

الجمهورية الجزائرية الديمقراطية الشعبية  
**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE**  
وزارة التعليم العالي و البحث العلمي  
**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**  
جامعة أبي بكر بلقايد – تلمسان  
Université Aboubakr Belkaïd – Tlemcen –  
Faculté de TECHNOLOGIE



## **MEMOIRE**

Présenté pour l'obtention du **diplôme** de **MASTER**

**En : ÉLECTRONIQUE**  
**Spécialité : INSTRUMENTATION**

**Par : TAGHLI MOHAMMED MOUNIR**  
**MOSBAH SID AHMED**

**Sujet**

**Etude et Simulation d'une carte Electronique pour le  
contrôle de la vitesse d'un moteur à courant continu**

Soutenu publiquement, le 22/ 06 /2026 , devant le jury composé de :

|                          |     |                       |              |
|--------------------------|-----|-----------------------|--------------|
| Mr NEMMICHE AHMED        | MCB | Université de Tlemcen | Président    |
| Mr BOURSALI AMIN         | MCB | Université de Tlemcen | Examineur    |
| Mr LACHACHI DJAMAL       | MCA | Université de Tlemcen | Encadreur    |
| Mme GHEFIR FATIHA DJAZIA | MAA | Université de Tlemcen | Co-Encadreur |

Année universitaire : 2025 /2026

## ***Remerciements***

*Nous tenons à remercier notre Dieu, le tout puissant, de nous avoir donné la santé et la volonté pour compléter ce modeste travail.*

*Tout d'abord à nos chers parents, d'avoir été là toutes ces années de nous avoir aidé à faire ce très long parcours, et de nous avoir guidé vers le droit chemin.*

*Nous remercions. Notre encadreur **Mr LACHACHI DJAMAL***

*Pour les efforts qu'il a déployés, pour nous aider, conseiller, encourager et corriger Nous tenons également à remercier notre Co-encadrante, **Mme GHEFIR FATIHA DJAZIA** pour son aide tout long de ce mémoire.*

*Nous tenons à remercier le président de jury **Mr NEMMICHE AHMED** et l'examineur **Mr BOURSALI AMIN** de nous avoir fait l'honneur d'examiner notre travail.*

*Nous remercions aussi tout le corps enseignant et administratif qui a contribué à notre formation universitaire. Sans oublier tous nos amis.*

## **DÉDICACE**

Nous dédions ce modeste travail :

À nos très chers parents,  
pour leur amour infini, leurs sacrifices, leur soutien moral et matériel, ainsi que pour tous les efforts qu'ils n'ont cessé de fournir afin de nous voir réussir dans notre parcours académique.

À nos familles,  
pour leur encouragement permanent, leur patience et leur confiance.

À tous nos enseignants,  
qui ont contribué à notre formation scientifique et technique durant toutes ces années d'études.

À nos amis et collègues,  
pour leur soutien, leurs conseils et les agréables moments partagés.

Enfin,  
Nous dédions ce travail à toute personne ayant contribué de près ou de loin à l'aboutissement de ce projet.

**Sid Ahmed et Mounir**

# RÉSUMÉ

Ce mémoire présente l'étude et la simulation d'une carte électronique dédiée au contrôle de vitesse d'un moteur à courant continu à l'aide d'une carte Arduino Uno et du logiciel Proteus ISIS.

L'étude porte sur la modélisation du moteur ainsi que sur l'analyse des principaux composants de commande, notamment l'Arduino Uno, le circuit L293D et la technique PWM. Un système de commande utilisant un potentiomètre a ensuite été conçu et simulé sous Proteus ISIS.

Les résultats obtenus montrent qu'il est possible de faire varier efficacement la vitesse du moteur dans les 2 sens de rotations tout en affichant les informations de fonctionnement et de l'état de charge de la batterie d'alimentation sur un écran LCD. Cette étude confirme l'efficacité de la solution proposée pour le contrôle de vitesse des moteurs à courant continu.

Mots-clés : Moteur à courant continu, Arduino Uno, L293D, PWM, Proteus ISIS, contrôle de vitesse.

# ABSTRACT

This thesis presents the study and simulation of an electronic board dedicated to speed control of a DC motor using an Arduino Uno board and Proteus ISIS software.

The study focuses on motor modeling and the analysis of the main control components, including the Arduino Uno, the L293D chip, and the PWM technique. A control system using a potentiometer was then designed and simulated using Proteus ISIS.

The results obtained show that it is possible to efficiently vary the motor speed in both directions of rotation while displaying operating information and the battery charge level on an LCD screen. This study confirms the effectiveness of the proposed solution for speed control of DC motors.

Keywords: DC motor, Arduino Uno, L293D, PWM, Proteus ISIS, speed control.

## المخلص

تُقدّم هذه الأطروحة دراسةً ومحاكاةً للوحة إلكترونية مُخصصة للتحكم في سرعة محرك التيار المستمر باستخدام لوحة أردوينو أونو وبرنامج بروتايوس آي إس آي إس.

تركز الدراسة على نمذجة المحرك وتحليل مكونات التحكم الرئيسية، بما في ذلك لوحة أردوينو أونو، وشرية L293D، وتقنية تعديل عرض النبضة (PWM). تم تصميم نظام تحكم باستخدام مقياس جهد ومحاكاته باستخدام برنامج بروتايوس ISIS

تُظهر النتائج المُتحصّل عليها إمكانية تغيير سرعة المحرك بكفاءة في كلا اتجاهي الدوران مع عرض معلومات التشغيل ومستوى شحن البطارية على شاشة LCD. تؤكد هذه الدراسة فعالية الحل المُقترح للتحكم في سرعة محركات التيار المستمر.

الكلمات المفتاحية: محرك تيار مستمر، أردوينو أونو، L293D، تعديل عرض النبضة، بروتايوس ISIS، التحكم في السرعة.

# Liste des figures

|                                                                                        |    |
|----------------------------------------------------------------------------------------|----|
| Figure 1.1 Moteur à courant continu .....                                              | 14 |
| Figure 1.2 Description d'un moteur à courant continu.....                              | 15 |
| Figure 1.3 Stator d'une machine à courant continu.....                                 | 16 |
| Figure 1.4 rotor d'un moteur à courant continu .....                                   | 17 |
| Figure 1.5 Collecteur tournant électrique .....                                        | 17 |
| Figure 1.6 Balais carbone de contact.....                                              | 17 |
| Figure 1.7 Loi de LAPLACE .....                                                        | 18 |
| Figure 1.8 Principe de création d'une force à partir d'un champ électromagnétique..... | 19 |
| Figure 1.9 Différents modes d'excitations d'un moteur à inducteur bobiné.....          | 20 |
| Figure 1.10 Moteur à aimant permanent .....                                            | 22 |
| Figure 1.11 Modélisation électrique du moteur.....                                     | 22 |
| Figure 1.12 Modélisation mécanique du moteur .....                                     | 23 |
| Figure 2.1 Arduino UNO.....                                                            | 30 |
| Figure 2.2 La carte Arduino MEGA 2560 .....                                            | 30 |
| Figure 2.3 La carte Arduino LEONARDO.....                                              | 31 |
| Figure 2.4 La carte Arduino DUE.....                                                   | 31 |
| Figure 2.5 La carte Arduino ZERO .....                                                 | 31 |
| Figure 2.6 La carte Arduino MICRO.....                                                 | 32 |
| Figure 2.7 La carte Arduino NANO .....                                                 | 32 |
| Figure 2.8 Description de la carte Arduino UNO.....                                    | 33 |
| Figure 2.9 Le microcontrôleur ATmega 328.....                                          | 34 |
| Figure 2.10. Broches d'entrées sorties numériques.....                                 | 36 |
| Figure 2.11 Broches d'entrées sorties analogique de l'Arduino.....                     | 37 |
| Figure 2.12 L'alimentation par le port USB. ....                                       | 37 |
| Figure 2.13 Alimentation par une pile 9V.....                                          | 38 |
| Figure 2.14 Principe d'une alimentation stabilisée. ....                               | 38 |
| Figure 2.15 L'interface de L'IDE Arduino.....                                          | 39 |
| Figure 2.16 Les boutons de l'IDE Arduino.....                                          | 40 |
| Figure 2.17 Les PIN de Le L293D .....                                                  | 41 |
| Figure 2.18 Le Schéma interne du L293D .....                                           | 42 |
| Figure 2.19 Le Fonctionnement du Pont en H : Rotation Horaire .....                    | 43 |
| Figure 2.20 Le Fonctionnement du Pont en H : Rotation anti-Horaire .....               | 44 |
| Figure 2.21 Schéma du L293D avec moteur au repos.....                                  | 45 |

|                                                                                     |    |
|-------------------------------------------------------------------------------------|----|
| Figure 2.22 commande du moteur en sens horaire .....                                | 45 |
| Figure 2.23 commande du moteur en sens anti-horaire.....                            | 46 |
| Figure 2.24 commande de l'arrêt du moteur .....                                     | 46 |
| Figure 2.25 schéma bloc du hacheur .....                                            | 47 |
| Figure 2.26 schéma de base du hacheur .....                                         | 47 |
| Figure 2.27 Chronogramme de commande .....                                          | 48 |
| Figure 2.28 La tension de sortie .....                                              | 49 |
| Figure 2.29 Hacheur série .....                                                     | 49 |
| Figure 2.30 Hacheur parallèle .....                                                 | 50 |
| Figure 2.31 Hacheur parallèle-série .....                                           | 50 |
| Figure 2.32 Forme d'onde d'un signal PWM.....                                       | 51 |
| Figure 2.33 Valeur moyenne d'un signal PWM.....                                     | 52 |
| Figure 2.34 LCD.....                                                                | 53 |
| Figure 2.35 brochage de LCD .....                                                   | 53 |
| Figure 3.1 Fenêtre de lancement ISIS de PROTEUS .....                               | 57 |
| Figure 3.2 Fenêtre de lancement ARES de PROTEUS.....                                | 58 |
| Figure 3.3 environnement de travail ISIS de Proteus .....                           | 59 |
| Figure 3.4 La barre d'outils principale .....                                       | 60 |
| Figure 3.5 Environnement de Travail ARES de PROTEUS .....                           | 61 |
| Figure 3.6 Schéma synoptique de la carte de contrôle de vitesse.....                | 62 |
| Figure 3.7 Schéma globale du système .....                                          | 64 |
| Figure 3.8 affichage de simulation en vitesse faible .....                          | 66 |
| Figure 3.9 signal PWM pour une vitesse faible .....                                 | 67 |
| Figure 3.10 affichage de simulation en vitesse élevée .....                         | 68 |
| Figure 3.11 signal PWM pour une vitesse élevée .....                                | 68 |
| Figure 3.12 sens anti-horaire du moteur CC .....                                    | 69 |
| Figure 3.13 affichage de la vitesse en Km/h .....                                   | 70 |
| Figure 3.14 affichage de l'état de charge de la batterie : LED rouge allumée .....  | 71 |
| Figure 3.15 affichage de l'état de charge de la batterie : LED orange allumée ..... | 72 |
| Figure 3.16 affichage de l'état de charge de la batterie : LED verte allumée .....  | 73 |
| Figure 3.17 étape 1 : choix du type de carte Arduino.....                           | 75 |
| Figure 3.18 étape 2 : écriture du programme d'Arduino.....                          | 75 |
| Figure 3.19 étape 3 : vérification et compilation du code d'Arduino.....            | 76 |
| Figure 3.20 étape 4 : importation de fichier HEX vers le montage ISIS.....          | 77 |
| Figure 3.21 étape 5 : simulation du montage .....                                   | 78 |

# Liste des tableaux

Tableau 2-1 Les caractéristiques de la carte Arduino UNO

Tableau 2-2 Description d'interface d'IDE Arduino

Tableau 2-3 Description des boutons de l'IDE Arduino

Tableau 2-4 Table de Commande de L293D

Tableau 2.5 les Broches de l’afficheur LCD

# Table de Matière

|                                                                          |           |
|--------------------------------------------------------------------------|-----------|
| <b>Remerciements.....</b>                                                | <b>2</b>  |
| <b>DÉDICACE.....</b>                                                     | <b>2</b>  |
| <b>RÉSUMÉ.....</b>                                                       | <b>3</b>  |
| <b>ABSTRACT .....</b>                                                    | <b>3</b>  |
| <b>المخلص .....</b>                                                      | <b>3</b>  |
| <b>INTRODUCTION GÉNÉRALE.....</b>                                        | <b>11</b> |
| <b>CHAPITRE 1 : Le MOTEUR à COURANT CONTINU .....</b>                    | <b>13</b> |
| Introduction .....                                                       | 14        |
| 1.1 Le moteur à courant continu.....                                     | 14        |
| 1.2 Constitution du moteur à courant continu .....                       | 15        |
| 1.2.1 L'inducteur (Le stator) : .....                                    | 155       |
| 1.2.2 Le rotor ou induit : .....                                         | 166       |
| 1.2.3 Le collecteur : .....                                              | 177       |
| 1.2.4 Les balais : .....                                                 | 177       |
| 1.3 Principe de fonctionnement du moteur à courant continu .....         | 188       |
| 1.4 Les différents types de moteurs à courant continu .....              | 19        |
| 1.4.1 Moteur à inducteur bobiné .....                                    | 200       |
| 1.4.1.1 Moteur à excitation séparée .....                                | 200       |
| 1.4.1.2 Moteur à excitation shunt (ou dérivation) .....                  | 200       |
| 1.4.1.3 Moteur à excitation série.....                                   | 211       |
| 1.4.1.4 Moteur à excitation composée (compound).....                     | 211       |
| 1.4.2 Moteur à aimants permanents .....                                  | 211       |
| 1.5 Modélisation électrique du moteur.....                               | 222       |
| 1.6 Modélisation mécanique du moteur .....                               | 233       |
| 1.7 Relation entre tension d'alimentation et vitesse de rotation .....   | 244       |
| 1.8 Intérêt de la commande progressive de tension.....                   | 255       |
| 1.9 Avantages et domaines d'utilisation du moteur à courant continu..... | 255       |
| 1.10 Inconvénients et limites .....                                      | 266       |
| 1.11 Fonction de transfert simplifiée du moteur .....                    | 266       |
| 1.12 Choix du moteur à courant continu : .....                           | 266       |
| 1.12.1 Puissance requise : .....                                         | 266       |
| 1.12.2 Tension d'alimentation : .....                                    | 277       |
| 1.12.3 Vitesse : .....                                                   | 277       |

|                                                                    |                                                            |           |
|--------------------------------------------------------------------|------------------------------------------------------------|-----------|
| 1.12.4                                                             | Couple :.....                                              | 277       |
| 1.12.5                                                             | Durée de vie :.....                                        | 277       |
| 1.12.6                                                             | Environnement :.....                                       | 277       |
| 1.12.7                                                             | Prix :.....                                                | 277       |
| Conclusion du chapitre 1 .....                                     |                                                            | 27        |
| <b>Chapitre 02 : COMPOSANTS ET ELECTRONIQUE DE PUISSANCE .....</b> |                                                            | <b>28</b> |
| Introduction .....                                                 |                                                            | 29        |
| 2.1                                                                | Composants utilisés.....                                   | 29        |
| 2.1.1.                                                             | Arduino .....                                              | 29        |
| 2.1.1.1.                                                           | Arduino UNO.....                                           | 322       |
| 2.1.1.2.                                                           | Le microcontrôleur ATmega328P .....                        | 333       |
| 2.1.1.3.                                                           | Caractéristiques de l'Arduino UNO .....                    | 355       |
| 2.1.1.4.                                                           | Configuration du brochage de l'Arduino UNO .....           | 366       |
| 2.1.1.5.                                                           | Alimentation de la carte Arduino UNO .....                 | 37        |
| 2.1.1.6.                                                           | Application.....                                           | 38        |
| 2.1.1.7.                                                           | IDE Arduino.....                                           | 39        |
| 2.1.2.                                                             | Le circuit L293D.....                                      | 400       |
| 2.1.2.1.                                                           | Définition .....                                           | 400       |
| 2.1.2.2.                                                           | Caractéristiques Principales .....                         | 400       |
| 2.1.2.3.                                                           | Description des broches du L293D .....                     | 411       |
| 2.1.2.4.                                                           | Schéma interne et principe de fonctionnement du L293D..... | 422       |
| 2.1.2.4.1.                                                         | Fonctionnement du Pont en H : Rotation Horaire .....       | 433       |
| 2.1.2.4.2.                                                         | Fonctionnement du Pont en H : Rotation Anti-Horaire.....   | 433       |
| 2.1.2.5.                                                           | Application des codes sur le L293D .....                   | 455       |
| 2.1.3.                                                             | Le Hacheur.....                                            | 477       |
| 2.1.3.1.                                                           | Définition .....                                           | 477       |
| 2.1.3.2.                                                           | Principe de base .....                                     | 477       |
| 2.1.3.3.                                                           | Schéma de base .....                                       | 477       |
| 2.1.3.4.                                                           | Types de hacheurs.....                                     | 49        |
| 2.1.3.4.1.                                                         | Hacheur série .....                                        | 49        |
| 2.1.3.4.2.                                                         | Hacheur parallèle .....                                    | 49        |
| 2.1.3.4.3.                                                         | Hacheur parallèle-série .....                              | 500       |
| 2.1.3.5.                                                           | Technique de commande du hacheur.....                      | 500       |
| 2.1.3.6.                                                           | Principe de MLI .....                                      | 511       |
| 2.1.4.                                                             | L'afficheur LCD .....                                      | 522       |

|          |                                                                                          |           |
|----------|------------------------------------------------------------------------------------------|-----------|
| 2.1.4.1. | Présentation générale .....                                                              | 522       |
| 2.1.4.2. | Brochage de LCD .....                                                                    | 533       |
| 2.1.4.3. | Mode de fonctionnent .....                                                               | 544       |
| 2.1.4.4. | Présentation des pins de LCD .....                                                       | 544       |
|          | Conclusion du chapitre 2 .....                                                           | 555       |
|          | <b>Chapitre 03 : SIMULATION DE LA CARTE DE CONTROLE DE VITESSE .....</b>                 | <b>56</b> |
|          | Introduction .....                                                                       | 57        |
| 3.1      | Présentation du Logiciel Proteus .....                                                   | 57        |
| 3.1.1    | Présentation de l'environnement ISIS de PROTEUS .....                                    | 58        |
| 3.1.2    | Présentation de l'interface d'ISIS de PROTEUS .....                                      | 58        |
| 3.1.3    | La barre d'outils principale .....                                                       | 59        |
| 3.1.4    | Présentation de l'interface ARES de PROTEUS .....                                        | 61        |
| 3.2      | Schéma synoptique du circuit.....                                                        | 622       |
| 3.2.1    | Description du montage .....                                                             | 622       |
| 3.2.1.1  | Le Bloc Alimentation & Gestion de l'Énergie.....                                         | 622       |
| 3.2.1.2  | Le Bloc Entrées Utilisateur .....                                                        | 623       |
| 3.2.1.3  | Le Bloc de Contrôle .....                                                                | 633       |
| 3.2.1.4  | Le Bloc Puissance & Commande Moteur .....                                                | 633       |
| 3.2.1.5  | Le Bloc Sorties d'Information (Affichage) .....                                          | 633       |
| 3.3      | Schéma électrique du montage .....                                                       | 644       |
| 3.3.1    | Description de montage .....                                                             | 644       |
| 3.3.1.1  | Étage de Commande et de Traitement .....                                                 | 644       |
| 3.3.1.2  | Étage de Puissance et de Commande Moteur .....                                           | 655       |
| 3.3.1.3  | Étage d'Affichage LCD .....                                                              | 655       |
| 3.3.1.4  | Indicateur d'État.....                                                                   | 655       |
| 3.4      | Différents cas de simulations.....                                                       | 666       |
| 3.4.1    | Cas de simulation faible vitesse .....                                                   | 666       |
| 3.4.2    | Cas de simulation vitesse élevée.....                                                    | 68        |
| 3.4.3    | Simulation inversion moteur .....                                                        | 69        |
| 3.4.4    | Affichage de la vitesse en Km/h .....                                                    | 700       |
| 3.4.5    | Simulation affichage de l'état de charge de la batterie faible : LED rouge allumée.....  | 711       |
| 3.4.6    | Simulation affichage de l'état de charge de la batterie moyen : LED orange allumée ..... | 722       |
| 3.4.7    | Simulation affichage de l'état de charge de la batterie pleine : LED verte allumée ..... | 733       |
| 3.5      | Développement et intégration du programme Arduino sous ISIS de Proteus .....             | 744       |
| 3.5.1    | Exemple de programme IDE.....                                                            | 744       |

|         |                                                                          |            |
|---------|--------------------------------------------------------------------------|------------|
| 3.5.1.1 | Étape 1 : Le choix du type de carte Arduino se fait en fonction de ..... | 744        |
| 3.5.1.2 | Etape 2 : écriture du programme IDE .....                                | 755        |
| 3.5.1.3 | Etape 3 : Vérification et correction du programme.....                   | 766        |
| 3.5.1.4 | Etape 4 : Importation du Ficher HEX .....                                | 777        |
| 3.5.1.5 | Etape 5 : Validation et simulation .....                                 | 78         |
| 3.5     | Validation des résultats .....                                           | 749        |
|         | Conclusion du chapitre 3 .....                                           | 79         |
|         | <b>CONCLUSION GÉNÉRALE.....</b>                                          | <b>800</b> |
|         | <b>Bibliographie.....</b>                                                | <b>811</b> |

# INTRODUCTION GÉNÉRALE

Au cours des dernières décennies, l'automatisation industrielle a connu une évolution remarquable grâce au développement des systèmes électroniques de commande et de régulation. Parmi les actionneurs électriques les plus utilisés dans les domaines industriel, domestique et pédagogique, le moteur à courant continu occupe une place importante. Sa structure simple, son faible coût, sa robustesse ainsi que la facilité de contrôle de sa vitesse de rotation en font un choix privilégié dans de nombreuses applications. Contrairement à d'autres types de machines électriques, le moteur à courant continu présente une relation presque directe entre la tension appliquée à ses bornes et la vitesse obtenue, ce qui le rend particulièrement adapté aux systèmes nécessitant une variation progressive ou précise de la vitesse.]

Dans plusieurs secteurs tels que les convoyeurs industriels, les robots mobiles, les systèmes de ventilation, les machines-outils, les ascenseurs miniatures ainsi que les vélos électriques, le contrôle de la vitesse du moteur constitue un élément essentiel afin d'assurer un fonctionnement optimal, sécurisé et économique. Cette nécessité a conduit les ingénieurs à développer diverses techniques de commande électronique permettant d'ajuster la tension moyenne appliquée au moteur tout en limitant les pertes énergétiques. [3], [4], [5], [10], [11], [14]

Avec l'apparition des microcontrôleurs programmables et des logiciels de simulation, il est désormais possible de concevoir des cartes électroniques intelligentes capables d'assurer un contrôle flexible, précis et efficace des moteurs électriques. Parmi les plateformes les plus utilisées, la carte Arduino Uno s'est imposée comme une solution à la fois pédagogique et technique grâce à sa simplicité de programmation, sa compatibilité avec plusieurs capteurs et modules électroniques ainsi que sa capacité à générer des signaux de modulation de largeur d'impulsion appelés signaux PWM (Pulse Width Modulation). Associée à un circuit de puissance tel que le L293D, cette carte permet d'alimenter et de contrôler un moteur à courant continu de manière simple et fiable. [7], [8], [15], [18], [21], [23], [28], [29], [30]

La simulation électronique représente une étape essentielle permettant de valider les choix techniques et d'optimiser le fonctionnement global du système. Le logiciel Proteus ISIS offre dans ce cadre un environnement performant permettant de dessiner des schémas électroniques, d'intégrer des composants programmables, de charger des programmes Arduino et d'observer le comportement dynamique du système en temps réel. Cet outil permet ainsi de réduire les erreurs de conception et de vérifier le bon fonctionnement du montage avant une éventuelle implémentation matérielle. [8], [9], [19]

Notre Projet de Fin d'Etudes s'inscrit dans cette perspective, dont l'objectif principal est d'étudier, de proposer et de simuler une carte électronique de commande permettant de contrôler la vitesse d'un moteur à courant continu dans les 2 sens de rotation.

Notre travail sera organisé sous forme de 3 chapitres et une conclusion générale :

Dans le chapitre1 : nous présenterons des généralités sur le moteur à courant continu, sa construction, sa modélisation, et son principe de fonctionnement, tout en introduisant les avantages, inconvénients et les différents critères de choix.

Le chapitre 2 : permet de mettre en évidence les différents éléments essentiels constituant la carte électronique, en mettant particulièrement l'accent sur la technique PWM et sur le principe de fonctionnement du circuit électronique monté en pont en H permettant de contrôler le sens de rotation du moteur.

Le chapitre3 présente les outils essentiels de simulations et de programmation en donnant d'abord une présentation du logiciel PROTEUS avec ses 2 environnements de travail ISIS et ARES, ainsi qu'une introduction sur l'IDE d'Arduino avec un exemple d'implémentation. Enfin nous présenterons le schéma proposé de la carte électronique de contrôle de vitesse sous l'environnement ISIS de PROTEUS avec les différents cas de simulation et différentes formes de fonctionnement du moteur et d'affichage de vitesse et d'état de charge de la batterie d'alimentation.

Enfin une conclusion générale mettra fin à ce projet.

# **CHAPITRE 1 : Le MOTEUR à COURANT CONTINU**

# Introduction

L'objectif de ce chapitre est de présenter une étude détaillée du moteur à courant continu, en commençant par sa structure interne, son principe de fonctionnement, puis sa modélisation électrique et mécanique. Une attention particulière sera accordée aux équations mathématiques reliant la tension d'alimentation, le courant d'induit, le couple électromagnétique et la vitesse de rotation. Cette base théorique permettra de justifier le choix d'une commande par modulation PWM. [1], [2], [25]

## 1.1 Le moteur à courant continu

Le moteur à courant continu représente l'un des actionneurs électriques les plus utilisés dans les systèmes industriels, pédagogiques et automatisés lorsqu'une variation souple et progressive de la vitesse est recherchée. Sa large utilisation est due principalement à la simplicité de son alimentation, à la facilité de sa commande électronique ainsi qu'à sa rapidité de réponse dynamique. Contrairement à plusieurs autres machines électriques nécessitant des convertisseurs complexes, le moteur à courant continu peut être piloté efficacement par une simple variation de la tension moyenne appliquée à son induit.

