



Ecole Nationale Supérieure d'Informatique  
et d'Analyse des Systèmes



جامعة محمد الخامس بالرباط  
Université Mohammed V de Rabat

# *Etude et réalisation d'un robot suiveur de ligne et détecteur d'obstacles*

---

Réalisé par : *Mr. Keddouch Larbi*

Encadré par : *Mme. TAMO NASSER*

Année Universitaire 2016-2017

## Remerciement

*Au terme de ce travail, nous voudrions nos vifs remerciements à nos professeurs de l'Ecole Nationale supérieure d'informatique et d'analyse des systèmes de RABAT qui nous ont fournis une formation riche et qui ont toujours été pour nous un exemple et une source de motivation et d'encouragement.*

*Nous tenons plus particulièrement à remercier chaleureusement Mme Tamo Nasser, l'encadreur technique et le tuteur pédagogique, pour l'aide précieuse qu'il nous a apportée tout au long de notre démarche, pour sa disponibilité et ses précieux conseils. Il a su nous guider avec compétence et patience tout au long de notre projet. Aussi, nous avons tiré pleinement profit de sa vaste expérience dans le domaine.*

*Nos remerciements s'adressent également à tous les membres de jurys qui ont accepté pour l'honneur qu'ils font d'avoir assister à notre soutenance.*

*Enfin, nous remercions tous ceux qui nous ont, de près ou de loin, aidé à réaliser notre projet.*

# Sommaire

<b>Remerciements</b>	2
<b>Liste des figures</b>	4
<b>Introduction générale</b>	5
<b>Chapitre 1 : Robotique et Arduino</b>	6
<b>Introduction</b>	7
1-Les robots	7
1-1 Historique	8
1-2 Définition d'un robot	9
1-3 Types de robot	9
1-4 Fonctionnement des robots évolués	9
1-5 Robotique industriels	9
1-6 Les générations des robots	9
2-Cartes Arduino	7
2-1 Historique	9
2-2 Définition d'un arduino	9
2-3 Les cartes Arduino	9
<b>Conclusion</b>	9
<b>Chapitre 2 : Analyse conception</b>	11
<b>Introduction</b>	12
1- Principe de fonctionnement de robot	13
2-Architecture de la carte de commande	14
2-1 Architecture interne du module Arduino	14
3-Choix des actionneurs	14
3-1 Les servomoteurs	14
3-2 Moteurs à courant continu	14
3-3 Moteur pas à pas	14
4-Architecture interne de carte de puissance	14
5-Capteur infrarouge émetteur récepteur	14
<b>Conclusion</b>	15
<b>Chapitre 3 : Outils de développement et réalisation du robot</b>	18
<b>Introduction</b>	19
1-Conception mécanique	19
2-Réalisation des cartes électroniques et simulation	19
2-1 Présentation d'ISIS	14
2-2 Carte de puissance	14
2-3 Carte de commande	14
3-Programmation	19
<b>Conclusion</b>	26
<b>Conclusion et perspectives</b>	32
<b>Liens externes &amp; bibliographie</b>	33

## Liste des figures

<u>Figure I.1 : Différents types des capteurs</u> .....	7
<u>Figure I.2 : Différents types d'actionneurs</u> .....	9
<u>Figure I.3 : Arduino Uno</u> .....	10
<u>Figure I.4 : Arduino MEGA</u> .....	11
<u>Figure II.1 : Expression fonctionnelle de besoin</u> .....	14
<u>Figure II.2 : Schéma synoptique du robot suiveur de ligne</u> .....	15
<u>Figure II.3 : Brochage de la carte Arduino Uno</u> .....	19
<u>Figure II.4 : Servomoteur HS-311</u> .....	22
<u>Figure II.5 : Servomoteur commandé pour PWM</u> .....	24
<u>Figure II.6 : Moteur à courant continu</u> .....	26
<u>Figure II.7 : Moteur pas à pas</u> .....	27
<u>Figure II.8 : Schéma de la carte de puissance</u> .....	27
<u>Figure II.9 : Capteur infrarouge QRD1114</u> .....	28
<u>Figure III.1 : Conception mécanique du robot</u> .....	28
<u>Figure III.2 : Schéma de simulation de la carte de puissance</u> .....	29
<u>Figure III.3 : Carte de puissance réelle</u> .....	30
<u>Figure III.4 : Brochage de capteur avec Arduino</u> .....	30
<u>Figure III.5 : Carte des capteurs</u> .....	31
<u>Figure III.6 : Brochage du servomoteur avec Arduino</u> .....	30
<u>Figure III.7 : Organigramme</u> .....	30
<u>Figure III.8 : Logiciel Arduino</u> .....	30

## Introduction général

Notre travail de projet fin d'étude porte sur : étude et réalisation d'un robot suiveur de ligne et détecteur d'obstacles d'une façon autonome, dont l'approche de commande proposée est basée essentiellement sur le module Arduino.

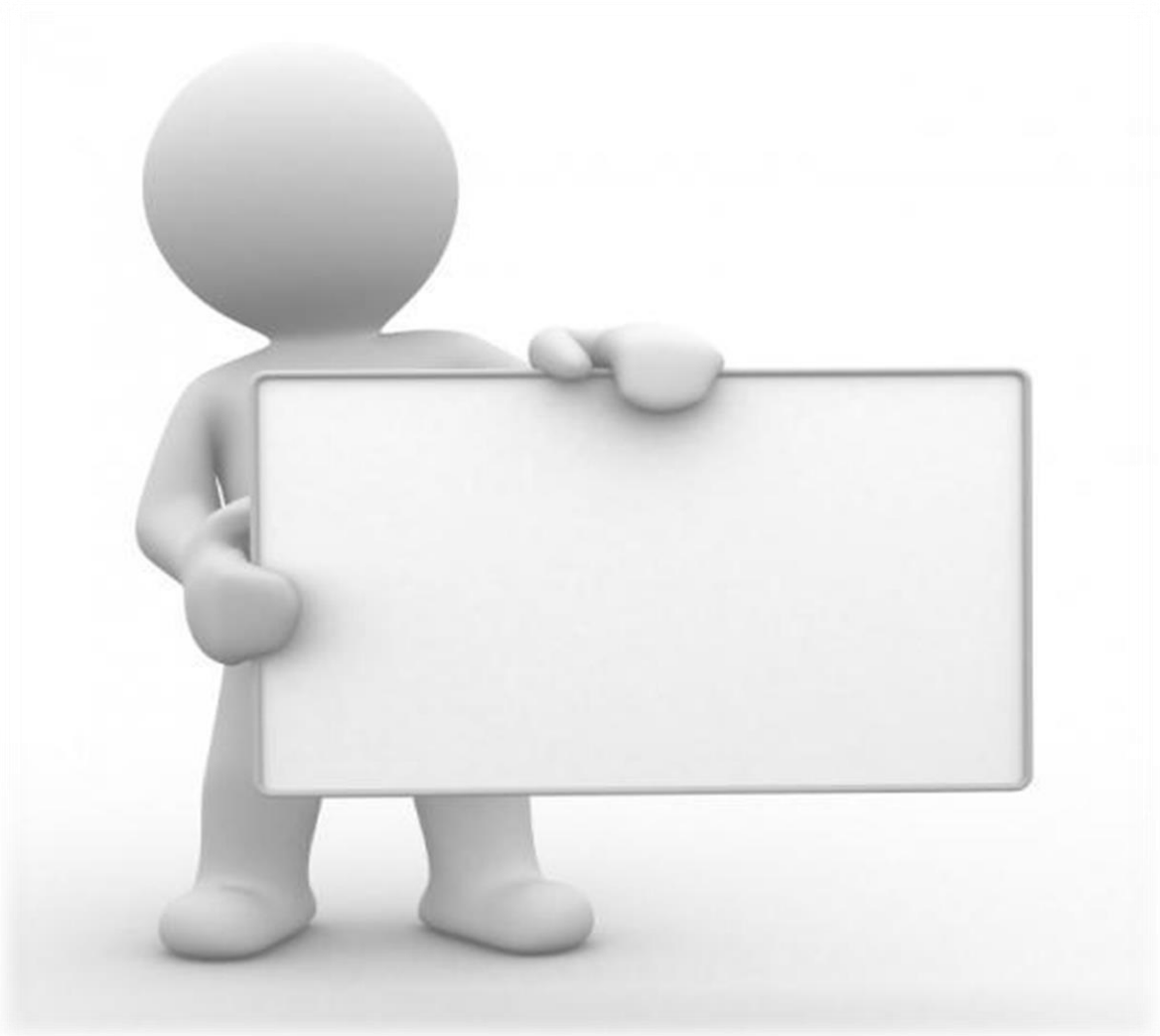
Pour aborder ce thème, on a eu recours à examiner en premier lieu le domaine de la robotique mobile. En deuxième lieu, une étude théorique détaillé portant sur les différentes cartes électroniques pour un exemple de robot mobile ; architecturé autour du module Arduino, est présentée et discutée.

Pour la réalisation et la simulation des cartes respectivement, de commande, de puissance et de capteurs, nous avons utilisé un simulateur de conception de carte électronique le logiciel ISIS. L'embarquement du programme, permettant le pilotage du robot, sur le module Arduino principalement le microcontrôleur ATmega328 est obtenu pour un compilateur de programmation spécifique à l'Arduino.

# *Chapitre 1*

## *Robotique et Arduino*

Ce chapitre présentera une description générale de notre sujet « conception et réalisation d'un robot suiveur de ligne et détecteur d'obstacle » puis présentera notre objectif et notre démarche.



## Introduction

Le travail élaboré dans ce premier chapitre, est scindé en deux parties. Dans la première, nous présentons une étude bibliographies sur la robotique. Une étude détaillée sur Arduino, module très utilisé dans domaine de la robotique, est consacré dans la deuxième partie.

## 1. Les robots

### 1.1. Historique

La robotique est issue des travaux effectués sur les automates, ancêtres des robots. Parmi les premiers constructeurs d'automates, on peut citer Léonard de Vinci, ou encore le mécanicien français Jacques de Vaucanson, dont les tentatives de reproduction des fonctions vitales des êtres humains (circulation, respiration et digestion) au moyen d'automates.

Le terme « robot » est introduit en 1920 par l'écrivain tchèque Karel Čapek dans sa pièce de théâtre RUR (Rossum's Universal Robots). Ce terme, provenant du tchèque robot, « travail forcé », désigne à l'origine une machine androïde capable de remplacer l'homme dans toutes ses tâches. Au XIXe siècle apparaissent les premières machines-outils, qui par leurs fonctions peuvent être considérées comme les précurseurs des robots. Dans les années 1940 et 1950, les progrès de l'électronique permettent de miniaturiser les circuits électriques (inventions du transistor et du circuit intégré), ouvrant ainsi de nouvelles voies à la fabrication de robots. Dans les premiers temps de la robotique, le robot est considéré comme une imitation de l'homme, aussi bien fonctionnelle que physique. Aujourd'hui, les constructeurs ne tentent plus de reproduire l'aspect humain sur un robot, privilégiant avant tout sa fonctionnalité.

