République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



UNIVERSITE DE MEDEA
Faculté des Sciences et de la technologie
Département de Génie électrique

MEMOIRE DE FIN D'ETUDES

En vue de l'obtention du Diplôme de Master en Génie électrique

Option : Commande électronique industrielle

Thème

Simulation et commande d'une machine CNC à base des moteurs pas à pas

Réalisé par :
Mlle. BEN MHIRIZ Habiba
Mlle. AMRAOUI Afaf

Dirigé par :
Dr. BETTACHE Farouk

Promotion : 2016/2017

Remerciment

Au terme de cette mémoire je tiens à remercier tout naturellement en premier lieu ***ALLAH*** *le* ***tout Puissant*** *qui m'a donné la force, le courage et la patience de bien mener ce travail.*

Nous remercions spécialement notre promoteur ***Dr. Bettache Farouk.*** *Pour l'effort fourni, les conseils prodigues, sa patience et nous remercions mombres de jurés.Nous remercions aussi tous les enseignants de la periode d'étude de primaire jusqu'à la fin de notre formation. Nous adressons également les remerciments à tout le groupe de commande electronique industriel .*

Ben mhiriz Habiba

Amraoui Afaf

DEDICACES

Je dédie ce mémoire

A mes très chers parents : Grâce à leurs tendres encouragements, leur. Amour et leurs grands sacrifices, ils ont pu créer le climat affectueux Et propice à la poursuite de mes études. Aucune dédicace ne pourrait Exprimer mon respect, ma considération et mes profonds sentiments Envers eux. Je prie le bon dieu de les bénir, de veiller sur eux, en Espérant qu'ils seront toujours fiers de moi.

A mes adorables sœurs hanan, nassira,

A mon binôme Amraoui Afaf qui est toujours avec moi quel que soit et dans des moments très dur.

A mes amis proches : Afaf, fadila Nadjet, choucho

A tous mes collègues de commande électronique industriel de la promotion 2016-2017

Benmhiriz Habiba

DEDICACES

Avant tous, je remercie Dieu le tout puissant de m'avoir donner le courage et la patience pour réaliser ce travail malgré toutes les difficultés rencontrées.

Je dédie ce modeste travail :

A mes très chers parents, que Dieu les garde et les protèges pour leurs soutien moral pour leurs encouragements et les sacrifices qu'ils ont endurés.

A mes sueurs

A ma grande famille

Je dédie également à tous mes amis .

AMRAOUI AFAF

ملخص

في هذه المذكرة قمنا بدراسة المحرك خطوة بخطوة من اجل استعماله على آلات التحكم الرقمي, بمعنى :

محاكاة التحكم في 3 محركات خطوة بخطوة باستخدام برنامج بروتوس مع استعمال بطاقة الاردوينو , دارة قيادة المحركات L298 و L297 .

كما تم دراسة برنامج التحكم في آلة التحكم الرقمي و استخدام برنامج من اجل تحويل الصورة أو الكتابة إلى لغة لآلة (G-code) و إرساله إلى بطاقة الاردوينو عن طريق برنامج universel Gcode Sender.

الكلمات الرئيسية: المحرك خطوة بخطوة, آلة التحكم الرقمي , بطاقة الاردوينو دارة قيادة المحركات .L297 و L298

Résumé

Le travail développé dans ce mémoire à nous porté sur l'étude des différents moteurs pas à pas utilises dans les machines à commande numérique (CNC), c'est à-dire :

Simulation de la commande de trois moteurs pas à pas avec le logiciel Proteus « isis» avec l'utilisation de la carte Arduino Uno et le circuit L297 et L298.

Le logiciel de commande de la machine CNC pour la lecture d'image, le traçage et la génération de fichier G-code a été étudie. Puis nous avons choisi le logiciel G-code Sender pour le transfère du fichier G-code vers la carte Arduino Uno.

Mots clé : moteur pas à pas, machine CNC, carte Arduino, circuit intégré L297 et L298 .

Abstract

The developed work in this memory is taking over the study of different stepper motors using to command a machine numerical control by computer (CNC), it means: the simulation of the command of three with the logiciel Proteus <Isis> with using the card Arduino Uno and the circuit L297 and L298.

The logiciel of the command of the machine CNC for the lecture of the image for draw it and the generation file is study. Then we have choose the G-code Sender logiciel to transfer the G-code file to the Arduino Uno card.

Keywords: Stepper motors, CNC machine, Adriano board, integrated circuit L297 and L298.

Introduction générale……...……..……...……..…… **1**

Chapitre I : Généralité sur le Machine CNC

I.1 Introduction...…….…..…… ……. **3**

I.2. Historique…….…….. **4**

I.3.Définition…… **4**

I.3.1. Les commandes numériques…… **5**

I.3.2.Les commandes CNC……... **5**

I.4 Principe de fonctionnement d'une machine numérique …… **5**

I.5 La commande numérique par calculateur(CNC) ……... **6**

I.5.1 La Conception Assistée par Ordinateur (CAO) …….. **7**

I.5.2 La Fabrication Assisté par Ordinateur (FAO)……..... **7**

I.5.3 Le post-processeur. ...…… ……...…….…… **7**

1.6 Les types de la machine CNC à 3 axes…….. **8**

I.7 Programmation de commande numérique…….. **9**

I.7.1 langage…….…… **9**

I.7.2 Fonctions préparatoires G……..... 10

1.8. Conclusion……...……... **11**

Chapitre II : Composants matériels de la machine CNC à 3axes

II.1.Introduction……... 12

II.2 Moteur pas à pas…… 12

II.2.1 Historique……... 12

II.2.2 Généralités……...…..…… 12

II.2.3. Technologie des moteurs pas à pas……..... 14

II.2.3.1. Moteurs pas à pas rotatifs …… 14

II.2.3.1.1. Moteur pas à pas rotatif à reluctance variable…… 14

II.2.3. 2. Moteur pas à pas rotatif aimants permanents…….. 16

II.2.3.2.1. Moteur à aimant permanent unipolaire……..……. 16

II.2.3.2.2. Moteur à aimant permanent bipolaire ….……... 19

II.2.3. 3. Moteur pas-à-pas hybride MH……... 21

II.2.4. Comparaison entre les différents types des moteurs pas à pas…….. 23

II.2.5.Avantages et inconvénients des moteurs pas à pas……. 23

II.3. Définition du module Arduino……. 24

II.3.1.Les différents types des cartes‘Arduino’ ……... 24
II.3.1.1.La carte Arduino Méga……... 24
II.3.1.2.La carte Arduino Uno et Duemilanove…….…….. 25
II.3.2.Les gammes de la carte Arduino……..... 25
II.3.3.Caractéristiques sélectifs de la carte Arduino UNO……..... 27
II.3.4. Mise en œuvre de la carte Arduino Uno…… 28
II.3.4.1. présentation et caractéristiques du module Arduino Uno……... 28
II.3.4. 2.Description matérielle……. 30
A) Les sources de l'alimentation de la carte ……. 31
B) Les entrées analogiques……...…….. 32
C) Connexions numériques……...…….…. 32
II.3.4.2.1. Connecteurs E/S ……...….. 33
II.3.4.2.2. Visualisation…… 33
II.3.4.3. Description logicielle……..…… …….. 33
a) Logiciel d’édition et programmation Arduino…… 34
b) Présentation des parties principales du logiciel..... 35
c) Les boutons……... 35
II.4.Les drivers ……...……..... 39
II.4.1. Circuit L298 ……..... 39
II.4.2. Le Circuit intégré L297……...……... 40
II.4.2.1 Le schéma théorique du montage …….……... 40
II.4.3. Commande moteur pas à pas……..... 41
II.4.3.1 alimentation…… 41
II.4.3.2 Commande directe avec un amplificateur L298……..... 42
II.4.3. 3 La commande dir/step…… 42
II.5.Conclusion……..... 43

Chapitre III : Logiciels du pilotage du CNC

III.1. Introduction……..... 44
III.2. Définition Inkscape …… 44
III.2.1 La fenêtre Inkscape……... 45
III.2.2 Création du document…….. 47
A. Paramétrer le format souhaité……..... 47
B. Vectorisation d'une image bitmap ……..... 48
C. Génération du Gcode……..….... 50

III.3. Définition de Universal Gode Sender…… 51
III.3.1. Logiciel de pilotage (Universal Gode Sender)……..... 51
III.3.1.1. mode de commande……..... 53
III.3.1.2. Mode Configuration…….. 56
III.3.1.3. Mode manuel……..... 56
III.3.1.4. Mode automatique ……..... 58
III.3. 2. Langue de contrôleurs numériques « G-code »……..... 58
III.3.2.1. Définition de G-Code …… 58
III.3.2.2. Format d'une ligne…… 59
III.3.2.3.Les principales fonctions de G-code……..... 59
III.3.2.3.1.Les fonctions préparatoires, instructions G……...……. 59
III.3.3. Langage de programmation la machine CNC ……..... 62
III.3.3.1. Programme Grbl……..... 62
III.4. Conclusion……..... 64

Chapitre IV : simulation et la commande de la machine CNC

IV.1. Introduction…… 65
IV.2.Schéma synoptique…….. 65
IV.3 Schéma électrique…… 69
IV.4 Simulation et test le programme…….. 69
IV.3. Conclusion ……. 71
Conclusion générale……...…… 72
Bibliographie.
Annexe.

Liste des figures

Figure I.1: Décomposition d'une machine numérique……...……...5

Figure I.2: machine à trois axes…….....6

Figure I.3: Cycle d'une pièce fabriquée par une commande numérique (CFAO)…….....7

Figure I.4: machine CNC à 3 axes……..8

Figure I.5 : machine CNC à 3 axes……...……8

Figure 1.6 : machine CNC à 3 axes……...……...……9

Figure II.1 : Vue d'ensemble d'un moteur pas à pas…….……...……13
Figure II.2: moteur à réluctance variable……...……15
Figure II.3: Représentation schématique d'un moteur unipolaire ……16

Figure II.4: fonctionnement d'un moteur pas-a-pas unipolaire…….....17

Figure II.5: Représentation schématique d'un moteur bipolaire……..... 19

Figure II.6: Fonctionnement à pas complet……...……..... 20

Figure II.7: Fonctionnement avec couple maximal…….. .20

Figure II.8: Fonctionnement à demi-pas……...…….21

Figure II.9: Moteur hybride (MH)……..…...22

Figure II.10: Carte Arduino "Méga" ……25
Figure II.11 : Carte Arduino "Duemilanove" et "Uno"……...…..25

Figure II.12: Module arduino Atmel uno ……..29

Figure II.13: Entrées/sorties numériques D0 à D13……..…….....32

Figure II.14 : Interface du logiciel Arduino……...……...34
Figure II.15 : présentation des parties principales du logiciel……...35
Figure II.16: présent les boutons de Carte Arduino……...……35
Figure II.17 : choix d'une carte…….36
Figure II.18 : Choix du port de connexion de la carte……...……37
Figure II.19 : ouverture d'un programme……38
Figure II.20 : Compilation du programme……..…….38
Figure II.21 : Téléchargement du programme ……..…...39
Figure II.22 : commande moteur par le couple L297 et L298Arduino……...…...41

Figure II.23: commande moteur par le circuit L298……...…42

Figure II.24 : le circuit l297……..…….…..….43

Figure.III.1 l'interface principale d'Inkscape……44

Figure.III.2 : Création du document…….....47
Figure III.3: Importation d'une image bitmap……...…….48

Figure III.4 : Vectorisation d'une image bitmap……...…….48

Figure III.5: image vectoriel à vocation "artistique……...…49

Figure III.6 : Image vectorielle et image bitmap……...……49

Figure III.7 : points d'orientation……..……50

Figure III.8:bibliothèque d'outils……...…50

Liste des figures

Figure III.9: chemin vers G-code…….....51
Figure III.10 : Universel Gcode Sender……...…..52
Figure III.11 : Onglet « Machine Control »de l'UGC…….....53
Figure III.12: l'interface de "G-code Sender" on mode manuel……..…..54
Figure III.13: Onglet « File commande » de l'UGCS……..…54
Figure III.14: l'interface de "G-code sender" on mode automatique…….……55
Figure III.15: envoyé fichée G code……..……56
Figure III.16: de mode manuel……..……57
Figure III.17 : Installations de **GRBL** dans l'arduino……...…….63
Figure. III.18 : ouvrire le GRBL……...…….....63
Figure III.19 : Moniteur série……...……..64
Figure IV.1 : Schéma synoptique……...…....65
Figure IV.2 : Conversion rosa .jpg vers rosa.nrg…….…..66
Figure IV.3 : Une partie du fichier **rosa**.nrg……...66
Figure IV.4 : visualisation du fichier rosa.nrg par universel Gcode sender……..67
Figure IV.5 : Chargment de fichier GRBL……68
Figure IV.6 : Envoyé le fichier vers la machine CNC……...….....68
Figure IV.7 : schéma électrique…….……69
Figure IV**.8 :** Montage électronique dans Proteus……..70
Figure IV.9 : Simulation de la machine CNC……....71
Figure IV.4 : simulation cnc machine……..……50
Figure IV.5: charge le fichier hex de programme…….....51
Figure IV.6 : Système de déplacement linéire 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…… **15**
Tableau I.2: Table Etat successifs des phases…… **18**
Tableau I.3: Etat successifs des phases lors de l'utilisation des demi-pas……. **18**
Tableau I.4: Comparaison entre les différents types des moteurs pas à pas……. **23**
Tableau I.5: Les avantages et les inconvénients des moteurs pas à pas……... **23**
Tableau III.1. Tableau d'index du G-code……. **61**

Abréviation

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

Introduction générale

Le monde de la machine-outil est en évolution permanente afin de répondre au marché, très réactif, des pièces usinées. De nos jours, il existe une grande demande de machines rapides à commande numérique, notamment pour l'industrie automobile.

Les premières machines-outils à architecture parallèle ont fait leur apparition sur le marché. Elles sont inspirées des mécanismes et des robots parallèles largement étudiés par les roboticiens au cours de ces 25 dernières années. Les outils proposés sont l'établissement des modèles géométrique, cinématique.

L'industrie exige des machines de plus en plus performantes en termes de vitesses, d'accélérations, de précision, de productivité, de fiabilité, ...

La conception assistée par ordinateur C.A.O. est devenue un outil privilégié en engineering. Elle permet un gain considérable en temps et en coût lors de l'élaboration d'un projet de conception d'une machine.

de notre travail est l'étude et simulation d'une machine CNC à trois axes qui permet de reporter une image sur **pc** vers un support papier ou bois..

Pour aboutir à cet objectif, nous avons divisé notre travail on quatre chapitres **:**

Le premier chapitre, est généralité sur les Machine CNC (commande numérique), dont le but de montrer la définition et l'historique et le principe de fonctionnement et types de la machines CNC à 3 axes et La programmation de commande numérique(CN).

Le deuxième chapitre est la description des parties matériels constituons notre machine CNC. Où nous allons voir en premier lieu les moteurs pas à pas et en seconde lieu, on donnera une description électrique de la carte Arduino puis on terminera par les circuits intégrés (L297, L298)

Dans le troisième chapitre nous allons présenter les différent logiciels utilisés dans le C.A.O (conception assister par ordinateur) de notre machine dont nous allons présenter en premier lieu l'éditeur des images et générateur de code**G** le logiciel **I**nkscape puis on seconde lieu on présentera le logiciel **U**niversel **G**code **S**ender dont le rôle est la transmission vers l'extérieur à travers le port série puis on termine par le programme GRBL qui joue le rôle d'interpréteur du Gcode.

Dans le dernier chapitre, nous allons présenter le lien entre la partie hardware et la partie software puis un test par le logiciel proteus.

Et on termine par une conclusion générale.

CHAPITRE I

Généralité sur le Machine CNC

I.1. Introduction

La commande numérique (CN) est une technique utilisant des données composées de codes alphanumériques pour représenter les instructions géométriques et technologiques nécessaires à la conduite d'une machine ou d'un procédé.

C'est également une méthode d'automatisation des fonctions des machines ayant pour caractéristique principale une très grande facilité d'adaptation à des travaux différents. À ce titre, la commande numérique constitue l'un des meilleurs exemples de pénétration du traitement de l'information dans les activités de production.

Exploitant au maximum les possibilités de la micro-informatique, toutes les données sont traitées en temps réel, c'est-à-dire au moment où elles sont générées, de manière à ce que les résultats du traitement contribuent également à piloter le processus.

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

Une CNC 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 :

- Impriment 3d.
- Fraiseuse numérique
- Tour numérique
- La machine de découplage.
- Mini traceur.

Ce premier chapitre présente le contexte d'étude à travers une description des machines CNC.

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.Définition :

I.3.1. Les commandes numériques :

Le terme NC a été repris du vocabulaire technique américain et est l'abréviation de Numérique Control, en français Contrôle numérique. Cela signifie que la commande se fait par des chiffres.

Une machine NC est librement programmable. Les déplacements dans les différents axes sont initiés par un programme qui contient des informations sur la géométrie de la pièce ainsi que les données techniques pour la production (fréquences de rotation, avance, etc.).

Les machines NC sont composées d'une combinaison d'axes linéaires et rotatifs. Chaque axe est équipé d'un système de mesure électronique et d'un entraînement réglable. Les commandes NC pures ne disposent en général d'aucune ou de très peu de possibilités pour mémoriser des données de programmation. Les données sont le plus souvent lues par un lecteur de bande perforée.

Aujourd'hui, les commandes de machines NC ne sont quasiment plus utilisées. Le terme Commande N C'est encore largement répandu, mais représente plutôt l'idée générique pour toute la famille des techniques NC et CNC.

I.3.2.Les commandes CNC :

Les techniques CNC se sont développées à partir des commandes NC. L'abréviation signifie Compteriez Numérique Control. Ces commandes sont basées sur l'utilisation de microprocesseurs, ce qui offre des avantages décisifs par rapport aux machines NC. Elles offrent entre autres une mémoire beaucoup plus grande. Grâce à un système opératoire et des logiciels correspondants, il est possible d'adapter les programmes de manière flexible et rapide. D'autre part, il est possible de traiter des grandes quantités de données, ce qui permet de commander plusieurs axes simultanément. Grâce à une approche conviviale, l'établissement de programmes et leur gestion sont plus rapides et plus flexibles.

I.4. 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.

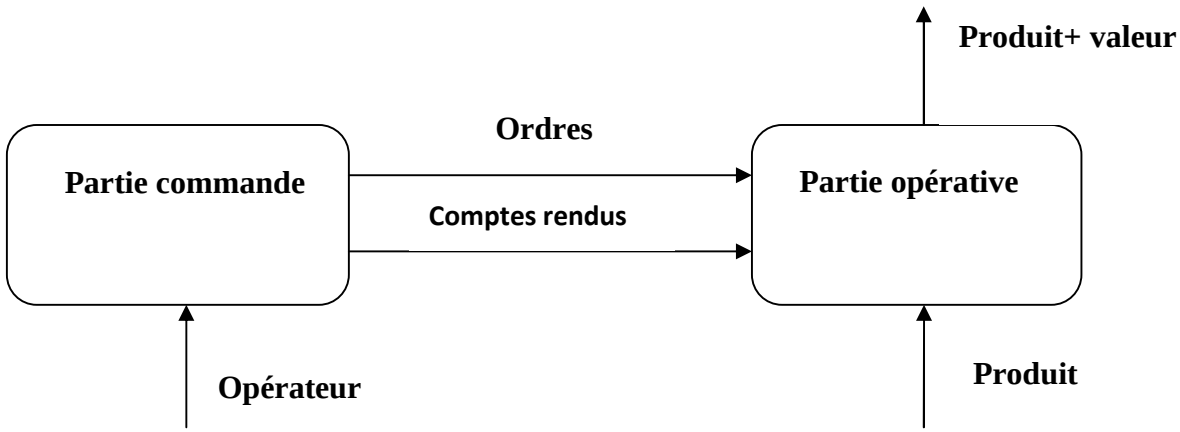


Figure I.1 Décomposition d'une machine numérique.

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 (Om) qui correspond à la position de référence de la machine où Xm=0, Ym=0 et Zm=0 (s'il y en a).
- L'origine pièce (Op) qui peut être décalée par rapport à l'origine machine faciliter l'usinage.

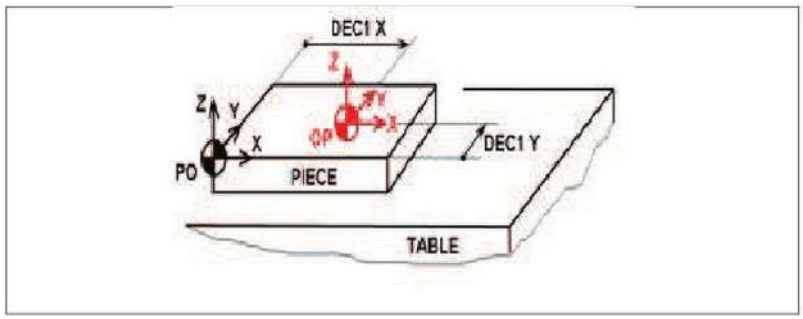


Figure I.2: machine à trois axes

I.5. La commande numérique par calculateur(CNC) :

C'est en 1942 aux États-Unis que la CN a commencé à être exploitée, pour permettre l'usinage de pompes à injection pour moteurs d'avions. Il s'agissait en fait de cames, dont le profil complexe était irréalisable au moyen d'une machine traditionnelle. La commande numérique a pour tâche de générer des mouvements d'axes. Ces instructions qui viennent du programme pièce ou de l'opérateur machine consisteront en une position et une vitesse de déplacement. Le processeur de la commande numérique va alors générer une consigne afin qu'on puisse commander les moteurs d'axes. Elle aura également pour tâche, lors du déplacement des axes, de vérifier la position de ces derniers et dans certain cas, la vitesse de déplacement. Les commandes numériques actuelles sont capables d'effectuer des mouvements en combinant simultanément les positions sur les axes X, Y et Z. Elles s'appellent commande numérique par calculateur. De plus, elles sont capables de gérer plusieurs axes simultanément. On parle alors d'interpolation. Cette dernière peut être linéaire ou circulaire selon la consigne générée par le microcontrôleur de la partie commande. Le rôle

de ce processeur est d'interpréter un code machine que l'on appelle un code G, puis de générer les signaux de commande des axes et de la broche.

I.5.1. 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.5.2. 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.5.3. 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 tien compte de la cinématique de la machine, de ses courses et de ses capacités. Il signale les erreurs et exprime les trajectoires dans le langage spécifique à la machine. Chaque machine possède donc son propre post-processeur (figure I.3)

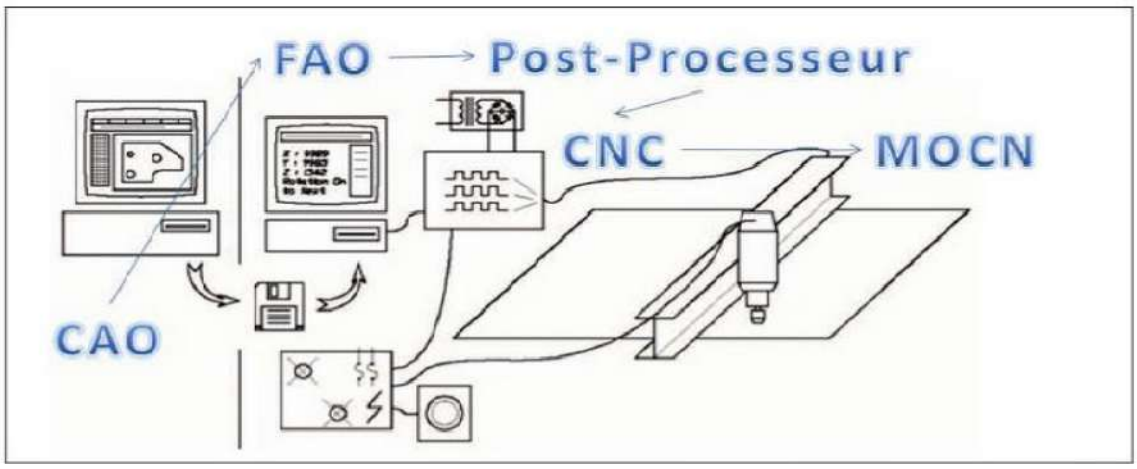


Figure I.3 : Cycle d'une pièce fabriquée par une commande numérique (CFAO)

1.6 Les types de la machine CNC à 3 axes :

Il y a plusieurs types de la machines CNC à 3 axes et ces types se classer par le déplacement des axes de la machine CNC [3], comme le montre les images suivantes :

- **Type 1:**La perceuse et son support se déplace et la table fixée

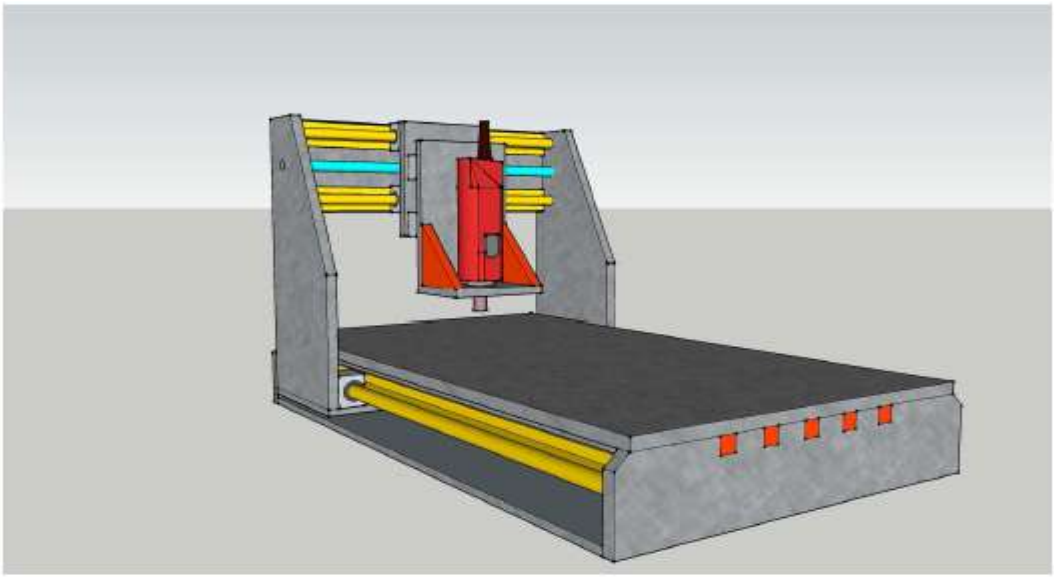


Figure 1.4: machine CNC à 3 axes

- **Type 2:**La perceuse se déplace verticale uniquement et la table se déplace sur tous les sens.

