

UNIVERSITE KASDI MERBAH OUARGLA

Faculté des sciences appliquées

Département de génie mécanique



Mémoire de Fin d'Etudes

De MASTER ACADEMIQUE

Domaine : Sciences et Technologies

Filière : Génie mécanique

Spécialité : Fabrication mécanique et Productique

Présenté par :

LEHZIEL MOHAMMED

Daoudi

MOHAMMED

Thème

Conception et réalisation d'une machine CNC

Soutenu publiquement le :

Devant le jury :

BENNOUNA Mohamed Salah

MCA

Président

MEZOUDJ Mourad

MAB

Examineur

Gherfi abdelhafid

MAA

Encadreur

Ziani Lotfi

MCB

Co-encadreur

Année universitaire 2021/2022

ملخص

في عصرنا الحالي يتطلب انشاء أي مشروع الات ومعدات متطورة لضمان سيرورة المشروع ونجاحه و هذا ما دفعنا الى تصنيع آلة التحكم العددي بالكمبيوتر بحيث قدمنا من خلاله بعض المفاهيم الرئيسية حول الات التحكم العددي بالكمبيوتر و المحرك الخطوي فيما يتعلق بالجانب الميكانيكي، قمنا بتصميم و تصنيع نموذج آلة التحكم العددي بالكمبيوتر.

الكلمات المفتاحية: آلة التحكم الرقمي، المتحكم الرقمي، المحرك الخطوي،

Résumé

À notre époque actuelle, la mise en place de tout projet nécessite des machines et des équipements de pointe pour assurer l'avancement et le succès du projet. C'est ce qui nous a incités à fabriquer une machine à commande numérique par ordinateur, à travers laquelle nous avons présenté quelques concepts clés sur les machines à commande numérique par ordinateur et le moteur pas à pas. En ce qui concerne l'aspect mécanique, nous avons conçu et fabriqué un modèle de machine à commande numérique par ordinateur.

Mots-clés : machine à commande numérique, contrôleur numérique, moteur pas à pas, SolidWorks.

Abstract

In our present time, setting up any project requires advanced machinery and equipment to ensure the progress and success of the project, and this is what prompted us to manufacture a CNC machine, through which we introduced some key concepts about CNC machines and stepper motor. Regarding the mechanical aspect, we designed and manufactured a model of CNC machine.

Keywords: CNC machine, digital controller, stepper motor, SolidWorks.

Remerciements

Tout d'abord, nous remercions Dieu Tout-Puissant de nous avoir donné la santé et la volonté de commencer et de terminer cet humble travail.

*Nous exprimons notre profonde gratitude à nos encadrants **Mr. Gherfi abdelhafid** et **Mr. Ziani Lotfi**, sans eux cet ouvrage ne serait pas riche en informations, nous les remercions pour la qualité de leur encadrement exceptionnel, leur patience, leur rigueur et leur disponibilité lors de la préparation de cette note.*

Nous tenons également à remercier les membres du jury pour avoir accepté l'évaluation de ce travail et pour tous leurs commentaires et critiques, ainsi que les membres du corps professoral et les professeurs du département de génie mécanique pour leur présence.

Dédicaces

Nous dédions cet humble travail à nos chères et belles mamans, chers papas qui étaient chaque jour Avec nous, je te remercie de nous avoir permis d'achever notre parcours académique et en échange du précieux, afin que nous puissions atteindre cette étape. Nous demandons à Dieu, le Puissant et Sublime, de t'accorder le paradis. Quant à nos professeurs et nos camarades étudiants, quelques mots de remerciement à vous, nous demandons à Dieu de vous récompenser en notre nom.

اهداء

نهدي هذا العمل المتواضع الى أمهاتنا العزيزات والجميلات، آبائنا
الأعزاء الذين كانوا كل يوم معنا، أشكركم على السماح لنا بإكمال
مشوارنا الدراسي وبدل الغالي والنفيس من أجل وصلنا الى هذه
المرحلة فنسأل الله عز وجل بأن يرزقكم الجنة
أما بالنسبة لأسادتنا واخوتنا الطلبة فكلمة شكر قليلة فيحققكم نسأل
الله أن يجازيكم عنا أحسن الجزاء.

Sommaire

Introduction générale	1
Historique.....	1
Chapitre 01 : généralité sur les machine à commande numérique par ordinateur.	
1 Introduction	3
2 Définition de la commande numérique	3
3 Définition d'une machine CNC	3
4 Domaine d'utilisation.	3
5 Principe de fonctionnement d'une machine CNC.....	4
6 Classification des machines CNC.....	5
7 Classification des CNC selon le mode de fonctionnement.....	5
8 Conclusion.....	7
Chapitre 02 : Arduino et moteur pas à pas.	
1 Introduction.....	9
2 La carte arduino.....	9
3 Les caractéristiques techniques de la carte Arduino	9
4 Les domaines d'utilisation.....	11
5 Les types de carte arduino.....	11
6Description de la carte Arduinouno.....	15
7 Les Avantages et les Inconvénients des cartes Arduino.....	16
8 Les moteurs pas à pas.....	16
9 Les types de moteurs pas à pas.....	17

10 Le principe de fonctionnement des moteurs pas à pas.....	26
11 Conclusion.....	26

Chapitre 03 : Conception et réalisation de la machine CNC

Partie 01 : Électronique de la machine

1 Matériels.....	28
2 LOGICIELS.....	34
3 MONTAGE.....	41
4 PARAMETRAGE.....	47

Partie 02 : Conception mécanique de la machine

1 Introduction	52
2 accouplement1.....	53
3 AXE INOX.....	54
4 ecroutrapez.....	55
5 GUIDE ARBRE 1.....	56
6 palier semelle.....	57
7 PLAQUE SUPPORT.....	58
8 RAIL 0.....	59
9 RAIL 1.....	60
10 RAIL 2.....	61
11 ROUE DE GUIDAGE.....	62
12 GUIDE VIS MER.....	63
13 POULIE.....	64
14 SUPPORT ARBRE.SLDDRW.....	65

15 SUPPORT BROCHE.....	66
16 SUPPORT MACHINE TOLE.....	67
17 SUPPORT MOTEUR RADIALE.....	68
18 SUPPORT MOTEUR RADIAL TOLE.....	69
19 SUPPORT POULIE.....	70



Liste des figures

Chapitre 01 : généralité sur les machine à commande numérique par calculateur.

Figure 01 : Décomposition d'une machine numérique [1].....	4
Figure 02 : Fonctionnement en boucle ouverte [3].....	5
Figure 03 : Fonctionnement en boucle fermée [3].....	6
Figure 04 : Fonctionnement avec commande adaptative [3].....	7
Chapitre 02 : Arduino et moteur pas à pas.	
Figure 01 :ArduinoUno avec le nom de chaque composent.....	11
Figure 02 :ArduinoUno.....	12
Figure 03 :ArduinoMega.....	13
Figure 04 :Arduino Leonardo.....	13
Figure 05 :LilyPadArduino.....	14
Figure 06 :Arduino Pro Mini.....	15
Figure 07 :Symbole générale d'un moteur pas a pas.....	17
Figure 08 :Moteur pas à pas bipolaire.....	17
Figure 09 :Moteur pas à pas unipolaire.....	18
Figure 10 :Moteur pas a pas a aimant permanente.....	18
Figure 11 :Commutation moteurs pas à pas 4 phases unipolaire.....	19
Figure 12 :Rotation d'un tour complet pour un moteur pas à pas bipolaire.....	20

Figure 13 :Fonctionnement en mode alimentation demi pas.....	21
Figure 14 :Moteur pas à pas a reluctance variable.....	23
Figure 15 : moteur hybride.....	24

Chapitre 03 : Conception et réalisation de la machine CNC

Partie 1 : Électronique de la machine

Figure 01 :ArduinoUno.....	27
Figure 02 :CNC Shield V3.....	27
Figure 03 :Moteur pas à pas NEMA 17.....	27
Figure 04 :Les DRV8825.....	28.
Figure 05 :Les A4988.....	28
Figure 06 :Alimentation 12V 5A.....	30
Figure 07 :Arduino IDE.....	31
Figure 08 :le port de communication (qui devrait apparaître).....	32
Figure 09 : "Source code (zip)"	33
Figure 10 : procédure a suivre pour inclure le dossier grbl.....	33
Figure 11 : Bibliothèque grbl qui apparaissait dans le même onglet.....	34
Figure 12 :Programmation GRBL sur Arduino.....	34
Figure 13 : La console de téléverssment	35
Figure 14 : Programme CNC JS.....	36
Figure 15 :Partie console.....	36
Figure 16 : Installation du CNC Shield sur l'Arduino.....	37
Figure 17 :Endroit du placement des cavaliers de micro-stepping.....	38
Figure 18 : Les drivers avec les radiateurs	39
Figure 19 : Mise en placeles drivers sur leur emplacement respectif, sur la CNC Shield...39	

Figure 20: Endroit d’emplacement des jumpers sur la CNC Shield, pour répliquer un axe ..40

Figure 21 : Raccordement de l'alimentation.....41

Figure 22:Connexion des moteurs au CNC Shield.....41

Partie 02 : Conception mécanique de la machine

Figure 01 :Accouplement1.....47

Figure 02: Axe inox.....48

Figure 03:Ecrou trapézoïdale.....49

Figure 04:Guide arbre 1.....50

Figure 05:Palier semelle.....51

Figure 06:Plaque support.....52

Figure 07:Rail 0.....53

Figure 08: Rail 1.....54

Figure 09: Rail 2.....55

Figure 10:Roue de guidage.....56

Figure 11 :Guide vis mer.....57

Figure 12: Poulie.....58

Figure 13:Support arbre.....59

Figure 14:Support broche.....60

Figure 15:Support machine61

Figure 16:Support moteur axe x.....62

Figure 17:Support moteur.....63

Figure 18: Support poulie.....64



Liste des tableaux

Chapitre 02 : Arduino et moteur pas à pas.

Tableau 01: séquences d'un moteur pas à pas unipolaire.....19

Tableau 02 : des séquences de commutation pour moteur pas à pas alimentation bidirectionnelle.....20

Tableau 03 : séquencesd'alimentation de moteur pas à pas en mode demi pas.....22

Chapitre 03 : Conception et réalisation de la machine CNC

Partie 01 : Électronique de la machine

Tableau 01 : le tableau de correspondance.....38



Introduction Générale

L'automatisation consiste à rendre automatique les opérations qui exigeaient auparavant l'intervention humaine, elle est considérée comme l'étape d'un progrès technique ou apparaissent des dispositifs techniques susceptibles de seconder l'homme, non seulement dans ses efforts musculaires, mais également dans son travail intellectuel de surveillance et de contrôle. Un automatisme est un sous-ensemble d'une machine destinée à remplacer l'action de l'être humain dans des tâches en générales simples et répétitives, réclamant précision et rigueur. On est passé d'un système dit manuel, à un système mécanisé, puis au système automatisé. Dans l'industrie, les automatismes sont devenus indispensables : ils permettent d'effectuer quotidiennement les tâches les plus ingrates, répétitives et dangereuses. Parfois, ces automatismes sont d'une telle rapidité et d'une telle précision, qu'ils réalisent des actions impossibles pour un être humain. L'automatisme est donc synonyme de productivité et de sécurité. Le savoir-faire de l'opérateur est transposé dans le système automatisé, il devient le PROCESSUS. Un processus peut-être considéré comme un système organisé d'activités qui utilise des ressources (personnel, équipement, matériels et machines, matière première et informations) pour transformer des éléments entrants en éléments de sortie dont le résultat final attendu est un produit. Le système automatisé de production industrielle doit également répondre à des contraintes économiques et donc à une certaine flexibilité. Il existe deux concepts technologiques : · Les automatismes séquentiels, · Les asservissements, la régulation ou « feedback control ». Cependant, dans la plupart des systèmes complexes modernes, ces deux types d'automatismes se trouvent imbriqués.

Historique

La fabrication de précision moderne exige une extrême précision dimensionnelle et la finition de surface. Une telle performance est très difficile à réaliser manuellement, voire impossible, même avec des opérateurs experts. Dans les cas où il est possible, il faut du temps beaucoup plus élevé en raison de la nécessité pour la mesure dimensionnelle fréquente pour prévenir la surexploitation. Il est donc évident que le contrôle de mouvement automatisé remplacerait le manuel de contrôle dans la fabrication moderne. Le développement de commande numérique par ordinateur (CNC) a également rendu possible l'automatisation des processus d'usinage avec souplesse pour gérer la production de petites et moyennes séries de pièces. C'était en 1947, lorsque la commande numérique est née. Tout a commencé lorsque John C. Parsons de la Parsons Corporation, Traverse City, Michigan, un fabricant de pales de rotor d'hélicoptère,

ne pouvait pas faire ses modèles assez vite. Donc, il a inventé une façon de coupler l'équipement informatique avec une pyrale du gabarit. M. Parsons a utilisé les cartes pour faire fonctionner son système de Ron chiffres de poing. 1949 a été l'année d'un autre «besoin urgent». Le commandement américain Air Matériel s'est rendu compte que ses pièces pour avions et missiles ont été de plus en plus complexe. En outre, comme les dessins ont été constamment améliorés, les changements dans le dessin ont souvent été faites. Ainsi, dans leur recherche de méthodes de production plus rapide, un contrat d'étude de la Force aérienne a été attribué à la Parsons Corporation. Le laboratoire des mécanismes d'asservissement de l'Institut de Technologie du Massachusetts (MIT) était le sous-traitant. En 1951, le MIT a repris le travail complet, et en 1952, le prototype de la machine NC d'aujourd'hui, une machine Cincinnati HydrotelMilling modifiée, a été démontrée avec succès. La commande numérique terme a été l'origine au MIT. Initialement, la technologie CNC a été appliqué sur les tours, fraiseuses, etc. qui pourrait effectuer un seul type d'opération de découpe de métal. Plus tard, on a tenté de traiter une variété de pièces qui peuvent nécessiter plusieurs types différentes opérations d'usinage et de les terminer en une seule mise en place. Ainsi centres d'usinage CNC capables d'exécuter plusieurs opérations ont été développés. Pour commencer, centres d'usinage CNC ont été développés pour les composants prismatiques usinage combinant des opérations comme fraisage, perçage, alésage et taraudage. Peu à peu, les machines pour la fabrication de composants cylindriques, appelés centres de tournage ont été développés

Chapitre 1 Généralités sur les machines CNC

1- Introduction

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

2- Définition de la commande numérique

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

Aujourd'hui, le progrès des machines à commande numérique est étroitement associé aux progrès de la microélectronique et de l'informatique. Ainsi, la CN voit ses performances augmenter régulièrement tandis que, son prix et son encombrement ne cessent de diminuer.

Cette diminution des prix, permet à la CN de pénétrer dans les petites entreprises et devient accessible à tous. Les CN actuelles sont capables d'effectuer des mouvements en combinant simultanément les positions sur plusieurs axes en même temps.

3- Définition d'une machine CNC

La machine-outil à commande numérique CNC (Computer Numerical Control en anglais) est une machine-outil dotée d'une commande numérique assurée par un ordinateur. C'est une machine totalement ou partiellement automatique à laquelle les ordres sont communiqués grâce à des codes qui sont porté sur un support matériel. Le premier rôle d'une machine CNC est de générer des mouvements, elle recevra des valeurs de positionnement de vitesse et d'accélération et générera suite à un traitement des consignes numériques en sortie [1].

4- Domaine d'utilisation

L'utilisation de la commande numérique ne se limite pas aux machines-outils travaillant par enlèvement de la matière avec des outils coupants. Elle est présente sur des installations de découpe par faisceau laser, en électroérosion que ce soit en défonçage ou en découpe par fil, en poinçonnage ou pliage de produits en feuille, pour la mise en place des composant, lors des

opérations d'assemblage,... Elle sert aussi à piloter des tables traçantes, les machines à mesurer tridimensionnelles, les robots.

La machine à outil à commande numérique représentent le moyen de production le plus important des pièces mécaniques. Elle nécessite des gestes précis et/ou répétitifs pour effectuer diverses opérations (percer, scier, rectifier, découper, fraiser, plier, graver, tarauder, souder, visser, déposer un matériau). Les matériaux qui peuvent être usinés sont très divers, la caractéristique principale qui les différencie est la dureté. Plus le matériau est dur, plus il faudra que la CNC, ainsi que l'outil qui l'équipe aient la qualité et la puissance nécessaires pour usiner le matériau. D'où de nombreux dispositifs CNC différent entre eux principalement par l'outil qui est utilisé.

Les machines CNC sont employées dans de nombreux secteurs industriels : métallurgies, bois, textiles... Elles sont associées à des nouvelles technologies ; laser, électroérosion, jet d'eau.

5- Principe de fonctionnement d'une machine CNC

Les machines à commande numérique se composent de deux parties complémentaires (figure 1) [2]:

- La partie opérative.
- La partie commande.

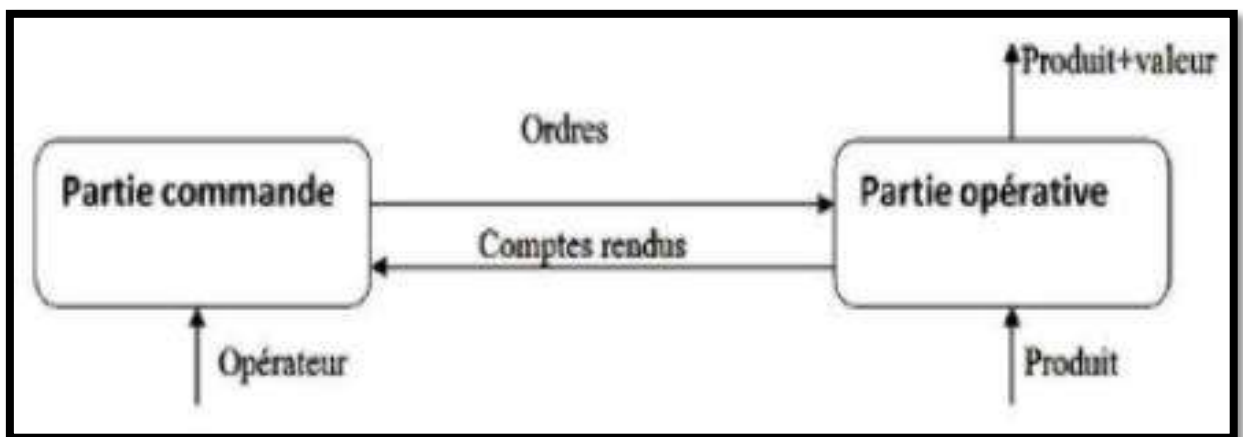


Figure 1 : Décomposition d'une machine numérique [1].

La partie commande est composée d'un calculateur ou d'un ordinateur et d'éléments électroniques capables de piloter les moteurs, cette partie permet de piloter la partie opérative.

La partie opérative comporte des axes de déplacement, la tête de l'outil et les actionneurs (les moteurs).

Des ordres sont générés vers la commande par le biais d'un code machine ou par action manuelle de l'opérateur. La commande va traiter ces informations et générer des consignes afin d'obtenir les déplacements voulus par le biais des moteurs d'axes. Des contrôles de vitesse et de position seront alors effectués de manière continue par la machine.

6- Classification des machines CNC

Les machines à commande numérique sont classées selon :

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

7- Classification des CNC selon le mode de fonctionnement

Selon le mode de fonctionnement les machines CNC sont classées en trois catégories

- fonctionnement en boucle ouverte
- fonctionnement en boucle fermée
- fonctionnement avec commande adaptative

a) Fonctionnement en boucle ouverte

En boucle ouverte (figure 2), le système assure le déplacement du chariot mais ne le contrôle pas.

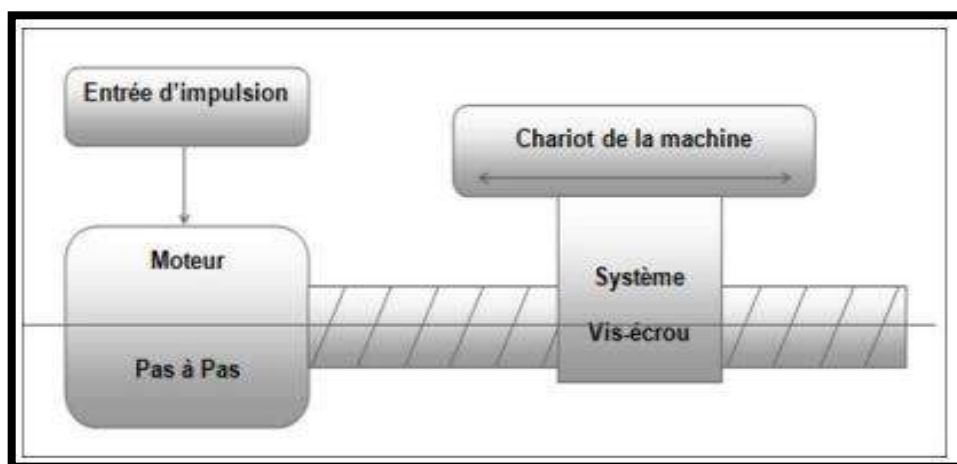


Figure 2 : Fonctionnement en boucle ouverte [3].

b) Fonctionnement en boucle fermé

En boucle fermée (figure 3) le système contrôle le déplacement ou la position jusqu'à égalité des grandeurs entrée (E) désirée et celui mesuré (Gm).

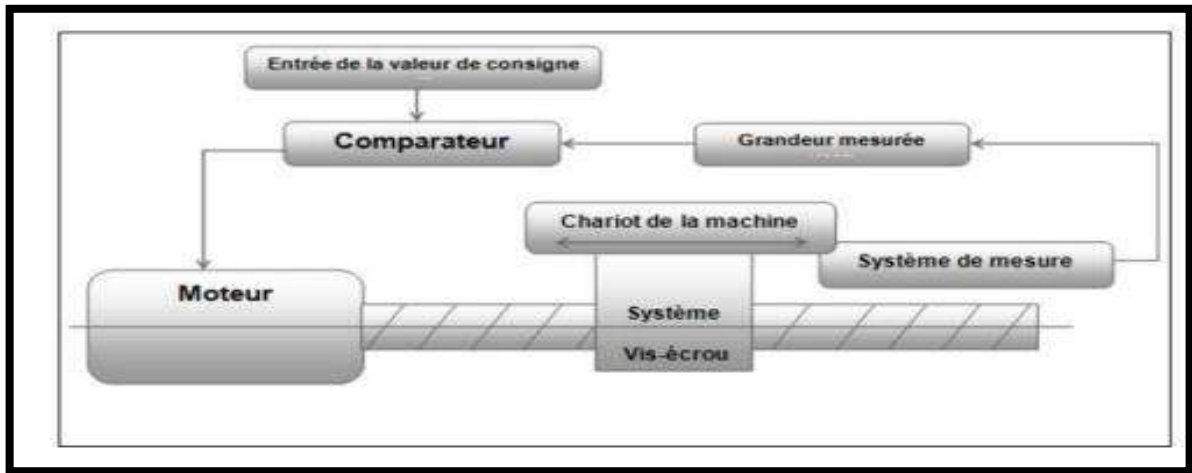


Figure 3 : Fonctionnement en boucle fermée [3].

c) Fonctionnement avec commande adaptative

La commande adaptative (figure 4) réalise d'une façon continue et automatique l'adaptation des conditions de coupe. Des capteurs relèvent les valeurs de couple de la broche, l'amplitude de vibration de la broche, la température au point de coupe. Ces informations sont transmises à une unité spéciale qui les envoie vers le directeur de commande numérique qui agit selon l'analyse des informations sur les conditions de coupe pour permettre une meilleure qualité de travail, une meilleure productivité et une plus grande sécurité.

