

N° Ordre...../Faculté/UMBB/2016

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE M'HAMED BOUGARA BOUMERDES



Faculté des Hydrocarbures et de la Chimie

**Mémoire de fin d'études
En vue de l'obtention du diplôme**

MASTER 2

Présenté par

KARA ALI REDHA

Spécialité : Automatisation des procédés industriels

Option : Commande automatique

THEME

Etude, Conception et Réalisation d'une Imprimante 3D à Dépôt de la Matière

Devant les membres de jury :

Dr :Président

Dr :Membre du jury

Dr :Membre du jury

Dr : FEKHAR.H Promoteur

Mr : ANCA.F CO-Promoteur

Année universitaire : 2016/2017

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE M'HAMED BOUGARA BOUMERDES



Faculté des Hydrocarbures et de la Chimie

**Mémoire de fin d'études
En vue de l'obtention du diplôme**

MASTER 2

Présenté par

KARA ALI REDHA

**Spécialité : Automatisation des procédés industriels
Option : Commande automatique**

THEME

Etude, Conception et Réalisation d'une Imprimante 3D à dépôt de la matière

Avis favorable de l'encadreur :

Dr.FEKHAR.H

Avis favorable de président du jury :

.....

Remerciements

Pour commencer, nous tenons à exprimer notre profonde gratitude à notre seigneur dieu le tout puissant, l'éternel lui seul, qui m'a guidé le long de ce voyage dans le jardin de savoir.

Je tiens à remercier mon encadreur Mr H.FEKHAR pour son accompagnement, ses encouragements et sa confiance en moi tout au long de ce travail.

Je remercier également l'ensemble des enseignants du département automatisé et électrification des procédés industriels, qui ont assuré ma formation, notamment Mr HAMADACHE, Mr KHALASSI, Mr HABBI et Mr CHAIB pour leurs orientations, conseils et encouragements.

Je remercie spécifiquement et profondément Mr A.FAROUK, qui n'a jamais hésité à m'offrir son aide et à qui je serai toujours reconnaissant, grâce à lui, ce projet a pu voir le jour.

Je remercie la société SARL ALGEK, notamment Mr G.BILAL qui a accepté de m'accueillir au sein de l'entreprise.

Mes remerciements les plus sincères aux fondateurs, membres et adhérents de club scientifique AEPI New Vision dans lequel j'ai passé une expérience formidable avec des gens passionnés et motivés. Merci pour la vision, le savoir et les valeurs que vous m'avez donné et que nous avons partagé, je suis fier de faire partie de cette grande famille que nous devons tous la garder.

Enfin, je remercier tous ceux qui m'ont aidé de près ou de loin et qui ont contribué à la réussite de ce projet, parmi lesquels je citerai Mr A.AISSAOUI et Mr M.EMBAREK.

DEDICACE

J'ai le grand honneur de dédier ce modeste travail

A ceux qui ont fait de moi un homme

A ceux qui comptent le plus pour moi

*« Ma MERE » et « Mon PERE » qui m'ont aidé, encouragé et
soutenu durant toutes mes études.*

A toute ma famille

A tous mes amis et mes proches

A tous ceux qui m'ont aidé de loin ou de près durant les moments difficiles

INTRODUCTION GENERALE.....	1
CHAPITRE 1 : Techniques d'impression et architecture de l'imprimante 3D	
INTRODUCTION.....	3
I.1. Description des techniques d'impression 3D.....	3
I.1.1. Impression Par Dépôt De Matière (FDM).....	4
I.1.2. La Solidification Par Lumière.....	5
I.1.3. Le procédé Polyjet.....	6
I.1.4. Le Frittage Laser.....	7
I.1.5. L'Agglomération De Poudre Par Collage (La 3DP).....	8
I.2. Architecture de l'imprimante 3D.....	9
I.2.1. Le Frame.....	9
I.2.1.1. Axe X.....	10
I.2.1.2. Axe Y.....	11
I.2.1.3. Axe Z.....	12
I.2.1.4. Distance de translation en fonction de nombre de pas de moteur.....	13
I.2.2. Extrudeuse.....	14
I.2.3. Les moteurs pas à pas.....	16
I.2.4. Les capteurs.....	18
I.2.4.1. Capteur de fin de course.....	18
I.2.4.2. Capteur de température (Thermistance).....	19
I.2.4.2.1. Présentation de la Thermistance.....	19
I.2.4.2.2. Modélisation de la thermistance.....	20
I.3. Mode Opérateur.....	21
CONCLUSION.....	22

CHAPITRE II : Description du Hardware

II.1. Description de la carte électronique de commande réalisée.....	23
II.1.1. Le Microcontrôleur.....	26
II.1.1.1. Choix du microcontrôleur.....	26
II.1.1.2. Présentation de la carte de développement Arduino Mega.....	27
II.1.1.3. Présentation du microcontrôleur Atmega2560.....	28
II.1.1.4. Architecture interne de l'Atmega2560.....	29
II.1.1.5. Fonction des différents pins du microcontrôleur Atmega2560.....	29
II.1.2. Le circuit de puissance A4988.....	33
II.1.2.1. Caractéristiques et détails techniques de circuit A4988.....	33
II.1.2.2. Brochage et description des pins de driver A4988.....	34
II.1.2.3. Ajustement de Driver A4988.....	36
II.1.3. Afficheur LCD.....	38
II.1.3.1. Présentation de l'afficheur.....	38
II.1.3.2. Brochage et description des pins.....	39
II.1.4. Le Transistor Mosfet.....	41
II.2. Techniques de communication et d'interfaçage.....	42
II.2.1. Le protocole de communication UART.....	42
II.2.1.1. Présentation de Protocole UART.....	42
II.2.1.2. Configuration d'UART sur le Microcontrôleur Atmega2560.....	43
II.2.2. Le protocole de communication SPI.....	45
II.2.2.1. Présentation de Protocole SPI.....	45
II.2.2.2. Configuration de SPI sur le microcontrôleur Atmega2560.....	46
CONCLUSION.....	46

CHAPITRE III : Description du Software

INTRODUCTION	47
III.1. Présentation du Software Utilisé	47
III.2. L'environnement de développement Arduino IDE	47
III.2.1. Structure d'un projet Arduino.....	48
III.2.2. Interface du logiciel	49
III.2.3. Chargement du programme dans le microcontrôleur Atmega2560	50
III.3. Visual Studio C#	50
III.3.1. Présentation de l'environnement de développement Visual Studio	50
III.3.2. Présentation de langage C#	51
III.3.3. Commencer avec le Visual Studio C#	51
III.4. ISIS Proteus	54
III.4.1. Présentation de logiciel ISIS Proteus	54
III.4.2. Commencer avec ISIS Proteuse	54
III.5. Logiciel EAGLE	56
III.5.1. Présentation du logiciel EAGLE	56
III.5.2. Commencer avec EAGLE	56
III.5.3. Résultats du travail sur EAGLE	57
III.6. SolidWroks	59
III.6.1. Présentation du SolidWorks	59
III.6.2. Commencer avec le SolidWorks	59
III.7. Logiciel du tranchage	61
III.7.1. Présentation de G-Code	61
III.7.2. Génération de G-Code	61

III.7.3. Présentation de Slic3r	62
III.7.4. Commencer avec Slic3r	63
CONCLUSION	64
CHAPITRE IV : Elaboration des Algorithmes et résultats expérimentaux	
INTRODUCTION	65
IV.1. Interface de commande Homme-Machine	65
IV.1.1. Présentation de l'interface Homme-Machine	65
IV.1.2. Structure de l'algorithme d'interface de commande Homme-Machine	67
IV.2. Le Firmware	68
IV.2.1. Implémentation de la boucle de régulation PID de température	68
IV.2.2. Implémentation de la boucle de commande principale	69
IV.3. Testes et résultats	70
CONCLUSION	73
CONCLUSION GENERALE ET PERSPECTIVES	74

LISTE DES FIGURES

Figure I.1 : Processus d'impression par dépôt de la matière FDM.....	4
Figure I.2 : Processus d'impression par la Stéréolithographie.....	5
Figure I.3 : Processus d'impression Polyjet.....	6
Figure I.4 : Processus d'impression par frittage laser.....	7
Figure I.5 : Processus d'impression 3DP.....	8
Figure I.6 : Photo réelle de châssis en bois et de la plateforme d'impression.....	9
Figure I.7 : Photo réelle de roulement utilisé.....	10
Figure I.8 : Photo réelle de l'axe X assemblé.....	10
Figure I.9 : Photo réelle de l'axe Y assemblé.....	11
Figure I.10 : Photo réelle de l'axe Z assemblé.....	12
Figure I.11 : Les pièces de fixation de l'axe Z.....	13
Figure I.12 : Le pas (pitch) de tige filetée.....	13
Figure I.13 : Schéma de différentes parties de l'extrudeuse.....	14
Figure I.14 : Filament plastique (ABS).....	15
Figure I.15 : Photo réelle de l'extrudeuse.....	16
Figure I.16 : Dimensions de moteur Nema 17.....	17
Figure I.17 : Photo réelle de Moteur pas à pas –Nema 17.....	17
Figure I.18 : Capteur fin de Course mécanique.....	18
Figure I.19 : Capteur fin de Course optique CNY70.....	18
Figure I.20 : Schéma de montage de la thermistance.....	19
Figure II.1 : Schéma fonctionnel de la carte de commande.....	23
Figure II.2 : Schéma de brochage de la carte électronique.....	24
Figure II.3 : Photo réelle de la carte électronique avec désignation de ces différents éléments	25

Figure II.4 : Photo réelle de la carte de développement ArduinoMega.....	27
Figure II.5 : Architecture interne de l'Atmega2560.....	29
Figure II.6 : Photo réelle de l'atmega2560.....	29
Figure II.7 : Schéma diagramme de driver A4988.....	34
Figure II.8 : Schéma de brochage de driver A4988.....	35
Figure II.9 : La structure interne simplifié d'un LCD 2*16.....	39
Figure II.10 : Photo réelle de l'afficheur LCD.....	40
Figure II.11 : Schéma de brochage simplifié de l'afficheur LCD.....	40
Figure II.12 : Photo réelle de transistor STP55NF06.....	41
Figure II.13 : Format de tram de données.....	43
Figure II.14 : Photo réelle d'un convertisseur USB to UART.....	43
Figure II.15 : Schéma de brochage Master-Slave et Master-Multiple Slave.....	45
Figure III.1 : Structure minimale d'un code C sur Arduino IDE.....	48
Figure III.2 : Exemple de programme sur Arduino IDE.....	48
Figure III.3 : Interface du logiciel Arduino.....	49
Figure III.4 : Utilité des boutons d'interface Arduino IDE.....	49
Figure III.5 : La fenêtre principale de Visual Studio.....	52
Figure III.6 : Fenêtre de choix de langage et de type de projet.....	52
Figure III.7 : Fenêtre d'espace de travail.....	53
Figure III.8 : Interface principale de Logiciel ISIS Proteus.....	55
Figure III.9 : Schéma de montage sur ISIS Proteus.....	55
Figure III.10 : Interface principale du logiciel EAGLE.....	56
Figure III.11 : Interface de conception de Schéma de montage.....	56
Figure III.12 : Bibliothèque de composants sur EAGLE.....	57

Figure III.13 : Fenêtre de conception de circuit imprimé.	57
Figure III.14 : Le design de circuit imprimé sur le logiciel EAGLE.	58
Figure III.15 : Fenêtre principale de SolidWorks.....	59
Figure III.16 : Espace de travail sur SolidWorks.....	60
Figure III.17 : Assemblage de l'imprimante 3D sur SolidWorks.....	60
Figure III.18 : Interface graphique principale de Slic3r.....	63
Figure IV.1 : Interface de commande Homme-Machine.....	65
Figure IV.2 : Organigramme global simplifiée de l'interface de commande Homme-Machine	67
Figure IV.3 : Organigramme simplifié de la boucle de régulation PID de température.....	68
Figure IV.4 : Organigramme simplifié de la boucle de commande principale.....	69
Figure IV.5 : Teste de communication entre le PC et la carte électronique.....	70
Figure IV.6 : Teste de la boucle de régulation PID.....	70
Figure IV.7 : Photo réelle de l'imprimante 3D réalisée.....	71
Figure IV.7 : Teste final d'impression.....	72

LISTE DES TABLEAUX :

Tableau I.1 : Types et propriétés des filaments.....	15
Tableau II.1 : Les fonctions alternatives des pins de l'Atmega2560.....	32
Tableau II.2 : La configuration de mode de résolution.....	37
Tableau II.3 : Description des broches du LCD.....	39
Tableau II.4 : Caractéristiques de transistor STP55NF06.....	41
Tableau III.1 : Les commandes de G-Code et leur description.....	62

INTRODUCTION GENERALE

INTRODUCTION GENERALE :

Apparue il y a seulement quelques dizaines d'années, la commande numérique (CN) impose actuellement sa technologie dans le monde de l'usinage. Conçue pour piloter le fonctionnement d'une machine à partir des instructions d'un programme sans intervention directe de l'opérateur pendant son exécution, elle a, dans un premier temps, permis de franchir un pas important dans l'automatisation des machines-outils traditionnelles. Tours, fraiseuses, imprimantes 3D, perceuses et aléseuses et qui sont ainsi devenues capables d'assurer, en quantité comme en qualité, une production à peine imaginable quelques années auparavant.

Aujourd'hui, de plus en plus étroitement associée aux progrès de la microélectronique et de l'informatique, la CN voit ses performances et sa convivialité augmenter régulièrement tandis que, en revanche, son prix et son encombrement ne cessent de diminuer. Elle pénètre, de ce fait, dans les plus petites entreprises et devient accessible à tous les secteurs industriels faisant appel aux procédés de positionnement ou de suivi de trajectoire.

Les machines à commande numériques utilisent une technique de traitement de données composées de codes alphanumériques (G-code) représentant les instructions géométriques et technologiques nécessaires à la conduite d'une machine ou d'un procédé. C'est également une méthode d'automatisation des fonctions des machines ayant pour caractéristique principale une très grande facilité d'adaptation à des travaux différents. À ce titre, la CN constitue l'un des meilleurs exemples de pénétration du traitement de l'information dans les activités de production.

Après une première génération de CN à logique câblée, les machines à **commandes numériques par ordinateur (CNC)**, ou par ordinateur sont apparues. Ces dernières intègrent un ou plusieurs contrôleurs/ordinateurs spécifiques pour réaliser tout ou une partie des fonctions de commande.

Dans le cadre de notre projet, nous faisons l'étude, la conception et la réalisation d'une imprimante 3D. Ces dernières font parties de l'ensemble des machines à commande numériques à ordinateur qui se présentent comme des systèmes embarqués.

De plus, les imprimantes 3D font actuellement l'objet d'une révolution industrielle importante vue ses domaines et secteurs d'utilisation qui sont nombreux, ce qui nous a

conduit à entamer cette étude dans l'objectif est de créer et de développer un projet d'aspect purement industriel.

A travers ce projet nous avons divisé le travail sur quatre chapitres :

- Chapitre I : Techniques d'impression et Architecture de l'imprimante 3D ; où nous présentons les différentes techniques d'impression 3D industrielles existant ainsi que le choix et l'architecture de l'imprimante 3D réalisée.
- Chapitre II : Description du Hardware ; où nous présentons la carte de commande réalisée avec ses différents composants électroniques ainsi que, les techniques de communication et d'interfaçage utilisées en donnant toutes les détails nécessaires.
- Chapitre III : Description du Software ; où nous présentons les logiciels utilisés lors de développement de notre projet, ainsi, nous présentons quelques exemples simples afin de comprendre leur aspect pratique dans le cadre de notre projet.
- Chapitre VI : Elaboration des Algorithmes et résultats expérimentaux ; où nous présentons les différents organigrammes de commande élaborés afin de créer l'interface Homme-Machine sur PC et le firmware implémenté dans le mémoire flash du microcontrôleur ainsi que, les résultats des tests que nous avons effectué.

I

CHAPITRE

*Techniques d'impression et Architecture de
l'imprimante 3D*

INTRODUCTION :

L'impression 3D porte une révolution industrielle très importante vue ses domaines et secteurs d'utilisation qui sont nombreux : santé, industrie agroalimentaire, industrie automobile et Aérospatial, prototypage rapide,...

Les imprimantes 3D ne devenaient publiques et connus que ces derniers temps vue leur utilisation qui était limitée seulement sur l'échelle industriel et n'étaient pas à la portée de grand public, et ça malgré l'apparition de cette technique depuis le milieu des années 80 par « Charles Hull » qui a inventé la 1^{er} imprimante 3D qui fonctionne par la technique de la Stéréolithographie.

L'impression 3D n'est pas une technologie fonctionnant d'une seule et même manière. Il existe en effet une multitude de procédés permettant d'imprimer un objet en 3D. Cependant, même si les techniques d'impression sont différentes sur la forme, le principe reste toujours le même. Il consiste à superposer des couches de matières les unes sur les autres en remontant de bas en haut selon les coordonnées transmises par un fichier 3D jusqu'à la finition de l'objet 3D.

I.1. Description des techniques d'impression 3D : [1]

L'impression 3D fonctionne donc selon plusieurs procédés, qui diffèrent selon le type d'imprimante 3D utilisée. On peut classer ces procédés dans trois grands groupes :

- Le dépôt de matière.
- La solidification par la lumière.
- L'agglomération par collage.

Ces trois procédés fonctionnent selon le même principe de base, c'est à dire superposer des couches de matières selon les coordonnées d'un fichier 3D. La différence se situe sur la manière dont sont déposées et traitées ses couches, ainsi que le type de matériau utilisé. Pour la plupart des procédés employés l'utilisateur aura besoin :

- d'un ordinateur et d'une imprimante 3D.
- de consommable (filament, poudre...).
- d'un fichier 3D (au format STL, OBJ, G-CODE,...).
- d'un logiciel de tranchage pour trancher le fichier et transmettre les indications à l'imprimante.

I.1.1. Impression Par Dépôt De Matière (FDM) :

La majorité des imprimantes 3D personnelles fonctionnent selon ce principe. FDM est l'acronyme anglais de Fused Deposition Modeling qui signifie « modelage par dépôt de filament en fusion ». Ce procédé qui a été inventé en 1988 par la société Stratasys, est une marque déposée. On parle aussi de FFF (Fused Filament Fabrication) ou même de MPD (Molten Polymer Deposition) qui sont eux des termes libres de droits.

Cette technique consiste à déposer un filament de matière thermoplastique fondu à 200°C (en moyenne) et extrudé à travers une buse afin d'imprimer par coupe transversale un objet en 3D couche par couche. La tête d'impression se déplace selon les coordonnées X, Y et Z (longueur, largeur et hauteur) transmises par un fichier 3D correspondant au modèle 3D de l'objet à imprimer. Limitée pendant longtemps à des matériaux de type plastique tels que les classiques PLA et l'ABS, l'impression 3D voit arriver de nouveaux filaments composites à base de métal (cuivre, bronze...) et même de bois. Plus rarement certaines machines utilisent des cires ou des polycarbonates. Aujourd'hui l'industrie agroalimentaire et la médecine s'emparent peu à peu de cette technique pour imprimer des aliments et des cellules en adaptant la tête d'extrusion.

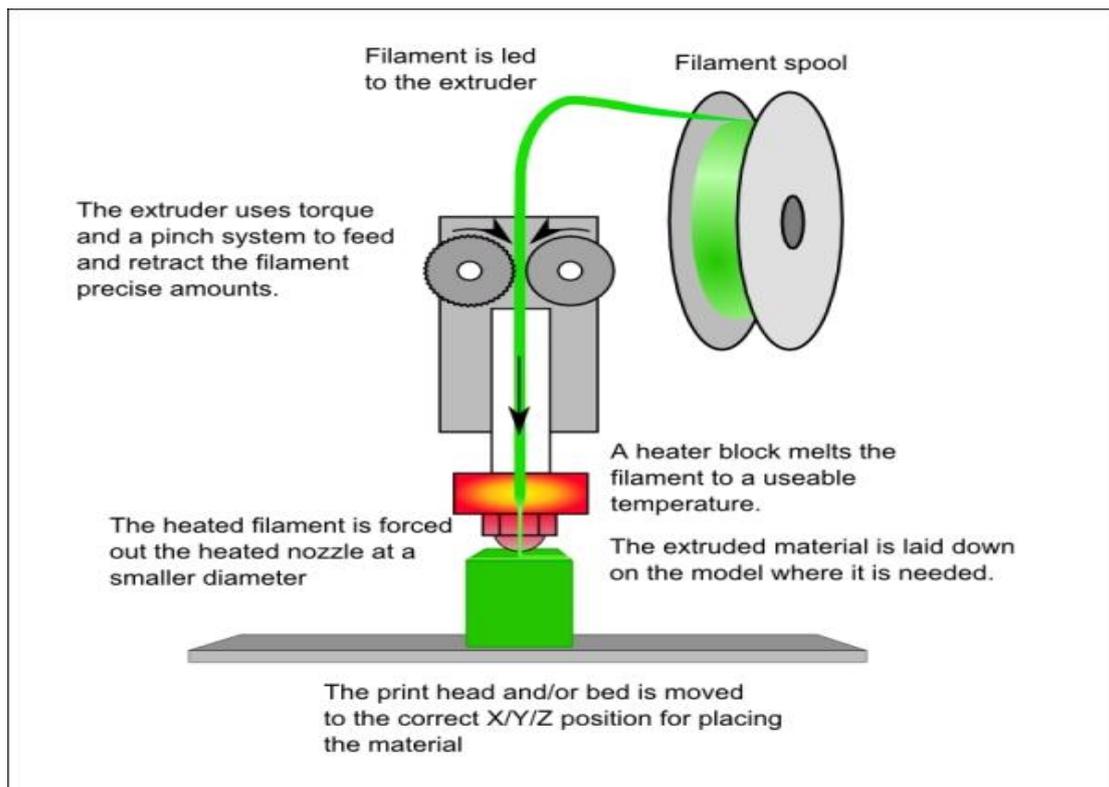


Figure I.1 : Processus d'impression par dépôt de la matière FDM.

I.1.2. La Solidification Par Lumière (LA STERÉOLITHOGRAPHIE OU SLA) :

La stéréolithographie est la première technique d'impression 3D à avoir été mise en évidence. La paternité de ce procédé est souvent attribuée à l'américain Charles Hull fondateur de 3D Systems. Appelée aussi SLA (Stéréolithographie Apparatus) cette technique consiste à solidifier un liquide photosensible par le biais d'un rayon laser ultraviolet. Les imprimantes fonctionnant par SLA ont quatre parties principales : un réservoir qui peut être rempli avec un liquide photopolymère, une plate-forme perforée qui est descendue dans le réservoir, un rayonnement ultraviolet (UV) et un ordinateur commandant la plate-forme et le laser.

Tout comme la FDM, l'imprimante va dans un premier temps analyser le fichier CAO, puis en fonction de la forme de l'objet le laser va commencer par toucher et durcir instantanément la première couche de l'objet à imprimer. Une fois la couche initiale de l'objet est durcie, la plate-forme s'abaissera et une nouvelle couche de surface de polymère liquide sera exposée. Le laser trace à nouveau une section transversale de l'objet qui colle instantanément à la couche déjà durcie. Ce processus se répète encore et encore jusqu'à ce que la totalité de l'objet ce soit formé et soit entièrement immergé dans le réservoir. La plateforme va ensuite se relever pour faire apparaître l'objet fini en trois dimensions. Après qu'il ait été rincé avec un solvant liquide pour le débarrasser de l'excès de résine, l'objet est cuit dans un four à ultraviolet pour durcir la matière plastique supplémentaire.

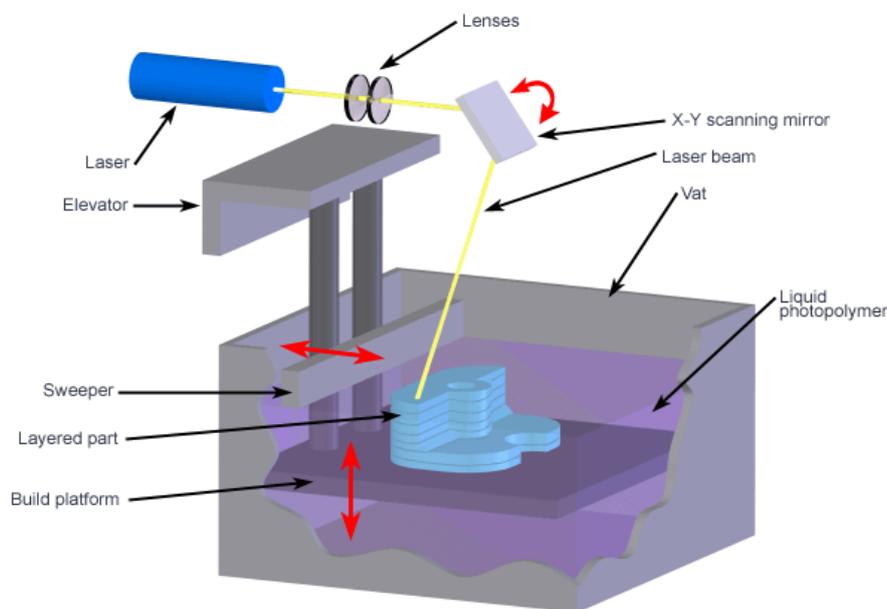


Figure I.2 : Processus d'impression par la Stéréolithographie.

I.1.3. Le procédé Polyjet :

Cette Technologie brevetée par la société « Objet Geometries Ltd », fonctionne aussi sur le principe de photo polymérisation. De la même manière, l'objet sera modélisé en 3D avec un logiciel spécialisé, puis, son fichier envoyé à l'imprimante. Les têtes d'impressions vont alors déposer en goutte à goutte de la matière photosensible sur un support de gel, selon les coordonnées transmises par le fichier. Une fois la matière déposée, celle-ci va être exposée à un rayon ultraviolet qui va alors la durcir instantanément. L'opération sera répétée jusqu'à obtention de l'objet final, il ne restera alors plus qu'à le nettoyer. Avec une précision de l'ordre de 0,005mm il est possible de réaliser des objets avec un haut niveau de détail.