Le moteur à courant continu est une machine électromécanique rotative destinée à transformer une énergie électrique continue en énergie mécanique de rotation. Son fonctionnement repose sur l'interaction entre un champ magnétique fixe et des conducteurs parcourus par un courant électrique. [1], [2], [25], [27]



**Figure 1.1 Moteur à courant continu**

## 1.2 Constitution du moteur à courant continu

D'une manière générale, un moteur à courant continu contient les composants suivants, (Figure 1.2) et les éléments principaux sont :

- L'inducteur
- L'induit
- Le collecteur et les balais [2], [25], [27]

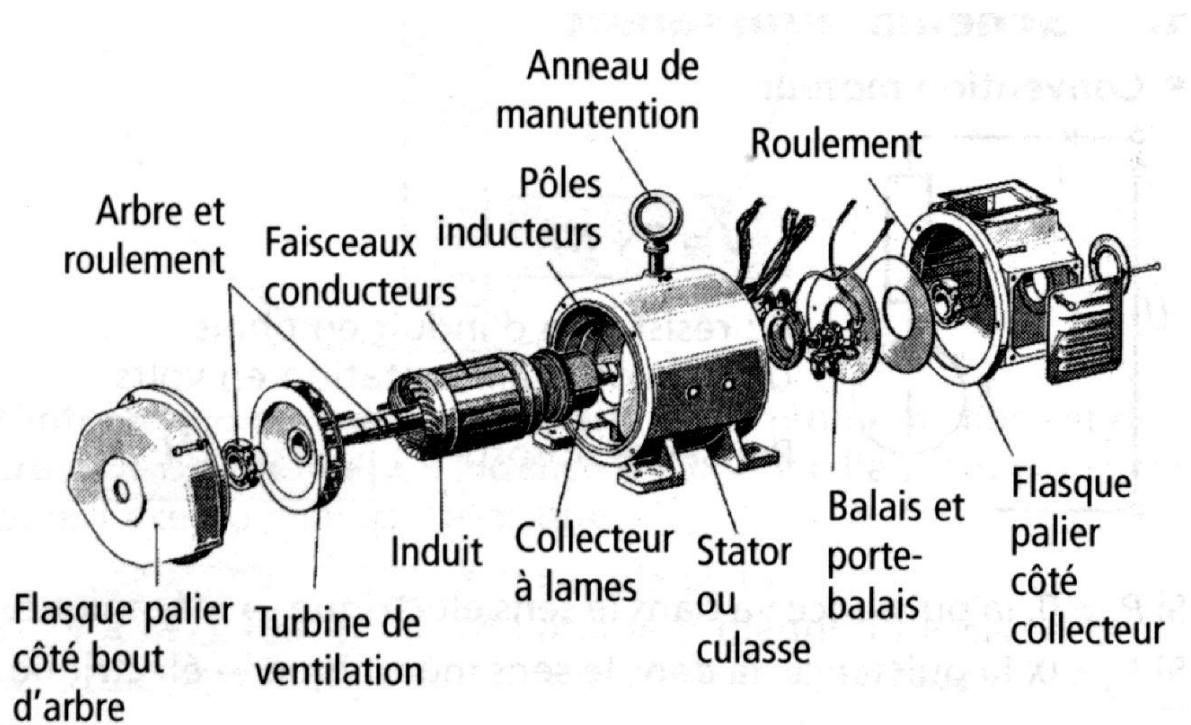


Figure 1.2 Description d'un moteur à courant continu

### 1.2.1 L'inducteur (Le stator) :

L'inducteur est la partie fixe de la machine. Son rôle est de créer le champ magnétique. Il peut être réalisé soit par aimants permanents, soit par bobinages alimentés en courant continu. Le circuit magnétique du stator crée un champ magnétique appelé champ inducteur ( $B_s$ ). L'inducteur magnétise le moteur en créant un flux magnétique ( $\Phi$ ) dans l'entrefer. L'entrefer est l'espace entre les pôles du stator et le rotor.

Il comprend :

- La carcasse
- Les pôles magnétiques
- Les enroulements d'excitation [2], [25], [27]

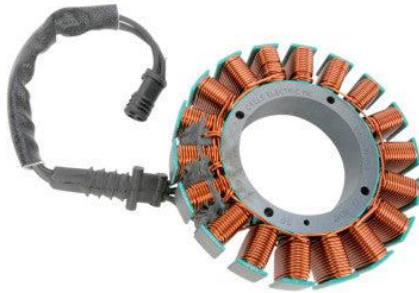


**Figure 1.3 Stator d'une machine à courant continu**

### **1.2.2 Le rotor ou induit :**

L'induit est la partie tournante et mobile de la machine électrique. Il porte les conducteurs soumis au flux magnétique de l'inducteur. Le fer de l'induit comporte des encoches à sa périphérie, dans lesquelles est logé le bobinage de l'induit parcouru par un courant continu.

Le bobinage de l'induit est formé de sections ; chaque section est constituée d'un groupe de spires isolées entre elles. Les extrémités du bobinage sont reliées au collecteur.

L'induit est constitué de :

- Un noyau ferromagnétique formé d'un empilement de tôles en fer au silicium isolées entre elles afin de limiter les pertes par hystérésis et les pertes par courants de Foucault.
- Des encoches contenant les conducteurs.
- Un bobinage parcouru par le courant d'induit

C'est dans l'induit que se développe le couple électromagnétique [2], [25], [27]



**Figure 1.4 rotor d'un moteur à courant continu**

### **1.2.3 Le collecteur :**

Est un ensemble de lames conductrices assurant la commutation du courant dans les bobines du rotor. [2], [25], [27]

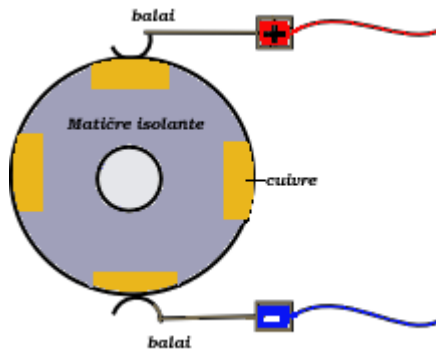


**Figure 1.5 Collecteur tournant électrique**

### **1.2.4 Les balais :**

Éléments assurant le contact électrique entre la partie fixe et la partie tournante. [2], [25], [27]

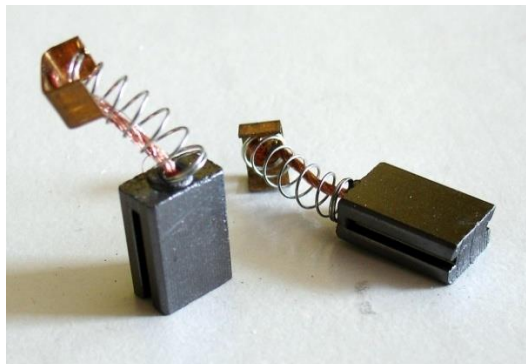


**Figure 1.6 Balais carbone de contact**

Rôle du collecteur et balais :

- Alimenter l'induit
- Inverser le courant dans les conducteurs
- Assurer la continuité électrique

### 1.3 Principe de fonctionnement du moteur à courant continu

Le principe fondamental de fonctionnement du moteur à courant continu est basé sur la loi de Laplace :

$$F = B \cdot I \cdot L \cdot \sin(\theta) \quad (1.1)$$

Avec :

- F : force électromagnétique (N)
- B : champ magnétique (T)
- I : courant (A)
- L : longueur du conducteur (m)
- $\theta$  : angle entre B et I

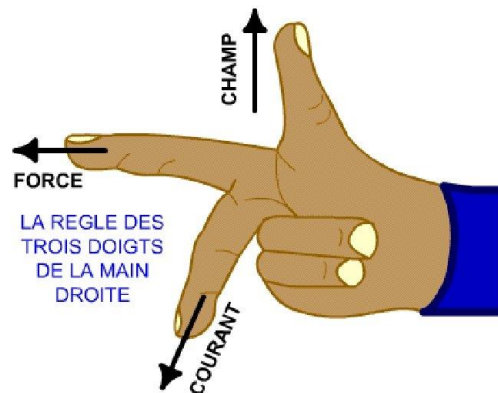


Figure 1.7 Loi de LAPLACE

Lorsqu'un conducteur parcouru par un courant électrique est placé dans un champ magnétique, il subit une force mécanique dont la direction dépend du sens du courant et de l'orientation du champ.

Dans le moteur :

- Le stator crée un flux magnétique  $\Phi$  ;
- L'induit reçoit un courant électrique  $I_a$  ;

- L'interaction entre  $\Phi$  et  $I_a$  produit un couple électromagnétique.

Ce couple provoque la rotation du rotor autour de son axe.

Au fur et à mesure que le rotor tourne, les conducteurs coupent les lignes du champ magnétique et engendrent une tension induite opposée à la tension d'alimentation. Cette tension interne est appelée force contre-électromotrice.

Ainsi, deux phénomènes se développent simultanément :

1. Un phénomène électrique lié à l'alimentation de l'induit,
2. Un phénomène mécanique lié au mouvement de rotation.

Le moteur devient alors un véritable convertisseur électromécanique. [1], [2], [6], [25], [31]

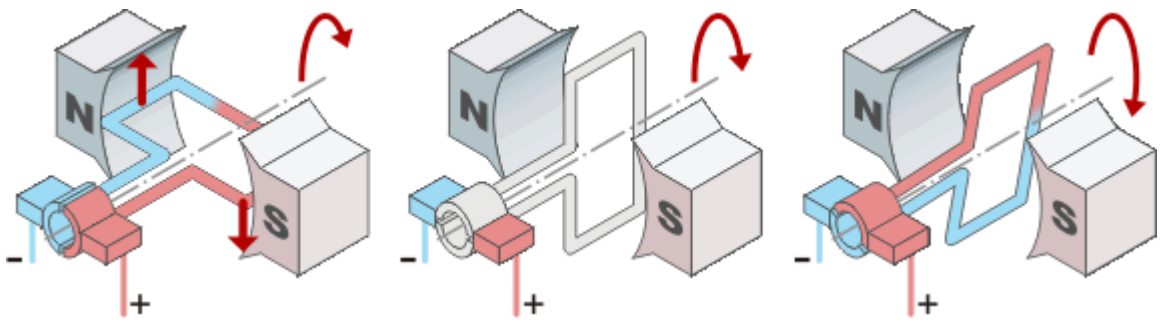


**Figure 1.8 Principe de création d'une force à partir d'un champ électromagnétique**

## 1.4 Les différents types de moteurs à courant continu

Il existe deux grands major type : Moteur à inducteur bobiné et Moteur à aimants permanents :

## 1.4.1 Moteur à inducteur bobiné

Dans ce type de moteur, le champ magnétique est produit par un enroulement inducteur alimenté en courant continu. Selon la connexion de l'inducteur, on distingue quatre types principaux :

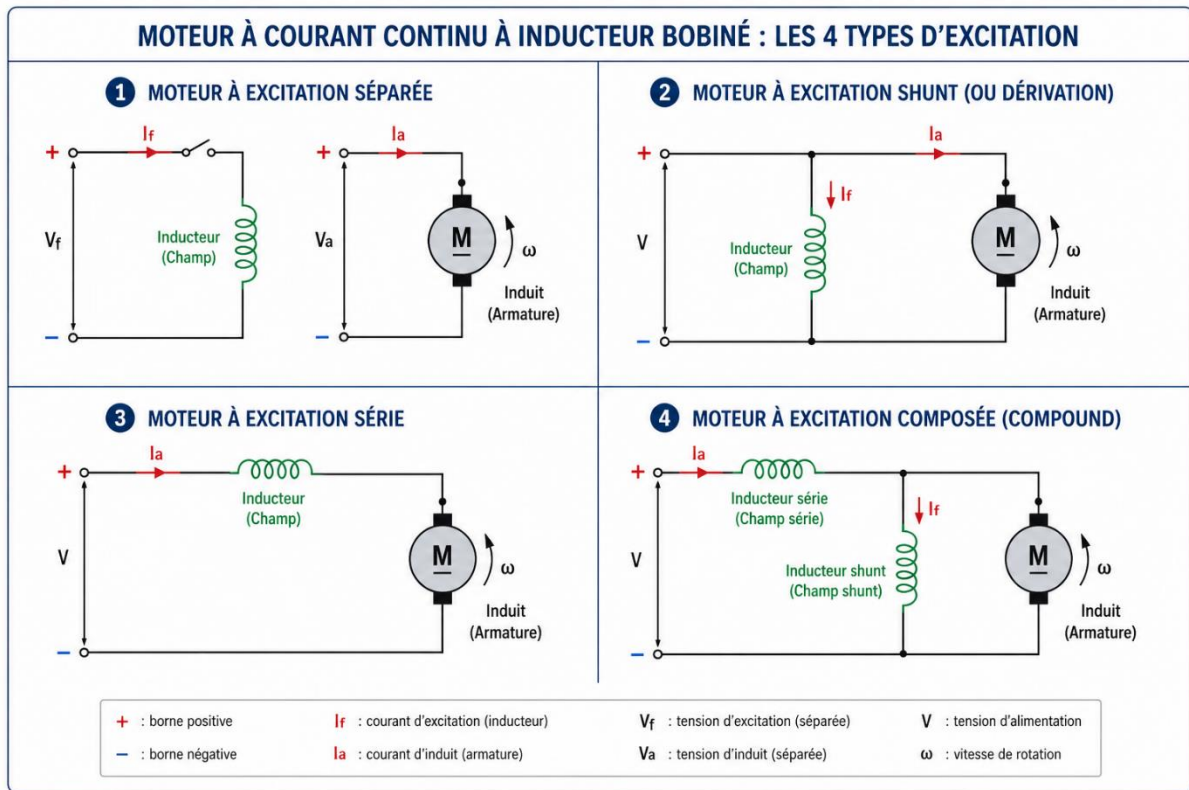


Figure 1.9 Différents modes d'excitations d'un moteur à inducteur bobiné

### 1.4.1.1 Moteur à excitation séparée

L'inducteur est alimenté par une source indépendante de celle de l'induit. Cette configuration permet un contrôle précis de la vitesse et des performances, au prix d'une alimentation supplémentaire.

Lorsque le flux est constant, la force contre-électromotrice dépend de la vitesse de rotation. Le moteur offre une grande souplesse de commande, une large plage de vitesse et est très utilisé dans les systèmes d'asservissement, les machines-outils et les entraînements industriels avec variateur. [2], [25]

### 1.4.1.2 Moteur à excitation shunt (ou dérivation)

L'inducteur est connecté en parallèle avec l'induit et alimenté par la même source.

Ce moteur présente une vitesse relativement constante malgré les variations de charge. Il offre un bon compromis entre performance et stabilité et est utilisé dans les machines-outils, les ascenseurs et certains équipements de levage. [2], [25]

### **1.4.1.3 Moteur à excitation série**

L'inducteur est connecté en série avec l'induit, donc le même courant traverse les deux enroulements. Ce moteur se caractérise par un très fort couple de démarrage, ce qui le rend adapté aux applications nécessitant une forte traction comme les grues, les ponts roulants et les systèmes de levage. Cependant, sa vitesse varie fortement avec la charge et il ne doit jamais fonctionner à vide afin d'éviter un emballement dangereux. [2], [25]

### **1.4.1.4 Moteur à excitation composée (compound)**

Ce moteur combine deux enroulements : un en série avec l'induit et un en parallèle. Il associe les avantages du moteur série (fort couple de démarrage) et du moteur shunt (bonne stabilité de vitesse). Il est utilisé lorsque l'on cherche un compromis entre puissance et régulation, notamment dans les ascenseurs, les machines-outils et les systèmes industriels. [2], [25]

## **1.4.2 Moteur à aimants permanents**

Le champ magnétique est fourni par des aimants permanents. Ce type de moteur est simple, compact, fiable et offre un bon rendement. Il est très utilisé dans les applications de faible puissance ainsi que dans les simulations pédagogiques.

Dans notre projet, le moteur considéré est assimilé à un moteur à aimants permanents, ce qui simplifie la modélisation théorique. [2], [22], [25]

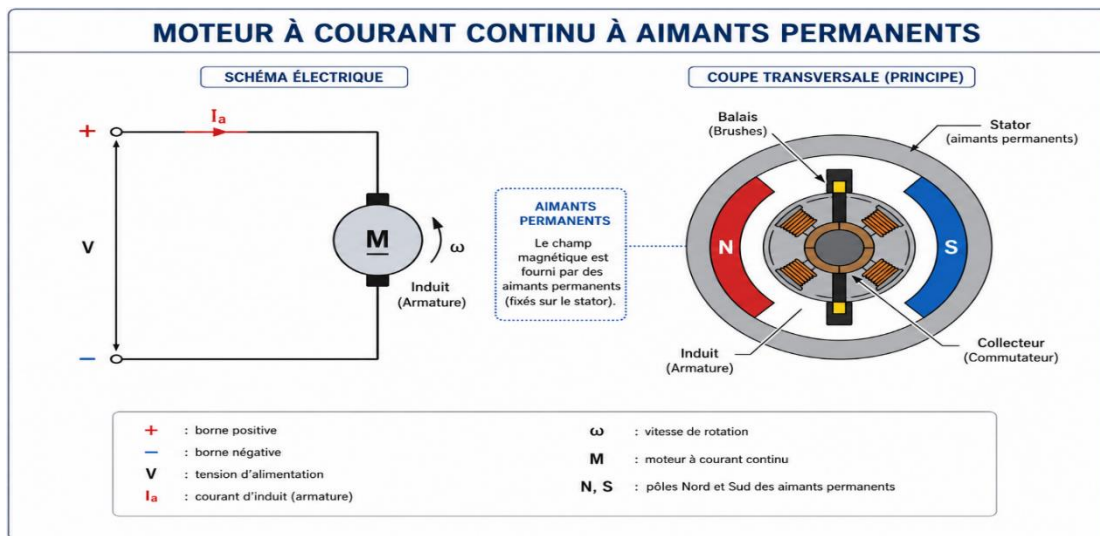


**Figure 1.10 Moteur à aimant permanent**

## 1.5 Modélisation électrique du moteur

L'induit du moteur peut être représenté électriquement par :

- Une résistance interne  $R_a$ ,
- Une inductance  $L_a$ ,
- Une force contre-électromotrice  $E$ . [1], [2], [6], [25]

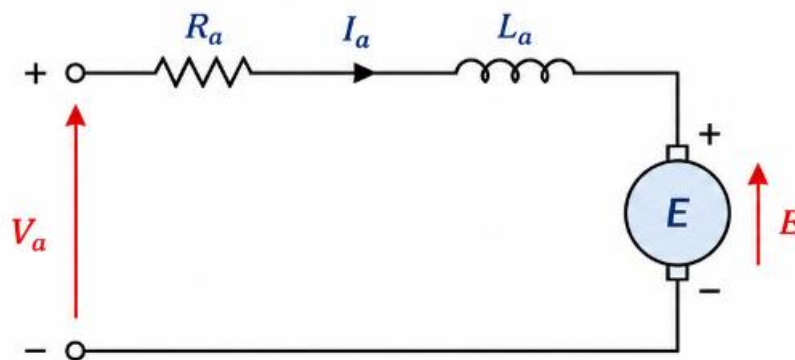


**Figure 1.11 Modélisation électrique du moteur**

En appliquant la loi des mailles au circuit d'induit, on obtient :

$$V_a = R_a \cdot I_a + L_a \cdot \frac{dI_a}{dt} + E \quad (1.2)$$

- $V_a$  : tension appliquée à l'induit,
- $R_a$  : résistance de l'induit,
- $L_a$  : inductance de l'induit,
- $I_a$  : courant d'induit,
- $E$  : force contre-électromotrice.

La force contre-électromotrice dépend directement de la vitesse du rotor : [1], [2], [25], [31]

$$E = K_e \cdot \Omega \quad (1.3)$$

- $K_e$  : constante électrique,
- $\Omega$  : vitesse angulaire.

Cette relation montre que lorsque la vitesse augmente, la tension interne  $E$  augmente également, ce qui tend à limiter automatiquement le courant absorbé.

## 1.6 Modélisation mécanique du moteur

Du point de vue mécanique, le moteur développe un couple électromagnétique proportionnel au courant absorbé : [1], [2], [6], [25], [31]

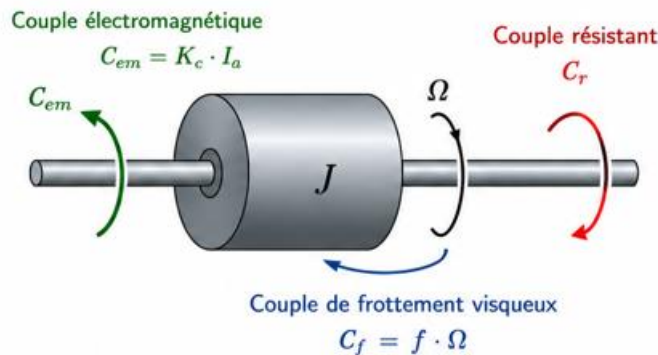


**Figure 1.12 Modélisation mécanique du moteur**

$$C_{em} = K_c \cdot I_a \quad (1.4)$$

Avec :

- $C_{em}$  : couple électromagnétique,
- $K_c$  : constante de couple.

Ce couple doit vaincre :

- L'inertie mécanique du rotor,
- Les frottements mécaniques,
- Le couple résistant de la charge.

En appliquant le principe fondamental de la dynamique de rotation :

$$J \cdot \frac{d\Omega}{dt} + f \cdot \Omega = C_{em} - C_r \quad (1.5)$$

En remplaçant  $C_{em}$  :

$$J \cdot \frac{d\Omega}{dt} + f \cdot \Omega = (K_c \cdot I_a) - C_r \quad (1.6)$$

Cette équation exprime l'équilibre mécanique du système.

- $J$  : moment d'inertie total du système en rotation ( $\text{kg} \cdot \text{m}^2$ ).
- $\Omega$  : vitesse angulaire du moteur ( $\text{rad/s}$ ).
- $\frac{d\Omega}{dt}$  : accélération angulaire du rotor ( $\text{rad/s}^2$ ).
- $f$  : coefficient de frottement visqueux ( $\text{N} \cdot \text{m} \cdot \text{s/rad}$ ).
- $K_c$  : constante de couple du moteur ( $\text{N} \cdot \text{m/A}$ ).
- $I_a$  : courant d'induit (A).
- $C_{em}$  : couple électromagnétique développé par le moteur ( $\text{N} \cdot \text{m}$ ).
- $C_r$  : couple résistant appliqué à l'arbre moteur ( $\text{N} \cdot \text{m}$ ).

## 1.7 Relation entre tension d'alimentation et vitesse de rotation

En régime permanent, lorsque les termes transitoires deviennent négligeables, l'équation électrique peut être simplifiée en :

$$V_a = R_a \cdot I_a + K_e \cdot \Omega \quad (1.7)$$

D'où :

$$\Omega = (V_a - R_a \cdot I_a) / K_e \quad (1.8)$$

Cette expression montre clairement que la vitesse du moteur dépend essentiellement de la tension moyenne appliquée à ses bornes. Ainsi :

- Si  $V_a$  augmente  $\rightarrow \Omega$  augmente,
- Si  $V_a$  diminue  $\rightarrow \Omega$  diminue.

Cette propriété constitue le fondement même du contrôle électronique de la vitesse développée. [1], [2], [6], [25]

## 1.8 Intérêt de la commande progressive de tension

Une alimentation directe du moteur sous tension fixe ne permet qu'une seule vitesse de rotation quasi constante. Pour obtenir plusieurs régimes de fonctionnement, il est nécessaire de faire varier la tension moyenne appliquée.

Les anciennes méthodes analogiques utilisant des résistances variables présentent plusieurs inconvénients :

- Pertes énergétiques importantes,
- Échauffement,
- Faible précision.

Pour remédier à cela, la commande moderne utilise des interrupteurs électroniques rapides permettant de hacher la tension d'alimentation. Cette solution offre :

- Un meilleur rendement,
- Une grande souplesse,
- Une commande numérique simple. [3], [4], [5], [6], [32]

## 1.9 Avantages et domaines d'utilisation du moteur à courant continu

Le moteur à courant continu présente plusieurs avantages :

- Facilité de commande,
- Bon couple de démarrage,
- Réponse rapide,
- Inversion facile du sens,
- Adaptation idéale avec les microcontrôleurs.

Il est utilisé dans :

- Robots mobiles,

- Systèmes de convoyage,
- Ventilateurs,
- Petites machines-outils,
- Systèmes automatisés pédagogiques. [1], [2], [6], [25]

## 1.10 Inconvénients et limites

Malgré ses qualités, ce moteur possède certaines limites :

- Usure des balais,
- Présence d'étincelles,
- Entretien périodique,
- Sensibilité aux surcharges.

Ces contraintes rendent nécessaire une commande électronique bien maîtrisée. [1], [2], [6], [25]

## 1.11 Fonction de transfert simplifiée du moteur

En combinant les équations électrique et mécanique, il est possible d'obtenir une relation dynamique entrée/sortie : [1], [2], [6], [10], [12], [14]

$$\frac{\Omega(s)}{V_a(s)} = \frac{K_c}{[(L_a \cdot J)s^2 + (R_a \cdot J + L_a \cdot f)s + (R_a \cdot f + K_e \cdot K_c)]} \quad (1.9)$$

Cette expression montre que le moteur se comporte comme un système du second ordre présentant une réponse progressive.

La vitesse n'évolue donc jamais instantanément mais suit une phase transitoire avant stabilisation.

## 1.12 Choix du moteur à courant continu :

Le choix d'un moteur à courant continu dépend de plusieurs critères, parmi lesquels : [1], [2], [25]

### 1.12.1 Puissance requise :

La puissance nécessaire pour entraîner la charge doit être prise en compte afin de choisir la taille appropriée du moteur à courant continu.

### **1.12.2 Tension d'alimentation :**

La tension disponible doit être considérée pour sélectionner un moteur à courant continu ayant la tension nominale adéquate.

### **1.12.3 Vitesse :**

La vitesse de rotation exigée par la charge doit être prise en compte afin de choisir un moteur possédant la vitesse nominale convenable.

### **1.12.4 Couple :**

Le couple nécessaire à la charge doit être considéré pour choisir un moteur à courant continu avec un couple nominal approprié.

### **1.12.5 Durée de vie :**

La durée de vie prévue du moteur doit être prise en compte afin de sélectionner un moteur ayant une longévité adaptée.

### **1.12.6 Environnement :**

L'environnement d'utilisation du moteur doit être pris en considération pour choisir un moteur présentant les caractéristiques adéquates, telles que la protection contre l'humidité et la poussière.

### **1.12.7 Prix :**

Le coût du moteur à courant continu doit également être pris en compte afin de choisir un moteur compatible avec le budget disponible.

## **Conclusion du chapitre 1**

Dans ce premier chapitre, nous avons abordé une étude théorique détaillée du moteur à courant continu utilisé comme charge principale dans ce projet. L'analyse de sa constitution, de son principe de fonctionnement ainsi que de ses modèles électrique et mécanique a permis de mettre en évidence que la vitesse de rotation dépend essentiellement de la tension moyenne appliquée à l'induit. Cette caractéristique rend possible une commande simple et efficace par modulation de largeur d'impulsion. [1], [2], [25]

## **CHAPITRE 02 : COMPOSANTS ET ELECTRONIQUE DE PUISSANCE**

# Introduction

L'objectif de notre projet est de réaliser un système qui permet de commander la vitesse d'un moteur à courant continu en utilisant une carte Arduino. Pour ce faire, il est indispensable d'utiliser les composants matériels et les structures de l'électronique de puissance nécessaires à la réalisation de cette commande. Dans ce chapitre, nous présentons donc les différents composants, tels que le hacheur et les dispositifs de commutation, qui ont contribué dans l'étude et la simulation de cette carte électronique. [3], [4], [8]

## 2.1 Composants utilisés

### 2.1.1.Arduino

- **Présentation de la Carte Arduino :**

L'Arduino est une plateforme de prototypage électronique à code source ouvert (open-source) basée sur une carte électronique équipée d'un microcontrôleur et d'un environnement de développement logiciel IDE (Integrated Development Environment). Elle est conçue pour faciliter la création d'objets interactifs et de systèmes automatisés. [8]

- **Architecture Matériel :**

La carte Arduino Uno, qui est notre modèle de référence, s'articule autour d'un microcontrôleur de type **Atmel AVR** (généralement l'ATmega328P). Elle dispose de ressources spécifiques pour l'interfaçage.

