

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة أيب بكر بلقايد— تلمسان

Université Aboubakr Belkaïd-Tlemcen

كلية التكنولوجيا

Faculté de Technologie

Département de Génie Electrique et Electronique (GEE)

Filière : Electronique



MASTER INSTRUMENTATION

PROJET DE FIN D'ETUDES

Présenté par : LARIBI Abdelmadjid & CHIKH Zakaria

Intitulé du Sujet

Étude, innovation, conception et simulation d'un prototype d'ascenseur pour personnes a mobilité réduite, commandé par Carte Arduino..

Soutenu en 2020, devant le jury composé de :

Mr LACHACHI Djamel

MCB

Univ. Tlemcen

Président

Mr BELARBI Boumediene

MAA

Univ. Tlemcen

Encadreur

Mr MASSOUM Nouredine

MCB

Univ. Tlemcen

Examineur

Année Universitaire 2019-2020

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

# Dédicace

*Je dédie ce travail à ma chère famille considérée comme ma force et mon courage. Je commence tout d'abord avec mes parents*

*À ma très chère mère NACERA*

*Vous êtes considéré comme ma vie et mon amour les mots ne seront jamais suffisant pour décrire mon amour que j'ai pour elle, c'est une mère, une sœur, une copine elle est tous pour moi, elle nous a grandi, souffert et combattu le monde pour nous rendre heureux,  
C'est notre héroïne.*

*Oh ! Mère, Tu es l'exemple du dévouement,  
qui n'a pas cessé de m'encourager et de prier pour moi.*

*À mon Père, ABDELGHANI*

*Tu as toujours été pour moi un exemple du père respectueux, honnête, de la personne méticuleuse, je tiens à honorer l'homme que tu es.*

*Grâce à toi papa j'ai appris le sens du travail et de la responsabilité. Je voudrais te remercier pour ton amour, ta générosité, ta compréhension... Ton soutien fut une lumière dans tout mon parcours*

*À mes braves frères*

*HEMZA et KHALED, mes coéquipiers et mes soldats, y qui m'ont toujours soutenu et que je sais qu'ils sont toujours là pour moi. Passant à ma grande sœur ASMA et à sa petite famille sont marie ADEL et ces enfants Anis et Amira , Merci de si bien accomplir ton rôle , Merci de me suivre et de m'encourager dans mes projets et dans mes rêves les plus fous; tu me fais toujours sentir que tu es derrière moi et que tu crois en moi et ton support fait une grande différence dans ma vie.*

*À mes amies*

*EL HADI, HICHEM, RIAD, SALIM, LOTFI, ANES, FETHI, Je ne peux trouver les mots justes et sincères pour vous exprimer mon affection et mes pensées, vous êtes pour moi des frères et des amies sur qui je peux compter.*

*En témoignage de l'amitié qui nous unit et des souvenirs de tous les moments que nous avons passés ensemble, je vous dédie ce travail et je vous souhaite une vie pleine de santé et de bonheur.*

*À mes chers collègues*

*Vous êtes si nombreux, je ne vais pas vous mentionner tous j'espère que je vous ai honoré, et que je vous ai rendue fière, je souhaite que vous n'êtes pas déçu et merci pour votre confiance.*

*Sans oublier tous les enseignants du laboratoire ELN  
Un profond respect pour leur dévouement*

# *Dédicace*

*Ma mère SAMIRA la source et l'espoir ! Mon père YAHYA le repère et l'exemplaire ! A vous que je dédie  
ce modeste mémoire pour m'avoir encouragé,*

*Poussé pour atteindre l'idéal et m'ont soutenu tout au long de mes études. J'espère qu'un jour, je peux leurs  
rendre un peu de ce qu'ils ont fait pour moi, que "Dieu" leurs prête bonheur et longue vie.*

*Je dédie ce modeste travail à : À mes chers Sœurs ... KAWTER ET FARAH...*

*À toute ma famille petite et grande.*

*À mes Cousins & Cousines*

*À Tous mes fidèles amis : ABDELATIF, RIAD, MADJID Une pensée  
affectueuse A mon chère neveux AYMEN*

*Sans oublier Tous les enseignants et étudiants du département électronique et particulièrement les étudiants de  
la filière instrumentation promotion 2019 Je dédie enfin ce mémoire à toute personne ayant contribué de près  
ou de loin À sa concrétisation.*

# Remerciement

*En préambule à ce mémoire nous remercions ALLAH le tout puissant et miséricordieux, qui nous a donné le courage, la force et la foi de mener à terme ce modeste travail.*

*À notre encadreur Monsieur BELARBI Boumediene*

*Nous tenons d'abord à le remercier très chaleureusement d'avoir accepté de suivre notre projet et pour son attention particulière qu'il nous a donnée au courant de l'année, pour ses conseils indispensables, sa disponibilité, son aide, ses orientations, le temps consacré et son enthousiasme envers ce mémoire, qui ont constitués un apport considérable grâce auquel ce travail a pu être mené à bon port.*

*Nos vifs remerciements vont également aux membres du jury, qui ont bien voulu nous honorer par leur précieuse présences parmi nous, afin d'examiner d'évaluer ce modeste travail.*

*À notre Président Pr. LACHECHI DJAMEL*

*Qui nous a donné le privilège d'avoir accepté de présider le jury de la soutenance de notre PFE. Vos qualités, vos valeurs, votre sérieux, votre compétence et votre sens du devoir nous ont énormément marqués. Veuillez trouver ici l'expression de notre respectueuse considération et notre profonde admiration pour toutes vos qualités scientifiques et humaines. Ce travail est pour nous l'occasion de vous témoigner notre profonde gratitude.*

*À notre examinateur*

*Vous nous avez honorés d'accepter avec grande sympathie de siéger parmi notre jury de thèse.*

# SOMMAIRE

<b>Introduction générale .....</b>	<b>1</b>
<b>Chapitre 01 .....</b>	<b>3</b>
<b>Systèmes Automatisés et Généralité sur les Ascenseurs .....</b>	<b>3</b>
<b>I. Introduction : .....</b>	<b>4</b>
<b>I.1 Définition d'un système : .....</b>	<b>4</b>
<b>I.1.1 Définition d'un système automatisés : .....</b>	<b>4</b>
<b>I.1.2 Structure d'un système automatisés : .....</b>	<b>5</b>
<b>I.2 les Ascenseurs : .....</b>	<b>5</b>
<b>I.2.1 Définition d'un Ascenseur : .....</b>	<b>5</b>
<b>I.2.2 Structure d'un Ascenseur : .....</b>	<b>6</b>
<b>I.3 Les différents types d'ascenseur : .....</b>	<b>8</b>
<b>I.3.1 Ascenseur de charge : .....</b>	<b>8</b>
<b>1.3.2 Ascenseur pour le transport de personnes handicapées : .....</b>	<b>8</b>
<b>1.3.3 Ascenseur sur plan incliné : .....</b>	<b>9</b>
<b>I.4 Les familles d'Ascenseur : .....</b>	<b>9</b>
<b>I.4.1 Les ascenseurs hydrauliques : .....</b>	<b>10</b>
<b>I.4.2 Les ascenseurs à traction à câble : .....</b>	<b>12</b>
<b>I.4.3 Critères de choix du type d'ascenseur : .....</b>	<b>13</b>
<b>1.5 Principe de fonctionnement d'un ascenseur à traction : .....</b>	<b>17</b>
<b>Chapitre 02 .....</b>	<b>18</b>
<b>Etude de la carte Arduino et logiciel de programmation .....</b>	<b>18</b>
<b>II.1. Introduction .....</b>	<b>19</b>
<b>II.2 La carte Arduino : .....</b>	<b>19</b>
<b>II 2.1 Définition : .....</b>	<b>19</b>
<b>II.2.2. Les gammes de la carte Arduino : .....</b>	<b>20</b>
<b>II.2.3. Architecture et composants de la carte : .....</b>	<b>21</b>
<b>II.3. Le logiciel Arduino (IDE Arduino) .....</b>	<b>25</b>
<b>II.3.1. Structure d'un programme .....</b>	<b>27</b>
<b>II.3.2. Syntaxe du langage Arduino .....</b>	<b>28</b>
<b>II.4. Conclusion .....</b>	<b>30</b>
<b>Chapitre 03 .....</b>	<b>31</b>
<b>Conception et programmation .....</b>	<b>31</b>
<b>III.1. Introduction.....</b>	<b>32</b>
<b>III.2. Etude théorique .....</b>	<b>33</b>
<b>III.2.1. Partie mécanique .....</b>	<b>33</b>

<b>III.2.2. Partie électronique pour l'ASCENSEUR .....</b>	<b>33</b>
<b>III.2.2.1. Le microcontrôleur (carte Arduino UNO R3) .....</b>	<b>33</b>
<b>III.2.2.2. Le moteur : .....</b>	<b>36</b>
<b>III.2.2.3. Le L293D : .....</b>	<b>40</b>
<b>III.2.3. Partie électronique pour la plateforme mobile : .....</b>	<b>42</b>
<b>III.2.3.1 Le module L298N : .....</b>	<b>42</b>
<b>III.2.3.2. Le moteur : .....</b>	<b>43</b>
<b>III .2.3.3 Autre matériels : .....</b>	<b>44</b>
<b>III .2.3.3 1 Afficheur 7segment .....</b>	<b>44</b>
<b>III.3. Etude pratique .....</b>	<b>45</b>
<b>III.3.1. Partie logiciel : .....</b>	<b>45</b>
<b>III.3.1.1.Présentation de Proteus .....</b>	<b>45</b>
<b>III.3.2. Partie matériels (conception des cartes) .....</b>	<b>46</b>
<b>III.4. Programmation .....</b>	<b>49</b>
<b>III.4.1. Cahier des charges .....</b>	<b>49</b>
<b>III.4.2 Organigramme de déroulement du programme: .....</b>	<b>51</b>
<b>III.4.3 Réalisation du programme .....</b>	<b>52</b>
<b>III.4.3.1 Réalisation du programme de l'ascenseur via l'environnement de .....</b>	<b>52</b>
<b>III.4.3.2 Réalisation du programme de la plateforme via l'environnement de .....</b>	<b>54</b>
<b>III.4.3.3 Conclusion : .....</b>	<b>54</b>
<b>Conclusion générale .....</b>	<b>55</b>



## Liste des figures

Figure I. 1. Structure d'un système automatisé.....	5
Figure I. 2: structure et composition d'un ascenseur.....	7
Figure I. 3 : Les familles d'ascenseurs.....	9
Figure I. 4: Schéma descriptif du principe de fonctionnement des ascenseurs hydrauliques. ....	10
Figure I. 5: Les différents modèles de l'ascenseur hydraulique.....	11
Figure I. 6: Les types d'ascenseur à traction à câble.....	12
Figure I. 7 : les différentes parties d'un ascenseur à traction .....	14
Figure I. 8 : Principe de fonctionnement ascenseur à traction.....	17
Figure II. 1: Carte Arduino .....	19
Figure II. 2 : Le microcontrôleur de la carte Arduino .....	22
Figure II. 3 : Les 14 entrées sorties numériques. ....	23
Figure II. 4: Les 6 entrées sorties analogiques. ....	23
Figure II. 5: Le schéma électrique de l'alimentation de la carte Arduino. ....	24
Figure II. 6: Les composants de la carte Arduino (Uno). ....	24
Figure II. 7: Les différentes parties de la fenêtre principale du logiciel Arduino. ....	25
Figure II. 8: Exemple d'un programme Arduino .....	27
Figure III. 1: Microcontrôleur Carte Arduino UNO .....	34
Figure III. 2: Le moteur de traction. ....	37
Figure III. 3 : Le pont en H. ....	38
Figure III. 4: Fonctionnement dans le sens direct.....	38
Figure III. 5 : Fonctionnement dans le sens indirect.....	38
Figure III. 6 : Freinage avec S1 et S2. ....	39
Figure III. 7 : Freinage avec S3 et S4. ....	39
Figure III. 8: Descriptif du module L298N .....	42
Figure III. 9 : Prise en 3d du câblage du moteur pas à pas .....	43
Figure III. 10 : Afficheur 7segment .....	44
Figure III. 11: Fenêtre du module de simulation Proteus ISIS.....	46
Figure III. 12: Schéma de conception de la carte d'alimentation sur ISIS. ....	47
Figure III. 13 : Implantation des composants.....	47
Figure III. 14 : PCB de la carte d'alimentation.....	47
Figure III. 15: Schéma de conception de la carte d'afficheur sur ISIS. ....	48
Figure III. 16: Implantation des composants.....	48
Figure III. 17 : Circuit imprimé de la carte d'afficheur. ....	48
Figure III. 18: conception de schéma de la carte de commande des afficheurs sur ISIS.....	49
Figure III. 19: Organigramme de fonctionnement de l'ascenseur.....	51

## Liste des tableaux

<b>Tableau II.1 : Tableau comparatif de différentes cartes Arduino .....</b>	<b>21</b>
<b>Tableau II.2:commandes du programme Arduino.....</b>	<b>28</b>
<b>Tableau II.3: Les différentes variables du programme Arduino. ....</b>	<b>29</b>
<b>Tableau II.4:Les variables du programme Arduino .....</b>	<b>29</b>
<b>Tableau III. 1 : Brochage du L293D.....</b>	<b>41</b>

# **Introduction Générale**

---

## Introduction générale

---

Depuis la plus haute antiquité les hommes ont cherché un moyen de favoriser le déplacement vertical des charges : la construction des pyramides, l'architecture romaine et leurs vestiges nous en laissent les preuves.

Jusqu'à la Renaissance c'est surtout le transport et le levage des marchandises qui préoccupent les inventeurs, puis au XVIIème au XVIIIème siècle apparaissent de nouveaux besoins en matière de transports des personnes.

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

De ce fait Le monde de l'industrie à connu ces dernières années des bouleversements importants, les machines automatisées ont été développées pour libérer l'homme de ses tâches quotidiennes, et aussi pour le remplacer dans l'exécution d'un certain nombre de travaux.

Les progrès de l'électronique et de l'informatique, ont donné naissance aux automates programmables industriels (API) et d'autres microcontrôleurs qui peuvent s'adapter et s'intégrer dans les processus industriels.

Ils peuvent accomplir des tâches plus complexes, non seulement de contrôle, mais aussi de traitement de données, de circulation d'informations et de simulation.

Les ascenseurs et les montes charges ont toujours constitués un moyen de transport et d'aide pour l'être humain.

Pendant l'utilisation de ces dispositifs la sécurité des usagers doit toujours être assurée, les constructeurs doivent assurer en même temps le confort et la sécurité des utilisateurs,

La tâche la plus difficile se trouve dans la réalisation est la gestion des conflits et le contrôle en temps réel. C'est dans ce cadre que s'inscrit notre projet.

En effet, ce projet consiste à étudier, à concevoir et à réaliser une carte de commande d'un ascenseur pour usage de personne à mobilités réduites à base d'une carte Arduino uno R3

Ce rapport présente le déroulement du projet et permet de suivre la progression de notre travail ainsi que les résultats obtenus et les améliorations possibles.

L'objectif initial de ce projet était de réaliser une maquette d'ascenseurs à quatre étages avec une plateforme mobile commandée intégrées au sol de l'ascenseur qui peut être utilisée par l'ensemble majorités des usages y compris des gens en mobilité réduite.

Et pour cela notre projet est composé de trois chapitres, on commencera tout d'abord par **une introduction générale** ensuite **le Premier chapitre** qui sera consacré à une présentation sur l'ascenseur, ses différents types et modes de fonctionnement, nous évoquant aussi les différentes catégories des ascenseurs avec leurs avantages et inconvénients, ensuite nous allons aborder les différentes parties de l'ascenseur de traction qui sera notre intérêt à ce projet. Par la suite **le deuxième chapitre** représentera une étude théorique sur le la carte de commande Arduino, son architecture ainsi que sa composition, nous citron aussi les différents types de ce dernier.

A la fin **Le troisième chapitre** qui sera dédié à l'interface de programmation.

Ce dernier sera consacré exclusivement à tout le travail réalisé d'un point de vue conception, réalisation et traitement de données.

Et on terminera par **une conclusion générale**.

# **Chapitre 01**

## **Systemes Automatisés et Généralité sur les Ascenseurs**

## **I. Introduction :**

Longtemps considéré comme un luxe, l'ascenseur est devenu de nos jours un élément

Indispensable dans la vie quotidienne.

Les ascenseurs constituent toujours un moyen de transport et d'aide pour l'homme lors de ces

Déplacements dans de hauts immeubles. Les usagers de ces dispositifs doivent se déplacer en sécurité et dans le confort.

Dans ce chapitre nous présenterons brièvement en premier lieu le concept d'un système automatisé les généralités sur les ascenseurs, la deuxième partie sera consacrée à la définition des différents éléments constituant

### **I.1 Définition d'un système :**

Un système est un ensemble d'éléments organisés en interaction mutuel et avec l'environnement pour raison d'atteindre un ou plusieurs tâches prédéfinis.