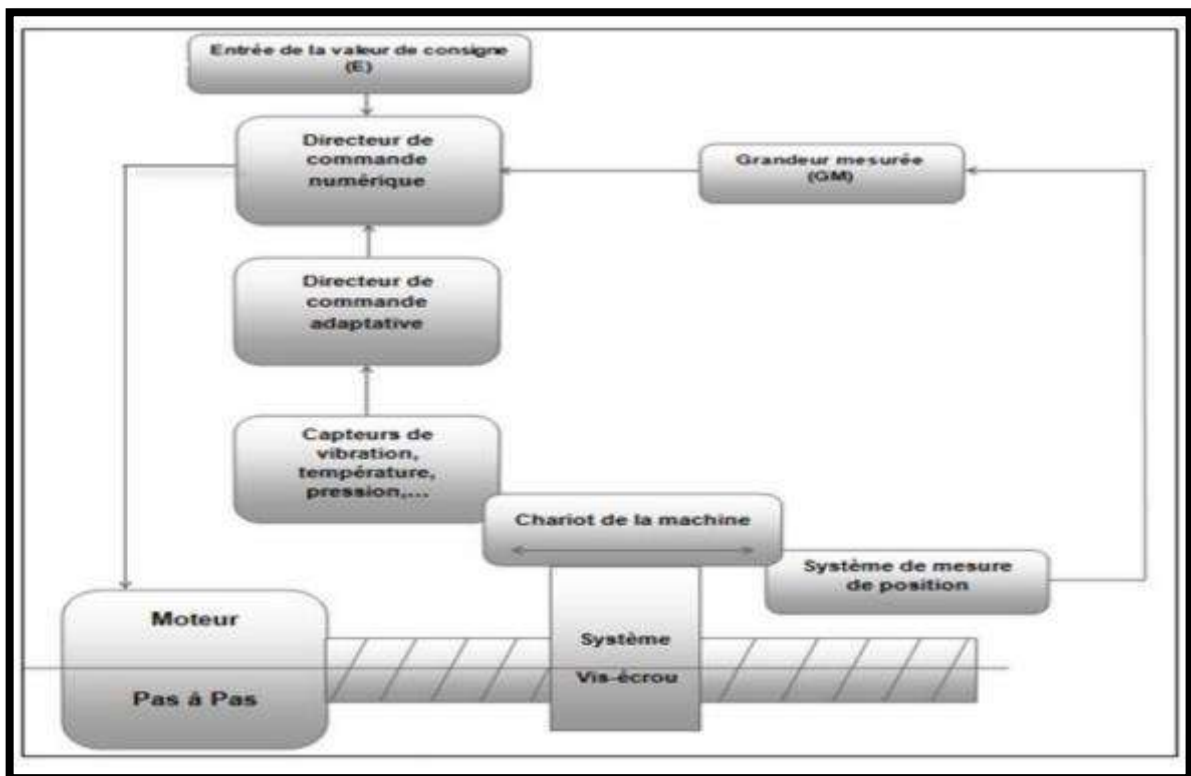


Figure 4 : Fonctionnement avec commande adaptative [3].

8- Conclusion

Dans ce chapitre on a décortiqué d'une manière générale une machine à commande numérique et montré l'intérêt de chaque partie qui constitue cette dernière.

On a cité les types de commandes les plus utilisées et parlé de la Classification des machines CNC.

Nous pouvons ainsi entamer la prochaine étape qui consiste à présenter la phase de conception

Chapitre 2

Arduino et moteur pas
à pas

1- Introduction

Les moteurs pas à pas et Arduino font partie intégrante de notre projet, car ils sont le pilier pour démarrer ce projet. Dans ce chapitre, nous discuterons plus en détail de leurs différentes caractéristiques et avantages.

2- La carte Arduino

La carte Arduino est une carte électronique basée autour d'un microcontrôleur et de composants minimum pour réaliser des fonctions plus ou moins évoluées à bas coût. Elle possède une interface USB pour la programmer. C'est une plateforme open-source qui est basée sur une simple carte à microcontrôleur (de la famille AVR), et un logiciel, véritable environnement de développement intégré, pour écrire, compiler et transférer le programme vers la carte à microcontrôleur.

Arduino est peut être utilisé pour développer des applications matérielles industrielles légères ou des objets interactifs, et il peut recevoir en entrées une très grande variété de capteurs. Les projets Arduino peuvent être autonomes, ou communiquer avec des logiciels sur un ordinateur (Flash, Processing ou MaxMSP, Labview). Les cartes électroniques peuvent être fabriquées manuellement ou bien être achetées pré assemblées, le logiciel de développement open-source est téléchargé gratuitement. [4]

La programmation de la carte Arduino présente les principales fonctionnalités de l'interface de l'application Arduino. L'application Arduino vous permet de créer et éditer un programme (appelé sketch) qui sera compilé puis télé-versé sur la carte Arduino. Ainsi, lorsque vous apportez des changements sur le code, ces changements ne seront effectifs qu'une fois le programme télé-versé sur la carte. Il existe plusieurs types de cartes (Uno, Méga, Similuno... etc). Dans notre projet on utilise carte Arduino Uno. [4]

3- Les caractéristiques techniques de la carte Arduino

- Version: Rev. 3
- Alimentation:
- Via port USB ou, 7 à 12 V sur connecteur alim

- Microprocesseur: ATmega328
- Mémoire flash: 32 kB
- Mémoire SRAM: 2 kB
- Mémoire EEPROM: 1 kB
- 14 broches d'E/S dont 6 PWM
- 6 entrées analogiques 10 bits
- Intensité par E/S: 40 mA
- Cadencement: 16 MHz
- Bus série, I2C et SPI
- Gestion des interruptions
- Fiche USB B
- Dimensions: 74 x 53 x 15 mm [5]

La carte Arduino Uno dispose de toute une série de facilités pour communiquer avec un ordinateur, une autre carte Arduino, ou avec d'autres microcontrôleurs. L'Arduino Uno dispose de quatre UARTs (Universal Asynchrones Receiver Transmitter ou émetteur récepteur asynchrone universel en français) pour une communication en série de niveau TTL (5V) et qui est disponible sur les broches 0 (RX) et 1 (TX). Un circuit intégré Atmega16U2 sur la carte assure la connexion entre cette communication série de l'un des ports série de l'Atmega Uno vers le port USB de l'ordinateur qui apparaît comme un port COM virtuel pour les logiciels de l'ordinateur. Le code utilisé pour programmer l'Atmega8U2 utilise le driver standard USB COM, et aucun autre driver externe n'est pas nécessaire. [6]

La Figure : Représente la carte Arduino Uno et le nom de chaque composant électronique sur cette carte

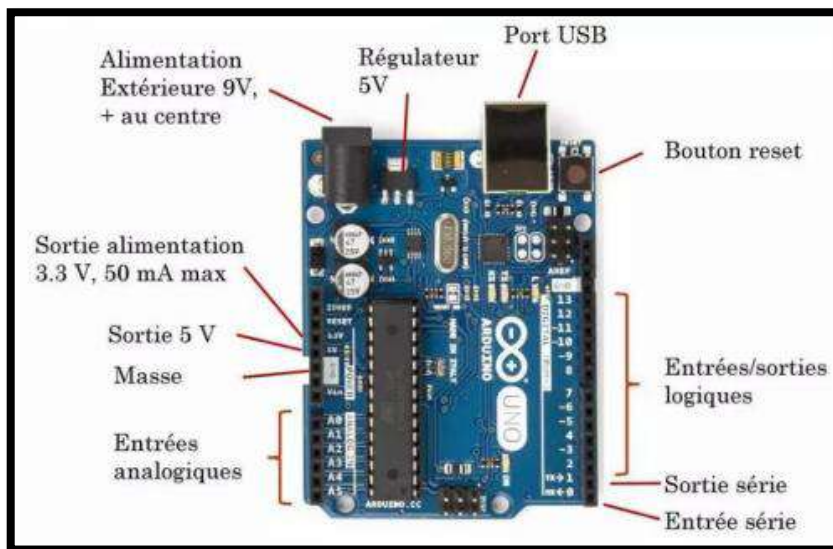


Figure 1 : Arduino Uno avec le nom de chaque composent

4- Les domaines d'utilisation

Au fil des ans, l'Arduino a été le cerveau de milliers de projets et d'objets du quotidien à des outils scientifiques complexes. Une communauté mondiale de créateurs - étudiants, amateurs, artistes, programmeurs et professionnels - s'est réunie autour de cette plateforme open source, et leurs contributions ont ajouté une quantité incroyable de connaissances accessibles qui peuvent être extrêmement utiles pour les débutants comme pour les experts. À l'Ivrea Interaction Design Institute (Ivrea Interaction Design Institute). Comme un outil simple de prototypage rapide, destiné aux étudiants sans formation en électronique et en programmation. Une fois dans une communauté plus large, la carte Arduino a commencé à changer pour s'adapter aux nouveaux besoins et défis et différencier son offre des simples cartes 8 bits aux produits pour les applications IoT, l'impression portable, l'impression 3D et les environnements embarqués. Open source, permettant aux utilisateurs de les construire en toute autonomie. Ils sont en solo et enfin adaptés à leurs propres besoins.

5- Les types de carte Arduino

Il existe de nombreux types d'Arduino, et chacun d'eux a des caractéristiques et des capacités qui le distinguent des autres, de sorte que chaque pièce peut être utilisée pour des choses différentes des autres, notamment :

a) Arduino Uno

C'est l'une des premières cartes Arduino les plus couramment utilisées car elle est considérée comme la plus facile à enseigner car il s'agit d'un petit circuit électronique en plus de fournir des ports pour connecter les composants électroniques au contrôleur directement via un port d'entrée et de sortie numérique. La chose la plus importante qui distingue ce type de carte Arduino est que la puce du contrôleur n'est pas fixée dans la carte, mais plutôt installée sur un support, et cette caractéristique en fait le meilleur choix pour les débutants en particulier, de sorte que si vous brûlez accidentellement la puce tout en travaillant sur votre projet, vous pouvez restaurer votre travail sur la carte en remplaçant la puce de contrôleur d'origine par une autre du même modèle.

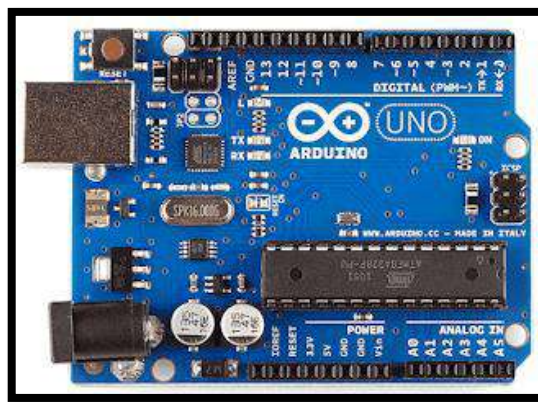


Figure 2: ArduinoUno

b) Arduino Mega

Le prochain type de carte Arduino s'appelle l'Arduino Mega. C'est une carte Arduino qui contient le plus grand nombre de ports et est la plus grande d'entre eux. Ce type de carte est conçu pour s'adapter à des projets plus complexes tels que l'impression 3D.

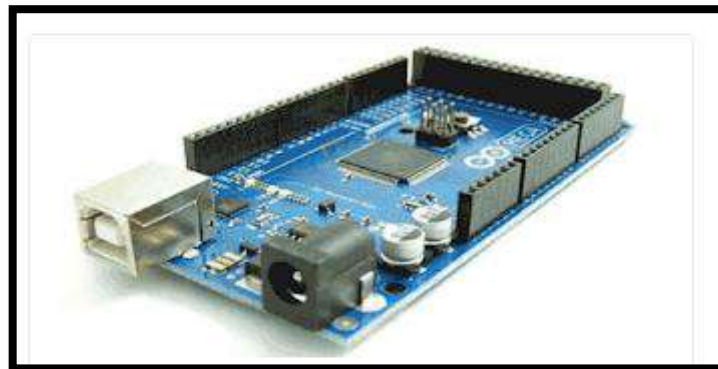


Figure 3 :ArduinoMega

c) **Arduino Leonardo**

Il s'agit d'une carte Arduino appelée "Arduino Leonardo." C'est un type amélioré du type précédent de carte Arduino, et elle s'appelait la première carte développée de cartes Arduino, qui a une caractéristique unique, qui est qu'elle contient un port USB interne port pour USB contrairement à d'autres de ses cartes sœurs, ce qui élimine le besoin d'utiliser un coprocesseur. Cette fonctionnalité permet à la carte lorsqu'elle est connectée à votre appareil d'apparaître comme une souris et un clavier, et c'est ce qui la rend idéale pour créer diverses applications qui vous permettent de contrôler votre propre ordinateur via la souris et le clavier.

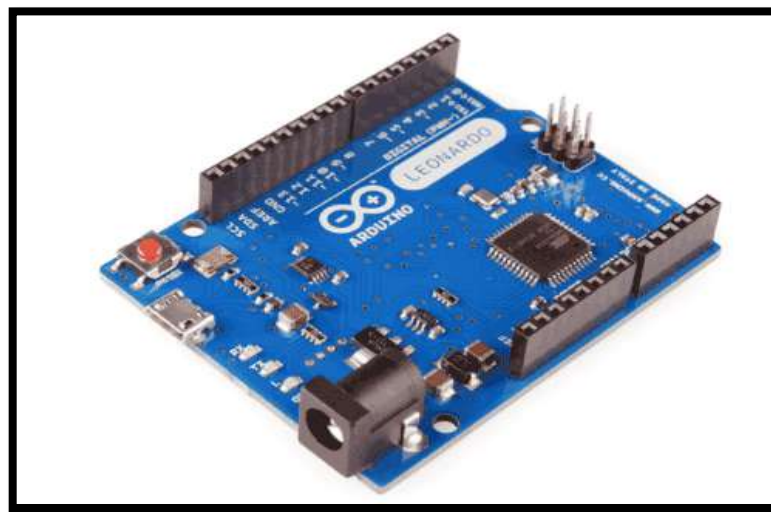


Figure 4 :Arduino Leonardo

d) **Lily Pad Arduino**

Arduino Lily Pad vous permet d'interagir avec plus de détails en fonction de ce que vous programmez. Cette version de la carte Arduino est principalement utilisée avec une technologie portable, car elle est conçue de manière à être installée facilement dans les tissus. Il existe également des capteurs et des composants spécialement conçus pour cette carte et un fil électriquement conducteur est utilisé à la place des fils ordinaires. Le Lily Pad Arduino fonctionne sur des tensions relativement basses, et il devient un outil électronique efficace lorsqu'il est connecté à des capteurs et autres actionneurs mécaniques tels que des moteurs en général, sans laisser de charge électrique sur les outils associés.

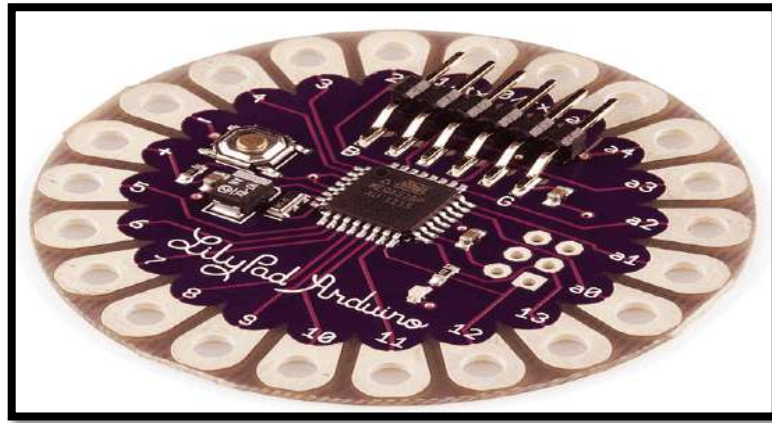


Figure 5 :Lily Pad Arduino

e) **Arduino Pro Mini**

La carte Arduino Pro Mini est une carte moins chère que la carte Arduino. Vous avez besoin d'une petite carte et de quelques connexions pour votre projet. Par exemple, vous souhaitez fabriquer des téléphones intelligents, des montres intelligentes, ou développer des applications en implantant des capteurs à distance. Ainsi, vous pouvez utiliser l'Arduino Mini. Ils sont conçus pour être placés de manière semi-permanente dans des projets comme celui-ci. Mais en raison de la petite taille de la carte Arduino Pro Mini, les façons de la connecter et de la gérer sont plus complexes et difficiles que les autres cartes Arduino, et elle ne convient donc pas aux débutants, car la puce du contrôleur est installée à l'intérieur. Avec la technologie d'installation en surface des éléments électroniques, donc si la puce est brûlée, la carte est définitivement perdue.

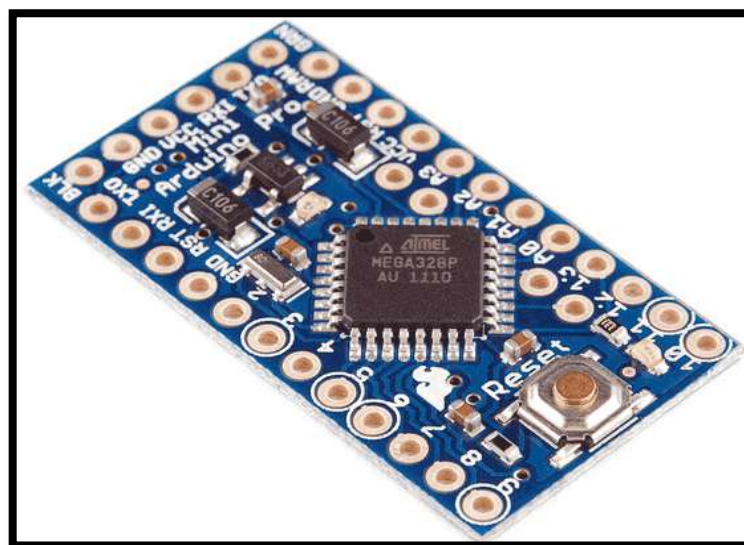


Figure 6 : Arduino Pro Mini

6- Description de la carte Arduino Uno

Le logiciel de programmation des modules Arduino est une application Java, libre et multiplateformes, serve d'éditeur de code et de compilateur, et qui peut transférer le firm ware et le programme à travers la liaison série (RS232, Bluetooth ou USB selon le module).

Il est également possible de se passer de l'interface Arduino, et de compiler les programmes en ligne de commande. Le langage de programmation utilisé est le C++, compilé avec avrg++, et lié à la bibliothèque de développement Arduino, permettant l'utilisation de la carte et de ses entrées/sorties.

La mise en place de ce langage standard rend aisé le développement de programme sur les plates-formes Arduino, à toute personne qui maîtrise le C ou le C++.

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

- Pouvoir écrire et compiler des programmes pour la carte Arduino.
- Se connecter avec la carte Arduino pour y transférer le fichier exécutable.
- Communiquer avec la carte Arduino

Le logiciel Arduino intègre également un TERMINAL SERIE (fenêtre séparée) à l'équipement d'afficher des messages textes reçus de la carte Arduino et d'envoyer des caractères vers la carte Arduino.

Cette fonctionnalité permet une mise au point facilite des programmes, permet d'afficher sur l'ordinateur l'état de variables, de résultats de calculs ou de conversions analogique-numérique : un élément essentiel pour améliorer, tester et corriger ses programmes. [7]

7- Les Avantages et les Inconvénients des cartes Arduino

Ce qui distingue les cartes Arduino des autres cartes de développement, c'est la facilité de prise en main et la simplicité du langage. Certains peuvent penser que l'Arduino a été conçu pour les amateurs, mais au contraire, il est disponible pour tous les publics, des amateurs aux plus avertis. Plus grands projets développés. Quant à l'Arduino, il a des avantages, il a aussi des inconvénients. La capacité de programmation de l'Arduino est bien inférieure à celle de l'ordinateur, et non Il peut exécuter des programmes tels que Windows ou Android, ou exécuter des écrans haute définition, etc. Ces œuvres conviennent à un ordinateur ou à un Raspberry Pi, dont nous parlerons peut-être dans un autre livre. Le courant que l'Arduino peut

obtenir des ports = 20mA, ce qui n'est généralement pas suffisant pour alimenter un moteur ou un relais. Par conséquent, des éléments électroniques doivent être utilisés pour amplifier l'énergie électrique dans certaines applications.

8- Les moteurs pas à pas

Le principe des moteurs pas à pas est connu depuis longtemps. Mais son développement a commencé dans les années 1960 grâce à l'avènement de l'électronique numérique. Un moteur pas à pas est une machine tournante dont le rotor se déplace d'un angle initial (α) appelé un pas à chaque fois que le circuit de commande effectue une impulsion de courant dans une ou plusieurs bobines. Cela signifie qu'il ne nécessite pas de chaîne de retour pour vérifier les résultats qui correspondent exactement aux commandes données, à condition que certaines limites de fonctionnement soient respectées.

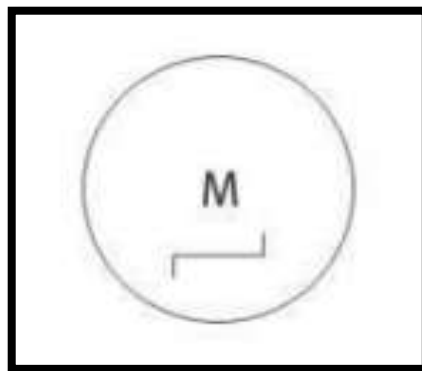


Figure 7: symbole générale d'un moteur pas à pas

9- Les types de moteurs pas à pas

Tous les moteurs pas-à-pas comprennent un stator portant des bobines dans lesquelles, le courant est commuté par l'électronique de commande. Par contre, les différences apparaissent au niveau du rotor. Selon l'alimentation des bobines, on distingue deux types :

a) Moteurs bipolaires

Les phases d'un moteur bipolaire sont alimentées une fois dans un sens et une autre fois dans le sens inverse. Ils créent une fois un pôle nord et l'autre fois un pôle sud d'où le nom de bipolaire.