- **Les entrées/sorties numériques :** Permettent de commander des composants en tout ou rien (0 ou 5V).
- **Les entrées analogiques :** Utilisées pour convertir des tensions variables (venant de capteurs) en données numériques.
- **Les ports PWM**

Sur une carte Arduino, un port PWM est une broche capable de simuler un signal analogique à partir d'un signal numérique **pour faire varier la puissance délivrée à un composant** (comme la vitesse d'un moteur ou l'éclat d'une LED) [8]

- **Environnement Logiciel**

Le langage Arduino est une implémentation des langages C/C++. Le logiciel IDE (Integrated Development Environment) permet d'écrire le code, de le compiler et de le téléverser dans la mémoire du microcontrôleur via un câble USB. [8]

- **Différents Types de cartes Arduino :**

La gamme Arduino propose une grande variété de cartes de développement, dont les modèles principaux sont représentés sur les figures ci-dessous : » [8]

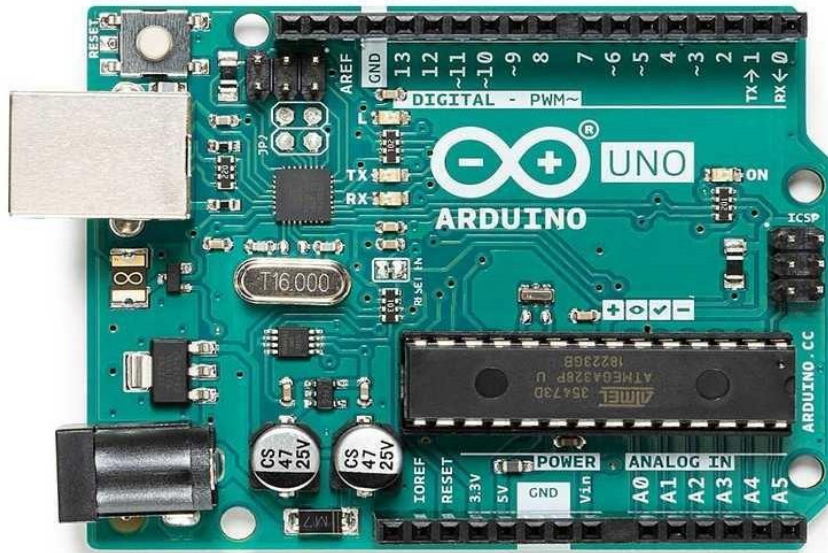


**Figure 2.1 Arduino UNO**

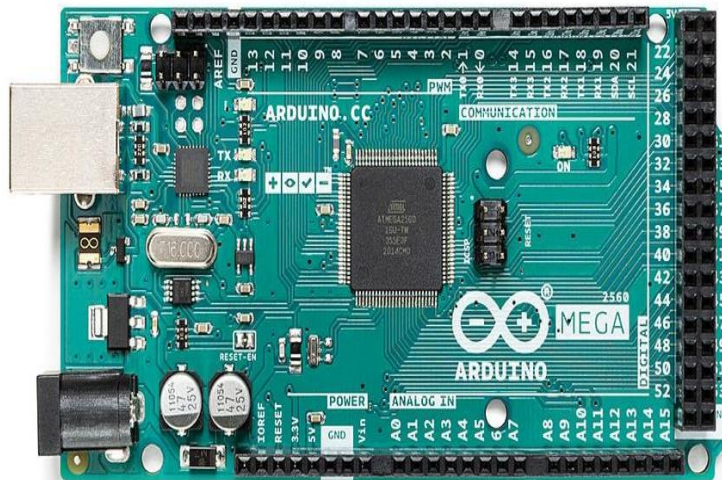


**Figure 2.2 La carte Arduino MEGA 2560**

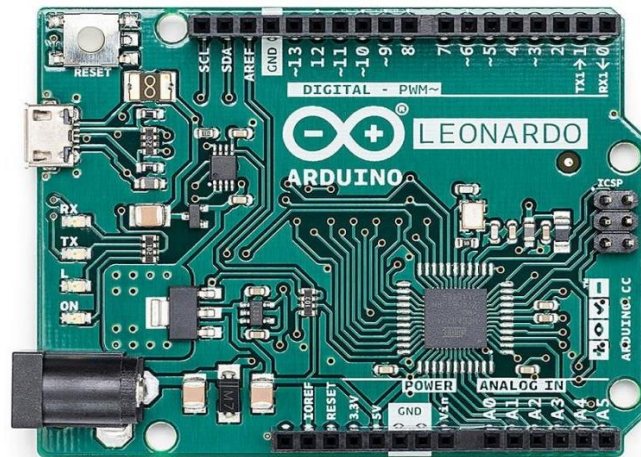


Figure 2.3 La carte Arduino LEONARDO

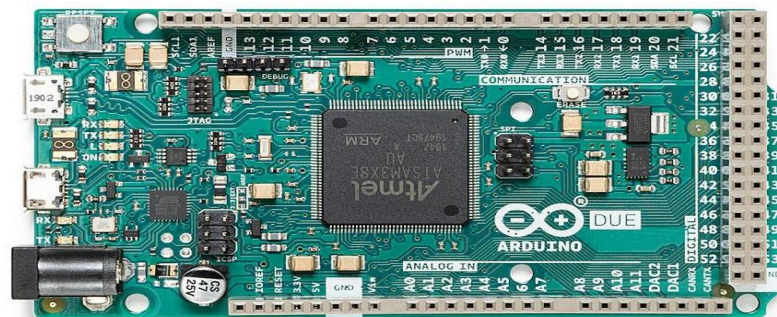


Figure 2.4 La carte Arduino DUE

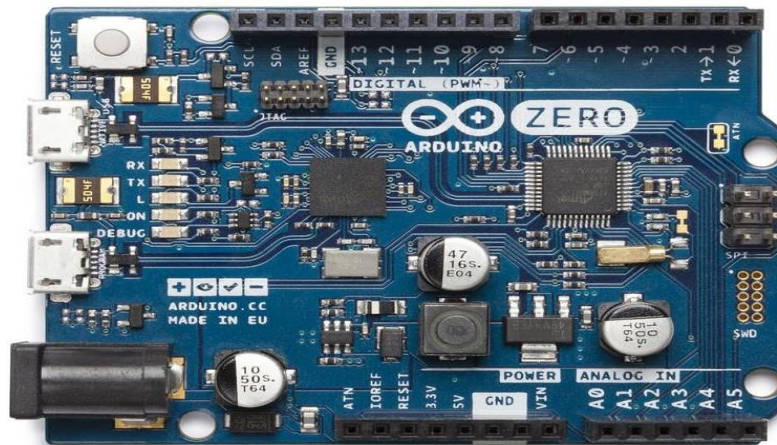


Figure 2.5 La carte Arduino ZERO



Figure 2.6 La carte Arduino MICRO

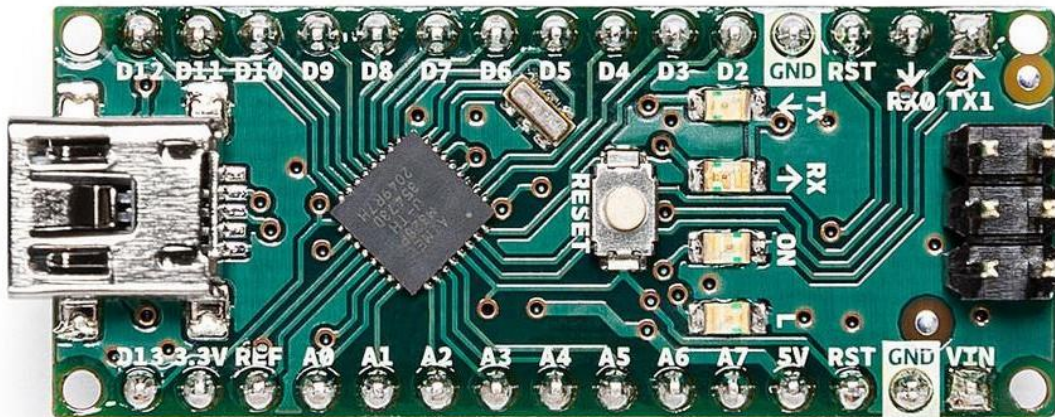


Figure 2.7 La carte Arduino NANO

#### 2.1.1.1. Arduino UNO

- **Définition :**

L'Arduino Uno s'impose comme la carte de référence au sein de la gamme Arduino. Cette plateforme de prototypage électronique en "open source" est privilégiée aussi bien dans le milieu académique qu'industriel pour la conception de systèmes interactifs.

L'architecture de cette carte repose sur le microcontrôleur **ATmega328P**. Elle offre une connectivité complète pour la réalisation de projets d'instrumentation, comprenant notamment :

- **14 broches d'entrée/sortie numériques** : utilisées pour les signaux logiques, dont **6 broches** : sont dotées de la fonctionnalité **PWM**
- **6 entrées analogiques** : permettant la lecture de tensions variables.
- **Interfaces de connexion** : un port USB pour la programmation, un connecteur d'alimentation externe et un port ICSP.
- **Un bouton RESET** : permettant de réinitialiser l'exécution du programme. [8]

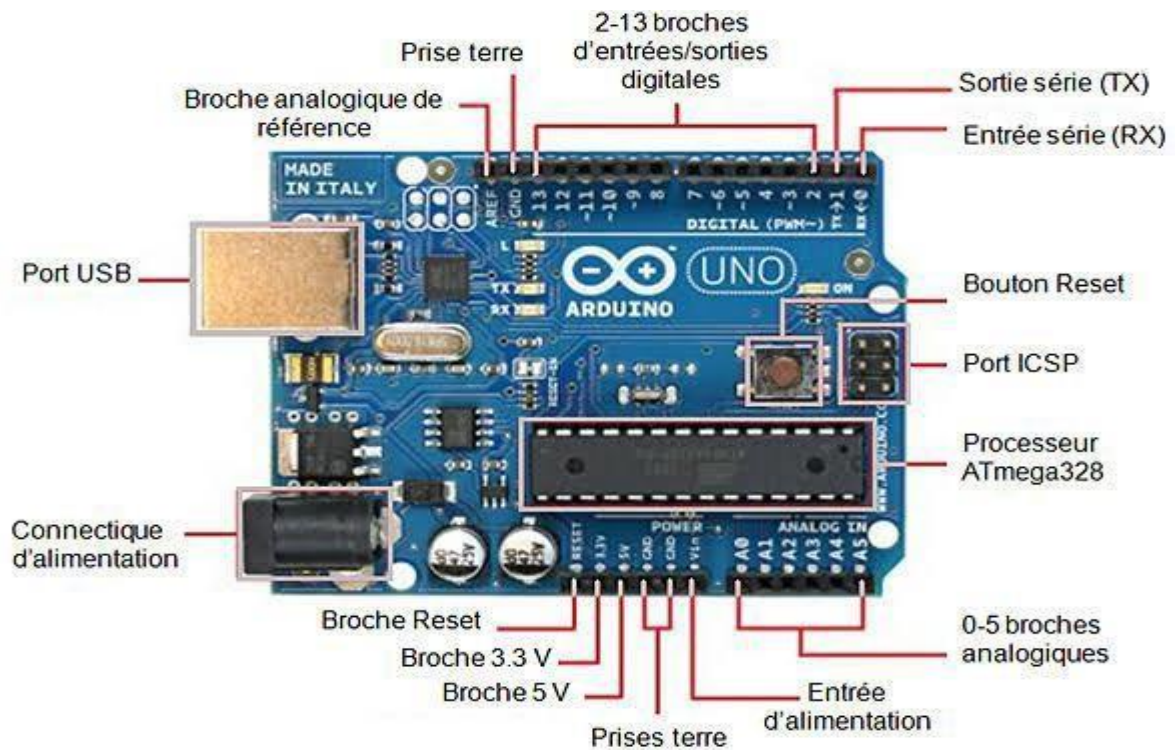


**Figure 2.8 Description de la carte Arduino UNO**

### 2.1.1.2. Le microcontrôleur ATmega328P

Le ATmega328 est une puce microcontrôleur populaire développée par Atmel, désormais une filiale de Microchip Technology. Il est couramment utilisé dans divers projets et applications électroniques en raison de sa simplicité, de sa polyvalence et de sa disponibilité étendue.

L'ATmega328 est un microcontrôleur 8 bits basé sur l'architecture AVR RISC, il offre de nombreuses fonctionnalités. Il dispose d'une mémoire flash de 32 Ko pour le code du programme, 2 Ko de mémoire SRAM et 1 Ko de mémoire EEPROM pour les données.

Avec ses 23 broches d'E/S, dont 14 sont numériques et 6 sont analogiques, il permet une connectivité

étendue. Il intègre un convertisseur analogique-numérique 10 bits et des temporisateurs pour la gestion du temps. De plus, il prend en charge les interfaces de communication UART, SPI et I2C. L'ATmega328 fonctionne dans une plage de tension de 1,8 V à 5,5 V, ce qui le rend adapté à une variété d'applications. [8]

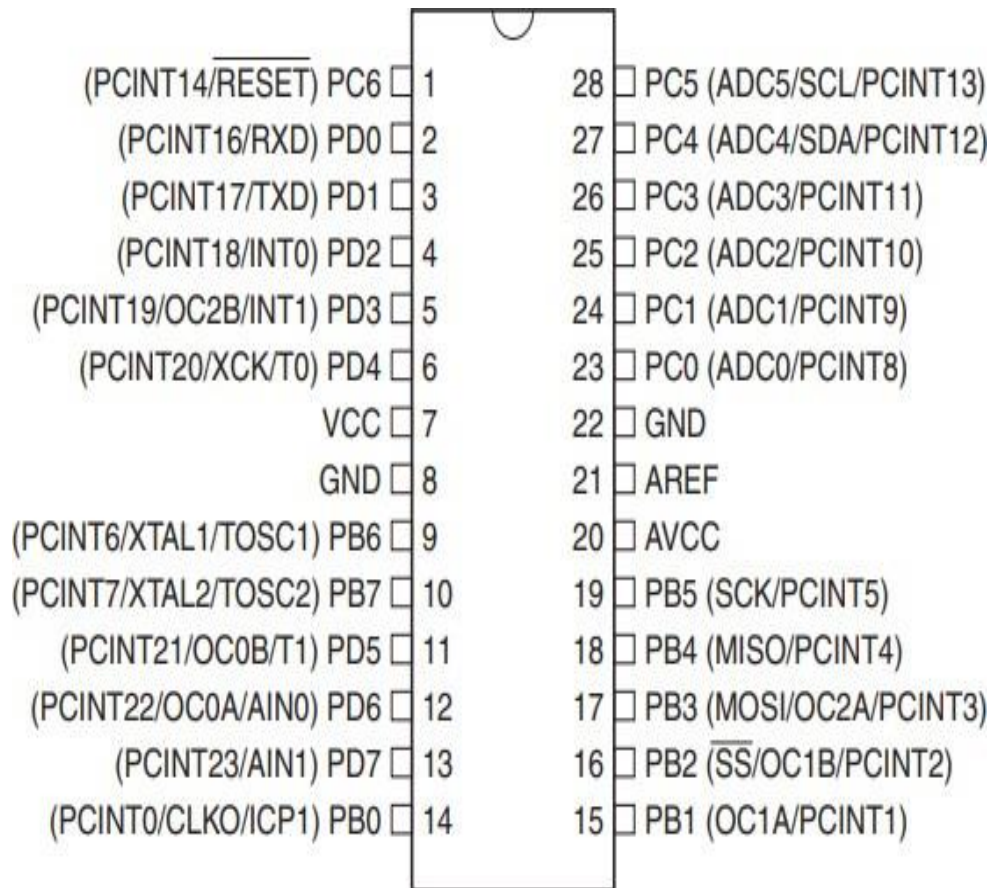


**Figure 2.9 Le microcontrôleur ATmega 328**

### 2.1.1.3. Caractéristiques de l'Arduino UNO

Les caractéristiques de la carte Arduino UNO sont représentées dans le tableau suivant : [8]

|                        |                                |                                  |
|------------------------|--------------------------------|----------------------------------|
| <b>Board</b>           | <b>Name</b>                    | Arduino UNO R3                   |
|                        | <b>SKU</b>                     | A000066                          |
| <b>Microcontroller</b> | ATmega328P                     |                                  |
| <b>USB connecter</b>   | USB-B                          |                                  |
| <b>Pins</b>            | <b>Built-in LED Pin</b>        | 13                               |
|                        | <b>Digital I/O Pins</b>        | 14                               |
|                        | <b>Analog input pins</b>       | 6                                |
|                        | <b>PWM pins</b>                | 6                                |
| <b>Communication</b>   | <b>UART</b>                    | Yes                              |
|                        | <b>I2C</b>                     | Yes                              |
|                        | <b>SPI</b>                     | Yes                              |
| <b>Power</b>           | <b>I/O Voltage</b>             | 5V                               |
|                        | <b>Input voltage (nominal)</b> | 7-12V                            |
|                        | <b>DC Current per I/O Pin</b>  | 20 mA                            |
|                        | <b>Power Supply Connector</b>  | Barrel Plug                      |
| <b>Clock speed</b>     | <b>Main Processor</b>          | ATmega328P 16 MHz                |
|                        | <b>USB-Serial Processor</b>    | ATmega16U2 16 MHz                |
| <b>Memory</b>          | <b>ATmega328P</b>              | 2KB SRAM, 32KB FLASH, 1KB EEPROM |
| <b>Dimensions</b>      | <b>Weight</b>                  | 25 g                             |
|                        | <b>Width</b>                   | 53.4 mm                          |
|                        | <b>Length</b>                  | 68.6 mm                          |

Tableau 2-1 Les caractéristiques de la carte Arduino UNO

#### 2.1.1.4. Configuration du brochage de l'Arduino UNO

- **Broches d'Entrées/Sorties numériques :**

Les broches d'entrées/sorties numériques (D0 à D13) de l'Arduino UNO peuvent être programmées pour fonctionner soit en tant qu'entrées soit en tant que sorties. Elles peuvent être dans deux états : HAUT (HIGH) ou BAS (LOW). Lorsqu'une broche est activée, elle présente une tension de 5V, tandis que lorsqu'elle est désactivée, la tension est de 0V.

Sur la carte Arduino UNO, il y a 6 broches compatibles avec la génération de signaux PWM. Elles sont identifiées par le symbole tilde (~). Les broches générant une PWM sont les suivantes : 3, 5, 6, 9, 10 et 11.

Chaque broche numérique peut supporter un courant maximal de 40 mA et possède une résistance interne variant entre 20 et 50 K $\Omega$ .

En ce qui concerne les broches TX (transmission) et RX (réception), elles sont utilisées pour établir une communication série avec d'autres modules tels que les modules Wi-Fi, Bluetooth, etc. Elles permettent d'échanger des données en utilisant le protocole de communication série. [8]

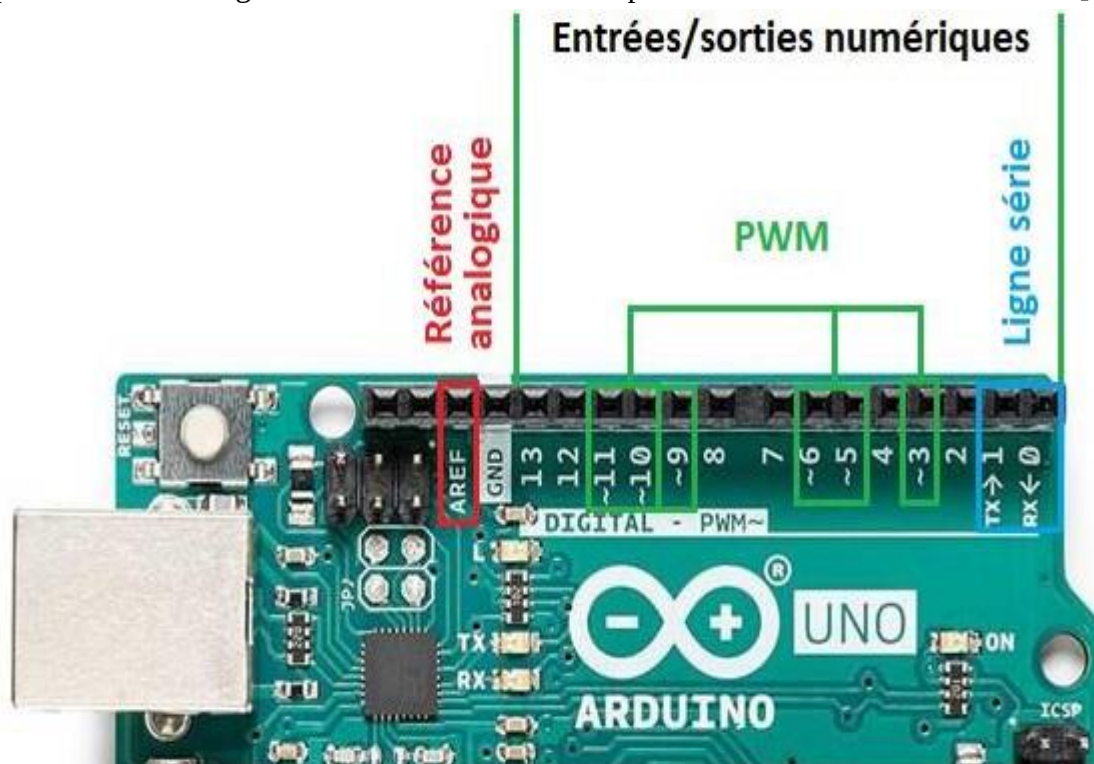


Figure 2.10. Broches d'entrées sorties numériques

- **Broches d'Entrées et Sorties analogiques :**

La carte Arduino UNO dispose de 6 broches analogiques (A0 à A5) qui peuvent être utilisées pour mesurer des signaux analogiques. Ces broches sont utilisées comme entrées/sorties numériques si nécessaire.

Les tensions mesurées sur les broches analogiques de l'Arduino UNO sont toujours comprises entre 0V et 5V. Le convertisseur analogique-numérique (CAN) des Arduino à base d'AVR effectue une conversion sur 10 bits, ce qui signifie qu'il peut représenter les tensions en 1024 niveaux distincts, allant de 0 à 1023. [8]

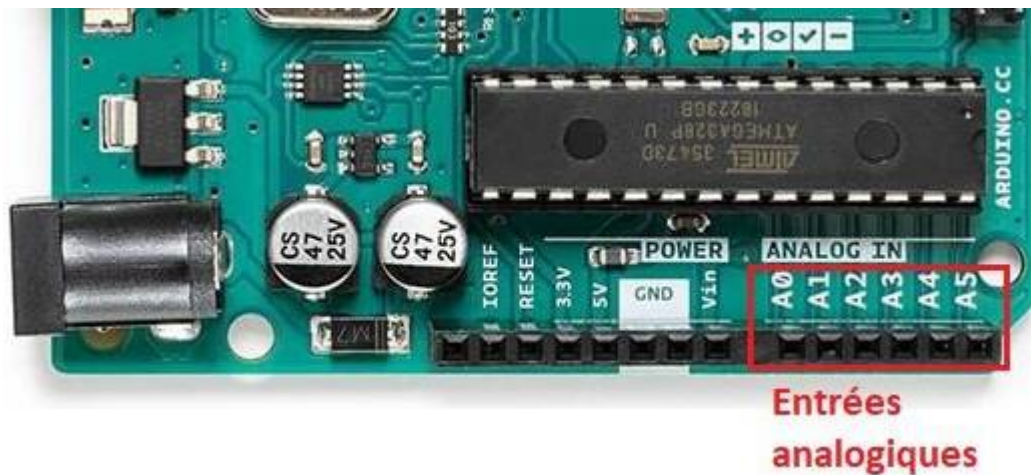


**Figure 2.11 Broches d'entrées sorties analogique de l'Arduino.**

#### **2.1.1.5. Alimentation de la carte Arduino UNO**

Pour fonctionner, la carte a besoin d'une alimentation. Le microcontrôleur fonctionnant sous 5V, la carte peut être alimentée en 5V par le port USB de l'ordinateur à l'aide d'un câble USB Type A/B.

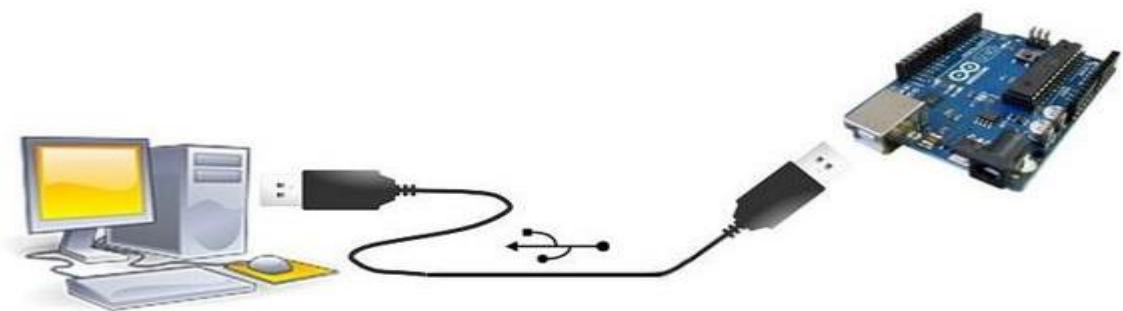


**Figure 2.12 L'alimentation par le port USB.**

Ou bien par une alimentation externe qui est comprise entre 7V et 12V. Cette tension doit être continue et peut par exemple être fournie par une pile 9V.



**Figure 2.13 Alimentation par une pile 9V.**

Ou bien par une alimentation continue stabilisée.

