



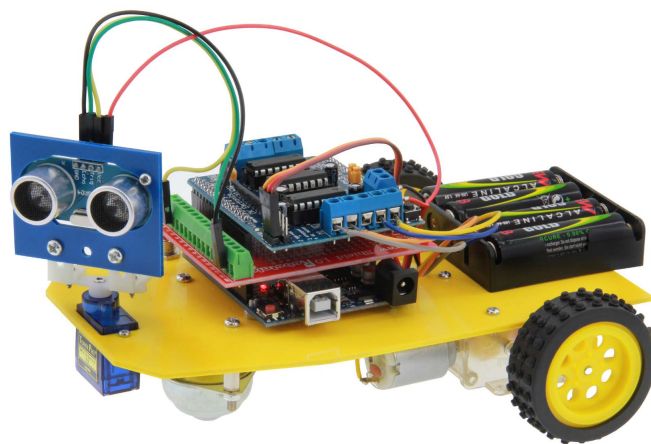
ÉCOLE HASSANIA DES TRAVAUX PUBLICS - DÉPARTEMENT GÉNIE ÉLECTRIQUE  
Casablanca - Maroc  
Juin 2020

## RAPPORT DE MINI PROJET

---

# ROBOT ÉVITEUR D'OBSTACLES

---



*Réalisé par :*

ABDELKADER BENAÏSSAT  
AMINE OUAZZANI TAYBI  
OUSSAMA ABDELHAKMI  
SALAH EDDINE CHAKIR

*Sous l'encadrement de :*

Mr EL RHARRAS ABDESSAMAD

# Résumé

La robotique est définie comme étant l'ensemble des techniques permettant la conception et la réalisation de machines automatiques ou de robots, elle est basée sur les capteurs, des systèmes logiques à base des microprocesseurs et les actionneurs. Le robot éviteur d'obstacle, machine servant à éviter la collision avec des obstacles inattendus, se compose aussi des dispositifs cités ci-dessus et qu'à travers lesquels il assure son mouvement. Dans les différentes parties de notre projet, on a compris le rôle de chaque composant dans la conception du robot, et par conséquent, en partant d'un robot particulier, on a pu déduire le fonctionnement des robots en général et le secret de leurs intelligences.

**Mots clés :** Eviteur d'obstacles, Arduino, Capteur, Robot, Electronique.

# Abstract

Robotics is defined as being the set of techniques allowing the design and the realization of automatic machines or robots, it is based on sensors, logic systems based on microprocessors and actuators. The obstacle avoidance robot, a machine used to avoid collision with unexpected obstacles, also consists of the devices mentioned above and that through which it ensures its movement. In the different parts of our project, we understood the role of each component in the design of the robot, and therefore, starting from a particular robot, we were able to deduce the functioning of robots in general and the secret of their intelligences.

**Keywords :** Obstacle avoidance, Arduino, Sensor, Robot, electronic.

# Table des matières

<b>I</b>	<b>Étude fonctionnelle</b>	<b>6</b>
1	Description du système . . . . .	7
2	Analyse fonctionnelle . . . . .	7
2.1	Graphe des interacteurs . . . . .	7
2.2	Caractérisation des Fonctions de Service . . . . .	8
2.3	Diagramme FAST du robot . . . . .	8
2.4	Grafcet fonctionnel du robot . . . . .	9
3	Conception du robot . . . . .	9
3.1	Choix de robot . . . . .	9
4	Cahier de charge . . . . .	12
<b>II</b>	<b>Matériaux et composants</b>	<b>14</b>
5	Vue globale . . . . .	15
6	La carte Arduino . . . . .	15
6.1	Choie du type de la carte . . . . .	15
6.2	Composantes de la carte Arduino UNO . . . . .	16
7	Capteurs ultra-son . . . . .	17
7.1	Le choix du capteur . . . . .	17
7.2	Fonctionnement du capteur . . . . .	17
8	Moteur DC . . . . .	19
8.1	Généralité sur les machines à courant continu . . . . .	19
8.2	La différence entre Moteur DC et Moteur AC . . . . .	19
8.3	Constitution d'un moteur DC . . . . .	20
8.4	Fonctionnement d'un moteur DC . . . . .	20
9	Servo-moteur . . . . .	21
9.1	Composition d'un servomoteur . . . . .	22
9.2	Commande d'un servomoteur à base de NE555 . . . . .	24
<b>III</b>	<b>Simulations</b>	<b>26</b>
10	Schéma du robot . . . . .	27
11	Simulation des différents circuits . . . . .	27
11.1	Premiers pas : initialisation en Arduino (Allumer une LED) . . . . .	27
11.2	Deuxième pas : utiliser un servomoteur . . . . .	29
11.3	Troisième pas : utiliser un capteur Ultra-son . . . . .	30
11.4	Dernier pas : simulation du robot . . . . .	34

# Table des figures

1	Graphe des interacteurs . . . . .	7
2	Diagramme FAST du robot . . . . .	8
3	Grafset fonctionnel du robot . . . . .	9
4	Robot tricycle . . . . .	10
5	Robot à vitesse différentielles symétrique . . . . .	10
6	Robot à à 4 roues . . . . .	11
7	Robot à vitesse différentielles asymétrique . . . . .	11
8	Cinématique du robot . . . . .	12
9	Un châssis du robot . . . . .	15
10	Une carte Arduino UNO . . . . .	16
11	Capteur ultra-son HC-SR04 . . . . .	17
12	Description du capteur ultra-son HC-SR04 . . . . .	17
13	Le champ de fonctionnement du capteur . . . . .	18
14	Moteur à courant continu . . . . .	19
15	Pont en H avec une branche active . . . . .	20
16	Schéma d'un Pont en H . . . . .	21
17	simulation d'un pont en H réaliser par Proteus à base des transistors . . .	22
18	Vue interne d'un servomoteur . . . . .	23
19	simulation d'un pont en H réaliser par Proteus à base des transistors . . .	23
20	Schéma de commande d'un servo-moteur à base d'un NE555 . . . . .	24
21	schéma électrique du robot . . . . .	27
22	Montage reliant une LED et un Arduino . . . . .	28
23	Montage reliant un servo-moteur et un Arduino . . . . .	29
24	Montage reliant un capteur ultra-son et un Arduino . . . . .	30
25	Première réponse du capteur ultra-son . . . . .	33
26	deuxième réponse du capteur ultra-son . . . . .	33
27	Montage reliant les déferent composantes du robot . . . . .	34

# Introduction

La robotique a depuis toujours été un domaine intéressant et en pleine croissance. Etant donné qu'elle est aussi considérée comme une branche de l'ingénierie, les applications de la robotique connaissent une considérable augmentation afin de se conformer au progrès technologique. C'est ainsi que le concept de robot mobile et ses complexités a évolué positivement avec différentes applications, notamment la planification de parcours l'auto-localisation et l'interprétation de cartes.

L'origine du mot robot provient de la langue tchèque dans laquelle sont ancêtre "robota" signifie travail forcé. Le terme de robotique quant à lui est apparu en 1942 dans le cycle universellement connu rédigé par Isaac Asimov et intitulé "Les robots".

Au cours de l'histoire on peut distinguer 3 types de robots correspondant en quelques sorte à l'évolution de cette "espèce" créée par l'Homme.

Le premier type de machine que l'on peut appeler robot correspond aux "Automates". Ceux-ci sont généralement programmés à l'avance et permettent d'effectuer des actions répétitives.

Le second type de robot correspond à ceux qui sont équipés de capteurs (en fait les sens du robot). On trouve des capteurs de température, photo-électronique, à ultrasons pour par exemple éviter les obstacles et/ou suivre une trajectoire. Ces capteurs vont permettre au robot une relative adaptation à son environnement afin de prendre en compte des paramètres aléatoires qui n'aurait pu être envisagés lors de leur programmation initiale. Ces robots sont donc bien plus autonomes que les automates mais nécessitent un investissement en temps de conception et en argent plus conséquent.

Enfin le dernier type de robot existant correspond à ceux disposant d'une intelligence dite "artificielle" et reposant sur des modèles mathématiques complexes tels que les réseaux de neurones. En plus de capteurs physiques comme leurs prédécesseurs, ces robots peuvent prendre des décisions beaucoup plus complexes et s'appuient également sur un apprentissage de leurs erreurs comme peut le faire l'être humain. Bien sûr il faudra attendre encore longtemps avant que le plus "intelligent" des robots ne soit égal, tant par sa faculté d'adaptation que par sa prise de décisions, à l'Homme.

Dans ce mini projet, nous allons nous intéresser à un type bien précis des robots mobiles : le robot évitant les obstacles, un robot autonome qui évite les collisions avec des obstacles inattendus, tout en focalisant notre intérêt sur la question suivante : comment serait-il possible de façonner un robot mobile évitant les obstacles qui combine à la fois efficacité et rapidité ?

Dans le but de répondre à cette problématique, nous allons dans un premier temps opter à une étude fonctionnelle portant sur notre robot mobile, en commençant par une analyse détaillée sur les différentes fonctions que doit satisfaire notre robot pour arriver enfin à obtenir une conception claire et nette de notre robot.

Ensuite, ayant déterminé les critères à satisfaire pour assurer un bon choix de matériaux et composants utilisés pour sa confection, on va présenter les différents outils technologiques en précisant leurs rôles puis leurs modes de fonctionnement.

On finira notre mini-projet par une partie dédiée à la simulation, en passant du local au global et où on va aborder des simulations des différentes parties du robot pour récolter à la fin une simulation complète de notre robot éviteur d'obstacles.

Première partie

Étude fonctionnelle

**Introduction.** Le robot éviteur d'obstacle, dans n'importe quel domaine où il est utilisé, doit nécessairement être accompagné d'une fiche technique qui décrira la totalité de ses fonctions et qui reflètera une image sur la qualité et la prestation de ses services. Ceci dit, une analyse fonctionnelle devait être mise en place pour exprimer le besoin de la construction du robot, définir les liens entre ses composants et rappeler les principales fonctions qu'il doit satisfaire. Les outils utilisés pour cette analyse fonctionnelle sont le diagramme pieuvre, le diagramme FAST et le GRAFCET.

## 1 Description du système

Notre robot éviteur d'obstacles utilise un capteur qui mesure la distance qui le sépare avec les différents objets qui l'entoure. L'information est traitée par un micro-contrôleur. Le robot avance et lorsqu'il rencontre un obstacle frontal, il s'arrête quelques instants, et il fait tourner le capteur respectivement vers la droite puis vers la gauche puis se tourne vers la direction la plus sûre et il continue en mouvement rectiligne.