À l'inverse de nombreuses autres techniques, le procédé Polyjet ne nécessite pas d'étapes de finition. La surface est bien lisse et la résolution parfaite. L'utilisateur peut retirer les matériaux de support à la main.

La technique d'impression 3D Polyjet offre un large choix de matières et de couleurs, elle présente aussi de nombreux avantages pour l'outillage et le prototypage rapide comme une vitesse et une précision optimales. Les pièces imprimées possèdent des détails incroyablement fins et des surfaces lisses. De nombreux domaines tels que la production d'objets commerciaux, l'architecture ou même la médecine utilisent cette technologie. Ainsi, vous pouvez :

- créer des prototypes lisses et détaillés qui ressemblent au produit final.
- produire en petit volume des outils de fabrication, des gabarits et des fixations.
- produire des objets possédant des formes complexes, des détails précis et des surfaces lisses.

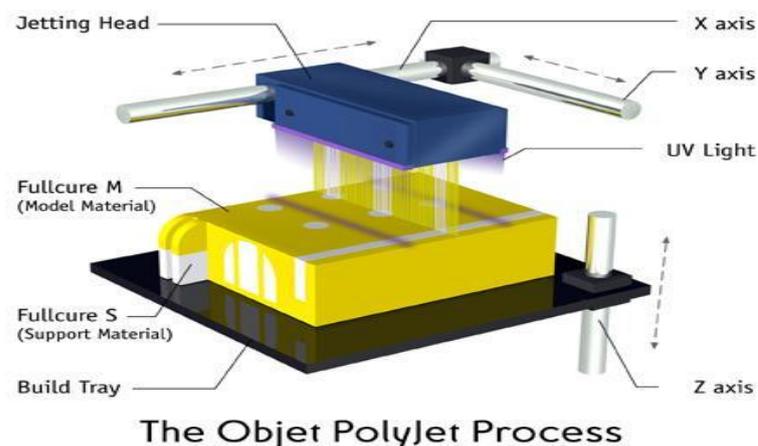


Figure I.3 : Processus d'impression Polyjet.

I.1.4. Le Frittage Laser :

Cette technique était créée par un étudiant américain dans une université du Texas en 1980, ensuite, elle a été développée plus tard (2003) par la société allemande EOS. Appelée aussi SLS (Selective Laser Sintering), il s'agit également d'un processus d'impression par laser. Cette fois un faisceau laser très puissant va fusionner une poudre (1mm d'épaisseur) à des points très précis définis par un fichier STL que communique votre ordinateur à votre imprimante. Les particules de poudre sous l'effet de la chaleur vont alors fondre et finir par se fusionner entre-elles. Une nouvelle couche de poudre fine est ensuite étalée et à nouveau durcie par le laser puis reliée à la première. Cette opération est répétée plusieurs fois jusqu'à ce que la pièce soit finie. L'objet nécessite une dernière étape pour la finition.

La poudre utilisée le plus souvent pour ce type d'impression est le polyamide de couleur blanche. Ce matériau est en fait un nylon, il va donner à l'objet une surface poreuse qui pourra être repeint. D'autres matières comme la poudre de verre, la céramique ou le plastique sont aussi utilisés. Souvent les fabricants utilisent un mélange de deux sortes de poudres pour obtenir des objets plus aboutis.

Le frittage laser permet de fabriquer des pièces avec un niveau de détail assez élevé (0.1mm) et à géométrie complexe. De plus la poudre restante qui n'aura pas été passée au laser pourra être réutilisée la fois suivante. Généralement les pièces obtenues avec ce processus demandent davantage de finitions (ponçage, peinture, vernis...).

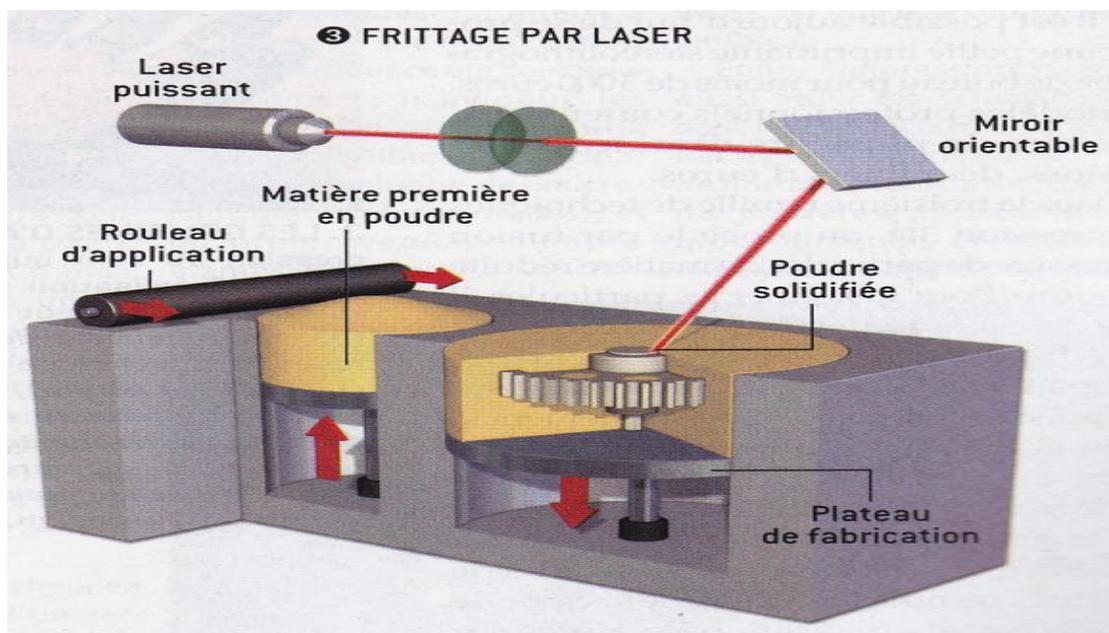


Figure I.4 : Processus d'impression par frittage laser.

I.1.5. L'Agglomération De Poudre Par Collage (La 3DP) :

Initialement développée en 1993 au Massachusetts à l'Institut of Technology (MIT) en 1993, la 3DP (Three-Dimensional Printing) constitue la base du processus d'impression 3D de Z Corporation.

Le procédé consiste sur l'étalement d'une fine couche de poudre de composite sur une plateforme. La tête d'impression va alors déposer sur celle-ci de fines gouttes de glue colorées qui seront combinées entre elles permettent d'obtenir un large panel de couleur. La plateforme s'abaissera au fur et à mesure que les couches de poudre sont collées jusqu'à l'obtention de l'objet final. Pour la finition il faut aspirer l'excédent de poudre, brosser et/ou poncer la pièce, puis la chauffer pour finaliser la solidification.

La 3DP a l'avantage d'être rapide et de proposer une large gamme de couleurs. Jusqu'à 6 fois moins chère qu'une imprimante SLA son prix est plus attractif malgré une précision et une qualité d'impression parfois inférieure. Parmi les inconvénients, les pièces sont plus fragiles et leur surface est plus rugueuse.

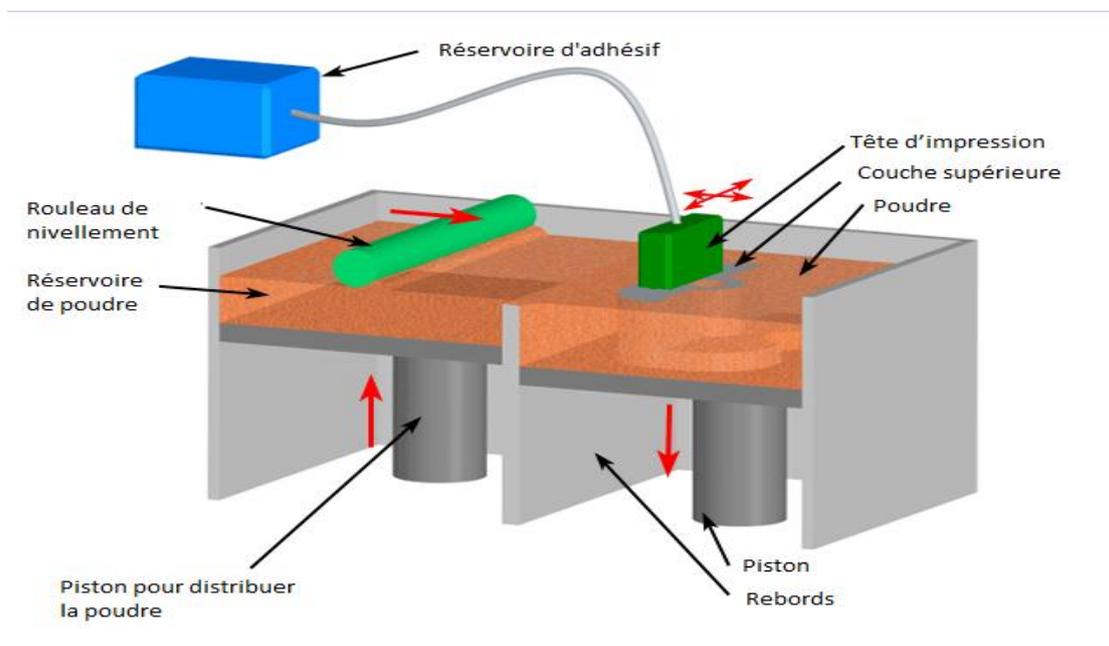


Figure I.5 : Processus d'impression 3DP.

I.2. Architecture de l'imprimante 3D :

Dans ce projet nous avons choisi de réaliser une imprimante 3D qui utilise la technique d'impression par dépôt de la matière FDM et cela revient à différents raisons qu'on les résume comme suit :

- La disponibilité de différentes pièces qui constituent l'ensemble de l'imprimante (moteurs pas à pas, extrudeuse, poulies et roulements,...).
- La simplicité de la commande de ces imprimantes qui ne nécessitent pas des techniques de commande avancées.
- Ce type d'imprimante est open source, il y'a toute une communauté de développeurs qui travaillent sur l'amélioration de ces imprimantes, ce qu'il signifie la disponibilité de documents nécessaires pour réaliser le projet ainsi, c'est une occasion pour nous pour contribuer dans cette communauté internationale.
- Ce type d'imprimantes est trop demandé par le grand public, ce qui représente un bon marché à envisager dans l'avenir.

Notre imprimante est constituée principalement d'un châssis en bois, des pièces de fixation en plastique, des capteurs et des actionneurs (capteurs fin de course, capteurs de température, moteurs pas à pas, extrudeuse), des roulements linéaires et des poulies, des courroies, tubes linéaires en inox, tiges filetées, ainsi une carte de commande à base de microcontrôleur.

I.2.1. Le Frame :

Le frame de l'imprimante est constitué principalement d'un châssis en bois sur lequel elles sont fixées les différentes pièces en plastique qui portent les trois axes de l'imprimante et d'une plateforme d'impression en Aluminium.

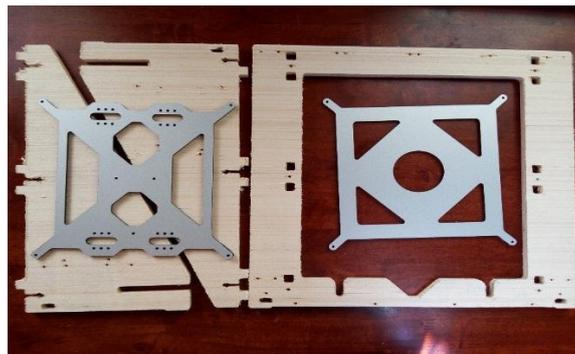


Figure I.6 : Photo réelle de châssis en bois et de la plateforme d'impression.

I.2.1.1. Axe X :

L'axe X permet le mouvement de la tête d'impression sur le plan horizontal d'impression. La transmission de mouvement sur cet axe est assurée par un moteur pas à pas bipolaire dont l'attraction est de type : Poulie-Courroie.

❖ Guidage de mouvement :

Afin de minimiser les frottements et assurer un mouvement libre de la tête d'impression sur l'axe X nous avons utilisé des roulements linéaires guidés sur deux tiges lisses. La figure I.7 représente une photo réelle de roulement utilisé.



Figure I.7 : Photo réelle de roulement utilisé.

❖ Assemblage de l'axe X :

La figure I.8 montre une photo réelle de L'axe X assemblé avec désignation de ses différents éléments.

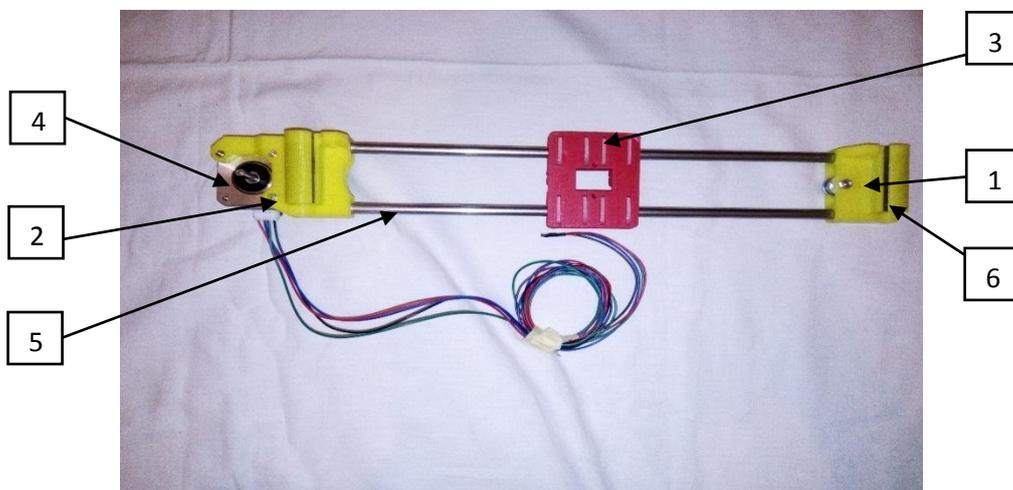


Figure I.8 : Photo réelle de l'axe X assemblé.

- (1) : X end idler (pièce de fixation).
- (2) : X end motor (pièce de fixation).
- (3) : X carriage (pièce de fixation).
- (4) : X axis stepper motor.
- (5) : tige lisse.
- (6) : roulement linéaire.

I.2.1.2. Axe Y :

L'axe Y permet le mouvement de la plateforme d'impression sur le plan latérale d'impression. La transmission de mouvement sur cet axe est assurée par un moteur pas à pas bipolaire dont l'attraction est de type : Poulie-Courroie de même façon que l'axe X.

❖ Guidage de mouvement :

De même façon que l'axe X, afin de minimiser les frottements et assurer un mouvement libre de la plateforme d'impression sur l'axe Y nous avons utilisé des roulements linéaires guidés sur deux tiges lisses.

❖ Assemblage de l'Axe Y :

La figure I.9 montre une photo réelle de L'axe Y assemblé avec désignation de ses différents éléments.

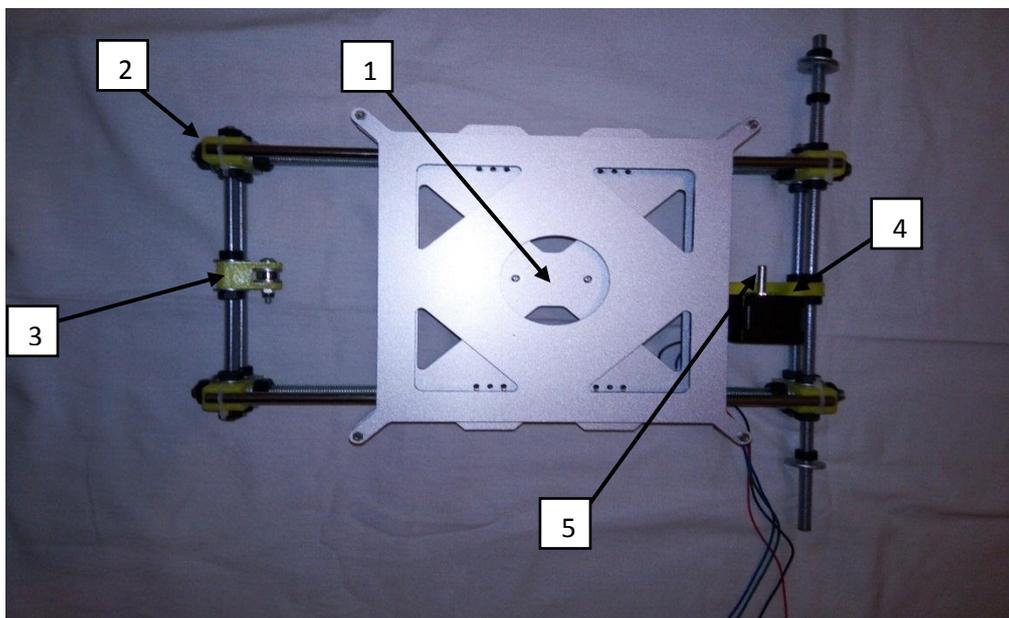


Figure I.9 : Photo réelle de l'axe Y assemblé.

- (1) : Y belt holder (pièce de fixation).
- (2) : Y corner (pièce de fixation).
- (3) : Y idler (pièce de fixation).
- (4) : Y motor bracket (pièce de fixation).
- (5) : Y axis stepper motor.

I.2.1.3. Axe Z :

L'axe Z permet le mouvement de la tête d'impression sur le plan vertical d'impression. La transmission de mouvement sur cet axe est assurée par deux moteurs pas à pas bipolaires dont l'attraction est de type : tige filetée-écrou.

❖ Guidage de mouvement :

De même façon que les deux axes X et Y, afin de minimiser les frottements nous avons utilisé des roulements linéaires guidés sur des tiges lisses. La seule différence c'est l'utilisation des tiges filetées à écrous pour la transformation du mouvement, et cela revient au poids appliqué sur cet axe.

❖ Assemblage de l'axe Z :

La figure I.10 montre une photo réelle de l'axe Z assemblé et la figure I.11 montre les différentes pièces de fixations de l'axe Z.



Figure I.10 : Photo réelle de l'axe Z assemblé.

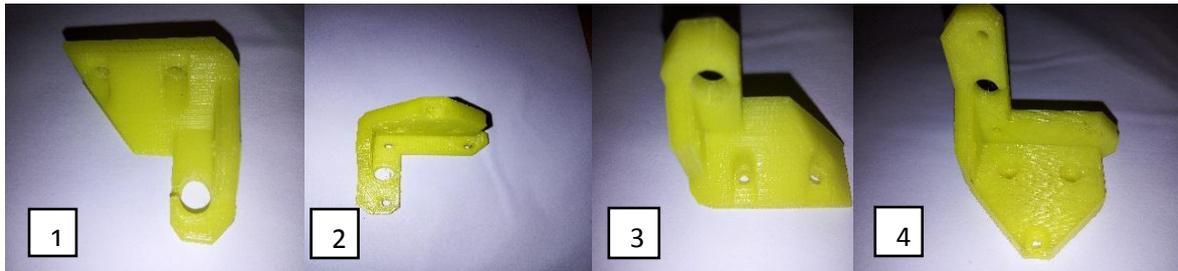


Figure I.11 : Les pièces de fixation de l'axe Z.

- (1) : Z axis top right.
- (2) : Z axis bottom left.
- (3) : Z axis top left.
- (4) : Z axis bottom right.

I.2.1.4. Distance de translation en fonction de nombre de pas de moteur :

Les moteurs pas à pas utilisés ont un degré de rotation égale à **1.8°/pas**, ce qui correspond à **200 pas/révolution** (rotation complète). On suppose que le diamètre de poulie utilisé soit égal à **Q**, donc pour 200 pas la distance de translation vaut $\pi * Q$. D'où on aura la relation suivante :

$$D = (N * \pi * Q) / 200.$$

Cette équation est valable pour les deux axes X et Y où nous avons utilisé des poulies et des courroies. Pour l'axe Z, la distance de translation est calculée de la façon suivante :

$$D = (N * \beta) / 200.$$

Avec :

- **N** : nombre de pas.
- **D** : distance de translation.
- **Q** : diamètre de poulie.
- **β** : le pas de tige filetée (pitch).

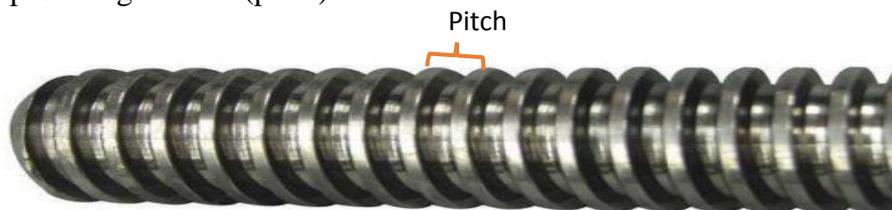


Figure I.12 : Le pas (pitch) de tige filetée.

I.2.2. Extrudeuse :

L'extrudeuse est une partie essentielle dans la composition des imprimantes 3D à dépôt de la matière, elle représente la partie de l'imprimante qui fait l'extrusion de filament plastique fondu au niveau de la tête chauffante et ensuite le déposer sur la plateforme d'impression afin de former l'objet, et ça en ajoutant des couches successives de plastique extrudé. L'extrudeuse est composée de plusieurs parties qui incluent :

- un moteur pour pousser le filament.
- une tête chauffante pour faire fondre le filament plastique.
- une buse d'impression.
- un capteur de température.

La figure I.13 représente un schéma explicatif de différentes parties de l'extrudeuse.

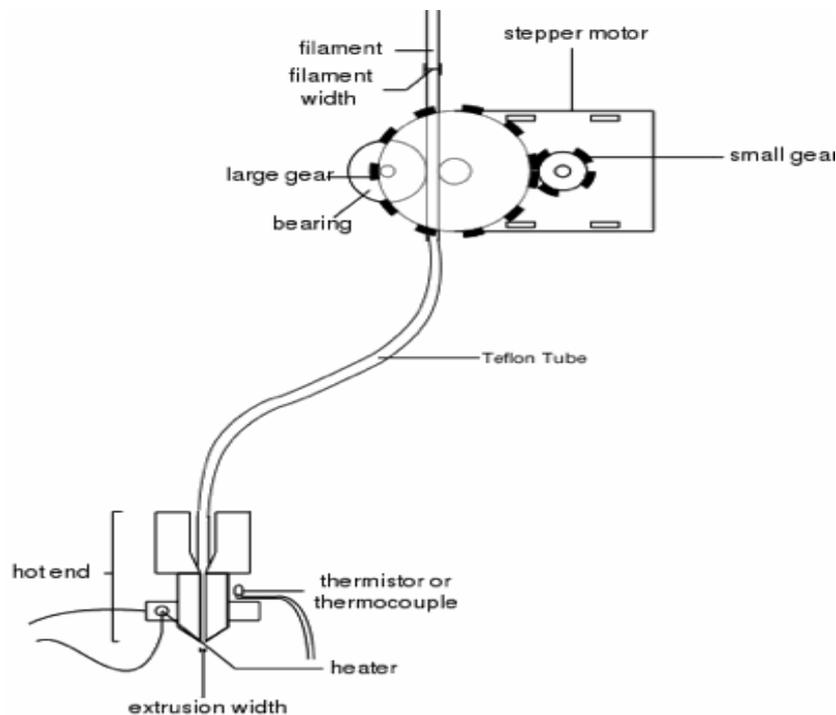


Figure I.13 : Schéma de différentes parties de l'extrudeuse.

Il existe une variété de choix de filaments plastique à utiliser lors de l'impression 3D, le choix de filament dépend de l'objet qu'on veut former et son usage.

On donne dans le tableau suivant les types de filaments et leurs propriétés.

Filament	Propriétés	Température D'impression °C	Température de plateforme D'impression °C
ABS (Acrylonitrile Butadiene Styrene)	- Plastique dure - Facile à imprimer avec. - Il résiste la température.	210-250	80-110
PLA (Polylactic)	- Facile à imprimer avec. - Faible déformation. - Biodégradable	190-230	60-80
Nylon	- Plastique dure et durable. - Résistant. - Flexible.	210-250	60-80
PVA (Polyvinyl alcohol)	- Non toxique. - Facile à utiliser.	180-230	45
CARBON FIBER	- Durable. - Faible déformation. - Soluble.	195-200	50
WOOD	- Souple et flexible. - Durable. - Contient de vraies fibres de bois.	200-260	90-110
HIPS	- Biodégradable. - faible coût.	220-230	50-60

Tableau I.1 : Types et propriétés des filaments.



Figure I.14 : Filament plastique (ABS).

La figure I.15 représente une photo réelle de l'extrudeuse avec désignation de ses différents éléments.

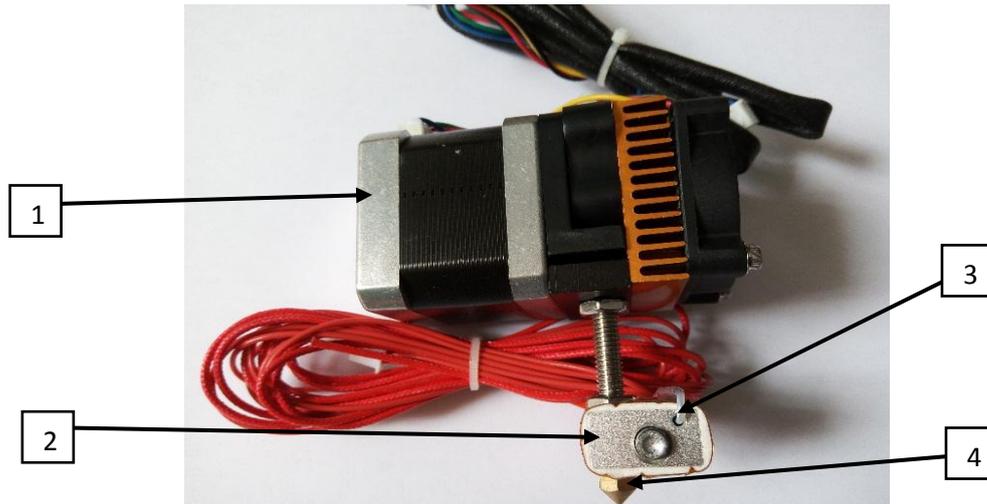


Figure I.15 : Photo réelle de l'extrudeuse.