Les robots sont actuellement très répandus dans l'industrie, en particulier en construction automobile, et chez la plupart des fabricants d'ordinateurs. Capables d'effectuer rapidement des travaux répétitifs, ils sont notamment utilisés dans les chaînes de fabrication et de montage. On les emploie également dans des environnements difficilement supportables par l'homme (conditions extrêmes de température ou de pression, radioactivité

élevée, etc.). L'industrie du nucléaire a ainsi largement contribué au développement de la robotique (notamment dans la conception de bras télémanipulateurs).

## 1.2 Définition d'un robot

Machine, automate à l'aspect humain capable d'agir comme un être humain. Appareil effectuant, grâce à un système de commande automatique à base de micro-processeur, une tâche précise, pour laquelle il a été conçu dans le domaine industriel, scientifique ou domestique.

### 1.2.1. Utilité d'un robot

Un robot peut servir à beaucoup de chose, nettoyer une maison, tondre une pelouse, assister les médecins lors de leurs opérations, faire des taches répétitives ou dangereuse dans les entreprises. Il peut aussi être un outil pour l'armée (exemple les robots sous marins que ECA développe).

### 1.2.2. Composition d'un robot

Un robot est composé de plusieurs parties, capteurs, actionneurs et une unité de traitement, le tout lié sur une base mécanique.

- **Capteur**

Il existe de nombreux types de capteurs, du simple interrupteur, au capteur de distance en passant par les capteurs de température. Ils permettent à votre robot d'avoir des informations sur l'extérieur.



Figure I.1 : Différents types des capteurs



- **Les actionneurs**

Les actionneurs permettent a votre robot de réagir suivant se que vous lui avez demandé. Il existe aussi beaucoup d'actionneurs principalement réalisé à base de moteur électrique.



Figure I.2 : Différents types d'actionneurs

- **Unités de traitement**

L'unité de traitement permet de traiter les données venant des capteurs, suivant les données, l'unité de traitement commande les actionneurs. Exemple : pour un suiveur de ligne, on utilise des capteurs de lignes, si on est en présence d'un virage, l'unité de traitement doit décider d'arrêter ou ralentir un moteur pour tourner.

### 1.2.3. Définition d'un robot mobile

Un robot mobile est une machine automatique capable de se mouvoir dans un environnement donné. On regroupe, sous cette appellation tous les robots autonomes (c'est à dire non télécommandés) capables de se déplacer, par opposition aux robots attachés à un point fixe, comme les robots manipulateurs en industrie. Il existe plusieurs types de robots mobiles, et ceux-ci sont en général, classifiés selon leur type de locomotion (c'est à dire le milieu dans lequel ils évoluent, ainsi que leur mode de propulsion). Les robots mobiles évolueront donc sur terre, dans les airs ou encore dans l'eau. Les robots terrestres sont, par exemple, actionnés par des roues, des chenilles ou encore des pattes.

### 1.3. Types de robots

On peut classer les robots en quatre catégories, par ordre de complexité croissante. Les robots les plus simples, et les plus courants se contentent de répéter les opérations inscrites dans leurs programmes. Les robots appartenant au deuxième type, sont capables de

reproduire certains mouvements humains enregistrés sur bande magnétique, tandis que les robots de troisième génération sont des dispositifs à commande numérique, dont les mouvements sont enregistrés sur une unité de stockage, et qui peuvent accomplir plusieurs opérations différentes. Enfin, les robots évolués sont plus complexes, et faisant appel à des capteurs d'environnement et aux techniques d'intelligence artificielle, sont dotés d'une grande capacité de traitement de l'information. Certains d'entre eux peuvent, ainsi reconnaître la forme d'objets sur un écran de télévision, ou encore réagir à des paroles employant un vocabulaire spécifique (voir reconnaissance vocale).

### 1.4. Fonctionnement des robots évolués

On peut définir un robot évolué, comme une machine agissant physiquement sur son environnement, en vue d'atteindre un but qui lui a été fixé. Cette machine doit être capable de percevoir son environnement, et le cas échéant de s'adapter à certaines variations de celui-ci (capacité d'autonomie). Par exemple, si le but du robot est de se déplacer d'un point à un autre dans une pièce, il doit être capable de contourner les obstacles qui y sont présents. Il lui faut donc assurer des fonctions de perception, de décision et d'action. Pour ce faire, un robot est un mélange de mécanique (bras articulé, système hydraulique, pneumatique et électrique, système de déplacement...), d'électronique (capteurs en tout genre, caméras...), d'informatique (micro-ordinateurs, microprocesseurs) et de logiciels. Les robots les plus anciens contiennent beaucoup de mécanique et d'électronique et peu d'informatique et de logiciels. La part de l'informatique tend à augmenter, car c'est elle qui influe sur les capacités de perception et de décision. Ainsi, le cœur du robot peut être vu comme une boucle perception-décision-action.

#### 1.4.1. Perception

La perception repose sur un ensemble de capteurs plus ou moins sophistiqués. Les robots sont de plus en plus équipés de caméras CCD (Charge Couple Device), qui font office d'interfaces visuelles. Ces capteurs fournissent un ensemble de mesures de bas niveau, que le robot va devoir être capable d'interpréter à plus haut niveau pour préparer la prise de décision. L'interprétation est très complexe, et met en jeu des outils mathématiques et

informatiques de haut niveau (c'est le cas par exemple de la vision par ordinateur). La qualité de perception détermine, ainsi les possibilités d'évolution d'un robot.

#### 1.4.2. Décision

Le processus de décision d'un robot prend en compte d'une part, le but qu'il doit atteindre et d'autre part, la perception de son environnement. Cette tâche est directement dépendante de la qualité de la perception. Les modèles de décision peuvent être très complexes, et sont fondés sur des outils mathématiques et informatiques très évolués, souvent issus de l'intelligence artificielle. La difficulté majeure de cette opération vient du nombre de l'ensemble des décisions possibles, qui peut être très grand, et empêche une énumération complète. La prise de décision s'appuie ainsi sur des heuristiques, qui permettent de déterminer les choix les plus efficaces en fonction de l'objectif visé.

#### 1.4.3. Action

L'action consiste à mettre en application la décision prise. Cela peut être très simple ou complexe en fonction du domaine considéré. Cette fonction reste très liée à la mécanique et à l'électronique, et beaucoup moins à l'informatique.

### 1.5. Robotique Industriels

Le marché de la robotique est aujourd'hui encore dominé par les robots industriels, qui sont souvent très simples. Selon la commission économique des Nations unies pour l'Europe (CEE-ONU) et la Fédération internationale de robotique (IFR), il y a, en ce début de XX<sup>e</sup> siècle, plus de 800 000 robots industriels dans le monde, dont près de la moitié est détenue par le Japon. Le reste du parc mondial de robots industriels se répartit principalement entre les États-Unis (100 000), l'Allemagne (100 000), l'Italie (45 000), la république de Corée (40 000), la France (25 000) et le Royaume-Uni (15 000). L'évolution est globalement à la hausse majoritairement aux États-Unis et en Europe.

La répartition est également différente selon le secteur industriel, les industries automobile et chimique étant les principales utilisatrices de robots. En moyenne au Japon, il y a 270 robots pour 10 000 salariés du secteur industriel (contre 130 en Allemagne, 120 en république de Corée, 60 en France, 50 aux États-Unis). Dans le secteur automobile japonais,

on compte 1 robot pour 6 salariés. Une évolution est attendue dans le monde des services (robot aspirateur ou robot tondeuse), mais elle tarde à se confirmer.

## 1.6. Les générations des robots

Des progressions s'opèrent dans tous les domaines :

- Mécanique,
- Micro-informatique,
- Energétique,
- Capteurs / actionneurs.

A l'heure actuelle, on peut distinguer 3 générations de robots :

### ❖ Le robot est passif

Il est capable d'exécuter une tâche qui peut être complexe, mais de manière répétitive, il ne doit pas y avoir de modifications intempestives de l'environnement. L'auto-adaptativité est très faible. De nombreux robots sont encore de cette génération.

### ❖ Le robot devient actif

Il devient capable d'avoir une image de son environnement, et donc de choisir le bon comportement (sachant que les différentes configurations ont été prévues). Le robot peut se calibrer tout seul.

### ❖ Le robot devient « intelligent »

Le robot est capable d'établir des stratégies, ce qui fait appel à des capteurs sophistiqués, et souvent à l'intelligence artificielle.

## 2. Cartes Arduino

### 2.1. Historique

Le projet Arduino est issu d'une équipe d'enseignants et d'étudiants de l'école de Design d'Interaction d'Ivrea (Italie). Ils rencontraient un problème majeur à cette période (avant 2003 - 2004) : les outils nécessaires à la création de projets d'interactivité étaient complexes et onéreux (entre 80 et 100 euros). Ces coûts souvent trop élevés rendaient difficiles le développement par les étudiants de nombreux projets et ceci ralentissait la mise en œuvre concrète de leur apprentissage.

Jusqu'alors, les outils de prototypage étaient principalement dédiés à l'ingénierie, la robotique et aux domaines techniques. Ils sont puissants mais leurs processus de développement sont longs, et ils sont difficiles à apprendre et à utiliser pour les artistes, les designers d'interactions et, plus généralement, pour les débutants.

Leur préoccupation se concentra alors, sur la réalisation d'un matériel moins cher et plus facile à utiliser. Ils souhaitaient créer un environnement proche de Processing, ce langage de programmation développé dès 2001 par Casey Reas et Ben Fry, deux anciens étudiants de John Maeda au M.I.T., lui-même initiateur du projet DBN.

En 2003, Hernando Barragan, pour sa thèse de fin d'études, avait entrepris le développement d'une carte électronique dénommée Wiring, accompagnée d'un environnement de programmation libre et ouvert. Pour ce travail, Hernando Barragan réutilisait les sources du projet Processing. Basée sur un langage de programmation facile d'accès, et adaptée aux développements de projets de designers, la carte Wiring a donc inspiré le projet Arduino (2005).

Comme pour Wiring, l'objectif était d'arriver à un dispositif simple à utiliser, dont les coûts seraient peu élevés, les codes et les plans « libres » (c'est-à-dire dont les sources sont

ouvertes et peuvent être modifiées, améliorées, distribuées par les utilisateurs eux-mêmes) et, enfin, « multi-plates-formes » (indépendant du système d'exploitation utilisé).

Conçu par une équipe de professeurs et d'étudiants (David Mellis, Tom Igoe, Gianluca Martino, David Cuartielles, Massimo Banzi ainsi que Nicholas Zambetti), l'environnement Arduino est particulièrement adapté à la production artistique, ainsi qu'au développement de conceptions, qui peuvent trouver leurs réalisations dans la production industrielle.

Le nom Arduino trouve son origine dans le nom du bar dans lequel l'équipe avait l'habitude de se retrouver. Arduino est aussi le nom d'un roi italien, personnage historique de la ville « Arduin d'Ivrée », ou encore un prénom italien masculin qui signifie « l'ami fort ».

