

N d'ordre:...../2019

En vue de l'obtention du diplôme de :

MASTER ACADEMIQUE

En Génie Mécanique

Option : fabrication mécanique et productive

Présenté par :

Benabderazag khalil

Thème

CONTRIBUTION A LA REALISATION D'UNE MOCN

Devant le jury composé de :

NOM et Prénom	Grade	Qualité
SLAMANI. M	MCA	Examineur
MECHTA.A	DOC	Examineur
ZAOUI. M	MCA	Président
ARSLANE .M	DOC	Encadreur
MAKRI. H	MCA	Co-encadreur

Année Universitaire : 2018 / 2019

Dédicace

Je dédie ce mémoire à :

Ma mère, qui a œuvré pour ma réussite, de par son amour, son soutien, tous les sacrifices consentis et ses précieux conseils, pour toute son assistance et sa présence dans ma vie, reçois à travers ce travail aussi modeste soit-il, l'expression de mes sentiments et de mon éternelle gratitude.

Mon père, qui peut être fier et trouver ici le résultat de longues années de sacrifices et de privations pour m'aider à avancer dans la vie. Puisse Dieu faire en sorte que ce travail porte son fruit ; Merci pour les valeurs nobles, l'éducation et le soutien permanent venu de toi.

Mes frères et sœurs qui n'ont cessé d'être pour moi des exemples de persévérance, de courage et de générosité.

Mes professeurs qui doivent voir dans ce travail la fierté d'un savoir bien acquis.

Remerciements

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

Je tiens à exprimer mes remerciements vifs à l'encadreur et co-encadreur Mr. ARSLANE mustapha et Mr. MAKRI hocin pour l'aide et pour ses conseils précieux qu'il nous a donné durant notre étude et la réalisation de ce travail.

Je remercie s'adressent ainsi aux : Mr. Le président et les membres de jury d'avoir accepté juger et évaluer ce travail .

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

Abréviations

DSP : Digital Signal Processor

UAL : Unité Arithmétique et Logique

CNC : Commande Numérique par Calculateur

CAO : Conception Assisté par Ordinateur

FAO : Fabrication Assisté par Ordinateur

DFN : Définition de Formes Numérisées

CFAO : Conception, Fabrication Assisté par Ordinateur

MOCN : Machine-Outil à Commande Numérique

RAM : Mémoire vive ou Mémoire PC (Random Access Memory)

ROM : Mémoire morte (Read Only Memory)

EEPROM : Mémoire morte effaçable électriquement et programmable

(Electrically Erasable Read Only Memory)

SCI : Interface de communication série

ISO : Organisation internationale de normalisation

DXF : Drawing eXchange Format

Table des matières

Résumé

Remerciement

Dédicace

Abréviation

Liste des figures & Liste des tableaux

introduction generale

CHAPITRE.1

I.1 INTRODUCTION	4
I.2 HISTORIQUE DES MACHINES-OUTILS [1]	4
I.3 LA COMMANDE NUMERIQUE.....	5
I.3.1 Origines de la commande numérique.....	5
I.3.2 Principe de la commande numérique	5
I.4 LA STRUCTURE D'UNE MACHINE A COMMANDE NUMERIQUE.....	6
I.4.1 Partie opérative	6
I.4.1.1 Tâches effectuées	7
I.4.1.2 Armoire électrique de relayage ou armoire de puissance.....	7
I.4.2 Partie commande.....	8
I.5. PRINCIPE DE FONCTIONNEMENT D'UNE MACHINE A COMMANDE NUMERIQUE	9
I.6 LES FAMILLES DES MACHINES A COMMANDE NUMERIQUE	9
I.6.1 Fraisage à commande numérique (FCN)	9
I.6.2 Tournage à commande numérique (TCN)	10
I.6.3 Rectification à commande numérique.....	10
I.6.4 Electro-érosion à commande numérique.....	11
I.7 CLASSIFICATION DES MOCN [14]	11
I.7.1 Classification des MOCN selon le mode de fonctionnement	12
I.7.1.1 Fonctionnement en boucle ouverte	12
I.7.1.3 Fonctionnement en boucle fermée	13
I.7.1.2 Fonctionnement avec commande adaptative.....	13
I.7.2 Classification des MOCN selon le nombre d'axe	14
I.7.3 Classification des MOCN selon le mode d'usinage.....	14
I.8 LA MACHINE A COMMANDE NUMERIQUE A TROIS AXES	14
I.9 LES MOTEURS PAS A PAS	15
I.9.1 Les différents types de moteur pas à pas.....	15
I.9.1.1 Les moteurs bipolaires	15
I.9.1.2 Les moteurs unipolaires	16
I.9.1.3 Les moteurs à réluctance variable	16
I.9.1.4 Les moteurs à aimant permanant.....	17

I.9.1.4.1 Le principe de fonctionnement.....	17
I.9.1.5 Les moteurs hybrides	18
I.9.1.5.1 Principe de fonctionnement.....	18
I.9.2 Comparaison des trois catégories des moteurs pas à pas	19
I.10 LA COMMANDE DES MOCN	19
I.10.1 La conception 3D	20
I.10.2 La génération des chemins de l'outil, création du G-code	20
I.10.2.1 Langue de contrôleurs numériques « G-code »	20
I.10.3 Interprétation du G-code et transformation en mouvement sur la machine	22
I.10.3.1 Description du micro-logiciel Grbl	22
I.11 LA CONCEPTION ASSISTEE PAR ORDINATEUR (CAO)	22
I.11.1 Définition	22
I.11.2 Les outils de la CAO (logiciels).....	22
I.12 FABRICATION ASSISTEE PAR ORDINATEUR	23
I.12.1 Définition	23
I.12.2 Les outils de la FAO (logiciels)	23
CHAPITRE.2	
II INTRODUCTION	23
II.1. GENERALITES	23
II.2. PRINCIPE ET METHODOLOGIE DE PROGRAMMATION MANUELLE	24
II.2.1. Préparer le dessin de la pièce.....	24
II.2.2. Définir le déroulement des opérations d'usinage	24
II.2.3. Définir la gamme de fabrication	24
II.2.4. Traduire les opérations dans le langage de programmation.....	25
II.2.5. Regrouper toutes les opérations en un programme.....	25
II.3. Constitution et normalisation des codes en programmation manuelle	25
II.3.1. Format d'un mot	26
II.3.2. Format de bloc	28
II.3.3. Structure générale d'un programme	28
II.3.3.1. Structure d'un programme ISO	28
II.4. Programmation ISO	29
II.4.1. Mode de programmation.....	29
II.4.2. Interpolation linéaire.....	31
II.4.3. Interpolation circulaire.....	32
II.4.4. Choix du plan d'interpolation (fraisage).....	33
II.4.5. Commande de la broche	33
II.4.5.1. Rotation.....	33

II.4.5.2. Vitesse.....	34
II.4.6. Mode d'avance.....	35
II.4.7. Programmation des outils	36
II.4.7.1. Appel d'outil et des correcteurs d'outil en tournage [FANUC].....	36
II.4.7.2. Appel d'outil et des correcteurs d'outil en fraisage [SINUMERIK-SIEMENS	37
II.4.8. Correcteur d'outils G41-G42.....	37
II.4.9. Condition d'enchaînement des trajectoires.....	40
II.5. Qualité et rôle du programmeur humain dans les systèmes à commande numérique	41
CHAPITRE.3	
III.1. INTRODUCTION.....	43
III.2. Présentation de la carte Arduino Uno.....	43
III.2.1. Présentation et caractéristiques du module Arduino Uno	43
III.2.2. Description matérielle	45
III.2.2.1. Alimentation	46
III.2.2.2. Connecteurs E/S	47
III.2.3. Description du logiciel	47
III.2.3.1. Logiciel d'édition et programmation Arduino.....	48
III.2.3.2. Structure d'un programme Arduino	49
III.3.1. La commande dir/step	49
III.3.1.1. Circuit L297	49
III.3.1.1.1. Le schéma théorique du montage	50
III.3.1.2 Carte EasyDriver 4.4:	51
III.3.1.2.1 Le schéma du montage Easy Driver.....	51
III.3.1.3. Programmation de La commande dir/step.....	52
III.4. Logiciel de pilotage de la machine CNC.....	53
III.4.1. Modes de commande.....	53
III.4.2. Mode Configuration	55
III.4.3. Mode manuel.....	56
III.4.4. Mode automatique	57
III.5. Langage de programmation la machine CNC	57
III.5.1. Programme Grbl.....	58
III.5.1.1. La communication avec la carte arduino.....	58
III.5.1.1.1. Bibliothèques de la commande des moteurs.....	59
III .6. Organigramme de la commande de la machine CNC	59
CHAPITRE.4	
IV Introduction.....	59
IV.1. Caractéristiques techniques de sélection des MOCN.....	60

IV.1.1. Aspect de la machine	60
IV.1.1. Bâti et structure	60
IV.1.2. Glissières	60
IV.1.3. Les vis-mères	60
IV.1.4. Organes de transmission.....	61
IV.1.5. Les moteurs	61
IV.2.1. Conception de la machine	61
IV.2.2. Moteur pas à pas.....	61
IV.2.3. Palier	61
IV.2.4. Accouplement	62
IV.3. La réalisation de la mocn	63
IV.3.1. BATI	63
IV.3.2. AXE X.....	63
IV.3.3. AXE Z.....	64
IV.3.4. AXE Y.....	65
IV.3.5. BRIDE.....	65
IV.3.6. SUPPORTS DES MOTEURS	66
IV.4. ASSEMBLAGE DE MACHINE	66
IV.4. La réalisation de partie électronique	67
IV.4. 1 L'installation de la carte d'alimentation.....	67
IV.4.3 L'installation des drivers.....	67
IV.5. Suggestions pour améliorer la machine	67
IV.6. Conclusion générale	69

LISTE DES FIGURES

CHAPITRE I : GENERALITE SUR LES MOSN.

Figure I.1- Structure d'une MOCN	3
Figure I.2- Les éléments de la partie opérative	4
Figure I.3- Fonction originale d'une commande numérique	5
Figure I.4- Fraisage à commande numérique	6
Figure I.5- Machine-outil à tour numérique	7
Figure I.6- rectification plane et cylindrique	7
Figure I.8- Electro-érosion	8
Figure I.9 - Fonctionnement en boucle ouverte	10
Figure I.10- Fonctionnement avec commande adaptative	10
Figure I.11- Fonctionnement en boucle fermée	12
Figure I.12- Machine à commande numérique à trois axes.....	12
Figure I.13- Moteur bipolaire.....	13
Figure I.14- Moteur unipolaire.....	13
Figure I.15- Moteur pas à pas à reluctance variable	14
Figure I.16- Moteur pas à pas à aimant permanent	15
Figure I.17- moteur pas à pas hybride	15

CHAPITRE II : METHODOLOGIE DE PROGRAMMATION MANUELLE

Figure II.1. Méthodes comparées de programmation.	21
Figure II.2. Structure et contenu d'un programme CN.	23
Figure II.3. Format d'un mot.	24
Figure II.4. Bloc de chiffres.	24
Figure II.5. Format de bloc.	26
Figure II.6. Structure d'un programme ISO.	27
Figure II.7. Mode de programmation.	28
Figure II.8. Les fonctions de programmation de G0 et G1.	29
Figure II.9. Interpolation circulaire.	30
Figure II.10. Les fonctions de programmation de G02 et G03.	31

Figure II.11. Choix du plan d'interpolation (fraisage).	31
Figure II.12. Rotation de la broche.	31
Figure II.13. Vitesse de la broche.	32
Figure II.14. Mode d'avance.	33
Figure II.15. Correcteur d'outils G41-G42.	36
Figure II.16. Exemple de programmation avec correction de trajectoire de rayon d'outil.	36
Figure II.17. Condition d'enchaînement des trajectoires en tournage.	38
Figure II.18. Condition d'enchaînement des trajectoires en Fraisage.	39

CHAPITRE III : LA CARTE DE COMMANDE ET L'INTERFACE GRAPHIQUE

Figure III.1 : Module ARDUINO ATMEL UNO.....	42
Figure III.2 : Interface du logiciel Arduino.....	46
Figure III.3 Figure : commande moteur par le couple L297 et L298 arduino.....	48
Figure III.4 : le circuit EasyDriver.....	49
Figure III.5 : commande moteur par Easy Driver.....	50
Figure III.6: de Universal Gcode Sender.....	52
Figure III.7: l'interface de "g code sender" on mode manuel.....	52
Figure III.8: l'interface de "g code sender" choisir le fichier Gcode.....	52
Figure III.9: l'interface de "g code sender" on mode automatique	53
Figure III.10: envoyé fichée G code	54
Figure III.11: de mode manuel.....	54
Figure III.12: de GRBL code.....	56

CHAPITRE IV: CONCEPTION ET REALISATION D'UNE MOCN 3 AXES

Figure IV.1. Moteur pas à pas	59
Figure IV.2. Palier.....	60
Figure IV.3. Accouplement.....	60
Figure IV.4 bâti de la machine.....	61
Figure IV.5 axe X.....	61
Figure IV.6 axe Z.....	62

Figure IV.7 axe Y.....	63
Figure IV.8 brides de fixation.....	63
Figure IV.9 supports des moteurs.....	64
Figure IV.4.10 assemblage de machine.....	64
Figure IV.4.11 la carte d'alimentation.....	65
Figure IV.4. 12 relais avec transistor.....	65

LISTE DES TABLEAUX

Tableau I.1. Représente les différents types de moteurs pas à pas	8
Tableau -I.2- Les principales fonctions du G-code	13
Tableau II.4. feuille de programme.....	13

INTRODUCTION GENERALE

Actuellement, les machines-outils à commande numérique (MOCN) permettent des réponses aux exigences de productivité modernes, vu le contrôle et la maîtrise des coûts qui reste une préoccupation importante, et vu aussi la qualité des pièces obtenues en terme de précision et qualité de l'état de surface. La MOCN et son évolution actuelle représentent aujourd'hui le moyen de production le plus important dans la mise en forme des pièces mécaniques.

Le présent travail est le résultat d'un couplage de trois spécialités différentes, mécanique, électrique et automatique en but de donner une contribution à la réalisation d'une MOCN, cette réalisation a été déroulée au niveau du hall technologique de la faculté de technologie à l'université Mohamed Boudiaf / M'sila.

La présente MOCN peut être utilisée dans différentes disciplines telles que l'usinage à CN, le sculptage sur bois à CN, le découpage plasma des métaux à CN, le découpage oxycoupage à CN même dans la photographie et la gravure.

Ce modeste travail se divise en deux parties essentielles :

- Fabrication et réalisation pratique de la MOCN.
- Réalisation du projet du mémoire.

Le mémoire comporte quatre chapitres structurés comme suit :

Le premier chapitre est consacré pour des généralités sur les MOCN.

Le second chapitre concerne la méthodologie de programmation manuelle.

Le troisième chapitre porte sur la carte de commande et l'interface graphique

Le quatrième chapitre est dédié à l'application

Ce mémoire comporte en outre une conclusion générale.

I.1 INTRODUCTION

Ce chapitre traite des connaissances fondamentales relatives à la commande Numérique et les machines-outils à commande numérique

I.2 HISTORIQUE DES MACHINES-OUTILS [1]

Selon le nouveau Larousse illustré de 1904 : « la machine-outil est une machine dont l'effet final est un outil mis en mouvement, et qui a pour but de remplacer la main de l'homme lors du façonnage des matières ». L'évolution des machines-outils commence il y a plus de cinq mille ans avec les premiers tours à potier.

A travers l'évolution des machines-outils, du tour du potier au tour numérique, apparaissent en filigrane de nombreuses tendances qui ont déterminé l'histoire de l'humanité.

D'abord, l'histoire des techniques qui a été influencée par la succession des types d'énergie utilisée : la force de l'homme (le pied sur la pédale), celle du vent ou de l'eau (au XVIe siècle, grâce à la mécanisation du système d'entraînement, le tour fut couplé à des moulins et autres roues à aubes), la combustion du charbon (la machine à vapeur), l'électricité (mais aussi l'énergie lumineuse, chimique, acoustique jusqu'au faisceau à particule) puis le développement de l'informatique.

Ensuite l'histoire des objets qui a suivi l'évolution des matériaux utilisés, la terre, le bois, le métal puis les matières les plus diverses et les alliages développés pour les besoins de la technologie moderne.

Enfin, l'histoire des machines-outils soulève des questions de société comme la condition du travailleur (de l'ouvrier à la chaîne, esclave de la machine-outil, jusqu'au technicien responsable des tours numériques...) et la condition de l'homme moderne. La production standardisée et les gains de productivité permis par les machines-outils ont en effet pu lui offrir la consommation de masse et le temps libre.

I.3 LA COMMANDE NUMERIQUE

I.3.1 Origines de la commande numérique

La Seconde Guerre Mondiale est à peine terminée que les États-Unis en amorcent une autre, la « guerre froide ». L'effort de recherche/développement en matière d'armement est à nouveau relancé. Dans ce contexte, Parsons Works, se voit confier par l'US Air Force la

mission d'améliorer la productivité des chaînes de fabrication et plus particulièrement du long, complexe et fastidieux processus de fabrication et de contrôle des surfaces gauches des pales d'hélicoptère. John T. Parsons, aidé de son ingénieur en chef Frank Stulen mènent les études. Ils imaginent différents concepts dont l'un consiste à motoriser les axes des machines concernées. Se pose alors le problème du pilotage de ces moteurs. Ce point les amène à faire appel aux moyens et compétences tout d'abord d'une de leurs filiales, La Parson Corp, spécialisée dans les solutions usant de calculateurs (on appelait ainsi l'ancêtre de l'ordinateur) alimentés en données par des bandes perforées et, d'autre part, à consulter le M.I.T (Massachusetts Institute of Technology). Finalité : commander par un calculateur (IBM y participera) le déplacement d'une table sur glissières mue par une vis à billes couplée à un moteur électrique. En 1952, le Dr Richard KEGG (Milacron) et le MIT (concepteur des servo contrôleurs) présentent au gratin de l'aéronautique le résultat prometteur de leurs travaux sur une fraiseuse Cincinnati Milacron Hydrotel. Une demande de brevet est aussitôt déposée par précaution au bureau des brevets américain (USPTO) par Parson et Stulen, qui l'accepte sous le n° 2,820,187.

Sur la base des succès précédents et après avoir travaillé pendant cinq ans à industrialiser le concept (affinage des tout premiers concepts de programmation de trajectoires avec interpolation axiale), Parson et Stulen confirment le brevet le 14 janvier 1958 par un brevet intitulé « Motor Controlled Apparatus for Positioning Machine Tool » qui sonne la naissance commerciale de cette technologie. General Electric sera historiquement le premier fabricant industriel de commandes numériques, rejoint en 1962 par son voisin japonais, FANUC. En 1964, les États-Unis comptaient 35 000 machines à commande numérique [2].

I.3.2 Principe de la commande numérique

La commande numérique a pour tâche de générer des mouvements d'axes. Ces instructions qui viennent du programme pièce ou de l'opérateur machine consisteront en une position et une vitesse de déplacement. Le processeur de la commande numérique va alors générer une consigne afin qu'on puisse commander les moteurs d'axes. Elle aura également pour tâche, lors du déplacement des axes, de vérifier la position de ces derniers et dans certain cas, la vitesse de déplacement. Les commandes numériques actuelles sont capables d'effectuer des mouvements en combinant simultanément les positions sur les axes X, Y et Z. Elles s'appellent commande numérique par calculateur. De plus, elles sont capables de gérer plusieurs axes simultanément. On parle alors d'interpolation. Cette dernière peut être linéaire

ou circulaire selon la consigne générée par le microcontrôleur de la partie commande. Le rôle de ce processeur est d'interpréter un code machine que l'on appelle un code G, puis de générer les signaux de commande des axes et de la broche [3].

