

الجمهورية الجزائرية الديمقراطية الشعبية
République algérienne démocratique et populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique
جامعة عين تموشنت بلحاج بوشعيب
Université -Ain-Temouchent- Belhadj Bouchaib
Faculté des Sciences et de la Technologie
Département de génie électrique



Projet de fin d'études
Pour l'obtention du diplôme de Master en :
Domaine : SCIENCES ET DE LA TECHNOLOGIE
Filière : Electrotechnique
Spécialité : Commandes Electriques
Thème

***Création un prototype de commande d'un ascenseur à base
d'un microcontrôleur de type ' Arduino Uno R3 '***

Présenté Par :

- 1) Melle BELOUFA Wafaa
- 2) Melle BELGHOMARI Djamila

Devant le jury composé de :

Dr. TOUHAMI A	M A A	UAT.B.B (Ain Temouchent)	Président
Dr. MENDAZ K	M C A	UAT.B.B (Ain Temouchent)	Examineur
Dr. ZEGAI Med. L	M C B	UAT.B.B (Ain Temouchent)	Encadrant

Année universitaire 2020/2021

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.ZEGAI Med.L 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

Avant tous, je remercie dieu tout puissant de m'avoir donné le courage et la patience pour réaliser ce travail.

Je dédie ce modeste travail à :

- Mes très chers parents, pour leur sacrifices, et qui n'ont jamais cessé de m'encourager que dieu me les garde.*
- Mes chères sœurs, mon cher frère.*
- Tous mes amis(es).*

BELOUFA Wafaa

Dédicace

Avant tous, je remercie dieu tout puissant de m'avoir donné le courage et la patience pour réaliser ce travail.

Je dédie ce modeste travail :

A ma mère qui m'a doté d'une éducation digne, son amour à fait de moi ce que je suis aujourd'hui. Je ne pourrais jamais oublier la tendresse et l'amour dévoué par lesquels ils ont toujours entourer depuis mon enfance, que dieu te protège pour moi.

A la mémoire de mon père

Depuis ma tendre enfance, tu es mon plus fort repère un être unique et magnifique, le meilleur des pères, quand je pense à toi les larmes montent avec les souvenirs, merci pour ce que tu as été pour moi. Pais à ton âme.

A mes frères et leurs femmes qui m'ont toujours encouragé.

A celui qui m'a soutenu et encouragé tout au long de ce projet mon fiancé Amine.

A tout mes amis.

BELGHOMARI Djamila

Sommaire

Introduction générale	1
Chapitre I : Généralité sur Les Ascenseurs	
I.1 Introduction	2
I.2 Définition	2
I.2.1 Ascenseur	2
I.2.2 Monte-charge	2
I.2.3 Monte-charge industriel	2
I.2.4 Monte voitures	3
I.2.5 Ascenseur pour le transport de personnes handicapées	3
I.2.6 Ascenseur sur plan incliné.....	3
I.3 Les familles d'ascenseurs.....	3
I.4 Les ascenseurs hydrauliques	4
I.4.1 Description.....	4
I.4.2 Énergie	5
I.4.3 Avantages et inconvénients	5
I.4.4 Principe de fonctionnement	5
I.5 Les ascenseurs à traction à câble	6
I.5.1 Description.....	6
I.5.2 Énergie	7
I.5.3 Avantages et inconvénients	7
I.5.4 Différentes parties d'un ascenseur à traction	8
I.5.5 Principe de fonctionnement d'un ascenseur à traction.....	13
I.6 Système de motorisation d'un ascenseur à traction.....	14
I.6.1 Les moteurs-treuil à vis sans fin à une ou deux vitesses	14
I.6.2 Les moteurs-treuil planétaires	15
I.6.3 Les moteurs à attaque directe « Gearless » ou « sans treuil »	15

Sommaire

I.7 Critère de choix de la motorisation.....	16
I.8 Commande d'un ascenseur.....	17
I.8.1 Armoire de manoeuvre.....	17
I.8.2 La logique câblée.....	18
I.8.2.1 Système de control à relais.....	18
I.8.3 Commande par carte à microprocesseur/microcontrôleur.....	19
I.9 Les critères du choix du type d'ascenseur.....	20
I.10 Conclusion.....	20

Chapitre II : Présentation de La Carte Arduino et son Environnement de Programmation

II.1 Introduction.....	21
II.2 Définition d'Arduino.....	21
II.3 Les gammes de la carte Arduino.....	22
II.4 Description de la carte Arduino UNO.....	23
II.5 Alimentation de la carte Arduino.....	27
II.6 Les Caractéristiques de la carte Arduino UNO.....	28
II.7 L'environnement de la programmation (logiciel IDE).....	28
II.7.1 Description.....	28
II.7.2 Structure du programme.....	30
II.7.3 Syntaxe du langage Arduino.....	32
II.7.4 Compilation et Téléversement.....	33
II.8 Conclusion.....	34

Chapitre III : Conception et Réalisation

III.1 Introduction.....	35
III.2 Partie d'étude.....	35
III.2.1 Le moteur à courant continue.....	35
III.2.1.1 Description.....	35

Sommaire

III.2.1.2 Principe de fonctionnement	35
III.2.1.3 Le pont en H.....	37
III.2.1.4 Le L293D	39
III.2.2 Détecteur de fin de course.....	41
III.2.2.1 Utilisation.....	41
III.2.2.2 Principe de fonctionnement	41
III.2.3 Bouton poussoir.....	41
III.2.3.1 Description	41
III.2.3.2 Fonctionnement	42
III.2.3.3 Bouton poussoir normalement ouvert (NO)	42
III.2.3.4 Bouton poussoir normalement fermé (NF)	42
III.2.4 Afficheur 7 segments.....	43
III.2.4.1 Définition	43
III.2.4.2 Des LED.....	43
III.2.4.3 Cathode commune ou Anode commune	44
III.2.4.4 Branchement de l'afficheur a cathode commune.....	44
III.2.4.5 Problématique liée au l'afficheur 7 segment avec l'Arduino Uno	45
III.2.4.6 Décodeur BCD – 74LS48	45
III.2.4.7 La configuration des pins ou des broches du 74LS48	46
III.2.5 Ecran LCD	46
III.2.5.1 Description	46
III.2.5.2 Fonctionnement	47
III.2.5.3 Brochage d'un afficheur LCD.....	47
III.2.5.4 La liaison I2C	49
III.2.6 Capteur de poids	50
III.3 Partie pratique	51
III.3.1 Partie logiciel	51

Sommaire

III.3.1.1 PROTEUS	51
III.3.1.2 Fritzing	52
III.3.1.3 Schéma synoptique	53
III.3.1.4 Schéma de commande globale	56
III.3.2 Partie Programmation	57
III.3.2.1 Organigramme.....	57
III.3.2.2 Cahier des charges	58
III.3.2.3 Vérification de fonctionnement.....	58
III.3.3 Partie réalisation	59
III.4 Conclusion	59
Conclusion générale	60

Annexes

Bibliographie

Liste des figures

Figure I-1	Les deux types d'ascenseur	3
Figure I-2	Les différents modèles des ascenseurs hydrauliques	4
Figure I-3	Principe de fonctionnement d'un ascenseur hydraulique	6
Figure I-4	Les deux types d'ascenseur à traction à câble	6
Figure I-5	Les composants d'un ascenseur à traction électrique.....	8
Figure I-6	Machine de traction	9
Figure I-7	Système barrage.....	13
Figure I-8	Principe de fonctionnement d'un ascenseur à traction	13
Figure I-9	Moteur-treuil à vis sans fin	14
Figure I-10	Moteur-treuil planétaire	15
Figure I-11	Moteur gearless	16
Figure I-12	Armoire de commande a relais électromécaniques	18
Figure I-13	Carte à microprocesseur.....	19
Figure II-1	Carte Arduino UNO	24
Figure II-2	La fenêtre principale du logiciel Arduino.....	29
Figure II-3	Barre de Bouton	30
Figure II-4	Définition des constantes et des variables	30
Figure II-5	Configuration des Entrées/Sorties	31
Figure II-6	Programmation des interactions et comportements.....	31
Figure II-7	Compilation.....	34
Figure III-1	Moteur à courant continu.....	35
Figure III-2	Stator d'un MCC.....	36
Figure III-3	Rotor d'un MCC	36
Figure III-4	Le pont en H	37
Figure III-5	Fonctionnement dans le sens horaire	37
Figure III-6	Fonctionnement dans le sens antihoraire.....	38
Figure III-7	Freinage avec S1 et S2	38
Figure III-8	Freinage avec S3 et S4	38
Figure III-9	Pont en H avec ses diodes de protection	39
Figure III-10	Configuration des broches du L293D	40
Figure III-11	Interrupteur fin de course	41

Figure III-12	Bouton poussoir	41
Figure III-13	Schéma d'un BP (NO).....	42
Figure III-14	Schéma d'un BP (NF)	42
Figure III-15	Afficheur 7 segments	43
Figure III-16	Schéma électrique interne d'un afficheur segments anode commune	43
Figure III-17	Représentation des Afficheur 7 segments identifier avec des lettres et des numéros	44
Figure III-18	Exemple du branchement d'un afficheur à cathode commun	45
Figure III-19	Décodeur BCD 74LS48.....	45
Figure III-20	Configuration des broches du décodeur BCD	46
Figure III-21	Afficheur LCD16*2	47
Figure III-22	Schéma d'un afficheur LCD	47
Figure III-23	Le circuit PCF8574	49
Figure III-24	Capteur de poids	50
Figure III-25	Fenêtre du module de simulation Proteus ISIS.....	51
Figure III-26	Fenêtre du module de simulation Proteus ARES.....	52
Figure III-27	L'interface de Fritzing.....	52
Figure III-28	Schéma synoptique de la commande d'ascenseur	53
Figure III-29	Branchement des boutons poussoirs	54
Figure III-30	Branchement des fins de courses	54
Figure III-31	Branchement de moteur.....	55
Figure III-32	Branchement de l'afficheur 7 segment.....	55
Figure III-33	Branchement l'afficheur LCD	55
Figure III-34	Branchement de capteur de poids	56
Figure III-35	Schéma de commande de l'ascenseur	56
Figure III-36	Maquette de l'ascenseur	59

Liste des tableaux

Tableau II-1	Les gammes d'Arduino..	22
Tableau II-2	Les différentes broches de la carte ARDUINO UNO..	27
Tableau II-3	Les commandes du programme Arduino...	32
Tableau II-4	Les fonctions du programme Arduino...	33
Tableau II-5	Les différentes variables du programme Arduino...	33
Tableau III-1	Brochage d'un afficheur LCD.....	48
Tableau III-2	Les pins de circuit PCF8574..	50

Introduction générale

Avec le développement de l'urbanisation de l'être humaine, l'ascenseur est devenu désormais un système indispensable pour répondre aux exigences modernes de notre vie en matière d'autonomie, de mobilité, d'accessibilité et de rapidité. Il est aussi un élément essentiel des immeubles résidentiels, des bureaux, des entreprises, dans les administrations, des hôtels, des hôpitaux etc..... Il contribue ainsi à gagner du temps et faciliter les déplacements.

En effet, ce projet consiste à étudier la réalisation d'une carte de commande d'un ascenseur à base d'un microcontrôleur 'Arduino Uno R3'.

Ce mémoire présente le déroulement du projet et permet de suivre la progression de notre travail ainsi il aura une validation totale pour le cahier de charge donné et des propositions par la suite réservés pour les améliorations possibles.

L'objectif principal de ce projet est de construire une carte électronique pour commander un ascenseur à quatre niveaux. Et pour cela, cette manuscrite sera composée de trois chapitres :

- Le premier chapitre sera consacré sur des descriptifs généraux sur l'ascenseur. Les différents types et leurs modes de fonctionnement, aussi nous avons montré les avantages et les inconvénients de chaque type, ensuite nous avons abordé les différentes parties de l'ascenseur de traction qui sera notre intérêt à ce projet et sur système de motorisation, et à la fin, nous avons cité les critères du choix du type d'ascenseur.
- Le deuxième chapitre, présente une étude basée sur la carte Arduino Uno et son environnement de programmation.
- Et le dernier chapitre, présente le contenu du cahier des charges ainsi que les différents logiciels et composants utilisés pour avoir la réalisation de carte de commande.
- Nous terminerons notre travail avec une conclusion générale et des perspectives considérés comme des propositions dans les futurs projets de ce sujet.

Chapitre I :

Généralité sur les ascenseurs

I.1 Introduction :

L'ascenseur était considéré comme un luxe des bourgeois dans les vieilles époques, mais actuellement, il devient aussi un élément indispensable dans la vie quotidienne.

Les ascenseurs constituent toujours un moyen de translation et de confort pour l'homme lors de ces déplacements dans de hauts immeubles. Les usagers de ces dispositifs doivent se déplacer en sécurité.

L'objectif de ce chapitre, est de présenter des généralités sur les ascenseurs et les différentes parties qui les constituent.

I.2 Définition :

I.2.1 Ascenseur :

C'est un appareil élévateur installé dans un immeuble à des endroits bien précis, afin de permettre aux usagers de se déplacer entre différents niveaux. Son déplacement se fait le long des guides verticaux dont l'inclinaison est inférieure à 15° [1].

I.2.2 Monte-charge :

C'est un appareil élévateur installé à demeure, desservant des niveaux définis, comportant une cabine inaccessible aux personnes par ses dimensions et sa constitution, se déplaçant au moins partiellement le long des guides verticaux ou dont l'inclinaison sur la verticale est inférieure à 15°. Pour satisfaire à la condition d'inaccessibilité, les dimensions de la cabine doivent être au plus égales à:

- Surface = 01 m²
- Hauteur = 01,20 m. [Une hauteur de plus de 1,20m peut toutefois être admise si la cabine comporte plusieurs compartiments fixes répondant chacun aux conditions ci-dessus] [1].

I.2.3 Monte-charge industriel :

C'est une appareil de levage installée à demeure, 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°, dont la commande ne peut se faire que de l'extérieur, et qui est interdit au transport de personnes [1].

I.2.4 Monte voitures :

Ascenseur dont la cabine est dimensionné pour le transport de véhicules automobiles de tourisme. Si les voitures sont accompagnées par des personnes, la réglementation régissant l'utilisation de ces appareils est identique à celle des ascenseurs [1].

I.2.5 Ascenseur pour le transport de personnes handicapées :

Toute installation installée à demeure, construite et utilisée principalement pour le transport des personnes handicapées, debout ou en fauteuil roulant, avec ou sans accompagnateur [1].

I.2.6 Ascenseur sur plan incliné :

Tout ascenseur sur plan incliné utilisé principalement au transport de personnes à mobilité réduite. Cet appareil peut être équipé 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'extérieur des bâtiments dans des rampes ou escaliers d'accès [1].

I.3 Les familles d'ascenseurs :

Il existe deux types de familles d'ascenseurs :

- ✓ Les ascenseurs hydrauliques ;
- ✓ Les ascenseurs à traction câble (électrique).

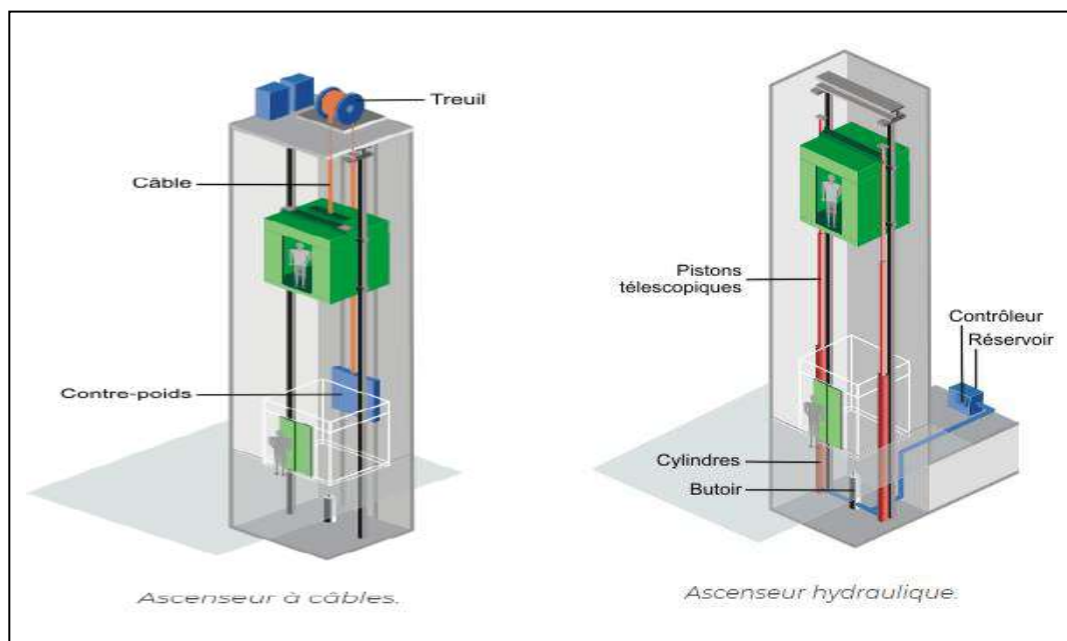


Figure I-1 : Les deux types d'ascenseur [2].

En général, ces deux types utilisent l'énergie électrique pour déplacer verticalement la cabine d'ascenseur, cependant, les ascenseurs hydrauliques sont nettement moins utilisés que les ascenseurs à treuil [2].

I.4 Les ascenseurs hydrauliques :

I.4.1 Description :

Les ascenseurs hydrauliques sont utilisés en général pour satisfaire des déplacements relativement courts de l'ordre de 15 à 18 m maximums.

Nous citerons les types ascenseurs hydrauliques :

1. À cylindre de surface ;
2. À cylindre enterré ;
3. Télescopiques à cylindre de surface. [Ce type d'ascenseur n'est pas très présent sur le marché] [2].

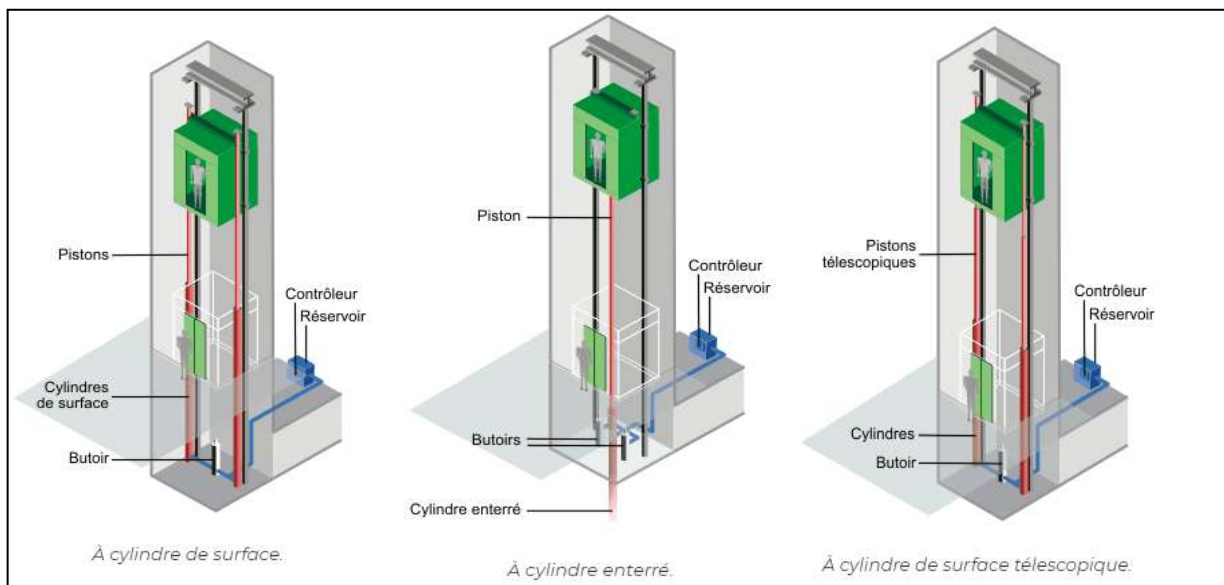


Figure I-2 : Les différents modèles des ascenseurs hydrauliques [2].

Les ascenseurs hydrauliques se composent principalement de :

1. D'une cabine ;
2. De guides ;
3. D'un ensemble pistons-cylindres hydrauliques placé sous la cabine de l'ascenseur ;
4. D'un réservoir d'huile ;
5. D'un moteur électrique accouplé à une pompe hydraulique ;
6. D'un contrôleur.

Les différents modèles permettent de tenir compte de critères :

1. De place ;
2. De hauteur d'immeuble à desservir ;
3. De stabilité de sol et de sous-sol ;
4. De risque de pollution par rapport au sol et plus spécifiquement aux nappes phréatiques ;
5. D'esthétique [2].

I.4.2 Énergie :

Énergétiquement parlant les ascenseurs hydrauliques posent un problème dans le sens où il n'y a pas de contrepoids qui équilibre la cabine comme dans les systèmes à traction à câble par exemple [2].

I.4.3 Avantages et inconvénients :

Ci-dessous, on trouvera les principaux avantages et inconvénients des ascenseurs hydrauliques :

Les avantages :

1. Précision au niveau du déplacement ;
2. Réglage facile de la vitesse de déplacement ;
3. Ne nécessite pas de salle de machinerie ;
4. Implantation facile dans un immeuble existant.

Les inconvénients :

1. Course verticale limitée à une hauteur entre 15 et 18 m ;
2. Risque de pollution du sous-sol ;
3. Consommation énergétique importante ;
4. Nécessiter de renforcer la dalle de sol [2].

I.4.4 Principe de fonctionnement :

Comme toute machine hydraulique 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 [2].

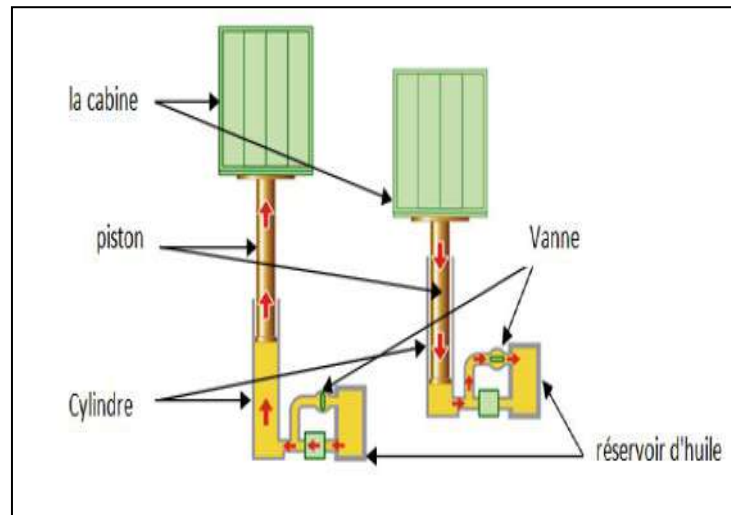


Figure I-3 : principe de fonctionnement d'un ascenseur hydraulique [2].

