République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la RechercheScientifique
Université Mouloud Mammeri de TIZI OUZOU

FACULTE DE GENIE ELECTRIQUE ET INFORMATIQUR
DEPARTEMENT ELECTRONIQUE
SPECIALTE : ELECTRONIQUE INDUSTRIELLE

Mémoire de fin d'étude sous thème :

CONCEPTION ET REALISATION D'UN ASCENSEUR A BASE D'UNE CARTE ARDUINO

PROPOSER PAR :
M^R. ZIRMI RACHID

REALISER PAR :
M^R. AIT GHERBI ALI
M^R. MOUFFEK HAKIM

Année universitaire : 2023/2024

. Remerciements

Avant tout, nous remercions le bon Dieu le tout puissant de nous avoir donné le courage, la volonté et la patience durant toutes les années d'études et que grâce à lui ce travail a pu être réalisé.

Nos vifs remerciements vont en premier lieu, à nos chers parents de nous avoir aidé pour arriver au terme de ce travail. Comme nous tenons à exprimer tous nos reconnaissances et nos gratitudes à notre promoteur Mr ZIRMI, de nous avoir encadré, suivi et orienté tout au long de notre travail. Nous remercions d'avance, les membres de jury d'accepter d'examiner notre travail.

A travers ce mémoire, nous adressons nos reconnaissances à tous nos enseignants qui ont contribué à notre parcours d'études depuis la première classe du primaire jusqu'à aujourd'hui. En fin, nous tenons à remercier tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

Décédas

Du profond de mon cœur, je dédié ce travail à ceux qui me sont chers Je dédié ce travail à mon très cher père, tu as toujours été pour moi un exemple du père respectueux et grâce à toi j'ai appris le sens du travail et de la responsabilité. Ce travail est le fruit de tous les sacrifices que tu as déployés pour mon éducation « Paix à ton âme »

Particulièrement à ma très chère maman qui a toujours été là pour moi, je vous remercie pour le soutien et l'amour que vous me portez depuis mon enfance

Est à tous membres de ma famille : MOURAD, NACER, YACIN, SAFIA,

YASSMINE, SABRINA, GHANIA, NACERA, NELYA, LEO, MOUHAND, KARIM.

Vous êtes un sacré symbole de fraternité et de sincère attachement.

Sans oublier mon cher ami (es) :

CHERIF, JUGURTHA, AMAR, RAMDHANE, BELAID, IDIR, MADJID, FERHAT, »

pour leur aide et encouragement à bien finir ce travail.

Et sans oublie mon cher binôme « AIT GHERBI ALI »

HAKIM

Décédas

Du fond de mon cœur, je dédie ce travail à ceux qui me sont chers.

Je dédie en premier lieu ce travail à mon très cher père, qui a toujours été pour moi un modèle de respect et de dévouement. Grâce à toi, j'ai appris le sens du travail et de la responsabilité. Ce travail est le fruit de tous les sacrifices que tu as consentis pour mon éducation.

Je tiens également à remercier ma très chère maman, qui a toujours été présente pour moi. Merci pour tout le soutien et l'amour que tu m'as donnés depuis mon enfance.

Je dédie aussi ce travail à tous les membres de ma famille, en particulier **BOUALEME** et **THINHINANE**.

Une mention spéciale pour la nouvelle naissance de la famille, **MASSILVA**, à qui je souhaite une grande réussite.

Vous êtes un symbole précieux de fraternité et de sincérité.

Je n'oublie pas non plus mes chers amis : **Nassim, Massy, Mourad, Yazid, Yacine, Madjid**, pour leur aide et leurs encouragements tout au long de ce travail.

Enfin, une pensée spéciale pour mon cher binôme **Mouffek Hakim**, avec qui j'ai partagé ce parcours.

ALI

Liste des figures :

Figure 1.1: Arduino 2
Figure 1.2 : Les composants d'Arduino 3
Figure 1.3: Le microcontrôleur 4
Figure 1.4 : Schéma fonctionnel d'un microcontrôleur 6
Figure 1.5 : Les entrées/sorties du microcontrôleur 8
Figure 1.6 : évolution d'un signal analogique en fonction du temps 9
Figure 1.7 : Graphe d'un signale numérique 10
Figure 1.8 : Durée d'impulsion et la période dans un signal PWM 11
Figure 1.9 : alimentation d'UNO par une batterie de 9v 12
Figure 1.10 : port série de la carte UNO. 13
Figure 1.11 : icone de l'IDE dans le bureau Windows 14
Figure 1.12 : la fenêtre de l'IDE vide 15
Figure 1.13: la barre de menu 16
Figure 1. 14 : La barre d'icônes 16
Figure 1.15 : La zone des onglets 16
Figure 1.16 : Page de sketch pour saisir le code source 17
Figure 1.17 : La ligne d'état 17
Figure 1.18 : code source d'un sketch Arduino 20
Figure 1.19: ARDUINO UNO 22
Figure 1.20 : Arduino Méga 2560 24
Figure 1.22 : Arduino Leonardo 26
Figure 1.21 : Arduino Esplora 27
Figure 1.23 : Boarduino V2.0 29
Figure 1.24 : Arduino Nano 31
Figure 1.25 : Arduino LilyPad 32
Figure 1.26 : Arduino Due 34
Figure 1.27 : Arduino Yun 36
Figure 2.1 : Composant de l'ascenseur hydraulique 42
Figure 2.2: ascenseur hydraulique 43
Figure 2.3: Les éléments d'un ascenseur 46
Figure 3.1: Module L298N 51

Figure :3.2Les ondes infrarouges 52
Figure 3.3 : Boutons poussoir NO et NF 53
Figure 3.4 :Afficheur LCD 53
Figure 3.5 : Le module I2C 55
Figure 3.6: Un buzzer 56
Figure 3.7: Schéma synoptique de la carte de commande Arduino 58
Figure 3.8: Schéma électrique du L298 et le moteur cc sur Proteus 59
Figure 3.9: Schéma électrique des différentes entrées sur Proteus 59
Figure 3.10: Schéma électrique du différentes sorties sur Proteus 60
Figure 3.11: Schéma électrique globale de l'ascenseur sur Proteus 60
Figure 3.12: Organigramme du programme principal 61
Figure 3.13: sous-programme de vérification des boutons 62
Figure 3.14: sous-programme de vérification des capteurs 62
Figure 3.15: Sous-programme Afficher Des informations 63
Figure 3.16: Sous-programme du deplacement et stopMotor 63
Figure 3.17: Sous-programme pour allumerLed et éteindreLed et Sbuzzer 64
Figure 3.18 : Vue interne et externe de la maquette. 65
Figure 3.19 : branchements de la carte Arduino et du module L298.. 65
Figure 3.20 : Afficheur LCD indique l'etage actuelle 66
Figure 3.21: Afficheur LCD indique l'etage actuelle. 66

Liste des tableaux :

Tableau 1 : la barre d'icône en détaillé..... 19
Tableau 2 : Caractéristique technique d'Arduino UNO..... 23
Tableau 3 : Caractéristique technique d'Arduino Méga 24
Tableau 4 : Caractéristique technique d'Arduino Leonardo 26
Tableau 5 : Caractéristique technique d'Arduino Esplora 28
Tableau 6 : Caractéristique technique Boarduino V2.0..... 29
Tableau 7 : Caractéristique technique d'Arduino Nano 31
Tableau 8 : Caractéristique technique d'Arduino Lilypad..... 33
Tableau 9 : Caractéristique technique d'Arduino Due 34
Tableau 10 : Caractéristique technique d'Arduino Yun (microcontrôleur AVR). 37
Tableau 11 : Caractéristique technique d'Arduino Yun (Processeur Linux)..... 37

Table des matières

Introduction Générale :

I. Introduction (histoire) : 2

II. Définition :..... 2

III. Composants de la carte Arduino : 3

IV. Le microcontrôleur : 4

1. Définition :..... 4

2. La structure d'un microcontrôleur : 5

3. La signification des blocs du schéma fonctionnel du microcontrôleur :..... 6

3.1. L'unité centrale (CPU)..... 6

3.2. Le bus de donnée :..... 6

3.3. Les zones mémoires :..... 6

3.4. La mémoires programme (Flash+EEPROM) : 7

3.5. La mémoire de données (SRAM) : 7

3.6. Portes d'entrée et de sortie du microcontrôleur :..... 7

3.7. Le contrôleur d'interruption :..... 7

3.8. L'horloge interne : 7

4. Les entrées/sorties d'Arduino : 7

4.1. Les entrées/sorties numérique d'Arduino : 8

4.2. Les entrées analogiques : 9

4.3. Les sorties analogiques : 10

5. Le contrôleur USB : 11

6. L'alimentation électrique :..... 12

V. L'interface série : 13

VI. Le logiciel de programmation Arduino 14

1. L'IDE, le programme officiel : 14

2. Lancement de l'environnement de développement :..... 15

2.1. La barre de titre :..... 16

2.2. La barre de menu : 16

2.3. La barre d'icônes : 16

2.4. La zone des onglets : 16

2.5. L'éditeur : 17
2.6. La ligne d'information :..... 17
2.7. La fenêtre de messagerie : 17
2.8. La ligne d'état : 17
2.9. La barre d'icone en détail :..... 18
2.10. L'éditeur en détail : 20
7. Les points importants :..... 20
7.1. Point 1..... 20
7.2. Point 2..... 21
7.3. Points 3 21
7.4. Points 4 21
VII. La famille d'Arduino..... 22
1. Les différentes cartes Arduino :..... 22
1. Arduino Uno :..... 22
2. Arduino Méga 2560..... 24
3. Arduino Leonardo :..... 25
4. Arduino Esplora 27
5. Boarduino V2.0 29
6. Arduino Nano..... 30
7. Arduino LilyPad 32
8. Arduino Due :..... 33
9. Arduino Yun :..... 35
CHAPITRE 02 : GENERALITES SUR LES ASCENSEUR..... 39
I. Introduction..... 39
Histoire..... 39
II. Définition..... 40
III. Type d'ascenseur 40
1. Monte-charge 40
2. Monte-charge industriel..... 40
3. Ascenseur pour des personnes..... 40
IV. Classification des ascenseurs..... 41
1. Les ascenseurs hydrauliques 41
1.1. Les avantages de l'ascenseur hydraulique 43
1.2. Les inconvénients de l'ascenseur hydraulique..... 43
2. Les ascenseurs à traction à câbles..... 44
2.1. Les avantages des ascenseurs à traction pas câbles..... 45
I. Introduction :..... 50

1. Objectif : 50
II. Définition des composant utiliser : 50
1. Arduino Mega 2560 (Déjà définie dans le 1er chapitre page 24) 50
2. Moteur a courant continu : 50
3. Module L298N : 50
3.1. Principe de fonctionnement : 51
3.2. Les caractéristiques principales de module : 51
4. Capteur Infrarouge : 52
4.1. Définition : 52
4.2. Caractéristiques de IR : 52
5. Bouton poussoir : 53
6. Afficheur LCD 16X2 Rétroéclairage bleu et le module I2C : 53
6.1. Afficheur LCD : 53
6.1.1. Définition : 53
6.1.2. Les caractéristiques : 54
6.1.3. Les brochages : 54
6.2. Le module I2C : 54
6.2.1. Les caractéristiques : 55
6.2.2. Connection Arduino, LCD et I2C : 55
7. Buzzer : 56
III. Réalisation de l'ascenseur : 57
1. Présentations de prototype : 57
2. Schéma et simulation : 57
3. Conception, réalisation et programmation : 57
3.1. Principe de fonctionnement : 57
3.2. Conception de circuit : 58
3.3. Schémas électriques : 58
3.3.1. Circuit de puissance : 58
3.3.2. Les entrées : 59
3.3.3. Les sorties : 60
3.3.4. Schéma globale : 60
4. Programmation : 61
3.4. Organigramme de déroulement du programme : 61
3.4.1. Programme principale (void Loop) : 61
3.4.4. Sous-programme Affichage des informations à partir de l'écran LCD : 63
3.4.5. Sous-programme allumerLed (), eteindreLed () et Sbuzzer () : 64
4. Réalisation de la maquette : 64

4.1. Description de la maquette : 64
4.1.1. Partie externe :..... 64
4.1.2. Partie interne : 64
5. Test :..... 65
IV. Conclusion :..... 67
CONCLUSION GÉNÉRALE 1
BIBLIOGRAPHIE

INTRODUCTION GENERALE

Introduction Générale :

L’électronique est un domaine vaste et en constante évolution, permettant aux ingénieurs et chercheurs de traduire leurs idées en réalité à travers des circuits et systèmes complexes. L’utilisation de microcontrôleurs, comme ceux de la famille Arduino, a révolutionné la manière de concevoir des systèmes électroniques interactifs. Ces dispositifs sont capables de traiter des signaux physiques captés dans leur environnement pour déclencher des actions précises, ouvrant ainsi la voie à des applications dans divers secteurs, notamment l'automatisation, la robotique, et les systèmes embarqués.

Dans ce contexte, notre projet se concentre sur la conception et réalisation d’un ascenseur à plusieurs étages basés sur une carte Arduino Mega 2560. L’objectif est de démontrer la faisabilité d’une telle réalisation en utilisant des composants électroniques accessibles, tout en garantissant des performances optimales et une sécurité appropriée.

Le projet combine des aspects matériels, tels que les capteurs et moteurs, avec une programmation soignée pour gérer les mouvements de l'ascenseur et l'affichage des informations pertinentes sur un écran LCD. À travers cette étude, nous illustrerons l'importance de l'électronique dans les solutions pratiques pour la vie quotidienne, en offrant une interface simple et efficace pour l'utilisateur final.

Chapitre 01

I. Introduction (*histoire*) :

L'Arduino est à d'origine un projet d'étudiants de l'école de design d'interaction d'Ivrea en Italie. Au début des années 2000, les outils de conception de projets dans le domaine du design d'interaction étaient onéreux, avoisinant une centaine d'euros. Ces outils étaient pour la plupart conçus pour le domaine de l'ingénierie et de la robotique. Maîtriser et utiliser ces composants demandait beaucoup de temps et d'apprentissage, ce qui ralentissait fortement le processus de création pour ces jeunes étudiants

Il leur vient alors l'idée de créer une plateforme plus abordable et plus simple à utiliser, reposant sur l'environnement de développement Processing, mis au point en 2001 par des étudiants du MIT. C'est donc en 2003, dans le cadre d'un projet de fin d'études, que la carte Wring, ancêtre de l'Arduino, fut conçue. Visant à rendre la plateforme toujours moins chère et plus accessible, une équipe d'étudiants et de professeurs finit par concevoir la toute première Arduino en 2005. Entièrement open source, l'Arduino présentait l'avantage d'être en perpétuelle optimisation par la communauté d'utilisateurs.

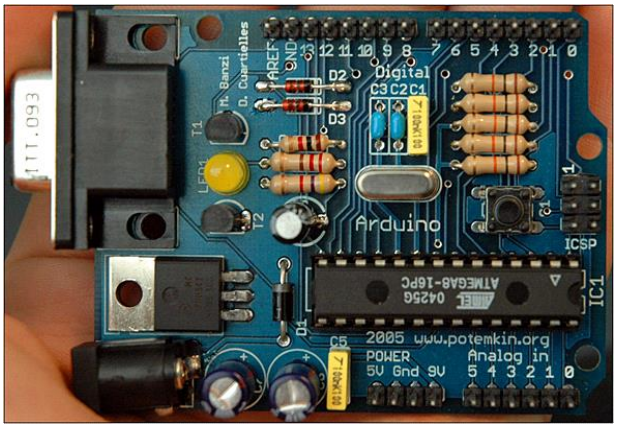


Figure1.1 : Arduino

II. Définition :

Arduino se présente généralement sous la forme d'une carte électronique bleue qui permet de construire des dispositifs capables d'interagir avec l'environnement qui les entoure en reliant à ses entrées des capteurs détectant des phénomènes physiques de la nature (vibrations, son, lumière, onde…). Cette carte, qui fait à peu près la taille d'une main, peut allumer une

lumière, changer sa couleur, mettre en route un moteur et bien d'autres choses. D'un autre aspect, Arduino est un microcontrôleur, autrement dit un ordinateur très simple. Il ne peut pas faire beaucoup de choses en même temps, mais ce qu'on lui demande de faire, il le fait très bien. Cette carte est également dotée de plusieurs broches d'entrée/sortie analogiques et numériques, ainsi que des broches d'alimentation (3.3V, 5V, GND), d'une broche ICSP, d'un bouton de remise à zéro, d'une prise USB et d'une prise d'alimentation.

III. Composants de la carte Arduino :

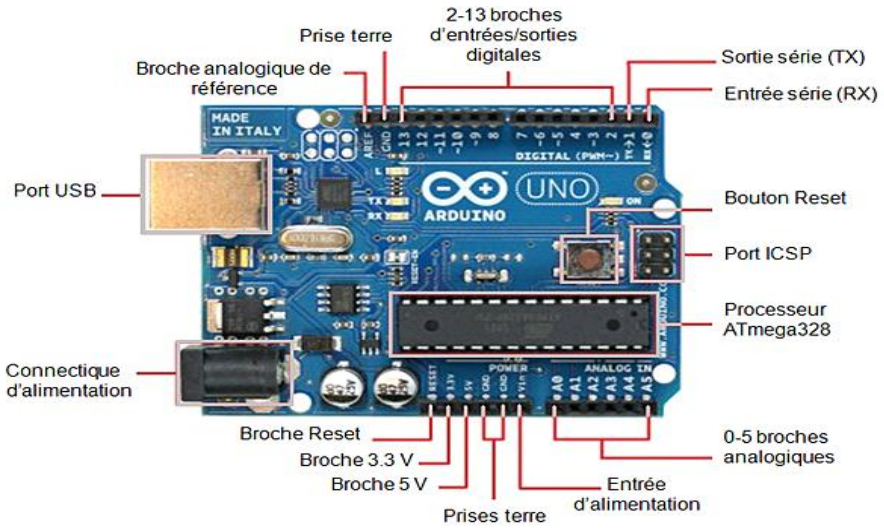


Figure1.2 : Composants de la carte Arduino

Voici les principales caractéristiques de la carte Arduino :

- **Microcontrôleur** (ATmega 328p, ATmega32u4, STM32H747 dual core…) : C'est le cerveau de la carte Arduino. Habituellement, il s'agit d'un microcontrôleur de la famille AVR d'Atmel, comme l'ATmega328 pour Arduino Uno.
- **Port d'alimentation** : il s'agit de connecteur pour fournir de l'énergie a la carte, généralement à travers un secteur ou une batterie ou un port USB.
- **Broches d'entrée/sortie (E/S)** : Ce sont les pins utilisés pour lire des signaux (capteurs, bouton …) ou contrôler des périphériques extérieurs (moteurs, pré-actionneurs, actionneur), il existe de sort d'entrées : entée analogique et entrée numérique.

- **USB/UART** : C'est l'interface USB qui permet de programmer notre carte et de la communiquer avec un ordinateur.
- **Régulateur de** tension : il sert à réguler la tension d'alimentation pour un fonctionnement correct de tous les composants.
- **Oscillateur/ cristal** : il fournit une horloge au microcontrôleur pour synchronisation des opérations.
- **LED d'alimentation** : elle indique l'état da la carte.
- **LED de transfert** : indique l'activité de la communication série entre la carte et l'ordinateur via le port USB, notamment le téléversement ou le transfert de programme ou de données vers la carte Arduino.

IV. Le microcontrôleur :

1. Définition :

Le microcontrôleur : Définition : Un microcontrôleur est un circuit intégré (Integrated Circuit, IC) qui regroupe sur une puce plusieurs éléments essentiels d'un ordinateur dans un espace réduit. C'est le cœur ou plutôt le cerveau de la carte. C'est le seul composant indispensable dans une carte Arduino ; les autres éléments sont simplement là pour l'alimenter en électricité ou pour l'aider à communiquer avec l'extérieur. Il est généralement moins puissant en termes de rapidité ou de taille de mémoire et présente un coût réduit par rapport aux microprocesseurs polyvalents utilisés dans les ordinateurs personnels. Ceci en fait un composant très bon marché, parfaitement adapté pour piloter les systèmes embarqués dans de nombreux domaines.

Figure1.3 : microcontrôleur

Les microcontrôleurs jouent un rôle prépondérant dans les domaines suivants. Cette liste est loin d'être exhaustive et sert surtout à donner une idée des diverses possibilités d'utilisation. Par exemple, ils remplissent des fonctions de surveillance dans des environnements critiques,

tels que les unités de soins intensifs (température, humidité, fréquence cardiaque, pression sanguine des prématurés…). Ils assurent également la commande de chauffage, permettant de contrôler la température externe ou interne pour un chauffage optimal des locaux. Dans le domaine des stimulateurs cardiaques, ils surveillent la fréquence cardiaque et, le cas échéant, stimulent le cœur. Les appareils ménagers modernes, tels que les lave-linges ou les lave-vaisselles, bénéficient de programmes enregistrés pour faciliter leur utilisation. En électronique de loisir, les microcontrôleurs se retrouvent dans des appareils comme les lecteurs MP3, les téléphones portables et les appareils photo. Dans la robotique, ils commandent des robots industriels pour le montage de pièces automobiles. Cette liste peut ainsi se poursuivre à l'infini, mais nous pouvons d'ores et déjà remarquer une chose : les microcontrôleurs perçoivent des influences extérieures via des capteurs, traitent ces informations en interne à l'aide d'un programme, puis envoient des ordres de commande correspondants vers l'extérieur. Ils démontrent donc une certaine intelligence, qui dépend bien évidemment du programme mis en œuvre.

En résumé, un microcontrôleur peut assurer des fonctions de mesure, de commande et de régulation.

2. La structure d'un microcontrôleur :

La structure d'un microcontrôleur : Comme la définition l'indique, un microcontrôleur est comparable à un mini-ordinateur. Il possède les mêmes éléments que ce dernier, sauf qu'ils sont regroupés dans un même boîtier, ce qui le rend simple et très pratique.