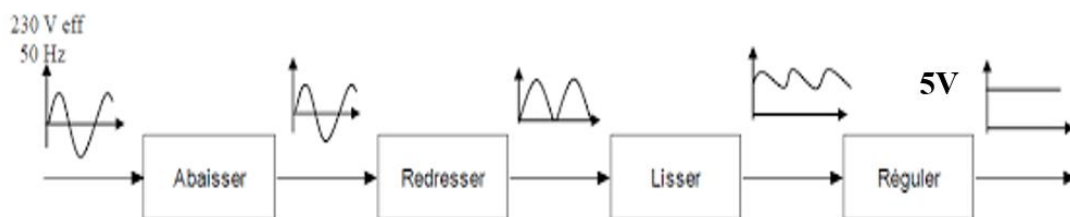


**Figure 2.14 Principe d'une alimentation stabilisée.**

Lorsque la carte Arduino est alimentée par une alimentation externe, un régulateur intégré dans la carte se charge ensuite de réduire la tension à 5V pour le bon fonctionnement de la carte Arduino. [8]

#### **2.1.1.6. Application**

Voici une liste non exhaustive des applications possibles réalisées grâce à Arduino :

- 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 [8]

### 2.1.1.7. IDE Arduino

L'IDE Arduino est le logiciel de développement open source contient un éditeur de texte et tous les outils nécessaires à l'activité de programmation (édition, compilation et charger le programme à la carte Arduino). Le langage de programmation utilisé dans l'IDE Arduino est principalement le C++. L'interface du logiciel Arduino se présente comme illustré par la figure 2.15. [8]

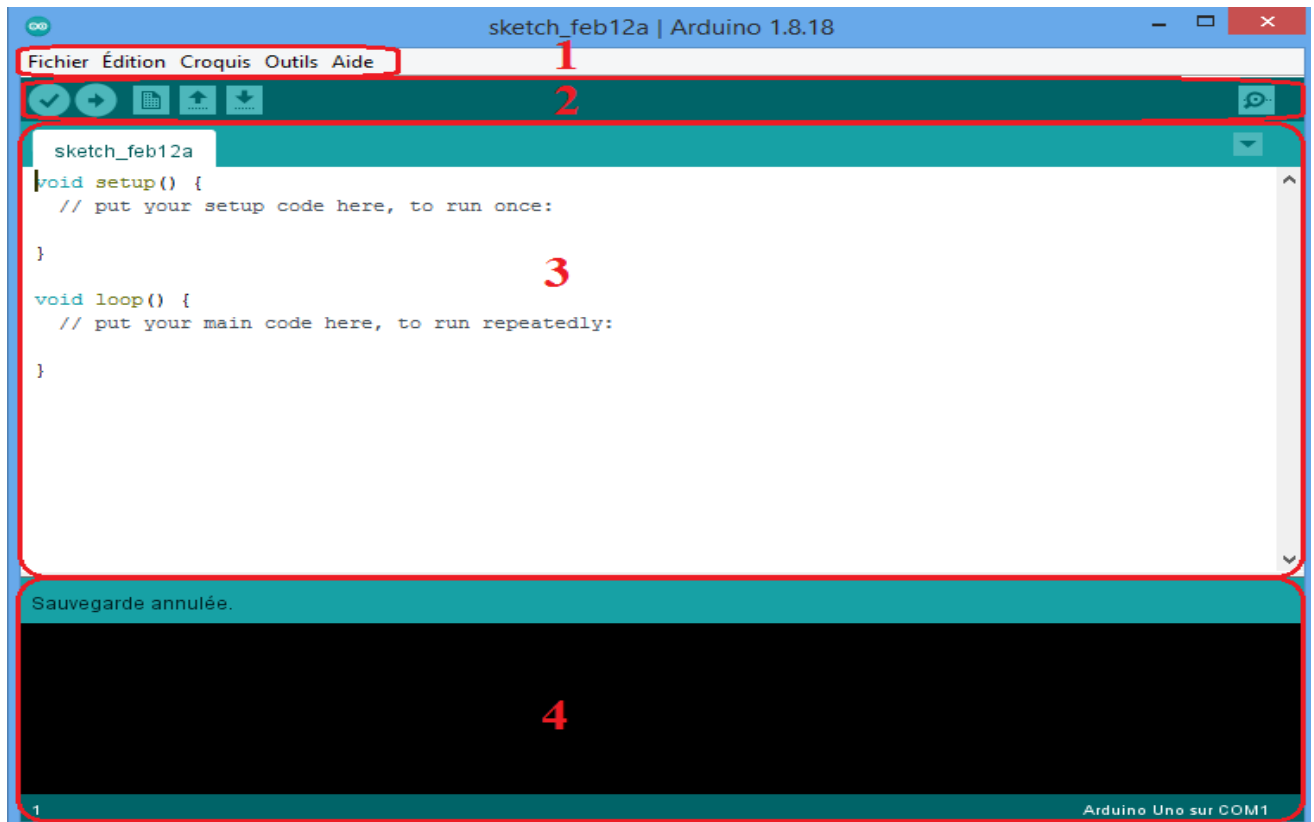


Figure 2.15 L'interface de L'IDE Arduino

|   |                                          |
|---|------------------------------------------|
| 1 | les options de configuration du logiciel |
| 2 | Boutons pour programmer les cartes       |
| 3 | Le programme à crée                      |
| 4 | Le débogueur                             |

Tableau 2-2 Description d'interface d'IDE Arduino

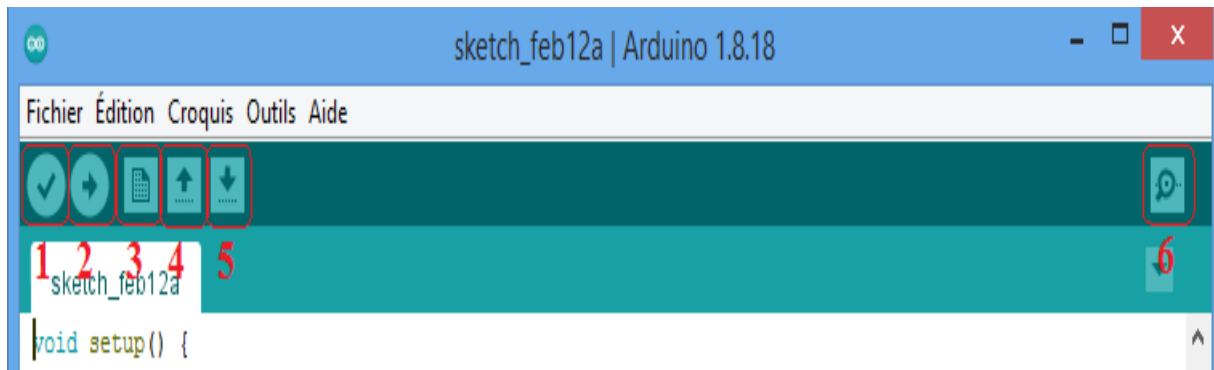


Figure 2.16 Les boutons de l'IDE Arduino

|   |                                                                                                 |
|---|-------------------------------------------------------------------------------------------------|
| 1 | Permet de vérifier le programme, il active un module pour trouver des erreurs dans le programme |
| 2 | Compiler et charger le programme dans la carte Arduino                                          |
| 3 | Crée un nouveau fichier                                                                         |
| 4 | Ouvre un fichier.                                                                               |
| 5 | Enregistre le programme                                                                         |
| 6 | Ouvre le moniteur série                                                                         |

Tableau 2-3 Description des boutons de l'IDE Arduino

## 2.1.2. Le circuit L293D

### 2.1.2.1. Définition

Le L293D est un **quadruple demi-pont en H** qui dispose de quatre canaux de sortie indépendants. Ces canaux peuvent être couplés par paires pour former deux ponts en H complets, permettant ainsi le contrôle bidirectionnel de deux moteurs. [7], [28], [29], [30]

### 2.1.2.2. Caractéristiques Principales

- **Courant de sortie** : Il peut fournir jusqu'à **600 mA** par canal
- **Tension d'alimentation** : Il accepte une tension de puissance  $V_{CC2}$  (ou  $V_{SS}$ ) allant jusqu'à **36V**.
- **Interface Logique** : Entrées compatibles avec les niveaux TTL (5V), avec une broche d'alimentation logique séparée  $V_{CC1}$  (ou  $V_{SS}$ ).
- **Protection Thermique** : Il intègre un mécanisme d'arrêt automatique en cas de surchauffe excessive. [7], [29]

### 2.1.2.3. Description des broches du L293D



Figure 2.17 Les PIN de Le L293D

- **Broche 1 (Enable 1) :** Activation du premier pont en H. Elle permet d'allumer ou d'éteindre les sorties du moteur 1.
- **Broche 2 (Input 1) :** Première entrée pour le moteur 1. Elle doit être connectée à une broche de votre microcontrôleur (comme l'Arduino).
- **Broche 3 (Output 1) :** Première sortie pour le moteur 1. C'est ici que l'on branche le premier fil du moteur.
- **Broches 4 & 5 (GND) :** Masse. Elles doivent être connectées au **0V (GND)**. Ces broches servent aussi de dissipateur thermique.
- **Broche 6 (Output 2) :** Deuxième sortie pour le moteur 1. C'est ici que l'on branche le second fil du moteur.
- **Broche 7 (Input 2) :** Deuxième entrée pour le moteur 1. Elle permet de contrôler le sens de rotation en combinaison avec l'Input 1.
- **Broche 8 (Vcc2) :** Tension d'alimentation en permettant d'alimenter des moteurs à courant continu (entre **4,5V et 36V**).

- **Broche 9 (Enable 2) :** Activation du second pont en H. Elle permet d'allumer ou d'éteindre les sorties du moteur 2.
- **Broche 10 (Input 3) :** Première entrée pour le moteur 2. À connecter au microcontrôleur.
- **Broche 11 (Output 3) :** Première sortie pour le moteur 2.
- **Broches 12 & 13 (GND) :** Masse (0V). Identiques aux broches 4 et 5.
- **Broche 14 (Output 4) :** Deuxième sortie pour le moteur 2.
- **Broche 15 (Input 4) :** Deuxième entrée pour le moteur 2.
- **Broche 16 (Vcc1) :** Tension d'alimentation logique +5V pour alimenter les circuits internes. [7], [29], [30]

#### 2.1.2.4. Schéma interne et principe de fonctionnement du L293D

Le schéma interne se présente sous forme d'un circuit électrique appelé pont en H (fig 2.18). Le processus d'un pont en H classique repose sur l'activation complémentaire de 2 transistors :

Un transistor PNP et un transistor NPN sont activés simultanément lorsqu'une entrée est à l'état bas et l'autre à l'état haut. [7], [28], [30]

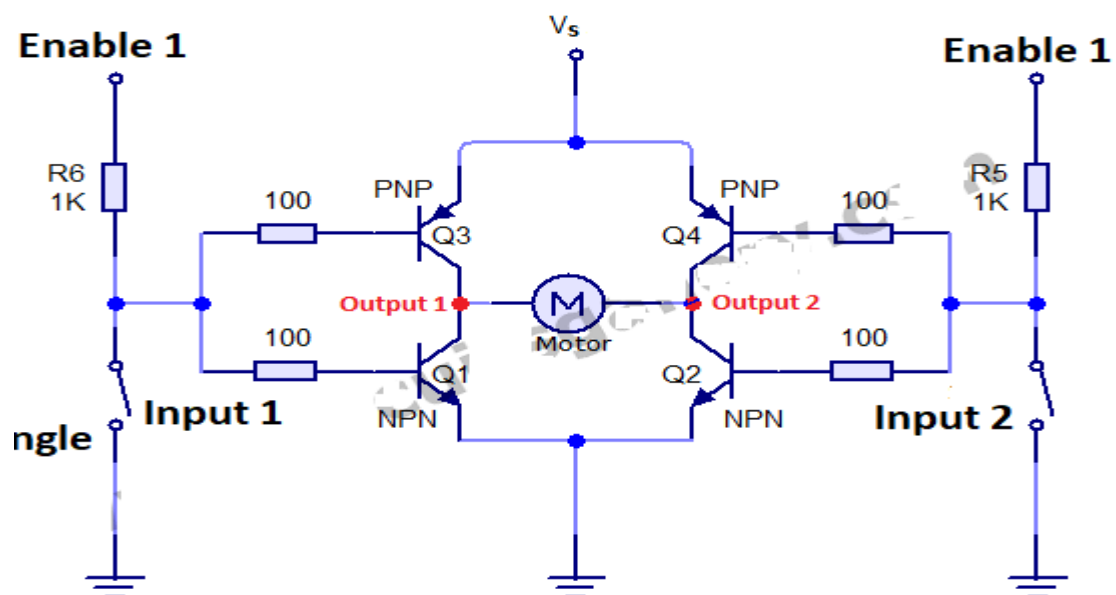


Figure 2.18 Le Schéma interne du L293D

#### 2.1.2.4.1. Fonctionnement du Pont en H : Rotation Horaire

Lorsque l'entrée 1 est fermée (état logique bas) et que l'entrée 2 est ouverte (état logique HAUT), les transistors Q3 et Q2 deviennent passants. , permettant au courant de circuler à travers eux. (Fig 2.19).

Le moteur, connecté entre les sorties 1 et 2, commence alors à tourner. Dans cette configuration, le sens de rotation du moteur est le sens des aiguilles d'une montre (sens horaire) [7], [28]

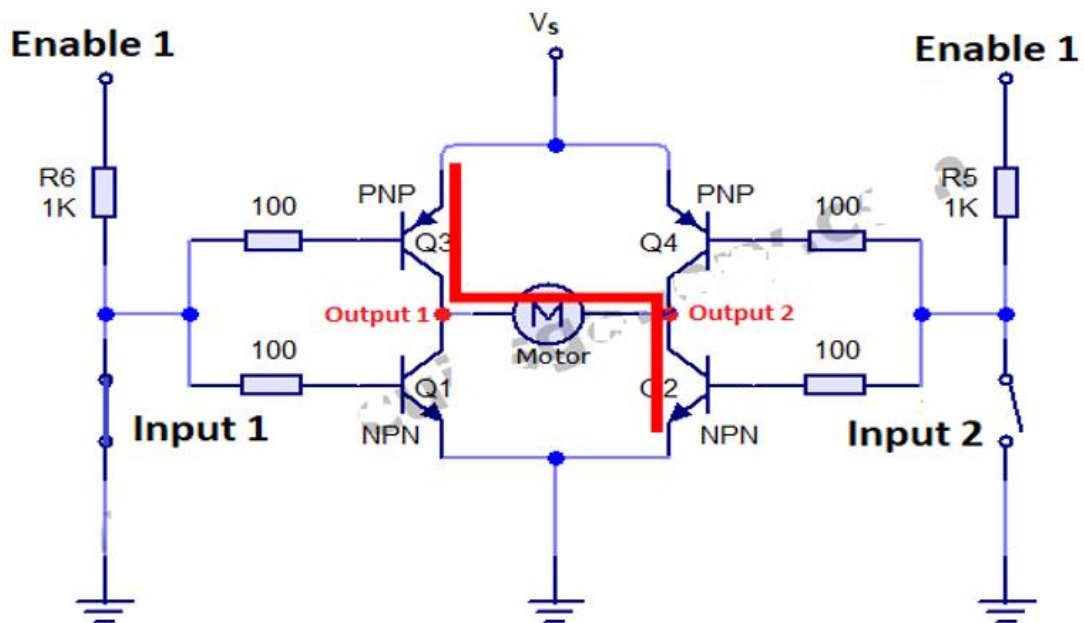


Figure 2.19 Le Fonctionnement du Pont en H : Rotation Horaire

#### 2.1.2.4.2. Fonctionnement du Pont en H : Rotation Anti-Horaire

Lorsque l'entrée 1 est ouverte (état logique haut) et que l'entrée 2 est fermée (état logique bas), les transistors Q1 et Q4 deviennent passants.

Dans ce cas, le courant change de direction à travers le moteur : il circule désormais de la sortie 2 vers la sortie 1. Cette inversion de polarité aux bornes du moteur provoque sa rotation dans le sens anti-horaire (ou sens inverse des aiguilles d'une montre). [7], [28]

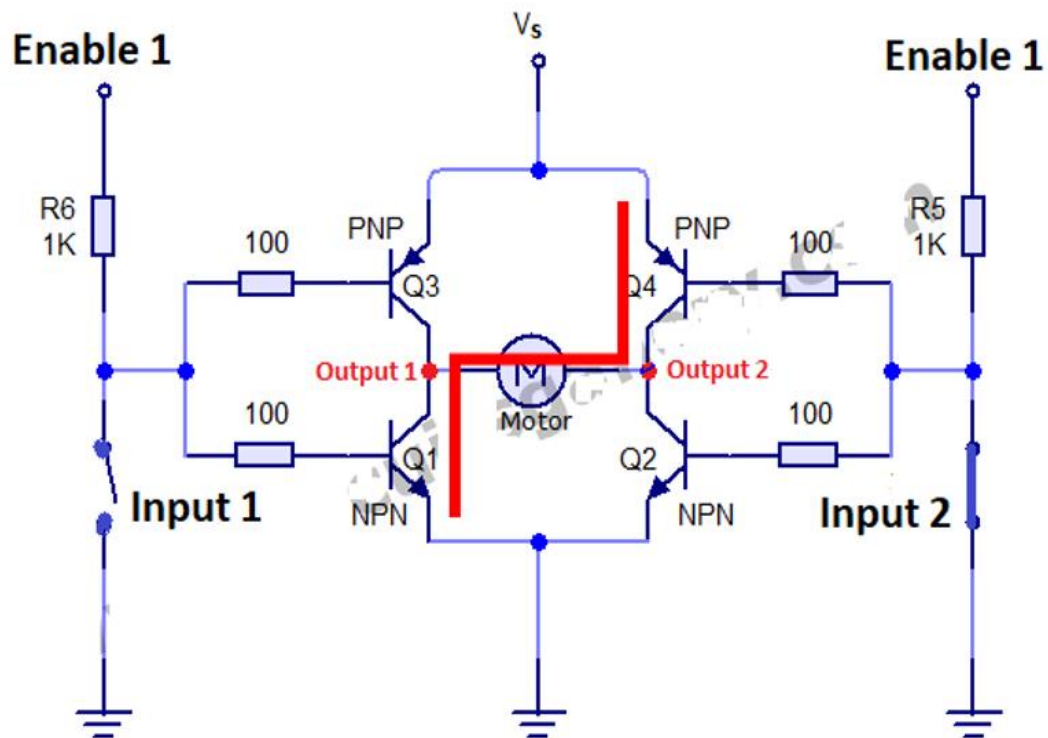


**Figure 2.20 Le Fonctionnement du Pont en H : Rotation anti-Horaire**

Donc on peut conclure la table de commande suivant :

| <b>ENABLE (1)</b>      | <b>INPUT (1)</b>       | <b>INPUT (2)</b>       | <b>Résultat</b>   |
|------------------------|------------------------|------------------------|-------------------|
| Etat « HIGH »<br>« 1 » | Etat « LOW »<br>« 0 »  | Etat « HIGH »<br>« 1 » | Sens Horaire      |
| Etat « HIGH »<br>« 1 » | Etat « HIGH »<br>« 1 » | Etat « Low »<br>« 0 »  | Sens anti Horaire |
| Etat « HIGH »<br>« 1 » | Etat « Low »<br>« 0 »  | Etat « Low »<br>« 0 »  | STOP              |
| Etat « HIGH »<br>« 1 » | Etat « HIGH »<br>« 1 » | Etat « HIGH »<br>« 1 » | STOP              |
| Etat « Low »<br>« 0 »  | X                      | X                      | STOP              |

**Tableau 2-4 Table de Commande de L293D**

### 2.1.2.5. Application des codes sur le L293D

La figure 2.21 illustre le montage du circuit L293D pour la commande d'un moteur à CC à l'arrêt (à l'état initial) :

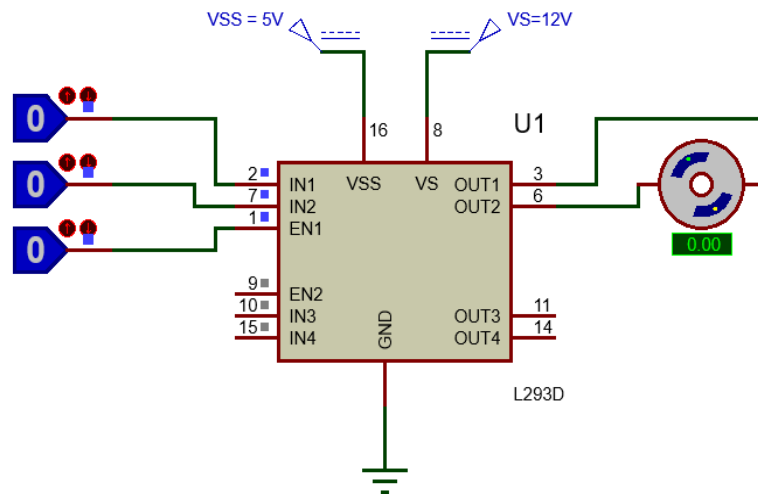


Figure 2.21 Schéma du L293D avec moteur au repos

➤ Le Code « 1 0 1 » (E1 IN1 IN2) :

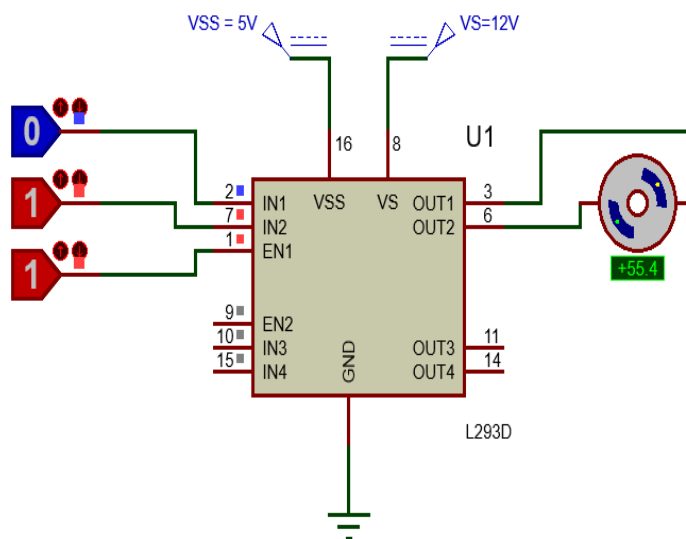


Figure 2.22 commande du moteur en sens horaire

➤ Le Code « 1 1 0 » (E1 IN 1 IN 2):

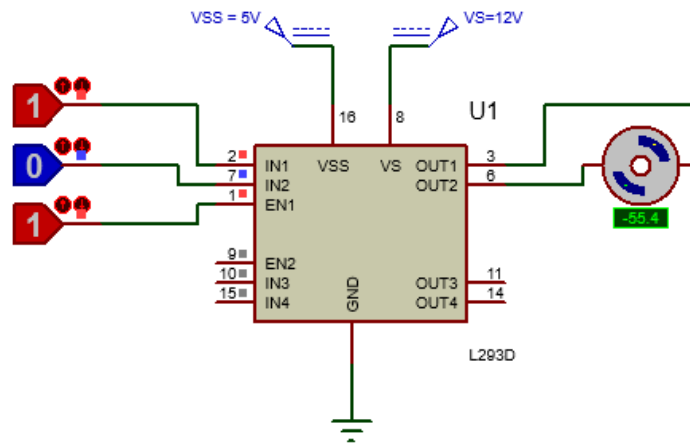


Figure 2.23 commande du moteur en sens anti-horaire

➤ Le Code « 0 1 0 » (E1 IN 1 IN 2) :

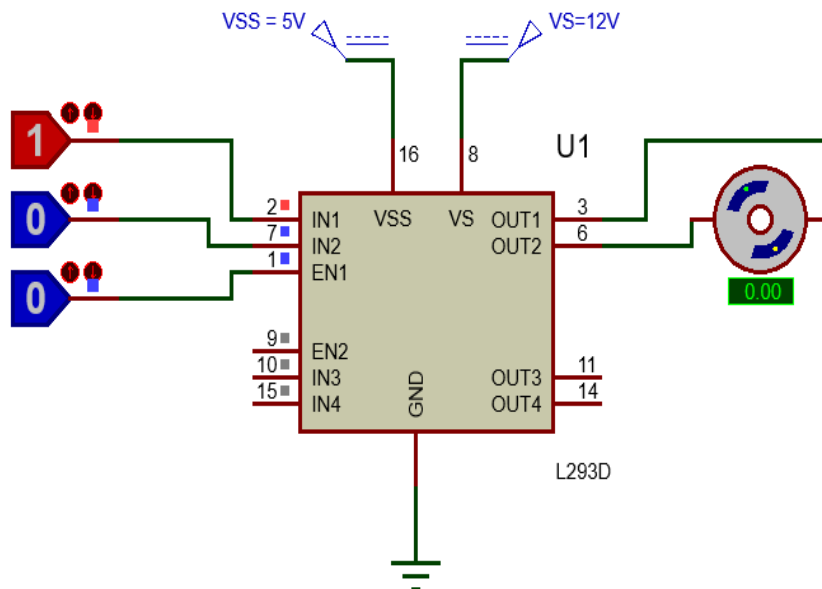


Figure 2.24 commande de l'arrêt du moteur

## 2.1.3. Le Hacheur

### 2.1.3.1. Définition

Un **hacheur** est un dispositif de l'électronique de puissance qui permet de transformer une tension continue fixe en une tension continue réglable (variable). [3], [4], [5], [32]

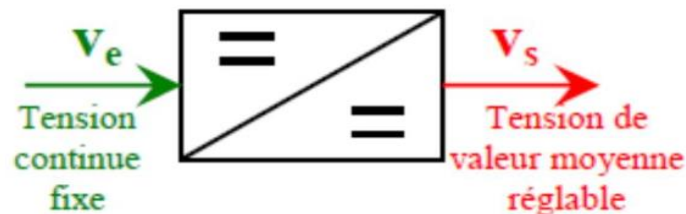


Figure 2.25 schéma bloc du hacheur

### 2.1.3.2. Principe de base

Le hacheur fonctionne comme un interrupteur électronique (utilisant généralement des transistors de puissance comme les MOSFET, IGBT, ...) qui s'ouvre et se ferme à une fréquence élevée. En faisant varier le temps pendant lequel l'interrupteur est fermé par rapport à la période totale ce qu'on appelle le rapport cyclique, noté  $\alpha$ . La valeur moyenne de la tension en sortie dépend de ce rapport cyclique. [3], [4], [5]

### 2.1.3.3. Schéma de base

Pour comprendre le fonctionnement du hacheur, on prend l'exemple suivant (figure 2.26) : [3], [4], [5],

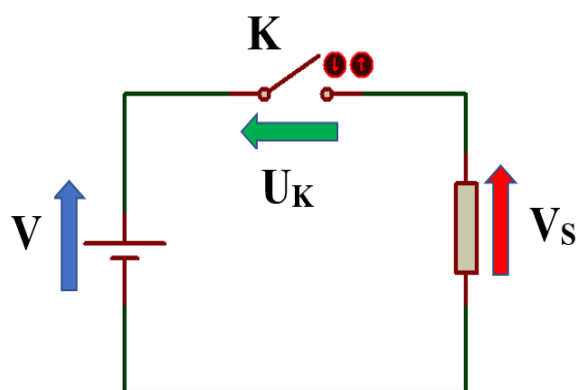


Figure 2.26 schéma de base du hacheur

Avec :

$V$  : tension d'entrée

$K$  : Interrupteur commandable

$V_s$  : tension de sortie

La figure 2.27 représente le temps de l'ouverture et de la fermeture de  $K$  pour le contrôle de tension.

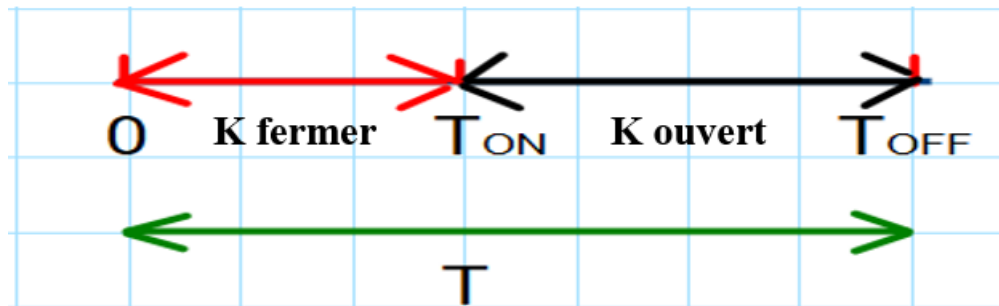


Figure 2.27 Chronogramme de commande

On définit le rapport cyclique  $\alpha$  par : [3], [5]

$$\alpha = \frac{T_{on}}{T} \quad (2.1)$$

Avec :

$$T_{on} = \alpha T \quad (2.2)$$

$$T_{off} = (1 - \alpha)T \quad (2.3)$$

Avec :  $0 \leq \alpha \leq 1$  et  $T$  : la période du signal impulsionnel

- **Le signal de sortie :**

Pour  $K$  fermé  $U_K = 0$  donc selon la loi de mailles :  $V_s = V$

Pour  $K$  ouvert  $U_K \neq 0$  donc il y a aucun passage de courant donc :  $V_s = 0$

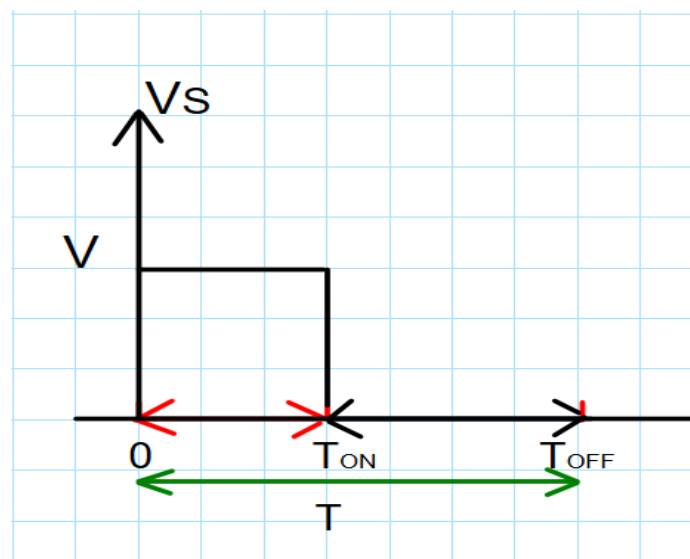


**Figure 2.28 La tension de sortie**

La valeur moyenne :

$$V_{moy} = \frac{1}{T} \int_0^{\alpha T} V dt \quad (2.4)$$

$$V_{moy} = \frac{V}{T} \int_0^{\alpha T} dt$$

$$V_{moy} = \frac{V}{T} \alpha T$$

Alors :

$$V_{moy} = \alpha V \quad (2.5)$$

Donc on peut Contrôler la valeur moyenne de sortie par la variation du rapport cyclique  $\alpha$ .

