République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU

FACULTE DE GENIE ELECTRIQUE ET DE L'INFORMATIQUE

DEPARTEMENT D'ELECTRONIQUE

Mémoire de fin d'étude de Master Professionnel

Spécialité : **Electronique industrielle**
Filière : **Génie Electrique**

Présenté par :
BENASSEL Hamza
BOUKESIL Rachid

Thème

Conception et réalisation d'un tapis roulant industrielle automatisé

Soutenu publiquement le 10/07/2018 devant les jurys :

Mr. OUALOUCHE	**MCB-UMMTO**	**President**
Mr. LAZRI	**MCA-UMMTO**	**Promoteur**
Mr. HAMEG	**MAA-UMMTO**	**Examinateur**

Remerciements

En préambule à ce mémoire
Nous remerciant le dieu qui nous aide et nous donne la patience et le courage durant
ces langues années d'étude.
Nous souhaitons adresser nos remerciements les plus sincères aux personnes qui nous
ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire ainsi qu'à la
réussite de cette formidable année universitaire.
***Nous tenant à remercier sincèrement Mr* LAZRI,**
en tant qu'Encadreur,
Aux membres de jury
Enfin, nous adressons nos plus sincères remerciements à tous nos proches et amis,
qui nous ont toujours soutenue et encouragée
au cours de la réalisation de ce mémoire.
Merci à tous

Dédicace

Je Tiens tout d'abord à remercier le bon Dieu de m'avoir aidé à réaliser ce mémoire

Que je dédie à :

A la prunelle de mes yeux, celle qui m'a soutenu surtout avec son Dou'a jour et nuit pour qu'elle me voit au sommet et comme une étoile, à la source de la Douceur.

A toi ma très chère mère.

A la personne qui a sacrifié sa vie pour moi, et qui a pris le défi pour mes études

Et mas éclairé le chemin de ma réussite.

A toi mon très cher Père.

A la mémoire de mes grands-pères.

A mon cher frère et ma chère sœur Qui sont toujours à mes côtés et qui m'ont soutenu durant mon parcours universitaire

A tous mes ami(e)s

A mon binôme et sa famille

BENASSEL Hamza

Dédicace

Ma Mère, Mon Père
Affable, honorable, aimable : vous représentes pour moi
Le symbole de la bonté par excellence, la source de tendresse
et l'exemple du
dévouement qui n 'a pas cessé de m 'encourager et de prier
pour moi.
Soyez sûrs que je continuerai mon chemin.
Je vous dédie ce travail en témoignage de mon profond amour.
Puisse Dieu, le tout puissant, vous préserver et vous accorder santé,
longue vie et bonheur.
A mes sœurs
A mes frères
En témoignage de l'attachement, de l'amour et de
L 'affection que je porte pour vous.
A tous les membres de ma famille, petits et grands
A tous mes amis de proches.

BOUKSIL Rachid

Table des matières

Chapitre II
Les différents capteurs et moteurs

Chapitre III
La réalisation pratique

Liste des tableaux

L’avènement de l’électronique embarquée a permis de simplifier considérablement plusieurs fonctions de l’électronique dans différents domaines dont celui de l’industrie.
En effet, on n’a pas besoin, dans certains cas, d’automates programmables industriels pour automatiser une chaine de production alors qu’il existe une solution peu onéreuse et relativement simple à mettre en œuvre.

Vu sous cet angle, nous avons choisit une fonction très utilisée dans le monde de l’industrie qui est celle du convoie de pièces dans une chaines de production ou de montage.
Cette fonction est réalisée par un tapis roulant qui est généralement commandé par un API.
Notre travail consiste à substituer L’API par une carte Arduino à la quelle on a rajouté des capteurs.

Pour se faire nous avons devisé notre travail en trois chapitres :

- Le premier chapitre décrit la carte arduino d’un point de vu structurel et fonctionnel.
- Le deuxième chapitre, quant à lui, est consacré aux généralités sur le moteur à courant continu et les détecteurs d’obstacle de type infrarouge.
- Le troisièmes chapitre est dédié à la conception et la réalisation de tapis roulant a base d’une carte arduino uno.

CHAPITRE I

GÉNERALITÉS SUR LA CARTE ARDUINO UNO

I.1 Préambule

Aujourd'hui, on fait appel à l'électronique embarqué pour réaliser différentes taches. L'avantage de l'utilisation de ce type de l'électronique réside dans la simplification de l'aspect matériel (Hardware), permettant ainsi de réduire considérablement les couts de fabrication et de maintenance du produit.

I.2 Définition du module Arduino

Arduino est une plate-forme open-source utilisée pour les projets de construction électronique. Arduino se compose à la fois d'une carte de circuit programmable physique (souvent appelée microcontrôleur) et d'un logiciel, ou IDE (Integrated Development Environment) qui fonctionne sur notre ordinateur, utilisé pour écrire et télécharger du code informatique sur la carte physique. [1]

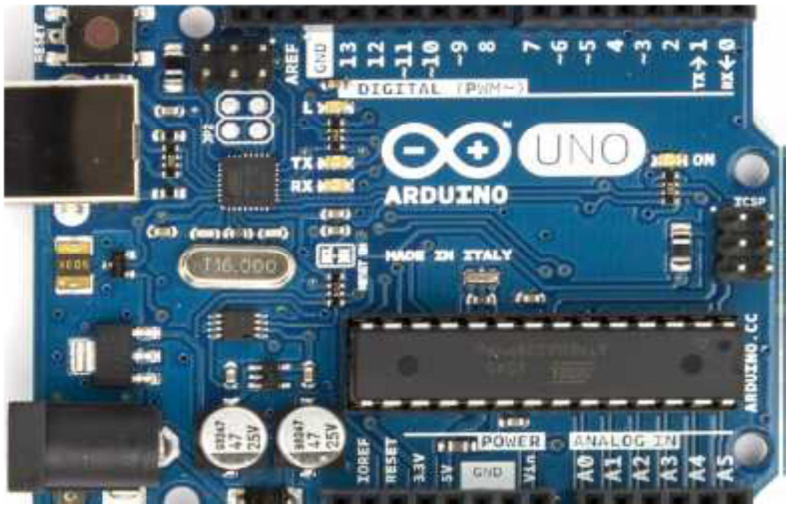


Figure I.1 La carte Arduino UNO

I.3 Pourquoi Arduino UNO

Il y a de nombreuses cartes électroniques qui possèdent des plateformes basées sur des Microcontrôleurs disponibles pour l'électronique programmée. Tous ces outils prennent en charge les détails compliqués de la programmation et les intègrent dans une présentation facile à utiliser. De la même façon, le système Arduino simplifie la façon de travailler avec les microcontrôleurs tout en offrant à personnes intéressées plusieurs avantages cités comme suit

- **Le prix (réduits)** : les cartes Arduino sont relativement peu coûteuses

Comparativement aux autres plates-formes. La moins chère des versions du module Arduino peut être assemblée à la main, (les cartes Arduino pré-assemblées coûtent moins de 2500 Dinars).

- **Multi plateforme** : le logiciel Arduino, écrit en JAVA, tourne sous les systèmes

D'exploitation Windows, Macintosh et Linux. La plupart des systèmes à microcontrôleurs sont limités à Windows.

- **Un environnement de programmation clair et simple**: l'environnement de programmation Arduino (le logiciel Arduino IDE) est facile à utiliser pour les débutants, tout en étant assez flexible pour que les utilisateurs avancés puissent en tirer profit également.
- **Logiciel Open Source et extensible :** le logiciel Arduino et le langage Arduino sont publiés sous licence open source, disponible pour être complété par des programmateurs expérimentés. Le logiciel de programmation des modules Arduino est une application JAVA multi plateformes (fonctionnant sur tout système d'exploitation), servant d'éditeur de code et de compilateur, et qui peut transférer le programme au travers de la liaison série (RS232, Bluetooth ou USB selon le module).
- **Matériel Open source et extensible** : les cartes Arduino sont basées sur les Microcontrôleurs Atmel ATMEGA8, ATMEGA168, ATMEGA 328, les schémas des modules sont publiés sous une licence créative Commons, et les concepteurs des circuits expérimentés peuvent réaliser leur propre version des cartes Arduino, en les complétant et en les améliorant. Même les utilisateurs relativement inexpérimentés peuvent fabriquer la version sur plaque d'essai de la carte Arduino, dont le but est de comprendre comment elle fonctionne pour économiser le coût.

I. 4 La constitution de la carte Arduino UNO

Un module Arduino est généralement construit autour d'un microcontrôleur ATMEL AVR, et de composants complémentaires qui facilitent la programmation et l'interfaçage avec D'autres circuits. Chaque module possède au moins un régulateur linéaire 5V et un oscillateur à quartz 16 MHz (ou un résonateur céramique dans certains modèles). Le microcontrôleur est préprogrammé avec un boot loader de façon à ce qu'un programmateur dédié ne soit pas Nécessaire.

I.4.1 Partie matérielle

Généralement tout module électronique qui possède une interface de programmation est basé toujours dans sa construction sur un circuit programmable ou plus.

I.4.1.1 Le Microcontrôleur ATMega328

Voilà le cerveau de notre carte C'est lui qui va recevoir le programmée qui va les stocker dans sa mémoire puis l'exécuter. Grâce à ce programme, il va savoir faire des choses, qui peuvent être : faire clignoter une LED, afficher des caractères sur un écran, envoyer des données à un ordinateur.

Un microcontrôleur ATMega328 est un circuit intégré qui rassemble sur une puce plusieurs éléments complexes dans un espace réduit au temps des pionniers de l'électronique. Aujourd'hui, en soudant un grand nombre de composants encombrants ; tels que les transistors; les résistances et les condensateurs tout peut être logé dans un petit boîtier en plastique noir muni d'un certain nombre de broches dont la programmation peut être réalisée en langage C. la figure I.2 montre un microcontrôleur ATmega 328, qu'on trouve sur la carte Arduino.[3]

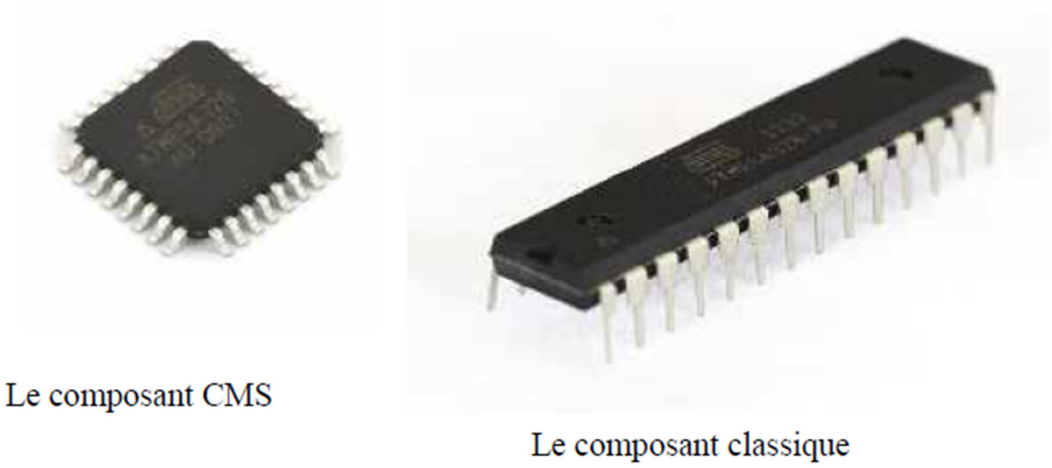


Figure I.2 Microcontrôleur ATMega328

Le microcontrôleur ATMega328 est constitué par un ensemble d'éléments qui ont chacun une fonction bien déterminée. Il est en fait constitué des mêmes éléments que sur la carte mère d'un ordinateur. Globalement, l'architecture interne de ce circuit programmable se compose essentiellement sur :

- **La mémoire Flash**: C'est celle qui contiendra le programme à exécuter. Cette

mémoire est effaçable et réinscriptible mémoire programme de 32Ko (dont boot loader de 0.5 ko).

- **RAM :** c'est la mémoire dite "vive", elle va contenir les variables du programme. Elle est dite "volatile" car elle s'efface si on coupe l'alimentation du microcontrôleur. Sa Capacité est 2 ko.
- **EEPROM**: C'est le disque dur du microcontrôleur. On y enregistre des infos qui ont besoin de survivre dans le temps, même si la carte doit être arrêtée. Cette mémoire ne s'efface pas lorsque l'on éteint le microcontrôleur ou lorsqu'on le reprogramme. [4]

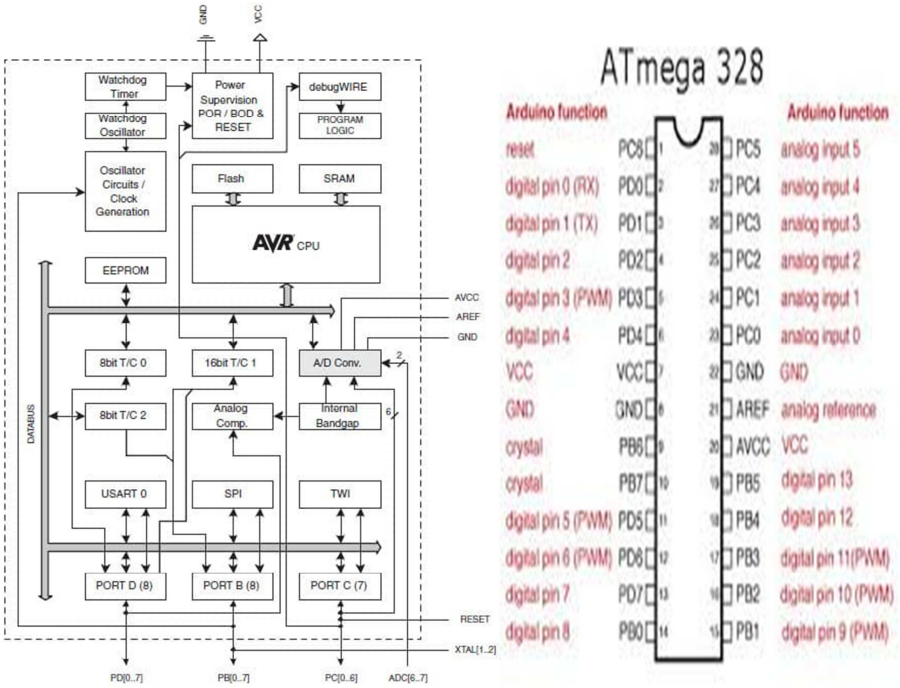


Figure I.3 Architecture du microcontrôleur ATMEGA328

I.4.1.2 Les sources de l'alimentation de la carte

On peut distinguer deux genres de sources d'alimentation (Entrée Sortie) et cela comme suit :

- **VIN** : La tension d'entrée positive lorsque la carte Arduino est utilisée avec une source de tension externe (à distinguer du 5V de la connexion USB ou autre source 5 régulée). On peut alimenter la carte à l'aide de cette broche, ou, si l'alimentation est fournie par le jack d'alimentation, accéder à la tension d'alimentation sur cette broche.
- **5V** : La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (pour info : les circuits électroniques numériques nécessitent

une tension d'alimentation parfaitement stable dite "tension régulée" obtenue à l'aide d'un composant appelé un régulateur et qui est intégré à la carte Arduino). Le 5V régulé fourni par cette broche peut donc provenir soit de la tension d'alimentation VIN via le régulateur de la carte, ou bien de la connexion USB (qui fournit du 5V régulé) ou de tout autre source d'alimentation régulée.

- **3V3**. Une alimentation de 3.3V fournie par le circuit intégré FTDI (circuit intégré faisant l'adaptation du signal entre le port USB de votre ordinateur et le port série de l'ATmega) de la carte est disponible : ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5V. L'intensité maximale disponible sur cette broche est de 50mA. [2]

I.4.1.3 Les entrées & sorties

Cette carte possède 14 broches numériques (numérotée de 0 à 13) peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions pinMode( ), digitalWrite( ) et digital Read( ) du langage Arduino. Ces broches fonctionnent en 5V. Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité et dispose d'une résistance interne de "rappel au plus" (pull-up) (déconnectée par défaut) de 20-50 KOhms. Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction digitalWrite (broche, HIGH).

