



Faculté de Génie Electrique Et D'Informatique
DEPARTEMENT D'ELECTRONIQUE

Mémoire de fin d'études

En vue de l'obtention du diplôme de MASTER en Electronique

Option : **Instrumentation**

Thème

**Conception, et réalisation d'un ascenseur à base
de la carte de développement ARDUINO UNO**

Dirigé par :

M^r LAZRI .M

Réalisé par :

M^{elle} MAMMERI Sonia

M^{elle} MAOUCHI Tassadit

Promotion 2017/2018

Remerciement :

On dit souvent que le trajet est aussi important que la destination. Les cinq dernières années à l'université de maitrise nous ont permis de bien comprendre la signification de cette phrase toute simple.

Nous tenons à remercier énormément le bon dieu de nous avoir donné la force et le courage d'accomplir ce travail.

On remercie notre promoteur M^r LAZRI mourad pour son encadrement, sa compréhension et sa gentillesse durant toute la période de la préparation de notre mémoire.

Nous tenons aussi à remercier tous nos enseignants et membre de l'administration de département électronique.

On remercie également les membres du jury d'avoir accepté de juger ce modeste travail.

En fin, nous remercions tous les membres de la section << électronique >> ainsi les amis de toute la faculté.

Dédicaces:

Je dédie ce modeste travail, à mes très chers parents Yousef et Nadia qui m'ont toujours entourés et élevés.

Mon frère Rachid, son épouse Narimane, et mes deux sœurs Lylia et Wardia qui n'ont cessé d'être pour moi des exemples de courage et de générosité.

Bilal qui m'a soutenu dans tous les moments difficiles.

Toute ma famille pour leur soutien et encouragement.

Mon professeur encadreur monsieur Lazri Mourad pour son aide et sa précieuse attention.

Tous mes professeurs pour leurs conseils et leurs soutiens apportés chaque jour.

MAOUCHI TASSADIT

Dédicaces :

Je dédie ce travail à mes chers parents, pour tous leurs sacrifices, leur soutien et leurs prières tout au long de mes études, qui sont sources de tendresse et d'amour, ont fait de moi ce que je suis aujourd'hui, que ce rapport soit le meilleur cadeau que je puisse vous offrir.

Et mon profond amour à mon cher mari, quant je t'ai connu, j'ai trouvé l'homme de ma vie, mon âme sœur et la lumière de mon chemin ; tes sacrifices, ton soutien moral, ta gentillesse sans égal, ton profond attachement m'ont permis de réussir.

À mes deux frères Hakim et Yousef qui m'avez toujours soutenu et encouragé durant ses années d'études.

À tous les membres de ma famille, petits et grands, cousine et cousins, je vous aime fort.

À ma chère copine Felicia avec laquelle j'ai partagé des moments inoubliables, je te souhaite un avenir plein de joie, de bonheur, de réussite et de sérénité.

Egalement à tous mes amis qui ont toujours été là pour moi.

SONIA MAMMERI

Sommaire :

Introduction générale.....	1
Chapitre I : Généralités sur les ascenseurs.	
I.1.Préambule.....	2
I.2. Définition de l'ascenseur.....	2
I.3. Les bases de classification des ascenseurs.....	2
I.4. Les familles d'ascenseur.....	4
I.4.1. Les ascenseurs hydrauliques.....	5
I.4.1.1. Principe de fonctionnement.....	5
I.4.1.2. Les modèle d'ascenseur hydraulique.....	5
I.4.1.3. Composition.....	6
I.4.1.4. Avantages et inconvénients.....	7
I.4.2. Les ascenseurs à traction à câbles.....	8
I.4.2.1. Description.....	8
I.4.2.2. Principe de fonctionnement.....	8
I.4.2.3. Composition.....	9
I.4.2.4. Avantages et inconvénients.....	13
I.5. Système de motorisation.....	13
I.5.1. Moteur-treuil ou moteur à traction.....	14
I.5.1.1. Les moteurs-treuil à vis à une ou deux vitesses.....	14
I.5.1.2. Les moteurs-treuil planétaire.....	15
I.5.1.3. Les moteurs à attaque directe (gearless ou sans treuil).....	16

I.6. Critères de choix de la motorisation.....	17
I.7. Discussions	18

Chapitre II : Description de la carte ARDUINO UNO

II.1. Préambule.....	19
II.2. Description et mise en œuvre de la carte ARDUINO UNO	19
II.2.1. Description matériel	19
II.2.2. LE logiciel.....	24
II.2.3. Les caractéristiques de la carte ARDUINO UNO.....	25
II.2.4. La configuration de l’interface ARDUINO UNO.....	25
II.2.5. L’environnement de développement.....	25
II.2.6. Le test de la carte ARDUINO UNO.....	26
II.2.7. Le montage.....	27
II.3. Les avantages de la carte ARDUINO UNO.....	27
II.4. Alimentation de la carte ARDUINO UNO.....	28
II.5. Protection du port USB contre la surcharge en intensité.....	29
II.6. Gestion de la mémoire.....	30
II.7. Les entrées/sorties numériques.....	30
II.8. Communication avec l’extérieure.....	33
II.9. Microcontrôleur ATMEL ATmega328.....	34
II.10. Discussion.....	36

Chapitre III : conception et programmation

III.1. Préambule.....	37
III.2. Partie théorique.....	37
III.2.1. Le moteur.....	37
III .2.2.Le pont en H.....	38
III.2.2.1. Principe de fonctionnement.....	39
III.2.2.2.Le L298N.....	41
III.2.3. Afficheur 7 segments.....	42
III.2.3.1. Description.....	42
III.2.3.2. Fonctionnement.....	43
III.2.3.3. Branchement de l’afficheur.....	43
III.3. Partie pratique.....	44
III.3.1. Partie logiciel.....	44
III.3.1.1. Schéma synoptique de la carte de commande	45
III.3.1.2. Branchement des boutons poussoirs.....	45
III.3.1.3. Branchement de l’afficheur 7 segments	46
III.3.1.4. Schéma de commande globale.....	46
III.3.2. Partie programmation.....	47
III.3.2.1. Réalisation du programme.....	47
III.3.3. Partie réalisation.....	52
III.4. Discussion.....	52

Chapitre VI : Test et validation

VI.1. Préambule.....	53
VI.2. Définition de processing	53
VI.3. Interface de precessing.....	53
VI.4. L'espace de dessin	54
VI.5. Teste de notre maquette sur processing	55
VI.6. Discussion.....	56
Conclusion générale	57

Bibliographie

Listes des figures :

Chapitre I : Généralités sur les ascenseurs

Figure (I-1) : les familles d'ascenseur.....	4
Figure (I-2) : schéma illustratif du principe des pompes hydrauliques.....	5
Figure (I-3) : différents ascenseurs hydraulique.....	6
Figure (I-4) : ascenseur à moteur treuil.....	8
Figure (I-5) : ascenseur à moteur à attaque directe.....	8
Figure (I-6) : schéma de principe de fonctionnement d'un ascenseur à traction.....	9
Figure (I-7) : la constitution de treuil.....	10
Figure (I-8) : système barrage.....	12
Figure (I-9) : le bouton d'appelle.....	12
Figure (I-10) : moteur-treuil à vis sans fin.....	14
Figure (I-11) : moteur gearless classique.....	15
Figure (I-12) : moteur-treuil planétaire.....	17

Chapitre II : Description de la carte ARDUINO UNO

Figure (II-1) : carte ARDUINO UNO	20
Figure (II-2) : La structure d'un programme avec ARDUINO.....	24
Tableau (II-3) : Les caractéristiques de la carte ARDUINO UNO.....	25
Figure (II-4) : l'environnement de développement ARDUINO.....	26
Figure (II-5) : le montage de la carte ARDUINO.....	27
Figure (II-6) : les différentes broches de la carte ARDUINO UNO.....	31
Figure (II-7) : Atmega328.....	34

Chapitre III : conception et programmation

Figure (III-1) : moteur à courant continu.....	38
Figure (III-2) : pont en H.....	38
Figure (III-3) : schéma interne d'un pont en H.....	39
Figure (III-4) : fonctionnement dans le sens horaire.....	39
Figure (III-5) : fonctionnement dans le sens antihoraire.....	40
Figure (III-6) : freinage avec S1 et S2 et avec S3 et S4 respectivement.....	40
Figure (III-7) : configuration des broches du L298N.....	41
Figure (III-8) : afficheur 7 segments.....	42
Figure (III-9) : afficheur 7 segments identifiés avec des lettres.....	42
Figure (III-10) : types d'afficheur 7 segments.....	43
Figure (III-11) : schéma synoptique de la carte de commande.....	45
Figure (III-12) : branchement des boutons poussoirs.....	45
Figure (III-13) : branchement de l'afficheur.....	46
Figure (III-14) : le branchement global de la maquette.....	46
Figure (III-15) : organigramme.....	47
Figure (III-16) : sous-programme 1.....	48
Figure (III-17) : sous-programme 2.....	49
Figure (III-18) : sous-programme 3.....	50
Figure (III-19) : sous-programme.....	51
Figure (III-20) : maquette de l'ascenseur.....	52

Chapitre VI : Test et validation

Figure (VI-1) : interface de processing.....	53
Figure (VI-2) la position 0 (rai de chaussé).....	54
Figure (VI-3) position niveau 1.....	55
Figure (VI-4) position niveau 2.....	56

Liste des tableaux :

Chapitre II : Description de la carte ARDUINO UNO.

Tableau (II-1) : Les différentes broches de la carte ARDUINO UNO.....21

Tableau (II-3) : Les caractéristiques de la carte ARDUINO UNO.....25

Résumé :

Tout au long de notre travail mené durant notre mémoire nous avons conçu et réaliser la maquette d'un ascenseur à base d'ARDUINO UNO.

Et vu le manque de la partie mécanique (pignons, axes, moteurs destiner pour les ascenseurs, poulie.....etc.), on s'est donc orienter vers le processing, qui est le logiciel de simulation de cette partie mécanique au lieu de rester sans rien faire.

Introduction générale :

Ce présent travail est réalisé au sein du laboratoire « Maquette » du département d'électronique, faculté de Génie Electrique et d'informatique, Université Mouloud Mammeri Tizi-Ouzou. Il s'inscrit dans l'optique de réalisation d'un ascenseur.

Pour cela, le domaine de l'électronique, l'instrumentation est l'un des principaux piliers du développement de la technologie.

Elle constitue un vaste domaine qui regroupe principalement les équipements de terrain et les dispositifs de contrôle qui permettent de mesurer et de contrôler les différents paramètres physiques.

L'électronique est très utilisée pour commander des systèmes mécaniques c'est le cas des ascenseurs ou une application électronique est mise en place pour commander leurs déplacements.

Les ascenseurs sont des dispositifs mobiles assurant le déplacement des personnes (et des objets) en hauteur sur des niveaux définis d'une construction.

Il est installé la plus part du temps dans une cage d'ascenseur, une trémie verticale fermée.

L'objectif de ce projet est la réalisation du fonctionnement d'un ascenseur sous le logiciel ARDUINO.

Pour ce faire, nous avons divisé notre travail en quatre chapitres :

Dans le chapitre I nous présentons les généralités sur les ascenseurs.

Ensuite nous ferons la description de la carte ARDUINO UNO dans le chapitre II.

Puis dans le chapitre III nous allons faire la Conception et programmation.

Les tests et validation seront réalisés dans le chapitre VI.

Enfin nous terminons notre travail avec une conclusion générale.

Chapitre I

Généralités sur les ascenseurs

I.1. Préambule :

Un ascenseur est un dispositif mobile ou semi-mobile assurant le déplacement des personnes (et des objets) en hauteur sur des niveaux définis d'une construction.

L'objectif de ce chapitre est de présenter des généralités sur les ascenseurs. Dans un premier temps, nous allons présenter la définition de l'ascenseur et leur base de classification, puis les familles de l'ascenseur et on verra ensuite les avantages et les inconvénients de chaque famille, on termine par le système de motorisation.

I.2. Définition de l'ascenseur :

Un ascenseur peut se définir comme un appareil élévateur vertical destiné au transport des personnes ou des charges entre différents étages d'un immeuble.

Il comporte une cabine entraînée par un moteur électrique à l'aide d'un câble métallique.

On distingue techniquement trois types d'appareils :

Les ascenseurs destinés à l'usage unique des personnes.

Les Monte-charge destinés aux marchandises, sa cabine n'est pas accessible à des personnes et les dimensions diffèrent. Ces dimensions sont comme suite :

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

Les Monte-charge industriels destinés aux marchandises uniques.

I.3. Les bases de Classification des ascenseurs :

1. Suivant la charge et la vitesse :

On distingue quatre types :

- Ascenseur des passagers :

La charge varie de 500 à 1000 kg, et la vitesse peut prendre les valeurs suivantes : 0,7 et 1,4m/s.

- Ascenseurs rapides des passagers :

La charge est généralement prise entre 1000 et 1600kg, et la vitesse varie entre 2 et 4 m/s.

- Ascenseurs des charges :

La charge peut prendre les valeurs : 500, 1000, 2000, 3200 et 5000kg, et la vitesse varie entre 0,25 et 0,5 m/s ou la vitesse 0,25 m/s est particulièrement utilisée pour la charge 5000kg.

- Ascenseurs des petites charges :

La charge varie de 100 à 160kg, et la vitesse égale 0,5 m/s.

2. Suivant la constitution des machines de traction :

Les machines de traction peuvent être à entraînement direct sans ou avec réducteur de vitesse.

- Machines de traction sans réducteur de vitesse : L'utilisation des machines de traction sans réducteur de vitesse, permet d'absorber le bruit du moteur, donc de diminuer les pertes d'énergie.
- Machines de traction avec réducteur de vitesse :

Elles sont constituées d'un moteur à courant continu, son axe est relié directement à l'élément de traction (poulie ou enroulement), qui tourne à la même vitesse que celle du moteur, cette vitesse est généralement supérieure à 1,8 m/s.

Dans ce genre de machines, l'axe extérieur de réducteur est relié à l'axe de l'élément de traction, et l'autre coté de réducteur est relié au moteur.

Pour ce type on utilise, les moteurs à courant continu ou alternatif, mais généralement, les moteurs asynchrones à rotor en court-circuit à une ou deux vitesses sont les plus souvent utilisés pour obtenir une grande précision d'arrêt.

3. Suivant les éléments de traction :

On distingue trois éléments de traction :

- la poulie d'adhérence :

Le moment de traction du moteur entraîne la poulie d'adhérence à travers le réducteur de vitesse (s'il existe) afin qu'il entraîne les câbles, comme il est indiqué sur la figure suivante :

- l'enroulement :

Il existe deux types d'enroulement :

Enroulement à un seul côté.

Enroulement à deux côtés.

Remarque

Les ascenseurs qui fonctionnent avec enroulements, ne contiennent pas des contrepoids.

- Le mouflage :

C'est un ensemble de poulies fixes et d'autres mouvements, elles sont utilisées pour diminuer la force sur les câbles de traction, en conséquence sur la poulie d'adhérence et le moteur, ce qui permet d'augmenter la charge du moteur.

I.4. Les familles d'ascenseur :

On distingue essentiellement deux types de familles d'ascenseur

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

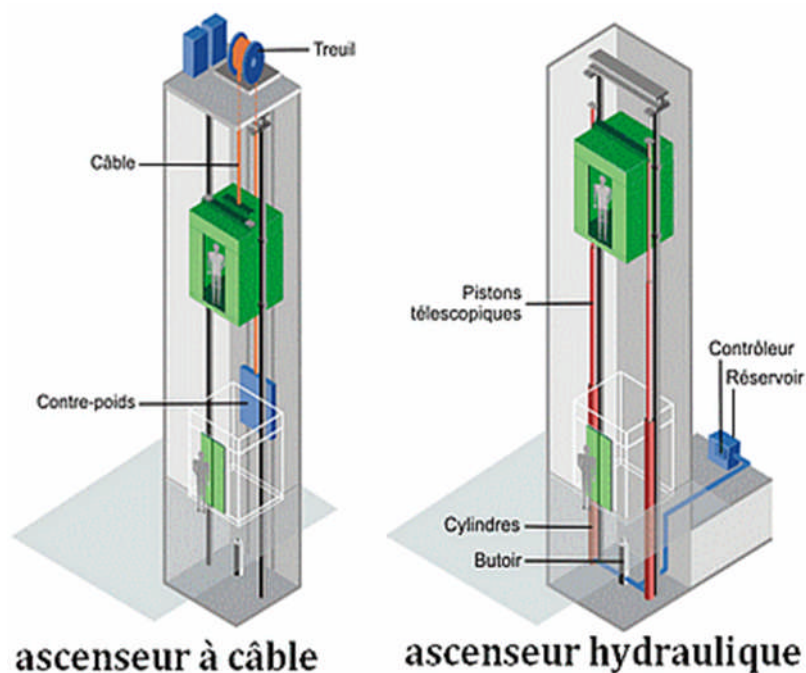


Figure (I-1) les familles d'ascenseur.

