

UNIVERSITE KASDI MERBAH OUARGLA

Faculté des nouvelles technologies de l'information et de la communication
Département d'Electronique et des Télécommunications



Mémoire

MASTER PROFESSIONNEL

Domaine : Sciences et technologies

Filière : Electronique

Spécialité : Automatique

Présenté par :

Bentaleb mokhtar

gasmi elhadi

Thème

Réalisation et commande d'une machine cnc à base des moteurs pas à pas

Soutenu publiquement

Le : 31/05/2016

Devant le jury :

M. ATAMNA Noura	MA (A)	Président	UKM Ouargla
M. AOUF Anouar Essadate	MA (A)	Encadreur/rapporteur	UKM Ouargla
M. OUARHLENT Saloua	MA (A)	Examineur	UKM Ouargla
M. DAHRAOUI Nadia	MA (A)	Examineur	UKM Ouargla

Année Universitaire : 2015/2016

Remerciements

Remerciements Premièrement, nous remercions dieu, le tout puissant, de nous avoir donné la santé et la volonté pour faire cet humble travail.

Nous tenons à exprimer nos remerciements vifs à notre promoteur M. Aouf Anouar Essadate pour l'aide et pour ses conseils précieux qu'il nous a donné durant notre étude et la réalisation de ce travail.

Nos remerciements s'adressent ainsi aux :M. Le président et les membres de jury d'avoir accepté juger et évaluer notre travail .

Nous remercions aussi tous les professeurs et les enseignants, nos parents, nos frères et sœurs et nos chers amis, pour leurs aides et leurs sacrifices.

Dédicaces

Nous dédions cet humble travail.

A ceux qui ont nous aide à faire ce modeste travail par leurs conseils et leurs encouragements :

- Nos parent
- Notre encadreur **M. Aouf Anouar Essadate**
- Nos professeurs
- Nos chers amis
- A toute la promotion 2016

Nous les remercions et les dédions ce travail

Sommaire

Introduction générale.....	1
----------------------------	---

Chapitre I : Moteurs pas à pas

I.1 Introduction.....	3
I.2 Historique	3
I.3 Principe de fonctionnement d'une machine numérique	4
I.4 La commande numérique par ordinateur(CNC)	5
I.4.1 La Conception Assistée par Ordinateur (CAO)	5
I.4.2 La Fabrication Assisté par Ordinateur (FAO).....	5
I.4.3 Le post-processeur.	6
I.5 Moteur pas à pas.....	6
I.5.1 Historique.....	7
I.5.2 Généralités.....	7
I.6 Les différents types de moteur pas à pas.....	7
I.6.1 Les moteurs à réluctance variable.....	8
I.6.1.1 Le principe de fonctionnement est très simple.....	8
I.6.2 Moteur à aimants permanents.....	9
I.6.2.1 Moteur à aimant permanent bipolaire.....	9
I.6.2.2 Moteur à aimant permanent unipolaire.....	11
I.6.2.3 Etats successifs des phases du moteur unipolaire.....	13
I.6.2.4 fonctionnement du moteur unipolaire en mode demi-pas.....	13
I.6.3 Moteur pas-à-pas hybride MH.....	14
I.6.3.1 Technologie des moteurs pas à pas hybrides.....	14
I.7 Comparaison entre les différents types des moteurs pas à pas.....	15
I.8 CONCLUSION.....	15

Chapitre II : description des composants électronique de la carte de commande

II.1. Introduction.....	16
-------------------------	----

II.2. Mise en œuvre de la carte Arduino Uno.....	16
II.2.1. présentation et caractéristiques du module Arduino Uno.....	16
II.2.2. Description matérielle.....	18
II.2.2.1. Alimentation.....	18
II.2.2.2. Connecteurs E/S.....	20
II.2.2.3. Leds externes.....	20
II.2.3. Description logicielle.....	20
II.2.3.1. Logiciel d'édition et programmation Arduino.....	21
II.2.3.2. Structure d'un programme Arduino.....	22
II.3 Commande moteur pas à pas.....	22
II.3.1 alimentation.....	23
II.3.2 Commande directe avec un amplificateur L298.....	23
II.3.2.1 Circuit L298.....	23
II.3.2.2 La programmation.....	24
II.3.3 La commande dir/step.....	26
II.3.3.1 Circuit L297.....	26
II.3.3.2 Carte EasyDriver 4.4.....	28
II.3.3.3 programmation de La commande dir/step.....	29
I.4 Conclusion.....	30

Chapitre III : Programmation et interface graphique

III.1. Introduction.....	31
III.2. Langue de contrôleurs numériques « G-code ».....	31
III.2.1 description.....	31
III.2.2 Format d'une ligne.....	32
III.3. L'éditeur du texte ou image compilateur de langage CNC	32
III.3.1 Definition Inkscape.....	32
III.4. Logiciel de pilotage de la machine CNC.....	33
III.4.1 modes de commande	34

III.4.2 mode Configuration.....	36
III.4.3 mode manuel	36
III.4.3. mode automatique	37
III.5. Langage de programmation la machine CNC.....	38
III.5.1 Programme Grbl	38
III.5.1.2 code de l'opération	39
III.5.1.3 L'interpolation de Bressenham	40
III.5.1.4 Algorithme adapté pour le tracé de tout segment	42
III.5.1.5 Bibliothèques de la commande des moteurs	44
III .6 Organigramme de la commande de la machine CNC.....	44
III.7 Conclusion :.....	46

Chapitre IV : Fabrication et la commande de la machine CNC

IV.1 : Introduction.....	47
IV.2 Simulation électrique sous Porteuse Professionnel.....	47
IV.2.1 Présentation générale.....	47
IV.2.2 ISIS.....	48
IV.2.3 ARES.....	48
IV.2.4 Simulation et test le programme.....	49
IV.2.4.1 Associer le programme au processeur.....	50
IV.3. fabrication de cnc machine 2D.....	51
IV.3 Conclusion.....	57
Conclusion générale.....	58

BIBLIOGRAPHIE.

ANNEXE.

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)
USART	Emetteur-récepteur synchrone/asynchrone universel (Universal Synchronous/Asynchronous Receiver Transmitter)
SCI	Interface de communication série (serial communication interface)
ASCII	Code américain normalisé pour l'échange d'information (American Standard Code for Information Interchange)
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 Organization for Standardization)
DXF	Drawing eXchange Format
IGES	Initial Graphics Exchange Specification

LISTE DES FIGURES

Figure I.1: Décomposition d'une machine numérique.....	4
Figure I.2: machine à trois axes.....	5
Figure I.3: Cycle d'une pièce fabriquée par une commande numérique (CFAO).....	6
Figure I.4: Commande des moteurs unipolaire et bipolaire.....	8
Figure I.5: Représentation schématique d'un moteur bipolaire.....	9
Figure I.6: Fonctionnement à pas complet	10
Figure I.7: Fonctionnement avec couple maximal.....	10
Figure I.8: Fonctionnement à demi-pas	11
Figure I.9: Représentation schématique d'un moteur unipolaire.....	11
Figure I.10: fonctionnement d'un moteur pas-a-pas unipolaire.....	13
Figure I.11 : Moteur hybride (MH).....	14
Figure II.1 : Module ARDUINO ATMEL UNO.....	17
Figure II.2 : Interface du logiciel Arduino.....	21
Figure II.3 : commande moteur par le circuit L298.....	24
Figure II.4 Figure : commande moteur par le couple L297 et L298 arduino.....	27
Figure II.5 : le circuit EasyDriver.....	28
Figure II.6 : commande moteur par Easy Driver.....	29
Figure.III.1: l'interface principale de Inkscape.....	33
Figure III.2: de Universal Gcode Sender.....	33
Figure III.3: l'interface de "g code sender" on mode manuel.....	34
Figure III.4: l'interface de "g code sender" choisir le fichier Gcode.....	34
Figure III.5: l'interface de "g code sender" on mode automatique	35
Figure III.6: envoyé fichée G code	35

Figure III.7: de mode manuel.....	36
Figure III.8: de GRBL code.....	38
Figure III.9 : Bons tracés.....	41
Figure III.10 : Mauvais tracés.....	41
Figure III.11: Interpolation linéaire de Bresenham.....	42
Figure III.12: Adaptation de l'interpolation de Bresenham	42
Figure III.13 : Organigramme de la commande de la machine CNC.....	45
Figure IV.1 : Logiciel ISIS.....	48
Figure IV.2 : Logiciel ARES.....	48
Figure IV.3 : Montage électronique dans Proteus.....	49
Figure IV.4 : simulation cnc machine.....	50
Figure IV.5: charge le fichier hex de programme.....	51
Figure IV.6 : Système de déplacement linéaire récupéré sur un lecteur de DVD.....	52
Figure IV.7: Souder les fils au niveau du moteur.....	52
Figure IV.8: montage cnc machin.....	53
Figure IV.9: cnc machine.....	54
Figure IV.10: arduino et Breadboard de cnc machin.....	55
Figure IV.11: test les axes X, Y et Z.....	55
Figure IV.12: universel G code sender.....	56
Figure IV.13: exemple dessin cnc machine.....	57

Liste des tableaux

Tableau I.1 : le mode monophasé biphasé ou demi-pas.....	8
Tableau I.2: Table Etat successifs des phases.....	13
Tableau I.3: Etat successifs des phases lors de l'utilisation des demi-pas.....	13
Tableau I.4: Comparaison entre les différents types des moteurs pas à pas.....	15

INTRODUCTION GENERALE

Aujourd'hui, les machines commande numérique par ordinateur (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.

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 une possibilité d'implanter des algorithmes plus complexes. Ces commandes sont en majorité basées sur les microprocesseurs, les DSP (Digital Signal Processor) et les microcontrôleurs PIC. Ces processeurs sont équipés d'unité arithmétique et logique (UAL) dédiée à la réalisation des calculs arithmétiques. Ils intègrent également des périphériques tels que les convertisseurs analogiques/numériques et les « timers » bien adaptés aux besoins de commande de machines électriques.

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 incrémental. Pour cela on utilise le port USB d'un ordinateur de type PC qu'on commande avec une carte Arduino Uno et des circuits intégrés.

Le travail est structuré comme suit:

Le premier chapitre, fera l'objet de la présentation de la théorie de la machine CNC et la description des moteurs pas à pas est faite d'une façon qui permet d'avoir une idée sur la richesse et la diversité des solutions qui peuvent être ajoutées grâce à ce moteur.

Le deuxième chapitre est consacré à la description des composants électroniques en vue de la commande des moteurs pas à pas, basée sur la carte Arduino Uno et les circuits intégrés (L297, L298 et Easy Driver). Un exemple de contrôle d'un moteur pas à pas est traité dans un but de montrer l'application de cette commande dans notre cas.

INTRODUCTION GENERALE

Le troisième chapitre concerne l'étude de la programmation et interface graphique et son application à la machine CNC. La présentation du logiciel Inkscape est un logiciel de dessin vectoriel qui offre une interface graphique de gestion des images vectorielles .

Dans le dernier chapitre, on présente le banc d'essai expérimental utilisé pour tester en temps réel les algorithmes de commande développés dans les parties précédentes. Les résultats obtenus sont présentés de façon à les comparer avec ceux de la simulation et aussi vérifier si les améliorations apportées en simulations seront gardées en expérimentation.

La dernière partie de ce mémoire englobe une conclusion générale et les perspectives du travail.



CHAPITRE I :
Moteur pas à
pas

I.1. 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 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 ordinateur (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 premier chapitre présente le contexte d'étude à travers une description des machines CNC et les moteurs pas à pas.

I.2. Historique

Les travaux menés par Falcon et Jacquard à la fin du XVIIIe 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. [1]

I.3. 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 ...)[2].

Ce type de machine se compose ainsi de deux parties complémentaires (figure I.1):

- La partie opérative.
- La partie commande.

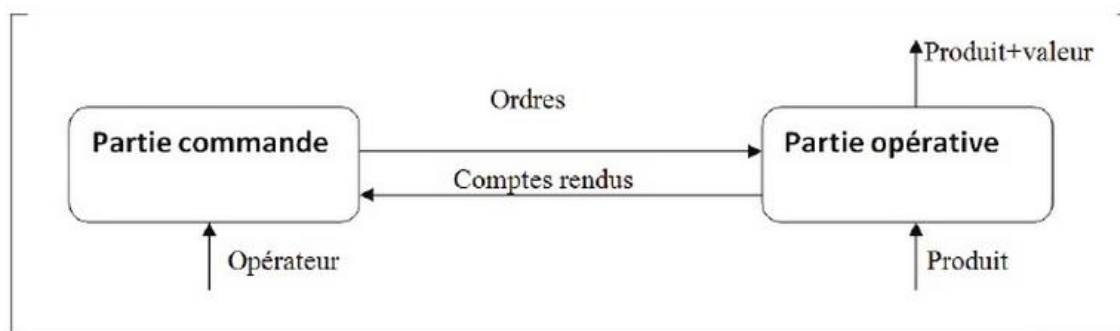


Figure I.1 Décomposition d'une machine numérique.

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 (O_m) 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 (O_p) qui peut être décalée par rapport à l'origine machine.
- L'origine Programme (O_P) qui est généralement confondu avec O_p pour faciliter l'usinage.

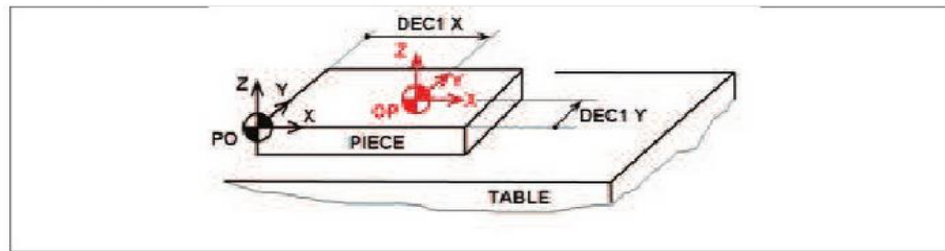


Figure I.2: machine à trois axes

I.4. La commande numérique par ordinateur(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 ordinateur. De plus, elles sont capables de gérer plusieurs axes simultanément. On parle alors d'interpolation. Cette dernière peut être linéaire ou circulaire selon la consigne générée par le microcontrôleur de la partie commande. Le rôle de ce processeur est d'interpréter un code machine que l'on appelle un code G, puis de générer les signaux de commande des axes et de la broche.

I.4.1. 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.

I.1.2. 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.

I.4.3. 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 tient 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 I.3)

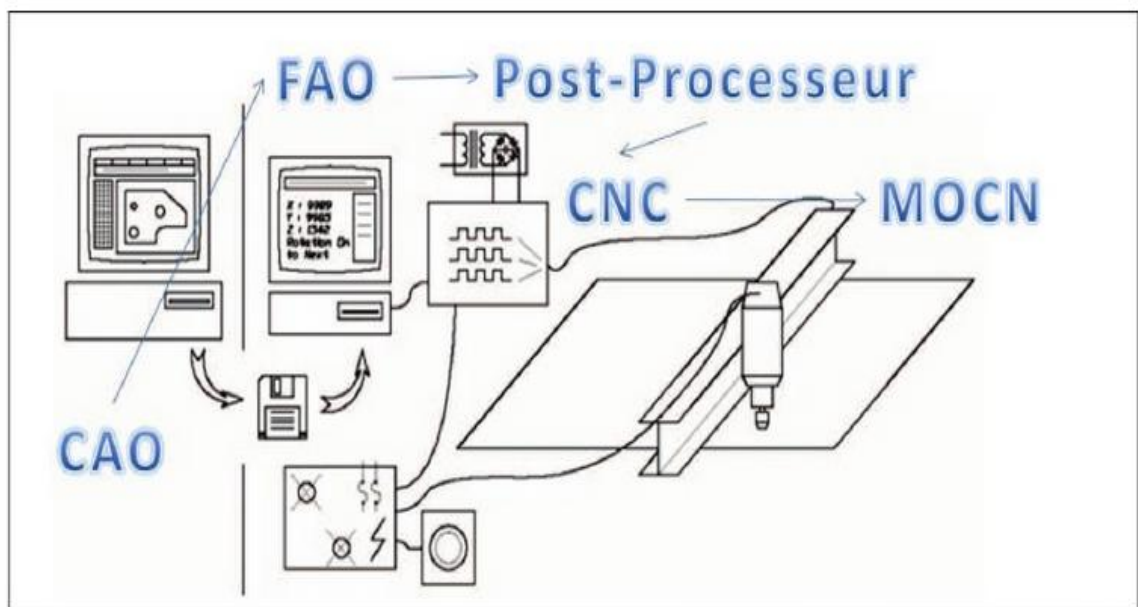


Figure I.3 : Cycle d'une pièce fabriquée par une commande numérique (CFAO)

I.5. Moteur pas à pas

On constate qu'il existe différents types de moteur pas à pas. Cependant le problème qui se présente maintenant c'est comment commander ces moteurs autrement dit comment peut-on faire varier le sens de rotation du moteur, l'angle de rotation, etc... Ce problème peut être facilement résolu en utilisant des circuits intégrés spécialisés dans la commande des moteurs pas à pas. Ces circuits sont nombreux dont on va choisir un des plus efficace.

I.5.1. Historique

Le moteur pas à pas a été inventé par Marius Lavet en 1936. Il s'agit d'un moteur électrique qui transforme une impulsion électrique en mouvement angulaire. Il possède un élément mobile se déplaçant par sauts successifs. Mis au point pour la fabrication de montres, on le retrouve dans presque tous les modèles à quartz. Il est composé de plusieurs éléments et reçoit des impulsions qui provoquent un mouvement saccadé de la masse oscillante. Le moteur pas à pas est également utilisé dans le secteur de l'informatique et notamment dans les imprimantes reliées à un ordinateur et le lecteur DVD.

I.5.2. 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 :

Les pôles magnétiques de rotation de même nom se repoussent et les pôles des noms contraires s'attirent, le champ magnétique entraînera le rotor alimenté dans le même sens.

Ceci traduit le fait qu'on transforme une grandeur numérique en une grandeur analogique. La fréquence de rotation, ou vitesse est donc commandée par des impulsions (consigne de rotation) contrôlées elle-même par un dispositif électronique en technologie câblée programmée.

I.6. 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.

La commande des moteurs pas à pas est relativement simple car on n'a pas besoin d'accessoires tels que des codeurs pour connaître la position, chaque **impulsion du système** de commande les fait avancer d'un pas. Il existe 3 types de moteurs pas à pas, à aimants permanents, à réluctance variable ou hybrides.

I.6.1. 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 I.4). On peut voir, dans cet exemple, que le stator est composé de 8 plots sur lesquels sont enroulés les bobinages, ce qui donne 4 phases. Le rotor, quant à lui, ne comporte que 6 dents [3].

