



République Algérienne Démocratique et Populaire
Ministère de L'enseignement Supérieur et de la Recherche Scientifique

Université Echahid Hamma Lakhdar -El Oued-

Faculté de la Technologie

Département de Génie mécanique

MÉMOIRE

Présenté en vue de l'obtention du diplôme de

MASTER ACADEMIQUE

Spécialité : Électromécanique

Présenté par :

- MESSAOUDI Khaled
- BERIBECHE Laiche
- TERKI Mohammed Elyazid

Intitulé :

**Conception Et Simulation D'un Système De Commande
Électrique Pour Un Ascenseur Électromécanique**

Soutenue le : 29 / 05 /2025

Devant le jury composé de :

Dr. ZEGHDI Zoubir.

Président

Dr. REHOUMA Youssef.

Examineur

Mr. GUIA Hossam Eddine.

Encadreur

Année Académique : 2024/2025

Résumé

Ce mémoire présente la conception et la simulation d'un système d'ascenseur électrique à trois étages utilisant un microcontrôleur Arduino, pour des applications spécifiques. L'objectif de ce projet est de fournir un moyen de transport pratique et accessible aux personnes en à mobilité réduite, en particulier celles ayant des difficultés à utiliser les escaliers traditionnels en raison de limitations physiques ou de problèmes de mobilité. Le système est composé d'un microcontrôleur Arduino, d'un clavier numérique, d'un pilote de moteur et d'une cabine d'ascenseur avec un mécanisme de levage. Le microcontrôleur reçoit les commandes de l'utilisateur et les traite pour déterminer l'étage souhaité. Le pilote de moteur active ensuite le mécanisme de levage pour déplacer la cabine vers l'étage sélectionné. Le système a été testé et s'est avéré fiable et efficace pour transporter les individus à l'étage désiré. Il a le potentiel d'améliorer considérablement l'accessibilité et la commodité des bâtiments pour les personnes à mobilité réduite.

Mots clés : Ascenseur Electromécanique, Proteus, Arduino Uno, Commande numérique.

Abstract

This thesis presents the design and the simulation of a three-story electric elevator system using an Arduino microcontroller for specific applications. The objective of this project is to provide a practical and accessible means of transportation for individuals with reduced mobility—particularly those who encounter difficulty using traditional staircases due to physical limitations or mobility impairments.

The system comprises an Arduino microcontroller, a numeric keypad, a motor driver, and an elevator cabin equipped with a lifting mechanism. The microcontroller receives user commands and processes them to determine the desired floor; the motor driver then activates the lifting mechanism to move the cabin to the selected level. The system has been tested and proven reliable and efficient in transporting users to their requested floor, and it holds significant potential to enhance building accessibility and convenience for persons with reduced mobility.

Keywords : Electromechanical Elevator, Proteus, Arduino Uno, Digital Control.

ملخص

تقدم هذه المذكرة تصميم ومحاكات نظام مصعد كهربائي مكوّن من ثلاثة طوابق باستخدام متحكم أردو وينو لأغراض تطبيقية محدّدة. يهدف هذا المشروع إلى توفير وسيلة نقل عملية وميسّرة للأشخاص ذوي الإعاقة الحركية، لا سيما الذين يعانون صعوبات في استخدام السلالم التقليدية بسبب قيود جسدية أو مشاكل في الحركة. يتكوّن النظام من متحكم Arduino، ولوحة مفاتيح رقمية، ومُشغّل محرك، وكابينة مصعد مزوّدة بآلية رفع. يستقبل المتحكم أوامر المستخدم ويعالجها لتحديد الطابق المطلوب، ثم يفعل مُشغّل المحرك آلية الرفع لنقل الكابينة إلى الطابق المختار. لقد خضع النظام للاختبار وأثبت موثوقيته وكفاءته في نقل الأفراد إلى الطابق المرغوب. وتمتاز هذه الحلول بإمكاناتها الكبيرة في تحسين سهولة الوصول وراحة استخدام المباني للأشخاص ذوي الإعاقة الحركية.

الكلمات المفتاحية: مصعد كهروميكانيكي، برنامج: **Proteus**، المتحكم أردو وينو أونو، تحكم رقمي.

Remerciement

Avant tout, nous remercions ALLAH le tout puissant de nous avoir donné le courage, la volonté pour accomplir ce travail.

Nos vifs remerciements vont en premier lieu, à nos chers parents et familles de nous avoir aidé, encouragées et soutenus tout au long de ces années.

Comme nous tenons à exprimer notre profonde gratitude à notre encadrant Mr. GUIA Hossam Eddine qui a cru en nous, qui nous a soutenus et orienter tout au long de notre travail.

Nous remercions aussi, les membres de jury qui nous font l'honneur de juger notre travail.
Nous espérons être à la hauteur de leurs attentes.

Enfin, nous adressons nos remerciements à tous nos amis et collègues surtout ceux qui nous apporté un soutien moral.

Dédicace

Au nom de Dieu, le Tout Miséricordieux, le Très Miséricordieux.

Louange à Dieu, le Très Haut, qui m'a accordé le courage et la persévérance nécessaires pour mener à bien ce travail.

J'offre ce modeste travail :

- **À ma chère mère**

qui a été mon pilier et mon inspiratrice tout au long de mon parcours académique, me prodiguant tendresse et conseils précieux depuis mon plus jeune âge. Maman, merci pour tout l'amour et le soutien que tu m'as offerts ; je prie Dieu de te préserver et de t'accorder santé et longue vie.

- **À la mémoire de mon cher père**

dont l'empreinte forte a marqué ma vie et qui fut le plus grand soutien que j'aie jamais connu. À chaque évocation de ton souvenir, les larmes coulent, et sans toi, je ne serais pas ce que je suis aujourd'hui. Qu'Allah te couvre de Sa miséricorde et t'accueille dans Son vaste paradis.

- **À mes frères, mes sœurs et à ma chère épouse**

qui m'ont encouragé et soutenu à chaque étape de ce projet, faisant de leurs mots un moteur et une source constante de motivation.

- **À mon oncle et frère aîné, Mohammedi Abderrahmane**

qui, par sa bienveillance et son soutien, a joué pour moi le rôle de père et d'ami. Merci pour tout ce que tu m'as apporté ; je demande à Allah de bénir ton existence, tes ressources et ta famille.

- **À mes proches, mes amis et à tous ceux qui m'ont soutenu**

je vous exprime ma profonde gratitude pour votre appui et vos encouragements sans relâche ; sans vous, cette aventure n'aurait pas connu un tel succès.

Je prie le Dieu Tout-Puissant que ce travail soit mené avec sincérité et qu'il soit utile à quiconque le lira.

Et. Messaoudi Khaled.

Dédicace

À la mémoire de mon cher père, paix à son âme, qui m'a appris la patience et l'effort.

À ma tendre mère, source d'amour et de prières, que Dieu te récompense pour tout.

À mes frères, mes amis, et à tous ceux qui m'ont soutenu de près ou de loin.

À mon encadrant Mr. GUIA Hossam Eddine, pour ses conseils avisés et son accompagnement précieux.

À l'ensemble des enseignants et cadres de l'université, en reconnaissance de leurs efforts et de
leur savoir.

Je vous dédie ce modeste travail en signe de gratitude et de reconnaissance.

Et. BERIBECHE Laiche.

Dédicace

Tout d'abord, je remercie le Dieu Tout-Puissant qui m'a donné le courage et la patience pour mener à bien ce travail.

J'offre ce modeste ouvrage à :

Mes chers parents : pour leurs sacrifices et leur soutien indéfectible ; que Dieu les préserve pour moi.

Mes chers frères et sœurs.

Tous mes amis et tous ceux qui m'ont soutenu.

Et. TERKI Mohammed Elyazid.

Sommaire :

Liste des figures.....	I
Liste des tableaux.....	II
Introduction générale.....	A-B

Chapitre I : Généralité sur les ascenseurs.

I.1. Introduction	04
I.2. Généralité sur Les Ascenseurs.....	04
I.2.1. Définition des ascenseurs.....	04
I.2.2. Importance Dans La Vie Quotidienne.....	04
I.2.3. Différent Types D'ascenseu.....	05
I.2.4. Familles D'ascenseur.....	08
I.2.5. Classification Des Ascenseurs Selon Le Mécanisme De Levage.....	09
I.2.6. Composants Des Systèmes De L'ascenseur.....	10
I.2.6.1. Composants mécaniques d'un ascenseur.....	10
I.2.6.2. Composants électriques d'un ascenseur	16
I.2.6.2.1. Moteur Electrique.....	16
I.2.6.2.2. Variateurs de vitesse.....	21
I.2.6.2.3. Capteurs	26
I.2.6.2.4. Tableau de commande et armoires électriques.....	27
I.2.6.3. Automate Programmable Industriel (API)	28
I.3. Conclusion.....	29

Chapitre II : Cahier de charge

II.1. Introduction	31
II.2. Cahier Des Charges	32
II.3. Comparaison entre différentes solutions disponibles.....	33
II.3.1. Analyse comparative	34
II.4. Avantages techniques et économiques de l'Arduino UNO.....	34
II.5. Justification du choix de l'Arduino UNO.....	35
II.6. Architecture d'un système à base de microprocesseur	36
6.1. Définition d'un Microprocesseur.....	37
6.1.1. Architecture d'un Microprocesseur CPU.....	37
6.2. Définition d'un Microcontrôleur.....	38
6.2.1. Structure interne d'un microcontrôleur.....	38
6.2.2. Types des boîtiers de microcontrôleurs.....	40

6.3. Structure des composants de système.....	40
II.7. Analyse structurelle des systèmes.....	41
II.7.1. Chaîne d'information.....	41
II.7.2. Chaîne d'énergie.....	42
II.7.3. Mise en situation.....	42
II.7.4. Démarche de conception.....	43
II.8. Conclusion	43

Chapitre III : Simulation & Programmation

III.1 Introduction.....	45
III.2 Partie Logiciel “Software”.....	45
III.2.1. Présentation de Proteus.....	45
III.2.2. Environnement de la programmation (Logiciel IDE)	47
III.2.2.1. Description.....	47
III.2.2.2. Langage de programmation.....	49
III.3. Architecture fonctionnelle de la carte de commande.....	49
III.3.1. Définition précise du contrôleur Arduino Uno.....	49
III.3.2. Description des différentes parties de la carte de Commande.....	49
III.3.2.1. Choix du microcontrôleur « Arduino Uno ».....	49
III.3.2.2. Broches de la carte Arduino Uno	50
III.3.2.3. Caractéristique de Base du Arduino Uno.....	52
III.4. Programmation.....	53
III.4.1. Structure du programme.....	54
III.4.2. Organisation du programme.....	56
III.4.2.1. Sous-programme moteur.....	56
III.4.2.1.1. Définition le moteur a courant continue.....	56
III.4.2.1.2. Caractéristiques principales.....	56
III.4.2.1.3. Choix du moteur à courant Continu.....	56
III.4.2.1.4. Pont En H.....	57
III.4.2.1.4.1. Principe de fonctionnement.....	57
III.4.2.1.4.2. L293D.....	58
III.4.2.1.4.3. Brochage.....	58
III.4.2.1.2. Programme de moteur.....	60
III.4.2.2. Sous-Programme mouvement de la porte	62

III.4.2.3. Écran LCD pour système d'ascenseur.....	62
III.4.2.3.1. Description Générale	62
III.4.2.3.2. Fonctionnement (Écran LCD).....	64
III.4.2.3.3. Brochage d'un afficheur LCD.....	63
III.4.2.3.4. Fonctionnement de l'algorithme LCD.....	64
III.4.2.4. Capteurs de position.....	67
III.4.2.4.1. Sous-programme : détection de la position.....	67
III.4.2.5. Capteur Infrarouge.....	67
III.4.2.5.1. Principe de fonctionnement.....	68
III.4.2.5.2. Caractéristiques Techniques.....	68
III.4.2.6. Bouton poussoir.....	69
III.5. Simulation du système de contrôle d'ascenseur par Arduino Uno	69
III.5.1. Gestion et fonctionnement de l'ascenseur.....	71
III.6. Conclusion.....	72

Chapitre IV : Résultats De Simulation.

IV.1. Introduction	74
IV.2. Fonctionnement du système	74
IV.2.1. Appel de la cabine d'ascenseur	74
IV.2.2. Appeler le Quatrième étage et demander le mot de passe	77
IV.3. Programme Arduino code.....	79
IV.4. Conclusion.....	81
Conclusion Générale.....	82

Références bibliographiques

Résumé

Liste des Figures

Figure I.1 : Ascenseur.	05
Figure I.2 : Ascenseur de charge.....	05
Figure I.3 : Monte-charge.	06
Figure I.4: Monte-charge industriel.	07
Figure I.5: Ascenseur sur plan incliné.....	07
Figure I.6: Ascenseur hydraulique.	08
Figure I.7: Ascenseur à traction à câble.	08
Figure I.8: Ascenseurs sans salle des machines	09
Figure I.9: Ascenseurs hydrauliques.....	09
Figure I.10: Ascenseurs à vis sans fin.	10
Figure I.11: Composants mécaniques d'un ascenseur.	11
Figure I.12: Cabine d'Ascenseur.	11
Figure I.13: Contrepoid.	12
Figure I.14: Câbles de traction.	12
Figure I.15: machine de traction.	13
Figure I.16: poulie de traction	13
Figure I.17: Rails de guidage.	14
Figure I.18: Amortisseurs de cuvette.	14
Figure I.19: Limiteur de vitesse d'un ascenseur	15
Figure I.20: Systèmes de freinage de sécurité.	16
Figure I.21: Moteurs à courant continu.	16
Figure I.22 : Schéma d'un moteur à excitation indépendante.	17
Figure I.23 : Schéma d'un moteur à excitation parallèle.	17
Figure I.24 : Schéma d'un moteur à excitation série.	18

Figure I.25 : Schéma d'un moteur à excitation composée.	18
Figure I.26 : Moteur pas à pas.	19
Figure I.27: Moteur servo.	19
Figure I.28 : Moteur brushless.	20
Figure I.29: Variateur de vitesse.	21
Figure I.30: Synoptique des variateurs de Vitesse	22
Figure I.31: Pilotage de la vitesse de déplacement.	22
Figure I.32: Pilotage de la vitesse en Boucle ouverte.	24
Figure I.33: Pilotage de la vitesse en Boucle fermée....	24
Figure I.36: Capteur fin de cours et magnétique.	26
Figure I.37: Capteurs de vitesse.....	26
Figure I.38: Capteurs de charge.....	27
Figure I.39: Automate programmable industriel.	28
Figure I.40: Structure d'un système automatisé	29
Figure II.1 : Carte Arduino UNO.....	34
Figure II.2 : Architecture d'un système à base de microprocesseur.	36
Figure II.3 : Architecture d'un microprocesseur.	37
Figure II.4 : Eléments principaux d'un microcontrôleur	39
Figure II.5 : Différents types de boîtiers de microcontrôleurs.....	39
Figure II.6: Structure des composants de système.....	40
Figure II.7 : Chaîne d'information.....	41
Figure II.8: Chaîne d'énergie d'ascenseur.....	41
Figure III.1: Fenêtre du module de simulation Proteus ISIS.	46
Figure III.2: Fenêtre du module de simulation Proteus ARES.	46
Figure III.3: Arduino IDE.....	47

Figure III.4: Barre d'Action	48
Figure III.5: Message d'Erreur	48
Figure III.6: Compilation.....	48
Figure III.7: Arduino Uno.....	50
Figure III.8: Différents broches de la carte Arduino Uno.....	50
Figure III.9: Structure du programme.....	56
Figure III.10: Moteur à courant continu	57
Figure III.11: Pont en « H ».....	58
Figure III.12: Fonctionnement dans le sens horaire.....	59
Figure III.13: Fonctionnement dans le sens antihoraire	59
Figure III.14: Configuration des Broches du L293D.....	60
Figure III.15: Organigramme du moteur.....	62
Figure III.16: Programme mouvement de la porte.....	63
Figure III.17: Afficheur LCD16*2.	64
Figure III.18: Circuit PCF8574.....	65
Figure III.19: Programme d'afficher LCD.....	64
Figure III.20 : Types des capteurs de présence.	65
Figure III.21 : Sous-programme : détection de la position.....	65
Figure III.22 : Capteur de présence.....	66
Figure III.23 : Ondes IR (infrarouge).	66
Figure III.24 : Bouton-poussoir.	67
Figure III.25: Simulation de système de contrôle d'ascenseur par Arduino Uno	67
Figure IV.1: Choix d'étage	76
Figure IV.2: Choix d'étage 4ème	78

Liste des tableaux

Tableau II.1: synthèse des besoins.....	34
Tableau II.2: Différentes microcontrôleur disponibles.....	34
Tableau III.1: Différentes broches de la carte ARDUINO UNO.	53
Tableau III.2: Tableau de Brochage d'un afficheur LCD.....	64
Tableau III.3 : Pins de circuit PCF8574.....	65

Introduction Générale

Introduction Générale

L'évolution et le développement de la modernisation des villes, avec les infrastructures verticales en continue progression, ont fait de l'ascenseur un incontournable moyen de mobilité dans les bâtiments à plusieurs étages. Longtemps considéré que comme un simple moyen de transport vertical, un ascenseur n'est plus qu'un "tube dans lequel nous pouvons monter et descendre d'un étage à l'autre" mais c'est un système technologique complexe dont la commande électrique est en jeu de sécurité, d'efficacité énergétique et de confort pour ces utilisateurs. En effet au fil des années les ascenseurs ont pris une grande transition de mécanique pure vers l'intelligence électromécanique intégrant des composants électroniques avancés, des capteurs, des automates programmables avancés et des algorithmes du contrôle.

Dans ce contexte, la conception d'un système de commande électrique et la simulation deviennent des efforts essentiels pour prévoir le comportement dynamique d'un ascenseur.

Cependant, malgré les avancées réalisées dans ce domaine, plusieurs problèmes persistent. En fait, les systèmes de commande classiques ont des lacunes en matière de réactivité et de consommation d'énergie, de tolérance aux pannes et de conformité aux normes de sécurité. De plus, le manque d'un processus de simulation de la conception détaillé pendant la phase conceptuelle peut entraîner divers niveaux d'inefficacité et même un mauvais fonctionnement du système. Ainsi, ce travail de recherche a pour objet la conception et la simulation d'un système de commande électrique. L'objectif principal de ce projet est de modéliser et de simuler un ascenseur électromécanique à trois étages plus le rez-de-chaussée. Ce projet est structuré en quatre chapitres :

Le premier chapitre est consacré aux fondements théoriques des ascenseurs. Il présente un aperçu général des différents types d'ascenseurs, de leur mode de fonctionnement, ainsi qu'une analyse de leurs avantages et inconvénients respectifs. L'accent y est mis sur les ascenseurs à traction, en détaillant leurs systèmes de conduite et de contrôle, ainsi que les composants mécaniques et électroniques qui les constituent. **Le deuxième chapitre** traite du cahier des charges fonctionnel, en proposant une évaluation comparative des différentes solutions techniques envisagées, accompagnée d'une analyse de leurs points forts et faibles. Il est également question de l'abandon de certaines chaînes fonctionnelles telles que la chaîne d'information et la chaîne énergétique dans le cadre de ce projet. **Le troisième chapitre** aborde la méthodologie adoptée pour la simulation et la programmation. Il y est décrit l'environnement de travail, les outils et logiciels utilisés, ainsi que

les composants matériels clés comme la carte Arduino, les capteurs, les moteurs et l'écran LCD. Ce chapitre inclut aussi la présentation des algorithmes et organigrammes régissant les séquences de fonctionnement de l'ascenseur. Enfin, **le quatrième chapitre** présente les résultats obtenus lors des simulations, avec une analyse détaillée des performances du système. Il décrit le déroulement du fonctionnement de l'ascenseur, depuis la réception de la commande jusqu'à l'arrivée de la cabine et le mécanisme d'ouverture et de fermeture des portes.

En fin de compte, nous résumerons les principaux résultats et les défis les plus significatifs de l'étude - notamment la limitation du nombre de broches de l'Arduino Uno et la complexité du programme - et suggérerons des moyens de les améliorer à l'avenir, tels que le passage à un Arduino Mega, l'ajout de capteurs de surcharge et de sécurité, et des mécanismes de protection active de recouvrement de portes.

Problématique :

Le problème de recherche découle des difficultés rencontrées par les employés, en particulier les personnes à mobilité réduite et les personnes âgées, lors de l'utilisation des ascenseurs. Ainsi, la question principale de l'étude est : **Est-il possible de concevoir un ascenseur électrique fonctionnant avec Arduino pour un bâtiment de trois étages ?**

Chapitre I :

Généralités sur les ascenseurs

I.1.Introduction :

Les ascenseurs occupent une place essentielle dans notre vie quotidienne, facilitant le transport vertical dans les bâtiments modernes, qu'ils soient résidentiels, commerciaux ou industriels. Véritables chefs-d'œuvre technologiques, ils incarnent l'évolution de l'ingéniosité humaine, répondant aux besoins croissants d'urbanisation et d'optimisation de l'espace. Dès leurs origines modestes comme simples systèmes de levage, les ascenseurs ont parcouru un long chemin pour devenir des dispositifs complexes, intégrant des technologies avancées telles que l'automatisation et l'efficacité énergétique. Leur importance réside non seulement dans leur capacité à économiser du temps et de l'effort, mais aussi dans leur rôle crucial dans l'accessibilité universelle, rendant les espaces accessibles à tous, sans exception.

I.2.Généralité sur Les ascenseurs.

I.2.1. Définition des ascenseurs:

Un ascenseur est un dispositif utilisé pour transporter des personnes ou des biens de manière verticale entre différents niveaux d'un bâtiment. Il est généralement constitué d'une cabine suspendue par des câbles ou soutenue par des pistons, et est actionné par un moteur électrique ou un système hydraulique. Les ascenseurs jouent un rôle crucial dans les infrastructures modernes, en facilitant l'accès, en optimisant l'espace urbain et en assurant une mobilité fluide et sécurisée, notamment dans les immeubles de grande hauteur[21].

I.2.2. Leur Importance Dans La Vie Quotidienne

Les ascenseurs jouent un rôle fondamental dans la vie quotidienne, en facilitant le déplacement vertical dans les bâtiments modernes. Ils permettent de gagner du temps, surtout dans les immeubles de grande hauteur, où grimper les escaliers serait fatigant et peu pratique. Leur importance est particulièrement remarquable pour les personnes à mobilité réduite, comme les personnes âgées ou en situation de handicap, en garantissant une accessibilité universelle. De plus, dans les espaces commerciaux et industriels, les ascenseurs optimisent le transport des biens lourds, améliorant ainsi l'efficacité opérationnelle. En somme, les ascenseurs contribuent à une meilleure qualité de vie en rendant les espaces plus fonctionnels, accessibles et pratiques pour tous[21].

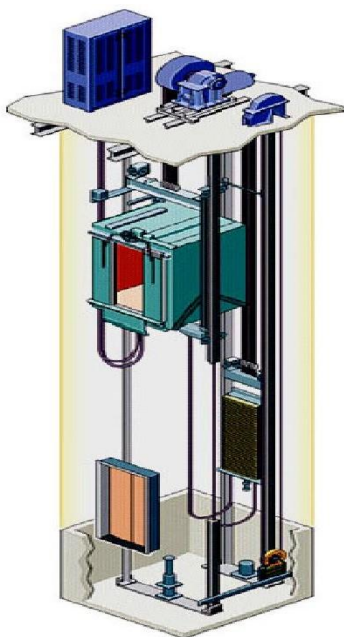


Figure I.1: Ascenseur.

I.3. Différent types d'ascenseur :

I.3.1. Ascenseur de charge :

Ascenseur de charge est un ascenseur principalement destiné au transport de charges qui sont généralement accompagnés par des personnes. En particulier cet appareil élévateur sert uniquement au transport des objets, lorsque les dimensions et la constitution de leur cabine permet l'accès à des personnes et d'atteindre les boutons de commande, ces dispositifs doivent être classés dans la catégorie "Ascenseurs" et non "Monte-charge"[21].



Figure I.2: Ascenseur de charge.

I.3.2. Monte-charge :

Le fonctionnement de ces appareils élévateurs reste le même avec les ascenseurs de charge, cependant la cabine n'est pas accessible à des personnes et les dimensions diffèrent, ces dernières sont données comme suit:

Surface = 1 m² ;

Profondeur = 1 m ;

Hauteur = 1,20 m ;

Toutefois la hauteur peut être supérieure à 1,20m si la cabine comporte plusieurs compartiments fixes répondant chacun aux dimensions ci-dessus. [1]



Figure I.3: Monte-charge

I.3.3. Monte-charge industriel :

Un appareil de levage installé à l'industrie, desservant des niveaux définis, qui comporte une cabine ou un plateau accessible aux personnes pour le chargement ou déchargement, qui se déplace le long d'un ou de plusieurs guides verticaux ou dont l'inclinaison est inférieure par rapport à la verticale à 15 degrés. La commande de ces appareils ne peut se faire que de l'extérieur. Ces dispositifs sont interdits au transport de personnes[16].



Figure I.4: Monte-charge.

I.3.4. Ascenseur sur plan incliné :

Tout ascenseur sur plan incliné utilisé principalement au transport de personnes à mobilité réduite. Cet appareil peut être équipée d'un siège et/ou d'une plate-forme pour le transport d'une personne en position debout ou d'une plate-forme pouvant recevoir un fauteuil roulant.

Les ascenseurs sur plan incliné peuvent être installés dans des cages d'escalier droites ou présentant des virages. Ils peuvent également être installés à l'extérieure des bâtiments dans des rampes ou escalier d'accès[21].



Figure I.4: Ascenseur sur plan incliné.