### 2.1.3.4. Types de hacheurs

#### 2.1.3.4.1. Hacheur série

Le hacheur série (ou abaisseur / Buck) est un convertisseur statique qui abaisse une tension continue fixe pour fournir une tension moyenne de sortie plus faible. Il fonctionne comme une liaison directe entre la source et la charge grâce à un interrupteur commandé (comme un MOSFET) qui s'ouvre et se ferme rapidement. [3], [4], [5]

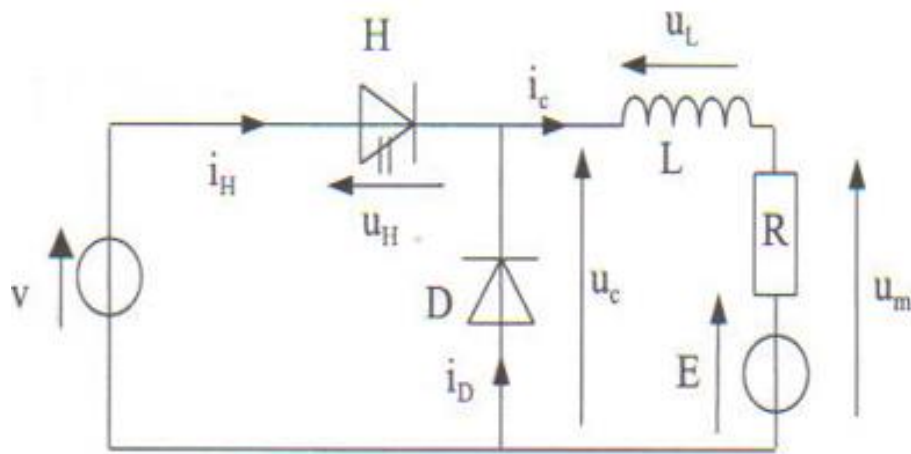


**Figure 2.29 Hacheur série**

#### 2.1.3.4.2. Hacheur parallèle

Le hacheur parallèle est un convertisseur statique qui permet d'obtenir une tension moyenne de sortie supérieure à sa tension d'entrée. Contrairement au montage série, l'interrupteur commandé est placé en parallèle avec la charge. L'énergie provenant de la source est d'abord stockée

temporairement dans une inductance, puis restituée à la charge avec une tension amplifiée lors de l'ouverture de l'interrupteur K. [3], [4], [5]

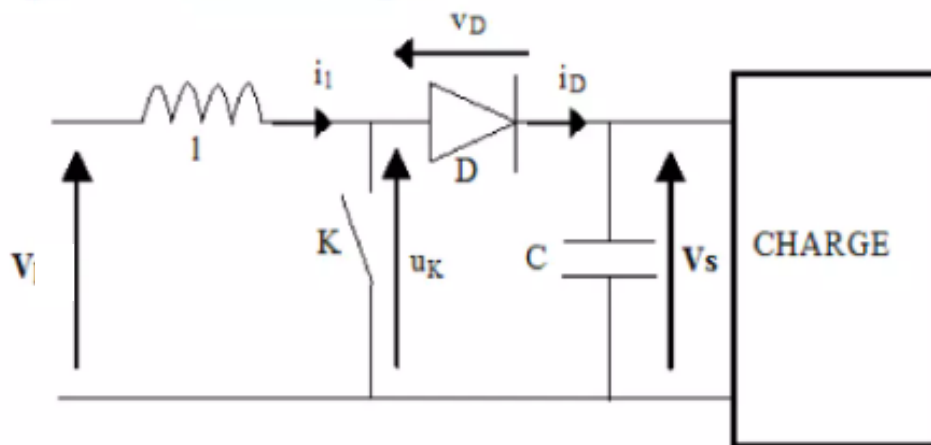


Figure 2.30 Hacheur parallèle

#### 2.1.3.4.3. Hacheur parallèle-série

Le hacheur parallèle-série (plus communément appelé hacheur à accumulation ou Buck-Boost) est un convertisseur statique qui permet de délivrer une tension moyenne de sortie soit inférieure, soit supérieure à sa tension d'entrée, tout en inversant sa polarité. [3], [4], [5]

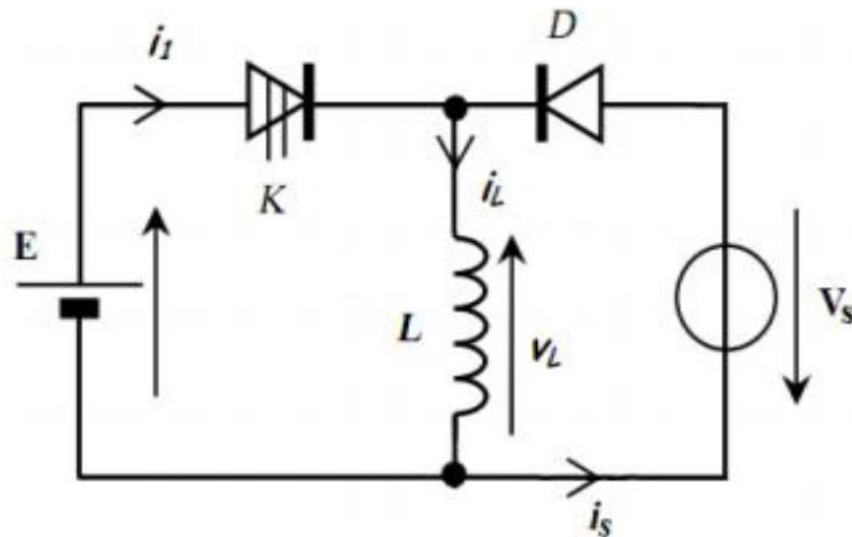


Figure 2.31 Hacheur parallèle-série

#### 2.1.3.5. Technique de commande du hacheur

La performance d'un convertisseur statique repose essentiellement sur la stratégie de pilotage de ses interrupteurs de puissance. Le choix de la technique de commande est dicté par les contraintes spécifiques de l'application visée (précision, rendement, taux d'harmoniques). On distingue principalement quatre méthodes de commande :

- **Commande par signaux carrés (Pleine onde) :** La méthode la plus simple, où les interrupteurs basculent une seule fois par période. Elle est robuste mais génère un taux d'harmoniques élevé.
- **Commande par onde en marches d'escalier :** Une amélioration de la commande carrée utilisant plusieurs niveaux de tension pour se rapprocher d'une forme sinusoïdale, réduisant ainsi les distorsions.
- **Commande par découpage :** Consiste à commuter les interrupteurs à une fréquence supérieure à la fréquence fondamentale pour segmenter la puissance transmise.
- **Modulation de Largeur d'Impulsion (MLI / PWM) :** C'est la technique la plus performante. En faisant varier la durée des impulsions à fréquence constante, elle permet de réguler la valeur moyenne de la tension tout en rejetant les harmoniques vers les hautes fréquences, facilitant ainsi le filtrage. [3], [4], [5]

#### 2.1.3.6. Principe de MLI

La carte Arduino Uno, équipée du microcontrôleur ATmega328P, permet un contrôle précis des actionneurs grâce à ses 6 sorties PWM localisées sur les broches numériques D3, D5, D6, D9, D10 et D11. Cette architecture permet de générer un signal électrique à fréquence fixe dont le rapport cyclique est ajustable dynamiquement. Ce procédé est essentiel pour le pilotage de précision du Moteur à Courant Continu ou le réglage de l'intensité d'indicateurs lumineux, en simulant une tension analogique variable à partir d'un signal numérique. [8], [15], [18], [20], [21], [23], [24], [26]

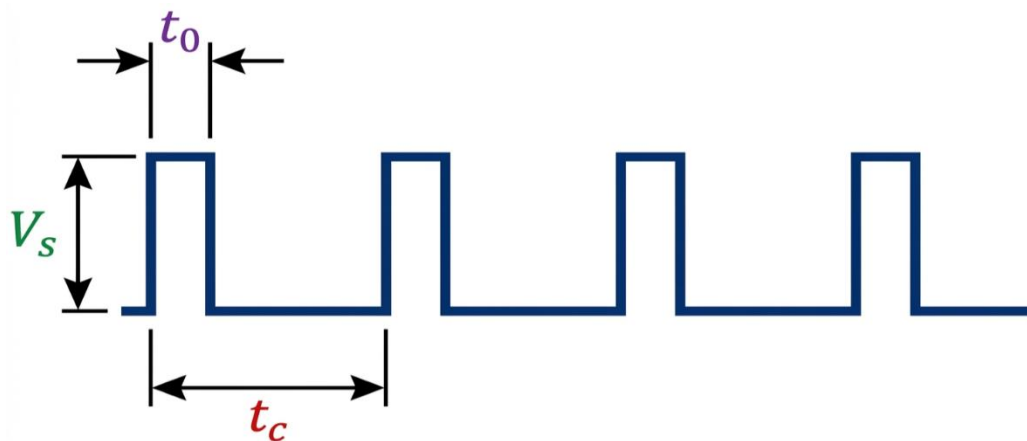


Figure 2.32 Forme d'onde d'un signal PWM

Avec :

$t_0$  : durée de l'impulsion

$t_c$  : période du signal

$V_s$  : L'amplitude de la tension

Lorsqu'un moteur est alimenté par ce type de tension, tout se passe comme s'il était alimenté par une tension continue moyenne ajustable entre 0 et  $V_s$

La figure 2.33 montre que la tension moyenne de sortie dépend du rapport cyclique du signal PWM

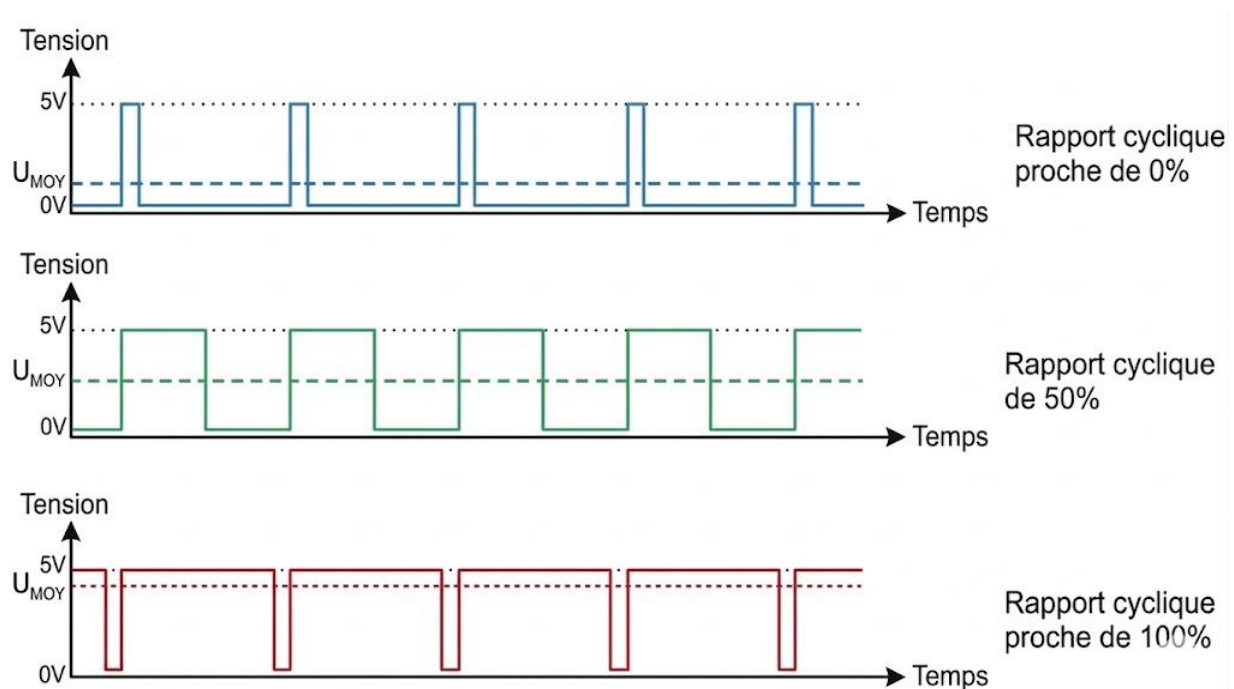


Figure 2.33 Valeur moyenne d'un signal PWM

## 2.1.4. L'afficheur LCD

### 2.1.4.1. Présentation générale

L'afficheur LCD (Liquid Crystal Display) de référence LM016L est un module d'affichage alphanumérique matriciel largement utilisé dans les systèmes embarqués pour restituer des informations textuelles en temps réel. Il s'agit d'un afficheur de type 16x2, ce qui signifie qu'il peut afficher 16 caractères par ligne sur 2 lignes au total (soit 32 caractères simultanément).

Chaque caractère est généré à l'intérieur d'une matrice de pixels (généralement 5x7 ou 5x11 pixels). Ce module intègre son propre contrôleur, qui gère la mémoire d'affichage et traduit les commandes reçues du microcontrôleur en signaux électriques pour activer les cristaux liquides. [9]

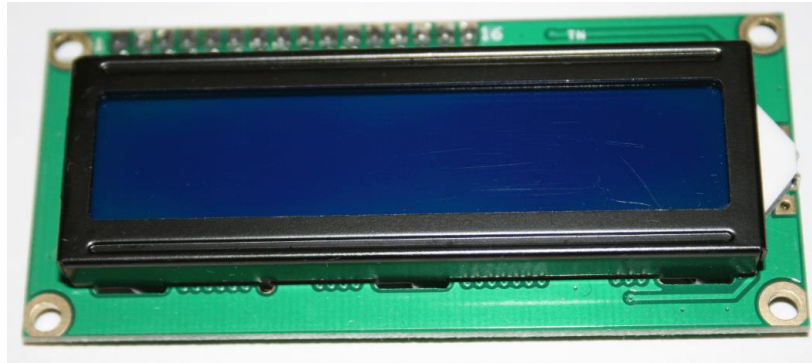


Figure 2.34 LCD

#### 2.1.4.2. Brochage de LCD

Le module LM016L dispose d'un connecteur standard à 14 broches. Le tableau suivant représente la fonction de chaque broche : [9]

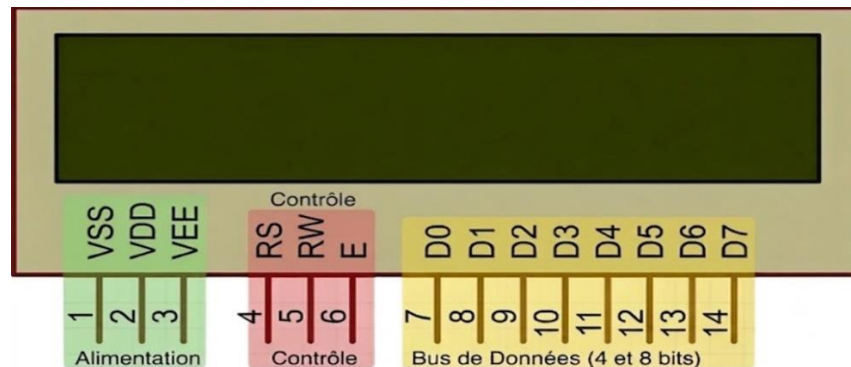


Figure 2.35 brochage de LCD

| Broche (N°) | Nom   | Type         | Description (FR)                          |
|-------------|-------|--------------|-------------------------------------------|
| 1           | VSS   | Masse        | Masse (0V)                                |
| 2           | VDD   | Alimentation | Alimentation (+5V)                        |
| 3           | VEE   | Contraste    | Réglage Contraste (via potentiomètre)     |
| 4           | RS    | Contrôle     | Sélection Register (0=Commande, 1=Donnée) |
| 5           | RW    | Contrôle     | Lecture/Écriture (0=Écriture, 1=Lecture)  |
| 6           | E     | Contrôle     | Activation (Validation)                   |
| 7-14        | D0-D7 | Données      | Bus de Données Parallèle (8 bits)         |

Tableau 2.5 les Broches de l'afficheur LCD

### 2.1.4.3. Mode de fonctionnement

L'afficheur permet deux modes de communication avec le microcontrôleur :

- **Mode 8-bits** : Utilise l'ensemble du bus de données (**D0 à D7**). C'est le mode le plus rapide car les données ou les commandes sont envoyées en un seul cycle d'horloge.
- **Mode 4-bits** : Utilise uniquement les broches de poids fort (**D4 à D7**). Les données de 8 bits sont alors fractionnées et envoyées en deux étapes (les 4 bits de poids fort d'abord, puis les 4 bits de poids faible).

### 2.1.4.4. Présentation des pins de LCD

Ces broches peuvent être regroupées en trois catégories fonctionnelles : **l'alimentation, le contrôle, et le bus de données.**

#### ➤ Les Broches d'Alimentation et de Contraste (1 à 3)

- **Broche 1 : VSS (Masse)**
  - C'est la référence de potentiel (0V) du circuit. Elle doit être connectée à la masse du microcontrôleur (GND).
- **Broche 2 : VDD (Alimentation Logique)**
  - Elle fournit la tension nécessaire au fonctionnement des circuits logiques internes de l'afficheur. Elle est généralement alimentée en +5V
- **Broche 3 : VEE ou V0 (Réglage du Contraste)**
  - Cette broche contrôle l'opacité des cristaux liquides (le **Contraste** des caractères).

#### ➤ Les Broches de Contrôle (4 à 6)

- **Broche 4 : RS (Register Select - Sélection du Registre)**
  - Elle indique à l'afficheur si l'information envoyée sur le bus de données est une commande de configuration ou un caractère à afficher.
    - ❖ **Niveau Bas (0)** : Mode Commande (ex: effacer l'écran, déplacer le curseur, initialiser l'écran).
    - ❖ **Niveau Haut (1)** : Mode Donnée (ex: envoyer le code ASCII de la lettre 'A' pour l'afficher).
- **Broche 5 : R/W (Read/Write - Lecture/Écriture)**
  - Elle détermine le sens du transfert de données entre le microcontrôleur et le LCD.
    - ❖ **Niveau Bas (0)** : Mode Écriture (Write). Le microcontrôleur envoie des données au LCD.

❖ **Niveau Haut (1) : Mode Lecture (Read).** Le microcontrôleur lit des informations ou l'état interne du LCD.

- **Broche 6 : E (Enable - Validation)**

- C'est la broche de synchronisation (l'horloge du LCD). Elle permet de valider les données présentes sur les broches D0 à D7.
- **Fonctionnement** : Pour que le LCD lise les données qu'on lui présente, le microcontrôleur doit générer une impulsion : envoyer un niveau haut (1), attendre quelques microsecondes, puis appliquer un **front descendant** (passer de 1 à 0). C'est lors de ce passage à 0 que le LCD enregistre l'information.

- **Le Bus de Données (7 à 14)**

- **Broches 7 à 14 : D0 à D7 (Data Bus)**

Ce bus bidirectionnel de 8 bits transporte les commandes logiques ou les codes ASCII des caractères. [9]

## Conclusion du chapitre 2

Ce chapitre a permis d'étudier l'architecture matérielle globale de notre système de motorisation en mettant en évidence la complémentarité entre l'électronique de commande et l'électronique de puissance. Le cœur du système s'articule autour de la carte microcontrôleur Arduino Uno, chargée de générer les signaux de modulation de largeur d'impulsion (PWM) pour le contrôle de la vitesse et de gérer l'interface utilisateur via un afficheur LCD.

L'étude s'est basée également sur l'étage de puissance qui intègre une carte d'interface L293D servant de pré-driver, associée à un circuit de pont en H configurée en hacheur pour assurer une variation de vitesse bidirectionnelle et réversible. [3], [4], [7], [8], [15], [18], [23]

## **CHAPITRE 03 : SIMULATION DE LA CARTE DE CONTROLE DE VITESSE**

## Introduction

L'objectif de ce chapitre est de simuler une carte électronique complète destinée au contrôle de vitesse d'un moteur à courant continu en utilisant une carte Arduino Uno, un circuit d'adaptation de puissance L293D et un pont en H intégré. Le système est alimenté par une batterie similaire à celles utilisées dans certaines applications de mobilité électrique. Un afficheur LCD est également intégré afin d'informer l'utilisateur sur la vitesse du moteur ainsi que sur l'état de la batterie.

Dans un premier temps, un schéma synoptique global sera présenté afin de mettre en évidence les différents blocs fonctionnels du système et les échanges d'informations entre eux. Ensuite, le schéma électrique complet sera réalisé sous l'environnement ISIS Proteus en intégrant l'ensemble des composants nécessaires à la simulation.

Par la suite, un programme de commande est développé sous l'environnement Arduino IDE, après compilation le programme est intégré dans la simulation Proteus. Enfin, les résultats obtenus seront illustrés par des figures afin de vérifier le bon fonctionnement de la carte électronique et de valider le principe de commande adopté. [7], [8], [9], [15], [18], [19]

### 3.1 Présentation du Logiciel Proteus

PROTEUS représente une suite logicielle de conception assistée par ordinateur (CAO) destinée au domaine de l'électronique.

Cet environnement, développé par la société Labcenter Electronics, permet la conception et la simulation d'un système électronique complet y compris le code du microcontrôleur. De manière générale, le logiciel PROTEUS est constitué principalement de deux modules :

- **ISIS** : permet d'éditer et de vérifier le fonctionnement du circuit. [9], [19]

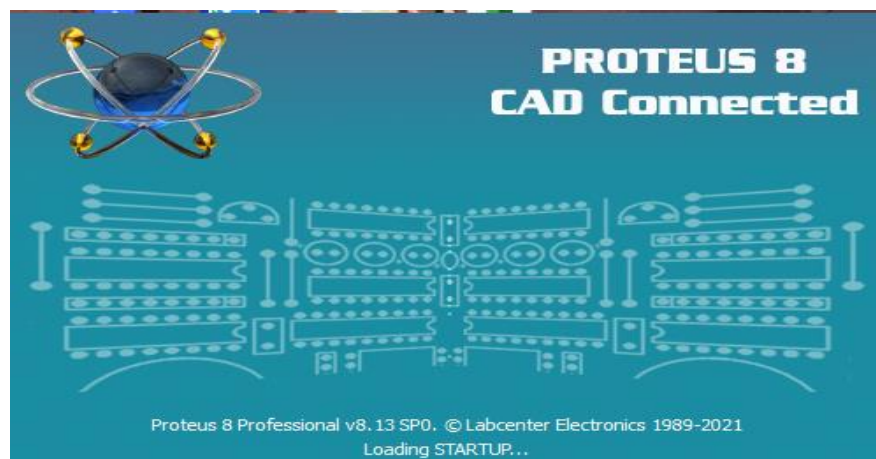


**Figure 3.1 Fenêtre de lancement ISIS de PROTEUS**

- **ARES** : est un outil destiné à la réalisation des circuits électroniques imprimés.



**Figure 3.2 Fenêtre de lancement ARES de PROTEUS**

#### **Avantages de PROTEUS :**

- Pack contenant des logiciels facile et rapide à comprendre et Utiliser.
- Le support technique est performant.
- L'outil de création de prototype virtuel permet de réduire les coûts Matériel et logiciel lors de la conception d'un projet. [9], [19]

### **3.1.1 Présentation de l'environnement ISIS de PROTEUS**

Le logiciel **ISIS** (Intelligent Schematic Input System), est principalement reconnu pour la saisie et l'édition de schémas électroniques. Mais le logiciel va plus loin en offrant deux types d'applications complémentaires :

- **Simulation et validation technique** : Il permet de tester virtuellement le comportement des circuits afin d'en vérifier le bon fonctionnement et de détecter d'éventuelles erreurs de conception dès la phase préparatoire.
- **Documentation et publication** : Grâce à une gestion avancée et soignée de l'aspect graphique, les circuits conçus sur ISIS offrent un aspect visuel optimal, idéal pour être directement intégrés dans des documentations techniques. [9]

### **3.1.2 Présentation de l'interface d'ISIS de PROTEUS**

Le lancement de **PROTEUS** présente une interface utilisateur standard de type Windows, articulée autour d'une fenêtre principale et de plusieurs barres d'outils. En plus des menus classiques dédiés à la gestion des fichiers, de l'affichage et des options de projet, cette interface s'organise autour de trois éléments clés :

- **La zone de travail** : L'espace central destiné au développement, à la simulation et au test des circuits électroniques.

- **Le sélecteur / bibliothèque d'objets** : Une liste latérale qui répertorie tous les composants électriques et électroniques actifs ou disponibles pour l'application en cours.
- **Le panneau de contrôle de l'animation (touches magnétoscope)** : Un ensemble de raccourcis situés généralement en bas à gauche, permettant de piloter la simulation (lancement, pause, exécution pas à pas et arrêt). [9]

Ceci est représenté sur la figure (3.2)

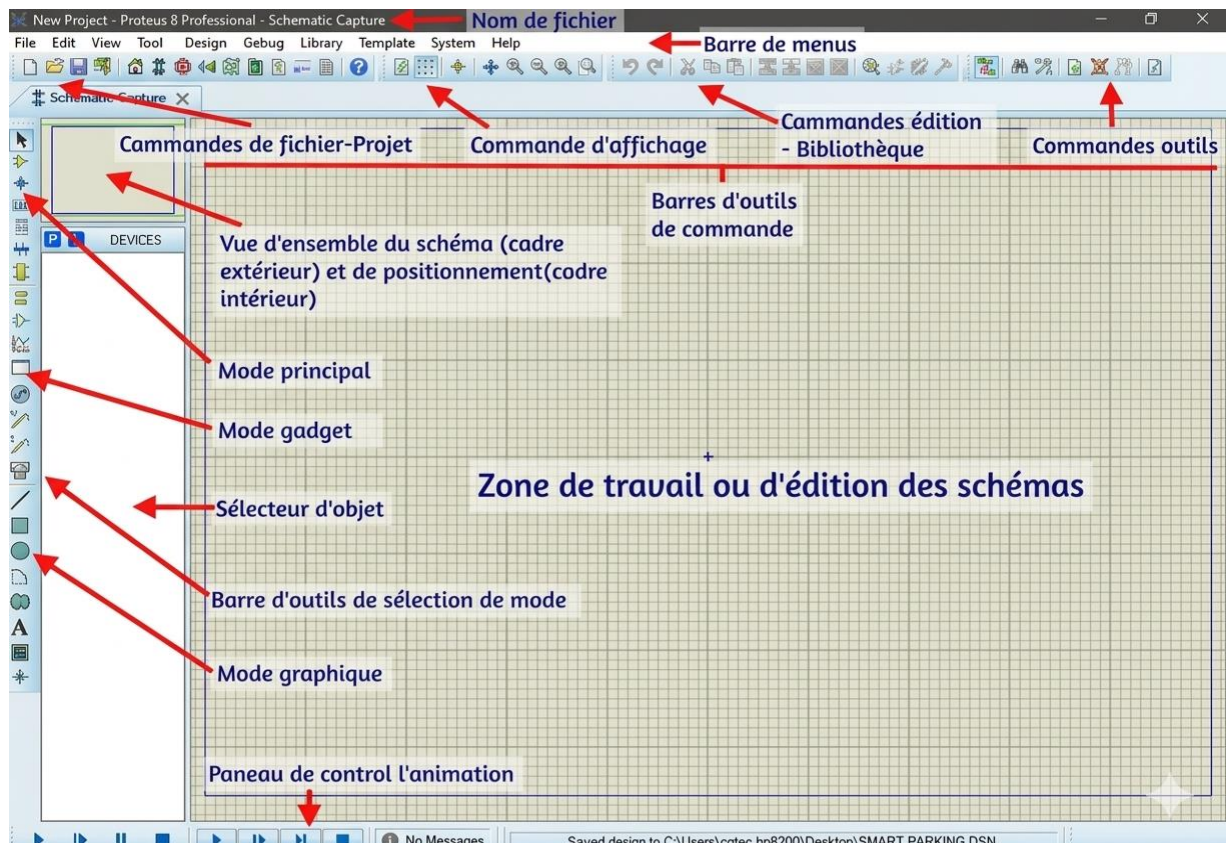


Figure 3.3 environnement de travail ISIS de Proteus

### 3.1.3 La barre d'outils principale

La barre verticale de boutons, comprend les principaux raccourcis nécessaires au développement rapide d'applications. En plaçant le curseur de la souris sur un bouton, sans cliquer, une info-bulle affiche le nom du bouton, dont les plus importants sont :

#### Component mode (Mode Composant)

Un clic sur ce bouton puis Component d'après libraires, permet l'ajout de différents composants.