## 2.2. Définition d'un Arduino

Le système Arduino est un outil pour fabriquer de petits ordinateurs qui peuvent capter et contrôler davantage de choses du monde matériel, que votre ordinateur de bureau. C'est une plateforme open-source d'électronique programmée, qui est basée sur une simple carte à microcontrôleur (de la famille AVR), et un logiciel, véritable environnement de développement intégré, pour écrire, compiler et transférer le programme vers la carte à microcontrôleur. Arduino peut être utilisé, pour développer des objets interactifs, pouvant recevoir des entrées d'une grande variété d'interrupteurs ou de capteurs, et pouvant contrôler une grande variété de lumières, moteurs ou toutes autres sorties matérielles. Les projets Arduino peuvent être autonomes, ou bien ils peuvent communiquer, avec des logiciels tournant sur votre ordinateur (tels que Flash, [Processing](#) ou MaxMSP). Les cartes électroniques peuvent être fabriquées manuellement, ou bien être achetées pré-assemblées.

## 2.3. Les cartes Arduino

### 2.3.1. La carte Arduino UNO

C'est la révision la plus récente de la carte de base Arduino USB. Elle se connecte sur l'ordinateur, avec un câble USB standard, et contient tout ce dont vous avez besoin pour, programmer et utiliser la carte. Elle peut être complétée avec toute une variété de circuits :



des cartes-femelles personnalisées avec des caractéristiques spécifiques. Elle est similaire à la carte précédente [Duemilanove](#), mais elle dispose d'un circuit intégré USB-vers-série différent, l'ATMega8U2, ainsi qu'un nouveau design du texte de la carte afin de rendre les entrées et les sorties plus faciles à identifier.

"Uno" signifie un en Italien, et ce nom marque la venue prochaine de la version 1.0 du logiciel Arduino. La carte UNO et la version 1.0 du logiciel seront la référence des versions Arduino à venir. La carte Uno est la dernière d'une série de carte USB Arduino, et le modèle de référence des plateformes Arduino; pour une comparaison avec les versions précédentes.

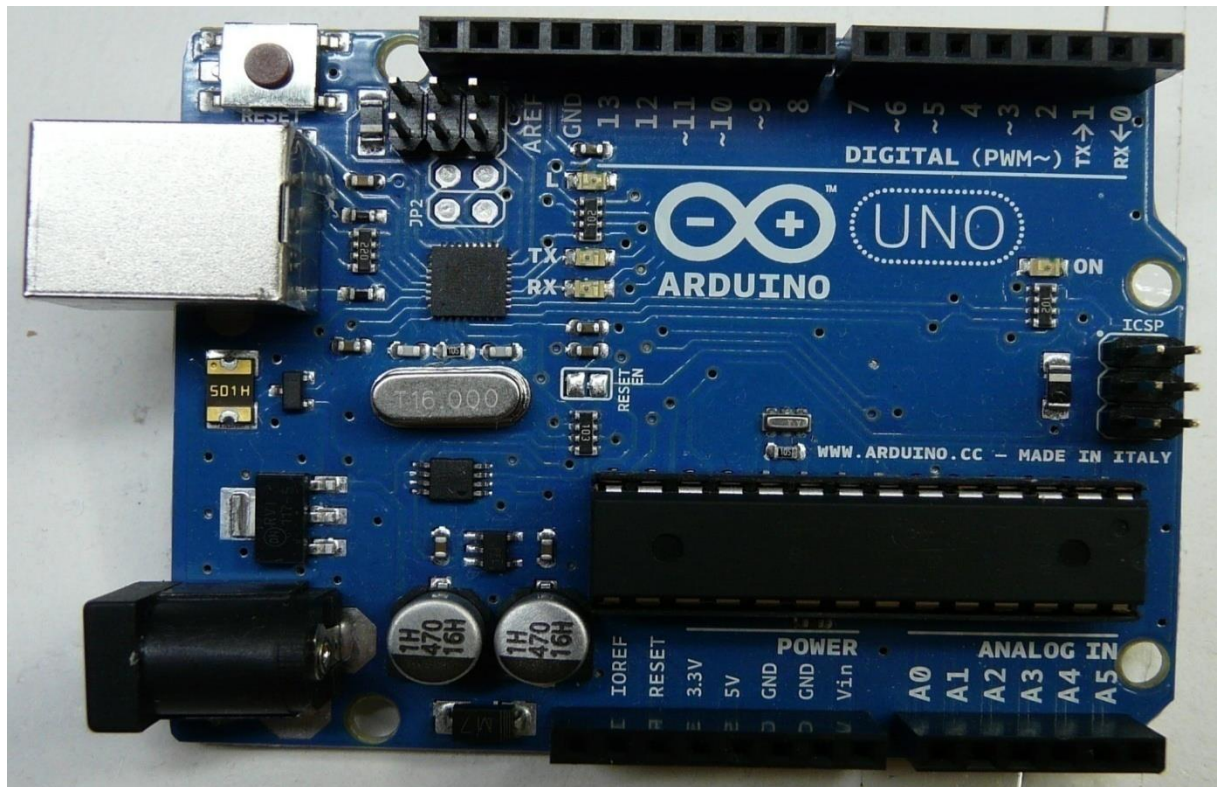


Figure I.3 : Arduino Uno

### 2.3.2. La carte Arduino Mega 2560

La version de la [Mega](#) sortie en même temps que la carte UNO. Cette version est basée sur l'ATmega2560, qui dispose de deux fois plus de mémoire, et utilise également l'ATMega 8U2 pour la communication USB-vers-série. Elle contient tout ce qui est nécessaire pour le fonctionnement du microcontrôleur; Pour pouvoir l'utiliser et se lancer, il

suffit simplement de la connecter à un ordinateur à l'aide d'un câble USB (ou de l'alimenter avec un adaptateur secteur ou une pile, mais ceci n'est pas indispensable, l'alimentation étant fournie par le port USB). L'ATmega 2560 à 256Ko de mémoire FLASH pour stocker le programme (dont 8Ko également utilisés par le bootloader). L'ATmega 2560 a également 8 ko de mémoire SRAM (volatile) et 4Ko d'EEPROM (non volatile - mémoire qui peut être lue à l'aide de la [librairie EEPROM](#)).

Le bootloader est un programme préprogrammé une fois pour toute dans l'ATméga et qui permet la communication entre l'ATmega et le logiciel Arduino via le port USB, notamment lors de chaque programmation de la carte.

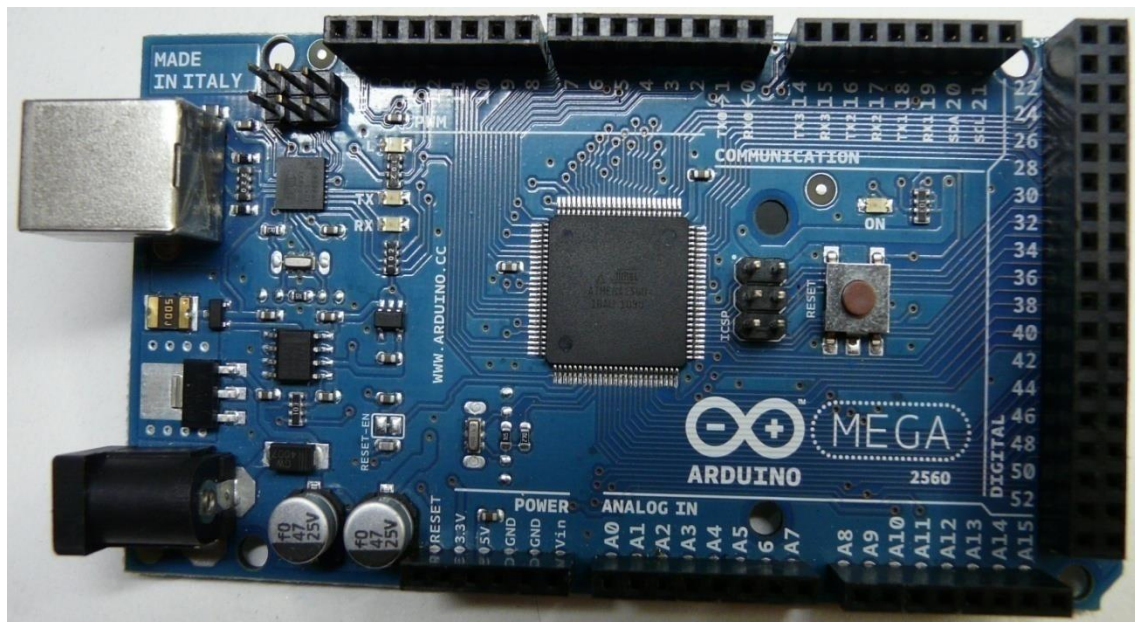


Figure I.4 : Arduino MEGA

### 2.3.3. Les autres cartes Arduino

Il existe plusieurs types d'Arduino, on cite à titre d'exemple :

- La carte Arduino Bluetooth
- La carte ArduinoMini
- La carte Arduino Nano
- La carte Arduino LilyPad



- La carte Arduino Fio
- La carte Arduino Pro
- La carte Arduino ProMini

## Conclusion

La conception d'un robot mobile est de nature multidisciplinaire et nécessite, souvent, beaucoup d'imagination et de créativité. Cette nature multidisciplinaire s'observe dans l'exploitation de l'électronique, les actionneurs et la conversion de l'énergie, la commande en temps réel, l'acquisition et la transmission d'images et du son, la programmation. Les télécommunications, sans oublier la conception mécanique du châssis et de ses composants. Dans cet esprit, nous consacrons le prochain chapitre à l'étude détachée nécessaire pour la conception et la mise en œuvre d'un exemple typique de robot mobile, dont la commande est architecturée au tour du module Arduino Uno.

## *Chapitre 2*

# *Analyse et Conception*

Dans ce chapitre nous avons présenté la partie conception et modélisation de notre robot. En particulier, nous allons rappeler dans un premier temps la méthode de conception utilisée à savoir l'analyse fonctionnelle. Et dans un deuxième temps, nous présenterons les modèles de conception de notre application.



## Introduction

On s'intéresse à la conception, et la commande d'un robot suiveur de ligne à base du module Arduino Uno, équipé de 4 capteurs infrarouge, une pile pour l'alimentation et des servomoteurs à rotation continu pour l'entraînement en mouvement du robot. La figure II.1 présente l'expression fonctionnelle de besoin, pour la mise en œuvre du robot.