I.4.Familles D'ascenseur :

On distingue essentiellement deux types de familles d'ascenseur :

I.4.1. Ascenseurs Hydrauliques :

Dans ces types la pompe met sous pression l'huile qui pousse le piston hors du cylindre vers le haut. Lorsque la commande de descente est programmée, la vanne de la pompe permet de laisser sortir l'huile du cylindre vers le réservoir. [1]

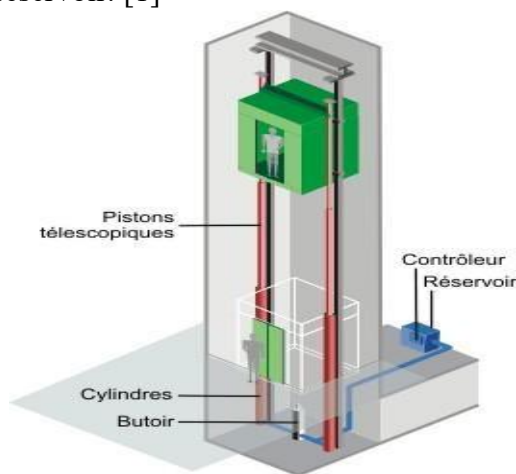


Figure I.6: Ascenseur hydraulique.

I.4.2. Les ascenseurs à traction à câble (électrique) :

Les ascenseurs à traction câble sont les types d'ascenseurs que l'on rencontre souvent. Ils se différencient entre eux selon le type de motorisation.

En générale, ses deux types utilisent l'énergie électrique pour déplacer les cabines verticalement (moteur électrique à courant continu ou alternatif)[21].

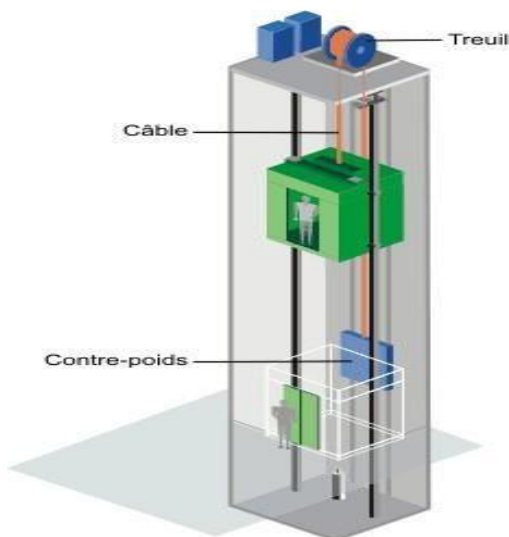


Figure I.7: Ascenseur à traction à câble.

I.5. Classification des ascenseurs selon le mécanisme de levage :

I.5.1. Ascenseurs à traction électrique avec salle des machines (MR) :

Moteur situé au-dessus de la gaine. Traditionnel, fiable mais encombrant.

I.5.2. Ascenseurs à traction sans salle des machines (MRL) :

Moteur compact intégré directement dans la gaine. Réduction de l'espace requis, maintenance facilitée.

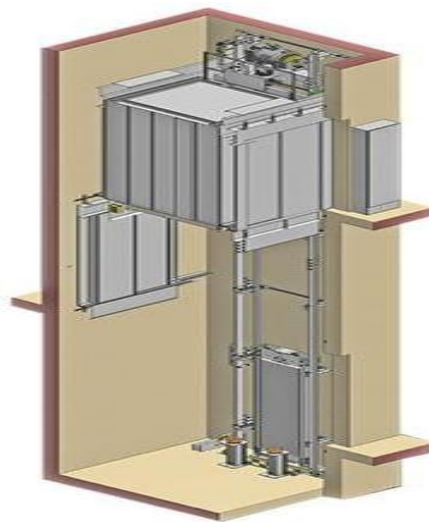


Figure I.8: Ascenseurs sans salle des machines.

I.5.3. Ascenseurs hydrauliques :

Moteur actionnant une pompe qui alimente un vérin. Montée sous pression, descente par gravité. Moins coûteux mais cycles plus lents et consommation plus élevée. [21]

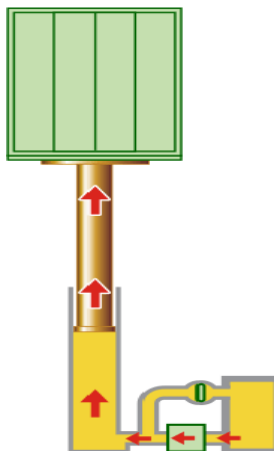


Figure I.9: Ascenseurs hydrauliques.

I.5.4. Ascenseurs à tambour ou à vis sans fin :

Systèmes spécialisés pour des installations atypiques ou industrielles, avec des limites en vitesse et hauteur. [5]

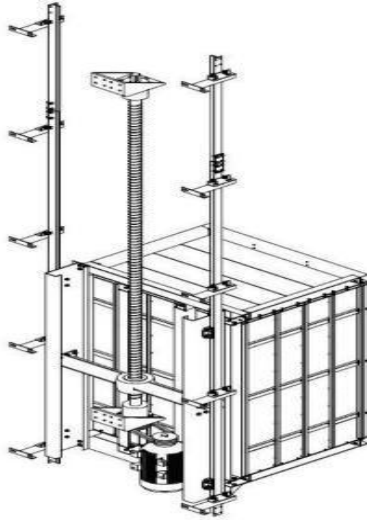


Figure I.10: Ascenseurs à vis sans fin.

I.6. Composants des systèmes de L'ascenseur :

Les ascenseurs sont des systèmes complexes, constitués de plusieurs sous-ensembles mécaniques, électriques et électroniques, et parfois des composantes hydrauliques, interagissant de manière coordonnée pour assurer le déplacement vertical sécurisé des personnes et des charges. La compréhension des composants fondamentaux de ces systèmes est une étape préalable et indispensable à toute démarche de conception et de simulation. Ce chapitre présente une vue d'ensemble des principaux éléments qui composent un ascenseur moderne, en les regroupant selon leur fonction. [21]

I.6.1. Composants mécaniques d'un ascenseur :

Les composants mécaniques forment la structure physique de l'ascenseur et jouent un rôle fondamental dans la transmission des forces et le guidage du mouvement[16].

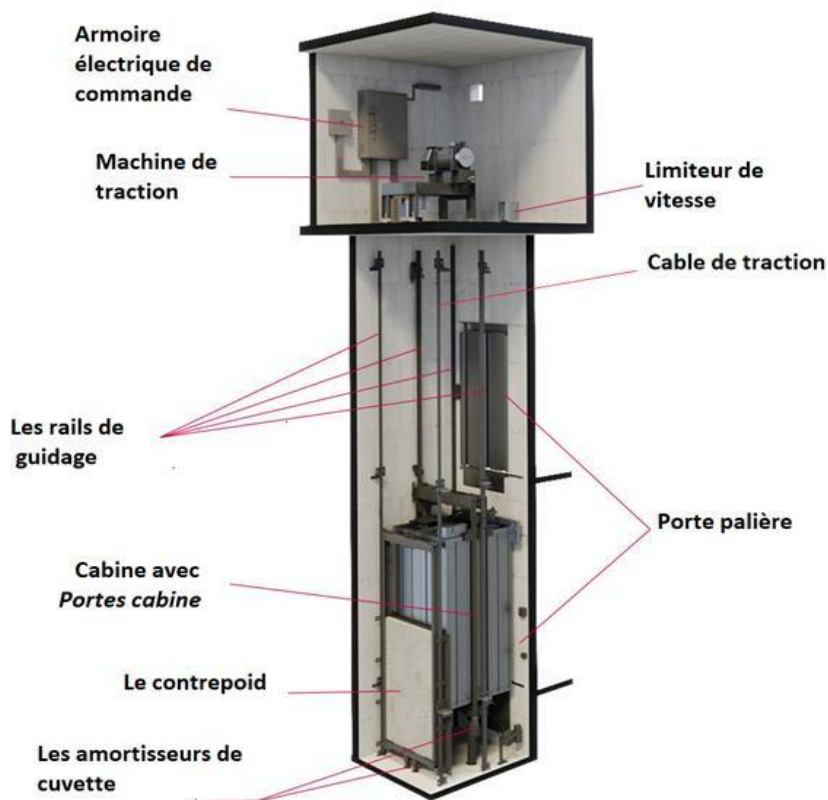


Figure I.11: Composants mécaniques d'un ascenseur.

1- La cabine d'ascenseur :

Fabriquée en matériaux robustes (acier, aluminium), elle est conçue pour transporter en toute sécurité les passagers ou les charges. Elle peut être suspendue (dans le cas d'un système à traction) ou posée sur un vérin (système hydraulique).



Figure I.12: Cabine d'ascenseur.

Chapitre I : Généralité sur Les Ascenseurs

Son design intérieur prend aussi en compte le confort, l'éclairage, la ventilation et parfois l'accessibilité aux personnes à mobilité réduite.

Elle est constituée de quatre (4) parties principales qui - dans l'ordre de leur montage pendant l'installation - sont : L'étrier (suspension cabine) ; Le plancher ou Le socle (plate-forme) ; Les parois et Le toit. [2]

2- Le contrepoids :

Il permet de compenser la masse de la cabine et sa charge nominale moyenne, réduisant ainsi la charge effective sur le moteur. Il est relié à la cabine par un ensemble de câbles qui passent par la poulie de traction. Cette disposition améliore la stabilité et diminue les pics de consommation électrique.

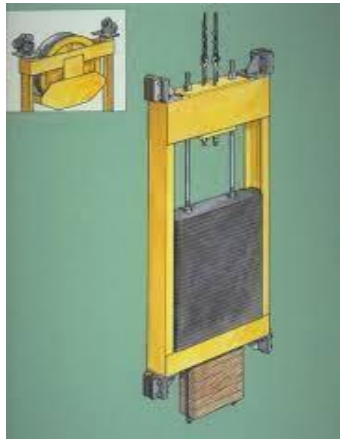


Figure I.13: Contrepoid.

3- Les câbles de traction :

Composés de fils d'acier torsadés, ils doivent résister à des charges dynamiques importantes. Leur diamètre, le nombre de brins et le type d'âme centrale sont choisis en fonction de la capacité de charge de l'ascenseur. (Voir la figure I.14).



Figure I.14: Câbles de traction.

4- machine de traction :

La machine de traction est l'élément principal responsable du mouvement vertical de la cabine et du contrepoids dans un ascenseur à traction. Elle transmet la puissance mécanique nécessaire au levage, en enroulant ou déroulant les câbles de traction autour d'une poulie motrice[21].

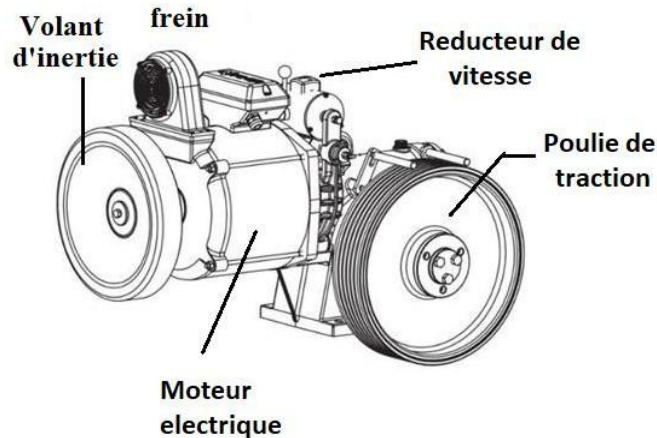


Figure I.15: Machine de traction.

Elle est composée principalement de : Un moteur électrique (asynchrone ou synchrone selon le type) ; Une poulie de traction (poulie motrice) autour de laquelle passent les câbles ; Un réducteur de vitesse (dans les modèles à réducteur) et Un frein électromagnétique pour immobiliser la cabine à l'arrêt. (Voir la figure I.15).

5- La poulie motrice (poulie de traction) :

Connectée à l'arbre moteur, elle assure le déplacement vertical par adhérence entre les câbles et sa gorge. Son usinage précis est crucial pour éviter le glissement et garantir un mouvement fluide. (Voir la figure ci-dessous)[21].

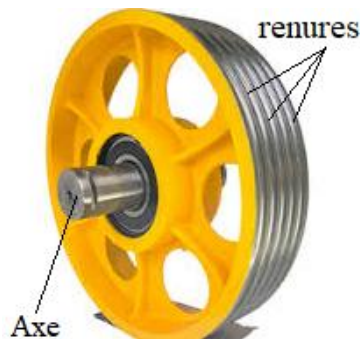


Figure I.16: Poulie de traction.

6- Les rails de guidage :

Généralement en acier laminé, ils maintiennent la cabine et le contrepoids dans une trajectoire verticale stricte. Ils absorbent aussi les vibrations et les déformations dues aux forces latérales. (voir la figure I.17) [21].



Figure I.17: Rails de guidage.

7- Les amortisseurs de cuvette (Tampons) :

Placés en fond de fosse, ces dispositifs absorbent l'énergie cinétique en cas de dépassement de course. On distingue les amortisseurs à ressort, à huile ou à gaz, selon la vitesse nominale de l'ascenseur. (Voir la figure I.18) [21].



Figure I.18: Amortisseurs de cuvette.

8- Les Portes d'ascenseur :

A- Portes cabine :

Portes de cabine utilisées pour fermer la cabine elle-même ; tous les ascenseurs doivent en être équipés pour éviter que les usagers soient coincés par la porte palière.

B- Portes palières :

Portes palières synchronisées avec la cabine, elles restent verrouillées sauf lors de l'arrêt exact de la cabine au palier concerné, assurant ainsi la sécurité des usagers[21].

8- Limiteur de vitesse d'un ascenseur :

Le limiteur de vitesse est un dispositif de sécurité mécanique et parfois électromécanique, conçu pour détecter tout dépassement de la vitesse nominale de la cabine d'ascenseur, notamment en descente, et pour activer le système de freinage d'urgence (parachute).

Il est généralement constitué d'une poulie reliée à un câble tendu verticalement dans la gaine de l'ascenseur[21].

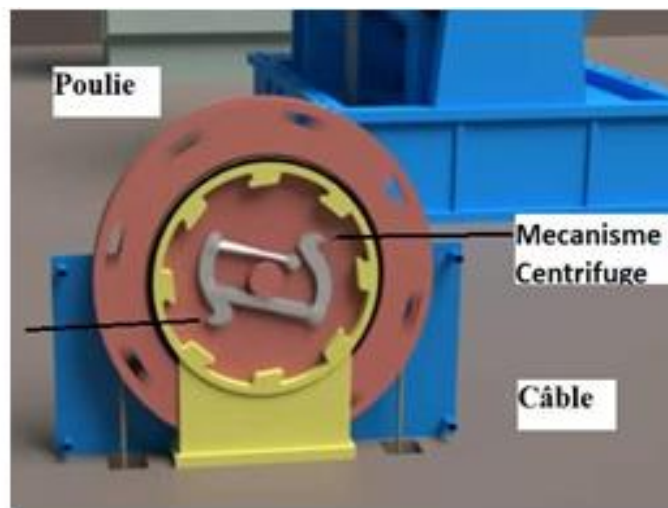


Figure I.19: Limiteur de vitesse d'un ascenseur.

➤ Principe de fonctionnement de limiteur de vitesse :

Lorsque la **vitesse de la cabine dépasse la limite réglée**, la poulie déclenche un

mécanisme centrifuge ou à came. Cela active:

- Soit un contact électrique qui coupe l'alimentation du moteur,
- Soit un frein mécanique qui agit sur la cabine via le parachute.

9- Systèmes de freinage de sécurité :

Ils interviennent en cas de survitesse ou de rupture de câble. Ils peuvent être électromécaniques ou actionnés par un régulateur de vitesse mécanique.

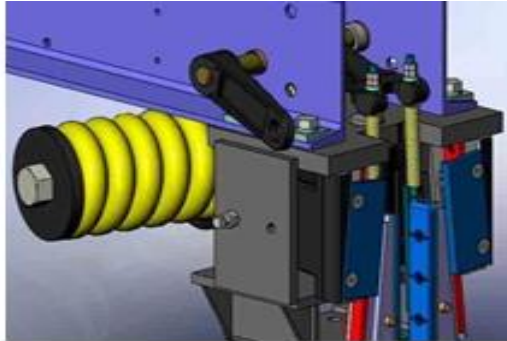


Figure I.20: Systèmes de freinage de sécurité.

I.6.2. Composants électriques d'un ascenseur :

Les composants électriques sont indispensables au fonctionnement et à la sécurité des ascenseurs électromécaniques. Ils assurent la distribution de l'énergie et le contrôle du système grâce à divers éléments comme le tableau de commande, les contacteurs, et les relais de sécurité. Les capteurs de position et interrupteurs de fin de course jouent un rôle clé dans la surveillance. L'unité centrale de contrôle, telle qu'un automate ou un microcontrôleur (ex : Arduino UNO), coordonne ces éléments. Une intégration optimale garantit fiabilité, efficacité et sécurité de l'installation.

1- Le Moteur Electrique :

Les moteurs électriques sont des dispositifs essentiels dans les systèmes modernes, permettant la conversion de l'énergie électrique en énergie mécanique.

Leur diversité répond à une large gamme d'applications industrielles et domestiques. Une classification claire s'avère donc nécessaire pour mieux comprendre leurs principes, leurs performances et leurs domaines d'utilisation[14].



Figure I.21: Moteurs à courant continu.

1-1- Moteurs à courant continu (DC Motors) :

Fonctionnent en convertissant l'énergie électrique en courant continu en énergie mécanique.

Les moteurs à courant continu (DC) se divisent en plusieurs types, chacun ayant des caractéristiques spécifiques adaptées à différentes applications :

1-1-1- Moteur à excitation séparée (Indépendante) :

Le moteur à excitation indépendante est raccordé à une alimentation à courant continu séparée.

Par conséquent, le courant qui alimente l'inducteur est indépendant de celui qui alimente l'induit.

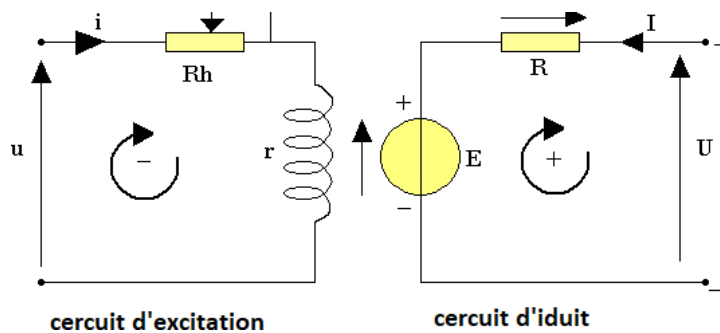


Figure I.22 : Schéma d'un moteur à excitation indépendante.

1-1-2- Moteur à excitation parallèle (shunt) :

Dans le système d'excitation en dérivation (shunt), l'induit et l'inducteur sont alimentés par la même source, avec l'inducteur connecté en parallèle à l'induit. Cette configuration permet une vitesse de rotation pratiquement constante, indépendamment de la charge, et un démarrage à vide.

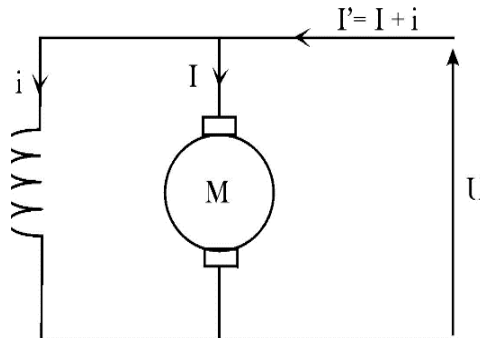


Figure I.23 : Schéma d'un moteur à excitation parallèle.

1-1-3- Le moteur à excitation série :

Moteur à excitation série est un type de moteur à courant continu (DC) dans lequel le bobinage de l'inducteur (l'excitation) est connecté en série avec l'induit (le rotor). Cela signifie que le même courant traverse à la fois l'inducteur et l'induit.

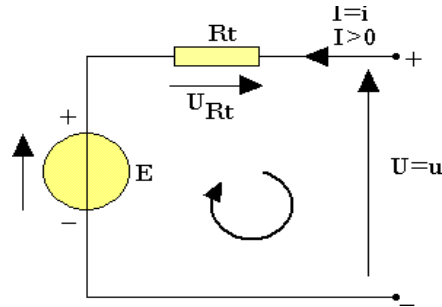


Figure I.24 : Schéma d'un moteur à excitation série.

1-1-4- Moteur à excitation composée :

Le moteur à excitation composée est un moteur à courant continu (CC) qui combine les caractéristiques des moteurs à excitation série et parallèle (shunt). Cette combinaison permet d'obtenir une meilleure régulation de la vitesse et un bon compromis entre couple et stabilité.

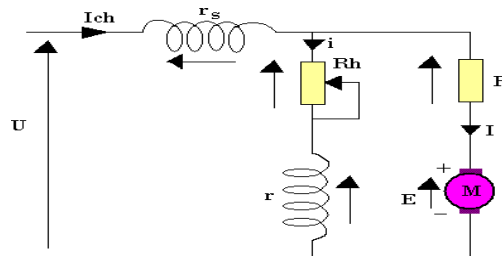


Figure I.25 : Schéma d'un moteur à excitation composée.

1-1-5- Moteur pas à pas (Stepper Motors) :

Le moteur pas-à-pas est un moteur électrique utilisé pour les applications nécessitant un contrôle précis de la position et de la vitesse. Il fonctionne en avançant par "pas" grâce à une commande électronique qui active ses bobines (1) et (2) dans un ordre spécifique. Utilisés dans les robots et les imprimantes 3D ...etc.

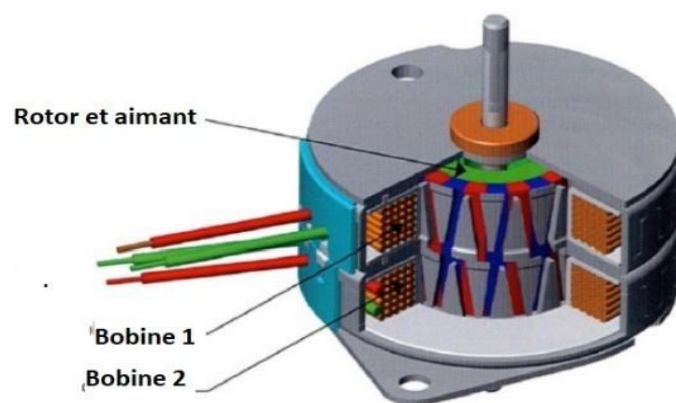


Figure I.26 : Moteur pas à pas.

1-1-6- Moteur servo (Servo Motors) :

Le moteur servo est un moteur électrique utilisé pour des applications nécessitant un contrôle très précis de la position, de la vitesse et du couple (utilisés dans l'automatisation industrielle et la robotique...etc.).

Il est souvent accompagné d'un capteur et d'un contrôleur électronique pour ajuster son mouvement en fonction des besoins.

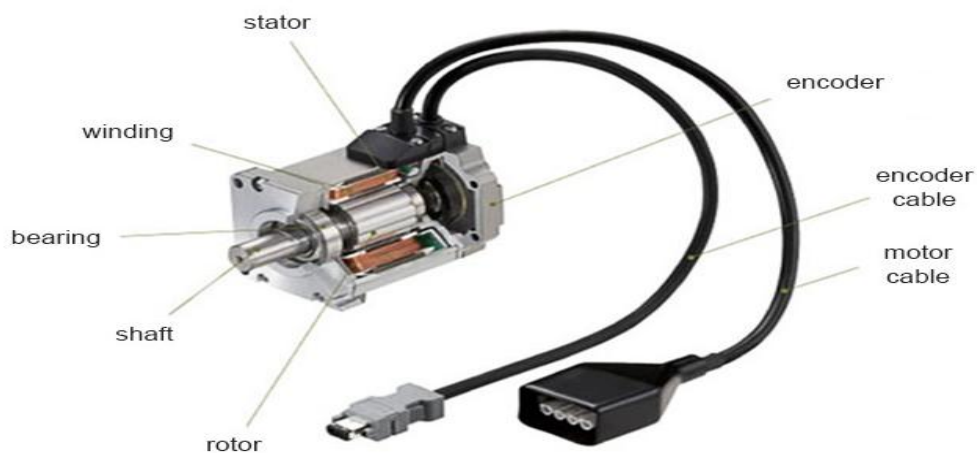


Figure I.27: Moteur servo.

1-1-7- Moteur brushless (Brushless DC Motor – BLDC):

Le moteur brushless, ou moteur à courant continu sans balais, est un type de moteur électrique qui fonctionne sans balais ni collecteur, contrairement aux moteurs

à courant continu. Elle est utilisée dans les **véhicules** électriques ; et **Drones** ...etc.

Le moteur brushless fonctionne grâce à une commutation électronique qui alimente les enroulements du stator de manière séquentielle, créant un champ magnétique tournant qui entraîne le rotor à aimants permanents sans contact mécanique[14].

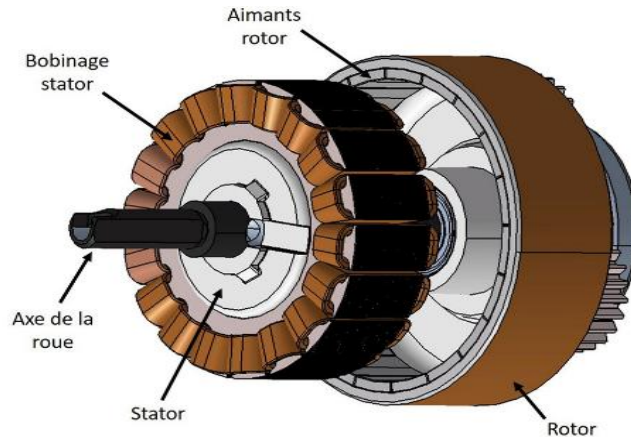


Figure I.28: Moteur brushless

1-2- Moteurs à courant alternatif (AC Motors) :

Les moteurs à courant alternatif se classent principalement en deux grandes catégories : les moteurs synchrones et les moteurs asynchrones.

