

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITÉ MOULOUD MAMMERI DE TIZI-OUZOU



FACULTÉ DE GENIE ELECTRIQUE ET D'INFORMATIQUE
DÉPARTEMENT D'AUTOMATIQUE

**Mémoire de Fin d'Etudes
de MASTER ACADEMIQUE**
Spécialité : **automatique option commande des
systèmes**

Présenté par

**ABDESSALAM IMMOUNE
DJOUNADI MOHAND**

Mémoire dirigé par **Mr Redouane KARA**

Thème

**Commande numérique d'un massicot
avec la carte Arduino MEGA 2560**

Mémoire soutenu publiquement le 14/09/2015 devant le jury composé de:

Mr Said DJENNOUNE

Professeur , UMMTO, Président

Mr Redouane KARA

MCA , UMMTO, Encadreur

Mr Ahmed MAIDI

MCA , UMMTO, Examineur

Mr Rabeh MELLAH

MCA , UMMTO, Examineur

Ce travail a été réalisé au laboratoire de l'ummtto

Nous dédions ce mémoire à :

Nos chers parents,
Nos familles,
Tous nos amis(es) et nos camarades.

Remerciements

Nous remercions d'abord DIEU le tout puissant pour la santé, le courage et la foi qu'il nous a donné pour arriver à ce jour.

Nous tenons à remercier notre encadreur
M. Redouane KARA pour son aide, ses précieux conseils,
et sa patience tout au long de ce travail.

Nous tenons aussi à remercier M. TAHANOUT pour son aide et
sa patience.

Nous remercions les membres du jury qui nous ont fait
l'honneur d'examiner notre travail.

Nos remerciements les plus chaleureux vont à nos chers
parents pour leurs encouragements, leur patience, et leur
grand soutien durant toutes ces années d'études.

Nos sincères remerciements à tous ceux qui ont participé de
près et de loin à l'achèvement de ce travail.

Sommaire

<i>Introduction générale</i>	1
<i>Chapitre I : Présentation du Massicot</i>	3
I.1 Définition du massicot.....	3
I.2 Histoire	3
I.3 Les différents types de massicots	4
I.3.1 <i>Massicot industriel</i>	4
I.3.2 <i>Massicot électrique</i>	4
I.3.3 Massicot de bureau	5
I.3.4 Massicots rotatifs	5
I.3.5 Massicot coup parquet	6
I.3.6. Massicot d'imprimerie	7
I.3.7 Les massicots modernes	7
I.4 Le principe de fonctionnement du massicot	8
I.5 Lame du massicot et autre accessoires de coupe	8
I.5.1 Les lames du massicot	8
I.5.2 Les pieds supports	9
I.6 Sécurité	10
I.6.1 Les dispositifs de sécurité	10
I.7 Conclusion	11
<i>Chapitre II : Moteurs et Encodeurs</i>	12
II.1.1 Introduction	12
II.1.2 Principe de fonctionnement.....	12
II.1.3 Les types de codeurs optiques.....	13
II.1.3.1 Le codeur incrémental	13
II.1.3.1.1 Particularités de fonctionnement.....	13

II.1.3.2Le codeur numérique (codeur absolu).....	14
II.1.4Avantages, Inconvénients des codeurs incrémentaux et absolus	15
II.1.4.1Codeur incrémental	15
II.1.4.2Codeur absolu.....	15
II.1.5Choix de codeur :	16
II.2.1 Moteur à courant continu	16
II.2.2Constitution	17
II.2.2.1L'inducteur :	17
II.2.2.2L'induit (circuit de puissance)	17
II.2.2.3Le collecteur et les balais.....	18
II.2.3description et principe de fonctionnement	18
II.2.4Les types de la machine	18
II.2.4.1Fonctionnement en moteur.....	19
II.2.4.2Fonctionnement en génératrice (dynamo).....	19
II.2.5Les différents types de la machines à courant continu	19
II.2.5.1Formules mathématique	21
II.2.6Puissance et rendement	22
II.2.7Modélisation et Asservissement du moteur	23
II.2.7.1Modélisation du moteur.....	23
II.2.7.2Asservissement du moteur.....	23
II.2.7.2.1Asservissement couple/courant	24
II.2.7.3La technique de la PWM pour La variation de vitesse	25
II.2.7.4Pilotage du sens et vitesse du moteur.....	25
II.8 conclusion	26
Chapitre III : Présentation de l'Arduino	27
III.1 Introduction :.....	27
III.2 Présentation Arduino	27

III.3	Avantage de l'Arduino	27
III.4	Les outils de l'Arduino	28
III.4.1	Architecture HARDWARE de la carte	29
III.4.2	Architecture de la carte	29
III.4.3	Le microcontrôleur	29
III.4.3.1	Caractéristiques du microcontrôleur ATMEGA 2560	30
III.4.4	Le entrées-sorties de la carte :.....	31
III.4.5	Mémoire :.....	31
III.4.6	L'alimentation :.....	31
III.4.7	Visualisation :	32
III.4.8	La connectique :.....	32
III.5	L'architecture SOFTWARE de la carte :.....	33
III.5.1	Présentation de l'Espace de développement Intégré (EDI) Arduino :.....	33
III.5.2	1Différentes composition de l'espace de développement intégré (EDI):.....	33
III.5.2.2	Une barre des boutons	34
III.5.2.3	Structure d'un programme Arduino :.....	35
III.5.2.4	Le programme Arduino	35
III.5.2.4.1	Description de la structure d'un programme Arduino :.....	35
III.5.2.4.2	Le jeu d'instructions du langage Arduino :.....	36
III.5.2.4.3	Les étapes à suivre pour programmer la carte :	39
III.6	CONCLUSION	41
	Chapitre IV : Réalisation et Conception	42
IV.1	Introduction	42
IV.2	Schémas synoptique de la carte d'essai mise en œuvre	42
IV.4	Le matériels utilisés	42
IV.4.1.1	Le moteur utilisé et ses caractéristiques	42

IV.4.1.2 Partie mécanique	43
IV.4.2.1 Présentation du boîtier L293D	43
IV.4.2.2 Brochage	44
IV.4.2.3 la Logique de commande Input	46
IV.5Afficheur à cristaux liquides (LCD)	46
IV.5.1 Introduction.....	46
IV.5.2 Schéma fonctionnel	47
IV.5.2 Brochage	48
IV.6Le clavier à matrice	49
IV.7Conclusion	51
Conclusion générale.....	52

Liste des figures

Figure 1 : massicot électrique

Figure 2 : massicot de bureau

Figure 3 : massicot rotatif

Figure 4 : massicot coupe parquet

Figure 5 : massicot d'imprimerie

Figure 6 : les lames de massicot

Figure 7 : principe de fonctionnement de photo-transistor

Figure 8 : schéma de codeur incrémental

Figure 9 : schéma pour indique le sens de rotation

Figure 10 : codeur absolu

Figure 11 : symbole de moteur

Figure 12 : l'inducteur de moteur

Figure 13 : l'induit de moteur

Figure 14 : les balais collecteurs de moteur

Figure 15 : convertisseur électromécanique

Figure 16 : fonctionnement en génératrice

Figure 17 : moteur à excitation indépendante

Figure 18 : moteur à aimant permanent

Figure 19 : excitation shunt

Figure 20 : moteur à excitation en série

Figure 21 : excitation composé

Figure 22 : perte d'énergie dans un moteur

Figure 23 : schéma bloc du moteur à courant continu (MCC)

Figure 24 : schéma bloc de l'asservissement du courant

Figure 25 : architecture interne de la carte Arduino Mega 2560

Liste des figures

Figure 26 : architecture de la carte Arduino Mega 2560

Figure 27 : architecture de microcontrôleur ATMEGA 2560

Figure 28 : les différentes parties de la fenêtre principale du logiciel Arduino

Figure 29 : les différentes composantes du programme Arduino

Figure 30 : l'écriture de programme

Figure 31 : compilation du programme

Figure 32 : console d'affichage

Figure 33 : sélection de la cible

Figure 34 : selection du port

Figure 35 : transfert du programme

Figure 36 : clignotement des deux LED du port série

Figure 37 : schéma global qui représente les blocs essentiels de la carte d'essai

Figure 38 : moteur réducteur avec la carte encodeur

Figure 39 : utilisation d'un encodeur optique sur l'axe du moteur

Figure 40 : les bronches de L293D

Figure 41 : schéma fonctionnel d'un LCD

Figure 42 : photo d'un LCD et son affichage

Figure 43 : clavier

Figure 44 : schéma de simulation

Introduction générale

Une machine-outil à commande numérique (MOCN, ou simplement CN) est une machine-outil dotée d'une commande numérique. Lorsque la commande numérique est assurée par un ordinateur, on parle parfois de machine CNC pour *computer numerical command*, francisé en « commande numérique par calculateur ».

Les machines à commandes numérique (CNC) sont utilisées aujourd'hui dans différents domaines d'application. Nous avons les machines-outils (fraiseuse, tours numériques etc.), les massicots utilisés pour faire du prototypage rapide ainsi que les machines de perçage automatique.

Assignées à leur origine à l'usinage dit « 3 axes », de par les performances et possibilités des matériels et logiciels associés sans cesse croissant, les commandes numériques sont de plus en plus utilisées dans les procédés où il est requis de déplacer un ou des mobiles dont les mouvements sont interpolés (vitesses des axes concourant au déplacement d'un point Liées entre elles) avec des contraintes de vitesse très faibles ou très élevées et/ou de très grande précision de trajet ou de positionnement.

Dans notre étude, on a essayé de rassembler les différents éléments qui constituent une machine à commande numérique. Nous avons la partie électronique qui constitue l'interface de commande entre l'utilisateur et la machine, elle permet à l'utilisateur de transmettre des commandes simples à la machine et ceci via une interface (dans notre cas, nous avons utilisé l'interface USB).

La partie électronique, sur laquelle se base notre étude, transforme les commandes envoyées par l'utilisateur en impulsion électrique dans le but de faire tourner des moteurs (dans notre cas, le moteur à courant continu) qui produit à son tour des mouvements mécanique de translation dans différentes directions.

Notre étude va se baser essentiellement sur la partie électronique. Il y a tout d'abord l'interface Homme-Machine, celle-ci permet à l'utilisateur d'effectuer des commandes via un logiciel que nous avons mis au point à l'aide d'un langage de programmation évolué (Arduino IDE). Ce logiciel transmet des données à la carte de commande via l'interface USB.

La carte électronique est pilotée par un microcontrôleur (ARDUINO MEGA 2560) qui assure la conversion des commandes envoyées par l'utilisateur aux signaux électriques permettant de faire tourner le moteur à courant continu.

Introduction générale

Pour la partie mécanique, nous avons essayé de la mettre au point du mieux qu'on a pu, et ceci afin de montrer le fonctionnement d'une machine à commande numérique même si elle ne rentre pas dans le domaine de notre spécialité et qu'elle relève du domaine des applications mécaniques.

Le premier chapitre est consacré pour la présentation des machines outil à commande numérique MOCN, un aperçu général sur la technologie des MOCN, citant la classification, l'architecture d'un massicot.

Dans le second chapitre nous avons présenté les codeurs optiques et les moteurs à courant continu et leur principe de fonctionnement.

Le troisième chapitre est consacré à la présentation de la carte Arduino et son logiciel ainsi que les différentes étapes pour le chargement du programme dans la carte.

Dans le quatrième chapitre nous avons présenté le matériel utilisé dans la réalisation avec la simulation des programmes en utilisant le logiciel Protues.

Pour terminer, une mise en pratique de notre travail a été réalisée, et une conclusion générale a été tirée.

I.1 Définition du massicot :

Le massicot est un appareil destiné à la coupe à angle droit du papier et d'autres matériaux, des feuilles métalliques par exemple. On trouve souvent les massicots dans le domaine de l'imprimerie qui ont beaucoup de préparation de feuille ou de façonnage d'imprimés à effectuer. Cependant on trouve des modèles de massicot de plus en plus variés.

Les plus petits (d'une ouverture de 66 cm et plus) peuvent trouver leur utilité dans des imprimeries intégrées, mais aussi dans des entreprises travaillant avec des formats plus petits. On trouve des massicots plus grands (ouverture jusqu'à 200 cm) dans des systèmes interconnectés complexes.

Il existe également des massicots à plusieurs lames pour pouvoir faciliter la découpe de certains travaux (exemple : le massicot trilatéral pour tout ce qui est des livres) [1].