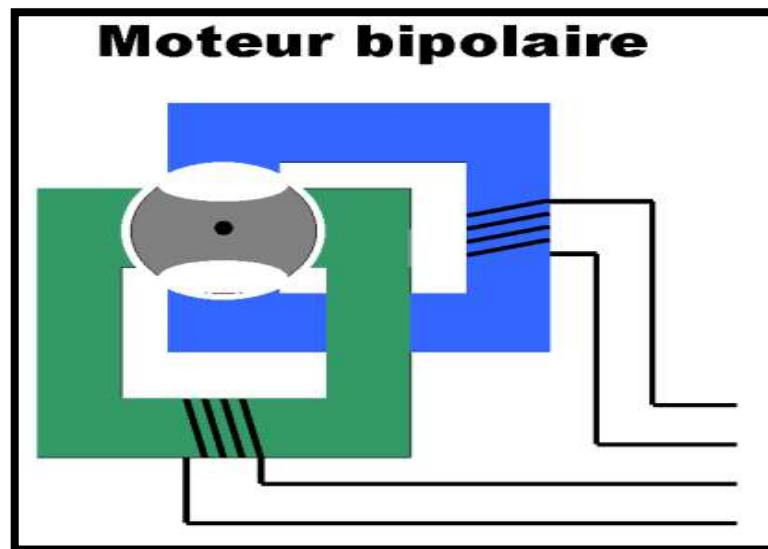


Figure 8 : moteur pas à pas bipolaire

b) Moteurs unipolaires

Les phases sont alimentées dans un seul sens

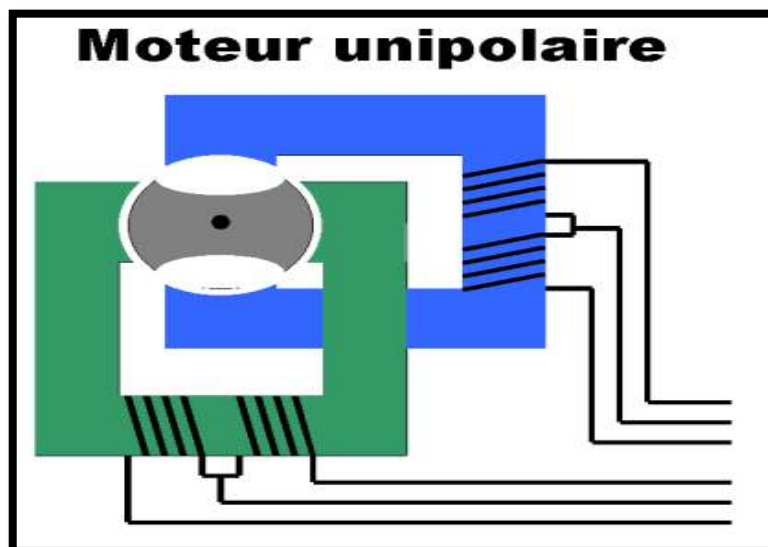


Figure 9 : moteur pas à pas unipolaire

Généralement les moteurs bipolaires possèdent 2 phases et les moteurs unipolaires possèdent 4 phases. Selon la conception on distingue 3 types :

- a) moteurs à aimant permanent
- b) moteurs à reluctance variable
- c) moteurs hybrides

a) moteurs à aimant permanent

Principe :

Le rotor porte des aimant permanents et considérons le stator comporte quatre bobines et le rotor est un aimant bipolaire

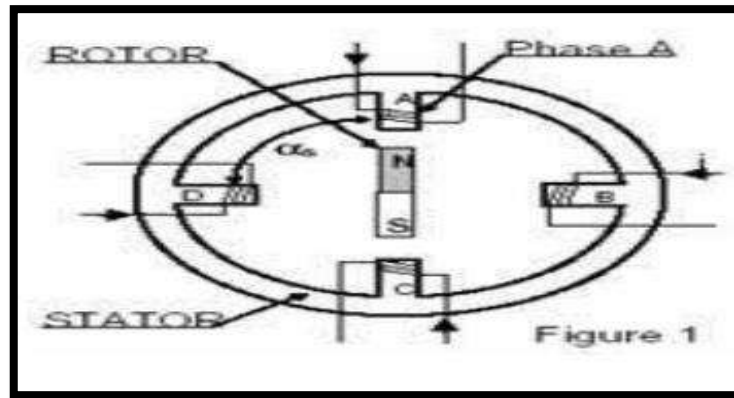


Figure 10 : moteur pas à pas à aimant permanente

En l'absence de courant électrique dans les enroulements, le rotor se positionne dans une position d'équilibre stable, sa paire de phase est en face de la paire de plots du rotor. Pour écarter le rotor de sa position de repos, il faut exercer un couple appelé couple de détente. On peut alimenter les quatre bobines de diverses manières. La plus simple est d'imposer un courant, toujours de même sens, dans une seule bobine à la fois. Les positions d'équilibre stable correspondent donne à l'axe magnétique du rotor en face de l'axe des bobines, pôle nord associe à pôle sud. Il y a ainsi quatre positions d'équilibre. Pour obtenir une rotation d'un tour, il faut effectuer quatre commutations.

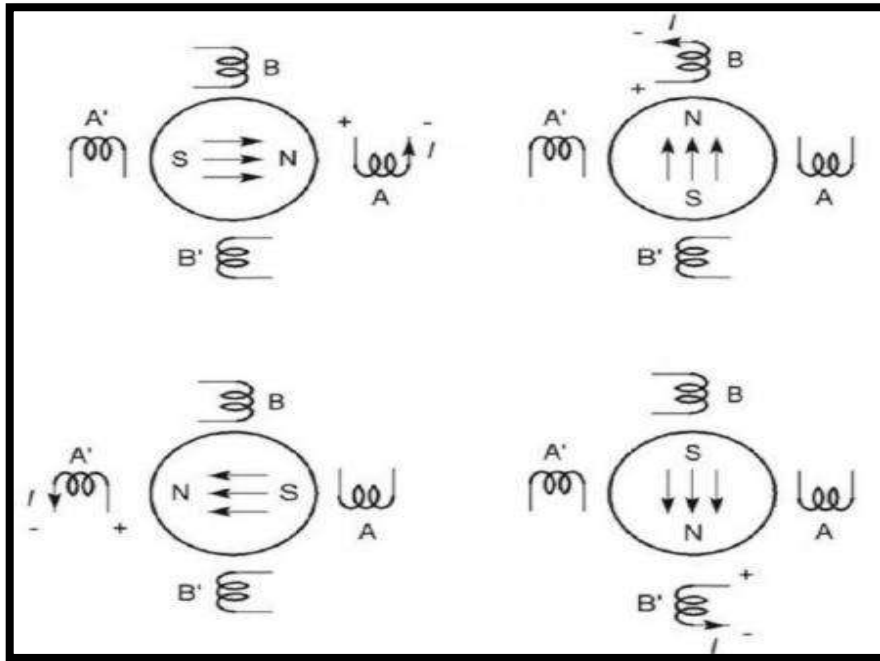


Figure 11 :Commutation moteurs pas à pas 4 phases unipolaire

Position du rotor	Courants dans les phases			
	A	B	A'	B'
1	I	0	0	0
2	0	I	0	0
3	0	0	I	0
4	0	0	0	I

Tableau 1: séquences d'un moteur pas à pas unipolaire

L'angle de déplacement du rotor d'une phase à l'autre appelé le pas dans notre cas le moteur se fait 4 pas pour un tour complet. Donc, le moteur se déplace par un angle de 90° pour faire un autre tour, il suffit de répéter les impulsions électriques des phases successivement. Donc, on peut asservir la position de déplacement d'un moteur pas à pas à l'aide des impulsions électriques vers les bobines de ce moteur.

- ✓ Pour inverser le sens de rotation de moteur, il suffit d'inverser l'ordre d'alimentation des phases successivement de 4 à 1.
- ✓ Pour une alimentation bidirectionnelle, on couple deux phases à la fois et on applique une alimentation inversé dans les enroulements.
- ✓ Pour obtenir un tour complet de rotor, il faut quatre commutations la figure 8 illustre le fonctionnement :

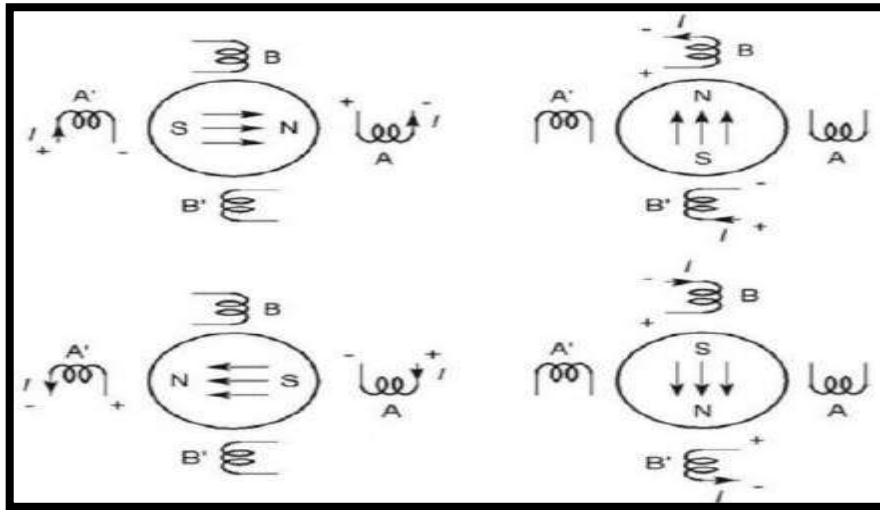


Figure 12 : rotation d'un tour complet pour un moteur pas à pas bipolaire

Position du rotor	Courants dans les phases	
	AA'	BB'
1	I	0
2	0	I
3	$-I$	0
4	0	$-I$

Tableau 2 : des séquences de commutation pour moteur pas à pas alimentation bidirectionnelle

Le pas est toujours 90° , ce mode est appelé pas complet pour améliorer la résolution, on alimente deux phases à la fois et ce mode d'alimentation est appelé demi pas.

Il est possible de se servir des deux types de positions d'équilibre, celles dans l'axe des plots et celles sur les bissectrices, en alimentant alternativement une phase puis deux : on parle alors de fonctionnement en demi pas (figure 9) et (table 3) précise la séquence de commutation dans ce cas.

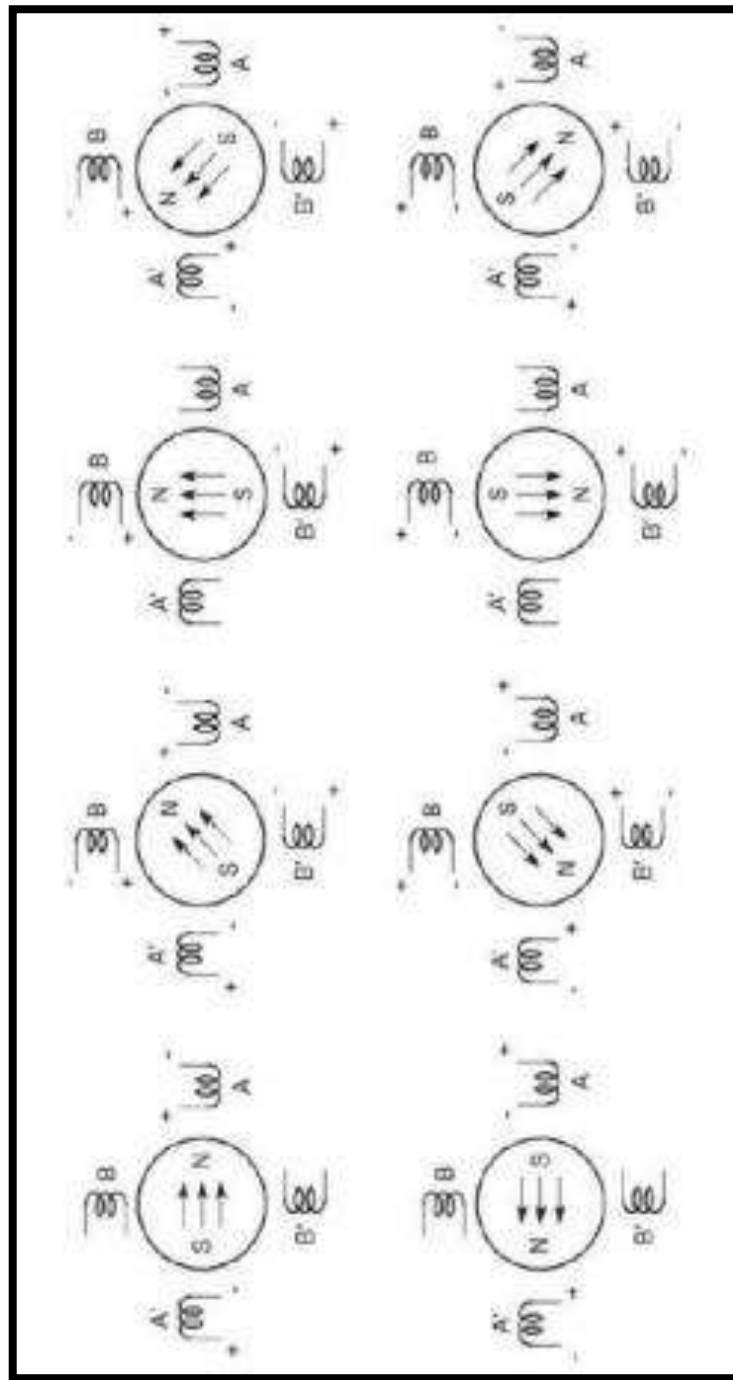


Figure 13 : fonctionnement en mode alimentation demi pas

Position du rotor	Courants dans les phases	
	AA'	BB'
1	I	0
2	I	I
3	0	I
4	$-I$	I
5	$-I$	0
6	$-I$	$-I$
7	0	$-I$
8	I	$-I$

Table 3 : Séquences d'alimentation de moteur pas à pas en mode demi pas .

Cette fois l'angle de pas est 45° donc 8 pas par tour pour plus de précision, on peut alimenter le moteur en mode micro pas grâce aux pilotes intégrés (drivers), on les parle dans le chapitre 3 dans la partie commande

b) Moteurs à reluctance variable

Le moteur pas-à-pas réductant (Variable reluctance motor VR) comporte nb bobines distinctes au stator et un rotor ferromagnétique sans aimant ($n_a=0$). Sans courant dans les bobinages statoriques, ce type de moteur ne présente pas de couple réductant. Les circuits magnétiques du rotor et du stator sont assemblés à partir de tôles magnétiques de haute perméabilité (fer-silicium ou même fer-cobalt). Certaines machines, destinées au positionnement ou à une rotation très lente, peuvent avoir un rotor ou/et un stator en fer massif. Il existe un grand nombre d'astuces de construction pour augmenter le nombre de pas par tour, comme par exemple les structures bifilaires qui ne seront pas étudiées dans cette recherche. Pour minimiser la force magnétomotrice et augmenter la puissance massique, l'entrefer des moteurs pas à pas à reluctance variable doit être aussi petit que possible. Plusieurs géométries du rotor et du stator ont été ainsi imaginées par les constructeurs pour remplir des cahiers des charges particuliers. On distingue trois géométries de base, à savoir :

- Les moteurs pas à pas à reluctance variable à plots statoriques non dentés

- Les moteurs pas à pas à réluctance variable à plots statoriques dentés
- Les moteurs pas à pas à réluctance variable à plots statoriques non dentés décalés

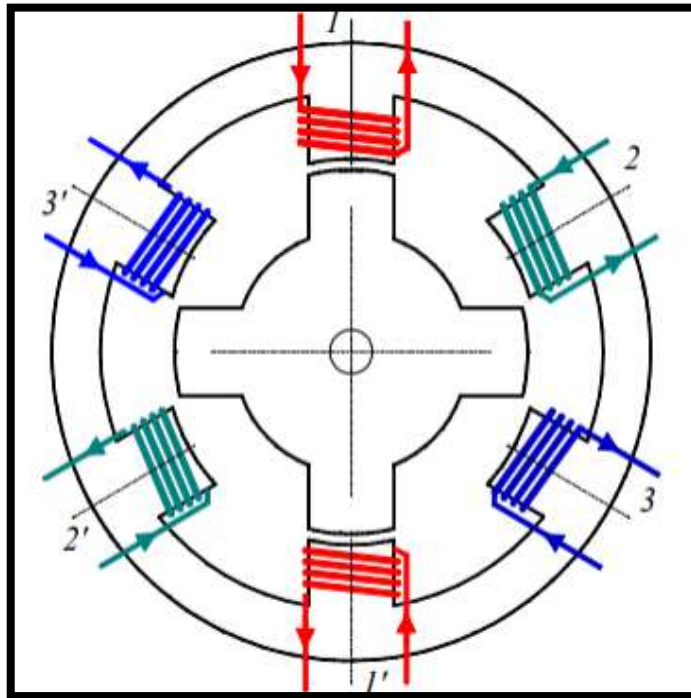


Figure 14 : moteur pas à pas à réluctance variable

c) Moteurs hybrides

En combinant les structures des deux moteurs précédents, c'est-à-dire en plaçant les aimants du moteur à aimants permanents dans un circuit ferromagnétique, on crée un nouveau type de moteur appelé moteur réluctant polarisé ou moteur hybride (Hybridmotor : HB). Dans ce cas, il existe un couple réluctant provoqué par la variation de perméances propres associées à chaque aimant et à chaque bobine.

- p : nombre de pôles magnétiques vu par le stator
- N_{dr} Nombre de dents sur une couronne rotorique
- m : nombre de phases (doit être pair)

L'augmentation du nombre de plots statoriques alimentés simultanément permet d'augmenter le nombre de dents du rotor, et donc de diminuer le pas angulaire du rotor. Le même résultat s'obtient par la subdivision des plots en plusieurs dents. Les moteurs pas à pas hybrides comptent parmi les moteurs pas à pas les plus fabriqués. Ils existent aussi bien en structure aux circuits simples (single stack), qu'en structure multiple (multi stack).

Dans presque tous les cas, les plots dentés du stator présentent le même pas que les dents aimantées du rotor L.

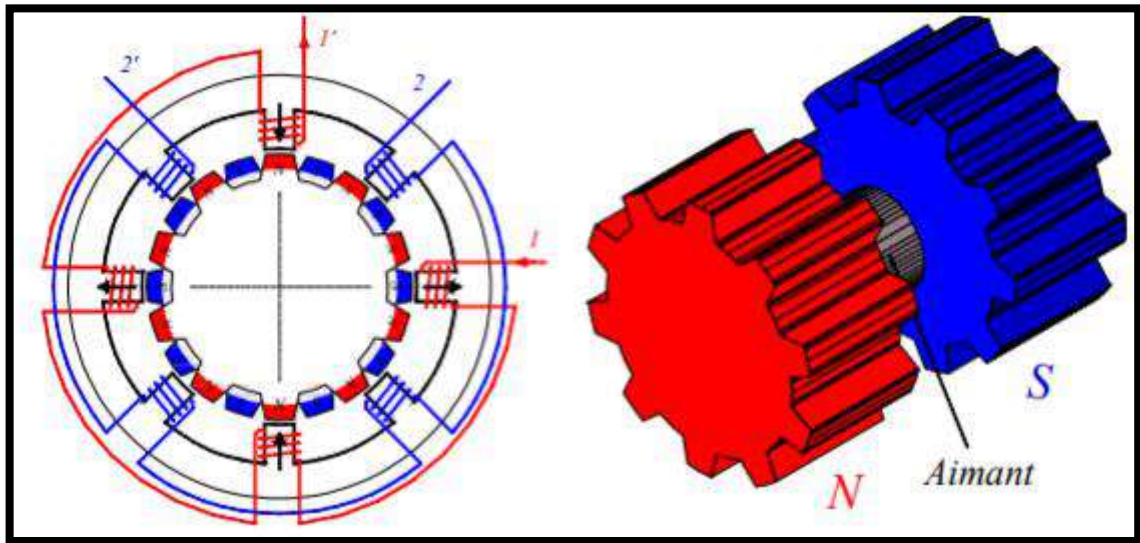


Figure 15 : moteur hybride

10- Le principe de fonctionnement des moteurs pas à pas

Lorsque la tension est appliquée aux bornes, les balais du moteur électrique démarrent et commencent à tourner constamment. Le ralenti du moteur à une propriété spéciale, c'est la conversion des impulsions rectangulaires entrantes en une position prédéterminée de l'arbre d'entraînement appliqué. L'arbre descend d'un angle constant à chaque poussée. S'il y a plusieurs électroaimants dentés autour de la pièce maîtresse du fer denté, alors les appareils avec une telle boîte de vitesses sont très efficaces. Le microcontrôleur active l'électroaimant. Un électroaimant à un seul engrenage sous l'influence de l'énergie attire les dents de la roue dentée à sa surface et, par conséquent, l'arbre du moteur tourne. Lorsque les dents sont alignées avec l'électroaimant, elles se déplacent légèrement vers la pièce magnétique adjacente. Pour que l'engrenage commence à tourner et à s'aligner, en utilisant la roue précédente, le premier électroaimant est éteint et l'aimant suivant est allumé. Ensuite, tout le processus est répété autant de fois que nécessaire. Cette rotation s'appelle un pas stationnaire. En calculant le nombre de pas à pleine rotation du moteur, la vitesse de sa rotation est déterminée.

11- Conclusion

Nous concluons que la carte Arduino est ce qui nous facilite la réalisation de tout projet, que le responsable soit un expert ou un amateur. Et le moteur pas à pas est l'élément responsable du déplacement des axes de la machine CNC grâce à ces caractéristiques de rotation au moyen d'une simple impulsion électrique. Il nous offre également la facilité de le localiser sans passer par une chaîne de retour en plus de la possibilité de contrôle de la vitesse.

***Chapitre 3 : Conception et
réalisation de la machine
CNC***

Partie 1 : Électronique de la machine

1- Matériels

Pour l'électronique de la machine que nous avons présenté, on va se baser sur une carte Arduino Uno figure 1 (à droite) qui est simple à prendre en main, équipé d'une CNC Shield V3 de chez Protoneer figure 2 (à gauche) :

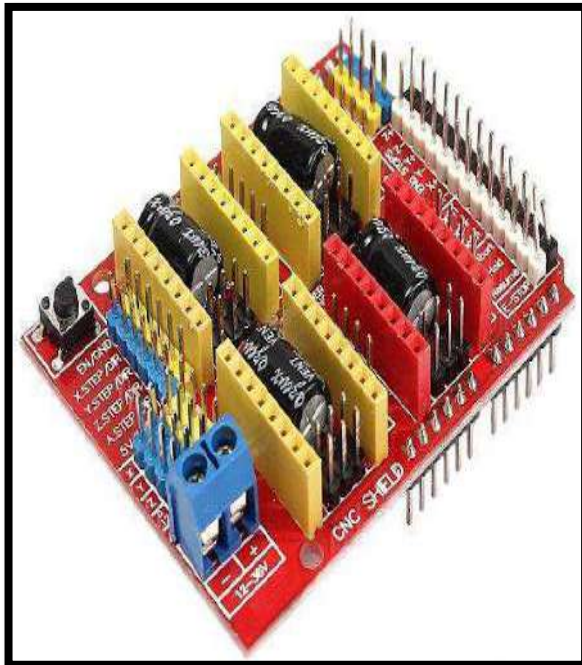


Figure 2 :CNC Shield V3

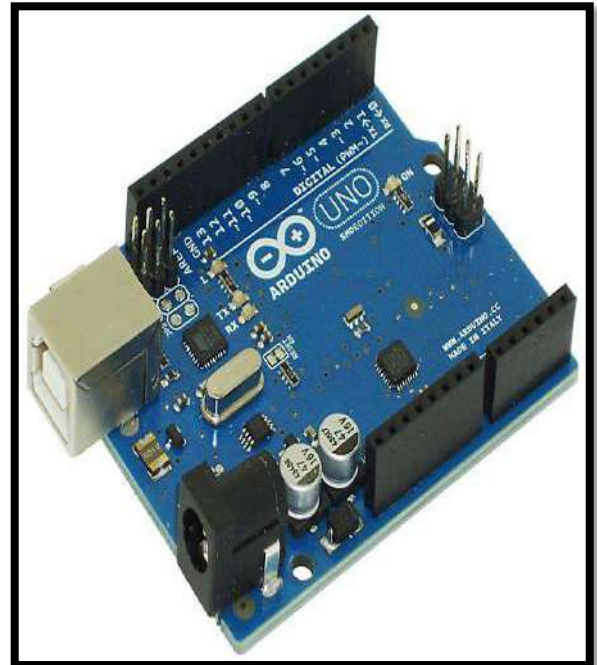


Figure 1 :ArduinoUno.

Pour la motorisation nous allons utiliser des moteurs Nema17 figure 3, qui sont des moteurs pas à pas, 200 pas par révolution. Ce sont des moteurs utilisés dans 99% des imprimantes 3D et CNC grand public qu'on peut trouver un peu partout.