1-2-1- Les Moteurs Asynchrones :

Le moteur asynchrone, aussi appelé moteur à induction, est l'un des moteurs électriques les plus utilisés dans l'industrie et les équipements domestiques. Son principe repose sur l'induction électromagnétique entre le stator et le rotor. (Moteurs asynchrones, ils nécessitent une excitation externe pour fonctionner).

1-2-2- Les moteurs Synchrones :

Les moteurs synchrones sont des machines électriques tournantes dont le rotor tourne à la même vitesse que le champ magnétique tournant du stator.

➤ Observation :

Dans les ascenseurs modernes, les moteurs à courant alternatif sont les plus utilisés. Les moteurs synchrones à aimants permanents sont privilégiés pour leur rendement élevé, leur compacité et leur fonctionnement silencieux, en particulier dans les

ascenseurs sans salle des machines (MRL : Machine Room-Less).

2- Le variateur de vitesse :

2.1. Définition :

Un variateur de vitesse est un dispositif électronique de puissance conçu pour réguler la vitesse de rotation, le couple et le sens de marche d'un moteur électrique, principalement les moteurs asynchrones triphasés, en modifiant la fréquence et l'amplitude de la tension d'alimentation. Il permet de modifier la fréquence et la tension du courant fourni au moteur, assurant un démarrage progressif, un arrêt doux et une régulation précise[27].



Figure I.29: Variateur de vitesse.

2.2. Principe de base des variateurs de vitesse :

Le variateur de vitesse est composé essentiellement :

D'un redresseur qui est connecté à une alimentation triphasée (le réseau), il génère une tension continue à ondulation résiduelle (le signal n'est pas parfaitement continu). Le redresseur peut être de type commandé ou non commandé.

D'un circuit intermédiaire agissant principalement sur le "lissage" de la tension de sortie du redresseur (améliore la composante continue). Le circuit intermédiaire peut aussi servir le dissipateur d'énergie lorsque le moteur devient un générateur.

D'un onduleur qui engendre le signal de puissance à tension et/ou fréquences variables.

D'une électronique de commande pilotant (transmission et réception des signaux) le redresseur, le circuit intermédiaire et l'onduleur.

2.3. Synoptique des variateurs de Vitesse :

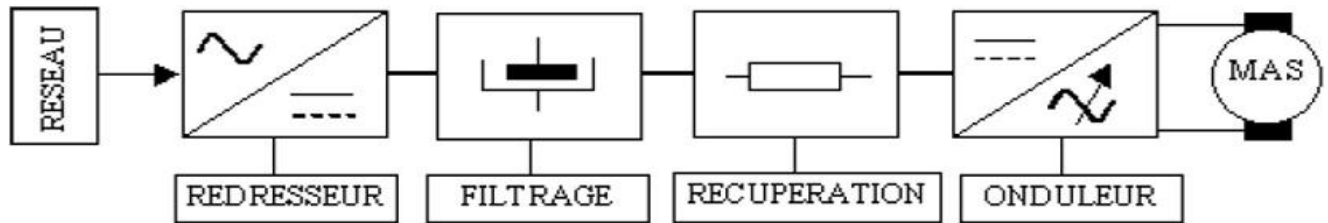


Figure I.30: Synoptique des variateurs de Vitesse.

Le variateur de vitesse est principalement caractérisé selon la séquence de commutation qui commande la tension d'alimentation du moteur. On a :

- Les variateurs à source de courant (CSI).
- Les variateurs à modulation d'impulsions en amplitude (PAM).
- Les variateurs à modulation de largeur d'impulsion (PWM/VVC)[27].

2.4. Fonctions des variateurs de vitesse :

Au niveau des ascenseurs, parmi la multitude de possibilités de fonctions qu'offrent les variateurs de vitesse actuels, on épinglera :

- L'accélération contrôlée ; - La décélération contrôlée ; - Le freinage.
- La variation et la régulation de vitesse ; - L'inversion du sens de marche.

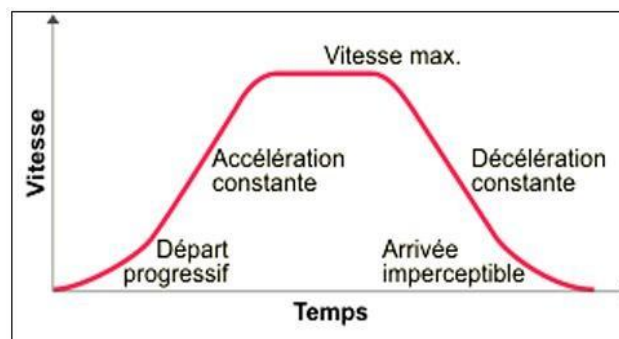


Figure I.31: Pilotage de la vitesse de déplacement

2-4-1- Accélération contrôlée :

Le profil de la courbe de démarrage d'un moteur d'ascenseur - qui peut être linéaire - est avant tout lié au confort des utilisateurs dans la cabine. Ce profil ou "rampe" est la plupart du temps ajustable en permettant de choisir le temps de mise en vitesse de l'ascenseur[27].

2-4-2- Décélération contrôlée :

Les variateurs de vitesse permettent une décélération contrôlée sur le même principe que l'accélération. Dans le cas des ascenseurs, cette fonction est capitale dans un sens où l'on ne peut pas se permettre simplement mettre le moteur hors tension et d'attendre son arrêt complet. Suivant l'importance du couple résistant (le poids du système cabine/contre-poids varie en permanence) ; il faut impérativement contrôler le confort et la sécurité des utilisateurs par le respect d'une décélération supportable et d'une mise à niveau correcte. Au niveau du variateur de vitesse, On distingue deux types de freinage : - En cas de décélération désirée plus importante que la décélération naturelle, le freinage peut être électrique, soit par renvoi d'énergie au réseau d'alimentation, soit par dissipation de l'énergie dans un système de freinage statique. - En cas de décélération désirée moins importante que la décélération naturelle, le moteur peut développer un couple moteur supérieur au couple résistant de l'ascenseur et continuer à entraîner la cabine jusqu'à l'arrêt[27].

2-4-3- Variation et régulation de vitesse :

Parmi les fonctionnements classiques des variateurs de vitesse, on distingue la variation de vitesse proprement dite où la vitesse du moteur est définie par une consigne d'entrée (tension ou courant) sans tenir compte de la valeur réelle de la vitesse du moteur qui peut varier en fonction de la charge et la tension d'alimentation, on est en boucle "ouverte" (pas de feedback)[27].

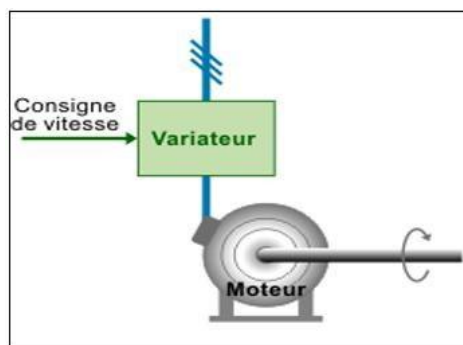


Figure I.32: Pilotage de la vitesse en Boucle ouverte[27].

La régulation de vitesse où la consigne de la vitesse du moteur est corrigée en fonction d'une mesure réelle de la vitesse à l'arbre du moteur introduite dans un comparateur. La consigne et la valeur réelle de la vitesse sont comparées, la différence éventuelle étant corrigée, on est en boucle "fermée"[27].

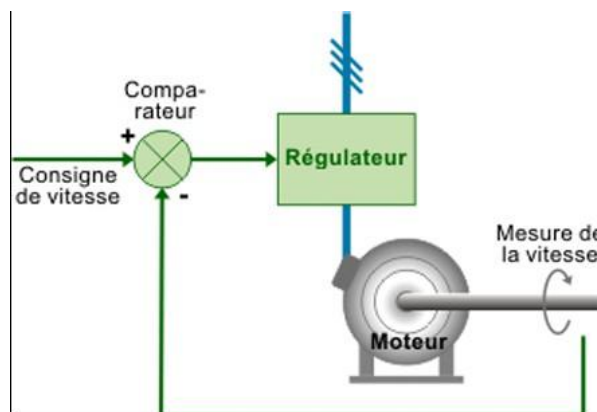


Figure I.33: Pilotage de la vitesse en Boucle fermée[27].

2-4-4- Inversion du sens de marche :

Sur la plupart des variateurs de vitesse, il est possible d'inverser automatiquement le sens de marche. L'inversion de l'ordre des phases d'alimentation du moteur de l'ascenseur s'effectue :

- ✓ Soit par une inversion de la consigne d'entrée.
- ✓ Soit par un ordre logique sur une borne.
- ✓ Soit par une information transmise par une connexion à un réseau de gestion[27].

2-4-5- Freinage d'arrêt :

C'est un freinage de sécurité pour les ascenseurs :

Avec des moteurs asynchrones, le variateur de vitesse est capable d'injecter un courant continu au niveau des enroulements statoriques et par conséquent stopper le champ tournant ; la dissipation de l'énergie mécanique s'effectuant au niveau du rotor (danger d'échauffement important)[27].

- Avec des moteurs à courant continu, le freinage s'effectue au moyen d'une résistance connectée sur l'induit de la machine[27].

2.5. Circuit de commande d'un variateur de fréquence

Le circuit de commande ne fait pas partie du circuit de la puissance du variateur de vitesse. Ce circuit doit garantir quatre fonctions essentielles :

- La commande des semi-conducteurs du variateur de vitesse.
- L'échange d'informations de commande, de régulation et d'analyse avec les périphériques.
- Le contrôle des défauts (interprétation et affichage).
- La protection du variateur de vitesse et du moteur.
- L'apparition des microprocesseurs a permis d'accroître la vitesse d'exécution des informations de commande et de régulation du circuit de commande vis à vis des autres circuits (circuit intermédiaire, onduleur, ...).
- Le circuit de commande est donc en mesure de déterminer le schéma optimum d'impulsions des semi-conducteurs pour chaque état de fonctionnement du moteur par rapport à la charge, au réseau, aux consignes de commande, ...

La régulation de vitesse des moteurs triphasés à courant alternatif évolue selon deux principes de commandes différents :

- La commande U/f (Scalaire).
- La commande vectorielle de tension (ou de flux).

Ces principes déterminent la manière de programmation des algorithmes de commande et de régulation des variateurs de vitesse. Les deux méthodes présentent des avantages en fonction des exigences spécifiques des performances (couple, vitesse, ...) et de la précision de l'entraînement[27].

3- Capteurs :

3.1. Capteurs de position (capteurs magnétiques, codeurs, fins de course) :

Informent le système de commande sur la position exacte de la cabine.



Figure I.36: Capteur fin de cours et magnétique.

3.2. Capteurs de vitesse : utilisés pour ajuster la décélération et contrôler les arrêts.



Figure I.37: Capteurs de vitesse.

3.3. Capteurs de charge : préviennent les surcharges et activent des alarmes en cas de dépassement.



Figure I.38: Capteurs de charge.

4- Tableau de commande et armoires électriques :

Ils centralisent toutes les commandes logiques et les circuits de puissance. On y trouve également les relais, contacteurs, disjoncteurs et dispositifs de protection.

4-1- Systèmes d'alimentation de secours :

Ils peuvent inclure des batteries ou onduleurs (UPS) pour assurer le retour cabine à l'étage le plus proche en cas de coupure d'électricité.

4-2- Les Calculateurs :

Les calculateurs sont des dispositifs électroniques capables de traiter des données en exécutant une série d'instructions préprogrammées. Ils sont utilisés dans de nombreux domaines pour faciliter les calculs, le contrôle, l'analyse et l'automatisation. Ces systèmes reposent sur des microprocesseurs et des logiciels pour accomplir leurs diverses fonctions.

4-3- Microprocesseur (MP) :

Le microprocesseur est l'unité centrale de traitement qui réalise les opérations arithmétiques et logiques. Il est caractérisé par sa taille compacte et sa capacité à exécuter rapidement des instructions spécifiques. Il est utilisé dans : Les ordinateurs personnels et portables ; Les applications de contrôle industriel simples...etc.

4-4- Microcontrôleur (MC) :

Le microcontrôleur est un système autonome combinant le processeur, la mémoire et les unités d'entrée et de sortie. Il est principalement utilisé dans les systèmes nécessitant un contrôle simple. Exemples d'utilisation : Les systèmes de contrôle automobile ; Les robots et systèmes embarqués.....etc.

5- Automate Programmable Industriel (API) :

L'automate programmable industriel (API) est utilisé dans le contrôle industriel et automatisé. Il est plus complexe et puissant que le microcontrôleur. Ses applications incluent : L'opération des chaînes de production industrielles ; Les systèmes d'automatisation dans les usines systèmes de commande et de sécurité : Le système de commande constitue l'“intelligence” de l'ascenseur, capable de prendre des décisions en temps réel pour garantir un fonctionnement optimal.

5-1- Automate programmable industriel (API/PLC) :

Cœur du système de commande, il exécute un programme logique basé sur les entrées (capteurs, boutons) et active les sorties (moteur, freins, portes). Il est programmé avec des langages normalisés comme le ladder (LD) ou le langage structuré.



Figure I.39: Automate programmable industriel

5-2-Les systèmes automatisés :

L'automatisation d'un procédé (c'est-à-dire une machine, un ensemble de machines ou plus généralement un équipement industriel) consiste à en assurer la conduite par un dispositif technologique. Le système ainsi conçu peut prendre en compte les situations pour les quelles sa commande a été réalisée. L'intervention d'un opérateur est souvent nécessaire pour assurer un pilotage global du procédé, pour surveiller les installations et prendre en commande manuelle (non automatique) tout ou une partie du système.

5-3-Structure d'un système automatisé :

L'analyse structurelle conduit à décomposer tout système automatisé comme suit :

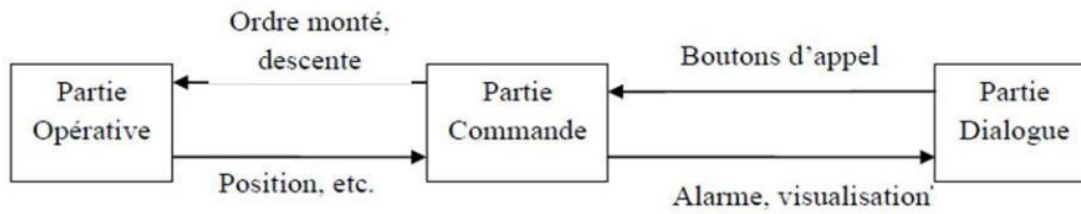


Figure I.40: Structure d'un système automatisé.

5-4-Algorithmes de gestion des appels :

A- Commande simple : traitement premier arrivé, premier servi.

B- Commande collective : optimisation du trajet en regroupant les appels selon leur direction (montée/descente).

C- Commande de groupe (dispatching) : dans les bâtiments à trafic élevé, plusieurs ascenseurs sont coordonnés pour minimiser le temps d'attente.

5-5-Interfaces utilisateur (IHM) :

Écrans tactiles, afficheurs de position, boutons d'appel lumineux, synthèse vocale pour l'accessibilité.

5-6-Télémaintenance et diagnostic à distance :

Permet la surveillance en temps réel, la détection anticipée des pannes et la gestion à distance des interventions de maintenance.

I.7. Conclusion :

En conséquence de ce qui a été exposé, il apparaît clairement que les ascenseurs se présentent sous divers types, différant selon leur conception, leur technologie et leur domaine d'exploitation. Cette diversité impose un choix judicieux et adapté aux besoins spécifiques du projet. Dès lors, il devient essentiel, dans le chapitre suivant, de s'attarder sur les critères fondamentaux qui permettront de définir les caractéristiques précises du système à concevoir, en tenant compte à la fois des exigences fonctionnelles, techniques et économiques.

La compréhension détaillée de l'architecture d'un ascenseur électromécanique, tant sur le plan mécanique qu'électrique, permet d'établir une base solide pour la modélisation et la simulation des systèmes de commande. Les interactions complexes entre les composants nécessitent une approche pluridisciplinaire alliant automatisme, électrotechnique et ingénierie mécanique. Dans le chapitre suivant, nous nous concentrerons sur la conception des systèmes de commande, en mettant en avant les méthodes de régulation, les choix technologiques et les outils logiciels utilisés pour leur mise en œuvre.

Chapitre II :

Cahier des charges

II.1. Introduction :

Dans tout système automatisé, le choix du contrôleur représente une étape cruciale qui conditionne la performance, la fiabilité et la flexibilité de la commande. Lorsqu'il s'agit d'un ascenseur électromécanique, ce choix devient encore plus stratégique, puisqu'il doit répondre à des exigences strictes en termes de sécurité, de gestion des entrées/sorties, de temps de réponse et de simplicité d'implémentation.

Dans le cadre de notre projet, qui vise **la conception et la simulation d'un système de commande électrique pour un ascenseur électromécanique**, il est nécessaire d'opter pour un contrôleur capable non seulement d'interagir efficacement avec les différents composants (capteurs, moteurs, relais, etc.), mais aussi d'être compatible avec les environnements de simulation utilisés dans le développement.

Ce chapitre est donc consacré à la **justification du choix du microcontrôleur Arduino UNO**, en s'appuyant sur une démarche rigoureuse :

- Définition du **cahier des charges fonctionnel**,
- **Analyse comparative** de plusieurs solutions disponibles,
- Identification des **avantages et limites techniques**,
- Enfin, validation du choix en fonction du **contexte pédagogique, économique et technique** du projet.

Cette démarche permet de démontrer en quoi l'Arduino UNO constitue une solution cohérente, pertinente et adaptée à nos objectifs.

Le système de commande joue un rôle fondamental dans le fonctionnement d'un ascenseur électromécanique. Il est responsable de la gestion des appels d'étages, de la régulation du mouvement de la cabine, du contrôle des portes, ainsi que de la sécurité des utilisateurs. Dans le cadre de ce projet, qui vise à concevoir et simuler un modèle réduit ou pédagogique d'un ascenseur, il est indispensable de disposer d'un contrôleur capable de :

- Lire les commandes utilisateur (boutons d'appel)
- Traiter des signaux provenant de capteurs de position ou de fin de course
- Gérer les sorties vers les moteurs et les afficheurs
- Implémenter une logique de commande séquentielle

Le choix du contrôleur doit ainsi répondre à des exigences à la fois **fonctionnelles, économiques, et pratiques**, en particulier dans un contexte de prototypage ou d'apprentissage.

II.2. Cahier Des Charges :

Dans le cadre de la conception et simulation d'un système de commande électrique pour un ascenseur électromécanique, il est indispensable de définir un ensemble clair d'exigences fonctionnelles. Ces exigences, regroupées dans un cahier des charges fonctionnel, guident le choix du contrôleur et assurent que le système sera capable de remplir ses missions principales de manière fiable et cohérente.

Donc ; Etant donné l'ascenseur initialement au repos, à n'importe quel étage.

Un appel émanant d'un étage quelconque se réalise comme suit :

- Si l'appel vient de l'étage où se trouve l'ascenseur, un bip sonore et un clignotement de l'afficheur indique la présence de la cabine à l'étage demandé.
- S'il vient d'un autre étage, alors l'ascenseur monte ou descend vers l'étage demandé, l'afficheur indique le niveau atteint au même temps.

A chaque arrivé à l'étage désiré, un bip sonore et un clignotement de l'afficheur l'indique, la cabine poursuit sa course après un petit retard (delay) (ex : 4 secondes) .

- S'il y a plusieurs appels, c'est le premier qui va être pris en considération.
- Si l'ascenseur doit monter ou descendre de plus d'un étage, alors à l'arrivée de chaque étage intermédiaire, un test de demande d'appel est effectué. La cabine s'arrête pour laisser monter ou descendre les personnes et continuera de répondre à la première demande.

Les priorités sont telles que le sens de la cabine est celui du premier appel.

En appuyant sur le bouton d'arrêt d'urgence le système s'arrête immédiatement, un affichage de la lettre "U" et une alarme (bip sonore sans arrêt) indique l'état d'urgence.

Dans ce cas on doit initialiser le système (Reset) pour qu'il soit prêt à être utilisé de nouveau.

La demande d'utilisation de la cabine peut se faire au niveau 0 ou à chaque niveau après l'appui sur l'un des boutons poussoirs.

La demande d'urgence peut se faire après l'appui sur le bouton d'urgence.

Le tableau suivant présente une synthèse des besoins identifiés pour ce projet :

Tableau II.1: synthèse des besoins.

Exigence technique	Description
Capacité de lecture d'entrées multiples	Gestion de plusieurs boutons d'appels d'étage, capteurs de fin de course, détecteurs de position
Commande de sorties numériques	Contrôle du moteur, activation de relais pour les portes, affichage LED ou écran LCD
Programmabilité flexible	Possibilité d'implémenter différents algorithmes de commande (simple, collective, etc.)
Faible consommation énergétique	Compatible avec une alimentation limitée, typique des prototypes ou systèmes embarqués. Fonctionne en basse tension (5V)
Coût réduit	Budget limité pour le prototypage ou l'expérimentation pédagogique
Facilité d'utilisation et d'intégration	Développement simplifié, documentation abondante, forte communauté de support.

Ce tableau résume les critères fondamentaux qui doivent être pris en compte lors du choix d'un contrôleur pour ce type de système. Il reflète les besoins fonctionnels spécifiques à un ascenseur, notamment en matière de gestion d'entrées/sorties, de capacité de traitement, de consommation énergétique et de coût. Ces exigences orientent naturellement vers une solution simple, fiable et ouverte, comme l'Arduino UNO.

II.3. Comparaison entre différentes solutions disponibles :

Avant de retenir le choix du contrôleur Arduino UNO, plusieurs alternatives ont été envisagées, chacune présentant des avantages et des limites. Le tableau comparatif suivant propose une évaluation de trois solutions populaires dans les systèmes embarqués et les applications de commande : **Arduino UNO**, **automate programmable industriel (API/PLC)**, et **Microcontrôleur PIC**.

Tableau II.2: différents Microcontrôleur disponibles.

Critère	Arduino UNO	Automate (PLC)	MC: PIC
Facilité de programmation	Très simple (langage C, C++)	Programmation structurée (ladder, STL)	Moyenne à élevée (C, C++, etc.)
Temps réel / Réactivité	Bonne pour systèmes simples	Excellente (temps réel industriel)	Moins fiable (OS multitâche non temps réel)
Coût	Très faible	Élevé	Moyen
Consommation électrique	Très faible	Variable selon le modèle	Moyenne à élevée
Maintenance	Très simple	Plus compliquer	Moyenne

II.3.1. Analyse comparative :

L'analyse montre que l'**Arduino UNO** constitue une solution équilibrée et particulièrement bien adaptée aux besoins du projet. Il offre une **grande simplicité de mise en œuvre**, une **programmabilité intuitive**, et une **excellente compatibilité** avec les composants usuels des systèmes embarqués.

Contrairement aux **automates industriels**, qui sont plus coûteux et rigides, ou au **Pic**, qui implique un système d'exploitation et une complexité logicielle plus élevée, l'Arduino permet une approche directe, idéale pour la simulation et la validation fonctionnelle du système de commande d'un ascenseur pédagogique.

Donc, Ces avantages font de l'Arduino UNO une plateforme parfaitement adaptée au contexte du projet, qui vise la conception **simple, économique et fonctionnelle** d'un système de commande pour ascenseur électromécanique.

II.4. Avantages techniques et économiques de l'Arduino UNO:

Dans le cadre du développement d'un système de commande pour un ascenseur électromécanique, le choix du microcontrôleur conditionne directement la performance globale du dispositif. C'est dans cette perspective que la carte Arduino UNO s'est imposée comme une solution à la fois efficace sur le plan technique et avantageuse sur le plan économique.



Figure II.1 :la carte Arduino UNO

II.4.1. Compatibilité avec divers capteurs et actionneurs :

L'Arduino UNO dispose de multiples entrées/sorties numériques et analogiques, idéales pour interagir avec les boutons d'appel, capteurs de fin de course, relais, moteurs, etc.

II.4.2. Facilité de programmation :

Grâce à l'environnement de développement (Arduino IDE) simple et accessible, basé sur le langage C/C++, le développement de la logique de commande est rapide et intuitif.

II.4.3. Support de la simulation :

Elle est compatible avec des plateformes comme Proteus, ce qui permet de tester et valider le système sans montage physique initial.

II.4.4. Modularité et flexibilité :

Son architecture permet des modifications faciles dans le code ou le câblage, ce qui est essentiel dans les phases de test et d'optimisation du système de commande.

II.4.5. Documentation abondante :

De nombreuses ressources, bibliothèques et tutoriels sont disponibles gratuitement, facilitant la mise en œuvre du projet.

II.4.6. Réutilisabilité :

La carte Arduino peut être facilement réutilisée dans d'autres projets, ce qui augmente sa rentabilité à long terme.

II.4.7. Coût réduit :

L'Arduino UNO est beaucoup moins coûteux que les automates programmables industriels (PLC), ce qui est un atout majeur pour les projets académiques à budget limité.

II.5. Justification du choix de l'Arduino UNO:

Le choix de la carte Arduino UNO comme contrôleur principal dans notre projet de conception et simulation d'un système de commande pour un ascenseur électromécanique repose sur une série de critères techniques, pédagogiques et économiques rigoureusement analysés.

Tout d'abord, l'Arduino UNO offre une simplicité d'utilisation et une accessibilité exceptionnelle, tant au niveau matériel que logiciel. Son environnement de développement (Arduino IDE), basé sur le langage C/C++, permet une prise en main rapide et une programmation intuitive de la logique de commande (commande de montée/descente, gestion des appels, arrêt aux étages, etc.).

Ensuite, ses capacités d'entrées/sorties sont suffisantes pour un ascenseur de petite échelle, typiquement à 2 ou 3 niveaux, avec des capteurs de fin de course, boutons

d'appel et moteurs à courant continu. Cela répond parfaitement au besoin fonctionnel de notre prototype.

