

وزارة التعليم العالي والبحث العلمي

BADJI MOKHTAR- ANNABA UNIVERSITY  
UNIVERSITE BADJI MOKHTAR ANNABA



جامعة باجي مختار- عنابة

Année : 2017

Faculté: Sciences de l'Ingénierat  
Département: Electronique

MEMOIRE

Présenté en vue de l'obtention du diplôme de : MASTER

**Commande en position du a MCC par Arduino**

Domaine : Sciences et Technologie

Filière : Génie électrique

Spécialité: Instrumentation et contrôle industriel

Par :GUENAOUA LILIA

DEVANT Le JURY

Président : GUERSI Noureddine Pr

Université Badji Moktar Annaba

Directeur de mémoire: SAADI Med Nacer MCB

Université Badji Moktar Annaba

Examineur : BENOUARETH Mohamed Pr

Université Badji Moktar Annaba

Examineur :KHERFANE Hamid MCA

Université Badji Moktar Annaba

## *Remerciement*

*Avant tout nous avons à remercier le bon Dieu, les tous  
puissants et nous disons :*

*« الحمد لله الذي هدانا لهذا و ما كنا لنهتدي لولا ان هدانا الله »*

*Mes remerciements s'adressent ensuite aux membres du jury  
qui me fait l'honneur de participer à ma soutenance.*

*C'est un plaisir de remercier Notre encadreur Mr : Mohamed  
Nacer Saadi pour le choix de ce sujet et aussi pour encadrement  
et ses précieux conseils.*

*Je remercie sincèrement Mr: Mohamed Benouarth pour Aidez-  
moi dans la pratique.*

*Nos remerciements les plus vifs reviennent aux étudiants du  
département d'électronique sans exception.*

*Je ne pouvais pas finir mes mots sans penser à ma famille parce  
qu'ils sont la raison pour laquelle j'atteindre mes objectifs*

*Un grand merci mes parents et mon frère RIAD et sa femme  
pour leur soutien indéterminé.*





# SOMMAIRE

I-Introduction général.....	3
<b>Chapitre I : moteur à courant continu</b>	
I.1.Introduction .....	4
I.2. Description d'un moteur à courant continu .....	4
I.3. Modélisation d'un moteur à courant continu.....	5
I.3. 1.Fonction de transfert du moteur.....	5
I.4. Les différents modes d'excitations.....	7
I.4. 1. Moteur à excitation séparée.....	8
I.4. 2. Moteur à excitation shunt (Dérivée).....	8
I.4. 3. Moteur à excitation série.....	8
I.4. 4. Moteur à excitation compound (Composée).....	9
I.5. Principe général de la régulation .....	9
I.6. Système en boucle ouverte.....	9
I.7. Système en boucle fermée .....	10
I.8. Représentation des systèmes linéaires.....	10
I.9. Précision et Stabilité des systèmes asservis.....	10
I.9.1. Précision.....	10
I.9.2. Stabilité.....	11
I.10. Influence des perturbations.....	11
I.11. Compensation des systèmes asservis.....	12
I.12 La commande PID.....	12
I.12. 1.La commande PID en régulation de vitesse et de position .....	13
I.13. Conclusion.....	15
<b>Chapitre II : Le dispositif programmable Arduino</b>	
II.1. Introduction.....	16
II.2. Définition du module Arduino.....	16
II.3. Les gammes de la carte Arduino.....	16
II.4 .Avantage de la carte Arduino UNO.....	18
II. 5. La constitution de la carte Arduino UNO.....	19
II.5.1. Partie matérielle.....	19
II.5.1.1 .Le Microcontrôleur ATmega328.....	19
II.5.1.2. Les sources de l'alimentation de la carte.....	20
II.5.1.3 .Les entrées et sorties.....	20
II.5.2. Partie programme.....	22
II.5.2.1. l'environnement de la programmation.....	22
II.5.2.2. Structure générale du programme (IDE Arduino).....	22
II.5.2.3 .Injection du programme.....	23
II.5.2.4. Description du programme.....	23
II.5.2.5 .Les étapes de téléchargement du programme.....	24
II.6 .Les Accessoires de la carte Arduino.....	25
H BridgeL298.....	25
Caractéristiques.....	26
Afficheur à cristaux liquides (LCD) .....	26
Schéma fonctionnel .....	27
Brochage .....	28
Clavier numérique électronique.....	29
Encodeur optique.....	29

Caractéristiques.....	29
Schéma des connexions.....	30
II.7 Conclusion.....	31
<b>Chapitre III : Réalisation d'un dispositif expérimental</b>	
III.1. Introduction.....	32
III.2. But .....	32
III.3. Schéma synoptique général.....	32
III.4. Composants utilisés.....	33
III.4.1. Bloc d'alimentation.....	33
III.4.2. bloc de traitement et commande.....	34
III.4.3 .Bloc de puissance.....	34
III.4.4. Bloc de mesure .....	35
III.5. Schéma du circuit électronique.....	36
III.6. La réalisation virtuelle « PORTEUS » .....	36
III.7. Présentation de l'organigramme. ....	38
III.8. Conclusion.....	40
Conclusion générale	
Bibliographie .	

## Résumé

Ce projet nous permet de plonger dans le monde d'interfaçage afin d'arriver à réaliser une carte électronique qui communique entre (le moteur à cc et la carte Arduino ainsi que les actionneurs)

L'objectif de notre travail est de concevoir la commande de position d'un moteur à courant continu. Pour faire on synthèse un correcteur PID continu, qui vérifié les performances désirés.

Dans ce travail en commençant par la commande PID continu jusqu'à l'implémentation d'un commande PID discrète pour le moteur à courant continu avec charge élastique.

## Abstract

This project allows us to delve into the world of interfacing in order to arrive to create an electronic card that communicates between (motor DC and the Adriano board and the actuators).

The objective of our work is design position control of a continuous current motor. To does this synthesizing a continuous PID controller, which verified the desired performance.

In this work, starting with the continuous PID control until the implementation of a discrete PID control for DC motor with load.

## المخلص

يسمح لنا هذا المشروع بالغوص في عالم التواصل من اجل الوصول الى انشاء بطاقة الكترونية تربط بين محرك التيار

عملنا التحكم في السرعة و السيطرة على المستمر و بطاقة الاردوينو و المنفدات  
الموقف من محرك التيار المستمر لهذا نقوم بتركيب وحدة تحكم الهدف من مستمر و التحقق من الاداء المطلوب

في هذا العمل نبدا بوحدة تحكم مستمر حتى وحدة التحكم المنفصلة لمحرك التيار المستمر

# Introduction générale

*La technologie moderne a permis le développement des sciences tout en imposant l'exploration de domaines théoriques de plus en plus complexes. Parmi ces sciences en pleine expansion et intégrant rapidement l'apport des technologies modernes, on compte l'automatique.*

*Dans la plupart des processus industriels, en particulier les moteurs électriques, il est indispensable de maîtriser certains paramètres physiques (vitesse, position, angle...etc.), il est donc très souvent nécessaire d'avoir recours à une commande.*

*Dans ces projets objectifs ont été visés :*

- *Le premier est de regrouper suffisamment d'informations sur une grand catégorie de cartes d'interfaçage (Arduino) : son langage de programmation, sa construction, son principe de fonctionnement.*
- *Le deuxième consiste à réaliser une carte électrique capable d'exécuter une action entre le moteur à courant continu et une carte d'interfaçage (Arduino) en expliquant les différents bloquent de sa construction.*

*La commande PID (Proportionnelle-Intégrale-Dérivée) est une méthode qui a fait ses preuves et qui donne de bons résultats, grâce à l'action proportionnelle qui améliore la rapidité, l'intégrale pour la précision, et la dérivée pour la stabilité.*

*L'objectif de notre travail est d'implémenter une commande PID pour un moteur à courant continu. La modélisation de la commande et du système (moteur MCC) est programmée sous l'environnement Arduino IDE*

*Notre projet de fin d'étude comporte 3 chapitres, dont le premier parle sur les généralités sur le moteur à courant continu la seconde sur le dispositif programmable Arduino, et le dernier sur la réalisation d'un dispositif expérimental.*

## I.1.Introduction

L'évolution des technologies conduit à utiliser des machines nécessitant des vitesses de rotation précises et variables pour l'entraînement d'engins industriels systèmes robotique...etc.

Dans notre étude, on s'intéresse aux moteurs à courant continu car ils offrent une introduction facile au fonctionnement de ses homologues, en donnant des repères clairs. Les moteurs à courant continu ont pendant longtemps été les seuls aptes à la vitesse variable à large bande passante, comme dans le domaine robotique. Ils ont donc fait l'objet de nombreuses améliorations, et beaucoup de produits commercialisés aujourd'hui.

## I.2. Description d'un moteur à courant continu

Un moteur à courant continu est une machine électrique. 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, d'où l'énergie électrique est transformée en énergie mécanique. Un moteur électrique à courant continu est constitué :

- D'un stator qui est à l'origine de la circulation d'un flux magnétique longitudinal fixe créé soit par des enroulements statoriques (bobinage) soit par des aimants permanents à l'arrière du stator, se trouve la partie porte balais et les balais assurant les contacts électriques avec le rotor. Il est aussi appelé inducteur.
- D'un rotor bobiné relié à un collecteur rotatif inversant la polarité dans chaque enroulement rotorique au moins une fois par tour de façon à faire circuler un flux magnétique transversal en quadrature avec le flux statorique. Les enroulements rotoriques sont aussi appelés enroulements d'induits, ou communément induit, [4]

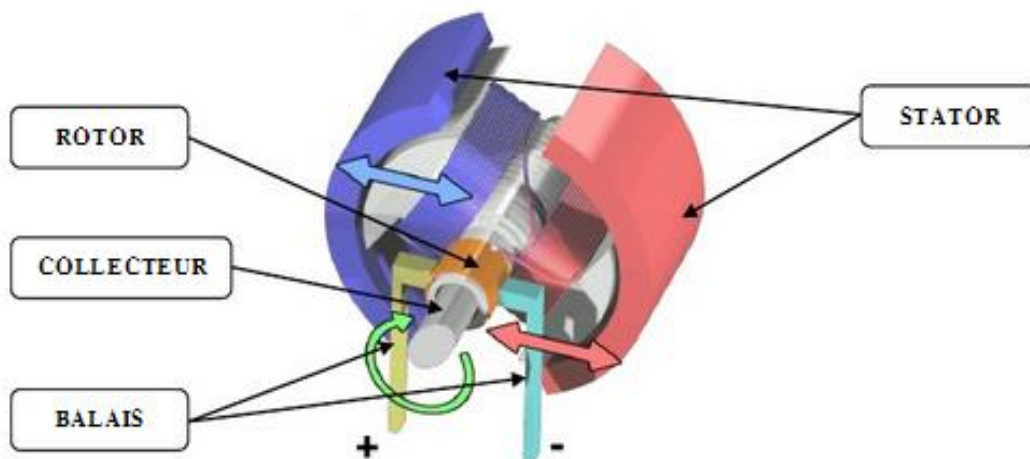


Figure. I.1: Schéma d'un moteur à courant continu



### I.3. Modélisation d'un moteur à courant continu

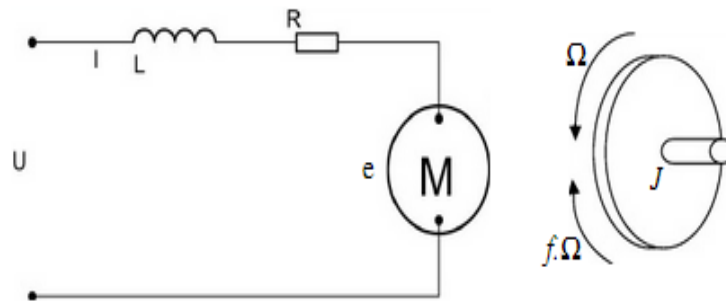


Figure. I.2: Schéma équivalent d'un moteur à courant continu

Selon le schéma de la (Figure. I.2) un moteur électrique à courant continu est régi par les équations physiques découlant de ses caractéristiques électriques, mécaniques et magnétiques. D'après la loi de Newton, combiné à des lois de Kirchhoff, On peut écrire les équations différentielles de premiers ordres suivantes :

$$u(t) = Ri(t) + L \frac{di(t)}{dt} + e(t) \quad (1)$$

$$e(t) = K_e \Omega(t) \quad (2)$$

D'après le principe fondamental de la dynamique on a :

$$\begin{aligned} J \frac{d\Omega}{dt} &= C_u - C_r \\ C_u &= K_c i(t) - C_p \\ C_r &= f \Omega(t) \end{aligned} \quad (3)$$

$u(t)$  : Tension appliquée au moteur

$e(t)$  : Force contre électromotrice

$i(t)$  : Intensité traversant le moteur

$\Omega(t)$  : Vitesse de rotation du rotor

$C_u$ : Couple moteur généré C

$r$  : Couple résistant

$f$  : Coefficient de frottement visqueux

$J$  : Moment d'inertie de l'axe du rotor

$K_e$ : Constante de vitesse

$K_c$ : Constante de couple

$C_p$ : Couple de pertes

#### I.3. 1.Fonction de transfert du moteur

On passe en Laplace

$$Ri(p) + Li(p) + E = U(p) \quad (4)$$

$$E = K_e \Omega(p) \quad (5)$$

$$Jp\Omega(p) = C_u - C_r \quad (6)$$

En combinant(4) et (5) on obtient :

$$U(p) - Ri(p) + Lpi(p) + K_e\Omega(p)$$

En modifiant (6) on a :

$$Jp\Omega(p) = Kc i(p) - C_p - f\Omega(p)$$

On en déduit l'expression de  $\Omega(p)$

$$\Omega(p) = \frac{Kc i(p) - C_p}{f + Jp}$$

On peut en sortir l'expression de  $i(p)$  :

$$I(p) = \frac{f + Jp}{Kc} \left( \Omega(p) + \frac{C_p}{f + Jp} \right)$$

On l'injecte à présent dans (4)

$$U(p) = \Omega(p) \left( \frac{(R+LP)(f+JP)}{Kc} + Ke \right) + \frac{R+LP}{f+JP} C_p$$

On suppose que le moment du couple de pertes (qui est vu comme une perturbation) est négligeable devant le moment du couple électromagnétique ( $Kc i(t)$ ) on peut alors prendre  $C_p$  nul pour simplifier le système

$$\text{On a donc : } U(p) - \Omega(p) \left( \frac{(R+LP)(f+JP)}{Kc} + Ke \right)$$

La fonction de transfert cherchée  $H(p)$  est entre la tension entrant dans le moteur  $U(p)$  et la vitesse de sortie  $\Omega(p)$ . [5], [6]

$$H(p) = \frac{\Omega(p)}{U(p)} = \frac{Kc}{(R+LP)(f+JP) + KcKe} \quad (7)$$

On peut établir le modèle mathématique de la réponse en vitesse du moteur électrique qui est donné par la figure suivante (figure. I.3)

