

République Algérienne Démocratique et Populaire  
Ministère de L'Enseignement Supérieur et de la Recherche Scientifique  
Université Akli Mohand Oulhadj - Bouira



FACULTE DES SCIENCES ET DES SCIENCES APPLIQUEES  
DÉPARTEMENT Génie Electrique

MEMOIRE DE FIN D'ÉTUDES  
Pour l'obtention du diplôme de  
**MASTER**

Spécialité: **Systèmes Electroniques Complexes**

**Thème:**

**Contribution à la conception et à la réalisation d'un système à base d'une commande numérique pour la fabrication des canaux micro fluidiques**

*Soutenu le : 26/09/2017*

Réalisé par :

Ouchefoun Djamel Eddine

Talhi Mohamed

**Jury:**

**Promoteurs:**

**Dr. REZKI Mohamed**  
**Dr. DERGUINI Noureddine**

**- DIB Riad**  
**- HAMOUCHE Sofiane**  
**- BENSAFIA Yassine**

**Année Universitaire : 2016/2017**

## *Dédicace*

*Nous dédions ce Modeste travail :*

- *À tous nos frères et sœurs et à toute la famille  
TALHI et OUCHFOUNE.*
- *Et à nos collègues : électroniques.*
- *Et toutes les personnes qui nous ont aidées, en  
particulier ceux que nous n'avons pas cités, ils  
resteront toujours dans notre cœur.*

*TALHI Mohamed*

*OUCHFOUNE Djamel Eddine*

# REMERCIEMENTS

*Nous tenons à remercier tout premièrement Allah le tout puissant pour la volonté, la santé et la patience, qu'il nous a donné durant toutes ces longues années.*

*Ainsi, nous tenons également à exprimer nos vifs remerciements à nos encadreurs Dr.DERGUINI Nour eddine et Dr.REZKI mohamed pour avoir d'abord proposée ce thème, pour le suivi continu tout le long de la réalisation de ce mémoire et qui n'ont pas cessée de nous donner leurs conseils.*

*Nous tenons à remercier les membres du jury pour leur présence et patience pendant la présentation de ce travail et leur critique constructive.*

*Nous tenons à remercier vivement tous les enseignants et employés du département de génie électrique à l'université de Bouira pour tous leurs aides et services qu'ils nous 'ont octroyé tout le long de notre parcours universitaire.*

*O'Uchfoune Djamel Eddine  
Talhi Mohamed*

## Liste d'abréviations

---

### Abréviations

CNC	Commande Numérique par Calculateur
MIT	(Massachusetts Institute of Technology)
MOCN	Machine-outil à Commande Numérique
CAO	Conception Assistée par Ordinateur
Om	Origine machine
Op	Origine pièce
OP	Origine Programme
PWM	pulse width modulation
M.R.V	Moteur à Réluctance variable
M.P	Moteur à Aimant permanent
MH	Moteur Hybrid
DC	moteur a courant continu
TTL	Transistor Transistor Logic
SPI	Interface Série Périphérique



# *Sommaire*

---

## *Sommaire*

Introduction générale .....1

### **Chapitre I : Généralités sur les systèmes à commande numérique**

I.1 Introduction.....2

I.2. Généralités sur la commande numérique ..... 2

I.3 Historique ..... 3

I.4. Les différentes étapes de développement de la CN ..... 3

I.5. Principe de fonctionnement d'une machine .....3

I.6. Nécessite de la CN .....5

I.6.1. Automaticité .....5

I.6.2. Flexibilité .....5

I.6.3. Sécurité .....5

I.7. les avantages et les inconvénients des CNC.....6

I.7.1 les avantages.....6

I.7.2 les inconvénients .....6

I.8. Conclusion .....6

### **Chapitre II : Description "Matériel"**

II.1 Introduction.....7

II.2 Choix du moteur .....7

II. 3 Moteur pas à pas .....7

III.3.1 Définition .....7

II.3.2 Les particularités de moteur pas à pas..... 8

II.3.3 L'intérêt du moteur pas à pas .....8

II.3.4 Constitutions d'un moteur pas à pas .....8

II.3.5 Principe de fonctionnement .....8

## ***Sommaire***

---

II.3.6 différents types de moteur pas à pas.....	9
II.3.6.1 Moteurs à réluctance variable .....	10
II.3.6.2 Moteur pas-à-pas hybride MH .....	10
II.3.6.3 Moteurs à aimant permanent .....	11
II.3.6.3.1 Moteur à aimant permanent unipolaire.....	11
II.3.6.3.1.a fonctionnement d'un moteur pas-a-pas unipolaire.....	12
II.3.6.3.2 Les moteurs bipolaires .....	13
a) Mode biphasé .....	14
b) Mode monophasé .....	14
c) Mode demi-pas .....	15
II.3.7 Propriétés et applications.....	15
II.3.8 Les avantages et les inconvénients de moteurs pas à pas .....	16
II.3.9 Comparaison entre les trois types de moteurs pas à pas .....	16
II.4 Commande des moteurs pas à pas par les driver .....	17
II.4.1 Définition d'un driver .....	17
II.4.2 le pont H .....	17
II.4.3 Commande directe avec un amplificateur L293D .....	19
II.4.3.1 Circuit L293D .....	19
II.4.3.2 présentation le shield moteur L293D.....	20
II.4.3.3 principe de fonctionnent .....	20
II.5 Carte Arduino .....	21
II.6 Histoire de la carte Arduino .....	21
II.6.1 Présentation de la carte Arduino .....	22
II.6.2 Caractéristique de carte arduino UNO.....	22
II.6.3 Pourquoi Arduino UNO .....	23
II.6.4 La constitution de la carte Arduino UNO.....	24
II.6.4.1 Partie matérielle.....	24

## ***Sommaire***

---

II.6.4.1.1 LeMicrocontrôleurATMega328.....	24
II.6.4.1.2 Les sources de l'alimentation de la carte .....	25
II.6.4.1.3 Les entrées & sorties.....	25
II.6.4.1.4 Les ports de communications.....	26
II.7 la porte série USB .....	27
II.7.1 Définition.....	27
II.7.2 caractéristiques générales .....	28
II.8 Servomoteur .....	28
II.8.1 Composition d'un servomoteur.....	29
II.8.2 Apparence.....	29
II.8.3 Connectique .....	30
II.9 Conclusion .....	31

## **Chapitre III : Description “Logiciels”**

III .1 Introduction .....	32
III .2 schéma synoptique .....	32
III .3 Logiciel « Arduino » .....	33
III .3.1 Structure générale du programme (IDE Arduino).....	33
III.3.2 description de logiciel 'ARDUINO' .....	33
III.3.2.1 Commentaires .....	33
III.3.2.2 Définition des variables .....	34
III.3.2.3 Configuration des entres et des sorties void setup () .....	34
III .3.2.4 Programmation des interactions void loop().....	34
III .3.2.5 Organigramme d'un programme arduino pour la commande des moteurs.....	34
III.4 Les étapes de téléchargement du programme .....	36
III.5 Inkscape.....	38

## ***Sommaire***

---

III.5.1 Fonctionnalités d'inkscape .....	38
III .5.2 L'espace de travail.....	39
III.5.3 Vectorisation d'une image matricielle ou un texte par inkscape.....	40
III.5.4 Génération du Gcode par inkscape .....	42
III.6 Langue de contrôleurs numériques « G-code » .....	43
III.6.1 description .....	43
III.6.1.1 Format d'une ligne .....	44
III.6.1.2 Format d'un mot .....	44
III .6.2 Les fonctions .....	45
III.7 Logiciel de pilotage de la machine CNC Processing .....	45
III.7.1 L'interface .....	46
III.7.2 L'interface de commande .....	47
III.7.3 L'organigramme pour l'interface de commande .....	48
III.8 Conclusion .....	49

## **Chapitre IV : Réalisation et tests**

IV.1 Introduction.....	50
IV.2 Simulation sous Porteuse Professionnel.....	51
IV.2.1.Schéma de montage réalise.....	51
IV.2.2 Réalisation et teste .....	53
IV.2.3 Résultats du teste et discussion .....	53
IV.3 matériel nécessaire pour réaliser la machine CNC .....	54
IV.3.1 démonté le lecteur de DVD .....	54
IV.3.2 Préparer les moteurs pas à pas des axes X, Y.et Z.....	55
IV.3.3 Les axes X et Y.....	55
IV.3.4 L'axe Z.....	56

## ***Sommaire***

---

IV.3.5	table de dessine.....	56
IV.3.6	Branchement du circuit.....	57
IV.3.7	Assemblage du CNC machin Axes X et Y et Z.....	57
IV.3.8	Test des axes X / Y/Z .....	58
IV.3.9	Des testes supplémentaire pour assurer la fiabilité du système .....	60
IV.4	Conclusion .....	61
	Conclusion générale.....	62
	Annexes.....	64
	Bibliographie.	

## Liste des figures

---

### Liste des figures

<b>Figure I.1</b> Décomposition d'une machine numérique.....	4
<b>Figure I.2:</b> machine à trois axes.....	4
<b>Figure II.1 :</b> un moteur pas à pas.....	8
<b>Figure II.2:</b> position de moteur pas à pas.....	9
<b>Figure II.3:</b> flux induit de moteur pas à pas.....	9
<b>Figure II.4:</b> Principe du moteur pas à pas.....	9
<b>Figure II.5:</b> moteur à réluctance variable.....	10
<b>Figure II.6:</b> moteur hybride.....	11
<b>Figure II.7:</b> Représentation schématique d'un moteur unipolaire.....	12
<b>Figure II.8 :</b> Représentation schématique d'un moteur bipolaire.....	13
<b>Figure II.9 :</b> Schéma de principe d'un pont en H.....	17
<b>Figure II.10 :</b> Schéma simplifié d'un pont en H réalisé avec des transistors NPN et PNP.....	18
<b>Figure II.11 :</b> commande moteur par le circuit L293D.....	19
<b>Figure II.12 :</b> shield L293D .....	20
<b>Figure II.13 :</b> description du bronchement des moteurs sur shield L293D.....	21
<b>Figure II.14 :</b> carte Arduino : (a) Arduino mini (b) Arduino UNO.....	22
<b>Figure II.15:</b> MicrocontrôleurATMega328.....	24
<b>Figure II.16 :</b> Constitution de la carte Arduino UNO.....	27
<b>Figure II.17:</b> un câble USB.....	27
<b>Figure II.18:</b> symbole de l'USB.....	28
<b>Figure II.19 :</b> servomoteurs.....	29
<b>Figure II.20:</b> image 3D de vue de l'extérieur et de l'intérieur d'un servomoteur .....	30
<b>Figure II.21 :</b> Fils de connexion d'un servomoteur.....	31
<b>Figure III.1</b> schéma synoptique.....	32

## Liste des figures

---

<b>Figure III.2:</b> interface IDE ARDUINO.....	33
<b>Figure III.3:</b> Organigramme d'un programme arduino pour la commande des moteurs .....	35
<b>Figure III.4:</b> sélection type arduino.....	36
<b>Figure III.5:</b> sélection port (COM) arduino.....	36
<b>Figure III.6:</b> téléverser code arduino.....	37
<b>Figure III.7:</b> téléversement terminé.....	37
<b>Figure III.8 :</b> La création d'un nouveau document entraîne l'affichage d'une nouvelle fenêtre possédant ses propres éléments d'interface. ....	39
<b>Figure III.9:</b> l'interface principale d'Inkscape.....	40
<b>Figure III.10 :</b> Le paramétrage général du document offre plus de possibilités s'il est effectué sur le document.....	40
<b>Figure III.11 :</b> Structure et contenu d'un programme CN (Gcode). ....	43
<b>Figure III.12 :</b> l'interface de processing .....	46
<b>Figure III.13:</b> l'interface de commande. ....	47
<b>Figure III.14:</b> L'organigramme pour l'interface de commande.....	48
<b>Figure IV.1 :</b> Schéma synoptique de système.....	50
<b>Figure IV.2:</b> Logiciel ISIS.....	51
<b>Figure IV.3:</b> schéma de montage réalisé dans ISISPROTEUS.....	51
<b>Figure. IV.4 :</b> organigramme d'un code arduino pour un moteur pas a pas.....	52
<b>Figure. IV.5:</b> circuit de teste.....	53
<b>Figure IV.6:</b> Système de déplacement linéaire récupéré sur un lecteur de DVD.....	54
<b>Figure. IV.7:</b> le support de machine.....	55
<b>Figure IV.8 :</b> Fixation des axes X et Y.....	55
<b>Figure IV.9:</b> Fixation axe Z.....	56
<b>Figure. IV.10:</b> Support base de papier.....	56
<b>Figure IV.11 :</b> montage CNC machin. ....	57

## *Liste des figures*

---

<b>Figure IV.12 :</b> CNC machin.....	57
<b>Figure. IV.13 :</b> un petit programme Gcode pour tracer un rectangle.....	58
<b>Figure. IV.14:</b> Programme GCTRRT.....	59
<b>Figure. IV.15:</b> le résultat de teste de rectangle.....	59
<b>Figure. IV.16:</b> le résultat des autres teste .....	60



## *Liste des tableaux*

---

### **Liste des tableaux**

<b>Tableau II.1.</b> Ordre des différentes commandes pour moteur unipolaire.....	13
<b>Tableau II.2 :</b> Ordre des différentes commandes pour moteur bipolaire.....	14
<b>Tableau II.3:</b> Comparaison des performances pour les trois types de moteurs pas à pas .....	16
<b>Tableau. IV.1:</b> tableau résume les résultats du test des moteurs. ....	53

### Introduction générale

Aujourd'hui, les machines à commande numérique par ordinateur (CNC) permettent une production économique et rentable car le contrôle des coûts reste une préoccupation importante.

Durant ces dernières années, la commande des machines électriques a subi des progrès significatifs. Ces progrès sont essentiellement dus à la révolution technologique en informatique industrielle, ce qui a permis le développement de solutions numériques efficaces avec une possibilité d'implanter des algorithmes plus complexes. Ces commandes sont en majorité basées sur les microprocesseurs, les DSP (Digital Signal Processor) et les microcontrôleurs PIC. Ces processeurs sont équipés d'unité arithmétique et logique (UAL) dédiée à la réalisation des calculs arithmétiques. Ils intègrent également des périphériques tel que les convertisseurs analogiques/numériques et les « timers » bien adaptés aux besoins de commande de machines électriques.

Le moteur pas à pas est l'interface idéale entre l'électronique numérique et la mécanique, il permet de convertir directement un signal électrique en un positionnement angulaire à caractère incrémental. Pour cela on utilise le port USB d'un ordinateur de type PC qu'on commande avec une carte Arduino Uno et des circuits intégrés.

Le travail est structuré comme suit:

Le premier chapitre, touche les généralités sur la commande numérique et leur parcours de développement, ainsi que la décomposition d'une machine outil à commande numérique et son principe de fonctionnement.

Le deuxième chapitre est consacré à la description des composants électroniques en vue d'une étude sur les moteurs pas à pas, la carte Arduino Uno et les circuits intégrés (L293D, Schield L293D).

Le troisième chapitre concerne l'étude de la programmation et interface graphique avec son application à la machine CNC. Il comporte aussi une présentation du logiciel Inkscape qui est un logiciel de dessin vectoriel générant un fichier data (Gcode) pour la commande numérique des moteurs, ce fichier doit être transféré par une interface graphique appelée Processing.

Dans le dernier chapitre (chapitre 4), on présente le prototype expérimental utilisé pour tester en temps réel les algorithmes de commande développés dans les parties précédentes. Un exemple de contrôle d'un moteur pas à pas est traité dans un but de montrer l'application de cette commande dans notre cas. La dernière partie de ce mémoire englobe une conclusion générale et les perspectives du travail.

## I.1. Introduction

Apparue il y a seulement quelques dizaines d'années, la commande numérique (CN) impose actuellement sa technologie dans le monde de l'usinage. Elle est conçue pour piloter le fonctionnement d'une machine à partir des instructions d'un programme indépendamment de l'opérateur pendant son exécution. Elle a, au début permis l'automatisation des machines-outils traditionnelles. [1]

La CN est également à l'origine de nouvelles conceptions de machines polyvalentes comme le centre d'usinage, par exemple.

Aujourd'hui, de plus en plus étroitement associée aux progrès de la micro-électronique et de l'informatique, la CN voit ses performances et sa convenabilité augmenter régulièrement tandis que, en revanche, son prix et son encombrement ne cessent de diminuer.

Parmi les objectifs des développements technologiques aux quels est parvenu l'être humain dans l'industrie mécanique et ce a travers les différentes phases et époques a commence par la révolution industrielle qui avait débuté par la création des outils de l'industrie traditionnelle et autres, et ensuite ces outils furent développés en machines semi-automatiques puis en machines automatiques qui fonctionnaient a l'aide de ordinateur et programmes spéciaux qui faisait fonctionner l'opération (d'industrialisation) transférant les instructions de la partie commande vers la partie opérationnelle de la machine par le biais de câbles spéciaux de connexion (câbles DB 9 ,15 et 25) avec les différent sorties de communication comme exemple RS 232 , RS 485 . [ 1]

## I.2. Généralités sur la commande numérique

La CN est une technique utilisant des données composées de codes alphanumériques pour représenter des instructions géométriques et technologique nécessaires à la conduite d'une machine ou d'un procédé. C'est également une méthode d'automatisation des fonctions des machines ayant pour caractéristiques principale une très grande facilité d'adaptation à des travaux différents. À ce titre, la CN constitue d'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 CN 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 CN commercialisés actuellement contenant au moins un microprocesseur, les termes CN et CNC peuvent être considérés comme des synonymes. Pour des raisons de simplicité. [2]

## I.3. Historique

Les travaux menés par Falcon et Jacquard à la fin du XVIII<sup>e</sup> siècle ont démontré qu'il était possible de commander les mouvements d'une machine à partir d'informations transmises par une carte perforée. Cette technique fut étendue à leurs métiers à tisser de 1805 lequel équipement est considéré comme l'ancêtre de la commande numérique après quoi l'exploitation industrielle de la CN devient rattachée 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.

Au printemps 1949, il confie alors au Massachusetts Institute of Technology (MIT) le soin de développer des asservissements capables de piloter une machine (fraiseuse prototype Cincinnati à broche verticale présentée en septembre 1952 dans le Servomechanisms Laboratory du MIT) qui recevra des instructions intermittentes à partir d'un lecteur de cartes. Il faut encore attendre quelques années de vastes fonds de l'US Air Force et l'appui des chercheurs du MIT pour rendre la première MOCN réellement opérationnelle [1]

## I.4. Les différentes étapes de développement de la CN

Sont les suivantes:

1954 : Bendix acquiert le brevet de Parsons et fabrique la première CN industrielle.

1955 : à Font du Lac (Wisconsin), le constructeur américain Giddins & Lewis commercialise la première MOCN.

1959 : apparition de la CN en Europe (foire de Hanovre).

1964 : en France, la Télémécanique Électrique lance la CN NUM 100 conçue à base de relais Téléstatic.

1968 : la CN adopte les circuits intégrés ; elle devient plus compacte et plus puissante.

1972 : les mini calculateurs remplacent les logiques câblées ; la CN devient CNC.

1976 : développement des CN à microprocesseurs.

1984 : apparition de fonctions graphiques évoluées et du mode de programmation conversationnel.

1986 : les CN s'intègrent dans les réseaux de communication, début de l'ère de la fabrication flexible (CIM).

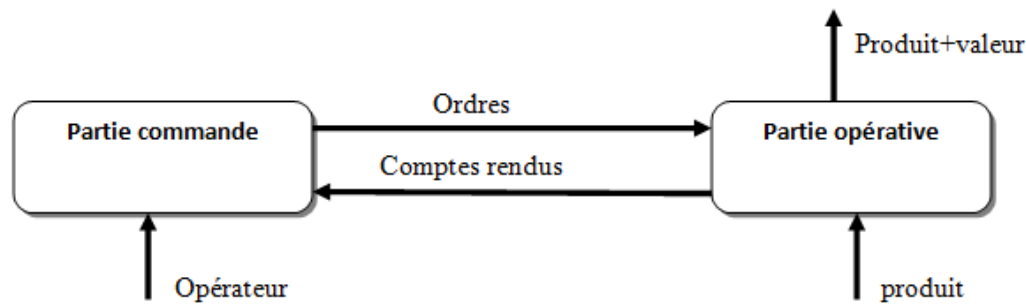
1990 : développement des CN à microprocesseurs 32 bits. [1]

## I.5. Principe de fonctionnement d'une machine numérique [3]

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

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

- La partie opérative.
- La partie commande.



**Figure I.1** Décomposition d'une machine numérique. [3]

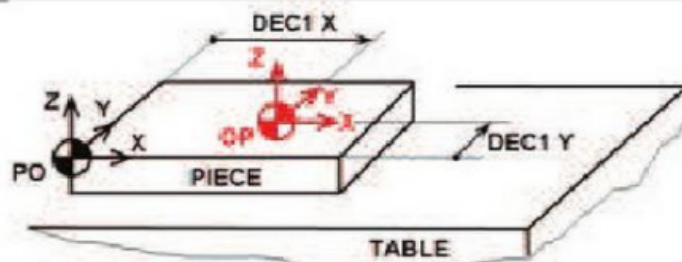
La partie opérative comporte les axes de déplacement et la tête.