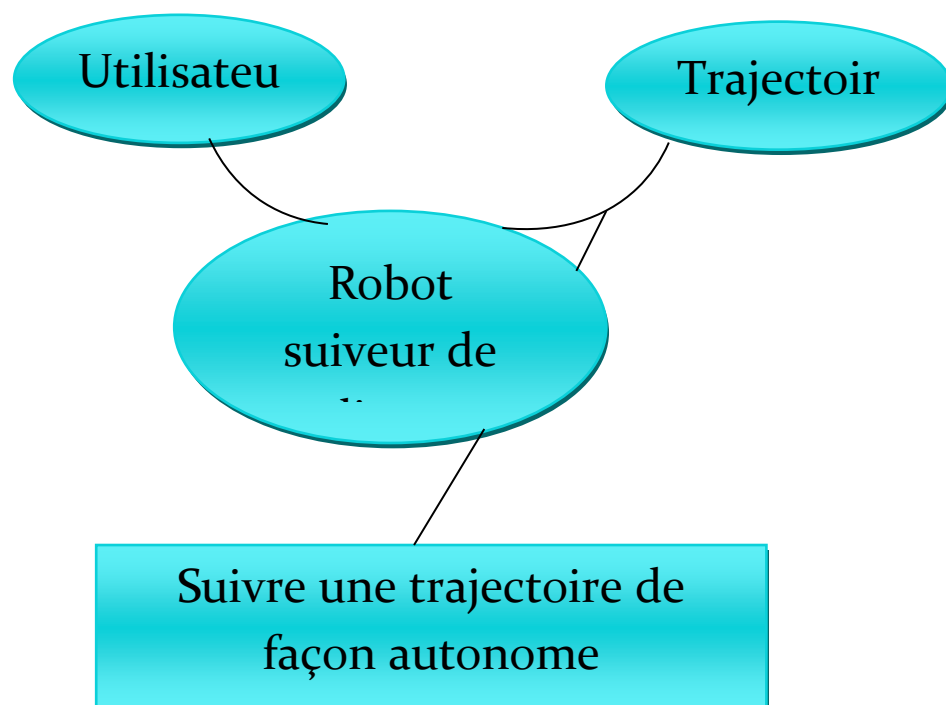


Figure II.1 : Expression fonctionnelle de besoin

## 1. Principe de fonctionnement du robot suiveur de ligne

Un châssis motorisé suit une ligne noire de manière autonome. Des détecteurs observent la position de la ligne et donnent les ordres de correction correspondants au robot. Une alimentation autonome basée sur l'association d'accumulateur de type batterie rechargeable (9 V), permettant de délivrer respectivement les tensions d'alimentation des moteurs et les circuits intégrés programmables du module Arduino. Le schéma synoptique décrivant les différents blocs incorporés dans le robot étudié est consigné dans la figure II.2.

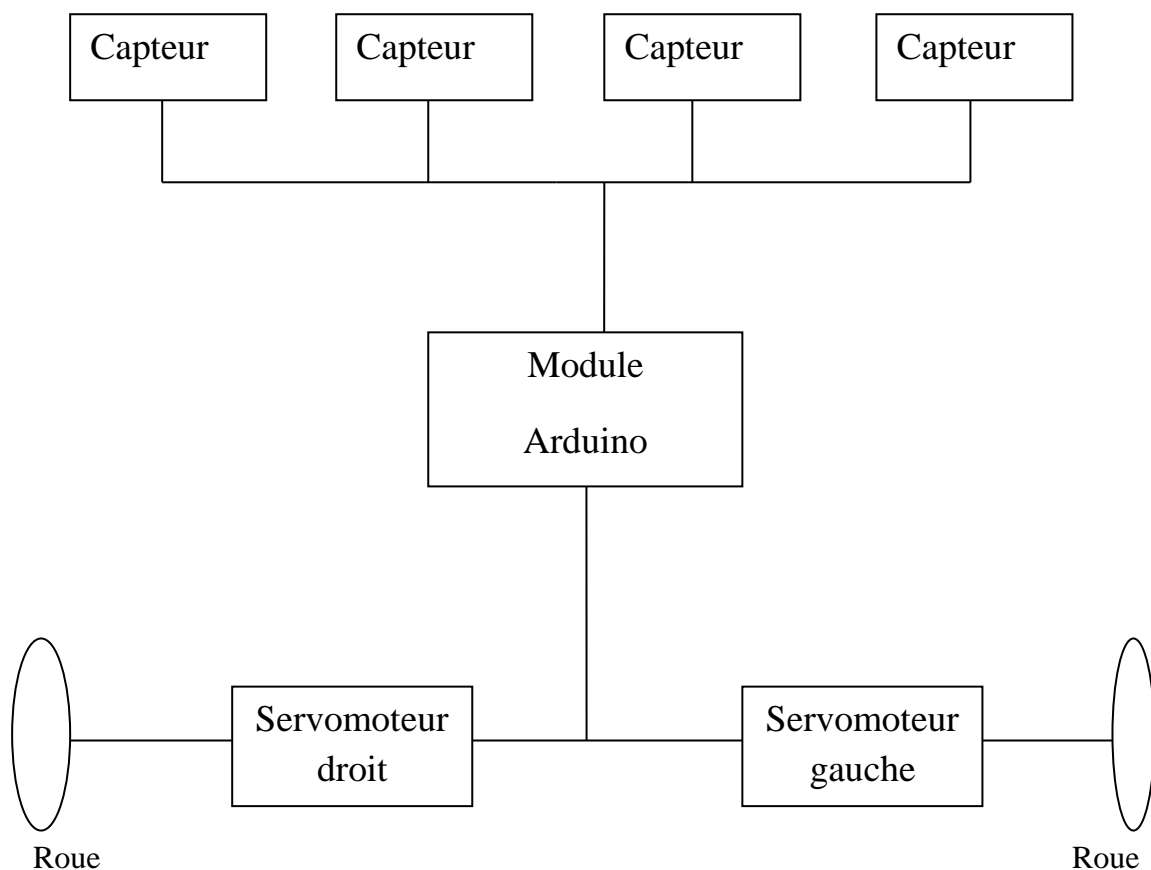


Figure II.2 : Schéma synoptique du robot suiveur de ligne

## 2. Architecture de la carte de commande

La carte de commande du robot mobile, est architecturée au tour du module Arduino associé à d'autres circuits intégrés. Dans cet esprit nous présentons une étude descriptive de ces différents composants. Dans notre projet nous allons utiliser l'Arduino Uno.

### 2.1. Architecture interne du module Arduino

Nous avons présenté dans cette partie l'étude globale du module Arduino Uno, comme le montre la figure II.3.

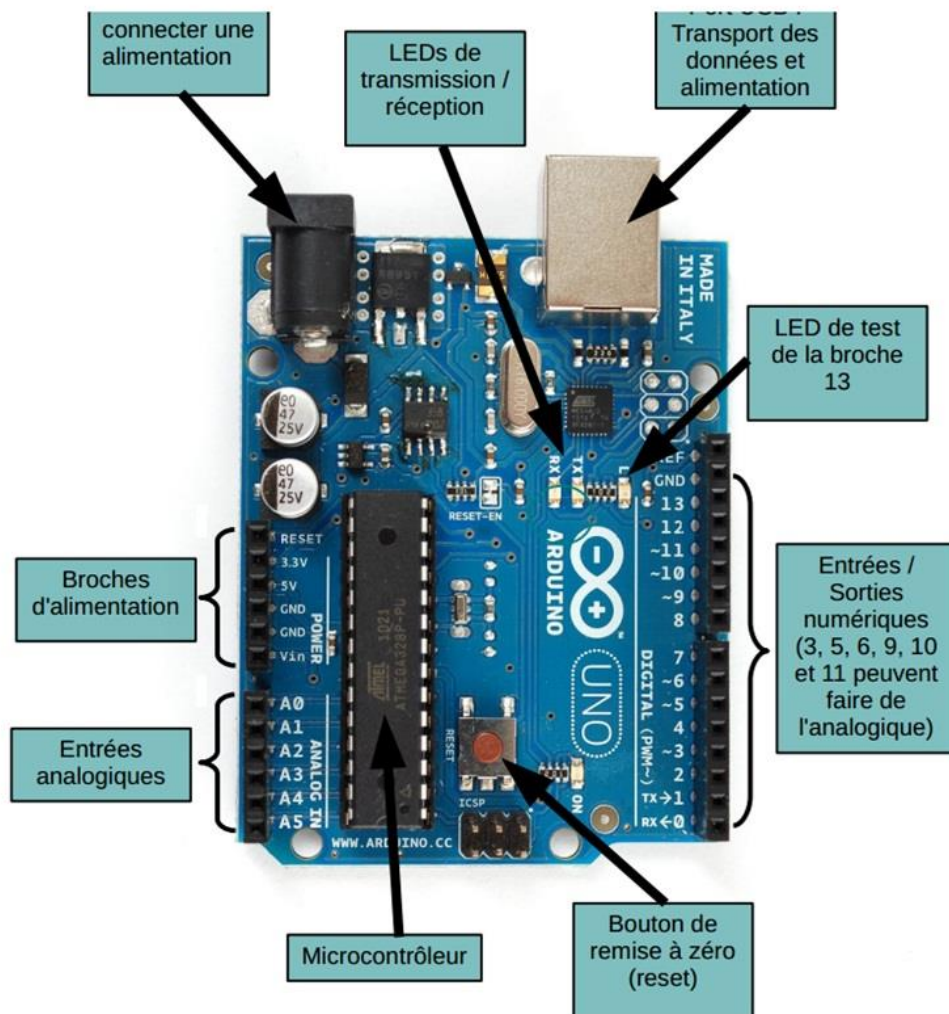


Figure II.3 : Brochage de la carte Arduino Uno

### ➤ Alimentation

La carte Arduino Uno peut-être alimentée, soit via la connexion USB (qui fournit 5V jusqu'à 500mA), ou à l'aide d'une alimentation externe. La source d'alimentation est sélectionnée automatiquement par la carte. L'alimentation externe (non-USB) peut être soit un adaptateur de secteur (pouvant fournir typiquement de 3V à 12V sous 500mA) ou des piles (ou des accumulateurs). L'adaptateur secteur peut être connecté en branchant une prise 2.1mm positif au centre dans le connecteur jack de la carte. Les fils en provenance d'un bloc de piles, ou d'accumulateurs peuvent être insérés dans les connecteurs des broches de la carte appelées Gnd (masse ou 0V) et Vin (Tension positive en entrée) du connecteur d'alimentation. La carte peut fonctionner avec une alimentation externe de 6 à 20 volts. Cependant, si la carte est alimentée avec moins de 7V, la broche 5V pourrait fournir moins de 5V et la carte pourrait être instable. Si on utilise plus de 12V, le régulateur de tension de la carte pourrait chauffer et endommager la carte. La plage idéale recommandée pour alimenter la carte Uno est entre 7V et 12V.

Les broches d'alimentation sont les suivantes :

- 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 5V régulée). La carte peut être alimentée à l'aide de cette broche, ou, si l'alimentation est fournie par le jack d'alimentation.
- 5V. La tension régulée utilisée pour faire fonctionner le microcontrôleur, et les autres composants de la carte. 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, ou de tout autre source d'alimentation régulée.
- Une alimentation de 3.3V fournie par le circuit intégré FTDI 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
- GND. Broche de masse (ou 0V).

### ➤ Mémoire

L'ATmega 328 a 32Ko de mémoire FLASH pour stocker le programme, dont 0.5Ko également utilisés par le bootloader. Ce dernier est un programme préprogrammé une fois pour toute, dans l'ATméga et qui permet la communication entre l'ATmega et le logiciel Arduino via le port USB, notamment lors de chaque programmation de la carte. L'ATmega 328 a également 2ko de mémoire SRAM et 1Ko d'EEPROM.