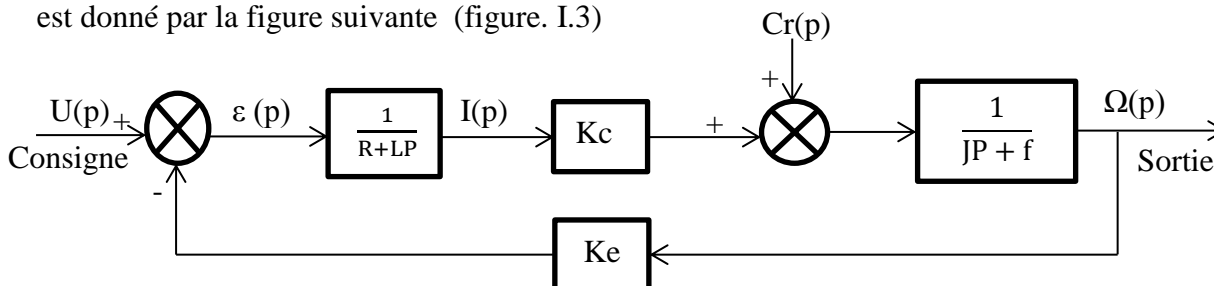


Figure. I.3: Schéma bloc du Modèle de moteur électrique en vitesse

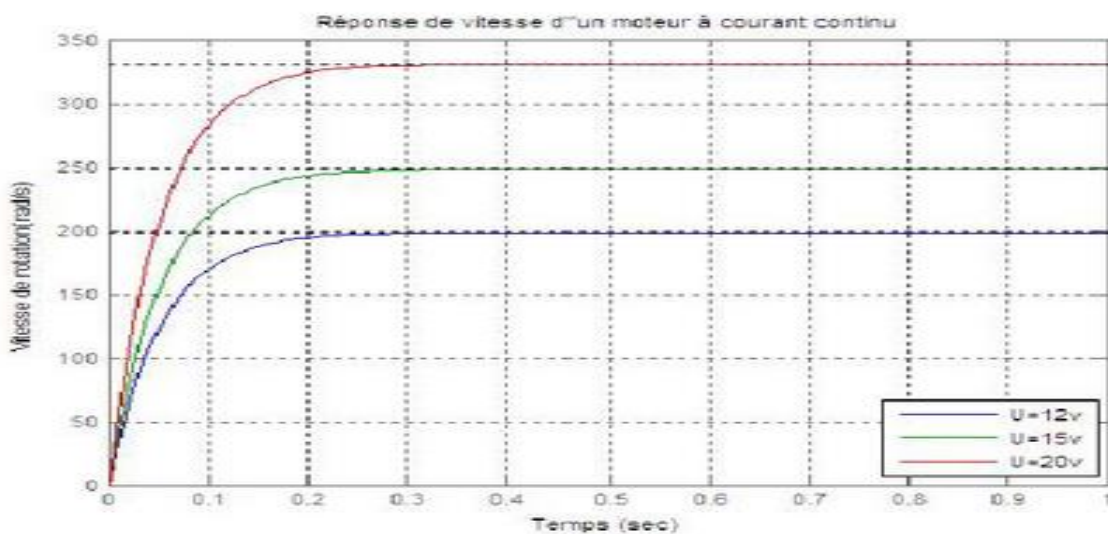


Figure. I.4: Modélisation sous Matlab de la réponse de vitesse

D'après l'équation (7) le système modélisé soit du second ordre, lorsque l'inductance interne est négligeable devant la résistance interne (ce qui est généralement le cas) il s'apparente à un système du premier ordre. On observe bien sur la (figure. I.4) ci-dessus que le moteur change de vitesse de rotation pour chaque valeur de la tension d'entrée, d'où la vitesse de rotation est proportionnelle à la tension d'entrée.

On peut établir le modèle mathématique de la réponse en position du moteur électrique qui est donné par la figure suivante (figure. I.5):

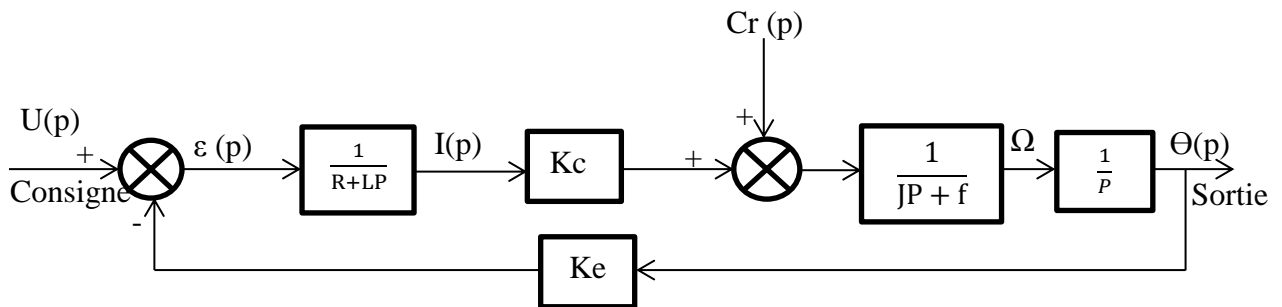


Figure. I.5: Schéma bloc du Modèle de moteur électrique en position

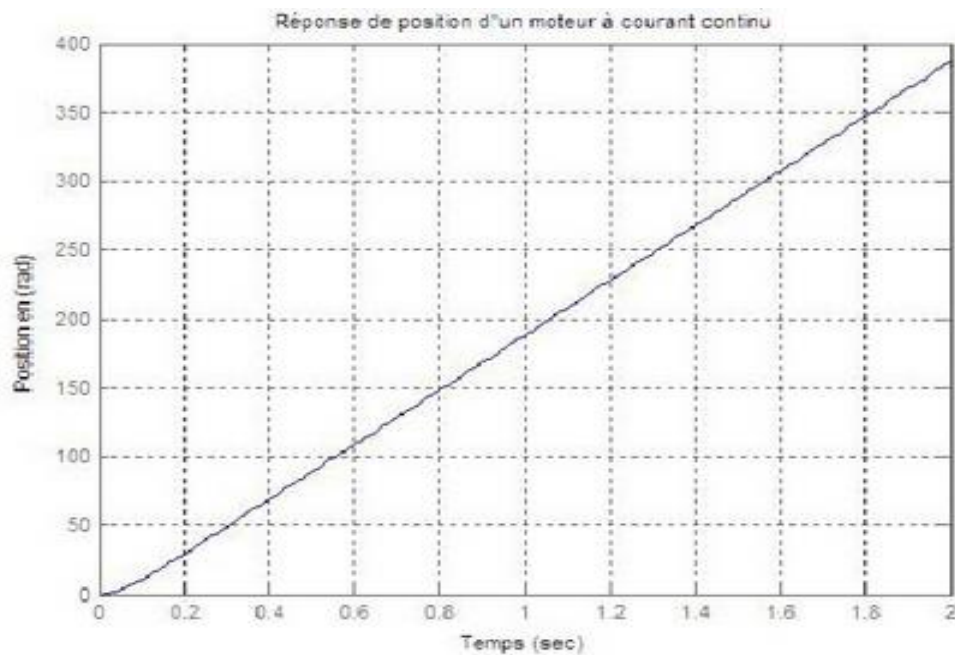


Figure. I.6: Modélisation sous Matlab de la réponse de position

D'après la (figure. I.6) on voit bien que la réponse en position est une rampe, ce qui signifie que le système dans un asservissement en position est instable, d'où il nécessite une régulation pour stabiliser la réponse en position et d'améliorer les performances en vitesse

#### I.4. Les différents modes d'excitations

Les moteurs à courant continu se différencient par la manière dont on fournit le courant d'excitation. Les différents cas possibles sont, [7].

### I.4. 1. Moteur à excitation séparée

Ce mode d'excitation nécessite deux sources d'alimentations distinctes. L'alimentation de l'enroulement inducteur est prise sur une source indépendante de la source principale. On change le sens de rotation en permutant les bornes de l'induit ou de l'inducteur. Le circuit électrique est représenté par la suivante :

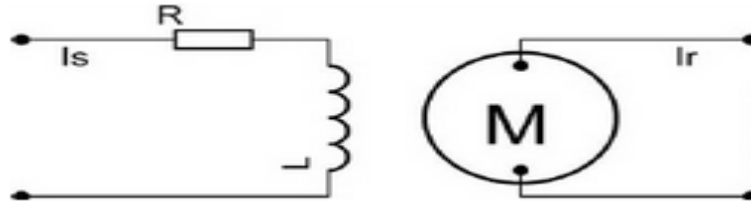


Figure. I.7: Modélisation électrique d'un Moteur à excitation séparée

### I.4. 2. Moteur à excitation shunt (Dérivée)

L'enroulement d'excitation est connecté en parallèle sur l'alimentation du moteur, il possède les mêmes propriétés que le moteur à excitation séparée du fait que, dans les deux cas, l'inducteur constitue un circuit extérieur à celui de l'induit. Le circuit électrique est représenté par la suivante :

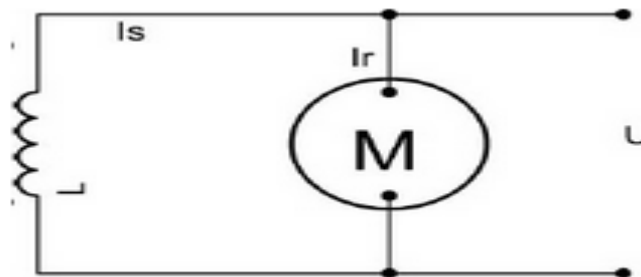


Figure. I.8: Modélisation électrique d'un Moteur à excitation shunt

### I.4. 3. Moteur à excitation série

Le circuit d'excitation est placé avec l'induit du moteur. Sa particularité est d'avoir un inducteur qui est traversé par le même courant, l'inducteur possède donc une résistance plus faible que celle des autres types de machines. L'inducteur est en série avec l'induit

- ❖ une seule source d'alimentation suffit.
- ❖ On change le sens de rotation en permutant les connexions de l'induit et de l'inducteur.

Le circuit électrique est représenté par la suivante :

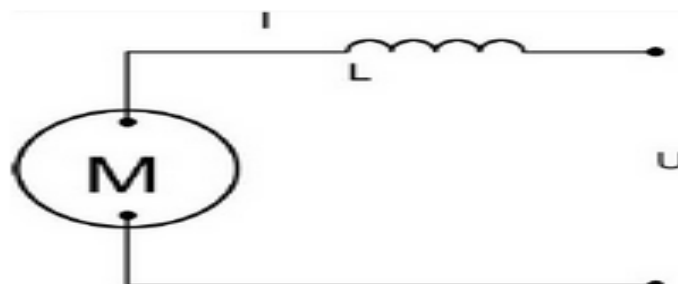


Figure. I.9: Modélisation électrique d'un Moteur à excitation série

### I.4. 4. Moteur à excitation compound (Composée)

C'est une combinaison des excitations Shunt et Série, on peut réaliser un compound-âge additif (Si les flux des deux enroulements s'additionnent) ou soustractif (dans le cas contraire).

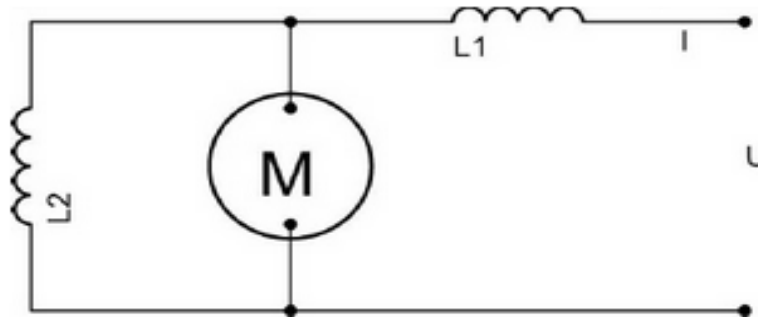


Figure. I.10: Modélisation électrique d'un Moteur à excitation compound

### I.5. Principe général de la régulation

Dans la plupart des appareils dans des installations industrielles et domestiques, il est nécessaire de maintenir des grandeurs physiques à des valeurs déterminées, en dépit des variations externes ou internes influant sur ces grandeurs. Par exemple, le niveau d'eau dans un réservoir, la température d'une étuve, la vitesse et la position des moteurs, étant par nature variables, doivent donc être réglés par des actions convenables sur le processus considéré. Si les perturbations influant sur la grandeur à contrôler sont lentes ou négligeables, un simple réglage dit en boucle ouverte, permet d'obtenir et de maintenir la valeur demandée (par exemple : action sur un robinet d'eau).

Dans la majorité des cas, cependant, ce type de réglage n'est pas suffisant, parce que trop grossier ou instable. Il faut alors comparer, en permanence, la valeur mesurée de la grandeur réglée à celle que l'on souhaite obtenir et agir en conséquence sur la grandeur d'action, dite grandeur réglant. On a, dans ce cas, constitué une boucle de régulation et plus généralement une boucle d'asservissement. Cette boucle nécessite la mise en œuvre d'un ensemble de moyens de mesure, de traitement de signal ou de calcul, d'amplification et de commande d'actionneur, constituant une chaîne de régulation ou d'asservissement.

### I.6. Système en boucle ouverte

Un système est dit en boucle ouverte lorsque la commande est élaborée sans l'aide de la connaissance des grandeurs de sortie. Au niveau des inconvénients, il n'y a aucun moyen de contrôler, à plus forte raison de compenser les erreurs, les dérives, les accidents qui peuvent intervenir à l'intérieur de la boucle, autrement dit, il n'y a pas de précision ni surtout de fidélité qui dépendent de la qualité intrinsèque des composants. Enfin, le système en boucle ouverte ne compense pas les signaux de perturbation, le schéma bloc de système en boucle ouverte est donné par la figure ci-dessous, [1].