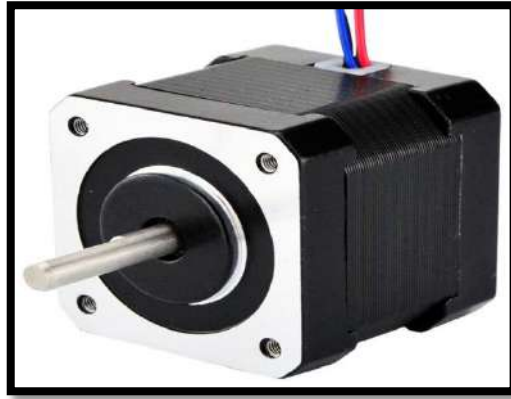


Figure 3 : moteur pas à pas NEMA 17

Les moteurs pas à pas ne sont pas des moteurs classiques que l'on branche sur une alimentation et qui se mettent à tourner comme par magie.

Ils effectuent une série "de cran", de pas, pour tourner, ce qui permet de contrôler précisément leur vitesse, leur sens de rotation et surtout leur position.

Dans notre cas les Nema17 font 200 pas pour faire un tour complet. Ce qui signifie que l'on peut contrôler leur position à $360 / 200 = 1.8^\circ$ près.

Pour leur faire un pas, il faut leur envoyer une impulsion électrique. C'est à dire que pour les faire tourner, il faut leur envoyer une série d'impulsions :

- En contrôlant le **nombre** d'impulsion qu'on envoi, on contrôle leur **position**
- En contrôlant la **fréquence** des impulsions, on contrôle leur **vitesse**

Vous devez vous dire que ça semble plus compliqué que prévu pour piloter un moteur pas à pas, mais en fait non. Il existe un composant électronique qui va gérer pour nous l'envoi des impulsions au moteur. Ce composant est appelé « driver moteur ».

Le driver moteur va réceptionner les ordres de vitesse, de sens de rotation et de position provenant de l'électronique de commande, dans notre cas l'Arduino peut transformer ces ordres en série d'impulsions électriques qu'il va envoyer au moteur.

Il existe 2 références de drivers moteurs HYPER connus :

Les DRV8825 (à gauche) figure 4, ainsi que les A4988 (à droite) figure 5

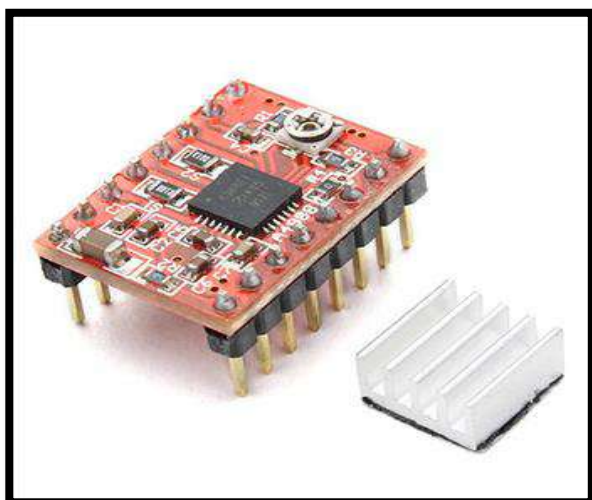


Figure 4 :Les DRV8825

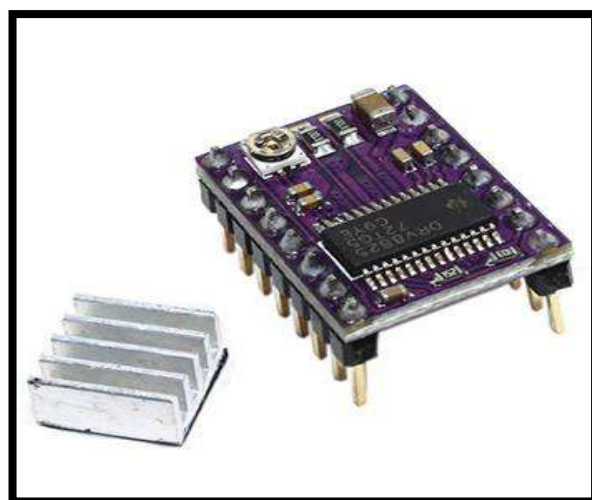


Figure 5 : Les DRV A4988

Ces deux drivers moteurs sont protégés contre les surchauffes (dans une certaine limite évidemment) mais ils ont 2 grosses différences importantes :

- le courant maximal que peut envoyer le driver au moteur
- le nombre de divisions de "micro-stepping"

Voici les valeurs pour ses deux drivers :

-RV8825 : Courant admis jusqu'à 1.5A **sans** refroidissement, 2.2A **avec** un refroidissement **très** efficace (radiateur + ventilateur). Division maximale de 1/32 pour le micro-stepping.

- A4988 : Courant admis 1.2A **avec** refroidissement, et 2A **avec** un refroidissement efficace (radiateur). Division maximale de 1/16 pour le micro-stepping.

Un courant plus élevé va permettre de mettre des moteurs plus gros donc plus gourmands et/ou pouvoir faire forcer les moteurs un peu plus. Ici le DRV8825 accepte plus de courant donc il permet plus de marge pour le choix des moteurs.

Le micro-stepping quand à lui consiste à diviser le pas d'un moteur en pas encore plus petit donc à multiplier le nombre de pas total du moteur pour faire 1 tour.

Par exemple, notre moteur Nema17 – 200 pas par révolution, avec un micro-stepping de 1/2, effectuera $200 \times 2 = 400$ pas pour faire un tour.

Donc en suivant la logique le DRV8825 est capable de multiplier par 32 le nombre de pas de notre moteur, alors que l'A4988 le multiplie au maximum par 16.

Généralement les DRV8825 sont préférés car ils admettent plus de courant et possèdent un microstepping bien plus fin, le seul point "négatif" c'est qu'il est un peu plus cher, ainsi que rare sur notre marché.

Et enfin, pour alimenter tout ça, nous allons avoir besoin d'une alimentation assez puissante principalement pour supporter la charge des moteurs. Chaque moteur consomme environ 5W, ce qui nous donne 10W sur un montage à 2 moteurs.

Dans notre machine on utilise une alimentation 12V 5A figure 6 (environ 60W) qui est largement suffisante pour mon utilisation :



Figure 6 : Alimentation 12V 5A

2- Logiciels

Maintenant que nous avons présenté le matériel, nous allons voir les logiciels nécessaires pour programmer, paramétrer et piloter une CNC.

Le premier logiciel dont on va avoir besoin c'est l'Arduino IDE qui est l'environnement de développement Arduino. Il est totalement gratuit et va nous permettre de programmer et

chargé le programme embarqué dans notre carte Arduino. Ce programme aussi appelé firmware en anglais.

Une fois que nous l'avez installé, on le lance pour vérifier qu'il s'est bien installé, ou cette fenêtre doit être apparaître figure 7.

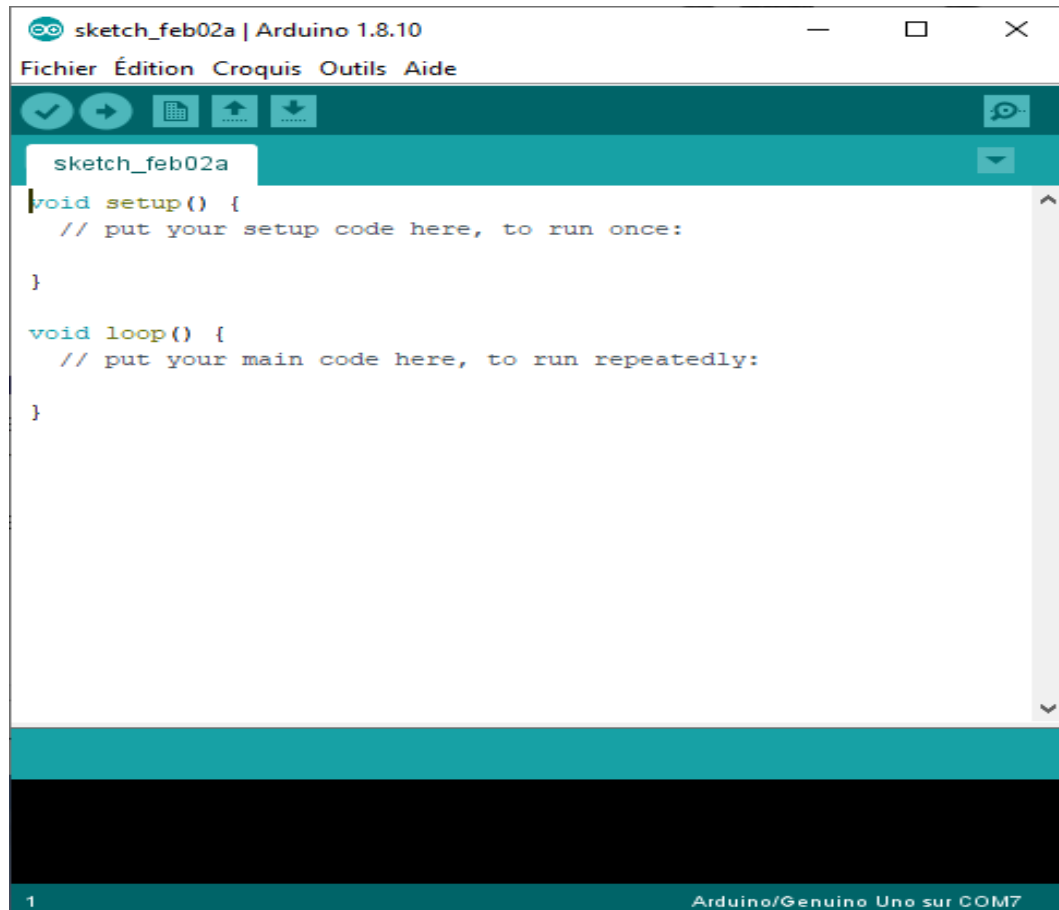


Figure 7 : Interface Arduino IDE

Tous d'abord on doit Branchez notre carte Arduino à l'ordinateur et vérifiez dans l'onglet "Outils" que le logiciel est capable de reconnaître notre carte correctement.

Après on doit sélectionner le port de communication (qui devrait apparaître) figure 8 et le type de carte. Vérifiez que tout est bien détecté, sinon choisissez le bon type de carte.

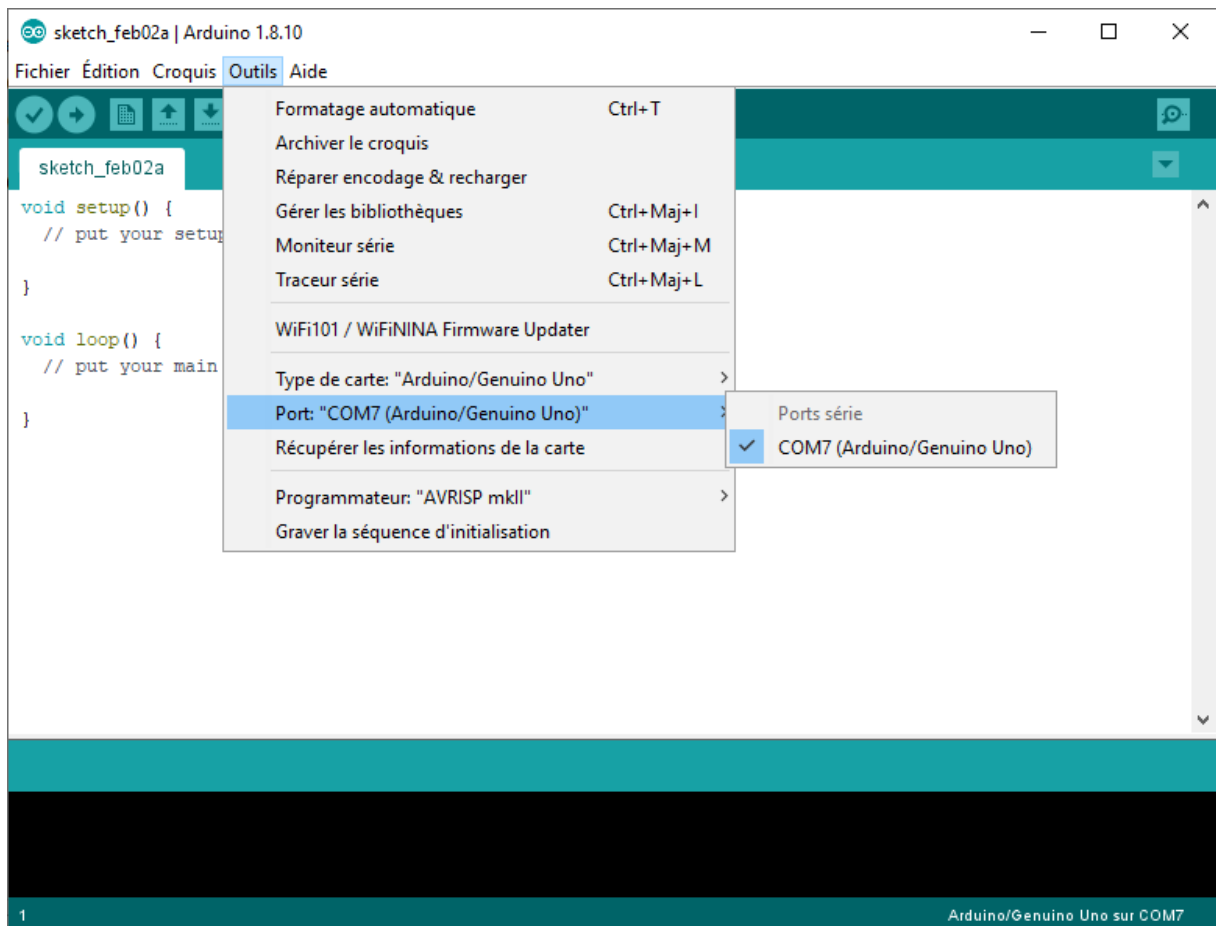


Figure 8 : Le port de communication (qui devrait apparaître).

A ce stade, on est capable de programmer notre Arduino mais la question est : Avec quel Firmware?

Nous allons utiliser ici le firmware GRBL qui est OpenSource (gratuit) et très largement utilisé dans les CNCs grand public.

Il est même assez utilisé pour que des logiciels très connus comme Fusion360 qui intègre l'exportation des chemins d'usinage au format GRBL.

Voici le lien pour le télécharger : **[GRBL - Last Release](#)**

On doit télécharger le zip "Source code (zip)" figure 9.



Figure 9 : "Source code (zip)".

Il suffit ensuite de l'extraire, ou nous allons avoir une arborescence de dossier et de fichier, le seul dossier qui nous intéresse c'est "grbl".

Ce dossier est une bibliothèque pour le logiciel Arduino IDE, on va donc retourner sur le logiciel et inclure ce dossier en cliquant sur "Ajouter la bibliothèque ZIP" dans l'onglet "Croquis > Inclure une bibliothèque".

Là, on peut naviguer jusqu'à l'endroit où nous aurons extrait le dossier et nous sélectionnons le dossier "grbl" figure 10.

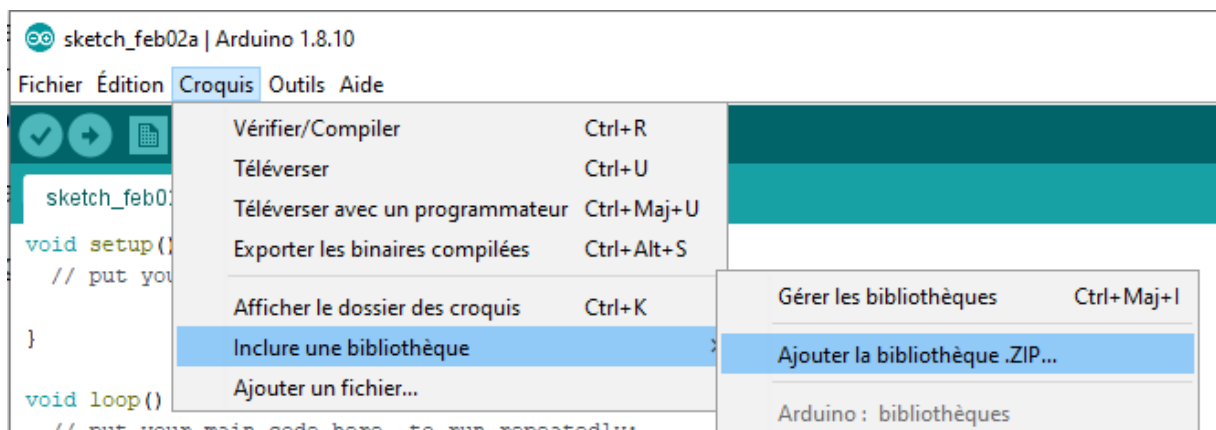


Figure 10 : Procédure à suivre pour inclure le dossier "grbl".

Après ça, en retournant dans l'onglet croquis, nous devrions avoir la bibliothèque grbl qui est apparu dans le même onglet figure 11 :

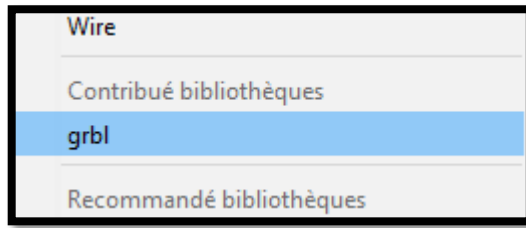


Figure 11: Bibliothèque grbl.

A ce stade nous avons la bibliothèque GRBL prêt ! Il ne nous reste plus qu'à programmer la carte Arduino avec.

Pour cela, nous allons simplement ouvrir le projet ArduinoIDE fournis avec GRBL. Comme on a ajouté GRBL en tant que bibliothèque on a juste à aller sur "Fichier >Exemples>grbl>grblUpload", et le projet va s'ouvrir dans une nouvelle fenêtre !

Il ne reste plus qu'à "téléverser" le programme sur la carte Arduino en cliquant sur ce petit bouton en haut a gauche de la fenêtre :

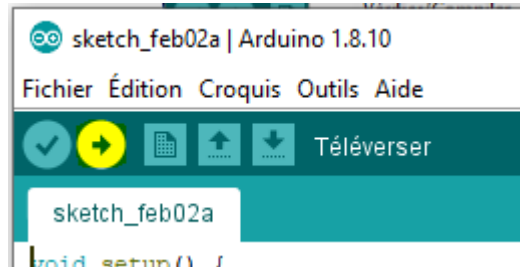
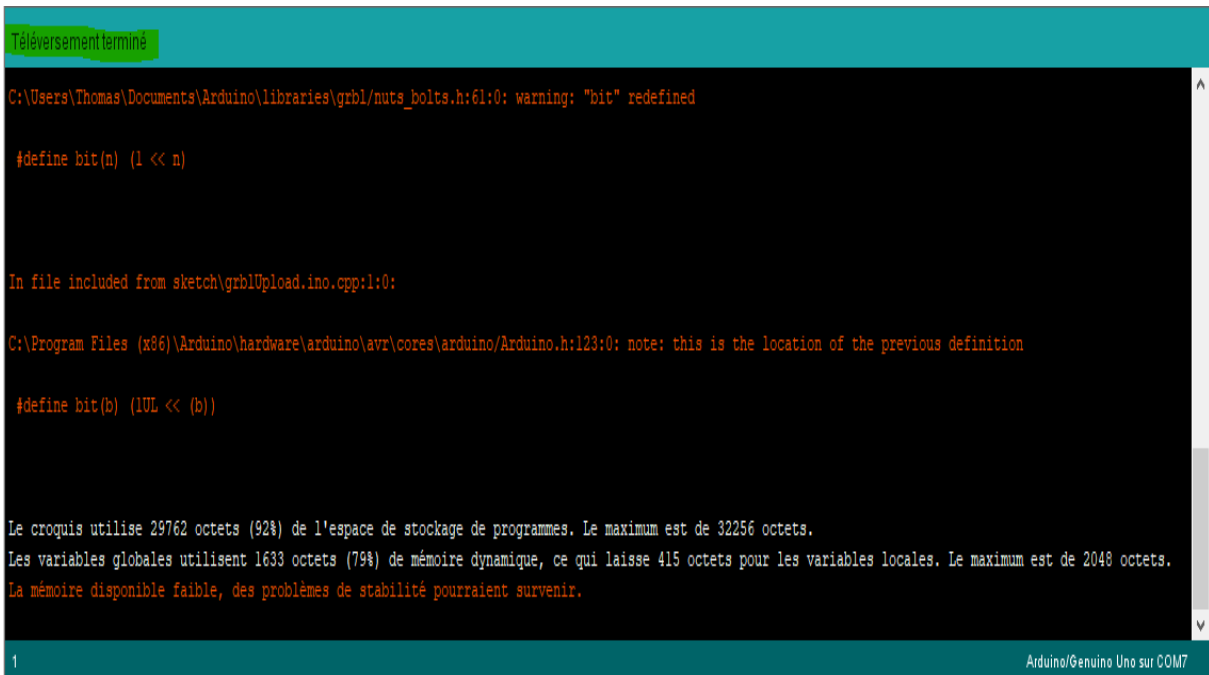


Figure 12 : Programmation GRBL sur Arduino.

Dans la console noire en bas de la fenêtre, on doit voir des choses passer, des informations de compilation et de téléversement, donc la programmation de GRBL dans votre Arduino est en train de faire figure 12.

Attendez un peu jusqu'à que la programmation soit terminée, la console devrait ressembler à cela, affichant "Téléversement terminé" si tout s'est bien passé figure 13:



```
Téléversement terminé
C:\Users\Thomas\Documents\Arduino\libraries\grbl\nuts_bolts.h:61:0: warning: "bit" redefined

#define bit(n) (1 << n)

In file included from sketch\grblUpload.ino.cpp:1:0:
C:\Program Files (x86)\Arduino\hardware\arduino\avr\cores\arduino/Arduino.h:123:0: note: this is the location of the previous definition

#define bit(b) (1UL << (b))

Le croquis utilise 29762 octets (92%) de l'espace de stockage de programmes. Le maximum est de 32256 octets.
Les variables globales utilisent 1633 octets (79%) de mémoire dynamique, ce qui laisse 415 octets pour les variables locales. Le maximum est de 2048 octets.
La mémoire disponible faible, des problèmes de stabilité pourraient survenir.
```

Figure 13: La console de Téléversement.

A cette étape donc on peut dire que nous avons programmé notre carte Arduino, avec GRBL, Arduino IDE nous indique que le téléversement s'est bien passé, mais nous n'avons pas encore le moyen de vérifier que GRBL lui même fonctionne bien...

C'est donc le moment d'introduire le troisième et dernier logiciel dont on va avoir besoin, nommé : CNC JS

CNC JS va nous permettre de communiquer avec GRBL. Nous allons pouvoir le paramétrer mais aussi et surtout, piloter notre CNC.