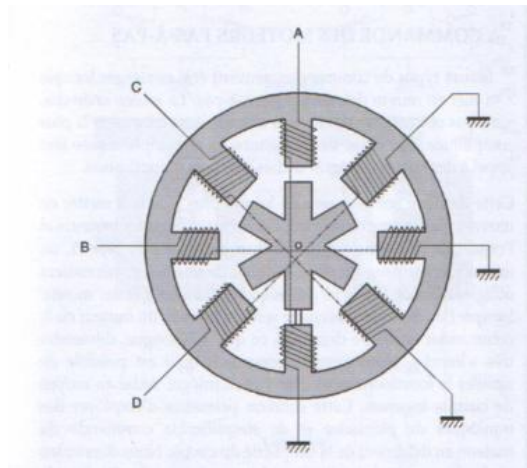


Figure I.4: moteur à réluctance variable

I.6.1.1. Le principe de fonctionnement est très simple

Le principe de fonctionnement est très simple : l'un des bobinages alimentant deux des plots opposés du stator est traversé par un courant afin d'aligner deux des dents du rotor avec ces plots. 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 suivant :

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

Tableau I.1 : le mode monophasé biphasé ou demi-pas

I.6.2 Moteur à aimants permanents:

Les moteur à aimants permanent sont semblables aux moteurs à reluctance variable, sauf que le rotor possède des pôles NORD et SUD .A cause des aimants permanent, 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'aiment va s'aligner avec le champ.

I.6.2.1 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 statorique[4].

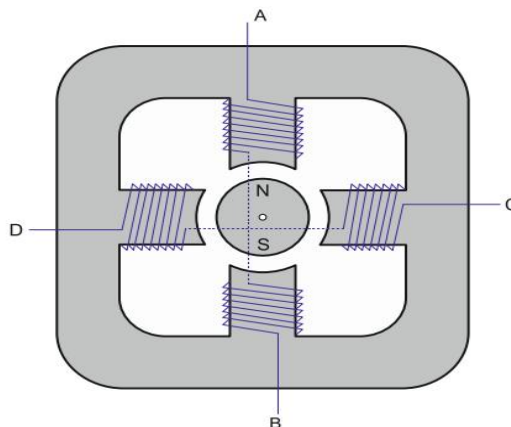


Figure I.5 : Représentation schématique d'un moteur bipolaire

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

I.6.2.1.1. Fonctionnement à pas complet

La première consiste à alimenter les enroulements en suivant la séquence A vers B/C vers D/B vers A/D vers C (BA est les mêmes enroulements que AB mais alimenté par un courant

de polarité inverse). Par la suite nous simplifierons la notation pour une meilleure correspondance avec les chronogrammes des phases en indiquant uniquement la phase qui est alimentée par un courant "positif". Soit A B C D.

Cette séquence est connue sous le nom de "one phase on full step" (traduisez phase par phase ou une phase à la fois en pas entier). A tout moment il n'y a qu'une seule phase d'alimentée et nous travaillons en mode pas entier.

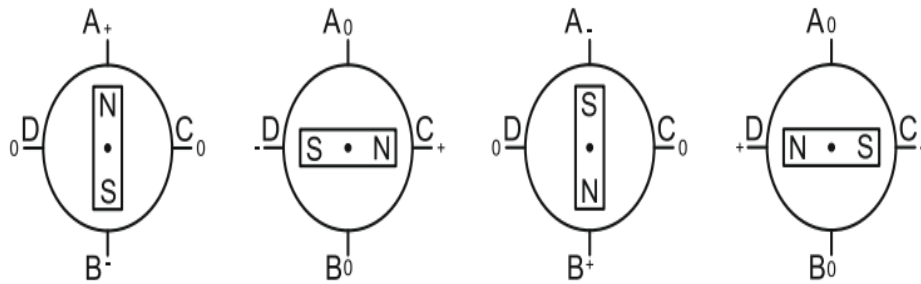


Figure I.6 : Fonctionnement à pas complet

I.6.2.1.2. Fonctionnement avec couple maximal

La seconde possibilité est d'alimenter une paire de phase en même temps de façon à ce que le rotor se positionne entre deux pôles.

Appelé "two-phase-on full step" (deux phases à la fois en pas entier) ce mode de commande est celui qui procure le couple le plus élevé.

La séquence sera donc AC/CB/BD/DA.

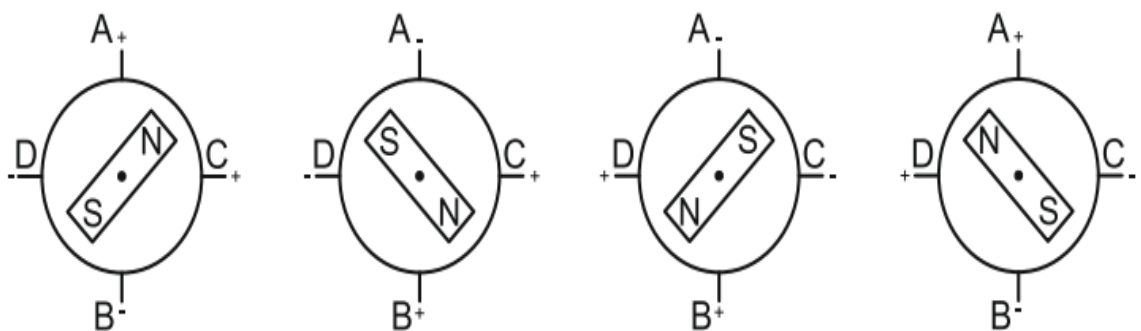


Figure I.7 : Fonctionnement avec couple maximal

I.6.2.1.3. Fonctionnement à demi-pas

La troisième option est un mélange des deux premières puisque l'on alimente tour à tour le moteur sur une phase puis deux puis une ... cette séquence connue sous le nom de mode demi pas procure effectivement une division par 2 de l'angle d'avance d'un pas, mais aussi un couple moins régulier.

La séquence qui en découle est la suivante : A/AC/C/CB/B/BD/D/DA.

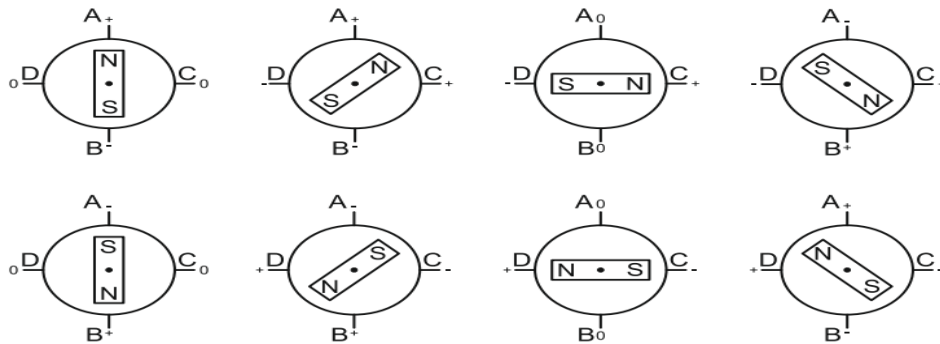


Figure I.8 : Fonctionnement à demi-pas

Pour obtenir une rotation dans la direction opposée les mêmes séquences sont utilisées en inversant l'ordre de défilement. Comme nous pouvons le constater sur les diagrammes, les moteurs sont représentés avec une avance de pas à 90°.

Dans les moteurs réels les nombres de pôles ont été multipliés pour réduire à quelques degrés seulement l'angle d'avance d'un pas. Le nombre d'enroulements et la séquence de commande restent, quant à eux, inchangés.

I.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 (soit quatre Darlington pour un moteur ou un réseau de 4 transistors- voir ULN 2075B).

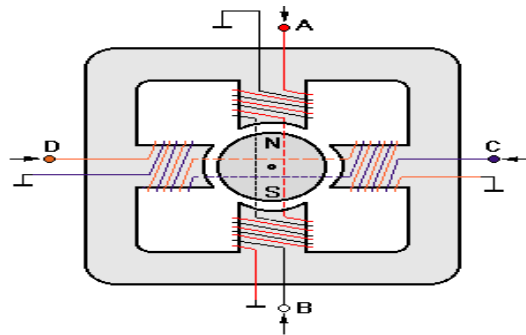


Figure I.9: Représentation schématique d'un moteur unipolaire

Simplement, les moteurs unipolaires sont plus chère car leur fabrication réclame un double bobinage. De plus, pour une taille donnée, ce type de moteur à un couple plus faible à cause des enroulements qui sont plus fins.

Il fût 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 L298, les moteurs bipolaires sont devenu populaires et d'une utilisation courante.

Tous les moteurs à aimant permanent souffrent des oscillations (et des harmoniques qui s'ensuivent) générées par le rotor qui limitent la vitesse de rotation.

Quand des accélérations et des vitesses plus élevées sont nécessaires on utilisera de préférence les moteurs à reluctance variable [4].

I.6.2.2.1. fonctionnement d'un moteur pas-a-pas unipolaire

On partira du principe que la rotation d'un moteur pas à pas s'effectue en 4 étapes, dans la réalité, le moteur est constitué d'une succession d'alternance de pôles : ainsi, l'axe du modèle dont nous disposons dans notre réalisation fait un tour complet en 48 pas (un pas correspond donc à $360^\circ/48 = 7,5^\circ$).

Dans les schémas, la flèche noire représente l'aiguille d'une boussole qui serait disposé 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[3].

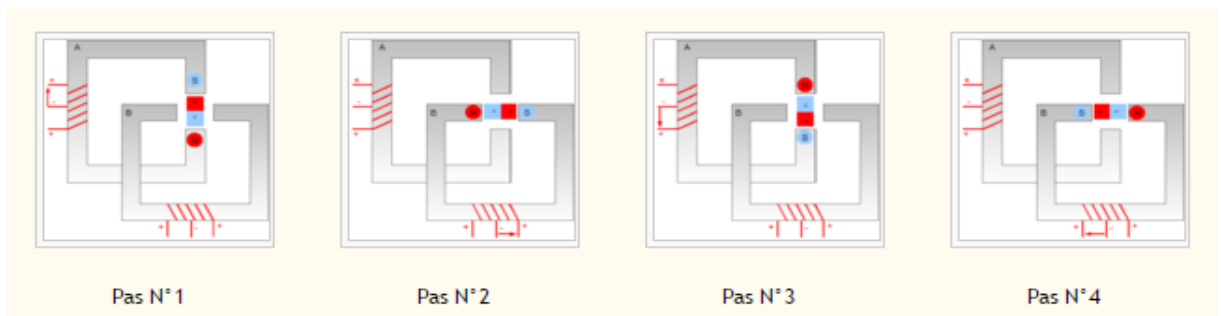


Figure I.10: fonctionnement d'un moteur pas-a-pas unipolaire

I.6.2.3. Etats successifs des phases du moteur unipolaire

La table de vérité B.2.2.1 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

Table I.2: Table Etat successifs des phases

I.6.2.4. fonctionnement du moteur unipolaire en mode demi-pas

Le moteur de notre schéma effectue une rotation en quatre pas. Il se caractérise par un fonctionnement dit "par pas" il existe également un mode de fonctionnement par "demi pas" : il consiste à intercaler entre deux étapes, une période au cours de laquelle on coupe l'alimentation du bobinage du stator dont l'alimentation s'apprête à changer de sens (elle passe donc par zéro) : durant cette nouvelle étape le rotor tourne d'un demi pas (45°) en s'alignant table de vérité des phases est représentée en table suivante :

Numéro de l'étape :	1	2	3	4	5	6	7	8
Bobinage 1, phase 1	0	0	1	1	1	0	0	0
Bobinage 1, phase 2	1	0	0	0	0	0	1	1
Bobinage 2, phase 1	1	1	1	0	0	0	0	0
Bobinage 2, phase 2	0	0	0	0	1	1	1	0

Table I.3: Etat successifs des phases lors de l'utilisation des demi-pas.

Le mouvement s'effectue à la suite d'une inversion du champ magnétique en alimentant l'une ou l'autre des phases d'un bobinage à point milieu ; seule une moitié du bobinage est donc utilisée à un instant donné. Un autre type de moteur. Dit moteur à deux phases, permet d'obtenir un couple plus important : son principe consiste à utiliser un bobinage sans point milieu, et à faire circuler le courant dans un sens ou dans l'autre...

I.6.3 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 (Hybrid motor : HB). Dans ce cas, il existe un couple réluctant provoqué par la variation de perméances propres associées à chaque aimant et à chaque bobine [3].

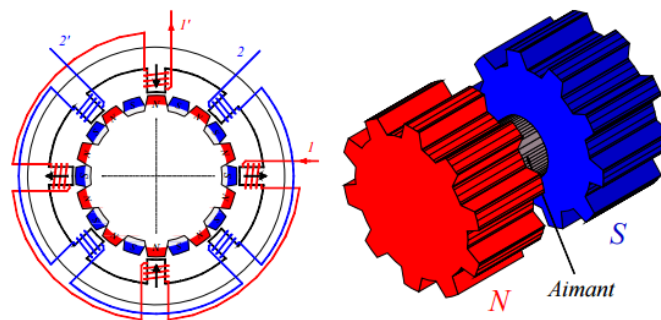


Figure I.11 : Moteur hybride (MH)

I.6.3.1. 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.

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 $1/2\tau_{dr}$ [3].

I.7. Comparaison entre les différents types des moteurs pas à pas

Cette comparaison peut se résumer selon le tableau suivant :

Critère de comparaison	Moteur pas à pas aimant permanent	Moteur pas à pas reluctance variable
Couple de travail	Elevé	faible
Vitesse de fonctionnement	faible	faible
Pas exécuté	faible	Elevé
Inertie propre	Elevé	faible
Couple de maintien	Non négligeable	négligeable
Amortissement des oscillations	Plus important	Moins important

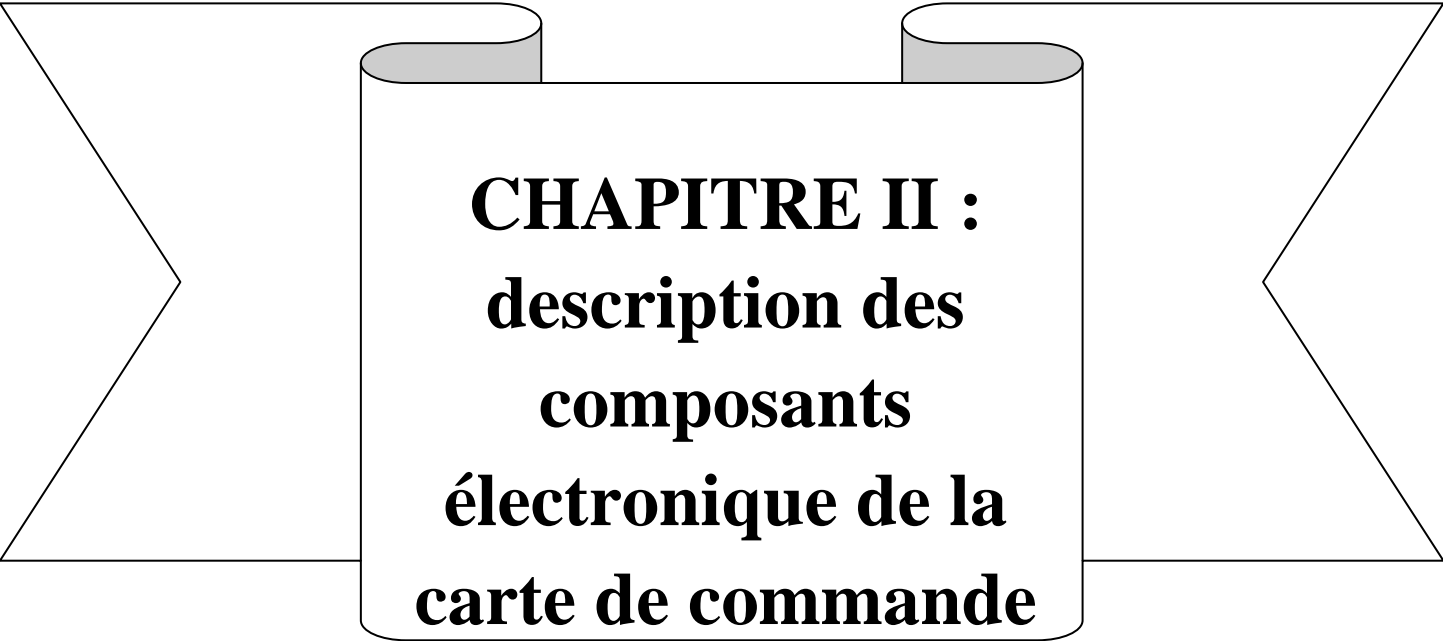
Table I.4: Comparaison entre les différents types des moteurs pas à pas

I.8. Conclusion

Ce premier chapitre s'appuie essentiellement sur quelques notions de la machine numérique, des moteurs pas à pas et de la commande numérique, elles sont nécessaires pour tenir compte des contraintes que toute réalisation se rapporte à ces études qui devraient être

On constate qu'il existe différents types de moteur pas à pas. Cependant le problème qui se présente maintenant c'est comment commander ces moteurs autrement dit comment peut-on faire varier le sens de rotation du moteur, l'angle de rotation, etc...

L'utilisateur peut choisir le moteur qui lui convient à son application, avec l'alimentation et le mode d'excitation correspondant. Pour notre projet, on a opté pour trois moteurs pas à pas, bipolaire.



CHAPITRE II :
description des
composants
électronique de la
carte de commande

II.1. 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.

II.2. 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 coût. Elle possède une interface USB pour la programmer. C'est une plateforme open-source qui est basée sur une simple carte à 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 utilisé 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. [5]

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 téléversé sur la carte Arduino. Ainsi, lorsque vous apportez des changements sur le code, ces changements ne seront effectifs qu'une fois le programme téléversé sur la carte.

