

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU



FACULTE DU GENIE ELECTRIQUE ET D'INFORMATIQUE  
DEPARTEMENT D'AUTOMATIQUE

**Mémoire de Fin d'Etudes**  
**De MASTER ACADEMIQUE**  
Domaine : **Sciences et Technologies**  
Filière : **Génie électrique**  
Spécialité : **commande des systèmes**

**Présenté par :**

**Djamel SMAINI**  
**Rafik BEN KADI**

Mémoire dirigé par    Mr MELLAH Rabah  
   Mr DJIOUA Smail

**Thème**

**Conception et réalisation d'une machine  
CNC**

*Mémoire soutenu publiquement le 17/09/ 2015 Devant les jurys composé de :*

**MCB, Hakim ACHOUR**  
Grade, Dpt d'automatique, Président

**MCA, Rabah MELLAH**  
Grade, Dpt d'automatique, Promoteur

**MAB, Ouiza BOUKENDOUR**  
Grade, Dpt d'automatique, Examineur

**MCA, Mhammed DAOUI**  
Grade, Dpt d'informatique, Examineur

**Mr    Smail DJIOUA , co-promoteur**

## Remerciements

---

Nous tenons à remercier profondément Dieu, le tout puissant de nous avoir donné le courage, la volonté, ainsi que la santé pour réaliser ce modeste travail.

Ce travail a été effectué au Laboratoire de Recherche ETS SAHI  
\_DZLABO TIZI OUZOU

Nous tenons à exprimer nos vifs remerciements et nos sincères reconnaissances à notre promoteur **Mr MELLAH Rabah** ainsi qu'à notre co-promoteur **DJIOUA SMAIL**, pour leurs disponibilités, leurs conseils judicieux, leurs directives et leurs orientations et d'avoir accepté de diriger ce projet et de nous avoir soutenues au long travail tout en nous accordant leurs confiances et en nous faisant profiter de leurs larges expériences tout au long de la réalisation de ce projet de fin d'étude.

Nos profondes gratitudes, d'avance aux membres du jury qui feront l'honneur d'évaluer ce travail de fin d'études.

Tous nos infinis remerciements vont à tous les enseignants qui ont contribué à notre formation durant notre cursus, pour le riche savoir qu'ils nous ont transmis avec rigueur et dévouement.

Enfin, on exprime notre reconnaissance envers toute notre famille, tous nos amis, camarades et envers tous ceux qui ont contribué de près ou de loin à l'aboutissement de ce travail ne serait ce par un modeste sourire d'encouragement à notre égard.

## Dédicaces

---

### **Je dédie ce modeste travail à :**

- Mes très chers parents qui ont fait de moi ce que je suis aujourd'hui et qui ont veillé à guider mes pas durant toute ma vie par leurs aides, leur grandes émotions, leur sacrifices, et leur soutien et encouragement pendant la réalisation du projet de fin d'étude.
- Mes très chères sœurs : Farida, Nedjwa, Fatiha, et Khalida
- Ma chère tante : Nadia
- Ma chère compagne : N.Malha
- Mon binôme : Nabil
- Mes amis de la promotion : Rafik Mansouri, Madjid Badja, Sofiane Djema, Nordine Nasri, Fateh Kouchi, Lilia Messaoudene, Kaoui Samir
- Mes amis : B.Sakina, A.Chabha, B.Malika, B.Tawes, D.Hocine, H.Hamza, A.Sid Ahmed, D.Sofiane, M.Idir, Kahina, M.Kamel, S.Karim, D.Kouceila, M.Madjid, K.Adam, A.Mouhand, M.Nafaa, H.Nabil, S.Yacine, S.Yacine, A.Aissa, S.Amar, A.Mourad, F.Ghiles, F.Rabah, O.Hecene, B.Mahdi, K.Omar, Y.Mohand, A.Tarik,
- Toute la promotion Automatique 2015

**DJAMEL**

## Dédicaces

---

### **Je dédie ce modeste travail à :**

- Mes très chers parents qui ont fait de moi ce que je suis aujourd'hui et qui ont veillé à guider mes pas durant toute ma vie par leurs aides, leur grandes émotions, leur sacrifices, et leur soutien et encouragement pendant la réalisation du projet de fin d'étude.
- Mes très chères sœurs : Farida, Nedjwa, Fatiha, et Khalida
- Ma chère tante : Nadia
- Ma chère compagne : N.Malha
- Mon binôme : Nabil
- Mes amis de la promotion : Rafik Mansouri, Madjid Badja, Sofiane Djema, Nordine Nasri, Fateh Kouchi, Lilia Messaoudene, Kaoui Samir
- Mes amis : B.Sakina, A.Chabha, B.Malika, B.Tawes, D.Hocine, H.Hamza, A.Sid Ahmed, D.Sofiane, M.Idir, Kahina, M.Kamel, S.Karim, D.Kouceila, M.Madjid, K.Adam, A.Mouhand, M.Nafaa, H.Nabil, S.Yacine, S.Yacine, A.Aissa, S.Amar, A.Mourad, F.Ghiles, F.Rabah, O.Hecene, B.Mahdi, K.Omar, Y.Mohand, A.Tarik,
- Toute la promotion Automatique 2015

**DJAMEL**

# Sommaire

---

Introduction générale.....	1
Historique.....	2
<b>Chapitre 1 : généralité sur les machine à commande numérique par calculateur</b>	
I.1.Introduction .....	3
I.2.Définition de la commande numérique .....	3
I.3.Définition d'une machine CNC .....	3
I.3.1 Partie opérative.....	4
I.3.2 Partie commande.....	4
I.4. Domaine d'utilisation.....	5
I.5.Principe de fonctionnement.....	5
I.6.Les différents types des CNC.....	6
I.7. Classification des machines CNC .....	7
I.7.1. Classification des CNC selon le mode de fonctionnement.....	7
I.7.2. Classification des CNC selon le mode d'usinage .....	9
I.8.La structure d'une machine CNC.....	10
I.8.1 Le système électronique.....	10
I.8.2.Le système mécanique .....	10
I.8.3.Le système de programmation .....	11
I.8.9.Conclusion.....	12
<b>Chapitre 2: conception d'une machine CNC</b>	
II.1.Introduction .....	13
II.2. Cahier de charge .....	13
II.2.1. Fonction principale.....	14
II.2.2. Fonction partielles.....	14
II.3. Variantes solutions proposées .....	15
II. Partie mécanique .....	15

# Sommaire

---

II.4. System de guidage.....	16
II.4.1. Douilles à billes .....	16
II.4.2. Rail de guidage pour guidage prismatique .....	16
II.5. Déplacement suivant les axes x y z.....	17
II.5.1. Courroie.....	17
II.6. Dimensionnement.....	17
II.6.1 Détermination des masses.....	17
II.6.2 Dimensionnement des vis trapézoïdale.....	19
II. La partie électronique .....	26
II.7. Les Actionneurs .....	26
II.7.1. Généralité sur les moteurs pas à pas.....	26
II.7.2. Les différents types de moteurs pas à pas.....	28
II.7.3. Comparaison des 3 types de moteurs.....	31
II.7.4. Configuration interne des bobines des moteurs .....	31
II.8. Les pilotes des moteurs pas à pas .....	32
II.8.1. Commande des moteur pas à pas avec A4983.....	32
II.8.2. Commande des moteurs pas à pas avec L297 .....	34
II.9. Les capteurs 36 .....	
II.9.1 Définition des capteurs.....	36
II.9.2 Les différents types de capteurs .....	36
II.9.3. Quelques exemples sur les capteurs .....	37
II.10. Les microcontrôleurs.....	37
II.10.1. Définition.....	37
II.10.2. Composants intégré.....	38
II.10.3. Les avantage des microcontrôleurs .....	39
II.10.4. Les défauts des microcontrôleurs.....	39

# Sommaire

---

II.10.5. Familles de microcontrôleurs.....	40
II.11. Arduino .....	41
II.11.1 Vue d'ensemble .....	41
II.11.2. Pourquoi Arduino.....	42
II.11.3. Alimentation.....	42
II.11.4.la mémoire .....	43
II.11.5. les entrées et sorties .....	43
II.11.6. communication .....	44
II.11.7.programmation .....	44
II.12. Commande PWM (Pulse Width Modulation).....	47
II.12.1. Définition.....	47
II.12.2. INTÉRÊT .....	47
II.14.3.FRÉQUENCE .....	48
II.12.4. RAPPORT CYCLIQUE.....	48
II. Partie Informatique et Programmation .....	49
II.13. Organigramme .....	49
II.14. Les fichiers Gode .....	50
Le fonctionnement du G-Code .....	50
II.15. Le fichier GRBL.....	51
II.15.1. Schéma de Pin pour Grbl .....	52
II.15.3. Le branchement .....	53
II.15.4. Connecter grbl à Arduino .....	53
II.15.5. Paramètres de grbl.....	54
II.16. Universal Gcode sender .....	56
II.17. Utilisation du firmware grbl .....	57
II.17.1. paramétrages de la machine CNC.....	57
II.17.2. Initialisation de la machine .....	58

# Sommaire

---

II.18. Conclusion .....	59
-------------------------	----

## **Chapitre 3 : réalisation pratique**

III.1. Introduction .....	60
III.2. Structure mécanique .....	60
III.2.1. Les différentes parties de la machine CNC.....	60
III.3. Assemblage complet de la machine CNC.....	62
III.4. Carte de commande pour la machine CNC .....	62
III.5. Les actionneurs .....	63
III.5.1. Caractéristiques des moteurs .....	63
III.6. Conclusion .....	64

L'automatisation consiste à *rendre automatique* les opérations qui exigeaient auparavant l'intervention humaine, elle est considérée comme l'étape d'un progrès technique ou apparaissent des dispositifs techniques susceptibles de seconder l'homme, non seulement dans ses efforts musculaires, mais également dans son travail intellectuel de surveillance et de contrôle.

Un automatisme est un sous-ensemble d'une machine destinée à remplacer l'action de l'être humain dans des tâches en générales simples et répétitives, réclamant précision et rigueur. On est passé d'un système dit manuel, à un système mécanisé, puis au système automatisé.

Dans l'industrie, les automatismes sont devenus indispensables : ils permettent d'effectuer quotidiennement les tâches les plus ingrates, répétitives et dangereuses. Parfois, ces automatismes sont d'une telle rapidité et d'une telle précision, qu'ils réalisent des actions impossibles pour un être humain. L'automatisme est donc synonyme de productivité et de sécurité.

Le savoir-faire de l'opérateur est transposé dans le système automatisé, il devient le PROCESSUS. Un processus peut-être considéré comme un système organisé d'activités qui utilise des ressources (personnel, équipement, matériels et machines, matière première et informations) pour transformer des éléments entrants en éléments de sortie dont le résultat final attendu est un produit.

Le système automatisé de production industrielle doit également répondre à des contraintes économiques et donc à une certaine flexibilité.

Il existe deux concepts technologiques :

- Les automatismes séquentiels,
- Les asservissements, la régulation ou « feedback control ».

Cependant, dans la plupart des systèmes complexes modernes, ces deux types d'automatismes se trouvent imbriqués.

### **Historique :**

La fabrication de précision moderne exige une extrême précision dimensionnelle et la finition de surface. Une telle performance est très difficile à réaliser manuellement, voire impossible, même avec des opérateurs experts. Dans les cas où il est possible, il faut du temps beaucoup plus élevé en raison de la nécessité pour la mesure dimensionnelle fréquente pour prévenir la surexploitation. Il est donc évident que le contrôle de mouvement automatisé remplacerait le manuel de contrôle dans la fabrication moderne.

Le développement de commande numérique par ordinateur (CNC) a également rendu possible l'automatisation des processus d'usinage avec souplesse pour gérer la production de petites et moyennes séries de pièces.

C'était en 1947, lorsque la commande numérique est née. Tout a commencé lorsque John C. Parsons de la Parsons Corporation, Traverse City, Michigan, un fabricant de pales de rotor d'hélicoptère, ne pouvait pas faire ses modèles assez vite. Donc, il a inventé une façon de coupler l'équipement informatique avec une pyrale du gabarit. M. Parsons a utilisé les cartes pour faire fonctionner son système de Ron chiffres de poing.

1949 a été l'année d'un autre «besoin urgent». Le commandement américain Air Matériel s'est rendu compte que ses pièces pour avions et missiles ont été de plus en plus complexe. En outre, comme les dessins ont été constamment améliorés, les changements dans le dessin ont souvent été faites. Ainsi, dans leur recherche de méthodes de production plus rapide, un contrat d'étude de la Force aérienne a été attribué à la Parsons Corporation. Le laboratoire des mécanismes d'asservissement de l'Institut de Technologie du Massachusetts (MIT) était le sous-traitant.

En 1951, le MIT a repris le travail complet, et en 1952, le prototype de la machine NC d'aujourd'hui, une machine Cincinnati HydrotelMilling modifié, a été démontrée avec succès. La commande numérique terme a été l'origine au MIT.

Initialement, la technologie CNC a été appliqué sur les tours, fraiseuses, etc. qui pourrait effectuer un seul type d'opération de découpe de métal. Plus tard, on a tenté de traiter une variété de pièces qui peuvent nécessiter plusieurs types différentes opérations d'usinage et de les terminer en une seule mise en place. Ainsi centres d'usinage CNC capables d'exécuter plusieurs opérations ont été développés. Pour commencer, centres d'usinage CNC ont été développés pour les composants prismatiques usinage combinant des opérations comme fraisage, perçage, alésage et taraudage. Peu à peu, les machines pour la fabrication de composants cylindriques, appelés centres de tournage ont été développés

## I.1. Introduction

Dans ce chapitre on présente un aperçu général sur la technologie des CNC, citant la Classification, l'architecture, la programmation des Machines CNC pour différentes commandes

## I.2. Définition de la commande numérique

Apparue il y a seulement quelques dizaines d'années, la commande numérique CN impose actuellement sa technologie dans le monde de l'usinage. conçue pour piloter le fonctionnement d'une machine à partir des instructions d'un programme sans intervention directe de l'opérateur pendant son exécution, elle a dans un premier temps, permis de franchir un pas important dans l'automatisation des machines-outils traditionnelles comme les fraiseuses, perceuses, aléseuses...etc. et elles sont ainsi devenue capable d'assurer en qualité et en quantité une production à peine imaginable quelques années auparavant.

Aujourd'hui, de plus en plus étroitement associé aux progrès de la microélectronique et de l'informatique, la CN voit ses performances augmenter régulièrement tandis que, son prix et son encombrement ne cessent de diminuer.

Elle pénètre, de ce fait dans les petites entreprises et devient accessible à tous les secteurs industriels faisant appel aux procédés de positionnement ou de suivi de trajectoire.

Les CN actuelles sont capables d'effectuer des mouvements en combinant simultanément les positions sur les axes X, Y et Z.

Elles s'appellent commande numérique par ordinateur. De plus, elles sont capables de gérer plusieurs axes simultanément. On parle alors d'interpolation.

## I.3. Définition d'une machine CNC

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

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

Elle dispose d'une grande puissance de calcul et d'une plus grande souplesse d'utilisation qu'un automate programmable. Ce dernier est néanmoins qu'un complément de la commande CNC pour ce qui est de la gestion des entrées telles que : interrupteur, bouton d'arrêt d'urgence, ...etc.. En résumé, la commande CNC va générer des mouvements selon des consignes numériques

La machine-outil à commande numérique forme un ensemble comprenant :

### I.3.1. Partie opérative

Les mouvements sont commandés par des moteurs presque comparables à une machine-outil classique, et elle comprend :

- ❖ Une base, assurant l'indépendance de la machine au sol,
- ❖ Un bâti, dont les larges glissières sont en acier traité,
- ❖ un support outil (broche, torche, laser, jet d'eau ...),
- ❖ Une table support pièce, mobile selon 2 ou 3 axes, équipée de système de commande à vis et écrou à bille.
- ❖ des moteurs chargés de l'entraînement de la table,
- ❖ Un élément de mesure ou capteur de position renseignant à tout moment sur la position du mobile sur chaque axe,
- ❖ Une dynamo tachymétrique assurant la mesure de la vitesse de rotation

### I.3.2. Partie commande

Différente d'une machine conventionnelle et constituée d'une armoire dans laquelle on trouve :

- ❖ Le pupitre permettant de rentrer les commandes à l'aide d'un clavier,
- ❖ Le lecteur de données (ce lecteur peut être une option lors de l'achat de la machine),
- ❖ La sortie RS 232 pour les liaisons avec les Périphériques externes,
- ❖ L'écran de visualisation de toutes les données enregistrées,
- ❖ Le calculateur,
- ❖ les cartes électroniques (commandes d'axes, mémoire ...).

La partie commande est alimentée par un courant faible et ne peut donc pas alimenter directement les moteurs de la machine (voir schémas ci-dessous).

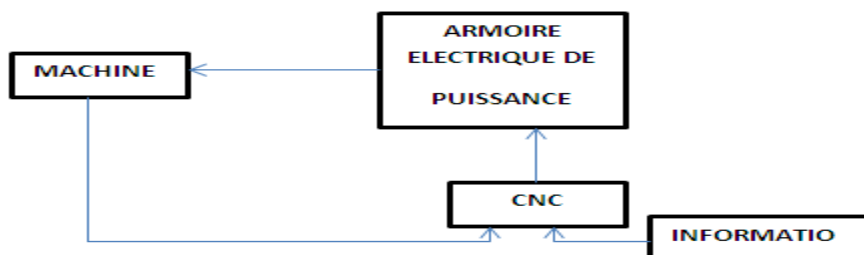


Figure I. 1. Fonction originale d'une commande numérique.

### I.4. Domaine d'utilisation

Les Machines CNC conviennent à la fabrication en petites et moyennes séries renouvelables. Elles permettent la réalisation, sans démontage, de pièces complexes comportant beaucoup d'opérations d'usinage.

Elles se situent entre les machines conventionnelles très "flexibles" réservées aux travaux unitaires (prototypes, maintenance) et les machines transferts, très productives, réservées aux grandes séries.

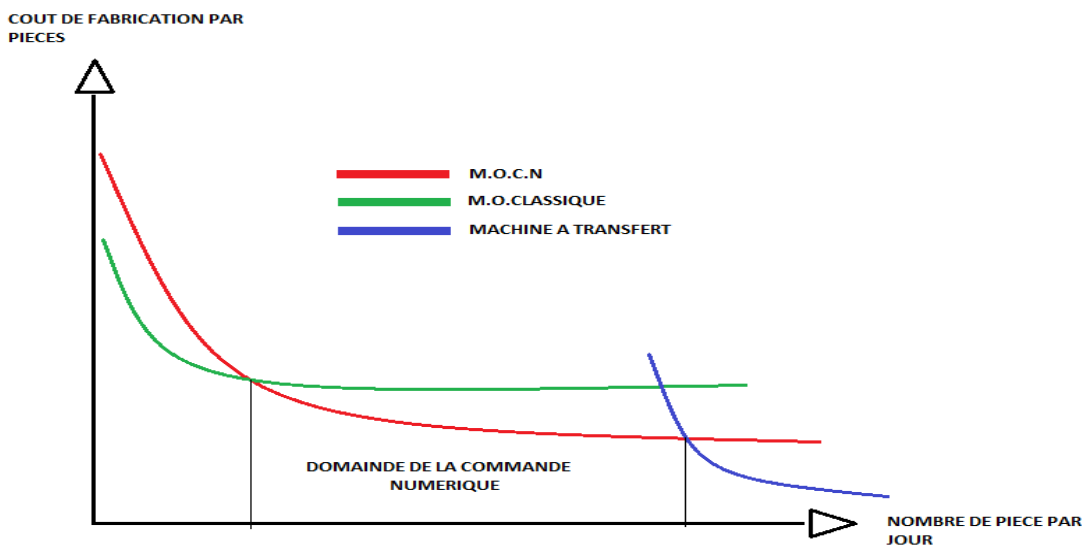


Figure. I. 2. Domaine d'utilisation des M.O.C.N.

### I.5. Principe de fonctionnement [1]

La machine CNC travaille avec des systèmes de contrôles en boucle fermée.

Des ordres vont être générés vers la commande par le biais d'un programme pièce 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.

La position sera régulée par la commande numérique alors que la vitesse sera le plus souvent régulée par le système d'asservissement moteur. On se trouve donc en face d'un système à deux boucles et l'on parle de système asservi.