I.4 LA STRUCTURE D'UNE MACHINE A COMMANDE NUMERIQUE

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 calculateur. Elle est appelée CNC (Commande Numérique par Calculateur). La plupart des MOCN sont des CNC [4].

La machine-outil à commande numérique forme un ensemble comprenant : une partie commande et une partie opérative, tel qu'elle est représentée dans la figure -I.1.

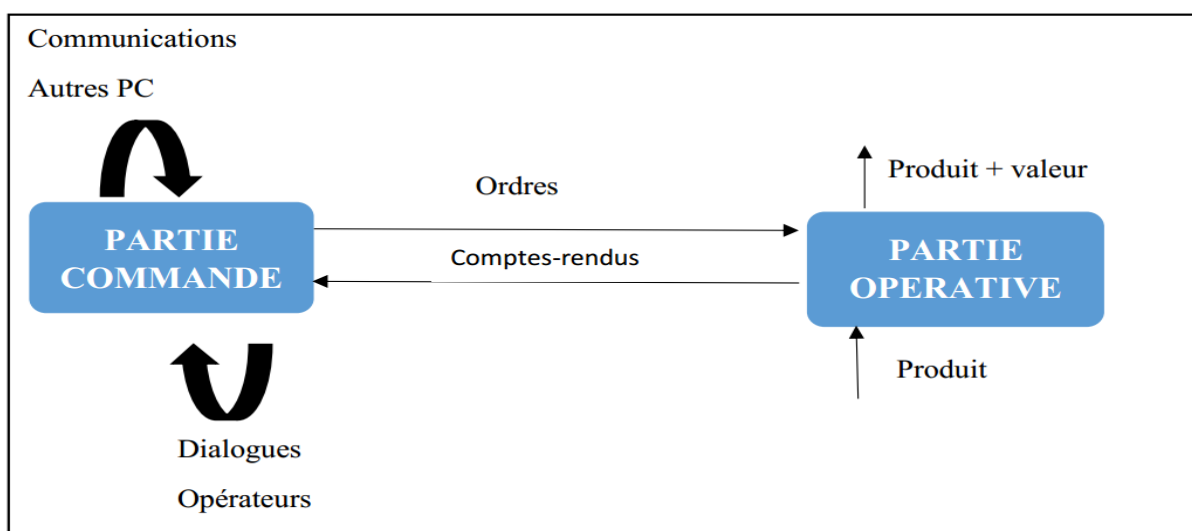


Figure -I.1- Structure d'une MOCN.

I.4.1 Partie opérative

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

- Un socle, très souvent en béton hydraulique vibré, assurant l'indépendance de la machine au sol,
- Un bâti, un banc, 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. Le granit, ou le granit reconstitué, est utilisé pour la fabrication des tables et des bâtis des machines à mesurer tridimensionnelles des rectifieuses et de certains tours,
- 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. Les éléments de la partie opérative sont représentés dans la figure –I.2-.

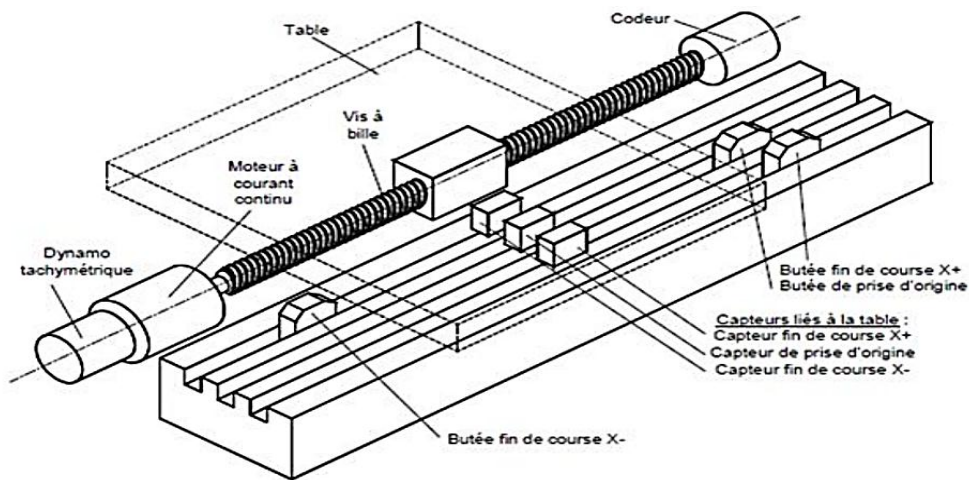


Figure-I.2- Les éléments de la partie opérative.

I.4.1.1 Tâches effectuées

Les tâches effectuées sur le site de la partie opérative sont :

- Chargement et déchargement (pièce port pièce).
- Chargement et déchargement (outils port outils).
- Intervention manuelles nécessitées par l'usinage et l'entretien.
- Surveillance de commande.

I.4.1.2 Armoire électrique de relaying ou armoire de puissance

Elle est composée :

- D'un automate programmable gérant toutes les entrées - sorties,
- D'un relais,

- D'électrovannes,
- De cartes variateurs d'axes (une par axe),
- De contacteurs (1 par élément de machine : axes, broche)
- D'un interrupteur général avec sécurité.

I.4.2 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. La fonction originale de cette dernière est représentée dans la figure –I.3-.

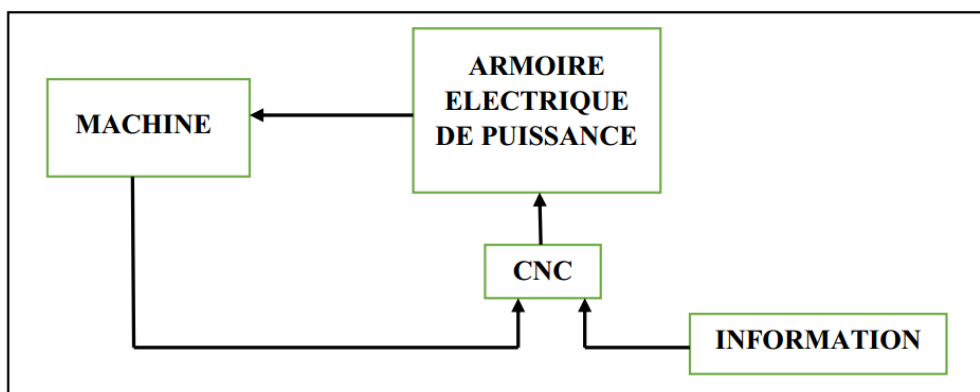


Figure-I.3- Fonction originale d'une commande numérique

I.5. PRINCIPE DE FONCTIONNEMENT D'UNE MACHINE A COMMANDE NUMERIQUE

Les programmes d'usinage sont réalisés à partir d'une origine appelée « origine programme » (OP) positionnée par le programmeur. Le programme commande les déplacements relatifs entre le brut et les outils dans le but de réaliser l'usinage de la pièce

finale. Ces déplacements sont réalisés dans un repère orthonormé normalisé (O, X, Y, Z) basé sur la structure de la machine.

L'axe (Z) de ce repère est un axe confondu avec celui de la broche de la machine — axe de rotation de la fraise en fraisage, axe de rotation de la pièce en tournage. Le sens positif de cet axe est donné par le sens d'éloignement de l'outil par rapport à la pièce. La détermination de **l'axe (X)** entre les 2 axes restants se fait en identifiant celui qui permet le plus grand déplacement. Le sens positif de (X) est déterminé par le sens logique d'éloignement de l'outil par rapport à la pièce.

L'axe(Y) est déterminé à partir de (X) et (Z) grâce à la règle du trièdre direct [5].

I.6 LES FAMILLES DES MACHINES A COMMANDE NUMERIQUE

On peut aussi découper la famille des commandes numériques en quatre sous familles de machines :

I.6.1 Fraisage à commande numérique (FCN)

Le fraisage est un procédé de fabrication où l'enlèvement de matière sous forme de copeaux résulte de la combinaison de deux mouvements : la rotation de l'outil de coupe, d'une part, et l'avancée de la pièce à usiner d'autre part. Le fraisage est habituellement réalisé par une machine-outil, la fraiseuse qui est particulièrement adaptée à l'usinage de pièces prismatiques et permet également, si la machine est équipée de commande numérique, de réaliser tous types de formes même complexes. L'outil classiquement utilisé est la fraise [6]. La figure –I.4- présente un exemple de fraisage numérique.

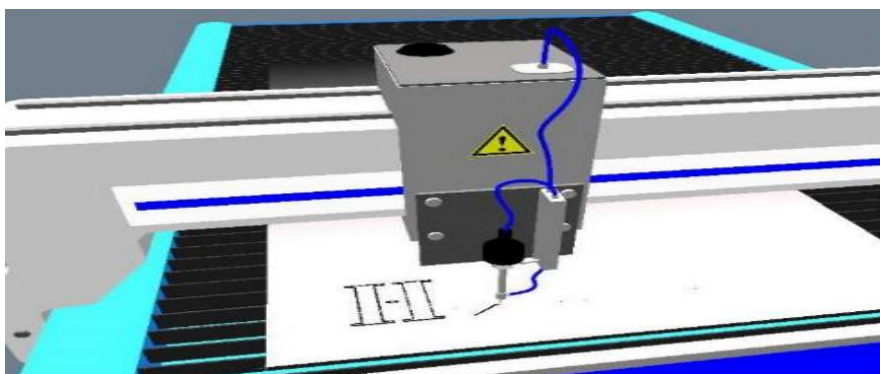


Figure-I.4- Fraisage à commande numérique.

I.6.2 Tournage à commande numérique (TCN)

Le tournage est une technique d'usinage qui consiste à enlever, à l'aide d'outils coupants, de la matière sur une pièce initiale cylindrique pour obtenir une pièce finale (figure-I.5). La matière est enlevée par la combinaison de la rotation de la pièce usinée et du mouvement de l'outil coupant [8].

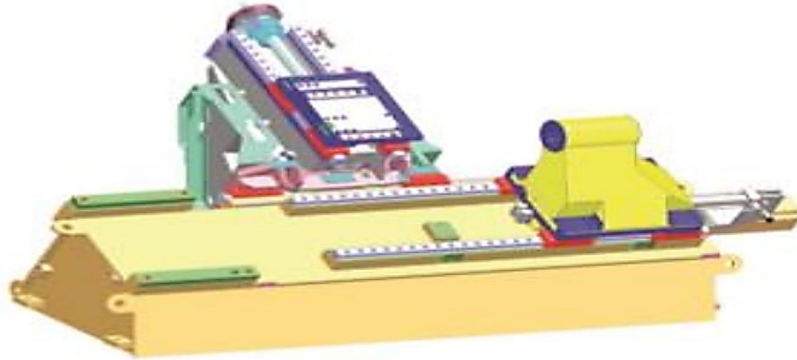


Figure -I.5- tour à commande numérique.

I.6.3 Rectification à commande numérique

La rectification d'une pièce mécanique est une opération destinée à améliorer son état de surface. Les deux techniques principales sont la rectification plane et la rectification cylindrique (figure -I.6-). La rectification s'effectue sur une machine-outil conçue à cet effet la rectifieuse. Il s'agit de rectifier, c'est-à-dire d'approcher une surface d'une forme parfaite (en général : plan, cylindre de révolution ou cône). La rectification est souvent utilisée dans le but de préparer des surfaces de frottement, par exemple, la portée d'un arbre qui tournera dans un palier lisse ou dans un joint d'étanchéité. Elle peut également être utilisée pour donner un profil particulier à la pièce lorsque la meule a été au préalable usinée au profil complémentaire [10].

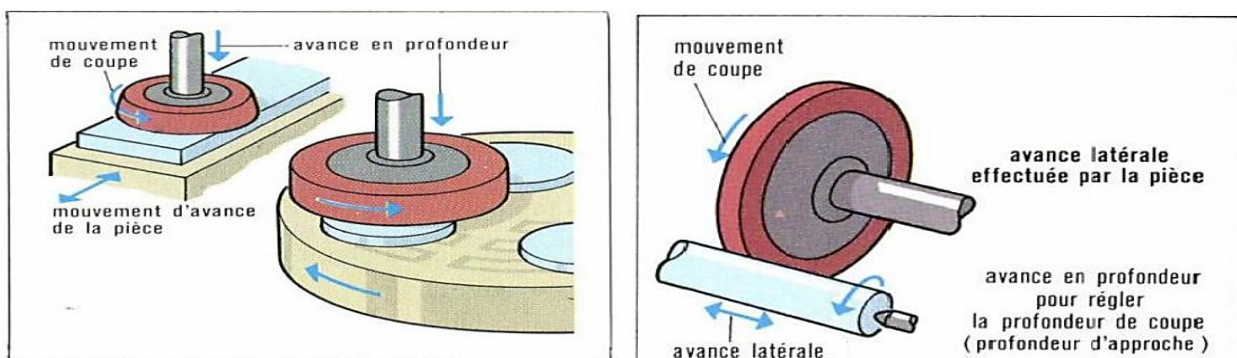


Figure-I.6- rectification plane et cylindrique.

I.6.4 Electro-érosion à commande numérique

L'électroérosion, appelée aussi EDM (electrical discharge machining), est un procédé d'usinage qui consiste à enlever de la matière dans une pièce en utilisant des décharges électriques.

On parle aussi d'usinage par étincelage. Cette technique se caractérise par son aptitude à usiner tous les matériaux conducteurs de l'électricité (métaux, alliages, carbures, graphites, etc.) quelle que soit leur dureté. La figure -I.7- représente un dessin d'une machine d'électroérosion à commande numérique [12].

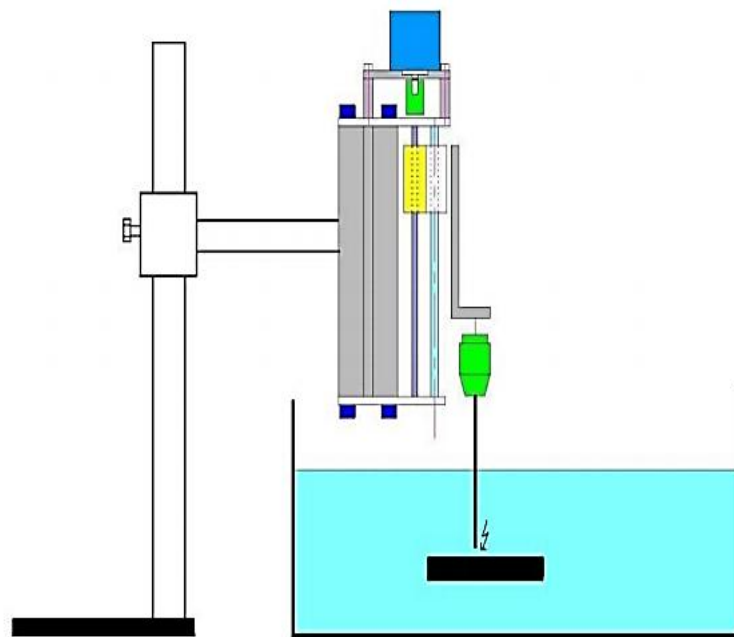


Figure-I.7- Electro-érosion.

I.7 CLASSIFICATION DES MOCN [14]

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

- Le mode de fonctionnement de la machine
- Le nombre d'axes de la machine
- Le mode d'usinage
- Le mode de fonctionnement du système de mesure
- Le mode d'entrée des informations

Les machines-outils à commande numérique (MOCN) peuvent être assistées d'une programmation extérieure et de mécanismes tendant à les rendre encore plus performantes, tels que :

- Ordinateur et ses périphériques.
- Commande adaptative.
- Préréglage des outils.
- Codage des outils.
- Chargeur d'outils et magasin.
- Chargeur et convoyeur de pièces.
- Combinaison de type d'usinages (centre de tournage, centre d'usinage).
- Table de montage.
- Évacuateur de copeaux.
- Dispositifs de contrôle de pièces.

I.7.1 Classification des MOCN selon le mode de fonctionnement

Les MOCN sont classifiées selon les modes de fonctionnement suivants :

I.7.1.1 Fonctionnement en boucle ouverte

En boucle ouverte (figure –I.8-), le système assure le déplacement du chariot mais ne le contrôle pas

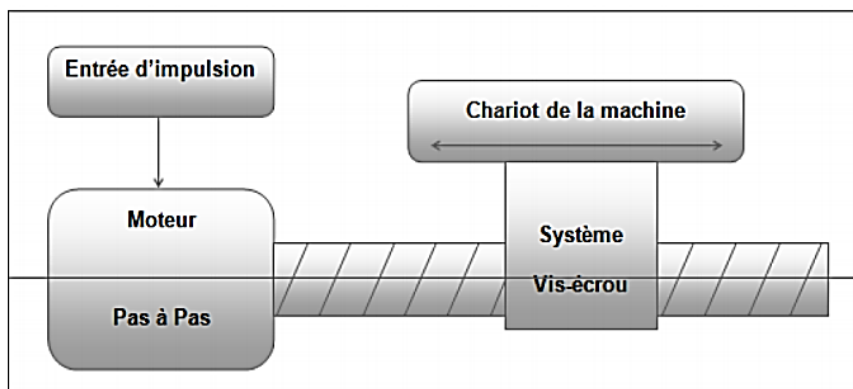


Figure-I.8- Fonctionnement en boucle ouverte

I.7.1.3 Fonctionnement en boucle fermée

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).

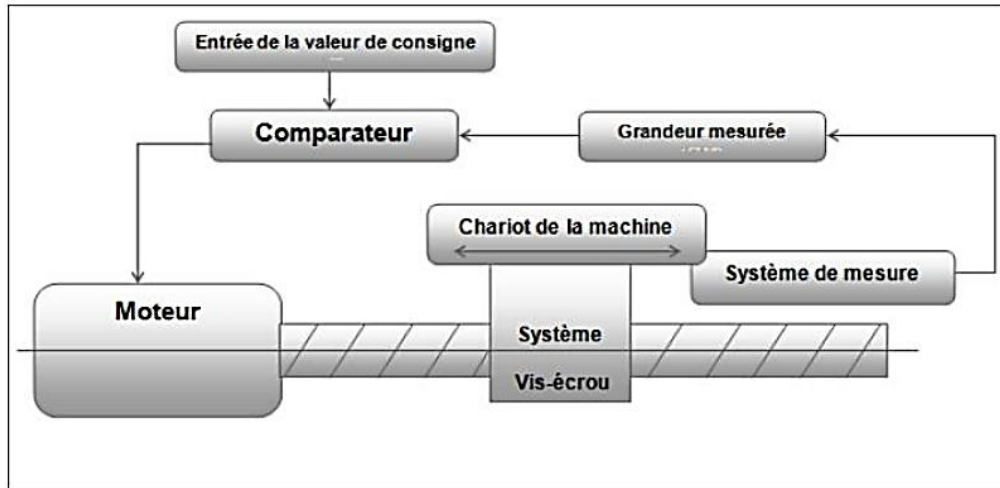


Figure -I.9- Fonctionnement en boucle fermée

I.7.1.2 Fonctionnement avec commande adaptative

La commande adaptative (figure -I.10-) 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.

