

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mouloud MAMMARI de Tizi-Ouzou



Faculté de génie électrique et informatique
Département Automatique

Mémoire de Fin d'Etudes

En vue de l'obtention du diplôme

De Master professionnel en Automatique

Option : Automatique et Informatique Industrielles

Thème

Commande numérique d'un massicot à base
d'une carte Arduino

Proposé et dirigé par :

M. R. Mellah

Présenté par :

AIMEUR Samira

SETBEL Yamina

Devant le jury composé de :

- Président : M. T. CHELLI
- Promoteur : M. R. MELLAH
- Examineur : M. ALI BAY
- Examineur : M.TAHANOUT

Promotion 2016

Remerciements

Tout d'abord, Nous tenons à remercier DIEU le miséricordieux de nos avoir donné la possibilité de réaliser notre projet, d'arriver à notre Souhails et d'atteindre notre objectifs.

Nos remerciements sont formulés à nos familles pour toutes leurs aides et conseils qui nous ont été très significatives.

Nous exprimons nos profondes gratitude à l'égard de notre promoteur, M.R.MELLAH pour non seulement ses précieux conseils durant de ce travail, mais aussi pour nous avoir fait l'honneur de nous encadrer.

Nous remercions les membres de jury qui ont acceptés de juger ce modeste travail.

Nos remerciements les plus chaleureux s'adressent à M.K.MOULOUDJ pour le temps qu'il nous a consacré, son aide, ses orientations et encouragements, et surtout pour sa patience tout au long de ce travail.

Nous voudrions, à cette occasion, exprimer notre profonde reconnaissance à tous nos enseignants du département Automatique pour leurs collaborations, leurs disponibilités et leurs soutiens moral tout au long de notre étude.

Nous tenons à remercier tous ceux qui ont contribués de loin ou de près pour que ce travail puisse être sous sa forme actuelle, HANICHE Leila.

Enfin, à tous nos amis (es), pour leurs soutien et leurs encouragements.

Dédicace

Je dédie ce modeste travail :

A la mémoire de mon père pour son soutien incessant, son soutien moral, que j'ai tellement souhaité sa présence dans cet événement, je souhaite que dieu lui accorde sa miséricorde et l'accueille dans son vaste paradis.

A ma mère qui es tant à ma réussite, et consenti beaucoup de sacrifices pour me permettre de réaliser mes objectifs.

A ma sœur : Hadjila, et mes frères : Kadar et Ali.

A mon très cher fiancé Sofiane.H et à toute ma belle famille.

A mes très chères amies : Leila et Djouher.

A tous mes amis(es), et à toute la promotion 2016.

A mon binôme : Yamina.

Et sans oublier mon promoteur Monsieur MELLAH.R et Monsieur MOULOUDJ.K,

Samira

Dédicace

Je dédie ce modeste travail :

A la meilleure et chère maman au monde pour son amour et son soutien et sa patience.

A mon très cher père qui m'a tout offert.

A mes très chers frères : Amrane, Amar, Mustapha, Achene, Hocine et leurs familles.

A ma très chère sœur et sa belle famille.

A toute la promotion « 2015-2016 ».

A mes amis(es) en particulier Meriem et Kahina et ma chère amie Doudou et à tous ce qui se reconnaîtront en ce mot « Ami »

A mon promoteur Monsieur MELLAH.R et à Monsieur MOULOUDJ.K

Et à mon binôme SAMIRA.

Ainsi qu'à tous ceux qui m'ont aidé de près ou de loin pour l'élaboration de ce travail.

YAMINA

Liste des figures

Figure I.1 : Massicot manuel	4
Figure I.2 : Massicot électrique	5
Figure I.3 : Massicot industriel	6
Figure I.4 : Massicot d'imprimerie.....	6
Figure I.5 : Lames du massicot	7
Figure I.6 : Vue générale des composants d'un massicot	9
Figure II.1 : Architecture interne de la carte Arduino Méga	14
Figure II.2 : Architecture de la carte Arduino Méga	15
Figure II.3 : Microcontrôleur ATMEGA2560	16
Figure II.4 : Interface IDE Arduino	19
Figure II.5 : L'écriture du programme Arduino	23
Figure II.6 : Compilation du programme	23
Figure II.7 : Console d'affichage	23
Figure II.8 : Sélection de la cible	24
Figure II.9 : Sélection du port	24
Figure II.10 : Transfert du programme	25
Figure II.11 : Clignotement des deux LED du port série	25
Figure III.1 : Diagramme de connexion du dispositif de contrôle et de commande	27
Figure III.2 : La carte Arduino Méga2560	28
Figure III.3 : Afficheur LCD (2x16)	28
Figure III.4 : Commande d'un afficheur LCD en mode 4 bits	30
Figure III.5 : Commande d'un afficheur LCD en mode 8 bits	31
Figure III.6 : Schéma de branchement de l'afficheur LCD à Arduino	31
Figure III.7 : Le clavier matriciel (4x4)	32
Figure III.8 : Moteur réducteur avec encodeur	33
Figure III.9: Structure du pont H	33
Figure III.10 : Structure du pont H électrique (pont H de base)	34
Figure III.11 : La carte de puissance L293D	35

Figure III.12 : Les broches du boitier L293D.....	36
Figure III.13 : Capteur de fin course.....	38
Figure III.14 : Disque d'un codeur incrémental.....	39
Figure III.15 : Particularités de fonctionnement d'un encodeur incrémental.....	40
Figure III.16: Le principe physique de la machine à courant continu.....	40
Figure III.17 : Schéma bloc du MCC.....	42
Figure III.18 : Schéma bloc de l'asservissement du courant.....	43
Figure IV.1 : Fenêtre du module de simulation Proteus ISIS	46
Figure IV.2 : Fenêtre du module Proteus ARES	46
Figure IV.3 : Schéma synoptique du système	47
Figure IV.4 : Schéma de la carte d'alimentation sur ISIS.....	48
Figure IV.5 : Implantation des composants	48
Figure IV.6 : Circuit imprimé sans composants	48
Figure IV.7 : Photo de la carte d'alimentation réalisée	49
Figure IV.8 : Schéma de la carte de puissance sur ISIS	49
Figure IV.9 : Implantation des composants	50
Figure IV.10 : Circuit imprimé sans composants	50
Figure IV.11: Photo de la carte réalisée	50
Figure IV.12 : Organigramme fonctionnel du système	52
Figure IV.13 : Les valeurs données par l'encodeur	54
Figure IV.14 : La maquette réalisée	55

Liste des tableaux

Tableau II.1 : Description des icones de la barre de menus de l'interface IDE Arduino	19
Tableau II.2 : Description des icones de la barre de boutons de l'interface IDE Arduino	20
Tableau II.3 : Description des commandes de programme Arduino	22
Tableau III.1 : Description des pins de l'afficheur LCD	27
Tableau III.2 : Le fonctionnement des relais du pont H	33
Tableau III.3 : Description des broches de L293D	35
Tableau III.4 : Les caractéristiques de L293D	35
Tableau III.5 : La logique de commande Inputs de L293D	36

SOMMAIRE

Introduction générale	1
------------------------------------	---

Chapitre I : Présentation du massicot

I.1 Introduction	3
I.2 Définition d'un massicot	3
I.3 Historique du massicot	3
I.4 Les différents types du massicot	4
I.4.1 Les massicots manuels	4
I.4.2 Les massicots électriques	5
I.4.3 Les massicots industriels	5
I.4.4 Les massicots d'imprimerie	6
I.5 Principe de fonctionnement	7
I.6 La constitution d'un massicot	7
I.6.1 La lame	7
I.6.2 Les pieds supports	8
I.6.3 les réglettes ou liteaux de coupe	8
I.6.4 Les buses soufflantes.....	8
I.6.5 L'équerre	8
I.6.6 Le moteur	8
I.7 La sécurité	9
❖ Les dispositifs de sécurité	9
I.8 Le choix d'un massicot	10
I.9 Conclusion.....	11

Chapitre II : Présentation de l'Arduino

II.1 Introduction	12
II.2 Présentation de l'Arduino	12
II.3 Le choix de la carte Arduino	12
II.4 Les gammes de la carte Arduino	13
II.5 Les outils de la carte Arduino MEGA2560	14
II.5.1 Architecture HARDWARE de la carte Arduino MEGA2560	14
II.5.2 Architecture de la carte Arduino MEGA2560	15

II.6 Constitution de la carte Arduino	15
II.6.1 Partie matérielle	15
II.6.1.1 Le microcontrôleur ATMEGA2560	15
A. Les caractéristiques du microcontrôleur ATMEGA2560	16
B. La structure interne du microcontrôleur ATMEGA2560.....	17
II.6.1.2 Les entrées sorties.....	17
II.6.1.3 L'alimentation	17
II.6.1.4 Visualisation	18
II.6.1.5 La connectique	18
II.6.2 Partie programme (SOFTWARE)	18
II.6.2.1 Présentation de l'espace de développement intégré (EDI) Arduino	19
II.6.2.2 Différents composants de l'espace de développement intégré (EDI).....	19
II.6.2.2.1 La barre de menus	19
II.6.2.2.2 La barre de boutons	20
II.6.2.2.3 Structure d'un programme Arduino	20
II.6.2.2.4 Le programme Arduino	20
A. Description de la structure d'un programme Arduino	20
B. Le jeu d'instructions du langage Arduino	21
➤ Les syntaxes du programme	21
➤ Les commandes du programme Arduino	21
C. Les étapes à suivre pour programmer la carte	22
➤ L'écriture du programme	22
➤ Compilation du programme	23
➤ Sélection de la cible (type de carte)	24
➤ Sélection du port	24
➤ Transfert du programme	25
II.7 Conclusion	25

Chapitre III : Présentation du matériel

III.1 Introduction	25
III.2 Première partie : Présentation du matériel	25
III.2.1 Carte d'acquisition	25
III.2.2 Afficheur LCD	26

III.2.2.1	Description des pins	27
III.2.2.2	Fonctionnement et Commande de l’afficheur LCD	27
1.	Fonctionnement de l’afficheur LCD	27
2.	Commande d’un afficheur LCD	28
A/	Commande en 4 bits	28
B/	Commande en 8 bits	28
III.2.2.3	Branchement de l’afficheur à la carte Arduino	29
III.2.3	Le clavier	30
III.2.3.1	Description de clavier	30
III.2.4	Le moteur utilisé	31
III.2.5	Présentation du pont H	31
III.2.5.1	Technique du pont H de base	32
III.2.6	Présentation du boîtier L293D	33
III.2.6.1	Le brochage	34
III.2.6.2	Description des broches du L293D	34
III.2.6.3	Les caractéristiques du L293D	35
III.2.6.4	Logique de commande Input	36
III.2.7	Les capteurs	36
III.2.7.1	Les capteurs de fin de course	36
III.2.7.2	L’encodeur incrémental	37
❖	Principe de fonctionnement	37
III.3	Deuxième partie : Description du moteur à courant continu (MCC)	38
III.3.1	Fonctionnement et constitution du MCC	38
III.3.2	Modélisation et asservissement du MCC	39
III.3.2.1	Modélisation du MCC	39
III.3.2.2	Asservissement du MCC	40
III.3.3	La technique de la PWM pour la variation de vitesse	41
III.4	Conclusion	42
Chapitre IV : Conception et réalisation		
IV.1	Introduction	45
IV.2	Conception	45
IV.2.1	Présentation de Proteus	45
IV.2.1.1	ISIS	45

IV.2.1.2 ARES	46
IV.2.2 Conception et simulation des cartes	47
IV.2.2.1 Présentation globale du système	47
IV.2.2.2 Les circuits imprimés des cartes	47
❖ La carte d'alimentation	47
❖ La carte de puissance	49
IV.3 Partie programmation	51
IV.3.1 Cahier de charge	51
IV.3.2 Organigramme fonctionnel de notre système	52
IV.3.3 Réalisation du programme	53
IV.3.4 Organigramme du programme	53
IV.3.5 Les fonctions utilisées	54
IV.3.6 Tests	54
IV.3.7 : La maquette réalisée	55
IV.4 Conclusion	55
Conclusion générale	56

Introduction générale

Introduction générale

La commande numérique est une technique relativement récente, qui consiste à commander le fonctionnement d'une machine à partir d'un programme, sans intervention directe de l'opérateur pendant l'exécution. Avec les progrès de la microélectronique et de l'informatique industrielle, dont les coûts n'ont cessé de baisser depuis des années alors que les performances et la convivialité ont augmenté de façon notable, ce mode de commande se démocratise et elle est de plus en plus présente dans les ateliers. Cette évolution technologique a permis de remplacer les commandes en logique câblée par des équipements à microprocesseurs qui assurent, en temps réel, des calculs qui, antérieurement, devaient être effectués lors de la programmation. L'aide informatique à la programmation était faible et le travail de programmation était parfois fastidieux.

Aujourd'hui, après l'apparition des aides performantes sur micro-ordinateurs, la programmation paramétrée permet une adaptation des procédures au métier. Les machines-outils à commande numérique (MOCN) sont des machines dotées d'une commande numérique, qui désignent l'ensemble des matériels et logiciels ayant pour fonction de donner les instructions de mouvements à tous les organes de la machine.

La création des MOCN correspond, par nature à des besoins en usinage de pièces de petites et moyennes séries. Ainsi, l'évolution des machines-outils à commande numérique, en usinage mécanique s'inscrit dans une évolution continue vers une automatisation industrielle de plus en plus intégrée.

L'objectif principale de notre travail est alors la commande numérique d'un massicot à base d'une carte Arduino, dont le but est d'automatiser cette machine et on a proposé un capteur de position (encodeur) pour pouvoir contrôler la position de papier sans l'intervention de l'utilisateur, et cela va faciliter les tâches et garantir une précision de découpe. Le projet s'articule ainsi autour d'une réalisation.

Pour mener à terme notre travail, nous avons organisé notre mémoire de la façon suivante :

Le premier chapitre porte sur la présentation des massicots. On y trouvera le principe de fonctionnement, les différents types des massicots, la constitution et enfin le choix de massicot à utiliser.

Nous présentons dans le deuxième chapitre la carte de commande Arduino, et le logiciel de programmation IDE Arduino.

Introduction générale

Le troisième chapitre aborde la présentation des matériels utilisés, selon le principe de fonctionnement et la commande de chaque composant.

Le quatrième chapitre présente la conception et la réalisation de la maquette.

Nous clôturons notre mémoire par une conclusion générale.



Chapitre I

Présentation du Massicot

I.1 Introduction :

La révolution industrielle du XIX^e siècle favorise l'émergence de l'information et du savoir technique ou culturel, la demande en livres est de plus en plus importante. Ils sont très recherchés par les étudiants dont le nombre augmente dans toute l'Europe et par les populations des villes en plein essor économique, ainsi de la bourgeoisie marchande. Des ateliers de copie se créent mais ce système est lent et les livres restent encore rares et encore trop chers pour la plupart des lecteurs.

A fin de répondre à ce besoin culturel, l'homme a pensé de développer ces techniques pour produire un produit qui satisfait le consommateur soit du côté qualité ou du côté quantité. Il a développé l'imprimerie, le papier et ensuite la découpe du papier.

Pour une réalisation plus aisée de découpe, il est nécessaire d'inventer une machine qui lui permet de réaliser une découpe nette et précise de différents documents et aussi d'améliorer la capacité de coupe qui veut dire combien de feuilles découpées en une fois et de faire des documents de différents formats : A4, A3, A2, etc, la machine qui lui a permis de répondre à toutes ces exigences est le massicot papier.

I.2 Définition d'un massicot :

Le Massicot est une machine destinée à couper selon un angle droit du papier et d'autres matériaux comme les plaques métalliques par exemple. Ces machines se trouvent surtout dans le secteur de l'imprimerie. Cependant on trouve des modèles de massicot de plus en plus variés. Les plus petits (d'une ouverture à partir de 66 cm) peuvent être utilisés dans des imprimeries intégrées ou des entreprises ayant à produire des formats plus petits. Quant aux massicots plus grands, d'une ouverture allant jusqu'à 2 m, on les trouve souvent dans des systèmes interconnectés d'une manière complexe.

Il existe, également, des massicots à plusieurs lames, pour pouvoir faciliter la découpe de certains travaux comme le massicot trilatéral pour les livres [1].

I.3 Historique du massicot :

Vers 1830, on commence à utiliser des machines pour couper le papier. En 1837, Thirault construit une machine à lame fixe. En 1844, Guillaume Massiquot (1797-1870) dépose un brevet pour sa machine. « Massicot » est ainsi tiré du nom de l'imprimeur Guillaume Massiquot.

Avant de breveter son invention Massiquot avait construit un modèle à levier tel qu'on peut le voir encore de nos jours : le levier tire une bielle qui entraîne un porte-lame jusqu'à la table et le remonte. Par contre, le massicot breveté était entraîné par un volant et un engrenage manuel : une presse bloque la pile de papier, tandis qu'une lame à mouvement oblique descend pour couper le papier. Le principe de coupe est celui de la lame contre support différent de celui de la cisaille qui utilise le principe de lame et contre-lame.

Ce principe est toujours celui des massicots contemporains. Les massicots sont dès le XIX^e siècle perfectionnés par les fabricants *Fomm* et *Krause* en Allemagne, *Furnival* en Angleterre, *Oswego* et *Seybold* aux États-Unis [1].

I.4 Les Différents types de massicot :

I.4.1 Les massicots manuels :

Les massicots manuels sont des machines de format A3. Ce type se caractérise par un système d'entraînement de la lame mécanique, comme montre la figure I.1.

La capacité de coupe d'un massicot manuel varie entre 15 mm et 80 mm soit environ de 150 feuilles à 180 feuilles selon les modèles et une longueur qui varie entre 370 mm et 480 mm.

De nombreux massicots manuels sont des massicots de table, c'est-à-dire que celui-ci pourra se poser sur n'importe quelle surface plane (table, bureau, ...). Le stand (pieds support) étant le plus souvent en option.

Ce massicot peut être utilisé dans de nombreux secteur d'activité, comme :

- Les administrations (école, mairies, etc.),
- La reprographie,
- l'imprimerie,
- les ateliers de façonnage [2].



Figure I.1 : Massicot manuel

I.4.2 Les massicots électriques :

Les massicots électriques sont très puissants mais aussi précis. Ils offrent un grand rendement qui les rendent très utiles dans les ateliers d'imprimeries principalement, en raison de leur capacité de coupe qui varie entre 500 à 550 feuilles simultanées.

