الجمهورية الجزائرية الديمقراطية الشعبية

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE DR. YAHIA FARES DE MEDEA

FACULTE DES SCIENCES ET DE LA TECHNOLOGIE

Département du Génie Électrique et de l'Informatique



Projet de Fin d'Etude

Présenté pour l'obtention du Diplôme
de Master en Génie Électrique

Spécialité : Automatique des systèmes

THEME

CONCEPTION ET REALISATION D'UNE IMPRIMANTE DELTA 3D

Proposé et dirigé par,

Dr.BOUDANA Djamel

M.LABOUKH Aboubakre

Préparé et présenté par,

M.SELAOUI Riadh

Melle BOUDAOUD Chahrazed

Dédicaces

Merci ***ALLAH*** *de m'avoir donné la capacité d'écrire et de réfléchir, la force d'y croire, la patience d'aller jusqu'au bout du rêve et le bonheur de lever mes mains vers le ciel et de dire "****Ya Kayoum****"*

Je dédie ce travail :
A mes très chers parents,
A ma famille,
A tous mes amis,

Et spécifiques Saïdi Madani,
Adlane Faraji, RABOUKH
Aboubakre et riadh selaoui

BOUDAOUD Chahrazed

Dédicaces

Merci ***ALLAH*** *de m'avoir donné la capacité d'écrire et de réfléchir, la force d'y croire, la patience d'aller jusqu'au bout du rêve et le bonheur de lever mes mains vers le ciel et de dire "****Ya Kayoum****"*

Je dédie ce travail à ma mère et à mon père,

A ma sœur,

A ma famille,

A mes amies,

A tous mes êtres chers.

Et spécifiques Saïdi Madani, Adlane Faraji, LABOUKH Aboubakre et chahrazed BOUDAOUD

SELAOUI Riadh

Remerciements

Nous tenons à remercier en premier lieu Dieu qui nous a permis d'achever notre travail.

Nous remercions aussi vivement le

Dr. BOUDANA Djamel et M.LABOUKH Aboubakre qui nous ont guidé, encouragé et aidé durant toute la durée de notre travail.

Nous remercîments vont aussi aux membres du jury qui vont évaluer notre travail.

Et un grand merci à tous ceux qui nous ont aidés de près ou de loin.

الخلاصـة

ركز هذا العمل على دراسة وتصنيع طابعة من نوع دالتا ثلاثية الأبعاد ، فقد قمنا في البداية بتصنيع جميع الأجزاء الميكانيكية اللازمة لمشروعنا، ثم شرعنا بعد ذلك في تجميع الأجزاء (الميكانيكية والإلكترونية) الضرورية ، وللتحكم فيها اعتمدنا على بطاقة برمجة "Arduino Mega 2560 " للسيطرة على نظام ها، وفي النهاية، اختبرنا دقة وسرعة وكذا استقرار طابعتنا دالتا ذات ثلاثية الأبعاد.

الكلمات المفتاحية: طابعة من نوع دالتا ثلاثية الأبعاد , بطاقة برمجة Arduino Mega 2560, دقة, سرعة, استقرار.

RESUME

À l'ère du développement durable, la technologie de l'impression 3D représente un enjeu industriel, économique, sociétal majeur. Ce travail est centré sur l'étude et la réalisation d'une imprimante 3D delta. Dans un premier temps, nous avons fabriqué tous les pièces mécaniques nécessaires à la réalisation de notre projet. Ensuite nous avons utilisé la carte Arduino Mega 2560 pour commander notre système. En fin, après l'assemblage des différentes parties (mécaniques et électronique) nous avons testé la précision, la rapidité et la stabilité de notre imprimante 3D.

Mots clés : Imprimante 3D delta, carte Arduino Mega 2560, précision, rapidité, stabilité.

ABSTRACT

In the era of sustainable development, 3D printing technology is an industrial challenge, economic, major societal. This work is centered on the study and implementation of a 3D delta printer. Firstly, we made all necessary mechanical parts in the realization of our project. Then we used the Arduino Mega 2560 to control our system. Then we used the Arduino Mega 2560 to control our system. In the end, after assembling the various parts (mechanical and electronic) we tested the precision, speed and stability of our 3D printer.

Keywords: 3D Printer delta, Arduino Mega 2560 map, precision, speed, stability.

Liste des figures

CHAPITRE 1: Généralité dur les imprimantes 3D

Figure 1.1: ÉchoRap 6

Figure 1.2: LeSLS (Fritage sélectif par un laser qui agglomère une couche de poudre). 7

Figure 1.3: FTI (Film Transfer Imaging)..... 7

Figure 1.4: MJM (Modelage à Jets Multiples). 8

Figure 1.5: SLA (Stéréolithographie Apparatus). 9

Figure I.6: Le robot delta 9

Figure 1.7: Robot delta, image CAO et graphe d'agencement. 10

Figure 1.8: Le procédé d'impression 3D en quelques étapes simplifiées. 11

CHAPITRE 2: description de l'imprimante delta 3D

Figure 2.1: Schéma synoptique de la réalisation..... 16

Figure 2.2: Schéma synoptique de la réalisation..... 17

Figure 2.3: Différentes étapes de l'assemblage de la partie mécanique de l'imprimante delta 3D…….. 20

Figure 2.4: Le schéma synoptique de la partie électrique 20

Figure 2.5: Les trois types de moteur pas à pas. 22

Figure 2.6: Echelons de courant appliqués aux bobines d'un moteur pas à pas pour réduire son pas…….. 23

Figure 2.7: Arduino Mega 2560..... 23

Figure 2.8: RAMP 1.4..... 31

Figure 2.9: Le driver A4988..... 31

Figure 2.10: La structure de driver A4988. 32

Figure 2.11: Affecheur LCD 33

Figure 2.12: Les points GND et DIR du driver..... 33

Figure 2.13: Placement des drivers à la RAMPS 1.4..... 34

Figure 2.14: Deux faces de l'Arduino avec le RAMP 1.4. 34

Figure 2.15: Connecteurs End Stop et les différentes positions Min et Max. 35

Figure 2.17: Aimantation lit chauffant, moteur et la carte ainsi les sorties d'extrudeur, ventilateur et lit chauffant. 36

Figure 2.18: Branchements sur le RAMP. 36

CHAPITRE 3: Softwares de commande du l'imprimante delta 3D

Figure 3.1: Chaine d'acquisition. 38

Figure 3.2: Installation de l'Arduino Software à l'ordinateur 39

Figure 3.3: Le chargement du firmware 39

Figure 3.4: Fichiers du firmware. 40

Figure 3.5: Programme de Marlin. 40

Figure 3.6: Règles générales des commentaires. 41

Figure 3.7: Vitesse de communication (Arduino baud rate). 42

Figure 3.8**:** La carte mère (Motherboard). 42

Figure 3.9: Nombre d'extrudeurs. 43

Figure 3.10: Choix de la sonde de la température. 44

Figure 3.11: Inversion de la logique des interrupteurs. 47

Figure 3.12: Inversion de direction des moteurs. 48

Figure 3.13: Limites de l'impression. 50

Figure 3.14: Enregistrement les modifications. 52

Figure 3.15: Les déférentes étapes de l'impression. 60

Figure 3.16: Fichier .stl. 61

Figure 3.17: Pièce à imprimer. 62

Figure 3.18: Operation de tranchage. 62

Figure 3.19: Pièce après le trancheur 63

Figure 3.20: Fichier g-code. 63

Figure 3.21: Contrôle manuel de l'ogiciel Repetier-Host. 64

Figure 3.22: Piece prête à l'impression. 64

Figure 3.23: Imprimante delta 3D 65

Liste des tableaux

Tableau 1.1 : Les conventions des graphes d'agencements 10

Tableau 2.1 : Paramètres de la carte Arduino Mega 2560 R3 25

Table de matiere

Introduction générale……. 1

CHAPITRE 1: Généralité dur les imprimantes 3D

1.1 Introduction 4

1.2 Historique 4

1.3 Différentes technologies d'impression 3D 5

1.3.1 AM ou ALM (Additive -Layer- Manufacturing) 5

1.3.2 FTI (Film Transfer Imaging) 7

1.3.3 MJM (Modelage à Jets Multiples) 8

1.3.4 SLA (Stéréolithographie Apparatus) 8

1.4 Définition de robot delta 9

1.4.1 Le graphe d'agencement 10

1.4.2 Principe de robot delta 3D 10

1.5 L'impact sur l'environnement 12

1.6 Avantages et les inconvénients de l'imprimante 3D 13

1.6.1 Les avantages 13

1.6.2 Les inconvénients 14

1.7 Conclusion 14

CHAPITRE 2: description de l'imprimante delta 3D

2.1 Introduction 16

2.2 Schéma synoptique de la réalisation 16

2.3 Partie mécanique 16

2.3.1 Modèle cinematique du robot 17

2.3.2 Construction de l'imprimante delta 3D 18

2.3.3 Assemblage de la partie mécanique 19

2.4 Partie électrique 20

2.4.1 Actionneur pas à pas 21
2.4.2 Partie électronique 23
2.4.2.1 Carte de commande 23
2.4.2.2 RAMP 1.4 31
2.4.2.3 Driver moteur pas à pas 31
2.4.2.4 LCD 32
2.5 Montage de la partie électrique 33
2.6 Conclusion 36

CHAPITRE 3: Softwares de commande du l'imprimante delta 3D

3.1 Introduction 38
3.2 Chaine d'acquisition 38
3.3 Etape de commande 38
3.4 Paramètres mécaniques 45
3.5 Paramètres de mouvements 50
3.6 Compilation 51
3.7 Impression d'un objet 58
3.7.1 Impression depuis la carte SD-Card 58
3.7.1.1 Arrêtement d'impression 58
3.7.1.2 Modification des paramètres d'impression 59
3.7.1.3 Retrait de notre pièce terminé 59
3.7.2 Impression depuis un ordinateur 60
3.7.2.1 Repetier-Host 61
3.7.2.2 Génération du format g-code 61
3.7.2.3 Imprission de g-code généré 63
3.5 Apliquation pratique de l'imprimante delta 65
3.6 Conclusion 65
Conclusion generale 67
Annexe 68
Bibliographique 80

Introduction générale

Introduction générale

L'impression tridimensionnelle se présente aujourd'hui comme le possible déclencheur d'une troisième révolution industrielle. Cette innovation technologique, qui se montre comme successeur de la longue histoire de l'impression, permet la réplication physique d'objets a partir d'un modèle digital. La recherche dans ce secteur ne cesse d'augmenter depuis sa naissance en 1984 et son industrie se place au sixième rang mondial pour la croissance. L'impression 3D n'est plus aujourd'hui réservée aux grandes entreprises, mais se démocratise notamment autour d'une communauté nommée Reprap.

La reprap (pour « replicating rapid prototyper ») est une imprimante 3D libre et open-source tant au niveau du matériel que du logiciel. Elle permet de réaliser des pièces en 3 dimensions à partir d'une modélisation sur ordinateur à l'aide de logiciels libres.

Dans ce mémoire, nous allons réaliser une imprimante delta 3D et étudier le coté hardware et software.

Dans le premier chapitre nous avons présenté les notions essentielles de l'imprimante delta 3D et ses différentes technologies, ainsi que la définition de robot delta et les avantages et les incontinents de l'imprimante 3D.

Nous nous intéressons dans le deuxième chapitre à la structure mécanique et à la partie électronique de l'imprimante 3D. Nous commencons par le dimensionnement de notre imprimante 3D. Puis, nous fabriquons les pièces mécaniques nécessaires à la réalisation de notre projet. Afin de commander notre système nous utilisons la carte Arduino Mega 2560 qui est simple à programmer. En fin, nous assemblons les parties mécaniques et électroniques de l'imprimante 3D.

Dans le troisième chapitre, nous adoptons les différentes étapes de la commande de notre imprimante 3D. Nous débutons par la schématisation d'une pièce par SolidWorks et la sauvegardé au format STL. . La pièce est ensuite tranchée avec le logiciel KISSlicer et enregistrée au format g-code. A la fin, nous branchons l'imprimante 3D à l'ordinateur via le port USB (ou avec la carte SD-Card) qui contient le fichier g-code dans l'interface de pilotage de l'imprimante Repetier-Host et nous lancons l'opération d'impression en exécutant le fichier g-code.

A la fin, nous terminons ce mémoire par une conclusion générale.

1.1 Introduction

L'impression 3D (ou Fabrication par addition) est un procédé de fabrication d'un objet solide en trois dimensions de pratiquement n'importe quelle forme à partir d'un modèle numérique. Elle est obtenue à l'aide d'un processus additif, où les couches successives de matériaux sont fixées dans des formes différentes.

1.2 Historique

Si le phénomène Impression 3D s'emballe depuis quelques années, le procédé existe pourtant depuis le milieu des années 1980. Voici un bref historique de l'impression 3D présentant les grands tournants techniques ou pratiques.

- **1983-1986 : la naissance de l'impression 3D**

Cette première imprimante 3D débouchera sur le premier modèle de série en 1988 : la SLA-2502 de 3D Systems. L'imprimante sert alors aux industriels à créer des objets pour tester leur design avant de décider la production des pièces en série.

- **1990-1992 : l'impression couche après couche**

Ce n'est pas encore parfait loin de là mais le potentiel énorme de l'impression 3D se dévoile avec la création couche après couche d'objets 3D. Pas super pratique mais plutôt efficace, l'impression 3D prouve qu'elle est capable de créer des pièces complexes.

- **2005 : Lancement du RepRap Project**

Le Dr Adrian Boyier et son équipe de l'Université de Bath imaginent une prouesse technologique en lançant la construction d'une imprimante 3D capable de créer les pièces utiles à son fonctionnement. Leur but ? Rendre le plus accessible possible l'impression 3D où une machine auto-replicante « viraliserait » son usage et les services qu'elle peut apportait.

- **2010 : Un vaisseau sanguin imprimé avec une imprimante 3D**

L'impression 3D à vocation médicale fait son petit bout de chemin mais à la fin 2010 tout s'accélère. En décembre 2010, une équipe de chercheur parvient à créer avec une bio-imprimante 3D Organovo un vaisseau sanguin fonctionnel.

- **2011 : Un premier drone à 5000 Livres Sterling**

Les ingénieurs de l'Université de Southampton parviennent à créer un avion sans pilote en une semaine pour le prix de 5000 Livres Sterling. Sulsa (c'est le nom de cet avion)

mesure prêt de 2m d'envergure et est propulsée par un moteur électrique lui permettant d'atteindre la vitesse de 160km/h. Imprimées grâce à la technologie SLS, les pièces du fuselage ont été produites en un temps record et avec une finesse remarquable.

- **2011 : Après l'avion : la voiture**

Lors de la conférence TedX de Winnipeg au Canada, la société Kor Ecologic présente Urbee, une voiture dont la carosserie est complètement composée de pièces imprimées à l'aide d'une imprimante 3D. Urbee « eco-friendly » est dessinée pour consommer le moins de carburant possible : 1,18 litres aux cent kilomètre sur autoroute et moitié moins en ville. On estime qu'elle couterait entre 10000 et 50000 dollars si le projet commercial est viable.

- **2011 : L'or et l'argent en impression Grand Public**

i.materialise.com propose l'impression d'objets en or ou en argent. Plus que jamais le monde de l'imprimante 3D fait les yeux doux aux artisans et bijoutiers. Plus besoin de moules pour tirer des pièces en série, le bijou est unique et à prix réduit.

- **2013 : Une arme à feu fonctionnelle**

Triste jour pour l'impression 3D avec l'un de ses plus gros buzz : la diffusion d'une vidéo, puis plus conséquemment, des plans d'une arme à feu. Le tout fait polémique et remonte jusqu'au gouvernement américain qui interdit leurs diffusions [1].

1.3 Différentes technologies d'impression 3D

Toutes les technologies de l'imprimante 3D sont basées sur la découpe de l'objet virtuel 3D en lamelles 2D de très fine épaisseur. Ces fines lamelles sont déposées une à une en les fixant sur les précédentes jusqu'à reconstituer l'objet réel. Différentes technologies sont aujourd'hui développées et utilisées pour réaliser des impressions tridimensionnelles.

1.3.1 AM ou ALM (Additive -Layer- Manufacturing)

La technique AM est défini par la norme ASTM comme « le processus d'assemblage de matériaux pour fabriquer des objets à partir des données du modèle 3D, le plus souvent couche après couche, par opposition aux méthodes de fabrication soustractives. Synonymes : fabrication additive, les procédés additifs, les techniques additives, la fabrication par couche additive, la fabrication des couches et fabrication de forme libre».

Le terme additif décrit les technologies qui peuvent être utilisées n'importe où dans le cycle de vie du produit, de la pré-production (prototypage rapide) à la production à pleine échelle (fabrication rapide) et même pour les applications d'outillage ou de personnalisation de post-production. Le Fused Deposition Modeling et le laser sintering sont deux exemples de technique AM.

- FDM (Fused Deposition Modeling)

Cette technique consiste à faire fondre une résine (généralement du plastique) à travers une buse chauffée à haute température. Un petit fil de plastique en fusion, d'un diamètre de l'ordre du dixième de millimètre, en sort. Ce fil est déposé en ligne et vient se coller par re-fusion sur ce qui a été déposé au préalable. Le Fused Deposition Modeling est une marque déposée par l'inventeur de la technologie de dépôt de filament en fusion, Stratasys [2]. La figure (1.1) montre une imprimante 3D québécoise Crée par Stéphane Rousseau [Robotic Sequencing.com] [3].



Figure 1.1: ÉchoRap

- SLS (Selective Laser Sintering)

Le procédé SLS est une technique de prototypage rapide par frittage laser sélectif sans phase liquide. Il est utilisé pour créer des objets 3D, strate par strate, à partir de poudres qui sont frittées ou fusionnées grâce à l'énergie d'un laser de forte puissance, comme un laser CO2. Grâce à la variété des matériaux pouvant être utilisés par ce procédé, une large gamme d'applications est possible [2]. La figure (1.2) illustre le fritage sélectif par un laser qui agglomère une couche de poudre [3].

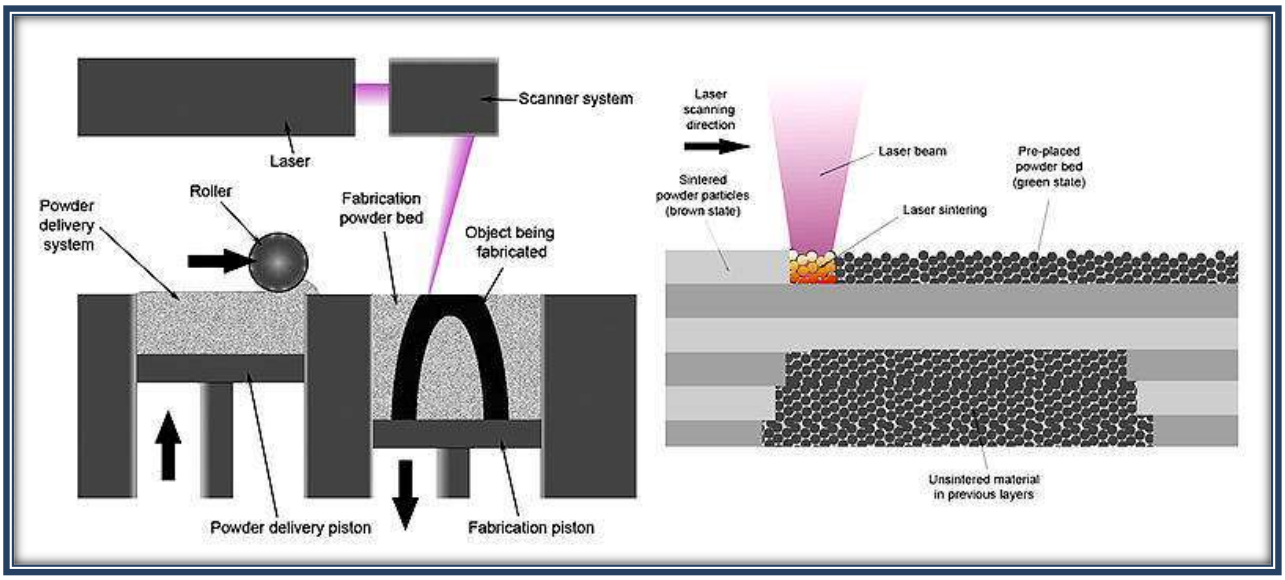


Figure 1.2: Le SLS (Fritage sélectif par un laser qui agglomère une couche de poudre).

1.3.2 FTI (Film Transfer Imaging)

Un film transparent recouvert d'une couche de résine photopolymère est placé devant le vidéo projecteur intégré à la machine, l'image de la coupe 2D projetée dessus va faire durcir la résine. Le plateau de production est remonté d'une épaisseur tandis que le film transparent fait un aller-retour dans la cartouche afin de recevoir une nouvelle couche de résine liquide, l'image de la coupe 2D suivante est projetée dessus et ainsi de suite. La pièce est ainsi reconstituée couche par couche [2] (figure 1.3).

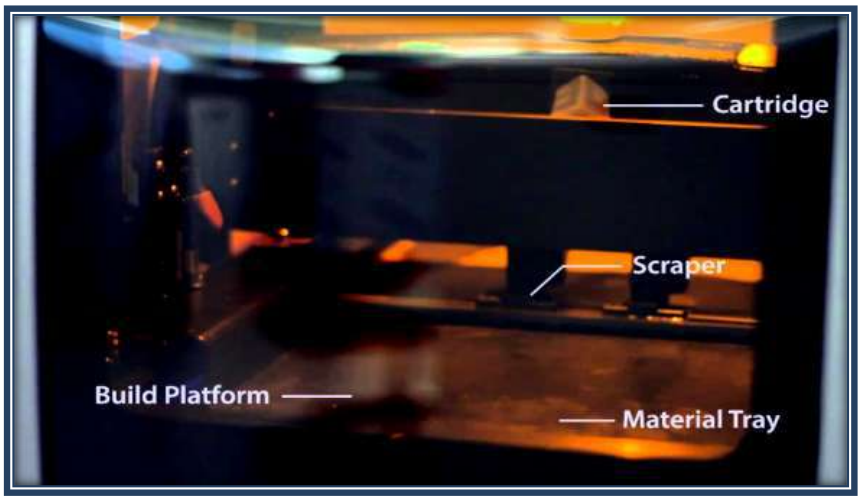


Figure 1.3: FTI (Film Transfer Imaging)

1.3.3 MJM (Modelage à Jets Multiples)

Cette technique consiste à déposer une couche de résine (du plastique type acrylate ou polypropylène) liquide de la même manière qu'une imprimante à jet d'encre avec une épaisseur de 2/100 à 4/100 de millimètres [2]. La figure (1.4) montre le Modelage à Jets Multiples.

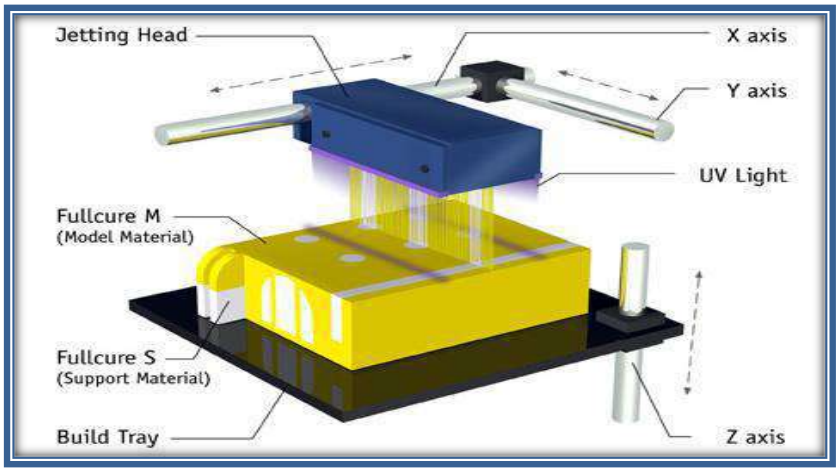


Figure 1.4: MJM (Modelage à Jets Multiples).

1.3.4 SLA (Stéréolithographie Apparatus)

Cette technique est similaire à la SLS mais utilise en général une résine liquide spéciale photosensible (au lieu d'une poudre). Elle consiste à fabriquer des pièces par ajout de matière. L'empilement successif de couches pouvant aller jusqu'à 0,05 mm permet de fabriquer les pièces. A chaque passe, la résine est traitée par un faisceau laser qui apporte la longueur d'onde nécessaire pour durcir le matériau. de base sous forme liquide ainsi que la grande finesse du faisceau laser double spot permet d'obtenir des pièces très précises avec un niveau de détails plus poussé [2] (figure 1.5).

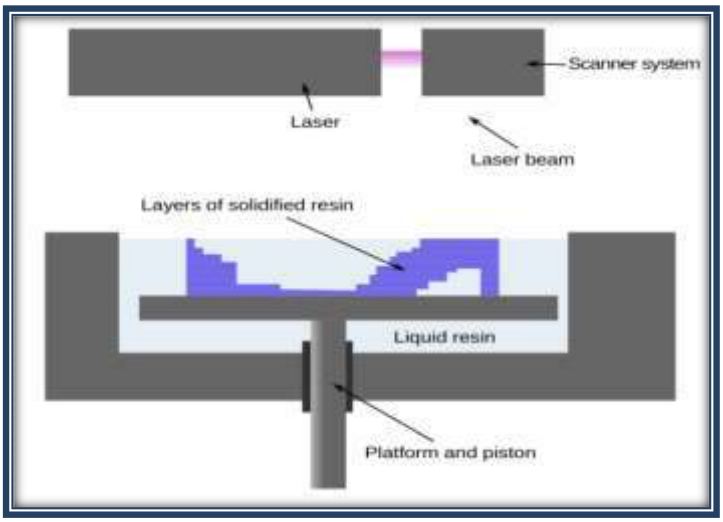


Figure 1.5: SLA (Stéréolithographie Apparatus).

1.4 Définition de robot delta

Le robot Delta est né d'un besoin industriel. Les performances visées (cadence de travail élevée, grande précision, puissance consommée faible) excluaient l'utilisation de robots du marché. Il a été développé début des années 80 à l'École Polytechnique Fédérale de Lausanne (EPFL) par Clavel. Le robot Delta qui a une structure pleinement légère, est constitué d'une base fixe et d'une plate-forme mobile appelée aussi nacelle. Elles sont liées par trois chaînes cinématiques identiques constituées d'un bras et de deux barres parallèles formant un parallélogramme.

Chaque bras est entraîné par un moteur-réducteur, solidaire de la base. La nacelle reste toujours parallèle à la base, les mouvements de translation de cette dernière résultent du mouvement combiné des trois actionneurs [4].



Figure 1.6 : Le robot delta

1.4.1 Le graphe d'agencement

Le graphe d'agencement représente la topologie de la structure des robots et mettent en évidence les diverses branches et boucles. La figure 1 présente les conventions utilisées pour ces graphes [5] [4].