### ➤ Entrées et sorties numériques

Chacune des 14 broches numériques de la carte UNO (numérotées des 0 à 13) peut être utilisée, soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions [pinMode\(\)](#), [digitalWrite\(\)](#) et [digitalRead\(\)](#) du langage Arduino. Ces broches fonctionnent en 5V. Chaque broche, peut fournir ou recevoir un maximum de 40mA d'intensité, et dispose d'une résistance interne 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\(broche, HIGH\)](#). De plus, certaines broches ont des fonctions spécialisées :

- **Communication Série:** Broches 0 (RX) et 1 (TX). Utilisées pour recevoir (RX) et transmettre (TX) les données sériel de niveau TTL. Ces broches sont connectées aux broches correspondantes du circuit intégré ATmega8U2 programmé en convertisseur USB-vers-série de la carte, composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur.
- **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. Voir l'instruction [attachInterrupt\(\)](#) pour plus de détails.
- **Impulsion PWM (Pulse Width Modulation):** Les broches 3, 5, 6, 9, 10, et 11 fournissent une impulsion PWM 8-bits à l'aide de l'instruction [analogWrite\(\)](#).
- **SPI (Serial Peripheral Interface):** Les broches 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK) supportent la communication SPI disponible avec la librairie

pour communication. Les broches SPI sont également connectées sur le connecteur ICSP qui est mécaniquement compatible avec les cartes Mega.

- **I2C:** Les broches 4 (SDA) et 5 (SCL), supportent les communications de protocole I2C (ou interface TWI (Two Wire Interface), disponible en utilisant la librairie Wire/I2C.
- **LED:** 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.

#### ➤ Broches Analogiques

La carte Uno dispose de 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 [analogRead\(\)](#) 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 [analogReference\(\)](#) du langage Arduino. Les broches analogiques peuvent être utilisées en tant que broches numériques numérotées de 14 à 19.

#### ➤ Autres broches

Il y a deux autres broches disponibles sur la carte :

- **AREF :** Tension de référence pour les entrées analogiques (si différent du 5V). Utilisée avec l'instruction `analogReference()`.
- **Reset :** Mettre cette broche au niveau BAS entraîne la réinitialisation du microcontrôleur. Typiquement, cette broche est utilisée pour ajouter un bouton de réinitialisation sur le circuit qui bloque celui présent sur la carte.

#### ➤ Protection du port USB contre la surcharge en intensité

La carte Arduino Uno, intègre un polyfusible réinitialisable qui protège le port USB de votre 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.

### **3. Choix des actionneurs**

#### **3.1. Les servomoteurs**

Pour le pilotage du robot mobile, nous avons fait le choix sur les servomoteurs. Ces actionneurs offrent la possibilité d'être commandés par un seul fil qui au travers d'une PWM donne l'ordre de tourner à la vitesse et dans le sens souhaités. Ces moteurs sont intéressants, car ils intègrent mécanique et électronique dans un même boîtier et offrent un couple suffisant pour notre application. D'un point de vue vitesse on parlera d'environ 60 à 120 RPM (tours par minute) en fonction du modèle. On les utilise donc comme actionneurs ou pour la motorisation de petits robots, éventuellement en modifiant leur mécanique pour qu'ils tournent continuellement, supprimant donc l'asservissement en position. Les plus évolués disposent de fonctions supplémentaires (contrôle du courant, asservissement en vitesse, connexion en bus sur des protocoles de communication divers, capteur de température - pour l'échauffement, etc.). Pour utiliser ce type de moteur, il faut modifier un servomoteur de modélisme existant. En effet ces derniers ont une course angulaire limitée (l'axe n'est pas prévu pour réaliser une rotation complète), la position de l'axe du moteur est mesurée par un potentiomètre qui permet de l'orienter à un angle bien précis. Si, on veut utiliser un pareil moteur pour faire avancer le robot suiveur de ligne, il faut court-circuiter le potentiomètre et retirer les blocages mécaniques limitant la rotation de l'axe. Les modèles diffèrent en taille, en puissance, en fiabilité et surtout en solidité des matériaux utilisés : plastique, laiton, acier. Certains sont équipés d'un ou plusieurs roulements à bille pour résister aux pressions transverses sur l'axe moteur.



Figure II.4 : Servomoteur HS-311

### 3.1.1. Commande électronique d'un servomoteur

En temps normal un servo est piloté par des impulsions électriques de largeur spécifique, qui permettent de faire pivoter l'arbre de sortie du servo d'un angle spécifique, par rapport à sa position neutre. Cette position est détectée par le circuit interne au servo, grâce à un potentiomètre dont l'axe est solidaire à l'axe de sortie du servo. Une butée est également présente, pour éviter de détériorer le potentiomètre au cas où l'angle réglé serait trop important. Comme il est consigné dans la figure (II.5) de commande est très simple, elle consiste à appliquer une alimentation de 4.8V à 6V sur deux fils du servomoteur et une impulsion sur le troisième. Une impulsion de 1,5 millisecondes centre l'axe à sa position dite neutre, une impulsion de 1 milliseconde fait pivoter l'axe de 45 degrés, tandis qu'une impulsion de 2 millisecondes le fait pivoter de - 45 degrés

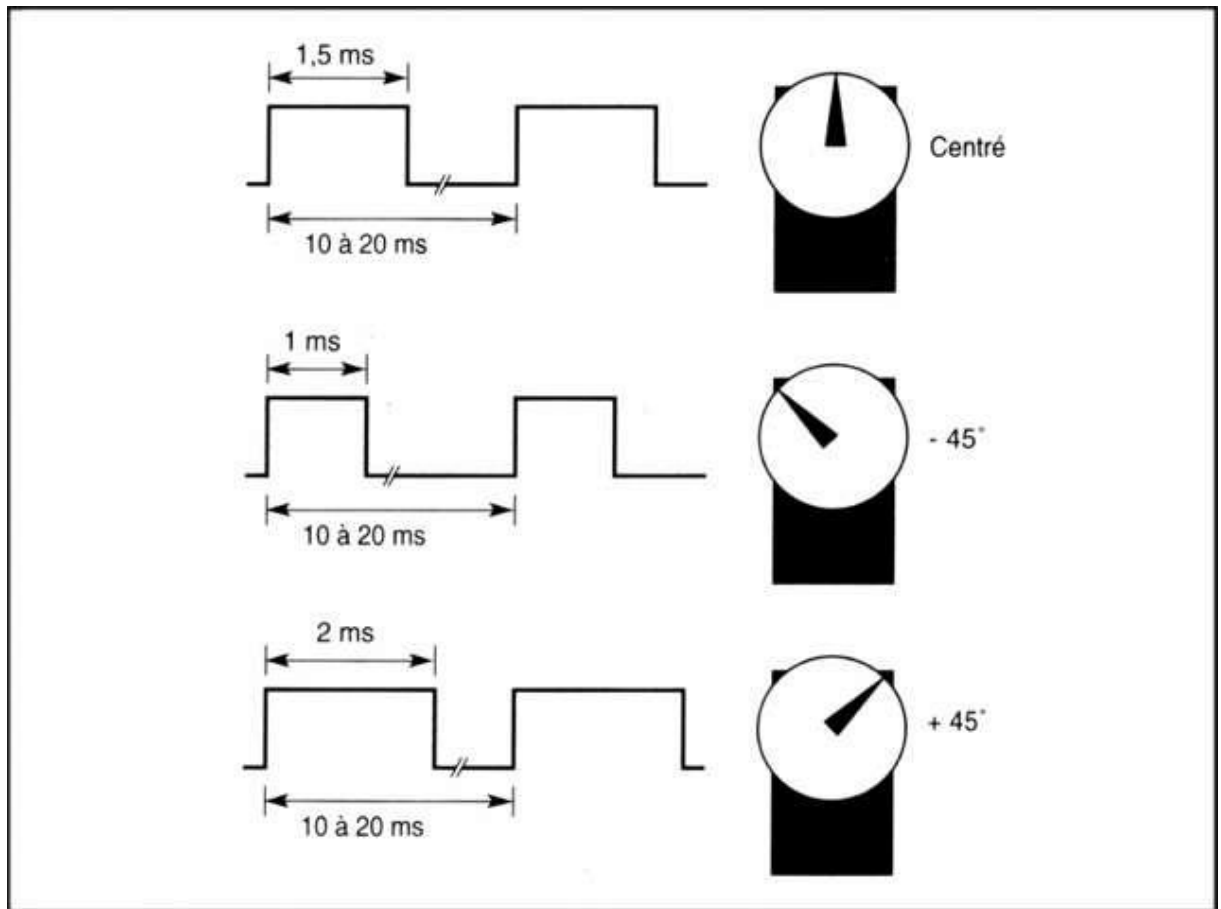


Figure II.5 : Servomoteur commandé pour PWM

#### Les avantages des servomoteurs sont les suivants

- Le fil signal à faible courant peut être raccordé directement à une sortie du PIC. Pas besoin de circuit d'interface.
- On peut commander l'arrêt, la marche, le sens de rotation et la vitesse du servo à l'aide d'un seul fil. Economie d'E/S.
- Le servo tourne à la bonne vitesse du robot.
- Le servo offre un couple important sous un volume réduit.

**Inconvénients des servomoteurs sont :**

- Modification du servo pour une rotation complète
- Le prix est légèrement plus élevé qu'un bloc motoréducteur à 2 moteurs CC.

### 3.2. Moteurs à courant continu

Ils ont l'avantage d'être commandables facilement au travers d'une petite électronique dédiée, et leur prix est souvent assez faible. Il faut néanmoins, placer un réducteur derrière car les petits moteurs CC ont souvent des vitesses de rotation dépassant les 5.000 RPM (tour par minute). Il serait impossible de placer des roues directement sur l'axe du moteur, premièrement, car la roue tournerait à des vitesses beaucoup trop importantes, deuxièmement car pour ce genre de petit moteur, le couple fourni est faible. Il existe cependant des moteurs CC avec réducteur intégré.



Figure II.6: Moteur à courant continu

### 3.3. Moteur pas à pas

Ce dernier offre la possibilité de régler très précisément la vitesse et la position de l'axe. Le couple du moteur est également important, même à très faible vitesse. Mais la commande

de ces moteurs est beaucoup plus complexe et le prix est également élevé. De plus, ce type de moteur est aussi rapidement encombrant.