#### Generation Mode (Mode Générateur)

Permettant l'accès aux différents types de générateurs.

✓ **Instruments**

Un raccourci permettant l'ajout des appareils de mesure tels que le voltmètre, l'ampèremètre, et l'oscilloscope.

✓ **Terminal**

Permettant d'ajouter des points particuliers dans un schéma tels que les entrées/sorties ou Gnd.

Ceci est représenté dans la figure (3.3) suivante : [9]

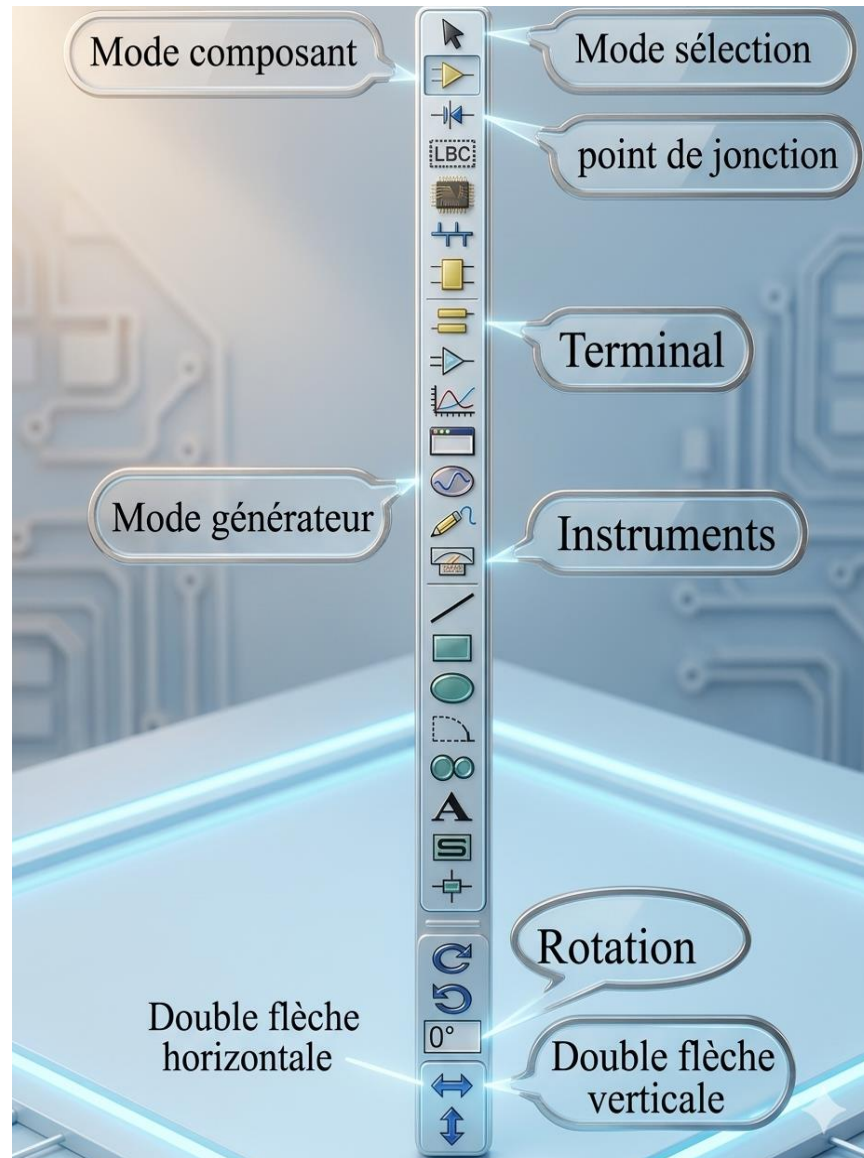


**Figure 3.4 La barre d'outils principale**

### 3.1.4 Présentation de l'interface ARES de PROTEUS

Le logiciel **ARES** (Advanced Routing and Editing Software) est l'outil d'édition et de routage de la suite PROTEUS qui complète parfaitement ISIS. Un schéma électronique conçu sous ISIS peut ainsi être importé directement dans ARES afin de concevoir le circuit imprimé (PCB) de la carte. Ce processus de réalisation s'organise généralement en deux étapes principales :

- **Le placement des composants** : Les différents éléments du circuit sont d'abord positionnés sur la carte en fonction de leur empreinte (*package* ou boîtier) disponible.
- **Le routage des pistes** : Les connexions électriques entre les composants sont ensuite réalisées. Pour cela, le logiciel propose deux approches : le **routage manuel** (généralement plus précis et efficient pour optimiser les pistes) ou le **routage automatique**.
- ARES intègre une fonctionnalité puissante de **visualisation en 3D**, permettant d'obtenir un aperçu réaliste et complet de la carte électronique avant même de lancer sa fabrication physique. [9]

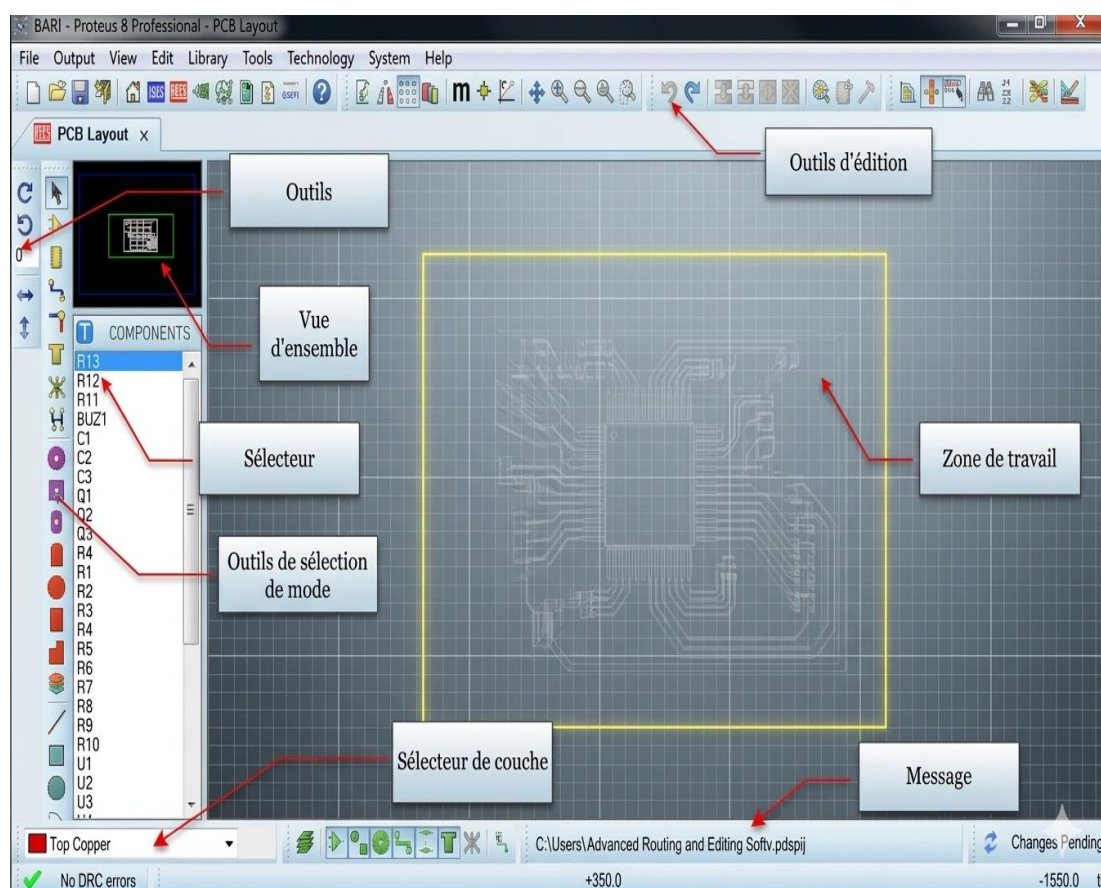


Figure 3.5 Environnement de Travail ARES de PROTEUS

## 3.2 Schéma synoptique du circuit

Notre montage peut être représenté selon le schéma synoptique suivant, qui décrit les principaux blocs fonctionnels de la carte de contrôle de vitesse.

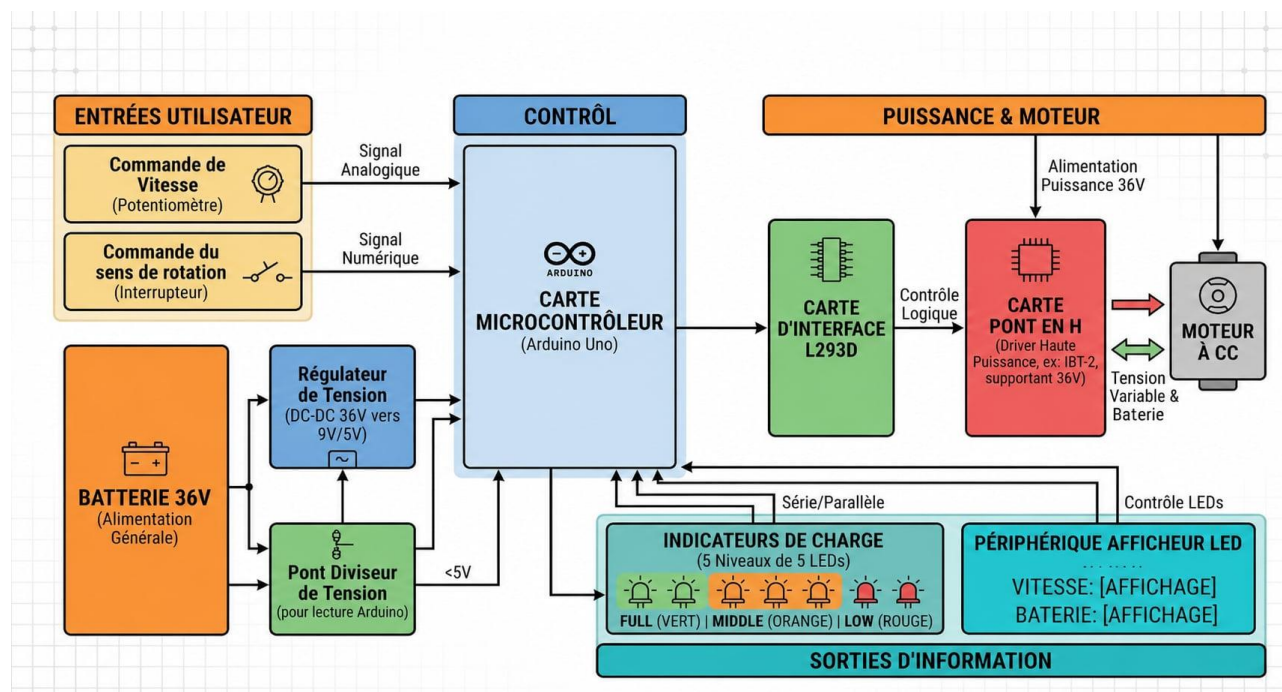


Figure 3.6 Schéma synoptique de la carte de contrôle de vitesse

### 3.2.1 Description du montage

Le système est structuré en plusieurs blocs fonctionnels suivants :

#### 3.2.1.1 Le Bloc Alimentation & Gestion de l'Énergie

- **Source d'énergie** : Tout le système est alimenté par une Batterie de 36V (Alimentation Générale).
- **Adaptation pour la logique (Régulateur)** : La tension de 36V étant bien trop élevée pour l'électronique de commande, un Régulateur de Tension (Convertisseur DC-DC) abaisse cette tension en 5V pour alimenter proprement la carte Arduino Uno.
- **Surveillance de la batterie (Pont Diviseur)** : Un Pont Diviseur de Tension est utilisé pour abaisser la tension de la batterie à une valeur inférieure à 5V. Cela permet à l'Arduino de mesurer en toute sécurité le niveau de charge restant de la batterie via une entrée analogique. [3], [4], [5], [7]

#### 3.2.1.2 Le Bloc Entrées Utilisateur

Ce bloc permet à l'opérateur d'envoyer des consignes au système :

- **Commande de Vitesse** : Un Potentiomètre envoie un signal analogique à l'Arduino (une tension variable) pour régler la vitesse du moteur.

- **Commande du sens de rotation** : Un Interrupteur transmet un Signal Numérique (Tout ou Rien) pour inverser le sens de marche (avant/arrière). [8], [23]

### 3.2.1.3 Le Bloc de Contrôle

- La Carte Microcontrôleur (Arduino Uno) reçoit les informations des entrées (potentiomètre, interrupteur) et l'état de la batterie. Elle exécute le programme interne pour calculer la vitesse théorique, le sens de rotation et gérer les affichages. [8], [15], [18]

### 3.2.1.4 Le Bloc Puissance & Commande Moteur

L'Arduino ne peut pas délivrer la puissance nécessaire pour faire tourner un moteur 36V. Ce bloc sert donc d'amplificateur de puissance :

- **Carte d'Interface L293D** : Elle sert de tampon logique pour adapter et sécuriser les signaux de commande issus de l'Arduino avant de les envoyer à la carte de puissance.
- **Pont en H** : Alimentée directement en 36V, cette carte reçoit les ordres logiques de l'interface. Grâce à sa structure en Pont en H, elle est capable de moduler la tension (souvent par PWM) pour varier la vitesse et d'inverser la polarité pour changer le sens de rotation du Moteur à CC. [3], [4], [7], [15], [21], [28], [29], [30]

### 3.2.1.5 Le Bloc Sorties d'Information (Affichage)

Ce bloc offre un retour visuel en temps réel à l'utilisateur :

- **Indicateurs de Charge** : Une barre de 3 LEDs (Vert, Orange et Rouge) indique le niveau d'énergie restant dans la batterie (Full, Middle, Low).
- **Périphérique Afficheur LED** : Un écran LCD affiche textuellement la Vitesse et l'état de charge de la batterie. [8]

### 3.3 Schéma électrique du montage

La figure suivante présente le schéma global de l'installation, structuré en étages fonctionnels allant des entrées utilisateur jusqu'à l'actionneur.

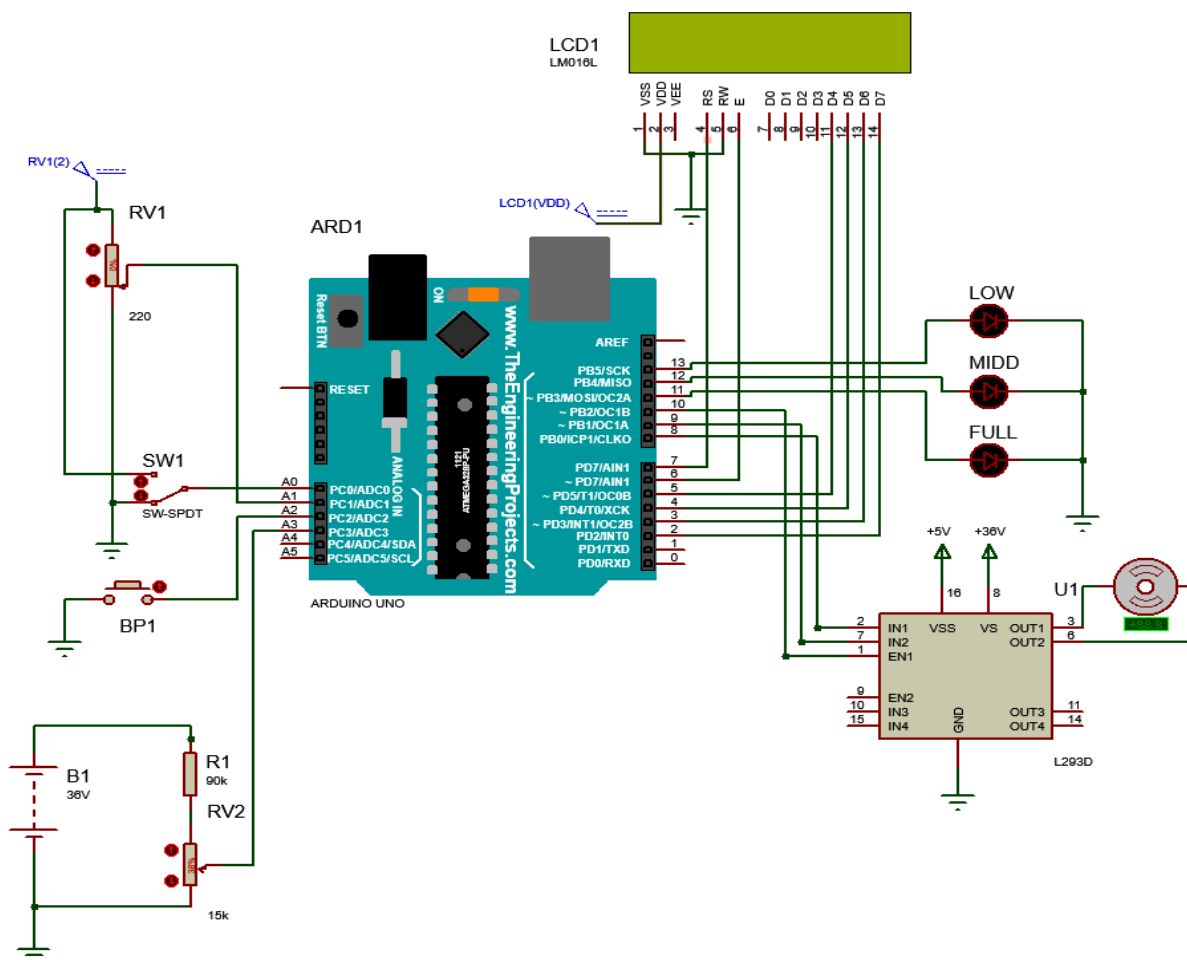


Figure 3.7 Schéma globale du système

#### 3.3.1 Description de montage

Ce montage se compose de quatre étages fonctionnels :

##### 3.3.1.1 Étage de Commande et de Traitement

Le cœur opérationnel du système est articulé autour de la carte microcontrôleur Arduino Uno (ARD1). Cet étage centralise la gestion globale en faisant office d'unité de traitement. Le microcontrôleur reçoit en continu les signaux de tension analogiques et logiques provenant du potentiomètre RV1 qui envoie une valeur de consigne à l'arduino, et un interrupteur SW1 pour commander le sens de rotation et un pont diviseur de tension qui permet de fournir une information sur l'état de charge de la batterie. Ces signaux intègrent les entrées d'arduino qui exécute l'algorithme de contrôle implanté dans sa mémoire flash, puis génère les signaux de commande appropriés. Ces signaux pilotent en temps réel les périphériques d'affichage, les indicateurs visuels d'état, ainsi que le circuit de puissance gérant vitesse et sens de rotation du moteur. [8], [15], [18]

### **3.3.1.2 Étage de Puissance et de Commande Moteur**

Les sorties logiques d'une carte Arduino étant incapables de fournir le courant nécessaire au fonctionnement d'un actionneur robuste, l'utilisation d'une interface de puissance est indispensable. Cet étage emploie le circuit intégré L293D, un double pont en H qui sépare efficacement la partie logique de commande et la partie puissance. Le driver reçoit les signaux de commande numériques et les impulsions de modulation de largeur d'impulsion (PWM) issus de l'Arduino sur ses entrées de contrôle (IN1, IN2, EN1). Il amplifie ensuite ces signaux en utilisant l'énergie depuis une ligne d'alimentation dédiée de +12V (broche VS), ce qui permet d'ajuster précisément la vitesse de rotation et le couple du moteur à courant continu U1. [3], [4], [7], [21], [28], [29], [30]

### **3.3.1.3 Étage d’Affichage LCD**

L'interface homme-machine visuelle est assurée par un afficheur alphanumérique LCD LM016L configuré sur deux lignes de seize caractères. Cet écran est interconnecté avec l'Arduino via un bus de commande (RS, E) et quatre lignes de données (D4 à D7), exploitant ainsi le mode de communication 4-bits pour économiser les broches du microcontrôleur. Son rôle est de traduire les variables internes du programme en informations textuelles claires et lisibles pour l'utilisateur, affichant en temps réel des informations utiles telles que la vitesse instantanée du moteur électrique et l'état de charge de la batterie. [8]

### **3.3.1.4 Indicateur d’État**

Complétant l'affichage textuel, cet étage fournit une signalisation visuelle instantanée grâce à trois diodes électroluminescentes (LED) connectées aux broches numériques 11, 12 et 13 de la carte. Ces indicateurs lumineux permettent au conducteur d'évaluer d'un simple coup d'œil le niveau global d'autonomie restante. La LED LOW s'active lorsque la batterie atteint un seuil critique nécessitant une recharge immédiate, la LED MIDD signale une capacité énergétique intermédiaire, et la LED FULL confirme que la batterie dispose d'une charge complète de batterie. [8]

### 3.4 Différents cas de simulations

On prend l'exemple du cas de la roue d'une trottinette de diamètre :  $D=20\text{ cm}$

#### 3.4.1 Cas de simulation faible vitesse

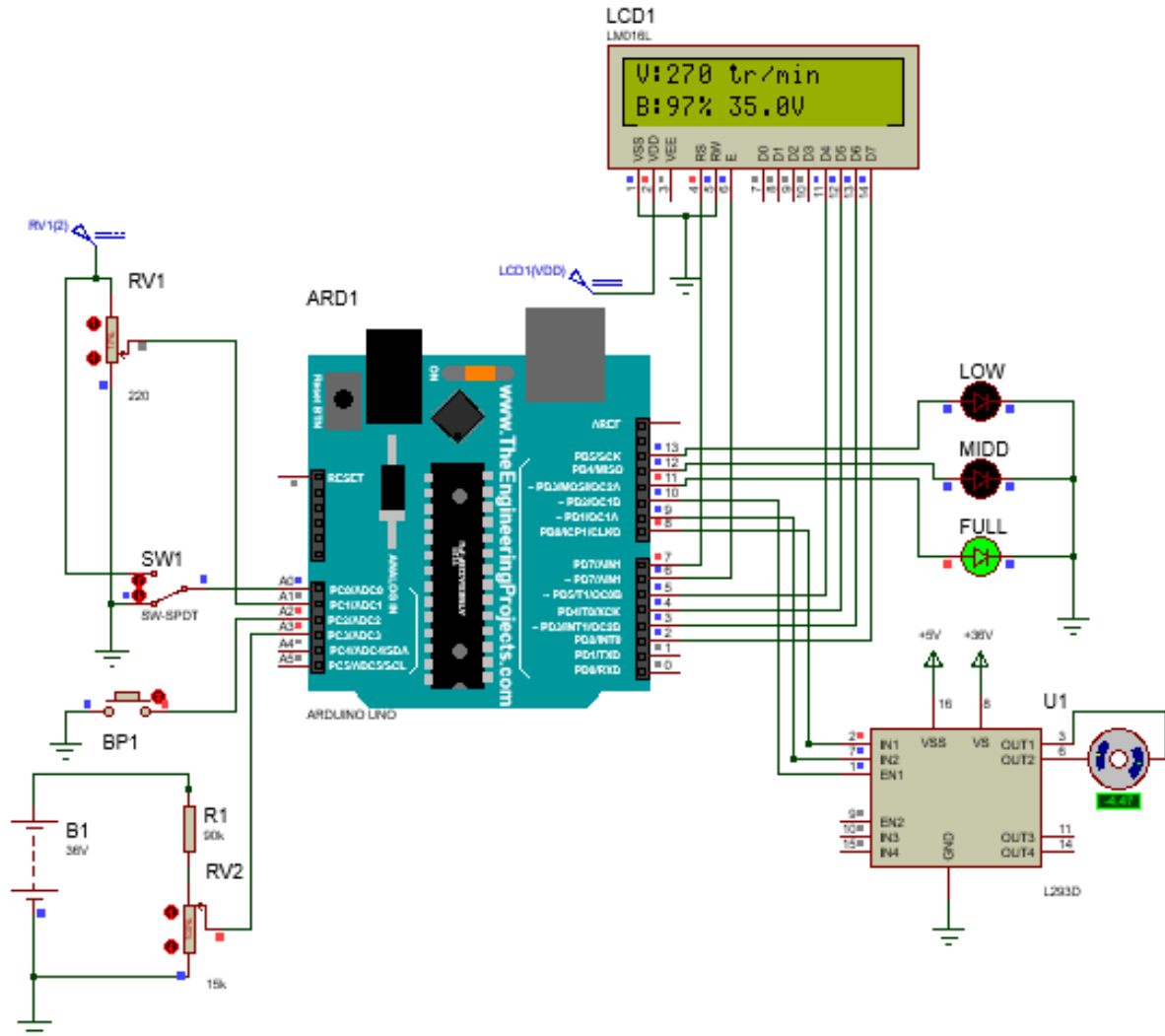


Figure 3.8 affichage de simulation en vitesse faible

La figure suivante illustre le signal PWM sortant de la pin 10 de arduino.

