

# TABLE DES MATIÈRES

|  |     |
|--|-----|
| DEDICACES 1  | i   |
| DEDICACES 2  | ii  |
| REMERCIEMENTS  | iii |
| NOMENCLATURE   | ix  |
| GLOSSAIRE  | x   |
| INTRODUCTION GÉNÉRALE  | 1   |
| I LES MOTEURS PAS À PAS                                      | 3   |
| I.1 INTRODUCTION   | 4   |
| I.2 TECHNOLOGIE DES MOTEURS PAS À PAS                        | 4   |
| I.2.1 Définition   | 5   |
| I.2.2 Applications des moteurs pas à pas                     | 6   |
| I.2.3 Principaux types de moteurs pas à pas                  | 6   |
| I.2.3-a Moteurs à aimant permanent (MP)                      | 6   |
| I.2.3-b Moteurs hybrides (MPPH)                              | 7   |
| I.2.3-c Moteurs pas à pas à réluctance variable (MRV)        | 8   |
| I.2.4 Comparaison des différents types                       | 9   |
| I.3 MODES DE CONTRÔLE DES MOTEURS PAS À PAS                  | 9   |
| I.3.1 Mode pas entier Full-step                              | 10  |
| I.3.2 Mode demi-pas Half-step                                | 11  |
| I.3.3 Mode micro pas Micro-step                              | 11  |
| I.4 COMPORTEMENT DYNAMIQUE DES MOTEURS PAS À PAS             | 12  |
| I.4.1 Couple et vitesse                                      | 12  |
| I.4.2 Couple d'arrêt ou couple de maintien                   | 13  |
| I.4.3 Plage de démarrage                                     | 13  |
| I.4.4 Fréquence limite de démarrage                          | 14  |
| I.4.5 Plage d'accélération                                   | 14  |
| I.4.6 Couple limite de travail                               | 14  |
| I.4.7 Fréquence maximale des pas                             | 14  |
| I.4.8 Comportement dynamique du moteur                       | 14  |
| I.5 CONFIGURATIONS INTERNES DES BOBINES <a href="#">[18]</a> | 15  |

|           |   |           |
|-----------|---|-----------|
| I.5.1     | Moteur à 4 fils (bipolaire)                                     | 15        |
| I.5.2     | Moteur à 5 fils (unipolaire)                                    | 16        |
| I.6       | COMMANDE DES MOTEURS PAS À PAS                                  | 16        |
| I.7       | CIRCUITS INTÉGRÉS DÉDIÉS À LA COMMANDE DES MOTEURS<br>PAS À PAS | 17        |
| I.7.1     | Commande par SAA1027  | 17        |
| I.7.2     | Commande par UCN5804B   | 18        |
| I.7.3     | Commande basées sur les par L297 et L298                        | 19        |
| I.7.3-a   | Le circuit L297   | 19        |
| I.7.3-b   | Le circuit L298   | 20        |
| I.8       | CRITÈRE DE CHOIX D'UN MOTEUR PAS À PAS                          | 21        |
| I.9       | CONCLUSION  | 22        |
| II        | PRÉSENTATION DE LA CARTE DE PROTOTYPAGE ARDUINO                 | <b>23</b> |
| II.1      | HISTORIQUE DU PROJET ARDUINO                                    | 24        |
| II.1.1    | Le logiciel   | 25        |
| II.1.2    | Le matériel   | 25        |
| II.2      | CARTE ÉLECTRONIQUE  | 26        |
| II.3      | QUELQUES OUTILS UTILISÉS AVEC ARDUINO                           | 27        |
| II.3.1    | Processing  | 27        |
| II.3.2    | Pure Data   | 27        |
| II.4      | BUT DE LA CARTE ARDUINO   | 28        |
| II.5      | LES APPLICATIONS POSSIBLES                                      | 29        |
| II.6      | LES DIFFÉRENTS TYPES DE CARTES                                  | 29        |
| II.6.0-a  | La carte Uno et Duemilanove                                     | 29        |
| II.6.0-b  | La carte Megae  | 31        |
| II.7      | LES «SHIELDS»   | 33        |
| II.8      | CONCLUSION :  | 34        |
| III       | COMMANDE DE MOTEURS PAS À PAS BASÉES SUR LE MO-<br>DULE ARDUINO | <b>35</b> |
| III.1     | INTRODUCTION  | 36        |
| III.2     | PRÉSENTATION DU PROJET  | 36        |
| III.3     | SPÉCIFICATIONS FONCTIONNELLES                                   | 36        |
| III.3.1   | Le circuit de commande  | 36        |
| III.3.2   | Le circuit de puissance   | 37        |
| III.4     | TESTS ET VALIDATION   | 37        |
| III.4.1   | Réalisation du circuit  | 37        |
| III.4.1-a | Carte de commande   | 38        |
| III.4.1-b | Carte de puissance  | 38        |
| III.4.2   | Réalisation des typons  | 38        |
| III.4.3   | Développement de la carte                                       | 42        |
| III.4.3-a | Partie simulation   | 42        |
| III.4.3-b | Partie réalisation des circuit imprimé                          | 42        |
| III.4.4   | Schéma bloc   | 42        |

---

|           |   |    |
|-----------|---|----|
| III.4.5   | Commande d'un seul moteur . . . . .                   | 44 |
| III.4.5-a | Généralisation des impulsions de la commande          | 44 |
| III.4.6   | Commande de deux moteurs pas à pas avec afficheur LCD | 47 |
| III.4.7   | Programmation du microcontrôleur . . . . .            | 49 |
| III.4.7-a | Programme . . . . .                                   | 49 |
| III.5     | CONCLUSION . . . . .                                  | 57 |
|           | CONCLUSION GÉNÉRALE                                   | 58 |
|           | BIBLIOGRAPHIE   | 61 |
|           | ANNEXE A  | 1  |
|           | ANNEXE B  | 4  |

# TABLE DES FIGURES

|       |  |    |
|-------|--|----|
| I.1   | Moteurs pas à pas . . . . .  | 5  |
| I.2   | Moteur à aimant permanent . . . . .  | 7  |
| I.3   | Moteur hybride . . . . .   | 8  |
| I.4   | Moteur à réluctance variable . . . . .   | 9  |
| I.5   | Première cas de séquence en Mode pas entier . . . . .  | 10 |
| I.6   | Deuxième cas de séquence en Mode pas entier . . . . .  | 10 |
| I.7   | Mode pas entier . . . . .  | 11 |
| I.8   | Séquence en Mode demi-pas . . . . .  | 12 |
| I.9   | Mode demi-pas . . . . .  | 12 |
| I.10  | Mode micro pas . . . . .   | 13 |
| I.11  | Caractéristiques Couple-fréquence . . . . .  | 14 |
| I.12  | La commande du moteur bipolaire . . . . .  | 15 |
| I.13  | La commande du moteur unipolaire . . . . .   | 16 |
| I.14  | Schéma de principe de fonctionnement de UCN5804B . . . . .   | 18 |
| I.15  | Séquence basique . . . . .   | 20 |
| I.16  | Séquence basique . . . . .   | 21 |
| II.1  | Logiciel de programmation Arduino (IDE) . . . . .  | 25 |
| II.2  | Carte Arduino « Uno » . . . . .  | 26 |
| II.3  | Exemple de schéma électronique de la carte Arduino Uno . . . . .   | 27 |
| II.4  | Exemple de typon de la carte Arduino . . . . .   | 28 |
| II.5  | Arduino UNO . . . . .  | 30 |
| II.6  | Microcontrôleur ATmega328 . . . . .  | 30 |
| II.7  | Arduino Mega 2560 . . . . .  | 31 |
| II.8  | Exemple de Shield . . . . .  | 34 |
| III.1 | Schéma de l'association entre les deux cartes . . . . .  | 38 |
| III.2 | Typon de la carte de commande et de puissance . . . . .  | 39 |
| III.3 | Typon du circuit adapté avec arduino mega 2560 . . . . .   | 40 |
| III.4 | Typon du clavier de manipulation . . . . .   | 41 |
| III.5 | Allures des tensions A, B, C et D à la sortie du circuit de<br>puissance destiné à contrôler un moteur pas à pas . . . . . | 42 |
| III.6 | Montage de simulation à l'aide d'une orloge à la place d'arduino . . . . .   | 43 |
| III.7 | Simulation de la carte de commande d'un moteur pas à pas . . . . .   | 43 |
| III.8 | Informations du menu principale du projet à afficher sur l'écran<br>LCD . . . . .  | 44 |
| III.9 | Le reste des information à afficher . . . . .  | 45 |

---

|        |   |    |
|--------|---|----|
| III.10 | Circuit imprimé de carte de commande et de puissance . . . .  | 45 |
| III.11 | Clavier de minupilation . . . . .   | 46 |
| III.12 | Circuit imprimé adapté avec arduino mega 2560 . . . . .   | 46 |
| III.13 | Schéma bloc simplifié du système . . . . .  | 47 |
| III.14 | Allures des tensions entre une borne de chaque enroulement<br>d'un moteur pas à pas et le Gnd . . . . . | 48 |
| III.15 | Alures des tensions aux bornes des deux enroulement d'un<br>moteur pas à pas bipolaire . . . . .        | 48 |
| A1     | L297 . . . . .  | 1  |
| A.2    | L298 . . . . .  | 2  |
| A.3    | Two Phase Bipolar Stepper Motor Circuit . . . . .   | 3  |

---

## Nomenclature

Les principales notations et abréviations utilisées dans ce mémoire sont explicitées ci-dessous, sous leur forme la plus couramment employée dans le domaine du génie électrique.

### Grandeurs électriques et mécaniques

| Nom                  | Symbole  | Unité  |
|----------------------|----------|--------|
| Temps                | $t$      | s      |
| Tension              | $U$      | V      |
| Courant              | $I$      | A      |
| Puissance            | $P$      | W      |
| Flux magnétique      | $\phi$   | Wb     |
| Induction magnétique | $B$      | T      |
| Fréquence            | $f$      | Hz     |
| Moment d'un couple   | $C$      | mN     |
| Angle (position)     | $\theta$ | °, rad |

### Paramètres

| Nom        | Symbole | Unité    |
|------------|---------|----------|
| Résistance | $R$     | $\Omega$ |
| Inductance | $L$     | H        |

---

## Glossaire

| Acronyme     | Signification   |
|--------------|---|
| MPP          | Moteur pas a pas  |
| MPPAP        | Moteur pas à pas à aimant permanent   |
| MPPH         | Moteur pas a pas hybride  |
| MRV          | Moteur pas à pas à réluctance variable  |
| VR           | Variable reluctance motor   |
| EEPROM       | le disque du microcontrôleurs   |
| RAM          | mémoire vive  |
| ICSP         | Serial Programming In-Circuit   |
| UART         | Universal Asynchronous Receiver Transmitter (un émetteur-récepteur asynchrone universel.) |
| (Rx) et (TX) | Il communique sur les repères numériques 0 et 1   |
| GND          | Les broches sol   |

---

## Introduction générale

Le moteur pas à pas a pris une très grande place dans le monde de la micro-informatique, il est actuellement le principal élément intermédiaire entre les dispositifs d'information et le monde électromécanique extérieur. Par ailleurs sa capacité à contrôler la position et la vitesse, par un train d'impulsion de commande assure à ce convertisseur des applications telle que :

- le déplacement de tête de lecture d'ordinateurs,
- l'avance de tête d'impression et de papier pour imprimantes,
- la commande d'axes pour les tables traçantes,
- la traction de robot mobile,
- l'indexage rotatif ou linéaire.

**Objectif du projet** Le but de notre travail est d'une part d'étudier le fonctionnement du moteur pas à pas d'autre part de commander ce convertisseur électromagnétique avec un dispositif de contrôle numérique et de mettre en œuvre différentes cartes de commande pour contrôler ce type de moteurs à l'aide d'une carte numérique de prototypage Arduino, en vue de les utiliser en travaux pratiques.

Dans sa version classique ce type de moteur est alimenté à partir d'une source continue, le contrôle de la vitesse et/ou de la position se fait en boucle ouverte. La commande en boucle ouverte constitue un des principaux avantages du point de vue économique (coût et fiabilité d'installation) mais aussi fonctionnel.

Toutefois ces moteurs présente un certain nombre d'inconvénients tels que :

- des instabilités de fonctionnement à certaines fréquences,
- des comportements non-linéaire,
- présence d'oscillations autour de la position voulue.



## **Organisation du mémoire**

Le travail présenté dans ce mémoire est organisé en trois chapitres :

### **Chapitre I**

Concernent les principaux types de moteurs pas à pas tels que les moteurs à réluctances variables, les moteurs à aimant permanent et les moteurs hybrides qui sont le fruit de la fusion des technologies précédentes. Les principes de fonctionnement, les caractéristiques techniques, le comportement dynamique de chacun d'entre eux, sont exposés ainsi que les différents types de circuits intégrés dédiés à leur commande.

### **Chapitre II**

Dédié à la présentation de la carte de commande Arduino Mega 2560, son architecture matérielle et sa programmation sont détaillées.

### **Chapitre III**

Consacré à la mise en œuvre des cartes de commande pour le contrôle d'un seul moteur pas à pas dans un premier temps, puis pour deux moteurs et un afficheur LCD. Enfin, la conception et la réalisation détaillées de la carte de commande sera décrite.

### **Conclusion**

Elle récapitule les principaux résultats obtenus dans ce projet et résume l'apport de ce travail.

## Chapitre I

# Les Moteurs pas à pas

### SOMMAIRE

|       |   |    |
|-------|---|----|
| I.1   | INTRODUCTION . . . . .  | 4  |
| I.2   | TECHNOLOGIE DES MOTEURS PAS À PAS . . . . .                               | 4  |
| I.2.1 | Définition . . . . .  | 5  |
| I.2.2 | Applications des moteurs pas à pas . . . . .                              | 6  |
| I.2.3 | Principaux types de moteurs pas à pas . . . . .                           | 6  |
| I.2.4 | Comparaison des différents types . . . . .                                | 9  |
| I.3   | MODES DE CONTRÔLE DES MOTEURS PAS À PAS . . . . .                         | 9  |
| I.3.1 | Mode pas entier Full-step . . . . .                                       | 10 |
| I.3.2 | Mode demi-pas Half-step . . . . .   | 11 |
| I.3.3 | Mode micro pas Micro-step . . . . .                                       | 11 |
| I.4   | COMPORTEMENT DYNAMIQUE DES MOTEURS PAS À PAS . . . . .                    | 12 |
| I.4.1 | Couple et vitesse . . . . .   | 12 |
| I.4.2 | Couple d'arrêt ou couple de maintien . . . . .                            | 13 |
| I.4.3 | Plage de démarrage . . . . .  | 13 |
| I.4.4 | Fréquence limite de démarrage . . . . .                                   | 14 |
| I.4.5 | Plage d'accélération . . . . .  | 14 |
| I.4.6 | Couple limite de travail . . . . .  | 14 |
| I.4.7 | Fréquence maximale des pas . . . . .                                      | 14 |
| I.4.8 | Comportement dynamique du moteur . . . . .                                | 14 |
| I.5   | CONFIGURATIONS INTERNES DES BOBINES [18] . . . . .                        | 15 |
| I.5.1 | Moteur à 4 fils (bipolaire) . . . . .                                     | 15 |
| I.5.2 | Moteur à 5 fils (unipolaire) . . . . .                                    | 16 |
| I.6   | COMMANDE DES MOTEURS PAS À PAS . . . . .                                  | 16 |
| I.7   | CIRCUITS INTÉGRÉS DÉDIÉS À LA COMMANDE DES MOTEURS<br>PAS À PAS . . . . . | 17 |
| I.7.1 | Commande par SAA1027 . . . . .  | 17 |
| I.7.2 | Commande par UCN5804B . . . . .   | 18 |
| I.7.3 | Commande basées sur les par L297 et L298 . . . . .                        | 19 |
| I.8   | CRITÈRE DE CHOIX D'UN MOTEUR PAS À PAS . . . . .                          | 21 |
| I.9   | CONCLUSION . . . . .  | 22 |

## I.1 Introduction

Aujourd'hui ans notre société moderne, les moteurs ont pris une place importante, ils sont devenus indispensables dans notre vie quotidienne. En effet, depuis l'imprimante jusqu' au robot ménager, les moteurs électriques sont présents.

*Le moteur pas à pas, est un petit pas pour l'homme, mais une grande révolution pour les électrotechniciens.* En effet, un moteur pas à pas, *Stepping Motor* en anglais, n'est rien d'autre qu'un moteur électrique d'une très grande précision qui peut atteindre des déplacements de quelques degrés seulement.

En effet, un moteur classique n'est pas assez précis pour certaines applications et nécessite des boucles de régulation pour améliorer la précision, ce qui est difficile à implanter et revient cher, par contre le MPP travaille en boucle ouverte ce qui simplifie sa commande. Dans le cas d'une imprimante par exemple, on doit pouvoir déplacer le rouleau qui guide le papier d'une même distance, et au millimètre près et cela à chaque fois.

Tout cela est bien beau, mais comme toujours rien n'est parfait et les moteurs pas à pas présentent aussi des inconvénients et nécessitent des circuits de commande très complexe, surtout si l'on désire utiliser des composants discrets. Par contre, l'emploi de circuits programmable facilite leur mise en œuvre.

L'objectif de ce projet consiste à réaliser des commandes de moteur pas à pas, simples et faciles à utiliser, à l'aide d'un module de prototypage Arduino. Dans ce chapitre, nous allons exposer le principe des différents types de moteurs pas à pas ainsi que les fondements de leurs commande, puis nous donnons une description des circuits intégrés spécialisés dédiés à leurs commande.

## I.2 Technologie des moteurs pas à pas

Les moteurs pas à pas sont des transducteurs électromécaniques qui assurent la conversion des signaux impulsions en mouvement de rotation de type incrémental. Ils sont utilisés dans de nombreuses applications liées aux systèmes informatiques ou de la robotique car ils permettent d'obtenir une très grande précision grâce à leur conception électrique et mécanique.

La conception du moteur pas à pas a été motivée à partir de deux démarches logiques très différentes :

- d'une part, le besoin d'un moteur capable de développer un couple important à faible vitesse, voire même à l'arrêt ;
- d'autre part, la nécessité d'un dispositif simple capable de convertir des informations de caractère discret.

Le convertisseur d'énergie à basse vitesse et le transmetteur de l'information ont donné naissance au moteur pas à pas moderne vers les années 1970, grâce au développement conjugué de l'électronique de puissance, et surtout grâce à l'apparition de l'électronique numérique à forte intégration. Ils nécessitent un circuit de commande qui comporte une partie logique et une partie puissance. La partie logique détermine pour chaque pas quelles sont les bobines à alimenter et le sens de rotation. La fréquence de l'horloge du circuit logique détermine la vitesse de rotation.



FIGURE I.1 – Moteurs pas à pas

### I.2.1 Définition

Les moteurs pas à pas sont des moteurs électriques d'une conception particulière, différentes des moteurs classiques. Leur structure permet, par une commande électronique appropriée, d'obtenir une rotation du rotor d'un angle égal à  $n$  fois un angle élémentaire appelé « pas » [11] .

Il fut inventé en 1936 par Marius Lavet, un ingénieur français des Arts et Métiers, pour l'industrie horlogère[17]. Leur véritable développement est lié à l'avènement de la micro-informatique (microprocesseur).

Le moteur pas à pas peut effectuer deux fonctions :

- Convertir l'information numérique en un positionnement angulaire.
- Convertir l'énergie électrique en énergie mécanique.

Le caractère synchrone du MPP lui permet de fonctionner sans la boucle de retour mais par contre une meilleure fiabilité impose un contrôle de la position, donc un asservissement.

### I.2.2 Applications des moteurs pas à pas

Ce type de moteur est pratiquement utilisé dans du positionnement en boucle ouverte, c'est-à-dire sans détecteur de position (codeur, résolveur, potentiomètre, etc.) ou tous les dispositifs où l'on souhaite faire du contrôle de vitesse.

L'usage le plus connu du grand public est dans les imprimantes, les scanner et les platines vinyle de DJ. Mais ils sont présents dans de nombreuses applications telles : les photocopieurs, imprimantes bancaires, robotique, instrumentation, pompes à perfusion, pousse-seringues (en), système de positionnement sur machine industrielle et machine-outil [17].

### I.2.3 Principaux types de moteurs pas à pas

Le MPP est comme tous les autres moteurs il comporte un Stator et un Rotor. Le Rotor est à aimant permanent, et le Stator est constitué d'un certain nombre de bobines d'excitation. Le schéma ci-dessous est une représentation simpliste du moteur. Et Plus le nombre de bobines et d'aimants augmente, plus le nombre de pas sera grand, pour faire un tour et plus faible seront les vibrations.

#### I.2.3-a Moteurs à aimant permanent (MP)

Ce moteur possède quatre fils d'alimentation pour piloter des bobines par paire. Le fonctionnement des moteurs à aimant permanents est assuré par l'action d'un champ électromagnétique statorique sur un rotor composé d'aimant permanents.