Le massicot électrique se caractérise par le fait que le système de levier de la lame est doté d'un moteur électrique. Par ailleurs, ils sont équipés d'un affichage digital qui indique des dimensions, et équipé également de carte de protection, et d'un frein à disque pour arrêter la lame instantanément [2].



Figure I.2 : Massicot électrique

I.4.3 Les massicots industriels :

Massicot industriel est généralement d'un grand format, lui permettant de pouvoir faire différentes découpes, sur plusieurs matériaux (feuilles, cartons, plastique,..). Le tout est effectué de manière automatisé avec un débit bien plus important qu'un massicot de bureau.

Le massicot industriel donné en figure I.3 peut être utilisé dans une imprimerie intégrée, un service prépresse, ou un copy shop, pour une utilisation professionnelle. Ils permettent de couper des longueurs de papier de 520 mm à 720mm.

La plus part des massicots industriels possèdent un barrage immatériel par cellule photo-électrique sur la table avant, qui arrête la descente de lame lors de la détection de mouvement. Certains possèdent même des freins à disque pour l'arrêt instantané de la lame [2].

La figure I.3 présente un massicot industriel.



Figure I.3 : Massicot industriel

I.4.4 Les massicots d'imprimerie:

Le massicot d'imprimerie représenté en figure I.4, est une grande machine de découpe selon des angles droits. Il peut, également servir pour la découpe de matériaux souples.

Les principales actions du massicot d'imprimerie c'est la préparation des feuilles et leurs mises au format, ainsi que le découpage et le façonnage des imprimés. En générale les massicots d'imprimerie ont une ouverture comprise entre 60 et 115 cm, sa table soufflante permet d'utiliser des piles de papier plus facilement.

On peut trouver des massicots d'imprimeries mono-lames et multi-lames, mais la plupart des massicots sont mono-lames. Les multi-lames, qui sont plus rares et chères, peuvent être utilisées pour des travaux spécifiques (ex. tri-lame pour la découpe de livres...) [2].



Figure I.4 : Massicot d'imprimerie

I.5 Principe de fonctionnement :

Le massicot se compose d'un bâti solide supportant une table de travail surmontée d'une arche qui contient la presse et la lame. Des tables latérales reçoivent éventuellement les piles de papier à ronger.

A l'arrière de la table se trouve une équerre dont la position est réglable de manière très précise, dans le but de définir la largeur de coupe. La pile de papier à couper vient en butée sur cette équerre et des repères permettent de contrôler l'alignement des repères de coupe imprimés. La table de travail est percée de buses soufflantes qui, grâce à un matelas d'air, facilitent le déplacement du papier.

Enfin, l'opérateur, par sécurité, appuie simultanément sur une pédale et sur deux boutons qui déclenchent la descente de la presse et de la lame.

I.6 La constitution d'un massicot:

I.6.1 La lame :

Un massicot ne peut être efficace que si sa lame est parfaitement entretenue. La lame massicot est en effet l'élément le plus important et son rôle dans le bon fonctionnement de l'appareil est déterminant. Cette lame est soumise à l'usage, notamment lorsque le massicot sert à couper des cartons épais. Il est absolument nécessaire de changer et d'affûter régulièrement la lame du massicot car une lame émoussée perd en qualité de précision de découpe [2].

La figure I.5 représente les lames du massicot.



Figure I.5 : Lames du massicot

I.6.2 Les pieds supports :

Les pieds supports pour un massicot sont destinés avant tout à optimiser le confort de l'utilisateur et donc sa sécurité. Très solides, ils sont fabriqués en métal et assurent une robustesse certaine à l'appareil. Ce dernier est rehaussé à bonne hauteur grâce aux pieds support ce qui garantit une position confortable à l'utilisateur [2].

I.6.3 Les réglettes ou liteaux de coupe :

La réglette joue un rôle de protection de la lame du massicot : elle est en fait une tige en caoutchouc qui empêche la lame de se cagner contre le métal de la table du massicot. Le fil de la lame massicot, très tranchant, est aussi très fragile et la réglette lui assure une protection indispensable [2].

I.6.4 Les buses soufflantes:

Il s'agit là d'une innovation technique destinée à faciliter le travail du massicot tout en optimisant sa sécurité. Les buses soufflantes, insérées dans la table, produisent un flux d'air qui agit comme un cousin d'air : l'ouvrier déplace ainsi sans effort les piles de papier. L'efficacité du flux d'air est décuplée par la proximité des buses soufflantes les unes par rapport aux autres : plus elles seront éloignées, moins le flux d'air sera puissant [2].

I.6.5 Le moteur :

Un moteur est une machine électrique servant à transformer l'énergie électrique en énergie mécanique. Il est basé sur le principe de l'action d'un champ tournant statorique et l'induction des courants rotoriques.

I.6.6 L'équerre :

L'équerre se trouve à l'arrière de la table latérale, et elle permet de régler la position et de définir la largeur de coupe.

La figure I.6 présente la vue générale des composants d'un massicot.

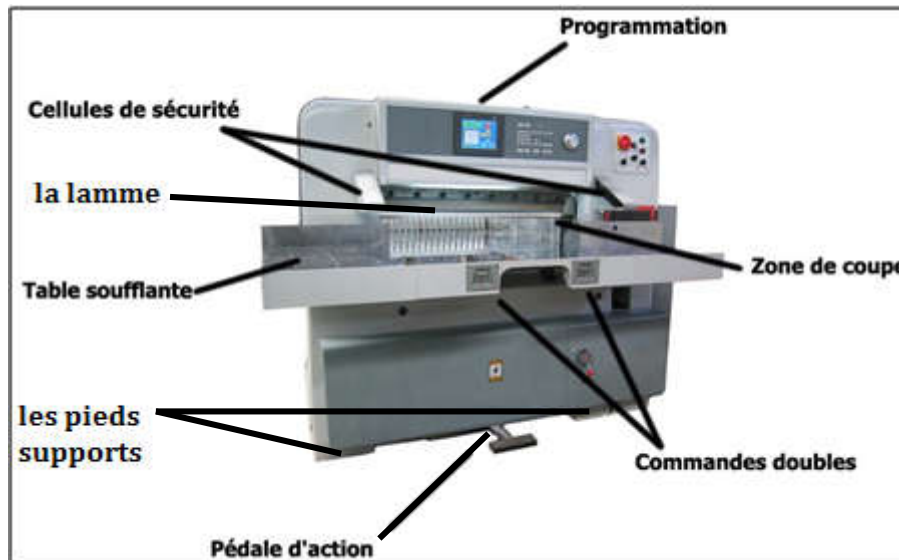


Figure I.6 : Vue générale des composants d'un massicot.

I.7 La sécurité :

Les massicots dans le passé ont provoqué de nombreux accidents. C'est pourquoi les mesures de sécurité sont particulièrement renforcées : obligation d'utiliser ses deux mains et un pied pour déclencher la coupe, protection par détecteurs infrarouges, le démontage du capot protecteur empêche toute mise en marche, et de nombreux boutons d'arrêt « coup de poing » en vue de garantir la sécurité au maximum [2].

I.7.1 Les dispositifs de sécurité :

Les fabricants de massicots pour papier ont doté leurs modèles de système visant à assurer une sécurité optimale aux utilisateurs tels que :

- Les pieds supports pour massicots : robustes et solides, ils empêchent tout basculement de la machine en l'écran au mieux dans le sol.
- Les réglettes pour massicots : ils protègent la lame de massicot papier, particulièrement tranchante en l'empêchant de buter contre la table.
- Un verrou de sécurité est placé sur chaque massicot de bureau : il est nécessaire de l'activer avant chaque utilisation car il débloque le levier de coupe.
- Les massicots sont munis de carters de protection avant et arrière selon la norme CE de janvier. Le carter placé à l'avant de la machine se verrouille pendant la coupe, empêchant ainsi tout basculement.
- Les massicots industriels sont dotés de dispositif de mise hors tension de l'appareil en cas de dysfonctionnement.

- Les massicots industriels sont munis de deux boutons qui doivent être actionnés en même temps pour déclencher l'action de coupe.

A fin de garantir une sécurité optimale lors des opérations de changements de la lame de massicot papier, un dispositif de protection muni d'un accrochage protège le tranchant de la lame. Dans le même esprit, les tablettes se changent sur les côtés de l'appareil. Ces systèmes permettant de raccourcir efficacement les opérations de maintenance des massicots papier [2].

I.8 Le choix d'un massicot :

Pour bien faire choisir notre massicot, il est primordial avant toute acquisition de prendre en compte plusieurs critères. Les principaux sont la longueur maximale des papiers ou documents ; la profondeur de la table, la capacité du massicot ainsi que le type massicot.

- **La longueur maximale des documents :**

Ce premier critère est très important, raison pour laquelle il est primordial de l'analyser avec soin. Il ne sert à rien d'acquérir le meilleur massicot qui soit, s'il ne peut réaliser les tâches pour lesquelles il a été acquis. Il est donc avant toute chose référer à la longueur de coupe du massicot.

-Pour la coupe de document du format A10 au A4, du B10 au B4 et du C10 au C4, il faudra un massicot ayant une longueur de découpe allant jusqu'à 370mm.

-Pour la coupe de papier du format A10 au A3, du B10 au B4 et du C10 au C4, il faudra un massicot ayant une longueur de découpe allant de 430 à 435mm.

Après avoir déterminé la longueur de découpe qui nous convient, on passe au second critère qui est lui aussi d'une grande importance.

- **La profondeur de table :**

Au premier abord, ce critère peut paraître de moindre importance pour certaines personnes, mais à y regarder de près, il est très essentiel. C'est en effet ce paramètre qui nous permettra d'avoir l'assurance de pouvoir couper nos documents à la taille souhaitée.

Pour un choix adapté, la profondeur de notre table doit être au moins égale à la longueur du document final. Dans le cas où on souhaite par exemple effectuer dans les deux sens la découpe d'un document A3, la profondeur de la table ainsi que la longueur de la coupe doivent être supérieures à 430mm.

▪ La capacité de coupe du massicot :

Pour ce type d'appareil, l'ouverture de coupe est d'une grande importance car c'est ce qui nous permet de savoir combien de feuilles le massicot est en mesure de découper en une seule fois. Un massicot d'une capacité de coupe de 15mm par exemple, ne peut couper que jusqu'à 150 feuilles de 70g/m². Si nous souhaitons par contre couper 200 feuilles en même temps, il nous faudra un massicot d'une capacité de 20 mm au moins. Il nous revient donc de déterminer la capacité de coupe qui nous convient le mieux avant de passer à l'achat de notre massicot.

I.9 Conclusion :

Le massicot est un outil de découpe qui simplifiera nos techniques de découpe. Pour nos pièces décoratives, nos tirages photos et nos documents papier, il est un élément indispensable à toute scrapbookeuse.

Le massicot est pour nos créations car il nous assure une parfaite stabilité et une adhérence à notre plan de travail. Il nous assure également une très grande sécurité pour toutes les découpes, jusqu'au format A4. Ainsi grâce au massicot, notre découpe nette et précise de différents documents, ainsi par l'assurance de pouvoir couper nos documents à la taille souhaitée par la détermination de la longueur de découpe qui nous convient et il nous offre aussi une capacité de coupe.

Dans ce chapitre, notre étude s'est localisée sur la présentation du massicot papier, Le prochain chapitre sera consacré à la présentation de la carte électronique qui est la carte Arduino.

Dans ce chapitre, notre étude s'est localisée sur la présentation du massicot papier. Le prochain chapitre sera consacré à la présentation de la carte électronique qui est la carte Arduino.



Chapitre II

Présentation de l'Arduino

II.1 Introduction :

Aujourd'hui, l'électronique est de plus en plus remplacée par de l'électronique programmée. On parle aussi de système embarquée ou d'informatique embarquée. Son but est de simplifier les schémas électroniques et par conséquent réduire l'utilisation de composants électroniques, réduisant ainsi le coût de fabrication d'un produit ; il en résulte des systèmes plus complexes et performants pour un espace réduit.

Depuis que l'électronique existe, sa croissance est fulgurante et continue encore aujourd'hui. L'électronique est devenue accessible à toutes personnes ayant l'envie : ce que nous allons apprendre dans ce travail est un mélange d'électronique et de programmation.

On va en effet parler d'électronique embarquée qui est un sous-domaine de l'électronique et qui a l'habileté d'unir la puissance de la programmation à la puissance de l'électronique.

II.2 Présentation de l'Arduino :

L'ARDUINO est un projet électronique qui est né en hiver 2005 en Italie, il a été créé et perfectionné par une équipe de développeurs, composée de six individus : Massimo Banzi, David Mellis, David Cuartielles, Tom Igoe, Gianluca Martino et Nicholas Zambetti. Cette équipe a créé ce qu'on appelle le 'système Arduino'. C'est un outil qui va permettre aux débutants, amateurs professionnels de créer des systèmes électroniques plus ou moins complexes.

Arduino est donc une plate-forme open source d'électronique programmée qui est basé sur une simple carte à microcontrôleur (de la famille AVR), et un logiciel, véritable environnement du développement intégré, pour écrire, compiler et transférer le programme vers la carte à microcontrôleur. Ces cartes sont basées sur une interface entrée/sortie simple pouvant recevoir des entrées d'une grande variété d'interrupteurs ou de capteurs, et pouvant contrôler une grande variété de lumières, moteurs ou tout autre types d'actionneurs et tous cela se fait sur un environnement de développement proche du langage C [3].

II.3 Le choix de la carte Arduino :

Parmi les caractéristiques les plus importantes de la carte Arduino, nous avons retenu les suivantes :

- **Le prix :** Les cartes Arduino sont relativement peu coûteuses comparativement aux autres plateformes,

- **La compatibilité (Multiplateforme) :** Le logiciel Arduino, écrit en Java, tourne sous les systèmes d'exploitation Windows, Macintosh et Linux. La plupart des systèmes à microcontrôleurs sont limités à Windows,
- **Logiciel Open source et extensible :** Le logiciel Arduino et le langage Arduino sont publiés sous licence open source, disponible pour être complété par des programmeurs expérimentés,
- **Matériel Open source et extensible :** Les cartes Arduino sont basées sur les microcontrôleurs Atmel : ATMEGA8, ATMEGA168, ATMEGA328, etc. Les schémas des modules sont publiés sous une licence Créative Commons, et les concepteurs des circuits expérimentés peuvent réaliser leur propre version des cartes Arduino, en les complétant et en les améliorant,
- **Un environnement de programmation clair et simple :** L'environnement de programmation Arduino (le logiciel Arduino IDE) est facile à utiliser pour les débutants, tout en étant assez flexible pour que les utilisateurs avancés puissent en tirer profit également,
- La carte Arduino est capable de stocker un programme et de le faire exécuter,
- La carte reçoit des informations analogiques ou numériques sur ses entrées et génère des informations analogiques numériques,
- Récupère les données des capteurs pour les transmettre à l'interface de commande (pc) et traduire les instructions pour faire fonctionner les actionneurs.

Finalement nous retiendrons ce projet pour toutes ses qualités.

II.4 Les gammes de la carte Arduino :

Actuellement, il existe plus de 20 versions de module Arduino, nous citons quelques un afin d'éclaircir l'évaluation de ce produit scientifique et académique :

-Le NG d'Arduino, avec une interface d'USB pour programmer et usage d'un Microcontrôleur ATmega8.

-L'Arduino Nano, une petite carte programme à l'aide porte USB cette version utilisant un microcontrôleur ATmega168 (Atmega328 pour une plus nouvelle version).

-L'Arduino Bluetooth, avec une interface de Bluetooth pour programmer en utilisant un microcontrôleur ATmega168.

-L'Arduino UNO, utilisations microcontrôleur ATmega328.

-L'Arduino Méga2560, utilisations un microcontrôleur ATmega2560, et possède toute la mémoire à 256KBS. Elle incorpore également le nouvel ATmega8U2 (ATmega16U2 dans le jeu de puces d'USB de révision 3) [4].

Parmi ces types on a choisi une carte Arduino MEGA2560.

II.5 Les outils de l'Arduino MEGA2560 :

Le système Arduino se repose essentiellement de deux composants principaux qui sont le matériel et le logiciel, soit la carte électronique Arduino et son programme Arduino

II.5.1 Architecture HARDWARE de la carte Arduino Méga:

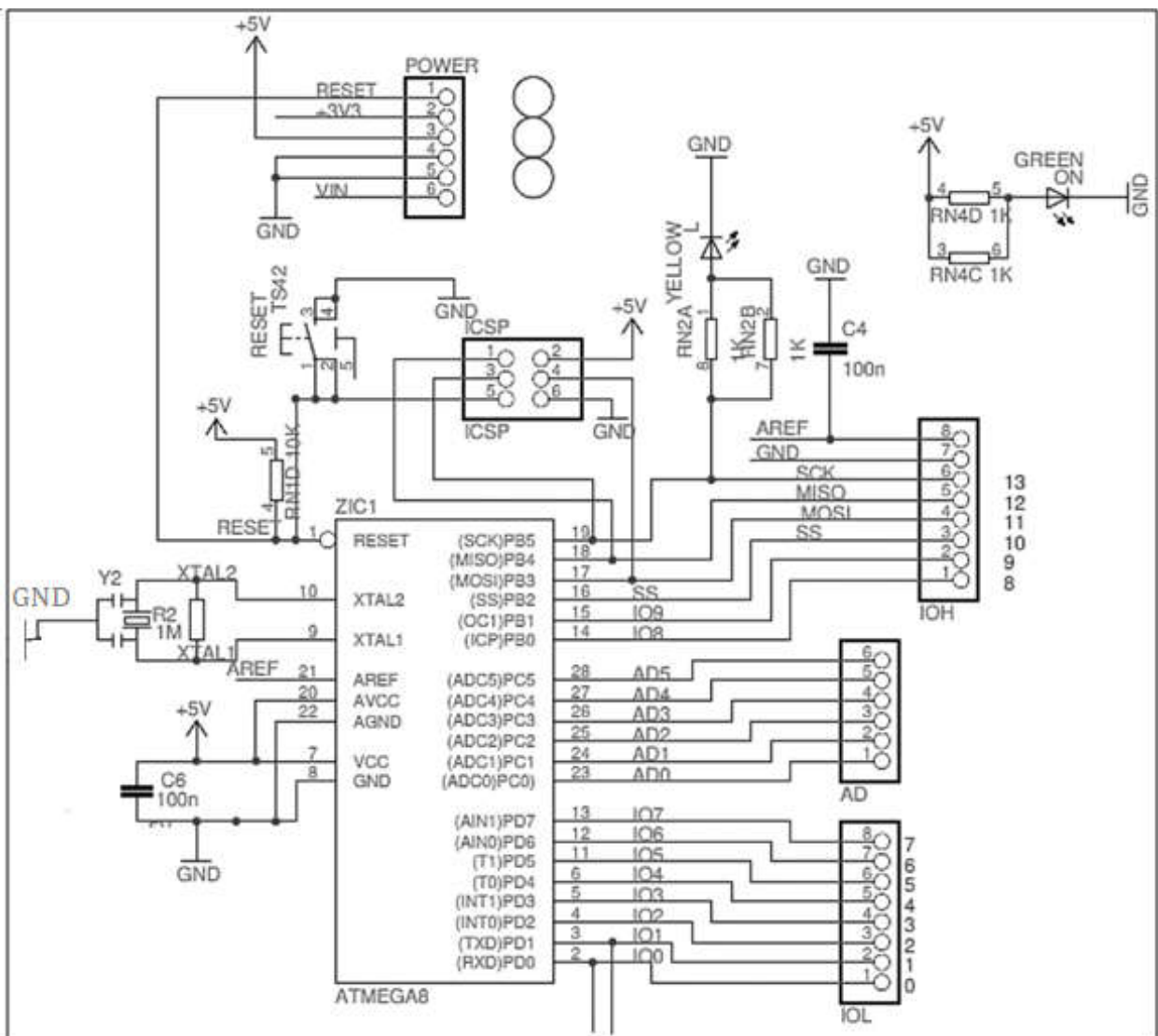


Figure II.1 : Architecture interne de la carte Arduino Méga

II.5.2 Architecture de la carte Arduino Méga :

La figure II.2 illustre le schéma synoptique de la carte Arduino.