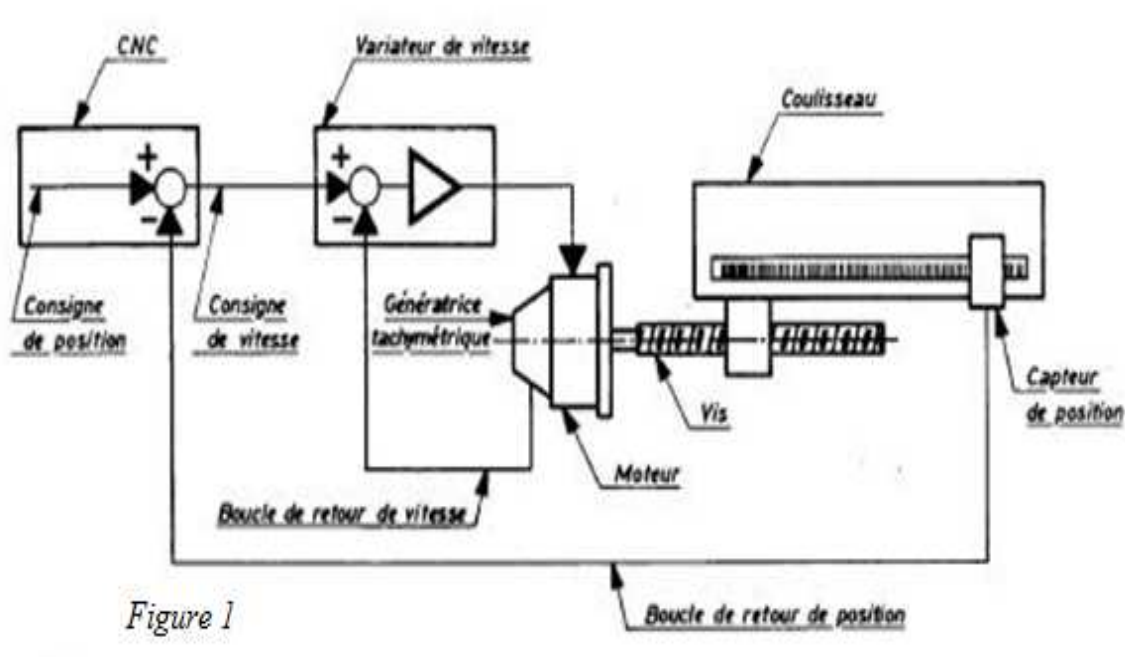


Figure 1

Figure. I.3 : Asservissement de la machine

Le schéma ci-dessous (figure. I.3), nous montre le flux d'informations avec ses directions, ainsi que les deux boucles de contrôles (position et vitesse).

## I.6. Les différents types de Machines CNC

On distingue plusieurs types de machines :

- ❖ les machines à enlèvement de copeaux : les perceuses, les tours, les centres de tournages, les fraiseuse, les centres d'usinage, les rectifieuses .....etc.
- ❖ les électroérosions : les machines à enfonçages, les machines à fil.
- ❖ les machines de découpes : oxycoupage, laser, jet d'eau...
- ❖ les presses : métal, injection plastique.
- ❖ les machines spéciales : à panneaux, à têtes multiples, de conditionnement (pour l'agroalimentaire)...

## I.7:Classification des machines CNC

Les machines-outils à commande numérique par calculateurs (CNC) sont classées suivant :

- ✓ Le mode de fonctionnement de la machine.
- ✓ Le mode d'usinage.

### I.7.1. Classification des CNC selon le mode de fonctionnement

#### a. Fonctionnement en boucle ouvert

En boucle ouverte, comme l'illustre la figure. I.4, le système assure le déplacement du chariot mais ne le contrôle pas.

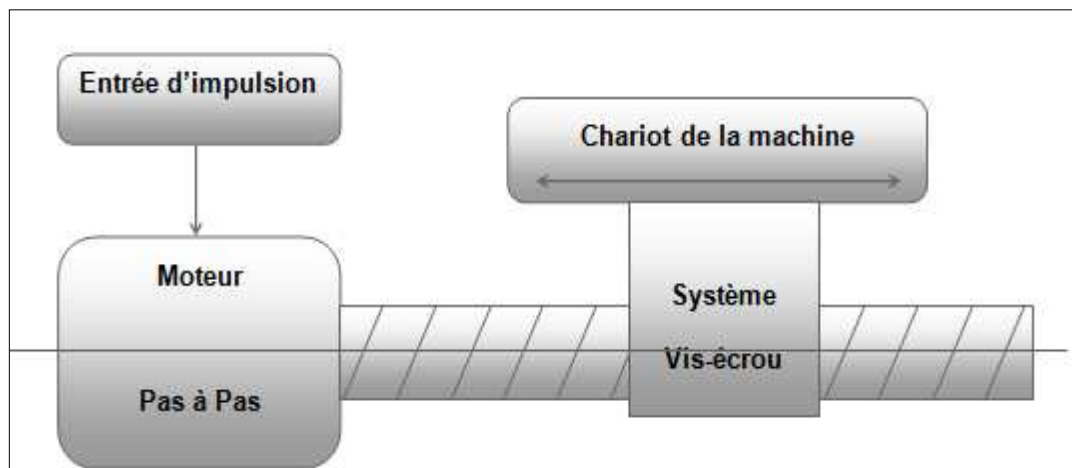


Figure .I.4 : Fonctionnement en boucle ouvert

#### b. Fonctionnement avec commande adaptative

La commande adaptative réalise d'une façon continue et automatique l'adaptation des conditions de coupe. Des capteurs relèvent les valeurs de couple de la broche, l'amplitude de vibration de la broche, la température au point de coupe. Ces informations sont transmises à une unité spéciale qui les envoie vers le directeur de commande numérique qui agit selon l'analyse des informations sur les conditions de coupe pour permettre une meilleure qualité de travail, une meilleure productivité et une plus grande sécurité.

La figure. I.5 illustre le fonctionnement de la commande adaptative

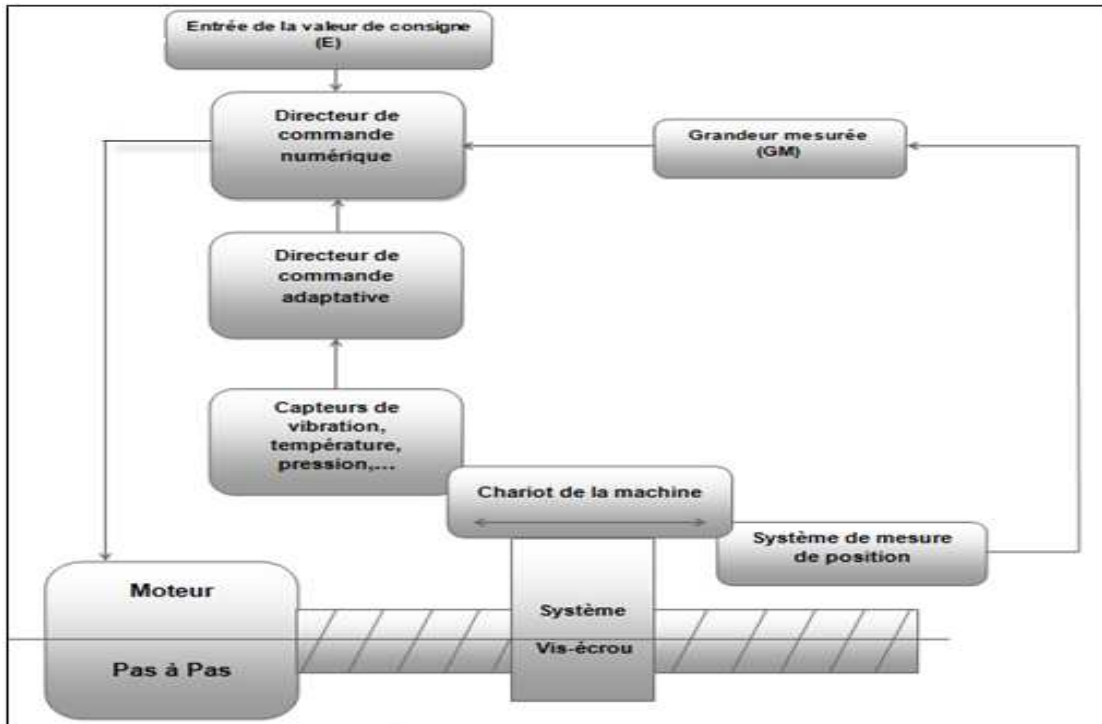


Figure .I.5.Commande adaptative.

**c. Fonctionnement en boucle fermé**

En boucle fermée le système contrôle le déplacement ou la position jusqu'à égalité des grandeurs entrée (E) dans le programme et celui mesuré (Gm).comme illustre la figure.I.6.

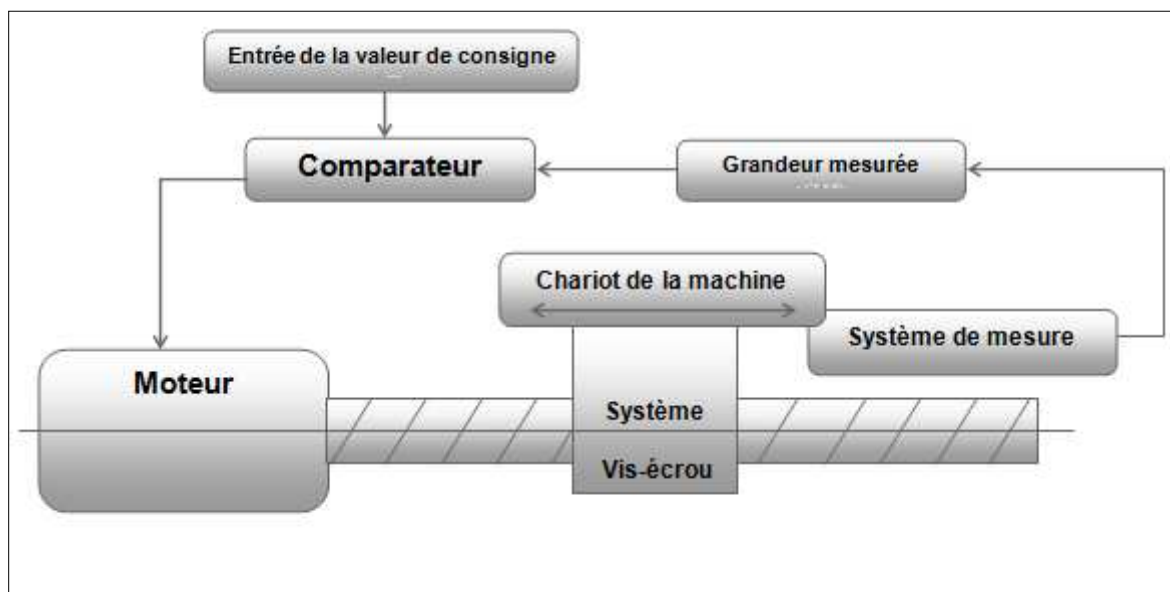


Figure .I. 6.Commande en boucle fermée.

### I.7.2. Classification des CNC selon le mode d'usinage

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

numérique point à point

Commande numérique de contournage

Commande numérique par axiale

**1) Commande numérique point à point :** c'est la mise en position de l'outil ou de la pièce Par déplacements non synchronisés. Le mouvement de coupe (usinage) n'est possible que lorsque le mouvement de positionnement est effectué.

- Exemples d'opération d'usinage : perçage, alésage, lamage taraudage, petit fraisage.

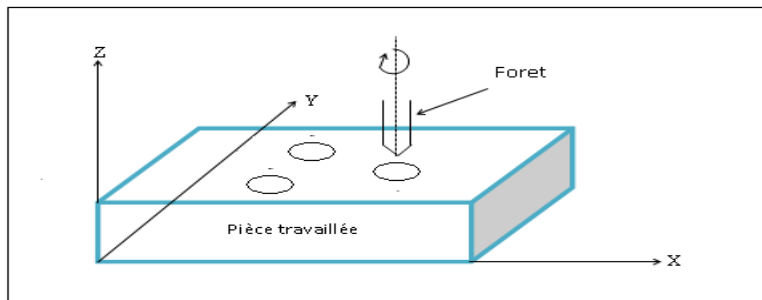


Figure .I.7. Commande Numérique Point A Point.

**2) Commande numérique par axiale :** ce sont des déplacements parallèles aux axes avec les vitesses d'avance programmée. Le mouvement de coupe et de positionnement sont synchronisés de façon à avoir un usinage selon des trajectoires parallèles aux axes de déplacement.

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

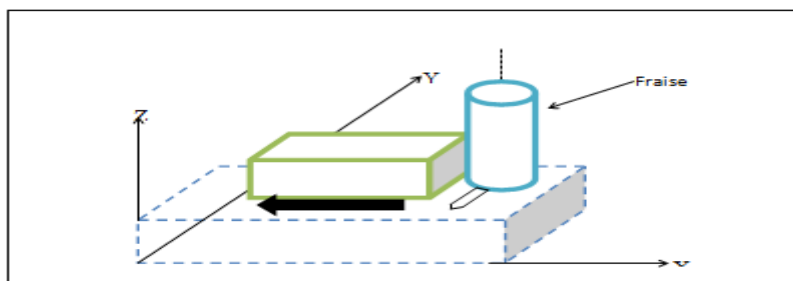


Figure.I. 8. Commande Praxiale.

3) **Commande numérique de contournage** : ce sont des déplacements qui synchronise des divers axes avec la vitesse d'avancement programmée. Les trajectoires sont décomposées en éléments de droites ou de cercles dans un ou plusieurs plans.

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

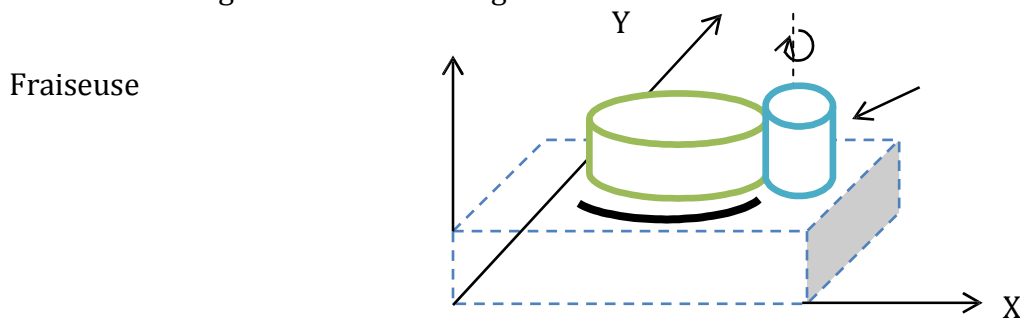


Figure.I. 9. Commande numérique de contournage.

## I.8. La structure d'une machine CNC

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

### I.8.1. Le système électronique

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

- [1] L'alimentation
- [2] La carte microcontrôleur
- [3] La carte de commande des moteurs pas à pas

### I.8.2. Le système mécanique

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

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

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

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

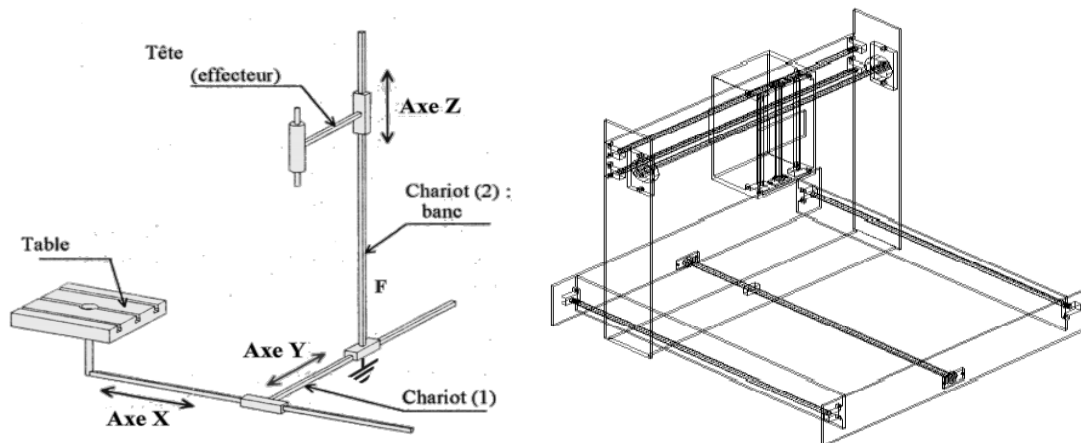


Figure I.10. : structure mécanique d'une machine CNC

### I.8.3. Le système de programmation

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

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

**I.9. Conclusion**

Dans ce chapitre on a décortiqué d'une manière générale une machine à commande numérique par ordinateur et l'intérêt de chaque partie qui constitue cette dernière, on a cité les commandes les plus utilisées et parlé sur la Classification des machines CNC, Nous pouvons ainsi entamer la prochaine étape qui consiste à présenter la phase de conception.

## II.1. Introduction

Le développement majeur dans la technologie informatique nous permis la fabrication d'une machine CNC à faible coût et sa grâce a la disponibilité du matériels source, tels que les microcontrôleurs Arduino et les cartes de puissance. Donc le temps de prototypage et de développement sont considérablement réduits. Du point de vue l'utilisant d'un système de commande basé sur Arduino facilite le développement d'un prototype CNC à 3 axes

Le travail est reparti en trois parties comme suit :

La partie électronique, la partie mécanique et la partie logiciel et programmation

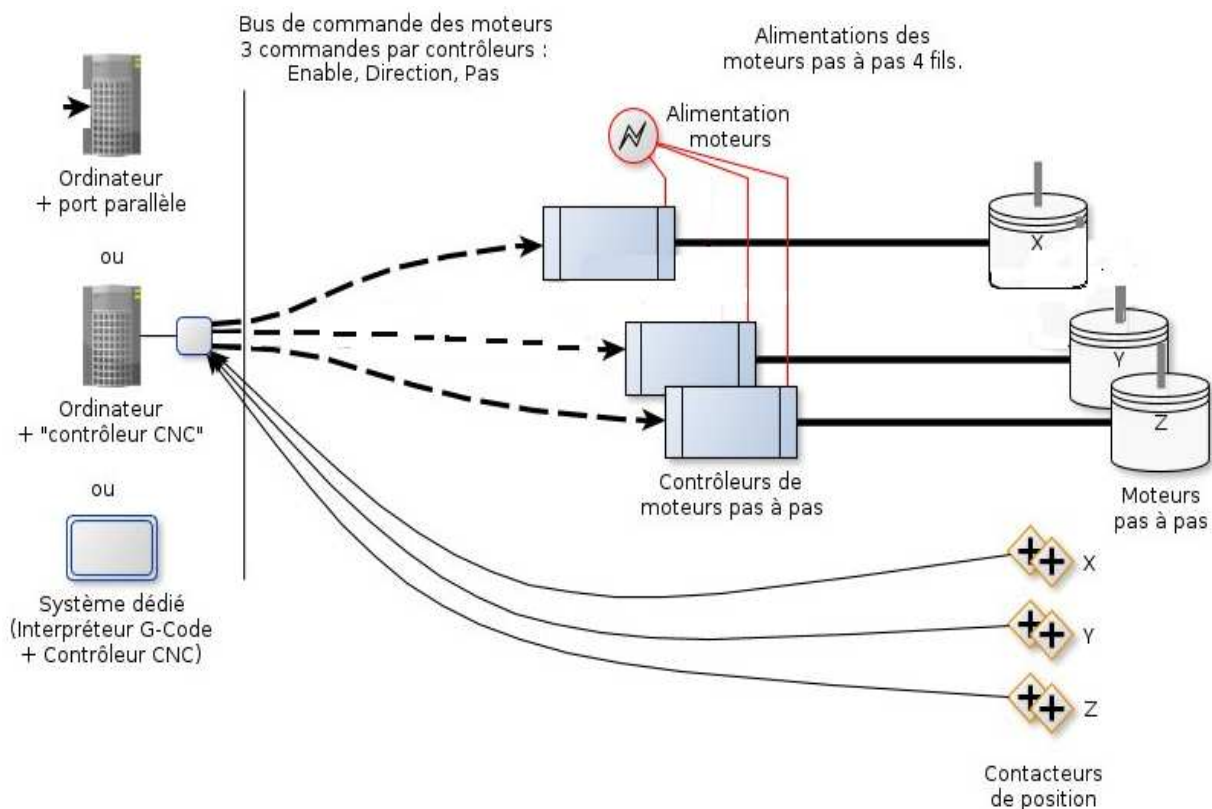


Figure II.1.: Schéma théorique d'une CNC 3 axes

## II.2. Cahier de charge:

Notre projet consiste à concevoir (conception et réalisation) d'une machine à commande numérique (mini fraiseuse 3 axes). Le projet s'inscrit dans le cadre d'un mémoire de fin d'étude

Le cahier des charges décrit les spécifications du point de vue technique, économique et opérationnel. Il exprime les exigences relatives au produit.

### II.2.1. Fonction principale :

La fonction essentielle de notre machine à commande numérique par ordinateur .

Cette machine a été réalisée pour travailler sur des matériaux tendres comme le bois, les matières plastiques, l'aluminium. En aucun cas, elle ne peut usiner des matériaux durs comme l'acier.

### II.2.2. Fonction partielles :

Il y a quatre fonctions partielles :

- ✓ Guidage des axes (X, Y, Z).
- ✓ Variation des coordonnées suivant les trois axes.
- ✓ Système de fixation des pièces.
- ✓ Partie commande.