Pareil que pour les autres logiciels, nous allons télécharger et installer CNCJS, voici le lien : **CNCJS - Last release**

Une fois installé, nous pouvons le lancer, vous devriez avoir cette fenêtre figure 14:

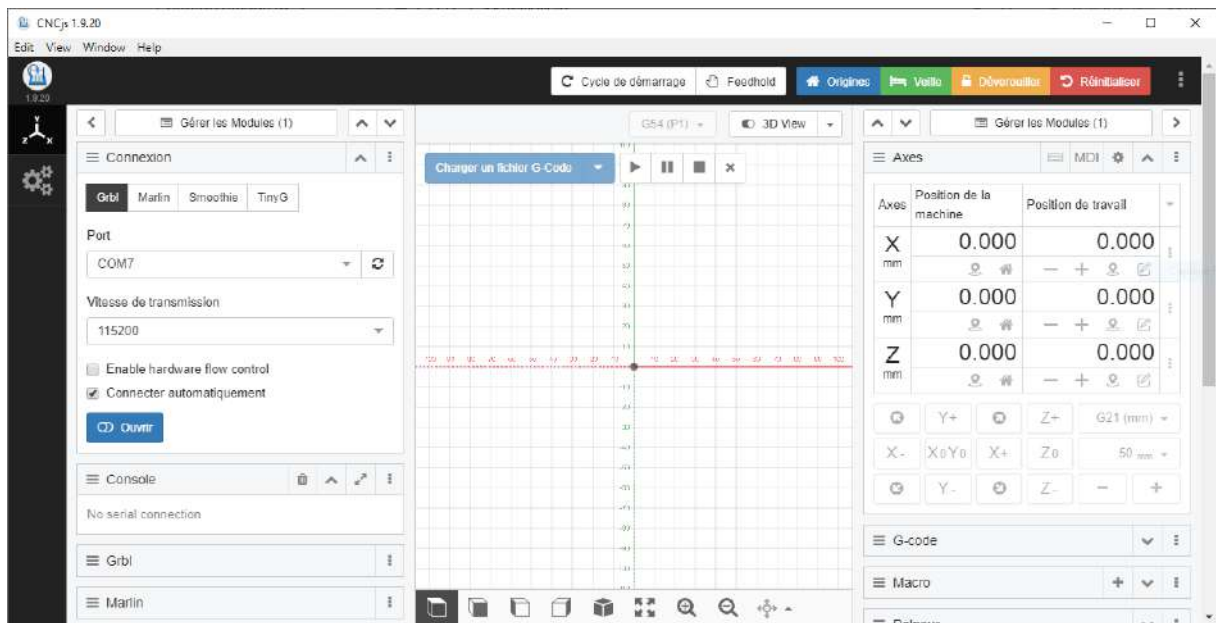


Figure 14: Le programme CNCjs

Nous pouvons voir à gauche une section intitulée "Connexion". Sélectionnez le port de communication de votre Arduino (le meme que dans ArduinoIDE), régler la vitesse de transmission a 115200 puis cliqué sur "Ouvrir".

La section "Console" juste en dessous devrait s'ouvrir et afficher un tas de paramètres en blanc sur un fond noir figure 15:

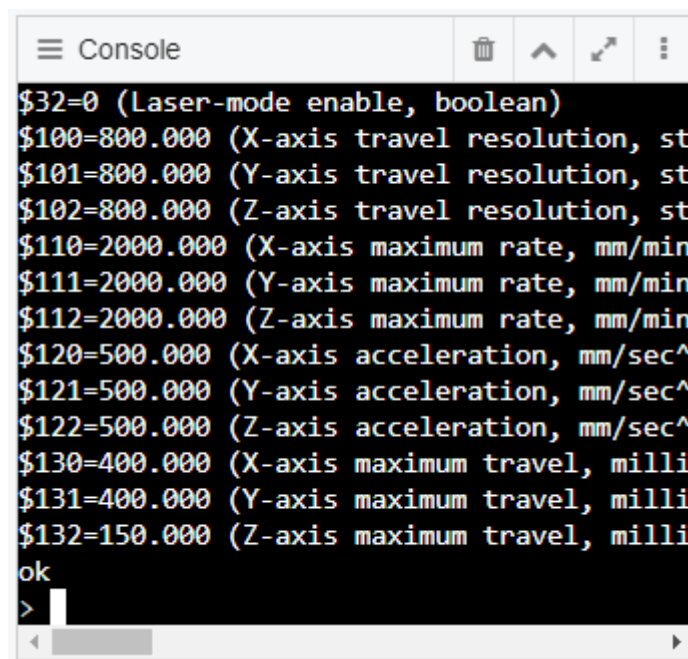


Figure 15: La console du programme CNCjs

Ces paramètres sont envoyés par GRBL, ce qui veut dire que GRBL est correctement exécuté. En remontant un peu dans la console, vous devriez même retrouver le numéro de version de GRBL que vous venez d'installer.

On vient de terminer l'installation des logiciels et la programmation de l'arduino. Nous allons aussi vérifier que GRBL été bien installé et qu'il répondait, nous allons maintenant passer au montage / assemblage des composants entre eux.

3- Montage

Pour commencer, nous allons monter la carte CNC Shield sur l'arduino, à ce niveau nous ne pouvons pas vraiment nous tromper figure 16.

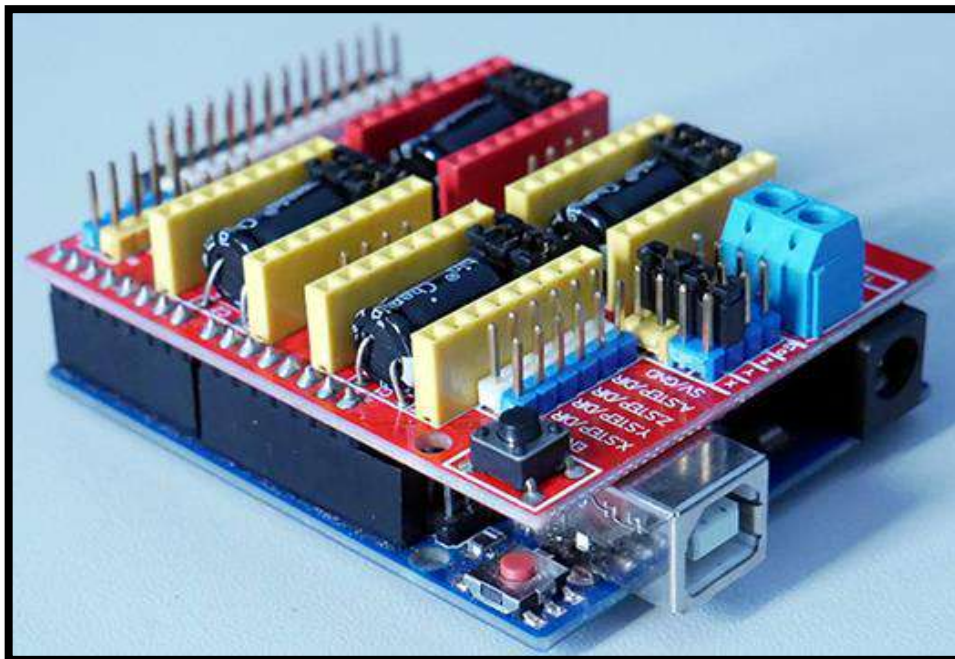


Figure 16: Montage du CNC Shield sur l'Arduino

Ensuite normalement on va placer les cavaliers de micro-stepping, il y a 3 emplacements possible sous chaque emplacement de driver moteur figure 17, mais pour notre cas on pas pu les achetés.

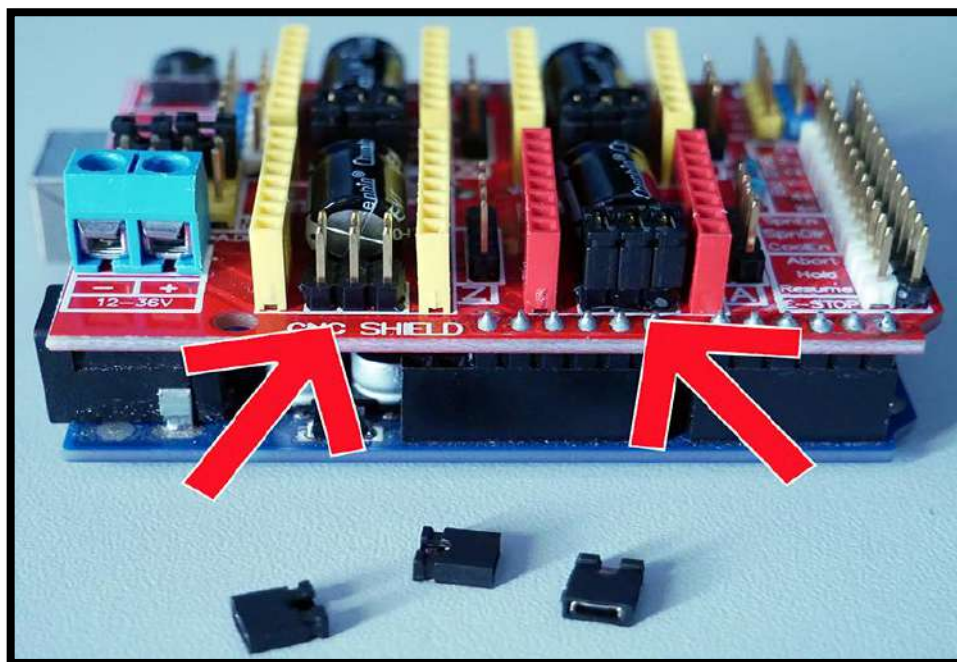


Figure 17 : Endroit du placement des cavaliers de micro-stepping

En fonction de comment on positionne les jumpers (cavaliers), on sélectionne le mode de division de micro-stepping, voici le tableau 1 de correspondance pour chaque référence de driver.

DRV8825			A4988			Mode de micro-stepping
J1	J2	J3	J1	J2	J3	
OFF	OFF	OFF	OFF	OFF	OFF	1 - aucun micro-stepping
ON	OFF	OFF	ON	OFF	OFF	1/2
OFF	ON	OFF	OFF	ON	OFF	1/4
ON	ON	OFF	ON	ON	OFF	1/8
OFF	OFF	ON	ON	ON	ON	1/16
ON	ON	ON				1/32

Tableau 1 : Le tableau de correspondance des cavaliers

Retenez bien quel mode vous avez sélectionné ça va être très important pour la suite.

Ensuite, on colle les radiateurs souvent livrés avec les drivers figure 18. Si, ca concerne une machine assez grosse, il nous faut d'acheter des radiateurs beaucoup plus gros qui permettront de mieux refroidir les drivers.

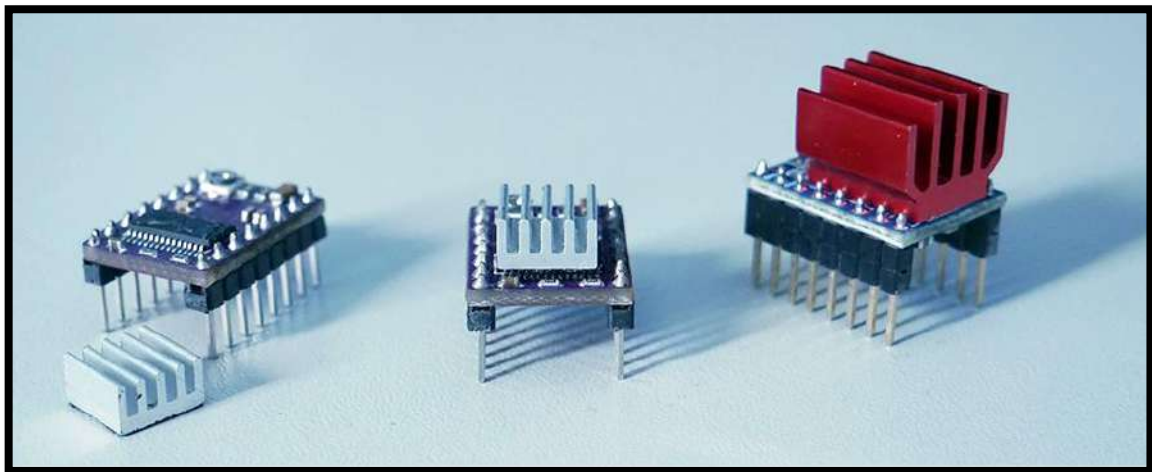


Figure 18 : Les drivers avec les radiateurs.

(À gauche le driver sans radiateur / au milieu le driver avec le radiateur d'origine / à droite le driver avec un radiateur plus performant)

Nous allons pouvoir placer les drivers sur leur emplacement respectif, sur la CNC Shield.

Remarque : Ici, il faut bien regarder et positionner la pin "Enable" (ou "En") du driver sur la pin Enable de la CNC Shield, sinon vous êtes sur de tous griller figure 19.

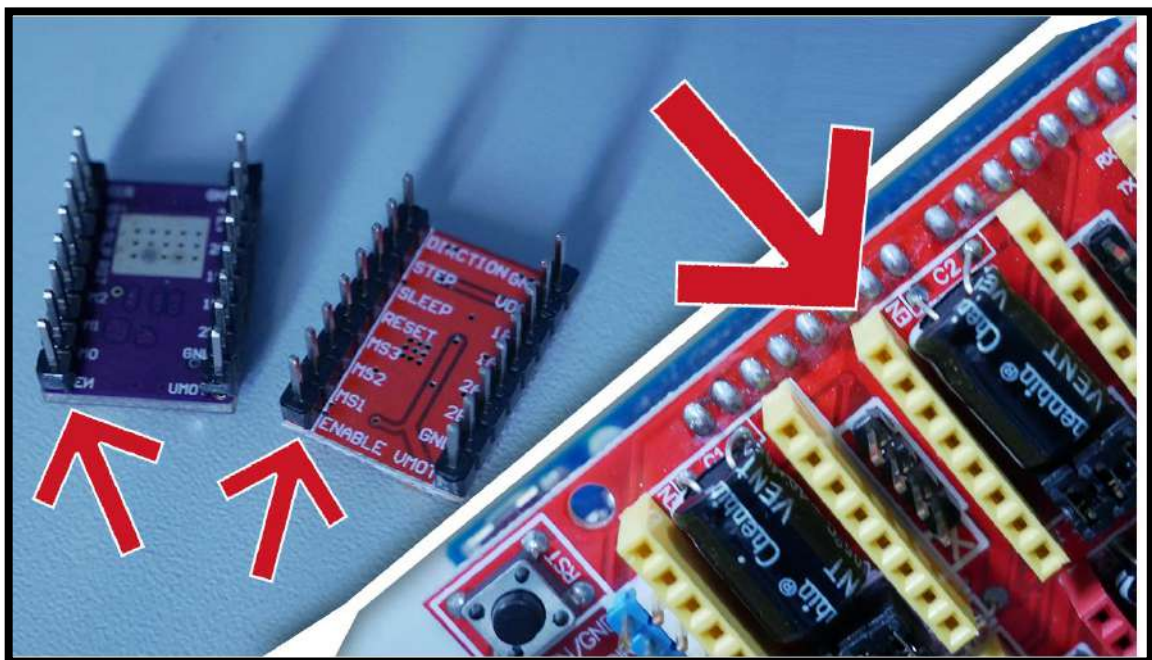


Figure 19 : Mise en place des drivers sur leur emplacement respectif, sur la CNC Shield

Une fois que vous avez mis tous vos drivers en faisant attention au pin Enable, nous allons nous attarder sur le 4ème emplacement de driver moteur.

Sur la CNC Shield, à coté de chaque emplacement de driver moteur, il y a le nom de l'axe que ce driver va piloter (X, Y ou Z), mais il y a aussi un 4ème driver intitulé 'A'.

Ce 4ème driver va pouvoir recopier tous les mouvements d'un des 3 autres axes, d'une CNC avec 2 moteurs pour 1 seul axe, nous allons devoir utiliser cette fonctionnalité de duplication d'un axe.

Sur le coté de la CNC Shield, vous avez des emplacements pour des jumpers portant le nom des axes, vous allez devoir placer 2 jumpers cote à cote pour sélectionner l'axe que vous voulez répliquer figure 20.

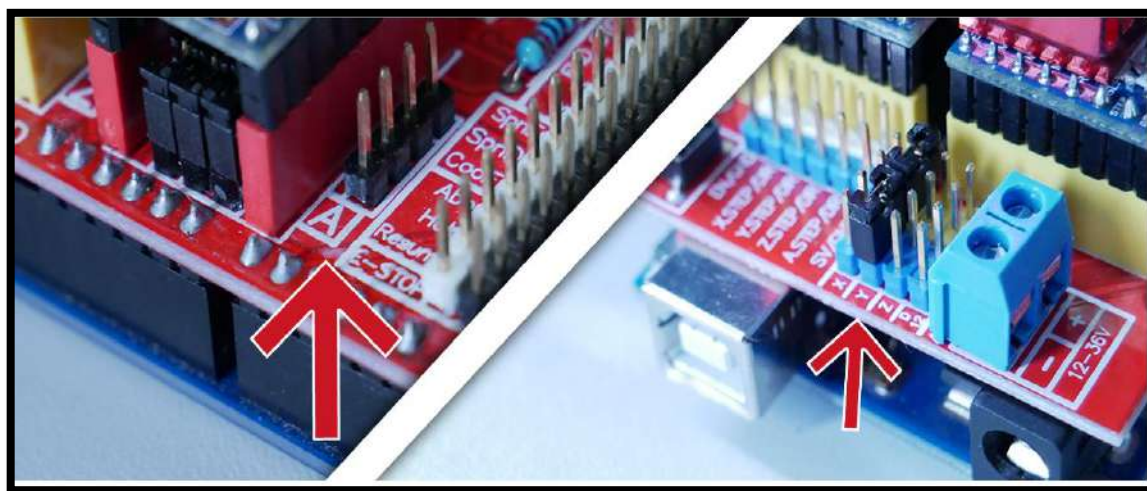


Figure 20 : Endroit d'emplacement des jumpers sur la CNC Shield, pour répliquer un axe.

Il nous reste plus qu'à brancher l'alimentation et les 2 moteurs. Pour l'alimentation c'est simple, il y a un bornier sur le CNC Shield, on met le + sur le fil rouge, le - sur le fil noir...

Remarque : Il ne faut pas brancher l'alimentation sur l'Arduino directement mais bien sur le bornier de la CNC Shield, l'Arduino sera simplement alimenté par le port USB figure 21.

Pour brancher les moteurs se trouve un connecteur à 4 pins acoté de chaque driver. Il n'y a pas de sens particulier car en retournant le connecteur, on inverse simplement le sens de rotation du moteur figure 22.

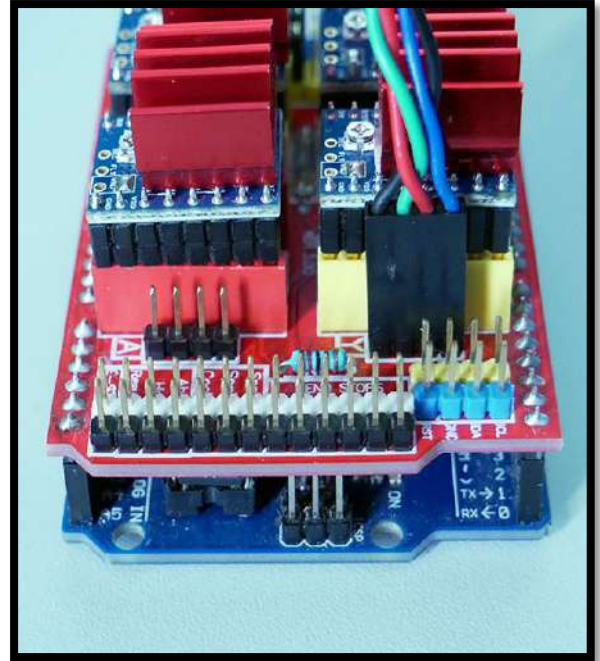
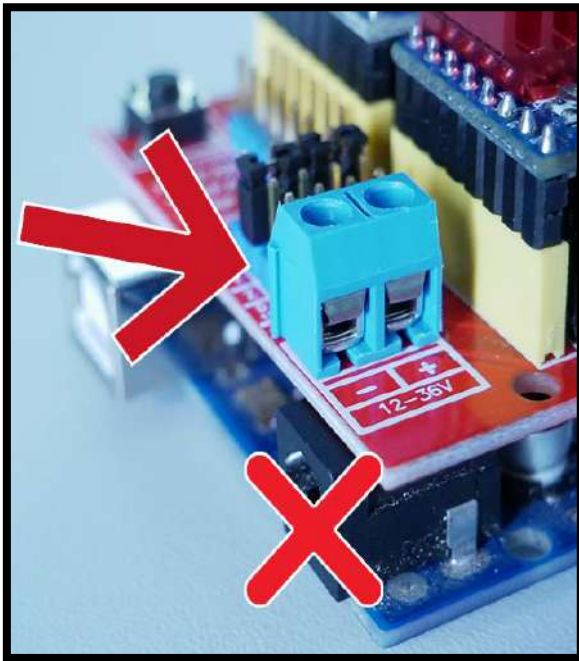


Figure 21 : Raccordement de l'alimentation. Figure 22 : Connexion des moteurs au CNC Shield

A cette étape le montage est terminé ! Nous pouvons passer au paramétrage de GRBL.

4- Paramétrage

Pour le paramétrage on va rebrancher notre Arduino en USB à l'ordinateur et on va alimenter la CNC Shield avec l'alimentation branchée juste avant.

Une fois que tout est branché nous pouvons retourner sur CNCJS, ouvrir la communication et ouvrir la section console. On doit avoir comme plus haut tout cela qui s'affiche :

```
CNCjs 1.9.20 [Grbl]
Connected to COM5 with a baud rate of 115200
Grbl 1.1h ['$' for help]
client> $$
$0=10 (Step pulse time, microseconds)
$1=25 (Stepidle delay, milliseconds)
$2=0 (Step pulse invert, mask)
$3=0 (Step direction invert, mask)
$4=0 (Invertstepenable pin, boolean)
$5=0 (Invertlimit pins, boolean)
```

\$6=0 (Invert probe pin, boolean)
\$10=3 (Status report options, mask)
\$11=0.010 (Junction deviation, millimeters)
\$12=0.002 (Arc tolerance, millimeters)
\$13=0 (Report in inches, boolean)
\$20=0 (Soft limitsenable, boolean)
\$21=1 (Hard limitsenable, boolean)
\$22=0 (Homing cycle enable, boolean)
\$23=0 (Homing direction invert, mask)
\$24=25.000 (Homing locatefeed rate, mm/min)
\$25=500.000 (Homing searchseek rate, mm/min)
\$26=250 (Homing switchdebouncedelay, milliseconds)
\$27=1.000 (Homing switch pull-off distance, millimeters)
\$30=17000 (Maximum spindle speed, RPM)
\$31=3000 (Minimum spindle speed, RPM)
\$32=0 (Laser-mode enable, boolean)
\$100=50.000 (X-axis travelresolution, step/mm)
\$101=50.000 (Y-axis travelresolution, step/mm)
\$102=50.000 (Z-axis travelresolution, step/mm)
\$110=500.000 (X-axis maximum rate, mm/min)
\$111=500.000 (Y-axis maximum rate, mm/min)
\$112=500.000 (Z-axis maximum rate, mm/min)
\$120=50.000 (X-axis acceleration, mm/sec²)
\$121=50.000 (Y-axis acceleration, mm/sec²)
\$122=50.000 (Z-axis acceleration, mm/sec²)
\$130=400.000 (X-axis maximum travel, millimeters)
\$131=400.000 (Y-axis maximum travel, millimeters)
\$132=150.000 (Z-axis maximum travel, millimeters)
ok
>

On va seulement s'attarder sur les 4 paramètres les plus importants.

En premier le paramètre "\$1=25", c'est le paramètre qui va indiquer à GRBL au bout de combien de temps il doit éteindre (dés-alimenté) le moteur après un mouvement. L'unité de ce paramètre est la milliseconde.

Par défaut, après un mouvement, le moteur est éteint au bout de 25ms.

Un moteur alimenté va forcer pour garder sa position actuelle, s'il est éteint, il n'exercera aucune force pour garder sa position et pourra tourner très facilement.

Prenons l'exemple de l'axe Z qui fait monter / descendre la fraise par exemple. Lorsque l'ont découpe une pièce, le moteur de l'axe Z va faire descendre la fraise dans la pièce, puis va rester à sa position durant toute la découpe. C'est là que l'on commence à comprendre le problème.