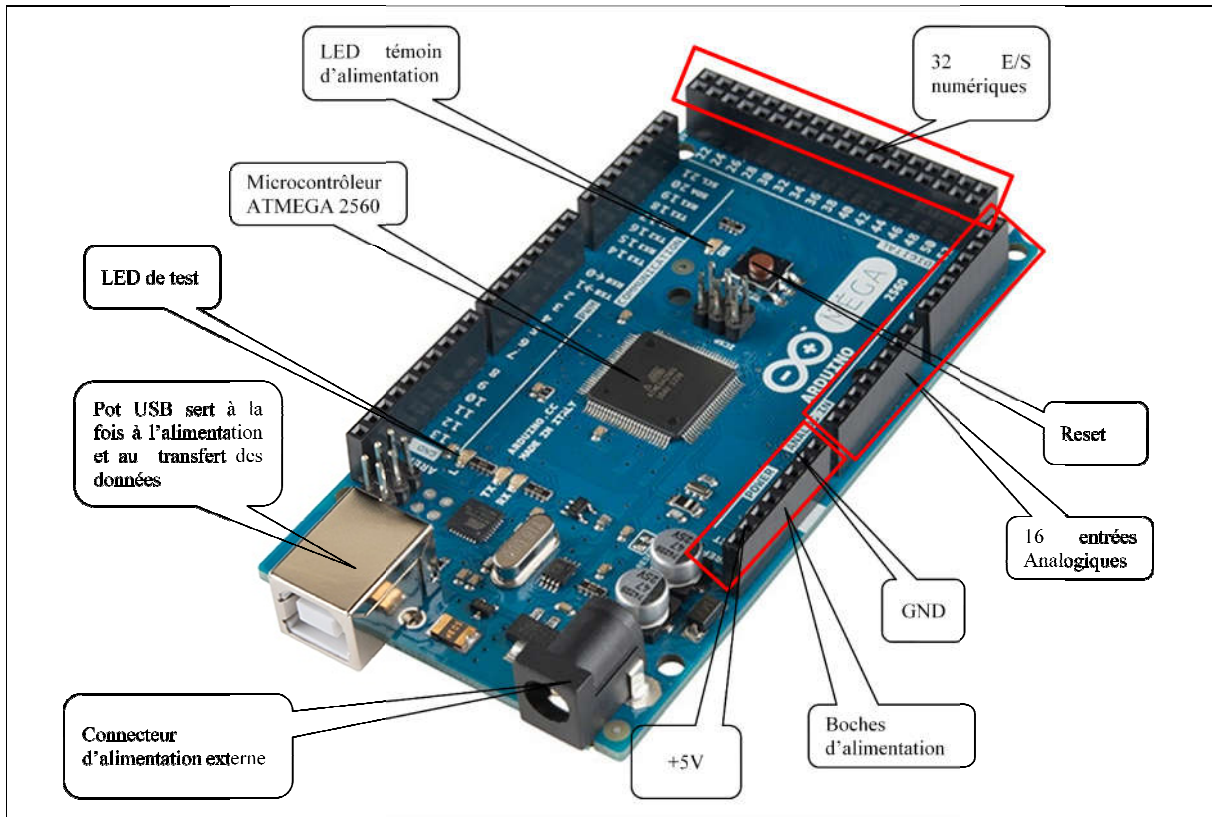


Figure II.2 : Architecture de la carte Arduino Méga

II.6 Constitution de la carte Arduino :

II.6.1 Partie matérielle :

Généralement tout module électronique qui possède une interface de programmation est basé toujours dans sa construction sur un circuit programmable ou plus.

II.6.1.1 Le microcontrôleur ATMEGA2560 :

Un microcontrôleur est un petit processeur informatique relié à des entrées et des sorties numériques (0 ou 1) ou analogiques (tension variable). Il est capable de mémoriser et d'exécuter un programme visant à interpréter les entrées pour agir sur les sorties. Il se programme en général à l'aide d'un ordinateur (grâce aux instructions de ce programme, il peut accomplir des tâches souhaitées, qui peuvent être : faire clignoter une LED, afficher des caractères sur un écran, commander des moteurs) mais il peut fonctionner aussi de manière autonome.

Aujourd'hui, en soudant un grand nombre de composants encombrants ; tels que les transistors, les résistances et les condensateurs tout peut être logé dans un petit boîtier en plastique noir muni d'un certain nombre de broches dont la programmation peut être réalisée en langage C [5]. (Voir la Figure II.3)

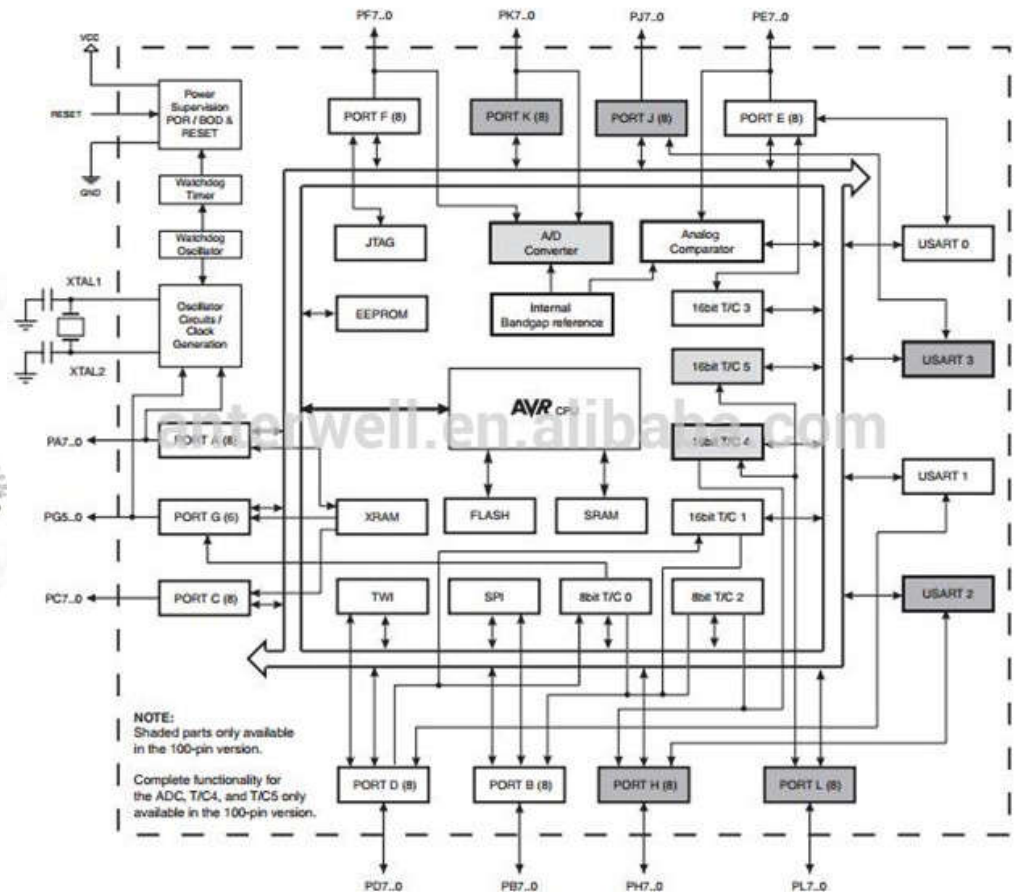


Figure II.3 : Microcontrôleur ATMEGA2560

A. Les caractéristiques du microcontrôleur ATMEGA 2560:

- 135 instructions puissantes, la plupart d'exécution simple de rythme.
- 32x8 registres d'usage universel de fonctionnement.
- Opération entièrement statique.
- Jusqu'à 16 MIPS de sorties à 16 Mhz.
- Mémoires non-volatile de programme et de données.
- 4K Bytes EEPROM.
- Serrure de programmation pour la sécurité de logiciel.
- Quatre canaux à 8 bits de PWM.
- Puissance sur la remise et la détection programmable d'arrêt partiel.

B. La structure interne du microcontrôleur ATMEGA2560 :

Le microcontrôleur ATmega2560 est constitué par un ensemble d'éléments qui ont chacun une façon bien déterminée. Il est en fait constitué des mêmes éléments que sur la carte mère d'un ordinateur. Globalement, l'architecture interne de ce circuit programmable se compose essentiellement sur :

- Une mémoire Flash d'une capacité 256 Ko pour le programme, cette mémoire contiendra le programme à exécuter. Cette mémoire conserve son contenu indéfiniment, même lorsque la carte n'est pas alimentée.
- Une mémoire RAM d'une capacité 8Ko. Cette mémoire est généralement utilisée pour les variables employées dans les programmes, pour stocker des résultats temporaires lors de calculs, etc. elle présente la particularité de pouvoir être lue et écrite à tout instant par le microcontrôleur mais elle oublie son contenu dès qu'il n'est plus alimenté. Il est donc hors de question de l'utiliser pour stocker des informations qui doivent être conservées lorsque la carte ne sera pas alimentée. Pour cela L' ATMEGA 2560 dispose également de mémoire EEPROM.
- Une EEPROM : C'est le disque dur du microcontrôleur. On y enregistre des infos qui ont besoin de survivre dans le temps, même si la carte doit être arrêtée. Cette mémoire ne s'efface pas lorsque l'on éteint le microcontrôleur ou lorsqu'on le reprogramme. Cette mémoire d'une capacité de 4Ko.

II.6.1.2 Les entrées sorties :

La carte Arduino possède 15 entrées analogiques, 54 entrées/sorties numériques dont les pins de (1 à 13) peuvent être utilisées comme des entrées/sorties analogiques. Les entrées/sorties de l'Arduino sont les éléments avec lesquels nous serons le plus souvent en contact. En effet, c'est par leur intermédiaire que le microcontrôleur communique avec le monde extérieur.

II.6.1.3 L'alimentation :

L'alimentation qui assure la distribution d'énergie aux différents composants de la carte Arduino, peut être effectuée via :

- Par la prise d'adaptateur qui fournit une tension comprise entre 7 et 12 v c'est ce qu'on appelle la tension recommandée (la tension est ensuite convertie toute seule en 5v par Arduino).
- Une connexion USB de l'ordinateur qui fournit toujours directement du 5v (la tension de fonctionnement).

Ainsi la carte peut fonctionner sur une alimentation externe de 6 à 20volts c'est ce qu'on appelle la tension limite. Si moins de 7 volts sont fournis, la broche 5v peut éventuellement fournir moins de 5 volts et la carte peut alors être instable. Si plus de 12 volts sont utilisés, le régulateur de tension peut surchauffer et abîmer la carte.

III.6.1.4 Visualisation :

Les trois points blancs entourés en rouge sont en fait des LED dont la taille est de l'ordre du millimètre. Ces LED servent à deux choses :

- Celle tout en haut du cadre : elle est connectée à une broche du microcontrôleur qui peut servir pour tester le matériel. Quand on branche la carte au PC ; elle clignote quelques secondes.
- Les deux LED du bas du cadre : servent à visualiser l'activité sur la voie série (une pour l'émission et l'autre pour la réception). Le téléchargement du programme dans le microcontrôleur se faisant par cette voie, on peut les voir clignoter lors du chargement.

II.6.1.5 La connectique :

La carte Arduino ne possédant pas de composants (résistances, diodes, moteurs...) qui peuvent être utilisés pour un programme, mis à part la LED connectée à la broche 13 du microcontrôleur, il est nécessaire de les rajouter. Mais pour ce faire, il faut les connecter à la carte. C'est là qu'intervient la connectique de la carte.

Sur les Arduino et sur beaucoup de cartes compatibles Arduino, les connecteurs se trouvent au même endroit. Cela permet de fixer des cartes d'extension, appelée shield . Cette connectique est importante et a un brochage qu'il faudra respecter.

Par exemple, la carte Arduino peut être étendue avec des shields, comme le ShieldEthernet qui permet de connecter cette dernière à internet.

II.6.2 Partie programme (SOFTWARE) [6]:

Une telle carte d'acquisition qui se base sur sa construction sur un microcontrôleur doit être dotée d'une interface de programmation comme est le cas de notre carte.

L'environnement de programmation open-source pour Arduino peut être téléchargé gratuitement (pour WINDOWS, Linux).

La carte Arduino présente le noyau de notre système, et pour qu'on puisse la programmer on doit disposer du logiciel compatible avec cette carte, ce logiciel est **Arduino IDE** (Espace de développement intégré).

II.6.2.1 Présentation de l'espace de développement intégré (EDI) Arduino :

Comme n'importe quel langage de programmation, une interface souple et simple est exécutable sur n'importe quel système d'exploitation Arduino basé sur la programmation.

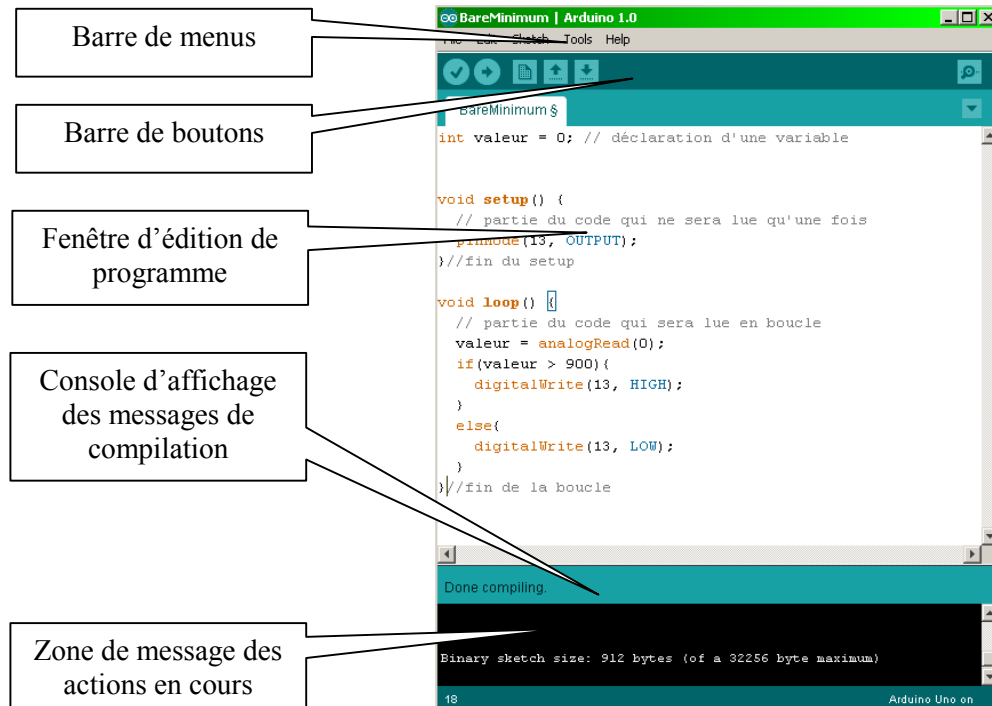


Figure II.4 : Interface IDE Arduino

II.6.2.2 Différents composants de l'espace de développement intégré (EDI):

II.6.2.2.1 La barre de menus :

Le tableau ci-dessous nous montre les icones de la barre de menus :

Fichier(File)	Ce menu contient les différentes options de création, d'ouverture, de sauvegarde, d'impression de programme, ou d'un exemple parmi les exemples qui accompagnent le logiciel Arduino.
Editer (Edite)	Ce menu contient les options de copier/coller, sélection, et les options de recherche.
Programme ou séquence (Sketch)	Ce menu contient les différentes fonctions de la barre des boutons, ainsi que les options d'ajouts de bibliothèques ou de fichiers.
Outils (Tools)	C'est dans ce menu qu'on sélectionne le type de carte à programmer, et le port série utilisé.
Aide (Help)	Ce menu est fait pour donner de l'aide concernant les différents problèmes rencontrés au niveau du logiciel Arduino.

Tableau II.1 : Description des icones de la barre de menus de l'interface IDE Arduino

II.6.2.2.2 La barre de boutons :

Le tableau ci-dessous nous montre les icônes de la barre de boutons :







	Vérifier/compiler	Vérifie le code à la recherche d'erreur.
	Transférer vers la carte	Compile votre code et transférer vers la carte Arduino
	Nouveau (New)	Créer un nouveau programme (ouvre une nouvelle fenêtre).
	Ouvrir (Open)	Ouvrir la liste de tous les programmes dans le "livre de programme".
	Enregistrer (Save)	Enregistrer un programme.
	Moniteur série (Serial Monitor)	Ouvrir la fenêtre du moniteur (ou terminal) série.

Tableau II.2 : Description des icônes de la barre de boutons de l'interface IDE Arduino

II.6.2.2.3 Structure d'un programme Arduino :

-**Un éditeur** : c'est l'espace utilisé pour écrire le programme à réaliser, comme il dispose aussi des onglets de navigation.