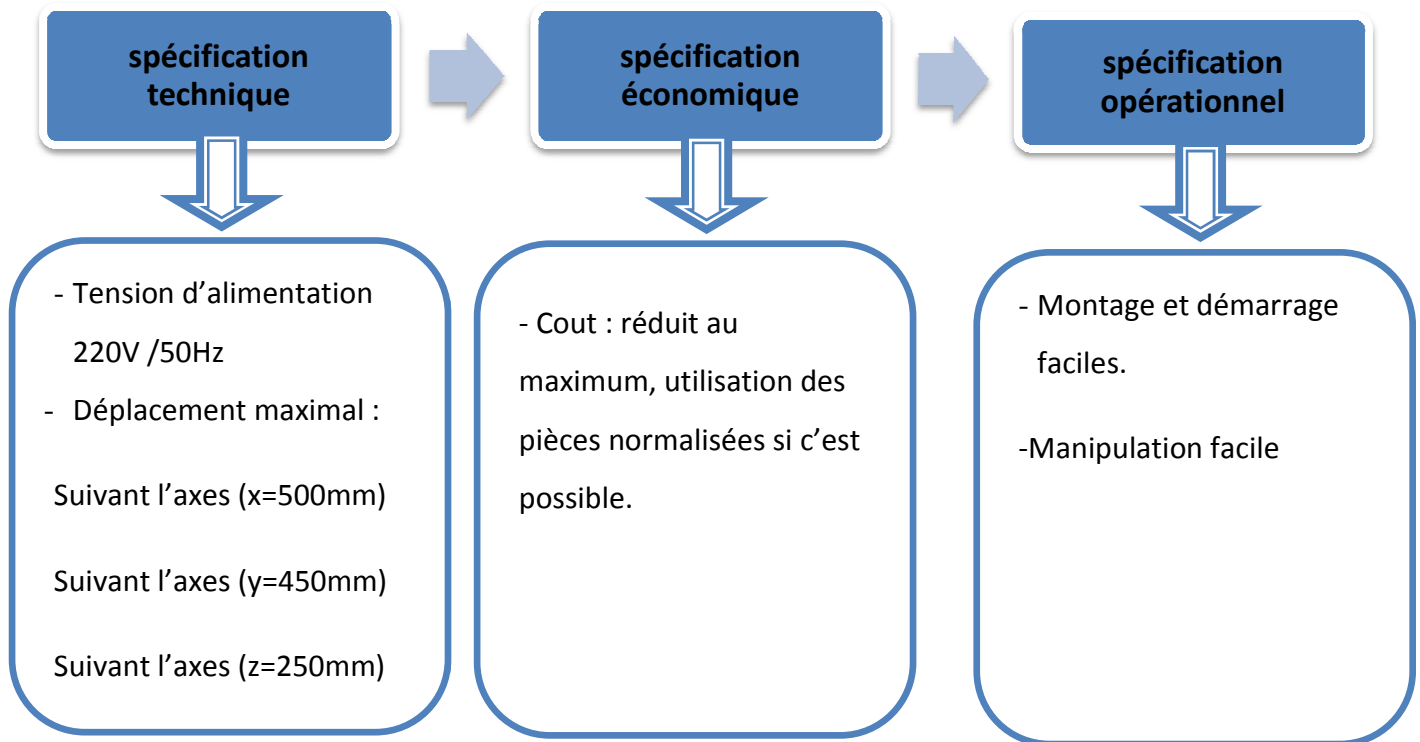


Figure.II. les spécification de la machine

### II.3. Variantes solutions proposées :

L'organigramme suivant récapitule les différentes solutions proposées afin d'accomplir les différentes fonctions du dispositif.

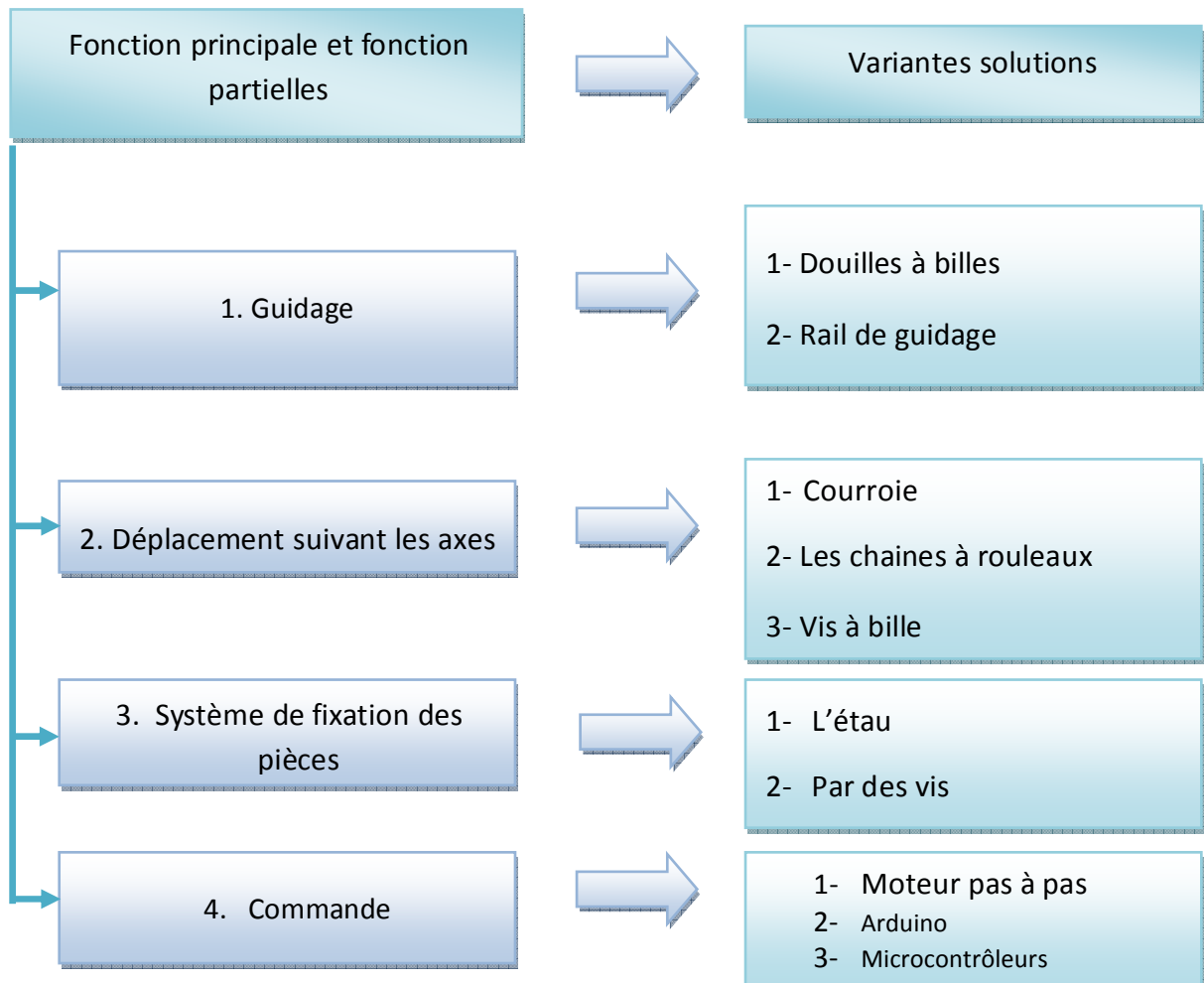


Figure II.2. : Les solutions proposé

### II. Partie mécanique :

Cette partie est pour but de dimensionner les différents éléments fonctionnels de notre dispositif pour assurer le bon fonctionnement de ce dernier.

## II.4. System de guidage :

**II.4.1. Douilles à billes :** Il s'agit de roulements à billes conçus pour se déplacer le long d'une tige lisse.

C'est à dire qu'ils ne tournent pas mais se déplacent linéairement (Guidage en translation).

Permettent des fonctionnements sans jeux, améliorent la précision et les performances. Valeur du coefficient de frottement de 0,001 à 0,005.

Utilisées sur les machines-outils, robots, systèmes automatisés...

Vitesse de déplacement 5 m/s.

Economiques pour arbres lisses, ne supportent que des charges radiales.

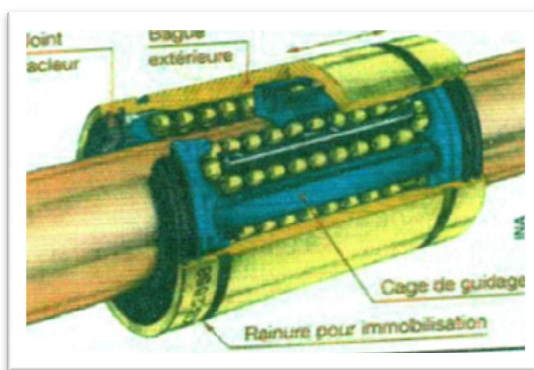


Figure II .3: **Douilles à billes.**

Par contre, on associe toujours cette douille avec une butée angulaire qui peut être réalisée par une autre douille sur une colonne parallèle.

C'est l'ensemble du montage qui les empêche de tourner sur leur axe. [...]

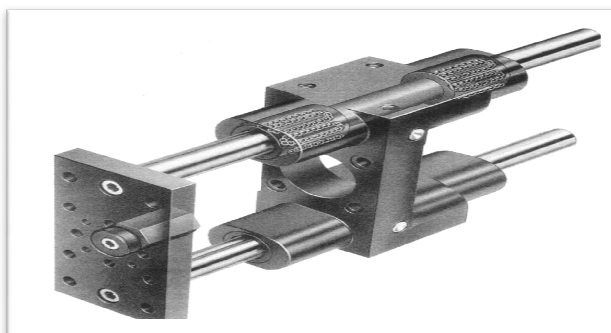


Figure II .4: **Module de guidage parallèle.**

### II.4.2. Rail de guidage pour guidage prismatique :

Les guidages de type prismatique associent des surfaces de contact planes. Guidages linéaires sur patins utilisés dans la mécanique de précision (automation, dispositifs de contrôle et de mesure...)

Ils permettent une absence totale de jeu et ils possèdent un très faible coefficient de frottement (0,0005 à 0,003).

Vitesse de déplacement de 3 à 5 m/s

Les systèmes linéaires sont généralement équipés de 2 ou 3 pistes de roulements assurant la mise en position et de pistes supplémentaires généralement réglables assurant le maintien en position.

Une recirculation des billes est souvent prévue.

## II.5. Déplacement suivant les axes x y z :

### II.5.1. Courroie :

Les courroies et les poulies permettent la transmission d'un mouvement de rotation d'un arbre menant à un arbre mené relativement éloignés l'un de l'autre. Le glissement qui résulterait du fonctionnement d'une transmission par friction exclut les courroies plates



Figure II .5 : Courroie

## II.6. Dimensionnement :

Les éléments fonctionnels de la mini fraiseuse à dimensionner sont les suivant :

- Les vis trapézoïdale,
- Les roulements,
- Les douilles à billes,
- Les axes de guidages,
- Accouplements semi-élastiques.

### II.6.1 Détermination des masses :

Avant de faire l'étude, on doit d'abord calculer la masse de chaque élément de notre machine (figure III.1). Nous avons utilisé deux types de pièces, normalisées et des pièces qui seront usinées. Les masses des pièces normalisées sont tirées d'après les catalogues des fournisseurs. Les masses des pièces à usinées sont calculées approximativement par le logiciel de conception Solid Works 2012. Le tableau III.1 représente les masses des pièces de notre dispositif :

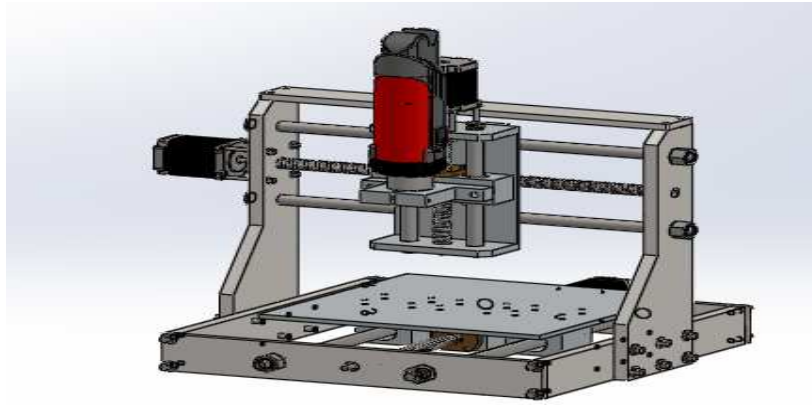


Figure II.6: dispositif 3D

déplacements	Nom de pièce	$m_i$	Masse (kg)
<b>Direction Z</b>	Mini perceuse	$m_1$	0.550
	Paliers à douilles à billes	$m_2$	0.496
	Porte outil	$m_3$	0.286
	Vis trapézoïdale + écrou	$m_4$	0.486
	Support porte outil	$m_5$	0.730
	Axes de guidages	$m_6$	0.7
	Moteur	$m_7$	0.35
<b>Direction Y</b>	Paliers à douilles à billes	$m_8$	1.124
	Vis trapézoïdale + écrou	$m_9$	0.726
	Axes de guidages	$m_{10}$	1.518
	Moteur	$m_{11}$	0.35
<b>Direction X</b>	Paliers à douilles à billes	$m_{12}$	1.435
	Vis trapézoïdale + écrou	$m_{13}$	0.726
	Axes de guidages	$m_{14}$	1.462
	Le plateau	$m_{15}$	1.696
	Moteur	$m_{16}$	0.35

Tableau III.1: les masses des pièces utilisées dans la conception

## II.6.2 Dimensionnement des vis trapézoïdale :

### Le rendement des vis et des écrous

$$\eta = (1 - f \cdot \tan \alpha) \div (1 + f / \tan \alpha)$$

Avec  $f = 0.18$  et  $\alpha = 5.2$

$f$  = coefficient de sécurité

$\alpha$  : L'angle de l'hélice

On trouve

$$\eta = 0.33$$

### 1) Dimensionnement de la vis trapézoïdale Z :

Où:  $F$  est la charge appliquée

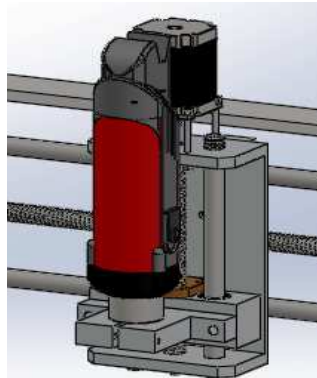


Figure II.7: système de mouvement Z

#### a. La charge appliquée sur la vis

$F$  la charge appliquée sur la vis trapézoïdale de l'axe Z est calculé par la formule suivante:

$$F_z = (m \times g) + (m \times \gamma) + F_p$$

Ou  $g$  : la gravité

$\gamma$  : L'accélération

$F_p$  : l'effort de pénétration

$m$  : l'ensemble des masse sur la vice

$$F_p = 1/3 F_c$$

$$F_c = a \cdot f_a \cdot K_c$$

$F_c$  : l'effort de couple

$K_c$  : coefficient de spécifique de coupe

$a$  : profondeur de passe en mm

$f_a$  : avance en mm/tr

$$K_c = 80 \text{ daN/mm}^2 = 800 \text{ N/mm}^2$$

$$a = 0.3 \text{ mm}$$

$$f_a = 0.4 \text{ mm}$$

d'où

$$F_c = 0.3 * 800 * 0.4$$

$$F_c = 96 \text{ N}$$

$$F_p = 1/3 F_c = 96/3$$

$$F_p = 32 \text{ N}$$

$$\alpha = \Delta v / \Delta t$$

On suppose que

$$\Delta t = 0,1 \text{ s}$$

et

$$v = 1000 \text{ tr / min}$$

$$v = 66,66 \text{ mm / s}$$

$$v = 0,06 \text{ m / s}$$

Donc

$$\alpha = 0,06 / 0,1$$

$$\alpha = 0,6 \text{ m / s}^2$$

$$m = m_1 + m_2 + m_3 = 1,332 \text{ Kg}$$

Avec:  $(m_i)$  sont l'ensemble des masses sur la vis et  $(g)$  est la gravitation. Après le calcul on trouve :

$$F_z = 46,12 \text{ N}$$

Pour des raisons de sécurité la force choisie doit être supérieure à la force calculée, donc on prend  $F = 50 \text{ N}$

### b. Le couple nécessaire

Il est donné par la formule suivante :

$$C_z = (F_z * p) / (2 * \eta * \pi * 1000)$$

$p$  = pas de la vis

$$p = 4 \text{ mm}$$

$$C_z = (46,12 * 4) / (2 * 3,14 * 0,33 * 1000)$$

$$C_z = 0,089 \text{ N.m}$$

### c. La puissance nécessaire

$$P_z = C * \omega$$

$$\omega = (2 * \pi * v) / 60$$

$$\omega = (2 * 3,14 * 1000) / 60$$

$$\omega = 104,66 \text{ rd / s}$$

$$P_z = 0,089 * 104,66$$

$$= 9,31 \text{ watt}$$

## 2) Dimensionnement de la vis trapézoïdale Y :

Où:  $F$  est la charge appliquée

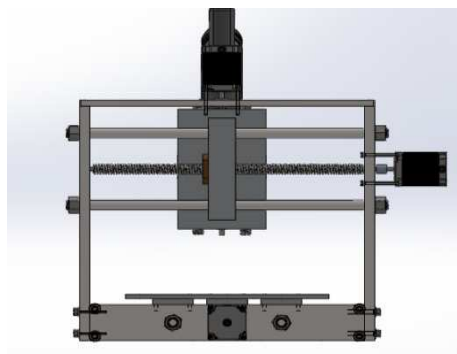


Figure II.8: système de mouvement Y

**a. La charge appliquée sur la vis**

**F** la charge appliquée sur la vis trapézoïdale de l'axe **Y** est calculé par la formule suivante:

$$F_y = (m \times g) + (m \times \gamma) + F_c$$

Ou  $g$  : la gravité

$\gamma$  : L'accélération

$m$  : l'ensemble des masse sur la vice

$F_c$  : l'effort de couple

$$F_c = a * f_a * K_c$$

$K_c$  : coefficient de spécifique de coupe

$a$  : profondeur de passe en mm

$f_a$  : avance en mm/tr

$$K_c = 80 \text{ daN/mm}^2 = 800 \text{ N/mm}^2$$

$$a = 0.3 \text{ mm}$$

$$f_a = 0.4 \text{ mm}$$

d'où

$$F_c = 0.3 * 800 * 0.4$$

$$F_c = 96 \text{ N}$$

$$\alpha = \Delta v / \Delta t$$

On suppose que

$$\Delta t = 0,1 \text{ s}$$

et

$$v = 1000 \text{ tr / min}$$

$$v = 66,66 \text{ mm / s}$$

$$v = 0,06 \text{ m / s}$$

Donc

$$\alpha = 0,06 / 0,1$$

$$\alpha = 0,6m / s^2$$

$$m = m_1 + m_2 + m_3 + m_4 + m_5 + m_6 + m_7 + m_8 = 4,722 \text{ Kg}$$

Avec:  $(m_i)$  sont l'ensemble des masses sur la vis et  $(g)$  est la gravitation. Après le calcul on trouve :

$$F_y = 146,58 \text{ N}$$

Pour des raisons de sécurité la force choisie doit être supérieur à la force calculée, donc on prend  $F = 150 \text{ N}$

### b. Le couple nécessaire

Il est donné par la formule suivante :

$$C_y = (F_y * p) / (2 * \eta * \Pi * 1000)$$

$p$  = pas de la vis

$$p = 4 \text{ mm}$$

$$C_y = (146,58 * 4) / (2 * 3,14 * 0,33 * 1000)$$

$$C_y = 0,28 \text{ N.m}$$

### c. La puissance nécessaire

$$P_y = C * \omega$$

$$\omega = (2 * \Pi * \nu) / 60$$

$$\omega = (2 * 3,14 * 1000) / 60$$

$$\omega = 104,66 \text{ rd / s}$$

$$P_y = 0,28 * 104,66$$

$$P_y = 29,30 \text{ watt}$$

### 3) Dimensionnement de la vis trapézoïdale X :

Où: F est la charge appliquée

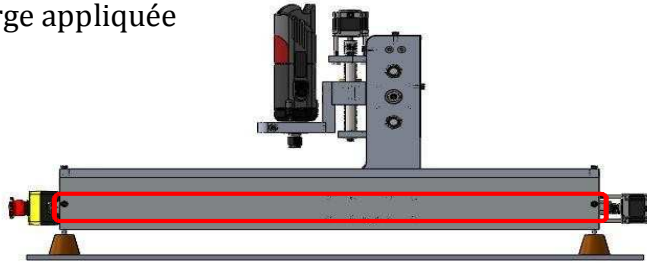


Figure II.9: système de mouvement X

#### a. La charge appliquée sur la vis

F la charge appliquée sur la vis trapézoïdale de l'axe X est calculé par la formule suivante:

$$F_x = (m \times g) + (m \times \gamma) + F_c$$

Ou g : la gravité

$\gamma$  : L'accélération

m : l'ensemble des masse sur la vice

$F_c$  : l'effort de couple

$$F_c = a \cdot f_a \cdot K_c$$

$K_c$  : coefficient de spécifique de coupe

a : profondeur de passe en mm

$f_a$  : avance en mm/tr

$$K_c = 80 \text{ daN/mm}^2 = 800 \text{ N/mm}^2$$

$$a = 0.3 \text{ mm}$$

$$f_a = 0.4 \text{ mm}$$

d'où

$$F_c = 0.3 \cdot 800 \cdot 0.4$$

$$F_c = 96 \text{ N}$$

$$\alpha = \Delta v / \Delta t$$

On suppose que

$$\Delta t = 0,1 \text{ s}$$

et

$$v = 1000 \text{ tr} / \text{min}$$

$$v = 66,66 \text{ mm} / \text{s}$$

$$v = 0,06 \text{ m} / \text{s}$$

Donc

$$\alpha = 0,06 / 0,1$$

$$\alpha = 0,6 \text{ m} / \text{s}^2$$

$$m = m_{12} + m_{15} = 3,131 \text{ Kg}$$

Avec:  $(m_i)$  sont l'ensemble des masses sur la vis et  $(g)$  est la gravitation. Après le calcul on trouve :

$$F_x = 129,19 \text{ N}$$

Pour des raisons de sécurité la force choisie doit être supérieure à la force calculée, donc on prend  $F = 130 \text{ N}$

### b. Le couple nécessaire

Il est donné par la formule suivante :

$$C_x = (F_y \cdot p) / (2 \cdot \eta \cdot \Pi \cdot 1000)$$

$p$  = pas de la vis

$$p = 4 \text{ mm}$$

$$C_x = (129,19 \cdot 4) / (2 \cdot 3,14 \cdot 0,33 \cdot 1000)$$

$$C_x = 0,25 \text{ N.m}$$

### c. La puissance nécessaire

$$P_x = C \cdot \omega$$

$$\omega = (2 \cdot \Pi \cdot v) / 60$$

$$\omega = (2 * 3,14 * 1000) / 60$$

$$\omega = 104,66 \text{rd} / \text{s}$$

$$P_x = 0,28 * 104,66$$

$$P_x = 26,16 \text{ watt}$$

## II. La partie électronique

Cette partie regroupe plusieurs composants électroniques qui apparaissent dans le schéma synoptique suivant.