En règle générale, ces deux types utilisent l'énergie électrique pour déplacer les cabines verticalement (moteur électrique continu ou alternatif).

I.4.1. Les ascenseurs hydrauliques :

I.4.1.1.Principe de fonctionnement :

L'ascenseur hydraulique s'adapte à des espaces restreints et à des vitesses de déplacement modérées.

Comme toute machine hydraulique :

- La cabine de l'ascenseur se déplace sur simple pression d'un bouton qui actionne un piston contenant de l'huile.
- L'huile est envoyée dans le vérin qui actionne le piston via une centrale hydraulique (pompe).
- Lorsque le piston se remplit, la cabine monte.
- La descente s'effectue par un simple arrêt de la pression : le piston évacue alors le surplus d'huile.

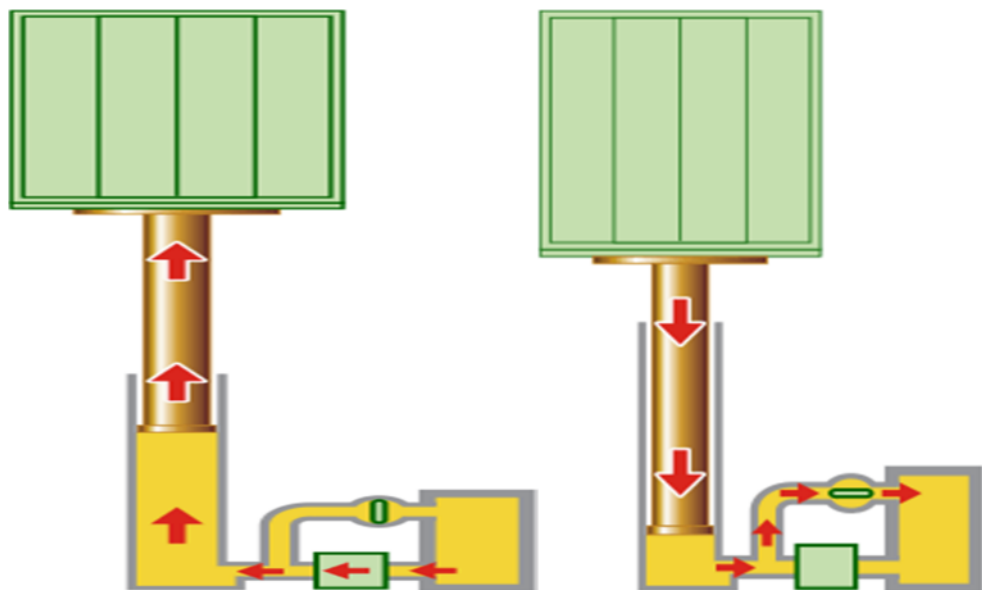
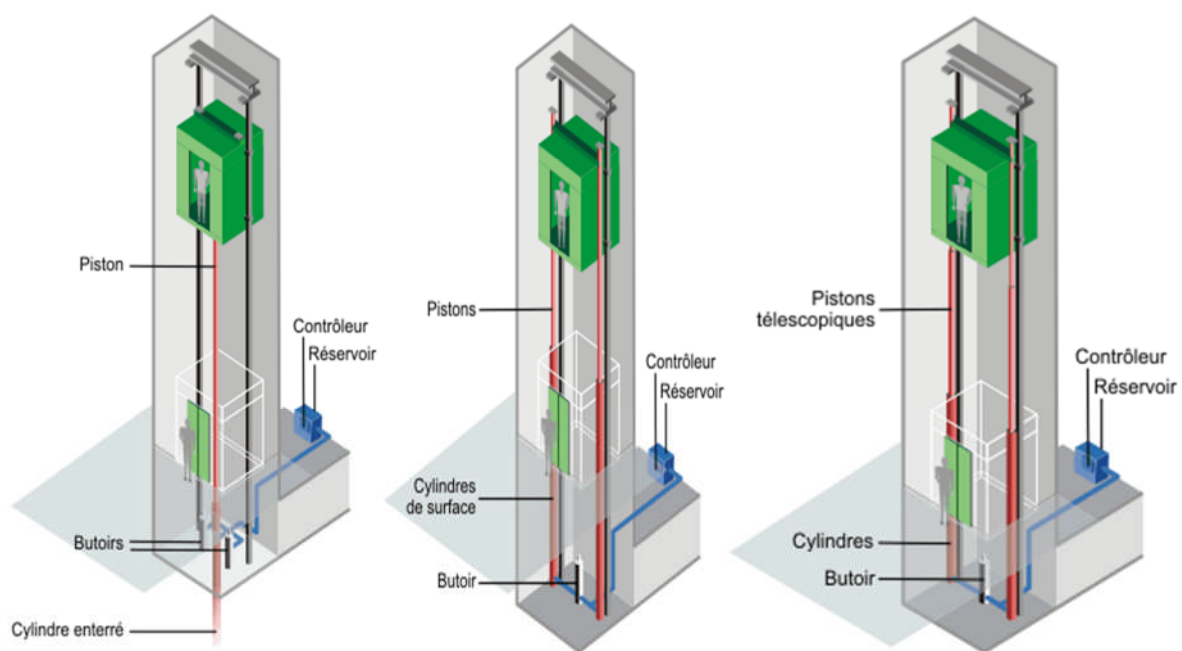


Figure (I-2) schéma illustratif du principe des pompes hydrauliques.

I.4.1.2. Les modèles d'ascenseur hydraulique.

Ce type d'ascenseur assure généralement des déplacements courts de l'ordre de quinze à dix-huit mètres maximum, Plusieurs modèles d'ascenseurs hydrauliques sont proposés :

- A cylindre enterré ;
- A cylindre de surface ;
- A cylindre de surface télescopique ;



A cylindre enterré

A cylindre de surface

A cylindre de surface

télescopique

Figure (I-3) différents ascenseurs hydrauliques.

“Télescopique” : utilisé lorsque la place disponible est réduite et la course utile importante. Constitués par autant de pistons plongeurs de longueurs variables, en fonction de la course totale nécessaire. Les pistons plongeurs sont creux.

I.4.1.3. Composition :

Les ascenseurs hydrauliques se composent :

- D'une cabine.
- De guides.

- Des groupes hydrauliques.
- D'un réservoir d'huile.
- D'un moteur électrique accouplé à une pompe hydraulique.
- D'un contrôle de pression.

Les différents modèles permettent de tenir compte d critères :

- De place ;
- De hauteur d'immeuble à desservir ;
- De stabilité de sol et de sous-sol ;
- De risque de pollution par rapport au sol et plus spécifiquement aux nappes phréatiques.

I.4.1.4. Avantages et inconvénients :

Tous comme l'ascenseur électrique, l'ascenseur hydraulique présente des avantages et inconvénients qu'il faut mesurer selon nos exigences à notre ascenseur particulier.

Avantage :

D'une manière générale, l'ascenseur hydraulique présente ces avantages par rapport à l'ascenseur électrique :

- ✓ Il est silencieux ;
- ✓ Il est facile à installer ;
- ✓ Son entretien est relativement simple
- ✓ Les déplacements de la cabine s'effectuent en douceur ;

Inconvénients :

L'ascenseur hydraulique présente néanmoins quelques inconvénients :

- ✓ Course vertical limité à une hauteur entre 15 et 18m ;
- ✓ Sa vitesse de déplacement réduite ;
- ✓ Le nécessaire renforcement de la dalle du sol lors de son installation ;
- ✓ Sa consommation importante en énergie et en huile ;
- ✓ L'absence de contrepoids provoque un surdimensionnement des consommations et des appels de puissance importantes (à charge et a vitesse égales, il faut de l'ordre de 3 fois plus puissance)

I.4.2. Les ascenseurs à traction à câbles :

I.4.2.1. Description :

Les ascenseurs à traction à câbles sont les types d'ascenseurs que l'on rencontre le plus notamment dans les bâtiments tertiaires.

Ils se différencient entre eux selon le type de motorisation :

- A moteur-treuil a vis sans fin
- A moteur-treuil planétaire.
- A moteur a attaque directe (couramment appelé <Gearless> ou sans treuil)

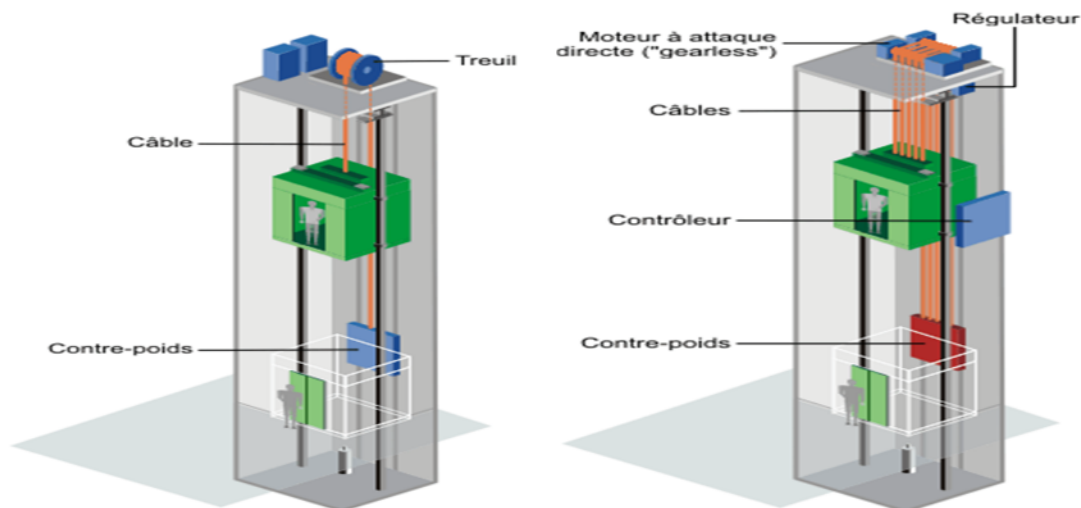


Figure (I-4) ascenseur à moteur treuil. Figure (I-5) ascenseur à moteur à attaque directe.

I.4.2.2. Principe de fonctionnement :

Ce type d'ascenseur est doté d'un moteur électrique à l'aide duquel il se déplace, qui se trouve dans la partie supérieure de la gaine.

La cabine est reliée au contrepoids au moyen de la poulie de traction et au moyen de câbles métalliques.

Les câbles, actionnés par un treuil permettent de mettre en mouvement la cabine et le contrepoids sur les câbles il Ya aussi des rainures qui empêchent le glissement des câbles métalliques.

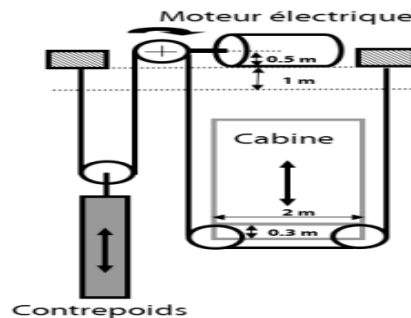


Figure (I-6) schéma de principe de fonctionnement d'un ascenseur à traction.

I.4.2.3. Composition :

Les ascenseurs à traction à câbles comprennent :

- La gaine : Gaine verticale dans laquelle se déplacent l'ascenseur et son contrepoids. Celle-ci est équipée de guides en acier destinés à guider la suspension de la cabine et le contrepoids profilée en acier, généralement en forme de T, destinés à guider la cabine et le contrepoids dans la gaine, ou la gaine est un volume dans lequel se déplace la cabine et le contrepoids, s'il en existe un. Ce volume est matériellement délimité par le fond de la cuvette, les parois et le plafond.
- Une cabine ou benne : Destinée à accueillir les personnes et les marchandises. Constituée de quatre parties principales qui, dans l'ordre de leur montage pendant l'installation, sont :
 - L'étrier ;
 - Le plancher ;
 - Les parois ;
 - Le toit ;
- Un contrepoids : comme pour la cabine, le contrepoids comprend aussi :
 - Un étrier ;
 - Des masses de fonte appelées gueuses de contrepoids ;
- Des Câbles : les câbles assurent la liaison entre cabine et contrepoids. Les extrémités des câbles sont reliées, l'une à l'étrier de la cabine et l'autre à l'étrier du contrepoids.

- Des guides : destiné à guider la cabine et le contrepoids dans la gaine.
- Un système de traction au-dessus de la cage de l'ascenseur.
- Le moteur : Le moteur le plus couramment employé est actuellement le moteur asynchrone à double cage, ce choix à des avantages de faible cout (y compris pour de fortes puissances), très robuste (pas de pièce de contact) et un bon rendement.
- Les portes palières : c'est la porte externe de l'ascenseur, chaque ascenseur est équipé d'autant de porte palière que de nombre d'étage. Elles peuvent être battante et commandées manuellement ou automatique et coulissantes (à ouverture centrale ou latérale).

Pour faire rentrer ou sortir (ouvertures ou fermeture de porte) un vérin double effet, on doit inverser l'alimentation en pression deux chambre.

Moins couteux qu'un vérin simple effet, il est utilisé dans l'application pneumatique ne nécessitant pas de retour en position initiale en cas de coupure de pression.

- Le treuil : Le treuil entraine la cabine et le contrepoids dans leur mouvement comme l'indique la figure I.9. Nous ne décrivons ici que les treuils à adhérence. Le treuil à tambour sur lequel les câbles s'enroulent en spirale n'est plus utilisé que dans les anciens bâtiments, il n'est pas possible de disposer un contrepoids. Le treuil est composé des éléments suivants (et comme indique sur la figure (1-7)):
- Un moteur de traction ;
 - Réducteur roue vie et sans fin ;
 - Poulie de traction ;
 - Volant de traction ;
 - Tambour de traction ;

Le treuil est constitué :

- D'un moteur électrique
- D'un réducteur roue et vis sans fin
- D'une poulie de traction
- D'un volant d'inertie
- D'un embrayage frein à tambour

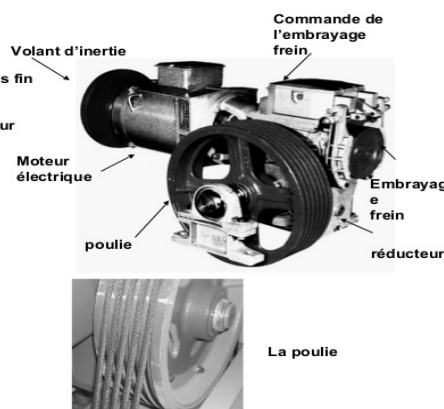
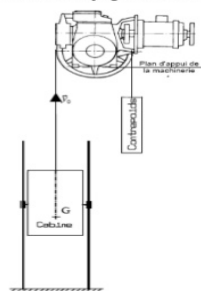


Figure (I-7) la constitution de treuil.