L'architecture matérielle d'un microcontrôleur est devisée en six grandes parties :

- L'unité centrale de traitement (CPU).
- Les mémoires (RAM et ROM).
- Les interfaces d'entrées/sorties.
- Une horloge ou un oscillateur
- Les bus des données
- Un contrôleur d'interruption

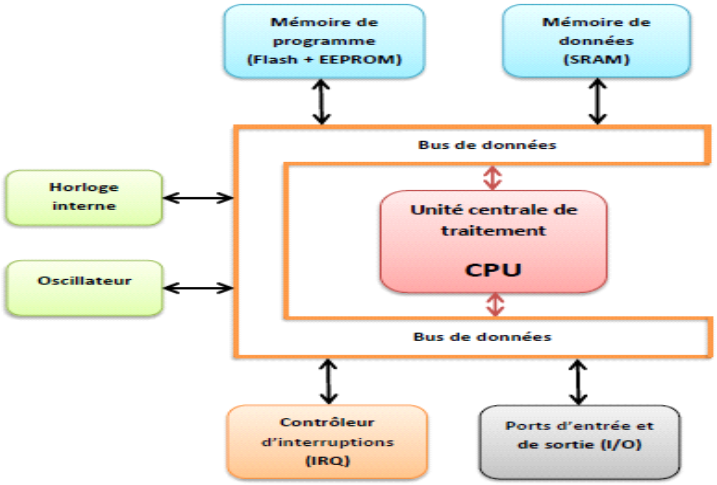


Figure 1.4 : Schéma fonctionnel d'un microcontrôleur

3. La signification des blocs du schéma fonctionnel du microcontrôleur :

3.1.L'unité centrale (CPU)

L'élément le plus important dans un microcontrôleur est l'unité centrale, appelée également CPU (Central Processing Unit). Sa fonction principale consiste à décoder et exécuter les instructions. Elle peut adresser des mémoires, gérer les entrées sorties, et réagir a des interruptions.

3.2.Le bus de donnée :

Le bus de données est un ensemble de fils électriques permettant l'interconnexion avec les différents blocs du CPU et de transporter les données d'un bloc à un autre. Par exemple, le CPU demande des données provenant de la mémoire, qui sont prises en charge par le bus, immédiatement mises à disposition pour traitement. Lorsque le résultat du calcul est disponible, il est à nouveau transféré sur le bus et transmis à un port de sortie qui, par exemple, pilote un moteur de robot pour atteindre un but précis. Cette structure de bus est une autoroute des données utilisable en commun par tous ceux qui en bénéficient.

3.3.Les zones mémoires :

En principe, il existe deux types de mémoires d'un microcontrôleur :

- La mémoire de programme.
- La mémoire des données.

3.4.La mémoires programme (Flash+EEPROM) :

Elle stocke le programme que le CPU doit exécuter. C'est une mémoire non volatile, c'est-à-dire qu'elle ne perd pas ses données lorsque le microcontrôleur n'est plus alimenté en tension externe. Un type de mémoire particulier est utilisé à cet effet, qu'on appelle mémoire flash.

3.5.La mémoire de données (SRAM) :

Utilisée pour gérer les résultats de calcul en cours, c'est une mémoire volatile. Au contraire de la mémoire programme, cela veut dire que dès que l'alimentation est coupée, les données mémorisées sont perdues. Pourtant, la SRAM offre un accès très rapide par rapport aux mémoires flash, ce qui présente un avantage très important.

3.6.Portes d'entrée et de sortie du microcontrôleur :

Une porte d'entrée/sortie, qu'elle soit numérique ou analogique, permet au microcontrôleur de communiquer avec le monde extérieur. Elle constitue une interface à laquelle une périphérie (résistance, LED, bouton poussoir, capteur de température, transistor, etc.) peut être connectée.

3.7.Le contrôleur d'interruption :

Il est utilisé lorsqu'une interruption matérielle (en anglais, Interrupt ReQuest ou IRQ) est déclenchée par un périphérique d'entrée/sortie du microcontrôleur. À ce moment-là, ce dernier doit arrêter temporairement l'exécution normale du programme afin de traiter un autre programme appelé service d'interruption. Ce type de programme est très utile dans des situations où il est nécessaire de réagir à des actions externes détectées par des capteurs, par exemple.

3.8.L'horloge interne :

Elle synchronise toutes les actions de l'unité centrale de traitement.

Arduino et microcontrôleur : quelle relation ?

En regardant une carte Arduino, on peut constater tout simplement qu'elle intègre un microcontrôleur avec d'autres périphériques et composants, ce qui permet de lui ajouter de nombreuses fonctionnalités.

4. Les entrées/sorties d'Arduino :

L'Arduino est conçu pour communiquer avec son environnement. Pour cela, il possède deux rangées de connecteurs Dupont femelles. La carte sert d'interface, et les broches sont reliées directement au microcontrôleur. Leur emplacement est normalisé, ce qui permet à tous les modèles d'utiliser les mêmes cartes d'extension. En général, le rôle d'une entrée de la carte

Arduino est de recevoir des informations provenant d'un capteur (bouton poussoir, capteur de température, potentiomètre, résistance photo-électrique, etc.) et de les transmettre au microcontrôleur, qui réagit selon le programme contenu dans sa mémoire pour commander la sortie. Cette dernière peut être liée à un actionneur, comme un servomoteur ou bien un dispositif lumineux ou sonore.

Leur principe de fonctionnement est illustré dans le schéma ci-dessous :

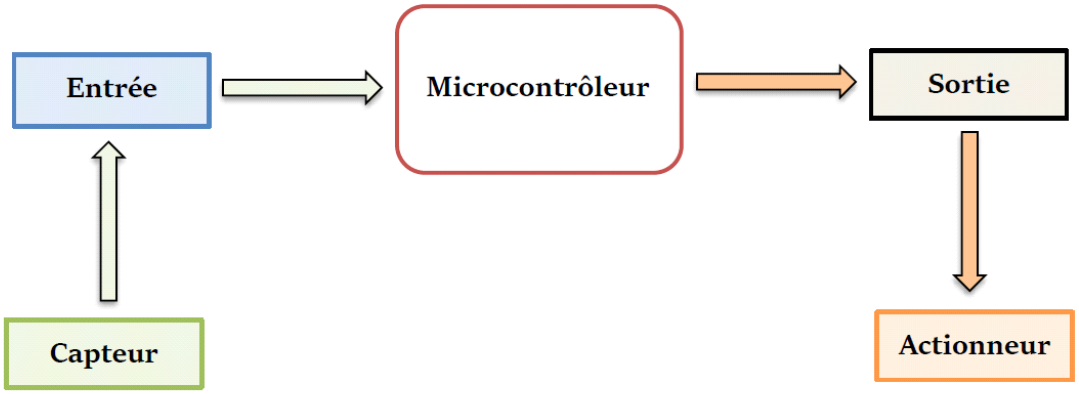


Figure1.5 : Shéma Les entrées/sorties d'Arduino microcontrôleur

Rappelons que la carte Arduino ne peut commander un actionneur de puissance directement, d'où la nécessité d'un pré-actionneur (relais, transistor de puissance …).

4.1.Les entrées/sorties numérique d'Arduino :

Une entrée/sortie est une connexion physique, une broche sur laquelle vous pouvez brancher un câble, un bouton, un capteur, etc. Une entrée/sortie numérique peut prendre l'un des trois états physiques suivants : "haut" (**HIGH**), "bas" (**LOW**) ou "haute impédance" (**INPUT**). L'état "haut" (**HIGH**) signifie que la broche génère un signal. Cet état se traduit généralement par une tension de 5 volts en sortie de la broche avec une carte Arduino classique. L'état "bas" (**LOW**) signifie que la broche ne génère pas de signal. Cet état se traduit généralement par une tension de 0 volt en sortie de la broche. L'état "haute impédance" (I**NPUT**) est un état particulier. Dans cet état, la broche ne génère aucun signal. L'état "haute impédance" signifie que la broche est "en lecture" (entrée). C'est cet état qui permet de "lire" un bouton, par exemple. La numérotation des broches commence de 0 jusqu'à 13. Il est indispensable de connaître le numéro de la broche à utiliser pour pouvoir communiquer avec elle lors de la programmation.

Une sortie numérique permet de transmettre une information binaire. Elle ne peut prendre que deux états, soit "haut" soit "bas". Dans le cas d'une sortie numérique, la tension de sortie est soit égale à 5 volts (pour "haut"), soit égale à 0 volt (pour "bas"). C'est cet état qui permet de "commander" un composant, par exemple pour allumer ou éteindre une LED.

4.2.Les entrées analogiques :

Un signal analogique est un signal qui varie dans le temps et peut prendre une infinité de valeurs intermédiaires, comme le montre l'exemple dans le graphique ci-dessous.

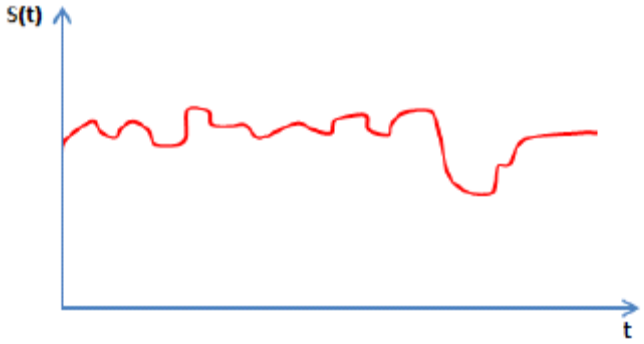


Figure1.6 : évolution d'un signal analogique en fonction du temps

On voit que son évolution est totalement différente d'un signal numérique où seule une alternance entre les niveaux 'HIGH' et 'LOW' est possible. Afin de traiter ce type de signaux, notre carte dispose d'entrées analogiques.

En électronique numérique, on travaille avec des bits et des octets. En analogique, on travaille avec des grandeurs physiques : tension, résistance, courant, fréquence.

Pour pouvoir exploiter des mesures analogiques avec le microcontrôleur, il faut convertir la mesure analogique en une grandeur numérique. C'est justement le but des convertisseur analogiques/numériques (CAN OU ADC : Analog to Digital Converter).

Un convertisseur analogique/numérique permet de mesurer une tension (valeur analogique) et de représenter cette tension au moyen d'une valeur numérique (nombre entier). L'idée est d'associer une valeur numérique à chaque valeur analogique d'une plage de tension bien précise.

L'Arduino Uno possède un convertisseur analogique/numérique avec une résolution de 10 bits ; il peut mesurer une tension analogique reçue sur une entrée analogique et renvoyer une valeur numérique comprise entre 0 et 1024. Cette dernière est due à la résolution du CAN

(2^10 = 1024). La résolution est le degré auquel quelque chose peut être représenté numériquement. Plus la résolution est élevée, Plus la résolution est élevée, plus la précision avec laquelle quelque chose peut être représenté est grande. Nous mesurons la résolution en termes de nombre de bits.

Par exemple, une résolution de 1 bit n'autoriserait que deux valeurs (2^1) : zéro et un. Une résolution de 2 bits permettrait quatre valeurs (2^2) : 0, 1, 2 et 3. Si nous essayions de mesurer une plage de 0 à 5 volts avec une résolution de 2 bits et que la tension mesurée était de 4 volts, notre CAN afficherait une valeur numérique de 3, car 4 volts se situent entre 3,75 et 5V. Il est plus facile d'imaginer cela avec la figure suivante :

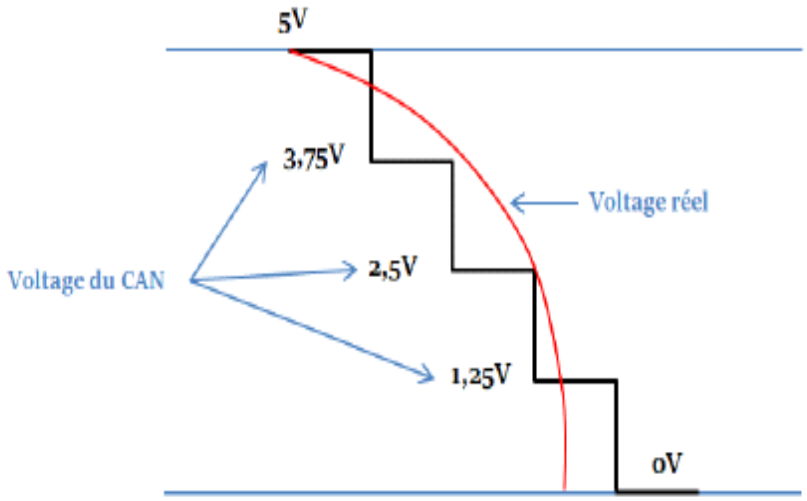


Figure1.7 : Graphe d'un signale numerique

Donc, avec notre exemple CAN avec une résolution de bits, il ne peut représenter la tension qu'avec quatre valeurs résultantes possibles. Si la tension d'entrées est comprise entre 0 et 1.25 v le CAN renvoie le 0 numérique, si la tension est entre 1.25 et 2.5 v Le CAN renvoie une valeur numérique de 1. Et ainsi de suite.

Remarque : si vous délivrer à une entrée analogique une tension supérieurs à 5 v, vous risquer de griller ce canal ou bien d'endommager complètement le microcontrôleur de la carte. Vérifiez-vous toujours sous quelles tensions vous travaillez.

4.3.Les sorties analogiques :

Sous certains chiffres des entrées/sorties numériques, se trouve un symbole (~) appelé TILD, indiquant que la broche est commutable en sortie analogique. Il s'agit d'une broche MLI.

Les broches 3, 5, 6, 9, 10 et 11 peuvent alors être utilisées comme sorties analogiques grâce à la technique de modulation de largeur d'impulsion (MLI ou PWM).

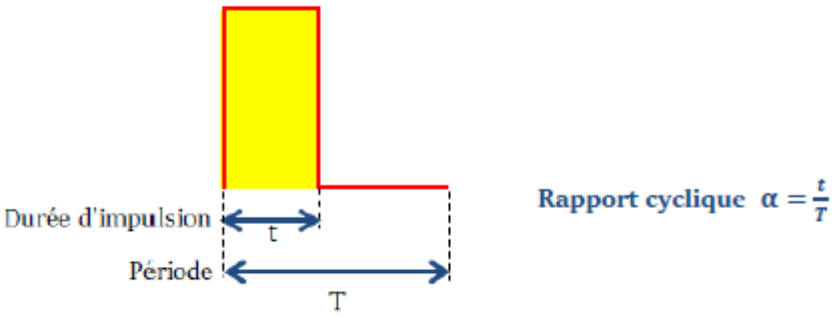


Figure1.8 : Durée d'impulsion et la période dans un signal PWM

Un signal MLI est un signal d'amplitude de tension et de fréquence constantes. On peut faire varier la largeur d'impulsion de ce signal en changeant la valeur d'un paramètre dit 'rapport cyclique' ; plus l'impulsion est large, plus la sortie reçoit d'énergie.

Certaines broches possèdent des caractéristiques supplémentaires :

- Les entrées/sorties numériques n 3,5,6,9,10,11 peuvent simuler une sortie analogique. Elles sont repérées sur la carte par le symbole -.
- Les broches n° 0 RX, (Reçoive) et 1 TX, (Transmit) sont reliées à la ligne de communication série. Elles permettent par exemple de connecter un module Bluetooth (ou un autre périphérique compatible avec le protocole sériel Les LED TX et RX clignotent pendant l'échange de données.
- La broche n 13 est reliée à la LED orange interne (L).

5. Le contrôleur USB :

Il est très utile, même si certains modèles n'en possèdent pas, il apporte un réel confort d'utilisation. Il assure la communication avec l'ordinateur (pour programmer l'Arduino). En convertissant le signal série en USB. Il permet aussi. Dans les limites de la prise USB (588 mA maximum). D'alimenter l'Arduino. C'est largement suffisant pour alimenter quelques LED. Mais si le montage est plus gourmand. Il Faudra prévoir une autre alimentation.

6. L’alimentation électrique :

L'Arduino n'est pas très difficile ; il accepte de nombreuses sources d'alimentation. Non seulement il peut être alimenté par l'ordinateur (sur la prise USB), mais il possède également un connecteur jack acceptant sans problème une tension continue comprise entre 7 et 12 V (même si elle n'est pas parfaitement régulée). Vous pouvez donc choisir parmi des alimentations aussi diverses qu'un transformateur de récupération, une pile de 9 V, une batterie rechargeable ou un ensemble de piles de 1.5 V. Vérifiez cependant la polarité de l'alimentation électrique avant de la brancher pour la première fois à l'Arduino. Sur la prise jack, le (+) est à l'intérieur et le (-) à l'extérieur.

L'Arduino peut aussi être alimenté en reliant une source électrique directement aux connecteurs réservés à l'alimentation. Les trois broches GND sont reliées à la masse. Choisissez-en une et branchez-y la borne moins (-). Si la tension de l'alimentation est comprise entre 7 et 12 V (même si elle n'est pas parfaitement stable), connectez la borne (+) à la broche Vin. Mais si vous possédez une alimentation parfaitement régulée de 5 V, vous pouvez l'utiliser en connectant le (+) à la broche 5 V et le (-) à l'une des bornes GND. Ce branchement est toutefois déconseillé si vous n'êtes pas sûr de votre alimentation. En effet, il contourne le régulateur de tension. Et si une chute de tension risque de provoquer des instabilités dans le programme (plantage de l'Arduino), un pic peut définitivement endommager la carte. Dans tous les cas. Vérifiez toujours attentivement vos branchements avant de mettre sous tension. Et surtout. N’inversez jamais le (+) et le (-), l'Arduino n'y survivrait certainement pas.

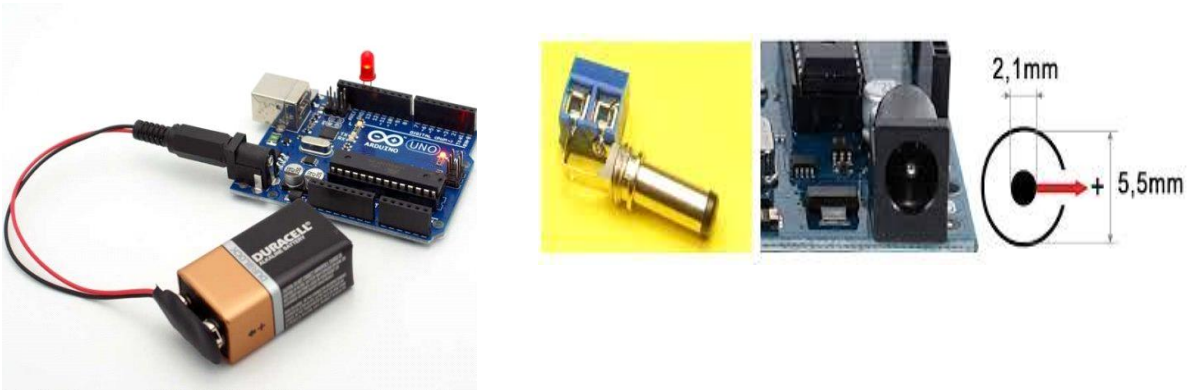


Figure 1.9 : alimentation de Uno par une batterie de 9V

V. L’interface série :

Les cartes Arduino, telles que les cartes Uno, MEGA2560 et Due, ont toutes un port série matériel qui utilise le port USB de la carte. Ce port permet de charger des sketchs sur la carte à l'aide d'un câble USB. Le code d’un sketch peut utiliser le même port USB/série pour communiquer avec le PC en utilisant une fenêtre du moniteur de l’interface série ou une application de traitement, par exemple. Le port USB apparaît en tant que port COM virtuel sur le PC.

L'Arduino Uno n'a qu'un seul port série car le microcontrôleur utilisé sur l'Uno ne possède qu'un seul port série intégré. L'Arduino MEGA 2560 et l'Arduino Due ont tous les deux trois ports série supplémentaires. Cette figure montre le seul port série disponible sur l'Arduino Uno, encadré en rouge.

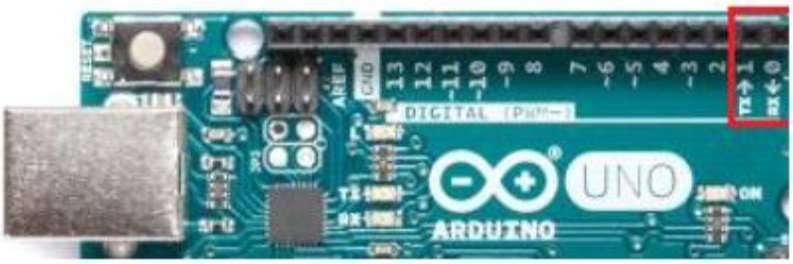


Figure 1.10 : port série de la carte Uno

Ce type de communication avec votre carte Arduino peut être utile dans de nombreux cas :

- Pour entrer des données pendant l’exécution d’un sketch.
- Pour afficher des données pendant l’exécution d’un sketch.
- Pour imprimer certaines informations pendant l’exécution du sketch, afin de rechercher des erreurs.

VI. Le logiciel de programmation Arduino

1. L'IDE, le programme officiel :

L'IDE (integrated Devlopment Environment) d'Arduino qu'on pourrait traduire par environnement de développement intégré est né en même temps que l'Arduino. Alors nous disposant d'un environnement de développement IDE pour la programmation du microcontrôleur Arduino, au moyen auquel on entre directement en contact avec la carte et on charge le programme.

C'est une évolution de Wiring et processing (d'autres langage écrit sous licences libres avant la création de l'Arduino. 0Le langage Arduino est très proche des langages C et C++ auxquels s'ajoutent les fonctions des nombreuses bibliothèques (libraries) d'Arduino.

Il est écrit en java, ce qui permet in portage facile et une interface quasiment identique, quel que soit votre système d'exploitation.