La partie commande permet de piloter la partie opérative.

Elle est composée d'un ordinateur (CNC) et d'éléments électroniques capables de piloter les moteurs : les cartes d'axes. Des ordres vont être générés vers la commande par le biais d'un code machine ou par action manuelle de l'opérateur. La commande va traiter ces informations et générer des consignes afin d'obtenir les déplacements voulus par le biais des moteurs d'axes. Des contrôles de vitesse et de position seront alors effectués de manière continue par la machine.

Il existe 3 principaux types de référentiels à prendre en compte (figure I.2):

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



**Figure I.2:** machine à trois axes [3]

## I.6. Nécessite de la CN [1]

### I.6.1. Automaticité

Le premier avantage d'une CN est d'offrir aux machines qui en sont équipées un très haut niveau d'automaticité. Sur de telles machines, l'intervention de l'opérateur nécessaire pour assurer la production des pièces peut être considérablement réduite voire supprimée.

Cette caractéristique présente certain nombre d'avantage par exemple : diminution notable de la fatigue de l'opérateur, moins d'erreurs d'origine humaine et un temps d'usinage constant et prévisible pour chaque pièce d'une même série.

### I.6.2. Flexibilité

Puisqu'elles sont pilotées à partir d'un programme, les MOCN peuvent usiner des pièces différentes aussi facilement que l'on charge un nouveau programme. Une fois vérifié puis exécuté pour la première série, ce programme peut être facilement rappelé lorsque la même série se représente.

Une MOCN se caractérise en outre par des temps de réglage très courts qui répondent parfaitement aux impératifs de la production en flux tendus.

La grande souplesse d'utilisation de la CN entraîne une quantité non négligeable d'autres avantages :

- Changement aisé du programme d'usinage des pièces.
- Réduction des outillages et suppression des gabarits.
- Diminution du nombre des outils.
- Réduction des temps de réparation et de réglage du poste de travail.
- Définition plus rapide et plus faible des conditions optimales d'usinage.
- Réduction du nombre de prises des pièces du fait de l'universalité de la machine. Diminution du temps d'attente entre les diverses machines d'usinage d'un atelier.
- Gain sur les surfaces au sol occupées dans l'atelier.
- Possibilité de réaliser des pièces complexes en gérant des déplacements simultanés sur plusieurs axes.
- Contrôle automatique des outils et des dimensions des pièces avec prise en compte par la CN des corrections à effectuer.

### I.6.3. Sécurité

La CN à beaucoup contribue à améliorer la sécurité des machines :

- En premier lieu, parce qu'elle connaît très précisément l'enveloppe de travail dans

Laquelle doivent évoluer ;

- Ensuite, parce qu'elle permet une simulation graphique hors usinage des programmes nouvellement créés pour vérification et détection des risque éventuels de collision ;

- Enfin, parce qu'en exerçant une surveillance permanente de l'usinage en cours, elle peut décider d'en interrompre le déroulement et d'alerter l'opérateur en cas d'incident ;

Il est par ailleurs admis que le niveau de performances très élevé atteint par la MOCN conduit les constructeurs à prévoir des dispositifs de protection très élaborés qui ne s'imposent par nécessairement sur une MO conventionnelle.

### **I.7. les avantages et les inconvénients des CNC [4]**

#### **I.7.1 les avantages**

- Réduction du temps improductif
- Accroissement du degré de précision et de répétabilité
- Réduction des taux de rebuts
- Réduction de la nécessité de contrôle
- Possibilité de traiter des géométries complexes
- Facilité d'adapter les changements techniques
- Réduction des temps de mise en production Réduction des stocks
- Réduction des espaces requis
- Réduction du niveau de qualification des opérateurs

#### **I.7.2 les inconvénients**

- Coûts d'investissement élevés
- Plus d'effort doit être fourni pour la maintenance
- Nécessité de programmation des équipements de CN
- Une plus grande utilisation des équipements à CN doit être faite pour maximiser leur rentabilité

### **I.8. Conclusion**

A travers ce chapitre on a donné notions générales sur la commande numérique avons de passer à la conception détaillés pour la commande numérique de système.

## II.1 Introduction

Dans ce chapitre nous présentons une description détaillée de chaque partie du système nécessaire pour la réalisation d'un système à base d'une commande numérique(CN) afin d'obtenir une schématisation complète et précise.

## II.2 Choix du moteur

Pour ce type de construction nous retrouvons de manière générale une sorte de motorisation c'est les moteurs pas à pas.

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

Les moteurs pas à pas étant moins cher et plus accessible nous n'envisagerons ce type de motorisation. [4]

## II. 3 Moteur pas à pas

### III.3.1 Définition

Le moteur pas à pas constitue un convertisseur électromécanique destiné à transformer le signal électrique (impulsion ou) en déplacement (angulaire ou linéaire) mécanique.

Au point de vue électrotechnique, le moteur classique ressemble à la machine synchrone, dont le stator (le plus souvent à pôles saillants) porte les enroulements de pilotage et le rotor (presque toujours à pôles saillants) est soit muni d'aimants permanents (structure dite polarisée ou active), soit constitué par une pièce ferromagnétique dentée (structure dite réluctante ou passive).

- Entre le moteur et son alimentation, sont intercalés trois éléments essentiels une unité de calcul, qui élabore les impulsions de commande.
- un modulateur PWM, qui génère les commandes des contacteurs électroniques de commutation.
- une électronique de commutation (puissance), qui, à partir d'une alimentation, fourni l'énergie vers les enroulements appropriés du moteur [5]





**Figure II.1** : un moteur pas à pas [5]

### II.3.2 Les particularités de moteur pas à pas

- La caractéristique de l'énergie électrique est un train d'impulsion continue.
- Le moment mécanique est un déplacement angulaire.
- Très faible vitesse de rotation. [4]

### II.3.3 L'intérêt du moteur pas à pas

- Positionnement plus précis
- Possibilité de bloquer la rotation dans une position très précise. [4]

### II.3.4 Constitutions d'un moteur pas à pas

Comme tous les moteurs, il est constitué par :

- **Une partie fixe** : C'est le stator, formé d'un circuit magnétique et des bobines (phases) dont le rôle est de créer un flux magnétique à directions multiples
- **Une partie mobile** : C'est le rotor, placé dans le flux du stator il se positionne suivant le flux maximum. [4]

### II.3.5 Principe de fonctionnement [5]

La circulation d'un courant électrique dans un bobinage entraîne l'apparition d'un champ magnétique, comme le détaille la figure de droite dans le cas du solénoïde, et donc la présence de pôles Nord et Sud (deux pôles de même nature se repoussent, deux pôles Nord et Sud s'attirent) ; c'est sur ce principe de base que repose le fonctionnement de tout moteur électrique, et, de

manière plus générale, de bon nombre de dispositifs électro-mécaniques : relais, compteurs, galvanomètres, certains hautparleurs ou microphones, gâches électriques de porte, etc.

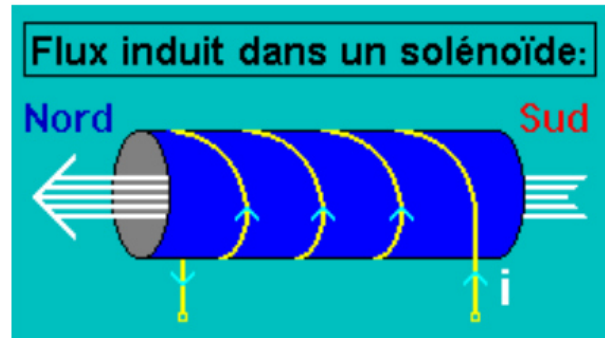
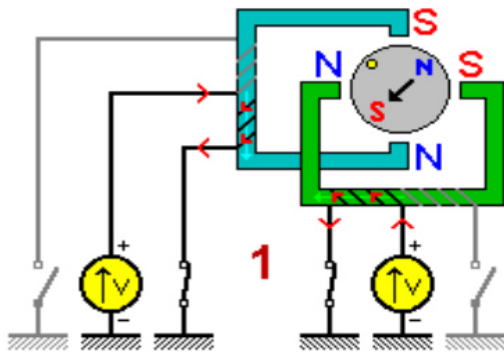


Figure II.2: position de moteur pas à pas [5]

Figure II.3: flux induit de moteur pas à pas [5]

### II.3.6 différents types de moteur pas à pas

Les moteurs pas à pas sont classés selon le principe physique à l'origine de l'interaction stator – rotor voire la (Figure II.2) ; il existe trois types de moteur pas à pas.

- Moteur à Réductance variable: Il utilise le principe du flux maximum (M.R.V).
- Moteur à Aimant permanent : Il utilise le principe de l'action d'un champ magnétique sur un aimant (M.P).
- Moteur Hybride : C'est la superposition des deux principes ci-dessus (MH). [6]

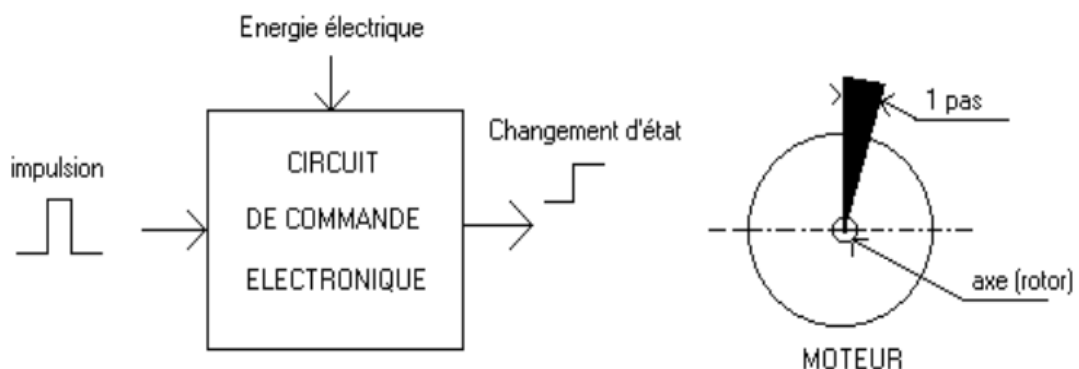


Figure II.4: Principe du moteur pas à pas [6]

### II.3.6.1 Moteurs à réluctance variable

Le rotor est en fer doux et comporte un nombre de pôles différent du stator. Le rotor se déplace pour que le flux le traversant soit maximum. Ces moteurs n'ont de couple de maintien si aucune bobine n'est alimentée.

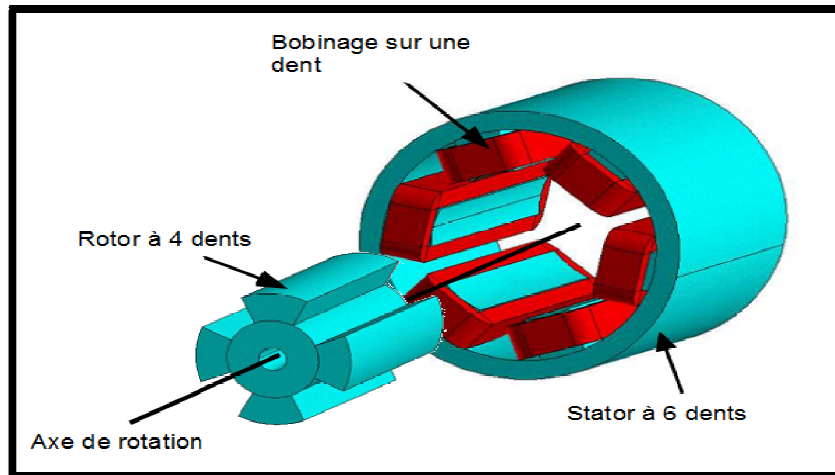
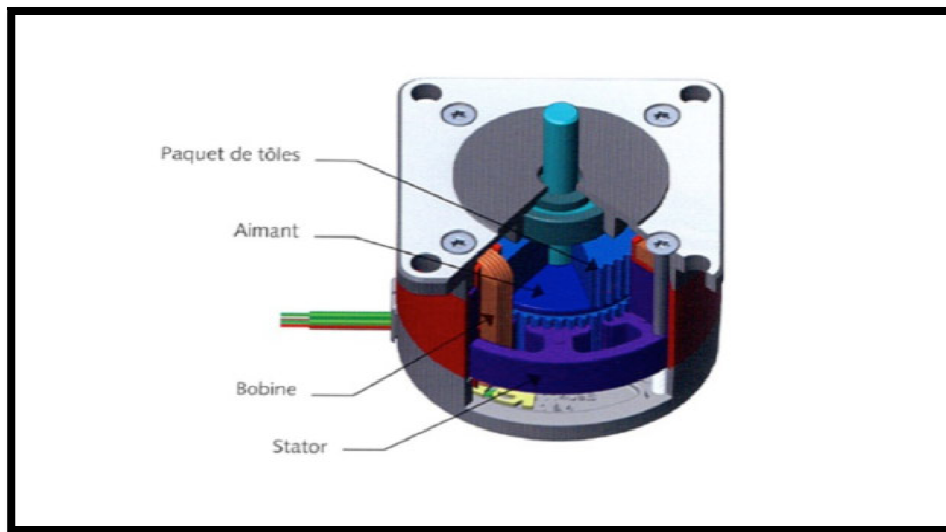


Figure II.5: moteur à réluctance variable [4]

- **Caractéristique principaux**
  - Les fréquences de fonctionnement peuvent être élevées.
  - Bonne résolution.
  - Construction simple.
  - Absence de couple résiduel avec le moteur hors tension. [4]

### II.3.6.2 Moteur pas-à-pas hybride MH

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



**Figure II.6:** Moteur hybride [4]

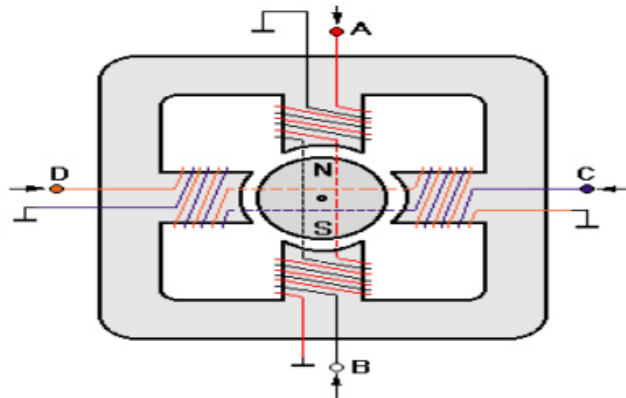
- **Caractéristique principale**
  - Hors tension, le rotor est maintenu en position
  - Bonne précision de la position du rotor
  - Grande vitesse de rotation [4]

### II.3.6.3 Moteurs à aimant permanent

Les moteurs à aimant permanent sont constitués d'un stator supportant les bobinages et d'un rotor magnétique (aimant bipolaire). Cette catégorie de moteurs se subdivise en deux types : le moteur unipolaire et le moteur bipolaire. [7]

#### II.3.6.3.1 Moteur à aimant permanent unipolaire [3]

Les moteurs unipolaires se différencient par le fait qu'ils sont à double bobinage. Le double bobinage est utilisé pour l'inversion du flux statorique et le moteur se commande de la même manière qu'un bipolaire excepté qu'un seul transistor pour chaque enroulement suffit dans l'étage de puissance (soit quatre Darlington pour un moteur ou un réseau de 4 transistors).



**Figure II.7:** Représentation schématique d'un moteur unipolaire [3]

Simplement, les moteurs unipolaires sont plus chers car leur fabrication réclame un double bobinage. De plus, pour une taille donnée, ce type de moteur a un couple plus faible à cause des enroulements qui sont plus fins. Il fut une époque où les moteurs unipolaires étaient intéressants pour les concepteurs parce qu'ils simplifiaient l'étage de commande électronique.

Maintenant, grâce aux circuits de commande (push pull monolithique) du genre L298, les moteurs bipolaires sont devenus 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.

#### II.3.6.3.1.a fonctionnement d'un moteur pas-à-pas unipolaire

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

Dans les schémas voire (**Figure II.7**), 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. [9]

Comme pour chaque type de moteur, le modèle unipolaire peut être commandé en mode monophasé, biphasé ou demi-pas. Le (**tableau II.1**) donne l'ordre des différentes commandes. [7]

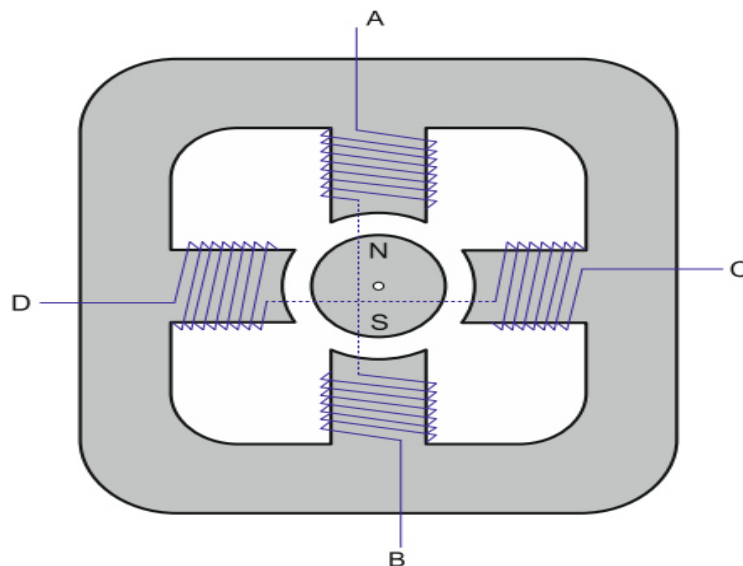
Mode monophasé	Mode biphasé	Mode demi-pas
<b>0001</b>	0101	0001
<b>0100</b>	0110	0101
<b>0010</b>	1010	0100
<b>1000</b>	1001	0110
		0010
		1010
		1000
		1001

**Tableau II.1.** Ordre des différentes commandes pour moteur unipolaire [7]

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

### II.3.6.3.2 Les moteurs bipolaires [7]

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



**Figure II.8 :** Représentation schématique d'un moteur bipolaire [7]

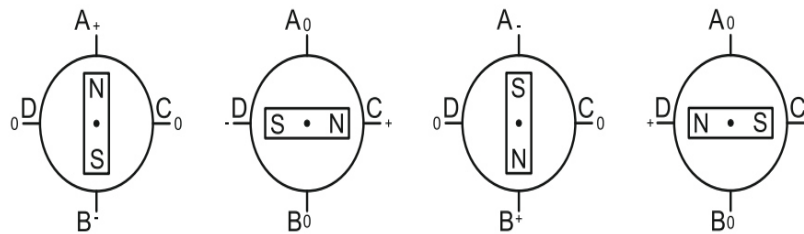
Comme pour le modèle précédent, ce moteur peut être alimenté sous trois Mode différentes (tableau II.2).

Mode monophasé	Mode biphasé	Mode demi-pas
AB	AB-CD	AB
CD	BA-CD	AB-CD
BA	BA-DC	CD
DC	AB-DC	BA-CD
AB	AB-CD	BA
etc..	etc..	BA-DC
		DC
		AB-DC
		AB
		etc..

Tableau II.2 : Ordre des différentes commandes pour moteur bipolaire [7]

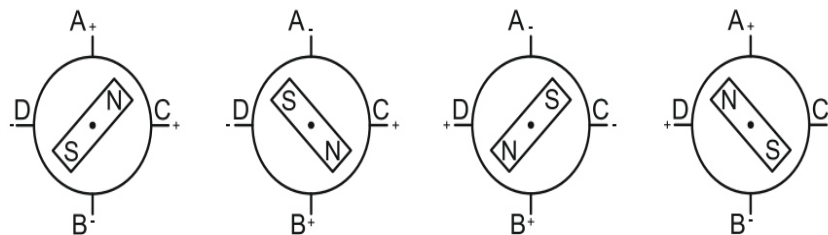
a) Mode monophasé

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



b) Mode biphasé

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



## c) Mode demi-pas

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

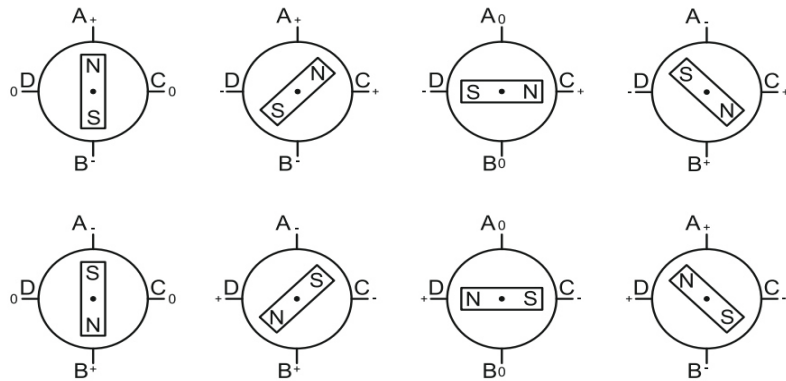


Figure II.9 : Séquences de commande d'un moteur bipolaire [7]

## II.3.7 Propriétés et applications [10]

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

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

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

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

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



### II.3.8 Les avantages et les inconvénients de moteurs pas à pas [11]

➤ **Avantages**

Rotation constante pour chaque commande (précision meilleure que 5% d'un pas).