II.2.1. présentation et caractéristiques du module Arduino Uno

La carte Arduino Uno, prêtée est une carte à 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.[6]

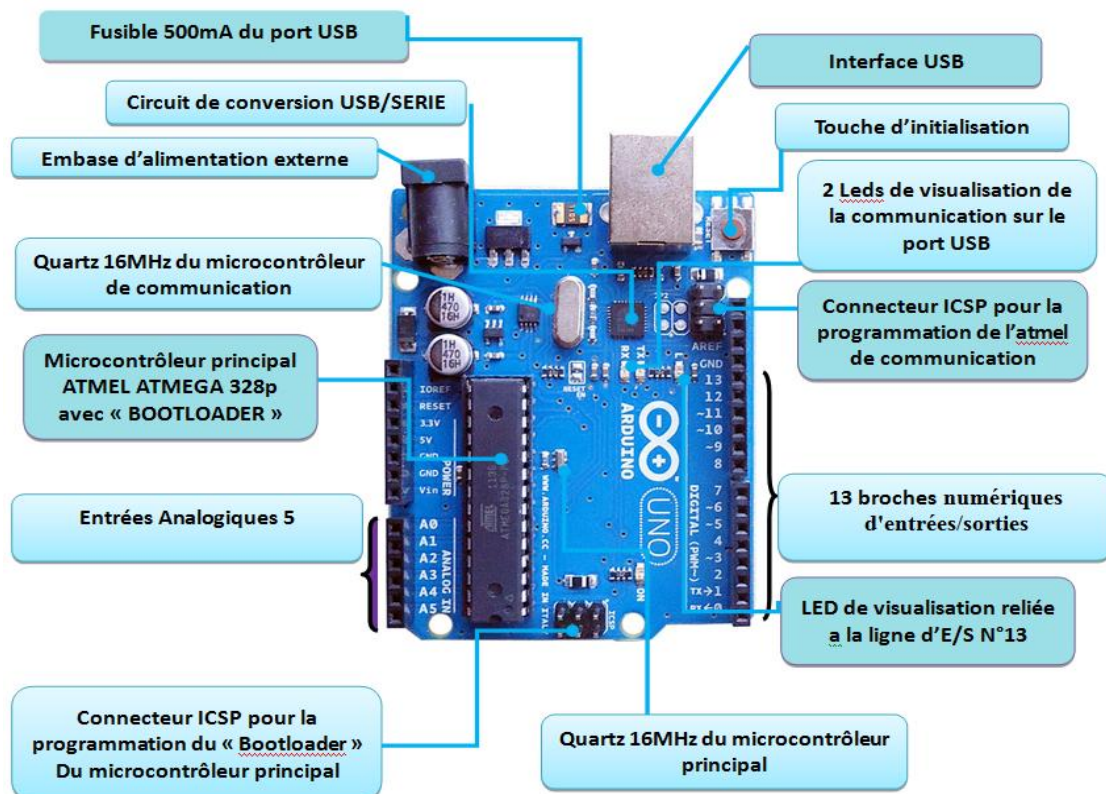


Figure II.1 : Module ARDUINO ATMEL UNO[7]

II.2.2. Description matérielle

Un module Arduino est généralement construit autour d'un microcontrôleur famille ATMEGA en technologie CMOS sont des modèles à 8 bits AVR. En exécutant des instructions dans un cycle d'horloge simple, l'ATMEGA réalise des opérations s'approchent de 1 MIPS par 1 MHz permettent de réaliser des systèmes à faible consommation électrique et simple au niveau électronique et des composants complémentaires qui facilitent la programmation et l'interfaçage avec d'autres circuits. Chaque module possède au moins un régulateur linéaire 5V et un oscillateur à quartz 16 MHz (ou un résonateur céramique dans certains modèles).[8]

Le microcontrôleur est préprogrammé avec un bootloader de façon à ce qu'un programmeur dédié ne soit pas nécessaire. Mais les connexions permettent cette programmation différente selon les modèles. Les premiers Arduino possédaient un port série, puis l'USB est apparu sur les nouveaux modèles, tandis que certains modules destinés à une utilisation portable se sont affranchis de l'interface de programmation, relocalisée sur un module USB-série dédié (sous forme de carte ou de câble).

La composition des éléments internes d'un microcontrôleur :

- ✚ **RAM :** c'est la mémoire dite "vive", elle va contenir les variables de votre programme. Elle est dite "volatile" car elle s'efface si on coupe l'alimentation du microcontrôleur (comme sur un ordinateur).
- ✚ **EEPROM:** C'est le disque dur du microcontrôleur. Vous pourrez y enregistrer 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.
- ✚ **Les registres:** c'est un type de mémoire utilisé par le processeur.
- ✚ **La mémoire cache:** c'est une mémoire qui fait la liaison entre les registres et la RAM. Nous n'en parlerons également pas tout de suite.
- ✚ **Le processeur:** C'est le composant principal du microcontrôleur. C'est lui qui va exécuter le programme que nous lui donnerons à traiter. On le nomme souvent le CPU.[9]

II.2.2.1. Alimentation

La carte Arduino Uno peut-être alimentée soit via la connexion USB (qui fournit 5V jusqu'à 500mA) ou à l'aide d'une alimentation externe. La source d'alimentation est sélectionnée automatiquement par la carte.

L'alimentation externe (non-USB) peut être soit un adaptateur secteur (pouvant fournir typiquement de 3V à 12V sous 500mA) ou des piles. L'adaptateur secteur peut être connecté en branchant une prise 2.1 mm positif au centre dans le connecteur jack de la carte. Les fils en provenance d'un bloc de piles ou d'accus peuvent être insérés dans les connecteurs des broches de la carte appelées GND (masse ou 0V) et Vin (Tension positive en entrée) du connecteur d'alimentation. La carte peut fonctionner avec une alimentation externe de 6 à 20 volts. Cependant, si la carte est alimentée avec moins de 7V, la broche 5V pourrait fournir moins de 5V et la carte pourrait être instable. Si on utilise plus de 12V, le régulateur de tension de la carte pourrait chauffer et endommager la carte. Aussi, la plage idéale recommandée pour alimenter la carte Uno est entre 7V et 12V. [14]

Connecteur alimentation

Le connecteur Power 8 broches (6 dans le cas de certaines cartes IoRef et la broche de réserve n'est pas implanté) a les fonctions suivantes :

- 1 : Reserve
- 2 : IoRef, relié au +5v
- 3 : Entrée ou sortie reset
- 4, 5 : Sortie 3.3v, 5v
- 6, 7 : Masse générale, 0v
- 8 : Vin, sortie avant le régulateur et après la diode de protection, ne l'utiliser pas comme entrée d'alimentation à la place du connecteur jack coaxial d'alimentation.[9]

Les broches d'alimentation sont les suivantes :

- 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). Vous pouvez 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 le 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 toute autre source d'alimentation régulée.

- 3.3V : 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
- GND Broche de masse (ou 0V).[10]

II.2.2.2. Connecteurs E/S

Suivant les types de cartes celles-ci disposent d'un nombre de connecteurs plus ou moins importants, soit de type digital, soit de type mixte analogique / digital. Ces entrées / sorties possèdent les caractéristiques suivantes :

- ✚ Courant d'entrée : Environ 1μA
- ✚ Valeur résistance de Pull Up optionnelle : 20 a 50 KΩ
- ✚ Courant de sortie : 40 mA max, 200 mA total microcontrôleur
- ✚ Tension d'entrée : Ne pas dépasser Vcc.
- ✚ Caractéristique convertisseur AD : Par approximation successive, 10bits, temps de conversion 13 a 260μs.
- ✚ Référence de tension interne : 1.1 ou 2.56 V.

II.2.2.3. Leds externes

Les cartes standards sont généralement équipées de quatre leds.

- ✚ Vert : Alimentation 5v
- ✚ Jaune Tx : Commandée par l'interface USB, transmission au PC
- ✚ Jaune Rx : réception du PC
- ✚ Jaune : Utilisateur, généralement connectée à la sortie digitale 13. [5]

II.2.3. Description logicielle

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++, compilé avec avr-g++, et lié à la bibliothèque de développement Arduino, permettant l'utilisation de la carte et de ses

entrées/sorties. La mise en place de ce langage standard rend aisé le développement de programme 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 facilite des programmes, permet d'afficher sur l'ordinateur l'état de variables, de résultats de calculs ou de conversions analogique-numérique : un élément essentiel pour améliorer, tester et corriger ses programmes. [5]

II.2.3.1. Logiciel d'édition et programmation Arduino



Figure II.2 : Interface du logiciel Arduino.

Il est totalement gratuit et proposé en libre téléchargement sur le site Internet d'Arduino. Téléchargez et installez la dernière version que vous souhaitez sur votre ordinateur (Windows.

Linux ou Mac). Une fois décompressée comprend le logiciel, les drivers L'archive, exemples et bibliothèques. Après l'indispensable installation des pilotes, lancez le logiciel à partir du dossier. La fenêtre de l'application Arduino comporte les éléments suivants.

II.2.3.2. Structure d'un programme Arduino

Un programme Arduino doit impérativement être divisé en 2 parties : une fonction setup et une fonction loop(). Ces deux fonctions sont de type void(), c'est à dire qu'elles ne peuvent pas prendre de valeurs. La fonction setup est la fonction qui se lance au début du programme. Elle permet d'initialiser les variables et de définir les broches de la carte Arduino qui seront utilisées. La fonction loop se lance après la fonction setup et, comme son nom l'indique, fait une boucle jusqu'à ce que la carte soit débranchée. Les constantes du programme Pour notre programme, nous avons utilisé de nombreuses constantes globales, définies en tout début de programme.

II.3. Commande moteur pas à pas

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 ou EASY DRIVER.

Pour réaliser une commande numérique de trois moteurs pas à pas on étudie comment commande un seul moteur.

Nous avons divisé ce projet en divers unités :

- Une alimentation.
- Une unité principale.
- Un interfaçage entre l'unité principale et la partie opérative de moteur.

II.3.1 alimentation:

Pour assurer le fonctionnement de nos cartes électroniques, nous avons besoin d'une alimentation continue de 5VDC et 12VDC :

- 5V DC 200mA: alimentation pour la partie commande.
- 12V DC 2A: alimentation pour la partie puissance.

II.3.2 Commande directe avec un amplificateur L298:

Pour cette commande on utilise un circuit de commande qui donne un mode de commande directe de moteur pas à pas (mode à pas entier).

Mode à pas entier, deux phases alimentées en même temps (Two Phase ON, Full Step) :1010/0110/0101/1001.

II.3.2.1 Circuit L298 :

Le circuit intégré L298 est le circuit complémentaire du L297. Il permet la simplification extrême de la construction d'une carte de commande pour moteur pas-à-pas. Le circuit L298 est un double pont de commande de puissance, il permet l'utilisation d'une tension élevée. Il peut être utilisé aussi pour la commande de deux moteurs à courant continu. Le schéma interne du L298 contient deux ponts de 4 transistors de puissances commandées par les portes logiques, ainsi que la connexion des résistances de mesure du courant consommée par la charge reliée aux émetteurs de chaque paire de transistors. [11]

La technique de commande dans cette carte est basée sur un arduino en configurant pour chaque moteur quatre broches en sortie afin d'envoyer les séquences des impulsions générées par le ARDUINO sur les bornes du moteur pas à pas (figure II.3). Chaque moteur reçoit une séquence indépendamment de l'autre à travers un amplificateur L298 pouvant fournir un courant de sortie de 0,5A.

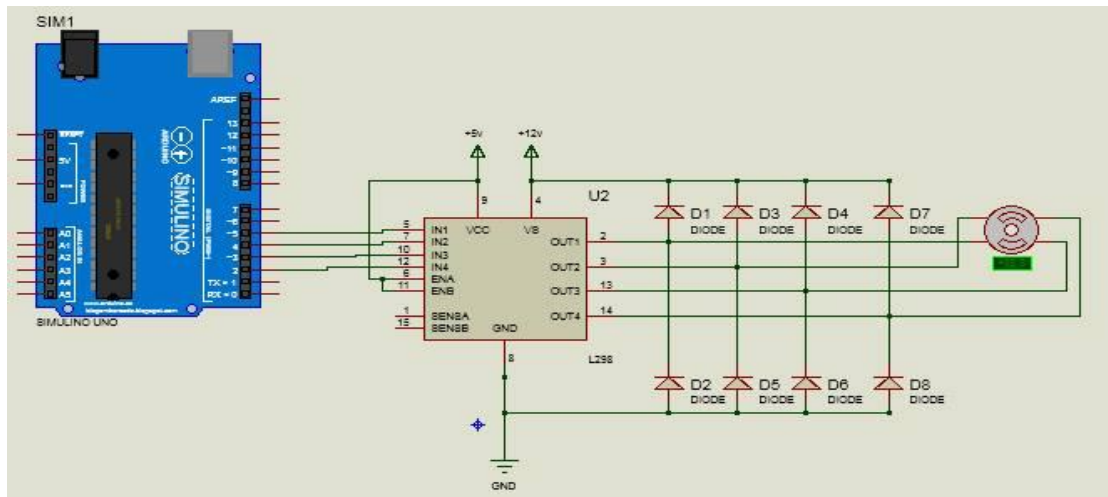


Figure II.3: commande moteur par le circuit L298.

II.3.2.2 La programmation :

Déclaration des constantes pour les broches utilisées dans le programme :

```
int inA1 = 2 , inA2 = 3 , inB1 = 4 , inB2 = 5 ;
```

```
int stepDelay = 25; //ce constantes utilise pour commande la vitesse de rotation.
```

Configuration des broches en sortie :

```
void stup () {
  digitalWrite(inA1, OUTPUT);
  digitalWrite(inA2, OUTPUT);
  digitalWrite(inB1, OUTPUT);
  digitalWrite(inB2, OUTPUT); }

```

Initialisation des fonctionnalités utilisées pour obtenir les pas :

```
void step1() { // La premier pas 0110 :
  digitalWrite(inA1, LOW);
  digitalWrite(inA2, HIGH);
  digitalWrite(inB1, HIGH);
  digitalWrite(inB2, LOW);

```

<pre>delay(stepDelay); }</pre>
<pre>void step2() { // La 2^{em} pas 0101 : digitalWrite(inA1, LOW); digitalWrite(inA2, HIGH); digitalWrite(inB1, LOW); digitalWrite(inB2, HIGH); delay(stepDelay); }</pre>
<pre>void step3() { // La 3^{em} pas 1001 : digitalWrite(inA1, HIGH); digitalWrite(inA2, LOW); digitalWrite(inB1, LOW); digitalWrite(inB2, HIGH); delay(stepDelay); }</pre>
<pre>void step4() { //La 4^{em}pas 1010: digitalWrite(inA1, HIGH); digitalWrite(inA2, LOW); digitalWrite(inB1, HIGH); digitalWrite(inB2, LOW); delay(stepDelay); }</pre>

Au niveau de la boucle principale, la fonction loop () :

<pre>void loop() { for (int i=0; i<=11; i++){ step1(); step2();step3();step4();} delay(2000); // Une pause de 2 seconde est réalisée entre chaque boucle</pre>

```
for (int i=0; i<=11; i++){ step3(); step2(); step1(); step4();  
}  
delay(2000);  
}
```

On réalise 2 boucles successives de comptage des pas :

- la première boucle, on assure un mouvement d'un pas à pas chaque passage à l'aide de fonction step1() step2() step 3() step4().
- la seconde boucle, on assure un mouvement d'un pas à chaque passage mais en sens inverse.

II.3.3. La commande dir/step

La commande dir/step dépend sur un circuit qui obtenir un commande direct soit Mode à pas entier, une phase alimentée à la fois (One Phase ON, Full Step) ou Mode deux phases alimentées en même temps (Two Phase ON, Full Step) .ce circuit commande par deux proche :

- Step : qui est l'horloge pour la vitesse. A chaque front descendant le moteur fera un pas.
- Dir : (ClockWise ou Counter ClockWise) qui décidera du sens de rotation du moteur

Pour réaliser cette commande numérique on doit étudier deux circuits de commande.

- le couple L297 et l298.
- Le circuit EASY DRIVER.

II.3.3.1.Circuit L297

Le circuit L297 est un contrôleur de moteur pas à pas, il fonctionne avec un circuit de puissance à double pont. Il lui suffit de lui fournir les signaux d'horloge (CLOCK : pour avancer des pas), de direction et de mode afin de piloter le moteur pas-à-pas. Le L297 génère alors la séquence de commande de l'étage de puissance. [11]

Le circuit L297 possède deux étages principaux :

- Un translateur (séquenceur) qui génère les différentes séquences de commande.

- Un double hacheur PWM (Pulse With Modulation) qui régule le courant traversant les bobinages du moteur.

Le translateur génère trois séquences différentes. Ces différentes séquences sont déterminées par le niveau logique appliqué sur l'entrée HALF/FULL :

- La commande en mode monophasé : une seule phase alimentée.
- La commande en mode biphasé : Deux phases alimentées.
- La commande en demi-pas : Alternativement une phase puis deux phases alimentées.

Le L297 génère deux signaux d'inhibition (INH1 et INH2) dans les modes monophasés et demi pas. Ces signaux qui sont appliqués directement aux entrées de validation du L298, permettent d'accélérer la décroissance du courant dans les bobinages moteur lorsque ceux-ci ne sont plus alimentés. Lorsque le L297 est utilisées pour la commande d'un moteur unipolaire, les hacheurs agissent sur ces lignes.

Une entrée «CONTROL» détermine le moment ou le hacheur devra agir sur les sorties A, B, C, D ou les entrées INH1 et INH2.

Le L297 contient un oscillateur interne qui commande les hacheurs. Lorsque le courant traverse l'un des bobinages du moteur atteint la tension qui est fixée par les résistances connectée aux entrées SENS1 SENS2, le comparateur correspondant interrompt l'alimentation du moteur jusqu'à la prochaine impulsion de l'oscillateur.[11]

II.3.3.1.1. Le schéma théorique du montage

L297 est un circuit qui contient toute la circuiterie de commande et de contrôle de moteurs pas à pas unipolaire et bipolaire. Ce circuit est utilisé conjointement avec un driver double pont tel que le L298 (figure I.4). L'ensemble forme une interface idéale pour le contrôle d'un moteur pas à pas par arduino.

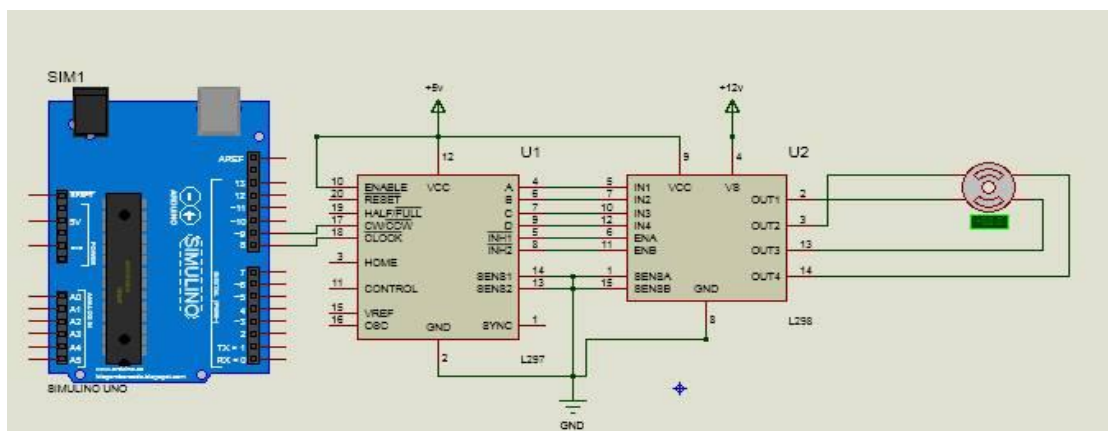


Figure II.4: commande moteur par le couple L297 et L298 arduino.

II.3.3.2 Carte EasyDriver 4.4:

Cette carte de commande permet de contrôler facilement un moteur pas-à-pas bipolaire jusqu'à 750 mA par phase. Elle fonctionne par défaut avec 8 micro-pas.

Elle est basée sur le circuit Allegro A3967 et permet de contrôler un moteur pas-à-pas (par ex: ITC-VNC-2) très facilement à partir d'un microcontrôleur (deux broches pour le mode, une broche pour la direction et une pour la vitesse). Le module permet de fonctionner en mode pas complet, demi-pas, quart de pas ou 1/8 de pas (mode par défaut)[12].