Nom de la liaison	Représentation	
	Liaison passive	Liaison motorisée
Rotoïde (pivot)	R	R
Prismatique (glissière)	P	P
Universelle (cardan)	U	
Sphérique (rotule)	S	

Tableau 1.1 : Les conventions des graphes d'agencements

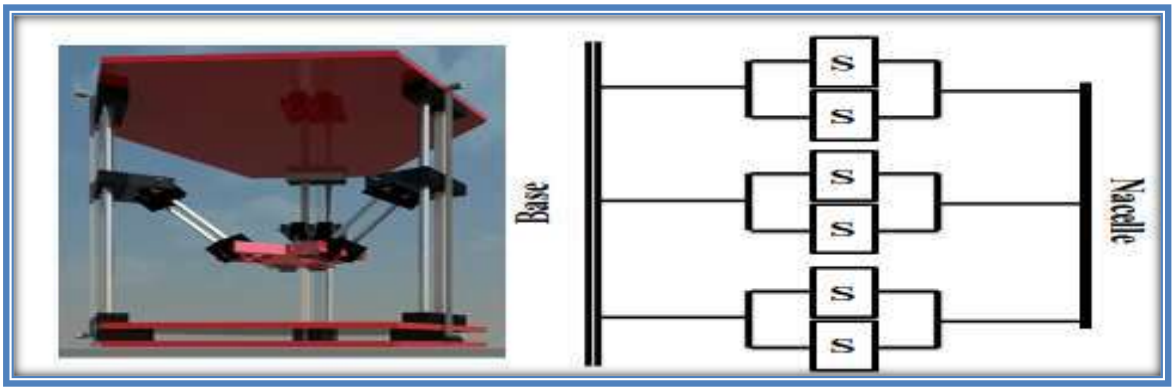


Figure 1.7 : Robot delta, image CAO et graphe d'agencement.

1.4.2 Principe de robot delta 3D

L'objectif fondamental d'une imprimante 3D est de transformer rapidement une idée en objet réel. Cette idée est d'abord consignée sous un modèle informatique 3D créé par des logiciels en CAO 3D tels que *SolidWorks*, *Autodesk*, *Inventor* ou *Pro/ENGINEER*. Il existe de nombreux logiciels de CAO capables de produire des fichiers 3D Par exemople: 3D *Studio Max, MicroStation, Mimics….*

Tous ces outils logiciels exportent des modèles 3D sous forme de fichiers aux formats standards pour des imprimantes en 3D, les plus courants étant les formats suivants : .STL, .WRL (VRML), .PLY, .3DS et .ZPR. Notons que les fichiers .STL tendent à s'imposer et

pourraient devenir la norme. Le fichier exporté est un maillage constitué d'une série de triangles orientés dans l'espace, lesquels déterminent un volume en 3D. Ce maillage doit être fermé pour être considéré comme un solide et pas seulement un ensemble de surfaces sans épaisseur. En d'autres termes, à ce stade, la création doit être prête à passer dans le monde réel, non plus seulement sur un ordinateur.

Le fichier .STL est lu par un logiciel spécialisé d'impression 3D fourni avec l'imprimante (des versions open-source existent). L'utilisateur peut alors modifier le fichier qu'il souhaite imprimer, orienter par exemple l'élément dans la chambre de construction ou piloter l'imprimante 3D afin d'imprimer plusieurs versions de l'élément. Le logiciel d'impression 3D découpe en tranches le fichier du modèle 3D sous forme de centaines de coupes transverses numériques. Le grain est déterminé selon les caractéristiques de l'imprimante et l'aspect désiré de l'objet final : 0,1 mm peut être considéré comme un grain moyen sur les imprimantes grand public.

Chaque tranche de 0,1 mm correspond à une section du modèle à fabriquer. Une fois l'impression lancée, les sections du futur objet sont envoyées une à une à l'imprimante 3D qui commence la construction immédiatement. La machine imprime séquentiellement chaque couche, l'une au-dessus de l'autre, construisant ainsi un objet réel à l'intérieur de la chambre de construction de la machine. Une fois que l'imprimante 3D termine la dernière couche, un cycle de séchage court commence. Puis l'objet réel peut-être retiré, et potentiellement subir un traitement de finition si nécessaire (ponçage, cuisson pour la dureté, etc.) [6].

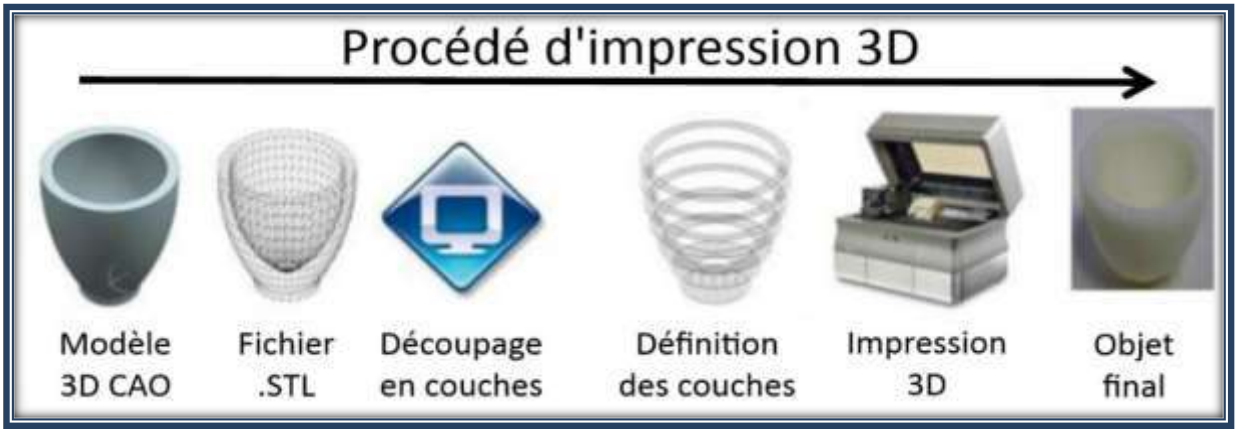


Figure1.8: Le procédé d'impression 3D en quelques étapes simplifiées.

1.4.3 Matériau

Selon le procédé, une panoplie de matériaux peut être utilisée [3]:

Matériaux organiques	Céramiques	Plastiques	Métaux
Cires	Alumine	ABS (acrylonitrile butadiène styrène)	Aluminium
Tissus/Cellules	Mullite	PLA (acide polylactique)	Acier d'outillage
Matières alimentaires Chocolat, fromage, houmous, glaçage, pâte, etc.	Zircone	Polyamide (nylon)	Titane
	Carbure de silicium	Polyamide renforcé	Inconel
	Phosphate tricalcique β	PEEK (polyétheréthercétone)	Cobalt-chrome
	Résines époxy chargées en céramique (nano)		Résines époxy thermodurcissables
	Silice (sable)	Cuivre	Acier inoxydable
	Plâtre	PMMA (polyméthacrylate de méthyle)	Or/Platine
	Graphite	PC (polycarbonate)	Hastelloy
		PPSU ou PPSF (polyphénylsulfone)	
		Ultem	
		Alumide	

1.5 L'impact sur l'environnement

L'impression 3D requiert des imprimantes assez importantes (pour les professionnels) et les constructeurs d'imprimantes 3D ne sont pas plus écologiques que n'importe quel autre constructeur de matériels de conception (machine outils, robots...).

Les consommables ne sont pas non plus exemptes d'impact écologique. Que ce soit les résines plastiques ou les poudres polymères, les matériaux utilisés ne sont pas plus et pas moins polluants que d'autres matériaux classiques équivalents.

Même si actuellement des avancées de l'industrie pétrochimique ont lieu avec "des éco-polymères", comme le PLA (polyactic acid) qui est peut être la plus écologique des matières utilisées pour l'impression 3D, car elle est d'origine renouvelable, c'est ailleurs qu'il faut chercher le caractère profondément écologique de l'impression 3D.

L'impression 3D permet de produire localement à l'unité ou en petite série toutes sortes de pièces. On économise donc :

- ✓ Le transport
- ✓ L'emballage
- ✓ Le stockage
- ✓ Les déchets de production

Chez ZCorporation les imprimantes sont présentées de la manière suivante :

"Le fonctionnement de la machine est écologique sans aucune production de produit toxique et les déchets sont recyclables."

" Les matériaux non utilisés sont recyclés pour la conception suivante afin d'éviter tout gaspillage."

"Système de gestion de la poudre parmi les plus avancés actuellement disponibles sur une imprimante 3D à jet d'encre :

- système en boucle fermé de chargement, élimination et recyclage de la poudre ;
- pression négative continue pour contenir les particules en suspension dans l'enveloppe de la machine ;
- pas de déchet liquide à manipuler ;
- utilise des matériaux de fabrication écologiques et innoffensifs" [7].

1.6 Les avantages et les inconvénients de l'imprimante 3D

Les avantages et les inconvénients de l'imprimante 3D sont :

1.6.1 Les avantages

- ➢ Fabriquer des objets de toutes sortes ;
- ➢ Réparer les objets cassés ;
- ➢ L'amélioration des conditions de vie ;
- ➢ Forte personnalisation possible des biens et produits, à bas coût ;
- ➢ Design et production efficace (légèreté, résistance, complexité) ;
- ➢ Variété de production (capacité à produire des biens différents) ;

- Capacité de production d'objets très petits (échelles macro et nanoscopiques) ;
- Peu de déchets lors de la fabrication ;
- Production à la demande, en volumes réduits ;
- Production locale, proche des lieux de consommation ;
- Production accessible à tous, peu de barrières à l'entrée ;
- Fort potentiel d'innovation dans la distribution et la fourniture de services.

1.6.2 Les inconvénients

- Matériaux et tailles limités ;
- La potentielle création d'armes ;
- Production non compétitive de grands volumes de produits ;
- Variété encore limitée des matériaux utilisés pour les impressions ;
- Coût potentiellement élevé des matériaux de base utilisés comme intrants ;
- Difficulté de combiner plusieurs matériaux (notamment dans l'électronique) ;
- Impression complexe d'objets de grande taille ;
- Qualité et durabilité limitées des biens finaux [1] [6].

1.7 Conclusion

Dans ce chapitre on a présenté des notions importantes sur l'imprimante 3D ainsi que leur différente technologie, puis on s'est intéressés au robot delta et son principe de fonctionnement. En fin on a présenté les avantages et les inconvénients de l'imprimante 3D.

Dans le chapitre suivant, on s intéresse à l'étude et à la modélisation du robot Delta.

2.1 Introduction

Nous avons toujours voulu créer des propres objets et de transformation nos dessins 3D en objets réels. Mais aujourd'hui Grâce à l'imprimante delta 3D printer nous pouvons construire n'importe quelque chose. Dans ce chapitre nous expliquons la partie mécanique et la partie électrique de notre robot delta 3D printer.

2.2 Schéma synoptique de la réalisation

Le schéma de la figure (2.1) explique comment les données sont transférées d'un micro-ordinateur (par câble) ou avec une carte vers le robot en passant par le système de communication qui est l'interface qui permet du transfert de la commande vers le robot.

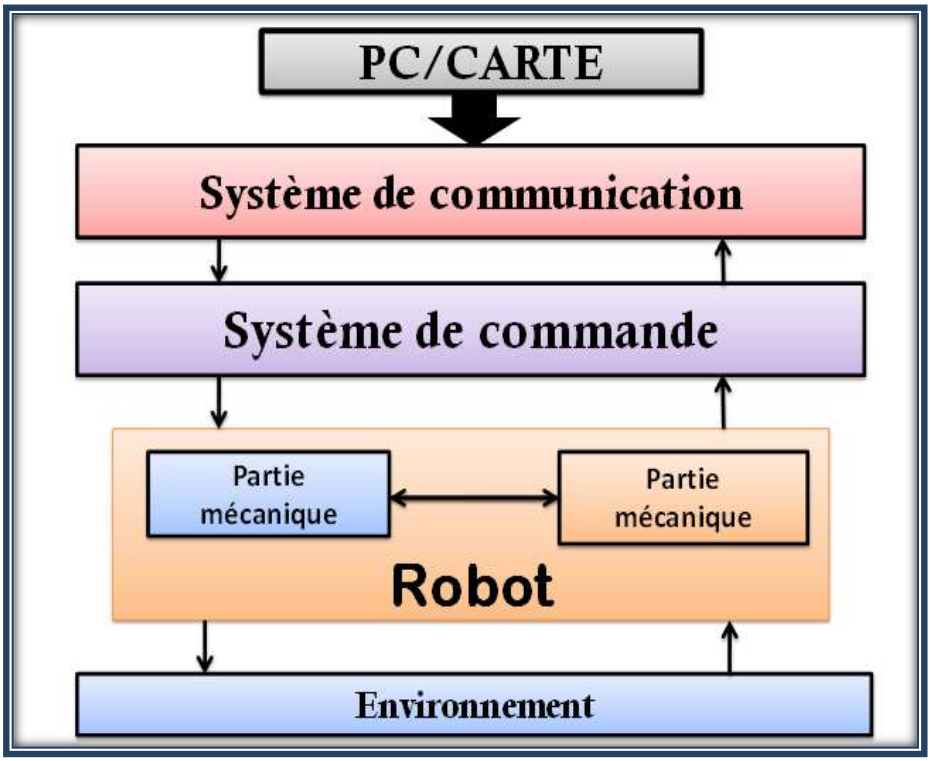


Figure 2.1 : Schéma synoptique de la réalisation

2.3 Partie mécanique

Le robot delta 3D est construit avec des matières légères et solides (bois, acier et téflon) pour diminuer le poids global du robot. Il est composé de :

2.3.1 Modèle cinimatique du robot

A partir de l'architecture voulue pour le robot, on fourni les épures suivantes caractérisant la géométrie globale du robot. On a remplacé les paires de biellettes par une unique biellette articulée en ses extrémités afin de simplifier l'étude.

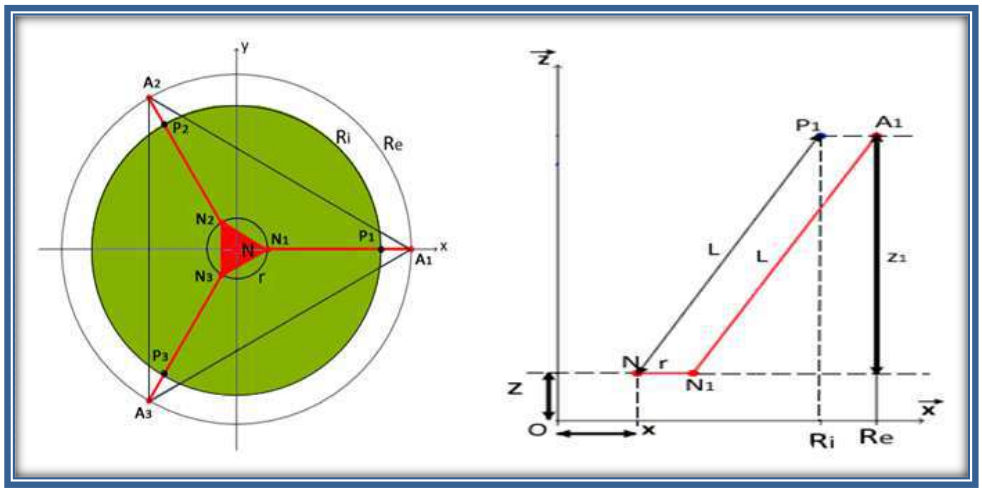


Figure 2.2 : Schéma synoptique de la réalisation

Le centre de l'effecteur est noté N et est défini par ses coordonnées x, y et z dans le repère R (O,x,y,z).Le point "O" représente le centre de la base inférieure du robot. On note A_i (i=1, 2, 3) les centres des liaisons Biellette /Coulisseau et N_i les centres des liaisons Effecteur /Biellette.

Les points P_i sont homothétiques aux points A_i et situés sur le cercle de rayon R_i. Les points P_i sont définis par leurs coordonnées x_i, y_i et z_i dans le repère (O,x,y,z) et seuls x_i, y_i sont constants lors du fonctionnement du robot.

- ✓ Z_i : altitude d'un coulisseau i ;
- ✓ L : longueurs des biellettes ;
- ✓ R_e: rayon du cercle passant par les points A_i ;
- ✓ R : rayon du cercle passant par les points N_i (le triangle formé par les points N est équilatéral). On a $R_i = R_e - r$;
- ✓ R_i : rayon du cercle limite ne pouvant pas dépassé (sinon l'effecteur peut heurter les axes linéaire).

L'altitude Z_i des coulisseaux est définie par la relation suivante [8]:

$$Z_i = Z + \sqrt{L^2 - (X - X_i)^2 - (Y - Y_i)^2} \qquad (1.1)$$

2.3.2 Construction de l'imprimante delta 3D

L'imprimante delta 3D est construite par les partie suivantes :

- **La partie mecanique**

➢ **Tiges**

 12x Tige	 6x Tige filtrie

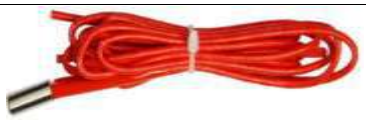
➢ **Pièces mécaniques**

 3x Courroie GT2	 3x Poulie GT2	 Poulie

➢ **Visserie**

 6xVis M3x10 mm 14**x** Vis M3x20 mm 20**x** Vis M3x25 mm 15**x** Vis M3x30 mm 11**x** Vis M3x50 mm 4x Vis M4x25mm 6x Vis M2.5x16mm 3x Vis à bois	 20x Rondelle Ø3 7x Rondelle Ø4 moyenne 5x Rondelle Ø4 grande	 25x Ecrou M3 4x Ecrou M3 à oreilles 8x Ecrou M4 6x Ecrou 2.5

- **Kit Hexagon**

 3x Plateau	 1x Thermistance	 1x Cartouche de chauffe
 1x Buse Hexagon		

- **La partie electrique**

➢ **Electronique**

 **1x** shield	 4x Moteur NEMA 17	 3x capteur fin de course
 3x Ventilateur 3x3	 **5x** driver moteur pas-à-pas	 1x Alimentation
 Arduino Mega 2560		

2.3.3 Assemblage de la partie mécanique

Les figures (2.3) montrent les différentes étapes de l'assemblage de la partie mécanique de l'imprimante 3D delta.

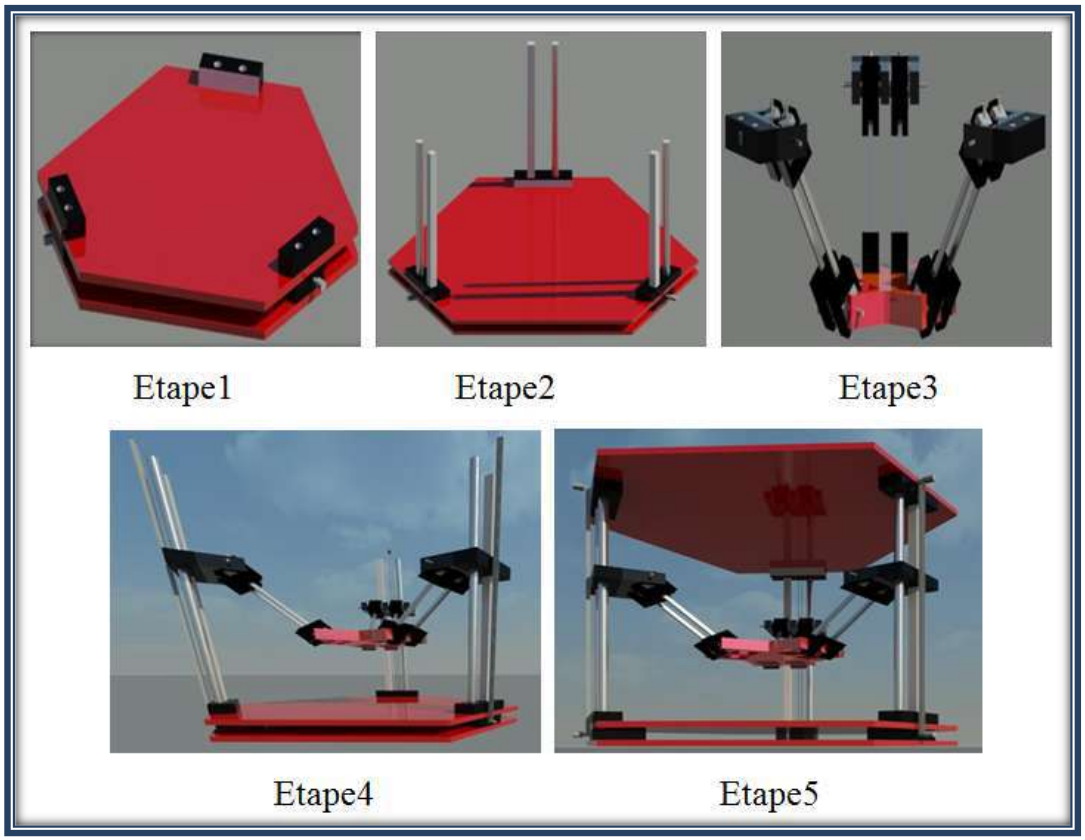


Figure 2.3: Différentes étapes de l'assemblage de la partie mécanique de l'imprimante delta 3D

2.4 Partie électrique

Le schéma de la figure (2.4) donne une description sur les constituants de la partie électrique et la liaison entre eux.

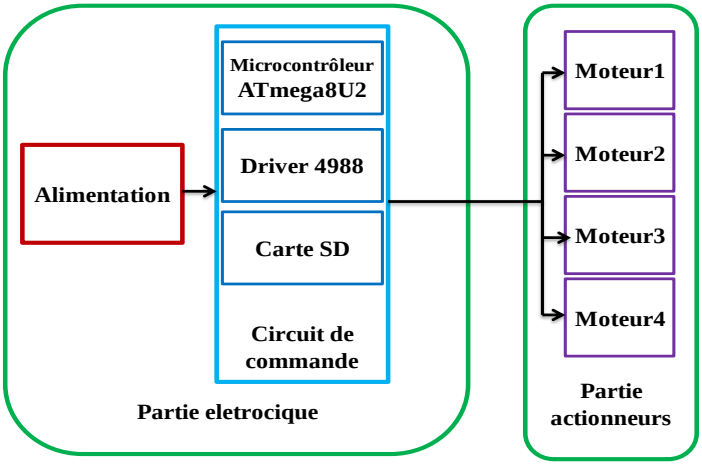


Figure 2.4: Le schéma synoptique de la partie électrique

2.4.1 Actionneur pas à pas

La circulation d'un courant électrique dans un bobinage entraine l'apparition d'un champ magnétique , et donc la présence de pôles Nord et Sud (deux pôles de même nature se repoussent, deux pôles Nord et Sud s'attirent) ; c'est sur ce principe de base que repose le fonctionnement de tout moteur électrique, et, de manière plus générale, de bon nombre de dispositifs électromécaniques : relais, compteurs, galvanomètres, certains haut-parleurs ou microphones, gâches électriques de porte, etc.

Le moteur pas à pas est un moteur qui tourne en fonction des impulsions électriques alimentant ses bobinages. Selon son alimentation électrique, il peut être de type :

- Unipolaire si ses bobinages sont toujours alimentés dans le même sens par une tension unique, d'où le nom d'unipolaire ;
- Bipolaire lorsque ses bobinages sont alimentés tantôt dans un sens, tantôt dans l'autre sens. Ils créent une fois un pôle Nord, une fois un pôle Sud d'où le nom de bipolaire.

Les moteurs pas à pas peuvent être à réluctance variable, à aimants ou une combinaison des deux (Figure 2.5).

L'angle de rotation minimal entre deux modifications des impulsions électriques s'appelle un pas. On caractérise un moteur par le nombre de pas par tour (c'est-à-dire pour 360°). Les valeurs courantes sont 48, 100 ou 200 pas par tour.

La rotation du moteur se fait donc de manière discontinue. Pour améliorer la résolution, ce nombre de pas peut être augmenté de manière purement électronique (fonctionnement en micropas).

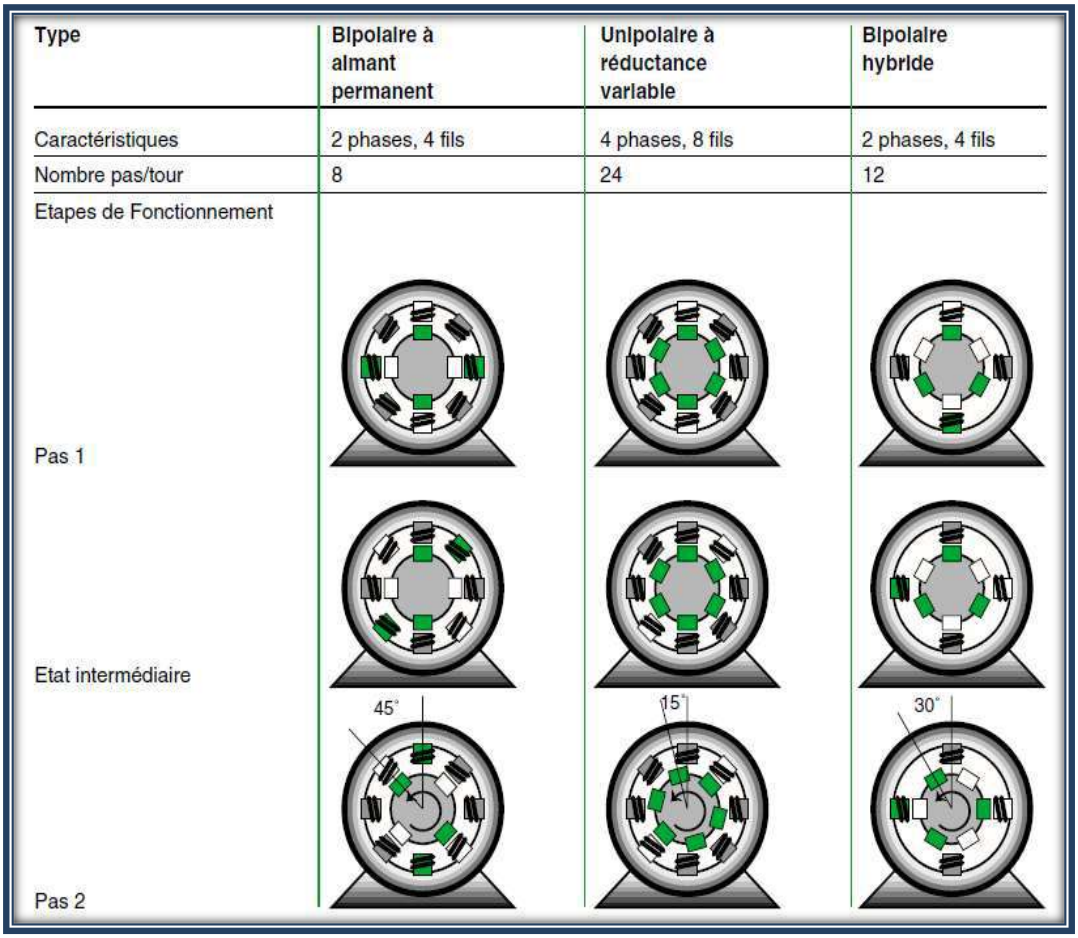


Figure 2.5: Les trois types de moteur pas à pas.

En faisant varier par échelon le courant (Figure 2.6) dans les bobines, on crée un champ résultant qui glisse d'un pas à un autre, ce qui a pour résultat la réduction effective du pas. Les circuits pour micropas multiplient par 500 le nombre de pas d'un moteur qui passe ainsi, par exemple, de 200 à 100 000 pas.