- Le stator présente un certain nombre de dents ayant un bobinage,
- Le rotor est un aimant qui s'alignera avec les pôles du stator qui sont alimentés [3],

#### Avantages du moteur à aimant permanent :

- Bon marché
- Dimensions réduites
- Bon rendement
- Bon amortissement des oscillations
- Grand angle de pas (nombre de pas faible : 48)

#### Inconvénients du moteur à aimant permanent [11] :

- Puissance faible
- Paliers en bronze ou plastique (pas de roulement)
- Couple résiduel sans courant
- Vitesse faible

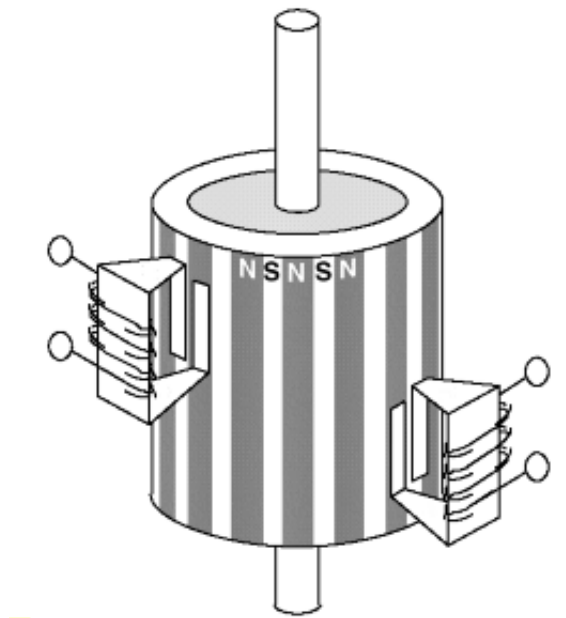


FIGURE I.2 – Moteur à aimant permanent

### I.2.3-b Moteurs hybrides (MPPH)

Les moteurs hybrides sont généralement constitués d'un rotor denté muni d'aimants permanents. le rotor possède généralement deux disques polaires décalés d'un angle électrique ?

Les moteurs pas à pas hybrides réunissent, au moins en partie, les avantages des moteurs pas à pas à réluctance variable et à aimants permanents, à savoir :

- un grand nombre de pas par tour,
- une fréquence propre mécanique importante,
- un couple massique élevé,
- un amortissement interne important,
- une mémoire de position.

Dans sa configuration de base le moteur pas à pas hybride comporte un stator en fer feuilleté à plots saillants et deux couronnes rotoriques dentées en matériau ferromagnétique, géométriquement identiques et réunies par un aimant permanent cylindrique magnétisé axialement. Les lignes de champs de l'aimant se ferment à travers les dents du rotor. Vu du stator, le rotor présente autant de pôles magnétiques actifs qu'il possède de dents. Les dents sur une des couronnes sont décalées par rapport aux dents de l'autre d'un demi pas dentaire [2].

#### Avantages du moteur pas à pas hybride :

- Couple important
- Plus de puissance
- Rendement assez bon

- Courbe start/stop assez élevée
- Bon amortissement
- Adapté au fonctionnement micro pas
- Roulement à billes pour une meilleure charge radiale et durée de vie
- Petit angle de pas.

**Inconvénients du moteur pas à pas hybride [11] :**

- Inertie élevée
- Couple résiduel sans courant
- Plus coûteux
- Plus volumineux

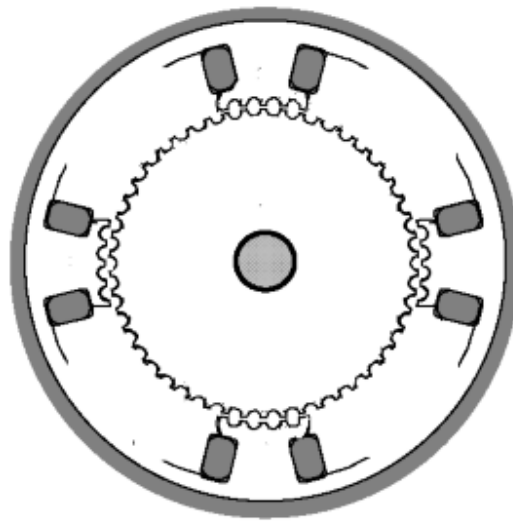


FIGURE I.3 – Moteur hybride

**I.2.3-c Moteurs pas à pas à réluctance variable (MRV)**

Le moteur pas-à-pas réluctant (Variable reluctance motor VRM) comporte  $nb$  bobines distinctes au stator et un rotor ferromagnétique sans aimant ( $n_a=0$ ). Sans courant dans les bobinages statoriques, ce type de moteur ne présente pas de couple réluctant.

Les circuits magnétiques du rotor et du stator sont assemblés à partir de tôles magnétiques de haute perméabilité (fer-silicium ou même fer-cobalt). Certaines machines, destinées au positionnement ou à une rotation très lente, peuvent avoir un rotor ou/et un stator en fer massif. Il existe un grand nombre d'astuces de construction pour augmenter le nombre de pas par tour, comme par exemple les structures bifilaires qui ne seront pas étudiées dans ce cours.

Pour minimiser la force magnétomotrice et augmenter la puissance massique, l'entrefer de ce type de moteurs à réluctance variable doit être aussi petit que possible.

Plusieurs géométries du rotor et du stator ont été ainsi imaginées par les constructeurs pour remplir des cahiers des charges particuliers. On distingue trois géométries de base, à savoir les moteurs pas à pas à réluctance variable [2] :

- à plots statoriques non dentés,
- à plots statoriques dentés,
- à réluctance variable à plots statoriques non denté décalés.

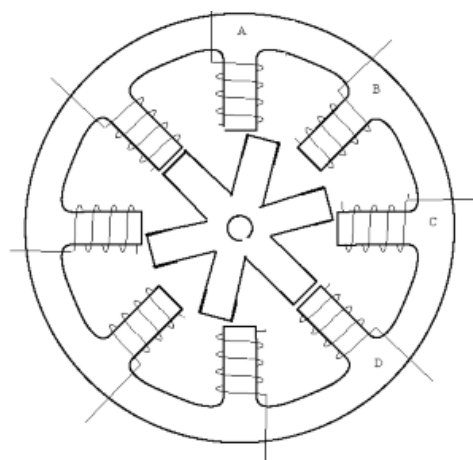


FIGURE I.4 – Moteur à réluctance variable

#### Avantages

- Grande vitesse de fonctionnement,
- Pas réduit donc bonne résolution le nombre de pas peut être compris entre 12 et 72 au tour.

#### I.2.4 Comparaison des différents types

| Type de MPP      | MRV        | MPPA                   | MPPH        |
|------------------|------------|------------------------|-------------|
| Nbre de pas/tour | bonne      | moyenne                | élevée      |
| Couple moteur    | faible     | élevée                 | élevée      |
| Fréquence        | grande     | faible                 | grande      |
| Puissance        | quelques W | quelques dizaines de W | quelques kW |

TABLE I.1 – Comparaison entre les différents types de moteur pas à pas

### I.3 Modes de contrôle des moteurs pas à pas

les moteurs se divisent en deux types : les bipolaires et les unipolaires dans ce cas ont choisi d'étudier les séquence de commande du moteur bipolaire. Le courant de commande est bidirectionnel et l'avance d'un pas s'effectue par une séquence de commutation des enroulements statoriques.



### I.3.1 Mode pas entier Full-step

Pour ce type de moteur nous avons trois possibilités de commande. La première consiste à alimenter les enroulements en suivant la séquence A vers B / C vers D / B vers A / D vers C (BA est le même enroulement que AB mais alimenté par un courant de polarité inverse).

Par la suite nous simplifierons la notation pour une meilleure correspondance avec les chronogrammes des phases en indiquant uniquement la phase qui est alimentée par un courant positif. Soit : A C B D. Cette séquence est connue sous le nom de "One phase on full step" (traduisez phase par phase ou une phase à la fois en pas entier). A tout moment il n'y a qu'une seule phase d'alimentée et nous travaillons en mode pas entier.

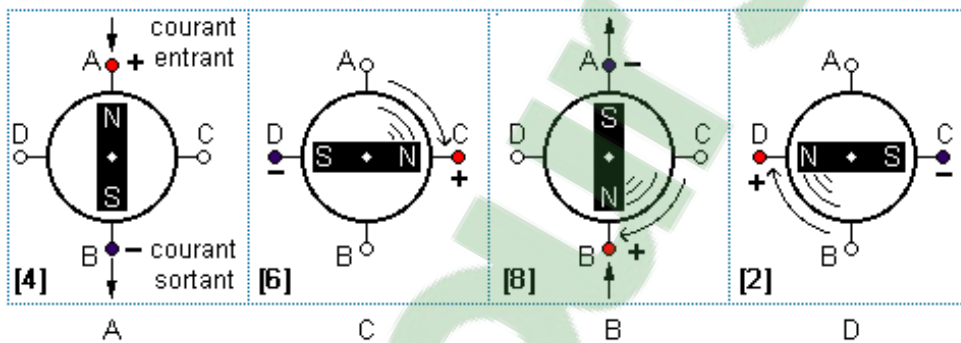


FIGURE I.5 – Première cas de séquence en Mode pas entier

La seconde possibilité est d'alimenter une paire de phase en même temps de façon à ce que le rotor se positionne entre deux pôles. Appelé "Two-phase-on full step" (deux phases à la fois en pas entier) ce mode de commande est celui qui procure le couple le plus élevé. La séquence sera donc : AC / CB / BD / DA Fig6 [15] .

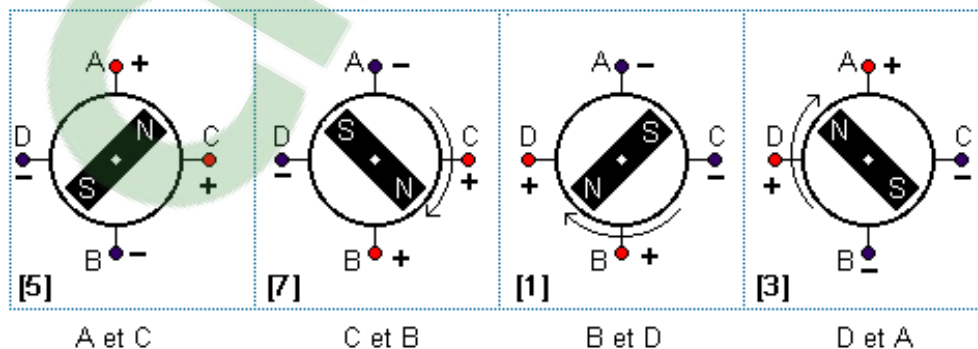


FIGURE I.6 – Deuxième cas de séquence en Mode pas entier

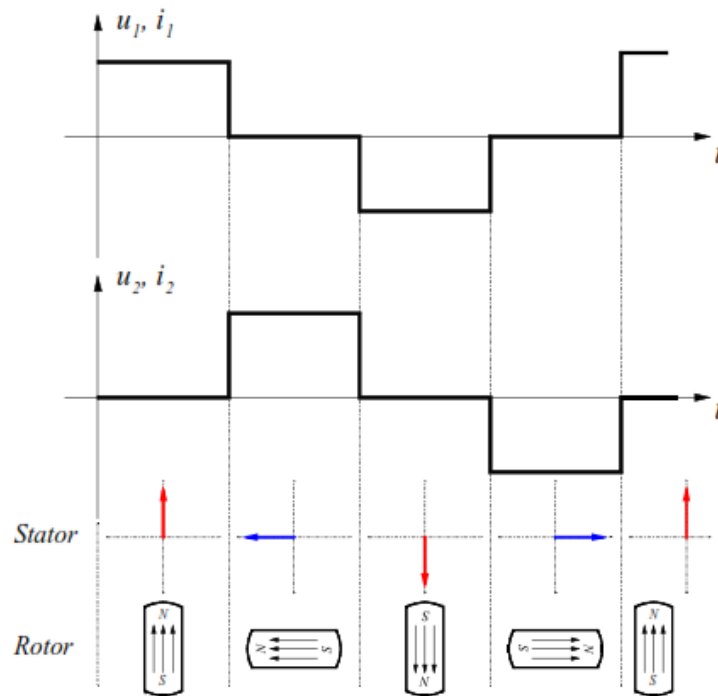


FIGURE I.7 – Mode pas entier

### I.3.2 Mode demi-pas Half-step

La troisième option est un mélange des deux première puisque l'on alimente tour à tour le moteur sur une phase puis deux puis une Cette séquence connue sous le nom de mode demi pas procure effectivement un division par 2 de l'angle d'avance d'un pas, mais aussi un couple moins régulier. La séquence qui en découle est la suivante : A / AC / C / CB / B / BD / D /DA. Fig8

Pour obtenir une rotation dans la direction opposée les mêmes séquences sont utilisée en inversant l'ordre de défilement. Comme nous pouvons le constater sur les diagrammes, les moteurs sont représentés avec une avance de pas à  $90^\circ$ . Dans les moteurs réels le nombres de pôles ont été multipliés pour réduire à quelques degrés seulement l'angle d'avance d'un pas. Le nombre d'enroulements et la séquence de commande restant, quand à eux, inchangés[15].

### I.3.3 Mode micro pas Micro-step

Des alimentations plus sophistiqués encore permettent de moduler l'amplitude du courant dans chaque phase, dans une relation sinus cosinus. Par combinaison des champs magnétiques produits par ces courants dans les enroulements", il est ainsi possible d'immobiliser le rotor dans une multitude de pas intermédiaires. On parle alors d'un système micro pas (« micro-step » en anglais). Tout se passe comme si le moteur avait encore plus de pas.

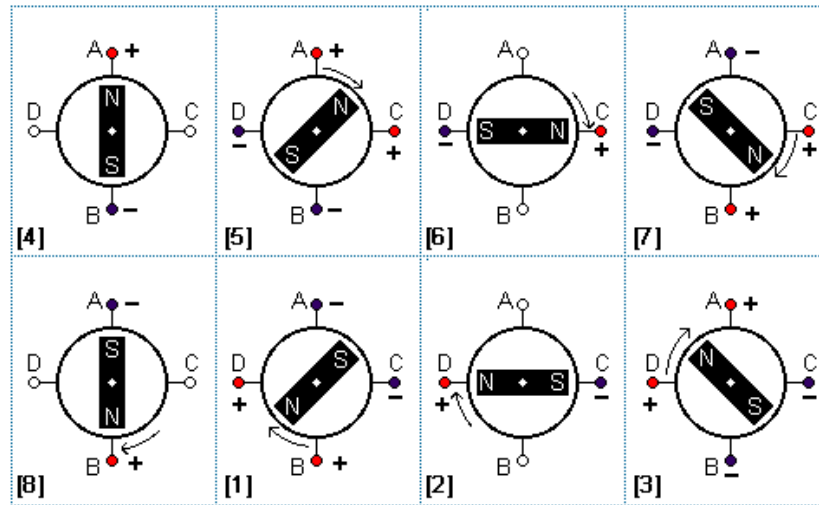


FIGURE I.8 – Séquence en Mode demi-pas

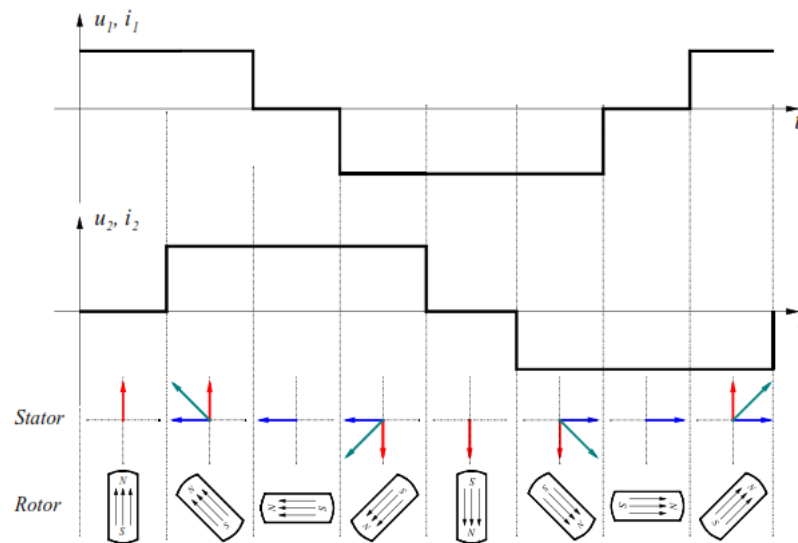


FIGURE I.9 – Mode demi-pas

## I.4 Comportement dynamique des moteurs pas à pas

### I.4.1 Couple et vitesse

Le couple du moteur dépend de l'intensité traversant les bobines de sa position angulaire à un instant donné quand il passe d'une bobine à l'autre de la vitesse de pilotage. On définit une vitesse limite au démarrage et une vitesse limite de fonctionnement, vitesses au-delà desquelles le moteur décroche.

Cette limitation de vitesse est surtout due à l'effet de self des bobines.

- la valeur du couple est en général donnée pour 5 pas à la seconde
- le couple de maintien est le couple mesuré à l'arrêt, les bobines étant alimentées.

- le couple de détente est le couple quand les bobines ne sont pas alimentées.
- le positionnement angulaire présente une oscillation à chaque pas.  
Le rotor dépasse la position souhaitée puis revient en oscillant de part et d'autre de la position. Ce défaut est gênant à certaines vitesses.

En astronomie, le moteur doit tourner à 2 vitesses pour permettre le suivi des étoiles et se positionner rapidement sur un astre. Le fonctionnement rapide doit se trouver dans la zone de démarrage.

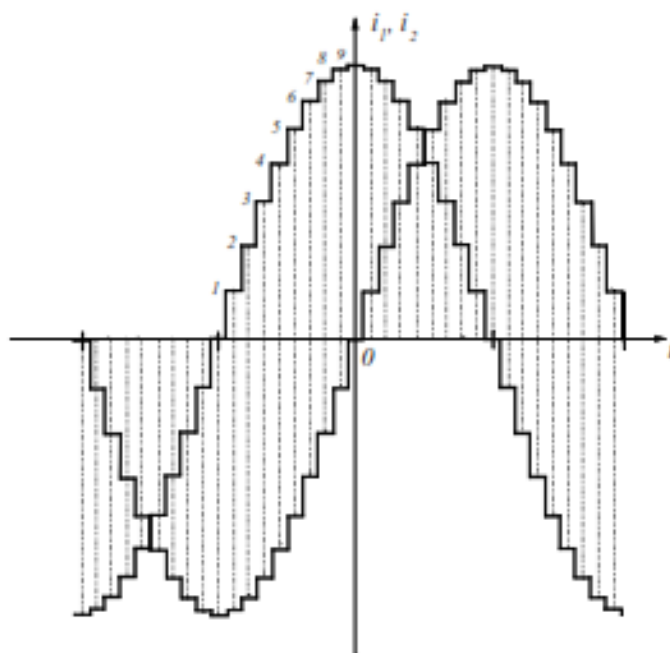


FIGURE I.10 – Mode micro pas

#### I.4.2 Couple d'arrêt ou couple de maintien

Le couple dépend aussi du mode fonctionnement. En mode biphase, le champ créé dans les bobines est 1,4 fois plus grand qu'en mode monophasé, le couple résultant est donc plus grand. Le positionnement angulaire présente une oscillation à chaque pas. Le rotor dépasse la position souhaitée puis revient en oscillant de part et d'autre de la position. Ce défaut est gênant à certaines vitesses [9].

#### I.4.3 Plage de démarrage

Plage dans laquelle un moteur pas à pas peut être actionné en synchronisation avec la fréquence de travail sans rampe d'accélération ou de décélération.

#### I.4.4 Fréquence limite de démarrage

Fréquence maximale avec laquelle un moteur pas à pas ne peut démarrer à la charge indiquée, sans perdre de pas.

#### I.4.5 Plage d'accélération

Plage de travail dans laquelle un moteur pas à pas peut être actionné en synchronisation avec la fréquence de travail, sans qu'il ne se produise d'erreur de pas. Il faut cependant qu'il soit actionné avec une rampe d'accélération et de décélération.

#### I.4.6 Couple limite de travail

Le couple de rotation maximale avec lequel on peut solliciter un arbre de rotation avant qu'il ne sorte de la cadence.

#### I.4.7 Fréquence maximale des pas

La fréquence maximale admise avec laquelle un moteur pas à pas est actionné à vide sans perte de pas. Cependant, le moteur ne peut être démarré ou stoppé avec cette fréquence sans perte de pas [8].

#### I.4.8 Comportement dynamique du moteur

Les moteurs pas à pas ne sont pas des moteurs rapides, les plus rapides dépassent rarement la vitesse maximale de 3 000 tr/min. Cette « lenteur » aidant, et ces moteurs étant naturellement sans balais (la majorité des moteurs pas à pas de haute qualité est de plus équipée de roulements à billes), ces moteurs ont une durée de vie extrêmement longue, sans nécessiter d'entretien [17].