Si le moteur descend à la bonne position puis s'éteint vu qu'il n'applique aucune force pour garder sa position, il va très certainement "glisser" plus bas à cause des vibrations pendant le fraisage.

Pour corriger ce problème, il suffit de mettre 255 au paramètre \$1. GRBL considère que 255 est un temps infini donc il n'éteindra jamais les moteurs même après un mouvement.

Remarque : de 1 à 254, GRBL prendra le paramètre comme des millisecondes, seule la valeur 255 est considérée comme l'infini.

Nous allons donc tout simplement taper cette ligne dans la console :

```
$1=255
```

Le second paramètre c'est le groupe "\$100" "\$101" et "\$102". Il s'agit de la résolution de déplacement pour chaque axe (X Y et Z dans l'ordre).

La résolution de déplacement s'exprime en pas par millimètre, c'est le nombre de pas que doit effectuer le moteur pour faire déplacer notre machine de 1mm.

- Pour calculer la résolution, il nous faut :
- Le nombre de pas de base de notre moteur (dans notre cas c'est 200)
- Le mode de micro-stepping choisi, c'est ici qu'il est très important (dans mon cas 1)

- Le système de guidage de la machine (dans mon cas des vis trapézoïdales 2mm par tour)

Sur notre machine on utilise des DRV A4988 , sur l'axe Y la transmission du mouvement est assuré par une vis trapézoïdale don l'écrou avance de 2 mm à chaque fois que la vis fait 1 tour.

Voici donc le calcul de la résolution basé sur les valeurs de notre machine :

$$(200\text{pas (Nema17)} \times 1 \text{ (micro-stepping)}) / 2 \text{ (vis mère (trapézoïdale))} = (200 \times 1) / 2 = \underline{100 \text{ pas / millimètre}}$$

Nous allons faire de la même manière pour l'axe X de notre machine ou cette fois la transmission du mouvement est assuré par l'intermédiaire d'une courroie crantée GT2. Ce soit les courroies classiques utilisées en impression 3D, elles ont un pas de 2mm par dent. Nous allons aussi dire que le moteur est équipé d'une poulie de 20 dents et qu'on utilise un A4988 en 1 micro-stepping.

Donc voici le calcul :

$$(200\text{pas (Nema17)} \times 1 \text{ (micro-stepping)}) / (2 \text{ (pas de la courroie)} \times 20 \text{ (nombre de dent de la poulie)}) = (200 \times 1) / (2 \times 20) = \underline{5 \text{ pas / millimètre}}$$

Nous allons donc renseigner ces valeurs a GRBL pour notre machine :

$$\text{\$100=5}$$

$$\text{\$101=100}$$

$$\text{\$102=100}$$

C'est avec cette valeur que l'on peut calculer la précision atteignable (théorique) par notre machine en divisant 1 millimètre par le nombre de pas :

$$1\text{mm} / 100\text{pas} = 0,01\text{mm}$$

$$1\text{mm} / 5 \text{ pas} = 0.2\text{mm}$$

Le troisième paramètre, c'est le groupe "\$110" "\$111" et "\$112". C'est la vitesse de déplacement maximale pour chaque axe (X Y et Z dans l'ordre).

Le quatrième paramètre, c'est le groupe "\$120" "\$121" et "\$122". C'est l'accélération pour chaque axe (X Y et Z dans l'ordre).

La vitesse s'exprime en mm/min, littéralement "combien de millimètre va parcourir la machine en 1 minute"

L'accélération s'exprime en mm/sec², en gros plus la valeur rentrée ici sera grande, plus l'accélération sera forte.

Ces deux valeurs vont beaucoup dépendre de notre machine, son poids, sa taille et le type d'usinage que nous allons effectuer (une gravure laser pour accepter des accélérations et une vitesse beaucoup plus grand qu'une découpe de bois par exemple).

Dans notre cas la broche n'a pas été montée pour des raisons du temps ont pas pu décider sur un type, ou ces valeurs seront fixés après avoir réalisé des essais sur la machine.

Malgré ça nous avons pu prendre des paramètres de départ pour nos essais :

Sur notre machine qui est petit, sans broche, on a 1000mm/min en vitesse, et 100mm/sec² en accélération :

\$110=1000

\$111=1000

\$112=1000

\$120=100

\$121=100

\$122=100

Partie 2 : Conception mécanique de la machine

1- Introduction :

Pour réaliser un système mécanique permettant de se déplacer selon deux directions, nous avons opté pour l'aluminium concernant l'ossature de la machine, les paliers de guidage, des arbres de transmission selon l'axe X, ainsi que l'accouplement les poulies, pour le reste de la machine les pièces ont été fabriquées en acier par oxycoupage.

La conception mécanique de la machine a été réalisée sur Solidworks, dans cette partie nous exposerons tous les composants de la machine avec le plan d'ensemble de cette dernière.

2- Les différentes parties de la machine

La figure 1 montre les trois projections de la vue d'ensemble de la machine CNC assemblée. On peut y distinguer les différents composants de la machine. Dans ce qui suit nous allons présenter ses éléments.

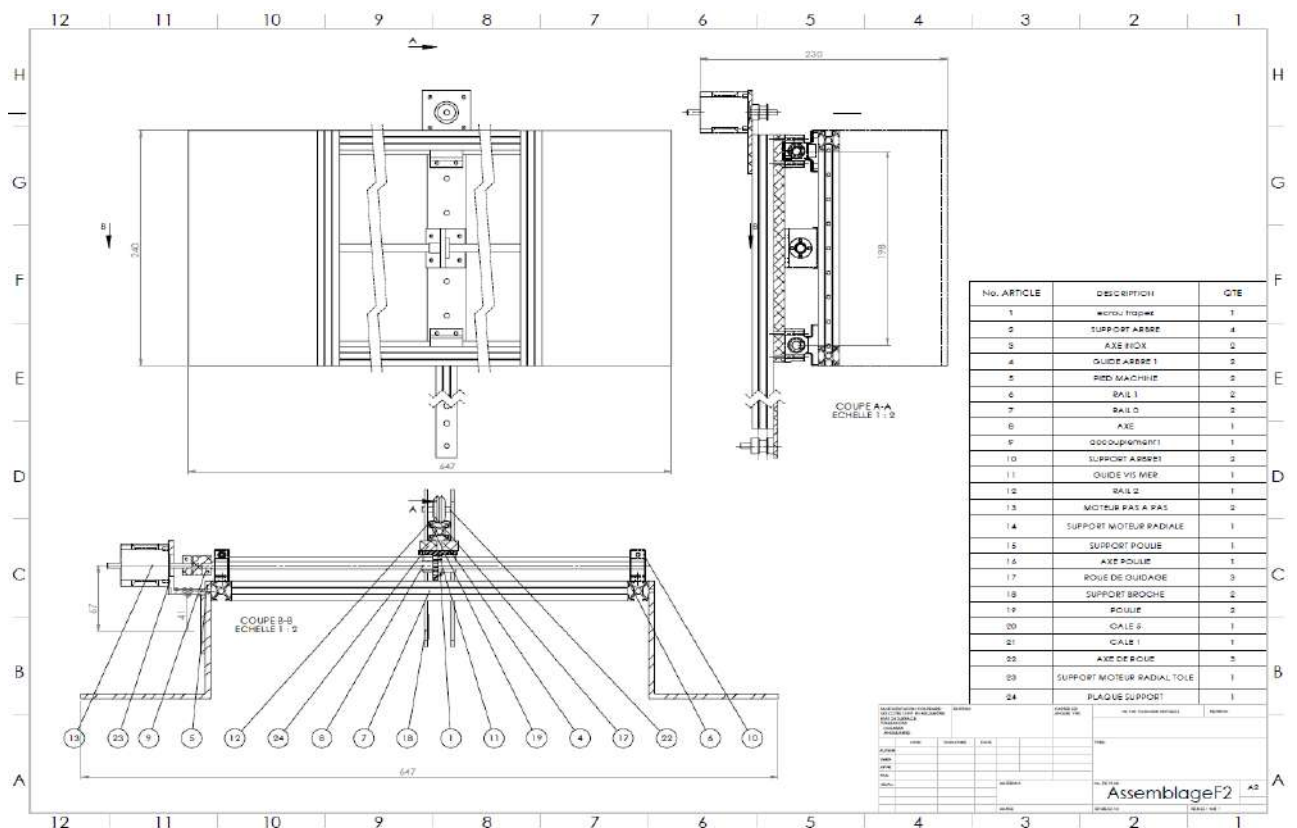


Figure 01 : assemblage

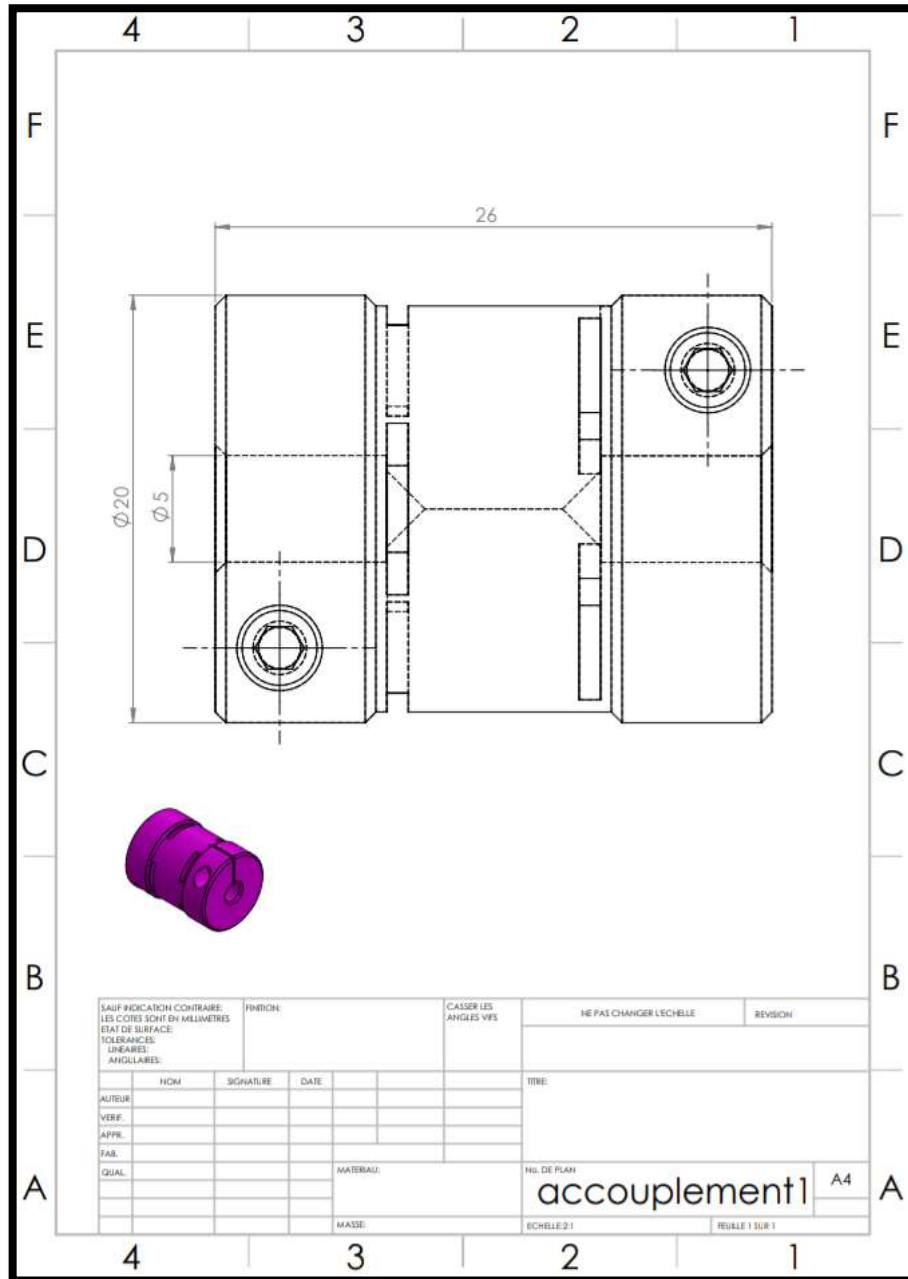


Figure 02 : accouplement1

L'accouplement 1 sur la figure 2 est une pièce, comme son nom l'indique, qui permet l'accouplement entre l'axe du moteur et celui de la vis sans fin. On peut distinguer sur la figure 2 les vise pour le serrage des axe.

No. ARTICLE	DESCRIPTION	QTE
1	ECROU TRAPEZ	1
2	SUPPORT ARBRE	4
3	AXE INOX	2
4	GUIDE ARBRE 1	2
5	PIED MACHINE	2
6	RAIL 1	2
7	RAIL 0	2
8	AXE	1
9	ACCOUPLLEMENT1	1
10	SUPPORT ARBRE1	2
11	GUIDE VIS MER	1
12	RAIL 2	1
13	MOTEUR PAS A PAS	2
14	SUPPORT MOTEUR RADIALE	1
15	SUPPORT POULIE	1
16	AXE POULIE	1
17	ROUE DE GUIDAGE	3
18	SUPPORT BROCHE	2
19	POULIE	2
20	CALE 5	1
21	CALE 1	1
22	AXE DE ROUE	3
23	SUPPORT MOTEUR RADIAL TOLE	1
24	PLAQUE SUPPORT	1

Tableau 1 : les éléments constitutifs de la machine CNC

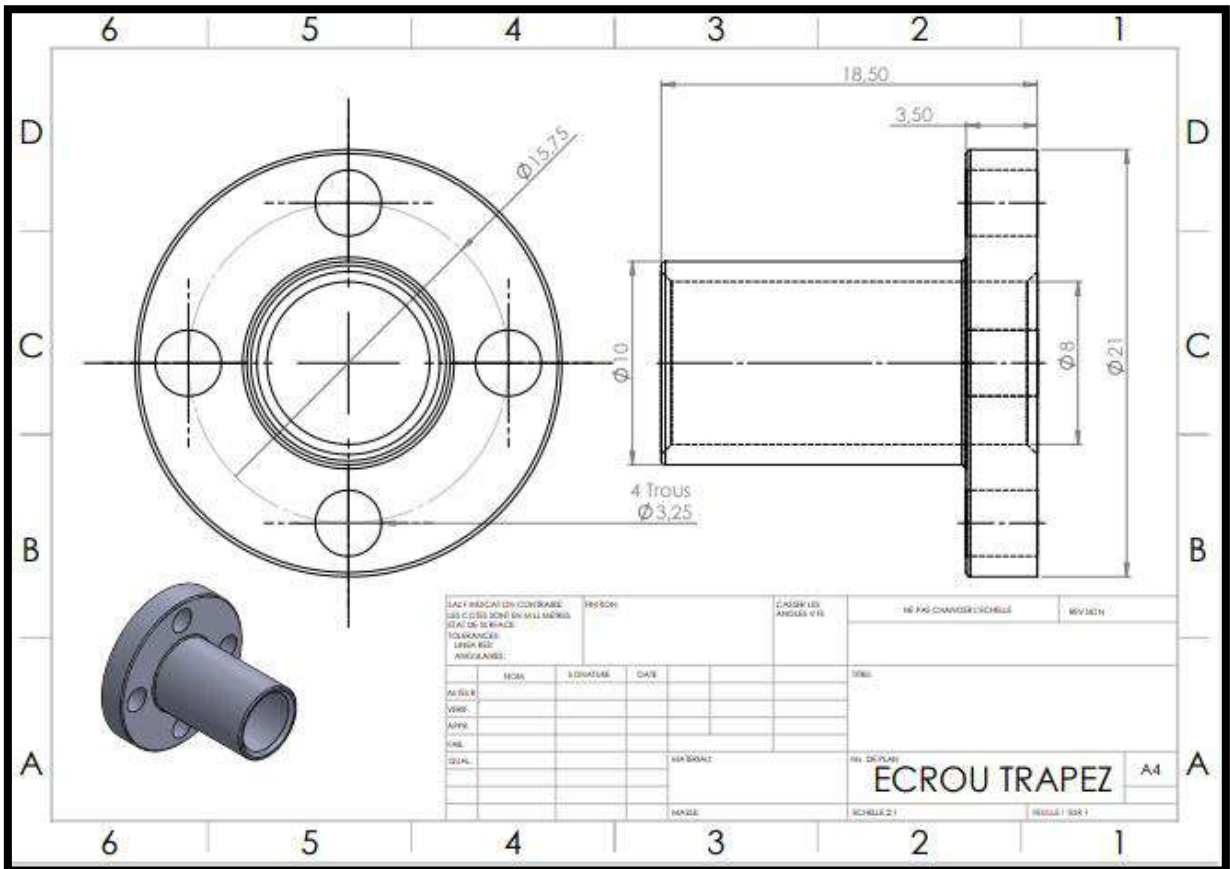


Figure 04 : ECROU TRAPEZ

La figure 04 montres l'écrou trapèze. C'est cet écrou qui permet de transformé le mouvement de rotation de la tige filetée (axe) en mouvement de translation. Cet écrou est solidaire du rail 2.

Dans la figure 05 on peut voir le guide arbre. Cet élément s'emboite dans la tige de guidage. Son rôle est de facilité le mouvement rectiligne grâce aux billes qu'il contient ce qui permet d'éliminer aussi les vibrations.

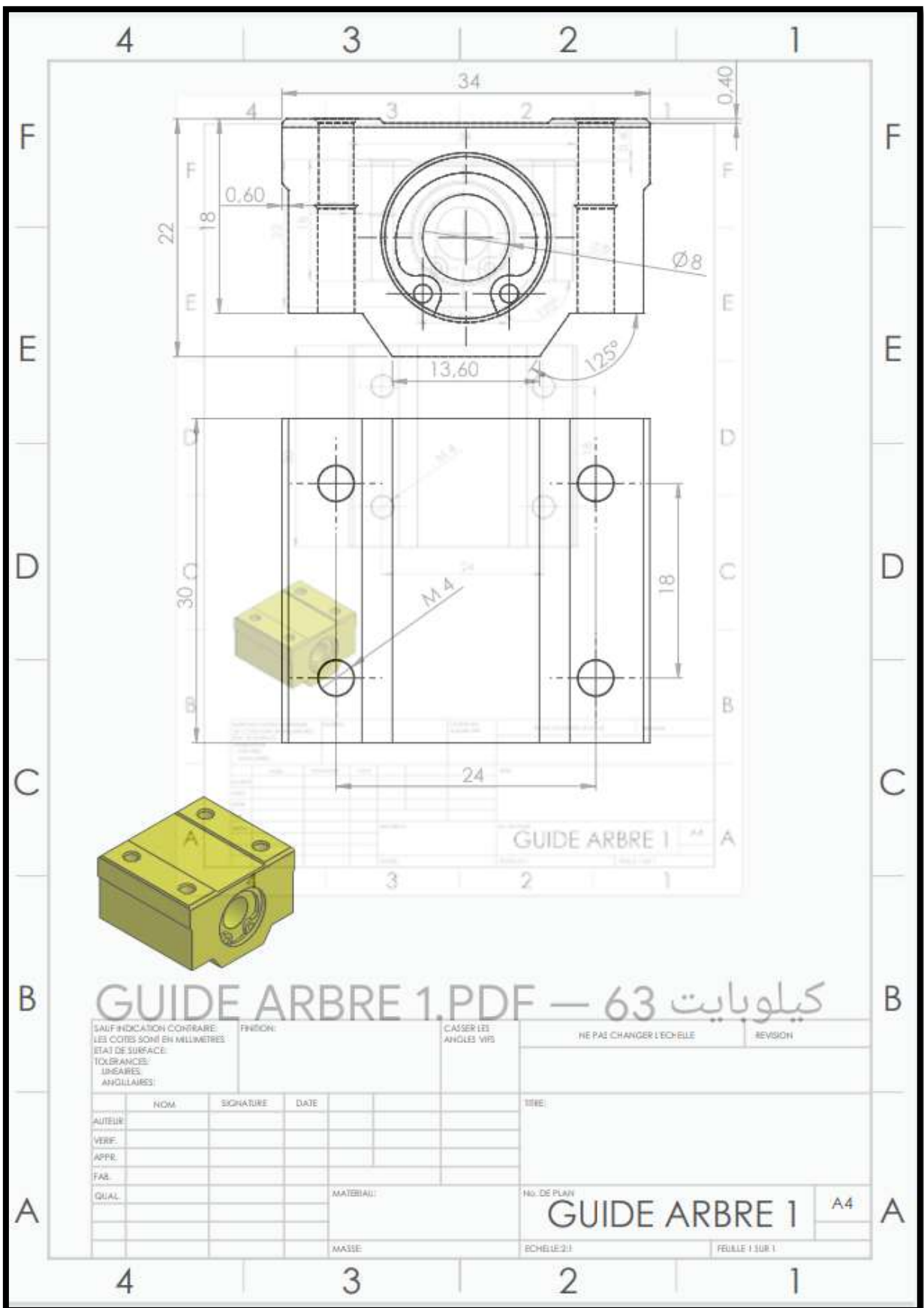


Figure 05: GUIDE ARBRE 1

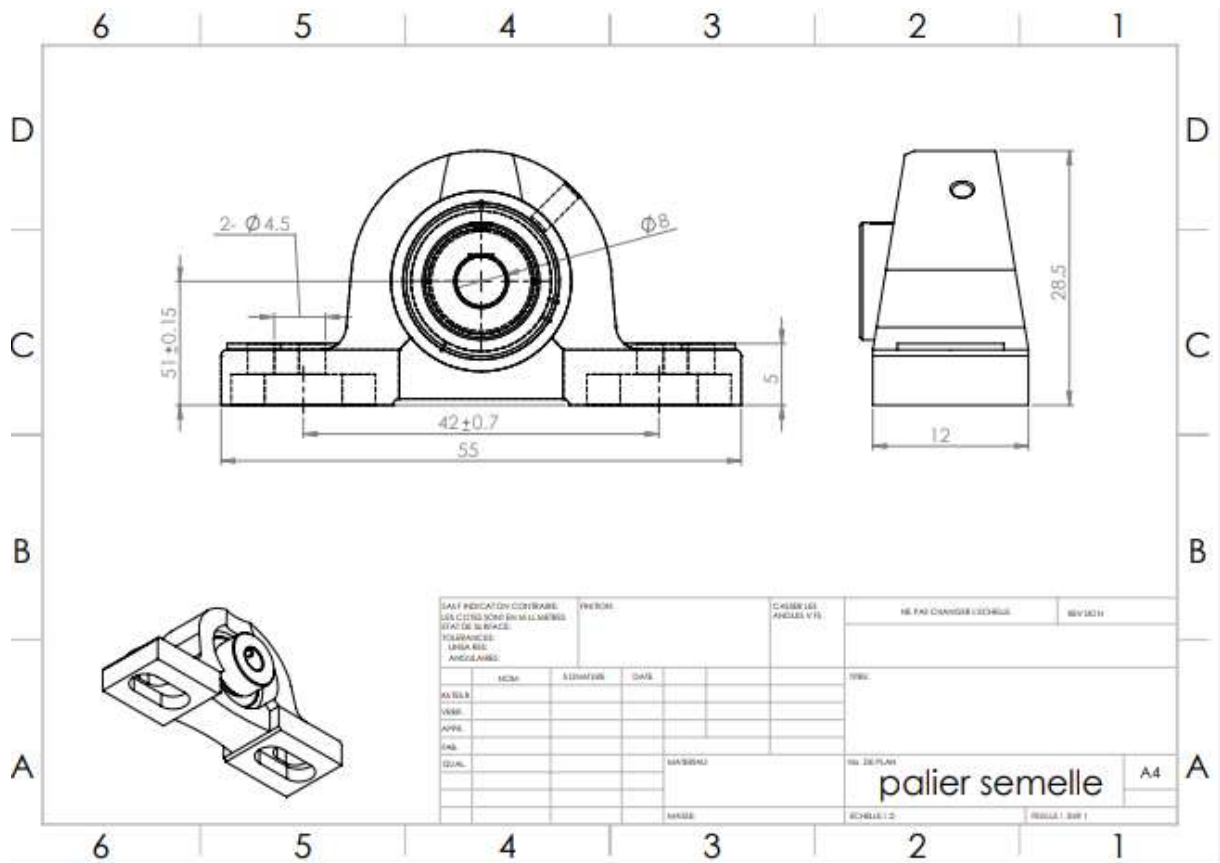


Figure 06 : palier semelle

On peut voir dans la figure 6 le palier semelle. Ce dernier a pour rôle de maintenir les axes (vis mètre) pour empêcher les mouvements de translation tout en facilitant le mouvement de rotation.