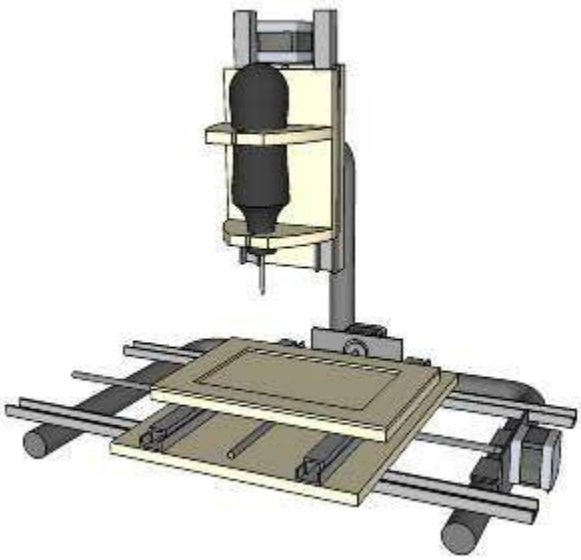


Figure 1.5 : machine CNC à 3 axes

- **Type 3 :** La perceuse se déplace horizontale et la table sur une seul sens

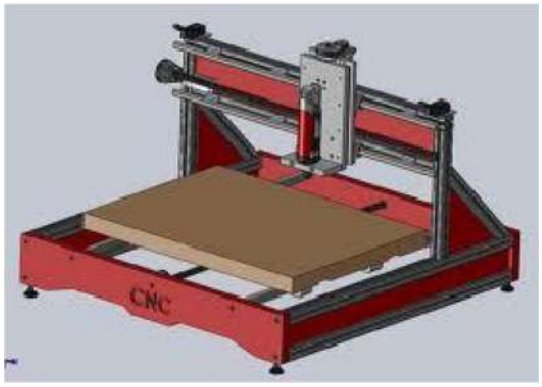


Figure 1.6 : machine CNC à 3 axes

I.7.Programmation de commande numérique

La programmation de commande numérique (CN) permet de commander numériquement, avec des instructions, les déplacements des différents organes mobiles.

C'est le directeur de commande numérique (DCN) qui interprète les instructions, reçoit les informations des capteurs et agit (par l'intermédiaire d'un variateur électronique) sur les moteurs.

Il existe plusieurs fabricants de DCN : Fanuc, NUM, Heidenhain, Siemens, Philips, Makino, Fagor, Selca, Fidia, MAZAK.

I.7.1.Langage :

Depuis l'établissement de la norme ISO 6983, et les technologies évoluant rapidement, de nombreuses extensions sont été ajoutées pour tenir compte des nouveautés et des nouvelles capacités des machines-outils. Ces extensions, bien que souvent utiles chez des constructeurs différents de Directeur de Commande Numérique, n'entrent pas dans la norme et compliquent la tâche des logiciels de Fabrication assistée par ordinateur, qui doivent créer les lignes de ce langage pour un DCN particulier. A côté de l'ISO, de nouveaux codes sont apparus, différents selon le Directeur de Commande Numérique. Ils intègrent de plus en plus, en plus de l'ISO, des langages propres aux constructeurs de DCN (symbolique, C, etc.) ainsi que des interfaces de programmation conversationnelle destinées à simplifier la programmation (voir CN Mazak, Heidenhain ou Selca).

- Fonctions préparatoires (G), fonctions d'appel de mode d'interpolation (G 0), cycles machine.
- Coordonnées de points (X, Y, Z… I, K…)
- Vitesses, avances… (S, F…)

- Fonctions auxiliaires (M…) qui permettent d'enclencher la lubrification, de changer d'outil, ou de déclencher des accessoires%1

I.7.2 Fonctions préparatoires G

La fonction d'interpolation linéaire rapide **G 0** (ou G0) (avance rapide de l'outil : quand l'outil se déplace).

La fonction d'interpolation linéaire (à la vitesse programmée) **G 01** (ou G1) (avance travaille: c'est à dire quand l'outil usine la pièce).

La fonction d'interpolation circulaire **G 02** (ou G2) dans le sens horaire) et **G 03** (ou G3) dans le sens trigonométrique.

La fonction de temporisation (programmable avec F, X(ou P) **G 04** (ou G4).

La fonction d'arrêt précis en fin de bloc **G 09** (ou G9).

Il est aussi possible sur certains pupitres de programmation d'utiliser ces deux codes (G2 G3) pour créer une interpolation circulaire, des fonctions d'interpolation à base de courbe NURBS **G 06.2**.

Sur les fraiseuses équipées de tête bi rotative les codes **G17, G18, G19** définissent l'axe des cycles de perçage, taraudage...et le plan dans lequel seront réalisées les interpolations circulaires et activé le correcteur de rayon d'outil.

- **G17** : Axe d'outil Z , interpolations G2,G3 et correction rayon dans le plan X Y.
- **G18** : Axe d'outil Y , interpolations G2,G3 et correction rayon dans le plan Z X.
- **G19** : Axe d'outil X , interpolations G2,G3 et correction rayon dans le plan Y Z.

Les codes de la famille **G52, G53, G54, G55...**sont utilisés pour:

- Programmer un **décalage d'origine.**
- Définir que les déplacements sont relatifs à **l'origine machine**.
- Choisir le numéro de **l'origine pièce**.

Certains codes G de la famille **G60 G70** peuvent être utilisés par les fabricants de DNC pour :

- Le choix de la programmation **cartésienne** ou **polaire**.
- L'activation d'un **facteur d'échelle**.
- La mise en action d'une fontion **miroire**.
- La programmation en mesure **métrique** ou en **pouce**.

Les codes **G90 G91**définissent la programmation **absolue** ou **incrémentale** des cotes.

Des cycles préprogrammés sont également accessibles sur la plupart des machines : **G 81, 82, 83...** Pour les cycles de perçage, taraudage, etc. avec l'annulation par **G 80**. D'autres cycles peuvent être présents selon le type de machine (tour "cycle d'ébauche G71, G72, G73..." , fraiseuse, aléseuse, fil, ...).

1.8 Conclusion :

Nous avons présentés dans ce chapitre une vue générale sur les machines-outils à commande numérique, dont le but de montrer la définition et l'historique et le principe de fonctionnement de ce dernier et la conception assistée par ordinateur (CAO) et(FAO).

En fin on donne quelques exemples pour les machines outil à commande numérique dans l'industrie.

CHAPITRE II

Composants matériels de la machine CNC à 3 axes

II.1.Introduction

Dans ce chapitre, on présentera les composants matériels principaux qui rentrent dans la constitution de la machine CNC à savoir :

- Les moteurs pas à pas.
- La carte d'interface et de commande : ArduinoUno.
- Les drivers L297 et L298

II.2.Moteur pas à pas

Le moteur pas à pas permet de transformer une impulsion électrique en un mouvement angulaire. Ce type de moteur est très courant dans tous les dispositifs où l'on souhaite faire du contrôle de vitesse ou de position, comme par exemple les imprimantes.

Le stator est constitué de bobines qui sont alimentées, à tour de rôle, en courant continu par une carte électronique. Le rotor est un croisillon, en métal ferreux ne conservant pas le magnétisme. Si on compte électroniquement les impulsions envoyées aux bobines on sait, connaissant le pas, le nombre de rotations que le rotor a effectuées. Représentation du moteur pas à pas [4].

II.2.1 Historique

Les premiers moteurs pas -à- pas à réluctances variable ont été utilisés par la marine de guerre britannique dans les années 1920 pour déplacer les indicateurs de direction de lance - torpilles et des canons. Dans les années 1930, l'ingénieur Marius Lavet a découvert un type particulier de moteur pas à pas à aimant, connu maintenant sous le nom de moteur Lavet , qui a permis le développement de ce dispositif dans les domaines de l'hodologie grâce à sa miniaturisation et à son faible cout.

Il équipe aujourd'hui presque toutes les montres à aiguilles .Le moteur pas à pas classique est apparu dans les années 1940, mais c'est l'avènement de l'électronique numérique dans les années 1960 qui a permis son développement [5].

II.2.2. Généralités

Il existe trois types de moteur pas à pas [6]:

- les moteurs à aimants permanents.
- moteurs à réluctance variable.
- moteurs hybrides.

Malgré les différences existant entre les moteurs, le résultat recherché est l'avance d'un seul pas, c'est-à-dire la rotation de leur axe suivant un angle déterminé à chaque impulsion

que l'une ou l'autre de leurs différentes bobines recevra. Cet angle, qui varie selon la constitution interne du moteur, est en général compris entre 0.9° et 90°.

Les moteurs les plus couramment rencontrés possèdent des pas de :

- 0,9° soit 400 pas par tour.
- 1,8° soit 200 pas par tour.
- 3,6° soit 100 pas par tour.
- 7,5° soit 48 pas par tour.
- 15° soit 24 pas par tour.

Il est évident que les moteurs pas à pas, de par leur technologie, présentent une très grande précision et une durée de vie quasi illimitée, l'usure mécanique étant pratiquement inexistante (absence de frottements). La figure ci-dessous nous montre l'aspect externe d'un moteur pas à pas.

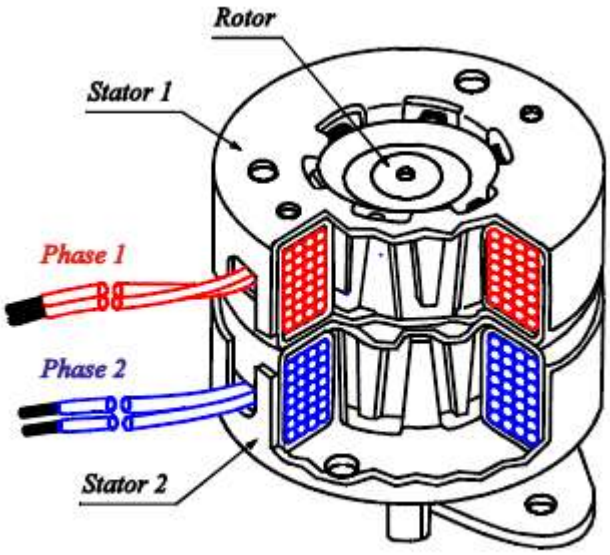


Figure II.1. Vue d'ensemble d'un moteur pas à pas

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 programmateurs mécaniques :

Les pôles magnétiques de rotation de même nom se 1repoussent 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.

II.2.3. Technologie des moteurs pas à pas :

Les moteurs pas à pas sont très utilisés dans toutes les applications mécaniques ou 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 [7].

II.2.3.1. Moteurs pas à pas rotatifs :

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

II.2.3.1.1. Moteur pas à pas rotatif à reluctance variable :

Les moteurs à reluctance variable (moteurs MRV) doivent leur nom au fait que le circuit magnétique qui les compose s'oppose de façon variable à sa pénétration par un champ magnétique. Ces moteurs sont composés d'un barreau de fer doux et d'un certain nombre de bobines. Lorsqu'on alimente une bobine, elle devient un électroaimant et le barreau de fer cherche naturellement à s'orienter suivant le champ magnétique. On alimente la phase 1, puis la phase 2, puis la phase 3 ... Si on veut changer le sens du moteur, il suffit de changer l'ordre d'alimentation des bobines.

Dans la pratique, le barreau de ferrite a plusieurs dents (ici 6). Dès qu'on alimente la phase 2, il y a une rotation de 15° (i.e. 60° - 45° = 15°), puis la phase 3, etc. Donc le moteur tourne de 15° dès qu'on alimente une phase. Il faut 24 impulsions pour faire un tour complet. C'est un moteur 24 pas [8].

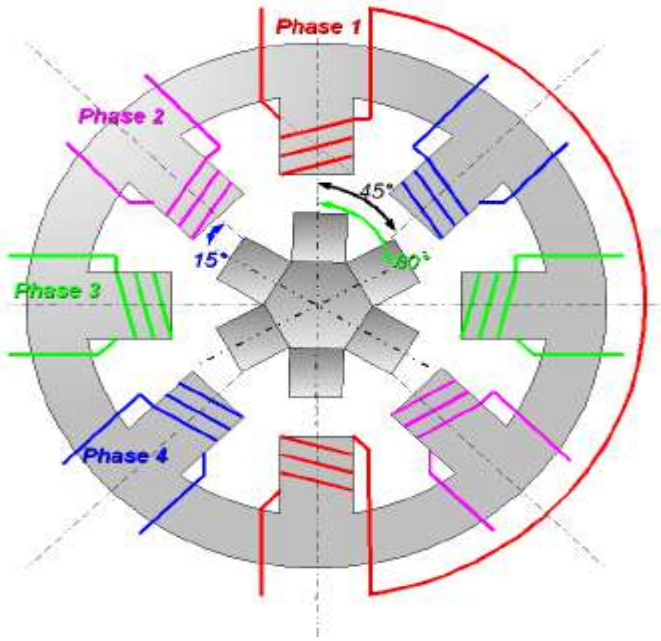


Figure II.2: moteur à réluctance variable

- **Principe de fonctionnement est très simple :**

Le principe de fonctionnement de ce type de moteur pas à pas est proche de celui du moteur hybride, avec une structure dentée au rotor et au stator.

Il n'y a pas d'aimant au rotor pour renforcer l'action du flux et donc pas de couple résiduel sans courant. Ce type de moteur pas à pas n'est presque plus utilisé ni fabriqué. [9]

Les séquences de commandes sont présentées dans le tableau suivant :

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

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

II.2.3. 2. Moteur pas à pas rotatif aimants permanents:

Les moteurs pas à pas à aimant permanent courant sont biphasés: ils sont constitués de deux enroulements (deux bobines distinctes) sur le stator, tandis que le rotor est un aimant permanent (un pôle nord et un pôle sud). Lorsqu'un enroulement est alimenté, il crée un champ magnétique avec lequel l'aimant permanent va tenter de s'aligner (le sud attirant le nord et vice/ versa).

En alimentant les enroulements successivement dans le bon ordre et dans le bon sens, on peut alors faire tourner le rotor dans le sens désiré. Cette succession d'alignement est ce qu'on appelle le pas. On distingue deux familles dans les moteurs pas à pas aimant permanent:

- Les moteurs unipolaires: le courant circule dans un seul sens dans des demi-bobines attirant toujours le même pole.
- Les moteurs bipolaires: le courant circule dans les deux sens dans des bobines, attirant tantôt le NORD tantôt le SUD [10].

II.2.3.2.1. Moteur à aimant permanent unipolaire

Les bobinages d'un moteur unipolaire sont alimentés toujours dans le même sens par une tension unique d'où le nom d'unipolaire [11].

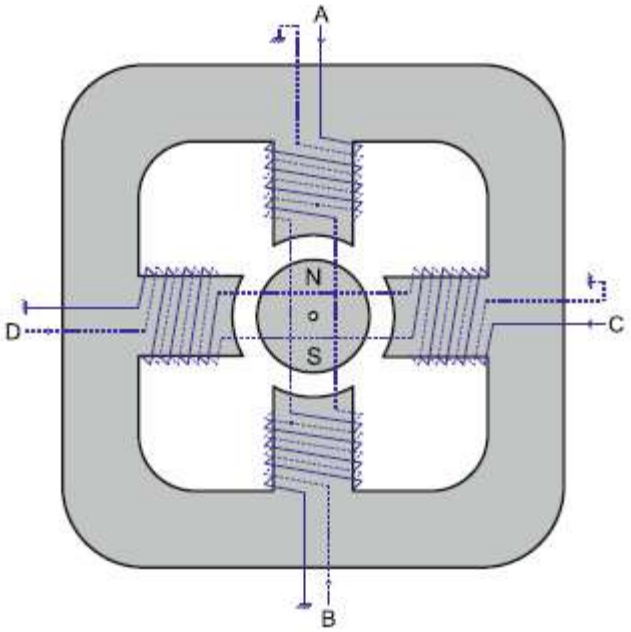


Figure II.3: Représentation schématique d'un moteur unipolaire.

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 [12].

➢ 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°/48 = 7,5°).

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 [13].

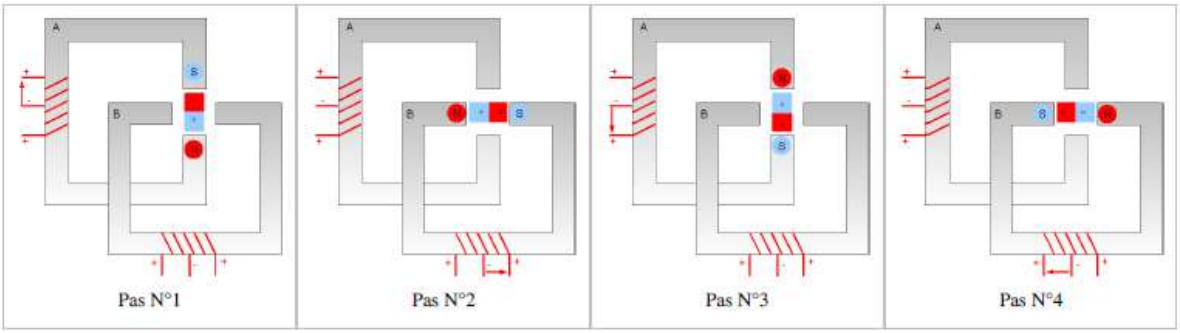


Figure II.4: Fonctionnement d'un moteur pas-a-pas unipolaire

➢ 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 II.2: Table Etat successifs des phases

➢ 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 l'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 suivent :

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

Le mouvement s'effectué à la suite d'une inversion du champ magnétique en alimentant l'une ou l'autre des phases d'un bobinage a point milieu ; seule une moitié du bobinage et donc utilisée a 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.

II.2.3.2.2. Moteur à aimant permanent bipolaire :

La Figure I.8 représente la constitution interne d'un moteur de type bipolaire. Ce type de moteur nécessite une commande plus complexe que celle du moteur unipolaire, le courant devant changer de sens dans les enroulements à chaque pas effectué.[14]

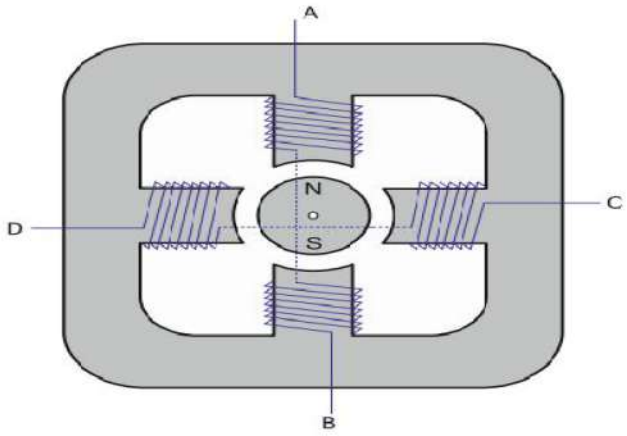


Figure II.5 : Représentation schématique d'un moteur bipolaire

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

- **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.

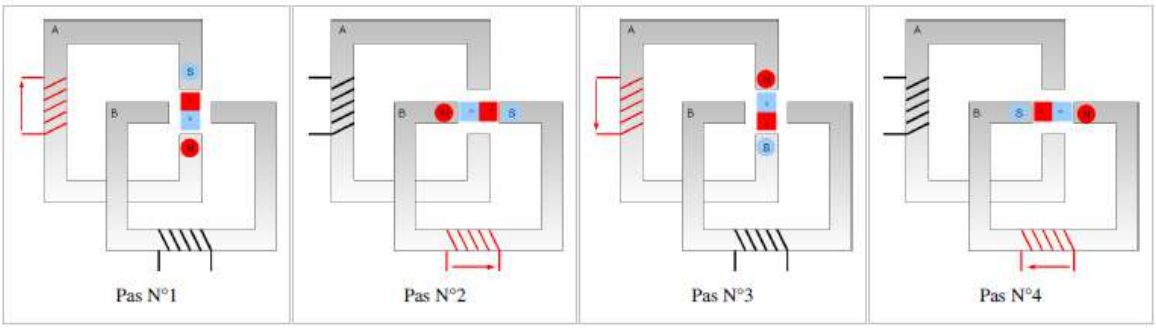


Figure II.6 : Fonctionnement à pas complet

➢ 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.

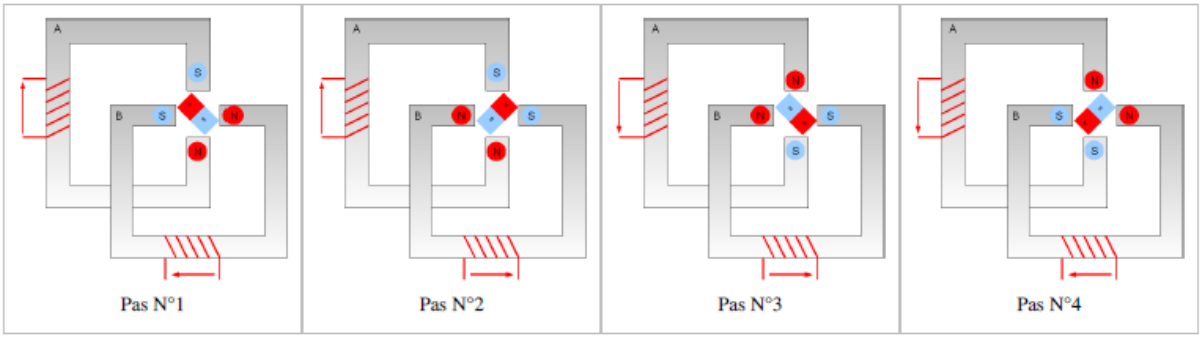


Figure II.7 : Fonctionnement avec couple maximal

➢ Fonctionnement à demi-pas

La troisième option est un mélange des deux première 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 affectivement un 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.

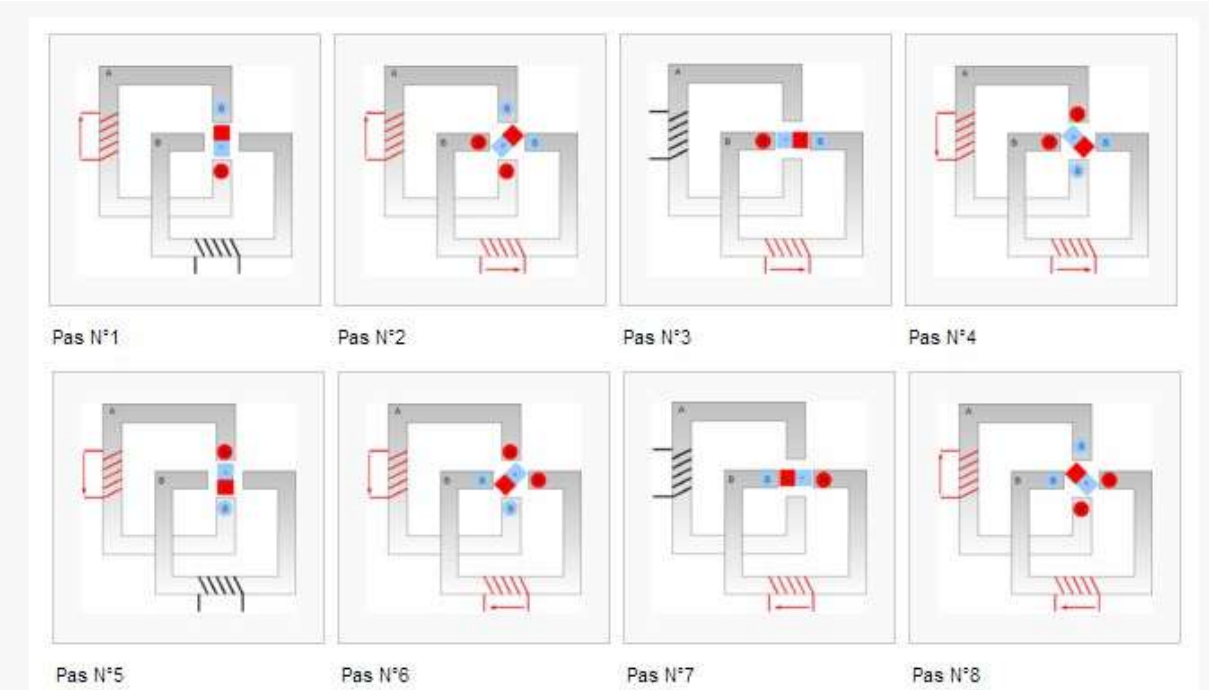


Figure II.8 : Fonctionnement à demi-pas

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 restant, quant à eux, inchangés.