#### **I.1.1 Définition d'un système automatisés :**

Simple ou complexes, les systèmes automatisés sont partout dans notre environnement quotidien. Connaître leur fonctionnement permet aussi de mieux comprendre notre environnement.

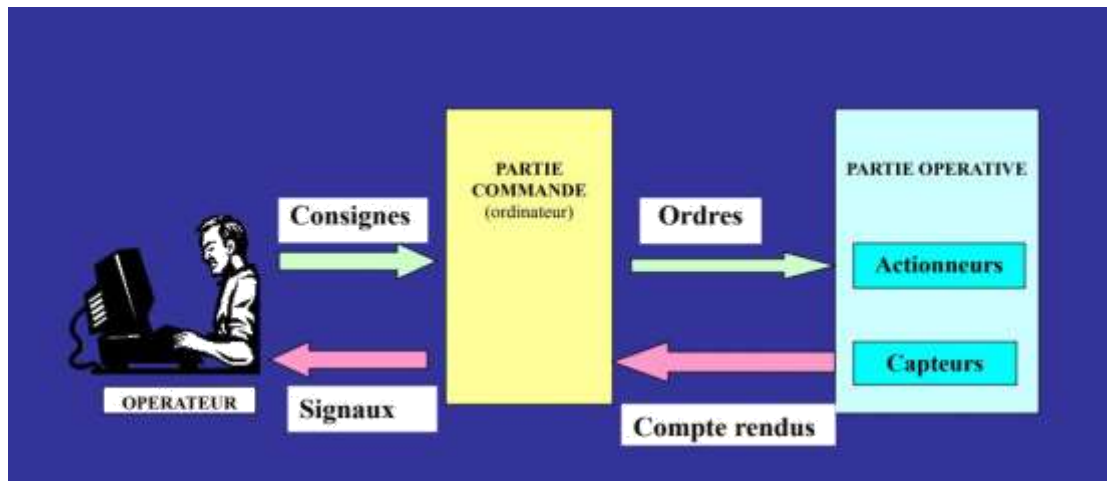
Un système est dit automatisé s'il exécute toujours le même cycle de travail après avoir reçu les consignes d'un opérateur.

Les systèmes automatisés peuvent être de nature technique très divers cela implique que différentes disciplines se rencontrent tel que : l'automatique, l'informatique, la mécanique, l'électronique, l'électrotechnique.

Plus le système est complexe, plus les disciplines concernées sont nombreuses.

Un système automatisé est composé d'une partie commande et d'une partie opérative.

## I.1.2 Structure d'un système automatisés :



*Figure I. 1. Structure d'un système automatisé.*

### **Partie Opérative(P.O):**

Dont les actionneurs agissent sur le processus automatisé.

### **Partie Commande (P.C):**

Coordonne les actions de la partie opérative

### **Partie Dialogue (P.D) :**

On l'appelle aussi la partie relation qui comporte le pupitre de dialogue homme-machine, équipé des organes de commande (boutons d'appel).

Dans un ascenseur, l'ensemble électromécanique (cabine, moteur, portes,...) constitue la partie opérative, les boutons d'appel, la logique (combinatoire, séquentielle, programmée) constituent la partie commande.

## **I.2 les Ascenseurs :**

### **I.2.1 Définition d'un Ascenseur :**

L'ascenseur est un dispositif mobile permettant le déplacement de personnes ou d'objets sur un axe prédéfini au sein d'une construction. Ce dernier est composé d'une cabine dans les dimensions sont prédéfinies par le constructeur et les tâches effectuées par l'ascenseur

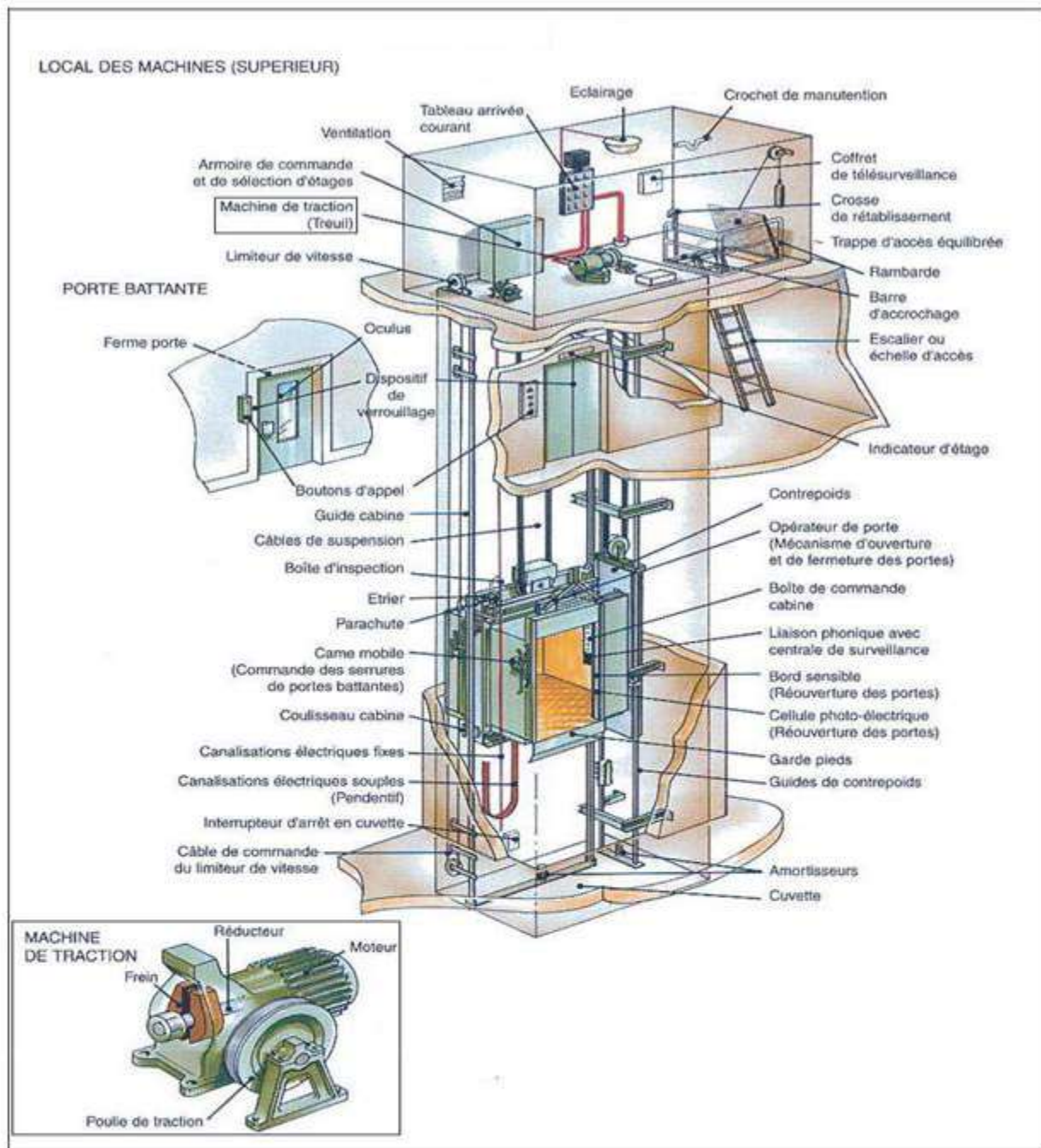
Les caractéristiques de l'ascenseur sont en étroite relation avec le lieu d'implantation et la fonction de ce dernier



### **I.2.2 Structure d'un Ascenseur :**

La composition d'un ascenseur est presque la même qu'avant le 19e siècle, les ascenseurs sont construits en bois, à la suite de la révolution de l'ascenseur, ils sont construits avec des matériaux plus solides (notamment avec du métal) à quelques exceptions près ils sont tous composés des matériels suivants ;

- **Bouton d'appel** : Bouton qui commande l'arrivée de la cabine d'ascenseur.
- **Rail-guide de la cabine** : Barre métallique dans laquelle glisse la cabine et qui l'empêche d'osciller latéralement.
- **Régulateur de vitesse** : Appareil permettant le déclenchement du parachute de cabine en cas de sur-vitesse.
- **Parachute de cabine** : Dispositif de sécurité qui permet d'immobiliser la cabine d'ascenseur en cas de sur-vitesse ou en cas de rupture ou d'amollissement du câble de levage.
- **Rail-guide de contrepoids** : Barre métallique dans laquelle glisse le contrepoids et qui l'empêche d'osciller latéralement.
- **Interrupteur de fin de course** : Dispositif qui permet l'arrêt de la cabine d'ascenseur à chaque niveau.
- **Poulie de tension du régulateur** : Dispositif servant à maintenir tendu le câble du régulateur de vitesse
- **Amortisseur** : Dispositif assurant l'arrêt de la cabine d'ascenseur tout en absorbant l'impact.
- **Cabine d'ascenseur** : Compartiment mobile de l'ascenseur, destiné à recevoir les passagers à transporter.
- **Câble de levage** : Câble permettant le déplacement vertical de la cabine d'ascenseur.
- **Treuil** : Appareil assurant le déplacement de la cabine par l'intermédiaire du câble de levage.



*Figure I. 2: structure et composition d'un ascenseur.*

### **I.3 Les différents types d'ascenseur :**

#### **I.3.1 Ascenseur de charge :**

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

##### ➤ **Monte-charge :**

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

- ✓ surface = 1 m<sup>2</sup>;
- ✓ profondeur = 1 m;
- ✓ hauteur = 1,20 m;

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

##### ➤ **Monte-charge industriel :**

Un appareil de levage installé à l'industrie, desservant des niveaux définis, qui comporte une cabine ou un plateau accessible aux personnes pour le chargement ou déchargement, qui se déplace le long d'un ou de plusieurs guides verticaux ou dont l'inclinaison est inférieure par rapport à la verticale à 15 degrés.

La commande de ces appareils ne peut se faire que de l'extérieur.

Ces dispositifs sont interdits au transport de personnes.

##### ➤ **Monte voitures :**

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

#### **1.3.2 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.

### 1.3.3 Ascenseur sur plan incliné :

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

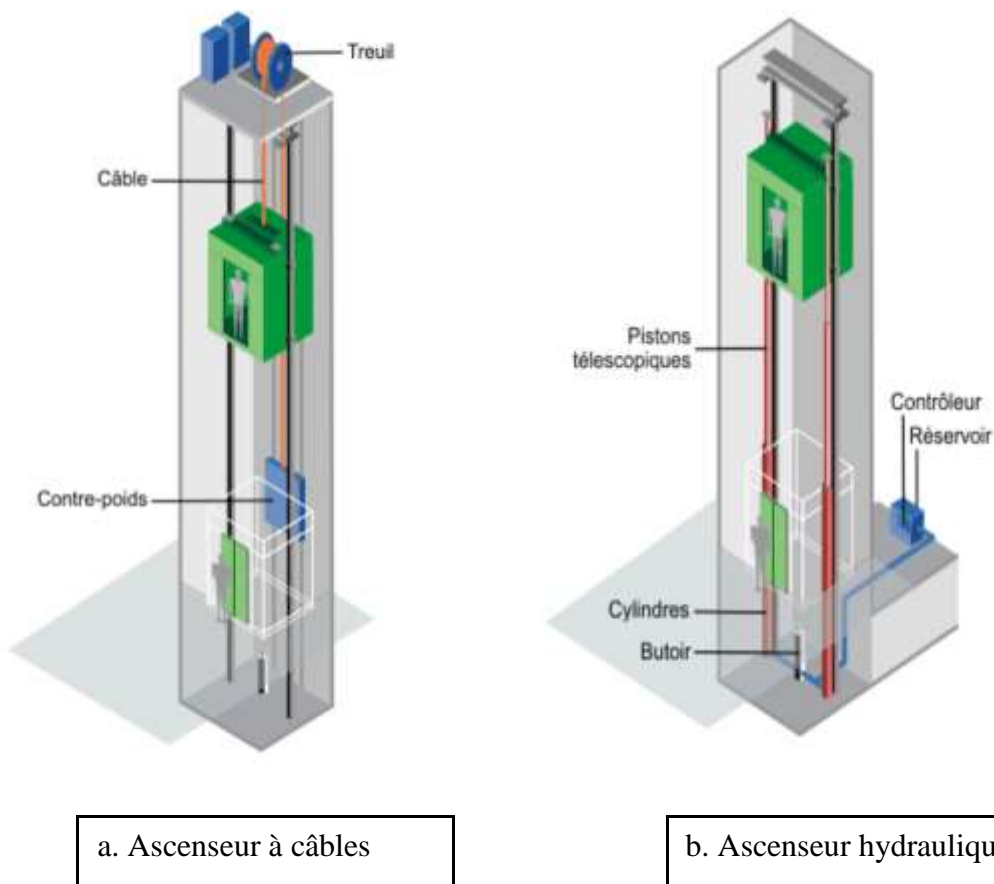
Les ascenseurs sur plan incliné peuvent être installés dans des cages d'escalier droites ou présentant des virages. Ils peuvent également être installés à l'extérieur des bâtiments dans des rampes ou escalier d'accès

### I.4 Les familles d'Ascenseur :

On distingue essentiellement deux types de familles d'ascenseur:

- Les ascenseurs à traction à câble ou électrique.
- Les ascenseurs hydrauliques.

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

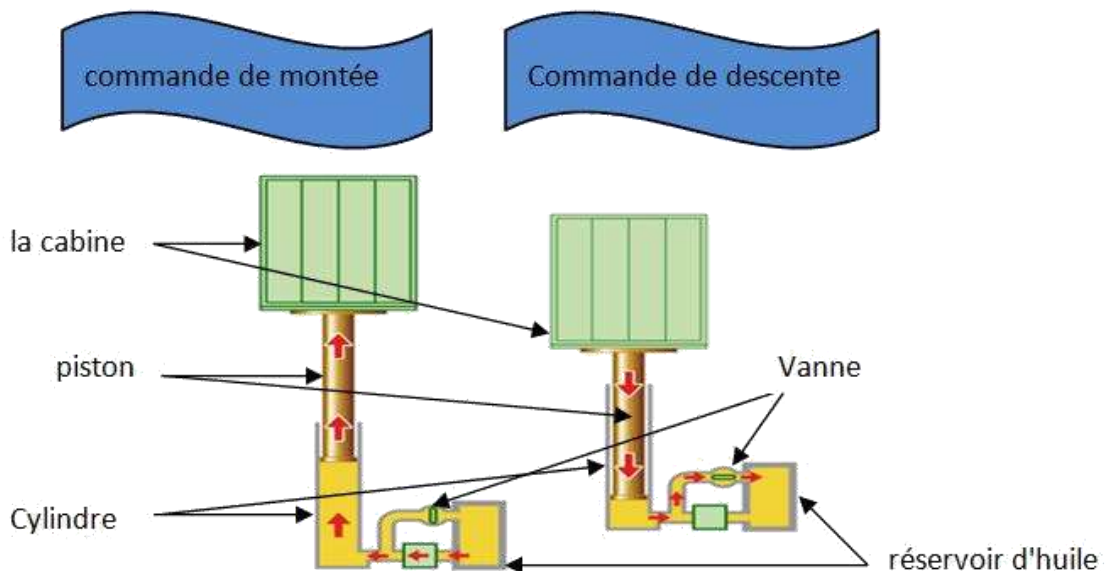


*Figure I. 3 : Les familles d'ascenseurs.*

#### I.4.1 Les ascenseurs hydrauliques :

##### ➤ 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.

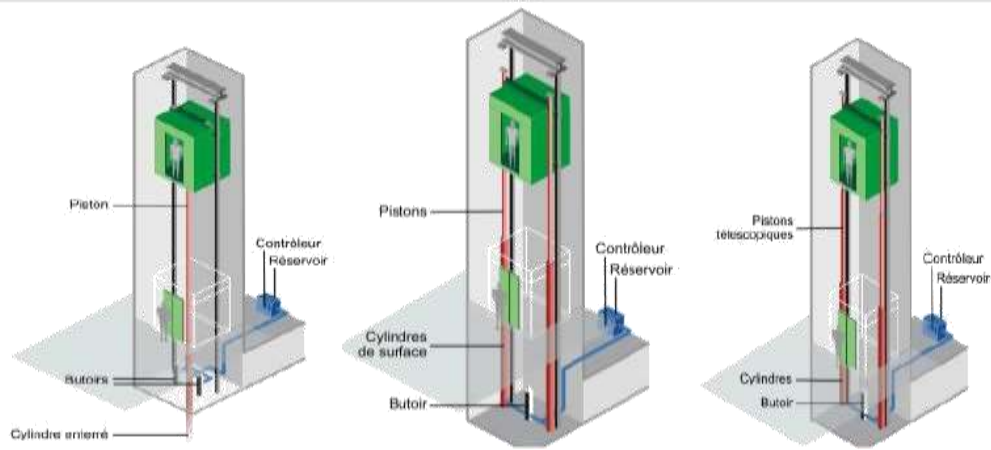


*Figure I. 4: Schéma descriptif du principe de fonctionnement des ascenseurs hydrauliques.*

##### ➤ Description :

Les ascenseurs hydrauliques sont utilisés en générale pour satisfaire des déplacements relativement courts de l'ordre de 15 à 18 m maximums. Plusieurs modèles existent sur le marché on citera :

- Les ascenseurs hydrauliques à cylindre de surface;
- Les ascenseurs hydrauliques à cylindre enterré;
- Les ascenseurs hydrauliques télescopiques à cylindre de surface.



a. A cylindre enterré

b. A cylindre de surface

c. A cylindre de surface  
Télescopique

*Figure I. 5: Les différents modèles de l'ascenseur hydraulique.*

**Les différents modèles permettent de tenir en compte les critères suivants:**

- De place;
- De hauteur d'immeuble à desservir;
- De stabilité de sol et de la rigidité du sous-sol;
- De risque de contamination et de pollution vis-à-vis le sol et plus spécifiquement aux nappes phréatiques;
- D'esthétique et de l'aspect visuel .... etc.