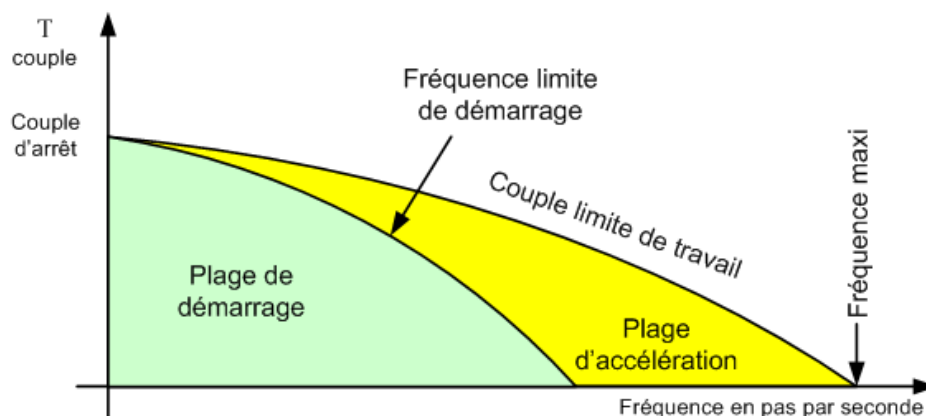


FIGURE I.11 – Caractéristiques Couple-fréquence

## I.5 Configurations internes des bobines [18]

### \* Les bipolaires :

- 4 fils, avec 2 bobines indépendantes (appelé «2 phases»).

### \* Les unipolaires :

- 5 fils, 1 commun aux 4 bobines, plus un par bobine,
- 6 fils, 2 bobines avec point milieu (parfois appelé «2 phases»),
- 8 fils, 4 bobines indépendantes, également appelé universel,
- Par abus de langage, ces moteurs sont tous appelés «4 phases»,

### \* et les moins classiques :

- 4 ou 6 fils avec un point commun à toutes les bobines.

### I.5.1 Moteur à 4 fils (bipolaire)

Les Moteurs bipolaire ils ont en général 4 fils et se commandent en inversant le sens du courant dans une des deux bobines, qui sont alimentés une fois dans un sens, une fois dans l'autre sens. Ils créent une fois un pôle nord, une fois un pôle sud d'où le nom de bipolaire.

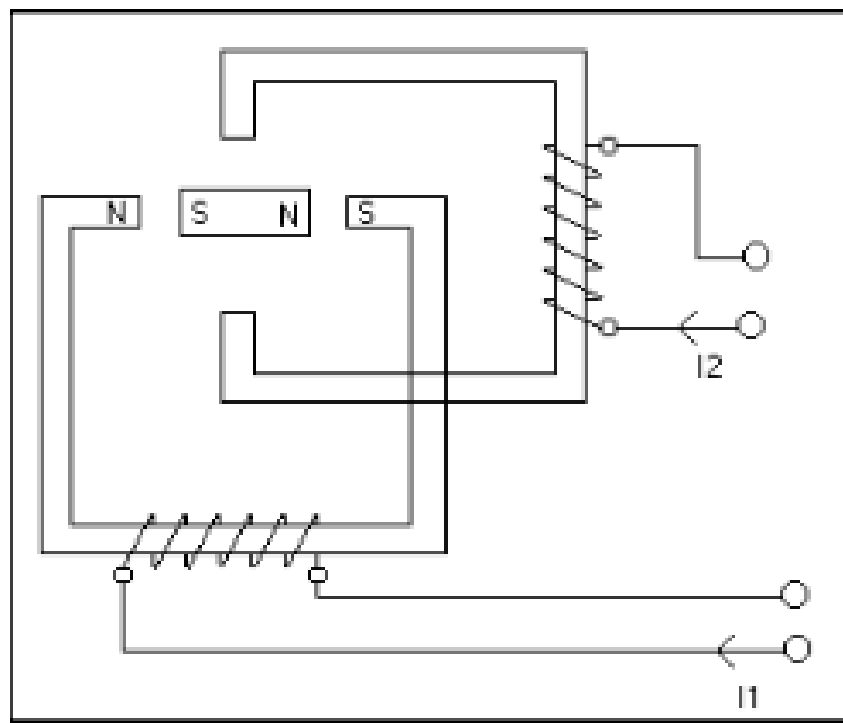


FIGURE I.12 – La commande du moteur bipolaire

La commande du moteur s'effectue par envoi successif, une à une, des combinaisons suivantes :

*Mode 1* : On alimente 1 enroulement à la fois :

*Mode 2* : On alimente 2 enroulements à la fois :

| Séquence | I1>0 | I1<0 | I2>0 | I2<0 | Angle |
|----------|------|------|------|------|-------|
| 1        | 1    | 0    | 0    | 0    | 0     |
| 2        | 0    | 0    | 1    | 0    | 90°   |
| 3        | 0    | 1    | 0    | 0    | 180°  |
| 4        | 0    | 0    | 0    | 1    | 270°  |

| Séquence | I1>0 | I1<0 | I2>0 | I2<0 | Angle |
|----------|------|------|------|------|-------|
| 1        | 1    | 0    | 1    | 0    | 45°   |
| 2        | 0    | 1    | 1    | 0    | 135°  |
| 3        | 0    | 1    | 0    | 1    | 225°  |
| 4        | 1    | 0    | 0    | 1    | 315°  |

### I.5.2 Moteur à 5 fils (unipolaire)

Les moteurs unipolaires ont en général 6 fils, dont deux sont reliés au milieu des bobines. Ils se commandent en les alimentant tour à tour, mais les bobinages sont alimentés toujours dans le même sens par une tension unique d'où le nom d'unipolaire.

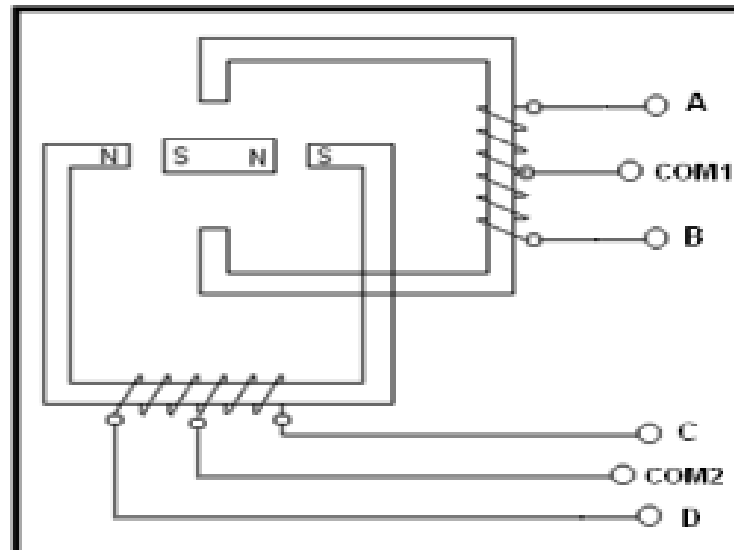


FIGURE I.13 – La commande du moteur unipolaire

Séquence d'alimentation des enroulements :

## I.6 Commande des moteurs pas à pas

la commande des moteurs pas à pas se fait par microcontrôleur, le choix du type du moteur unipolaire ou bipolaire et dans notre cas nous allons commander les moteurs par une carte de prototypage arduino

| Séquence | A | B | C | D | Angle |
|----------|---|---|---|---|-------|
| 1        | 1 | 0 | 0 | 0 | 0     |
| 2        | 0 | 0 | 1 | 0 | 90°   |
| 3        | 0 | 1 | 0 | 0 | 180°  |
| 4        | 0 | 0 | 0 | 1 | 270°  |

## I.7 Circuits intégrés dédiés à la commande des moteurs pas à pas

### I.7.1 Commande par SAA1027

Le SAA1027 est un circuit intégré qui réalise une commande directe, simple et fiable utilisé en technique bipolaire des moteurs pas à pas ( 4 phases), sans étage de puissance ni niveau de compensation .

Ce circuit comporte un compteur 4 états bidirectionnels et un décodeur pour commander les 4 sorties. Il est destiné particulièrement aux applications où le moteur fonctionne uniquement dans un milieu parasite et au démarrage. Sans architecture interne est donnée par la figure ci-dessous :

Le circuit SAA1027 comporte un étage d'entrée, un étage logique et un étage puissance :

\*

**Etage d'entrée :** Se divise en trois étages :

- étage de mise en forme des impulsions de commande du moteur.
- étage qui permet de définir le sens de rotation du moteur.
- étage de positionnement du compteur avant application d'impulsions.

\* **Etage logique :** Composé d'un compteur en anneau piloté par les impulsions d'entrée et commutant les différents étages de sortie selon une séquence adéquate

\* **Etage de puissance :** Composée de 4 étages de puissance délivrant chacun 350 mA. Et la protection de cet amplificateur de puissance est assurée par des diodes intégrées.

\* **Avantages :**

- Simple à utiliser.
- facile à réinitialiser.
- Il peut atteindre jusqu'à 85°C lors de fonctionnement.
- Il Contient un circuit de puissance
- Hautes immunité contre les bruits
- Haut courant de production.

\* **Inconvénients :**

- Il ne peut pas commander un très grand nombre de moteurs.



### I.7.2 Commande par UCN5804B

Le UCN5804 est un Microsystems Allégo ,il est simple à utiliser et à intégrer, il est obtenu par la combinaison de la logique CMOS de bas pouvoir avec haut courant et productions bipolaires de haut voltage.

Les circuits UCN5804B et UCN5804LB BiMOS sont des conducteurs fournissant le contrôle complet et pilotent un moteur pas à pas à quatre phases avec production continue de courants à 1.25 A par phase.

**\* Avantages :**

- Il a une grande Immunité contre les bruits.
- Il peut Fonctionne dans une gamme de température qui vraie entre  $[25 \text{ à } +85]^{\circ}\text{C}$ .
- Il contient un circuit de puissance intégré.
- Simple à utiliser et à intégrer.

**\* Inconvénients :**

- Il ne peut pas réaliser une commande idéale.
- On a besoin d'ajouter des diodes pour le circuit de puissance.

Le schéma ci-dessous montre le principe de fonctionnement de UCN5804B :

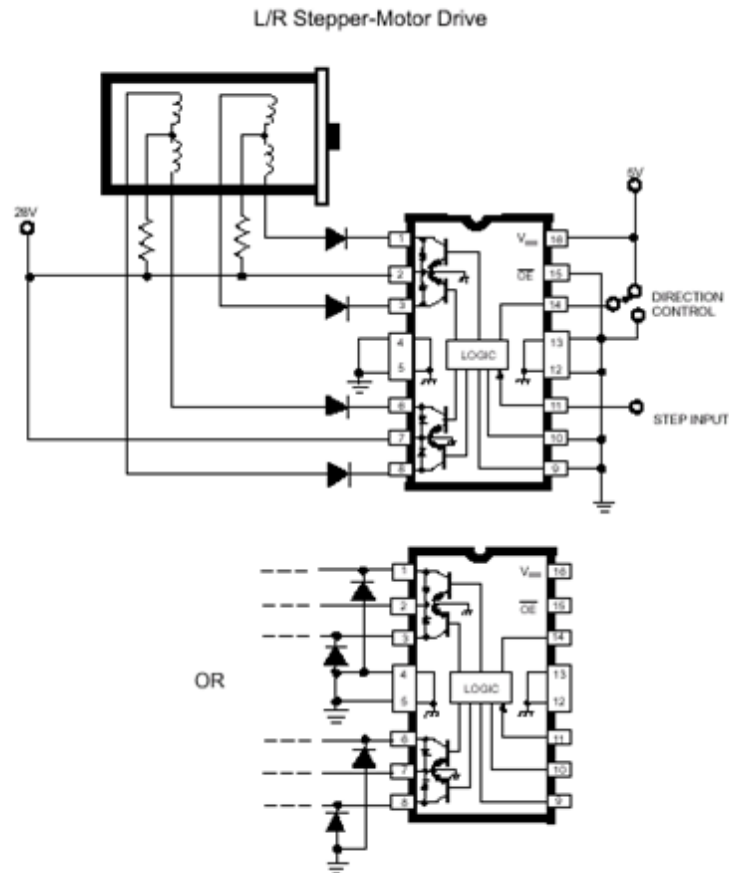


FIGURE I.14 – Schéma de principe de fonctionnement de UCN5804B

### I.7.3 Commande basées sur les par L297 et L298

Pour commande un moteur pas à pas, on emploie généralement de deux circuits intégrés spécialisés qui sont le L297 et le L298 qui présentent des avantages certains. Le L297 est un circuit qui contient toute la circuiterie de commande et de contrôle de moteurs pas à pas unipolaire et bipolaire. Utilisé avec un driver double pont tel que le L298, l'ensemble forme une interface idéale pour le contrôle d'un moteur pas à pas bipolaire par microprocesseur ou micro contrôleur.

Il est possible de contrôler un moteur pas à pas unipolaire avec un L297 en lui adjoignant quatre transistors Darlington ou Mosfet

#### I.7.3-a Le circuit L297

Le L297 est un circuit de commande de moteur pas à pas il est initialement prévu pour être utilisé avec un driver L298 ou bien L293E. Il reçoit les signaux de contrôle qui provient d'un système à base de microcontrôleur et il génère tous les signaux nécessaires pour l'étage de puissance. Ce circuit contient des hacheurs à modulation de largeur d'impulsion (MLI) pour réguler le courant des enroulements du moteur.

De plus il est capable de gérer le mode demi pas et pas complet. Le L297A est une version spéciale pour le positionnement des têtes de lecteur de disquette.

Il possède 4 broches de sorties pour générer la séquence et plusieurs en entrée pour paramétrer le fonctionnement voulu. Parmi elles on en retrouve trois principales :

- CW/CCW : (ClockWise ou Counter ClockWise) qui décidera du sens de rotation du moteur (horaire ou anti-horaire).
- Half/Full : Qui décide si on est en mode pas entier ou demi-pas.
- Clk : (Clock) qui est l'horloge pour la vitesse. A chaque front descendant le moteur fera un pas [5].

#### Avantages

- Permet la commande de moteur pas à pas dans deux sens de rotation,
- Limitation de courant moteur et régulation dans les enroulements du moteur,
- Entrée de validation,
- Grande fiabilité,
- Système intégré simple et robuste il peut effectuer : half/full step, réglage intensité par PWM, etc..
- Se branche directement sur un pont en H, comme par exemple le L298,
- Ce circuit intégré peut être commandé par ordinateur implique que les signaux de commande peuvent être fournis par un logiciel de commande en utilisant les port, série et parallèle d'un PC.

### Inconvénients

- Nécessite une carte de puissance fournie par l'utilisateur,
- Nécessite d'autres circuits intégrés.

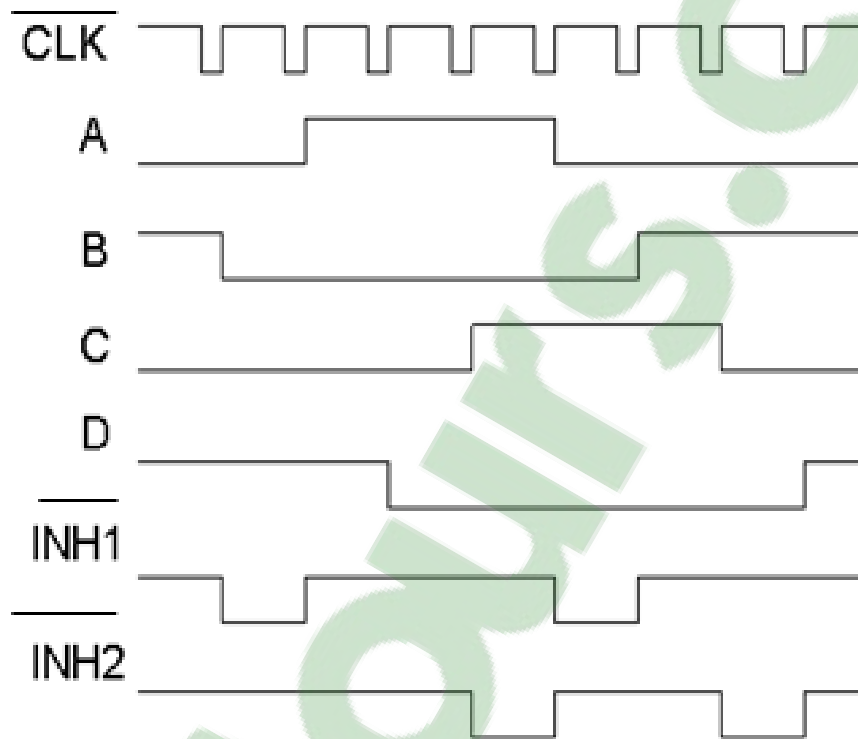


FIGURE I.15 – Séquence basique

#### I.7.3-b Le circuit L298

Le circuit intégré L298 contient deux étages de puissance configurés en pont, (un double pont en H) chacun commandé par deux entrées logiques (A, B et C, D) ainsi que deux entrées de validation (INH1 et INH2). De plus deux broches sont connectées en interne aux émetteurs des transistors (paires inférieures) qui permettent la connexion de résistances de courant.

Il est à noter que, contrairement à la majorité des circuits, le L298 ne contient pas les diodes de protections des transistors internes. Il conviendra donc de les câbler à l'extérieur du boîtier. Ces diodes devront présenter des temps de commutation faible et devront être capable de laisser passer un courant important.

Le L298 est présenté en boîtier MULTIWATT à 15 broches, ce qui simplifie sa fixation sur un refroidisseur [7].

### Avantages

- Gamme de voltage jusqu'à 46 V ,
- Commande de moteur pas à pas dans deux sens de rotation,
- Courant totale jusqu'à 4 A,
- Bas voltage de saturation,
- Protection sûre de la température,
- Voltage de l'entrée logique jusqu'à 1.5 V,
- Haute immunité contre les bruits,
- Gamme de température ambiante très acceptable (-40 à 130 °C).

### Inconvénients

- Nécessite une carte de puissance fournie par l'utilisateur
- Nécessite d'autre circuit intégré pour réaliser une bonne commande de moteur pas à pas donc on a besoin d'une grande carte pour implanter les composants

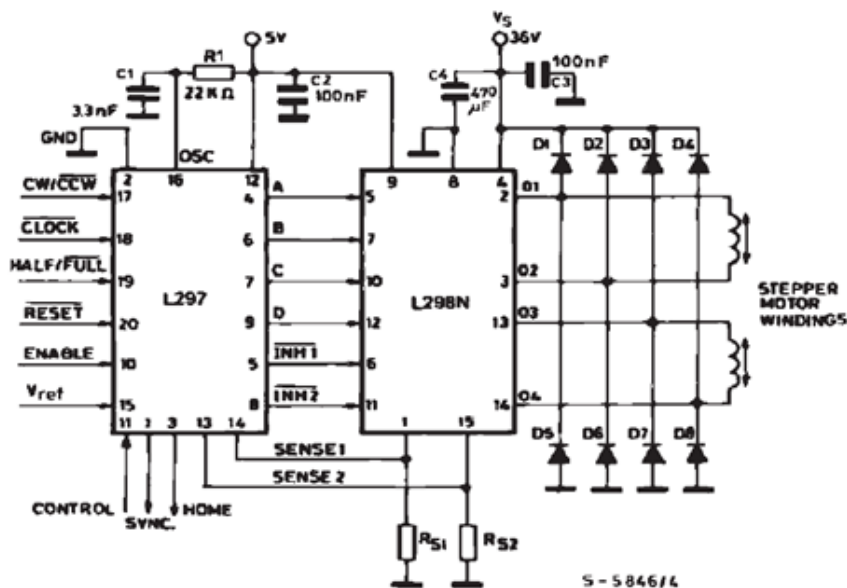


FIGURE I.16 – Séquence basique

## I.8 Critère de choix d'un moteur pas à pas

Le choix d'un moteur pas à pas repose sur les besoins en vitesse et en couple de l'application. Utilisez la courbe de couple-vitesse du moteur (qui figure dans les spécifications de chaque moteur, par exemple dans la figure C) pour sélectionner un moteur qui conviendra.

Chaque entraînement de moteur pas à pas de la gamme *Omegamation* indique les courbes de couple-vitesse pour les moteurs recommandés pour cet entraînement. Si vos besoins en couple et vitesse peuvent être satisfaits par plusieurs moteurs pas à pas, choisissez un entraînement en vous basant sur les besoins de

vosre système de mouvement (pas/direction, programmation autonome, entrées analogiques, micropas), puis choisissez l'un des moteurs recommandés pour cet entraînement.

La liste recommandée de moteurs est basée sur des tests approfondis effectués par le fabricant pour garantir des performances optimales de la combinaison moteur pas à pas/entraînement[14].

## I.9 Conclusion

Dans ce chapitre, nous avons donné un aperçu général sur les différents types de moteurs pas à pas, ainsi que leur mode de fonctionnement.

### Avantages

- Le moteur peut être contrôlé en boucle ouverte sans problème de stabilisation ;
- Facilité d'intégration dans un système complexe ;
- Positionnement précis ;
- Contrôle de la position (pas besoin de contre-réaction).
- Réponse très rapide.

### Inconvénients

- Risque de perte de pas (donc de dysfonctionnement du contrôle en position) en cas de perturbation anormale notamment de la charge ou de fréquence excessive des impulsion ;
- Plus difficile à faire fonctionner qu'un moteur à courant continu ;
- Couple décroissant rapidement lorsque la vitesse augmente.