De plus, l'Arduino UNO est largement compatible avec les plateformes de simulation comme Proteus, ce qui nous permet de modéliser et tester virtuellement le système avant tout déploiement physique, réduisant ainsi les risques d'erreurs et le temps de développement.

Enfin, son faible coût et sa disponibilité sur le marché en font une solution idéale dans un cadre académique à budget limité, sans compromettre la qualité de l'apprentissage ni la pertinence technique du projet.

II.6. Architecture d'un système à base de microprocesseur :

Un calculateur, dans le cadre d'un ascenseur électromécanique, désigne une unité électronique programmée (souvent un microcontrôleur ou microprocesseur) chargée de recevoir, traiter et exécuter les commandes liées au mouvement, à la sécurité et à l'affichage de l'ascenseur.

Avant d'étudier l'architecture détaillée d'un micro-processeur précis, on aborde ici l'architecture globale d'un micro-processeur. Tous les micro-processeurs conçus jusqu'à aujourd'hui s'organisent globalement de même façon.

Un système à base de microprocesseur est formé des trois éléments :

- ✓ Une unité CPU (Central Processing Unit).
- ✓ Une mémoire (ROM et RAM)
- ✓ Interfaces (Des ports d'entrées/sorties).

On peut ajouter une quatrième (4eme.) élément – **selon le Modèle de von Neumann** – c'est les bus qui assurent la communication interne et externe du microprocesseur ; bus de données, bus d'adresses et bus de contrôles et commandes

Ces trois modules sont interconnectés comme le montre la figure suivante.

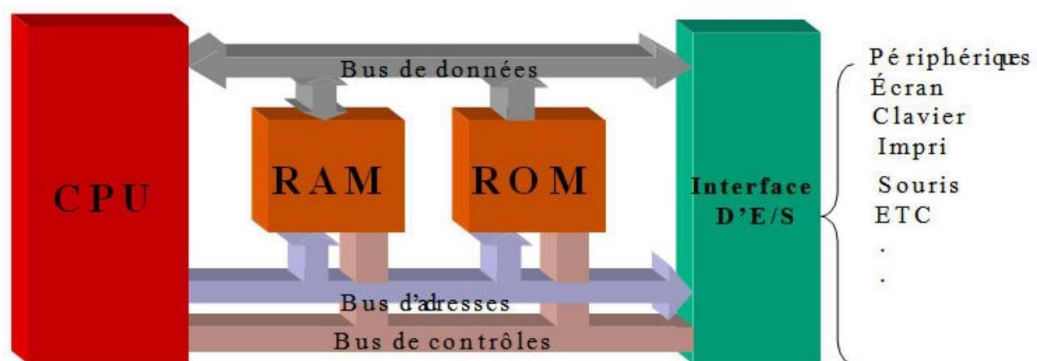


Figure II.2 : Architecture d'un système à base de microprocesseur.

6.1. Définition d'un microprocesseur :

Le microprocesseur, noté aussi C.P.U. (Central Processing Unit) est un circuit intégré complexe appartenant à la famille des VLSI (Very Large Scale Intégration) capable d'effectuer séquentiellement et automatiquement des suites d'opérations élémentaires.

6.1.1. Architecture d'un Microprocesseur CPU :

Un microprocesseur (CPU) est formé par les trois éléments fonctionnels interconnectés suivants :

- **Registres** : Un registre est un emplacement de mémoire interne à un processeur où les données sont temporairement stockées.
- **UAL : Unité arithmétique et logique** : Comme son nom l'indique, cette unité peut exécuter deux types d'opérations. Opérations arithmétiques et Opérations logiques.
- **Circuit de contrôle et commande** : C'est la partie commande du microprocesseur. Elle permet de séquencer le déroulement des instructions ; elle effectue la recherche en mémoire de l'instruction

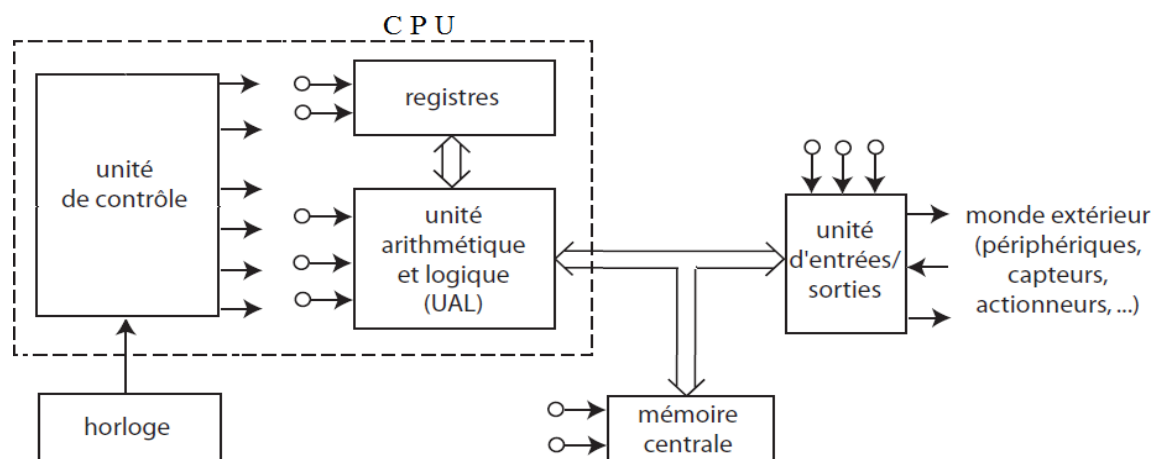


Figure II.3 : Architecture d'un microprocesseur.

6.2. Définition d'un Microcontrôleur :

Un microcontrôleur est un circuit intégré qui combine un processeur, de la mémoire et des interfaces d'entrées/sorties (E/S) pour gérer des tâches spécifiques dans des applications embarquées.

6.2.1. Structure interne d'un microcontrôleur :

Le microcontrôleur utilise ces composants pour exécuter des tâches complexes. La figure II.4, illustre les blocs de base sur lesquels repose le fonctionnement d'un microcontrôleur. Chacun de ces éléments a une fonction spécifique, et ensemble, ils permettent au microcontrôleur :

- D'exécuter des tâches variées,
- De traiter des données,
- De communiquer avec d'autres systèmes (capteurs, actionneurs, autres microcontrôleurs, modules de communication sans fil...)
- De contrôler des appareils externes.

Dans la suite, une explication simple de l'architecture interne d'un microcontrôleur, avec des descriptions des différents composants illustrés dans la figure :

1. **Oscillateurs** : Ce sont des composants essentiels qui fournissent l'horloge nécessaire au fonctionnement du système.
2. **Mémoire** : La mémoire stocke les informations et les programmes que le microcontrôleur utilise. RAM (Random Access Memory), ROM (Read-Only Memory).
3. **Ports (I/O)** : Ce sont des points de connexion pour que le microcontrôleur puisse communiquer avec l'extérieur.
4. **PWM (Pulse Width Modulation)** : C'est une technique utilisée pour contrôler la vitesse des moteurs ou la luminosité des voyants en modulant la durée d'un signal électrique.
5. **ADC/DAC (Convertisseur Analogique/Numérique)** :
 - ADC : Convertit un signal analogique (comme la température) en un signal numérique que le microcontrôleur peut comprendre.
 - DAC : Convertit un signal numérique en un signal analogique.
6. **Timers (Temporisateurs)/Counters (Compteurs)** :
 - Les temporisateurs sont utilisés pour créer des délais dans les programmes ou pour générer des signaux périodiques.
 - Les compteurs sont utilisés pour compter des événements.
7. **Microprocesseur (CPU)** : Le microprocesseur est le "cerveau" du microcontrôleur. Il exécute le programme, effectue des calculs et prend des décisions.
8. **UART/USART/SPI/I2C/USB** : Ce sont des protocoles de communication qui permettent au microcontrôleur d'interagir avec d'autres appareils.
 - UART/USART : Permet d'envoyer et recevoir des données série.
 - SPI et I2C : Permettent de connecter plusieurs appareils ensemble.

- **USB** : Utilisé pour connecter des appareils comme des claviers ou des souris.

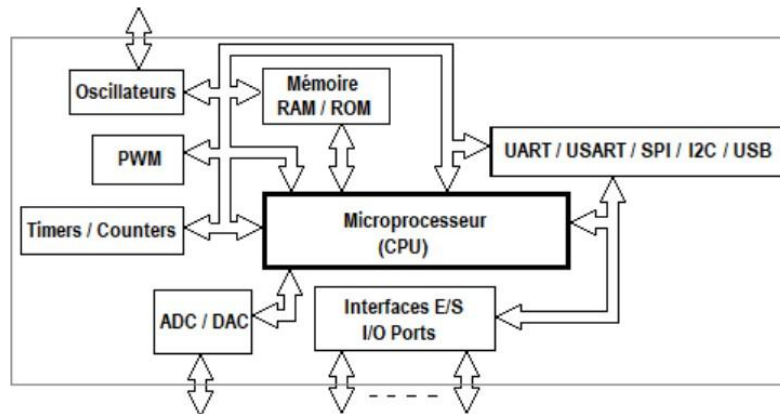


Figure II.4 : Eléments principaux d'un microcontrôleur.

Langages de programmation Les microcontrôleurs peuvent être programmés dans plusieurs langages, les plus courants étant :

- **C** : Le langage C est largement utilisé en raison de sa portabilité et de sa capacité à interagir directement avec le matériel.
- **Assembleur** : Bien que plus complexe, la programmation en assembleur permet un contrôle précis des ressources matérielles et est souvent utilisée pour des applications critiques en termes de performance.
- **C++** : Utilisé dans certains environnements, notamment pour les projets plus complexes qui nécessitent une programmation orientée objet.

6.2.2. Types de boîtiers de microcontrôleurs :

Les microcontrôleurs existent dans plusieurs types de boîtiers, chacun ayant des caractéristiques qui les rendent plus ou moins adaptés à certaines applications :

DIP (Dual Inline Package) **SOIC** (Small Outline Integrated Circuit), **QFP** (Quad Flat Package) / **TQFP** (Thin QFP), **MLP** (Micro Lead Frame Package), **BGA** (Ball Grid Array), **PGA** (Pin Grid Array).

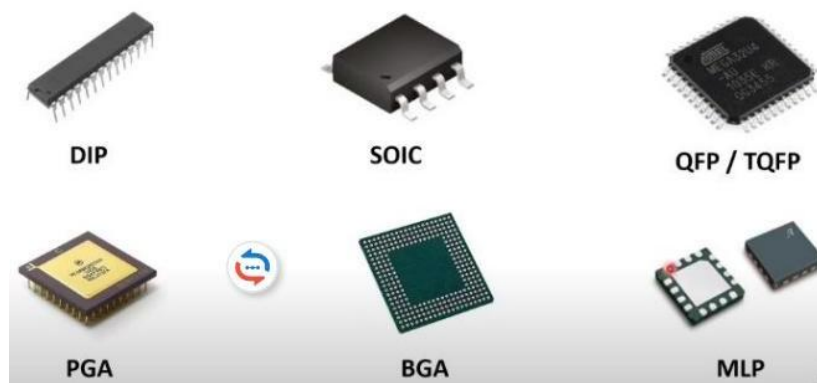


Figure II.5 : Différents types de boîtiers de microcontrôleurs

6.3. Structure des composants de système :



Figure II.6: Structure des composants de système

II.7. Analyse structurelle des systèmes :

L'ascenseur constitue un système automatisé capable de s'adapter à son environnement et de répondre aux instructions de ses utilisateurs. Il convertit l'énergie électrique en énergie mécanique pour assurer le déplacement de la cabine. Afin de connaître la position et la destination de celle-ci, il intègre une chaîne d'information (partie commande) qui garantit le bon déroulement des opérations : pilotage du mouvement, arrêt précis de la cabine, et gestion de l'ouverture et de la fermeture des portes[2].

L'ascenseur est constitué de deux parties :

II.7.1. Chaîne d'information :

Elle pilote la chaîne d'énergie et prend les décisions en fonction des tâches à réaliser et des données reçues. On distingue trois blocs fonctionnels :

a. **Acquérir :**

Partie du système automatisé chargée de capter l'information à l'aide de capteurs.

b. **Traiter :**

Partie commande, constituée d'un automate ou d'un microcontrôleur, qui analyse et décide.

c. **Communiquer :** Gestion des échanges d'informations selon le programme de la partie commande :

- ✓ L'opérateur transmet des consignes à la partie commande.
- ✓ La partie commande émet des ordres vers la partie opérative.
- ✓ Les actionneurs reçoivent ces ordres et produisent un effet physique.
- ✓ Les capteurs détectent les variations d'état et fournissent un retour à la partie commande.
- ✓ La partie opérative adresse des comptes-rendus à la partie commande.
- ✓ La partie commande informe l'opérateur de l'état du système ou de son environnement.[28]

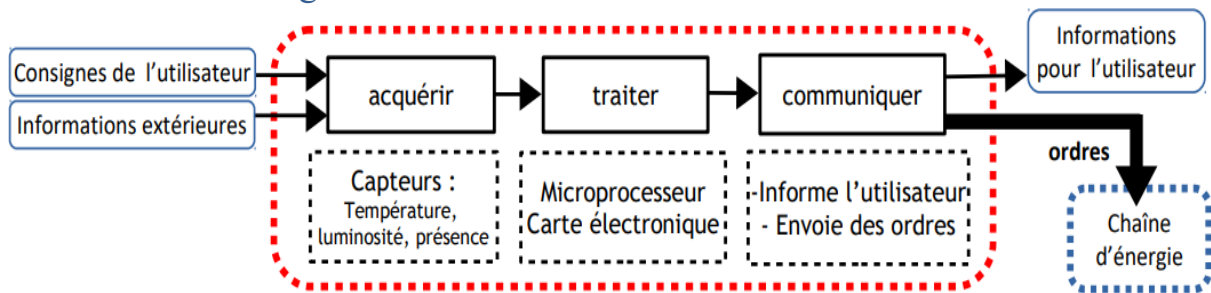


Figure II.7 : chaîne d'information.

II.7.2. Chaîne d'énergie :

Qui gère et contrôle l'énergie disponible destinée à faire réaliser la ou les actions du système en fonction des ordres reçus de la chaîne d'informations. Chaque objet technique a besoin d'une énergie pour fonctionner, il n'est pas possible en général de l'utiliser dans sa forme initiale, il faut donc la stocker, réaliser sa commande d'utilisation, la transporter aux différents éléments, la convertir, l'utiliser. Les composants internes de cet objet peuvent être représentés sous forme de blocs ayant chacun une fonction élémentaire qui agit sur l'énergie[28].

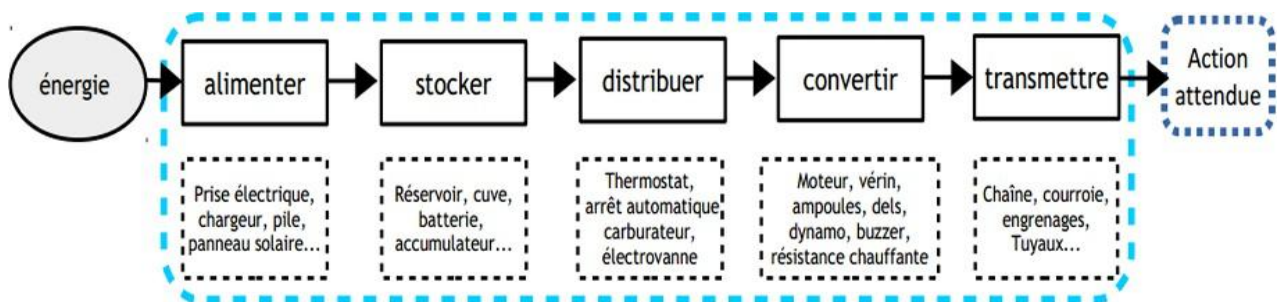


Figure II.8: Chaîne d'énergie d'ascenseur.

- **Alimenter** : apporter l'énergie nécessaire au système.
- **Stocker** : certains objets doivent pouvoir stocker de l'énergie en vue de la restituer.
- **Distribuer** : commander, contrôler et répartir la quantité d'énergie nécessaire.
- **Convertir** : transformer une énergie en une autre nécessaire à l'effet attendu.
- **Transmettre** : transporter l'énergie d'un lieu à un autre[12].

II.7.3. Mise en situation :

La cabine de l'ascenseur doit se déplacer d'étage en étage en fonction des demandes des utilisateurs, le déplacement de la cabine dépend de sa position initiale et de la demande. C'est l'unité de commande qui gère l'ensemble, et établit les liens entre les différentes parties de ce dernier. Elle commande la montée ou la descente de la cabine en gérant les désirs des utilisateurs, le démarrage et l'arrêt du moteur, l'ouverture et la fermeture de la porte cabine[2].

II.7.4. Démarche de conception :

Les processus de conception standard sont devenus incapables de répondre aux normes de sécurité de fonctionnement, de qualité de, confort et de performance. Aujourd'hui la conception d'ascenseurs est de plus en plus complexe, et demande l'utilisation de nouvelles démarches, qui sont basées sur des technologies et un savoir-faire, faisant apparaître la nécessité d'une analyse spécifique des systèmes complexes critiques. Cette démarche nécessite l'utilisation de nouveaux outils de travail.

A l'origine de tout projet d'installation d'ascenseur, il y a bien évidemment la demande d'un client. Chacune d'entre elle est différente et donnera donc lieu à un projet unique qui déterminera les contraintes et impératifs qui en découlent. Bien définir son projet est primordial pour la conception, pour ce faire, on commence par mettre au clair les besoins et les souhaits des usagers consignés dans un cahier des charges, puis on détermine l'ensemble des contraintes techniques à respecter, de l'appareil (taille de la cabine, emplacement des portes palières ...), en s'appuyant sur les plans, les observations réalisées et l'ensemble des documents disponibles[2].

II.8. Conclusion :

Au terme de l'analyse des exigences fonctionnelles, des contraintes techniques et du cadre pédagogique du projet, le choix de l'Arduino UNO s'est imposé comme une solution à la fois pragmatique, économique et techniquement suffisante pour assurer la commande d'un ascenseur électromécanique simulé.

Bien que cette carte présente certaines limites en termes de mémoire, de robustesse industrielle et de capacités multitâches, ces dernières sont largement compensées par ses nombreux avantages : simplicité de programmation, compatibilité avec les logiciels de simulation, coût réduit, et richesse de la documentation disponible.

Ce choix répond pleinement aux objectifs pédagogiques du projet, tout en permettant la mise en œuvre fiable de la logique de commande de base d'un ascenseur à travers un prototype fonctionnel et évolutif. Il constitue ainsi une base solide pour valider les principes fondamentaux de l'automatisation électromécanique dans un contexte académique.

Nous avons aussi exposé l'objectif du projet, qui est basé sur les besoins et les désirs des utilisateurs, en présentant les processus de conception. En raison de leur complexité, il est crucial d'utiliser de nouvelles méthodes, avec des analyses spécifiques nécessitant l'utilisation de nouveaux outils.

Chapter III:

Simulation et Programmation

III.1. Introduction

Dans ce chapitre, nous expliquons en détail la méthodologie de simulation et de programmation qui a été mise en œuvre pour le contrôle de l'ascenseur électromécanique. Ce chapitre illustre comment, grâce à l'utilisation d'outils de simulation (Proteus) et de la carte Arduino, nous avons pu concevoir, valider et affiner la logique de contrôle du système.

La carte électronique programmable sert à faciliter l'interfaçage avec le monde extérieur en assurant la communication entre les données (entrées) et les commandes (sorties).

Dans ce chapitre, nous commencerons par présenter une description générale des éléments de base qui composent la carte de commande ainsi que les fonctionnalités offertes, puis nous examinerons, à l'aide d'une simulation, les différentes parties du système et les organigrammes des divers programmes.

III.2 Partie logiciel « Software » :

Avant la réalisation pratique de nos cartes, nous ferons la conception et la simulation de ces dernières. Pour ce faire nous allons utiliser **le logiciel Proteus**.

PROTEUS est un outil qui permet de dessiner des schémas électroniques, de les simuler et de réaliser le circuit imprimé correspondant

III.2.1. Présentation de Proteus :

Cette suite logicielle est très connue dans le domaine de l'électronique. Elle est utilisée dans de nombreuses entreprises et organismes de formation. Outre la popularité de l'outil, Proteus possède d'autres avantages :

- Pack contenant des logiciels faciles et rapides à comprendre et à utiliser ;
- Support technique performant ;
- Outil de création de prototype virtuel permettant de réduire les coûts matériel et logiciel lors de la conception d'un projet ;

Le logiciel comprend deux principaux modules : ISIS et ARES[11].

- **ISIS**

Le logiciel ISIS de Proteus est principalement connu pour éditer des schémas électriques. Par ailleurs, le logiciel permet également de simuler ces schémas, ce qui permet de déceler certaines erreurs dès l'étape de conception. Indirectement, les circuits électriques conçus grâce à ce logiciel peuvent être utilisés dans des documentations car le logiciel permet de contrôler la majorité de l'aspect graphique des circuits[11].

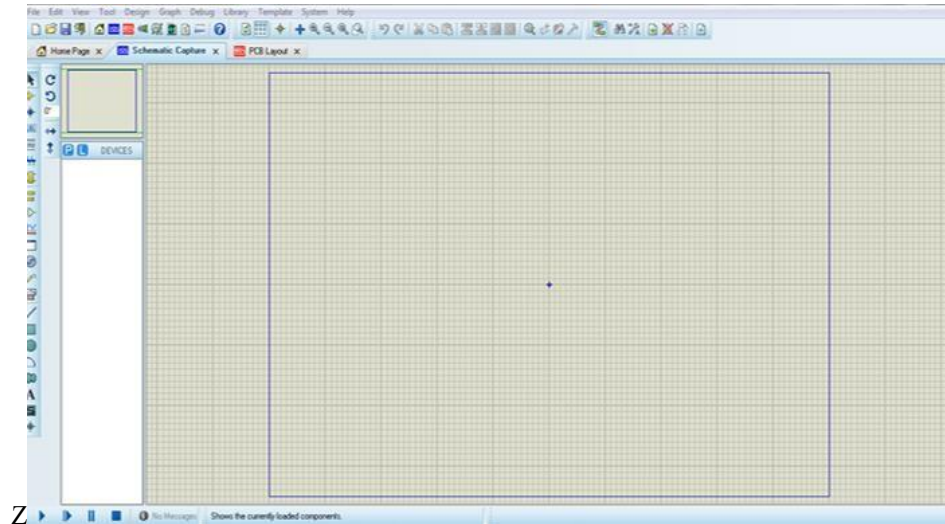


Figure III.1: Fenêtre du module de simulation Proteus ISIS.

- **ARES**

Le module ARES est un outil d'édition et de routage qui complète parfaitement ISIS. Un schéma électrique réalisé sur ISIS peut alors être importé facilement sur ARES pour réaliser le PCB de la carte électronique. Bien que l'édition d'un circuit imprimé soit plus efficace lorsqu'elle est réalisée manuellement, ce logiciel permet de placer automatiquement les composants et de réaliser le routage automatiquement. La figure ci-dessous montre la fenêtre du module de simulation Proteus ARES[17].

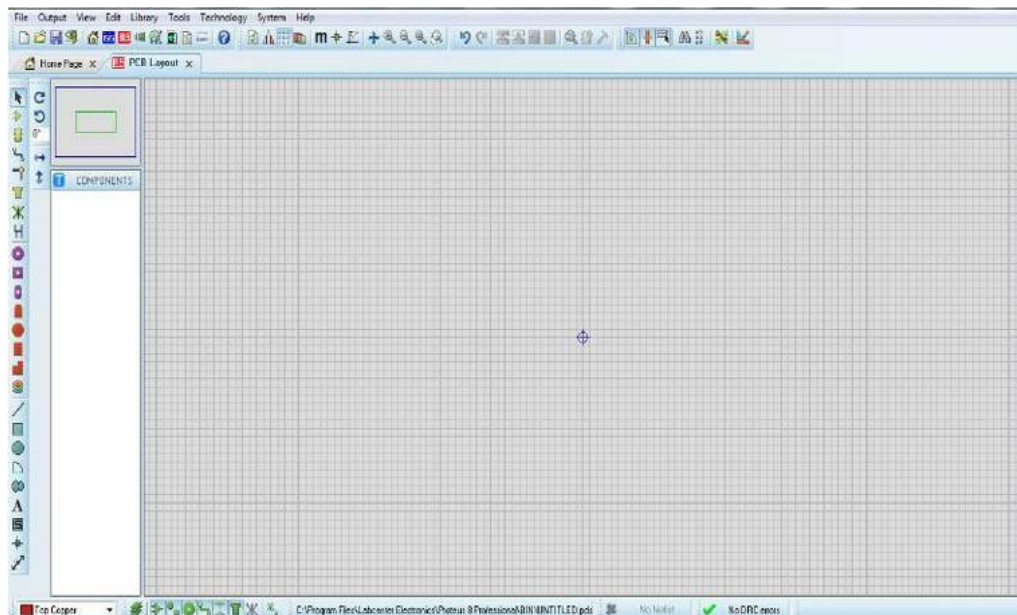


Figure III.2: Fenêtre du module de simulation Proteus ARES.