- Existence de couple à l'arrêt.
- Contrôle de la position, de la vitesse et synchronisation de plusieurs moteurs (pas de besoin de contre-réaction).
- Moteur sans balais.

➤ **Inconvénients**

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

### II.3.9 Comparaison entre les trois types de moteurs pas à pas [10]

Cette comparaison peut se résumer selon le tableau suivant :

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 - du sens du courant dans les bobines	Dépend : - de l'ordre d'alimentation des phases - du sens du courant dans les bobines
Fréquence de travail	Grande	Faible	Grande
Puissance Quelques	Quelques Watts	dizaines de Watts	Quelques KWatts
Inconvénients Pas de	mémoire de position		

**Tableau II.3:** Comparaison des performances pour les trois types de moteurs pas à pas

## II.4 Commande des moteurs pas à pas par les driver

### II.4.1 Définition d'un driver

Drivers ou contrôleurs de moteur sont des circuits spécialisés nécessaires pour transmettre une puissance à un moteur électrique. Certains vont seulement se charger de l'alimentation, d'autres vont gérer des consignes précises : vitesse, position, couple, etc.

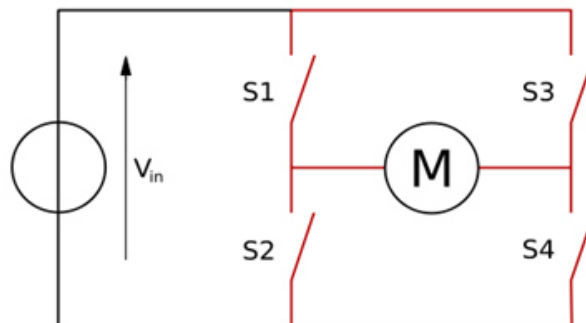
Mais comme il existe différents types de moteurs électriques (courant continu ou pas à pas, avec ou sans balais, bipolaire ou unipolaire, ...) il y a aussi différents types de contrôleurs.

Avant de faire le choix, il faut d'abord bien identifier son moteur, puis définir son besoin (contrôle en vitesse, contrôle en position) et en fin définir ses priorités (prix, qualité, précision, temps passé, fiabilité, longévité, consommation, etc...).[4]

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

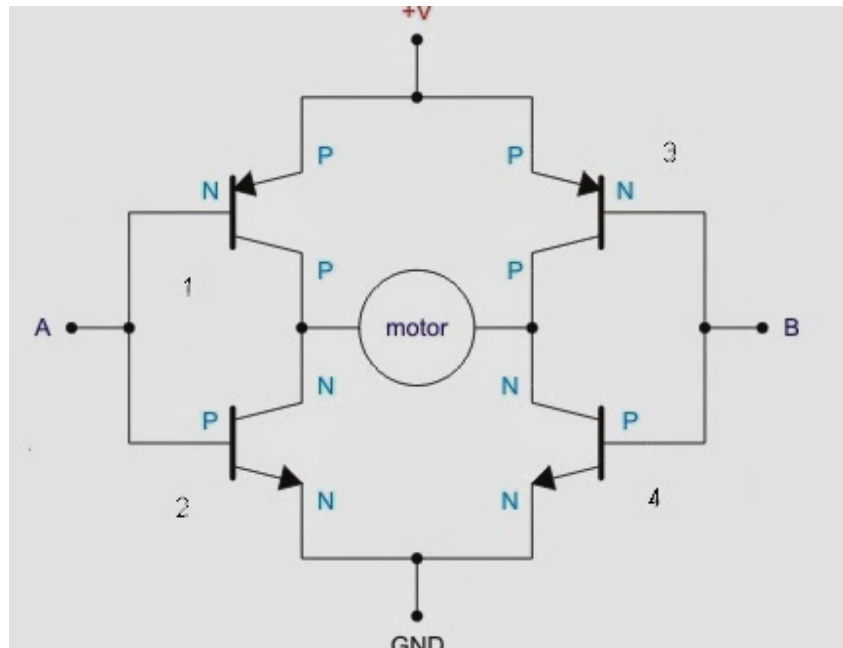
### II.4.2 le pont H

Le pont en H est un «convertisseur» qui traduit ces états logiques haut ou bas en source d'énergie de haute tension ou de basse tension.



**Figure II.10 :** Schéma de principe d'un pont en H [12]

Ci-dessus un schéma de principe de pont en H (il a la forme d'une lettre « H »). Un pont en H a deux modes de travail : (S1 S4 fermés, S2 S3 ouverts) et (S2 S3 fermés, S1 S4 ouverts). Dans le premier mode, le courant circule vers la droite à travers le moteur et dans le second mode, le courant circule vers la gauche à travers le moteur. En réalité, ce pont en H n'est jamais utilisé tel quel. On préfère utiliser des transistors comme interrupteurs. Voir figure ci-dessous :



**Figure II.11** : Schéma simplifié d'un pont en H réalisé avec des transistors NPN et PNP[12]

- Lorsque A est au niveau logique bas (0V) et B au niveau logique haut (+ V), alors les transistors 1 et 4 conduisent et 2 et 3 sont bloqués.
- Lorsque A est au niveau logique haut (+ V) et B au niveau logique bas (0V), alors les transistors 1 et 4 sont bloqués tandis que 2 et 3 conduisent.
- Lorsque A et B sont au niveau logique haut, 2 et 4 conduisent, 1 et 3 sont bloqués, le moteur se bloque.
- Lorsque A et B sont au niveau logique bas, 1 et 3 sont conducteurs, 2 et 4 sont bloqués, le moteur se bloque.

Par conséquent, en fixant A et B à un niveau logique haut ou bas, nous pouvons contrôler la direction du courant à travers une charge. Pour chaque moteur pas à pas 4 fils/2 phases, il y a deux bobines indépendantes que nous devons contrôler. [12]

Il faut donc deux ponts H pour commander un moteur pas à pas.

Le pont en H peut être implémenté par des relais des transistors ou un circuit intégré dédiée. Dans le cadre de cette séquence, le circuit L293D de Texas Instrument et qui contient deux ponts en H sera utilisé mais il existe d'autres références, comme le LN293 ou le LN298.

### II.4.3 Commande directe avec un amplificateur L293D

Pour cette commande on utilise un circuit de commande qui donne un mode de commande directe de moteur pas à pas (mode à pas entier). Mode à pas entier, deux phases alimentées en même temps (Two Phase ON, Full Step) :1010/0110/0101/1001.

#### II.4.3.1 Circuit L293D

Le L293 et L293D sont des pilotes demi-H quadruples haute intensité. Ces dispositifs sont conçus pour conduire un large éventail de charges inductives tel que des relais, des solénoïdes, DC et moteurs pas à pas bipolaires, ainsi que d'autres charges à haute intensité et à haute tension. Toutes les entrées sont compatibles TTL et tolérant jusqu'à 7 V. Chaque sortie est un circuit d'entraînement totem complète, avec un transistor Darlington évier et une source pseudo-Darlington. Les pilotes sont activés par paires, avec des pilotes 1 et 2 activées par 1,2EN et 3 et 4 activés par 3,4EN. Quand une entrée de validation est Haute, les pilotes associés sont activés, et leurs sorties sont actives et en phase avec leurs entrées. Lorsque l'entrée de validation est basse, ces pilotes sont désactivés, et leurs sorties sont éteints et dans l'état de haute impédance. Avec les entrées appropriées de données, chaque paire de conducteurs forme un pont en H d'entraînement réversible adapté aux applications des électrovannes ou moteurs. Sur le L293, des diodes de blocage externes à grande vitesse doivent être utilisés pour la suppression des décharges transitoires inductive. Sur le L293D, ces diodes sont intégrées pour réduire la complexité du système et la taille globale du système. Un terminal de VCC1, séparée de VCC2, est prévu pour les entrées logiques pour minimiser la dissipation de puissance périphérique. Le L293 et L293D sont caractérisées pour le fonctionnement de 0 ° C à 70 ° C. [13]

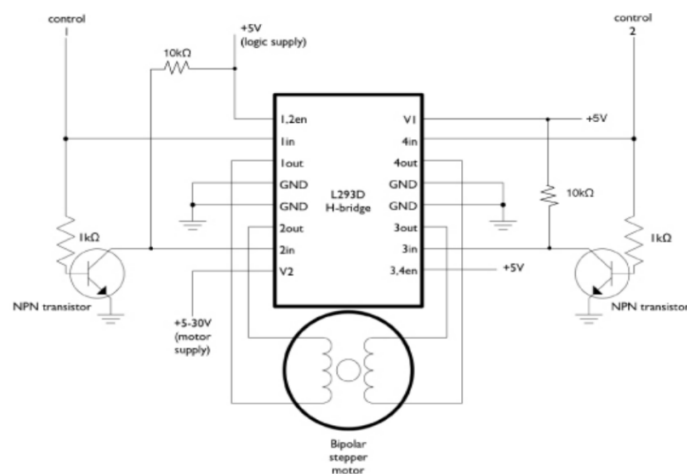


Figure II.12: commande moteur par le circuit L293D [13]

Il faut donc deux circuit L293D pour commander deux moteurs pas à pas.

### II.4.3.2 PRÉSENTATION le shield moteur L293D

Lorsque l'on souhaite utiliser une motorisation dans un projet utilisant la technologie Arduino, on se rend compte qu'il existe de nombreuses solutions techniques pour mener à bien son projet.

Dans le cas du projet de la commande du système numérique, on propose d'étudier différentes solutions techniques pour la motorisation, comme l'utilisation de servomoteurs à rotation continue ou l'utilisation de moto-réducteurs à courant continu. Très vite, on s'aperçoit que si la solution du servomoteur est très facile à mettre en œuvre avec les élèves, la solution du moteur à courant continu est bien plus difficile à mettre en œuvre avec une carte Arduino. En effet, elle nécessite des connaissances (pont en H entre autre) qui ne sont pas à la portée des élèves. D'où l'utilisation d'un shield moteur comme celui présenté ici qui permet de s'affranchir de ces connaissances et rend la mise en œuvre très simple. [14]

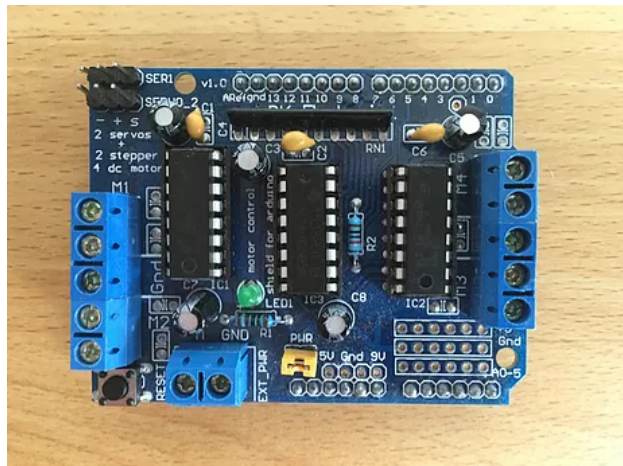


Figure II.13 : Shield L293D [14]

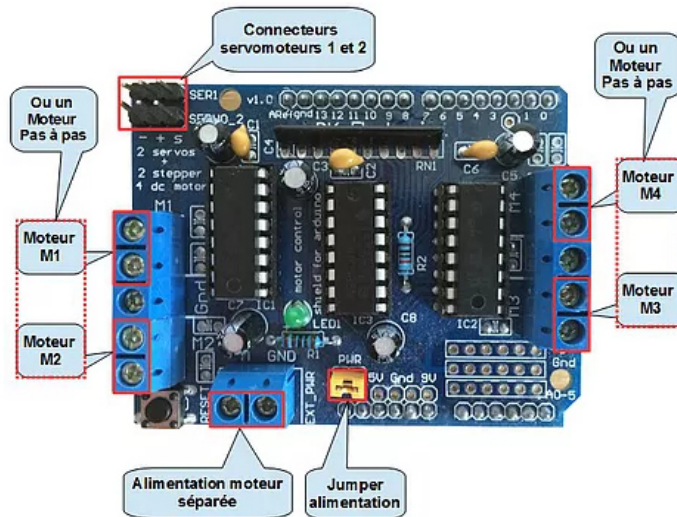
### II.4.3.3 principe de fonctionnement

Ce shield s'enfiche directement sur une carte Arduino UNO.

Il permet de piloter en même temps grâce à quatre ponts en H :

- 4 moteurs à courant continu ou 2 moteurs pas à pas.
- 2 servomoteurs standards (Utilité limitée, sachant que l'on peut piloter directement les servomoteurs depuis la carte Uno).

Pour l'alimentation électrique des moteurs, il existe deux possibilités, soit en passant par l'alimentation de la carte Arduino ou soit par une alimentation séparée. [14]



**Figure II.14:** description du branchement des moteurs sur shield L293D [14]

## II.5 Carte Arduino

Les cartes Arduino sont conçues pour réaliser des prototypes et des maquettes des cartes électroniques pour l'informatique embarquée. Ces cartes permettent un accès simple et peu coûteux à l'informatique embarquée. De plus, elles sont entièrement libres de droit, autant sur l'aspect du code source (Open Source) que sur l'aspect matériel (Open Hardware). Ainsi, il est possible de refaire sa propre carte Arduino dans le but de l'améliorer ou d'enlever des fonctionnalités inutiles au projet. Le langage Arduino se distingue des langages utilisés dans l'industrie de l'informatique embarquée par sa simplicité. En effet, beaucoup de bibliothèques et de fonctionnalités de base occultent certains aspects de la programmation de logiciel embarquée afin de gagner en simplicité. [15]

## II.6 Histoire de la carte Arduino [15]

Le projet Arduino est né en hiver 2005. Massimo Banzi enseigne dans une école de design à Ivrea en Italie, et souvent ses étudiants se plaignent de ne pas avoir accès à des solutions bas prix pour accomplir leurs projets de robotique. Banzi en discute avec David Cuartielles, un ingénieur Espagnol spécialisé sur les microcontrôleurs.

Ils décident de créer leur propre carte en embarquant dans leur histoire un des étudiants de Banzi, David Mellis qui sera chargé de créer le langage de programmation allant avec la carte. En deux jours David écrira le code, trois jours de plus et la carte était créée. Tout

le monde arrive à en faire quelque chose très rapidement sans même avoir de connaissances particulière ni en électronique ni en informatique : réponse à des capteurs, faire clignoter des leds, contrôler des moteurs ... etc. ils publient les schémas, investissent 3000 euros pour créer les premiers lots de cartes .

### II.6.1 Présentation de la carte Arduino

L'Arduino est une carte électronique en matériel libre pour la création artistique interactive, elle peut servir : pour des dispositifs interactifs autonomes simple, comme interface entre capteurs/actionneurs et ordinateur et comme programmeur de certains microcontrôleurs.



**Figure II.15** : carte Arduino : (a) Arduino mini (b) Arduino UNO [15]

Une carte Arduino, comme toutes les cartes à microcontrôleur, permet de piloter un système de manière interactive à partir du programme que l'on aura défini et mis dans sa mémoire. Par exemple gérer automatiquement l'ouverture d'une porte de garage, envoyer un SMS quand le jardin est trop sec et gérer le système d'arrosage à distance, piloter un nouveau robot. Il faut pour cela associer à la carte Arduino des capteurs comme capteur de lumière, de température et de position. Et des actionneurs comme moteurs et pompe, ainsi des organes de sortie comme lampe et chauffage. On peut associer aussi des circuits de puissance, une alimentation (piles, panneaux solaire, ...), des interfaces de dialogue (boutons, leds, écran, ...), des interfaces de communication (réseau filaire, réseau sans fil, ...) .[15]

### II.6.2 Caractéristique de carte arduino UNO

Un des modèles les plus répandus de carte Arduino est l'Arduino UNO (voir FigII.15). C'est la première version stable de carte Arduino. Elle possède toutes les fonctionnalités d'un microcontrôleur classique en plus de sa simplicité d'utilisation.

Elle utilise une puce ATmega328P cadencée à 16Mhz. Elle possède 32ko de mémoire flash destinée à recevoir le programme, 2ko de SRAM (mémoire vive) et 1 ko d'EEPROM (mémoire morte destinée aux données). Elle offre 14 pins (broches) d'entrée/sortie numérique (données acceptées 0 ou 1) dont 6 pouvant générer des PWM (Pulse Width Modulation, détaillé plus tard). Elle permet aussi de mesurer des grandeurs



analogiques grâce à ces 6 entrées analogiques .Chaque broche est capable de délivrer un courant de 40mA pour une tension de 5V.

Cette carte Arduino peut aussi s'alimenter et communiquer avec un ordinateur grâce à son port USB .On peut aussi l'alimenter avec unes alimentations comprise en 7V et 12V grâce à sa connecteur. [16]

### II.6.3 Pourquoi Arduino UNO [17]

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 microcontrôleurs 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 programmeurs 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, ATMEGA328,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.

## II.6.4 La constitution de la carte Arduino UNO

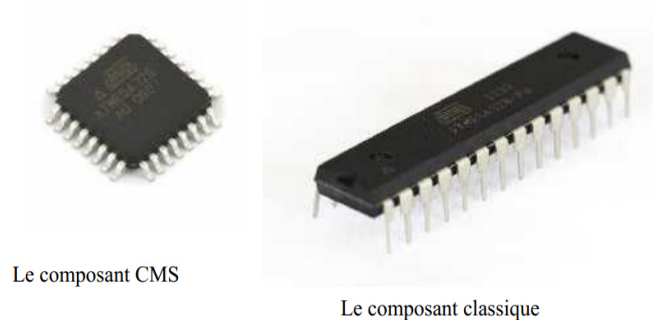
Un module Arduino est généralement construit autour d'un microcontrôleur ATMEL AVR, et de composants complémentaires qui facilitent la programmation et l'interfaçage avec d'autres circuits. Chaque module possède au moins un régulateur linéaire 5V et un oscillateur à quartz 16 MHz (ou un résonateur céramique dans certains modèles). Le microcontrôleur est pré programmé avec un boot loader de façon à ce qu'un programmeur dédié ne soit pas nécessaire.

### II.6.4.1 Partie matérielle

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

#### II.6.4.1.1 Le Microcontrôleur ATmega328

Un microcontrôleur ATmega328 est un circuit intégré qui rassemble sur une puce plusieurs éléments complexes dans un espace réduit au temps des pionniers de l'électronique. Aujourd'hui, en soudant un grand nombre de composants encombrants; tels que les transistors; les résistances et les condensateurs tout peut être logé dans un petit boîtier en plastique noir muni d'un certain nombre de broches dont la programmation peut être réalisée en langage C. la figure II.16 montre un microcontrôleur ATmega328, qu'on trouve sur la carte Arduino. [18]



**Figure II.16:** Microcontrôleur ATmega328 [18]

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

- **La mémoire Flash:** C'est celle qui contiendra le programme à exécuter. Cette mémoire est effaçable et réinscriptible mémoire programme de 32Ko (dont boot loader de 0.5 ko).
- **RAM:** c'est la mémoire dite "vive", elle va contenir les variables du programme. Elle est dite "volatile" car elle s'efface si on coupe l'alimentation du microcontrôleur. Sa capacité est 2ko.
- **EEPROM:** C'est le disque dur du microcontrôleur. On y enregistre des infos qui ont besoin de survivre dans le temps, même si la carte doit être arrêtée. Cette mémoire ne s'efface pas lorsque l'on éteint le microcontrôleur ou lorsqu'on le reprogramme. [19]

#### II.6.4.1.2 Les sources de l'alimentation de la carte

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

- **VIN.** La tension d'entrée positive lorsque la carte Arduino est utilisée avec une source de tension externe (à distinguer du 5V de la connexion USB ou autre source 5V régulée). On peut alimenter la carte à l'aide de cette broche, ou, si l'alimentation est fournie par le jack d'alimentation, accéder à la tension d'alimentation sur cette broche.
- **5V.** La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (pour info : les circuits électroniques numériques nécessitent une tension d'alimentation parfaitement stable dite "tension régulée" obtenue à l'aide d'un composant appelé un régulateur et qui est intégré à la carte Arduino). Le 5V régulé fourni par cette broche peut donc provenir soit de la tension d'alimentation VIN via le régulateur de la carte, ou bien de la connexion USB (qui fournit du 5V régulé) ou de tout autre source d'alimentation régulée.
- **3V3.** Une alimentation de 3.3V fournie par le circuit intégré FTDI (circuit intégré faisant l'adaptation du signal entre le port USB de votre ordinateur et le port série de l'ATmega) de la carte est disponible : ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5V. L'intensité maximale disponible sur cette broche est de 50mA. [20]

#### II.6.4.1.3 Les entrées & sorties

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

En plus, certaines broches ont des fonctions spécialisées :

- Interruptions Externes: Broches 2 et 3. Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur.-Impulsion PWM (largeur d'impulsion modulée): Broches 3, 5, 6, 9, 10, et 11. Fournissent une impulsion PWM 8-bits à l'aide de l'instruction `analog Write ()`.
- SPI(Interface Série Périphérique): Broches 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ces broches supportent la communication SPI (Interface Série Périphérique) disponible avec la librairie pour communication SPI. Les broches SPI sont également connectées sur le connecteur ICSP qui est mécaniquement compatible avec les cartes Mega.
- I2C: Broches 4 (SDA) et 5 (SCL). Supportent les communications de protocole I2C (ou interface TWI (Two Wire Interface -Interface "2 fils"), disponible en utilisant la librairie `Wire/I2C` (ou `TWI-Two-Wire interface-interface "2 fils"`).
- LED: Broche 13. Il y a une LED incluse dans la carte connectée à la broche 13. Lorsque la broche est au niveau HAUT, la LED est allumée, lorsque la broche est au niveau BAS, la LED est éteinte.