L'IDE Arduino permet de regrouper dans le même outil les programmes nécessaires au pilotage de la carte. Il comprend in éditeur de texte, une débogueur/compilateur, une interface permettant de gérer les ports COM et le type de la carte. De plus, il fournit de nombreux exemples, installe automatiquement les drivers les plus courants et assurer ensuite la communication avec l'Arduino (téléversement et moniteur série).

Figure 1.11 : icone de l'IDE dans le bureau de Windows

Un programme est appelé sketch dans le contexte Arduino, qu'on peut traduire approximativement par esquisse. A l'avenir, nous parlons alors des sketchs pour désigner le programme Arduino.

Qu'est-ce qu'un un environnement de développement ?

Un environnement de développement permet, au développeur ou l'expert Arduino de transporter ses idées de programmation sur des objets matériels (Hardware), avec comme composant principale la carte Arduino, à laquelle peuvent être raccordés les éléments électronique ou électrique les plus divers. Ce sont là des choses simples, mais hard par leurs structures, d'où le terme hardware employé. Seulement, a quoi sert ce matériel si on ne lui dit pas ce qu'il doit faire, il a, en effet, quelque chose qui manque, et ce quelque chose c'est le logiciel (software), le monde des données et des programmes ou sketches dans le cas d'Arduino, le logiciel est soft, autrement dit immatériel, il permet au matériel d'interpréter et d'exécuter des instructions.

Le hard et le soft forment un entier indissociable ; ils ne sont rien sans l'autre

Pour télécharger la dernière version de l'IDE, allez sur le site officiel Arduino, et sélectionnez l'onglet SOFTWARE : https://www.arduino.cc/en/Main/Software.

Choisissez la version correspondant à votre système.

2. Lancement de l'environnement de développement :

Pour démarrer l'IDE sous Windows, vous y accédez depuis le menu Démarrer ou sur le bureau à travers l'icône de l'Arduino IDE en double-cliquant avec le bouton droit de la souris. La fenêtre suivante apparaîtra à l'ouverture de l'IDE.

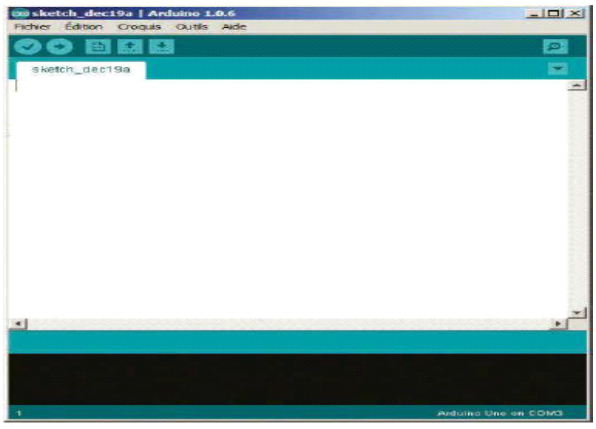


Figure1.12 : La fenêtre de L'IDE vide

En l'observant de plus près, vous pourrez remarquer différentes zones dans lesquelles il se passera peut-être quelque chose plus tard… Nous allons les passer toutes en revue en partant du haut de la fenêtre vers le bas.

2.1.La barre de titre :

La barre de titre, qui se situe tout en haut de la fenêtre, comprend deux informations :

- Le nom du sketch (ici, sketch_sep22a) et est attribué automatiquement et commence toujours par sketch. Viennent ensuite le mois, le jour et une lettre prise dans l'ordre, entre a et z, dans le cas ou d'autres sketchs seraient créés ces jours-là.
- Le numéro de la version de l'IDE Arduino (ici, la version 1.0.6), qui augmentera au fil du temps dès que les erreurs auront été éliminées où de nouvelles fonctions ajoutées.

2.2.La barre de menu :

Dans la barre de menu, vous trouverez les différents menus grâce auxquels vous pourrez appeler certaines fonctions de l'IDE

Figure1.13: La barre de menu

2.3. La barre d'icônes :

La barre d'icônes se situe sous celle des menus.

Figure1.14 : La barre d'icones

2.4.La zone des onglets :

La zone des onglets indique combien de fichiers de code source font parte du projet Arduino actuellement ouvert.

Pour le moment, seul un onglet ayant pour nom sketch_dec19a apparait. Cependant, d'autres peuvent être ajoutés au gré de ka programmation. Pour cela il faut se servir de l'icône située sur le côté droit.

Figure1.15 : La zone des onglets

2.5.L'éditeur :

C'est le cœur de l'IDE. La zone d'éditeur qui est pour le moment encore complètement vierge est le lieu central ou vous pouvez étaler vos idées. Vous y saisissez le code source ainsi que les instructions qui conduiront le microcontrôleur à faire ce que vous voulez.

Figure1.16 : Page de sketch pour saisir le code source

2.6.La ligne d'information :

La ligne d'information vous renseigne sur certaines actions entreprises par l'IDE. Tout est en anglais, naturellement."

Par exemple si vous ayez enregistre avec succès un sketch sur le disque dur c'est ici que vous en êtes informé. En outre si par exemple le compilateur a détecté une erreur de saisie dans le sketch lors de la transcription vous êtes prévenu par un message.

D'autres détails sur les erreurs détectées s'affichent dans la fenêtre de messagerie (voir la capture précédente).

2.7.La fenêtre de messagerie :

L'IDE vous fournit dans la fenêtre de messagerie tout un tas d'informations :

- Sur le transfert d'un sketch sur la carte Arduino (succès ou échec) ;
- Sur les activités de traduction du compilateur (succès ou échec) ;
- Sur le moniteur série (succès ou port COM non trouve).

2.8.La ligne d'état :

La ligne d'état indique soit le numéro de la ligne du curseur (ici, ligne 3)

Figure1.17 : La ligne d'état

A droit, vous pouvez voir en plus le nom de la carte Arduino et le port COM utilisé par votre interface série.

2.9.La barre d'icone en détail :

À force d'utiliser quotidiennement l'IDE, vous vous apercevrez que la barre d'icônes est votre compagnon le plus précieux. Même si la barre ne contient pas beaucoup d'icônes, il vous faut néanmoins en maîtriser les fonctionnalités.

	Pour vérifier la syntaxe du sketch qui se trouve dans l'éditeur et compiler le programme.  Cette barre horizontale s'affiche au début de la vérification/compilation, laquelle indique la progression. Si aucune erreur n'est constatée, l'opération se termine par le message Done compiling. Dans la fenêtre d'édition, se trouve une indication relative aux besoins en mémoire du sketch. 
	Pour créer un nouveau sketch. Souvenez-vous que l'IDE ne peut pas gérer qu'un seul sketch à la fois. Si vous en démarrez un nouveau, n'oubliez pas surtout d'enregistrer l'ancien, faute de quoi vous perdez toutes les informations.
	Tous les sketches sont consignes dans un livre de sketches qui se trouve dans le répertoire **C : \ Utilisateur\<Nom d 'utilisateur> \Mes documents\Arduino. Le nom d'utilisateur à saisir est le vôtre.** Cette icone sert à charger un sketch enregistré sur le disque dur dans l'IDE. Elle vous permet aussi d'accéder aux nombreux exemples de sketches livres avec l'IDE. Regardez-les car ils peuvent vous aider.
	Pour enregistrer votre sketch sur un support de données. L'enregistrement s'effectue par défaut dans le répertoire du livre de sketches mentionné plus haut.
	Pour transmettre le sketch compilé avec succès sur la carte Arduino dans le microcontrôleur.

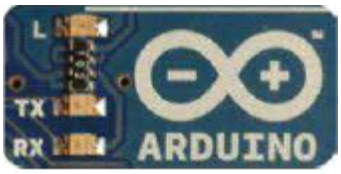
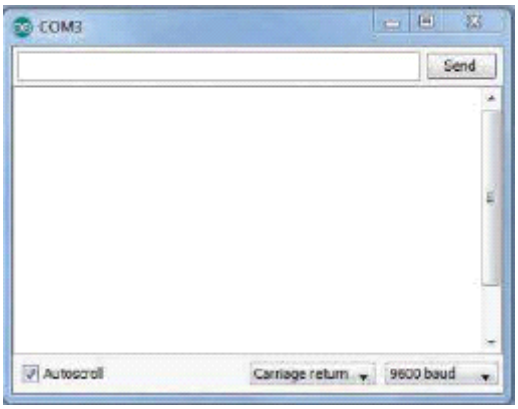
	Pendant I édit téléchargement du sketch, voici ce qui se produit - sur la carte se trouvent des petites diodes lumineuses qui vous tiennent au courant de certain es activités.  - LED L **:** reliée la broche 13 pour la Uno, elle s'allume brièvement quand la transmission commence. - LED TX : ligne émettrice de l'interface série de la carte. Elle clignote pendant la transmission. - LED RX : ligne réceptrice de l'interface série de la carte. Elle clignote pendant la transmission. La ligne émettrice (TX) est matériellement reliée à la broche numérique 1, et la ligne réceptrice (RX) à la broche numérique 0.
	Le moniteur série peut être ouvert avec cette icône. Une boîte de dialogue ressemblant à un terminal s'ouvre. Dans le champ de saisie supérieur, vous pouvez entrer des commandes qui seront envoyées à la carte quand vous appuierez sur la touche Send. La zone centrale de la fenêtre est consacrée aux données envoyées par la carte via l'interface série. Certaines valeurs auxquelles vous vous intéressez peuvent y être affichées. Dans la partie inférieure, vous pouvez, grâce à une liste déroulante à droite, régler la vitesse de transmission (baud) qui doit correspondre à la valeur que vous avez employée pour programmer le sketch. Si ces valeurs ne correspondent pas, aucune communication n'est possible. 

Tableau 1 : la barre d'icône en détaillé

2.10. L'éditeur en détail :

L'éditeur, dans lequel vous saisissez votre code source, vous assistera à maintes reprises dans la programmation. La figure vous présente le contenu de la fenêtre : il s'agit d'un code source pour faire afficher le message 'bonjour mon ami Arduino'.

```
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  Serial.println("Bonjour mon ami Arduino");
}
```

Figure1.18 : code source d'un Sketch Arduino

Voici quelques caractéristiques qui sautent aux yeux dans ce programme :

- L'IDE est capable de faire ressortir certains mots en couleurs dans l'éditeur.
- Les caractères sont plus au moins gras selon les mots.
- Certains éléments ressortent plus particulièrement. Il s'agit ici de l'accolade finale.
- La représentation du code source obéit à une certaine hiérarchisation visuelle. Certaines zones sont plus décalées à droite que d'autre. Bien évidemment, ce n'est pas pour rien, ni seulement pour faire beau : toute a une raison d'être.

7. Les points importants :

7.1.Point 1

Certains mots, appelés également mots-clés, apparaissent en couleurs. Il s'agit de noms réservés qui ont été, par exemple, attribués à des instructions. Notre environnement de développement ou le compilateur dispose d'un certain vocabulaire que nous pouvons utiliser pour programmer notre sketch. Quand on saisit un mot (ou un mot-clé) qu'il connaît, l'IDE réagit en le faisant aussitôt ressortir en couleurs. Dans le cas présent, les mots-clés apparaissent toujours en orange. Ainsi, vous conservez une meilleure vue d'ensemble et vous pourrez visualiser immédiatement si une instruction a été mal orthographiée. En effet, si tel est le cas, elle n'apparaîtra pas dans la couleur appropriée.

7.2.Point 2

L'IDE représente en gras certains mots reconnus en tant que mots-clés. Il s'agit ici, par exemple, des mots SETUP et LOOP, qui sont appelés à jouer un rôle fondamental dans un sketch. Ce sont des noms de fonctions. Pour le moment, peu importe ce que c'est exactement et ce qu'ils veulent dire, disons simplement qu'ils sont en gras pour mieux attirer l'attention.

7.3.Points 3

Les instructions sont toujours présentées par blocs dans l'environnement de programmation de l'IDE. Cela signifie que les instructions affichées l'une en dessous de l'autre font partie d'un bloc, et que l'accolade finale marque la fin. Ces deux accolades sont indissociables : si l'un des deux vient à manquer, il s'ensuit obligatoirement une erreur, car la structure du bloc n'est pas complète. Si vous placez le curseur derrière une accolade, l'autre accolade de la paire se retrouve automatiquement encadrée. Sur la figure, on le remarque pour la fonction SETUP : j'ai placé le curseur derrière l'accolade initiale, et l'accolade finale correspondante s'est alors dotée d'un cadre. Ceci est également valable pour les parenthèses.

7.4.Points 4

Dans un bloc d'exécution, le code source est généralement décalé à droite par rapport au bloc ou libellé du bloc proprement dit. Ainsi, la vue d'ensemble est bien meilleure et la recherche d'erreurs en est facilitée. Cette distinction visuelle permet, par ailleurs, de mieux différencier les blocs quand il y en a plusieurs. Bien entendu, rien ne vous empêche d'écrire l'intégralité du code source sur une seule ligne. Même si le compilateur ne détecterait aucune erreur de syntaxe, la vue d'ensemble serait néanmoins catastrophique. De même, vous pourriez aligner toutes les lignes de code à gauche, mais le style de programmation ne serait alors pas terrible. Notez qu'il existe une option pour indenter automatiquement le code via Tools > Auto format.

VII. La famille d'Arduino

1. Les différentes cartes Arduino :

Les cartes Arduino doivent satisfaire des exigences diverses et variées. Certains utilisateurs souhaiteront effectuer du prototypage et tester de nouveaux montages ou idées, la carte Arduino Uno disposant d'assez d'entrées-sorties pour des projets d'envergure raisonnable. De par sa taille, la carte devrait aussi pouvoir être utilisée à l'avenir dans des projets plus ambitieux. D'autres auront besoin d'un grand nombre de ports afin de pouvoir raccorder de nombreux capteurs ou actionneurs. La forme, la taille ou les possibilités de connexion jouent un rôle décisif dans le choix de la carte adaptée. C'est pourquoi des développeurs d'Arduino ont mis au point un vaste choix de cartes à microcontrôleur afin que chacun trouve le modèle qui réponde à ses besoins.

1. Arduino Uno :

Figure1.19 :Arduino UNO

C'est simplement l'évolution du premier Arduino. Il s'agit donc du plus simple, du plus utilisé et du plus documenté. Il est parfait pour débuter, car il n'est ni trop cher, ni trop grand. Il est suffisamment robuste et puissant pour commencer et possède un nombre d'entrées et de sorties suffisant pour la plupart des projets. De plus, il est compatible avec quasiment toutes les cartes d'extension, ce qui lui offre d'énormes possibilités d'évolution.

- **Caractéristique technique :**

Catégorie	Valeur
Microcontrôleur	ATmega 32U4
Fréquence d'horloge	16MHz
Tension de service	5V
Tension d'entrée (recommander)	7-12 V
Tension d'entrée (limites)	6-20 V
Ports numériques	20 entrées et sorties (7 sorties commutables en MLI)
Ports analogiques	12 entrées analogiques
Courant maxi. Par broche d'E/S (c.c)	40mA
Courant maxi. Par broche 3,3 V	50mA
Mémoire	32 Ko Flash, 2,5 Ko SRAM, 1 Ko EEPROM
Chargeur d'amorçage	4 Ko (en mémoire Flash)
Interface	USB

Tableau 2 : Caractéristique technique d'Arduino UNO

- **Avantages**
  - Carte Arduino par excellence pour laquelle de nombreux exemples de montage sont disponibles sur Internet ;
  - Nombre suffisant de broches d'entrées-sorties pour des projets élémentaires - vaste choix de shields ;
  - Coût moins cher ;
- **Inconvénients**
  - Nombre insuffisant de broches d'entrées-sorties pour les projets ambitieux ;
  - La mémoire disponible risque d'être un peu juste pour les gros projets ;

- Ne peut pas être utilisée comme hôte USB pour simuler un clavier ou une souris, par exemple (voir Arduino Leonardo). Vous trouverez de plus amples informations a la page http://arduino.cclen/Main/arduinoBoardUno.

2. Arduino Méga 2560

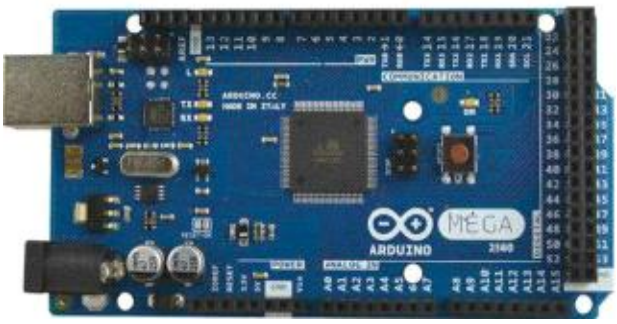


Figure1.20 : Arduino Méga 2560

Son microcontrôleur (ATmega2560), bien que cadencé à la même fréquence que l'Arduino Uno, dispose de beaucoup plus de mémoire. Il est plus long, car il possède beaucoup plus de connecteurs. Cependant, il reste compatible avec les cartes d'extension au format classique tout en bénéficiant de ses propres shields. Il n'intègre pas moins de seize entrées analogiques et de cinquante-quatre entrées/sorties numériques, dont quinze sorties analogiques virtuelles (PWM). Il permet, grâce à ses nombreux connecteurs, de réaliser des projets beaucoup plus ambitieux que l'Arduino Uno, mais il faudra certainement connecter une alimentation externe à la prise jack, car la prise USB de l'ordinateur ne peut fournir que 500 mA. Cela risque d'être insuffisant et peut poser des problèmes de fonctionnement (plantages aléatoires de l'Arduino).

- **Caractéristique technique :**

Catégories	**Valeurs**
Microcontrôleur	ATmega2560
Tension d'entrées	7-12 V
Tension de fonctionnement	5V
Fréquence	16MHz
Ports numériques	54 entres et sorties (15 sorties commutables en MLI)
Ports analogiques	16 entrées analogiques
Courant maxi par broche d'E/S	40mA
Courant maxi par broche 3,3 V	50mA

Mémoire	256Ko Flash,8Ko SRAM ,4Ko EEPROM
Chargeur d'amorçage	8Ko en mémoire flash
Interface	USB
Dimension	10,16cm x5,3cm

Tableau 3 : Caractéristique technique d'Arduino Méga

- **Avantages :**
  - Nombreuses entrées et sorties pour raccorder des capteurs ou des actionneurs.
  - Capacité de mémoire suffisante pour les gros projets.
  - Plus de broches UART (4 ports de communication série).
  - Plus de broches MLI.
  - Compatible avec la plupart des shields conçus pour l'Arduino Uno, par exemple.
  - Il existe des shields spéciaux pour le prototypage qui, en raison de leur surface supérieure, peuvent recevoir plus de composants ; de nombreux schémas et exemples sont disponibles sur Internet.
- **Inconvénients :**
  - Facteur de forme plus élevé que pour l'Arduino Uno, par exemple ;
  - Deux fois plus cher que l'Arduino Uno.

3. Arduino Leonardo :

La carte Arduino Leonardo est considérée comme le successeur officiel de l'Arduino Uno. Elle est équipée du microcontrôleur ATmega32U4 qui communique également via l'interface USB, de sorte qu'un processeur supplémentaire est inutile. La carte peut être programmée en tant qu'hôte USB pour lequel différentes classes de clavier et de souris sont disponibles. D'ailleurs, la carte Arduino Leonardo peut aussi être programmée en tant que HID (Human Interface Device). En plus du port matériel UART, auquel le microcontrôleur a accès depuis la classe Serial, un port COM virtuel (VCP : Virtual COM-Port) est accessible via USB et peut être sollicité par le biais de Serial.

Figure1.21 : Arduino Leonardo

- **Caractéristique technique :**

Catégorie	**Valeur**
Microcontrôleur	ATmega 32U4
Fréquence	16MHz
Tension service	5V
Tension d'entrée	7-12V
Tension d'entrée (limite)	6-20V
Ports numériques	20 entrées et sorties (7sortis commutable en MLI)
Ports analogiques	12 entrées analogiques
Courant maxi par broche d'E/S	40Ma
Courant maxi par broche 3,3 V	50mA
Mémoire	32 Ko Flash, 2,5 Ko SRAM, 1 Ko EEPROM
Chargeur d'amorçages	4 Ko (en mémoire Flash)
Interface	USB

Tableau 4 : Caractéristique technique Arduino Leonardo

- **Avantages**
  - Nombreuses entrées et sorties pour raccorder des capteurs ou des actionneurs ;
  - Capacite de mémoire suffisante pour les gros projets ;
  - Plus de broches UART (4 ports de communication série) ;

- Plus de broches MLI ;
- Compatible avec la plupart des shields conçus pour l'Arduino Uno, par exemple ;
- Il existe des shields spéciaux pour le prototypage qui, en raison de leur surface supérieure, peuvent recevoir plus de composants ;

- **Inconvénients**
  - Facteur de forme plus élevé que pour l'Arduino Uno, par exemple ;
  - Deux fois plus cher que l'Arduino Uno ;

4. Arduino Esplora

L'Arduino Esplora est une carte sur laquelle est basée l'Arduino Leonardo. Si vous l'examinez de plus près, vous constaterez qu'elle est dotée d'un certain nombre de capteurs qui ne sont pas présents sur la carte Arduino Uno ni sur d'autres cartes.