L'électronique de commande permet de piloter la chronologie de ces impulsions et d'en comptabiliser le nombre. Les moteurs pas à pas et leur circuit de commande permettent ainsi la rotation d'un axe avec beaucoup de précision en vitesse et en amplitude [9].

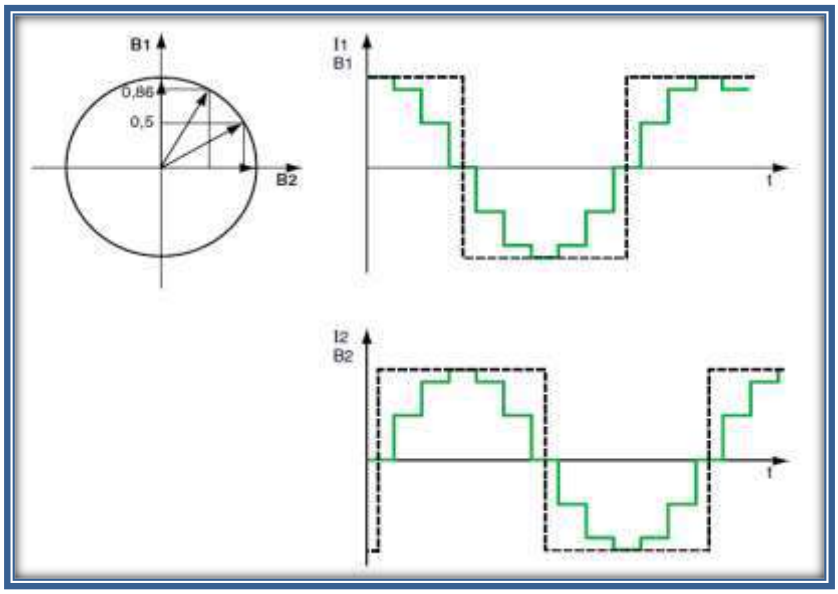


Figure 2.6: Echelons de courant appliqués aux bobines d'un moteur pas à pas pour réduire son pas.

2.4.2 Partie électronique

La partie électronique contient :

2.4.2.1 Carte de commande

La carte Arduino Mega 2560 est une carte à microcontrôleur basée sur un ATmega2560. Cette carte dispose :

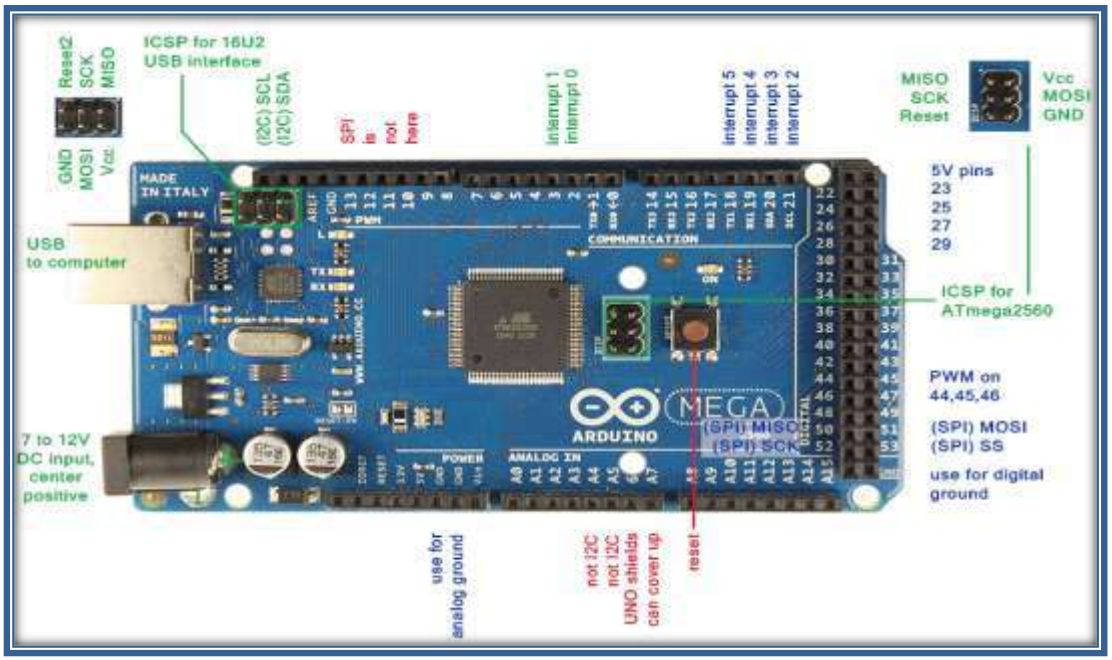


Figure 2.7: Arduino Mega 2560

- 54 broches numériques d'entrées/sorties (dont 14 peuvent être utilisées en sorties PWM (largeur d'impulsion modulée)) ;
- 16 entrées analogiques (qui peuvent également être utilisées en broches entrées/sorties numériques) ;
- 4 UART (port série matériel) ;
- D'un quartz 16Mhz ;
- D'une connexion USB ;
- D'un connecteur d'alimentation jack ;
- D'un connecteur ICSP (programmation "in-circuit") ;
- Et d'un bouton de réinitialisation (reset).

Elle contient tout ce qui est nécessaire pour le fonctionnement du microcontrôleur. Pour pouvoir l'utiliser et se lancer, il suffit simplement de la connecter à un ordinateur à l'aide d'un câble USB.

La carte Arduino Mega 2560 est compatible avec les circuits imprimés prévus pour les cartes Arduino Uno, Duemilanove ou Diecimila.

1. Paramètres de la carte Arduino Mega 2560

Les paramètres de la carte Arduino Mega 2560 sont illustrés dans le tableau suivant :

Microcontrôleur	**ATmega2560**
Tension de fonctionnement	5V
Tension d'entrée (recommandé)	7-12V
Tension d'entrée (limites)	6-20V
E / S numériques Pins	54(Dont 14 fournissent la sortie PWM)
Pins d'entrée analogique	16
DC Courant par E / S Pin	40 mA
Courant DC pour 3.3V Pin	50 mA
Mémoire flash	256 KB dont 8 KB utilisé par bootloader
SRAM	8 KB
EEPROM	8 KB

Vitesse de l'horloge	16 MHz

Tableau 2.1 : Paramètres de la carte Arduino Mega 2560 R3

2. Alimantation de la carte

➢ La carte Arduino Mega 2560 peut-être alimentée soit via la connexion USB (qui fournit 5V jusqu'à 500mA) ou à l'aide d'une alimentation externe. La source d'alimentation est sélectionnée automatiquement par la carte.

➢ L'alimentation externe (non-USB) peut être par un adapteur secteur (pouvant fournir typiquement de 3V à 12V sous 500mA) ou des piles (ou des accus). L'adaptateur secteur peut être connecté en branchant une prise 2.1mm positif au centre dans le connecteur jack de la carte. Les fils en provenance d'un bloc de piles ou d'accus peuvent être insérés dans les connecteurs des broches de la carte appelées Gnd (masse ou 0V) et Vin (Tension positive en entrée) du connecteur d'alimentation.

La carte peut fonctionner avec une alimentation externe de 6 à 20 volts. Cependant, si la carte est alimentée avec moins de 7V, la broche 5V pourrait fournir moins de 5V et la carte pourrait être instable. Si on utilise plus de 12V, le régulateur de tension de la carte pourrait chauffer et endommager la carte.

Les broches d'alimentation sont les suivantes :

➢ VIN. La tension d'entrée positive lorsque la carte Arduino est utilisée avec une source de tension externe (à distinguer du 5V de la connexion USB ou autre source 5V régulée). Vous pouvez alimenter la carte à l'aide de cette broche, ou, si l'alimentation est fournie par le jack d'alimentation, accéder à la tension d'alimentation sur cette broche.

➢ 5V. La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (pour info : les circuits électroniques numériques nécessitent une tension d'alimentation parfaitement stable dite "tension régulée" obtenue à l'aide d'un composant appelé un régulateur et qui est intégré à la carte Arduino). Le 5V régulé fourni par cette broche peut donc provenir soit de la tension d'alimentation VIN via le régulateur de la carte, ou bien de la connexion USB (qui fournit du 5V régulé) ou de tout autre source d'alimentation régulée.

➢ 3V3. Une alimentation de 3.3V fournie par le circuit intégré FTDI (circuit intégré faisant l'adaptation du signal entre le port USB de notre ordinateur et le port série de

l'ATmega) de la carte est disponible : ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5V). L'intensité maximale disponible sur cette broche est de 50mA.

3. Memoire

L'ATmega 2560 a 256Ko de mémoire FLASH pour stocker le programme (dont 8Ko également utilisés par le bootloader). L'ATmega 2560 a également 8 ko de mémoire SRAM (volatile) et 4Ko d'EEPROM (non volatile - mémoire qui peut être lue à l'aide de la librairie EEPROM).

4. Entrées et sorties numériques

Chacune des 54 broches numériques de la carte Mega peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions pinMode(),digitalWrite() et digitalRead() du langage Arduino. Ces broches fonctionnent en 5V. Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité et dispose d'une résistance interne de "rappel au plus" (pull-up) (déconnectée par défaut) de 20-50 KOhms. Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction digitalWrite(broche,HIGH).

De plus, certaines broches ont des fonctions spécialisées :

- **Communication Serie** : Port Serie Serial : 0 (RX) and 1 (TX) ; Port Serie Serial 1: 19 (RX) and 18 (TX); Port Serie Serial 2: 17 (RX) and 16 (TX); Port Serie Serial 3: 15 (RX) and 14 (TX). Utilisées pour recevoir (RX) et transmettre (TX) les données sérient de niveau TTL. Les broches 0 (RX) and 1 (TX) sont connectées aux broches correspondantes du circuit intégré ATmega8U2 programmé en convertisseur USB-vers-série de la carte, composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur.
- **Interruptions Externes**: Broches 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), et 21 (interrupt 2). Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur. Voir l'instruction attachInterrupt() pour plus de détails.
- **Impulsion PWM (largeur d'impulsion modulée)**: Broches 0 à 13. Fournissent une impulsion PWM 8-bits à l'aide de l'instruction analogWrite().

- **SPI (Interface Série Périphérique)**: Broches 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). Ces broches supportent la communication SPI (Interface Série Périphérique) disponible avec la librairie pour communication SPI.
- **I2C**: Broches 20 (SDA) et 21 (SCL). Supportent les communications de protocole I2C (ou interface TWI (Two Wire Interface - Interface "2 fils"), disponible en utilisant la librairie Wire/I^2C (ou TWI - Two-Wire interface - interface "2 fils") . Noter que ces broches n'ont pas le même emplacement que sur les cartes Uno, Duemilanove ou Diecimila.
- **LED**: Broche 13. Il y a une LED incluse dans la carte connectée à la broche 13. Lorsque la broche est au niveau HAUT, la LED est allumée, lorsque la broche est au niveau BAS, la LED est éteinte.

5. Broches analogiques :

La carte Mega 2560 dispose de 16 entrées analogiques, chacune pouvant fournir une mesure d'une résolution de 10 bits (càd sur 1024 niveaux soit de 0 à 1023) à l'aide de la très utile fonction analogRead() du langage Arduino. Par défaut, ces broches mesurent entre le 0V (valeur 0) et le 5V (valeur 1023), mais il est possible de modifier la référence supérieure de la plage de mesure en utilisant la broche AREF et l'instruction analogReference() du langage Arduino.

Notes :

- ✓ Les broches analogiques peuvent être utilisées en tant que broches :
- ✓ Il y a deux autres broches disponibles sur la carte :
  - **AREF** : Tension de référence pour les entrées analogiques (si différent du 5V). Utilisée avec l'instruction analogReference().
  - **Reset** : Mettre cette broche au niveau BAS entraîne la réinitialisation (le redémarrage) du microcontrôleur. Typiquement, cette broche est utilisée pour ajouter un bouton de réinitialisation sur le circuit qui bloque celui présent sur la carte.

6. Communication :

La carte Arduino Mega2560 communique facilement avec un ordinateur, une autre carte Arduino, ou avec d'autres microcontrôleurs. L'ATmega2560 dispose de quatre UARTs (Universal Asynchronous Receiver Transmitter ou émetteur-récepteur asynchrone universel en français) pour communication série de niveau TTL (5V) et qui est disponible sur les broches 0 (RX) et 1 (TX). Un circuit intégré ATmega8U2 sur la carte assure la connexion entre cette communication série de l'un des ports série de l'ATmega 2560 vers le port USB de l'ordinateur qui apparaît comme un port COM virtuel pour les logiciels de l'ordinateur. Le code utilisé pour programmer l'ATmega8U2 utilise le driver standard USB COM, et aucun autre driver externe n'est nécessaire.

Le logiciel Arduino inclut une fenêtre terminal série (ou moniteur série) sur l'ordinateur et qui permet d'envoyer des textes simples depuis et vers la carte Arduino. Les LEDs RX et TX sur la carte clignote lorsque les données sont transmises via le circuit intégré ATmega8U2 utilisé en convertisseur USB-vers-série et la connexion USB vers l'ordinateur (mais pas pour les communications série sur les broches 0 et 1).

L'ATmega2560 supporte également la communication par protocole I2C (ou interface TWI (Two Wire Interface - Interface "2 fils") et SPI :

- Le logiciel Arduino inclut la librairie Wire qui simplifie l'utilisation du bus I2C.
- Pour utiliser la communication SPI (Interface Série Périphérique), la librairie pour communication SPI est disponible.

7. Programmation

La carte Mega 2560 peut être programmée avec le logiciel Arduino. Il suffit de sélectionner "Arduino Mega" dans le menu **Tools** > **Board**. Pour plus de détails sur le langage Arduino.

Le microcontrôleur ATmega2560 présent sur la carte Arduino Mega2560 est livré avec un bootloader (petit programme de démarrage) préprogrammé qui vous permet de transférer le nouveau programme dans le microcontrôleur sans avoir à utiliser un matériel de programmation externe. Ce bootloader communique avec le microcontrôleur en utilisant le protocol original, STK500.

Nous pouvons bien sûr passer outre le bootloader et programmer le microcontrôleur via le connecteur ICSP (In-Circuit Serial Programming - "Programmation Série Dans le circuit" en français).

La source du code pour le circuit intégré ATmega8U2 est disponible. L'ATmega8U2 est chargé avec un bootloader DFU qui peut être activé en connectant le cavalier au dos de la carte et en réinitialisant le 8U2. Nous pouvons alors utiliser le logiciel FLIP de chez Atmel (Windows) ou le programmeur DFU (Mac OS X et Linux) pour charger le nouveau code. Ou bien nous pouvons utiliser le connecteur ICSP avec un programmateur externe (pour réécrire le bootloader DFU).

8. Rienialisation logiciel automatique :

Plutôt que de nécessiter un appui sur le bouton poussoir de réinitialisation avant un transfert de programme, la carte Arduino Mega 2560 a été conçue de telle façon qu'elle puisse être réinitialisée par un logiciel tournant sur l'ordinateur. Une des broches matérielles de contrôle du flux (DTR) du circuit intégré ATmega8U2 est connecté à la ligne de réinitialisation de l'ATmega2560 via un condensateur de 100 nanofarads. Lorsque cette broche est mise au niveau BAS, la broche de réinitialisation s'abaisse suffisamment longtemps pour réinitialiser le microcontrôleur. Le logiciel Arduino utilise cette possibilité pour nous permettre de transférer notre programme dans la carte par un simple clic sur le bouton de transfert de la barre de boutons de l'environnement Arduino. Cela signifie que le bootloader peut avoir un temps mort plus court, la mise au niveau bas de la broche DTR étant bien coordonnée avec le début du transfert du programme.

Ce fonctionnement a d'autres implications. Quand la carte Mega2560 est connectée à un ordinateur fonctionnant soit sous Mac Os X ou Linux, la carte se réinitialise à chaque fois qu'une connexion se fait entre elle et le logiciel (via le port USB). Pendant la demi-seconde suivante, le bootloader s'exécute sur la carte Mega2560. Puisqu'il est programmé pour ignorer les données mal formatées (càd toute donnée en plus du transfert d'un nouveau programme), il interceptera les quelques premiers octets de donnée envoyé à la carte après qu'une connexion soit établie. Si un programme s'exécutant sur la carte reçoit une configuration initiale ou d'autres données quand il démarre, nous assrons que le logiciel avec lequel il communique attend une seconde après l'ouverture de la connexion avant d'envoyer ces données.

La carte Mega2560 comporte une piste (du circuit imprimé) que nous pouvons couper pour désactiver la réinitialisation automatique. Les bords de chaque côté de la piste peuvent solidarisés ensemble pour la réactiver. Nous pouvons également désactiver la réinitialisation automatique en connectant une résistance de 110 Ohms entre le +5V et la ligne de réinitialisation (reset).

9. Protection du port USB contre la surcharge en intensité :

La carte Arduino Mega2560 intègre un polyfusible réinitialisable qui protège le port USB de notre ordinateur contre les surcharges en intensité (le port USB est généralement limité à 500mA en intensité). Bien que la plupart des ordinateurs aient leur propre protection interne, le fusible de la carte fournit une couche supplémentaire de protection. Si plus de 500mA sont appliqués au port USB, le fusible de la carte coupera automatiquement la connexion jusqu'à ce que le court-circuit ou la surcharge soit stoppé.

10. Caractéristique mécanique :

Les longueurs et largeurs maximales du circuit imprimé de la carte Mega2560 sont respectivement 10.16 cm et 5.33 cm, avec le connecteur USB et le connecteur d'alimentation Jack s'étendant au-delà des dimensions de la carte. Plusieurs trous de vis permettent à la carte d'être fixée sur une surface ou dans un boîtier. Noter que la distance entre les broches 7 et 8 est de 0.16 pouces, et non un multiple des 0.1 pouces séparant les autres broches.

La carte Mega2560 a été conçue pour être compatible avec la plupart des circuits imprimés des modules prévus pour les cartes Uno, Diecimila ou Duelmilanove. Les broches numériques 0 à 13 (et les broches adjacentes AREF et GND), les entrées analogiques 0 à 5, le connecteur de tension et le connecteur ICSP ont des emplacements identiques pour toutes ces cartes. De la même façon, l'UART principale (port série) est connectée sur les mêmes broches (0 et 1), ainsi que les interruptions externes 0 et 1 (respectivement sur les broches 2 et 3). La communication SPI est disponible via le connecteur ICSP à la fois pour les cartes Mega2560 et Uno, Duemilanove, Diecimila. Noter cependant que la connexion I2C n'est pas disponible sur les mêmes broches pour la Mega (broches 20 et 21) que pour les cartes Uno / Duemilanove / Diecimila (broches analogiques 4 et 5) [10].

2.4.2.2 RAMP 1.4

RAMPS version 1.4 est la carte "shields" qui se connecte à la carte Arduino 2560. C'est la carte de puissance et d'interface qui à l'aide des pilotes(driver A4988) enfichable vas commander les moteur pas à pas de notre imprimante 3D sur les axe X,Y,Z, la commande de rotation du fil d'extrusion, les résistances de chauffe de notre tête d'impression, du lit chauffant, des capteur de température, ventilateur, et même optionnellement l'affichage LCD et lecteur de carte SD. Cette interface peut être pilotée facilement par le programme Marlin telechargable sur la catre controleur Arduino 2560.



Figure 2.8: RAMP 1.4

2.4.2.3 Driver moteur pas à pas

Driver moteur pas à pas A4988 est une carte porteuse ou carte de dérivation pour A4988 DMOS d'Allegro Microstepping pilote avec le traducteur et la protection contre les surintensités. Driver pas à pas nous permet de contrôler un moteur pas à pas bipolaire jusqu'à 2 A par bobine courant de sortie. La figure (2.9) illustre la photo du driver A4988 :

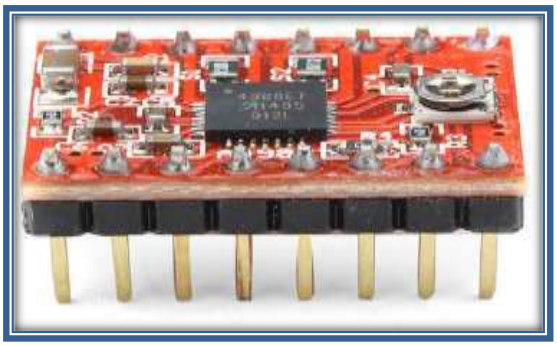


Figure 2.9: Le driver A4988

Caractéristiques :

- Utilisation simple de l'interface de contrôle de pas et de direction ;
- Contrôle saccadé intelligent qui sélectionne automatiquement le régime adéquat (haut ou bas régime) ;
- Cinq résolutions de micropas différentes : Pas complet, demi-pas, quart de pas, huitième de pas et seizième de pas ;
- Arrêt thermique de surchauffe, verrouillage de sous-tension, protection courant transversal ;
- Protection contre les courts-circuits et les surcharges.

La figure (2.10) montre la structure du driver moteur pas à pas A4988 et son fonctionnement.

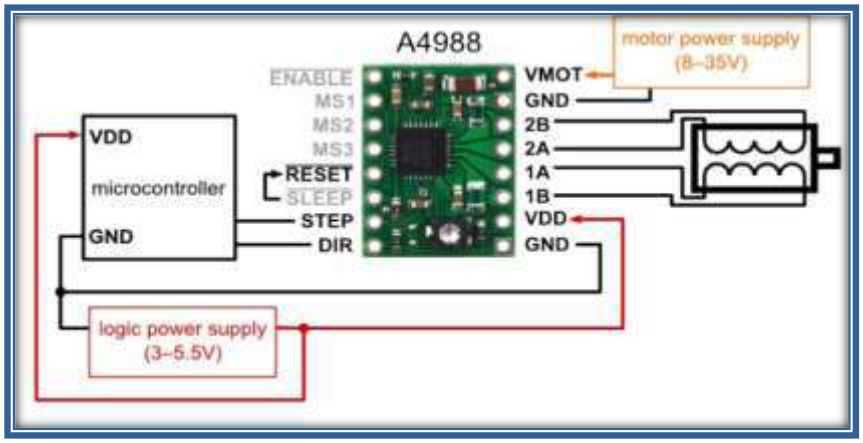


Figure 2.10: La structure de driver A4988.

2.4.2.4 LCD

Ce contrôleur intelligent contient un lecte connecteur de cartes SD, une molette et un x4 écran LCD de ligne de 20 caractères. Nous pouvons facilement le connecter à notre conceil rampes.

Apres avoir connecté ce panneau pour nos rampes nous n'avons pas besoin du PC, le Smart Controller fournit de l'énergie pour notre carte SD. D'autre plus toutes les actions comme l'étalonnage, des haches mouvements peuvent être effectués en utilisant simplement la molette sur le Smart Controller (figure 2.11).



Figure 2.11: Affecheur LCD

2.5 Montage de la partie électrique

1. Placer les drivers (StepStick)

Nous plaçons les 4 x StepStick sur la RAMP. L'image (2.12) contient les points GND et DIR des différents drivers.

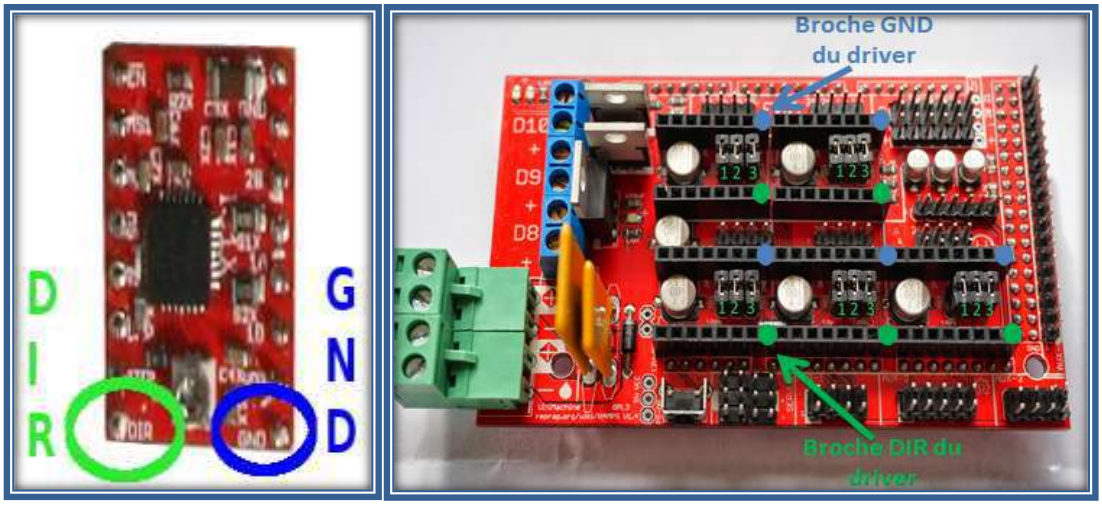


Figure 2.12: Les points GND et DIR du driver.

Et la figure (2.13) montre les drivers connectés à la carte RAMPS 1.4.



Figure 2.13: Placement des drivers à la RAMPS 1.4.

2. **Placer la carte Mega 2560**

La figure (2.14) montre les deux faces de l'arduino avec le RAMP 1.4.

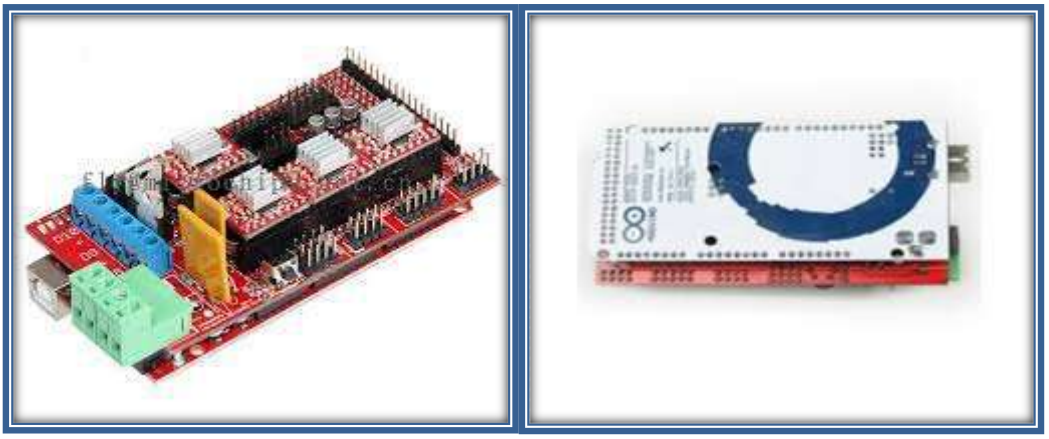


Figure 2.14: Deux faces de l'Arduino avec le RAMP 1.4.

3. **Connecter les End-Stop aux composants fin de course**

La figure (2.15) montre les connecteurs End Stop et les différentes positions Min et Max.