La carte UNO dispose 6 entrées analogiques (numérotées de 0 à 5), chacune pouvant fournir une mesure d'une résolution de 10 bits (càd sur 1024 niveaux soit de 0 à 1023) à l'aide de la très utile fonction `analog Read()` du langage Arduino. Par défaut, ces broches mesurent entre le 0V (valeur 0) et le 5V (valeur 1023), mais il est possible de modifier la référence supérieure de la plage de mesure en utilisant la broche AREF et l'instruction `analog Référence ()` du langage Arduino.

La carte Arduino UNO intègre un fusible qui protège le port USB de l'ordinateur contre les surcharges en intensité (le port USB est généralement limité à 500mA en intensité). Bien que la plupart des ordinateurs aient leur propre protection interne, le fusible de la carte fournit une couche supplémentaire de protection. Si plus de 500mA sont appliqués au port USB, le fusible de la carte coupera automatiquement la connexion jusqu'à ce que le court-circuit ou la surcharge soit stoppé. [21]

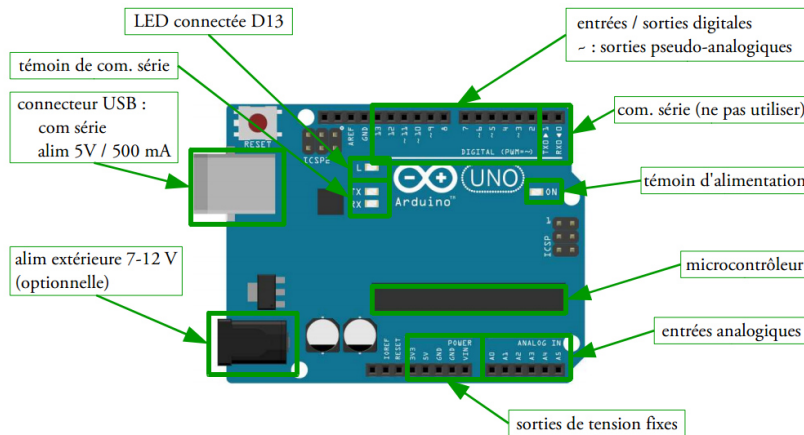
#### II.6.4.1.4 Les ports de communications

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

On utilise (RX) pour recevoir et (TX) transmettre (les données séries de niveau TTL). Ces broches sont connectées aux broches correspondantes du circuit intégré ATmega328

programmé en convertisseur USB-vers-série de la carte, composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur.

Comme un port de communication virtuel pour le logiciel sur l'ordinateur, La connexion série de l'Arduino est très pratique pour communiquer avec un PC, mais son inconvénient est le câble USB, pour éviter cela, il existe différentes méthodes pour utiliser ce dernier sans fil:



**Figure II.17 :** Constitution de la carte Arduino UNO [21]

## II.7 la porte série USB [4]

### II.7.1 Définition

Le Universal Serial Bus (USB), en français Bus universel en série, dont le sigle, inusité, BUS est une norme relative à un bus informatique en transmission série qui sert à connecter des périphériques informatiques à un ordinateur. Le bus USB permet de connecter des périphériques à chaud (quand l'ordinateur est en marche) et en bénéficiant du Plug and Play (le système reconnaît automatiquement le périphérique). Il peut alimenter les périphériques peu gourmands en énergie (disques SSD en particulier).



**Figure II.18:** un câble USB [4]

### II.7.2 caractéristiques générales

L'Universal Serial Bus est une connexion à haute vitesse qui permet de connecter des périphériques externes à un ordinateur (hôte dans la terminologie USB). Il permet le branchement simultané de 127 périphériques par contrôleur (hôte). Le bus autorise les branchements et débranchements à chaud (« Hot-Plug », sans avoir besoin de redémarrer l'ordinateur) et fournit l'alimentation électrique des périphériques sous 5 V, dans la limite de 0,5 A, soit 2,5 W.

D'un point de vue logiciel, le bus possède une topologie arborescente (dite également en étoile) : les feuilles de cet arbre sont les périphériques ; les nœuds internes sont des hubs qui permettent de greffer des sous-arborescences dans l'arborescence principale. On trouve dans le commerce ces hubs sous forme de petits boîtiers alimentés soit sur le bus, soit sur le secteur, et qui s'utilisent comme des multiprises. Certains périphériques intègrent également un hub (moniteurs, claviers...). Cependant, tout bus USB possède au moins un hub situé sur le contrôleur : le hub racine, qui peut gérer les prises USB de l'ordinateur. Le nombre de hubs connectés en cascade est limité : hub racine compris, il ne doit pas exister plus de sept couches dans l'arborescence.

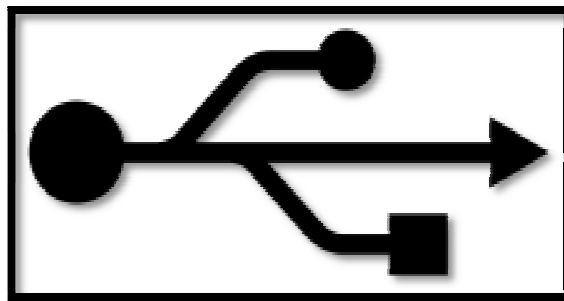


Figure II.19: symbole de l'USB [4]

### II .8 Servomoteur [22]

le Servomoteur (abrégé : “servo”). C'est un moteur un peu particulier, puisqu'il confond un ensemble de mécanique et d'électronique, mais son principe de fonctionnement reste assez simple. Les parties seront donc assez courtes dans l'ensemble car les servomoteurs contiennent dans leur “ventre” des moteurs à courant continu que vous connaissez à présent. Cela m'évitera des explications supplémentaires.



**Figure II.20** : servomoteurs [22]

### II.8.1 Composition d'un servomoteur

Les servomoteurs ont donc l'avantage d'être asservis en position angulaire. Cela signifie, je vous l'expliquais, que l'axe de sortie du servomoteur respectera une consigne d'orientation que vous lui envoyez en son entrée. En plus, tenez-vous bien, si par malheur les roues venaient à changer d'orientation en passant sur un caillou par exemple, l'électronique interne du servomoteur essaiera tant bien que mal de conserver cette position ! Et quelle que soit la force que l'on exerce sur le bras du servomoteur, il essaiera de toujours garder le même angle (dans les limites du raisonnable évidemment). En quelque sorte vous ne pilotez pas directement le moteur, mais plutôt vous imposez le résultat que vous voulez avoir en sortie.

### II.8.2 Apparence

On en trouve de toutes les tailles et de toutes les puissances. La plupart du temps la sortie peut se positionner entre 0 et 180°. Cela dit, il en existe également dont la sortie peut se débattre sur seulement 90° et d'autres, ayant un plus grand débattement, sur 360°. Ceux qui ont la possibilité de faire plusieurs tours sont souvent appelés servotreuils. Enfin, les derniers, qui peuvent faire tourner leur axe sans jamais se buter, sont appelés servomoteurs à rotation continue. Les servomoteurs sont très fréquemment employés dans les applications de modélisme pour piloter le safran d'un bateau, le gouvernail d'un avion ou bien même les roues d'une voiture téléguidée dont on a parlé jusqu'à présent. Maintenant que les présentations sont faites, mettons-le à nu ! Il est composé de plusieurs éléments visibles ... :

- Les fils, qui sont au nombre de trois (nous y reviendrons)
- L'axe de rotation sur lequel est monté un accessoire en plastique ou en métal
- Le boîtier qui le protège

... mais aussi de plusieurs éléments que l'on ne voit pas :

- Un moteur à courant continu
- Des engrenages pour former un réducteur (en plastique ou en métal)
- Un capteur de position de l'angle d'orientation de l'axe (un potentiomètre bien souvent)
- Une carte électronique pour le contrôle de la position de l'axe et le pilotage du moteur à courant continu

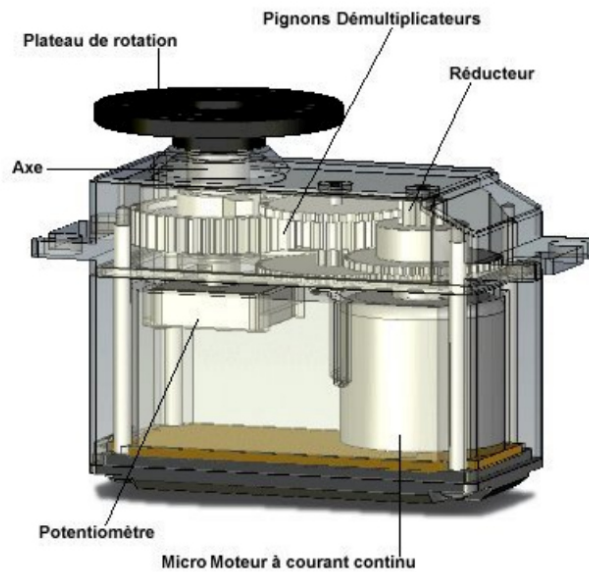


Figure II.21 : image 3D de vue de l'extérieur et de l'intérieur d'un servomoteur [22]

### II.8.3 Connectique

Le servomoteur a besoin de trois fils de connexion pour fonctionner. Deux fils servent à son alimentation, le dernier étant celui qui reçoit le signal de commande :

- **rouge** : pour l'alimentation positive (4.5V à 6V en général)
- **noir** ou **marron** : pour la masse (0V)
- **orange, jaune, blanc, ...** : entrée du signal de commande

Nous verrons tout à l'heure ce que nous devons entrer sur le dernier fil. utilisation avec l'Arduino.



**Figure II.22 :** Fils de connexion d'un servomoteur [22]

### **II.9 Conclusion**

Après avoir vu en détail la partie 'matériel' (ou hardware) de la commande numérique en décrivant chaque élément constituant (servomoteur, moteur pas à pas, carte arduino, etc.), on peut s'attaquer au chapitre suivant contenant les programmes ainsi que les logiciels utilisés dans la commande numérique, c'est ce qu'on appelle la partie software en général.



### III .1 Introduction

Les machines-outils à commande numérique sont complètement pilotées par ordinateur. Pour que la machine comprenne les instructions données par l’ordinateur de contrôle, il faut que ces instructions soient transmises dans un langage de programmation spécial, qui est le G-code, ou code-G. Les outils informatiques actuels de la chaîne soft CAO permettent de se passer de la connaissance pratique du langage, mais pour ceux qui voudraient éventuellement faire des programmes simples directement avec un éditeur pour les envoyer dans le contrôleur ARDUINO, il est important de connaître quelques bases de ce langage. Tell que l’Inkscape, Processing et Arduino.

### III .2 schéma synoptique

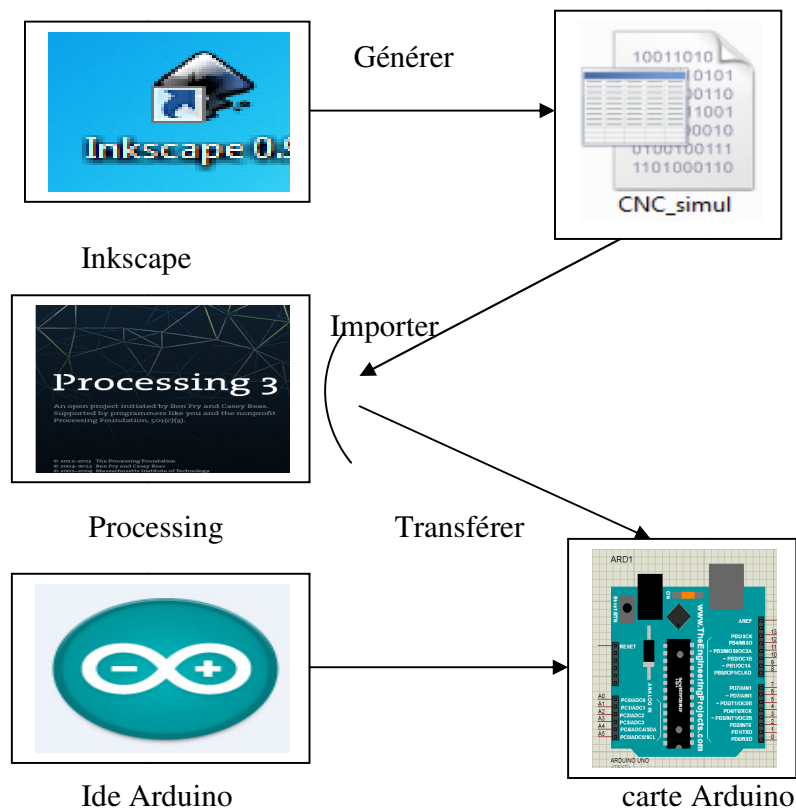


Figure III.1 schéma synoptique de la partie logiciel

### III .3 Logiciel « Arduino »

#### III .3.1 Structure générale du programme (IDE Arduino)

Comme n'importe quel langage de programmation, une interface souple et simple est exécutable sur n'importe quel système d'exploitation Arduino basé sur la programmation en C.

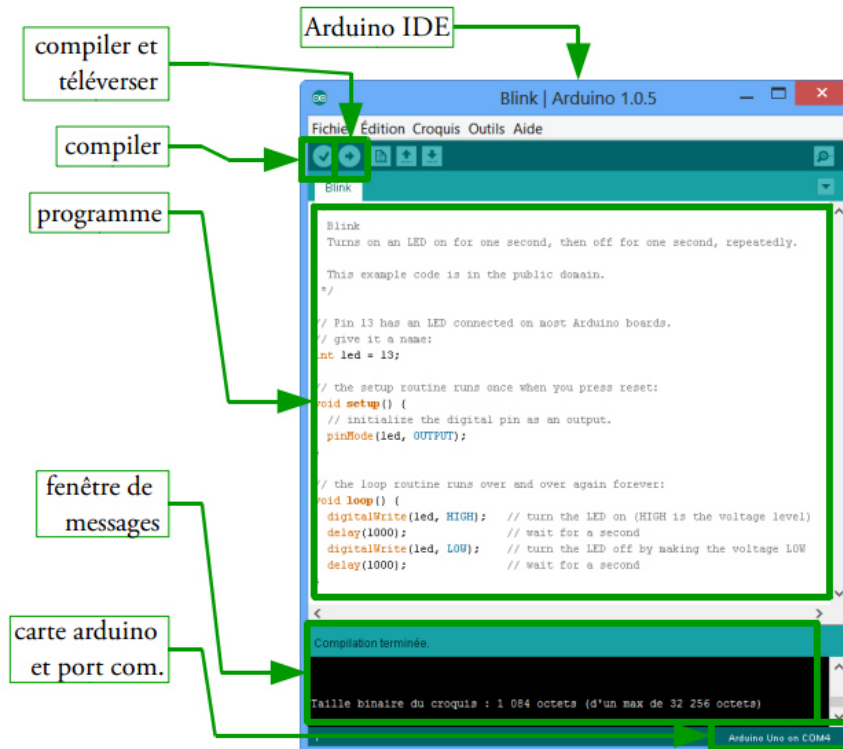


Figure III.2:interface IDE ARDUINO [4]

#### III.3.2 description de logiciel 'ARDUINO' [4]

Un programme ARDUINO est une suite d'instructions élémentaires sous forme textuelle (ligne par ligne). La carte lit puis effectue les instructions les unes après les autres dans l'ordre défini par les lignes de codes.

##### III.3.2.1 Commentaires

Les commentaires sont, en programmation informatique, des portions du code source ignorées par le compilateur ou l'interpréteur, car ils ne sont pas censés influencer l'exécution du programme.

### III.3.2.2 Définition des variables

Pour notre montage, on va utiliser une sortie numérique de la carte qui est par exemple la 3 émet sortie numérique ; cette variable doit être définie et nommée ici moteur pin 3 ; la syntaxe est pour désigner un nombre entier est **Int**.

### III.3.2.3 Configuration des entres et des sorties void setup ()

Les broches numériques de l'arduino peuvent aussi bien être configurées en entrées numériques ou en sorties numériques; ici on va configurer moteur pin en sortie ; pin mode (nom, état) est une des quatre fonctions relatives aux entrées – sorties numériques.

```
1 //fonction d'initialisation de la carte
2 void setup()
3 {
4     //contenu de l'initialisation
5     //on écrit le code à l'intérieur
6 }
```

### III .3.2.4 Programmation des interactions void loop()

Dans cette boucle, on définit les opérations à effectuer dans l'ordre **digital write**(nom, état) est une autre des quatre fonctions relatives aux entrées – sorties numériques.

- delay(temps en millisecondes) est la commande d'attente entre deux instructions.
- Chaque ligne d'instruction est terminée par un point-virgule.
- Ne pas oublier les accolades qu'encadre la boucle.

```
1 //fonction principale, elle se répète (s'exécute) à l'infini
2 void loop()
3 {
4     //contenu de votre programme
5 }
```

### III .3.2.5 Organigramme d'un programme arduino pour la commande des moteurs

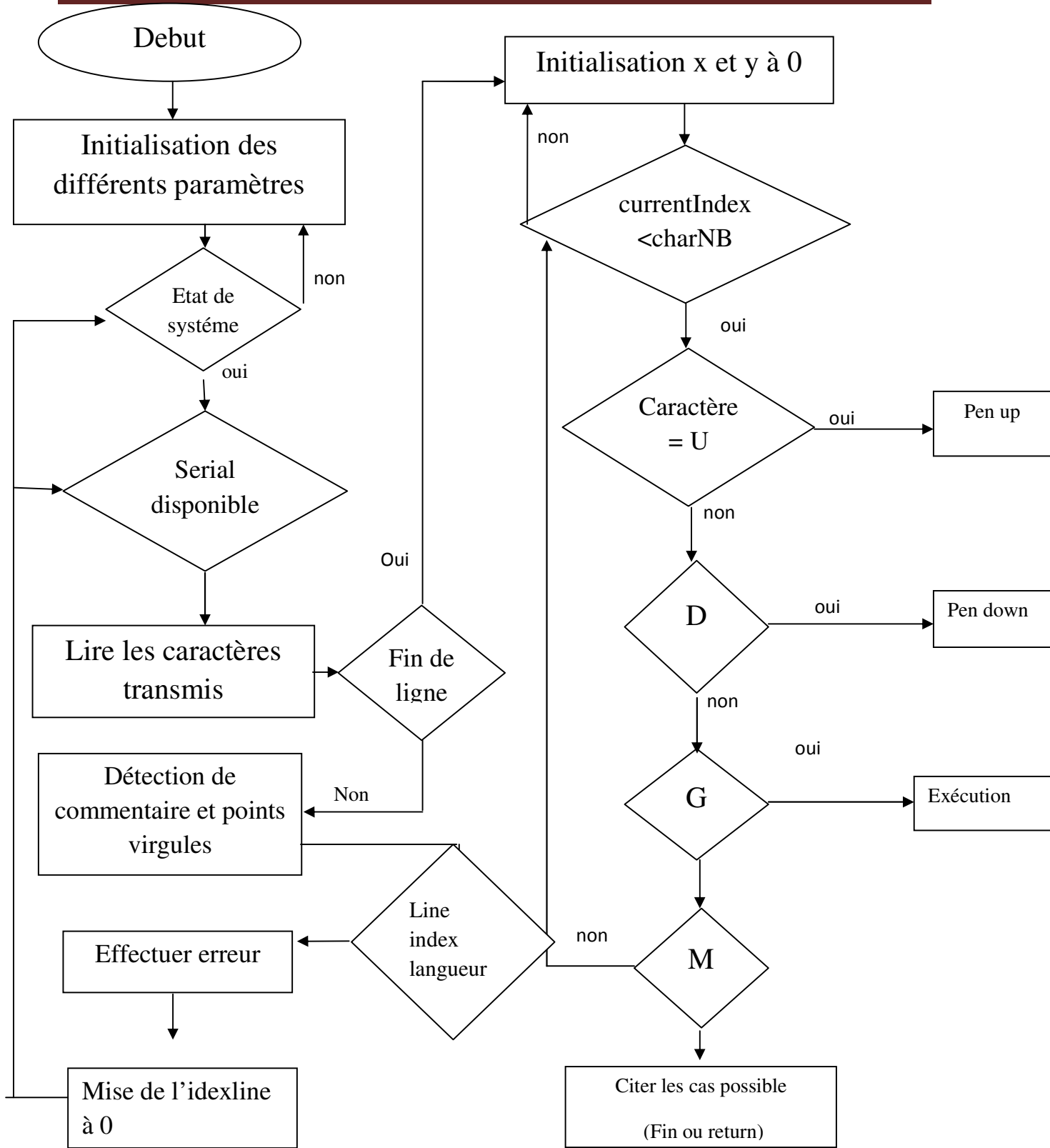


Figure III.3 : Organigramme d'un programme arduino pour la commande des moteurs

### III.4 Les étapes de téléchargement du programme [22]

Une simple manipulation enchaînée doit être suivie afin d’injecter un code vers la carte Arduino via le port USB.