I.5 Les ascenseurs à traction à câble :

I.5.1 Description :

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 :

1. À moteur-treuil à vis sans fin ;
2. À moteur-treuil planétaire ;
3. À moteur à attaque directe (couramment appelé "Gearless" ou sans treuil) [2].

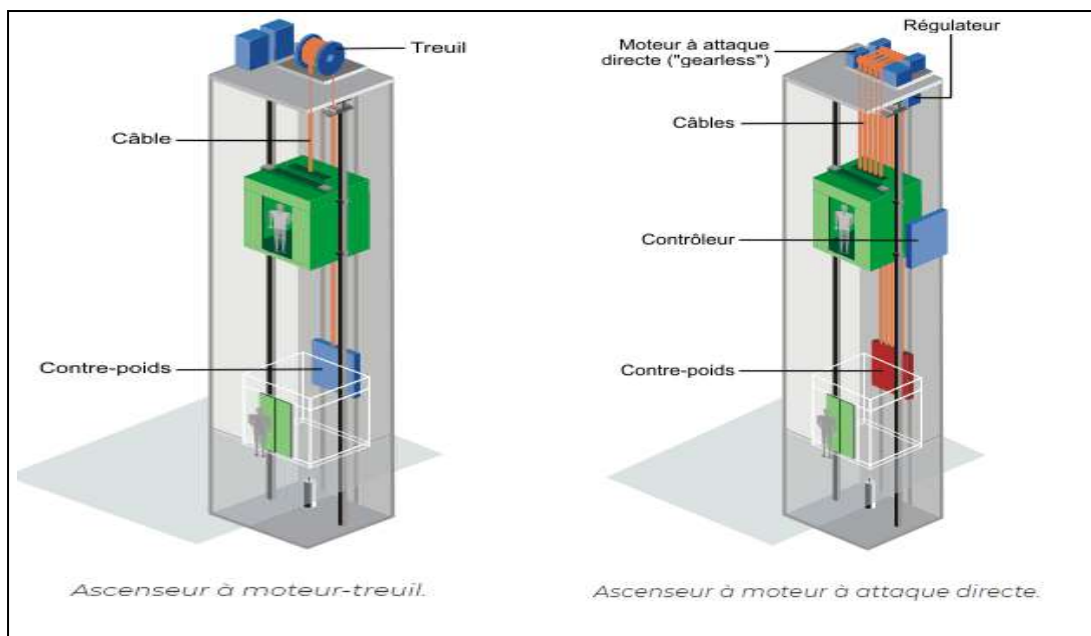


Figure I-4 : Les deux types d'ascenseur à traction à câble [2].

Quel que soit le type, les ascenseurs à traction à câbles comprennent généralement :

1. Une cabine ;
2. Un contrepoids ;
3. Des câbles reliant la cabine au contrepoids ;
4. Des guides ;
5. Un système de traction au-dessus de la cage de l'ascenseur [2].

I.5.2 Énergie :

Énergétiquement parlant les ascenseurs à traction à câbles sont plus intéressants que les ascenseurs hydrauliques dans le sens où le contrepoids réduit fortement la charge quelle que soit le type de motorisation. Les consommations et les courants de démarrages sont réduits par rapport aux ascenseurs hydrauliques [2].

I.5.3 Avantages et inconvénients :

Ci-dessous, on trouvera les principaux avantages et inconvénients des ascenseurs à câbles :

Les avantages :

1. Course verticale pas vraiment limitée ;
2. Suivant le type de motorisation précision au niveau de la vitesse et du déplacement ;
3. Rapidité de déplacement ;
4. Efficacité énergétique importante ;
5. Pas de souci de pollution.

Les inconvénients :

1. En version standard, nécessite une salle de machinerie;
2. Exigence très importante sur l'entretien [2].

I.5.4 Différentes parties d'un ascenseur à traction :

Les ascenseurs à traction sont en général constitués de ces parties :

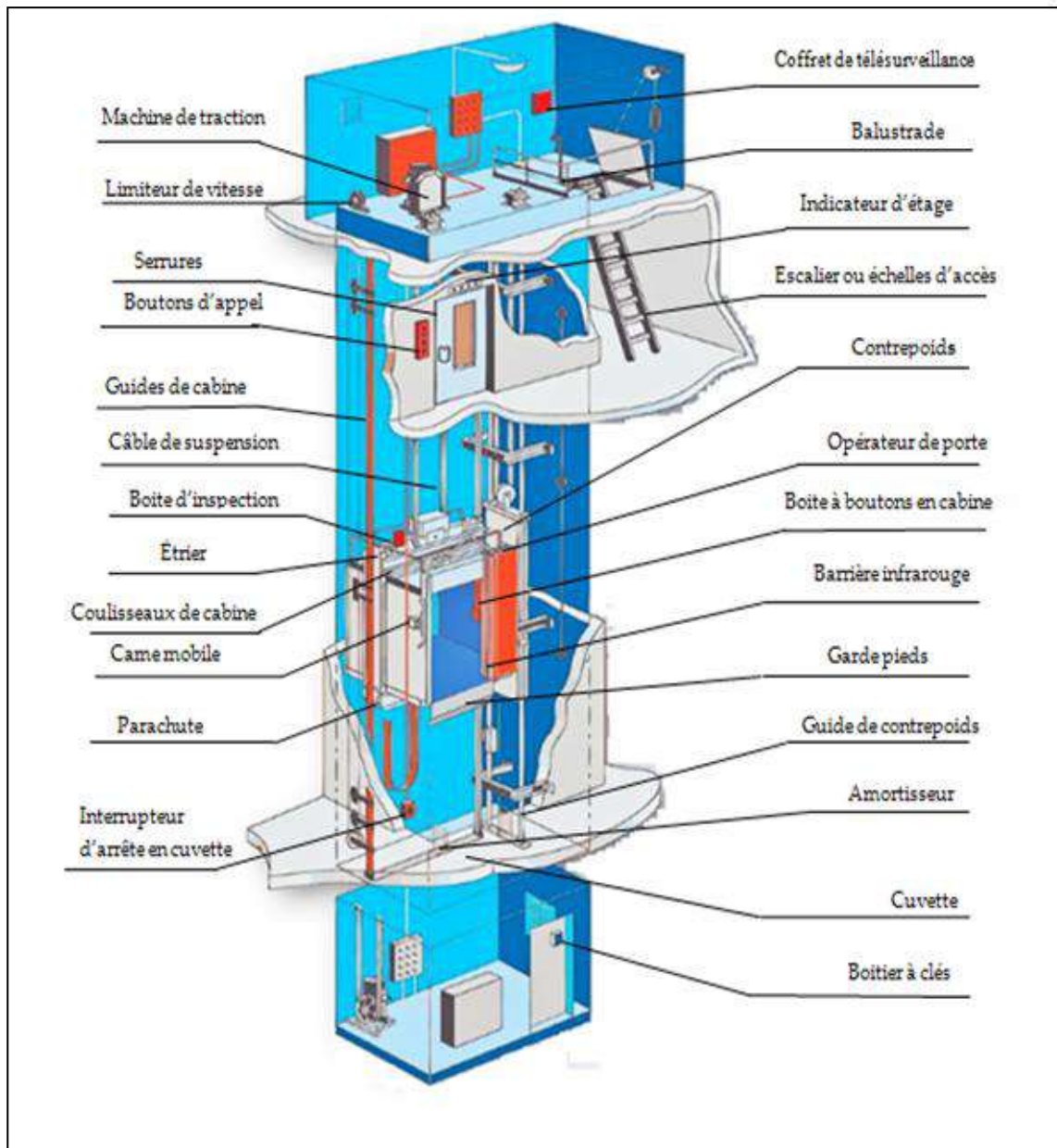


Figure I-5 : Les composants d'un ascenseur à traction électrique [11].

Nous nous limitons aux composants d'un ascenseur à traction, car il représente la majorité des ascenseurs qui existent sur le marché.

- **Machine de traction (treuil):** Machine composée généralement d'un dispositif de freinage, d'un moteur, d'un réducteur et d'une poulie de traction et destinée à actionner les câbles de traction de l'ascenseur [1]-[10].

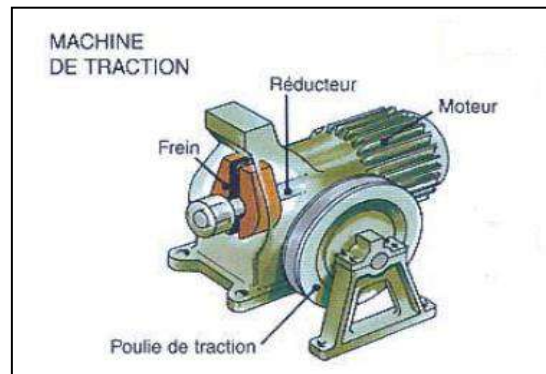


Figure I-6 : Machine de traction (Moteur-Réducteur) [9].

Le treuil est constitué d'un cylindre appelé tambour qui est actionné par le moteur, d'un câble qui s'enroule autour d'un cylindre (tambour) et qui est actionné par un moteur, il est fixé à une de ces extrémités à l'étrier qui soutient la cabine, il passe sur la poulie du treuil et relié à son autre extrémité relié au contre poids [4].

Elle est composée d'éléments suivants :

- **Moteur de traction** : moteur équipant le treuil de l'ascenseur et placé dans la machinerie [1].
 - **Réducteur** : le réducteur de treuil est une boîte de vitesse composée soit d'une vis et d'une couronne soit d'un réducteur planétaire contenus dans un carter rempli d'huile. Son rôle consiste à démultiplier la vitesse du moteur électrique pour la rendre compatible avec les conditions de l'utilisation de l'ascenseur. Pour assurer la lubrification de l'ensemble, la couronne baigne généralement dans l'huile du carter et par sa rotation ramène d'huile vers les autres organes [1].
 - **Poulie de traction** : poulie équipée généralement de gorges taillées en forme de V de manière à agripper les câbles de traction. Cette poulie, solidaire du treuil, fait lors de sa rotation, déplacer l'ensemble cabine et contrepoids [1].
 - **Le Frein** : Le frein permet d'arrêter le tambour, ce frein est constitué d'une mâchoire articulée qui vient serrer le tambour, Lors d'un fonctionnement normal, il est commandé par un électroaimant. En cas de panne, nous pouvons le débloquent manuellement à l'aide d'un levier, ce qui permet de déplacer la cabine lorsque des usagers sont bloqués dans celle-ci, par exemple en cas de coupure de courant [4].
- **Limiteur de vitesse** : Organe mécanique, si la vitesse dépasse anormalement la vitesse maximale autorisée, les masselottes se lèvent et coupent un contact de sécurité [5].

- **Cabine d'ascenseur:** Destinée à accueillir les personnes et les marchandises, elle est constituée de quatre parties principales qui dans l'ordre de leur montage pendant l'installation sont:
- **L'étrier (suspension cabine)** : Il est constitué de deux cadres en acier, réunis entre eux de telle sorte que l'ensemble est symétrique par rapport à l'axe transversal des guides. Chaque cadre comporte à la portée inférieure et supérieure deux traverses horizontales réunies par deux profilés verticaux parallèles aux guides. A la traverse supérieure sont attachés les câbles de suspension. Le traverse inférieure est constituée en générale d'une ou de plusieurs plaques soudées et les butées des amortisseurs. L'étrier est le premier élément mis en place pendant l'installation de la cabine.
 - **Le plancher** : Le socle (plate-forme) se monte directement sur les traverses de l'étrier auxquelles il est fixé par des vis. Il est formé également d'un cadre en acier. Sur le socle est disposé le plancher de la cabine. Le plancher est mise en place en fin de montage.
 - **Les parois** : Sont le plus souvent en tôles pliées ou en bois. Après assemblage des parois et de l'encadrement de la baie de la cabine, le toit est mis en place.
 - **Le toit** : Est le plus souvent assemblé en atelier et monté d'un seul bloc. Le toit de la cabine est formé d'un cadre en tôles pliées ou cintré [1]-[7].
- **Porte de cabine:** Porte à fermeture généralement automatique destinée à confiner l'utilisateur dans la cabine pendant le déplacement de celle-ci, lui interdisant tous contacts avec les parties extérieures à la cabine [1].
- **Porte palières:** Une porte extérieure fixe est installée à chaque étage, isolant les personnes extérieures sur le palier du puits de gaine et de la cabine éventuellement en mouvement. Elles peuvent être battantes ou coulissantes et commandées manuellement ou automatiques. Elles doivent être équipées d'un dispositif empêchant leurs ouvertures si la cabine n'est pas sur le niveau et bloquant le départ pendant leur ouverture [1].
- **Serrures:** La serrure est l'élément qui sécurise un ascenseur pour le public. Son rôle est de verrouiller mécaniquement les portes palières. Elle possède également un contact électrique qui contrôle que le verrouillage mécanique est bien en place [7].
- **Came mobile:** Le rôle de cette organe est d'agir sur les galets de la serrure afin d'en effectuer le déverrouillage, mais seulement lorsque la cabine s'y arrête. Il est constitué essentiellement d'un électroaimant pouvant attirer une armature qui porte une came [1].
- **Boutons d'appels:** Ce sont les boutons qui commandent l'arrivée de la cabine, ils sont installés aux paliers [1].

- **Boutons d'envois:** Ce sont les boutons de précision d'étage; ils sont installés dans la cabine [1].
- **Charge utile:** C'est la charge pour laquelle l'appareil a été construit, elle varie en fonction de la surface de la cabine. Au-delà de cette capacité, le système de traction n'est plus en mesure de contrôler le déplacement et l'arrêt correct de la cabine. Dans certains cas de surcharge exagérée, des blocages intempestifs peuvent se produire [1].
- **Garde (chasse) pieds:** est une tôle fixe ou rétractable, destinée à protéger les chutes en gaines lorsque la cabine est immobilisée en dehors de la zone de déverrouillage [7].
- **Contrepoids:** Élément destiné à contre balancer le poids de la cabine (lorsque la cabine monte, le contre poids descend et vis-versa). Il est constitué d'un étrier (suspension) métallique et des masses de fonte appelées gueuses de contrepoids. La masse du contrepoids (P_{cp}) est plus lourde que l'ascenseur, elle représente l'équivalent du poids de la cabine (P_{ca}) et la moitié de sa charge utile (P_{ch}), ce calcul est donné par cette formule $P_{cp} = P_{ca} + P_{ch} / 2$ [1].
- **Gaine ou trémie:** (Appelée aussi pylône) c'est une gaine verticale dans laquelle se déplace l'ascenseur et son contrepoids. Elle est équipée de rails de guidages (guides) des éléments mobiles (cabine et contrepoids). Elle doit être fermée sur toute sa hauteur [1]-[7].
- **Guides:** Profilés en acier, situés de part et d'autre, le long de la course de la cabine, ils sont habituellement en forme de T. Ils sont destinés à guider la cabine et le contrepoids dans la gaine [1].
- **Coulisseau:** Ils sont situés à chaque coin de l'étrier, et sont en appui sur les guides. Durant le déplacement de la cabine, ceux-ci glisse sur les guides, huilés régulièrement pour limiter les frottements et les accrocs, et donc le bruit, et augmenter le confort [7].
- **Cuvette:** Partie de la gaine située en bas du niveau d'arrêt inférieur desservi par la cabine, contenant les poulies de renvoi et les amortisseurs [3].
- **Amortisseurs:** Les amortisseurs sont destinés à assurer le ralentissement et l'arrêt de la cabine en cas de dépassement des fins de course de sécurité [7].
- **Commande de révision (Boitier d'inspection):** La commande de révision est composée d'un boîtier placé sur le toit de la cabine de l'ascenseur. Ce boîtier, équipé de bouton de marche montée et descente ainsi que d'un bouton d'arrêt d'urgence, permet au préposé à l'entretien de manœuvrer, en toute sécurité et à faible allure, l'ascenseur pour inspecter et graisser les organes placés en gaine [1].

➤ **Parachute:** Organe mécanique placé sur la suspension de cabine est commandé par un câble de limiteur. En cas de rupture des câbles de traction ou de survitesse exagérée en descente, le mécanisme du parachute assure un blocage mécanique de la suspension dans les guides évitant la chute libre de la cabine [1].

➤ **Indicateur d'étage :**

- **Dans la cabine :**

Indique la position de l'ascenseur, ainsi que dans la plupart des cas, la direction. Les premiers indicateurs furent "analogiques", les numéros des étages étant tout d'abord un cadran avec une aiguille, puis alignés sur le mur, et chaque nombre s'allumait lorsque l'ascenseur y passait, ainsi qu'une flèche lumineuse qui indiquait la direction de déplacement.

Puis vint l'indicateur LCD numérique, le plus souvent rouge sous une plaque noire. La flèche fut intégrée à l'afficheur.

- **Depuis l'extérieur**

L'étage est souvent indiqué uniquement au rez-de-chaussée, dans ce cas l'utilisateur attendant depuis un étage supérieur ne peut pas savoir à quel niveau la cabine est située, excepté le sens de déplacement qui est parfois indiqué par une flèche qui clignote lorsque la cabine est en déplacement ou qui reste continue lorsque la cabine est à l'arrêt.

L'étage de positionnement est toutefois indiqué sur quelques ascenseurs depuis l'extérieur sur tous les étages.

L'arrivée d'une cabine est signalée par un petit retentissement de sonnerie [6].

➤ **Fin de course :** Contact de sécurité placé généralement en gaine et destiné à stopper l'ascenseur en cas de dépassement de sa course normale [1].

➤ **Cellule de porte:** On appelle cellule l'équipement qui permet de détecter le passage d'une personne devant les portes de la cabine. Dans la majorité des cas, elle est optique.

Principe de fonctionnement: Un émetteur émet une lumière infrarouge (invisible pour les humains) vers le récepteur. Si celui-ci "voit" le faisceau, il en déduit que personne n'est devant la porte. Si le faisceau est coupé, c'est qu'un obstacle se trouve devant. Dans ce cas la porte ne ferme pas.

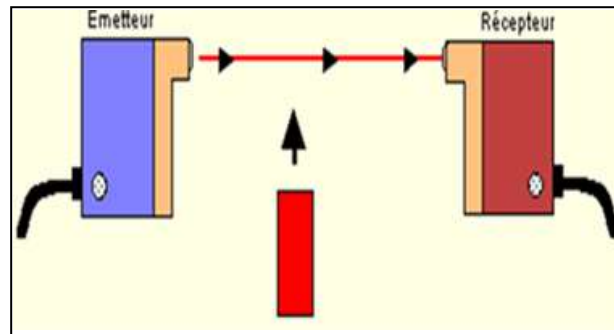


Figure I-7 : système barrage [8].

I.5.5 Principe de fonctionnement d'un ascenseur à traction :

Un ascenseur à contrepoids se compose d'une cabine qui se déplace dans un couloir vertical nommé gaine et qui est guidée par des rails afin d'éviter une collision avec le contre poids. Un frein situé dans la machinerie du moteur permet de stopper la cabine à l'étage demandé. Le déplacement en translation de la cabine est permis par un système de transmission de mouvement. Des câbles, actionnés par un treuil permettent de mettre en mouvement la cabine et le contrepoids. Le moteur du treuil permet la mise en mouvement. Le contrepoids est une charge lourde qui sert à équilibrer la charge de la cabine et à diminuer l'énergie à fournir par le moteur. Lorsque la cabine monte, le contrepoids descend. Le système comprend aussi des organes de commande pour enregistrer les appels des usagers et optimiser les déplacements de la cabine afin de répondre le plus rapidement possible aux différents appels. Enfin, l'ascenseur est équipé d'organes assurant la sécurité des passagers. Des freins d'urgence ou parachutes sont placés de chaque côté de la gaine et se déclenchent en cas de rupture du câble tracteur pour éviter la chute de la cabine. Ils sont déclenchés par un limiteur de vitesse lorsque la vitesse de la cabine est supérieure à la vitesse de déplacement normale (de 2 à 9 km/h selon les ascenseurs). Les parachutes bloquent alors de façon brutale la cabine sur les guides [6].

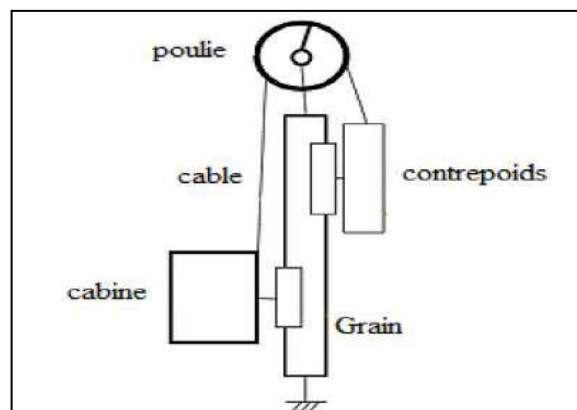


Figure I-8 : Principe de fonctionnement d'un ascenseur à traction [6].

I.6 Système de motorisation d'un ascenseur à traction :

Les ascenseurs à traction à câble se différencient entre eux selon le système de motorisation :

I.6.1 Les moteurs-treuil à vis sans fin à une ou deux vitesses :

Dans ce type de motorisation, la vis sans fin entraîne beaucoup de pertes mécaniques et, par conséquent, des consommations électriques plus importantes. Les moteurs électriques couplés au treuil à vis sans fin étaient généralement des moteurs à courant continu excitation indépendante ou shunt avec la faculté bien connue de pouvoir faire varier très facilement la vitesse de rotation. Les moteurs électriques à courant alternatif sont en principe des moteurs à deux vitesses. On peut remarquer que lorsqu'on se trouve dans la cabine de moteur treuil à deux vitesses :

- Au démarrage, la vitesse est plus lente ;
- Pour atteindre la vitesse de déplacement optimale, le moteur passe en seconde vitesse en provoquant un léger choc d'accélération (passage de petite en grand vitesse).

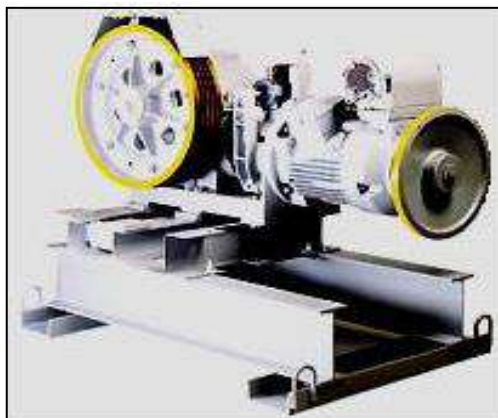


Figure I-9 : moteur-treuil à vis sans fin [8].

Les moteurs ont les principaux avantages et inconvénients qui sont les suivants :

Avantage :

1. Couple élevé ;
2. Grande plage de variation de vitesse ;
3. Précision dans les déplacements et sur la régulation de vitesse.