- **Contact de surcharge :** Lorsque la surface de la cabine d'un ascenseur excède la valeur imposée, le contact de surcharge est nécessaire, car il peut éviter des incidents graves tels que la détérioration du moteur lorsque l'appareil est en montée ou à l'arrêt sur les amortisseurs du fond de cuvette pour la descente .ce contact est placé soit sous le plancher de la cabine, soit sur le dispositif d'attache de l'arcade à la suspente, dans les deux cas, la surcharge provoqueun déplacement excessif du plancher ou de l'attache et interdit le démarrage de l'appareil. Il est indispensable de doubler l'action de ce contact par une signalisation optique et acoustique .un voyant (surcharge) sur la boîte à boutons cabine et un ronfleur ou une sonnerie sont alors prévus .
- **Contact de présence de la cabine :** Le contact de présence de la cabine se fait à l'aide des capteurs de proximités ou «détecteurs de présences » qui sont des dispositifs autrefois mécaniques, et aujourd'hui de plus en plus caractérisés par l'absence de liaison mécanique entre le dispositif de mesure et l'objet ciblé (personne, animal, objet animé tel qu'un véhicule). L'interaction entre le capteur et sa "cible" est alors réalisée par l'intermédiaire d'une caméra associée à un système d'analyse de l'image, ou plus souvent d'un champ (magnétique, électrique, électromagnétique).
Selon les capteurs, objets et situation, l'objet détecté doit être plus ou moins proche du capteur ou illuminé par une source rayonnante (éventuellement non-visible, par exemple dans l'infrarouge).
- **capteur de proximité :** Les capteurs de proximité sont utilisés soit en mode analogique, soit en mode binaire.
Dans le premier cas, l'amplitude du signal est une fonction de la position relative de l'objet ciblé.
Dans le second cas, le signal ne peut avoir que deux niveaux (haut et bas), selon que l'objet est présent à proximité ou non du capteur inductif
- **Photocellules :** Elles sont destinées à la détection d'objets de toute nature. Le concepteur a le choix entre trois systèmes de base :
 - Le système reflex.
 - Le système de proximité.Dans les portes d'ascenseurs on utilise le système barrage (Figure I-8), pour les longues portes et objets réfléchissants. Ce système de détection contient un émetteur et

récepteur séparés permet de détecter tout objet interrompant le faisceau lumineux (les personnes). Dans ce cas la porte ne ferme pas.

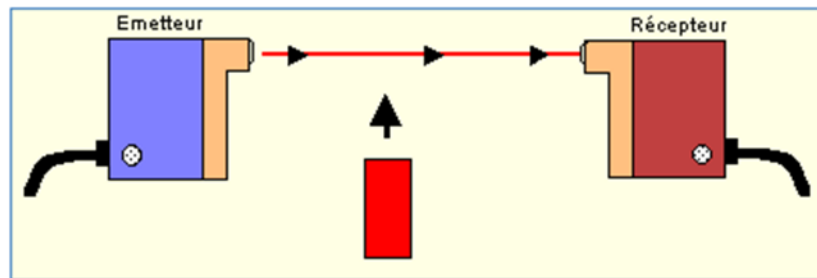


Figure (I-8) système barrage.

- Les commandes d'appel : Les commandes d'appel ont pour but de faire venir l'appareil aux étages d'où elles émanent respectivement .cette utilisation, a posteriori par rapport à la commande d'envoi, justifie la priorité que celle-ci doit avoir sur la commande d'appel.

L'appel peut être spécialisé lorsqu'il est effectué sur des boutons différents suivant que l'utilisateur désire monter ou descendre.

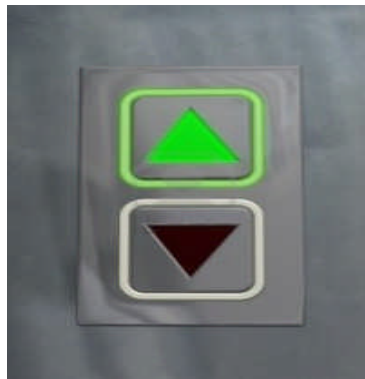


Figure (1-9) le bouton d'appelle

- Les commandes d'envoi : Leur rôle est d'envoyer la cabine du niveau où elle se trouve au niveau auquel la personne qui effectue l'envoi le désire. La commande d'envoi s'effectue par l'intermédiaire d'un bouton qui peut être placé soit en cabine dans la majorité des cas, soit pour certains monte –charge sur le palier. Lorsque la commande d'envoi est faite en cabine, elle doit évidemment avoir priorité sur la commande

extérieure. Pour les appareils du groupe 1 (ascenseur et monte-charge accompagnés), l'utilisateur qui est entré dans la cabine doit disposer, pour appuyer sur le bouton de son choix, d'un certain laps de temps après la fermeture des portes avant qu'une commande d'appel faite de l'extérieur puisse être exécutée.

I.4.2.4. Avantages et inconvénients :

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

Avantages :

- ✓ Course verticale pas vraiment limitée ;
- ✓ Suivant le type de motorisation précision au niveau de la vitesse et du déplacement ;
- ✓ Rapidité du déplacement ;
- ✓ Efficacité énergétique importante ;
- ✓ Pas de souci de pollution ;

Inconvénients :

- ✓ En version standard (treuil à réduction), nécessite un locale de machinerie en toiture ;
- ✓ Exigence très importante sur l'entretien ;
- ✓ Peut imposer un volume construit inesthétique visible sur le toit ;
- ✓ Problème d'accessibilité ;

I.5. Système de motorisation :

Entre ascenseur hydraulique et électrique ; on remarque que Les ascenseurs à tractions à câbles sont de loin ceux le plus répandu au niveau du parc machine. C'est une des raisons pour laquelle on détaillera plus la motorisation des ascenseurs à traction à câbles. De plus, énergétiquement parlant, la motorisation des ascenseurs hydrauliques a des rendements assez faibles (de l'ordre de 20 %) et, de par l'absence entre autres de contrepoids, les moteurs hydrauliques consomment plus et ont des appels de puissance au démarrage beaucoup plus importants.

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

I.5.1. Moteur-treuil ou moteur à traction :

1.5.1.1 les moteur-treuil à vis sans fin à une ou deux vitesses :

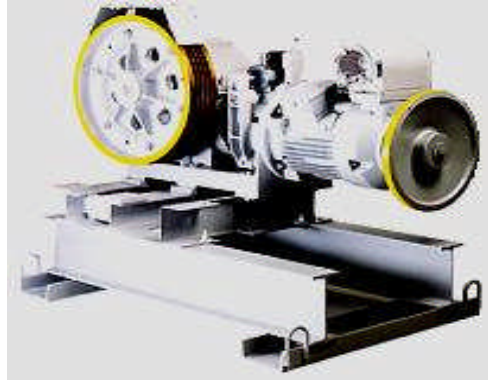


Figure (I-10) moteur-treuil à vis sans fin.

Les moteurs-treuil avec vis sans fin sont abandonnés au profit des moteurs à attaque directe (sans réducteur ou "gearless").

Dans ce type de motorisation, la vis sans fin entraîne beaucoup de pertes mécaniques et, par conséquent, des consommations électriques plus importantes. Les moteurs électrique 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 a 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).

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

Avantage :

- ✓ couple élevé.
- ✓ grande plage de variation de vitesse.
- ✓ précision dans les déplacements et sur la régulation de vitesse.

Inconvénients :

- ✓ Entretien important.
- ✓ efficacité énergétique faible.
- ✓ consommation électrique non négligeable.

I.5.1.2. les moteurs-treuil planétaires :

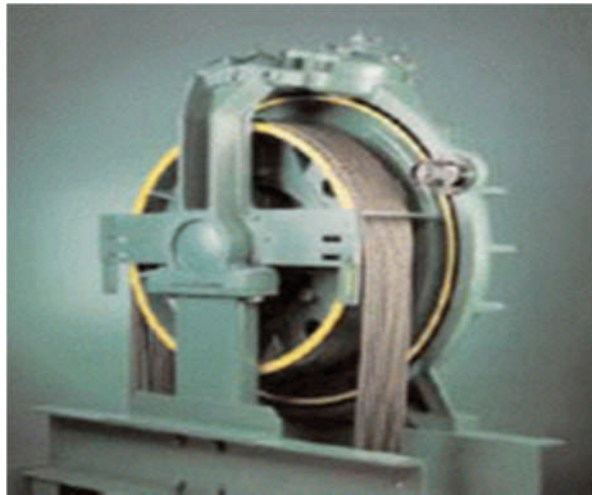


Figure (I-11) moteur gearless classique.

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 souhaitée.

Ce système a 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 au niveau du moteur-treuil (de l'ordre de 80 %).

Les réducteurs planétaires peuvent être accouplés à des moteurs électriques :

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

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

Avantage :

- ✓ Couple important.
- ✓ précision dans les déplacements et sur la régulation de vitesse.

Inconvénients :

- ✓ entretien nécessite une main d'œuvre qualifié.

I.5.1.3 Les moteurs a 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 part la présence d'un variateur de fréquence qui optimise la consommation énergétique ; aussi les pertes mécaniques sont réduite vu l'absence des engrenages.

Certains constructeurs annoncent des rendements énergétiques de l'ordre de 80 %.

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

Avantage :

- ✓ Vitesse optimisé par le variateur de fréquence.
- ✓ Compacité du système.
- ✓ pas de local des machines nécessaire pour les ascenseurs.
- ✓ pertes mécaniques réduite.
- ✓ faible niveau de bruit.
- ✓ poids réduit.

Inconvénients :

- ✓ la capacité peut entrainer des difficultés de maintenance.
- ✓ difficulté d'intervention dans la cage d'ascenseur.

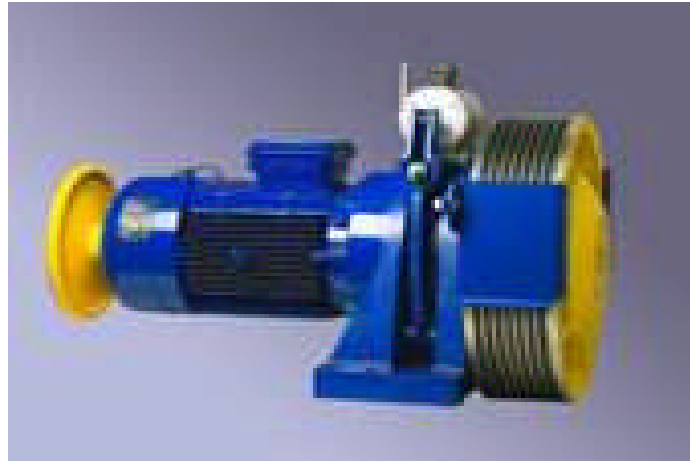


Figure (I-12) moteur-treuil planétaire

I.6. Critère de choix de la motorisation :

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

1) 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 :

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

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.

2) 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.

3) 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).

I.7. Discussion

Dans ce chapitre nous avons présenté le principe de fonctionnement des ascenseurs. Aussi, nous avons donné les différents types de famille d'un ascenseur, la différence entre ces deux types et leurs avantages et inconvénients. Nous avons également présenté les systèmes de motorisation.

Chapitre II

Description de la carte Arduino uno

II.1. Préambule :

Dans ce chapitre, nous allons donner les généralités sur la carte ARDUINO UNO qui est une plateforme de prototypage d'objets interactifs à usage créatif constituée d'une carte électronique et d'un environnement de programmation. Cet environnement matériel et logiciel permet à l'utilisateur de formuler ses projets par l'expérimentation directe avec l'aide de nombreuses ressources disponibles en ligne.

Dans un premier temps, nous allons voir la description et la mise en œuvre de la carte ARDUINO UNO, on verra ensuite ses avantages et son alimentation puis la protection du port USB contre la surcharge en intensité et la gestion de la mémoire ainsi que les entrées sorties numériques et la communication avec l'extérieur. Nous terminons avec le microcontrôleur Atmega 328.

II.2. Description et mise en œuvre de la carte ARDUINO UNO.

ARDUINO UNO est un circuit imprimé en matériel libre sur lequel se trouve un microcontrôleur qui peut être programmé pour analyser et produire des signaux électrique, d'une manière à effectuer des tâches très diverses.

II.2.1. Description matériel.

C'est une plateforme électronique basée sur une interface Entrée /Sortie simple autour d'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/I2C ; exploitation de servomoteurs et d'autres applications.

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 femelle 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 I2C.

La carte ARDUINO UNO dispose aussi de 14 entrées/sorties numériques chacune est repérées 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 la broche numéro 11 ; peut être utilisée 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 remettre le processus à zéro.

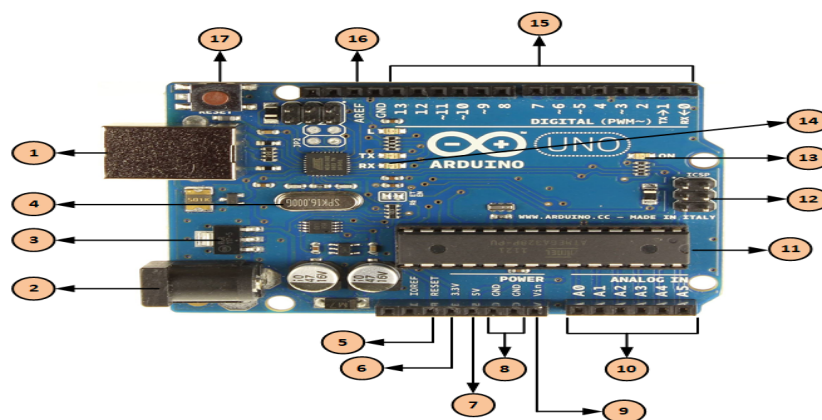


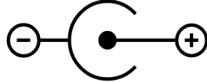


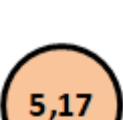

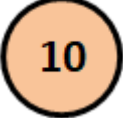






Figure (II-1) carte ARDUINO UNO

	<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><i>Diamètre interne 2.1mm, externe 5.5mm</i></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>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. 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>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>Broches (3.3, 5, GND, Vin)</p> <p>3.3V (6) – Broche d'alimentation de tension 3.3 Volts 5V (7) – Broche d'alimentation de tension 5 Volts</p> <p>La plupart des composants destinés à fonctionner avec Arduino fonctionnent bien en 3.3 Volts ou 5 Volts.</p> <p>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.</p> <p>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>

	<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 1024 points.</p>
	<p>Microcontrôleur principal</p> <p>Chaque carte Arduino possède son propre microcontrôleur (11). 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. 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. Pour plus de détails sur la construction et les fonctions du microcontrôleur, vous pouvez vous référer à la fiche technique (<i>data sheet</i>).</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...). On ne va pas se préoccuper de ce connecteur au début des tutoriels.</p>
	<p>Indicateur LED d'alimentation</p> <p>Ce voyant doit s'allumer lorsque vous branchez votre Arduino sur une source 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, et je ne parle pas de nourriture ici.</p>
	<p>LEDs TX et RX</p> <p>Sur votre carte, vous trouverez deux indicateurs : TX (émission) et RX (réception). Ils apparaissent à deux endroits sur la carte Arduino UNO. Tout d'abord, sur les broches numériques 0 et 1, pour indiquer les broches responsables de la communication série. Deuxièmement, les LEDs TX et RX (13). 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. La vitesse de transmission s'exprime en bauds, soit l'équivalent du bits/seconde si le signal est binaire.</p>



	<p>Entrées/Sorties numériques</p> <p>La carte Arduino UNO possède 14 broches d'Entrées / Sorties numériques (15), 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 LEDs, des relais, etc. Les broches étiquetées “~” peuvent être utilisées pour générer des PWM.</p>
	<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-1) les différents broches de la carte ARDUINO UNO

II.2.2. le logiciel.

Le logiciel de programmation des modules ARDUINO est une application java, libre et multiplateformes, servant d'éditeurs de code et de compilateur, et qui peut transférer le firm Ware et le programme au travers de la liaison série (RS-232, BLUETOOTH ou USB selon le module). Il est également possible de se passer de l'interface ARDUINO, et de compiler les programmes via l'interface en ligne de commande.

Le langage de programmation utilisé est le C++, compilé avec AVR-g++3, et lié à la bibliothèque de développement ARDUINO, permettant l'utilisation de la carte et de ses entrées/sorties.

La mise en place de ce langage standard rend aisé le développement de programme sur les plates formes ARDUINO, à toute personne maitrisant le C et le C++.

Programmer avec Arduino

Un programme utilisateur Arduino est une suite d'instructions élémentaires sous forme textuelle, ligne par ligne. La carte lit puis effectue les instructions les unes après les autres, dans l'ordre défini par les lignes de code.