II.2.3. 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 [15].

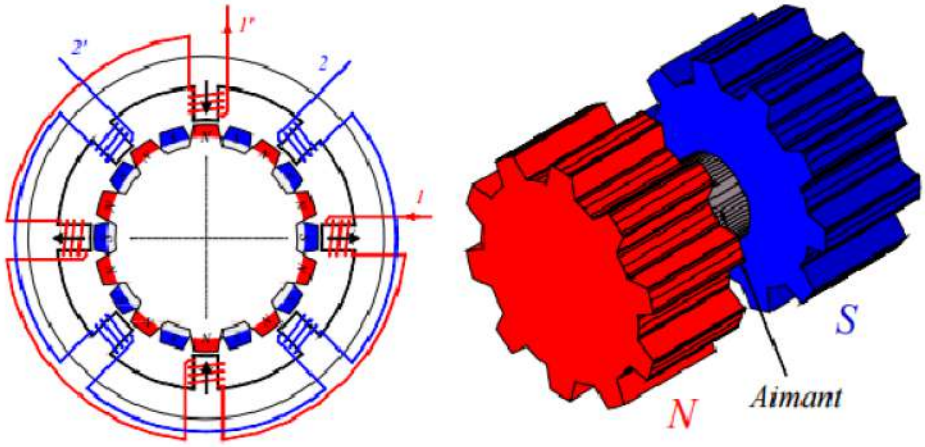


Figure II.9 : Moteur hybride (MH)

➢ 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 ½ τdr [15].

II.2.4. Comparaison entre les différents types des moteurs pas à pas

Cette comparaison peut se résumer selon le tableau suivant [16]:

Type de moteur pas à pas	Moteur à réluctance variable	Moteur à aimants Permanents	Moteur hybride
Résolution (nb de pas par tour)	Bonne	Moyenne	Elevée
Couple moteur	Faible	Elevée	Elevée
Sens de rotation	Dépend : - de l'ordre d'alimentation des phases	Dépend : - de l'ordre d'alimentation des phases	Dépend : - de l'ordre d'alimentation des phases
Fréquence de travail	Grande	Faible	Grande
Puissance	Quelques Watts	Quelques dizaines de Watts	Quelques KWatts
Inconvénients	Pas de mémoire de position		

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

II.2.5.Avantages et inconvénients des moteurs pas à pas

Les caractéristiques de ces moteurs sont particulièrement intéressantes, Le Tableau II.5 Résume les avantages et les inconvénients de ces actionneurs [16]:

Avantages	Inconvénients
▪ Asservissement de position ou de vitesse en boucle ouverte ▪ Fort couple à basse vitesse ▪ Simplicité de mise en œuvre ▪ Positionnement statique ▪ Fiabilité ▪ Faible prix	▪ Positionnement discret ▪ Faible vitesse maximale ▪ Faible puissance ▪ Bruyant, source d'oscillations ▪ Faible rendement

Table II.5: les avantages et les Inconvénients des moteurs pas à pas

II.3. Définition du module Arduino

Le module Arduino est un circuit imprimé en matériel libre (plateforme de contrôle) dont les plans de la carte elle-même sont publiés en licence libre dont certains composants de la carte : comme le microcontrôleur et les composants complémentaires qui ne sont pas en licence libre. Un microcontrôleur programmé peut analyser et produire des signaux électriques de manière à effectuer des tâches très diverses. Arduino est utilisé dans beaucoup d'applications comme l'électrotechnique industrielle et embarquée ; le modélisme, la domotique mais aussi dans des domaines différents comme l'art contemporain et le pilotage d'un robot, commande des moteurs et faire des jeux de lumières, communiquer avec l'ordinateur, commander des appareils mobiles (modélisme). Chaque module d'Arduino
Possède un régulateur de tension +5 V et un oscillateur à quartez 16 MHz (ou un résonateur céramique dans certains modèles). Pour programmer cette carte, on utilise l'logiciel IDE Arduino [17].

II.3.1.Les différents types des cartes 'Arduino' :

Il y a trois types de cartes :

- Les « officielles » qui sont fabriquées en Italie par le fabricant officiel : Smart Projets.
- Les « compatibles » qui ne sont pas fabriqués par Smart Projets, mais qui sont totalement compatibles avec les Arduino officielles.
- Les « autres »fabriquées par diverse entreprise et commercialisées sous un nom différent (Freeduino, Seeduino, Femtoduino, ...) [18].

II.3.1.1.La carte Arduino Méga

La carte Arduino Méga est une autre carte qui offre toutes les fonctionnalités de la carte précédente, mais avec des fonctionnalités supplémentaires. On retrouve notamment un nombre d'entrées et de sorties plus important ainsi que plusieurs liaisons séries. Bien sûr, le prix est plus élevé.

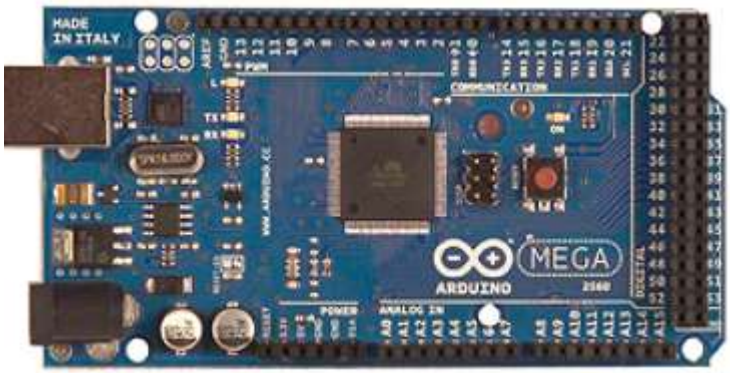


Figure II.10: carte Arduino "Méga"

II.3.1.2.La carte Arduino Uno et Duemilanove

Nous choisirons d'utiliser la carte portant le nom de « Uno » ou « Duemilanove ». Ces deux versions sont presque identiques.



Figure II.11: Carte Arduino “Duemilanove” et “Uno”

II.3.2.Les gammes de la carte Arduino

Actuellement, il existe plus de 20 versions de module Arduino, nous citons quelques un afin d’éclaircir l’évaluation de ce produit scientifique et académique:

- Le NG d'Arduino, avec une interface d'USB pour programmer et usage d'un ATmega8.
- L'extrémité d'Arduino, avec une interface d'USB pour programmer et usage d'un Microcontrôleur ATmega8.
- L'Arduino Mini, une version miniature de l'Arduino en utilisant un microcontrôleur ATmega168.
- L'Arduino Nano, une petite carte programme à l’aide porte USB cette version utilisant un microcontrôleur ATmega168 (ATmega328 pour une plus nouvelle version).

- Le Lily Pad Arduino, une conception de minimaliste pour l'application wearable autour du microcontrôleur ATmega168.
- Le NG d'Arduino plus, possédant une interface USB pour la programmation autour du microcontrôleur ATmega168.
- L'Arduino Bluetooth, possédant une interface de Bluetooth pour la programmation autour du microcontrôleur ATmega168.
- L'Arduino Diecimila : interface USB, un microcontrôleur ATmega168.
- L'Arduino Duemilanove ("2009"), en utilisant un microcontrôleur l'ATmega168 (ATmega328 pour une plus nouvelle version) et actionné par l'intermédiaire de puissance d'USB/DC.
- L'Arduino Méga, en utilisant un microcontrôleur ATmega1280 pour les E/S additionnel et mémoire.
- L'Arduino UNO : microcontrôleur ATmega328.
- L'Arduino Mega2560 : un microcontrôleur ATmega2560, mémoire de 256 KBS. Elle incorpore également le nouvel ATmega8U2.
- L'Arduino Leonardo : ATmega3U4 qui élimine le besoin de raccordement d'USB et peut être employé comme clavier.
- L'Arduino Esplora : ressemblant à un contrôleur visuel de jeu, avec un manche et des sondes intégrées pour le bruit, la lumière, la température, et l'accélération. [19]

Parmi ces types, nous avons choisi la carte Arduino UNO (carte Basique) qui permet la mise en œuvre d'une telle commande et qui sera détaillée par la suite.

L'Arduino fournit un environnement de développement s'appuyant sur des outils open Source comme interface de programmation. L'injection du programme déjà converti par l'environnement sous forme d'un code « HEX » dans la mémoire du microcontrôleur se fait d'une façon très simple par la liaison USB. En outre, des bibliothèques de fonctions "clé en main" sont également fournies pour l'exploitation d'entrées-sorties. Cette carte est basée sur un microcontrôleur ATmega 328 et des composants complémentaires. La carte Arduino contient une mémoire morte de 1 kilo. Elle est dotée de 14 entrées/sorties digitales (dont 6 peuvent être utilisées en tant que sortie PWM), 6 entrées analogiques et un cristal a 16 MHz, une connexion USB et Possède un bouton de remise à zéro et une prise jack d'alimentation.
La carte est illustrée dans la figure si dessous [20].

II.3.3.Caractéristiques sélectifs de la carte Arduino UNO

Il y a de nombreuses cartes électroniques qui possèdent des plateformes basées sur des microcontrôleurs disponibles pour l'électronique programmée. Tous ces outils prennent en charge les détails compliqués de la programmation et les intègrent dans une présentation facile à utiliser. De la même façon, le système Arduino simplifie la façon de travailler avec les Microcontrôleurs tout en offrant à personnes intéressées plusieurs avantages cités comme suit:

- **Le prix (réduits) :** les cartes Arduino sont relativement peu coûteuses Comparativement aux autres plates-formes. La moins chère des versions du module Arduino peut être assemblée à la main, (les cartes Arduino pré-assemblées coûtent Moins de 2500 Dinars).
- **Multi plateforme :** le logiciel Arduino, écrit en JAVA, tourne sous les systèmes d'exploitation Windows, Macintosh et Linux. La plupart des systèmes à micro contrôleurs sont limités à Windows.
- **Un environnement de programmation clair et simple :** l'environnement de programmation Arduino (le logiciel Arduino IDE) est facile à utiliser pour les débutants, tout en étant assez flexible pour que les utilisateurs avancés puissent en tirer profit également.
- **Logiciel Open Source et extensible:** le logiciel Arduino et le langage Arduino sont publiés sous licence open source, disponible pour être complété par des programmateurs expérimentés. Le logiciel de programmation des modules Arduino est une application JAVA multi plateformes (fonctionnant sur tout système d'exploitation), servant d'éditeur de code et de compilateur, et qui peut transférer le programme au travers de la liaison série (RS232, Bluetooth ou USB selon le module).
- **Matériel Open source et extensible:** les cartes Arduino sont basées sur les Microcontrôleurs Atmel ATMEGA8, ATMEGA168, ATMEGA 328, les schémas des modules sont publiés sous une licence créative Commons, et les concepteurs des circuits expérimentés peuvent réaliser leur propre version des cartes Arduino, en les complétant et en les améliorant. Même les utilisateurs relativement inexpérimentés peuvent fabriquer la version sur plaque d'essai de la carte Arduino, dont le but est de comprendre comment elle fonctionne pour économiser le coût.[21]

II.3.4. 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. [22]

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.3.4.1. 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 [23].

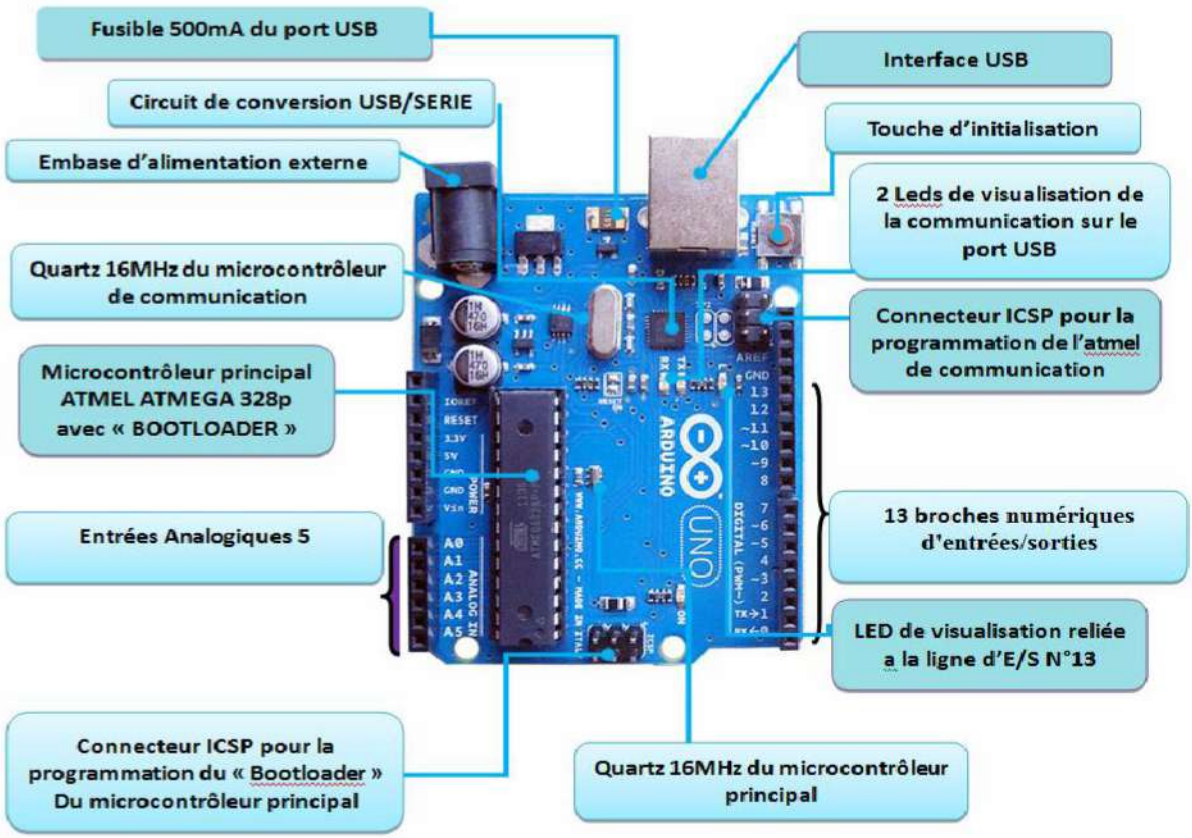


Figure II.12 : Module Arduino Atmel Uno

La carte "ARDUNO UNO" que nous allons utiliser dispose de 14 broches ("pin" en anglais, numérotées de 0 à 13) qui peuvent être configurées en "entrées digitales" ou en "sorties digitales" susceptibles de délivrer une intensité maximale de 40mA sous une tension égale à 0V ou 5V. Certaines de ces broches peuvent être configurées en "sorties PWM" (Pulse Width Modulation ou modulation de largeur d'impulsion).

Elle possède également 6 entrées analogiques (notées A0 à A5) permettant de mesurer des tensions comprises entre 0V et 5V grâce à un convertisseur A/N à 10 bits.

Les broches pin0 et pin1 ne sont pas disponibles pour l'utilisateur de la carte car elles sont utilisées pour communiquer avec l'ordinateur via le port USB, et sont notées RX (réception) et TX (transmission) sur la carte

Le port USB de la carte ARDUINO est géré par l'ordinateur comme un port série virtuel (COM) et il est nécessaire d'installer les drivers correspondants avant d'établir la liaison ordinateur-Arduino.

II.3.4. 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).[24]

Le microcontrôleur est préprogrammé avec un boot loader de façon à ce qu'un programmateur 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 [25].

A) Les sources de l'alimentation de la carte :

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

- **Vin**. La tension d'entrée positive lorsque la carte Arduino est utilisée avec une source de tension externe (à distinguer du 5V de la connexion USB ou autre source 5Vrégulée). On peut alimenter la carte à l'aide de cette broche, ou, si l'alimentation est fournie par le jack d'alimentation, accéder à la tension d'alimentation sur cette broche.
- **5V.** La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (pour info : les circuits électroniques numériques nécessitent une tension d'alimentation parfaitement stable dite "tension régulée" obtenue à l'aide d'un composant appelé un régulateur et qui est intégré à la carte Arduino).le 5Vrégulé 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 5Vrégulé)ou de toute autre source d'alimentation régulée.
- **3V3**. Une alimentation de 3.3V fournie par le circuit intégré FTDI (circuit intégré faisant l'adaptation du signal entre le port USB de votre ordinateur et le port série de l'ATmega) de la carte est disponible : ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5V. L'intensité maximale disponible sur cette broche est de 50mA. [19]

B) Les entrées analogiques

Le second connecteur du bas comporte les légendes A0 à A6. Ces broches d'entrée permettent de mesurer une tension pour pouvoir en utiliser les valeurs dans les programmes. Précisons que les broches servent à mesurer une tension et non un courant.

Le circuit d'entrée auquel chacune des broches donne accès possède une résistance interne (impédance) très importante, ce qui fait que le courant passant par la broche est toujours très faible. Ces entrées sont désignées comme étant de type analogique, et elles le sont par défaut, mais on peut également les exploiter en tant qu'entrées numériques ou même comme sorties numériques [26].

C) Connexions numériques

Passons maintenant aux deux connecteurs du haut de la carte, et commençons par celui de droite. Les légendes désignent des entrées numériques de 0 à 13 en partant de la droite (Digital). Ces broches peuvent servir d'entrées ou de sorties.

Lorsqu'on les utilisé en tant que sorties, elles se comportent comme les broches de source de tension présentées plus haut, sauf qu'elles fonctionnent toutes en 5 V et qu'elles peuvent être activées et désactivées depuis un programme. Si on active une de ces broches, la tension présente sera à 5 V. Si on désactive la broche, la tension sera à 0 V. Comme pour les broches de tension, il faut veiller à ne jamais dépasser le courant maximal qu'elles peuvent supporter. Les deux premières broches à droite (0 et 1) portent une légende supplémentaire (RX et TX) pour la transmission et la réception. Ces deux broches sont réservées à la communication ; elles sont, d'une façon indirecte, les connexions en émission et en réception de la connexion USB vers votre ordinateur.

Toutes les broches numériques peuvent supporter jusqu'à 40 mA (milliAmpères) à 5 V. c'est largement suffisant pour alimenter une LED standard, mais assurément pas pour alimenter directement un moteur électrique [26].

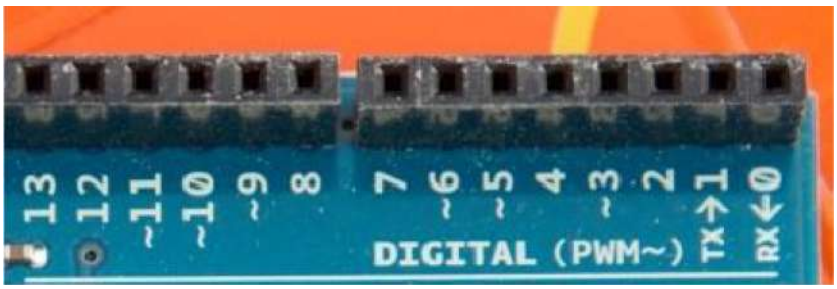


Figure II.13: Entrées/sorties numériques D0 à D13

II.3.4.2.1. 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 successives, 10bits, temps de conversion 13 a 260µs.
- Référence de tension interne : 1.1 ou 2.56 V.

II.3.4.2.2. Visualisation

Les trois "points blancs" entoures en rouge sont des LED dont la taille est de l'ordre du Millimètre. Ces LED servent à deux choses :

- Celle tout en haut du cadre : elle est connectée à une broche du microcontrôleur et va servir pour tester le matériel.

Nota : Quand on branche la carte au PC, elle clignote quelques secondes.

- Les deux LED du bas du cadre : servent a visualiser l'activite sur la voie série (une pour l'émission et l'autre pour la réception). Le téléchargement du programme dans le microcontrôleur se faisant par cette voie, on peut les voir clignoter lors du chargement.

II.3.4.3. 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 maitrise 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 [22].

a) Logiciel d'édition et programmation Arduino



Figure II.14 : 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 librairies. Après l'indispensable installation des pilotes, lancez le logiciel à partir du dossier. La fenêtre de l'application Arduino comporte les éléments suivants.

b) Présentation des parties principales du logiciel



Figure II.15: Présentation des parties principales du logiciel

c) Les boutons :

Voyons à présent à quoi servent les boutons, encadrés en rouge et numérotés par le par le Chiffre.



Figure II.16: présentation des boutons de la carte arduino

Bouton 1: Ce bouton permet de vérifier le programme, il actionne un module qui cherche les erreurs dans votre programme.

Bouton 2 : Charge (téléversé) le programme dans la carte Arduino

Bouton 3: Crée un nouveau fichier.

Bouton 4: Ouvre un fichier.

Bouton 5: Enregistre le fichier.

Bouton 6 : Ouvre le moniteur série (on verra plus tard ce que c'est Enfin, on va pouvoir s'occuper du matériel que vous devriez tous posséder en ce moment même : la carte Arduino.

La mise en marche :

Avant de commencer toute manipulation, il faut passer par les points suivants :

➢ Choix de la carte :

Le choix d'une carte se fait à partir de l'onglet **Outils** comme il est montré dans la figure suivante :

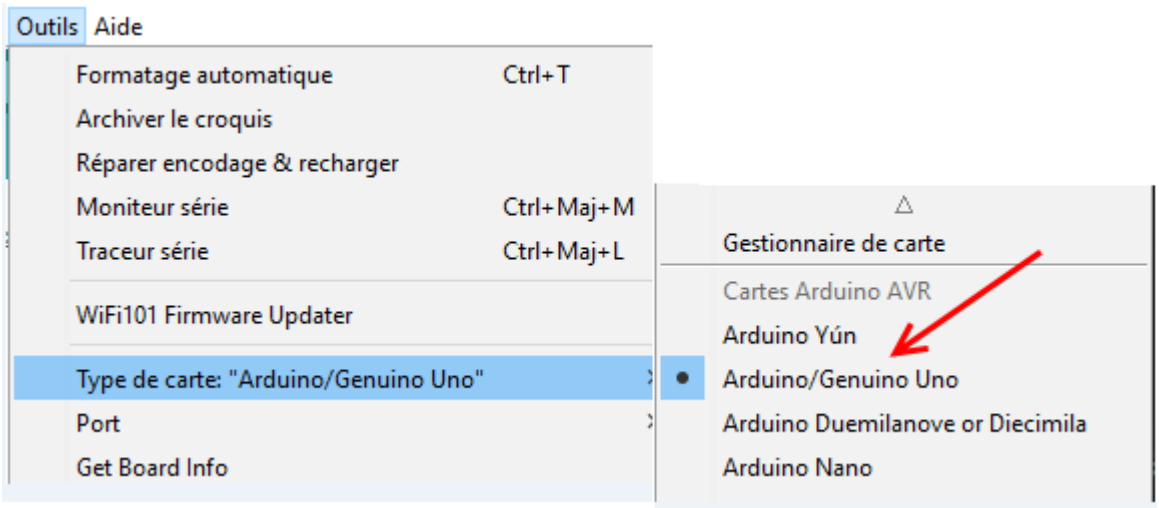


Figure II.17 : choix d'une carte

➢ Choix du port de connexion :

Cette étape consiste a sélectionné le port série active qui permettra la liaison entre le Pc et la carte suivant la configuration du Pc comme il est montré dans la figure suivante:

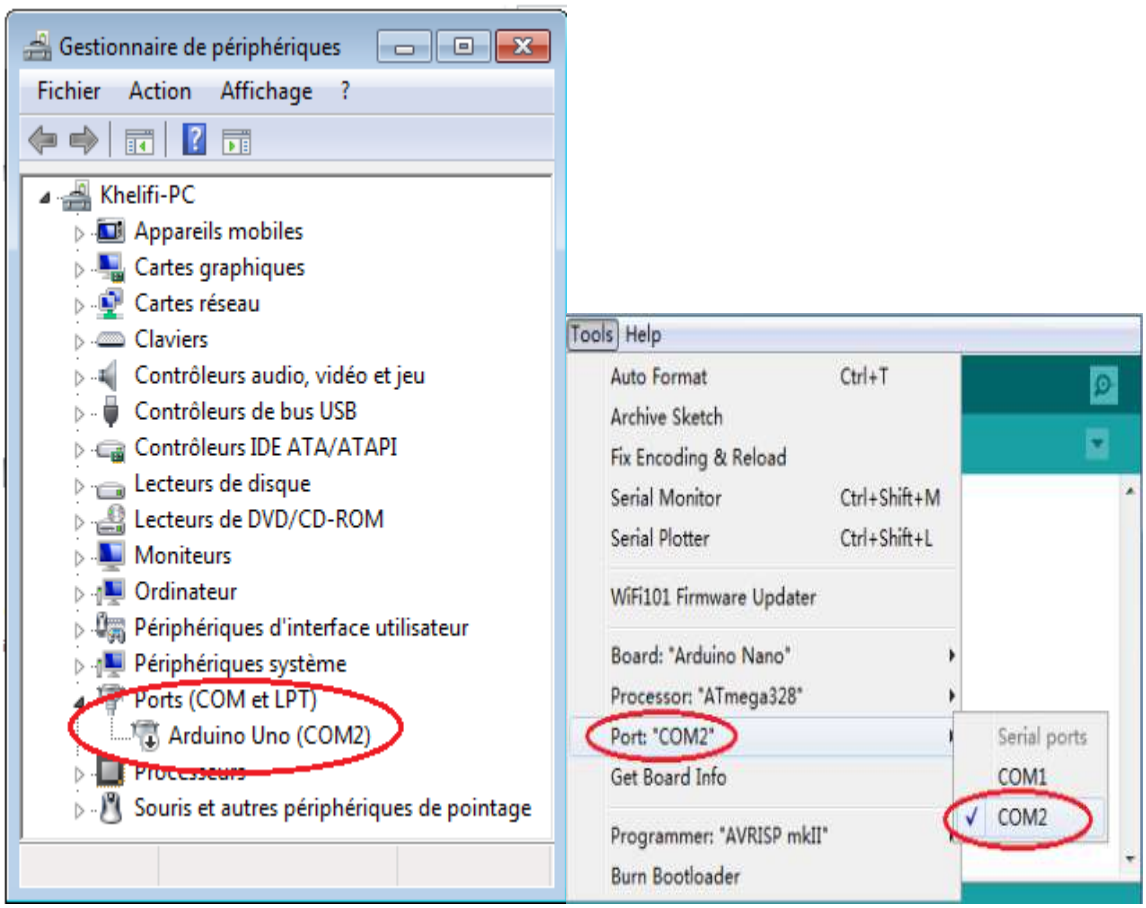


Figure II.18: Choix du port de connexion de la carte

➢ Test de la carte :