Inconvénients :

1. Entretien important ;
2. Efficacité énergétique faible ;
3. Consommation électrique non négligeable [8].

I.6.2 Les moteurs-treuil planétaires :

Les appareils à treuil planétaire utilisent le système de réduction de vitesse par engrenages planétaires. Accouplés à un moteur électrique, ils permettent d'avoir un rapport de réduction appréciable pour obtenir une plage de vitesse compatible avec le confort et l'efficacité de déplacement souhaité.

Ce système à un rendement mécanique de l'ordre de 97 à 98 % permettant, pour autant que les moteurs d'entraînement soient performants, d'obtenir des rendements énergétiques globaux intéressants (de l'ordre de 80%).

Les réducteurs planétaires peuvent être accouplés à des moteurs électriques :

- A courant continu (grande plage de variation de vitesse) ;
- A courant alternatif asynchrone à deux vitesses ;
- A courant alternatif asynchrone commandé par un variateur de fréquence.



Figure I-10 : moteur-treuil planétaire [8].

Les moteurs ont les principaux avantages et inconvénient qui sont les suivants

Avantage :

1. Couple important ;
2. Précision dans les déplacements et sur la régulation de vitesse.

Inconvénients :

1. Entretien nécessite une main d'œuvre qualifié [8].

I.6.3 Les moteurs à attaque directe « Gearless» ou « sans treuil» :

Il s'agit d'un moteur sans réducteur, la poulie de traction est monté directement sur l'arbre de sortie du moteur et la régulation de vitesse est obtenue grâce à un variateur de fréquence.

Ce système est énergétiquement performant principalement de par la présence d'un variateur de fréquence qui optimise la consommation énergétique, aussi, les pertes mécaniques sont réduites vu l'absence des engrenages.

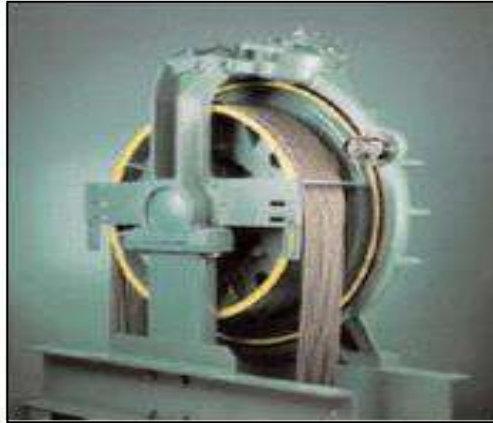


Figure I-11 : moteur gearless [8].

Les moteurs ont les principaux avantages et inconvénient qui sont les suivants :

Avantage :

1. Vitesse optimisé par le variateur de fréquence ;
2. Compacité du système ;
3. Pas de local des machines nécessaire pour les ascenseurs ;
4. Pertes mécaniques réduite ;
5. Faible niveau de bruit ;
6. Poids réduit.

Inconvénients :

1. La capacité peut entrainer des difficultés de maintenance ;
2. Difficulté d'intervention dans la cage d'ascenseur [8].

I.7 Critère de choix de la motorisation :

Il existe 3 critères principaux de choix de motorisation qui sont les suivant :

• **Le rendement global :**

Le rendement global de la motorisation influence le dimensionnement de l'installation et les consommations futures. En effet, à puissance mécanique égale (pour déplacer la charge), meilleur sera le rendement de la motorisation, moins :

- le surdimensionnement du moteur et de l'installation électrique sera important ;
- les consommations énergétiques durant la vie de l'ascenseur seront grandes ;
- les chutes de tension en ligne perturberont le réseau électrique interne voire externe.

Le rendement global d'une motorisation correspond à :

$$\eta_{\text{global}} = \eta_{\text{élec commande}} * \eta_{\text{élec moteur}} * \eta_{\text{méca réducteur}} * \eta_{\text{méca poulie}}$$

Le cas des moteurs à traction avec réducteur de vitesse, par rapport à la même motorisation sans réducteur, montre que les intermédiaires occasionnent des pertes et, par conséquent, réduisent la puissance mécanique disponible à la roue de traction pour une même puissance électrique absorbée.

- **Performances énergétiques :**

La performance énergétique est liée principalement à la gestion des démarrages et des arrêts par le variateur de vitesse. Le variateur de vitesse est un gage de performance énergétique certain de par :

- Le confort renforcé des utilisateurs grâce au démarrage progressif et à la mise à niveau précise ;
- Le contrôle permanent du couple et de la puissance en optimisant les courants de démarrage et les consommations ;
- La possibilité de renvoyer de l'énergie sur le réseau électrique durant le freinage.

- **L'encombrement des équipements :**

Une réduction des coûts d'investissement et un gain de place sont liés à la limitation de l'espace nécessaire à la machinerie.

Il existe aussi des critères secondaires, mais néanmoins importants, prennent en compte le poids, la consommation d'huile, le niveau acoustique(ou sonore) [2].

I.8 Commande d'un ascenseur :

I.8.1 Armoire de manoeuvre :

Le contrôleur ou armoire de manoeuvre:

Le contrôleur est un des composants les plus importants d'un ascenseur. C'est lui qui se charge de mémoriser les appels et envoie la cabine ; c'est lui qui commande le démarrage et l'arrêt du moteur, de l'ouverture de la porte cabine, etc. En fait, c'est le cerveau de l'ascenseur. Dans les anciens modèles d'ascenseur il était constitué d'un assemblage de relais électromécaniques, et de blocs de temporisation, et un appareil appel sélecteur ; le tout étant assemblé suivant la logique câblée. Les ascenseurs modernes ont leurs armoires de manoeuvre qui est constitué de matériel électrique (des cartes électroniques et des relais) [4].

I.8.2 La logique câblée :

La logique câblée permet de réaliser des conditions de fonctionnement dans un système électrique. Elle est réalisée avec des composants reliés entre eux électriquement (fil ou circuit imprimé) selon un schéma conforme au fonctionnement de l'équipement. La logique câblée permet de définir la manière dont les contacts, capteurs relais électromécaniques devront être branchés (en série et/ou en parallèle) afin que le système fonctionne dans les conditions voulues.

I.8.2.1 Système de control à relais :

Ces systèmes électromécaniques utilisaient des contrôleurs logiques à relais de complexité croissante pour contrôler la vitesse, la position de la cabine et l'ouverture et fermeture des portes d'un ascenseur ou d'une rangée d'ascenseurs. Les ascenseurs avec des contrôleurs logiques à relais ont un dispositif appelé "sélecteur". Les sélecteurs mécaniques utilisent des systèmes d'interrupteur et de nombreuses pièces mobiles pour déterminer la position de la cabine. Le sélecteur utilise des bandes magnétiques (souvent appelées ruban sélecteur) qui sont fixées en haut de la cabine. Lorsque l'ascenseur se déplace, il en va de même pour la bande, qui déplace les engrenages mécaniques sur le sélecteur. C'est un dispositif électromécanique se trouvant dans la machinerie et qui montre la position de la cabine dans la gaine. En utilisant les circuits électriques elle est reliée au capteur dans chaque étage. Et joue un rôle important dans le transport de la cabine vers le niveau sélectionné [4].

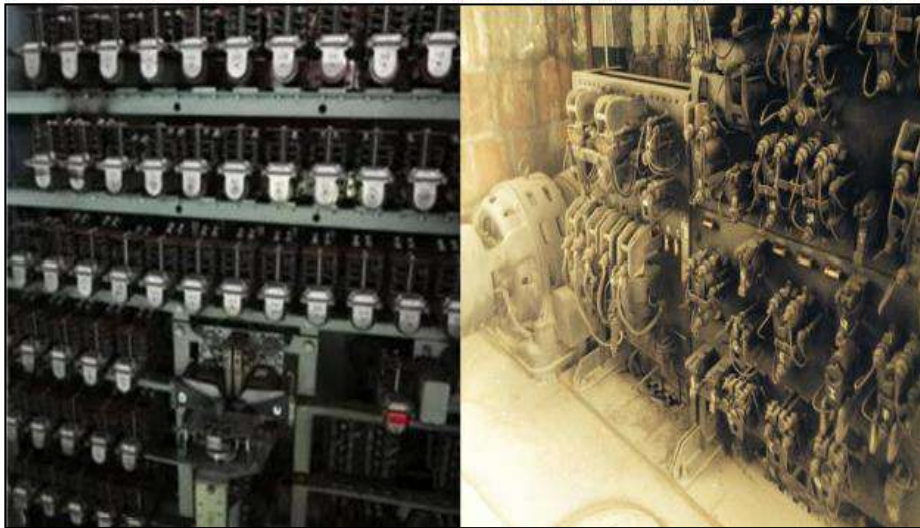


Figure I-12 : Armoire de commande a relais électromécaniques [4].

I.8.3 Commande par carte à microprocesseur/microcontrôleur :

Les ascenseurs dotés de contrôleurs à microprocesseur sont apparus pour la première fois en 1979 lorsqu'Otis a introduit l'ascenseur "Elevonic 101". Ce système d'ascenseur utilise un microprocesseur pour contrôler tous les aspects du fonctionnement de l'ascenseur. Plus tard en 1981, Otis introduit un autre système d'ascenseur entièrement informatisé appelé "Elevonic 401". Au-delà de l'interaction avec les passagers, il existe une série de capteurs, de contrôleurs, de séquences de fonctionnement et de calculs ou algorithmes en temps réel qui équilibrent la demande des passagers et la disponibilité des cabines. Les capteurs d'ascenseur fournissent des données sur les positions de la cabine, la direction de déplacement de la cabine, la charge, l'état des portes, les alarmes, etc. Les contrôleurs d'ascenseur fabriqués dans les années 2000 ou avant sont généralement des contrôleurs logiques programmables (API) qui peuvent être configurés pour une seule ou plusieurs cabines en fonction du nombre d'arrêts et prenant en compte des interfaces de surveillance, des synthétiseurs vocaux, etc. Le contrôleur peut également avoir une fonction permettant de tester les systèmes sans arrêt de l'ascenseur [4].

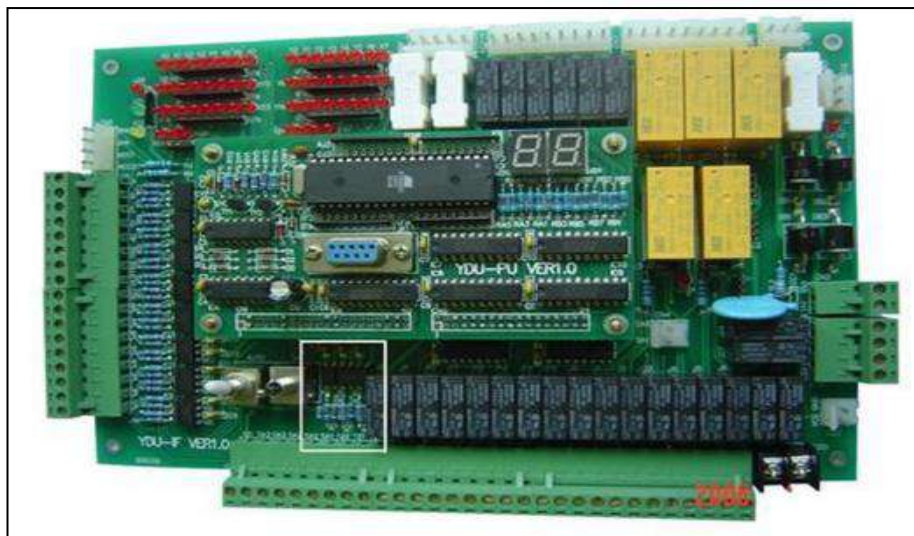


Figure I-13 : Carte à microprocesseur [4].

I.9 Les critères du choix du type d'ascenseur :

En générale, les dépenses énergétiques des ascenseurs ne sont pas la priorité des gestionnaires de bâtiment. En effet la préoccupation première reste avant tout et d'emmener un maximum de monde on toute sécurité avec un maximum de confort.

Le choix de type d'ascenseur (électrique ou hydraulique) est basé sur les critères suivants:

- **Constructifs:** tels que la hauteur du bâtiment, l'espace disponible au niveau des étages, la possibilité de placer une salle des machines au sommet de la gaine, stabilité du terrain.
- **Energie :** en considérant que la consommation de puissance doit être limitée.
- **De sécurité :** par rapport aux risques liés au principe de la technologie car l'utilisation d'une quantité importante d'huile pour les ascenseurs hydrauliques complique la sécurité incendie et augmente le risque de pollution des sols.
- **Organisationnels :** comme le type de fonction de bâtiment, son occupation et son type de fonctionnement en garantissant une performance de confort et de trafic [2].

I.10 Conclusion :

Dans ce chapitre nous avons donné une idée générale sur les ascenseurs avec ses différents types de familles, la différence entre types citées et les avantages et les inconvénients de chacun.

Nous avons basé sur l'étude des ascenseurs à traction électrique puisqu'ils sont les plus utilisés de nos jours grâce à la progression technologique des circuits électroniques, pour cette raison notre travail sera basé sur la réalisation d'une carte commande pour piloter ce genre d'ascenseurs.

Chapitre II :

***Présentation de la carte arduino et son
environnement de programmation***

II.1. Introduction :

Ce chapitre présente la description de la carte Arduino qui sera utilisée pour notre application, Ses caractéristiques avec les différentes catégories de ces cartes, puis on aura basé sur le logiciel IDE qui est l'environnement obligatoire pour les programmés en langages C++.

II.2. Définition d'Arduino :

Arduino est une plate-forme de prototypage d'objets interactifs à usage créatif constituée d'une carte électronique et d'un logiciel de programmation, cet environnement hardware et software permet à l'utilisateur de formuler ses projets par l'expérimentation directe avec l'aide de nombreuses ressources disponibles en ligne.

Arduino est un projet open source (les plans, les schémas, etc. sont accessibles et libres de droits) la grande communauté d'utilisateurs et de concepteurs permet à chacun de trouver les réponses à ces questions et apporte un travail énorme de documentation au projet.

Ces cartes sont faites à base d'une interface entrée/sortie simple et d'un environnement de développement proche du langage [23]-[12].

II.3. Les gammes de la carte Arduino :

Arduino	Type de Microcontrôler	Flash ko	E/S Numériques	PWM	Entrée Analogique	Type d'interface USB
Diecimila	ATmega168	16	14	6	6	FTDI
Due	AtmelSAM3X8E	512	54	12	12	ATmega16U2
Leonardo	ATmega32U4	32	20	7	12	ATmega32U4
Duemilanove	ATmega168/328P	16/32	14	6	6	FTDI
Esplora	ATmega32U4	32	N/A	N/A	N/A	ATmega32U4
Fio	ATmKega328P	32	14	6	8	Aucune
LilyPad	ATmega168V or ATmega328V	16	14	6	6	Aucune
Mega	ATmega1280	128	54	15	16	FTDI
Mega2560	ATmega2560	256	54	15	16	ATmega8U2
Nano	ATmega168	16	14	6	8	FTDI
Uno	ATmega328P	32	14	6	6	ATmega16U2
Zero	ATSAMD21	256	14	12	6	ATmega32U2

Tableau II-1: Les gammes d'Arduino

Parmi ces types, nous avons choisie la carte Arduino UNO.

II.4. Description de la carte Arduino UNO :

ARDUINO UNO, est un circuit imprimé en matériel libre sur lequel se trouve un microcontrôleur ATMEL de type Atmega 328 de la famille AVR pour réaliser des fonctions plus au moins évoluées à un coût bas.

Elle possède une interface USB pour la programmer et l'alimenter. C'est une carte multiplateforme et open-source qui possède un logiciel de programmation, véritable environnement de développement intégré, pour écrire, compiler et transférer le programme vers la carte.

Le chargement de programme dans la mémoire du microcontrôleur se fait de façon très simple via port USB .En outre ; des bibliothèques de fonctions « clé en main » sont également fournies pour l'exploitation d'entrées/sorties courantes : gestion des entrées/sorties TTL ; gestion des convertisseurs ADC ; génération des signaux PWM ; exploitation de bus TWI/12C; exploitation de servomoteurs et d'autres applications...etc.

Elle comporte six entrées analogiques, repérées A0 à A5 (au niveau du Atmega 328 PC0 à PC5, voir la figure II.1). Les connexions sont établies à travers de connecteurs femelles HE14 situés sur le dessus de la carte, peuvent admettre toute tension analogique comprise entre 0 et 5V.

Ces entrées analogiques sont gérées par un convertisseur analogique /numérique de 10 bits dont la sortie peut varier de 0 à 1023. Les entrées A4 et A5 peuvent également être utilisées respectivement comme la ligne de donnée SDA et la ligne d'horloge SCL de l'interface série 12C.

La carte ARDUINO UNO dispose aussi 14 entrées/sorties numériques chacune est repérée de 0 à 13 et dont 6 pouvant être utilisées comme étant des sorties PWM (Pulse Width Modulation) qui sont les broches numérotées 3, 5, 6, 9,10 et 11 ; elle peut être utilisé en entrée (input) ou en sortie (output) sous le contrôle du programme.

Le sens de fonctionnement pouvant même changer de manière dynamique pendant son exécution. Elles fonctionnent en logique TTL (0V/5V) ; chacune pouvant fournir (source) ou

recevoir un courant maximal de 40 mA et dispose si besoin d'une résistance interne de «pull up». Elle est équipée aussi d'un quartz de 16 MHz, d'un connecteur ICSP (In Circuit Serial Programming) qui permet d'injecter le boot loader à l'intérieur du microcontrôleur, un connecteur jack pour une alimentation extérieure, un bouton de reset pour redémarrer le processeur [8].

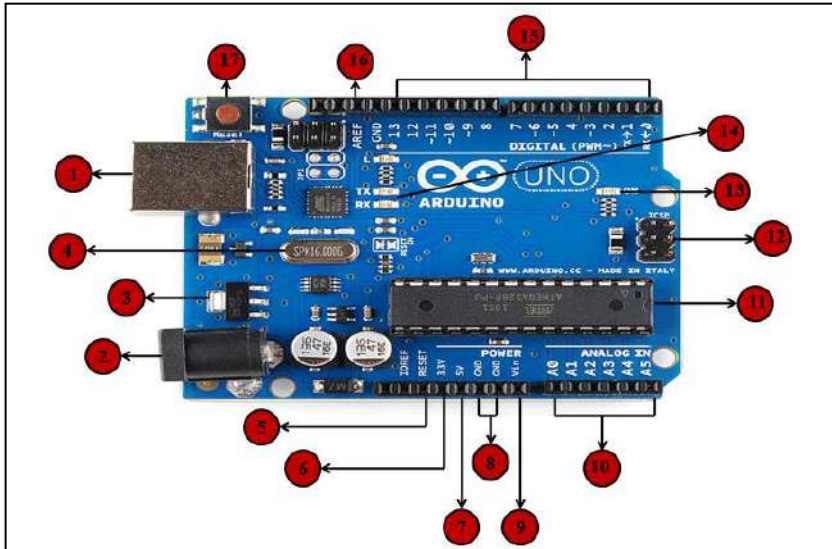













Figure II-1 : Carte Arduino UNO [8].

	<p>Alimentation / Programmation par USB</p> <p>La carte Arduino peut être alimentée avec un câble USB relié à votre ordinateur. Tout ce dont vous avez besoin, c'est de connecter votre carte Arduino à votre ordinateur avec le câble USB type A/B.</p>
	<p>Alimentation via connecteur Jack DC</p> <p>Diamètre interne 2.1mm, externe 5.5mm</p> <p>La carte Arduino peut être directement alimentée par ce connecteur Jack DC. Ce connecteur (2) est relié au régulateur de tension intégré à la carte. L'alimentation via ce connecteur (2) doit être comprise entre 5 et 12 V.</p>
	<p>Régulateur de tension</p> <p>La fonction du régulateur de tension (3) est de contrôler la tension d'alimentation de l'Arduino pour la stabiliser à la bonne tension du microcontrôleur et de chaque élément de la carte. La tension de stabilisation est de 5 Volts sur les cartes UNO.</p>

 <p>4</p>	<p>Oscillateur à quartz</p> <p>Un oscillateur à quartz est un élément électronique qui a la particularité de posséder un quartz à l'intérieur qui vibre sous l'effet piézoélectrique.</p> <p>Les propriétés électromécaniques du quartz sont telles qu'on arrive à faire vibrer le quartz à une fréquence très précise. Cet élément aide l'Arduino UNO à calculer les données de temps. Sur le dessus du composant, on peut lire 16.000H9H. Cela signifie que la fréquence est de 16, 000,000 Hertz, soit 16 MHz.</p>
 <p>5,17</p>	<p>Arduino Reset</p> <p>Vous pouvez redémarrer un Arduino avec un "Reset". Cela aura pour effet de redémarrer votre programme depuis le début. Vous pouvez redémarrer l'Arduino UNO de deux manières : soit en utilisant le bouton "Reset" (17), soit en connectant un bouton externe sur la broche de la carte Arduino mentionnée "RESET" (5).</p>
 <p>6,7,8, 9</p>	<p>Broches (3.3, 5, GND, Vin)</p> <ul style="list-style-type: none"> • 3.3V (6) – Broche d'alimentation de tension 3.3 Volts • 5V (7) – Broche d'alimentation de tension 5 Volts • La plupart des composants destinés à fonctionner avec Arduino fonctionnent bien en 3.3 Volts ou 5 Volts. • GND (8) (Ground / Masse) – Il y a plusieurs broches de ce type présentes sur la carte Arduino, elles sont toutes communes et peuvent être utilisées comme masse (potentiel 0 Volts) pour vos circuits. • Vin (9) – Cette broche permet d'alimenter l'Arduino depuis une source de tension extérieure. Elle est reliée au circuit d'alimentation principale de la carte Arduino.
 <p>10</p>	<p>Broches analogiques</p> <p>L'Arduino UNO possède 5 broches d'entrées analogiques numérotée de A0 jusqu'à A5. Ces broches permettent de lire un signal analogique d'un capteur comme un capteur d'humidité ou de température. La carte Arduino utilise un convertisseur analogique/numérique (convertisseur CAN) pour permettre la lecture du signal par le microcontrôleur. Un signal sera converti sur 10 bits. La valeur pourra être lue sur une échelle de 1024 points.</p>

 <p>11</p>	<p>Microcontrôleur principal</p> <p>Chaque carte Arduino possède son propre microcontrôleur. Vous pouvez le considérer comme le cerveau de la carte Arduino. Le microcontrôleur sur l'Arduino est légèrement différent d'une carte à l'autre. Les microcontrôleurs sont généralement de la société ATMEL.</p> <p>Vous devez savoir quel est le microcontrôleur de votre carte avant de charger un nouveau programme depuis l'IDE Arduino. Cette information est disponible directement sur le composant.</p>
 <p>12</p>	<p>Connecteur ICSP</p> <p>Avant tout, le connecteur ICSP (<i>In-Circuit Serial Programming</i>) est une connectique AVR comprenant les broches MOSI, MISO, SCK, RESET, VCC et GND. Il s'agit d'un connecteur de programmation. Ce connecteur permet entre autre de programmer directement le microcontrôleur sur les couches les plus basses (boot loader, code ASM...). C'est aussi un port appelé port SPI (Serial Peripheral Interface), qui permet de dialoguer avec d'autres composants SPI (écrans, capteurs, etc....).</p>
 <p>13</p>	<p>Indicateur LED d'alimentation</p> <p>Ce voyant doit s'allumer lorsque vous branchez votre Arduino sur une d'alimentation pour indiquer que votre carte est correctement alimentée. Si cette lumière ne s'allume pas, il y a un problème avec votre alimentation.</p>
 <p>14</p>	<p>LEDs TX et RX</p> <p>Sur la carte, y'a deux indicateurs : TX (émission) et RX (réception). Ils apparaissent à deux endroits sur la carte Arduino UNO.</p> <p>Les broches numériques 0 et 1, pour indiquer les broches responsables de la communication série. Autrement, les Leds TX et RX. Le voyant TX clignote à une vitesse variable lors de l'envoi des données série. La vitesse de clignotement dépend de la vitesse de transmission utilisée par la carte. RX clignote pendant le processus de réception.</p>



 15	<p>Entrées/Sorties numériques</p> <p>La carte Arduino UNO possède 14 broches d'Entrées / Sorties numériques, dont 6 peuvent fournir une sortie PWM (Pulse Width Modulation). Ces broches peuvent être configurées pour fonctionner comme des broches numériques d'entrée pour lire des valeurs logiques (0 ou 1) ou numériques. Elles peuvent également être utilisées comme des broches de sortie pour piloter différents modules comme des Led's, des relais, etc. Les broches étiquetées "~" peuvent être utilisées pour générer des PWM.</p>
 16	<p>Broche AREF</p> <p>AREF est l'acronyme anglais de "référence analogique". Cette broche est parfois utilisée pour définir une tension de référence externe (entre 0 et 5 Volts) comme limite supérieure pour les broches d'entrée analogiques.</p>

Tableau II-2: Les différentes broches de la carte ARDUINO UNO

II.5. Alimentation de la carte Arduino :

La carte Arduino peut-être alimentée via un port USB qui l'alimente par +5 [V]. La norme USB limite à 500 [mA] maximum l'alimentation de la carte. Cela suffit pour des leds mais est généralement insuffisante pour des moteurs ou des servomoteurs (avec cette alimentation seule, lors de la mise en route d'un moteur, cela peut entraîner une chute de tension).