- (1) : Moteur pas à pas (Stepper Motor).
- (2) : Tête chauffante (Heated Head).
- (3) : Capteur de Température (Temperature Sensor).
- (4) : Buse d'impression (Printing Nozzle).

I.2.3. Les Moteurs pas à pas : [2] [3]

Il existe trois types de moteurs pas à pas : les moteurs bipolaires, les moteurs unipolaires et les moteurs Hybrides. Ces moteurs existent entre différentes tailles, tout dépend des applications dans lesquelles ils seront utilisés.

Ce type de moteur est très courant dans tous les dispositifs où l'on souhaite faire du contrôle de vitesse ou de position en boucle ouverte, typiquement dans les systèmes de positionnement et d'indexation. L'usage le plus connu du grand public est dans les imprimantes classiques et imprimantes 3D, les scanners et les platines vinyle de DJ. Mais ils sont présents dans de nombreuses applications telles : les photocopieurs, imprimantes bancaires, robotique, instrumentation, pompes à perfusion, pousse-seringues, système de positionnement sur machine industrielle et machine-outil.

Dans ce projet on a utilisé cinq moteurs pas à pas de type bipolaire pour assurer le mouvement de l'imprimante 3D sur les 03 axes X, Y, Z. Deux moteurs sont utilisés pour

commander le mouvement sur l'axe Z, un sur l'axe X et un autre sur l'axe Y, ainsi, un dernier moteur pour commander l'extrudeuse.

Ces moteurs bipolaires sont de type NEMA17, ils sont très utilisés dans les imprimantes 3D, vue leur caractéristiques mécaniques, ces moteurs ont un angle de déplacement par pas égale à 1.8° ce qui signifie 200 pas par révolution. La figure I.16 représente les dimensions de moteur NEMA17 et la figure I.16 représente une photo réelle de ce moteur.

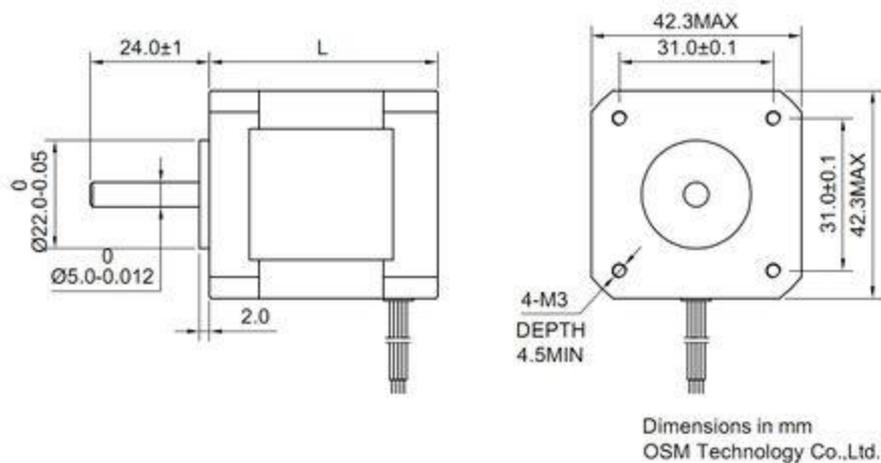


Figure I.16 : Dimensions de moteur Nema 17.

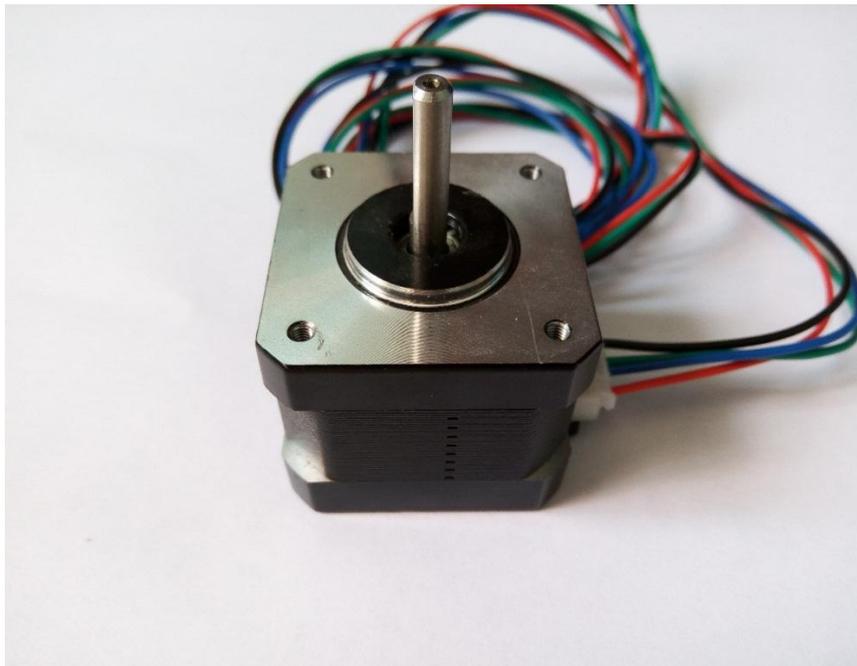


Figure I.17 : Photo réelle de Moteur pas à pas –Nema 17.

I.2.4. Les Capteurs :

Dans ce projet nous avons utilisé deux types de capteurs pour assurer le bon fonctionnement de la machine, ces capteurs sont les suivants :

- capteurs de fin de course.
- capteurs de température.

I.2.4.1. Capteur de fin de course :

Les capteurs de fin de course sont utilisés pour détecter la présence de la tête d'impression sur les extrémités des axes X, Y, Z. ainsi ils sont utilisés comme référence pour déterminer la position de la tête d'impression. Les figures I.18, I.19 représentent respectivement des photos réelles d'un capteur fin de course mécanique et optique.

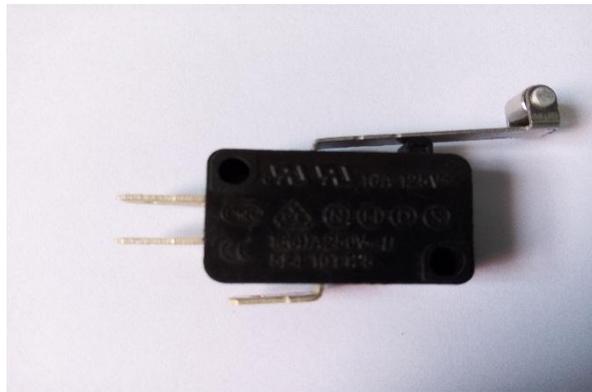


Figure I.18 : Capteur fin de Course mécanique.



Figure I.19 : Capteur fin de Course optique CNY70.

I.2.4.2. Capteur de température (Thermistance) :

I.2.4.2.1 Présentation de la Thermistance : [4]

Les thermistances sont des résistances variables en fonction de la température, la variation de la résistance par degré kelvin ou degré Celsius dépend de la sensibilité de la thermistance. Il existe deux types de thermistances : Positive thermal coefficient PTC et Négative thermal coefficient NTC, la résistance des thermistances PTC augmente avec l'augmentation de la température, tandis que, la résistance des thermistances NTC diminue avec l'augmentation de la température.

Dans ce projet nous avons utilisé la thermistance pour mesurer la température de la tête d'impression. La thermistance utilisée est de type NTC 100 K Ω (100 k Ω à 25 °C) qui a un étendu de mesure entre -55 °C à 250 °C.

La thermistance est reliée au microcontrôleur à travers un montage de diviseur de tension. La figure I.20 montre le schéma de montage de la thermistance.

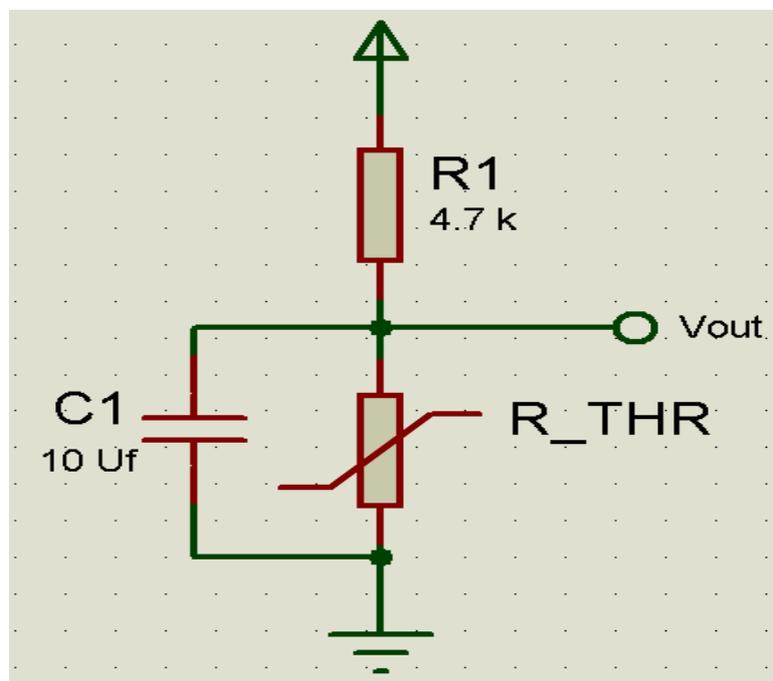


Figure I.20 : Schéma de montage de la thermistance.

La relation entre la variation de la température et la variation de la résistance de thermistance n'est pas linéaire, pour cela nous aurons besoin d'utiliser une équation non linéaire pour trouver un modèle approximatif qui exprime la variation de la résistance en fonction de la variation de la température.

I.2.4.2.2. Modélisation de la thermistance : [4]

Afin de modéliser la thermistance nous avons utilisé l'équation de Steinhart-Hart. Cette équation permet de donner un modèle approximatif pour la thermistance. Le modèle obtenu est valable sur tout l'étendu de mesure de cette dernière.

L'équation de Steinhart-Hart est donnée par :

$$1/T = a + b \cdot \ln(R) + c \cdot [\ln(R)]^3.$$

- **T** : La température en degré Kelvin.
- **R** : La résistance en Ohms.
- **a, b, c** : Les coefficients de Steinhart-Hart qui varient selon le modèle de la thermistance. Ces coefficients sont généralement donnés dans la fiche technique de la thermistance.

Les thermistances NTC peuvent être caractérisées aussi en utilisant l'équation β (β parameter equation) qui est la même équation que l'équation de Steinhart-Hart avec les coefficients a, b et c sont donnés comme suit :

- $a = (1/T_0) - (1/\beta) \cdot \ln(R_0)$,
- $b = 1/\beta$.
- $c = 0$.

On aura donc :

$$1/T = 1/T_0 + (1/\beta) \cdot \ln(R/R_0).$$

$$R = R_0 \cdot e^{-\beta \cdot ((1/T_0) - (1/T))}.$$

I.3. Mode Opérateur :

L'impression de l'objet en 3D est faite sur 04 étapes qu'on les résume comme suit :

❖ La première étape :

On doit d'abord préparer le fichier 3D de l'objet qu'on souhaite imprimer. Ce fichier 3D peut être créé à partir d'un logiciel de CAO ou de CAD comme SOLIDWORKS ou AUTOCAD ou directement téléchargé à partir d'un site internet spécialisé. Une fois que cette étape est vérifiée on passe à la seconde étape.

❖ La deuxième étape :

Cette étape concerne le processus d'impression. Dans un premier temps, on doit choisir la matière à utiliser et ça en fonction des propriétés de l'objet. La gamme des matériaux utilisés dans la recherche et le développement de l'impression 3D est immense. Cette dernière comprend différents Plastiques, Céramiques, Résines, Métaux, Sables, Textiles, Matériaux biologiques, le Verre, différents types de nourritures et même de la poussière Lunaire. Cependant dans ce projet nous utilisons seulement le PLA et souvent l'ABS comme matière d'impression et ça revient à leur accessibilité et disponibilité dans le marché ainsi les caractéristiques de l'extrudeuse utilisée.

❖ La troisième étape :

C'est la mise en marche de l'imprimante en l'interfaçant avec le PC via une application desktop (interface Homme-Machine). On attend jusqu'à ce qu'ils soient connectés. Après on introduit tous les paramètres nécessaires comme la matière utilisée et la température d'impression. L'interface Homme-Machine envoie le Code G de l'objet 3D à la carte de commande ligne par ligne, ainsi, il visualise les différents paramètres comme l'état de l'imprimante et l'évolution de l'impression en cours de temps.

❖ La quatrième étape :

Quand l'impression est terminée, on passe à l'étape de la finition où l'objet imprimé doit être vernis ou peint afin d'obtenir son aspect final.

CONCLUSION :

Dans ce chapitre nous avons abordé les différentes techniques et procédés utilisées dans l'impression 3D, ainsi, nous avons fait le choix de la technique sur laquelle on va travailler.

Nous avons également effectué une description détaillée et présenter l'architecture de l'imprimante 3D et ses différents parties qui la composent et sa méthode de fonctionnement.

II

CHAPITRE

Description du Hardware

INTRODUCTION :

Après avoir traité dans le premier chapitre la description de procédé à commander et son architecture, nous entamons dans ce chapitre l'étude de la partie hardware de projet en présentant la carte électronique de commande ainsi que, la description détaillée des différents éléments qui la constituent et leurs schémas de brochages.

II.1. Description de la carte électronique de commande réalisée :

Le système de commande que nous avons réalisé est une carte électronique à microcontrôleur qui assure le control du procédé. Elle est composée de deux parties essentielles : le microcontrôleur qui est responsable de la partie commande, communication et acquisition des données et le circuit électronique contenant la partie puissance de la carte qui sert à convertir les signaux de commande à des signaux de puissance pour piloter l'ensemble des moteurs, l'extrudeuse et la tête chauffante. La figure II.1 montre le schéma fonctionnel de la carte de commande.

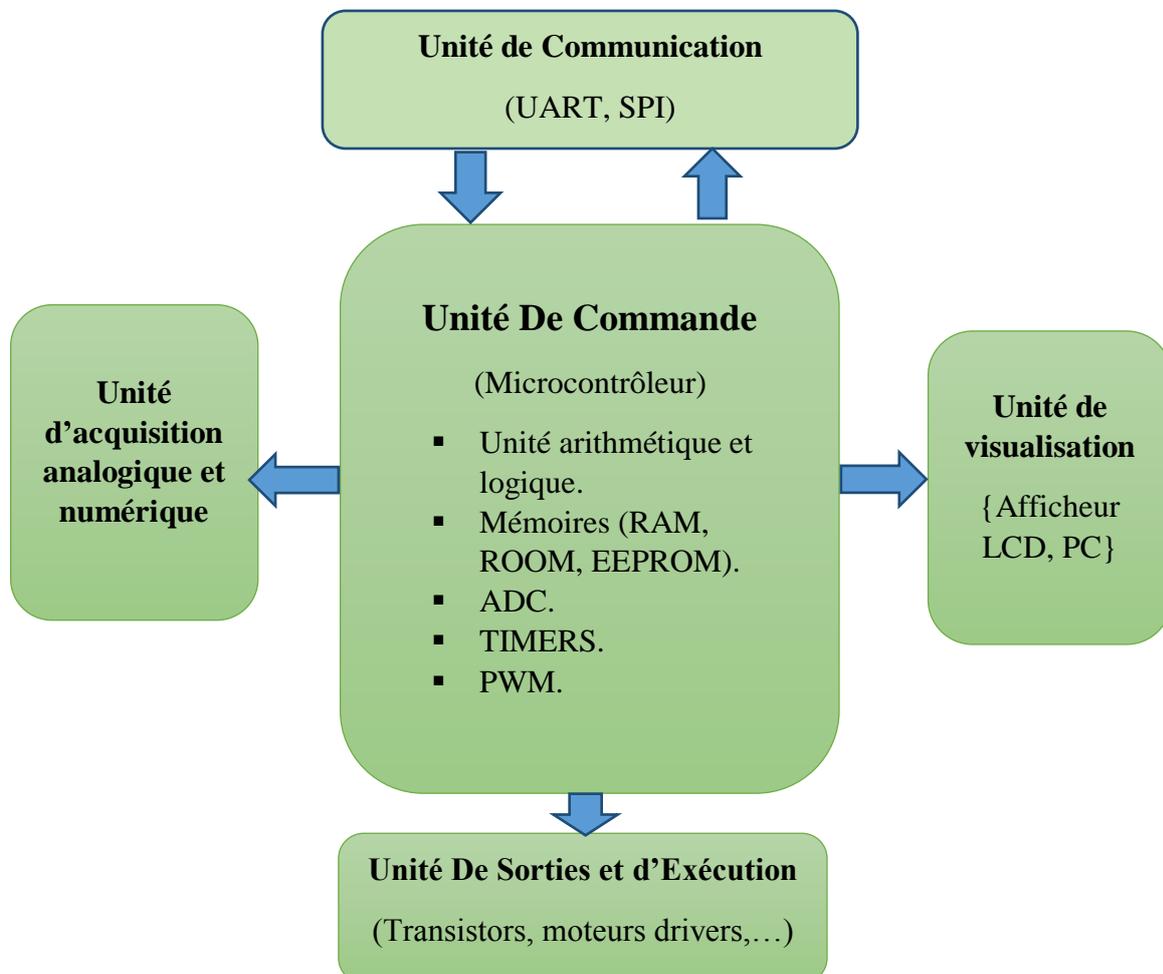


Figure II.1 : Schéma fonctionnel de la carte de commande.

La carte électronique est constituée principalement des éléments et des unités suivants :

- La carte de développement ARDUINO MEGA à base de microcontrôleur Atmega2560.
- Une interface de communication avec le PC via une liaison USB to UART intégrée dans la carte de développement ARDUINO MEGA.
- Une interface de communication avec la carte mémoire via une liaison SPI.
- Un organe de visualisation de type afficheur LCD.
- Une interface d'acquisition de données analogiques et numériques via des capteurs de températures (thermistance NTC 100k) et de fin de cours.
- Unité de sorties et d'exécution qui représente la partie puissance de la carte électronique composée principalement de transistors MOSFET et les drivers des moteurs pas à pas A4988.

Pour alimenter la carte électronique nous utilisons une alimentation à découpage de 12V avec un régulateur de type LM7805 pour délivrer les 05 volts nécessaires à l'alimentation de microcontrôleur Atmega2560. La figure II.2 représente le schéma de brochage de la carte électronique.

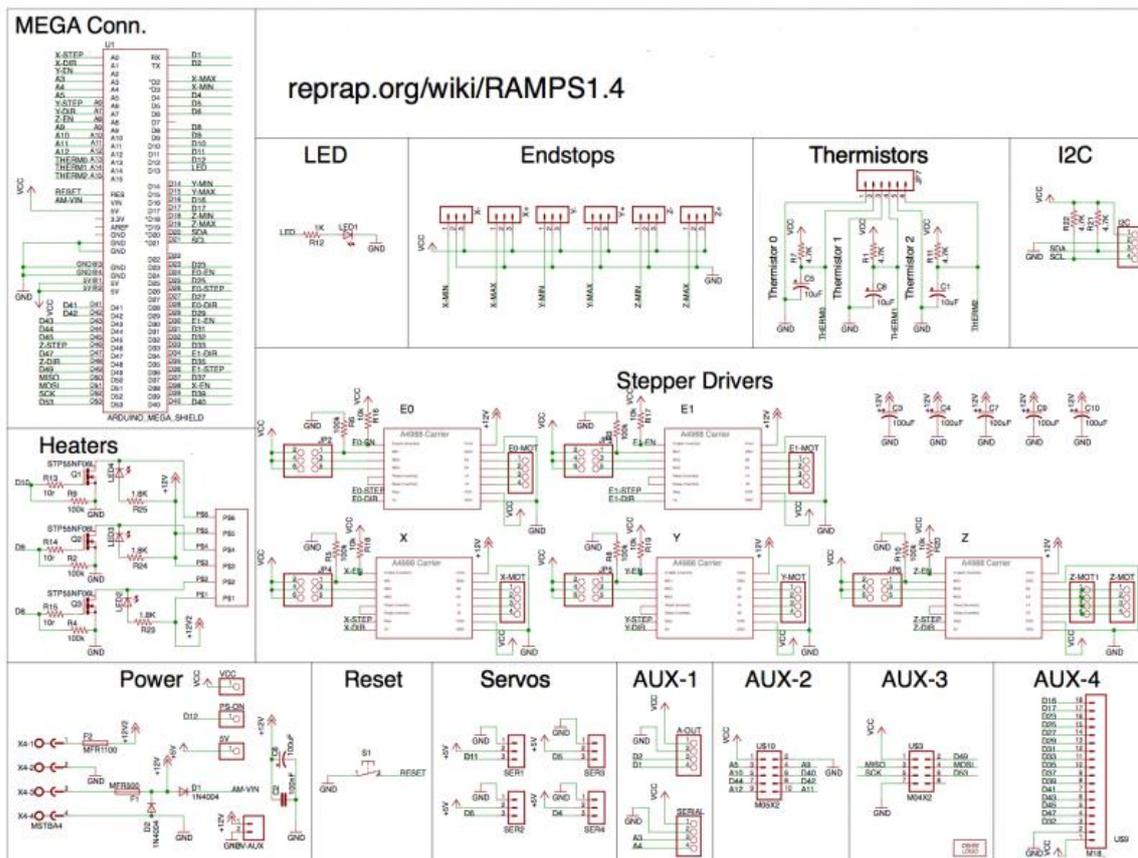


Figure II.2 : Schéma de brochage de la carte électronique.

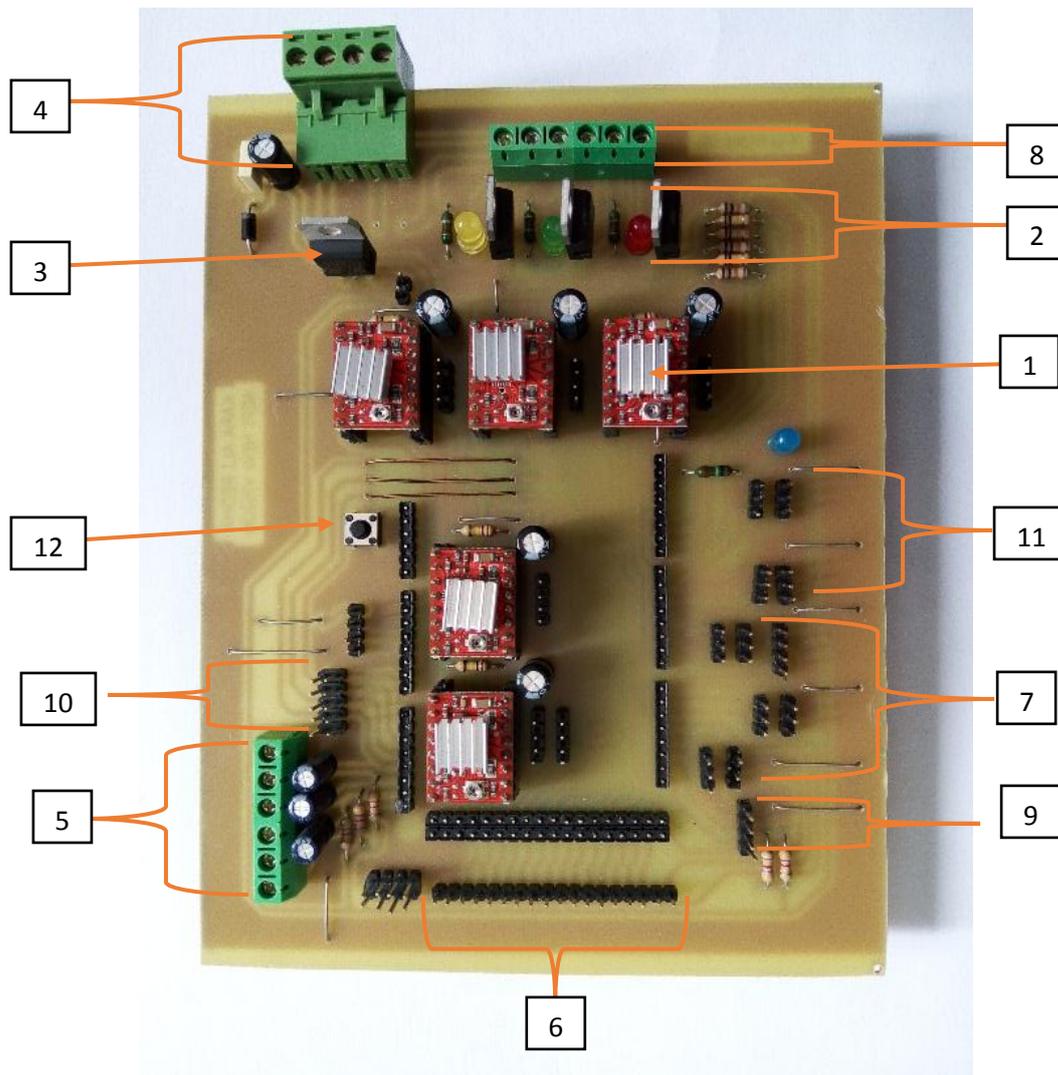


Figure II.3 : Photo réelle de la carte électronique avec désignation de ces différents éléments.

- (1) : Le driver de puissance A4988.
- (2) : Les transistors à effet de champs (Mosfet) STP55NF06.
- (3) : Le régulateur de tension 5V LM7805.
- (4) : Connecteurs d'alimentation 12V.
- (5) : Connecteurs pour relier les thermistances.
- (6) : Interface de connexion avec l'afficheur LCD.
- (7) : Interface pour connecter les capteurs de fin de course.
- (8) : Connecteurs pour relier la tête chauffante et le ventilateur.
- (9) : Interface pour connexion I2C.
- (10) : Interface pour connecter la carte mémoire (connexion SPI).
- (11) : Connecteurs supplémentaires.
- (12) : Bouton de Reset.