## 2 Analyse fonctionnelle

### 2.1 Graphe des interacteurs

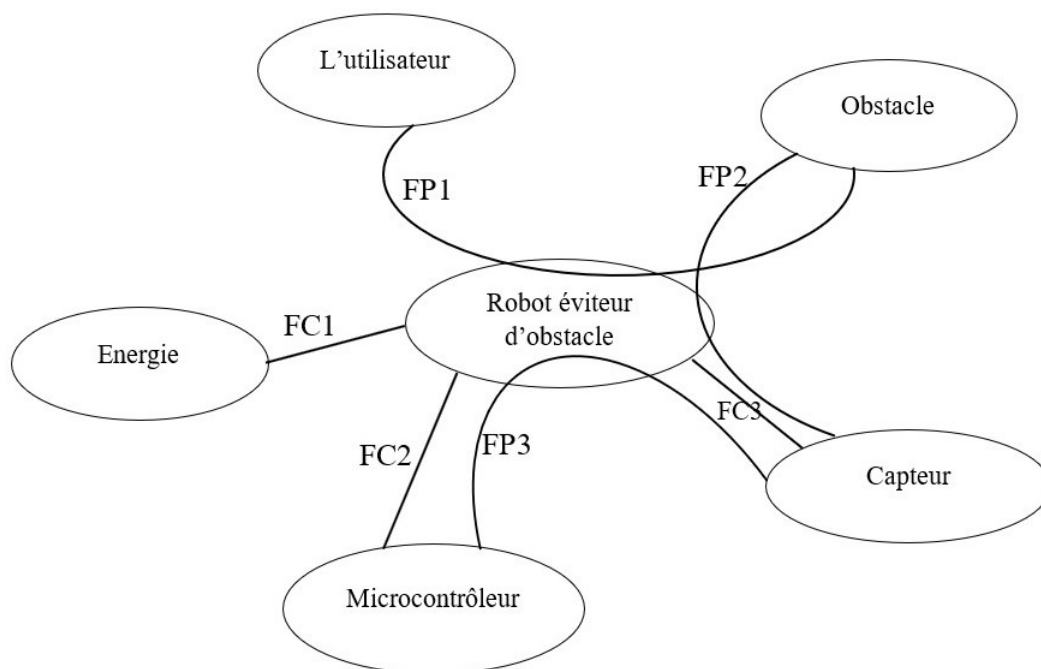


FIGURE 1 – Graphe des interacteurs

- FP : Fonctions Principales.
- FC : Fonctions Complémentaires ou Contraintes.



Fonction	Description
FP1	Eviter les obstacles pour servir l'utilisateur
FP2	Capter l'existence d'un obstacle
FP3	Analyser les données fournies par le capteur
FC1	Utiliser une source d'énergie pour alimenter les composants de robot
FC2	Contrôler le déplacement du robot
FC3	Indiquer l'existence d'un obstacle

## 2.2 Caractérisation des Fonctions de Service

FS	Critère	Niveau
FP1	Tourner à chaque fois un obstacle est rencontré	90
FP2	Avoir un minimum de distance pour détecter un obstacle	30cm
FP3	Détection rapide de la meilleur direction	Max 2 sec
FC1	Avoir une réserve d'énergie (en termes de tension)	9V
FC2	Changement rapide de direction	Une seconde

## 2.3 Diagramme FAST du robot

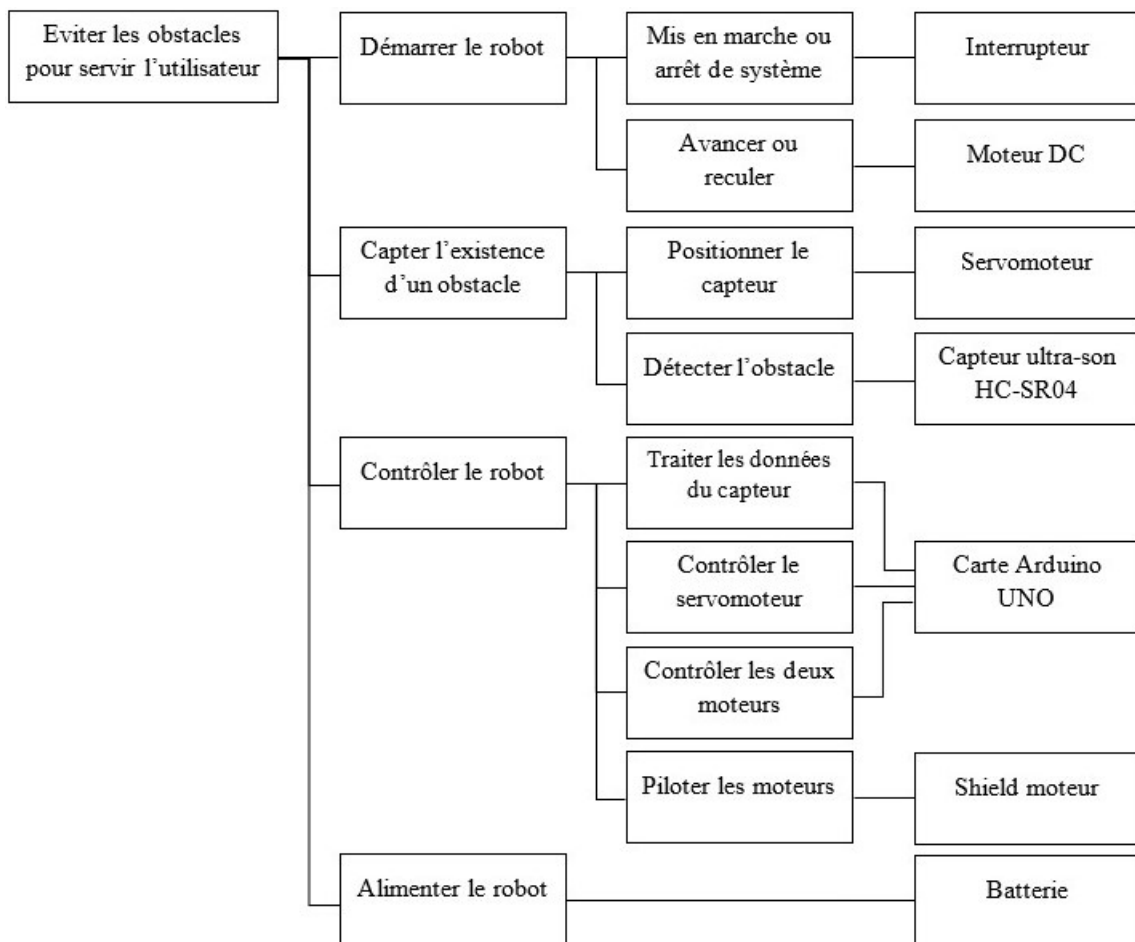


FIGURE 2 – Diagramme FAST du robot

## 2.4 Grafcet fonctionnel du robot

En s'appuyant sur les informations fournies charge ci-dessus, On a réalisé le grafcet de niveau 1 suivant :

On définit les actionneurs suivants :

m : mis en marche du robot

o : présence d'un obstacle

d : la probabilité de trouver un obstacle à droite plus sûr que le trouver à gauche

g : la probabilité de trouver un obstacle à gauche plus sûr que le trouver à droite.

Donc :

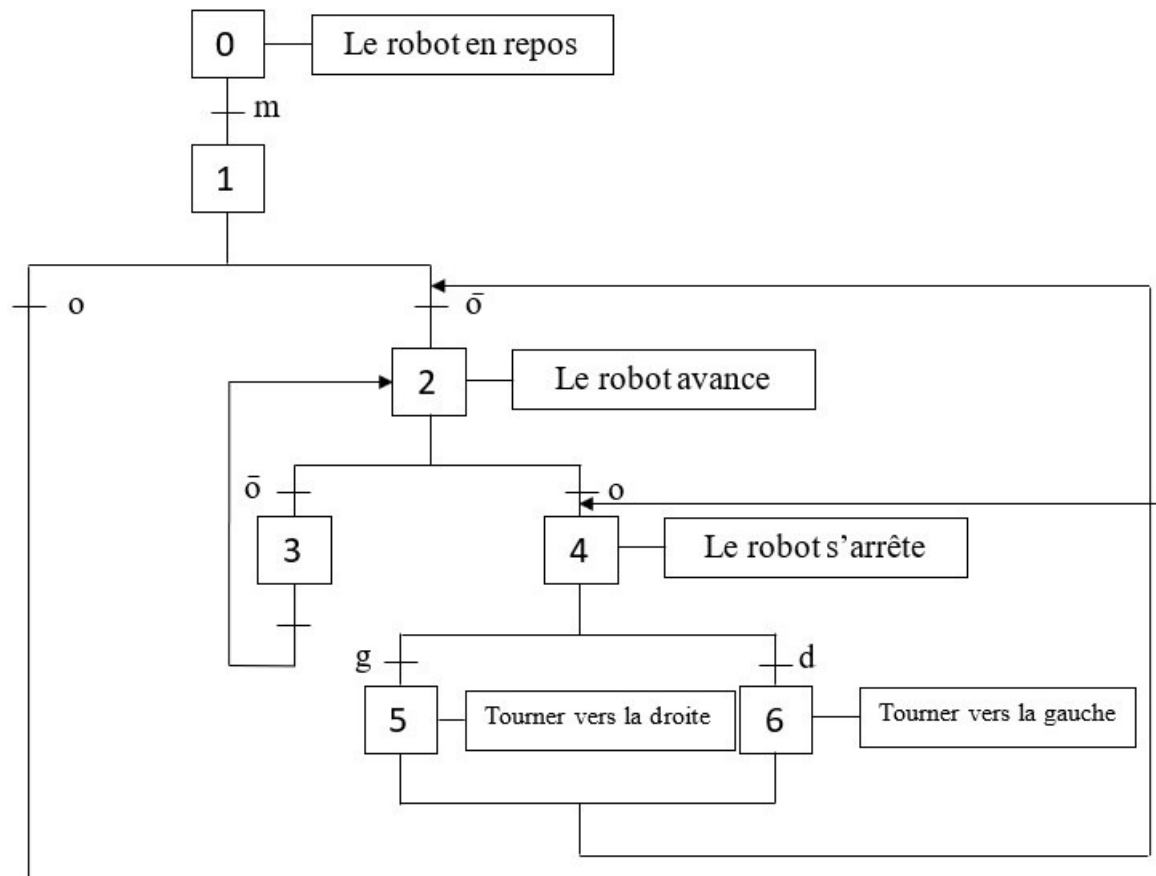


FIGURE 3 – Grafcet fonctionnel du robot

## 3 Conception du robot

### 3.1 Choix de robot

Afin de bien assimiler la conception de notre robot, on va présenter les différentes configurations possibles, on cite entre autres :

## Robot tricycle

Dans ce modèle on trouve, un moteur de propulsion couplé à 2 roues motrices et un moteur de direction couplé à une roue directionnelle. Son centre de masse est le plus près possible de l'axe des roues motrices.

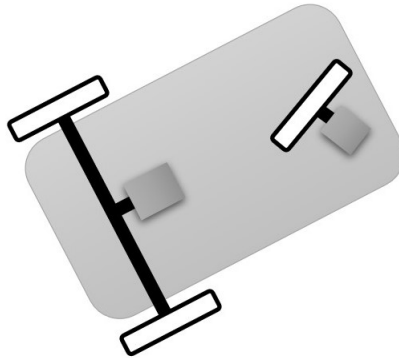


FIGURE 4 – Robot tricycle

## Robot à vitesse différentielles

On peut distinguer deux configurations pour le modèle différentiel, asymétrique avec 2 moteurs de propulsion et de direction couplés à deux roues motrices. Ces dernières sont disposées en avant ou en arrière et une roue libre assure la stabilité est disposée à l'autre extrémité.

Dans le modèle symétrique, les roues motrices sont disposées au centre du robot et 2 roue libre libres assurant la stabilité sont disposées en arrière et en avant. Le centre de masse du premier se trouve le plus près possible de l'axe des roues motrices et pour le deuxième, le plus près du centre du robot.

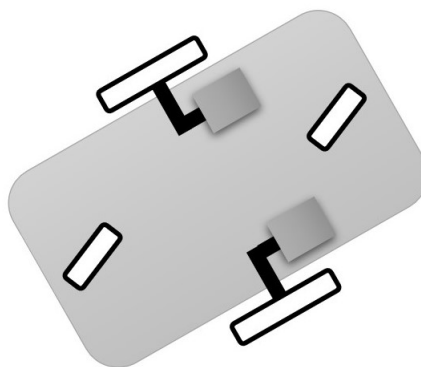


FIGURE 5 – Robot à vitesse différentielles symétrique

## Robot à 4 roues motrices et directionnelles

Avec 4 roues motrices jumelées 2 à 2 (côté gauche ensemble et droite ensemble), 2 moteurs de propulsions (vitesse différentielles), 2 moteurs de direction un pour les roues

avant et un pour les roues arrière. Le centre de masse se trouve le plus près possible du centre du robot.