III.2.2. L'environnement de la programmation (Logiciel IDE) :

III.2.2.1. Description :

L'IDE est un logiciel de programmation qui permet d'écrire, de modifier un programme et de le convertir en une série d'instructions compréhensibles pour la carte. Il programme par code, contenant une cinquantaine de commandes différentes. A l'ouverture, l'interface visuelle du logiciel contient le menu, des boutons de commande en haut, une page blanche vierge, une bande noire en bas[21] :

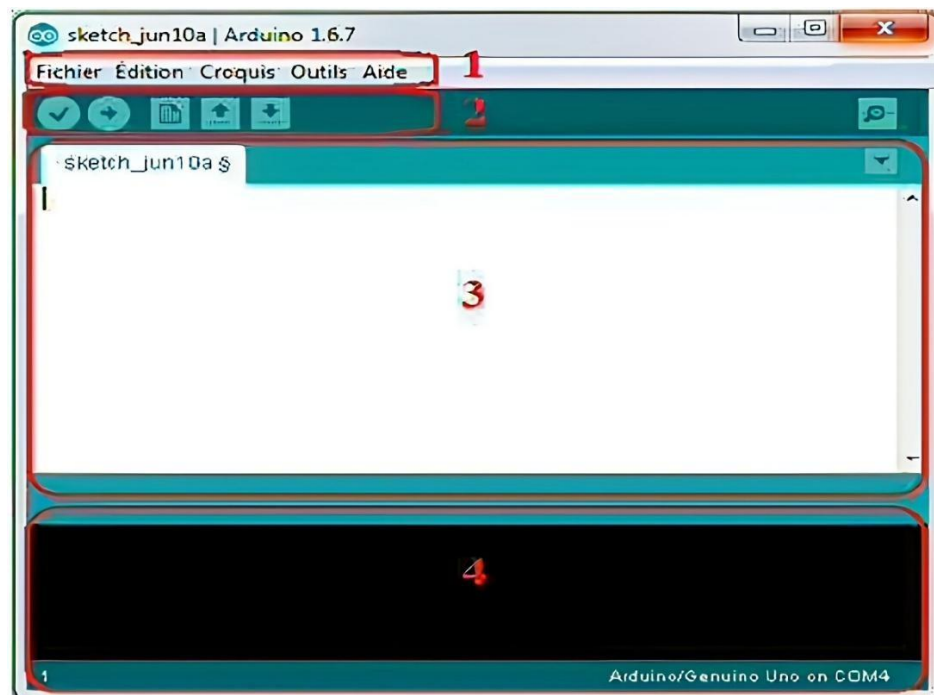


Figure III.3: Arduino IDE.

1. Barre de menu :

Les différents éléments du menu permettent de créer de nouveaux sketches (programmes), de les sauvegarder, et de gérer les préférences du logiciel et les paramètres de communication avec votre carte Arduino. Le menu comprend :

- **Fichier** : Pour créer, sauvegarder en spécifiant la destination, et d'appeler un programme.
- **Édition** : Pour couper, copier, coller, supprimer, sélectionner,...etc.
- **Croquis** : Regroupe les fichiers réalisés.
- **Outils** : Pour spécifier le type de la carte, le port série, formater, recharger et réparer l'encodage, graver la séquence d'initialisation, de la carte branchée sur l'ordinateur.
- **Aide** : Pour donner de l'aide concernant les différents problèmes rencontrés au niveau du logiciel Arduino.

2. Barre de Bouton:



Figure III.4: Barre d'Action.

- **Vérifier/compiler** : Ce bouton permet de vérifier les erreurs dans le programme.
- **Téléverser** : Compiler le code et envoyer vers la carte Arduino.
- **Nouveau** : Créer un nouveau fichier.
- **Ouvrir** : Il suffit de cliquer sur ce bouton pour ouvrir des programmes déjà présents.
- **Sauvegarder** : Sauvegarder le programme.
- **Moniteur série** : Ce bouton permet d'accéder au port série [21].

3. Fenêtre de Programmation:

Est l'éditeur où s'écrit le programme, chaque logiciel obéit à quelques notions pour pouvoir bien structurer le programme à fin de le compiler et éviter les erreurs de syntaxe et autres.

4. Barre des erreurs :

La barre des erreurs affiche les erreurs faites au cours du programme, comme l'oubli d'un point virgule, le manque d'une accolade ou toute autre erreur dans les instructions

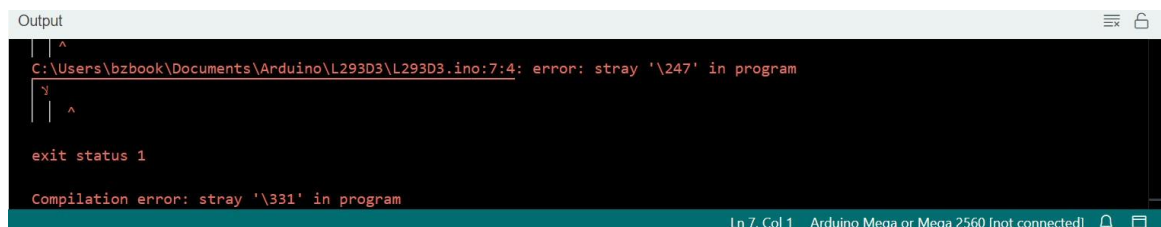


Figure III.5: Message d'erreur.

Compilation et Téléversement :

Une fois le programme terminé et vérifié, on passe au test, en appuyant sur le bouton Compiler, une barre de progression s'affiche au-dessus de la barre des erreurs, s'il n'y a pas d'erreur, on verra s'afficher le message « Compilation terminée », suivi de la taille du programme. Dans le cas contraire, un message d'erreur s'affiche[21].

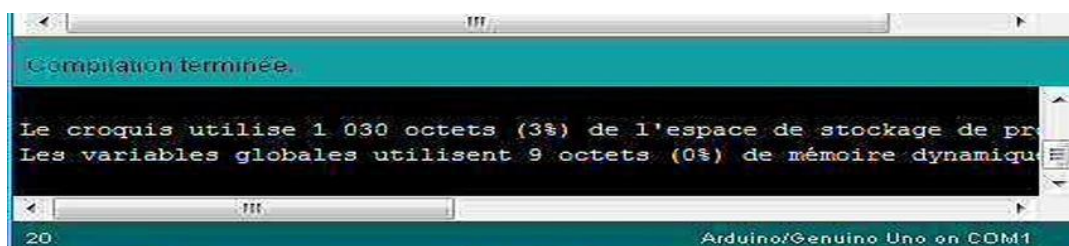


Figure III.6: Compilation.

III.2.2.2. Le Langage de programmation :

Comme nous l'avons mentionné, le langage de l'IDE Arduino est un mélange entre le C et le C++, il possède un jeu d'instruction très riche, Ces instructions décrivent :

- **Déclaration et configuration** : grâce à `pinMode(pin, INPUT_PULLUP)` et `pinMode(pin, OUTPUT)`, vous définissez en un seul geste l'usage de chaque broche, qu'il s'agisse du bouton d'arrêt d'urgence ou des sorties du driver L293D.
- **Lecture des entrées** : avec `digitalRead(pin)` et `analogRead(A0)`, vous captez instantanément l'état des capteurs de position, des boutons d'appel et du capteur de poids.
- **Pilotage des actionneurs** : `digitalWrite(MOTOR_IN1, HIGH/LOW)` et `analogWrite(MOTOR_EN, PWM)` permettent de commander la direction et la vitesse du moteur 12 V en toute finesse.
- **Maîtrise du temps** : la fonction `delay(ms)` rythme les phases d'arrêt et de déplacement, assurant des temporisations strictes entre chaque étage.
- **Communication et débogage** : `Serial.begin(9600)` et `Serial.print()` nous ont ouverts les coulisses de la simulation Proteus, rendant visible chaque valeur et chaque transition d'état[21].

III.3. Architecture Fonctionnelle De La Carte De Commande

La carte de commande est l'élément de base d'un ascenseur. Elle assure le contrôle du fonctionnement et optimise le déplacement de la cabine, tout en préservant la sécurité des personnes. on voit qu'elle est à la fois récepteur d'informations et transmetteur d'ordres. Elle a donc pour rôle de surveiller l'état des capteurs présents sur la cabine afin de donner les différentes options de contrôle telles que la marche, l'arrêt, la vitesse, le sens de déplacement et l'arrivée / départ des étages. Des dispositifs de fermeture des portes, éclairage, affichages, bips sonores peuvent également être traités.

III.3.1. Définition précise du contrôleur Arduino Uno :

L'Arduino Uno est une carte de microcontrôleur open-source basée sur le microcontrôleur ATmega328P d'architecture AVR 8 bits. Elle intègre un oscillateur à quartz de 16 MHz, 14 broches d'entrées/sorties numériques (dont 6 pouvant être utilisées en sortie PWM) et 6 entrées analogiques. La carte dispose également d'une interface USB facilitant la programmation et l'alimentation, ainsi que d'un système de régulation de tension pour offrir une stabilité en alimentation. Conçue pour la simplicité d'utilisation, l'Arduino Uno s'appuie sur un

Chapitre III : Simulation et programmation.

environnement de développement intégré (Arduino IDE) convivial et sur une vaste bibliothèque logicielle, ce qui en fait une plateforme de choix pour la réalisation de prototypes, d'applications éducatives et de projets de recherche nécessitant un compromis optimal entre performances, coût et flexibilité[21].

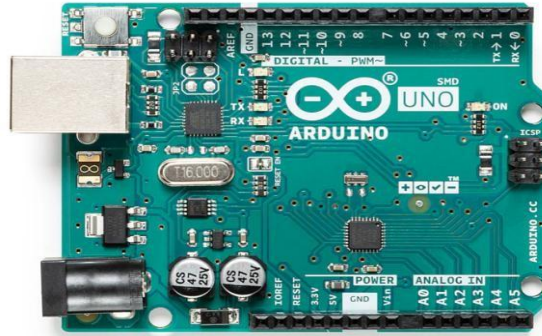


Figure III.7: Arduino Uno.

III.3.2. Description des différentes parties de la carte de commande :

Le système de contrôle des ascenseurs se compose d'entrées et de sorties et d'un panneau de contrôle qui fonctionne entre eux pour recevoir et exécuter les commandes.

III.3.2.1. Choix du microcontrôleur « arduino uno »

Toute la difficulté du choix d'un microcontrôleur pour une application donnée réside dans la sélection du "bon" circuit adapté pour cette application, les critères principaux que nous devons retenir pour choisir un microcontrôleur sont :

- Leur capacité de gérer un grand nombre de fonctionnalités variées,
- Leur architecture flexible.
- Le nombre des Ports d'E/S.
- L'existence d'interfaces d'E/S analogiques, interfaces de communications série et parallèle.

On peut dire que le choix est toujours le résultat d'un compromis entre : performances et coûts, efficacité et facilité de construction, et performances d'ensemble et facilité de programmation.

III.3.2.2. Broches de La carte Arduino Uno :

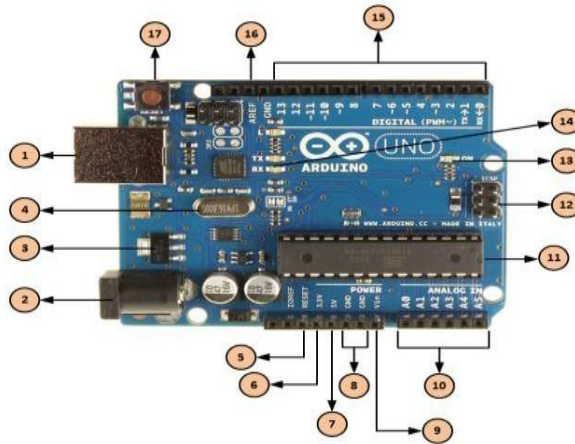


Figure III.8: Différents broches de La carte Arduino Uno.

Tableau III.1: les différentes broches de la carte ARDUINO UNO[21]

1	Alimentation / Programmation par USB L'Arduino se recharge via un câble USB connecté à votre ordinateur. Il suffit simplement de relier votre carte Arduino à l'ordinateur en utilisant un câble USB de type A/B.
2	Alimentation via connecteur Jack DC Diamètre interne 2.1mm, externe 5.5mm La carte Arduino peut être alimentée directement à travers ce connecteur Jack DC. Ce port (2) est connecté au régulateur de tension intégré sur la carte. Pour une alimentation par ce connecteur (2), la tension doit se situer entre 5 et 12 V.
3	Régulateur de tension Le régulateur de tension (3) a pour rôle de maîtriser la tension d'alimentation de l'Arduino afin de la stabiliser au voltage approprié pour le microcontrôleur et les composants de la carte. Sur les cartes UNO, cette tension de stabilisation est fixée à 5 Volts.
4	Oscillateur à quartz Un oscillateur à quartz est un composant électronique qui contient un quartz vibrant sous l'effet piézoélectrique. Les caractéristiques électromécaniques du quartz permettent d'obtenir une vibration à une fréquence extrêmement précise. Ce composant permet à l'Arduino UNO de mesurer les données temporelles. L'inscription "16.000H9H" visible sur le dessus du composant indique une fréquence de 16,000,000 Hertz, équivalant à 16 MHz.

5.17	Arduino Reset Vous pouvez réinitialiser un Arduino par un "Reset". Cette action redémarrera votre programme depuis le début. Deux méthodes permettent de réinitialiser l'Arduino UNO : soit en appuyant sur le bouton "Reset" (17), soit en branchant un bouton externe sur la broche "RESET" (5) de la carte Arduino.
6.7.8.9	Broches (3.3, 5, GND, Vin) 3.3V (6) – Broche d'alimentation fournissant une tension de 3.3 Volts 5V (7) – Broche d'alimentation délivrant une tension de 5 Volts La majorité des composants conçus pour Arduino fonctionnent correctement en 3.3 Volts ou 5 Volts. GND (8) (Ground / Masse) – Plusieurs broches de ce type sont disponibles sur la carte Arduino, toutes sont identiques et utilisables comme masse (potentiel 0 Volts) pour vos montages. Vin (9) – Cette broche permet d'alimenter l'Arduino depuis une source externe de tension. Elle est connectée au circuit d'alimentation principal de la carte Arduino.
10	Broches analogiques

	L'Arduino UNO dispose de 5 broches d'entrées analogiques, numérotées de A0 à A5. Ces broches permettent la lecture de signaux analogiques provenant de capteurs tels que les capteurs d'humidité ou de température. La carte Arduino intègre un convertisseur analogique/numérique (convertisseur CAN) qui rend possible l'interprétation du signal par le microcontrôleur. La conversion s'effectue sur 10 bits, permettant une lecture sur une échelle de 1024 points.
11	Microcontrôleur principal Chaque carte Arduino est équipée de son propre microcontrôleur (11). Ce composant peut être considéré comme le cerveau de la carte Arduino. Le microcontrôleur varie légèrement d'une carte Arduino à l'autre. Ces microcontrôleurs sont généralement fabriqués par la société ATMEL. Il est nécessaire d'identifier le microcontrôleur de votre carte avant de télécharger un nouveau programme depuis l'IDE Arduino. Cette information est inscrite directement sur le composant. Pour des informations plus complètes sur la structure et les fonctionnalités du microcontrôleur, consultez la fiche technique (data sheet).
12	Le connecteur ICSP (In-Circuit Serial Programming) Est une interface utilisée pour la programmation des microcontrôleurs AVR. Il regroupe les broches MOSI, MISO, SCK, RESET, VCC et GND. Principalement, ce connecteur sert à programmer directement le microcontrôleur au niveau des couches les plus basses, comme le bootloader ou du code en langage assembleur (ASM). En plus de la programmation, ce connecteur fonctionne également comme un port SPI (Serial Peripheral Interface), permettant la communication avec d'autres périphériques SPI tels que des écrans, des capteurs, etc.

13	Indicateur LED d'alimentation Ce voyant LED s'allume dès que l'Arduino est connecté à une source d'alimentation, signalant ainsi que la carte est correctement alimentée. Si cette lumière reste éteinte, cela indique un problème avec l'alimentation.
14	LEDs TX et RX Sur la carte Arduino UNO, deux voyants sont présents : TX (transmission) et RX (réception). Ils sont visibles à deux endroits sur la carte. Les broches numériques 0 et 1 sont associées à ces LEDs et indiquent les broches utilisées pour la communication série. Le voyant TX clignote à une vitesse variable lors de l'envoi de données série, cette vitesse de clignotement dépendant du débit de transmission. De son côté, le voyant RX clignote pendant la réception des données.
15	Entrées/Sorties numériques La carte Arduino UNO dispose de 14 broches numériques pouvant servir d'entrées ou de sorties. Parmi elles, 6 broches sont capables de générer un signal PWM (Pulse Width Modulation). Ces broches peuvent être configurées en mode entrée pour lire des valeurs logiques (0 ou 1) ou numériques, ou en mode sortie
	pour contrôler différents dispositifs tels que des LEDs, des relais, etc. Les broches marquées d'un symbole "~" indiquent leur capacité à produire un signal PWM.
16	Broche AREF AREF, qui signifie "référence analogique", est une broche utilisée pour définir une tension de référence externe, généralement comprise entre 0 et 5 Volts. Cette tension sert de limite supérieure pour les mesures effectuées par les broches d'entrée analogiques.

III.3.2.3. Caractéristique de base du Arduino Uno :

Microcontrôleur : ATmega328P (architecture AVR 8 bits).

Fréquence d'horloge: 16 MHz

Mémoire :

- Flash : 32 KB (dont 0,5 KB utilisés par le bootloader).
- SRAM : 2 KB.
- EEPROM : 1 KB.

Broches d'E/S numériques : 14 broches (avec 6 sorties PWM).

Entrées analogiques : 6 broches.

Interfaces de communication :

- USB (pour la programmation et la communication série).
- UART (Serial).
- I2C.
- SPI.

Alimentation :

- Tension de fonctionnement : 5 V.
- Tension d'entrée recommandée : 7-12 V via l'alimentation externe.

Connectivité : Port USB-B pour la programmation et l'alimentation.

Caractéristiques supplémentaires :

- Régulateur de tension intégré.
- Compatible avec une large gamme de shields et bibliothèques open- source.

III.4. Programmation :

La programmation de notre projet s'est déroulée de manière itérative et méthodique en suivant ces étapes :

• Rédaction et structuration du code

- La programmation de notre application consiste à écrire en langage C l'algorithme de chaque proces sus cité ci-dessus puis à les combiner séquentiellement de manière à orchestrer l'atteinte de l'objectif final dans les performances désirées. L'environnement de développement (IDE) de l'arduino est très convivial et très facile d'utilisation. D'autant plus il propose plusieurs exemples de fichiers sources que nous pouvons modifier et adapter à notre besoin. Sa bibliothèque est très riche dans la mesure où elle est open source, sa grande communauté l'enrichit de jours en jours. Alors nous gagnons un temps important dans le développement d'une application.
- L'utilisation de bibliothèques comme LiquidCrystal_I2C nous a permis de simplifier la gestion de l'affichage via le module PCF8574.

• Utilisation d'Arduino IDE :

- Arduino IDE nous a fourni un environnement de développement intégré convivial et open-source pour écrire, compiler et téléverser notre code sur la carte Arduino Uno.
- Grâce à son interface intuitive, nous avons pu intégrer facilement des

Chapitre III : Simulation et programmation.

exemples et des bibliothèques, ce qui a accéléré le développement et le débogage grâce au moniteur série[21].

- **Simulation avec Proteus :**

- Une fois le code compilé, nous avons exporté le fichier.hex généré par Arduino IDE et l'avons chargé dans Proteus pour simuler l'ensemble du système.
- Cette simulation nous a permis de vérifier en temps réel le comportement de l'ascenseur (déplacement du moteur, lecture des capteurs, affichage sur le LCD) avant de passer au montage physique, garantissant ainsi que la logique de commande fonctionnait comme prévu.

III.4.1. Structure du Programme :

L'ordre chronologique de déroulement des actions à effectuer est établi dans le chronogramme suivant :

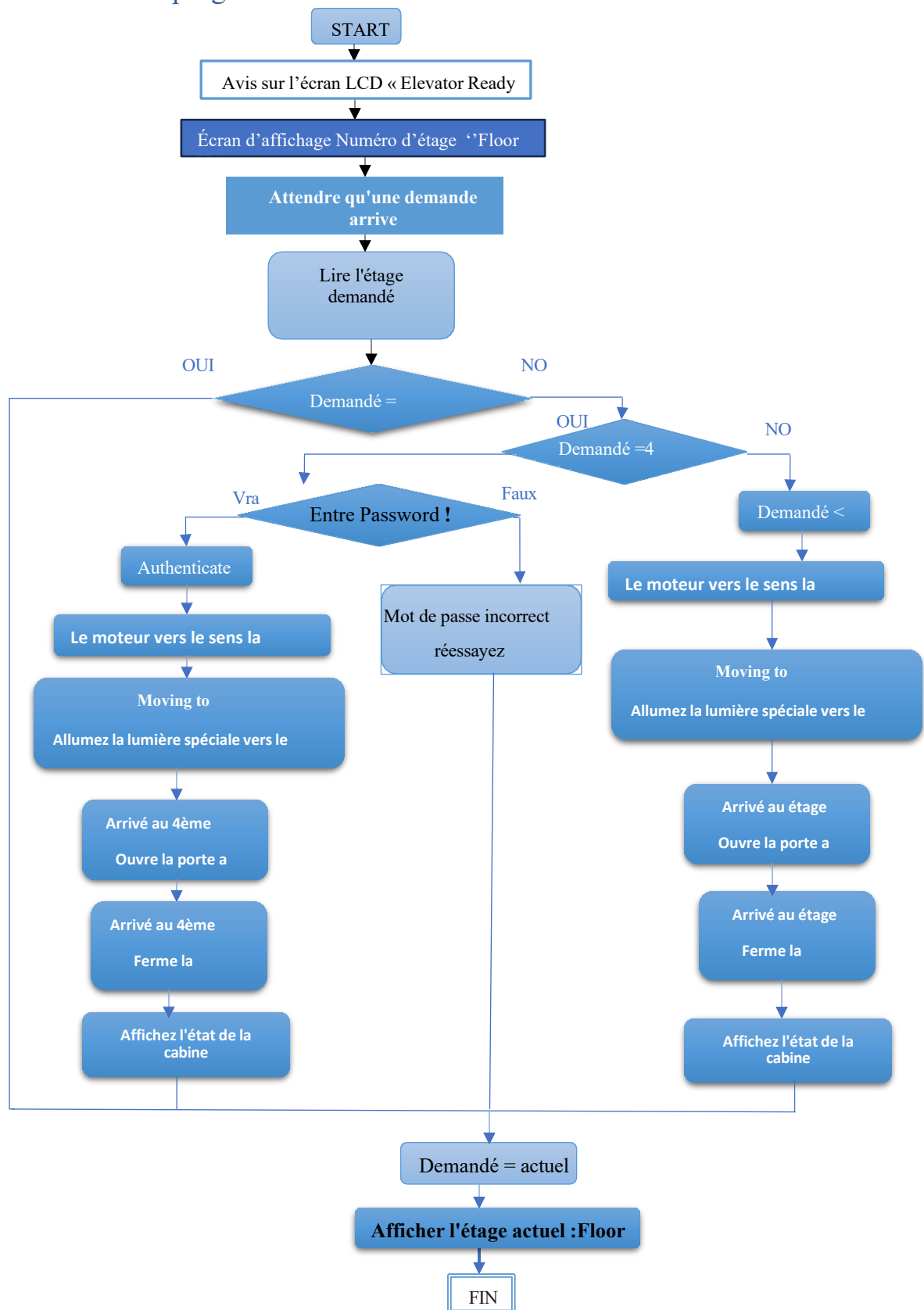


Figure III.9: Structure du programme

III.4.2. L'organisation du programme :

Le programme a été conçu de sorte à obtenir une fonction principale la plus simple possible. Le programme a été partagé en un ensemble de fonctions possibles afin de faciliter la lecture et la compréhension. En effet, en attribuant à chaque sous-programme l'exécution d'une tâche spécifique, nous avons pu obtenir un programme simple, structuré et clair.

III.4.2.1. Sous-Programme Moteur :

III.4.2.1.1. Définition Le Moteur A Courant Continue

Un moteur à courant continu est constitué de deux parties : le stator et le rotor. Lorsqu'on alimente le moteur, il se crée une interaction magnétique qui met le moteur en mouvement. Lorsqu'on bascule le sens de la tension qui alimente le moteur, il tourne en sens inverse

Est un dispositif électromécanique qui convertit l'énergie électrique en énergie mécanique (rotation) lorsqu'il est alimenté par un courant continu[17].



Figure III.10: Moteur à courant continu.

III.4.2.1.2 Caractéristiques Principales :

- Alimentation en courant continu (batterie ou alimentation DC).
- Contrôle simple de la vitesse : la variation de tension ou l'utilisation d'un signal PWM (Pulse Width Modulation) permettent de réguler la vitesse.
- Inversion de sens de rotation : en inversant la polarité (les bornes + et -), on peut changer le sens de rotation du moteur.

III.4.2.1.3. Choix du moteur à courant continu :

Le moteur à courant continu est un choix pertinent pour un ascenseur d'étude ou un prototype grâce à sa simplicité de commande, son coût abordable, et sa grande disponibilité sur le marché. Pour un usage réel à grande échelle (ascenseurs d'immeubles), on utilise cependant des moteurs plus robustes (asynchrones triphasés avec variateur), mais pour un projet didactique ou de démonstration, le moteur DC reste l'option la plus accessible.

- **Inversion de sens :** Pour faire monter ou descendre la cabine, il suffit d'inverser la

- **Polarité** (ou via un pont en H comme le L293D).
- **Réglage de la vitesse** : Un signal PWM (modulation de largeur d'impulsion) appliqué sur l'alimentation du moteur permet de contrôler la vitesse de façon simple.
- **Les moteurs DC sont très répandus** et se trouvent dans une large gamme de puissances et de vitesses.
- Ils sont généralement moins coûteux que d'autres types de moteurs (p. ex., moteurs pas à pas ou moteurs synchrones) pour des applications de base.
- **Circuit de commande** : Un seul driver (comme le L293D ou L298N) suffit pour gérer la montée/descente (deux sens de rotation).
- **Adaptabilité** : On peut utiliser un potentiomètre (ou un capteur de rétroaction) pour ajuster ou surveiller la vitesse, ou encore coupler un codeur (si besoin de précision supplémentaire).