On teste maintenant le matériel en chargeant un petit programme existant (p.ex. : clignoter une LED) on passant par les points suivants :

- Ouverture d'un exemple :

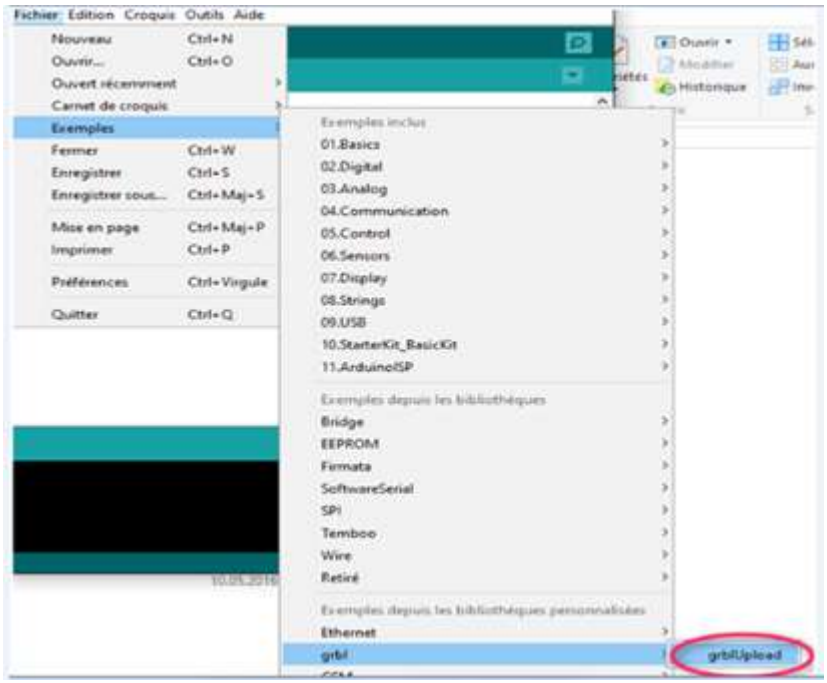


Figure II.19 : ouverture d'un programme

- Compilation du programme

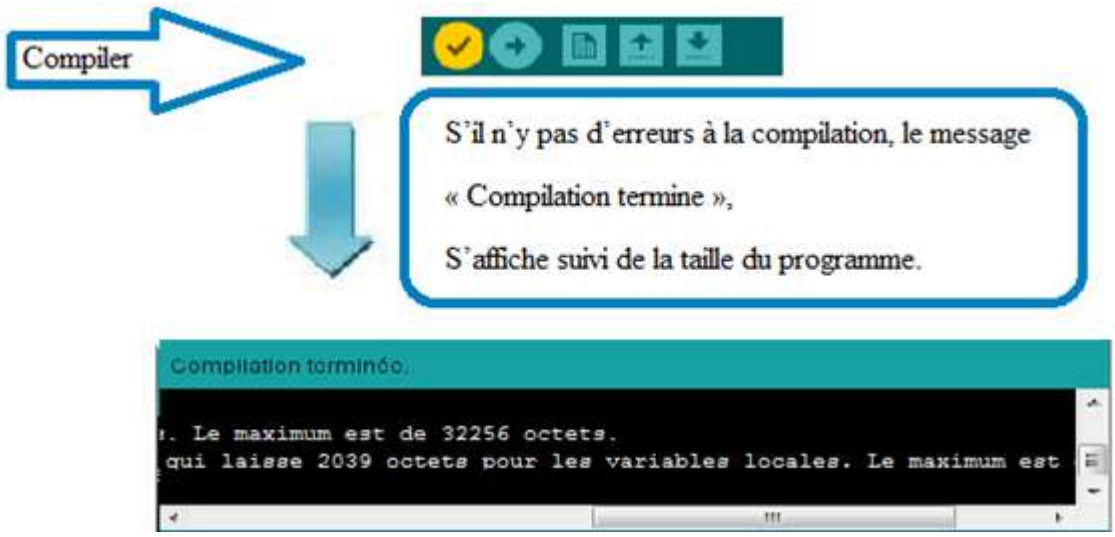


Figure II.20 : Compilation du programme

- Téléchargement du programme dans la carte :



Figure II.21 : Téléchargement

En bas dans l'image, vous voyez le texte : " compilation du croquis" cela signifie que le logiciel est en train d'envoyer le programme dans la carte. Une fois qu'il a fini, il affiche un autre message :

II.4.Les drivers :

Les circuits intégrés L297 ET L298 sont les composants les plus utilisées lorsqu'il s'agit de piloter des moteurs pas à pas. Ils permettent en effet de commander n'importe quel moteur, qu'il soit unipolaire ou bipolaire. Ils peuvent alimenter des organes consommant un courant compris entre quelques dizaines de milliampères et plusieurs ampères. Ce sont donc des composants standards par excellence car ils ont été conçus afin d'être utilisées ensemble, mais ils peuvent également être inclus dans un quelconque montage, et ce, séparément.

II.4.1. 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 élevée. Il peu 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. [27]

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 .Chaque moteur reçoit une séquence indépendamment de l'autre à travers un amplificateur L298 pouvant fournir un courant de sortie de 0,5A.

- Le circuit EASY DRIVER.

II.4.2. Le Circuit intégré L297

Le L297 est un circuit qui contient toute le circuiterie de commande et de contrôle de moteurs pas à pas unipolaire et bipolaire. Utilisé conjointement avec un driver double point tel que le L298, l'ensemble forme une interface idéale pour le contrôle d'un moteur pas à pas bipolaire par microprocesseur ou micro contrôleur. Il est possible de contrôler un moteur pas à pas unipolaire avec un L297 en lui adjoignant quatre transistors Darlington ou mosfet.

Le circuit de commande de moteur pas à pas L297 est initialement prévu pour être avec un driver L298 ou L293E. Il reçoit les signaux de contrôle en provenance d'un système à base de microprocesseur et génère tous les signaux nécessaires pour l'étage de puissance.

Ce circuit inclue deux régulations à découpage type PWM (modulation de largeur d'impulsion) pour réguler le courant des enroulements moteur. Avec un étage de puissance adapté le L297commande un moteur pas à pas bipolaire à aimant permanent, un moteur unipolaire à quatre phases à aimant permanent ou un moteur à reluctance variable à quatre phases. De plus il est capable de gérer le mode demi pas et pas complet. Le L297A est une version spéciale pour le positionnement des têtes de lecteur de disquette. [10]

Le circuit L297 est disponible en boîtier DIP plastique 20 pins. Il s'alimente en 5V et Le circuit L297 est disponible en boîtier DIP plastique 20 pins. Il s'alimente en 5V et les lignes d'entrées sont compatibles TTL, CMOS et s'interface aussi avec des transistors en collecteur ouvert. La haute densité d'intégration du L297 est l'une des caractéristiques technologique clé de ce circuit.

II.4.2.1 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 II.22). L'ensemble forme une interface idéale pour le contrôle d'un moteur pas à pas par Arduino

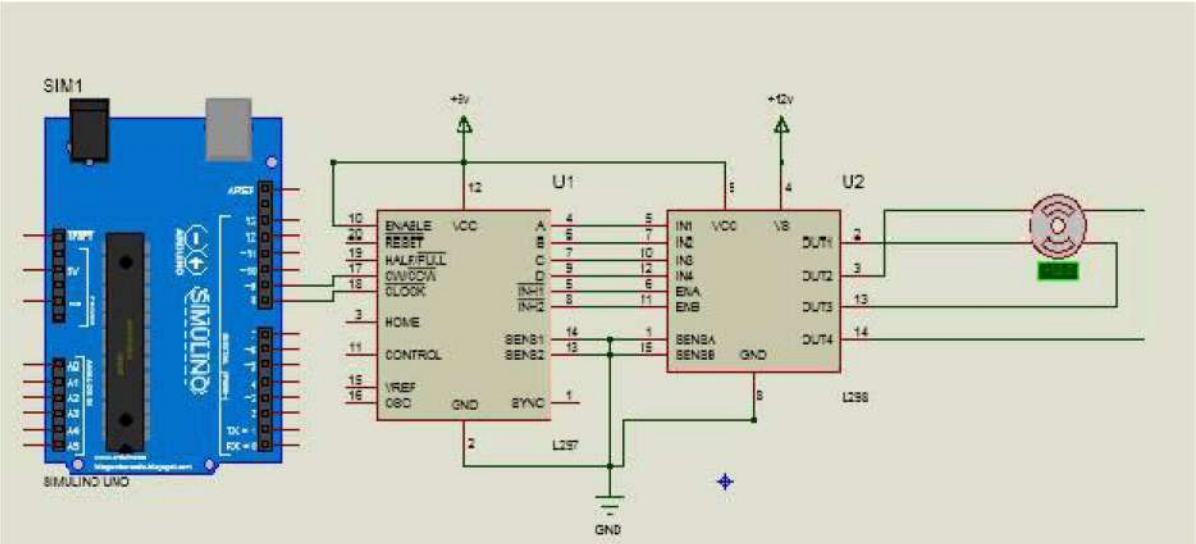


Figure II.22: commande moteur par le couple L297 et L298 arduino.

II.4.3. 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.4.3.1 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.4.3.2 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.

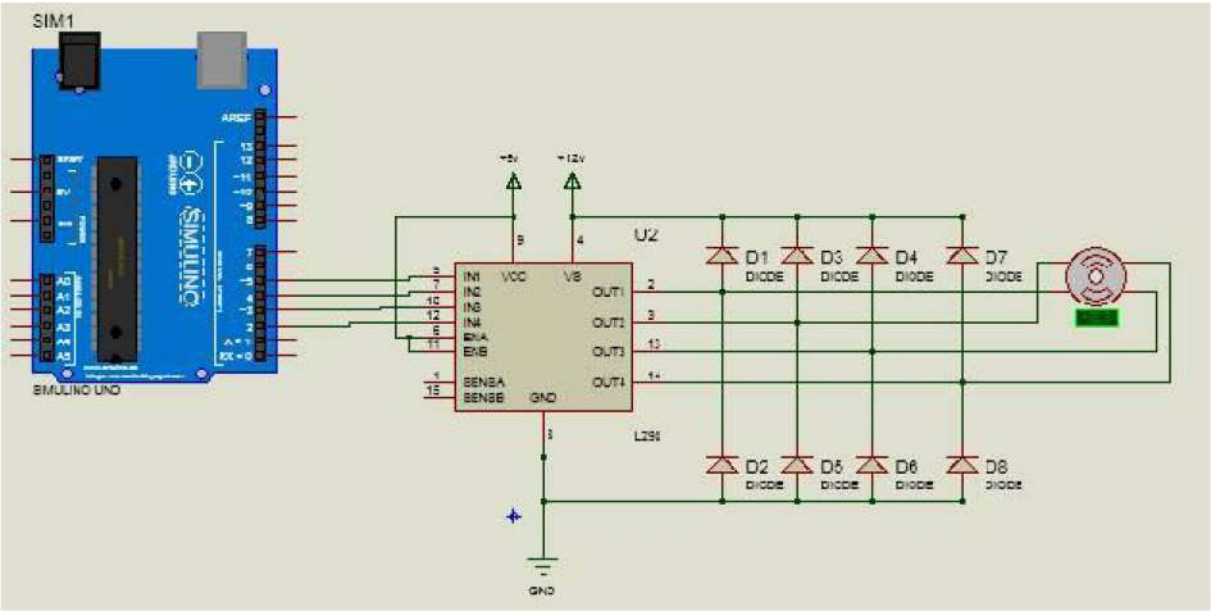


Figure II.23: commande moteur par le circuit L298.

II.4.3.3 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 proches :

- 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.

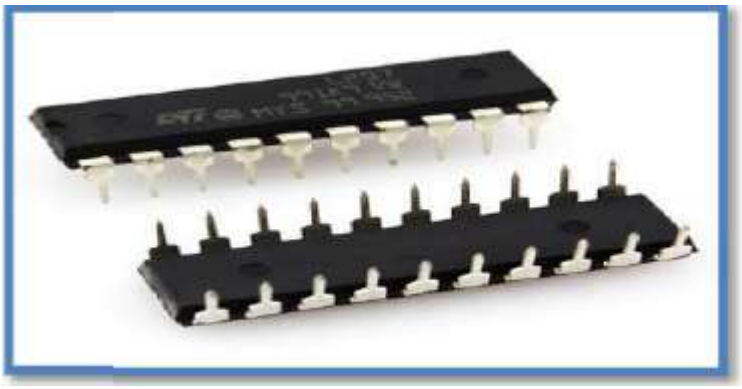


Figure II .24 Le circuit L297

Constitution du L 297

- Le séquenceur

Le cœur du L297 est le block appelé "translator" sur le schéma-block. C'est lui qui génère Les séquences appropriées pour l'alimentation des phases en mode demi-pas, pas entier Une phase/Biphasé.

II.5. Conclusion

Ce chapitre est un portail pour notre réalisation et conception matérielle. Ce travail nous a permis de maîtriser la commande.

La 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 suivi par les options de notre carte ArduinoUno avec circuit de commande et ses caractéristiques afin de les exploiter d'une manière correcte. Nous avons présenté également des principes et description logicielle, 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.

CHAPITRE III

Logiciels du pilotage du CNC

III.1. Introduction

Trois logiciels sont utilisés pour piloter notre machine CNC à savoir :

-Le logiciel Inkspace : c'est un logiciel de dessin et d'écriture permettant la génération le **code G** d'une image.

- **Universel G code Sender** : il permet la conversion des coordonnées cartésiennes (X, Y, Z) en **code G** et de les envoyer manuellement l'un après l'autre vers le port série ou en série sous forme d'un fichier.

- **Interpréteurs Grbl :** ce programme est charge au préalable dans la mémoire de la carte arduino afin d'interprété les instructions du **code G** et les transformés à des commandes sous forme signaux numériques.

III.2. Définition Inkscape

Inkscape est un logiciel puissant 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 intéresse 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 [28].

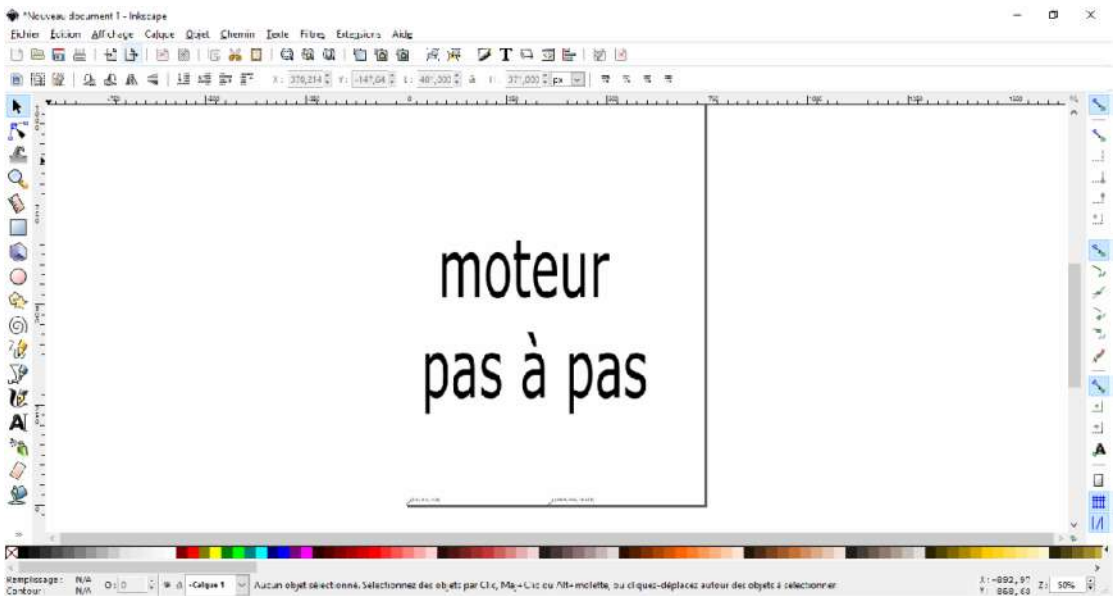


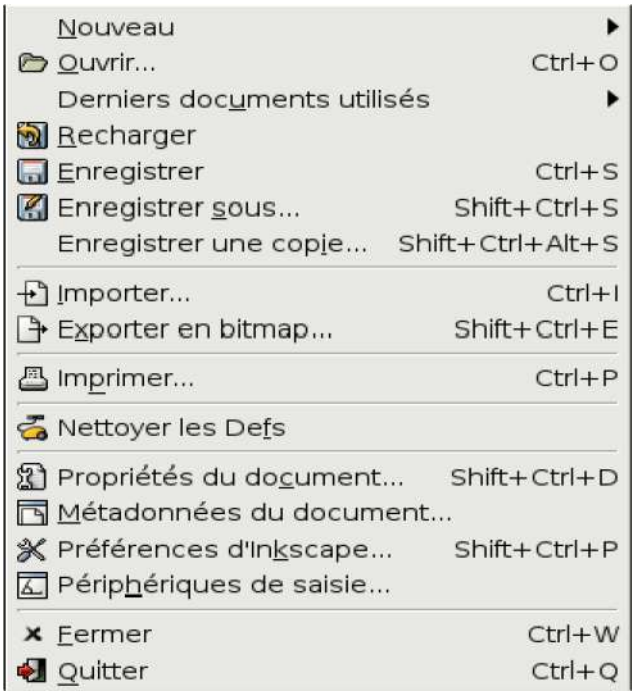
Figure.III.1 l'interface principale d'Inkscape

III.2.1 La fenêtre Inkscape

L'interface d'Inkscape est constituée d'un ensemble d'éléments originaux permettant un travail simple, harmonieux et contextuel. Elle est constituée principalement de la fenêtre de document dans laquelle le dessin peut être créé. Cette fenêtre contient elle-même plusieurs zones qu'il est important de bien identifier pour se repérer aisément dans l'application.
On peut la diviser en six grandes zones.

- le menu situé tout en haut;
- la barre des commandes;
- la barre de contrôles des outils;
- les boutons de la boîte à outils;
- la palette
- la barre d'état et d'information.

➢ **Le menu**



Comme dans la plupart des applications GTK, le menu de Inkscape contient les fonctions essentielles du logiciel, celles qui s'appliquent à l'application elle-même : nouveau, ouvrir, enregistrer, exporter, quitter. Les fonctions relatives à la manipulation du document y sont aussi présentes. Une bonne partie de ce manuel constitue la référence des fonctions disponibles dans ces menus.

- **La barre de commande**



La barre de commande comporte un certain nombre d'icônes qui sont autant de raccourcis autrement accessibles dans les divers menus de l'application. Signe de complémentarité, elle est placée sous le menu lui-même. Il est possible : d'ouvrir un nouveau document depuis un modèle, d'ouvrir un document existant, d'enregistrer, d'imprimer, d'importer un bitmap d'exporter en bitmap (PNG), d'annuler la dernière action, de refaire la dernière action annulée, de copier vers le presse-papier, de couper vers le presse-papier, et de coller du presse-papier à Inkscape, d'ajuster la sélection à la fenêtre, d'ajuster le dessin à la fenêtre, d'ajuster la page à la fenêtre, de dupliquer l'objet sélectionné, de créer un clone, de couper le lien entre le clone et son original, de grouper des objets, de dégrouper des objets, d'ouvrir la fenêtre qui édite les styles des objets, d'ouvrir la fenêtre qui édite le texte, de voir et d'éditer l'arbre du code XML, d'ouvrir la fenêtre d'alignement et de distribution des objets, d'éditer les préférences du logiciels, d'éditer les préférences du document.

- **Les outils**



La boîte à outils est un élément primordial d'inkscape. Elle contient la quasi-totalité des fonctions de dessin disponibles dans le logiciel, en particulier la création de forme. Les boutons de la boîte à outils sont regroupés en vertical sur la gauche de la fenêtre de document. Ils donnent pour la plupart accès à des fonctions de dessins de formes géométriques ou libres et directement applicables à l'aide de la souris, ou de la palette sur le canevas (le canevas est l'espace blanc) représentant le support du dessin. Certains outils sélectionnés peuvent être paramétrés à l'aide d'options qui apparaissent dans une barre placée sous la barre de commandes et nommée barre d'options.

- **La barre de contrôle des outils ou les options des outils**



Dans ce tutoriel nous allons aborder des notions simples du logiciel Inkscape.

L'objectif défini est le suivant : créer un fond de page personnalisé pour réaliser une carte de visite originale.

III.2.2 Création du document

L'ouverture du document propose une interface conviviale et intuitive et vous propose Un accès aux barres d'outils sur trois côtes de la fenêtre.

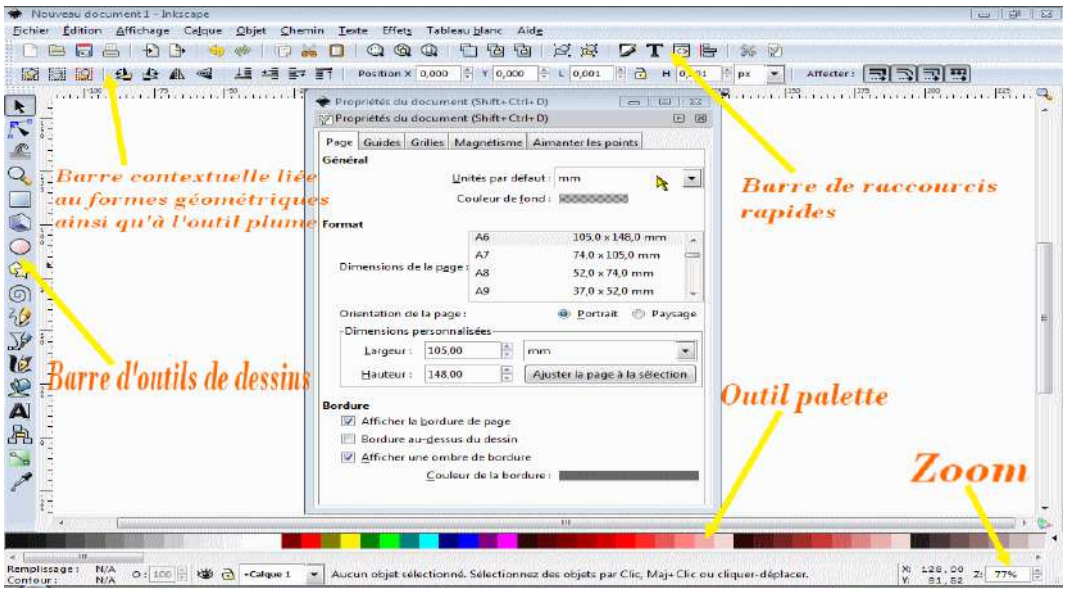


Figure.III.2 : Création du document.