**Les ascenseurs hydrauliques se composent principalement de:**

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

**Bilan Energétique :**

En se basant sur leur bilan énergétique 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

**Avantages et inconvénients :**

Ci-dessous, on citera les avantages majeurs et inconvénients des ascenseurs hydrauliques:

## Avantages

- ✓ Précision au niveau du déplacement ;
- ✓ variation de la vitesse de déplacement est facile ;
- ✓ Ne nécessite pas de salle de machinerie ;
- ✓ Implantation facile dans un immeuble existant ;
- ✓ Etc.

## Inconvénients

- ✓ une hauteur limitée entre 15 et 18 m;
- ✓ Risque de propagation de pollution des sous-sols;
- ✓ bilan énergétique élevé ;
- ✓ nécessité un renforcement du sol;

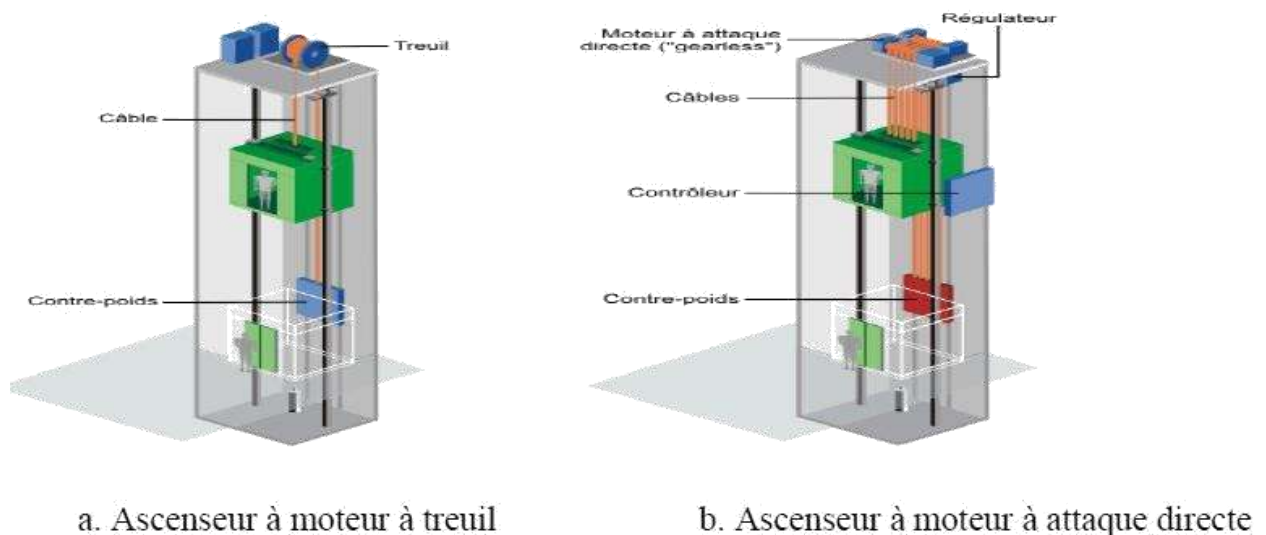
### **I.4.2 Les ascenseurs à traction à câble :**

#### **I.4.2.1 Description :**

Les ascenseurs à traction à câbles sont les types d'ascenseurs dite les plus utilisés, il existe deux types.

Ils se différencient entre eux selon leur type de motorisation :

- ✚ À moteur à r treuil,
- ✚ À moteur à attaque directe (appelé aussi "Gearless" ou sans treuil),



**Figure I. 6: Les types d'ascenseur à traction à câble.**

Les ascenseurs à traction à câble comprennent généralement et cela est pour n'importe quel type:

- ✓ Une cabine;
- ✓ Des contrepoids;
- ✓ Des câbles assurant la liaison entre la cabine et les contrepoids;
- ✓ Des guides;
- ✓ Un système de traction au-dessus de la cage de l'ascenseur;

#### **Bilan Energétique :**

Sachant que les ascenseurs hydrauliques ce sont montrer moins intéressantes sur l'aspect Energétique, les ascenseurs à traction à câble ce sont montrer plus efficaces dans le sens où le contrepoids réduit fortement la charge quel que soit le type de la motorisation.

#### **Avantages et inconvénients :**

##### Avantages

- ✓ la hauteur de déplacement n'est pas vraiment limitée;
- ✓ Suivant le type de motorisation précision au niveau de la vitesse et du déplacement;
- ✓ déplacement rapide;
- ✓ Efficacité énergétique ;
- ✓ pas de risque pollution;

##### Inconvénients

- ✓ En version standard, nécessite une salle de machinerie;
- ✓ Exigence très importante sur l'entretien et le maintien de l'appareil ;

#### **I.4.3 Critères de 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.

**Energétiques :** 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.

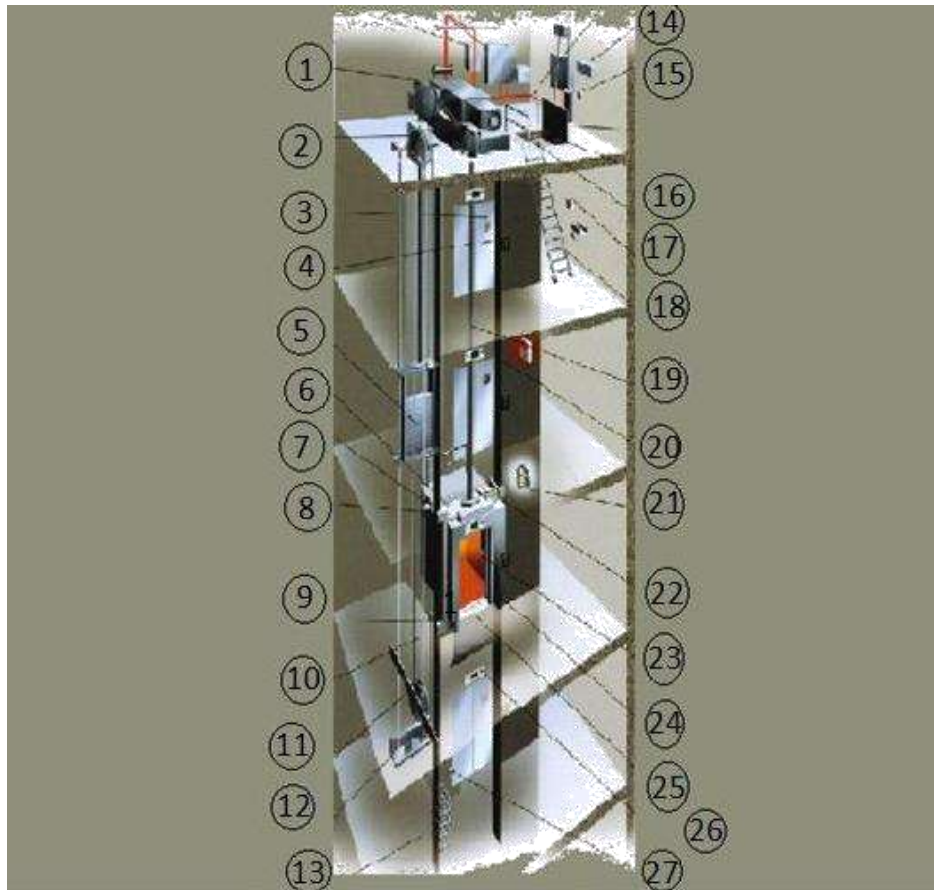


**Organisationnels** : comme le type de fonction de bâtiment, son occupation et son type de fonctionnement en garantissant une performance de confort et de trafic [2].

Dans ce qui suit nous nous intéressons aux ascenseurs à traction électrique.

#### I.4.4 Différentes parties d'un ascenseur à traction :

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



*Figure I. 7 : les différentes parties d'un ascenseur à traction*

Les composants d'un ascenseur à traction, représente la majorité des ascenseurs qui sont présente sur le marché.

- 1. Machine de traction** : Assurant le mouvement et l'arrêt de l'ascenseur. [12]
- 2. Limiteur de vitesse** : pour de raison de sécurité, Si la vitesse dépasse anormalement la vitesse maximale autorisée, les masselottes se lèvent et coupent un contact afin d'assurer l'arrêt de l'appareil. [12]
- 3. Boutons d'appel** : Comme son nom l'indique les boutons installés aux paliers Ce sont les boutons qui commandent l'arrivée de la cabine. [12]
- 4. Contrepoids** : Elément destiné à contre balancer le poids de la suspension cabine augmenté de la moitié de la charge utile. [12]
- 5. Étrier** : Ossature métallique portant la cabine ou le contrepoids, attelée aux organes de suspension. Cette ossature peut faire partie intégrante de la cabine elle-même. [19]

- 6. Parachute (montée et descente) :** Organe mécanique placé sur la suspension de cabine et 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. [12]
- 7. Opérateur de porte (mécanisme d'ouverture et de fermeture des portes) :** Porte à fermeture généralement automatique destinée à confiner l'utilisateur dans la cabine pendant le déplacement de celle-ci, lui interdisant tout contact avec les parties extérieures à la cabine. [16]
- 8. Coulisseaux de cabine :** Eléments fixés à la suspension, garnis d'une fourrure épousant la forme des guides et destinés à guider celle-ci dans la gaine. [16]
- 9. Câble de sélecteur d'étage :** Généralement, les sélecteurs d'étage mécaniques sont entraînés par le treuil ou le limiteur de vitesse. Cependant, certains sélecteurs d'étage ont leur propre câble d'entraînement. Celui-ci, relié entre la cabine d'ascenseur et le contrepoids entraîne un petit tambour qui actionne le sélecteur d'étage. [16]
- 10. Guide de contrepoids :** Profilés en acier, généralement en forme de T, destinés à guider la cabine et le contrepoids dans la gaine. [16]
- 11. Amortisseurs :** Ressorts puissants placés en cuvette et destinés à ralentir la suspension cabine ou le contrepoids en cas de dépassement des "fin de course" de sécurité. Dans le cas d'un ascenseur à grande vitesse, on utilise des amortisseurs à huile. [16]
- 12. Cuvette :** Partie la plus basse de la gaine de l'ascenseur contenant les poulies de renvoi et les amortisseurs. [16]
- 13. Eclairage de secours :** éclairage efficace de tous les locaux contenant machines et poulies. [17]
- 14. Frappe d'accès équilibrée.**
- 15. Balustrade :** système de fixation sur la cabine est assuré par deux pieds.
- 16. Boitier à clés :** permettant de garder à disposition les clés des locaux techniques.
- 17. Escalier ou échelles d'accès :** Pour faciliter l'accès aux fosses d'ascenseur.
- 18. Câble de suspension :** L'espace libre entre la cabine et le sol de la fosse doit être assez grand et entièrement disponible sur toute la hauteur de la boucle du câble. Les câbles doivent être suspendus sous la cabine en conservant leur courbe naturelle.
- 19. Guides de cabine :** Rails en acier en forme de T sur lesquels coulisse la cabine. [12]
- 20. Coffret de télésurveillance et de télé-arme :** Coffret principal d'alimentation se trouvant en machinerie. Avant toute intervention en machinerie, il va de la sécurité de l'utilisateur de déclencher le levier de ce coffret pour couper l'alimentation de l'appareil. [16]
- 21. Boite d'inspection :** Le boîtier d'inspection est un boîtier électrique situé sur le toit de la cabine. Son usage est réservé aux intervenants. Il permet de prendre le contrôle de

l'appareil et de se déplacer (avec la cabine), en étant sur le dessus de la dite cabine. Son bon fonctionnement doit être testé avant de monter sur la cabine ; chose qui, si elle n'est pas faite, peut générer un accident pouvant aller jusqu'à la mort ! [16]

**22. Boîte à boutons en cabine :** boutons d'envois sont installés dans la cabine.

**23. Barrière infrarouge :** Une protection fiable et sûre de la porte de l'ascenseur.

**24. Garde pieds :** C'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. [16]

**25. 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.

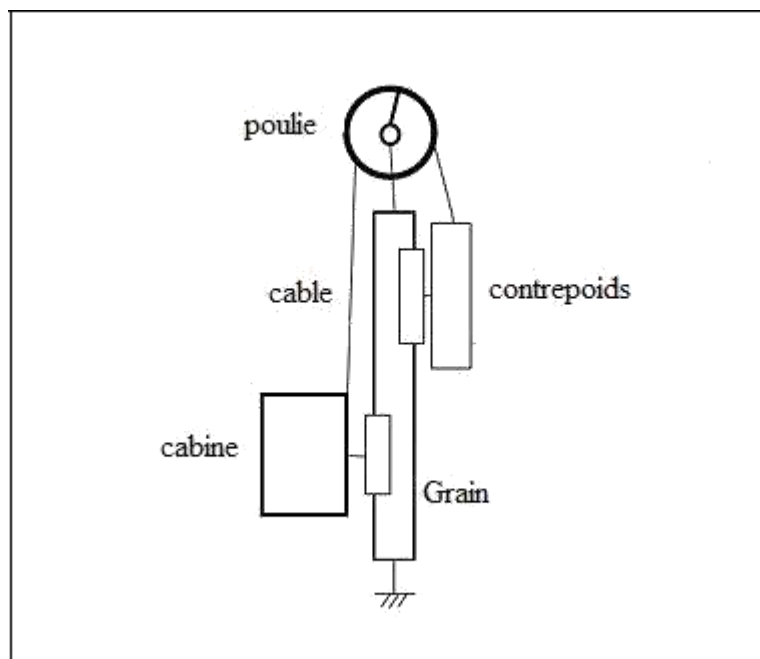
L'arrivée d'une cabine est signalée par un petit retentissement de sonnerie.

**26. Interrupteur d'arrêt en cuvette :** Dispositif qui permet l'arrêt de la cabine d'ascenseur à chaque niveau. [18]

## 1.5 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.



*Figure I. 8 : Principe de fonctionnement ascenseur à traction.*

# **Chapitre 02**

## **Etude de la carte Arduino et logiciel de programmation**

## II.1. Introduction

Le projet Arduino (2005) est issu d'une équipe d'enseignants et d'étudiants de l'école de Design d'Interaction d'Ivrea (Italie) suite au problème rencontré à cette période-là : Les outils nécessaires à la création de projets étaient complexes et trop chers ce qui rendait difficile le développement des projets et ralentissait l'apprentissage des étudiants. Pour cela ils se décident pour réaliser un matériel moins cher et plus facile à l'utilisation. Ils souhaitent créer un environnement proche de Processing (langage de programmation développé en 2001 par " Benjamin Fry" et " Casey Reas") et de la carte Wiring (un langage de programmation) facile d'accès et adaptée aux développements de projets de designers créée par Hernando Barragan en 2001 pour sa thèse de fin d'étude ) pour arriver à un dispositif simple à utiliser, dont les coûts seraient peu élevés, les codes et les plans sont libres ( les sources peuvent être modifiés, améliorés ou distribués par les utilisateurs) et multi-plates-formes (indépendant du système d'exploitation utilisé) .

## II.2 La carte Arduino :



*Figure II. 1: Carte Arduino*

### II 2.1 Définition :

Arduino est une plate-forme de prototypage sous la forme d'une carte électronique équipée d'un microcontrôleur qui interagit avec un milieu extérieur grâce à des capteurs

La carte électronique et un environnement de programmation interactifs à usage créatif constitués d'un environnement matériel et logiciel et qui 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 (accessibles et libres de droits) la grande communauté d'utilisateurs et de concepteurs permet à chacun de trouver les réponses ces questions et apporte un travail énorme de documentation au projet.