Pour simplifier la connectique, elle peut être installée sur une carte mini Shield.

- Alimentation: 7 à 30 Vcc maxi
- Commande: via un microcontrôleur 0 - 5 Vcc (mode, direction et vitesse)
- Sortie: 150 à 750 mA maxi par phase
- Dimensions: 48 x 21 x 18 mm
- Schéma: : EasyDriver v4.4
- Fiche technique: : EasyDriver Stepper Motor Driver
- Distributeur: GOTRONIC

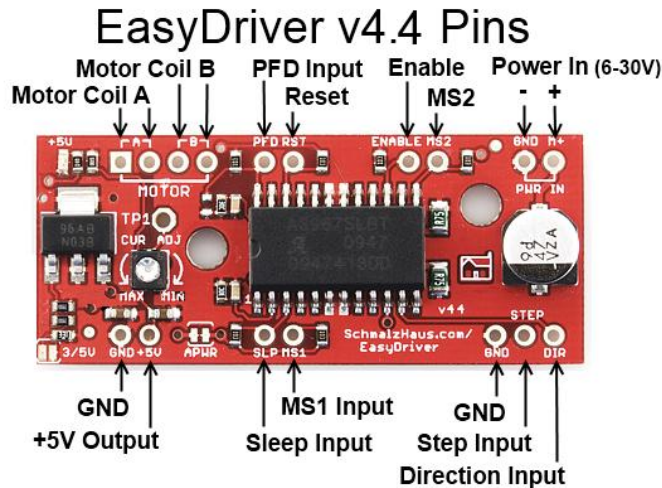


Figure II.5 : le circuit EasyDriver

II.3.3.2.1 Le schéma du montage Easy Driver

La commande avec ce circuit ne défère pas sur la commande avec le couple 1297-1298 qui contient un parte puissance et parte qui obtenir la séquence de pas. (pas entier ou les autre mode a partir le mode choisie).

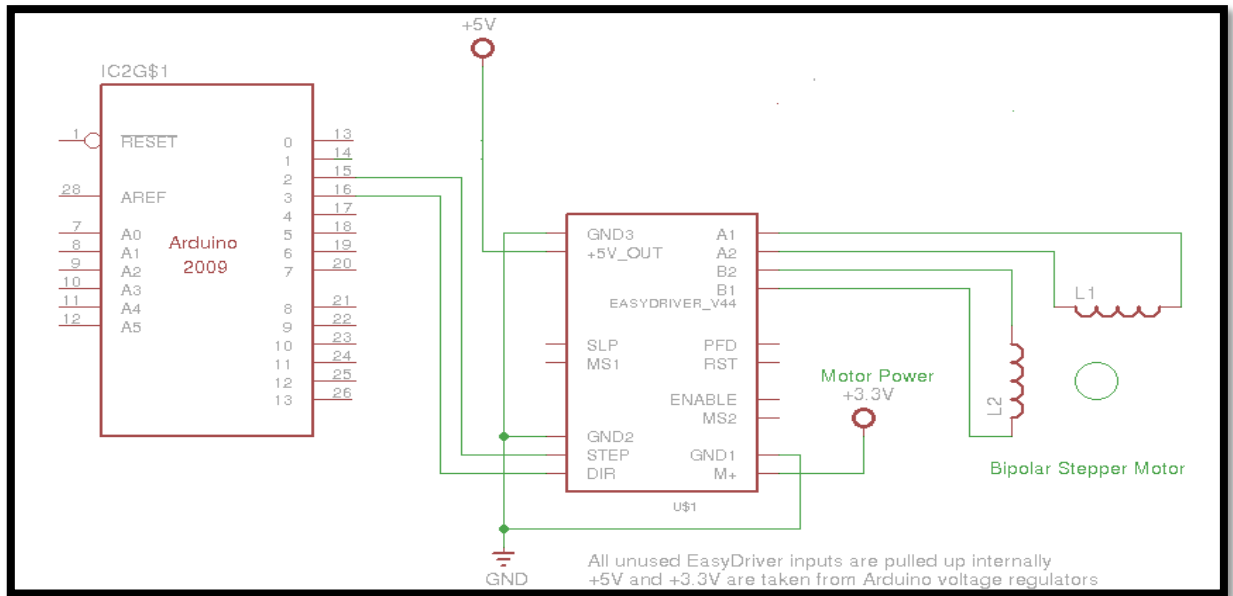


Figure II.6 : commande moteur par Easy Driver.

II.3.3.3. programmation de La commande dir/step

La commande à partir du couple L297-L298 ou Easy Driver est plus simple puisque on l'utilise que deux lignes de pilotage « Step et Dir ».

Déclaration des constantes pour les broches utilisées dans le programme :

```
int steppin = 2; // broches utilise pour la « step »
int dirpin = 3; //broches utilise pour la sens
int nbstep =1600 ; // nombre de pas
```

La partie du code suivante permet la Configuration des broches en sortie :

```
void stup () {
  pinMode(steppin, OUTPUT);
  pinMode(dirpin, OUTPUT);
}
```

Au niveau de la boucle principale, la fonction loop () :

```
Void loop() {
  digitalWrite(dirpin, HIGH); //dirpin = high pour sans direct low pour sans inverse
  for (i = 0; i<nbstep; i++) // boucle pour générer le signal rectangulaire
```


```
{  
digitalWrite(steppin, LOW);  
digitalWrite(steppin, HIGH); //transition état bas vers état haut (front montant) pour faire  
marcher le moteur.  
delayMicroseconds(500); }  
} //temporisation pour ajuster la vitesse de rotation
```

Vous pouvez ensuite ajuster la vitesse et nombre pas selon votre besoin.

I.4. 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.

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



**Chapitre III :
Programmation
et interface
graphique**

III.1. Introduction

Les machines-outils à commande numérique sont complètement pilotées par ordinateur. 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.

III.2. Langue de contrôleurs numériques « G-code »

III.2.1. 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 (G pour Général et M pour Miscellaneous (auxiliaire)). Les termes pour ces commandes sont *G-codes* et *M-codes*[13].

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* ou

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[13].

III.2.2. 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).

III.3. 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

III.3.1. 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 [14].

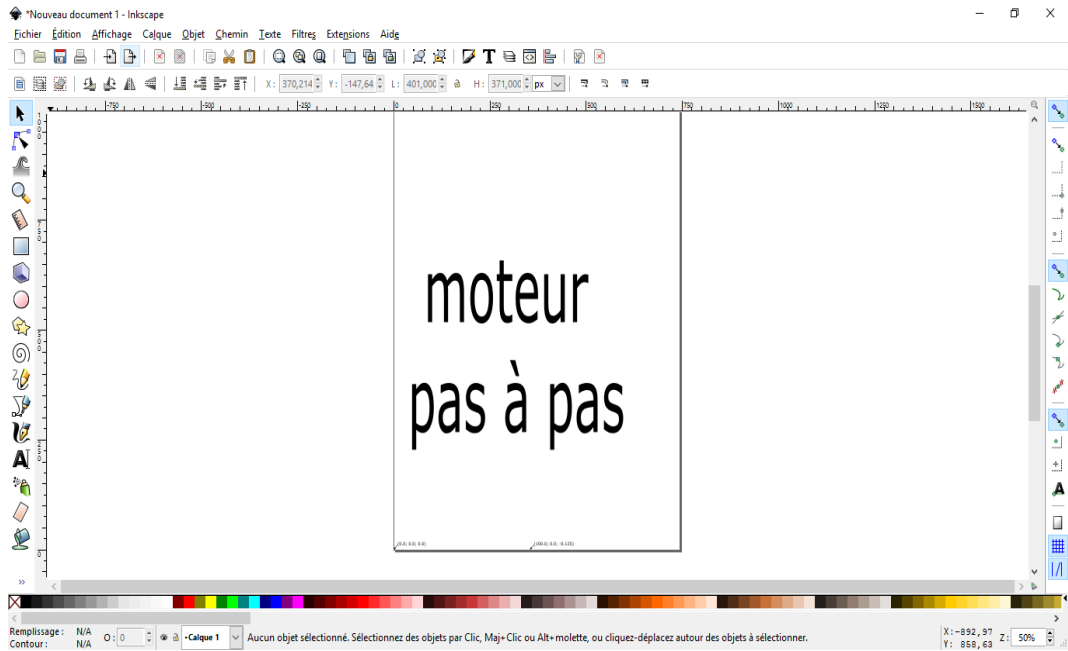


Figure.III.1 l'interface principale de Inkscape

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

III.4. Logiciel de pilotage de la machine CNC

La solution la plus fonctionnelle que j'ai trouvée est une évolution de Universal Gcode Sender (développé initialement par l'auteur de GRBL), qui est cours de développement par Winder[15].

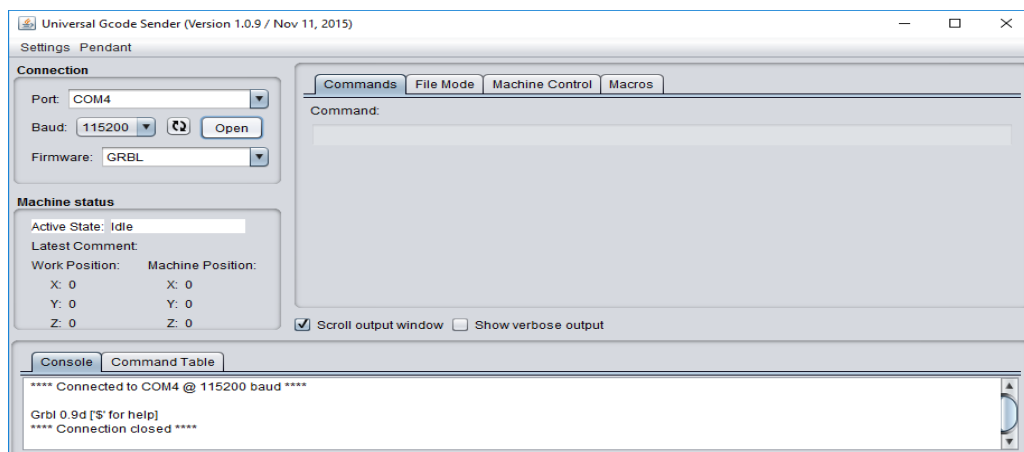


Figure III.2: de Universal Gcode Sender.

III.4.1. modes de commande

Il y'a deux mode utilise 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.

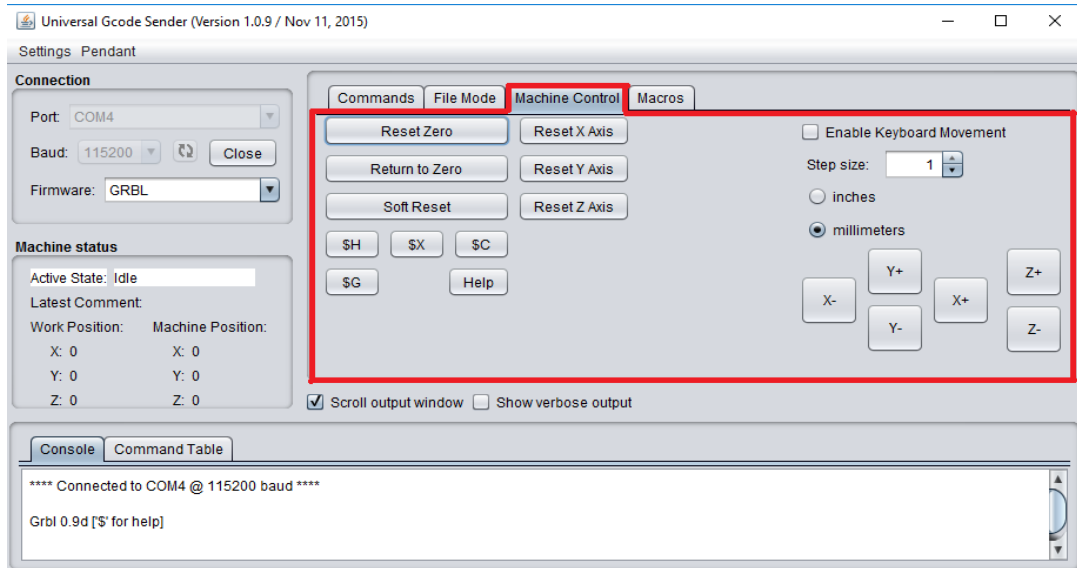


Figure III.3: l'interface de “g code sender” on mode manuel.

Pour l'utilisation graphique : le mode automatique, Charge notre fichier G-code: dans l'onglet "file mode" faites "browse" et sélectionnez fichier G-code (.ngc) 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 ($Z > 0$).

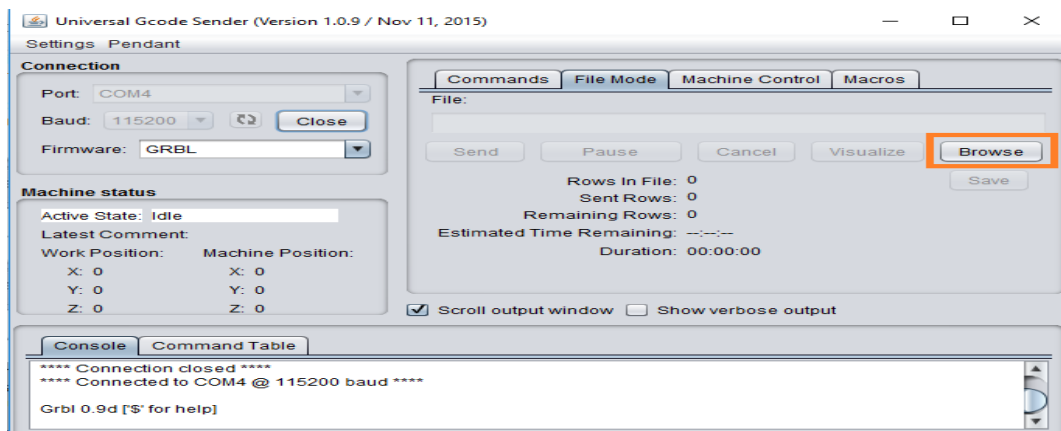


Figure III.4: l'interface de “Gcode sender” choisir le fichier Gcode.

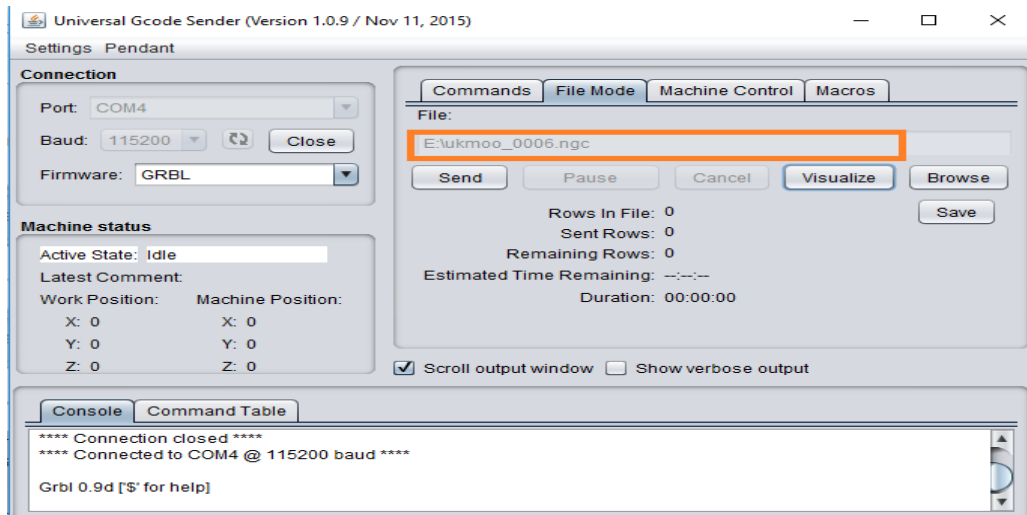


Figure III.5: l'interface de "g code sender" on mode automatique.

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".

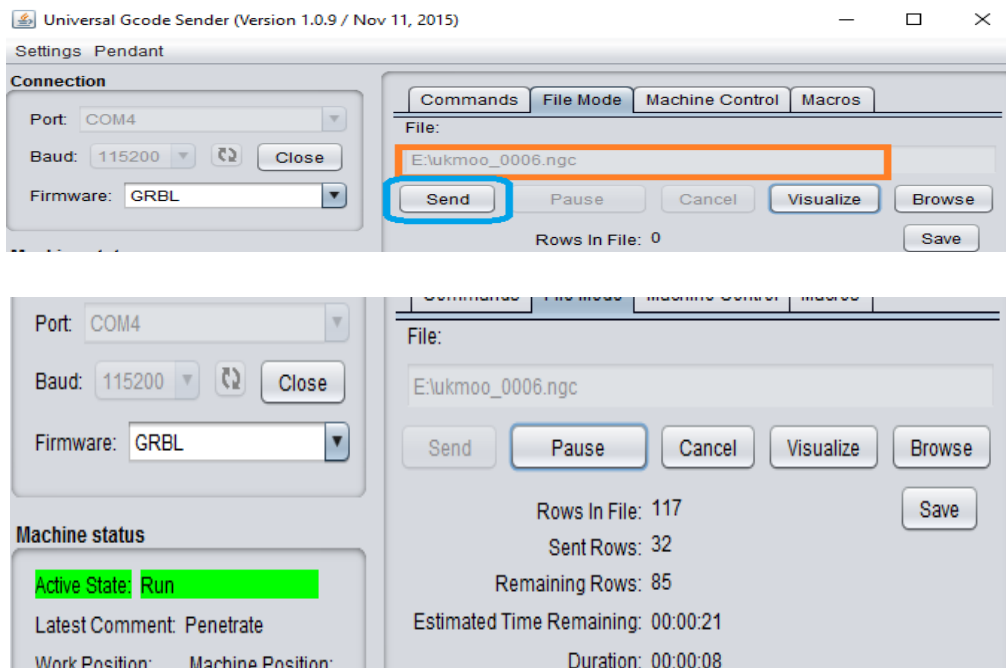


Figure III.6: envoyé fichée G code

III.4.2. 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.

III.4.3. 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 , si En cliquant sur ses boutons nous générons un instruction de code G qui envoyé à la carte de commande pour commander le moteur connecté avec ce bouton.