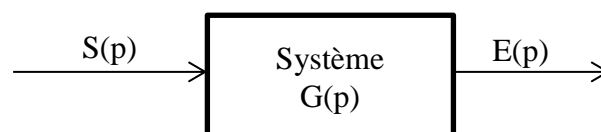


Figure. I.11: Schéma bloc d'un système en boucle ouverte

Dans ce dernier cas, le comportement du processus est décrit par la relation :

- $S(p)=G(p) E(p)$  (8)
- S: grandeur réglée (sortie)
- E : grandeur réglant (entrée)
- G(p): fonction de transfert

### I.7. Système en boucle fermée

La boucle fermée (contre réaction) est capable de stabiliser un système instable en boucle ouverte. Dans une régulation en boucle fermée, une bonne partie des facteurs perturbateurs externes sont automatiquement compensés par la contre-réaction à travers le procédé. L'utilisation du retour d'information est le principe fondamental en automatique. La commande appliquée au système est élaborée en fonction de la consigne et de la sortie. La (figure. I.10) représente le principe de retour unitaire, [1]

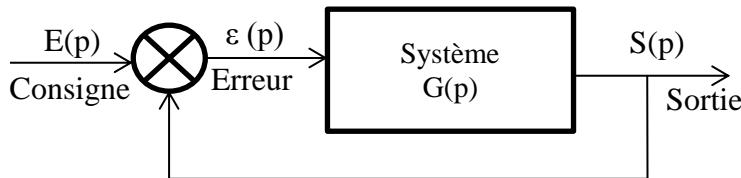


Figure. I.12: Schéma bloc d'un système en boucle fermée avec un retour unitaire

Dans ce dernier cas, le comportement du processus est décrit par la relation :

$$H_{BF}(p) = \frac{G(p)}{1+G(p)} \quad (9)$$

$H_{BF}$  : fonction de transfert en boucle fermée

E : grandeur réglant (consigne)

S: grandeur réglée: erreur

$\varepsilon$ : erreur= E(p) – S(p)

### I.8. Représentation des systèmes linéaires

Un système est dit linéaire invariant si l'équation liant la sortie à l'entrée est une équation différentielle linéaire à coefficients constants. La forme générale de cette équation est :

$$a_0s(t) + a_1 \frac{ds(t)}{dt} + \dots + a_n \frac{d^n s(t)}{dt^n} = b_0e(t) + b_1 \frac{de(t)}{dt} + \dots + b_m \frac{d^m e(t)}{dt^m} \quad (10)$$

D'où la fonction de transfert d'un système linéaire est le rapport entre la sortie et de l'entrée dans le domaine de Laplace :

$$G(p) = \frac{S(p)}{E(p)} = \frac{b_0 + b_1 p + \dots + b_m p^m}{a_0 + a_1 p + \dots + a_n p^n} \quad (11)$$

### I.9. Précision et Stabilité des systèmes asservis

#### I.9.1. Précision

Un système asservi en boucle fermée est dit précis, si sa sortie  $s(t)$  est proche de la consigne (valeur désirée)  $e(t)$ .

On peut représenter l'erreur entre la consigne et la sortie :

$$\varepsilon(p) = E(p) - S(p) \quad (12)$$

Dans ce cas, on s'intéresse à  $\lim_{t \rightarrow \infty} \varepsilon(t) = \lim_{p \rightarrow 0} p \cdot (E(p) - S(p))$

Cette erreur dépend de l'entrée et du gain de la fonction de transfert, car l'augmentation du gain permet d'avoir une meilleure précision, [2] On peut trouver deux types de précision et d'erreur :

- **Précision dynamique** : caractérisé pendant le régime transitoire essentiellement pour une sollicitation en échelon de position. Elle est liée directement au degré de la stabilité (marge de gain et marge de phase). La précision statique: a pour but d'évoluer l'aptitude de système à suivre. Elle est caractérisée par la différence en régime permanent entre l'entrée (consigne) et la sortie (réponse), cette différence appelée écart ou erreur.
- **Erreur statique** : on appelle erreur statique la valeur lorsque le temps tend vers l'infini de (t) pour une réponse d'asservissement à un échelon unitaire
- **Erreur de vitesse** : on appelle erreur de vitesse la valeur de lorsque le temps tend vers l'infini pour une réponse d'asservissement à une rampe unitaire

## I.9.2. Stabilité

Dans le cas des systèmes linéaires représentés par une fonction de transfert, l'analyse des pôles permet de conclure sur la stabilité du système. On rappelle que, si les pôles de la fonction de transfert  $G(p)$  sont  $P_0, P_1 \dots P_n$ . Dans le cas d'une fonction de transfert continue utilisant la transformée de Laplace, tous les pôles doivent être à partie réelle strictement négative pour que le système soit stable, [2]

Pour :  $i = 0 : n$   $P_i = Re_i + j.Im_i$   
 $P_i$ : pôles de la fonction de transfert  
 $Re_i$ : partie réelle de  $P_i$   
 $Im_i$ : partie imaginaire de  $P_i$

On définira la stabilité par une des propositions suivantes : Un système linéaire est stable :

- ✓ Lorsque sa réponse à un échelon prend une valeur finie en régime Permanent.
- ✓ Lorsque sa réponse à une impulsion tend vers 0.
- ✓ Lorsque sa réponse à une sinusoïde est une sinusoïde d'amplitude finie.

## I.10. Influence des perturbations

Une perturbation est une entrée supplémentaire au système qu'on ne peut contrôler ou maîtriser. Ces perturbations ont une influence sur l'asservissement des systèmes, voir la (Figure. I.13)

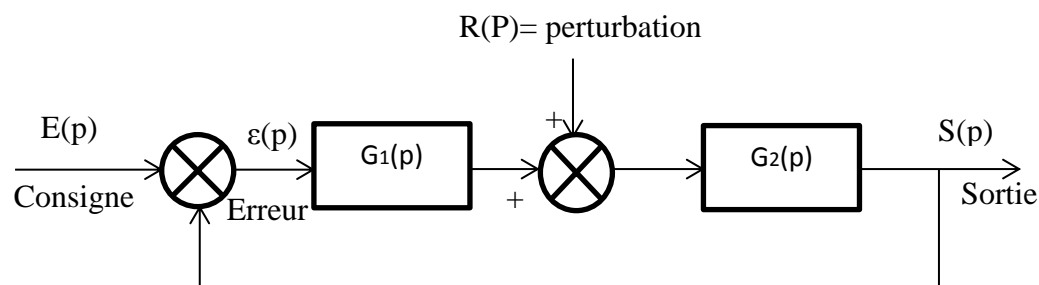


Figure. I.13: Schéma bloc d'un asservissement avec une perturbation (Retour unitaire)

La fonction de transfert du bloc d'asservissement est représentée comme suit :  
Calculons  $S(p)$  en fonction de  $R(p)$ , (En posant  $E(p) = 0$ ) :

$$S_{R(p)} = \frac{G_2(P)}{1+G_1(P)G_2(P)} R(p) \quad (13)$$

Calculons  $S(p)$  en fonction de  $E(p)$ , (En posant  $R(p) = 0$ ) :

$$S_{E(p)} = \frac{G_1(P)G_2(P)}{1+G_1(P)G_2(P)} E(P) \quad (14)$$

La sortie totale est

$$S(p) = \frac{G_2(P)}{1+G_1(P)G_2(P)} R(P) + \frac{G_1(P)G_2(P)}{1+G_1(P)G_2(P)} E(P) \quad (15)$$

## I.11. Compensation des systèmes asservis

La plupart des processus ont besoin de correcteurs pour compenser et d'améliorer la précision et la stabilité. Un correcteur est un système qui va élaborer la commande d'un système en fonction de l'erreur mesurée entre la sortie et la consigne.

Si on prend le cas d'un correcteur proportionnel est un système qui donne une commande proportionnelle à l'erreur mesurée. Beaucoup de systèmes peuvent être commandés par ce type de correcteur, qui est simple à mettre en œuvre. Le principe c'est l'ajustement du gain qui va consister à obtenir un bon compromis entre la stabilité et la précision. [3]

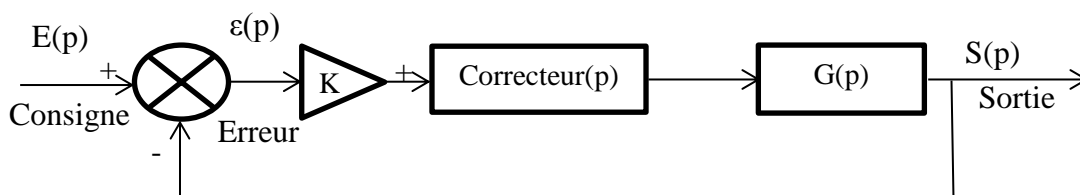


Figure. I.14: Schéma bloc d'un système avec correcteur (Retour unitaire)

Ce genre de correcteur n'est pas toujours possible ou suffisant. Des correcteurs plus sophistiqués peuvent permettre de :

- Stabiliser un système instable
- Augmenter le degré de la stabilité sans réduire le gain K
- Réduire ou annuler les erreurs statiques sans toucher à la stabilité

## I.12 La commande PID

La commande PID est dite aussi (correcteur, régulateur, contrôleur), se compose de trois termes P, I et D, d'où le 'P' correspond au terme proportionnel, 'I' pour terme intégral et 'D' pour le terme dérivé de la commande. Les régulateurs PID sont probablement les plus largement utilisés dans le contrôle industriel. Même les plus complexes systèmes de contrôle industriel peut comporter un réseau de contrôle dont le principal élément de contrôle est un module de contrôle PID.

Le régulateur PID est une simple implémentation de retour d'information (Feedback). Il a la capacité d'éliminer la compensation de l'état d'équilibre grâce à l'action intégrale, et il peut anticiper le futur grâce à une action dérivée, [8].



Ce chapitre a pour but, d'implémenter la commande PID classique pour un moteur à courant continu à excitation séparée, pour un seul objectif est d'annuler l'erreur statique, diminuer le dépassement, diminuer le temps de réponse et le temps de montée afin d'obtenir une réponse adéquate du procédé et de la régulation et d'avoir un système précis, rapide, stable et robuste.

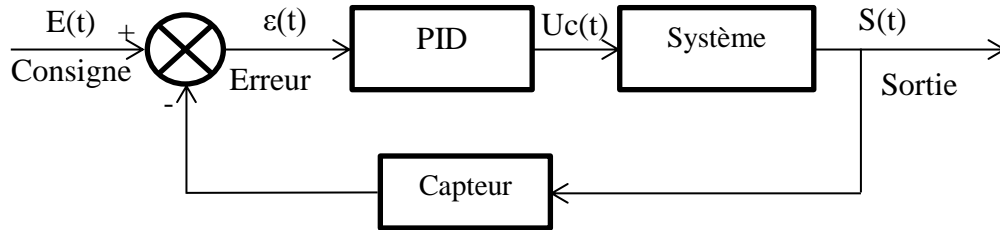


Figure. I.15: Schéma bloc d'un système avec correcteur (Boucle fermée)

### I.12. 1.La commande PID en régulation de vitesse et de position

La commande proportionnel-intégral-dérivé (PID) est insérée dans la chaîne directe de l'asservissement, en série avec le processus, comme indiqué dans la (Figure.I.16).Ce régulateur élabore à partir du signal d'erreur (t) une commande  $u_c(t)$  en fonction de trois actions proportionnelle, intégrale, dérivée, [9].

$$\begin{aligned}
 U_c(t) &= K_p \varepsilon(t) + \frac{1}{T_i} \int_0^t \varepsilon(t) dt + T_d \frac{d\varepsilon(t)}{dt} \\
 &= K_p \varepsilon(t) + K_i \int_0^t \varepsilon(t) dt + K_d \frac{d\varepsilon(t)}{dt} \\
 \xrightarrow{TL} U_c(p) &= K_p \varepsilon(p) + K_i \frac{d\varepsilon(p)}{p} + K_d \varepsilon(p)p \quad (16)
 \end{aligned}$$

- $K_p$ : gain d'action proportionnelle
- $K_i = 1/T_i$ : gain d'action intégrale.
- $K_d = T_d$ : gain d'action dérivée.
- $T_i$ : constante de temps, dite temps d'action intégrale.
- $T_d$ : constante de temps, dite temps d'action dérivée.

Le régulateur PID est donc conçu dans le domaine temporel comme la somme des trois actions. On obtient alors un asservissement composée d'un terme proportionnel, un terme intégral et un terme dérivé, mises en parallèle, on parle d'asservissement PID:

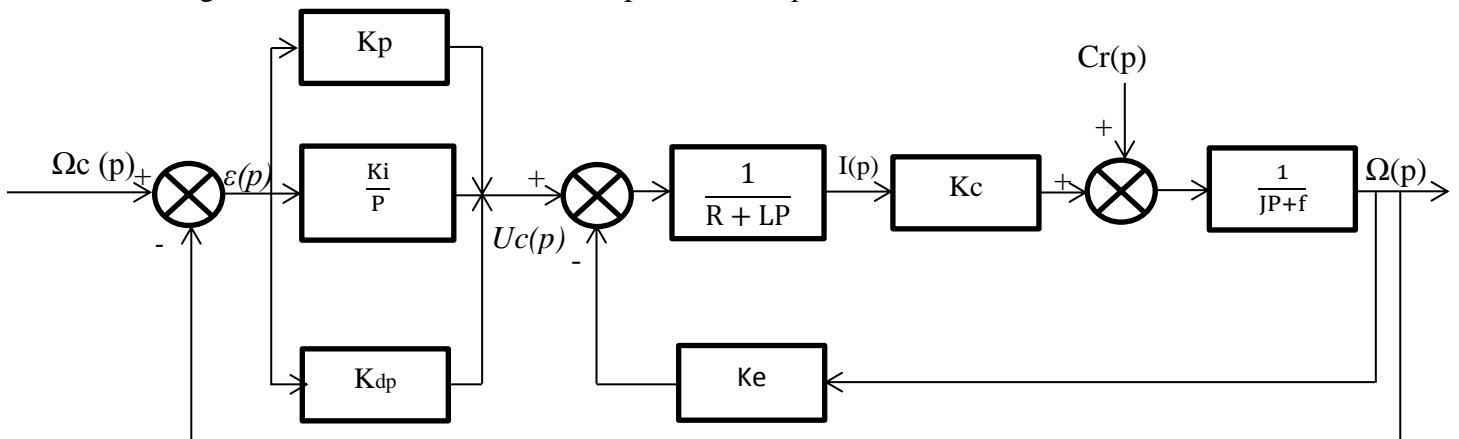


Figure. I.16: Schéma bloc du correcteur PID en régulation de vitesse

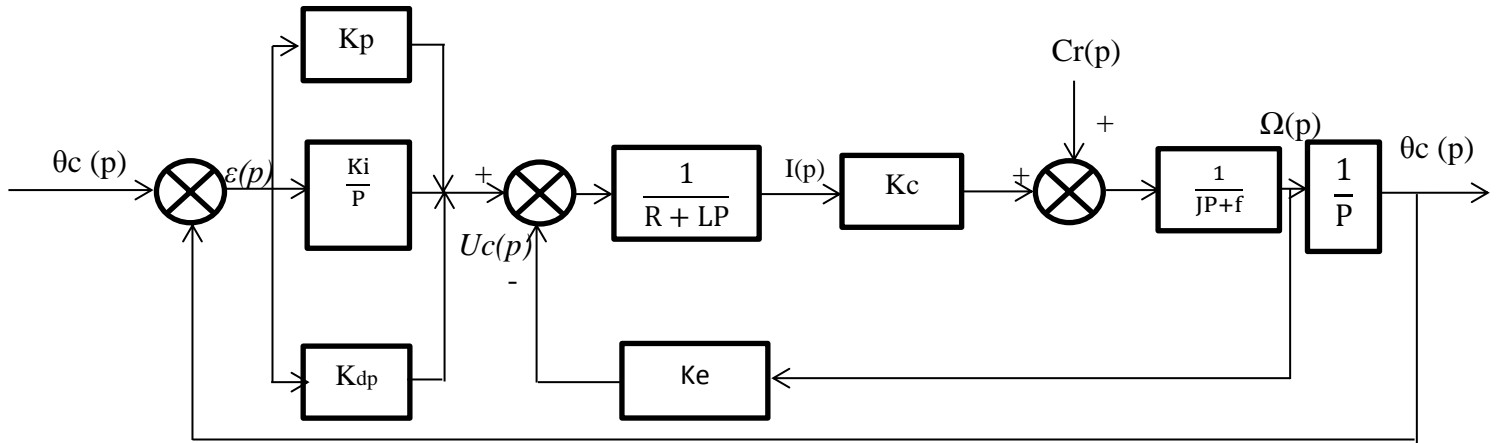


Figure. I.17: Schéma bloc du correcteur PID en régulation de position

Dans les modélisations qui suivent, posons comme références :  $\Omega_c=200$  rad/s qui correspond à 12volts d'alimentation pour la vitesse de rotation et  $\theta_c =90^\circ$  pour la position angulaire.