II.1.1. Le Microcontrôleur : [6]

Un microcontrôleur (en notation abrégée μc , **MCU** en anglais) est un circuit intégré qui rassemble les éléments essentiels d'un ordinateur : processeur, mémoires (RAM et ROM), unités périphériques et interfaces d'entrées-sorties. Les microcontrôleurs se caractérisent par un plus haut degré d'intégration, une plus faible consommation électrique, une vitesse de fonctionnement plus faible (de quelques mégahertz jusqu'à plus d'un gigahertz) et un coût réduit par rapport aux microprocesseurs polyvalents utilisés dans les ordinateurs personnels.

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

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

II.1.1.1 Choix du microcontrôleur :

Il y'en a plusieurs fabricants des semi-conducteurs dans le monde et chaque fabricant a son propre famille de microcontrôleur et son propre architecture, par exemple on trouve les microcontrôleurs AVR de ATMEL et les microcontrôleurs PIC de Microchip, de Motorola, d'Intel ...etc.

Notre choix s'est porté sur la carte de développement Arduino Mega qui est à base de microcontrôleur ATMEL Atmega2560 et cela pour des différentes raisons que je les résume comme suit :

- La disponibilité.
- Connait une large utilisation actuellement.
- Peu coûteux.
- Performant.
- Outil de développement open source.

On trouve plusieurs IDE qui sont dédiés à la programmation de ce Microcontrôleur comme : ARDUINO IDE, AVR studio et ATMEL Studio.

II.1.1.2. Présentation de la carte de développement Arduino Mega : [5] [6]

La carte de développement Arduino Mega est une carte électronique à base de microcontrôleur Atmega2560, elle contient tous les éléments nécessaires pour le fonctionnement de microcontrôleur (Quartz 16 M, bouton reset, régulateur 5 et 3.3 V). De plus, cette carte dispose :

- Un connecteur USB.
- Un connecteur ICSP (programmation "in-circuit").
- Le circuit Atmega16u2.

Le circuit Atmega16u2 sert comme un convertisseur USB vers UART, il assure la connexion entre la carte Arduino Mega et le PC via cette liaison série.

Le microcontrôleur ATmega2560 présent sur la carte Arduino Mega est livré avec un bootloader (petit programme de démarrage) préprogrammé qui permet de transférer le nouveau programme dans le microcontrôleur sans avoir utiliser un matériel de programmation externe. Cependant, on peut toujours utiliser un programmeur ICSP pour charger le programme dans le microcontrôleur, ces programmeurs ICSP utilisent généralement une liaison SPI pour charger le programme. Cette liaison elle peut exister comme une véritable ressource interne comme elle peut être seulement présente pour cette phase de programmation.

L'utilité de la carte de développement Arduino Mega est dans la possibilité de l'utiliser comme programmeur pour le microcontrôleur Atmega2560 d'un côté et d'un autre elle nous permet de bénéficier de différents outils existant sur le logiciel Arduino IDE. De plus, son prix est moins cher que le prix des programmeurs ICSP. La figure II.4 donne une photo réelle de la carte de développement Arduino Méga avec désignation de ces éléments.



Figure II.4 : Photo réelle de la carte de développement Arduino Méga.

- (1) : Microcontrôleur Atmega2560.
- (2) : Circuit Atmega16u2.
- (3) : Connecteur USB.
- (4) : Connecteur ICSP.
- (5) : Entrée d'alimentation.
- (6) : Quartz 16 M.

II.1.1.3. Présentation du microcontrôleur Atmega2560 : [7]

Le microcontrôleur Atmega2560 est un microcontrôleur 8 bits de la famille AVR dont la programmation est faite avec le langage C.

Les principales caractéristiques de microcontrôleur ATMEL Atmega2560 sont les suivantes :

- **Mémoire Flash** : 256 Ko.
- **Mémoire SRAM** : 8 Ko.
- **EEPROM** : 4 Ko.
- **Fréquence maximale** : 16 MHz.
- **Nombre des E/S** : 54 pins entrées/sorties programmables.
- **Timers /Counters** : deux timers de 08 bits : timer0 et timer2 et quatre timers de 16 bits : timer1, timer3, timer4 et timer5.
- **Watchdog timer programmable.**
- **PWM** : Quatre canaux de PWM de 08 bits.
- Six/onze chaînes de PWM avec une résolution programmable de 02 à 16 bits.
- **UART** : Deux/quatre sorties UART programmables.
- **Master/slave SPI interface.**
- **Interface de communication I2C.**
- **ADC** : 16 entrées analogiques avec une résolution de 10 bits.
- **Six sleep mode.**
- **Interruption** : il y'en a 57 sources d'interruption possibles qu'on les résume comme suit :
 - Interruption liées au changement d'état des pins.
 - Interruptions liées aux timers, UsART, SPI, I2C.
 - Interruption liée au Watchdog.
 - Interruptions liées aux comparateurs analogiques.

I.1.1.4. Architecture interne de l'Atmega2560 : [7]

La figure II.5 représente le schéma de l'architecture interne du microcontrôleur.

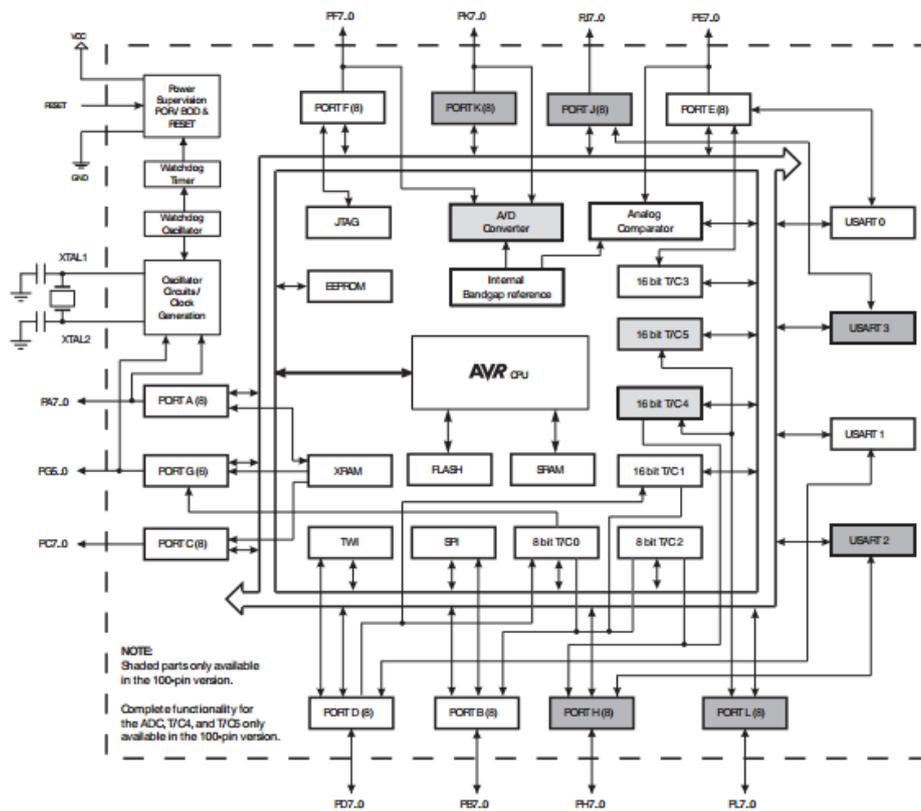


Figure II.5 : Architecture interne de l'Atmega2560.

I.1.1.5. Fonction des différents pins du microcontrôleur Atmega2560 : [7]

La figure II.6 représente une photo réelle de l'atmega2560.



Figure II.6 : Photo réelle de l'atmega2560.

Les pins de Atmega2560 peuvent être configurés comme des entrées/sorties numériques comme ils peuvent être configurés aussi pour faire d'autres fonctions alternatives, le tableau II.1 donne les fonctions alternatives des pins de l'atmega2560.

Nom du PORT	NOM Du pin	Fonction Alternative
PORT A	PA0	AD0 (External memory interface address and data bit 0)
	PA1	AD1 (External memory interface address and data bit 1)
	PA2	AD2 (External memory interface address and data bit 2)
	PA3	AD3 (External memory interface address and data bit 3)
	PA4	AD4 (External memory interface address and data bit 4)
	PA5	AD5 (External memory interface address and data bit 5)
	PA6	AD6 (External memory interface address and data bit 6)
	PA7	AD7 (External memory interface address and data bit 7)
PORTB	PB0	SPI Slave Select input or Pin Change Interrupt 0
	PB1	SPI Bus Serial Clock or Pin Change Interrupt 1
	PB2	SPI Bus Master Input/Slave Output or Pin Change Interrupt 2
	PB3	SPI Bus Master Input/Slave Output or Pin Change Interrupt 3
	PB4	Output compare, PWM, pin change interrupt 4.
	PB5	Output compare, PWM, pin change interrupt 5.
	PB6	Output compare, PWM, pin change interrupt 6.
	PB7	Output compare, PWM, pin change interrupt 7.
PORTC	PC0	A08 (External Memory interface address bit 08)
	PC1	A09 (External Memory interface address bit 09)
	PC2	A10 (External Memory interface address bit 10)
	PC3	A11 (External Memory interface address bit 11)
	PC4	A12 (External Memory interface address bit 12)
	PC5	A13 (External Memory interface address bit 13)
	PC6	A14 (External Memory interface address bit 14)
	PC7	A15 (External Memory interface address bit 15)
PORT D	PD0	External Interrupt0 Input or T WI Serial CLock
	PD1	External Interrupt1 Input or T WI Serial data
	PD2	External Interrupt2 Input or USART1 Receive Pin

	PD3	External Interrupt3 Input or USART1 Transmit Pin
	PD4	ICP1 (Timer/Counter1 Input Capture Trigger)
	PD5	XCK1 (USART1 External Clock Input/Output)
	PD6	T1 (Timer/Counter1 Clock Input)
	PD7	T0 (Timer/Counter0 Clock Input)
PORT E	PE0	USART0 Receive Pin, Pin Change Interrupt 8
	PE1	USART0 Transmit Pin
	PE2	Analog Comparator Positive Input, USART0 external clock
	PE3	Aanalog comparator negative, PWM
	PE4	External Interrupt 4 Input, Output Compare, PWM
	PE5	External Interrupt 5 Input, Output Compare, PWM
	PE6	External Interrupt 6 Input or Timer/Counter3 Clock Input
	PE7	External Interrupt 7 Input, Timer/Counter3 Input trigger
PORT F	PF0	ADC input channel 0
	PF1	ADC input channel 1
	PF2	ADC input channel 2
	PF3	ADC input channel 3
	PF4	ADC input channel 4
	PF5	ADC input channel 5
	PF6	ADC input channel 6
	PF7	ADC input channel 7
PORT G	PG0	RTC Oscillator Timer/Counter2
	PG3	RTC Oscillator Timer/Counter2
	PG4	Output Compare, PWM
PORT H	PH0	USART2 Receive Pin
	PH1	USART2 Transmit Pin
	PH2	USART2 External Clock
	PH3	Output Compare, PWM
	PH4	Output Compare, PWM
	PH5	Output Compare, PWM
	PH6	Output Compare, PWM
	PH7	Timer/Counter4 Clock Input
	PJ0	USART3 Receive Pin, Pin Change Interrupt 9

PORT J	PJ1	USART3 Transmit Pin, Pin Change Interrupt 10
	PJ2	USART3 External Clock, Pin Change Interrupt 11
	PJ3	Pin Change Interrupt 12
	PJ4	Pin Change Interrupt 13
	PJ5	Pin Change Interrupt 14
	PJ6	Pin Change Interrupt 15
PORT K	PK0	ADC Input Channel 08, Pin Change Interrupt 16
	PK1	ADC Input Channel 09, Pin Change Interrupt 17
	PK2	ADC Input Channel 10, Pin Change Interrupt 18
	PK3	ADC Input Channel 11, Pin Change Interrupt 19
	PK4	ADC Input Channel 12, Pin Change Interrupt 20
	PK5	ADC Input Channel 13, Pin Change Interrupt 21
	PK6	ADC Input Channel 14, Pin Change Interrupt 22
	PK7	ADC Input Channel 15, Pin Change Interrupt 23
PORT L	PL0	Timer/Counter4 Input Capture Trigger
	PL1	Timer/Counter5 Input Capture Trigger
	PL2	Timer/Counter5 Clock Input
	PL3	Output Compare, PWM
	PL4	Output Compare, PWM
	PL5	Output Compare, PWM

Tableau II.1 : Les fonctions alternatives des pins de l'Atmega2560.

II.1.2. Le circuit de puissance A4988 :

L'A4988 est un circuit intégré dédié à la commande des moteurs pas à pas bipolaires. Ce driver permet de faire fonctionner le moteur dans cinq modes de résolutions différents ainsi, il peut délivrer un courant jusque' à 2A par phase.

II.1.2.1. Caractéristiques et détails techniques de circuit A4988 : [8]

Voici quelques caractéristiques essentielles de ce driver de moteur pas à pas :

- Tension de fonctionnement minimum : 8 V.
- Tension maximale de fonctionnement : 35 V.
- Courant par phase (en continu) : 1,2 A.
- Courant maximum par phase : 2.3 A.
- Logique de commande :
 - Tension minimale : 3 V.
 - Tension maximale : 5.5 V.
- Interface de contrôle de pas et de direction simple et facile.
- 05 résolutions différentes de pas :
 - Full-step : pas complet.
 - Halfstep : $\frac{1}{2}$ pas.
 - Quarter-step : $\frac{1}{4}$ pas.
 - Eighth-step : $\frac{1}{8}$ ième de pas.
 - Sixteenth-step : $\frac{1}{16}$ ième de pas.
- Contrôle en courant ajustable - un potentiomètre permet de fixer le courant maximum à l'aide d'un potentiomètre. Cela permet d'utiliser une tension supérieure à celle recommandée par le moteur (puisque le courant est limité) et permet ainsi d'atteindre un débit de pas plus élevé (et donc une vitesse plus élevée).
- Une commande de hachage intelligente qui sélectionne automatiquement le bon mode de décroissance de courant (fastdecay = baisse/chute rapide du courant OU slow decay = chute lente).
- Protection (arrêt) en cas de surchauffe, sous tension, surcharge/sur-courant/court-circuit. Utile en cas de surcharge/blocage moteur (car le courant va monter en flèche, ce qui doit provoquer la mise en protection).

II.1.2.2. Brochage et description des pins de driver A4988 : [8]

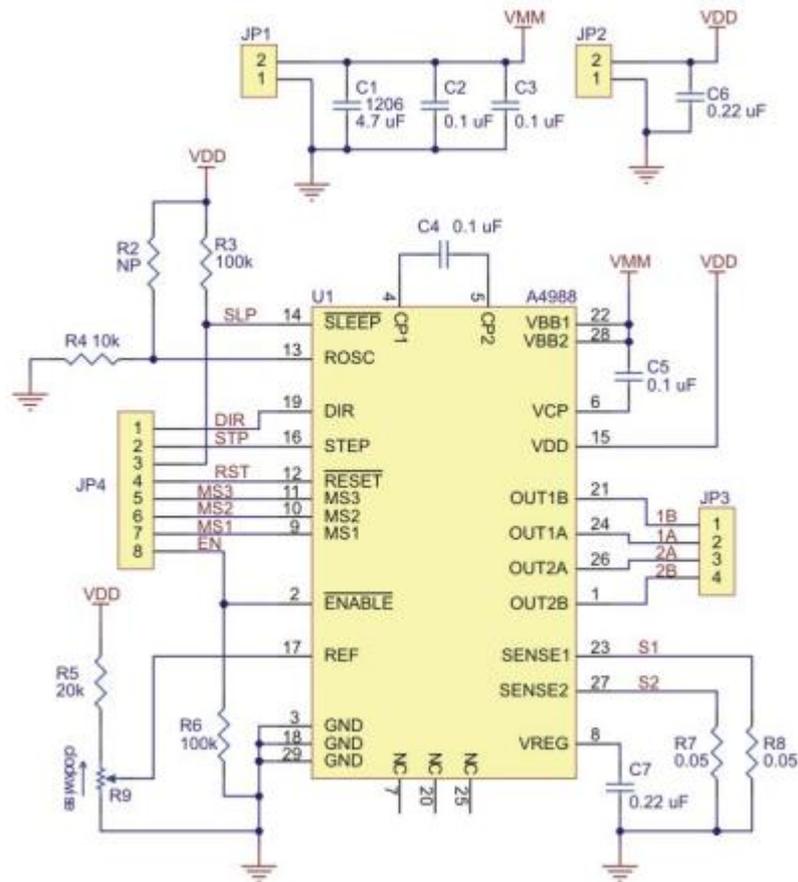


Figure II.7 : Schéma diagramme de driver A4988.

- **ENABLE** : Logique Inversée, ce pin permet d'activer ou désactivé le moteur. Quand il est mis dans un état haut (5V) le moteur est désactivé et quand il est mis en état bas (0V) le moteur est activé.
- **MS1, MS2, MS3** : Ces pins permettent de sélectionner la configuration de mode de résolution de moteur pas à pas. Ces broches disposent de résistances Pull-Down ramenant le potentiel à 0v lorsque rien n'est connecté sur ces broches (ils sont connectés au GND par défaut).
- **RESET** : Logique inversée. Permet de faire une réinitialisation du module. Généralement connecté sur la broche "SLEEP".
- **SLEEP** : Logique inversée. Généralement connecté sur la broche "Reset" du module.

- **STEP** : En l'envoyant un signal d'horloge (haut puis bas) le moteur avancera d'un pas.
- **DIR** : Il permet de sélectionner la direction de rotation du moteur soit dans le sens horaire (CW) ou le sens antihoraire (CCW) et cela selon l'état de pin.
- **VMOT** : Tension d'alimentation du moteur. Habituellement 12V pour les moteurs pas à pas. Tension entre 8 et 12v.
- **GND** : C'est la masse (GROUND). Habituellement on mit toutes les masses en commun.
- **1B 1A** : Première bobine du moteur pas à pas bipolaire.
- **2A 2B** : Deuxième bobine du moteur pas à pas bipolaire.
- **VDD** : Alimentation de la logique de commande entre 3 et 5.5v. Habituellement 5V.

La **Figure II.8** représente le schéma de brochage de circuit A988.

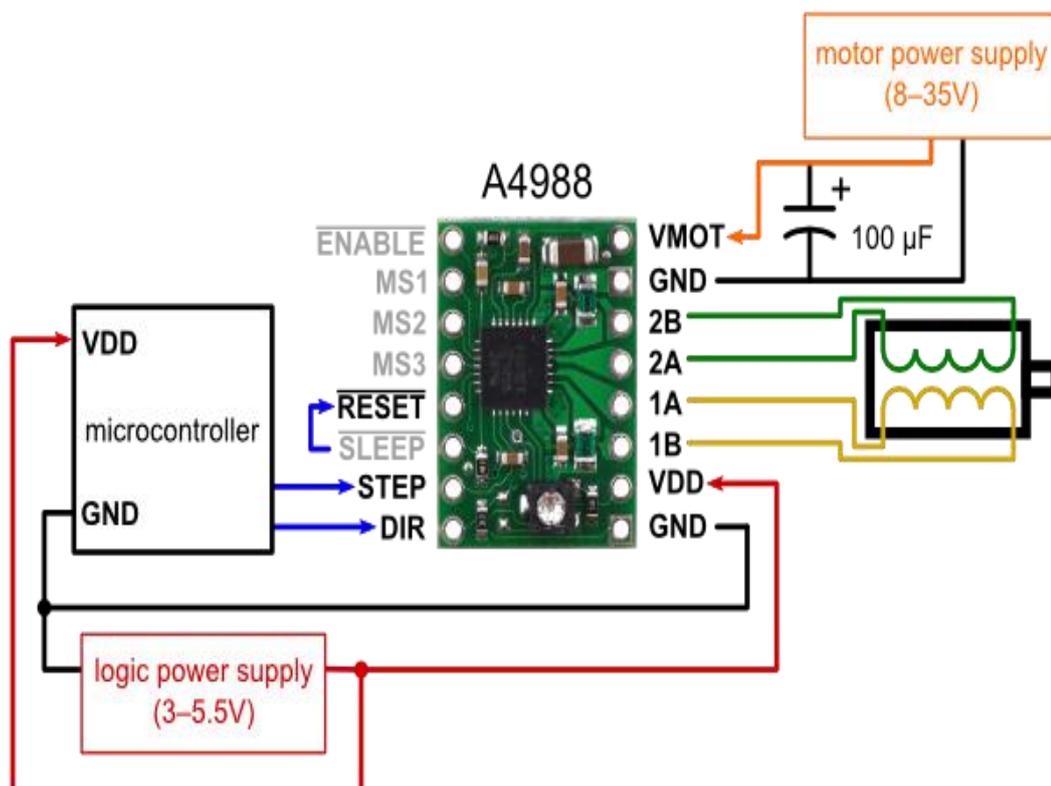


Figure II.8 : Schéma de brochage de driver A4988.

II.1.2.3. Ajustement de Driver A4988 : [9]

❖ Ajustement des entrées du contrôle :

Chaque impulsion sur l'entrée STEP correspond à un pas ou micro pas du moteur pas-à-pas dans la direction sélectionnée sur la broche DIR. Les broches STEP et DIR ne disposent pas de résistances pull-down ou pull-up interne. Donc le contrôle de deux états Haut et Bas de ces entrées est fait directement par le microcontrôleur. La tension sur ces broches ne doit jamais être flottante.

Le circuit dispose de trois entrées différentes pour contrôler son étage de puissance : RST (Reset), SLP (SLEEP) et EN (ENABLE), On peut connecter la broche RST avec la broche SLP pour la ramener à l'état haut et donc activer la carte.

❖ Configuration de mode de résolution :

Les moteurs pas-à-pas disposent de leur propre spécification physique de "pas" connu comme un "pas complet". Un pilot Micro-pas tel que l'A4988 permet d'obtenir une plus grande résolution en autorisant des positions intermédiaires dans un pas. Cela est rendu possible en modulant intelligemment la quantité de courant dans les bobines du moteur pas-à-pas. Par exemple, piloter un moteur en mode "1/4 de pas" permet d'obtenir 800 micro-pas sur un moteur prévu pour 200 pas par révolution et cela en utilisant 4 niveau de courants différents pour chacun des Microsteps. La configuration de mode de résolution se fait à l'aide des broches MS1, MS2 et MS3.

- MS1 et MS3 disposent d'une résistance pull-down interne de 100k Ω . Une résistance pull-down signifie que si on ne place pas ces broches au niveau logique haut : VDD (la tension choisie pour la logique de commande) alors elles seront automatiquement ramenées au niveau logique bas.
- MS2 dispose d'une résistance pull-down de 50k Ω .

Si les broches MS1, MS2 et MS3 sont laissées flottantes, l'A4988 fonctionnera en mode "pas complet". Pour que le Microstepping fonctionne correctement, il faut que le courant soit limité d'une façon à ne pas activer la protection en sur-courant. Sinon, les niveaux des courants intermédiaires ne seront pas correctement maintenus et le moteur pourrait sauter des microsteps.

Le **tableau II.2** représente le tableau de configuration de mode de résolution.

MS1	MS2	MS3	Mode de résolution
LOW	LOW	LOW	Pas complet.
HIGH	LOW	LOW	½ pas.
LOW	HIGH	LOW	¼ pas.
HIGH	HIGH	LOW	1/8 pas.
HIGH	HIGH	HIGH	1/16 pas.

Tableau II.2 : La configuration de mode de résolution.

❖ **Limitation du courant :**

Un moteur pas à pas typique peut avoir un courant maximal égale à un Ampère avec une résistance de bobine égale à 5Ω , ce qui exige une alimentation de 5V ($U=R*I$), l'utilisation de tel moteur avec une alimentation de 12V va l'endommager car le débit de courant sera très important. Le circuit A4988 permet de limiter ce courant à l'aide d'un petit potentiomètre disponible sur la carte, pour déterminer la valeur de courant limite on suit les étapes suivantes :

- On place le pilote en mode de résolution « pas complet » et on mesure le courant passant dans l'une des deux bobines sans envoyer un signal d'horloge sur la broche STEP.
- On calcule le courant limite qui est équivalent à 0.7 fois le courant mesuré
 $I_{\text{Mesuré}} = 0.7 * I_{\text{Limite}}$.

Une autre façon d'ajuster le courant limite est de mesurer la tension sur la broche « REF » et de calculer le courant limite par la méthode suivante :

$$I_{\text{Limite}} = V_{\text{REF}} * 2,5.$$

Par exemple si la tension de référence est de 0.3 V, le courant limite est de 0.75 A. Le courant dans les bobines sera limité à 70% du courant limite. Donc, pour obtenir un courant de 1 A en mode « pas complet » le courant limite devrait être égale à 1.4 A, ce qui correspond à un V_{REF} de 0.56 V.

NB : Pour mesurer le courant le multimètre doit être placé en série avec la bobine de moteur pas à pas.

II.1.3. Afficheur LCD : [10]

Un afficheur LCD (Liquid Cristal Display) Alphanumérique (2 lignes x 16 caractères) est interfacé avec la carte électronique pour nous permettre de visualiser en temps réelle l'évolution de processus d'impression et l'état d'imprimante (position de la tête d'impression, température de la tête chauffante, ... etc.).

II.1.3.1. Présentation de l'afficheur :

Le type d'afficheur étudié permet d'afficher :

- tous les caractères ASCII internationaux : chiffres et lettres, caractères de ponctuations, opérateurs arithmétiques, etc.
- les caractères nationaux. Le jeu de caractère dépend de la version du contrôleur.
- des caractères définis par l'utilisateur (jusqu'à 8 caractères définis par l'utilisateur).
- L'afficheur permet aussi de placer un curseur ou de faire clignoter le caractère à l'endroit où le prochain caractère doit être affiché. On peut réaliser des animations en décalant la fenêtre d'affichage.

L'afficheur est constitué de :

- Un CI contrôleur Hitachi HD44780 ou équivalent.
- Un ou plusieurs CI unité d'extension (le contrôleur seul ne permet que de piloter 2 lignes de 8 caractères) : un CI pour 2 lignes de 16 caractères.
- une matrice LCD de 2 lignes de 16 caractères.