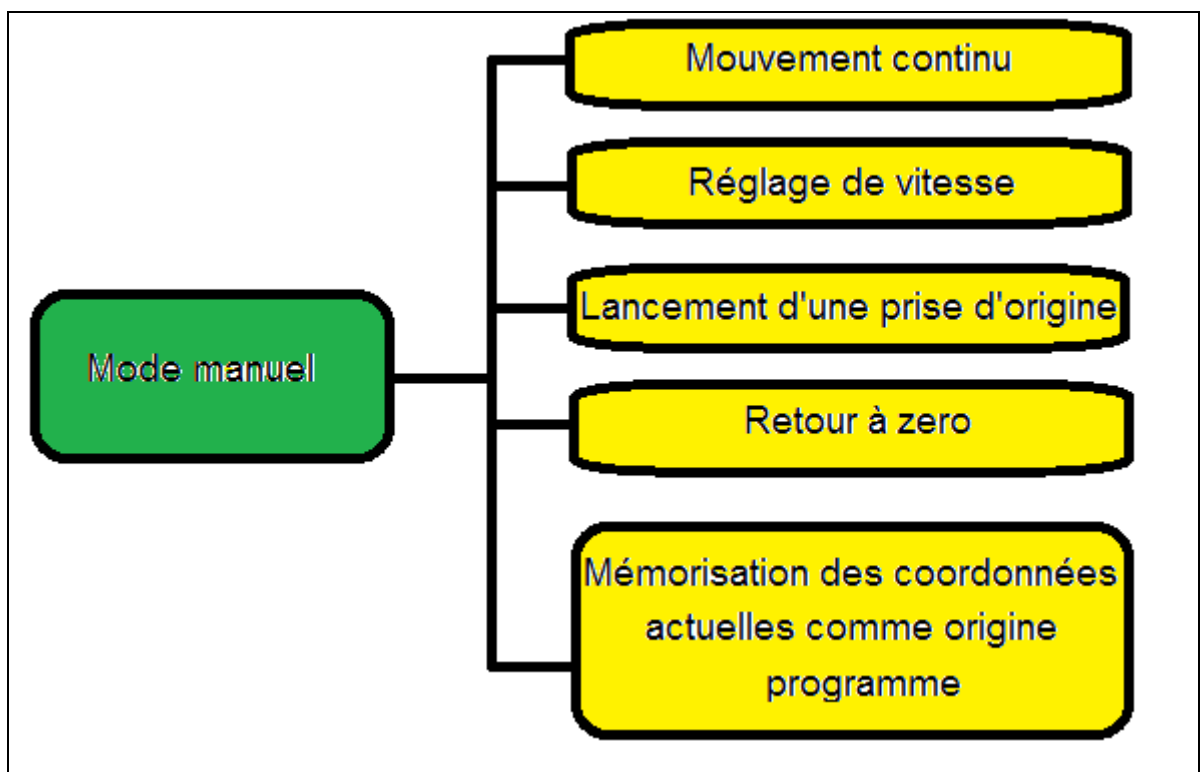


Figure III.7: de mode manuel.

- ✚ Mouvement continue : 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 de la 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.

III.4.4. 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. Commencant 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...et s'il s'agit d'un déplacement, l'arduino fait appel à la fonction qui contient l'interpolation de Bresenham [18] liée à ce déplacement (interpolation linéaire et interpolation circulaire). Enfin, le arduino envoie le mot « » pour indiquer au PC qu'il est prêt à recevoir l'instruction suivante. L'appui sur le bouton 'cancel' termine le dialogue.

III.5. 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.

III.5.1. Programme Grbl

Grbl est un micrologiciel 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 micrologiciel[16]:

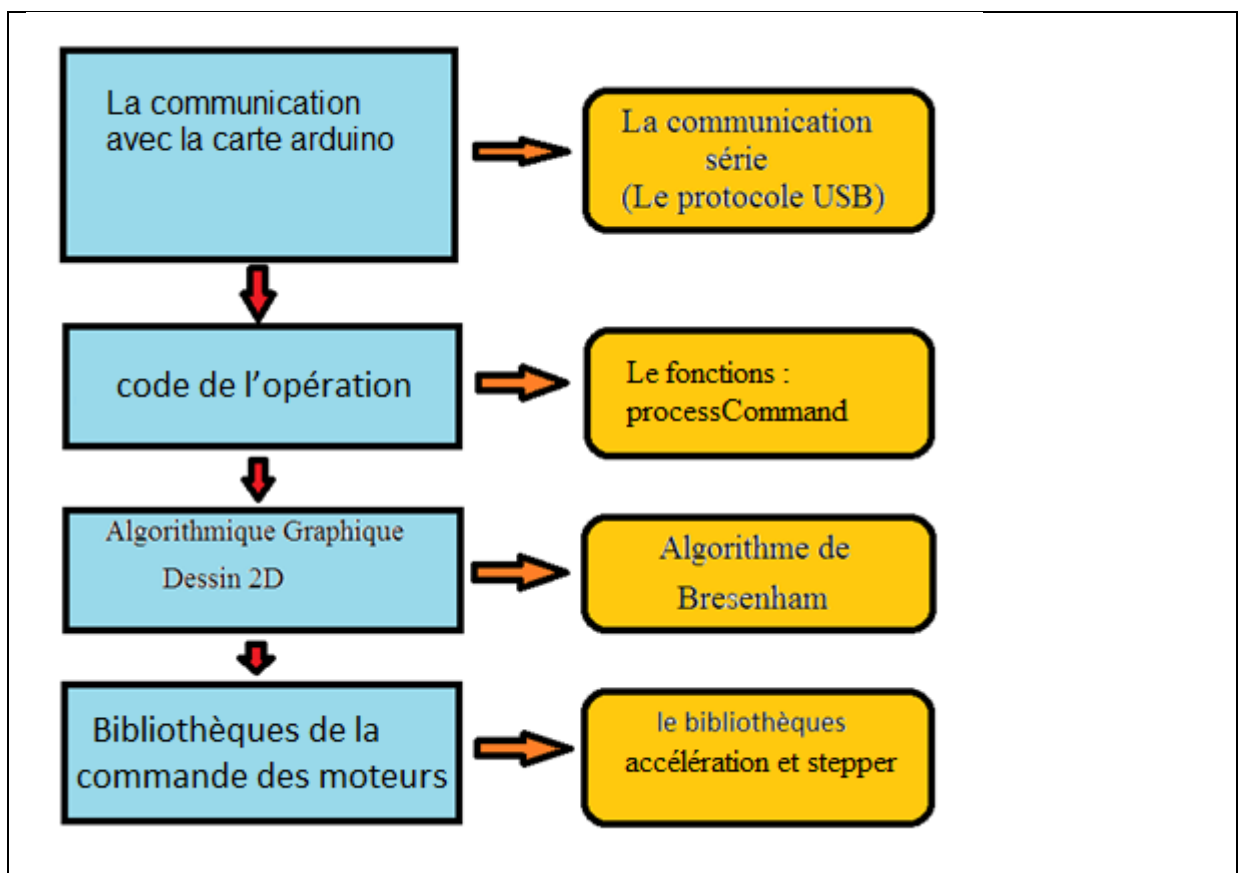


Figure III.8: de GRBL code.

III.5.1.1. 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[17].

III.5.1.2 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, Après la compilation nous obtenons l'opcode (code de l'opération) et évidemment 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[16] :

```
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;
}
}
```

III.5.1.2. L'interpolation de Bresenham :

L'interpolation de Bresenham[19] est un algorithme de base pour beaucoup de traitements de l'Informatique Graphique: dessin en fil de fer, remplissage, élimination des parties cachées... Nous désirons tracer un segment entre deux points (X_i, Y_i) et (X_f, Y_f) . Ce tracé est effectué sur un écran bitmap (composé d'une matrice de $n \times m$ pixels carrés) qui représente notre plateforme commandée par les deux moteurs pas à pas. Donc, le segment doit être discrétisé de la manière suivante (**Figure III.9** et 10):

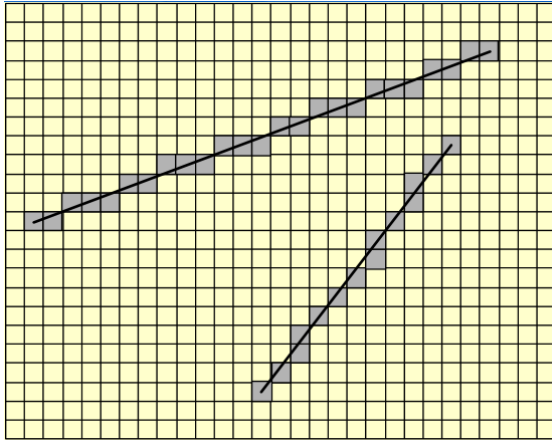


Figure III.9 : Bons tracés

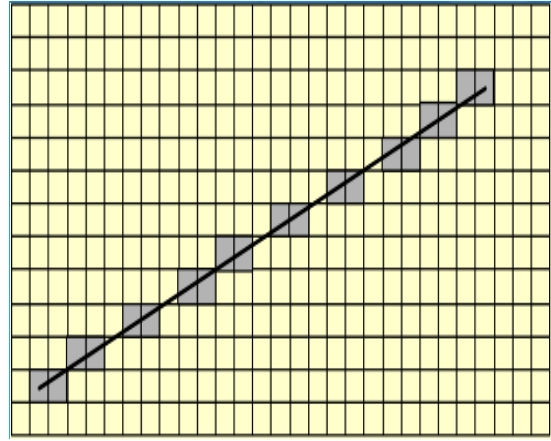


Figure III.10 : Mauvais tracés

Notre but est de tracer le moins de pixels possibles. Pour cela nous traçons un nombre de pixels égal à :

$$N=1+\max (\text{abs} (X_i-X_f), \text{abs} (Y_i-Y_f))$$

Si $\text{abs}(X_i-X_f) < \text{abs}(Y_i-Y_f)$, nous traçons un pixel et un seul par ligne interceptant le segment, sinon nous traçons un pixel et un seul par colonne interceptant le segment (figure III.9).

Nous posons comme hypothèse simplificatrice:

$$\begin{aligned} X_f &> X_i \\ Y_f &> Y_i \\ \text{et } (X_f-X_i) &\geq (Y_f-Y_i) \end{aligned}$$

Tous les autres cas peuvent s'y rapporter. Le coefficient directeur de la droite passant par les deux sommets il est positif et inférieur ou égal à 1. Au cours de l'exécution, nous quantifions, pour chaque pixel allumé, la différence entre ses coordonnées entières (utilisées pour l'affichage raster) et réelles telles qu'elles sont calculées sur le segment continu. Nous allumons un pixel en chaque abscisse entière x de X_i à X_f et nous utilisons une variable cumul pour stocker une estimation de la différence entre l'ordonnée entière et l'ordonnée réelle. Le dernier pixel allumé a pour coordonnées (X_f, Y_f) car $\frac{dx}{2}+dy \cdot dx$ est la valeur totale de ce qui a été accumulé dans cumul au cours de l'exécution (Figure III.11) :

$$\frac{\frac{dx}{2}+dy \cdot dx}{dx} = dy \text{ est le nombre de fois où } y \text{ a été incrémenté de } 1.$$

- L'ordonnée du dernier pixel allumé est $Y_i + dy = Y_f$.

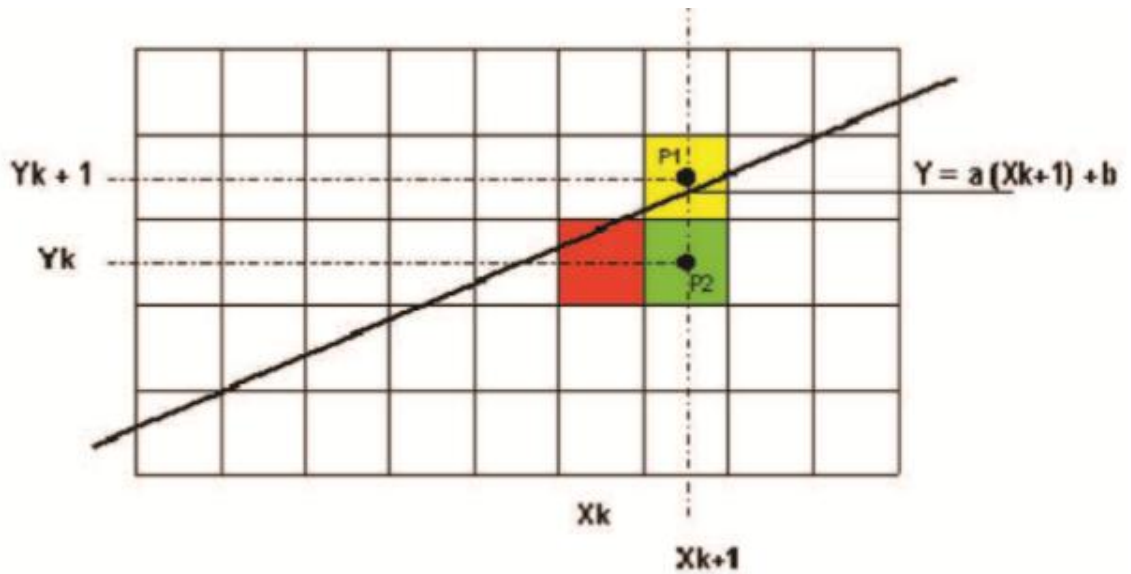


Figure III.11: Interpolation linéaire de Bresenham

Caractéristiques :

- Plus grande complexité algorithmique.
- Rapidité due à l'utilisation exclusive d'entiers courts (valeurs maximales de l'ordre de la résolution de l'écran -> petites valeurs) et d'opérations arithmétiques simples sur ces entiers (additions, soustractions et comparaisons).

III.5.1.4. Algorithme adapté pour le tracé de tout segment

Utilisation de deux variables X_{inc} et Y_{inc} pour gérer des incréments de 1 ou -1 pour x variant de X_i à X_f et y variant de Y_i à Y_f (**Figure III.12**).

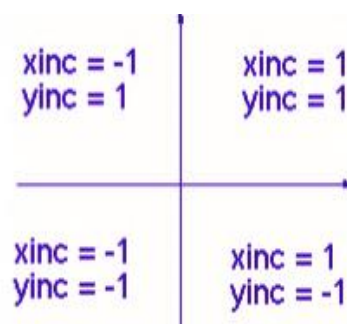


Figure III.12: Adaptation de l'interpolation de Bresenham .

Deux parties alternatives dans l'algorithme:

- ✚ Une pour l'incrément en x

✚ L'autre pour l'incréméntation en y, en fonction de la pente du segment.

```

procédure Tracer_Ligne                                'procédure d'interpolation Linéaire
  erreur=0
  Si px>maxx alors erreur=1                            'contrôle dépassement course maxi
    sinon si py>maxy alors erreur=2
  sinon si px<0 alors erreur=3    'contrôle de dépassement course mini
sinon si py<0 then fail=4 end if
  fin si
  fin si
  fin si
  dx = px - ax dy = py - ay 'déterminer le mouvement des axes
  Cx = ax Cy = ay
  Tdx = dx * 2    Tdy = dy * 2
  Xinc = 1    Yinc = 1
  mx=0    my=0
  si dx < 0 alors Sens_M1 = 0 dirx = 0 Xinc = -1 Tdx = - Tdx dx = -dx 'déterminer la 'direction
du mouvement
    sinon Sens_M1 = 1 dirx = 1 fin si
  si dy < 0 alors Sens_M2 = 0 diry = 0 Yinc = -1 Tdy = - Tdy dy = -dy
    sinon Sens_M2 = 1 diry = 1
  fin si
  si (erreur=0) et ((Dx<>0) ou (Dy<>0)) alors
    si dy <= dx alors          ErrX = 0
    Tant que (Cx <> px) et (Arret_Urg=1) et (pause=0) et (butee_X=1) et
(butee_Y=1)
      Cx = Cx + Xinc
      mx = mx + 1
      Incrémente(M1)
      ErrX = ErrX + Tdy
      Si ErrX > dx alors
        Cy = Cy + Yinc
        my = my + 1
        Incrémente(M2)
        ErrX = ErrX - Tdx
      Fin si
    Fin tant que

```

```

Sinon

    ErrY = 0
    tant que (Cy <> py)
        Cy = Cy + Yinc
        my = my + 1
        Incrémenter(M2)
        ErrY = ErrY + Tdx
        si ErrY > dy alors
            Cx = Cx + Xinc

            mx = mx + 1
            Incrémenter(M1)
            ErrY = ErrY - Tdy
    Fin si
    Fin tant que
    Fin si
    si dirx = 1 alors ax = ax + mx 'mise à jour des compteurs
        sinon ax = ax - mx fin si
    si diry = 1 alors ay = ay + my
        sinon ay = ay - my fin si
fin procédure 'fin procédure d'interpolation Linéaire.

```

Algorithme : Procédure d'interpolation linéaire adoptée après modification.

III.5.1.5. Bibliothèques de la commande des moteurs

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

III .6. Organigramme de la commande de la machine CNC

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

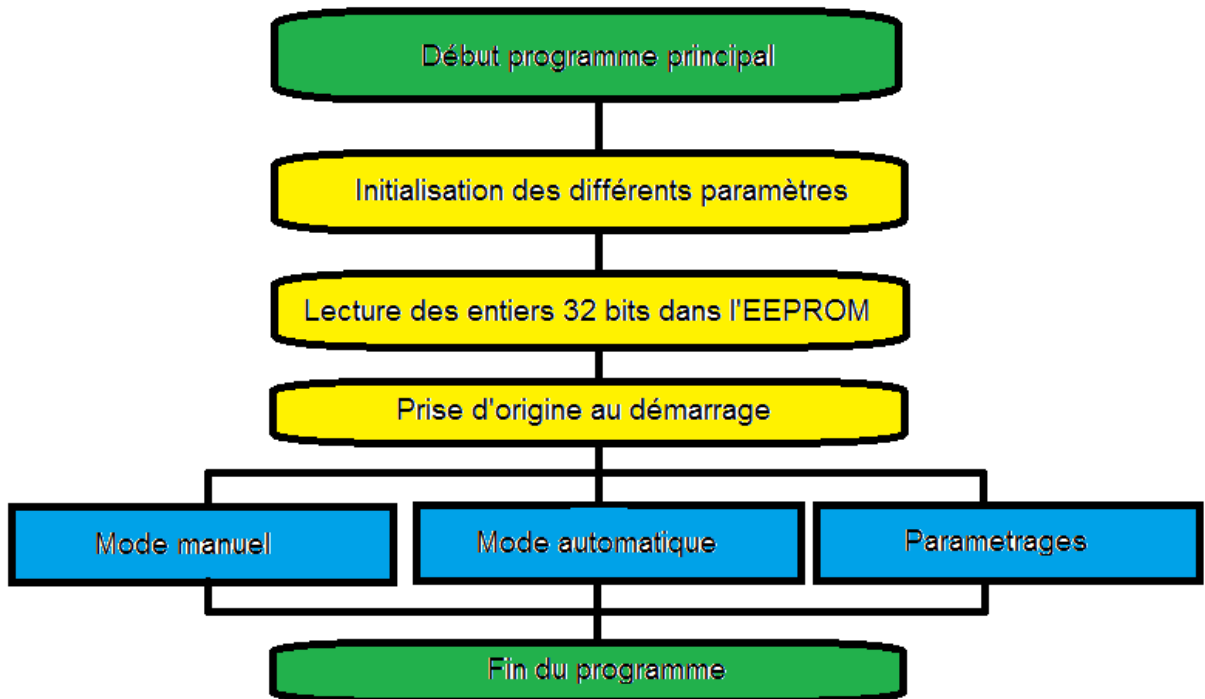


Figure III.13 : Organigramme de la commande de la machine CNC.