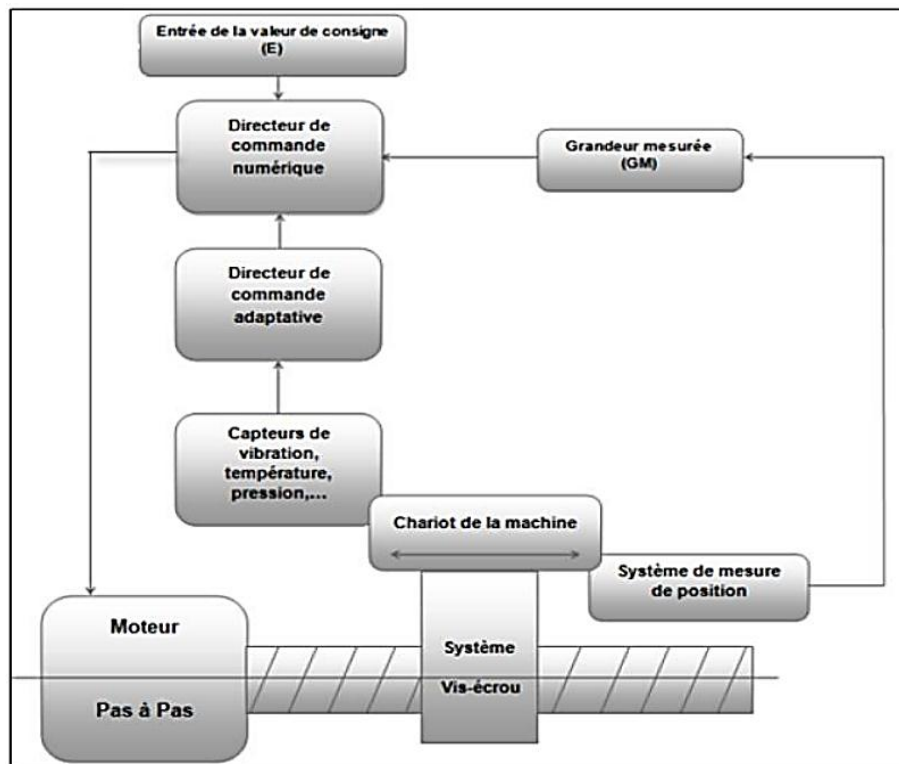


Figure -I.10- Fonctionnement avec commande adaptative

I.7.2 Classification des MOCN selon le nombre d'axe

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

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

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

I.7.3 Classification des MOCN selon le mode d'usinage

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

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

I.8 LA MACHINE A COMMANDE NUMERIQUE A TROIS AXES

L'usinage trois (3) axes est l'une des techniques les plus employées dans la fabrication de pièces mécaniques. Tel que les machines-outils classiques comme la fraiseuse qui permet un travail de la matière sur trois (3) axes (X, Y et Z) (figure –I.11-).

La machine-outil procède alors à l'enlèvement des copeaux suivant trois directions de base correspondant aux axes d'une surface plane. Tout à fait adaptée aux pièces peu profondes. Cependant cette technique a de grosses limites quand il s'agit de traiter une pièce profonde avec des cavités étroites. Pour cela de nouvelles technologies d'usinage ont été développé tel que 4 axes [16].

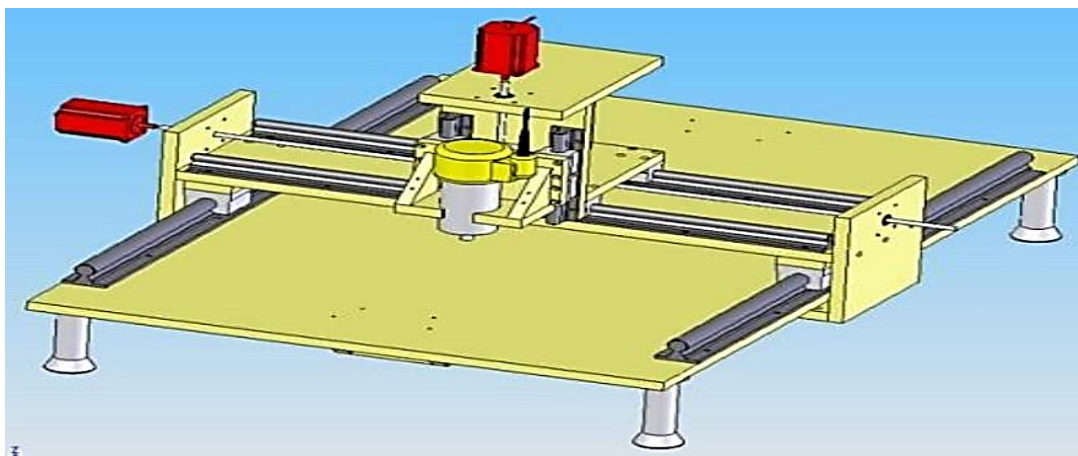


Figure -I.11- Machine à commande numérique à trois axes

I.9 LES MOTEURS PAS A PAS

Le moteur pas à pas est un convertisseur électromécanique qui assure la transformation d'un signal électrique impulsionnel en un déplacement mécanique (angulaire ou linéaire). Sa structure de base se présente sous la forme de deux pièces séparées mécaniquement, le Stator et le Rotor. L'interaction électromagnétique entre ces deux parties assure la rotation.

I.9.1 Les différents types de moteur pas à pas

Les moteurs pas à pas sont très utilisés dans toutes les applications mécaniques où l'on doit contrôler simplement la position ou la vitesse d'un système en boucle ouverte. Ces moteurs sont par exemple utilisés dans les imprimantes jet d'encre ou laser, pour positionner les têtes d'impression ou pour l'avancée du papier.

Il existe 3 types de moteurs pas à pas, à aimants permanents, à réluctance variable ou hybrides. On peut les classer aussi en deux types selon l'alimentation de bobinage, les moteurs bipolaires, les moteurs unipolaires.

I.9.1.1 Les moteurs bipolaires

Ils ont en général quatre (4) fils et se commandent en inversant le sens du courant dans une des deux bobines, tel que c'est représenté dans la figure -I.12, qui sont alimentés une fois dans un sens, une fois dans l'autre sens. Ils créent une fois un pôle nord, une fois un pôle sud d'où le nom de bipolaire.

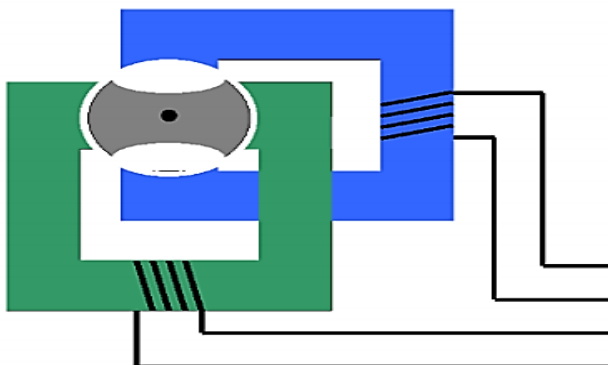


Figure -I.12- Moteur bipolaire

I.9.1.2 Les moteurs unipolaires

Ils ont en général six (6) fils, dont deux sont reliés au milieu des bobines, tel que c'est représenté dans la figure-I.13. Ils se commandent en les alimentant tour à tour, mais les bobinages sont alimentés toujours dans le même sens par une tension unique d'où le nom d'unipolaire.

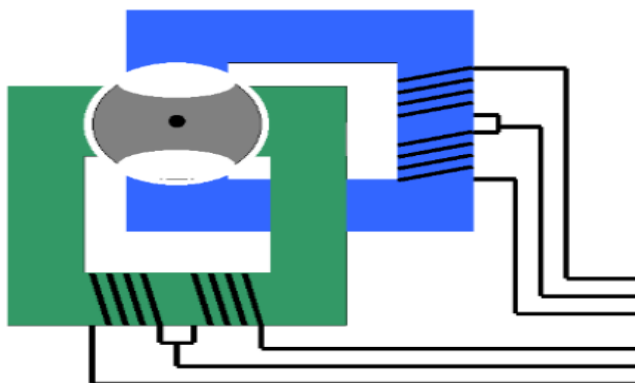


Figure-I.13 - Moteur unipolaire

I.9.1.3 Les moteurs à réluctance variable

Les moteurs à réluctance variable fonctionnent selon un principe différent de celui des moteurs à aimant permanent. Ils possèdent bien évidemment un stator, mais ce dernier est fabriqué en acier doux non magnétique. Il n'est pas lisse et possède plusieurs dents. Le stator est composé de 8 plots sur lesquels enroulés les bobinages, ce qui donne 4 phases. Le rotor, quant à lui, ne comporte que 6 dents [19]

I.9.1.3.1 Le principe de fonctionnement

Le principe de fonctionnement est très simple l'un des phases alimentant deux des plots opposés du stator est traversée par un courant afin d'aligner deux des dents du rotor avec ces plots (figure –I.14). Le mode de commande peut dans ce cas, de la même façon que pour les autres moteurs, être monophasé, biphasé ou demi-pas.

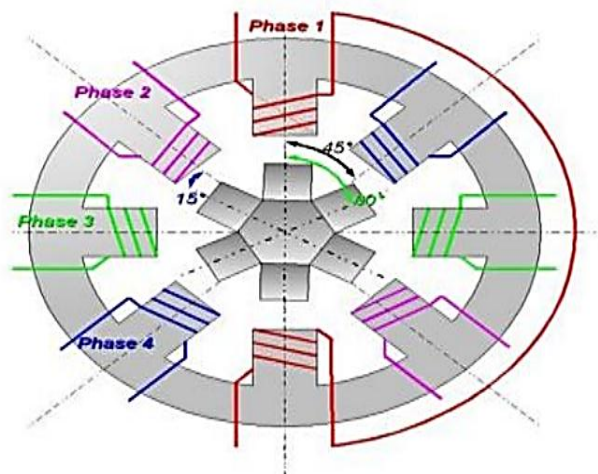


Figure-I.14- Moteur pas à pas à reluctance variable

I.9.1.4 Les moteurs à aimant permanent

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

I.9.1.4.1 Le principe de fonctionnement

C'est le modèle dont le fonctionnement est le plus simple. Le rotor est constitué d'un aimant permanent, et le stator comporte deux paires de bobines (figure –I.15). En agissant sur les bobines alimentées, et le sens des courants, on fait varier le champ créé par le stator. A chaque pas, la direction du champ induit par le stator tourne de 90° . L'aimant permanent suit le déplacement du champ magnétique créé par les bobines et s'oriente selon une de ses quatre positions stables. Comme le rotor est aimanté, lorsque le moteur n'est pas alimenté le flux magnétique dû à l'aimant permanent crée un couple résiduel en se plaçant dans l'axe de l'une des bobines [20].

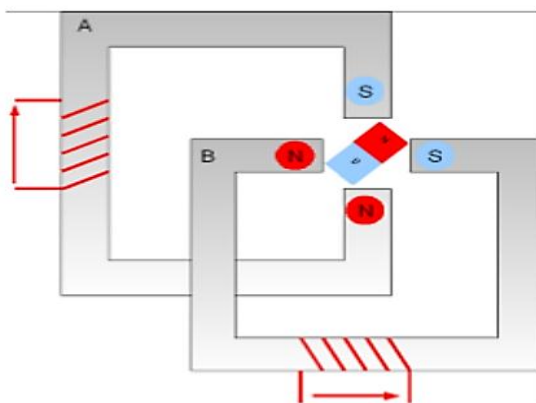


Figure-I.15- Moteur pas à pas à aimant permanent

I.9.1.5 Les moteurs hybrides

Ce sont des moteurs qui superposent le principe de fonctionnement des moteurs à aimant permanent et à reluctance variable et combine leurs avantages. Le rotor est constitué de deux disques dentés décalés mécaniquement. Entre ces deux disques est inséré un aimant permanent.

Leur intérêt réside dans un meilleur couple, une vitesse plus élevée, une résolution de 100 à 400 pas/tour, et sont plus chers [21].

I.9.1.5.1 Principe de fonctionnement

En mode pas entier, les bobines sont alimentées une par une alternativement, dans un sens puis dans l'autre. Dans la figure -I.16, on voit bien l'intérêt d'avoir un rotor polarisé : quand on coupe l'alimentation de la bobine verte et qu'on alimente la bobine rose, en l'absence de polarisation du rotor on n'aurait aucun contrôle sur le sens de rotation, les deux dents bleues et rouges étant attirés de la même façon par la bobine. Avec un rotor polarisé, on peut choisir une de ces deux dents en agissant sur le sens du courant dans la bobine .

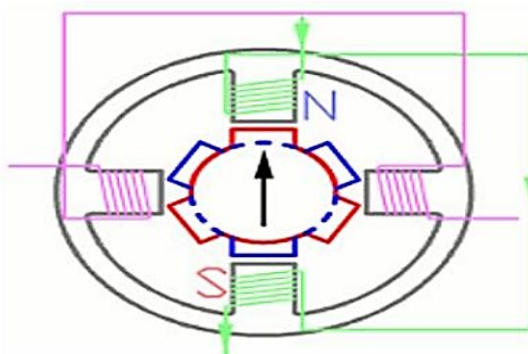


Figure -I.16- moteur pas à pas hybride

I.9.2 Comparaison des trois catégories des moteurs pas à pas

Le tableau –I.1. Représente les différents types de moteurs pas à pas et leurs avantages et inconvénients.

Type de moteur	Moteur à aimant permanent	Moteur hybride	Moteur à reluctance variable
Résolution (Nombre de pas)	Moyenne	Elevée	Bonne
Couple moteur	Elevé	Elevé	Faible
Vitesse	Faible	Elevé	Elevé
Sens de rotation	Dépend : <ul style="list-style-type: none">• De l'ordre d'alimentation des phases.• Du sens de courant dans les bobines.	Dépend : <ul style="list-style-type: none">• De l'ordre d'alimentation des phases.• Du sens de courant dans les bobines.	Dépend : <ul style="list-style-type: none">• De l'ordre d'alimentation des phases.
Fréquence de travail	Faible	Grande	Grande
Coût	Bas	Haut	Moyen/haut

I.10 LA COMMANDE DES MOCN

De l'idée de départ jusqu'à la fabrication de la pièce sur la machine CNC, tout va se passer sur l'ordinateur, utilisant différents logiciels selon l'étape du processus à effectuer, constituant ainsi une chaîne logicielle. Les étapes de création :

- La conception 3D.
- La génération des chemins de l'outil, création du G-code.
- Interprétation du G-code et transformation en mouvement sur la machine.

I.10.1 La conception 3D

La conception permet d'élaborer, dessiner, visualiser un objet en trois (3) dimensions. La modélisation va plus loin, permettant de modéliser des mouvements des objets en relation les uns avec les autres, de simuler des efforts, etc. Autrement dit, tous les outils de conception 3D ne sont pas nécessairement aussi des outils de modélisation.

Il existe plusieurs logiciels de conception et modélisation 3D, tel que le logiciel Solid Works qui est le plus populaire.

I.10.2 La génération des chemins de l'outil, création du G-code

Après la conception en 3D de la pièce, par exemple avec le logiciel SolidWorks, on obtient un fichier de format STL, OBJ, Image ou autre. Pour la génération d'un fichier G-code il existe des logiciels spécialement conçus pour cela, tel que le logiciel Slic3r, Inkscape, Charlygraal et bien d'autre. Aussi, Il est possible de créer un fichier d'usinage directement à partir d'un site spécialement conçu pour cela, tel que Makercam.com.

I.10.2.1 Langue de contrôleurs numériques « G-code »

Le G-code est le langage de programmation pour contrôler une machine à commande numérique. Il est basé sur des lignes de code, plusieurs de ces lignes peuvent être regroupées dans un fichier pour créer un programme G-code, qui sert à programmer les mouvements que la machine va effectuer (droite, arc de cercle, ect). Il est à noter que le G-code est utilisé pour des fraiseuses à commande numérique, des tours, des imprimantes 3D et des lasers de découpe.

a- Les principales fonctions du G-code

Les principales fonctions du G-code sont résumées dans le tableau suivant (Tableau-I.2).

Tableau -I.2- Les principales fonctions du G-code :

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
G94/95	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
M00/M01	Arrêt du programme, arrêt optionnel ou avec condition
M06	Changement d'outil
M30	Fin du programme, réinitialisation, rembobinage

I.10.3 Interprétation du G-code et transformation en mouvement sur la machine

Après la génération d'un fichier G-code, il reste à l'interpréter de façon à pouvoir contrôler les moteurs avec un microcontrôleur. Pour cela il existe des logiciels et des micrologiciels spécialement conçu pour cela, tel que le Grbl.

I.10.3.1 Description du micro-logiciel Grbl

Le Grbl est un micrologiciel libre développé pour contrôler des graveuses CNC (Computer Numerical Control), des fraiseuses munis d'une tête mobile contrôlée en X, Y et Z par un ordinateur. Grbl interprète du G-code et déplace en conséquence un outil sur trois (3) axes (X, Y et Z). Il comprend de multiples optimisations sur l'usage et le déplacement des moteurs afin de gérer correctement les accélérations, les trajectoires, etc .

I.11 LA CONCEPTION ASSISTEE PAR ORDINATEUR (CAO)

I.11.1 Définition

La CAO (Conception Assistée par Ordinateur) est un ensemble d'outils et de programmes informatiques permettant d'assister l'ingénieur dans la conception et la mise au point d'un produit. Un système de CAO permet de représenter et d'étudier le fonctionnement d'un objet sans l'avoir fabriqué réellement, c'est-à-dire en virtuel .

I.11.2 Les outils de la CAO (logiciels)

Un grand nombre de logiciels de CAO sont disponibles en open-source :

- Art of Illusion.

- Blender.
- FreeCAD.
- ImplicitCAD.
- QCAD.
- pythonOCC.
- OpenCASCADE.
- Solidworks.

I.12 FABRICATION ASSISTEE PAR ORDINATEUR

I.12.1 Définition

La fabrication assistée par ordinateur ou FAO est d'écrire le fichier contenant le programme de pilotage d'une machine-outil à commande numérique. Ce fichier va décrire précisément les mouvements que doit exécuter la machine-outil pour réaliser la pièce demandée. On appelle également ce type de fichiers : programme ISO ou blocs ISO.

Les logiciels de FAO utilisent les modèles et les ensembles créés dans les logiciels de CAO pour générer des trajectoires d'usinage sur lesquelles s'appuient les machines qui convertissent les conceptions en pièces physiques [25].