Avant d’envoyer le programme vers la carte, il faut dire au logiciel quel est le nom de la carte et sur quel port elle est branchée. Choisir la carte que l’on va programmer. Ce n’est pas très compliqué, le nom de votre carte est indiqué sur elle. Pour nous, il s’agit de la carte “Uno”. Allez dans le menu “Tools” (“outils” en français) puis dans “Board” (“carte” en français). Vérifiez que c’est bien le nom “Arduin Uno” qui est coché.

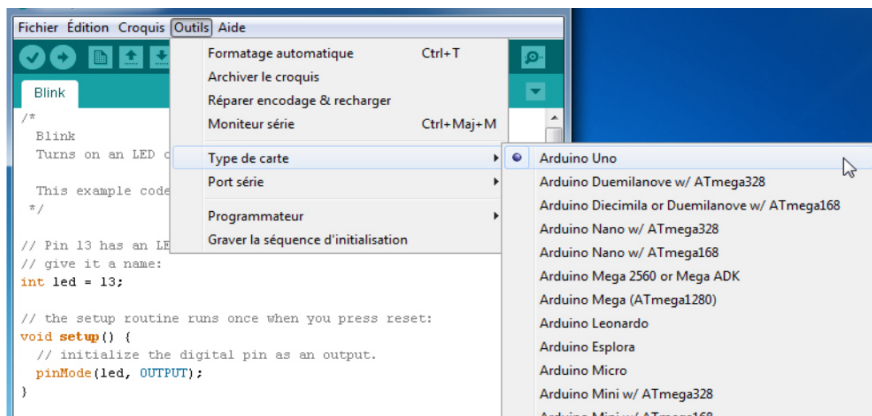


Figure III.4:sélection type arduino.

Choisissez le port de connexion de la carte. Allez dans le menu Tools, puis Serial port. Là, vous choisissez le port COMX, X étant le numéro du port qui est affiché. Ne choisissez pas COM1 car il n’est quasiment jamais connecté à la carte. Dans mon cas,il s’agit d’COM4 :

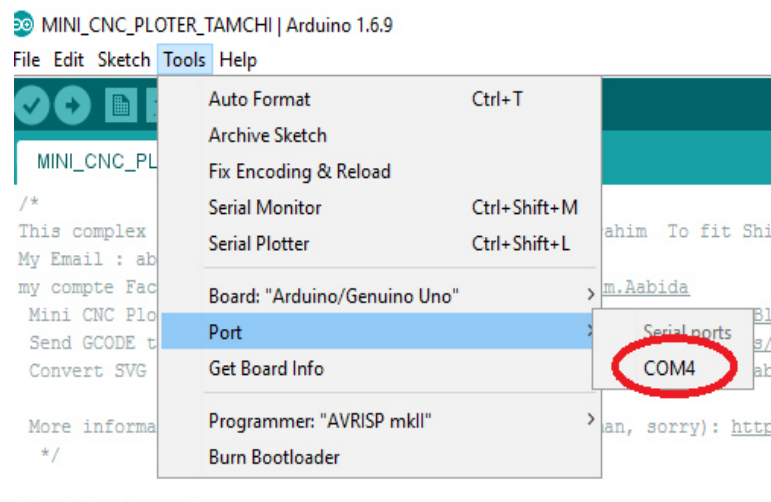


Figure III.5:sélection port (COM) arduino.

Maintenant, il va falloir envoyer le programme dans la carte. Pour ce faire, il suffit de cliquer sur le bouton Téléverser, en jaune-orangé sur la photo :

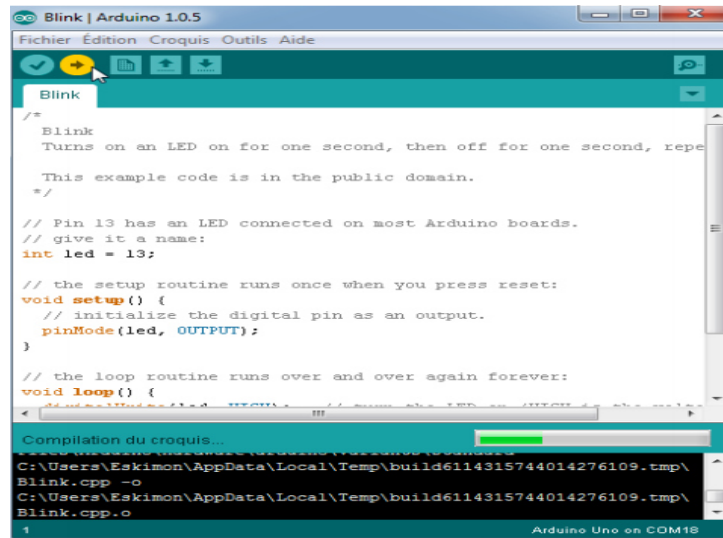


Figure III.6: téléverser code arduino.

En bas dans l’image, vous voyez le texte : “Téléversement...”, 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 :

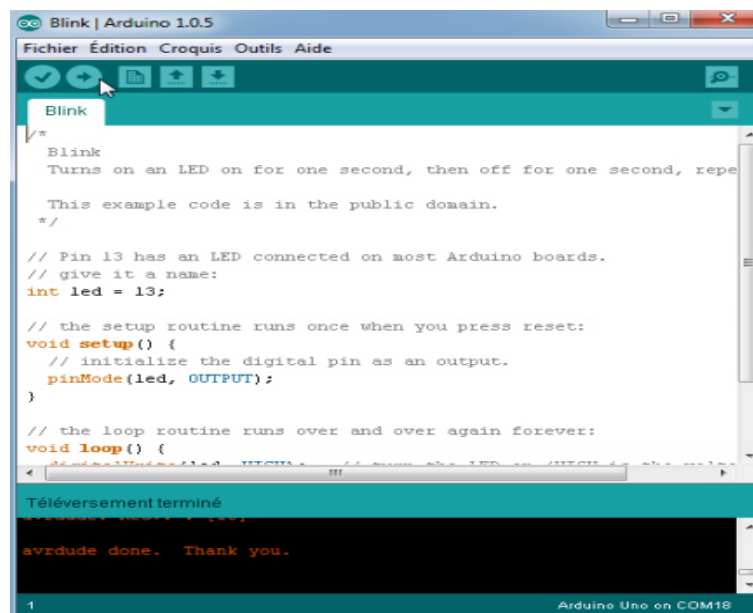


Figure III.7: téléversement terminé.

Le message affiché : “Téléversement terminé” signale que le programme à bien été chargé dans la carte. Si votre matériel fonctionne, vous devriez avoir une LED sur la carte qui clignote :



LED sur la carte qui clignote

Une fois le programme Arduino est implémenté dans la carte c.-à-d. que la carte est prête à recevoir des commandes sous forme d'un fichier data pour contrôler nos moteurs (axes X et Y), ce dernier (fichier data) doit se numériser, il y'en a plusieurs logiciels de numérisation mais on a choisit qu'un seul, en occurrence Inkscape.

### III.5 Inkscape

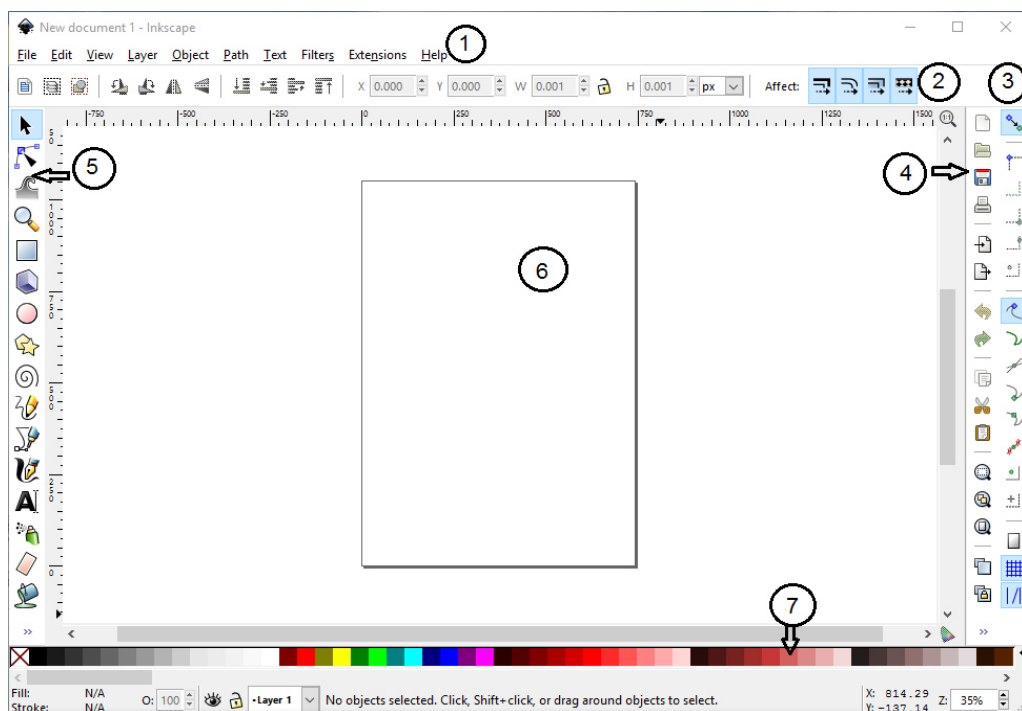
Inkscape est un puissant logiciel de dessin vectoriel à vocation "artistique", le dessin y est enregistré sous forme d'équations de courbes (chemins) et non par des pixels comme les images bitmap. Ce programme est vaste (mais très bien fait), les fonctions qui nous intéressent dans le cadre de ce tutoriel c'est qu'il permet d'ouvrir un nombre très important de type d'image (PS, DXF, PDF, SVG, PNG, JPG, BMP, XCF,...) et qu'il est capable de transformer des images bitmap en image vectorielle. [23]

#### III.5.1 Fonctionnalités d'inkscape

- Dessin : outil pinceau (dessin à main levée avec des chemins simples), outil crayon (création de courbes de Bézier et de segments), outil calligraphie (dessin à main levée utilisant des chemins remplis représentant des pleins et des déliés)
- Outils de forme : rectangles (dont les coins peuvent être arrondis), ellipses (comprend les cercles, arcs, cordes et rayons), étoiles/polygones (qui peuvent être arrondis et/ou avoir des paramètres aléatoires) et spirales
- Outil texte (texte multi-lignes, édition sur le canevas)
- Images matricielles incorporées (avec une commande pour créer et incorporer les bitmaps des objets sélectionnés)
- Clones (copies d'objets liées « en direct » à l'original), comprenant un outil pour créer des motifs et ordonner des clones.

### III .5.2 L’espace de travail

L’interface d’Inkscape est composée de différentes parties qu’il est assez facile de repérer. Contrairement à celle de Gimp, l’interface d’Inkscape est unifiée autour de l’espace de dessin, et rares sont les fenêtres qui sont flottantes par défaut. Ceci permet à l’utilisateur de se repérer facilement mais a aussi un impact particulier : tout nouveau document possède ses propres éléments d’interface. Bien entendu, la barre de menus et la barre de commandes sont également présentes, comme dans la majorité des applications. [24]



**Figure III.8:** La création d’un nouveau document entraîne l’affichage d’une nouvelle

Fenêtre possédant ses propres éléments d’interface.

- 1- les menus, positionnés tout en haut
- 2- la barre de contrôle, juste sous la barre de commande ;
- 3- la barre de contrôle du magnétisme, à droite
- 4- la barre de commande, juste en dessous du menu
- 5- la boîte à outils, à gauche
- 6- le canevas, au centre
- 7- la palette de couleurs



### III.5.3 Vectorisation d'une image matricielle ou un texte par inkscape

#### ➤ Lancer inkscape

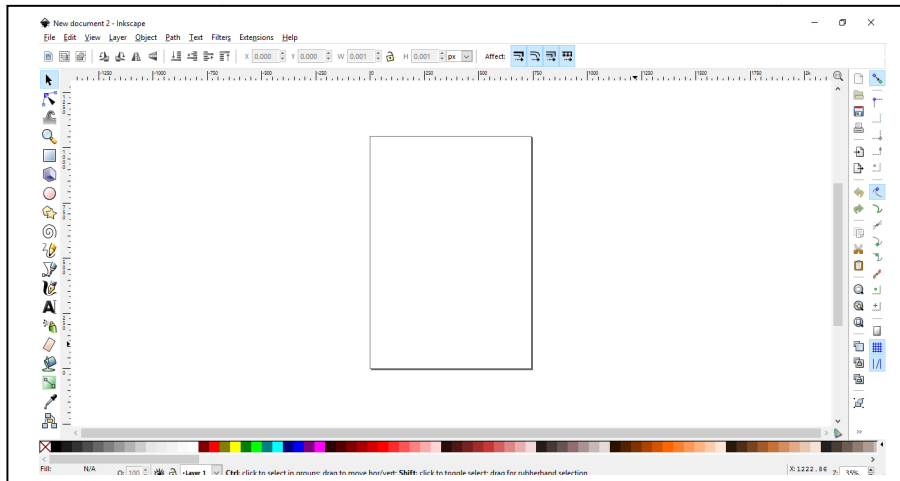


Figure III.9 : l'interface principale d'Inkscape

#### ➤ Personnaliser un modèle

Par défaut, Inkscape se lance avec un cadre au format A4. Ce cadre sert de repère, de visualisation du fichier à partir d'un navigateur et d'exportation au format de la page.

- File => Document propriétés ou [Maj] + [Ctrl] + [D].
- Dans la partie (page) du document, choisir le format souhaité.

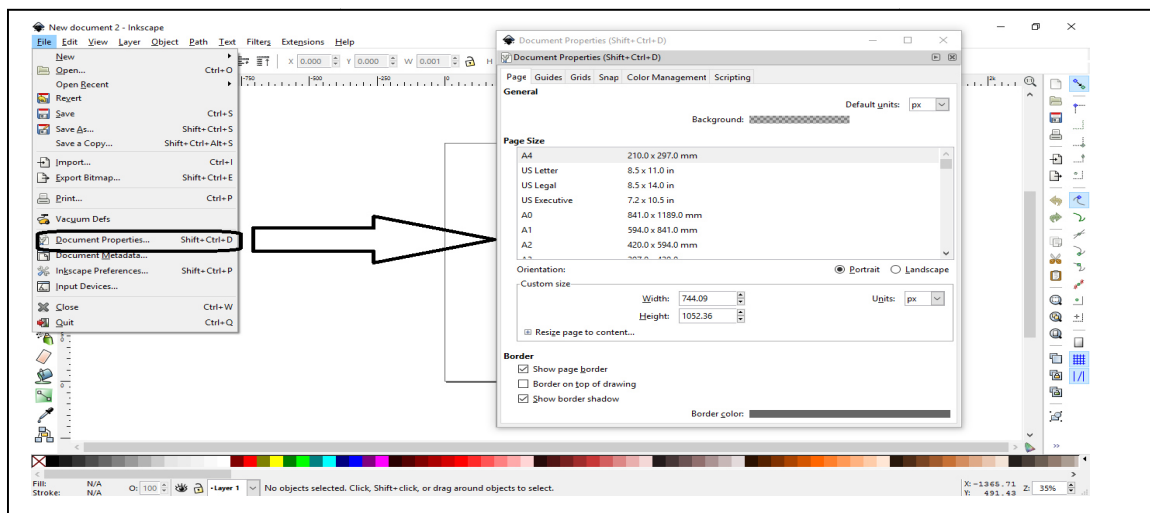
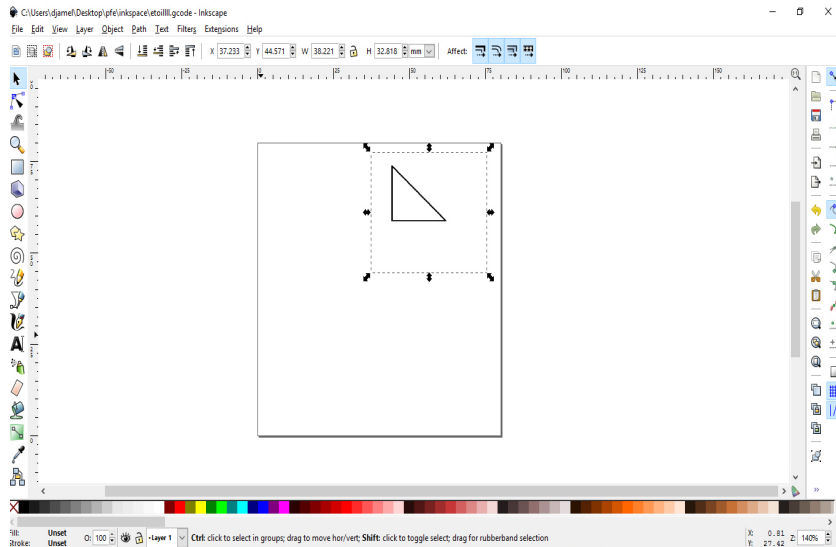


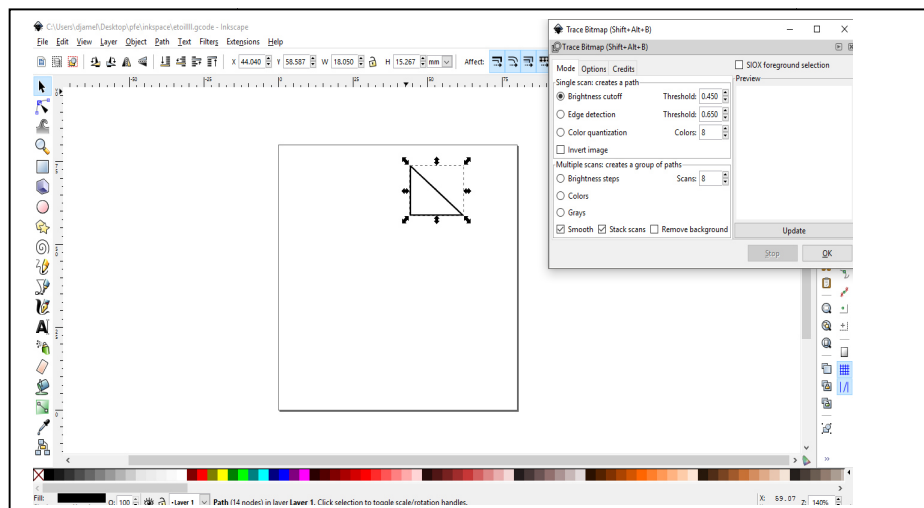
Figure III.10 : Le paramétrage général du document offre plus de possibilités s'il est effectué sur le document

➤ charger l’image avec inkscape

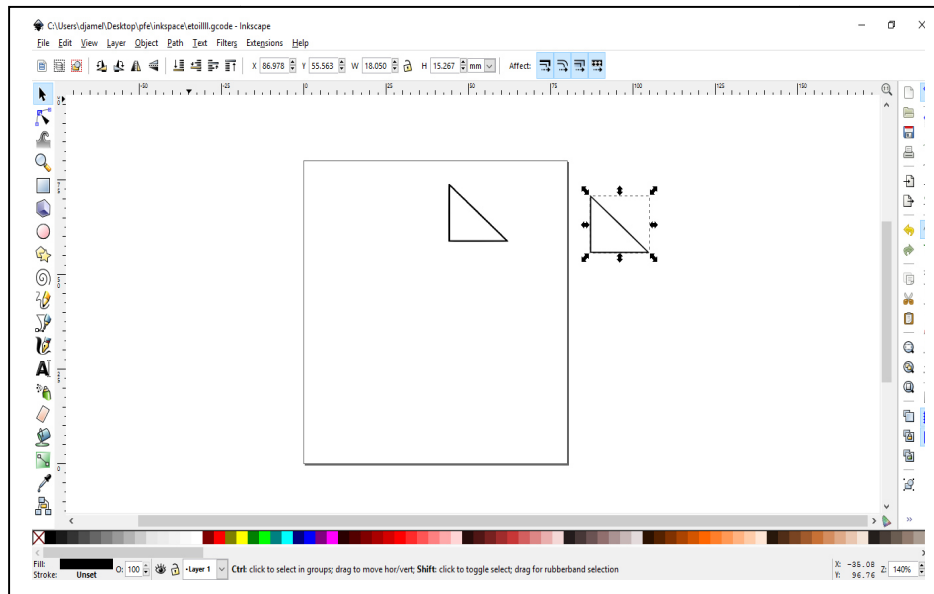
- Fichier => ouvrir .... Et sélectionner votre image



- Aller dans le menu (path => trace bitmap....)’une boite de dialogue va s’ouvrir et cliquer sur ubdat => ok