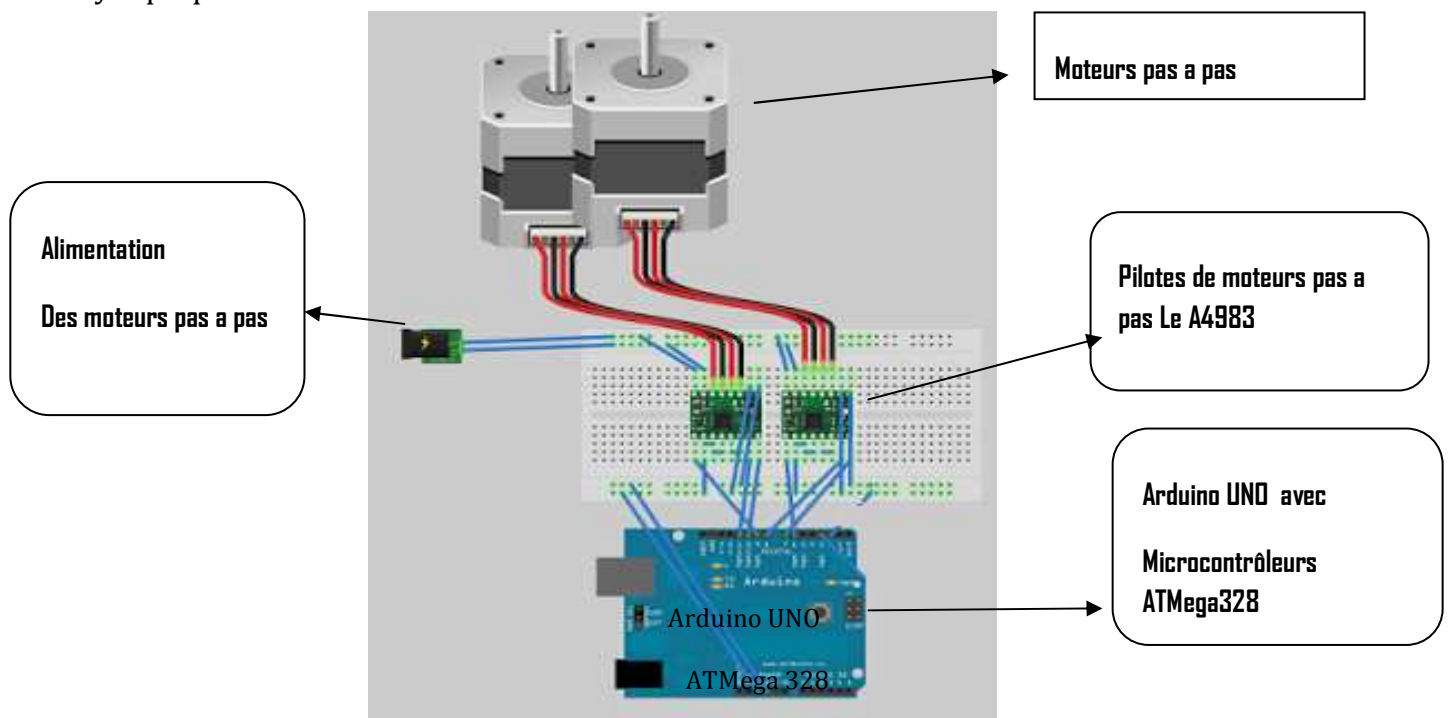


Figure.II. la partie électronique

### II.7. Les Actionneurs

#### II.7.1. Généralité sur les moteurs pas à pas [2]

Le moteur pas à pas constitue un convertisseur électromécanique destiné à transformer le signal électrique (impulsion ou train d'impulsions de pilotage) 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 (Figure 1)

- 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, fournit l'énergie vers les enroulements appropriés du moteur.

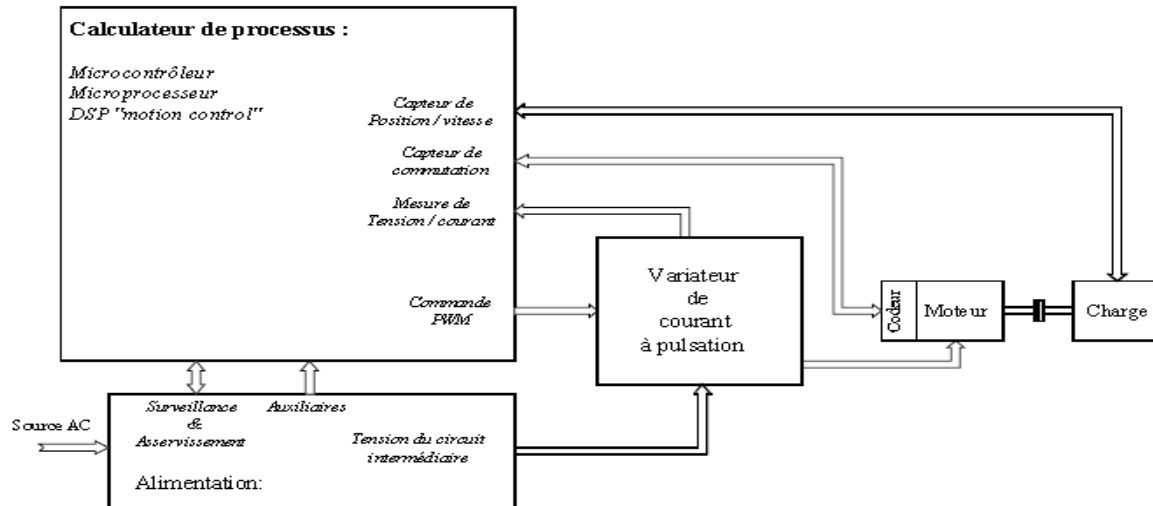


Figure II.10.: **Schéma bloc de principe du contrôle d'un moteur pas-à-pas**

L'alimentation de chaque bobinage du moteur par une tension particulière provoque l'apparition d'un courant qui engendre un champ magnétique de direction précise. Le changement séquentiel des tensions particulières à chaque bobinage permet de déplacer la position du champ statorique selon une résolution élémentaire appelée pas. Autrement dit, toutes configurations des tensions aux bornes des bobinages correspondent à un déplacement de la position stable du rotor. Une série bien déterminée de commutation de tensions entraîne un déplacement sur un nombre correspondant de pas. La succession des configurations d'alimentation, à une fréquence donnée, impose un champ statorique tournant avec une résolution d'un micro pas, d'un demi pas ou d'un pas entier.

Le mouvement discret du champ d'entrefer est suivi par le rotor soumis au couple synchronisant. La nature du couple découle soit :

- d'interaction champ du stator- fer du rotor, lorsque ce dernier présente une structure dentée passive.
- d'interaction champ du stator- champ du rotor, lorsque le rotor comporte un aimant.
- les deux interactions précédentes simultanément.

Les moteurs couramment rencontrés selon le nombre de pas par tour sont :

0,9° soit 400 pas par tour

1,8° soit 200 pas par tour

15° soit 24 pas par tour

Les moteurs pas à pas présentent une très grande précision et une durée de vie quasi illimitée, l'usure mécanique étant pratiquement inexistante et ils existent en différentes tailles qui varient entre 1cm et plus d'une dizaine de centimètre. Tout dépendra des applications dans ils seront utilisées.

### II.7.2. Les différents types de moteurs pas à pas [3]

On peut classer les moteurs pas à pas en 3 catégories :

- les moteurs à aimant permanent
- les moteurs à reluctance variable
- les moteurs hybrides

#### a) Moteurs à reluctance variable

Les moteurs à reluctance variable (moteurs MRV) doivent leur nom au fait que le circuit magnétique qui les compose s'oppose de façon variable à sa pénétration par un champ magnétique.

Ces moteurs sont composés d'un barreau de fer doux et d'un certain nombre de bobines. Lorsqu'on alimente une bobine, elle devient un électroaimant et le barreau de fer cherche naturellement à s'orienter suivant le champ magnétique. On alimente la phase 1, puis la phase 2, puis la phase 3 ... Si on veut changer le sens du moteur, il suffit de changer l'ordre d'alimentation des bobines.

Dans la pratique, le barreau de ferrite a plusieurs dents (dans notre exemple est 6). Dès qu'on alimente la phase 2, il y a une rotation de  $15^\circ$  (i.e.  $60^\circ - 45^\circ = 15^\circ$ ), puis la phase 3, etc. Donc le moteur tourne de  $15^\circ$  dès qu'on alimente une phase.

Il faut 24 impulsions pour faire un tour complet. C'est un moteur 24 pas.

Comme le montre le figure suivante :

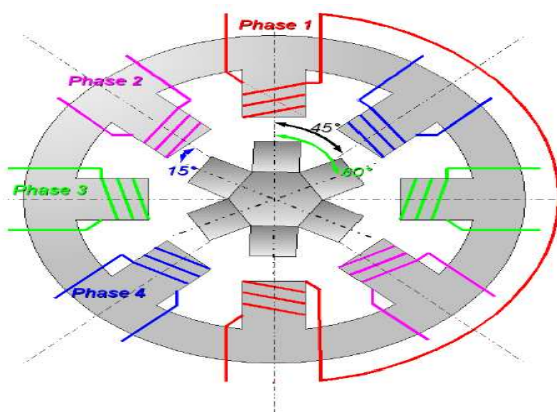


Figure II.11: moteur a reluctance variable

Inconvénients : nécessite au moins trois bobinages, pour obtenir un cycle complet, pas de couple résiduel, c'est-à-dire que hors tension, le rotor est libre, ce qui peut être problématique pour ce genre de moteur. La fabrication est assez délicate, les entrefers doivent être très faibles.

Avantages du système : peu coûteux, d'une bonne précision. Dans l'exemple, avec

seulement 4 enroulements, on obtient 24 pas (on peut facilement obtenir 360 pas). Le sens du courant dans la bobine n'a aucune importance.

### b) Moteur à aimants permanents

Les moteurs à aimants permanents sont semblables aux moteurs à réluctance variable, sauf que le rotor possède des pôles NORD et SUD. À cause des aimants permanents, le rotor reste freiné à sa dernière position lorsque le bloc d'alimentation cesse de fournir des impulsions.

Une façon simple de voir le système, est de placer une boussole entre deux aimants. Suivant la bobine qui est alimentée et le sens du courant, l'aimant va s'aligner avec le champ.

Il utilise le principe de l'action d'un champ magnétique sur un moment magnétique (aimant).

On distingue deux (2) types de ces moteurs :

#### 2.1 Moteur unipolaire

Le rotor est constitué par un aimant permanent (en ferrite par exemple) comportant une paire de pôles. Le stator, comprend deux circuits magnétiques décalés de 90°. Les enroulements à point milieu sont alimentés avec une polarité toujours du même signe, d'où le nom d'unipolaire.

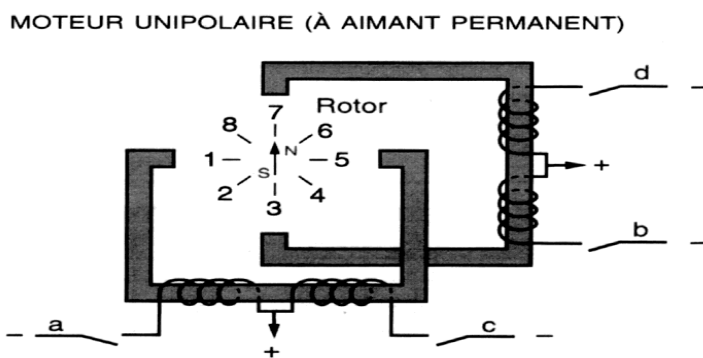


Figure II.12: Moteur unipolaire

Pour inverser le sens de rotation il suffit d'inverser les séquences de commutation.

#### 2.2 Moteur bipolaire

Les enroulements au stator n'ont pas de points milieu

Chaque borne de chaque enroulement est alimentée successivement par une polarité positive puis négative d'où le terme bipolaire.

En inversant les polarités des enroulements statoriques, on inverse les pôles nord et sud au stator.

Le nombre de phases est égale au nombre d'enroulement

Moteur 2 phases, 2 pôles au rotor, alimentation bipolaire.

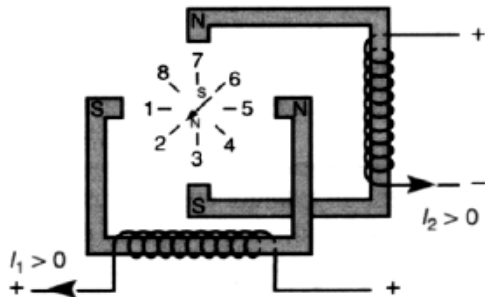


Figure II.13. : **Moteur bipolaire**

Le sens de rotation dépend du sens du courant et de l'ordre d'alimentation du bobinage

*Caractéristiques principales*

- Faible résolution : nombre de pas / tour peu important ;
- Couple d'utilisation plus élevé par rapport au moteur à reluctance variable ;
- Présence d'un couple résiduel lorsque le moteur est hors tension.

### c) Moteurhybride

C'est un moteur reluctant polarisé, il superpose le principe de fonctionnement des moteurs à aimant permanent et à reluctance variable et combine leurs avantages.

Le rotor est constitué de deux(2) disques dentés décalés mécaniquement. Entre ces deux(2) disques, est inséré un aimant permanent.

Deux modes de fonctionnement peuvent être envisagés selon la valeur de temps T :

*Caractéristiques principales*

- hors tension, le rotor est maintenu en position
- bonne précision de la position du rotor
- grande vitesse de rotation

### II.7.3. Comparaison des 3 types de moteurs

<b>Comparaison des performances des 3 types de moteurs pas à pas</b>			
<b>Types de moteurs</b>	Moteur à aimant permanent	Moteur à reluctance variable	Moteur hybride
<b>Résolution (nbr de pas/tour)</b>	Moyenne	Bonne	Elevé
<b>Couple moteur</b>	Elevé	faible	Elevé
<b>Sens de rotation</b>	Il dépend :-du sens du courant pour le moteur bipolaire -de l'ordre d'alimentation des bobines	Il dépend : uniquement de l'ordre d'alimentation des bobines	Il dépend : -du sens du courant - de l'ordre d'alimentation des bobines
<b>Fréquence de travail</b>	Faible	Grande	Grande

### II.7.4. Configuration interne des bobines des moteurs

#### Les moteurs à 5 fils

Le moteur comporte deux (2) bobines à point centrale, on relie le point centrale à l'alimentation et les autres fils à l'interface de puissance.

#### Les moteurs à 6 fils

Avec le moteur à 6 fils, on a le choix d'une commande bipolaire en ignorant simplement les connexions centrales, ou une commande unipolaire en reliant les points centraux au (+) de l'alimentation.

#### Les moteurs à 8 fils

Avec celui-ci aussi on a le choix des deux commandes bipolaire ou unipolaire. Dans le premier cas on ignore les 4 fils centraux, et dans le deuxième cas on relie les 4 fils ensemble.

#### Les moteurs à 4 fils

Ce moteur est obligé d'être alimenté soit une bobine à la fois soit les deux au même temps. Donc à tout moment le moteur est à la moitié ou à la totalité de ces bobines alimentées, ce qu'il a comme avantage de lui donner plus de force.

## II.8. Les pilotes des moteurs pas à pas [4]

Les moteurs pas à pas peuvent être commandés soit en utilisant une alimentation en tension constante qui est très utilisée pour les faibles vitesses mais dissipe une puissance assez grande, soit une alimentation en courant constant qui est recommandée pour la commande des moteurs bipolaire et permet d'obtenir un grand couple.

Notre choix s'est porté sur l'utilisation d'une alimentation à courant constant pour minimiser la puissance et avoir un grand couple.

La commande d'un moteur pas à pas est assurée par plusieurs pilotes ou « drivers » de moteurs pas à pas et parmi ces drivers, on trouve le A4983 celui qu'on a choisi pour notre conception.

### II.8.1. Commande des moteurs pas à pas avec A4983 [5]

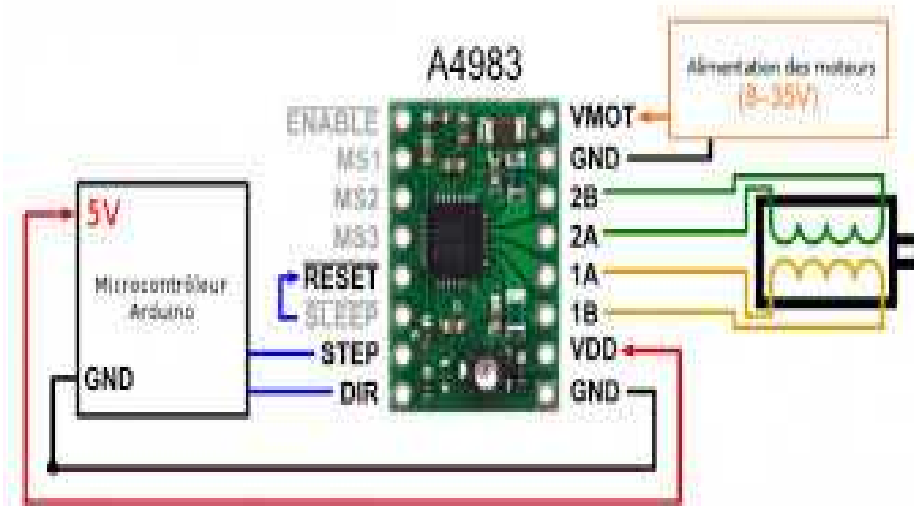
Le support de commande de moteur pas à pas A4983 est une carte de platine pour Allegro facile à utiliser.

Le conducteur dispose de courant réglable limitant et cinq résolutions différentes de micro-pas. Il fonctionne 8-35 V et peut fournir jusqu'à 2 A par bobine.

Il permet de contrôler un moteur pas à pas bipolaire jusqu'à 2 A comme courant de sortie par bobine

Voici quelques-unes des principales caractéristiques du conducteur:

- 1) Étape simple et l'interface de commande de direction
- 2) Cinq résolutions de pas différentes: pas entier, demi-pas, quart-pas, huitième-pas, et seizième-pas
- 3) Contrôle de courant réglable vous permet de définir le courant de sortie maximale avec un potentiomètre, ce qui vous permet d'utiliser des tensions supérieures à la tension nominale de votre moteur pas à pas pour obtenir des taux plus élevés d'étape.
- 4) Le contrôle de découpage intelligent qui sélectionne automatiquement le mode de décroissance du courant.
- 5) Surchauffe arrêt thermique et la protection de croisement courant.



**a) Les connexions d'alimentation**

Le pilote nécessite une tension d'alimentation logique (3 à 5,5 V) connectée entre les broches VDD et GND et une tension d'alimentation du moteur de (8-35 V) devant être connectée aux bornes VMOT et GND. Ces fournitures devraient avoir des condensateurs de découplage appropriés à proximité de la carte, et ils devraient être capables de fournir les courants attendus (pics jusqu'à 4 A pour l'alimentation du moteur).

Les entrées de sélection (MS1, MS2, MS3) permettent la sélection des cinq étapes de résolutions selon le tableau ci-dessous. MS2 et MS3 ont 100kΩ de résistance interne, mais MS1 n'a pas, donc il doit être connecté à l'extérieur. Pour que les modes de micro pas fonctionnent correctement, la limite de courant doit être réglée assez bas de sorte que la limitation de courant se fiance. Sinon, les niveaux actuels intermédiaires ne seront pas correctement entretenus, et les moteurs fonctionneront effectivement dans un mode pas entier.

MS1	MS2	MS3	Résolutions de pas
BAS	BAS	BAS	Pas entiers
HAUT	BAS	BAS	Demi-pas
BAS	HAUT	BAS	Quart de pas
HAUT	HAUT	BAS	Huitième de pas
HAUT	HAUT	HAUT	Seizième de pas

**b) Les entrées de commande**

Chaque impulsion à l'entrée d'étape correspond à un micro-pas du moteur pas à pas dans la direction sélectionnée par l'axe DIR. Notons que les STEP et DIR broches ne sont pas tirés à toute tension particulière en interne, de sorte qu'on ne peut pas laisser une de ces broches flottantes

dans notre application. Si on a besoin juste d'une rotation dans une seule direction, on peut attacher la broche DIR directement à VCC ou GND.

**II.8.2. Commande des moteurs pas à pas avec L297 :**

Le L297 est un circuit qui contient toute la circuiterie de commande et de contrôle de moteurs pas à pas unipolaire et bipolaire.

Utilisé souvent avec un driver double pont tel que le L298, l'ensemble forme une interface idéale pour le contrôle d'un moteur pas à pas par microprocesseur ou microcontrôleur.

Il est possible de contrôler un moteur pas à pas avec un L297 en lui associant quatre transistors darlington ou mosfet

**a) Constitution du L297 :**

Le cœur du L297 est le block appelé "translator" sur le schéma-block. C'est lui qui génère les séquences appropriées pour l'alimentation des phases en mode demi-pas ou pas entier .