Structure d'un programme
Il y a trois phases consécutives:

- 1/La définition des constantes et des variables
- 2/La configuration des entrées et sorties
- 3/La programmation des interactions et comportements

Une fois la dernière ligne exécutée, la carte revient au début de la troisième phase et recommence sa lecture et son exécution des instructions successives. Et ainsi de suite.

Cette boucle se déroule des milliers de fois par seconde et anime la carte.

Commentaires multilignes pour se souvenir du patch ==>

Commentaires

```

Arduino - 0006 Alpha
sketch_061111a.g

/* Ce programme fait clignoter une LED branchée sur la broche 13
 * et fait également clignoter la diode de test de la carte
 */

int ledPin = 13; // LED connectée à la broche 13

void setup()
{
  pinMode(ledPin, OUTPUT); // configure ledPin comme une sortie
}

void loop()
{
  digitalWrite(ledPin, HIGH); // met la sortie à l'état haut (led allumée)
  delay(3000); // attente de 3 secondes
  digitalWrite(ledPin, LOW); // met la sortie à l'état bas (led éteinte)
  delay(1000); // attente de 1 seconde
}
  
```

Done compiling.

22

Figure (II-2) La structure d'un programme avec ARDUINO

II.2.3. Les caractéristiques de la carte ARDUINO UNO.

Le tableau suivant donne les caractéristiques de la carte Arduino.

Microcontrôleur	ATmega328
Fréquences Horloge	16 Mhz
Tension de fonctionnement	5V
Tension d'alimentation (recommandée)	7-12V
Tension d'alimentation (limites)	6-20V
Broches E/S numériques	14 (dont 6 disposent d'une sortie PWM représenté par le symbole tilde)
Broches d'entrées analogiques	6 (utilisables en broches E/S numériques)
Intensité maxi disponible par broche E/S (5V)	40 mA (Attention 200mA max total entrée/sortie cumulées)
Intensité maxi disponible pour la sortie 3.3V	50 mA
Intensité maxi disponible pour la sortie 5V	500 mA max Sur port USB
Mémoire Programme Flash	32 KB (ATmega328) dont 0.5 KB sont utilisés par le bootloader (chargeur d'amorçage)
Mémoire SRAM (mémoire volatile)	2 KB (ATmega328)
Mémoire EEPROM (mémoire non volatile)	1 KB (ATmega328)
Interface	USB (permet d'alimenté la carte et de transféré les programmes vers la carte)
Alimentation externe	Jack
Dimensions	6,86 cm x 5,3 cm

Tableau (II-2) les caractéristiques de la carte ARDUINO UNO

L'utilisation du port USB est au maximum de 500mA au-delà cela peut endommager votre équipement.

II.2.4. La configuration de l'interface ARDUINO UNO.

Avant d'utiliser pour la première fois sa carte, il est nécessaire de :

- ✓ Sélectionner sa carte dans le menu Outils/Type de Carte
- ✓ Sélectionner le port série : Outils/Port

De plus, il est nécessaire de paramétrer l'emplacement des croquis qui contiendra le sketch.

II.2.5. L'environnement de développement.

Dans l'environnement de développement, il y a plusieurs boutons :

- ✓ Vérifier : Permet de vérifier et de compiler le programme.
- ✓ Téléverser: Envoie le programme à la carte ARDUINO.
- ✓ Nouveau : Crée un nouveau sketch.
- ✓ Ouvrir : Ouvre un programme.
- ✓ Enregistrer : Enregistre le programme.
- ✓ Moniteur série : Affiche une fenêtre de visualisation de donnée série.

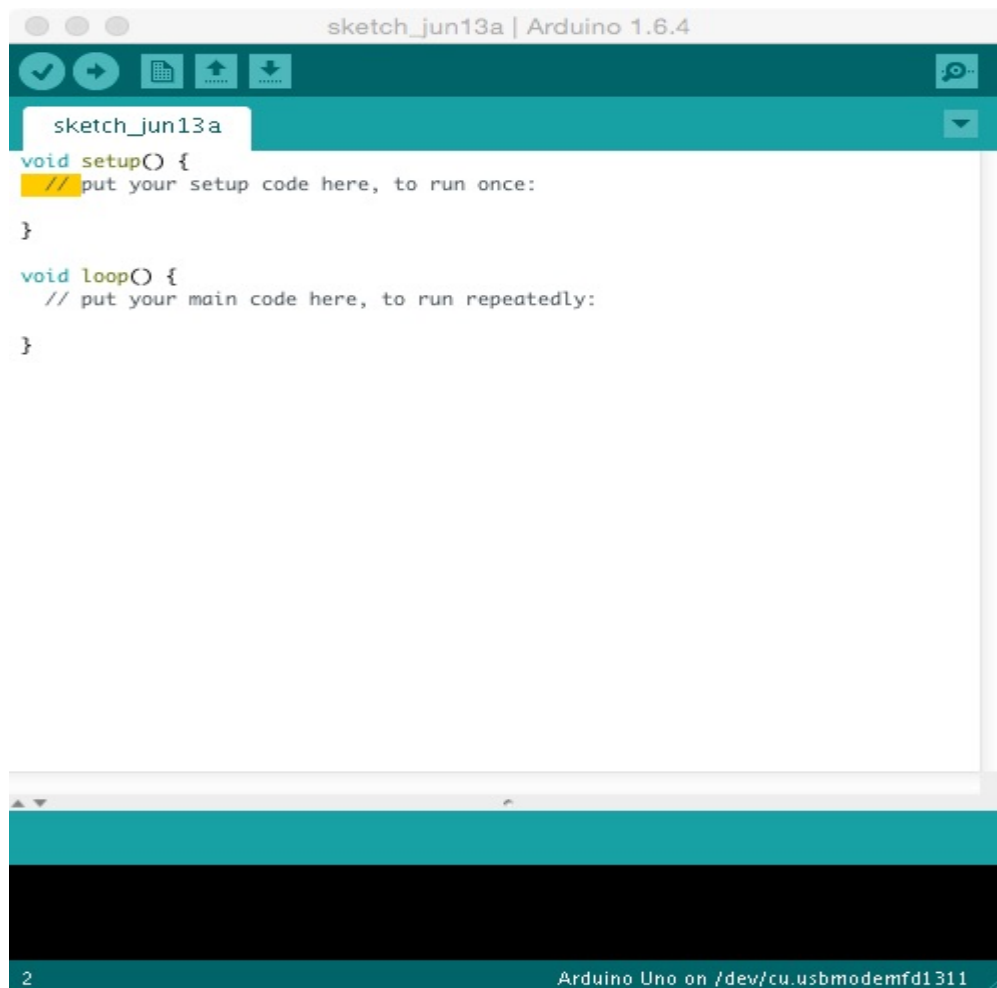


Figure (II-3) l'environnement de développement ARDUINO

II.2.6. Le test de la carte ARDUINO UNO.

- ✓ Branchez la carte ARDUINO.
- ✓ Chargez le fichier exemple BLINK d'ARDUINO.
- ✓ Téléversez le programme

Le résultat est d'allumer la diode de fonctionnement à l'alimentation de la carte, de faire clignoter les DEL Tx/Rx au téléversement et de faire clignoter La diode pin 13

II.2.7. Le montage.

Le montage met en œuvre le programme BLINK de ARDUINO, le but est de faire clignoter la diode pin 13 avec une diode rouge en parallèle, branchée avec une résistance série.

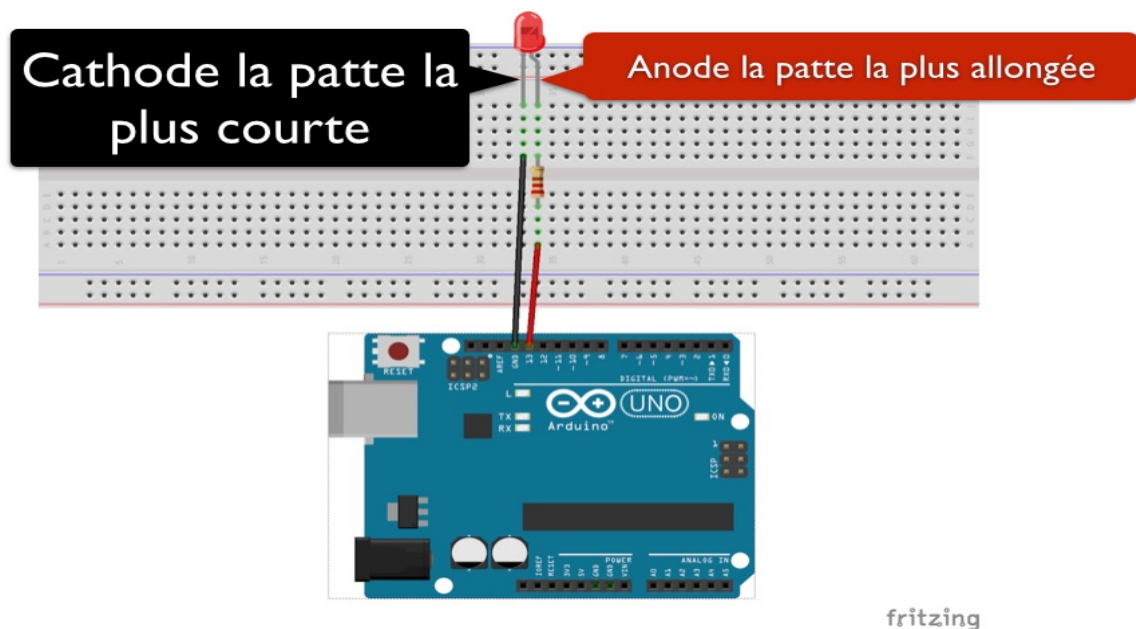


Figure (II-4) le montage de la carte ADUINO

II.3. Les avantages de la carte ARDUINO UNO.

Les principaux avantages de la technologie ARDUINO sont :

- ✓ Un accès libre aux schémas électroniques des cartes qui permettent à qui le souhaité (entreprise, amateurs de fabriquer et vendre sa propre carte à condition de ne pas utiliser le nom de la marque).

- ✓ La possibilité d'ajouter des applications au logiciel ARDUINO comme le logiciel gratuit ARDUBLOCK qui permet d'associer une interface graphique de programmation très simple.
- ✓ La connectique des cartes est standardisée, ce qui permet de connecter très facilement des cartes additionnelles compatibles appelées SHIELDS ARDUINO.
- ✓ Les cartes ARDUINO sont peu coûteuses relativement aux autres cartes de développement, les moins chères peuvent même être assemblées à la main et les préassemblées coûtent moins de 25euro.
- ✓ On n'a pas besoin d'un programmeur extérieur, grâce au BOOTLOADER on peut transférer le nouveau programme dans le microcontrôleur sans avoir à utiliser un matériel de programmation externe.
- ✓ ARDUINO fournit un environnement de développement s'appuyant sur des outils open-source, le chargement du programme dans la mémoire du microcontrôleur se fait de façon très simple par port USB.
- ✓ Multiplateforme : le logiciel ARDUINO, écrit en java, tourne sous le système d'exploitation Windows.
- ✓ L'intérêt principal des cartes ARDUINO est leur facilité de mise en œuvre.

II.4. Alimentation de la carte ARDUINO.

La carte ARDUINO UNO peut être alimentée par L'USB ou par une alimentation externe. La source est sélectionnée automatiquement. La tension d'alimentation extérieure (Hors USB) peut venir soit d'adaptateur AC-DC ou de piles.

L'adaptateur peut être connecté grâce à une prise "jack" de 2.1 mm positif au centre. Le raccordement vers un bloc de piles peut utiliser les bornes GND et VIN du connecteur d'alimentation (power).

La carte peut fonctionner à l'aide d'une tension extérieure de 7 à 12 Volts .Les broches (pins) d'alimentation sont les suivantes.

- ✓ VIN (à distinguer du 5V de la connexion USB de la connexion USB ou **5V** régulée). C'est la tension d'entrée positive lorsque la carte ARDUINO est utilisée avec une source de tension externe (à distinguer de 5V de la connexion USB ou autre source 5V régulée).

Vous pouvez alimenter la carte à l'aide de cette broche, ou, si l'alimentation est fournie par le jack d'alimentation, accéder à la tensions d'alimentation sur cette broche)

- ✓ 5V : c'est la tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (la tension régulée est obtenue grâce à un régulateur intégré dans la carte ARDUIONO).

Le 5V fourni par cette peut broche peut donc parvenir soit de la tension d'alimentation VIN via le régulateur de la carte ou bien de la connexion USB (qui fournit du 5V régulée) ou de toutes autres source d'alimentation régulée.

- ✓ 3V3 : une alimentation de 3,3V fournie par le circuit intégré FTDI (circuit intégré faisant d'adaptation du signal entre le port USB de votre ordinateur et le port série de l'ATmega) de la carte est disponible : ceci est intéressant pour certain circuits externes nécessitant cette tension ou lieu de 5V.

L'intensité maximale disponible sur cette broche est de 50 mA.

- ✓ GND : broche de masse (ou 0V).

Dès lors que l'on teste un programme sous ARDUINO, la carte est reliée vie le port USB a l'ordinateur .le câble USB sert a deux choses : Alimenter la carte mais aussi téléverser le programme que l'on teste.

Quand le programme est testé et approuvé, la carte ARDUINO peut alors être utilisée de manière autonome. Elle n'a plus besoin de l'ordinateur pour fonctionner.

II.5. protection du port USB contre la surcharge en intensité.

Un fusible réarmable ou (polyfuse) est présent sur la connexion d'alimentation 5V de la prise USB.

Toute consommation supérieure à 500mA provoque le déclenchement de ce fusible, protégeant ainsi le port USB de l'ordinateur auquel la carte est liée. Le fusible étant de type réarmable, il retrouvera son état normal quelques secondes après que la consommation excessive aura cessée. Bien que la plupart des ordinateurs aient leur propre protection interne, le fusible de la carte fournit une couche supplémentaire de protection.

Si plus de 500mA sont appliquées au port USB, le fusible de la carte coupera automatiquement la connexion jusqu'à ce que le court-circuit ou la surcharge soit stoppée.

II.6. gestion de la mémoire.

Le microcontrôleur ATmega 328 dispose de 32 KO de mémoire de programme flash (dont 0.5 KO également utilisés pour le bootloader). Il contient aussi 2 KO de mémoire vive (SRAM). Cette mémoire est généralement utilisée pour stocker les résultats temporaires lors de calculs.

Elle peut être lue et écrite à tout instant par le microcontrôleur mais son contenu est perdu dès que la carte n'est plus alimentée.

L'ATmega 328 dispose également 1 KO de mémoire EEPROM.

Le contenu de cette mémoire est accessible grâce aux fonctions de la librairie « EEPROM ».

II.7. les entrées /sorties numériques.

Chacune des 14 broches numériques de la carte UNO (numérotées de 0 à 13) peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions `PinMode ()` ; `DigitalWrite ()` et `DigitalRead ()` du langage ARDUINO. Ces broches fonctionnent en 5V. Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité et dispose d'une résistance interne « résistance de rappel » (Pull-up) (déconnectée par défaut) de 20-25 KOhms.

Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction `DigitalWrite (broche, HIGH)`.

Il ya entre ces broches celles qui ont des fonctionnalités en plus :

- ✓ Communication série :

Broches 0 (RX) et 1 (TX). Utilisées pour recevoir (RX) et transmettre (TX) les données séries de niveau TTL.

Ses broches sont connectées aux broches correspondantes du circuit intégré ATmega8U2 programmé en convertisseur USB-vers-série de la carte (composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur). On fait appel à la transmission série à travers ces broches avec l'instruction `Serial.Print()`, à condition que le câble USB soit déconnecté, sinon il va y avoir un chevauchement.

✓ Interruptions externes :

Broche 2 et 3. Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur changement de valeurs. Voir l'instruction `attachInterrupt()` pour plus de détails.

✓ Impulsion PWM (Largeur d'Impulsion Modulée).