Les projets Arduino peuvent être autonomes, comme ils peuvent communiquer avec d'autres logiciels installés sur l'ordinateur tel que Flash, Processing ou MaxMSP, Matlab). Ces cartes

sont faites à base d'une interface entrée/sortie simple et d'un environnement de développement proche du langage

### **II.2.2. Les gammes de la carte Arduino :**

Actuellement, il existe plus de 20 versions de module Arduino, nous citons quelques une afin d'éclaircir l'évaluation de ce produit scientifique et académique:

- Le NG d'Arduino, avec une interface d'USB pour programmer et usage d'un ATmega8.
- L'Arduino Mini, une version miniature de l'Arduino en utilisant un microcontrôleur ATmega168.
- L'Arduino Nano, une petite carte programmable à travers le port USB ; cette version utilise un microcontrôleur ATmega168 (ATmega328 pour une nouvelle version).
- Le LilyPad Arduino, une conception de minimaliste pour l'application wearable (que l'on peut porter) en utilisant un microcontrôleur ATmega168
- Le NG d'Arduino plus, avec une interface d'USB pour programmer et usage d'un ATmega168.
- L'Arduino Bluetooth, avec une interface de Bluetooth pour programmer en utilisant un microcontrôleur ATmega168.
- L'Arduino Diecimila, avec une interface d'USB et utilise un microcontrôleur ATmega168.
- L'Arduino Duemilanove ("2009"), en utilisant un microcontrôleur l'ATmega168 (ATmega328 pour une nouvelle version) et actionné par l'intermédiaire de la puissance d'USB/DC.
- L'Arduino Mega, en utilisant un microcontrôleur ATmega1280 pour I/O additionnel et mémoire.
- L'Arduino UNO, utilisations microcontrôleur ATmega328.
- L'Arduino Mega2560, utilisations un microcontrôleur ATmega2560. Elle incorpore également le nouvel ATmega8U2.
- L'Arduino Leonardo, avec un morceau ATmega3U4 qui élimine le besoin de raccordement d'USB et peut être employé comme clavier.
- L'Arduino Esplora, ressemblant à un contrôleur visuel de jeu, avec un manche et des sondes intégrées pour le bruit, la lumière, la température, et l'accélération

Cartes Arduino	UNO R3 (classique & CMS)	UNO R3 Ethernet (classique & POE)	Leonardo	Mega 2560	Mega ADK	DUE	Esplora	Mini	Nano	Yun (classique & POE)	Zero PRO
Microcontrôleur	ATmega328P	ATmega328P	ATmega32u4	ATmega2560	ATmega2560	AT91SAM3X8E	ATmega32u4	ATmega328P	ATmega328P	ATmega32u4	ATSAM021G18
Cadencement Horloge	16 MHz	16 MHz	16 MHz	16 MHz	16 MHz	84 MHz	16 MHz	16 MHz	16 MHz	16 MHz	48 MHz
Tension d'entrée	7-12V	7-12V	7-12V	7-12V	7-12V	7-12V	7-12V	7-9V	7-9V	5V	5V
Tension de fonctionnement	5V	5V	5V	5V	5V	3.3V	5V	5V	5V	5V	3.3V
Entrée/Sortie Numérique	14/6	14/4	20/7	54/15	54/15	54/12	ⓧ	14/6	14/6	20/7	14/12
Entrée-Sortie (PWM) Analogique	6/0	6/0	12/0	16/0	16/0	12/2 (DAC)	ⓧ	8/0	8/0	12/0	6/1 (DAC)
Mémoire vive (Flash)	32 Ko	32 Ko	32 Ko	256 Ko	256 Ko	512 Ko	32 Ko	32 Ko	32 Ko	32 Ko	256 Ko
Mémoire vive (SRAM)	2 Ko	2 Ko	2,5 Ko	8 Ko	8 Ko	96 Ko	2,5 Ko	2 Ko	2 Ko	2,5 Ko	32 Ko
Mémoire morte (EEPROM)	1 Ko	1 Ko	1 Ko	4 Ko	4 Ko	ⓧ	1 Ko	1 Ko	1 Ko	1 Ko	16 Ko
Interface USB	USB-B mâle	USB-B mâle	Micro-USB	USB-B mâle	USB-B mâle & USB-A pour Android	2 ports micro- USB (Native et programming)	Micro-USB	ⓧ	Mini-USB	Micro-USB	2 ports micro- USB (Native et programming)
Port UART	1	1	1	4	4	4	ⓧ	ⓧ	1	1	2
Carte SD	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ
Ethernet	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ
Wi-Fi	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ	ⓧ
Dimensions	68x53mm	68x53mm	68x53mm	101x53mm	101x53mm	101x53mm	165x60mm	30x18mm	45x18mm	68x53mm	68x53mm

*Tableau II.1 : Tableau comparatif de différentes cartes Arduino*

### II.2.3. Architecture et composants de la carte :

L'Arduino est composée de deux parties indissociables : La carte qui est la partie hardware avec laquelle on travaille en construisant chaque projet et l'environnement de développement (IDE ARDUINO) qui est la partie logicielle assurée par l'ordinateur celle-ci permet de mettre au point et de transférer le programme qui sera par la suite exécuté par la carte ARDUINO

Dans ce qui suit nous allons présenter les composants de la carte Arduino UNO utilisé dans notre thème de recherche



#### **Le microcontrôleur:**

C'est le cerveau de la carte, il s'occupe de tout ce qui est exécution calculs gestion de flux de données, instructions du programme et gestion des ports d'entrées/sorties.

Il dispose d'une mémoire vive d'une capacité de 1Ko, et de 32 Ko de mémoire volatile pour stocker les Programmes.

Pour les besoins de l'Arduino le microcontrôleur devait être performant et de petit prix, les développeurs ont donc opté pour un circuit ATmega328 qui regroupe ces deux conditions.

Il est associé à des ports entrées/sorties qui permettent à l'utilisateur de communiquer et d'interagir avec l'environnement extérieure.





*Figure II. 2 : Le microcontrôleur de la carte*

**Arduino** ➤ **Les 14 entrées / sorties numériques (pattes 0\_13) :**

Ces dernières peuvent être configurées en entrée ou en sortie en les indiquant dans le programme.

Côté entrées, grâce à des capteurs qui rassemblent et collectent des informations de l'environnement, le contact via un bouton-poussoir.

Côté sorties, des actionneurs qui agissent sur le monde physique telle voyant lumineux, un moteur qui actionne une plateforme ; rampe ; cabine d'ascenseur...etc.

Les broches peuvent délivrer ou recevoir un maximum de 40mA d'intensité.

D'autres broches ont des fonctionnalités spéciales :

- Les broches 0 et 1: Utilisées pour la réception (RX) et la transmission (TX) des données séries, Ces broches sont connectées aux broches correspondantes du circuit intégré ATmega programmé en convertisseur USB-vers-série de la carte (Le FTDI), qui assure l'interface entre le microcontrôleur et le port USB de l'ordinateur.
- Broches 2 et 3 : Ces broches peuvent faire l'objet pour un déclenchement d'interruption interruption sur une valeur basse, sur un front montant ou descendant ou sur une variation de valeur ; cette instruction se présente sous la forme `attachInterrupt ()`.
- Broches 3, 5, 6, 9, 10, et 11 : L'impulsion PWM (Modulation à Largeur d'Impulsion) est générer par ces broches à l'aide de l'instruction `analogWrite ()`.
- Broches 10, 11, 12, 13 : la communication SPI (Interface Série Périphérique) est supporté par ces broches, disponible avec la librairie pour communication SPI.

- Broches 4 et 5 : les communications de protocole I2C est gérer par ces deux broches en général.
- Broche 13 : Connecter à une LED dans la carte, lorsque la broche est au niveau HAUT, la LED s'allumée, lorsque la broche est au niveau BAS, la LED est éteinte.



*Figure II. 3 : Les 14 entrées sorties numériques.*

➤ **Les 6 entrées analogiques (pattes A0\_A5) :**

Les 6 pattes sont réservées à la réception et mesure des signaux analogiques et offre la possibilité de la conversion de tension analogique ( $V_e$ ) de 0 à 5 V en une valeur numérique).



*Figure II. 4: Les 6 entrées sorties analogiques.*

➤ **Alimentation de la carte :**

La carte Arduino peut-être alimentée via un port USB qui l'alimente par +5V.

La norme USB limite à 500mA maximum l'alimentation de la carte, cela suffit afin d'alimenter des LED mais est généralement insuffisante pour actionner des moteurs ou des servomoteurs (avec cette alimentation seule, lors de la mise en route d'un moteur, cela peut entrainer une chute de tension).

On peut aussi faire fonctionner la carte avec une alimentation externe comprise entre 6 à 20 volts, cependant, si la carte est alimentée avec moins de 7V la broche de 5V pourrait fournir moins de 5V et la carte risque d'être instable.

L'utilisation d'une alimentation supérieure à 12Vest déconseillé, le régulateur de tension de la carte pourrait surchauffer et endommager la carte, la plage idéale recommandée pour alimenter la carte Arduino est entre 7V et 12V.

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

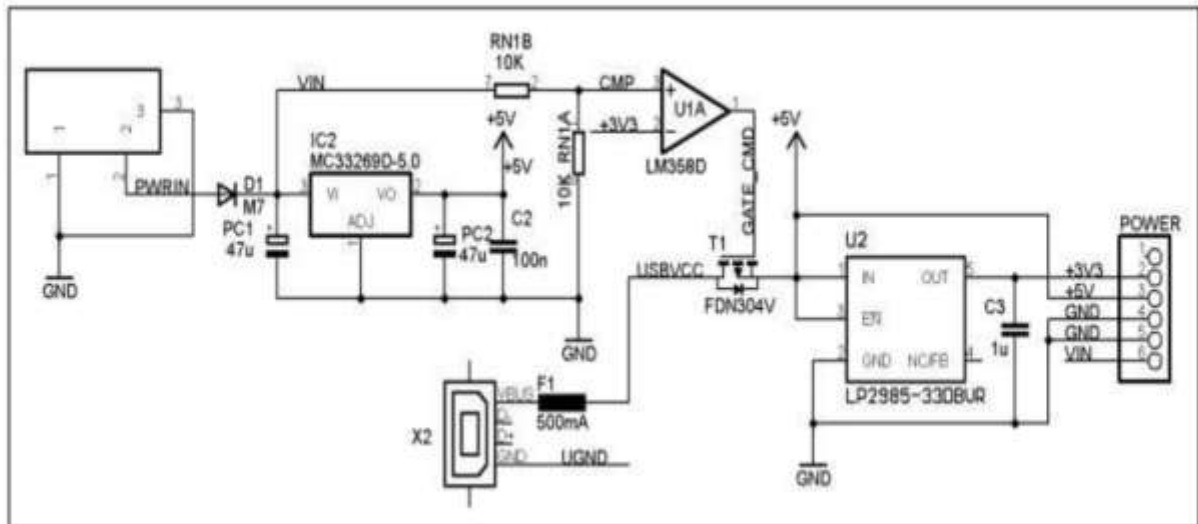


Figure II. 5: Le schéma électrique de l'alimentation de la carte Arduino.



### Les LED:

- LED verte d'alimentation;
- LED jaune ou verte active lors de transmission ou l'envoi du programme vers la carte (LED de transmission ; réception).
- LED jaune de test.

**Reset** la réinitialisation du microcontrôleur (remise à zéro)

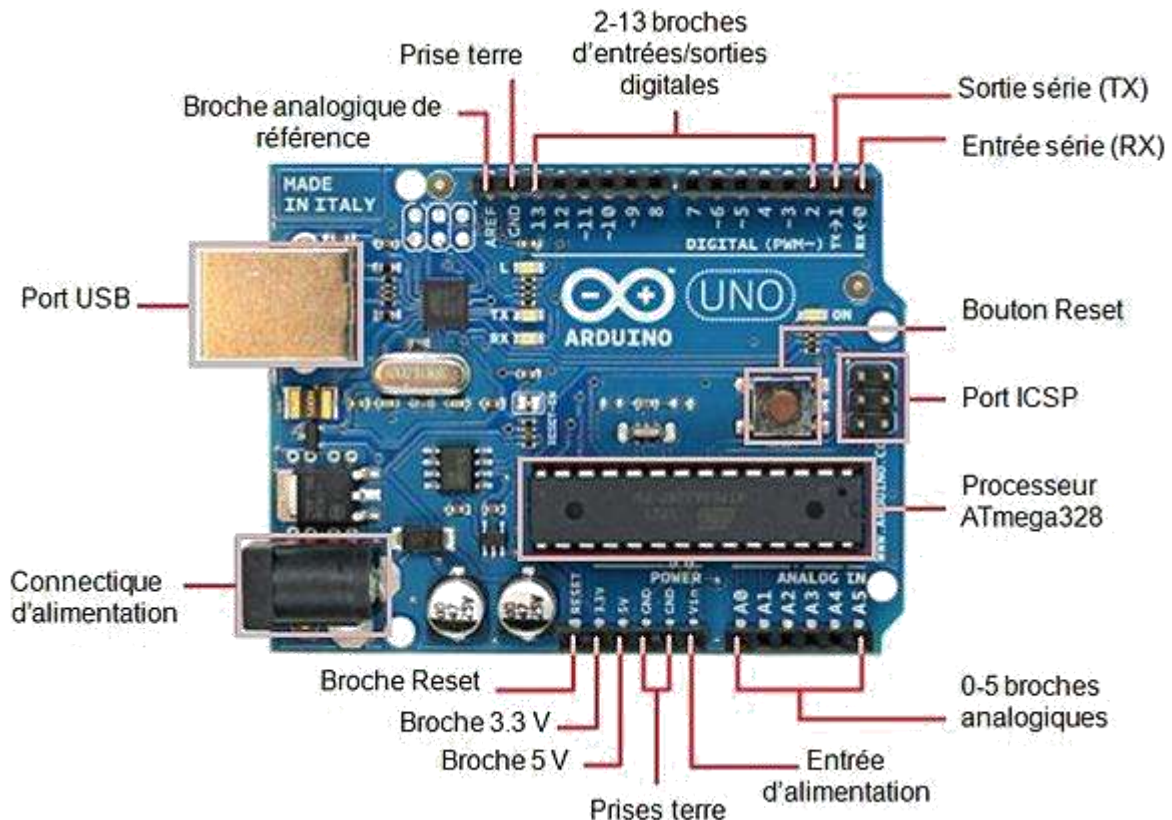


Figure II. 6: Les composants de la carte Arduino (Uno).

**Nb :** Sachant qu'un signal numérique peut se présenter sous deux états différents un signal analogique par contre peut prendre une infinité de valeurs en tension qui varient entre 0V et 5V.

La carte Arduino fonctionne en numérique, le microcontrôleur ne comprend que des « 0 » et des « 1 ». Pour cela les entrées A0 à A5 sont dotées de convertisseurs analogiques/numériques qui convertissent une tension en un entier variant de 0 à 1023.

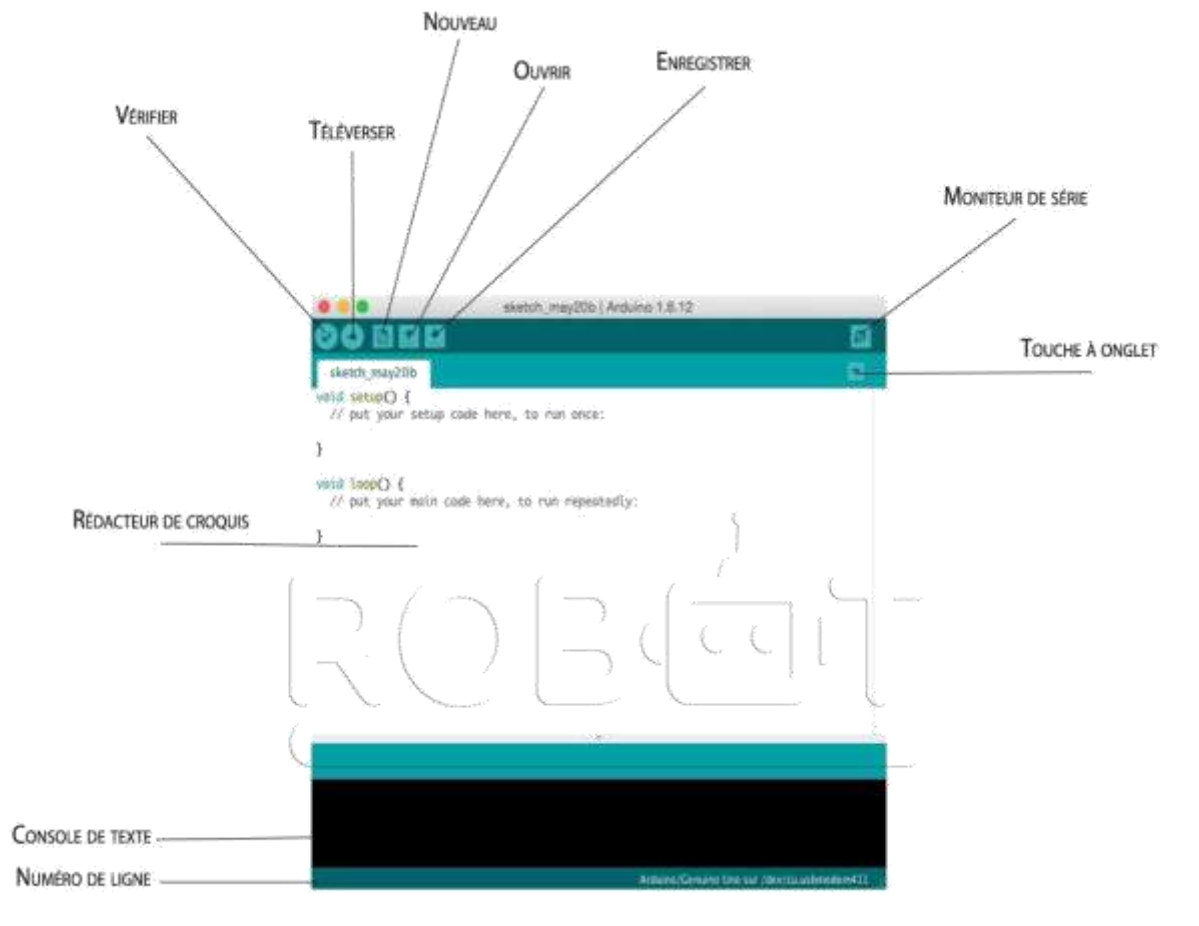
### II.3. Le logiciel Arduino (IDE Arduino)

L'environnement de programmation Arduino le plus reconnu développé et géré par la maison créatrice se nomme (IDE en anglais) est une application écrite en Java inspirée du langage Processing.