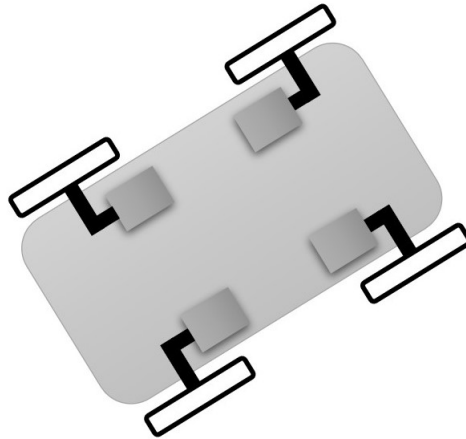


FIGURE 6 – Robot à 4 roues

### Choix de robot

Après la comparaison des différentes configurations possible on choisit le modèle différentiel asymétrique pour les raisons suivantes :

- Faible cout de fabrication : La quantité de matériaux et de composants est minimale.
- Bonne fiabilité mécanique : Le nombre de composantes mécaniques étant réduit.
- Simplicité de contrôle.

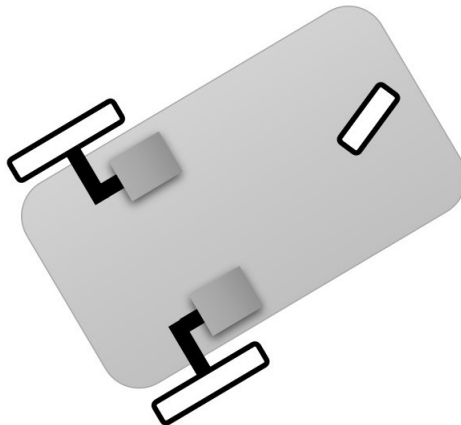


FIGURE 7 – Robot à vitesse différentielles asymétrique

Notre robot à deux roues motrices disposées sur un axe virtuelle commun, chaque roue a un moteur de propulsion indépendant qui peut se tourner dans et contrairement au sens des aiguilles pour produire une translation en avant et en arrière. On peut aussi varier la vitesse de moteur associé à chaque roue pour performer un mouvement roulant ou circulaire.

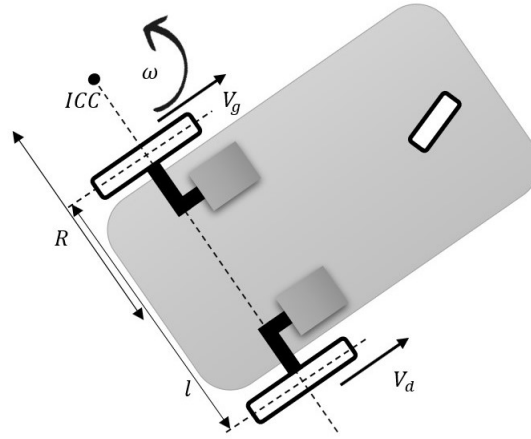


FIGURE 8 – Cinématique du robot

Le point auquel le robot tourne sur est connue par ICC (Instantaneous Center of Curvature). Les équations qui suivent donnent la relation entre les vitesses des roues droite  $V_d$  et gauche  $V_g$  et la vitesse angulaire de rotation sur ICC  $\omega$  :

$$\omega(R + l/2) = V_d \text{ et } \omega(R - l/2) = V_g$$

Où,  $l$  est la distance entre les centres des deux roues,  $R$  est la distance qui sépare l'ICC du point de centre entre les roues. On résout le système pour trouver  $R$  et  $\omega$ , ce qui va donner :

$$R = (V_g + V_d)/2\omega \text{ et } \omega = (V_d - V_g)/l$$

On peut discuter trois cas intéressants :

- Si  $V_g = V_d$ , le mouvement est rectiligne,  $R \rightarrow \infty$  et  $\omega = 0$  donc il n'y a pas de rotation.
- Si  $V_g = -V_d$ ,  $R = 0$  et on a une rotation sur le point centre de l'axe des roues.
- Si  $V_g = 0$  et  $V_d = 0$  donc le robot ne bouge pas.

## 4 Cahier de charge

En se basant sur l'étude effectuée ci-dessus, on peut assumer que pour un bon fonctionnement du robot, ce dernier doit satisfaire les critères suivants :

- Il doit assurer un mouvement de translation et de rotation.
- Il doit se déplacer aisément avant et après la détection d'un obstacle.
- Il doit être très léger, facile à prendre et à manipuler, contrôlable et rigide.
- Son montage et démontage doit se faire d'une manière simple et rapide.
- Il doit avoir plus d'autonomie concernant sa batterie.
- Le positionnement du robot à l'aide de placement de ses senseurs doit être relativement précis.
- Symétrie : ceci constitue le point déterminant de cette structure. La symétrie permet des performances intéressantes en rotation puisque le centre de masse peut être situé au centre du robot, sur l'axe des roues motrices.

Notre robot éviteur d'obstacles est composée de 2 moteurs de propulsion et de direction couplés à deux roues motrices. Les roues motrices sont disposées à l'arrière et une roue libre assurant la stabilité est disposée à l'autre extrémité du robot. Le centre de masse se retrouve le plus près possible de l'axe des roues motrices pour faciliter les différents mouvements. La base du robot tient généralement tous les composants qui font fonctionner le robot. Par exemple, la base comporte des roues motrices, ainsi que les batteries (qui peut être assez lourdes) et le système de contrôle.

**Conclusion.** En plus de la fonction principale de ce robot qui est bien évidemment l'évitement d'obstacles, les différentes études qu'on a menées nous ont aidés à conclure que ce qui fera différence entre un robot et un autre est le temps que mettra chacun d'eux face à un obstacle. Dans cette première partie, une large discussion s'est entretenue entre nous à propos de nombres de roues motrices qu'on peut utiliser, leurs emplacements et aussi même le châssis qui devait les raccompagnés. En effet, il devait être choisi de telles sortes de ne pas ralentir le robot lors de son déplacement sans oublier de tenir compte à sa rigidité et sa ténacité. Jusqu'à maintenant, on est tous convaincus que les choix que nous avons effectués sont les plus convenables en mesure de rapidité et de maintenance. Arrivés à ce stade, il ne nous reste que d'espérer que le choix des matériaux soit aussi décisif qu'on pourra de plus agir sur le temps de réponse de notre robot ...

Deuxième partie

Matériaux et composants

**Introduction.** Le choix des matériaux est une tâche fondamentale et très complexe, et pour réussir à bien l'aborder, on a posé un groupe de critères qu'on devait les satisfaire pour bien réussir notre choix. Les critères imposés sont les suivants :

- Au niveau de matériau : caractéristiques mécaniques, esthétiques, thermiques, électriques, économiques, environnementales, physiques.
- Au niveau du procédé : volume, masse, géométrie, taille de la série, caractéristiques économiques et environnementales.

## 5 Vue globale

L'idée est d'utiliser une carte Arduino pour lire les données reçues par un, ou plusieurs, capteur(s) à ultrasons. Lorsque le capteur détecte un obstacle devant lui, l'Arduino doit envoyer des instructions aux moteurs qui pilotent les roues du robot pour l'éviter. Le robot serait en partie constitué de châssis et de composantes électroniques (Arduino, capteur à ultrasons, moteurs, ...) serait fixé au châssis à l'aide de pièces imprimées en 3D (ou simplement à l'aide d'une colle). Les matériels dont on a besoin pour réaliser ce robot sont :

1. Arduino UNO.
2. Un capteur à ultrasons HC-SR04.
3. Deux moteurs DC pour piloter les roues du robot.
4. Un servo-moteur pour faire pivoter le capteur à ultrasons (la tête du robot).
5. Une pile de 9V et des connecteurs pour alimenter les moteurs DC.
6. Servomoteur
7. Un châssis + roue à bille
8. Des câbles jumper, résistances, LED, une breadboard pour tester le circuit électronique.



FIGURE 9 – Un châssis du robot

## 6 La carte Arduino

Une carte Arduino est une petite carte électronique équipée d'un microcontrôleur. Le microcontrôleur permet, à partir d'événements détectés par des capteurs, de programmer et commander des actionneurs ; la carte Arduino est donc une interface programmable. Elle peut être programmée pour analyser et produire des signaux électriques, de manière à effectuer des tâches très diverses comme le contrôle des appareils domestiques, éclairage, chauffage, le pilotage d'un robot, de l'informatique embarquée, etc.

### 6.1 Choix du type de la carte

Il existe plusieurs types de cartes Arduino, on s'est orienté uniquement vers le type le plus fréquent qui est la carte Arduino UNO. Pour réaliser ce montage, on doit assurer la liaison entre chaque composante électronique et notre carte Arduino.



- Le Shield a besoin de 6 broches numériques dont deux PWM et une GND sans oublier l'alimentation.
- Le capteur ultra son a besoin également à être alimenté, d'une GND, et 2 broches le suffit.
- Le servo moteur a besoin également d'être alimenté puis d'une GND qui prendra seulement une entrée analogique

Donc en totale on aura besoin d'une carte Arduino qui contient au moins 9 broches numérique dont deux PWM, et aussi qui sera capable d'exécuter le plus vite possible l'algorithme gravé dans sa mémoire. En revanche, la carte Arduino UNO est la carte la plus convenable à cette situation vue qu'elle possède 14 broches numériques (dont 6 avec PWM) et 6 broches d'entrée analogique.

La carte UNO ainsi que les autres types cartes Arduino (Micro et Mega) ont chacune leur propre capacité de traitement ; parlons à présent de leurs fréquences/vitesses d'horloge respectives. La fréquence/vitesse d'horloge de ces cartes indique simplement la vitesse avec laquelle elles peuvent exécuter une commande. Les trois cartes possèdent toutes la même vitesse d'horloge, soit 16 MHz.

Au niveau de la mémoire La Uno et la Micro possèdent toutes les deux une mémoire Flash de 32 ko, tandis que la Mega 2560 en propose 256 ko, soit 8 fois plus d'espace mémoire ! La mémoire Flash représente simplement la taille maximale du code ou du modèle que vous pouvez charger sur votre Arduino. Si votre code est lourd, la Mega 2560 est donc la solution idéale.

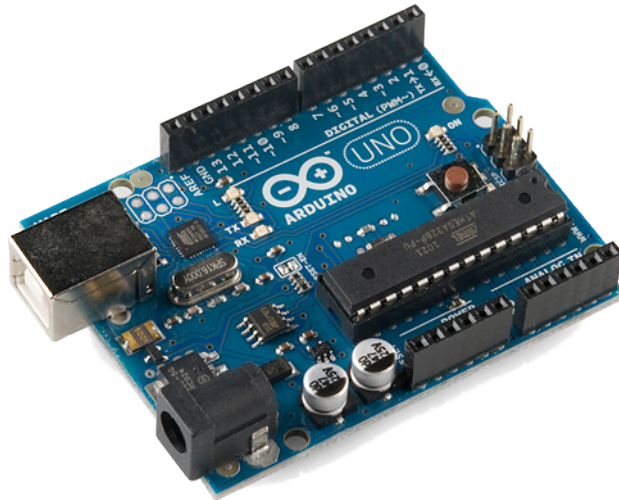


FIGURE 10 – Une carte Arduino UNO

## 6.2 Composantes de la carte Arduino UNO

Une carte Arduino UNO se compose essentiellement de :

- Prise jack : Permet de brancher une alimentation (pile, batterie, adaptateur secteur, + au centre Vin 7 à 12 V).
- Microcontrôleur : stocke le programme et l'exécute.
- Entrées analogiques : Permet de brancher des capteurs et des détecteurs analogiques.