## Chapitre II

---

# Présentation de la carte de prototypage Arduino

### SOMMAIRE

---

|        |   |    |
|--------|---|----|
| II.1   | HISTORIQUE DU PROJET ARDUINO . . . . .          | 24 |
| II.1.1 | Le logiciel . . . . .                           | 25 |
| II.1.2 | Le matériel . . . . .                           | 25 |
| II.2   | CARTE ÉLECTRONIQUE . . . . .                    | 26 |
| II.3   | QUELQUES OUTILS UTILISÉS AVEC ARDUINO . . . . . | 27 |
| II.3.1 | Processing . . . . .                            | 27 |
| II.3.2 | Pure Data . . . . .                             | 27 |
| II.4   | BUT DE LA CARTE ARDUINO . . . . .               | 28 |
| II.5   | LES APPLICATIONS POSSIBLES . . . . .            | 29 |
| II.6   | LES DIFFÉRENTS TYPES DE CARTES . . . . .        | 29 |
| II.7   | LES «SHIELDS» . . . . .                         | 33 |
| II.8   | CONCLUSION : . . . . .                          | 34 |

---

## II.1 Historique du projet arduino

Le projets Arduino est né en hiver 2005 il a été créé par une équipe de développeurs composée de Massimo Banzi (enseignant dans une école de Design), David Cuartielles (Ingénieur espagnol spécialisé sur les micro-contrôleurs), Tom Igoe, Gianluca Martino, David Mellis et Nicholas Zambetti , de l'Interaction Design Institute d'Ivrea, en Italie .

cette équipe a imaginé un projet répondant au doux et mettant en œuvre une petite carte électronique programmable et un logiciel multiplateforme, qui puisse être accessible à tous a pour but de créer facilement des systèmes électroniques.. Sous le nom de Arduino.

Le nom Arduino trouve son origine dans le nom du bar dans lequel l'équipe avait l'habitude de se retrouver. Arduino est aussi le nom d'un roi italien, personnage historique de la ville « Arduin d'Ivrée », ou encore un prénom italien masculin qui signifie « l'ami fort » [1].

Cette équipes est partie sur une « philosophie » et L'idée c'était d'utiliser la carte Arduino comme un macro-composant dans des applications de prototypage électronique. Le concepteur n'a plus qu'à développer des interfaces et programmer le macro-composant pour réaliser son application !

l'Arduino a été conçu aux étudiants de pouvoir disposer d'une plate forme valant un prix bas pour réaliser des projets interactifs et aussi pour être bon marché , et destiné aussi aux artistes, designers, et des amateurs et à n'importe qu'elle personne intéressé à créer des objets ou environnements interactifs.

Et vu sa simplicité d'utilisation, Arduino est utilisé dans beaucoup d'applications comme l'électronique industrielle et embarquée, le modélisme, la domotique mais aussi dans des domaines différents comme l'art contemporain ou le spectacle !

Ca devient un hit tout de suite auprès des étudiants. Tout le monde arrive à en faire quelque chose très rapidement sans même avoir de connaissances particulière ni en électronique ni en informatique : réponse à des capteurs, faire clignoter des leds, contrôler des moteurs . . . Ils publient les schémas, investissent 3000 euros pour créer le premier lots de cartes : 200. Les 50 premières partent directement à des élèves de l'école. En 2006, 5 000 cartes ont été vendue. [12].

Le système Arduino est une plate-forme de prototypage open source d'objets interactifs à usage créatif constituée d'une carte électronique programmée à microcontrôleur AVR et un environnement de développement permettant d'écrire, compiler et tester un programme.

Il est composé d'entrées-sorties qui peuvent recevoir des signaux de capteurs ou interrupteurs et peuvent commander des moteurs, éclairages, etc.

Les projets Arduino peuvent être autonomes ou communiquer avec un PC.

### II.1.1 Le logiciel

L'environnement de programmation Arduino (IDE en anglais) est une application gratuite et open source, développé en Java elle était inspirée du langage Processing, et dont la simplicité d'utilisation L'IDE permet d'écrire et de modifier un programme et de le convertir en une série d'instructions compréhensibles pour la carte.

Le schéma si dessous montre le logiciel avec lequel on écrit les lignes de programmes qui vont être transmises à la mémoire de la carte arduino par un câble qui relie l'ordinateur avec la carte.

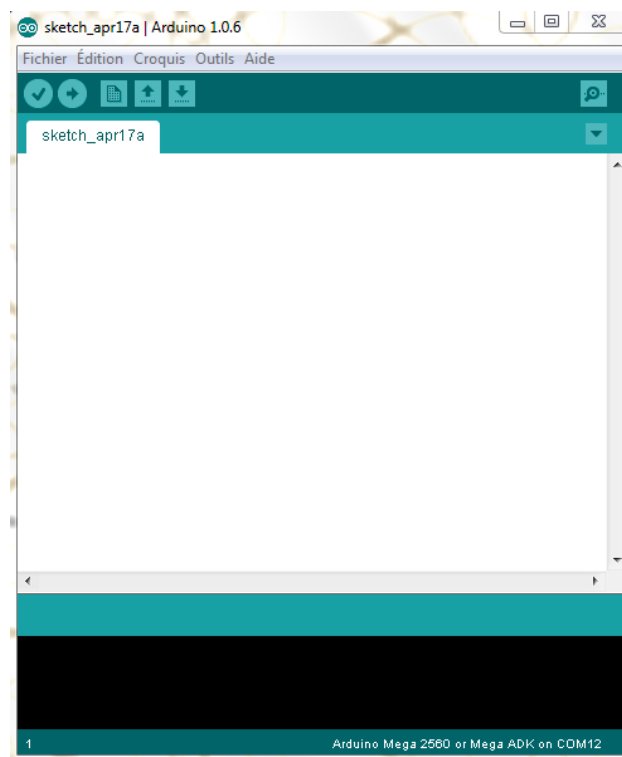


FIGURE II.1 – Logiciel de programmation Arduino (IDE)

### II.1.2 Le matériel

Cartes électroniques dont les schémas sont en libre circulation sur internet [5], elle contient un circuit intégré (un microcontrôleur) associée à des entrées et sorties qui permettent à l'utilisateur de brancher différents types d'éléments externes :



- **des capteurs** : Côté entrées, ils collectent des informations sur leur environnement comme par exemple la variation de température (sonde thermique), le mouvement par un détecteur de présence ou un accéléromètre, ou le contact via un bouton-poussoir, ... etc.
- **des actionneurs** : Côté sortie ils agissent sur le monde physique telle qu'un moteur qui actionne un bras articulé ou petite lampe qui produit de la lumière, ... etc. Cette carte électronique peut être autonome et fonctionner sans ordinateur ou servir d'interface.

Il existe plusieurs variétés de cartes Arduino. La figure ci-dessous montre par exemple, la dernière version nommée « Uno », sortie en 2010 [1].

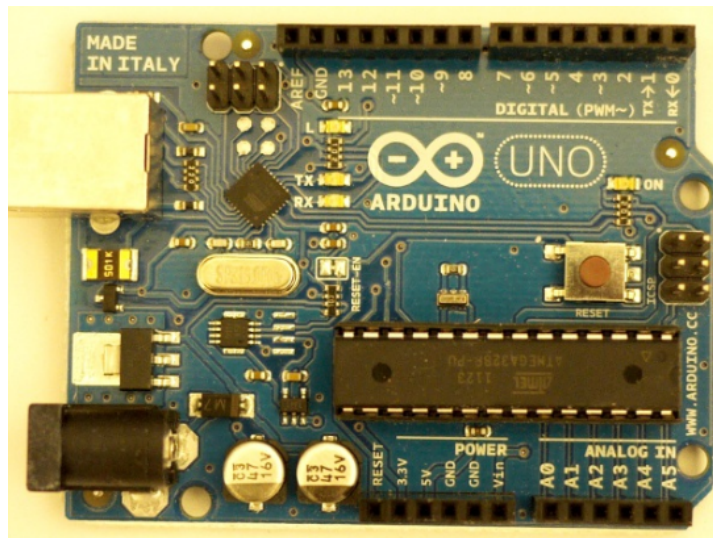


FIGURE II.2 – Carte Arduino « Uno »

## II.2 Carte électronique

Une carte électronique est un support plan, flexible ou rigide, elle est généralement composée de fibre de verre ou d'époxy. Elle possède des pistes électriques disposées sur une, deux ou plusieurs couches (en surface et/ou en interne) qui permettent la mise en relation électrique des composants électroniques.

Chaque piste relie un composant à un autre, de telle façon à créer un système électro-nique qui fonctionne et qui réalise des opérations demandées. Évidemment le câblage des composants suit un plan spécifique à chaque carte électronique, qui se nomme le schéma électronique.

Il faut tout d'abord transformer le schéma électronique en un schéma de câblage, appelé aussi typon, avant de passer à la réalisation d'une carte électronique.

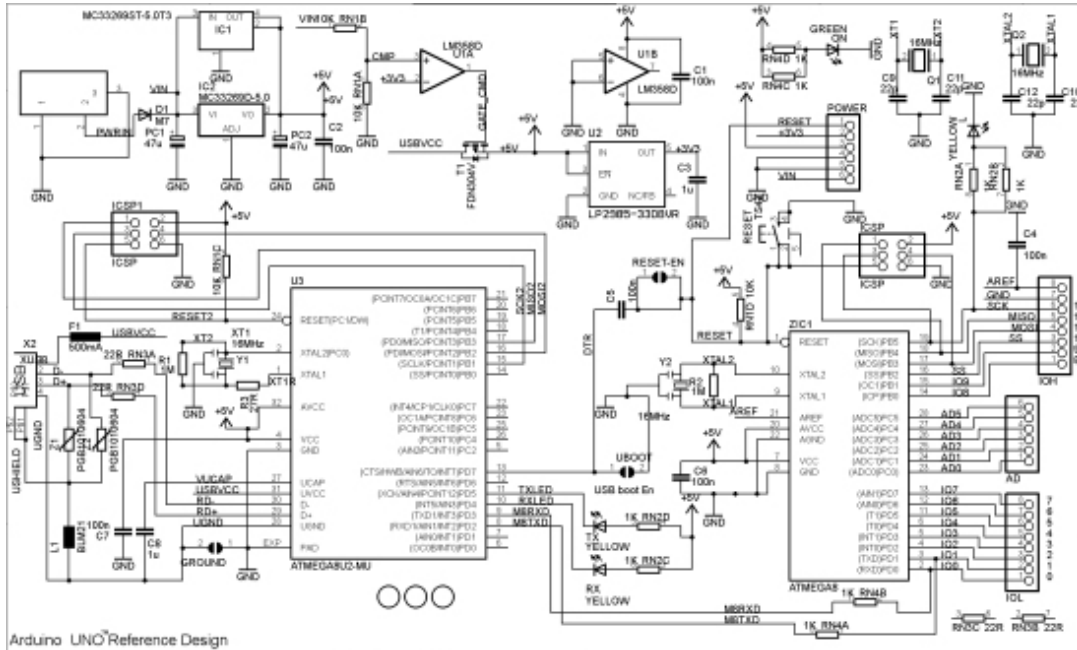


FIGURE II.3 – Exemple de schéma électronique de la carte Arduino Uno

## II.3 Quelques outils utilisés avec Arduino

Lorsque l'Arduino est connecté à un ordinateur, il est capable de communiquer avec diverses applications, notamment :

### II.3.1 Processing

Conçu par des artistes, pour des artistes, Processing est un environnement de création. Fréquemment utilisé pour générer des œuvres multimédias à partir d'un code informatique sur ordinateur. L'attrait de ce logiciel réside dans sa simplicité d'utilisation et dans la diversité de ses applications : image, son, applications sur Internet et sur téléphones mobiles, conception d'objets électroniques interactifs [1].

### II.3.2 Pure Data

Pure Data 4 (souvent abrégé Pd) est un logiciel de création multimédia interactive couramment utilisé dans les domaines artistique, scientifique et pédagogique. Sa popularité réside dans ses possibilités de programmation en temps réel.

Plutôt qu'un langage de programmation textuel, Pure Data propose un environnement de programmation graphique dans lequel l'utilisateur est invité à manipuler des icônes représentant des fonctionnalités et à les brancher ensemble [10].

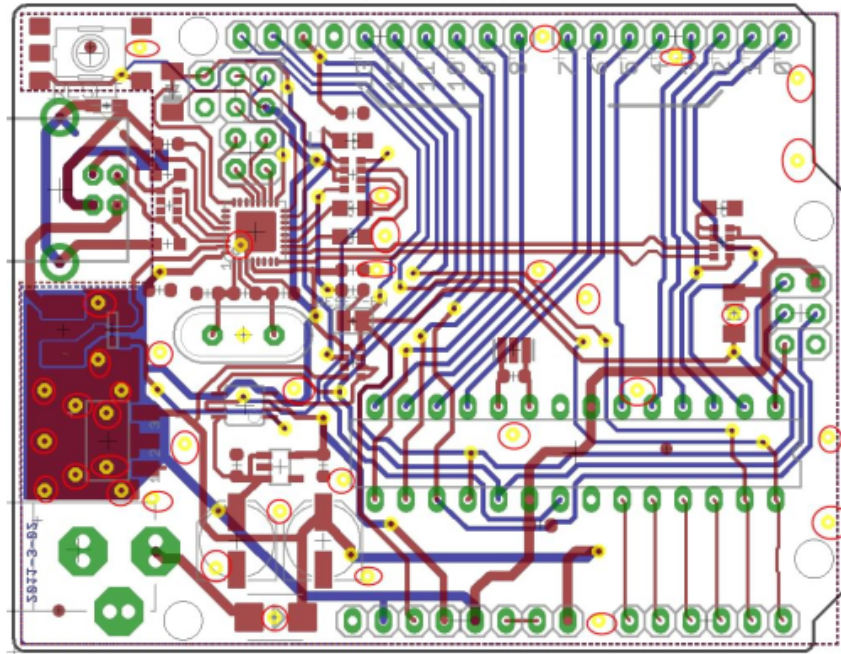


FIGURE II.4 – Exemple de typon de la carte Arduino

## II.4 But de la carte Arduino

Tout d'abord :

- Arduino une carte électronique programmable et son logiciel est gratuit,
- de plus un prix dérisoire elle est peu coûteuses et Le logiciel est fourni gratuitement étant donné l'étendue des applications possibles
- une compatibilité sous toutes les plate formes Windows, Linux et Mac OS Contrairement aux autres logiciel qui ne sont en général compatibles qu'avec Windows ;
- Une communauté ultra développée, de présentations de projets, de propositions de programmes et de bibliothèques, on constate une évolution sur internet ;
- Un site en anglais et un autre en français [arduino.cc](http://arduino.cc) : toute la référence Arduino, le matériel, des exemples d'utilisations, de l'aide pour débiter, des explications sur le logiciel et le matériel, ... etc ;
- Une liberté quasi absolue ;
- Existence de « shield » (boucliers en français) : ce sont des cartes supplémentaires qui se connectent sur le module Arduino pour augmenter les possibilités comme par exemple : afficheur graphique couleur, interface Ethernet, GPS, ... etc.

Cette liberté a une condition : le nom « Arduino » ne doit être employé que pour les cartes « officielles » . En somme, vous ne pouvez pas fabriquer votre propre carte sur le modèle Arduino et lui assigner le nom « Arduino »[6].

## II.5 Les applications possibles

Cette plate forme logicielle et matérielle de création d'objets numériques permet de programmer des circuits électroniques qui interagissent avec son environnement en la reliant notamment à des capteurs (thermiques, de mouvement, sonores, ...etc ).

Ces circuits électroniques peuvent générer des images, actionner un bras articulé, envoyer des messages sur Internet, ...etc, avec un prix peu coûteux, pour cela des dizaines de milliers d'artistes, de designers, d'ingénieurs, de chercheurs, d'enseignants et même d'entreprises l'utilisent pour réaliser des projets incroyables dans de multiples domaines [5] :

- contrôler des appareils domestiques,
- donner une intelligence à un robot,
- réaliser des jeux de lumières,
- permettre à un ordinateur de communiquer avec une carte électronique et différents capteurs,
- télécommander un appareil mobile (modélisme), ...etc.

## II.6 Les différents types de cartes

Il y existe trois types de cartes :

- Les dites « officielles » qui sont fabriquées en Italie par le fabricant officiel : Smart Projects.
- Les dits « compatibles » qui ne sont pas fabriqués par Smart Projects, mais qui sont totalement compatibles avec les Arduino par-officielles.
- Les « autres » fabriquées par diverse entreprise et commercialisées sous un nom différent (Freeduino, Seeduino, ...etc.

### II.6.0-a La carte Uno et Duemilanove

Nous choisirons d'utiliser la carte portant le nom de « Uno » ou « Duemilanove », ces deux versions étant presque identiques. Le kit Arduino UNO est basé sur un micro-contrôleur ATmega 328 cadencé à 16 MHz et disposant de 32 kB de Flash et 2 kb d'EEPROM. Il dispose de 14 entrées/sorties digitales (dont 6 peuvent être utilisé en sortie PWM) ainsi que 6 entrées analogiques.

Par rapport au kit Arduino Duemilanove, on observera le changement de sérigraphie, beaucoup plus design et pratique à utiliser, avec une clarté des fonctionnalités. Mais la plus grosse différence vient de l'absence sur circuit FTDI permettant une connexion via un port série virtuel. Cette fois ci c'est un AT-Mega8U2 qui permet de réaliser cette fonction et il permet aussi d'ouvrir les portes de l'USB, en pouvant facilement modifier les piles et donc transformer le kit Arduino en périphérique USB modulable à souhait [4].



FIGURE II.5 – Arduino UNO

### Caractéristiques de la carte Arduino uno

Le kit Arduino UNO est basé sur :

- Micro contrôleur ATmega328,
- Tension d'alimentation interne égale à 5V,
- Tension d'alimentation (recommandée) égale 7 à 12 V, limites égale a 6 à 20 V,
- Entrées/sorties numériques : 14 dont 6 sorties PWM,
- Entrées analogiques égale a 6,
- Courant max par broches E/S = 40 mA,
- Courant max sur sortie 3,3V = 50 mA,
- Mémoire Flash 32 kB dont 0.5 kB utilisée par le bootloader,
- Mémoire SRAM 2 kB,
- Mémoire EEPROM 1 kB,
- Fréquence horloge = 16 MHz,
- Dimensions = 68,6 mm x 53,3 mm [16].

La carte s'interface au PC par l'intermédiaire de sa prise USB. La carte est alimentée par le jack d'alimentation (utilisation autonome) mais peut être aussi alimentée par l'USB (en phase de développement par exemple).

La figure ci-dessous montre le microcontrôleur ATmega328

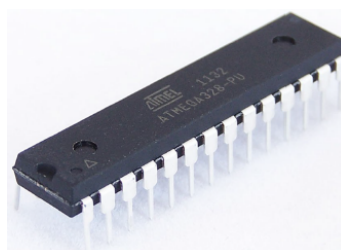


FIGURE II.6 – Microcontrôleur ATmega328



### II.6.0-b La carte Megae

La carte Arduino Mega est une autre carte qui offre toutes les fonctionnalités des précédentes, mais avec des options en plus.

On retrouve notamment un nombre d'entrées et de sorties plus importantes ainsi que plusieurs liaisons séries. En revanche, le prix est plus élevé : plus de 50 !

L'Arduino Mega 2560 est une carte microcontrôleur basée sur le ATmega2560 ( fiche technique ). Il dispose de 54 broches numériques d'entrée / sortie (dont 15 peuvent être utilisés comme sorties PWM), 16 entrées analogiques, 4UART (ports série matériels), un 16 MHz oscillateur en cristal, une connexion USB, une prise d'alimentation, d'une embase ICSP, et un bouton de réinitialisation.

Il contient tout le nécessaire pour soutenir le microcontrôleur ; suffit de le brancher à un ordinateur avec un câble USB ou de la puissance avec un adaptateur ou la batterie AC-DC pour commencer. Le Mega est compatible avec la plupart des blindages conçus pour l'Arduino Duemilanove ou Diecimila [\[13\]](#).

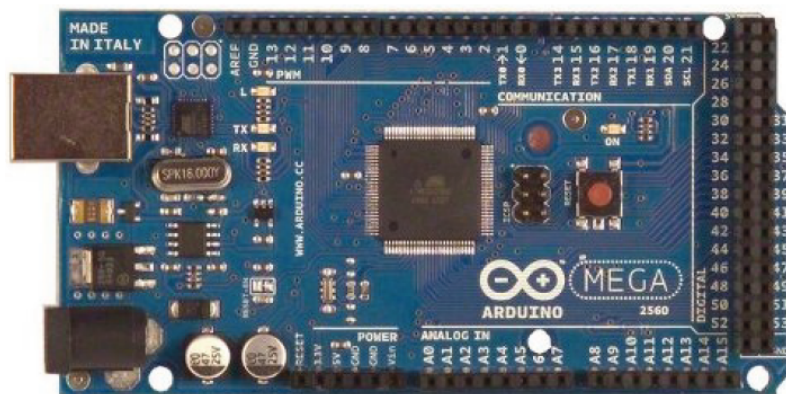


FIGURE II.7 – Arduino Mega 2560

### Caractéristiques de la carte Arduino mega

#### Mémoire

La Mega2560 contient le microcontrôleur ATmega2560 avec mémoire flash de 256 kB pour stocker le code dont :

- 8 ko est utilisée pour le chargeur de démarrage,
- 8 ko de SRAM,
- 4 ko de mémoire EEPROM (qui peut être lu et écrit avec la bibliothèque de EEPROM).