Broches 3, 5, 6, 9, 10 et 11. Fournissent une impulsion PWM 8bits à l'aide de l'instruction `analogWrite()`.

✓ SPI (Interface Série Périphérique).

Broche 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).

Ces broches supportent la communication SPI disponible avec la librairie pour communication SPI.

Les broches SPI sont également connectées sur le connecteur ICSP.

✓ 12C.

✓ Broches 4 (SDA) et 5 (SCL), supportent les communications de protocole 12C, disponible en utilisant la librairie `Wire` /12C

✓ Ou TWO Wire.

✓ LED.

Broche 13. Il y a une LED incluse dans la carte connectée à la broche 13. Lorsque la broche est au niveau haut, la LED est allumée, lorsque la broche est au niveau bas, la LED est éteinte.

❖ Autres broches :

Il Ya deux autres broches disponibles sur la carte :

✓ AREF :

Tension de référence pour les entrées analogiques (si différent du 5V), utilisée avec l'instruction `analogReference()`.

Elle s'utilise pour réduire.

✓ Reset :

Mettre cette broche au niveau bas entraine la réinitialisation du microcontrôleur. Typiquement, cette broche est utilisée pour ajouter un bouton de réinitialisation sur le circuit qui bloque celui présent sur la carte.

La figure suivante montre les différentes broches de la carte ARDUINO UNO.

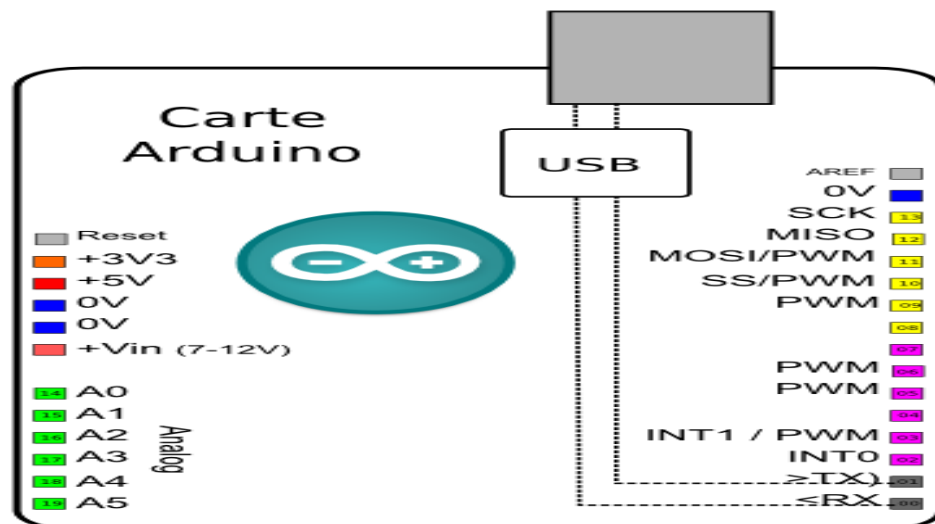


Figure (II-5) les différentes broches de la carte ARDUINO UNO

II.8. communication avec l'extérieure.

La carte ARDUINO dispose de toutes une série de facilités pour communiquer avec un ordinateur, avec une autre carte ARDUINO, ou avec d'autres microcontrôleurs.

L'Atmega 328 dispose d'une UART (Universal Asynchronous Receiver Transmitter ou émetteur-récepteur universel asynchrone en français) pour une communication série de niveau TTL (5V) et qui est disponible sur les broches 0 (RX) et 1 (TX).

Un circuit intégré Atmega 8U2 sur la carte assure la connexion entre cette communication série et le port USB de l'ordinateur et apparaît comme un port COM virtuel pour les logiciels de l'ordinateur. Le code utilisé pour programmer l'Atmega 8U2 utilise le driver standard USB COM, et aucun autre driver externe n'est nécessaire.

Le logiciel ARDUINO inclut une fenêtre terminal série (ou moniteur série) sur l'ordinateur et qui permet d'envoyer des textes simples depuis et vers la carte ARDUINO.

Des LEDS RX et TX sur la carte clignotent lorsque les données sont transmises via le circuit intégré USB vers-série et la connexion USB vers l'ordinateur (mais pas pour les communications série sur les broches 0 et 1).

Une librairie série logicielle permet également la communication série (limitée cependant) sur n'importe quelle broche numérique de la carte UNO.

L'Atmega 328 supporte également la communication par protocole 12C et SPI :

- ✓ Le logiciel ARDUINO inclut la librairie WIRE qui simplifie l'utilisation du bus 12C.
- ✓ Pour utiliser la communication SPI (Interface Série Périphérique), la librairie pour communication SPI est disponible, il suffit de la faire inclure dans le programme au niveau du logiciel ARDUINO lors de la programmation.

❖ Dimensions de la carte :

Les longueurs et largeurs maximales de la UNO sont respectivement 6.86 CM et 5.33 CM, avec le connecteur USB et le connecteur d'alimentation jack s'étendant au-delà des dimensions de la carte.

Quatre trous de vis permettent à la carte d'être fixée sur une surface ou dans un boîtier (pour l'embarquer sur un système).

Noter que la distance entre les broches 7 et 8 est de 0.16 pouces et 0.1 pouces séparant les autres broches.

II.9. Microcontrôleur ATMEL ATmega328

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)

Plusieurs broches multifonctions : certaines broches peuvent avoir plusieurs fonctions différentes, choisies par programmation. Elles ont alors plusieurs noms sur le brochage voir la figure (II-6)

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.

Watchdog Timer programmable.

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)

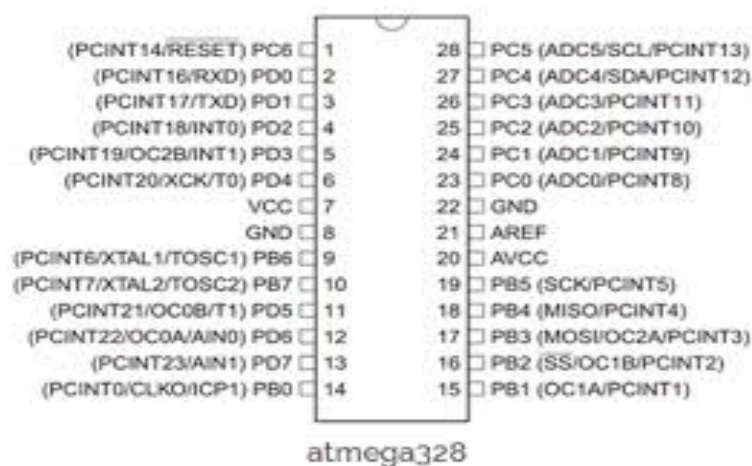


Figure (II-6) Atmega328

I.10. Discussion

Nous avons dans ce chapitre présenté les caractéristiques de la carte Arduino UNO. Pour ce faire, nous avons décrit la partie matérielle et logicielle. Dans la partie matérielle, nous avons mis en évidence les entrées/sorties analogiques et numériques de la carte et ses constituants internes. Le logiciel utilisé pour la programmer a été présenté.

Chapitre III

Conception et programmation

Chapitre III : conception et programmation de la maquette

III.1. Préambule :

Après les généralités sur les ascenseurs et l'étude de la carte ARDUINO UNO, respectivement au premier et deuxième chapitre, nous présenterons dans ce chapitre la réalisation de la carte de commande et les étapes de programmation de notre maquette d'ascenseur.

Dans un premier temps, nous allons donner la partie théorique qui concerne la description des composants utilisés et leurs caractéristiques suivie de la partie pratique axée sur la réalisation de circuit et la conception des cartes par logiciels.

III.2. Partie théorique :

Les composants utilisés pour notre réalisation à savoir un ascenseur :

III.2.1. Le moteur :

Le moteur qu'on a utilisé dans notre réalisation est le moteur à courant continue parce qu'il présente des facilités d'utilisation qui se traduisent par les caractéristiques suivantes :

Couple nominal 1,6 Ncm

Puissance utile de 4,9 à 5,3 W

Tension nominale 12 ou 24 Vdc

Vitesse nominale de 3150 à 2900 tr/mn

Courant nominal de 0,45 à 0,95 A

Constante de vitesse de 1281 à 1375

Constante de couple de 2 à 4,2 (Ncm / A)

Inertie (gcm²) 14

Masse 0,1 Kg

Longueur L 50 (mm)

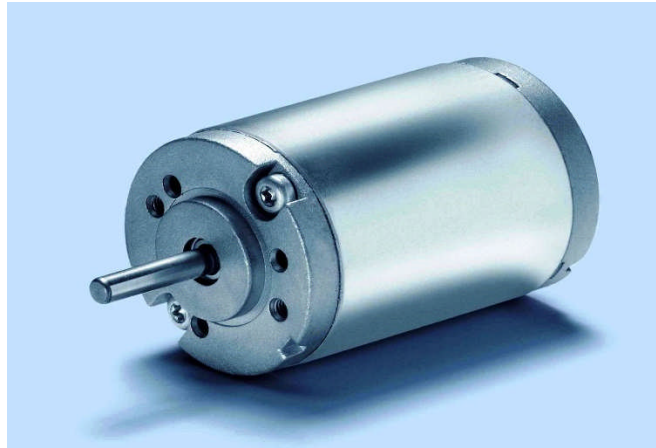


Figure (III-1) moteur à courant continu.

Comme on le sait le moteur à courant continu a un seule sens de rotation, pour avoir deux sens de rotation nous avons utilisé un composant très répondu dans le monde de l'électronique appelé << pont H >>.

III.2.2. Le pont en H :

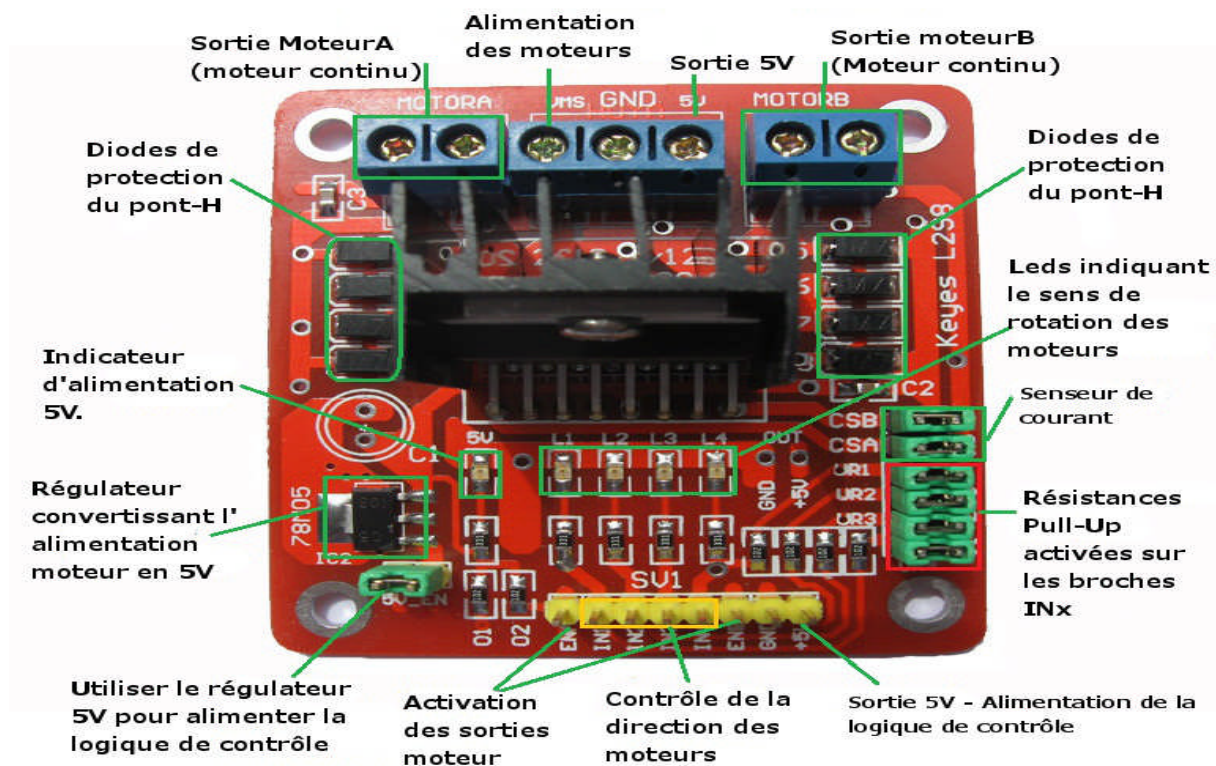


Figure (III-2) pont en H.

Chapitre III : conception et programmation de la maquette

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 d'où le nom. Cette structure se trouve dans plusieurs applications de l'électronique de puissance incluant le contrôle des moteurs, les convertisseurs et les hacheurs. 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 compose.

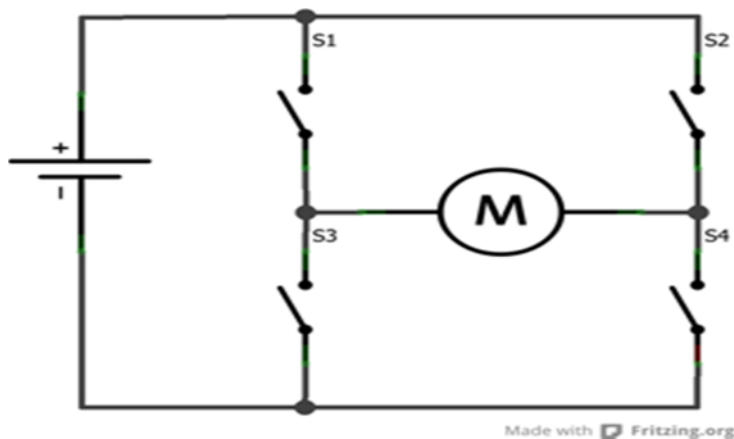


Figure (III-3) schéma interne d'un pont en H.

III.2.2.1. Principe de fonctionnement :

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

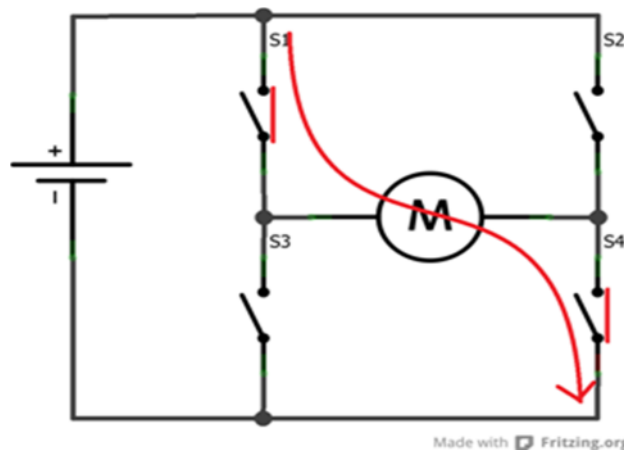


Figure (III-4) fonctionnement dans le sens horaire.

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

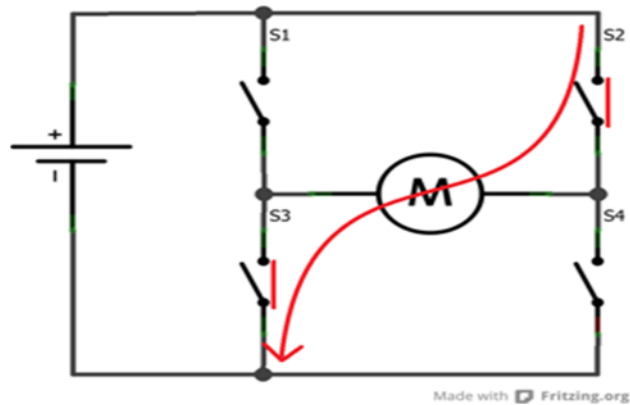


Figure (III-5) fonctionnement dans le sens antihoraire.

On remarque que les transistors fonctionnent deux par deux, en effet, si on ferme juste un seul et laisse ouvert les trois autres le courant n'a nulle part où aller et rien ne se passe, le moteur est en roue libre. Maintenant 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.