La carte peut aussi fonctionner avec une alimentation externe de 6 à 20 volts. Cependant, si la carte est alimentée avec moins de 7 [V] la broche de 5 [V] pourrait fournir moins de 5 [V] et la carte risque d'être instable. Si on utilise plus de 12 [V], le régulateur de tension de la carte pourrait surchauffer et endommager la carte. Aussi, la plage idéale recommandée pour alimenter la carte Arduino est entre 7 [V] et 12 [V].

La carte peut être alimentée aussi par les broches Vin(+) GND(-) qui peuvent aussi supporter une tension comprise entre 7 et 12 volt (tension convertie aussi en 5 [V] par l'Arduino) [23]-[13].

II.6. Les Caractéristiques de la carte Arduino UNO:

1. Micro contrôleur : ATmega328 ;
2. Tension d'alimentation interne = 5[V] tension d'alimentation (recommandée) = 7 à 12[V], limites = 6 à 20 [V] ;
3. Entrées/sorties numériques : 14 dont 6 sorties PWM ;
4. Entrées analogiques = 6 ;
5. Courant max par broches E/S = 40 [mA] ;
6. Courant max sur sortie 3,3V = 50 [mA] ;
7. Mémoire Flash 32 KB dont 0.5 KB utilisée par le bootloader ;
8. Mémoire SRAM 2 KB ;
9. Mémoire EEPROM 1 KB ;
10. Fréquence horloge = 16 MHz ;
11. Dimensions = 68.6mm x 53.3mm [19].

II.7. L'environnement de la programmation (logiciel IDE) :

II.7.1 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:

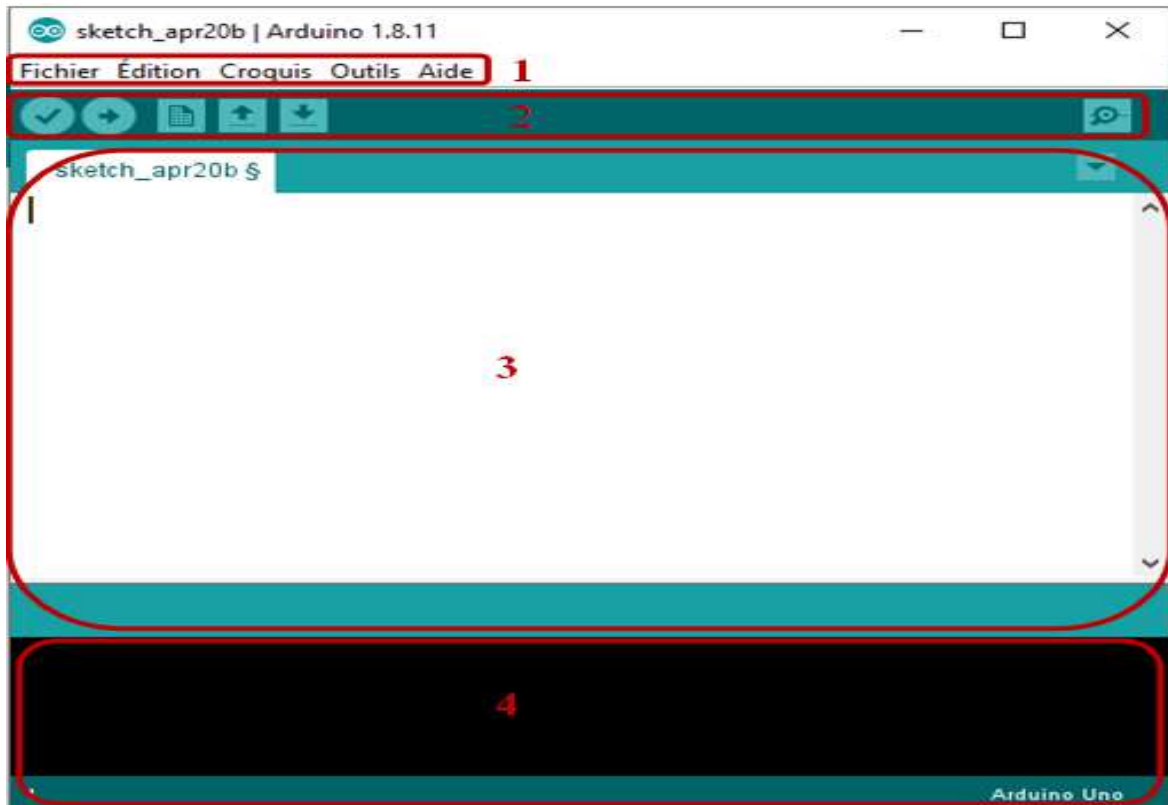


Figure II-2 : la fenêtre principale du logiciel Arduino.

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 [20].

2. Barre de Bouton :



Figure II-3 : Barre de Bouton.

- **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 présent.
- **Sauvegarder :** Sauvegarder le programme.
- **Moniteur série :** Ce bouton permet d'accéder au port série.

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 [20].

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 [20].

II.7.2 Structure du programme :

Le programme comporte de trois phases consécutives :

a) La Définition des constantes et des variables :

Cette partie est optionnelle, chaque entrée et sortie est définie et déclarée, en lui donnant un nom arbitraire et en lui affectant le numéro de l'entrée ou celui de la sortie voulue, sans oublier de préciser le type de la variable.

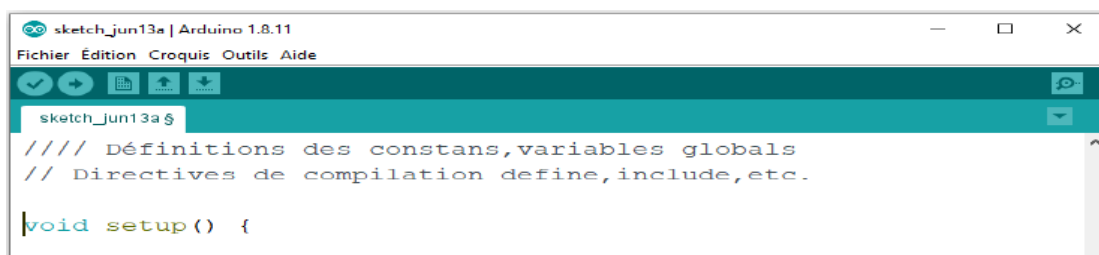
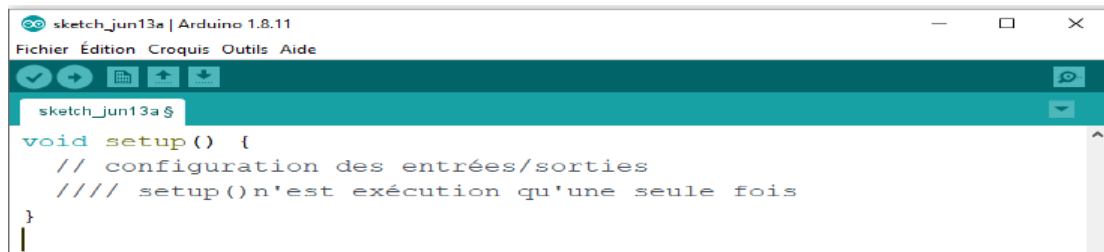


Figure II-4 : Définition des constantes et des variables.

b) Configuration des entrées/sorties :

Les instructions viennent après le **void setup ()**, après avoir ouvert une accolade, on peut manipuler les broches de la carte en les configurant comme étant des entrées ou des sorties, selon les besoins. Les entrées analogiques pour les capteurs par exemple, ne sont soumises à aucune configuration, car la carte possède 6 entrées analogiques qui ne font que cela.


The image shows a screenshot of the Arduino IDE interface. The title bar reads "sketch_jun13a | Arduino 1.8.11". The menu bar includes "Fichier", "Édition", "Croquis", "Outils", and "Aide". The toolbar contains icons for saving, undo, redo, and other IDE functions. The main text area shows the following code:

```
sketch_jun13a $  
void setup() {  
  // configuration des entrées/sorties  
  //// setup() n'est exécution qu'une seule fois  
}
```

Figure II-5: Configuration des Entrées/Sorties.

c) Programmation des interactions et comportements :

Celles-ci viennent après le **void loop ()**, c'est la partie principalement, où on rédige les instructions et les opérations comme la lecture des données, les boucles, les affectations,...etc. Chacune d'elle doit obligatoirement finir par un point-virgule.

The image shows a screenshot of the Arduino IDE interface. The title bar reads "sketch_jun13a | Arduino 1.8.11". The menu bar includes "Fichier", "Édition", "Croquis", "Outils", and "Aide". The toolbar contains icons for saving, undo, redo, and other IDE functions. The main text area shows the following code:

```
sketch_jun13a $  
void loop() {  
  // les instructions contenues ici sont exécutées indéfiniment en boucle  
  // seule une coupure de l'alimentation de la carte ou un appui sur le bouton reset  
  // permet de quitter le programme  
}
```

Figure II-6 : Programmation des interactions et comportements.

d) Les commentaires :

Comme chaque IDE, des commentaires peuvent être ajoutés au programme. Dans la configuration des entrées/sorties, les commentaires doivent être écrits après un slash ou une étoile ou les deux, tandis que sur une ligne de code, on les écrit après deux slash [20].

II.7.3 Syntaxe du langage Arduino :

La syntaxe des commandes d'Arduino est donnée dans la table des matières citée en ci-dessous. Chaque instruction est suivie de sa traduction entre-parenthèses [21]-[22].

Commandes de structure du programme	
Structure générale	<ul style="list-style-type: none"> • Void setup () (configuration, préparation) • Void loup () (exécution)
Contrôle et conditions	<ul style="list-style-type: none"> • if (si...) • if...else (si...sinon...) • for (pour...) • Switch case (dans le cas où...) • while (pendant que...)
Opérations de comparaison	<ul style="list-style-type: none"> • == (équivalent à) • != (différent de) • < (inférieur à) • > (supérieur à) • <= (inférieur ou égal à) • >= (supérieur ou égal à)
Opérations booléennes	<ul style="list-style-type: none"> • && (et) • (ou)
Autres commandes	<ul style="list-style-type: none"> • // (commentaires simple ligne) • /**/ (commentaires multi-lignes)

Tableau II-3: Les commandes du programme Arduino.

Fonctions	
Entrées-sorties numériques	<ul style="list-style-type: none"> • <code>pinMode (broche, état)</code> (configuration des broches). • <code>digitalWrite (broche, état)</code> (écrire un état sur une broche numérique). • <code>digitalRead (broche)</code> (lire un état sur une broche numérique).
Entrées analogiques	<ul style="list-style-type: none"> • <code>analogRead (broche)</code> (lire la valeur d'une broche analogique). • <code>analogWrite (broche, valeur)</code>(PWM : écrire une valeur analogique sur les broches 9, 10 ou 11).
Gestion du temps	<ul style="list-style-type: none"> • <code>millis()</code> (temps de fonctionnement du programme) • <code>delay (ms)</code> (attente, en millisecondes) • <code>delayMicroseconds (us)</code> (attente, en μs).

Tableau II-4: Les fonctions du programme Arduino.

Variables	
Variables	<ul style="list-style-type: none"> • <code>char</code> (variable 'caractère') • <code>int</code> (variable 'nombre entier') • <code>long</code> (variable 'nombre entier de très grande taille') • <code>string</code> (variable 'message')
Niveaux logiques des connecteurs numériques	<ul style="list-style-type: none"> • <code>HIGH</code> (état 1) • <code>LOW</code> (état 0)
Configuration	<ul style="list-style-type: none"> • <code>INPUT</code> (configuré en entrée) • <code>OUTPUT</code> (configuré en sortie)

Tableau II-5: Les différentes variables du programme Arduino.

II.7.4 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.

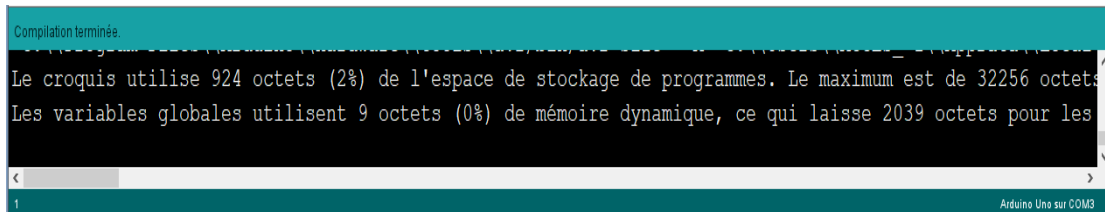


Figure II-7 : Compilation.

Une fois la compilation terminée, le programme sera sauvegardé. Mais avant de téléverser, il faut:

- ✓ La branchée sur l'ordinateur. Une fois branchée, on clique sur le bouton « **Outils** » ;
- ✓ On sélectionne « **Type de carte** », puis on coche la carte correspondante;
- ✓ Toujours dans le menu « **Outils** », on clique sur « **Serial port** », pour sélectionner le type de la connexion;
- ✓ On télécharge le programme en cliquant sur « **Téléverser** » plus haut sur la barre d'action, le message « **Compilation du croquis** » s'affiche au-dessus de la barre des erreurs, ce message signifie que le programme est en train d'être traduit en langage machine. Puis, on verra « **Téléversement...** », Qui veut dire que le programme se charge dans la carte [20].

Remarque :

Pour connaître le type de port, on doit se rendre dans le « **Panneau de configuration** », on sélectionne « **Gestionnaire des périphérique** », puis on cherche la ligne « **Port (COM et LPT)** », et là on peut voir le nom de l'Arduino suivi du type de la connexion entre parenthèses, par exemple « Arduino Uno → (COM3) ».

II.8. Conclusion :

Ce chapitre explique les deux parties essentielles d'Arduino :

La partie matérielle (Hardware) : on a cité la description particulière d'Arduino Uno et leur caractéristiques fondamentales tels que les types et valeur d'alimentations, types des pins et ses fonctionnement partuculieres tels que Rx,Tx et PWM.

La partie logicielle (Software) : on a décrit en détaille le logiciel IDE par son interface, son environnement de programmation, les bases des instructions utilisés dans l'algorithme de commande...

Dans le chapitre suivant, on essayer d'exploiter ces savoirs faire pour développer une carte de commande qui va gérer les séquences d'ascenseur.

Chapitre III :
Conception et réalisation

III.1 Introduction :

Ce chapitre présente la réalisation pratique et la programmation de la carte de commande qui gère les séquences d'un ascenseur.

La première phase, nous allons donner la 1^{ère} partie l'étude qui concerne la description des composants utilisés, suivie par une partie pratique axée sur la réalisation de circuit et la conception de la carte par logiciel, ensuite on la mise sous l'exploitation.

III.2 Partie d'étude :

Les composants utilisés pour notre réalisation:

III.2.1 Le moteur à courant continue :

III.2.1.1 Description :

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 [14].



Figure III-1 : Moteur à courant continu [27].

III.2.1.2 Principe de fonctionnement :

- Le stator :

Le stator d'un moteur à courant continu est la partie fixe du moteur (statique). Le stator est aussi nommé l'inducteur ou l'excitation : on fait passer un courant dans le bobinage du stator et c'est lui qui crée (qui induit) un champ magnétique. Le stator pose le décor pour le rotor qui se retrouve ainsi plongé dans ce champ magnétique. Le stator (inducteur) crée un champ magnétique B appelé champ inducteur [14].

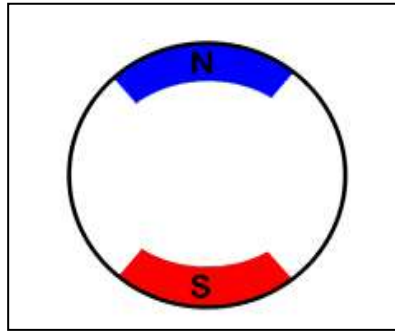


Figure III-2: Stator d'un MCC [23].

- Le rotor :

Le rotor est la partie rotative de moteur. Il est constitué du bobinage induit. Il faut alimenter cette bobine pour la transformer en électroaimant qui entrera en interaction avec le stator. Si on n'alimente pas le rotor, il ne serait l'objet d'aucune force et ne tourne pas.

Un système de frottement spécial permet d'alimenter le rotor : des balais ou charbons montés sur des ressorts frottent sur les contacts en rotation : le collecteur [14].

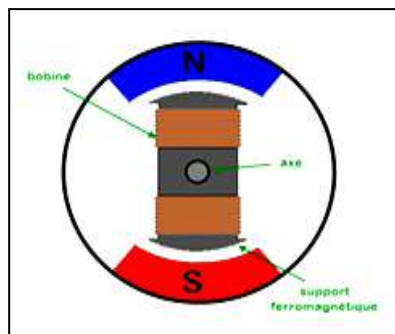


Figure III-3: Rotor d'un MCC [23].

- Le collecteur :

Le collecteur est un ensemble de plages métalliques qui font contact avec les charbons. Il appartient au rotor. Le frottement des charbons fait qu'ils s'usent : ils sont montés sur ressort pour garantir un contact même lorsqu'ils raccourcissent à cause de l'usure [14].

Comme on le sait le moteur à courant continu a un seul sens de rotation, pour avoir deux sens de rotation nous avons utilisé un composant très répandu dans le monde de l'électronique de puissance appelé « **pont en H** ».

III.2.1.3 Le pont en H :

Le pont en H est une structure électronique servant à contrôler la polarité aux bornes d'un dipôle, il est composé de quatre éléments de commutation généralement disposés schématiquement en une forme de H. Cette structure se trouve dans plusieurs applications de l'électronique de puissance incluant le contrôle des moteurs. Le pont en H permet de choisir le sens du courant en fonction de l'état des 4 interrupteurs (ils peuvent être remplacés par des transistors) qui le composent [23].

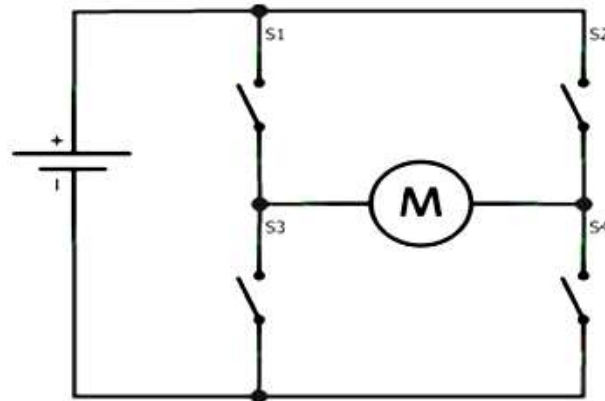


Figure III-4: Le pont en H [23].

- 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.

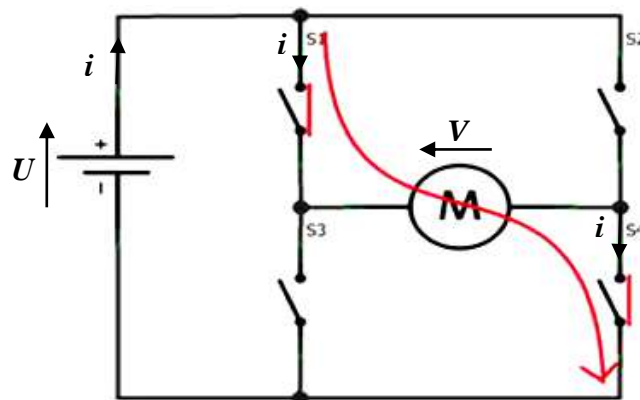


Figure III-5: Fonctionnement dans le sens horaire [23].

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

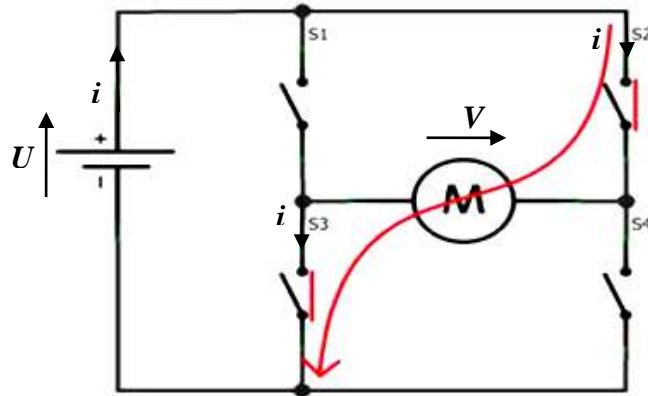


Figure III-6: Fonctionnement dans le sens antihoraire [23].