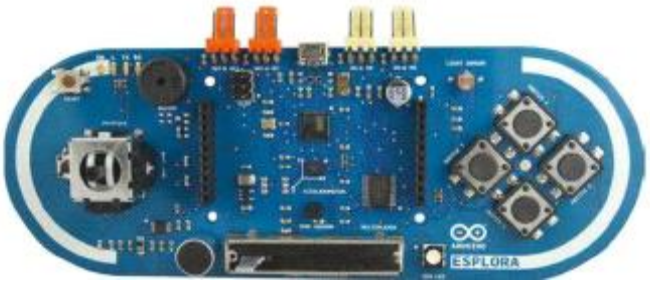


Figure1.22 : Arduino Esplora

- **Les capteurs suivants sont préinstallés :**
  - Un joystick avec un bouton-poussoir central ;
  - Un microphone ;
  - Un potentiomètre (linéaire) ;
  - Un capteur de température ;
  - Un accéléromètre 3 axes (x, y et z) ;
  - Quatre bouton-poussoir disposés en diamant ;
  - Un capteur de lumière ;
  - La carte comporte aussi les sorties suivantes ;
  - Un buzzer ;

- Une LED RVB ;

Comme vous pouvez le remarquer, cette carte est donc équipée de nombreux capteurs que vous devrez acheter en plus si vous choisissez un autre modèle Arduino. Sur le bord supérieur, vous trouverez également :

- Deux entrées Tinker pour relier des modules de capteurs TinkerKit à l'aide des connecteurs 3 broches (broches jaunes) ;
- Deux sorties Tinker pour relier des modules d'actionneurs TinkerKit a l'aide des connecteurs 3 broches (broches orange) ;
- Un connecteur pour écran couleur TFT avec lecteur de carte SD utilisant le protocole SPI ;

- **Caractéristique technique :**

Catégorie	**Valeur**
Microcontrôleur	**ATmega 32U4**
Fréquence d'horloge	**16MHz**
Tension de service	**5V**
Mémoire	32 Ko Flash, 2,5 Ko SRAM, 1 Ko EEPROM
Interface	USB
Chargeur d'amorçage	**4 Ko (en mémoire Flash)**
Dimensions	**16,51cmx6 cm**

Tableau 5 : Caractéristique technique d'Arduino Esplora

- **Avantages**
  - Nombreux capteurs préinstallés ;
  - Connecteurs pour modules TinkerKit ;
- **Inconvénients**
  - Il n'est pas possible d'utiliser des shields ;
  - Possibilités d'extension limitées ;
  - Relativement peu de circuits ou d'exemples de montages disponibles sur Internet par rapport aux cartes Arduino Uno ou Arduino Mega 2560 ;

- Prix assez élevé ;

5. Boarduino V2.0

Si vous envisagez d'acheter une carte Boarduino, sachez que vous devez avoir un fer à souder sous la main, car elle est livrée en pièces détachées à assembler soi-même, à savoir :

- Un microcontrôleur.
- Un support de circuit à 28 broches.
- Une carte.
- Des connecteurs.

Bien qu'il soit annoncé que seuls les modèles Arduino originaux seraient présentés, une exception est faite pour le Boarduino, qui est compatible avec Arduino. Il est conçu pour être monté sur une plaque d'essai sans soudure.

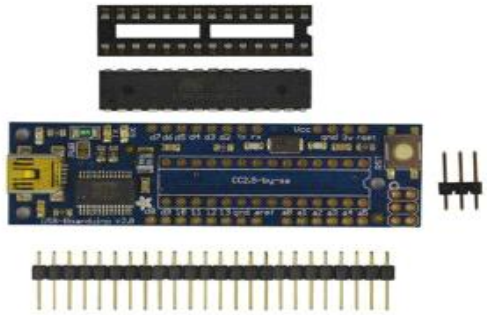


Figure1.23 : Boarduino V2.0

.

- **Spécifications de la carte Boarduino :**

Catégorie	**Valeur**
Microcontrôleur	ATmega 328
Fréquence d'horloge	16MHz
Tension de service	5V
Tension d'entrées (recommandée)	7-17V
Ports numériques	14 entrées et sorties (6 sorties commutables en MLI)
Ports analogiques	6 entrées analogiques
Courant maxi. Par broche d'E/S	40mA

Mémoire	32 Ko Flash, 2 Ko SRAM, 1 Ko EEPROM
Chargeur d'amorçages	O, 5 Ko (en mémoire Flash)
Interface	USB
Dimensions	7,5 cmx2cm

Tableau 6 : Caractéristique technique Boarduino V2.0

- **Avantages**
  - Encombrement réduit ;
  - Peut-être enfichée directement sur la plaque d'essai.
- **Inconvénients**
  - Nécessite un peu de soudure avant d'être prête à l'emploi (ce n'est évidemment pas un problème pour les experts).

6. Arduino Nano

La carte Arduino Nano possède des connecteurs au dos qui permettent de l'enficher facilement sur une plaque d'essai, ce qui évite d'avoir recours à des cavaliers flexibles, comme c'est le cas pour l'Arduino Uno. Ne vous laissez pas abuser par les dimensions de cette mini carte, dont les performances n'ont (presque) rien à envier à l'Arduino Uno.

Ne vous laissez pas abuser par les dimensions de cette mini carte dont les performances n'ont (presque) rien à envier à I' Arduino Uno.

Figure1.24 : Arduino Nano

- **Spécifications de la carte Arduino Nano :**

Catégorie	**Valeur**
Microcontrôleur	AT mega 168 OU 328
Fréquence d'horloge	16MHz

Tension de service	5V
Tension d'entrées (recommandée)	7-12V
Tension d'entrées (limites)	6-20V
Ports numériques	14 entrées et sorties (6 sorties commutables en MU)
Ports analogiques	8 entrées analogiques
Courant maxi. Par broche d'E/S (c.c.)	40mA
Mémoire	ATmega 168: 16 Ko memoire Flash 1 KoSRAM 512 octets d'EEPROM AT mega 328: 32 Ko memoir Flash 2 Ko SRAM
Chargeur d'amorçages	2 Ko (en mémoire Flash)
Interface	USB
Dimensions	1,9cmx4,3cm

Tableau 7 : Caractéristique technique d'Arduino Nano

- **Avantages**
  - Encombrement réduit ;
  - Peut-être enfichée directement sur la plaque d'essai.
- **Inconvénients**
  - Il n'est pas possible d'utiliser des shields.

7. Arduino LilyPad

La plateforme Arduino LilyPad est destinée aux créatifs qui souhaitent coudre des circuits électroniques sur leurs vêtements. Les raccordements électriques ne se font pas par câbles, mais par des fils conducteurs. D'après le fabricant, le composant est lavable. Toutefois, il est conseillé de ne pas le passer en machine. Il est préférable de le laver délicatement à la main avec un détergent doux.

Autre précaution : Pensez bien à débrancher l'alimentation électrique avant le nettoyage.

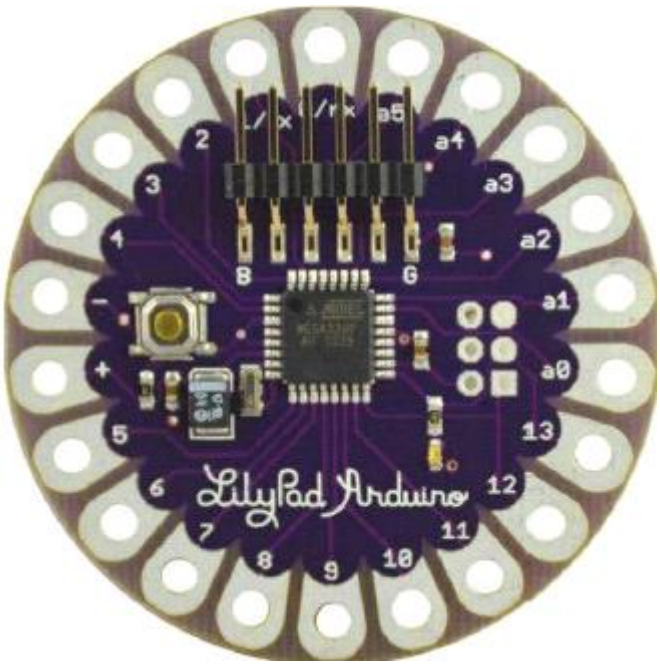


Figure1.25 : Arduino LilyPad

- **Spécifications de la carte Arduino LilyPad :**

Catégorie	**Valeur**
Microcontrôleur	AT mega 168 V ou 328 V
Fréquence d'horloge	8MHz
Tension de service	2,7-5,5 v
Tension d'entrées (recommandé)	2,7-5,5 v
Ports numériques	14 entrées et sorties (6 sorties commutables en MLI)
Ports analogiques	6 entrées analogiques

Courant maxi. Par broche d'E/S	40mA
Mémoire	16 Ko Flash, 1 Ko SRAM, 512 octets EEPROM
Chargeur d'amorçages	2 Ko (en mémoire Flash)
Dimensions	Cm de diamètre

Tableau 8 : Caractéristique technique d'Arduino Lilypad

- **Avantages**
  - Fringale : carte extraplate conçue pour être intégrée à des vêtements.
- **Inconvénients**
  - La programmation ne se fait pas par USB, mais par un module adaptateur FTDI (5 V).
  - Comme les raccordements électriques peuvent uniquement être soudes sur la plate-forme Lily Pad, cette carte se prête davantage aux montages qui ne nécessitent pas d'ajustements fréquents.

8. Arduino Due :

L'Arduino Due est la première carte Arduino équipée d'un processeur 32 bits. La fréquence d'horloge de 84 MHz permet de réaliser des calculs complexes en un temps record. De plus, les programmes les plus lourds disposent désormais d'une capacité de mémoire suffisante, ce qui évite d'avoir à faire attention au moindre octet, comme c'est le cas sur une Arduino Uno.

Comme toutes les cartes Arduino antérieures utilisent une tension d'entrée de 5 V, des problèmes risquent de se poser si l'on ne sait pas que les entrées de la carte Arduino Due sont limitées à une tension de 3,3 V. La carte risque d'être irrémédiablement détruite.

Au lieu de la résolution 8 bits habituelle (analogWrite (0 ... 255)), les 12 broches numériques, qui peuvent être utilisées comme sorties MLI, ont désormais une résolution étendue à 12 bits (analogWrite (0 ... 4095)), qui peut être changée par analogWriteResolution (12).

Encore quelques mots sur les protocoles des interfaces. En plus de l'FC, TWI (Two-Wire Interface) et SPI (Serial Peripheral Interface), on trouve aussi le bus CAN (Controller Area Network), qui est utilisé aussi bien dans l'automatisation de maquettes de chemin de fer que pour la mise en réseau de systèmes divers ou d'organes de commande dans les automobiles. L'interface USB OTG permet également de raccorder une souris, un clavier ou un smartphone.

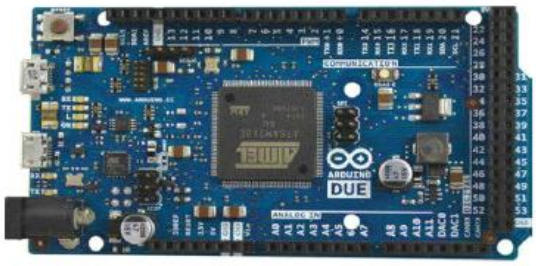


Figure1.26 : Arduino Due

Catégorie	Valeur
Microcontrôleur	*AT mega* SAM3X8E base sur une architecture ARM Cortex M3 à 32 bits
Fréquence d'horloge	84MHz
Tension de service	3,3 V (attention, ce n'est pas du *5* V !)
Tension d'entrées (recommandée)	7-12 V
Tension d'entrées (limite)	6-20V
Ports numériques	54 entrées et sorties (12 sorties commutables en MLI)
Ports analogiques	12 entrées analogiques
Courant maxi. Par broche 3,3 V	800mA
Courant maxi. Par broche *5* V	800mA
Mémoire	512 Ko Flash, 94 Ko SRAM {2 banes: 64 Ko+ 32 Ko), 512 octets EEPROM

Tableau 9 : Caractéristique technique d'Arduino Due

- **Avantages :**
  - Processeur ARM 32 bits.
  - Nombreuses entrées et sorties pour raccorder des capteurs ou des actionneurs.
  - Broches UART étendues (4 ports de communication série matériels).
  - Connection USB-OTG (On-The-Go).
  - Deux CNA (convertisseur numérique-analogique) qui peuvent être utilisés pour générer des signaux audio, par exemple.

- Deux bus TWI (Two Wire Interface).
- Peut recevoir des shields Arduino à condition qu'ils fonctionnent en 3,3 V et qu'ils soient conformes avec le brochage de l'Arduino 1.0 (à vérifier absolument).

- **Inconvénients :**
  - Tension de fonctionnement de 3,3 V ! Ne pas utiliser de shields Arduino qui fonctionnent en 5 V.

9. Arduino Yun :

La carte Arduino Yun est la première à être équipée de deux processeurs. Du côté Arduino, il y a un microcontrôleur de type ATmega32U4 qui se charge de l'exécution des sketches. De l'autre côté, il y a la machine Linux. Comme vous l'avez bien lu : c'est un processeur du type Atheros AR9331 fonctionnant sous Linino, une distribution Linux basée sur OpenWRT. Les deux couches sont interconnectées par un bridge afin d'échanger des informations ou des données. Ce bridge est un logiciel qui peut être utilisé par les processeurs.

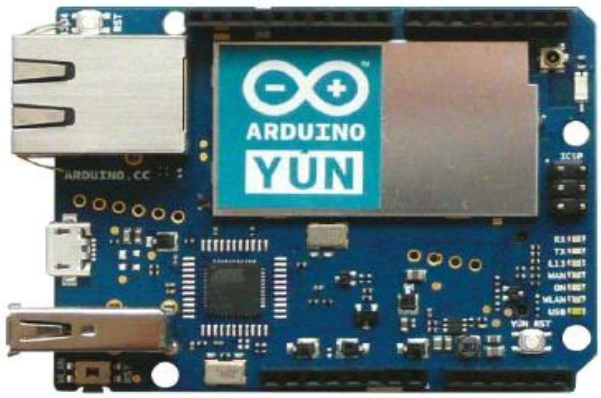


Figure1.27 : Arduino Yun

Spécifications de la carte Arduino Yun :

Microcontrôleur AVR :

Catégorie	**Valeur**
Microcontrôleur	ATmega 32U4
Fréquence d'horloge	16MHz
Tension de service	5V
Tension d'entrées	5V
Ports numériques	20 entrées et sorties (7 sorties commutables en MU)
Ports analogiques	12 entrées analogiques
Courant maxi. Par broche 3,3 V	50Ma

Courant maxi. Par broche 5 V	40Ma
Mémoire	32 Ko Flash, 2,5 Ko SRAM, 1 Ko EE PROM
Dimensions	cmxS,3 cm

Tableau 10 : Caractéristique technique d'Arduino Yun (microcontrôleur AVR)

Processeur Linux :

Catégorie	**Valeur**
Processeur	Atheros AR9331
Architecture/fréquence d'horloge	MIPS@400MHz
Tension de service	3,3V
Ethernet	IEEE 802.310/100 Mbit/s
Wi-Fi	EEE 802.11 b/g/n
USB Type-A	2.0 Host/Device
Lecteur de cartes	Pour Micro-SD uniquement
Mémoire	64 Mo DDR2 RAM, 16 Mo Flash

Tableau 11 : Caractéristique technique d'Arduino Yun (Processeur Linux)

Avantages :

Combinaison d'une carte Arduino standard basée sur une carte Leonardo avec un microprocesseur Linux (système d'exploitation Linino basé sur OpenWRT) ;

Module Wi-Fi préinstallé ;

Interface Ethernet préinstallée ;

Les environnements Arduino et Linux peuvent échanger des données ou des informations via un bridge ;

Nombreuses possibilités d'extension par le biais d'API préprogrammées disponibles gratuitement sur le site Temboo (Https://temboo.coml). Voir le montage n° 20. Il s'agit notamment de collections de services web, comme Twitter ou Google+, facilement accessibles via une couche d'abstraction normalisée à l'aide de fonctions homogènes. Les bibliothèques disponibles réunissent plus d'une centaine d'API ;

Possibilités d'extension logicielle par carte micro SD.

Inconvénients :

Plus énergivore de sorte que le port USB 2.0 peut vite atteindre

Ses limites.

Conclusion

Dans ce chapitre, nous avons axé notre étude sur la présentation de la carte Arduino et son environnement de programmation. Nous avons commencé par expliquer l'origine et la définition de cette carte, avant de décrire en détail ses aspects matériels, notamment les différents modèles et composants qu'elle intègre. Ensuite, nous avons présenté le logiciel de programmation associé, en détaillant sa structure et son fonctionnement. La maîtrise des concepts abordés dans ce chapitre est essentielle pour comprendre l'utilisation et la programmation de la carte Arduino dans la suite de notre projet.

Chapitre 02

CHAPITRE 02 : GENERALITES SUR LES ASCENSEUR

I. Introduction

La nécessité pour les humains de se déplacer en hauteur a suscité leur réflexion sur la création d'un moyen de déplacement rapide et économe en énergie, tout en assurant un niveau de sécurité élevé. Cette réflexion s'est inspirée de l'ascenseur rudimentaire déjà utilisé au Moyen Âge pour soulever des charges lourdes. Au fil du temps, des améliorations significatives ont été apportées tant sur le plan technique que sur le plan esthétique pour développer des ascenseurs plus performants.

Aujourd'hui, les ascenseurs sont devenus des équipements indispensables dans la vie quotidienne, garantissant un accès facile et sécurisé aux différents étages des bâtiments. Ils jouent un rôle essentiel dans la facilitation des déplacements des individus, en particulier des personnes handicapées.

Histoire

L'histoire de l'ascenseur remonte à l'Antiquité, où Archimède avait mis au point un treuil comportant des cordes et des poulies, la corde s'enroulant sur un tambour actionné par l'homme.

Au Moyen Âge, des treuils servaient à monter des personnes et des marchandises dans des endroits isolés, tels que les monastères.

En 1835, un monte-charge à vapeur, appelé le « teagle », servait à monter les matériaux dans une usine anglaise.

En 1845, Sir William Thompson mit au point le premier élévateur hydraulique, dont la commercialisation ne débutera qu'en 1850. Il faut attendre 1852 pour voir le premier « ascenseur » sécurisé grâce à l'invention d'Elisha Graves Otis, qui réalisa un « parachute ».

Le 23 mars 1857, Otis inaugure le premier ascenseur du monde dans un magasin de porcelaine. Le bâtiment comportait cinq étages et l'ascenseur fonctionnait grâce à une série d'arbres et de courroies entraînés par une centrale à vapeur. Sa capacité était de 450 kilogrammes, à une vitesse de 0,2 mètre par seconde.

En 1868, Otis mit au point un ascenseur à vapeur avec des dispositifs de sécurité si élaborés que les étages supérieurs prirent de la valeur, entraînant une augmentation de la hauteur des bâtiments.

En 1878, Otis proposa deux nouveaux produits : un ascenseur hydraulique très rapide (de 3 à 4 mètres par seconde) et un dispositif de parachute actionné par un limiteur de vitesse, permettant un arrêt progressif en cas d'urgence.

En 1880, c'est en Allemagne que l'idée d'un ascenseur électrique fut développée. Les moteurs électriques et transformateurs se développèrent alors pour l'industrie et la traction ferroviaire, utilisant le câble et le treuil fixe pour tracter les trains à locomotive dans les fortes côtes.

II. Définition

Un ascenseur est un dispositif de levage qui comprend une cabine conçue pour transporter des personnes ou des marchandises entre différents niveaux. Les ascenseurs se déplacent le long de guides verticaux ou inclinés avec une pente inférieure à 15 degrés.

III. Type d'ascenseur

Selon leur utilisation, les ascenseurs peuvent être classés en trois catégories principales :

1. Monte-charge

Il s'agit d'un type spécifique d'ascenseur, caractérisé par des dimensions généreuses de la cabine et une construction robuste, permettant le transport de charges lourdes et de personnes.

2. Monte-charge industriel

Également appelés "monte-charges industriels", ces ascenseurs sont des dispositifs de levage comprenant une cabine ou une plateforme spacieuse dédiée exclusivement au transport de marchandises entre différents niveaux. Leur commande est externe, c'est-à-dire qu'elle ne peut être effectuée que depuis l'extérieur de la cabine, et leur utilisation pour le transport de personnes est strictement interdite. Néanmoins, ce type d'ascenseur reste accessible aux personnes pour le chargement ou le déchargement de marchandises.

3. Ascenseur pour des personnes

Conçus exclusivement pour le transport de personnes, ces ascenseurs offrent un niveau de sécurité supérieur et un confort accru par rapport aux types précédents. Ils sont couramment

installés dans des édifices publics et des résidences privées, assurant, par exemple, le déplacement des occupants d'une maison individuelle. Dans cette catégorie, on trouve des modèles spécialement adaptés pour transporter des personnes handicapées, qu'elles soient debout ou en fauteuil roulant, accompagnées ou non.

IV. Classification des ascenseurs

En fonction de la méthode de propulsion, il est possible de diviser les ascenseurs en deux classifications principales :

- **Ascenseurs hydrauliques**
- **Ascenseurs à traction électrique** (ou à traction à câbles)

1. Les ascenseurs hydrauliques

Le fonctionnement des ascenseurs hydrauliques repose sur un système de vérin. La cabine est déplacée grâce au piston d'un vérin alimenté par de l'huile sous pression fournie par une centrale hydraulique (une pompe).

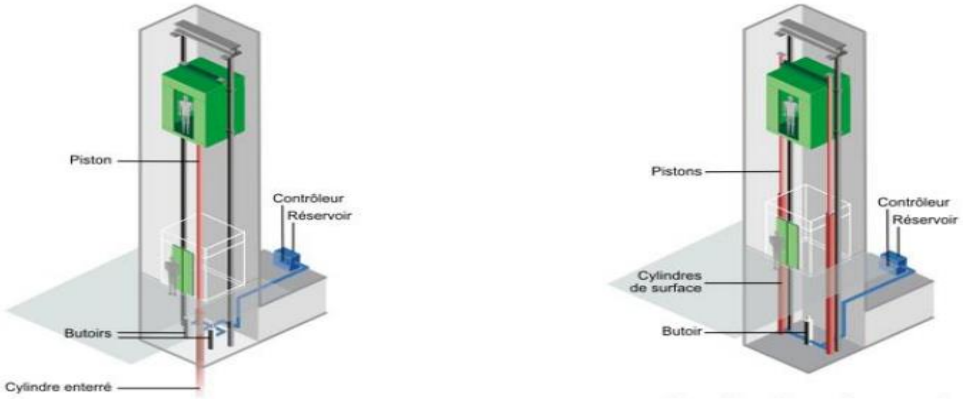
L'huile sous pression pousse le piston hors du cylindre, entraînant le déplacement de la cabine vers le haut. Pour descendre, une vanne (by-pass) de la pompe s'active, permettant à l'huile de s'écouler du cylindre vers un réservoir dans un système en circuit fermé. Cette action réduit la pression, facilitant la descente de la cabine.