#### Entrée et sortie

Le Mega2560 dispose de 54 broches numériques Chacune peut être utilisé comme une entrée ou une sortie, on utilisant des fonctions comme pinMode, digitalWrite et digitalRead pour initialiser les ports.

Chaque broches fonctionnent à 5 volts et peut fournir ou recevoir un maximum de 40 mA et a une résistance de rappel interne (déconnecté par défaut) de 20-50 kOhms. En outre, certaines broches ont des fonctions spécialisées et dispose de 16 entrées analogiques, chacune fournissent 10 bits de résolution (à savoir 1024) des valeurs différentes. Par défaut, ils mesurent 5 Volts par rapport à la masse.

Cependant il est possible de changer l'extrémité supérieure de sa gamme en utilisant la broche AREF et la fonction `analogReference`.

### **Puissance**

L'Arduino Mega peut être alimenté avec une alimentation externe ou bien via la connexion USB et la source d'alimentation est automatiquement sélectionné.

L'alimentation externe (Non-USB) peut provenir de la batterie ou d'un adaptateur (AC-DC). L'adaptateur peut être connecté en branchant une prise 2.1 mm. Leads partir d'une batterie peuvent être insérés dans les en-têtes de broches GND et Vin du connecteur d'alimentation et peut fonctionner sur une alimentation externe de 6 à 20 Volts, cependant, la broche de 5V peut fournir moins de cinq volts. Si on utilise plus de 12 Volts, le régulateur de tension peut surchauffer et endommager la carte, la plage recommandée est de 7 à 12 Volts.

### **Communication**

L'Arduino Mega2560 a un certain nombre de moyens pour communiquer avec un ordinateur, ou un autre Arduino, ou bien d'autres microcontrôleurs. L'ATmega2560 fournit quatre matériel UART TTL pour (5V) de communication série.

Un ATmega16U2 sur la carte canaux, un de ces plus USB et fournit un port COM virtuel pour le logiciel sur l'ordinateur (machines Windows auront besoin d'un fichier .inf, mais OSX et Linux machines seront reconnaître le conseil comme un port COM automatiquement.

Le logiciel Arduino comprend un moniteur de série qui permet aux données textuelles simples pour être envoyés vers et depuis le bord. Les RX et TX LED sur la carte clignote lorsque des données sont transmises via le ATmega8U2 / ATmega16U2 puce et la connexion USB à l'ordinateur (mais pas pour la communication série sur les broches 0 et 1).

Une bibliothèque `SoftwareSerial` permet pour la communication série sur l'une des broches numériques de la Mega2560.

Le ATmega2560 supporte également la communication TWI et SPI. Le logiciel comprend une bibliothèque Arduino de fil de simplifier l'utilisation du bus TWI; voir le documentation pour plus de détails. Pour la communication SPI, utiliser la librairie SPI .

### Programmation

L'Arduino Mega peut être programmé avec le logiciel Arduino qui se télécharge gratuitement sur le site officiel Les ATmega2560 sur l'Arduino Mega vient preburned avec un bootloader qui vous permet de télécharger le nouveau code à elle sans l'utilisation d'un programmeur de matériel externe. Il communique en utilisant l'original STK500protocole ( référence , les fichiers d'en-tête C ).

Vous pouvez également contourner le bootloader et programmer le microcontrôleur à travers le (Serial Programming In-Circuit) ICSP tête en utilisant Arduino FAI ou similaire; voir les instructions pour plus de détails. Le ATmega16U2 (ou 8U2 dans le rev1 et rev2 conseils) code source du firmware est disponible dans le référentiel Arduino. Le ATmega16U2 / 8U2 est chargé avec un chargeur de démarrage DFU, qui peut être activé par

### Protection contre les surintensités

L'Arduino Mega2560 a une *polyfuse* remise à zéro qui protège les ports USB de l'ordinateur contre les surintensités, de sorte que la plupart des ordinateurs fournissent leurs propres protections internes, le fusible fournit une couche supplémentaire de protection. S'il est appliqué plus de 500 mA sur le port USB, le fusible cassera automatiquement la connexion tant que le court ou de surcharge est enlevée.

## II.7 Les «Shields»

Il existe de nombreux shields que l'on traduit parfois dans les documentations par «boucliers ». Un «Shield» Arduino est une petite carte qui se connecte sur une carte Arduino pour augmenter ses fonctionnalités.

Quelques exemples de « shields » :

- afficheur graphique,
- Ethernet et carte SD,
- GPS,
- carte de prototypage (type labdec),
- ... etc.





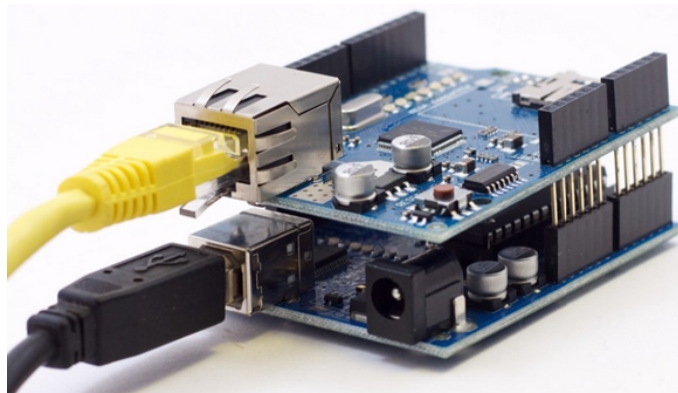


FIGURE II.8 – Exemple de Shield

## II.8 Conclusion :

Dans ce chapitre, nous avons donné un aperçu général sur les différents types de carte Arduino, ainsi que leurs modes de fonctionnement et leurs caractéristiques. Dans ce projet de fin d'étude, nous avons privilégié la carte Arduino Mega2560, ses performances étant suffisante pour le projet envisagé.

Ce qui prime dans Arduino, c'est sa simplicité qui permet de mettre en œuvre de nombreux objets numériques à moindre coût sans être un spécialiste en électronique ou en programmation des microcontrôleurs, cela nécessite juste des petites connaissances en électronique. C'est cette qualité là qui donne à Arduino le succès qu'on lui connaît.

## Chapitre III

---

# Commande de moteurs pas à pas basées sur le module Arduino

### SOMMAIRE

---

|   |    |
|---|----|
| III.1 INTRODUCTION . . . . .  | 36 |
| III.2 PRÉSENTATION DU PROJET . . . . .                                  | 36 |
| III.3 SPÉCIFICATIONS FONCTIONNELLES . . . . .                           | 36 |
| III.3.1 Le circuit de commande . . . . .                                | 36 |
| III.3.2 Le circuit de puissance . . . . .                               | 37 |
| III.4 TESTS ET VALIDATION . . . . .                                     | 37 |
| III.4.1 Réalisation du circuit . . . . .                                | 37 |
| III.4.2 Réalisation des typons . . . . .                                | 38 |
| III.4.3 Développement de la carte . . . . .                             | 42 |
| III.4.4 Schéma bloc . . . . .   | 42 |
| III.4.5 Commande d'un seul moteur . . . . .                             | 44 |
| III.4.6 Commande de deux moteurs pas à pas avec afficheur LCD . . . . . | 47 |
| III.4.7 Programmation du microcontrôleur . . . . .                      | 49 |
| III.5 CONCLUSION . . . . .  | 57 |

---

## III.1 Introduction

Aujourd'hui, les systèmes automatiques utilisés dans l'industrie ont connus des très grandes avancées en électronique, en robotique, en effet la programmation a donné naissance à de nouvelles techniques innovantes. D'autre part, la nécessité et le besoin croissant des ces systèmes industriels a causé la naissance à de nouveaux types de moteur qui ont pour but de pouvoir imposer des mouvements avec une très grande précision.

L'apparition des petits systèmes électroniques associés à la programmation illustrées dans le deuxième chapitre portant le nom Arduino ont simplifié le prototypage en temps réel et la réalisation de systèmes automatiques avancés. Dans ce chapitre, nous présentons la réalisation d'une carte de commande pour le contrôle de deux moteurs pas à pas à l'aide d'une carte arduino de type mega 2560.

## III.2 Présentation du projet

Ce projet a pour but la conception et réalisation et la mise en œuvre des différentes cartes de commande pour contrôler des moteurs pas à pas à l'aide d'une carte numérique de prototypage Arduino.

Dans un premier temps, l'organisation et le principe de fonctionnement des moteurs pas à pas seront exposés, ainsi que leur modélisation respective. Les cartes à concevoir se divisent en deux parties. Une carte de commande comportant le microcontrôleur et un pour moteur pas à pas bipolaire (circuit intégré spécialisé pour le contrôle de ce type de moteur) et une carte de puissance comportant une alimentation externe et un hacheur (deux ponts en H).

Les approches de contrôle retenues, seront analysées, et leurs conceptions détaillées. Les solutions envisagées seront réalisées et imprimées sur différentes cartes de commande. Des essais expérimentaux seront effectués en vue de mettre en évidence les performances obtenues pour chacune des cartes. Tous les résultats obtenus seront évalués et comparés.

## III.3 Spécifications fonctionnelles

### III.3.1 Le circuit de commande

Le moteur est un moteur pas à pas est commandé par des séquences d'impulsions, et grâce à sa technologie et son constitution l'ensemble lui permet de faire une rotation discrète en pas (angle). Donc il sera facile à faire une rotation en angle pré-déterminé avec une grande précision.

Le circuit de commande est composé d'une carte de prototypage arduino mega 2560 et un pilote du moteur pas à pas. Le pilote L297 est un circuit spé-

cialisé pour la commande du moteur pas à pas bipolaire. Il suffit de leur indiquer le sens de rotation, le mode pas ou demi pas et bien sur la fréquence des signaux générés pour la commande du moteur en fournissant une horloge provenant du micro contrôleur (Arduino). Il est aussi possible d'utiliser un astable comme le ne555 pour générer le signal d'horloge.

### III.3.2 Le circuit de puissance

Le L298 est un hacheur (double pont en H) permis l'amplification des séquences d'impulsions provenant de la carte de commande a pour but d'alimenter les enroulement du moteur selon les signaux de la commande. C'est un module extrêmement utile pour le contrôler de robots et ensembles mécanisés. Il peut contrôler un moteur pas-à-pas bipolaire (4 fil) ou deux moteur courant continu ou chaque MC est équivalent à une bobine dans le moteur pas à pas.

Cet hacheur est conçu pour supporter des tensions plus élevées jusqu'à 36V, et des puissance jusqu'à 25 W. Son alimentation externe depend de la valeur nominale de la tension d'alimentation de fonctionnement le moteur mentionné sur sa plaque signalétique.

#### Moteur pas à pas

Le moteur pas à pas constitue un convertisseur électromécanique destiné à transformer le signal électrique (impulsion ou train d'impulsions de pilotage) en déplacement (angulaire ou linéaire) mécanique.

C'est un mécanisme qui positionne des charges en fonctionnant par pas de décalage angulaire. Cette rotation pas à pas est réalisée en commutant l'alimentation aux bobines de façon à ce que les phases du moteur soient activées selon une séquence déterminée.

## III.4 Tests et validation

### III.4.1 Réalisation du circuit

Il existe plusieurs cartes qui permettent le contrôle d'un moteur pas à pas, dont la plus part de ces dernier se composent de deux partie, une partie de commande et une partie de puissance. la partie de commande est essentielle mais la partie de puissance n'est pas obligatoire lors de l'application des petits moteurs de très faibles puissances.

La figure suivante [III.1](#) illustre les différents composants du système composé d'un microcontrôleur et un pilote L297 et un hacheur L298 (deux ponts en H) et l'association entre les deux cartes à réalisés.

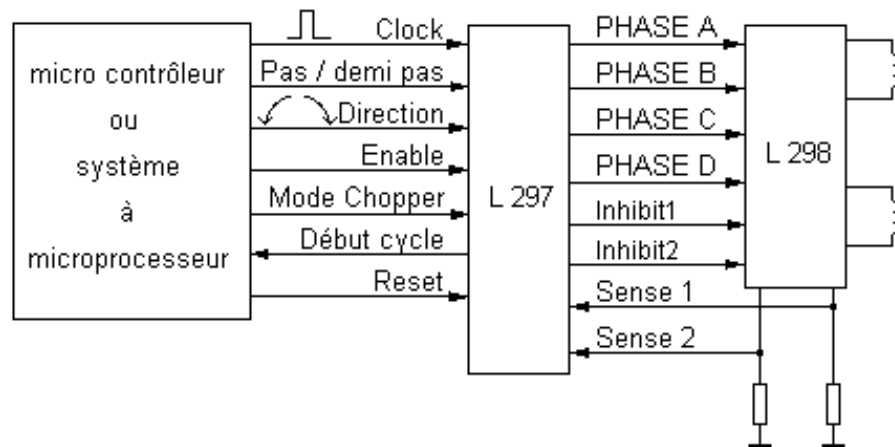


FIGURE III.1 – Schéma de l'association entre les deux cartes

#### III.4.1-a Carte de commande

Le moteur est un moteur commandé par une suite d'impulsions, et grâce à sa technologie et son constitution l'ensemble lui permis de faire une rotation discrète en pas (angle). Donc il sera facile à faire une rotation en angle pré déterminé avec une grande précision.

Le pilote L297 est un circuit spécialisé pour la commande du moteur pas à pas bipolaire. Il suffit de leur indiquer le sens de rotation, le mode pas ou demi pas et bien sur la fréquence des signaux généré pour la commande du moteur en fournissant une horloge provenant du micro contrôleur (Arduino). Il est possible d'utiliser un astable comme le ne555 pour généré le signal d'horloge[15].

#### III.4.1-b Carte de puissance

Le L298 est un hacheur (double pont en H) permis l'amplification des séquences d'impulsions provenant de la carte de commande a pour but d'alimenter les enroulement du moteur selon les signaux de la commande. C'est un module extrêmement utile pour le contrôler de robots et ensembles mécanisés. Il peut contrôler un moteur pas-à-pas bipolaire (4 fil) ou deux moteur courant continu ou chaque MC est équivalent à une bobine dans le moteur pas à pas. Cet hacheur est conçu pour supporter des tensions plus élevées jusqu'à 36 V, et des puissance jusqu'à 25W. Son alimentation externe depend de la valeur nominale de la tension du moteur, mentionné sur sa plaque signalétique [?].

### III.4.2 Réalisation des typons

L'étape suivante consiste à faire le routage avec Ares. Il en découle, l'obtention de trois typons car on a trois cartes à réaliser, une pour réaliser un adaptateur avec les pin de la carte arduino et la deuxième pour l'ensemble du circuit de commande et de puissance et la troisième pour le clavier de manipulation.