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.

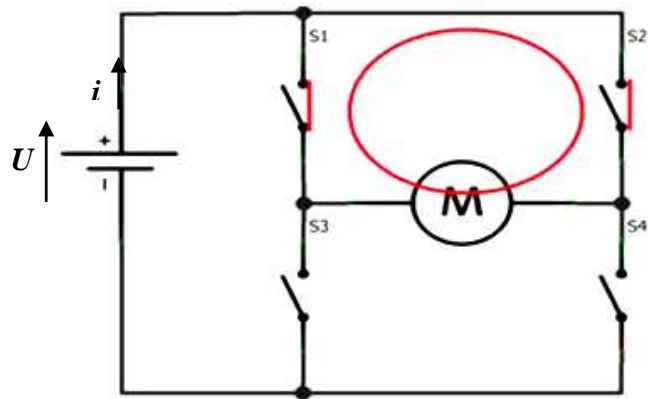


Figure III-7: Freinage avec $S1$ et $S2$ [23].

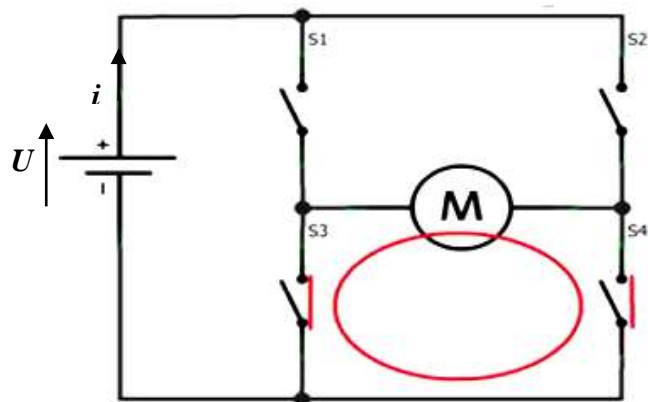


Figure III-8: Freinage avec $S3$ et $S4$ [23].

Si on ferme S1 et S3 et/ou S2 et S4 ensemble, cela ferait un court-circuit de l'alimentation et les transistors risqueraient de griller immédiatement si l'alimentation est capable de fournir un courant plus fort que ce qu'ils ne peuvent admettre.

- **Protections du pont en H par des diodes de roue libre :**

Pour protéger un transistor des parasites ou lors du freinage électronique du moteur, nous plaçons une diode qui devra être en parallèle aux bornes du transistor, le courant trouvera toujours un moyen de passer sans risquer de forcer le passage dans les transistors en les grillant.

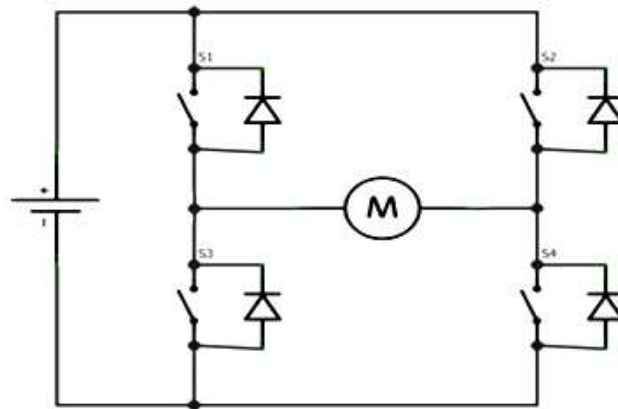


Figure III-9: Pont en H avec ses diodes de protection [23].

Afin d'éviter les branchements des transistors et leur logique de contrôle, des composants "clés en main" ont été développées et produits. Nous allons maintenant présenter un d'entre eux que nous retrouvons dans quasiment tous les shields moteurs Arduino qui est le **L293D** [23].

III.2.1.4 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 [15]-[16].

• **Brochage :**

Ci-dessous la configuration des broches du L293D.

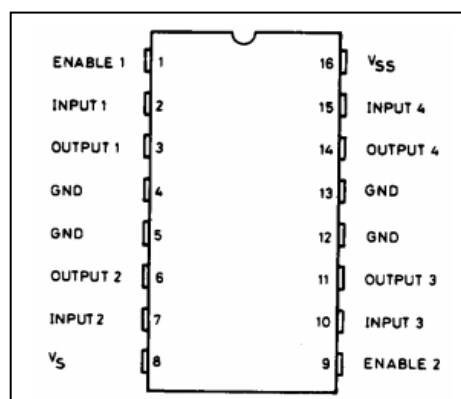


Figure III-10: Configuration des broches du L293D [16].

- **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.
- **VSS:** Alimentation de la logique de commande (5V). A raccorder à la borne +5V de votre Arduino [16].

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

III.2.2 Détecteur de fin de course :

Un interrupteur de fin de course est un dispositif mécanique qui nécessite le contact physique d'un objet avec l'actionneur de l'interrupteur pour faire changer l'état du contact (ouvert/fermé) [18].

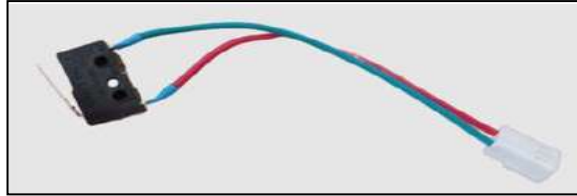


Figure III -11: Interrupteur fin de course.

III.2.2.1 Utilisation :

1. La position d'un vérin (début et fin) ;
2. La position d'une porte (ouverte ou fermée) ;
3. La position d'un ascenseur (étage) ;
4. La position d'une barrière automatique (ouverte ou fermée) ;
5. La position d'une manette de jeu (bouton appuyé, direction souhaitée,...) [25].

III.2.2.2 Principe de fonctionnement :

C'est un commutateur, commandé par le déplacement d'un organe de commande (corps d'épreuve).

Lorsque le corps d'épreuve est actionné, il ouvre ou ferme un contact électrique [26].

III.2.3 Bouton poussoir :

III.2.3.1 Description :

Les boutons poussoirs sont des composants très simples. Lorsque on presse le bouton, un contact électrique fait et laisse passer le courant, Les boutons poussoirs ont 4 pattes ce qui peut créer une certaine confusion [14].



Figure III -12: Bouton poussoir.

III.2.3.2 Fonctionnement :

Lorsqu'il est ouvert, la tension à ses bornes ne peut être nulle (ou alors c'est que le circuit n'est pas alimenté). En revanche, lorsqu'il est fermé cette même tension doit être nulle. En effet, aux bornes d'un fil la tension est de 0V.

Ensuite, lorsque le bouton est ouvert, aucun courant ne peut passer, le circuit est donc déconnecté. Par contre, lorsqu'il est fermé, le courant nécessaire au bon fonctionnement des différents composants le traverse [23].

III.2.3.3 Bouton poussoir normalement ouvert (NO) :

Ils ont deux positions :

- lorsqu'on relâche le bouton-poussoir, le courant ne passe pas, on dit que le circuit est ouvert.
- Lorsqu'on appuie sur le bouton-poussoir, le courant passe, on dit que le circuit est fermé.

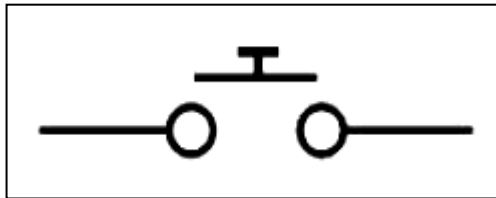


Figure III -13: Schéma d'un BP (NO) [23].

III.2.3.4 Bouton poussoir normalement fermé (NF) :

Ce type de bouton est l'opposé du type précédent :

- lorsqu'on relâche le bouton-poussoir, le courant passe, on dit que le circuit est fermé.
- Lorsqu'on appuie sur le bouton-poussoir, le courant ne passe pas, on dit que le circuit est ouvert.

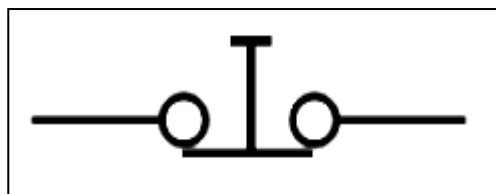


Figure III -14: Schéma d'un BP (NF) [23].

III.2.4 Afficheur 7 segments :

III.2.4.1 Définition :

Comme son nom l'indique, l'afficheur 7 segments possède 7 segments. C'est une portion de l'afficheur, qui est allumée ou éteinte pour réaliser l'affichage. Cette portion n'est en fait rien d'autre qu'une LED qui au lieu d'être ronde comme d'habitude est plate et encastre dans un boîtier. On dénombre donc 8 portions en comptant le point de l'afficheur (mais il ne compte pas en tant que segment à part entière car il n'est pas toujours présent) [23].



Figure III -15: Afficheur 7 segments.

III.2.4.2 Des LED :

Voici un schéma qui présentant un modèle d'afficheur sans le point (Modèle de 07 leds).

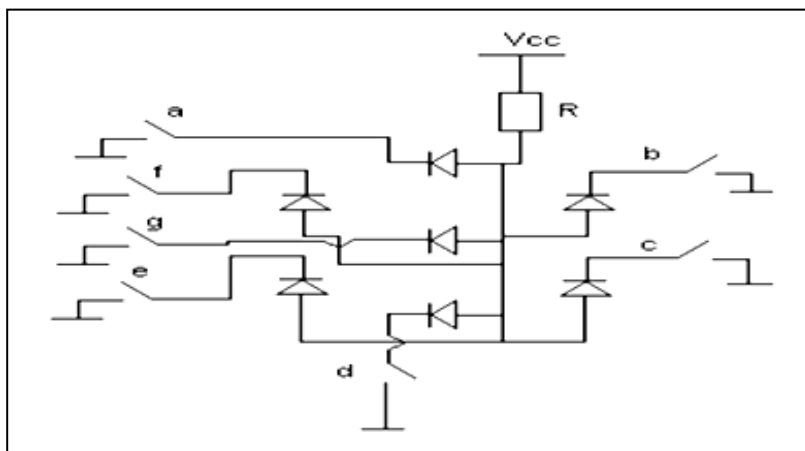


Figure III-16: Schéma électrique interne d'un afficheur 7 segments anode commune [23].

Les interrupteurs a, b, c, d, e, f, g représentent les signaux pilotant chaque segments. Toutes les LED possèdent une broche commune, reliée entre elle. Selon que cette broche est la cathode ou l'anode qui représente l'afficheur à cathode commune ou anode commune. Dans l'absolu, ils fonctionnent de la même façon, seule la manière de les brancher diffère (actif sur état bas ou sur état haut) [23].

III.2.4.3 Cathode commune ou Anode commune :

Dans le cas d'un afficheur à cathode commune, toutes les cathodes sont reliées entre elles en un seul point lui-même connecté à la masse. Ensuite, chaque anode de chaque segment sera reliée à une broche de signal. Pour allumer chaque segment, le signal devra être une tension positive. En effet, si le signal est à 0, il n'y a pas de différence de potentiel entre les deux broches de la LED et donc elle ne s'allumera pas ! Si nous sommes dans le cas d'une anode commune, les anodes de toutes les LED sont reliées entre elles en un seul point qui sera connecté à l'alimentation. Les cathodes elles seront reliées une par une aux broches de signal [23].

Pour notre projet nous allons fait le choix d'utiliser des afficheurs a cathode commune.

III.2.4.4 Branchement de l'afficheur a cathode commune :

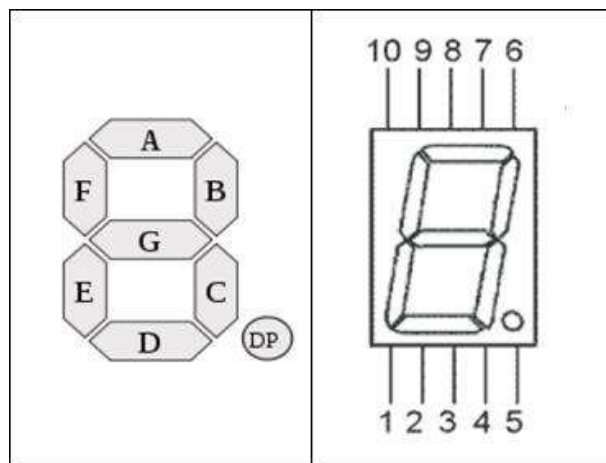


Figure III-17: Représentation des Afficheur 7 segments identifier avec des lettres et des numéros [23].

La signification des différentes broches sont comme suit :

1. LED de l'anode E.
2. LED de l'anode D.
3. Cathode commune des LED.
4. LED de l'anode C.
5. (facultatif) le point décimal.
6. LED de l'anode B.
7. LED de l'anode A.
8. Cathode commune des LED.
9. LED de l'anode F.
10. LED de l'anode G [23].

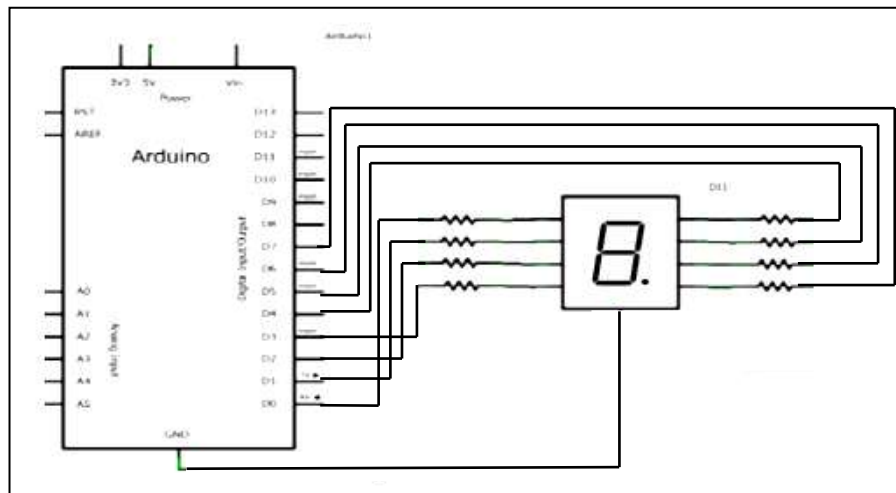


Figure III-18: Exemple du branchement d'un afficheur à cathode commun.

III.2.4.5 Problématique liée au l'afficheur 7 segment avec l'Arduino Uno :

Si on utilise l'afficheur 7 segment dans notre projet, notamment qu'on a utilisé le microcontrôleur arduino Uno qui possède 14 broches entrées/sorties numériques et l'afficheur 7 segment a 7 broches ou 7 sorties, donc va prendre la moitié des broches de microcontrôleur.

En effet pour résoudre ce problème on a besoin d'une procédure qui minimise le nombre des ports utilisés, cette solution est valable lorsque on utilise le circuit intégré 74LS48, qu'on appelle aussi le décodeur BCD.

III.2.4.6 Décodeur BCD – 74LS48 :

Ce décodeur permet de convertir le code Binaire vers le code BCD, présent à son entrée sous 4 bits, en un code 7 segments disponible à sa sortie. Il est utilisé pour piloter un afficheur 7 segments afin d'avoir des chiffres de 0 à 9, certaines lettres.

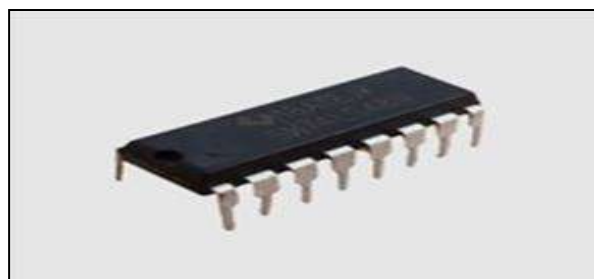


Figure III-19: Décodeur BCD 74ls48.

III.2.4.7 La configuration des pins ou des broches du 74LS48 :

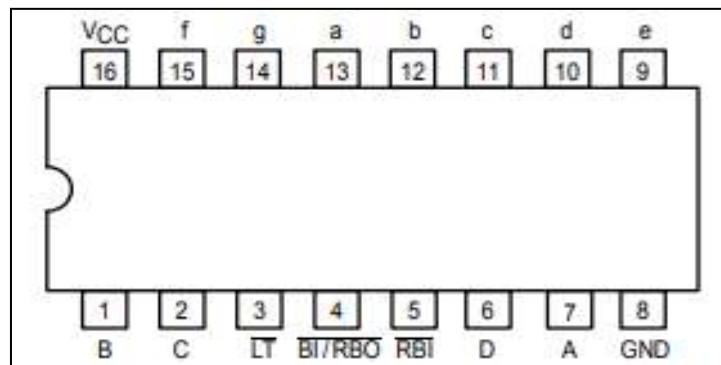


Figure III-20: configuration des broches du décodeur BCD [16].

- Quatre entrées A, B, C, D destinées à recevoir la logique BCD (binaire codé décima).
- Une entrée LT (lamp test) ou «lamp test» qui permet de vérifier le fonctionnement de l'afficheur en allumant tous les segments si BI est à l'état 1.
- L'entrée **BI / RBO** ou «blanking input» qui permet l'effacement des segments de l'afficheur quelque soit l'état des autres entrées.
- L'entrée **RBI** ou «ripple blanking input» qui permet l'effacement des 0 à gauche si A, B, C, D sont à 0.
- Sept sortie a, b, c, d, e, f et g sont prévues pour alimenter en logique positive les segments de l'afficheur.
- GND la masse.
- V_{CC} l'alimentation [16].

III.2.5 Ecran LCD :

III.2.5.1 Description :

L'afficheur LCD est en particulier une interface visuelle entre un système (projet) et l'homme (utilisateur). Son rôle est de transmettre les informations utiles d'un système à un utilisateur. Il affichera donc des données susceptibles d'être exploitées par l'utilisateur d'un système [17].

III.2.5.2 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 [6].

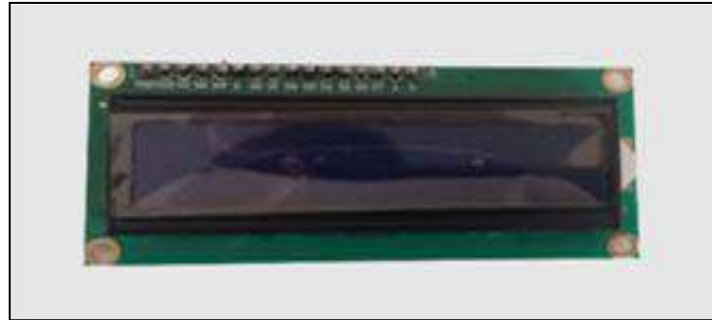


Figure III -21: Afficheur LCD16*2.

III.2.5.3 Brochage d'un afficheur LCD :

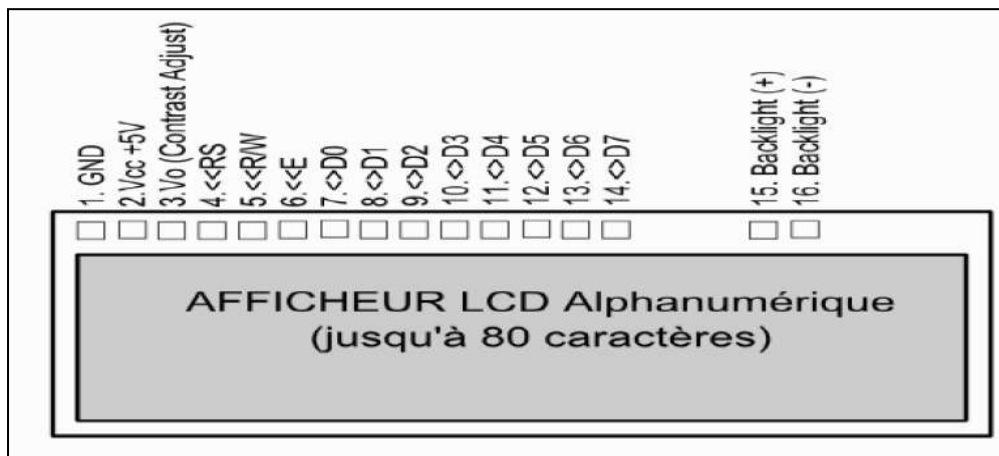


Figure III -22: Schéma d'un afficheur LCD [6].

NO	Nom	Role de broche
1	Masse	Masse
2	Vcc	Alimentation positive +5V de l'afficheur
3	Vo	Réglage du contraste de l'afficheur entre 0 et +5V (plus proche de la masse)
4	RS	Commutation de registre entre les instructions « 0 » et les données « 1 »
5	R/W	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)
8	D1	
9	D2	
10	D3	
11	D4	
12	D5	
13	D6	
14	D7	
15	A	Anode rétro-éclairage (+5V)
16	K	Cathode rétro-éclairage (masse)

Tableau III-1: Brochage d'un afficheur LCD.

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 entrées /sorties à notre microcontrôleur, Nous avons utilisé la liaison I2C, Cette liaison est utilisable avec seulement 2 broches (une broche de donnée et une broche d'horloge) et nécessite l'utilisation de deux broches analogiques de l'Arduino (broche 4 et 5).

III.2.5.4 La liaison I2C :

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)

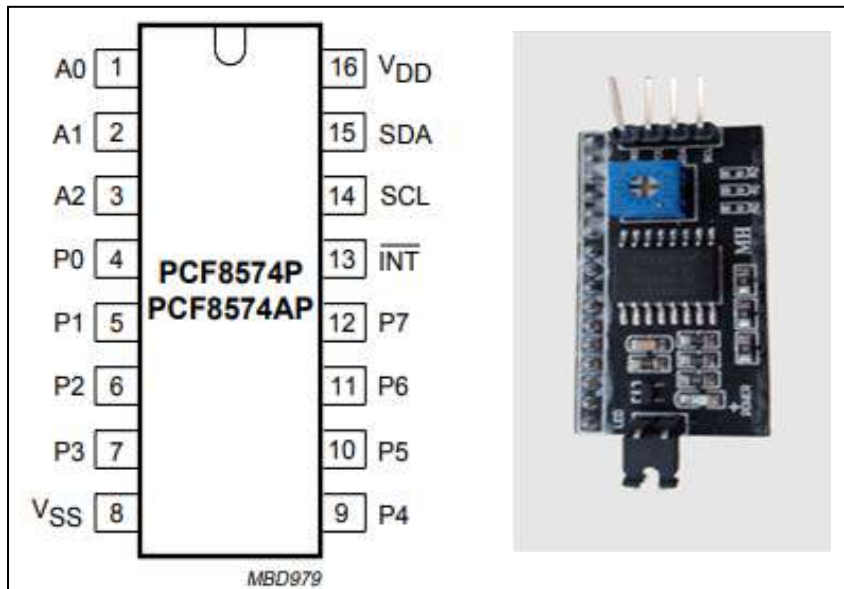


Figure III-23: le circuit PCF8574 [16].