L'IDE permet la création, la modification et la gestion du programme offre aussi la possibilité de convertir en une succession d'instructions compréhensibles pour la carte.

C'est un logiciel de programmation par code, qui contient une cinquantaine de commandes différentes. A l'ouverture, l'interface visuelle du logiciel ressemble à ceci :

Des icônes de commande en haut, une page blanche vierge, une bande noire en bas.



*Figure II. 7: Les différentes parties de la fenêtre principale du logiciel Arduino.*

**Vérifier** : Avant que votre programme "code" puisse être envoyé à la carte, Il a besoin d'être vérifié afin de s'assurer qu'il ne comporte pas d'erreur avant d'être téléversé (transmis) sur la carte Arduino.

**Arrêt** : Cela arrête le processus de compilation. Il apparaît sur les plus vieilles versions de l'IDE

**Nouveau** : Cela ouvre une nouvelle fenêtre pour créer le nouveau croquis ou sketch (programme)

**Ouvrir** : Cela charge un croquis (sketch ou programme) à partir d'un fichier (extension .ino) sur votre ordinateur. Ce bouton vous permet aussi d'accéder à tous les exemples de programmes accessibles par défaut sur l'IDE

**Enregistrer** : Ceci sauve les changements du croquis sur lequel vous travaillez.

**Téléverser** : Cela compile et transmet le croquis via le câble USB à votre carte Arduino.

**Moniteur série** : le moniteur série est un outil indispensable pour les tests de programmes.

Il est utile pour avoir un retour de données.

**Rédacteur de croquis** - c'est là où on écrit ou en édite des croquis

**Touche à onglet** : Cela vous permet de créer plusieurs fichiers dans votre croquis. Cela concerne des programmes plus avancés

**Console de texte** : Cela vous montre ce que l'IDE fait actuellement et Également où les messages d'erreur s'affichent si vous faites une erreur typique de programme (Souvent appelée erreur de syntaxe).

**Numéro de ligne** : cela vous indique le numéro de ligne sur lequel votre curseur est activé. C'est utile puisque le compilateur donne des messages d'erreur avec un numéro de ligne.

**Rb** : suite à la découverte des différentes fonctionnalités de L'IDE Arduino la réalisation de programme devrait se faire sous la forme suivante :

- ✓ On conçoit d'abord un programme avec le logiciel Arduino ;
- ✓ On vérifie ce programme avec le logiciel (compilation) ;
- ✓ Des messages d'erreur apparaissent éventuellement... On corrige puis on vérifie à nouveau ;
- ✓ On retire le programme précédent sur la carte Arduino (cette opération se fait automatiquement lors de la télétransmission (l'upload) du nouveau programme ;
- ✓ On télécharge (upload) ce programme sur la carte Arduino ;
- ✓ L'exécution du programme sur la carte est automatique quelques secondes plus tard ou à ses prochains allumages

### II.3.1. Structure d'un programme

```
1  /* Ce programme fait clignoter une
   * led qui se situe dans la pin N°13
   */
2  int Led=13;           //Déclaration de la variable "Led" à utiliser
3  void setup ()
   {
   pinMode (Led,OUTPUT); // configure la broche N°13 comme une sortie
   }
4  void loop ()
   {
   digitalWrite(13,HIGH); // met la sortie N°13 à l'état haut (led allumée)
   delay(1000);           //attendre 1 seconde
   digitalWrite(13,LOW); // met la sortie N°13 à l'état bas (led éteinte)
   delay(3000);          // attendre 3 seconde
   }
```

**Figure II. 8: Exemple d'un programme**

**Arduino** Les parties du programme 1, 2, 3,4 sont expliqués ci-dessous :

#### **1 : Commentaires :**

Les commentaires sur le programme aident à la relecture du programme et sa compréhension par une autre personne pour cela on à 2 possibilités :

Soit en mettant les lignes du programme entre les signes `/** */`.

Soit sur une ligne de code en les séparant du code avec les signes `//`.

#### **2 : Définition et déclaration des variables:**

Pour notre exemple, on utilise une des sorties numériques de la carte qui est par exemple la 13<sup>ème</sup> sortie numérique. Cette variable doit être affectée dans cette partie ; On lui donne un nom arbitraire Led.

#### **3 : Configuration des entrées-sorties voidsetup ():**

Les broches numériques de l'Arduino font l'objet de configuration soit en entrées numériques ou en sorties numériques selon les besoins. Ici on va configurer Broche LED en sortie

PinMode (nom, état) est une des fonctions relatives aux entrées-sorties numériques.

#### 4 : Programmation des interactions voidloop ():

Dans cette boucle, on définit les opérations à effectuer, dans l'ordre:

- **digitalWrite (nom, état)** : Une des fonctions relatives aux entrées-sorties numériques.
- **delay (temps en millisecondes)** : La commande d'attente entre deux autres instructions
- : •chaque ligne d'instruction est terminée par un point-virgule.
- les accolades encadrent la boucle

#### II.3.2.Syntaxe du langage Arduino

Les différentes syntaxes des commandes d'Arduino sont données dans la table des matières suivante. Chaque instruction est suivie de sa traduction entre-parenthèses

Commandes de structure du programme :	
Structure générale	<ul style="list-style-type: none"><li>•Voidsetup() (configuration,préparation)</li><li>•Voidloop()(exécution)</li></ul>
Syntaxe Contrôle et conditions	<ul style="list-style-type: none"><li>• if(si..)</li><li>• if...else(si...sinon...)</li><li>• for (pour...)</li><li>• switch case(dans le cas où...)</li><li>• while(pendant que...)</li></ul>
Syntaxe Opérations de comparaison	<ul style="list-style-type: none"><li>• ==(équivalent à)</li><li>• !=(différent de)</li><li>• &lt;(inférieur à)</li><li>• &gt;(supérieur à)</li><li>• &lt;=(inférieur ou égal à)</li><li>• &gt;=(supérieur ou égal à)</li></ul>
Syntaxe Opérations booléennes	<ul style="list-style-type: none"><li>• &amp;&amp;(et)</li><li>•    (ou)</li><li>• !(et pas)</li></ul>
Autres commandes	<ul style="list-style-type: none"><li>• //(commentaires simple ligne)</li><li>• /**/(commentaries multi-lignes)</li><li>• #define(donner une valeur à un nom)</li></ul>

Tableau II.2:commandes du programme Arduino.

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

*Tableau II.3: Les différentes variables du programme Arduino.*

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

*Tableau II.4: Les variables du programme Arduino*



## **II.4. Conclusion**

Dans ce chapitre, l'étude s'est focalisée dans ce chapitre sur la présentation de la carte Arduino et son logiciel de programmation, nous avons commencé par quelques détails concernant la création et la définition de cette carte.

Puis nous avons parlé de son côté matérielle dont on a cité les gammes et les composants, enfin nous avons donné une présentation le logiciel de programmation.

La compréhension de données exposées dans ce chapitre nous offre des bases de connaissances pour l'usage de la carte de programmation.

# **Chapitre 03**

## **Conception et programmation**

### **III.1. Introduction**

Après avoir traité les généralités sur les ascenseurs et l'étude de la carte et l'environnement de programmation, respectivement dans le premier et deuxième chapitres, nous allons s'intéresser dans ce chapitre à la conception des différentes cartes (commande, puissance...) et les étapes de réalisation de notre maquette d'ascenseur suivant notre cahier des charges prédéterminé.

Afin de concevoir et de réaliser notre projet, nous suivrons une démarche comme suite.

Nos démarches porteront sur une étude théorique qui sera dédiée à la description des différents composants utilisés et de leurs caractéristiques; suivi d'une étude pratique orientée vers la conception des cartes par des logiciels de simulation tels que (ISIS, PROTEUS ; ARES).

Le but est d'établir une conception et un modèle adapté à notre projet.

#### **➤ Démarche de conception :**

Naturellement, il serait inconcevable de mener une conception sans avoir au préalable identifié et défini le besoin consigné dans le cahier des charges fourni par le client ou transcrit pour lui.

Dans ce cahier des charges nous y trouvons les spécifications techniques, fonctionnelles, budgétaires et temporelles.

Nous allons commencer par lister les données d'entrée que l'on dispose puis les données de sortie désirées, ensuite nous choisirons les fonctions à mettre en œuvre pour obtenir le résultat souhaité en faisant en sorte d'utiliser des technologies les plus récentes possible, de respecter les normes en vigueur, de penser au coût, au délai, à la maintenabilité, fiabilité, sécurité, au moyen de fabrication, au moyen de transport, aux accès.

En deuxième étape, il faudra faire des plans pour matérialiser les contraintes dimensionnelles.

En troisième étape pour ce qui est des schémas fonctionnels, écrire les lois physiques reliant les entrées et sorties pour obtenir un modèle que nous veillerons à identifier tous les paramètres. Pour ce qui est des schémas géométriques, suivant les sollicitations subies par les pièces, nous pouvons écrire les équations mécaniques et les dimensionner de façon à ce qu'elles supportent les efforts les plus défavorables en respectant également les normes en vigueur sur le choix de certains paramètres.

En quatrième étape, faire des simulations de ces modèles sur logiciel si possible avant de faire les plans d'ensemble, de fabrication, et la nomenclature complète de matériel.

➤ **Cahier des charges :**

Pour des besoins de télésurveillance, l'ascenseur doit pouvoir être commandé à distance. L'ascenseur desservira 4 étages. La porte cabine sera équipée d'un opérateur avec moteur bipolaire. Pour faciliter son utilisation par les personnes mal voyants et malentendants, il doit être doté d'une synthèse vocale, de boutons de commande émettant un bip lors d'un appui et à l'arrivée.

➤ **Matériels :** Maquette ascenseur, Logiciel de programmation Arduino,,

Logiciel step 7 siemens, Logiciel Matlab, Logiciel ISIS, ARES et un budget alloué au projet.

## **III.2. Etude théorique**

### **III.2.1. Partie mécanique**

Dans notre projet nous avons l'ascenseur didactique disponible.

Constituée d'une cabine en métal, de 230cm<sup>2</sup> de hauteur de 27cm, qui se déplace le long des guides métalliques de 140cm.

La cabine est dotée d'un capteur de fin de course mécanique pour l'indication d'étage.

Tout cela est entouré par une carcasse de 3 (trois) niveaux (niveau0, niveau1, niveau2).

### **III.2.2. Partie électronique pour l'ASCENSEUR**

#### **III.2.2.1. Le microcontrôleur (carte Arduino UNO R3)**

Le microcontrôleur de la carte Arduino UNO est un ATmega328. C'est un microcontrôleur ATMEL de la famille AVR 8bits. Les principales caractéristiques sont :

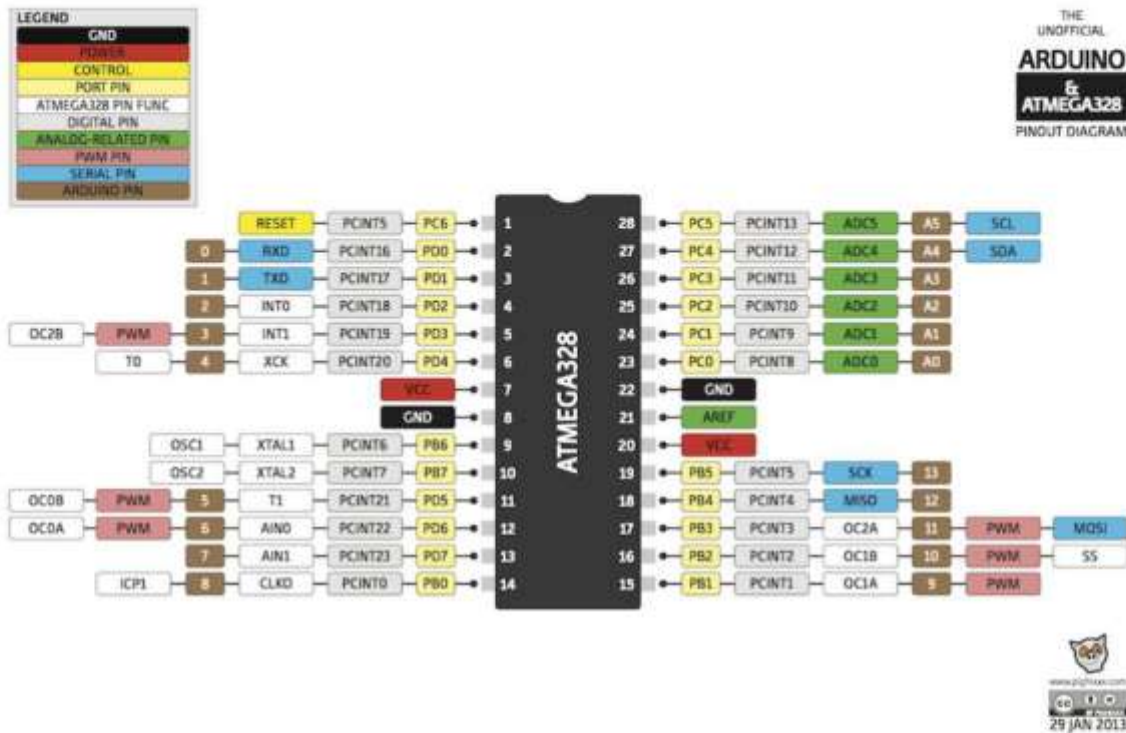
**FLASH** = mémoire programme de 32Ko

**SRAM** = données (volatiles) 2Ko

**EEPROM** = données (non volatiles) 1Ko

**Digital I/O (entrées-sorties Tout Ou Rien)** = 3 ports PortB, PortC, PortD (soit 23 broches en tout I/O)

**Timers/Counters** : Timer0 et Timer2 (comptage 8 bits), Timer1 (comptage 16bits) Chaque timer peut être utilisé pour générer deux signaux PWM. (6 broches OCxA/OCxB)



**Figure III. 1: Microcontrôleur Carte Arduino UNO**

Plusieurs broches multi-fonctions : certaines broches peuvent avoir plusieurs fonctions différentes, choisies par programmation. Elles ont alors plusieurs noms sur le brochage

Par exemple, les broches PB1, PB2, PB3, PD3, PD5, PD6 peuvent servir de sortie PWM (Pulse Width Modulation), c'est-à-dire des sorties qui joueront le rôle de sorties analogiques.

Elles correspondent aux broches des connecteurs 3, 5, 6, 9,10 et 11. Cet autre rôle possible est lié aux timers et ces broches sont alors appelées OCxA ou OcxB dans la documentation.

Ce sont les mêmes broches, mais pour une autre fonction. Si vous regardez à nouveau le brochage, vous constaterez que toutes les broches sont multi-fonctions.

**PWM** = 6 broches OC0A(PD6), OC0B(PD5), OC1A(PB1), OC1B(PB2), OC2A(PB3), OC2B(PD3)

Les broches du PORTC peuvent être converties par un convertisseur Analog toDigital.