A. Paramétrer le format souhaité

La première étape consiste à formater la page de travail comme suite: **F**ichier / Propriété du document, ce qui provoque l'ouverture de la fenêtre dialogue ci-dessous **:**

- dans l'onglet Page**,** cocher les unités souhaitées :
- dans ce cas précis **mm**
- format de page **A6.**

Une fois le formatage terminé, deux choix sont envisageables :

- Soit l'image est dessinée manuellement.
- Soit l'image est importée comme il est montré dans la figure suivante :

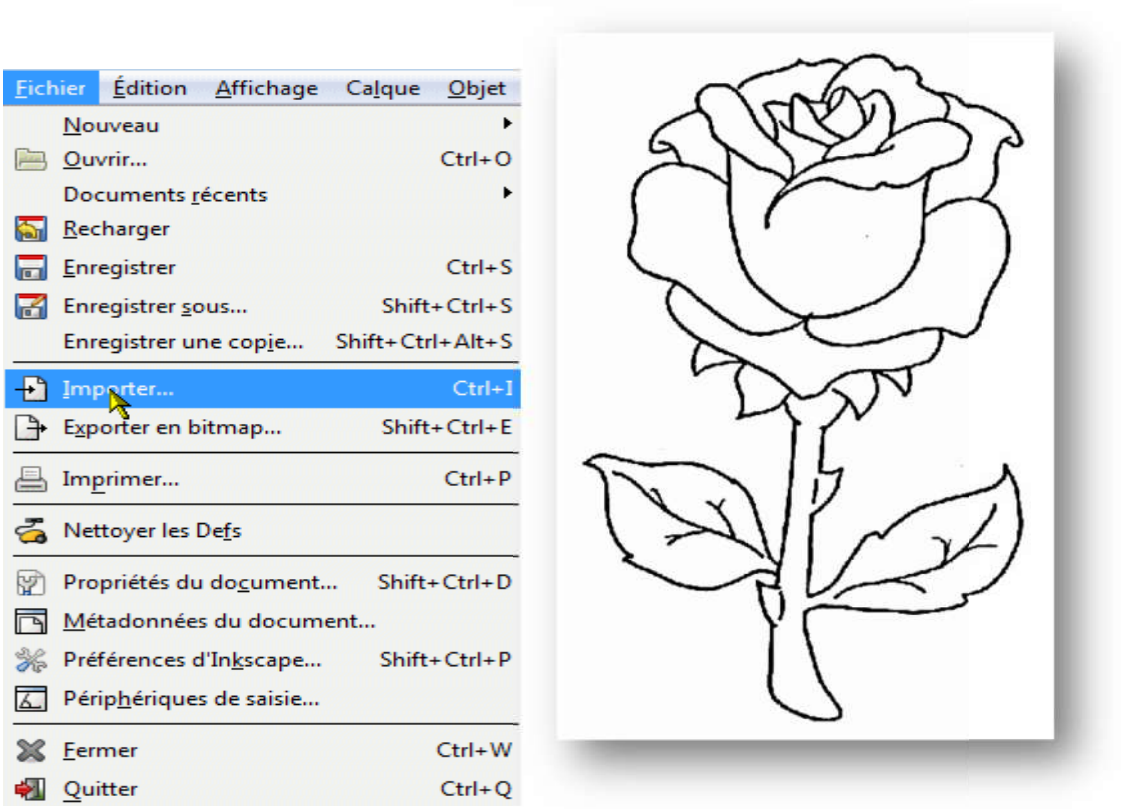


Figure III.3: Importation d'une image bitmap

B. Vectorisation d'une image bitmap :

Cette étape permet de vectorisé une image bitmap :

- Chemin =>Vectoriser puis valider

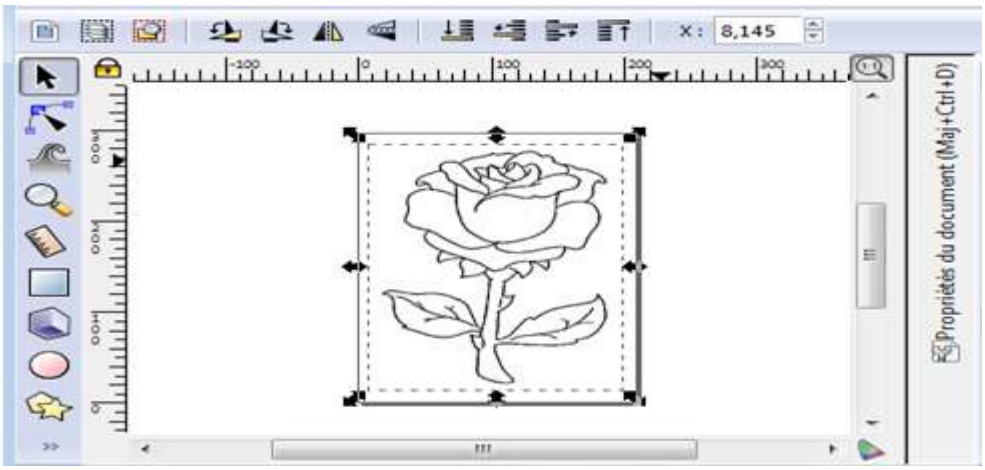


Figure III.4 : Vectorisation d'une image bitmap

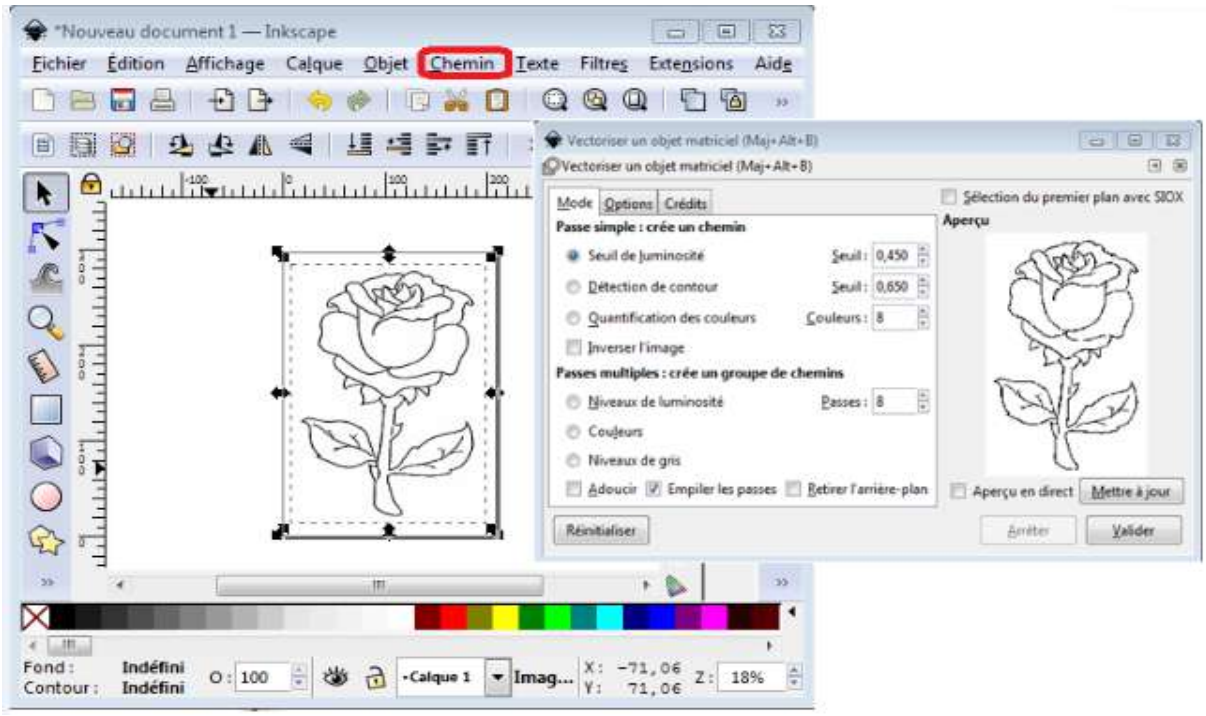


Figure III.5:image vectoriel à vocation "artistique

Une fois valider, on aura deux images superposés (l'image bitmap et l'image vectorielle).On les sépare par un simple "glissez déposé" (Vous pouvez zoomer en gros plan pour apprécier la différence entre une image vectorielle et une image bitmap).

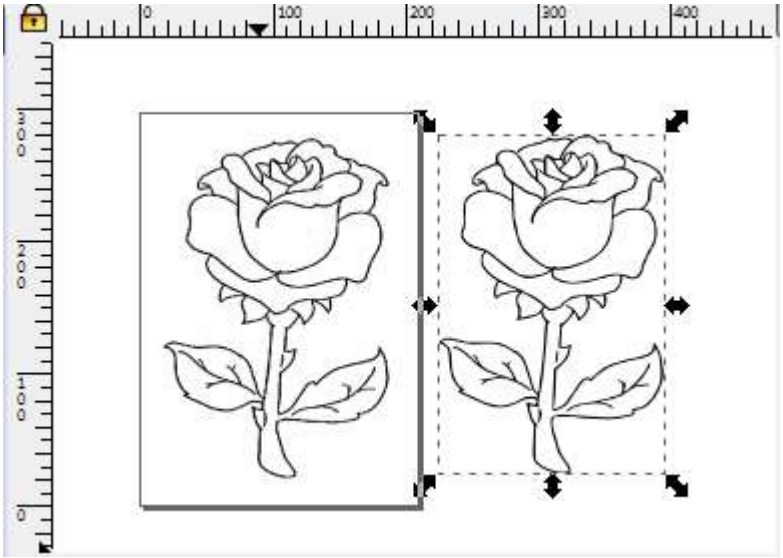


Figure III.6: Image vectorielle et image bitmap.

C. Génération du Gcode

On clique sur la barre de menu **Extensions** puis sur **Gcodetools**, ensuite, nous choisissons de bibliothèques d'outils.

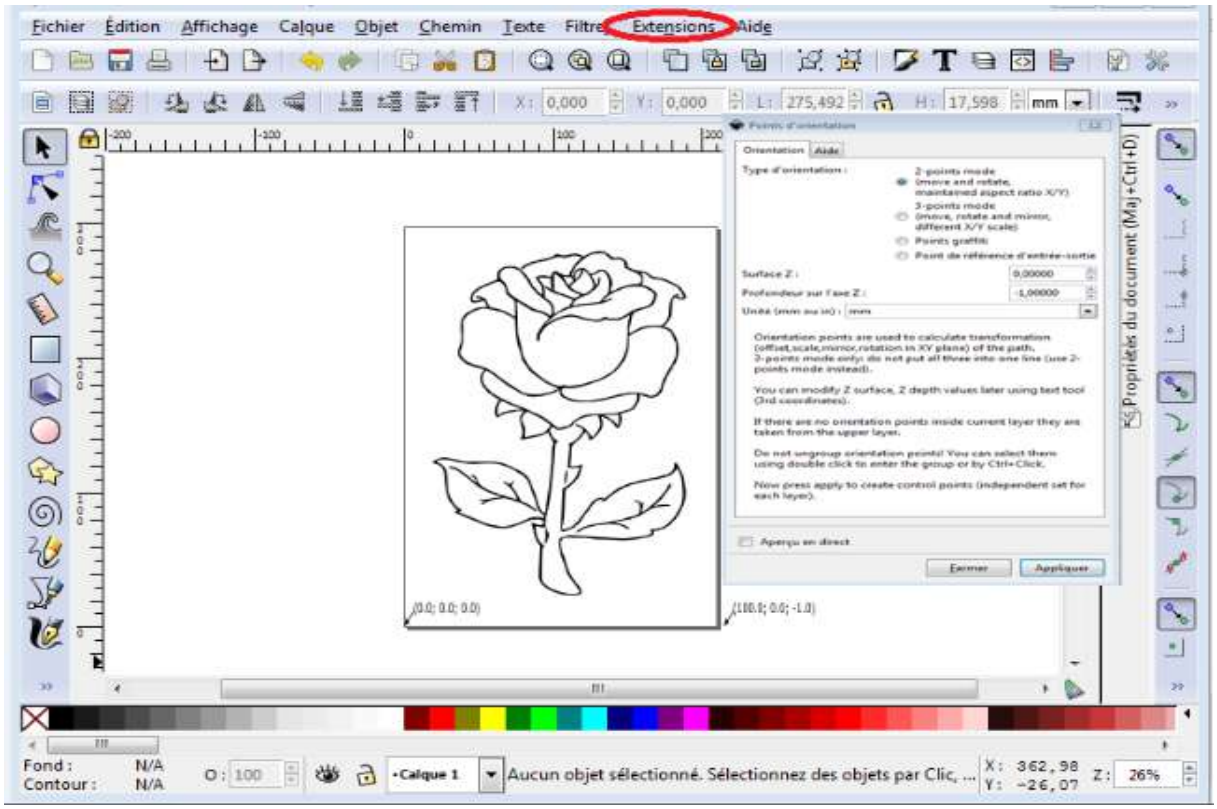


Figure III.7 : points d'orientations.

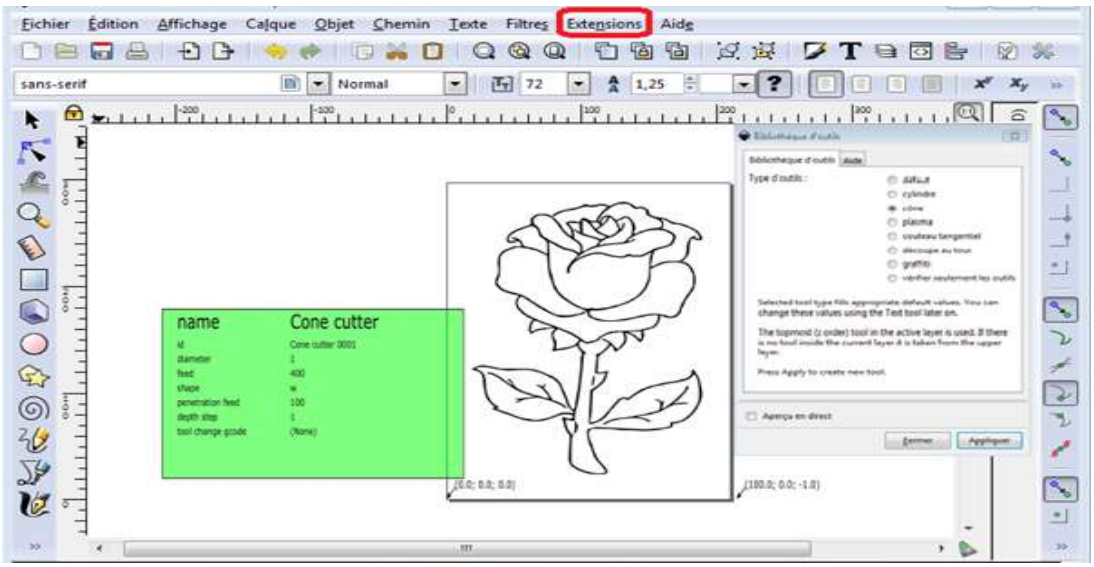


Figure III.8:bibliothèque d'outils

Pour générer le fichier **.ngc**, nous appuyons sur l'onglet **Extensions** puis sur la commande **Chemin vers Gcode.**

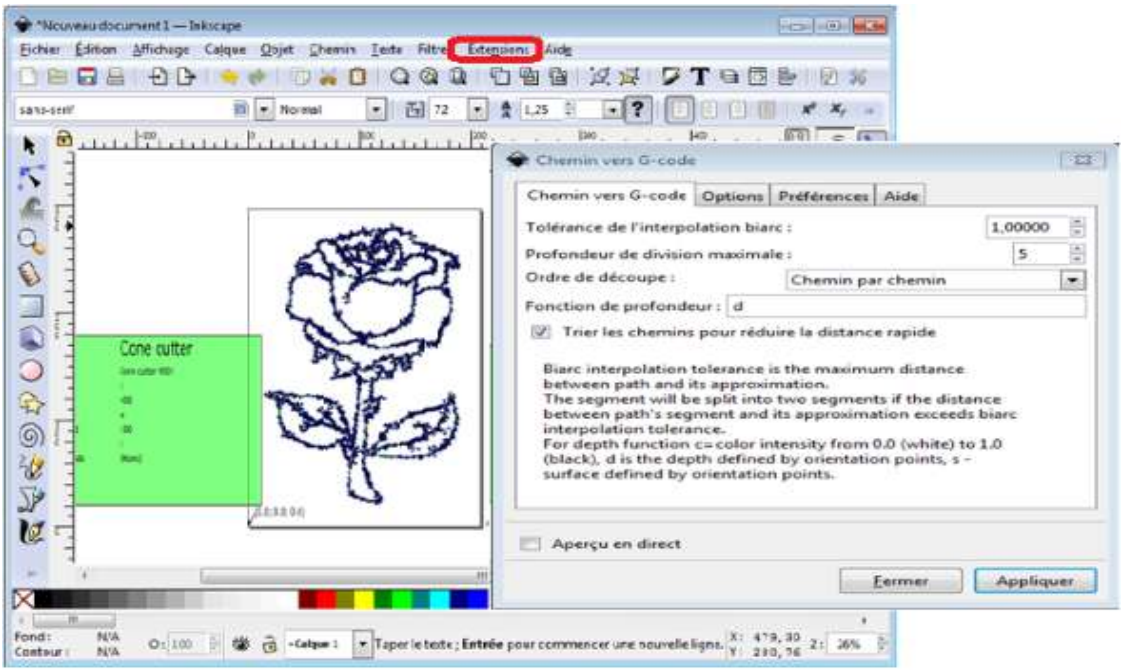


Figure III.9 : chemin vers G-code

III.3. Définition de Universal G code Sender :

Ce logiciel permet d'écrire et d'envoyer des instructions en code **G** via le port série vers une carte de type arduino dont cette dernière contient en mémoire un interpréteur dite GRBL.

III.3.1. Logiciel de pilotage (Universal Gode Sender)

La solution la plus fonctionnelle que nous avons trouvée une évolution pour le pilotage d'une CNC se fait à l'aide de séquences d'instructions en Gcode Sender vers ARDUINO UNO d'interpréter et contrôler par GRBL, qui est cours de développement par Winder [29].

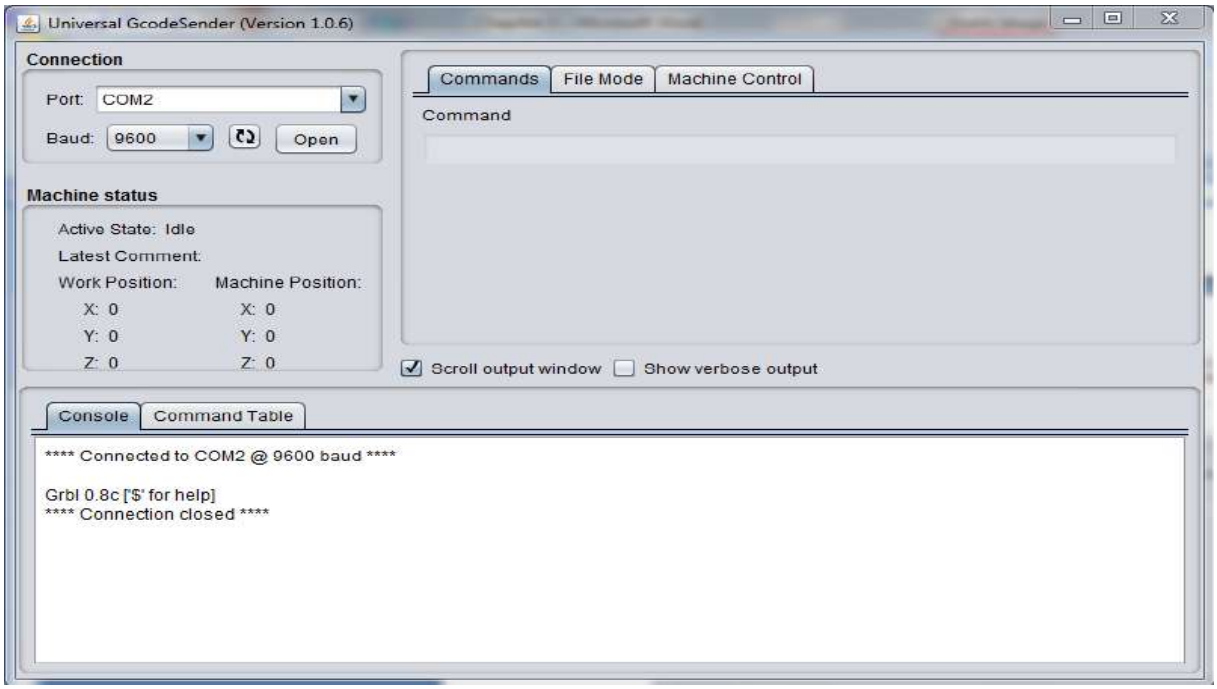


Figure III.10 : Universel Gcode Sender

Nous avons sélectionnées le port COM sur lequel est branché votre Arduino (pour moi COM2) et nous allons réglées la vitesse à sur **9600 baude** et on clique sur Open pour ouvrir la connexion.

Trois onglets sont à votre disposition **:**

- **Commands** afin d'envoyer directement du G Code à la machine ou pour configurer GRBL comme avec le moniteur série (avec $$).
- **File Mode** pour envoyer un fichier G Code à la CNC.
- **Machine Control** pour piloter manuellement votre CNC.

Voyons voir de plus près le dernier onglet.

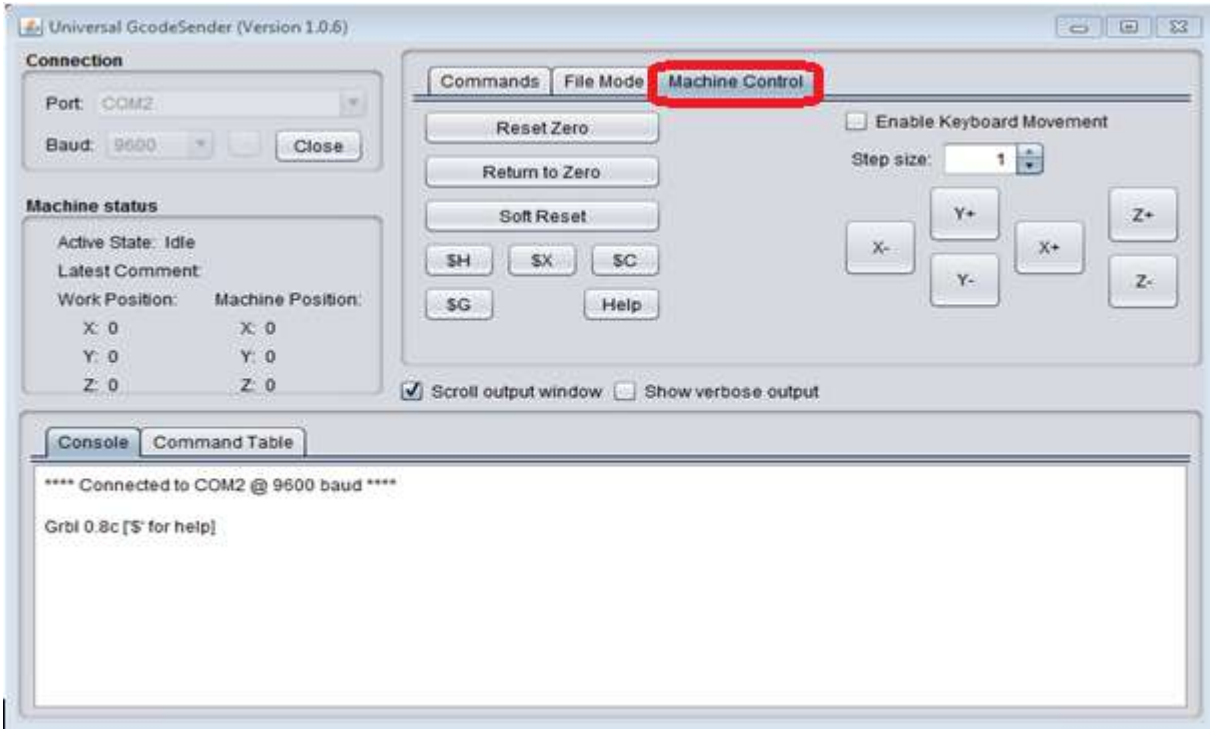


Figure III.11 : Onglet « Machine Control »de l'UGC

En cliquant sur les touches, vous pouvez faire tourner votre moteur pas à pas, en indiquant avec step size la distance parcourue pour un pas. C'est la manière la plus simple de vérifier que vos moteurs fonctionnent correctement avant de les fixer définitivement dans la CNC et de voir si vous avez bien configuré GRBL.

Une fois que tout fonctionne bien avec le pilotage manuel, vous pouvez aller dans **File Mode** et envoyer directement votre fichier G code.

III.3.1.1. 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.

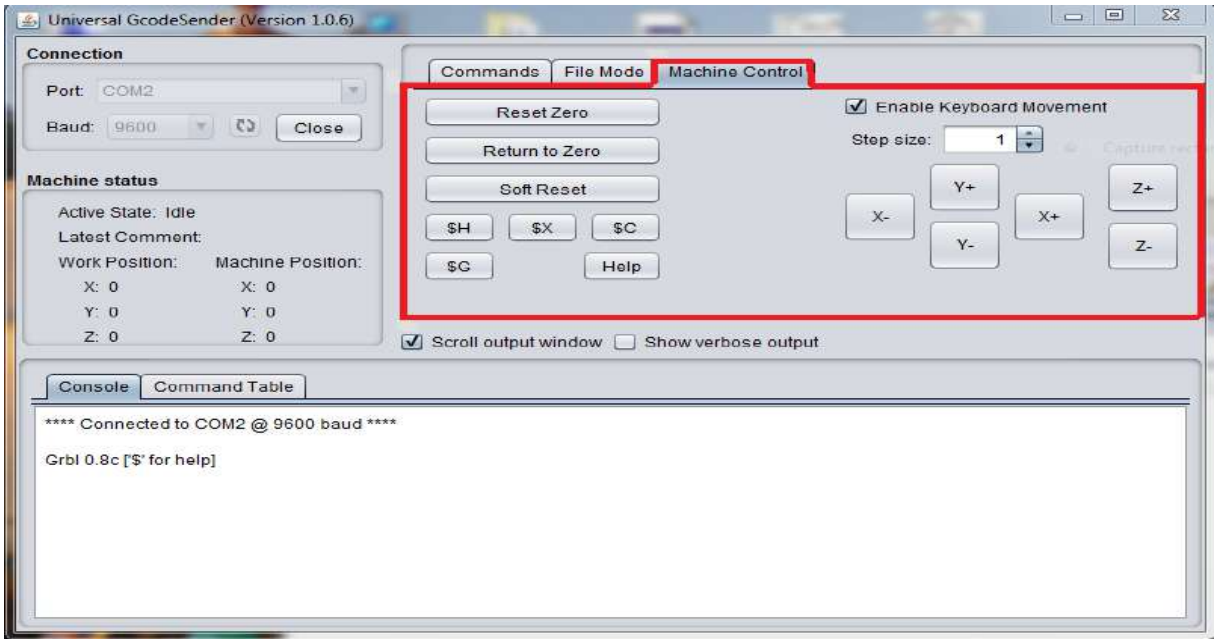


Figure III.12: l'interface de "G-code Sender" on mode manuel.