Figure II.7 : Moteur pas à pas

#### 4. Architecture de la carte de puissance

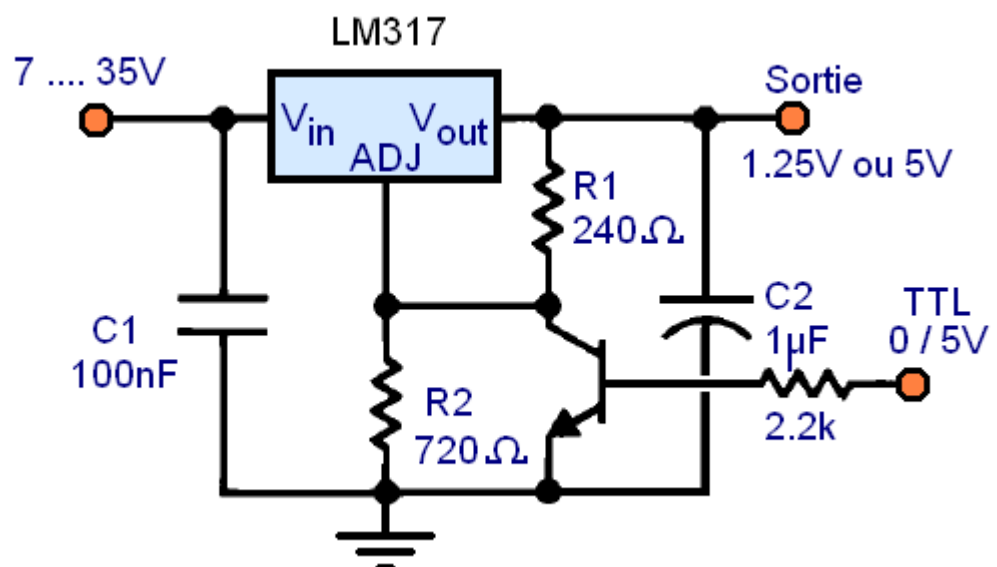


Figure II.8 : Schéma de la carte de puissance

Principe de fonctionnement de la carte de puissance, la tension de sortie vaut 5V si l'entrée TTL n'est reliée à rien ou à la masse. Lorsque l'entrée TTL est à 5V, le transistor court-circuite R2 et la tension de sortie passe à 1.25V. Avec une alimentation de 1.25V, la plupart des circuits consomment très peu et sont ainsi en veille.

## 5. Capteur infrarouge émetteur récepteur (QRD1114)

Le QRD1114 est en fait 2 composants, un transmetteur IR et un phototransistor, réunis en un seul paquet. Ceci, comme le GP2Y0A21YK, fonctionne par projection d'une lumière infrarouge, et de voir dans quelle mesure il rebondit. Un objet qui est plus proche va rebondir plus léger qu'un lointain. Le QRD1114 est seulement capable de détecter des objets entre 0 et 3cm de suite. Il ce n'est pas destiné à être utilisé pour déterminer la distance exacte, mais de vérifier la proximité des objets sous 3cm loin. Ce capteur peut également être utilisé pour détecter les surfaces blanches / noir, car une surface blanche reflétera plus de lumière, qu'une surface noire pour résultat une lecture plus élevé. En effet, le capteur mesure simplement l'intensité IR et n'est pas au courant de la source de la lumière, il est sensible aux faux positifs causés par des sources extérieures qui mettent hors IR tels que les flashes, lampes, et même le soleil.



Figure II.9 : Capteur infrarouge QRD1114

## Conclusion

Dans ce chapitre, nous avons présenté une étude descriptive des différents composants nécessaire pour la mise en œuvre de la carte de commande, carte de puissance et carte du capteur du robot mobile et qui sont respectivement (le module Arduino, servomoteur, capteur infrarouge).

## *Chapitre 3*

# *Outils de développement et réalisation du robot*

La partie réalisation et la partie la plus pesante dans un tel projet, car il constitue la partie de mise en évidence de l'application de notre projet.  
Dans ce chapitre, on va traiter le processus de la réalisation de ce robot , tout en parlant sur les outils utilisés dans le développement de robot.





## Introduction

L'objectif de ce chapitre est de présenter une description détaillée des différentes étapes de simulation de la carte : saisie des schémas électroniques de la carte, exécution de la simulation des différents blocs avec le logiciel ISIS pour vérifier le fonctionnement du programme à partir des résultats obtenus.

### 1. Conception mécanique

Le robot est constitué de deux roues motrices indépendantes qui permettront d'avancer/reculer mais également, en donnant des vitesses de rotation différentes. La troisième roue est libre permettra de maintenir l'équilibre et le guidage du robot comme le montre la figure (III.1)

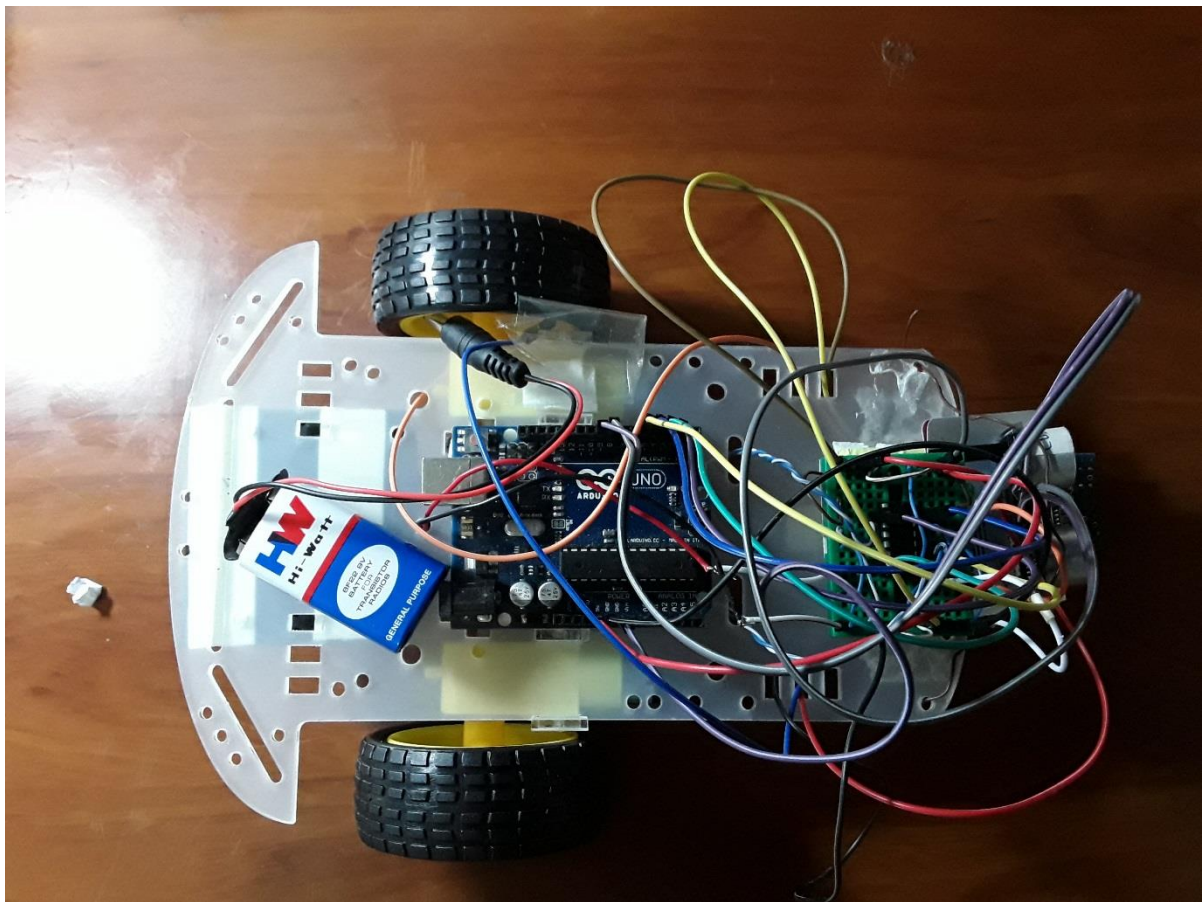


Figure III.1 : Conception mécanique du robot

## 2. Réalisation des cartes électroniques et simulation

Après études générales des différents éléments constituant nos cartes électroniques, on passe maintenant à la réalisation physique de notre projet. Dans cette partie on touchera aux différents logiciels et outils utilisés pour la création de nos trois cartes.

### 2.1. Présentation d'ISIS

L'ISIS est un logiciel professionnel, utilisé dans l'électronique pour simuler des circuits et créer des typons. Il est également capable de simuler le fonctionnement du PIC avec tous les périphériques de la carte de commande. L'utilisation du logiciel « ISIS » permet de mieux visualiser le bon déroulement du système, ainsi, que d'avoir une idée claire sur la partie matérielle, et la conception des circuits imprimés.

### 2.2. Carte de puissance

Comme le montre la figure (III.2), la carte de puissance est une alimentation stabilisée permet de régler la tension pour alimenter les servomoteurs et les capteurs.