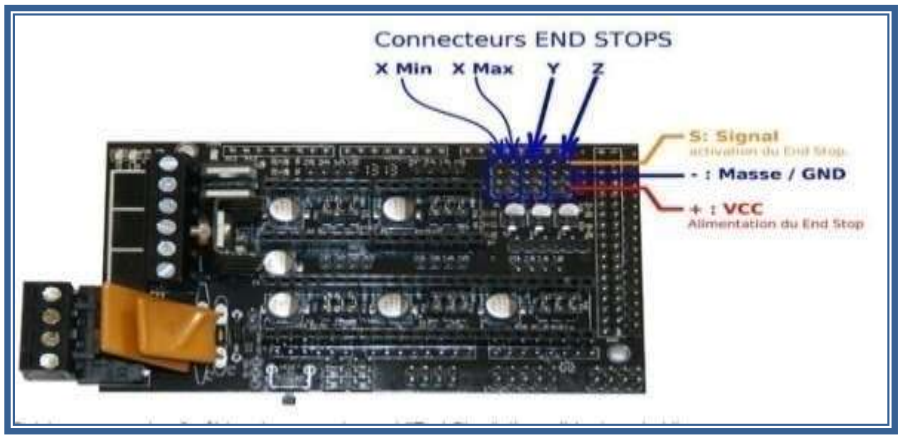


Figure 2.15: Connecteurs End Stop et les différentes positions Min et Max.

4. Brancher les thermistances

Commencer par le branchement des thermistances ; indiqué par les flèches vertes sur la figure (2.16).

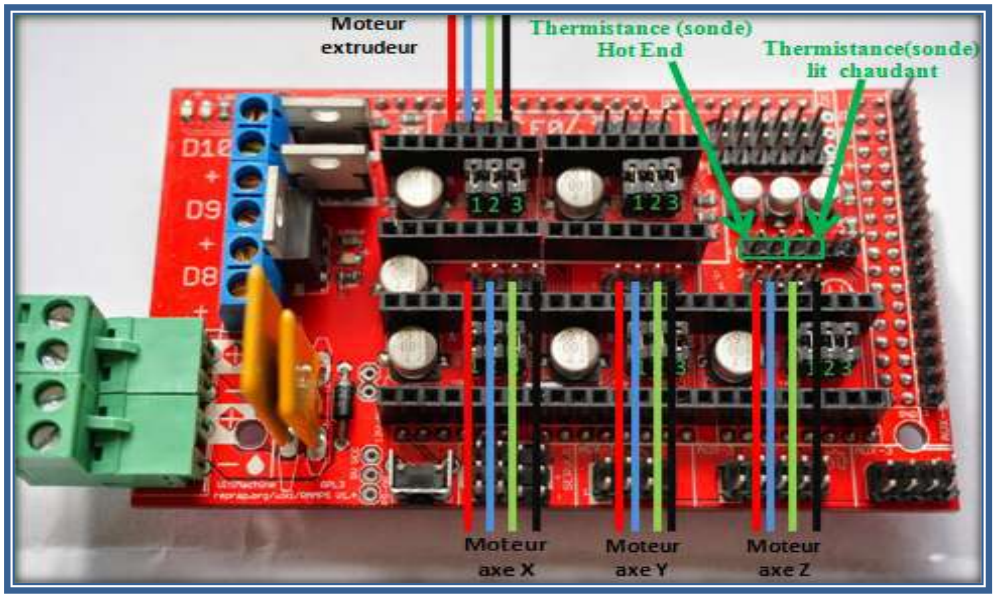


Figure 2.16: le branchement des thermistances.

5. Brancher la partie chauffante

- Du lit chauffant sur D8
- Du Hot-end sur D10

Sur le borner du de la RAMPS comme indiqué la figure (2.17).

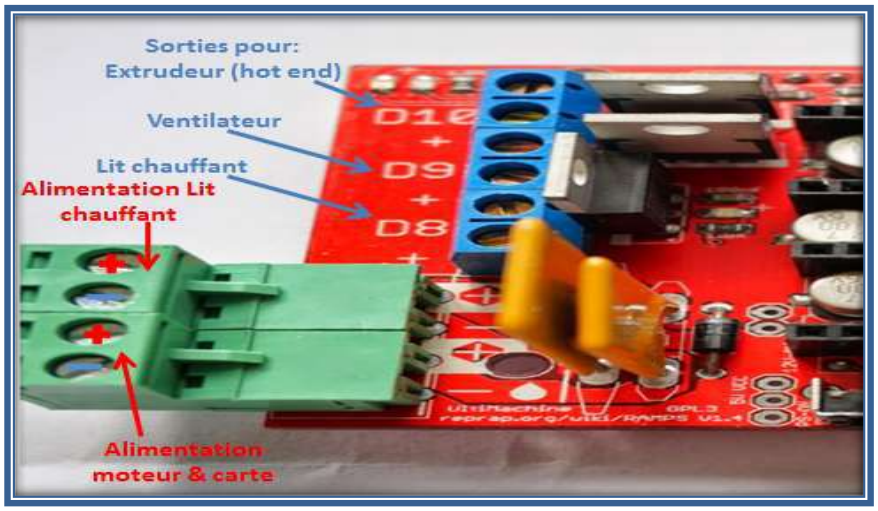


Figure 2.17: Aimantation lit chauffant, moteur et la carte ainsi les sorties d'extrudeur, ventilateur et lit chauffant.

La figure (2.18) montre tous les branchemants possible sur le RAMP

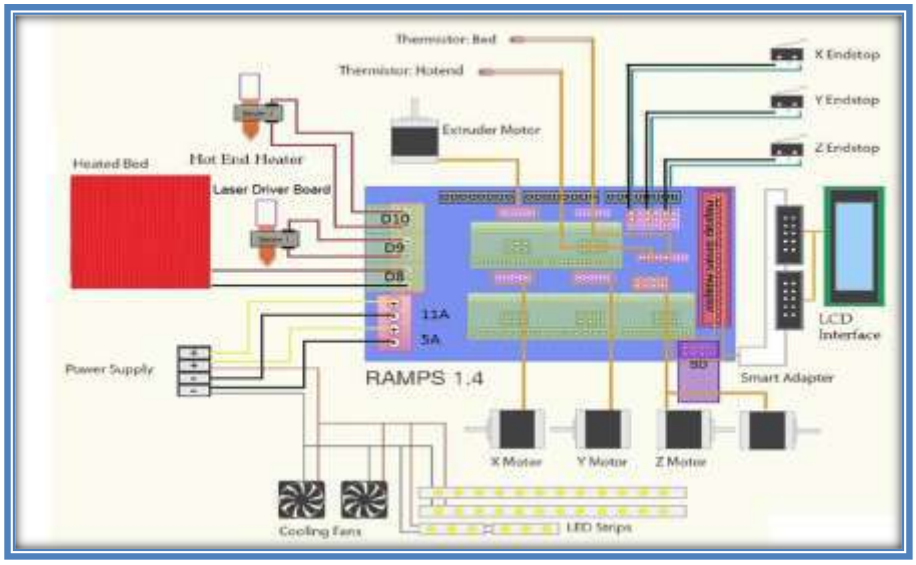


Figure 2.18: Les branchements sur le RAMP

2.6 Conclusion

Dans ce chapitre, nous avons présenté toutes les parties nécessaires à notre réalisation en passant par la structure mécanique et la partie électronique. Puis, après le dimensionnement de notre imprimante 3D delta, nous avons fabriqué la partie mécanique. Enfin, les détails de la partie électrique de l'imprimante 3D delta sont donnés ; où les circuits d'alimentation, de commande et de communication sont présentés.

3.1 Introduction

Dans une impression 3D, nous avons besoin qu'un logiciel de tranchage s'occupe de découper en tranches la pièce imprimé. C'est la succession d'impressions de chacune de ces couches et leur empilement qui formeront notre objet. Dans ce chapitre on utilise pour le tranchage le logiciel Repetier-host.

3.2 Chaine d'acquisition

La chaine d'acquisition est indiqué à la figure suivante :

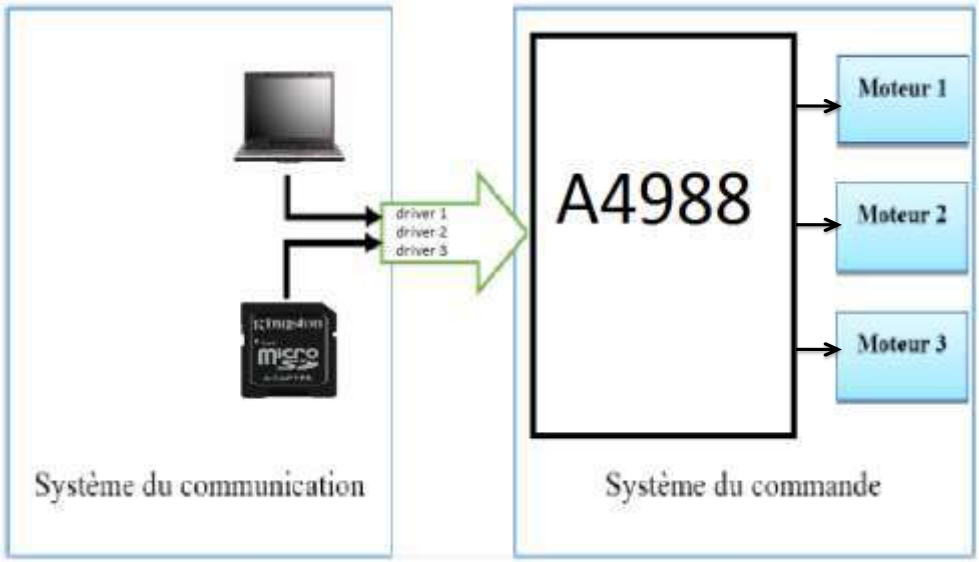


Figure 3.1 : Chaine d'acquisition.

3.3 Etape de commande

Les étapes de commande sont :

A. Instalation de l'arduino MEGA 2560

Nous installons l'arduino Software dans l'ordinateur en exécutant le fichier.Ensuite une fois l'instalation terminé, nous connectons l'ordinateur au port USB de l'arduino. En fin nous verifions le numéro de port COM attribué à l'arduino Mega 2560 comme indiqué sur la figure (3.2).



Figure 3.2 : Installation de l'Arduino Software à l'ordinateur

B. Chargement du firmware

Le chargement du firmware est indiqué à la figure (3.3) :

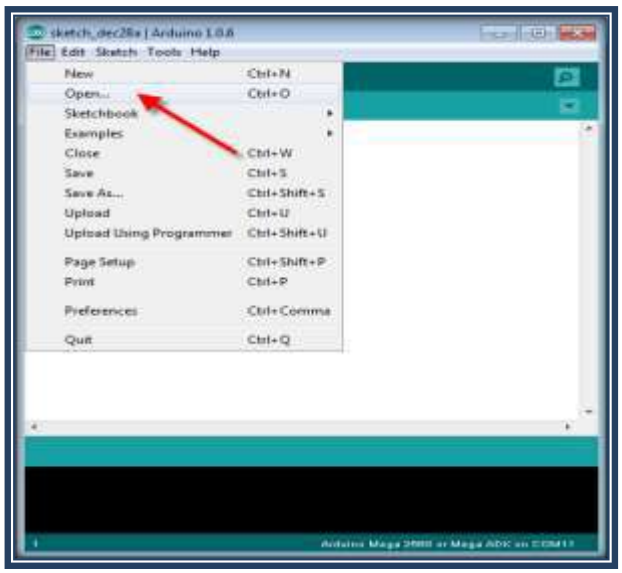


Figure 3.3: Le chargement du firmware

Nous accédons au dossier Marlin où les fichiers du firmware sont stockés comme indiqué au figure (3.4).

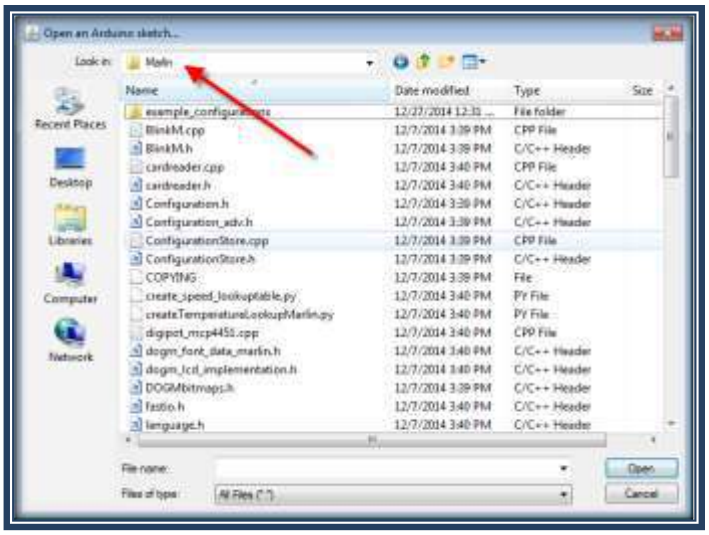


Figure 3.4: Fichiers du firmware.

- Localisons le fichier Marlin.ino
- Cliquons une fois sur Marlin.ino (figure 3.5)
- Cliquons sur le bouton Ouvrir

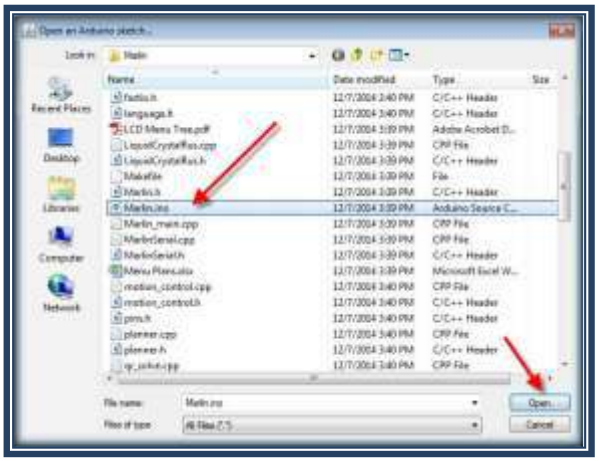


Figure 3.5: Programme de Marlin.

- Cliquons sur Configuration.h.

Dans ce fichier Configuration.h, il y a des instructions destiné à l'ordinateur et d'autres qui sont réservé aux humains pour une meilleure compréhension figure(3.6). Pour éviter que l'ordinateur tente de déchiffrer les explications destinés aux humains il faut utiliser les symboles de commentaire "//". La double barre oblique délimite sur chaque ligne ce qui est du language machine et les commentaires. Dans le cas qui nous intéresse, les commentaires serviront aussi à activer certaines options.

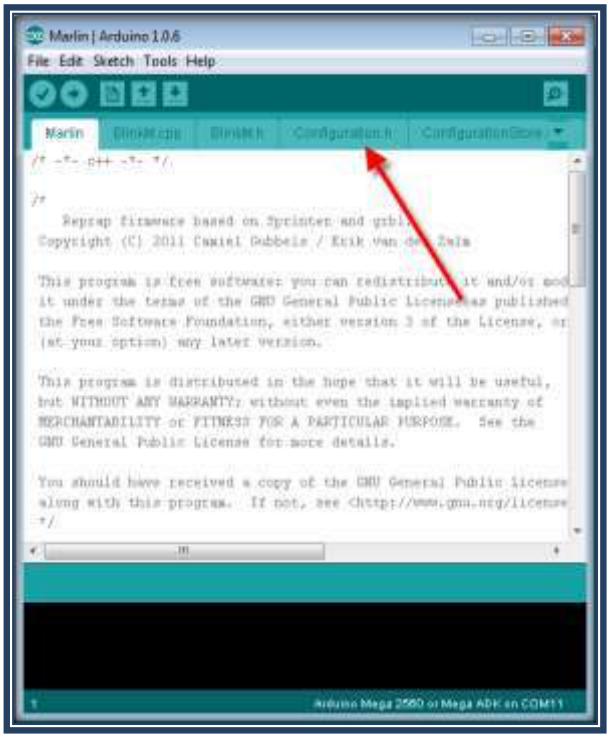


Figure 3.6 : Règles générales des commentaires.

C. Sélection de la vitesse de communication (Arduino baud rate):

La vitesse de transfert entre la carte mère de l'imprimante 3D et l'ordinateur est définie en baud:

#define BAUDRATE 250000

//#define BAUDRATE 115200

La vitesse doit être la même pour le contrôleur, pour le matériel USB et pour le logiciel hôte, ce qui peut parfois limiter la vitesse. Deux vitesses sont plus couramment utilisées: 250000 et 115200 bauds. Si l'une ne fonctionne pas, essayez avec l'autre(figure 3.7).

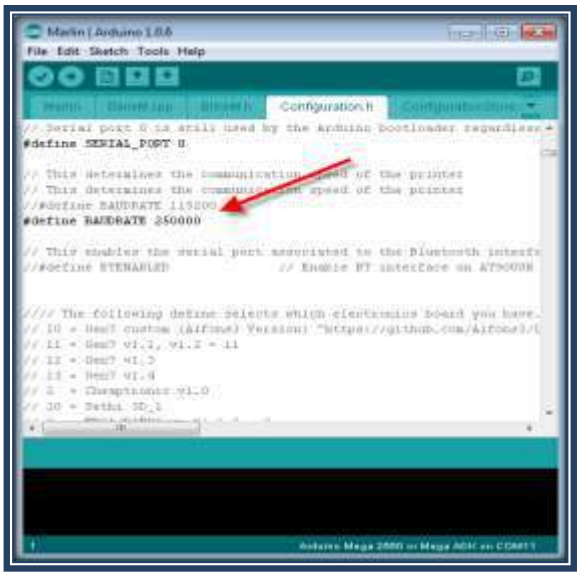


Figure 3.7: Vitesse de communication (Arduino baud rate).

D. Sélection de la carte mère (MOTHERBOARD):

Au dessus de la ligne #define MOTHERBOARD se trouve une liste des différentes cartes mères disponibles. Il suffit de trouver notre carte et sa configuration s'il y a lieu pour utiliser le bon numéro. Par exemple la carte RAMPS 1.4 (Power outputs: Extruder, Fan, Bed) sera configurée avec le numéro 33(figure 3.8).

#define MOTHERBOARD 33

Utilisez celui-ci même si vous n'utilisez pas de ventilateur (fan) ou de plaque chauffante (bed).

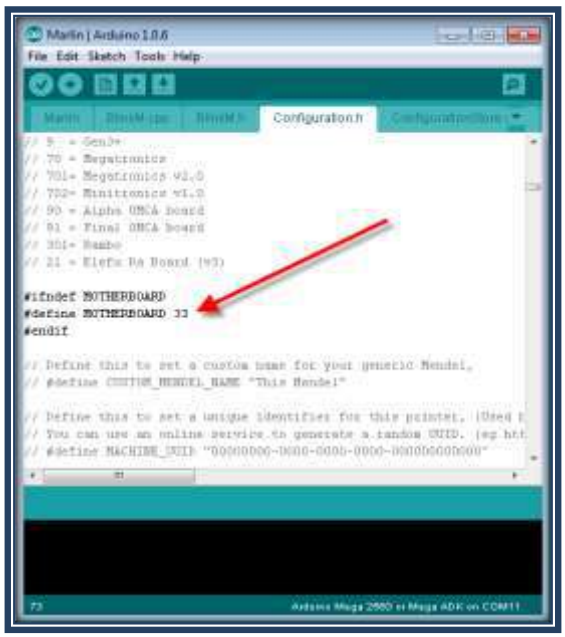


Figure 3.8**:** La carte mère (Motherboard).

E. Nom de l'imprimante:

En connectant un écran LCD à l'imprimante, on peut afficher le nom de notre imprimante.

#define CUSTOM_MENDEL_NAME "Solid Utopia"

F. Nombre d'extrudeurs :

Indiquez le nombre d'extrudeurs que vous pensez utiliser(figure3.9).

#define EXTRUDERS 1



Figure 3.9: Nombre d'extrudeurs.

G. Paramètres thermiques :

Les parametre termiques sont :

- **Choix de la sonde de température :**

Parmi la liste énumérée dans le code, ci-dessus, il faut choisir le thermistor adéquad. Si le thermistor est inconnu, il est possible d'obtenir une erreur de lecture tout en permettant le contrôle de température.

#define TEMP_SENSOR_0 7
#define TEMP_SENSOR_1 0
#define TEMP_SENSOR_2 0
#define TEMP_SENSOR_BED 1

TEMP_SENSOR_0 est utilisé par la première tête d'impression alors que TEMP_SENSOR_BED est connecté à la plaque chauffante. Utilisons le 0 pour désactiver une sonde.



Figure 3.10: Choix de la sonde de la température.

- **Température minimum :**

Définissons la température minimum d'utilisation supérieure à 0°C. La raison étant qu'un capteur de température déconnecté indiquera une température de 0°C. Cette sécurité empêchera le contrôleur de chauffer les têtes d'impressions et la plaque chauffante au maximum de puissance indéfiniment. C'est un paramètre important qui est définit à 5°C par défaut.

#define HEATER_0_MINTEMP 5
#define HEATER_1_MINTEMP 5
#define HEATER_2_MINTEMP 5
#define BED_MINTEMP 5

- **Température maximum :**

Tout dépend des têtes d'impressions choisies, certaines résiste mieux à la température que d'autres. Par exemple un tube de PTFE (teflon) est souvent utilisé dans les têtes d'impression pour guider le filament dans la zone chaude, mais ce matériau résiste à des températures jusqu'à 240°C environ. Il est préférable de laisser un peu de marge ici puisqu'un contrôleur peut dépasser légèrement la température cible avant de la maintenir.

```
#define HEATER_0_MAXTEMP 235
#define HEATER_1_MAXTEMP 235
#define HEATER_2_MAXTEMP 235
#define BED_MAXTEMP 120
```

- **Contrôleur de température PID :**

Ceci est une option avancée, mais pour le moment l'imprimante peut fonctionner sans toucher à cette option.

- **Empêcher les extrusions dangereuses:**

Ces paramètres sont définis par défaut. Pour une raison de sécurité nous voulons éviter d'extruder un matériel trop froid et solide. Comme il est impossible d'extruder un matériau solide, l'imprimante pourrait se briser en tentant de le faire.

```
#define PREVENT_DANGEROUS_EXTRUDE
```

En cas d'erreur dans les instructions de mouvement de l'imprimante tel que la position d'une virgule ou d'un point, il serait pratique d'empêcher une extrusion trop longue.

```
#define PREVENT_LENGTHY_EXTRUDE
```

Ici nous pouvons fixer la température minimum d'extrusion, au dessus du point de fusion du PLA et de l'ABS.

```
#define EXTRUDE_MINTEMP 170
```

3.4 Paramètres mécaniques

Les parametres mécaniques sont

- **Interrupteurs de fin de course :**

Les interrupteurs de fin de course sont destinés à être déclenchés lorsqu'un axe atteint sa limite. En d'autres termes il prévient l'imprimante de sortir de son propre bâtît. Les interrupteurs de fin de course sont utilisés aussi comme position de référence. Lorsqu'on demande à l'imprimante de trouver son repère, elle déplacera chaque axe un par un dans la direction définie jusqu'à ce que l'interrupteur soit déclenché.

- **Choix du circuit d'interrupteur**

Une résistance de rappel est nécessaire entre les bornes de source et le retour à la masse sinon on risque de griller le circuit électronique avec un court-circuit.

Une résistance de rappel est déjà intégrée dans le micro-contrôleur et il est possible de l'activer par le firmware. Il est nécessaire de choisir cette option pour connecter un interrupteur mécanique directement entre le signal et le retour à la masse. Pour tout autre type d'interrupteur (optique, magnétique, mécanique) il est recommandé d'étudier le circuit afin d'éviter les courts-circuits. Laisser la ligne sans commentaire // pour activer les résistances de rappel sur chaque interrupteur.

```
#define ENDSTOPPULLUPS
```

Dans le cas contraire, Marlin permet de configurer chaque interrupteur individuellement. Il est possible d'utiliser différents types d'interrupteurs sur la même imprimante (optique, magnétique, mécanique, etc.).

```
#ifndef ENDSTOPPULLUPS
  #define ENDSTOPPULLUP_XMAX
  #define ENDSTOPPULLUP_YMAX
//  #define ENDSTOPPULLUP_ZMAX
  #define ENDSTOPPULLUP_XMIN
  #define ENDSTOPPULLUP_YMIN
//  #define ENDSTOPPULLUP_ZMIN
#endif
```

- **Inversion de la logique des interrupteurs :**

Certains interrupteurs laissent normalement passer le courant et l'interrompent lorsqu'ils sont déclenchés (NC). D'autres ne laisse passer le courant que lorsqu'ils sont déclenchés (NO). Certains offrent les deux types de comportement et il suffit de souder les bons connecteurs (NC ou NO). Un circuit (NC, normally closed) laisse passer le courant et un circuit ouvert (No, normally open) ne le laisse pas passer au repos. L'avantage de laisser passer le courant en situation normale est que lorsque les fils sont coupés ou que l'interrupteur est brisé, l'imprimante interprétera cette erreur comme un signal de fin de course et ne bougera pas plus loin.

NO = true

NC = false

```
const bool X_MIN_ENDSTOP_INVERTING = false;
const bool Y_MIN_ENDSTOP_INVERTING = false;
const bool Z_MIN_ENDSTOP_INVERTING = false;
```

const bool X_MAX_ENDSTOP_INVERTING = false;

const bool Y_MAX_ENDSTOP_INVERTING = false;

const bool Z_MAX_ENDSTOP_INVERTING = false;

La figure (3.11) montre l'inversion de la logique des interrupteurs.

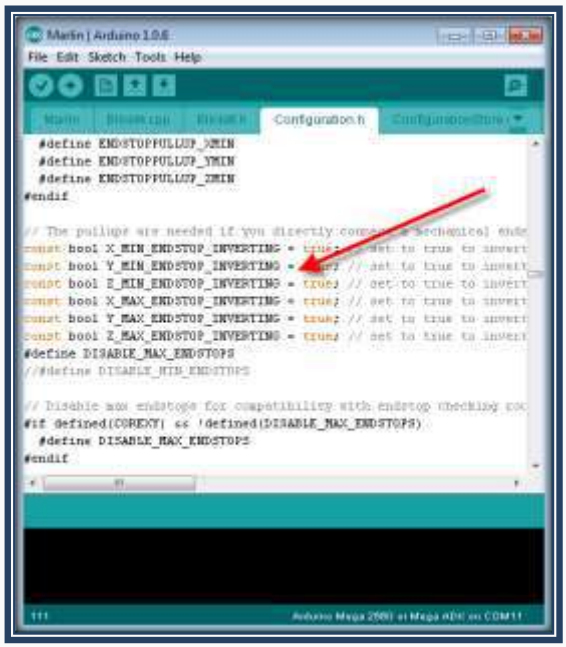


Figure 3.11: Inversion de la logique des interrupteurs.

- **Utiliser 3 ou 6 interrupteurs.**

Dans des circonstances normales, la position prévue par l'imprimante serait toujours adéquate. Dans ce cas, un interrupteur de fin de course serait suffisant pour déterminer la position de référence (origine de l'imprimante). Mais lorsqu'un incident survient et empêche le moteur de suivre la commande (collision, manque de puissance, pas manquée) la position prévue n'est plus la position réelle et il est possible en théorie que l'imprimante commande un mouvement au-delà de la fin des axes. L'utilisation d'interrupteurs de fin de course au début et à la fin de chaque axe empêche ce genre de comportement.

Enlever les symboles de commentaires // sur les lignes suivantes pour désactiver les interrupteurs maximums et minimums.