En plus, certaines broches ont des fonctions spécialisées :

- **Interruptions Externes**: Broches 2 et 3. Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur. -Impulsion PWM (largeur d'impulsion modulée): Broches 3, 5, 6, 9, 10, et 11. Fournissent une impulsion PWM 8-bits à l'aide de l'instruction analogWrite ( ).
- **SPI** (Interface Série Périphérique): Broches 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ces broches supportent la communication SPI (Interface Série Périphérique) disponible avec la librairie pour communication SPI. Les broches SPI sont également connectées sur le connecteur ICSP qui est mécaniquement compatible avec les cartes Méga.
- **I2C**: Broches 4 (SDA) et 5 (SCL). Supportent les communications de protocole I2C (ou interface TWI (TwoWire Interface - Interface "2 fils"), disponible en utilisant la librairieWire/I2C (ou TWI - Two-Wire interface - interface "2 fils").
- **LED**: Broche 13. Il y a une LED incluse dans la carte connectée à la broche 13. Lorsque la broche est au niveau HAUT, la LED est allumée, lorsque la broche est au

niveau BAS, la LED est éteinte.

La carte UNO dispose 6 entrées analogiques (numérotées de 0 à 5), chacune pouvant fournir une mesure d'une résolution de 10 bits (c.à.d. sur 1024 niveaux soit de 0 à 1023) à l'aide de la très utile fonction analog Read( ) du langage Arduino. Par défaut, ces broches mesurent entre le 0V (valeur 0) et le 5V (valeur 1023), mais il est possible de modifier la référence supérieure de la plage de mesure en utilisant la broche AREF et l'instruction analog Reference( ) du langage Arduino.

La carte Arduino UNO intègre un fusible qui protège le port USB de l'ordinateur contre les surcharges en intensité (le port USB est généralement limité à 500mA en intensité).Bien que la plupart des ordinateurs aient leur propre protection interne, le fusible de la carte fournit une couche supplémentaire de protection. Si plus de 500mA sont appliqués au port USB, le fusible de la carte coupera automatiquement la connexion jusqu'à ce que le court-circuit ou la surcharge soit stoppé. [1]

I.4.1.4 Les ports de communications

La carte Arduino UNO a de nombreuses possibilités de communications avec l'extérieur. L'Atmega328 possède une communication série UART TTL (5V), grâce aux broches numériques 0 (RX) et 1 (TX).

On utilise (RX) pour recevoir et (TX) transmettre (les données séries de niveau TTL). Ces broches sont connectées aux broches correspondantes du circuit intégré ATmega328 programmé en convertisseur USB – vers – série de la carte, composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur.

Comme un port de communication virtuel pour le logiciel sur l'ordinateur, La connexion série de l'Arduino est très pratique pour communiquer avec un PC, mais son inconvénient est le câble USB, pour éviter cela, il existe différentes méthodes pour utiliser ce dernier sans fil.

I.4.2 Partie programme

Une telle carte d'acquisition qui se base sur sa construction sur un microcontrôleur doit être dotée d'une interface de programmation comme est le cas de notre carte.

L'environnement de programmation open-source pour Arduino peut être téléchargé gratuitement (pour Mac OS X, Windows, et Linux).

I.4.2.1 Environnement de développement IDE

Un environnement de développement est un ensemble d'outils qui permet d'augmenter la productivité des programmeurs qui développent des logiciels. Il comporte un éditeur de texte destiné à la programmation, des fonctions qui permettent, par pression sur un bouton, de démarrer le compilateur ou l'éditeur de liens ainsi qu'un débogueur en ligne, qui permet d'exécuter ligne par ligne le programme en cours de construction. Certains environnements sont dédiés à un langage de programmation en particulier.

Dans un environnement de développement « intégré » (abrégé **EDI** en français ou **IDE** en anglais, pour ***integrated development environment***), les outils sont prévus pour être utilisés ensemble (le produit d'un outil peut servir de matière première pour un autre outil). Les outils peuvent être intégrés dès le départ, c'est-à-dire qu'ils sont construits dans le but d'être utilisés ensemble. Il peut aussi s'agir d'un ensemble d'outils développés sans lien entre eux et intégrés a posteriori

L'objectif d'un environnement de développement est d'augmenter la productivité des programmeurs en automatisant une partie des activités et en simplifiant les opérations. Les environnements de développement visent également à améliorer la qualité de la documentation en rapport avec le logiciel en construction. Certains environnements de développement offrent également la possibilité de créer des prototypes, de planifier les travaux et de gérer des projets.

Depuis 1980, le développement et la maintenance de logiciels sont partiellement automatisés à l'aide d'un lot d'outils l'environnement de développement. Les outils des environnements de développement sont un sujet d'étude en génie logiciel le savoir-faire en ingénierie du logiciel.[5].

I.4.2.2 Structure générale du programme (IDE Arduino)

Comme n’importe quel langage de programmation, une interface simple et exécutable sur n’importe quel système d’exploitation Arduino basé sur la programmation en C.

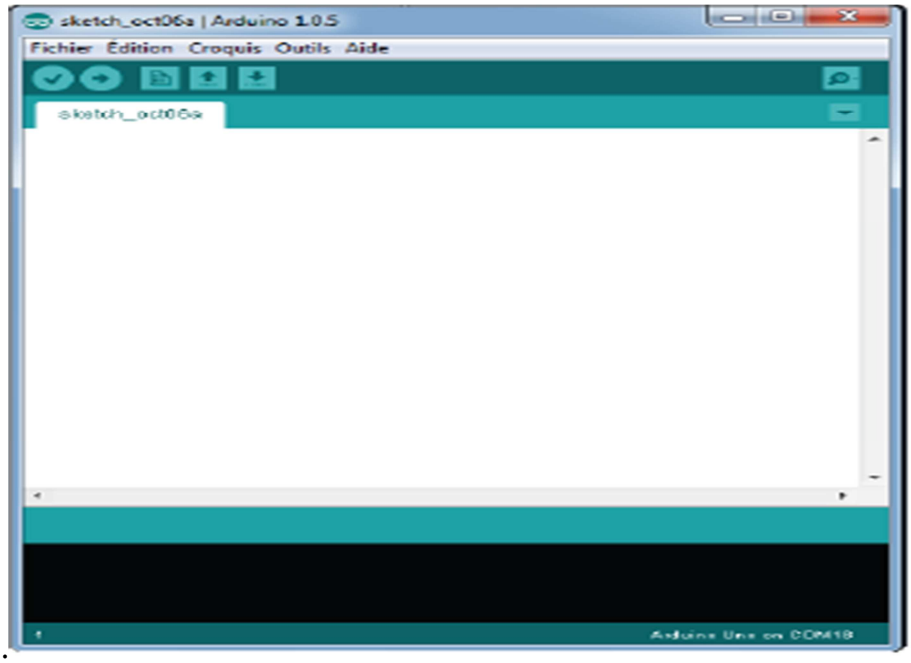


Figure I.5 Interface de l'IDE Arduino

Ce qui saute aux yeux en premier, c'est la clarté de présentation du logiciel. On voit tout de suite son interface intuitive. Voyons comment se compose cette interface.

I .5 Présentation du logiciel

On a découpé l'image précédente en plusieurs parties :

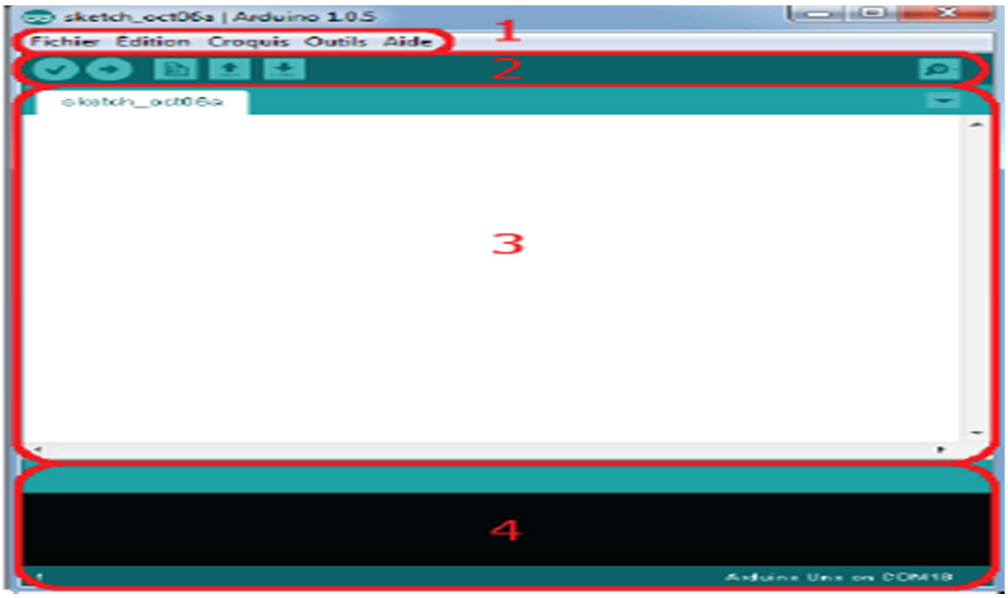


Figure I.5 Les différentes parties de l'interface IDE Arduino

I .5.1 Correspondance

Le cadre numéro 1: ce sont les options de configuration du logiciel

Le cadre numéro 2 : il contient les boutons qui vont nous servir lorsque l'on va Programmer nos cartes

Le cadre numéro 3 : ce bloc va contenir le programme que nous allons créer

Le cadre numéro 4 : celui-ci est important, car il va nous aider à corriger les fautes dans notre programme. C'est le débogueur.

I.5.2 Le menu file

C'est principalement ce menu que l'on va utiliser le plus. Il dispose d'un certain nombre de choses qui vont nous être très utiles. Il a été traduit en français progressivement, nous allons donc voir les quelques options qui sortent de l'ordinaire [7]

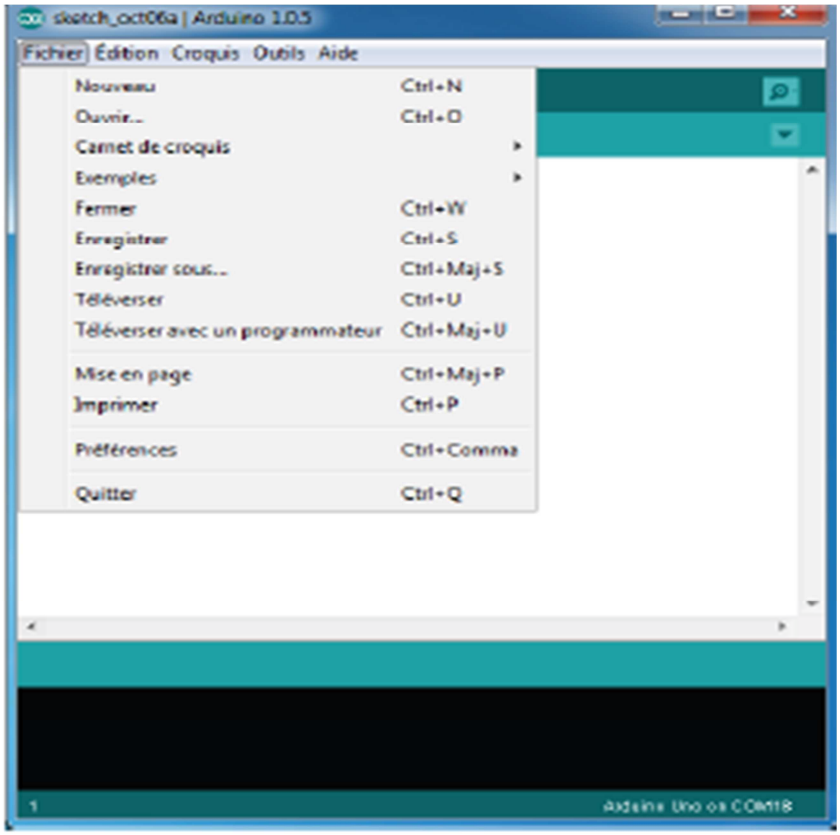


Figure I.6 Le principale menu du IDE arduino

Carnet de croquis : Ce menu regroupe les fichiers que nous avons pu faire jusqu'à maintenant (et s'ils sont enregistré dans le dossier par défaut du logiciel)

Exemples (exemples) : ceci est important, toute une liste se déroule pour afficher les noms d'exemples de programmes existants qui peut nous aider/inspirer pour créer nous propres programmes ou tester de nouveaux composants

Téléverser: Permet d'envoyer le programme sur la carte Arduino.

Préférences : on peut régler ici quelques paramètres du logiciel le reste des menus n'est pas intéressant pour l'instant, on y reviendra plus tard.

I.5.3Les boutons

Voyons à présent à quoi servent les boutons, encadrés en rouge et numérotés par le chiffre 2.

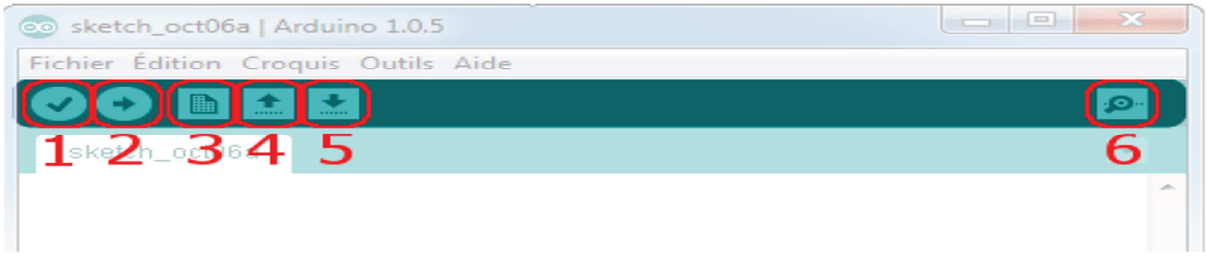


Figure I.7 les boutons de commande

Bouton 1 : Ce bouton permet de vérifier le programme, il actionne un module qui cherche les erreurs dans votre programme permet de vitrifie la syntaxes de programme une barre horizontale s'affiche au début de la vitrification

si aucune erreur n'est constatée ; l'opération se termine par le message done compi-ling .dans la fenêtre d'Edition se trouve une indication relative aux besoin en mémoire du sketch.

Bouton 2 : Charge (télé verse) le programme dans la carte Arduino

pendant le téléchargement du sketch ; sur la carte se trouve des petites diodes qui tiennent au courant de certaines activités

Bouton 3 : Crée un nouveau fichier

Bouton 4 : Ouvre un fichier cette icone sert a charge un sketchs enregistre sur le disque dur dans l'ide .il permet aussi d'accéder aux nombreux exemples de sketches

Bouton 5 : Enregistre le fichier sur un support de donne l'enregistrement s'effectue par défaut

Bouton 6 : Ouvre le moniteur série

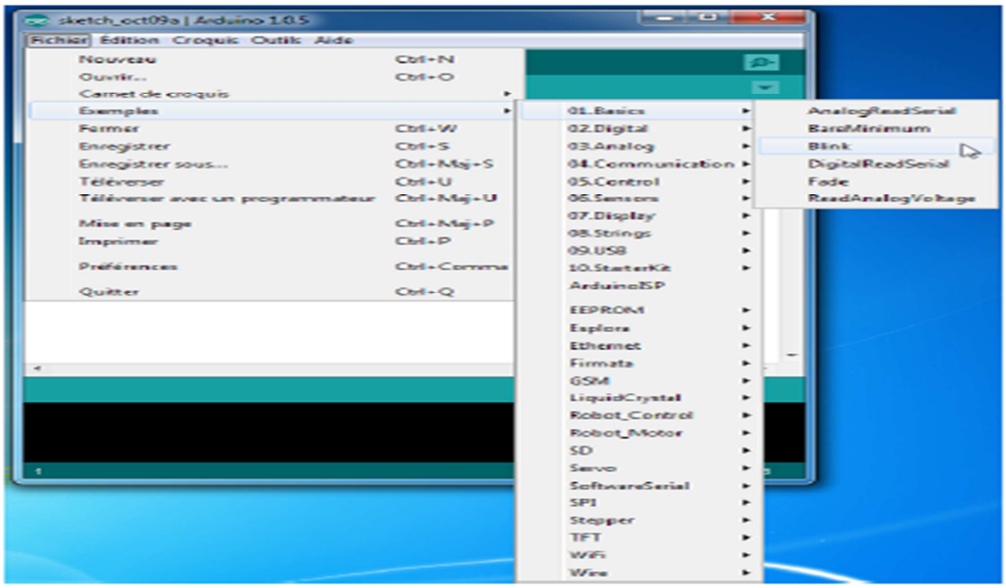
I .6Tester son matériel

Avant de commencer à programmer la tête baissée, il faut, avant toutes choses, tester le bon fonctionnement de la carte. Car ce serait idiot de programmer la carte et chercher les

erreurs dans le programme alors que le problème vient de la carte ! Nous allons tester notre matériel en chargeant un programme qui fonctionne dans la carte Mais, on n'en a pas encore fait de programmes tout juste Mais le logiciel Arduino contient des exemples de programmes. Et bien ce sont ces exemples que nous allons utiliser pour tester la carte.