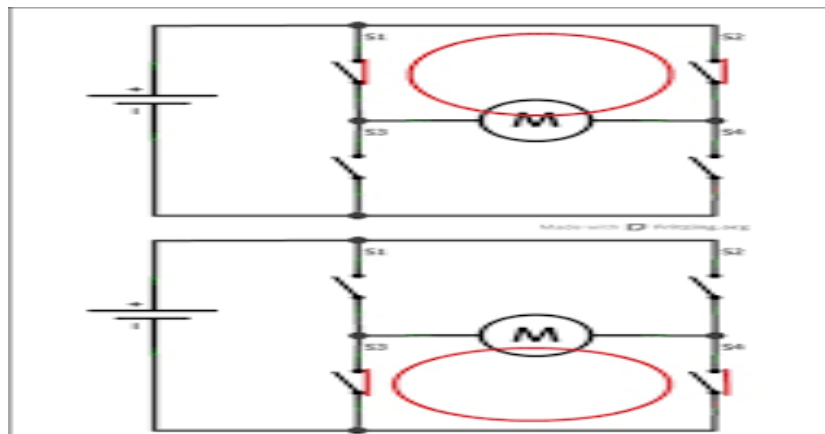


Figure (III-6) freinage avec S1 et S2 et avec S3 et S4 respectivement.

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.

Afin d'inviter les branchements des transistors et leur logique de contrôle, des composants "clés en main" ont été développés et produits. Nous allons maintenant présenter un d'entre eux que nous retrouvons les shields moteur ARDUINO qui est le **L298N** :

III.2.2.2. Le 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.

Il peut contrôler deux moteurs à courant continu ou un moteur pas-à-pas 4 fils 2 phases. Il est conçu pour supporter des tensions plus élevées.

Caractéristiques :

- ✓ Léger, petit
- ✓ Des capacités hors-pair pour contrôle moteur
- ✓ Diodes de protections
- ✓ Un dissipateur (pour dissiper la chaleur en cas de forte charge)
- ✓ Un sélecteur pour sélectionner la source d'alimentation
- ✓ 4 Sélecteurs pour les résistances pull up
- ✓ Sortie pour 2 moteurs continu/ 1 moteur pas-à-pas (4 bobines, deux phases)
- ✓ Indicateur LED pour sens de rotation moteur
- ✓ Indicateur LED pour alimentation 5V
- ✓ 4 trous de fixation standard

Brochage :

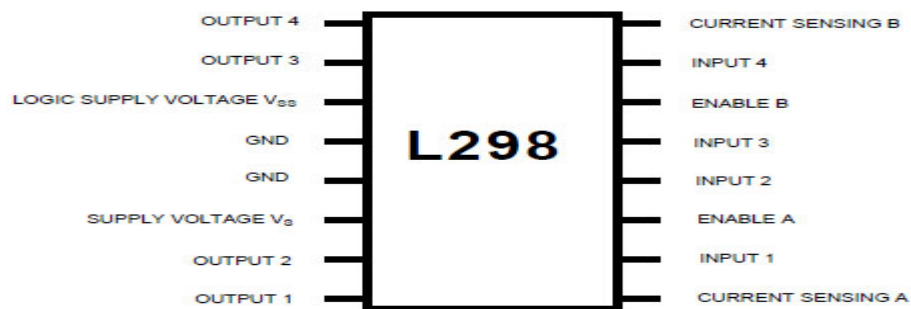


Figure (III-7) configuration des broches du L298N.

III.2.3. Afficheur 7 segments :

III.2.3.1. Description :

Comme son nom l'indique, l'afficheur 7 segments possède 7 segments, c'est un composant qui permet d'afficher les chiffres de 0 à 9. Ce type d'afficheur très présent sur les calculatrices et les montres à affichage numérique : les caractères (des chiffres, bien que quelques lettres soient utilisées pour l'affichage hexadécimal) s'écrivent en allumant ou en éteignant des segments, au nombre de sept. Quand les 7 segments sont allumés, on obtient le chiffre 8.

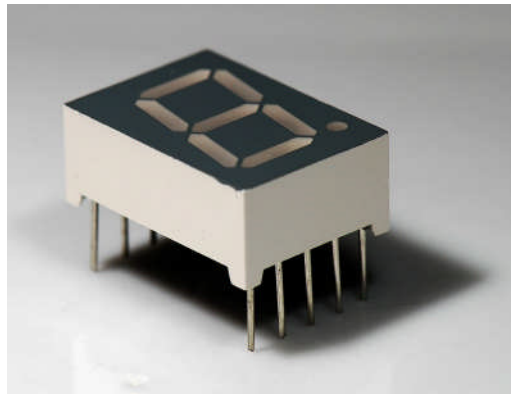


Figure (III-8) afficheur 7 segments.

III.2.3.2. Fonctionnement :

Un segment est en fait un LED plate, et les 7 sont arrangées en forme de huit. Pour disposer les chiffres il faut allumer certain segment, et en laisse d'autre éteints. C'est segment pour être identifié facilement sont associés a des lettres.

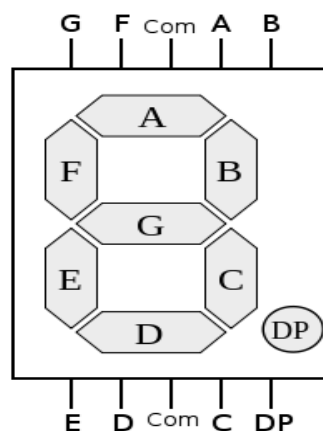


Figure (III-9) afficheur 7 segments identifiés avec des lettres.

Chapitre III : conception et programmation de la maquette

Si l'on s'intéresse de plus près au brochage, on s'aperçoit que chaque LED possède une de ses extrémités connectée à une broche commune. Lorsque toutes les anodes des LED sont mises en commun on parle d'un 7 segment à anode commune, et cathode commune lorsque les LED ont leur cathode en commun. Les boîtiers de ces équipements sont donc équipés de 10 broches (8 broches+ 1 commune + alimentation+ masse).

Afficheur à anode commune : toutes les 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 : toutes les cathodes sont reliées et connectées au potentiel bas. La commande du segment se fait par son anode mise au potentiel haut.

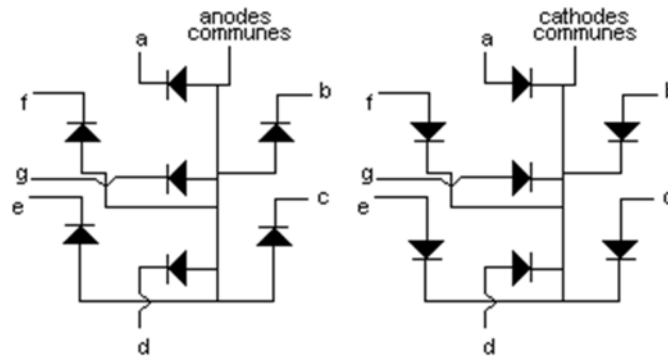
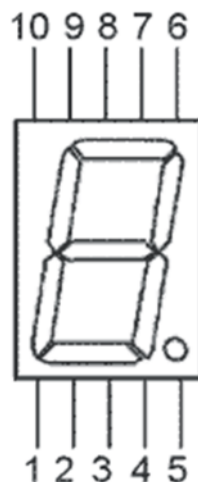


Figure (III-10) types d'afficheur 7 segments.

III.2.3.3. Branchement de l'afficheur :



Chapitre III : conception et programmation de la maquette

Voici la signification des différentes broches et l'exemple de branchement d'un afficheur a anode commune :

1. LED de la cathode E ;
2. LED de la cathode D ;
3. Anode commune des LED ;
4. LED de la cathode C ;
5. Le point décimal ;
6. LED de la cathode B ;
7. LED de la cathode A ;
8. Anodes commune des LED ;
9. LED de la cathode F ;
10. LED de la cathode G ;

III.3. partie pratique :

III.3.1. Partie logiciel :

Avant la réalisation pratique de notre maquette, nous ferons la conception et la simulation de ces dernières, pour ce faire nous allons utiliser le logiciel PROTEUS.

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

Ce logiciel comprend deux principaux modules : ISIS et ARES.

ISIS :

Le module ISIS de proteus est principalement utilisé pour éditer un schéma structurel d'un circuit électronique (assemblage de composants électroniques dont on fixe les valeurs et les références) reliés par des connexions électrique (fils). Par ailleurs, le logiciel permet également de simuler les différents types de montages ce qui permet de déceler certaines erreurs dès l'étape de conception.

ARES :

Le module ARES est un outil d'édition et de routage qui complète parfaitement ISIS. Un schéma électrique réaliser sur ISIS peut alors être importé facilement sur PCB de la carte électronique.

Chapitre III : conception et programmation de la maquette

III.3.1.1. Schéma synoptique de la carte de commande.

Premièrement en alimente le système, on appuie sur le bouton d'appelle le microcontrôleur reçoit le message, un affichage indique l'état du système, le pont H permet de faire tourner le moteur vers le HAUT ou vers le BAS.

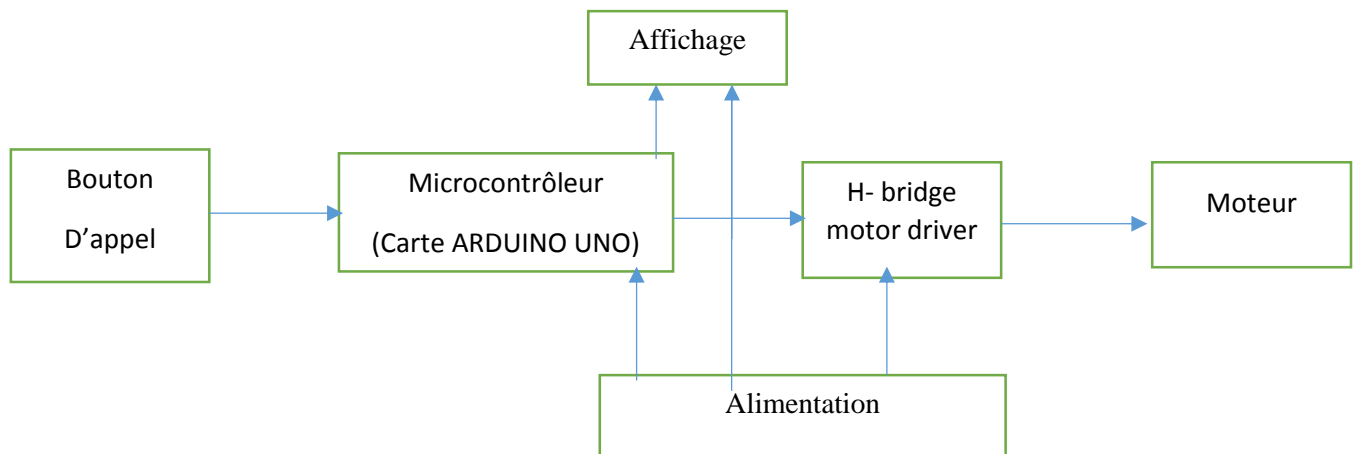


Figure (III-11) schéma synoptique de la carte de commande.

III.3.1.2. branchement des boutons poussoirs :

La figure ci-dessous représente le schéma de branchement des boutons poussoirs à la carte ARDUINO sur ISIS, pour cela on a utilisé les pines A1, A2 et A3.

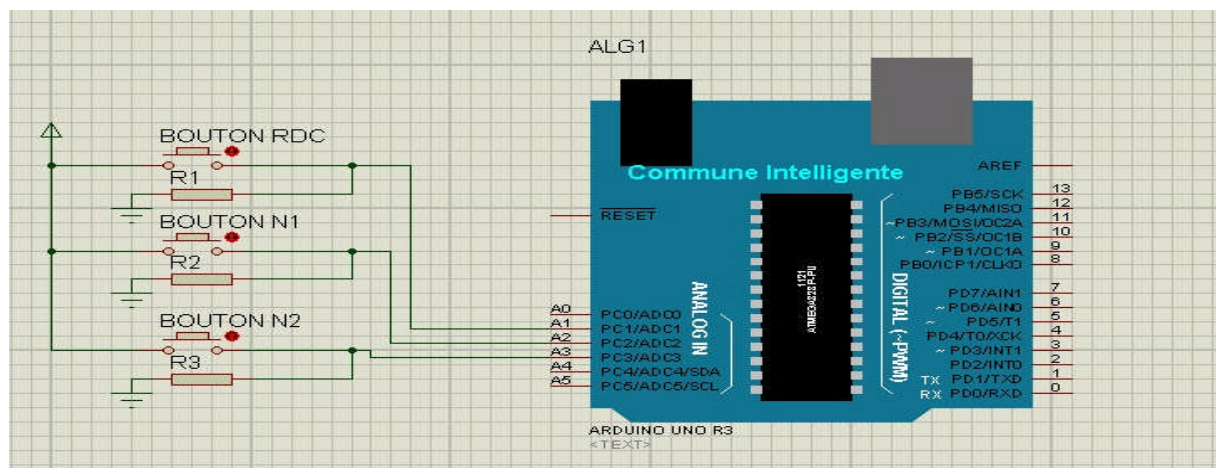


Figure (III-12) branchement des boutons poussoirs.

Chapitre III : conception et programmation de la maquette

III.3.1.3. branchement de l'afficheur 7 segments :

Cette figure représente le branchement de l'afficheur 7 segment a la carte ARDUINO, pour cela on a utilisé les pines 7, 8, 9, 10, 11, 12 et 13. Et la résistance de protection de 220 Ohm.

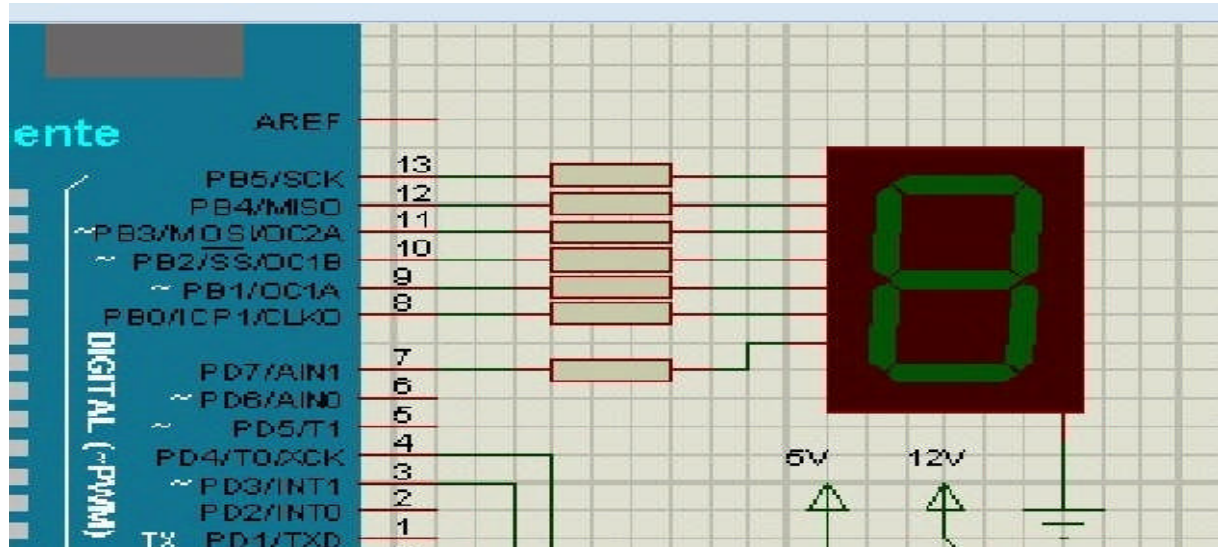


Figure (III-13) branchement de l'afficheur.

III.3.1.4. Schéma de commande globale :

La conception de cette maquette sur ISIS est faite selon nos besoins d'alimentation, le moteur doit être alimenté par 12 V, la carte ARDUINO par 9V.