Un inconvénient majeur des ascenseurs hydrauliques est l'absence de contrepoids pour équilibrer la cabine, entraînant une consommation énergétique élevée. De plus, ces ascenseurs offrent des déplacements plus lents et sur de plus courtes distances par rapport aux ascenseurs à traction. Leur course maximale est généralement limitée à environ 18 mètres.

Les ascenseurs hydrauliques peuvent être installés à l'intérieur d'une structure maçonnée ou d'un pylône métallique, appelé "ascenseur hydraulique en pylône".

Diverses variantes d'ascenseurs hydrauliques sont disponibles :

- À cylindre enterré.
- À cylindre de surface Télescopique.



Ascenseur à cylindre D'entrée

Ascenseur à cylindre de surface

Figure 2.1. Composant de l'ascenseur hydraulique

L'ascenseur hydraulique est composé notamment des éléments suivants :

- La cabine : où les usagers se positionnent pour se faire transporter.
- Les guides verticaux : assurent le déplacement de la cabine.
- La gaine ou conduit : structure de l'ascenseur dans laquelle la cabine s'insère.
- Les portes palières : situées à chaque étage, permettent l'accès au palier.
- Les portes cabines : assurent la séparation entre la cabine et la gaine. Ces portes s'ouvrent en parallèle aux portes palières quand l'ascenseur est en marche normale.
- Les pistons : reliés à la cabine, permettent son mouvement.
- La machinerie : inclut l'armoire de contrôle et le réservoir d'huile.

Les différents modèles d'ascenseurs hydrauliques sont conçus en fonction de certains critères, tels que :

- La hauteur de l'immeuble à desservir, car la hauteur de déplacement est limitée.
- La stabilité du sol et du sous-sol.
- Le risque de pollution vis-à-vis du sol, en particulier en ce qui concerne les nappes phréatiques.

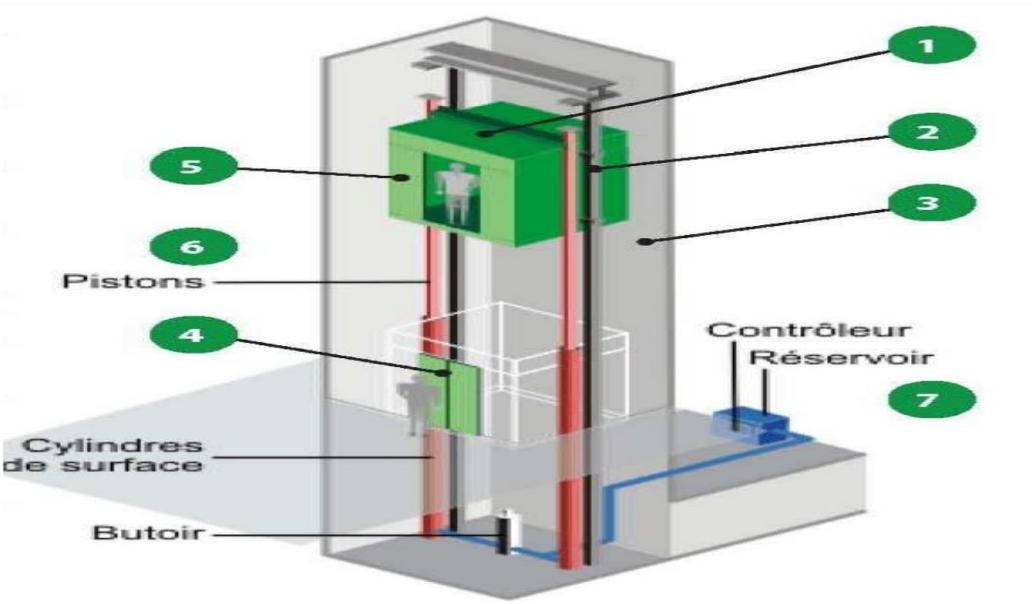


Figure 2.2. Composant de l'ascenseur hydraulique

1.1.Les avantages de l'ascenseur hydraulique

- Faible niveau sonore.
- Maintenance relativement aisée.
- Grande capacité de charge.
- Ajustement simple de la vitesse de déplacement.
- Absence de nécessité d'un local de machinerie.
- Implémentation facile, notamment dans un immeuble existant.
- Les déplacements de la cabine sont précis et s'effectuent en douceur.

1.2.Les inconvénients de l'ascenseur hydraulique

- Consommation d'énergie significative.
- Consommation importante d'huile, ce qui pose des défis en matière de sécurité incendie.
- Vitesse de déplacement relativement lente.
- Nécessité de renforcer la dalle de sol lors de l'installation.
- Risque de pollution des niveaux inférieurs.
- Limitation en termes de hauteur de déplacement.

2. Les ascenseurs à traction à câbles

Les ascenseurs à traction par câbles sont les plus couramment utilisés et sont fréquemment empruntés au quotidien, en particulier dans les bâtiments tertiaires. En fonction du système de propulsion, plusieurs catégories d'ascenseurs à traction par câbles existent :

- **À moteur-treuil planétaire**
- **À moteur sans treuil** ou « Gearless »
- **À moteur-treuil avec une vis sans fin**

La cabine et son contrepoids, suspendus dans la gaine, sont reliés par des câbles de traction. Le déplacement de l'ensemble cabine-contrepoids est réalisé par un moteur électrique qui actionne une poulie. Cette poulie entraîne les câbles de traction, provoquant le déplacement simultané de la cabine et du contrepoids dans des directions opposées. Il existe deux variantes en fonction du mode de traction :

- **Traction directe** : Le treuil à tambour tire directement la cabine. Ce mode de traction présente l'avantage d'une empreinte réduite, mais il est principalement adapté aux applications nécessitant des charges légères.
- **Traction par moufle** : Ce mode de traction est utilisé pour des charges plus importantes.

Les ascenseurs à traction par câbles sont composés des éléments suivants :

- Une cabine
- Un contrepoids
- Des câbles reliant la cabine au contrepoids
- Des guides
- Un système de traction
- Un dispositif de sécurité antichute

Les ascenseurs à traction par câbles sont considérablement plus économes en énergie que les ascenseurs hydrauliques. Cette économie est due à la présence du contrepoids, qui équilibre le poids de la cabine, réduisant ainsi significativement la demande en énergie et en puissance.

2.1.Les avantages des ascenseurs à traction pas câbles

En comparaison aux ascenseurs hydrauliques, les ascenseurs à traction par câbles offrent plusieurs avantages :

- Réalise des déplacements rapides et précis.
- Préserve l'environnement.
- Présente une efficacité énergétique élevée.
- Facilite le déplacement des utilisateurs.
- Offre une grande flexibilité en termes de vitesse.
- Limite la consommation d'énergie et les besoins en puissance.

2.2.Les inconvénients des ascenseurs à tractions par câble

- Un entretien régulier doit être effectué pour garantir le bon fonctionnement de l'appareil, avec une attention particulière à la sécurité des utilisateurs.
- Dans la version standard des ascenseurs, l'installation d'une structure technique sur le toit est nécessaire.
- Il est essentiel de prendre en compte le poids de la cabine, des câbles, du contrepoids, de la structure de la salle des machines et des équipements de cette dernière. L'ensemble de ce poids repose sur la structure du bâtiment, comme les colonnes ou les murs porteurs renforcés, et influe sur les fondations.
- La présence d'un volume construit sur le toit peut avoir un impact esthétique indésirable.
- Des problèmes d'accessibilité peuvent se poser.
- La compacité de la gaine est limitée en raison de la présence de la cabine et du contrepoids, ce qui réduit la surface utilisable aux étages du bâtiment.

Il est à noter que chaque modèle d'ascenseur électrique présente des avantages spécifiques, et le choix dépendra des besoins de l'application particulière. Voici les éléments composant un ascenseur à traction par câbles :

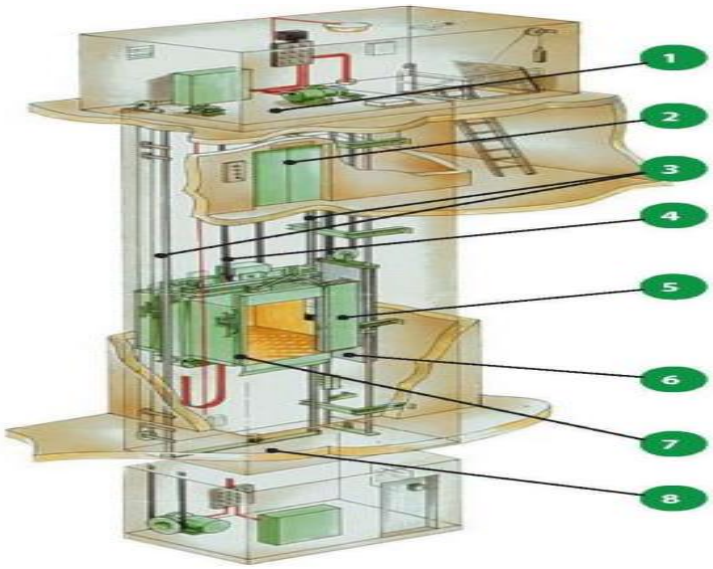


Figure 2.3 :

Les éléments d'un ascenseur

1. Machinerie : Inclut un moteur qui fournit la force nécessaire au mouvement de la cabine à travers les câbles.
2. Portes palières : Situées à chaque étage, elles permettent l'accès au palier.
3. Guides verticaux : Assurent le déplacement de la cabine.
4. Câbles : Réalisent le déplacement de la cabine.
5. Portes cabines : Séparent la cabine de la gaine. Ces portes s'ouvrent en parallèle des portes palières lorsque l'ascenseur est en marche normale. Les portes de la cabine entraînent les portes palières lorsque la cabine est au niveau de celles-ci.
6. Contrepoids : Assure le contrebalancement. Le contrepoids est généralement 50 % plus lourd que la cabine vide, et la cabine devient 50 % plus lourde que le contrepoids lorsqu'elle est en pleine charge.
7. Cabine : Où les usagers se positionnent pour se faire transporter.
8. Gaine ou conduit : Structure de l'ascenseur dans laquelle s'insère la cabine.

La machinerie peut se trouver :

- En haut (suspension directe).
- En bas (partie inférieure, longueur des câbles trois fois plus grande).
- Latéralement (en partie haute ou basse).
- La gaine ou conduit est la structure de l'ascenseur dans lequel il s'insère.

V. Critères de choix des ascenseurs

Les critères de choix d'un ascenseur incluent :

- Facteurs de construction : Hauteur du bâtiment, espace disponible aux différents étages, possibilité d'installer une salle des machines en haut de la gaine, stabilité du terrain, etc.
- Facteurs organisationnels : Type d'utilisation du bâtiment, taux d'occupation, mode de fonctionnement, niveaux de confort (comme le temps d'attente moyen maximal et la vitesse de déplacement de la cabine), et trafic (rapport entre la capacité de charge et la vitesse).
- Considérations énergétiques : Consommation d'énergie et demande de puissance (par exemple, le courant de démarrage), qui influencent directement le choix de la technologie.
- Préoccupations en matière de sécurité : La sécurité est la priorité principale des fabricants d'ascenseurs et un critère déterminant dans le choix de la technologie appropriée.

VI. La Sécurité des Ascenseurs : Risques et Contre-mesures

- Dangers potentiels dans les ascenseurs :

1. Blocage de la cabine entre deux étages, pouvant causer de la panique ou de l'anxiété chez les passagers.
2. Surcharge : Le risque que la cabine dépasse la capacité de charge, ce qui pourrait endommager le système.
3. Défaillance de l'alimentation électrique : Panne de courant pouvant immobiliser l'ascenseur avec des passagers à l'intérieur.
4. Risque de chute libre : Défaillance des câbles ou du système de traction pouvant entraîner une chute de la cabine.

5. Piégeage des personnes : Risque de coincement des passagers par les portes.
6. Incendie : Danger accru si l'ascenseur n'est pas équipé de systèmes de détection et de protection incendie.
7. Interférence avec des objets étrangers : Objets coincés dans les portes ou dans le mécanisme de l'ascenseur.

- Contre-mesures pour assurer la sécurité des usagers :

1. Systèmes de ventilation dans la cabine pour assurer le confort en cas de blocage.
2. Dispositifs d'alerte et de communication d'urgence : Téléphones d'urgence et alarmes pour demander de l'aide.
3. Détecteurs de surcharge : Empêchent l'ascenseur de démarrer si la charge maximale est dépassée.
4. Alimentation de secours : Batterie de secours ou générateur pour assurer le retour au niveau le plus proche en cas de panne.
5. Frein parachute et limiteur de vitesse : Empêchent les chutes libres en cas de rupture des câbles.
6. Capteurs de détection d'obstacles : Équipent les portes pour éviter les accidents lors de la fermeture.
7. Dispositifs antichute et système de secours manuel : Permettent une descente en toute sécurité en cas de défaillance.

.

VII. Les éléments garantissant la sécurité des ascenseurs actuels comprennent

- Systèmes de ventilation et d'aération permanente dans la cabine.
- Portes coulissantes automatiques des cabines, dotées de dispositifs d'arrêt en ouverture et en fermeture, entre autres.
- Dispositifs d'alerte en cas de blocage de la cabine entre deux étages, comprenant des téléphones, des boutons d'alarme, des caméras, etc.
- Détecteurs d'obstacles, tels que des capteurs empêchant la fermeture des portes en présence d'objets ou lors du passage de personnes.

- Des dispositifs spécifiques pour chaque type d'ascenseur, permettant de l'empêcher de démarrer si la charge maximale (poids de la cabine et des passagers) est dépassée.
- Des dispositifs de sécurité tels que le frein parachute, le limiteur de vitesse, le dispositif antichute et de secours (permettant une descente manuelle en cas de panne de courant.

VIII. Conclusion

Les ascenseurs continuent d'évoluer pour améliorer l'expérience des utilisateurs et réduire les coûts de production. Bien qu'ils occupent une place de choix dans les systèmes de transport vertical, des défis persistent, notamment en ce qui concerne la gestion des flux de personnes dans les grands bâtiments et le contrôle de la vitesse dans les édifices de grande hauteur.

Dans ce deuxième chapitre, nous avons étudié en profondeur les différents types d'ascenseurs, en analysant leurs avantages et inconvénients, ainsi que leurs critères de performance énergétique. Nous nous sommes concentrés principalement sur les ascenseurs à traction électrique, qui sont les plus utilisés aujourd'hui, en raison des avancées technologiques récentes qui les rendent plus efficaces et mieux adaptés aux besoins modernes

.

Chapitre 03

I. Introduction :

Dans les chapitres précédents, nous avons abordé les généralités concernant les cartes Arduino et leur microcontrôleur, ainsi que les principes fondamentaux des ascenseurs. Dans ce chapitre, nous allons nous concentrer sur l'étude et la conception d'un ascenseur. Nous examinerons en détail tous les composants nécessaires et présenterons le code qui permettra de réaliser efficacement le fonctionnement d'un ascenseur.

1. Objectif :

L'objectif est de réaliser une commande d'ascenseur à l'aide d'un microcontrôleur Arduino (Mega 2560), en utilisant une maquette représentative d'un ascenseur. Cette maquette inclut un moteur CC (courant continu) situé au sommet, qui contrôle le mouvement vertical de la cabine.

II. Définition des composant utiliser :

1. Arduino Mega 2560 (Déjà définie dans le 1er chapitre page 24)

2. Moteur à courant continu :

Dans notre projet, nous utilisons un moteur à courant continu (CC) pour contrôler le mouvement de la cabine, permettant ainsi de la faire monter et descendre. Ce moteur fournira la puissance nécessaire pour assurer un déplacement fluide et efficace de la cabine, répondant ainsi aux exigences de notre système de transport vertical.

Il y a deux parties électriques dans un moteur à courant continu : le stator et le rotor. Quand le moteur est alimenté, une interaction magnétique se produit qui entraîne le mouvement du moteur. Quand la tension qui alimente le moteur est inversée, il tourne dans le sens opposé.

3. Module L298N :

Le module L298N est un circuit intégré contient essentiellement :

- 2 ponts en H, permettant de piloter chacun 1 moteur électrique DC (dans un sens, ou dans l'autre)
- Et une logique de commande à « faible courant », pour piloter ces ponts à « fort courant »

Le L298N requiert 2 alimentations distinctes pour fonctionner :

- Une tension pour la partie puissance (nommée Vs), qui servira à alimenter les moteurs, au travers de transistors de puissance
- Une tension pour la partie commande (nommée Vss), qui servira à alimenter toute la partie logique de commande, dont ces transistors de puissance.

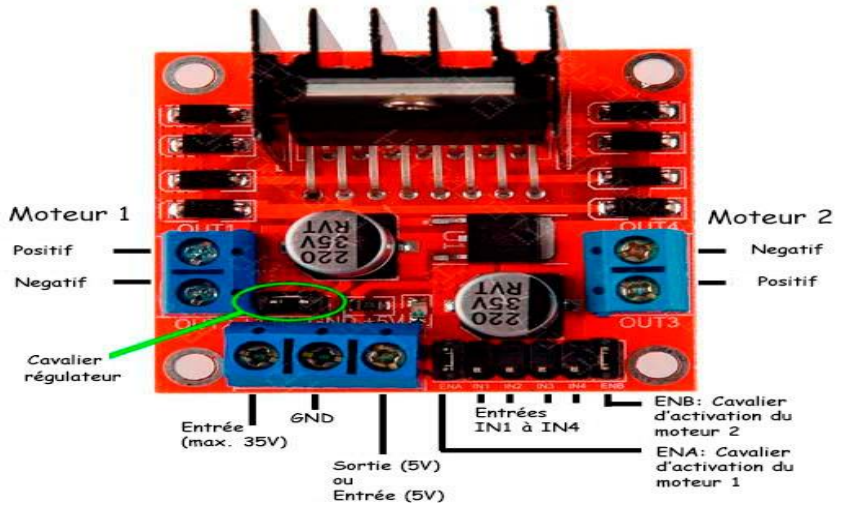


FIGURE 3.1. Module L298n

3.1. Principe de fonctionnement :

Le L298N est construit en deux ponts en H. Dans l'électronique, le pont en H est une configuration de quatre transistors ou d'autres commutateurs que l'on peut contrôler pour inverser la polarité du courant qui, à son tour, pour inverser le sens du courant qui circule à travers un moteur. Il est appelé ainsi parce qu'un diagramme de transmission montre des lignes obliques le long desquelles se trouvent des transistors formant un H. expliquons simplement le fonctionnement du pont en H pour les moteurs. En utilisant deux transistors à la fois, par exemple 1 et 2, il est possible de contrôler le courant qui varie en fonction de la charge connectée au moteur. Ainsi, sens fin et M1 tourne dans le sens inverse des aiguilles d'une montre. Si vous changez le sens du courant, le moteur tournera dans la direction opposée.

3.2. Les caractéristiques principales de module :

- Nombre de Moteurs : Joue le rôle de contrôleur pour 2 moteurs à courant continu ou moteur pas à pas.
- Tension d'Alimentation : 5 V à 35 V.
- Courant de Sortie : Jusqu'à 2 A par canal (3 A en crête).
- Contrôle de Vitesse : Compatible avec le contrôle par PWM (modulation de largeur d'impulsion).
- Dissipation de Chaleur : Équipé d'un dissipateur thermique.

- Alimentation Séparée : Permet l'utilisation de différentes sources d'alimentation pour les moteurs et la logique.
- Interface : Commande via signaux logiques HIGH/LOW pour la direction et l'activation.

4. Capteur Infrarouge :

Dans notre projet, nous allons utiliser des capteurs infrarouges à chaque comme détecteurs de position. Cela nous permettra de gérer à la fois la destination et la position de la cabine. Ces capteurs fourniront des informations précises sur l'emplacement de la cabine, facilitant ainsi le contrôle et la navigation dans le système de transport vertical.

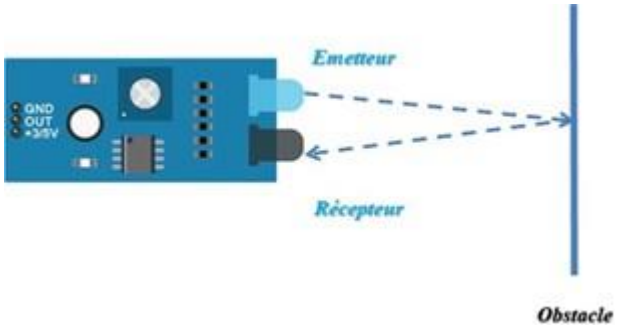


FIGURE 3.2. Les ondes infrarouges

4.1. Définition :

Le capteur infrarouge est un type de dispositif électronique qui capte le rayonnement infrarouge émis par des objets devant lui.

Depuis son nom, le capteur IR est composé de deux parties principales, à savoir l'émetteur IR et le récepteur IR. Il transmet des ondes infrarouges alors que le rôle du récepteur est de capter ces ondes. Le récepteur IR transmet des informations numériques en permanence sous forme de 0 faible ou haut 1 à la broche V out du capteur.

4.2. Caractéristiques de IR :

- Alimentation Polyvalente : Compatible avec 3,3 à 5 V.
- Portée Réglable : De 2 à 30 cm pour une détection sur mesure**.**

5. Bouton poussoir :

Un bouton poussoir est un composant électromécanique qui peut ouvrir ou fermer un circuit électrique. Il est conçu pour être enfoncé à la main, et il existe 2 types de bouton, normalement ouvert (no) ou normalement fermé (NC).