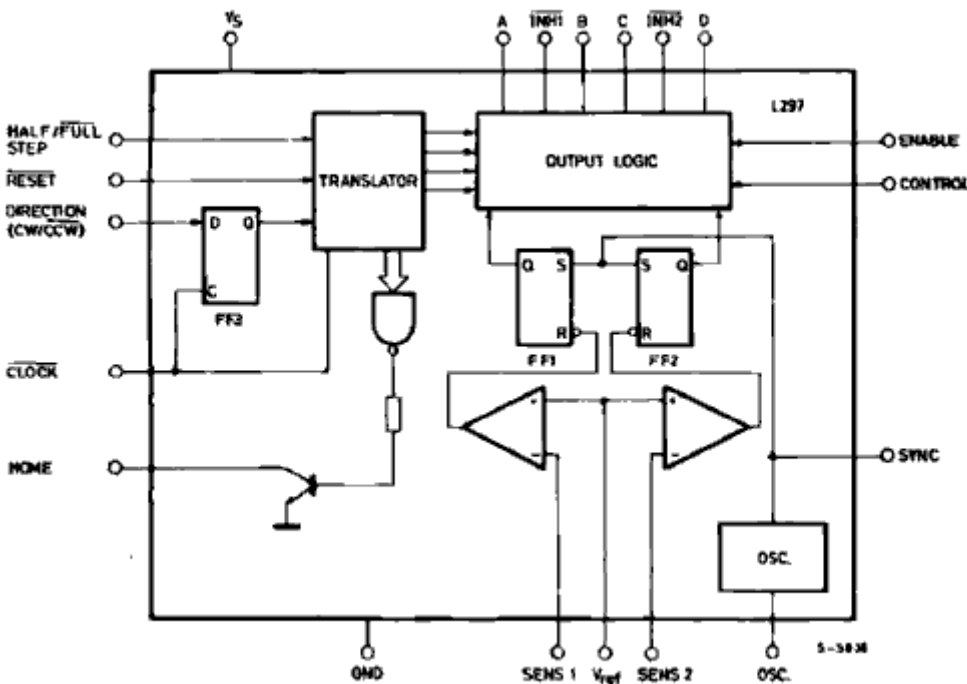


Figure.II schéma blok du L297

Ce block est contrôlé par deux entrées de mode, le mode direction (CW / CCW) et le mode pas / demi-pas (HALF/ FULL) et l'entrée clock dont les impulsions d'horloge qu'on lui applique font évoluer le séquenceur vers le prochain pas.

Quatre sorties sont fournies par le séquenceur et attaquent le block logique qui contient les fonctions "inhibe" et "chopper".

La structure interne du séquenceur consiste en un compteur 3 bits (plus une logique combinatoire) qui génère les huit étapes de base (en code gray) comme nous le voyons sur la figure suivante.

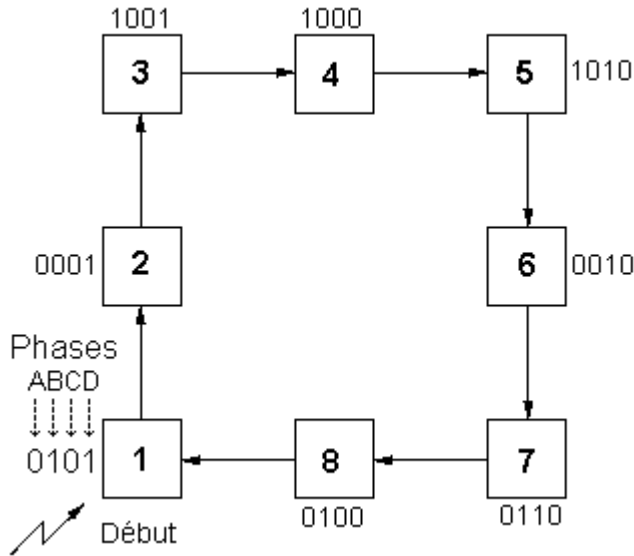
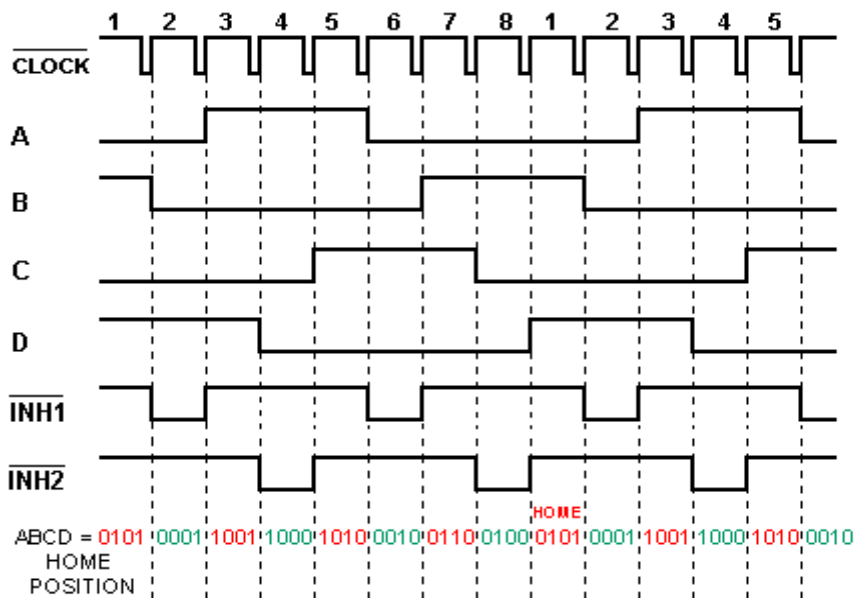


Figure.II. schéma de code gray

Sur ce diagramme nous sommes en mode demi-pas (Half) et en sens horaire (CW). Les trois séquences de commande possibles sont extraites directement de cette séquence principale.

Le chronogramme des sorties en mode demi pas est visible ci-après.



Nous remarquons deux signaux supplémentaires, INH1 et INH2, générés dans cette séquence. Leurs usages sont expliqués un peu plus en avant dans le texte.

Le circuit de commande de moteur pas à pas L297 est initialement prévu pour être utilisé avec un driver L298 ou L293E.

Il reçoit les signaux de contrôle en provenance d'un système à base de microprocesseur et génère tous les signaux nécessaires pour l'étage de puissance.

Ce circuit inclut deux régulations à découpage type PWM (modulation de largeur d'impulsion) pour réguler le courant des enroulements moteur.

Avec un étage de puissance adapté le L297 commande un moteurs pas à pas bipolaire à aimant permanent, un moteur unipolaire à quatre phases à aimant permanent ou un moteur à reluctance variable à quatre phases.

De plus il est capable de gérer le mode demi pas et pas complet. Le L297A est une version spéciale pour le positionnement des têtes de lecteur de disquette.

Le circuit intégré L298N est un étage de puissance (driver) pour les moteurs à courant continu. Toutefois, le L297 est utilisable avec toute sorte d'étage de puissance et peut s'affranchir la commande des composants discrets tel que des transistors de puissance ; il fournit jusqu'à 20 mA sur ces sorties.

Le schéma suivant représente une configuration typique du circuit de commande de moteur pas à pas L297 et le double pont L298 pour former une interface avec microprocesseur.

## II.9. Les capteurs [ 6]

### II.9.1 Définition des capteurs

Un capteur est un organe de prélèvement d'information qui élabore à partir d'une grandeur physique, une autre grandeur physique de nature différente (très souvent électrique). Cette grandeur représentative de la grandeur prélevée est utilisable à des fins de mesure ou de commande.

### II.9.2 Les différents types de capteurs

#### a) Capteur actif

Fonctionnant en générateur, un capteur actif est généralement fondé dans son principe sur un effet physique qui assure la conversion en énergie électrique de la forme d'énergie propre à la grandeur physique à prélever, énergie thermique, mécanique ou de rayonnement. Les plus classiques sont :

- Effet thermoélectrique : Un circuit formé de deux conducteurs de nature chimique différente, dont les jonctions sont à des températures  $T_1$  et  $T_2$ , est le siège d'une force électromotrice  $e(T_1, T_2)$ .

- Effet piézo-électrique : L'application d'une contrainte mécanique à certains matériaux dits piézo-électrique (le quartz par exemple) entraîne l'apparition d'une déformation et d'une même

Charge électrique de signe différent sur les faces opposées.

- Effet d'induction électromagnétique : La variation du flux d'induction magnétique dans un circuit électrique induit une tension électrique.

- Effet photo-électrique : La libération de charges électriques dans la matière sous l'influence d'un rayonnement lumineux ou plus généralement d'une onde électromagnétique dont la longueur d'onde est inférieure à un seuil caractéristique du matériau.

- Effet Hall : Un champ B crée dans le matériau un champ électrique E dans une direction perpendiculaire.
- Effet photo voltaïque : Des électrons et des trous sont libérés au voisinage d'une jonction PN illuminée, leur déplacement modifie la tension a ses bornes.

### b) Capteur passif

Il s'agit généralement d'impédance dont l'un des paramètres déterminants est sensible à la grandeur mesurée. La variation d'impédance résulte :

- Soit d'une variation de dimension du capteur, c'est le principe de fonctionnement d'un grand nombre de capteurs de position, potentiomètre, inductance à noyaux mobile, condensateur à armature mobile.
- Soit d'une déformation résultant de force ou de grandeur s'y ramenant, pression accélération (armature de condensateur soumise à une différence de pression, jauge d'extensomètre liée a une structure déformable).

L'impédance d'un capteur passif et ses variations ne sont mesurables qu'en intégrant le capteur dans un circuit électrique, couplé avec une alimentation qui représente son conditionneur.

### II.9.3. Quelques exemples sur les capteurs

Les Capteurs de Lumière, les Capteurs de Force, les Capteurs de Son, les Capteurs de Vitesse, les Capteurs Météorologiques, et les Capteurs de Position.

Afin d'éviter les dépassements de la machine et avoir plus de sureté on va utiliser des contacteurs de fin de course, qui peuvent aussi être appelés "Déecteur de position" ou "Interrupteur de fin de course". Ils coupent ou établissent un circuit lorsqu'ils sont actionnés par un mobile.

La détection s'effectue par contact d'un objet extérieur sur le levier ou un galet. Ce capteur peut prendre alors deux états :

- Enfoncé (en logique positive l'interrupteur est fermé).
- Relâché (en logique positive l'interrupteur est ouvert).

## II.10. Les microcontrôleurs [7]

### II.10.1. Définition :

Ce sont des circuits programmables capable d'exécuter un programme et qui possède des circuits d'interface intégrés avec le monde extérieur.

Les microcontrôleurs sont apparus quand les technologies d'intégrations ont suffisamment progressées et quand les applications domestiques ou industrielles avaient besoin de systèmes « intelligents » ou tout au moins programmables.

Les microcontrôleurs se caractérisent par un plus haut degré d'intégration, une plus faible consommation électrique, une vitesse de fonctionnement plus faible (de quelques mégahertz

jusqu'à plus d'un gigahertz<sup>1</sup>) et un coût réduit par rapport aux microprocesseurs polyvalents utilisés dans les ordinateurs personnels.

Par rapport à des systèmes électroniques à base de microprocesseurs et autres composants séparés, les microcontrôleurs permettent de diminuer la taille, la consommation électrique et le coût des produits. Ils ont ainsi permis de démocratiser l'utilisation de l'informatique dans un grand nombre de produits et de procédés.

Les microcontrôleurs sont fréquemment utilisés dans les systèmes embarqués, comme les contrôleurs des moteurs automobiles, les télécommandes, les appareils de bureau, l'électroménager, les jouets, la téléphonie mobile, etc.

Les microcontrôleurs améliorent l'intégration et le coût (lié à la conception et à la réalisation) d'un système à base de microprocesseur en rassemblant ces éléments essentiels dans un seul circuit intégré. Un microcontrôleur est donc un composant autonome, capable d'exécuter le programme contenu dans sa mémoire morte dès qu'il est mis sous tension. Selon les modèles et les conditions de fonctionnement, les microcontrôleurs peuvent avoir besoin de quelques composants externes (quartz, quelques condensateurs, etc), mais ceci reste très limité.

### II.10.2. Composants intégré :

Un microcontrôleur se compose de :

- un processeur (CPU), avec une largeur du chemin de données allant de 4 bits pour les modèles les plus basiques à 32 ou 64 bits pour les modèles les plus évolués ;
- de la mémoire vive (RAM) pour stocker les données et variables ;
- de la mémoire morte (ROM) pour stocker le programme. Différentes technologies peuvent être employées : EPROM, EEPROM, mémoire flash (la plus récente) ;
- souvent un oscillateur pour le cadencement. Il peut être réalisé avec un quartz, un circuit RC ou encore une PLL ;
- des périphériques, capables d'effectuer des tâches spécifiques. On peut mentionner entre autres :
  - les convertisseurs analogiques-numériques (CAN) (donnent un nombre binaire à partir d'une tension électrique),
  - les convertisseurs numériques-analogiques (CNA) (effectuent l'opération inverse),
  - les générateurs de signaux à modulation de largeur d'impulsion (MLI, ou en anglais, PWM pour *Pulse Width Modulation*),
  - les timers/compteurs (compteurs d'impulsions d'horloge interne ou d'événements externes),
  - les chiens de garde (watchdog),
  - les comparateurs (comparent deux tensions électriques),
  - les contrôleurs de bus de communication (UART, I<sup>2</sup>C, SSP, CAN, FlexRay, USB, Ethernet, etc.).



Ces PIC intègrent des quantités de composants très différentes.

Le fonctionnement des périphériques peut être paramétré et commandé par le programme et/ou les entrées-sorties. Les périphériques peuvent générer une interruption qui contraint le processeur à quitter le programme en cours pour effectuer une routine de traitement de l'interruption, lorsque l'événement qui la déclenche survient.

Les microcontrôleurs peuvent généralement se placer dans un état de sommeil, dans lequel ils présentent une très faible consommation électrique. Un signal envoyé par l'un de leurs périphériques (timer, broche d'entrée-sortie, watchdog, etc.) permet de les faire sortir de cet état de sommeil.

Certains microcontrôleurs ont un nombre très restreint de broches, si bien qu'une broche donnée peut correspondre à plusieurs périphériques internes. La fonction choisie doit alors être sélectionnée par logiciel.

Le choix des périphériques à intégrer dans un microcontrôleur est délicat. Les fabricants doivent réaliser un compromis entre des besoins contradictoires : utiliser des fréquences élevées, réduire la taille du circuit, apporter des fonctionnalités nombreuses, élaborer une architecture flexible, assurer des coûts modérés, etc.

### II.10.3. Les avantages des microcontrôleurs

- Diminution de l'encombrement du matériel et du circuit imprimé
- Simplification du tracé du circuit imprimé (plus besoin de tracer de bus !)
- Augmentation de la fiabilité du système
  - nombre de composants ☒
  - connexions composants/supports et composant circuit imprimé ☒
- Intégration en technologie MOS, CMOS, ou HCMOS
  - diminution de la consommation
- Le microcontrôleur contribue à réduire les coûts à plusieurs niveaux:
  - moins cher que les composants qu'il remplace
  - Diminution des coûts de main d'oeuvre (conception et montage)
- Environnement de programmation et de simulation évolués

### II.10.4. Les défauts des microcontrôleurs

- le microcontrôleur est souvent surdimensionné devant les besoins de l'application
- Investissement dans les outils de développement

- Écrire les programmes, les tester et tester leur mise en place sur le matériel qui entoure le microcontrôleur
- Incompatibilité possible des outils de développement pour des microcontrôleurs de même marque.
- Les microcontrôleurs les plus intégrés et les moins coûteux sont ceux disposant de ROM programmables par masque

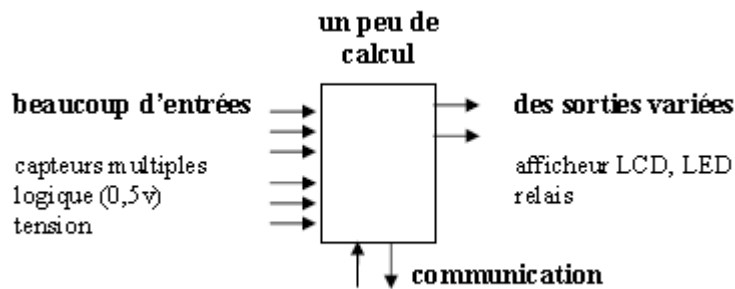
Fabrication uniquement en grande série >1000

Défaut relatif car il existe maintenant systématique des version OTPROM un peu plus chère.

### II.10.5. Familles de microcontrôleurs

- la famille Atmel AT91 ;
- la famille Atmel AVR (utilisée par des cartes Arduino) ;
- le C167 de Siemens/Infineon ;
- la famille Hitachi H8 ;
- la famille Intel 8051, qui ne cesse de grandir ; de plus, certains processeurs récents utilisent un cœur 8051, qui est complété par divers périphériques (ports d'E/S, compteurs/temporisateurs, convertisseurs A/N et N/A, chien de garde, superviseur de tension, etc.) ;
- l'Intel 8085, à l'origine conçu pour être un microprocesseur, a en pratique souvent été utilisé en tant que microcontrôleur ;
- le Freescale 68HC11 ;
- la famille Freescale 68HC08 ;
- la famille Freescale 68HC12 ;
- la famille FreescaleQorivva MPC5XXX ;
- la famille des PIC de Microchip ;
- la famille des dsPIC de Microchip ;
- la famille des ST6, ST7, STM8, ST10, STR7, STR9, STM32 de STMicroelectronics ;
- la famille ADuC d'AnalogDevices ;
- la famille PICBASIC de ComfileTechnology;
- la famille MSP430 de Texas Instruments ;
- la famille 8080, dont les héritiers sont le microprocesseur Zilog Z80 (désormais utilisé en tant que contrôleur dans l'embarqué) et le microcontrôleur Rabbit ;
- la famille PSoC de Cypress ;
- la famille LPC21xxARM7-TDMI de Philips ;
- la famille V800 de NEC ;
- la famille K0 de NEC.

**En conclusion :**



Le microcontrôleur présente l'avantage des  $\mu$ processeurs mais limités aux applications ne nécessitant pas trop de puissance de calcul (architecture courante 8bits)

Il existe plusieurs architecture de microcontrôleurs de 4 à 128 bits pour les applications demandant une certaine puissance de calcul (injecteurs automobile)

Dans notre conception on a utilisé un microcontrôleur de la famille ATmel AVR (utilisée par des cartes Arduino), il existe plusieurs microcontrôleurs de cette même famille parmi eux l'ATMega328 qui est utilisé pour les cartes Arduino UNO, qu'on a choisis pour notre conception et qu'on va présenter dans ce qui suit.

## II.11. Arduino UNO [8]

### II.11.1 Vue d'ensemble

Le modèle UNO de la société ARDUINO est une carte électronique dont le cœur est un microcontrôleur ATMEL de référence ATMega328. Le microcontrôleur ATMega328 est un microcontrôleur 8bits avec une vitesse d'horloge de 16Mhz de la famille AVR dont la programmation peut être réalisée en langage C. L'intérêt principal des cartes ARDUINO (d'autres modèles existant) est leur facilité de mise en œuvre. ARDUINO fournit un environnement de développement s'appuyant sur des outils open source. Le chargement du programme dans la mémoire du microcontrôleur se fait de façon très simple par port USB. En outre, des bibliothèques de fonctions "clé en main" sont également fournies pour l'exploitation d'entrées-sorties courantes : gestion des E/S TOR, gestion des convertisseurs ADC, génération de signaux PWM, exploitation de bus TWI/I2C, exploitation de servomoteurs...

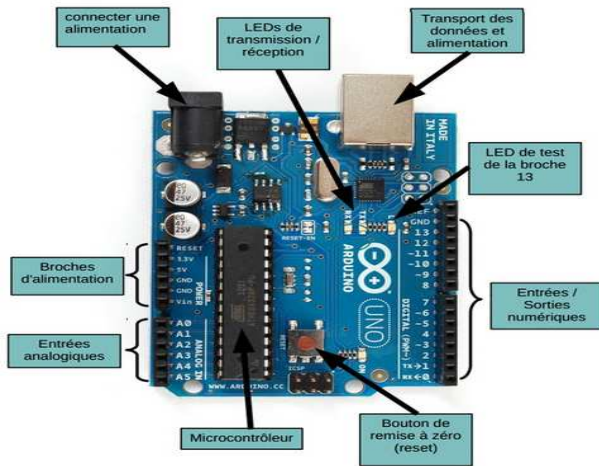


Figure II.14 : Carte Arduino UNO

(PCINT14/RESET) PC8	1	28	PC5 (ADC5/SCL/PCINT13)
(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)
(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)
(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)
(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)
VCC	7	22	GND
GND	8	21	AREF
(PCINT6/XTAL1/TOSC1) PB8	9	20	AVCC
(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)
(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)
(PCINT22/OC0A/AIN0) PD8	12	17	PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)
(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)

Figure II.15. : Brochage de l'ATMega328.

### II.11.2. Pourquoi Arduino :

- ✓ Pas cher : 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, et même les cartes Arduino pré-assemblées coûtent moins de 25 euros (microcontrôleur inclus).
- ✓ multiplateforme : 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) 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.
- ✓ matériel Open source et extensible : les cartes Arduino sont basées sur les microcontrôleurs Atmel ATMEGA8, ATMEGA168, ATMEGA 328, etc... Les schémas des modules sont publiés sous une licence Creative Commons, et les concepteurs de 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, dans le but de comprendre comment elle fonctionne.

### II.13.3. Alimentation

Arduino UNO peut être alimentée via la connexion USB ou avec une alimentation externe. La source d'alimentation est automatiquement sélectionnée.

La plage de tension recommandée est de 7 à 12 volts, les broches d'alimentation sont les suivantes :

VIN : la tension d'entrée de la carte arduino, quand c'est une source d'alimentation externe (par opposition à 5 volts de la connexion USB ou tout autre source d'énergie réglementés).