La figure 07 donne les dimensions de la plaque support. Cette plaque a pour but de supporter le rail 2. Et de stabilisé l'ensemble.

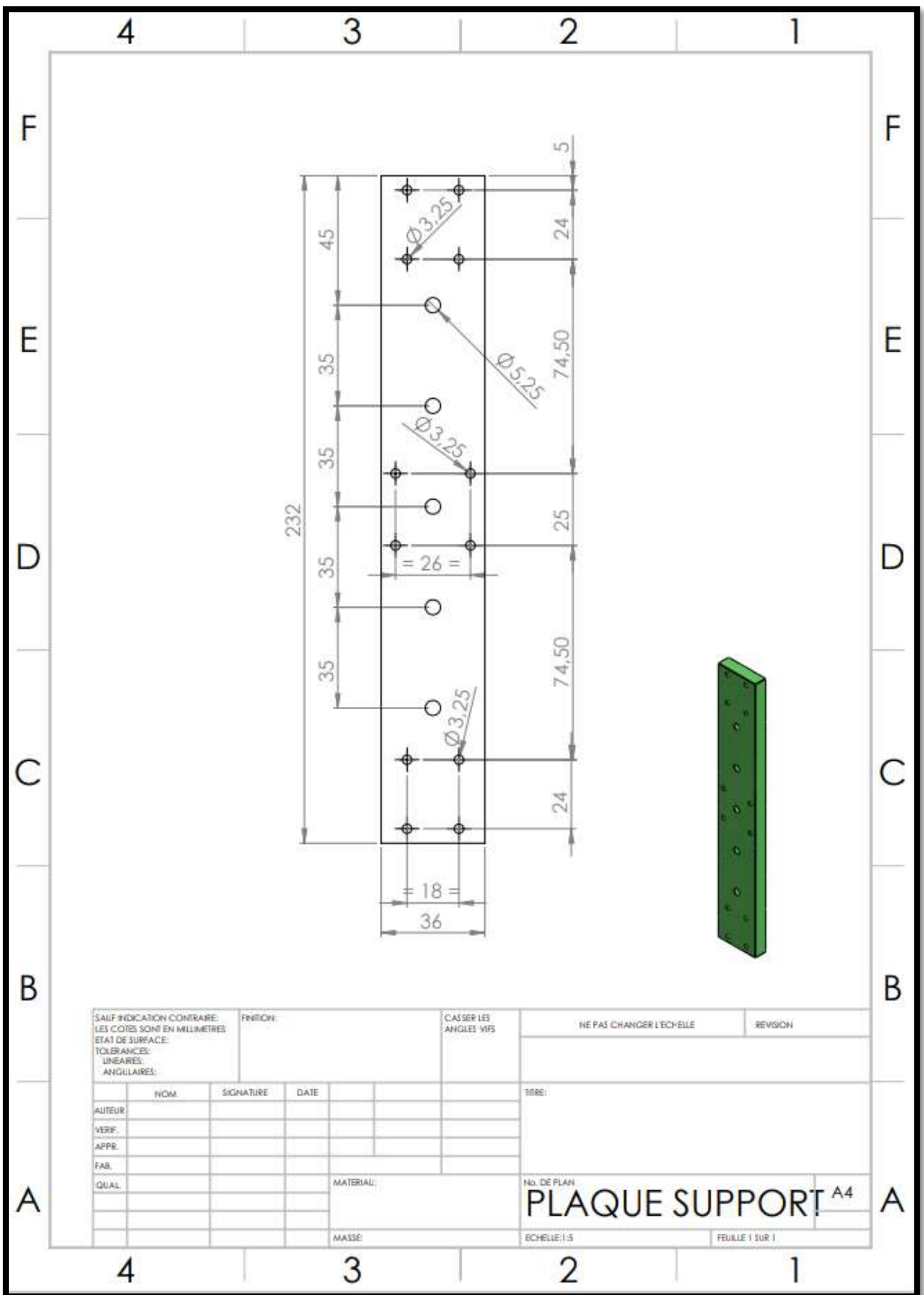


Figure 07 : PLAQUE SUPPORT

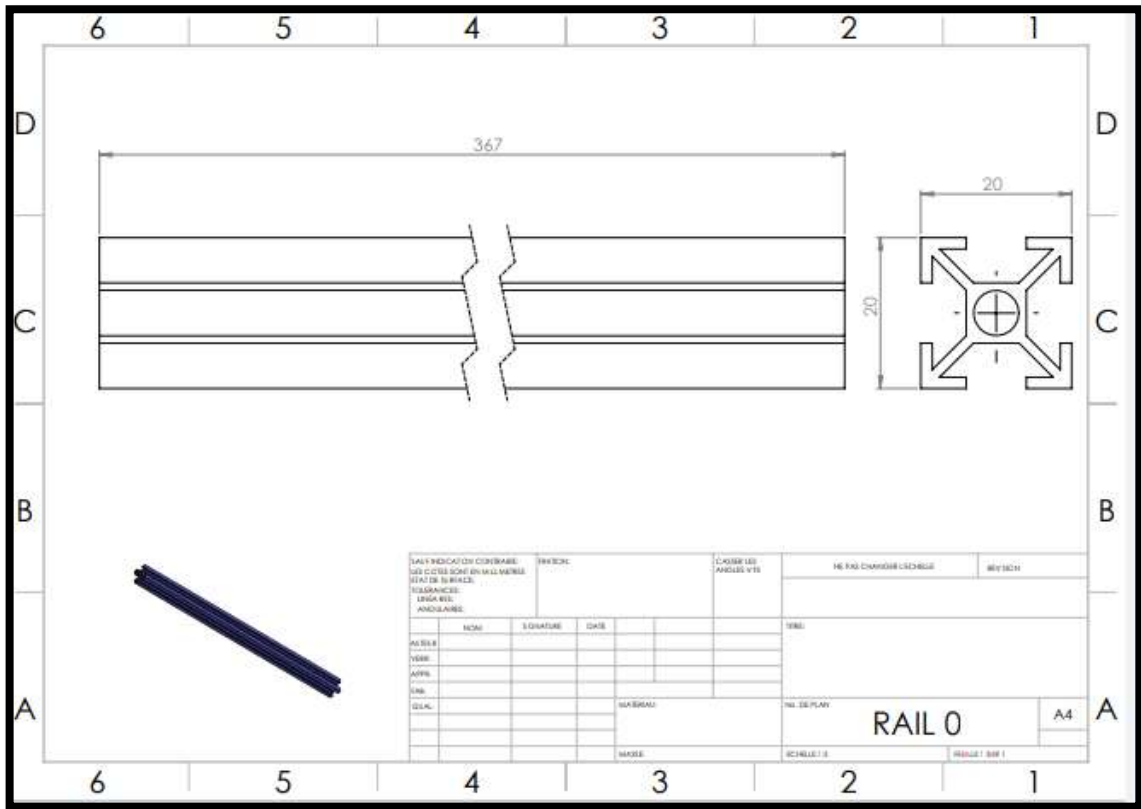


Figure 08 : RAIL 0

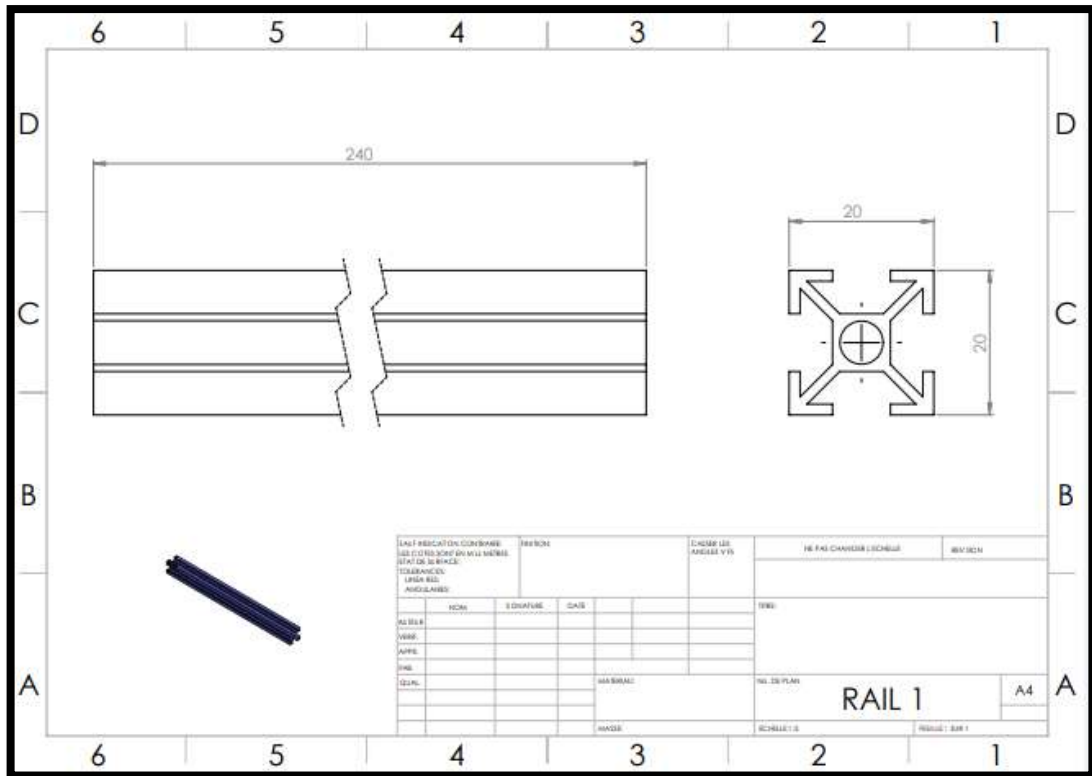


Figure 09 : RAIL 1

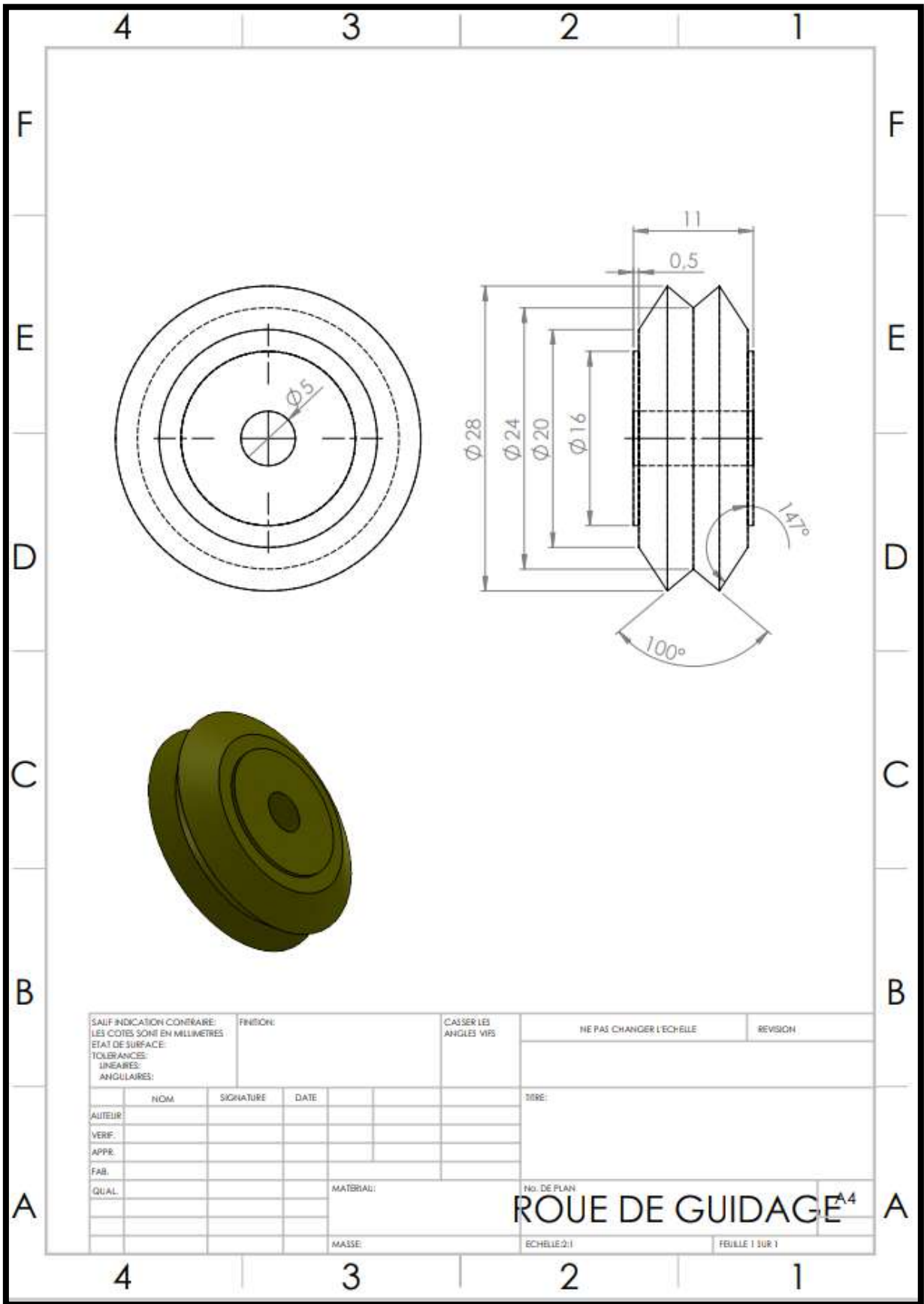


Figure 11 : ROUE DE GUIDAGE

La figure 11 représente une roue de guidage. Ces roue serve à maintenir le chariot sur les rails tout en permettant et facilitant la translation. (Mouvement rectiligne).

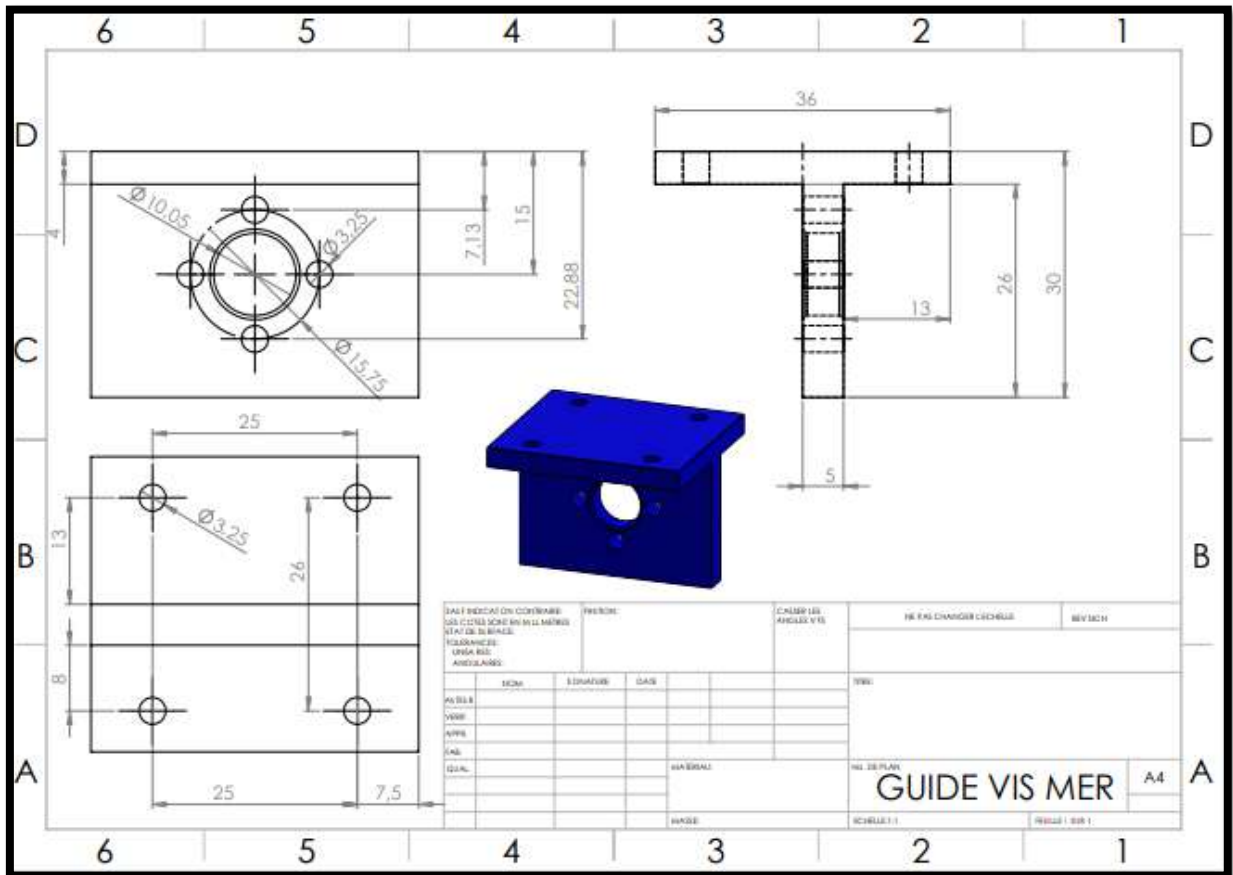


Figure 12 : GUIDE VIS MERE

Sur la figure 12 on peut voir une pièce appelé guide vis mer. Sur cette élément vient se fixé l'écrou trapèze d'un côté et la plaque support d'un autre côté.

Sur la figure 13 est représentée une poulie. Cette dernière sert à guidé le mouvement de la courroie qui commande le déplacement du chariot.