Le module dispose aussi de :

- signaux de liaison avec le μ C ou le système à μ P de commande : DB7-DB0, EN, R/W, RS.
- 2 broches pour l'alimentation VDD et VSS ($VDD - VSS = 5V$).
- une broche pour régler de contraste de l'affichage (V0).
- parfois deux broches pour l'alimentation de LEDs pour le rétro-éclairage.

La figure II.9 représente la structure interne simplifiée d'un LCD 2*16.

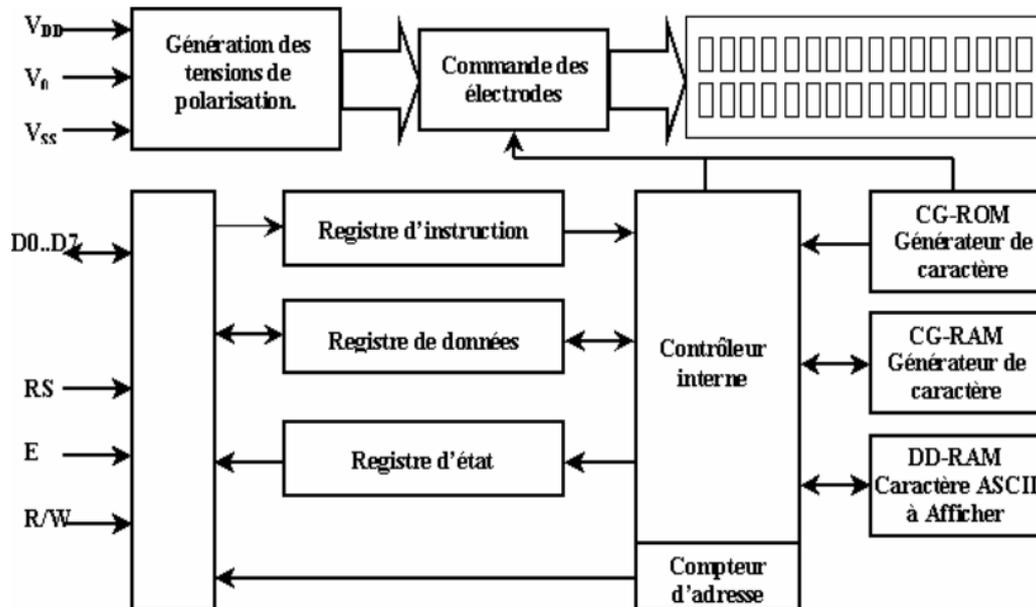


Figure II.9 : La structure interne simplifiée d'un LCD 2*16.

II.1.3.2. Brochage et description des pins :

Le tableau II.3 donne la description des broches du LCD.

Numéro de pin	Symbole	Type	Description
01	GND	Alimentation	0V
02	VDD	Alimentation	5V
03	VEE	Alimentation	Réglage de contraste
04	RS	Entrée	Registre select
05	R/W	Entrée	Write/Read
06	E	Entrée	Enable
07	DB0	Entrée/sortie	Data bit 0
08	DB1	Entrée/sortie	Data bit 1
09	DB2	Entrée/sortie	Data bit 2
10	DB3	Entrée/sortie	Data bit 3
11	DB4	Entrée/sortie	Data bit 4
12	DB5	Entrée/sortie	Data bit 5
13	DB6	Entrée/sortie	Data bit 6
14	DB7	Entrée/sortie	Data bit 7

Tableau II.3 : Description des broches du LCD.

NB1 : Pour régler le contraste on applique une tension variable sur le pin VEE.

NB2 : Lors de l'utilisation en mode 4 bits seul les bits D4 à D7 sont utilisés. Les lignes correspondant de D0 à D3 sont laissées flottantes.

Les figures II.10 et II.11 montrent une photo réelle de l'afficheur LCD utilisé ainsi que son schéma de brochage.



Figure II.10 : Photo réelle de l'afficheur LCD.

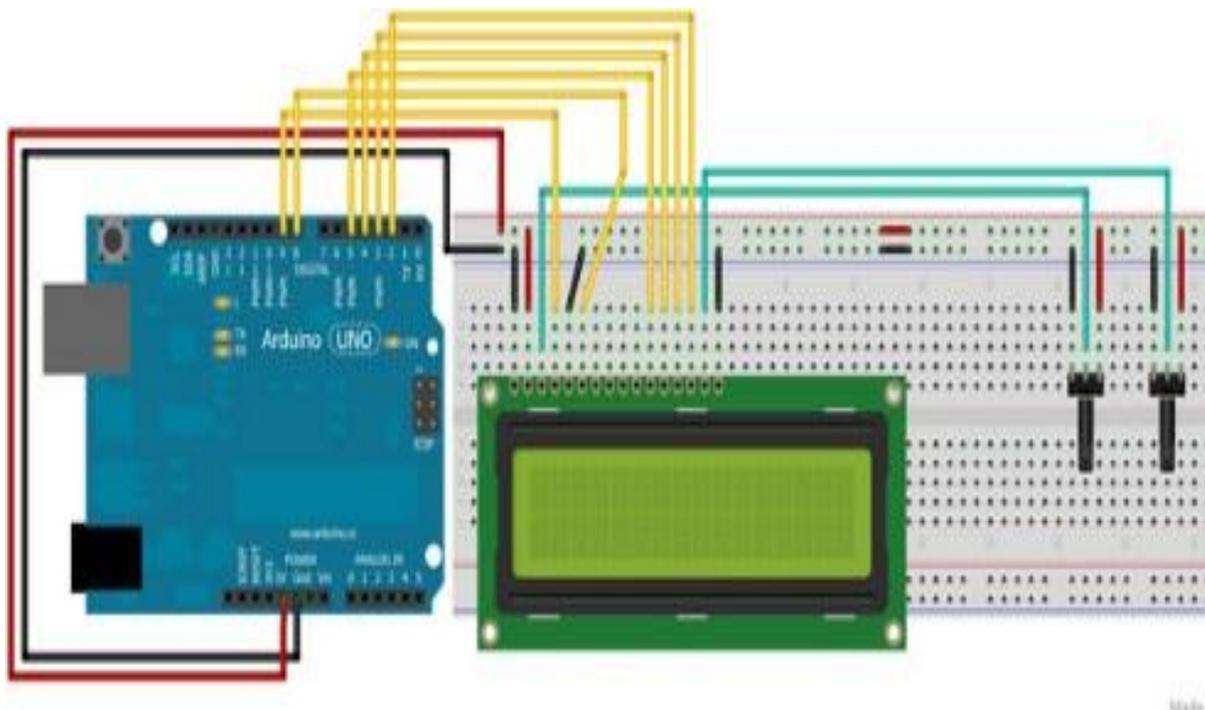


Figure II.11 : Schéma de brochage simplifié de l'afficheur LCD.

II.1.4. Le Transistor Mosfet : [11]

Afin de commander la tête chauffante et la plateforme d'impression il est nécessaire et obligatoire de les séparer du microcontrôleur pour ne pas l'endommager, pour cela j'ai utilisé des transistors de type Mosfet (transistor à effet de champ) pour séparer la partie commande de la partie puissance.

Les transistors Mosfet font partie de la famille de semi-conducteurs, leur particularité est d'utiliser un champ électrique pour contrôler la conductivité et donc ils nécessitent peu de puissance pour leur commande. On trouve deux types de transistors Mosfet : Type "P" et Type "N", la commande de transistor Mosfet se fait en appliquant une tension V_{GS} négative dans le cas d'un type P et une tension positive dans le cas d'un type N. Le Mosfet utilisé dans ce projet et de type N dans la référence est STP55NF06. La figure II.12 représente une photo réelle de transistor STP55NF06 et le tableau II.4 donne ses caractéristiques.

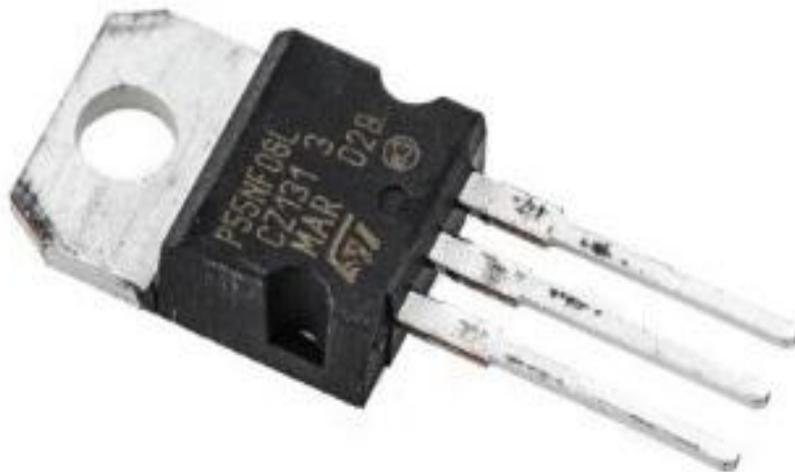


Figure II.12 : Photo réelle de transistor STP55NF06.

Symbole	Paramètre	Valeur
V_{DS}	Le voltage du Drain.	60 V.
V_{GS}	Le voltage du Gate.	+/- 16 V.
I_D	Le courant du Drain.	55 A.
I_{DM}	Le courant du Drain pulsé.	220 A.
R_{DS}	La résistance statique.	0.016 Ω .

Tableau II.4 : Caractéristiques de transistor STP55NF06.

II.2. Techniques de communication et d'interfaçage :

La mise en marche de l'imprimante nécessite qu'elle soit reliée au PC et /ou la carte mémoire contenant le fichier G-code de l'objet à imprimer, pour cela nous faisons appel à des protocoles de communication afin d'interfacer le microcontrôleur avec le PC et la carte mémoire. Les protocoles de communication utilisés lors de la réalisation de projet sont : l'UART (Universal Asynchronous Receiver/Transmitter) et le SPI (Serial Communication Interface). La communication entre microcontrôleur-PC se fait par le protocole UART et la communication entre carte mémoire -PC se fait par le protocole SPI.

II.2.1. Le protocole de communication UART :

II.2.1.1. Présentation de Protocole UART : [12]

L'UART (Universal Asynchronous Receiver/Transmitter) est un protocole de communication série intégré dans la pluparts des microcontrôleurs modernes, il permet la communication entre deux microprocesseurs. Chaque UART dispose d'une entrée R_X et d'une sortie T_X , le transmetteur envoie une trame de données bit par bit au récepteur, cette trame est composée :

- D'un bit de début toujours à 0 (Low level).
- D'un octet de données (08 bits).
- D'un bit de parité (paire ou impaire).
- D'un ou deux bits d'arrêt «stop» toujours à 1 (High level).

La transmission des données est faite d'une façon asynchrone, cela signifie qu'il n'y a pas de signal d'horloge (CLK) entre l'émetteur et le récepteur, d'où Le récepteur se base sur la durée de ses bits qui doit être fixé et connue de l'émetteur et du récepteur pour identifier les différents bits d'un octet. La durée d'un bit est définie par le BAUDRATE, nombre de bits par secondes. Le BAUDRATE peut avoir les valeurs suivantes : 300 bps, 1200 bps, 2400 bps, 4800 bps, 9600 bps, 19200 bps, 38400 bps, des vitesses rapides sont parfois aussi possibles.

L'interfaçage du microcontrôleur avec le PC nécessite l'utilisation d'un circuit intermédiaire dit « convertisseur USB to UART » pour connecter le microcontrôleur directement au PC via le port série.

La figure II.13 donne le format de tram de données envoyées par UART et la figure II.14 représente une photo réelle de convertisseur USB to UART.

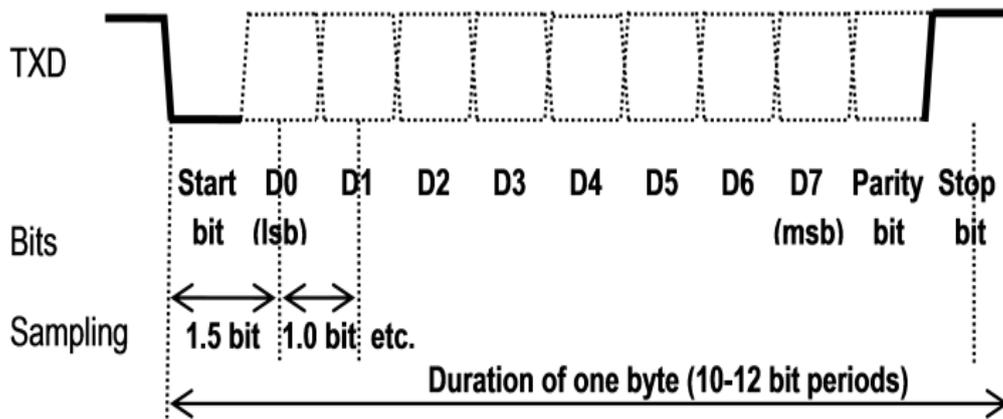


Figure II.13 : Format de tram de données.



Figure II.14 : Photo réelle d'un convertisseur USB to UART.

II.2.1.2. Configuration d'UART sur le Microcontrôleur Atmega2560 : [7]

Le microcontrôleur Atmega2560 dispose quatre chaînes de communication UART ; UART0, 1, 2 et 3. La configuration de ces chaînes se fait à travers les registres suivants :

- UDR_n : Input/Output Data Register.
- UBRR_{nH} : USART_x Baud Rate Register High Byte.
- UBRR_{nL} : USART_x Baud Rate Register Low Byte.
- UCSR_{nA} : USART Control and Status Register A.
- UCSR_{nB} : USART Control and Status Register A.
- UCSR_{nC} : USART Control and Status Register A.

NB : "n" désigne le numéro de la chaîne de communication (0, 1, 2, 3).

- **UDRn** : USART input/output Data Register, c'est un registre à 08 bits partagé entre le buffer de transmission (TXB) et le buffer de réception (RXB), le lecteur de ce registre renvoie le contenu de buffer de réception, tandis que, la transmission de données écrits dans ce registre dans le buffer TXB n'est activée que lorsque le bit UDREn de registre UCSRnA est mis à l'état 1, sinon le contenu de registre UDRn sera ignoré par le buffer TXB.
- **UBRRnH** : C'est un registre à 8 bits contient les 04 chiffres significatifs de Baud Rate.
- **UBRRnL** : C'est un registre à 8 bits contient les 08 chiffres moins significatifs de Baud Rate.
- **UCSRnA** : C'est un registre à 8 bits qui donne l'état de transmission et de la réception (fin de transmission, fin de réception, prêt à recevoir, erreur de transmission, erreur de réception...).
- **UCSRnB** : C'est un registre à 8 bits qui permet de :
 - activer ou désactiver les interruptions proviennent de : la fin de transmission, la fin de réception.
 - Activer la transmission et la réception.
- **UCSRnC** : C'est un registre à 8 bits qui permet de :
 - choisir le mode de communication : synchrone, Asynchrone, Master SPI.
 - Le mode de parité.
 - Nombre des bits d'arrêt (1 ou 2).
 - Nombre de bits de données.

II.2.2. Le protocole de communication SPI :

II.2.2.1. Présentation de Protocole SPI : [13]

Le SPI (Serial Peripheral Interface) est un protocole de communication série synchronisé utilisé par plusieurs dispositifs comme les modules de carte mémoire, les lecteurs RFID pour se communiquer avec les microcontrôleurs. La relation entre les dispositifs communiquant par SPI est généralement de type Master-Slave où le Master (Microcontrôleur) fait le contrôle et le Slave (carte mémoire, capteur,..) reçoit et applique les instructions de Master. La plus simple configuration est : un Master-un Slave, cependant, Le Master peut contrôler plusieurs Slaves à la fois (Multiple Slave). La communication SPI nécessite quatre lignes : MOSI, MISO, SCLK et SS/CS.

- **SCLK** : c'est le signal d'horloge généré par le Master qui permet la synchronisation des données envoyées. L'envoi de donnée est fait bit par bit dans chaque cycle d'horloge (Clock), d'où la vitesse de communication dépend de la fréquence de signal d'horloge généré.
- **SS/CS** : Slave Select/Chip Select, cette ligne est utilisée par le Master pour choisir le slave avec lequel se communiquer, le Master peut avoir un ou plusieurs pins SS/CS ce qui permet la communication avec plusieurs slave soit en parallèle soit en enchaînement.
- **MOSI/MISO** : Le Master envoie les données en série bit par bit à travers la ligne MOSI et le slave reçoit ces données à travers le pin MOSI, le slave aussi peut envoyer des données au Master à travers la ligne MISO. Généralement l'envoi de donnée de master au slave commence par les bits les plus significatifs.

La figure II.15 représente un Schéma de brochage Master-Slave et Master-Multiple Slave.

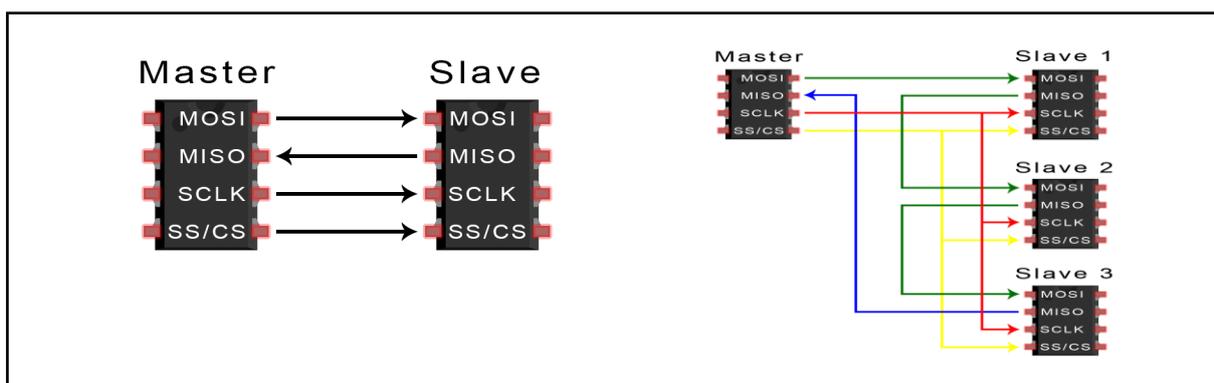


Figure II.15 : Schéma de brochage Master-Slave et Master-Multiple Slave.

II.2.2.2. Configuration de SPI sur le microcontrôleur Atmega2560 : [7]

Le Atmega2560 dispose un seul “Master/Slave SPI interface”, la configuration de l’interface SPI se fait à travers les registres suivantes :

- SPCR : SPI Control Register.
- SPSR : SPI Status Register.
- SPDR : SPI Data Register.
- **SPCR** : C’est un registre à 8 bits qui permet de :
 - Activer l’interruption prévenante de l’SPI.
 - Activer/Désactiver la communication SPI.
 - Sélectionner le mode de transmission (Plus/Moins Significatif bits).
 - Sélectionner le Mode Master ou Slave.
 - Choisir la polarité de signal d’Horloge et le mode d’échantillonnage.
 - Choisir le la fréquence de signal d’Horloge.
- **SPSR** : C’est un registre à 8 bits qui contient les Flags des interruptions liées à la communication SPI.
- **SPDR** : C’est un registre à 8 bits qui supporte la lecture et l’écriture (Read/Write) il contient le data à transmettre ou le data reçu.

CONCLUSION :

Dans ce chapitre nous avons abordé la partie Hardware (matérielle) de notre projet en décrivant les différentes parties qui composent la carte électronique de commande et leurs caractéristiques, aussi, nous avons vu les techniques d’interfaçage utilisées pour relier le microcontrôleur avec le PC et la carte mémoire et leur configuration sur le microcontrôleur Atmega2560.

III

CHAPITRE

Description du Software

INTRODUCTION :

Dans le chapitre précédent nous avons fait la description de la partie hardware de notre projet où nous avons donnée tous les détails sur la carte électronique de commande et les techniques de communication utilisées.

Dans ce chapitre nous allons présenter l'ensemble des softwares utilisés lors de la réalisation de notre projet ainsi les étapes de leur utilisation.

III.1. Présentation du Software Utilisé :

La partie software est indispensable pour la réalisation de notre projet vue que le fonctionnement de la partie mécanique et la partie hardware de projet est lié directement au software utilisé pour modéliser les différentes pièces mécaniques ainsi, le développement de firmware et le circuit de commande. Les différents softwares utilisés sont les suivants :

- L'environnement de développement Arduino IDE.
- Le Visual studio C Sharp.
- ISIS Proteus.
- EAGLE Cadsoft.
- Solidworks.
- Logiciel du tranchage.

III.2. L'environnement de développement Arduino IDE : [14]

Un environnement de développement se présente comme un ensemble d'outils qui comporte des bibliothèques de code, un éditeur de texte, un compilateur et un débogueur. Il est souvent dédié à un seul langage de programmation.

Le développement de Firmware de notre projet sur le microcontrôleur Atmega2560 nécessite l'utilisation d'un environnement de développement dédié à ce microcontrôleur de la famille Atmel. Parmi ces environnements il y'a le Arduino IDE, AVR Studio et Atmel Studio. Le développement de notre Firmware est fait sous l'environnement Arduino IDE qui utilise un compilateur C/C++ avec un éditeur de texte et une interface Homme-machine très simple.

Le chargement de programme dans le microcontrôleur se fait directement par port USB d'une façon simple et facile.

III.2.1. Structure d'un projet Arduino :

A l'ouverture d'un nouveau projet sur Arduino IDE, on trouvera deux fonctions déjà prédéfinies dans l'éditeur du texte qui sont les fonctions « void setup () » et « void loop() », l'appellation de ces deux fonctions est fixe et non modifiable.

- La fonction « void setup() » doit contenir la configuration des ports entrées/sorties, la configuration des registres de timer, PWM, ADC,Etc..., ainsi, le programme qui s'exécute une seule fois.
- La fonction « void loop() » contient le programme principale qui se répète en boucle.

Les fonctions personnelles sont écrites hors de ces deux fonctions « void setup() » et « void loop() », cependant, leur appel est faite dans ces deux fonctions directement ou à travers une autre fonction.

La figure III.1 donne la structure minimale d'un code C sur Arduino IDE.

```
void setup() {  
    // put your setup code here, to run once:  
  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
  
}
```

Figure III.1 : Structure minimale d'un code C sur Arduino IDE.

La figure III.2 montre un exemple de programme sous l'environnement de développement Arduino.

```
// the setup function runs once  
void setup() {  
    // initialize digital pin 13 as an output.  
    pinMode(13, OUTPUT);  
}  
  
// the loop function runs over and over again forever  
void loop() {  
    digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)  
    delay(1000);           // wait for a second  
    digitalWrite(13, LOW); // turn the LED off by making the voltage LOW  
    delay(1000);           // wait for a second  
}
```

Figure III.2 : Exemple de programme sur Arduino IDE.

III.2.2. Interface du logiciel :

L'interface graphique de logiciel Arduino IDE est simple et peu encombrée et contient toutes les options nécessaires. La figure III.3 représente l'interface graphique du logiciel Arduino IDE.

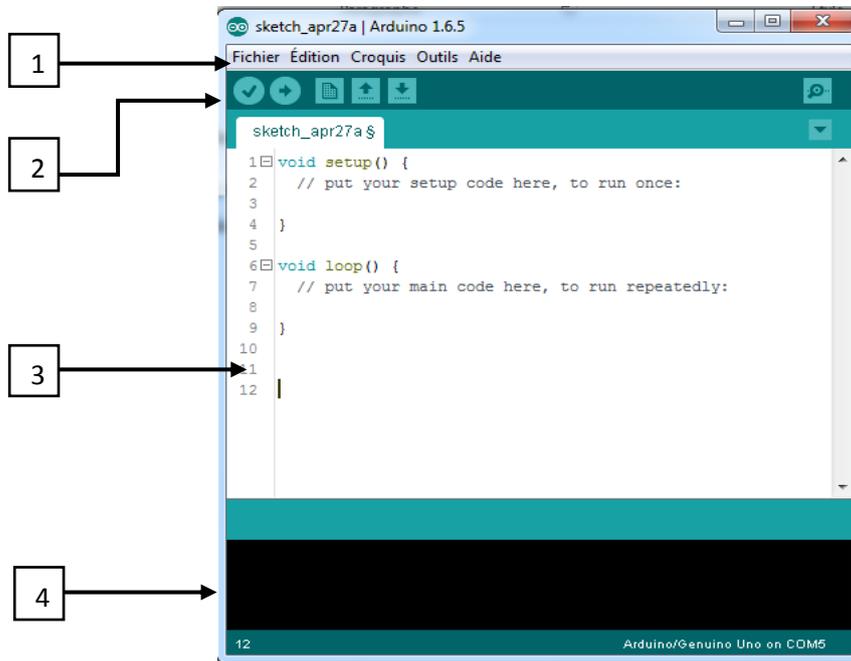


Figure III.3 : Interface du logiciel Arduino.

- (1) : la barre d'options.
- (2) : barre de travaille.
- (3) : le bloc de programme.
- (4) : le bloc d'erreurs et de résultats de débogage.

❖ Les Boutons :

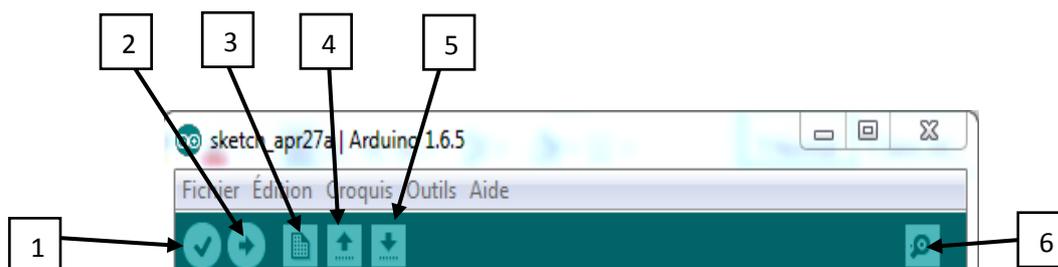


Figure III.4 : Utilité des boutons d'interface Arduino IDE.