5V : l'alimentation régulée utilisée pour alimenter le microcontrôleur et les autres composants sur la carte. Cela peut provenir soit de VIN via un régulateur de bord, ou par l'intermédiaire d'un port USB ou d'une autre alimentation 5V régulée.

3.3V : une alimentation de 3.3 Volts générée par un circuit intégrée FTDI, avec un courant maximal de 50 mA.

GND : broche de masse.

#### II.11.4. La mémoire

Le ATmega328 a 32Ko de mémoire flash pour stocker le code dont 0.5 Ko utilisé par le bootloader, 2 Ko de SRAM et 1 Ko de mémoire EEPROM qui peuvent être lues et écrites avec la bibliothèque EEPROM

#### II.11.5. Les entrées et sorties

Broche numérique :

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

Broche analogique :

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

De plus, certaines broches ont des fonctions spécialisées :

Communication Série: Port série (USART) Broches 0 (RX) et 1 (TX). Utilisées pour recevoir (RX) et transmettre (TX) les données série.

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. Voir l'instruction `attachInterrupt()` pour plus de détails

Timers/Counters: Timer0 et Timer2 (comptage 8 bits), Timer1 (comptage 16bits), on peut les utiliser comme temporisateur ou compteur.

WatchdogTimer : programmable.

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

SPI (Interface Série Périphérique): Broches 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ces broches supportent la communication SPI (Interface Série Périphérique) disponible avec la bibliothèque 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 (TwoWire Interface - Interface "2 fils"), disponible en utilisant la bibliothèque `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.

Reset : bouton de réinitialisation du microcontrôleur.

### II.11.6. Communication

Arduino UNO peut communiquer avec un ordinateur, ou un autre arduino, ou d'autres microcontrôleurs, L'ATmega 328 dispose d'une UART (UniversalAsynchronousReceiverTransmitter) pour communication série.

Le logiciel Arduino inclut une fenêtre terminal série (ou moniteur série) sur l'ordinateur et qui permet d'envoyer des textes simples depuis et vers la carte Arduino. Les LEDs RX et TX sur la carte clignotent lorsque les données sont transmises via la connexion USB vers l'ordinateur

Le logiciel Arduino inclut la bibliothèque `Wire`, qui simplifie l'utilisation du bus I2C, et la bibliothèque pour communication SPI, pour utiliser la communication SPI (Interface Série Périphérique).

### II.11.7. Programmation

Pour programmer la carte Aduino UNO, nous avons utilisé le logiciel Arduino C prévu à cet effet. C'est un logiciel libre et gratuit développé pour Windows, Mac et Linux. Ce logiciel présente une interface épurée et permet de développer facilement des programmes qui peuvent être transférés sur la carte en branchant celle-ci sur un port USB. Le langage utilisé par le logiciel est basé sur le C, auquel s'ajoutent des instructions spécifiques au langage Arduino. Le logiciel dispose d'une banque d'exemples très riche qui permet d'utiliser des morceaux de codes pré-écrit.

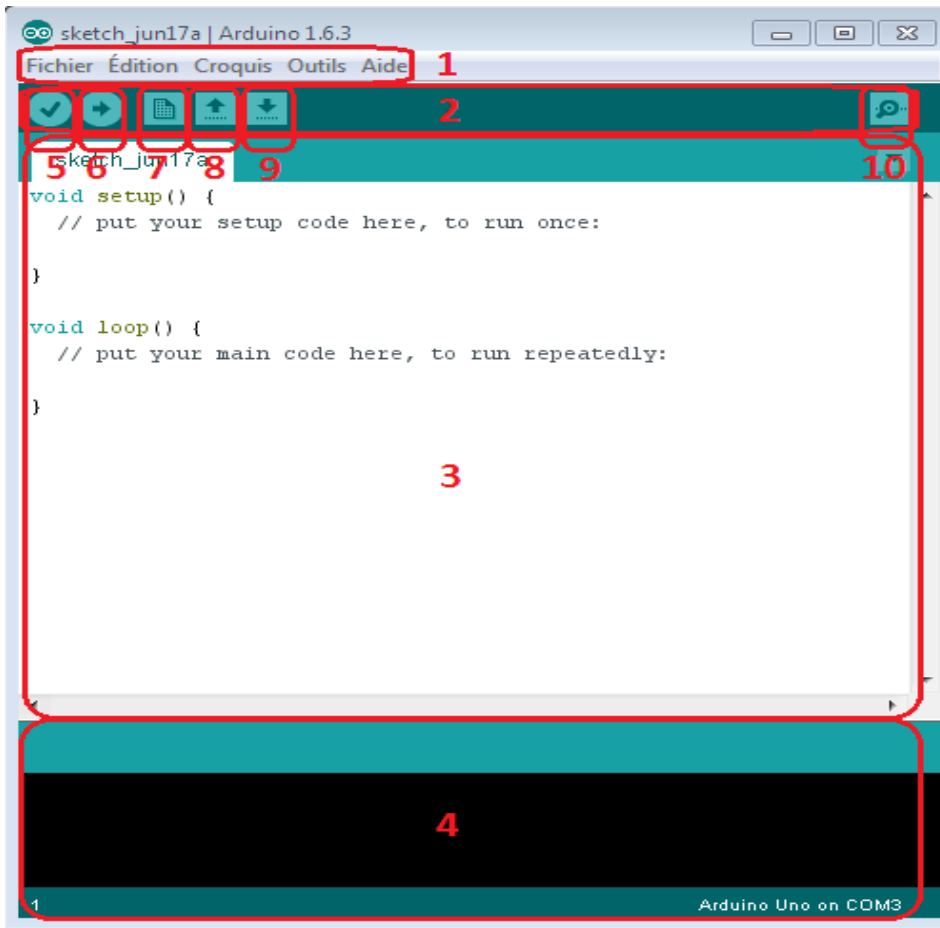


Figure II.16. : **Présentation du logiciel Arduino 1.6.3**

Le cadre numéro 1 : ce sont les options de configuration du logiciel

Le cadre numéro 2 : il contient les boutons qui vont nous servir lorsque l'on va programmer la carte Arduino

Le cadre numéro 3 : ce bloc va contenir le programme que nous allons écrire

Le cadre numéro 4 : celui-ci est important, car il va nous aider à corriger les fautes dans notre programme.

Bouton 5 : Ce bouton permet de vérifier le programme, il actionne un module qui cherche les erreurs dans le programme

Bouton 6 : Charge le programme dans la carte Arduino

Bouton 7 : Crée un nouveau fichier

Bouton 8 : Ouvre un fichier

Bouton 9 : Enregistre le fichier

Bouton 10 : Ouvre le moniteur série.

Exemple : commander un moteur pas à pas pour faire 400 pas en avance

```
#define Enable 6 // Activation du driver/pilote
#define pinStep 7 // Signal de PAS (avancement)
#define pinDir 8 // Direction

void setup(){
  Serial.begin(9600);
  Serial.println("Test A4988");
  pinMode( Enable, OUTPUT );
  pinMode( pinDir , OUTPUT );
  pinMode( pinStep , OUTPUT );}

void loop(){
  int i = 0;
  digitalWrite( pinDir , HIGH); // Direction avant // Avance de 200 pas
  for( i=0; i < 400; i++){
    Serial.println( i );
    digitalWrite( pinStep, LOW );
    digitalWrite( pinStep, HIGH );
    delay (1); }
  Serial.println("Axe bloqué + attendre 5 sec");
  delay( 5000 );
  Serial.println("Debloce axe");
  digitalWrite( Enable, HIGH); // logique inversée
  delay(5000);
  digitalWrite( Enable, LOW); // logique inversée
  Serial.println("Fin de programme");
  while( true );
}
```

## II.12. Commande PWM (Pulse Width Modulation)

### II.12.1. Définition

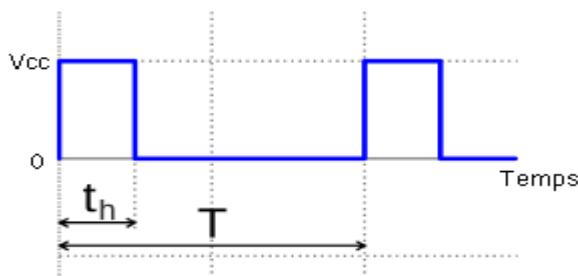
PWM signifie Pulse Width Modulation, modulation par largeur d'impulsion.

La commande par PWM permet de produire simplement une variation de la valeur moyenne d'une tension.

La commande "PWM" est un signal numérique où la tension peut prendre deux valeurs seulement.

Dans certains cas très spécifiques (onduleurs à MLI par exemple) on fabrique un troisième niveau en inversant la tension du niveau haut.

Le signal de cette commande est carré et le niveau bas correspond généralement à 0 Volt. La période est notée  $T$  ; la durée de l'impulsion (pour laquelle la tension est celle de l'état haut) est appelée  $t_h$ .



Si la période change, le signal n'est plus vraiment périodique au sens strict.

On appelle alors  $T$  la pseudo période.

### II.12.2. INTÉRÊT

Le principal intérêt de la technique PWM est de limiter la chauffe des composants électroniques.

Par exemple : une lampe de 20 Watts allumée au maximum consomme 20 W. Si par une commande de gradation elle est allumée au quart de sa puissance, elle consomme 5 W. Le composant analogique devrait alors dissiper 15 W, ce qui implique un énorme radiateur. En PWM, la puissance fournie est soit maximale, soit nulle. Lorsqu'elle est maximale, pendant un quart du temps par exemple, il n'y a pas besoin de dissiper de puissance résiduelle. Lorsqu'elle est nulle, il n'y a pas besoin de dissiper non plus de puissance, car elle n'est pas fournie du tout.

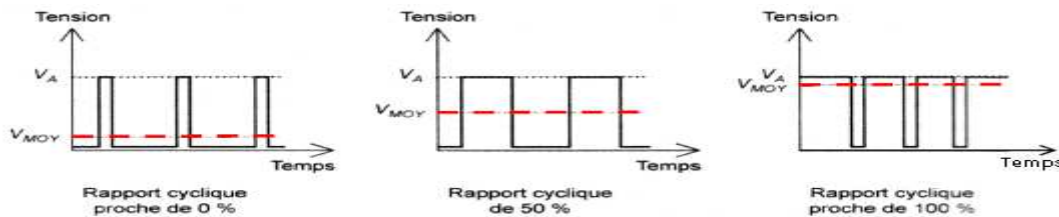
Un autre intérêt du PWM est que la tension appliquée au moteur pendant  $t_h$  est  $V_{cc}$ . Celle-ci est suffisante pour vaincre les frottements et faire tourner le moteur. La tension moyenne appliquée au moteur est proportionnelle au rapport cyclique, ce qui permet d'avoir des consignes de vitesse faibles :

$$V_{MOY} = \frac{t_h \times V_{CC}}{T}$$

**II.14.3.FRÉQUENCE**

La commande d'actionneurs de puissance par PWM est très liée à la notion de fréquence. Pour que l'impression d'une valeur moyenne constante d'allumage apparaisse, il faut que l'alternance d'allumage/extinction soit suffisamment rapide pour qu'elle ne se remarque pas.

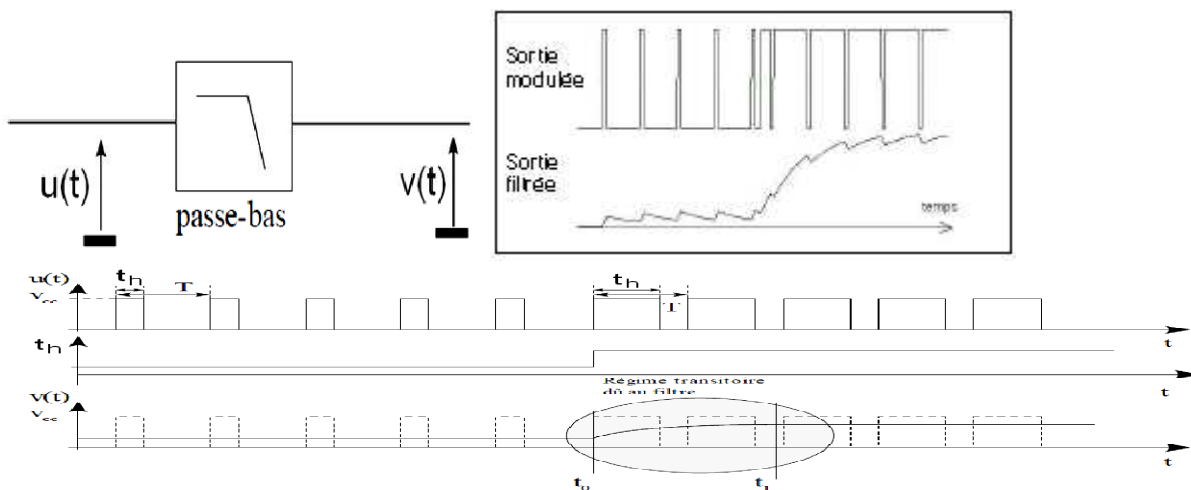
Selon les utilisations la fréquence du PWM va de 100 Hz (100 cycles par seconde) à 200 kHz.



**Remarques:**

Même si le signal n'est nullement une tension analogique, il peut le devenir si on lui fait subir un filtrage de type passe-bas.

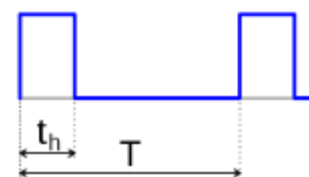
Filtrer ce signal carré permet d'obtenir la tension moyenne. La fréquence de coupure du filtre doit être plus faible que la fréquence du signal carré. Autrement dit la fréquence porteuse doit être significativement supérieure à la fréquence de coupure du filtre, pour que l'ondulation résiduelle soit faible.



**II.12.4. RAPPORT CYCLIQUE**

On appelle rapport cyclique le rapport :

$$\alpha = 100 \times \frac{t_h}{T} \text{ exprimé en pourcentage.}$$



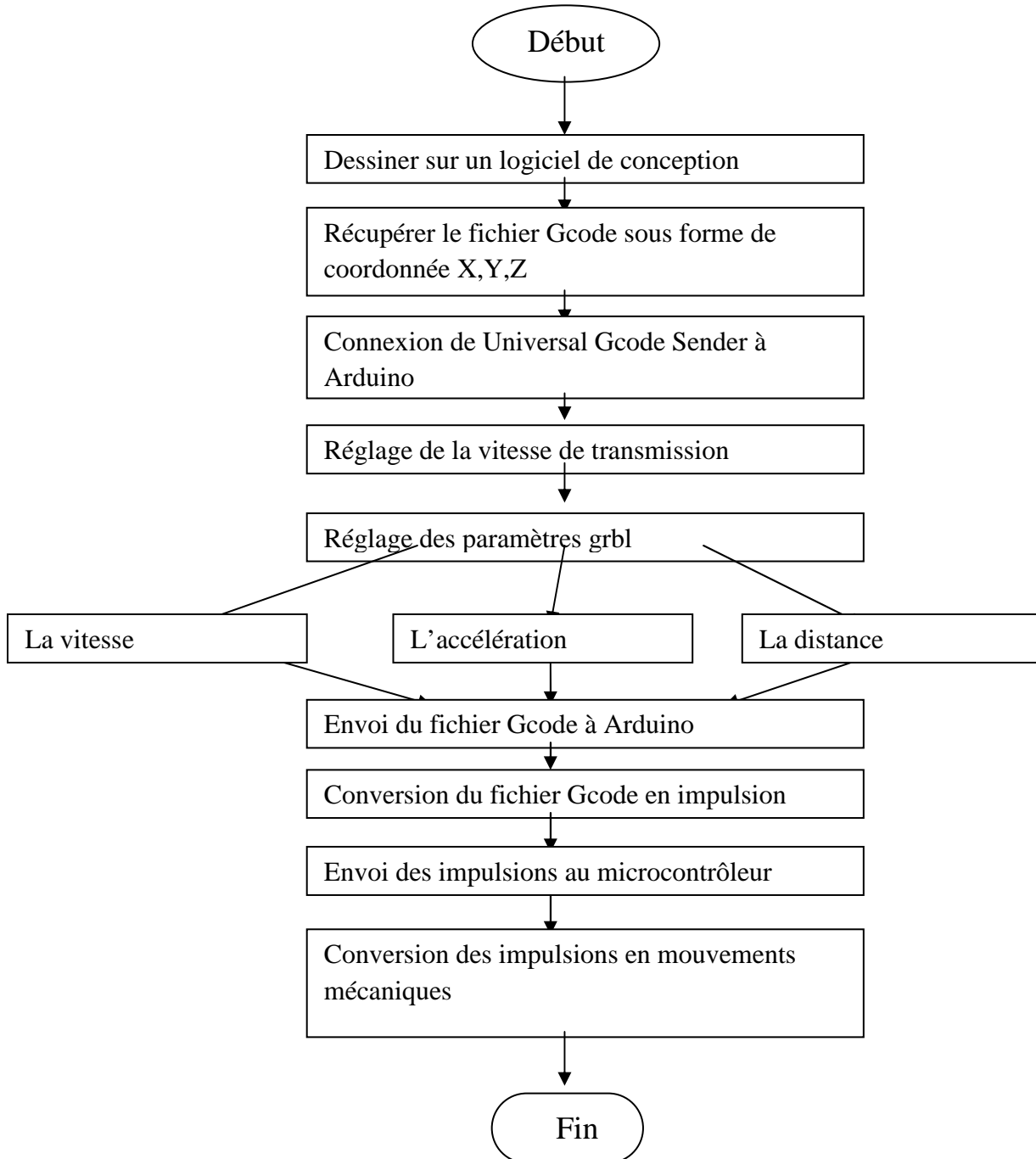
Si  $t_h = 0$  alors  $a = 0\%$  et la tension moyenne de sortie est nulle.

Si  $t_h = T$  alors  $a = 100\%$  et la tension moyenne de sortie est égale à  $V_{cc}$ .

## II. Partie Informatique et Programmation

### II.13. Organigramme

Il représente la partie programmation d'une machine CNC



## II.14. Les fichiers Gcode :

Développé par l'EIA au début des années 1960, et finalement normalisé par l'ISO en février 1980' est un type de fonction utilisé dans le langage de programmation à commande numérique qui contient l'information pour positionner un outil dans les démentions x y z .

G-code est séparé de code M et T-code, c est des codes qui contrôlent la machine et l'outillage.

La version standard de G-code est connue comme RS-274D Développé à l'origine pour des machines-outils par enlèvement de matière, le G-code est désormais utilisé dans un domaine très vaste de la fabrication, avec des adaptations :

- Usinage par enlèvement de matière: tournage, fraisage, perçage, gravure, défonçage
- Découpe avec : couteau, laser, jet d'eau, plasma, flamme/oxydation
- Poinçonnage
- Impression 3D: Par dépôt de matière

En résumé le G-CODE est un langage de programmation simple dédié a la programmation des machines a commandes numériques par ordinateur on l'appelle G-CODE parce que toutes les lignes d'instruction commencent par une commande de type G xx (xx étant le type d'opération que on veut faire).

Il nous permet de faire des programmes simples pour les envoyer dans le contrôleur ARDUINO, il est important de connaître quelques bases de ce langage.

### II.14.1. Le fonctionnement du G-Code :

Le principe est simple: on va écrire une suite de lignes comportant quelques instructions et commandes qui vont demander a la machine de réaliser des mouvements dans l'espace en 3 dimensions. On va pour cela utiliser un repère cartésien dont les 3 axes de référence sont les axes X - Y - Z

Voici quelques codes individuels communs, que lorsqu'il est combiné, guident le mouvement d'une machine

- G00: Positionnement rapide  
Ce code provoque un fonctionnement rapide sur la machine.
- Interpolation linéaire : G01  
La machine se déplacera en ligne droite, effectuer l'usinage approprié (fraisage, coupe, etc.).
- G02 : mouvement circulaire  
La machine se déplace vers la droite dans un mouvement circulaire ou d'effectuer le procédé d'usinage approprié
- G03 : mouvement circulaire antihoraire  
Ce code est le même que G02, mais de l'autre sens
  
- G17 : Sélection du plan XY
- G18: Sélection du plan XZ
- G19: Sélection du plan YZ

Ces codes manœuvrer la machine sur des plans différents pour le mouvement coordonné.

- G20: Programmation en pouces

- G21: Programmation en mm

Voici un petit tableau récapitulatif des codes de base :

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

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

## II.15. Le fichier GRBL : [9]

Grbl : est un firmware de haute performance conçue pour contrôler le mouvement des machines praticable sur un Arduino. Il a été adapté pour être utilisé dans certaines projets comme les machine CNC et les imprimantes 3D, Le programme est écrit en C hautement optimisé utilisant chaque fonctionnalité intelligente de l'AVR-puces pour atteindre un timing précis et le fonctionnement asynchrone. Il est capable de maintenir plus de 30 kHz de la stabilité, la rapidité des impulsions de commande libres.

En 2009, Simen Svale Skogsrud a honoré la communauté open-source en écrivant et en libérant les premières versions de Grbl à tout le monde. Depuis 2011, Grbl va de l'avant comme un projet

open-source mené par la communauté sous la direction pragmatique de Sonny Jeon Ph.D. (chamnit).