- Connexion USB (Universal Serial Bus) : Permet d'alimenter la carte en énergie électrique (5V). Permet de téléverser le programme dans la carte.
- Entrées et sorties numériques (Digital) : Permet de brancher des actionneurs. Permet de brancher des détecteurs.

## 7 Capteurs ultra-son

### 7.1 Le choix du capteur

L'utilisation d'un capteur est indispensable dans notre projet. En effet, c'est un outil qui permet la détection et par conséquent l'évitement d'obstacles qui est la cible de notre étude. Ceci dit, le problème auxquelles on devait s'imposer était le choix du meilleur capteur à utiliser dans les mesures de concilier le temps de réponse et le coût de production. Après plusieurs comparaisons et distinctions des capteurs, on était convaincus que celui dont on avait besoin était le capteur ultra-son HC-SR04.



FIGURE 11 – Capteur ultra-son HC-SR04

### 7.2 Fonctionnement du capteur

Le capteur ultra-son HC-SR04 est un dispositif chinois qui comme les autres capteurs ultra-son émet à intervalles réguliers de courtes impulsions sonores à haute fréquence, qui à leurs tours se propagent dans l'air à la vitesse du son, et se réfléchissent et reviennent sous forme d'écho lorsqu'elles rencontrent un objet. Puis, ce dernier calcule la distance le séparant de la cible sur la base du temps écoulé entre l'émission du signal et la réception de l'écho.

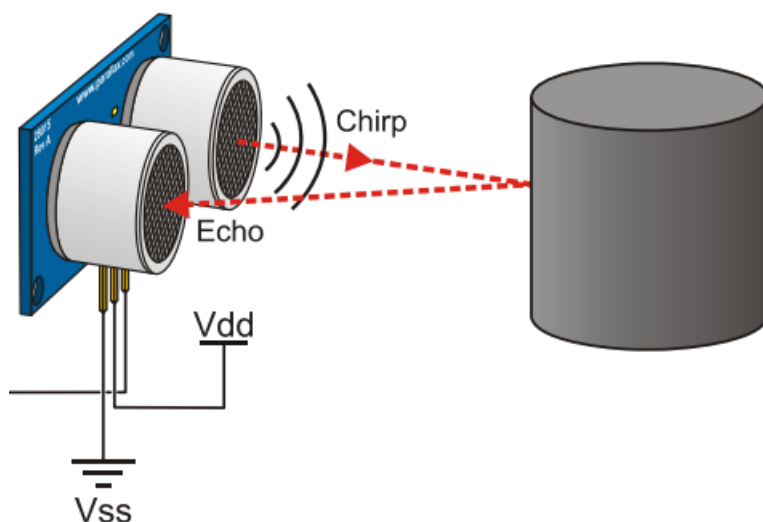


FIGURE 12 – Description du capteur ultra-son HC-SR04

Pour déclencher une mesure, il faut présenter une impulsion "high" (5 V) d'au moins 10  $\mu s$  sur l'entrée "Trig". Le capteur émet alors une série de 8 impulsions ultrasoniques à 40 kHz, puis il attend le signal réfléchi. Lorsque celui-ci est détecté, il envoie un signal "high" sur la sortie "Echo", dont la durée est proportionnelle à la distance mesurée.

Nous utilisons la relation  $d = v.t$  avec  $d$  pour la distance,  $v$  pour la vitesse et  $t$  pour le temps. On connaît la vitesse d'impulsion qui est égale à 340m/s. Le son doit effectuer un aller (en partant du capteur) pour ensuite rebondir sur l'obstacle et revenir à ce point de départ. Nous diviserons donc son temps par deux. Le HC-SR04 donne une durée d'impulsion en dizaines de  $\mu s$ . Il faut donc multiplier la valeur obtenue par 10  $\mu s$  pour obtenir le temps  $t$ . La vitesse du son est égale à environ 340m/s ce qui nous donne :

$$d = \frac{340000cm.10\mu s}{1000000us}.t$$

Donc

$$d = \frac{170cm}{1000}.t$$

La formule  $d = dure/58cm$  figure dans le manuel d'utilisation du HC-SR04 car la fraction 17/1000 est égale à 1/58.8235.

La particularité du capteur chinois HC-SR04 s'incarne au début dans son prix qui n'est vraiment pas cher (30 DHs), puis dans ses caractéristiques qui sont moyennement satisfaisantes. En effet, ce dernier fonctionne avec une tension d'alimentation de 5 volts, dispose d'un angle de mesure de 15° environ et permet de faire des mesures de distance d'un mètre à peu près, une précision de 3mm (en théorie, dans la pratique ce n'est pas tout à fait exact), et c'est ce qui va nous permettre de rendre le projet plus optimal en ajoutant 2 autres d'avoir une vision claire des obstacles et de diminuer le temps de réponses.

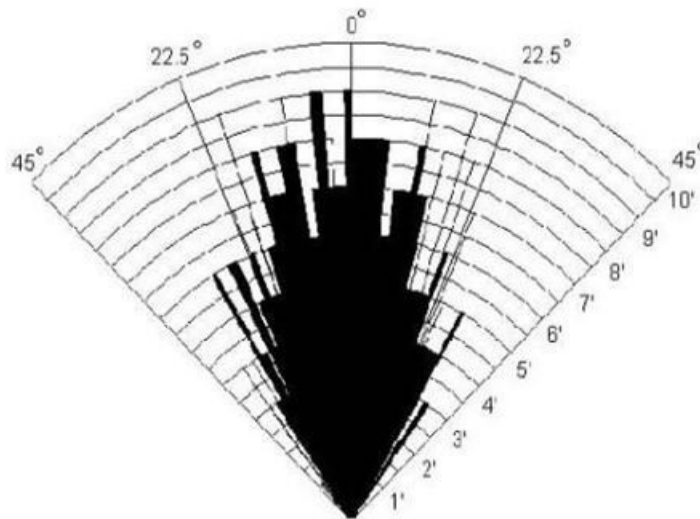


FIGURE 13 – Le champ de fonctionnement du capteur

## 8 Moteur DC

### 8.1 Généralité sur les machines à courant continu

Une machine à courant continu est une machine électrique Inventée par ZÉNOBE GRAMME et présentée à l'Académie des Sciences, à Paris, en 1871, c'était au départ un simple générateur de courant continu (pour applications galvanoplastiques : une technique électrolytique d'orfèvrerie).

Il s'agit d'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.



FIGURE 14 – Moteur à courant continu

- En fonctionnement moteur, l'énergie électrique est transformée en énergie mécanique.
- En fonctionnement générateur, l'énergie mécanique est transformée en énergie électrique (elle peut se comporter comme un frein). Dans ce cas elle est aussi appelée dynamo.

Cependant, la machine à courant continu étant réversible et susceptible de se comporter soit en « moteur » soit en « générateur » dans les quatre quadrants du plan couple-vitesse, la distinction moteur/générateur se fait « communément » par rapport à l'usage final de la machine.

### 8.2 La différence entre Moteur DC et Moteur AC

Alors que les moteurs AC et DC servent la même fonction de convertir l'énergie électrique en énergie mécanique, ils sont alimentés, construits et contrôlés différemment. La source la plus fondamentale est la source d'alimentation. Les moteurs à courant alternatif sont alimentés à partir du courant alternatif (AC), tandis que les moteurs à courant continu sont alimentés par courant continu (DC), tels que des batteries, des alimentations CC ou un convertisseur AC-DC.

Les moteurs de champ à enroulement DC sont construits avec des balais et un collecteur, qui ajoutent à la maintenance, limitent la vitesse et réduisent généralement l'espérance de vie des moteurs à courant continu brossé. Les moteurs à induction à courant alternatif n'utilisent pas de brosses ; Ils sont très robustes et ont de longues espérances de vie. La dernière différence de base est le contrôle de la vitesse. La vitesse d'un moteur à courant continu est contrôlée en faisant varier le courant de l'enroulement de l'induit tandis que la vitesse d'un moteur à courant alternatif est contrôlée en faisant varier la fréquence, ce qui n'est pas évident, généralement effectué avec une commande de commande de fréquence réglable.

### 8.3 Constitution d'un moteur DC

Un moteur électrique à courant continu est constitué :

- D'un stator qui est à l'origine de la circulation d'un flux magnétique longitudinal fixe créer soit par des enroulements statoriques (bobinage) soit par des aimants permanents.
- D'un rotor bobiné relié à un collecteur rotatif inversant la polarité de chaque enroulement rotorique au moins une fois par tour de façon à faire circuler un flux magnétique transversal en quadrature avec le flux statorique.

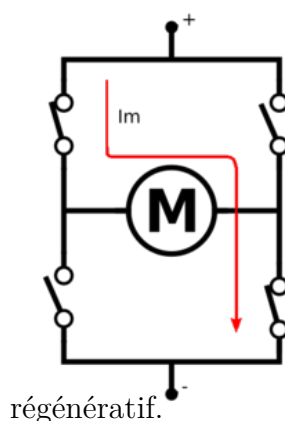
### 8.4 Fonctionnement d'un moteur DC

**Partie Théorique** Les moteurs DC avec des aimants permanents sont très courants dans différentes applications, avec de petites dimensions, lorsqu'on recherche beaucoup de puissance à bas prix. A cause de leur vitesse assez importante, ils sont utilisés dans plusieurs sortes de transmission (pour obtenir beaucoup de couple avec peu de vitesse).

Les moteurs DC à aimant permanent sont d'une fabrication assez simple et les commander est assez élémentaire. Bien que les contrôler soit assez simple, leur vitesse n'est pas déterminée précisément par le signal de contrôle parce qu'il dépend de nombreux autres facteurs. La relation entre le couple et la vitesse d'un moteur DC parfait est linéaire, ce qui signifie que : plus grand est l'effort sur l'arbre, plus petite est la vitesse de l'arbre et plus grand est le courant dans la bobine.

Les moteurs DC utilisent la tension DC et ne nécessitent pas de contrôle électronique supplémentaire puisque toutes les communications nécessaires sont faites dans le moteur. Lorsque le moteur fonctionne, deux broches statiques glissent dans le commutateur de rotation et garde la tension dans la bobine. La direction de la rotation du moteur est déterminée par la polarité du courant appliqué. Si le moteur ne doit tourner que dans une seule direction, alors le courant peut passer à travers un relai ou tout autre connexion simple. Si le moteur doit tourner dans plusieurs directions, on utilisera un circuit électronique appelé Pont en H (ou "H-bridge").

#### Utilisation de Pont en H avec les moteurs DC



régénératif.

Le pont en H permet de réaliser 2 fonctions qui sont d'inverser le sens de rotation du moteur en inversant le courant aux bornes du moteur et la variation de la vitesse du moteur en modulant la tension aux bornes du moteur.

De plus, le pont en H permet d'effectuer un freinage magnétique s'il est capable d'en dissiper la puissance générée. Cette opération s'effectue en actionnant soit les deux commutateurs supérieurs ou inférieurs en même temps, ce qui court-circuite les bornes du moteur, et le fait par conséquent freiner. Mieux encore, il est possible avec un peu d'électronique et un contrôleur perfectionné d'effectuer un freinage

FIGURE 15 – Pont en H avec une branche active

**Partie pratique** On utilise le pont en activant les commutateurs de différentes combinaisons pour obtenir le branchement voulu. Le tableau suivant résume les combinaisons permises. Toutes les combinaisons qui ne figurent pas dans le tableau sont interdites et créent un court-circuit de la source. Le courant de référence pour la charge est considéré comme étant de gauche à droite.