-**Une zone de message** : comme son nom l'indique cette espace est utilisée par l'IDE pour communiqué à l'utilisateur des informations sur l'état du programme et des actions en cours.

-**Une console texte** : qui affiche les messages concernant le résultat de la compilation du programme (il indique si le programme comporte des erreurs).

II.6.2.2.4 Le programme Arduino :

A. Description de la structure d'un programme Arduino :

Le langage Arduino est basé sur les langages « C/C++ ». Un programme utilisateur Arduino est une suite d'instructions élémentaires sous forme textuelle, ligne par ligne. La carte lit puis effectue les instructions les unes après les autres, dans l'ordre défini par les lignes de code, comme lors d'une programmation classique. Cette structure se décompose en trois parties :

***Définition des constantes et variables globale** : La déclaration des variables se fait généralement dans l'espace global (de façon à partager les variables les plus importantes entre les deux fonctions principales).

***Fonction principale : void setup() :**

- Initialisation des ressources de la carte,
- Configuration des entrées/sorties,
- Définition de la vitesse de fonctionnement.

***Fonction boucle : void loop() :** cette partie s'exécute en boucle.

- Description du fonctionnement général du programme
- Gestion des interactions entre les entrées/sorties.

B. Le jeu d'instructions du langage Arduino :**➤ Les syntaxes du programme :**

- Chaque instruction se termine par un « ; »,
- Les accolades « { » et « } » sont les conteneurs du code du programme. Elles sont propres aux fonctions, aux conditions et aux boucles. Les instructions du programme sont écrites à l'intérieur de ces accolades,

-Les commentaires sont des lignes de texte incluses dans le programme et qui ont pour but de vous informer vous-même ou les autres de la façon dont le programme fonctionne. Ces lignes ajoutées sont ignorées par le compilateur. Les commentaires sont précédés des caractères « // » ou bien encadrés par « /* » et « */ »,

-Il est formellement interdit de mettre des accents en programmation, sauf dans les commentaires,

-Un nombre en binaire doit être précédé de la lettre « B »,

-Un nombre écrit en hexadécimal doit être précédé par les caractères « 0x ».

➤ Les commandes du programme Arduino :

le tableau ci-dessous nous montre les commandes du programme Arduino :

Structure générale	<ul style="list-style-type: none"> • Void setup() (configuration-préparation) • Void loop() (Exécution)
Contrôle des conditions	<ul style="list-style-type: none"> • if (si) • if...else (si...alors...) • for (pour...) • switch case (dans le cas où...) • while (pendant que...)
Opération de comparaison	<ul style="list-style-type: none"> • =(Equivalent) • != (Différent) • < (Inférieur à) • (Supérieur à)

	<ul style="list-style-type: none"> • <= (Inférieur ou égal à) • >= (Supérieur ou égal à)
Autres commandes	<ul style="list-style-type: none"> • // (Commentaire simple), • /**/ (Commentaire multi ligne), • # défine (donner une valeur à un non)
Opération booléennes	<ul style="list-style-type: none"> • && (Et) • (Ou) • ! (Et pas)
Variables	<ul style="list-style-type: none"> • char (Variable 'caractère') • int (Variable 'nombre entier') • long (Variable 'nombre entier de très grande taille') • string (Variable 'chaîne de caractères') • array (Tableau de variables)
Niveau logique des connecteurs numériques	<ul style="list-style-type: none"> • HIGH (état 1) • LOW (état 0) • INPUT (configuré en entrée) • OUTPUT (configuré en sortie)
Fonctions *Entrées-sorties numériques	<ul style="list-style-type: none"> • Entrées-sorties numériques : • pinMODE(broche, état) (configuration des broches en entrée ou en sortie) • digitalWrite(broche, état) (Ecrire un état sur une broche numérique) • digitalRead(broche) (lire un état sur une broche numérique) • unsigned long pulseIn(broche, état)(lire une impulsion sur une broche numérique)
Fonctions *Entrées analogiques	<ul style="list-style-type: none"> • int analogRead (broche) (lire la valeur d'une broche analogique) • analogWrite (broche, valeur) (PWM : écrire une valeur analogique sur les broches 5, 6, 9,10 et 11)
Fonctions *Gestion du temps	<ul style="list-style-type: none"> • unsigned long millis() (temps de fonctionnement du programme) • delay(ms) (attente, en millisecondes). • delayMicroseconds(us) (attente, en microsecondes)

Tableau II.3 : Description des commandes de programme Arduino

C. Les étapes à suivre pour programmer la carte Arduino :

➤ L'écriture du programme :

Cette première étape consiste à saisir le programme.

La figure II.5 présente la fenêtre du programme.

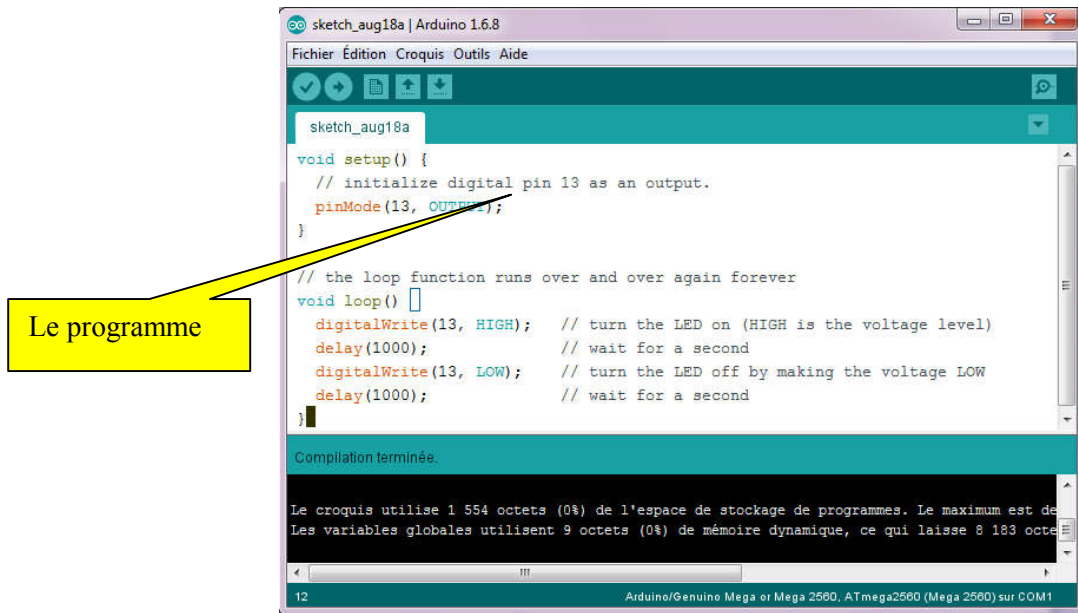


Figure II.5 : L'écriture du programme Arduino

➤ **Compilation du programme :**

Dans cette partie, on vérifie si le code contient des erreurs de syntaxes. En cas d'anomalie dans le programme, le compilateur nous enseigne sur le type d'erreur et la ligne où elle se trouve. (Pour la correction les erreurs il suffit de consulter l'aide disponible dans la barre de menus).

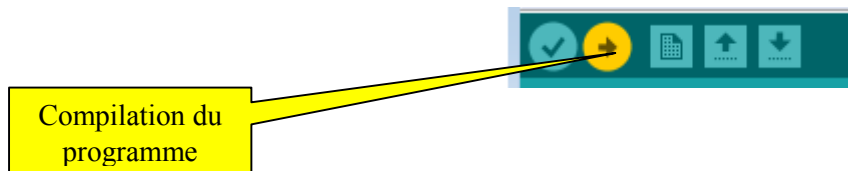


Figure II.6 : Compilation du programme

Si aucune erreur n'est détecté, le programme est compilé et le message « compilation terminée » apparaît, suivi de la taille du programme.

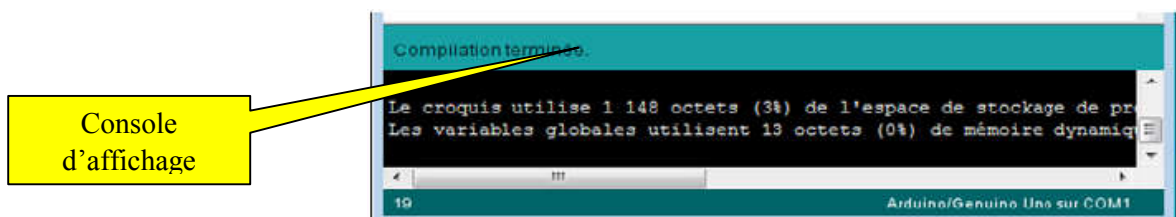


Figure II.7 : Console d'affichage

➤ **Sélection de la cible (type de carte) :**

Avant de transférer le programme vers la carte Arduino, il faut sélectionner le type de carte Arduino (cible) depuis le menu **Outils>Board**.

La carte doit évidemment être connectée à l'ordinateur via un câble USB.

Vérifions que c'est bien le nom " Arduino Méga 2560" qui est coché. Si ce n'est pas le cas, on le coche.

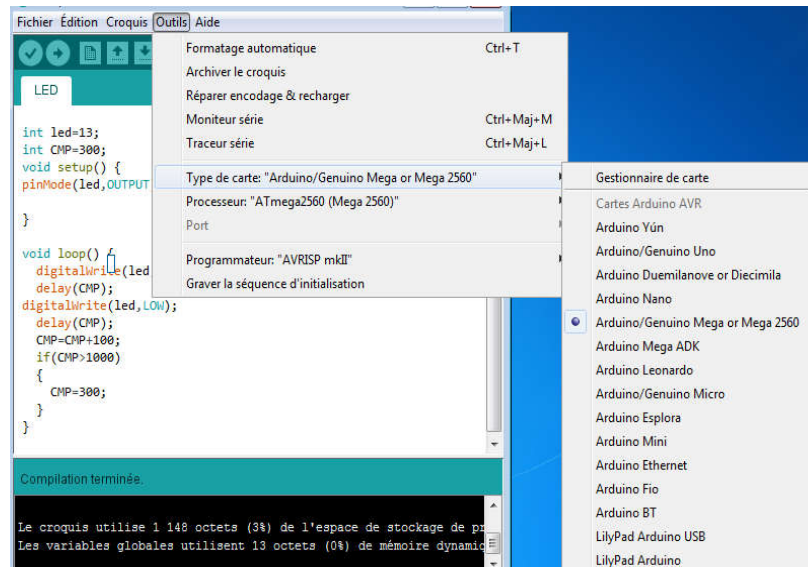


Figure II.8 : Sélection de la cible

➤ **Sélection du port :**

Sélectionner le port série sur lequel est branché cette (depuis le menu **Outils>Port**).

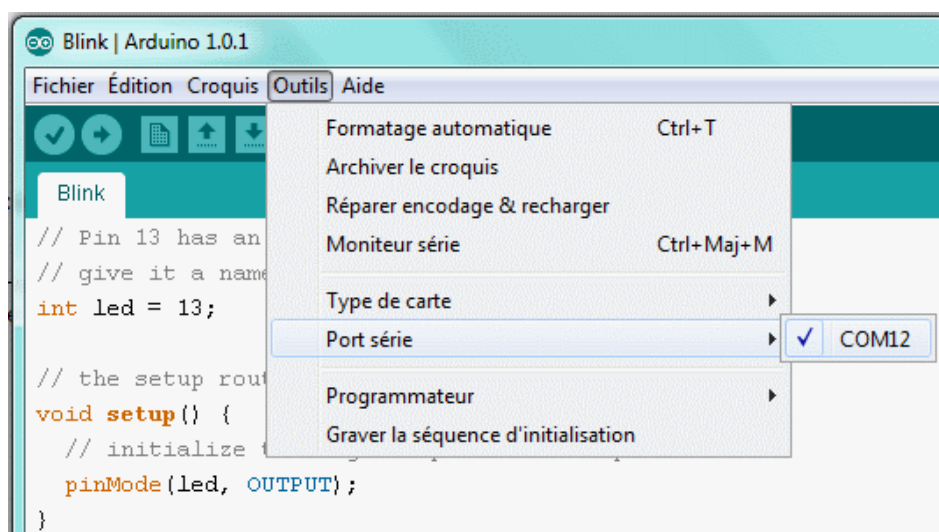


Figure II.9 : Sélection du port

➤ **Transfert du programme :**

Cela se fait en cliquant sur le bouton.

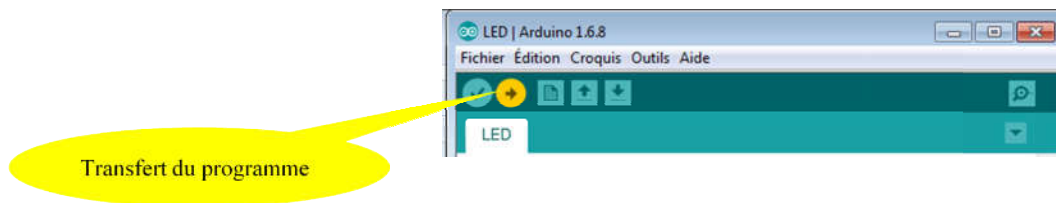


Figure II.10 : Transfert du programme

Le programme est téléchargé vers la carte, le message « télé versement terminé » s'affiche une fois l'opération terminée.

Sur la plupart des cartes, nous devons voir les LEDs des lignes RX et TX clignoter rapidement, témoignant que le programme est bien transféré. Durant le transfert le bouton devient jaune et le logiciel Arduino affiche un message indiquant que le transfert est en cours.

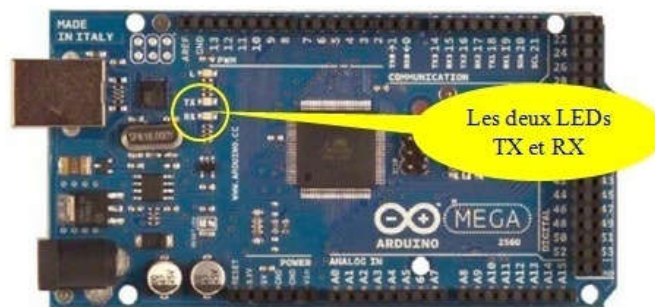


Figure II.11: Clignotement des deux LED du port série

II.7 Conclusion :

On peut conclure sur le fait que les cartes Arduino sont un puissant outil de prototypage pour les cartes électroniques. Mais aussi, elles permettent un accès facile et intuitif à l'informatique embarquée. Ces cartes sont conçues pour réaliser des prototypes et des maquettes de cartes électroniques pour l'informatique embarquée et elles lui permettent un accès simple et peu coûteux.

Le langage Arduino se distingue des langages utilisés dans l'industrie de l'informatique embarquée de par sa simplicité. En effet, beaucoup de bibliothèques de fonctionnalités de base occulte certains aspects de la programmation de logiciel embarquée

afin de gagner en simplicité. Cela en fait un langage parfait pour réaliser des prototypes ou des petites applications dans le cadre de hobby.

Les possibilités des cartes Arduino sont énormes, un grand nombre d'applications ont déjà réalisées et testées par bon nombre d'internautes. On retrouve par exemple diverse forme de robot, des stations météo. D'autres exemples comme notre projet qui est la commande numérique d'un massicot avec une carte Arduino.

Dans ce chapitre, notre étude s'est localisée sur la présentation de la carte d'acquisition qui est la carte Arduino et le prochain chapitre sera consacré à la conception de notre système.



Chapitre III

Présentation du matériel

III.1 Introduction :

Dans ce chapitre, nous présenterons le matériel utilisé, les différents composants électroniques et leurs caractéristiques techniques, ainsi les différents techniques à appliquer afin d'avoir une idée précise sur la maquette à réaliser.

Ce chapitre va faire apparaître des notions de physique, moment cinétique et des équations d'électromagnétique, électrique, qui sont indispensables pour comprendre le mouvement.

Enfin, nous devons avoir une idée précise sur le principe de fonctionnement de l'encodeur incrémental, les caractéristiques de moteur utilisé et leur fonctionnement, asservissement, etc.

III.2 Première partie : Présentation du matériel :

Afin de réaliser notre projet qui est une maquette de massicot, on est en besoin d'une carte Arduino Méga 2560 (carte d'acquisition), une carte de puissance L293D pour l'alimentation et la commande du moteur, moteur réducteur avec encodeur, un clavier, deux capteurs de fin de course et un afficheur LCD.

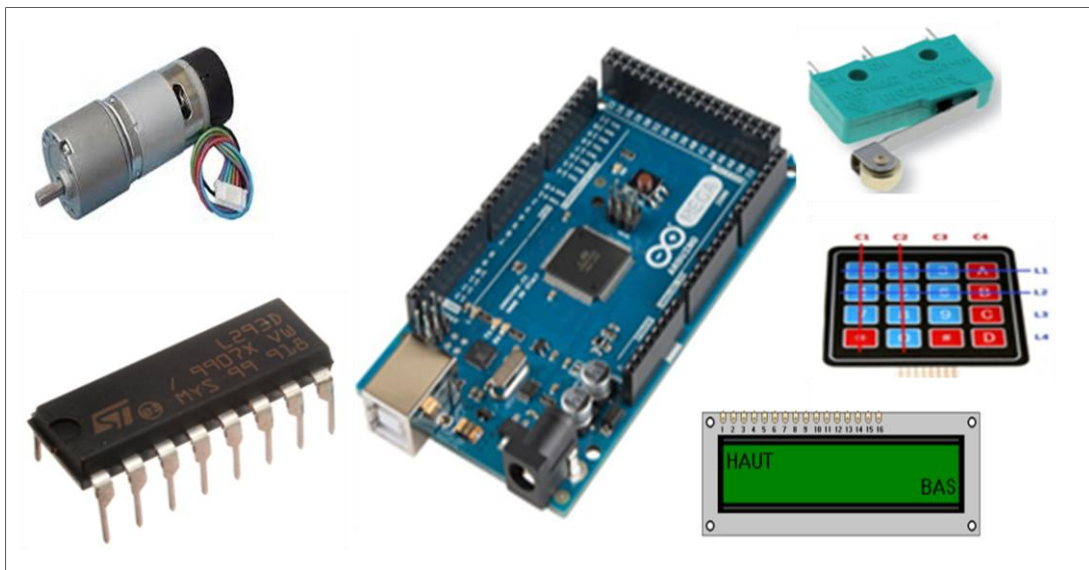


Figure III.1 : Diagramme de connexion du dispositif de contrôle et de commande

III.2.1 Carte d'acquisition :

S'occupe de l'acquisition et de traitement des signaux. Elle s'articule autour d'un microcontrôleur (AtMega2560 à 16MHz) qui permet une grande souplesse dans la

programmation de l'asservissement. Ce microcontrôleur acquiert les signaux venant de l'encodeur optique et calcule les corrections à appliquer en fonction de la commande initiale.

