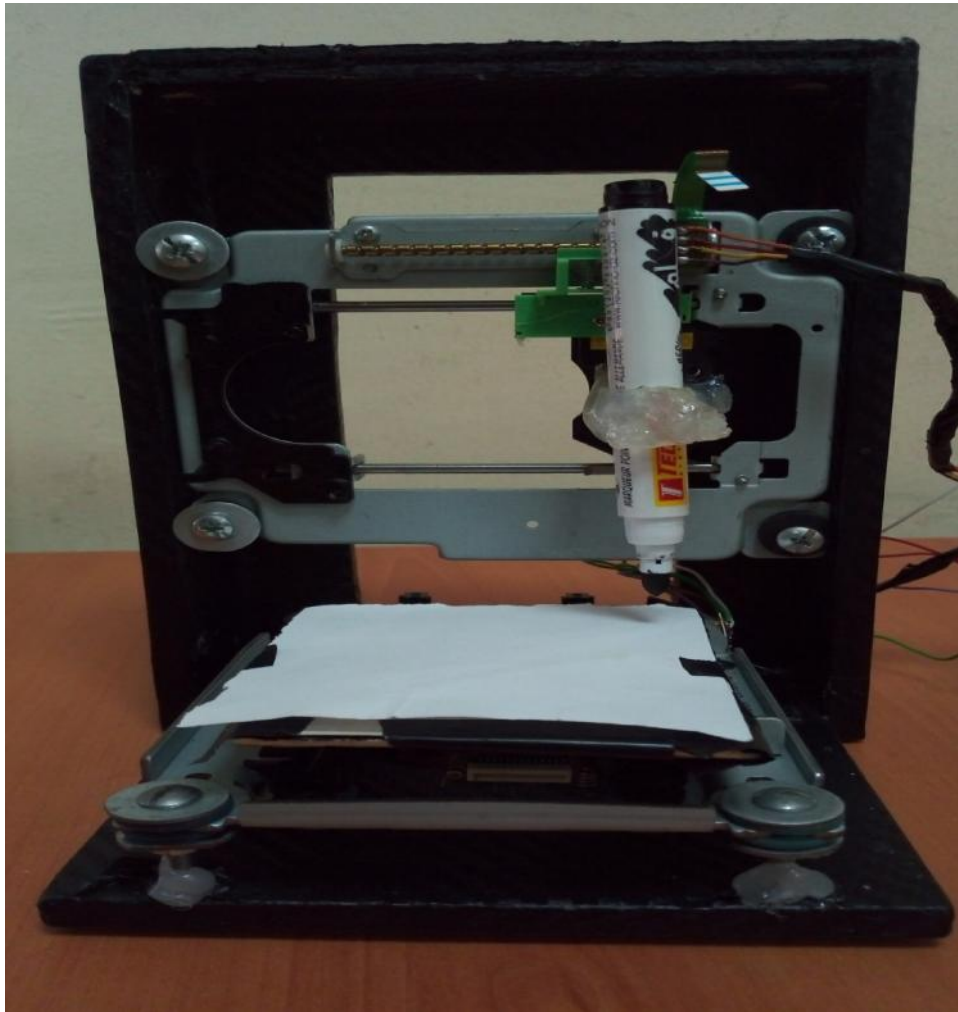


Figure III.6 : Assemblage du support avec les lecteurs

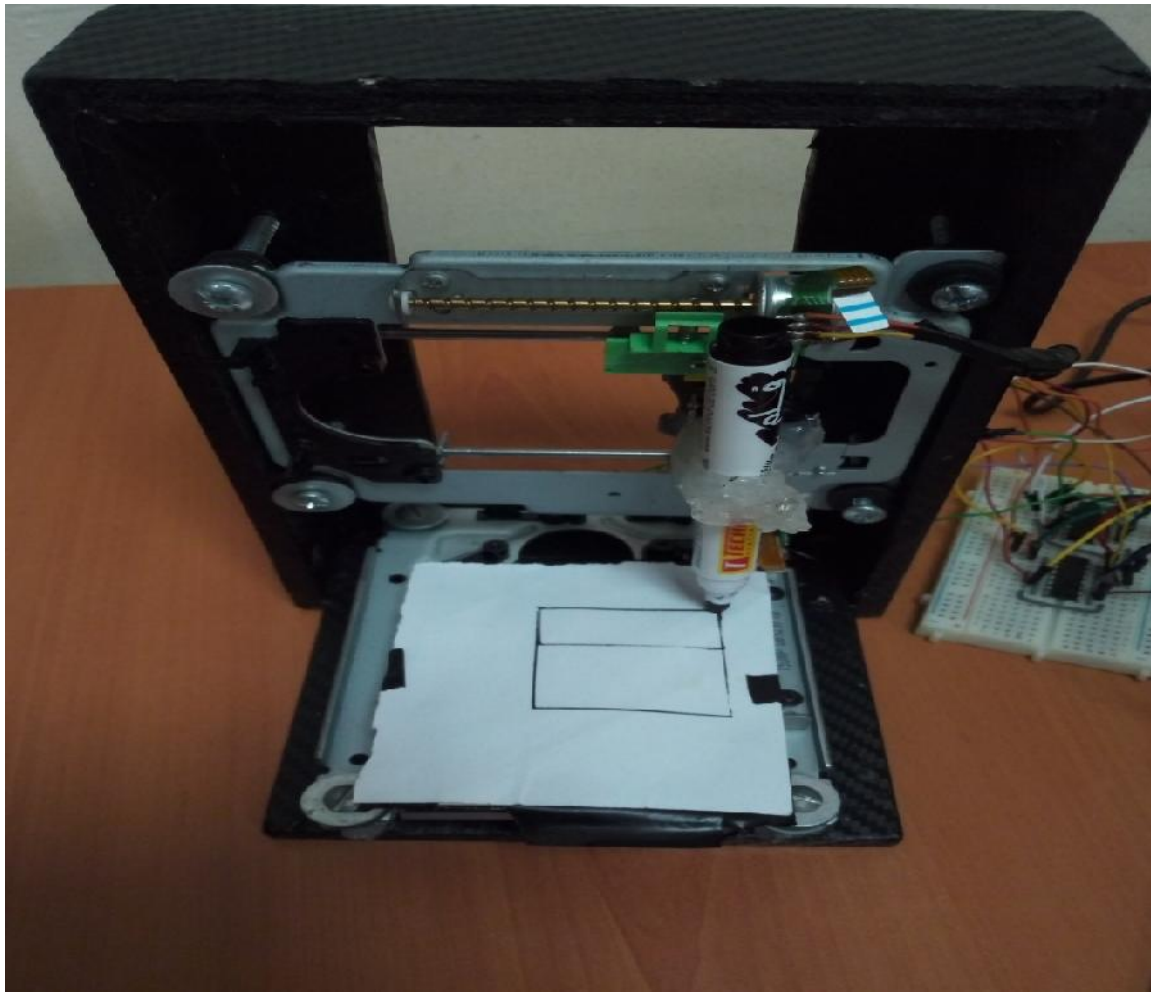
III.6 Traçage du premier dessin

Figure III.7 : Exemple d'une forme réalisée par la machine.

CONCLUSION

Ce mémoire est le résultat d'un travail de recherche dans le domaine de la commande des machines électriques.

Dans ce travail nous avons présenté quelque généralité sur les machines CNC afin d'introduire notre propre machine CNC.

Pour réaliser notre machine CNC nous avons partagé le travail en deux phases. La conception de la machine CNC, ou nous avons spécifié les différents composants nécessaires et étudié leur structure et leur architecture, ensuite, nous sommes passés à la réalisation. Dans cette partie nous avons étudié les spécifications de chaque composant utilisé et l'architecture de notre machine.

Une fois la réalisation finie, nous sommes passés aux tests. Les résultats obtenus sont très satisfaisants.

Comme perspective future, nous proposons :

- Le contrôle des autres axes (imprimante 3D).
- L'application de la commande avec un modèle intelligent.
- L'utilisation de la communication sans fils entre le système d'acquisition et le PC.