### ***I.13. Conclusion***

*Dans ce chapitre, nous avons présenté les différents types de moteurs à courant continu et les principaux éléments des systèmes asservis en boucle ouverte et en boucle fermée, l'influence des perturbations externes ainsi l'utilité de la commande (correction) pour un système donné et définir le système dont on doit appliquer la régulation PID.*

## 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 en 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 Définition du module Arduino

Le module Arduino est un circuit imprimé en matériel libre (plateforme de contrôle) dont les plans de la carte elle-même sont publiés en licence libre dont certains composants de la carte : comme le microcontrôleur et les composants complémentaires qui ne sont pas en licence libre. Un microcontrôleur programmé peut analyser et produire des signaux électriques de manière à effectuer des tâches très diverses. Arduino est utilisé dans beaucoup d'applications comme l'électrotechnique industrielle et embarquée ; le modélisme, la domotique mais aussi dans des domaines différents comme l'art contemporain et le pilotage d'un robot, commande des moteurs et faire des jeux de lumières, communiquer avec l'ordinateur, commander des appareils mobiles (modélisme). Chaque module d'Arduino possède un régulateur de tension +5 V et un oscillateur à quarcez 16 MHz (ou un résonateur céramique dans certains modèles). Pour programmer cette carte, on utilise l'logiciel IDE Arduino. [3]

## II.3 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 ATmega8.
- ✓ **L'extrémité d'Arduino**, avec une interface d'USB pour programmer et usage d'un Microcontrôleur ATmega8.
- ✓ **L'Arduino Mini**, une version miniature de l'Arduino en utilisant un microcontrôleur ATmega168.
- ✓ **L'Arduino Nano**, une petite carte programme à l'aide porte USB cette version utilisant un microcontrôleur ATmega168 (ATmega328 pour une plus nouvelle version).
- ✓ **Le LilyPad Arduino**, une conception de minimaliste pour l'application wearable en utilisant un microcontrôleur ATmega168.
- ✓ **Le NG d'Arduino plus**, avec une interface d'USB pour programmer et usage d'un ATmega168.

- ✓ **L'Arduino Bluetooth**, avec une interface de Bluetooth pour programmer en utilisant un microcontrôleur ATmega168.
- ✓ **L'Arduino Diecimila**, avec une interface d'USB et utilise un microcontrôleur ATmega168.
- ✓ **L'Arduino Duemilanove** ("2009"), en utilisant un microcontrôleur l'ATmega168 (ATmega328 pour une plus nouvelle version) et actionné par l'intermédiaire de la puissance d'USB/DC.
- ✓ **L'Arduino Mega**, en utilisant un microcontrôleur ATmega1280 pour I/O additionnel et mémoire.
- ✓ **L'Arduino UNO**, utilisations microcontrôleur ATmega328.
- ✓ **L'Arduino Mega2560**, utilisations un microcontrôleur ATmega2560, et possède toute la mémoire à 256 KBS. Elle incorpore également le nouvel ATmega8U2 (ATmega16U2 dans le jeu de puces d'USB de révision 3).
- ✓ **L'Arduino Leonardo**, avec un morceau ATmega32U4 qui élimine le besoin de raccordement d'USB et peut être employé comme clavier.
- ✓ **L'Arduino Esplora** : ressemblant à un contrôleur visuel de jeu, avec un manche et des sondes intégrées pour le bruit, la lumière, la température, et l'accélération. [4]

Parmi ces types, nous avons choisi une carte Arduino UNO (carte Basique). L'intérêt principal de cette carte est de faciliter la mise en oeuvre d'une telle commande qui sera détaillée par la suite.

L'Arduino fournit un environnement de développement s'appuyant sur des outils open source comme interface de programmation. L'injection du programme déjà converti par l'environnement sous forme d'un code « HEX » dans la mémoire du microcontrôleur se fait d'une façon très simple par la liaison USB. En outre, des bibliothèques de fonctions "clé en main" sont également fournies pour l'exploitation d'entrées-sorties. Cette carte est basée sur un microcontrôleur ATmega 328 et des composants complémentaires.

La carte Arduino contient une mémoire morte de 1 kilo. Elle est dotée de 14 entrées/sorties digitales (dont 6 peuvent être utilisées en tant que sortie PWM), 6 entrées analogiques et un cristal à 16 MHz, une connexion USB et Possède un bouton de remise à zéro et une prise jack d'alimentation.

La carte est illustrée dans la figure ci-dessous. [1]

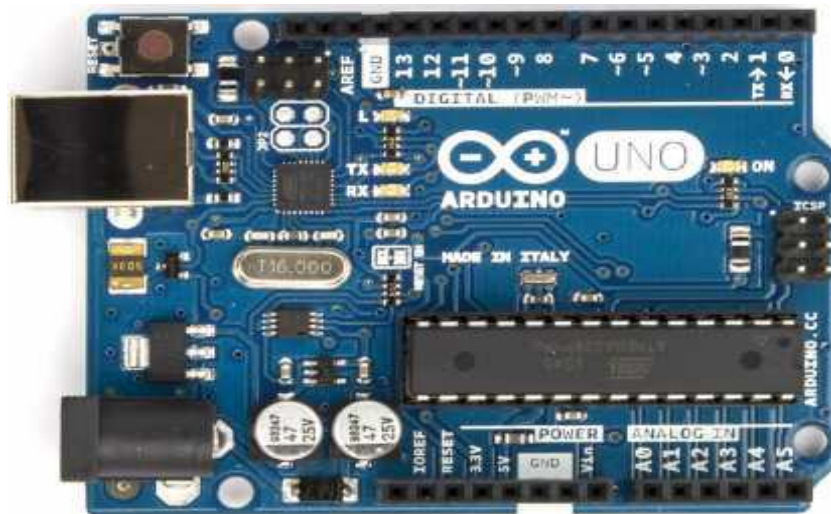


Figure II.1 La carte Arduino UNO

## II.4 Avantages de la carte Arduino

Il y a de nombreuses cartes électroniques qui possèdent des plateformes basées sur des microcontrôleurs disponibles pour l'électronique programmée. Tous ces outils prennent en charge les détails compliqués de la programmation et les intègrent dans une présentation facile à utiliser.

De la même façon, le système Arduino simplifie la façon de travailler avec les microcontrôleurs tout en offrant à personnes intéressées plusieurs avantages cités comme suit:

- ✓ **Prix (réduits) :** les cartes Arduino sont relativement peu coûteuses comparativement aux autres plates-formes.
- ✓ **Multi plateforme :** 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.
- ✓ **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.
- ✓ **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. Le logiciel de programmation des modules Arduino est une application JAVA multi plateformes (fonctionnant sur tout système d'exploitation), servant d'éditeur de code et de compilateur, et qui peut transférer le programme au travers de la liaison série (RS232, Bluetooth ou USB selon le module).
- ✓ **Matériel extensible :** les cartes Arduino sont basées sur les microcontrôleurs Atmel ATMEGA8, ATMEGA168, ATMEGA 328, 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. Même les utilisateurs relativement inexpérimentés peuvent fabriquer la

version sur plaque d'essai de la carte Arduino, dont le but est de comprendre comment elle fonctionne pour économiser le coût. [20].

## II. 5 Constitution de la carte Arduino UNO

Un module Arduino est généralement construit autour d'un microcontrôleur ATMEL AVR, et de composants complémentaires qui facilitent la programmation et l'interfaçage avec d'autres circuits. Chaque module possède au moins un régulateur linéaire 5V et un oscillateur à quartz 16 MHz (ou un résonateur céramique dans certains modèles). Le microcontrôleur est préprogrammé avec un bootloader de façon à ce qu'un programmeur dédié ne soit pas nécessaire.

### II.5.1 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.5.1.1 Microcontrôleur ATmega328

Un microcontrôleur ATmega328 est un circuit intégré qui rassemble sur une puce plusieurs éléments complexes dans un espace réduit au temps des pionniers de l'électronique. 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.

La figure II.2 montre un microcontrôleur ATmega 328, qu'on trouve sur la carte Arduino.[1]



Le composant CMS



Le composant classique

Figure II.2 Microcontrôleur ATmega328

Le microcontrôleur ATmega328 est constitué par un ensemble d'éléments qui ont chacun une fonction 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 :

- **Mémoire Flash:** C'est celle qui contiendra le programme à exécuter. Cette mémoire est effaçable et réinscriptible mémoire programme de 32Ko (dont bootloader de 0.5 ko).

- **RAM** : c'est la mémoire dite "vive", elle va contenir les variables du programme. Elle est dite "volatile" car elle s'efface si on coupe l'alimentation du microcontrôleur. Sa capacité est 2 ko.
- **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. [2]

### II.5.1.2 Les sources de l'alimentation de la carte

On peut distinguer deux genres de sources d'alimentation (Entrée Sortie) et cela comme suit :

- **VIN**: La tension d'entrée positive lorsque la carte Arduino est utilisée avec une source de tension externe (à distinguer du 5V de la connexion USB ou autre source 5V régulée). On peut alimenter la carte à l'aide de cette broche, ou, si l'alimentation est fournie par le jack d'alimentation, accéder à la tension d'alimentation sur cette broche.
- **La tension régulée 5V** : utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (pour info : les circuits électroniques numériques nécessitent une tension d'alimentation parfaitement stable dite "tension régulée" obtenue à l'aide d'un composant appelé un régulateur et qui est intégré à la carte Arduino).

Le tension régulée 5V régulé fourni par cette broche peut donc provenir soit de la tension d'alimentation VIN via le régulateur de la carte, ou bien de la connexion USB (qui fournit du 5V régulé) ou de tout autre source d'alimentation régulée.

- **Une alimentation de 3.3V** : fournie par le circuit intégré FTDI (circuit intégré faisant l'adaptation du signal entre le port USB de votre ordinateur et le port série de l'ATmega) de la carte est disponible , ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5V. L'intensité maximale disponible sur cette broche est de 50mA. [4]

### II.5.1.3 Les entrées et sorties

Cette carte possède 14 broches numériques (numérotée de 0 à 13) peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions `pinMode()`, `digitalWrite()` et `digitalRead()` du langage Arduino. Ces broches fonctionnent en 5V.

Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité et dispose d'une résistance interne de "rappel au plus" (pull-up) (déconnectée par défaut) de 20-50 KOhms. Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction `digitalWrite` (broche, HIGH).

En plus, certaines broches ont des fonctions spécialisées :

- ✓ **Interruptions Externes**: Broches 2 et 3. Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur. Impulsion PWM (largeur d'impulsion modulée) : Broches 3, 5, 6, 9, 10, et 11. Fournissent une impulsion PWM 8-bits à l'aide de l'instruction `analogWrite()`.
- ✓ **SPI (Interface Série Périphérique)**: Broches 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ces broches supportent la communication SPI (Interface Série Périphérique)



disponible avec la librairie pour communication SPI. Les broches SPI sont également connectées sur le connecteur ICSP qui est mécaniquement compatible avec les cartes Mega.

- ✓ **I2C:** Broches 4 (SDA) et 5 (SCL). Supportent les communications de protocole I2C (ou interface TWI (Two Wire Interface - Interface "2 fils"), disponible en utilisant la librairie Wire/I2C (ou TWI - Two-Wire interface - interface "2 fils").
- ✓ **LED:** Broche 13. Il y a une LED incluse dans la carte connectée à la broche 13. Lorsque la broche est au niveau HAUT, la LED est allumée, lorsque la broche est au niveau BAS, la LED est éteinte.

La carte UNO dispose 6 entrées analogiques (numérotées de 0 à 5), chacune pouvant fournir une mesure d'une résolution de 10 bits (càd sur 1024 niveaux soit de 0 à 1023) à l'aide de la très utile fonction `analogRead()` du langage Arduino. Par défaut, ces broches mesurent entre le 0V (valeur 0) et le 5V (valeur 1023), mais il est possible de modifier la référence supérieure de la plage de mesure en utilisant la broche AREF et l'instruction `analogReference()` du langage Arduino.

La carte Arduino UNO intègre un fusible qui protège le port USB de l'ordinateur contre les surcharges en intensité (le port USB est généralement limité à 500mA en intensité). Bien que la plupart des ordinateurs aient leur propre protection interne, le fusible de la carte fournit une couche supplémentaire de protection. Si plus de 500mA sont appliqués au port USB, le fusible de la carte coupera automatiquement la connexion jusqu'à ce que le court-circuit ou la surcharge soit stoppé. [3]

### Les ports de communications

La carte Arduino UNO a de nombreuses possibilités de communications avec l'extérieur. L'Atmega328 possède une communication série UART TTL (5V), grâce aux broches numériques 0 (RX) et 1 (TX).

On utilise (RX) pour recevoir et (TX) transmettre (les données séries de niveau TTL). Ces broches sont connectées aux broches correspondantes du circuit intégré ATmega328 programmé en convertisseur USB – vers – série de la carte, composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur.