I.12.2 Les outils de la FAO (logiciels)

Les outils de la FAO sont des modeleurs, nous citons :

- Catia.
- RhinosCAM.
- SolidCAM.
- Power MILL.
- Hyper MILL.
- Camworks

INTRODUCTION

La programmation est le travail de préparation qui consiste à transposer la gamme d'usinage de la pièce en un ensemble ordonné d'instructions comprises et exécutées par la CN en vue de réaliser son usinage. Ce travail peut être effectué manuellement ou avec l'assistance d'un ordinateur utilisant un langage de programmation évolué

II.1. GENERALITES

À titre indicatif, la (Figure II.1) classe différentes méthodes de programmation en fonction des compétences du programmeur et de la complexité des machines à piloter

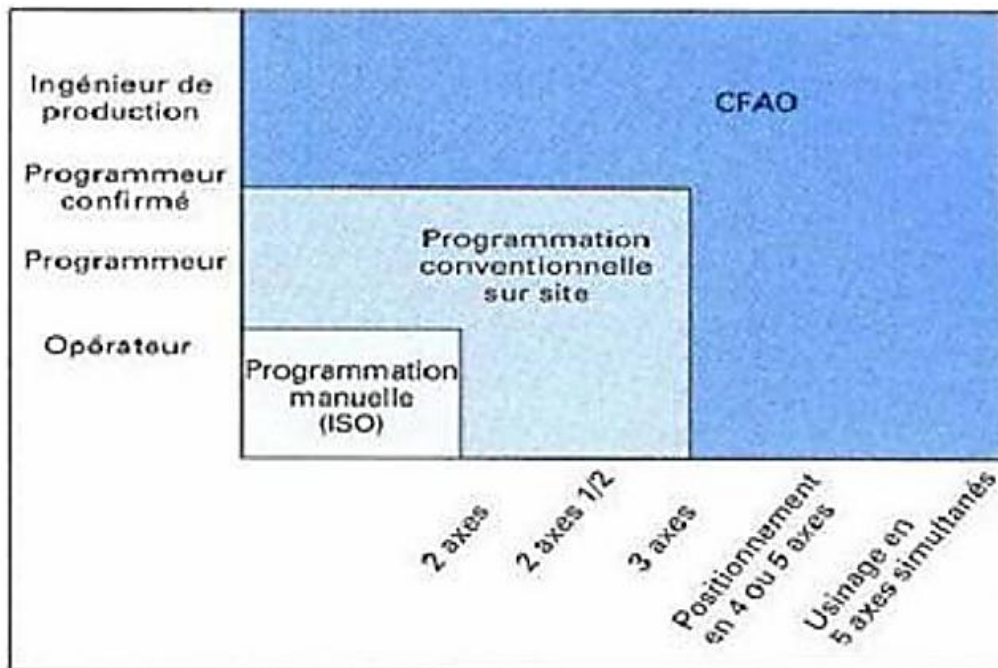


Figure II.1. Méthodes comparées de programmation

La création d'un programme est soumise à des règles de structure, syntaxe ou format.

Les instructions programmées doivent contenir toutes les données nécessaires à la commande et au séquencement des opérations à réaliser pour assurer l'usinage de la pièce sur la machine. Elles regroupent :

- Les données géométriques, qui permettent à la CN de calculer les positions successives de l'outil par rapport à la pièce pendant les diverses phases de l'usinage. Les positions sont définies par rapport à une origine connue.
- Les instructions indiquant le mode d'interpolation, le choix du mode de cotation, absolue ou relative, le choix du cycle d'usinage, le choix de l'outil, etc. ;

- Les données technologiques qui précisent les conditions de coupe optimales dans lesquelles pourra s'effectuer l'usinage. Elles concernent principalement la vitesse de rotation de la broche, les vitesses d'avance et la commande de l'arrosage.

II.2. PRINCIPE ET METHODOLOGIE DE PROGRAMMATION MANUELLE

Avant toute programmation proprement dite, il est important de planifier et de préparer méticuleusement les opérations d'usinage. Plus votre préparation aura été précise quant à la structure de votre programme CN, plus la programmation proprement dite sera simple et rapide et moins vous aurez d'erreurs dans le programme terminé. [5] [6]

II.2.1. Préparer le dessin de la pièce

- Définir l'origine de la pièce.
- Indiquer le système de coordonnées.
- Eventuellement calculer les coordonnées manquantes.

II.2.2. Définir le déroulement des opérations d'usinage

- Quels sont les outils à mettre en œuvre, à quel moment et pour le traitement de quel contour?
- Dans quel ordre les différents éléments de la pièce devront-ils être usinés ?
- Quels sont les éléments qui se répètent et qui devraient figurer dans un sous-programme ?
- Est-ce que d'autres programmes pièce ou sous-programmes contiennent des contours susceptibles d'être utilisés pour la pièce actuelle ?

II.2.3. Définir la gamme de fabrication

Définir pas à pas toutes les phases d'opération de la machine, par exemple :

- Déplacements à vitesse rapide pour le positionnement.
- Changement d'outil.
- Définition du plan d'usinage.
- Dégagement pour les mesures.
- Mise en marche / arrêt de la broche, de l'arrosage.
- Appel des données d'outil.
- Approche de l'outil.
- Correction de trajectoire.
- Accostage du contour.

- Retrait de l'outil.

II.2.4. Traduire les opérations dans le langage de programmation

Transcrire chaque opération sous la forme d'un bloc CN (ou de blocs CN).

II.2.5. Regrouper toutes les opérations en un programme

Le regroupement des opérations doit obéir aux règles de programmation.

II.3. Constitution et normalisation des codes en programmation manuelle

Un programme CN se compose d'une suite de blocs CN. Chaque bloc contient les données pour l'exécution d'une opération d'usinage. Il est divisé en 3 domaines : En-tête de programme, corps de programme et fin de programme. Ensemble, ces domaines constituent la gamme d'usinage.

Les blocs CN sont formés des composantes suivantes :

- Instructions selon un langage de programmation (exemple : DIN 66025)
- Éléments du langage évolué CN

Les instructions sont constituées d'un symbole d'adresse et d'un chiffre ou d'une suite de chiffres qui décrit une valeur arithmétique. [5] [6]

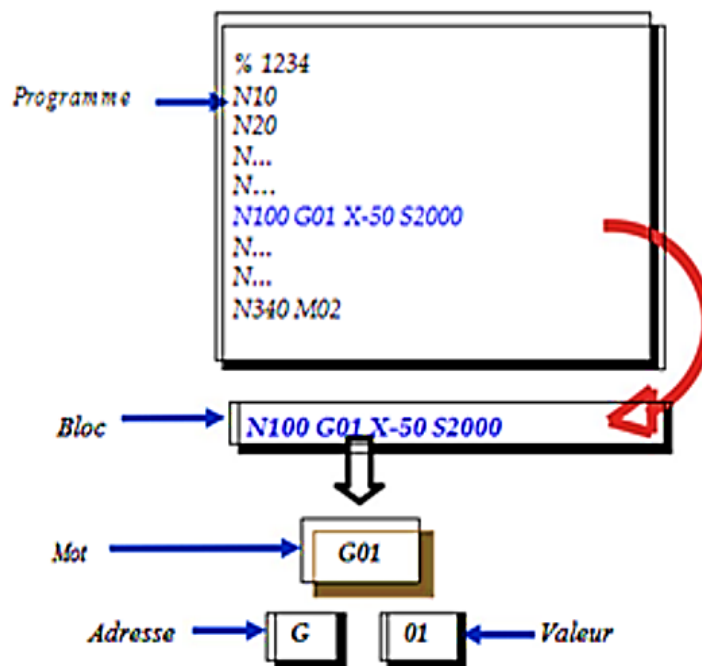


Figure II.2. Structure et contenu d'un programme CN

II.3.1. Format d'un mot

Le mot définit une instruction ou donnée à transmettre au système de commande.

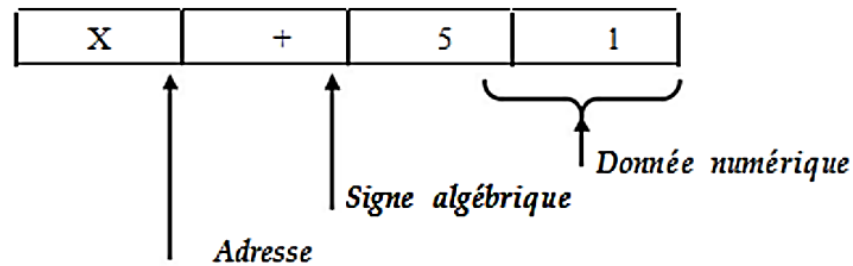


Figure II.3. Format d'un mot.

Le symbole d'adresse (généralement une lettre alphabétique) définit la signification de l'instruction. Exemples :

Symbole d'adresse	Signification
G	Fonction G (fonction préparatoire)
X	Information de déplacement pour l'axe X
S	Vitesse de rotation de broche

La suite de chiffre est la valeur affectée au symbole d'adresse. Elle peut contenir un signe et un point décimal, le signe étant toujours placé entre la lettre adresse et la suite de chiffres. Les signes positifs (+) et les zéros de tête (0) n'ont pas besoin d'être écrits.

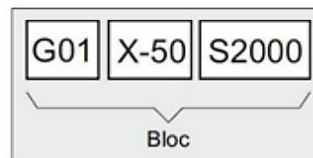


Figure II.4. Bloc de chiffres.

Dans le cas de la programmation des opérations d'usinage complexes des machines-outils modernes, il a été complété par les éléments du langage évolué CN. Il s'agit, entre autres, des éléments suivants :

- Les instructions du langage évolué CN se composent de plusieurs lettres d'adresse. Par exemple :
- OVR pour la correction de vitesse de rotation.
- SPOS pour le positionnement de broche.

Descripteurs (noms définis) pour :

- Variables système cas de "Programmation CN flexible ou paramétrée".
- Variables définies par l'utilisateur cas de "Programmation CN flexible ou paramétrée".

- Sous-programmes. Etc....

Les instructions peuvent être modales ou non modales :

Modal :

Les instructions à effet modal restent valides avec la valeur programmée au-delà du bloc dans lequel elles sont programmées, jusqu'à :

- La programmation d'une nouvelle valeur sous la même instruction
- La programmation d'une instruction qui annule la validité de l'instruction qui était valide jusque-là

Non modal.

- Les instructions à effet non modal sont valides uniquement dans le bloc dans lequel elles ont été programmées.
- Le dernier bloc des séquences d'exécution contient un mot spécifique pour la fin du programme : M2, M17 ou M30.

Les règles applicables pour affecter des valeurs aux adresses sont les suivantes :

- Il faut écrire le caractère "=" entre l'adresse et la valeur qui suit si :
 - L'adresse se compose de plusieurs lettres,
 - La valeur se compose de plusieurs constantes.
- On peut omettre le signe "=" si l'adresse est constituée d'une seule lettre et la valeur à affecter d'une seule constante.
 - Les signes sont autorisés.
 - Les caractères de séparation sont permis après les lettres de l'adresse. [5]

Exemples :

X10 : Affectation d'une valeur (10) à l'adresse X, "=" n'est pas requis.

X1=10 : Affectation d'une valeur (10) à l'adresse (X) avec extension numérique (1), "=" requis.

X=10*(5+SIN (37.5)) : Affectation d'une valeur par le biais d'une expression numérique, "=" requis.

II.3.2. Format de bloc

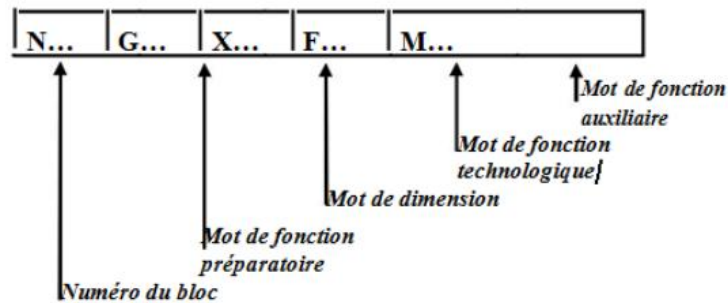


Figure II.5. Format de bloc.

La numérotation n'intervient pas dans l'ordre de déroulement du programme. Il est malgré tout conseillé de numéroter les blocs dans l'ordre d'écriture.

Dans des cas particulier, un bloc se termine par le caractère (;)(EOB = End Of Bloc) ou LF (LINE FEED = nouvelle ligne).

Afin d'obtenir une structure de bloc claire, il est conseillé de placer les instructions d'un bloc dans l'ordre suivant :

N... G... X... Y... Z... F... S... T... D... M... H...

Pour qu'un programme CN soit plus compréhensible, il est possible d'ajouter des commentaires aux blocs CN. Un commentaire se situe à la fin d'un bloc et est séparé de la partie programme du bloc CN par un point-virgule (";"). [5]

II.3.3. Structure générale d'un programme

Un programme est exécuté dans l'ordre d'écriture des blocs situés entre les caractères de début et de fin de programme.

II.3.3.1. Structure d'un programme ISO

La programmation structurée permet d'analyser et de concevoir plus rapidement un programme de commande numérique. En effet, toutes les opérations d'usinage font appel à un certain nombre de fonctions identiques (appel d'outil, rotation de broche, mise en route et arrêt de la lubrification, conditions de coupe, dégagement d'outil, etc.). [6]

Il est donc intéressant de choisir une structure de programmation commune pour toutes ces opérations, valable quelle que soit la machine utilisée et indépendante de la pièce à obtenir.

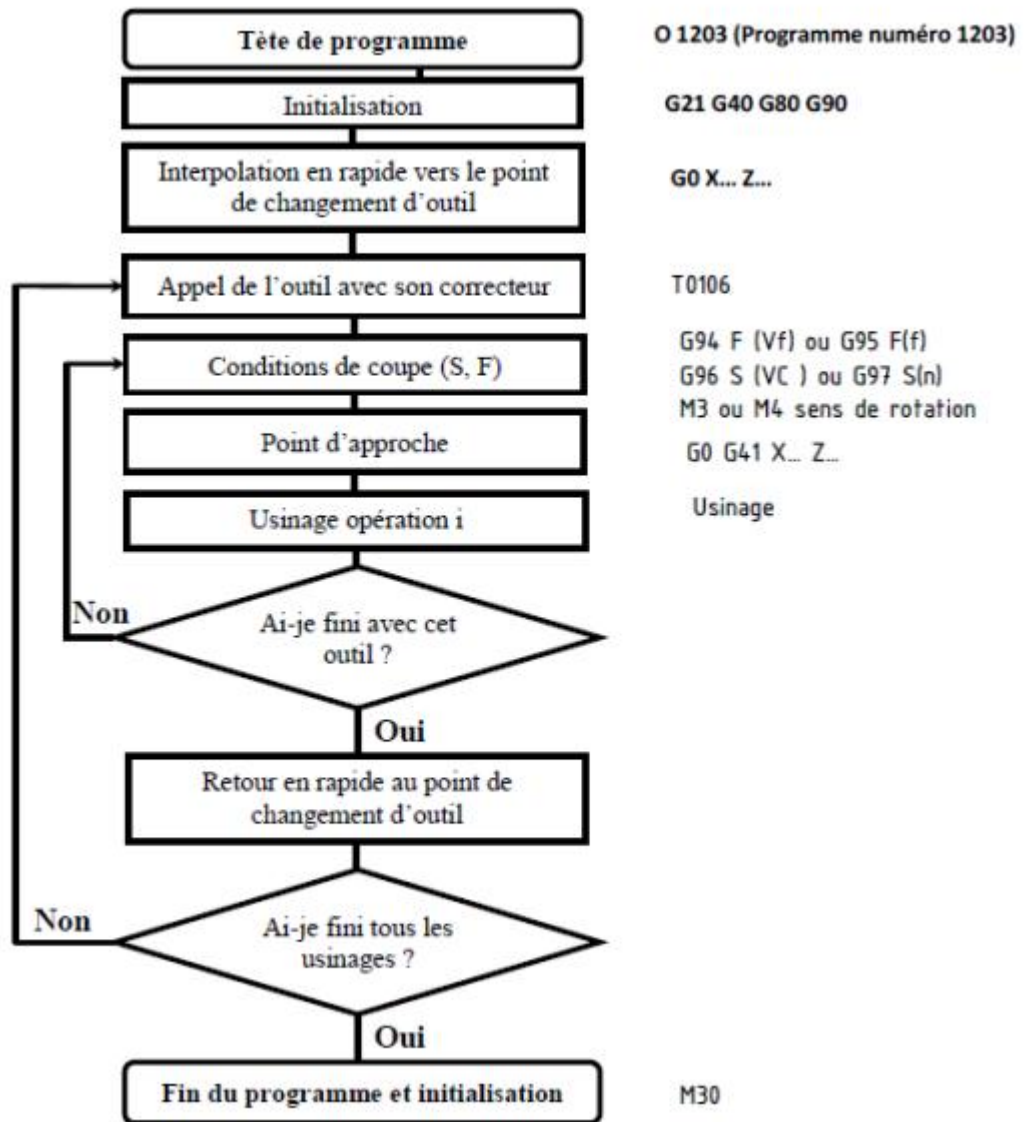


Figure II.6. Structure d'un programme ISO.

II.4. Programmation ISO

II.4.1. Mode de programmation

En programmation absolue, la cotation se réfère à l'origine du système de coordonnées après décalage total.

En programmation relative, la valeur numérique programmée de l'information de déplacement correspond à la distance à parcourir. Le signe indique le sens de déplacement.[7]

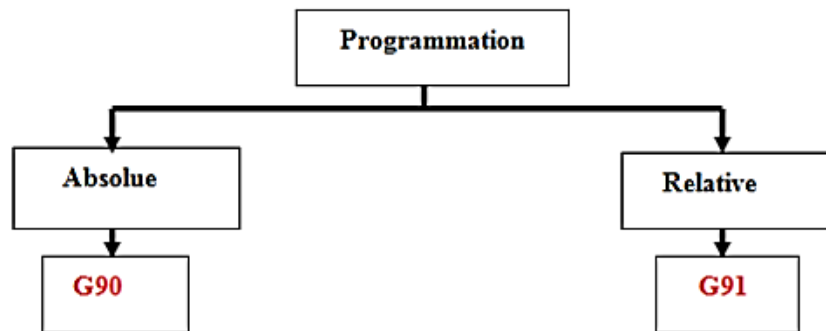


Figure II.7. Mode de programmation.

Syntaxe : N.. G90 /G91 X..

G90 : Programmation absolue par rapport à OP.

G91 : Programmation relative par rapport à OP.

Propriétés :

G90 et G91 sont deux fonctions modales. G90 est initialisée à la mise sous tension.

Révocation : Les fonctions G90 et G91 se révoquent mutuellement. [7]

Cas de tournage :

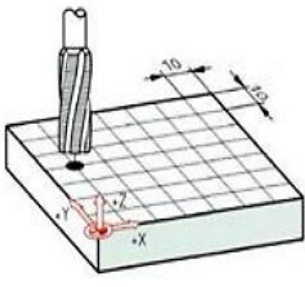
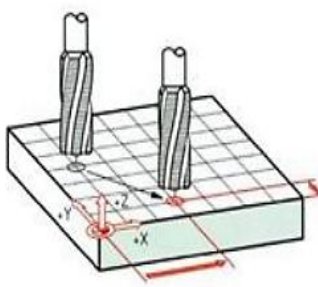
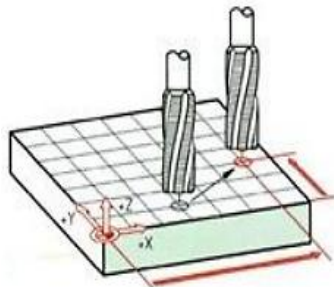
Tableau II.2. Mode de programmation d'un tournage. [6]

En G90	Position actuelle : X40 Z0	Position X50 Z-40	Position Z1
En G91		X5 mm sens (+) Z40 mm sens (-)	Z41 mm sens (+) Z41

		X5 Z-40	
--	--	---------	--

Cas de Fraisage :

Tableau II.3. Mode de programmation d'un fraisage. [6]

			
En G90	Position actuelle : X10 Y30	Position X30 Y10	Position X60 Y30
En G91		X20 mm sens (+) Y30 mm sens (-) X20 Y-30	X30 mm sens (+) Y20 mm sens (+) X30 Y20

II.4.2. Interpolation linéaire

Syntaxe : N... [G90 / G91] G00 / G01 X.. Y... Z..

G00 : Interpolation linéaire à vitesse rapides.