//#define DISABLE_MAX_ENDSTOPS

//#define DISABLE_MIN_ENDSTOPS

- **Inverser la direction des moteurs**

La meilleure façon de savoir si les moteurs sont branchés correctement ou à l'envers est de les essayer. Pour ce faire il est préférable d'attendre que la configuration soit complétée et que le firmware soit envoyée au contrôleur. Il faut utiliser les instruction suivante comme indique la figure (3.12).

#define INVERT_X_DIR false
#define INVERT_Y_DIR true
#define INVERT_Z_DIR true
#define INVERT_E0_DIR true
#define INVERT_E1_DIR true
#define INVERT_E2_DIR false

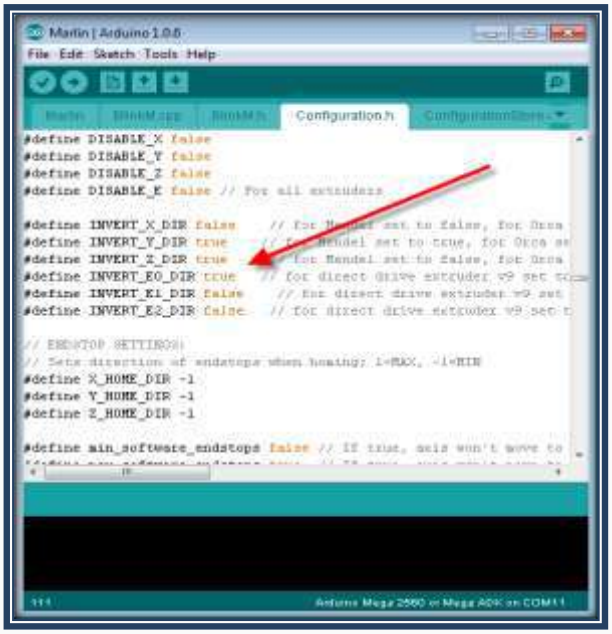


Figure 3.12: Inversion de direction des moteurs.

- **Définir le point de repère :**

La position de repère est le point vers lequel l'imprimante ira toucher les interrupteurs pour trouver sa position réelle. Elle sera ensuite "aveugle" et elle calculera sa position par rapport à ce point.

Il faut penser que l'imprimante puisse rejoindre sa position de référence à partir de n'importe quel emplacement possible. Comme les interrupteurs de fin de course se situent aux extrémités, il reste à définir dans quelle direction l'imprimante devra se déplacer pour atteindre les trois point que l'on aura choisi.

✓ -1 pour un déplacement en direction négative et 1 en direction positive.

```
#define X_HOME_DIR -1
#define Y_HOME_DIR -1
#define Z_HOME_DIR 1
```

À noter que la position de repère peut être différente du centre du plan cartésien.

L'axe Z est celui de la surface d'impression et elle sera placée très près de la buse. Il faut que le signal d'interruption soit très précis et très rapide pour signaler à l'imprimante le moment d'arrêter le mouvement sans risquer de collision. Dans certains cas, il est préférable de prendre la référence au maximum de la course en Z pour ramener ensuite la surface près de la buse lorsque l'imprimante sait où s'arrêter.

- **Volume d'impression :**

C'est ici que nous pouvons dire au contrôleur la plage de déplacement maximum pour chaque axe en partant du point de référence. Par exemple, si l'imprimante permet d'imprimer un volume de 200x200x200mm on dira que la position maximum en X est de 200 et le minimum est 0.

✓ À noter que l'imprimante n'as pas encore été calibrée et on ne sait pas si elle se déplacera bien de 200mm lorsqu'on lui demandera ou bien si elle dépassera légèrement. Pour éviter tout dépassement ou collision, il est donc préférable de réduire légèrement la distance maximum pour le moment. Nous pourrons y revenir après la calibration.

```
#define X_MAX_POS 190
#define X_MIN_POS 0
#define Y_MAX_POS 190
#define Y_MIN_POS 0
#define Z_MAX_POS 190
#define Z_MIN_POS 0
```

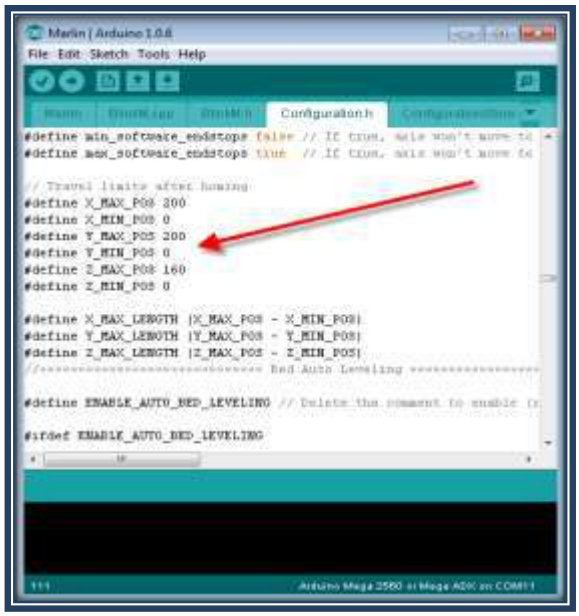


Figure 3.13: Limites de l'impression.

3.5 Paramètres de mouvements

Les paramètres de mouvement sont :

- **Définir le nombre de mouvements**

Il s'agit de définir combien de mouvements l'imprimante peut contrôler. Il s'agit du nombre d'axes (3) en plus du nombre d'extrudeurs.

#define NUM_AXIS 4

À noter que lorsqu'on utilise plus de 1 extrudeur (défaut) il faut ajouter une virgule pour chaque variable contenant des mouvements dans les lignes qui suivent.

Ex: variable {X, Y, Z, E1, E2, E3, etc.}

- **Vitesse d'avance vers le point de repère**

C'est la vitesse de l'imprimante lorsque qu'elle se déplace vers sont point de repère. Attention aux unités. Plusieurs vitesses ou accélérations dans marlin sont exprimé en [mm/s] ou [mm/s2], mais il s'agit ici de [mm/min].

#define HOMING_FEEDRATE {50*60, 50*60, 4*60, 0}

- **Nombre d'impulsions par unité**

Les moteurs pas à pas se déplacent par impulsions, les moteurs couramment utilisés sont des NEMA 17 ayant une résolution de 200 impulsions par rotation. Les contrôleurs envoient aussi des impulsions partielles (microsteps) pour augmenter la résolution de contrôle. Tout dépendamment du mécanisme, le contrôleur doit savoir le nombre d'impulsions à envoyer aux moteurs pour effectuer un déplacement d'un mm sur un axe donné.

La façon de le calculer est de diviser le nombre d'impulsions par le nombre de mm pour la même distance parcourue, dans ce cas une rotation complète du moteur. Le nombre d'impulsions total vu du firmware est le nombre d'impulsions par révolution du moteur multiplié par le nombre de microsteps par impulsion du contrôleur.

- **Décalage de plusieurs têtes d'impression (optionnel)**

Dans le cas où l'on voudrait utiliser plus d'une tête d'impression, il faudrait instruire le firmware de la distance à déplacer pour bien positionner la seconde buse. Normalement la distance entre les buses est mesurée à partir de la première. On a donc les coordonnées 0, 0 pour la première. Dans notre exemple, la seconde buse est décalée de 25mm en X et reste aligné sur la première en Y. Poursuivre de la même manière pour les buses successives.

```
#define EXTRUDER_OFFSET_X {0.0, 25}
#define EXTRUDER_OFFSET_Y {0.0, 0}
```

Il est aussi possible d'effectuer le changement de position lors du changement de buse à partir du G-code. Cependant en laissant le firmware prendre en charge cette opération on a pas à définir le décalage ailleur et le G-code est un peu plus universel.

3.6 Compilation

Dans l'application Arduino IDE, sauvegarder les modifications faites au fichier Configuration.h en cliquant sur save.

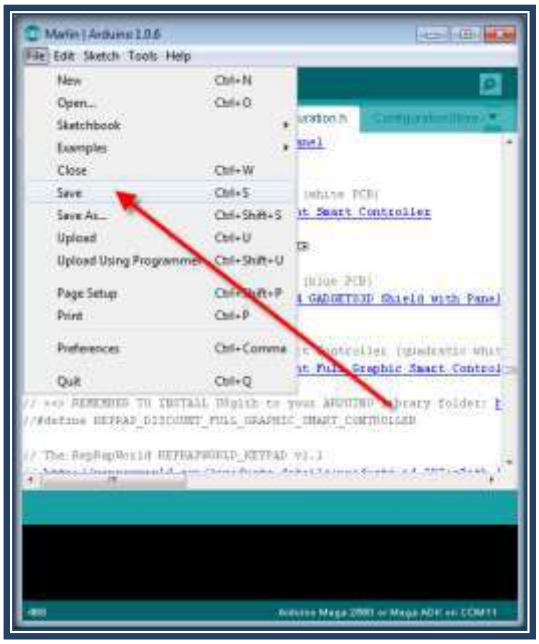


Figure 3.14: Enregistrement les modifications.

Si tout est en ordre, la vérification devrait retourner “Done compiling” sinon nous devrions avoir un ou plusieurs messages d’erreur. Habituellement la ligne de l’erreur est indiquée et il s’agit souvent d’erreur de frappe ou un manque de délimitation des commentaires //.

Une fois que la vérification est passée, il est temps de brancher la carte mère de l’imprimante au port USB de l’ordinateur est :

1. Sélectionner port correspondant à l’imprimante 3D.
2. Téléchargez le code compilé sur la carte mère (upload)
3. Positionner manuellement l’imprimante au centre de chaque axe. Il faut qu’elle puisse bouger dans un sens ou dans l’autre.
4. Allumer l’alimentation.
5. Sur l’ordinateur démarrer un logiciel tel que Pronterface ou Repetier pour piloter l’imprimante 3d.
6. Appuyer sur le bouton connecter
7. Garder votre doigt prêt à intervenir sur le bouton reset de la carte mère de l’imprimante. (juste au cas…)
8. Envoyer la commande pour déplacer l’axe X d’une petite distance tel que +1 ou +10mm
9. Si l’imprimante se déplace dans l’autre direction, il faudra inverser le sens du moteur.
10. Répéter pour chaque axe.

Pour tester le mouvement des extrudeurs, il faut passer par dessus la protection. La commande M302 permet de faire tourner l'extrudeur même si la buse n'est pas à la température d'extrusion. Il faut être certain d'effectuer le test sans filament.

Envoyer la commande pour extruder 10mm de matériel et vérifier le sens de rotation du moteur.

#define INVERT_X_DIR false

#define INVERT_Y_DIR true

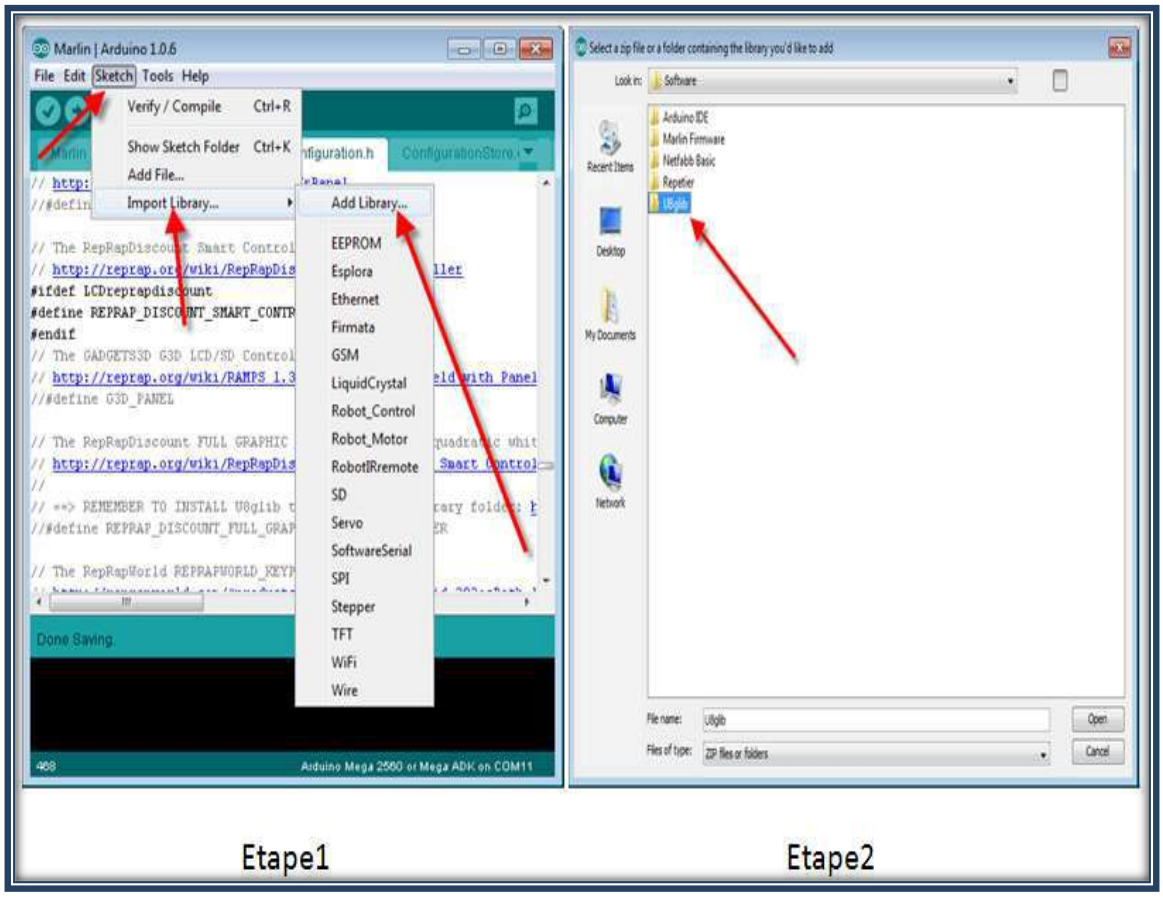
#define INVERT_Z_DIR true

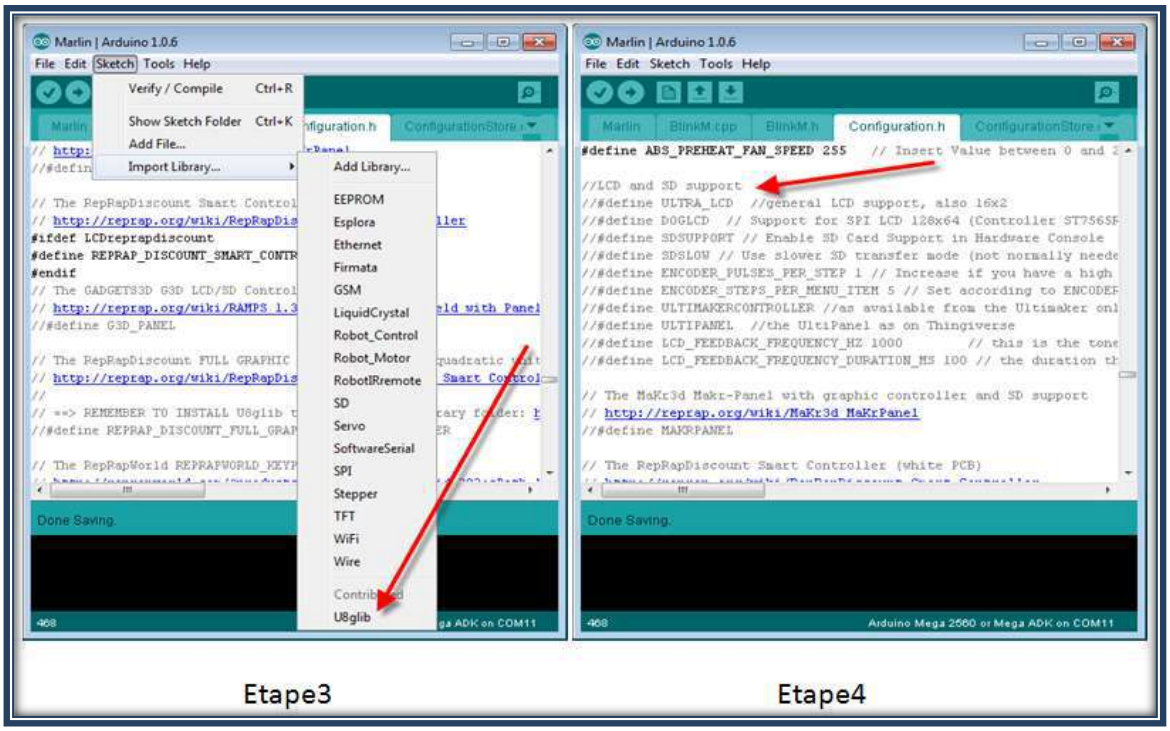
#define INVERT_E0_DIR true

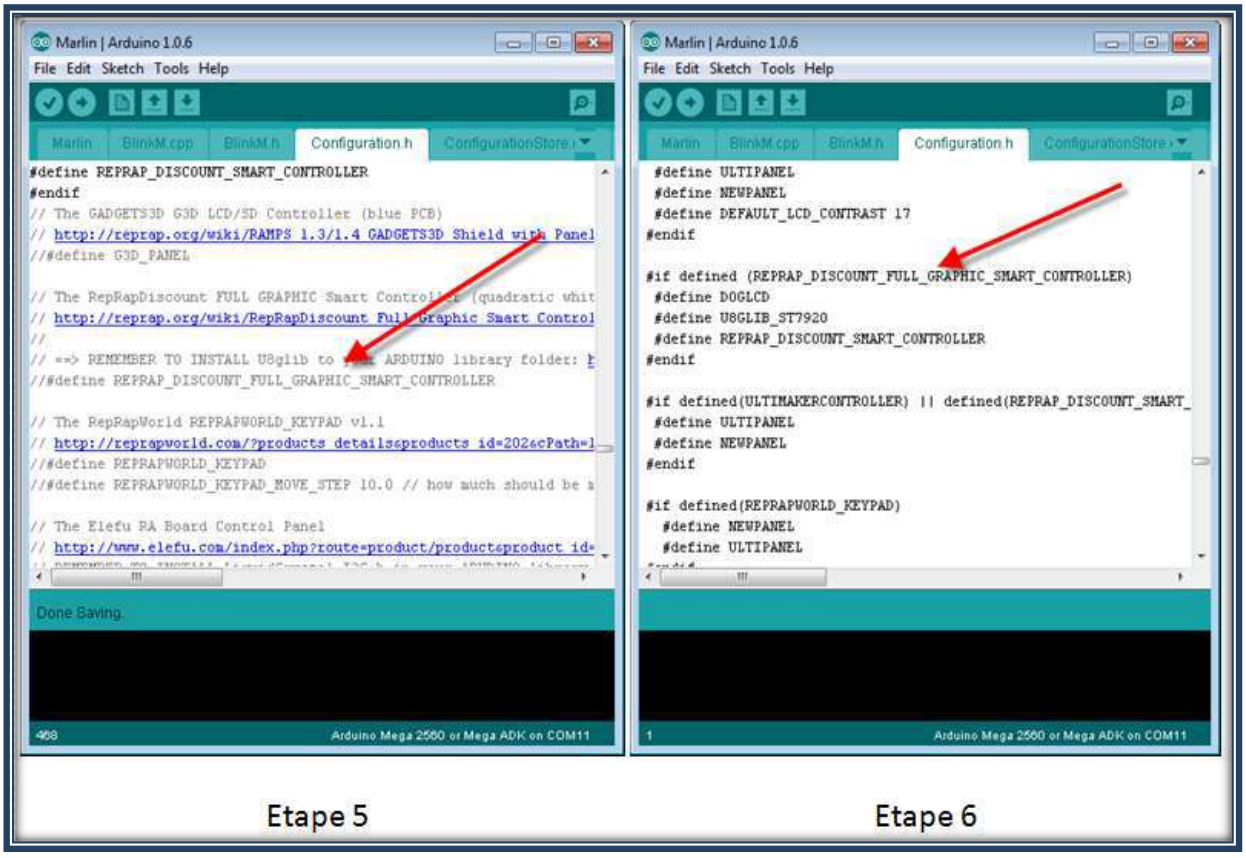
#define INVERT_E1_DIR true

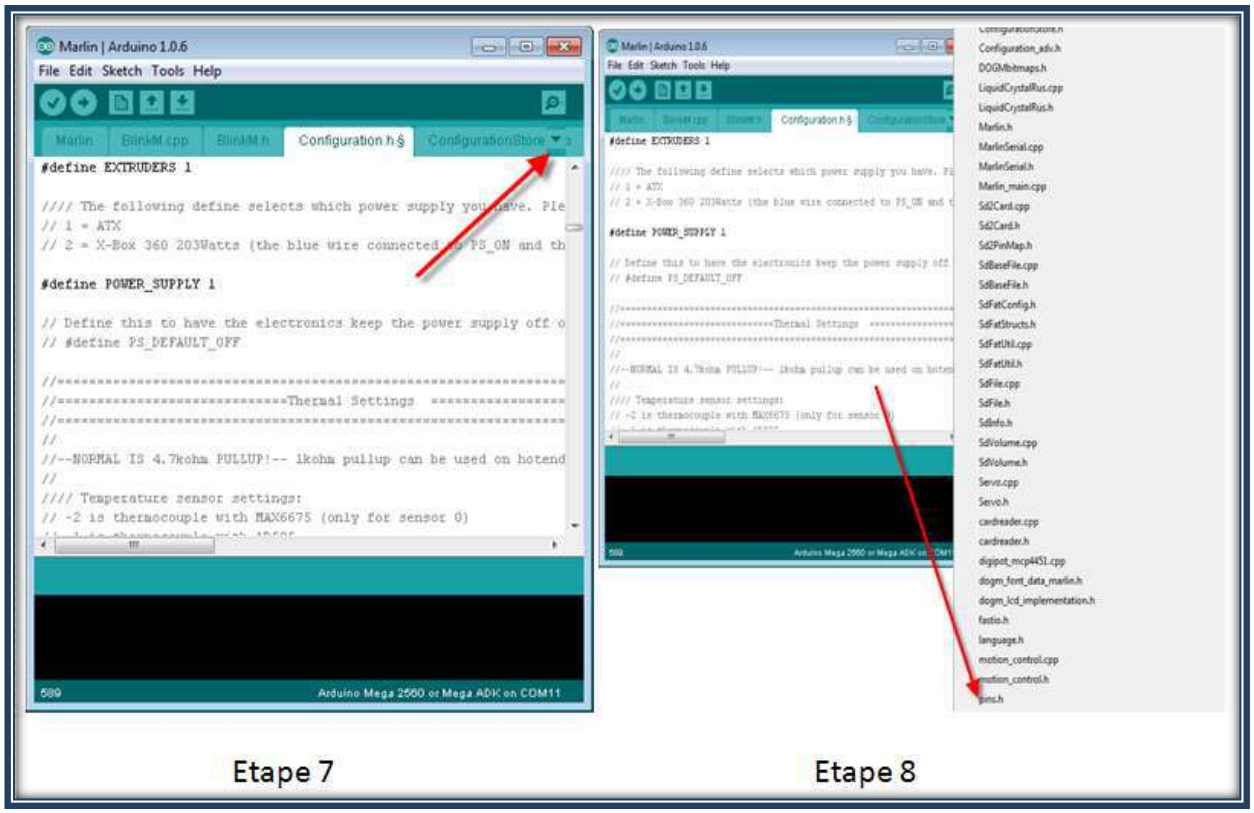
#define INVERT_E2_DIR false

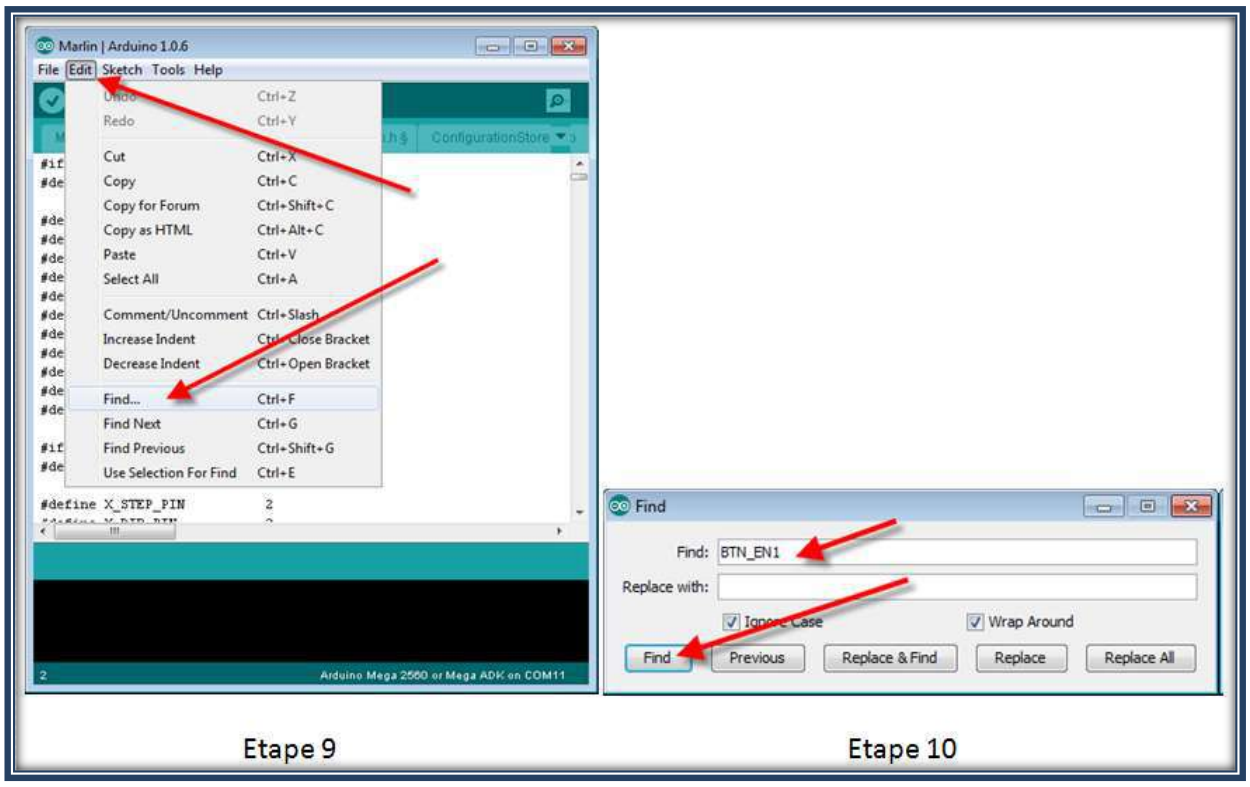
La suite de la configuration est indiquée sur les figures suivantes :