Comme un port de communication virtuel pour le logiciel sur l'ordinateur, La connexion série de l'Arduino est très pratique pour communiquer avec un PC, mais son inconvénient est le câble USB, pour éviter cela, il existe différentes méthodes pour utiliser ce dernier sans fil

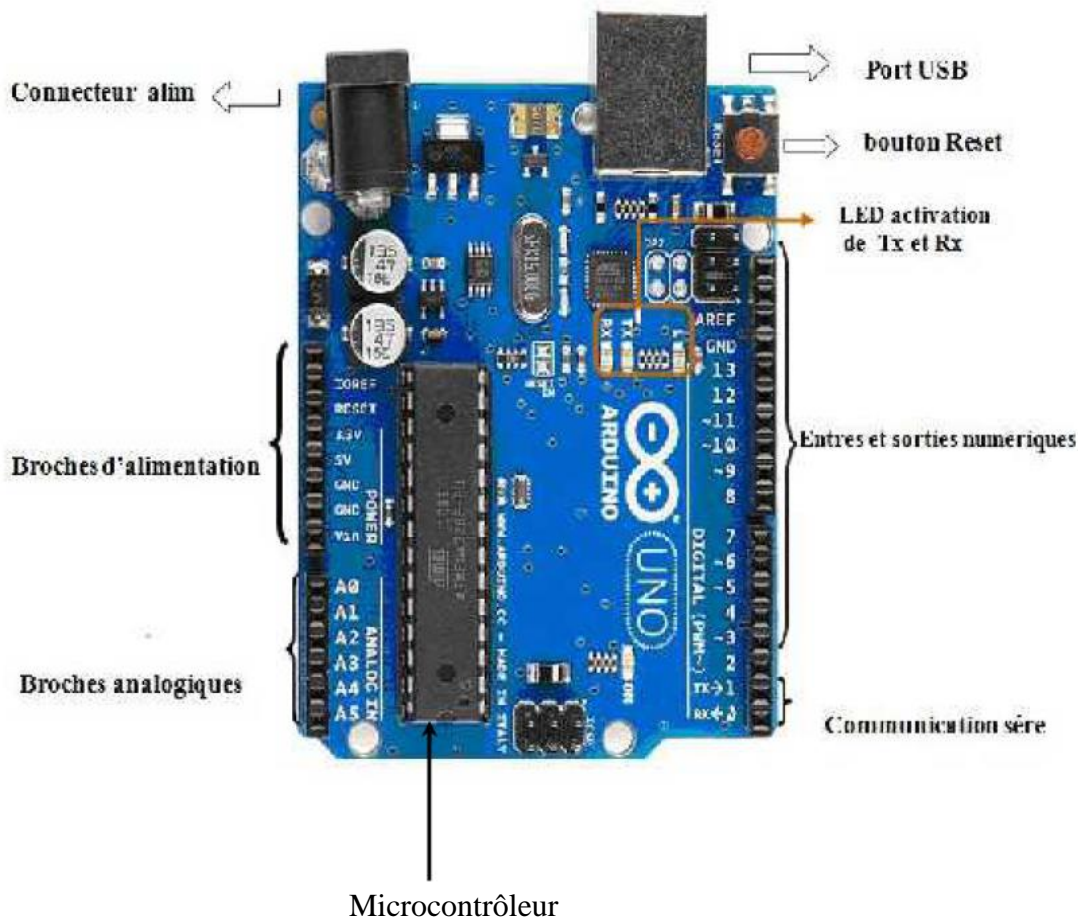


Figure II.3 Constitution de la carte Arduino UNO

## II.5.2 Partie programme

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 Mac OS X, Windows, et Linux).

### II.5.2.1. l'environnement de la programmation

Le logiciel de programmation de la carte Arduino sert d'éditeur de code (langage proche du C). Une fois, le programme tapé ou modifié au clavier, il sera transféré et mémorisé dans la carte à travers de la liaison USB. Le câble USB alimente à la fois en énergie la carte et transporte aussi l'information ce programme appelé IDE Arduino. [7]

### II.5.2.2 .Structure générale du programme (IDE 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 en C.

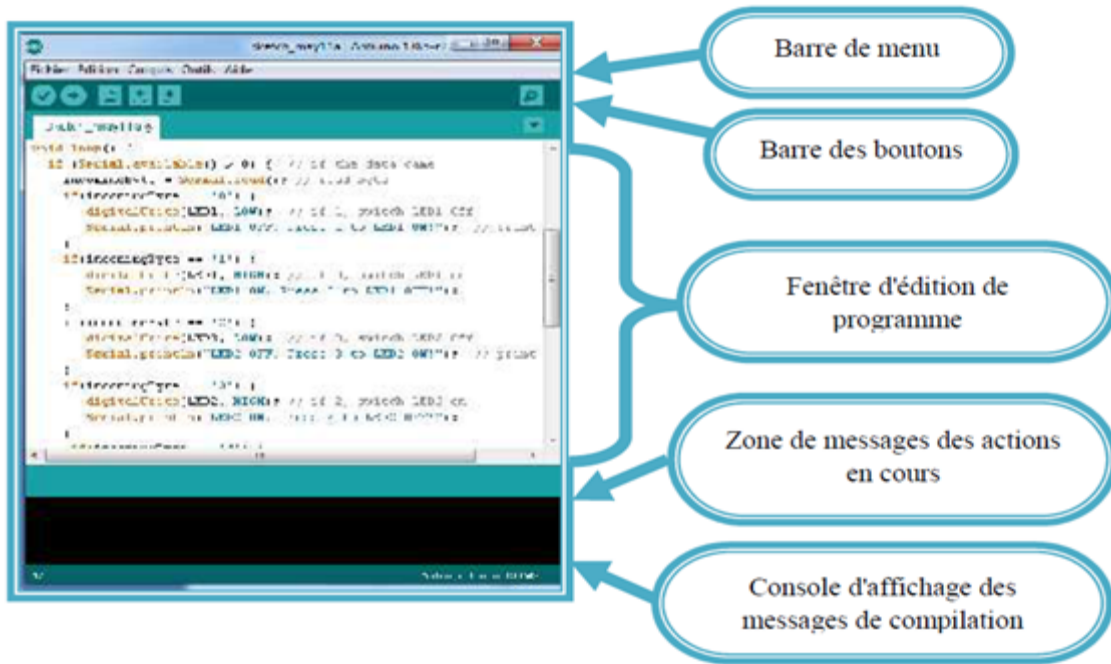


Figure II.4 Interface IDE Arduino

### II.5.2.3 Injection du programme

Avant d'envoyer un programme dans la carte, il est nécessaire de sélectionner le type de la carte ( Arduino UNO) et le numéro de port USB ( COM 3) comme à titre d'exemple cette figure suivante.

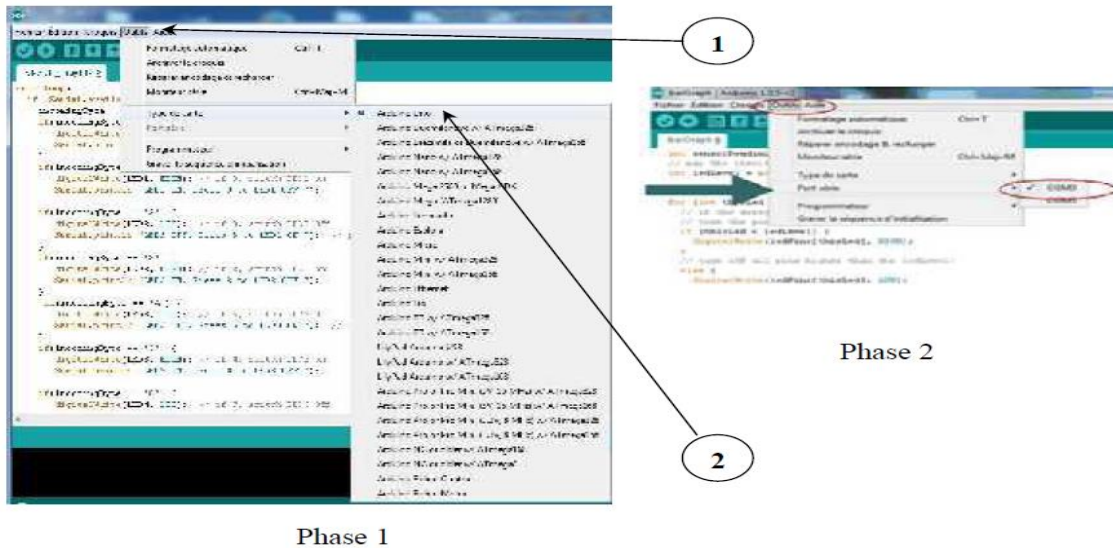


Figure II.5 Paramétrage de la carte

### II.5.2.4 Description du programme

Un programme 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 codes.

Commentaires

Les commentaires sont, en programmation informatique, des portions du code source ignorées par le compilateur ou l'interpréteur, car ils ne sont pas censés influencer l'exécution du programme.

```
1 /* programme de command DC moteur avec Arduino-----
2 *et fait également clignoter la diode de test de la carte-----
3 */-----
```

Définition des variables pour notre montage, on va utiliser une sortie numérique de la carte qui est par exemple la 3<sup>ème</sup> sortie numérique ; cette variable doit être définie et nommée ici moteur pin 3 ; la syntaxe est pour désigner un nombre entier est int.

```
4 int moteur 1 = 3; // mettre le moteur au pin 3-----
Configuration des entres et des sorties : void setup ( )
```

Les broches numériques de l'arduino peuvent aussi bien être configurées en entrées numériques ou en sorties numériques; ici on va configurer moteur pin en sortie ; pin mode (nom, état) est une des quatre fonctions relatives aux entrées – sorties numériques :

### Exemple

```
5 void setup() {-----
6 // mettre le moteur 1 comme sortie:-----
7 pinMode(motor 1, OUTPUT); // lorsque le pin 3 est activé le moteur tourne-----
8 }-----
```

### Programmation des interactions : void loop

Dans cette boucle, on définit les opérations à effectuer dans l'ordre digital write (nom, état) est une autre des quatre fonctions relatives aux entrées – sorties numériques. delay (temps en mili-seconde ) est la commande d'attente entre deux instructions. chaque ligne d'instruction est terminée par un point-virgule. ne pas oublier les accolades qu'encadre la boucle.

### Exemple

```
9 void loop() {-----
10 digital write ( moteur 1,HIGH); -----
11 delay (3000)-----
12 digital Write(moteur 1, LOW);-----
13 delay (1000)-----
14 }-----
```

## II.5.2.5 Les étapes de téléchargement du programme

Une simple manipulation enchaînée doit être suivie afin d'injecter un code vers la carte Arduino via le port USB.

1. On conçoit ou on ouvre un programme existant avec le logiciel IDE Arduino.
2. On vérifie ce programme avec le logiciel Arduino (compilation).
3. Si des erreurs sont signalées, on modifie le programme.
4. On charge le programme sur la carte.

5. On câble le montage électronique.
6. L'exécution du programme est automatique après quelques secondes.
7. On alimente la carte soit par le port USB, soit par une source d'alimentation autonome (Pile 9 volts par exemple).
8. On vérifie que notre montage fonctionne.

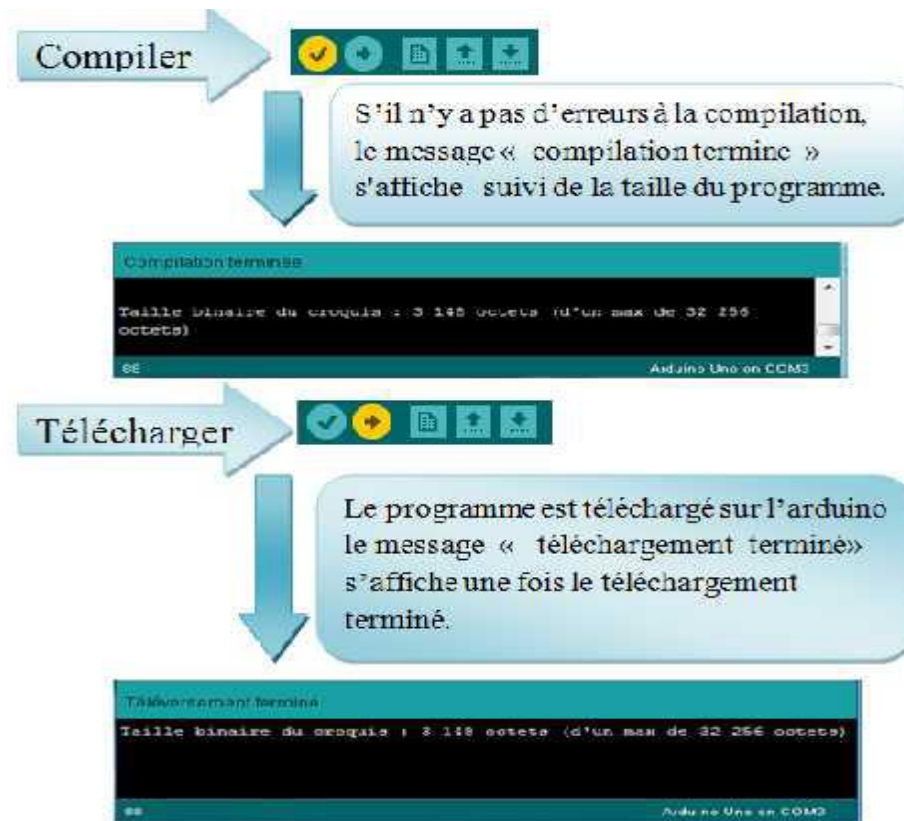


Figure II.6 Les étapes de téléchargement du code

## II.6 Les Accessoires de la carte Arduino

La carte Arduino généralement est associée aux accessoires qui simplifient les réalisations.

### H Bridge L298

Est un Double Pont-H destiné au contrôle de moteur continu (H-Bridge Motor Driver). Il est basé sur le composant L298N qui est un double Pont-H conçu spécifiquement pour ce cas d'utilisation.

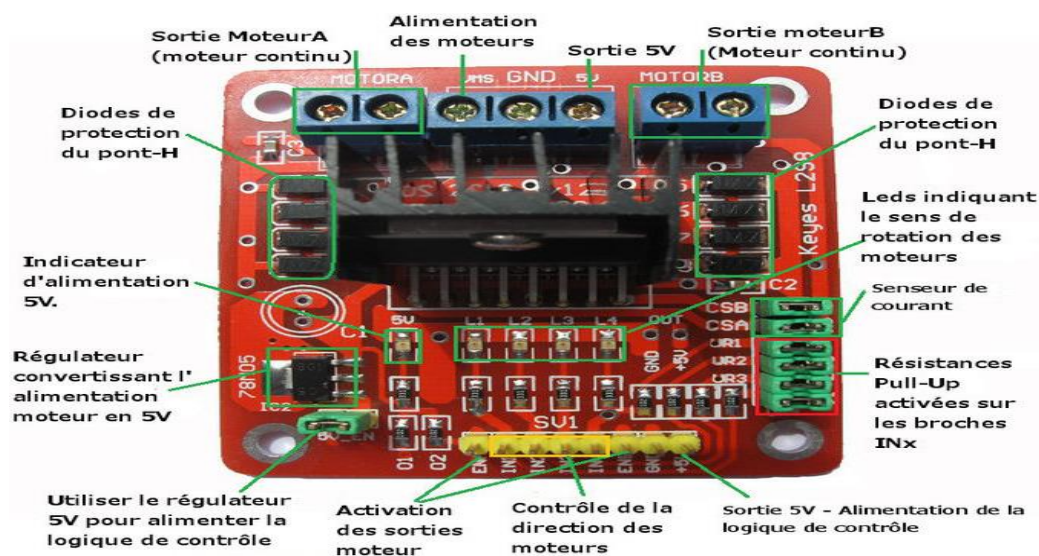
Il peut contrôler deux moteurs courant continu ou un moteur pas-à-pas 4 fils 2 phases. Il est conçu pour supporter des tensions plus élevées, des courants importants tout en proposant une commande logique TTL (basse tension, courant faibles, idéal donc pour un microcontrôleur).