Le projet a d'abord été inspiré par l'interprète Arduino Gcode par Mike Ellery

### II.15.1. Schéma de Pin pour Grbl :

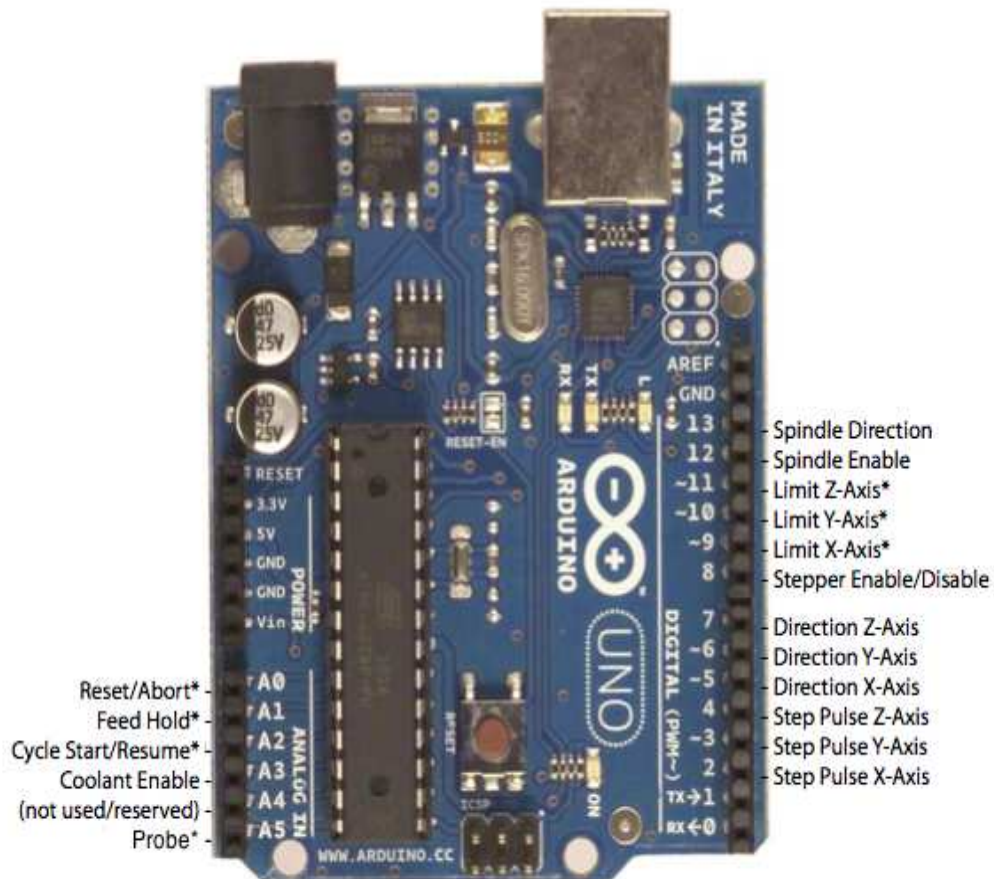


Figure II.17 : Schéma de Pin pour Grbl :

### II.15.2.Fonctionnement de grbl :

Après avoir flashé Grbl à l'Arduino, la connexion est assez simple. Il faut utiliser l'Arduino IDE lui-même pour se connecter à Grbl, ou D'autres programmes de port série, tels que DIVA ou PuTTY,

Avec le Xloader on peut modifier les paramètres comme suit:

- Sélectionner le fichier HEX.
- sélectionner la configuration dans L'Arduino.
- Sélectionner le port COM correct.
- Gardez la vitesse de transmission est que ... (Il va changer pour chaque type de carte.

### II.15.3. Le branchement :

La broche 'step »pour les axes X, Y et Z sont broches sur les pins 2, 3 et 4.  
La broche "Dir» pour les axes X, Y et Z broches sur les pins 5, 6 et 7

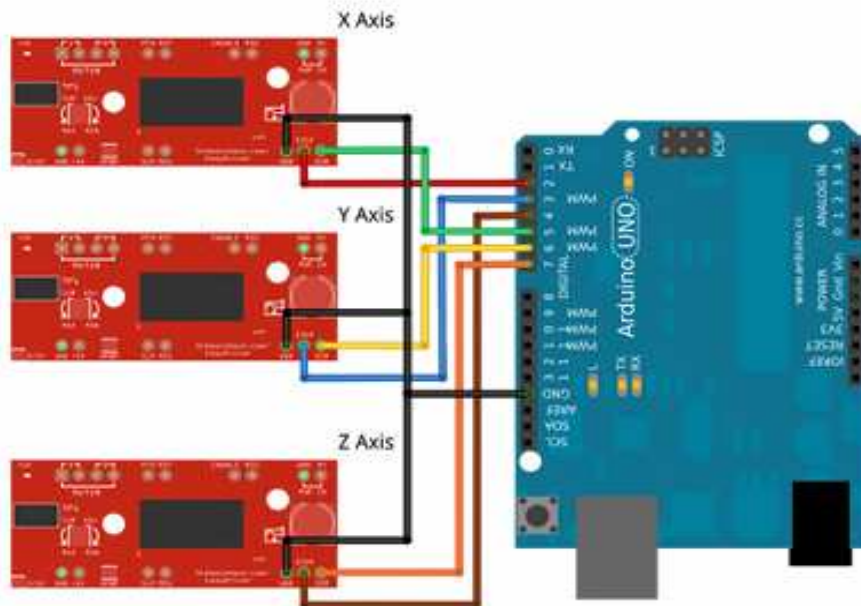


Figure II.18 : branchement des pins STEP et DIR sur arduino

### II.15.4. Connecter grbl à Arduino :

1. Télécharger XLoader.

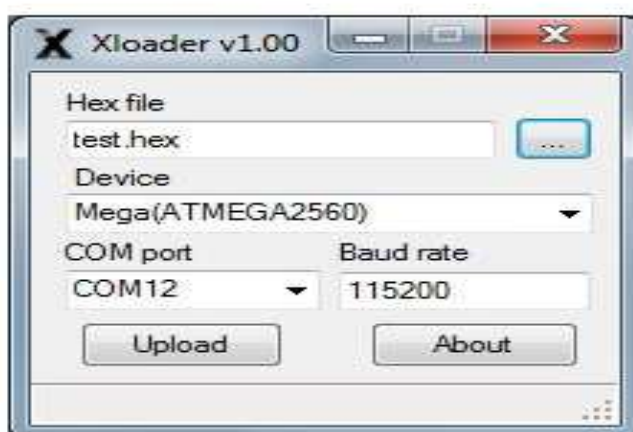


Figure II.19 : XLoader

2. Ouvrir XLoader et sélectionner le port COM de votre Arduino dans le menu
3. Sélectionner le périphérique approprié dans la liste «Appareil».

4. Vérifier le taux de transmission pour le périphérique est de 57600 pour (Atmega 328) et de 115200 pour Uno (Atmega 328).

### II.15.5. Paramètres de grbl :

Grbl fonctionne avec un certain nombre de variables internes sauvegardées en mémoire. Pour changer la valeur d'une variable, Il suffit de taper "\$suivant d'un numéro de variable =valeur.

Par exemple pour changer la résolution de l'axe X à 300 pas par millimètre, il suffit de taper : \$0=300

le paramètre des pas par millimètres se calcule avec le nombre de pas par tour du moteur pas à pas et le pas de la vis.

- Exemple2 pour l'axe X : si on utilise une vis d'avance avec un pas de 1mm et un moteur de 200pas, \$0 = 200.
- exemple : si sur les 3 axes X,Y,Z on utilise des vis sans fin de 5mm / tour + des moteurs à 200 pas par tour, et un contrôleur fonctionnant en micropas (1/16e de pas). Dans ce cas, il faudra  $200 \times 16 = 3200$  pas pour faire 5mm. La valeur à indiquer en résolution des axes est donc de  $3200 / 5\text{mm} = 640$  pas / mm (paramètres \$0 \$1 \$2).
- \$3 permet de régler la largeur d'impulsion de pas, en général entre 5 et 50 microsecondes, voir la fiche de donnée du driver (A4988)
- \$4 permet de régler la vitesse par défaut en usinage (G1 G2 G3)
- \$5 permet de régler la vitesse en déplacement rapide (G0)
- \$6 permet d'inversion les signaux de pas et de direction pour chaque axe. Pour chacun d'entre eux devant être inversé, prendre la valeur ci dessous, en faire la somme et l'envoya à \$6 :

" Pas X : bit 2 = 4

" Pas Y : bit 3 = 8

" Pas Z : bit 4 = 16

➤ " Direction X :bit 5 = 32

" Direction Y : bit 6 = 64

" Direction Z : bit 7 = 128

➤ les bits 0 et 1 ne sont pas utilisés

➤ Ainsi pour inverser la direction de X et Y, il suffirait de mettre  $32+64 = 96$  soit \$6=96 => en binaire : 1100000

- \$7 permet de régler le délai de mise au repos après un mouvement complet. on peut le mettre simplement a zéro, ou entre 25 et 50 millisecondes. on pouvez également garder les valeurs axes actives tout le temps en mettant le paramètre a la valeur max soit 255=> \$7=255
- \$8 permet de régler l'accélération MAX.

- \$9 permet de régler les problèmes dans les virages.
  - \$10 sert à régler la résolution des arcs. Ceux ci sont en effet transformés en une succession de petits segments de droite.
  - \$11 ce paramètre doit être changé avec une extrême prudence. ne changer cette valeur si ya des problèmes avec les arcs.
  - \$12 permet de fixer le nombre de décimales après la virgule.
  - \$13 permet de changer l'unité de l'affichage de position temps réel :
    - ✓ \$13= 0 affichage en mm
    - ✓ \$13= 1 affichage en pouces
  - \$14 permet de changer le mode de démarrage de cycle :
    - ✓ \$14= 1 démarrage auto => valeur généralement retenue
    - ✓ \$14= 0 démarrage manuel sur bouton démarrage de cycle
  - \$15 change le mode d'activation du moteur. par défaut la broche d'activation du moteur est à niveau haut pour désactiver et niveau bas pour activer. si on souhaite un fonctionnement opposé, il faut inverser en changeant la valeur \$15=1.
  - \$16 pour activer les fin de course=> \$16=1
  - \$17 pour activer le cycle de référencement absolu (position zero machine => nécessite des capteurs sont a l'état initial).\$17=1
  - \$18 change le sens de direction des mouvements de référencement. par défaut, les Switch sont supposés être dans le sens des directions positives, l'axe Z puis les axes X,Y, pour inverser, taper \$18=1
  - \$19 permet de paramétrer la vitesse d'arrivé (lente) sur les Switch de fin de course.
  - \$20 permet de paramétrer la vitesse de recherche rapide sur les Switch de fin de course.
  - \$21 permet de fixer le temps d'amortissement sur contacts de fin de course.
  - \$22 active le pull off des switchs de fin de course.
- 
- \$ 100 = (x, pas / mm).
  - \$ 101 = (y, pas / mm).
  - \$ 102 = (z, pas / mm).
  - \$ 110 = (taux maximum de x, mm / min).
  - \$ 111 = (taux max y, mm / min).
  - \$ 112 = (taux maximum de z, mm / min).
  - \$ 120 = (x accélérations, mm / s ^ 2).
  - \$ 121 = (y accélérations,, mm / s ^ 2).
  - \$ 122 = (z, mm / s ^ 2).
  - \$ 130 = (x max, mm).
  - \$ 131 = (y max , mm).
  - \$ 132 = (z max, mm).
  - # \$ (Voir n paramètres).
  - G \$ (vue de l'état de l'analyseur).
  - \$ I (Voir l'info de construction).
  - \$ N (visionner les autres blocs de démarrage).
  - \$ X = valeur (enregistrer les paramètres Grbl).

- \$ Nx = ligne (sauvegarde bloc de démarrage).
- \$ C (mode gcode à cocher).
- \$ X (verrouillage de l'alarme de tuer).
- \$ H (cycle de ralliement de fonctionner).
- ~ (Début de cycle).
- ! (Alimentation en attente).
- ? (statut actuel).
- Ctrl-X (réinitialiser Grbl).

## II.16. Universal Gcode sender :

Universal Gcode sender (UGS) est un programme Java utilisé pour envoyé un fichier gcode vers le GRBL d'une machine à commande numérique.

Avec l'expéditeur G-Code Universel il est possible de contrôler le système avec des commandes G-code, comme le déplacement des moteurs pas à pas a la position initiale.

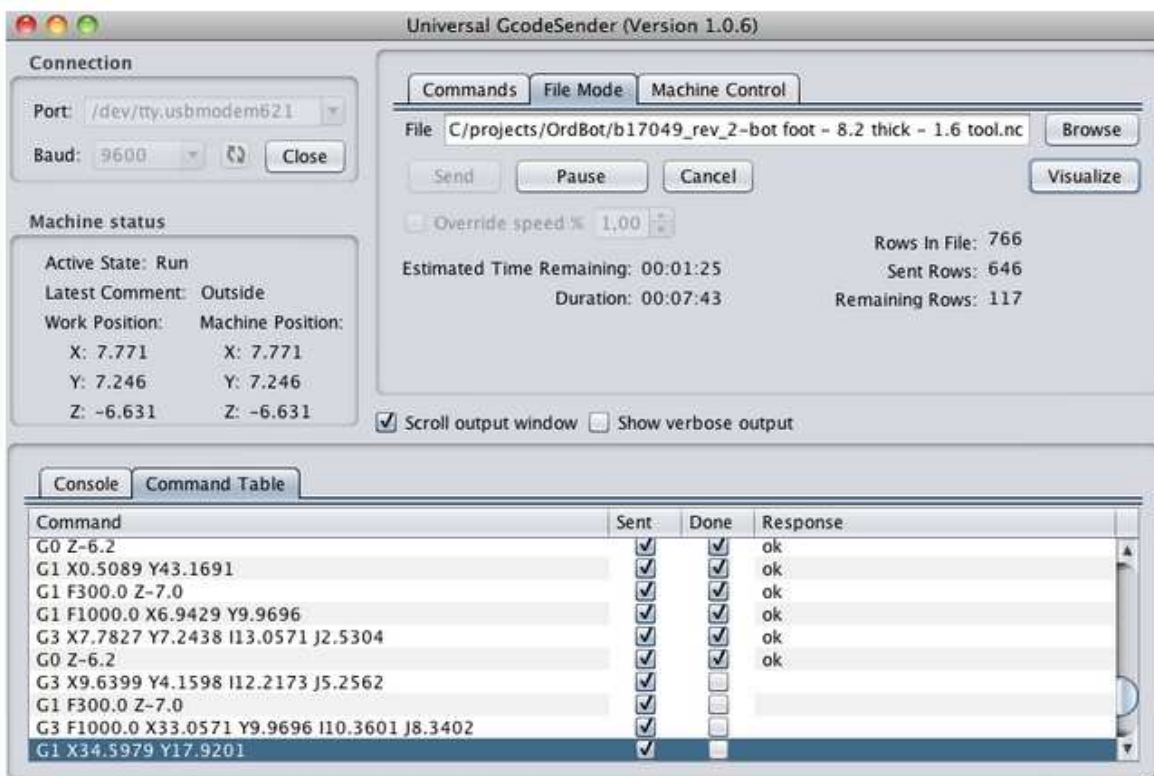


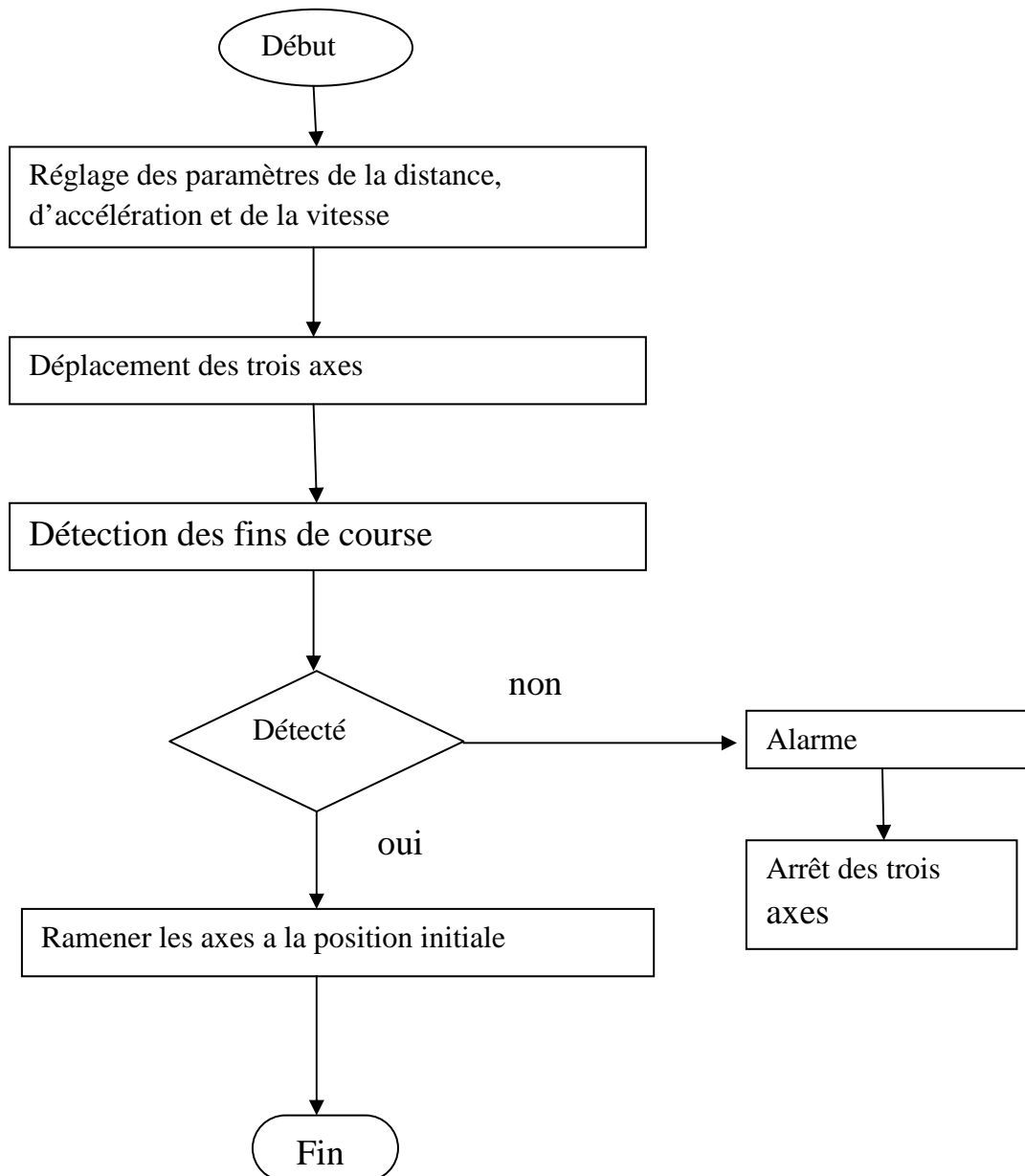
Figure:II.20 : Universal Gcode sender

## II.17. Utilisation du firmware grbl

### II.17.1. paramétrages de la machine CNC

Pour le premier démarrage de la machine on doit la paramétrer afin qu'elle détecte les contacteurs de fin course et positionner les axes à leurs position initial.

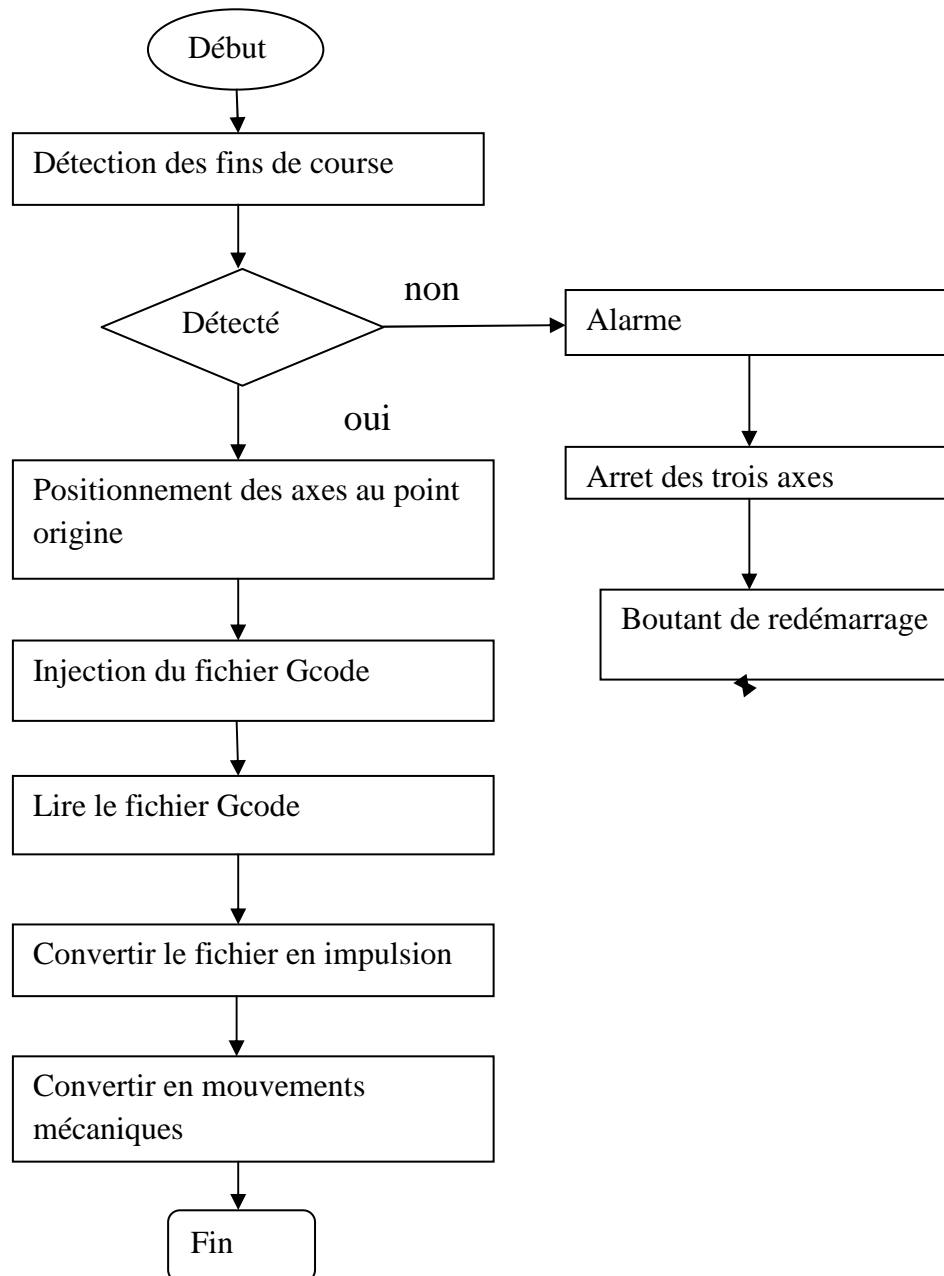
L'organigramme suivant nous montre les étapes à suivre



### II.17.2. Initialisation de la machine

A chaque démarrage de la machine l'initialisation effectue la détection des fins de course X,Y,Z dans le cas échéant une alarme sera retenue et déclenche un arrêt des trois axes.