- **Bouton 1** : c'est le bouton de vérification de programme, il cherche les erreurs de programmation.
- **Bouton 2** : le bouton de compilation et de téléversement de programme vers la carte.
- **Bouton 3** : créer un nouveau fichier.
- **Bouton 4** : ouvrir un programme qui existe déjà.
- **Bouton 5** : le bouton d'enregistrement.
- **Bouton 6** : ouvrir le moniteur série.

III.2.3.Chargement du programme dans le microcontrôleur Atmega2560 :

Afin de charger le programme dans le microcontrôleur on suit les étapes suivantes :

- Préparer le programme sur le logiciel Arduino IDE.
- Connecter la carte Arduino Mega avec le PC.
- Si la carte Arduino Mega n'a pas été détectée, il faut vérifier si son driver est installé ou pas.
- Sélectionner sur l'icône « Outils » :
 - Type de la carte : Arduino Mega 2560.
 - Port : le port où la carte Arduino est connectée.
- Cliquer sur le bouton « Téléverser ».
- Attendre jusqu'à ce que le chargement soit terminé.
- La carte est prête à être utilisée.

III.3. Visual Studio C# : [15]

III.3.1. Présentation de l'environnement de développement Visual Studio :

Le Visual Studio est un environnement de développement de Microsoft. Il est utilisé pour développer des programmes sur Windows ainsi des sites et des applications Web et Mobile, le Visual Studio utilise les plateformes de développement de Microsoft comme Windows API, Windows Forms et Windows Store.

Le Visual Studio inclue un éditeur de code intelligent qui supporte l'option de complétion de code, un débogueur et d'autres outils comme le profilage de code, un designer de formes pour la conception des application GUI, applications Web, les classes et d'autres

formes. De plus, le Visual Studio supporte plusieurs langages de programmation qui incluent le : C, C++, C#, VB.net, F#. Ainsi, Il supporte aussi le XML/XSLT et HTML/XHTML.

III.3.2. Présentation de langage C# :

C# est un nouveau langage de programmation orienté objet de type sécurisé. Il a été conçu par Microsoft pour permettre la création d'une large gamme d'applications qui s'exécutent sur .NET Framework. Le C# combine les avantages des logiciels permettant le développement rapide des applications et la souplesse de langage C++. Le code C# est compilé en tant que un code managé, c'est à-dire qu'il bénéficie des services de Common Language Runtime (CLR). Ces services incluent l'interopérabilité entre les langages, un Garbage collection et une sécurité améliorée.

III.3.3. Commencer avec le Visual Studio C# :

Dans ce projet nous avons utilisé le Visual Studio C# 2013 pour développer l'interface de commande Homme-Machine qui sert à assurer la communication entre le PC et la carte électronique de commande de l'imprimante 3D.

Pour démarrer avec le Visual Studio C#, on suit les étapes suivants :

- On ouvre la fenêtre principale de Visual Studio et on clique sur New Project.
- On sélectionne le Visual C #.
- On sélectionne le type de projet : Windows Forms Application.
- On donne le nom de projet et l'emplacement sur PC.
- On clique sur OK.

Après la création de nouveau projet une autre figure apparaît contenant plusieurs fenêtres comme le Toolbox, fenêtre de design et la fenêtre de propriétés. Pour ajouter un objet (bouton, TextBox, ...), on le traîne directement de Toolbox vers la fenêtre de design, ensuite, pour accéder au code on fait double clics sur l'objet et on commence la programmation. Les figures suivantes montrent les différents fenêtres de Visual Studio et les étapes de création de projet C#.

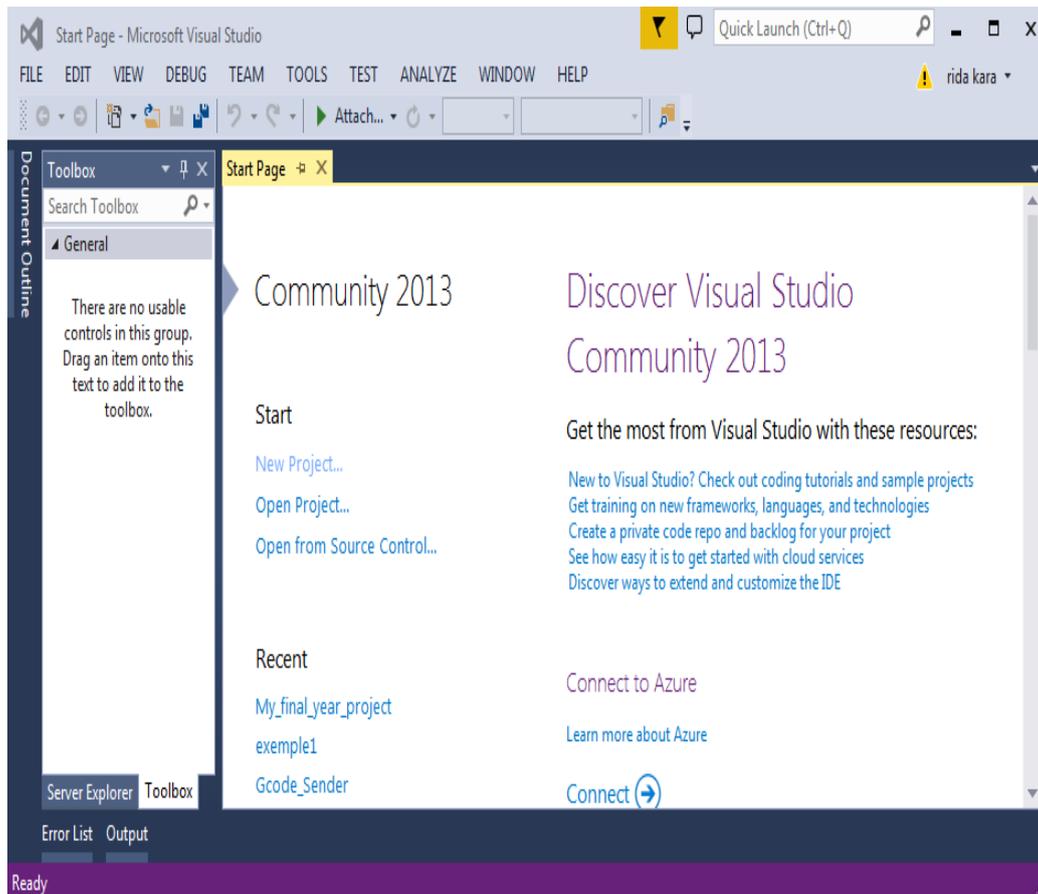


Figure III.5 : La fenêtre principale de Visual Studio.

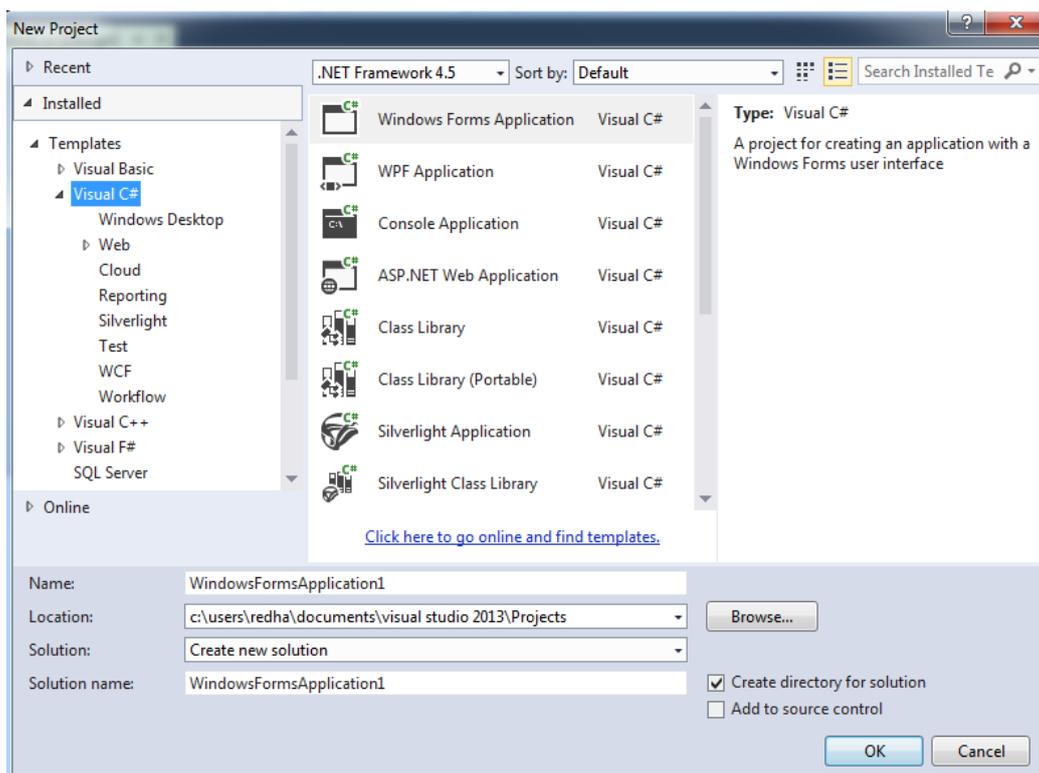


Figure III.6 : Fenêtre de choix de langage et de type de projet.

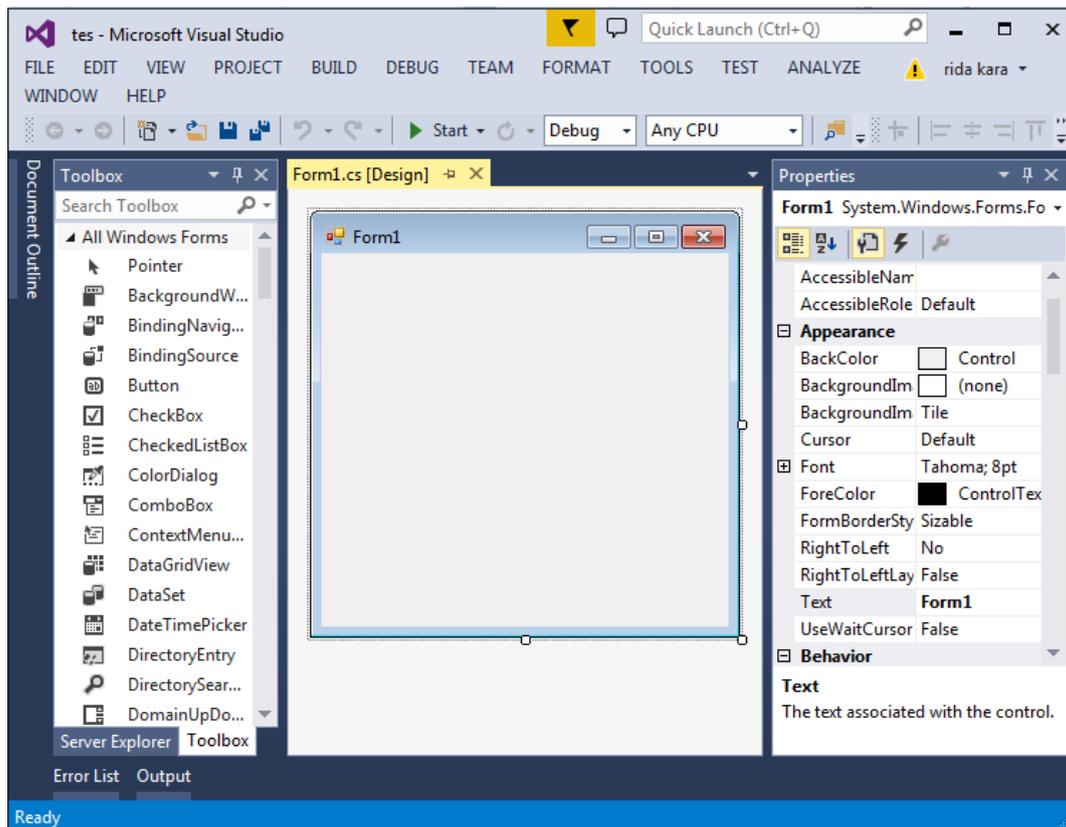


Figure III.7 : Fenêtre d'espace de travail.

Le Visual C# est constitué de trois composantes principales :

❖ **La fenêtre de ToolBox :**

Cette fenêtre contient tous les outils nécessaires à la création d'une interface de GUI (Graphical User Interface) comme les boutons, textbox, serialport, timer et plein d'autres objets.

❖ **La fenêtre de propriétés :**

Cette fenêtre donne les propriétés de l'objet sélectionné sur la fenêtre de design comme la couleur, le font, le background, la taille et l'emplacement. La modification de ces propriétés est simple, il suffit de les changer directement sur cette fenêtre sans avoir taper de code.

❖ **La fenêtre de design :**

C'est la fenêtre où nous faisons la conception de notre interface graphique en faisant trainer les objets directement de toolbox vers cette fenêtre.

III.4. ISIS Proteus : [6] [16]

III.4.1. Présentation de logiciel ISIS Proteus :

Le logiciel Proteus est composé de trois modules :

- Editeur de Schéma ISIS.
- Le simulateur LISA.
- L'outil de conception de circuit imprimé Ares.

ISIS : produit d'une part une liste d'équipements qui peut être utilisées par le simulateur LISA et l'outil de conception de circuit imprimé ARES, d'autre part, une liste de matériel et des rapports de contrôle des règles électriques.

LISA : c'est un ensemble de modules de simulation lié à ISIS. Le noyau de simulation PROSPICE est basé sur la version 3F5 du moteur SPICE publié par l'université de Berkeley.

ARES : c'est un module de conception de circuit imprimé. Il permet le placement des composantes et le routage des liaisons en mode automatique, manuel ou semi-automatique.

III.4.2. Commencer avec ISIS Proteuse :

Pour créer un nouveau projet, on clique sur « file-new design » puis on clique sur « Save design » et on donne le nom et l'emplacement de fichier sur PC. Une fois le nouveau projet est créé, on peut commencer la conception de schéma électrique de projet en ajoutant les composantes directement de la bibliothèque de composantes.

Afin de tester le montage que nous avons fait, il suffit juste de démarrer la simulation en cliquant sur « Run the simulation », Proteus dispose aussi d'une boîte à outils pour la mesure (Ampèremètre, Voltmètre, Oscilloscope, ...).

Pour passer de schéma électrique au design PCB on clique sur le bouton « ARES » dans la barre d'options, une autre fenêtre apparaît contenant les composantes que nous avons utilisées dans le schéma avec toutes les connexions que nous avons fait. Il suffit juste de commencer le routage et la structuration de la carte PCB de la façon que nous souhaitons. Les figures III.8 et III.9 montrent l'interface principale d'ISIS Proteus et le schéma de montage.

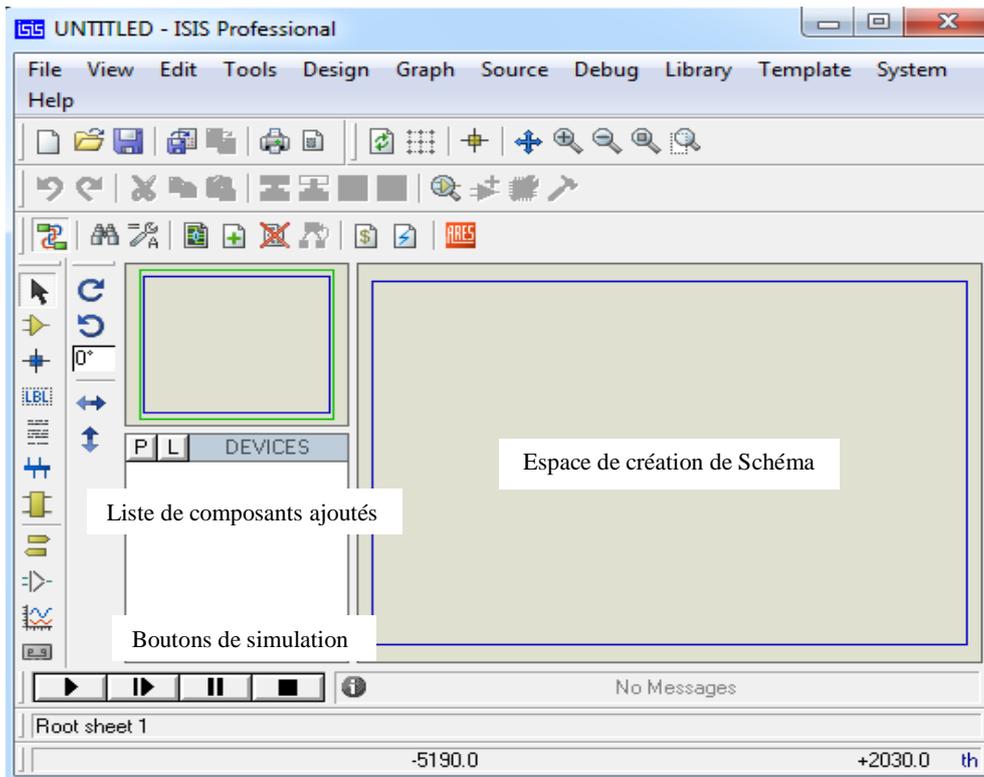


Figure III.8 : Interface principale de Logiciel ISIS Proteus.

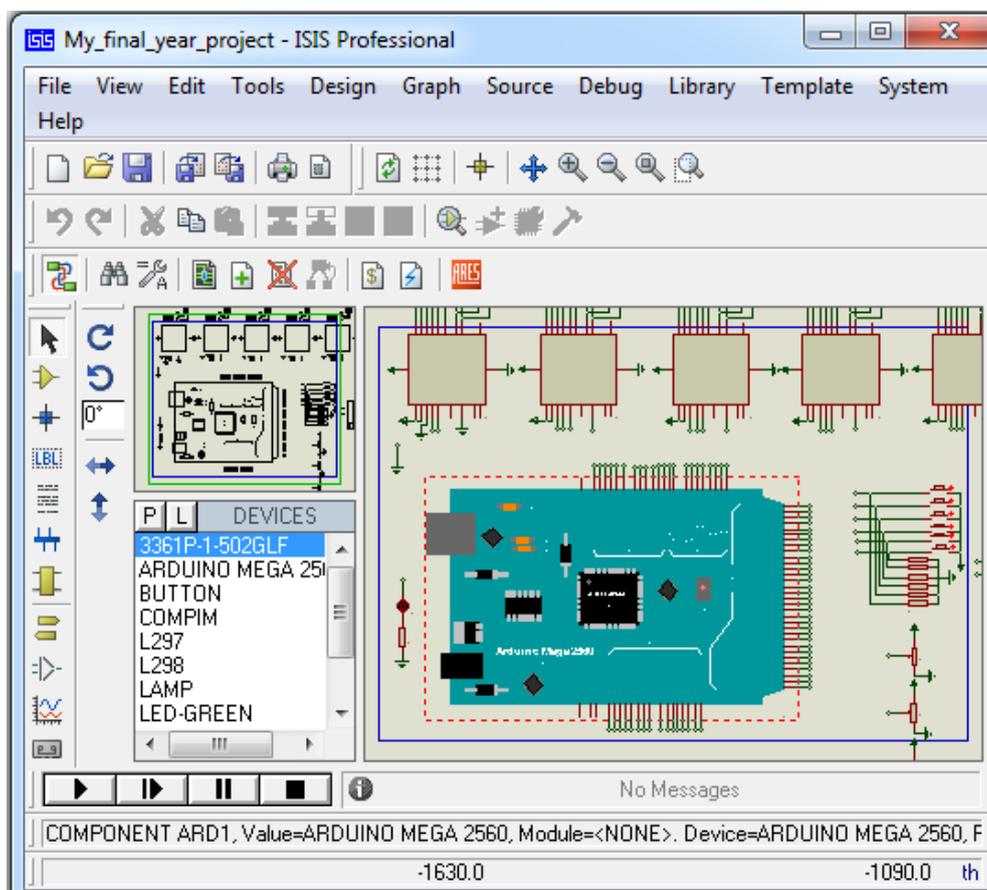


Figure III.9 : Schéma de montage sur ISIS Proteus.

III.5. Logiciel EAGLE : [17]

III.5.1. Présentation du logiciel EAGLE :

EAGLE est un logiciel de routage professionnel très utilisé dans la conception et le design des cartes électroniques imprimées PCB (Printed Circuit Board), il est simple à manipulé et il dispose une large bibliothèque de composants utiles pour la conception des cartes électroniques.

III.5.2. Commencer avec EAGLE :

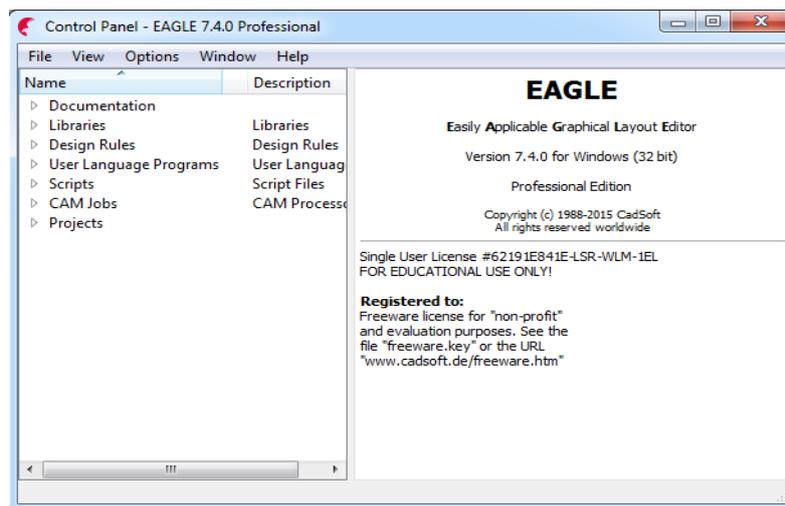


Figure III.10 : Interface principale du logiciel EAGLE.

Pour créer un nouveau schéma on clique sur File – New Schematic et la fenêtre ci-dessous apparaîtra. Cette fenêtre est l'espace de travail, elle contient tous les outils nécessaires pour la création de notre schéma de montage.

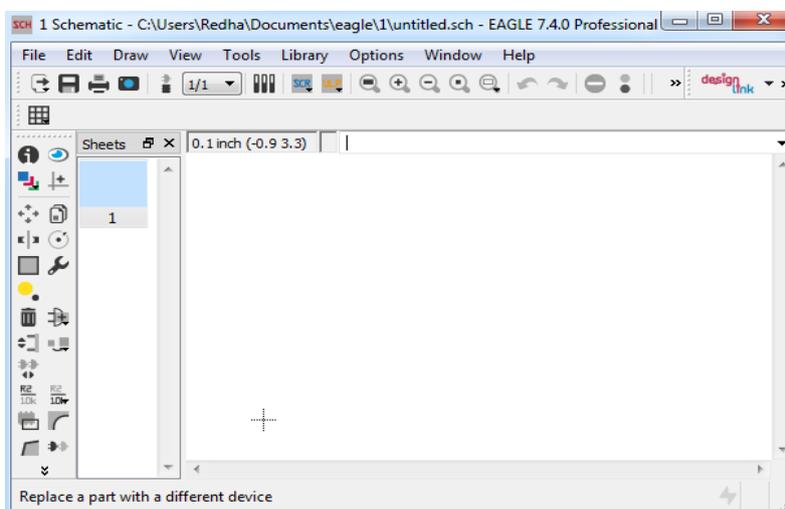


Figure III.11 : Interface de conception de Schéma de montage.

Pour ajouter un composant à l'espace de travail, on clique sur le bouton « Add » à gauche et on cherche le composant désiré sur la fenêtre qui apparaîtra. Cette fenêtre contient la bibliothèque des composants disponibles sur EAGLE.

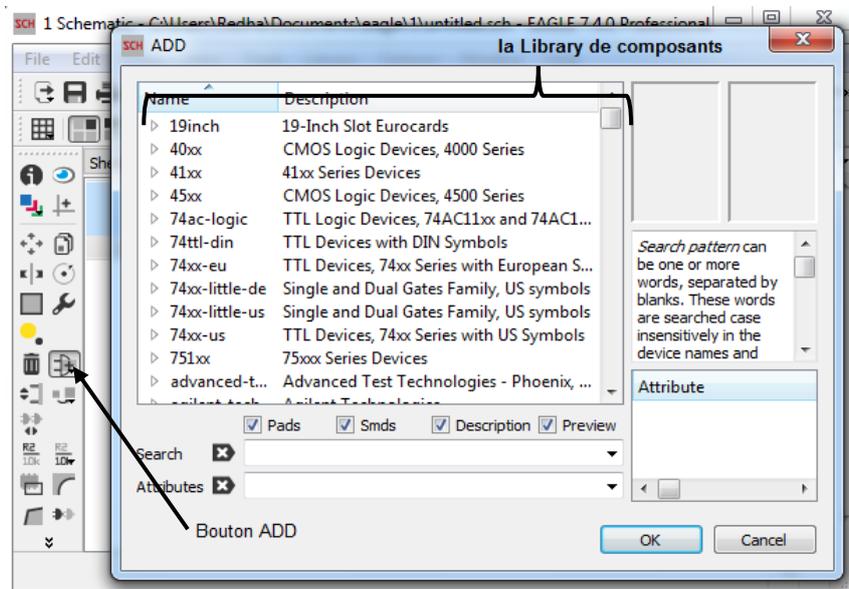


Figure III.12 : Bibliothèque de composantes sur EAGLE.

Pour aller à l'espace de conception de circuit imprimé (Routage) on clique sur « Generate/switch to Board » en haut et la fenêtre suivante apparaîtra. Cette fenêtre contient tous les outils nécessaires pour faire le routage entre les composants que nous avons ajouté et câblé dans le schéma de montage. Toute modification sur le schéma apparaîtra aussi sur cette interface.