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).

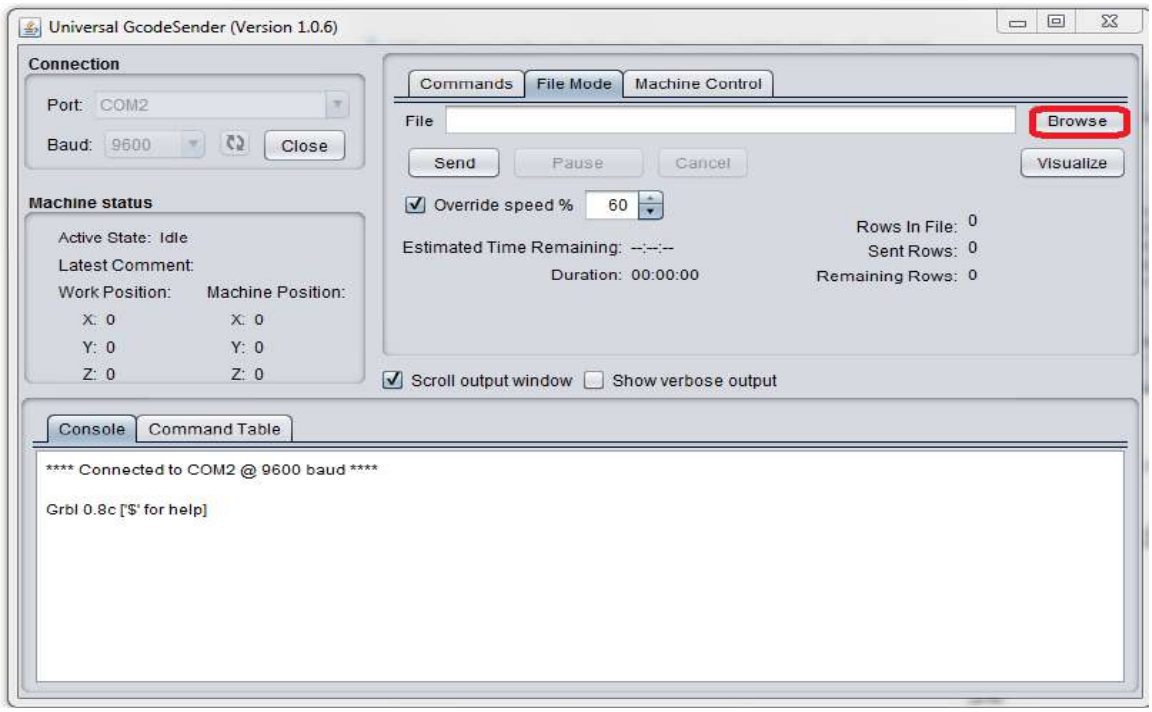


Figure III.13: Onglet « File commande » de l'UGCS

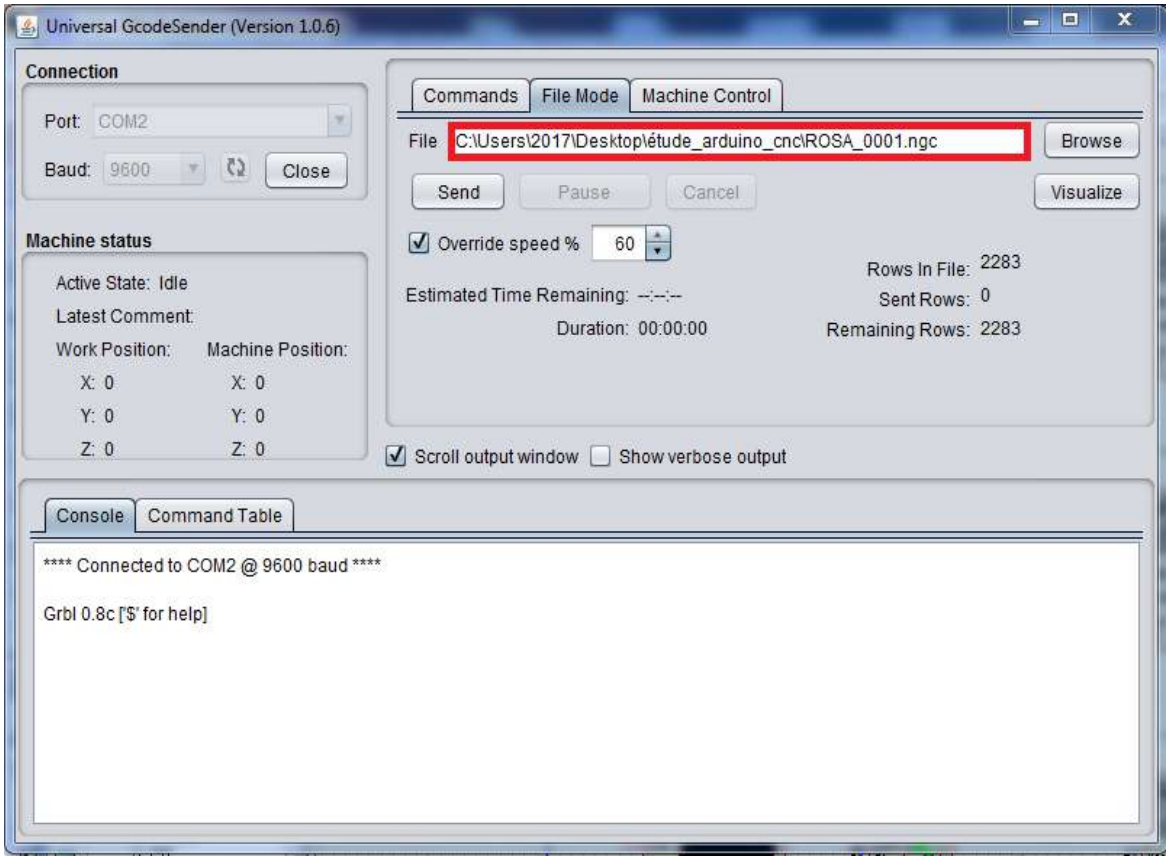
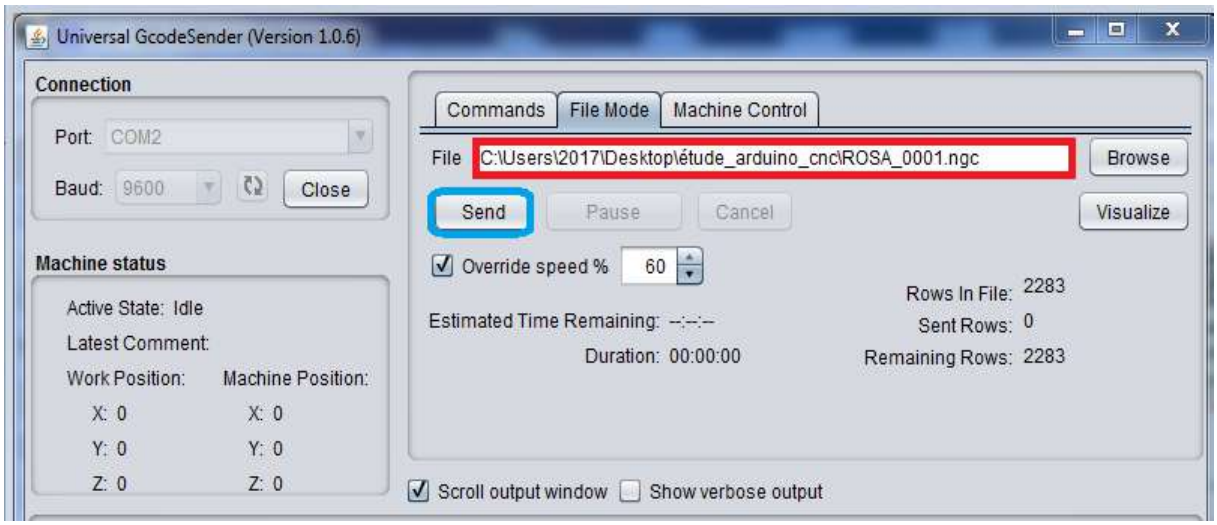


Figure III.14: l'interface de "G-code Sender" on mode automatique.

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".



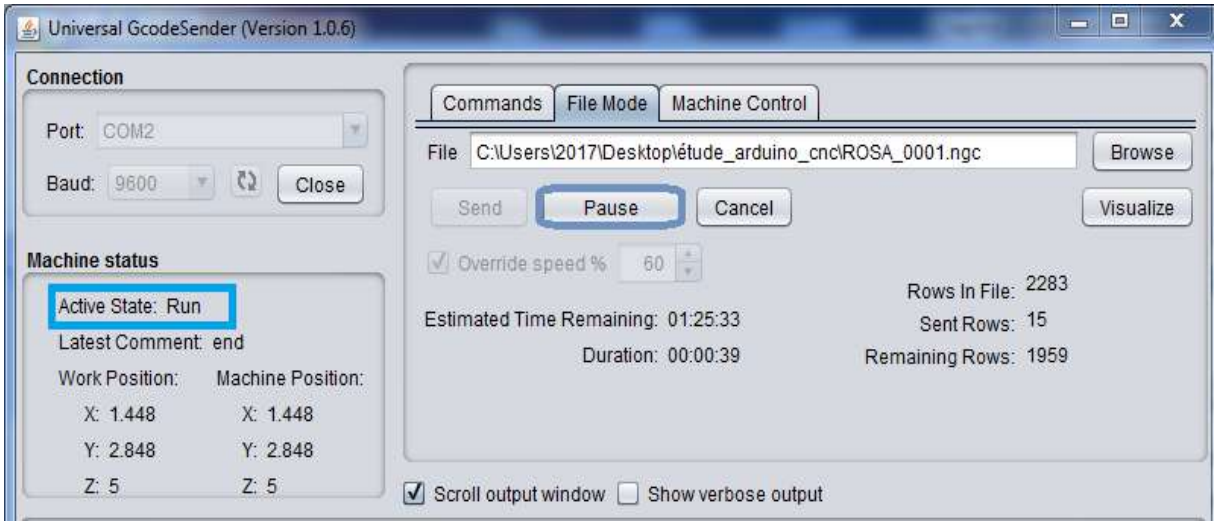


Figure III.15: envoyé fichée G code

III.3.1.2. 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.3.1.3. 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.

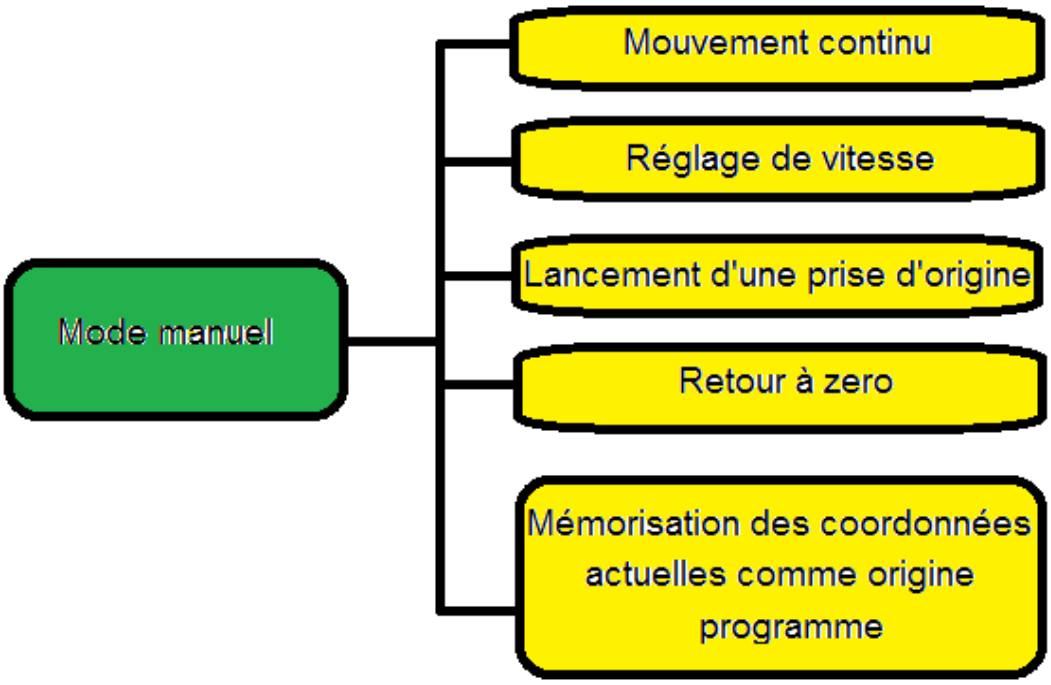


Figure III.16: de mode manuel.

- 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 (Vmin, Vmax).
- Retour à zéro: il s'agit de déplacer la tête à l'origine absolue. Ce qui diffère de la prise d'origine décrite précédemment c'est que la tête retourne à l'origine sans passer par les butées, donc il est basé sur l'estimation de la position calculé par l'arduino.
- Retour à l'origine programme : l'origine programme est par défaut l'origine absolue de la table, en fait, c'est l'origine du texte ou image. Cette origine peut être changée par le code en G en mode automatique ou même manuellement. Une fois changé, l'arduino le mémorise dans son EEPROM.

- Mémoriser l'origine programme : si l'opérateur veut changer cette origine, il doit déplacer la tête manuellement à l'aide du déplacement continu et l'incrémentation par pas, puis il exécute la mémorisation.

III.3.1.4. 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. Commençant par la réception de l'instruction jusqu'à la commande des moteurs et de la tête. Si l'opérateur clique sur le bouton « send » après le chargement de fichier G-code (soit texte ou image) dans le programme de transfert, l'arduino fonctionne en mode automatique, le Arduino entre dans une boucle et prêt pour recevoir les données [30].

III.3. 2. Langue de contrôleurs numériques « G-code »

III.3.2.1.Définition de G-Code

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 X3est une ligne de code valide avec deux mots. G1est une commande qui signifie déplace toi en ligne droite à la vitesse programmée et X3fournit 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 [31].

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 M2ou 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 [31].

III.3.2.2. 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.3.2.3.Les principales fonctions de G-code

III.3.2.3.1.Les fonctions préparatoires, instructions G

Le tableau **III.3.** Représente Les instructions G

G0	**Positionnement à vitesse rapide**
G1	Interpolation linéaire
G2	Interpolation circulaire/hélicoïdale en sens horaire
G3	Interpolation circulaire (sens anti-horaire, trigo)
G10	Définition de l'origine du système de coordonnées
G12	Poche circulaire en sens horaire
G13	Poche circulaire en sens antihoraire
G15/G16	Déplacement sur des coordonnées polaires selon G0/G1
G17	Sélection du plan XY
G18	Sélection du plan XZ
G19	Sélection du plan YZ
G20/G21	Unités en pouces/mm
G28	Retour à l'origine
G30	Retour à l'origine
G31	Trajectoire directe du palpeur
G40	Pas de compensation de rayon d'outil
G41/42	Démarrer compensation de rayon d'outil à gauche/droite
G43	Appliquer la compensation de longueur d'outil (positif)

G49	Désactive la compensation de longueur d'outil
G50	Réinitialise les facteurs d'échelle à 1.0
G51	Activation du facteur d'échelle pour les valeurs d'axes saisies
G52	Décalage temporaire du système de coordonnées
G53	Déplacement dans le système de coordonnées absolu machine
G54	Utilisation décalage du gabarit 1
G55	Utilisation décalage du gabarit 2
G56	Utilisation décalage du gabarit 3
G57	Utilisation décalage du gabarit 4
G58	Utilisation décalage du gabarit 5
G59	Utilisation décalage du gabarit 6 / utilisation du gabarit général
G61/G64	Arrêt précis / Vitesse constante
G68/G69	Programmer système de Coordonnée des axes rotatifs
G70/71	Unités en pouces/mm
G73	Cycle préprogrammé : perçage par à-coups
G80	Annuler les modes de déplacement (y compris les cycles préprogrammés)
G81	Cycle préprogrammé : perçage
G82	Cycle préprogrammé : perçage avec pauses
G83	Cycle préprogrammé : perçage par à-coups

Tableau III.1. Tableau d'index du G-code **[32].**

➢ G0 Interpolation linéaire en vitesse rapide

Pour un mouvement linéaire en vitesse rapide, programmer G0 axes, tous les mots d'axe sont facultatifs. Le G0est facultatif si le mode mouvement courant est déjà G0. Cela produit un mouvement linéaire vers le point de destination à la vitesse rapide courante (ou moins vite si la machine n'atteint pas cette vitesse). Il n'est pas prévu d'usiner la matière quand une commande G0 est exécutée. Un G0 seul peut être utilisé pour passer le mode de mouvement courant en G0 **[33].**

Exemple avec G0:

```
G90 (Fixe les déplacements en mode absolu) G0 X1 Y-2.3 (mouvement linéaire en vitesse
```

rapide du point courant à X1 Y-2.3) M2 (fin de programme)

- Si la compensation d'outil est active, le mouvement sera différent de celui.
- Si **G53** est programmé sur la même ligne, le mouvement sera également différent.

C'est une erreur si:

- Un mot d'axe est indiqué sans valeur réelle.
- Un mot d'axe est indiqué qui n'est pas configuré.

➢ G1 Interpolation linéaire en vitesse travail

Pour un mouvement linéaire en vitesse travail, (pour usiner ou non) programmer G1 axes, tous les mots d'axe sont facultatifs. Le G1est facultatif si le mode de mouvement courant est déjà **G1.** Cela produira un mouvement linéaire vers le point de destination à la vitesse de travail courante (ou moins vite si la machine n'atteint pas cette vitesse). Un G1 seul peut être utilisé pour passer le mode de mouvement courant en G1 **[33].**

Exemple avec G1:

G90 (Fixe les déplacements en mode absolu) G1 X1.2 Y-3 F10 (mouvement linéaire à 10 unités/mn du point courant à X1.2 Y-3) Z-2.3 (mouvement linéaire à 10 unités/mn du point courant à Z-2.3) Z1 F25 (mouvement linéaire de l'axe Z à 25 unités/mn vers Z1) M2 (Fin de programme)
G2, G3

➢ G3 Interpolation circulaire en vitesse travail

Un mouvement circulaire ou hélicoïdal est spécifié en sens horaire avec G2ou en sens antihoraire avec G3. La direction est vue depuis le côté positif de l'axe autour duquel le mouvement se produit **[33].**

Exemple avec G2 G3

G2 ou G3 axes décalages (format centre) G2 ou G3 axes R- (format rayon) G2 ou G3 décalages <P-> (cercles complet)

Les axes de cercle ou les hélicoïdes, doivent être parallèles aux axes X, Y ou Z du système de coordonnées machine. Les axes (ou, leurs équivalents, les plans perpendiculaires aux axes) sont sélectionnés avec **G17** (axe Z, plan XY), G18 (axe Y, plan XZ), ou **G19**(axe X, plan YZ).

Pour programmer un hélicoïde, inclure le mot d'axe perpendiculaire au plan de l'arc. Par exemple, si nous sommes dans le plan **G17,** inclure un mot **Z**, ceci provoquera un mouvement de l'axe **Z** vers valeur programmée durant tout le mouvement circulaire **XY**.

Pour programmer un arc supérieur à un tour complet, utiliser un mot **P** spécifiant alors le nombre de tours complets en plus de l'arc. Si **P** n'est pas spécifié, le comportement sera comme si **P1**avait été donné: ceci étant, un seul tour complet ou partiel sera effectué, donnant un arc plus petit ou égal à un tour complet. Par exemple, si un arc de 180° est programmé avec P2, le mouvement résultant sera d'un tour et demi. Pour chaque incrément de P au-delà de 1, un tour complet sera ajouté à l'arc programmé. Les arcs hélicoïdaux multi-tours sont supportés ce qui donne des mouvements très intéressants pour usiner des alésages ou des filetages.

III.3.3. 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.3.3.1. 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) [34].

- **Installation de Grbl**

Le pilotage d'une CNC se fait à l'aide de séquences d'instructions en Gcode. GRBL est un outil pratique qui permet d'interpréter le Gcode et de piloter très simplement une CNC avec Arduino.

- Téléchargez la dernière version .zip de GRBL sur github.
- Décompressez le dossier **Grbl-master**

Nous allons maintenant inclure GRBL aux bibliothèques Arduino. Pour cela, deux méthodes sont possibles.

- Via le logiciel Arduino, on clique sur Croquis puis Inclue une bibliothèque et on ajouter la bibliothèque .ZIP. Puis nous sélectionnons le dossier Grbl situé dans le dossier Grbl-master.

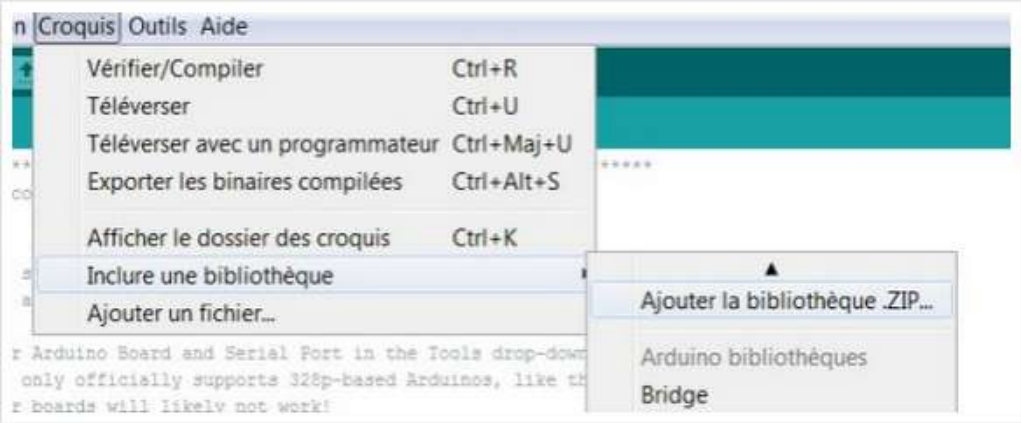


Figure III.17 : Installations de **GRBL** dans l'arduino.

- Autre méthode : nous allons vers C:\Program Files\ Arduino. puis Cliquez sur **Arduino** puis **librairies** et on colle le dossier **Grbl** situé dans le dossier **Grbl-master**. nous redémarrons ensuite l'IDE Arduino.

  Pour implémenter GRBL sur l'Arduino, rien de plus simple. Il suffit d'aller dans **:**

  Fichier => Exemples => grbl = > grbl-upload

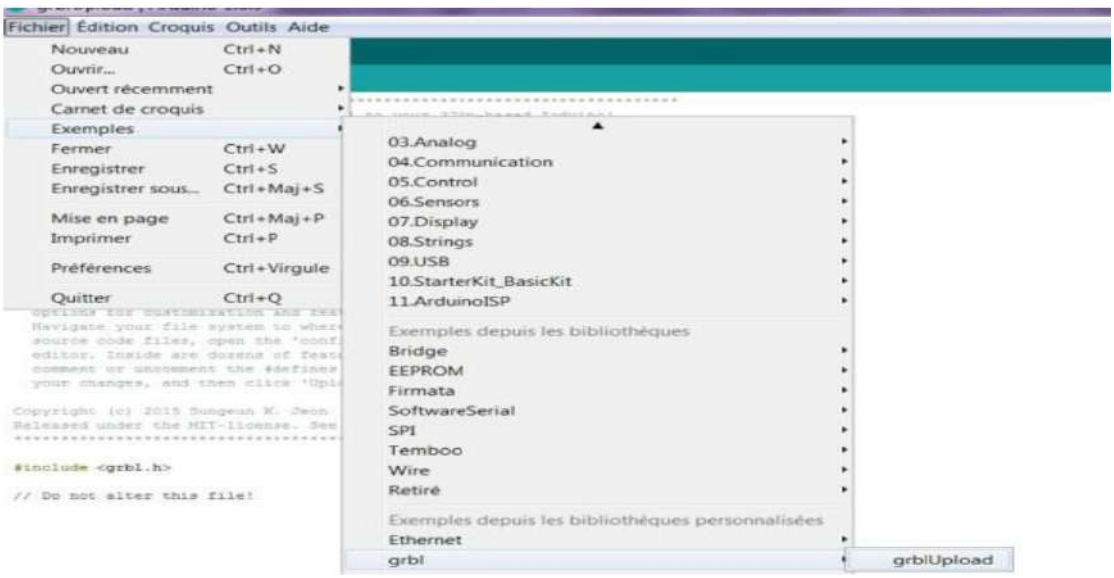


Figure. III.18 : ouvrire le GRBL

Une fois télécharge, on appui sur téléverser. On peut configurer GRBL afin qu'il s'adapte aux réglages de notre CNC grâce au moniteur série. Pour cela, ouvrez le moniteur série (**Outils > Moniteur série**). Réglez la vitesse à **115200 baud** et sélectionnez **nouvelle ligne**. En tapant $$ vous verrez apparaître la configuration de GRBL.