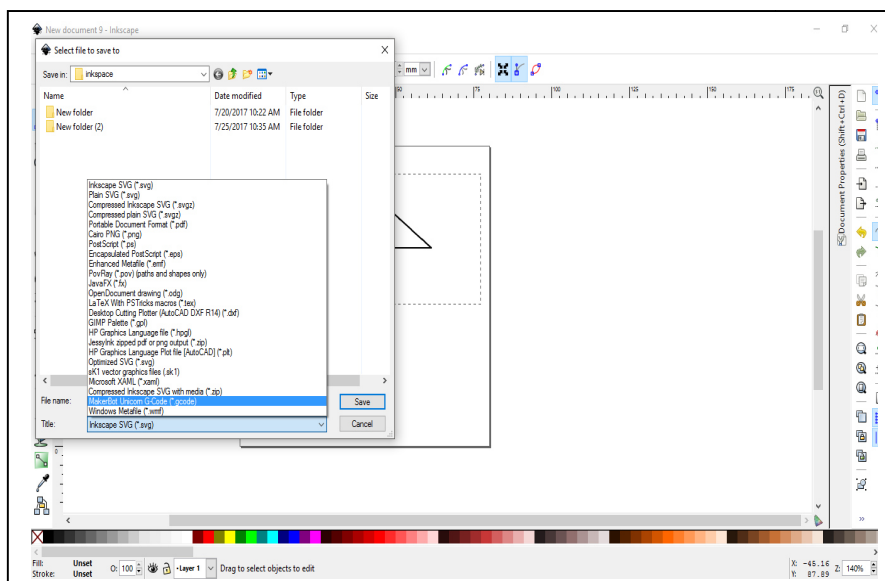
C'est fait vous pouvez fermer la boîte de dialogue, les 2 images (l'originale et l'image vectorielle) sont superposés, un simple «glissez déposé» les sépareront .vous pouvez zoom en gros pour apprécier la différence entre une image vectorielle et une image matricielle (bitmap)



### III.5.4 Génération du Gcode par inkscape

Avant de générer le fichier Gcode il faut intégrer une bibliothèque dans l'inkscape qui nous permet de sauvegarder de type d'extension Gcode, Puis Sélectionner l'image vectorielle précédemment généré, finalement allez dans le => (file) => (Save as) file name ( ) et title select (\*.gcode)

Avec l'emplacement de sauvegarde, ensuite cliquer ok sur la fenêtre de dialogue



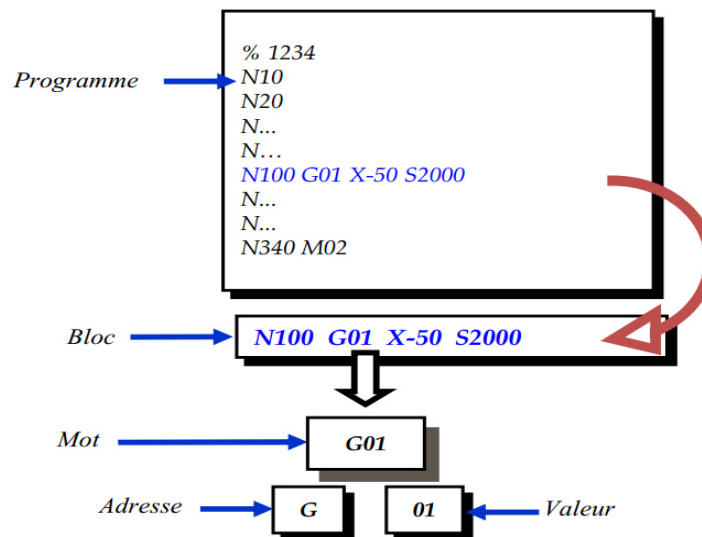
Une fois le fichier compilé est enregistré sous forme d'une extension « gcode ». Cette extension permet de convertir des chemins obtenus dans Inkscape en langage Gcode, ce dernier fichier permet de piloter des machines à commandes numériques ou CNC en anglais.

### III.6 Langue de contrôleurs numériques « G-code »

#### III.6.1 description

A l'origine, le code G est basé sur un principe de programmation qui remonte à la période des cartes perforées, au début des années 60. Il a tout d'abord été développé par l'EIA (Electronic Industries Alliances) et a été normalisé sous la référence RS274D ou ISO 6983 en février 1980.

Cette programmation s'appuie sur des fonctions préparatoires de type G. Elle est complétée par des fonctions auxiliaires (de type M) et technologiques (F, S, etc....). Par l'intermédiaire de ce code, l'utilisateur communique à la machine un ensemble d'instructions explicites. Un programme en G-code est donc la traduction d'une suite d'actions et de déplacements élémentaires (ligne droite ou arc de cercle par exemple) qui permettent de générer des conditions d'usinage ainsi qu'une trajectoire plus ou moins approximée, ce qu'il en est aujourd'hui du respect de la norme ISO dans la programmation des MOCN [25]

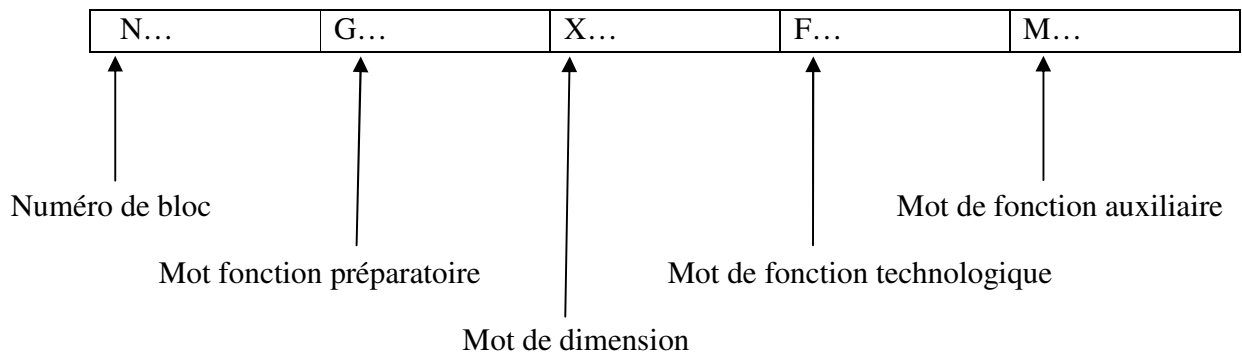


**Figure III.11 :** Structure et contenu d'un programme CN (Gcode). [25]

### III.6.1.1 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. [26]

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).

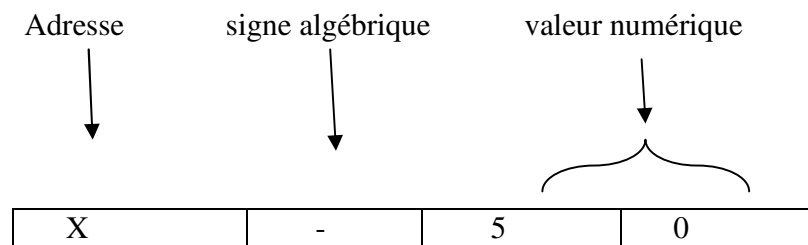


### III.6.1.2 Format d'un mot

Le mot définit une instruction ou donnée à transmettre au système de commande. Dans l'exemple ci-dessous le bloc N100 est composé de 4 mots.

**N100 G01 X-50 S2000**

Un mot est composé :



Les adresses peuvent être:

- Positions
- Fonctions préparatoires
- Fonctions auxiliaires
- Appel outil

**Les signes :** Le signe "+" est implicite s'il n'est pas écrit. Il faut obligatoirement écrire le signe "-" "quand cela est nécessaire. [26]

### III .6.2 Les fonctions

- Fonctions préparatoires (G), fonctions d'appel de mode d'interpolation (G 0), cycles machine :
  - Coordonnées de points (X, Y, Z, I, J, K)
  - Vitesses, avances... (S, F)Fonctions auxiliaires (M) qui permettent d'enclencher la lubrification, de changer d'outil, ou de déclencher des accessoires.

Le fichier Gcode qui est généré par l'inkscape (voir étape précédente) est maintenant prêt à être envoyé à la carte de traitement (dans notre cas arduino ) mais malheureusement pour nous la carte arduino ne peut pas lire les fichiers de type Gcode, et donc on est obligé de passer par d'autres logiciels qui nous permettent d'exécuter sur arduino le programme dédié à la commande des moteurs, parmi ce logiciels on trouve Processing .[26]

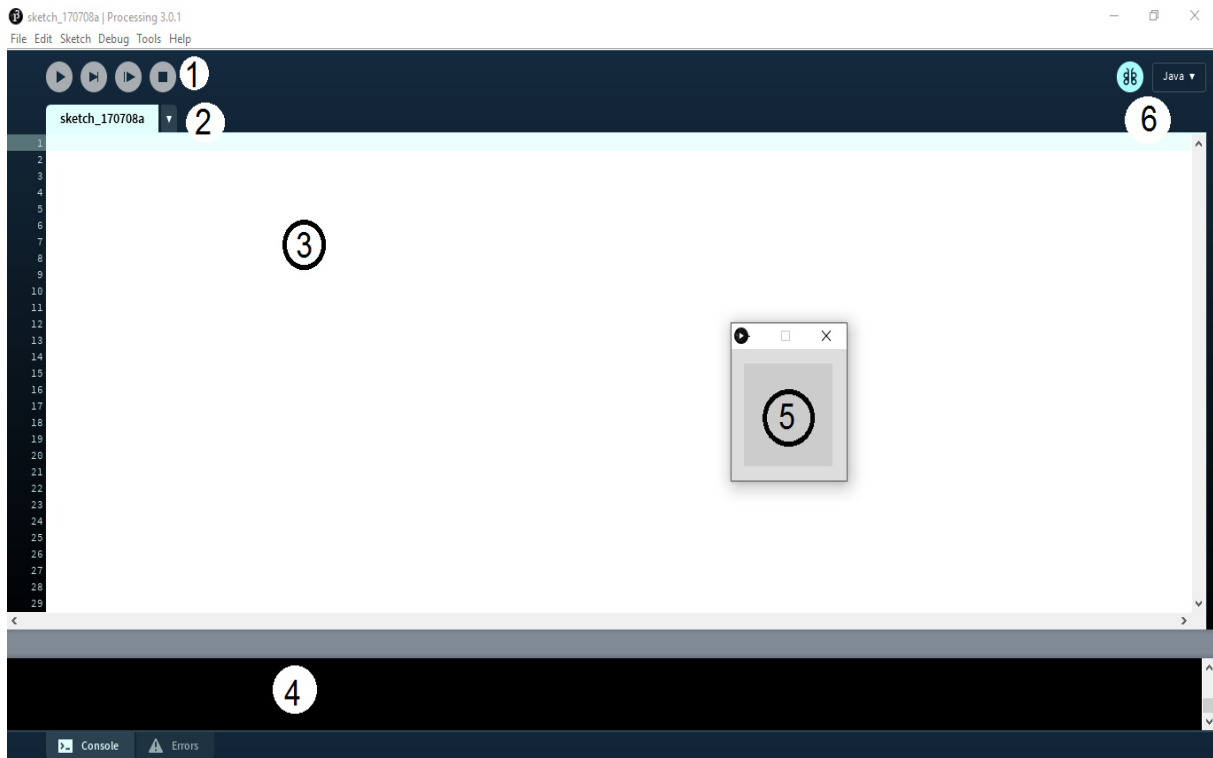
### III.7 Logiciel de pilotage de la machine CNC Processing

Processing est un langage de programmation dédié à la production artistique, et notamment à la production d'images, ce qui explique sa large diffusion dans les écoles d'art et de design graphique et interactif. Grâce à ses nombreux modules additionnels, Processing ne se limite pas à la création visuelle et peut communiquer avec des dispositifs électroniques de type Arduino, avec des services Internet, peut manipuler du son, de la vidéo, etc. C'est un langage à la fois simple, puissant et bien conçu, appartenant à la famille de Java et de C++, qui constitue pour ces raisons une excellente initiation à la programmation informatique. Le logiciel Processing, qui sert à rédiger et à exécuter des programmes dans le langage du même nom est par ailleurs gratuit et disponible sur trois plates-formes : Mac OS, Windows et Linux. [26]

### III.7.1 L'interface

L'interface d'utilisation de Processing est composée de deux fenêtres distinctes :

La fenêtre principale dans laquelle vous allez créer votre projet et la fenêtre de visualisation dans laquelle vos réalisations apparaissent. On trouve plus précisément les éléments suivants dans l'interface: [26]

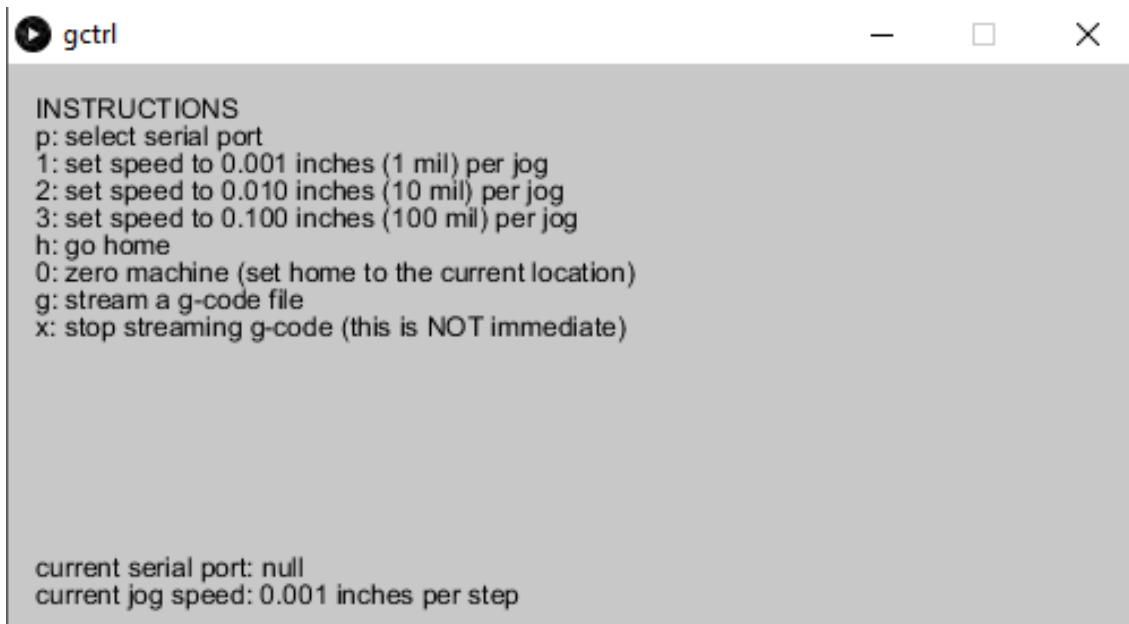


**Figure III.12 :** l'interface de processing

1. Barre d'actions
2. Barre d'onglets
3. Zone d'édition (pour y saisir votre programme)
4. Console (destinée aux tests et messages d'erreur)
5. Fenêtre de visualisation (espace de dessin)
6. Barre de menu (non représentée sur la copie d'écran)

### III.7.2 L’interface de commande

Voilà un exemple de ce que nous avons réalisé avec Processing pour faire communiquer notre ordinateur avec la carte Arduino



**Figure III.13:** l’interface de commande

Cette interface nous permet de contrôler notre système à l’aide des étapes suivantes :

- P : la sélection du sérial porte (assurer al communication entre le pc et la carte arduino)
- 1 ,2 et 3 : la configuration de vitesse des moteur
- G : exporter le fichier G-code
- X : arrêtez le G-code ruisselant
- H : retour a l’état initial
- 0 : initialisation



## III.7.3 L'organigramme pour l'interface de commande

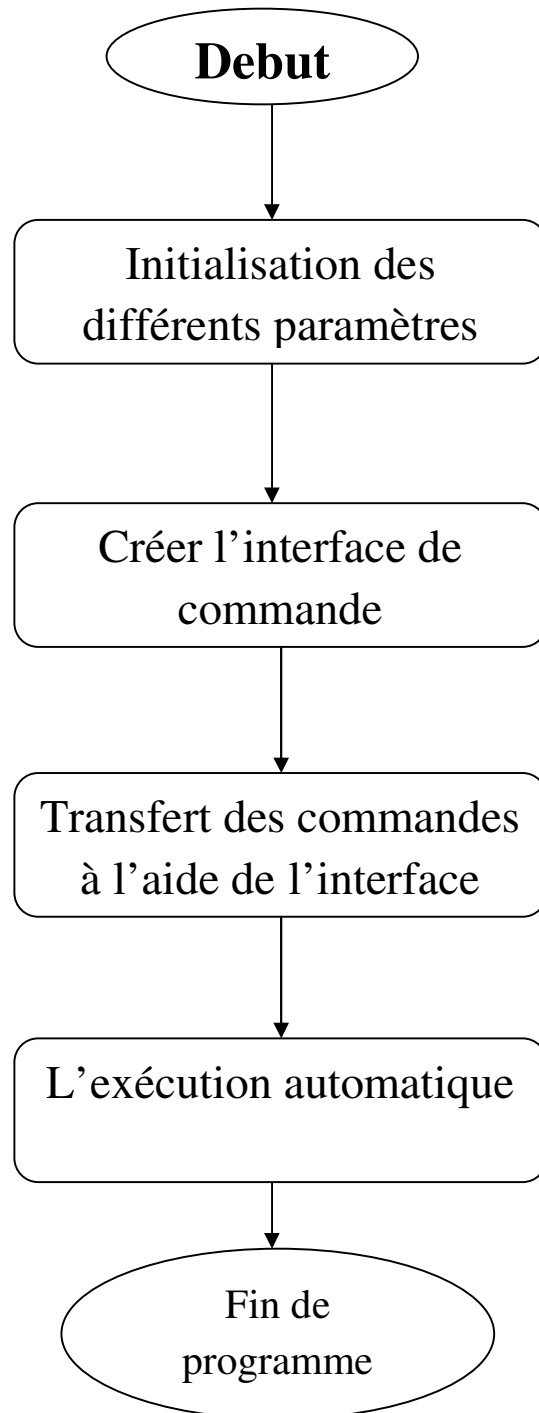


Figure III.14: L'organigramme pour l'interface de commande

### **III.8 Conclusion**

Dans ce chapitre nous avons traité la partie logiciels et programmation. Une description détaillée du logiciel Inkscape dont le rôle principale dans notre projet est de traduire l’image des motifs en images vectoriels afin de générer le G-code nécessaire pour le pilotage et la commande des moteurs pas à pas, exécuté via le logiciel Processing ; ce dernier donne la possibilité de communiquer avec des dispositifs électroniques type Arduino.

Enfin, nous avons décrit l’organigramme d’exécution de la chaîne logicielle nécessaire pour le pilotage de notre machine à réaliser à base de deux moteurs pas à pas.

### IV.1 Introduction

Ce chapitre est consacré à la fabrication et teste de l'application d'une machine CNC. Pour cette raison, plusieurs étapes ont été envisagées :

- La Simulation sur logiciel Isis Proteus : pour assurer la faisabilité et le bon fonctionnement des moteurs piloté par des circuits implémenté sur un shield L293D, ce dernier va se poser directement sur une carte arduino uno. afin de s'assurer de son fonctionnement.
- Une fois la simulation est bien vérifiée, on passe à l'étape de réalisation du système, on va essayé d'employer des matériaux que l'on peut se procurer facilement et des composants les moins chers possibles pour la fabrication de la machine CNC. Pour piloter les moteurs pas à pas des axes X et Y, nous avons utilisé la carte Arduino Uno.
- Un fois la réalisation est terminer, on passe à l'étape du teste de la machine CNC.