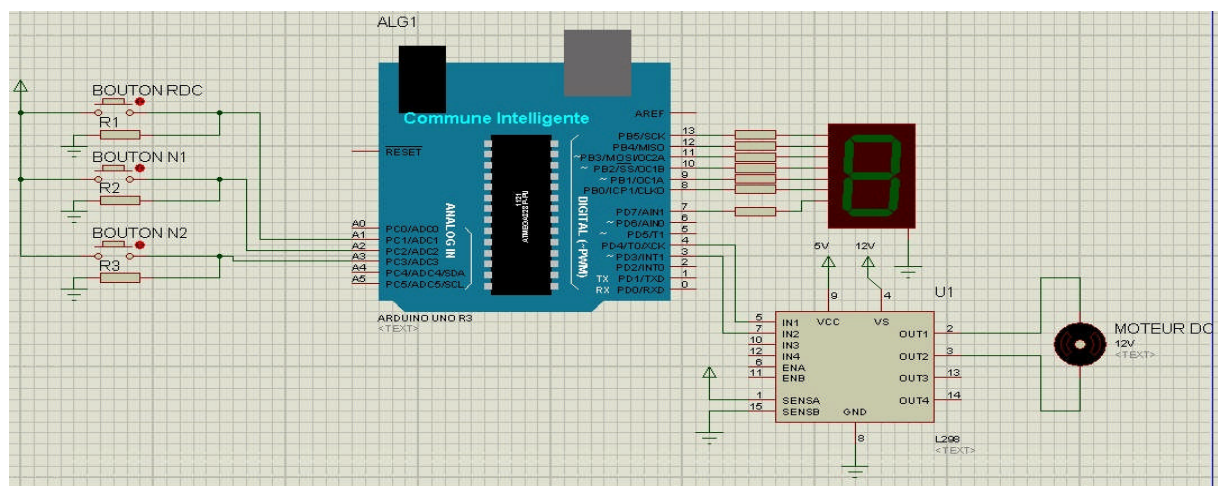


Figure (III-14) le branchement global de la maquette.

III.3.2. partie Programmation :

III.3.2.1. réalisation du programme :

Vue la complexité de notre programme nous l'avons partagé en un ensemble de sous-programme, qu'on peut résumer dans l'organigramme suivant :

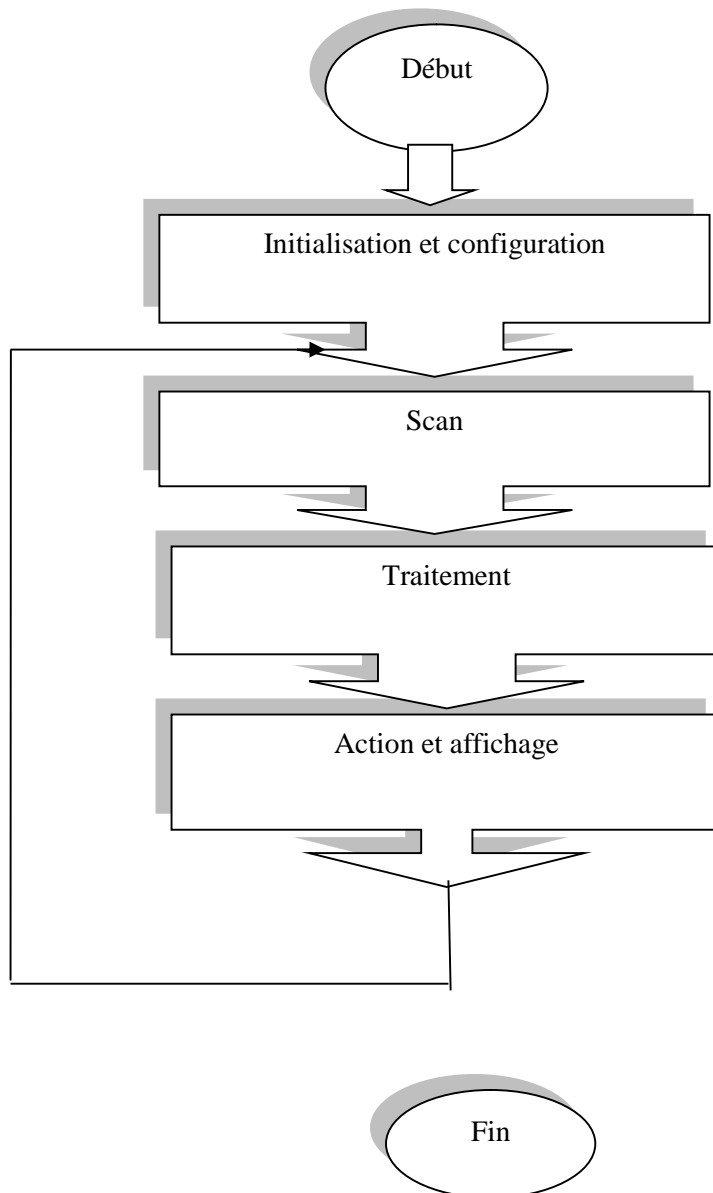


Figure (III-15) organigramme du système.

Chapitre III : conception et programmation de la maquette

A. Sous-programme 1 : “initialisation et configuration”

Cette partie consiste à initialiser toutes les variables, et configurer les entrées et les sorties.

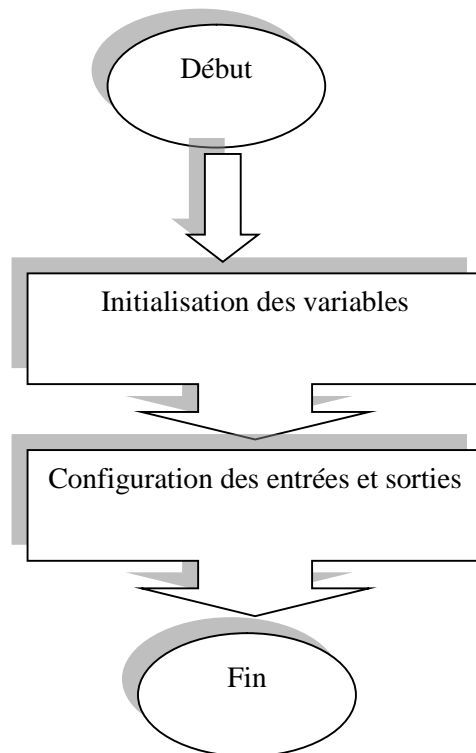


Figure (III-16) sous-programme 1.

B. Sous-programme 2 : "scanne"

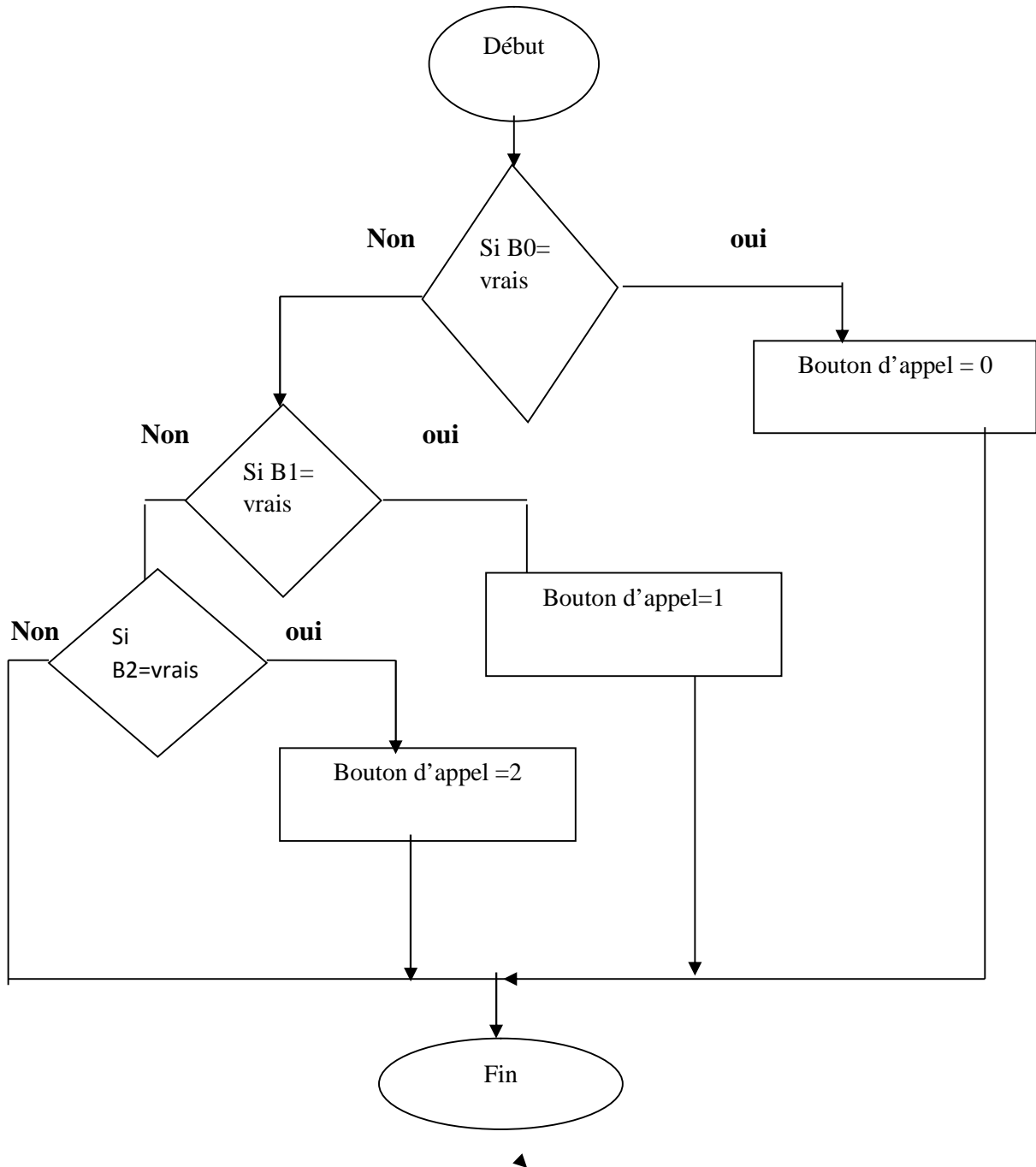


Figure (III-17) sous-programme 2

C. Sous-programme 3 : "traitement "

Sur le sous-programme suivant on remarque que : si la position de la cabine supérieur à la position d'appel, le moteur activé ver le BAS veut dire tourne dans le sens ou la cabine décent, si non la cabine monte.

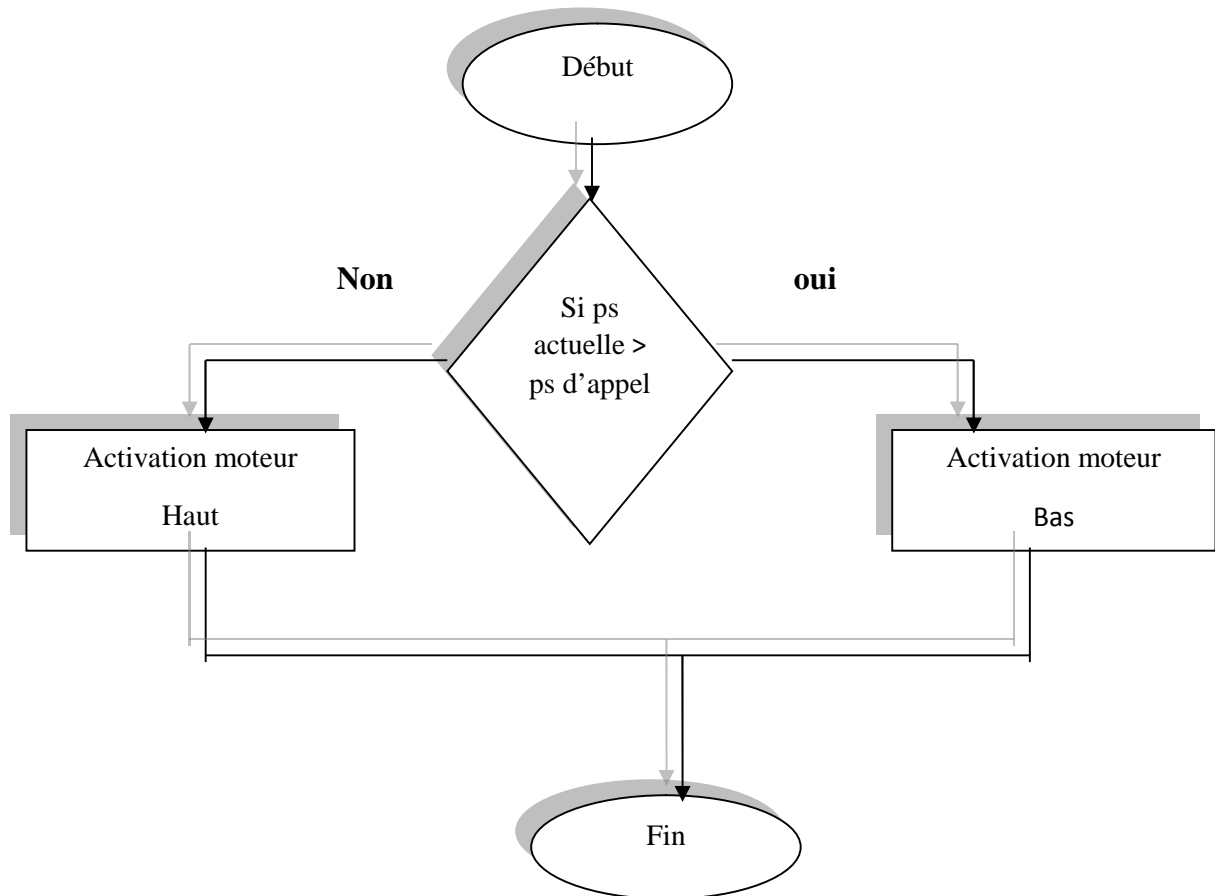


Figure (III-18) sous-programme 3.

D. Sous-programme 4 : ‘action et affichage ‘

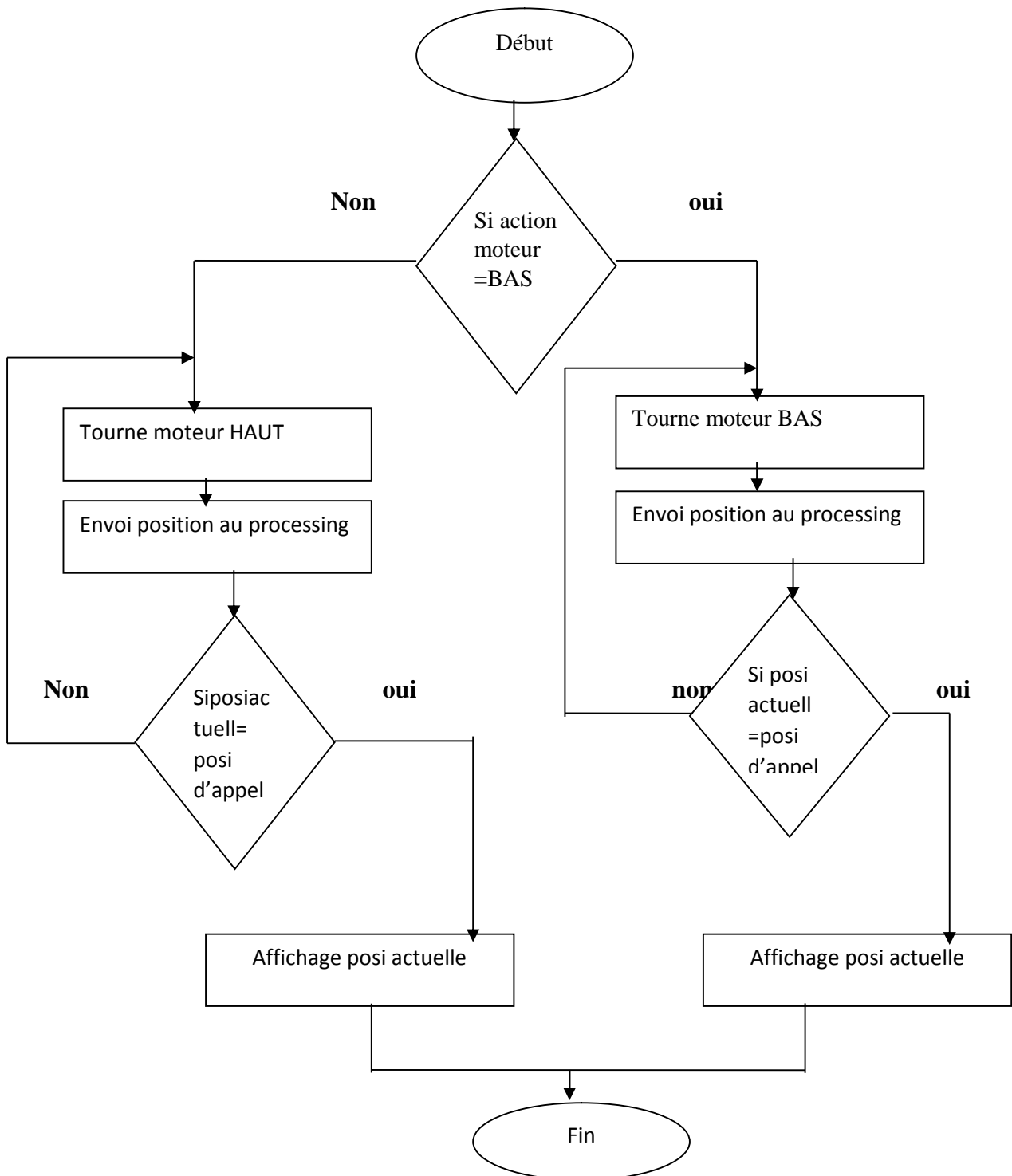


Figure (III-19) sous-programme.