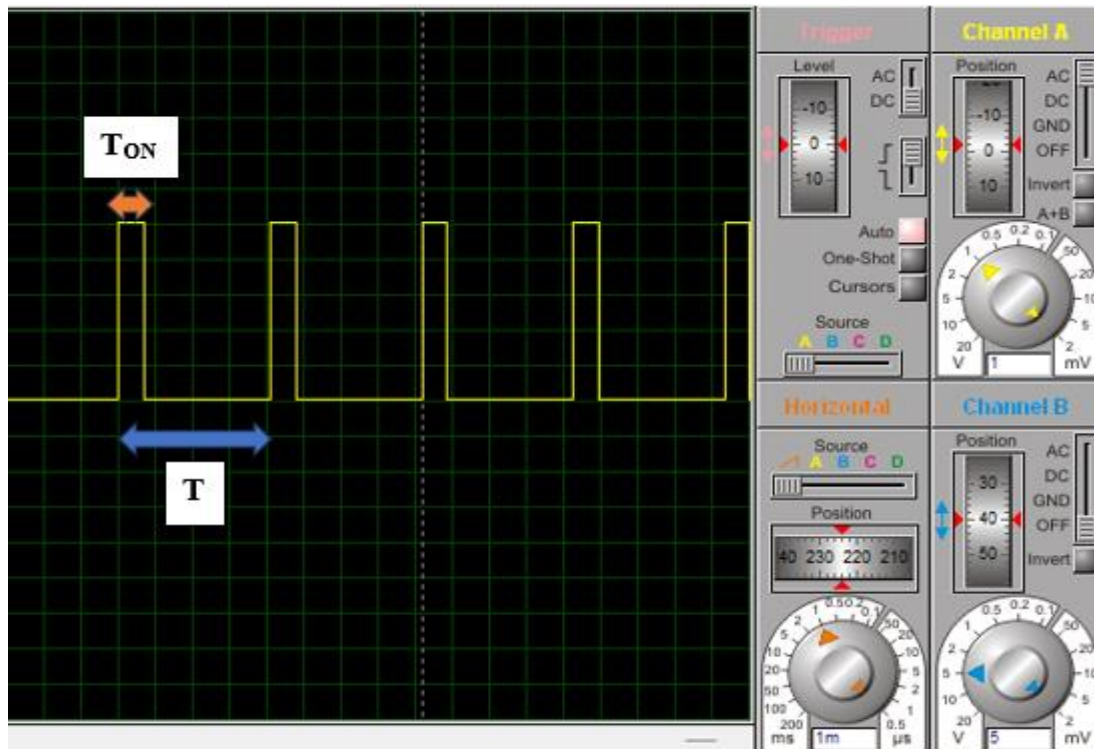


**Figure 3.9 signal PWM pour une vitesse faible**

Le calcul suivant présente la conversion simple, basée sur la valeur de 4.49tr/s relevée directement aux bornes du moteur et affichée en Tr/mn lors de la simulation

La vitesse affichée par le moteur sur Isis est donnée en tr/s alors :

$$V_{tr/min} = V_{tr/s} \times 60 \quad (3.1)$$

Application numérique :  $V_{tr/min} = 4.49 \times 60$

$$V_{tr/min} \approx 270 \text{ tr/min}$$

**La période mesurée est :  $T = 4\text{ms}$  donc  $f = 250 \text{ Hz}$**

**Le temp de fonctionnement :  $T_{ON} : 0.6 \text{ ms}$**

**Le rapport cyclique :  $\alpha = \frac{T_{on}}{T} = \frac{0.6}{4} = 15\%$**

### 3.4.2 Cas de simulation vitesse élevée

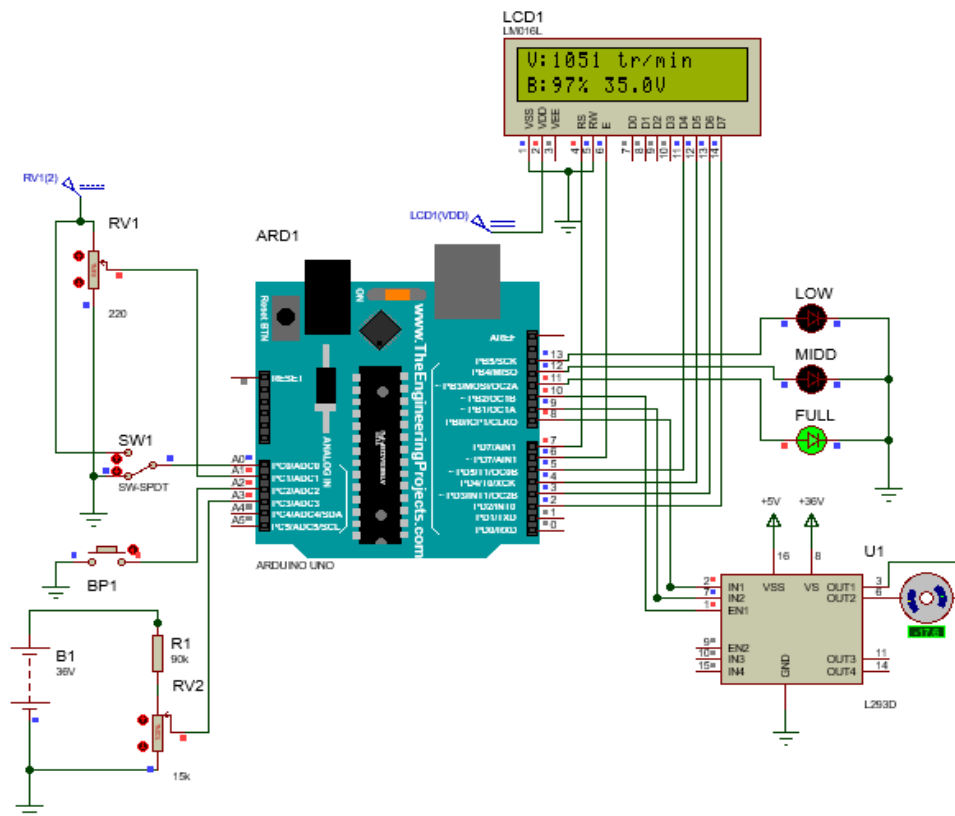


Figure 3.10 affichage de simulation en vitesse élevée

La figure suivante illustre le signal PWM sortant de la pin 10 de arduino.

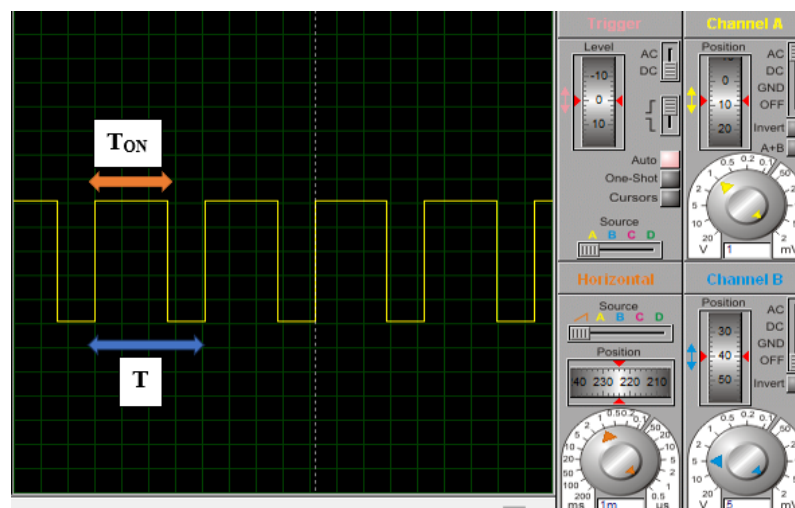


Figure 3.11 signal PWM pour une vitesse élevée

Le calcul suivant présente cette conversion simple, basée sur la valeur de 17,5tr/s relevée directement aux bornes du moteur lors de la simulation :

La vitesse affichée par le moteur en tr/mn est alors :

$$V_{tr/min} = V_{tr/s} \times 60 \quad (3.2)$$

Application numérique :  $V_{tr/min} = 17.5 \times 60 \approx 1051 \text{ tr/min}$

### 3.4.3 Simulation inversion moteur

En inversant la position de SW1 le moteur change de sens de rotation.

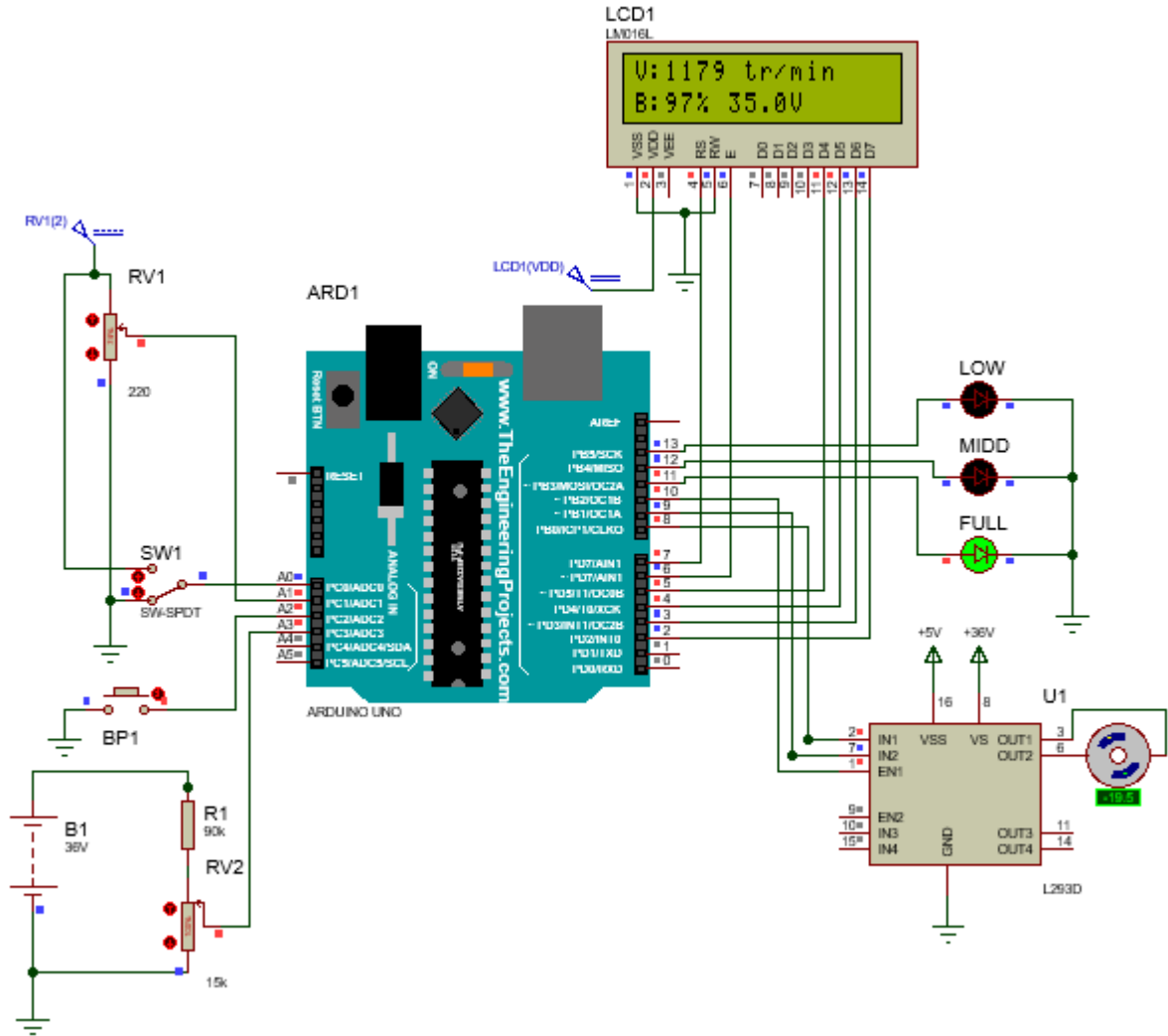


Figure 3.12 sens anti-horaire du moteur CC

### 3.4.4 Affichage de la vitesse en Km/h

On applique la formule suivante pour obtenir la vitesse en km/h:

$$V_{Km/h} = \frac{\pi \times D \times 60}{1000} \times V_{tr/min} \quad (3.3)$$

D : diamètre de la roue de la trottinette

$V_{tr/min}$  : Vitesse en tour/min

App num :

$$V_{Km/h} = \frac{\pi \times 20 \times 10^{-2} \times 60}{1000} \times 1592$$

$$V_{Km/h} \approx 60 \text{ km/h}$$

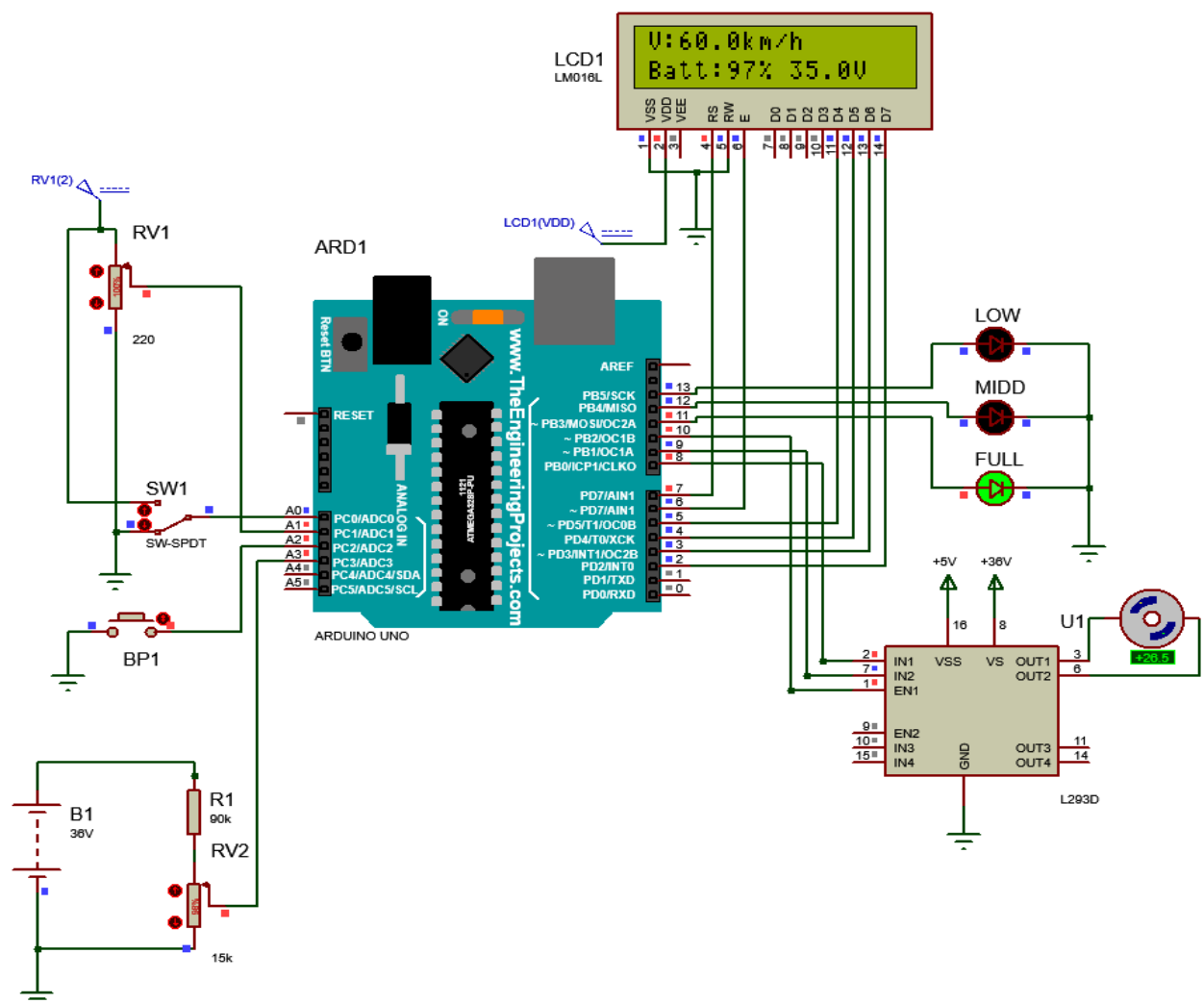


Figure 3.13 affichage de la vitesse en Km/h

### 3.4.5 Simulation affichage de l'état de charge de la batterie faible : LED rouge allumée

Afin de vérifier en simulation l'affichage des différentes LED nous avons posé les conditions suivantes :

Allumage de la LED Rouge si  $0 < V_{batt} < 10V$

Allumage de la LED Orange si  $10V < V_{batt} < 35V$

Allumage de la LED Verte si  $35V < V_{batt} < 36V$

On prend l'exemple du cas de la batterie = 15% :

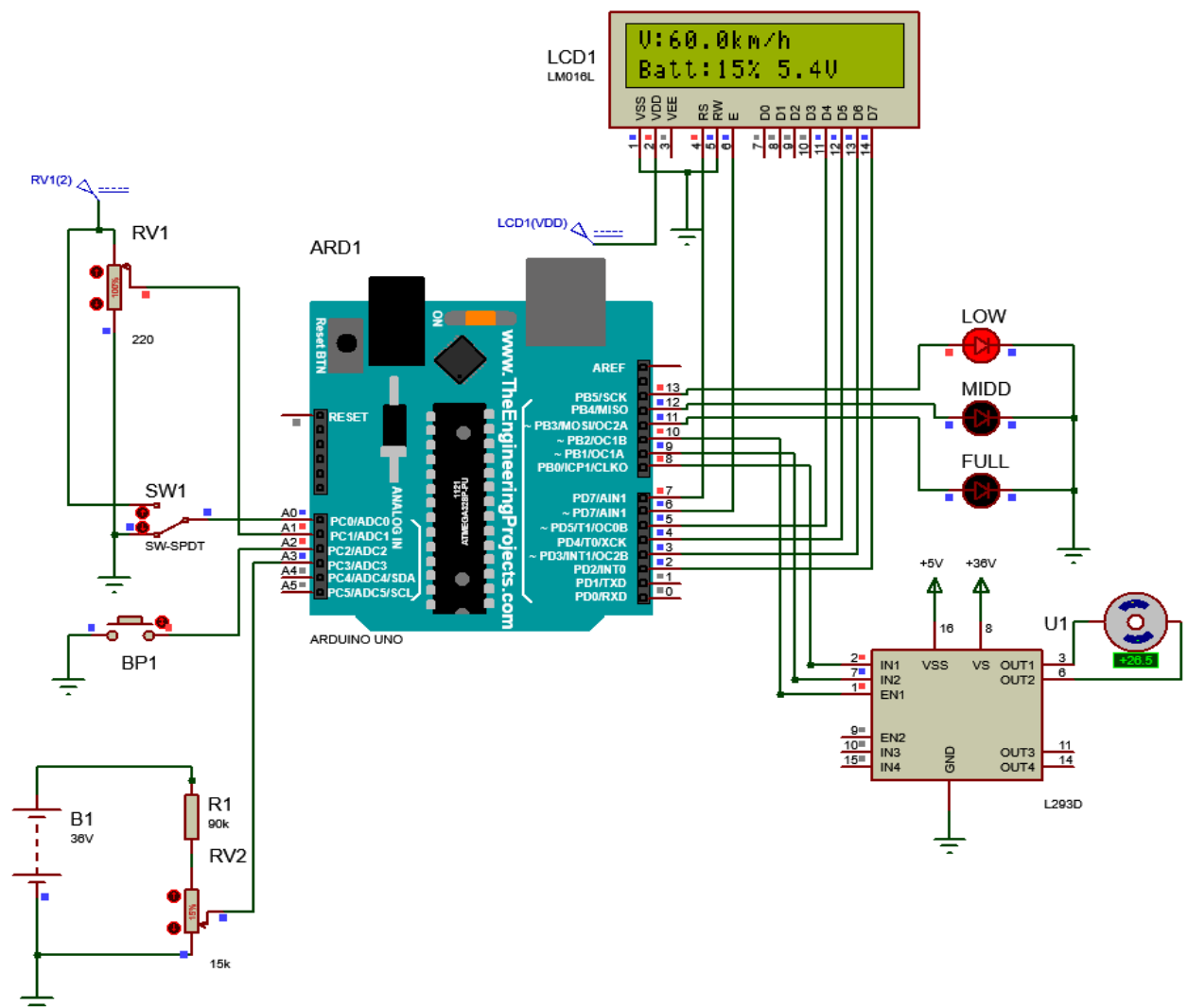


Figure 3.14 affichage de l'état de charge de la batterie : LED rouge allumée

### 3.4.6 Simulation affichage de l'état de charge de la batterie moyen : LED orange allumée

On prend l'exemple du cas de batterie = 54% :

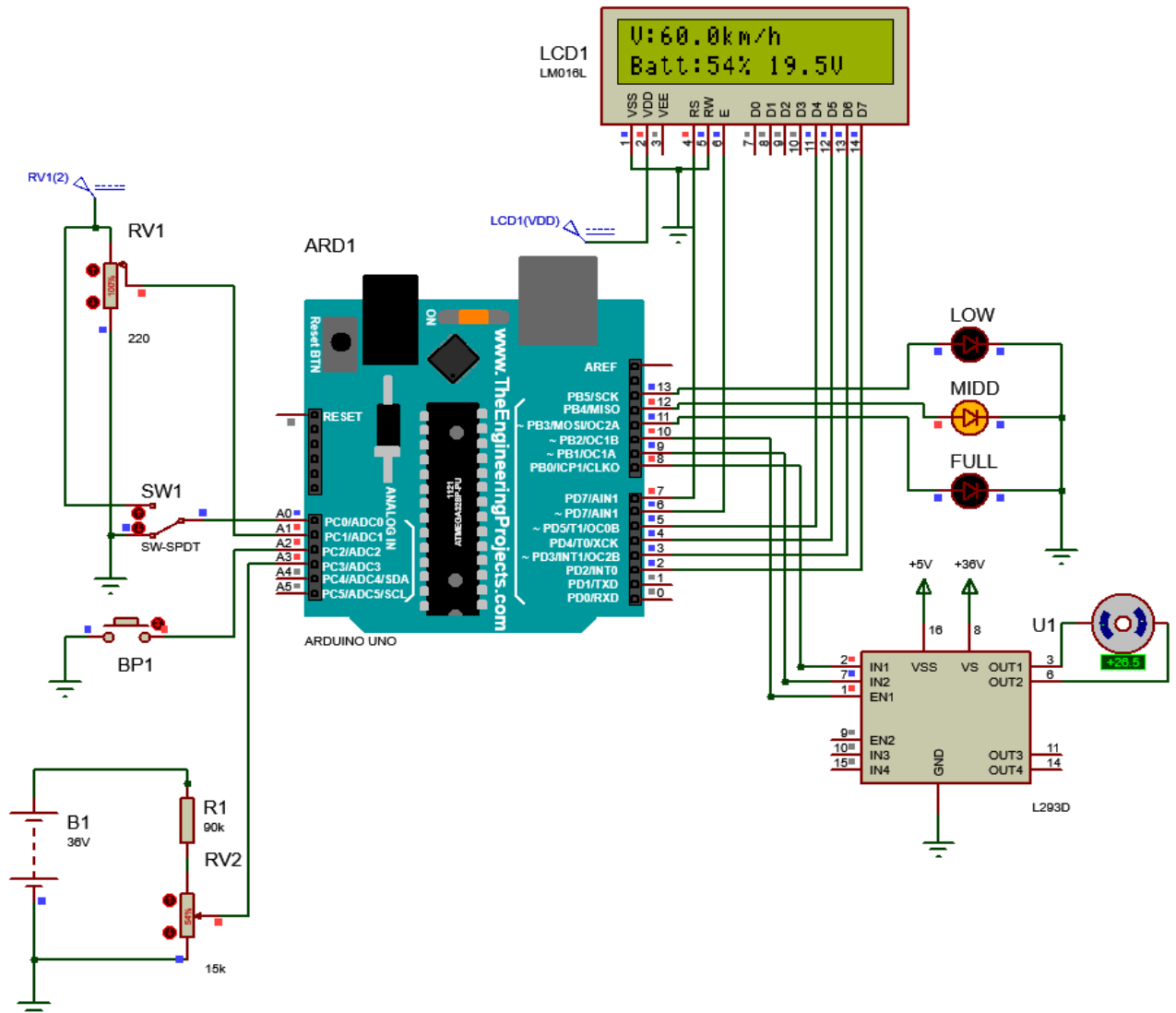


Figure 3.15 affichage de l'état de charge de la batterie : LED orange allumée

### 3.4.7 Simulation affichage de l'état de charge de la batterie pleine : LED verte allumée

On prend l'exemple du cas de batterie = 97% :

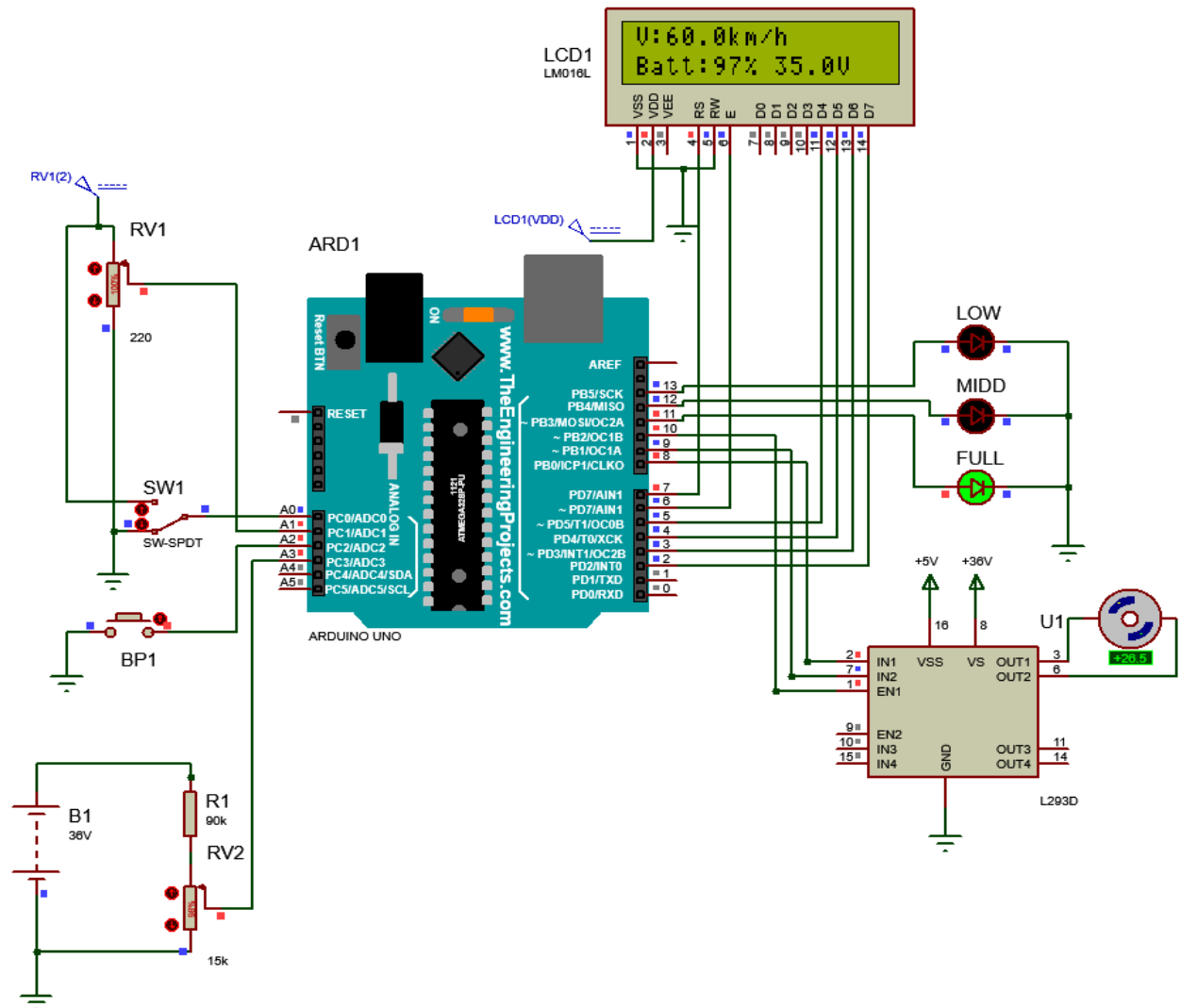


Figure 3.16 affichage de l'état de charge de la batterie : LED verte allumée

## 3.5 Développement et intégration du programme Arduino sous ISIS de Proteus

Afin d'assurer le contrôle de vitesse du moteur à courant continu, un programme a été développé sous l'environnement Arduino IDE. Ce programme permet de lire la valeur du potentiomètre, de générer un signal PWM adapté et de commander le moteur à travers le circuit L293D. Il assure également l'affichage des informations sur l'écran LCD.