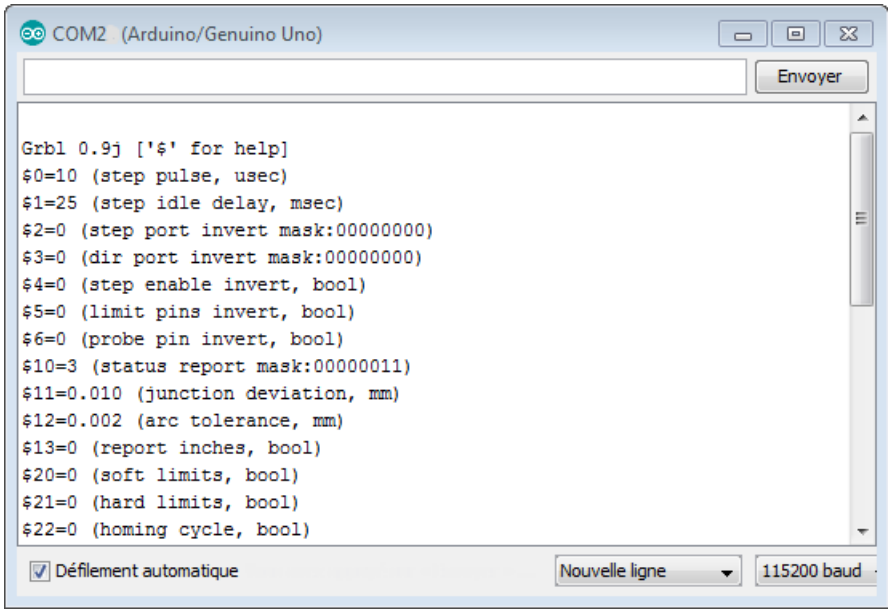


Figure III.19 : Moniteur série

III.4. Conclusion

Dans ce chapitre, nous avons vu trois parties **:**

- La première partie a été consacrée au logiciel de dessin 2D **Inkscape** et les étapes de la vectorisation des images bitmap en image vectorielle afin de générer leur fichier Gcode.

- La deuxième partie a été consacrée au logiciel **Universel G code sender** qui a la tâche du transfert des instructions Gcode vers la carte Arduino Uno via le port série.

- Dans la troisième partie on a défini le rôle du programme GRBL, qui permet d'interprète le G-code sous forme mouvement (un outil sur 3 axes (X, Y et Z)).

CHAPITRE IV

Simulation et commande de la machine CNC

IV.1. Introduction

Après avoir vu tous les composants matériels et softwares qui rentrent dans la conception d'une machine CNC à trois axes, dans ce chapitre on va voir comment travailler avec, afin de générer une image sur papier ou sur d'autres supports.

IV.2.Schéma synoptique :

Le schéma suivant montre le cheminement à faire pour arriver à un produit fini.

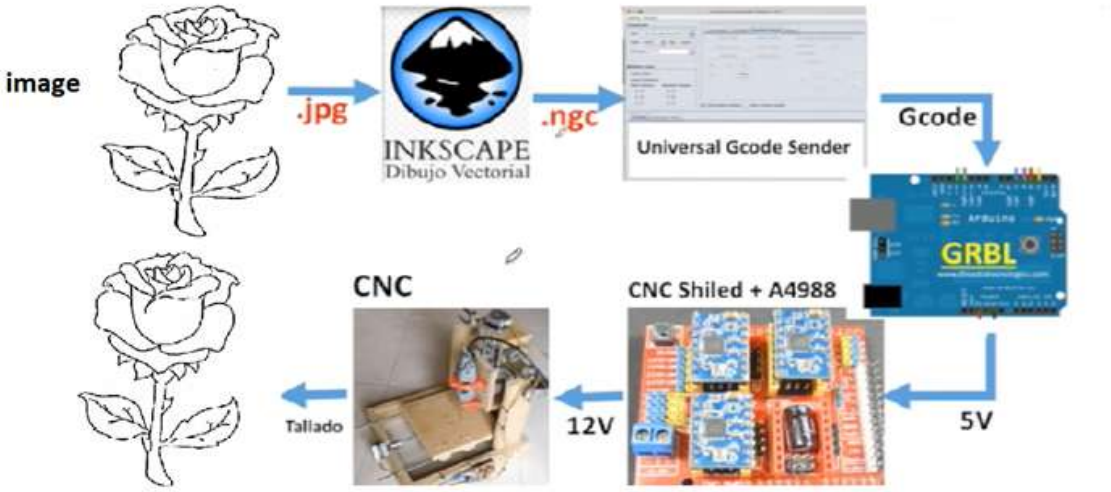


Figure IV.1 : Schéma synoptique de la commande.

D'après ce schéma, l'image de type .**JPG** ou .**BMP** passe tout d'abord par le logiciel Inkscape afin de générer le Code-G de cette image sous forme d'un fichier, ce fichier est ensuite pris en charge par le logiciel Universel Gcode sender pour le transmettre à la carte Arduino via le port série (ou USB), par la suite cette dernière interprète les instructions de Code-G à l'aide d'un interpréteur (GRBL) qui est chargé au préalable dans la mémoire du microcontrôleur de la carte, et les faire (instructions) sortir sous forme des signaux numérique afin de piloter les drivers qui sont relier aux moteurs

Avec l'exemple ci-dessous d'une image d'une fleur qu'on a appelé **rosa,** on montre tous les étapes à suivre :

Etape1 : Générer le G code avec inkscape

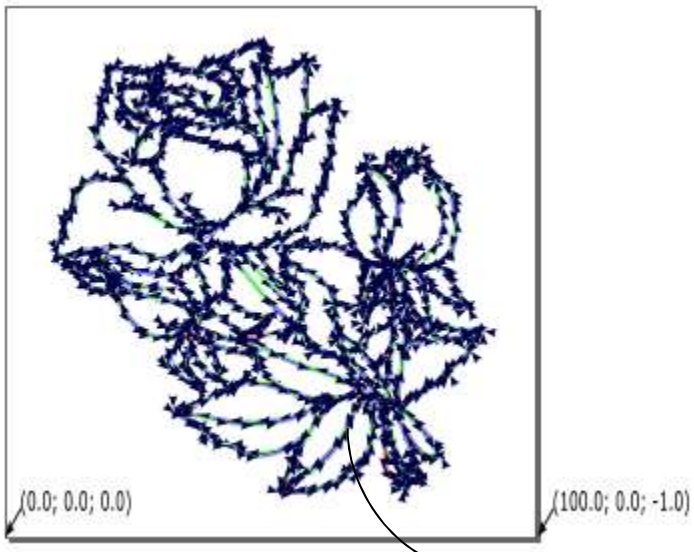


Figure IV.2 : Conversion rosa .jpg vers rosa.nrg

(Start cutting path id: path21)

(Change tool to Cone cutter)

G00 Z5.000000

G00 X46.819325 Y92.085185

G01 Z-1.000000 F100.0 (Penetrate)

G03 X45.491251 Y91.801285 Z-1.000000 I0.619283 J-6.145285 F400.000000

G03 X43.453325 Y90.954775 Z-1.000000 I4.774255 J-14.370117

G02 X41.116146 Y89.937963 Z-1.000000 I-10.195156 J20.239463

G02 X39.206284 Y89.384819 Z-1.000000 I-4.779680 J12.929305

G03 X37.357826 Y88.773018 Z-1.000000 I1.847879 J-8.681378

G03 X36.442846 Y88.118566 Z-1.000000 I1.231572 J-2.688678

G02 X35.597078 Y87.361275 Z-1.000000 I-5.030238 J4.767000

G02 X35.321169 Y87.268575 Z-1.000000 I-0.275909 J0.364252

G02 X35.015460 Y87.320405 Z-1.000000 I-0.000000 J0.927493

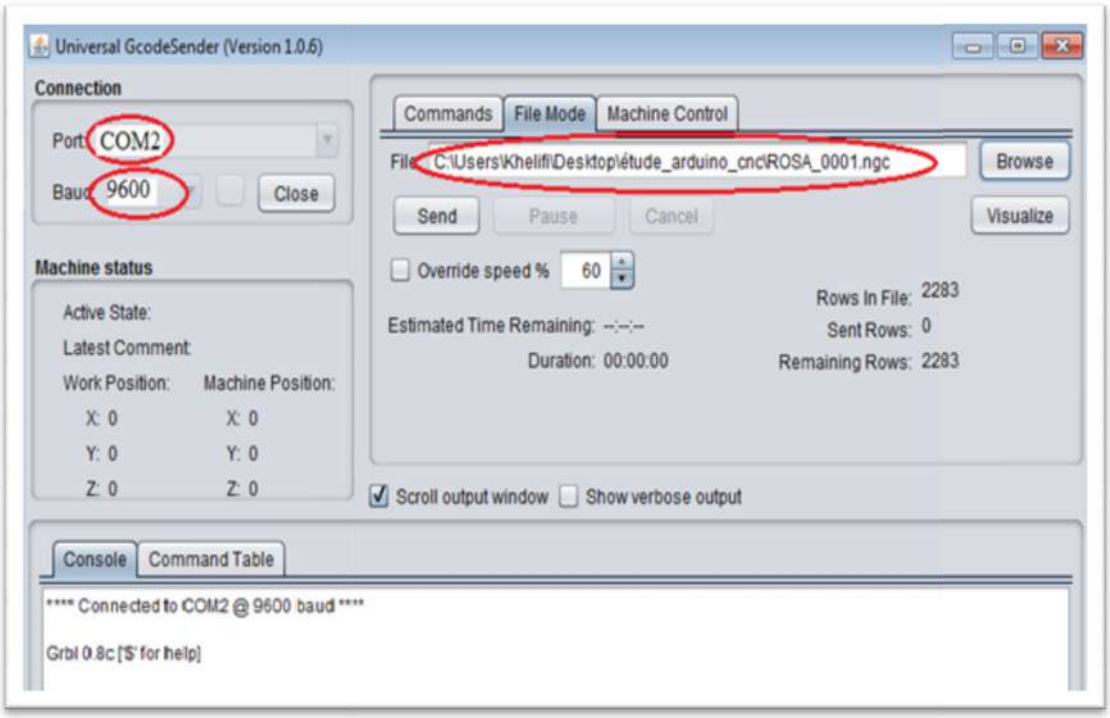
G02 X32.714920 Y88.177754 Z-1.000000 I19.415730 J55.613735

G01 X30.424022 Y89.085961 Z-1.000000

G01 X29.844748 Y88.324270 Z-1.000000

Figure IV.3 : Une partie du fichier **rosa**.nrg

Etape 2 : Chargement et visualisation du fichier rosa.nrg par universel Gcode sender



- Port : COM2
- Vitesse de transfert : 9600
- Fichier : ROS_8001.nrg

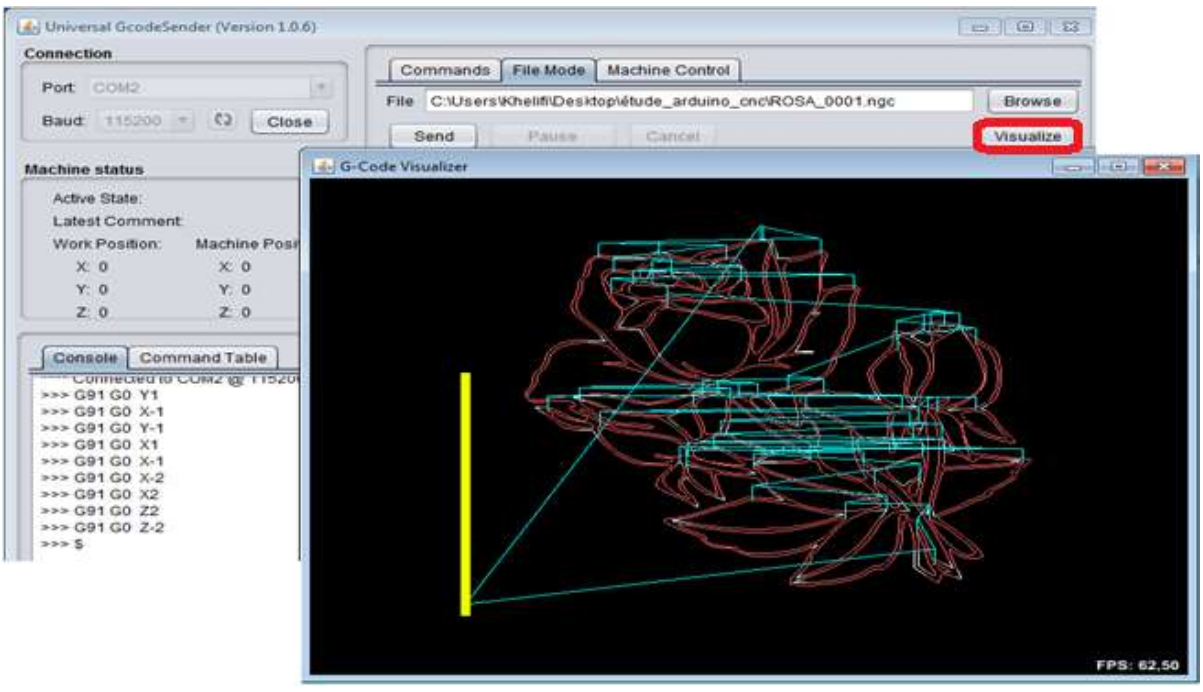


Figure IV.4 : visualisation du fichier rosa.nrg par universel Gcode sender

Etape 3 : chargement de fichier GRBL

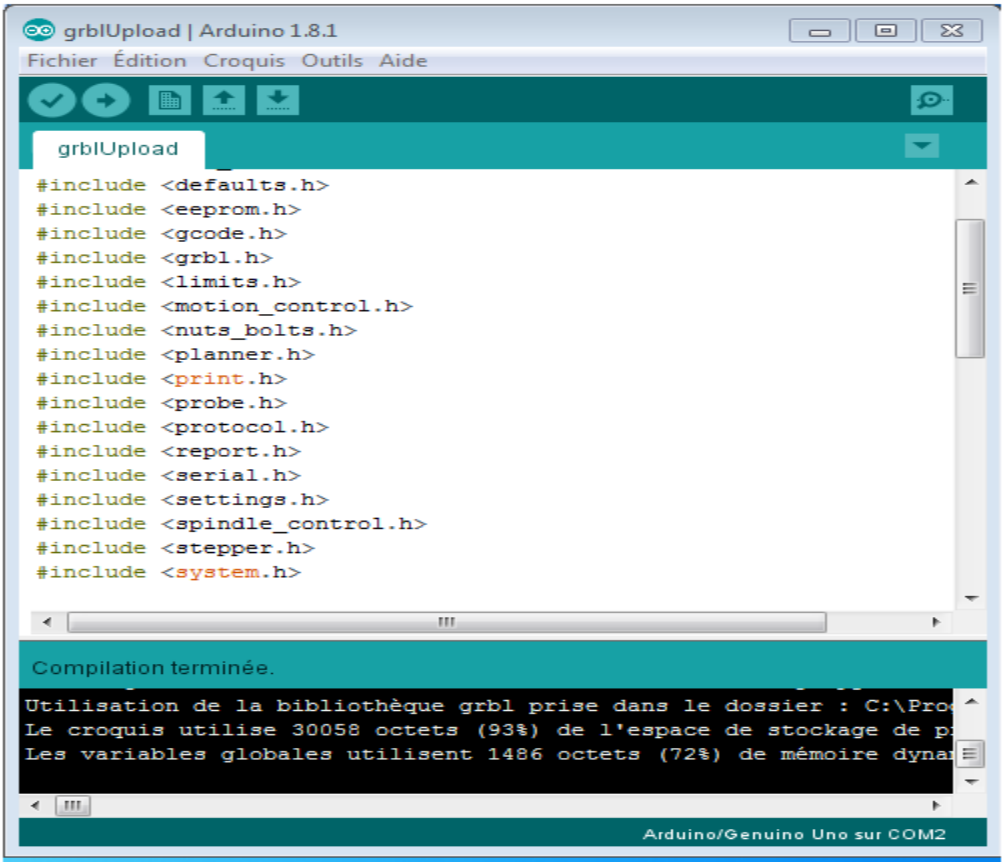


Figure IV.5 : Chargement de fichier GRBL

Etape 4 :l'envoi du fichier rosa_001.nrg (voir chapitre III)

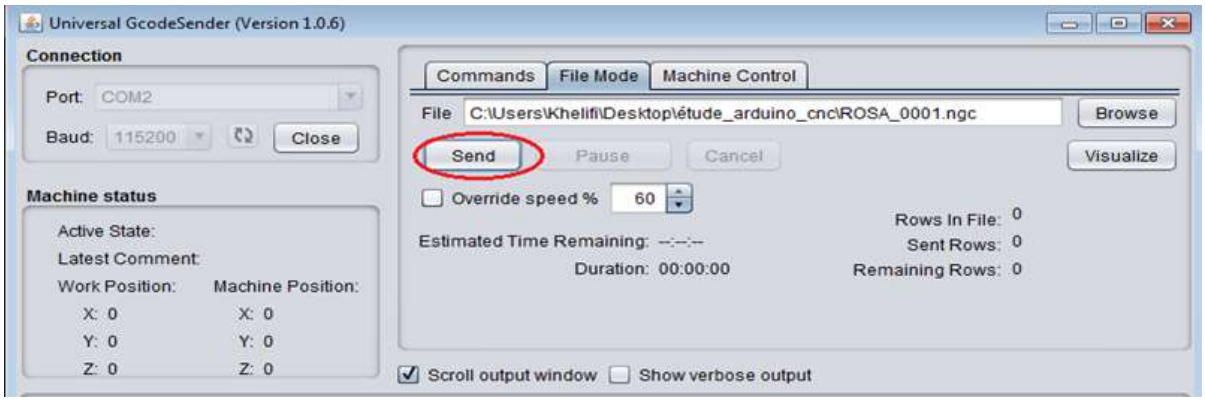


Figure IV.6 : Envoyé le fichier vers la machine CNC

IV.3 Schéma électrique :

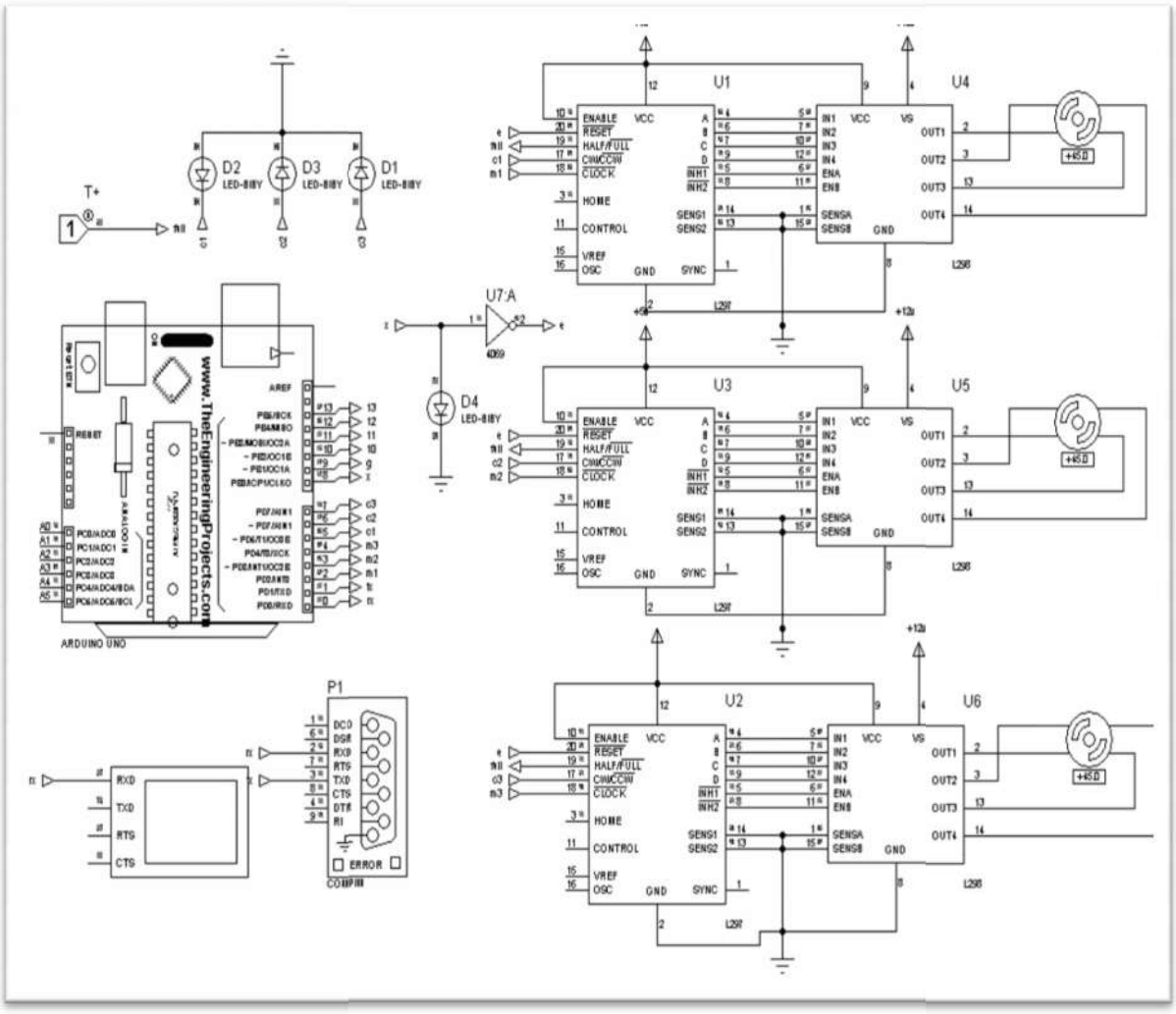


Figure IV.7 : schéma électrique.

D'après ce schéma, chaque driver (L297) est relié d'un coté à la carte arduino à travers trois broches :

- e: pour la réinitialisation des signaux
- **c**x (x=1, 2,3) : pour l'inversion du sens de rotation (CW : clockwise / CCW)
- **m**x (x=1, 2,3) : Entrée d'horloge (pour commander la vitesse des moteurs)
- la ligne **full** : c'est pour sélectionner le fonctionnement demi-pas ou pas complet

et d'autre coté au circuit de puissances L298 comme il est montré dans la figure ci-dessus

IV.4 Simulation et test le programme

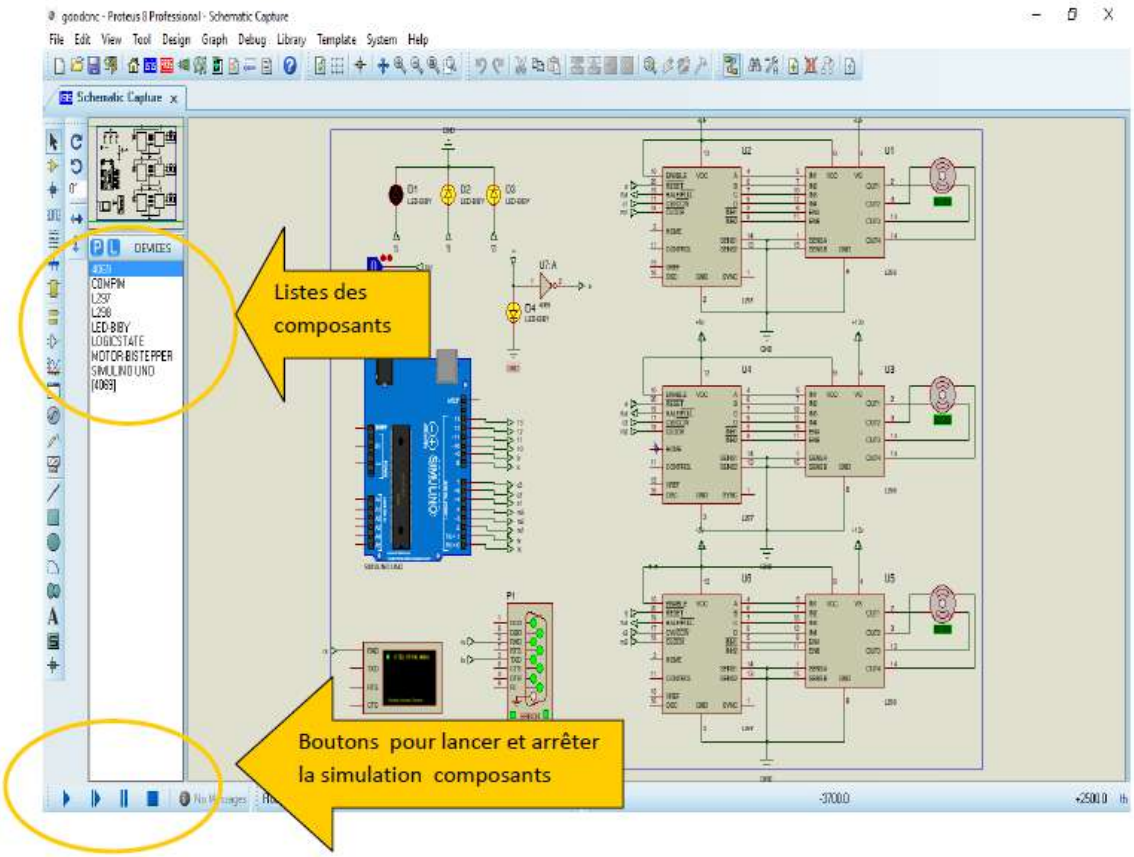


Figure IV.**8 :** Montage électronique dans Proteus

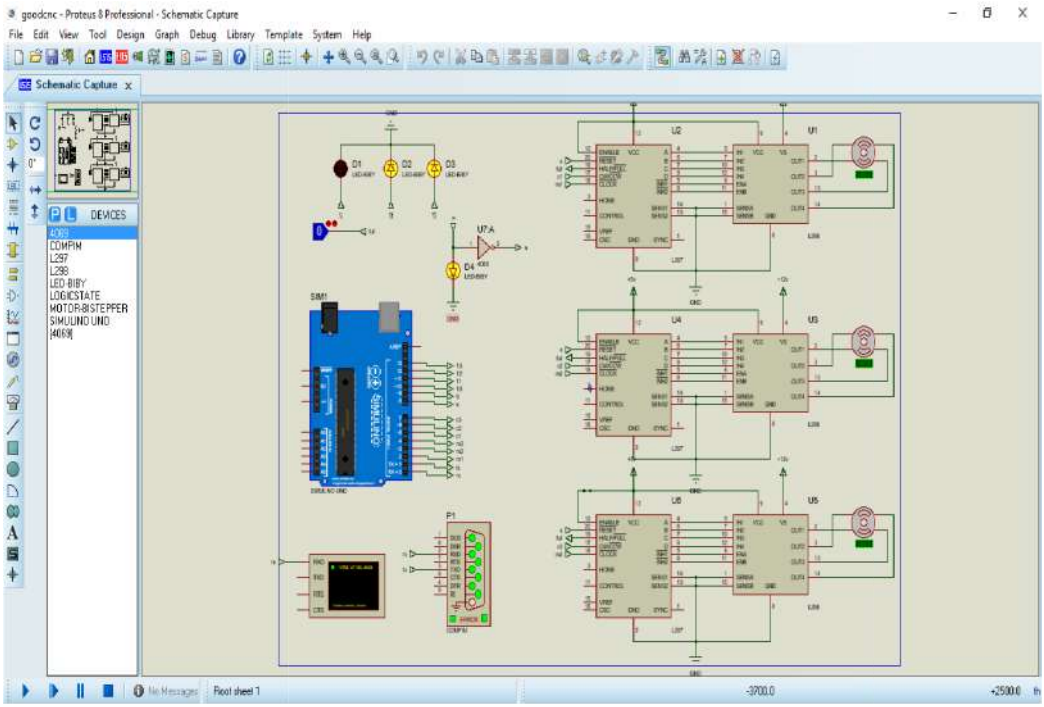


Figure IV.9 : Simulation de la machine CNC

IV.3. Conclusion :

Dans ce chapitre nous avons détaillée tous les étapes pour reporter une image ou une écriture sur un support on passant par l'éditeur de dessin et générateur du code-G Inkspace ensuite par le universel Gcode Sender pour transmettre le code G vers la carte arduino dont cette dernière est reliée aux moteurs pas à pas à travers des drivers. Puis on a terminé notre travail par un test à travers Proteus.