BIBLIOGRAPHIE

Les mémoires

Bentaleb,M / gasmi,E . Réalisation et commande d'une machine cnc à base des moteurs pas à pas. UNIVERSITE KASDI MERBAH OUARGLA,2016

Mr. MAHDJOUBLA. Etude et réalisation d'une machine outil à Nommande Numérique (fraiseuse 3 axes). UNIVERSITE M'HAMED BOUGARA BOUMERDES, 2017.

IAMRACHE, M /BENTOUMI,N-étude et réalisation d'un bras robotique .Université Mouloud Mammeri ,2017.

Livre

BARTMAN, E.2015, le grand livre d'arduino.2eme édition EYROLLES

Patrice, O, 2004 Moteurs pas-à-pas et PC - 2ème édition
ETSF, Dunod

Liens internet

<https://www.instructables.com>

<http://fablab37110.ovh/doku.php?id=start:cnc:grbl>

<http://fritzing.org/building-circuit/>

<https://fr.wikipedia.org>

Résumé

Dans ce mémoire nous nous sommes intéressés à la commande numérique, plus précisément aux machines CNC.

Au cours de ce projet, nous nous sommes familiarisés avec les cartes de type Arduino et son environnement de développement IDE, le circuit L293D et les moteurs pas à pas.

Nous avons enrichies nos connaissances en commande numérique.

Nous avons appris à développer matériellement une machine CNC.

Mot clés : Commande numérique, carte Arduino, GRBL, circuit L293D.....