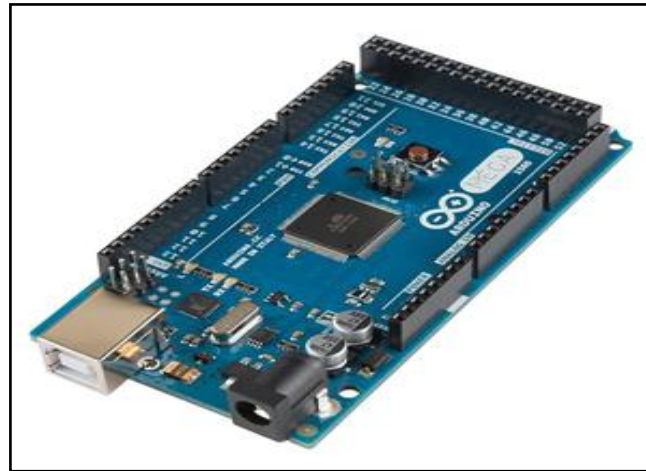


Figure III.2 : La carte Arduino Méga2560

III.2.2 Afficheur LCD :

Les afficheurs LCD (Liquid Crystal Display) sont devenues incontournables dans toutes applications qui demandent la visualisation de paramètres, il s'agit donc d'une interface Homme/Machine. Ils sont très utilisés dans les montages à microcontrôleur, et permettent une grande convivialité. Ils peuvent aussi être utilisés lors de la phase de développement d'un programme, car on peut facilement y afficher les valeurs de différentes variables. Au paravent onéreux et difficile à mettre en œuvre, ils sont maintenant bon marchés et l'interface parallèle au standard Hitachi permet un pilotage facile [7].

L'afficheur est présenté dans la figure III.3 ci-dessous :

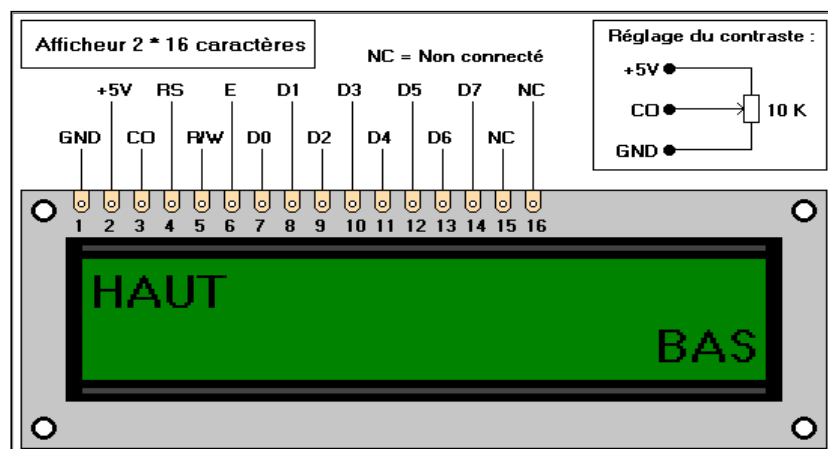


Figure III.3 : Afficheur LCD (2x16)

III.2.2.1 Description des pins :

Le tableau III.1 ci-dessous représente la description des pins de notre afficheur LCD :

Broche	Non	Description
1	Vss	Masse
2	Vdd	Alimentation 5v
3	CO	Variables de 0 à 5v permet de modifier le contraste de l'afficheur
4	RS	Indique une commande ou une donnée à afficher (0 : Commande/ 1 : Donnée)
5	R/W	Indique une écriture ou une lecture (0 : Ecriture/ 1 : Lecture)
6	E	Indique une validation (le niveau haut doit être maintenu 500µ)
7	D0	Bus de données bidirectionnel
8	D1	
9	D2	
10	D3	
11	D4	
12	D5	
13	D6	
14	D7	
15	A	Anode rétroéclairage (+5V)
16	K	Cathode rétroéclairage (masse)

Tableau III.1 : Description des pins de l'afficheur LCD

III.2.2.2 Fonctionnement et commande de l'afficheur LCD :**1. Fonctionnement de l'afficheur LCD :**

On envoie deux types d'information à l'afficheur :

- Les commandes qui permettent de l'initialiser : positionnement du curseur, effacement écran, etc.
- Les données à afficher.

L'entrée RS permet de spécifier si on envoie une commande ou une donnée :

- RS=0 : instruction (commande),
- RS=1 : caractère (donnée).

L'afficheur dispose d'une entrée R/W pour spécifier une lecture ou une écriture :

- R/W=0 : écriture vers l'afficheur,
- R/W=1 : lecture de l'afficheur.

Pour valider tous les échanges sur le bus de données (D7-D0) on utilise l'entrée E de l'afficheur. Un front descendant sur cette entrée valide la donnée. En programmation, il faudra placer un court instant E à l'état haut puis à l'état bas [7].

2. Commande d'un afficheur LCD :

Il existe deux modes de commande d'un afficheur LCD :

A/ Commande en 4 bits :

En mode de commande 4 bits l'intérêt est de limiter le nombre de broches du microcontrôleur.

Vu du côté du microcontrôleur, cela donne :

- 3 sorties (reliées aux 3 entrées RS, R/W et E du module LCD),
- 4 broches (reliées aux 4 bits du bus de données D7-D4),
- Configurées en sortie en mode écriture (R/W=0),
- Configurées en entrée en mode lecture (R/W=1) [8].

La figure de commande d'un afficheur LCD en mode 4 bits est présentée par la figure III.4 :

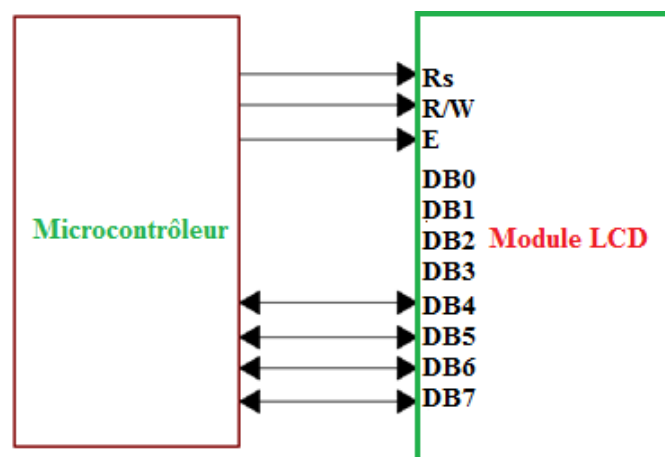


Figure III.4 : Commande d'un afficheur LCD en mode 4 bits

B/ Commande en 8 bits :

Dans ce cas, la commande du l'afficheur LCD nécessite 11 broches.

Vu du côté du microcontrôleur, cela fait :

- 3sorties (reliées aux 3 entrées RS, R/W et E du module LCD)
- 8 broches (reliées au bus de données D7-D0)
- Configuration en sortie en mode écriture (R/W=0)
- Configuration en entrée en mode lecture (R/W=1) [8].

La figure III.5 présente la commande d'un afficheur LCD en mode 8 bits

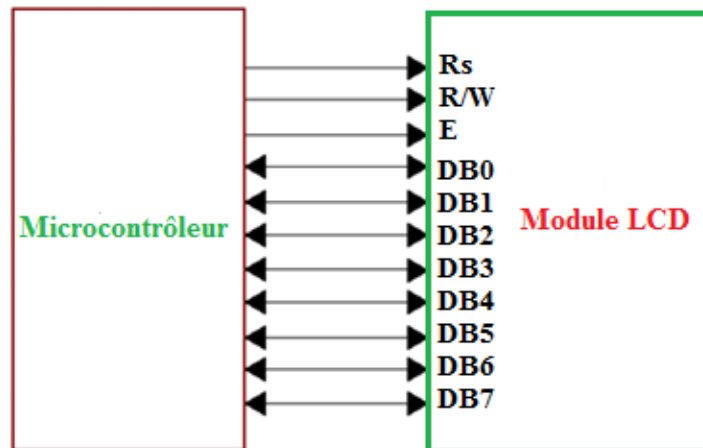


Figure III.5 : Commande d'un afficheur LCD en mode 8 bits

III.2.2.3 Branchement de l'afficheur LCD à la carte Arduino :

L'afficheur est lié à notre carte Arduino comme suit :

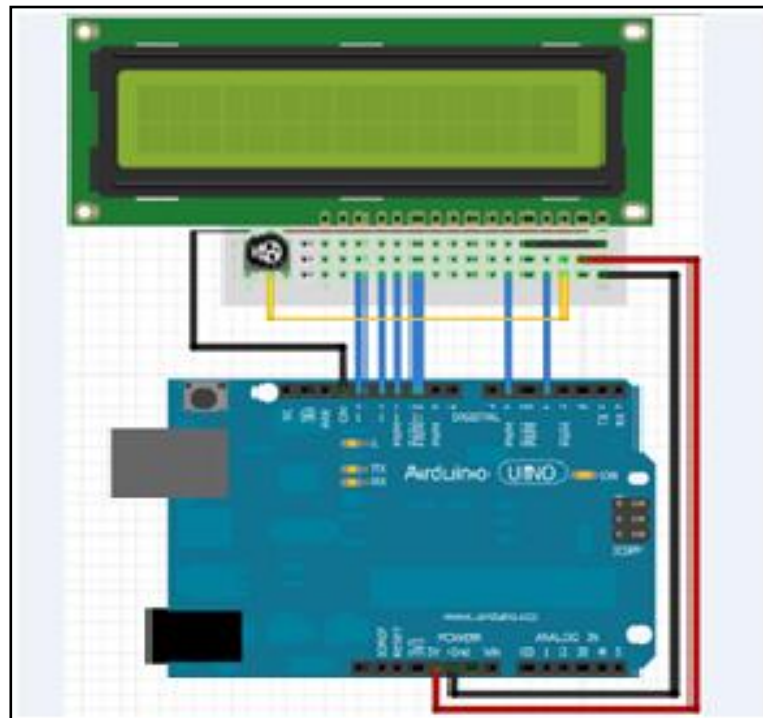


Figure III.6 : Schéma de branchement de l'afficheur LCD à l'Arduino

III.2.3 Le clavier :

Un clavier est une interface Homme/Machine, il compose de touches envoyant des instructions à la machine une fois actionnées.

Une touche est un interrupteur électronique similaire aux boutons d'une télécommande ou d'une manette de console de jeu. Un symbole, une lettre, un chiffre, un mot ou une image sont habituellement imprimées ou gravées sur la touche pour permettre à l'utilisateur de saisir les caractères pour écrire du texte ou pour exécuter une fonction particulière.

➤ **Description de clavier :**

Le clavier comporte 16 touches, dont 10 numériques (0-9) et 6 touches marquées " #", "*", "A", "B", "C", "D".

Il est matriciel : au lieu d'avoir 16 fils (1 par touche) et une masse, le multiplexage n'utilise que 8 sorties : 4 lignes et 4 colonnes.

Les lignes sont des sorties, les colonnes sont des entrées maintenues au niveau haut par une résistance interne à Arduino.

Le système envoie par balayage un niveau bas sur chaque ligne (1 seule à la fois) et balaye les colonnes en lecture. Quand il lit un niveau bas, c'est que la colonne est reliée par une touche appuyée à la ligne qui est basse à ce moment. On n'a pas besoin de programmer ce balayage : c'est la bibliothèque qui le fait.

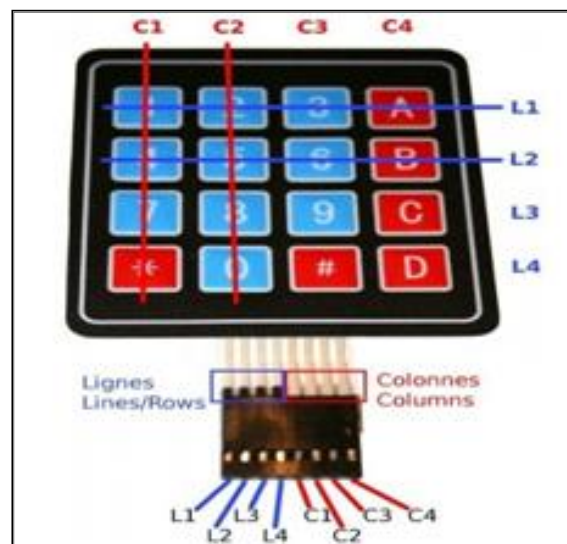


Figure III.7 : Le clavier matriciel (4x4)

III.2.4 Le moteur utilisé :

Le moteur utilisé dans notre réalisation est un moteur réducteur qu'on peut alimenter avec 12v. Ce moteur présenté à la figure III. 7, possède un réducteur (rapport 30 :1) et un encodeur intégré, dont la consommation de 530 mA environ. (150mA à vide). Son courant de blocage est de 2,5A, et sa vitesse de rotation est de 170 tr/min (216 tr/min à vide), son couple est de 1,5kg.cm. Et le nombre d'impulsion de son encodeur est 360 impulsions par tour. Il comporte 6 fils de branchement, à savoir le fils :

Il comporte 6 fils de branchement, à savoir le fils :

- Violet (1) : capteur à effet Hall B V out
- Bleu (2) : capteur à effet Hall A V out
- Vert (3) : capteur à effet Hall "terre"
- Brun(4) : capteur à effet Hall Vcc
- Rouge (5) : + moteur
- Noir (6) : - moteur



Figure III.8 : Moteur réducteur avec encodeur

III.2.5 Présentation du pont H :

Le pont en H est une structure électronique servant à contrôler la polarité aux bornes d'un dipôle, il est composé de quatre éléments de communication généralement disposés schématiquement en une forme de H d'où le nom. Cette structure se trouve dans plusieurs applications de l'électronique de puissance incluant le contrôle des moteurs, les convertisseurs et hacheurs, la figure III.9 ci- dessous représente la structure de pont H.

- Une partie puissance composée de 4 interrupteurs commandés (relais, transistor bipolaire ou MOS, thyristor, etc.)

▪ Une partie de commande qui permet de contrôler l'ouverture ou la fermeture des interrupteurs. Cette commande se fait le plus souvent par l'intermédiaire d'un microcontrôleur afin de contrôler le pont en TOR (Tout ou Rien) ou en PWM [9].

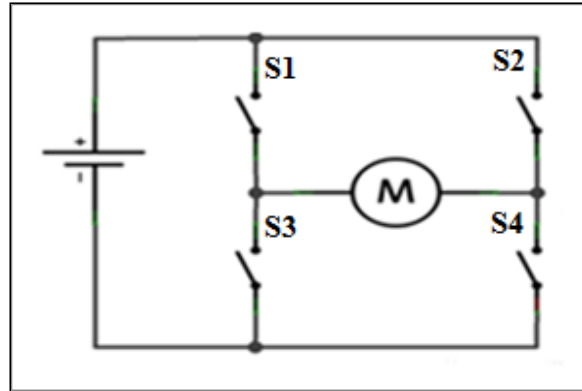


Figure III.9 : Structure du pont H

➤ **Technique du pont H de base :**

Le pont H est un montage électrique simple qui permet de changer le sens du courant dans un moteur ou d'inverser la tension, c'est ce qu'on envoie dans les deux schémas ci-dessous :

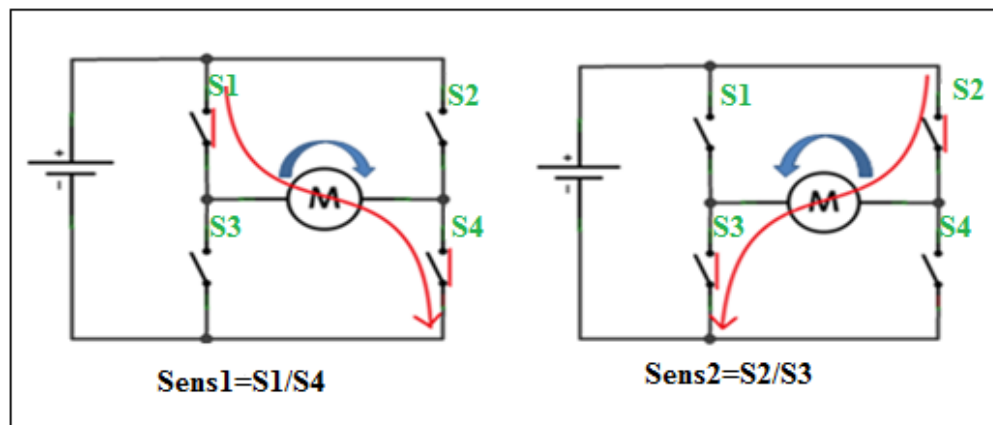


Figure III.10 : Structure du pont H électrique
(pont H de base)

M : moteur

S1, S2, S3, S4 : Interrupteurs peuvent être modalisés dans la réalité par des relais électromécaniques (R_{L1} , R_{L2} , R_{L3} , R_{L4}).

On remarque bien que ces relais fonctionnent par paire et le tableau III.2 montre ce fonctionnement :

R_{L1}	R_{L2}	R_{L3}	R_{L4}	Résultats
X	OK	OK	X	Le moteur tourne dans le sens inverse des aiguilles d'une montre
OK	X	X	OK	Le moteur tourne dans le sens des aiguilles d'une montre
OK	X	OK	X	Court circuit
X	OK	X	OK	Court circuit

Tableau III.2 : Le fonctionnement des relais du pont H

Si on ouvre les relais R_{L1} , R_{L4} et on ferme les relais R_{L2} , R_{L3} on revanche le courant se dirige dans le sens1, et pour les dipôles récepteurs la tension va dans le sens inverse du courant.

Si on ferme les relais R_{L1} , R_{L4} et on ouvre les relais R_{L2} , R_{L3} le moteur tourne dans le sens2.

Il ne faut jamais effectuer une autre combinaison du type R_{L3} et R_{L1} car on relie directement l'alimentation à la masse ce qui fait gréer l'alimentation et c'est ce qui fait carrément un court circuit.

III.2.6 Présentation du boîtier L293D :

Le L293D est un double pont-H, ce qui signifie qu'il est possible de l'utiliser pour commander quatre moteurs distincts (dans un seul sens) grâce à ses 4 canaux.