Le tableau ci-dessous représente les pins de circuit PCF8574 [16] :

Symbole	Pin	Description
A0	1	adresse d'entrée 0
A1	2	adresse d'entrée 1
A2	3	adresse d'entrée 2
P0	4	E/S quasi-bidirectionnelles 0
P1	5	E/S quasi-bidirectionnelles 1
P2	6	E/S quasi-bidirectionnelles 2
P3	7	E/S quasi-bidirectionnelles 3
V _{SS}	8	terre d'alimentation
P4	9	E/S quasi-bidirectionnelles 4
P5	10	E/S quasi-bidirectionnelles 5
P6	11	E/S quasi-bidirectionnelles 6
P7	12	E/S quasi-bidirectionnelles 7
INT	13	sortie d'interruption (active LOW)
SCL	14	ligne d'horloge série
SDA	15	ligne de données série
V _{DD}	16	tension d'alimentation

Tableau III-2 : Les pins de circuit PCF8574.

III.2.6 Capteur de poids :

L'exploitation des ascenseurs est soumise à des règles strictes concernant la sécurité des personnes. La charge de la cabine ne doit pas dépasser une valeur limitée, sinon la cabine sera immobilisée. Pour simuler cette fonction, nous avons utilisé un potentiomètre en guise de capteur de poids [27].



Figure III-24: Potentiomètre.

III.3 Partie pratique :

III.3.1 Partie logiciel :

III.3.1.1 PROTEUS :

PROTEUS est un logiciel qui permet de dessiner des schémas électroniques, simuler et réaliser le circuit imprimé correspondant. Ce logiciel comprend deux principaux modules : ISIS et ARES.

ISIS :

Le logiciel ISIS de Proteus est principalement connu pour éditer des schémas électriques et électroniques. 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.

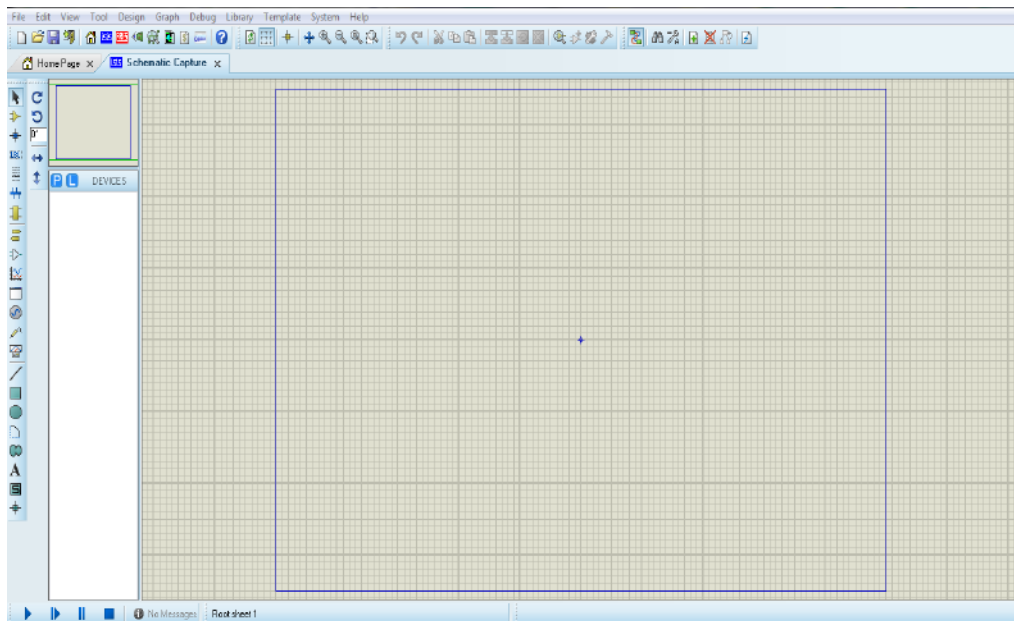


Figure III-25: Fenêtre du module de simulation Proteus ISIS.

ARES :

Le logiciel 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 [15].

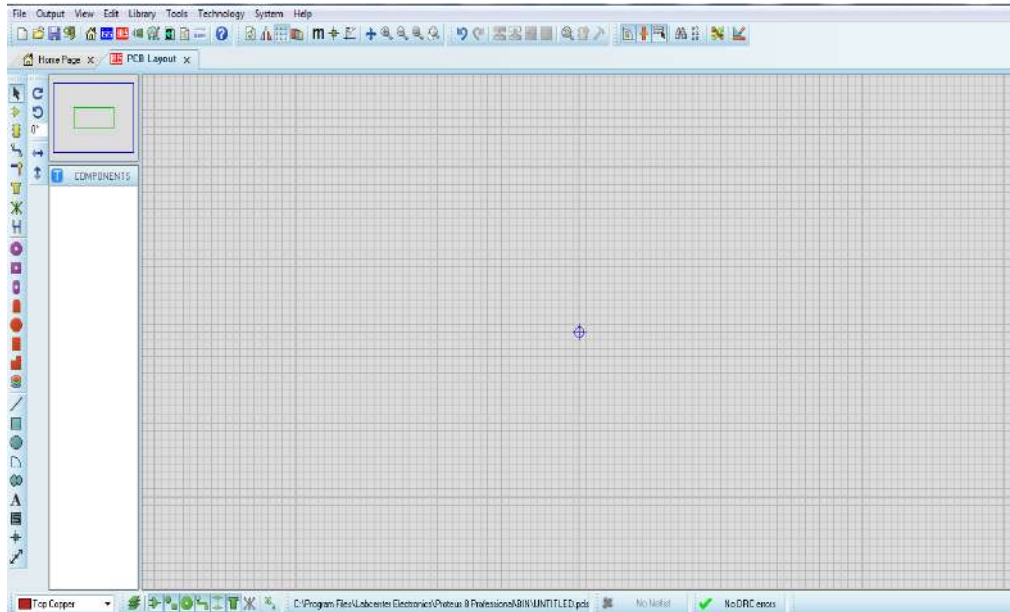


Figure III-26: Fenêtre du module de simulation Proteus ARES.

III.3.1.2 Fritzing :

Fritzing est un logiciel open-source multiplateforme permettant de construire des schémas et des circuits électroniques que nous utilisons dans ce projet. Plusieurs vues sont disponibles : platine d'essai, schémas électriques et circuit imprimé [24].

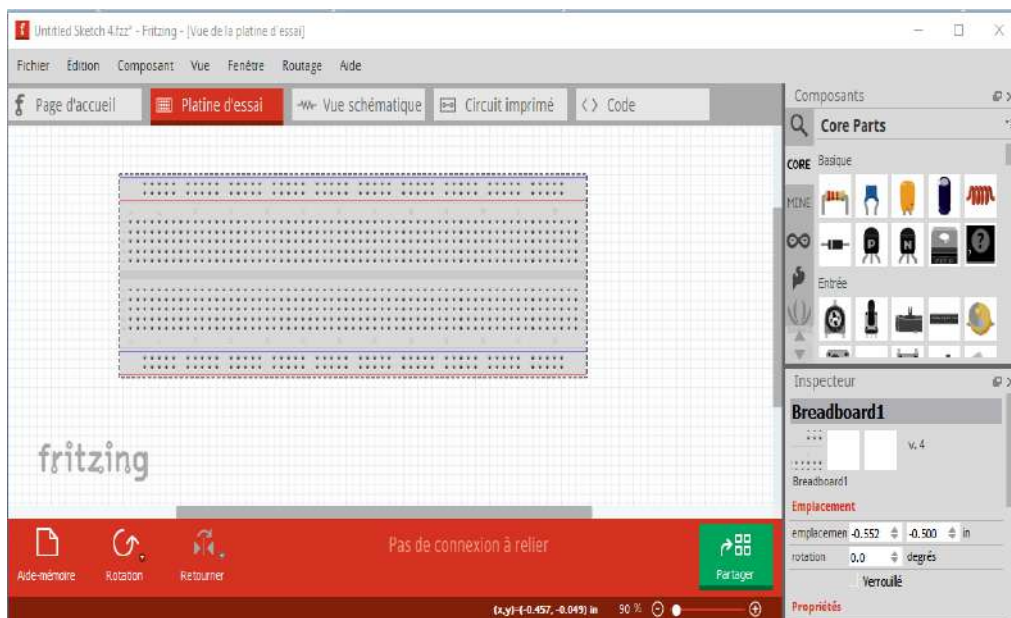


Figure III-27: L'interface de Fritzing.

III.3.1.3 Schéma synoptique :

Pour réaliser notre projet ci-dessus, nous proposons une conception dont le synoptique est donnée par la figure III-28. Les éléments de ce schéma synoptique sont :

- Quatre boutons poussoirs : Chaque étage possède un bouton de commande ;
- Un Sélecteur : Utiliser pour la fonction de l'arrêt d'urgence ;
- Quatre fins de course : Installés dans chaque étage pour savoir la position de la cabine ;
- Un potentiomètre : pour surveiller la surcharge de la cabine ;
- Un moteur à courant continue: Pour le déplacement de la cabine vers la destination demandée ;
- Les afficheurs : un afficheur 7 segment et un afficheur LCD Alph-numérique ;
- Une carte de commande "Arduino Uno R3" : C'est le cerveau du système.

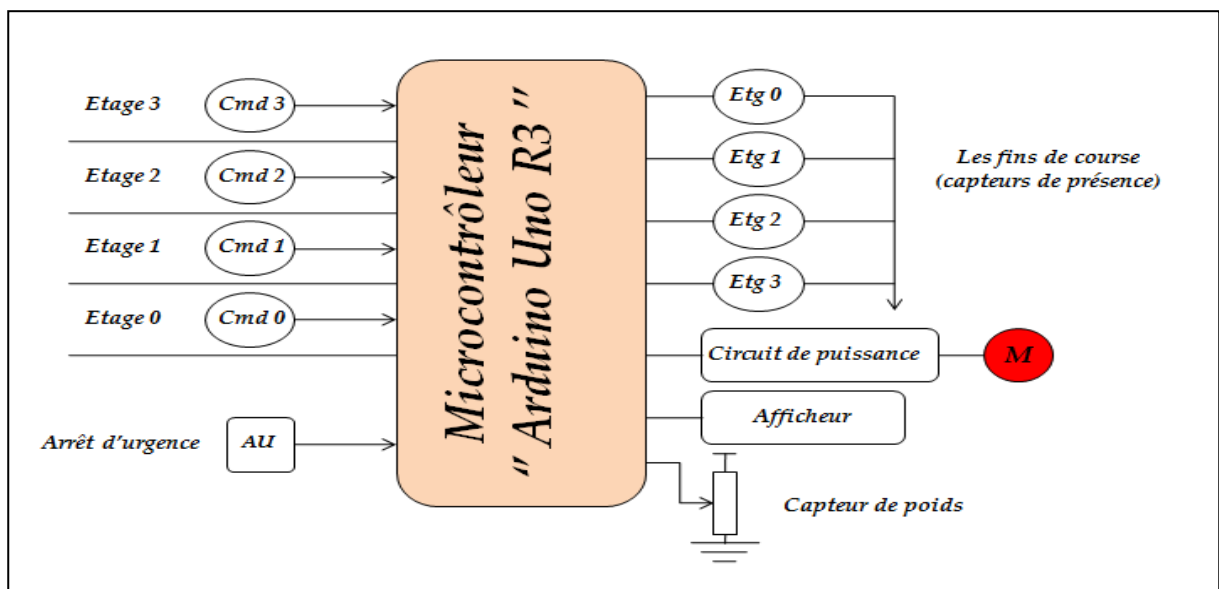


Figure III-28 : Schéma synoptique de la commande d'ascenseur.

➤ Les boutons poussoirs:

Les appels étages peuvent s'effectuer a partir de quatre boutons poussoirs :

- Cmd 0 : Bouton d'appel pour appeler la cabine à l'étage 0.
- Cmd 1 : Bouton d'appel pour appeler la cabine à l'étage 1.
- Cmd 2 : Bouton d'appel pour appeler la cabine à l'étage 2.
- Cmd 3 : Bouton d'appel pour appeler la cabine à l'étage 3.

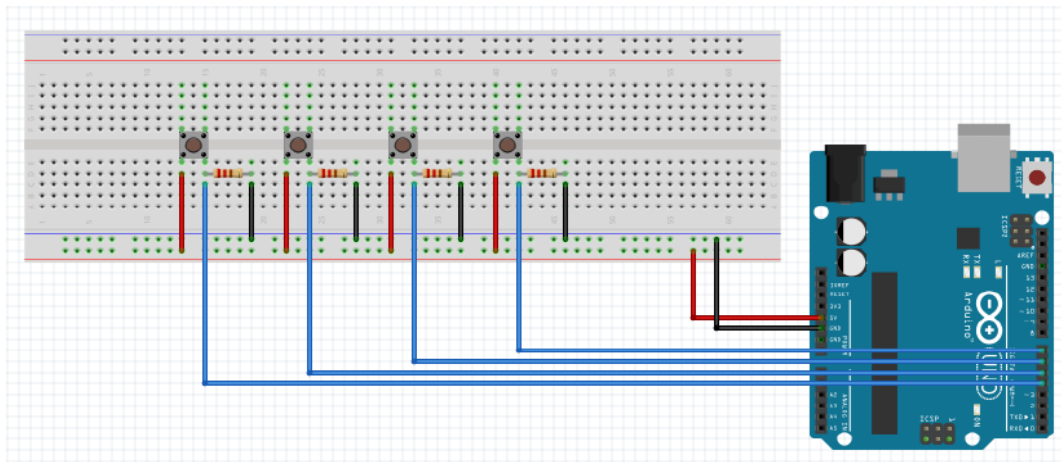


Figure III-29 : Branchement des boutons poussoirs.

➤ **Arrêt d'urgence :**

On utilise un interrupteur qui permet de bloquer l'ascenseur en cas d'anomalie.

➤ **Les fins de courses (capteur de présence):**

Quatre fins de courses sont situées aux quatre étages et détectent la présence de la cabine.

Ces capteurs sont ouverts en l'absence de la cabine et fermés lorsque la cabine est à l'étage.

Cette information est exploitée par la carte Arduino afin de gérer les déplacements de la cabine.

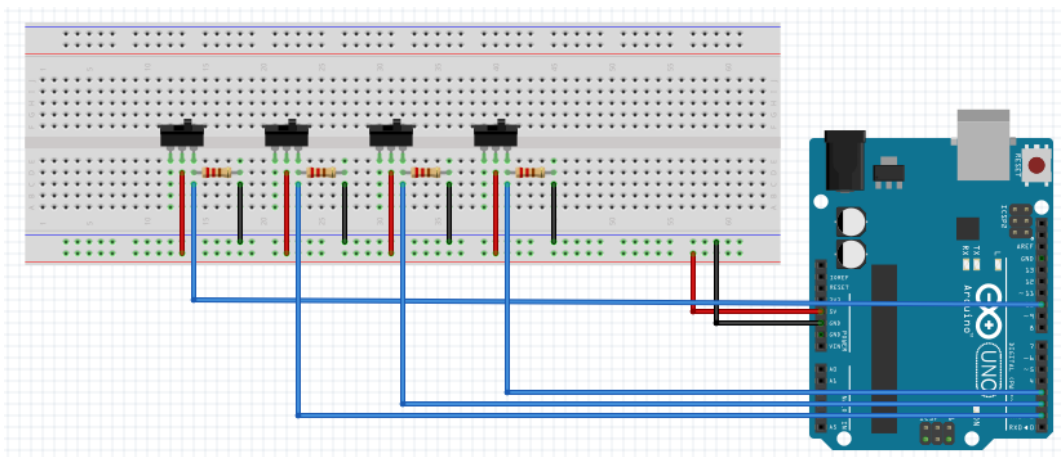


Figure III-30: Branchement des fins de courses.

➤ **Le moteur**

Le moteur qu'on a utilisé dans notre réalisation est le moteur à courant continu pour commander le déplacement vertical de la cabine dans les deux sens.

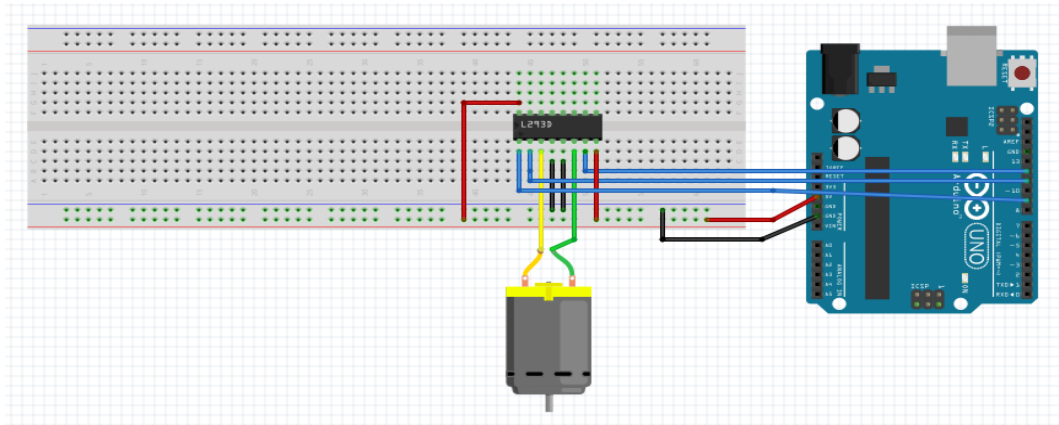


Figure III- 31: Branchement le moteur.

➤ **Les afficheurs:**

Afficheur 7 segments : Il indique la position de l'ascenseur en affichant le numéro de l'étage correspondant.

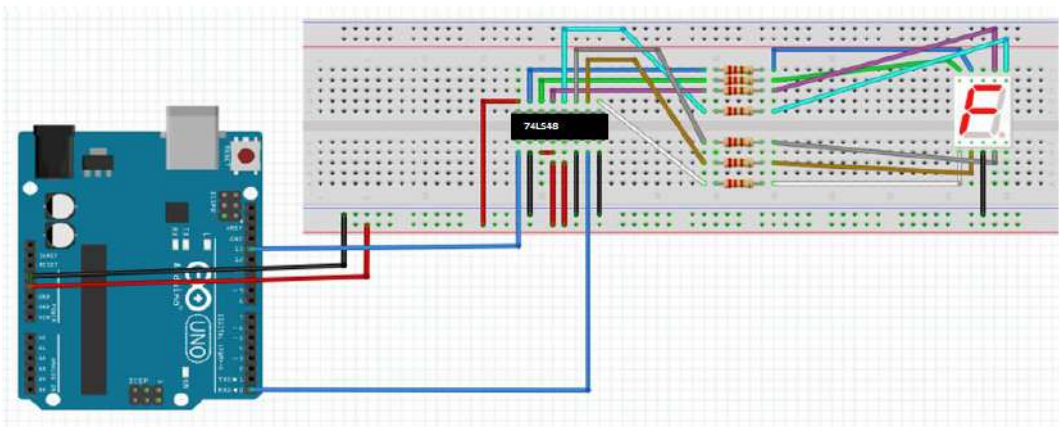


Figure III-32: Branchement de l'afficheur 7 segment.

Ecran LCD : Il donne des informations sur la destination de l'ascenseur.

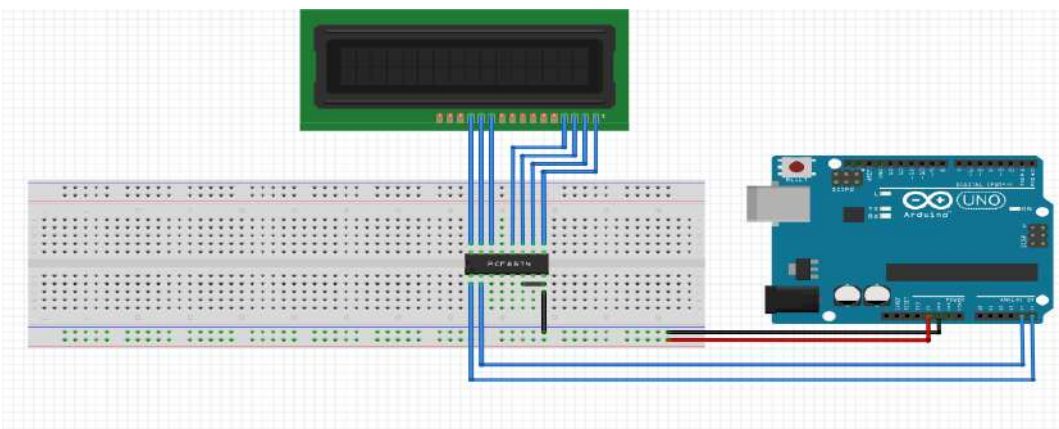


Figure III-33 : Branchement l'afficheur LCD.

➤ Capteur de poids:

Le poids de la charge placée dans la cabine est limité par des directives de sécurité strictes. Si cette charge atteint ce poids la cabine ne démarrera pas et un message est affiché à l'utilisateur.

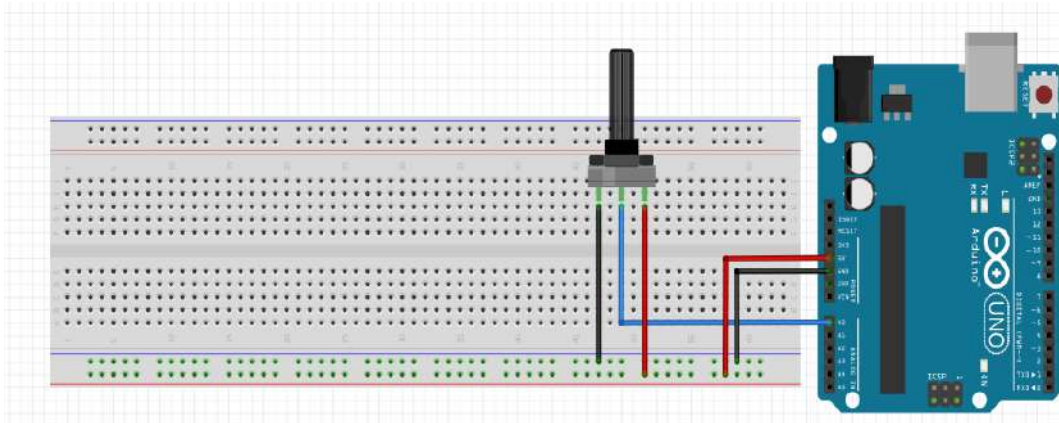


Figure III-34: Branchement de capteur de poids.