I. 6.1 ouvrir un programme

Nous allons choisir un exemple tout simple qui consiste à faire clignoter une LED son nom est Blinket qui se trouve dans la catégorie *Basics* :



FigureI.8 Ouvrir un exemple du programme

Une fois on a cliqué sur Blink, une nouvelle fenêtre va apparaître. Elle va contenir le programme Blink. On peut fermer l'ancienne fenêtre qui va ne nous servir plus à rien.

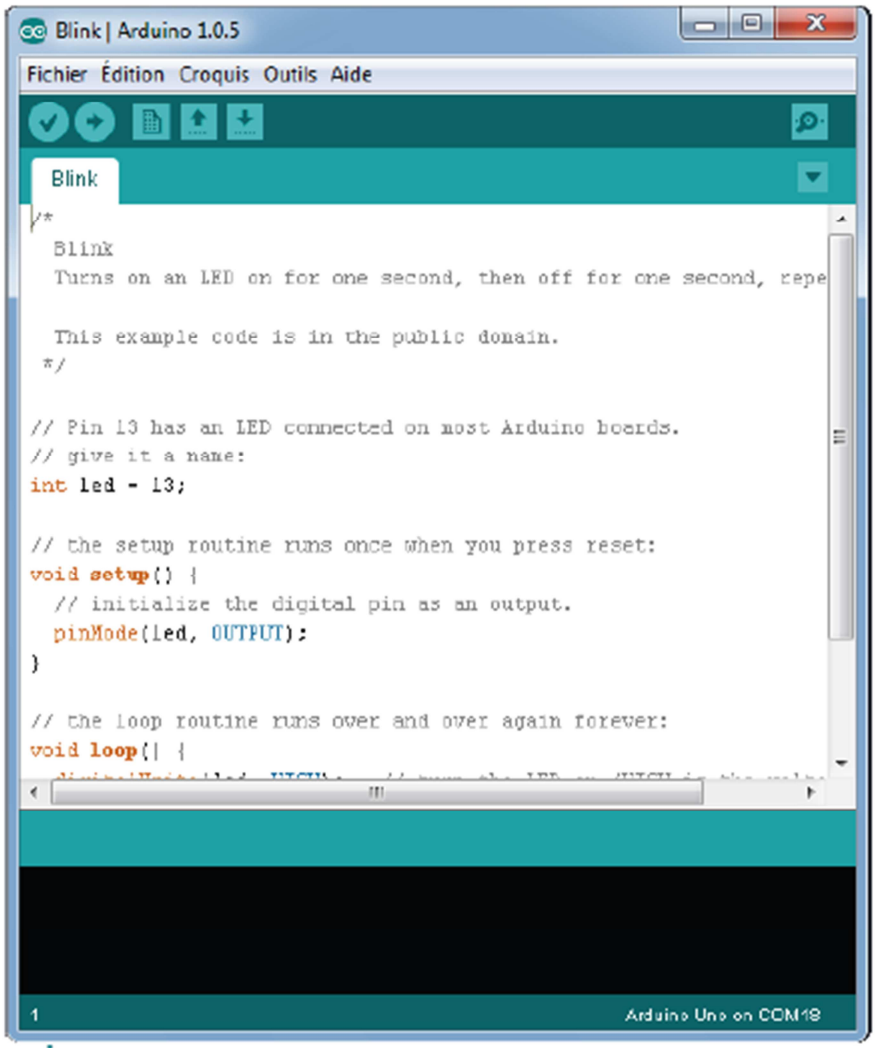


Figure I.9 Exemple de programme

I .6.2 Le choix de la carte

Avant d'envoyer le programme Blinkvers la carte, il faut dire au logiciel quel est le nom de la carte et sur quel port elle est branchée. Choisir la carte que l'on va programmer.

Ce n'est pas très compliqué, le nom de notre carte est indiqué sur elle. Pour nous, il s 'agit de la carte "Uno". On va dans le menu "*Tools*" ("outils" en français) puis dans "*Board*" ("carte" en français). On vérifie que c'est bien le nom "ArduinUno" qui est coché.

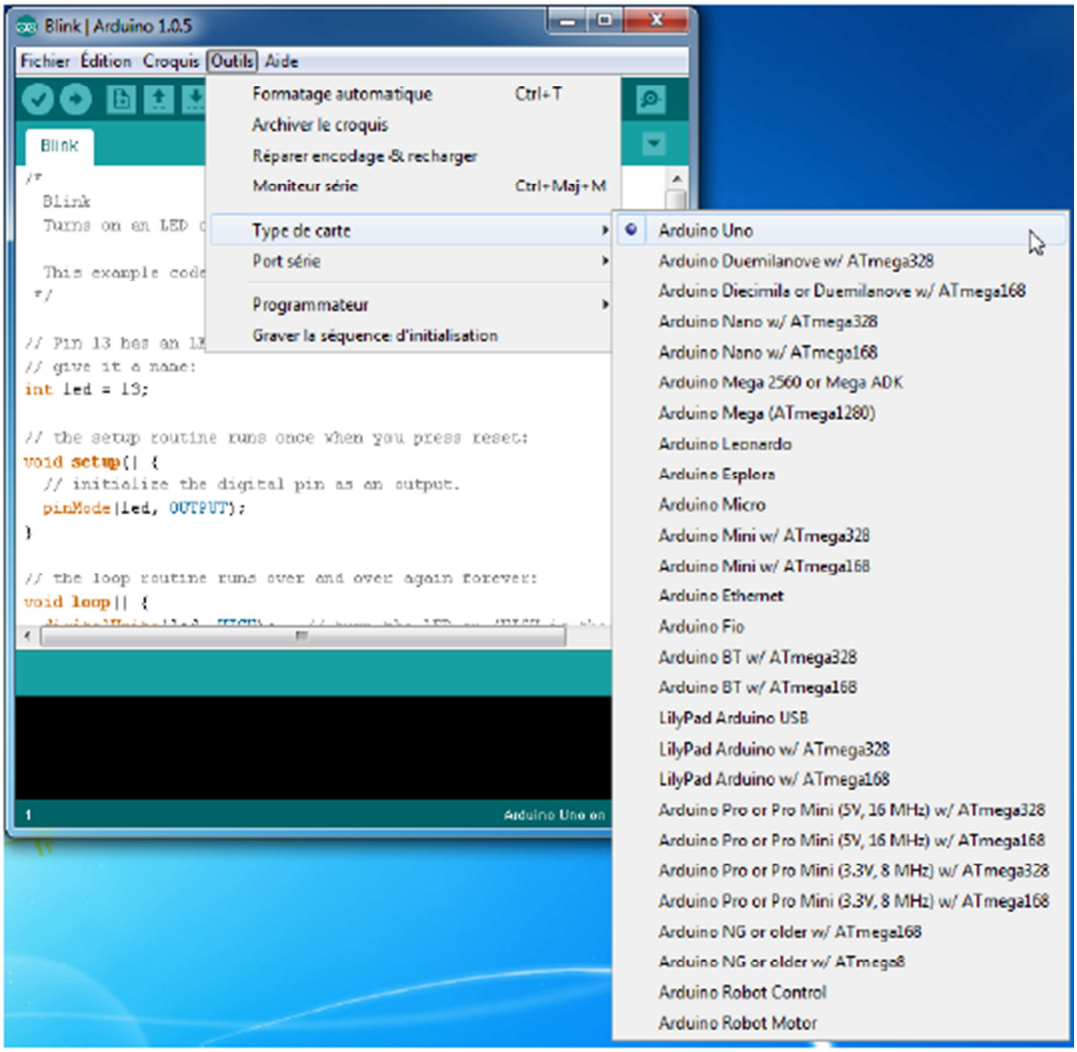


Figure I.10 Choisir la carte que l'on va programmer

I.6.3 Choisir le port de connexion de la carte

On va dans le menu Tools, puis Serial port. Là, on choisit le port COMX, X étant le numéro du port qui est affiché. On ne prend pas COM1 car il n'est quasiment jamais connecté à la carte. Dans notre cas, il s'agit de COM5.

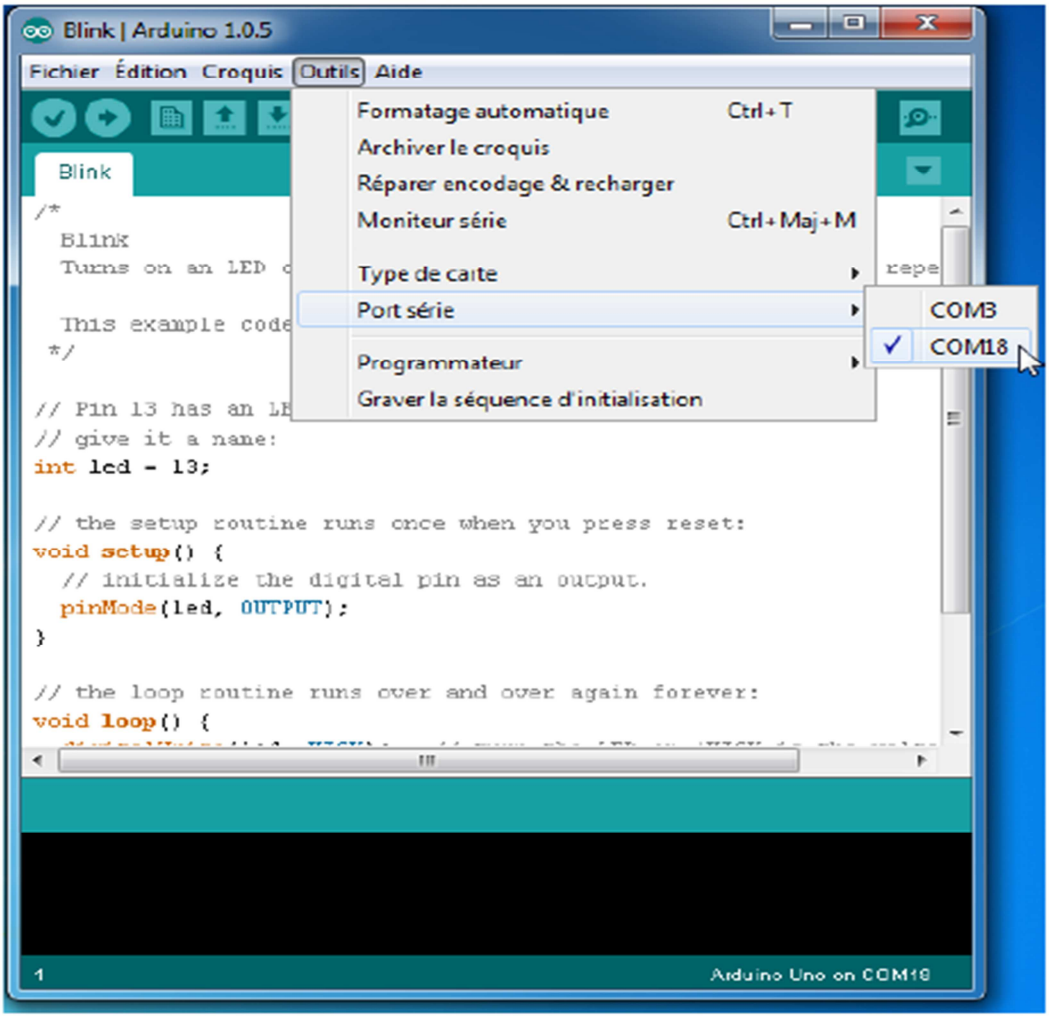


Figure I.11 Choisir le port de connexion de la carte

Pour trouver le port de connexion de la carte, on peut aller dans le ***gestionnaire de périphérique*** qui se trouve dans le ***panneau de configuration***. On regardent à la ligne ***Ports (COM et LPT)*** et là, on voix ***Arduino Uno (COMX)***.

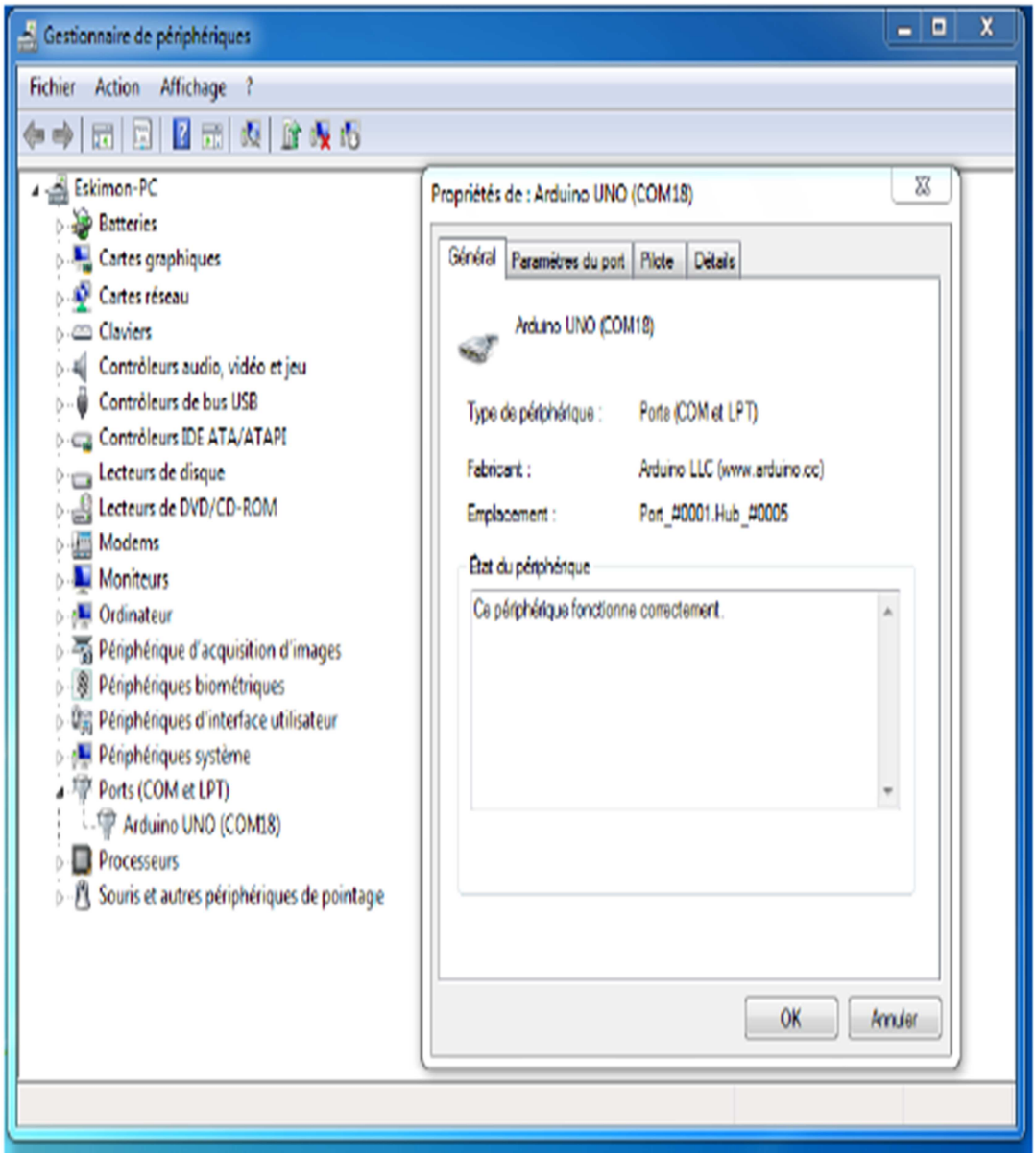


Figure I .12 La configuration du port de connexion

I .6.4 Téléversement

Maintenant, il va falloir envoyer le programme dans la carte. Pour ce faire, il suffit de cliquer sur le bouton *Téléverser*, en jaune orangé sur la photo.