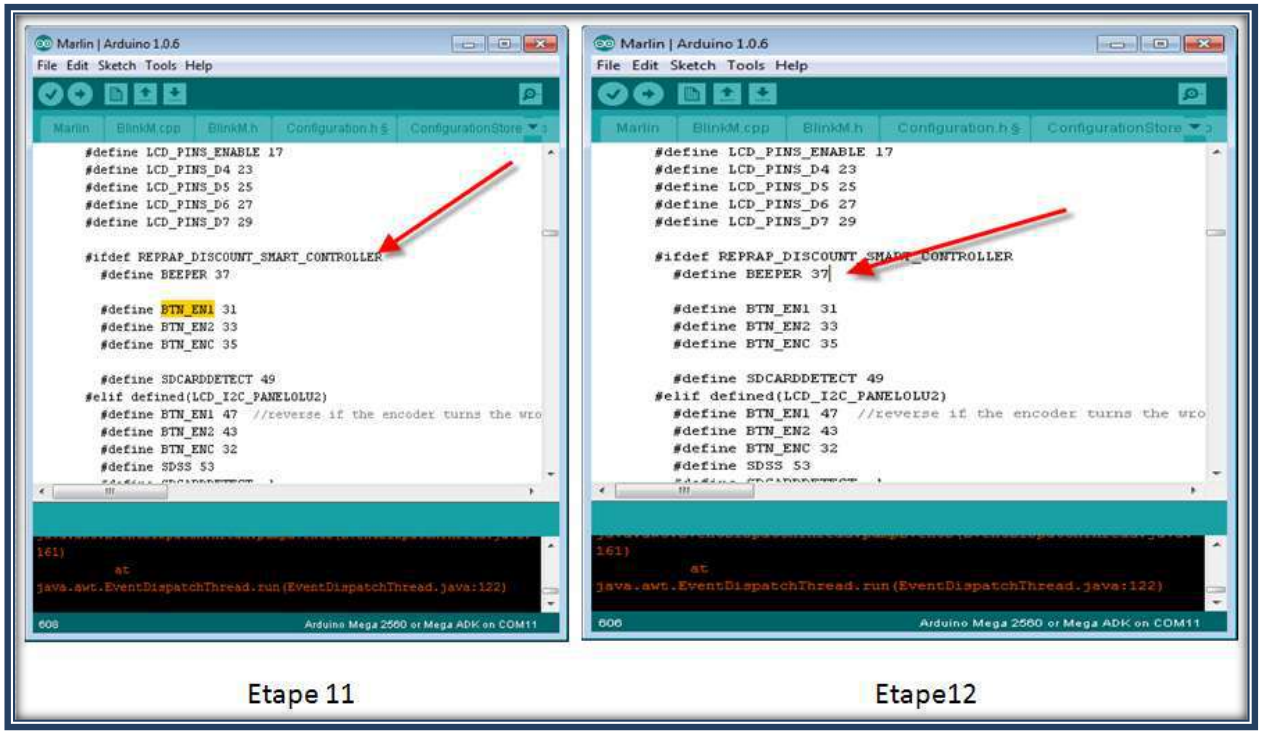


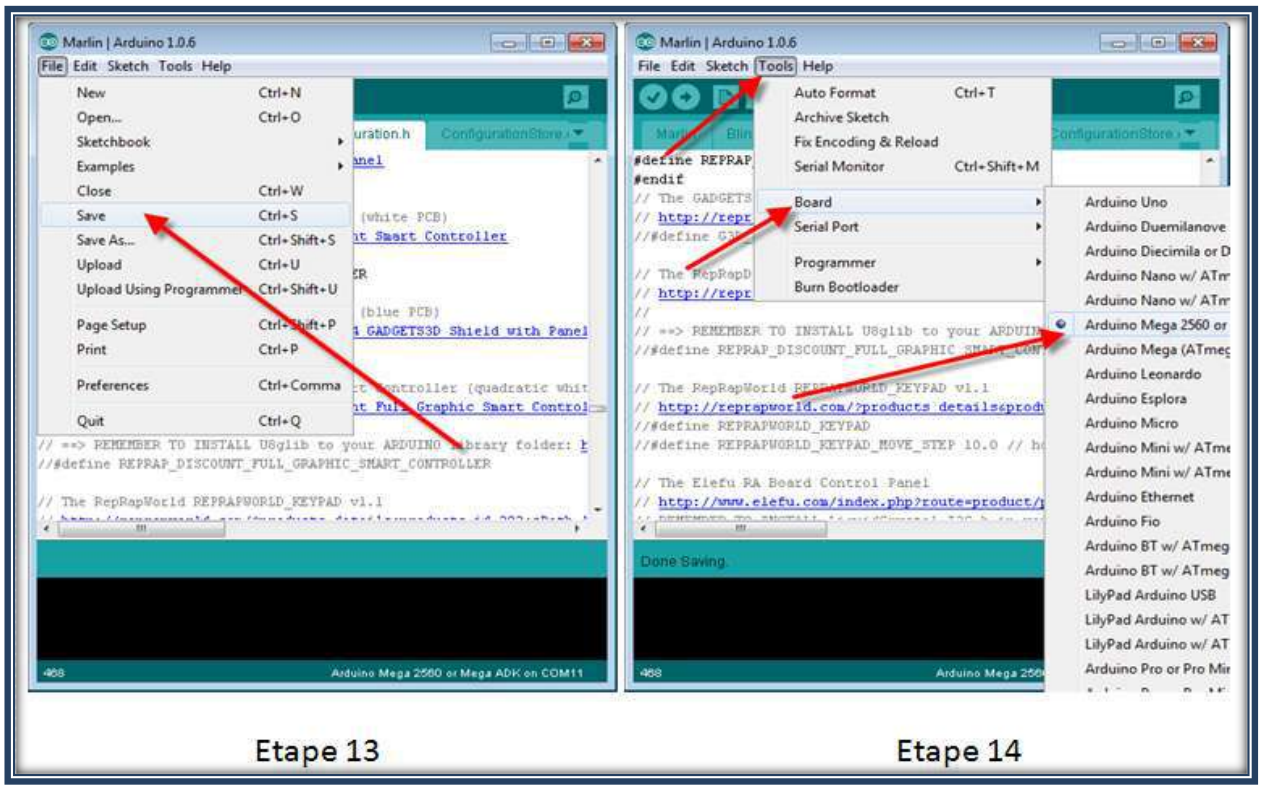

Marlin | Arduino 1.0.6
File Edit Sketch Tools Help
Verify / Compile Ctrl+R
Show Sketch Folder Ctrl+K
Add File...
Import Library...
Add Library...
EEPROM
Esplora
Ethernet
Firmata
GSM
LiquidCrystal
Robot_Control
Robot_Motor
RobotIRremote
SD
Servo
SoftwareSerial
SPI
Stepper
TFT
WiFi
Wire
U8glib
Done Saving.
Arduino Mega 2560 or Mega ADK on COM11
Etape3
Etape4

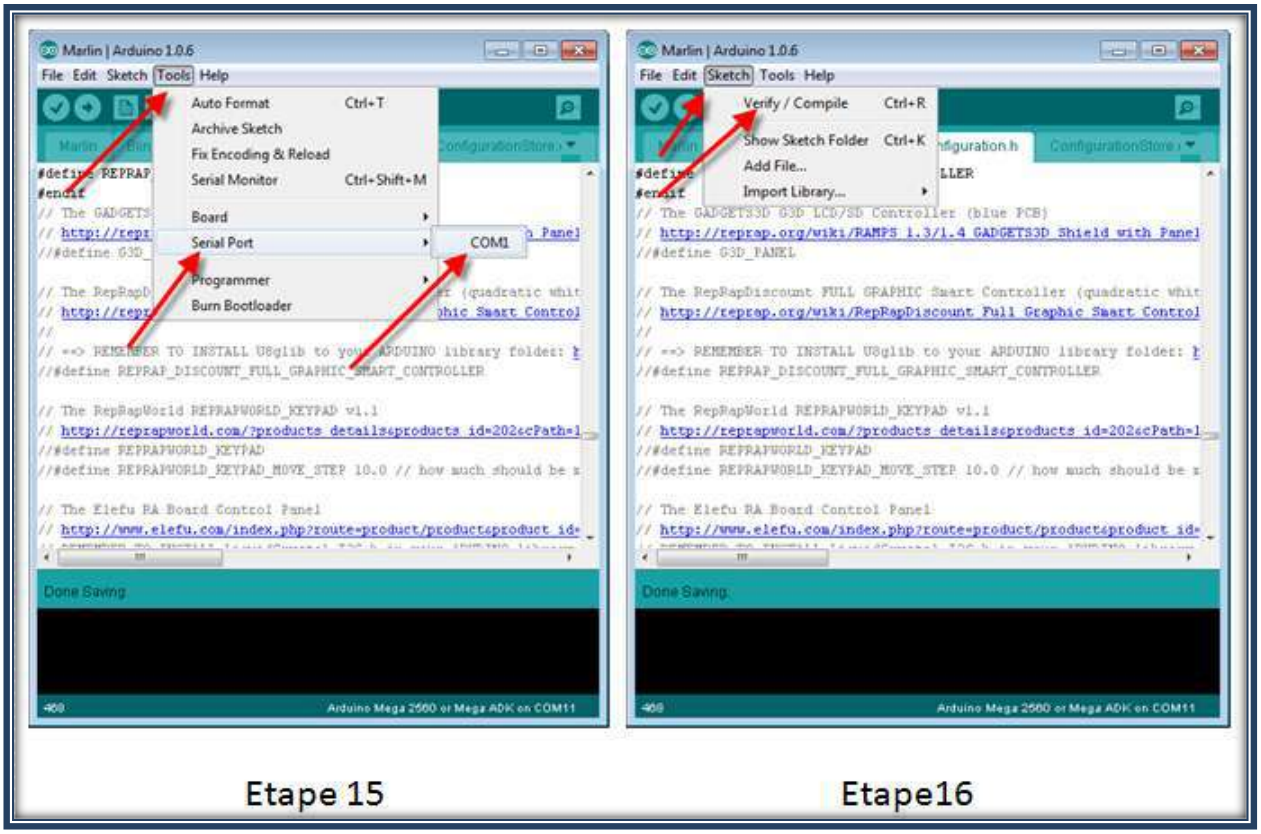

Marlin | Arduino 1.0.6
File Edit Sketch Tools Help
Done Saving.
Arduino Mega 2560 or Mega ADK on COM11
Etape 5
Etape 6

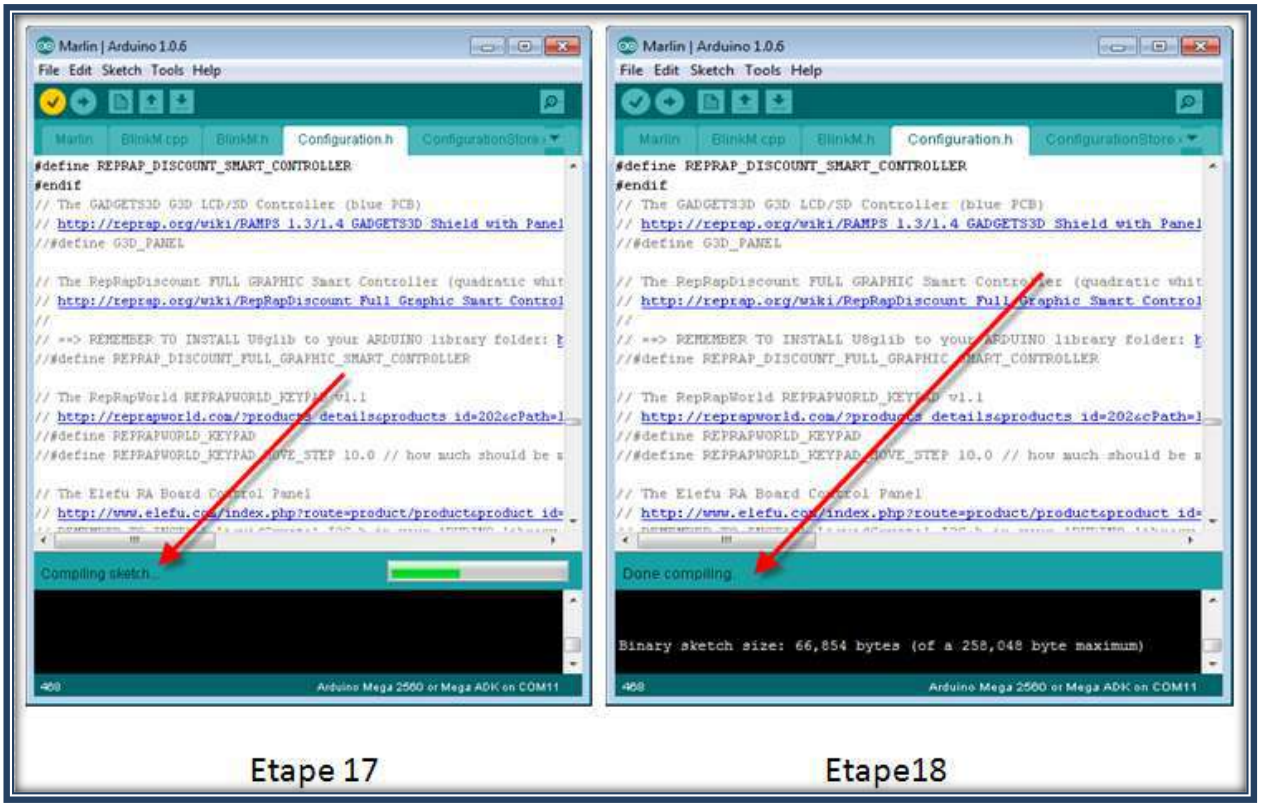

Marlin | Arduino 1.0.6
File Edit Sketch Tools Help
Configuration.h
#define EXTRUDERS 1
#define POWER_SUPPLY 1
Arduino Mega 2560 or Mega ADK on COM11
Etape 7
Etape 8

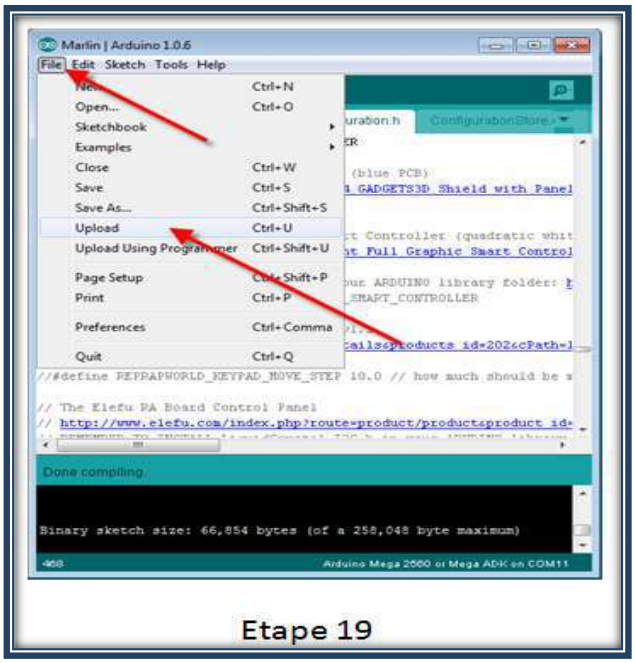

Marlin | Arduino 1.0.6
File Edit Sketch Tools Help
Undo Ctrl+Z
Redo Ctrl+Y
Cut Ctrl+X
Copy Ctrl+C
Copy for Forum Ctrl+Shift+C
Copy as HTML Ctrl+Alt+C
Paste Ctrl+V
Select All Ctrl+A
Comment/Uncomment Ctrl+Slash
Increase Indent Ctrl+Close Bracket
Decrease Indent Ctrl+Open Bracket
Find... Ctrl+F
Find Next Ctrl+G
Find Previous Ctrl+Shift+G
Use Selection For Find Ctrl+E
#define X_STEP_PIN 2
Arduino Mega 2560 or Mega ADK on COM11
Find
Find: BTN_EN1
Replace with:
Ignore Case
Wrap Around
Find
Previous
Replace & Find
Replace
Replace All
Etape 9
Etape 10


Marlin | Arduino 1.0.6
File Edit Sketch Tools Help
#define LCD_PINS_ENABLE 17
#define LCD_PINS_D4 23
#define LCD_PINS_D5 25
#define LCD_PINS_D6 27
#define LCD_PINS_D7 29
#ifdef REPRAP_DISCOUNT_SMART_CONTROLLER
#define BEEPER 37
#define BTN_EN1 31
#define BTN_EN2 33
#define BTN_ENC 35
#define SDCARDDETECT 49
#elif defined(LCD_I2C_PANELOLU2)
#define BTN_EN1 47 //reverse if the encoder turns the wro
#define BTN_EN2 43
#define BTN_ENC 32
#define SDSS 53
java.awt.EventDispatchThread.run(EventDispatchThread.java:122)
Arduino Mega 2560 or Mega ADK on COM11
Etape 11
Etape12


Marlin | Arduino 1.0.6
File Edit Sketch Tools Help
New Ctrl+N
Open... Ctrl+O
Sketchbook
Examples
Close Ctrl+W
Save Ctrl+S
Save As... Ctrl+Shift+S
Upload Ctrl+U
Upload Using Programmer Ctrl+Shift+U
Page Setup Ctrl+Shift+P
Print Ctrl+P
Preferences Ctrl+Comma
Quit Ctrl+Q
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Board
Serial Port
Programmer
Burn Bootloader
Arduino Uno
Arduino Duemilanove
Arduino Diecimila or D
Arduino Nano w/ ATm
Arduino Nano w/ ATm
Arduino Mega 2560 or
Arduino Mega (ATmeg
Arduino Leonardo
Arduino Esplora
Arduino Micro
Arduino Mini w/ ATme
Arduino Mini w/ ATme
Arduino Ethernet
Arduino Fio
Arduino BT w/ ATmeg
Arduino BT w/ ATmeg
LilyPad Arduino USB
LilyPad Arduino w/ AT
LilyPad Arduino w/ AT
Arduino Pro or Pro Mi
Done Saving.
Arduino Mega 2560 or Mega ADK on COM11
Etape 13
Etape 14


Marlin | Arduino 1.0.6
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Board
Serial Port
COM1
Programmer
Burn Bootloader
Verify / Compile Ctrl+R
Show Sketch Folder Ctrl+K
Add File...
Import Library...
Done Saving.
Arduino Mega 2560 or Mega ADK on COM11
Etape 15
Etape16


Marlin | Arduino 1.0.6
File Edit Sketch Tools Help
Marlin BlinkM.cpp BlinkM.h Configuration.h ConfigurationStore.cpp
#define REPRAP_DISCOUNT_SMART_CONTROLLER
#endif
Compiling sketch...
Done compiling.
Binary sketch size: 66,854 bytes (of a 258,048 byte maximum)
Arduino Mega 2560 or Mega ADK on COM11
Etape 17
Etape18



3.7 Impression d'un objet

Pour imprimer un objet en 3D il y a deux manières :

- Soit en utilisant la carte SD-Card ;
- Soit en utilisant l'ordinateur.

3.7.1 Impression depuis la carte SD-Card

Nous avons Appuyons sur le bouton du LCD pour accéder au menu, puis nous sélectionnons "print from SD" puis nous appuyons dessus lorsque le symbole «>». Maintenant, il affiche le contenu de la Carte SD. En tournant le bouton, nous pouvons faire défiler la liste des fichiers nécessaires. Nous sélectionnons le fichier pour l'impression. Une fois sélectionner le fichier l'impression commence d'imprimé par la première couche.

3.7.1.1 Arrêtement d'impression

Lors de l'impression à partir de la carte SD, nous pouvons mettre l'imprimante en pause ou abandonner l'impression. Les deux options sont disponibles directement dans le menu principal de " Pause" et "Stop printing ". Dans le premier cas (pause), bloque seulement le processus de création des couches, tout en maintenant l'extrudeuse à l'endroit où il est en attente de la commande "Play ", tandis que dans le second cas (stop printing), il n'est plus possible de continuer et l'extrudeur est repositionné à l'origine des coordonnées X et Y.

Malheureusement , en cas de pause trop longue, le point qui est fixé par l'extrudeur peut être affecter par la chaleur , est déforme l'impression. Ou on peut également créer une goutte de matériau qui est laissé par gravité de l'extrusion. Une pause, doit être utilisée qu'en cas de nécessité et que sur de courtes périodes, sinon il va laisser une marque dans le modèle.

Note :

Nous suggérons de l'activer uniquement lorsque l'extrudeur procède au remplissage : de cette manière toute marque restera caché à l'intérieur de la pièce moulée.

3.7.1.2 Modification des paramètres d'impression

Une des caractéristiques de Sharebot NG fournies par Merlin firmware est la possibilité d'intervenir sur certains paramètres lors de l'impression. Le menu «Tune» remplace «prepare» dès que l'impression commence. En sélectionnant «Tune», nous pouvons accéder au temps réel du réglage des paramètres. Par exemple nous pouvons modifier la vitesse d'alimentation, augmenter ou diminuer la vitesse d'impression globale pour gérer toutes les situations imprévues, telle qu'une matière qui nécessite plus de temps pour refroidir (et par conséquent devrait être imprimée lentement) ou d'un modèle qui peut être réalisé à une vitesse plus élevée que les paramètres. Un autre paramètre important est la vitesse d'écoulement qui augmente ou diminue en pourcentage la quantité de matière extrudée. En fait, avec ce paramètre, nous pouvons compenser les variations éventuelles dans le diamètre du fil. Le même paramètre nous permet d'épaissir ou d'amincir les murs du modèle.

L'onglet « FAN SPEED " peut être utile, dans le cas où le moteur de l'extrudeur est en surchauffe ou à réguler le flux d'air lors de l'impression.

3.7.1.3 Retrait de notre pièce terminé

Dès que l'impression est terminée, notre Sharebot NG positionne le plan vers le bas et arrête l'extrudeur. Dans cette position, nous pouvons facilement accéder à la surface d'impression et retirons le travail terminé. Il s'agit d'une opération délicate et inévitable parce que la première couche de l'objet doit bien adhérer et doit rester accrochée à la surface d'impression pendant l'ensemble du processus d'impression. Sans une bonne couche d'accroche tout stress mécanique ou même juste une petite bosse de la tête de l'extrudeur sur le modèle pourrait irrémédiablement endommager l'impression.

Détacher le modèle requiert donc une certaine habileté et une spatule mince, ou un couteau à mastic.

Dans les deux cas, nous devons essayer d'insérer les quelques millimètres de la lame sous le côté de l'objet imprimé. Par conséquent, nous devons plonger la lame sur le côté opposé par un mouvement horizontal sans essayer de soulever la pièce. Habituellement, le morceau saute sans trop d'effort, et sans que la lame ne ressorte de l'autre côté. Dans le cas où notre modèle a des pattes, ou, des pièces minuscules, nous devrions essayer de les détacher sans lever.

Après cette opération, nous pouvons retirer la partie avec le cadre le plus large sans casser les petites pièces. Dans le cas de vase ou de parties avec une base unique, nous devrions commencer par un côté en essayant de mettre la lame plus profondément par un mouvement oscillatoire pour faire avancer, mais sans pousser directement.

3.7.2 Impression depuis un ordinateur

Le schéma suivant résume les déférentes étapes :

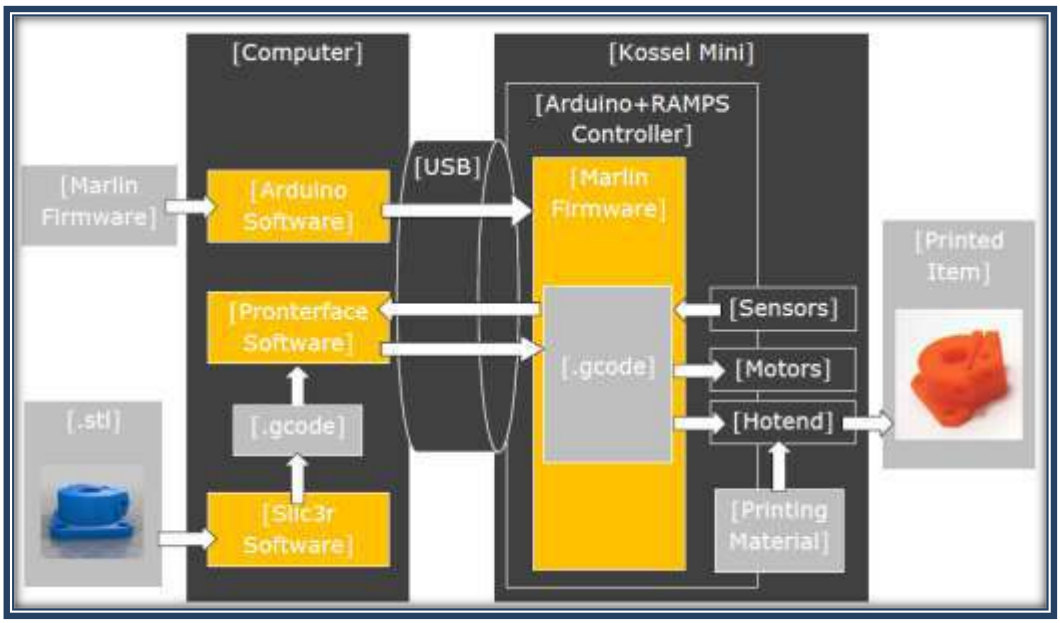


Figure 3.15: Les déférentes étapes de l'impression.

Pour utiliser un ordinateur avec Sharebot NG, nous devons installer un programme qui envoie le G-code à l'imprimante et contrôle son fonctionnement, tout en fonctionnant de manière similaire à l'imprimante via le bouton et le menu de l'écran LCD. En outre,

l'ordinateur permet à partir d'un modèle 3D - au format OBJ ou STL - de le transformer par le découpage et de traiter le fichier à imprimer. Sur internet, il existe plusieurs solutions, mais dans notre travail, nous recommandons le fichier Repetier - host et Slic3r. Ce " mélange " est plus que prouvé et il est utilisé par une grande partie de la communauté des utilisateurs d'imprimante 3D.

3.7.2.1 Repetier-Host

Le logiciel Repetier-Host est une interface simple qui facilite la connexion USB avec l'imprimante 3D, la visualisation de pièces en format .stl et leur laminage en .gcode, grâce aux laminoirs Slic3r ou Skeinforge.

3.7.2.2 Génération du format g-code

Ouvrons tout d'abord le fichier .stl sur le menu Fichier > Ouvrir. Sur l'onglet Slicer, nous avons la possibilité d'utiliser Slic3r ou Skeinforge. Dans notre travail, nous utilisons le Slic3r figure (3.16). Nous avons sélectionner les configurations et les profils de l'impression, de l'imprimante et de l'extrudeuse.