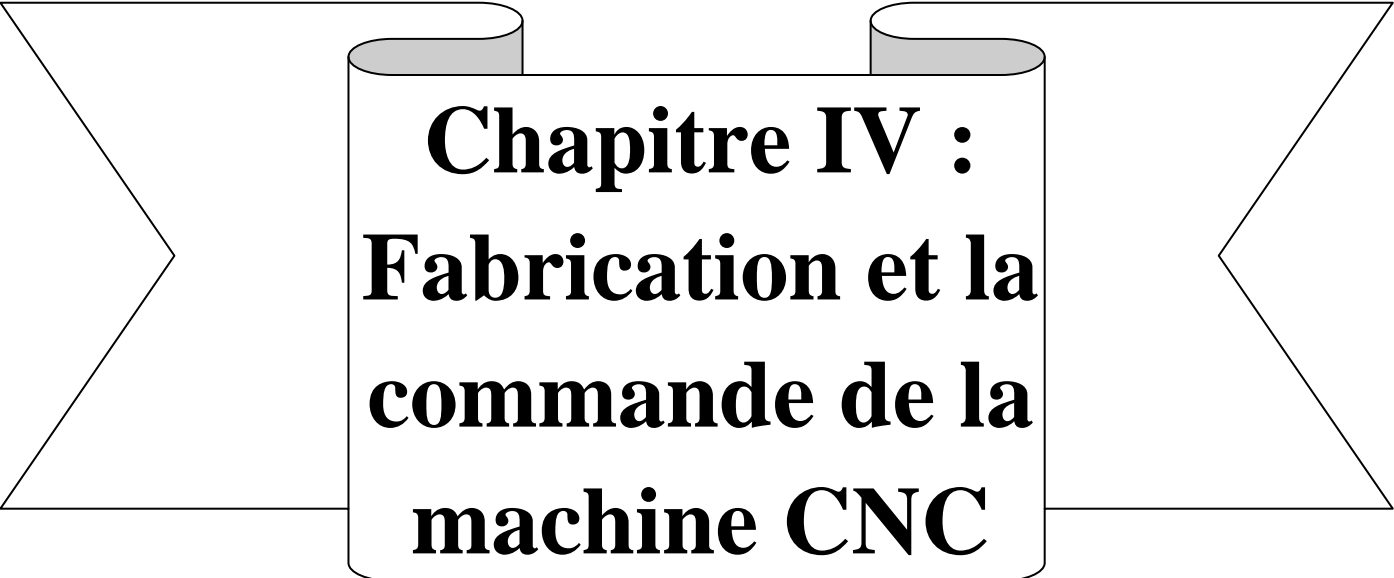
Une fois la machine CNC fonctionne, il existe trois différents modes majeurs pour entrer des commandes. Les modes Manuel, Auto et Paramétrage. Passer d'un mode à une autre marque une grande différence dans le comportement de la machine CNC. Des choses spécifiques à un mode ne peuvent pas être faites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

III.7. Conclusion

Ce chapitre est 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 Gcode Sender pour transfert fichier Gcode vert la carte Arduino Uno. Nous avons constaté que l'algorithme de BRESSENHAM que nous avons présenté est suffisamment précis pour tracer une trajectoire.



**Chapitre IV :
Fabrication et la
commande de la
machine CNC**

IV.1. Introduction

Ce chapitre est consacré à la fabrication et l'application 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 développés théoriquement. Implémentation des circuits simulés sur la plaque à essais (plaque à trous) afin de s'assurer de son fonctionnement.
- ✚ Une fois la simulation est bien vérifiée, on passe à l'étape de routage en utilisant l'outil ARES. Réalisation de la carte électronique en imprimant les circuits 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 traceur.

IV.2. Simulation électrique sous Porteuse Professionnel

Proteus est une suite logicielle destinée à l'électronique. Développé par la société Labcenter Electronics, les logiciels inclus dans Proteus permettent la CAO dans le domaine électronique. Deux logiciels principaux composent cette suite logicielle : ISIS, ARES.

IV.2.1. Présentation générale

Cette suite logicielle est très connue dans le domaine de l'électronique. De nombreuses entreprises et organismes de formation (incluant lycée et université) utilisent cette suite logicielle. Outre la popularité de l'outil, Proteus possède d'autres avantages.

- Pack contenant des logiciels facile et rapide à comprendre et utiliser
- Le support technique est performant
- L'outil de création de prototype virtuel permet de réduire les coûts matériel et logiciel lors de la conception d'un projet.

IV.2.2. ISIS

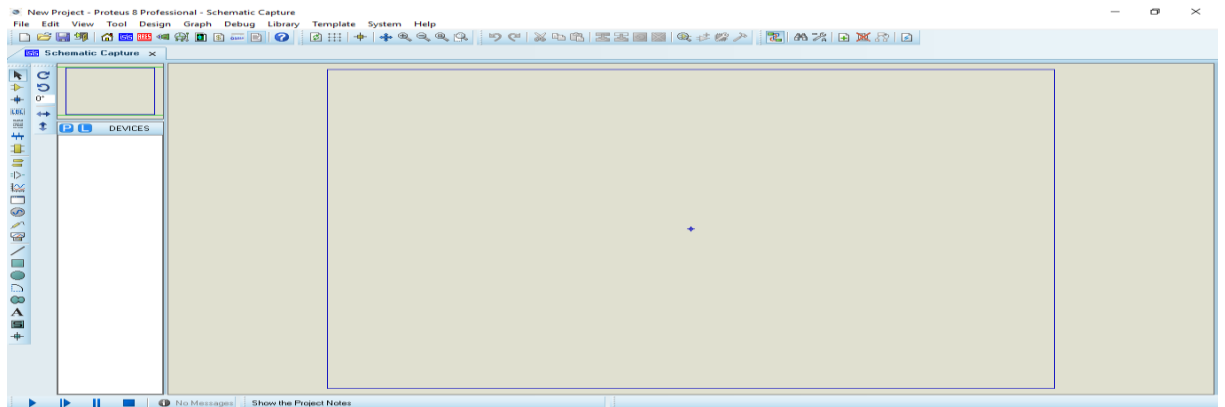


Figure IV.1: Logiciel ISIS.

Le logiciel ISIS de Proteus est principalement connu pour éditer des schémas électriques. Par ailleurs, le logiciel permet également de simuler ces schémas ce qui permet de déceler certaines erreurs dès l'étape de conception. Indirectement, les circuits électriques conçus grâce à ce logiciel peuvent être utilisés dans des documentations car le logiciel permet de contrôler la majorité de l'aspect graphique des circuits.

IV.2.3. ARES

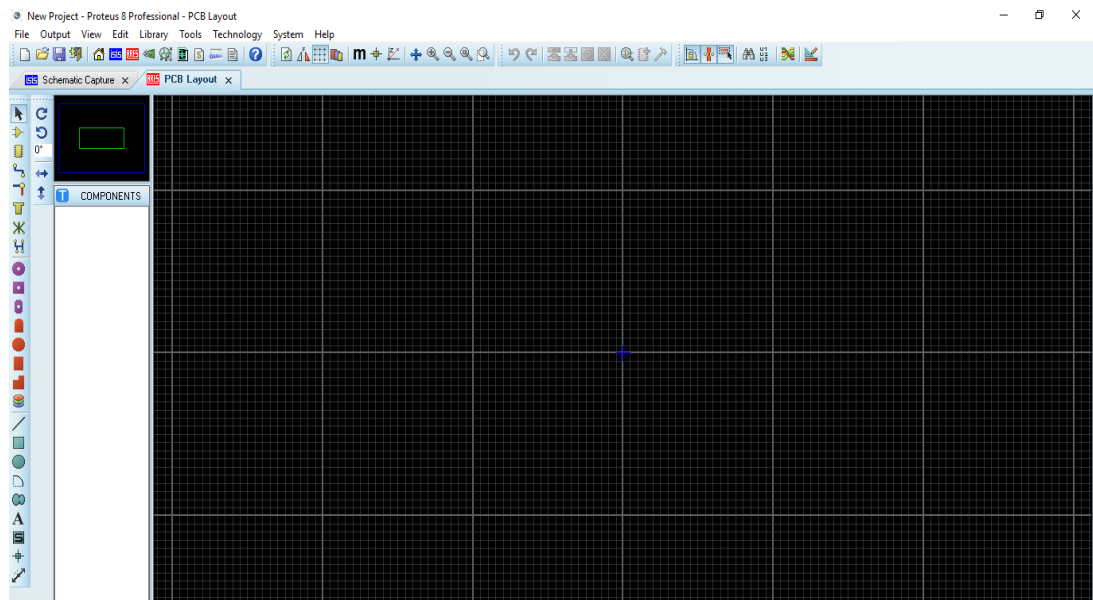


Figure IV.2: Logiciel ARES.

Le logiciel ARES est un outil d'édition et de routage qui complètement parfaitement ISIS. Un schéma électrique réalisé sur ISIS peut alors être importé facilement sur ARES pour

réaliser le PCB de la carte électronique. Bien que l'édition d'un circuit imprimé soit plus efficace lorsqu'elle est réalisée manuellement, ce logiciel permet de placer automatiquement les composants et de réaliser le routage automatiquement.

Isis est orienté vers les électroniciens débutants, ainsi que les développeurs et les professionnels.

IV.2.4. Simulation et test le programme

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

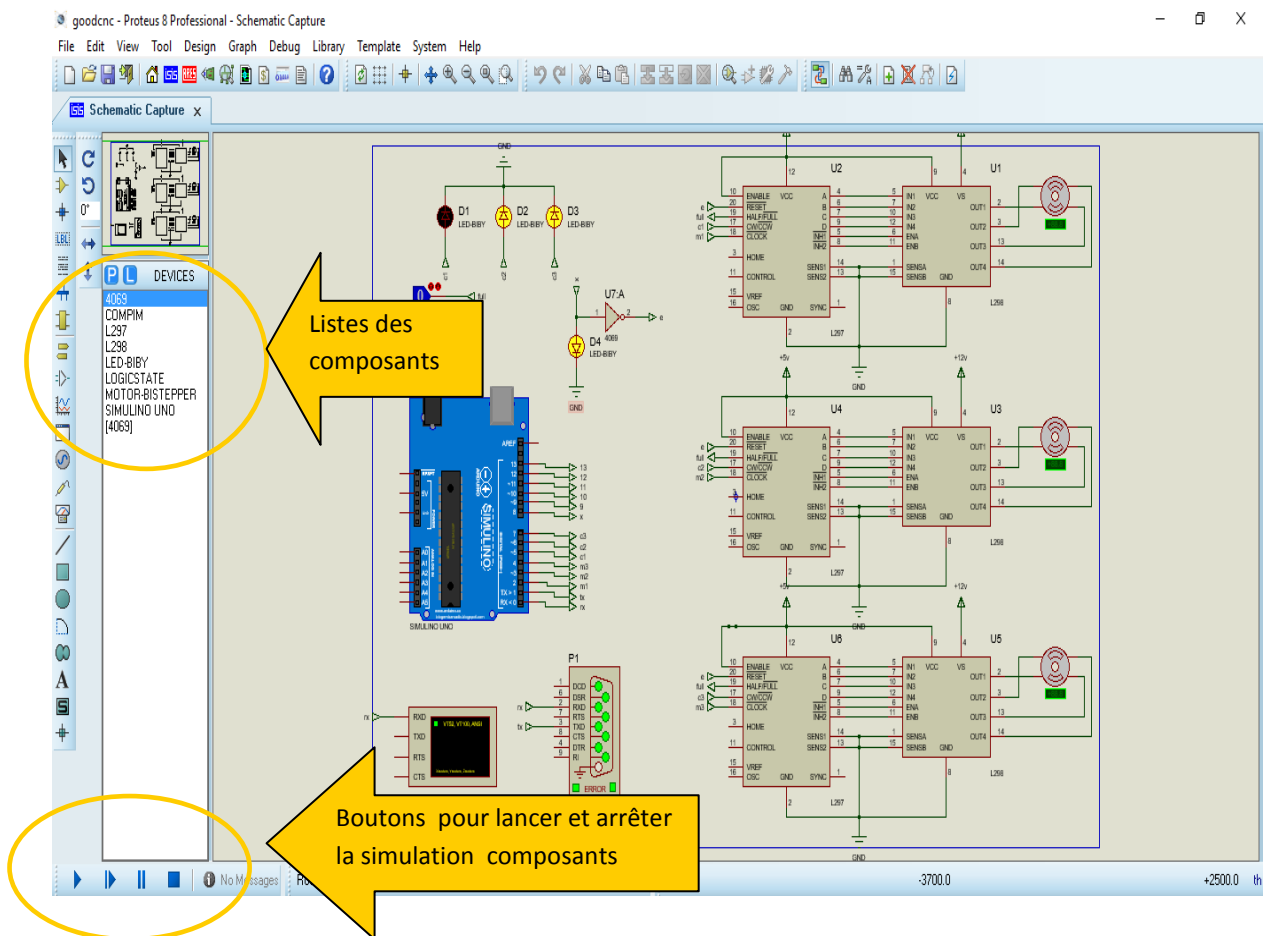


Figure IV.3 : Montage électronique dans Proteus.

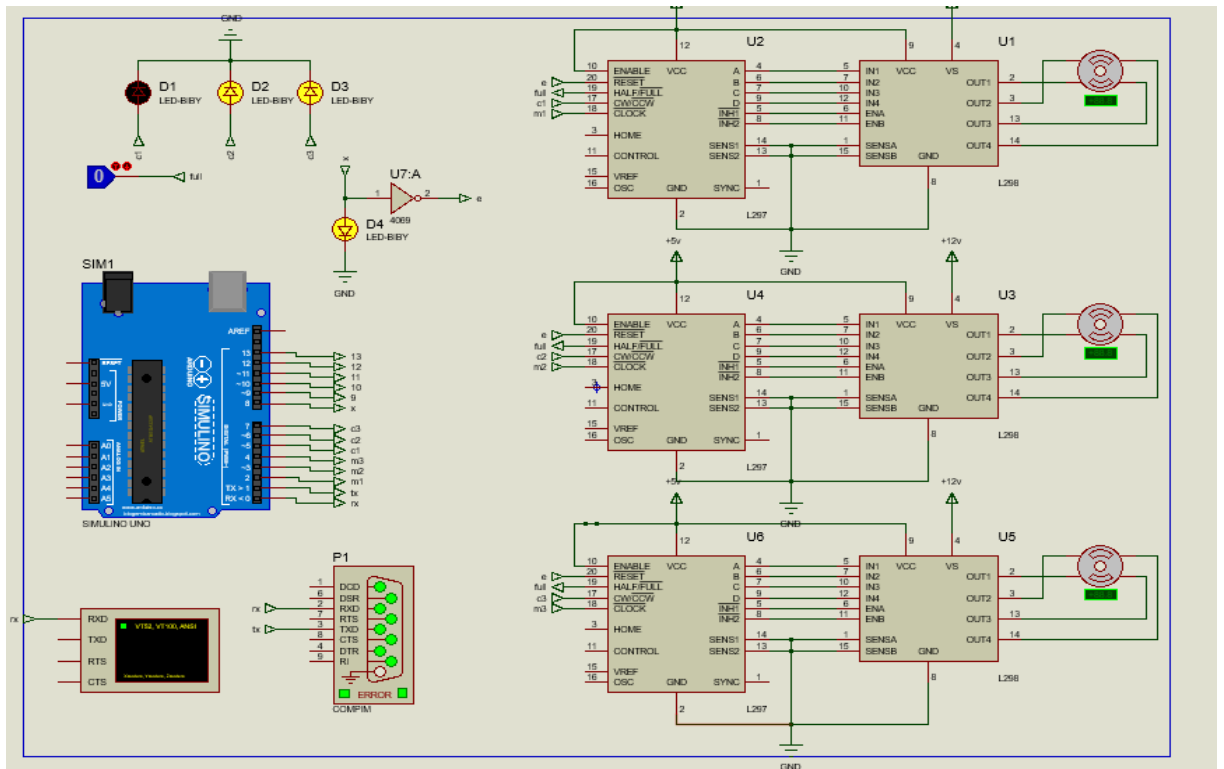


Figure IV.4 : simulation cnc machine.

IV.2.4.1. Associer le programme au processeur

- Dans le menu « source » sélectionnez la commande « Ajout/suppression fichiers source » puis sélectionnez le fichier ASM du programme à utiliser dans la simulation.
- Puis choisissez « Outil de génération de code » pour générer un fichier HEX
- Editez le arduino et ajoutez le fichier HEX dans le champs « programme file »
- Afficher les fenêtres « Registers » « Source Code » « Data memory » par l'intermédiaire du menu « Debug » pour contrôler le fonctionnement du programme, détecter et corriger les erreurs.

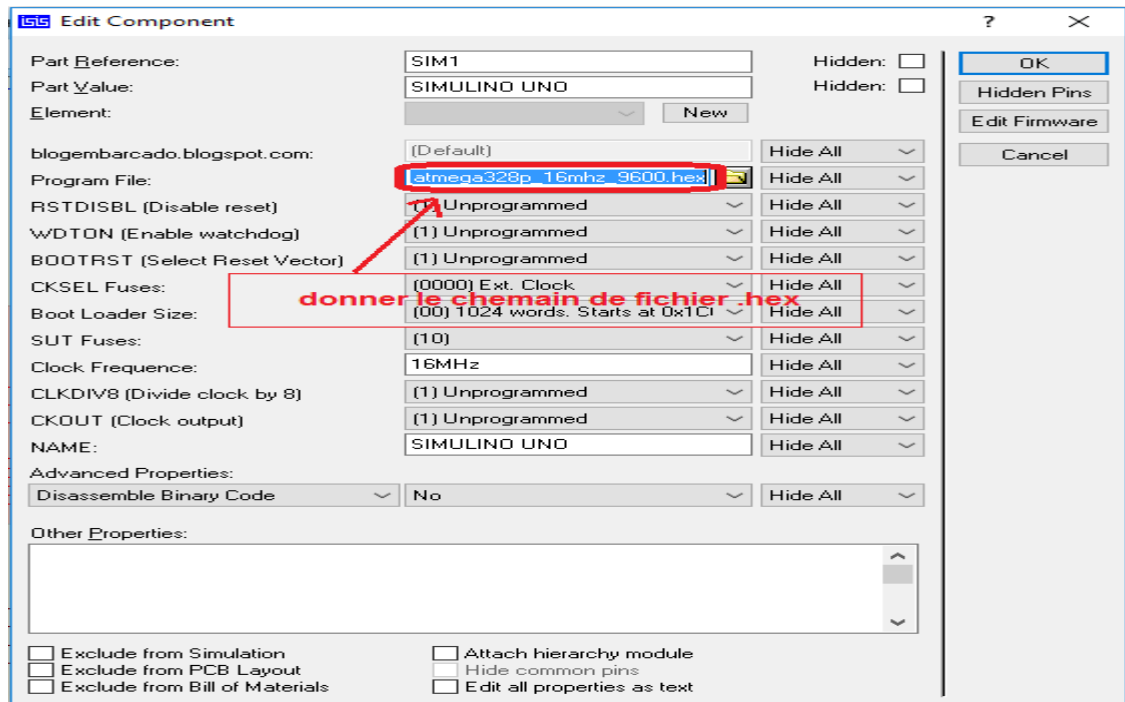


Figure IV.5: charge le fichier hex de programme.