III.4.2.1.4. Le pont En H :

Le pont en H est un circuit électronique permettant de contrôler la polarité aux bornes d'un dipôle. Il est composé de quatre interrupteurs ou transistors, disposés en forme de H. Cette structure est couramment utilisée en électronique de puissance, notamment pour le contrôle des moteurs.

En fonction de l'état des interrupteurs, le pont en H permet d'inverser le sens du courant traversant la charge. Lorsque deux interrupteurs opposés sont fermés, le courant circule dans un sens. En inversant ces interrupteurs, le courant s'inverse, changeant ainsi le sens de rotation du moteur. Le pont en H[11].

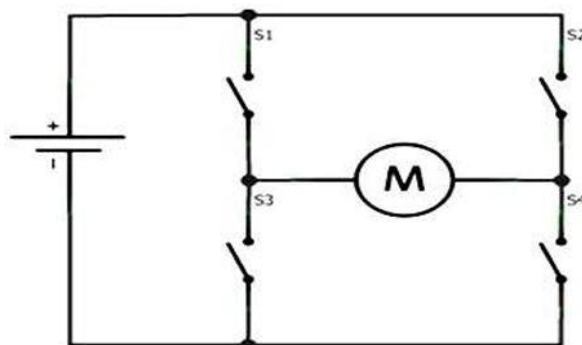


Figure III.11: Pont en « H »

Ce système est essentiel pour piloter des moteurs à courant continu, permettant leur marche avant et arrière.

III.4.2.1.4.1.Principe de fonctionnement :

Si on ferme les deux transistors S1 et S4 en laissant ouverts le S2 et S3. Le courant passe de la gauche vers la droite.

Si en revanche on fait le contraire (S2 et S3 fermés et S1 et S4 ouverts), le courant ira dans l'autre sens.

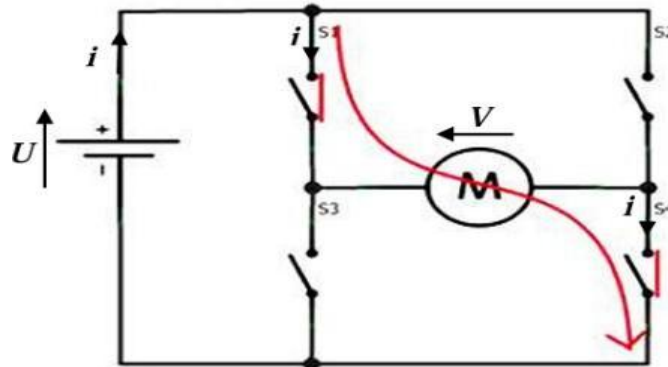


Figure III.12: Fonctionnement dans le sens horaire.

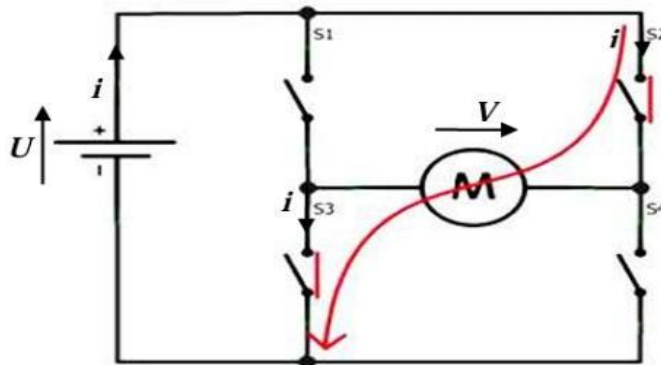


Figure III.13: Fonctionnement dans le sens antihoraire.

On remarque, que les transistors fonctionnent deux par deux. En effet, si on ferme juste un seul et laisse ouvert les trois autres le courant n'a nulle part où aller et rien ne se passe, le moteur est en roue libre. Si on ferme S1 et S2 en laissant S3 et S4 ouverts, cette action va créer ce que l'on appelle un frein magnétique. Le moteur se retrouve alors court-circuité. En tournant à cause de son inertie, le courant généré va revenir dans le moteur et va le freiner[17].

III.4.2.1.4.2. Le L293D :

Le composant L293D est un pont de puissance composé de plusieurs transistors et relais qui permet d'activer la rotation d'un moteur. Ce composant ne coûte pas très cher. Le L293D est un double pont-H, ce qui signifie qu'il est possible de l'utiliser pour commander deux moteurs

distincts, dans les deux sens et indépendamment l'un de l'autre.

III.4.2.1.4.3. Brochages :

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

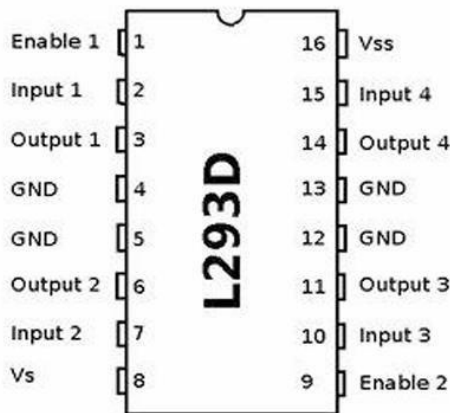


Figure III.14: Configuration des broches du L293D.

ENABLE 1 : permet d'envoyer (ou pas) la tension sur les sorties du moteur via OUTPUT 1 et OUTPUT 2 et commande l'activation/désactivation du premier Pont-H.

- Si ENABLE1 = GND, Le pont-H est déconnecté et le moteur ne fonctionne pas.
- Si ENABLE1 = VSS, Le pont-H est connecté aux sorties et le moteur fonctionne dans un sens ou l'autre ou pas en fonction des tensions appliquée sur INPUT 1 & INPUT 2.

INPUT 1, INPUT 2 : sont les broches de commande du Pont-H OUTPUT 1/OUTPUT 2. Il

Sera directement brancher à votre Arduino pour commander le sens du courant entre OUTPUT 1 et OUTPUT 2.

OUTPUT 1, OUTPUT 2 : sera branché directement sur le moteur.

GND : qui doit être raccordé à la masse de la source d'alimentation de puissance VS et à la masse de la source d'alimentation de la logique "VSS" (donc GND Arduino).

VS : Alimentation de puissance des moteurs.

ENABLE 2 : commande l'activation du second pont-H constitué d'OUTPUT 3/OUTPUT 4

INPUT 3, INPUT 4 : sont les broches de commande du Pont-H Output 3/Output 4. Il sera directement brancher à votre Arduino pour commander le sens du courant entre OUTPUT 3 et OUTPUT 4.

OUTPUT 3, OUTPUT 4 : sera branché directement sur le moteur.

Chapitre III : Simulation et programmation.

VSS : Alimentation de la logique de commande (5V). A raccorder à la borne +5V de votre Arduino.

Pour faire la variation de vitesse des moteurs 1 / 2 il faut raccorder ENABLE 1 / 2 a des bornes PWM d'Arduino[11].

III.4.2.1.2. Programme de moteur :

L'élaboration des fonctions de base du sous-programme moteur, s'articule autour des étapes suivantes :

- ✓ A l'état initial le moteur est au repos.
- ✓ Attente jusqu'à ce qu'un utilisateur appui sur un bouton.
- ✓ Si l'étage demandé par l'utilisateur est :
 - Supérieur à celui de la position actuelle de l'ascenseur, alors un ordre est envoyé sur la sortie commandant la montée de l'ascenseur.
 - Inférieur, alors un ordre est envoyé sur la sortie commandant la descente de l'ascenseur.

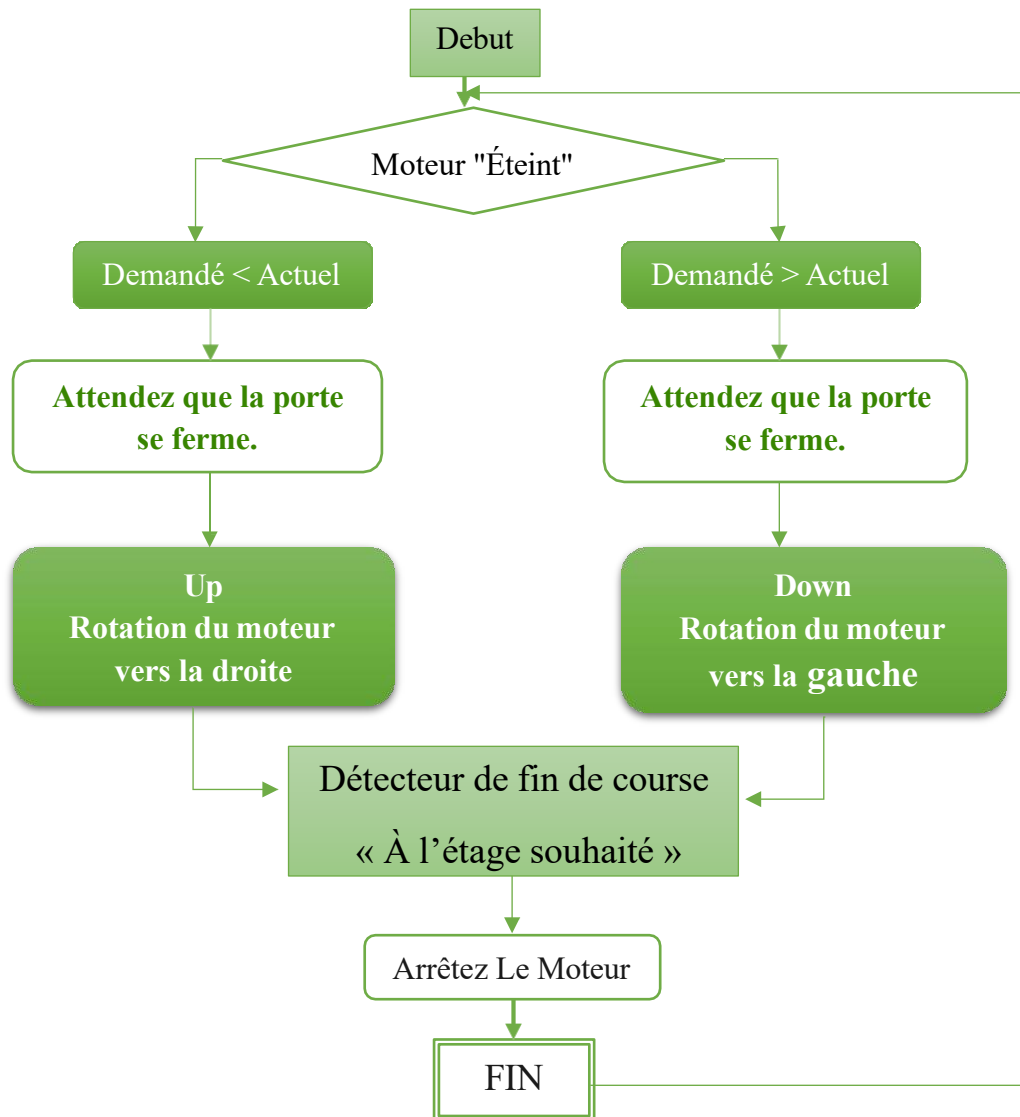


Figure III.15: Organigramme du moteur.

III.4.2.2. Sous-programme mouvement de la porte :

- ✓ Lorsque l'ascenseur arrive à un étage demandé, les portes s'ouvrent et doivent être maintenue ouvertes durant un laps de temps 30 de secondes, garantissant ainsi la sécurité de l'utilisateur durant l'accès à la cabine, après elles doivent se refermer.
- ✓ Cette période de temps doit être comptée à partir du moment où l'ascenseur se trouve immobilisé dans l'étage d'appel et avec les portes d'accès complètement ouvertes les portes d'accès de la cabine ne pourront s'ouvrir qu'au moment où la cabine de l'ascenseur sera immobilisée.
- ✓ Suite a une demande, l'ouverture ou la fermeture des portes ne peut s'effectuer que 30s après la demande pour permettre l'évacuation aisée de la cabine. Cette demande est prise en considération :

- ✓ Soit sur l'appui des boutons d'ouverture et de fermeture manuelle des portes qui sont places a l'intérieur de la cabine.
- ✓ Soit par l'appui sur les boutons d'appel.
- ✓ Si une présence est détectée, les portes doivent être ouverte complètement ou main tenues ouvertes, jusqu'à l'arrêt de la détection.

Terminée la détection, le système doit attendre 5secondes supplément aire savant de fermer les portes et de reprendre son fonctionnement normal.

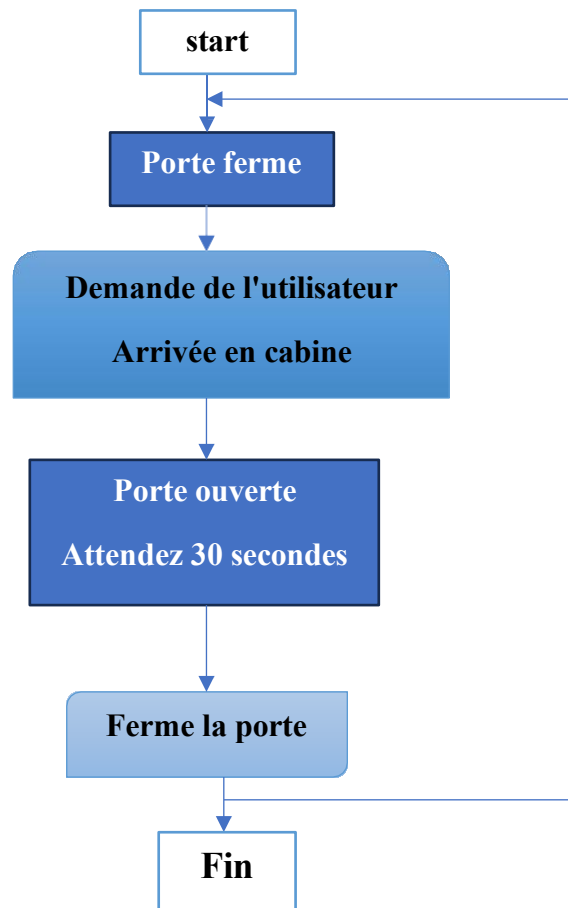


Figure III.16: Programme mouvement de la porte.

III.4.2.3. Écran LCD pour système d'ascenseur :

III.4.2.3.1. Description Générale :

L'afficheur LCD constitue une interface visuelle cruciale entre le système d'ascenseur et l'utilisateur. Son rôle fondamental est de transmettre de façon claire et ordonnée les informations essentielles du système, permettant ainsi à l'utilisateur d'interagir efficacement avec l'ascenseur et de comprendre son état de fonctionnement.

III.4.2.3.2. Fonctionnement :

Un afficheur LCD est capable d'afficher tous les caractères alphanumériques usuels et quelques symboles supplémentaires. Pour certains afficheurs, il est même possible de créer ses propres caractères. Chaque caractère est identifié par son code ASCII qu'il faut envoyer sur les lignes D0 à D7 broches 7 à 14. Ces lignes sont aussi utilisées pour la gestion de l'affichage avec l'envoi d'instructions telles que l'effacement de l'écran, l'écriture en ligne 1 ou en ligne 2 et le sens de défilement du curseur.

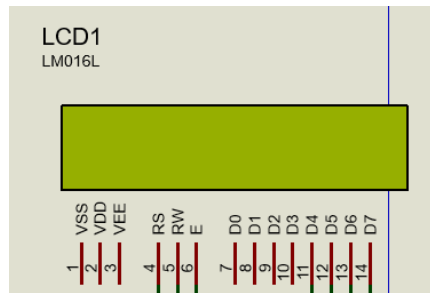


Figure III.17: Afficheur LCD16*2.

III.4.2.3.3. Brochages d'un Afficheur LCD :

Tableau III.2: Tableau de brochages d'un Afficheur LCD.

1	VSS	Alimentation positive +5V de l'afficheur « Power supply »
2	VDD	
3	VEE	Réglage du contraste de l'afficheur entre 0 et +5V
4	RS	Commutation de registre entre les instructions « 0 » et les données « 1 »
5	RW	Commutation entre lecture « 1 » (Read) et écriture « 0 » (Write)
6	E	Entrée de validation (Enable) active sur front descendant
7	D0	Bus de données bidirectionnel 3 états (haute impédance lorsque E=0) “NON UTILISE”
8	D1	
9	D2	
10	D3	
11	D4	Bus de données bidirectionnel 3 états (haute impédance lorsque E=0) “UTILISE”
12	D5	
13	D6	
14	D7	

L'afficheur LCD utilise 6 à 10 broches de données ((D0 à D7) ou (D4 à D7) + RS + E + deux d'alimentations (+5V et masse)), pour ajouter 8 entres /sorties à notre microcontrôleur, Nous avons utilisé la liaison I2C, Cette liaison est utilisable avec seulement 2 PIN (une broche de donnée et une broche d'horloge) et nécessite l'utilisation de deux broches analogiques de l'Arduino (PIN 4 et 5).

Chapitre III : Simulation et programmation.

La liaison I2C (Inter Integrated Circuit), est un protocole de communication qui permet de faire communiquer entre eux des composants électroniques variés grâce à seulement deux câbles :

- Un pour signal de donnée SDA (Serial Data).
- Un pour signal d'horloge SCL (Serial Clock).

- **Circuit PCF8574:**

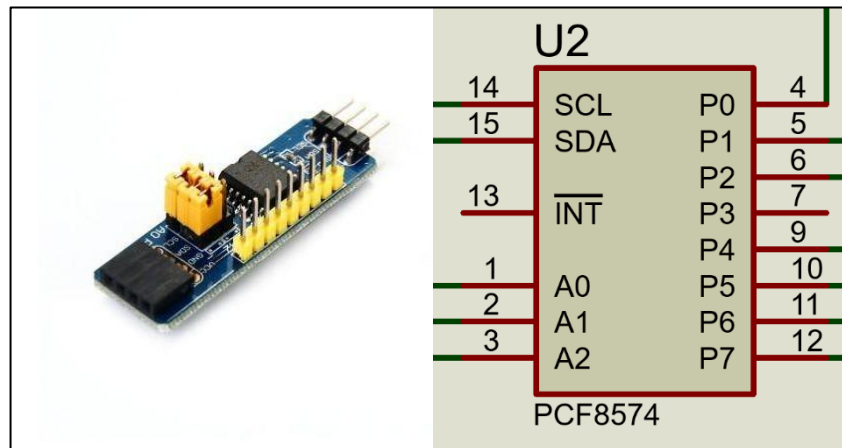


Figure III.18: Le circuit PCF8574

Le tableau ci-dessous représente les pins de circuit PCF8574 :

Tableau III.3 : Les pins de circuit PCF8574.

PIN	Symbole	Description
1	A0	adresse d'entrée 0
2	A1	adresse d'entrée 1
3	A2	adresse d'entrée 2
4	P0	E/S quasi-bidirectionnelles 0
5	P1	E/S quasi-bidirectionnelles 1
6	P2	E/S quasi-bidirectionnelles 2
7	P3	E/S quasi-bidirectionnelles 3
9	P4	E/S quasi-bidirectionnelles 4
10	P5	E/S quasi-bidirectionnelles 5
11	P6	E/S quasi-bidirectionnelles 6
12	P7	E/S quasi-bidirectionnelles 7
13	INT	sortie d'interruption (active LOW)
14	SCL	ligne d'horloge série
15	SDL	ligne de données série
16	VDD	tension d'alimentation

III.4.2.3.4. Fonctionnement de l'algorithme LCD :

- **Initialisation :**

- Au démarrage, l'écran LCD est initialisé, effacé, puis affiche un message d'accueil ("Elevator Ready") pendant quelques secondes.

Chapitre III : Simulation et programmation.

- Cela permet à l'utilisateur de savoir que le système est opérationnel.

➤ **Affichage de l'état courant :**

- Lorsque l'ascenseur est à l'arrêt, le LCD affiche le numéro de l'étage actuel, par exemple "Floor: 1".
- Pendant le déplacement, le LCD affiche un message indiquant la destination ("Moving to floor X").
- Lorsqu'il arrive, il affiche "Arrived at X" pour informer l'utilisateur.

➤ **Gestion des actions de la porte :**

- Pendant l'ouverture ou la fermeture de la porte, l'écran affiche respectivement "Opening door..." ou "Closing door...".
- Ces messages sont temporaires et servent à indiquer l'état de la porte.

➤ **Saisie du mot de passe :**

- Si le contrôle (par exemple pour accéder à l'étage 4) nécessite une authentification, l'écran affiche "Enter password!" et, au fur et à mesure de la saisie, montre des caractères masqués (étoiles) à la place des chiffres.

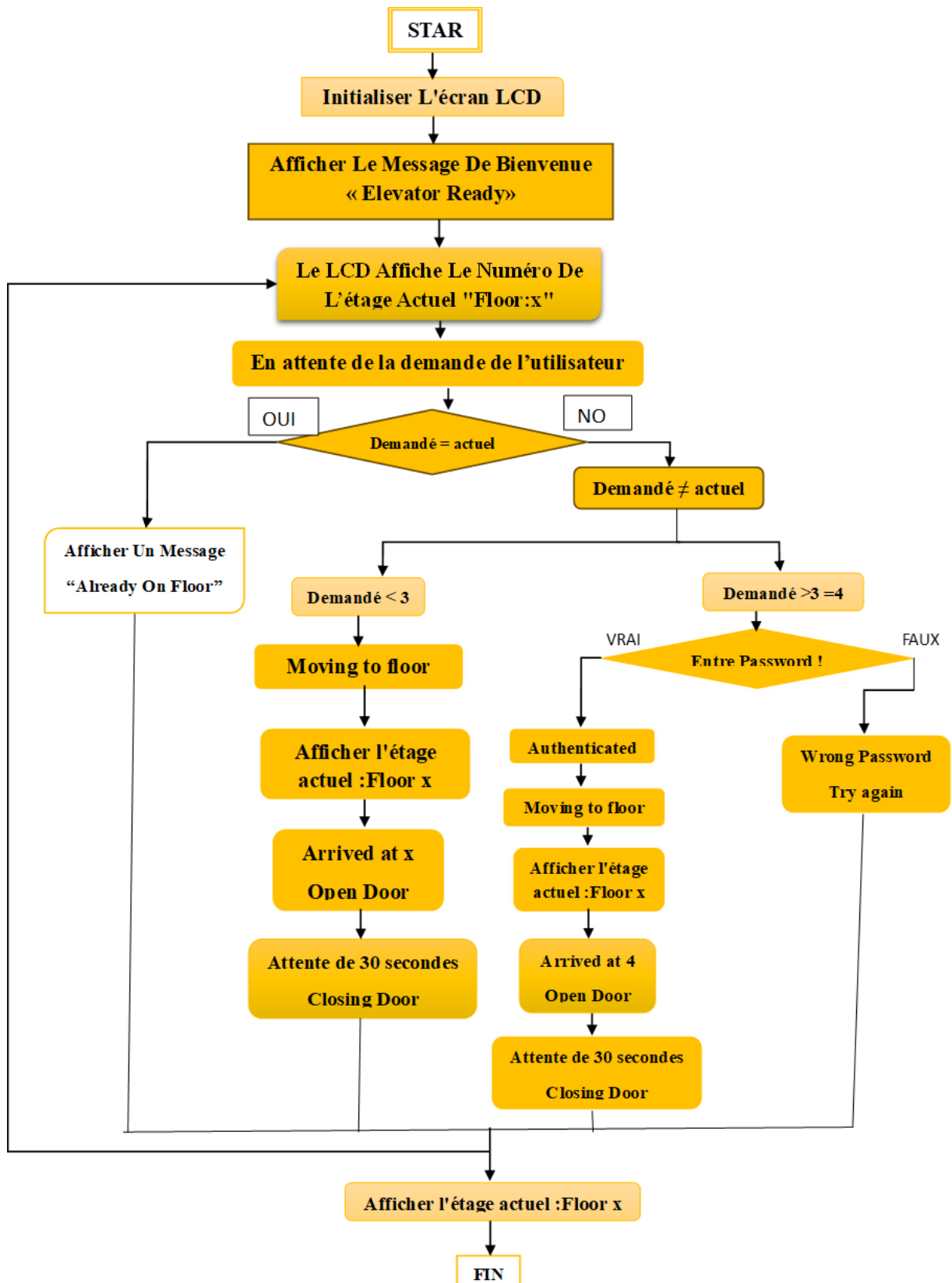


Figure III.19: Programme De Afficher LCD

III.4.2.4. Capteurs de position :

Dans les ascenseurs les capteurs de position détecte la position de la cabine. L'information est transmise à l'unité de traitement afin d'en tenir compte dans la gestion du mouvement de l'ascenseur, dans ce projet on utilise des capteurs de présence type infrarouge, nous avons monté un capteur par étage[2].



Figure III.20 : A- capteurs fin de course ; B- capteur infrarouge ; C- capteurs magnétiques.

III.4.2.4.1. Sous-programme : détection de la position :

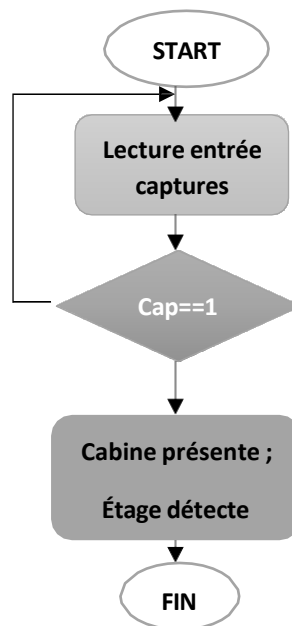


Figure III.21 : Sous-programme : détection de la position

III.4.2.5. Capteur Infrarouge :

Le capteur Infrarouge est un capteur qui se présente comme un équipement électronique

capable de réagir au rayonnement infrarouge renvoyé par les objets qui lui font face. Ces rayons se situent en dehors du spectre de la lumière visible. Le capteur infrarouge ou le capteur IR a deux parties principales. Émetteur IR et récepteur IR. La responsabilité de l'émetteur IR ou de l'émetteur infrarouge est la transmission des ondes infrarouges tandis que le travail du récepteur IR est de

recevoir ces ondes infrarouges. Le récepteur IR envoie constamment des données numériques sous la forme de 0 (LOW) ou 1 (HIGH) à la broche V out du capteur[2].