Après l'écriture du programme, une compilation est réalisée dans Arduino IDE afin de générer un fichier exécutable au format HEX. Ce fichier contient les instructions qui seront exécutées par la carte Arduino.

L'intégration dans Proteus ISIS consiste à charger ce fichier HEX dans la carte Arduino virtuelle présente dans le schéma de simulation. Une fois le fichier associé à la carte, le programme peut être exécuté directement lors du lancement de la simulation.

Le principe de fonctionnement du programme est simple :

- Lecture de la position du potentiomètre ;
- Génération du signal PWM correspondant ;
- Commande du moteur via le circuit L293D ;
- Affichage de la vitesse et de l'état de la batterie sur l'écran LCD.

Cette méthode permet de reproduire sous Proteus le comportement réel du système et de vérifier son bon fonctionnement avant toute réalisation pratique. [8], [9], [15], [18], [19]

### 3.5.1 Exemple de programme IDE

On prend l'exemple de l'affichage des 3 LED et on cite les principales étapes :

#### 3.5.1.1 Étape 1 : Le choix du type de carte Arduino se fait en fonction de

- **La Complexité du projet** → simple (Uno), complexe (Mega, ESP32).
- **Le Nombre d'entrées/sorties** → peu (Uno/Nano), beaucoup (Mega).
- **La Connectivité** → besoin Wi-Fi/Bluetooth (ESP32).
- **La Taille & budget** → compact (Nano), économique (Uno).

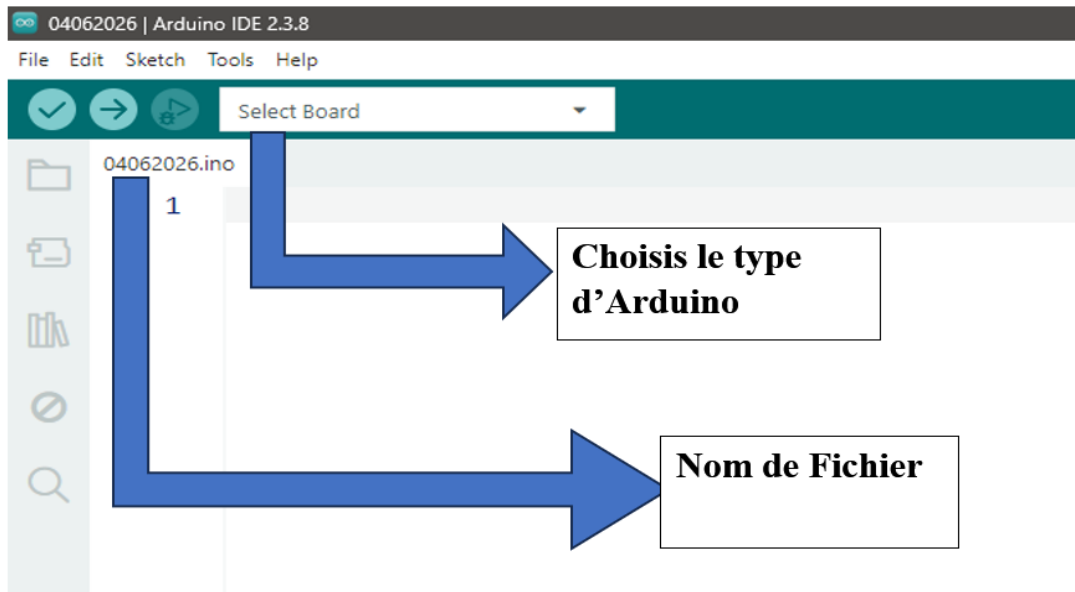


Figure 3.17 étape 1 : choix du type de carte Arduino

### 3.5.1.2 Etape 2 : écriture du programme IDE

L'Arduino lit la tension de la batterie grâce à une entrée analogique. Cette valeur est comparée à des seuils prédéfinis. Selon le résultat, une LED s'allume.

- Verte → batterie en bon état.
- Orange → batterie moyenne.
- Rouge → batterie faible.

```
// ----- LEDS BATTERIE -----

if (tensionReelle >= 35)
{
  digitalWrite(LED_GREEN, HIGH);
  digitalWrite(LED_ORANGE, LOW);
  digitalWrite(LED_RED, LOW);
}
else if (tensionReelle > 9.0)
{
  digitalWrite(LED_GREEN, LOW);
  digitalWrite(LED_ORANGE, HIGH);
  digitalWrite(LED_RED, LOW);
}
```

Figure 3.18 étape 2 : écriture du programme d'Arduino

### 3.5.1.3 Etape 3 : Vérification et correction du programme

Dans cette étape :

Compiler le code → le programme est traduit en langage machine pour vérifier qu'il est correct.

Détecter les erreurs → s'il y a une faute de syntaxe ou une mauvaise instruction, l'IDE le signale.

Valider la logique → si tout est bon, on obtient un message "Compilation terminée avec succès".

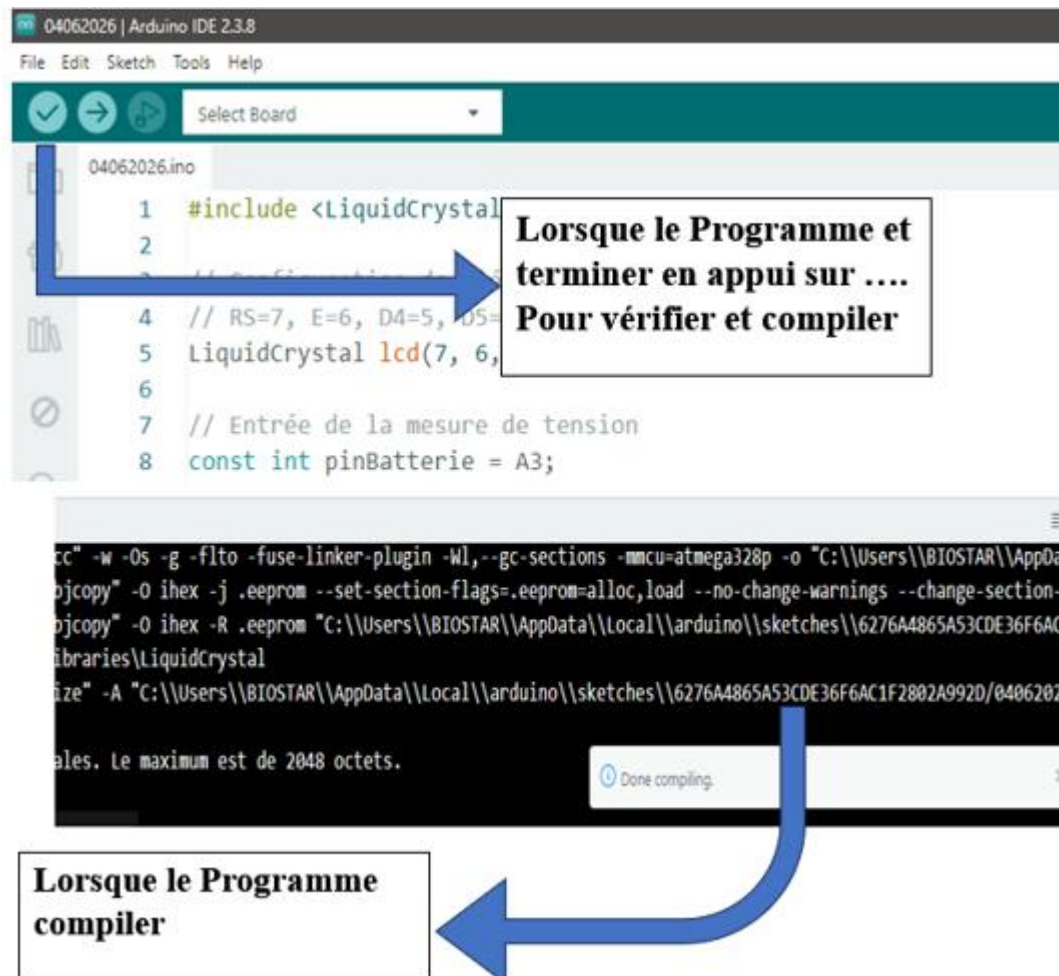


Figure 3.19 étape 3 : vérification et compilation du code d'Arduino

### 3.5.1.4 Etape 4 : Importation du Fichier HEX

Après la Compilation de code Arduino dans l'IDE (bouton "check mark" puis "upload"). Cela génère automatiquement un fichier ".hex".

Après on ouvre ISIS Proteus, double-cliquer sur l'Arduino dans le montage pour ouvrir ses **propriétés**.

Dans la case "Program File", sélectionner et importer le fichier ".hex" généré par l'IDE Arduino.

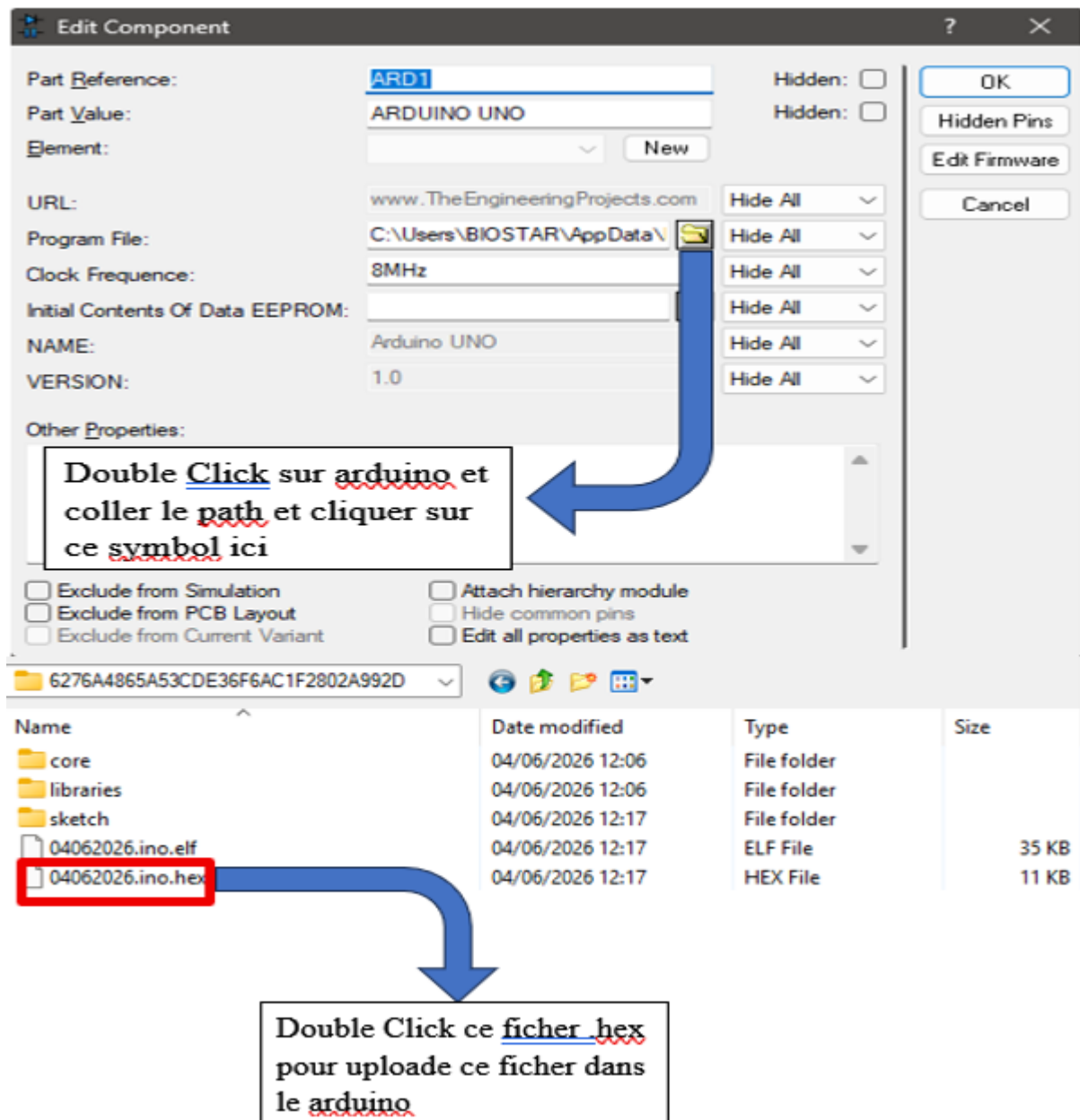


Figure 3.20 étape 4 : importation de fichier HEX vers le montage ISIS

### 3.5.1.5 Etape 5 : Validation et simulation

Finalement on valide puis on lance la simulation. L'Arduino dans Proteus exécutera le programme contenu dans le fichier HEX.

La simulation du montage dans Proteus ISIS consiste à tester virtuellement le circuit avant de le réaliser sur un circuit imprimé.

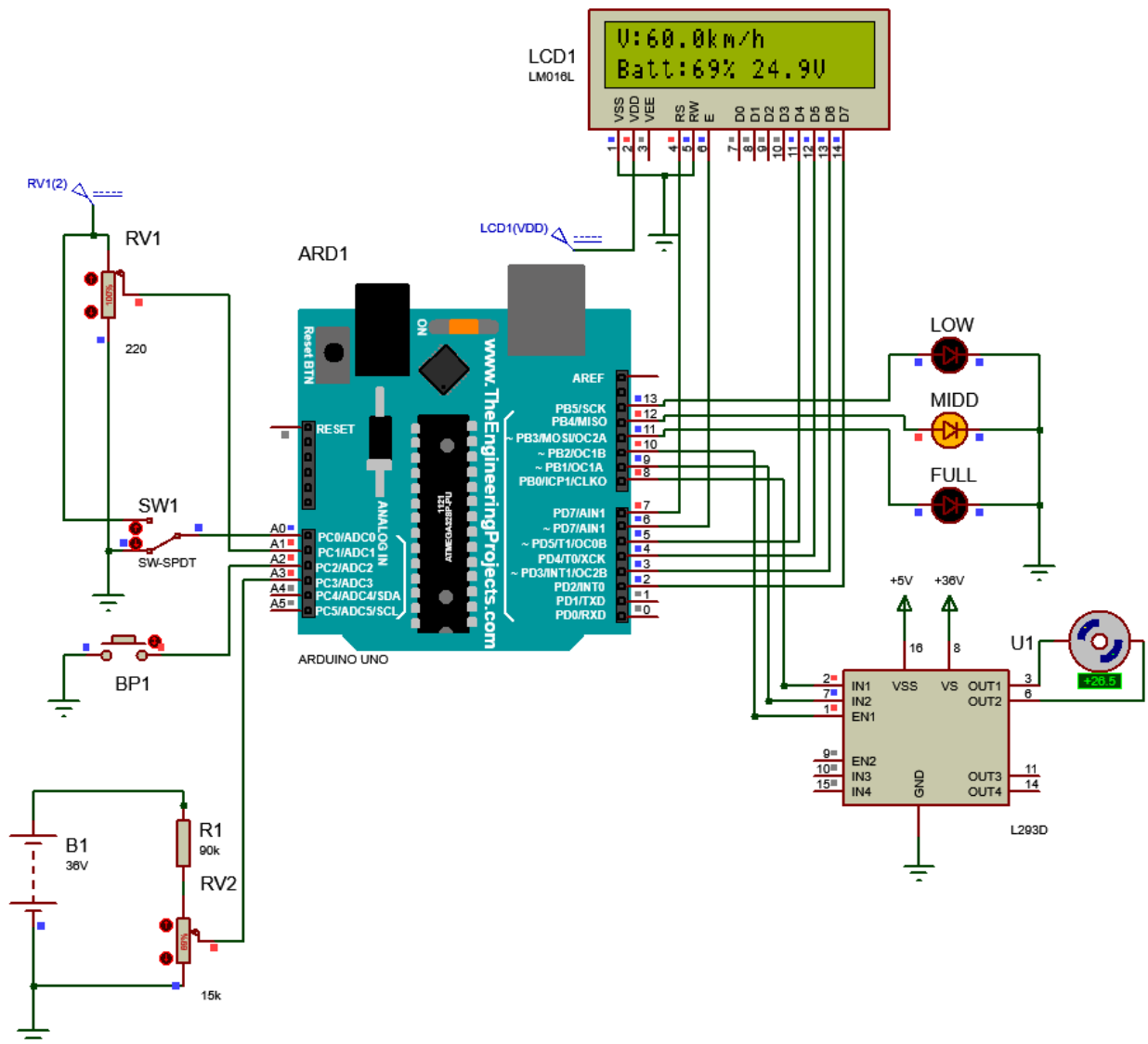


Figure 3.21 étape 5 : simulation du montage

### 3.5 Validation des résultats

Après la réalisation du schéma électronique et l'intégration du programme Arduino dans l'environnement Proteus ISIS, plusieurs simulations ont été effectuées afin de vérifier le bon fonctionnement du système proposé. L'objectif principal de cette phase consiste à valider le comportement global de la carte électronique.

La première étape de validation a consisté à observer la réaction du moteur à courant continu lors de la variation de la position du potentiomètre. Les résultats obtenus montrent que toute modification de la consigne entraîne une variation correspondante du rapport cyclique du signal PWM généré par la carte Arduino Uno. Cette variation du signal de commande se traduit directement par une modification de la tension moyenne appliquée au moteur via le circuit L293D.

Lorsque le potentiomètre est positionné à sa valeur minimale, le rapport cyclique du signal PWM est faible, ce qui entraîne une vitesse de rotation réduite du moteur. À mesure que la position du potentiomètre augmente, le rapport cyclique augmente progressivement et provoque une accélération du moteur. Lorsque le potentiomètre atteint sa valeur maximale, le moteur fonctionne à sa vitesse maximale prévue par le système.

Les observations effectuées au cours de la simulation montrent également que la variation de vitesse est progressive et stable. Aucun comportement instable ou variation brusque n'a été observé, ce qui confirme la bonne adaptation entre la partie commande assurée par l'Arduino Uno et la partie puissance réalisée à l'aide du circuit L293D.

Les informations relatives à la vitesse du moteur ainsi qu'à l'état de la batterie sont correctement affichées et mises à jour en temps réel par l'afficheur LCD 16×2. Cette fonctionnalité améliore l'interaction entre l'utilisateur et le système en permettant une surveillance continue des paramètres de fonctionnement.

Par ailleurs, la simulation a permis de vérifier le bon fonctionnement de l'ensemble de la chaîne de commande, depuis la lecture de la consigne jusqu'à l'actionnement du moteur.

Les résultats de simulation confirment ainsi que la solution proposée répond aux objectifs fixés au début du projet. La carte électronique conçue permet un contrôle simple, efficace et fiable de la vitesse du moteur à courant continu tout en assurant une visualisation des informations essentielles de fonctionnement. [7], [8], [9], [15], [18], [19], [21], [23]

## Conclusion

Dans ce chapitre, nous avons procédé à la conception et à la simulation complète de la carte électronique destinée au contrôle de vitesse d'un moteur à courant continu. Après l'élaboration du schéma synoptique mettant en évidence les différents blocs fonctionnels du système, nous avons réalisé le schéma électrique sous ISIS Proteus en intégrant la carte Arduino Uno, le circuit de puissance L293D, le moteur à courant continu, la batterie d'alimentation, le potentiomètre de commande ainsi que l'afficheur LCD.

Les différentes simulations réalisées ont permis de vérifier le bon fonctionnement de la chaîne de commande. Les résultats obtenus montrent que la variation de la consigne entraîne une modification progressive de la vitesse du moteur, tandis que l'afficheur assure correctement l'affichage des informations relatives à la vitesse et à l'état de la batterie. [7], [8], [9], [15], [18], [19]

# CONCLUSION GÉNÉRALE

Le présent travail de fin d'études a été consacré à l'étude et à la simulation d'une carte électronique destinée au contrôle de vitesse d'un moteur à courant continu à l'aide de la carte Arduino Uno et du logiciel ISIS de Proteus. L'objectif principal de ce projet était d'étudier et de proposer une solution simple, économique et performante permettant de faire varier progressivement la vitesse de rotation du moteur en utilisant une commande manuelle basée sur un potentiomètre et une modulation de largeur d'impulsion PWM.

Dans un premier temps, une étude théorique du moteur à courant continu a été réalisée pour comprendre son principe de fonctionnement. Il a été mis en évidence que la vitesse du moteur dépend directement de la tension moyenne appliquée à l'induit, ce qui justifie pleinement l'utilisation d'une commande électronique adaptée.

Dans un second temps, l'étude s'est orientée vers l'analyse des composants électroniques nécessaires à la réalisation de la carte de commande. La carte Arduino Uno a été choisie comme unité de traitement numérique grâce à sa capacité à lire une tension analogique et à générer des signaux PWM précis. Le circuit intégré L293D a été introduit comme interface de puissance capable d'assurer l'adaptation énergétique entre le microcontrôleur et le moteur. L'ensemble de ces composants, associé au principe du hacheur et du pont en H, et incluant l'afficheur LCD a permis de constituer une architecture cohérente de contrôle.

Dans un troisième temps, la phase de conception de simulation a été effectuée sous ISIS Proteus. Le schéma électronique global a été élaboré puis testé simulé sous ISIS. Les différentes simulations ont montré que la variation de la position du potentiomètre entraîne une modification progressive du rapport cyclique du signal PWM, ce qui agit directement sur la tension moyenne appliquée au moteur et permet ainsi de contrôler efficacement sa vitesse et son sens de rotation.

Comme perspectives futures, ce travail pourrait être approfondi par :

- La réalisation pratique de la carte électronique sur support réel ;
- Intégration d'un capteur de température pour contrôler l'état de chauffe du moteur.

En définitive, ce projet nous a permis de mettre en application plusieurs connaissances acquises durant notre formation, notamment dans les domaines de l'électronique de puissance, de la programmation embarquée, de la modélisation électromécanique et de la simulation assistée par ordinateur. Il constitue ainsi une expérience enrichissante ouvrant la voie à des développements plus avancés dans le domaine de la commande électronique des systèmes motorisés.

# Bibliographie

- [1] Chauprade, R., *Commande électronique des moteurs à courant continu*, Éditions Eyrolles, France, 1978.
- [2] Kostenko, M., Piotrovski, L., *Machines électriques : Machines à courant continu et transformateurs*, Tome 1, Éditions MIR, Moscou, 1979.
- [3] Séguier, G., *Électronique de puissance : Convertisseurs continus*, Dunod, Paris.
- [4] Séguier, G., Labrique, F., *Électronique de puissance : Convertisseurs directs*, Dunod, 2014.
- [5] Laroche, J., *Électronique de puissance : Convertisseurs, cours et exercices corrigés*, Dunod, France, 2005.
- [6] Pinard, M., *Commande électronique des machines électriques*, Dunod, France, 2004.
- [7] STMicroelectronics, *L293 Dual Full Bridge Driver Datasheet*, Technical Documentation, 2024.
- [8] Arduino, *Arduino Uno Official Documentation*.
- [9] Labcenter Electronics, *Proteus Design Suite Professional Manual*, Version 8 Professional Documentation.
- [10] Boulakroune, S., *Commande à vitesse variable d'un moteur à courant continu alimenté par hacheur bidirectionnel en courant*, Mémoire de Master, Université Badji Mokhtar, Annaba, 2018.
- [11] Grar, M., Yousfi, I., *Variateur de vitesse de deux moteurs à courant continu par PIC16F876 en MLI*, Mémoire de Master, Université Aboubakr Belkaïd, Tlemcen, 2014.
- [12] Manseri, S., *Commande adaptative indirecte d'un moteur à courant continu*, Mémoire de fin d'études, Université A. Mira, Béjaïa, 2012.
- [13] Dich B., Cherigui N., *Etude d'une Commande d'un moteur à courant continu* Mémoire de fin d'études, ESSAT, Tlemcen 2020.
- [14] Chiha, Y., *Commande à vitesse variable d'un moteur à courant continu*, Mémoire de fin d'études, Université Larbi Ben M'hidi, Oum El Bouaghi, 2017.
- [15] Djalal, D., Belkacem, B., Babziz, M. N., *Speed and Position Control of DC Motor by Arduino*, Master Thesis, Kasdi Merbah University, Ouargla, 2023.
- [16] Douis, B., Gadi, N., *Commande de vitesse d'un moteur à courant continu par réseau de neurones artificiels avec une carte Arduino*, Mémoire de Master, Université d'El Oued, 2019.
- [17] Bouras, M. H., Ghalem, A., *Amélioration de l'implémentation de la commande intelligente d'un servomoteur à courant continu à base d'une carte Arduino*, Mémoire de Master, Université Akli Mohand Oulhadj, Bouira, 2017.

- [18] Tesfaye Hordofa Kaffale, Jia Liu, “Speed and Direction Control of DC Motor Using Arduino Uno Microcontroller”, *Open Access Library Journal*, Vol. 12, 2025.
- [19] Naima Rahoua, Hani Benguesmia, Meriem Elhamami et al., “Simulation and Implementation of BLDC Motor Control with Arduino Mega in Proteus”, *ITEGAM Journal of Engineering and Technology for Industrial Applications*, Vol. 11, No. 56, 2025.
- [20] Alfian Ma’arif, Iswanto, Nia Maharani Raharja et al., “Control of DC Motor Using Proportional Integral Derivative (PID): Arduino Hardware Implementation”, *International Conference on Industrial Electrical and Electronics*, 2020.
- [21] Aiman Mahar, Pirah Peerzada, Wasi Hyder Larik, “DC Motor Speed Control Through Arduino and L298N Motor Driver Using PID Controller”, *International Journal of Electrical Engineering & Emerging Technology*, 2022.
- [22] Prince Ogbodum, Sampson Bendor, “Dynamic Performance and Microcontroller-Based Control of Permanent Magnet Direct Current Motors”, *World Academics Journal of Engineering Sciences*, Vol. 12, No. 3, 2025.
- [23] Maman Rukmana, “DC Motor Speed Control with Potentiometer Based on Arduino Uno”, *Journal of Renewable Energy, Electronics and Control (JREEC)*, 2023.
- [24] Bryan Hidayat Soelaiman, *Pengendalian Motor DC Menggunakan Pulse Width Modulation (PWM) Berbasis Arduino Uno*, Faculty of Engineering Thesis, University of Jember, 2015.
- [25] Kerboua, A., *Cours d’Électrotechnique : Machine à Courant Continu*, Support pédagogique.
- [26] Hordofa Kaffale, T., Liu, J., “Speed and Direction Control of DC Motor Using Arduino Uno Microcontroller”, 2025.
- [27] Chapitre Machine à Courant Continu : <https://www.scribd.com/document/578036083/Chapitre-3-Machine-a-courant-continu-4>
- [28] L293D Présentation et fonctionnement : <https://www.rs-online.com/designspark/basics-of-l293d>
- [29] L293D Pinout and Features: <https://components101.com/ics/l293d-pinout-features-datasheet>
- [30] L293D Pin Description and Working: <https://www.engineersgarage.com/l293d-pin-description-and-working/>
- [31] Loi de Laplace appliquée au moteur à courant continu : <https://premiere-scientifique.simulatus.info/comprendre/loilaplacemcc.pdf>
- [32] Les différents types de convertisseurs statiques : <https://studylibfr.com/doc/5519406/les-diff%C3%A9rents-types-des-convertisseurs-statiques>