III.3.1.4 Schéma de commande globale :

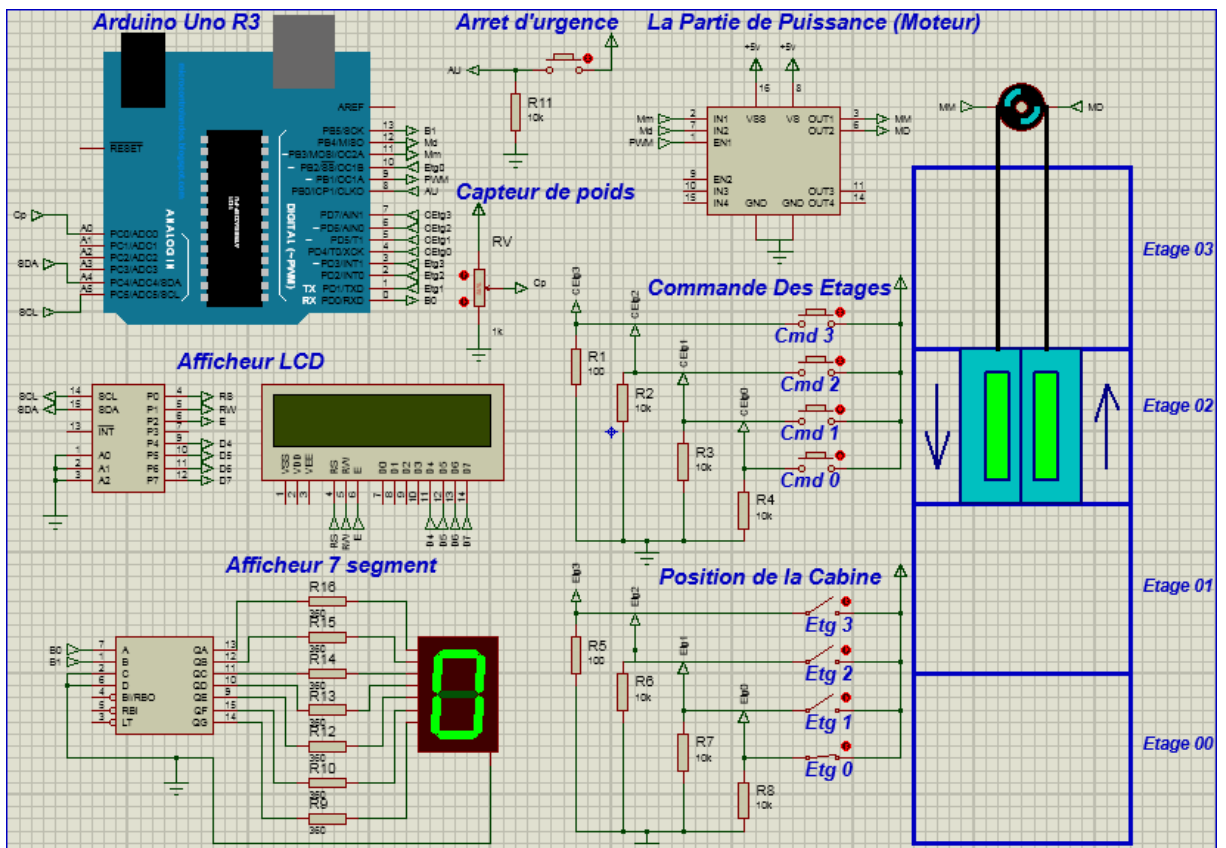
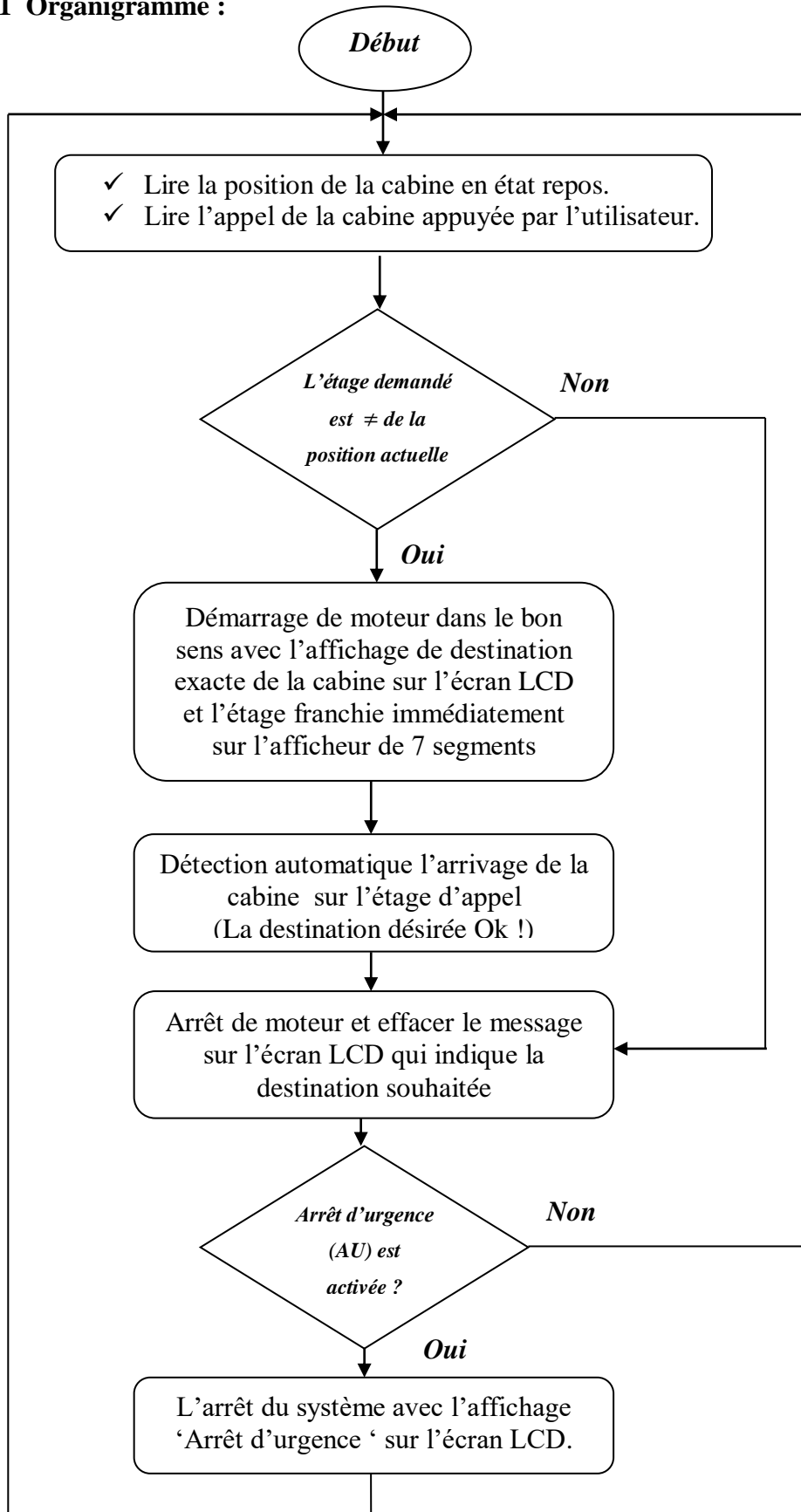


Figure III-35 : Schéma de commande de l'ascenseur.

III.3.2 Partie Programmation :

III.3.2.1 Organigramme :



III.3.2.2 Cahier de charge :

Le système que nous allons étudier représente un ascenseur qui doit pouvoir parcourir 4 niveaux, 3 étages plus le rez de chaussée.

Plusieurs fonctions devront être remplies :

- L'ascenseur doit prendre en compte les appels et les mémoriser pour servir l'étage correspondant.
- La destination de la cabine (monté, descendre) est affichée sur l'écran LCD, et à chaque fois qu'il y a un changement de la position, le numéro de l'étage est indiqué dans l'afficheur 7 segment.
- Si le bouton d'arrêt d'urgence est actionné, le système s'arrête, et un message est affiché sur l'écran LCD.

Si un usager veut accéder à un étage, il lui suffit d'appuyer sur un des boutons d'appel situé à l'extérieur de la cabine. Par exemple une personne veut partir du rez de chaussée vers le 2^{ème} étage. Pour cela il lui suffit d'appuyer sur le bouton d'appel (Cmd 0), si la cabine ne se trouve pas au niveau de rez de chaussée le moteur s'active pour faire descendre la cabine jusqu'au niveau de rez de chaussée. Puis une fois que le client sélectionne ici le bouton de 2^{ème} étage (Cmd 2) le moteur entraîne la cabine vers le haut. Lorsque la cabine aura atteint le 2^{ème} étage le moteur s'arrête et si il n'y a pas d'autre appel la cabine reste au niveau de son dernier parcourt. La commande de notre ascenseur sera gérée par une carte Arduino UNO.

III.3.2.3 Vérification de fonctionnement :

Exemple sur l'arrêt d'urgence :

On a cité précédemment dans le cahier de charge quelques mesures de sécurité, qu'on a utilisées pour notre ascenseur. Voici les fonctions qui leur correspondent dans le programme :

```
{
    if (AU == HIGH)
    {
        lcd.clear ();
        delay (100);
        lcd.setCursor (3, 0);
        lcd.print (" ARRET ");
        lcd.setCursor (3, 1);
        lcd.print ("D'URGENCE ");
        delay (100);
    }
}
```

Lorsque on appuie sur le bouton poussoir de l'arrêt d'urgence (AU), le système s'arrête (le moteur s'arrête), et un message sera afficher sur l'écran LCD.

III.3.3 Partie réalisation :

Après la simulation de notre maquette sur le logiciel ISIS on passe maintenant à la partie de la réalisation.

La figure ci-dessous représente le schéma global de notre travail ou on a réalisé une maquette d'un ascenseur.

Le but de ce branchement est de faire tourner le moteur dans le bon sens ce qui permet de déplacer la cabine vers l'étage souhaité.

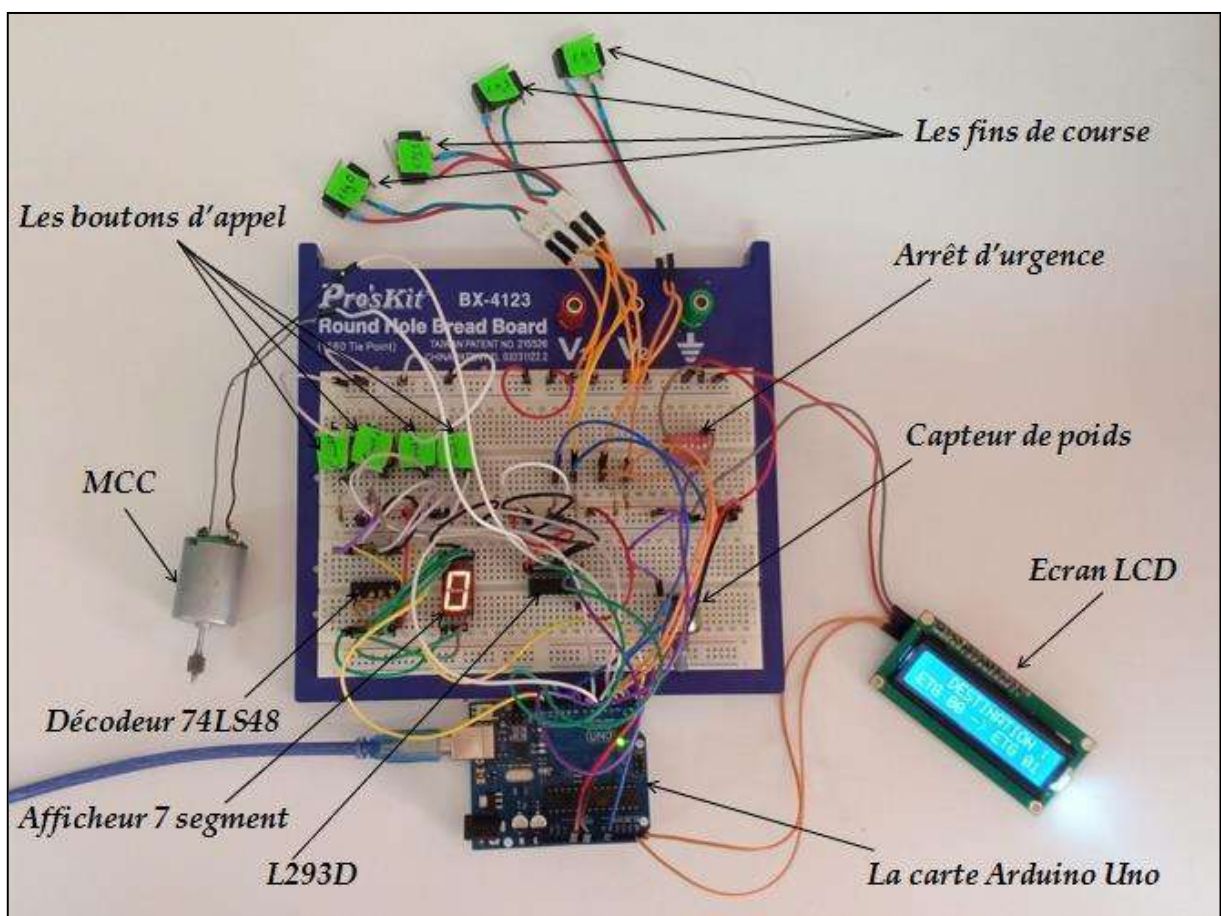


Figure III-36 : Maquette de l'ascenseur.

III.4 Conclusion :

Ce chapitre donne la description des composants utilisés dans la réalisation de notre maquette, nous avons aussi développé un organigramme qui va nous aider à la conception de la partie logiciel.

Conclusion générale

L'objectif initial de ce travail était de créer un prototype de commande d'un ascenseur à quatre niveaux trois étages plus le rez de chaussée (Rch + 03) à base d'une carte Arduino Uno R3.

Nous avons commencé notre rapport par une présentation générale sur les ascenseurs et leurs différents éléments.

Ensuite, une étude sur le microcontrôleur Arduino Uno, a été ainsi faite.

Puis, nous avons élaboré un organigramme de fonctionnement d'un ascenseur selon son principe de fonctionnement par une logique programmée et par une construction réel de sa maquette. La conception des schémas électronique a été faite et simulé sur Proteus avant la mise en œuvre en pratique. Après nous avons validé ce travail par des testes sur la maquette.

Malgré les difficultés que nous avons rencontrées et qui étaient un obstacle pour l'avancement de notre projet, les essais sur la carte réalisée étaient très répondu à les exigences souhaitées données par notre encadrant. , car elle est fonctionne parfaitement et avec une grande précision.

Comme des perspectives de ce projet dans les futurs travaux, on ajoutant quelques options pour améliorer cette maquette.

Pour cela nous proposons:

- Augmentation le nombre des étages avec l'utilisation un autre carte d'Arduino (Méga).
- Intégration un système d'alimentation de secours en cas de coupure soudaine de l'électricité.
- Faire la translation de la cabine avec des vitesses variable (variateur de vitesse).
- Faire des accès limités avec l'utilisation des cartes RFID.
- Faire une commande par l'utilisation d'un système HMI (GUI Matlab).
- Faire l'appel de la cabine par des effets sonores ou bien par une télécommande.
- Intégration la gestion d'ouverture et la fermeture manuelle et automatique de la porte.
- Faire en système de sécurité tel que :
 - Freinage automatique en cas d'une chute libre de la cabine.
 - Détection automatique de la fume (anti incendie) et faire une alarme.

Annexes

Les datasheets des circuits utilisés :

1. Le circuit L293D :



L293D
L293DD

PUSH-PULL FOUR CHANNEL DRIVER WITH DIODES

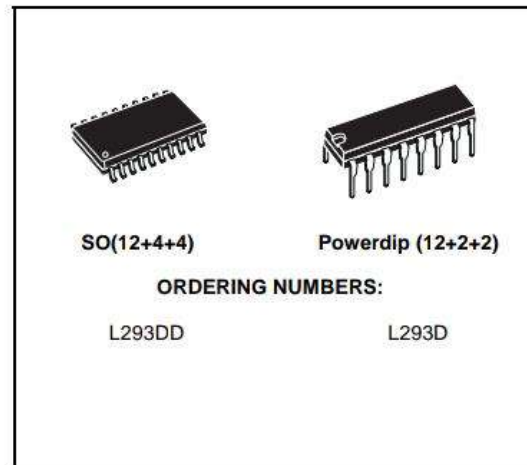
- 600mA OUTPUT CURRENT CAPABILITY PER CHANNEL
- 1.2A PEAK OUTPUT CURRENT (non repetitive) PER CHANNEL
- ENABLE FACILITY
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)
- INTERNAL CLAMP DIODES

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.

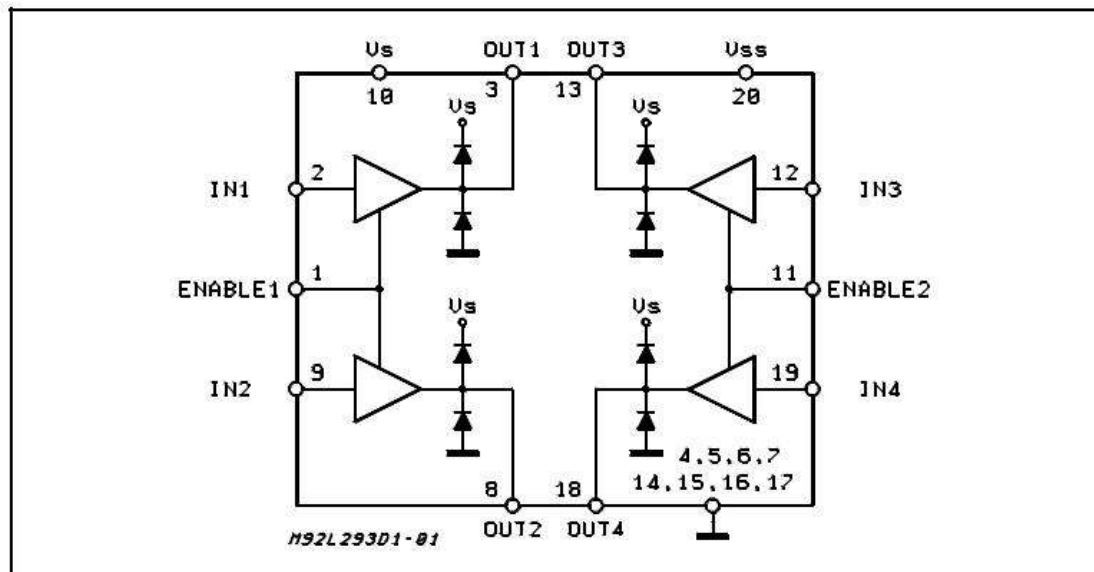
This device is suitable for use in switching applications at frequencies up to 5 kHz.



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 package which has 8 center pins connected together and used for heatsinking.

BLOCK DIAGRAM

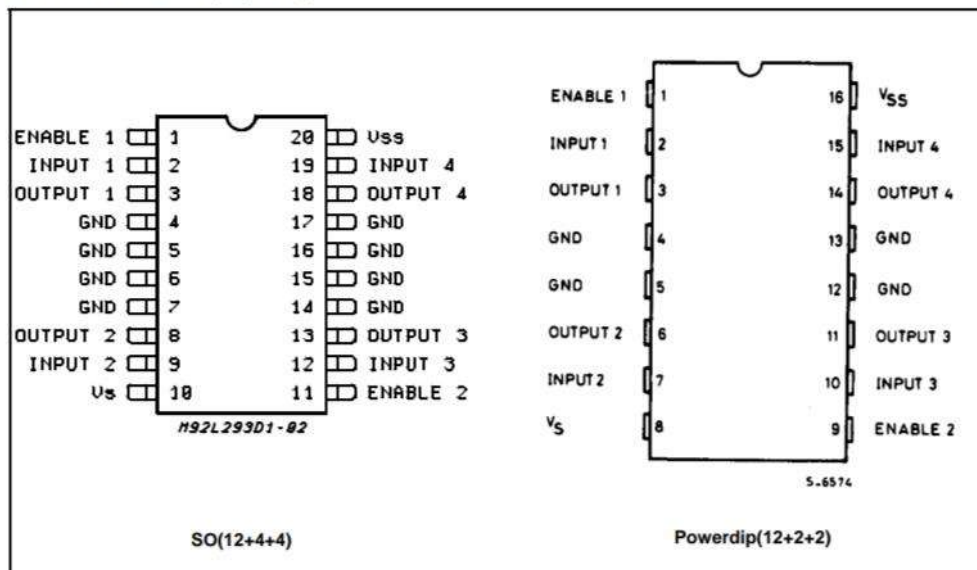


L293D - L293DD

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 $T_{pins} = 90$ °C	4	W
T_{stg}, T_j	Storage and Junction Temperature	- 40 to 150	°C

PIN CONNECTIONS (Top view)



THERMAL DATA

Symbol	Description	DIP	SO	Unit	
$R_{th\ j-pins}$	Thermal Resistance Junction-pins	max.	14	°C/W	
$R_{th\ j-amb}$	Thermal Resistance junction-ambient	max.	80	50 (*)	°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)	$V_i = L$; $I_O = 0$; $V_{\text{en}} = H$		2	6	mA
		$V_i = H$; $I_O = 0$; $V_{\text{en}} = H$		16	24	mA
		$V_{\text{en}} = L$			4	mA
I_{SS}	Total Quiescent Logic Supply Current (pin 20)	$V_i = L$; $I_O = 0$; $V_{\text{en}} = H$		44	60	mA
		$V_i = H$; $I_O = 0$; $V_{\text{en}} = H$		16	22	mA
		$V_{\text{en}} = L$		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
$V_{\text{en}L}$	Enable Low Voltage (pin 1, 11)		-0.3		1.5	V
$V_{\text{en}H}$	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
$I_{\text{en}L}$	Low Voltage Enable Current (pin 1, 11)	$V_{\text{en}L} = 1.5\text{ V}$		-30	-100	μA
$I_{\text{en}H}$	High Voltage Enable Current (pin 1, 11)	$2.3\text{ V} \leq V_{\text{en}H} \leq V_{SS} - 0.6\text{ V}$			± 10	μA
$V_{\text{CE(sat)H}}$	Source Output Saturation Voltage (pins 3, 8, 13, 18)	$I_O = -0.6\text{ A}$		1.4	1.8	V
$V_{\text{CE(sat)L}}$	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 décodeur BCD – 74LS48 :



BCD TO 7-SEGMENT DECODER

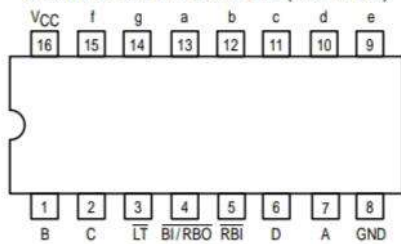
The SN54/74LS48 is a BCD to 7-Segment Decoder consisting of NAND gates, input buffers and seven AND-OR-INVERT gates. Seven NAND gates and one driver are connected in pairs to make BCD data and its complement available to the seven decoding AND-OR-INVERT gates. The remaining NAND gate and three input buffers provide lamp test, blanking input/ripple-blanking input for the LS48.