Figure III.22 : Capteur de présence.

III.4.2.5.1. Principe de fonctionnement :

L'émetteur infrarouge envoie un signal infrarouge qui, en cas de surface réfléchissante (par exemple de couleur blanche), rebondit dans certaines directions, y compris celle du récepteur infrarouge qui capte le signal détectant l'objet. Lorsque la surface est absorbante (par exemple de couleur noire), le signal IR n'est pas réfléchi et l'objet ne peut pas être détecté par le capteur. Ce résultat se produirait même si l'objet est absent[2].

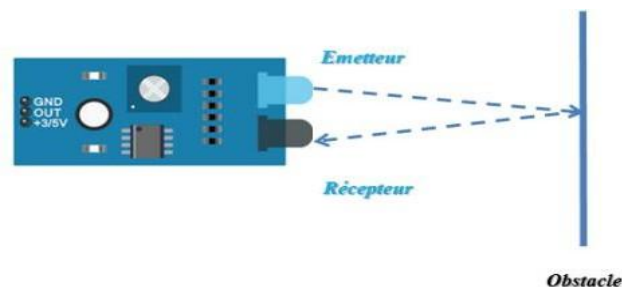


Figure III.23 : Onde IR (Infrarouge).

III.4.2.5.2. Caractéristiques Techniques :

- Voltage de fonctionnement : DC : 3,3 V-5 V
- Intensité du courant $\geq 20\text{mA}$

- Température de fonctionnement : $-10\text{ }^{\circ}\text{C}$ --> $+50\text{ }^{\circ}\text{C}$
- Plage de distance : 2-40 cm
- Interface IO : interface 4 fils (GND / Vcc / OUT / EN)
- Signal de sortie : tension TTL
- Mode d'hébergement : régulation de résistance multi-cercle
- Angle effectif : 35° .
- Taille : $41,7 * 16,7\text{ mm}$
- Poids : 5g

III.4.2.6. Bouton poussoir :

- Définition :

Le bouton poussoir ou interrupteur poussoir, est la base de l'interactivité, entre l'homme et la machine. Il permet de commander un objet, une machine ou simplement une lumière[2].



Figure III.24 : Bouton-poussoir.

III.5. Simulation du système de contrôle d'ascenseur par Arduino Uno :

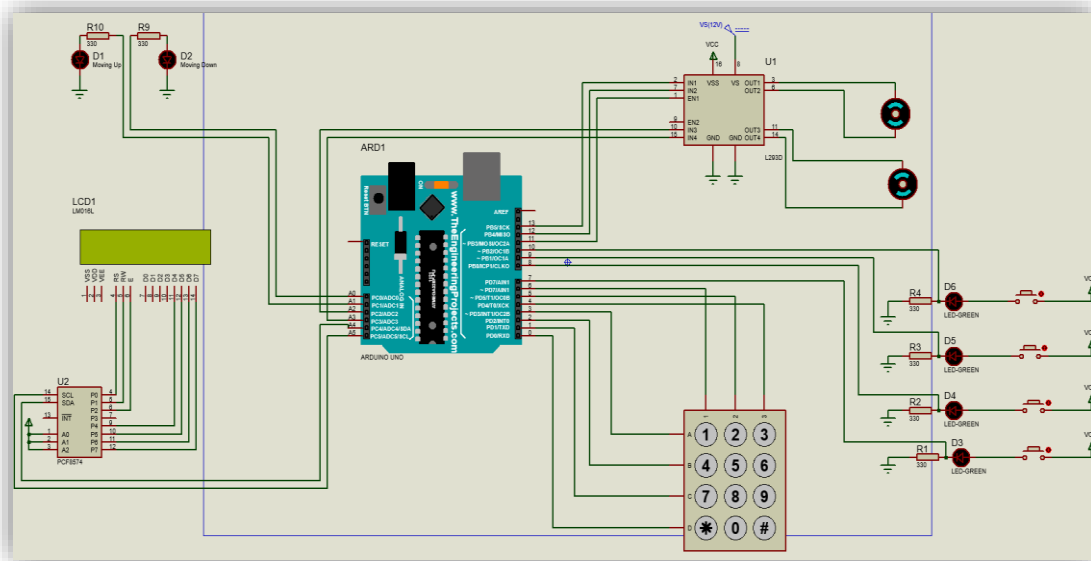


Figure III.25: Simulation de système de contrôle d'ascenseur par Arduino Uno.

III.5.1. Gestion et Fonctionnement de l'Ascenseur :

1. Initialisation :

- Au démarrage, la cabine est considérée au 1^{er} étage, les portes fermées, tous les moteurs à l'arrêt et l'écran affiche « Elevator Ready ».

2. Attente de commande :

- Le système affiche « Floor: X » (X=étage actuel).
- Il surveille le clavier matriciel : dès qu'un chiffre 1–4 est saisi, c'est le nouvel étage demandé.
- Si l'étage 4 est demandé, un code à 4 chiffres doit d'abord être validé.

3. Mouvement :

- Si l'étage demandé diffère de l'actuel, on commande le moteur de translation :
 - Vers le haut (LED “Moving Up” allumée) si cible > actuel.

- Vers le bas (LED “Moving Down” allumée) si cible < actuel.
- La cabine avance d’un étage par seconde, l’affichage LCD se met à jour à chaque palier.

4. Arrêt et portes :

- À l’arrivée (étage actuel = demandé), on coupe le moteur, puis on ouvre la porte (brève impulsion moteur porte), on reste ouvert quelques instants, puis on referme la porte.
- Une fois la porte refermée, on retourne en attente de nouvelle commande.

Ces points exposent brièvement le principe de fonctionnement de l’ascenseur ; ils seront détaillés au Chapitre IV avec la présentation des résultats de simulation.

III.6. Conclusion :

Dans notre projet intitulé « Conception Et Simulation D'un Système De Commande Électrique Pour Un Ascenseur Électromécanique ». Nous avons également présenté le logiciel de simulation « Proteus » ainsi que l'environnement de programmation « Arduino IDE ». Comme nous avons également abordé les algorithmes propres à chaque composant “moteur, mouvement des portes, écran LCD, etc.”, ainsi que le principe de fonctionnement de chacun d’entre eux.

Dans le chapitre suivant, nous allons exploiter ces connaissances pour présenter les résultats de la simulation et expliquer comment contrôler l'ascenseur.

Chapter IV:

Résultats De Simulation.

IV.1. Introduction :

Dans ce quatrième chapitre, nous présenterons les résultats détaillés de la simulation de notre système d'ascenseur électromécanique, ainsi qu'une analyse complète de chaque étape de la commande. Nous verrons d'abord comment les moteurs sont pilotés pour assurer un déplacement précis de la cabine, puis comment est gérée la séquence d'ouverture et de fermeture des portes pour garantir la sécurité des utilisateurs. Nous décrirons également l'interaction avec l'écran LCD et le clavier matriciel, et la manière dont les divers capteurs de fin de course et indicateurs LED contribuent à la fiabilité du système. Enfin, nous tirerons des conclusions sur les performances obtenues et proposerons des pistes d'optimisation pour améliorer encore la rapidité, la sûreté et le confort d'utilisation.

IV.2. Fonctionnement du système :

IV.2.1. Appel De la cabine d'ascenseur :

1. Appui sur le bouton d'appel : (Figure IV.1-a) ;

- L'utilisateur presse le poussoir situé sur la porte du palier de l'étage souhaité « ex : le 3^e ». (Figure IV.1-a)

2. Enregistrement de la demande: (Figure IV.1-b) ;

- Le contrôleur lit l'entrée numérique du poussoir et stocke « requestedFloor = 3 ».
- Si la cabine est au rez-de-chaussée “currentFloor = 1”, la demande est validée.

3. Activation du moteur de traction: (Figure IV.1-c) ;

- Le système commande le L293D pour alimenter le moteur de translation dans le sens « monter ».
- La LED « Moving Up » s'allume pour indiquer visuellement la montée.

4. Affichage du niveau de la cabine: (Figure IV.1-d) ;

- À chaque palier franchi, la variable currentFloor s'incrémente et l'écran LCD se met à jour.

5. Arrêt à destination : (Figure IV.1-e) ;

- Dès que `currentFloor == requestedFloor (4)`, le contrôleur coupe l'alimentation du moteur.

6. Ouverture automatique des portes : (Figure IV.1-f) ;

- Le moteur de porte s'active dans le sens « ouverture » pendant une durée fixe (5 s).
- L'écran LCD affiche simultanément « Opening door... ».

7. Maintien ouvert et fermeture : (Figure IV.1-g)

- La porte reste ouverte 5 secondes.
- Passé ce délai, le contrôleur inverse la polarité du moteur de porte pour la fermeture (`closeDoor()`), puis met les I/O moteur à 0.

8. Retour en veille :

- Une fois la porte refermée, le système remet `requestedFloor = currentFloor` et attend la prochaine sollicitation.

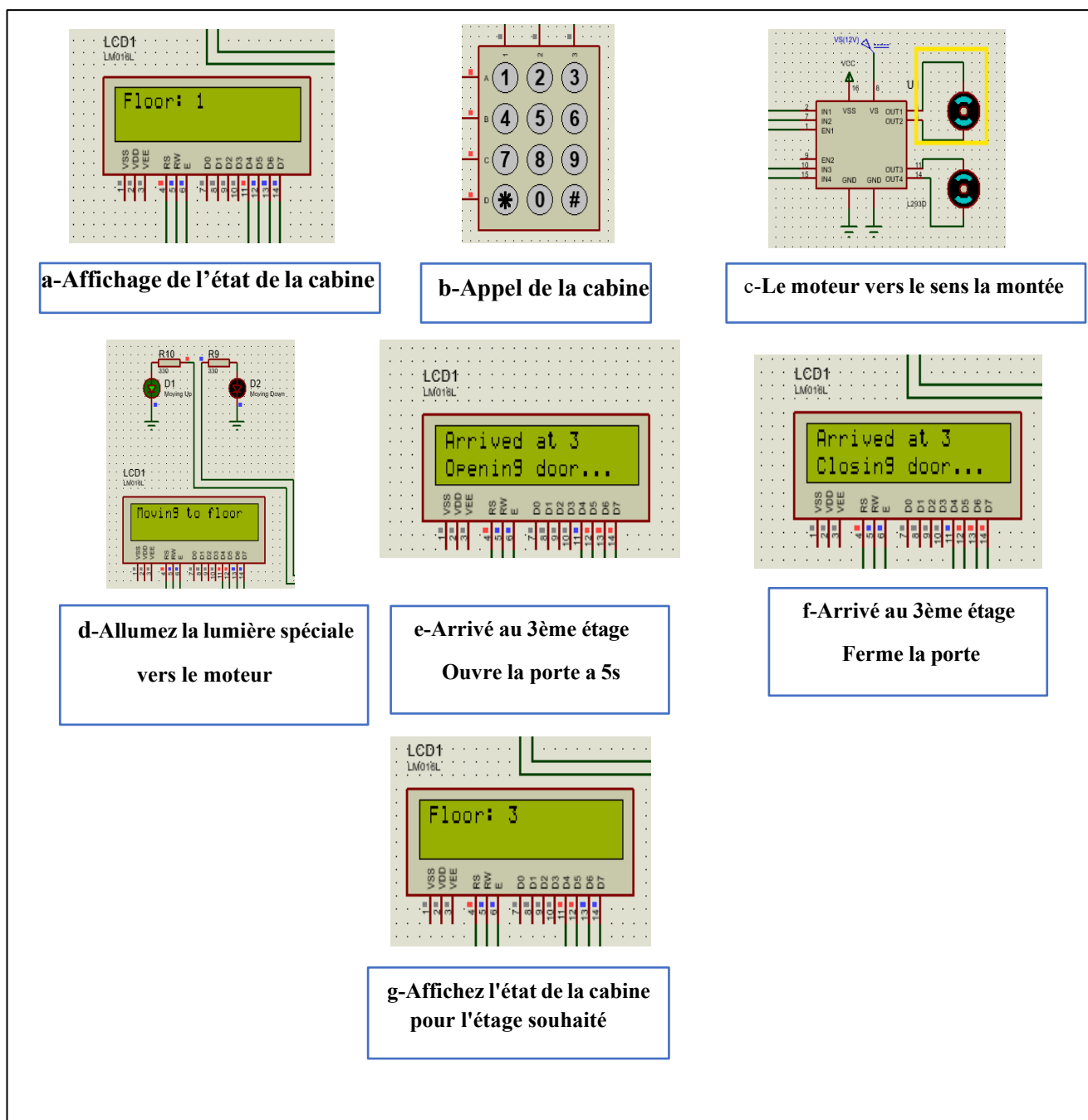


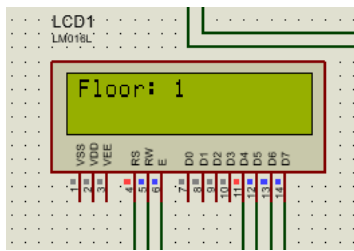
Figure IV.1: choix d'étage.

IV3. Appeler le 4^{ème}. étage et demander le mot de passe:

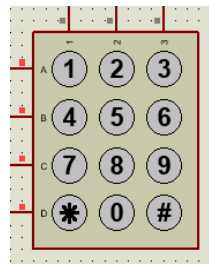
Appel du 4^e étage avec demande de mot de passe :

1. Affichage de l'état de la cabine. (Figure IV.2- a).
2. L'utilisateur sur le palier appuie sur la touche « 4 » de la matrice (La fonction keypad.getKey() détecte l'appui et retourne le caractère '4',(Dans readKeypad(), on convertit '4' en entier reqFloor = 4). (Figure IV.2- b).
3. Invitation à saisir le mot de passe “Enter password!”,(Le curseur se place en début de seconde ligne pour la saisie masquée, On affiche un '*' sur le LCD pour masquer la saisie.

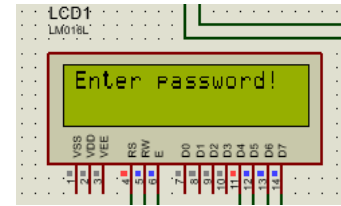
Une fois les 4 chiffres saisis, on compare le tableau pass[] au mot de passe de référence ("1234").). (Figure IV.2- c).
4. Si les quatre chiffres correspondent (Figure IV.2- d):
 - L'écran affiche brièvement « Authenticated »
 - La fonction renvoie true et la demande d'étage est confirmée.
5. Le moteur vers le sens la montée. (Figure IV.2- e).
6. Allumez la lumière spéciale vers le moteur. (Figure IV.2- f).
7. Arrivé au 4^{ème} étage 'Ouvre la porte a 5s' . (Figure IV.2- g).
8. Arrivé au 3^{ème} étage 'Ferme la porte', (Figure IV.2- h).
9. Affichez l'état de la cabine pour l'étage souhaité, (Figure IV.2-k).
10. Si non le mot de passe est incorrect. ; (Figure IV.2 -L)
 - L'écran affiche « Wrong Password – Try again »
 - La fonction renvoie false, requestedFloor est remis à la valeur du currentFloor et aucune montée n'est lancée.



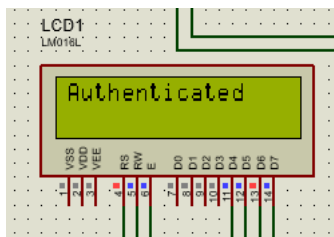
a-Affichage de l'état de la cabine



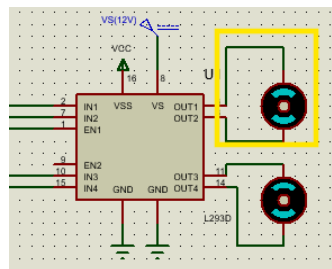
b-Appel de la cabine 4



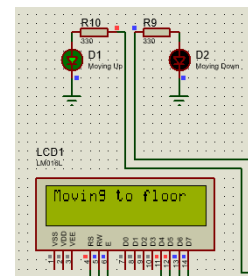
c-Demande de mot de passe



d-Validation du mot de passe



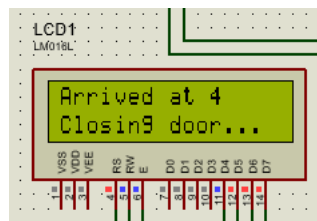
e-Le moteur vers le sens la montée



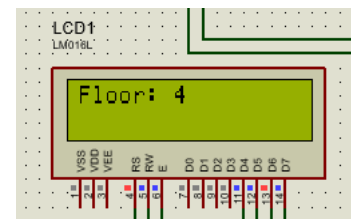
f-Allumez la lumière spéciale vers le moteur



g-Arrivé au 4ème étage
Ouvre la porte a 5s

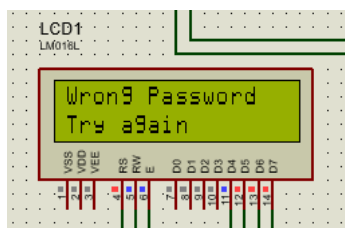


h-Arrivé au 3ème étage
Ferme la porte



k-Affichez l'état de la cabine pour l'étage souhaité

- Si le mot de passe est incorrect:



L- Afficher le message d'erreur
« Wrong Password – Try again »

Figure IV.2: Choix d'étage 4ème.

IV4 Programme Arduino Code :

```
1. #include <Keypad.h>
#include <Wire.h>

#include <LiquidCrystal_I2C.h>

const byte ROWS = 4;
const byte COLS = 3;
char keys[ROWS][COLS] = {
    {'1','2','3'},
    {'4','5','6'},
    {'7','8','9'},
    {'*','0','#'}};
};
byte rowPins[ROWS] = {3, 2, 1, 0};
byte colPins[COLS] = {6, 5, 4};
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

LiquidCrystal_I2C lcd(0x27, 16, 2);

int currentFloor = 1;
int targetFloor = 1;
char password[] = {'1', '2', '3', '4'};

#define MOTOR_UP 12
#define MOTOR_DOWN 13
#define MOTOR_SPEED 11

#define DOOR_MOTOR_IN1 A2
#define DOOR_MOTOR_IN2 A3

#define MOVING_UP_LED A1
#define MOVING_DOWN_LED A0

void setup() {
```

```
lcd.init();
lcd.backlight();

pinMode(MOTOR_UP, OUTPUT);
pinMode(MOTOR_DOWN, OUTPUT);
pinMode(MOTOR_SPEED, OUTPUT);

pinMode(DOOR_MOTOR_IN1, OUTPUT);
pinMode(DOOR_MOTOR_IN2, OUTPUT);

pinMode(MOVING_UP_LED, OUTPUT);
pinMode(MOVING_DOWN_LED, OUTPUT);

digitalWrite(MOTOR_UP, LOW);
digitalWrite(MOTOR_DOWN, LOW);
analogWrite(MOTOR_SPEED, 0);

digitalWrite(DOOR_MOTOR_IN1, LOW);
digitalWrite(DOOR_MOTOR_IN2, LOW);

currentFloor = 1;
targetFloor = 1;

lcd.clear();
lcd.print("Elevator Ready");
delay(2000);
lcd.clear();
}

void loop() {
    printCurrentFloor();
    char key = keypad.getKey();
    if (key) {
        int reqFloor = key - '0';
```

```
    if (reqFloor >= 1 && reqFloor <= 4) {  
        targetFloor = reqFloor;  
        if (targetFloor == 4) {  
            if (!checkPassword()) {  
                targetFloor = currentFloor;  
                return;  
            }  
        }  
        moveElevator(targetFloor);  
    }  
}  
  
void moveElevator(int target) {  
    if (target == currentFloor) {  
        lcd.clear();  
        lcd.print("Already on floor ");  
        lcd.print(target);  
        delay(1000);  
        return;  
    }  
    lcd.clear();  
    lcd.print("Moving to floor ");  
    lcd.print(target);  
  
    if (target > currentFloor) {  
        up();  
        digitalWrite(MOVING_UP_LED, HIGH);  
        while (currentFloor < target) {  
            delay(1000);  
            currentFloor++;  
            printCurrentFloor();  
        }  
        digitalWrite(MOVING_UP_LED, LOW);  
    }
```

```
} else {  
    down();  
    digitalWrite(MOVING_DOWN_LED, HIGH);  
    while (currentFloor > target) {  
        delay(1000);  
        currentFloor--;  
        printCurrentFloor();  
    }  
    digitalWrite(MOVING_DOWN_LED, LOW);  
}  
stopElevator();  
lcd.clear();  
lcd.print("Arrived at ");  
lcd.print(currentFloor);  
delay(500);  
openDoor();  
delay(100);  
closeDoor();  
delay(100);  
}  
  
void up() {  
    digitalWrite(MOTOR_UP, HIGH);  
    digitalWrite(MOTOR_DOWN, LOW);  
    analogWrite(MOTOR_SPEED, 125);  
}  
  
void down() {  
    digitalWrite(MOTOR_UP, LOW);  
    digitalWrite(MOTOR_DOWN, HIGH);  
    analogWrite(MOTOR_SPEED, 125);  
}  
  
void stopElevator() {
```

```
digitalWrite(MOTOR_UP, LOW);
digitalWrite(MOTOR_DOWN, LOW);
analogWrite(MOTOR_SPEED, 0);
}

bool checkPassword() {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Enter password!");
    byte counter = 0;
    char pass[4];
    while (counter < 4) {
        char key = keypad.getKey();
        if (key) {
            pass[counter] = key;
            lcd.setCursor(counter, 1);
            lcd.print(key);
            delay(300);
            lcd.setCursor(counter, 1);
            lcd.print('*');
            counter++;
        }
    }
    for (int i = 0; i < 4; i++) {
        if (pass[i] != password[i]) {
            lcd.clear();
            lcd.setCursor(0, 0);
            lcd.print("Wrong Password");
            lcd.setCursor(0, 1);
            lcd.print("Try again");
            delay(1000);
            lcd.clear();
            return false;
        }
    }
}
```

```
}

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Authenticated");
delay(1000);
lcd.clear();
return true;
}

void openDoor() {
    lcd.setCursor(0, 1);
    lcd.print("Opening door...");
    digitalWrite(DOOR_MOTOR_IN1, HIGH);
    digitalWrite(DOOR_MOTOR_IN2, LOW);
    delay(110);
    digitalWrite(DOOR_MOTOR_IN1, LOW);
    delay(110);
}

void closeDoor() {
    lcd.setCursor(0, 1);
    lcd.print("Closing door...");
    digitalWrite(DOOR_MOTOR_IN1, LOW);
    digitalWrite(DOOR_MOTOR_IN2, HIGH);
    delay(110);
    digitalWrite(DOOR_MOTOR_IN2, LOW);
    delay(110);
}

void printCurrentFloor() {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Floor: ");
    lcd.print(currentFloor);
}
```

```
delay(500);  
}
```

IV5. Conclusion :

Ce chapitre consiste à expérimenter la simulation d'ascenseur en utilisant le logiciel professionnel ISIS Proteus 8, et à programmer un microcontrôleur Arduino Uno en utilisant l'environnement de développement Arduino IDE. C'est ainsi que nous avons obtenu les résultats présentés

Conclusion Générale

L'objectif de ce travail était de concevoir et de simuler un ascenseur électromécanique piloté par un microcontrôleur Arduino Uno, tout en élaborant une méthodologie rigoureuse pour étudier chacune de ses phases de fonctionnement. Ce projet nous a permis d'approfondir nos connaissances en automatisme industriel et de mettre en pratique notre expertise technique acquise durant nos études.

La revue bibliographique associée a enrichi notre compréhension des systèmes de traction, de sécurité et de commande des ascenseurs, ouvrant la voie à une implémentation à la fois pédagogique et fonctionnelle. Cependant, la complexité croissante du programme—en termes de nombre de fonctions et de structure logicielle—ainsi que le nombre limité de broches de l'Arduino Uno ont révélé leurs limites lors de l'ajout de fonctionnalités avancées (protection anti-surcharge, gestion prioritaire des appels, interverrouillage des portes, etc.).

Afin d'enrichir et de pérenniser ce prototype, nous proposons les améliorations suivantes :

1. Adoption de l'Arduino Mega :

Pour bénéficier d'un plus grand nombre d'entrées/sorties et faciliter l'intégration de nouveaux capteurs et actionneurs.

2. Ajout d'un capteur de surcharge :

Pour garantir une limitation de charge automatique et renforcer la sécurité des usagers.

3. Intégration de capteurs de sécurité :

(détection d'obstacle dans l'arbre, arrêt d'urgence, interverrouillage des portes) afin de répondre aux normes en vigueur.

4. Verrouillage des déplacements :

Interdiction de tout mouvement de la cabine tant que les portes ne sont pas entièrement fermées, assurant une sécurité accrue.

Ces évolutions dessinent une feuille de route claire pour transformer notre démonstrateur en un système plus robuste, modulable et conforme aux exigences industrielles et de sécurité.