G01 : interpolation linéaire à vitesse programmée. [7]

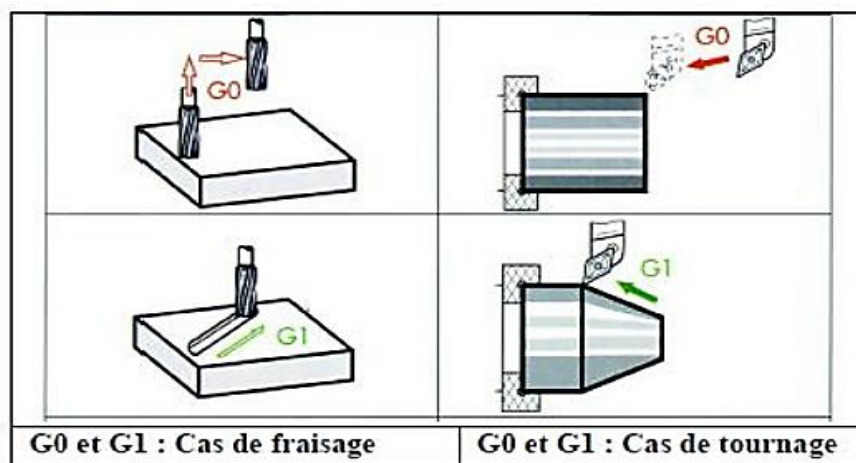


Figure II.8. Les fonctions de programmation de G0 et G1

II.4.3. Interpolation circulaire

Syntaxe : N.. [G90/G91] G02/G03 X.. Y.. I..J../R..[F..].

G02 : Interpolation circulaire sens horaire.

G03 : Interpolation circulaire sens antihoraire.

X, Y : Coordonnées du point d'arrivée de l'arc.

I, J : Coordonnées du point central.

I : suivant X.

J : suivant Y (Dans la plupart des cas en relatif).

Propriétés : G02 et G03 sont deux fonctions modales.

Révocation :

La fonction G02 est révoquée par G00, G01, G03.

La fonction G03 est révoquée par G00, G01, G02. [7]

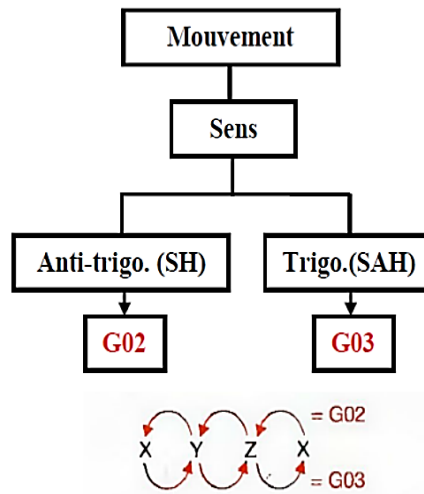


Figure II.9. Interpolation circulaire.

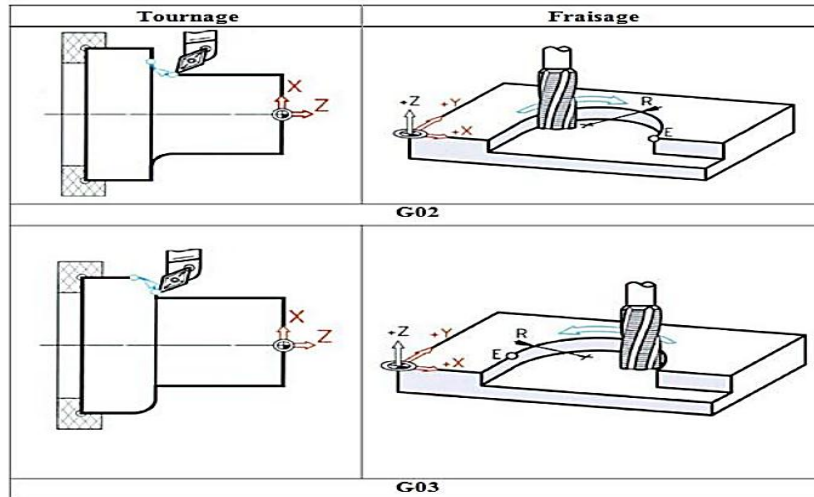


Figure II.10. Les fonctions de programmation de G02 et G03.

II.4.4. Choix du plan d'interpolation (fraisage)

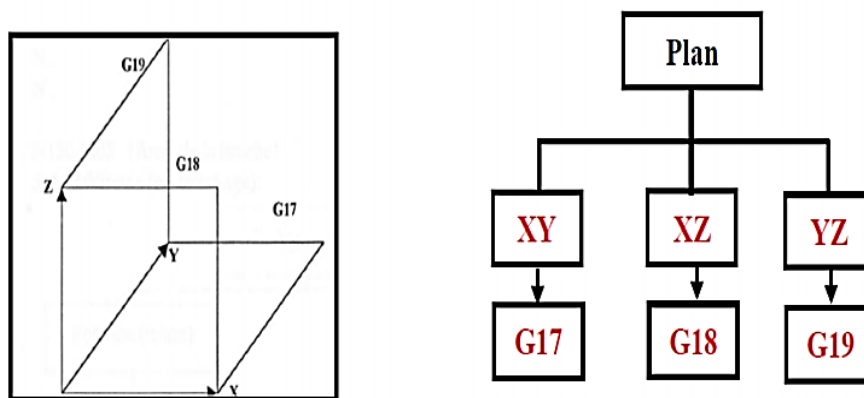


Figure II.11. Choix du plan d'interpolation (fraisage).

Remarque : Si le plan n'est pas précisé G17 est retenu par défaut par le système. [7]

II.4.5. Commande de la broche

II.4.5.1. Rotation

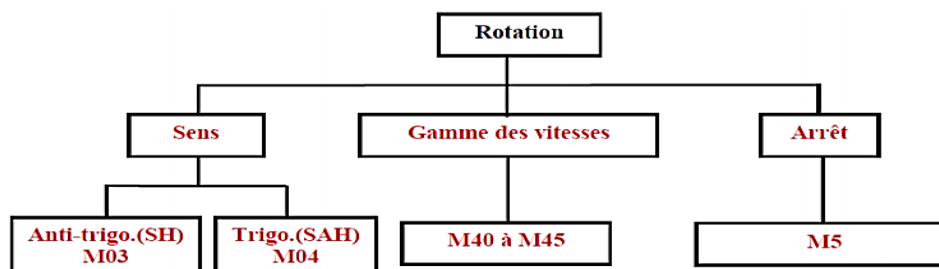


Figure II.12. Rotation de la broche.

Syntaxe : **N.. M03/M04/M05**

Propriétés :

- Les fonctions M03 et M04 sont modales « avant ».
- La fonction M05 est une fonction modale « après » initialisée à la mise sous tension.
- Les fonctions M40 à M45 sont des fonctions modales « avant ».

Révocation : Les fonctions M03, M04 et M05 se révoquent mutuellement. Les fonctions M00, M19 et M01 révoquent l'état M03 ou M04. [7]

Exemple :

N50 S500 M3 M41 (Rotation sens anti-trigonométrique, gamme M41)

N150 M05 (Arrêt de la broche)

II.4.5.2. Vitesse

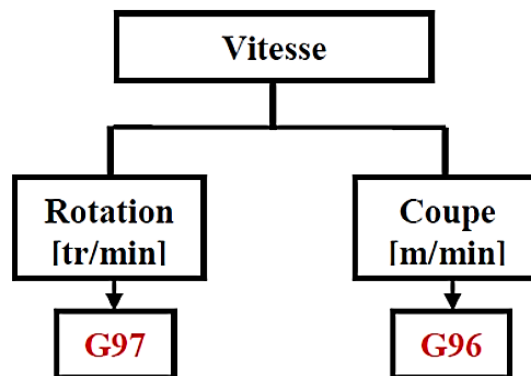


Figure II.13. Vitesse de la broche.

Syntaxe : **N.. G97 S... [M03/M04]**

N.. G96 [X.] S... [M03/M04]

G97 : Vitesse de broche exprimée en tr/mn.

G96 : Vitesse de coupe constante exprimée en m/min.

S : Argument obligatoire lié à la fonction et définissant la vitesse programmée.

Propriétés :

La fonction G97 est une fonction modale initialisée à la mise sous tension. La fonction G96 est une fonction modale.

Révocation :

La fonction G97 est révoquée par G96 S.. .

La fonction G96 est révoquée par G97 S.. .

Exemple :

N100 G97 S900 M40 M04 (Rotation de broche à 900 tr/mn).

N110 ... X50 Z70 (Positionnement du nez de l'outil sur diamètre 50).

N120 G96 S200 (Initialisation de la V.C.C sur X=50).

Remarque :

Concernant l'usinage en tournage avec vitesse de coupe constante et pendant une opération de dressage, le diamètre tend vers zéro. Alors il faut penser à limiter la vitesse de rotation maximale avec la fonction G92

Syntaxe :

N... G92 S...;

Propriétés :

La fonction G92 est modale.

Révocation :

La limite de la vitesse est annulée par :

- La fonction d'annulation G92 S0.
- La fonction G92 S... affectée d'une vitesse limite différente.
- La fonction de fin de programme M02.
- Une remise à l'état initial (RAZ). [7]

II.4.6. Mode d'avance

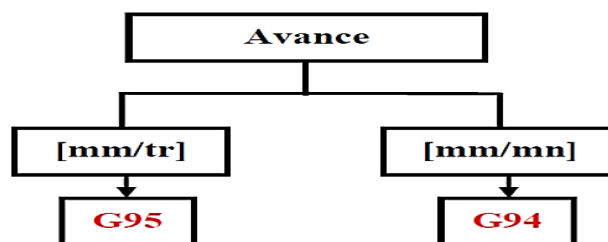


Figure II.14. Mode d'avance.

Syntaxe : N.. G95F.. G01/G02/G03 X..Z..

N.. G94F.. G01/G02/G03 X..Y../X..Z..

Propriétés :

La fonction G94 est une fonction modale initialisée à la mise sous tension. La fonction G95 est une fonction modale.

Révocation :

Les fonctions G94 et G95 se révoquent mutuellement. [7]

Exemple 1 : (en fraisage).

N...

N50 G0 X..Y..

N60 G94 F200 (Vitesse d'avance en mm/min).

N70 G1 X..Y..

N..

Exemple 2 : (en tournage).

N..

N50 G0 X..Z..

N60 G95 F0.3 (vitesse d'avance en mm/tr)

N70 G 1 X..Z..

N..

II.4.7. Programmation des outils

II.4.7.1. Appel d'outil et des correcteurs d'outil en tournage [FANUC]

Syntaxe : N.. T0106

T: La fonction « T » appelle l'outil.

01: Appel ou chargement de l'outil n°1.

06: Appel ou chargement du correcteur n°6. [6]

II.4.7.2. Appel d'outil et des correcteurs d'outil en fraisage [SINUMERIK-SIEMENS 840D]

Syntaxe : **N... T=" nom de l'outil" N5 M6 D...**

Propriétés :

Dans le cas de magasins à chaîne, à plateau tournant et à râtelier, le changement d'outil s'effectue normalement en deux opérations :

- L'instruction T recherche l'outil dans le magasin.
- Puis l'instruction M charge l'outil dans la broche.

Dans le cas des tourelles révolvers installées sur les tours, le changement d'outil (autrement dit la recherche et la mise en place de l'outil) est uniquement exécuté par l'instruction T. Le changement d'outil doit être accompagné des opérations suivantes :

- Activation des valeurs de correction d'outil enregistrées sous un numéro D
- Programmation du plan de travail correspondant. Ceci est nécessaire pour que la correction de longueur d'outil soit affectée au bon axe. [6]

Exemple :

N4T=" FRAISE_2T_Diam20" D1..... « Recherche de l'outil "FRAISE_2T_Diam20" dans le magasin et activation de correcteur D1.

N5 M6..... Montage de l'outil dans la broche.

N6 S1200 M3 F350

II.4.8. Correcteur d'outils G41-G42

- G41 : correction du rayon à gauche du profil à usiner.
- G42 : correction du rayon à droite du profil à usiner.
- G40 : annulation du correcteur de rayon

Syntaxe : **N.. [D..] [G0/G1/G2/G3] G41/G42 X..Y../X..Z..**

N.. [G0/G1] G40 X.. Y../X..Z..

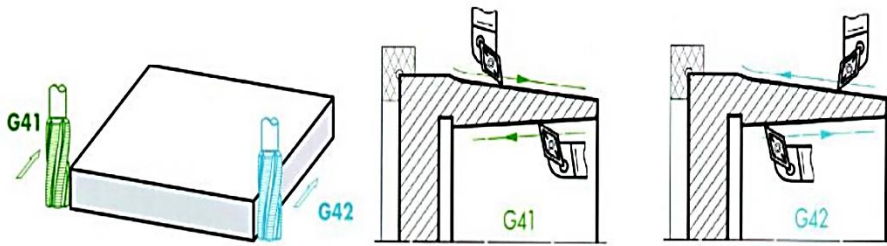


Figure II.15. Correcteur d'outils G41-G42.

Propriétés :

Les fonctions G41 et G42 sont modales.

La fonction G40 est initialisée à la mise sous tension.

Révocation :

Les fonctions G41 et G42 se révoquent mutuellement.

La fonction G40 révoque les fonctions G41 et G42. [5]

Exemple de programmation avec correction de trajectoire de rayon d'outil.

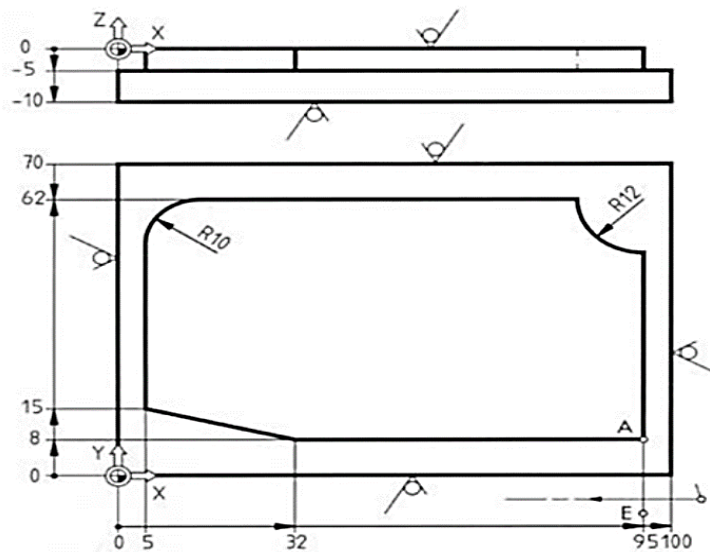


Figure II.16. Exemple de programmation avec correction de trajectoire de rayon d'outil.

Données :

- L'outil « Fraise 2 T_20 » Fraise à deux tailles de diamètre 20 mm.
- Vitesse de coupe : $V_c = 30$ m/min.
- Avance : $F = 80$ mm/min.
- A début de contour. et E Fin de contour.

Tableau II.4. feuille de programme.

N1 G17 G40G71 G90 G94	En-tête de programme.
N2 G54	Décalage d'origine.
N3 WORKSPIECE (,»BOX", 112, 0,-10, - 100, 0, 0, 100, 70)	Définition de la pièce brute.
N4 T="Fraise 2T_20"	Sélection de l'outil "Fraise 2T_20" du magasin.
N5 M6	Montage de l'outil dans la broche.
N6 S478 M3F80	Conditions de coupe, vitesse de broche et avance
N7 G0 X112 Y-2	Approche au point d'insertion cycle d'usinage
N8 G41 Z-5	CRF à gauche du profil et prise de la profondeur de la passe.
N9 G1 X95Y8 M8	Approche vers début de contour (A) et l'activation arrosage.
N10 X32	Usinage du contour.
N11 X5 Y15	
N12 Y52	
N13 G2 X15Y62 I10 J0	
N14 G1 X83	
N15 G3 X95Y50 I12 J0	
N16 G1 Y-12	
N17 G40 G0 Z100 M9	Dégagement en Z et désactivation de l'arrosage
N18 X150 Y150	Dégagement en X et en Y.
N19 M30	Fin de programme.

II.4.9. Condition d'enchaînement des trajectoires

L'arrêt précis est utilisé pour l'exécution d'angles saillants ou la finition d'angles rentrants. Le critère d'arrêt précis détermine la précision d'accostage du coin et l'instant où a lieu le changement de bloc. [6]

En tournage

G09 : Arrêt précis en fin de bloc avant enchaînement sur le bloc suivant.

G09 permet de résoudre l'erreur de poursuite E_p . [6]

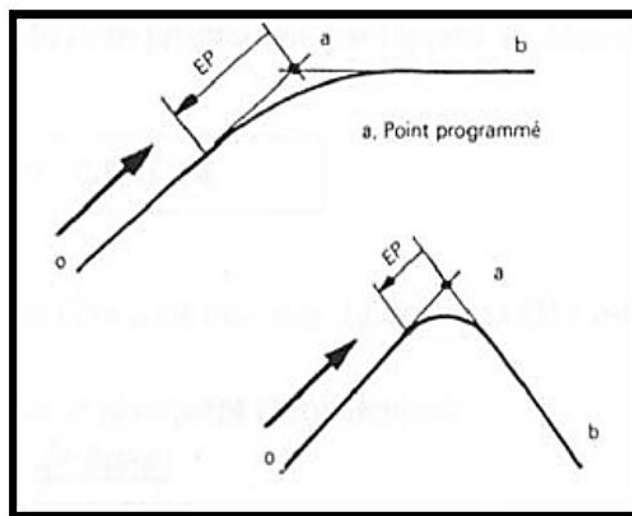


Figure II.17. Condition d'enchaînement des trajectoires en tournage.

En fraisage

G60 : Instruction d'activation de l'arrêt précis à effet modal.

G09 : Instruction d'activation de l'arrêt précis à effet non modal.

G601 : Instruction d'activation du critère d'arrêt précis "Arrêt précis fin".

G602 : Instruction d'activation du critère d'arrêt précis "Arrêt précis grossier".

G603 : Instruction d'activation du critère d'arrêt précis "Fin de l'interpolation". [6]

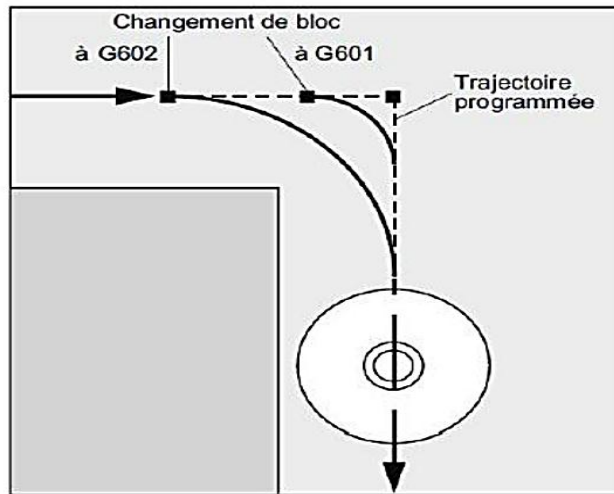


Figure II.18. Condition d'enchaînement des trajectoires en Fraisage.

Syntaxe : N.. G09 [G00/G1/G2/G3] X.. Z.. [F..]