The circuit accepts 4-bit binary-coded-decimal (BCD) and, depending on the state of the auxiliary inputs, decodes this data to drive other components. The relative positive logic output levels, as well as conditions required at the auxiliary inputs, are shown in the truth tables.

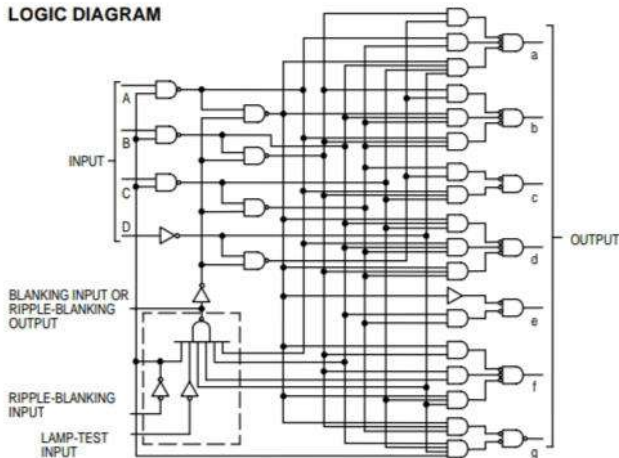
The LS48 circuit incorporates automatic leading and/or trailing edge zero-blanking control (RBI and RBO). Lamp Test (LT) may be activated any time when the BI/RBO node is HIGH. Both devices contain an overriding blanking input (BI) which can be used to control the lamp intensity by varying the frequency and duty cycle of the BI input signal or to inhibit the outputs.

- Lamp Intensity Modulation Capability (BI/RBO)
- Internal Pull-Ups Eliminate Need for External Resistors
- Input Clamp Diodes Eliminate High-Speed Termination Effects

CONNECTION DIAGRAM DIP (TOP VIEW)



LOGIC DIAGRAM



SN54/74LS48

**BCD TO 7-SEGMENT DECODER
LOW POWER SCHOTTKY**



**J SUFFIX
CERAMIC
CASE 620-09**



**N SUFFIX
PLASTIC
CASE 648-08**

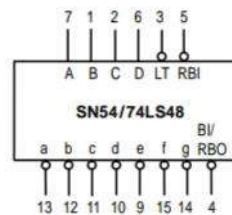


**D SUFFIX
SOIC
CASE 751B-03**

ORDERING INFORMATION

SN54LSXXJ	Ceramic
SN74LSXXN	Plastic
SN74LSXXD	SOIC

LOGIC SYMBOL



VCC = PIN 16
GND = PIN 8

SN54/74LS48

PIN NAMES

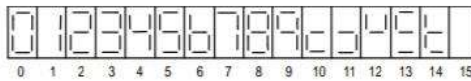
A, B, C, D	BCD Inputs
$\overline{\text{RBI}}$	Ripple-Blanking (Active Low) Input
$\overline{\text{LT}}$	Lamp-Test (Active Low) Input
$\overline{\text{BI/RBO}}$	Blanking Input or Ripple-Blanking Output (Active Low)
$\overline{\text{BI}}$	Blanking (Active Low) Input

LOADING (Note a)

	HIGH	LOW
A, B, C, D	0.5 U.L.	0.25 U.L.
$\overline{\text{RBI}}$	0.5 U.L.	0.25 U.L.
$\overline{\text{LT}}$	0.5 U.L.	0.25 U.L.
$\overline{\text{BI/RBO}}$	0.5 U.L.	0.75 U.L.
	1.2 U.L.	2(1) U.L.
	0.5 U.L.	0.25 U.L.
Open-Collector		3.75 (1.25) U.L. (48)

NOTES:

- a) Unit Load (U.L.) = 40 μ A HIGH/1.6 mA LOW
- b) Output current measured at $V_{\text{OUT}} = 0.5$ V
Output LOW drive factor is SN54LS/74LS48: 1.25 U.L. for Military (54), 3.75 U.L. for Commercial (74).



NUMERICAL DESIGNATIONS — RESULTANT DISPLAYS

**TRUTH TABLE
SN54/74LS48**

DECIMAL OR FUNCTION	INPUTS						OUTPUTS							NOTE	
	$\overline{\text{LT}}$	$\overline{\text{RBI}}$	D	C	B	A	$\overline{\text{BI/RBO}}$	a	b	c	d	e	f		g
0	H	H	L	L	L	L	H	H	H	H	H	H	L	L	1
1	H	X	L	L	L	H	H	L	H	H	L	L	L	L	1
2	H	X	L	L	H	L	H	H	H	L	H	H	L	H	
3	H	X	L	L	H	H	H	H	H	H	H	L	L	H	
4	H	X	L	H	L	L	H	H	L	H	L	L	H	H	
5	H	X	L	H	L	H	H	H	L	H	H	L	H	H	
6	H	X	L	H	H	L	H	L	L	H	H	H	H	H	
7	H	X	L	H	H	H	H	H	H	H	L	L	L	L	
8	H	X	H	L	L	L	H	H	H	H	H	H	H	H	
9	H	X	H	L	L	H	H	H	H	H	L	L	H	H	
10	H	X	H	L	H	L	H	L	L	L	H	H	L	H	
11	H	X	H	L	H	H	H	L	L	H	H	L	L	H	
12	H	X	H	H	L	L	H	L	H	L	L	L	H	H	
13	H	X	H	H	L	H	H	H	L	L	H	L	H	H	
14	H	X	H	H	H	L	H	L	L	L	H	H	H	H	
15	H	X	H	H	H	H	H	L	L	L	L	L	L	L	
$\overline{\text{BI}}$	X	X	X	X	X	X	L	L	L	L	L	L	L	L	2
$\overline{\text{RBI}}$	H	L	L	L	L	L	L	L	L	L	L	L	L	L	3
$\overline{\text{LT}}$	L	X	X	X	X	X	H	H	H	H	H	H	H	H	4

NOTES:

- (1) $\overline{\text{BI/RBO}}$ is wired-AND logic serving as blanking input ($\overline{\text{BI}}$) and/or ripple-blanking output ($\overline{\text{RBO}}$). The blanking out ($\overline{\text{BI}}$) must be open or held at a HIGH level when output functions 0 through 15 are desired, and ripple-blanking input ($\overline{\text{RBI}}$) must be open or at a HIGH level if blanking of a decimal 0 is not desired. X=input may be HIGH or LOW.
- (2) When a LOW level is applied to the blanking input (forced condition) all segment outputs go to a LOW level, regardless of the state of any other input condition.
- (3) When ripple-blanking input ($\overline{\text{RBI}}$) and inputs A, B, C, and D are at LOW level, with the lamp test input at HIGH level, all segment outputs go to a HIGH level and the ripple-blanking output ($\overline{\text{RBO}}$) goes to a LOW level (response condition).
- (4) When the blanking input/ripple-blanking output ($\overline{\text{BI/RBO}}$) is open or held at a HIGH level, and a LOW level is applied to lamp-test input, all segment outputs go to a LOW level.

SN54/74LS48

GUARANTEED OPERATING RANGES

Symbol	Parameter		Min	Typ	Max	Unit
V _{CC}	Supply Voltage	54 74	4.5 4.75	5.0 5.0	5.5 5.25	V
T _A	Operating Ambient Temperature Range	54 74	-55 0	25 25	125 70	°C
I _{OH}	Output Current — High \bar{a} to \bar{g}	54, 74			-100	μA
I _{OH}	Output Current — High $\overline{BI/RB\bar{O}}$	54, 74			-50	μA
I _{OL}	Output Current — Low \bar{a} to \bar{g}	54 74			2.0 6.0	mA
I _{OL}	Output Current — Low $\overline{BI/RB\bar{O}}$ $BI/RB\bar{O}$	54 74			1.6 3.2	mA

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

Symbol	Parameter	Limits			Unit	Test Conditions
		Min	Typ	Max		
V _{IH}	Input HIGH Voltage	2.0			V	Guaranteed Input HIGH Voltage for All Inputs
V _{IL}	Input LOW Voltage	54		0.7	V	Guaranteed Input LOW Voltage for All Inputs
		74		0.8		
V _{IK}	Input Clamp Diode Voltage			-1.5	V	V _{CC} = MIN, I _{IN} = -18 mA
V _{OH}	Output HIGH Voltage	2.4	4.2		μA	V _{CC} = MIN, I _{OH} = -50 μA, V _{IN} = V _{IH} or U.L. per Truth Table
I _O	Output Current \bar{a} to \bar{g}	-1.3	-2.0		mA	V _{CC} = MIN, V _O = 0.85 V Input Conditioner as for V _{OH}
V _{OL}	Output LOW Voltage \bar{a} to \bar{g}	54, 74		0.4	V	V _{CC} = MIN, V _{IH} = 2.0 V V _{IL} = V _{IL} MAX
		74		0.5	V	
V _{OL}	Output LOW Voltage $\overline{BI/RB\bar{O}}$	54, 74		0.4	V	V _{CC} = MAX, V _{IH} = 2.0 V V _{IL} = V _{IL} MAX
		74		0.5	V	
I _{IH}	Input HIGH Current (Except $\overline{BI/RB\bar{O}}$)			20	μA	V _{CC} = MAX, V _{IN} = 2.7 V
				0.1	mA	V _{CC} = MAX, V _{IN} = 7.0 V
I _{IL}	Input LOW Current (Except $\overline{BI/RB\bar{O}}$)			-0.4	mA	V _{CC} = MAX, V _{IN} = 0.4 V
I _{IL}	Input LOW Current $\overline{BI/RB\bar{O}}$			-1.2	mA	V _{CC} = MAX, V _{IN} = 0.4 V
I _{CC}	Power Supply Current		25	38	mA	V _{CC} = MAX
I _{OS}	Short Circuit Current $\overline{BI/RB\bar{O}}$ (Note 1)	-0.3		-2.0	mA	V _{CC} = MAX

Note 1: Not more than one output should be shorted at a time, nor for more than 1 second.

AC CHARACTERISTICS (V_{CC} = 5.0 V, T_A = 25°C)

Symbol	Parameter	Limits			Unit	Test Conditions
		Min	Typ	Max		
t _{PHL}	Propagation Delay Time, HIGH-to-LOW Level Output from A Input			100	ns	C _L = 15 pF, R _L = 4.0 kΩ
t _{PLH}	Propagation Delay Time, LOW-to-HIGH Level Output from A Input			100	ns	
t _{PHL}	Propagation Delay Time, HIGH-to-LOW Level Output from $\overline{RB\bar{I}}$ Input			100	ns	C _L = 15 pF, R _L = 6.0 kΩ
t _{PLH}	Propagation Delay Time, LOW-to-HIGH Level Output from $\overline{RB\bar{I}}$ Input			100	ns	

FAST AND LS TTL DATA

3. L'afficheur 7 segments 5161AS :

XLITX®

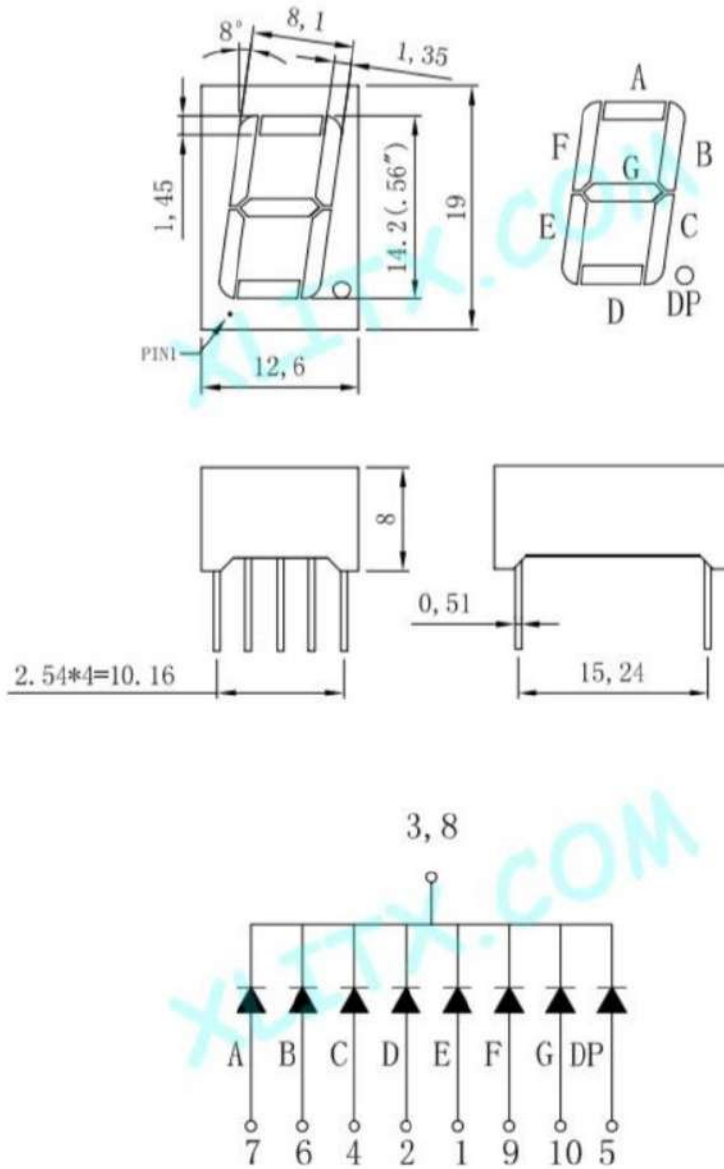
www.xlitx.com



Model :	5161AS
Size :	0.56-inch
Emitting color :	Red (Ultra-Bright)
Mode :	Common-Cathode (CC)
Digit :	Single Digit
Category :	LED 7-Segment Display
Maker :	XLITX Technology

Prices : \$0.45 = 100 Pieces \$0.21 = 500 Pieces \$0.16 = 1000 Pieces (Free shipping)

Dimension & Diagram



Ultra-Bright Red Electro-Optical Parameter

1. Electro-Optical Characteristics(Ta=25°C)

PARAMETER	SYMBOL	DEVICES (ULTRA-BRIGHT RED)		UNIT	TEST CONDIONS
		TYP	MAX		
Peak Emission Wavelgrth	λ_p	640		nm	IF=10mA
Forward Voltage	VF	1.8		V	IF=10mA
Reverse Current	IR		50	μ A	VR=5V
Segment To Segment (Dot To Dot) Luminonous Intensity Ratio	IV-M	1.5:1			IF=20

2. Absolute Maximun Ratings(Ta=25°C)

PARAMETER	SYMBOL	DEVICES (ULTRA-BRIGHT RED)	UNIT
Power Dissipation Per Dice	pad	100	mw
Derating Linear From 25°C Per Dice		0.5	mA/°C
Continuous Forward Current Per Dice	Iaf	30	mA
Peak Forward Ourrent Per Dice (Duty Cycle 1/10,10KHz)	Ipf	200	mA
Reverse Voltage Per Dice	Vr	5	V
Operating Temperature	Topr	-20°C (to) +75°C	
Storage Temperature	Tstg	-20°C (to) +85°C	
Solder Temperature		1.6 Inch Below Seating Place for 5 seconds at 230°C	

4. 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 μ A 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.



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.

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).

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

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

Remote 8-bit I/O expander for I²C-bus

PCF8574

4 BLOCK DIAGRAM

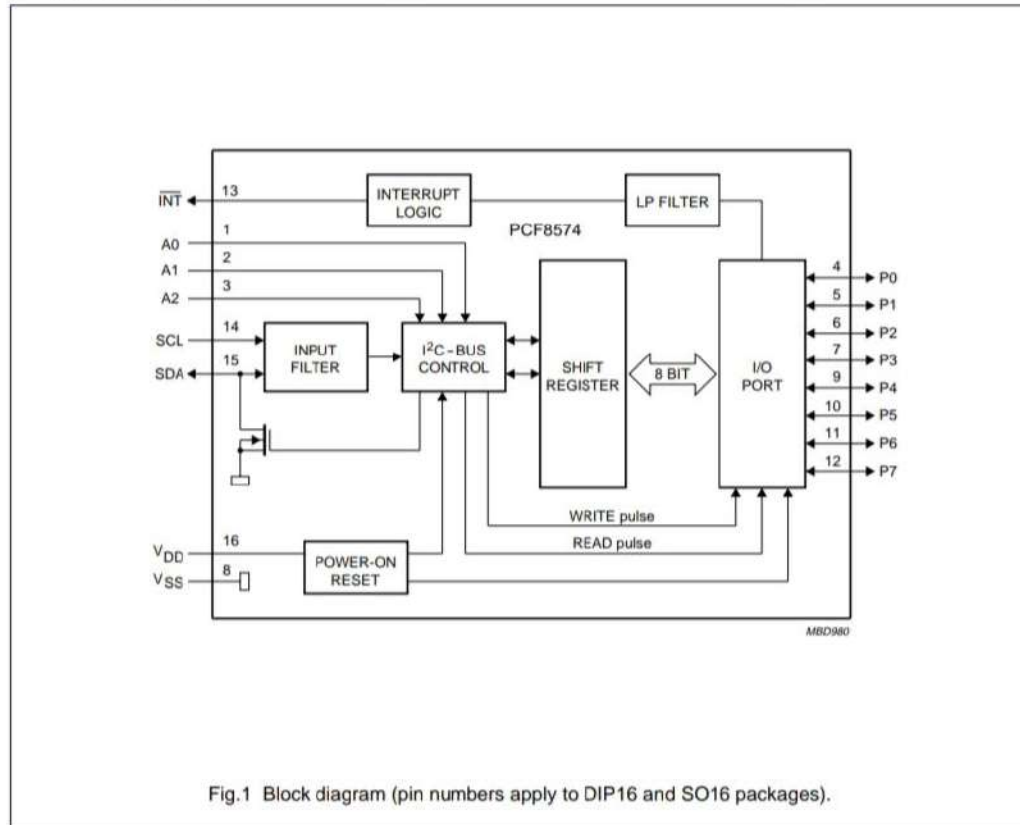


Fig.1 Block diagram (pin numbers apply to DIP16 and SO16 packages).

Remote 8-bit I/O expander for I²C-bus

PCF8574

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
$\overline{\text{INT}}$	13	interrupt output (active LOW)
SCL	14	serial clock line
SDA	15	serial data line
V _{DD}	16	supply voltage

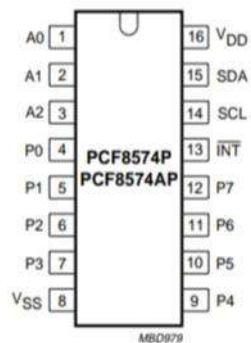


Fig.2 Pin configuration (DIP16).

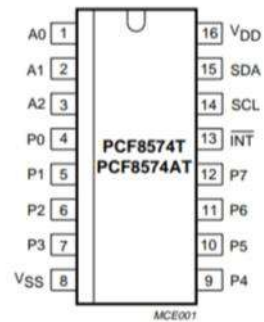


Fig.3 Pin configuration (SO16).

Bibliographie

Bibliographie

- [1] Lbouhmadi M, Laayoun J «Etude d'ascenseur commandé par automate programmable », Projet de fin d'étude, Université sidi Mohammed ben Abdallah, Fès, Maroc 2007.
- [2] <http://www.energieplus-lesite.be>
- [3] Zidane Taous «Développement d'un outil numérique destiné à l'étude de performances dynamiques d'un entraînement électrique: Application aux ascenseurs», Projet de fin d'étude en électrotechnique, Université Mouloud Mammeri, Tizi-Ouzou.
- [4] Mbow M, Diocou I « Etude de la commande d'un ascenseur», Projet de fin d'étude, Centre National de qualification professionnelle, Senegal 2019-2020.
- [5] VASEUX Loïc & OUINE Corentin, Lycée Le Corbusier Dossier, TPE2007-2008
- [6] Embouazza L, Mostefaoui I «Etude, conception et réalisation d'un prototype d'ascenseur commandé à base d'un Microcontrôleur (PIC) », Projet de fin d'étude en électronique, Université Aboubakr Belkaïd-Tlemce.
- [7] Bouguetof H, Bentchakal D, Bensenouci S, «Etude d'un ascenseur», Projet de fin d'étude en ETH, UMMTO 2002.
- [8] Mammeri S, Maouchi T « 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.
- [9] Boualam L, Hachiche N «Conception et réalisation d'une carte de commande d'une maquette d'ascenseur à base d'une carte Arduino Mega2560», Projet de fin d'études en automatique, Université Mouloud Mammeri de Tizi-Ouzou.
- [10] [Ascenseurs.free.fr/ type fonct/ward_leonard_a_Contacteurs](http://ascenseurs.free.fr/type_fonct/ward_leonard_a_Contacteurs)
- [11] https://www.pdfprof.com/PDF_Image.php?id=22539&t=17
- [12] Christian Tavenir, «Arduino, maîtrisez sa programmation et ces cartes d'interface (Schiels)», Collection Dunod, Mars 2014, 256p
- [13] Massimo Banzi, Michael Shiloh «Getting started with Arduino », 3rd Edition, Décembre 2014, 245p
- [14] Madi K, Asloun K, «Etude commande d'un ascenseur à base d'Arduino», Projet de fin d'études en automatique, Université A. MIRA-Bejaia.
- [15] <http://www.elektronique.fr/>
- [16] www.alldatasheet.com
- [17] <https://openclassrooms.com/fr/courses/3290206-perfectionnez-vous-dans-la-programmation-arduino/3342221-programmez-un-ecran-lcd>

Bibliographie

- [18] <https://www.eaton.com/ca/fr-ca/products/controls-drives-automation-sensors/sensors---limit-switches/understanding-sensors-and-limit-switches--.html>
- [19] <http://www.louisreynier.com>
- [20] Djafri M, Chelouche D, «Etude et réalisation d'une carte Arduino », Projet de fin d'études en électronique, Université A.MIRA de Bejaia.
- [21] Brian W. Evans «Arduino programming Notebook », First edition, August 2007, 38p
- [22] [www.arduino .cc](http://www.arduino.cc)
- [23] SIMON Landranlt, HIPPOLYTE Weisslinger, «Premier pas en informatique embarquée», Edition CC BY-NC-SA, 01 Juin 2014, 268p.
- [24] www.fritzing.org
- [25] http://technomoussi.free.fr/IMG/pdf/SEQ25-S2Ressource_Capteurs_Fin_de_course.pdf
- [26] http://lycees.acrouen.fr/maupassant/Melec/co/Techno/Detecteur/web/co/grain_Detect_meca.html
- [27] Souaidia Zohra «Etude et simulation d'un monte-charge par Arduino», Mémoire de Master, Université Badji Mokhtar Annaba.