I.2 Historique :

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 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.

I.3 Les différents types de massicots :

I.3.1 Massicot industriel :

Un massicot industriel, est généralement sous un grand format et permet de faire des coupes sur plusieurs matériaux (feuille, carton, plastique, ...), avec une manière automatiser d'un débit bien plus important qu'un massicot de bureau.

Comme n'importe quel massicot, le massicot industriel doit répondre à des normes de sécurité pour minimiser les risques dus à la découpe.

En général un massicot industriel répond à une très grande précision dans sa découpe et possède un panneau de bord digital pour pouvoir entrer des mesures précises et répétées autant de fois souhaité pour un résultat en parfaite égalité [1].

I.3.2 Massicot électrique :

Un massicot électrique permet une coupe précise et sécurisé, puisqu'il n'est plus question de levier à utiliser manuellement. Sur le massicot électrique l'entraînement de la lame se fait électriquement, pouvant cependant garder un système de pression par levier. Selon les modèles il y a une réglette de mesure de couper avec précision.

Un massicot électrique peut exister en plusieurs versions : massicot électrique de table, sur stand ou sur meuble. Le système de sécurité est toujours très important. Certains sont dotés d'un système infrarouge pour détecter des mouvements qui ne devraient pas avoir lieu, stoppant la machine sur le champ, d'autres ne démarrent que sous la pression de deux boutons simultanés, et ainsi de suite.

On trouve également des massicots électriques avec un système de réglage par commande électronique pour le déplacement de la butée, avec un affichage digital des dimensions pour une plus grande position.



Figure 1 : massicot électrique

I.3.3 Massicot de bureau :

Le massicot de bureau est souvent d'un format A4, est assez petit pour être concerné sur une table ou sur un bureau, afin de pouvoir effectuer des découpes à n'importe quel moment. Simple à utiliser, le massicot de bureau peut disposer de trois têtes de découpe, dont une servira à découper, l'autre à plier et la troisième pour effectuer un pré découpage sous forme de pointillé. On trouve des massicots de bureau complètement métallique, avec un système de tracé des formats pour permettre de mieux visualisée la découpe.

A savoir qu'un massicot de bureau peut couper le papier, les chemises en cartons ou plastique, les documents plastifier, les films transparents, le bristol.



Figure 2 : massicot de bureau

I.3.4 Massicots rotatifs :

Un massicot rotatif peut avoir plusieurs particularités et de nombreuses fonctions, selon le modèle choisi. Leurs points communs sont que, ce sont leurs lames qui sont rotatives pour permettre une découpe rapide et sans le moindre effort. Ce système permet des découpes

nettes, mais aussi des trainages, des pré-découpages et des dentelles. On peut trouver des massicots rotatifs en format A4, A3, Comme sur n'importe quel modèle, les mesures de sécurité ne sont pas à négliger sur un massicot rotatif.



Figure 3 : massicot rotatif

I.3.5 Massicot coupe parquet :

Un massicot coupe parquet, ou massicot guillotine pour parquet, permet de pouvoir faire des découpes nettes et précises de manière à mettre la planche de parquet à la hauteur souhaitée pour un meilleur rendu. Il permet une découpe sans avoir à utiliser d'électricité, grâce à un système de levier. Contrairement à une scie, le massicot a parquet évite les déchets de poussières, évite les pertes et permet une coupe franche et directe pour un résultat professionnel.



Figure 4 : massicot coupe parquet

I.3.6 Massicot d'imprimerie :

Le massicot d'imprimerie est une grande machine de découpe à angles droit, qui 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.

La différence entre le massicot imprimerie et la machine mono lame ou multi lame. la deuxième est beaucoup plus chère et pas indispensable.

Comme pour les autres massicots, ceux de l'imprimerie laisse le choix entre le manuel et l'automatique selon les besoins. La sécurité est toujours importante dans n'importe quelle machine [2].



Figure 5 : massicot d'imprimerie

I.3.7 Les massicots modernes :

Les massicots modernes sont munis d'un logiciel ou d'un système embarqué permettant de les intégrer dans des systèmes interconnectés en réseau. L'informatique permet de mémoriser des programmes complexes qui intègrent une multitude de fonctions et, avec l'ajout de périphériques tels que des élévateurs, traqueuses, chargeurs, dépileurs, de constituer

des lignes (ou chaînes) de coupe, rendant ainsi le travail moins pénible, plus efficace et plus sûr.

I.4 Principe de fonctionnement du massicot :

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 rogner. À l'arrière de la table se trouve une équerre dont la position est réglable de manière très précise, elle définit la largeur de coupe. La pile de papier à couper vient en butée sur cette équerre et des repères (aujourd'hui lumineux) 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 appuie simultanément sur une pédale et sur deux boutons qui déclenchent la descente de la presse et de la lame.

Il existe aussi des massicots à plusieurs lames, rendant la découpe de certains travaux plus aisée, comme le massicot trilatéral pour les livres.

I.5 Lames du massicot et autre accessoires de coupe :

I.5.1 Lames du massicot :

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'usure, notamment lorsque le massicot sert à couper des cartons épais. C'est pourquoi 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.



Figure 7 : les lames de massicot

On se rend compte qu'une lame de massicot est abîmée lorsque :

- La lame ne coupe plus de façon précise le papier : un défaut de précision directement provoqué par l'usure de la lame.
- La lame se coince régulièrement dans une ramette.
- La lame laisse des rayures sur le papier.

Ces signes bien visibles indiquent qu'il faut absolument changer ou affûter la lame de massicot pour en assurer le bon fonctionnement et un rendu efficace.

Le fil d'une lame de massicot étant particulièrement tranchant, il convient de changer la lame de l'appareil avec les plus grandes précautions. Une seule personne doit être dédiée à la tâche et y apporter le plus grand soin [2].

I.5.2 Les pieds supports :

Les pieds supports pour 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 supports, ce qui garantit une position confortable à l'utilisateur.

Certains fabricants proposent des pieds supports munis de tablettes pour déposer le papier et faciliter ainsi la manipulation de ramettes, ce qui confère encore un avantage en termes de confort pour l'utilisateur.

I.6 Sécurité :

Les massicots ont dans le passé provoqué de nombreux accidents. C'est pourquoi les mesures de sécurité sont particulièrement renforcées : l'obligation d'utiliser ses deux mains et un pied pour déclencher la coupe, mais aussi des protections par infrarouges, le démontage du capot de protection empêche toute mise en marche, et de nombreux boutons d'arrêt « coup de poing » garantissent une sécurité maximum [3].

I.6.1 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'encrant au mieux dans le sol.
- Les réglottes pour massicots : ils protègent la lame de votre 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 1995. 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.

Afin de garantir une sécurité optimale lors des opérations de changements de 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 permettent de raccourcir efficacement les opérations de maintenance des massicots papier [3].

I.7 Conclusion :

Dans ce chapitre on a présenté une recherche bibliographique sur La machine –outils à commande numérique (MOCN), qui est le massicot en présentant les modèles plus utilisées

L'utilisation des massicots présente un grand intérêt pour les imprimeries

II.1.1 Introduction :

La croissance de la puissance des systèmes de traitement ainsi que les impératifs de productivité dans tous les domaines de production industrielle poussent à l'acquisition et commandes continue, comme : le déplacement, la position, la vitesse des outils ou des produits.

Selon les besoins, les codeurs montés sur un système d'entraînement ont des tâches différentes. Une des tâches principales d'un codeur que nous avons utilisé dans notre montage est de permettre de mesurer la vitesse et l'angle de rotation du moteur électrique alimenté en courant continu.

II.1.2 Principe de fonctionnement des encodeurs :

Tous les codeurs optiques exploitent des principes de fonctionnement similaires. Ils sont constitués d'un disque comportant des zones opaques et des zones translucides. Le nombre de ces zones et leur disposition dépend de la nature du codeur et du type d'information que l'on souhaite obtenir.

Des diodes électroluminescentes (LED) émettent une lumière qui peut traverser les zones translucides. Des photos-transistors, situés de l'autre côté du disque en regard des LED, captent cette lumière lorsqu'ils sont face à une ouverture et délivrent un signal électrique, image de la présence de cette ouverture [4].

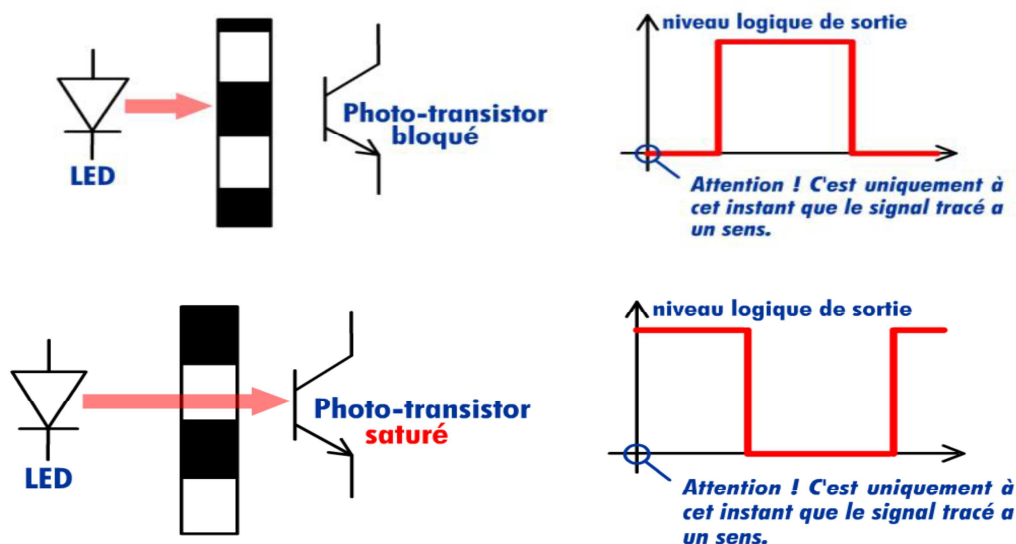


Figure 7 : principe de fonctionnement de photo-transistor

II.1.3 Les types de codeurs optiques :

Il existe deux types de codeurs optiques :

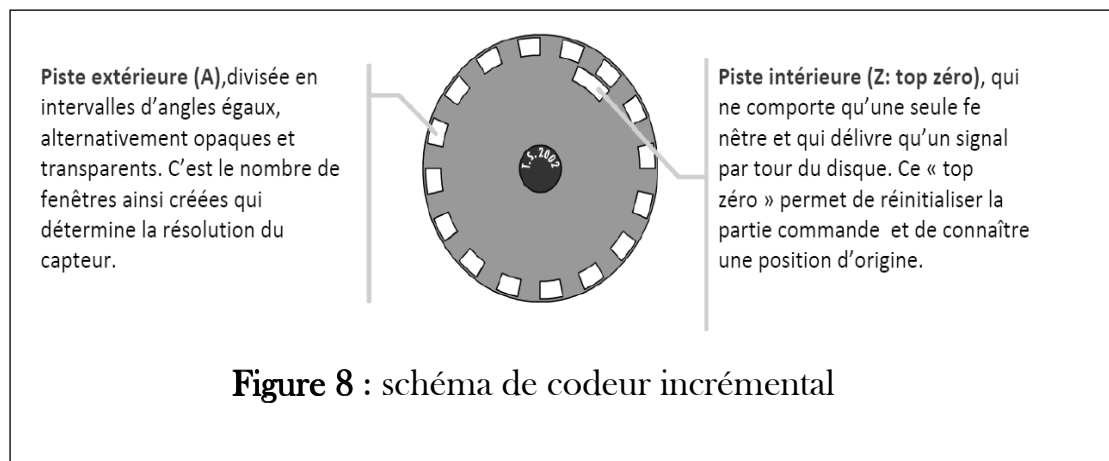
Les codeurs incrémentaux qui délivrent une information de déplacement angulaire du disque sous forme d'un train d'impulsions.

Les codeurs numériques de position (codeurs absolus), pour lesquels chaque position du disque correspond à une valeur numérique différente identifiable par la partie commande [4].

II.1.3.1 Le codeur incrémental :

Le codeur incrémental est destiné à des applications où l'information de position est obtenue par la mesure du déplacement de l'objet. Le codeur délivre un train d'impulsions dont le nombre permet de déduire la valeur du déplacement ainsi que la vitesse car ce dernier est proportionnel à la fréquence des impulsions.

Il est constitué d'un disque comportant deux à trois pistes : A et Z.