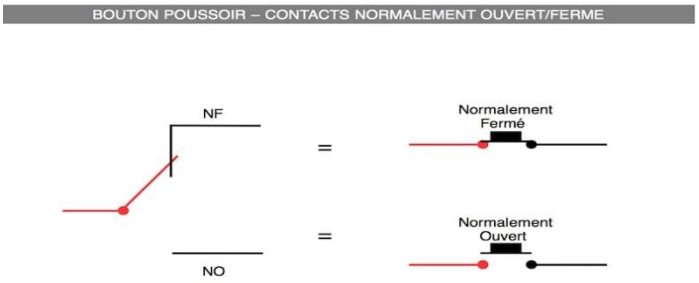


FIGURE 3.3. Boutons poussoir NO et NF

- Le circuit est normalement ouvert, mais il se ferme lorsque le bouton est enfoncé.
- Le circuit est généralement fermé par défaut, et il s'ouvre lorsque le bouton est enfoncé.

6. Afficheur LCD 16X2 Rétroéclairage bleu et le module I2C :

Dans ce projet, nous utiliserons un afficheur LCD pour montrer l'étage où se trouve la cabine

6.1. Afficheur LCD :

FIGURE 3.4. Afficheur LCD

6.1.1. Définition :

LCD et l'abréviation du terme anglais **« Liquid Crystal Display**" qui signifie en francais " Ecran à cristaux liquides ".

L'afficheur LCD et le principale interface visuelle entre un système et l'homme, son premier rôle et de transmettre les informations utiles d'un système a un utilisateur.

Comme son nom l'indique LCD 16x2, il comporte 16 colonnes et 2 lignes. Il peut donc afficher 32 caractères au totale et chaque caractère et composé de 5x8 de points de pixels au totale 1280 pixels.

La couleur des caractères et blanche et l'écran a un fond bleu lisible a l'obscurité.

6.1.2. Les caractéristiques :

En consultant la datasheet du l'écran LCD on trouve les caractéristiques suivantes :

- Module d'affichage : STN, BLUB
  Mode d'affichage STN (Super Twisted Nematic) avec rétroéclairage bleu (BLUB).
- Format d'affichage : 16 caractères x 2 ligne.
- Données d'entrées : 4 bits ou 8 bits disponible.
- Police d'affichage : 5 x 8 points.
- Alimentation : Alimentation simple (5V ±10%).
- Schéma de conduite : 1/16 Duty, 1/5 bias.
  Le schéma de balayage est 1/16 Duty (une ligne activée sur 16) et 1/5 Bias (rapport de tension entre les segments activés et non activés).
- Direction de visualisation : 6 heures
  L'angle de visualisation optimal est situé à 6 heures, c'est-à-dire en bas de l'écran, correspondant à une meilleure lisibilité lorsque l'écran est vu sous un angle inférieur.

6.1.3. Les brochages :

- Alimentation (Power Supply) : Pins 1 à 3 pour la gestion de l'alimentation et du contraste.
- Contrôle des Signaux : Pins 4 à 6 pour les signaux de sélection de registre, de lecture/écriture, et d'activation.
- Bus de données (Data Bus) : Pins 7 à 14 pour la communication des données entre le microcontrôleur et le LCD, en 4 bits ou 8 bits.
- Rétroéclairage (Backlight) : Pins 15 et 16 pour la gestion du rétroéclairage du LCD.

6.2. Le module I2C :

Le module I2C veut dire en anglais (Inter Integrated Circuit) simplifie la connexion entre un écran LCD et un microcontrôleur en réduisant le nombre de broches nécessaires grâce au

protocole I2C, passant de 6 à 2 (SDA et SCL). Il utilise un circuit PCF8574 pour gérer les entrées/sorties et est compatible avec des écrans LCD 1602 et 2004, ainsi qu'avec des microcontrôleurs comme Arduino et Raspberry Pi. Le module inclut un potentiomètre pour ajuster le contraste et une résistance pour gérer le rétroéclairage.

FIGURE 3.5. Le module I2C

6.2.1. Les caractéristiques :

- Type de connexion : I2C.
- Circuit intégré : PCF8574.
- Nombre de broches : 4 (VCC, GND, SDA, SCL).
- Tension d’alimentation : 5V.
- Courant de consommation : dépend de l’écran LCD connecte
- Type d'écran LCD compatible : 1602 ou 2004.
- Réglage du contraste : potentiomètre.
- Contrôle du rétroéclairage : résistance.
-

6.2.2. Connection Arduino, LCD et I2C :

Un afficheur LCD peut être contrôlé par un module I2C, qui se connecte aux 16 broches de l'écran. Le module I2C simplifie la connexion en réduisant le nombre de fils nécessaires à seulement 4 : VCC, GND, SDA et SCL, allégeant ainsi le câblage et évitant d'encombrer les broches de l'Arduino.

Les cartes Arduino, comme la Mega 2560, possèdent des broches dédiées pour la communication I2C : SDA (broche 20) et SCL (broche 21). Ces broches permettent de communiquer

efficacement avec des périphériques compatibles I2C, comme un afficheur LCD, facilitant ainsi l'affichage des données avec un minimum de connexions.

7. Buzzer :

Un buzzer est une sorte de haut-parleur mais de faible puissance qui va émettre un son en fonction de la fréquence et amplitude de vibration. Il permet de jouer des notes et de recréer des mélodies simples

.

FIGURE 3.6. Un buzzer

III. Réalisation de l'ascenseur :

1. Présentations de prototype :

Notre système représente un ascenseur de 4 etages (R+3) il constitue :

- Une maquette de 125.5×31×41 cm^3
- Une cabine de 24.5×23×21 cm^3 et cm et un câble d'attraction.
- Un moteur cc
- Un module L298n pour la gestion du moteur.
- Un afficheur à LCD avec le module I2C.
- 4 LEDs pour indique l'étage où se trouve la cabine.
- 8 infrarouges pour détecter la position exacte de la cabine.
- 4 boutons d'appel extérieur et 4 boutons d'appel intérieur.
- Un microcontrôleur Arduino pour gérer tout l'ascenseur.

2. Schéma et simulation :

Pour pourvoir et simulé et schématisée notre ascenseur on utiliser le logiciel Proteus 8 qui un logiciel de conception et de simulation Electronique développer par Labcenter Electronics, il est 0utilisé par les spécialistes dans le domaine pour la conception de circuit électronique, la modalisation de microcontrôleur et la réalisation de projet électronique.

3. Conception, réalisation et programmation :

3.1. Principe de fonctionnement :

Lorsqu'un utilisateur se trouve devant l'ascenseur et souhaite l'appeler alors que la cabine n'est pas à son niveau, la carte Arduino enregistre cette requête. Le processeur analyse ensuite les différentes variables associées à cet appel et génère les signaux de commande nécessaires pour le moteur à courant continu.

Pour déterminer la position actuelle de la cabine, le processeur se base sur les informations fournies par les capteurs de présence infrarouges, et ces données sont affichées sur un afficheur à 7 segments. La décision d'arrêter la cabine à un étage donné repose sur un calcul de priorité élaboré par l'algorithme, ce qui garantit une gestion efficace et fluide des déplacements de la cabine.

3.2. Conception de circuit :

Notre ascenseur se base sur 3 circuits, un circuit de puissance pour commande du moteur, un circuit pour l'afficheur a 7 segments, et un circuit pour les boutons poussoirs et les infrarouges.

- Pour alimenter le moteur j'ai utilisé un module d'alimentation de 8v.

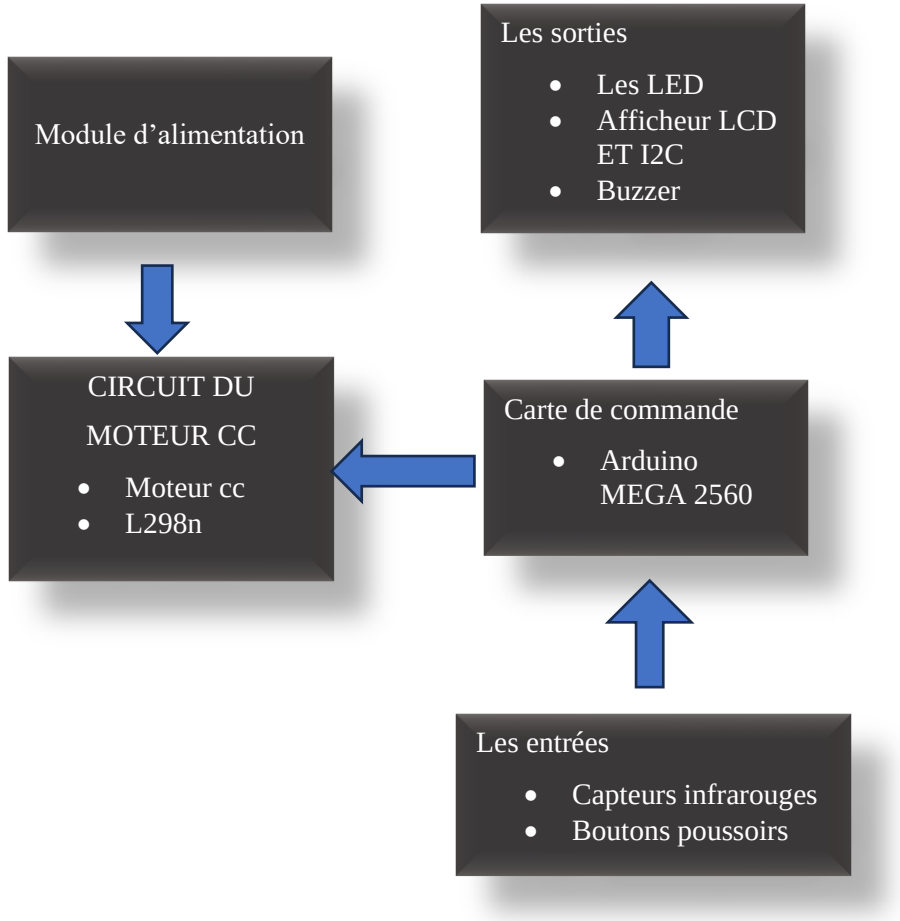


FIGURE 3.7. Schéma synoptique de la carte Arduino

3.3. Schémas électriques :

3.3.1. Circuit de puissance :

Cette figure représente le circuit de puissance (le moteur a cc et le module L298n).

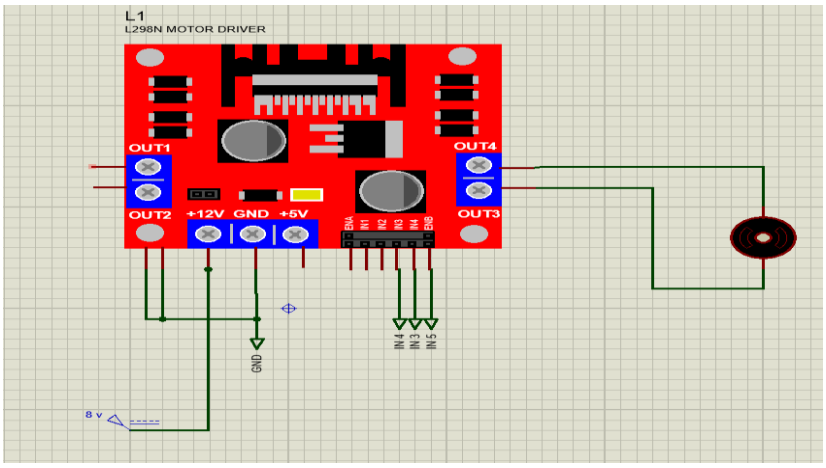


FIGURE 3.8. Schéma électrique du L298 et le moteur cc sur Proteus

3.3.2. Les entrées :

Cette figure montre les déférente entrées tel que les boutons poussoir et les capteurs infra-rouge.

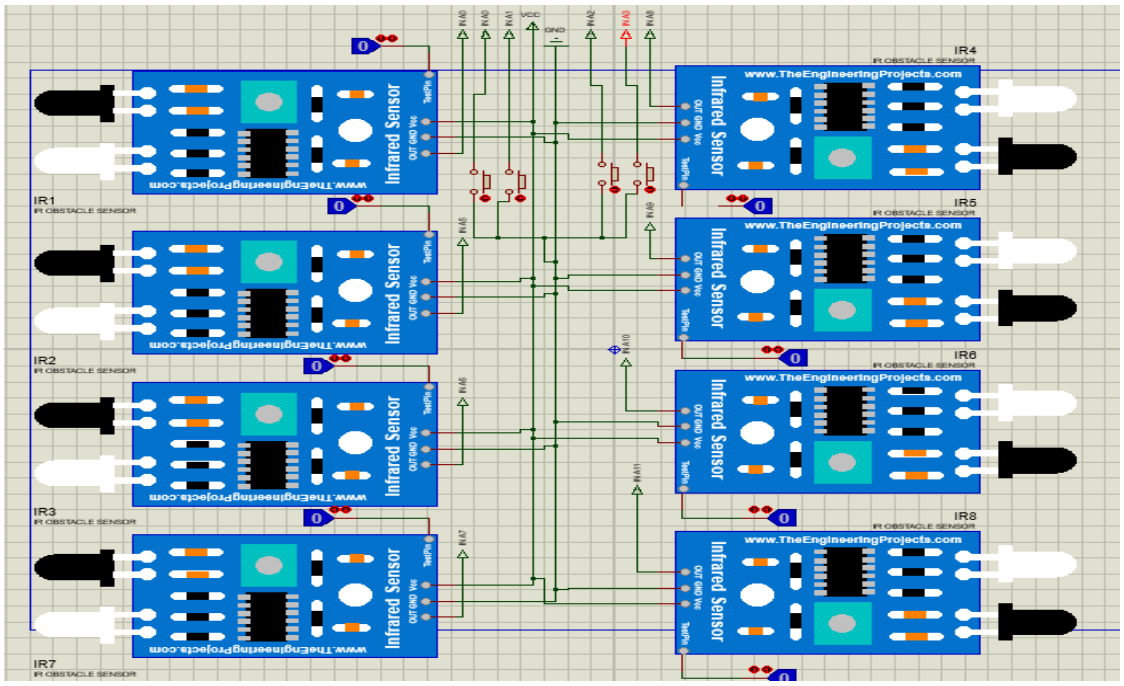


FIGURE 3.9. Schéma électrique des différentes entrées sur Proteus

3.3.3. Les sorties :

Dans ce schéma on définit les différentes sorties de la carte Arduino

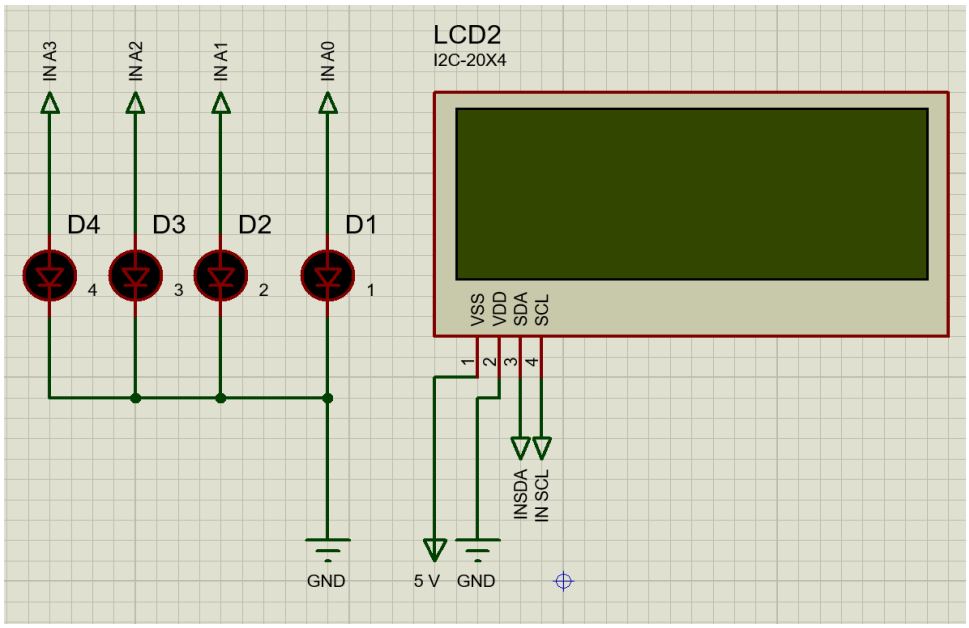


FIGURE 3.10. Schéma électrique de la différente sortie sur Proteus

3.3.4. Schéma globale :

Dans cette figure on illustre le schéma global de notre ascenseur.

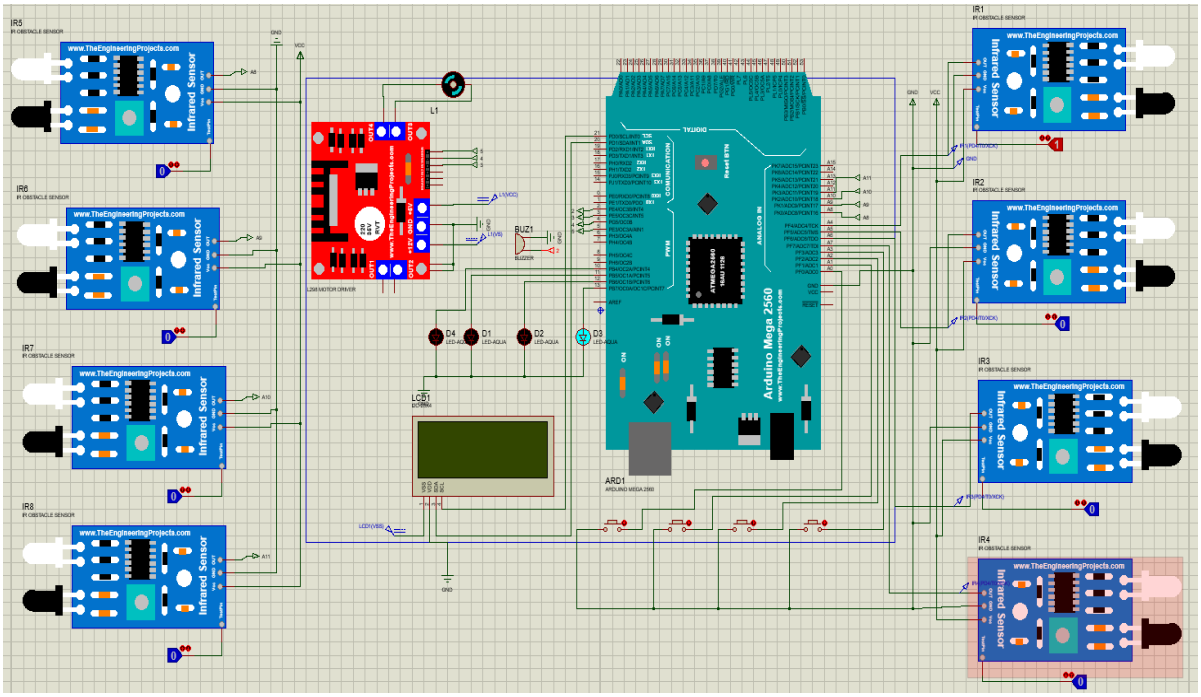


FIGURE 3.11. Schéma électrique globale de l'ascenseur sur Proteus

4. Programmation :

Pour pouvoir réaliser et bien structuré le programme nous avions partagé en un ensemble de sous-programme, qui sont appelés dans le programme principal.