**Analog to Digital Converter (résolution 10bits)** = 6 entrées multiplexées ADC0(PC0) à ADC5(PC5)

**Gestion bus I2C (TWI Two Wire Interface)** = le bus est exploité via les broches SDA(PC5)/SCL(PC4).

**Port série (USART)** = émission/réception série via les broches TXD(PD1)/RXD(PD0)

Comparateur Analogique = broches AIN0(PD6) et AIN1 (PD7) peut déclencher interruption

**Gestion d'interruptions (24 sources possibles (cf interrupt vectors)) : en résumé**

- Interruptions liées aux entrées INT0 (PD2) et INT1 (PD3)
- Interruptions sur changement d'état des broches PCINT0 à PCINT23
- Interruptions liées aux Timers 0, 1 et 2 (plusieurs causes configurables)
- Interruption liée au comparateur analogique
- Interruption de fin de conversion ADC
- Interruptions du port série USART
- Interruption du bus TWI (I2C)

Les UNO R3 sont conçues pour des projets moyennement complexes. Pour cela nous avons porté notre choix sur la carte Arduino Uno qui nous offrent toutes les fonctionnalités, nécessaires afin d'aboutir à la réalisation de notre projet .

**Caractéristiques principales de la carte Arduino UNO :**

- Alimentation:
  - via port USB ou
  - 7 à 12 V sur connecteur alim 5,5 x 2,1 mm
- Microprocesseur: ATmega328
- Mémoire flash: 32 kB
- Mémoire SRAM: 2 kB
- Mémoire EEPROM: 1 kB
- Interfaces:
  - 14 broches d'E/S dont 6 PWM
  - 6 entrées analogiques 10 bits
  - Bus série, I2C et SPI
- Intensité par E/S: 40 mA
- Cadencement: 16 MHz
- Gestion des interruptions
- Fiche USB B
- Version: Rev. 3

- Dimensions: 74 x 53 x 15 mm

### **III.2.2.2. Le moteur :**

La détermination et le choix d'un moteur doit se faire suivant le type d'application en terme de vitesse, couple, puissance, environnement, secteur d'alimentation, système de contrôle de commande.

Dans le cas nous concernant notre application est de type levage descente, l'objectif est de contrôler la vitesse de sortie, la position. De soulever une masse t sur une hauteur de 1 vitesse maximale prédéfini ou selon les standards.

Nous utilisons un convertisseur de puissance de type pont H piloté par un microcontrôleur pour commander le moteur.

Le moteur utilisé est à courant continu avec réducteur, reconnue pour sa haute résistance, sans bruits, petit mais qui offre une grande efficacité.

#### **Les caractéristiques du moteur :**

**Nom du produit :** SAYAMA GEARED MOTOR

**Modèle :** RB-35GM-E35-12

**Catégorie :** Moteur de transmission

**Courant de charge Max :** 2.1 A

**Charge de transmission Max :** 18Kg/cm (permanent 6Kg/cm)

**Tension de fonctionnement :** 12 V (DC )

**La vitesse de charge :** 9tr/min

**Arbre Ø :** 6mm

**Le couple maximal :** 180 N.cm

**longueur :** 97.5mm



*Figure III. 2:Le moteur de traction.*



### **Le pont en H :**

Le pont en H est une structure électronique servant : - à contrôler la polarité aux bornes d'un moteur à courant continu, c'est à dire à contrôler le sens de rotation du moteur et - à faire varier la vitesse du moteur en modulant la tension à ses bornes

Il est composé de quatre commutateurs (transistors, ...) généralement disposés en forme de H sur les schémas, d'où le nom.



### **Schéma de principe :**

Un pont en H est composé de 2 parties :

- une partie puissance composée de 4 interrupteurs commandés (relais, transistor bipolaire ou MOS, thyristor...)
- une partie commande qui permet de contrôler l'ouverture (O) ou la fermeture (F) des interrupteurs.

Cette commande se fait le plus souvent par l'intermédiaire d'un microcontrôleur afin de contrôler le pont en TOR (tout ou rien : tension dans la charge maximale ou nul) ou en PWM (pulse width modulation connue en francophonie sous l'abrégié (MLI : modulation en largeur d'impulsion)) : tension moyenne dans la charge réglable.



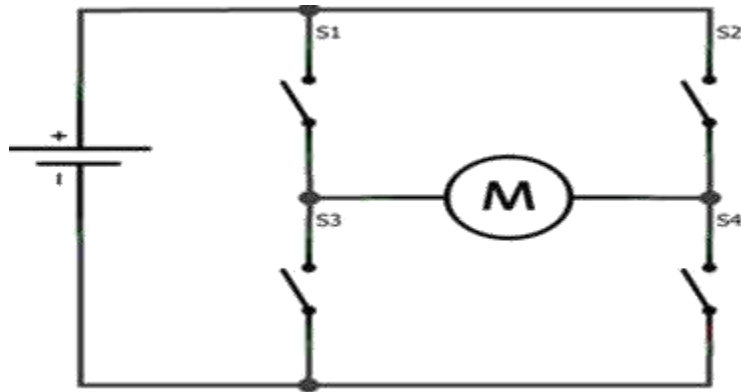


Figure III. 3 : Le pont en H.

➤ **Principe de fonctionnement :**

- Dans son principe de fonctionnement, le pont en H est un assemblage de 4 transistors (T1 à T4) montés de telle façon que le courant puisse circuler dans les deux sens au travers du moteur, ce qui permettra de changer son sens de rotation.
- La structure du pont permet de modifier le sens du courant dans le moteur en gérant correctement l'ouverture et la fermeture des commutateurs :
  - Si S1 et S4 sont fermés alors que S2 et S3 sont ouverts, le courant circulera de A vers B

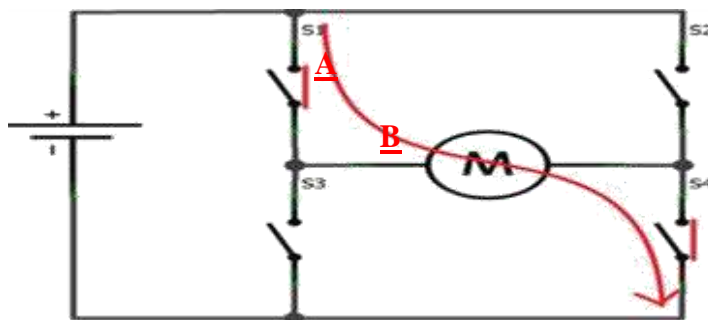


Figure III. 4: Fonctionnement dans le sens direct

- Si S2 et S3 sont fermés alors que S1 et S4 sont ouverts, le courant circulera de B vers A

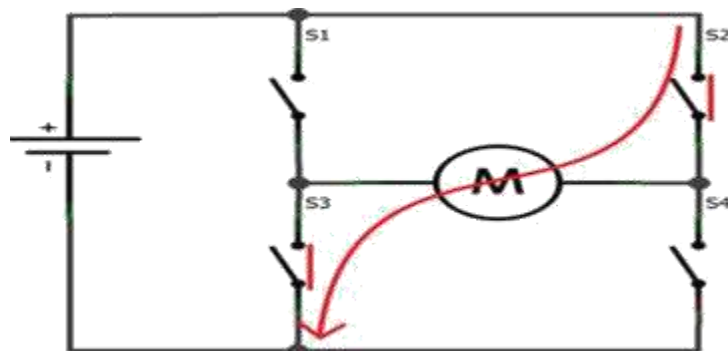


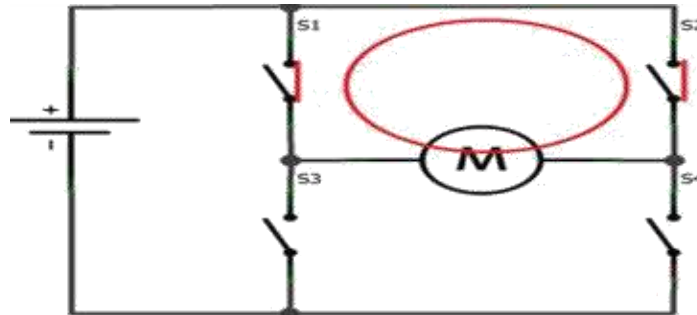
Figure III. 5 : Fonctionnement dans le sens indirect

On remarque, que les transistors fonctionnent en mode pair donc deux transistors à la fois.

En effet, on fermant juste un seul et laissant ouvert les trois autres le courant n'a nulle part où aller et rien ne se passe, le moteur est en roue libre.

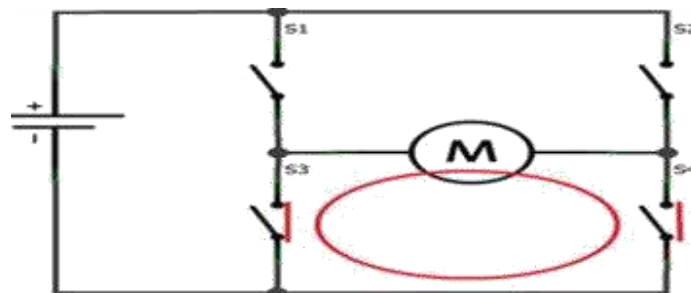
La fermeture de S1 et S2 en laissant S3 et S4 ouverts, va créer ce qu'on appelle un **frein magnétique**.

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. 6 : Freinage avec S1 et S2.*

Dans le cas contraire on obtient le même freinage mais avec S3 et S4



*Figure III. 7 : Freinage avec S3 et S4.*

**Attention** : S1 et S3 ne doivent jamais être commutés en même temps (tout comme S2 et S4) car cela provoquerait un court-circuit franc et destructeur.

D'où vient l'intérêt de protéger notre pont en H avec des diodes de roue libres (RLB)

Pour protéger l'ensemble des transistors des parasites générés lors du freinage électronique du moteur, nous plaçons des diodes en parallèle aux bornes des transistors, le courant trouvera toujours un moyen de passer sans risquer de forcer le passage dans les transistors en les grillant.

Suite au développement de la fabrication des composants électroniques et de leurs performances le **L293D** est un composant facilitant l'utilisation du principe pont en H toute en évitant l'utilisation de la logique des transistors et leurs comportements

### III.2.2.3. 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 se trouve facilement sur Ebay et ne coûte pas très cher.

Le **L293D** est un double pont-H, ce qui signifie qu'il est possible de l'utiliser pour commander quatre moteurs distincts (dans un seul sens) grâce à ses 4 canaux. Il est également possible de constituer deux pont-h afin de piloter deux moteurs distincts, dans les deux sens et indépendamment l'un de l'autre.

Ci-dessous dans le tableau la logique du L293D

#### Brochage du L293D avec moteur sous Arduino :

Avec un seul pont **L293D** et un **Arduino** on va être capable de piloter 2 moteurs à courant continu indépendamment l'un de l'autre

Cela nous offrirait la possibilité de piloter notre ascenseur

Broche	Nom	Fonction
1	Enable1	Permet d'envoyer (ou pas) la tension sur les sorties du moteur via OUTPUT1 et OUTPUT2. ENABLE1 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 appliquées sur INPUT1 & INPUT2
2	Input1	Avec Input 2, sont les broches de commande du Pont-H Output1/Output2. Se raccorde à Arduino, permet
3	Output1	Avec Output 2, seront les broches à raccorder à la charge (le moteur).

4/5	GND	Doit être raccorder à la masse (GND) de la source d'alimentation de puissance VS (ex: la borne négative de l'accumulateur +9.2v) et à la masse de la source d'alimentation de la logique "VSS" (donc GND Arduino). Si vous n'avez qu'une source d'alimentation pour le tout, c'est forcément plus simple.
7	Ouput2	Avec Output 1, seront les broches à raccorder à la charge (le moteur).
8	VS	Alimentation de puissance des moteurs
9	Enable2	activation du second pont-H constitué de Output3/Output4.
10	Input3	broches de commande du Pont-H Output3/Output4
11	Ouput3	sorties du second pont-H (Output3/Output4)
12	GND	
13	GND	
14	Ouput4	
15	Input4	A utiliser conjointement avec Input 3 pour commander le pont-H Output3/Output4.
16	VSS	Alimentation de la logique de commande (5V). A raccorder à la borne +5V d'Arduino .
6	Input 2	Avec Input 1, sont les broches de commande du Pont-H Output1/Output2. Se raccorde à Arduino, permet de commander le sens du courant entre Output 1 et Output 2.

**Tableau III. 1 : Brochage du L293D**

### III.2.3. Partie électronique pour la plateforme mobile :

Notre plateforme mobile située dans le planche de notre ascenseur sera à elle piloter par un autre type de driver moteur connue sous le nom de L298N le choix de ce dernier est fait pour piloter notre plateforme avec précision et pour cela nous utiliserons un moteur pas à pas

#### III.2.3.1 Le module L298N :

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

C'est un module extrêmement utile pour le contrôler de robots et ensembles mécanisés. Il peut contrôler deux moteur courant continu ou un moteur pas-à-pas 4 fils 2 phases.

Il est conçu pour supporter des tensions plus élevées, des courants importants tout en proposant une commande logique TTL (basse tension, courant faibles, idéal donc pour un microcontrôleur).

Il peut piloter des charges inductives comme des relais, moteurs continus et moteurs pas-à-pas.

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

Il s'agit d'un module prêt à l'emploi.

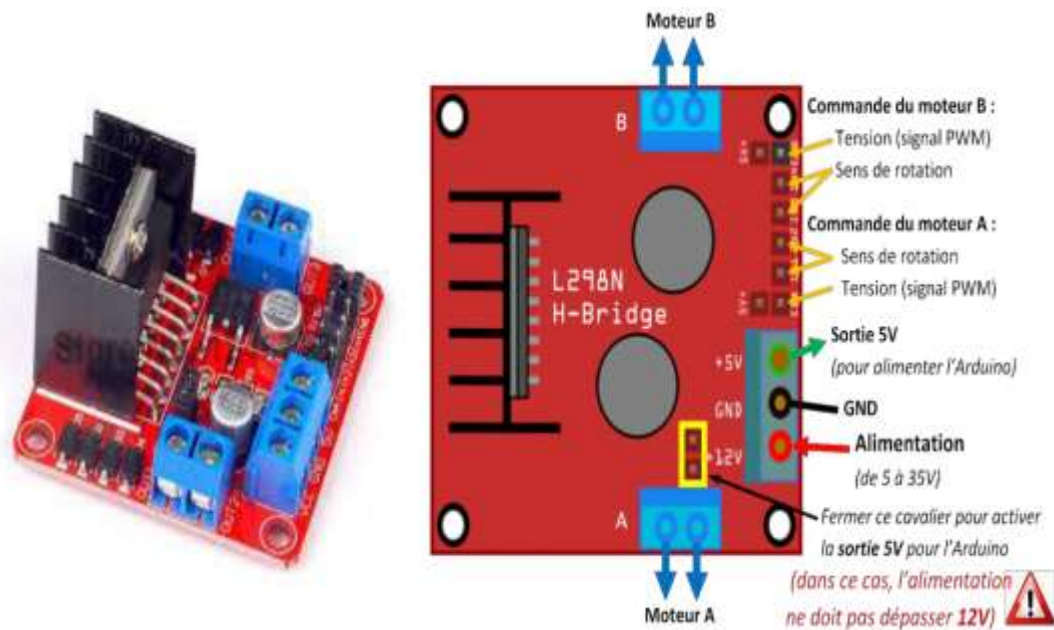


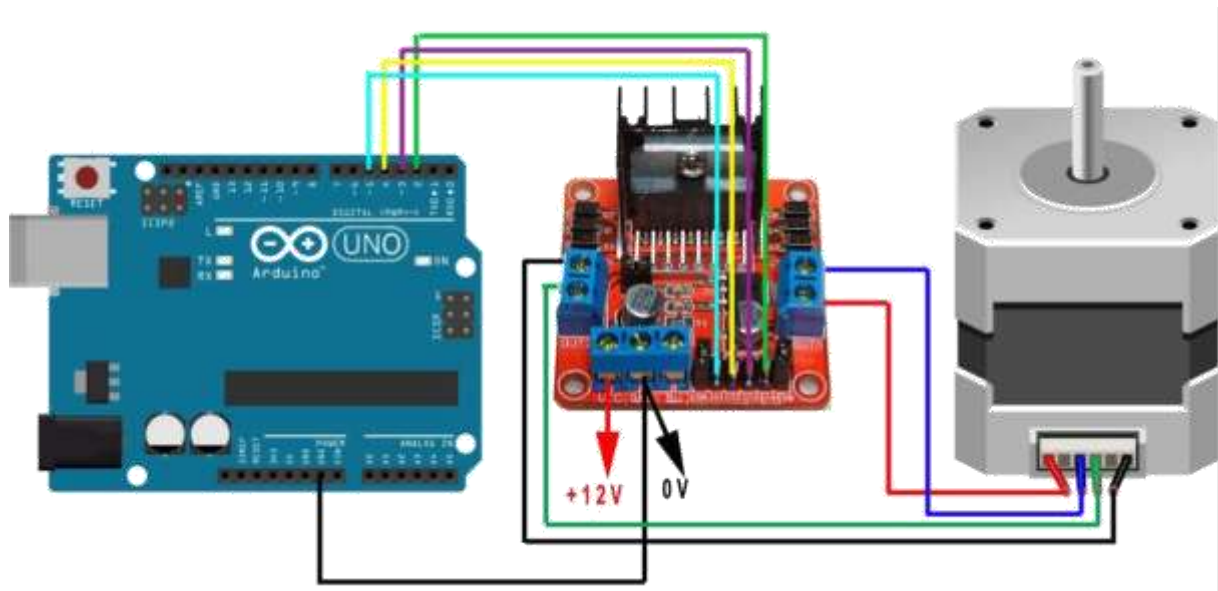
Figure III. 8: Descriptif du module L298N