Pour un tour complet de l'axe du codeur, la partie commande reçoit autant d'impulsions électriques qu'il y a de fenêtres, dont la durée dépend de la vitesse de rotation du disque.

II.1.3.1.1 Particularités de fonctionnement :

Un codeur incrémental possède trois têtes de lecture :

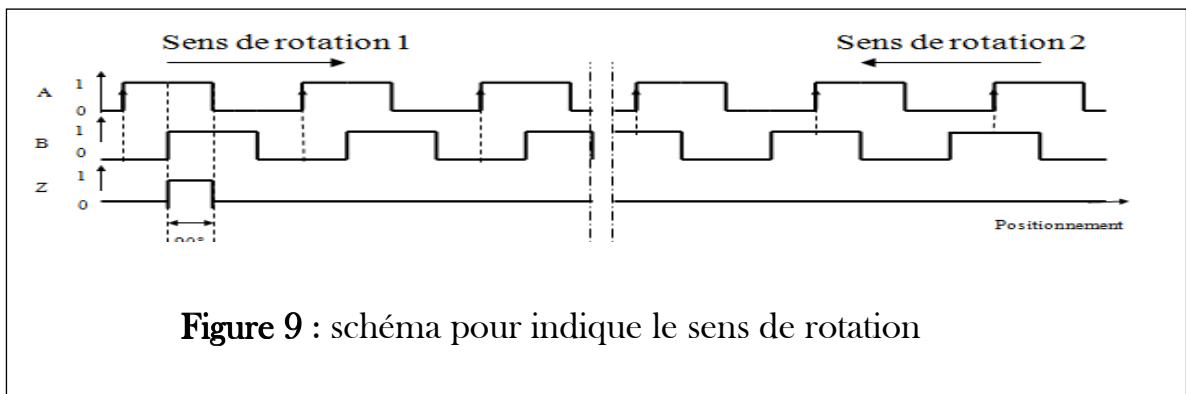
Une tête de lecture est affectée à la piste intérieure et délivre une impulsion par tour, permettant à la commande de compter le nombre d'impulsions reçues.

Deux têtes de lecture sont placées sur la piste extérieure. 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 émis sont décalés dans le temps. 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 :

- dans le sens de rotation 1, B = 0 au front montant de A.
- dans le sens de rotation 2, B = 1 au front montant de A.



II.1.3.2 Le codeur numérique (codeur absolu) :

Le codeur numérique de position est destiné à des applications pour lesquelles on souhaite obtenir l'information de position sans traitement par la partie commande. Il est constitué d'un disque comportant plusieurs pistes concentriques et d'une tête de lecture par piste. Le nombre de piste détermine le nombre de positions différentes qui peuvent être définies à l'intérieur d'un tour de disque. Les codeurs industriels comportent jusqu'à 24 pistes.

La partie commande reçoit directement un code numérique sur n bits (n étant le nombre de pistes), image de la position du disque à un instant donné. A l'intérieur d'un tour de disque, cette information est donc une information de position absolue (à la différence d'un codeur incrémental qui ne délivre qu'une information de déplacement par rapport à une origine qu'il a fallu définir au préalable).

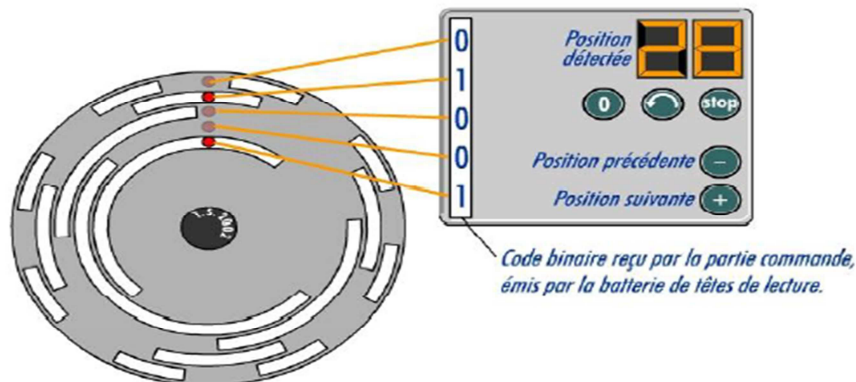


Figure 10 : codeur absolu

II.1.4 Avantages, Inconvénients des codeurs incrémentaux et absolus :

II.1.4.1 Codeur incrémental :

➤ **Avantage :**

Le codeur incrémental est de conception simple (son disque ne comporte que deux pistes) donc plus fiable et moins onéreux qu'un codeur absolu.

➤ **Inconvénients :**

Il est sensible aux coupures du réseau : chaque coupure du courant peut faire perdre la position réelle du mobile à l'unité de traitement. Il faudra alors procéder à la réinitialisation du système automatisé.

Il est sensible aux parasites en ligne, un parasite peut être comptabilisé par le système de traitement comme une impulsion délivrée par le codeur.

II.1.4.2 Codeur absolu :

➤ **Avantage :**

Il est insensible aux coupures du réseau : la position du mobile est détenue dans une onde qui est envoyé en parallèle au système de traitement. L'information de position est donc disponible dès la mise sous tension.

Si le système de traitement « saute » une information de position délivrée par le codeur, la position réelle du mobile ne sera pas perdue car elle restera valide à la lecture suivante.

➤ **Inconvénients :**

Il est de conception électrique et mécanique plus complexe aussi son coût sera plus élevé qu'un codeur incrémental.

Les informations de position sont délivrées « en parallèle » ; son utilisation mobilisera donc un nombre important d'entrées du système de traitement.

II.1.5 Choix de codeur :

Un critère majeur pour le choix du codeur est la robustesse du système. En effet, certains codeurs sont montés directement sur le moteur ; ils doivent donc être insensible à la variation de température et aux vibrations afin de ne pas être endommagés. Le niveau de sensibilité du codeur joue aussi un rôle primordial.

II.2.1 Moteur à courant continu :

Beaucoup d'applications nécessitent un couple de démarrage élevé. Le Moteur à Courant Continu (MCC) possède une caractéristique couple/vitesse de pente importante, ce qui permet de vaincre un couple résistant élevé et d'absorber les à-coups de charge : la vitesse du moteur s'adapte à sa charge. Le moteur à courant continu est utilisé quand on dispose d'une source d'alimentation continue (batterie par exemple). Il se caractérise par des lois de fonctionnement linéaires qui rendent l'exploitation de ses caractéristiques faciles d'emploi [5].

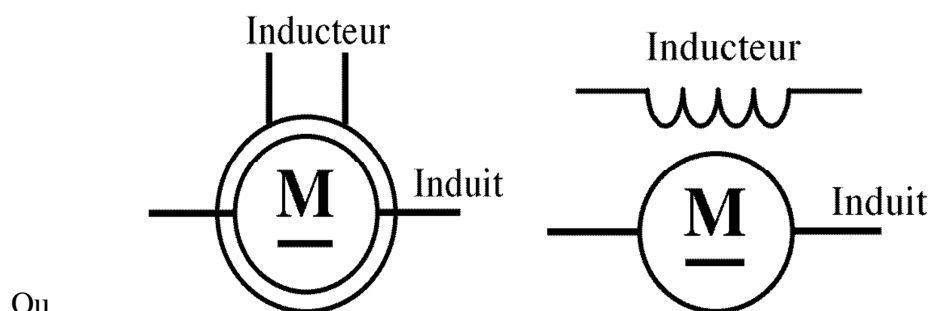


Figure 11 : Symbole de Moteur

II.2.2 CONSTITUTION :

La machine à courant continu est constituée de trois parties principales :

- L'inducteur
- L'induit
- Le dispositif collecteur /ballais

II.2.2.1 L'inducteur :

C'est un aimant ou un électroaimant (bobinage parcouru par un courant continu ' i ').

Il est situé sur la partie fixe de la machine (le stator) :

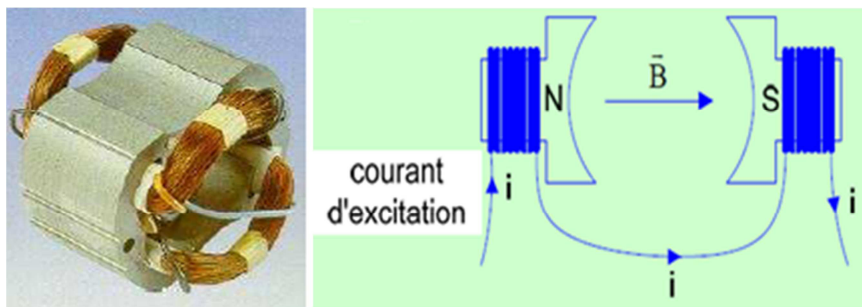


Figure12 : l'inducteur de moteur

Il sert à créer un champ magnétique (champ "inducteur") dans le rotor.

II.2.2.2 L'induit (circuit de puissance) :

L'induit est situé au rotor (partie tournante de la machine) :

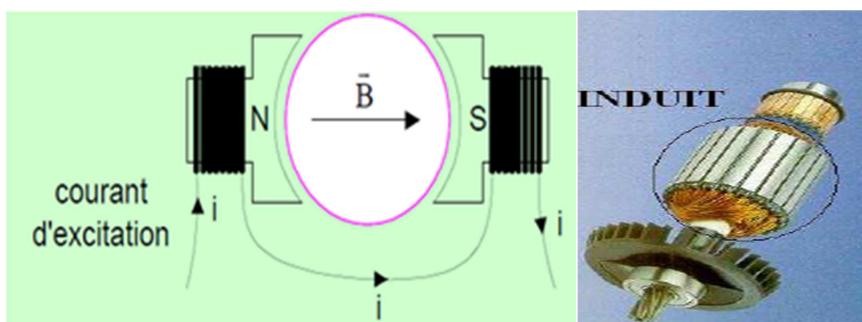


Figure13 : l'induit de moteur

C'est un bobinage parcouru par un courant continu I (courant d'induit).

II.2.2.3 Le collecteur et les balais :

Le collecteur est un ensemble de lames de cuivre où sont reliées les extrémités du bobinage de l'induit.

Les balais (ou charbons) sont situés au stator et frottent sur le collecteur en rotation.

Le dispositif collecteur / balais permet donc de faire circuler un courant dans l'induit.

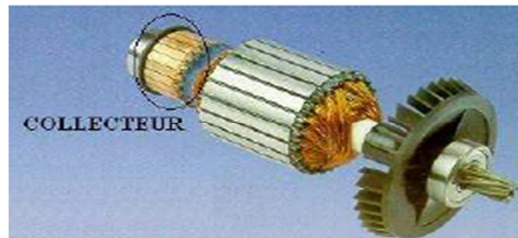


Figure14 : les balais collecteurs de moteur

II.2.3 PRINCIPE DE FONCTIONNEMENT :

L'inducteur (ou stator) crée un champ magnétique fixe \vec{B} . Ce stator peut être à « aimants permanents » ou constitué d'électro-aimants.

L'induit (ou rotor) porte des conducteurs parcourus par un courant continu (alimentation de moteur) ; ces spires soumises à des forces (forces dites « de Laplace »), entraînant la rotation du rotor.

Il en résulte une variation du flux du champ magnétique à travers chaque spire ; elle engendre une f.é.m. qui est « redressé » par l'ensemble {collecteur +ballais}

La valeur moyenne E de cette f.é.m. est proportionnelle à la vitesse angulaire de rotation du rotor, au flux maximal du champ magnétique crée par l'inducteur à travers une spire ($\Phi = B \times S$) et à une constante K qui dépend des caractéristiques de la conception du moteur (nombre de conducteurs, surface de chaque spire, nombre de paires de pôles,...) [5].

II.2.4 LES DEUX TYPES DE LA MACHINES :

La machine à courant continu est réversible c'est-à-dire qu'elle peut être soit en moteur soit en génératrice.

Le moteur est un convertisseur d'énergie électrique en énergie mécanique, la génératrice est le convertisseur inverse.

II.2.4.1 Fonctionnement en moteur :

Conversion d'énergie électrique en énergie mécanique :

Il s'agit d'un convertisseur électromécanique permettant la conversion bidirectionnelle d'énergie entre une installation électrique parcourue par un courant continu et un dispositif mécanique ; selon la source d'énergie.