Le L293D un circuit intégré monolithique, à haut voltage, grand courant et 4 canaux. Cela veut dire que ce circuit intégré peut être utilisé pour des moteurs DC alimentés jusqu'à 36 Volts. Le circuit peut fournir un maximum de 600 mA par canal. Le L293D est aussi connu pour être un excellent pont-H facile à mettre en pratique.

Avec deux signaux de commande Input2 fournis par Arduino, il est possible d'inverser la direction du courant dans le pont-H et donc renverser le sens de rotation du moteur qui y est raccordé. En utilisant différentes combinaisons d'Input1 et Input2 il devient possible de démarrer, stopper ou inverser le courant [9].

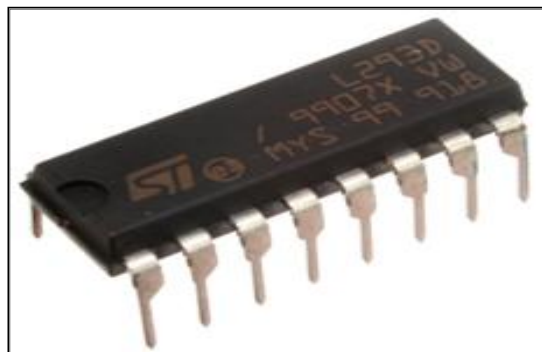


Figure III.11 : La carte de puissance L293D

III.2.6.1 Le brochage :

La figure III.12 présente le brochage de la carte de puissance L293D.

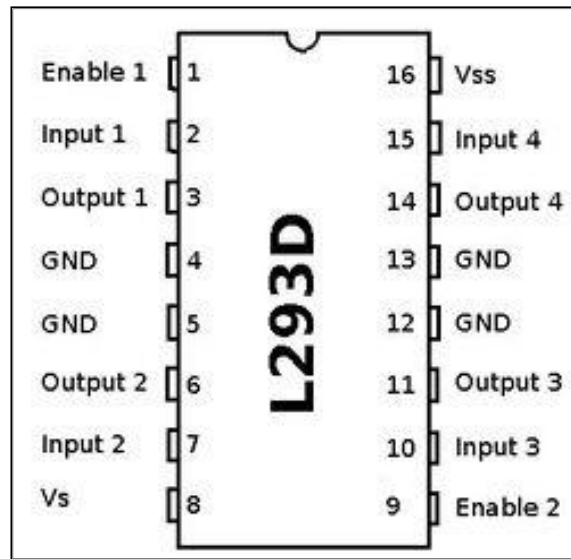


Figure III.12 : Les broches du boîtier L293D

III.2.6.2 Description des broches du L293D :

Le tableau III.3 ci-dessous représente la description des branches de L293D:

Broche	Nom	Description
1	Enable1	Permet d'envoyer (ou pas) la tension sur les sorties du moteur via Output1 et Output2. Enable1 commande l'activation/désactivation du premier pont-H. <ul style="list-style-type: none"> Si Enable1=GND, le pont-H est déconnecté et le moteur ne fonctionne pas. Si Enable1= Vss, le pont-H est connecté aux sorties et le moteur fonctionne dans un sens ou l'autre ou pas en fonction des tensions appliquée sur Input1 et Input2.
2	Input 1	Broches de commande du pont -H. Se raccorde à la carte Arduino.
7	Input 2	
3	Output1	Output1 et Output2 sont les broches à raccorder à la charge (le moteur)
6	Output2	
4	GND	Doit être raccorder à la masse (GND) de la source d'alimentation de puissance Vs (ex : la borne négative de l'accumulateur +9.2v) et à la masse de la source d'alimentation de la logique « Vss » (donc GND Arduino).
5		
12		
13		

8	Vs	Alimentation de puissance des moteurs.
9	Enable2	Commande l'activation du second pont-H constitué de Output3/Output4
10	Input 3	A utiliser conjointement avec Input4 pour commander le pont-H Output3/Output4.
11	Output3	Constitue une des deux sorties du second pont-H (Output3/Output4)
14	Output4	
15	Input 4	A utiliser conjointement avec Input3 pour commander le pont-H Output3/Output4.
16	Vss	Alimentation de la logique de commande (5V). A raccorder à la borne +5V d'Arduino (donc sur le régulateur d'Arduino).

Tableau III.3 : Description des broches de L293D

III.2.6.3 Les caractéristiques du L293D :

Le tableau III.4 présente les caractéristiques de la carte de puissance L293D :

Caractéristique	Valeur
Nombre de pont-H	2
Courant Max Régime continu	600 mA (x2)
Courant de point Max<2ms	1200mA
Vs Max Alimentation moteur	36V
Vss Max Alimentation logique	7V
Nombre de broches	16 DIP
Perte de tension	1.3 à 1.4V

Tableau III.4 : Le rôle des broches de L293D

III.2.6.4 Logique de commande Input :

Le tableau III.5 présente la logique de commande Input de la carte de puissance L293D :

Enable 1	Input 1	Input 2	Fonction
High	Low	High	Tourne dans le sens horlogique
High	High	Low	Tourne dans le sens anti-horlogique
High	Low	Low	Stop
High	High	High	Stop
Low	Non applicable	Non applicable	Stop

**Tableau III.5 : La logique de commande
Inputs de L293D**

III.2.7 Les capteurs :

Les capteurs permettent de transformer une grandeur physique en une grandeur généralement électrique, comme dans notre cas : la position.

III.2.7.1 Les capteurs de fin de courses :

Le capteur de fin de course, encore appelé soit interrupteur de fin de course, interrupteur de position, ou détecteur de position. C'est un commutateur commandé par le déplacement d'un organe de commande (corps d'épreuve). Lorsqu'il est actionné, il ouvre ou ferme un ou plusieurs circuits électriques ou pneumatiques.

Dans notre cas, le capteur de position utilisé est de type TOR, qui peut prendre deux états : en repos et actionné, à chaque état correspond un signal de sortie (1 ou 0).

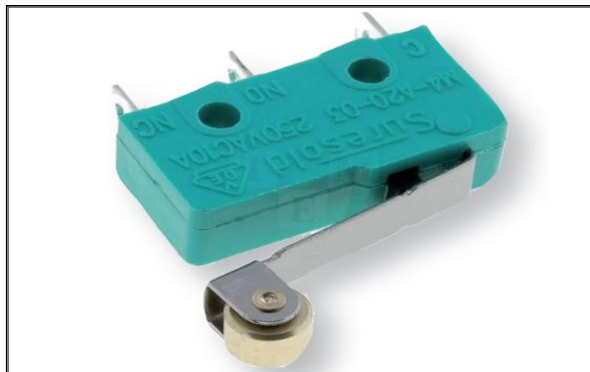


Figure III.13 : Capteur de fin course

III.2.7.2 L'encodeur incrémental :

Les codeurs incrémentaux sont destinés à des applications où l'information de position est obtenue par mesure du déplacement de l'objet. L'encodeur délivre un train d'impulsions dont le nombre permet de déduire la valeur du déplacement ainsi que la vitesse car cette dernière est proportionnelle à la fréquence des impulsions. Ils sont constitués d'un disque comportant deux à trois pistes : A et Z.

- ✓ Une ou deux Pistes extérieures (A et B), divisées en ' $2n$ ' intervalles d'angles égaux, alternativement opaques et transparents, n est la '**résolution**' ou le '**nombre de points/tour**' du codeur (c'est aussi le nombre de périodes/tour ou le nombre d'impulsions/tour).
- ✓ Piste intérieure (Z : top zéro), qui ne comporte qu'une seule fenêtre transparente et qui délivre qu'un signal par tour du disque donc qu'une impulsion /tour, ce signal 'Z' d'une durée de 90° électrique. Ce « top zéro » permet de réinitialiser la partie commande et de connaître une position d'origine et le comptage des tours [10].

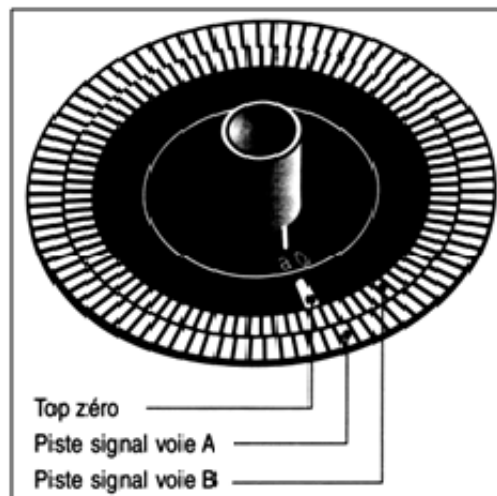


Figure III.14 : Disque d'un codeur incrémental

➤ Principe de fonctionnement :

Un encodeur incrémental possède trois têtes de lecture :

Une tête de lecture est affectée à la piste intérieure (Z : top zéro) et délivre une impulsion par tour, permettant à la commande de compter le nombre d'impulsions reçues.

Deux têtes de lecture sont placées sur la piste extérieure (A et B). Chaque tête, prise isolément, permet à la partie commande de déterminer l'angle de rotation du disque en comptant le nombre d'impulsions reçues.

Les deux têtes sont décalées l'une par rapport à l'autre d'un quart de largeur de fente. Ainsi, les signaux carrés (A et B) émis sont décalés dans le temps de 90° électrique. La partie commande, en détectant quelle voie change d'état en premier peut déterminer le sens de rotation du disque. Ce décalage permet de déterminer le sens de rotation [10].

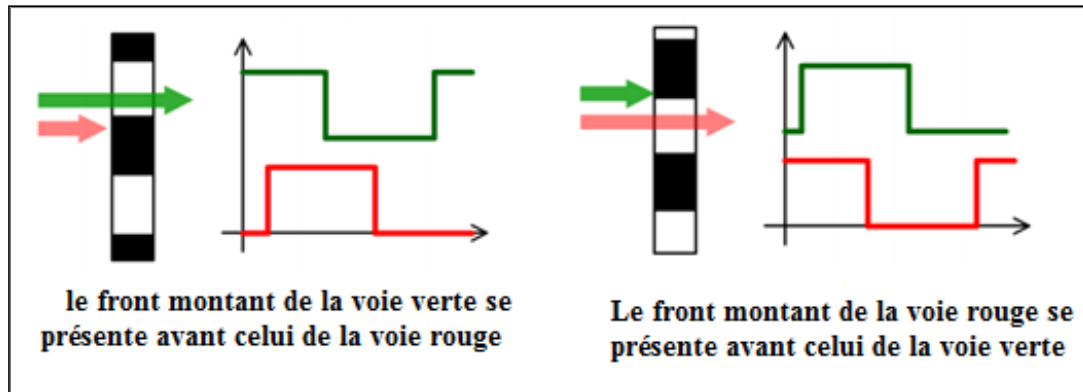


Figure III.15 : Particularités de fonctionnement d'un encodeur incrémental

III.4 Deuxième partie : Description du moteur à courant continu (MCC):

III.4.1 Fonctionnement et constitution du MCC :

Si un conducteur en forme de spire, parcouru par un courant continu « I », est placé dans un champ magnétique, il est soumis à des forces de Laplace. Ces forces créent un couple de rotation qui fait tourner la spire sur son axe. Quand la spire a fait un demi-tour, il faut inverser la polarité pour inverser le sens des forces et continuer le mouvement. Ce sera le rôle du collecteur.

La figure III.16 ci-dessous montre le principe physique de la machine à courant continu.

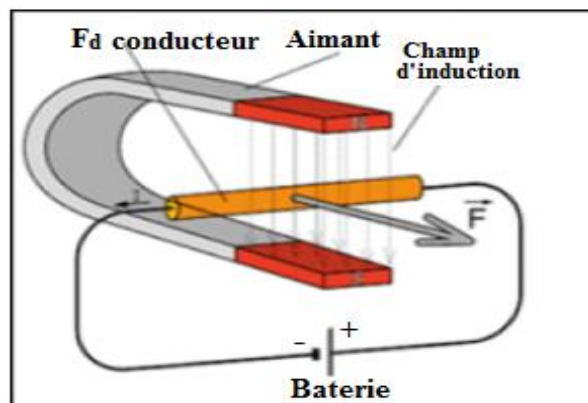


Figure III.16: Le principe physique de la machine à courant continu

III.4.2.2 Asservissement du MCC :

A partir des équations électriques et mécanique ci-dessus (relation de proportionnalité entre force contre électromotrice et vitesse de rotation ainsi la même relation avec le même coefficient entre le courant et le couple), on pourra établir le schéma bloc d'un MCC ayant en entrée $u(s)$ et en sortie $\Omega(s)$ on pourra voir clairement l'asservissement (un asservissement qui est réalisé ou qui est due à la force contre électromotrice E , qui s'oppose à la tension qu'on met aux bornes de l'induit et qui est proportionnelle à la vitesse de rotation) qu'il y a dans la machine :

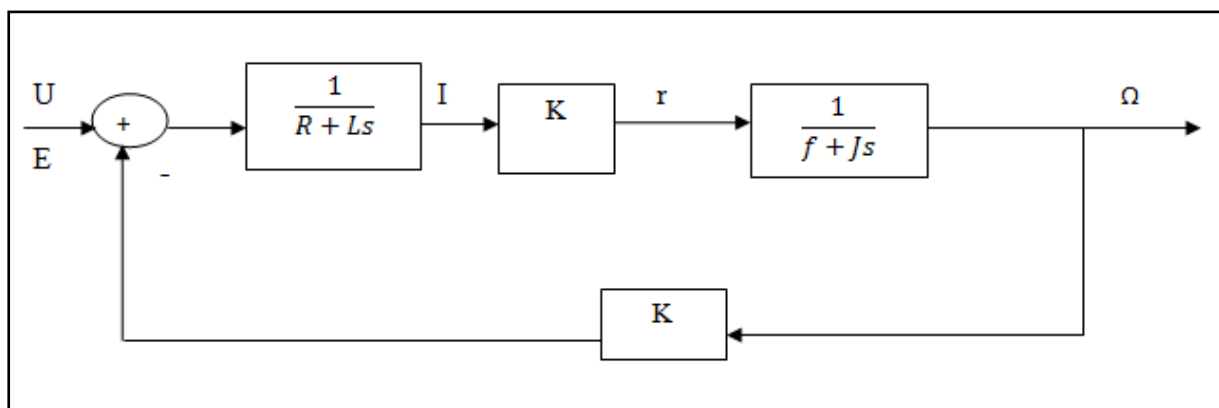


Figure III.17 : Schéma bloc du MCC

On obtient :
$$\frac{\Omega(s)}{U(s)} = \frac{K}{LJs^2 + (RJ + Lf)s + Rf + K^2} \dots\dots\dots(III.5)$$

La fonction de transfert d'un moteur de robotique et, en général, de tout moteur utilisé dans un servomécanisme est une fonction de transfert du second ordre.

➤ **Asservissement couple/courant :**

Ce qui est sympa pour un moteur à courant continu c'est que on n'a pas besoin de faire un asservissement de couple, le couple est contrôlable très facilement : on a une relation linéaire entre le courant et le couple, naturellement y a déjà linéarisation du comportement mécanique de la machine, donc on veut se rendre maître du courant, le maîtriser parfaitement en temps réelle avec une rapidité qui correspond à la dynamique la plus rapide du système. Pour une raison technologique pourquoi se rendre maître du courant, le maîtriser à tout moment, tout simplement pour qu'il ne dépasse pas la valeur maximale et cette valeur est imposée par les composants de puissance. C'est des composants qui sont relativement sensible à tout sur

courant même très sensible, il suffit de dépasser une valeur nominale que ca entraîne leur destruction.

Le correcteur C(s) de l'asservissement de courant et en général du type PI :

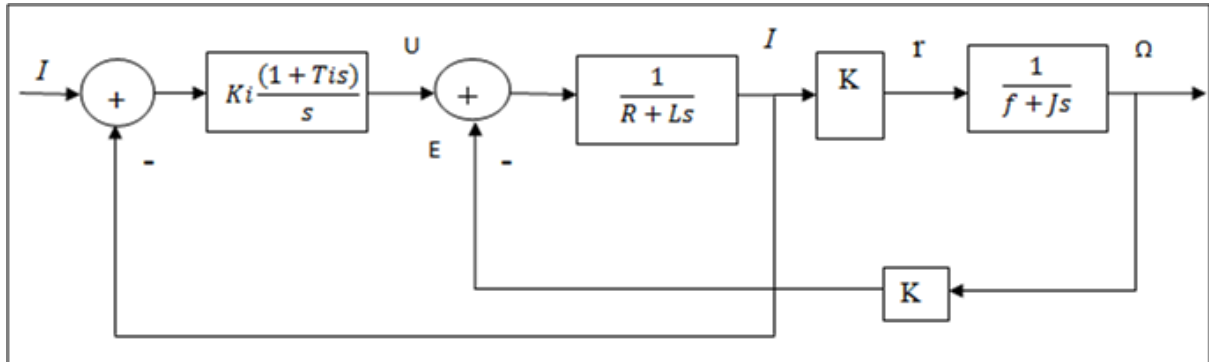


Figure III.18 : Schéma bloc de l'asservissement du courant

Tous les actionneurs ont une commande qui réalise asservissement de couple. Comme le couple est proportionnel au courant de l'induit, la fonction de transfert F(s) entre U(s) et I(s) est donnée comme suit :

$$\frac{I(s)}{U(s)} = \frac{f+Js}{LJs^2+(RJ+Lf)s+Rf+K^2} = F(s) \dots \dots \dots (III.6)$$

La fonction de transfert H(s) entre I* (la consigne de courant) et I (la valeur mesuré du courant), on a :

$$H(s) = K \frac{C(s)F(s)}{1+C(s)F(s)} \dots \dots \dots (III.7)$$

Avec : $C(s)F(s) = \frac{K_i(1+T_i s)}{s(LJs^2+(RJ+Lf)s+Rf+K^2)} \dots \dots \dots (III.8)$

D'où on obtient : $H(s) = \frac{KK_i(1+T_i s)(f+Js)}{Lf s^3+(RJ+Lf+K_i T_i J)s^2+(Rf+K^2+J+T_i f)s+K_i f} \dots \dots \dots (III.9)$

La fonction de transfert entre la consigne de courant et le couple moteur est de 3^{ème} degré. La fonction de transfert entre la consigne de courant et la vitesse du rotor est donc de degré 3.