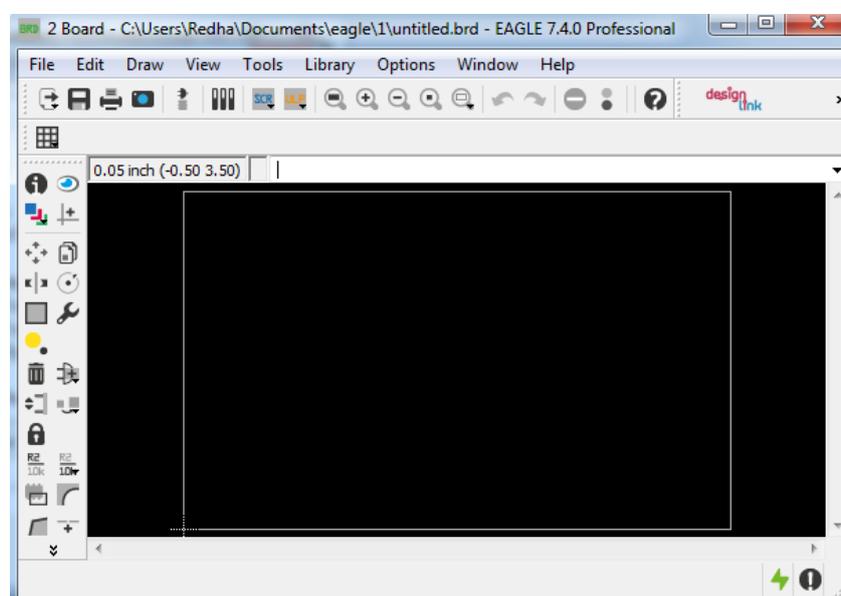


Figure III.13 : Fenêtre de conception de circuit imprimé.

III.5.3. Résultats du travail sur EAGLE :

Après avoir ajouté les composants nécessaires ainsi créer le schéma de montage, nous avons commencé le design de circuit imprimé PCB (Printed Circuit Board) de notre carte électronique sur l'interface de conception. Nous avons fait le maximum pour que la carte soit d'une seule face pour faciliter la tâche lors de sa conception. Les liaisons en bleu sont de côté soudure (Bas) de la carte, les liaisons en rouge sont de côté composant (Top). Ces dernières sont des simples fils électriques que nous avons utilisés pour relier le plan de masse. La largeur des pistes varie selon l'intensité du courant qui va passer, un courant important nécessite que la piste soit largement suffisante sinon il aura des problèmes dans la carte électronique. La figure III.14 représente le design de circuit imprimé sur le logiciel EAGLE.

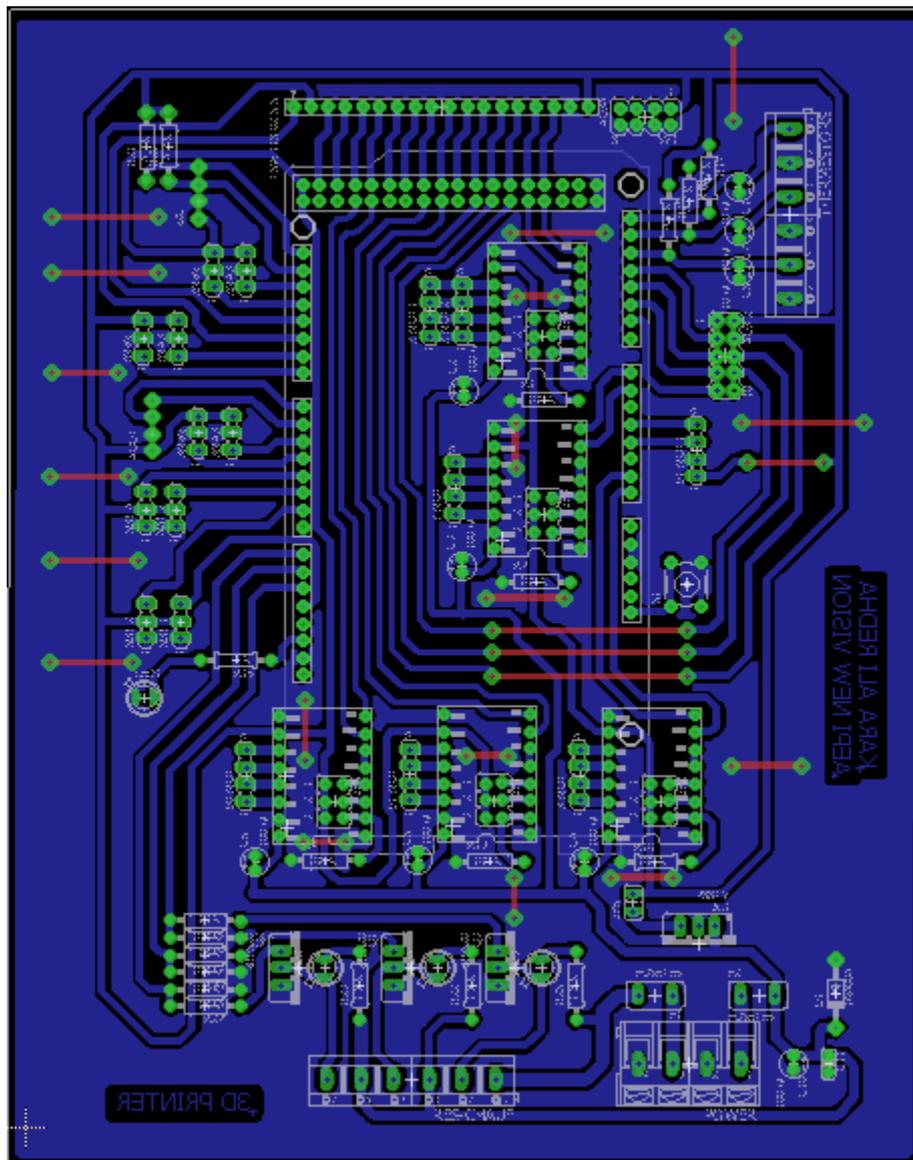


Figure III.14 : Le design de circuit imprimé sur le logiciel EAGLE.

III.6. SolidWorks : [18]

III.6.1. Présentation du SolidWorks :

SolidWorks est un logiciel de CAD (computer Aided Design) très puissant et facile à manipuler, dédié au design et la modélisation 3D. Il est très utilisé dans l'industrie pour la fabrication de différentes pièces mécaniques, hydrauliques, électriques ainsi l'assemblage des pièces et la conception des machines et des appareils de A à Z.

De plus, SolidWorks offre une variété de choix de matériaux à utiliser ainsi, il donne la possibilité de faire des simulations sur l'objet créé pour tester les effets de différents contraintes appliqués sur l'objet avant de le construire ou tout simplement pour le relier à un autre logiciel comme Matlab pour tester sa fonctionnalité avec le code écrit.

Dans ce projet nous avons utilisé le SolidWorks pour modéliser et modifier les différentes pièces de notre imprimante 3D, ainsi, pour créer des pièces en 3D pour les imprimer par la suite à l'aide de cette imprimante.

III.6.2. Commencer avec le SolidWorks :

La figure III.15 représente la fenêtre principale du SolidWorks.



Figure III.15 : Fenêtre principale de SolidWorks.

Pour créer un nouveau projet on clique sur file - new, une nouvelle fenêtre de dialogue apparaîtra, on sélectionne le type de projet entre : création d'un dessin 2D, création d'un objet 3D ou assemblage des objets 3D. Après la sélection nous serons pris vers l'espace de travail.

L'espace de travail de SolidWorks est très riche, il contient les outils de dessin 2D et 3D, outils de simulation, outil pour la sélection de la vue et le plan, outils pour la mesure des distances et les angles et plein d'autres outils nécessaires pour la modélisation 3D. La figure III.16 représente l'espace de travail de SolidWorks et la figure III.17 représente l'assemblage de notre imprimante 3D sur SolidWorks.

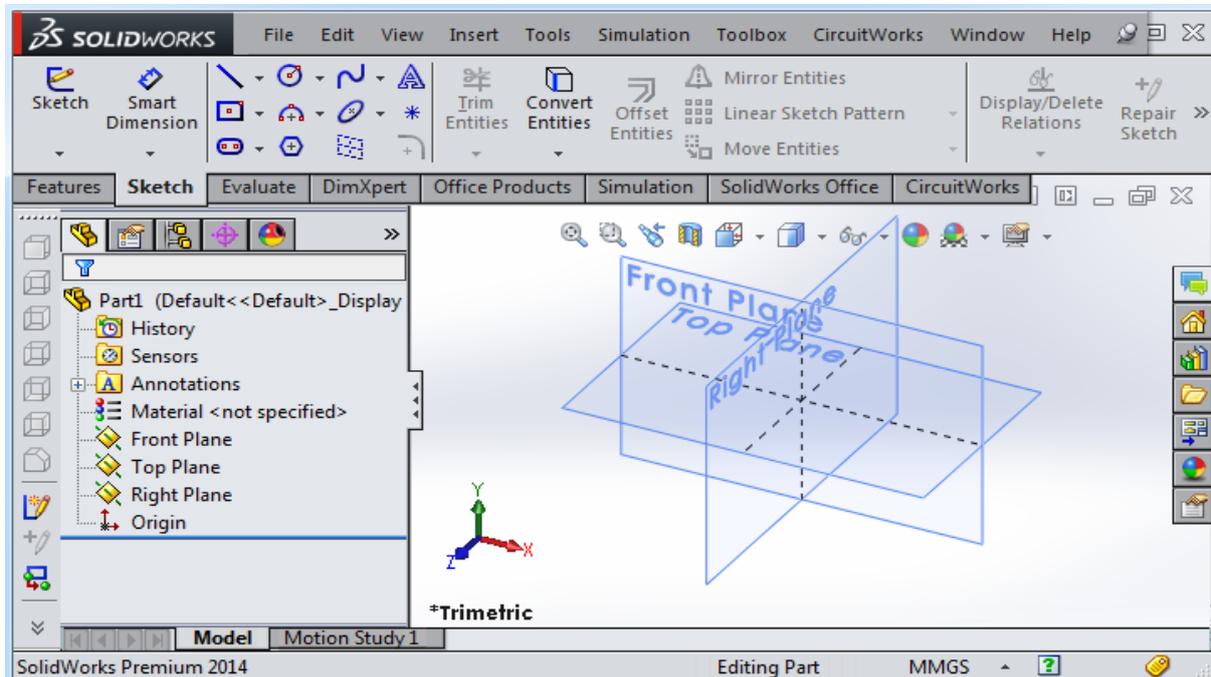


Figure III.16 : Espace de travail sur SolidWorks.

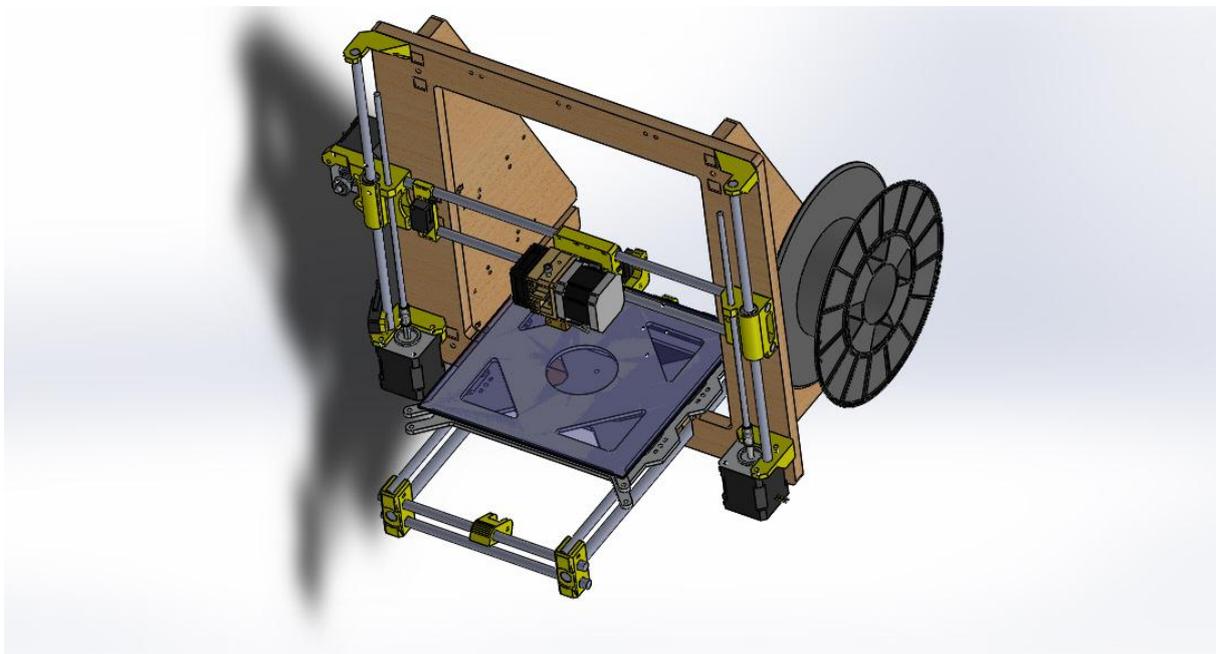


Figure III.17 : Assemblage de l'imprimante 3D sur SolidWorks.

III.7. Logiciel du tranchage :

Après la modélisation de l'objet avec un logiciel 3D, et avant l'impression 3D, vient le tranchage du fichier 3D. L'étape de tranchage passe alors par un logiciel, appelé Slicer, qui découpe en tranches le fichier numérique et fournit à la machine les informations nécessaires à l'imprimante pour réaliser l'impression. La pièce est définie par un ensemble de données telles que l'épaisseur des couches, la quantité de matières, la vitesse d'impression... Généralement les informations sont écrites dans un langage appelé G-Code, déjà utilisé depuis les années 60 cela permet encore aujourd'hui de piloter les machines-outils à commande numérique.

III.7.1. Présentation de G-Code : [19]

Le G code est le nom le plus souvent utilisé pour décrire le langage de programmation dédié à la commande numérique, c'est avec ce langage de programmation qu'on fait la commande des machines CNC (Découpeuse Laser, Fraiseuse, Imprimantes 3D, ...). En effet, ce code est une suite de caractères Alphanumériques contenant des commandes et des instructions pour la machine comme : la trajectoire à suivre, la vitesse de déplacement, les outils à utiliser..., et en suivant ces commandes la machine réalisera des tâches bien précises.

III.7.2. Génération de G-code :

Pour générer le G code on n'a pas besoin d'apprendre la programmation avec ce langage car il y'en a plusieurs générateurs de Code G qui sont disponibles et Open source, ces générateurs de Code G prennent un fichier .STL généré par un software de CAD ou de CAO et ils le convertissent en un fichier « .GCODE » pour qu'il devienne traitable par l'imprimante 3D. Parmi ces générateurs de G code on trouve :

- **SkeinForge** : c'est un petit script qui génère le G code. Il est développé avec Python.
- **Reprap Host Software**.
- **SuperSkein** : il est développé avec Processing, simple à utiliser et open source.
- **Repetier** : il est multiplateforme. Il permet la visualisation de la pièce en 3D lors de l'impression.
- **Slic3r** : il est rapide, facile à utiliser, multiplateforme et Open source.

Dans ce projet nous avons utilisé le logiciel Slic3r pour faire le tranchage de fichier STL et générer le G-Code. Le tableau III.1 donne les différentes commandes de G-Code et leur description.

Code	Description
G00	Mouvement linéaire rapide de la machine sans impression.
G01	Mouvement linéaire avec impression.
G02	Mouvement circulaire sens horaire ($X_C = X+I$, $Y_C = Y+J$).
G03	Mouvement circulaire sens anti horaire ($X_C = X+I$, $Y_C = Y+J$).
G04	P2000 : attendre 2000 millisecondes. S2 : attendre 02 seconds.
G20	Les unités sont en pouce.
G21	Les unités sont en millimètre.
G28	G28 : Home All axis. G28 X, Y, Z : Home X axis, Y axis, Z axis.
G90	Mis en mode de positionnement Absolu. Toutes les coordonnées sont relatives à l'origine de la machine.
G91	Mis en mode de positionnement relative. Toutes les coordonnées sont relatives à la dernière position.
M17	Activer/Alimenter tous les moteurs pas à pas.
M82	Mis l'extrudeuse en mode absolue.
M83	Mis l'extrudeuse en mode relative.
M105	Lire la Température actuelle.
M109	définie la température de l'extrudeuse et attendre jusque 'elle soit atteinte.
M112	Arrête d'urgence.
M350	Activer le Microstepping mode.

Tableau III.1 : Les commandes de G-Code et leur description.

III.7.3. Présentation de Slic3r : [20]

Le Slic3r est un logiciel de tranchage qui fait la conversion de model 3D de l'objet à des instructions pour l'imprimante. Ce logiciel permet de réaliser les taches suivantes :

- Découper le modèle sur des couches horizontales.
- Générer le parcours à suivre.
- Calculer la quantité de filament à extrudé.

La figure III.18 représente l'interface graphique principale de Slic3r.

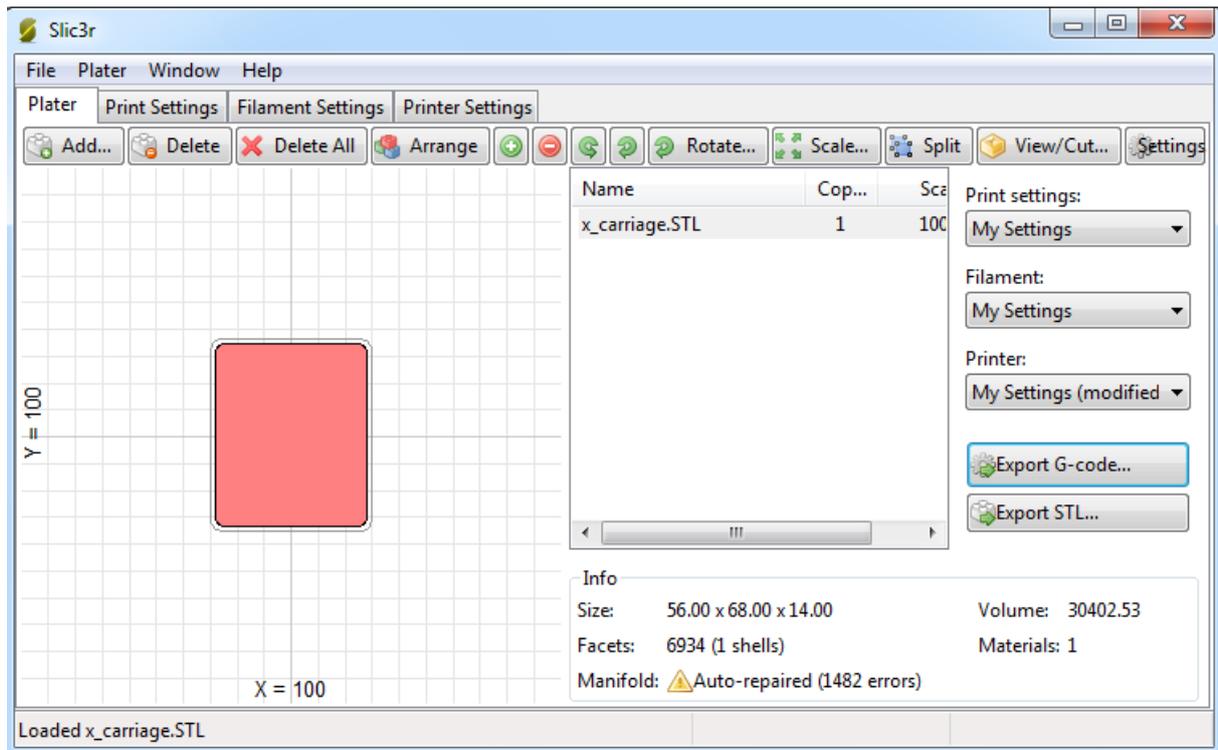


Figure III.18 : Interface graphique principale de Slic3r.

III.7.4. Commencer avec Slic3r :

Pour ajouter un fichier STL on clique sur « Add » qui se trouve sur la fenêtre Plater, ensuite, on sélectionne le fichier désiré et on clique sur « ouvrir ».

Le Slic3r dispose de plusieurs fenêtres de travaux qui permettent de paramétrer l'impression.

- ❖ **Plater** : cette fenêtre permet de positionner l'objet à imprimé sur la plateforme d'impression, elle contient plusieurs boutons d'options qui permettent d'ajouter, supprimer, arranger et séparer, ainsi faire une coupure de vue suivant l'axe Z de l'objet sélectionné.
- ❖ **Print Settings** : cette fenêtre permet de sélectionner les paramètres de l'impression comme la hauteur de la couche, la densité de filament et la vitesse d'impression.
- ❖ **Filament Settings** : elle permet de sélectionner le diamètre de filament, la température de la tête d'impression et la plateforme d'impression pour la 1^{ère} couche et le reste des couches, ainsi, l'activation ou la désactivation des ventilateurs.

- ❖ **Printer settings** : elle permet de sélectionner la taille de la plateforme d'impression, le centre d'impression, le nombre d'extrudeuses disponibles, le diamètre de la buse d'impression et la vitesse de rétraction.

Enfin, après avoir introduire toutes les paramètres nécessaires on clique sur « Export G-Code » sur la fenêtre « Plater », ensuite, on clique sur « Enregistrer ».

CONCLUSION :

Dans ce chapitre nous avons effectué une description relativement détaillée de tous le software utilisé lors de la réalisation de notre projet, où nous avons donné des descriptions de logiciels utilisés, leur utilité et la méthode de leur utilisation, nous avons également présenté quelques exemples simples afin de comprendre leur aspect pratique dans le cadre de notre projet.

IV

CHAPITRE

*Elaboration des Algorithmes et résultats
expérimentaux*

INTRODUCTION :

Dans les chapitres précédents nous avons expliqué toutes les parties de projet, en commençant par son architecture ou nous avons donné toutes les détails nécessaires sur les composants mécaniques et les différents capteurs utilisés. Nous avons également effectué une description détaillée sur la partie Hardware et Software de projet en présentant la carte électronique réalisée et les différents logiciels utilisés.

Dans ce chapitre nous présentons les organigrammes de commande élaborés afin de réaliser l'interface Homme-Machine sur PC et le firmware implémenté dans le microcontrôleur ainsi les résultats expérimentaux que nous avons obtenus.

IV.1. Interface de commande Homme-Machine :

Afin de commander l'imprimante 3D via le PC, nous avons réalisé une interface de commande Homme-Machine en utilisant le Visual Studio C#. Cette interface de commande envoie le G-code de fichier à imprimer à l'imprimante 3D ligne par ligne via une liaison série. De plus elle permet la visualisation de l'état d'imprimante et l'avancement de processus d'impression. Ainsi, cette interface graphique donne la possibilité d'effectuer une commande manuelle sur l'imprimante 3D.

IV.1.1. Présentation de l'interface Homme-Machine :



Figure IV.1 : Interface de commande Homme-Machine.

Cette GUI (Graphic User Interface) comporte essentiellement les parties suivantes :

- ❖ CONNECTION Box :
 - Serial Port : sélectionner le port dont la carte Arduino Mega est connectée.
 - Baud Rate : sélectionner la vitesse de communication (bit par second).
 - Filament : sélectionner le type de filament à utiliser.
 - Open : ouvrir la communication série.
 - Close : fermer la communication série.
 - Refresh : chercher encore une fois sur les ports connectés.
- ❖ OPERATING MODE Box :
 - Automatic Mode : Activer le mode Automatique.
 - Manual Mode : Activer de mode Manuelle.
- ❖ MACHINE STATUS :
 - Machine Position : visualise la position X, Y, Z actuelle de la machine.
 - Working Position : visualise la position X, Y, Z que l'imprimante va atteindre.
 - Emergency Stop : il permet de mettre l'imprimante en état d'arrêt d'urgence.
 - Resume : il permet de reprendre l'impression à nouveau après un arrêt d'urgence.
- ❖ WORK PROGRESS Box :
 - Il permet de visualiser l'état d'avancement d'impression en temps réel.
- ❖ FILE Box :
 - File Name : il visualise le fichier sélectionné.
 - Send : il permet l'envoi de G-code à l'imprimante.
 - Hold : il permet de faire une pause.
- ❖ G-CODE SENDER Box :
 - Automatic Command : effectue une visualisation des lignes de code envoyées à l'imprimante quand cette dernière est en mode automatique.
 - Manual Command : il permet d'obtenir une commande manuelle sur l'imprimante.
- ❖ File : il permet d'ajouter le fichier G-code.
- ❖ View : il permet de visualiser ou désactiver le G-CODE SENDER Box.

L'ensemble de ces actions sont résumés par l'organigramme de la figure IV.2.