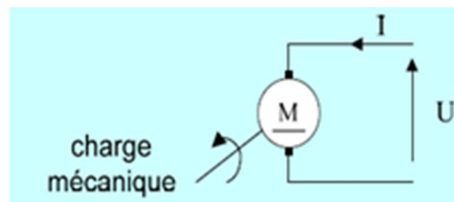


Figure 15 : convertisseur électromécanique

II.2.4.2 Fonctionnement en génératrice (dynamo) :

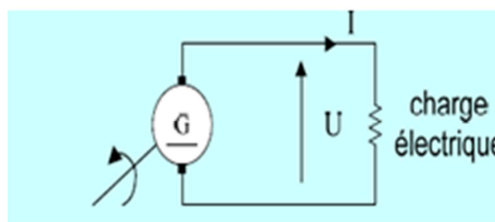


Figure16 : Fonctionnement en génératrice

II.2.5 Les différents types de la machine à courant continu :

Suivant l'application, les bobinages du l'inducteur et de l'induit peuvent être connectés de manière différente.

➤ **Machine à excitation indépendante :**

Comme dans le cas des génératrices à excitation indépendante, le bobinage inducteur des moteurs à excitation indépendante est raccordé à une alimentation à courant continu séparée. Par conséquent, le courant qui alimente l'inducteur est indépendant de celui qui alimente l'induit.

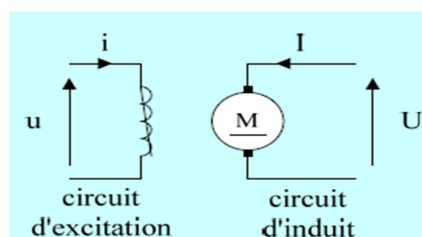


Figure 17 : Moteur à excitation indépendante

➤ **Machine à aimants permanents :**

L'excitation, dans ce cas, est créée par les aimants permanents. Le couple instantané (dans n'importe quelle configuration de machine à aimants) est la somme de trois couples. Les moteurs à aimants permanents peuvent accepter des courants de surcharge importants pour démarrer rapidement. Associés à des variateurs de vitesse électroniques

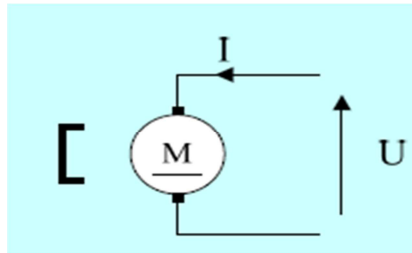


Figure 18 : Moteur à aimants permanents

➤ **Machine shunt (excitation en dérivation) :**

Dans le moteur shunt, le stator est câblé en parallèle avec le rotor. La tension aux bornes du rotor est la même que celle aux bornes du stator,

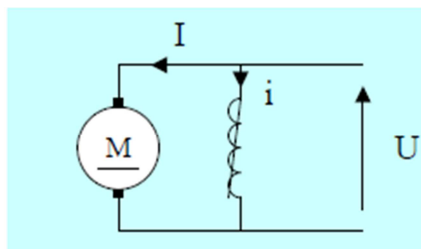


Figure 19 : Excitation shunt

➤ **Machines à excitation en série :**

L'excitation série étant principalement réservée à des moteurs, il n'est pas habituel d'utiliser le terme de machine à excitation série (exception faite dans les systèmes ferroviaires). Ce type de moteur est caractérisé par le fait que le stator (inducteur) est raccordé en série avec le rotor (induit).

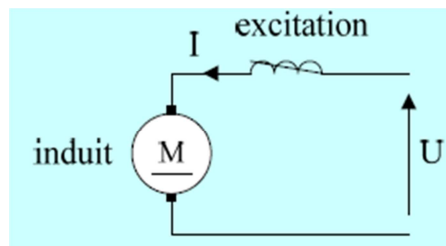


Figure 20: Moteur à excitation en série

➤ **Machine compound (excitation composée) :**

Dans le moteur compound une partie du stator est raccordé en série avec le rotor et un autre est de type parallèle ou shunt. Ce moteur réunit les avantages des deux types de moteur : le fort couple à basse vitesse du moteur série et l'absence d'emballement (survitesse) du moteur shunt [5].

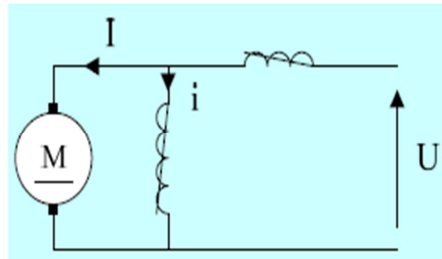


Figure 21 : excitation composée

II.2.5.1 Formules mathématique :

➤ **Expression de la f.e.m induite**

Loi de Faraday : $E = k \Phi B$

E : f.e.m induite (tension continue en V)

Φ : flux magnétique crée sous un pôle par l'inducteur (cf. fig. 1)

B : vitesse de rotation (en rad/s)

k : constante qui dépend de la machine considérée

➤ **Expression du couple électromagnétique**

Loi de Laplace : $T_{em} = k' \Phi I$

T_{em} : couple électromagnétique (en Nm)

I : courant d'induit (en A)

k' : constante qui dépend de la machine

➤ **Conversion de puissance**

La puissance électromagnétique P_{em} mise en jeu a deux formes :

• électrique $P_{em} = E I$

• mécanique $P_{em} = T_{em} \Omega$

Il vient : $E I = T_{em} \Omega$

$(k \Omega \Phi) I = (k' \Phi I) \Omega$

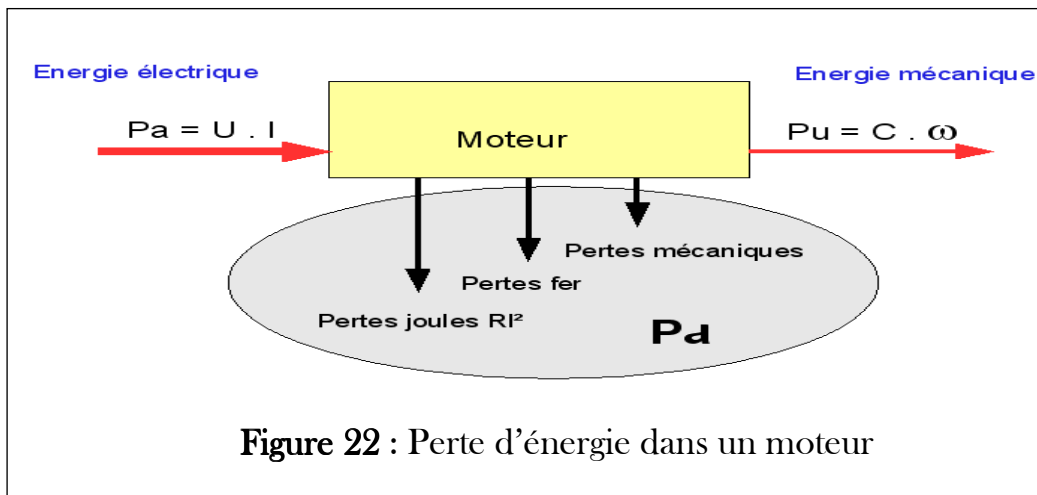
$K = k'$

En résumé :

$E = k \Omega \Phi$ $T_{em} = k I \Omega$

II.2.6 Puissance et rendement :

Les puissances intervenantes sur un système à moteur sont :



Puissance absorbée (Watt) : c'est la puissance électrique prélevée sur l'alimentation $P_a = U \cdot I$

Puissance utile (Watt) : c'est la puissance mécanique disponible sur l'arbre

$$P_u = C \cdot \omega$$

Puissance dissipée (Watt) : cette puissance correspond aux pertes électriques par effet joules ($R \cdot I^2$), aux pertes mécaniques et aux pertes magnétiques, La puissance utile est donc toujours plus faible que la puissance absorbée

$$P_d = P_a - P_u$$

Le rendement (sans unité) est le rapport entre la puissance utile et la puissance absorbée. Il est toujours inférieur à 1 ($\mu = P_u / P_a$).

II.2.7 Modélisation et Asservissement du moteur :

II.2.7.1 Modélisation du moteur :

Un moteur électrique à courant continu est régi par les équations physiques découlant de ses caractéristiques électriques, mécaniques et magnétiques. En utilisant le Théorème du moment cinétique et des équations d'électromagnétique, on obtient un système d'équations différentielles linéaires :

1-Équation électrique : $U(t) = E(t) + Ri(t) + \frac{Ldi(t)}{dt}$

2-Équation mécanique : $J\frac{d\Omega(t)}{dt} = r(t) - f\Omega(t)$

3-Équations électromécaniques : $E(t)=K\Omega(t)$

$r(t)=Ki(t)$

- U : Tension appliquée au moteur
- E : Force contre électromotrice
- i : Le courant de l'induit
- Ω : Vitesse de rotation de l'axe
- r : Couple moteur
- f : Couple de frottement
- J : Inertie du rotor

II.2.7.2 Asservissement du moteur :

A partir des équations électriques et mécaniques 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 où 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 ya dans la machine :

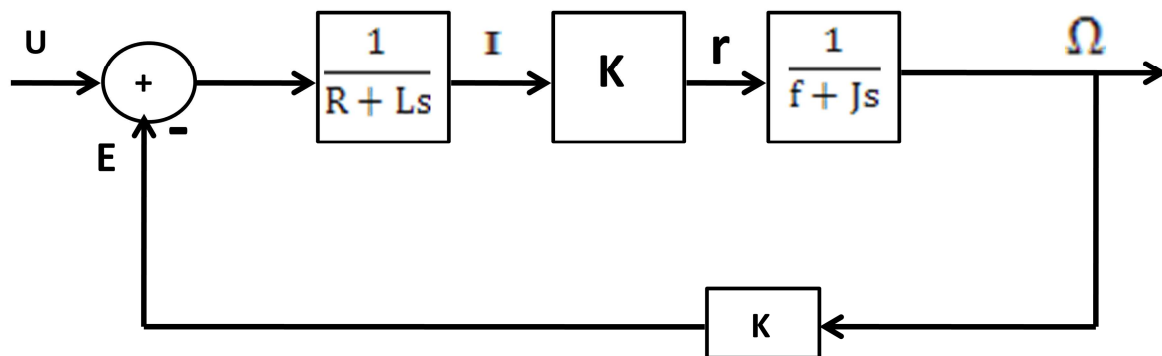


Figure 23 : Schéma bloc du moteur à courant continu (MCC)

On obtient : $\frac{\Omega(s)}{U(s)} = \frac{K}{LJs^2 + (RJ + Lf)s + Rf + K^2}$(II.1)

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.

II.2.7.2.1 Asservissement couple/courant :

Ce qui est sympa pour un moteur à courant continu c'est que on a pas besoin de faire un asservissement de couple, il est contrôlable très facilement : on a une relation linéaire entre le courant et le couple, naturellement il ya 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 tels que les MOSFET.....etc. C'est des composants qui sont relativement sensible à tout sur courant même très sensible, il suffit de dépasser une valeur nominale que ça entraine leur destruction.

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

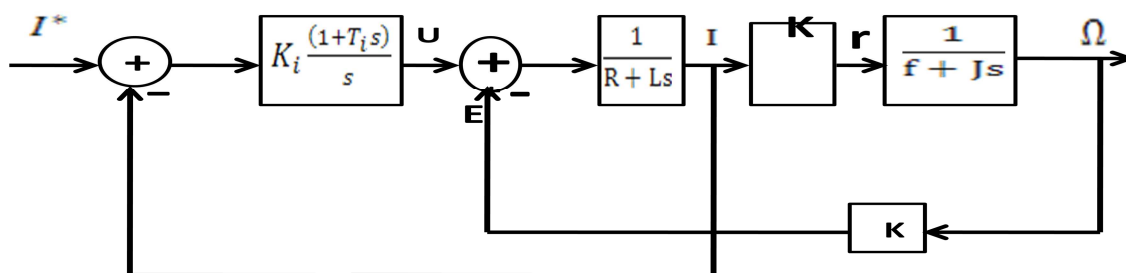


Figure 24 : Schéma bloc de l'asservissement du courant.

Tous les actionneurs de robotique 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 :

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

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

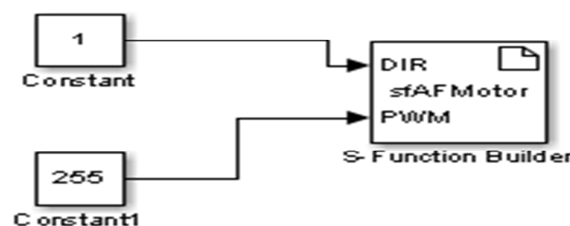
II.2.7.3 Technique de la PWM pour variation de vitesse :

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.