Références bibliographiques :

- [1]: Mémoire de Thème " Étude, conception et réalisation d'un prototype d'ascenseur " Ms. Eln. (EMBOUAZZA + MOSTEFAOUI Juillet 2017/2018).
- [2] : Mémoire de Thème "Command d'un ascenseur par PLC" (Mahdi LARIBI...4 Juillet 2017).
- [3] : Mémoire de Thème " étude de principe de fonctionnement d'un ascenseur". Réaliser par :
BOUFAFA MUSTAPHA KAMEL ; UNIVERSITE BADJI MOKHTAR ANNABA. Année : 2014/2015.
- [4] : Mémoire de Thème « Etude et Réalisation d'une Carte Arduino » Projet de fin d'études en vue de l'obtention d'un Master Mr.CHELOUCHE D, Mr.DJAFRI M. Année : 2020/2021.
- [5] : Mémoire de Thème " Ascenseurs à vis sans fin ". Réaliser par : MOURAD Ali. Institut des sciences Appliquées et économiques. Université de Libanaise ; Paris.2013.
- [6] Sharkawy, Abdel-Nasser, and Gamal T. Abdel-Jaber. "Design and Implementation of a Prototype of Elevator Control System: Experimental Work." SVU-International Journal of Engineering Sciences and Applications 3.2 (2022): 80-86.
- [7] Fakhruldeen, Hassan Falah, et al. "An Arduino-based voice-recognition elevator for special purposes." Indonesian Journal of Electrical Engineering and Computer Science 31.2 (2023): 828-834.
- [8] Bienfait, M. U. C. Y. O. "DESIGN AND IMPLEMENTATION OF INTELLIGENT ELEVATOR." PhD diss., ULK, 2024.
- [9] Patjoshi, Rajesh. "Design and implementation of embedded based elevator control system." PhD diss., 2010.
- [10] Ascenseur réglementation et programmation (Michel CHALAUX 2019)
- [11] Mémoire de Thème « Conception, et réalisation d'un ascenseur à base de la carte de développement ARDUINO UNO », Projet de fin d'études en électronique, Université Mouloud Mammeri de Tizi-Ouzou.2018.
- [12] : La chaîne information Livret des compétences essentielles de seconde ISI. Fiche N°1-1.
- [13] : FC-chaîne-energie_information (JEZEGOU 2017).
- [14] <https://www.fuuu.be/polytech/ELECH300/CHAP2-ELEC-H-300>.
- [15] : CT-2.2-La-chaîne-fonctionnelle-Élève (CHABRIER + NOURY 2019).
- [16] :<https://fr.scribd.com/document/843989299/Introduction-Generale-sur-les-Ascenseurs>.

- [17] : Souaidia Zohra «Etude et simulation d'un monte-charge par Arduino», Mémoire de Master, Université Badji Mokhtar Annaba.
- [18] : Djafri M, Chelouche D, «Etude et réalisation d'une carte Arduino », Projet de fin d'études en électronique, Université A.MIRA de Bejaia.
- [19] : CHERROUN Lakhmissi « Navigation Autonome d'un Robot Mobile par des Techniques NeuroFloues », Thèse de Doctorat en Automatique, Université Mohamed Khider, Biskra ,2014. [25] A poros de mchob by-Wiki. Servo-Moteur0URL : <http://dspace.univ-tlemcen.dz>.
- [20] Kafi Mohammed Mustafa, Houamed Abdelhak, Un radar de recule à Ultrasons, Mémoire professionnel. Université de Kasdi Merbah_Ouargla.2018. URL : <https://dspace.univ-ouargla.dz>.
- [21] guide d'utilisateur de logiciel Proteus, consulté en Mai 2016.
- [22] KRAMA Abdelbaset, GOUGUI Abdelmoumen, «Etude et réalisation d'une carte de contrôle par Arduino via le système Androïde », projet de fin d'étude Université KASDI Merbah, Ouargla, 2015.
- [23] Christian Tavenir, « Arduino, maîtrisez sa programmation et ces cartes d'interface (Shields) »,
- [24] <https://www.arduino.cc>. Consulté le 20/08/2021.
- [25] O. Lefebvre « Navigation Autonome sans Collision pour Robots Mobiles non holonomes », Thèse de Doctorat de l'Institut National Polytechnique de Toulouse, 2006.
- [26] FILLIAT David, « Robotique Mobile », Ecole Nationale supérieur de Techniques avancées, Paris Tech, 2013.
- [27] Mémoire de Thème :'' ETUDE D'UN ASCENSEUR INDUSTRIEL ET MISE EN ŒUVRE DE LA TRANSMISSION DES DONNEES EN SERIE'' Projet de fin d'études en vue de l'obtention d'un Master, Mr LEMAAREG T, Mr MOKRANI F (2006/2007)
- [28] Chaîne d'énergie et chaîne d'information : <https://technobriez.fr/wp-content/uploads/2019/11/CT-2.2-La-cha%C3%Aene-fonctionnelle-correction.pdf>

Annexes

1. Les datasheets des circuits utilisés :

1. Le circuit L293D :



L293D
L293DD

PUSH-PULL FOUR CHANNEL DRIVER WITH DIODES

- 600mA OUTPUT CURRENT CAPABILITY PER CHANNEL
- 1.2A PEAK OUTPUT CURRENT (non repetitive) PER CHANNEL
- ENABLE FACILITY
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)
- INTERNAL CLAMP DIODES

DESCRIPTION

The Device is a monolithic integrated high voltage, high current four channel driver designed to accept standard DTL or TTL logic levels and drive inductive loads (such as relays solenoids, DC and stepping motors) and switching power transistors.

To simplify use as two bridges each pair of channels is equipped with an enable input. A separate supply input is provided for the logic, allowing operation at a lower voltage and internal clamp diodes are included.

This device is suitable for use in switching applications at frequencies up to 5 kHz.



SO(12+4+4)



Powerdip (12+2+2)

ORDERING NUMBERS:

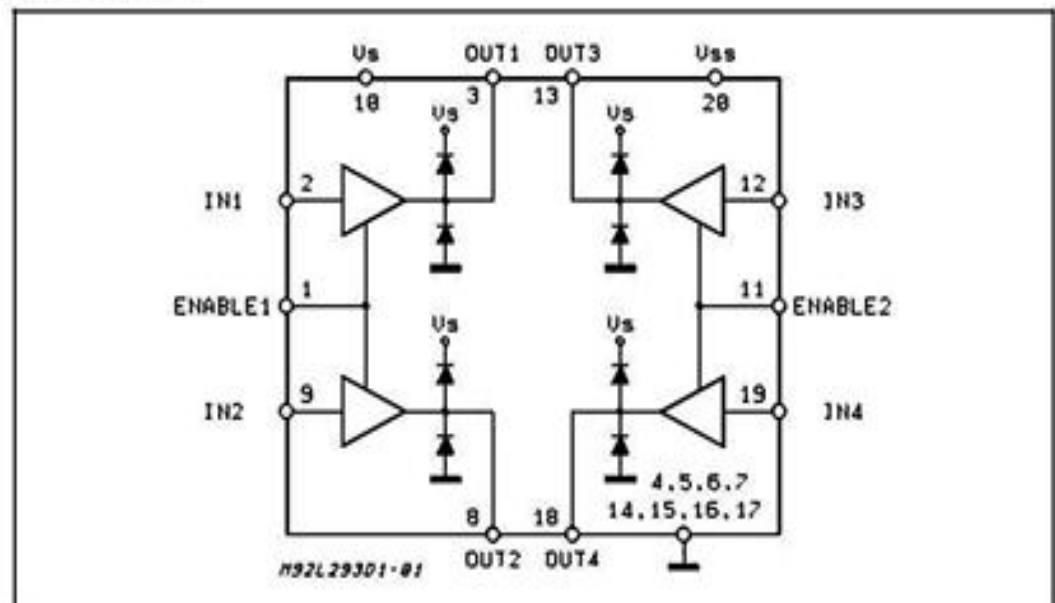
L293DD

L293D

The L293D is assembled in a 16 lead plastic package which has 4 center pins connected together and used for heatsinking

The L293DD is assembled in a 20 lead surface mount which has 8 center pins connected together and used for heatsinking.

BLOCK DIAGRAM

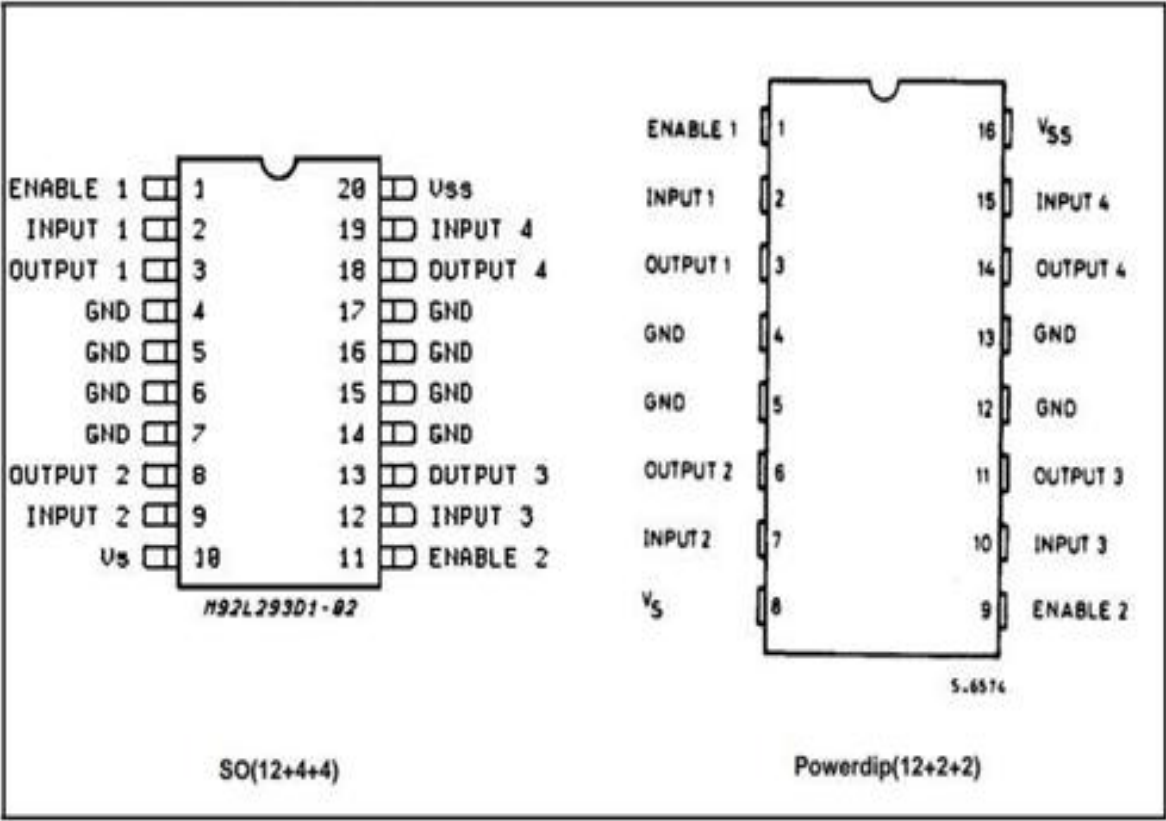


L293D - L293DD

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_S	Supply Voltage	36	V
V_{SS}	Logic Supply Voltage	36	V
V_i	Input Voltage	7	V
V_{en}	Enable Voltage	7	V
I_o	Peak Output Current (100 μ s non repetitive)	1.2	A
P_{tot}	Total Power Dissipation at $T_{pins} = 90\text{ }^{\circ}\text{C}$	4	W
T_{stg}, T_j	Storage and Junction Temperature	- 40 to 150	$^{\circ}\text{C}$

PIN CONNECTIONS (Top view)



THERMAL DATA

Symbol	Decription	DIP	SO	Unit
$R_{th\ j-pins}$	Thermal Resistance Junction-pins	max.	14	$^{\circ}\text{C/W}$
$R_{th\ j-amb}$	Thermal Resistance junction-ambient	max.	50 (*)	$^{\circ}\text{C/W}$
$R_{th\ j-case}$	Thermal Resistance Junction-case	max.	14	

(*) With 6sq. cm on board heatsink.

ELECTRICAL CHARACTERISTICS (for each channel, $V_S = 24\text{ V}$, $V_{SS} = 5\text{ V}$, $T_{\text{amb}} = 25\text{ }^{\circ}\text{C}$, unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V_S	Supply Voltage (pin 10)		V_{SS}		36	V
V_{SS}	Logic Supply Voltage (pin 20)		4.5		36	V
I_S	Total Quiescent Supply Current (pin 10)	$V_i = L; I_O = 0; V_{en} = H$		2	6	mA
		$V_i = H; I_O = 0; V_{en} = H$		16	24	mA
		$V_{en} = L$			4	mA
I_{SS}	Total Quiescent Logic Supply Current (pin 20)	$V_i = L; I_O = 0; V_{en} = H$		44	60	mA
		$V_i = H; I_O = 0; V_{en} = H$		16	22	mA
		$V_{en} = L$		16	24	mA
V_{iL}	Input Low Voltage (pin 2, 9, 12, 19)		-0.3		1.5	V
V_{iH}	Input High Voltage (pin 2, 9, 12, 19)	$V_{SS} \leq 7\text{ V}$	2.3		V_{SS}	V
		$V_{SS} > 7\text{ V}$	2.3		7	V
I_{iL}	Low Voltage Input Current (pin 2, 9, 12, 19)	$V_{iL} = 1.5\text{ V}$			-10	μA
I_{iH}	High Voltage Input Current (pin 2, 9, 12, 19)	$2.3\text{ V} \leq V_{iH} \leq V_{SS} - 0.6\text{ V}$		30	100	μA
V_{enL}	Enable Low Voltage (pin 1, 11)		-0.3		1.5	V
V_{enH}	Enable High Voltage (pin 1, 11)	$V_{SS} \leq 7\text{ V}$	2.3		V_{SS}	V
		$V_{SS} > 7\text{ V}$	2.3		7	V
I_{enL}	Low Voltage Enable Current (pin 1, 11)	$V_{enL} = 1.5\text{ V}$		-30	-100	μA
I_{enH}	High Voltage Enable Current (pin 1, 11)	$2.3\text{ V} \leq V_{enH} \leq V_{SS} - 0.6\text{ V}$			± 10	μA
$V_{CE(sat)H}$	Source Output Saturation Voltage (pins 3, 8, 13, 18)	$I_O = -0.6\text{ A}$		1.4	1.8	V
$V_{CE(sat)L}$	Sink Output Saturation Voltage (pins 3, 8, 13, 18)	$I_O = +0.6\text{ A}$		1.2	1.8	V
V_F	Clamp Diode Forward Voltage	$I_O = 600\text{ nA}$		1.3		V
t_r	Rise Time (*)	0.1 to 0.9 V_O		250		ns
t_f	Fall Time (*)	0.9 to 0.1 V_O		250		ns
t_{on}	Turn-on Delay (*)	0.5 V_i to 0.5 V_O		750		ns
t_{off}	Turn-off Delay (*)	0.5 V_i to 0.5 V_O		200		ns

(*) See fig. 1.

2. Le bus I2C :

Philips Semiconductors

Product specification

Remote 8-bit I/O expander for I2C-bus

PCF8574

1 FEATURES

- Operating supply voltage 2.5 to 6 V
- Low standby current consumption of 10 μA maximum
- I2C-bus to parallel port expander
- Open-drain interrupt output
- 8-bit remote I/O port for the I2C-bus
- Compatible with most microcontrollers
- Latched outputs with high current drive capability for directly driving LEDs
- Address by 3 hardware address pins for use of up to 8 devices (up to 16 with PCF8574A)
- DIP16, or space-saving SO16 or SSOP20 packages.

2 GENERAL DESCRIPTION

The PCF8574 is a silicon CMOS circuit. It provides general purpose remote I/O expansion for most microcontroller families via the two-line bidirectional bus (I2C-bus).

3 ORDERING INFORMATION

TYPE NUMBER	PACKAGE		
	NAME	DESCRIPTION	VERSION
PCF8574P; PCF8574AP	DIP16	plastic dual in-line package; 16 leads (300 mil)	SOT38-4
PCF8574T; PCF8574AT	SO16	plastic small outline package; 16 leads; body width 7.5 mm	SOT162-1
PCF8574TS; PCF8574ATS	SSOP20	plastic shrink small outline package; 20 leads; body width 4.4 mm	SOT266-1



The device consists of an 8-bit quasi-bidirectional port and an I2C-bus interface. The PCF8574 has a low current consumption and includes latched outputs with high current drive capability for directly driving LEDs. It also possesses an interrupt line (INT) which can be connected to the interrupt logic of the microcontroller. By sending an interrupt signal on this line, the remote I/O can inform the microcontroller if there is incoming data on its ports without having to communicate via the I2C-bus. This means that the PCF8574 can remain a simple slave device.

The PCF8574 and PCF8574A versions differ only in their slave address as shown in Fig.10.

4 BLOCK DIAGRAM

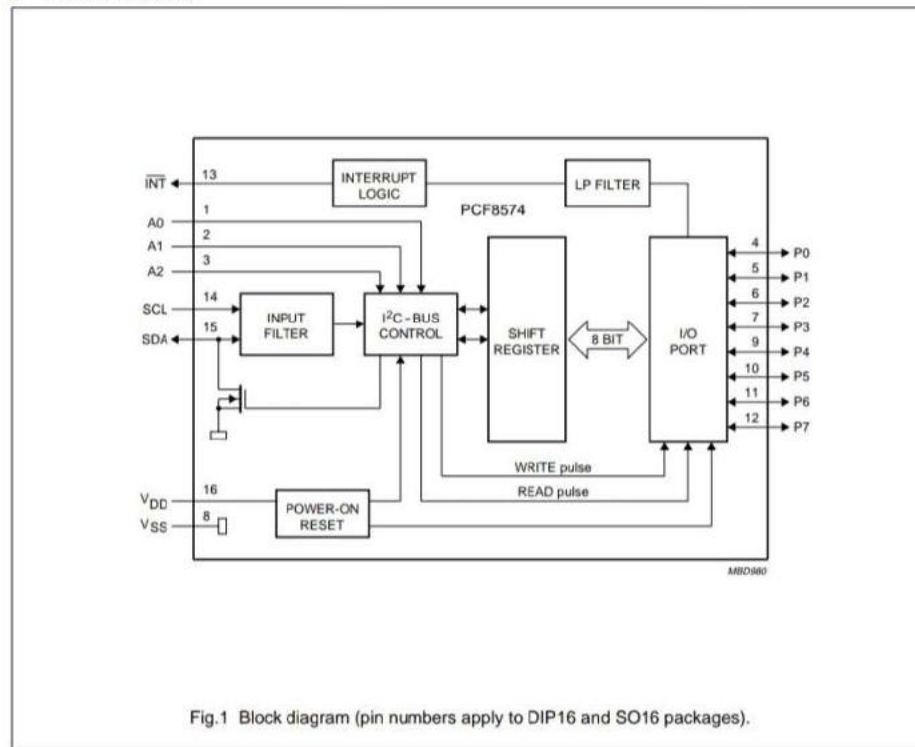


Fig.1 Block diagram (pin numbers apply to DIP16 and SO16 packages).

5 PINNING

5.1 DIP16 and SO16 packages

SYMBOL	PIN	DESCRIPTION
A0	1	address input 0
A1	2	address input 1
A2	3	address input 2
P0	4	quasi-bidirectional I/O 0
P1	5	quasi-bidirectional I/O 1
P2	6	quasi-bidirectional I/O 2
P3	7	quasi-bidirectional I/O 3
V _{SS}	8	supply ground
P4	9	quasi-bidirectional I/O 4
P5	10	quasi-bidirectional I/O 5
P6	11	quasi-bidirectional I/O 6
P7	12	quasi-bidirectional I/O 7
INT [‾]	13	interrupt output (active LOW)
SCL	14	serial clock line
SDA	15	serial data line
V _{DD}	16	supply voltage

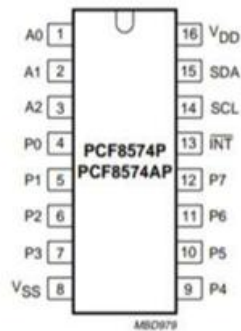


Fig.2 Pin configuration (DIP16).

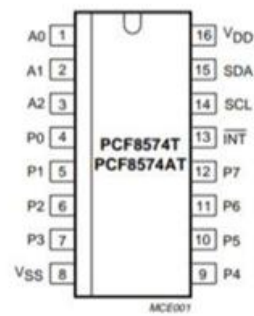
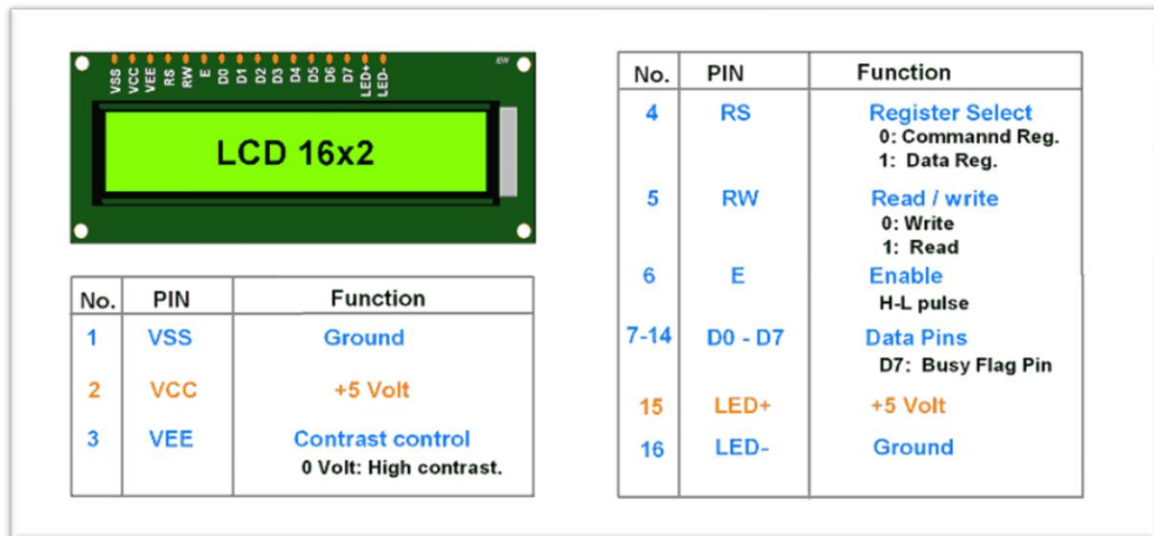


Fig.3 Pin configuration (SO16).

3. LCD 16x2 Pin Description



16x2 LCD Pin Description

First two pins of LCD16x2 are used for ground and supply (+5 V).

Pin 3 - VEE pin

This pin is used for adjusting the contrast of the display. Voltage on this pin defines contrast on display, lower the voltage, higher the contrast. We can connect 4.7 k pot for contrast adjustment or simply connect this pin to ground to get maximum contrast.

Pin 4 –RS: Register Select pin

RS = 0: Data on the D0 to D7 pins is considered as a command.

RS = 1: Data on the D0 to D7 pins is considered as data to display on LCD16x2.

Pin 5 – RW: Read / Write pin

RW = 0: Write data to the LCD

RW = 1: Read data from the LCD

Pin 6 –E: Enable

This pin is used to latch the data present on the data pins D0 to D7. High to low pulse with a minimum width of 450 ns is required to latch the data to the display.

Pins 7:14 - DATA pins D0 to D7

Data pins are used to send data/command to the LCD16x2 as parallel 8 data bits.

Pin 15:16 - LED + and LED -

Liquid Crystal Displays don't have their own light like seven segment displays. Therefore, the module has a backlight LED. Supply to this LED is provided through these pins.

Specification of LCD16x2

1. Display Type: Alphanumeric character display
2. Character Format: 5x8 dots matrix format
3. Display Size: 16 characters x 2 lines
4. Display Color: Blue or Green
5. Backlight: LED backlight
6. Voltage Supply: 5V DC
7. Operating Temperature: -20°C to +70°C
8. Interface: 4-bit or 8-bit mode
9. Dimension: 84.0 x 44.0 x 13.0 mm

LCD 16x2 Commands

While interfacing an LCD16x2 with any microcontroller, firstly we need to initialize the LCD. For that, we need to send some commands. Similarly, to clear the display or for changing the position we need to send commands. So basically, we can say that LCD16x2 is controlled by using commands.

Tableau III.2: Tableau de brochages d'un Afficheur LCD.

1	VSS	Alimentation positive +5V de l'afficheur « Power supply »
2	VDD	
3	VEE	Réglage du contraste de l'afficheur entre 0 et +5V
4	RS	Commutation de registre entre les instructions « 0 » et les données « 1 »
5	RW	Commutation entre lecture « 1 » (Read) et écriture « 0 » (Write)
6	E	Entrée de validation (Enable) active sur front descendant
7	D0	Bus de données bidirectionnel 3 états (haute impédance lorsque E=0) “NON UTILISE”
8	D1	
9	D2	
10	D3	
11	D4	Bus de données bidirectionnel 3 états (haute impédance lorsque E=0) “UTILISE”
12	D5	
13	D6	
14	D7	

LCD 16x2 4-bit Mode Configuration

Only four GPIO pins are connected to LCD data (D4-D7) pin which helps to save GPIO pins. By default, LCD16x2 is in 8-bit mode. To use LCD16x2 in 4-bit mode, we need to send some commands for LCD initialization and configuration. To send these commands following data formats should follow.

D7	D6	D5	D4	D3	D2	D1	D0
0	0	1	DL	N	F	-	-

DL: Interface Data length

0 = 4-bit

1 = 8-bit

N: Number of display lines

0 = 1 line

1 = 2 line

F: Character Font

0 = 5x8 dots

1 = 5x10 dots

D1:D0: These bits are set to 0.

D7:D6: These bits are set to 0.

D5: This bit should set to 1.

Note: To configure LCD 16x2 in a 4-bit mode, we should send 2 at higher nibble (D4 - D7) to LCD 16x2. As only four (D4 - D7) data lines need to be connected to the microcontroller, we need to send 0x02 as a command, which will write 0 to higher nibble (D4-D7) and then again write 2 to higher nibble, which will set the LCD in 4-bit mode. Note that we can provide any command with lower nibble as 2 i.e. 0x32, 0x42 etc. except 0x22 to initialize display in 4-bit mode.