IV.3. fabrication de cnc machine 2D

Etape 1 : matériel nécessaire pour réaliser le machine CNC

Nous essayons d'employer des matériaux que l'on peut se procurer facilement en magasin de bricolage et des composants les moins chers possibles pour la fabrication de la machine CNC. Pour piloter les moteurs pas à pas des axes X , Yet Z, nous avons deux possibilités. Soit utiliser trois couples « L297/L298 » (un pour chaque axe), ou on utilise trois carte Easy Driver pour Arduino.

- 1x Arduino Uno
- 1x Breadboard
- 3x easy driver
- 3x lecteurs de DVD
- 1x plaque de contreplaque pour la base du Mini Traceur (axe X)
- Des rondelles
- 18x vis
- 12x écrou
- 20x fils

Etape 02 : démonté le lecteur de DVD

Nous avons retirés tous les éléments électroniques de la tête de lecture. Il faut obtenir une surface plane pour pouvoir coller correctement le plateau de dessin et le porte de crayon. Nous avons démonté le lecteur de DVD pour ne conserver que le mécanisme de déplacement de la tête de lecture. Les systèmes obtenu est système de déplacement linéaire récupéré sur un lecteur de DVD qui présente un porteur d'un axe de cnc machine.

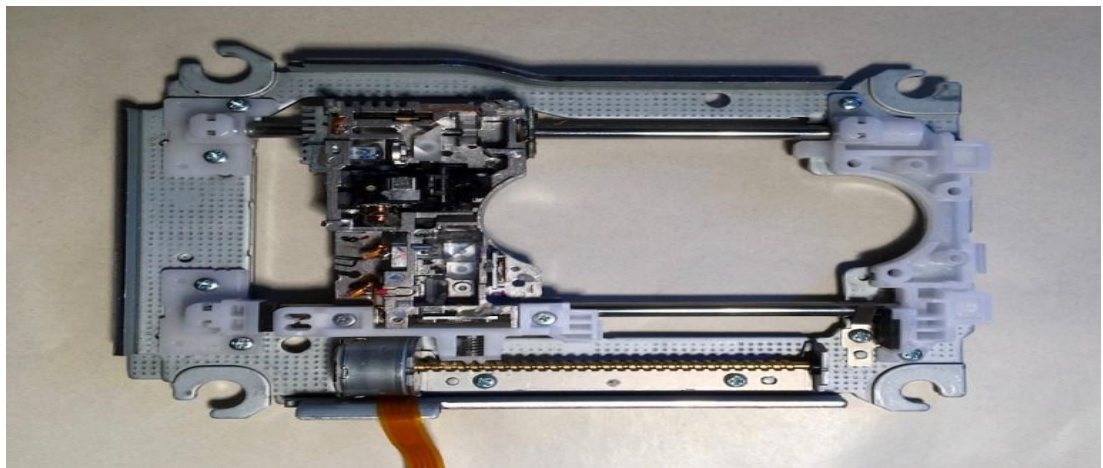


Figure IV.6 : Système de déplacement linéaire récupéré sur un lecteur de DVD.

Etape 3 : Préparer les moteurs pas à pas des axes X, Y et Z

Dans cette étape, nous allons réaliser les soudures sur les moteurs pas à pas des axes X, Y et Z Souder directement les fils au niveau du moteur. Sur certains moteurs comme celui-ci c'est assez facile car les 4 pistes sont bien distinctes.



Figure IV.7: Souder les 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 de bricolage. La fabrication est très simple. Il n'y a que 3 plaques de contreplaqué à découper. Vous devrez adapter le perçage pour la fixation des axes en fonction de vos lecteurs de DVD.

Etape 5 : Branchement du circuit

Le câblage du cnc machin est assez simple. Il est composé de 3 Easy Driver. Une plaque d'essai.

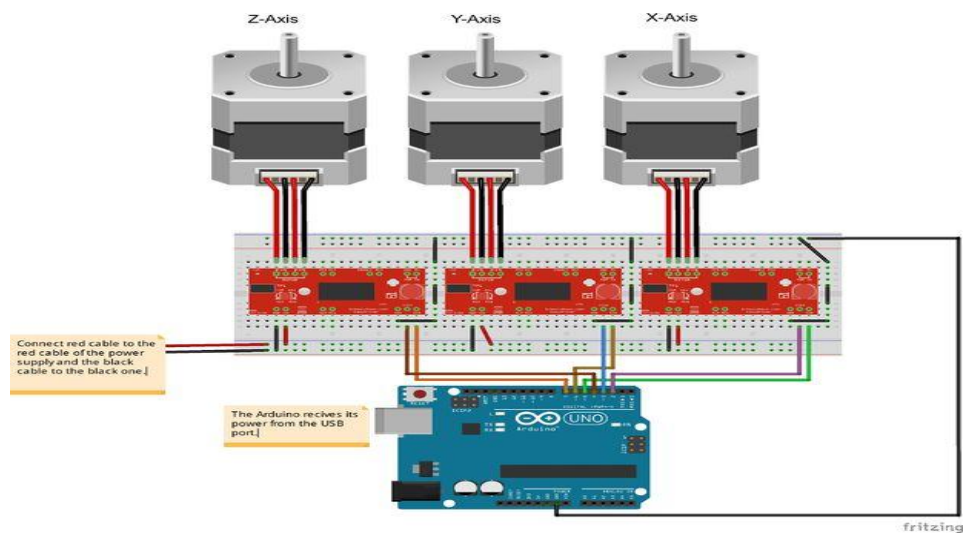


Figure IV.8: montage cnc machin.

Etape 6: Assemblage du cnc machin Axes X et Y et Z

Commençons par fixer l'axe X à l'aide des 12 écrous et de 4 vis et 4 rondelles. Fixons l'axe Y sur le montant verticale l'aide des 12 écrous et de 4 vis et 8 rondelles. La hauteur n'est pas 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 avec un vis, pour Fixer l'axe z à cette plaque à l'aide des 12 écrous et de 4 vis et 4 rondelles

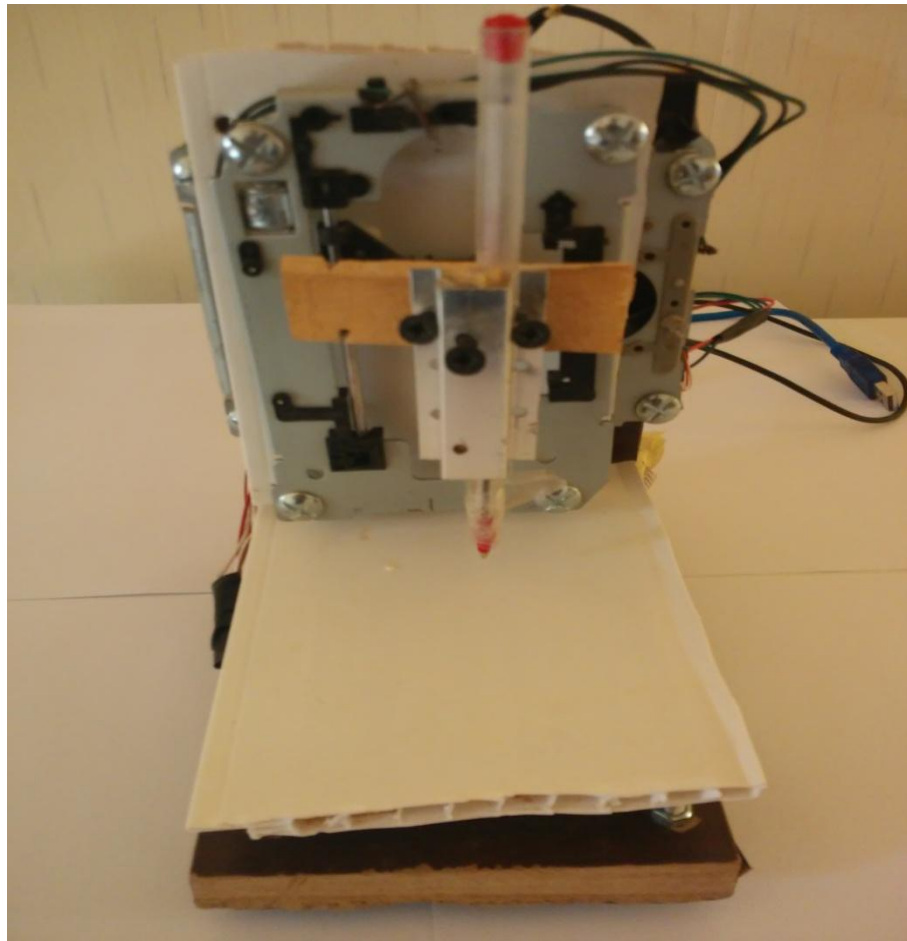


Figure IV.9: cnc machin

Porte crayon (axe Z)

Fixons un support de crayon avec 2 vis dans l'axe Z et 3eme vis pour le système de serrage plaque d'essai et Arduino. Ensuite fixons la plaque d'essai sur le montant à l'aide de deux vis. L'Arduino prend place sous la plaque d'essai à l'arrière de la machine CNC.

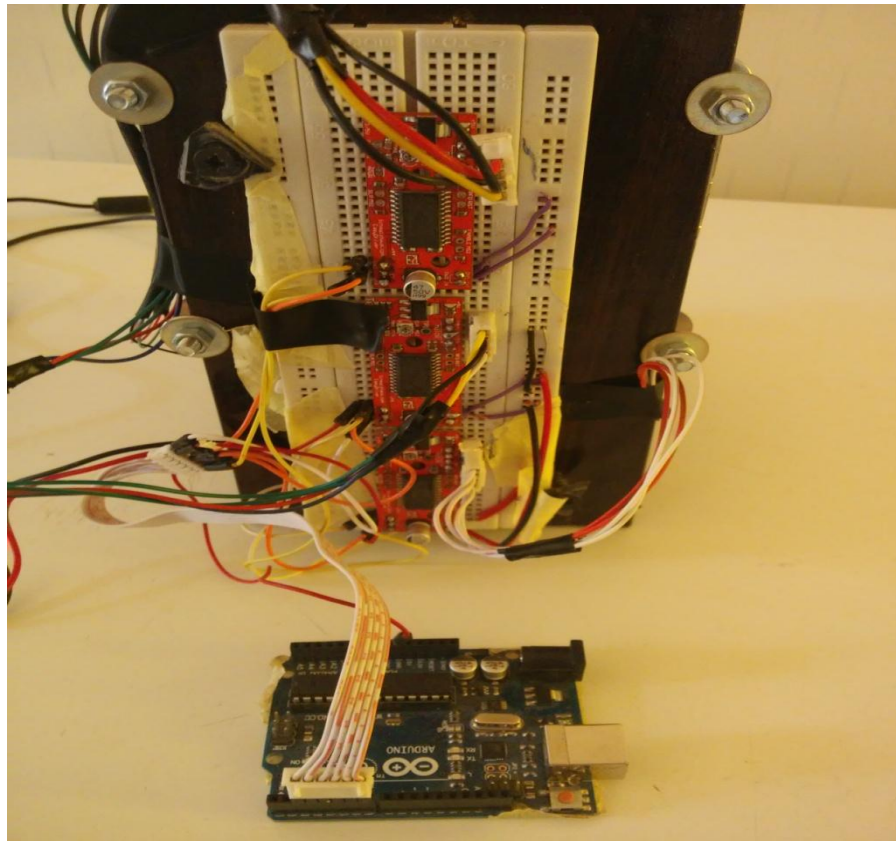


Figure IV.10: arduino et Breadboard de cnc machin.

Etape 7 : Test des axes X / Y/Z

Le câblage et l'assemblage réalisé, nous allons maintenant pouvoir tester le bon fonctionnement de mouvements des axes X et Y et Z

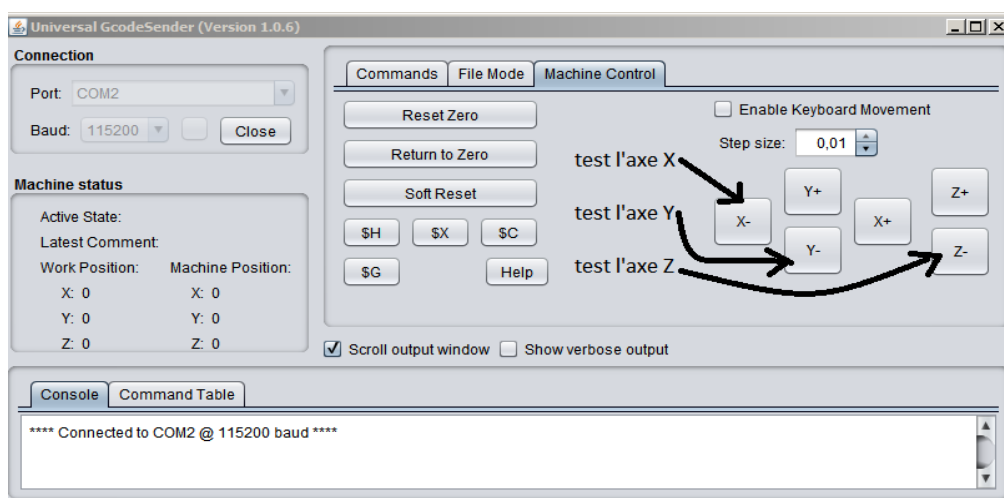


Figure IV.11: test les axes X ,Y et Z.

Etape 8 : Préparez votre ordinateur à envoyer des fichiers G code

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

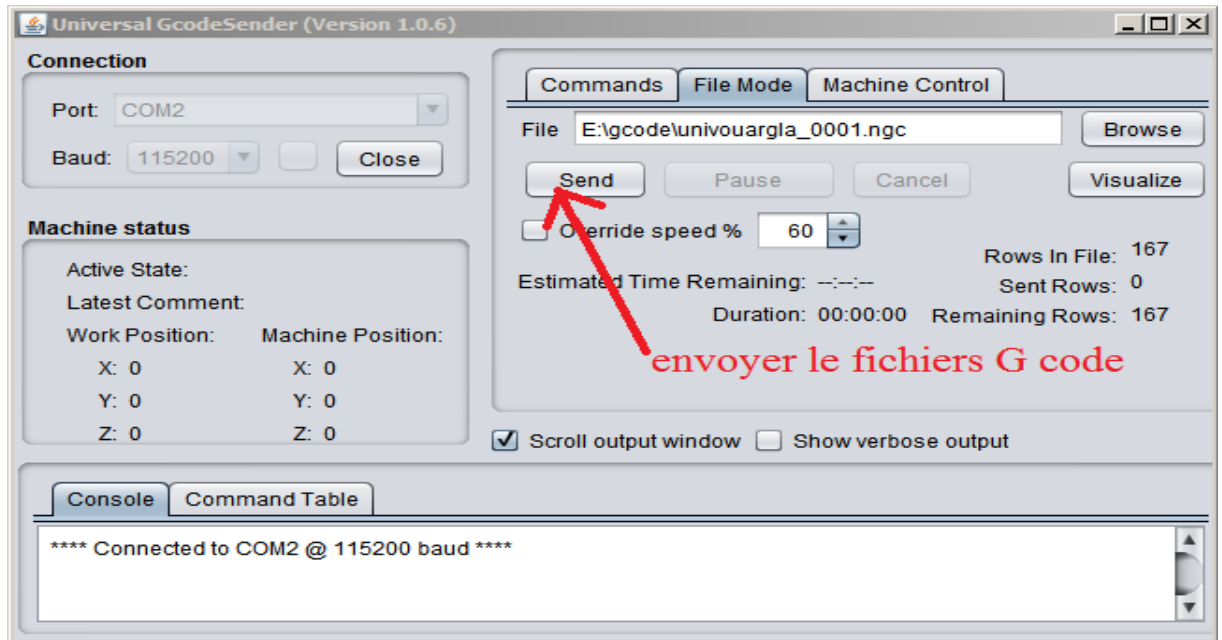


Figure IV.12: universel G code sender

Etape 9 : Traçons notre premier dessin

Avant de faire votre premier tracer :

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



Figure IV.13: exemple dessin cnc machine.

IV.3. Conclusion :

Comme nous venons de le voir dans ce chapitre, beaucoup de travaux ont été réalisés concernant la machine CNC, nous avons présenté plusieurs types de logiciels utilisés pour simulation et la commande de machine CNC. Pour réaliser cette commande on ait utilisé une carte microcontrôleur "Arduino Uno" qui possède un espace de programmation qu'est très claire et simple et les circuit intègres L297, L298 et Easy Driver. Cette carte sert à contrôler les moteurs pas à pas bipolaire. A l'aide du logiciel "Inkscape" nous avons stoker les codes G dans la carte arduino Uno. Pour finir, nous branchons les moteurs pas à pas, et nous testons la carte de commande des moteurs pas à pas. La machine CNC fonctionne correctement et répond aux différentes commandes.

Conclusion générale

L'élaboration de ce travail dans le cadre du projet de la fin d'étude, nous a permis d'approfondir nos connaissances théoriques en électronique et d'acquérir une bonne expérience au niveau de la réalisation pratique.

Lors de cette manipulation, on a essayé de fournir l'automatisation de commande de trois moteurs pas à pas et applique à la machine CNC. Notre projet de fin d'étude consiste à détailler ainsi qu'à réaliser une carte électronique de commande de trois moteurs pas à pas. Ce projet m'a donné une meilleure idée sur la complémentarité entre le volet théorique et le volet pratique.

Dans la première partie, nous avons d'abord entamé la première étape qui est basée sur la présentation de la machine CNC et la deuxième étape consiste sur les moteurs pas à pas que nous avons étudié en détail. Cette partie est essentielle au fonctionnement des machines à commande numérique.

Dans la deuxième partie, nous avons fait une description et la mise en œuvre de la carte Arduino Uno et les circuits intégrés (L297, L298 et Easy Driver), pour tester les différentes interfaces électroniques d'acquisition de données et de commandes des moteurs pas à pas.

Ensuite, la troisième partie, est consacrée à des logiciels de simulation pour cette raison, plusieurs étapes ont été envisagées :

- ✚ Nous avons validé la faisabilité et le bon fonctionnement des circuits déjà développées théoriquement. Implémentation des circuits simulés sur la plaque à essais (plaque à trous) afin de s'assurer de son fonctionnement.
- ✚ Une fois la simulation est bien vérifiée, nous avons passé à l'étape de routage en utilisant l'outil ARES pour la réalisation de la carte électronique en imprimant les circuits et la soudure des composants sur une carte principale (carte de commande) dont le rôle principal est de gérer les moteurs pas à pas.