II.2.7.4 Pilotage du sens et vitesse du moteur :

Avant de s'intégrer dans la télé opération c'est-à-dire dans la commande de du moteurs.il faut d'abord commander (sens et vitesse) du moteur seul pour voir s'il fonctionne correctement ou non, ainsi pour visualiser les signaux du moteur. Pour ce faire nous avons intégré la librairie AFMotor de Arduino à MATLAB, toute en créant une nouvelle fonction sur SIMULINK qui nous permettra d'exécuter le programme de pilotage du sens et de vitesse de rotation du moteur comme suit :



Pour la direction (le sens) :

Entrées :	Déplacements :
1	Gauche
2	Droite
3	Relâcher

Pour la vitesse :

Entrées :	Rotation :
0	Vitesse nulle
255	Grande vitesse
[0..... 255]	Variation de vitesse

II.8 CONCLUSION :

Dans ce chapitre nous avons présenté les différents composants utilisés pour la réalisation de la maquette, ainsi présenter le principe de fonctionnement de tout actionneurs, capteurs utilisés dans la réalisation, à la fin nous avons donné un exemple sur MATLAB/SIMULIK et Arduino dont le but de comprendre la technique de la PWM pour la commande du moteur, tout en le testant indépendamment afin d'avoir une idée précise sur le fonctionnement du moteur et quels sont les moyens de le piloter.

III.1 INTRODUCTION :

Arduino est une plate-forme de prototypage d'objets interactifs à usage créatif constituée d'une carte électronique et d'un environnement de programmation.

Sans tout ne connaître ni tout comprendre de l'électronique, cet environnement matériel et logiciel permet à l'utilisateur de formuler ses projets par l'expérimentation directe avec l'aide de nombreuses ressources disponibles en ligne.

Pont tendu entre le monde réel et le monde numérique, Arduino permet d'étendre les capacités de relations humain/machine ou environnement/machine.

Arduino est un projet en source ouverte : la communauté importante d'utilisateurs et de concepteurs permet à chacun de trouver les réponses à ses questions.

III.2 Présentation Arduino :

Arduino est un projet électronique qui à vue le jour en Italie en 2005, Il a été créé et perfectionné par une équipe de développeurs, composée de six individus : (MassimBanzi, David Cuartielles, Tom Igoe, Gianluca Martino, David Mellis et Nicholas Zambetti). Cette équipe a créé ce qu'on appelle le "système Arduino". C'est un outil qui va permettre aux débutants, amateurs ou professionnels de créer des systèmes électroniques plus ou moins complexes. En outre ce système permet de faire le pont entre le monde virtuel de la programmation informatique (concepts formels et logiques) et le monde physique (interaction électromécanique des objets).

Arduino est donc une plate-forme open-source d'électronique programmées qui est basé sur une simple carte a microcontrôleur, et un logicielle, 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.

III.3 Avantage de l'Arduino :

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

- **Pas cher** : les cartes Arduino sont relativement peu coûteuses comparativement aux autres plateformes.
- **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.
- **Un environnement de programmation clair et simple** : l'environnement de programmation Arduino est facile à utiliser pour les débutants, tout en étant assez flexible pour que les utilisateurs avancés puissent en tirer profit également.
- **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, ATMEGA 328, etc. Les schémas des modules sont publiés sous une licence Creative Commons, et les concepteurs de circuits expérimentés peuvent réaliser leur propre version des cartes Arduino, en les complétant et en les améliorant.
 - 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 ou numériques.
 - Le microcontrôleur traitera ces informations et les transmettra vers les sorties numériques.
 - Récupère les données des capteurs pour les transmettre à l'interface de commande (pc) et traduire les instructions pour faire fonctionner les actionneurs.

III.4 Les outils de l'Arduino :

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

III.4.1 Architecture HARDWARE de la carte :

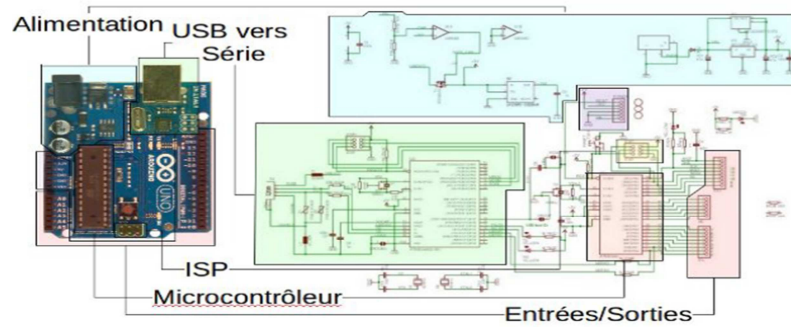


Figure 25 : Architecture interne de la carte Arduino méga 2560

III.4.2 Architecture de la carte :

La figure suivante illustre le schéma synoptique de la carte Arduino.

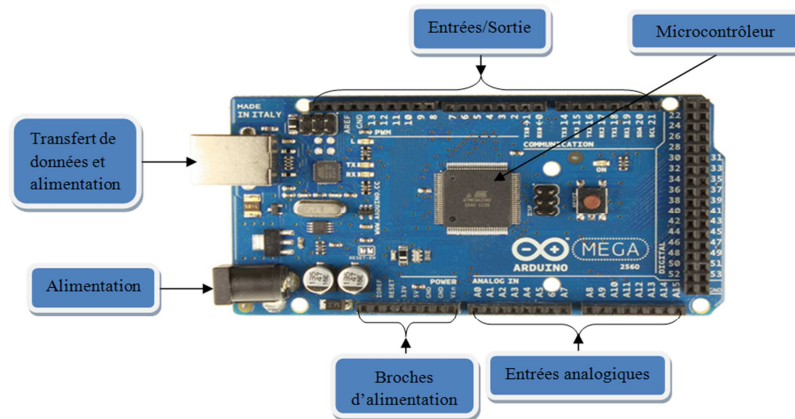


Figure 26 : Architecture de la carte Arduino méga 2560

Les éléments de base qui composent la carte Arduino sont :

III.4.3 Le microcontrôleur :

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 comme dans notre système).

III.4. 3.1 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-volatiles de programme et de données.
- Bytes 64K/128K/256K de flash Individu-Programmable de Dans-Système.
- Résistance : 10.000 écrire /cycles d'effacement.
- 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.

La figure ci-dessous montre la structure interne du microcontrôleur ATMEGA 2560

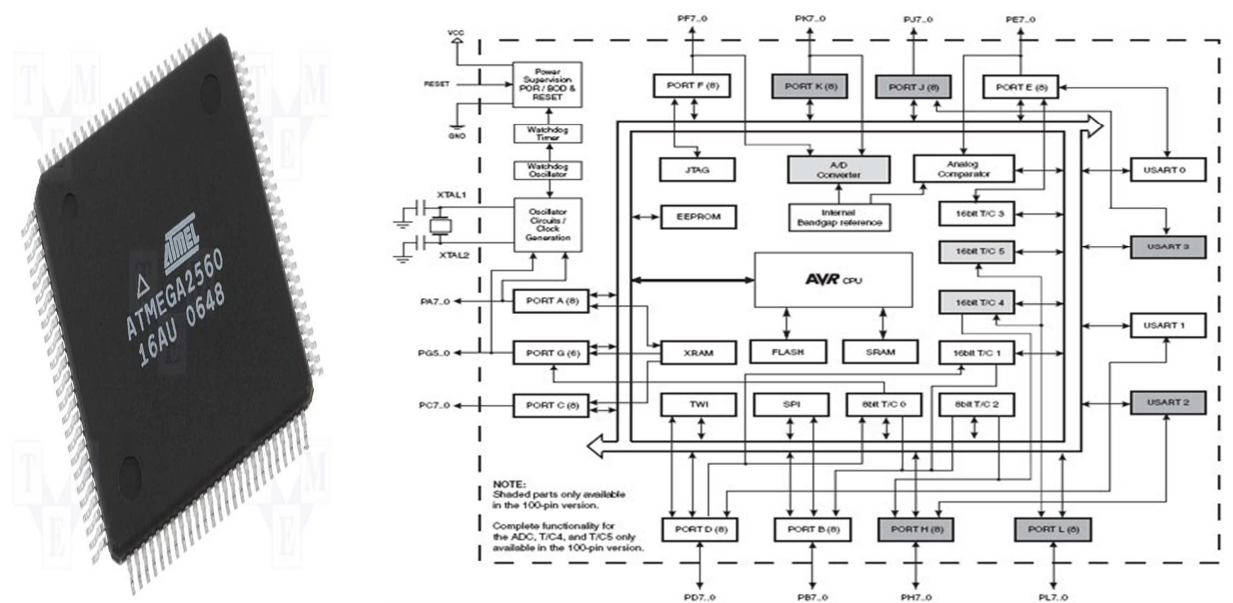


Figure 27 : Architecture du microcontrôleur ATMEGA 2560

III.4.4 Le entrées-sorties de la carte :

La carte Arduino possède 15 entrées analogiques, 54 entrées /sorties numériques dont les pins de (1 à 13) peuvent être utilisés comme des entrées/sorties analogiques.

Les entrées/sorties de l'Arduino sont les éléments avec lesquels nous seront le plus souvent en contact. En effet, c'est par leur intermédiaire que le microcontrôleur communique avec le monde extérieur.

- Les pins utilisés en numérique ont une sortie binaire soit (0 logique qui correspond à l'état bas du pin ou 0V) ou (1 logique qui correspond à l'état haut du pin ou 5V) ce qui amène le fonctionnement des pins on mode tout ou rien.
- Les pins utilisés en analogique ont une sortie de valeur (0 à 255)

III.4.5 Mémoire :

L'ATmega2560 qui équipe l'Arduino Mega 2560 dispose de trois types de mémoires :

1. La mémoire flash d'une capacité 256 Ko pour le programme, analogue à celle que l'on trouve par exemple dans les clés USB. Une fois programmée, cette mémoire Flash conserve son contenu indéfiniment, même lorsque la carte n'est pas alimentée.
2. La mémoire vive ou RAM, analogue dans son principe à la mémoire vive du PC, mais en plus petite quantité puisque l'on ne dispose ici de 8 Ko (contre plusieurs Go dans un PC par exemple). 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 devront être conservées lorsque la carte ne sera pas alimentée. Pour cela l'ATmega 2560 dispose également de mémoire EEPROM.
3. La mémoire EEPROM qui se voit doter de 4 Ko, dans lequel le microcontrôleur peut lire à tout instant. Il peut aussi y écrire et effacer son contenu avec plus de facilité.

III.4.6 L'alimentation :

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

- Une connexion USB de l'ordinateur qui fournit toujours directement du 5v (la tension de fonctionnement).
- 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).

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

III.4.7 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.

III.4.8 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 « **Shield Ethernet** » qui permet de connecter cette dernière à internet.

III.5 L'architecture SOFTWARE de la carte :

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 EDI (Espace de développement intégré)**.

III.5.1 Présentation de l'Espace de développement Intégré (EDI) Arduino :

Comme déjà dit La carte Arduino présente le noyau de notre système, elle est programmée par un logiciel compatible appelé **Arduino EDI (Espace de développement intégré)** qui est une application écrite en Java inspiré du langage Processing.

L'IDE permet d'écrire, de modifier un programme et de le convertir en une série de d'instructions compréhensible pour la carte.

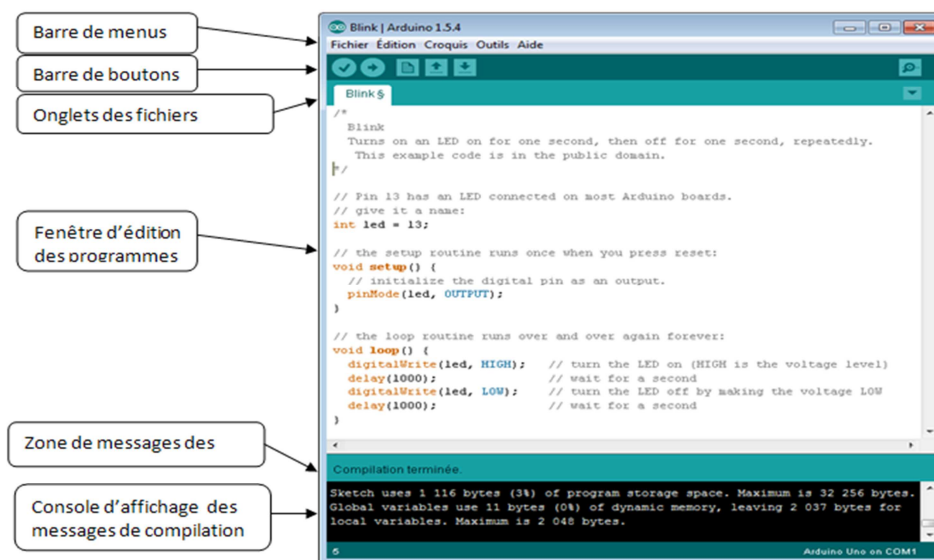
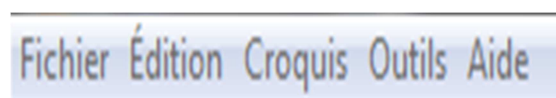


Figure 28 : les différentes parties de la fenêtre principale du logiciel Arduino.