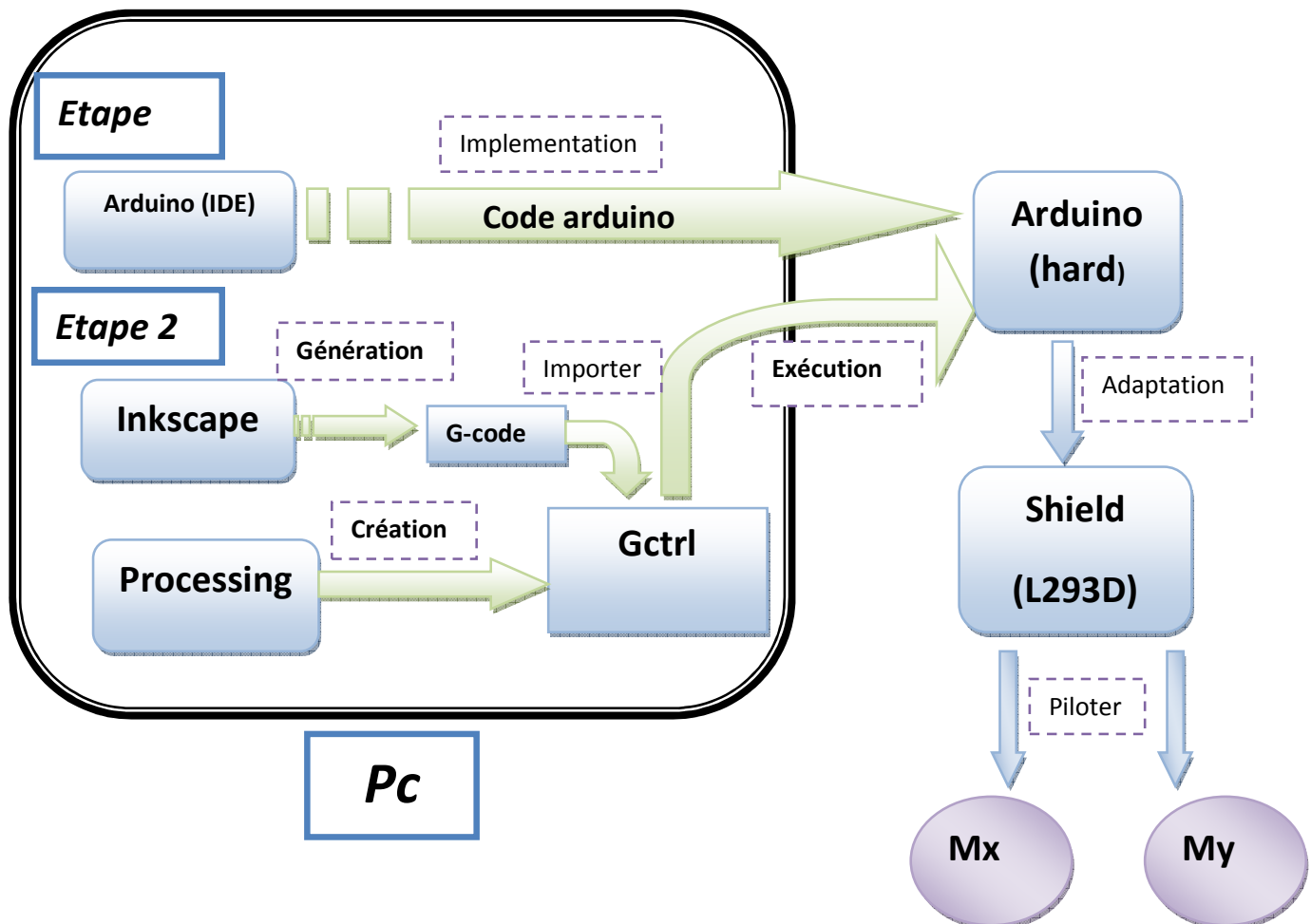
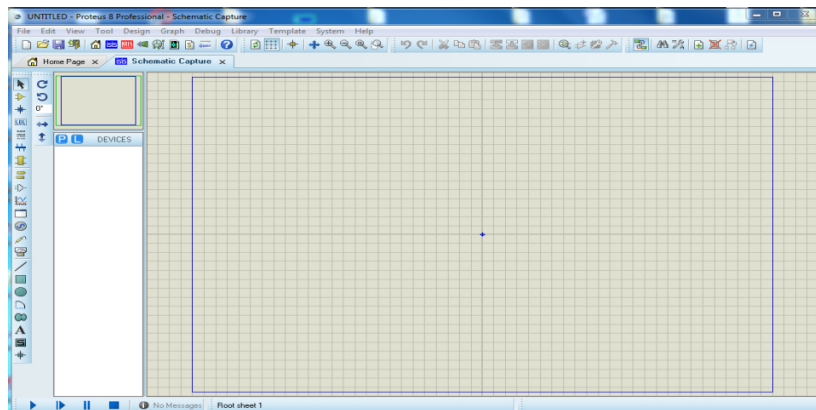


Figure IV.1 : Schéma synoptique de système

## IV.2 Simulation sous Porteuse Professionnel

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

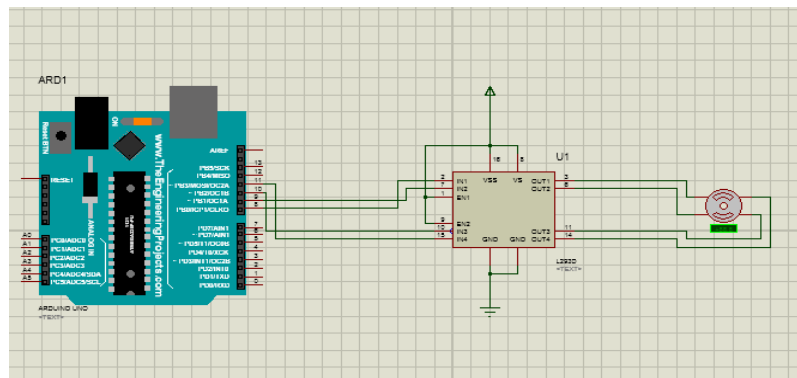
- Pack contenant des logiciels facile et rapide à comprendre et utiliser
- Le support technique est performant
- L'outil de création de prototype virtuel permet de réduire les coûts matériel et logiciel lors de la conception d'un projet.



**Figure IV.2:** Logiciel ISIS.

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

### IV.2.1. Schéma de montage réalisé

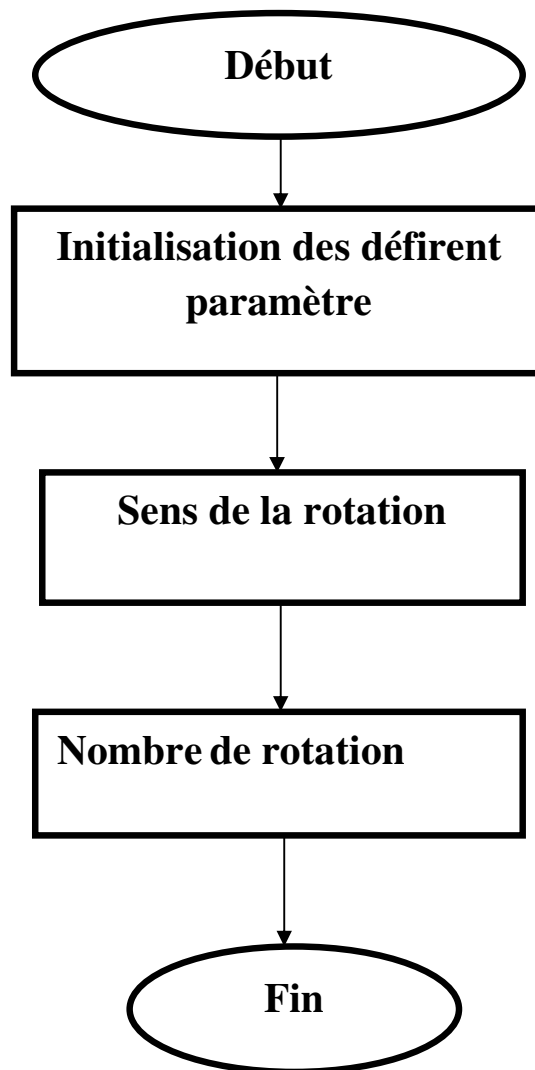


**Figure IV.3:** schéma de montage réalisé dans ISISPROTEUS.

Une fois on a terminé l'assemblage de circuit sur proteus on passe a l'étape qui nous permet de commander le moteur.

Alors on va écrire un code arduino permet la commande d'un moteur pas a pas a partir d'un circuit intégré (L293D) a deux pont en H, les 4 (1--4) pins d'entrer du moteur doit connecter avec les sortie analogique de la carte arduino.

Ce programme permet de le sens de rotation du moteur, la vitesse et le nombre de rotation par minute, il sera exécuter comme il indiquer l'organigramme suivant :

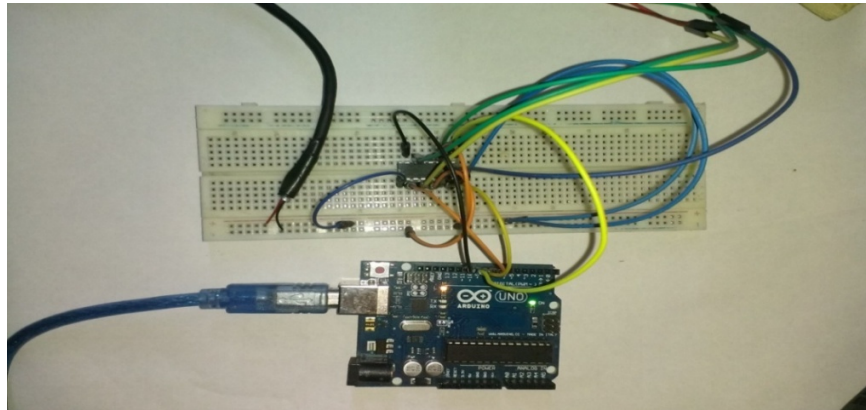


**Figure. IV.4 :** organigramme d'un code arduino pour un moteur pas a pas

Si la simulation est bien fait, on passe maintenant a la réalisation du ce circuit sur une plaque de teste pour tester et caractériser les moteurs à utiliser.

**IV.2.2 Réalisation et teste**

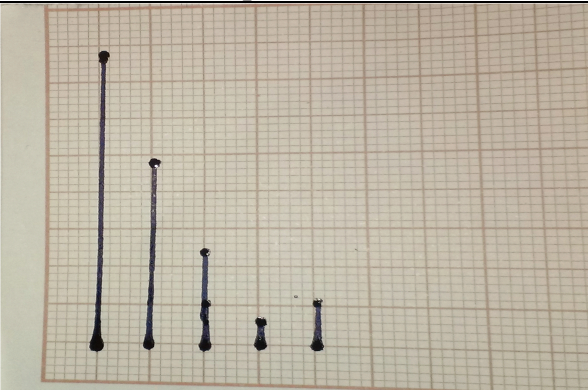
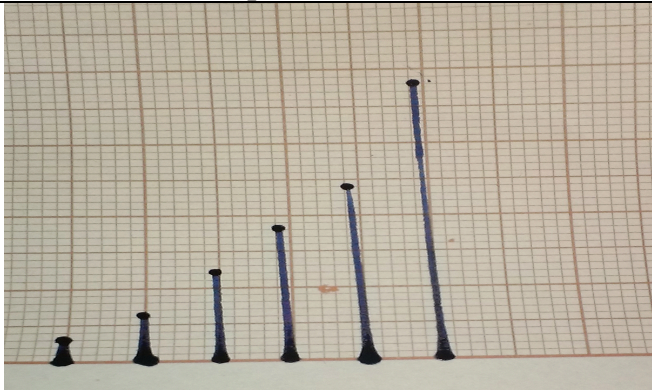
Avant de faire les testes nécessaire, On effectue l’assemblage présenté sur la figure ci-dessous:



**Figure. IV.5:** circuit de teste.

**IV.2.3 Résultats du teste et discussion**

On résume tous les résultats obtenus par les deux moteurs sur le tableau suivant sachant que la vitesse est fixée à 300 rpm :

Résultats obtenus par l’axe X	Résultats obtenus par l’axe Y
	
Maximum de déplacement : 38 mm	Maximum de déplacement : 38 mm
Nombre de rotation par le max de déplacement : 64	Nombre de rotation par le max de déplacement : 64
Le parcoure par révolution : 0,59mm	Le parcoure par révolution : 0,59mm

**Tableau. IV.1:** tableau résume les résultats du test des moteurs.

Finalement on conclue de telles sortes que les deux moteurs sont identiques, donc ils sont fiables pour les utiliser comme des axes X et Y.

### IV. 3 matériels nécessaires pour réaliser la machine CNC

Nous essayons d'employer des matériaux que l'on peut se procurer facilement en magasin de bricolage et des composants les moins chers possibles pour la fabrication de la machine CNC. Pour ce projet, nous avons besoin de Liste des pièces suivante :

- 1x Arduino Uno
- 1x circuit de command shield L293D
- 1x servons moteur
- 2x lecteurs de DVD
- Câble USB
- Des rondelles
- 10x vis et 10x écrous
- Tournevis
- Fer à souder et Souder
- 10x fils
- 1x plaque de bois pour la base du Mini Traceur (axe X)

#### IV.3.1 démonté le lecteur de DVD

Nous avons démonté le lecteur de DVD pour ne conserver que le mécanisme de déplacement de la tête de lecture. Les systèmes obtenu est système de déplacement linéaire récupéré sur un lecteur de DVD qui présente un porteur d'un axe de CNC machine.



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



### IV.3.2 Préparer les moteurs pas à pas des axes X, Y et Z

Dans cette étape, nous allons réaliser un support de bois pour fixer les systèmes de déplacement linéaire récupéré à partir des lecteurs DVD. Pour réduire le coût du projet, nous avons opté pour réaliser la structure en bois de 2cm d'épaisseur disponible dans n'importe quel magasin de bricolage. La fabrication est très simple. Il n'y a que 7 plaques de bois à découper. Vous devrez adapter le perçage pour la fixation des axes en fonction de vos lecteurs de DVD.



Figure. IV.7: le support de machine

### IV.3.3. Les axes X et Y

Dans la première image ci-dessous, nous avons les axes X et Y de notre machine CNC. Attachez-le sur notre surface, dans cette partie, nous avons besoin de quelques vis et écrous.

Pour fixer les axes X directement sur la base de CNC et Y verticalement dans ce (90 degrés).

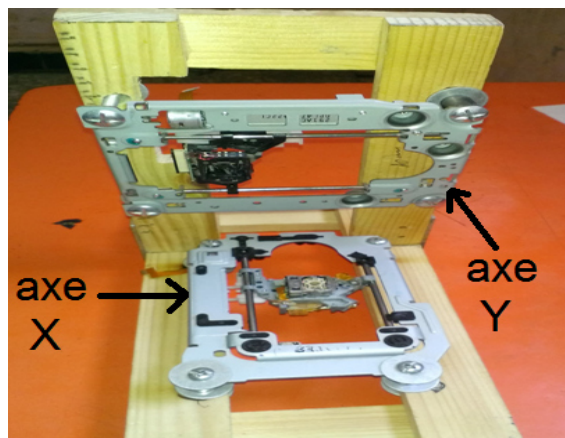


Figure IV.8 : Fixation des axes X et Y



#### IV.3.4 L'axe Z

C'est la partie la plus difficile de notre construction. Nous avons besoin de quelques outils pour l'attacher sur l'axe des Y, une surface plane. Sur cette surface, raccordez le servomoteur (axe Z) et la base du stylo. Le stylo (ou le crayon) doit être en mesure de déplacer vers le haut et vers le bas à l'aide du servomoteur.

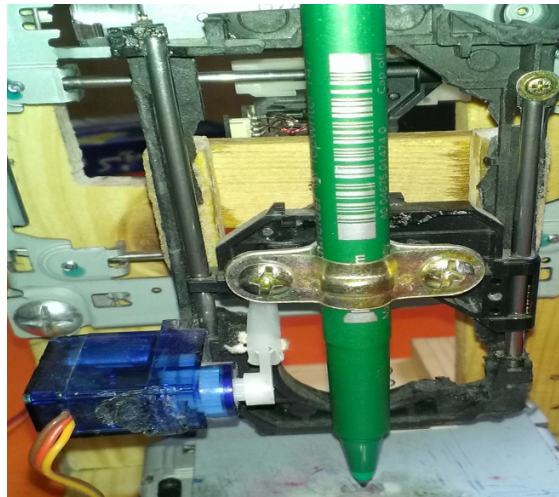


Figure IV.9 : Fixation axe Z

#### IV.3.5 table de dessin

Maintenant, l'étape suivante c'est l'attacher d'une surface en métal (ou en plastique) sur l'axe des X (5x5cm seront bien).



Figure. IV.10: Support base de papier.

### IV.3.6 Branchement du circuit

Le câblage du CNC machin est assez simple. Il est composé de 2 circuits L293D. Une plaque d'essai.

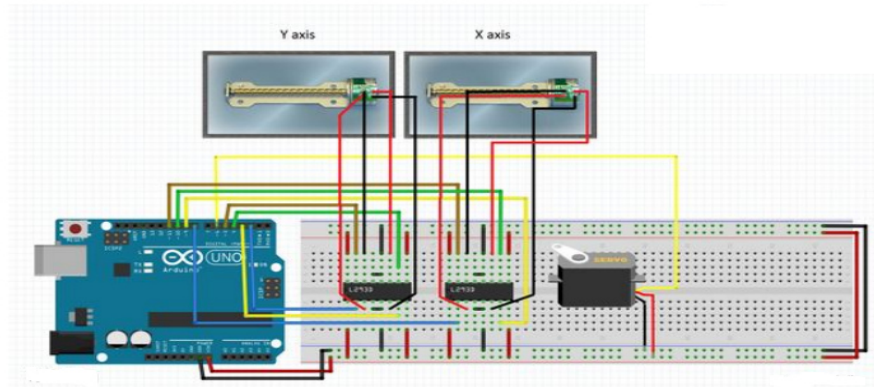


Figure. IV.11: montage CNC machin.

### IV.3.7 Assemblage du CNC machin Axes X et Y et Z

Commençons par fixer l'axe X à l'aide des 8 écrous et de 4 vis et 8 rondelles. Fixons l'axe Y sur le montant verticale l'aide des 8 écrous et de 4 vis et 8 rondelles. La hauteur n'est pas importante, elle sera réglée ultérieurement en ajustant simplement la position du crayon dans son support. Puis nous fixons une plaque plastique dans l'axe Y avec un vis, pour Fixer l'axe z à cette plaque à l'aide des 2 écrous et de 1 vis et 2 rondelles



Figure. IV.12: CNC machin

Maintenant les deux axes sont prêt a utiliser juste on fait le câblage avec une carte arduino UNO et un circuit de pilotage, il est le temps de tester le bon fonctionnement les moteurs pas à pas (axe X et Y).

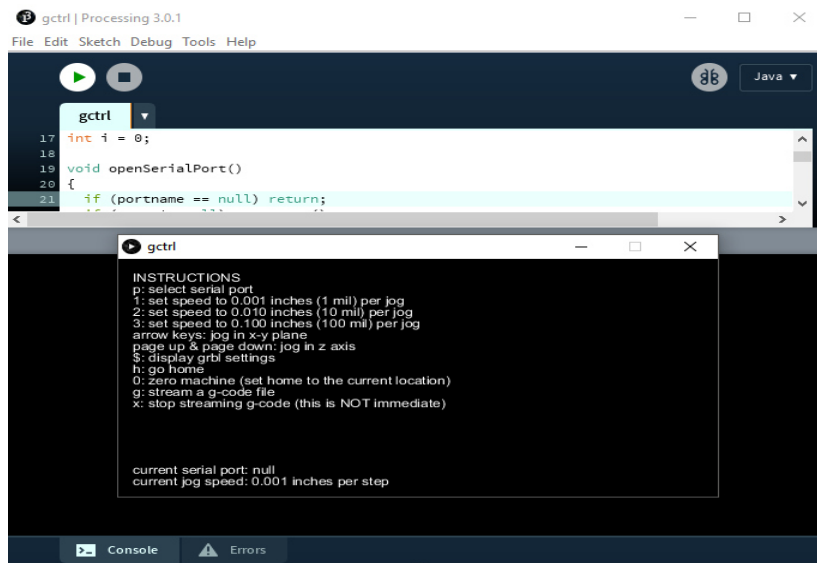
#### IV.3.8 Test des axes X / Y/Z

Le câblage et l'assemblage est réalisé, nous allons maintenant pouvoir tester le bon fonctionnement de mouvements des axes X et Y, et pour cela, on a besoin de crie un fichier code-G simple un rectangle qui permet de déplacé le stylo sur le plan (x, y).

```
//  
G1 X0.00 Y0.00 F3500.00 // point de départ le trace  
M300 S30.00 (pen down) // le stylo a l'état bas  
G4 P150 (wait 150ms) // temps d'arrêt 150 ms  
G1 Y40.00 F3500.00 // déplacement sur axe Y 40 mm (direction +)  
G1 Y30.00 F3500.00 // déplacement sur axe Y 30 mm (direction +)  
G1 X-40.00 F3500.00 // déplacement sur axe X 40 mm (direction -)  
G1 Y-30.00 F3500.00 // déplacement sur axe Y 30 mm (direction -)  
M300 S50.00 (pen up) // le stylo à l'état haut  
G4 P150 (wait 150ms) // temps d'arrêt 150 ms  
//
```

**Figure. IV.13** : un petit programme Gcode pour tracer un rectangle.

Maintenant, nous sommes prêts à imprimer notre première image Pour ce la, nous utiliserons le programme de traitement gctrl.pde. Ce programme envoie des fichiers "Gcode" au traceur CNC.