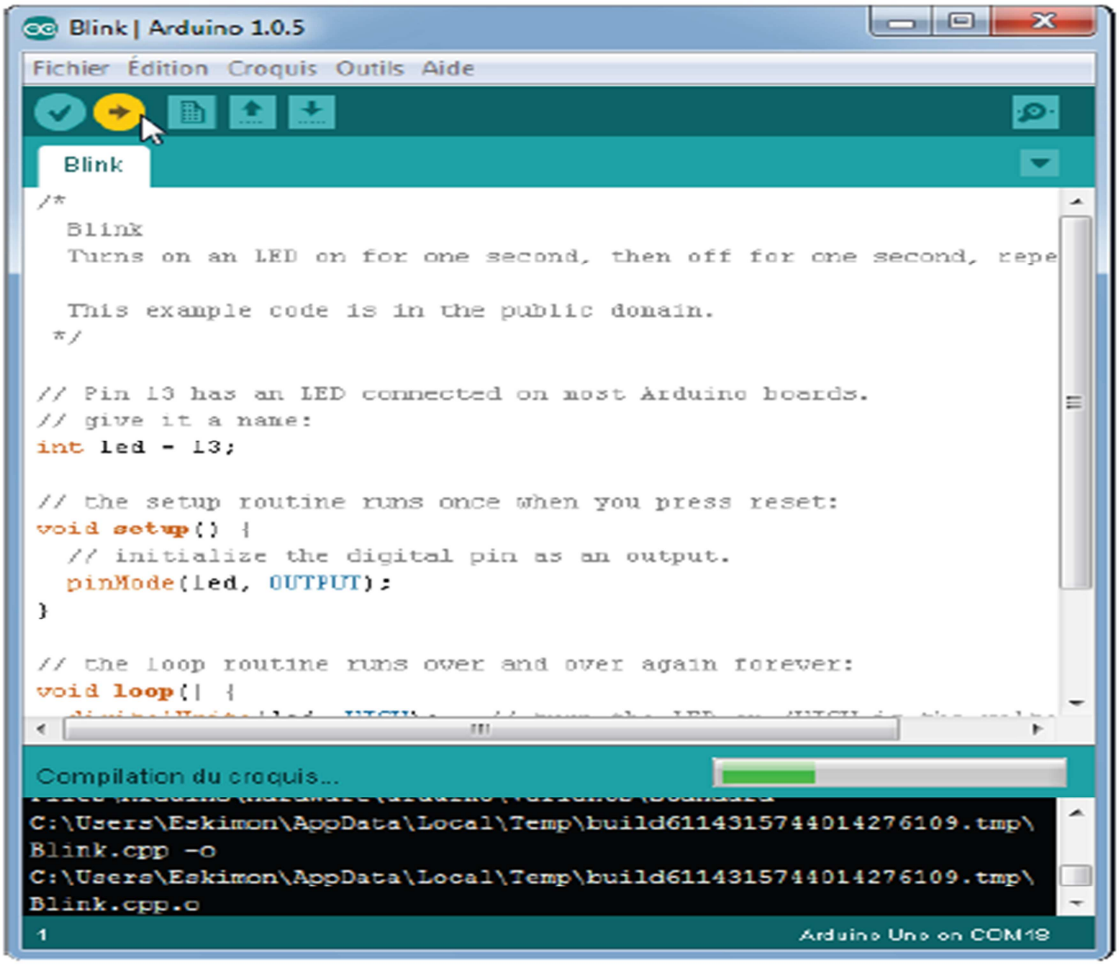


Figure I.13 Début Téléchargement du programme

On voix tout d'abord le message "Compilation du croquis en cours…" pour vous informer que le programme est en train d'être compilé en langage machine avant d'être envoyé. Ensuite on obtient ceci :

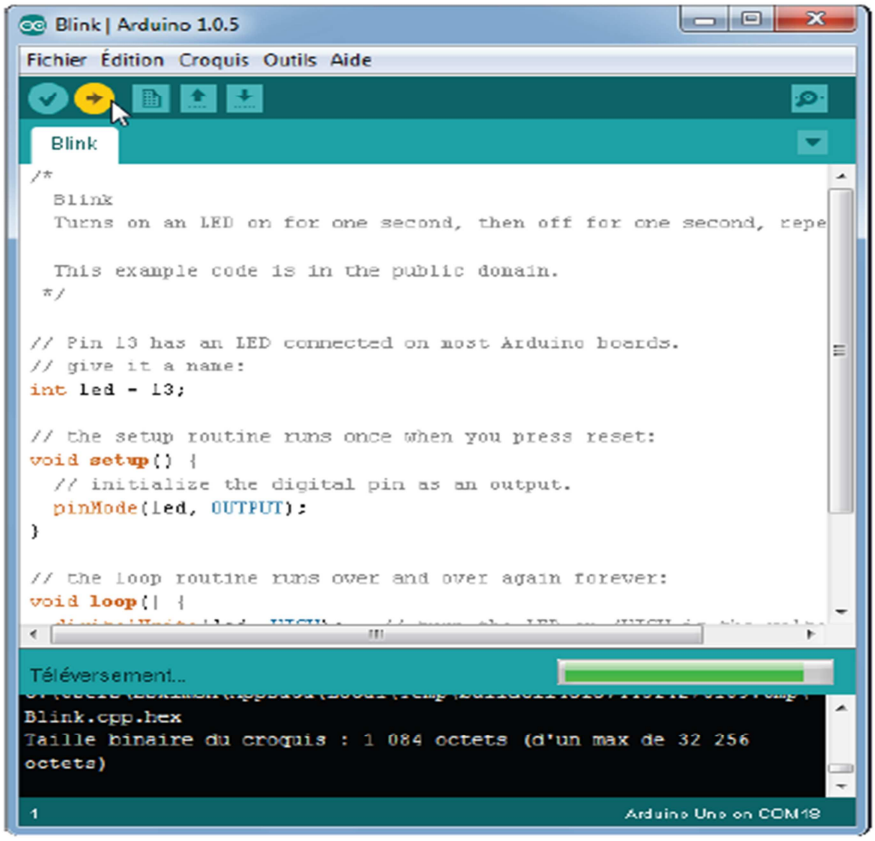


Figure I.14 Fin de téléversement du programme dans la carte arduino

En bas dans l'image, vous voyez le texte : "Téléversement…", cela signifie que le logiciel est en train d'envoyer le programme dans la carte. Une fois qu'il a fini, il affiche un autre message :

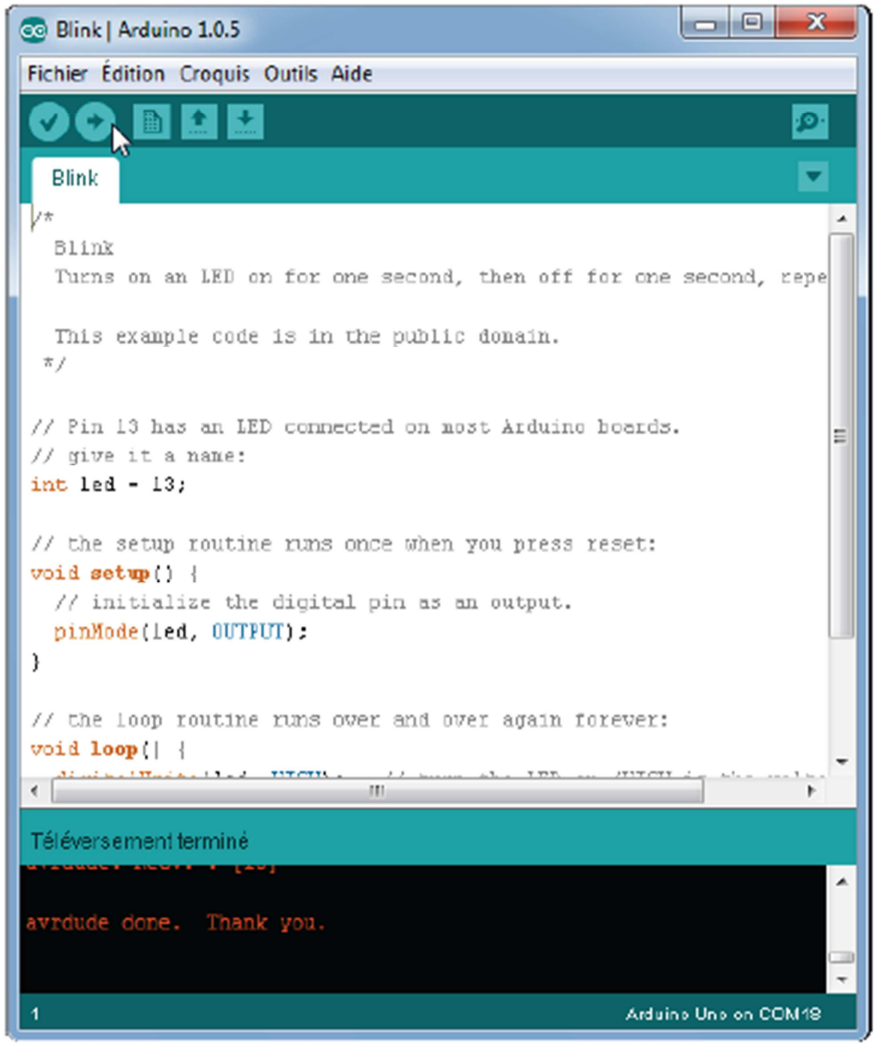


Figure I.15 Compilation du programme

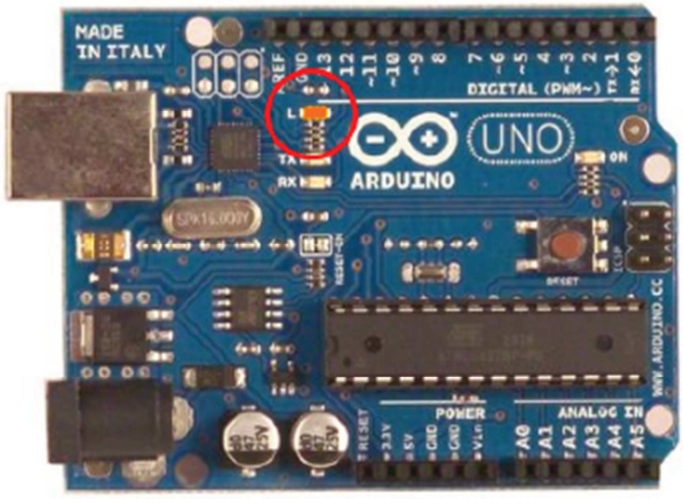
Le message afficher : “Téléversement terminé” signale que le programme à bien été chargé dans la carte. Si votre matériel fonctionne, vous devriez avoir une LED sur la carte qui clignote :

Si on obtient pas ce message mais plutôt un truc en rouge, pas d’inquiétude, le matériel n’est pas forcément défectueux! En effet, plusieurs erreurs sont possibles:

●l’IDE recompile avant d’envoyer le code, vérifier la présence d’erreur

●Lavoie série est peut être mal choisi, vérifier les branchements et le choix de la voie série

●l’IDE est codé en JAVA, il peut être capricieux et bugger de temps en temps (surtout avec la voie série…) : réessayez l’envoi



FigureI.16 LED 13 clignote

I .7 Partons du programme

Le contenu

Le contenu du programme, donc le programme en lui-même,est ce qui va définirchaque action que va exécuter la carte Arduino. Mais ce n'est pas tout Dans leprogramme il y a plusieurs zones, que nous verrons plus en détail tout au long de lalecture de ce cours, qui ont chacune un rôle particulier.

●La première zone sert principalement (je ne vais pas m'étendre) à dire à la cartedégrader en mémoire quelques informationsqui peuvent être : l'emplacementd'un élément connecté à la carte, par exemple une LED en broche 13, ou bienune valeur quelconque qui sera utile dans le programme. [6]

```
// constants won't change. They're used here to
// set pin numbers:
const int ledPin =  13;      // the number of the LED pin
```

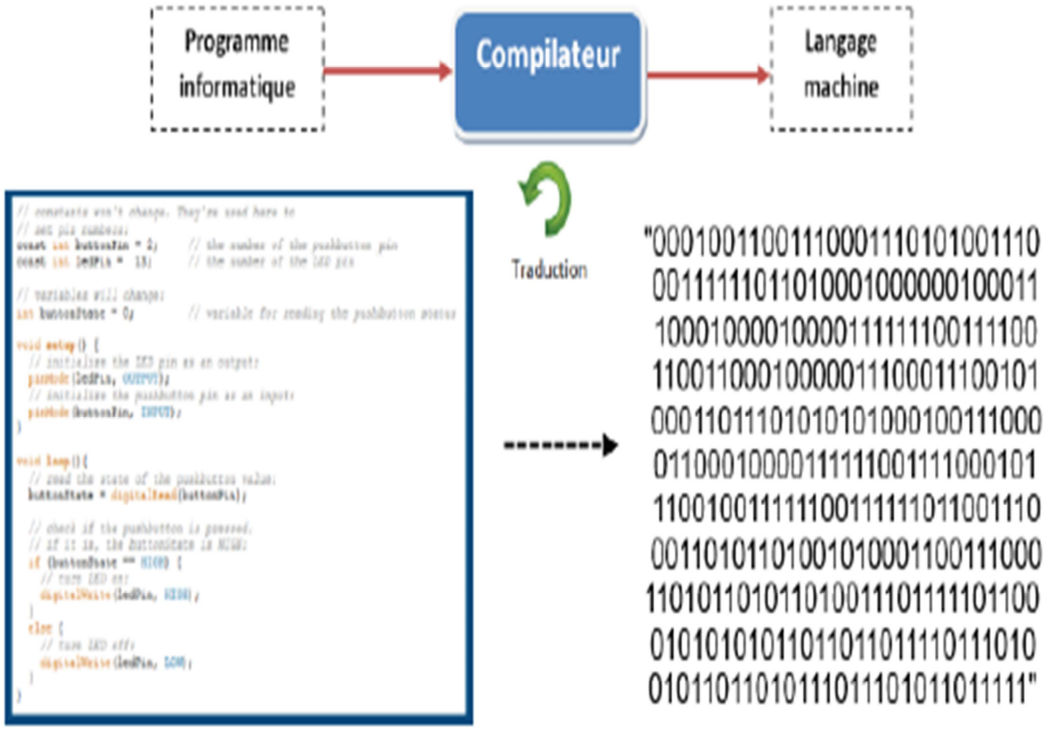
●La zone secondaire est l'endroit où l'on va **initialiser certains paramètres** duprogramme. Par exemple, on pourra dire à la carte qu'elle devra communiqueravec l'ordinateur ou simplement lui dire ce qu'elle devra faire de le LED qui estconnectée sur sa broche 13. On peut encore faire d'autres choses, mais nous leverrons plus tard.

```
void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
}
```

●La dernière zone est la **zone principale où se déroulera le programme**. Tout ce qui va être écrit dans cette zone sera exécuté par la carte, se sont les actions quela carte fera. Par exemple, c'est ici qu'on pourra lui dire de faire clignoter la LEDsur sa broche 13. On pourra également lui demander de faire une opération telleque 2+2ou bien d'autres choses encore.

Et l'envoi

Le programme est envoyé dans la carte lorsque vous cliquez sur le bouton. Le logiciel Arduino va alors vérifier si le programme ne contient pas d'erreur et ensuite le compiler (le traduire) pour l'envoyer dans la carte.



FigureI.18 La traduction du programme en langage machine

I .8 Discussion

Dans ce chapitre, nous avons projeté la lumière sur une carte d'acquisition qui estl'Arduino donnant ainsi les raisons pour lesquelles on l'a choisie, puis nous avons cité desdifférents types de cette dernière. Ensuite, nous avons expliqué les deux parties essentielles de l'Arduino; (la partie matérielle et la partie de programmation) plus précisément. Nousavons également expliqué le principe de fonctionnement de la carte Arduino sans oublier sescaractéristiques.

Le chapitre suivant sera consacré à l'étude et la présentation de différent matériel nécessaire pour la réalisation noter tapis roulants.

CHAPITRE II

LES DIFFERENTS CAPTEURS ET MOTEURS

II.1 Préambule

Dans ce chapitre, on va présenter des généralités sur les capteurs infrarouges et le moteur à courant continu.

Ce travail d'initiation a base d'une carte Arduino UNO permet de commander un tapis roulant industriel automatisé.

II .2 Capteur

II.2.1 Définition

Un **capteur** est un dispositif transformant l'état d'une grandeur physique observée en une grandeur utilisable, telle qu'une tension électrique, une hauteur de mercure, une intensité ou la déviation d'une aiguille. On fait souvent la confusion entre capteur et transducteur : le capteur est minimum constitué d'un transducteur.

II .2.2 Principes des capteurs

Avant de donner la définition d'un capteur, il est nécessaire de connaître quelques définitions de métrologie.

Tout d'abord, il faut savoir que mesurer une grandeur physique c'est attribuer une valeur quantitative en prenant pour référence une grandeur de même nature appelée unité. Ensuite d'autres définitions doivent être connues comme :

- Le mesurande : c'est l'objet de la mesure ou plus simplement la grandeur à mesurer.
- Le mesurage : c'est l'ensemble des opérations pour déterminer la valeur du mesurande.
- La mesure c'est le résultat du mesurage. Autrement dit c'est la valeur du mesurande.

Ces définitions permettent de donner une définition claire d'un capteur. En effet un capteur est un dispositif dont les caractéristiques physiques sont sensibles à un mesurande. Lorsque celui-ci est soumis à ce mesurande il fournit une réponse sous la forme d'une grandeur physique exploitable qui est en général de nature électrique. [8]

Le schéma suivant synthétise bien ce qu'est un capteur :

Une relation mathématique tirée des lois physiques entre la grandeur d'entrée et la grandeur de sortie doit exister. Cette relation entre le mesurande m et la sortie s (s = f(m)) s'appelle courbe d'étalonnage du capteur.