Chapitre III : conception et programmation de la maquette

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 les deux sens ce qui permet de déplacer la cabine vers le haut et vers le bas.

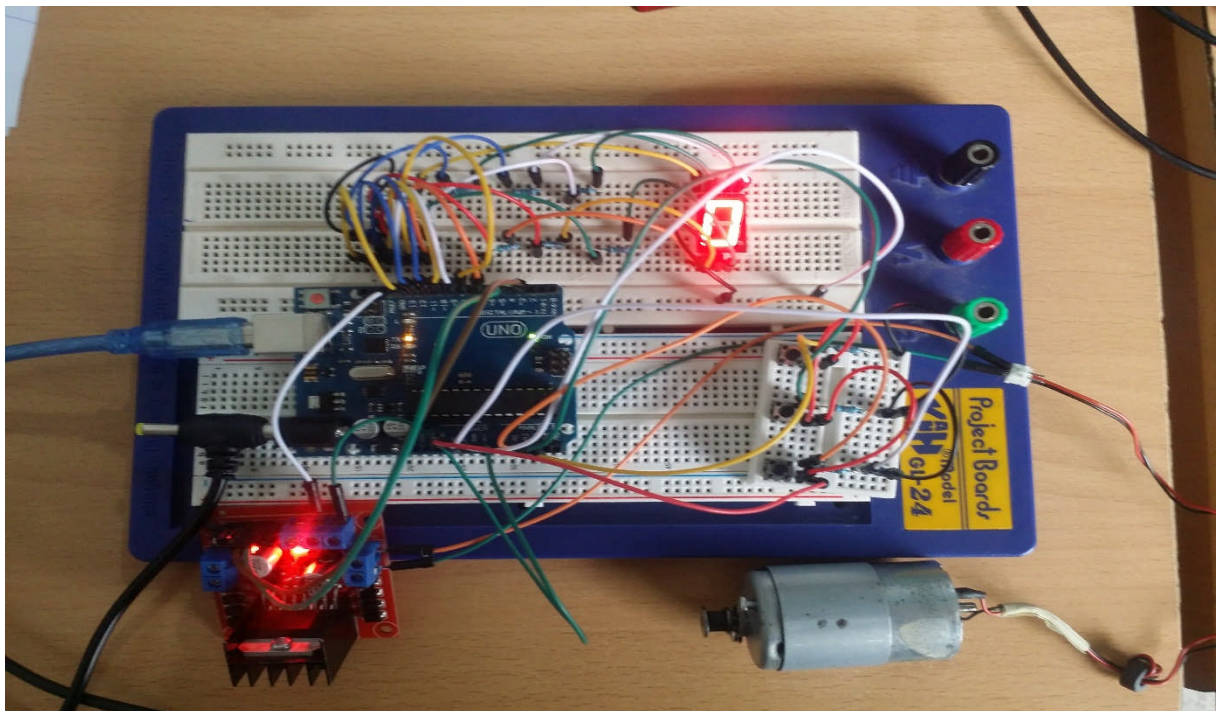


Figure (III-20) maquette de l'ascenseur

III.4. Discussion :

Dans ce chapitre nous avons donné la description des composants a utilisé dans la réalisation de notre maquette, nous avons aussi développé un organigramme qui va nous aider à la conception de la partie logiciel.

Chapitre VI

Test et validation

VI.1. Préambule :

Ce chapitre ce n'est qu'une suite du chapitre précédent, son objectif est de tester la maquette d'ascenseur avec le logiciel de programmation processing.

VI.2. Définition de processing :

Le Processing possède la particularité d'utiliser des instructions informatiques pour dessiner, réaliser des animations en 2 ou 3 dimensions, créer des œuvres sonores et visuelles, concevoir des objets communicants qui interagissent avec leur environnement.

Le Processing permet également de programmer des circuits électroniques qui interagissent avec le milieu qui les entoure. Connectés à des capteurs sonores, thermiques, de mouvement, ces circuits électroniques peu coûteux, dénommés microcontrôleurs.

VI.3. Interface de precessing :

On trouve plus précisément les éléments suivants dans l'interface :

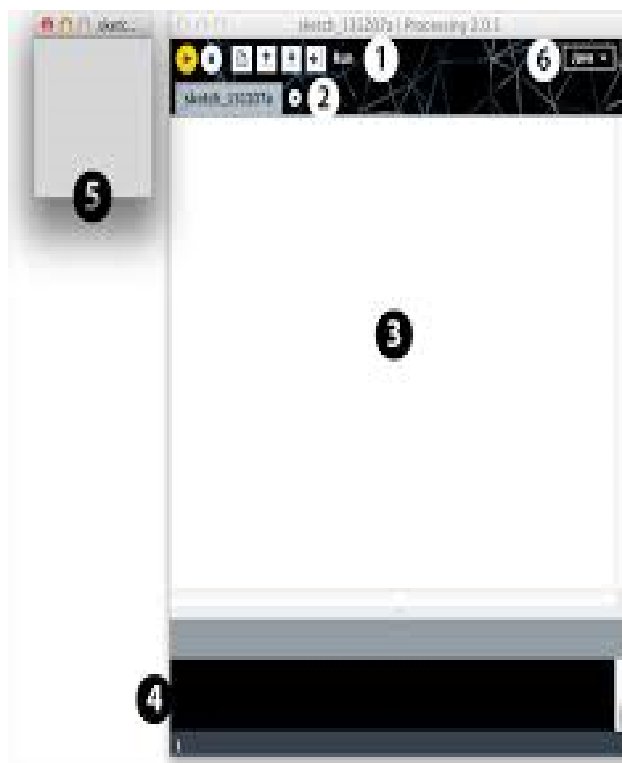


Figure (VI-1) : interface de processing.

1. Barre d'actions.

2. Barre d'onglets.

3. Zone d'édition (pour y saisir votre programme).
4. Console (destinée aux tests et messages d'erreur).
5. Fenêtre de visualisation (espace de dessin).
6. Liste déroulante pour les modes.

VI.4. L'espace de dessin :

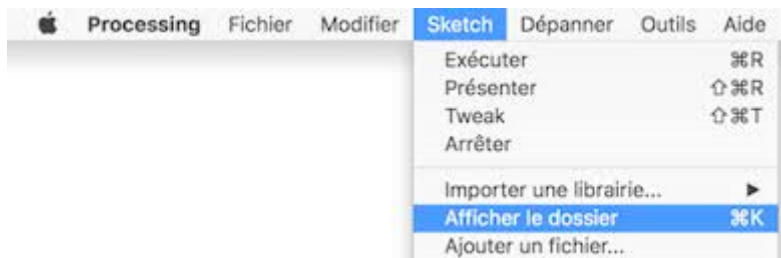
L'espace de dessin constitue l'espace de représentation proprement dit. Cette fenêtre de visualisation affichera nos réalisations dans Processing en 2 ou 3 dimensions.

Cet espace est créé par l'instruction `<<size()>>` qui prend deux arguments :

`size (largeur, hauteur) ;`

Pour cela on a suivi les étapes suivantes :

- Premièrement, nous avons commencé par chercher une image adéquate pour expliquer au mieux le déroulement de notre simulation puis nous l'avons importé dans le sketch et nous l'avons ensuite sauvegardé dans le Dossier " mes document " sur notre ordinateur.



- Puis nous avons glissé le fichier d'image sur la fenêtre processing.
- Processing copie automatiquement cette image dans un dossier nome `<<data>>`.
- Nous avons donné à notre espace de dessin une taille plus grande que notre image.

Size (600,490) ;

COUPE=LoadImage ("COUPE, png) ;

CABINE=LoadImage ("CABINE, png") ;

- Puis nous avons cliqué sur le bouton `<< run >>`, pour visualiser cet espace.

VI.5. Teste de notre maquette sur processing :

Dans cette partie le programme ARDUINO affiche dans l'afficheur 7 segments, la position demander par l'utilisateur est de transmettre la position exacte de la cabine au programme de simulation processing.

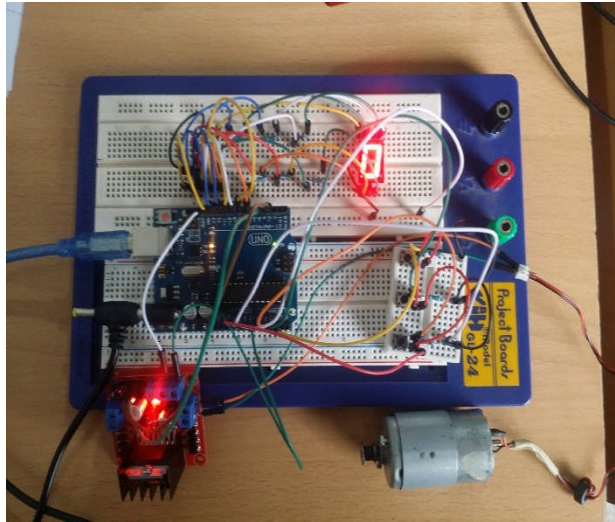
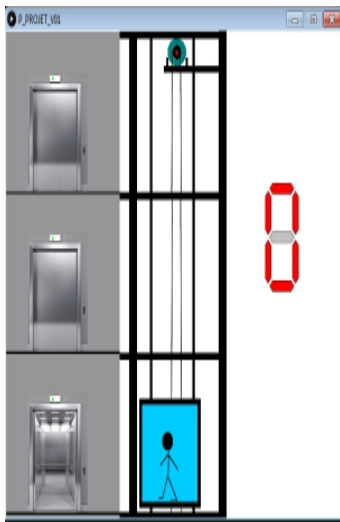


Figure (VI-2) la position 0 (rai de chaussée).

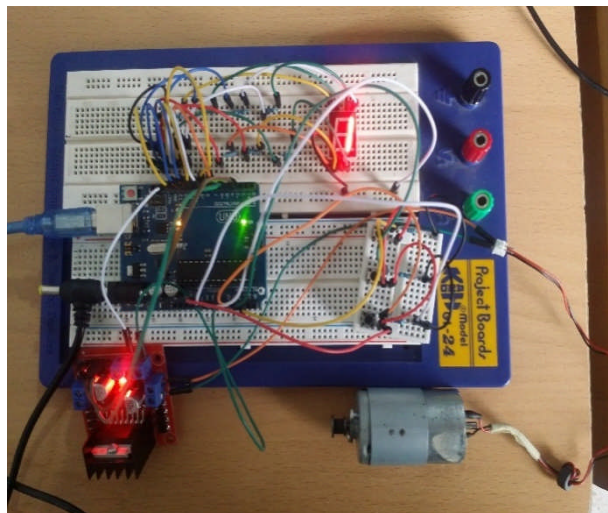
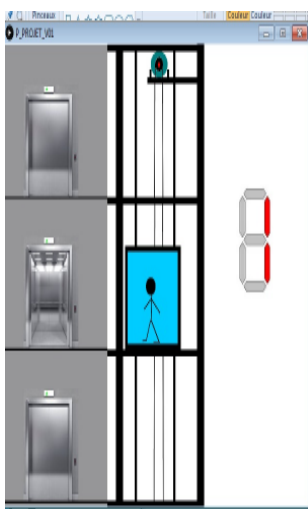
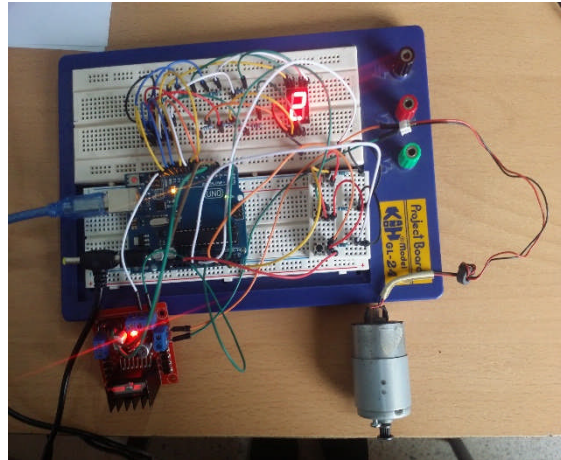
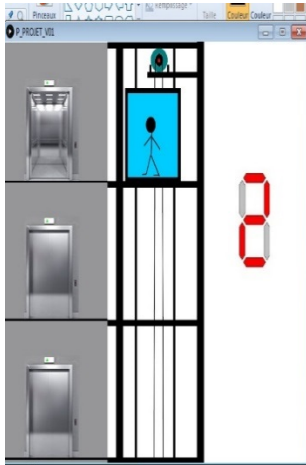


Figure (VI-3) position niveau 1.**Figure (VI-4) position niveau 2.**

VI.6. Discussion :

Dans ce chapitre nous avons pu tester et simuler sur logiciel processing le grosso modo de notre travail.

Conclusion générale :

Au terme de ce projet, nous avons pu réaliser un ascenseur commandé par Arduino. Pour ce faire, nous avons utilisé un moteur à courant continu, une carte Arduino et un afficheur 7 segments. Nous avons ajouté des boutons poussoir utilisés pour sélectionner le niveau auquel l'ascenseur doit s'arrêter. L'afficheur indique le niveau sélectionné. Le moteur est commandé par Arduino interfacé par le Sheeld.

Pour vérifier son fonctionnement, nous avons réalisé des tests sur l'application. Nous avons pu montrer les déplacements de l'ascenseur selon le niveau sélectionné. En fonction du niveau où se trouve l'ascenseur est le bouton appuyé le moteur tourne le sens adéquat. Par conséquent, l'application a bien fonctionné.

Pour perfectionner notre application, il est intéressant d'ajouter un système de sécurité. Aussi,

Bibliographie :

[1].<https://knowledge.parcours-performance.com/utiliser-arduino-uno>

[2].FR .[wikipedia.org /wiki/Acsenseur](https://fr.wikipedia.org/wiki/Acsenseur), Site Internet, consulté en septembre 2018

[3].M.Lbouhmadi, J.Laayoun « Etude d'ascenseur commandé par automate programmable »projet de fin d'étude, Université sidi Mohammed ben Abdallah, Fès, Maroc 2007

[4].<http://www.electronique.fr/>,consulté en septembre 2018

[5].www.alldatasheet.com

[6].guide d'utilisateur de logiciel proteus, consulté en septembre 2018

[7].Huayong Yang, wei Sun ,and Bing Xu New "Investigation in Energy Regeneration of Hydraulic Elevator "IEEE/ASME Transaction on mechatronics,vol.12,no.5,Octobre 2007

[8].Sandor Markon, Hajime Kita, Hiroshi Kise and Thomas Bartz-Beielstein"Control of traffic Systems in Buildings"Springer-verlag London Limited 2006