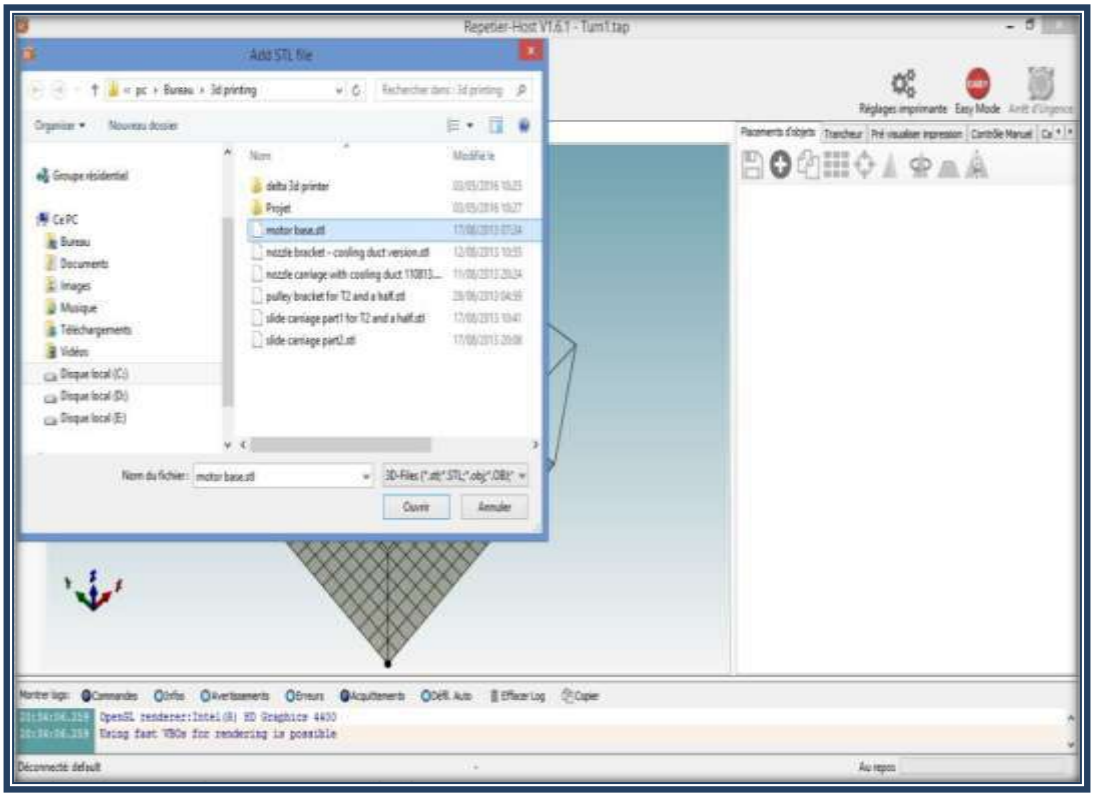


Figure 3.16: Fichier .stl.

La figure (3.17) montre le profil de la piece que nous souhaitons imprimeé.

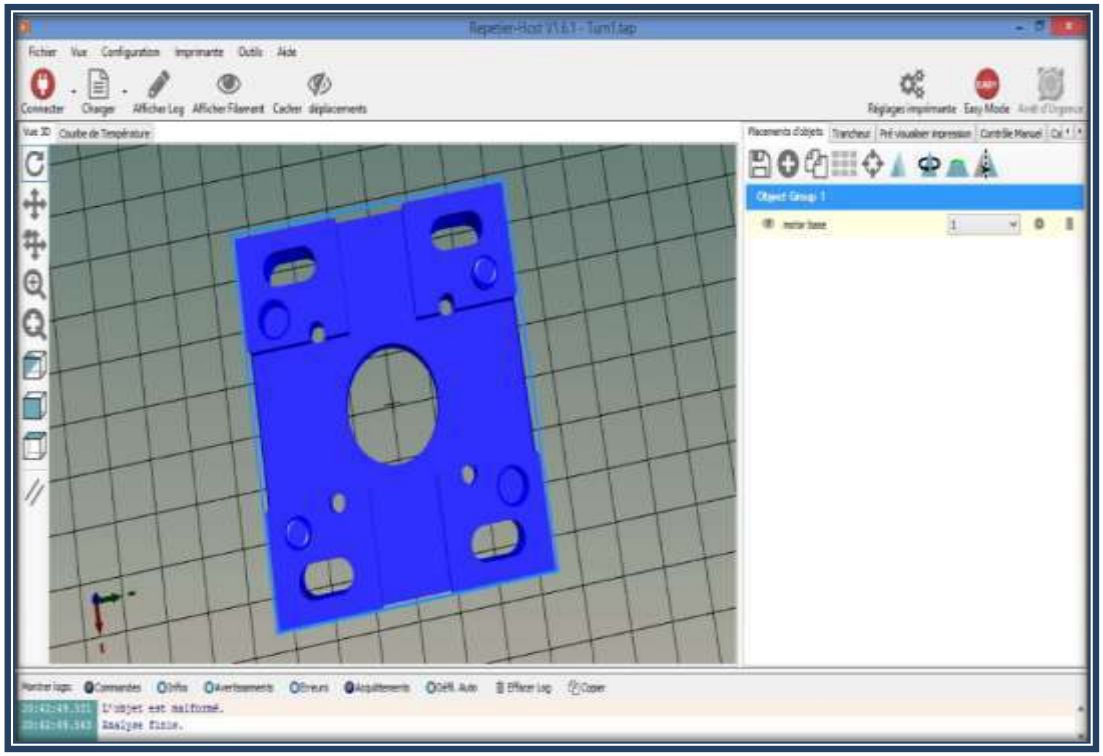


Figure 3.17: Pièce à imprimer.

Une fois le profil sélectionné, appuyons sur le bouton Trancher avec Slic3r comme l'indique la figure (3.18).

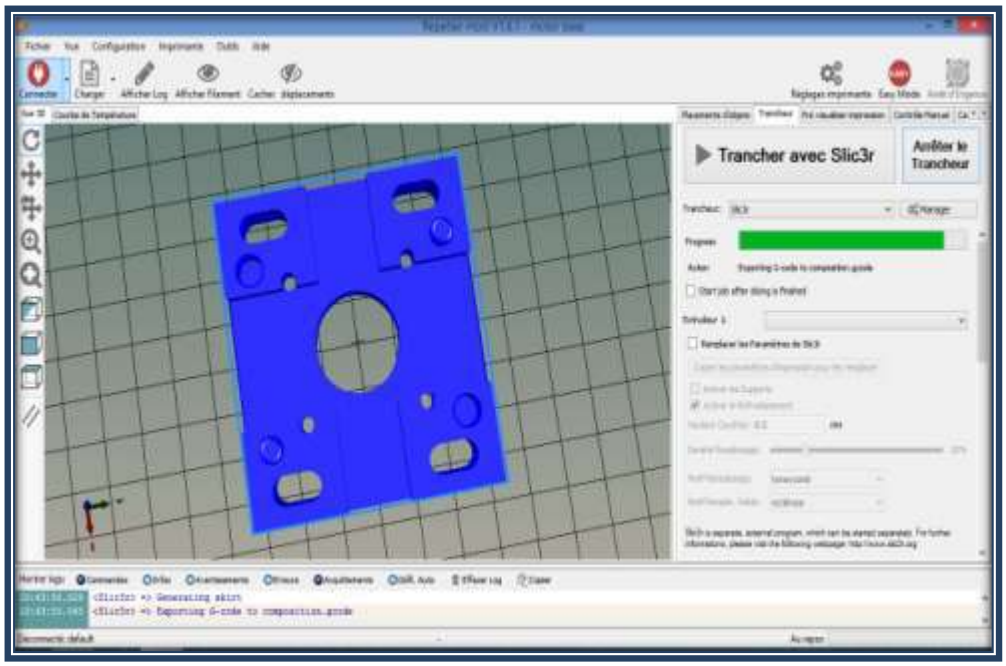


Figure 3.18:Operation de tranchage.

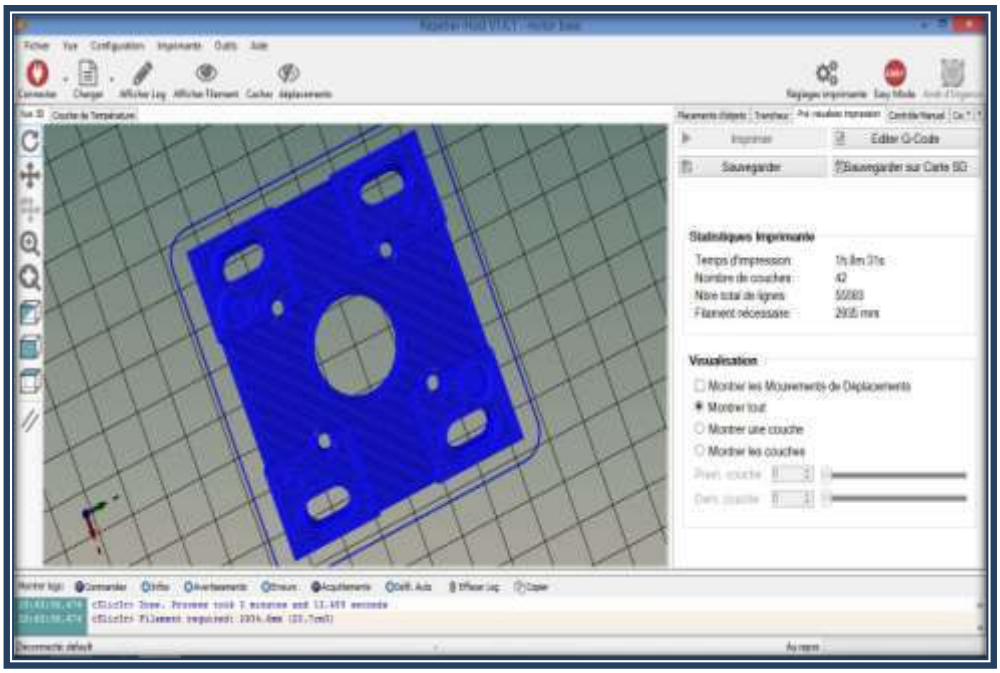


Figure 3.19: Pièce après le trancheur

Une fois l'operation de tranchage est terminée, appuyons sur le bouton Slice avec Slic3r qui génèrera automatiquement le fichier g-code(figure 3.20).

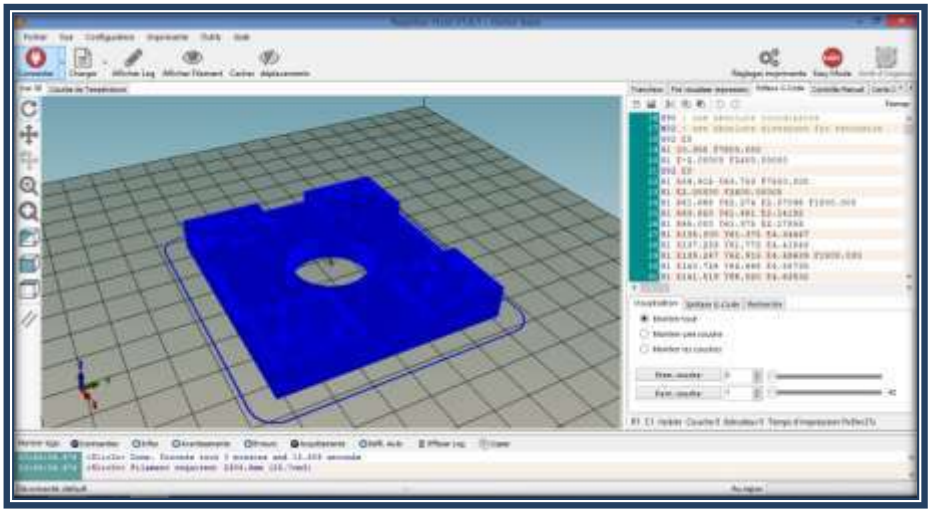


Figure 3.20: Fichier g-code.

3.7.2.3 Imprission de g-code généré

Une fois que nous disposerons d'un format g-code généré, nous pourrons imprimer la pièce que nous voudrons si la Witbox est connectée au Repetier-Host par USB (appuyons sur le bouton Connecter et attendons que la lumière verte s'allume). Nous pouvons vérifier que tout fonctionne correctement sur l'onglet Contrôle Manuel :

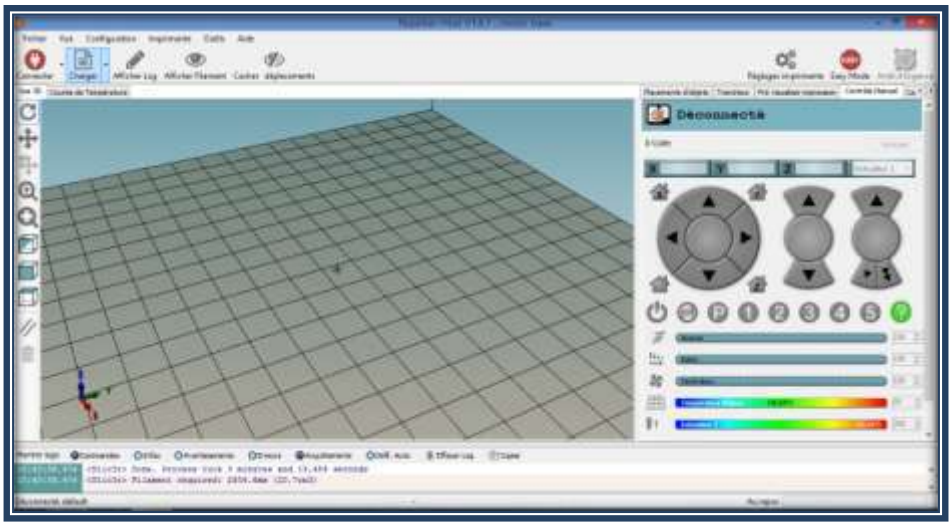


Figure 3.21: Contrôle manuel de l'ogiciel Repetier-Host.

Le menu indequé à la figure 3.21 permet de déplacer les axes de l'imprimante en appuyant sur les flèches ou sur les icônes en forme de maison (position home), ainsi que de faire fonctionner l'extrudeuse. Pour imprimer le g-code, il nous suffit d'appuyer sur le bouton Démarrer Job**.** Nous savons à présent utiliser les fonctions de base du logiciel Repetier-Host pour imprimer une pièce et commander l'imprimante 3D connectée par USB, mais ce logiciel nous offre d'autres options d'utilisation telles que, par exemple, sauvegarder directement le g-code généré sur la carte SD de l'imprimante ou accéder aux profils Slic3r pour les modifier à chaque impression. La figure 3.22 indique une piece prete à l'impression.

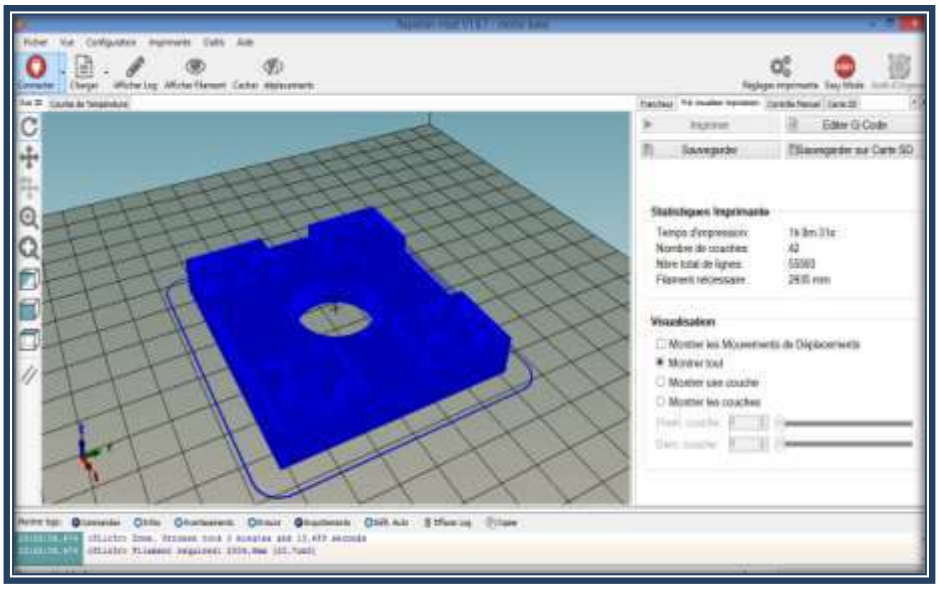


Figure 3.22: Piece prête à l'impression.

Nous pouvons également imprimer différents modèles à la fois en utilisant les options de l'ongle Placements d'objects et rajouter, éliminer, modéliser et placer les pièces dans la position que nous voulons.

3.5 Apliquation pratique de l'imprimante delta

Dans cette partie nous testons les performances de notre imprimante 3D delta à realiser des formes 2D (figure 3.23) . Apres assemblage des différentes parties, nous avons conçu une pièce par SolidWorks et la sauvegarde au format STL. La pièce est ensuite tranchée avec le logiciel KISSlicer et enregistrée au format g-code. Nous avons branché le robot printer 3D à l'ordinateur via le port USB ou avec la carte SD-Card qui contient le fichier g-code dans l'interface de pilotage du robot Repetier-Host et lancé l'operation d'impression en executant le fichier g-code. Les résultats pratiques obtenus motrent que notre imprimante presente de bonnes performances.

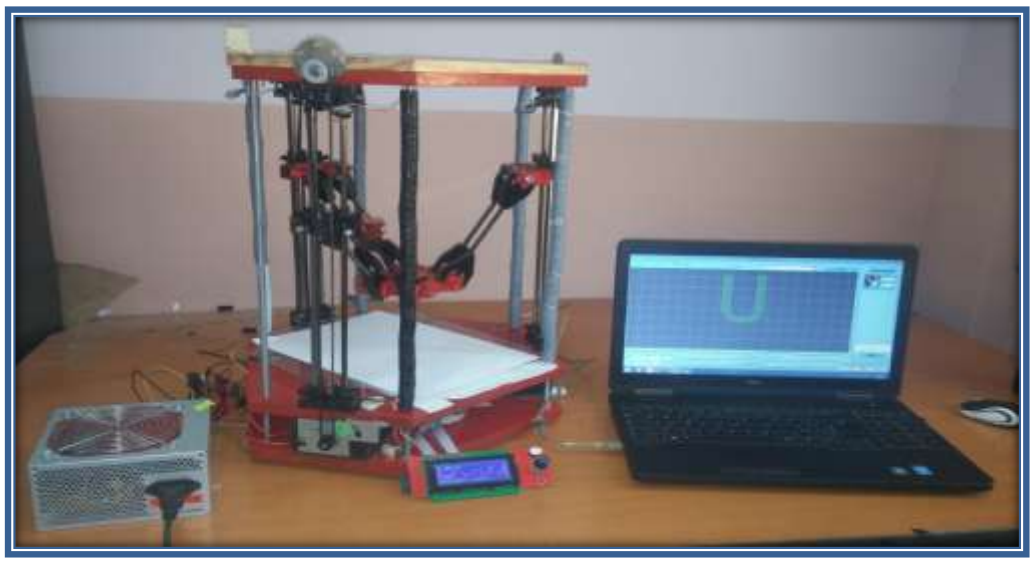


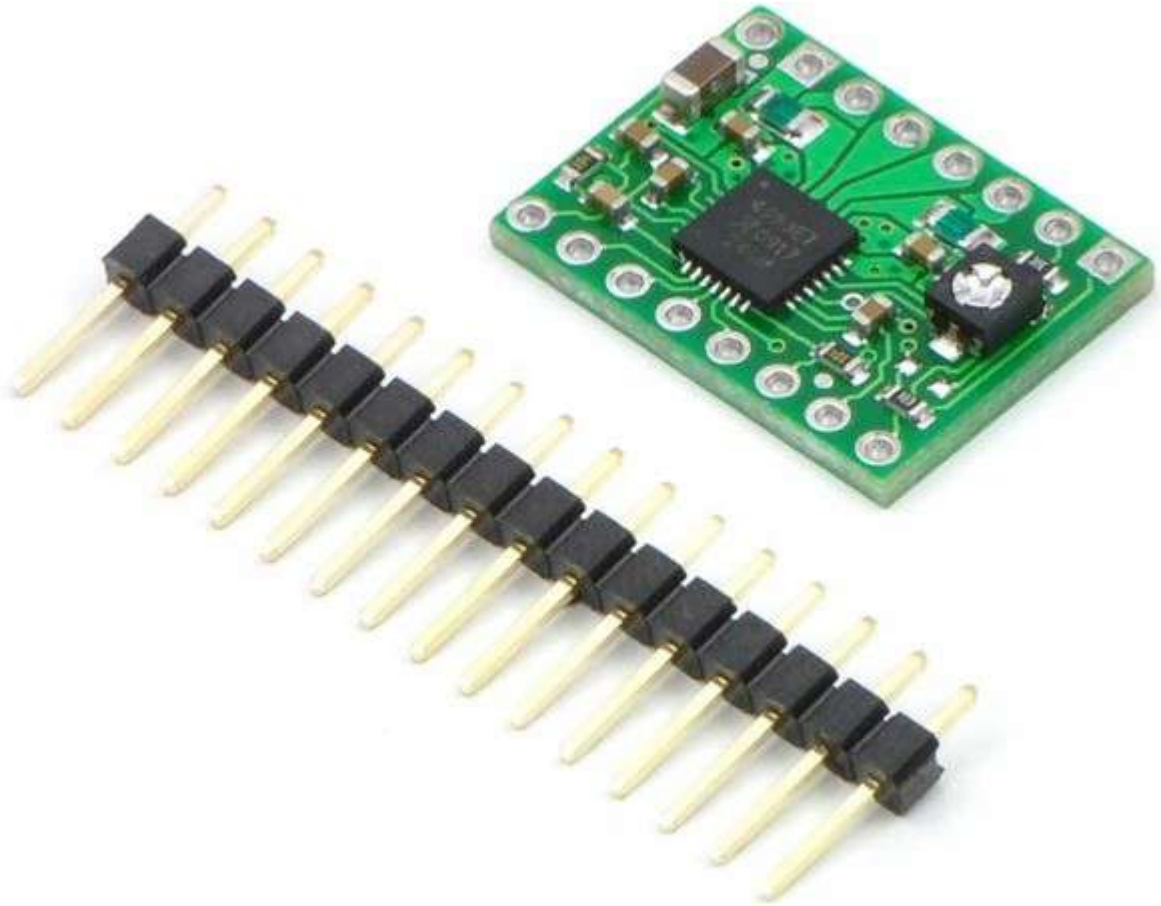
Figure 3.23 :Imprimante delta 3D

3.6 Conclusion

Dans ce chpitre, aprés l'assemblage des différentes parties, nous avons testé les performances de l'imprimante 3D delta que nous avons realisé. D'après les testes de l'imprimante montre de bonnes performances.

RB-Pol-176

Pololu 8-35V 2A Single Bipolar Stepper Motor Driver A4988



Stepper Motor Driver Carrier

The A4988 stepper motor driver carrier is a breakout board for Allegro's easy-to-use A4988 microstepping bipolar stepper motor driver and is a drop-in replacement for the A4983 stepper motor driver carrier. The driver features adjustable current limiting, overcurrent protection, and five different microstep resolutions. It operates from 8 – 35 V and can deliver up to 2 A per coil.

Note: This board is a drop-in replacement for the original A4983 stepper motor driver carrier. The newer A4988 offers overcurrent protection and has an internal 100k pull-down on the MS1 microstep selection pin, but it is otherwise virtually identical to the A4983.

Description

Overview

This product is a carrier board or breakout board for Allegro's A4988 DMOS Microstepping Driver with Translator and Overcurrent Protection; we therefore recommend careful reading of the A4988 datasheet (380k pdf) before using this product. This stepper motor driver lets you control one bipolar stepper motor at up to 2 A output current per coil (see the Power Dissipation Considerations section below for more information). Here are some of the driver's key features:

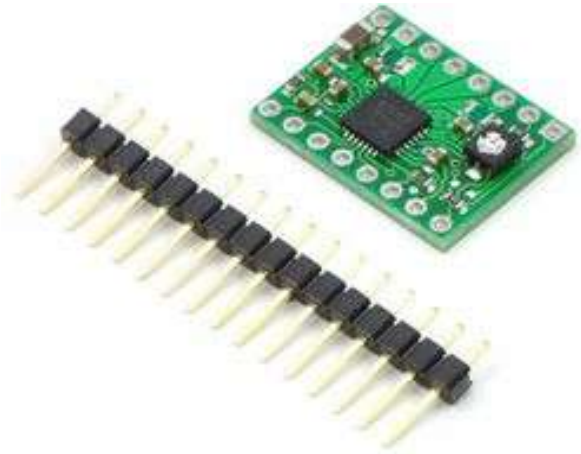
- Simple step and direction control interface
- Five different step resolutions: full-step, half-step, quarter-step, eighth-step, and sixteenth-step
- Adjustable current control lets you set the maximum current output with a potentiometer, which lets you use voltages above your stepper motor's rated voltage to achieve higher step rates
- Intelligent chopping control that automatically selects the correct current decay mode (fast decay or slow decay)
- Over-temperature thermal shutdown, under-voltage lockout, and crossover-current protection
- Short-to-ground and shorted-load protection (this feature is not available on the A4983)

Like nearly all our other carrier boards, this product ships with all surface-mount components —including the A4988 driver IC—installed as shown in the product picture.

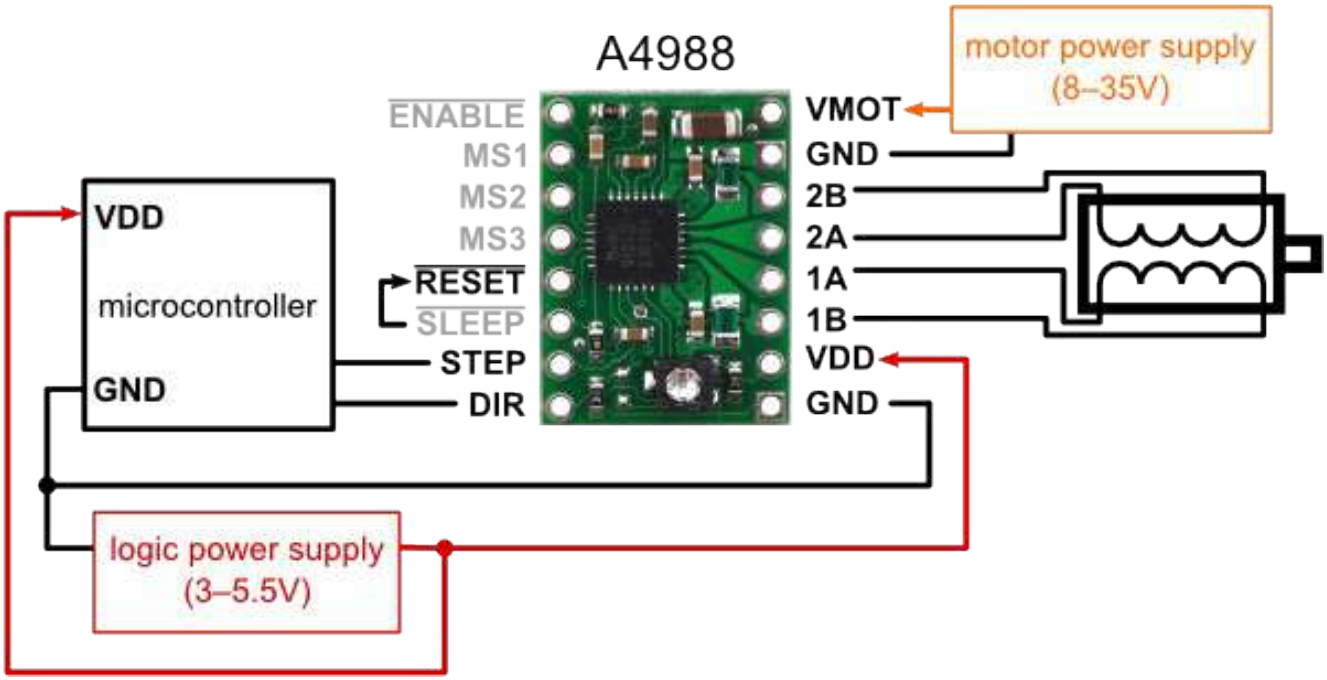
We also sell a larger version of the A4988 carrier that has reverse power protection on the main power input and built-in 5 V and 3.3 V voltage regulators that eliminate the need for separate logic and motor supplies.