Le capteur est dit linéaire si la courbe d'étalonnage est une droite ou sinon le capteur est dit non linéaire.

II .2.3 Classification des capteurs

On peut classer les capteurs de plusieurs manières

- par le mesurande qu'il traduit (capteur de position, de température, de pression, etc)
- par son rôle dans le processus industriel (contrôle de produit finis, de sécurité, etc.)
- par le signal qu'il fournit en sortie qui peut être numérique ou analogique.
- par leur principe de traduction du mesurande(capteur résistif, piézoélectrique, etc)
- par leur principe de fonctionnement : capteur Actif ou Passif .[9]

II .2.3.1 Les capteurs actifs

- Ce capteur fonctionne comme un générateur, dès qu'il est soumis à l'action d'un mesurande celui-ci transforme celle-ci en une grandeur directement exploitable à savoir en énergie électrique.
- Le tableau suivant donne les principes physiques les plus utilisés en fonction d'un mesurande :

Mesurande	Energie propre du mesurande	Principe physique	Grandeur de sortie
Température	Énergie thermique	Effet thermoélectrique	Tension
Fluxlumineux	Énergie électromagnétique	Effet photovoltaïque	Charge
Force	Énergie mécanique	Effet piézoélectrique	Courant
Pression	Énergie mécanique	Effet piézoélectrique	Tension
Accélération	Énergie mécanique	Effet piézoélectrique	Charge
Vitesse	Énergie mécanique	effet élecomagnétique	Tension
Position	Énergie mécanique	Effet hall	Tension

Tableau II.1 Tableau des capteurs actifs

II .2.3.2 Les capteurs passifs

Un capteur passif est considéré comme une impédance dont l'un des paramètres est sensible au mesurande. Cette impédance doit ensuite être intégrée dans un circuit pour pouvoir retrouver une grandeur électrique en sortie. Le montage qui permet ceci est appelé conditionneur. Il existe plusieurs sortes de conditionneur comme le montage potentiométrique, le pont de Wheatstone, les circuits oscillants ou les amplificateurs opérationnels.

Le tableau suivant donne différents capteurs passifs :

Mesurande	Caractéristique électrique sensible	Type de matériaux utilises
Température	Constante diélectrique	Plaine : nickel :cuivre :verres
Flux lumineux	Résistivité	Semi-conducteur
Déformation	Perméabilité magnétique	Alliages de nickel, silicium
Position	Résistivité	Matériaux magnéto résistants
Humidité	Constante diélectrique	Alumine, polymères
Niveau	Constante diélectrique	Liquides isolants

Tableau II.2 Tableau des capteurs passifs

II .2 .4 Caractéristiques d'un capteur

Un capteur est caractérisé selon plusieurs critèresdont les plus courants sont :

Etendue de mesure : valeurs extrêmes pouvant être mesurée par le capteur

Résolution : plus petite variation de grandeur mesurable par le capteur.

Sensibilité : variation du signal de sortie par rapport a la variation du signal d'entée.

Précision : aptitude de capteur adonner une mesure proche de la valeur vraie.

Rapidité : tempes de réaction du rapporte, la rapidité et liée a la bande passante.

Linéarité : représente l'écart des sensibilités sur l'étendue de mesure.

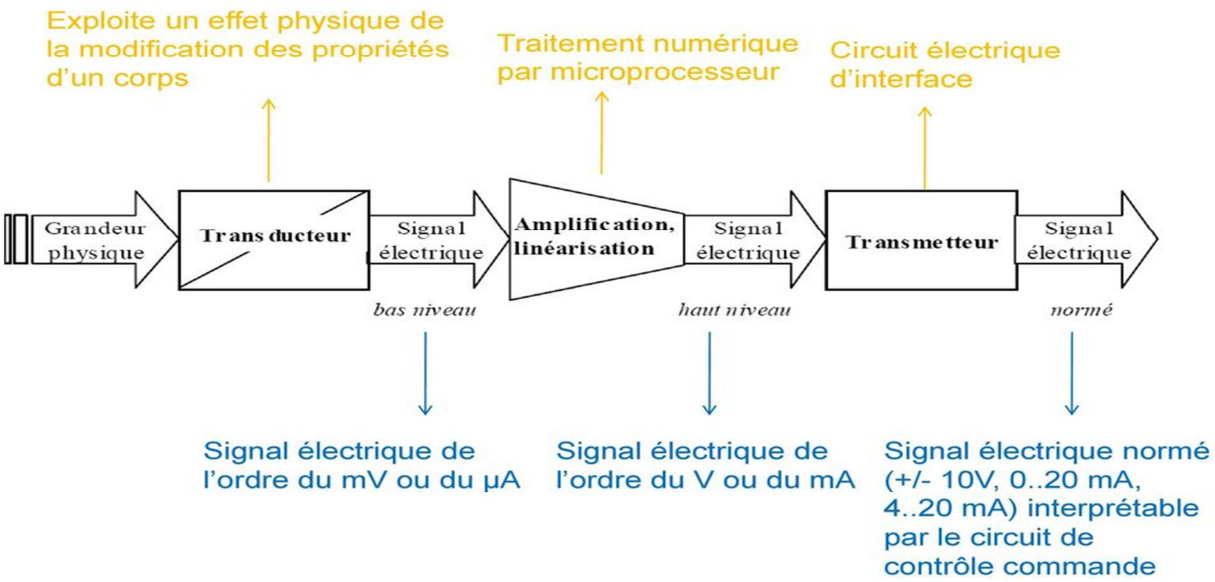


Figure II.1 Schéma de principe d'n capteur industriel

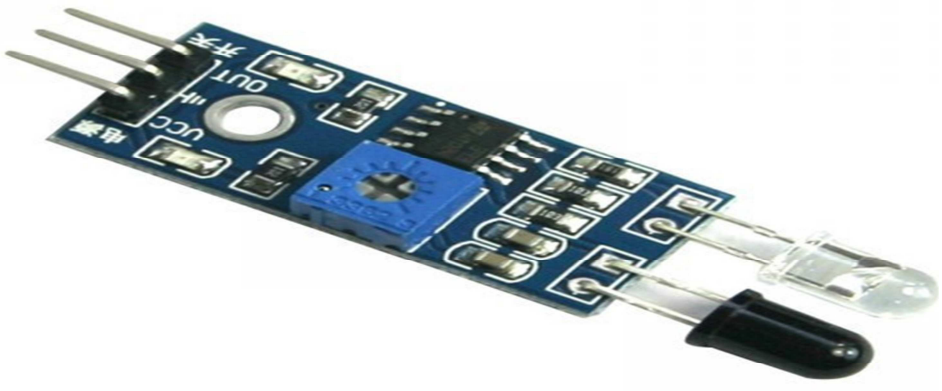
II .3 Le capteur d'obstacle infrarouge MH SENSOR SERIES

II .3.1 Définition

Le rayonnement **infrarouge** (**IR**) est un rayonnement électromagnétique de longueur d'onde supérieure à celle du spectre visible mais plus courte que celle des micro-ondes.

Cecapteur est idéal pour détecter un obstacle facilement avec une carte Arduino. Il sera un outil parfait pour tous nos projets de robotique.

Il est composé d'une led émettrice infrarouge, et d'une photodiode qui va détecter la lumière infrarouge en réflexion sur un objet. Un potentiomètre permet de régler la sensibilité. Un câblé au standard grove est fourni avec ce module.[10]



FigureII.2 Capteur d’obstacle infrarouge

II .3.2 Caractéristiques

- Compatible de l'interface Grove
- Alimentation : 5V
- Dimension : 20 mm x 20mm

II.3.3 Broches de connexion

•VCC = alimentation +5v

•GND=Masse de l’alimentation

•OUT = sortie de mesure donnée

Spécification du capteur MH senor-series

Paramètre	Min	Type	Max	Unité
Tension d'alimentation	4.5	5.0	5.5	V
Courant de fonctionnement	10	15	20	mA
Courant de repos	1.5	2	2.5	mA
Fréquence de fonctionnement		38		KHZ

Tableau II.3 Tableau de spécification capteur Mh senor-series

II.3.4 Principe de fonctionnement

Le capteur infrarouge est basé sur ce principe ; ilest constitué d'un émetteur de lumière infrarouge et un récepteurdétecté la réflexion d'intensité c'est-à-dire après l'émission de la lumière infrarouge si un obstacle cours dans le champs de vision du corrupteur la radiation sera réfléchi par l'obstacle et le récepteur ; la mesure allait de potentiomètre on pourra régler la distance ou bien l'ordre de détection c'est-à-dire choisir à quelle distance le capteur nous alertent la présence d'un obstacle il est constitué également d''une LED qui s'allume une fois le capteur est bronché et une LED qui s'allume lors de la détection de l'objet.

II .4 Moteur à courants continu

II.4.1 Généralité

On rencontre encore régulièrement des moteurs à courant continu à excitation indépendante dans les salles des machines des immeubles d'un certain âge. En général, ils font partie qui permet d'aisément faire varier la vitesse de rotation.

II.4.2 Définition du moteur courant continu

Les moteurs courant continu sont des convertisseurs de puissance :

Soit ils convertissent l'énergie électrique absorbée en énergie mécanique lorsqu'ils sont capables de fournir une puissance mécanique suffisante pour démarrer puis entraîner une charge en mouvement. On dit alors qu'ils ont un fonctionnement en moteur.

Soit ils convertissent l'énergie mécanique reçue en énergie électrique lorsqu'ils subissent l'action d'une charge entraînante. On dit alors qu'ils ont un fonctionnement en générateur.

- En mode « moteur », l'énergie électrique est convertie en énergie mécanique.
- En mode « générateur » ou « dynamo », l'énergie mécanique est convertie en une énergie électrique capable de se comporter comme un frein. [11]

II .4.3 Fonctionnement d'un moteur DC

Il existe des dizaines de **moteurs DC** différents : taille, voltage, ampérage, …

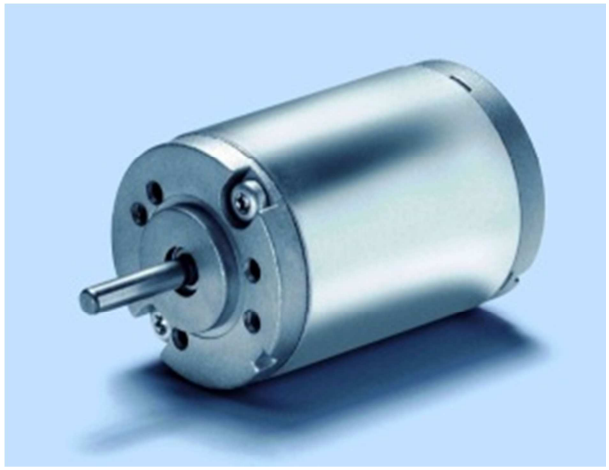


Figure II.3 Moteur à courant continu

Un **moteur DC** est un convertisseur électromécanique permettant la conversion bidirectionnelle d'énergie entre une installation électrique parcourue par un courant continu et un dispositif mécanique ; selon la source d'énergie. Pour faire simple, cela signifie qu'un moteur à courant continu va pouvoir convertir de l'électricité en énergie mécanique. **Les moteurs DC** ont ainsi la particularité de pouvoir fonctionner dans les 2 sens, suivant la manière dont le courant lui est soumis.

Le schéma suivant explique le fonctionnement d'un **moteur à courant continu**:

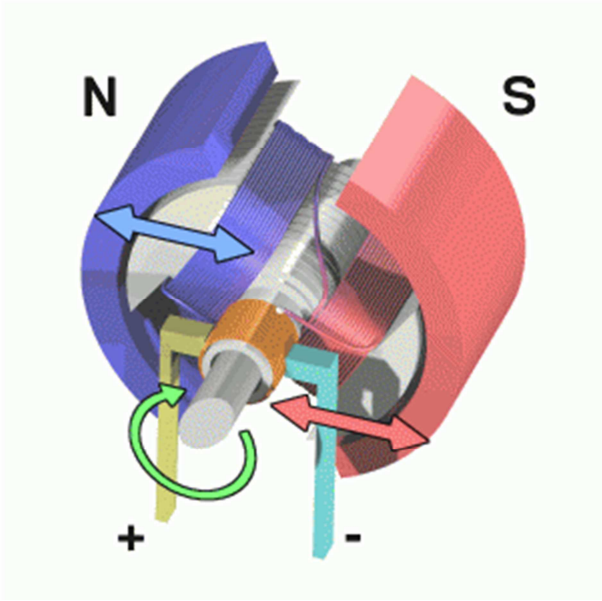


Figure II.4 Schéma de l'inducteur ou du stator

II .4.4 Principe de fonctionnement

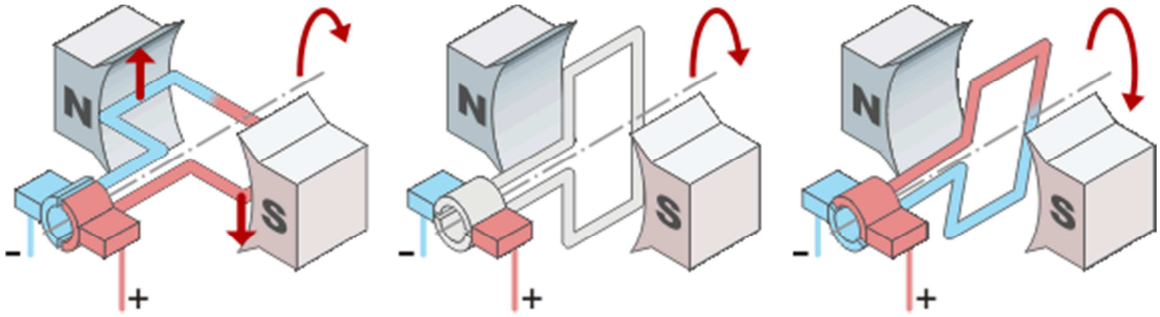
Le moteur à courant continu se compose :

- de l'inducteur ou du stator,
- de l'induit ou du rotor,
- du collecteur et des balais.

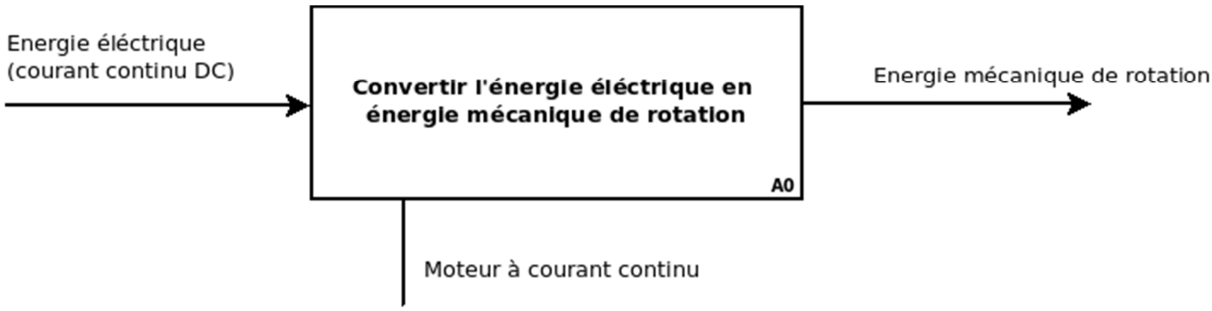
Lorsque le bobinage d'un inducteur de moteur est alimenté par un courant continu, sur le même principe qu'un moteur à aimant permanent (comme la figure ci-dessous), il crée un champ magnétique (flux d'excitation) de direction Nord-Sud.

Une spire capable de tourner sur un axe de rotation est placée dans le champ magnétique. De plus, les deux conducteurs formant la spire sont chacun raccordés électriquement à un demi collecteur et alimentés en courant continu via deux balais frotteurs.

D'après la loi de Laplace (tout conducteur parcouru par un courant et placé dans un champ magnétique est soumis à une force), les conducteurs de l'induit placés de part et d'autre de l'axe des balais (ligne neutre) sont soumis à des forces F égales mais de sens opposé en créant un couple moteur : l'induit se met à tourner.



Si le système balais-collecteurs n'était pas présent (simple spire alimentée en courant continu), la spire s'arrêterait de tourner en position verticale sur un axe appelé communément "ligne neutre". Le système balais-collecteurs a pour rôle de faire commuter le sens du courant dans les deux conducteurs au passage de la ligne neutre. Le courant étant inversé, les forces motrices sur les conducteurs le sont aussi permettant ainsi de poursuivre la rotation de la spire.