Lors de la mise en marche de l'ensemble du projet, l'afficheur LCD affiche immédiatement la position actuelle de la cabine

3.4. Organigramme de déroulement du programme :

3.4.1. Programme principale (void Loop) :

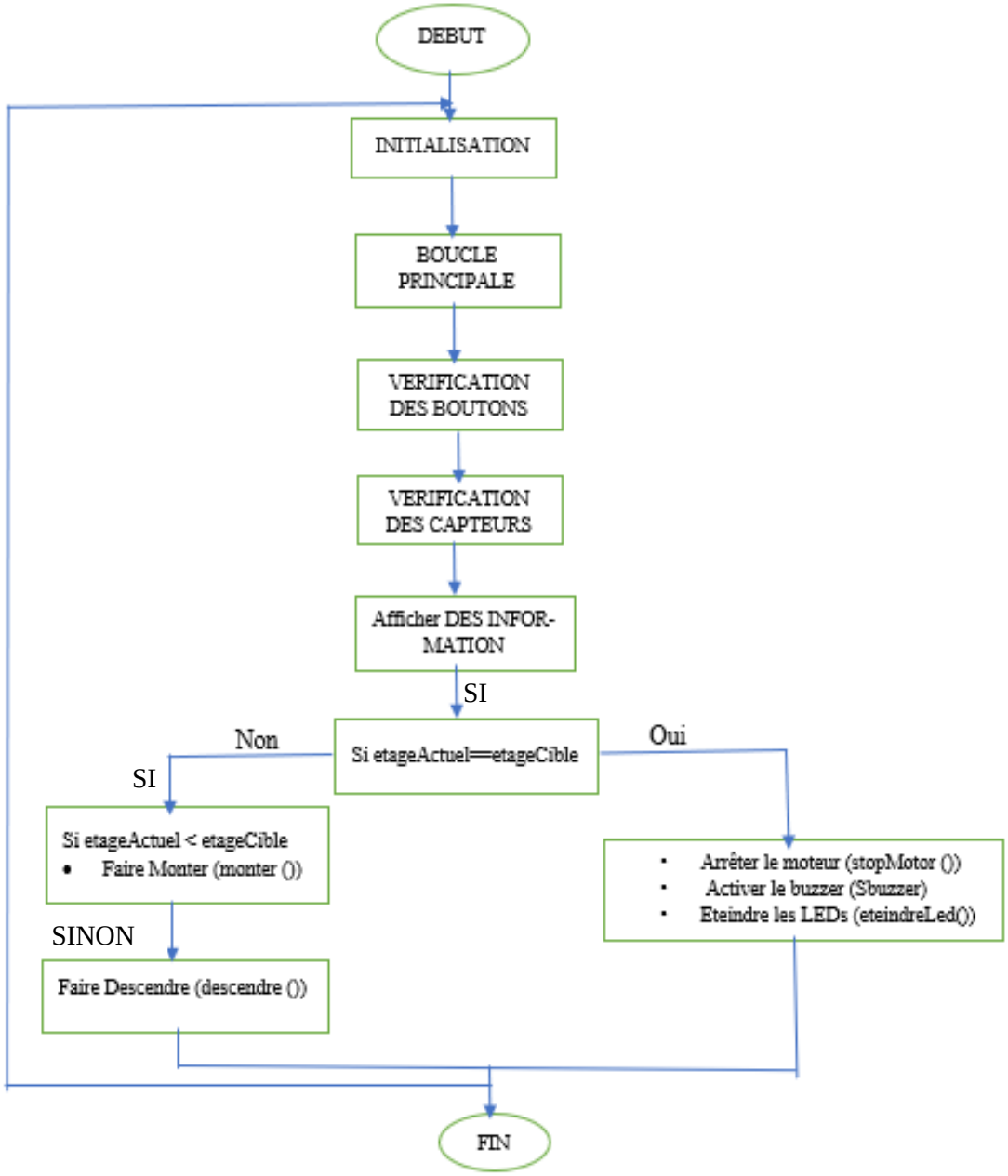


FIGURE 3.12. Organigramme du programme principal

3.4.2. Sous-programme vérification des boutons (void cible ()) :

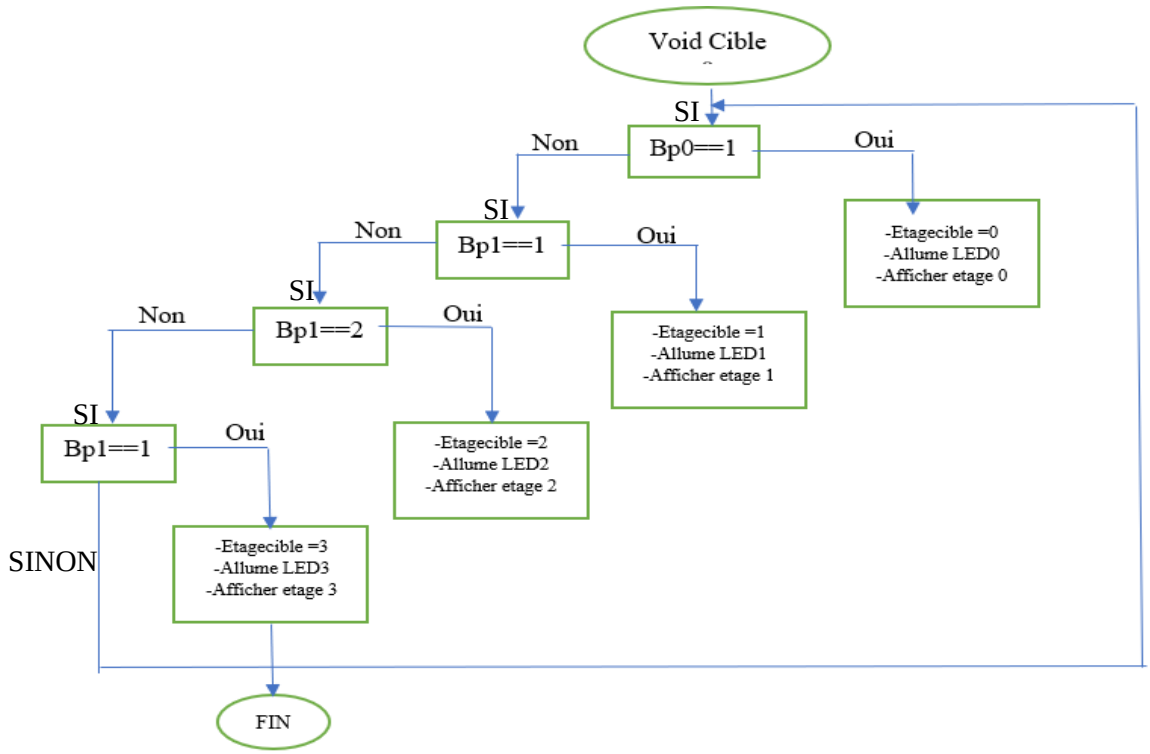


FIGURE 3.13. Organigramme du sous-programme de vérification des boutons

3.4.3. Sous-programme de vérification des capteurs infra-rouge (void detection ())

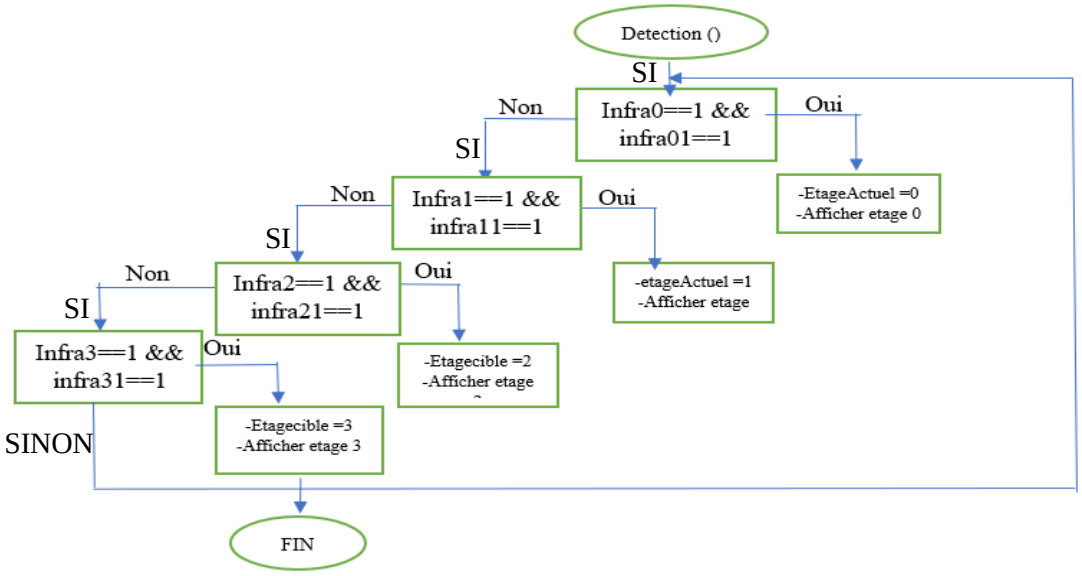


FIGURE 3.14. Organigramme du sous-programme de vérification des capteurs

3.4.4. Sous-programme Affichage des informations à partir de l'écran LCD :

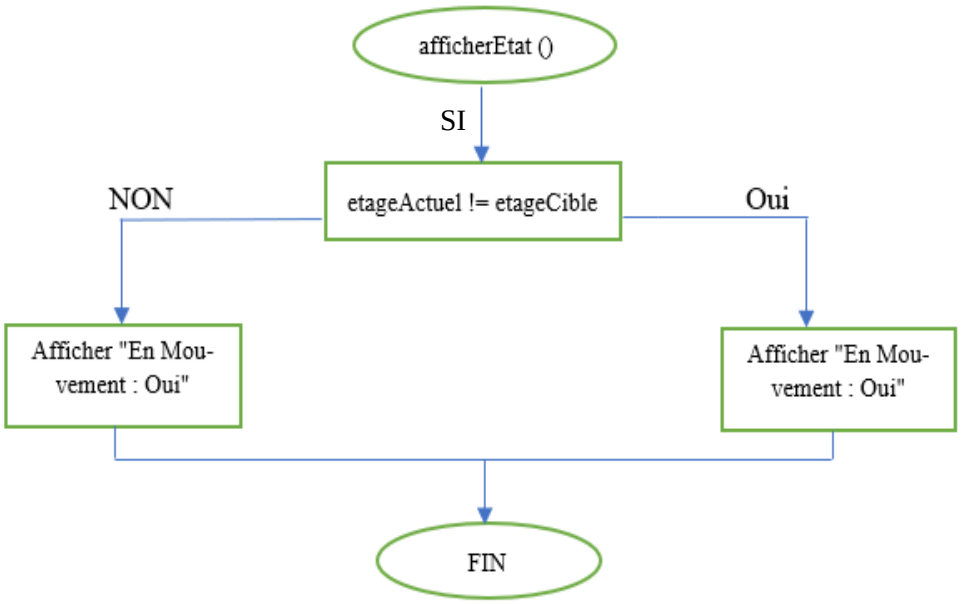


FIGURE 3.15. Sous-programme Afficher Des informations

Sous-programme, Descendre (), monter () et stopMotor :

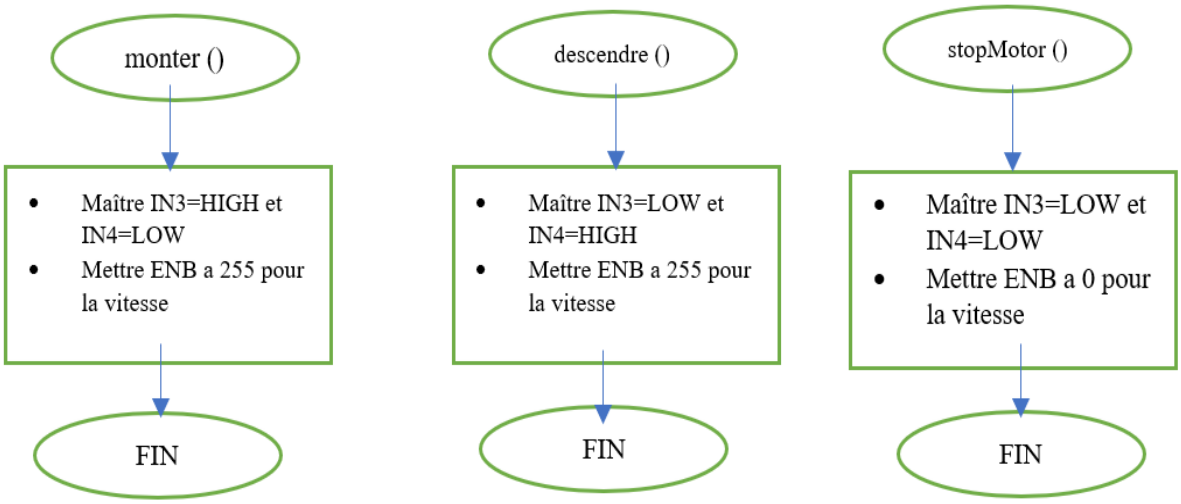


FIGURE 3.16. Sous-programme du deplacement et stop du moteur

3.4.5. Sous-programme allumerLed (), eteindreLed () et Sbuzzer () :

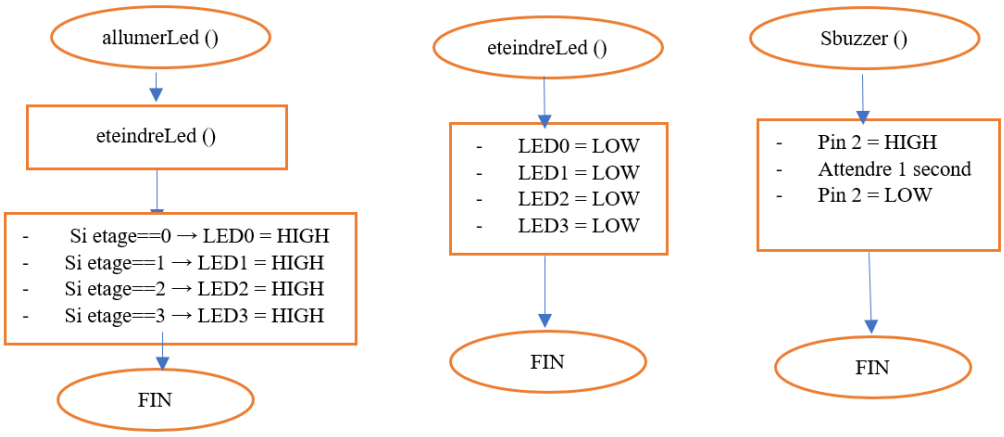


FIGURE 3.17. Sous-programme pour allumer et éteindre les LEDs et le buzzer

4. Réalisation de la maquette :

4.1. Description de la maquette :

4.1.1. Partie externe :

La maquette, réalisée en MDF (Panneau de fibres à densité moyenne), comprend 4 étages. Chaque étage est doté d'un bouton poussoir situé à droite de chaque etage, accompagné d'une LED qui indique l'étage sélectionné. La cabine, représenté par un module mobile fait en MDF, se déplace verticalement entre les étages. Au sommet de la maquette, un afficheur LCD est installé pour fournir des informations visuelles en temps réel sur la position et l'état de la cabine, offrant ainsi une interface claire et interactive, est le moteur à courant continu.

4.1.2. Partie interne :

À l'intérieur de la maquette, la cabine est suspendue à un câble métallique relié au moteur. De chaque côté de la cabine, deux roulettes sont fixées pour glisser le long de rails situés sur les murs gauche et droit, assurant ainsi un équilibre parfait lors des mouvements. Sur le mur droit, deux capteurs infrarouges sont installés à chaque étage, placés dans des encoches spécialement prévues pour détecter la position précise de la cabine. Un espace a été laissé entre le mur droit et le mur extérieur afin de permettre le passage du câblage, assurant ainsi une installation propre et fonctionnelle.

Figure 3.18 : Vue interne et externe de la maquette.

5. Test :

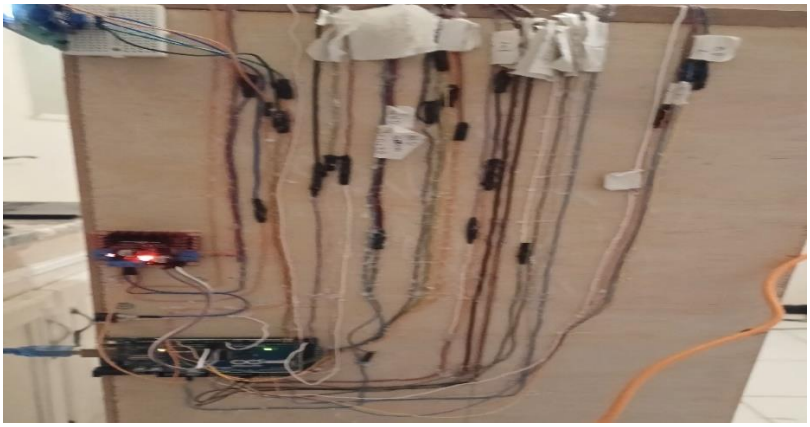


Figure 3.19 : branchements de la carte Arduino et du module L298.

Dans cette figure on voit le branchement de tous les composant vers la carte Arduino.

Figure 3.20 : Afficheur LCD indique l'etage actuelle.

Dans cette figure l'afficheur indique que la cabine est au première étage et elle en mouvement.

Figure 3.21 : Afficheur LCD indique l'etage actuelle.

Dans cette figure l'afficheur indique que la cabine est au troisième etage et qu'elle est arrivé à l'etage ciblé et que la cabine est en arrêt.

Figure 3.22 : Afficheur LCD indique l'etage actuelle.

Dans cette figure on voit que la LED et allumer lorsque le bouton correspondant a l'etage et appuyé.

IV. Conclusion :

Dans ce chapitre qui représentait l'élément central de notre projet de fin d'études, nous avons détaillé la conception et la réalisation de notre maquette d'ascenseur en expliquant le rôle de chaque composant utilisé. Nous avons démontré comment l'Arduino Mega 2560, le moteur à courant continu, le module L298N, les capteurs infrarouges, l'afficheur LCD, ainsi que les boutons poussoirs et les LEDs, interagissent ensemble pour gérer efficacement le déplacement de la cabine entre les différents étages. Chaque élément a été soigneusement intégré pour assurer un fonctionnement fluide, précis et sécurisé de l'ascenseur, avec une interface visuelle claire via l'afficheur LCD.

CONCLUSION GÉNÉRALE

CONCLUSION GÉNÉRALE

L’objectif initial de ce travail était de concevoir et de réaliser une carte de commande pour un ascenseur didactique en utilisant une carte Arduino Mega 2560, tout en adoptant une méthodologie permettant d’explorer les différentes étapes de fonctionnement de ces systèmes. Ce projet nous a permis de renforcer nos compétences techniques et d’appliquer concrètement nos connaissances à la réalisation d’une maquette fonctionnelle.

Au fil du projet, nous avons enrichi nos savoirs sur la technologie et l’industrie des ascenseurs. Toutefois, la complexité croissante de notre programme, en termes de taille et de nombre de fonctions, a limité notre capacité à ajouter certaines fonctionnalités avancées, telles que le système d’interruption ou la gestion des priorités des appels.

Dans une perspective d'amélioration et d'expansion de notre projet, nous proposons plusieurs évolutions pour la maquette :

- Intégrer des capteurs de sécurité en cas de dépassement de la course normale ;
- Automatiser l’ouverture et la fermeture des portes ;
- Interdire le déplacement de la cabine lorsque les portes sont ouvertes ;
- Ajouter des boutons pour la sélection d’étage a l’intérieur de la cabine (boutons d’envoi).

Ces améliorations permettront d'enrichir le projet et d’approcher un fonctionnement plus proche des ascenseurs réels**.**

ANNEXE 01

Datasheet

I2C 1602 Serial LCD Module

Product features:

The I2C 1602 LCD module is a 2 line by 16 character display interfaced to an I2C daughter board. The I2C interface only requires 2 data connections, +5 VDC and GND to operate

For in depth information on I2C interface and history, visit: http://www.wikipedia/wiki/i2c

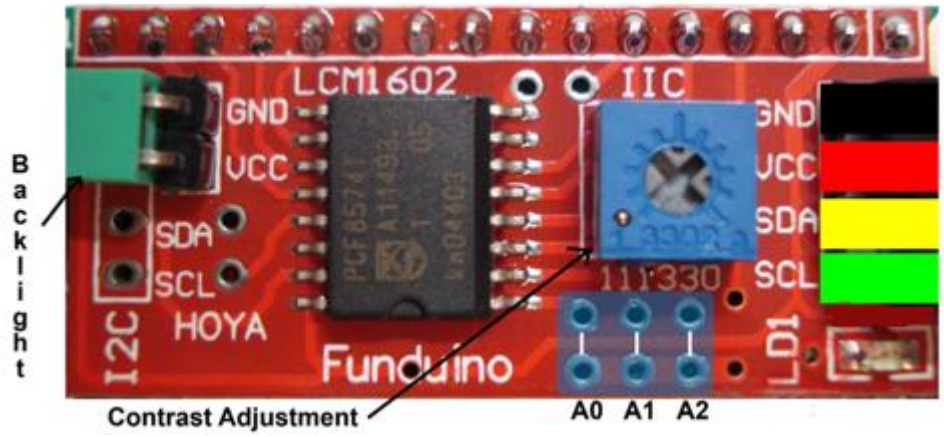
Specifications:

	2 lines by 16 character
I2C Address Range	0x20 to 0x27 (Default=0x27, addressable)
Operating Voltage	5 Vdc
Backlight	White
Contrast	Adjustable by potentiometer on I2c interface
Size	80mm x 36mm x 20 mm
Viewable area	66mm x 16mm

Power:

The device is powered by a single 5Vdc connection.

Pinout Diagram:



Pin/Control Descriptions:

Pin #	Name	Type	Description
1	GND	Power	Supply & Logic ground
2	VCC	Power	Digital I/O 0 or RX (serial receive)
3	SDA	I/O	Serial Data line
4	SCL	CLK	Serial Clock line
A0	A0	Jumper	Optional address selection A0 - see below
A1	A1	Jumper	Optional address selection A1 - see below
A2	A2	Jumper	Optional address selection A2 - see below
Backlight		Jumper	Jumpered - enable backlight, Open - disable backlight
Contrast		Pot	Adjust for best viewing

Addressing:

A0	A1	A2	Address
Open	Open	Open	0x27
Jumper	Open	Open	0x26
Open	Jumper	Open	0x25
Jumper	Jumper	Open	0x24
Open	Open	Jumper	0x23
Jumper	Open	Jumper	0x22
Open	Jumper	Jumper	0x21
Jumper	Jumper	Jumper	0x20

Software:

Download the required LCD Arduino™ library for this device from:

http://www.circuitattic.com/downloads/category/3-sample-code.html?download=9%3Aanother-i2c-library-easier-to-use

Replace current liquid crystal library found in the Arduino library directory with the above
(Note: If you use the examples included with the library, be sure to change address to 0x27)

Simple example using library above.