Les deux types de moteurs peuvent être contrôlés aussi bien en vitesse (PWM) qu'en direction. Toutes les sorties en puissance sont déjà protégées par des diodes anti-retour.



## Caractéristiques

- Des capacités hors-pair pour contrôle moteur.
- Diodes de protections.
- Un dissipateur (pour dissiper la chaleur en cas de forte charge).
- Un sélecteur pour sélectionner la source d'alimentation.
- Sélecteurs pour les résistances pull up.
- Sortie pour 2 moteurs continu/ 1 moteur pas-à-pas (4 bobines, deux phases).
- Indicateur LED pour sens de rotation moteur.
- Indicateur LED pour alimentation 5V.



FigureII.7.shield H BridgeL2

## Afficheur à cristaux liquides (LCD)

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).

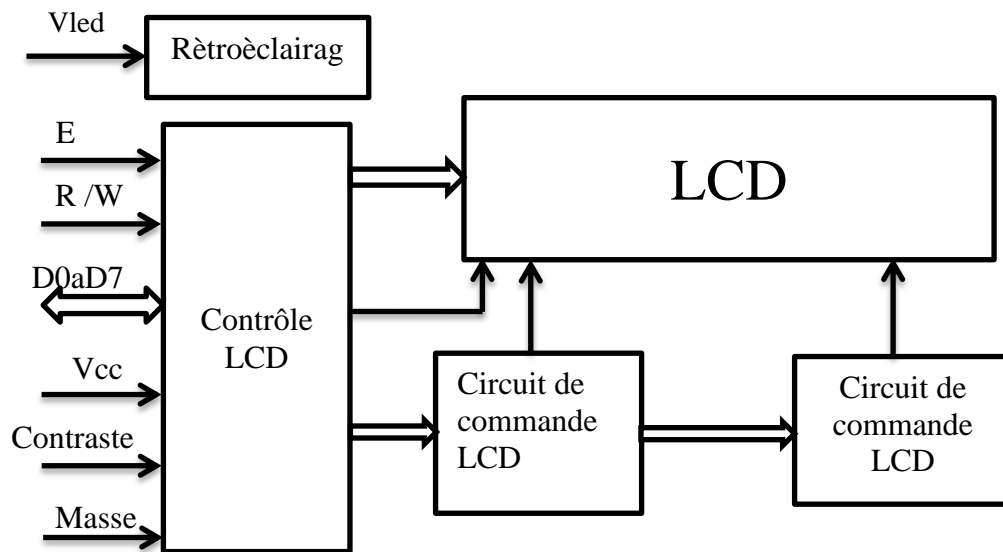
**Schéma fonctionnel :**

Figure II.8 .schéma fonctionnel d'un LCD

Suivant 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.

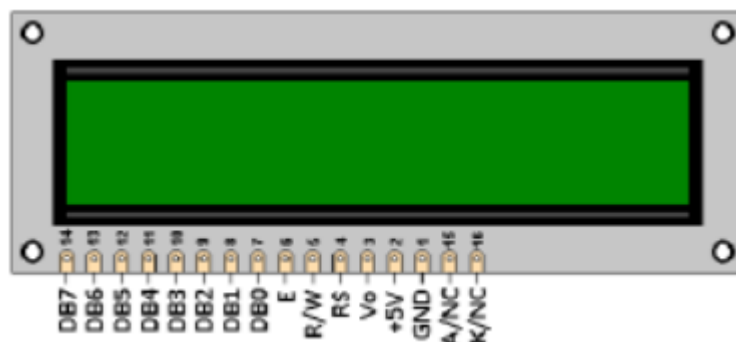


Figure II.9. Photo d'un LCD et son brochage

**Brochage**

Broche	Nom	Niveau	Fonction
1	VSS	-	Masse
2	Vdd	-	Alimentation positive +5V
3	Vo	0 – 5V	Cette tension permet en la faisons varier en 0 et +5V le réglage du constante de l’afficheur
4	RS	TTL	Sélection de registre (registre select) Grace a cette broche la ficheur et capable de faire la différence entre une commande et une donnée . un niveau bas indique une donnée
5	R/W	TTL	Lecture et écriture (Read/Write)L :Ecriture / H :Lecture
6	E	TTL	Entrée de validation (Enable) active sur front descendant.Le niveau haut doit être maintenue au moins 450ns a l’état haut .
7	D0	TLL	
8	D1	TLL	
9	D2	TLL	
10	D3	TLL	D0->D7 Bus de données bidirectionnel 3 états (haute impédance lorsque E=0)
11	D4	TLL	
12	D5	TLL	
13	D6	TLL	
14	D7	TLL	
15	A	-	Anode rétro éclairage (+5V)
16	K	-	Cathode rétro éclairage (masse)

Tableaux II.1.Brochage d'un Afficheur LCD

Les broches 15 et 16 ne sont présentes que sur les afficheurs LCD avec retro- éclairage. Voici l’image pour expliquer Brochage d'un Afficheur LCD avec Arduino :

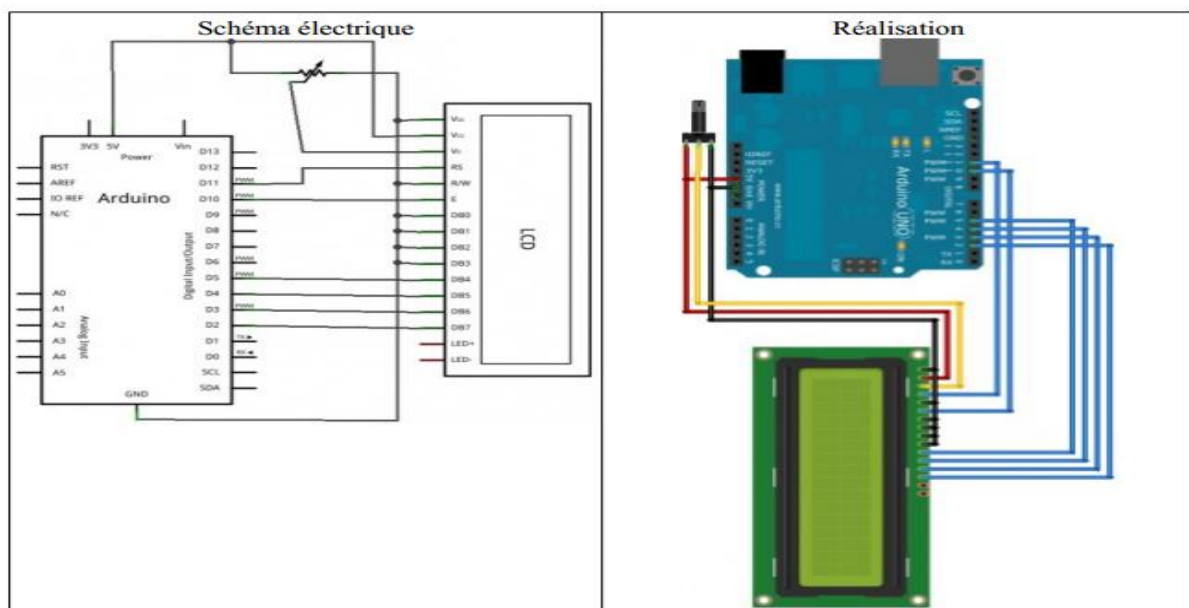


Figure II.10.Brochage d'un Afficheur LCD avec Arduino



## Clavier numérique électronique

Le clavier est généralement constitué de 12 ou 16 touches. L'encodage d'un clavier consiste à délivrer un code (en général sur un octet) fonction de la touche activée. Le Brochage de Arduino est principalement utilisé pour saisir un code, paramétrer une consigne.

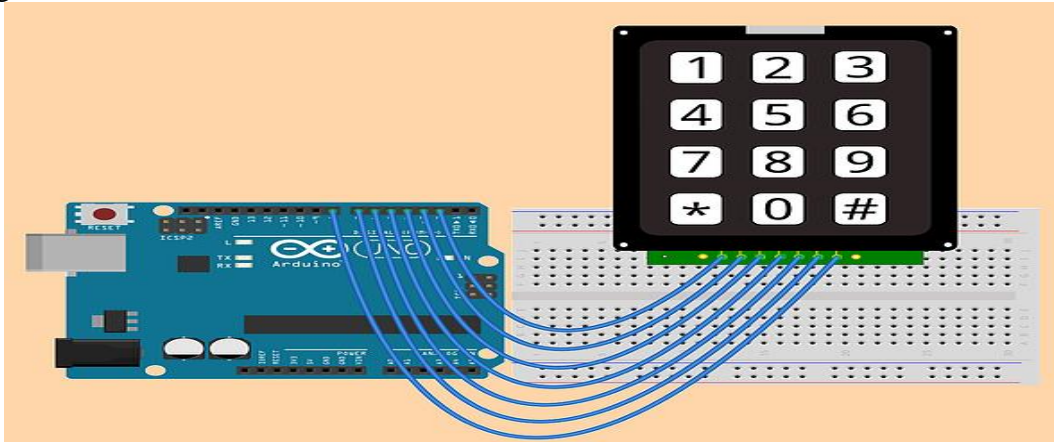


Figure II.11. Brochage Clavier numérique électronique avec Arduino

## Encodeur optique

Encodeur optique c'est un capteur est largement utilisé dans la détection de la vitesse de moteurs, le comptage d'impulsions, la détection de positions limites.

### Caractéristiques

- Utilisation possible avec une roue à encoches
- Largeur de fente de 5 mm
- Indicateur lumineux d'état de sortie
- Signal d'obscurité high, signal non obstrué low
- Signal de sortie du comparateur mis en forme et de plus de 15 mA.
- Tension de fonctionnement de 3,3 V à 5 V
- Sorties digitale (DO digital output) : commutation numérique (0 et 1)
- Installation facile par un trou de boulon de fixation
- Petites dimensions du PCB: 32 x 14 mm
- Utilise un comparateur de tension LM393
- Fonctionnement
- Fente du capteur dégagée, sortie DO sur LOW
- Fente obstruée, sortie DO sur HIGH
- A connecter sur un relais, un buzzer d'alarme, un dispositif de comptage d'impulsions, une roue codeuse de roue de robot ou de machine, etc...

## Schéma des connexions

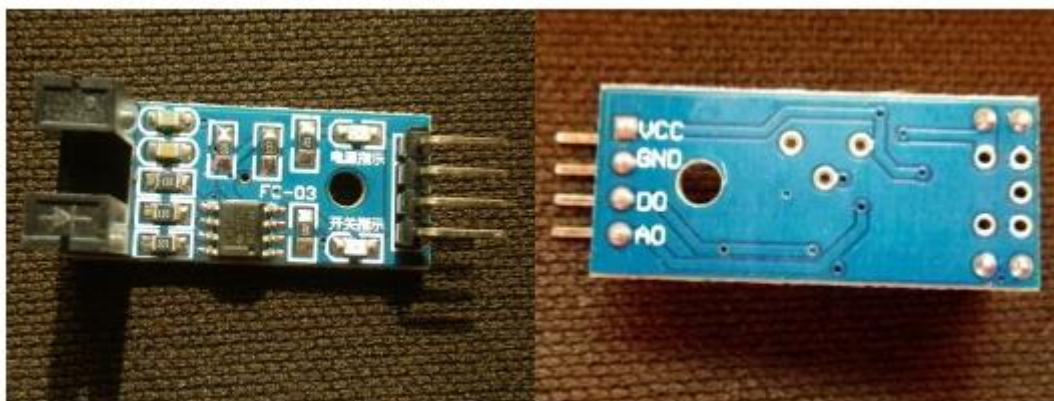


Figure II.12.Photo Encodeur optique

### *II.7 Conclusion*

*Dans ce chapitre, nous avons projeté la lumière sur une carte d'acquisition qui est l'Arduino donnant ainsi les raisons pour lesquelles on l'a choisie, puis nous avons cité des différents types de cette dernière. Ensuite, nous avons expliqué les deux parties essentielles de l'Arduino; (la partie matérielle et la partie de programmation) plus précisément. Nous avons également expliqué le principe de fonctionnement de la carte Arduino sans oublier ses caractéristiques.*

### III.1. Introduction

Dans ce chapitre, on présentera de manière sommaire une vue d'ensemble du dispositif expérimental « réalisation d'une carte électronique commandée par Arduino et mesure par deux encodeur optique.

Ce travail d'initiation à base d'une carte Arduino UNO permet de commander la position d'un moteur à courant continu, après avoir donné dans le chapitre précédent une description théorique sur le module Arduino et son environnement de développement, on va procéder à l'application expérimentale ; pour cette raison, plusieurs blocs ont été nécessaires afin de réaliser une telle combinaison.

### III.2. But

Dans ce but notre travail comporte trois axes principaux :

- Contrôle un moteur à courant continu
- Mise en œuvre de la régulation par un contrôleur PID
- Réalisation d'une interface pour commander le moteur à courant continu

### III.3. Schéma synoptique général

Le schéma synoptique général de notre dispositif est indiqué par la figure (III.1). Pour faciliter cette étude, on a décomposé le schéma synoptique en quatre blocs :

- ✓ Bloc d'alimentation
- ✓ Bloc de mesure
- ✓ Bloc de traitement et commande
- ✓ Bloc de puissance

En ce qui concerne l'élément principal de ce dispositif, notre choix était le dispositif Arduino UNO.