III.4.3 La technique de la PWM pour la variation de vitesse :

Un moteur à courant continu possède une relation directe entre sa tension d'alimentation et sa vitesse de rotation. En effet, plus la tension à ses bornes est élevée et plus son axe tournera rapidement. Nous sommes en mesure de produire à l'aide de microcontrôleur un signal carré dont le rapport cyclique est variable. Et grâce à cela, nous obtenons une

tension moyenne (comprise entre 0 et 5v) en sortie de la carte Arduino. En utilisant la PWM, on va générer une tension par impulsions plus ou moins grandes.

La technique de la MLI (modulation de largeur d'impulsion) appelée en anglais PWM, me permettra donc de générer une tension par impulsions plus ou moins grandes, Ce signal va commander le transistor ou le relais qui va à son tour commander le moteur. Le moteur va donc être alimenté par intermittences à cause des impulsions de la PWM. Ce qui aura pour effet de modifier la vitesse de rotation du moteur en modifiant le rapport cyclique de la PWM. Plus il est faible (un état BAS plus long qu'un état HAUT), plus le moteur ira doucement. Inversement, plus le rapport cyclique sera élevé (état HAUT plus long que l'état BAS), plus le moteur ira vite.

III.5 Conclusion :

Dans ce chapitre nous avons présenté les différents composants utilisés pour la réalisation de la maquette, ainsi on a fait la description du moteur à courant continu, et nous avons présenté les capteurs utilisés dans la réalisation et son principe de fonctionnement, et à la fin nous avons expliqué le principe de la PWM pour la commande du moteur.



Chapitre IV

Conception et réalisation

IV.1 Introduction :

Réaliser un système quel que soit sa complexité est bien c'est la tâche la plus délicate à faire, vue comme disaient les chercheurs « la preuve de la connaissance est l'expérience ». Donc il faut combiner entre la théorie et la pratique pour pouvoir faire un système performant qui aboutira au cahier de charge souhaité. Dans ce chapitre qui consiste à la réalisation d'un massicot, nous présenterons la maquette réalisée, avec ses différents composants.

IV.2 Conception :

Avant la réalisation pratique de notre maquette nous ferons la conception et la simulation des différents cartes qu'on a besoin à l'utilisé dans cette dernier, qui sont la carte d'alimentation et la carte de puissance. Pour ce faire nous allons utiliser un logiciel de conception assisté par ordinateur (CAO) : il s'agit de logiciel Proteus.

IV.2.1 Présentation de Proteus :

Proteus est un logiciel de la société labo center destiné pour la simulation et le développement des circuits électroniques. Les applications inclus « Proteus » permettent la CAO dans le domaine d'électronique [11].

Le logiciel comprend deux principaux modules : ISIS et ARES.

IV.2.1.1 ISIS :

Le logiciel « ISIS »est un simulateur des cartes électroniques intégré dans le logiciel Proteus de l'entreprise Labcenter Electrnics. L'utilisation de ce logiciel est simple grâce à l'interface graphique. Principalement connu pour éditer des schémas électroniques. Par ailleurs, il permet également de simuler les schémas, ce qui nous permet de déceler certaines erreurs dès l'étape de conception. Indirectement, les circuits conçus grâce à ce logiciel peuvent être utilisé dans des documentations car il permet de contrôler la majorité de l'aspect graphique des circuits. Pour saisir le schéma, il faut créer un nouveau projet puis placer les composants qui doivent être sélectionné à partir de la bibliothèque des composants.

La figure IV. 1 ci-dessous montre la fenêtre du module de simulation Proteus ISIS [11].

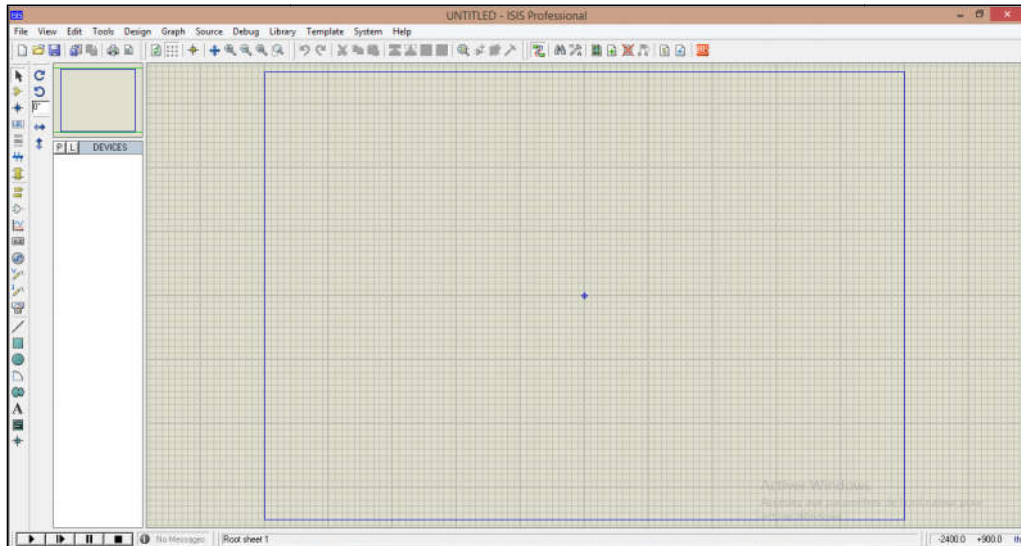


Figure IV.1 : Fenêtre du module de simulation Proteus ISIS

IV.2.1.2 ARES :

Le module ARES est un outil d'édition et de routage qui complète parfaitement ISIS. Un schéma électrique réalisé sur ISIS peut alors être importé facilement sur ARES pour réaliser le PCB de la carte électronique. Bien que l'édition d'un circuit imprimé soit plus efficace lorsqu'elle est réalisée manuellement, ce logiciel permet de placer automatiquement les composants et de réaliser le routage automatiquement. La figure IV.2 ci-dessous montre la fenêtre du module de simulation Proteus ARES [11].

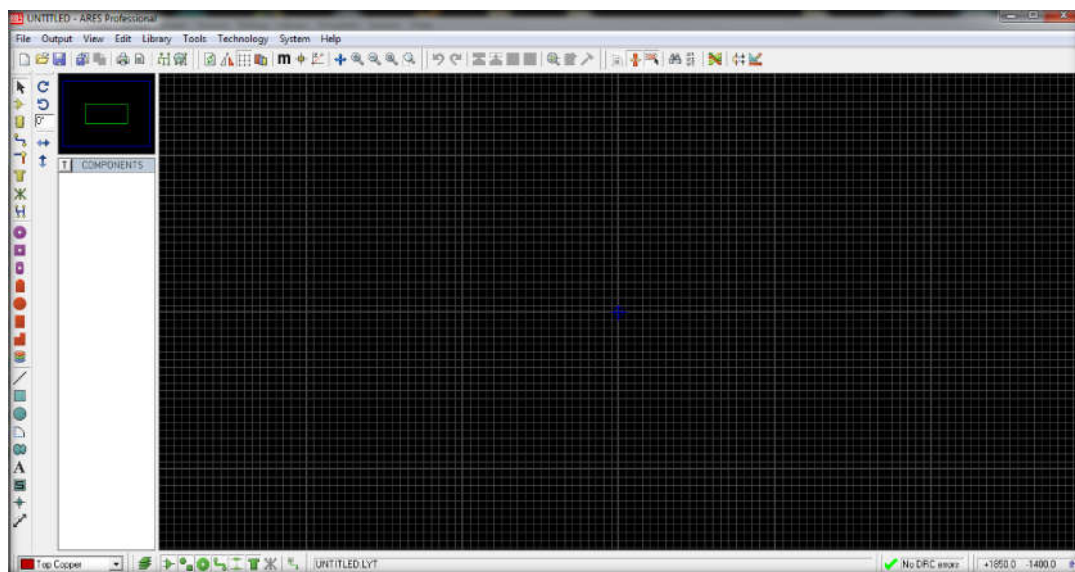


Figure IV.2 : Fenêtre du module Proteus ARES

IV.2.2 Conception et simulation des cartes :

IV.2.2.1 Présentation globale du système :

La conception de ce dispositif prend en considération les problèmes de stabilité, ce qui conduit à de bonnes performances du système global. Elle nécessite une conjonction de plusieurs composants électroniques d'où la création de plusieurs secteurs : d'alimentation, de commande, de puissance et d'affichage.

La configuration globale du système est représentée dans le schéma synoptique de la figure IV.3.

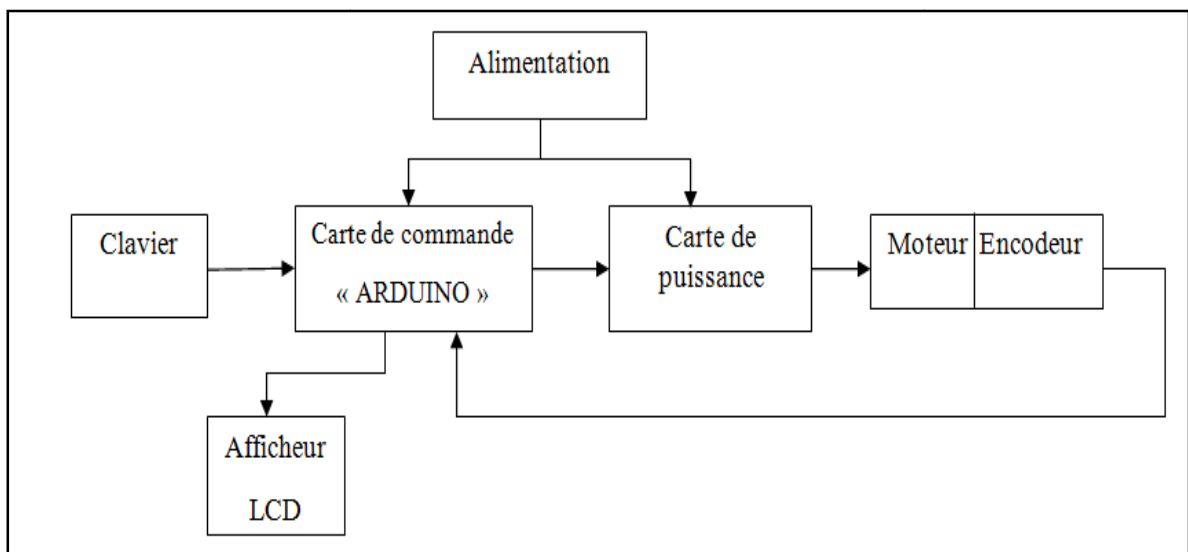


Figure IV.3: Schéma synoptique du système

IV.2.2.2 Les circuits imprimés des cartes :

Cette réalisation est effectuée avec le module ARES du logiciel Proteus pour l'obtention du circuit de la carte imprimé. Après l'impression de la carte on passe à l'implantation des différents composants. Les processus d'obtention de la carte peuvent être vus sur les figures ci-dessous :

❖ La carte d'alimentation :

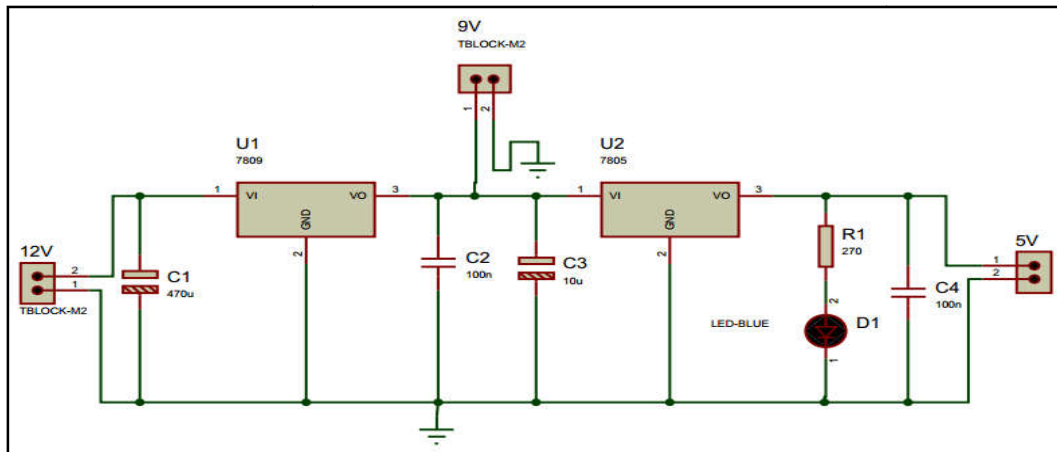


Figure IV.4: Schéma de la carte d'alimentation sur ISIS

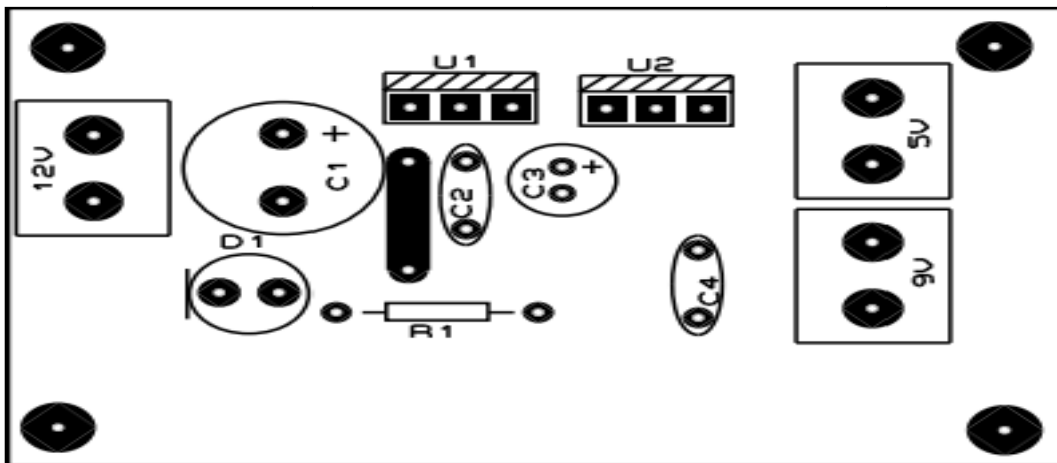


Figure IV.5 : Implantation des composants

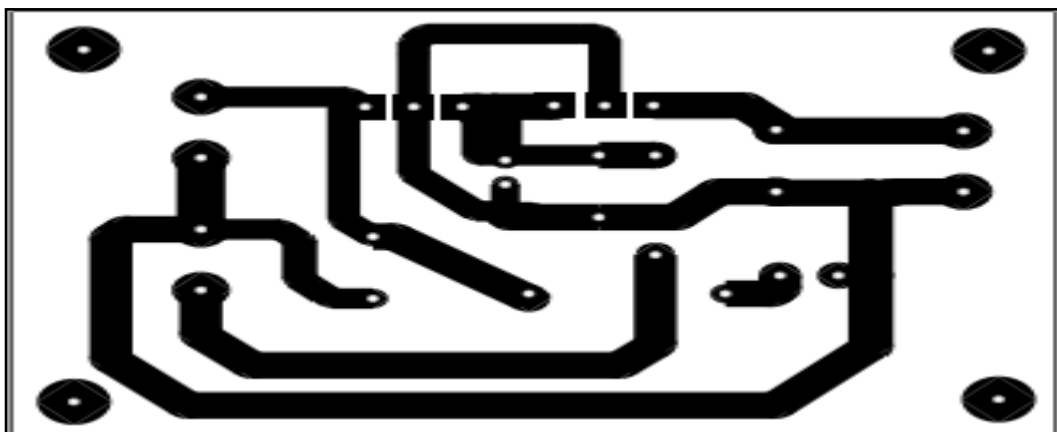


Figure IV.6 : Circuit imprimé sans composants

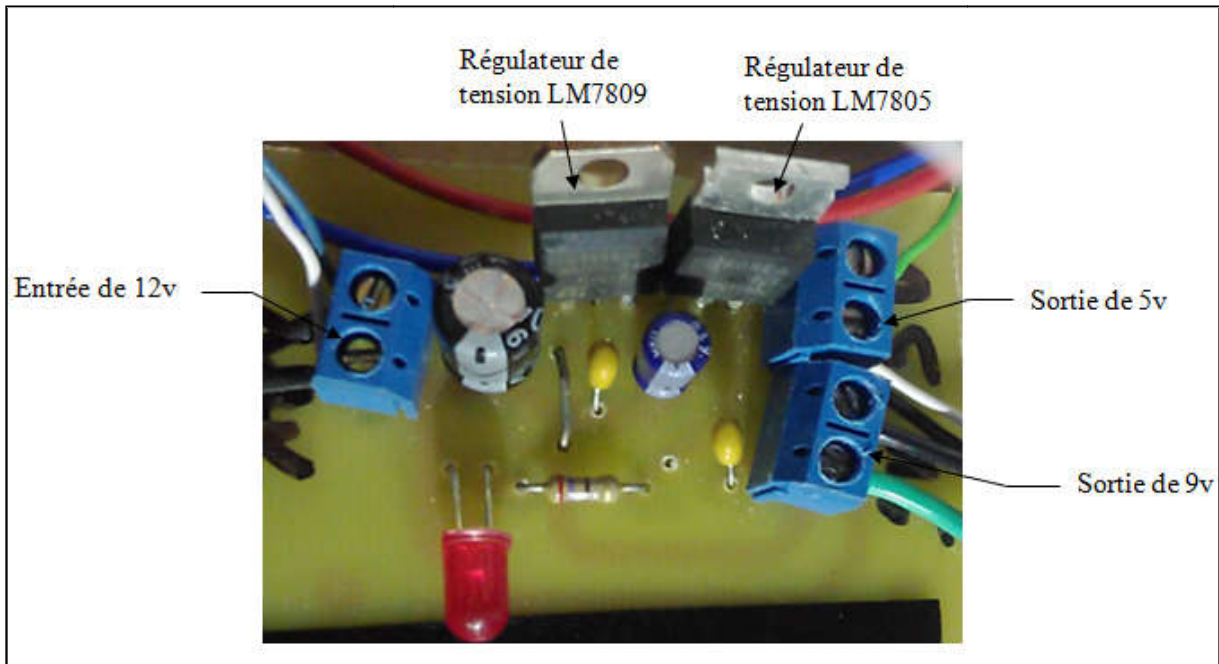


Figure IV.7 : Photo de la carte d'alimentation réalisée

❖ La carte de puissance :