### III.2.3.2. Le moteur :

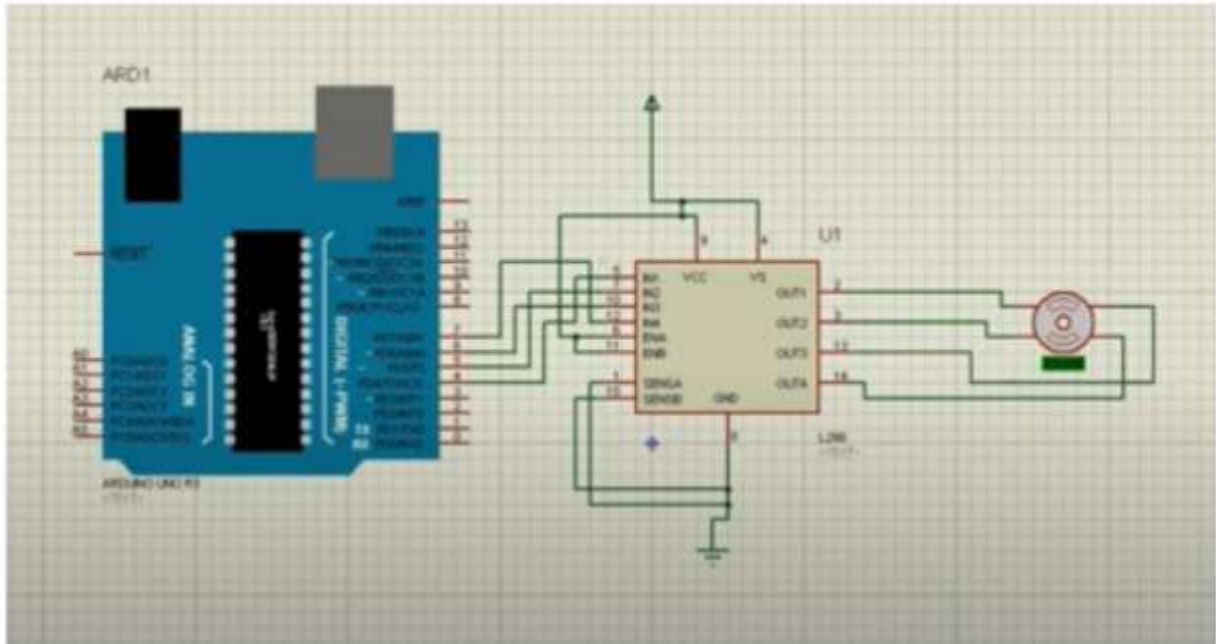
Moteur pas-à-pas bipolaire de taille réduite Nema 14 présentant un couple de maintien de 12,5 Ncm, une grande précision et des niveaux de vibrations et de bruits très faibles.

#### Les caractéristiques du moteur :

- Nombre de pas: 200
- Angle par pas: 1,8°
- Alimentation: 3,5 V
- Résistance/phase: 3,5  $\Omega$
- Courant/phase: 1 A
- Inductance/phase: 3,5 mH
- Couple de maintien: 1,25 kg.cm
- Connexion: 4 fils
- Dimensions de l'axe:  $\varnothing 5 \times 23$  mm
- Dimensions: 35 x 35 x 29 mm
- Poids: 120 g



*Figure III. 9 : Prise en 3d du câblage du moteur pas à pas*



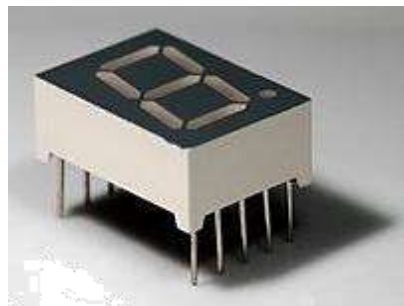
*Figure III. 10 : Prise du câblage du moteur pas à pas*

### III .2.3.3 Autre matériels :

#### III .2.3.3 1 Afficheur 7segment

##### Présentation de l’afficheur 7 segment :

Les afficheurs 7 segments sont un type d'afficheur particulièrement présent sur les calculatrices et les montres à affichage numérique : les caractères (des chiffres, quoique quelques lettres soient utilisées pour l'affichage hexadécimal) s'écrivent en allumant ou en éteignant des segments, au nombre de sept. Lorsque les 7 segments sont allumés, nous obtiendrons le chiffre 8.



*Figure III. 11 : Afficheur 7segment*

Dans un afficheur 7 segments, les segments sont le plus souvent désignés par les lettres allant de A à G. Dans le cas où l'afficheur comporte un point, servant de séparateur décimal, ce dernier

est désigné DP (de l'anglais décimal point) ; certains parlent dans ce cas d'un afficheur «8 segments ».

Dans le cas d'afficheurs à DEL, deux cas de figures sont présents :

- ✚ Afficheur à anode commune : l'ensemble des anodes sont reliées et connectées au potentiel haut. La commande du segment se fait par sa cathode mise au potentiel bas.
- ✚ Afficheur à cathode commune : l'ensemble des cathodes sont reliées et connectées au potentiel bas. La commande du segment se fait par son anode mise au potentiel haut.

### **III.3. Etude pratique**

#### **III.3.1. Partie logiciel :**

Avant de passer la réalisation de nos cartes et afin de s'assurer de leur bon fonctionnement nous ferons la conception et la simulation de ces dernières. Pour cela nous utilisons un logiciel très performant connue sous le nom de PROTEUS.

PROTEUS est un outil qui offre un large choix de fonctionnalités, permet de dessiner des schémas électroniques, de les simuler et de réaliser leur circuit imprimé correspondant.

##### **III.3.1.1.Présentation de Proteus**

Le logiciel est très connu dans le domaine de l'électronique. il est utilisé et reconnu par de nombreuses entreprises et organismes de formation.

Outre la performance et la popularité de l'outil, Proteus possède d'autres avantages :

Pack contenant des logiciels faciles à entreprendre et rapides à comprendre et à utiliser ;

Support technique performant ;

Outil qui permet la création de prototype virtuel permettant de réduire les coûts matériels et logiciels lors de la conception d'un projet et d'éviter des pertes matériels en cas d'échec.

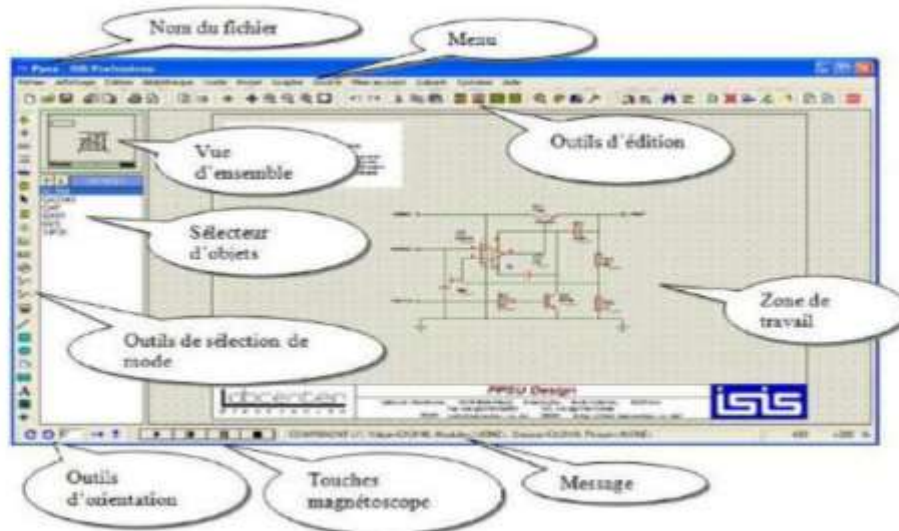
Le logiciel est constitué de deux principaux modules : **ISIS** et **ARES** .

##### **III.3.1.1.1. ISIS**

Le module ISIS de Proteus est conçu principalement pour la réalisation d'un schéma structurel d'un circuit électronique (assemblage de composants électroniques dont on fixe les valeurs et les références selon notre besoin) reliés par des connexions électriques (fils).

Par ailleurs, le logiciel permet également de simuler les différents types de montages ce qui permet de s'éviter certaines erreurs dans l'étape de conception.





*Figure III. 12: Fenêtre du module de simulation Proteus ISIS.*

### III.3.1.1.2. ARES

Le module ARES est un outil complémentaire pour le module ISIS chaque schéma électrique réalisé sur ISIS peut alors être importé facilement sur ARES pour réaliser sont PCB ( Printed Circuit Board ) de la carte électronique.

Bien que la création et l'édition d'un circuit imprimé soit plus correcte lorsqu'elle est réalisée manuellement, ce logiciel permet un gain de temps en plaçant automatiquement les composants et de réaliser le routage par lui d'une manière automatique.

### III.3.2. Partie matériels (conception des cartes)

Cette partie se base sur la conception de 3 (trois) cartes électroniques;

**Carte d'alimentation** : nous allons concevoir une carte d'alimentation adapté à notre besoin qui peut alimenter nos circuits correctement avec des tensions adapter aux caractéristiques de notre matériels Le moteur doit être alimenté par 12V; la carte Arduino par 9V; et la logique de commande se fait par 5V, nous avons utilisé des régulateurs de tension "LM7805" et "M7809"

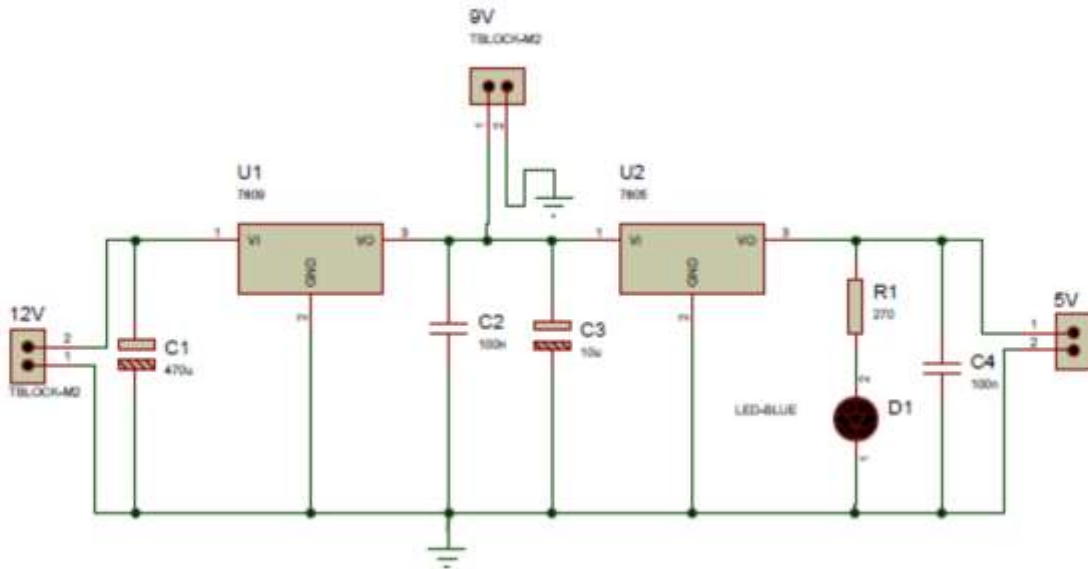


Figure III. 13: Schéma de conception de la carte d'alimentation sur ISIS.

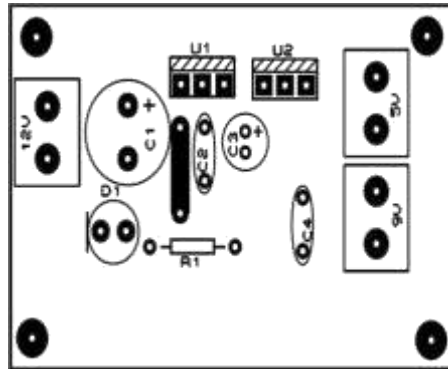


Figure III. 14 : Implantation des composants

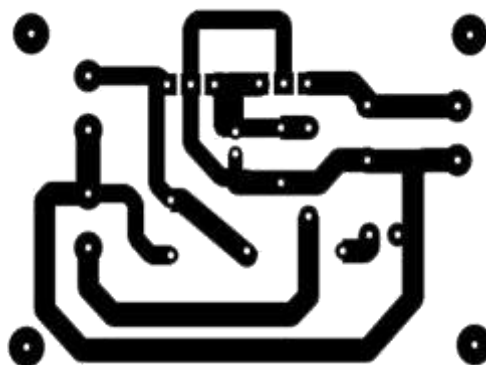
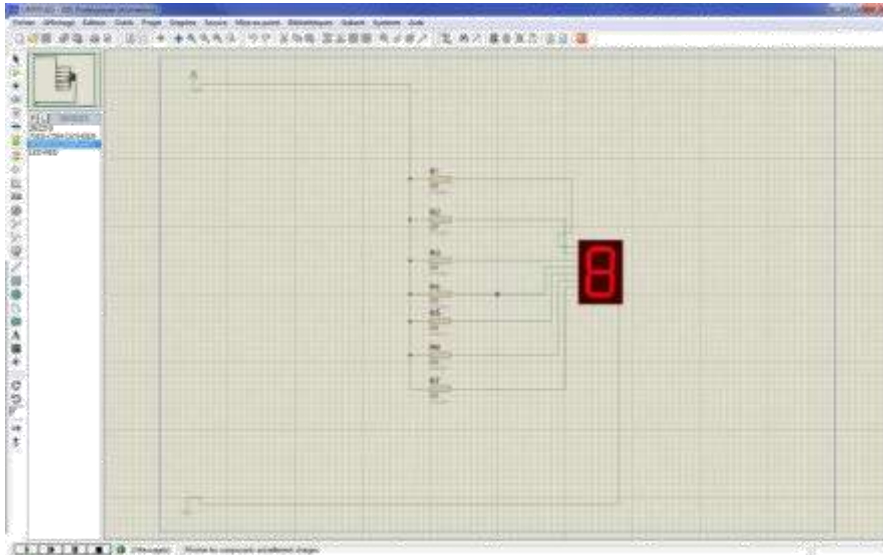


Figure III. 15 : PCB de la carte d'alimentation

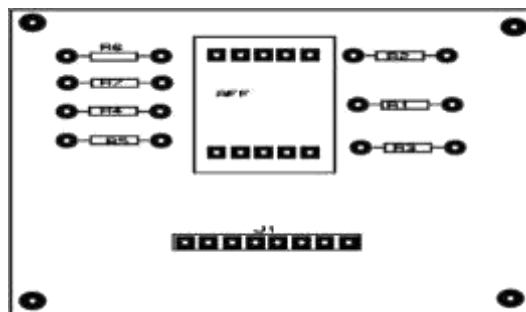
**Carte de puissance :** pour la commande du moteur, celle si seras remplacer par les deux modules L293D et le L298N

**Cartes d'afficheurs 7 segments :** une pour chaque niveau une résistance de protection doit être ajoutée en série avec chaque segment de l'afficheur.

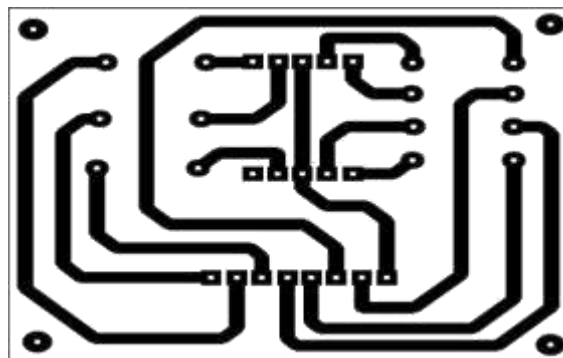
Une résistance de 330 Ohm suffit pour cela.



*Figure III. 16: Schéma de conception de la carte d'afficheur sur ISIS.*



*Figure III. 17: Implantation des composants*



*Figure III. 18 : Circuit imprimé de la carte d'afficheur.*

### Carte pour la commande des afficheurs :

Les trois afficheurs consomment plus qu'une patte d'arduino délivré pour remédier à ce problème l'utilisation de transistors intermédiaires semble être d'une grande efficacité (PNP BC 327) .