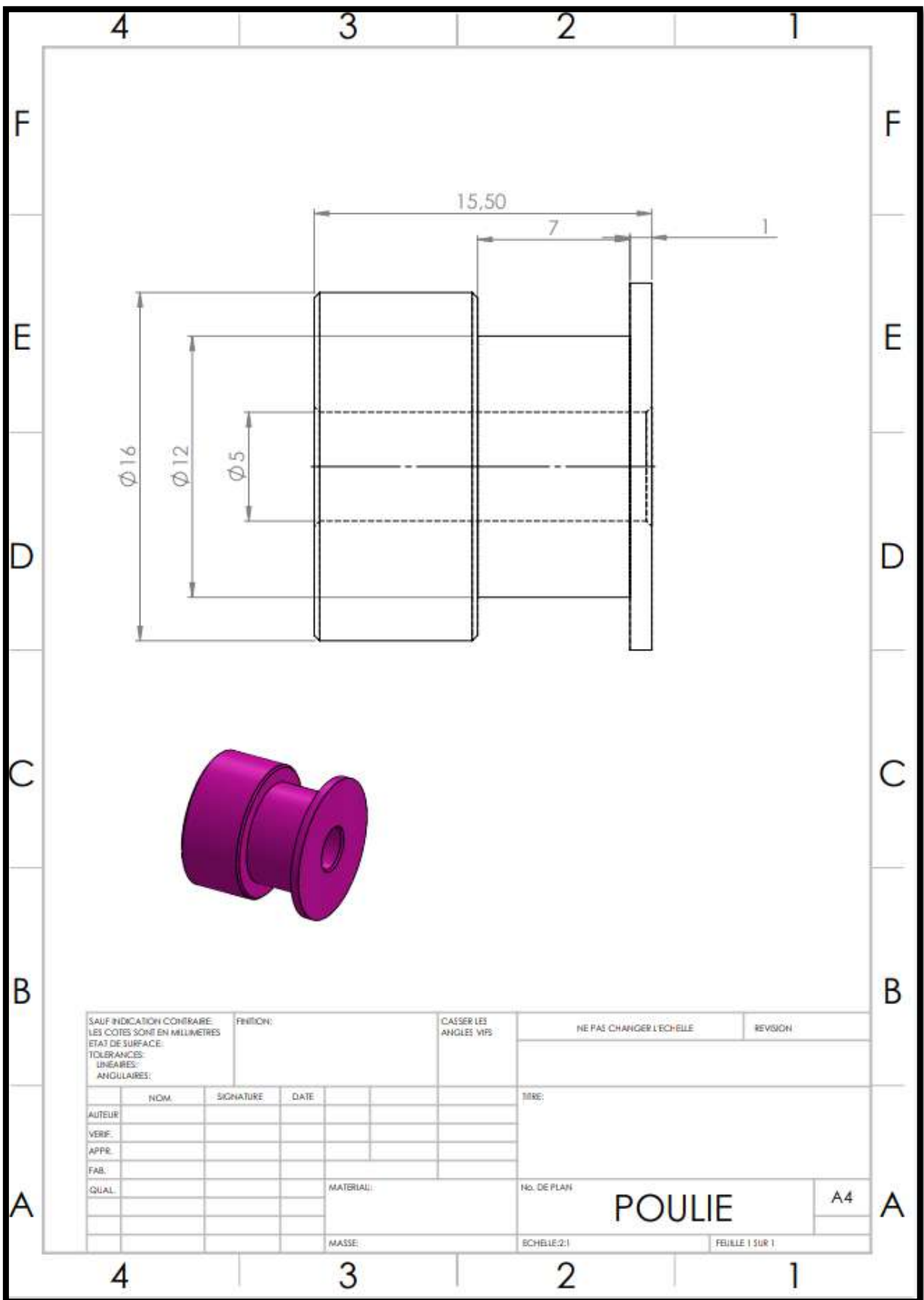


Figure 13 : POULIE

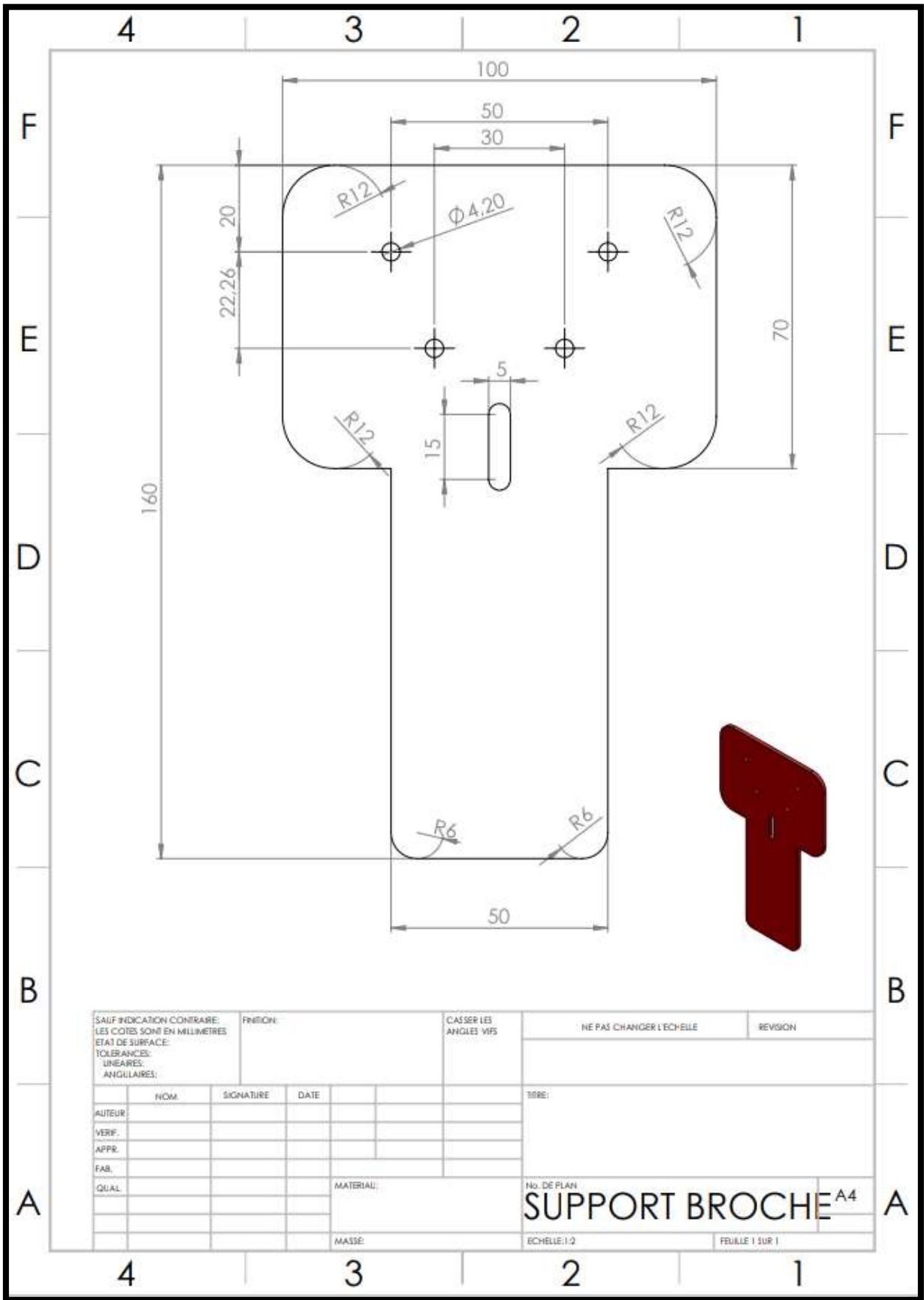


Figure 15 : SUPPORT BROCHE

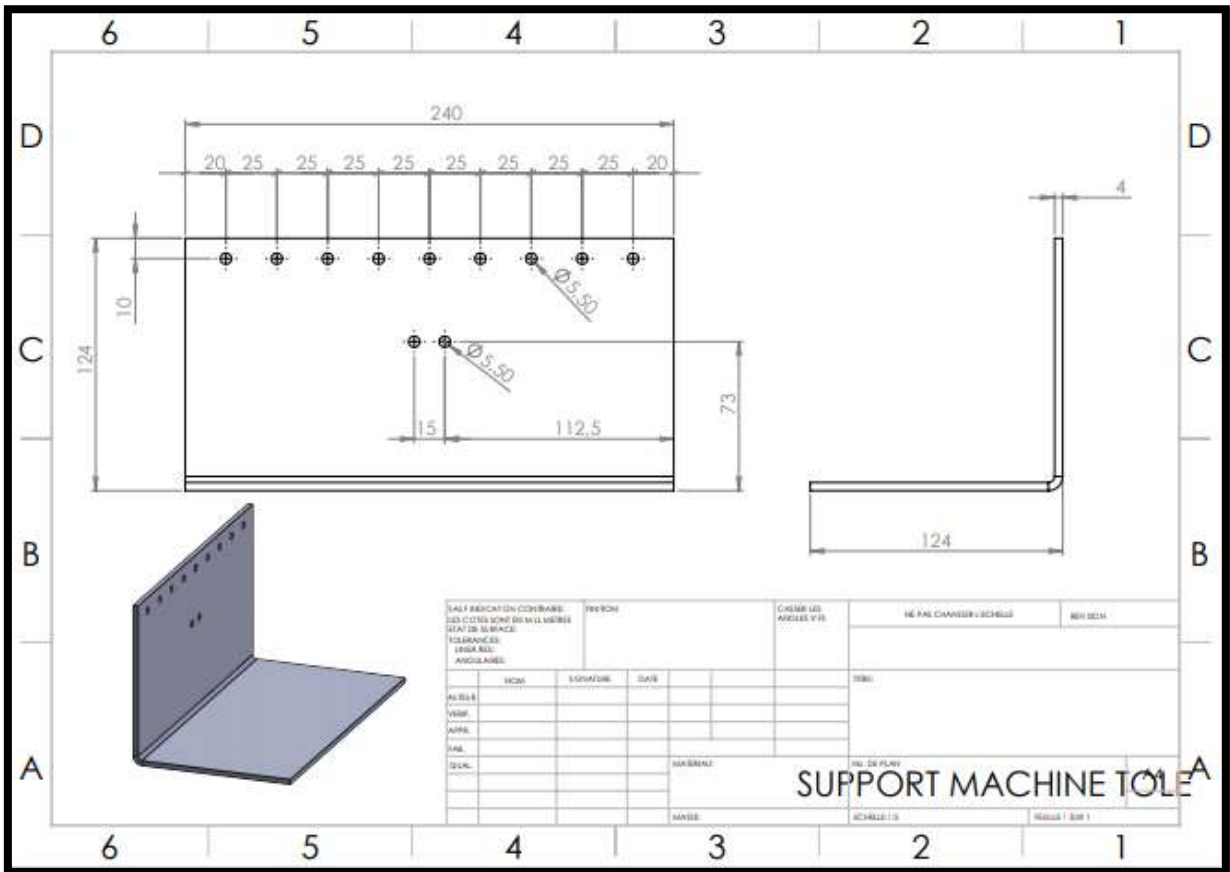


Figure 16 : SUPPORT MACHINE

Afin de stabiliser la machine, notamment lors de son fonctionnement, il est impératif d'installer des supports sur la machine. La figure 16 donne les dimensions des supports choisies pour cette machine.

Sur la figure 17 est illustré le support moteur de l'axe x. comme son nom l'indique c'est sur cette élément que se fixe le moteur pour être fixé au châssis de la machine CNC à son tour. De même sur la figure 18 on peut voir le support moteur de l'axe y.

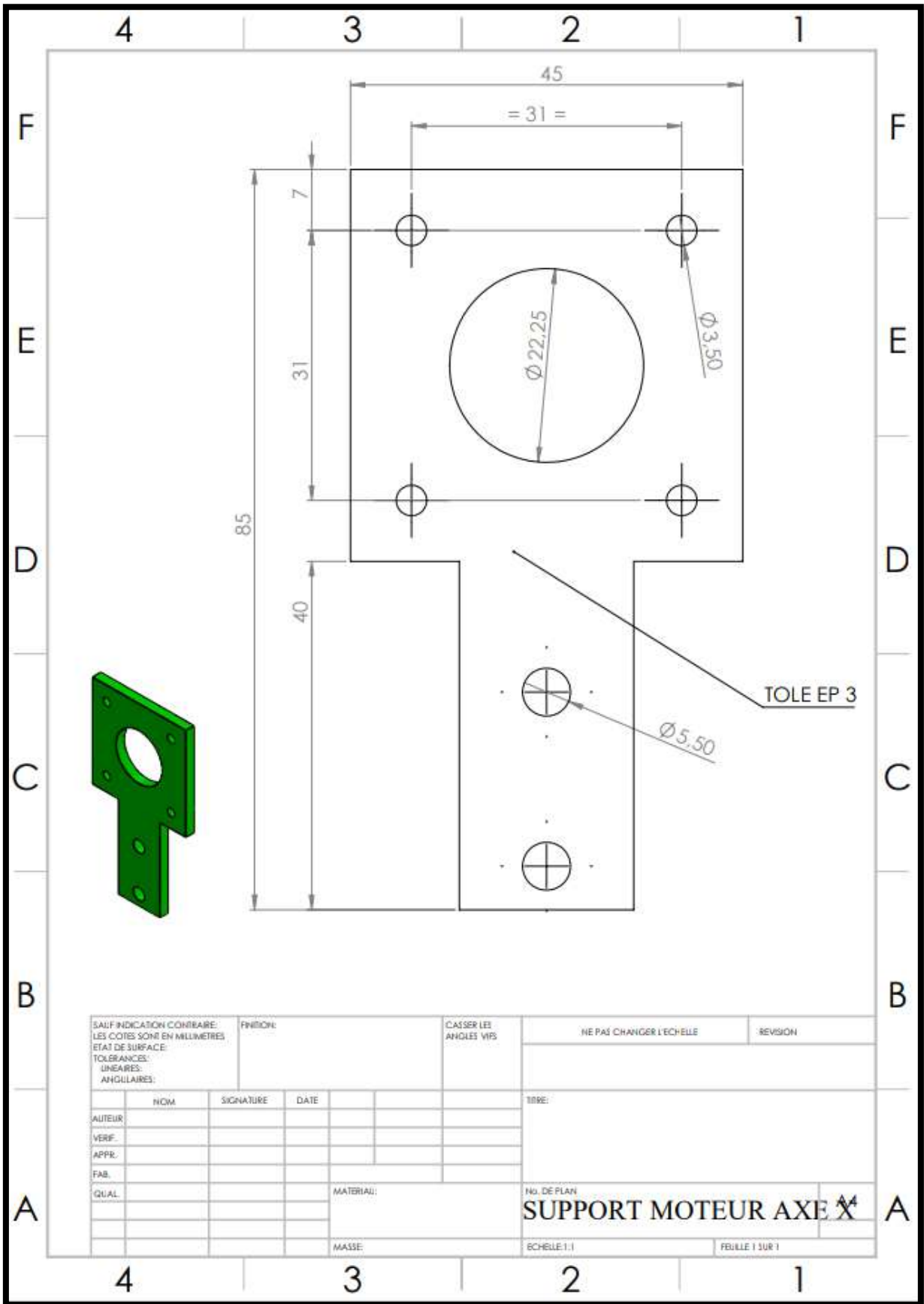


Figure 17 : SUPPORT MOTEUR AXE X

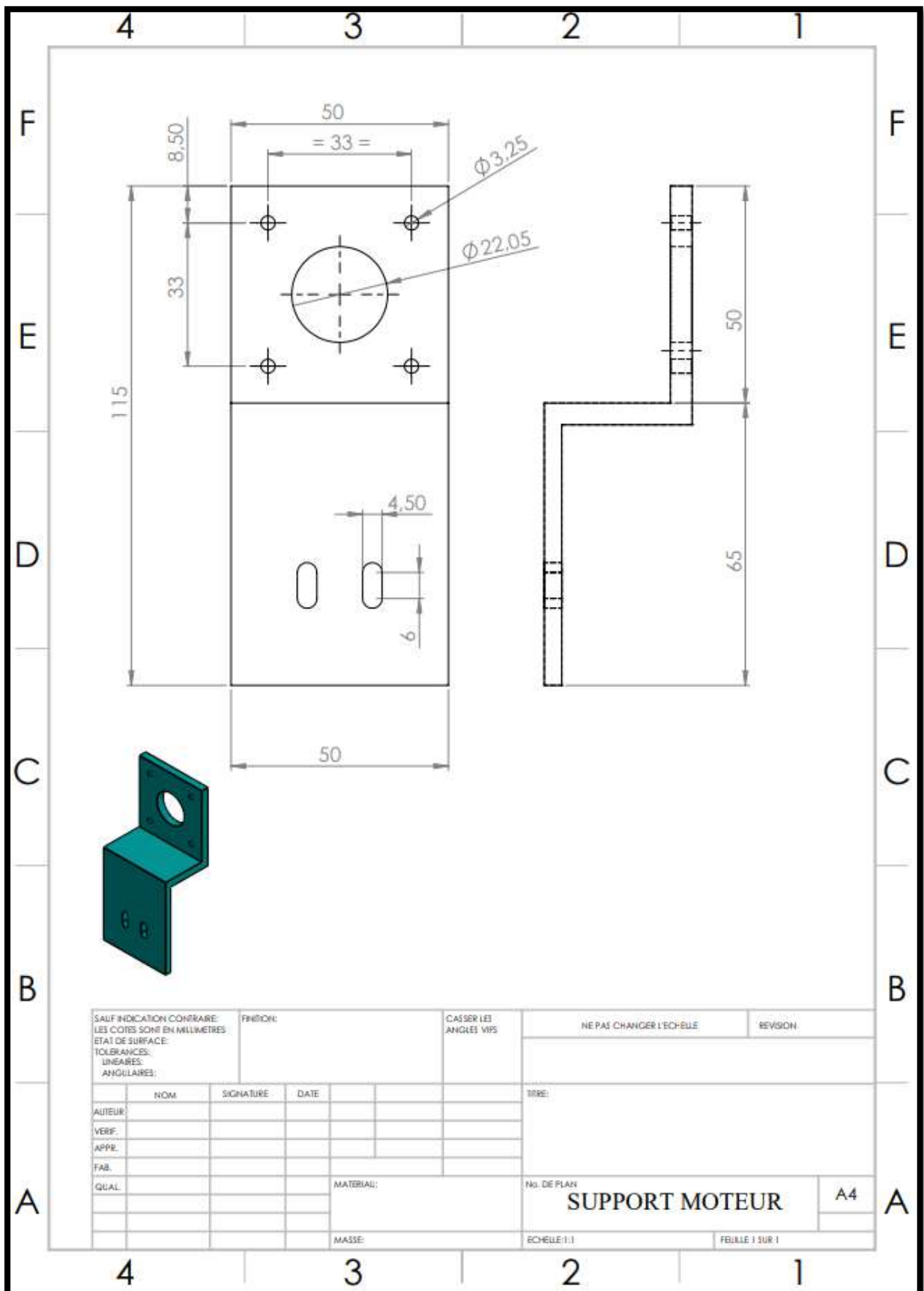


Figure 18 : SUPPORT MOTEUR

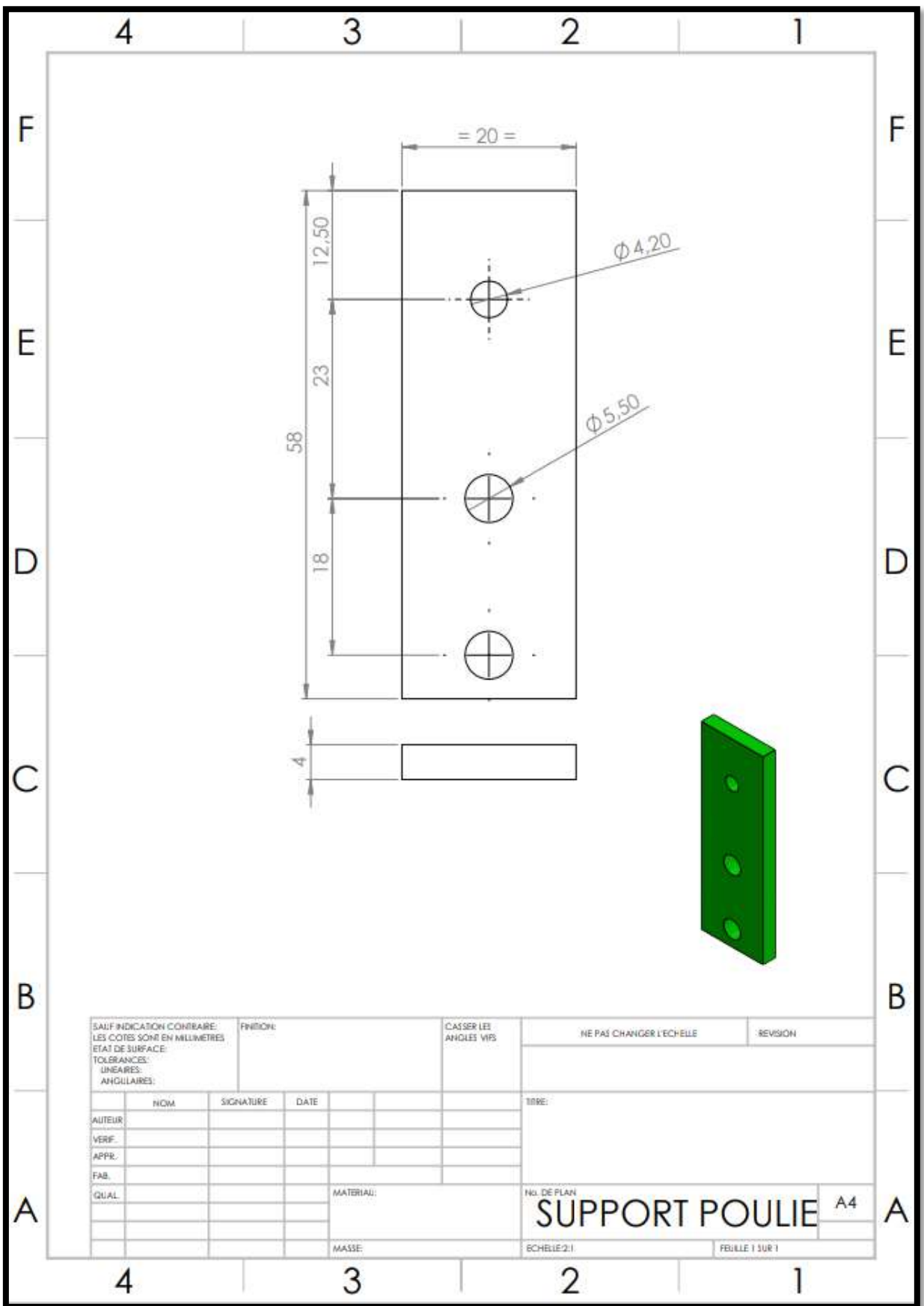


Figure 19 : SUPPORT POULIE

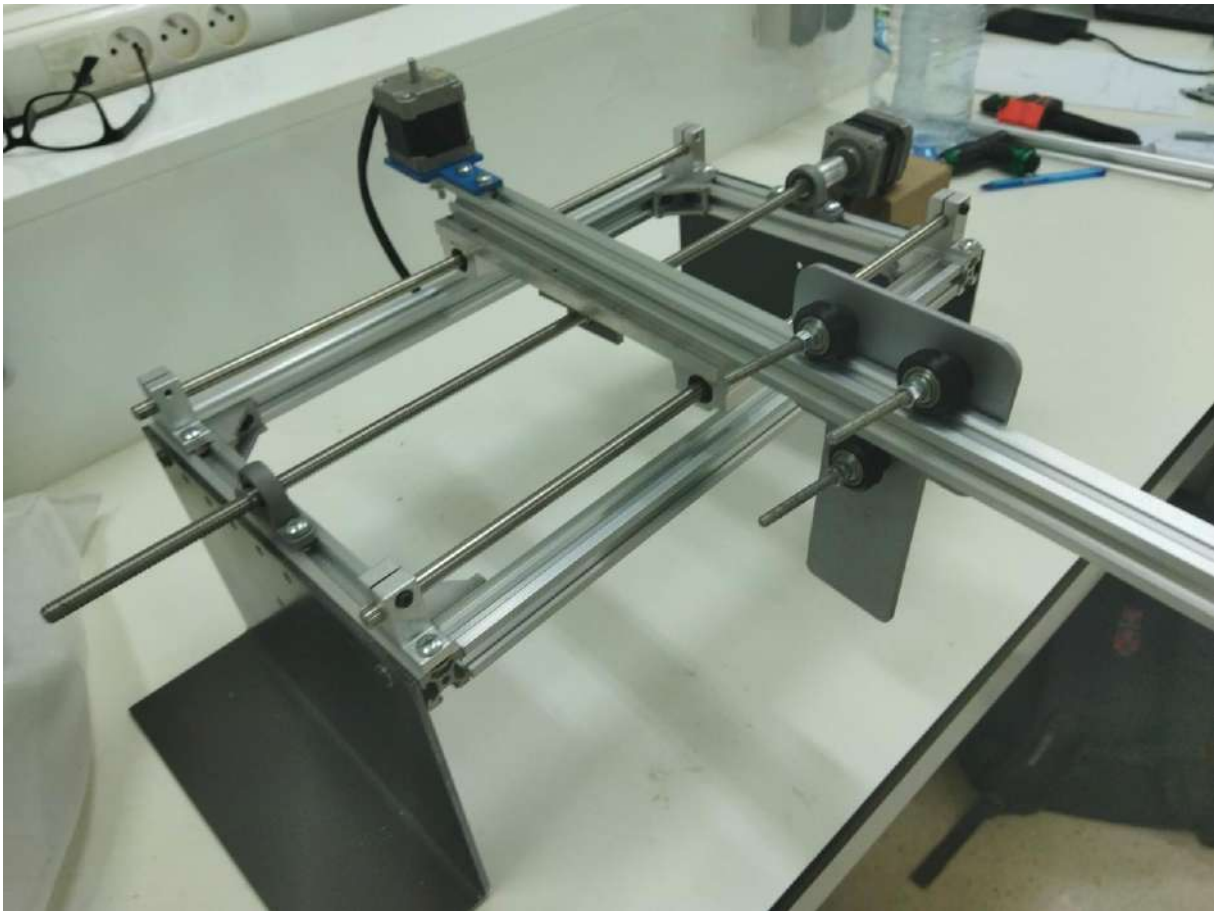


FIGURE 21 : MACHINE CNC APRES MONTAGE

Dans ce chapitre nous avons vu comment le projet de machine s'est construit de pièce en pièce. Ainsi que l'utilité de chaque élément. Et au finale comment chaque élément prend sa place dans le montage définitif.

Conclusion générale

Notre projet de fin d'études a porté sur la conception et la fabrication d'une machine à commande numérique (CNC) à base d'Arduino. Cette machine est capable de réaliser des trajectoires selon deux directions X et Y. En conclusion, les acquis de ce projet ainsi que les étapes suivies lors de sa réalisation sont énumérées comme suit :

- Une maîtrise du fonctionnement ainsi que la commande des moteurs pas à pas
- La maîtrise de la conception mécanique d'une machine CNC sur SolidWorks.
- Fabrication et réalisation des différentes parties de la machine en regroupant différents aspects mécanique et électromécanique (hardware et software).

Ce projet nous a servi à la découverte et l'apprentissage de plusieurs domaines d'études tels que la programmation des Arduino, la réalisation des différentes parties mécaniques, la programmation des moteurs pas à pas, ainsi que la conception et assemblage de différentes pièces mécanique à base de SOLID WORKS.

Références

- [1] Chanez Guerrouabi , Roza Ait Rahmane.2018. Etude et conception d'une machine CNC (Découpeuse laser). Mémoire de Master, Université Mouloud Mammeri, Tizi-Ouzou,3p.
- [2] Abdennadji Youssef, Ahmadi Anouar.juin 2010, Conception et réalisation d'une commande numérique d'une machine de decoup laser , Ingénieur d'école nationale de Gabés.
- [3] D. Gelin, M. Vincent. Mars 1995. Éléments des fabrications, Edition marketing, paris
- [4] Christian Tavernier « Arduino Maitrise sa programmation et ses cartes d'interface (shield) », paris 2011, DUNOD
- [5] www.arduino.cc
- [6] <http://www.ELECTRONIQUEPRATIQUE.com>
- [7] Christian Tavernier « Arduino Maitrise sa programmation et ses cartes d'interface (shield) », paris 2011, DUNOD