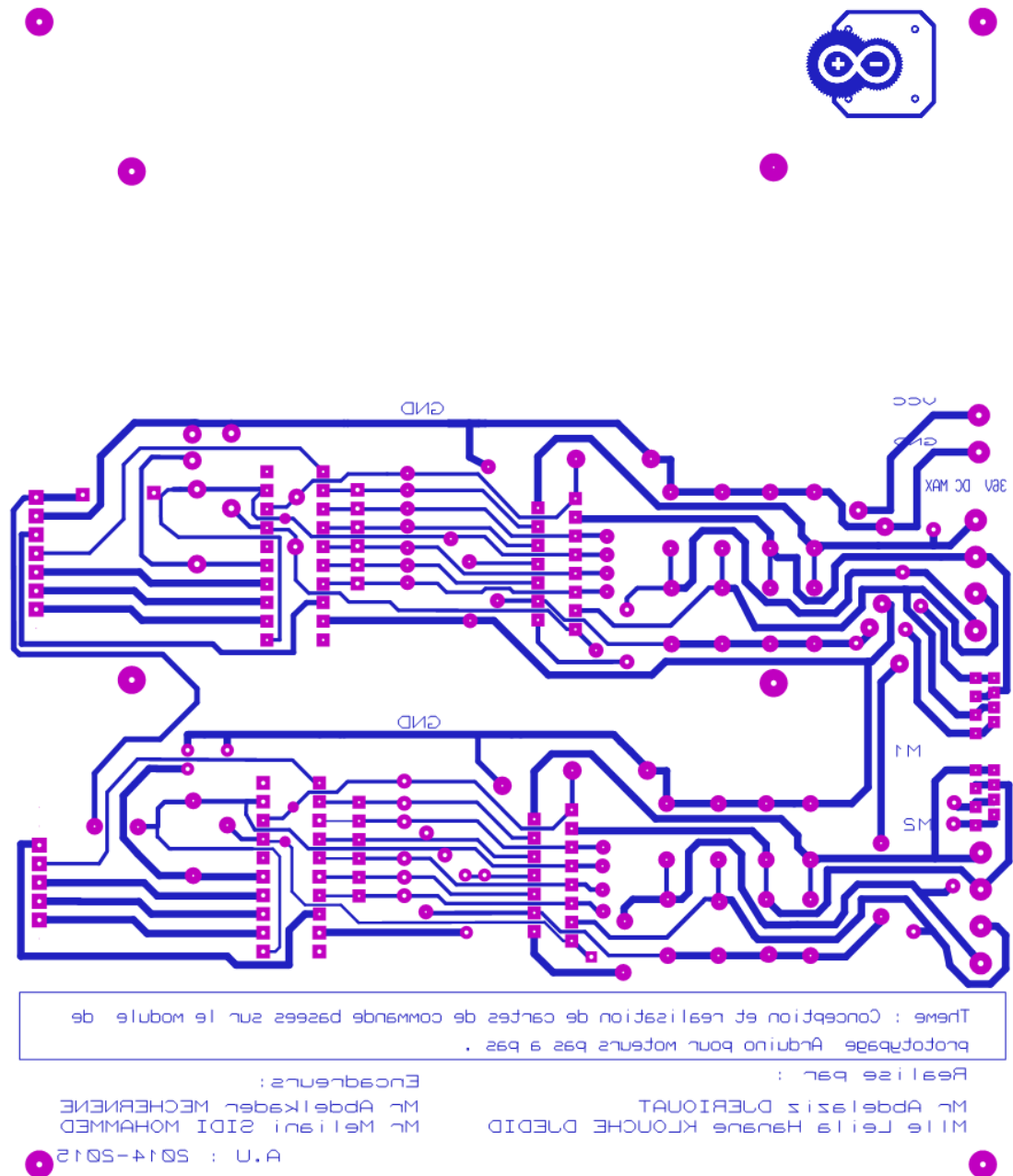


FIGURE III.2 – Typon de la carte de commande et de puissance

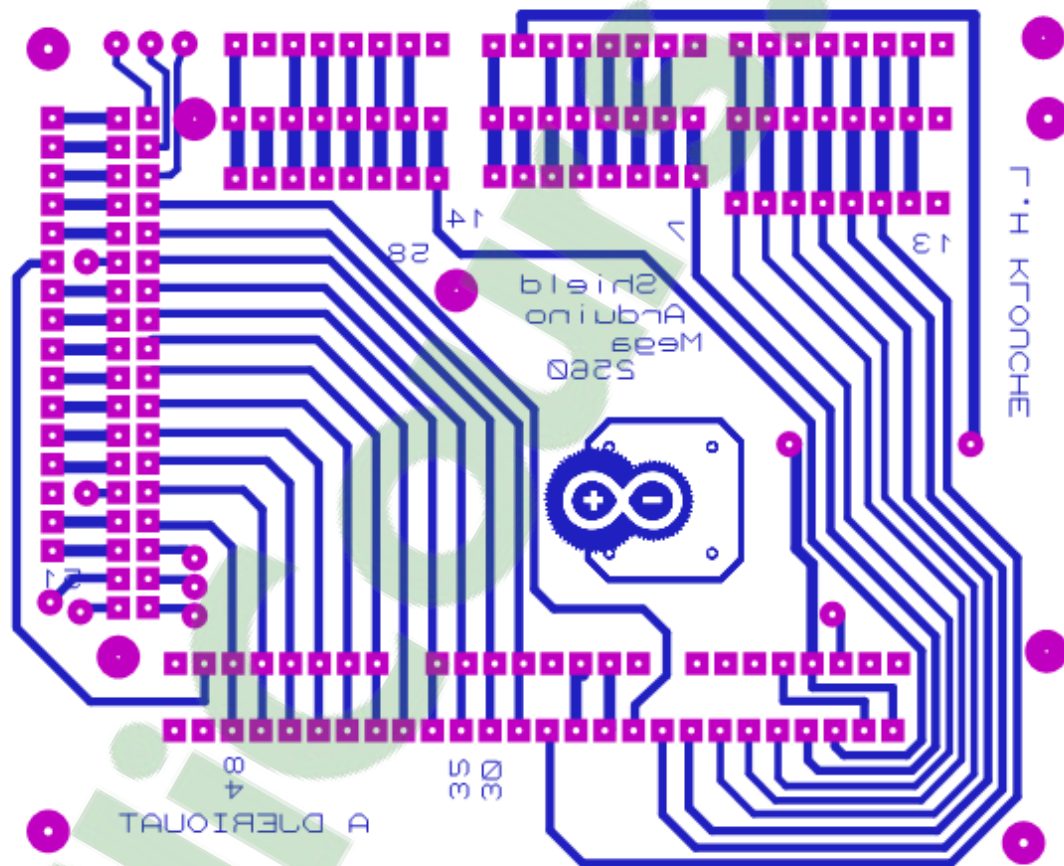


FIGURE III.3 – Typon du circuit adapté avec arduino mega 2560

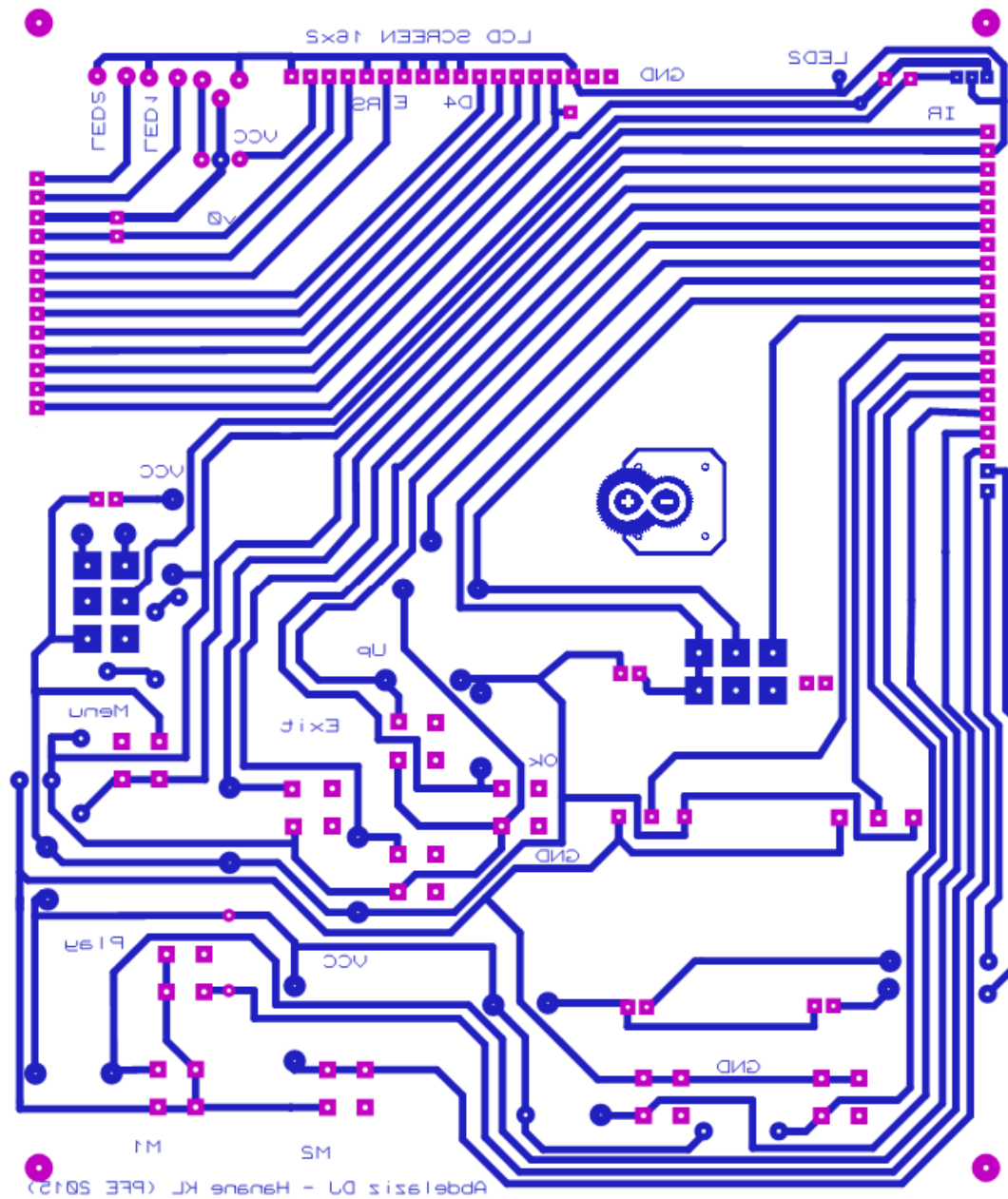


FIGURE III.4 – Typon du clavier de manipulation



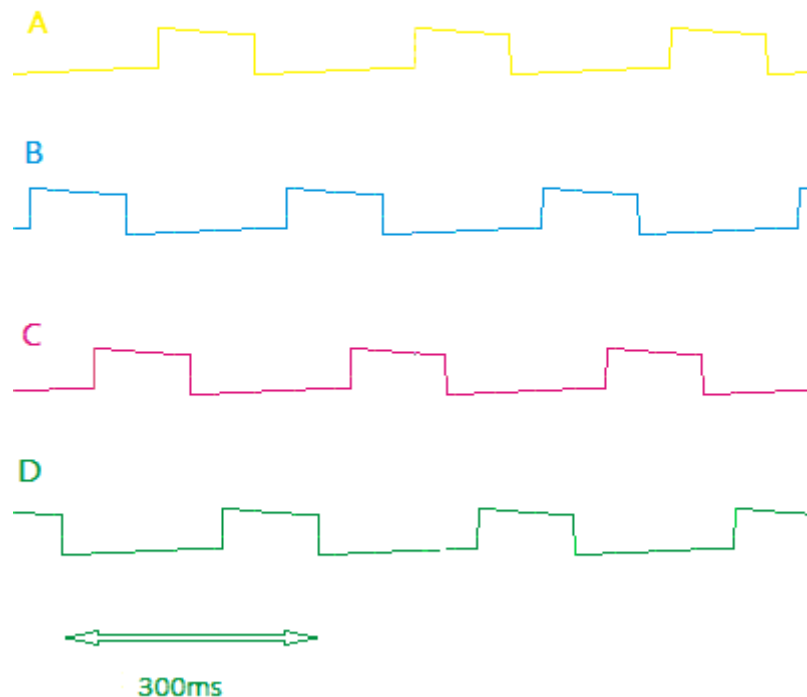


FIGURE III.5 – Allures des tensions A, B, C et D à la sortie du circuit de puissance destiné à contrôler un moteur pas à pas

### III.4.3 Développement de la carte

#### III.4.3-a Partie simulation

A l'aide de l'outil **Isis proteus**, on a fait un système de contrôle du moteur pas à pas avec une carte arduino mega 2560, et les résultats de simulation sont comme la suite :

#### III.4.3-b Partie réalisation des circuit imprimé

La figure [III.10](#) représente le circuit imprimé des deux cartes, carte de puissance et de commande.

### III.4.4 Schéma bloc

La figure suivante illustre notre système contenant une carte de commande composé d'un micro contrôleur Arduino Mega 2560 et un pilote spécialisé pour contrôler le moteur pas à pas et une carte de puissance (Hacheur L298).

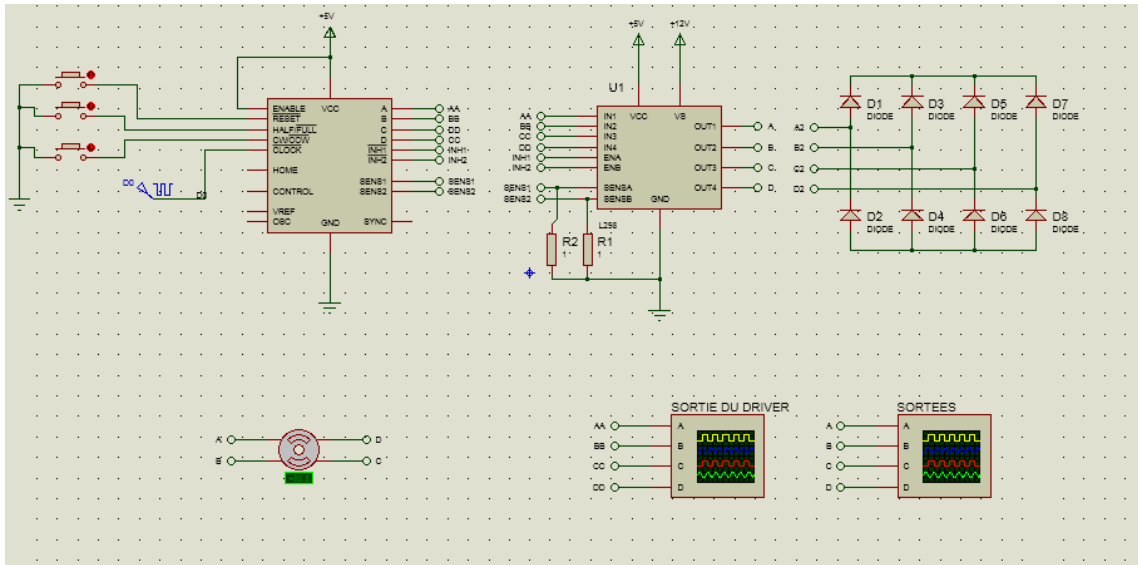


FIGURE III.6 – Montage de simulation à l'aide d'une orloge à la place d'arduino

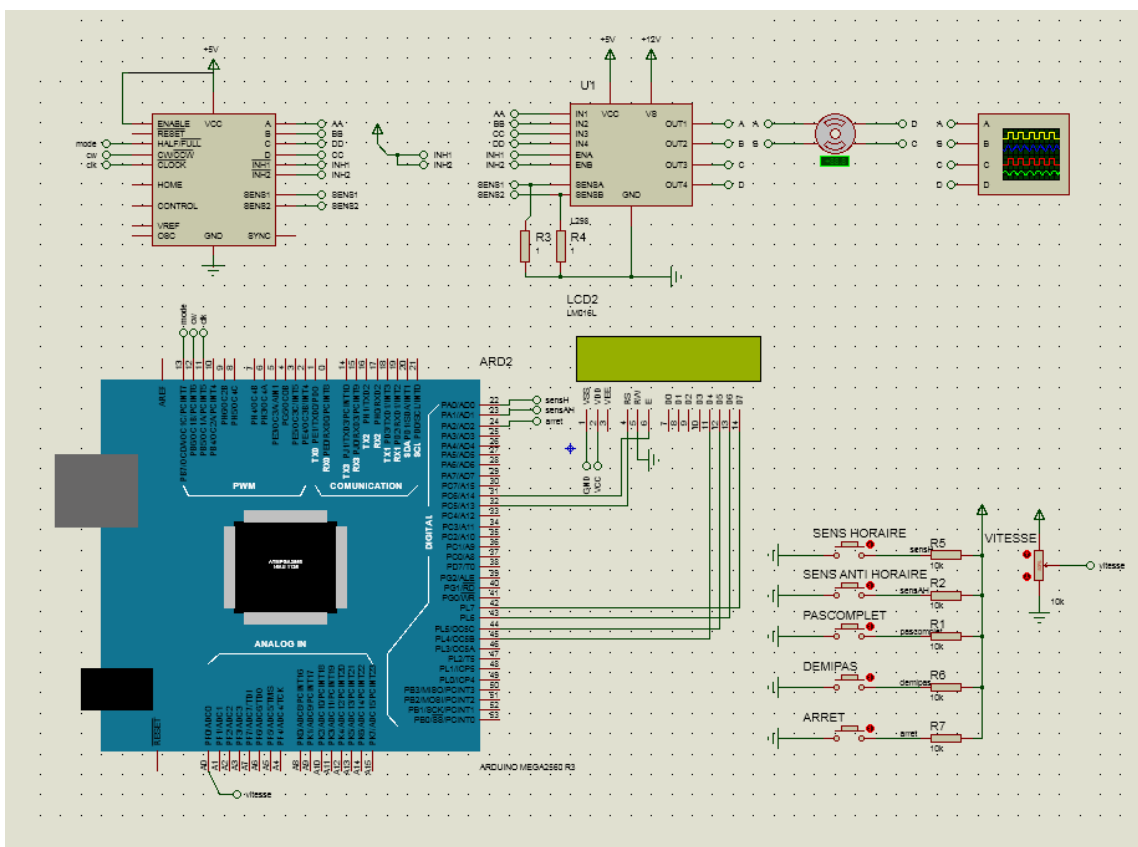


FIGURE III.7 – Simulation de la carte de commande d'un moteur pas à pas

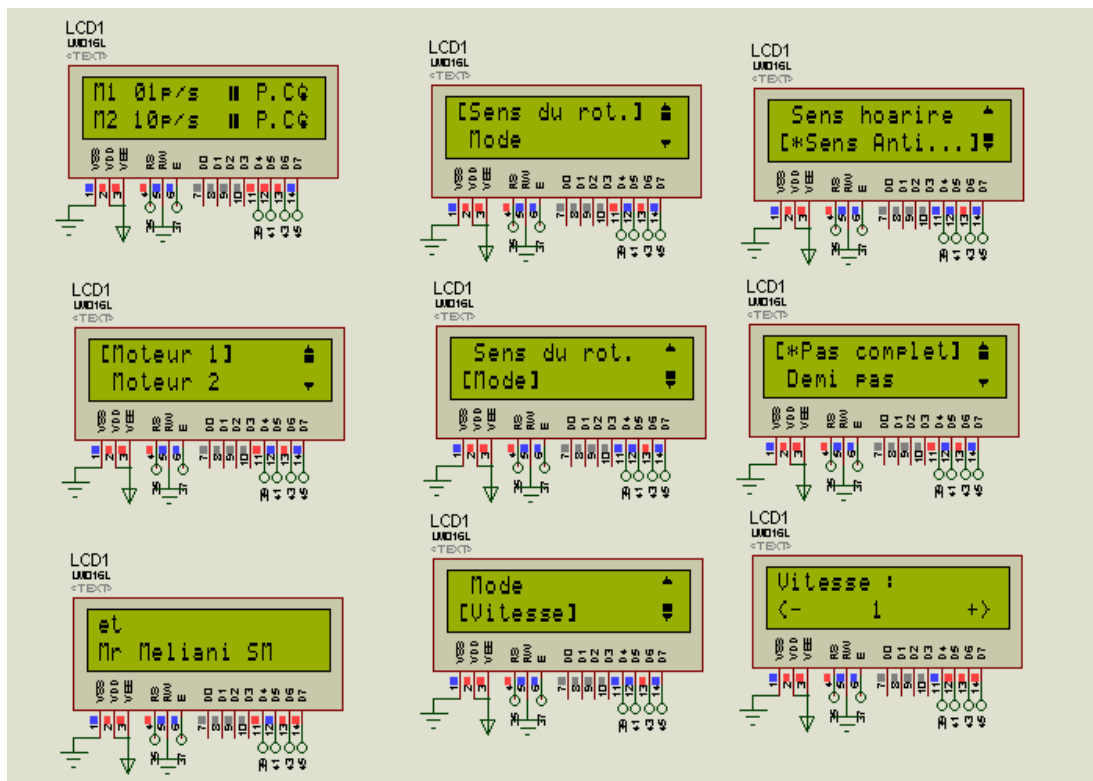


FIGURE III.8 – Informations du menu principale du projet à afficher sur l'écran LCD

### III.4.5 Commande d'un seul moteur

#### III.4.5-a Généralisation des impulsions de la commande

Impulsions de la commande Notre carte de commande composé d'une carte arduino et un driver L297 spécialisé pour le contrôle des moteur pas à pas nous permis d'avoir aisément les impulsion nécessaire pour faire tourner un moteur pas à pas. Il suffit juste de préciser sa fréquence et le sens du rotation du moteur en mettant 1 ou 0 dans la pin du sens de rotation dans le pilote L297.

Il est même possible de faire tourner le moteur en mode demi-pas en mettant 1 dans la pin du mode (pas, ou demi pas).

On a choisit l'usage du driver pour l'isolation (pour ne pas bruler la carte arduino en cas du problème au niveau de la carte de puissance), mais il est possible d'utiliser des lignes de programme à la place du pilote L297 pour le compenser en générant des impulsions similaires destinées à la commande de ce type de moteur.

Les impulsions provenant de la carte de commande ont une faible différence de potentielle. Dans la plus part des cas cette tension inférieure à la tension nominale du moteur ciblé, donc elle ne suffit pas pour le faire tourner. La carte de puissance et à travers une alimentation extérieur permet l'alimentation du moteur en pris en compte la nature des impulsions de la carte de commande.

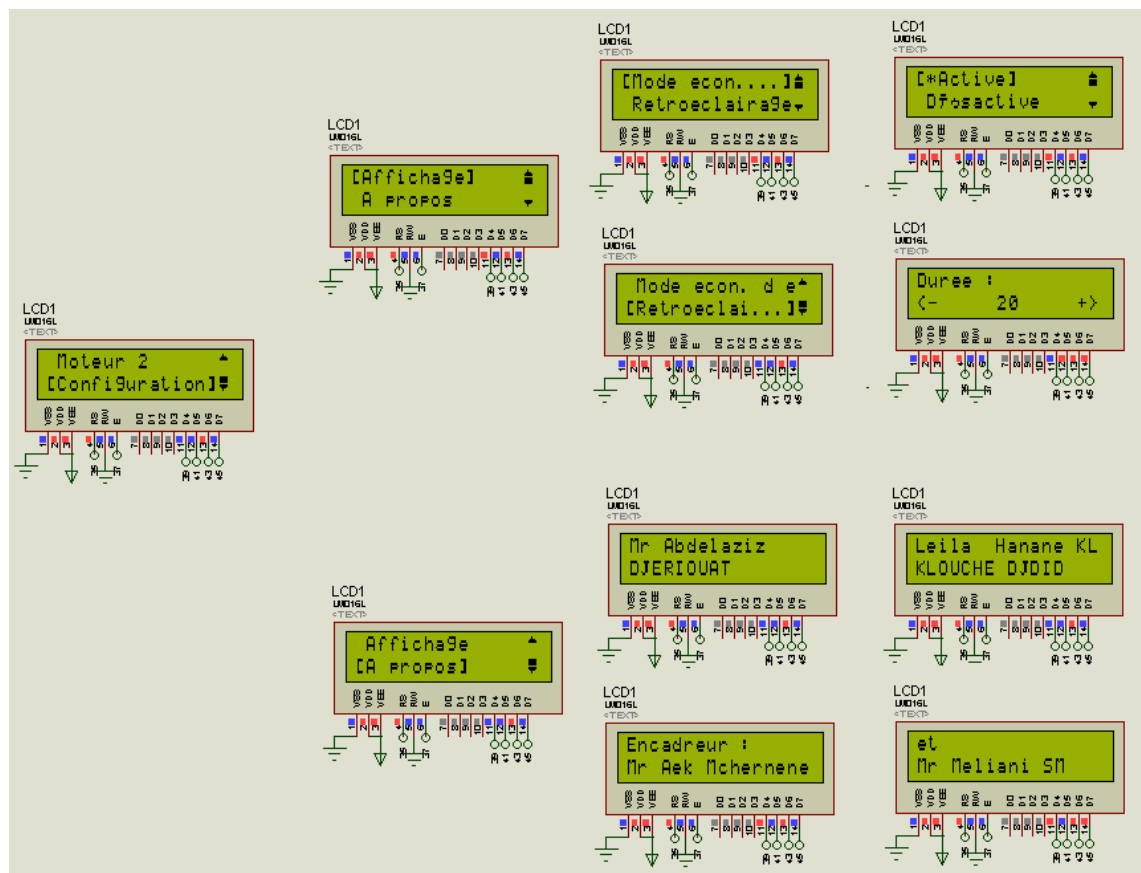


FIGURE III.9 – Le reste des information à afficher

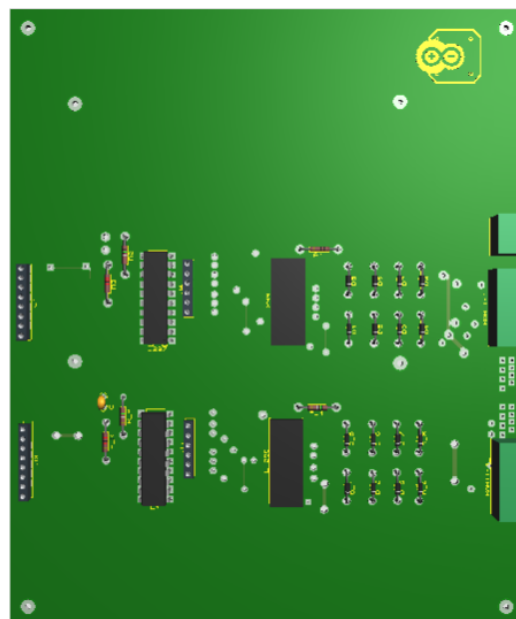


FIGURE III.10 – Circuit imprimé de carte de commande et de puissance

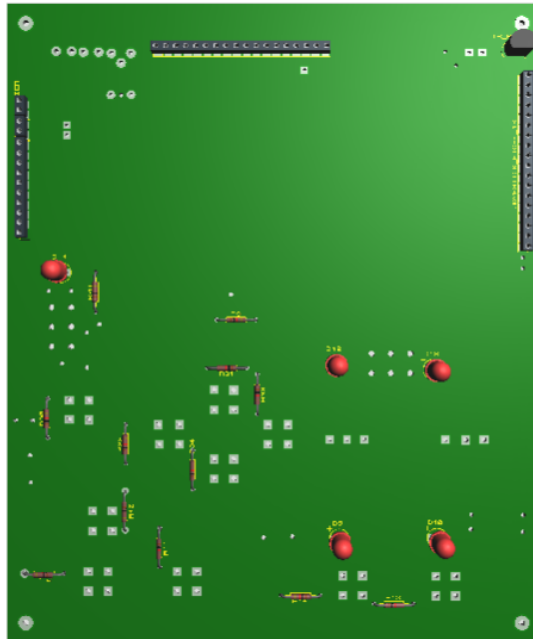


FIGURE III.11 – Clavier de minupilation

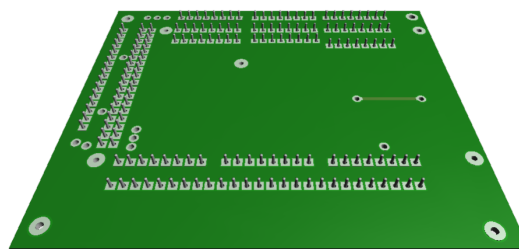


FIGURE III.12 – Circuit imprimé adapté avec arduino mega 2560

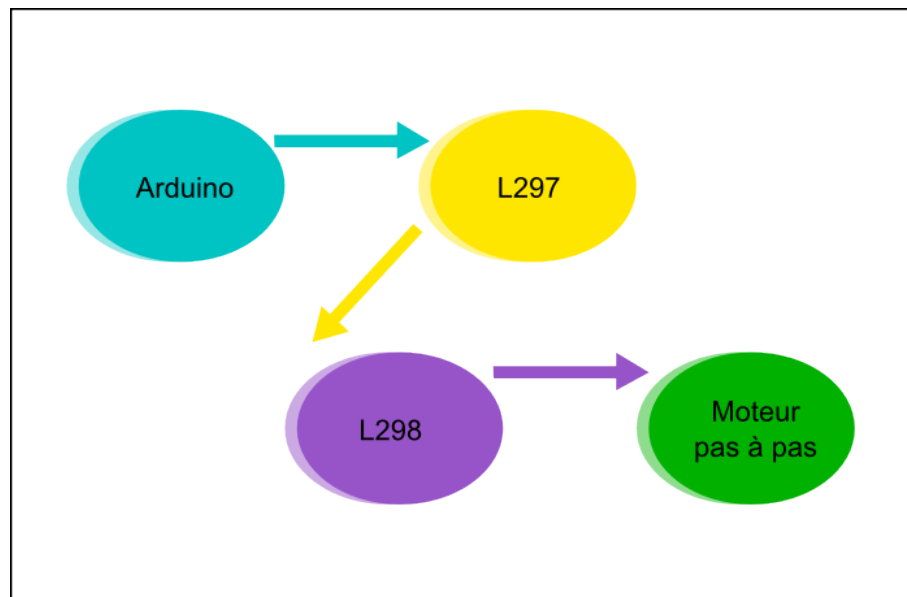


FIGURE III.13 – Schéma bloc simplifié du système

#### Fréquence max des signaux d'impulsions

Chaque moteur pas à pas et d'après ces caractéristique possède une fréquence maximale dont à l'application des fréquences supérieur il ne tourne pas convenablement ou il ne tourne plus. Les moteurs qui sont à notre disposition ont une fréquence maximale généralement inférieure à 50000Hz ( $T=20\mu s$ )