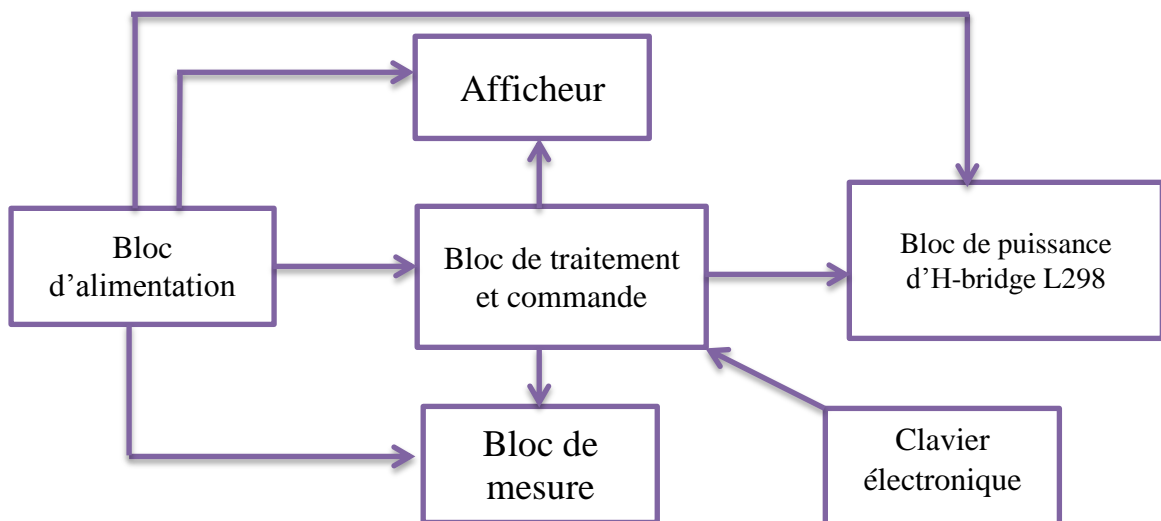


Figure III.1 Schéma synoptique du dispositif

### III.4.Composants utilisés

Pour une telle réalisation, nous avons assemblé les différents composants:

- Une Carte Arduino UNO
- Deux encodeurs optiques
- Schild H-bridge L298.
- Une carte Alimentations
- Moteur à courant continu
- Disque
- Clavier électronique
- Afficheur
- Les files des connexions
- Câble USB

#### II.4.1 Bloc d'alimentation

L'ensemble des dispositifs « pile de +9 volts et source externe, résistances, condensateurs, relais, Arduino UNO, transistors, photo-coupleurs » exige dans la plupart des cas, des alimentations stabilisées de (+5V) , alors dans notre travail, nous avons réalisé une alimentation simple à partir d'une alimentation externe de (<+12V) avec une régulation et filtrage comme il est indiqué sur la figure (III.2).

Nous avons réalisé une alimentation continue de +5 volts Pour l'alimentation de l'Arduino, et une autre alimentation de +12 volts pour l'alimentation de l'étage de puissance.

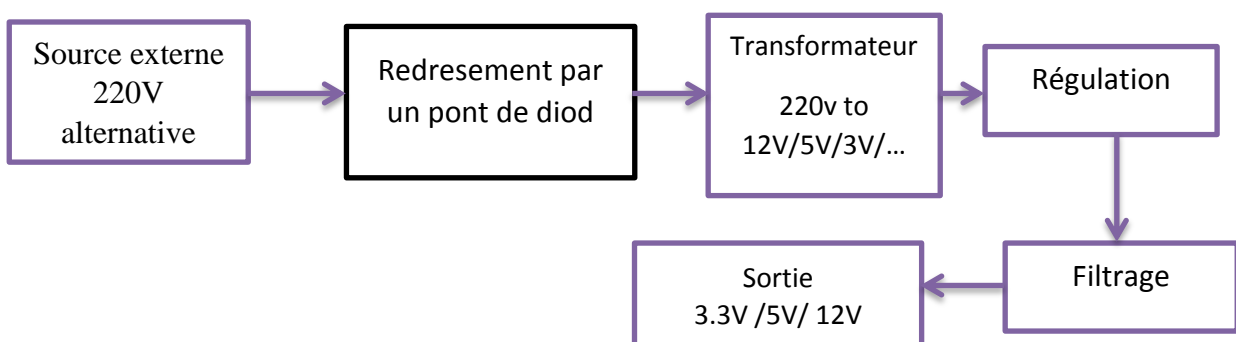


Figure III.2 Schéma synoptique de l'alimentation

Pour la réalisation électrique de ce schéma synoptique «l'alimentation du bloc de commande», figure (III.3), on a utilisé:

- Une résistance 10 k ohms « pour protection » ;

- Des condensateurs « pour filtrage ». Condensateur de 15 $\mu$ F, tous deux placés avant le régulateur le régulateur LM7812.
- le régulateur LM7805.



Figure III.3 Alimentation utilisé dans le projet

Le principe de fonctionnement d'une alimentation est très simple. Il consiste à générer ou réguler +5 volts à partir d'une source continue (+12V) ; cette dernière est filtrée via un filtre passif et régulée par un régulateur type (LM7805) à la sortie duquel, on aura une tension continue d'une valeur de +5 volts.

### III.4.2 bloc de traitement et commande

Le bloc de commande se résume au module Arduino UNO décrit précédemment au chapitre II.

Le bloc de traitement et commande est relié à l'encodeur optique et H-BRIDGE dans le but de faire tourner le disque du moteur selon la position sollicité.

### III.4.3 Bloc de puissance

Notre bloc de puissance comprend un H-bridge L298 Nous avons utilisé un modèle de pont-H vraiment robuste. Nous avons choisi ce composant pour ses les avantages suivants :

- Supporte une tension de puissance jusqu'à 46 volts maximum.
- Débite un courant de service de 2A.
- Compatible TTL (peut donc être commandé directement avec Arduino) .
- Dispose d'un dispositif de mesure du courant.

Ce composant dispose de 15 broches décrites comme suit (figure III.4):  
VSS : alimentation de la logique de commande (+5V) (à raccorder à la borne +5V de d'Arduino ).

- VS : alimentation de puissance des moteurs
- GND : masse de la source d'alimentation de puissance.
- OUTPUT1,OUTPUT2 : broches à raccorder au me moteur.
- INPUT1, INPUT2 : broches de commande du Pont-H. Se raccorde à Arduino.

- ENABLE A = VSS, le pont-H est connecté aux sorties et le moteur fonctionne dans un sens ou l'autre ou pas en fonction des tensions appliquées sur INPUT1 et INPUT2.
- CURRENT SENSING A, CURRENT SENSING B : permet de faire une mesure du courant dans le circuit de puissance (à placer impérativement sur GND si cette fonctionnalité n'est pas utilisée).

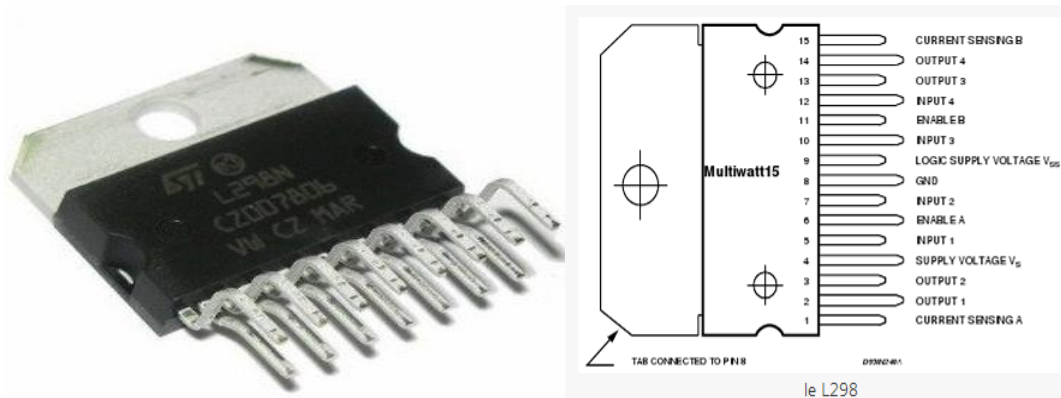


Figure III.4. Schéma de brochage du composant L298

- Pour le premier pont :
  - Les sorties sont situées sur les broches 2 et 3.
  - Les entrées pour le sens de rotation sont la 5 et 7 et la PWM (*enable*) ira sur la broche 6.
- Pour le second pont :
  - Les sorties sont situées sur les broches 13 et 14.
  - Les entrées pour le sens de rotation sont la 10 et 12 et la PWM (*enable*) ira sur la broche 11.
- Pour les deux ponts :
  - La masse, qui est au milieu sur la broche 8.
  - L'alimentation de la logique de commande (le 5V) sur la broche suivante, la 9.
  - Et l'alimentation de la partie puissance sur la broche 4.
  - Je ne mentionne pas les broches 1 et 15 qui sont celles servant à mesurer le courant traversant les ponts. Je doute que vous vous en serviez dans un premier temps et si vous arrivez jusque-là je n'ai aucun doute que vous arriverez à les mettre en œuvre

#### III.4.4. Bloc de mesure

Le bloc de mesure est un capteur dont le rôle est de détecter la position du disque selon les zones (transparente, noire).



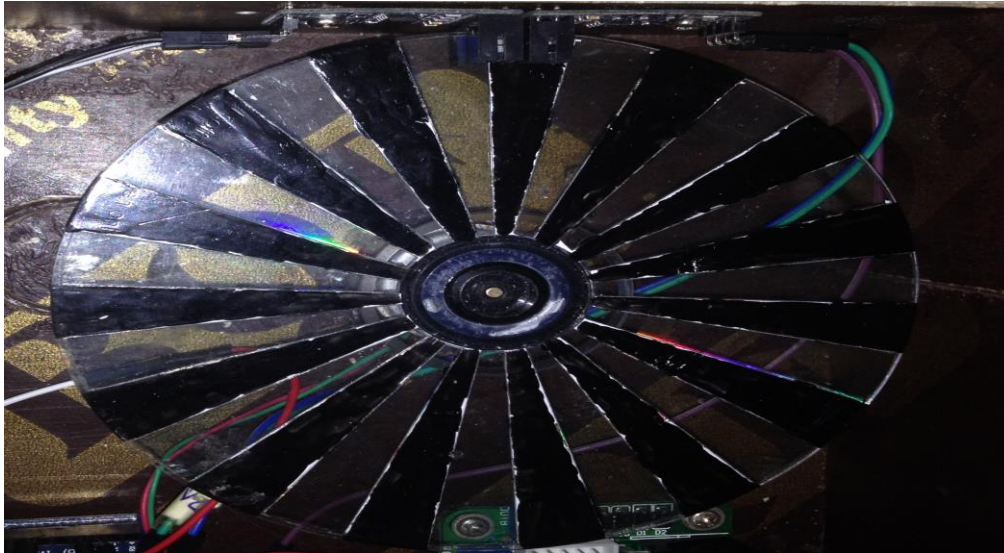


Figure III.5. Disque pour l'encodeur optique (mesure de position de moteur)

### III.5. Schéma électrique du circuit :

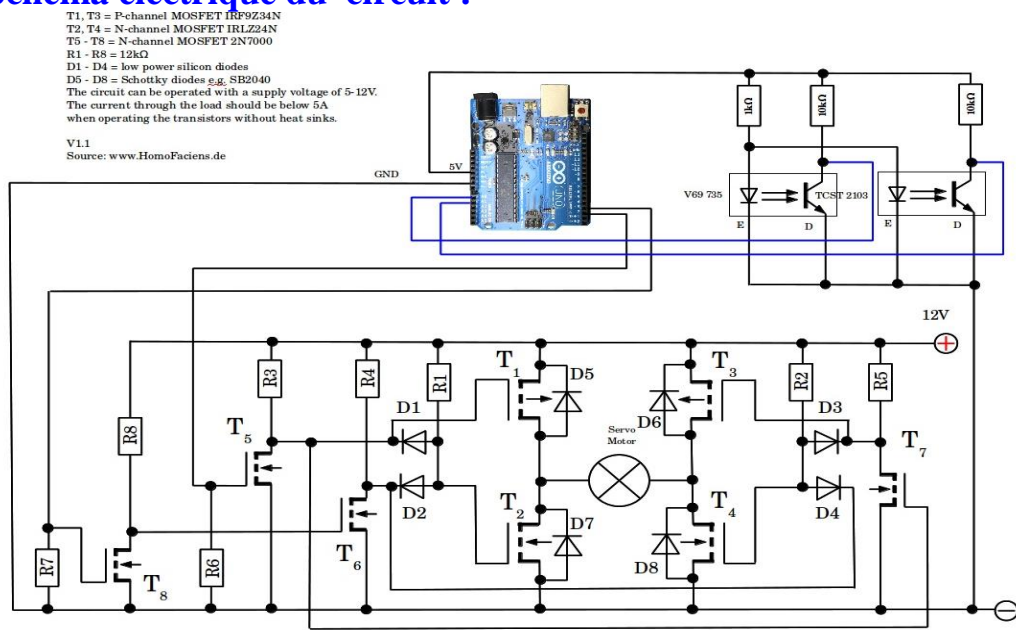


Figure III.6. Circuit électrique de projet

### III.6. La réalisation virtuelle « PORTEUS »

Pour la réalisation pratique nous utilisons ISIS-PORTEUS est un CAO électronique perfectionné conçu par L'absent Electronics qui permet de dessiner des schémas électroniques, de les simuler et de réaliser le circuit imprimé correspondant. Le CAO électronique « PROTEUS » se compose de nombreux outils regroupés en modules au sein d'une interface unique.

La figure III.5 Schéma de simulation montre la simulation de la carte électronique.