III.5.2.1 Différentes composition de l'espace de développement intégré(EDI) :

Une barre de menus :



Cette barre contient les icones suivantes :

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

III.5.2.2 Une barre des boutons :



Vérifier/compiler : Vérifie le code à la recherche d'erreur.



Transférer vers la carte : Compile votre code et le transférer vers la carte Arduino.



Nouveau : Crée un nouveau code (ouvre une nouvelle fenêtre)



Ouvrir : Ouvre la liste de tous les programmes dans le "livre de programmes".



Enregistrer : Enregistre un programme.



Moniteur Série : Ouvre la fenêtre du moniteur (ou terminal) série.

III.5.2.3 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 messages :**

Comme son nom l'indique cette espace est utilisée par l'EDI 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).

III.5.2.4 Le programme Arduino :**III.5.2.4.1 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 :

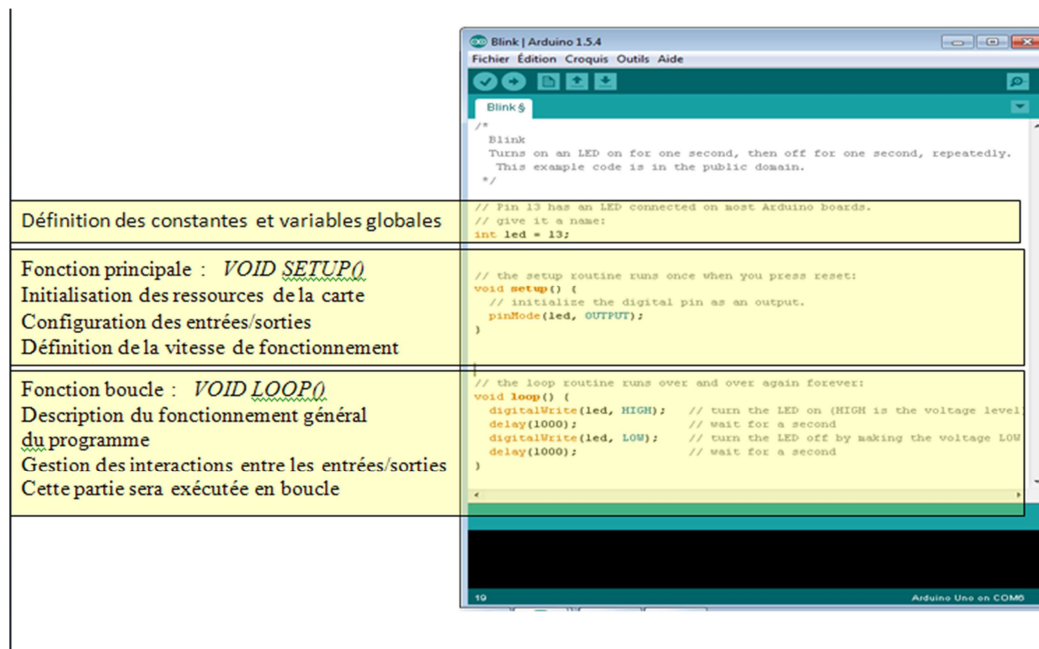


Figure 29 : les différentes composantes du programme Arduino.

Le code minimal :

Avec Arduino, nous devons utiliser un *code minimal* lorsque l'on crée un programme. Ce code permet de diviser le programme que nous allons créer en deux grosses parties, soit `voidsetup()` et `voidloop()` décrites en haut.

III.5.2.4.2 Le jeu d'instructions du langage Arduino :

1. 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** ».

2. Commandes du programme Arduino :**❖ Structure générale**

- `voidsetup()` (configuration-préparation)
- `voidloop()` (exécution)

❖ Contrôle et conditions

- `if` (si...)
- `if...else`(si...alors...)
- `for` (pour...)
- `switch case` (dans le cas où...)
- `while`(pendant que ...)

❖ Opérations de comparaison

- `==` (équivalent à)
- `!=` (différent de)
- `<`(Inférieur à)
- `>`(Supérieur à)
- `<=` (inférieur ou égal à)
- `>=` (supérieur ou égal à)

❖ Opérations booléennes

- `&&`(Et)
- `||` (ou)
- `!` (Et pas)

❖ Autres commandes

- `//` (Commentaire simple ligne)
- `/* */` (Commentaire multi-lignes)
- `#define`(donner une valeur à un nom)

❖ Variables

- **char** (variable 'caractère')
- **int** (variable 'nombre entier')
- **long** (variable 'nombre entier de très grande taille')
- **string** (variable 'chaîne de caractères')
- **array** (tableau de variables)

Variables

❖ Niveaux logiques des connecteurs numériques

- **HIGH** (état 1)
- **LOW** (état 0)
- **INPUT** (configuré en entrée)
- **OUTPUT** (configuré en sortie)

Fonctions

❖ Entrées-sorties numériques

- **pinMode** (**broche**, état) (configuration des broches en entrée ou en sortie)
- **digitalWrite**(**broche**, état) (écrire 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.)

❖ Entrées analogiques

- **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)

❖ Gestion du temps

- **unsigned long millis**() (temps de fonctionnement du programme)
- **delay**(ms) (attente, en millisecondes)
- **delayMicroseconds**(us) (attente, en microsecondes).

III.5.2.4.3 Les étapes à suivre pour programmer la carte :

1. L'écriture du programme :

Cette première étape consiste à saisir le programme.

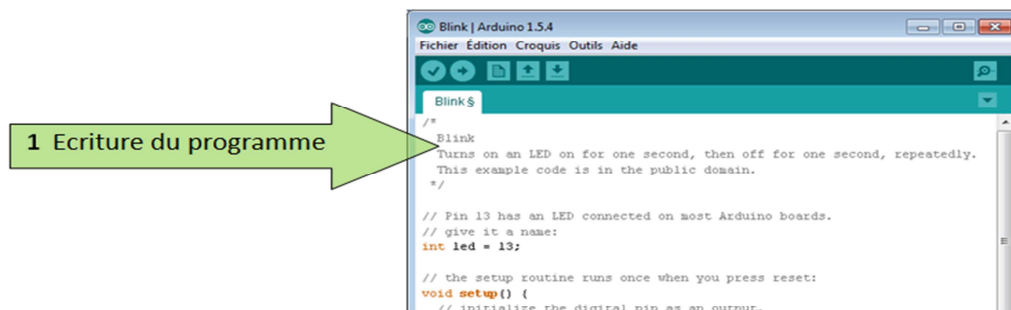


Figure 30 : l'écriture du programme

2. 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 renseigne sur le type d'erreur et la ligne où elle se trouve. (Pour la correction des erreurs il suffit de consulter l'aide disponible dans la barre de menus).

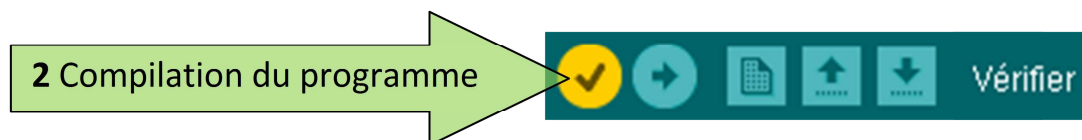


Figure 31 : compilation du programme

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

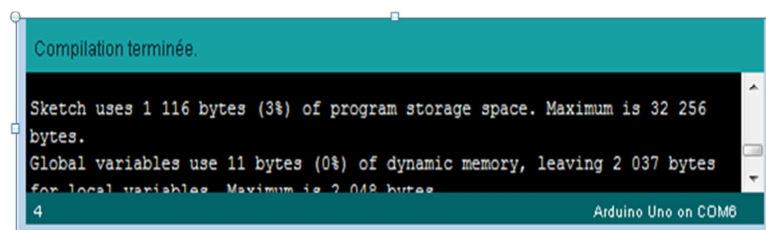


Figure 32 : Console d'affichage

3. Sélection de la cible (type de carte) :

Avant de transférer le programme vers la carte Arduino, il faut sélectionner le type 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.

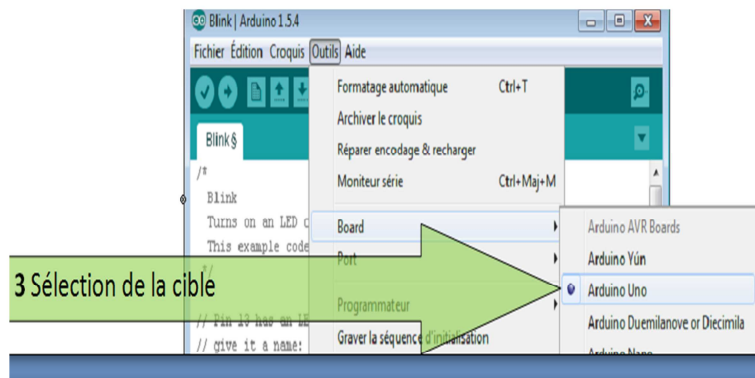


Figure 33 : Sélection de la cible

4. Sélection du port :

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

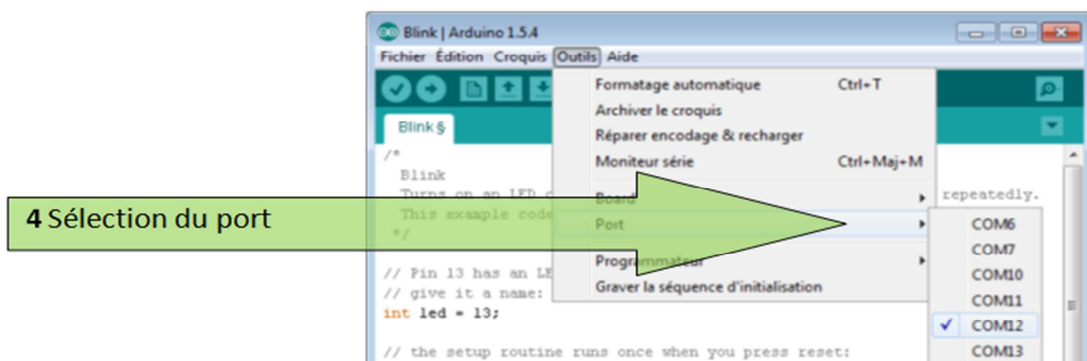


Figure 34 : Sélection du port

5. Transfert du programme :

Cela se fait en cliquant sur le bouton.

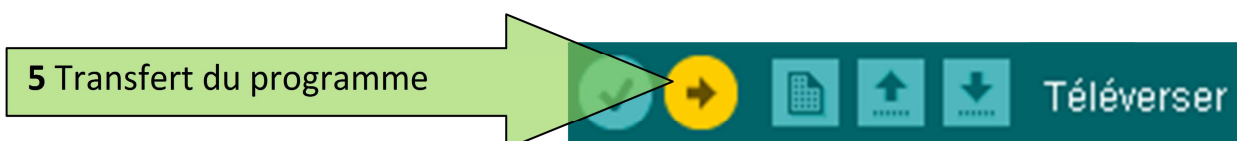


Figure 35 : Transfert du programme

Le programme est téléchargé vers la carte, le message « téléversement 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.

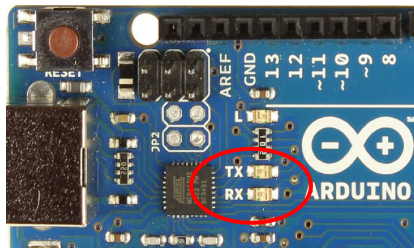


Figure36 : clignotement des deux LED du port série

III.6 CONCLUSION :

Dans ce chapitre, notre étude s'est focalisée sur la présentation de la carte (Arduino), tout en commençant par un bref historique et quelques détails concernant la création et l'utilité de cette carte. Puis nous avons parlé de son côté matériel, enfin présenter le logiciel et le langage de programmation

La compréhension de données exposées dans ce chapitre nous offre des bases de connaissances pour l'usage de la carte d'acquisition.