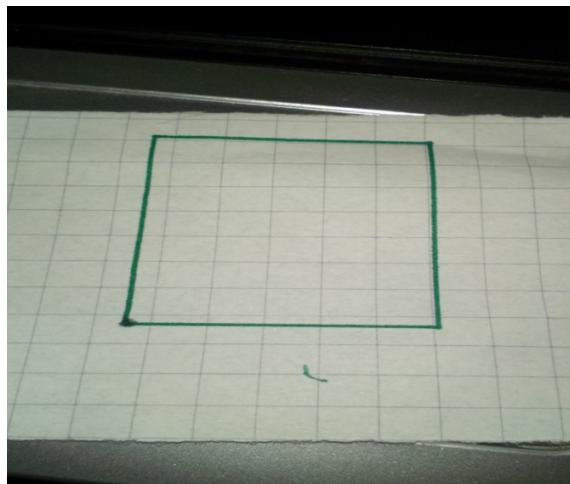


**Figure. IV.14:**Programme GCTRL.

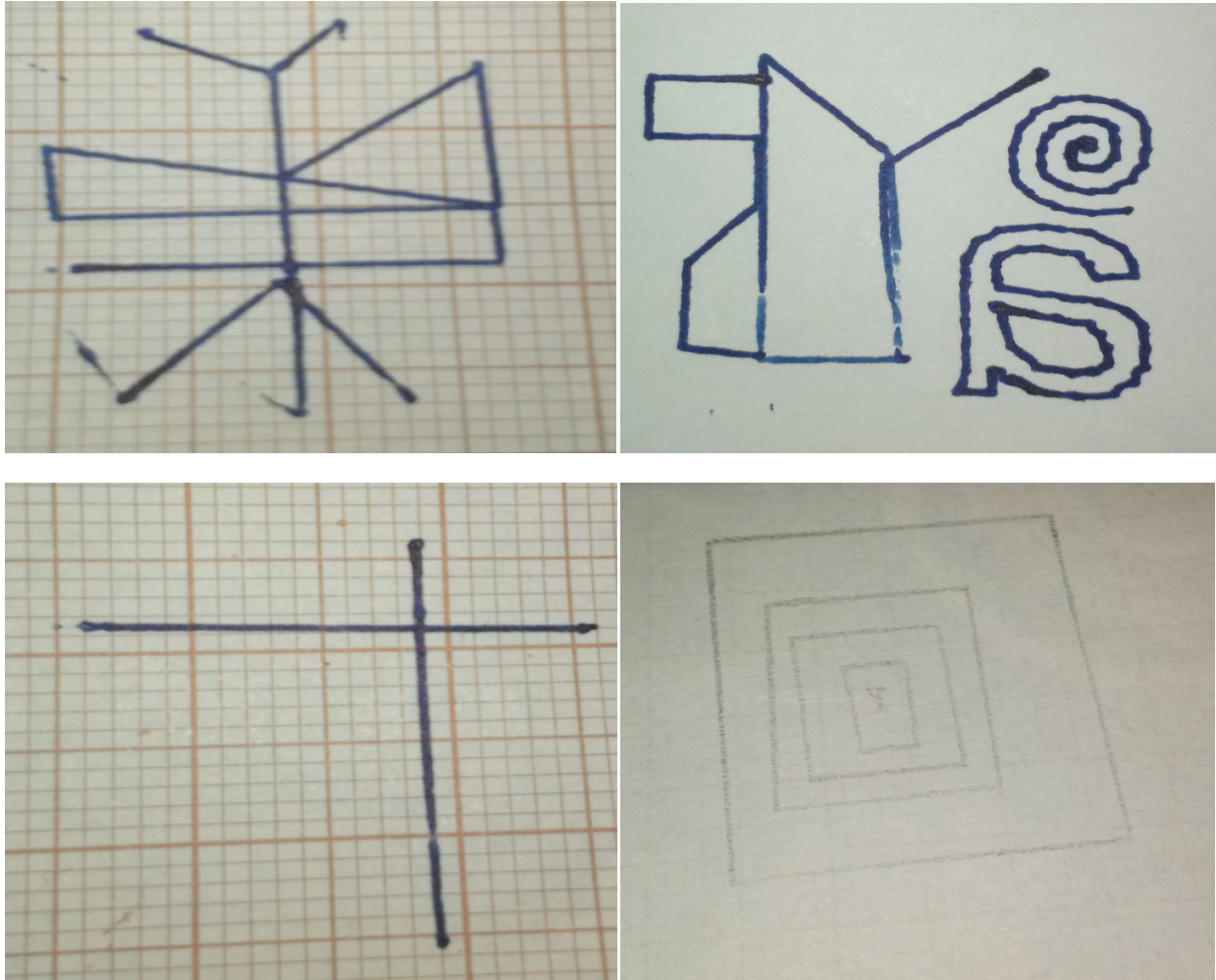
Cliquez sur l'icône "play" pour démarrer le programme et affiche une interface de GCTRL.

Cette interface nous permet de contrôler notre système à l'aide des étapes suivantes :

- Taper sur P : la sélection du sérial porte (COM) assuré al communication entre le pc et la carte arduino.
- Taper sur G : exporter le fichier Gcode



**Figure. IV.15:** le résultat de teste de rectangle.

**IV.3.9 Des tests supplémentaires pour assurer la fiabilité du système****Figure. IV.16:** le résultat des autres tests**IV.4 Conclusion**

A travers ce chapitre nous décrivons toutes les étapes pour la réalisation de notre CNC ; avant de passer à la réalisation nous avons simulés d'abord la commande du moteur pas à pas via la carte Arduino avec le logiciel Isis Proteus afin de bien s'assurer du bon fonctionnement de notre programme qu'on doit implémenter sur la carte Arduino. Le bon fonctionnement de la commande a été validé avec succès par des tests des moteurs pas à pas.

La phase la plus cruciale est le test du programme implémenté à travers une interface IDE Arduino ainsi que la maîtrise des logiciels Inkscape et Processing, en tenant compte des liens existants entre ces logiciels et le programme implémenté sur Arduino en respectant les étapes d'exécution pour



parvenir à transférer les coordonnées X, Y traduit sous forme de mouvement via les moteurs pas à pas. Ceci a nécessité un temps important de documentation pour extraire l'essentiel d'information pour la commande de notre machine.

Nous citerons aussi la difficulté de la mise en œuvre de notre machine à partir des composants de récupération et la fixation et l'emplacement exacte des supports pour le pilotage des moteurs pas à pas suivant les directions X et Y afin d'assurer une bonne stabilité du mouvement.

En fin de parcours, nous avons réussi à effectuer les tests nécessaires et valider le bon fonctionnement de notre réalisation avec des motifs basiques tels que, l'exécution graphique des traits horizontaux, verticaux et horizontaux, des rectangles et des triangles a différentes dimensions.

À travers ce modeste travail, nous avons eu l'opportunité d'aborder plusieurs domaines qui constituent notre système pour la réalisation de notre projet à savoir, l'électronique, la microinformatique et la mécanique et surtout mis en pratique les connaissances acquises à travers notre formation et pu découvrir et explorés un domaine très intéressant dans le monde de l'industrie.

## Conclusion général

Ce mémoire nous a été très intéressant car il nous a montré le chemin de toute une démarche afin de régler une problématique qui est celle de la conception et la réalisation d'un système à base d'une commande numérique pour la fabrication des canaux micro fluidique. Ces canaux sont très utiles dans le des MEMS (Micro Electro Mechanical Systems) et ils ont beaucoup d'applications surtout dans le biomédical comme les laboratoires sur puce, circuits très développés ayant pour but l'analyse automatique du sang ou même d'urine. Et nous avons pu matérialiser ce système à travers une réalisation (un prototype) qui comme tout prototype préliminaire reste en phase de développement.

Nous avons d'abord entamé la partie de la commande numérique mouvement qui nous a donner des notions générale sur les machines outil a commande numérique (MOCN), leur fonctionnement et sont Domain d'application que nous avons étudié en général dans le chapitre 1. Cette partie est essentielle au fonctionnement des machines à commande numériques.

Dans la deuxième partie, nous avons fait une description bien détaillée sur les dispositifs nécessaires pour la réalisation d'un système à base d'une commande numérique tels que les moteurs pas à pas , la carte Arduino Uno et les circuits intégrés nécessaires pour le pilotage des moteurs pas à pas (L293D,schiold L293D).

Ensuite nous avons enchainée une étude sur le processus de communication entre les divers dispositifs. C'est pour cela qu'on a opté pour un logiciel de numérisation afin de convertir des images ou textes en un fichier data (G-code), qui est se composé d'infinité de points définies par des coordonnées bien déterminées. Ce fichier (G-code) va être transféré par un autre logiciel (Processing) pour assurer l'implémentation du fichier Gcode sur la carte Arduino. Cette dernière a été déjà préparée par un code arduino afin de faciliter la commande des moteurs a travers le fichier Gcode.

Avant de passer a l'étape de réalisation, on a fait des simulations des circuits déjà développés théoriquement, puis on a matérialisé ces circuits en implantant les différents composants dans une plaque d'essai afin de tester pour assurer le bon fonctionnement de ces circuits. une fois que ces tests ont été terminés, on a passé à la phase de réalisation de notre système (prototypage).

Nous avons rencontré plusieurs problèmes durant la conception de la commande et la réalisation pratique (protocole de communication, couple des moteurs et leurs vitesses...), mais grâce à ce que nous avons appris durant nos études académiques et l'assistance de nos encadreurs, nous avons pu confronter ces difficultés.

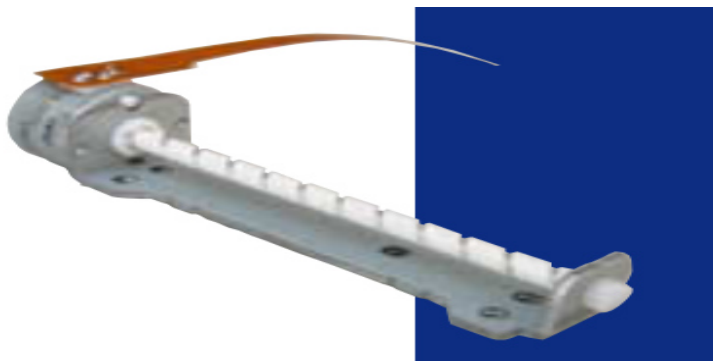
En effet, ce projet a été une source de découverte de plusieurs domaines d'études telles que l'informatique pour la programmation embarquée et le design des schémas électroniques, sans oublier le savoir-faire dans le domaine micro-électronique qui consiste à fabriquer des canaux microfluidiques.

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

- La commande intelligente.
- La communication son fil.
- Développement du système pour la fabrication des canaux microfluidiques.



**Annexe 1 : Moteur pas à pas (lecteur DVD) [28]**



**Figure A.1-1 :** Vue générale du moteur pas à pas employé.

**Reference Characteristics**

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

**Tableau A1-1 :** Caractéristiques de références

**Switching Sequence  
(Viewed from Lead Screw Side)**

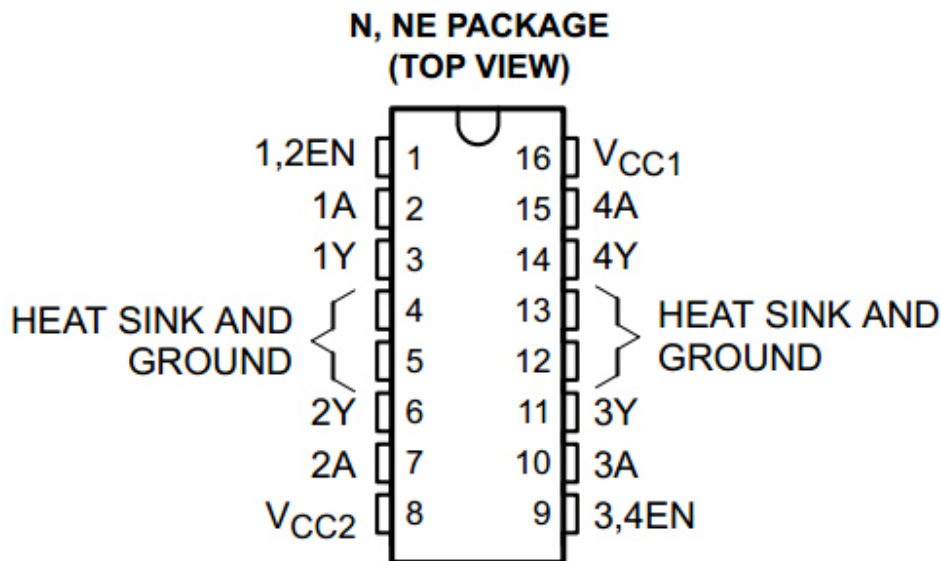
STEP \ TERMINAL	A1	A2	B1	B3
1	-	+	+	-
2	-	+	-	+
3	+	-	-	+
4	+	-	+	-

↓  
 CW
 

 ↑  
 CCW

**Tableau A1-2 :** Table de fonctionnement du moteur pas à pas utilisé.

**Annexe 2 : Circuit de commande (L293D) [13]**



**Figure A.2-1 : Circuit de brochage (PIN circuit)- L293**

- **Featuring Unitrode L293 and L293D Products Now From Texas Instruments**
- **Wide Supply-Voltage Range: 4.5 V to 36 V**
- **Separate Input-Logic Supply**
- **Internal ESD Protection**
- **Thermal Shutdown**
- **High-Noise-Immunity Inputs**
- **Functional Replacements for SGS L293 and SGS L293D**
- **Output Current 1 A Per Channel (600 mA for L293D)**
- **Peak Output Current 2 A Per Channel (1.2 A for L293D)**
- **Output Clamp Diodes for Inductive Transient Suppression (L293D)**

**Figure A.2-2 : Principales caractéristiques de circuit L293D**

### Annexe 3 : Le G code [25]

X	Position absolue
Y	Position absolue
Z	Z Position absolue
F	vitesse de déplacement
N	Numéro de ligne
P	Temps de pause ou option passée à un sous programme ou appel de sous programme
T	Sélection d'outil
I	Axe X des données d'un arc
J	Axes-Y des données d'un arc

**Tableaux A3-1** : Liste explicative des lettres de commande.

## Common Fanuc G Codes

G00	Déplacement rapide
G01	Interpolation linéaire
G02	Interpolation circulaire (sens horaire)
G03	Interpolation circulaire (sens anti-horaire)
G04	Arrêt programme et ouverture carter (pour nettoyer) (temporisation - suivi de l'argument F ou X en secondes)
G10/G11	Écriture de données/Effacement de données
G17	Sélection du plan X-Y
G18	Sélection du plan X-Z
G19	Sélection du plan Y-Z
G20	Programmation en pouces
G21	Programmation en mm
G28	Retour à la position d'origine
G31	Saute la fonction (utilisé pour les capteurs et les mesures de longueur d'outil)
G33	Filetage à pas constant
G34	Filetage à pas variable
G40	Pas de compensation de rayon d'outil
G41	Compensation de rayon d'outil à gauche
G42	Compensation de rayon d'outil à droite
G90	Déplacements en coordonnées absolues
G91	Déplacements en coordonnées relatives
G94/G95	Déplacement en Pouce par minute/Pouce par tour
G96/G97	Vitesse de coupe constante (vitesse de surface constante)/Vitesse de rotation constante ou annulation de G96

**Tableau A3-2 :** Liste des Fonctions communes du G code.

Code	Désignation
M00	Arrêt programmé
M01	arrêt optionnel ou avec conditio
M02	Fin de programme pièce
M03	Rotation de broche sens horaire
M04	Rotation de broche sens anti-horaire
M05	Arrêt de broche
M06	Changement d'outil
M07	Arrosage N°1
M08	Arrosage N°2

## Annexe

---

M09	Arrêt des arrosages
M41	Gamme de rotation 1
M42	Gamme de rotation 2

**Tableau A3-3** : Liste des Fonctions auxiliaires M.

## Références

- [1] Fenniche abdeerrezak. « Etude et réalisation 'une connexion RS232 de la fraiseuse EMOC F1 CNC et le tour EMCO compact 5 CNC avec le PC ». Mémoire Master Professionnel Univ-kasdi merbah Ouargla, soutenu le : 26/06/2013.
- [2] Farid ASMA..Notes de cours. 2007/2008
- [3] Abdennadji Youssef, Ahmadi Anouar « Conception et réalisation d'une commande numérique d'une machine de coup laser », Ingénieur d'école nationale de Gabés, soutenu en juin 2010.
- [4] Chaia.H et Ledouali.N. « Etude et réalisation d'un table CNC ». Mémoire Master .Univ-mhamed bougerra- Boumerdès. Année : 2015/2016.
- [5] « Systèmes électromécaniques» ; Haute Ecole d'ingénierie et de Gestion Du Canton du Vaud, CD/SEM/Cours/Chap07, 04/2014.
- [6] Djedouani. Belal,Ould amhed haiballa mohamed lemine, « Développement d'un Banc d'essai pour l'étude est le diagnostic des moteurs à pas Hybrides », Mémoire de Fin d'Etudes pour l'obtention du Diplôme de Master en Electrotechnique Université de Tébessa. Année : 2015/2016.
- [7] Mr Taleb Bendiab Zakaria. « Etude et réalisation d'une carte de commande de trois moteurs pas a pas pour machine de prototypage rapide ». Mémoire Master. Université Abou bekr Belkaid – Tlemcen. Année : 2013 /2014.
- [8] <http://www.positron-libre.com/electronique/moteur-pas-a-pas/sequence-commande-moteur-pas-a-pas.php>, téléchargé le : 07/2017.
- [9] Takashi Kenjo & Akira Sugawara, «Stepping motors and their microprocessor controls » Oxford University Press , ISBN 0-19-859385-6. Année: 1983
- [10] M. Correvon. « LES MOTEURS PAS À PAS ». (CD\SEM\Cours\Chap7.doc, téléchargé le : 08/09/2017
- [11] Gerard Yvraut. (1999). « Les Moteurs Pas à Pas ». Séminaire Bellegarde Novembre 1999.
- [12] <http://www.framboise314.fr/un-graveur-laser-a-commande-numerique-a-base-de-raspberry-pi/>, téléchargé le : 08/2017
- [13] [www.datasheetcatalog.com](http://www.datasheetcatalog.com) téléchargé le : 08/2017
- [14] <https://www.cf-techno.com/shield-moteur-l293d>, téléchargé le : 08/2017
- [15] A. Ahmed Azi, « Contribution au développement d'un outil d'aide au diagnostic de performances des MPPT dans la chaine de conversion photovoltaïque », Mémoire magister, Université de M'sila, Algérie, 2014.

- [16] LECHALUPÉ Julien.. « Cours d'initiation à Arduino ». Université Paul Sabatier : Mai 2014.
- [17] X.HINAULT.www.mon-club-elec.fr. Téléchargé le : 04/09/2017
- [18] Christian Tavernier,Dound,«Arduino applications avancées».17/10/2012.
- [19] <http://www.acm.uiuc.edu/sigbot/tutorials/2009-11-17-arduino-basics>
- [20] S.V.D.Reyvanth, G.Shirish, «PID controller using Arduino».
- [21] <http://www.generationrobots.com/fr/152-arduino>.Consulter le: mars 2015.Téléchargé le 04/09/2017
- [22] Simon landrault ,hippoltye weisslinger, « Arduino : Premiers pas en informatique embarquée » ; déposé le : 19 juin 2014, Le blog d'Eskimon
- [23] <http://inkscape.fr>, téléchargé le : 04/09/2017
- [24] Cédric Gémy.. »Réussir ses dessins vectoriels Inkscape efficace », ISBN : 978-2-212-12425-5, 2009.
- [25] FARID BETTINE. « Etude du processus d'usinage des pièces mécaniques sur les machines a 5 axes». Mémoire Magistère en Génie Mécanique Université Mohamed Khider – Biskra. 2012.
- [26] Hechmi chermiti, « programation iso, leçon 3 », ISET kairouan ; téléchargé le : 04/09/2017
- [27] J.michel et géridan J.noel lafargue, Poressing. « s'initier à la programmation créative », ISBN 978210073840. Janvier 2016.
- [28] <http://robocup.idi.ntnu.no/wiki/images/c/c6/PL15S020.pdf>, téléchargé le : 04/09/2017

## Résumé :

Dans ce travail, nous avons fait une description généralisée sur les machines à commande numérique (machines CNC) ainsi que sur les moteurs pas à pas. On a utilisé ces moteurs afin de commander les machines CNC. Pour atteindre ce but on a employé des circuits de commande (type L293 D) ainsi qu'une carte Arduino UNO et on a expliqué leurs caractéristiques en général. Néanmoins, cela nécessite une partie de programmation de tout ce système. Cette programmation a été réalisée par des logiciels tels que : - Proteus afin de simuler la machine CNC, -Inkscape pour convertir les textes et dessins en un fichier G Code (langage de commande des machines CNC), - Processing afin de créer une interface virtuelle permettant le transfert du fichier G Code vers la carte d'Arduino.

Ce travail a été couronné par une réalisation d'un prototype pratique d'une machine CNC, on l'a testé pour faire la tâche de dessin sur papier tout en respectant toutes les étapes nécessaires citées en haut. Les résultats étaient probants.

## Abstract :

In this work, we have made a general description of computer numerical controller of machines (CNC machines) as well as stepper motors. These motors have been used to control CNC machines. To achieve this aim, we used control circuits (type L293 D) as well as an Arduino UNO card and we explained their characteristics in general. Nevertheless, this requires a programming part of this whole system. This programming was carried out by software such as: - Proteus to simulate the CNC machine, -Inkscape to convert text and drawings into a G Code file (CNC machine control language), - Processing to create a virtual interface allowing the transfer of the G Code file to the Arduino card.

This work was finalized by a realization of a practical prototype of a CNC machine; it was tested by making different drawings on paper while respecting all the necessary steps mentioned above. The results were convincing.

## ملخص:

لقد قمنا في مذكرتنا هاته بنظرة عامة حول آلة التحكم الرقمي, و بدراسة حول محرك خطوة بخطوة وأيضا تطبيق التحكم على هذا الأخير باستعماله في الآلة .  
وقمنا باستعمال متحكم من نوع L293D و كارت اردوينو مع شرح خصائصهما , إضافة إلى عدة برامج كالبر و تيس (protues) من اجل محاكاة آلة التحكم , و برنامج إنكسكيب ( inkscape ) من اجل تحويل النصوص والصور إلى لغة الآلة الجي كود (Gcode) , ثم التحكم في إرساله عن طريق برنامج البروسببنغ (processing) لذلك قمنا بواسطة هذا الأخير بتحرير واجهة من اجل التحكم في إرسال آل ( Gcode ) إلى الكارت اردوينو . وأخيرا قمنا بصناعة آلة تحكم رقمي واختبارها بالرسم على الورق.