Propriétés :

La fonction G09 est une fonction non modale, elle est révoquée en fin de bloc. G60 le génère dans le bloc courant et dans tous les blocs suivants :

Avec les instructions de contournage G64 ou G641 - G645, G60 est désactivé. [6]

Exemple :

N5 G602 ; Critère ‘ Arrêt précis grossier ‘ activé

N10 G0 G60 Z... ; Arrêt précis modal actif

N20 X... Z... ; G60 continue d’agir

N50 G1 G601 ; Critère ‘ arrêt précis fin ‘ activé

N80 G64 Z... ; Commutation sur contournage

N100 G0 G9 ; Arrêt précis s’applique uniquement dans ce bloc

N110 ... ; Réactivation du contournage

II.5. Qualité et rôle du programmeur humain dans les systèmes à commande numérique

Dans tous les systèmes à commande numérique, c’est l’ensemble homme-machine qui doit être considéré.

Si l'opérateur qui utilise la machine a un rôle plus réduit que dans la conduite manuelle d'une machine, son rôle étant alors essentiel de la surveillance. Par contre, toute la responsabilité du travail incombe à un nouveau venu qui intervient au stade de la préparation, il s'agit du programmeur chargé de préparer les instructions qui permettent à la machine d'effectuer dans les meilleures conditions le travail demandé

. Les qualités exigées du programmeur sont nombreuses :

- Excellente connaissance de la machine-outil.
- Entraînement très poussée collecte de toutes les informations détenues par les déférentes responsables des processeurs etc....

En contrepartie, le résultat final ne dépend plus de l'adresse de l'opérateur surveillent le fonctionnement de la machine ainsi qu'il a été vu plus haut les temps improductifs sont nettement réduits. Enfin, les rebuts de fabrications se trouveront supprimer.

III.1. INTRODUCTION

Après avoir décrit la programmation manuelle de code-g dans le chapitre précédent, nous allons dans ce chapitre investir la partie électronique responsable de la commande du système. Nous avons commencé par une description théorique des différents composants utilisés pour la réalisation de cette commande pour présenter ensuite la programmation sur Arduino.

III.2. Présentation de la carte Arduino Uno

La carte Arduino est une carte électronique basée autour d'un microcontrôleur et de composants minimum pour réaliser des fonctions plus ou moins évoluées à bas coût. Elle possède une interface USB pour la programmer.

C'est une plateforme open-source qui est basée sur une simple carte à microcontrôleur (de la famille AVR), et un logiciel, véritable environnement de développement intégré, pour écrire, compiler et transférer le programme vers la carte à microcontrôleur.

Arduino est peut-être utilisé pour développer des applications matérielles industrielles légères ou des objets interactifs, et il peut recevoir en entrées une très grande variété de capteurs. Les projets Arduino peuvent être autonomes, ou communiquer avec des logiciels sur un ordinateur (Flash, Processing ou MaxMSP, Labview). Ces cartes électroniques peuvent être fabriquées manuellement ou bien être achetées pré-assemblées, le logiciel de développement open-source est téléchargé gratuitement. [5]

La programmation de la carte Arduino présente les principales fonctionnalités de l'interface de l'application Arduino. L'application Arduino permet de créer et d'éditer un programme (appelé sketch) qui sera compilé puis téléversé sur la carte Arduino. Ainsi, lorsque des changements sur le code sont apportés, ces changements ne seront effectifs qu'une fois le programme téléversé sur la carte.

III.2.1. Présentation et caractéristiques du module Arduino Uno

La carte Arduino Uno, est une carte à microcontrôleur basée sur un Atmega328p. elle dispose de :

- Broches numériques d'entrées/sorties,
- Entrées analogiques,

- Quatre UART (port série matériel),
- Quartz 16Mhz,
- Connexion USB,
- Connecteur d'alimentation jack,
- Connecteur ICSP,
- Bouton de réinitialisation.

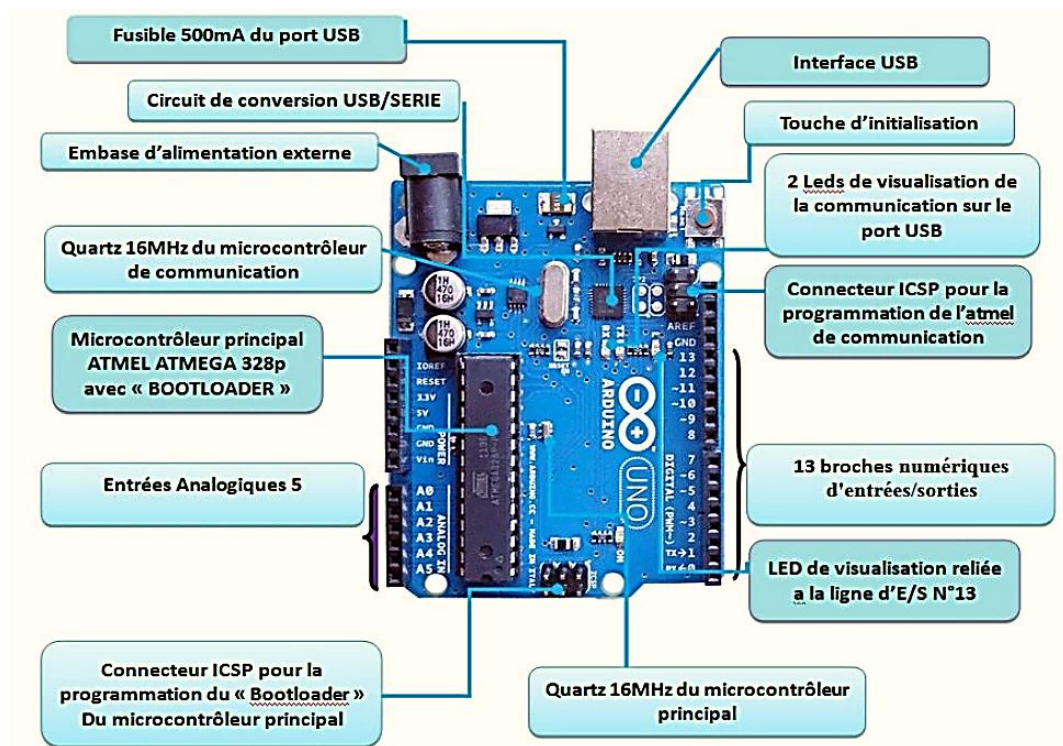


Figure III.1 : Module ARDUINO ATMEL UNO [7]

Elle contient tout ce qui est nécessaire pour le fonctionnement du microcontrôleur. Pour pouvoir l'utiliser et se lancer, il suffit simplement de la connecter à un ordinateur à l'aide d'un câble USB (ou de l'alimenter avec un adaptateur secteur ou une pile, mais ceci n'est pas indispensable, l'alimentation est fournie par le port USB).

La carte Arduino Uno dispose de toute une série de facilités pour communiquer avec un ordinateur, une autre carte Arduino, ou avec d'autres microcontrôleurs. Elle dispose de quatre UARTs (Universal Asynchronous Receiver Transmitter) ou (en français : émetteur-récepteur asynchrone universel) pour une communication en série de niveau TTL (5V) et qui est disponible sur les broches 0 (RX) et 1 (TX). Un circuit intégré Atmega16U2 assure la connexion entre cette communication série de l'un des ports série de l'Atmega Uno vers le port USB de l'ordinateur qui apparaît comme un port COM virtuel pour les logiciels de

l'ordinateur. Le code utilisé pour programmer l'Atmega8U2 utilise le driver standard USB COM, et aucun autre driver externe n'est nécessaire.[6]

III.2.2. Description matérielle

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

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

La composition des éléments internes d'un microcontrôleur :

- **RAM** : c'est la mémoire dite "vive", elle va contenir les variables de votre programme. Elle est dite "volatile" car elle s'efface si on coupe l'alimentation du microcontrôleur (comme sur un ordinateur).
- **EEPROM** : C'est le disque dur du microcontrôleur. Vous pourrez y enregistrer des infos qui ont besoin de survivre dans le temps, même si la carte doit être arrêtée. Cette mémoire ne s'efface pas lorsque l'on éteint le microcontrôleur ou lorsqu'on le reprogramme.
- **LES REGISTRES** : c'est un type de mémoire utilisé par le processeur. La mémoire cache : c'est une mémoire qui fait la liaison entre les registres et la RAM. Nous n'en parlerons également pas tout de suite.
- **LE PROCESSEUR** : C'est le composant principal du microcontrôleur. C'est lui qui va exécuter le programme que nous lui donnerons à traiter. On le nomme souvent le CPU.[9]

III.2.2.1. Alimentation

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

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

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

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

❖ **Les broches d'alimentation sont les suivantes :**

- **VIN.** La tension d'entrée positive lorsque la carte Arduino est utilisée avec une source de tension externe (à distinguer du 5V de la connexion USB ou autre source 5V régulée). Vous pouvez alimenter la carte à l'aide de cette broche, ou, si l'alimentation est fournie par le jack d'alimentation, accéder à la tension d'alimentation sur cette broche.
- **5V.** La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (pour info : les circuits électroniques numériques nécessitent une tension d'alimentation parfaitement stable dite "tension régulée" obtenue à l'aide d'un

composant appelé le régulateur et qui est intégré à la carte Arduino). Le 5V régulé fourni par cette broche peut donc provenir soit de la tension d'alimentation VIN via le régulateur de la carte, ou bien de la connexion USB (qui fournit du 5V régulé) ou de tout autre source d'alimentation régulée.

- **3.3V** : Une alimentation de 3.3V fournie par le circuit intégré FTDI (circuit intégré faisant l'adaptation du signal entre le port USB de votre ordinateur et le port série de l'ATmega) de la carte est disponible : ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5V. L'intensité maximale disponible sur cette broche est de 50mA
- GND Broche de masse (ou 0V).[10]

III.2.2.2. Connecteurs E/S

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

- Courant d'entrée : Environ 1 μ A
- Valeur résistance de Pull Up optionnelle : 20 à 50 K Ω
- Courant de sortie : 40 mA max, 200 mA total microcontrôleur
- Tension d'entrée : Ne pas dépasser Vcc.
- Caractéristique convertisseur AD : Par approximation successives, 10bits, temps de conversion 13 à 260 μ s.
- Référence de tension interne : 1.1 ou 2.56 V.

III.2.3. Description du logiciel

Le logiciel de programmation des modules Arduino est une application Java, libre et multiplateformes, serve d'éditeur de code et de compilateur, et qui peut transférer le firmware et le programme à travers la liaison série (RS232, Bluetooth ou USB selon le module). Il est également possible de se passer de l'interface Arduino, et de compiler les programmes en ligne de commande. Le langage de programmation utilisé est le C++, compilé avec avr-g++, et lié à la bibliothèque d'entrées/sorties. La mise en place de ce langage standard rend aisé le développement de programme sur les plates-formes Arduino, à toute personne qui maîtrise le C ou le C++.

Afin de pouvoir programmer la carte Arduino, il faudra le logiciel Arduino pour charger le code en question. Le logiciel Arduino a pour fonctions principales :

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

Le logiciel Arduino intègre également un TERMINAL SERIE (fenêtre séparée) qui permet d'afficher des messages textes reçus de la carte Arduino et d'envoyer des caractères vers la carte Arduino. Cette fonctionnalité permet une mise au point facilitée des programmes, permet d'afficher sur l'ordinateur l'état de variables, de résultats de calculs ou de conversions analogique-numérique. Un élément essentiel pour améliorer, tester et corriger les programmes. [5]

III.2.3.1. Logiciel d'édition et programmation Arduino

Il est totalement gratuit et proposé en libre téléchargement sur le site Internet d'Arduino. Téléchargez et installez la dernière version que vous souhaitez sur votre ordinateur (Windows, Linux ou Mac). Une fois décompressée comprend le logiciel, les drivers L'archive, exemples et bibliothèques. Après l'indispensable installation des pilotes, lancez le logiciel à partir du dossier. La fenêtre de l'application Arduino comporte les éléments suivants :



Figure III.2 : Interface du logiciel Arduino.

III.2.3.2. Structure d'un programme Arduino

Un programme Arduino doit impérativement être divisé en 2 parties : une **fonction setup** et une **fonction loop**. Ces deux fonctions sont de type **void**, c'est à dire qu'elles ne peuvent pas prendre de valeurs.

La fonction setup est la fonction qui se lance au début du programme. Elle permet d'initialiser les variables et de définir les broches de la carte Arduino qui seront utilisées.

La fonction loop se lance après la fonction setup et, comme son nom l'indique, fait une boucle jusqu'à ce que la carte soit débranchée.

Les constantes du programme : Pour notre programme, nous avons utilisé de nombreuses constantes globales, définies en tout début de programme.

III.3.1. La commande dir/step

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

- **Step** : qui est l'horloge pour la vitesse. A chaque front descendant le moteur fera un pas.
- **Dir** : (ClockWise ou Counter ClockWise) qui décidera du sens de rotation du moteur pour réaliser cette commande numérique on doit étudier deux circuits de commande.
- **le couple L297 et l298.**
- **Le circuit EASY DRIVER.**

III.3.1.1. Circuit L297

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

- Un translateur (séquenceur) qui génère les différentes séquences de commande.
- Un double hacheur PWM (Pulse With Modulation) qui régule le courant traversant les bobinages du moteur.

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

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

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

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

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

III.3.1.1.1. Le schéma théorique du montage

L297 est un circuit qui contient toute la circuiterie de commande et de contrôle de moteurs pas à pas unipolaire et bipolaire. Ce circuit est utilisé conjointement avec un driver double pont tel que le L298 (figure III.3). L'ensemble forme une interface idéale pour le contrôle d'un moteur pas à pas par Arduino.

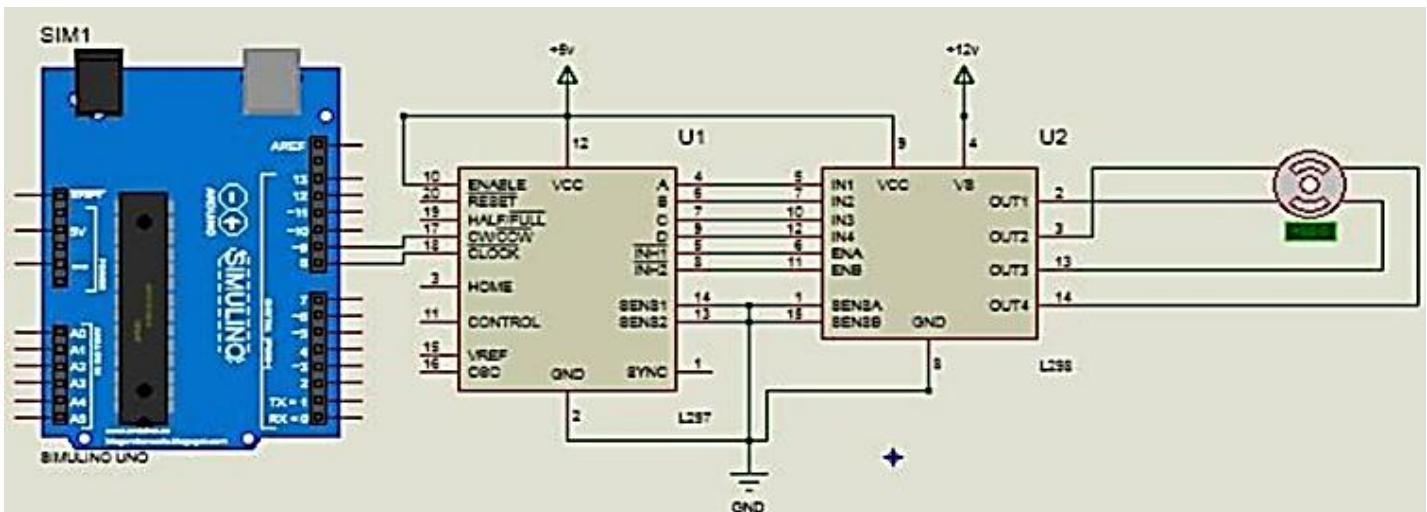


Figure III.3: commande moteur par le couple L297 et L298 Arduino.

III.3.1.2 Carte EasyDriver 4.4:

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

Elle est basée sur le circuit Allegro A3967 et permet de contrôler un moteur pas-à-pas (par ex : ITC-VNC-2) très facilement à partir d'un microcontrôleur (deux broches pour le mode, une broche pour la direction et une pour la vitesse). Le module permet de fonctionner en mode pas complet, 1/2 pas, 1/4 de pas ou 1/8 de pas (mode par défaut) [12]. Pour simplifier la connectique, elle peut être installée sur une carte mini Shield.

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

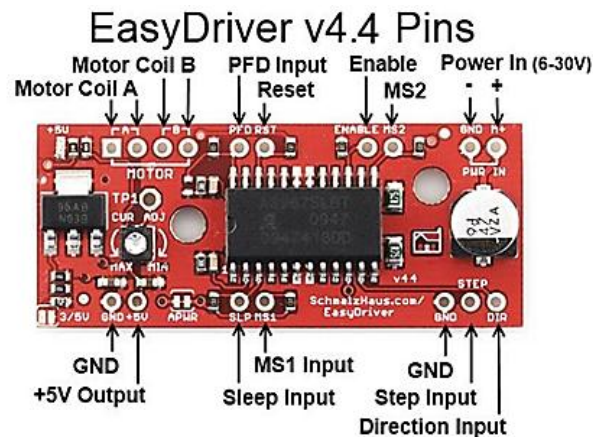


Figure III.4 : Circuit EasyDriver

III.3.1.2.1 Le schéma du montage Easy Driver

La commande avec ce circuit ne défère pas sur la commande avec le couple 1297-1298 qui contient une partie puissance et une partie qui obtient la séquence de pas. (Pas entier ou les autres modes à partir du mode choisi).

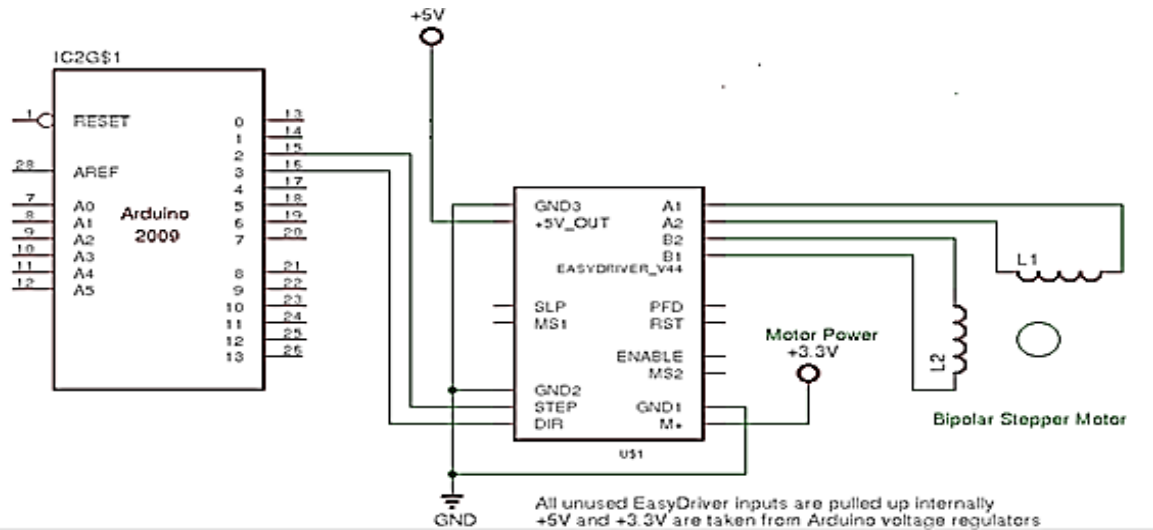


Figure III.5 : Commande moteur par Easy Driver.