IV.1 Introduction :

Ce dernier chapitre concerne la phase finale dans la réalisation du prototype de ce projet.

Nous y abordons les difficultés que nous avons rencontrées. Nous y présentons également le matériel utilisé dans la réalisation en utilisant les datasheets et nous avons planifié la réalisation avec un schéma synoptique.

IV.2 Schémas synoptique de la carte d'essai mise en œuvre :

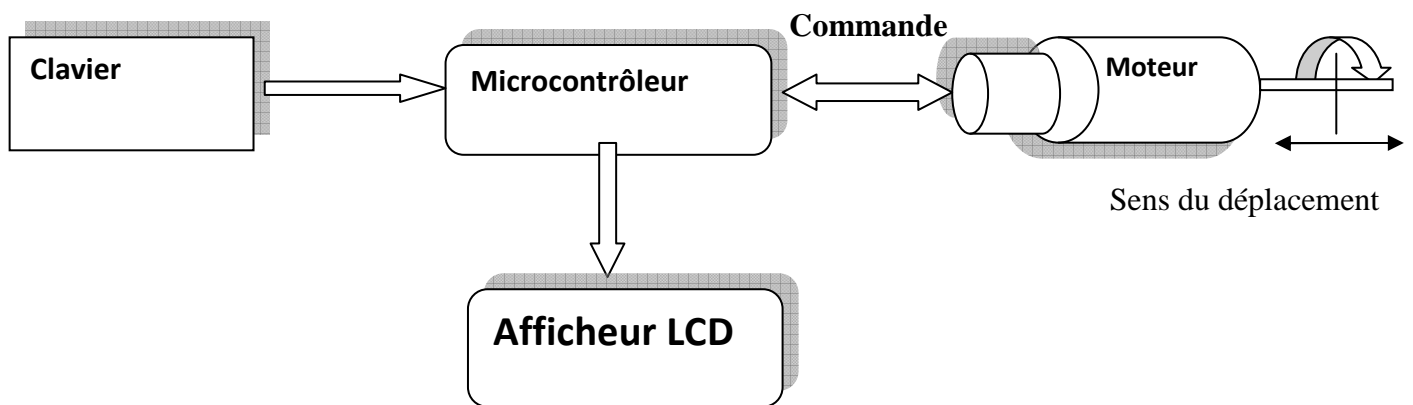


Figure 37 : Schéma global qui représente les blocs essentiels de la carte d'essai

IV.4 Le matériels utilisés :

IV.4.1.1 Le moteur utilisé et ses caractéristiques :

Le moteur utilisé dans notre réalisation est un moteur réducteur qu'on peut alimenter avec 12 V. Ce moteur possède un réducteur (rapport 30 :1) et un encodeur intégré, dont la consommation de 530 mA environ. (150 mA à vide). Son courant de blocage est de 2,5 A, et sa vitesse de rotation est de 170 t/min (216 t/min à vide), son couple est de 1,5 kg.cm. Et le nombre d'impulsion de son encodeur est 360 impulsions par tour Il comporte six fils de branchement, à savoir le fil :

- 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 38 : Moteur réducteur avec encodeur

IV.4.1.2 Partie mécanique :

Un encodeur optique incrémental à quadrature de phase est relié au rotor du moteur, ce qui nous permet de récupérer une information sur l'incrémentation angulaire du rotor, à partir de laquelle on peut calculer la vitesse de rotation du rotor et sa position angulaire par rapport à sa position initiale. Il s'agit d'un disque en rotation doté de deux rangées de fentes décalées. Un système optique détecte ces fentes et émet 360 impulsions par tour de rotor, sur deux signaux en quadrature de phase.

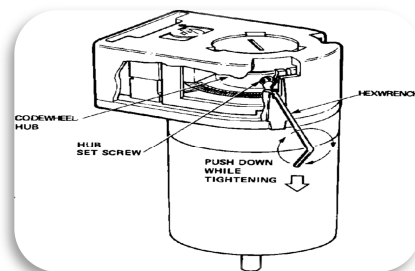


Figure39 : Utilisation d'un encodeur optique sur l'axe du moteur

IV.4.2.1 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. En raccordant les sorties de façon appropriées, il est possible de constituer deux pont-H. Il est ainsi possible de commander deux moteurs distincts, dans les deux sens et indépendamment l'un de l'autre.

Même si cette documentation n'utilise qu'un seul des deux pont-H pour la commande du moteur, il vous sera facile de faire le nécessaire pour commander un deuxième moteur tout aussi simplement.

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 (ce sont déjà des jolis petits moteurs). Le circuit peut fournir un maximum de 600mA par canal. Le L293D est aussi connu pour être un excellent Pont-H facile à mettre en pratique.

Chapitre IV Réalisation et Conception

Avec deux signaux de commande Input 1 et Input 2 fournis par Arduino (nous laissons volontairement Enable de côté pour le moment), 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'Input 1 et input 2 il devient possible de démarrer, Stopper ou Inverser le courant [7].

IV.4.2.2 Brochage :

Ci-dessous la configuration des broches du L293D et la table de la logique de commande.

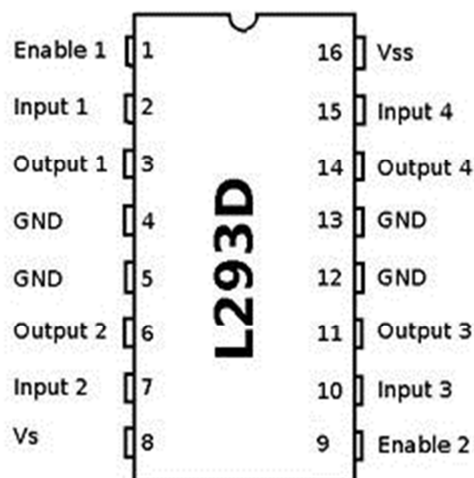


Figure 40 : Les bronches du L293D

Broche	Nom	Description
1	Enable 1	<p>Permet d'envoyer (ou pas) la tension sur les sorties du moteur via OUTPUT1 & OUTPUT2.</p> <p>ENABLE1 commande l'activation/désactivation du premier Pont-H.</p> <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 & INPUT2.
2	Input 1	Avec Input 2, sont les broches de commande du Pont-H Output1/Output2.

Chapitre IV Réalisation et Conception

		Se raccorde a Arduino, permet de commander le sens du courant entre Output 1 et Output 2.
3	Output 1	Avec Output 2, seront les broches à raccorder à la charge (le moteur).
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). Si vous n'avez qu'une source d'alimentation pour le tout, c'est forcément plus simple.
5	GND	
6	Output 2	Avec Output 1, seront les broches à raccorder à la charge (le moteur).
7	Input 2	Avec Input 1, sont les broches de commande du Pont-H Output1/Output2. Se raccorde a Arduino, permet de commander le sens du courant entre Output 1 et Output 2.
8	VS	Alimentation de puissance des moteurs. Par exemple, s'il s'agit d'une ancien véhicule téléguidé transformé, il s'agira de la borne positive de l'accumulateur (souvent +9.2v).
9	Enable 2	Commande l'activation du second pont-H constitué de Output3 /Output4
10	Input 3	A utiliser conjointement avec Input 4 pour commander le pont-H Output3/Output4.
11	Output 3	Constitue une des deux sorties du second pont-H (Output3/Output4)
12	GND	
13	GND	

Chapitre IV Réalisation et Conception

14	Output 4	Constitue une des deux sorties du second pont-H (Output3/Output4)
15	Input 4	A utiliser conjointement avec Input 3 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).

IV.4.2.3 la Logique de commande Input :

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

IV.5 Afficheur à cristaux liquides (LCD)

IV.5.1 Introduction :

Les afficheurs à cristaux liquides, appelés afficheurs LCD (Liquid Crystal Display), sont des modules compacts intelligents et nécessitent peu de composants externes pour un bon fonctionnement. Ils consomment relativement peu (de 1 à 5 mA).

Plusieurs afficheurs sont disponibles sur le marché et diffèrent les uns des autres, par leurs dimensions (de 1 à 4 lignes de 6 à 80 caractères), et aussi par leurs caractéristiques

Chapitre IV Réalisation et Conception

techniques et leur tension de service. Certains sont dotés d'un rétro-éclairage. Cette fonction fait appel à des LED montées derrière l'écran du module [8].

IV.5.2 Schéma fonctionnel :

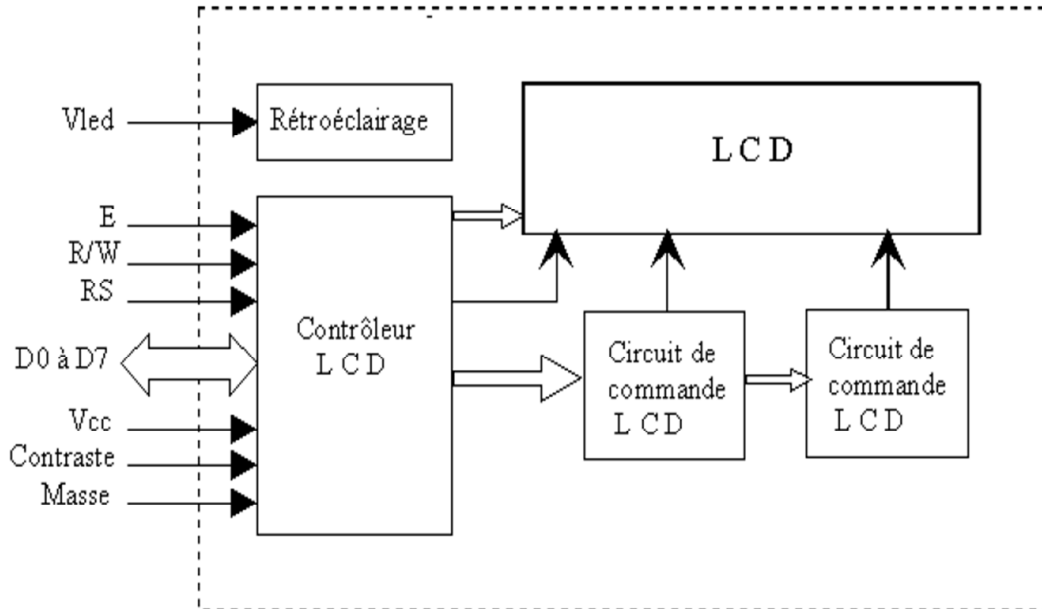


Figure 41 : schéma fonctionnel d'un LCD

Comme il le montre le schéma fonctionnel, l'affichage comporte d'autres composants que l'afficheur à cristaux liquides (LCD) seul. Un circuit intégré de commande spécialisé, le LCD-Controller, est chargé de la gestion du module. Le "contrôleur" remplit une double fonction : d'une part il commande l'affichage et de l'autre se charge de la communication avec l'extérieur.

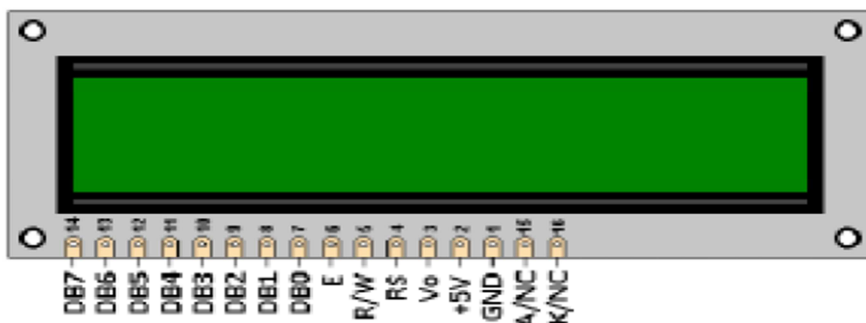


Figure 41 : Photo d'un LCD et son brochage

Les broches 15 et 16 ne sont présentes que sur les afficheurs LCD avec rétro-éclairage.