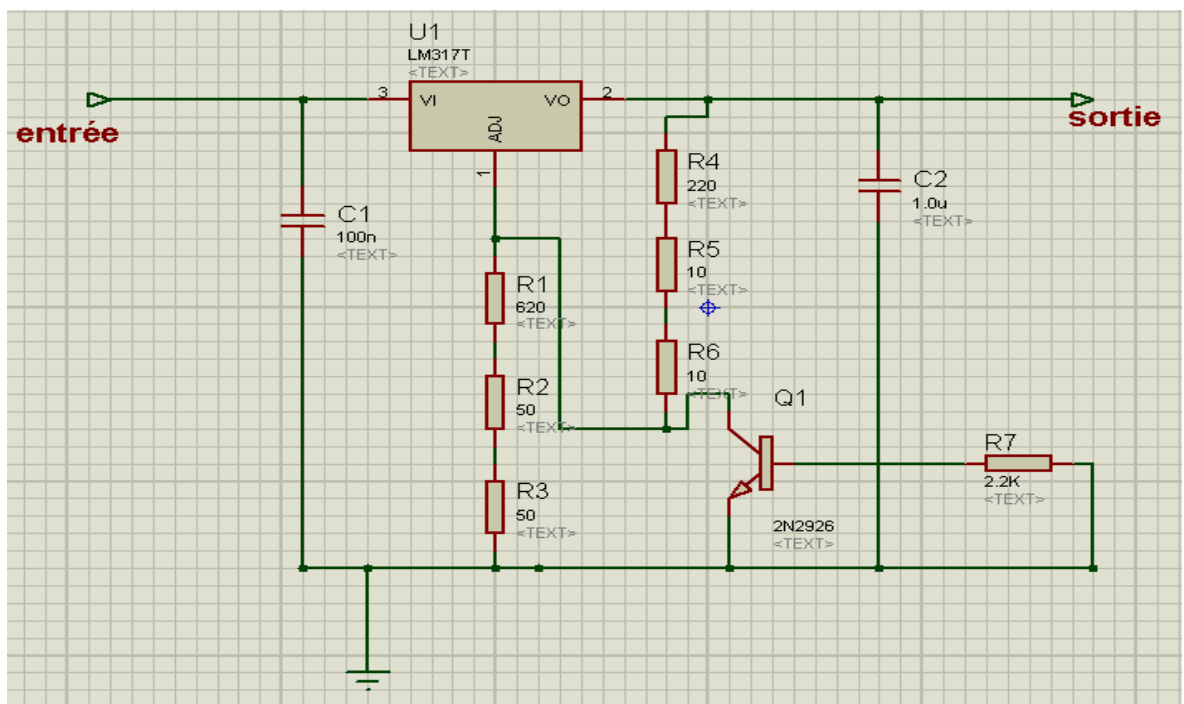


Figure III.2 : Schéma de simulation de la carte de puissance

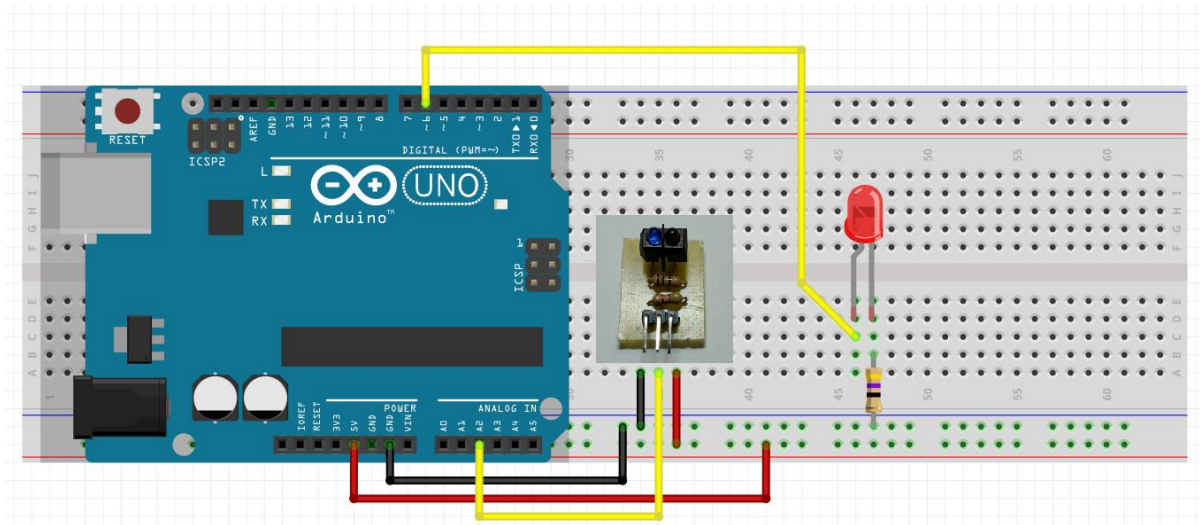


Figure III.3 : Montage de capteur TCRT5000 sur arduino

## 2.3. Carte de commande

### 2.3.1. Brochage d'Arduino avec la carte des capteurs

La carte des capteurs contient quatre capteurs infrarouges émetteur-récepteur, placé en avant du robot, lui permettant de détecter la ligne noire ensuite de la suivre.

Le QRD1114 est un peu plus compliqué pour brancher que les autres, mais c'est parce que c'est en fait seulement deux composants bruts. Mais ... il ne nécessite que deux résistances, (200ohm et un 4.7k - 5.6k ohm) pour l'obtenir en place et fonctionne. Comme il est illustré dans la figure (III.4). Les résistances 4.7k $\Omega$  et 5.6k $\Omega$  sont de type pull-up.

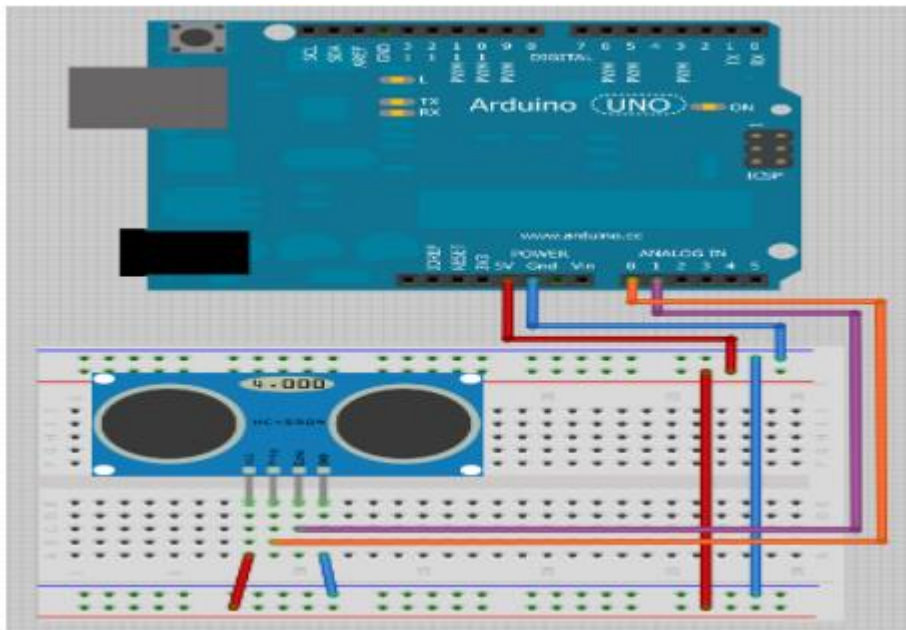
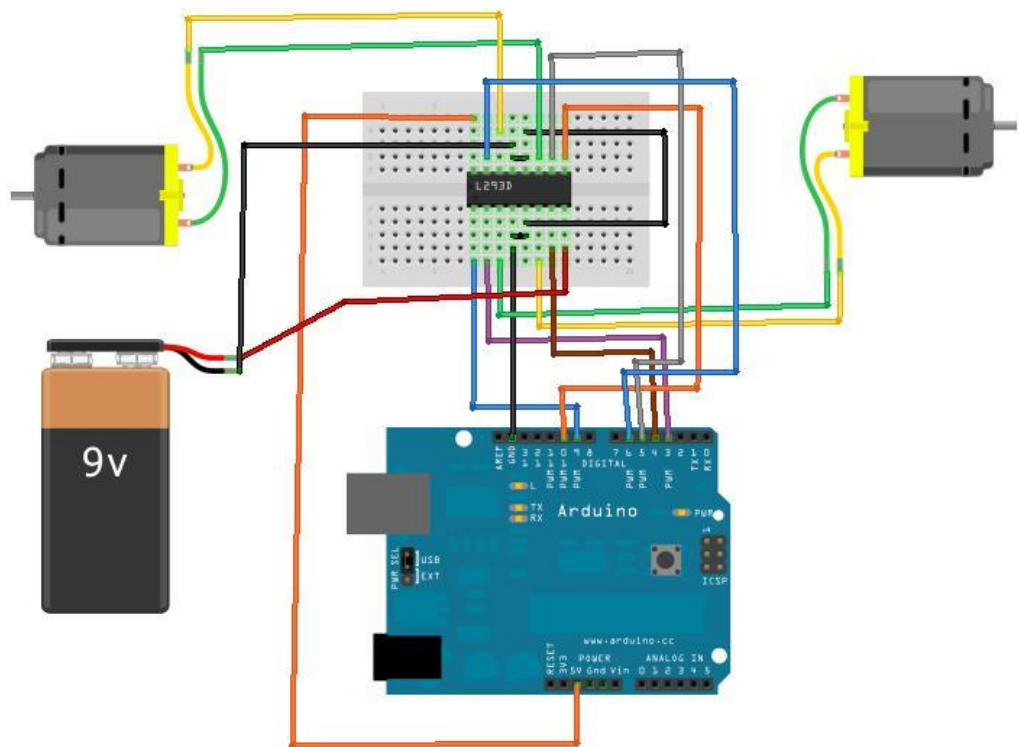


Figure III.4 : Brochage de capteur avec Arduino



Made with  Fritzing.org

Figure III.5 : Montage Motors

### 2.3. Brochage d'Arduino avec le servomoteur

Tout comme le servomoteur standard, le servomoteur à rotation continue dispose d'un câble à 3 broches :

- d'une broche +5V (fil rouge)
- d'une broche 0V (fil noir)
- d'une broche de contrôle qui reçoit l'impulsion de commande (fil jaune) à connecter sur une broche E/S de la carte Arduino en sortie.

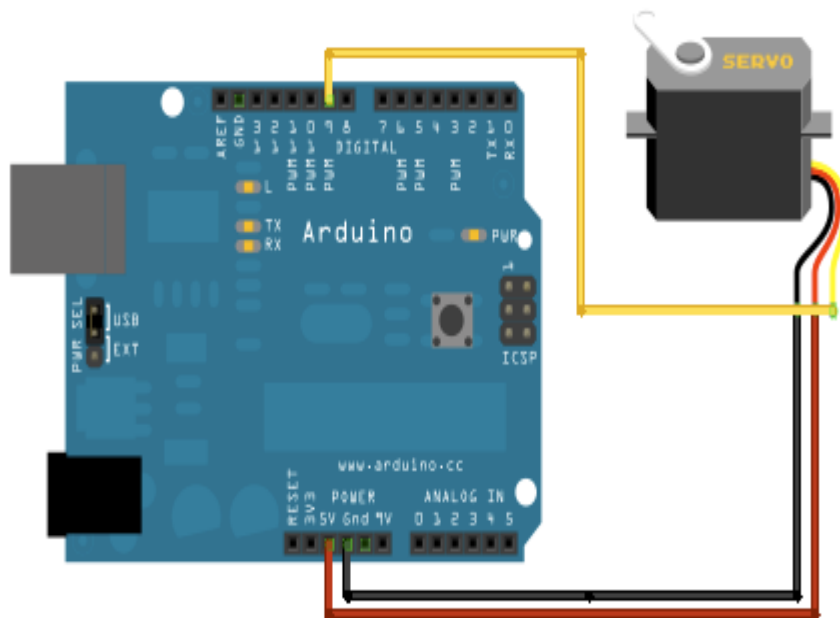


Figure III.6 : Brochage du servomoteur avec Arduino

## 3. Programmation

### 3.1. Algorithme de l'approche de commande

L'organigramme est un diagramme qui se lit à différents niveaux. Ici, le système est abordé de deux points de vue différents. Le diagramme est une suite de directives composées d'actions et de décisions qui doivent être exécutées selon un enchaînement strict pour réaliser une tâche (appelée : séquence).