III.3.1.3. Programmation de La commande dir/step

La commande à partir du couple L297-L298 ou Easy Driver est plus simple puisque on l'utilise que deux lignes de pilotage « Step et Dir ». Les déclarations des constantes pour les broches utilisées dans le programme :

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

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

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

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

```
Void loop() {
```

```
digitalWrite(dirpin, HIGH); //dirpin = high pour sans direct low pour sans inverse
for (i = 0; i<nbstep; i++) // boucle pour générer le signal rectangulaire
{
digitalWrite(stepin, LOW);
digitalWrite(stepin, HIGH); //transition état bas vers état haut (front montant) pour faire
marcher le moteur.
delayMicroseconds(500); }
} //temporisation pour ajuster la vitesse de rotation
```

III.4. Logiciel de pilotage de la machine CNC

La solution la plus fonctionnelle qu'on a trouvé est une évolution de Universal Gcode Sender (développé initialement par l'auteur de GRBL), qui est cours de développement par Winder [15].

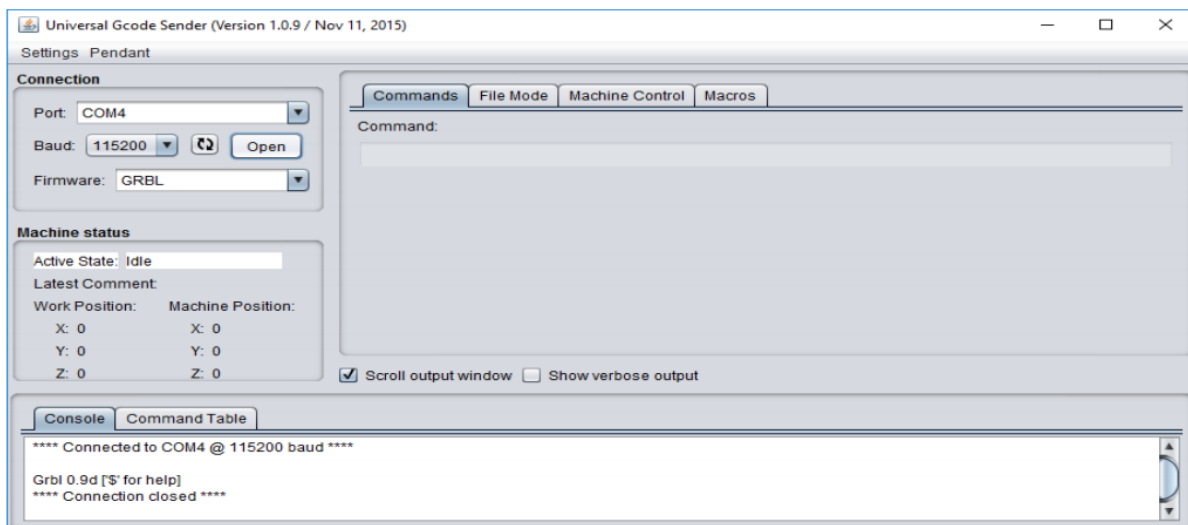


Figure III.6 : de Universal Gcode Sender.

III.4.1. Modes de commande

Il y'a deux modes utilisés dans ce programme, soit le mode manuel ou le mode automatique pour commander la machine CNC. Le programme fournit des boutons spéciaux dans l'interface de base qui contrôlent chaque moteur séparément, le fonctionnement de ces boutons après avoir choisi le mode manuel.

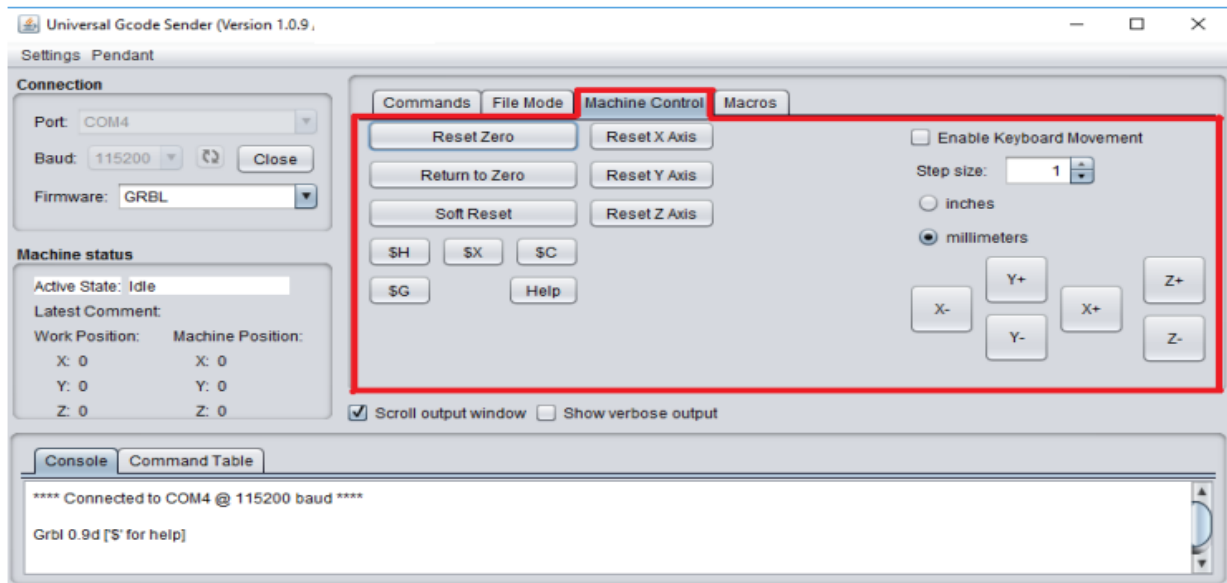


Figure III.7: l'interface de "g code sender" on mode manuel

Pour l'utilisation graphique : le mode automatique, Charge notre fichier G-code : dans l'onglet "file mode" et on fait "**browse**" puis on sélectionne le fichier G-code (.ngc) précédemment créé.

En faisant "visualize" nous pourrons vérifier notre tracé et le parcours de l'outil : En rouge les zone de travail (Z<0); en bleu les déplacements (Z>0).

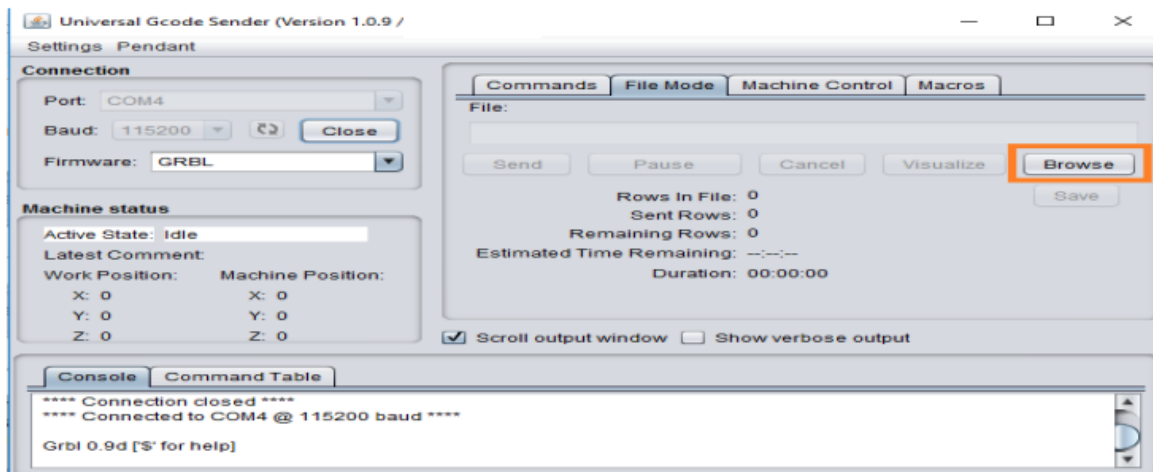


Figure III.8: l'interface de "Gcode sender" choisir le fichier Gcode.

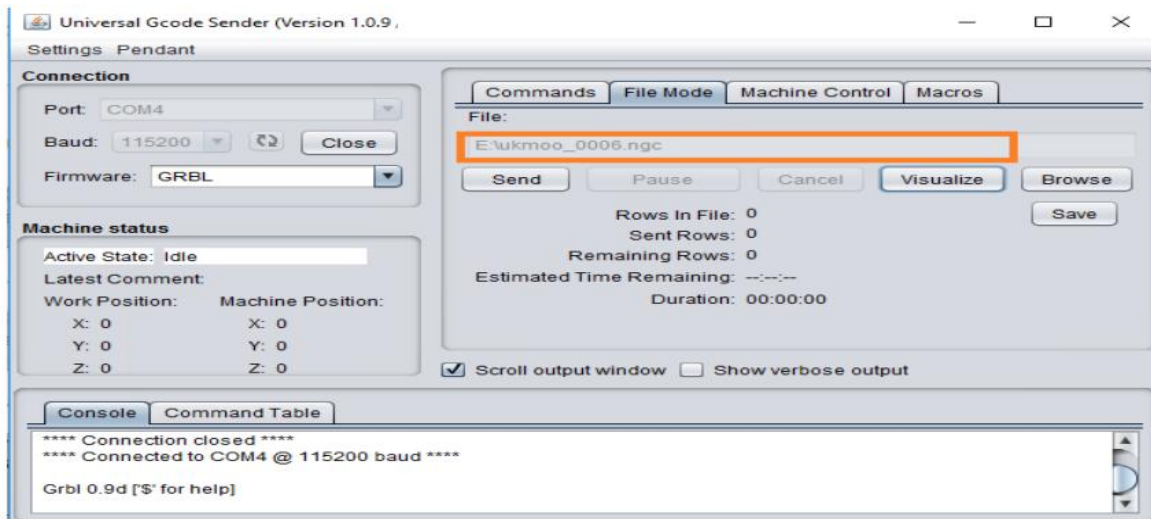


Figure III.9: l'interface de "g code sender" on mode automatique.

Pour faire le zéro (le "home") : dans l'onglet "machine control" on déplace l'outil avec les boutons X, Y et Z. Nous pouvons modifier le pas de déplacement en modifiant le "step size". Une fois l'outil positionné là où on souhaite on établit le zéro (en contact avec notre support) on clique sur "reset zéro".

Pour lancer le travail : on retourne dans l'onglet file mode et on clique sur "Send".

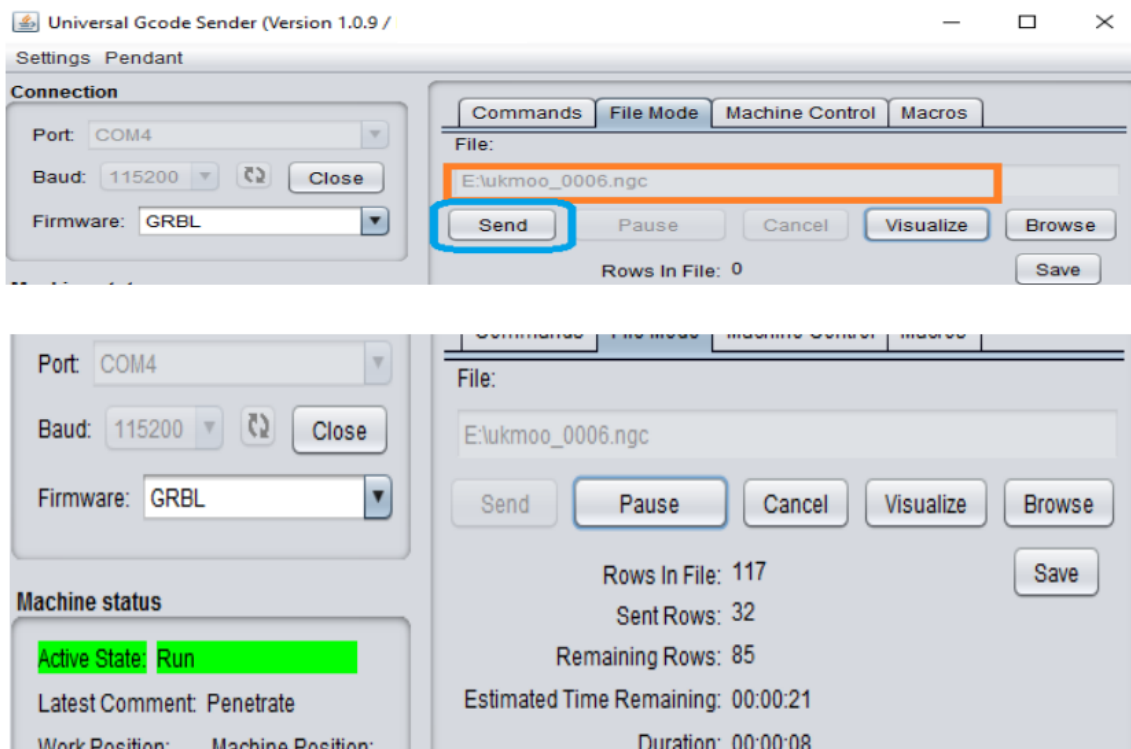


Figure III.10 : envoyé fichée G code

III.4.2. Mode Configuration

Autre que le mode manuel et le mode automatique, il y a le mode configuration. Dans ce mode, nous pouvons configurer les dimensions de la table. Afin de configurer les dimensions, l'opérateur est appelé à mettre la tête aux coins inverses de l'origine table après le lancement d'une prise d'origine. Cela est nécessaire pour mémoriser les distances maximales des axes X et Y. et la vitesse et nombre de pas par unité de tel moteur. Ainsi, ces paramètres sont enregistrés dans la mémoire EEPROM.

III.4.3. Mode manuel

En mode manuel, l'opérateur peut commander les moteurs directement par les boutons (X+, X-, Y+, Y-, Z+, Z-,) dans le programme de transfert G-code, si en cliquant sur ses boutons nous générons une instruction de code G qui envoyé à la carte de commande pour commander le moteur connecté avec ce bouton.

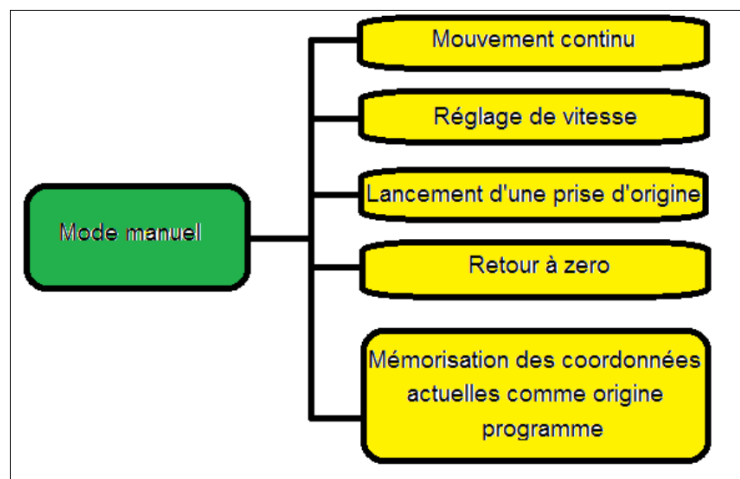


Figure III.11: mode manuel.

- **Mouvement continue** : lors de déplacement des moteurs, si l'on appuye sur un bouton de déplacement, la tête se déplace d'une manière continue avec une vitesse réglable.
- **Incrémentati n par pas** : pour un réglage fin, plusieurs distances de déplacement sont proposées. Dans ce mode, un seul appui g n re un d placement d'une seule unit  choisie par l'op rateur.
- **R glage de la vitesse** : la vitesse de la t te en mode manuel est r glable. Le choix de cette vitesse est limit  par la vitesse minimale et maximale indiqu es au manuel des moteurs (V_{min} , V_{max}).

- **Retour à zéro** : il s'agit de déplacer la tête à l'origine absolue. Ce qui diffère de la prise d'origine décrite précédemment c'est que la tête retourne à l'origine sans passer par les butées, donc il est basé sur l'estimation de la position calculé par l'Arduino.
- **Retour à l'origine programme** : l'origine programme est par défaut l'origine absolue de la table, en fait, c'est l'origine du texte ou image. Cette origine peut être changée par le code en G en mode automatique ou même manuellement. Une fois changé, l'Arduino le mémorise dans son EEPROM.
- **Mémoriser l'origine programme** : si l'opérateur veut changer cette origine, il doit déplacer la tête manuellement à l'aide du déplacement continu et l'incrémentation par pas, puis il exécute la mémorisation.

III.4.4. Mode automatique

En mode automatique, la carte Arduino Uno reçoit les instructions en G-code et les exécute l'une après l'autre. Il y a plusieurs étapes pour achever cette tâche.

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

L'Arduino fait l'enregistrement dans la mémoire RAM, Puis fait la compilation pour savoir l'opcode (code de l'opération) et évidemment ses paramètres s'ils existent. Chaque opcode de code G a une fonction spécifique, s'il s'agit d'un paramétrage, l'instruction affecte une variable en EEPROM tels que l'origine programme, origine pièce, vitesse de déplacement...et s'il s'agit d'un déplacement, l'Arduino fait appel à la fonction qui contient l'interpolation de Bresenham [18] liée à ce déplacement (interpolation linéaire et interpolation circulaire). Enfin, l'Arduino envoie le mot « » pour indiquer au PC qu'il est prêt à recevoir l'instruction suivante. L'appui sur le bouton 'cancel' termine le dialogue.

III.5. Langage de programmation la machine CNC

Un langage de programmation est un langage permettant à un être humain d'écrire un ensemble d'instructions (code source) qui seront directement converties en langage machine grâce à un compilateur (c'est la compilation). L'exécution d'un programme Arduino s'effectue de manière séquentielle, c'est-à-dire que les instructions sont exécutées les unes à la suite des

autres. Voyons plus en détail la structure d'un programme écrit en Arduino. Le programme utilise l'équipe de Grbl est un programme de commande dans la carte Arduino.

III.5.1. Programme Grbl

Grbl est un micrologiciel libre développé sur Arduino pour contrôler des graveuses CNC (Computer Numerical Control), i.e. des fraiseuses munis d'une tête mobile contrôlée en X, Y et Z par un ordinateur.

Le Grbl interprète le G-code et déplace en conséquence un outil sur 3 axes (X, Y et Z). Il comprend de multiples optimisations sur l'usage et le déplacement des moteurs afin de gérer correctement les accélérations, les trajectoires...l'organigramme qui assure ce micrologiciel [16]

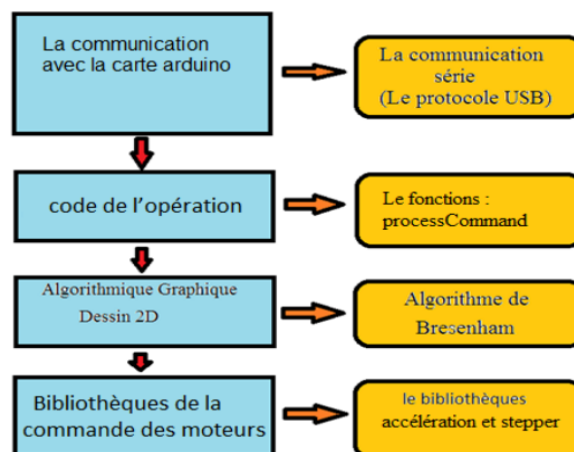


Figure III.12: schéma de GRBL code.

III.5.1.1. La communication avec la carte arduino

La communication série est indispensable pour dialoguer avec Arduino puisque c'est le mode de transmission utilisé pour communiquer avec la carte. Les protocoles de communication série le plus connus sont : Le protocole USB.