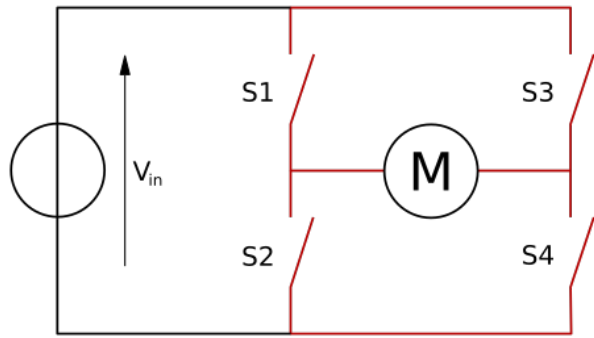


FIGURE 16 – Schéma d'un Pont en H

Le tableau correspond à ce circuit :

Combinaisons d'états des commutateurs				
État des commutateurs				Résultat à la charge
S1	S2	S3	S4	
0	0	0	0	Aucune tension aux bornes de la charge
1	0	0	1	Courant positif à travers la charge
0	1	1	0	Courant négatif à travers la charge
1	1	0	0	Charge court-circuitée
0	0	1	1	Charge court-circuitée

## Simulation d'un pont en H es ISIS

Quand les deux boutons sont relâchés le moteur ne tourne pas car les transistors sont en état bloquant donc le courant électrique ne traverse pas le moteur. Quand on clique sur un de deux et on lâche l'autre (par exemple le bouton 1) la base du transistor reçoit un courant grâce auquel les transistors  $Q_1$  et  $Q_4$  laissent passer le courant. Puisque les autres transistors sont en état bloquant l'émetteur respectivement le collecteur du  $Q_3$  et  $Q_2$  sont en niveau bas (0V) et en niveau haut. La différence de potentiel entre le moteur laisse passer le courant de  $Q_1$  à  $Q_4$  ; le moteur tourne dans un sens (il tourne dans l'autre sens si les rôles des boutons sont inversés). Quand on clique sur les deux boutons les bornes du moteur sont en niveau haut (pas de différence de potentiel) donc le courant ne traverse pas le moteur par la suite il ne tourne pas.

## 9 Servo-moteur

Un servomoteur est un système asservi (ou système bouclé) ; il s'agit d'un moteur à courant continu enfermé dans un boîtier de petite taille avec des constituants électroniques et mécanique. Sa fonctionnalité est de tenir une position angulaire donnée fournie au

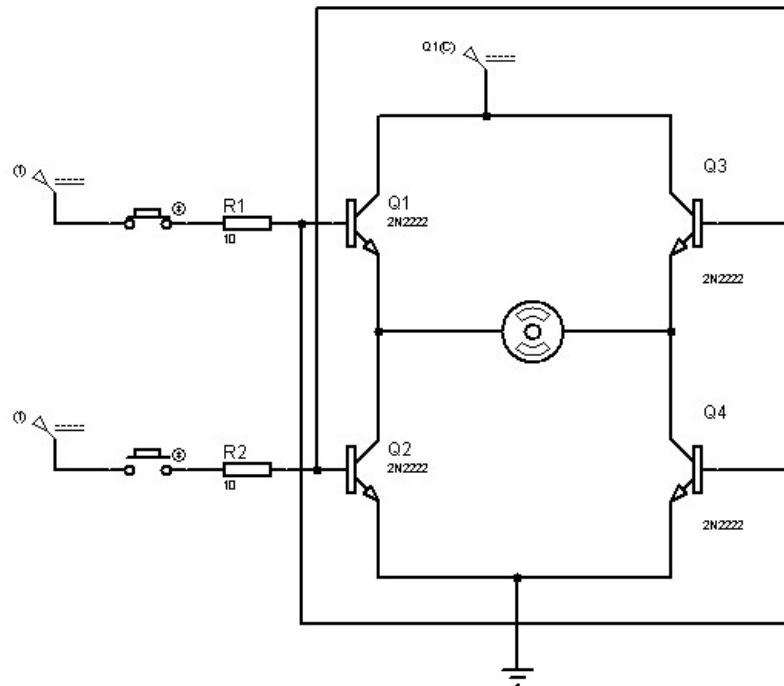


FIGURE 17 – simulation d’un pont en H réaliser par Proteus à base des transistors

système en entrée sous forme d’un signal électrique. On en trouve de toutes les tailles et de toutes les puissances. La plupart du temps la sortie peut se positionner entre 0 et 180°. Il en existe également dont la sortie peut se débattre sur seulement 90° et d’autres, ayant un plus grand débattement, sur 360°. Ceux qui ont la possibilité de faire plusieurs tours sont souvent appelés servo-treuil. Enfin, les derniers, qui peuvent faire tourner leur axe sans jamais se buter, sont appelés servomoteurs à rotation continue. Les servomoteurs sont très fréquemment employés dans les applications de modélisme pour piloter le safran d’un bateau, le gouvernail d’un avion ou bien même les roues d’une voiture téléguidée.

## 9.1 Composition d’un servomoteur

Le servomoteur est composé de plusieurs éléments visibles et invisible :

- Un moteur à courant continu
- Des engrenages pour former un réducteur (en plastique ou en métal)
- Un capteur de position de l’angle d’orientation de l’axe (un potentiomètre)
- Une carte électronique pour le contrôle de la position de l’axe et le pilotage du moteur à courant continu
- Les fils, qui sont au nombre de trois
- L’axe de rotation sur lequel est monté un accessoire en plastique ou en métal
- Le boîtier qui le protège

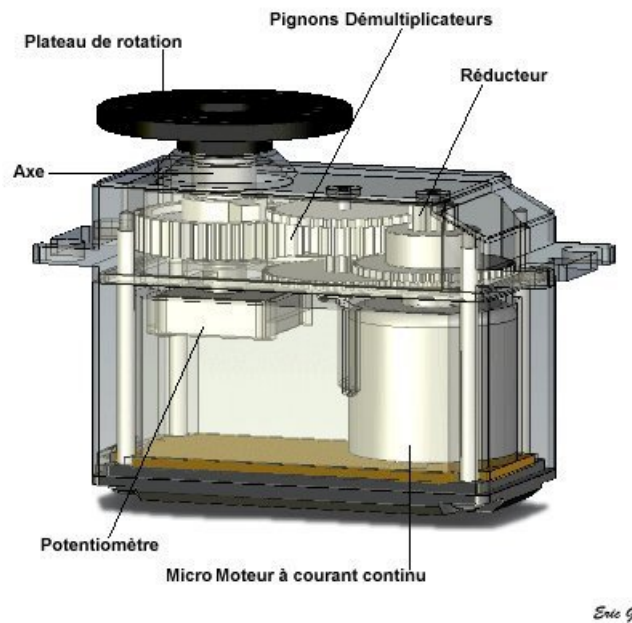


FIGURE 18 – Vue interne d'un servomoteur

### Principe de fonctionnement d'un servomoteur

La plupart des servomoteurs sont commandés par l'intermédiaire d'un câble électrique à trois fils qui permet d'alimenter le moteur et de lui transmettre des consignes de position sous forme d'un signal codé en largeur d'impulsion plus communément appelé PWM. Cela signifie que c'est la durée des impulsions qui détermine l'angle absolu de l'axe de sortie et donc la position du bras de commande du servomoteur. Le signal est répété périodiquement, en général toutes les 20 millisecondes, ce qui permet à l'électronique de contrôler et de corriger continuellement la position angulaire de l'axe de sortie, cette dernière étant mesurée par le potentiomètre.

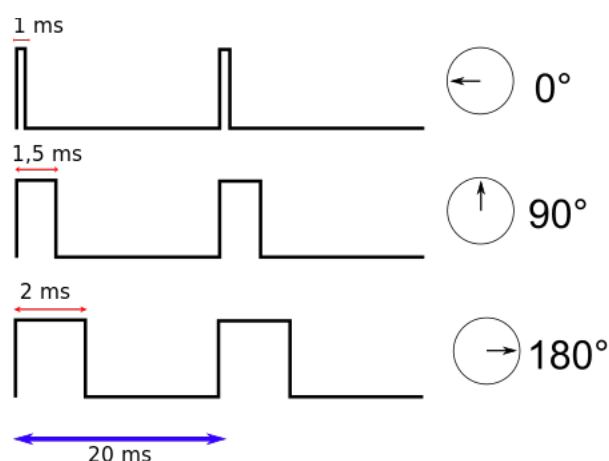


FIGURE 19 – simulation d'un pont en H réaliser par Proteus à base des transistors



## 9.2 Commande d'un servomoteur à base de NE555

On cherche dans cette sous partie de savoir comment piloter la position angulaire de l'axe d'un servomoteur par un potentiomètre, en utilisons quelques composants électroniques simples et courants. On réalise le schéma suivant ou le NE555 génère une impulsion périodique, on varie la valeur du potentiomètre pour obtenir la commande en position du servomoteur :

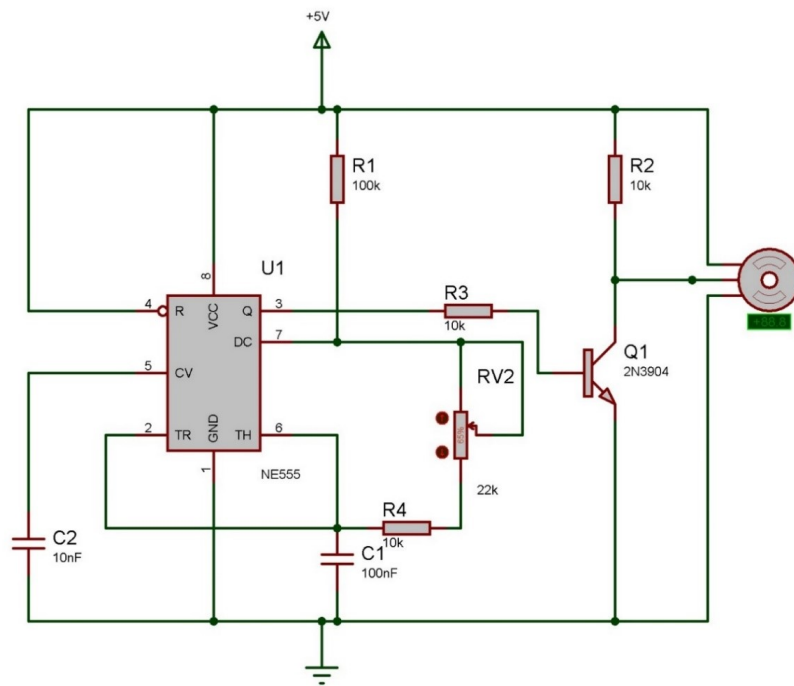
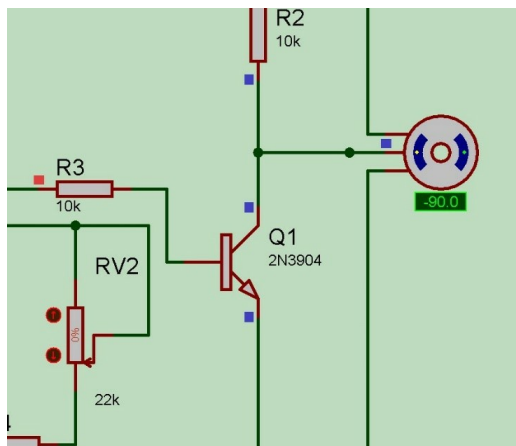
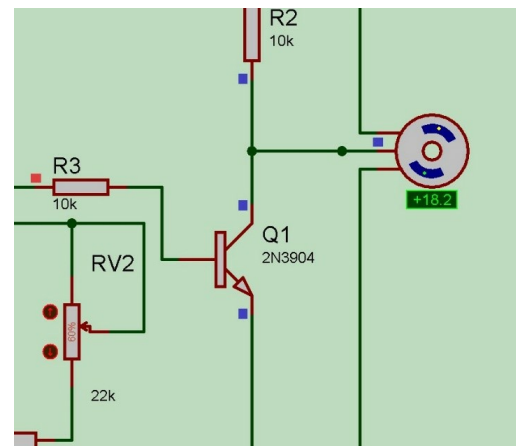