#### Amplitude des impulsions

Les impulsions ont une très basse tension à la sortie de la carte de commande. Donc le rôle mmmmm de la carte de puissance est d'amplifier l'amplitude des ces impulsions à l'aide d'une alimentation externe qui dépend de la valeur nominale de la tension du moteur à commandé. Il ne faut pas dépasser le courant maximal (2A dans notre cas) et la puissance maximale de la carte de puissance (25 W).

### III.4.6 Commande de deux moteurs pas à pas avec afficheur LCD

les mote pas à pas sont devenu généralement dans un grand croissance dans les domaines de l'industrie et même dans la vie quotidienne on trouve plain d'applications des ces types de moteurs. On a fait le cotrôle de deux moteurs pas à pas afin de pouvoir les utilisé dans une application donnée, et on a inclure un afficheur LCD 16x2 (2 lignes, 16 caractères) pour qu' on puisse afficher les informations qui ont relation avec tout le système. le contrôle d'un seul moteur ou de deux moteur ou plus ne ressemble aucun différence ou le nombre des moteurs à commandés n'est jamais un problème.

L'application des moteurs qui tournent à la même vitesse (même consigne de commande) est équivalent à l'application d'un seul moteur ou on est besoin

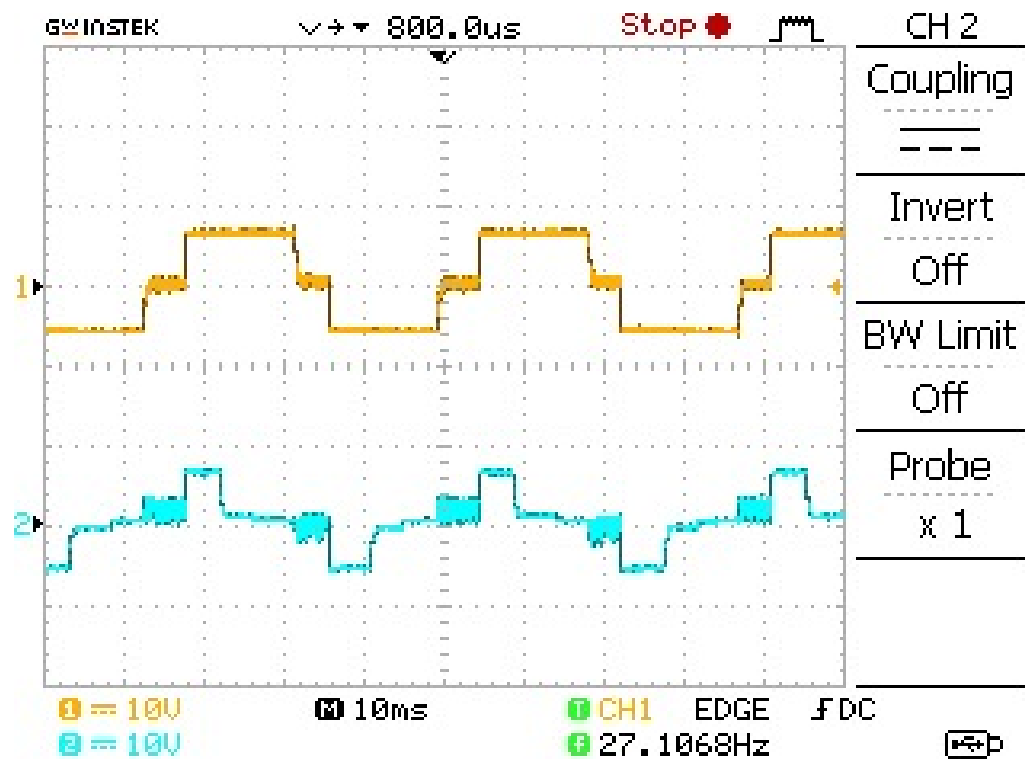


FIGURE III.14 – Allures des tensions entre une borne de chaque enroulement d'un moteur pas à pas et le Gnd

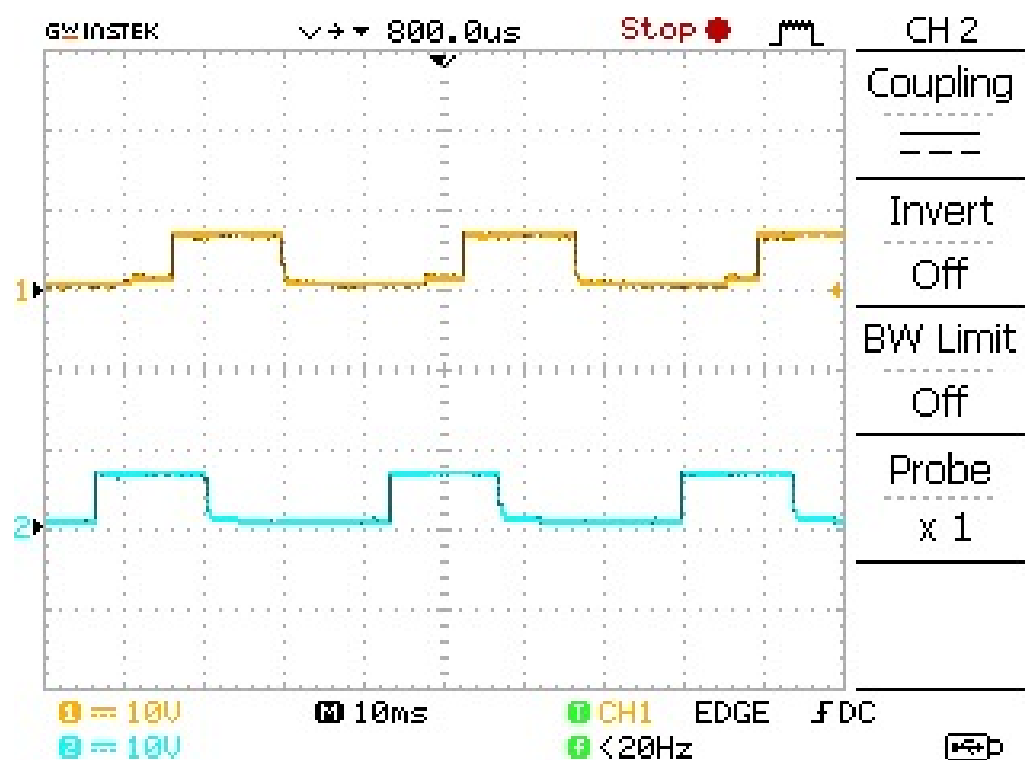


FIGURE III.15 – Alures des tensions aux bornes des deux enroulement d'un moteur pas à pas bipolaire

que d'élargir notre carte de puissance pour qu'il puisse alimenter le reste des moteurs. Le seul problème qui se pose est comment faire tourner chaque moteur à une vitesse donnée et faire associer une consigne différence à chacun de ces moteurs.

La solution est de faire des modifications aux niveaux de la carte de commande côté programme, ou on utilise des compteurs de compteurs avec un compteur pour chaque moteur. Le rôle de ces compteurs est l'incrémentation en fonction du temps afin de générer des séquences d'impulsions à fréquences suivables avec les consignes de vitesse de chaque moteur.

### III.4.7 Programmation du microcontrôleur

#### III.4.7-a Programme

Les lignes de programme permettant la mise en marche de notre projet sont rédigées sous l'environnement de développement Arduino IDE

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(35, 37, 39, 41,43,45);

#include <IRremote.h>
int RECV_PIN = 51;

#include <EEPROM.h>

int ROM_SensM1 = 0 ; // variable qui indique
le sens de la rotation du premier moteur

Const int ClockM1 = 6; // pin 18 du L297,
Clock est déclaré dans comme
entier constant &gale à 6 pour faire sortir le signal
d'horloge de la pin numéro 6

const int BtnPpSys = 34 ;           // La touche Play/Pasue
- marche/arret du système

byte SensHoraire[8] = { // matrice de 5x8 bits pour
imprimer un caractère dans l'écran LCD
0b00100,
0b01110,
0b10001,
0b00001,
0b01101,
0b11110,
0b01100,
```



```

0b000000
};

String M[38][5]={
{"Moteur 1","Moteur 2","Configuration"},    // page=00
{"Sens du rot.,"Mode","Vitesse","Caracteristiques",""}, Sel(0 1 2 3)
{"Sens hoarire","Sens Anti Hor."},          // page=02   Sel(0 1)
{"Ok","Choix valide"},                      // page=03   Sel(0 1)
{"Pas complet","Demi pas"},                 // page=04   Sel(0 1)
{"Ok","Choix valide"},                      // page=05   Sel(0 1)
{"Vitesse : ","<- n tr/min +>"},            // page=06   Sel(0 1)
{"Nombres de pas","frequence max"},         // page=07   Sel(0 1)
{"Nobre de pas :", "n pas/s"},              // page=08   <-   +>
{"Vitesse max:tr/s","n pas/s"},             // page=09   <-   +>
{" "},                                     // page=10   vide

{"Sens du rot.,"Mode","Vitesse","Caracteristiques",""}, Sel(0 1 2)
{"Sens hoarire","Sens Anti Hor."},          // page=12   Sel(0 1)
{"Ok","Choix valide"},                      // page=13   Sel(0 1)
{"Pas complet","Demi pas"},                 // page=14   Sel(0 1)
{"Ok","Choix valide"},                      // page=15
{"Vitesse : ","<- 5 tr/min +>"},            // page=16   Sel(0 1)
{"Nombres de pas","frequence max"},         // page=17
{"Nobre de pas :", "n pas/s"},              // page=18
{"Vitesse max:tr/s","n pas/s"},             // page=19   <-   +>
{" "},    // page=20 <-   +>

{"Affichage","A propos"},                  // page=21   Sel(0 1)
{"Mode econ. d'ecran", "Retroeclairage"},  // page=22   Sel(0 1)
{"Active", "Desactive"},                   // page=23   Sel(0 1)
{"Ok","Choix valide"},                     // page=24   msg
{"      1 min      "},                     // page=25   <-   +>
{"Ok","Choix valide"},                     // page=26   Sel(0 1)
{"PFE realise par :", "Mr Abdelaziz DJERIOUAT et
Mlle Layla Hanane KLOUCHE DJDID"},
};

// On a onlevé les "<é"> parce que notre afficheur ne peut pas les
//afficher correctement
// int page=0, selected=0 ; int Menu = 0 ;
// déclaration des variables essentielle pour l'indexation dans
//le menu

void setup()
{

```

```

// initialisation -----

lcd.begin(16,2); // initialisation de l'afficheur
lcd 16 caractères, 2 lignes

irrecv.enableIRIn(); // initialisation du recepteur
Infra rouge

analogWrite(A15, 500) ; // LCD_SCREEN_LED ENABLE
(mode économiseur dans le cas du 0 en sortie)

// sortie analogique relié avec la led du fond de
l'afficheur LCD.

/*
Configuration des entrées et des sorties
*/
pinMode(ClockM1, OUTPUT);
pinMode(sensM1, OUTPUT);

pinMode(BtnPpSys,INPUT); // Play pause pour le système
(Moteur1 & moteur2)

lcd.createChar(4, SensHoraire); // addition du
caractère du sens horaire à l'afficheur LCD

fond() ; // cette fonction affiche les informations
de base du système du fond (bureau)

// -----
// Bureau > | M1 10Pas/s >> P.C > |
//          | M2 20Pas/s || D.P -> |
// -----
}

void PlayPauseSystem()
{
    if(EEPROM.read(ROM_StatM1)==1 || EEPROM.read(
ROM_StatM2)==1)
// si l'un des 2 moteur est en marche donc le
système va arreter
    {
        EEPROM.write(ROM_StatSys,0);

```

```
        EEPROM.write(ROM_StatM1,0);
        EEPROM.write(ROM_StatM2,0);
    }
    else
// Si non le systeme sera mis en marche s'il est à
l'arret et vis vers sa
    {
        EEPROM.write(ROM_StatSys,!EEPROM.read(ROM_StatSys));
        EEPROM.write(ROM_StatM1,EEPROM.read(ROM_StatSys));
        EEPROM.write(ROM_StatM2,EEPROM.read(ROM_StatSys));
    }
    fond();
}

void PlayPauseM1()
{
    EEPROM.write(ROM_StatM1,!EEPROM.read(ROM_StatM1));
    fond();
}

void ChangeSensM1()
{
    EEPROM.write(ROM_SensM1,!EEPROM.read(ROM_SensM1));
    fond();
}

void ChangeStatClock1()
{
    digitalWrite(Clock,!digitalRead(Clock));
    // ineverser l'état de la sortie nommé Clock
}

float getTemperature()
// température provenant d'un capteur LM35 placé
à coté de l'hacheur pour lancer un alerte en cas de
détection des valeurs de température inattendus
{
    tensionAn = (((sensorValue / 10.23) * 5)) ;
    return tensionAn ;
}

void alertHotElectronicCoponent()
{
//    Temperature pour L298 dans le datasheet
//    max que support l'hacheur est de 75°C
```

```

PlayPauseSystem(); // l'arret du système on

noTone(tonePin);
  delay(200);
  tone(tonePin, 523, 300);
  delay(60);
}

int CounterClock1=0, CounterClock2=0;
void every_us() // exécuter ces instructions chaque 1 ms
{
  int SpeedM1Value = EEPROM.read(ROM_SpeedM1) ;
  // Obtention de la période en microsecond
  int SpeedM2Value = EEPROM.read(ROM_SpeedM2) ;
  // Obtention de la période en microsecond

  //float MySpeed = 60 ; // 1pas/sec;

  int pas_du_compteur = 10; // 10us
  int demiPeriodeClk1 = 240 / SpeedM1Value
* pas_du_compteur ;
  int demiPeriodeClk2 = 240 / SpeedM2Value
* pas_du_compteur ;
  delayMicroseconds(pas_du_compteur);

  CounterClock1++; CounterClock2++;

  // demiPeriodeClk1 = 5000 ;
  if(CounterClock1==(demiPeriodeClk1) && demiPeriodeClk1!=0 )
if(EEPROM.read(ROM_StatM1)==1) ChangeStatClock1(); else {}
else if(CounterClock1>demiPeriodeClk1) CounterClock1=0;
  if(CounterClock2==(demiPeriodeClk2) && demiPeriodeClk2!=0)
if (EEPROM.read(ROM_StatM2)==1) ChangeStatClock2(); else {}
else if(CounterClock2>demiPeriodeClk2) CounterClock2=0;
}

void loop()
{
  every_us(); // cette fonction s'exécute à chaque
micro seconde
  Actualiser_Les_Entrees();
  // cette fonction vérifier les entrées (bouttons et
capteurs) à chaque instant
}

```

```

void generateNewMenuList(boolean IsAList )
// actualiser la liste après chaque changement de la
selection ou de la page
{
    String TextToPrint = M[page][selected] ;
    ...
}

void incrementer(String MSG_L1,int EepromAdress)
{
    lcd.clear(); lcd.setCursor(0, 0);
// Vider l'écran et positionner le curseur dans
la 1ère ligne dans la position 0
    lcd.print(MSG_L1);
// Afficher le message Vitesse
    int Param = EEPROM.read(EepromAdress);
// Lire la valeur de la vitesse enregistrée dans
la mémoire ROM
    Param++ ;
// Diminuer la vitesse
    EEPROM.write(EepromAdress,Param);
// Sauvegarder la nouvelle valeur de la vitesse
dans la mémoire ROM
    espace_1 = ((12 - (String(Param).length() ))/2);
// Calcule des nombre d'espaces pour afficher la
vitesse dans le centre de l'afficheur
    if( (espace_1 * 2 ) + 4 + String(Param).length() ==16 )
    espace_2 = espace_1 ; else  espace_2 = espace_1+1 ;
// si le nombre d'espace est impaire On le divise
en 2 parties, Une inférieure que l'autre de 1
    lcd.setCursor(0, 1);
// Positionner le curseur de l'afficheur LCD dans
la position 0 de la 2ème ligne
    lcd.print(String("<-") + FreeScreen.substring(0,espace_1)
+ EEPROM.read(EepromAdress) + FreeScreen.subst-
ring(0,espace_2) + String(">")); // Afficher la vitesse
}

void decrementer(String MSG_L1,int EepromAdress)
{
    ...
}

void LireLavaleurDeSauvegarde(String MSG_L1,int EepromAdress)
{

```

```
...
}

void inverserUnChoix(int pin, int value) // inverser
un sortie numérique
{
digitalWrite(pin,value);

}

void menu()
{
    if(Menu ==0 ) // si le menu n'est pas encore exécuté
    {
        page=0; selected=0; generateNewMenuList(1) ; Menu = 1 ;
    }
    else
    {
        exit(); // quitter le menu si on est dans le menu
principale et faire fonctionner comme la touche exit
si on est dans un sous menu
    }
}

void ok()
{
    switch(page)
    {
        case 0 :
        if(Menu==1)
        {
            switch(selected)
            {
                case 0 : page=1 ;break ; // moteur 1 est sélectionné
                case 1 : page=11 ;break ; // moteur 2 est sélectionné
                case 2 : page=21 ;break ; // Configuration est
selectionnée
            }
            selected = 0 ;
            generateNewMenuList(1) ;
        }
        ;break ;
        case 1 :      ...
    }
}
```

```

void fond()
{
    int SpeedM1 = EEPROM.read(ROM_SpeedM1) ;
    int SpeedM2 = EEPROM.read(ROM_SpeedM2) ;
    String SpeedToPrintM1,SpeedToPrintM2;

    if(String(SpeedM1).length()==1) SpeedToPrintM1 = "0"
+ String(SpeedM1);
else SpeedToPrintM1 = String(SpeedM1);
    if(String(SpeedM2).length()==1) SpeedToPrintM2 = "0"
+ String(SpeedM2);
else SpeedToPrintM2 = String(SpeedM2);

    String TextToPrintInL1 = "M1 " + SpeedToPrintM1 + "p/s" ;
    String TextToPrintInL2 = "M2 " + SpeedToPrintM2 + "p/s" ;
    ...