- **Serial.begin(speed)** (configuration de la vitesse de communication Série).
- **Serial.available()** (donne combien de caractères disponibles dans la zone tampon Série).
- **Serial.read()** (lit les données Série).

- **Serial.print(data)** (envoi des données Série).
- **Serial.println(data)** (envoi des données Série suivies de caractères spécifiques).

L'avantage de la communication série est qu'elle nécessite moins de lignes, donc moins de broches, donc moins de composants. Son coût est donc plus faible [17].

III.5.1.1. Bibliothèques de la commande des moteurs

Cette partie contient les fonctions qui transfèrent les calculs de l'interpolation vers un signal de commande pour le contrôle de la carte de commande. La fonction principale utilisée est la **fonction accélération** et la **fonction stepper**.

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

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

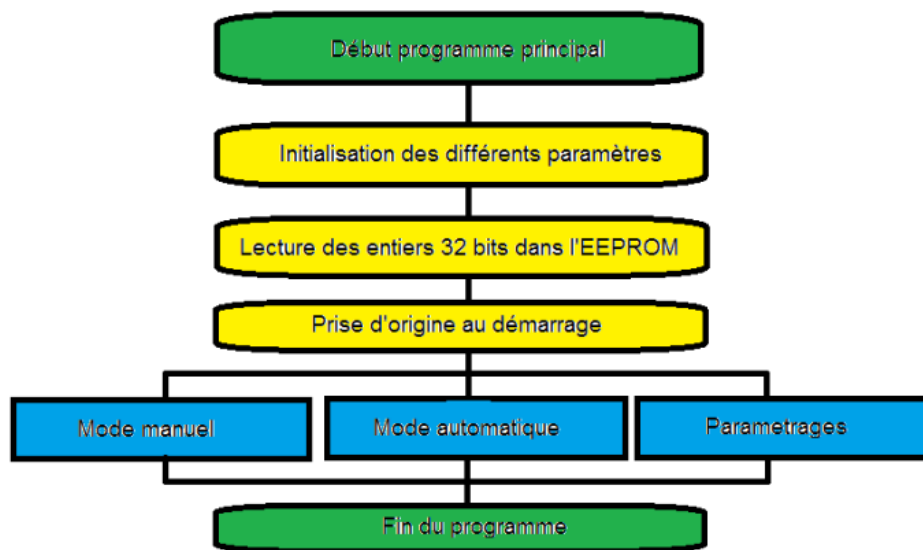


Figure III.13 : Organigramme de la commande de la machine CNC.

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

L'opérateur peut faire une prise d'origine sur un axe en mode manuel mais pas en mode auto. L'opérateur peut lancer l'exécution complète d'un programme de G-codes en mode auto mais pas en mode manuel. En mode manuel, chaque commande est entrée séparément. En termes humains une commande manuelle pourrait être active un moteur pas à pas.

Ces commandes sont normalement contrôlées en pressant un bouton de l'interface graphique avec la souris ou en maintenant appuyée une touche du clavier.

En mode auto, un bouton similaire ou l'appui d'une touche peuvent être utilisés pour charger ou lancer l'exécution complète d'un programme de G-codes stocké dans un fichier.

Introduction

L'introduction d'une CN dans une entreprise ne peut en aucun cas être considérée comme un simple remplacement de la machine conventionnelle. Par contre son adaptation doit être bâtie sur une base solide qui se résume en la progression de la production. Donc la CN se justifie par son aspect économique et sa qualité productive qui conduisent l'entreprise à des résultats profitables avec des chances réelles d'évolution rapide.

IV.1. Caractéristiques techniques de sélection des MOCN

Avant de choisir une machine-outil, il faut consulter plusieurs constructeurs, où la plupart de ces derniers offrent un large choix de machine.

Donc pour faire un bon choix, il faut bien contrôler et comparer entre ces différents facteurs qui suivent : [10]

IV.1.1. Aspect de la machine

IV.1.1.1. Bâti et structure

Pour éviter les vibrations et obtenir une grande précision, il faut que la structure de la machine soit d'une grande rigidité.

Les éléments mécaniques doivent avoir :

- Des paliers sans jeux.
- Un faible coefficient de frottement.
- Une faible inertie.

IV.1.1.2. Glissières

les différents critères de choix sur les glissières qui existent A frottement avec coefficient de frottement moyen et précision moyen .hydraulique avec faible coefficient de frottement et grande précision ou avec roulement de frottement négligeable et grande précision.

D'après nos recherches, nous avons constaté que les glissières à roulement seront les plus convenables à utiliser.

IV.1.1.3. Les vis-mères

Les critères de choix sur les vis-mères

Vis-mère classique avec coefficient de frottement moyen et jeux de fonctionnement faible et grande fluxion. Crémaillère et pignon avec coefficient de frottement faible jeux moyen et fluxion moyenne. ou bien vis à billes avec frottement très faible jeux très faible

D'après nos recherches, nous avons constaté que les vis à billes seront les plus convenables à utiliser.

IV.1.4. Organes de transmission

Les machines à commande numérique possèdent des vis avec noix pourvues de billes précontraintes donc exemptes de jeu, ce qui permet le travail en avalant qui soulage l'effort de coupe.

IV.1.5. Les moteurs

Les moteurs peuvent être choisis selon la gamme de vitesse, leurs précisions ou bien leurs moments transmis.

Les moteurs à courant-continu avec précision moyen vitesses élevée Moment transmis faible. Moteurs pas à pas avec grande précision et vitesses moyen

IV.2.1. Conception de la machine

IV.2.2. Moteur pas à pas

Les moteurs pas-à-pas diffèrent, par leur mode de commande, des moteurs classiques. Les moteurs pas à pas permettent de transformer un signal électrique numérique en un mouvement angulaire. Chaque impulsion envoyée par le système de commande au module de puissance se traduit par la rotation d'un pas du moteur. La résolution angulaire d'un moteur pas à pas va de 4 à 400 pas.

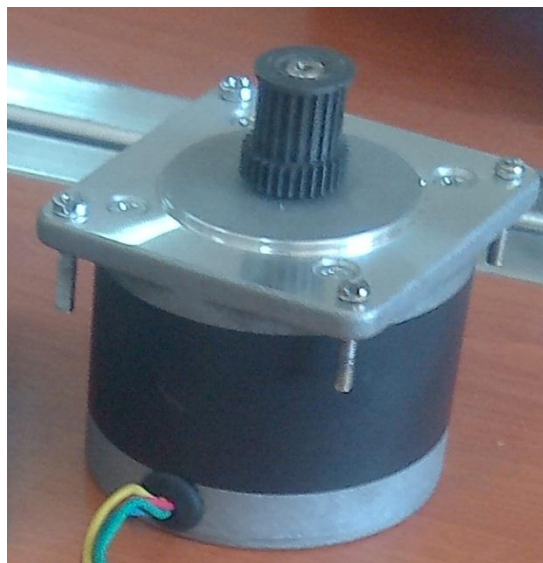


Figure IV.1. Moteur pas à pas

IV.2.3. Palier

Les paliers sont des organes utilisés en construction mécanique pour supporter et guider, en rotation, des arbres de transmission.

Suivant l'usage désiré, ces paliers peuvent être :

- Lisses où les arbres qui reposent sur des coussinets sont soumis au frottement de glissement entre les surfaces en contact.
- À roulement où le contact s'effectue par l'intermédiaire de billes ou de rouleaux contenus dans des cages. On a là un phénomène de résistance au roulement (parfois appelé improprement « frottement de roulement ») qui permet une plus grande charge sur les paliers et une plus grande vitesse de rotation.



Figure IV.2. Palier.

IV.2.4. Accouplement

En mécanique, un accouplement ou joint de transmission est un dispositif de liaison entre deux arbres en rotation, permettant la transmission du couple.

Il permet éventuellement un certain désalignement (accouplement élastique, joint de cardan...), autorise des décalages angulaires (accouplement hydraulique...) et peut être temporairement inactif (embrayage, crabotage).

Un accouplement mécanique peut être un accouplement élastique, un accouplement à membranes, un accouplement à denture métallique ou encore un accouplement à soufflet.



Figure IV.3. Accouplement.

IV.3. La réalisation de la mocn

IV.3.1. BATI

Le bâti de la machine réaliser avec des tubes souder avec MAG



Figure- IV.4 bâti de la machine

IV.3.2. AXE X

Pour la réalisation d'axe x nous utilisons une tôle de 5mm d'épaisseur pour assurer la rigidité



Figure- IV.5 axe X

IV.3.3. AXE Z

L'axe z avoir le petit cours par rapport les autres axes. Réaliser avec une tôle de 3.5mm et deux rails parallèles. La longueur de course et de 160mm



Figure- IV.6 axe Z

IV.3.4. AXE Y

Axe y réaliser avec deux rails parallèles. Le mouvement s'effectuée par 8 roulements et la distance de course est de 650mm



Figure- IV.7 axe Y

IV.3.5. BRIDE

Nous utilisons une tôle de 3.5mm avec des trous pour fixer la pièce dans tout l'espace de travail de la mocn



Figure- IV.8 brides de fixation

IV.3.6. SUPPORTS DES MOTEURS

Les supports des moteurs réalisés par une tôle de 0.8mm



Figure- IV.9 supports des moteurs

IV.4. ASSEMBLAGE DE MACHINE

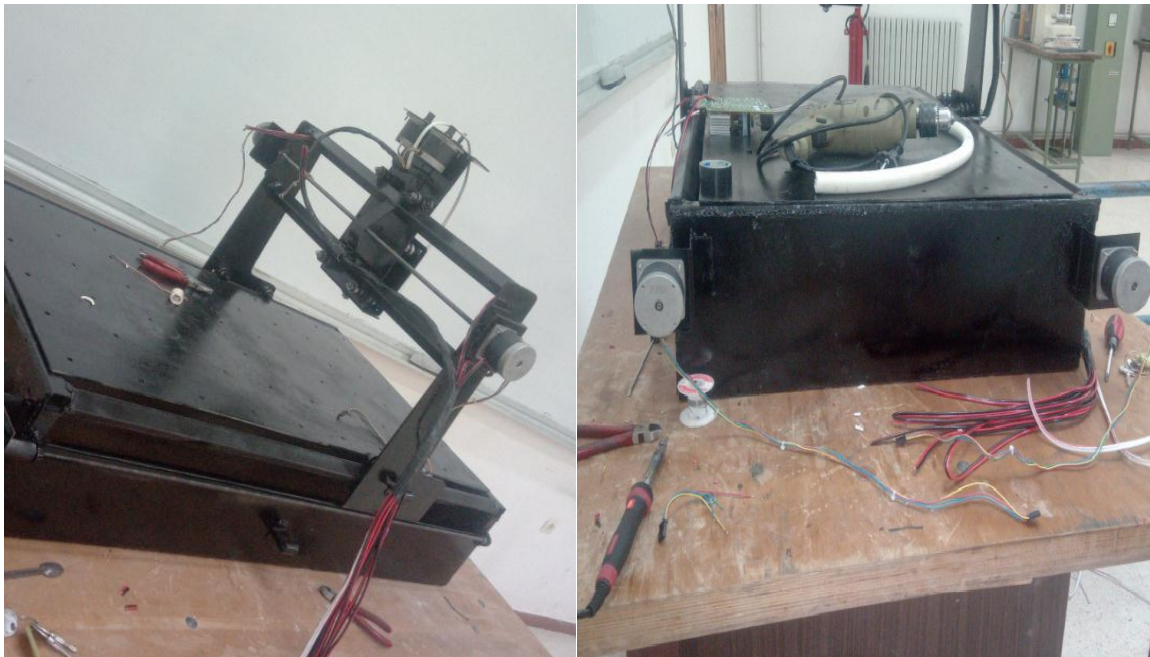


Figure IV.4.10 assemblage de machine

IV.4. La réalisation de partie électronique

IV.4. 1 L'installation de la carte d'alimentation

Une carte d'alimentation 24V et intensité plus de 15A pour les driver



Figure IV.4.11 la carte d'alimentation

pour amplifier la tension de signal de la carte d'interface pour ouvrir ou bien fermer la circuit électrique qui commander de la broche



Figure IV.4.12 relais avec transistor

IV.4.3 L'installation des drivers

- Alimenter chaque driver avec tentions de 24V avec la carte d'alimentation
- les sortie step et dire de carte d'interface
- les sortie de 3.3V et 5V de la carte d'interface
- les quatre sortie de le moteur pas-a-pas

IV.5. Suggestions pour améliorer la machine

La machine a besoin de nombreuses améliorations que nous n'avons pas pu apporter en raison du manque de temps et d'argent. Nous les mentionnons :

L'utilisation d'une structure en fonte pour absorber les vibrations

L'utilisation de freins pour éviter le problème de collision

L'utilisation d'un détecteur de force pour détecter les collisions et autres problèmes

Le changement des glissières avec un autre type plus précis

Le changement des moteurs avec un autre type plus puissant.....

Coût des pièces mécaniques utilisées

6 moteurs pas à pas	2000 00 DA *6	rolay module 500 00 DA*1
18 m de tige en acier	800 00 DA *3	3 vis mère 200 00 DA*3
40 roulement	150 00 DA*40	les autres accessoires 3500 00 DA
Carte arduino	4500 00 DA*1	
Child pour arduino	1500 00 DA*1	
6 Driver a4910	1200 00 DA*6	
Broche	9000 00 DA*1	
Carte alimentation	2000 00 DA*1	

Conclusion générale

Lors de cette réalisation de la MOCN, j'ai essayé de donner une configuration qui pourra être adaptable à plusieurs domaines d'utilisation, tels que l'usinage, le sculptage, le découpage...etc.

Puisque le stage pratique a été fait au niveau du hall technologique de la faculté de technologie précisément dans le laboratoire de CFAO, tout prêt de la machine à CN 3 axes Ainsi que de l'imprimante 3D, ce stage était très bénéfique vu la cohérence entre le présent travail et ces machines, ce qui me permet d'inspirer des bonnes idées vis-à-vis la conception et la réalisation de la cinématique de la machine.

Le but de ce travail est de donner un premier modèle référence, ce qui me permet dans le futur, de bien déterminer les améliorations nécessaires à la lumière des manques et inconvénients qu'ont été commis pendant l'achèvement de ce travail, en ce qui concerne le choix des moteurs, des glissières, le matériau du bâti de la machine qui devrait être amortisseur des vibrations, le système de bridage, les chemins de câbles, attachement outil, système de sécurité, pupitre... etc.

Ce travail pourra être bénéfique aussi pour les gens qui veulent investir dans ce volet et créer ces propres entreprises de services en but de contribuer au développement du pays.

L'élaboration de ce travail dans le cadre du projet de fin d'étude, m'a permis d'approfondir mes connaissances théoriques acquises durant notre formation et d'acquérir une bonne expérience au niveau de la réalisation pratique. Ce projet m'a donné une meilleure idée sur la complémentarité entre le volet théorique et le volet pratique.

Bibliographies

- [1] A. Fouillé Dunod, la commande des machines-outils, Histoire du Fer Bibliothèque de Travail n° 918
- [2] Gilles Patri, « Aérospatiale : une nouvelle ère industrielle. Des outils classiques à la commande numérique », dans Revue aérospatiale, N° hors-série 20 ans d'Aérospatiale, janvier 1990
- [3] <https://bu.univ-ouargla.dz/master/pdf/Bentaleb-gasmi.pdf?idmemoire=55>
- [4] Gilles prud'homme, « commande numérique des machines-outils » Technique de l'ingénieur, génie mécanique, usinage, B7130, D. Duret, « simulation de gamme d'usinage », revue de l'ingénieur et du Technicien de l'enseignement technique, n°229,1981, pp. 34-37,1995.
- [5] Claude Barlier, Lionel Girardin, "Mémotech productique : matériaux et usinage" Ed. Casteilla, pp 406, 1992.
- [6] Jean-Pierre Urso "Mémotech : commande numérique" Educationalivre, pp 334, 1999.
- [7] René Magnin, Jean-Pierre Urso "Mémotech commande numérique : programmation" Ed. Casteilla, pp 220, 1991.
- [8] Andre Chevalier, Jacques Bohan, "Guide du technicien en productique" Hachette, pp 272, 1988.
- [9] Andre Chevalier, Jacques Bohan, "Guide du technicien en productique : pour maitriser la production industrielle" Hachette technique, pp 288, 1998.
- [10] Iddir Smail, Slimani samir "Essais et Méthodologie en vue d'une exploitation rationnelle des MOCN" pp163, 1991, [11] <http://www.larousse.fr/archives/grande-encyclopedie/page/11526>
- [12] https://fr.wikipedia.org/wiki/%C3%89lectro-%C3%A9rosion#cite_ref-1
- [13] <http://www.metabricoleur.com/t2002-electro-erodeuse-de-conception-amateur>
- [14] A. Cheikh, N. Cheikh, cours de CFAO1, Université de Tlemcen, Algérie, 2011
- [15] D. Gelin, M. Vincent « éléments des fabrications », Edition marketing, paris, Mars 1995
- [16] <http://www.usimm.ca/quelles-sont-les-differences-entre-lusinage-3-axes-et-5-axes/>
- [17] <http://akou91.over-blog.com/article-construction-d-une-table-de-fraisage-a-commande-numerique-diy-cnc-router-98803054.html>
- [18] <http://sam.electroastro.pagesperso-orange.fr/dossiers/pasapas/moteurpas2.htm>

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

[20]<http://ressources.univ-lemans.fr/AccesLibre/UM/Pedago/physique/02/electri/pasapas.html>

[21] Pierre Mayé : «Moteurs électriques pour la robotique» ; 2^{ém} édition, octobre 2013

ملخص

تتطلب صناعة القطع الميكانيكية البسيطة تحكم جيد في الآلات الميكانيكية لكن مع التقدم التكنولوجي وزيادة المتطلبات الصناعية خاصة القطع الأكثر تعقيدا وجب اللجوء الى آلات التحكم الرقمي المخصصة لذلك من خلال عملنا هذا قمنا بإعطاء عموميات حول الات التحكم الرقمي والبرمجة المخصصة لها وقمنا أيضا بصناعة نموذج لالة تحكم رقمي انجزناها على مستوى مخابر الجامعة وتطرقنا الى المكونات الالكترونية والواجهة البرمجية للتحكم بهذه الالة

كلمات مفتاحية

CFAO ،CAO ،DAO ،MOCN

Résumé

Le secteur de la fabrication mécanique nécessite un bon contrôle mécanique, mais avec les progrès technologiques et les exigences industrielles croissantes, en particulier les pièces les plus complexes, Il est nécessaire de contrôler les technologies des machines à commande numérique.

Au cours de ce travail, nous avons généralisé les dispositifs de commande numériques et la programmation nécessaire.

Nous avons également fabriqué un modèle de modèle de contrôle numérique que nous avons réalisé au niveau du laboratoire universitaire.

Nous avons abordé les composants électroniques et l'interface logicielle pour contrôler cette machine.

Mots-clés

CFAO ،CAO ،DAO ،MOCN

Abstract

The mechanical manufacturing sector requires good mechanical control, but with technological advances and increasing industrial requirements, especially the most complex parts, it is necessary to control the numerical control technology,

In the course of this work, we have generalized the numerical control devices and the programming necessary,

We also made a model of mocn that we made at the university laboratory level.

We approached the electronic components and the software interface to control this machine,

Keywords

CFAO ،CAO ،DAO ،MOCN