Finalement, notre travail nous permis la réalisation pratique, et le test des différentes cartes assemblées, nous affirmons que nous avons achevé notre tâche demandée : la

Conclusion générale

commande d'un prototype d'une machine de dessin 2D. L'avantage de notre commande est qu'elle est à la norme de la commande numérique des machines à 3 axes. Toutefois, elle peut être facilement exploitée pour ajouter une commande de troisième pour l'imprimante 3D et même un quatrième axe. Nous avons rencontré plusieurs problèmes durant la conception de la commande et la réalisation pratique (protocole de communication, couple des moteurs et leurs vitesses...), mais grâce à ce que nous avons appris durant nos études académiques et l'assistance de nos encadrants, nous avons pu confronter ces difficultés

En fait, ce projet a été une source de découverte de plusieurs domaines d'études tel que l'informatique pour la programmation embarquée et le design des schémas électroniques, sans oublier le savoir-faire dans le domaine électronique qui consiste à réaliser pratiquement les circuits électroniques.

En perspective, plusieurs points sont à approfondir et ouvrent la voie à de nouveaux axes de travaux :

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

BIBLIOGRAPHIE

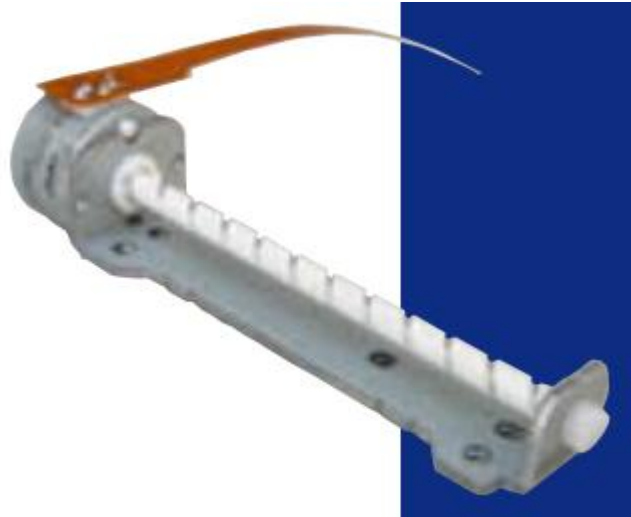
- [1] **Benbekhti Ahmed**, « *Etude de réalisation d'un support pour affutage des forets sur une machine à commande numérique* », mémoire de fin d'étude, Université Abou Bekr Belkaid-Tlemcen 2013
- [2] **Abdennadji Youssef, Ahmadi Anouar** « *Conception et réalisation d'une commande numérique d'une machine de découplaser* », Ingénieur d'école nationale de Gabés, juin 2010.
- [3] **Takashi Kenjo and Akira Sugawara**, « *stepping motors and their microprocessor controls* » Oxford University Press, ISBN 0-19-859385-6.
- [4] <http://www.positron-libre.com/electronique/moteur-pas-a-pas/sequence-commande-moteur-pas-a-pas.php>
- [5] **Christian Tavernier** « *Arduino Maitrise sa programmation et ses cartes d'interface (shield)* », paris 2011, DUNOD
- [6] <http://www.ELECTRONIQUEPRATIQUE.com>
- [7] <http://arduino.cc/en/Main/ArduinoBoardMega2560>
- [8] www.reality.be/elo/labos2/files/AtMega32DocFr.pdf
- [9] www.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-solution-pour-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] **John Kennedy** « *A Fast Bresenham Type Algorithm For Drawing Circles*

[20] **Sundar Pandian, S. Raj Pandian** « *a low-cost build-your-own three axis cnc mill prototype* » Velammal College of Engineering & Technology, 2014.

ANNEXE

Annexe -1- : Moteur pas à pas (lecteur disquette)



Moter Size	PL15S-020
Number of Steps per Rotation	20
Drive Method	2-2 PHASE
Drive Circuit	BIPOLAR CONST. VOLT.
Drive Voltage	5[V]
Coil Resistance/Phase	10[Ω]
Magnet Material	Nd-Fe-B bonded magnet (MS70)
Holding Torque	30[X10 ⁻⁴ N.m]
Maximum PULL-IN Frequency	1450[PPS]

Caractéristiques de référence

TERMINAL STEP	A1	A2	B1	B3
1	-	+	+	-
2	-	+	-	+
3	+	-	-	+
4	+	-	+	-

CW
CCW

Séquence de commande

Annexe -2- : Circuit de commande

Circuit L297

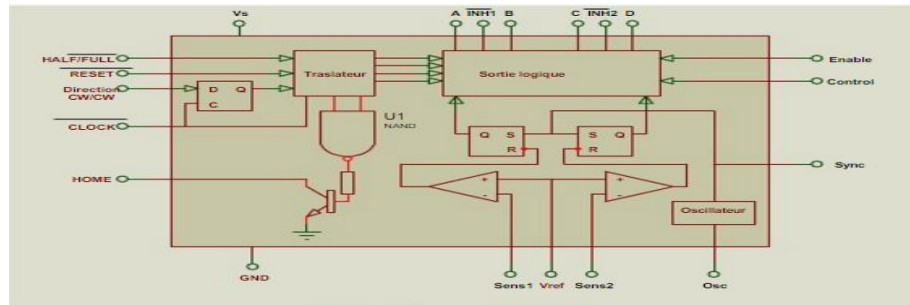


PIN FUNCTIONS - L297

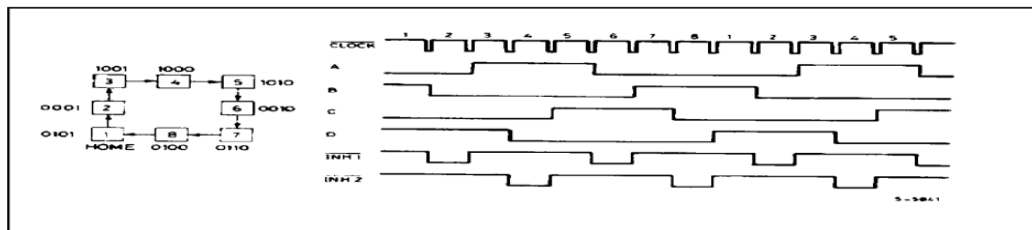
N°	NAME	FUNCTION
1	SYNC	Output of the on-chip chopper oscillator. The SYNC connections The SYNC connections of all L297s to be synchronized are connected together and the oscillator components are omitted on all but one. If an external clock source is used it is injected at this terminal.
2	GND	Ground connection.
3	HOME	Open collector output that indicates when the L297 is in its initial state (ABCD = 0101). The transistor is open when this signal is active
4	A	Motor phase A drive signal for power stage.
5	$\overline{\text{INH1}}$	Active low inhibit control for driver stage of A and B phases. When a bipolar bridge is used this signal can be used to ensure fast decay of load current when a winding is de-energized. Also used by chopper to regulate load current if CONTROL input is low.
6	B	Motor phase B drive signal for power stage.
7	C	Motor phase C drive signal for power stage.
8	$\overline{\text{INH2}}$	Active low inhibit control for drive stages of C and D phases. Same functions as INH1
9	D	Motor phase D drive signal for power stage.
10	ENAB E	Chip enable input. When low (inactive) INH1, INH2, A, B, C and D are brought low
11	CONTR OL	Control input that defines action of chopper. When low chopper acts on INH1 and INH2; when high chopper acts on phase lines ABCD.
12	Vs	5V supply input
13	SENS2	Input for load current sense voltage from power stages of phases C and D
14	SENS1	Input for load current sense voltage from power stages of phases A and B.
15	Vref	Reference voltage for chopper circuit. A voltage applied to this pin determines the peak load current.
16	OSC	An RC network (R to VCC, C to ground) connected to this terminal determines the chopper rate. This terminal is connected to ground on all but one device in synchronized multi - L297 configurations. $f \cong 1/0.69 RC$
17	CW /CCW	Clockwise/counterclockwise direction control input. Physical direction of motor rotation also depends on connection of windings. Synchronized internally therefore direction can be changed at anytime
18		Step clock. An active low pulse on this input advances the motor one increment.

	$\overline{\text{CLOCK}}$	The step occurs on the rising edge of this signal
19	$\overline{\text{HALEF}} / \overline{\text{FULL}}$	Half/full step select input. When high selects half step operation, when low selects full step operation. One-phase-on full step mode is obtained by selecting FULL when the L297's translator is at an even-numbered state. Two-phase-on full step mode is set by selecting FULL when the translator is at an odd numbered position. (The home position is designate state 1)
20	$\overline{\text{RESET}}$	Reset input. An active low pulse on this input restores the translator to the home position (state 1, ABCD = 0101).

Fonctionnement des broches



Structure interne du circuit L297.



Mode monophasé

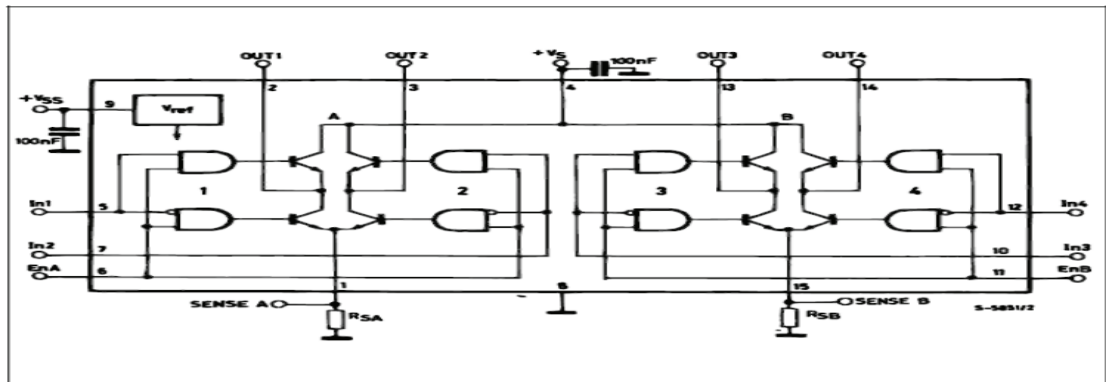
Circuit L298



MW.15	PowerSO	Name	Function
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load
2;3	4;5	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	VS	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.

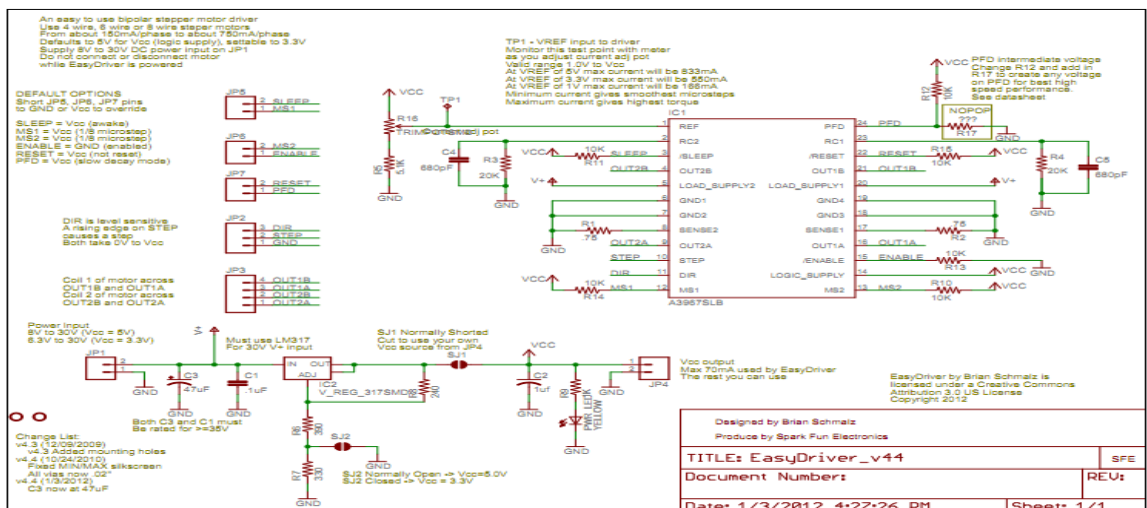
5;7	7;9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	Enable A; Enable B	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1,10,11,2 0	GND	Ground.
9	12	VSS	Supply Voltage for the Logic Blocks. A100nF capacitor must be connected between this pin and ground.
10; 12	13;15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
-	3;18	N.C.	Not Connected

Fonctionnement des broches



Architecture interne du L298

Circuit easy driver

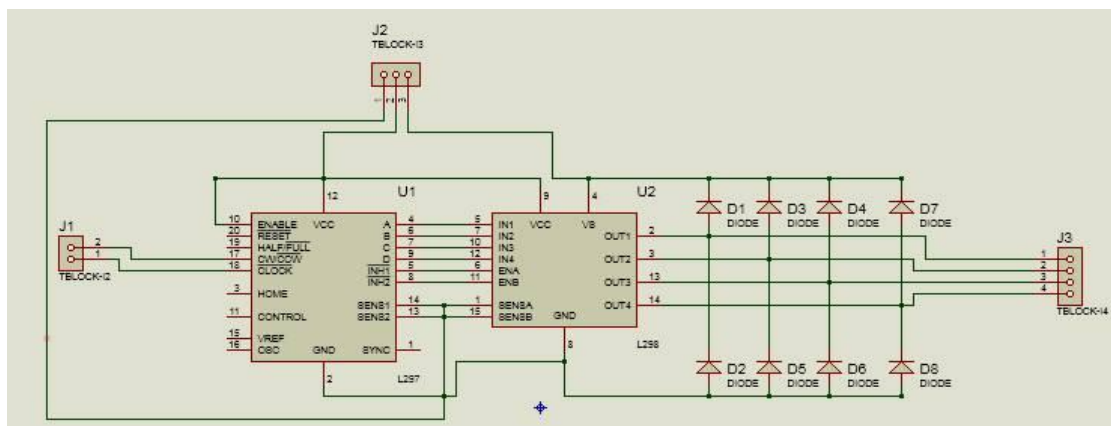


Schématique easy driver 4.4

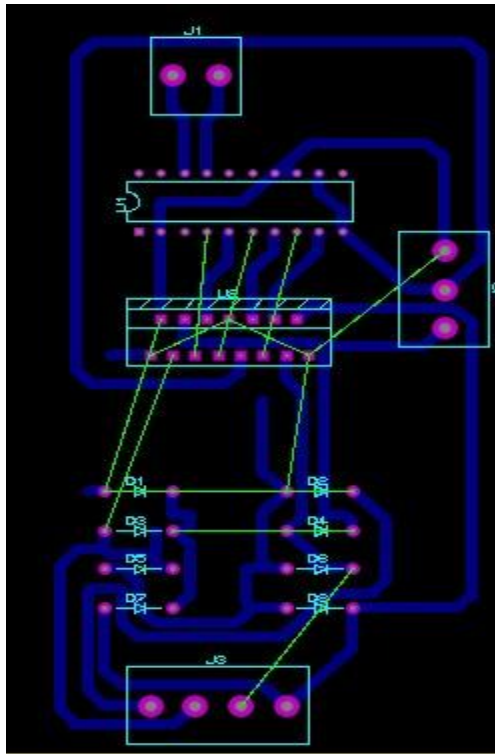
ANNAXE-3- G CODE

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

Annaxe-4- circuit imprimée



Isis le couple I297/I298



ARES du circuit I297/I298

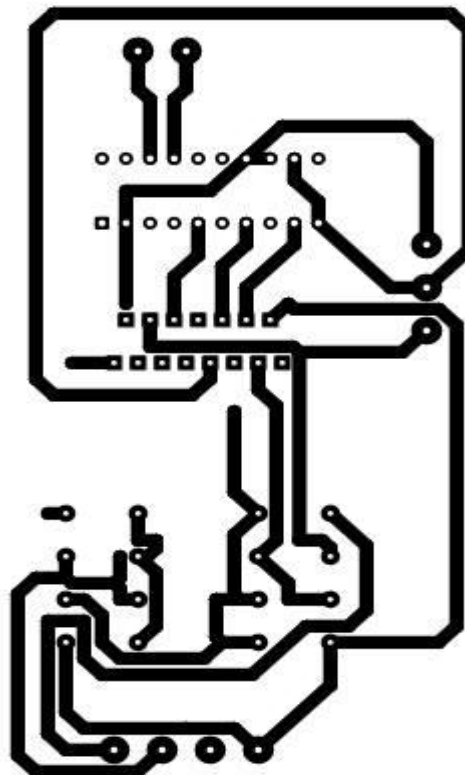


Schéma de réalisation

Résumé

Le travail développé dans le cadre de ce mémoire a porté sur l'étude des différents moteurs pas à pas proposés dans la littérature en vue de commander une machine CNC, c'est-à-dire :

- ✚ La simulation de la commande de trois moteurs pas à pas avec le logiciel proteus « isis ». On a utilisé la carte Arduino Uno et le circuit L297 avec L298.
- ✚ On a exploité les logiciels de commande de la machine CNC pour la lecture d'image pour la tracer et la génération fichier G-code. Puis nous avons choisi le logiciel Gcode Sender pour transférer le fichier G-code vers la carte Arduino Uno.
- ✚ On a utilisé la carte Arduino Uno et Easy Driver dans la réalisation pratique.

Dans la dernière partie du mémoire, nous présentons la fabrication de la machine CNC et les tests sur le fonctionnement de cette machine.

Les mots clé : moteur pas à pas, cnc machine, commande, arduino, circuit intégré .

Abstract

The developed work in this memory is taking over the study of different motors pas à pas proposed in the literature to command a machine CNC , it means : The simulation of the command of three motors pas à pas with the logiciel Proteus <isis> .we used the card Arduino Uno and the circuit L297 with L298 .

We have exploited the logiciels of the command of the machine CNC for the lecture of the image for draw it ,and the generation file. Then we have choose the Gcode Sender logiciel to transfer the G-code file to the Arduino Uno card. We have used the Arduino Uno card and Easy Driver in the pratical realisation . In the last part of the memory , we present the fabrication. Of the CNC machine and the tests over the functioning of this machine .

المخلص

في اطار المذكرة قمنا بدراسة المحرك خطوة بخطوة و تطبيق التحكم في المحرك على الة التحكم الرقمي. محاكاة التحكم في 3 محركات خطوة بخطوة باستخدام برنامج بروتوس و استخدمنا الادوات التالية : كارط اردوينو , دائرة التحكم 1297 و 1298 .

شرح برنامج التحكم في الة التحكم الرقمي و استخدام برنامج لتحويل الصورة او الكتابة الى لغة الالة (Gcode) و إرساله الى كارط اردوينو عن طريق برنامج universel Gcode sender .

استخدمنا المتحكم في الموتور easy driver مع الاردوينو في الجانب التطبيقي .

في الاخير قمنا بصناعة الة تحكم رقمي و تجربتها برسم على ورق .