L'organigramme suivant nous montre les étapes à suivre



## II.18. Conclusion :

Dans ce chapitre nous avons montré les étapes de conception que nous avons répartie en trois parties partie mécanique, partie électrique et partie informatique (programmation)

Et nous avons essayé d'expliquer la commande des trois moteurs avec une carte de commande basée sur Arduino et les circuits de puissance **A4983** et nous avons fais une étude sur les choix des moteurs et des circuits de puissance à utiliser.

Après cela nous proposons de valider nos résultats via la réalisation d'un prototype.

Dans le chapitre suivant nous présentons la fabrication du prototype suivant la démarche de conception présentée précédemment.

### III.1. Introduction :

que ce soit pour l'usinage 3D ou pour l'Impression 3D , la maîtrise du déplacement d'un outil dans les 3 dimensions de l'espace est devenue primordiale. ce déplacement dans les 3 axes de l'espace ( X, Y, Z) nécessite de maîtriser 3 grands domaines techniques « électronique, mécanique et informatique »

Dans ce chapitre nous allons décrire les solutions adoptées par détail pour répondre aux spécifications de notre cahier de charge et ce en abordant la conception détaillée de chaque partie du système

### III.2. Structure mécanique

Pour réaliser un système mécanique permettant de déplacer un objet dans l'espace, nous avons opté pour l'aluminium pour la structure des axes et de la base car il est plus léger que l'acier et pour les vis sans fin et les glissières on les a faites avec de l'acier.

#### III.2.1. Les différentes parties de la machine CNC

La machine est constituée de quatre parties qui sont : la base, l'axe X, l'axe Y et l'axe Z

##### a) La base



Figure III.1 : la structure de la base de la machine

**b) L'axe X**

L'axe X est constitué de deux parties reliées par des vis comme le montre la figure suivante et il est entraîné par un couroïl.



Figure III.2 : l'axe X



Figure III.3 : le couroïl qui entraîne l'axe X

**c) L'axe Y**

L'axe Y est constitué d'une plaque de métal et de deux glissières entraînés par un couroïl.

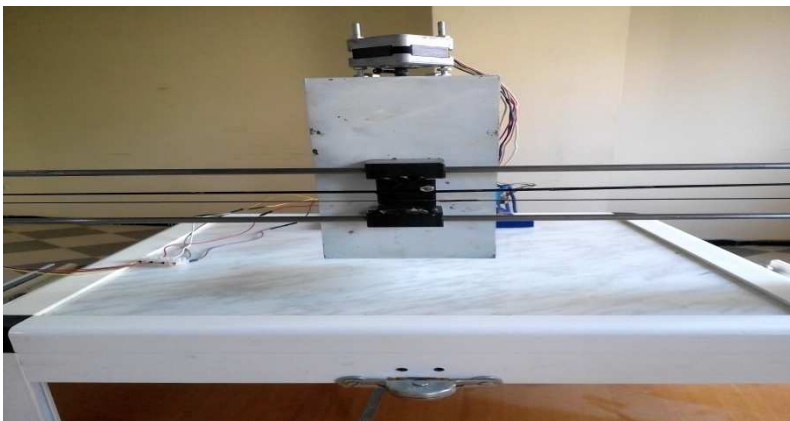


Figure III.4. : l'axe Y

### d) L'axe Z

L'axe Z est constitué d'un organe terminal et de deux glissières entraîné par une vis sans fin

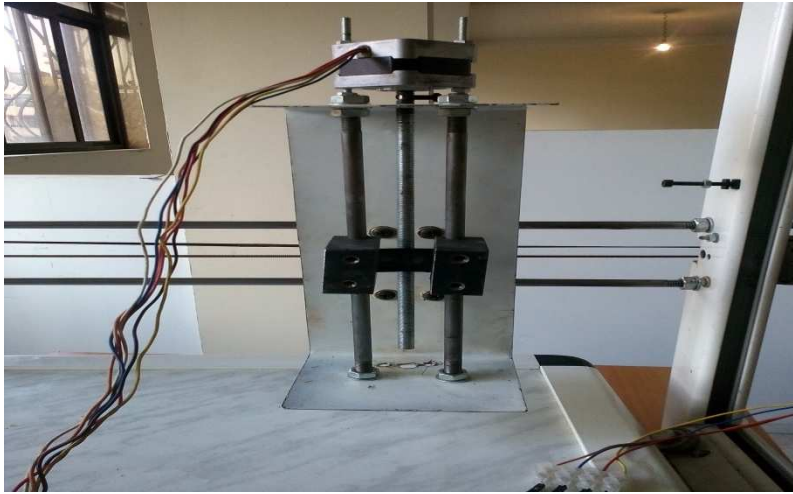


Figure III.5. : l'axe Z

### III.3. Assemblage complet de la machine CNC

Après avoir cité les différentes parties de la machine CNC, nous allons les assembler pour avoir une CNC complète, la figure ci-dessous le montre.

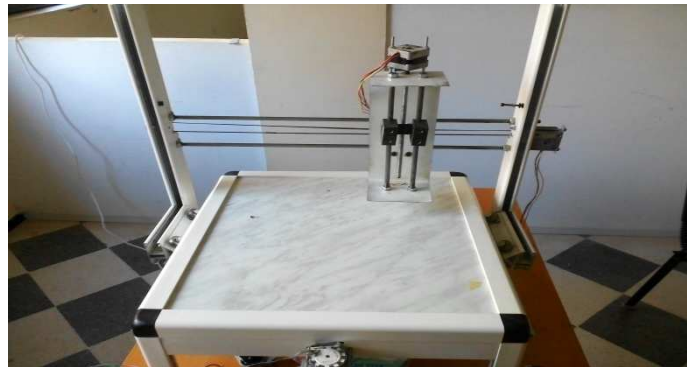


Figure III.6 : assemblage complet des trois axes avec la base

### III.4. Carte de commande pour la machine CNC

La carte de commande est constituée d'une carte Arduino UNO, alimentée par l'ordinateur, et de trois drivers de moteurs pas à pas

La carte Arduino délivre les signaux PWM aux drivers de moteurs pas à pas pour commander les différents moteurs qui contrôlent les axes de la machine CNC, qui sont de leurs cotés alimentés avec une tension de 12volts délivré par un générateur de tension,

La figure ci-dessous montre le schéma de branchement sur la platine d'expérimentation.

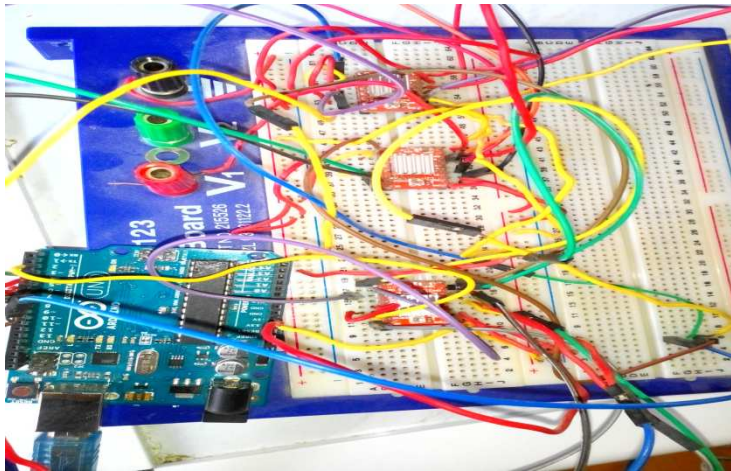


Figure III.7 : le schéma de branchement

Les moteurs pas à pas peuvent être commandés par plusieurs drivers et notre choix s'est porté sur le A4983 car il dispose des caractéristiques suivantes

- Simple à utiliser
- Possède une interface de contrôle de la direction
- Fonctionne avec 8-35 V et fournit jusqu'à 2A par bobine
- Limite de courant ajustable
- Cinq résolutions de micro pas différentes : Pas complet, demi-pas, quart de pas, huitième de pas et seizième de pas
- Protection contre les courts-circuits et les surcharges

### III.5. Les actionneurs

Il existe plusieurs types de moteurs pas à pas mais nous avons opté pour des moteurs qui ont une plus grande résolution et un grand couple qui sont de référence « Sanyo Denki »

#### III.5.1. Caractéristiques des moteurs

- Model 103-591-0155 Sanyo Denki 0,9deg/pas
- Tension de source 24V
- Courant opérationnel 0,85A/phase
- Inductance 3,7 mH/phase
- Résistance 4,3Ω/phase

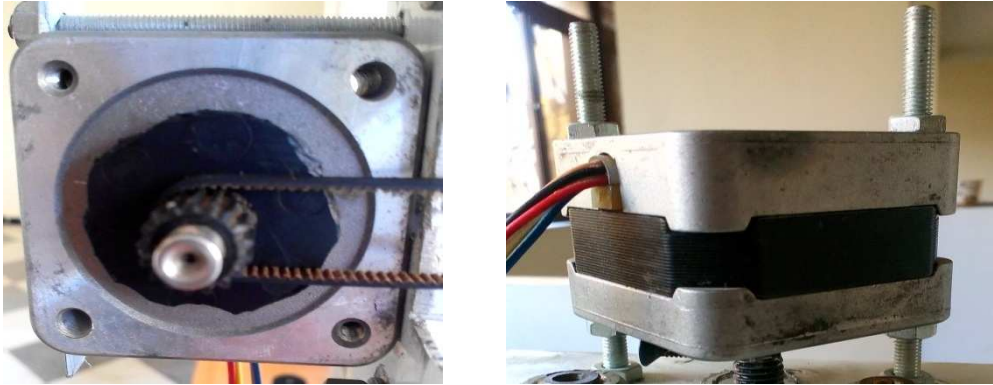


Figure III.8 : Moteur pas à pas Sanyo Denki

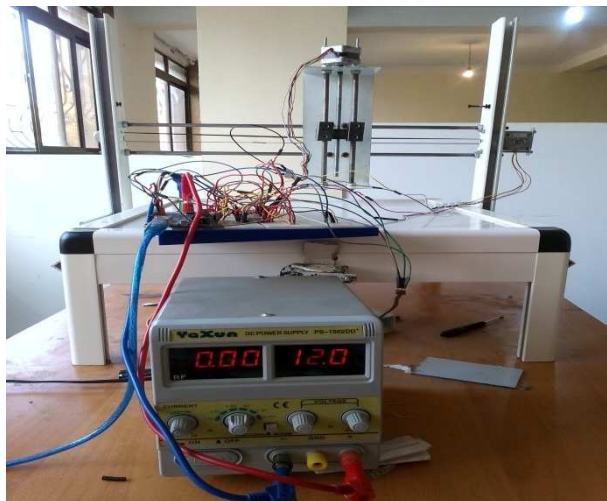


Figure III.9 : Structure complète de la machine

### III.6. Conclusion

Dans ce chapitre, nous avons pu décrire les différents éléments constitutifs de La machine CNC, et injecter le programme dans la carte Arduino UNO, pour commander les axes de la machine et ainsi voir son fonctionnement lorsqu'elle effectue sa tâche

## Conclusion Générale

---

Ce projet nous a permis d'approfondir nos connaissances dans le domaine de l'automatique ainsi que la mise en application des connaissances théorique acquise durant notre formation.

Nous avons rencontré plusieurs problèmes durant la conception de la commande et la réalisation pratique, problèmes d'ordre mécanique, électrique et informatique.

mais grâce à ce que nous avons appris durant nos études académiques et l'assistance de nos encadrant nous avons pu associer les différents éléments dont la partie la plus essentiel qui est la commande des trois moteur pas à pas logiciel, interface, programmes...etc.

En conclusion, nous espérons que ce modeste travail pourrait servir de référence aux Projets future des prochaines promotion et les inciter à s'intéresser d'avantage au côté pratique de l'automatique.

## Référence bibliographique

---

[1] [www.technologystudent.com](http://www.technologystudent.com)

[2] **Huglres Austin**, « Electric Motors and Drives » 2013

[3] **Oqui C Patrice**, « Moteurs pas à pas et PC » 2004

[4] **Kenjo Takashi**, « Stepping Motors and their Microprocessor controls » 1995

[5] <http://www.reprap-france.com/produit/363-pilote-de-moteur-pas-a-pas-stepstick>

[6] **Alain Pruski**, « Robotique générale » 1988, 207p.

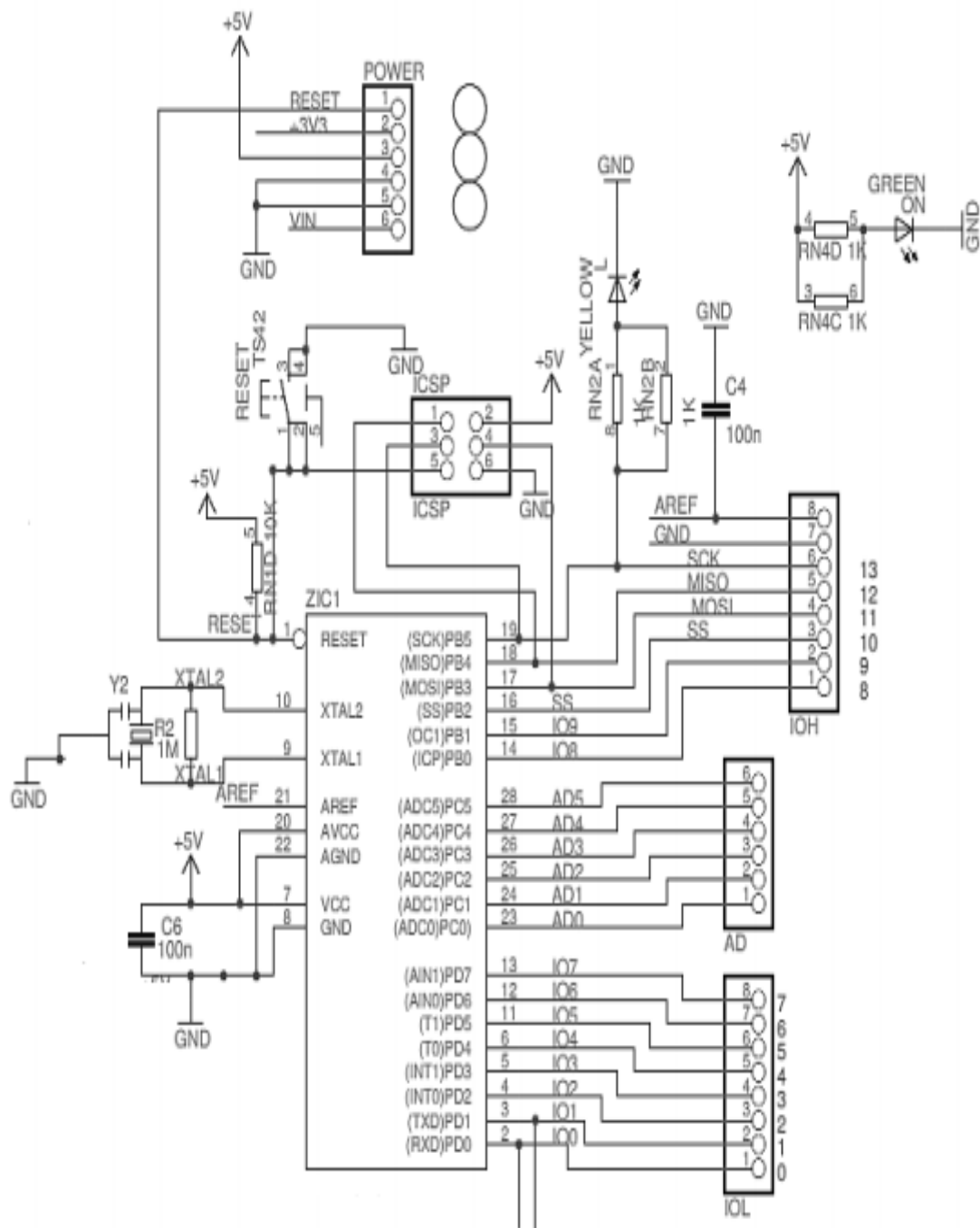
[7] **Jack Lpovski . G**, « Introduction to Microcontrôleur » 2004

[8] **Nassimo BANZI**, « Démarrez avec Arduino » 2013 102P.

[9] <http://www.shapeoko.com/wiki/index.php/Grbl>

# Annexe


## Schéma simplifié de la carte Arduino UNO



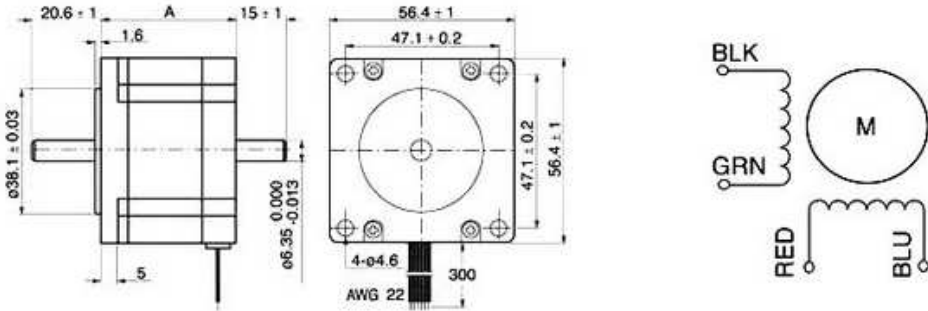
# Annexe

## Datasheet d un moteur pas a pas de type sanyo denki

Item	Specifications
Step Angle Accuracy	± 5% (full step, no load)
Resistance Accuracy	± 10%
Inductance Accuracy	± 20%
Temperature Rise	80 °C Max. (rated current, 2 phase on)
Ambient Temperature	-10 °C ~ +50 °C
Insulation Resistance	100MΩMin., 500 VDC
Dielectric Strength	500VAC for one minute
Shaft Radial Play	0.06 Max. (450 g-load)
Shaft Axial Play	0.08 Max. (450 g-load)



Model No.	Rated Voltage	Current /Phase	Resistance /Phase	Inductance /Phase	Holding Torque		# of Leads	Rotor Inertia	Weight
Single Shaft	V	A	Ω	mH	Oz-in	Kg-cm		g-cm <sup>2</sup>	Kg
TL57STH76-04A2	3.2	2.8	1.13	3.6	262	18.9	4	480	1



## Schéma des pins d'un atmega328

**ATmega328P pin mapping**

<p><b>Arduino function</b></p> <ul style="list-style-type: none"> <li>reset</li> <li>digital pin 0 <b>RX</b></li> <li>digital pin 1 <b>TX</b></li> <li>digital pin 2</li> <li>digital pin 3 <b>PWM</b></li> <li>digital pin 4</li> <li>VCC</li> <li>GND</li> <li>crystal</li> <li>crystal</li> <li>digital pin 5 <b>PWM</b></li> <li>digital pin 6 <b>PWM</b></li> <li>digital pin 7</li> <li>digital pin 8</li> </ul>		<ul style="list-style-type: none"> <li>PC6 1</li> <li>PDO 2</li> <li>PD1 3</li> <li>PD2 4</li> <li>PD3 5</li> <li>PD4 6</li> <li>VCC 7</li> <li>GND 8</li> <li>PB6 9</li> <li>PB7 10</li> <li>PD5 11</li> <li>PD6 12</li> <li>PD7 13</li> <li>PB0 14</li> <li>PC5 28</li> <li>PC4 27</li> <li>PC3 26</li> <li>PC2 25</li> <li>PC1 24</li> <li>PC0 23</li> <li>GND 22</li> <li>AREF 21</li> <li>AVCC 20</li> <li>PB5 19</li> <li>PB4 18</li> <li>PB3 17</li> <li>PB2 16</li> <li>PB1 15</li> </ul>	<p><b>Arduino function</b></p> <ul style="list-style-type: none"> <li>analog input 5</li> <li>analog input 4</li> <li>analog input 3</li> <li>analog input 2</li> <li>analog input 1</li> <li>analog input 0</li> <li>GND</li> <li>analog reference</li> <li>AVCC</li> <li>digital pin 13</li> <li>digital pin 12</li> <li><b>PWM</b> digital pin 11</li> <li><b>PWM</b> digital pin 10</li> <li><b>PWM</b> digital pin 9</li> </ul>
--	---	---	--

When using ISP to program the chip

## Annexe

### Schéma de câblage d'un driver A4983