IV.5.2 Brochage :

Broche	Nom	Fonction
1	Vss	Masse
2	Vdd	Alimentation positive +5V
3	Vo	Cette tension permet, en la faisant varier entre 0 et +5V, le réglage du contraste de l'afficheur.
4	RS	Sélection du registre (Register Select) Grâce à cette broche, l'afficheur est capable de faire la différence entre une commande et une donnée. Un niveau bas indique une commande et un niveau haut indique une donnée.
5	R/W	Lecture ou écriture (Read/Write) L : Écriture/H : Lecture
6	E	Entrée de validation (Enable) active sur front descendant. Le niveau haut doit être maintenue pendant au moins 450 ns à l'état haut.
7	D0	
8	D1	
9	D2	
10	D3	D0 → D7 Bus de données bidirectionnel 3 états (haute impédance lorsque E=0)
11	D4	
12	D5	

Chapitre IV Réalisation et Conception

13	D6	
14	D7	
15	A	Anode rétro éclairage (+5V)
16	K	Cathode rétro éclairage (masse)

IV.6 Le clavier à matrice :

Pour ce projet, le type de clavier, nous allons utiliser un clavier de matrice. Ceci est un clavier qui suit un schéma de codage qui lui permet d'avoir beaucoup moins de broches de sortie qu'il y a de touches. Dans notre exemple le clavier que nous avons utilisé, dispose de 16 touches (0-9, AD, *, #), mais seulement 8 broches de sortie. Par contre Avec un clavier linéaire, il faudrait 17 broches de sortie (un pour chaque clé et une broche de terre) afin de travailler. Le système de codage de la matrice permet de moins de broches de sortie et donc beaucoup moins de connexions qui doivent être faites pour le clavier pour travailler. De cette façon, ils sont plus efficaces que les claviers linéaires, étant qu'ils ont moins de câblage.



Figure 43 : Clavier

Chapitre IV Réalisation et Conception

Simulation :

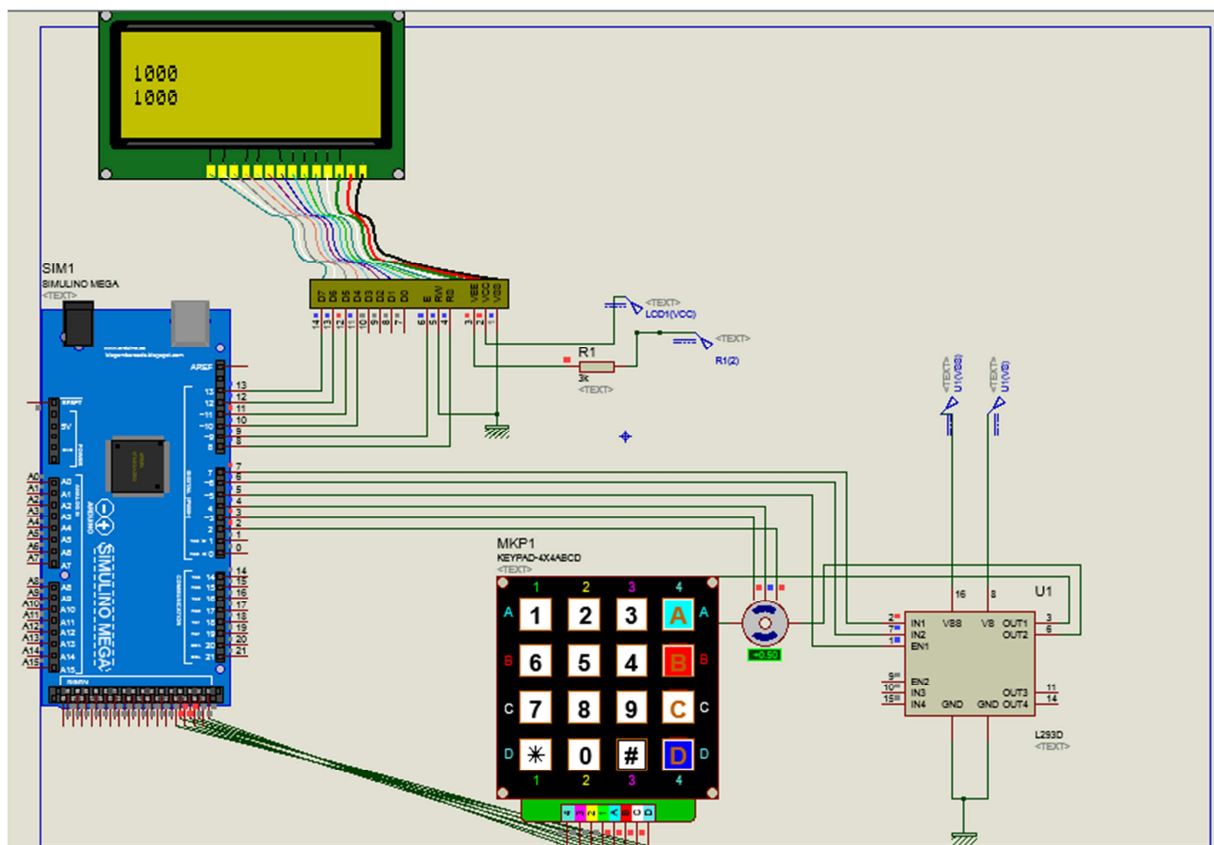
Avant de passer à la réalisation pratique de notre système nous avons eu recours à la simulation des différentes parties du système.

Pour cela on a utilisé le logiciel Proteus qui est un très bon logiciel de simulation en électronique.

Isis est un éditeur de schémas qui intègre un simulateur analogique, logique ou mixte. Toutes les opérations se passent dans cet environnement, aussi bien la configuration des différentes sources que le placement des sondes et le tracé des courbes.

La simulation permet d'ajuster et de modifier le circuit comme si on manipulait un montage réel. Ceci permet d'accélérer le prototypage et de réduire son coût.

Il faut toujours prendre en considération que les résultats obtenus de la simulation sont un peu différents de celles du monde réel.



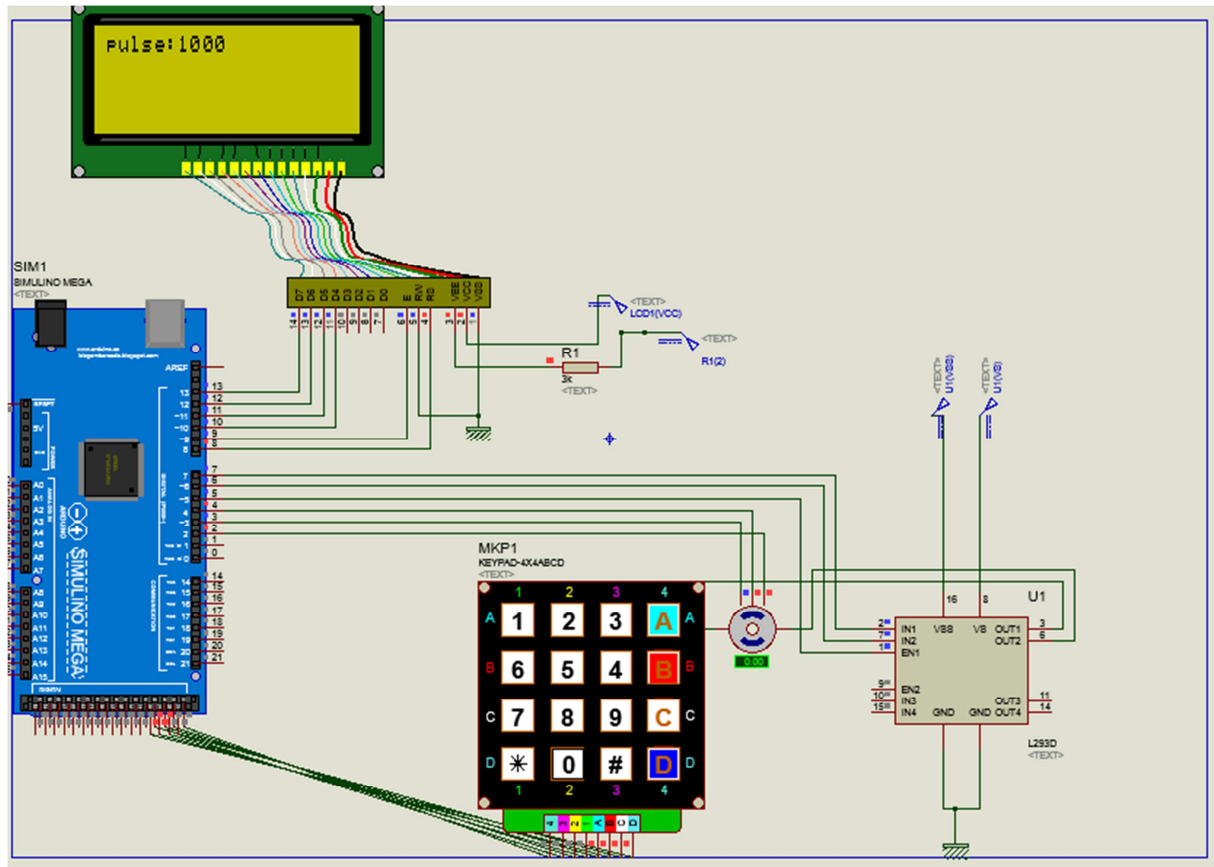


Figure 44 : schéma simulation

En utilisant un clavier digital, on a introduit les valeurs d'impulsion MLI (modulation en largeur d'impulsion) qui par la suite vont être transmises via le port serial vers la carte Arduino, qui a son tour va les transmettre au moteur cc (moteur à courant continu).

IV.7 Conclusion :

La réalisation de ce projet nous a énormément appris, autant au niveau de l'électronique, de la programmation des microcontrôleurs. Nous avons aussi appris comment interpréter les résultats de la simulation et la bonne maîtrise du logiciel Proteus.

Conclusion générale

Durant tout le temps qui nous a été alloué pour effectuer ce travail, nous avons essayé d'expliquer avec beaucoup de détail le fonctionnement d'une machine à commande numérique, avec tout ce que sa englobe comme étapes à étudier.

Nous avons d'abord entamé la partie mouvement qui est basée sur les moteurs à courant continu que nous avons étudié en détail dans le chapitre 2. Cette partie est essentielle au fonctionnement des machines à commande numériques.

Ensuite nous avons entamé une étude sur le processus de communication entre la carte de commande et l'ordinateur. Ceci a été fait en détail dans le chapitre 3. Aucune machine à commande numérique ne peut fonctionner sans un système qui lui envoie les données à transformer en mouvements. C'est pour cela qu'on a opté pour le protocole USB, qui est un mode de communication moderne et rapide.

Enfin, dans le dernier chapitre nous avons essayé de rassembler ce qui a été étudié dans les chapitres précédent. Nous avons vu les différents types de programmation effectués au niveau du microcontrôleur ainsi qu'au niveau de l'ordinateur et ceci pour former un ensemble homogène qui permettra le bon fonctionnement de notre réalisation.

Le travail qu'on a réalisé a bien fonctionné en simulation sous Proteus et Matlab mais les tests pratiques n'étaient pas concluants à cause du facteur temps.

Pour conclure, nous souhaiterions qu'il y'ait une continuation pour cette étude par d'autres étudiants afin de concevoir d'autres utilisations possibles pour cette carte.

Bibliographie

[1] <http://www.materiel-pour-bureau.fr/>

[2] <http://www.massicot.net/>

[3] <http://www.pro-massicots.com/>

[4] sectionmei.buisson.free.fr/cours/codeur-incremental.pdf

[5] nicole.cortial.net/revisions/continu.pdf

[6] www.arduino.cc

[7] www.engineersgarage.com/.../l293d-motor-driver-ic

[8] www.didel.com/kits/Af2x8c.pdf

[9] <http://nalhossri.free.fr/>

Résumé

Le premier chapitre est consacré pour la présentation des machines outil à commande numérique MOCN, un aperçu général sur la technologie des MOCN, citant la classification, l'architecture d'un massicot.

Dans le second chapitre nous avons présenté les codeurs optiques et les moteurs à courant continue et leur principe de fonctionnement.

Le troisième chapitre est consacré à la présentation de la carte Arduino et son logiciel ainsi que les différentes étapes pour le chargement du programme dans la carte.

Dans le quatrième chapitre nous avons présenté le matériel utilisé dans la réalisation avec la simulation des programmes en utilisant le logiciel Protues.

Pour terminer, une mise en pratique de notre travail a été réalisée, et une conclusion générale a été tirée.

Mots clés :

Massicot

Arduino

Moteur à courant continu

Encodeurs

Simulateur Proteus