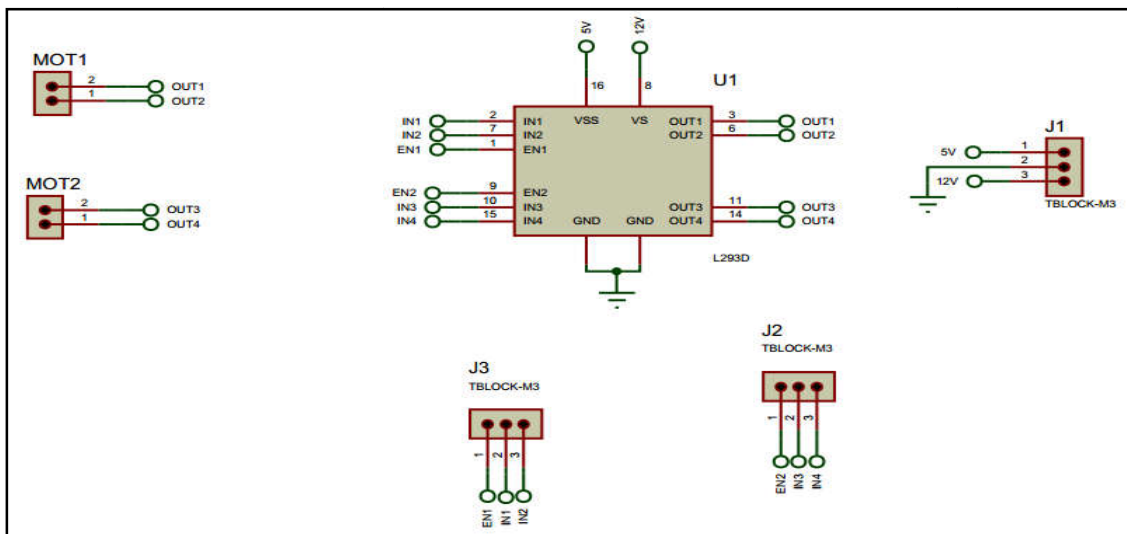


Figure IV.8 : Schéma de la carte de puissance sur ISIS

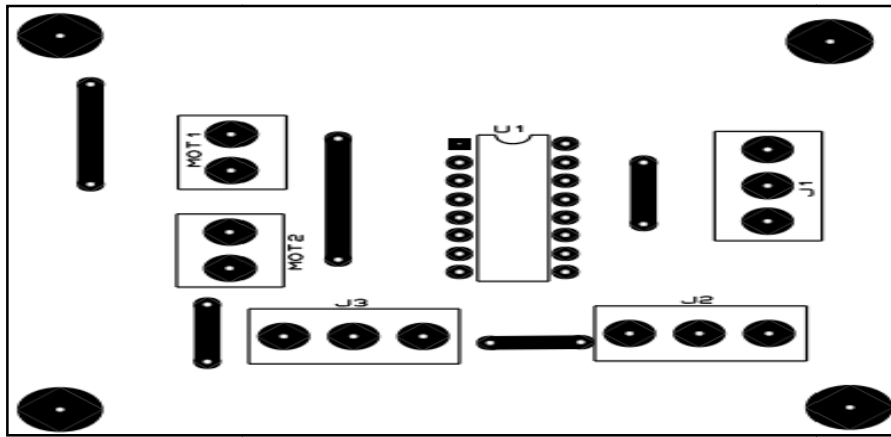


Figure IV.9: Implantation des composants

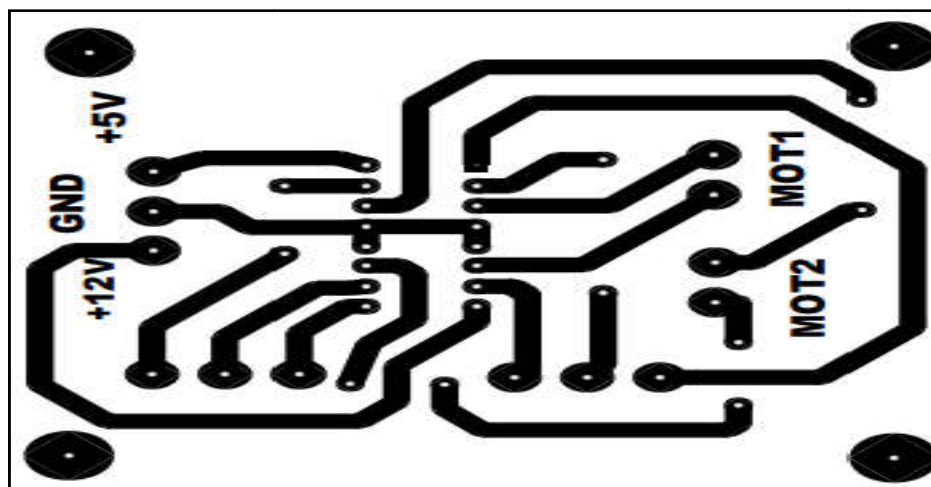


Figure IV.10: Circuit imprimé sans composants

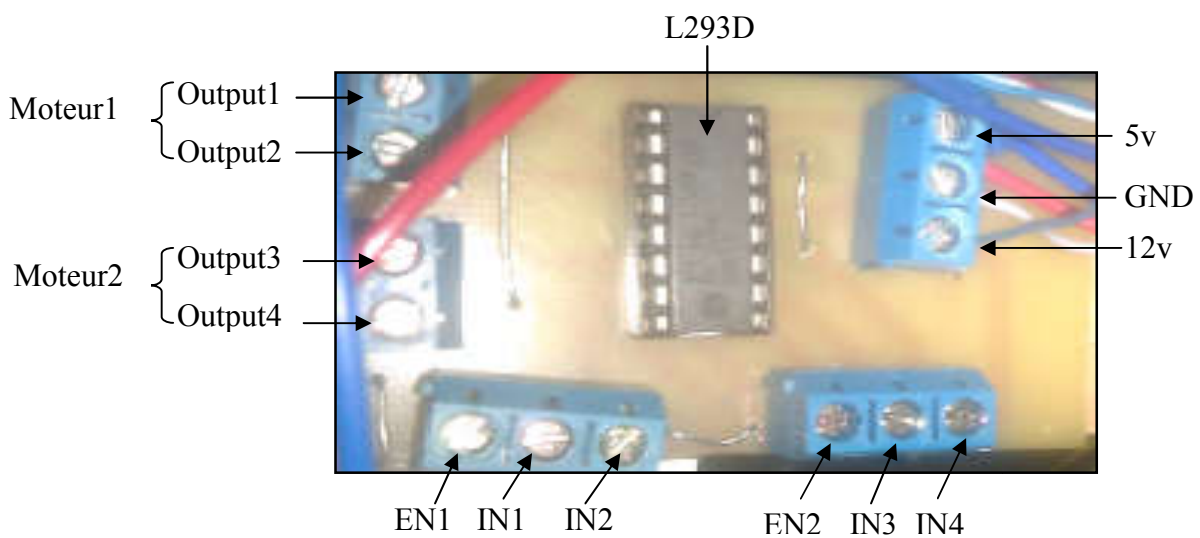


Figure IV.11: Photo de la carte réalisée

IV.3 Partie programmation :**IV.3.1 Cahier de charge :**

- L'allumage et initialisation de la machine,
- Si le capteur de fin de course MIN est actionné donc la machine est initialisée,
- On va donner une commande, saisir une dimension sur le clavier et elle va l'afficher sur l'écran de l'afficheur LCD,
- Dès qu'on valide et cela de taper sur la touche "#" le moteur démarre, la table doit avancer et l'encodeur va calculer la position, lorsque il arrivera à la dimension de papier désirer, le moteur doit s'arrêter et il va donner l'ordre à la descente de la lame pour la coupe.
- Une fois la coupe est terminé, la table va revenir à sa position initial (le capteur de fin de course MIN=1),
- Si le capteur de fin de course MAX est actionné donc il y a une panne dans le système, et ce dernier doit s'arrêter.

IV.3.2 Organigramme fonctionnel de notre système :

Nous avons réalisé un simple organigramme illustré dans la figure IV.12 comporte le fonctionnement de notre machine selon le cahier de charge cité au dessus.

Avec :

CFC MIN : capteur de fin de course minimum.

CFC MAX : capteur de fin de course maximum.

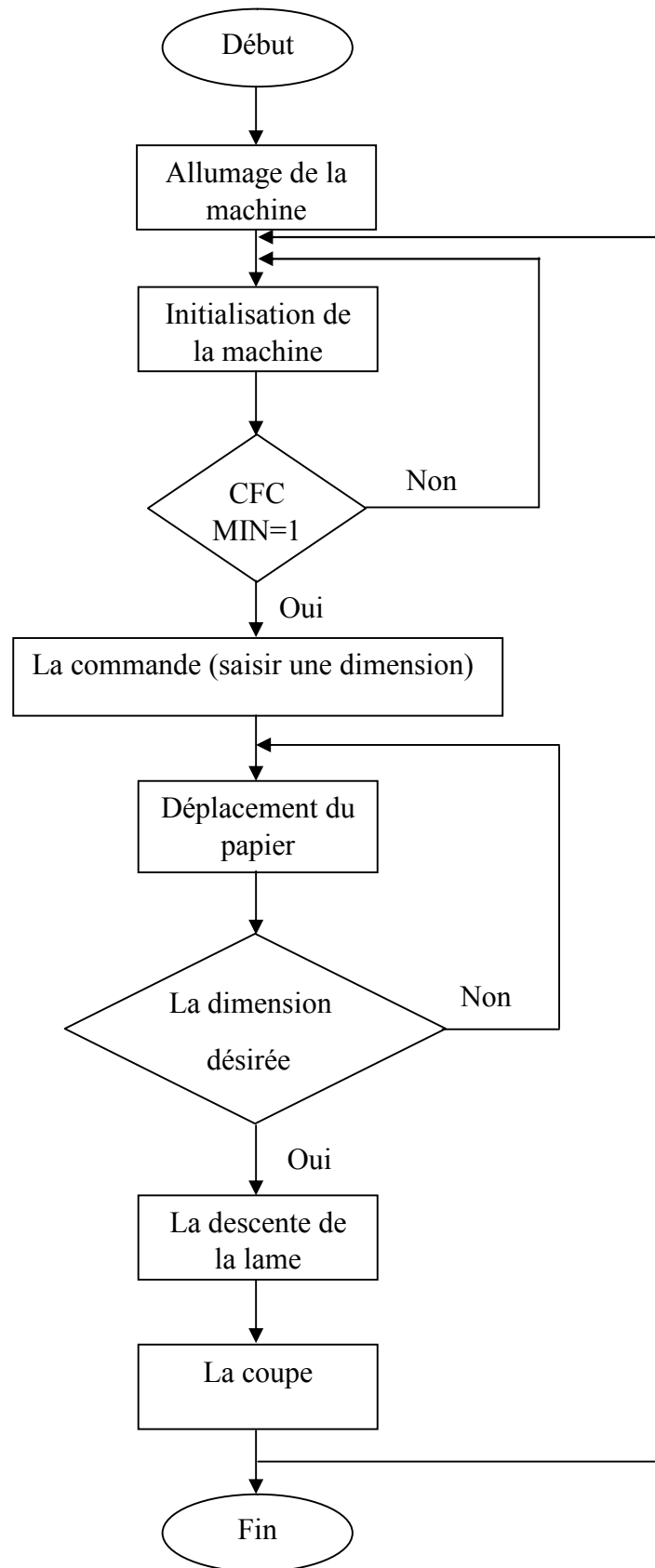


Figure IV.12 : Organigramme fonctionnel du système

IV.3.3 Réalisation du programme :

Dans notre projet, nous avons effectué en langage Arduino qui est très proche du langage C et C++. Le logiciel IDE Arduino permet de compiler et d'envoyer le code directement sur la carte en utilisant un câble USB. Pour que la liaison soit correcte il est nécessaire de sélectionner le type de la carte (Arduino Mega 2560 dans notre cas) et le numéro du port USB.

Avant de commencer la programmation avec ce logiciel, dans notre application on a utilisé un moteur réducteur avec encodeur, pour cela nous avons intégré une nouvelle bibliothèque qui est la bibliothèque de « Rotary Encoder » à Arduino pour commander l'encodeur et donc commander le moteur.

IV.3.4 Organigramme du programme :

Cet organigramme comporte les étapes suivantes :

- ✓ Initialisation de la machine (capteur de fin de course MIN =1),
- ✓ Lecteur de la touche appuyée,
- ✓ Affichage de la touche appuyée,
- ✓ Faire le travail demandé que soit expliqué dans le sous programme.

Nomenclature :

VF : La valeur finale c'est-à-dire la dimension qu'on veut pour la coupe.

CFC MAX : Capteur de fin de course maximum qui se trouve à la fin de la table.

CFC MIN : Capteur de fin de course minimum qui se trouve au début de la table.

Moteur «MAX» : Pour le déplacement de moteur dans le sens positif pour l'avancement de la table donc l'avancement de papier.

Moteur «MIN» : Pour le déplacement de moteur dans le sens négatif pour reculer la table.

Moteur coupe «BAS» : Pour la descente de la lame.

Moteur coupe «HAUT» : Pour la monter de la lame.

Bip sys : Bip sonore du système.

IV.3.5 Les fonctions utilisées :

Les fonctions utilisées dans le programme sont :

clavier.getKey () : pour lire la touche de clavier appuyée.

lcd.clear () : pour effacer l'écran de l'afficheur LCD.

lcd.setCursor() : pour la position de curseur de l'afficheur LCD.

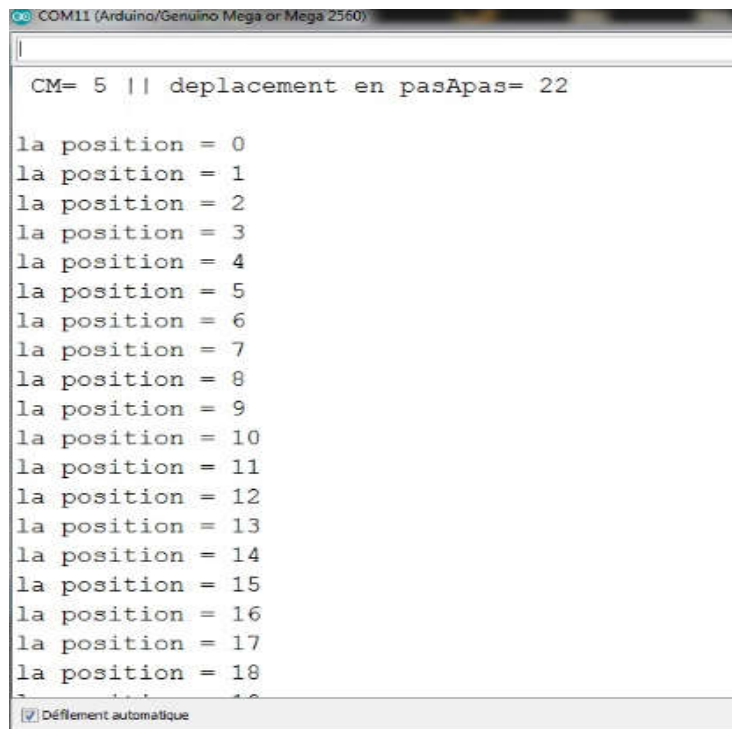
lcd.Print () : pour écrire sur l'écran de l'afficheur.

encoder.getPosition () : pour lire la valeur de la position souhaitée.

IV.3.6 Tests :

Après la réalisation de la maquette, nous avons fait un dernier test, tel que nous avons donné une dimension de 5cm et nous avons obtenu les résultats illustrés dans la figure IV.13.

On remarque que pour une valeur de 5 cm l'encodeur à déplacer de 22 pas.



```
COM11 (Arduino/Genuino Mega or Mega 2560)
|
CM= 5 || déplacement en pasApas= 22
la position = 0
la position = 1
la position = 2
la position = 3
la position = 4
la position = 5
la position = 6
la position = 7
la position = 8
la position = 9
la position = 10
la position = 11
la position = 12
la position = 13
la position = 14
la position = 15
la position = 16
la position = 17
la position = 18
|
 Défilement automatique
```

Figure IV.13: Les valeurs données par l'encodeur

IV.3.7 La maquette réalisée :

Notre maquette réalisée est donnée par la figure IV.14 ci-dessous :



Figure IV.14 : La maquette réalisée

IV.4 Conclusion :

Dans ce dernier chapitre nous avons fait une conception des cartes utilisée, et on a présenté notre réalisation, ainsi les différents fonctions pour la programmation de la commande, et la fin nous avons donné les résultats pour un test fait.

Conclusion générale

Conclusion générale

L'élaboration de ce modeste travail nous a conduits à combiner théorie et pratique afin de mettre en œuvre un dispositif de commande numérique d'un massicot.

En effet, ce mémoire de fin d'étude est une expérience très bénéfique, il nous a permis d'enrichir nos connaissances déjà acquises durant notre formation et surtout de mettre nos connaissances dans le domaine pratique.

Primo, du point de vue logiciel, ce projet nous a orientés directement vers de différents logiciels de programmation telle que le logiciel IDE Arduino, PROTEUS et ARES.

Secundo, sur le plan personnel, nous avons pu s'intégrer dans un milieu professionnel, nous avons pris une certaine expérience concernant la conception et la réalisation d'une maquette qui était la tâche la plus difficile car il faut faire une combinaison entre la théorie et la pratique afin de réaliser un système performant qui aboutira au cahier de charge souhaité.

Malgré les difficultés, le résultat final est un succès car la maquette fonctionne très bien. Enfin on espère que nos efforts puissent servir à quelque chose et que ce mémoire soit un bon guide pour les futures promotions.

Bibliographie

- [1] : Pro.matériels.com, consulté le 24/02/2016.
- [2] : matériel-pour-bureau.fr, consulté le 24/02/2016.
- [3] : SIMON Landranlt, HIPPOLYTE Weisslinger, « Premier pas en informatique embarquée », Edition CC BY-NC-SA, 01Juin 2014, 268p.
- [4] : Ouahmed MALKI, «Commande en force et position d'une architecteur de télé opération », projet de fin d'étude UMMTO, 2015.
- [5] : Massimo Banzi, Michael Shiloh « Getting started with Arduino », 3rd Edition, Décembre 2014, 245p.
- [6] : WWW.arduino.cc, consulté le 16/05/2016.
- [7] : <http://nalhossri.free.fr>, consulté en Mai 2016.
- [8] : <http://bonemine.lycee>, consulté en Mai 2016.
- [9] : <http://www.elekronique.fr>, consulté en Mai 2016.
- [10] : D. GRIDAINE «S.T.S Maintenance Industriel », 10p.
- [11] : Guide d'utilisateur du logiciel Proteus.