Dans la pratique, la spire est remplacée par un induit (rotor) de conception très complexe sur lequel sont montés des enroulements (composés d'un grand nombre de spires) raccordés à un collecteur "calé" en bout d'arbre. Dans cette configuration, l'induit peut être considéré comme un seul et même enroulement semblable à une spire unique.

II.4.5 Les caractéristiques du moteur courant continu

Le moteur courant continu (DC) est caractérisé par une constante de vitesse, et une pente vitesse/couple. Le courant est proportionnel à la charge ; et la vitesse est proportionnelle à la tension d'alimentation.

Le moteur courant continu sans fer ne présente pas de couple magnétique résiduel et les agitations électromagnétiques sont dérisoires. Son rendement, qui atteint 90%, surpasse celui des autres technologies de moteurs.

Son rotor en forme de cloche donne la possibilité d'accélérations très importantes et d'un couple de retenue inexistant.

Cela permet d'obtenir des positionnements précis et des vitesses faibles.

Le rotor est traditionnellement composé d'un stator à aimant permanent en terre rare de type Al-nico, samarium cobalt ou néodyme fer bore qui expliquent les caractéristiques dynamiques très élevées..

II .4.6 Les avantages / inconvénients du moteur courant continu

II.4.6.1 Les avantage

- possibilité d'entrainer de très fortes inerties
- forte constante de temps mécanique
- forte capacité à entrainer des surcharges élevées imprévisibles ralentissant le moteur : puisque son courant est proportionnel au couple, le moteur courant continu peut franchir des pointes de couple, et ainsi éviter les phénomènes de décrochage.

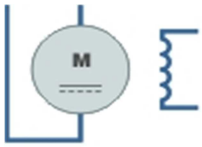
II.4.6.2 Les inconvénients

- plus la vitesse de rotation est élevée, plus la pression des balais doit augmenter pour rester en contact avec le collecteur donc plus le frottement est important ;
- aux vitesses élevées les balais doivent donc être remplacés très régulièrement ;
- le collecteur imposant des ruptures de contact provoque des arcs, qui usent rapidement le commutateur et génèrent des parasites dans le circuit d'alimentation, ainsi que par rayonnement électromagnétique (réduit dans le cas des moteurs maxon par le système CLL (long life capacitor).

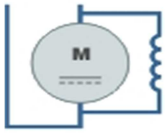
II.4.7 Type de moteur à courant continu

Suivant l'application, les bobinages du l'inducteur et de l'induit peuvent être connectés de manière différente. On retrouve en général :

Des moteurs à excitation indépendante.

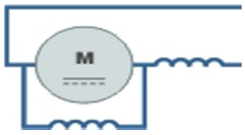


Des moteurs à excitation parallèle.



Des moteurs à excitation série.

Des moteurs à excitation composée.



La plupart des machines d'ascenseur sont configurées en excitation parallèle ou indépendante. L'inversion du sens de rotation du moteur s'obtient en inversant soit les connections de l'inducteur soit de l'induit.

II.4.8 L'INDECTEUR

L'inducteur d'un moteur à courant continu est la partie statique du moteur. Il se compose principalement :

- de la carcasse,
- des paliers,
- des flasques de palier,
- des portes balais.

Le cœur même du moteur comprend essentiellement :

- Un ensemble de paires de pôles constitué d'un empilement de tôles ferromagnétiques.
- Les enroulements (ou bobinage en cuivre) destinés à créer le champ ou les champs magnétiques suivant le nombre de paires de pôles.

Pour des moteurs d'une certaine puissance, le nombre de paires de pôles est multiplié afin de mieux utiliser la matière, de diminuer les dimensions d'encombrement et d'optimaliser la pénétration du flux magnétique dans l'induit.

L'induit du moteur à courant continu est composé d'un arbre sur lequel est empilé un ensemble de disques ferromagnétiques. Des encoches sont axialement pratiquées à périphérie du cylindre formé par les disques empilés. Dans ces encoches les enroulements (bobines de l'induit) sont "bobinés" selon un schéma très précis et complexe qui nécessite une maind'œuvre particulière (coûts importants). Pour cette raison, on préfère, en général, s'orienter vermoteurs à courant alternatif plus robuste et simple dans leur conception.

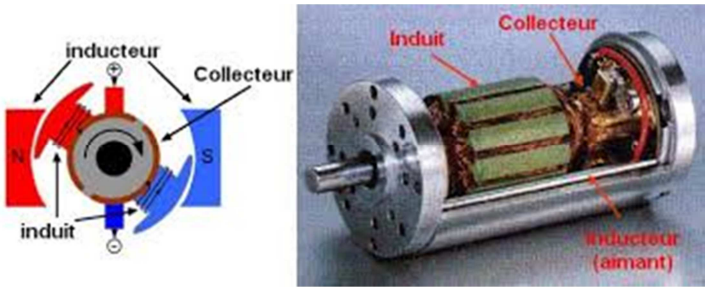


Figure II.5 Les principe bloc d’un moteur a courant continu

Chaque enroulement est composé d'une série de sections, elles même composées de spire une spire étant une boucle ouverte dont l'aller est placé dans une encoche de l'induit et le relou dans l'encoche diamétralement opposée. Pour que l'enroulement soit parcouru par un courant, ses conducteurs de départ et de retour sont connectés aux lames du collecteur (cylindre calé sur l'arbre et composé en périphérie d'une succession de lames de cuivre espacée par un isolant).

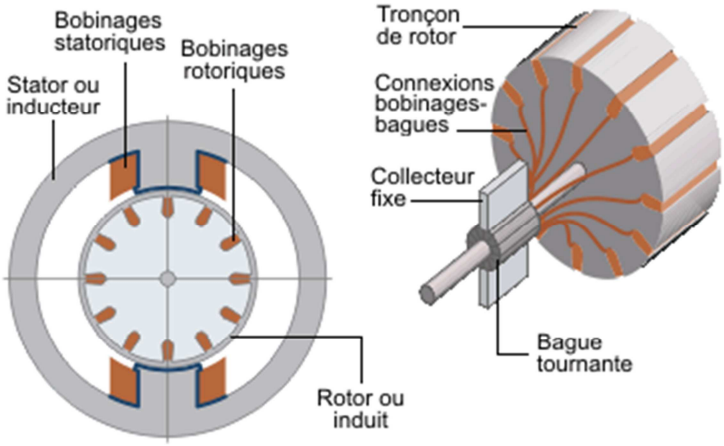


Figure II.6

Composition de l'induit.

L'interface entre l'alimentation à courant continu et le collecteur de l'induit est assuré par les balais et les porte-balais.

II.4.9 Les balais

Les balais assurent le passage du courant électrique entre l'alimentation et le bobinage de l'induit sous forme d'un contact par frottement. Les balais sont en graphite et constituent, en quelques sortes, la pièce d'usure. Le graphite en s'usant libère une poussière qui rend le moteur à courant continu sensible à un entretien correct et donc coûteux.

Figure II.6 Les balais d'un moteur a courant continu.

Le point de contact entre les balais et le collecteur constitue le point faible du moteur à courant continu. En effet, c'est à cet endroit, qu'outre le problème d'usure du graphite, la commutation (inversion du sens du courant dans l'enroulement) s'opère en créant des micros-

arcs (étincelles) entre les lamelles du collecteur; un des grands risques de dégradation des collecteurs étant leur mise en court-circuit par usure.

II .5 Le L293D

II.5.1 Définition

Le composant **L293D** est un pont de puissance composé de plusieurs transistors et relais qui permet d'activer la rotation d'un moteur. Ce composant ne coûte pas cher.Le**L293D** est un double pont-H, ce qui signifie qu'il est possible de l'utiliser pour commander quatre moteurs distincts (dans un seul sens) grâce à ses 4 canaux.
Il est également possible de constituer deux pont-h afin de piloter deux moteurs distincts, dans les deux sens et indépendamment l'un de l'autre.

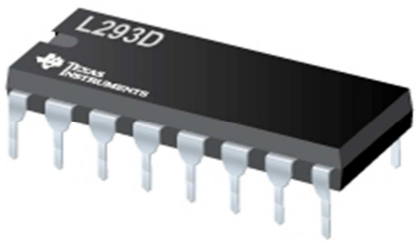


Figure II.7 Le L293d

Il est important de noter que le **L293D** peut délivrer au maximum 600mA, veuillez donc choisir vos moteurs en conséquence. [12]

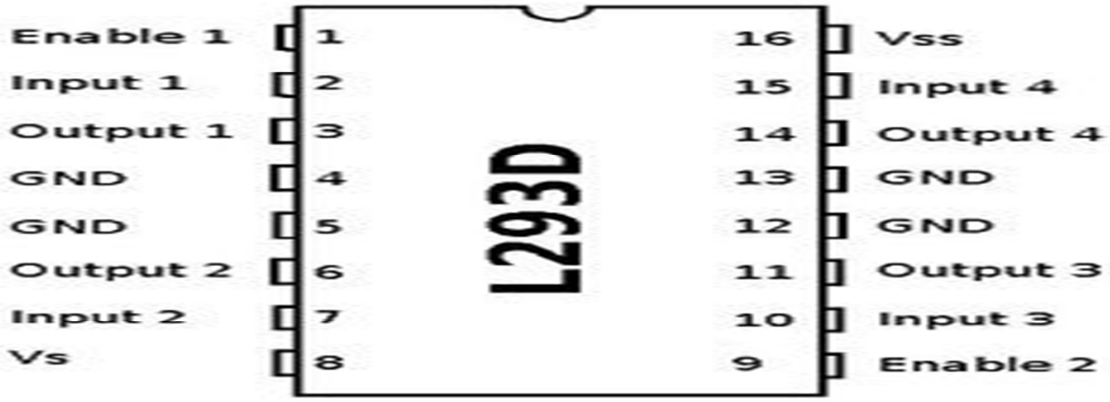
II .5.2 Caractéristiques techniques du L293D

Voici les caractéristiques techniques du composant **L293D**:

- Nbre de pont-H: 2
- Courant Max Régime continu: 600mA (x2)
- Courant de pointeMax< 2ms: 1200mA
- VS Max Alim moteur: 36v
- VSS Max Alim logique: 7v
- Nbre de Broches: 16 DIP
- Perte de tension: 1.3v

II 5.3 Bronchement du L293D

Le schéma suivant détaille les différentes broches du composant **L293D**:



FigureII.8 Le datasheet du l293d

1. Enable1: permet d'envoyer (ou pas) la tension sur les sorties du moteur via OUTPUT1 et OUTPUT2 et commande l'activation/désactivation du premier Pont-H. Si ENABLE1 = GND, le pont-H est déconnecté et le moteur ne fonctionne pas. Si ENABLE1 = VSS, le pont-H est connecté aux sorties et le moteur fonctionne dans un sens ou l'autre ou pas en fonction des tensions appliquée sur INPUT1 & INPUT2.
2. Input1: avec Input 2, sont les broches de commande du Pont-H Output1/Output2. Il sera directement brancher à votre Arduino pour commander le sens du courant entre Output 1 et Output 2.
3. Ouput1: avec Output 2, sera branché directement sur le moteur.
4. GND: qui doit être raccorder à la masse de la source d'alimentation de puissance VS et à la masse de la source d'alimentation de la logique "VSS" (donc GND Arduino).
5.
6. Ouput2: avec Output 1, sera branché directement sur le moteur.
7. Input2: avec Input 1, sont les broches de commande du Pont-H Output1/Output2. Il sera directement brancher à votre Arduino pour commander le sens du courant entre Output 1 et Output 2.
8. VS: Alimentation de puissance des moteurs.
9. Enable2: commande l'activation du second pont-H constitué de Output3/Output4

10. Input3: avec Input 4, sont les broches de commande du Pont-H Output3/Output4. Il sera directement brancher à votre **Arduino** pour commander le sens du courant entre Output 3 et Output 4.
11. Ouput3: avec Output 4, sera branché directement sur le moteur.
12. GND
13. GND
14. Ouput4: avec Output 3, sera branché directement sur le moteur.
15. Input4: avec Input 3, sont les broches de commande du Pont-H Output3/Output4. Il sera directement brancher à votre **Arduino** pour commander le sens du courant entre Output 3 et Output 4.
16. VSS: Alimentation de la logique de commande (5V). A raccorder à la borne +5V de votre **Arduino**.

Veuillez noter que les pins Enable1 et Enable2 permettent moduler la vitesse du moteur en utilisant des broches **PWM** de votre **Arduino**. Si vous ne souhaitez pas moduler la vitesse du **moteur DC**, il nous suffit de brancher Enable1 sur la broche VSS de votre **Arduino**.

II.5.4 Fonctionnement du L293D

Le tableau suivant nous permet de faire fonctionner notre moteur DC en utilisant le L293d

Enable 1	Input 1	Input 2	Fonction
High	Low	High	Tourne dans le sens horlogique
High	High	Low	Tourne dans le sens anti-horlogique
High	Low	High	Stop
High	High	High	Stop
Low	Non applicable	Non applicable	Stop

Tableau II.4 Le Fonctionnement du L293D

II .6 Discussion

Dans ce chapitre nous avons fait une étude approfondie sur tout le matériel nécessaire pour la réalisation de notre tapis roulant, ainsi que le choix des composants convenables pour notre application.

Le chapitre suivant sera réservé à la réalisation pratique et la programmation de notre application.

CHAPITRE III

LA REALISATION PRATIQUE

III .1 Préambule

Apre avoir étudiéthéoriquement, ensuite on passe a la réalisation pratique, passant par des processus de la réalisation.

Dans le 2éme chapitre on précisé tout les composants nécessaire pour la réalisation de notre tapis roulant, et dans ce dernier il nous reste que à le réaliser pratiquement. Pour cela il nous faut suivre une méthode que ca soit dans la réalisation matérielle ou logicielle.

III.2 La réalisation matérielle :

Structure de notre tapis roulant :

Pour construire notre tapis roulant on a choisi plusieurs matériaux, tout d'abord le chassie en bois, et puis on a jouté a chaque extrémité des rouleaux en plastique avec des axes qui sont attaché a des roulements sur le chassie, ainsi que assurer une rotation libre, et dans l'autre cotéon placé un axe de commande relié avec le moteur, pour assurer le déplacement de l'objet on a mis un tapis posé sur les rouleaux de plastique.

La commande de moteur exige trois capteurs infrarouges, deux capteurs pour détecter la présence d'objet et le troisième pour détecter la fin course de l'objet, et un bouton poussoir dans le cas ou l'objet n'est pas détecté par le capteur infrarouge.

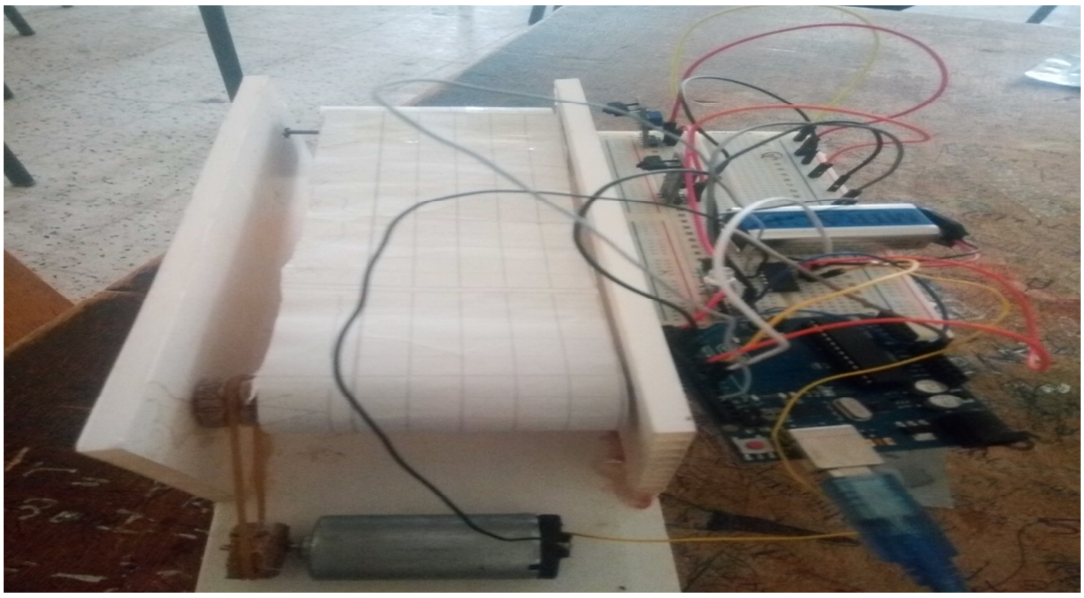


Figure III.1 : Le tapis roulant

III.3 Branchement du capteur Infrarouge :

Comme son nom l'indique il s'agit d'un capteur de proximité, muni de deux LEDinfrarouges. Une transparente et une plus foncée. L'une correspondant à la LED émettrice et la LED réceptrice. Lorsqueun corps quelconque passe devant le faisceau il le renvoi alors a la diode réceptrice. Et détecte ainsi le corps à proximité.[13]

Il est constitué de trois branches, on peut le relier facilement à une carte Arduino.