Included hardware

The A4988 stepper motor driver carrier comes with one 1×16-pin breakaway 0.1" male header. The headers can be soldered in for use with solderless breadboards or 0.1" female connectors. You can also solder your motor leads and other connections directly to the board.



Using the driver



Minimal wiring diagram for connecting a microcontroller to an A4988 stepper motor driver carrier (full-step mode).

Power connections

The driver requires a logic supply voltage (3 – 5.5 V) to be connected across the VDD and GND pins and a motor supply voltage of (8 – 35 V) to be connected across VMOT and GND. These supplies should have appropriate decoupling capacitors close to the board, and they should be capable of delivering the expected currents (peaks up to 4 A for the motor supply).

Motor connections

Four, six, and eight-wire stepper motors can be driven by the A4988 if they are properly connected; a FAQ answer explains the proper wirings in detail.
Warning: Connecting or disconnecting a stepper motor while the driver is powered can destroy the driver. (More generally, rewiring anything while it is powered is asking for trouble.)

Warning: Connecting or disconnecting a stepper motor while the driver is powered can destroy the driver. (More generally, rewiring anything while it is powered is asking for trouble.)

Step (and microstep) size

Stepper motors typically have a step size specification (e.g. 1.8° or 200 steps per revolution), which applies to full steps. A microstepping driver such as the A4988 allows higher resolutions by allowing intermediate step locations, which are achieved by energizing the coils with intermediate current levels. For instance, driving a motor in quarter-step mode will give the 200-step-per-revolution motor 800 microsteps per revolution by using four different current levels.

The resolution (step size) selector inputs (MS1, MS2, MS3) enable selection from the five step resolutions according to the table below. MS1 and MS3 have internal 100kΩ pull-down resistors and MS2 has an internal 50kΩ pull-down resistor, so leaving these three microstep selection pins disconnected results in full-step mode. For the microstep modes to function correctly, the current limit must be set low enough (see below) so that current limiting gets engaged. Otherwise, the intermediate current levels will not be correctly maintained, and the motor will effectively operate in a full-step mode.

MS1	MS2	MS3	Microstep Resolution
Low	Low	Low	Full step
High	Low	Low	Half step
Low	High	Low	Quarter step
High	High	Low	Eighth step
High	High	High	Sixteenth step

Control inputs

Each pulse to the STEP input corresponds to one microstep of the stepper motor in the direction selected by the DIR pin. Note that the STEP and DIR pins are not pulled to any particular voltage internally, so you should not leave either of these pins floating in your application. If you just want rotation in a single direction, you can tie DIR directly to VCC or

GND. The chip has three different inputs for controlling its many power states: RST, SLP, and EN. For details about these power states, see the datasheet. Please note that the RST pin is floating; if you are not using the pin, you can connect it to the adjacent SLP pin on the PCB.

Current limiting

To achieve high step rates, the motor supply is typically much higher than would be permissible without active current limiting. For instance, a typical stepper motor might have a maximum current rating of 1 A with a 5Ω coil resistance, which would indicate a maximum motor supply of 5 V. Using such a motor with 12 V would allow higher step rates, but the current must actively be limited to under 1 A to prevent damage to the motor.

The A4988 supports such active current limiting, and the trimmer potentiometer on the board can be used to set the current limit. One way to set the current limit is to put the driver into full-step mode and to measure the current running through a single motor coil without clocking the STEP input. The measured current will be 0.7 times the current limit (since both coils are always on and limited to 70% in full-step mode). Please note that the current limit is dependent on the Vdd voltage.

Another way to set the current limit is to measure the voltage on the “ref” pin and to calculate the resulting current limit (the current sense resistors are 0.05Ω). The ref pin voltage is accessible on a via that is circled on the bottom silkscreen of the circuit board. See the A4988 datasheet for more information.

Power dissipation considerations

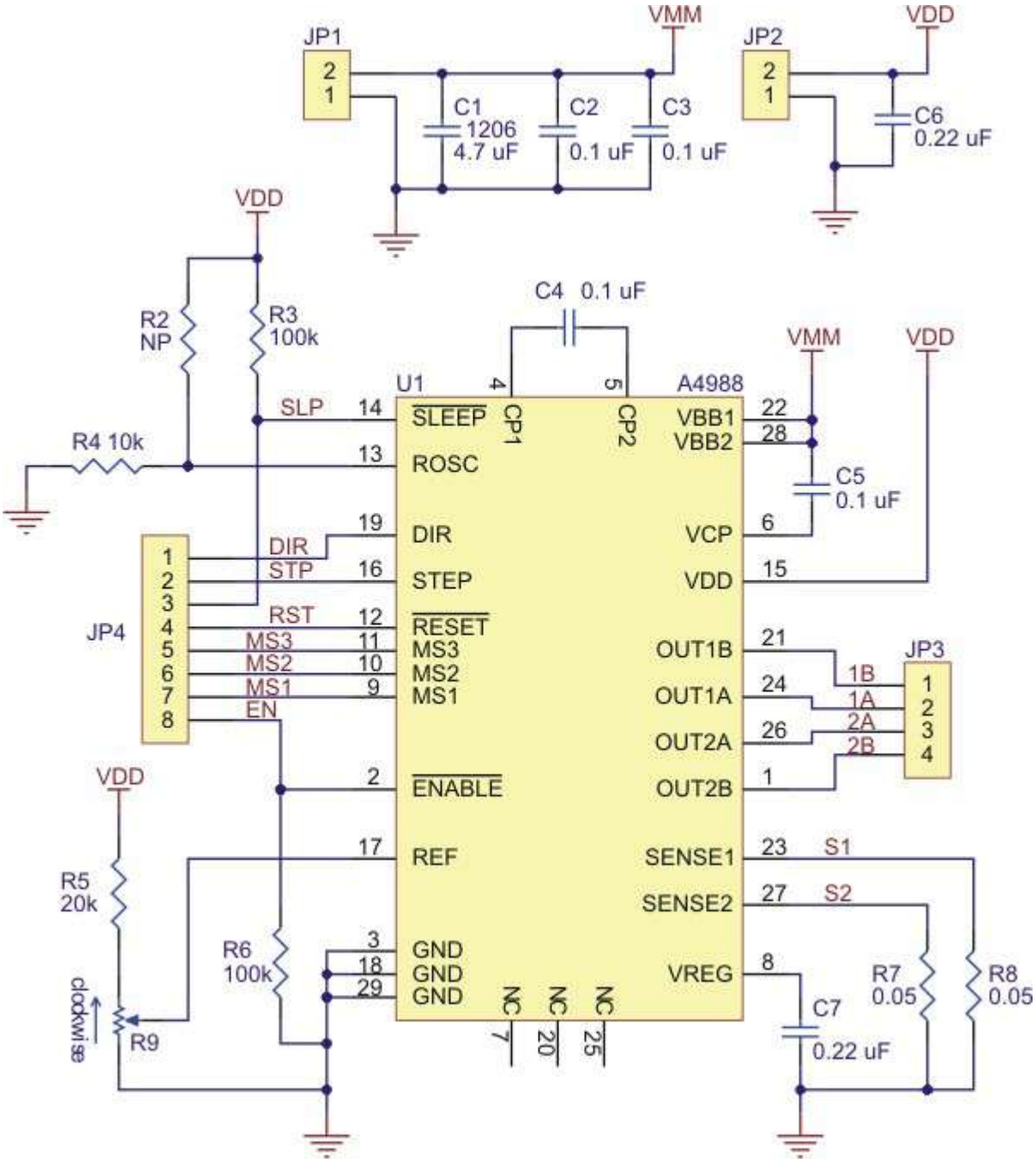
The A4988 driver IC has a maximum current rating of 2 A per coil, but the actual current you can deliver depends on how well you can keep the IC cool. The carrier’s printed circuit board is designed to draw heat out of the IC, but to supply more than approximately 1 A per coil, a heat sink or other cooling method is required.

This product can get hot enough to burn you long before the chip overheats. Take care when handling this product and other components connected to it.

Please note that measuring the current draw at the power supply does not necessarily provide an accurate measure of the coil current. Since the input voltage to the driver can be significantly higher than the coil voltage, the measured current on the power supply can be quite a bit lower than the coil current (the driver and coil basically act like a switching step-down power supply). Also, if the supply voltage is very high compared to what the motor needs to achieve the set current, the duty cycle will be very low, which also leads to significant differences between average and RMS currents.

Schematic diagram

Schematic diagram of the md09b A4988 stepper motor driver carrier.



PBC LINEAR
A PACIFIC BEARING CO.

Stepper Motor NEMA 17

This document describes mechanical and electrical specifications for PBC Linear stepper motors; including standard, hollow, and extended shaft variations.

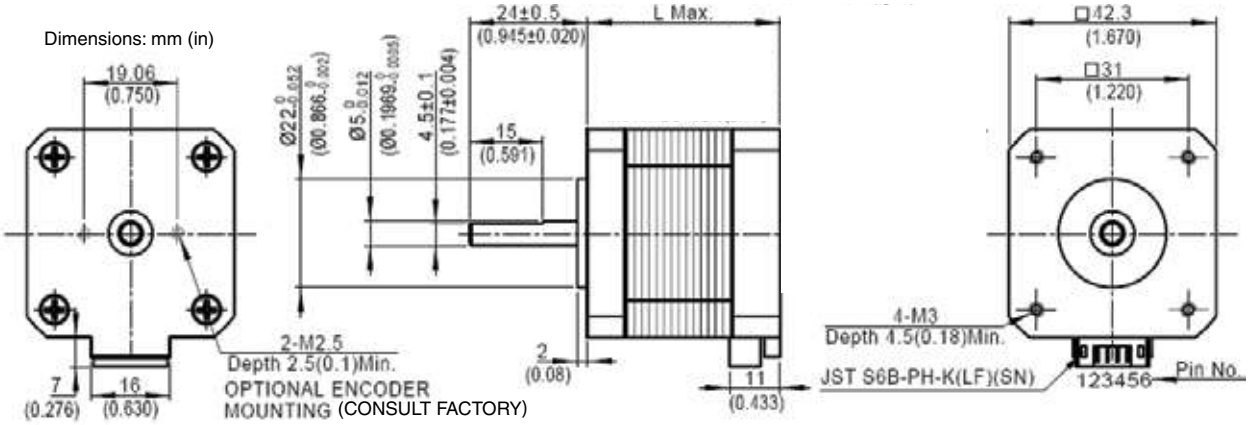


Standard shaft motor shown.

Phases	2
Steps/Revolution	200
Step Accuracy	±5%
Shaft Load	20,000 Hours at 1000 RPM
Axial	25 N (5.6 lbs.) Push
	65 N (15 lbs.) Pull
Radial	29 N (6.5 lbs.) At Flat Center
IP Rating	40
Approvals	RoHS
Operating Temp	-20° C to +40° C
Insulation Class	B, 130° C
Insulation Resistance	100 MegOhms

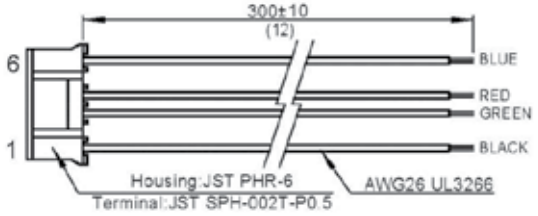
Description	Length	Mounted Rated Current	Mounted Holding Torque		Winding Ohms mH		Detent Torque		Rotor Inertia		Motor Weight	
(Stack)	"L" Max	Amps	Nm Typ.	oz-in Typ.	±10% @ 20°C	Typ.	mNm	oz-in	g cm2	oz-in2	kg	lbs
Single	39.8 mm (1.57 in)	2	0.48	68	1.04	2.2	15	2.1	57	0.31	0.28	0.62
Double	48.3 mm (1.90 in)	2	0.63	89	1.3	2.9	25	3.5	82	0.45	0.36	0.79
Triple	62.8 mm (2.47 in)	2	0.83	120	1.49	3.8	30	4.2	123	0.67	0.6	1.3

*All standard motors have plug connector. Consult factory for other options.



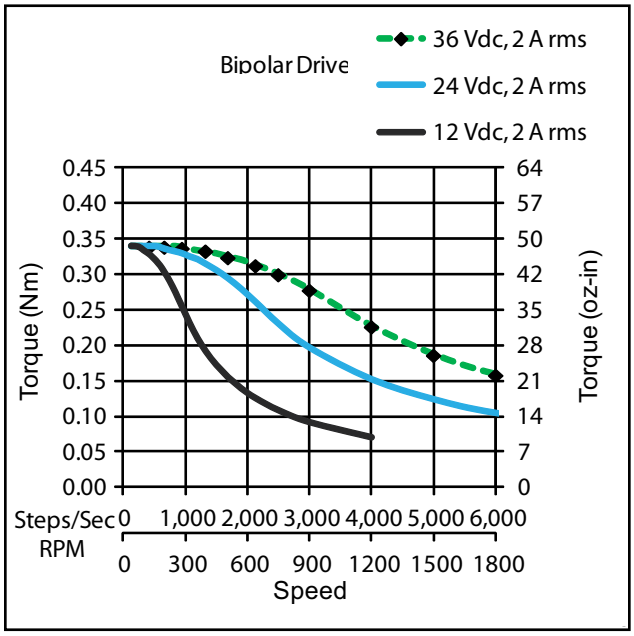
Standard shaft dimensions shown. All other dimensions apply to hollow and extended shaft options.

Dimensions: mm (in)
4 Lead Connector, PBC Part#6200490
(Consult factory for optional motor connectors)

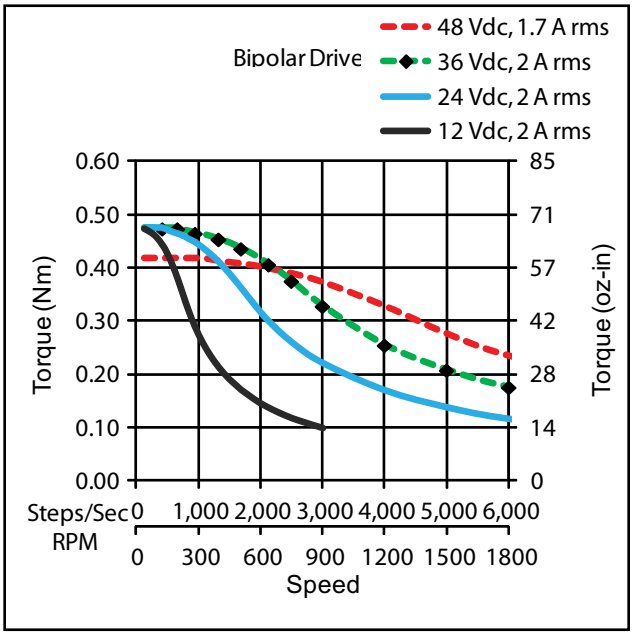


NEMA 17 **Stepper Motor**

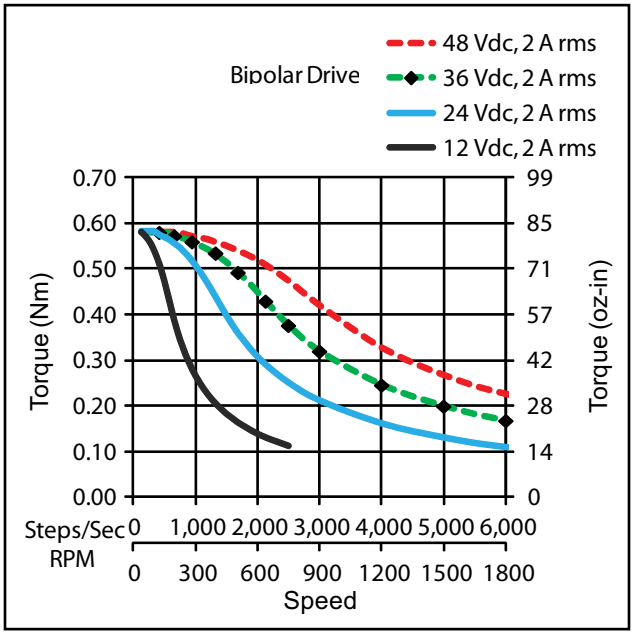
*Performance curves apply to continuous duty cycles.
Consult factory for intermittent cycles or other voltages.



Single Stack



Double Stack



Triple Stack

PBC LINEAR
A PACIFIC BEARING CO.

Stepper Motor NEMA 23

This document describes mechanical and electrical specifications for PBC Linear stepper motors; including standard, hollow, and extended shaft variations.

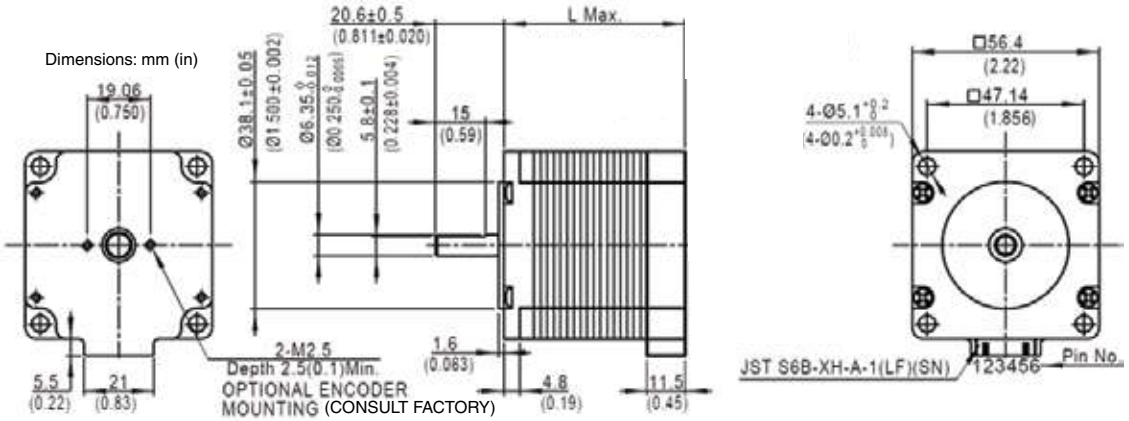


Standard shaft motor shown.

Phases	2
Steps/Revolution	200
Step Accuracy	±5%
Shaft Load	20,000 Hours at 1000 RPM
Axial	40 N (9 lbs.) Push 130 N (30 lbs.) Pull
Radial	70 N (15.5 lbs.) At Flat Center
IP Rating	40
Approvals	RoHS
Operating Temp	-20° C to +40° C
Insulation Class	B, 130° C
Insulation Resistance	100 MegOhms

Description	Length	Mounted Rated Current	Mounted Holding Torque		Winding Ohms mH		Detent Torque		Rotor Inertia		Motor Weight	
(Stack)	"L" Max	Amps	Nm Typ.	oz-in Typ.	±10% @ 20°C	Typ.	mNm	oz-in	g cm2	oz-in2	kg	lbs
Single	55.0 mm (2.17 in)	2.2	1.50	210	1.6	6.9	45	6.4	220	1.2	0.6	1.3
Double	77.0 mm (3.03 in)	3	2.30	330	1.1	4.5	75	11	390	2.1	1	2.2
Power Plus (Triple)	77.0 mm (3.03 in)	3	3.30	470	1.1	3.7	150	21	390	2.1	1.1	2.4

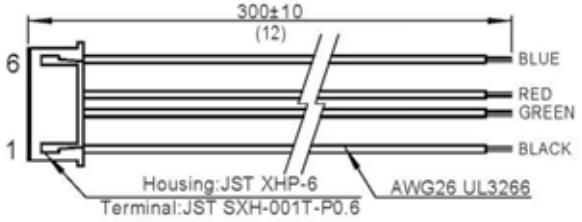
*All standard motors have plug connector. Consult factory for other options.



Motor with leads: Lead wire is 22 AWG UL3266, 300 ±10 (12 ±.5) long

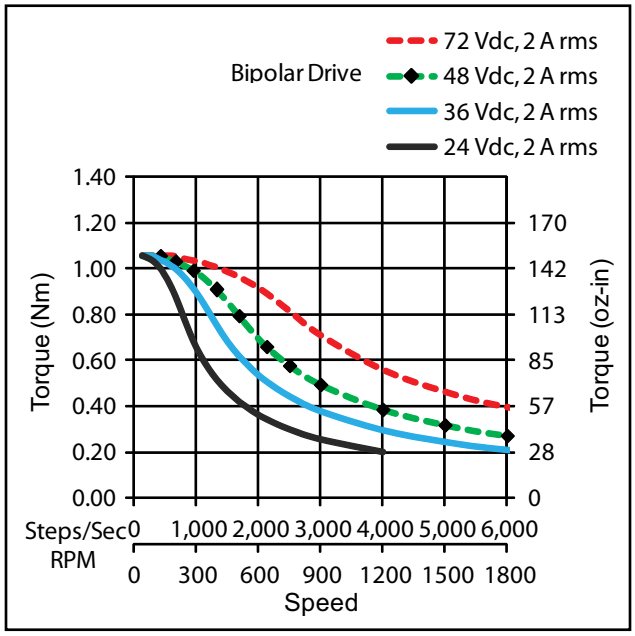
Standard shaft dimensions shown. All other dimensions apply to hollow and extended shaft options.

Dimensions: mm (in)
4 Lead Connector, PBC Part#6200491
(Consult factory for optional motor connectors)

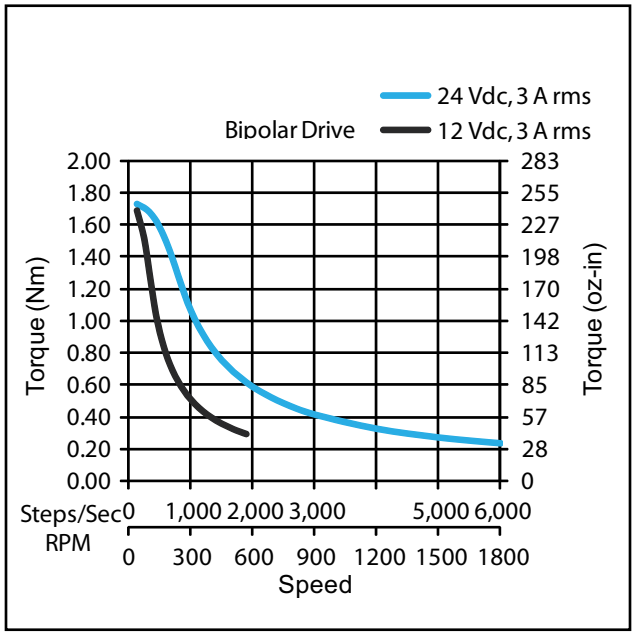


NEMA 23 **Stepper Motor**

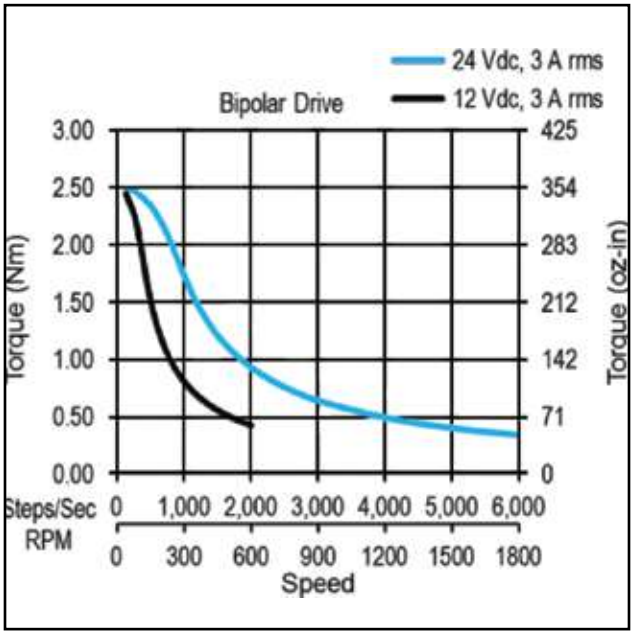
*Performance curves apply to continuous duty cycles.
Consult factory for intermittent cycles or other voltages.



Single Stack



Double Stack



Power Plus (Triple Stack)

Conclusion générale

Conclusion générale

Dans ce mémoire nous avons étudié, dimensionné et réalisé une imprimante 3D delta.

Au début, nous avons présenté des notions importantes sur l'imprimante 3D ainsi que leurs différentes technologies, puis nous nous sommes intéressés au robot delta et à son principe de fonctionnement. En fin nous avons présenté les avantages et les inconvénients de l'imprimante 3D.

Nous nous somme intéressé dans un deuxième temps à la structure mécanique et à la partie électronique de l'imprimante 3D. Nous avons commencé par le dimensionnement de notre imprimante 3D. Puis, nous avons fabriqué les pièces mécaniques nécessaires à la réalisation de notre projet. Afin de commander notre système nous avons utilisé la carte Arduino Mega 2560 qui est simple à programmer. En fin, nous avons assemblé les parties mécaniques et électroniques de l'imprimante 3D.

Dans la dernière partie, nous avons adopté les différentes étapes de la commande de notre imprimante 3D. Nous avons débuté par la schématisation d'une pièce par SolidWorks et la sauvegardé au format STL. La pièce est ensuite tranchée avec le logiciel KISSlicer et enregistrée au format g-code. A la fin, nous avons branché l'imprimante 3D à l'ordinateur via le port USB (ou avec la carte SD-Card) qui contient le fichier g-code dans l'interface de pilotage de l'imprimante Repetier-Host et nous avons lancé l'opération d'impression en exécutant le fichier g-code. Les résultats obtenus montrent que l'imprimante 3D que nous avons réalisé présente de bonnes performances.

Bibliographique

Bibliographie

[1] **Said BOUGOUR. « L'impression 3D comme moyen d'évolution des processus de production »**. CAHIERS Du LAB.RII- Document de travail-. Université Du Littoral Cote D'opale. Janvier 2015.

[2] **Christelle PAYEN. «L'impression 3D ».** Eurasanté à l'attention des entreprises de la région Nord-Pas de Calais. Décembre 2013.

[3] **Marc BURBRIDGE. «Introduction à l'impression 3D».** cours 2013.

[4] **Toufik BENTALEB. « Contribution à l'étalonnage géométrique des robots industriels »**. Mémoire de Magister. Ecole Militaire Polytechnique. 2006.

[5] **David CORBEL. «Contribution à l'amélioration de la précision des robots parallèles».** Mémoire de doctorat. Université Montpellier II. 2008.

[6] **Pierrick BOUFFARON. « Impression 3D».** Rapport d'Ambassade / Consulat Général de France à San Francisco, Californie septembre 2014.

[7] **Charline CHOISNARD et Romain ELINEAU. « L'impression en volumes».** Mémoire de licence professionnel. juin 2011.

[8] **M. BOURI «Projet robot delta Travail et recherches préparatoire».** «Projet 2014-2015 PCSI». France.

[9] **Etienne. GAUCHERON. «Les moteurs électriques pour mieux les piloter et les protéger».** Cahier technique n° 207.Juin 2004.

[10] Arduino Mega 2560 DataSheet.