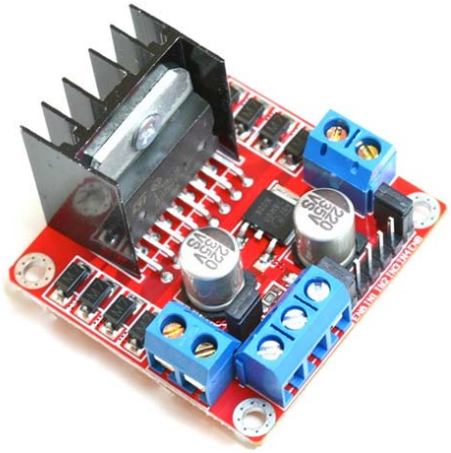
```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#if defined(ARDUINO) && ARDUINO >= 100
#define printByte(args) write(args);
#else
#define printByte(args) print(args,BYTE);
#endif
LiquidCrystal_I2C lcd(0x27,16,2); // set the LCD address to 0x27 for a
//chars and 2 line display
void setup()
{
      lcd.init(); // initialize the lcd
      lcd.backlight();
      lcd.clear();
      delay(100);
      for(int i = 0; i< 3; i++)
      {
            lcd.backlight();
            delay(250);
            lcd.noBacklight();
            delay(250);
      }
      lcd.backlight();


   }
   void loop()
   {
         int x=0;
         lcd.clear();
         lcd.setCursor(2,0); //Start at character 0 on line 0
         lcd.print("Hello World");
         lcd.setCursor(0,1); //Start at character 0 on line 1
         lcd.print(" opencircuit.nl");
         delay(3000); //Wait 3 seconds
         lcd.clear();
         lcd.setCursor(0,0); //Start at character 0 on line 0
         lcd.print("Cursor Blink");
         lcd.blink();
         delay(2000);
         lcd.setCursor(0,0);
         lcd.print("Cursor noBlink");
         lcd.noBlink();
         delay(2000);
   }
```

ANNEXE 02

User Guide

L298N Dual H-Bridge Motor Driver

This dual bidirectional motor driver, is based on the very popular L298 Dual H-Bridge Motor Driver Integrated Circuit. The circuit will allow you to easily and independently control two motors of up to 2A each in both directions.It is ideal for robotic applications and well suited for connection to a microcontroller requiring just a couple of control lines per motor. It can also be interfaced with simple manual switches, TTL logic gates, relays, etc. This board equipped with power LED indicators, on-board +5V regulator and protection diodes.

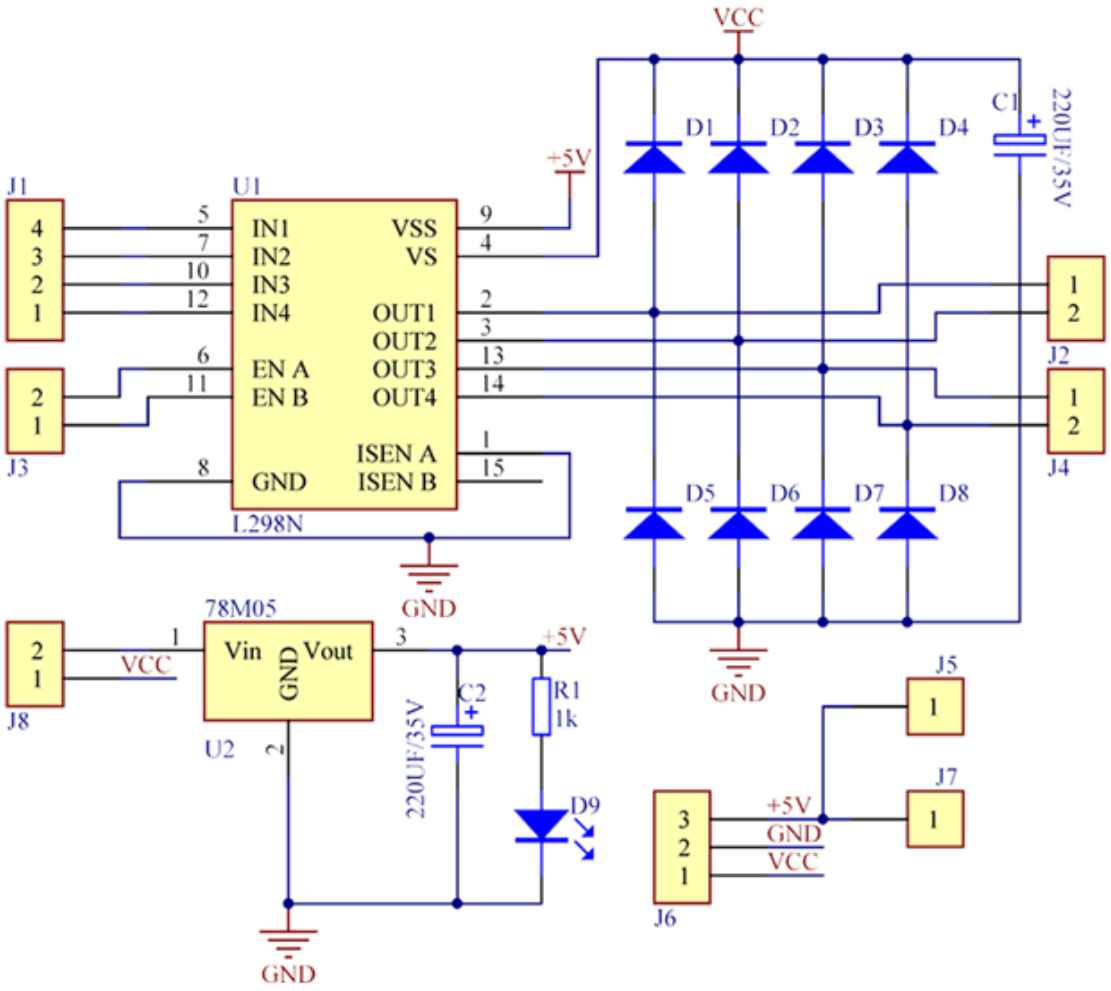


SKU: MDU-1049

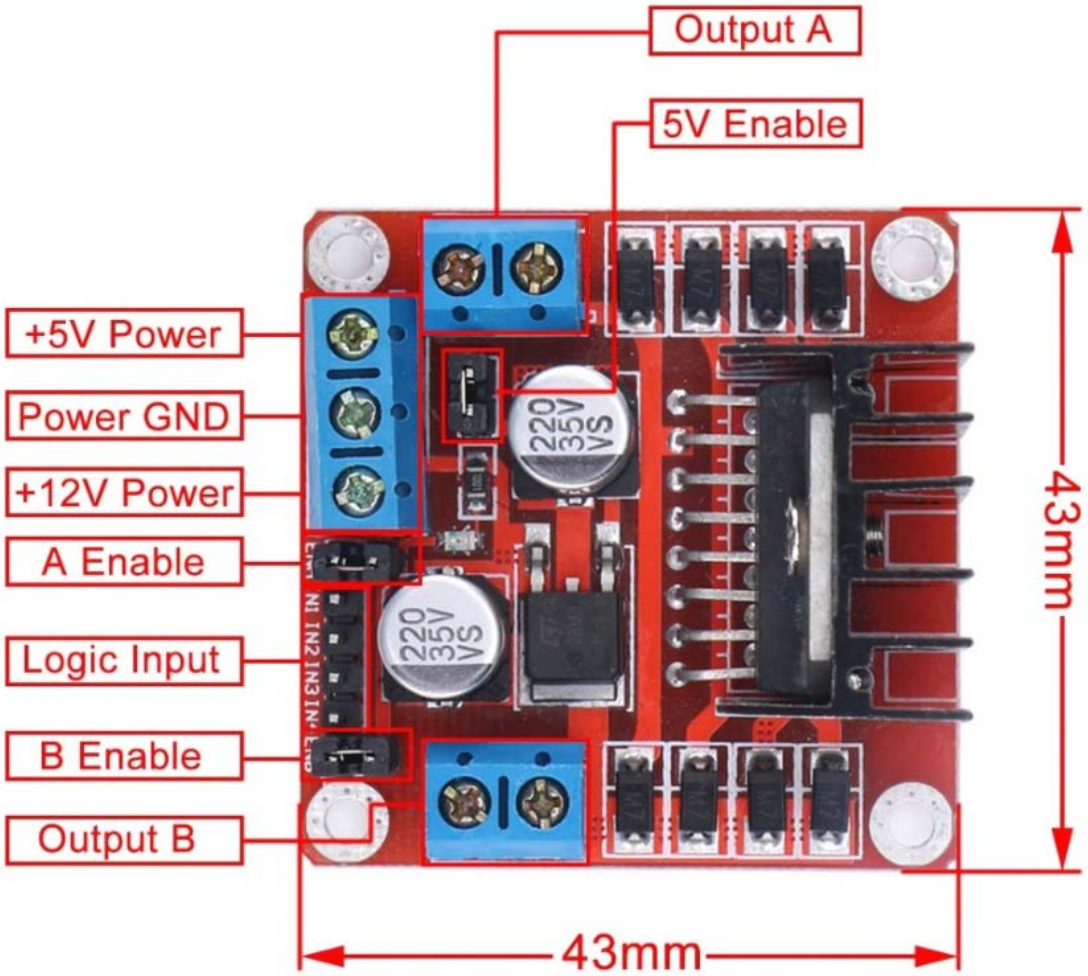
Brief Data:

- Input Voltage: 3.2V~40Vdc.
- Driver: L298N Dual H Bridge DC Motor Driver
- Power Supply: DC 5 V - 35 V
- Peak current: 2 Amp
- Operating current range: 0 ~ 36mA
- Control signal input voltage range :
- Low: -0.3V ⩽ Vin ⩽ 1.5V.
- High: 2.3V ⩽ Vin ⩽ Vss.
- Enable signal input voltage range :
    - Low: -0.3 ⩽ Vin ⩽ 1.5V (control signal is invalid).
    - High: 2.3V ⩽ Vin ⩽ Vss (control signal active).
- Maximum power consumption: 20W (when the temperature T = 75 ℃).
- Storage temperature: -25 ℃ ~ +130 ℃.
- On-board +5V regulated Output supply (supply to controller board i.e. Arduino).
- Size: 3.4cm x 4.3cm x 2.7cm

Schematic Diagram:



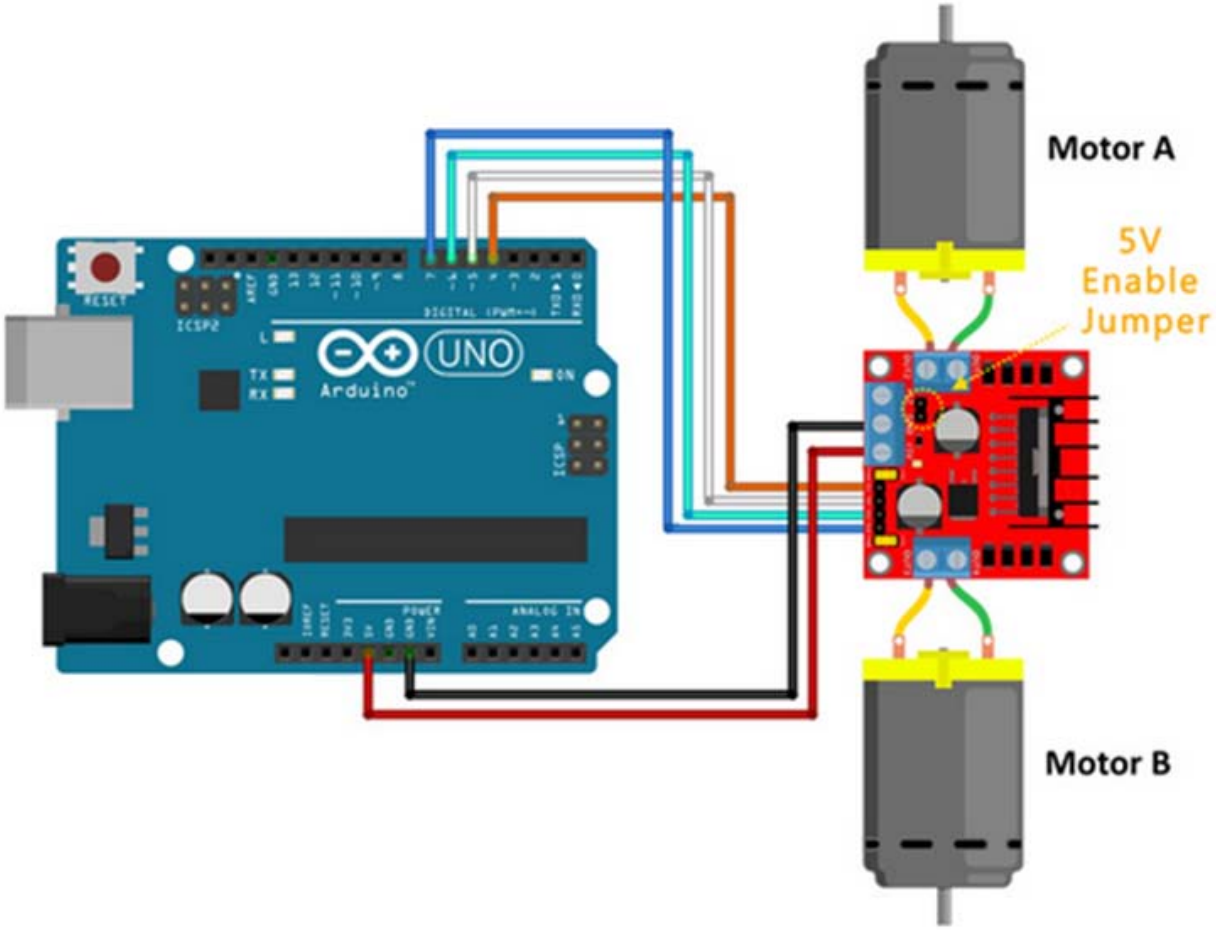
Board Dimension & Pins Function:



Connection Examples:

Controlling 2-DC Motor with +5V Arduino onboard Power Supply:

Below is the circuit connection use the on-board +5V power supply from Arduino board, and should be done without the 5V Enable Jumper on (Active 5V). This connection can drive two 5V DC motors simultaneously.



Sketch Listing:

Copy and paste the sketch below to Arduino IDE and upload to Arduino Uno/Mega board.

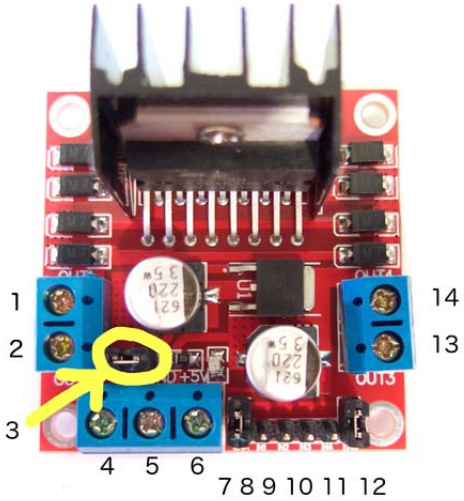
```
/*=====================================================================
//  Author      : Handson Technology
//  Project     : Arduino Uno
//  Description : L298N Motor Driver
//  Source-Code : L298N_Motor.ino
//  Program: Control 2 DC motors using L298N H Bridge Driver
//=====================================================================
*/

// Definitions Arduino pins connected to input H Bridge
int IN1 = 4;
int IN2 = 5;
int IN3 = 6;
int IN4 = 7;

void setup()
{
 // Set the output pins
```

```
 pinMode(IN1, OUTPUT);
 pinMode(IN2, OUTPUT);
 pinMode(IN3, OUTPUT);
 pinMode(IN4, OUTPUT);
}

void loop()
{
 // Rotate the Motor A clockwise
 digitalWrite(IN1, HIGH);
 digitalWrite(IN2, LOW);
 delay(2000);
 // Motor A
 digitalWrite(IN1, HIGH);
 digitalWrite(IN2, HIGH);
 delay(500);

 // Rotate the Motor B clockwise
 digitalWrite(IN3, HIGH);
 digitalWrite(IN4, LOW);
 delay(2000);
 // Motor B
 digitalWrite(IN3, HIGH);
 digitalWrite(IN4, HIGH);
 delay(500);

 // Rotates the Motor A counter-clockwise
 digitalWrite(IN1, LOW);
 digitalWrite(IN2, HIGH);
 delay(2000);
 // Motor A
 digitalWrite(IN1, HIGH);
 digitalWrite(IN2, HIGH);
 delay(500);

 // Rotates the Motor B counter-clockwise
 digitalWrite(IN3, LOW);
 digitalWrite(IN4, HIGH);
 delay(2000);
 // Motor B
 digitalWrite(IN3, HIGH);
 digitalWrite(IN4, HIGH);
 delay(500);
}
```

Controlling Stepper Motor

In this example we have a typical NEMA-17 stepper motor with four wires:

The key to successful stepper motor control is identifying the wires - that is which one is which. You will need to determine the A+, A-, B+ and B- wires. With our example motor these are red, green, yellow and blue. Now let's get the wiring done.



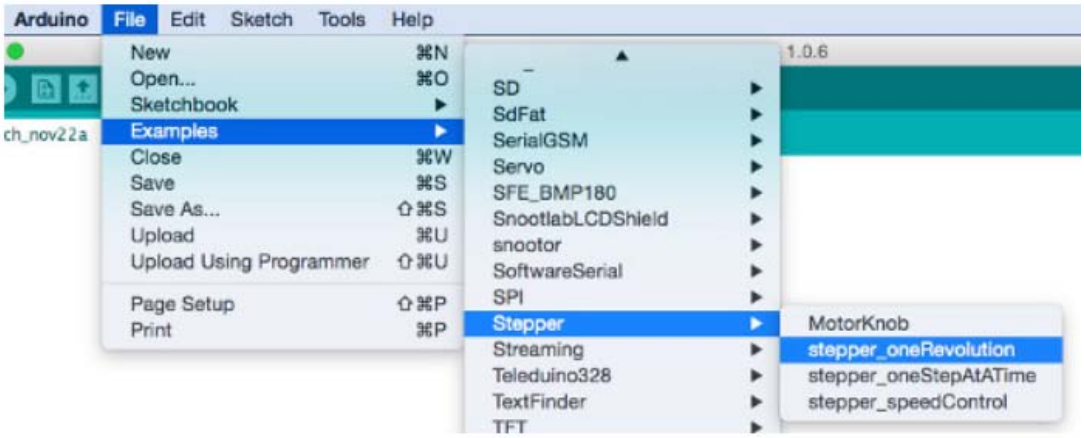
Connect the A+, A-, B+ and B- wires from the stepper motor to the module connections 1, 2, 13 and 14 respectively. Place the jumpers included with the L298N module over the pairs at module points 7 and 12. Then connect the power supply as required to points 4 (positive) and 5 (negative/GND).

Once again if your stepper motor's power supply is less than 12V, fit the jumper to the module at point 3 which gives you a neat 5V power supply for your Arduino.

Next, connect L298N module pins IN1, IN2, IN3 and IN4 to Arduino digital pins D8, D9, D10 and D11 respectively. Finally, connect Arduino GND to point 5 on the module, and Arduino 5V to point 6 if sourcing 5V from the module.

Controlling the stepper motor from your sketches is very simple, thanks to the *Stepper* Arduino library included with the Arduino IDE as standard.

To demonstrate your motor, simply load the *"stepper_oneRevolution"* sketch that is included with the *Stepper* library, for example:



Finally, check the value for

```
const int stepsPerRevolution = 200;
```

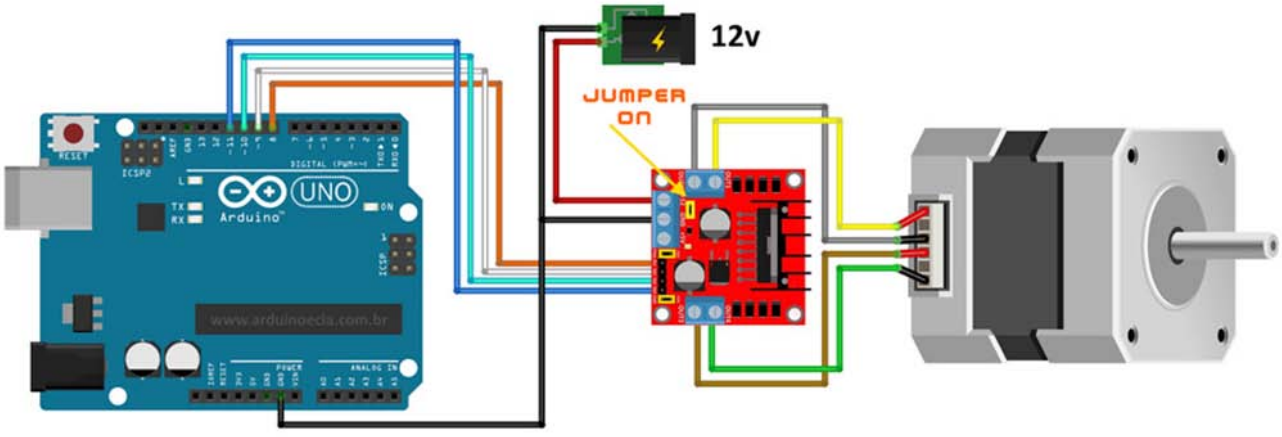
in the sketch and change the 200 to the number of steps per revolution for your stepper motor, and also the speed which is preset to 60 RPM in the following line:

```
myStepper.setSpeed(60);
```

Now you can save and upload the sketch, which will send your stepper motor around one revolution, then back again. This is achieved with the function

```
myStepper.step(stepsPerRevolution); // for clockwise
myStepper.step(-stepsPerRevolution); // for anti-clockwise
```

Connection for the sketch "*stepper_oneRevolution*":



Web Resources:

BIBLIOGRAPHIE

[1]: Erik Bartmann, Danielle Lafarge, Patrick Chantereau - Le grand livre d'Arduino-Eyrolles (2015).

[2]: Jean-Christophe Quetin - Arduino Apprivoisez l'électronique et le codage-éditions ENI (2018)

[3]: AKKOUCHE Fani « Réalisation d'un ascenseur à base d'Arduino » mémoire fin d'étude, université Akli mohend oulhadj-Bouira spécialité Electronique des systèmes embarqués en 2021

[4]: https://www.arduino.cc .

[5]: https://fr.wikipedia.org/wiki/Ascenseur

[6]: https://energieplus-lesite.be,

[7]: EONE-1602.PDF

[8] : Becky Stewart, Jean Boyer - A l'aventure avec Arduino-Eyrolles (2015)

[9]: Aide-mémoire - Électronique (Bogdan Grabowski, Christian Ripoll.

[10] : https://passionelectronique.fr/tutoriel-l298n/.

[11] : Afficheur LCD : https://www.gotronic.fr/pj2-sbc-lcd16x2-fr-1441.pdf

[12] : https://e-techno-tutos.com/

[13] : Boutons poussoir : https://www.locoduino.or