IV.1.2. Structure de l'algorithme d'interface de commande Homme-Machine :

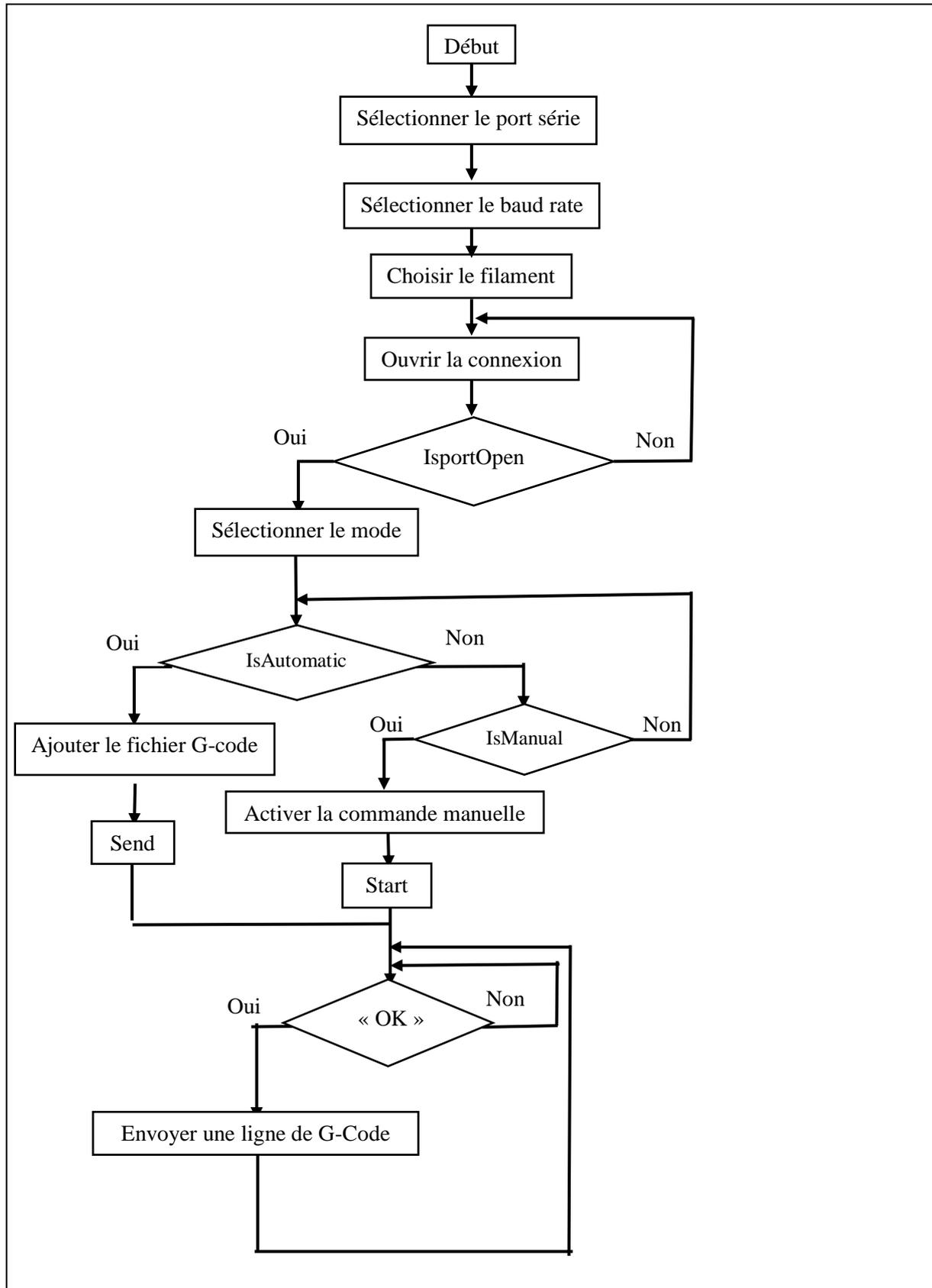


Figure IV.2 : Organigramme global simplifié de l'interface de commande Homme-Machine.

IV.2. Le Firmware :

Afin d'assurer la commande de différents E/S de notre imprimante 3D, nous avons élaboré le « firmware » qui est sensé d'assurer les tâches suivantes :

- Assurer la communication entre le Microcontrôleur et le PC ou la carte mémoire.
- Assurer le contrôle de la température de la tête d'impression.
- Décoder le G-code et récupérer les informations contenant dans chaque ligne de code G.
- Assurer le control de positions des moteurs pas à pas sur les 03 axes X, Y, Z.
- Assurer l'extrusion de la quantité nécessaire de filament plastique.

IV.2.1. Implémentation de la boucle de régulation PID de température

Comme nous avons montré dans le 1^{er} chapitre, le filament plastique que nous utilisons pour l'impression 3D nécessite que la température de la tête chauffante ne soit pas au-dessous ou au-delà de l'intervalle de température donné. Si la température n'est pas dans cet intervalle, on risque de provoquer des problèmes lors de l'impression et l'extrusion de filament, d'où, l'implémentation d'une boucle de régulation PID est nécessaire pour maintenir la température dans la valeur désirée. Les constants K_p , K_i et K_d on les a obtenu par expérience en faisant varier leurs valeurs jusqu'à l'obtention d'un résultat satisfaisant.

La boucle de régulation est implémentée dans une fonction de timer dont la période d'échantillonnage notée « ΔT ».

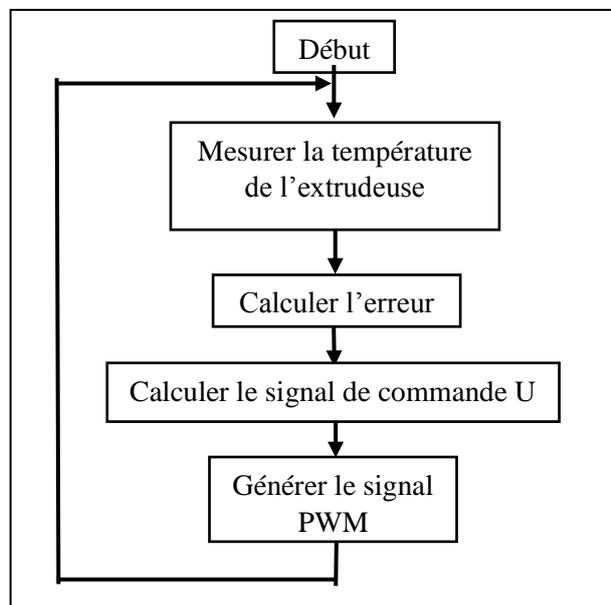


Figure IV.3 : Organigramme simplifié de la boucle de régulation PID de température.

IV.2.2. Implémentation de la boucle de commande principale :

La boucle de commande principale exécute plusieurs tâches, elle assure la communication entre le Microcontrôleur et le PC selon le protocole que nous avons fait, ainsi, elle permet de décoder le G-code et récupérer toutes les données contenant dans chaque ligne de code et ensuite les exécuter en donnant les ordres aux moteurs pour positionner la tête d'impression dans les coordonnées désirés sur les axes X, Y, Z et d'extruder la quantité nécessaire de filament. L'organigramme de la figure IV.4 résume l'ensemble de ces tests.

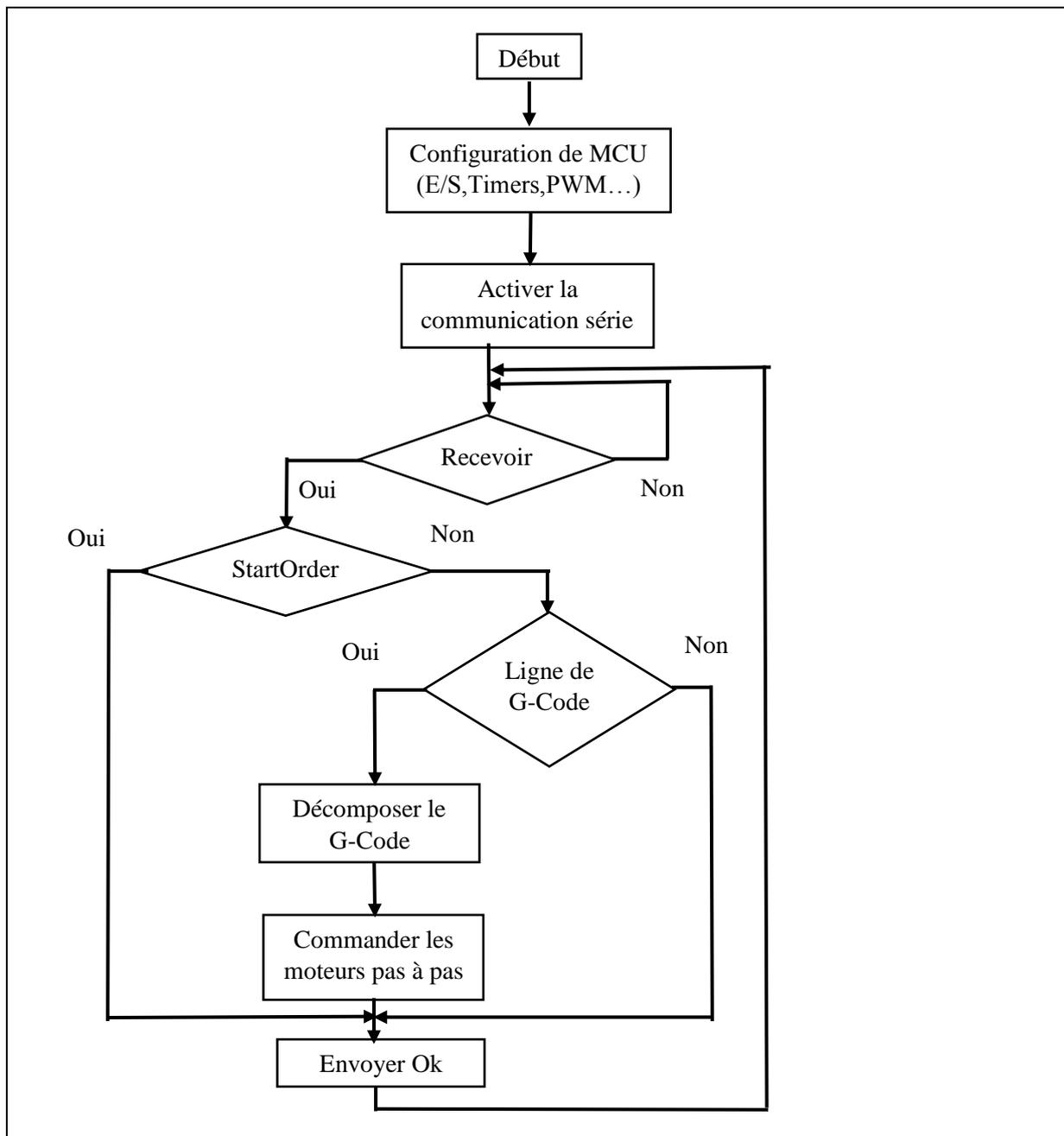


Figure IV.4 : Organigramme simplifié de la boucle de commande principale

IV.3. Testes et résultats :

❖ Teste de communication entre le PC et la carte électronique :



Figure IV.5 : Teste de communication entre le PC et la carte électronique.

❖ Teste de la boucle de régulation PID de température :

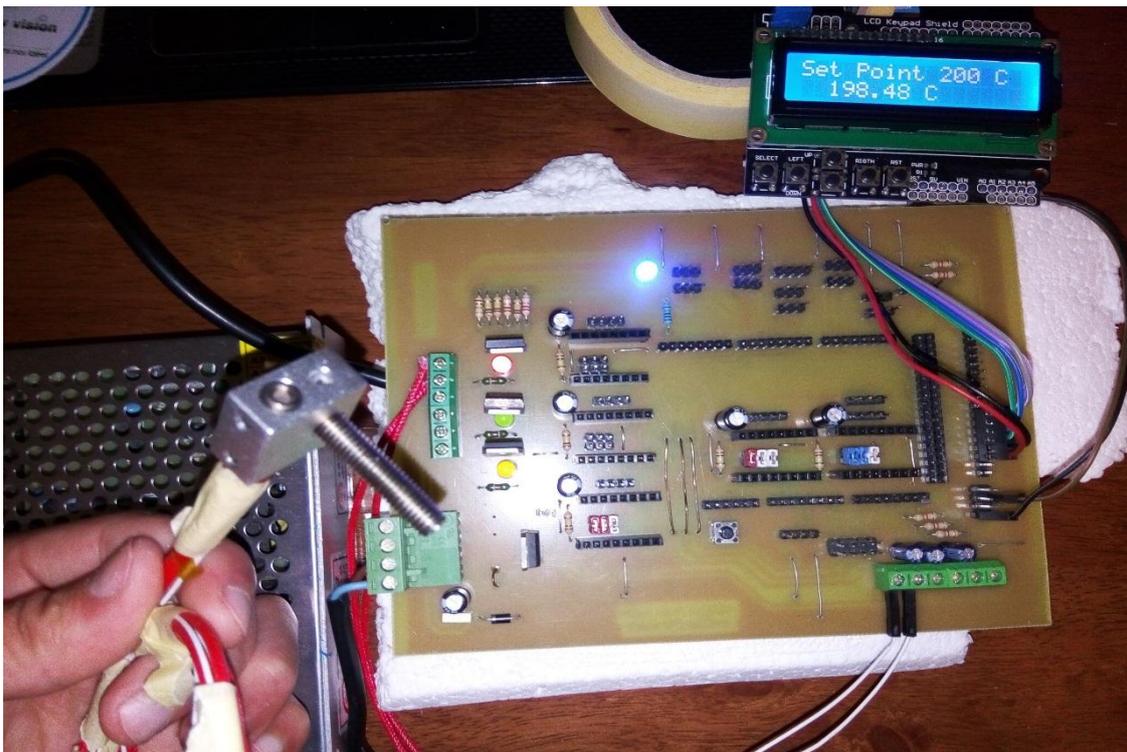


Figure IV.6 : Teste de la boucle de régulation PID.

❖ Assemblage de l'imprimante et le teste final d'impression :

Après avoir faire l'assemblage de notre imprimante 3D, nous avons effectué des tests d'impression de quelques pièces que nous avons modélisé sur le logiciel SolidWorks, les figures IV.7 et IV.8 représentent des photos réelles de l'imprimante 3D réalisée et les tests d'impression effectués.

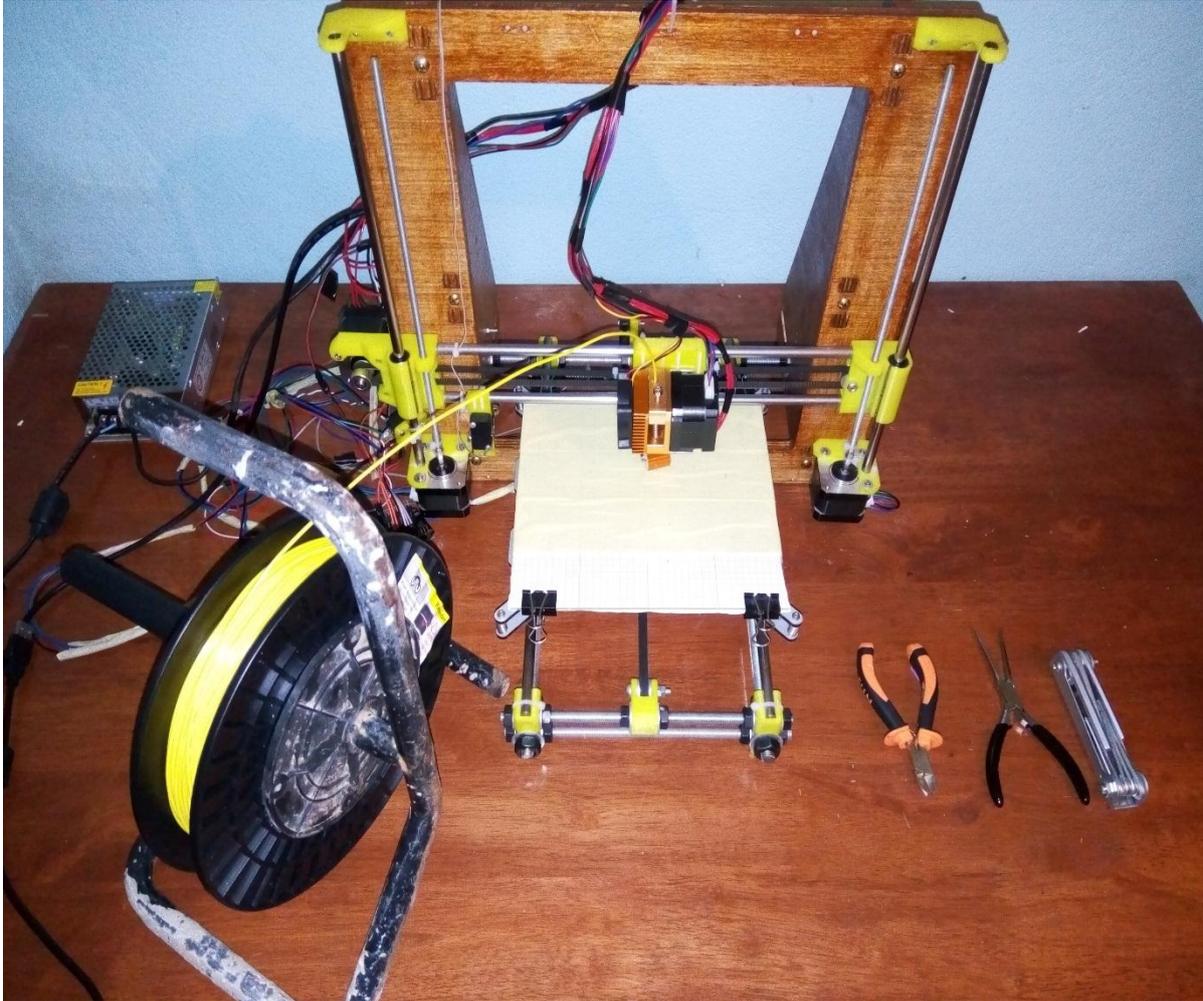


Figure IV.7 : Photo réelle de l'imprimante 3D réalisée.

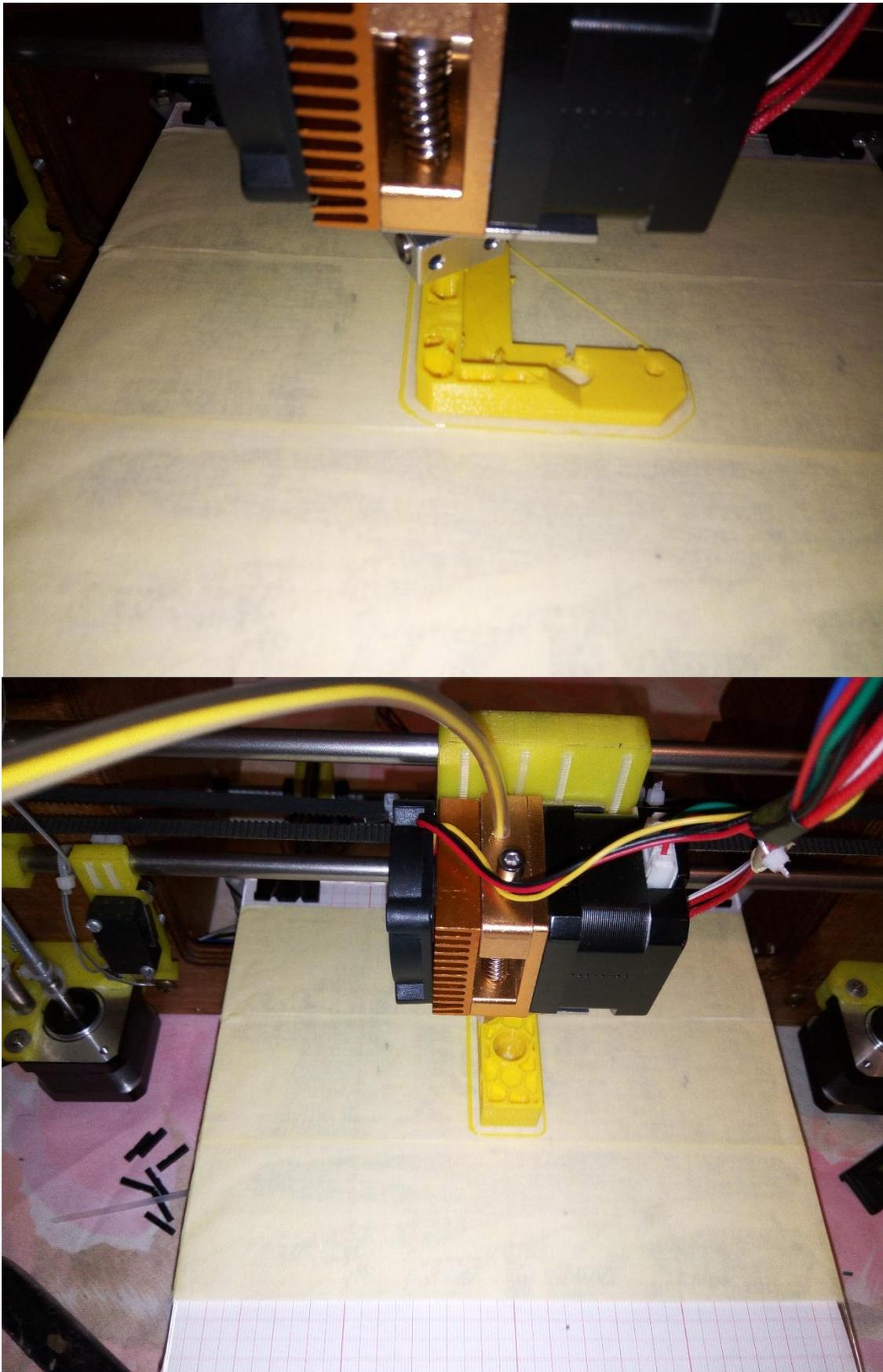


Figure IV.8 : Teste finale d'impression.

CONCLUSION :

Dans ce chapitre nous avons élaboré l'interface de commande Homme-Machine sur PC et nous avons implémenté le firmware de commande dans le microcontrôleur Atmega2560. Nous avons également effectué des tests réels de la boucle de régulation de température PID avec une vérification de la communication entre le PC et la carte électronique. Enfin, après assemblage, câblage et paramétrage de différentes parties de l'imprimante 3D, on l'a mis en marche en faisant un test d'impression d'une pièce déjà modélisée sur le logiciel SolidWorks.

Avec ce chapitre nous arrivons à la fin de notre travail d'étude, de conception et de réalisation de notre imprimante 3D. L'ensemble des essais que nous avons effectué confirment la validité de notre projet.

CONCLUSION GENERALE :

Dans le cadre de ce travail, nous avons étudié et réalisé une imprimante 3D à dépôt de la matière avec son interface Homme-Machine sur PC et son algorithme de commande dit « firmware » implémenté dans le microcontrôleur Atmega2560.

En commençant, nous avons effectué une étude générale sur les différentes techniques d'impression 3D existants dans l'industrie ainsi que leurs principes de fonctionnement. Après le choix et la validation de la technique à utiliser, nous avons fait une description détaillée sur l'architecture de l'imprimante 3D à réaliser en expliquant le côté mécanique (moteurs, roulements,..) et instrumentation (capteurs) de cette dernière.

Après cela, nous avons entamé la partie hardware de projet en présentant la carte électronique développée pour la commande de procédé en faisant une description détaillée sur les différents éléments qui la composent et leurs caractéristiques. Nous avons également présenté les techniques d'interfaçage utilisées pour relier le procédé au PC et/ou la carte mémoire et leur méthode de configuration sur le microcontrôleur Atmega2560.

Ensuite, nous avons effectué une description de la partie Software en commençant par l'environnement de développement « Arduino IDE » de microcontrôleur Atmel2560, le compilateur « Visual Studio C # » pour le développement d'interfaces graphiques, le logiciel de simulation « ISIS Proteuse », le logiciel de design électrique « EAGLE », le logiciel de conception et de la modélisation 3D « Solidworks », et en terminant avec le logiciel de tranchage et de génération de G-code le « Slic3r ».

Enfin, nous avons présenté dans le dernier chapitre les différents organigrammes de commande élaborés afin de créer l'interface Homme-Machine sur PC et le firmware implémenté dans le microcontrôleur ainsi que, les résultats des tests que nous avons fait.

Perspectives :

L'imprimante 3D réalisée est fonctionnelle, cependant, cette variante est un prototype que nous comptons à l'avenir l'améliorer pour qu'elle soit un produit fini et commercialisable en apportant les modifications et les nouvelles fonctionnalités suivantes :

- Modifier l'architecture mécanique de l'imprimante en fixant la plateforme d'impression et en ajoutant un autre degré de translation pour la tête d'impression sur l'axe Y et cela pour minimiser les vibrations.

- Ajouter une plateforme chauffante pour avoir la possibilité d'utiliser plusieurs types de filaments.
- Ajouter un capteur de distance pour calibrer la plateforme d'impression et prévenir que cette dernière soit inclinée par rapport à la tête d'impression.
- Améliorer l'interface de commande Homme-Machine en ajoutant une fenêtre de visualisation 3D en temps réel de la pièce en cours de fabrication.
- Ajouter un clavier et un LCD graphique pour la commande manuelle d'imprimante directement sans avoir besoins d'utiliser l'interface Homme-Machine.

BIBLIOGRAPHIE

- [1] : AISSAOUI ABDELWAHAB, « *Étude conception et réalisation d'une imprimante 3D* », Mémoire de Master, UMBB, FSI, 2016.
- [2] : OULD ELHADJ Mohammed, « *Etude, Conception et Réalisation d'une Machine de Rebobinage des moteurs électriques Automatique, avec implémentation du Firmware des microcontrôleurs* », Mémoire de Master, UMBB, FHC, 2016.
- [3] : Le datasheet de : « *Nema 17 Stepper Motor* ».
- [4] : Le site web : « <https://www.wikipedia.org> », article sur « les thermistances ».
- [5] : Le datasheet de : « *ArduinoMega* ».
- [6] : ABDAOUI WALID, AMZIANI NAZIM, « *Conception et réalisation d'un procédé commandé par PC avec application au traitement d'image* », Mémoire de Master, UMBB, FHC, 2016.
- [7] : Le datasheet de microcontrôleur : « *Atmega2560* ».
- [8] : Le datasheet de driver : « *A4988* ».
- [9] : Le site web : « <https://www.pololu.com/product/2128> ».
- [10] : Le datasheet de : « *LCD display Hitachi HD44780* ».
- [11] : Le datasheet de transistor Mosfet : « *STP55NF06* ».
- [12] : BOUROUIS THABET, « *Conception et réalisation d'un Système de supervision et de contrôle d'un four électrique avec un software sous Visual C# et une carte électronique à base de microcontrôleur* », Mémoire de Master, UMBB, FHC, 2016.
- [13] : Le site web : « <http://www.circuitbasics.com> », basics of the spi communication protocol.
- [14] : Le site web : « <https://www.arduino.cc> », site officiel de « Arduino ».
- [15] : Le site web : « <https://msdn.microsoft.com> », site officiel de « Microsoft ».
- [16] : Le site web : « www.labcenter.com », une description générale de logiciel « Proteuse ».

[17] : Le site web : « <https://www.autodesk.com/products/eagle> », site officiel de Autodesk.

[18] : Le site web : « <http://www.solidworks.in> », site officiel de « Solidworks ».

[19] : Le site web : « <http://reprap.org/wiki/G-code> », site de la communauté de « Reprap ».

[20] : Le site web : « <http://slic3r.org/> », site officiel de logiciel « slic3r ».