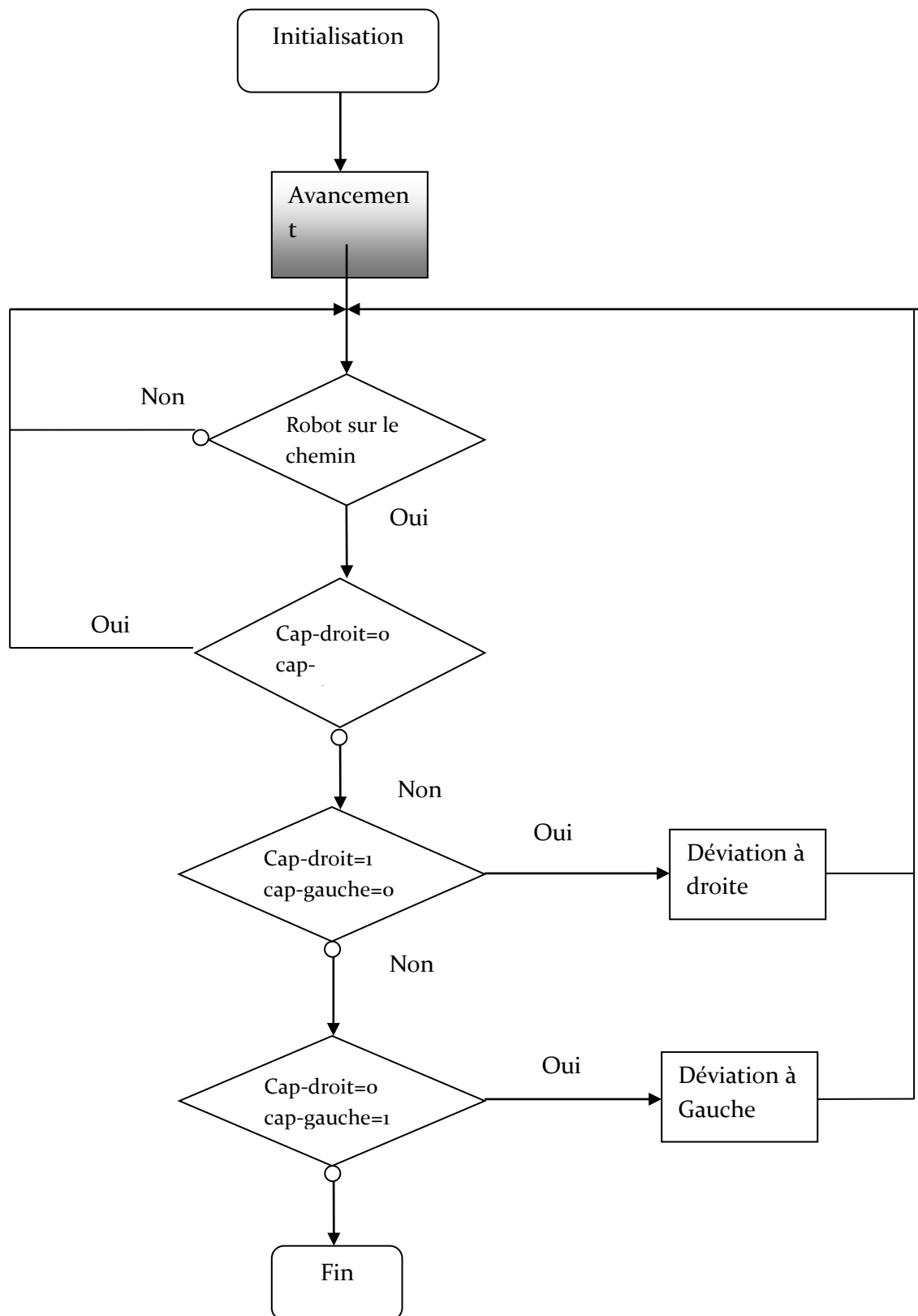


Figure III.7 : Organigramme

### 3.2. Embarquement du programme sur l'Arduino

Le logiciel Arduino a pour fonctions principales :

- de pouvoir écrire et compiler des programmes pour la carte Arduino
- de se connecter avec la carte Arduino pour y transférer les programmes
- de communiquer avec la carte Arduino

Cet espace de développement intégré (EDI) dédié au langage Arduino et à la programmation des cartes Arduino comporte :

- une **BARRE DE MENUS** comme pour tout logiciel une interface graphique.
- une **BARRE DE BOUTONS** qui donne un accès direct aux fonctions essentielles du logiciel et fait toute sa simplicité d'utilisation.
- un **EDITEUR** (à coloration syntaxique) pour écrire le code de vos programmes, avec onglets de navigation.
- une **ZONE DE MESSAGES** qui affiche indique l'état des actions en cours.
- une **CONSOLE TEXTE** qui affiche les messages concernant le résultat de la compilation du programme.

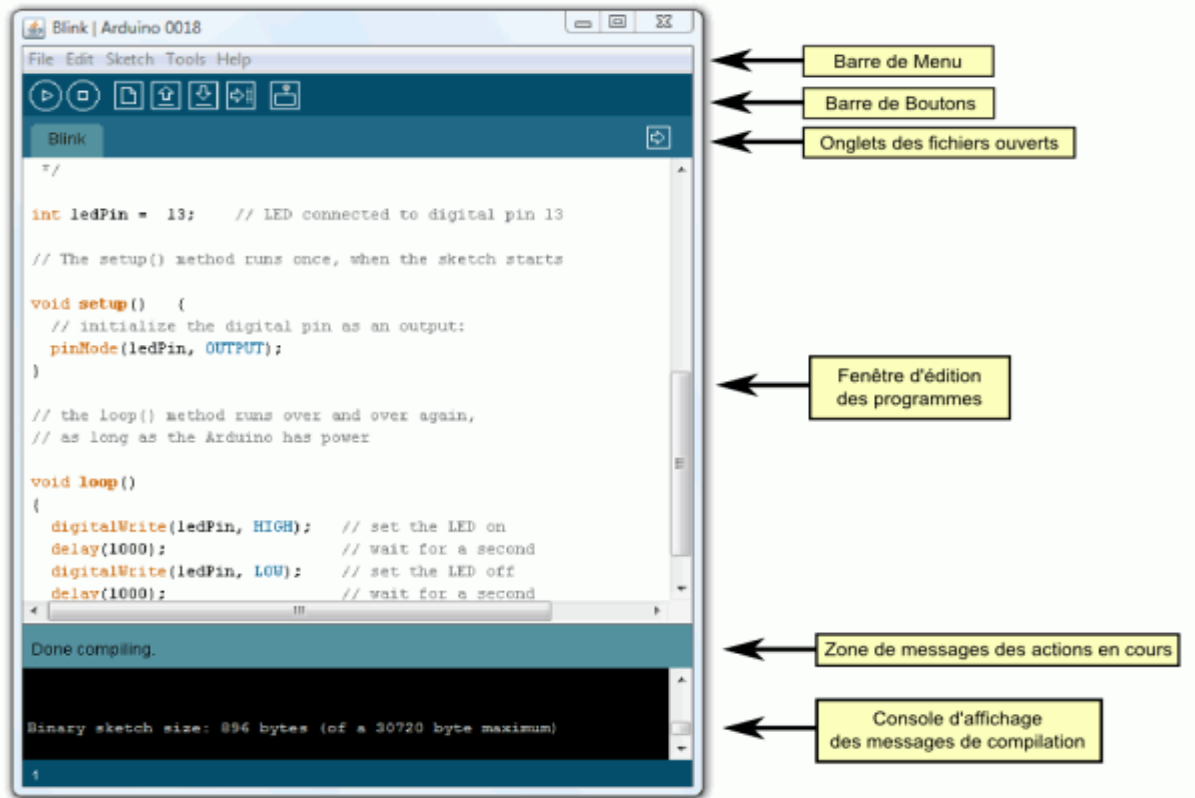


Figure III.8 : Logiciel Arduino

Le langage Arduino est basé sur les langages C/C++ et supporte toutes les constructions standards du langage C et quelques-uns des outils C++. Le langage Arduino repose sur l'utilisation du compilateur C pour les microcontrôleurs AVR, AVR Libc, et nous permet d'utiliser la plupart de ses fonctions.

Nous avons créé le programme du module Arduino en se basant sur la programmation des servomoteurs et des capteurs infrarouges.

### Le code :

```
const int leftsensor=12;

const int rightsensor=11;

const int trig = 9;

const int echo = 8;
```



```
const int leftForward = 2;

const int leftBackward = 3;

const int rightForward = 4;

const int rightBackward = 5;


int duration = 0;

int distance = 0;

int abx,xyz;


void setup()

{

    pinMode(trig , OUTPUT);

    pinMode(echo , INPUT);

    pinMode(leftsensor,INPUT);

    pinMode(rightsensor,INPUT);

    pinMode(leftForward , OUTPUT);

    pinMode(leftBackward , OUTPUT);

    pinMode(rightForward , OUTPUT);

    pinMode(rightBackward , OUTPUT);


    Serial.begin(9600);
```

```
}

void loop()
{
    digitalWrite(trig , HIGH);
    delayMicroseconds(1000);
    digitalWrite(trig , LOW);
    abx=digitalRead(leftsensor);
    xyz=digitalRead(rightsensor);

    duration = pulseIn(echo , HIGH);
    distance = (duration/2) / 28.5 ;
    Serial.println(distance);

    if ( distance < 20 )
    {
        digitalWrite(leftForward , LOW);
        digitalWrite(leftBackward , HIGH);
    }
}
```

```
digitalWrite(rightForward , HIGH);

digitalWrite(rightBackward , LOW);

delay(100);

}

else

{

digitalWrite(leftForward , HIGH);

digitalWrite(leftBackward , LOW);

digitalWrite(rightForward , HIGH);

digitalWrite(rightBackward , LOW);

}

if(abx==HIGH && xyz==HIGH)

{

digitalWrite(leftForward,HIGH);

digitalWrite(rightForward,HIGH);

digitalWrite(leftBackward,LOW);

digitalWrite(rightBackward,LOW);

}

else if(abx==HIGH && xyz==LOW)

{

digitalWrite(leftForward,HIGH);
```

```
digitalWrite(leftBackward,LOW);

digitalWrite(rightForward,LOW);

digitalWrite(rightBackward,LOW);

}

else if(abx==LOW && xyz==HIGH)

{

digitalWrite(leftForward,LOW);

digitalWrite(leftBackward,LOW);

digitalWrite(rightForward,HIGH);

digitalWrite(rightBackward,LOW);

}

else

{

digitalWrite(leftForward,LOW);

digitalWrite(leftBackward,LOW);

digitalWrite(rightForward,LOW);

digitalWrite(rightBackward,LOW);

}

}
```

## Conclusion

Ce travail s'est révélé être une expérience enrichissante pour moi. Le fait d'avoir un nouveau langage informatique à apprendre et à maîtriser me faisait peur au début mais il s'est avéré que celui utilisé pour coder sur Arduino est plutôt facile d'utilisation et les explications de Simon m'ont beaucoup aidé à comprendre. Ayant déjà de bonnes bases en pascal, je n'ai pas eu trop de mal à assimiler ce nouveau langage. Pour ce qui est de la partie relationnelle, tout s'est très bien passé. Nous n'avons eu aucun désaccord et nous nous sommes très bien entendus. De plus, cela m'a beaucoup appris sur les manières de travailler en équipe et comment échanger avec ses collègues de manière constructive. Je regrette cependant que nous n'ayons pas eu le réflexe de désigner un élève en tant que chef de groupe, ce qui aurait sûrement permis une meilleure transversalité entre les différentes tâches à réaliser. Cela nous aurait également fourni une vision plus claire sur l'avancement du projet, ce qui nous a fait défaut. Pour conclure, cet EC P6 s'est révélé enrichissant sur de nombreux plans, tant sur celui de l'informatique et de la conception, tant sur celui des relations entre collègues