Vcc : à l'alimentation

Gnd : à la masse de l'alimentation

Out : à la carte Arduino

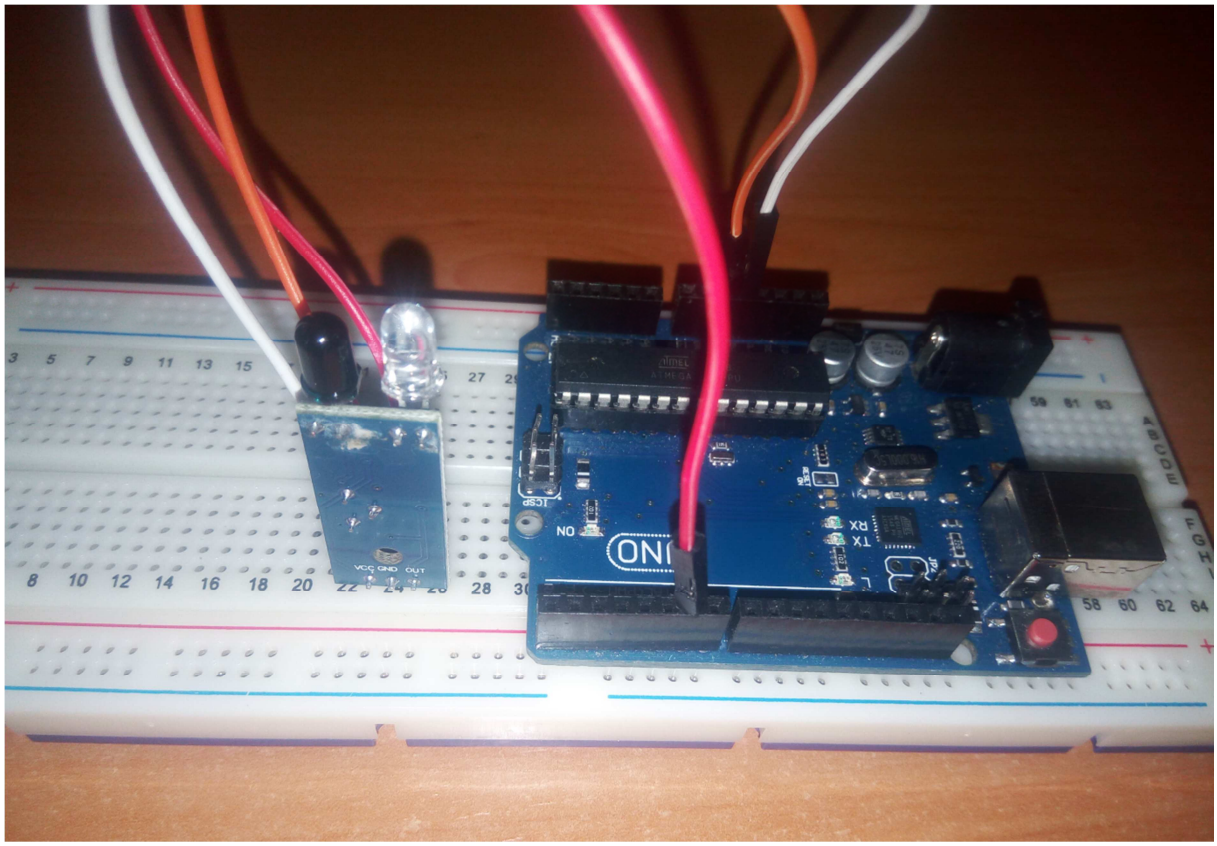


Figure III.2 : Bronchement de capteur infrarouge à la carte Arduino

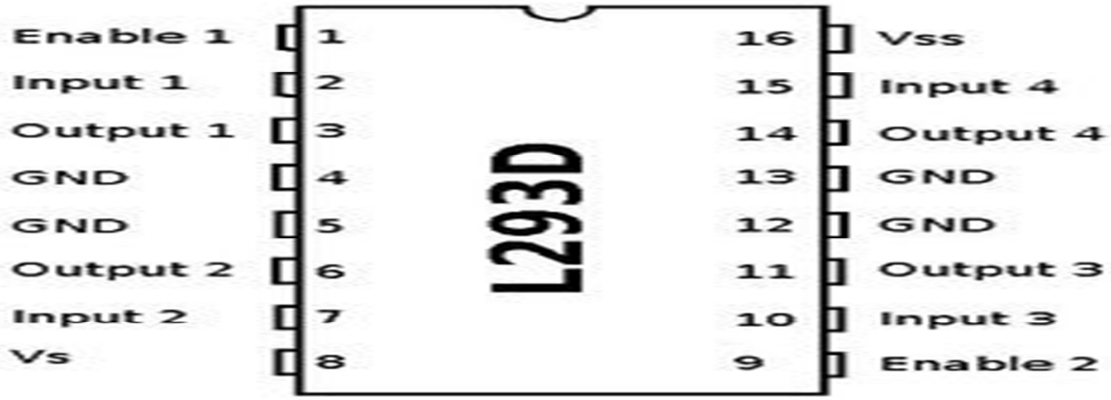
III.4 Branchement du Moteur DC a la carte Arduino

Pour que on puisse fonctionner un moteur a courant continue sur carte arduinoon doit utiliser le contrôleur des moteurs DC qui est le composant L293D.

III.5 Branchement du L293D

Avec un seul pont **L293D** et un **arduino** on va être capable de piloter 2 moteurs à courant continu indépendamment l'un de l'autre. Si la puissance de vos moteurs est faible vous pourrez même utiliser le 5V en sortie de votre **arduino** pour alimenter vos **moteurs DC**. [14]

Le schéma suivant détaille les différentes broches du composant L293D:



1. Enable1: permet d'envoyer (ou pas) la tension sur les sorties du moteur via OUTPUT1 et OUTPUT2 et commande l'activation/désactivation du premier Pont-H. Si ENABLE1 = GND, le pont-H est déconnecté et le moteur ne fonctionne pas. Si ENABLE1 = VSS, le pont-H est connecté aux sorties et le moteur fonctionne dans un sens ou l'autre ou pas en fonction des tensions appliquée sur INPUT1 & INPUT2.
2. Input1: avec Input 2, sont les broches de commande du Pont-H Output1/Output2. Il sera directement brancher à notreArduino pour commander le sens du courant entre Output 1 et Output 2.
3. Ouput1: avec Output 2, sera branché directement sur le moteur.
4. GND: qui doit être raccordé à la masse de la source d'alimentation de puissance VS et à la masse de la source d'alimentation de la logique "VSS" (donc GND Arduino).

5. Ouput2: avec Output 1, sera branché directement sur le moteur.
6. Input2: avec Input 1, sont les broches de commande du Pont-H Output1/Output2. Il sera directement brancher à notre
7. Arduino pour commander le sens du courant entre Output 1 et Output 2.
8. VS: Alimentation de puissance des moteurs.
9. Enable2: commande l'activation du second pont-H constitué de Output3/Output4
10. Input3: avec Input 4, sont les broches de commande du Pont-H Output3/Output4. Il sera directement brancher à notreArduino pour commander le sens du courant entre Output 3 et Output 4.
11. Ouput3: avec Output 4, sera branché directement sur le moteur.
12. GND
13. GND
14. Ouput4: avec Output 3, sera branché directement sur le moteur.
15. Input4: avec Input 3, sont les broches de commande du Pont-H Output3/Output4. Il sera directement brancher à notreArduino pour commander le sens du courant entre Output 3 et Output 4.
16. VSS: Alimentation de la logique de commande (5V). A raccorder à la borne +5V de notreArduino.

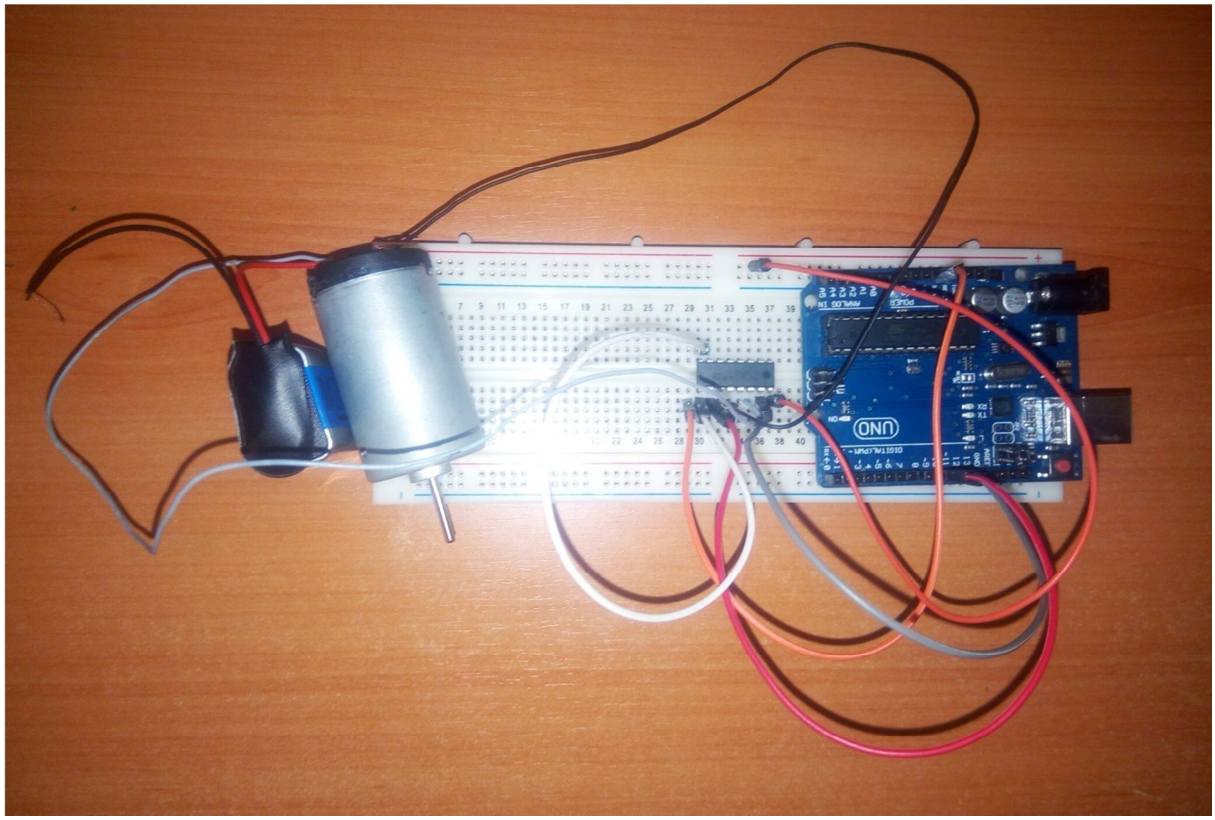


Figure III.3 : Branchement du moteur DC à la carte Arduino

III.6 La réalisation logicielle :

III.6.1 La programmation de l'Arduino (Ecrire, Compiler et Transférer) :

Une fois on a notre carte Arduino entre les mains et l'environnement de développement Arduino sur PC, On commence par lancer le logiciel arduino, puis on relie la carte à notre PC.

Figue III.4 : USB câble (2.0, type A / B) 50cm

III.6.2 Si La carte Arduino n'est pas reconnue !

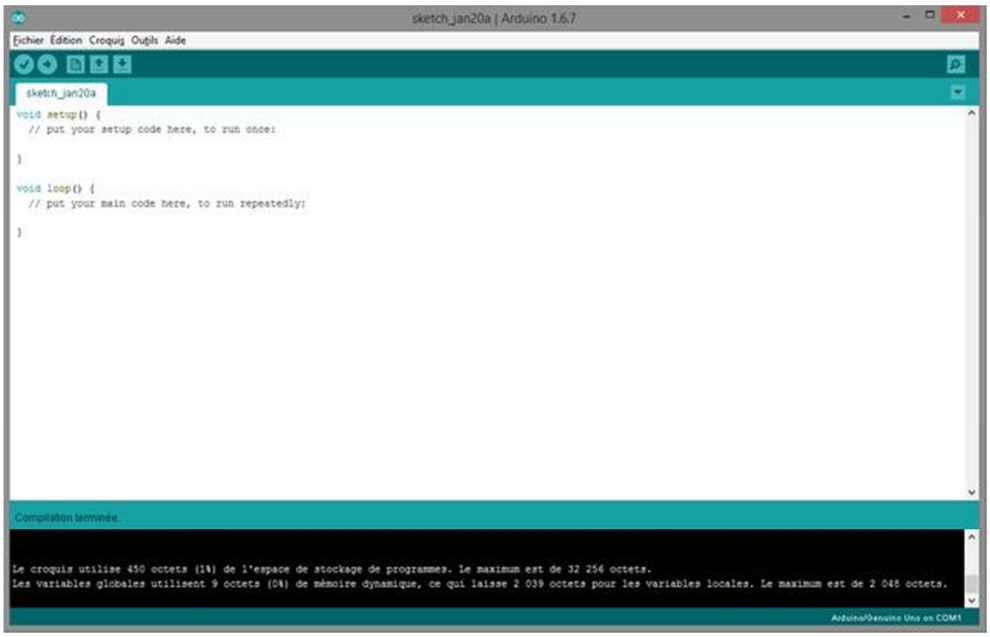
Une fois notre carte est reconnue on la sélectionne dans la liste du menu Outils / Ports et on vérifie que le bon type de carte est sélectionné dans le menu Outils / Types Cartes si tout est bon on peut continuer.

A l'ouverture d'un nouveau projet de programme Arduino on se retrouve devant ceci

```
1  void setup() {
2    // put your setup code here, to run once:
3
4  }
5
6  void loop() {
7    // put your main code here, to run repeatedly:
8
9  }
```

Il s'agit du strict minimum en terme de programme Arduino.

N.B. Tout ce qui se trouve à droite des // sont des commentaires. Ces commentaires sont ignorés lors de la compilation.



On clique sur le bouton "vérifier" et quelques secondes plus tard on devrait avoir confirmation dans la zone d'information que notre programme a bien été compilé. Nous viende compiler notre tout premier programme Arduino.

On peut maintenant cliquer sur le bouton téléverser pour transférer le programme sur notre carte Arduino.

Ce programme est une coquille vide avec le strict minimum pour faire plaisir au compilateur.

Un programme Arduino est toujours composé de deux fonctions : setup() et loop().

La fonction setup() est exécutée une seule fois lors du démarrage de la carte Arduino. On place typiquement dans le corps de cette fonction tout ce qui touche à l'initialisation du matériel (entrées, sorties, capteurs, etc.).

La fonction loop() est appelée en boucle tant que la carte Arduino est alimentée. C'est dans le corps de cette fonction que nous allons plus tard écrire notre programme.