FIGURE 20 – Schéma de commande d'un servo-moteur à base d'un NE555

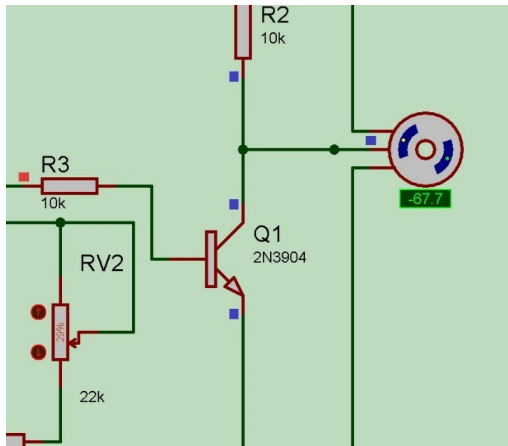
### Résultats



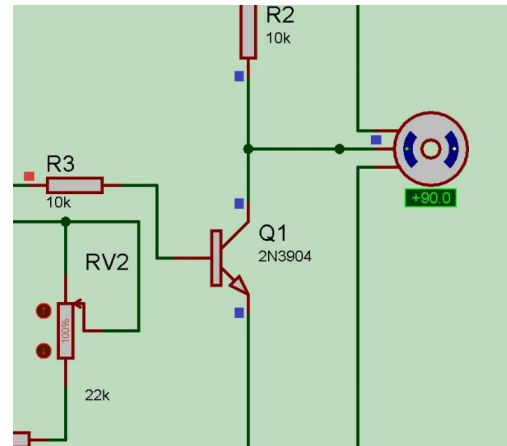
Le potentiomètre en 0% et le servomoteur indique -90



Le potentiomètre en 80% et le servomoteur indique +18



Le potentiomètre en 20% et le servomoteur indique  $-67$



Le potentiomètre en 100% et le servomoteur indique  $+90$

**Conclusion.** Le choix des matériaux était la phase la plus délicate dans notre projet. En effet, il ne portait pas seulement sur un aspect purement technique répondant à des exigences fonctionnelles, mais aussi à des attentes relevant des préférences des utilisateurs dans le cadre d'un marché spécifique. En gros, le choix des matériaux a été analysé sous l'angle de l'ingénierie des matériaux, du temps de réponse, sans oublier enfin le côté esthétique (le design industriel).

## Troisième partie

### Simulations

**Introduction.** La simulation de notre robot était la dernière tâche qu'on devait réaliser, les études ont déjà été terminées et les matériaux déjà mis en place, il ne manquait qu'un châssis pour assembler ces dispositifs puis vérifier si les mesures qu'on a pris nous ont emmenés à concilier entre le côté technique et esthétique.

## 10 Schéma du robot

Voici le schéma de notre robot, on a abouti ce schéma en utilisant le logiciel fritzing.

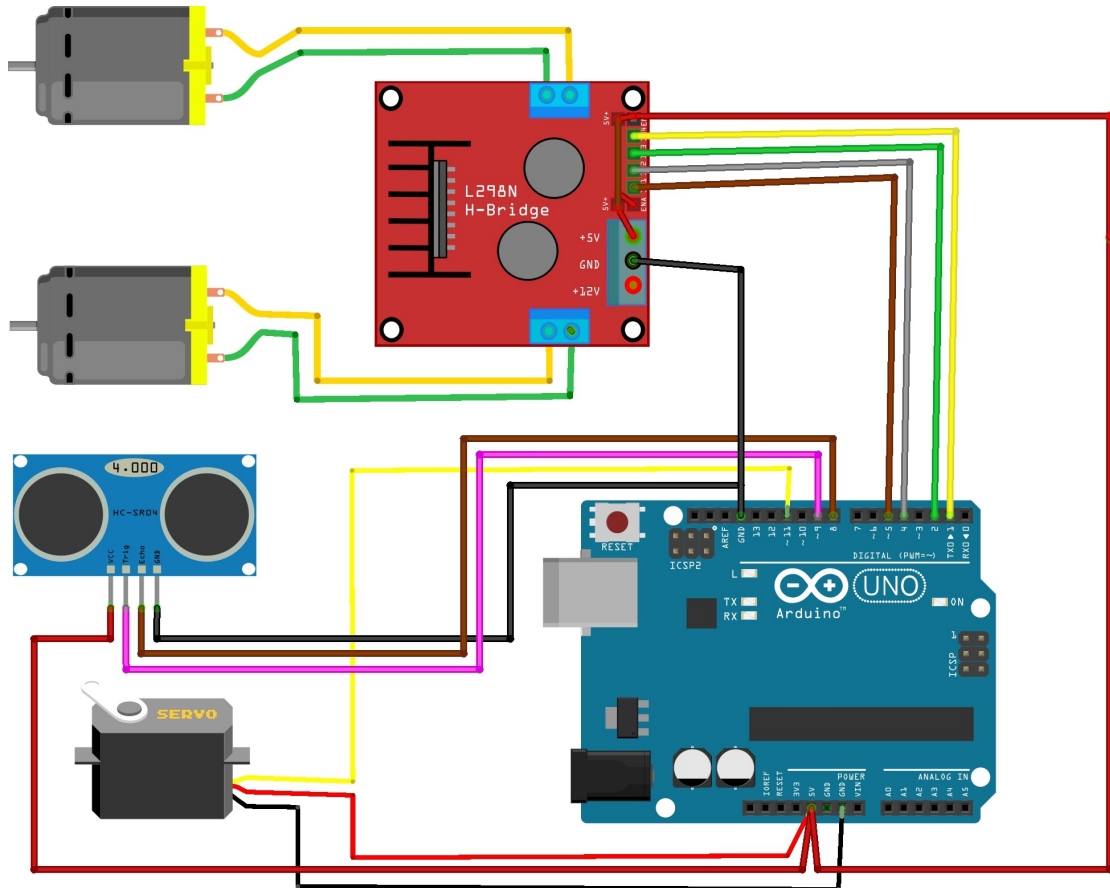


FIGURE 21 – schéma électrique du robot

## 11 Simulation des différents circuits

### 11.1 Premiers pas : initialisation en Arduino (Allumer une LED)

Afin de bien maîtriser la programmation et l'utilisation de la carte Arduino, on va tout d'abord commencer par des programmes simples dont le but est de réaliser un programme optimal pour notre robot.

Commençons au début par un programme assez simple, dans lequel on cherche à allumer puis éteindre une LED pendant une seconde infiniment de fois. Le schéma réaliser sur ISIS :

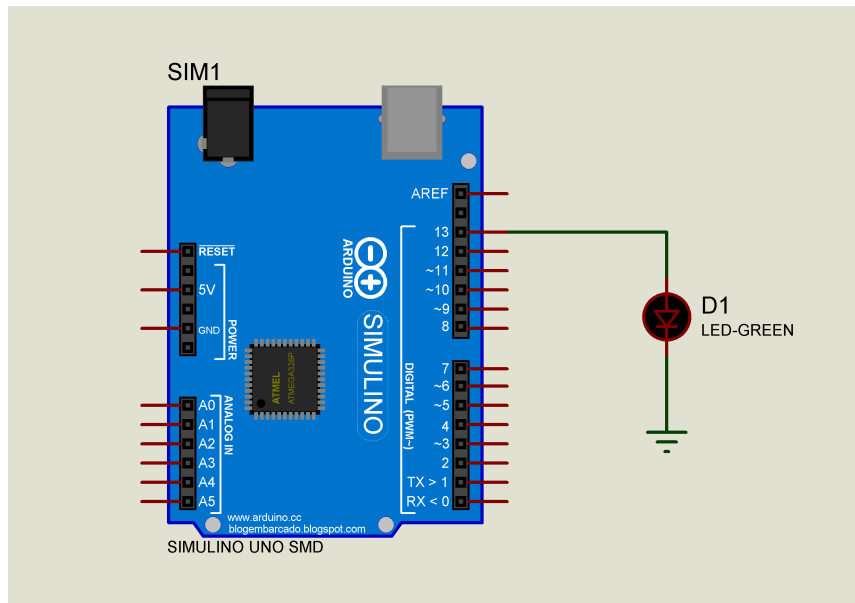


FIGURE 22 – Montage reliant une LED et un Arduino

**Le code Arduino utilisé :**

```
/* Arduino 1 :
Allumer une LED pour une seconde et l'éteindre pour une seconde */

void setup() {
// initialisation du digital pin 13 comme une sortie.
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // allumer la LED
  delay(1000);            // attend d'une seconde
  digitalWrite(13, LOW);  // atteind la LED
  delay(1000);            // attend d'une seconde
}
```

## 11.2 Deuxième pas : utiliser un servomoteur

Le schéma réaliser sur ISIS

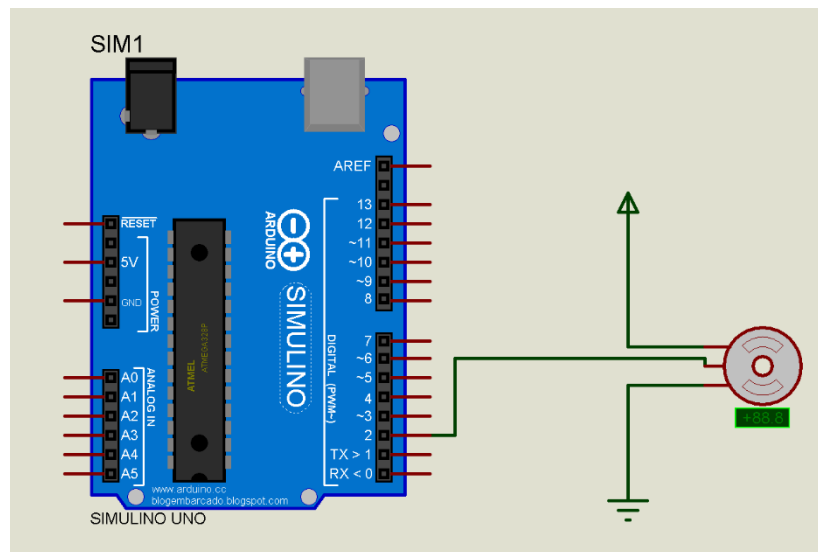


FIGURE 23 – Montage reliant un servo-moteur et un Arduino

Le code Arduino utilisé :

```
//mettre l'angle du bras de votre servomoteur à 90°
#include <Servo.h>

Servo monServo; // création d'un nouveau servomoteur

void setup()
{
    monServo.attach(2, 1000, 2000);
    monServo.write(90);
}

void loop()
{
}
```

### Commentaires

- On a utilisé la fonction `monServo.attach(2, 1000, 2000)` pour Attacher le servo moteur au "pin" 2 et fixer l'angle minimal et maximal du tournage respectivement sur les angles correspondants aux valeurs 1000 et 2000 (la durée de la pulsation en microsecondes).
- La fonction `monServo.write(90)` sert à Faire tourner le bras du servo moteur pour une angle de 90

## 11.3 Troisième pas : utiliser un capteur Ultra-son

### Le schéma réaliser sur ISIS

Afin de mettre en évidence le fonctionnement du capteur ultra-son, on réalise la simulation suivante sur Proteus, on relie le capteur a la carte Arduino avec un écran d'affichage où on va lire les résultats indiqués par le capteur, on modélise un obstacle dans notre simulation par un potentiomètre.