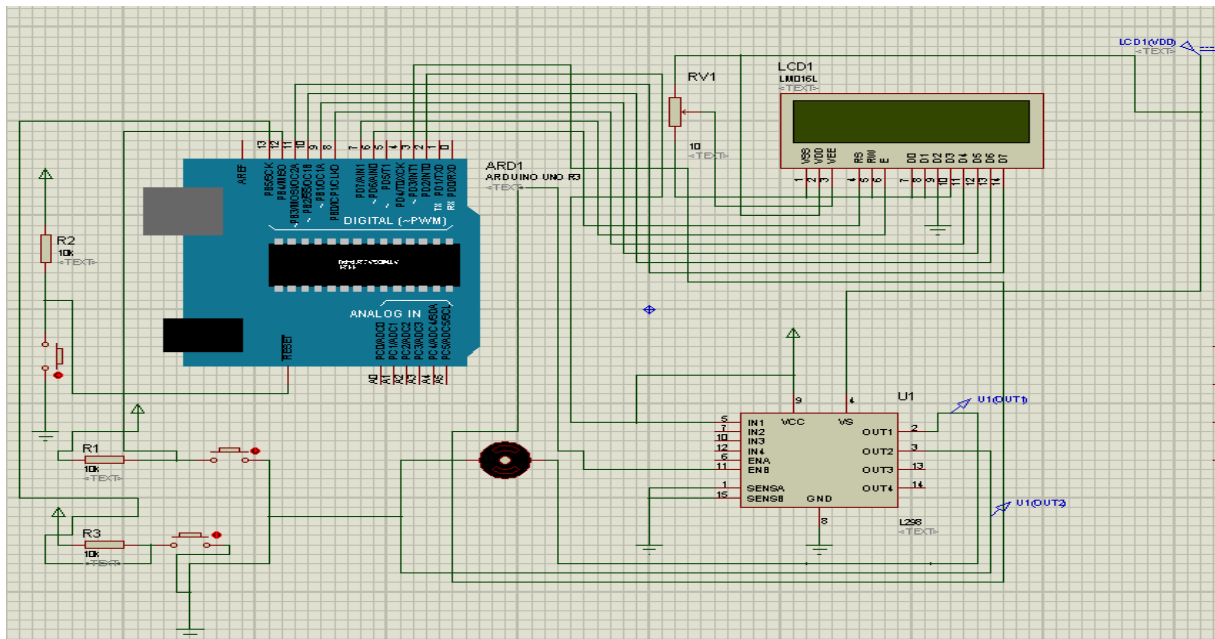


Figure III.7. Schéma de simulation

### III.7. Présentation de l'organigramme

L'organigramme est une représentation schématique des liens fonctionnels, organisationnels et hiérarchiques d'un organisme, d'un programme,

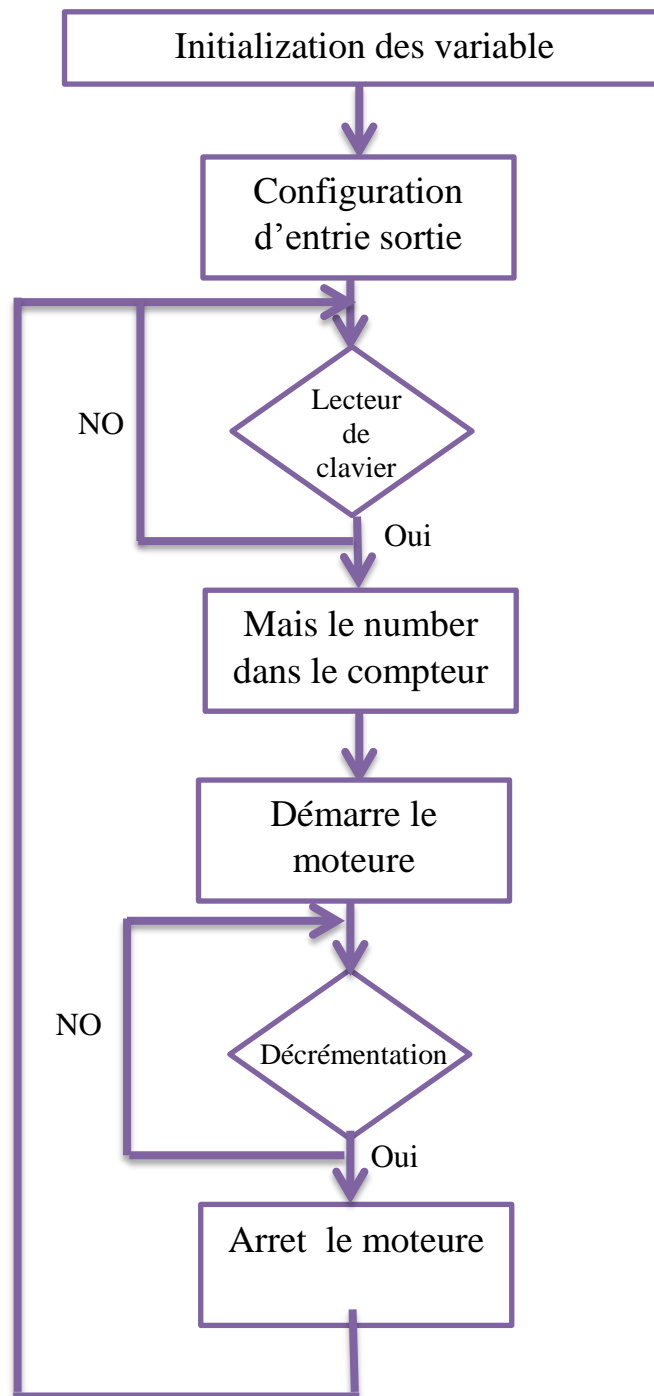
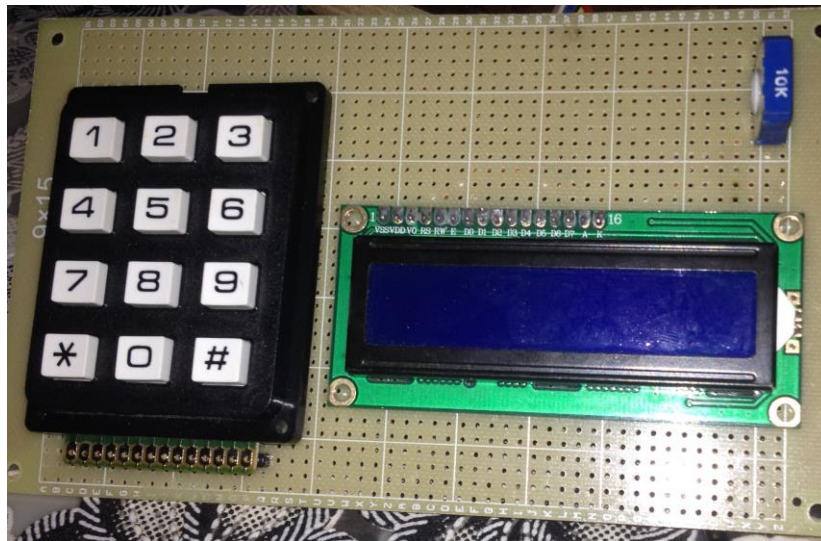
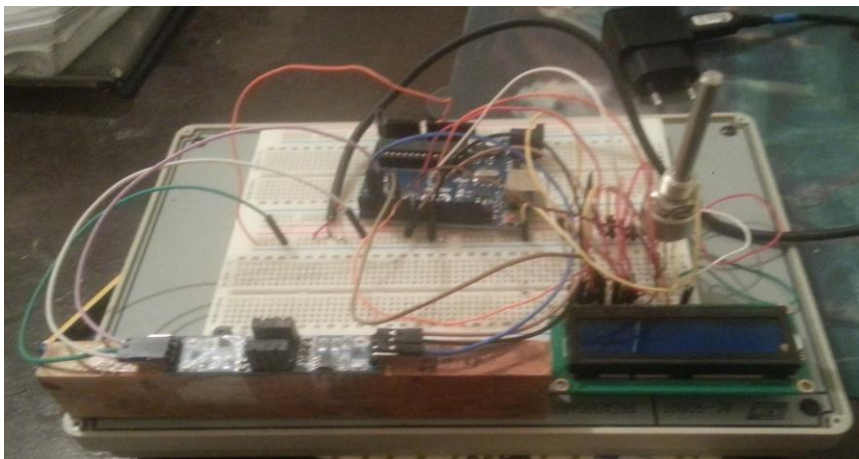


Figure III.6. Algorithme

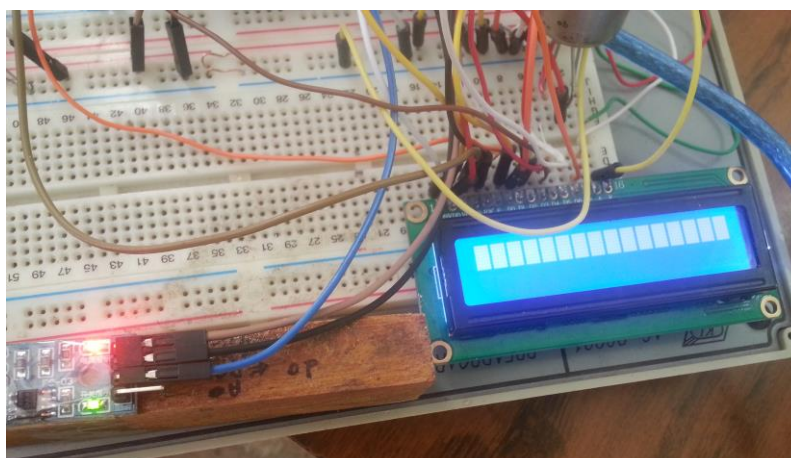
Après l'expérience dans le laboratoire on a les photos suivantes :



Clavier et afficheur LCD utilisé dans le projet



Montage sur la plaque de test



Montage sur la plaque de test

### **III.8. Conclusion**

*Sur le plan pratique, une manipulation adéquate du logiciel « Arduino » nous permet d'utiliser un compilateur ; il s'agit de « IDE ». Ce dernier possède une capacité de créer un code HEX, qui est injecté sur son microcontrôleur d'une part.*

*Le moteur fait tourner le disque selon la valeur saisie par le clavier, et un message est affiché sur le LCD.*

*On peut conclure que les résultats obtenus sont satisfaisants compte tenu des limitations du matériel et des moyens dont nous disposons.*

## **Conclusion Générale**

*Durant notre projet, nous avons vu que la commande PID appliquée aux moteurs électriques à courant continu était un régulateur qui pouvait offrir de bonnes performances, que ce soit au niveau de la précision, du temps de réaction, du dépassement, ou de la stabilité.*

*Diverses méthodes permettent d'optimiser les coefficients du PID, mais le régulateur idéal n'existe pas, chaque optimisation d'un paramètre (précision, ...) se fait aux dépens d'un autre. Il convient alors de trouver le meilleur compromis en fonction des exigences initiales du cahier de charge.*

*Divers éléments, tels que le choix de la méthode d'implantation du PID, peuvent aussi avoir des conséquences sur les performances. L'essentiel est alors de s'éloigner le moins possible du modèle théorique pour qu'il reste valable.*

*Le système Arduino est un outil pour fabriquer des dispositifs qui peuvent capter et contrôler davantage de choses du monde matériel que votre ordinateur. C'est une plateforme open-source d'électronique programmée qui est basée sur une simple carte à microcontrôleur (de la famille AVR), et un logiciel véritable environnement de développement intégré, pour écrire, compiler et transférer le programme vers la carte à microcontrôleur.*

*La programmation du microcontrôleur AVR exige une connaissance en langage « C » comme soft et la maîtrise des outils « IDE », afin d'arriver à l'objectif souhaité.*

*Une telle réalisation n'est pas dénuée de difficultés. Il est à noter que nous nous sommes confrontés à plusieurs problèmes surtout dans la partie réalisation de la carte.*

*Cependant, on peut dire que malgré ces difficultés, les résultats obtenus à travers cette étude qu'ils soient pratiques ou théoriques, permettent d'ouvrir la porte à d'autres études.*



## Bibliographie

- [1]P. Rousseau «Asservissement linéaires continus», Ellipses, Paris, 2004.
- [2]H.Egon, M.Marie, P.Porée «Traitement du signal et automatique», Hermann, Paris, 2000.
- [3]P. Clerc «Automatique Continue & Echantillonnée», Masson, Paris, 1996.
- [4]P. Barret «Machines Electriques», Ellipses, Paris, 2002.
- [5]B. Deforge & Q. David «Asservissement en position d'un axe linéaire», projet d'automatique, 2008.
- [6]C. le Lann «Le PID utilisé en régulation de position et/ou de vitesse de moteurs électriques», Projet de fin d'étude, 2007.
- [7]L. Couffignal «Le moteur à courant continu», cours, 2006.
- [8] M. A. Johnson & Mohammed H.Moradi «PID Control (New identification and design methods) », Springer, 2005
- [9]K. J. Astrom & T. Hagglund «Advanced PID Control», ISA, New York, 2006.
- [01] C. Tavernier, « Arduino applications avancées ». Version Dunod.
- [02] <http://www.acm.uiuc.edu/sigbot/tutorials/2009-11-17-arduino-basics>. consulter le: mars 2015.
- [03] <http://www.generationrobots.com/fr/152-arduino>. consulter le: mars 2015
- [04] S.V.D.Reyvanth, G.Shirish, « PID controller using Arduino ».
- [07] Jean- Noël, « livret Arduino en français », centre de ressources art sensitif
- [20] X.HINAULT. [www.mon-club-elec.fr](http://www.mon-club-elec.fr).



# LISTE DES FIGURES

<b>Numéro</b>	<b>Titre</b>	<b>Page</b>
Figure. I.1	Schéma d'un moteur à courant continu	<b>4</b>
Figure. I.2	Schéma équivalent d'un moteur à courant continu	<b>5</b>
Figure. I.3	Schéma bloc du Modèle de moteur électrique en vitesse	<b>6</b>
Figure. I.4	Modélisation sous Matlab de la réponse de vitesse	<b>6</b>
Figure. I.5	Schéma bloc du Modèle de moteur électrique en position	<b>7</b>
Figure. I.6	Modélisation sous Matlab de la réponse de position	<b>7</b>
Figure. I.7	Modélisation électrique d'un Moteur à excitation séparée	<b>8</b>
Figure. I.8	Modélisation électrique d'un Moteur à excitation shunt	<b>8</b>
Figure. I.9	Modélisation électrique d'un Moteur à excitation série	<b>8</b>
Figure. I.10	Modélisation électrique d'un Moteur à excitation compound	<b>9</b>
Figure. I.11	Schéma bloc d'un système en boucle ouverte	<b>9</b>
Figure. I.12	Schéma bloc d'un système en boucle fermée avec un retour unitaire	<b>10</b>
Figure. I.13	Schéma bloc d'un asservissement avec une perturbation (Retour unitaire)	<b>11</b>
Figure. I.14	Schéma bloc d'un système avec correcteur (Retour unitaire)	<b>12</b>
Figure. I.15	Schéma bloc d'un système avec correcteur (Boucle fermée)	<b>13</b>
Figure. I.16	Schéma bloc du correcteur PID en régulation de vitesse	<b>13</b>
Figure. I.17	Schéma bloc du correcteur PID en régulation de position	<b>14</b>
Figure II.1	La carte Arduino UNO	<b>18</b>
Figure II.2	Microcontrôleur ATmega328	<b>19</b>
Figure II.3	Constitution de la carte Arduino UNO	<b>22</b>
Figure II.4	Interface IDE Arduino	<b>23</b>
Figure II.5	Paramétrage de la carte	<b>23</b>
Figure II.6	Les étapes de téléchargement du code	<b>25</b>
Figure II.7	Shield H Bridge L298	<b>26</b>
Figure II.8	Schéma fonctionnel d'un LCD	<b>27</b>
Figure II.9	Photo d'un LCD et son brochage	<b>27</b>
Figure II.10	Brochage d'un Afficheur LCD avec Arduino	<b>28</b>
Figure II.11	Brochage Clavier numérique électronique avec Arduino	<b>29</b>
Figure II.12	Photo Encodeur optique	<b>30</b>

Figure III.1	Schéma synoptique du dispositif	<b>32</b>
Figure III.2	Schéma synoptique de l'alimentation	<b>33</b>
Figure III.3	Alimentation utilisée dans le projet	<b>34</b>
Figure III.4	Schéma de brochage du composant	<b>35</b>
Figure III.5	Disque pour l'encodeur optique (mesure de position de moteur)	<b>36</b>
Figure III.6	Circuit électrique	<b>36</b>
Figure III.7	Simulation de simulation	<b>37</b>
Figure III.8	Algorithme	<b>38</b>