    lcd.clear();
    lcd.print(TextToPrintInL1);

    setCursor(0,1);
    lcd.print(TextToPrintInL2);
}

void exit()
{
    switch(page)
    {
        case 0: lcd.clear(); lcd.setCursor(0, 0); fond() ;
Menu = 0 ; break ;
        case 1: page=0; selected=0; generateNewMenuList(1) ;
break ;
        case 2: page=1; selected=0; break ;
        case 3: page=2; selected=EEPROM.read(ROM_SensM1);
/* la selection depend du eeprom ou d'1 variable de
description d'état lors le msg "Ok, Choix Validé !"
est disparu */ break ;
        ...
    }
}

void About()
{

```

```
    lcd.clear();  
    lcd.print("Pfe realise par :");  
    ...  
}
```

## III.5 Conclusion

Ce projet consiste à commander un seul moteurs puis deux moteurs pas-à-pas à travers une carte numérique de prototypage Arduino, en vue de les utiliser plus tard en travaux pratiques. Dans ce chapitre La commande est entièrement assurée par la programmation de Arduino de façon à répondre exactement à nos besoins. On a utilisé des petits systèmes électroniques associés à la programmation ce qui concerne le microcontrôleur avec l'électronique afin de pouvoir arriver au buts du projet.

Nous avons choisis la carte Arduino Mega 2650, vu ces caractéristiques et sa simplicité d'utilisation, comme déjà mentionné dans le chapitre précédent. Dans la partie de commande on a utilisé un hacheur L298 qui contient deux pond en H pour amplifier l'amplitude des signaux du circuit de commande qui vont attaquer les moteurs pas à pas.



---

## Conclusion générale

Il est donc le temps de conclure sur l'ensemble des travaux présentés dans ce mémoire : chaque un des chapitres a permis d'avoir une solution à la problématique essentielle de ce mémoire, et qui est la difficulté d'implantation d'un système de contrôle du moteur pas à pas. C'est pour ça qu'on a utilisé des circuits intégrés spécialisés.

Dans le premier chapitre on a parlé sur le moteur pas à pas, ces différents types, et les circuits spécialisés de leur commande. Nous avons aussi mentionné la présence aujourd'hui indispensable du moteur pas à pas dans de nombreux systèmes industriels. La grande précision en position du moteur pas à pas lui permet d'être favorable, dans la plus part des applications automatisées.

La combinaison entre des différents domaines (électrique, mécanique, automatique et information) a causé la naissance de nouvelles technologies qui facilitent la mise en œuvre et la création et la réalisation des nouveaux systèmes destinés qui nous facilitent la vie.

Dans le deuxième chapitre on a parlé sur les projets arduino, son historique et ces cartes à base microcontrôleur Atmel. On a parlé aussi sur la différence entre arduino Mega et le Uno.

Dans le troisième chapitre on a fait la réalisation pratique du projet contenant deux moteurs pas à pas et un afficheur LCD 2 lignes, 16 caractères. On a utilisé un circuit intégré spécialisé pour le contrôle du moteur pas à pas (L297) et deux ponts en H (hacheur L298). Le contrôle de vitesse et de position, avec l'usage d'un afficheur LCD pour afficher les paramètres instantanés internes du système (vitesse, sens de rotation, mode de rotation) contenant deux actionneurs.

Notre projet contient deux cartes, une carte de commande qui se compose d'une carte de prototypage arduino mega 2560 et un pilote spécialisé pour piloter un moteur pas à pas et une carte de puissance composée de deux ponts en H et une alimentation externe dépend de la valeur nominale du moteur à contrôler. La

carte de puissance amplifie les signaux qui forment des séquences d'impulsions pour attaquer le moteur pas à pas. Ces impulsions sont des ordres électriques destinés à l'actionneur pour qu'il fasse agir et tourner en pas selon les informations envoyées à travers ces impulsions. Il est possible de ne pas utiliser des circuits spécialisés pour le contrôle du moteur, mais il est favorable de le faire juste pour l'isolement.

Ce mémoire a permis d'exposer la carte de prototypage arduino mega 2560, et leur facilité dans la réalisation des différentes applications. Dans notre cas on a fait le contrôle de deux moteurs pas à pas. On a ajouté un capteur de température LM35 pour contrôler nos composants de la carte de puissance qui ont le risque de chauffer dans le cas du montage d'un moteur plus puissant. Une baffe sonore est intégrée pour déclencher une alerte en cas d'augmentation de température des interrupteurs commandés dans la carte de puissance.

Puisque le développement des applications à l'aide d'un microcontrôleur arduino et avec l'interface de développement Arduino IDE (Interface Developing Environment), il est facile d'ajouter de nombreux éléments et fonctions à votre projet. Donc nous avons ajouté un capteur infrarouge numérique (photo transistor) à notre projet afin de recevoir des signaux d'une télécommande infrarouge pour pouvoir contrôler le système à distance.

---

## Bibliographie

- [1] Arduino. Mémoire de D.E.A., 2011-12-22.
- [2] M. CORREVON : Systèmes électromécaniques. Mémoire de D.E.A., Haute Ecole d'Ingénierie et de Gestion Du Canton du Vaud, 2002.
- [3] COURS : Les moteurs pas à pas. [www.cours.etsmtl.ca/gpa668/aCours/GPA668-Cours13b.pdf](http://www.cours.etsmtl.ca/gpa668/aCours/GPA668-Cours13b.pdf), consulté en 2015.
- [4] Olivier DESPONT : Kit arduino. [www.semageek.com/](http://www.semageek.com/), mise en ligne 2009, consulté en 2015.
- [5] ESKIMON : Les différents moteurs pas-à-pas et leur fonctionnement. [www.eskimon.fr/290-arduino-603-petits-pas-le-moteur-pas-pas](http://www.eskimon.fr/290-arduino-603-petits-pas-le-moteur-pas-pas), mise en ligne 2014 ,consulté en 2015.
- [6] olyte ESKIMON : Arduino pour bien commencer en électronique et en programmation. [www.openclassrooms.com/](http://www.openclassrooms.com/), Dernière mise à jour le 2/12/2012, consulté en 2015.
- [7] FREE : circuits de commande du moteur pas à pas. [www.omega.ca/prodinfo\\_fr/Stepper\\_Motors.html](http://www.omega.ca/prodinfo_fr/Stepper_Motors.html), mise en ligne 2003 ,consulté en 2015.
- [8] Marc JAKUBOWICZ : Moteurs pas à pas. [www.moodle.lp-andre-argouges.ac-grenoble.fr/pluginfile.php/143/mod\\_resource/content/6/moteurPP3.htm](http://www.moodle.lp-andre-argouges.ac-grenoble.fr/pluginfile.php/143/mod_resource/content/6/moteurPP3.htm), consulté en 2015.
- [9] la ELECTRICITE : Séquence de commande du moteur pas à pas. [www.la-electricite.com/moteur-pas-pas/](http://www.la-electricite.com/moteur-pas-pas/), Mise en ligne 2012,consulté en 2015.
- [10] Floss MANUALS : Arduino. [www.fr.flossmanuals.net/arduino/vous-avez-dit-arduino/](http://www.fr.flossmanuals.net/arduino/vous-avez-dit-arduino/), consulté en 2015.
- [11] MDPMOTOR : Motors systems solutions. [www.mdpmotor.fr/documentation/lexique/pas-a-pas/definition.html](http://www.mdpmotor.fr/documentation/lexique/pas-a-pas/definition.html), year = Mise en ligne 2015,consulté le 23 mai 2015,.
- [12] Jean-Noël MONTAGNÉ : Initiation à la mise en oeuvre matérielle et logicielle de l'arduino. Centre de Ressources Art Sensitif.
- [13] Site officiel ARDUINO : Arduino. [www.arduino.cc/](http://www.arduino.cc/), consulté en 2015.
- [14] OMEGA : Choix d'un moteur pas à pas. [www.omega.ca/prodinfo\\_fr/Stepper\\_Motors.html](http://www.omega.ca/prodinfo_fr/Stepper_Motors.html), mise en ligne 2003 ,consulté en 2015.

- 
- [15] positron LIBRE : Séquence de commande du moteur pas à pas. [www.positron-libre.com/electronique/moteur-pas-a-pas/sequence-commande-moteur-pas-a-pas.php](http://www.positron-libre.com/electronique/moteur-pas-a-pas/sequence-commande-moteur-pas-a-pas.php), Mise en ligne 10 août 2011, consulté en 2015.
- [16] Louis REYNIER : C'est quoi arduino ? url [www.louisreynier.com](http://www.louisreynier.com), consulté en 2015.
- [17] WIKIPEDIA : Moteurs pas à pas. [www.fr.wikipedia.org/wiki/Moteur\\_pas\\_%C3%A0\\_pas](http://www.fr.wikipedia.org/wiki/Moteur_pas_%C3%A0_pas), consulté le 12 avril 2015.
- [18] Gerard YVRAUT : Les moteurs pas a pas. *In Séminaire Bellegarde Nov 99*, 1999.

## Annexe A : Documentation technique

### Fiches techniques des CI(s) utilisés (Datasheets)

Datasheet : L297

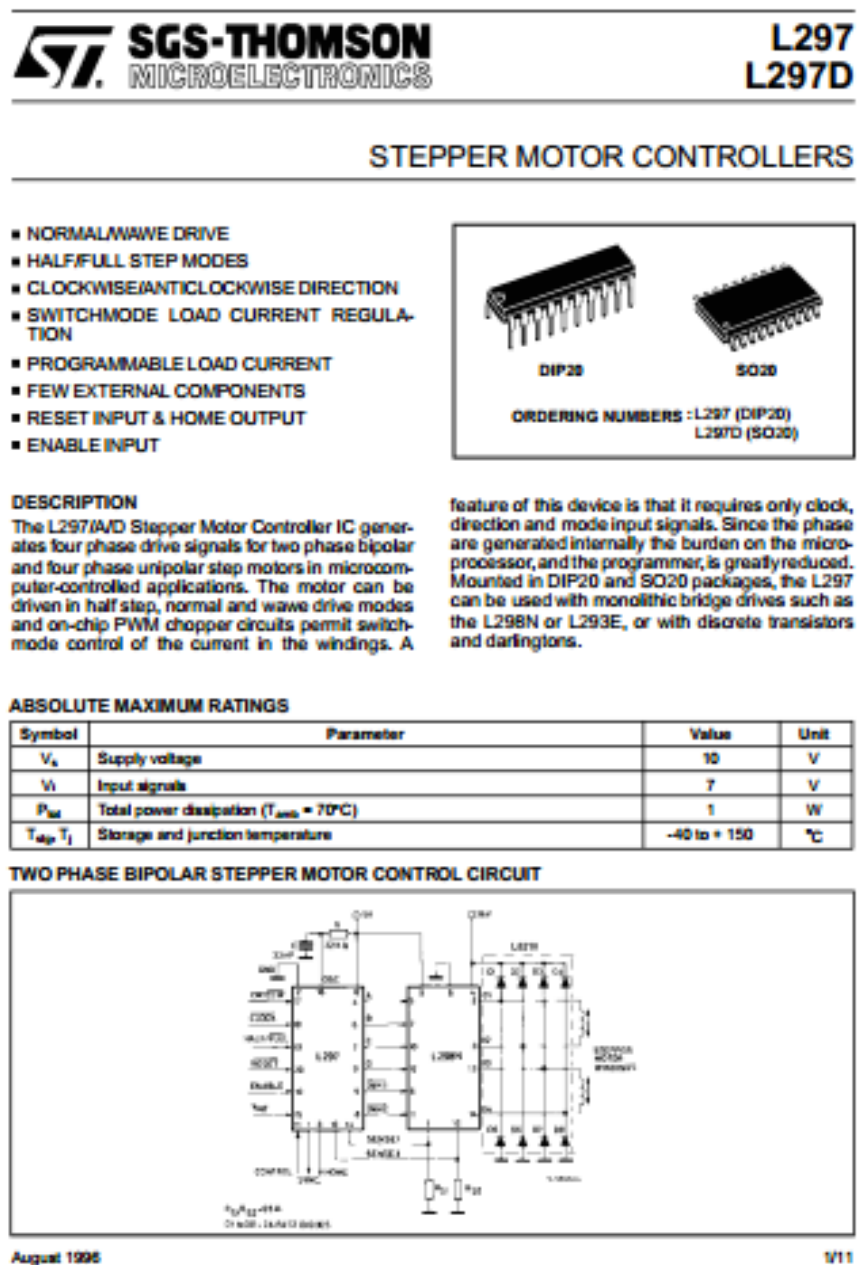


FIGURE A1 – L297

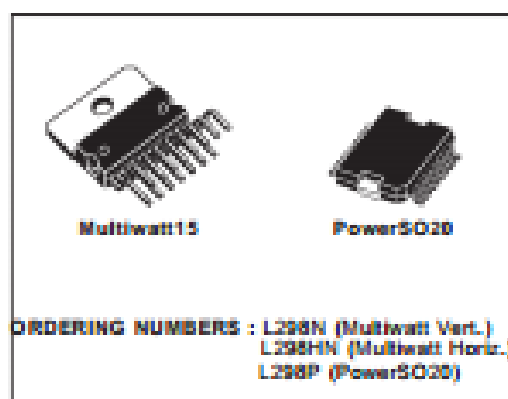
## Datasheet : L298

**L298****DUAL FULL-BRIDGE DRIVER**

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

**DESCRIPTION**

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the con-



nection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

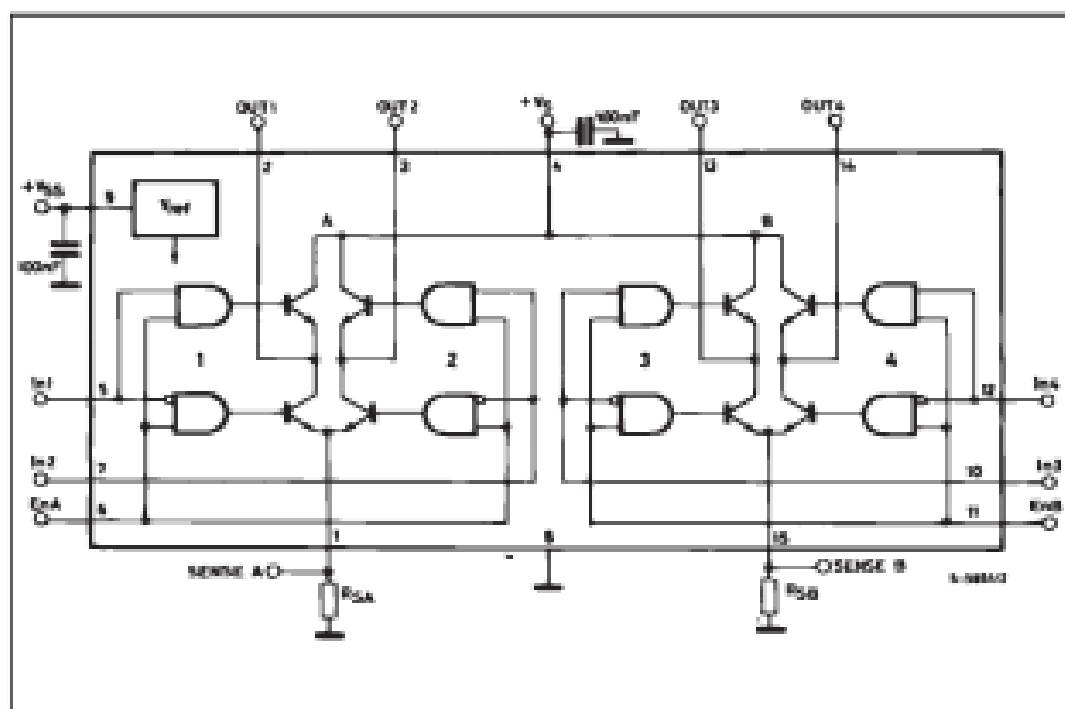
**BLOCK DIAGRAM**

FIGURE A2 – L298

## Association L297-L298

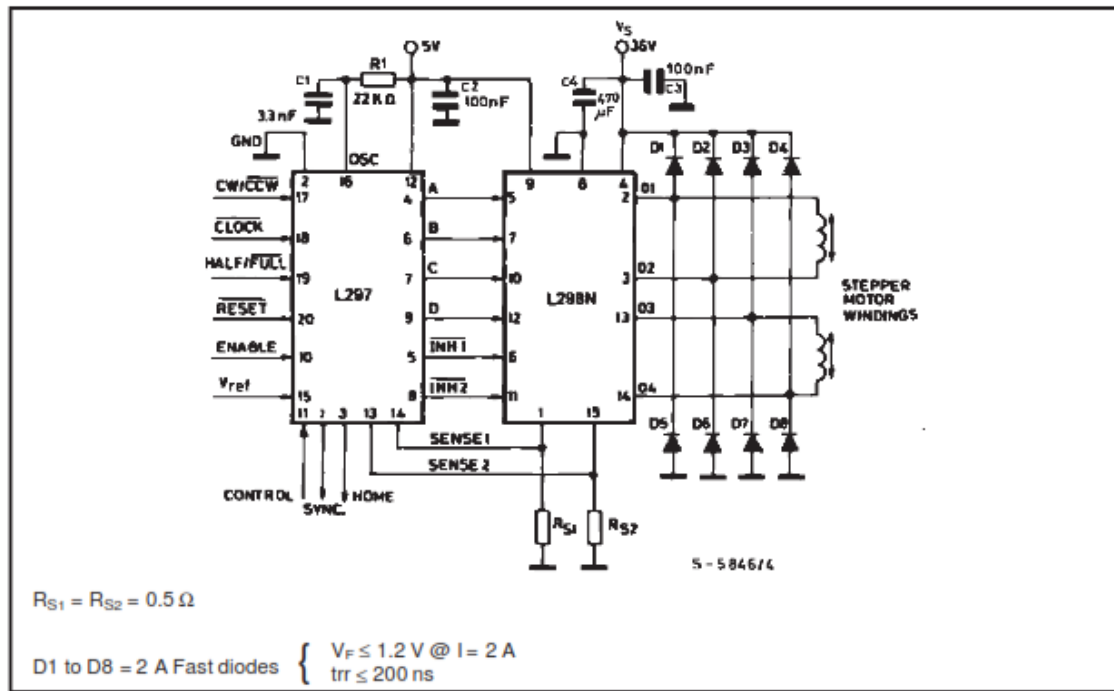


FIGURE A.3 – Two Phase Bipolar Stepper Motor Circuit

---

## Annexe B :

A chaque période du signal d'horloge le pilote du moteur pas à pas inverse le signaux en sortie de tel façon à avoir des séquence d'impulsions suivante :

\* Exemple : mode pas complet, sens horaire :

| <b>Clc</b>  | 1 0 | 1 0 | 1 0 | 1 0 |
|-------------|-----|-----|-----|-----|
| <b>A</b>    | 1   | 0   | 0   | 0   |
| <b>B</b>    | 0   | 0   | 1   | 0   |
| <b>C</b>    | 0   | 1   | 0   | 0   |
| <b>D</b>    | 0   | 0   | 0   | 1   |
| <b>InH1</b> | 1   | 0   | 1   | 0   |
| <b>InH2</b> | 0   | 1   | 0   | 1   |



# CONCEPTION ET RÉALISATION DE CARTES DE COMMANDE BASÉES SUR LE MODULE DE PROTOTYPAGE ARDUINO POUR MOTEURS PAS À PAS

## Résumé

Le présent travail porte de la conception, la réalisation et la mise en œuvre de cartes de commande basées sur un module arduino en vue de contrôler des moteurs pas à pas. Dans un premier temps, l'organisation et le principe de fonctionnement des moteurs pas à pas, l'étude des commandes les plus couramment utilisées pour ce type d'actionneur sont également passées en revue. Puis, l'architecture matérielle, les particularités et la programmation de la carte de prototypage arduino à base de microcontrôleur Atmel 2560 sont détaillées. Les approches de commande retenues, leurs principales fonctions, ainsi que leurs conceptions sont présentées. Les solutions envisagées ont été réalisées et imprimées sur cartes électroniques. Des essais expérimentaux ont été effectués et mettent en évidence les performances de la carte de commande. Les résultats obtenus sont évalués et comparés. Enfin, une conclusion générale, donne une synthèse des principaux résultats et résume l'apport de ce travail. Des perspectives pour de futurs travaux sont finalement proposées.

## Mots clés

*Moteur pas à pas, unipolaire, bipolaire, pas, commande numérique, microcontrôleur, Arduino mega, Atmel 2560, L297, L298, pas entier .*

---

# DESIGN AND IMPLEMENTATION OF CONTROL CARDS BASED ON THE ARDUINO PROTOTYPING MODULE FOR STEPPER MOTORS

## Abstract

The work of this master thesis manuscript deals with the design, realization and implementation of control card based on a module Arduino to controlling stepper motors. Initially, the organization and the principle of operation of stepper motors, the study of the most commonly used controls for this type of actuator are also reviewed. Then, the hardware architecture, features and programming the Arduino prototyping board based on Atmel microcontroller 2560 are detailed. Retained control approaches, their main functions and their designs are presented. The proposed solutions have been made and printed on electronic cards. Experimental tests were performed and demonstrate the performance of the control board. The results obtained are evaluated and compared. Finally, a general conclusion, gives a summary of key findings and summarizes the contributions of this work. The prospects for future work are finally proposed.

## Keywords

*Stepper motor, unipolar, bipolar, step, digital control, microcontroller, Arduino mega, Atmel 2560, L297, L298, Full-step.*