Conclusion générale

Dans ce travail, notre objectif est la simulation et la commande d'une machine CNC à base des moteurs pas à pas.

Durant tout le temps qui nous a été alloué pour effectuer ce travail, nous avons essayé, du mieux qu'on pouvait, d'expliquer avec beaucoup de détail les étapes.

De conversion une image sur **pc** sous forme g code vers un mouvement (un outil sur 3 axes (X, Y et Z)). Et pour effectué à cet objectif, nous avons vu notre travail on quatre :

Dans le premier chapitre, nous avons entamé une généralité sur les machines-outils à commande numérique, le principe de fonctionnement de ce dernier et la conception assistée par ordinateur (CAO) et (FAO).

Dans la deuxième chapitre, nous avons entamé la sur Composants matériels de la machine CNC à 3 axes, consiste sur les moteurs pas à pas que nous avons étudié en détail. Cette partie est essentielle au fonctionnement des machines à commande numérique. Fait une description et la mise en œuvre de la carte Arduino Uno et les circuits intégrés (L297, L298), pour tester les différentes, Interfaces électroniques d'acquisition de données et de commandes des moteurs pas à pas.

Ensuite, la troisième chapitre, est consacrée à des logiciels du pilotage du CNC de simulation pour cette raison, plusieurs parties ont été envisagées :

- La première partie a été consacrée au logiciel de dessin 2D **Inkscape** et les étapes de la vectorisation des images bitmap en image vectorielle afin de générer leur fichier Gcode.
- La deuxième partie a été consacrée au logiciel **Universel G code sender** qui a la tâche du transfert des instructions Gcode vers la carte Arduino Uno via le port série.
- Dans la troisième partie on a défini le rôle du programme GRBL, qui permet d'interprète le G-code sous forme mouvement (un outil sur 3 axes (X, Y et Z)).

Enfin, dans la dernière partie nous avons essayé de rassembler ce qui a été étudié dans les chapitres précédent. Nous avons vu détaillée tous les étapes pour reporter une image ou une écriture sur un support on passant par l'éditeur de dessin et générateur du code-G Inkspace ensuite par le universel Gcode Sender pour transmettre le code G vers la carte arduino dont Cette dernière est reliée aux moteurs pas à pas à travers des drivers. Puis on a terminé notre travail par un test à travers Proteus, la simulation elle nous a donné une bon resultat.

Par ailleurs, un certain nombre de perspectives sont envisageables dans ce travail. A très court terme, il s'agit d'appliquer une autre méthode de commande de la machine CNC avec :

- Les PIC 16F84 ou Raspberry PI
- 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.

[1] Benbekhti Ahmed, « **Etude de réalisation d'un support pour affutage des forets sur une machine à commande numérique** », mémoire de fin d'etude, Universite Abou Bekr Belkaid-Tlemcen 2013.

[2] Abdennadji Youssef, Ahmadi Anouar « **Conception et réalisation d'une commande numérique d'une machine de decouplaser** », Ingénieurd'école nationale de Gabés, juin 2010.

[3] Oukaci Mourad, Beziou Mohamed« **Etude et réalisation d'une machine à commande numérique CNC à 3 axes**» Projet de fin d'études 2012-2013.

[4] Mohammed Maatou, Abderrahman Bellagh «**Automatisation et realisation a petite echelle (maquette) d'une chaine transporteuse de briques**» mémoire master Universite Hassiba Benbouali de Chlef, Juin 2016.

[5] Pierre Mayé **«Moteurs électriques pour la robotique»** ; 2éme édition, octobre 2013

[6] A. Zonait, S. Kerrouche. **« Commande de deux moteurs pas à pas via le port parallèle avec Delphi** » Mémoire d'ingénieur d'état en électronique, Université Bejaia, Juin 2009.

[7] Bentaleb mokhtar, gasmi elhadi **«Réalisation et commande d'une machine cnc à base des moteurs pas à pas**» mémoire master, et de la Technologie Houari boumediene, option génie électrique, (2011).Universite, Kasdi Merbah Orglaua **,**2015/2016.

[9] Guide du micro motorisation.

[10] Ammar.Adel, Guehria, Amine, **« Conception et réalisation d'une carte de commande pour un moteur pas à pas** », Université des Sciences.

[11] Gérad Yvraut **«Les Moteurs Pas a Pas»**

[13] Takashi Kenjo and Akira Sugawara, «**stepping motors and their microprocessor controls »** Oxford University Press, ISBN 0-19-859385-6.

[14] Mr Taleb Bendiab Zakaria, «**Eude et realisation d'une carte de commande de trois moteurs pas à pas pour machine de prototypage rapide»** Projet de Fin d'Etudes, Université Abou bekr Belkaid – Tlemcen Faculté de Technologie 2013 /2014.

[15] Takashi Kenjo and Akira Sugawara, «**stepping motors and their microprocessor controls »** Oxford University Press , ISBN 0-19-859385-6.

[16] Ait Dahmane Kahina, Ait Ziane Meziane« **Conception et Réalisation d'un Bras Manipulateur Commandé par API »** mémoire fin d'etude master, Université de Khemis miliana 2014/2015.

[19] S.V.D. Reyvanth, G. Shirish, « **PID controller using Arduino** ».

[20] C. Tavernier, « Arduino applications avancées ». Version Dunod.

[22] Christian Tavernier « **Arduino Maitrise sa programmation et ses cartes d'interface**

(Shield) », Paris 2011, Dunod.

[26] Benkortbi Abdelhakim, Ferroukh Mohamed Amine « **conception et realistion d'un suiveur du soleil a base de la carte arduino uno** ». mémoire fin d'etude master**,**
Universite dr yahia fares de medea, 2014/2015.

[27] Patrique Oguic « Moteur pas à pas et PC deuxième édition » dunod, Paris, 2004

[30] John Kennedy **« Bresenham's Integer Only Line Drawing Algorithm»** Santa Monika college 1900 pico blvd.

[31] Steve Krar, Arthur Gill, **«Computer numerical control programming basics»** 200 Madison Avenue, New York, NY 1001.6.

[32] MACH 3 Un guide utilisateur pour l'installation, **la configuration et les opérations** version 3.01.

[34] Torjus Spilling **« Self-Improving CNC Milling Machine »**, master, university Oslo, 2014.

Liens Internet:

[8] http://www.lyrfac.com/soutiens/knbase/pdf/moteur%20pas%20a%20pas.pdf

[12] http://www.positron-libre.com/electronique/moteur-pas-a-pas/sequence commande Moteur-pas-a-pas.php.

[17] http://www.generationrobots.com/fr/152-arduino. consulter le : mars 2015.

[18] http://projet.eu.org/pedago/sin/tutos/arduino.pdf

[21] X.Hinault. www.mon-club-elec.fr.

[23] http://www.electroniquepratique.com.

[24] www.reality.be/elo/labos2/files/AtMega32DocFr.pdf.

[25] www.jp79dsfr.free.fr/_.../Elec%20_%20Arduino%20-%20Memo%20et%20condense.pdf

[28] http://inkscape.fr/

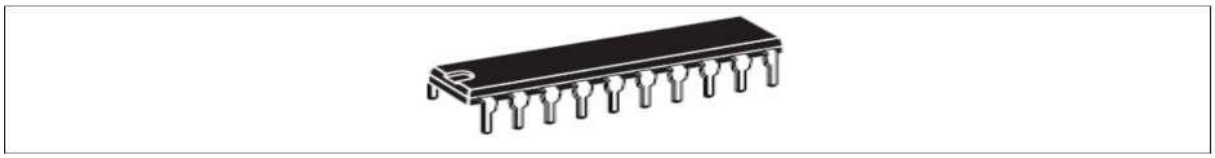
[29] http://www.civade.com/post/2014/01/02/Arduino-et-GRBL-l-incontournable solution pour-piloter-une-petite-CNC ,2016.

[33] http://www.linuxcnc.org/docs/devel/html/gcode/gcode_fr.html#sec:Table-des-index-du-G-code.

BIBLIOGRAPHIE

ANNEXES

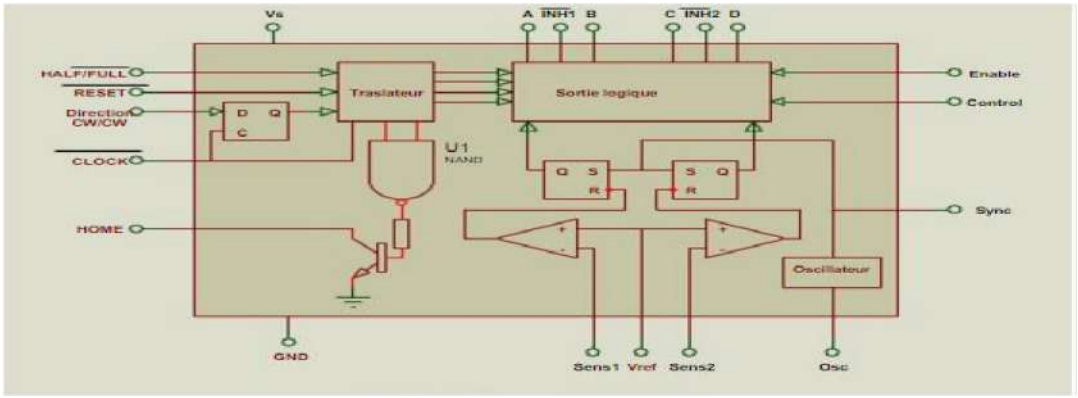
Annexe -1- : Circuit de commande



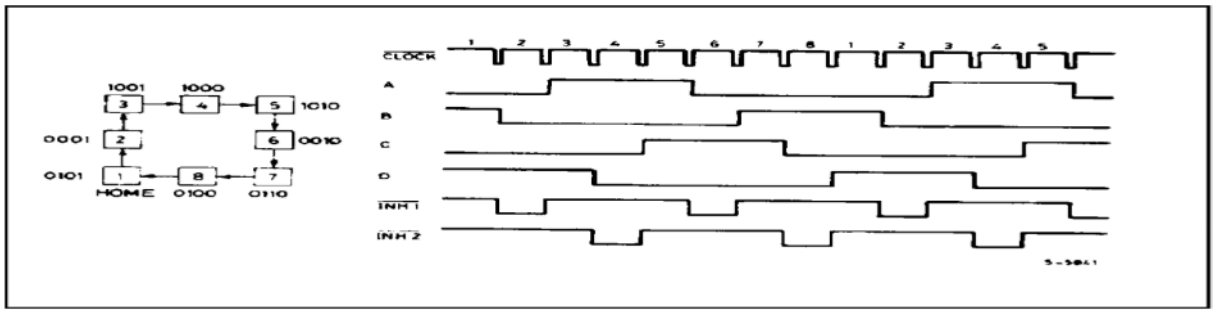
PIN FUNCTIONS - L297

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	$\overline{INH1}$	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	$\overline{INH2}$	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	ENABL 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 ▫ 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.
	CLOCK	The step occurs on the rising edge of this signal
	HALEF /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 aneven-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
	RESET	Reset input.An active low pulse on this input restores thetranslator to the home position (state 1, ABCD = 0101).

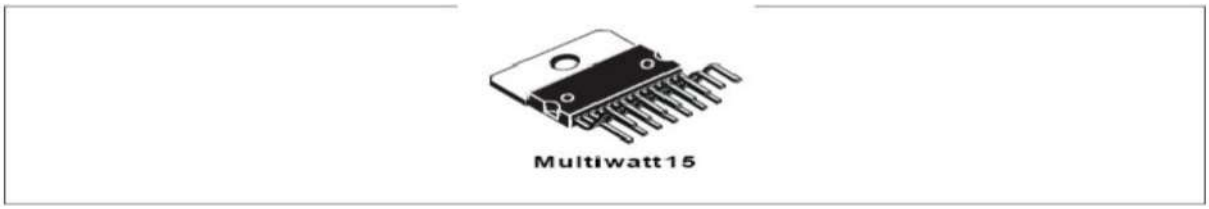


Structure interne du circuit L297.



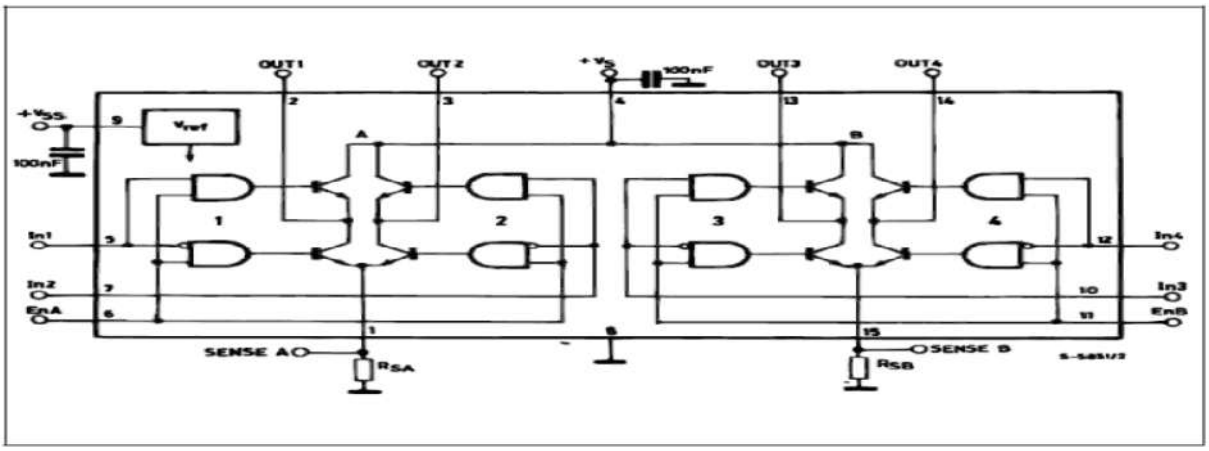
Mode monophasé

Circuit L298



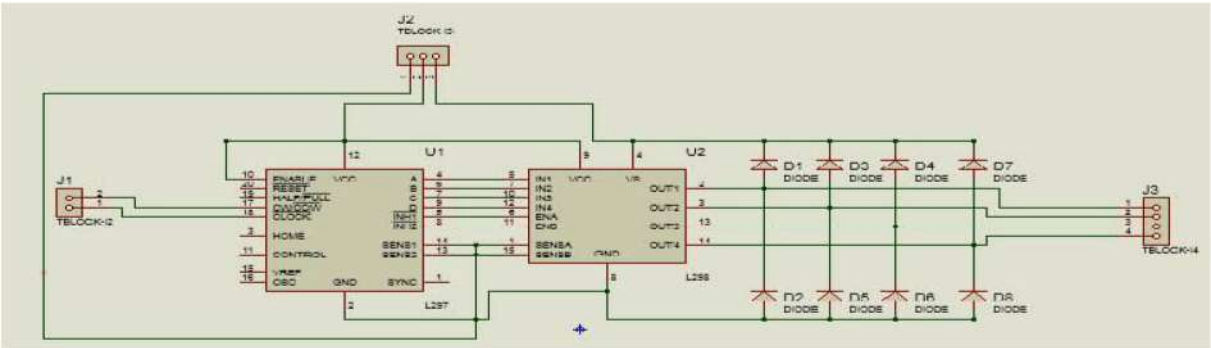
MW.15	PowerSO	Name	Function
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load
2;3	4;5	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	VS	Supply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

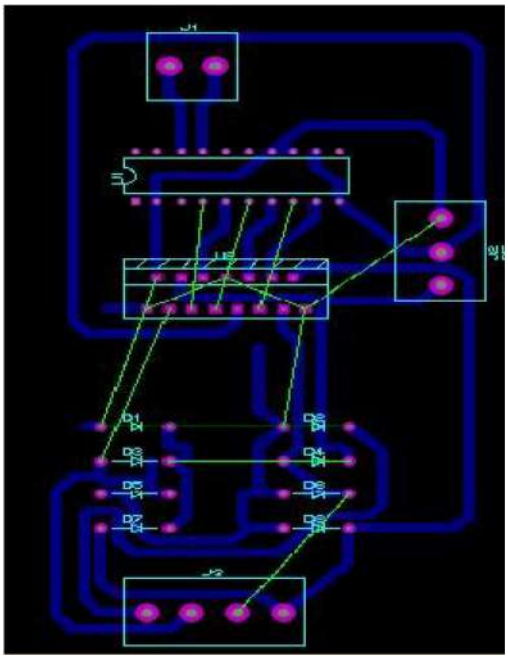


Architecture interne du L298

Annaxe-2- circuit imprimée



Isis le couple l297/l298



ARES du circuit l297/l298

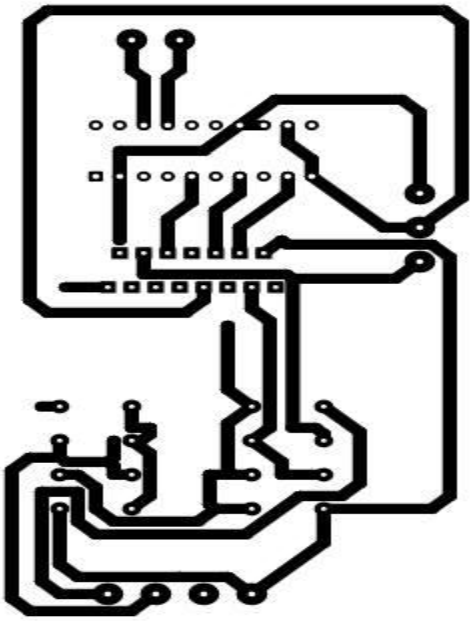


Schéma de réalisation

Annaxe-3-VSPE pour virtuel serial port

virtual Serial Port Driver est un pilote qui crée des ports série virtuels et les connecte par deux via un câble virtuel null-modem. Les applications des deux côtés d'une paire pourront échanger les données de façon que tout ce qui est écrit sur le premier port apparaîtra sur le seconde et vice versa. Tous les ports série virtuels fonctionnent et se comportent exactement comme ceux réels, en émulant tous leur paramètres. Vous pouvez créer autant de paires de ports virtuels que vous voulez, donc vous aurez toujours assez de ports série et pas besoin de matériel supplémentaire surchargeant votre bureau

- **Création du nombre illimité des ports COM virtuels**

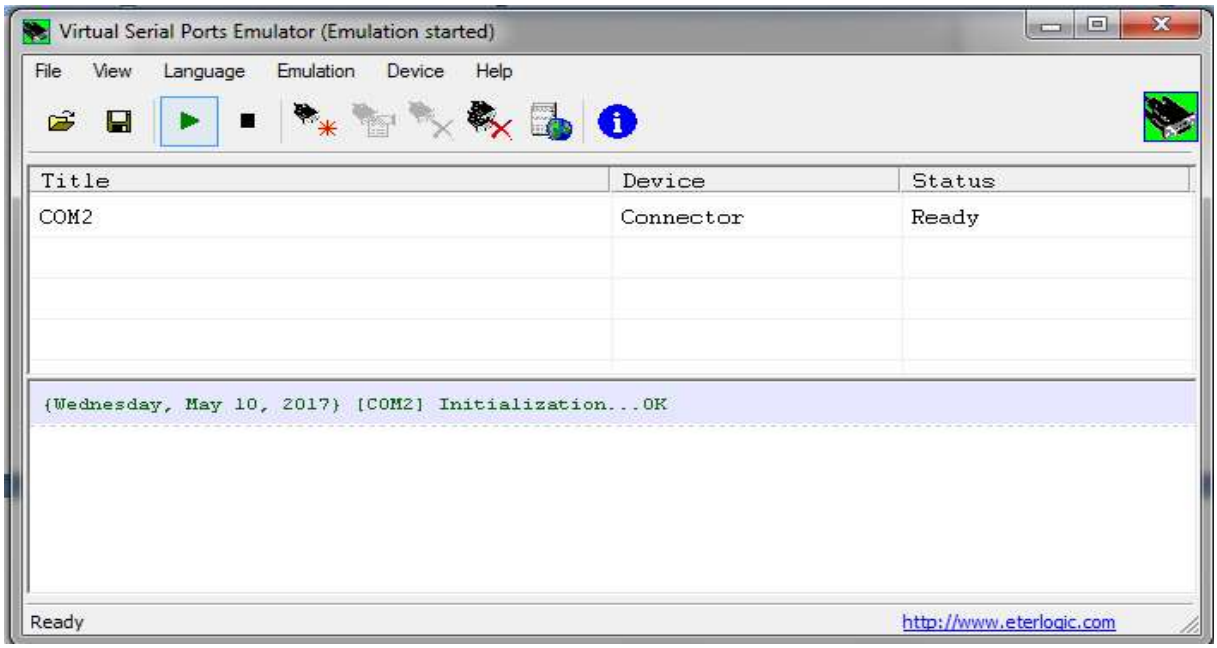
Virtual Serial Port Driver vous offre la possibilité de créer un nombre illimité des ports série virtuels. Les ports créés sont situés dans le groupe "ports" du Gestionnaire des périphériques, ce qui signifie qu'ils seront visibles et facile à accéder par tout le matériel que vous installez. Vous pourrez indiquer quelle application a un accès au port défini.

- **Émulation complète des paramètres de ports réels**

Ports virtuels, créés par Virtual Serial Port Driver, fonctionnent et se comportent exactement comme ceux réels. Ils émulent tous les paramètres de ports série réels et accordent une émulation stricte de débit en baud, une émulation de la rupture de la ligne de connexion et un support complet du contrôle HandFlow. En plus, VSPD utilise la technologie avancée du pilote noyau Windows, ce qui garantit un support de WDM, WMI, Power Management, PnP, etc. Virtual Serial Port Driver est entièrement compatible avec API COM et supporte tous les messages de Windows.

- **Gestion de toutes les lignes de signaux**

Ports série virtuels créés avec Virtual Serial Ports Driver émulent et supportent toutes les lignes de signaux matérielles standards (DTR/DSR, RTS/CTS, RING, ERROR, DCD, etc.). Comme dans la connexion null-modem réelle, DTR du port local est connecté à DSR et DCD du port distant, et RTS du port local est connecté à CTS du port distant. En plus, vous pouvez laisser le schéma de brochage par défaut ou choisir une boucle de retour ou bien personnaliser selon votre configuration matérielle.



Logiciel VSPE pour virtuel serial port