III.6.3 Ajuster les préférences d'affichage et de compilation :

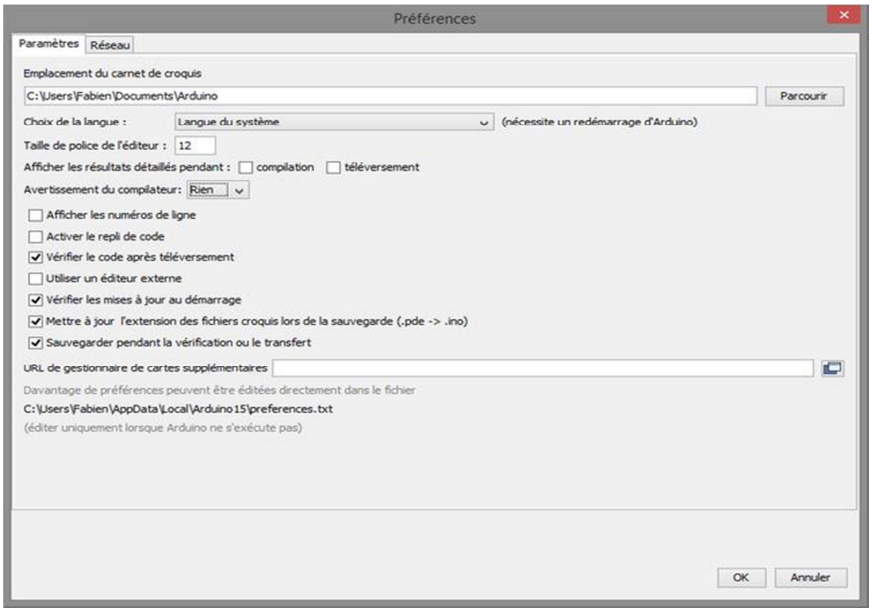


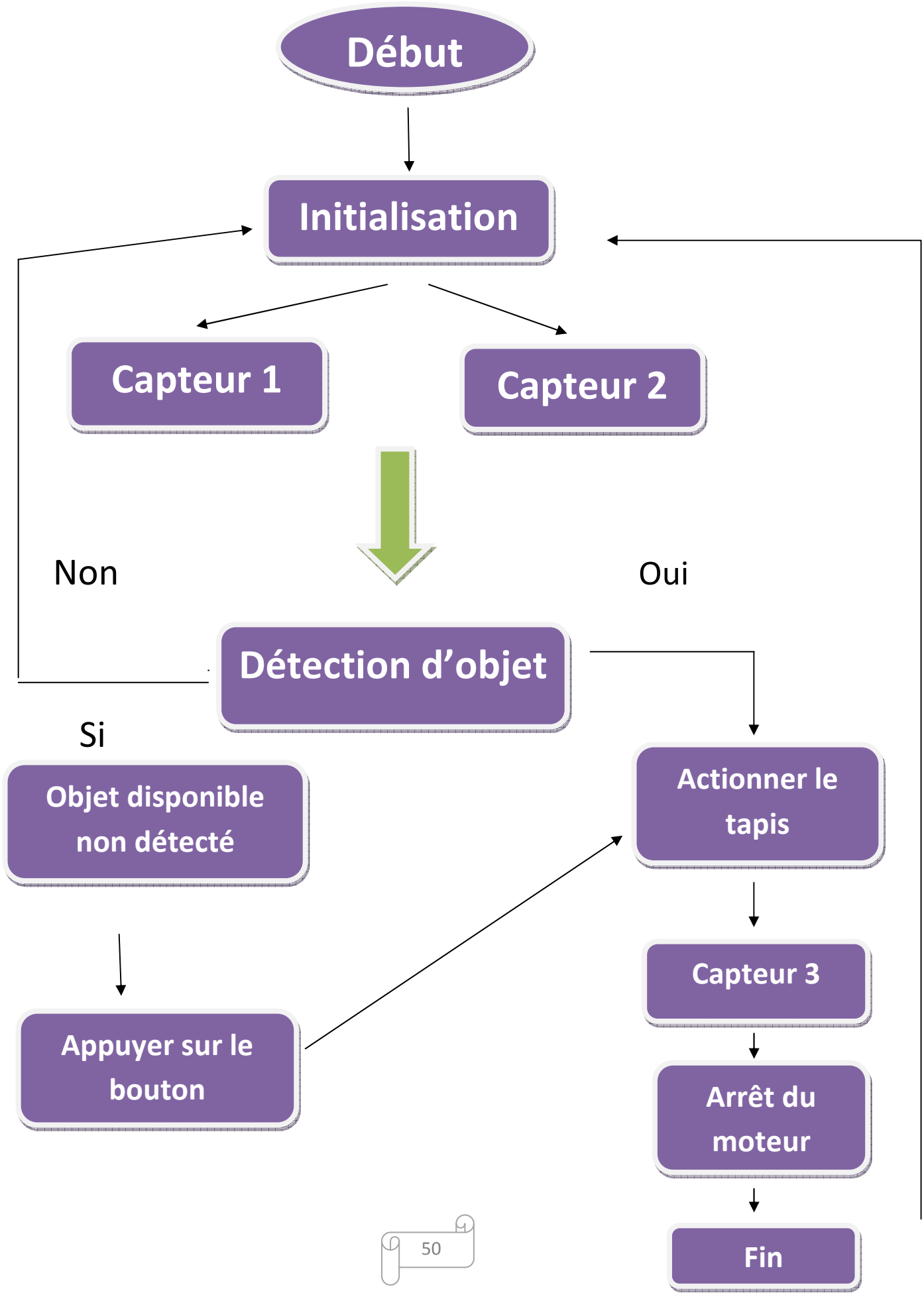
Figure III.5 Capture d'écran des préférences par défaut du logiciel Arduino

Pour modifier les préférences d'affichage et de compilation en passant par le menu Fichier / Préférences.

Il est préférable d'activer l'affichage des numéros de ligne, le repli de code et tous les avertissements du compilateur. Ainsi, onpeut détecter les erreurs de programmation rapidement au moment de la compilation, faire des modifications à certaines lignes sans chercher leurs numéros et replier les morceaux de code qui ne nous intéressent pas lors de l'écrire de notre programme.

Si ona une certaine expérience en programmation, il vaut mieux d'activer l'affichage détaillé lors de la compilation et du transfert du programme. Le logiciel Arduino cache certaines informations pour ne pas troubler les utilisateurs débutants, mais parfois, ces informations sont très utiles pour tordre le cou à un bug récalcitr

Organigramme fonctionnel



III.7 Discussion :

Au terme de ce chapitre, nous avons fait la réalisation pratique d'un tapis roulant automatisé commandé par un système de détection.

Cette tache est la mise en œuvre des différents composants a fin de réaliser ce tapis roulant automatique. Cela se fait à la base de la carte Arduino Uno, ainsi qu'un moyen de convoyé rage qui peut s'insérer facilement dans n'importe quel chaine de production.

L'objectif fixé au départ est, d'après nous, atteint. Cependant, notre application reste perfectible notamment en ce qui concerne le volet détection est plus précisément le choix de l'emplacement des dispositifs de détection.

CONCLUSION GENERALE

Conclusion générale :

Notre projet consiste à réaliser et commander un tapis roulant qui est dispositif permettant de déplacer des charges à l'horizontale ou sur des plans légèrement inclinés.

En premier on a donné un aperçu général sur la carte d'acquisition Arduino Uno que nous avons utilisé. Nous avons ensuite enchainé avec une description des composants annexes que nous avons utilisés (moteur dc et capteur infrarouge).

Dans ce projet, on a pu réaliser avec succès un tapis roulant automatique à base d'une carte Arduino Uno, et un circuit intégré L293D le locomoteur de notre moteur à courant continu.

Notre application est constitué de trois capteurs infrarouges de détection d'obstacle, le premier sert à détecter la présence de l'objet sur le début de tapis roulant en suite donner l'ordre au moteur à courant continu de démarrer, le deuxième est placé au milieu de tapis fait en sort pour détecter l'objet dans le cas où il s'est mal placé sur le début et n'est pas détecté par le premier capteur, et le troisième capteur fait pour arrêter le moteur DC lorsque l'objet est arrivé a la fin du tapis roulant en donnant une impulsion au moteur, et en suite il s'arrête

Il est possible d'améliorer notre tapis roulant comme en envisageant une commande à distance en utilisant un Lily pad ou un Smartphone.

[1] http://www.generationrobots.com/fr/152-arduino. consulter le: mai 2018.

[2] S.V.D.Reyvanth, G.Shirish, « PID controller using Arduino ».

[3] C. Tavernier, « Arduino applications avancées ». Version Dunod.

[4] http://www.acm.uiuc.edu/sigbot/tutorials/ 2009-11-17-arduino-basics. consulter le: mai 2018.

[5] Jean- Noël, « livret Arduino en français » , centre de ressources art sensitif .

[6] Simon Landrault& Hippolyte Weisslinger « arduino :premier pas en informatique embarquée».

[7] Le grande livre d'arduino Erik bratmann«2émé Edition».

[8] F. Baudoin, M. Lavabre, Capteurs : principes et utilisations, Éd. Casteilla, 2007 (ISBN 978-2-7135-2749-4).

[9] Georges Asch, Les capteurs en instrumentation industrielle, Dunod (ISBN 2100057774)

[10] https://www.suva.ch/materiel/fiche-thematique/rayonnement-infrarouge

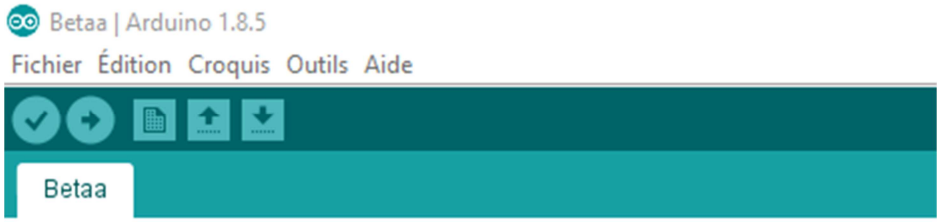
[11] Charles Harel, Machines électriques et essais de Machines, Société française des électriciens – École supérieure d'électricité, Paris, 1960.

[12] nagashur.com/blog/.../utilisation-dun-circuit-l293d-pour-commander-des-moteurs/

[13] www.manuel-esteban.com/tuto-lire-un-capteur-infrarouge-avec-arduino

[14] https://www.zem.fr/arduino-controler-des-moteurs-dc-avec-le-composant-l293d

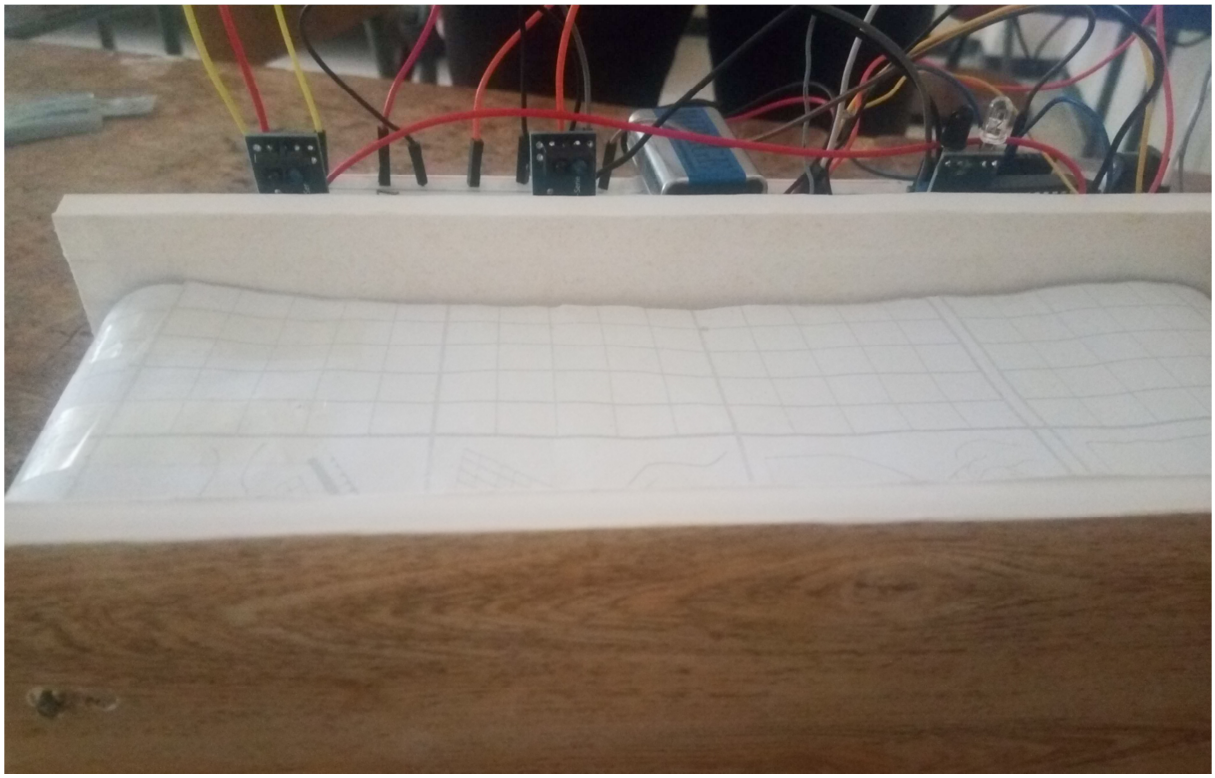
Notre Programme

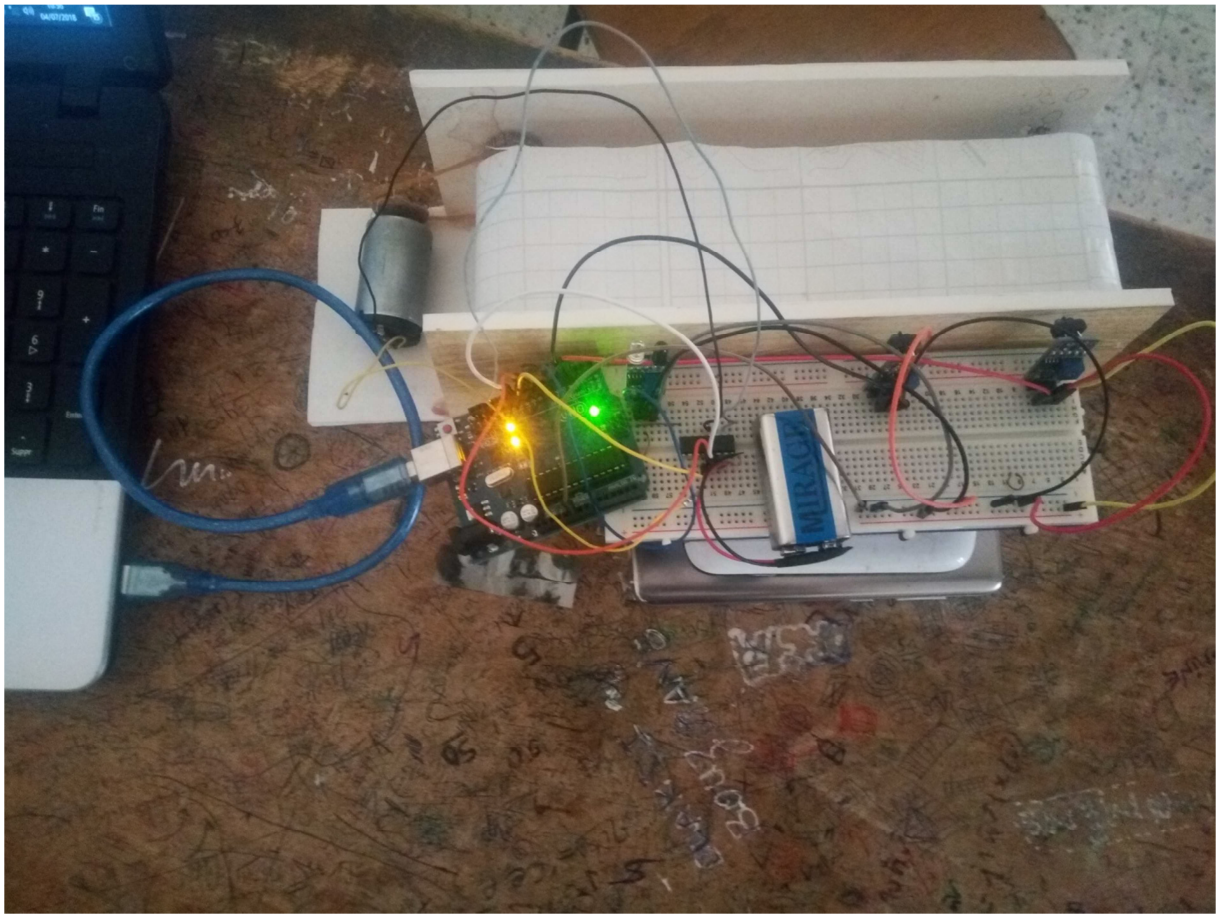



```
int pin1=6;
int pin2=7;
int pir=5;
int pir2=8;
int pir3=9;
boolean stat1=0;
boolean stat2=0;
boolean stat3=0;
byte a,b,c;
void setup() {
  // put your setup code here, to run once:
pinMode(pin1,OUTPUT);
pinMode(pin2,OUTPUT);
pinMode(pir,INPUT);
pinMode(pir2,INPUT);
pinMode(pir3,INPUT);
Serial.begin(9600);
a=0;
b=0;
c=0;
}

void loop() {
  stat1=digitalRead(pir);
  stat2=digitalRead(pir2);
  stat3=digitalRead(pir3);
  Serial.println(stat2);
```

```
//
//  if (stat1==0){
//  delay(140);
//  a=!a;
//
//  }
//  if(stat2==0) {
//    delay(140);
//    b=!b;
//  }
//  if(stat3==0) {
//    delay(140);
//    c=!c;
//  }

  while(a==1 || b==1&&c!=0) {
    marche();
     if (c==0 || stat3==0) {
      arret();
      break;
     }
     arret();
  }
  // put your main code here, to run repeatedly:

}

void marche() {
  digitalWrite(pin1,HIGH);
  digitalWrite(pin2,LOW);

}
void arret() {
  digitalWrite(pin1,LOW);
  digitalWrite(pin2,LOW);
}
```

ANNEX B





ANNEX B