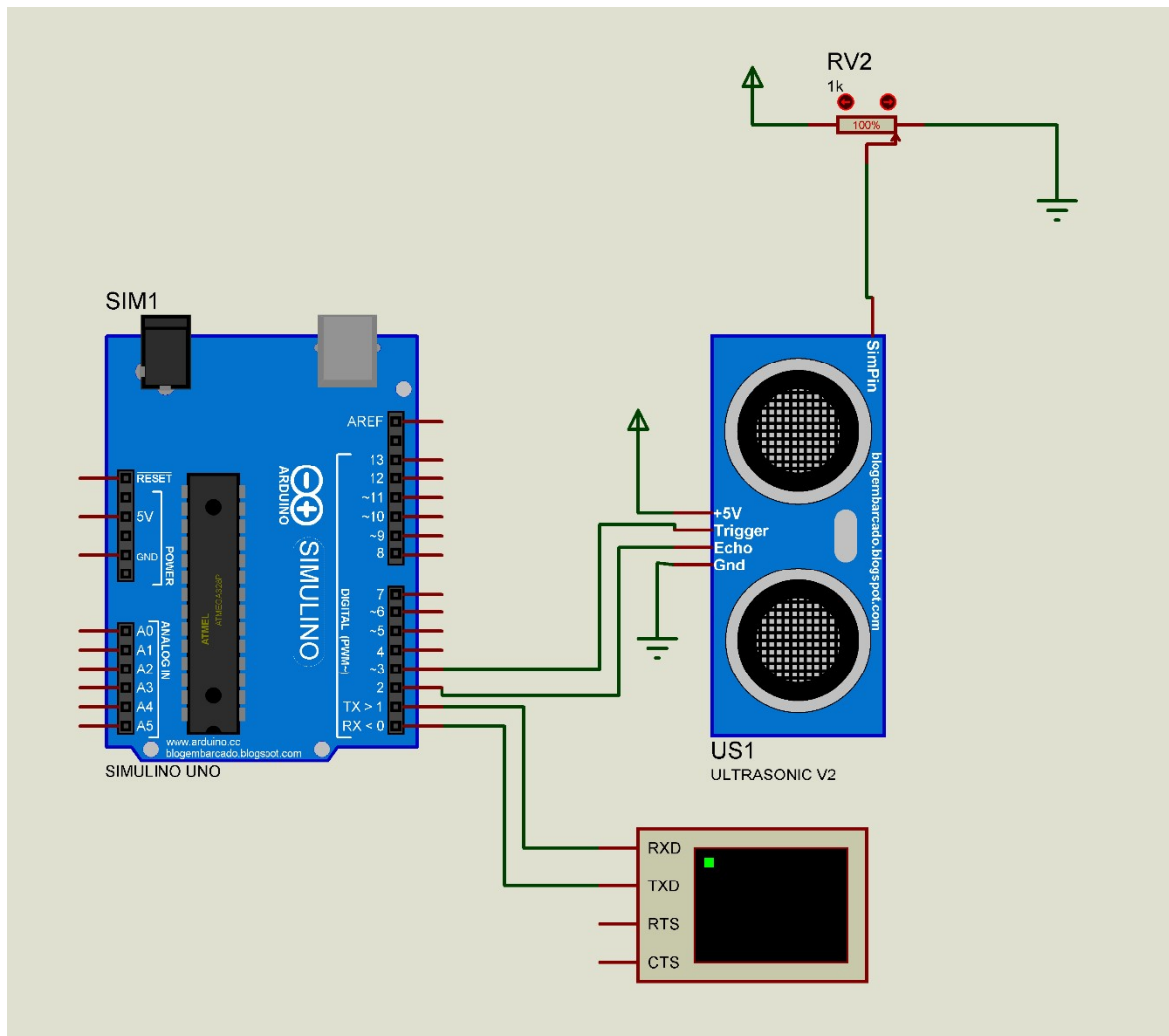


FIGURE 24 – Montage reliant un capteur ultra-son et un Arduino

### Le code Arduino utilisé :

On injecte dans la carte Arduino le code ci-dessus et on lance la simulation, les résultats sont donnés dans les pages suivantes :

```

/* Arduino 2: Mesurer une distance à l'aide d'un capteur Ultra-son */

const int echoPin = 2;
const int pingPin = 3;

void setup()
{
    Serial.begin(9600);
    pinMode(pingPin, OUTPUT);
    pinMode(echoPin, INPUT);
}

void loop()
{
    long duration, cm;

    digitalWrite(pingPin, LOW);
    delayMicroseconds(2);

    digitalWrite(pingPin, HIGH);
    delayMicroseconds(10);

    digitalWrite(pingPin, LOW);

    duration = pulseIn(echoPin, HIGH);
    cm = microsecondsToCentimeters(duration);

    Serial.print(cm);
    Serial.print("cm");
    Serial.println();

    delay(100);
}

long microsecondsToCentimeters(long microseconds)
{
    return (360*microseconds)/ 2*10^4;
}

```



## Éclaircissements et Commentaires

- L'instruction `const int echoPin = 2` sert à attacher la constante `echoPin` au port 2 de la carte Arduino.
- L'instruction `const int pingPin = 3` attache la constante `pingPin` au port 3 de la carte Arduino.
- L'instruction `Serial.begin(9600)` ouvre le port "serial" qui affiche sur l'écran des données et fixe le taux de transmission des données sur 9600 bits par seconde.
- L'instruction `pinMode(pingPin, OUTPUT)` sert à fixer le port 3 (`pingPin`) pour être une sortie.
- L'instruction `pinMode(echoPin, INPUT)` sert à fixer le port 2 (`echoPin`) pour être une entrée.
- L'instruction `digitalWrite(pingPin, LOW)` sert à mettre le port 3 (`pingPin`) au niveau bas (0V).
- L'instruction `delayMicroseconds(2)` sert à arrêter temporairement l'exécution d'algorithmes pour une durée de 2 microsecondes.
- L'instruction `digitalWrite(pingPin, HIGH)` sert à mettre le port 3 (`pingPin`) au niveau haut (5V).
- L'instruction `delayMicroseconds(10)` sert à arrêter temporairement l'exécution d'algorithmes pour une durée de 10 microsecondes.
- L'instruction `digitalWrite(pingPin, LOW)` sert à remettre le port 3 (`pingPin`) au niveau bas (0V).
- L'instruction `duration = pulseIn(echoPin, HIGH)` indique que la variable `duration` reçoit le temps pendant lequel le port `echoPin` est en niveau haut.
- La fonction `long microsecondsToCentimeters` a pour but de donner la distance en cm qui sépare le capteur ultrason de l'obstacle.
- La distance est la vitesse du son dans l'air multipliée par la durée qu'il passe en l'y traversant aller-retour d'où le retour de la fonction.
- Les deux instructions `Serial.print(cm)` et `Serial.print("cm")` servent à afficher la distance en cm qui sépare le capteur ultrason et l'obstacle.
- L'instruction `Serial.println()` sert à afficher le nombre de bites écrit.
- L'instruction `delay(100)` réalise une pause dans l'exécution du programme pour la durée.

## Résultats

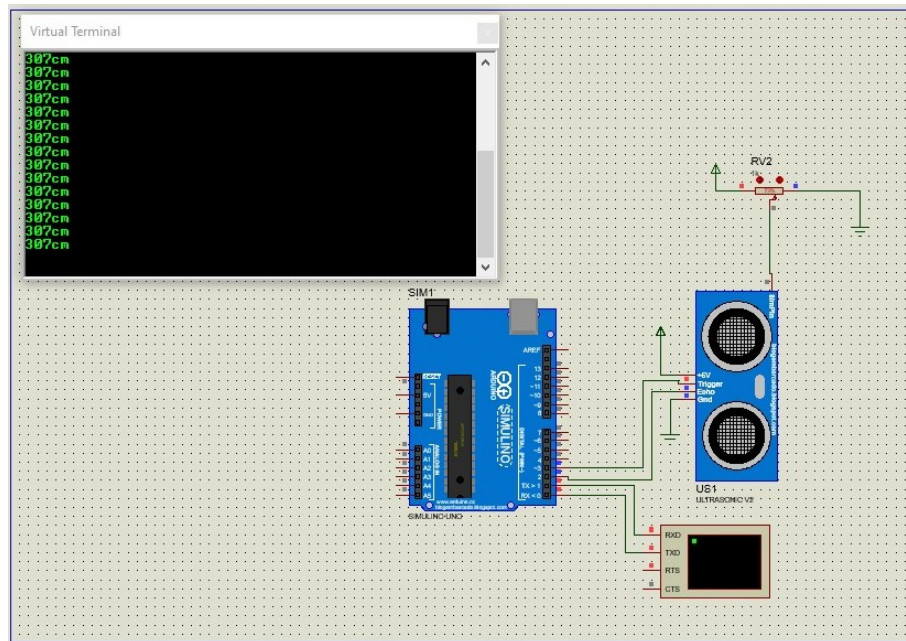


FIGURE 25 – Première réponse du capteur ultra-son

Le potentiomètre en 77%, et le capteur indique 307cm, on change le pourcentage du potentiomètre à 20% et on remarque que le capteur indique 852cm, ce qu'on peut l'interpréter comme l'éloignement d'obstacle.

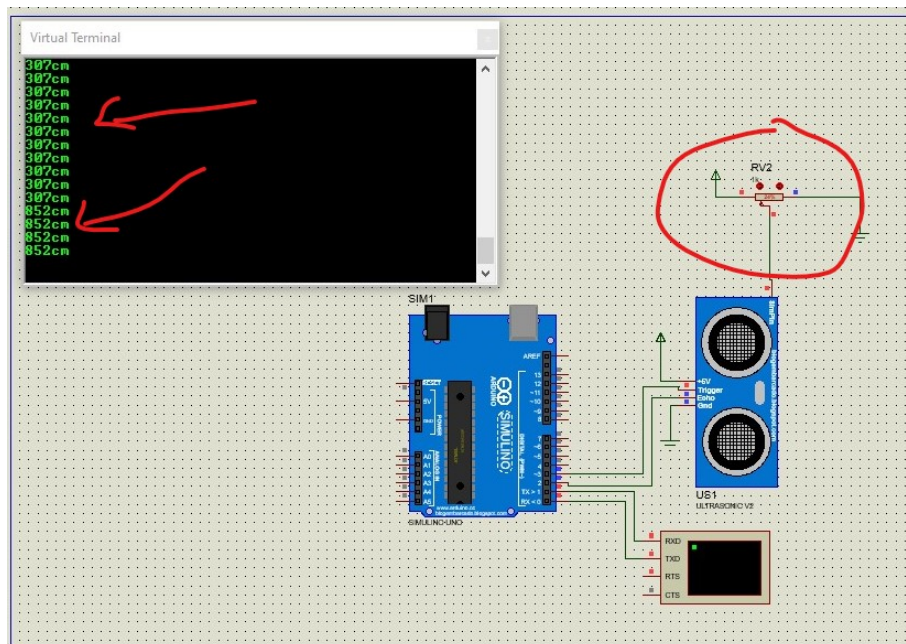


FIGURE 26 – deuxième réponse du capteur ultra-son

## 11.4 Dernier pas : simulation du robot

Après avoir étudié séparément et choisi minutieusement les composants contribuant dans la construction de notre robot éviteur d'obstacle, on les a assimilés tous ensemble et voilà ce qu'on a obtenu :