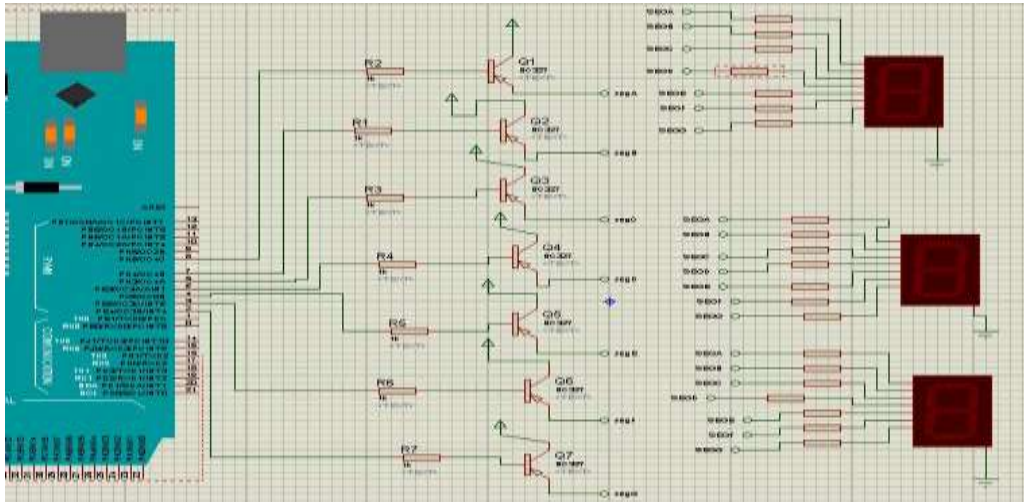


Figure III. 19: conception de schéma de la carte de commande des afficheurs sur ISIS

## III.4. Programmation

### III.4.1. Cahier des charges

Etant donné l'ascenseur initialement au repos, à n'importe quel étage, le système va s'initialiser à niveau "0" à chaque démarrage.

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

\_ Si l'appel vient de l'étage où se trouve l'ascenseur, un bip sonore et un clignotement de l'afficheur indique la présence de la cabine à l'étage demandé.

\_ S'il vient d'un autre étage, alors l'ascenseur réagit de selon sa position donc il va soit monter ou descendre vers l'étage demandé, l'afficheur indique le niveau atteint au même temps.

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

\_ S'il y a plusieurs appels, c'est le premier qui va être pris en considération.

\_ Si l'ascenseur doit monter ou descendre de plus d'un étage, alors à l'arrivée de chaque étage intermédiaire, un test de demande d'appel est effectué.

La cabine s'arrête pour laisser monter ou descendre les personnes et continuera de répondre à la première demande.

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

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

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

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

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

La plateforme sera utilisée en cas de besoins grâce à un bouton qui actionne le moteur qui commande la plateforme.

### III.4.2 Organigramme de déroulement du programme:

Programme principale :

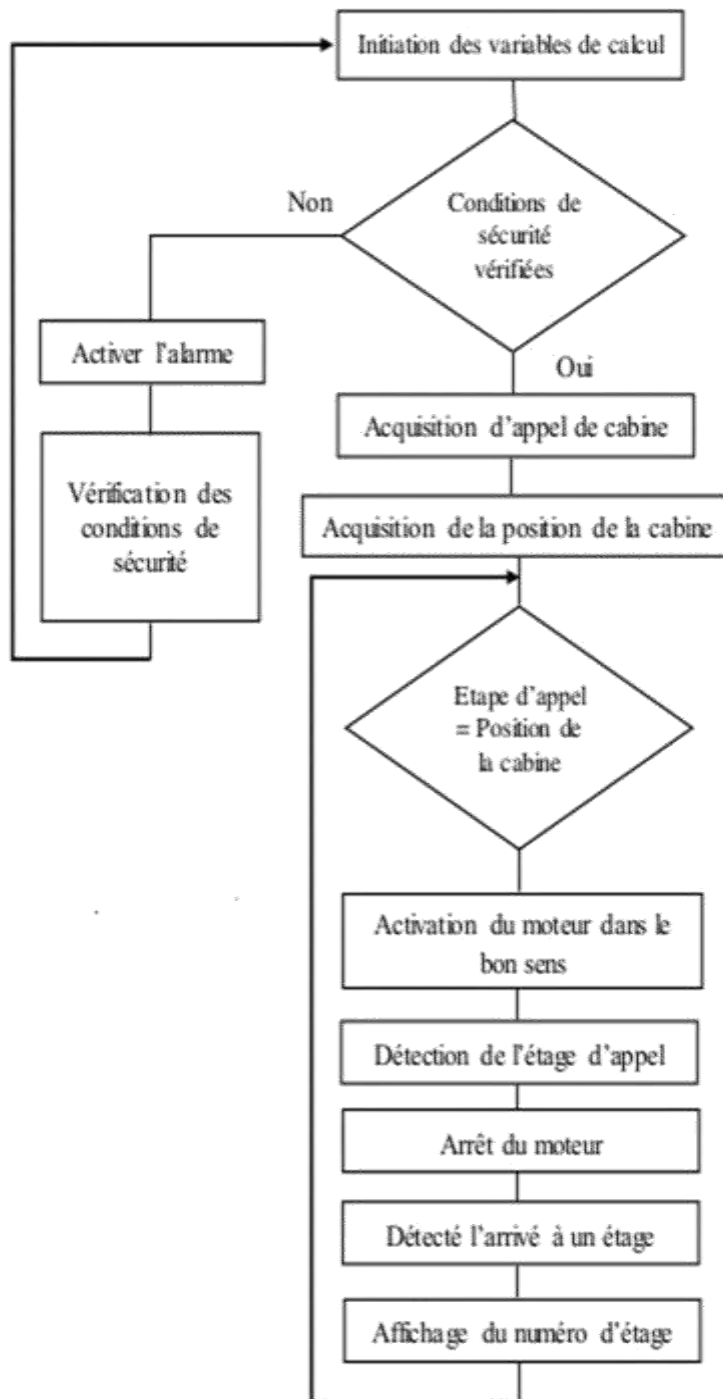


Figure III. 20: Organigramme de fonctionnement de l'ascenseur

### III.4.3 Réalisation du programme

Vue la complexité de notre programme et synchronisation de plusieurs paramètres qui se déroule en même temps nous l'avons partagé en un ensemble de sous-programme, qu'on peut appeler dans le programme principale.

#### III.4.3.1 Réalisation du programme de l'ascenseur via l'environnement de programmation IDE :

Les 3 premières broches sont configurées comme sortie vers le blindage du moteur

```
pinMode(12, SORTIE); // Direction
```

```
pinMode (3, OUTPUT); // PWM
```

```
pinMode (9, SORTIE); // Frein
```

Les trois suivants sont pour les boutons-poussoirs, définis comme pull-up

```
pinMode (4, INPUT_PULLUP);
```

```
pinMode (5, INPUT_PULLUP);
```

```
pinMode (6, INPUT_PULLUP);
```

Vérifiez les boutons du panneau de commande et assignez la valeur comme étage de destination

```
if(firstFloor_Button==0)
```

```
    destinationFloor = 1;
```

```
else si (secondFloor_Button == 0)
```

```
    destinationFloor = 2;
```

```
else si (troisièmeBouton_étage == 0)
```

```
    destinationFloor = 3;
```

Comparer la variable de plancher actuelle au plancher de destination, est-elle supérieure, inférieure ou identique

```
if(destinationFloor>currentFloor){
```

```
    digitalWrite (9, LOW); // perdre le frein
```

```
    digitalWrite (12, HIGH); // définir la direction
```

```
    analogWrite (3 180); // régler la vitesse
```

```
}
```

```

elseif(destinationFloor<currentFloor){
  digitalWrite (9, LOW); // perdre le frein
  digitalWrite (12, FAIBLE); // définir la direction
  analogWrite (3 180); // régler la vitesse
}

```

```

else if (destinationFloor == currentFloor) {
  Serial.print ("Nous sommes arrivés");
  Serial.println ();
  digitalWrite (9, HIGH); // Frein
  analogWrite (3,0); // régler la vitesse }

```

Maintenant nous passons au capteur à ultrasons pour mesurer la distance, il utilise un code simple, envoie une onde sonore, ferme et écoute.

Enfin faire tes devoirs et convertir la différence de temps en distance

```

pinMode (trig, OUTPUT); // vérifier la distance entre l'élève et le plafond digitalWrite
(trig, LOW);

delayMicrosecondes (2);

digitalWrite (trig, HIGH);

delayMicrosecondes (5);

digitalWrite (trig, LOW); pinMode (écho, INPUT);

durée = pulseIn (écho, HAUT);

cm = durée / 29/2; // convertir le temps en distance en unités humaines normales

```

Prenez la distance de la dernière partie et déterminez à quel étage se trouve l'ascenseur, c'est une simple déclaration avec des distances prédéfinies, vous devrez l'adapter à votre modèle car il aura une proportion différente

```

si(cm<10)
  currentFloor = 3;
sinon si (25> cm && cm> 20)
  currentFloor = 2;
sinon si (cm> = 35)

```



```
currentFloor = 1;
```

### **III.4.3.2 Réalisation du programme de la plateforme via l'environnement de programmation IDE :**

```
Stepper MonMoteur(NbrPas, 8, 9, 10, 11);
```

```
void setup() {
```

```
    // Vitesse à 60 tours/min
```

```
    MonMoteur.setSpeed(60);
```

```
    Serial.begin(9600);
```

```
}
```

```
void loop() {
```

```
    // Faire un tour dans un sens
```

```
    Serial.println("Sens 1");
```

```
    MonMoteur.step(NbrPas);
```

```
    delay(500);
```

```
    // Faire un tour dans l'autre
```

```
    sens Serial.println("Sens 2");
```

```
    MonMoteur.step(-NbrPas);
```

```
    delay(500);
```

```
}
```

### **III.4.3.3 Conclusion :**

Dans ce chapitre, nous avons essayé de simuler la maquette d'ascenseur via le logiciel Isis Proteus 8 professionnel par une commande à microcontrôleur chargé par un programme en langage C contenant des instructions plus réduites et très simples afin de faciliter et de comprendre le fonctionnement de ce dernier .

Nous avons aussi fait une description conventionnelle de la maquette.

# Conclusion générale

# Conclusion générale

---

Au cours de ce projet de fin d'étude, nous avons pu concrétiser notre but principal qui consiste à faire un ascenseur auto-agissant à travers un système électronique.

Ce mémoire de fin d'étude a eu pour objectif de concevoir et réaliser un prototype d'un ascenseur aux personnes de mobilité réduite afin de faciliter leur déplacement commandé à base d'un microcontrôleur.

Nous nous sommes notamment aperçus que nous avons besoin de construire quatre cartes dans notre projet, que nous avons appelé : carte d'alimentation, carte de puissance, carte de commande et la carte d'affichage et pour cela nous avons fait appel à nos connaissances acquises au terme de notre formation académiques afin de mener à bien la réalisation de notre projet.

Ainsi ce que nous avons dû effectuer a été de créer un programme capable de mémoriser les étages comme dans certains grands immeubles, écrit en langage C, sous la gestion de microcontrôleur et aussi offrir un modeste attribut en intégrant une plateforme mobile rétractable afin de faciliter la mise en place des personnes à mobilité réduite .

Ce projet était intéressant pour plusieurs raisons. D'abord, l'ascenseur est un système répandu dans la vie courante qui nous est familier. Mais surtout cela nous a permis de travailler en groupe, en collectivité. Cette expérience nous a donné, d'ores et déjà, une idée du monde du travail. Ainsi que nous avons dû faire face à plusieurs problèmes plus ou moins faciles à résoudre.

Nous avons acquis de nombreuses connaissances durant ces six mois, et eu des compétences plus approfondies dans les domaines de l'électronique et de la programmation. Nous avons ainsi pu suivre et faire nous-mêmes les différentes étapes de la création carte de commande.

Tout cela et malgré la situation sanitaire causée par le Covid-19 et la fermeture des établissements d'enseignement supérieur, nous avons pu aboutir à un résultat convaincant et satisfaisant.

# Bibliographie

---

- [1] <http://www.ascenseurs.fr>
- [2] <http://www.energieplus-lesite.be>
- [3] <http://www.microchip.com>
- [4] <http://clubelek.insa-lyon.fr/joomla/fr/>
- [5] <http://bigonoff.com>
- [6] Les Microcontrôleurs PIC -1STI
- [7] Bernard Le wandowicz 4 C\_BL\_Le\_microcontrolleur\_PIC16F876.odt 16/05/14
- [8] <http://www.alldatasheet.com>
- [9] <https://www.aurel32.net/elec/lcd.php>
- [10] [http://www.composelec.com/afficheur\\_7\\_segments.php](http://www.composelec.com/afficheur_7_segments.php)
- [11] <http://klervithuault.wixsite.com/tpe-ascenseurs/ascenseurs-lectriques>
- [12] VASEUX Loïc & OUINE Corentin Lycée Le Corbusier Dossier.TPE2007-2008
- [13] <http://schema-montage-electronique.blogspot.com/2013/05/preparation-dun-afficheur-lcd-pour-une.html>
- [14] Université sidi Mohammed ben Abdallah École Supérieure de Technologie Département Maintenance Industrielle (2006).
- [15] ANNAELLE BEAU, CELINE THEVENOT. Entre fiction et réalité, l'étude du fonctionnement de l'ascenseur. Education (2012) : Céline Master Métiers de l'enseignement scolaire Mémoire professionnel de deuxième année.
- [16] MOHAMED LAMINE DILMI, UNIVERSITE SETIF -ABBAS FERHAT, Faculté de Technologie Département d'Electrotechnique, Mémoire de Master (2014).
- [17] <https://www.lemoniteur.fr/article/la-mise-en-securite-des-ascenseurs-existants.346199>.
- [18] <http://www.ikonet.com/fr/ledictionnairevisuel/arts-et-architecture/architecture/ascenseur/ascenseur.PHP>.
- [19] BOUALAM Lydia, **UNIVERSITE MOULOUD MAMMERI, TIZI-OUZOU, FACULTE DE GENIE ELECTRIQUE ET D'INFORMATIQUE DEPARTEMENT D'AUTOMATIQUE.**
- [20] EMBOUAZZA Lotfi, Université Aboubakr Belkaïd-Tlemcen Département de Génie Electrique et Electronique (GEE) Filière : Electronique

## Résumé :

Depuis la plus haute antiquité, les hommes ont cherché un moyen mécanique leur permettant de réduire considérablement les efforts nécessaires au déplacement de charges dans le sens vertical.

Jusqu'à la Renaissance, l'évolution des techniques de construction et du transport de marchandises va multiplier les systèmes de levage (cabestans, treuils simples avec poulies de rappel, élévations hydrauliques, monte-charge mais aussi grues pivotantes).

Les XVIIe et XVIIIe siècles voient apparaître de nouvelles idées et surtout un nouveau besoin : le transport des personnes. L'" ascenseur ». Equilibrée au moyen d'un contrepoids, son déplacement est assuré par la traction à bras. C'est seulement au XIXe siècle que le transport vertical des personnes et des charges prend toute son importance, en raison notamment de l'industrialisation.

Ce mémoire de fin d'études consiste à réaliser une maquette d'ascenseurs à étages a usage ordinaire est spécialement a usage pour personnes a mobilité réduite le système est commander à l'aide d'une carte microélectronique connue sous le nom de **ARDUINO** a base d'un microcontrôleur **ATMega328**.

**Mots clés :** Ascenseur, Transport vertical, Cabine, Microcontrôleur, Capteur

## ملخص

منذ العصور القديمة ، بحث الناس عن وسيلة ميكانيكية تسمح لهم بتقليل الجهود المطلوبة بشكل كبير لتحريك الأحمال في الاتجاه الرأسي.

حتى عصر النهضة ، أدى تطوير تقنيات البناء ونقل البضائع إلى زيادة عدد أنظمة الرفع (الرافعات ، الرافعات البسيطة (ذات البكرات المرتجعة ، المصاعد الهيدروليكية ، مصاعد الشحن ولكن أيضاً الرافعات المحورية

شهد القرنان السابع عشر والثامن عشر ظهور أفكار جديدة وقبل كل شيء حاجة جديدة: نقل الناس. "مصعد". المتوازنة من خلال ثقل موازن ، يتم ضمان حركته عن طريق الجر اليدوي. ولم يكتسب النقل العمودي للأشخاص والأحمال أهمية كبيرة إلا في القرن التاسع عشر ، لا سيما بسبب تصنيع

تتكون أطروحة نهاية الدراسة هذه من صنع نموذج للمصاعد متعددة الطوابق للاستخدام العادي خصيصاً للاستخدام للأشخاص ذوي القدرة المحدودة على الحركة ، ويتم التحكم في النظام باستخدام بطاقة إلكترونية دقيقة معروفة باسم **ARDUINO** من **ATMega328**.

**الكلمات المفتاحية:** مصعد ، نقل عمودي ، كابينة ، متحكم ، حساس