### Schéma électrique du Robot

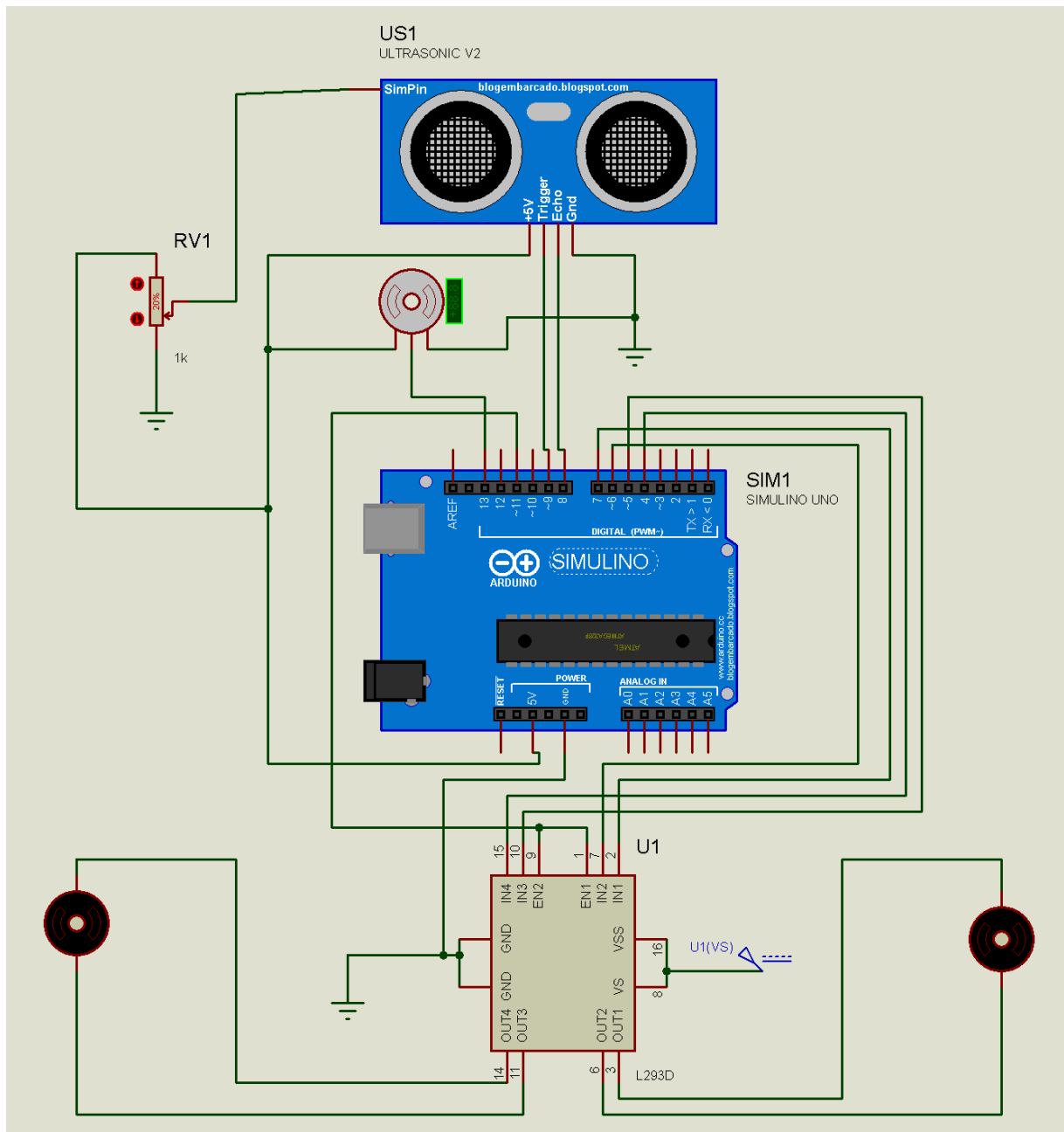


FIGURE 27 – Montage reliant les différents composants du robot

vous trouvez Le code Arduino utilisé pour programmer le robot tout entier dans l'annexe.

**Conclusion.** La simulation est la dernière phase qu'on a traité, dans cette partie la programmation de l'ARDUINO était la tâche la plus délicate, surtout pour des personnes qui utilisent pour la première fois une carte ARDUINO, donc on devaient tout d'abord se familiariser avec ce programme pour arriver enfin a programmer notre robot éviteur d'obstacle.

## Conclusion générale

En guise de conclusion, nous avons pu constater par le biais de notre étude, qu'il existe un critère fondamental distinguant 2 robots de même nature : leurs temps de réaction face à chaque obstacle rencontré. Cette faculté dépend évidemment d'autres facteurs comme la rigidité et la ténacité propre au robot. Toutefois, assurer cette aptitude nécessite une bonne maîtrise de programmation mais avant tout un choix minutieux des matériaux. Ce choix repose non seulement sur l'aspect technique des exigences fonctionnelles du robot mais aussi sur son aspect esthétique dépendant des préférences de l'utilisateur, dont les décisions sont conditionnées par les offres du marché mondial et des avancées technologiques.

En somme, on voit que les choix qu'on a opté pour ont été les plus adéquates pour la conception de ce robot éviteur d'obstacle, soit en essayant d'améliorer son temps de réponse, ou tout simplement en le programmant pour qu'il réalise la tâche demandée qui est l'évitement d'obstacle. Toutefois, et malgré notre grande attention pour le choix des constituants de ce robot, **on ne peut rien garantir au niveau de performances tant qu'on n'a pas passé à la phase de réalisation.** Cette phase qu'on n'a pas pu aborder à cause de cette pandémie que connaît le monde actuellement, et vu la rapide propagation du virus Covid-19 dans notre planète, en effet, on était tous forcés de faire le travail depuis nos domiciles donc en absence tout d'abord de nous comme groupe qui va s'entraider pour réaliser ce robot et puis des matériaux et outils qui vont constituer notre robot.

Aujourd'hui, beaucoup d'états comme la Chine et les États-Unis investissent des milliards de dollars pour le convoi du développement exponentiel des machines intelligentes comme dans le domaine militaire où on entend parler des drones qui s'avèrent être très efficaces, ou dans le domaine industriel où on utilise de nos jours les robots de soudure et de peinture notamment dans l'automobile. Arrivé à ce stade de développement, beaucoup de gens commencent de nos jours à s'inquiéter de ces robots performants, rapides et infatigables et se demandent : Sommes-nous donc tous voués à être remplacés et finir sur la paille ? Et pour demain, quelle sera la place de l'humain ?

Des questions qui nous mènent à revoir les enjeux et les risques qui accompagnent ces machines intelligentes et savoir si ces robots vont constituer vraiment l'avenir de l'être humain ou sa fin dans cette planète.

# Bibliographie

- [1] <https://wiki.mdl29.net/doku.php?id=projets:robotarduino>
- [2] <http://nononux.free.fr/index.php?page=elec-brico-testeur-servomoteur>
- [3] <http://eskimon.fr/tuto-arduino-602-un-moteur-qui-a-de-la-t%C3%A4te-le-servomoteur>
- [4] <https://openclassrooms.com/fr/courses/5224916-developpez-un-robot-mobile-connecte-par-bluetooth>
- [5] <https://openclassrooms.com/fr/courses/2778161-programmez-vos-premiers-montages-avec-arduino>
- [6] <http://dspace.univ-tlemcen.dz/bitstream/112/11642/1/Ms.ELN.Adouna%2BYousfi.pdf>
- [7] [https://www.researchgate.net/publication/220143477\\_Holonomy\\_in\\_mobile\\_robots](https://www.researchgate.net/publication/220143477_Holonomy_in_mobile_robots)
- [8] <http://maximefarin.weebly.com/robot-eacuteviteur-dobstacles.html>
- [9] <https://gist.github.com/ypelletier/6586591>
- [10] [https://www.academia.edu/36200606/ROBOT\\_EVITEUR\\_DOBSTACLES\\_Ce\\_Projet\\_est\\_R%C3%A9alis%C3%A9\\_par](https://www.academia.edu/36200606/ROBOT_EVITEUR_DOBSTACLES_Ce_Projet_est_R%C3%A9alis%C3%A9_par)

# Annexe

Le code Arduino du Robot éviteur d'obstacle :

```
/* Eviteur d'obstacles */

#include <Servo.h>

#define motorPin1a 6 // Marche avant du premier moteur
#define motorPin1b 7 // Marche arrière du premier moteur
#define speedPin1 11 // L293D enable pin pour le premier moteur
#define motorPin2a 4 // Marche avant du deuxième moteur
#define motorPin2b 5 // Marche arrière du deuxième moteur
#define speedPin2 10 // L293D enable pin pour le deuxième moteur

int trigPin = 9; // capteur à ultrasons:
int echoPin = 8;

int Mspeed = 0; // a variable to hold the current speed value
int seuil = 30; // dis min pour laquelle on accepte un obstacle

long distance, distance_gauche, distance_droite;
boolean sensRotation;

Servo monservo; // crée l'objet pour contrôler le servomoteur

void look (void){ // évaluation de la distance de l'obstacle

    long temps;

    //Nous envoyons un signal haut d'une durée de 10 ms sentre deux
    //signaux bas.ultrasons sont émis pendant que le signal est haut

    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    // Lors de la réception de l'écho, le module HC-SR04 émet
```

```

// un signal logique haut (5 v) dont la durée est égale au
// temps écoulé entre l'émission et la réception de l'ultrason.
pinMode(echoPin, INPUT);
temps = pulseIn(echoPin, HIGH);

distance = temps * 340/(2*10000);

}

void setup() {
  pinMode(motorPin1a, OUTPUT); // réglage des broches à output
  pinMode(motorPin1b, OUTPUT);
  pinMode(speedPin1, OUTPUT);
  pinMode(motorPin2a, OUTPUT);
  pinMode(motorPin2b, OUTPUT);
  pinMode(speedPin2, OUTPUT);

  pinMode(echoPin, INPUT);
  pinMode(trigPin, OUTPUT);
  digitalWrite(trigPin, LOW);

  monservo.attach(13);
  monservo.write(70); //positionne le servomoteur en fait à 90°

  // on choisit aléatoirement le sens de la rotation
  look ();
  randomSeed(distance);
  sensRotation = random(2);
}

void droite(){
  analogWrite(speedPin1, Mspeed);
  digitalWrite(motorPin1a, !(sensRotation));
  digitalWrite(motorPin1b, sensRotation);
  analogWrite(speedPin2, Mspeed);
  digitalWrite(motorPin2a, sensRotation);
  digitalWrite(motorPin2b, !(sensRotation));
  delay(100);
}

void gauche(){
  analogWrite(speedPin1, Mspeed);
  digitalWrite(motorPin1a, sensRotation);
  digitalWrite(motorPin1b, !(sensRotation));

```



```

    analogWrite(speedPin2, Mspeed);
    digitalWrite(motorPin2a, !(sensRotation));
    digitalWrite(motorPin2b, sensRotation);
    delay(100);
}

void avant(){
    analogWrite(speedPin1, Mspeed);
    digitalWrite(motorPin1a, HIGH);
    digitalWrite(motorPin1b, LOW);
    analogWrite(speedPin2, Mspeed);
    digitalWrite(motorPin2a, HIGH);
    digitalWrite(motorPin2b, LOW);
}

void loop() {
    monservo.write(70);
    delay(1000);
    Mspeed = 700; // vitesse du moteur 0 à 1023
    look(); // y a-t-il un obstacle devant?

    if (distance > seuil){
        avant(); // marche avant
    }

    else { // on a détecté un obstacle
        look(); // nouvelle vérification pour éviter les faux positifs

        if (distance <= seuil)// obstacle confirmé,
        {
            monservo.write(30);
            delay(1000);
            look();
            distance_droite = distance;

            monservo.write(120);
            delay(1000);
            look();
            distance_gauche = distance;

            if (distance_droite < distance_gauche)
            {
                gauche();// Robot tourne 90° à gauche
            }
        }
    }
}

```

```
if (distance_droite > distance_gauche)
{
    droite(); // Robot tourne 90° à droite
}

if (distance_droite == distance_gauche)
{
    gauche();// Robot demi-tour
    gauche();
}
}
}
```