

21/07/2019

# Rapport de mini projet

Réaliser par :

Ziane khouloud

Ndiaye serigne mourtalla

Ait Taleb Meriem

## TABLE DES MATIERES

Index des figures 1

Introduction générale 01

### CHAPITRE 01 : généralité sur la robotique mobile

1. Introduction.....	03
2. Généralité sur la robotique mobile.....	03
2.1 Définition.....	03
2.2 robot mobile.....	03
2.2.1 Architecture d'un robot mobile.....	03
3. conclusion.....	04

### CHAPITRE 02 : Réalisation pratique d'un robot mobile éviteur d'obstacles

1. Introduction.....	04
2. Problématique.....	04
3. Cahier de charge.....	04
4. Schéma électronique.....	05
5. Partie de commande.....	05
5.1 les capteurs.....	05
5.1.1 capteur à ultrason.....	07
5.2 les microcontrôleurs.....	07
5.2.1 La carte arduino.....	07
5.3 Alimentation.....	08
6. partie mécanique.....	08
6.1 types de moteurs utilisés.....	08
6.1.1 moteur dc.....	08
7. partie logiciel et programmation.....	09
7.1 logiciel de programmation arduino.....	09
8. organigramme principal du mouvement de notre robot.....	09
8.1 Sous-programme détection d'obstacle.....	10
8.2 Sous-programme évitement d'obstacle .....	12
9. Conclusion.....	14

## INDEX DES FIGURES

CHAPITRE 01 : introduction générale à la robotique mobile

Figure 1.1 robot de type voiture.....03

CHAPITRE 02 : Réalisation pratique

Figure 2.1 Schéma électronique du robot .....04

Figure 2.2 Schéma synoptique d'un capteur à ultrason.....06

Figure 2.3 Schéma électronique de la carte arduino.....07

Figure 2.4 moteur DC.....08

## 1. Introduction

La robotique est un ensemble de disciplines (mécanique, électronique, automatique, informatique), elle se subdivise en deux types : les robots industriels et les robots mobiles. Les robots industriels sont généralement fixes, ils sont utilisés dans des nombreuses applications industrielles: l'assemblage mécanique, la soudure, la peinture... Les robots mobiles ne sont pas fixes, ils sont classifiés selon la locomotion en robots marcheurs, à roues, à chenilles... comme ils peuvent être classifié selon le domaine d'application en robots militaires, de laboratoire, industriels et de services. L'aspect particulier de la mobilité impose une complexité technologique (capteurs, motricité, énergie) et méthodologique tel que le traitement des informations par utilisation des techniques de l'intelligence artificielle ou de processeurs particuliers (vectoriels, cellulaires).L'autonomie du robot mobile est une faculté qui lui permet de s'adapter ou de prendre une décision dans le but de réaliser une tâche même dans un environnement peu connu ou totalement inconnu. En vue de bien vouloir comprendre le fonctionnement d'un robot mobile nous allons passer à la réalisation d'un robot détecteur de présence qui circule dans un espace et évite les obstacles lorsqu'il en rencontre

## 2. Généralité sur la robotique mobile :

### 2.1définition :

Le terme (Robot) prend son origine du mot slave (Paboma) (se prononce robota) qui veut dire en russe travail ou en tchèque corvée ou travail forcé. Il désigne aussi une machine à l'aspect humain, capable de se mouvoir et d'agir, qu'un mécanisme automatique pouvant effectuer certaines opérations, et capable par fois de modifier de lui-même son cycle de fonctionnement et d'exercer un certain choix.

### 2.2 Robot mobile :

Contrairement au robot industriel qui est généralement fixé, le robot mobile est doté de moyens qui lui permettent de se déplacer dans son espace de travail. Suivant son degré d'autonomie ou degré d'intelligence, il peut être dote de moyens de perception et de raisonnement. Certains sont capables, sous contrôle humain réduit, de modéliser leur espace de travail et de planifier un chemin dans un environnement qu'ils ne connaissent pas forcément d'avance.

Actuellement, les robots mobiles les plus sophistiqués sont essentiellement orientés vers des applications dans des environnements variables ou incertains, souvent peuplés d'obstacles, nécessitant une adaptabilité à la tâche.

#### 2.2.1 Architecture d'un robot mobile

En général un robot mobile est constitué de trois structures :

**A. Structure mécanique:** elle assure le mouvement du robot par des roues motrices placées selon le type de mouvement et la précision de la tâche voulue.

**B. Structure instrumentale :** un robot est équipé d'un certain nombre de capteurs de sécurité afin de leur donner une certaine connaissance de l'environnement.

C. Structure informatique : une commande numérique est impérative, afin de bien analyser les différentes informations, soit du système de perception ou de localisation. Cette commande peut être à base d'un microprocesseur ou microcontrôleur.

Exemple de robot mobile :



Figure 1.1 robot mobile de type voiture

## CONCLUSION

Nous avons présenté, dans ce chapitre les robots mobiles qui sont des véhicules, les robots sont utilisés à jouer un rôle de plus en plus important dans notre vie mais ceci n'annule pas l'existence de certains problèmes pour assurer une bonne application de ces robots. Comme exemples de ces problèmes nous citons l'analyse de l'environnement, planification, navigation.

### CHAPITRE 02 : Réalisation pratique d'un robot mobile éviteur d'obstacles :

#### 1. Introduction :

L'idée principale de ce projet est de réaliser un robot mobile type voiture puis de le guider automatiquement en l'obligeant à suivre une trajectoire. Ce qui fait que ce travail a été divisé en trois parties, une partie commande et l'autre mécanique et informatique.

La réalisation de ce véhicule requiert diverses notions dans plusieurs domaines : électronique, informatique, électrotechnique et mécanique, ce qu'on appelle brièvement **Robotique**.

La commande de ce véhicule sera assurée principalement par carte de commande à base de microcontrôleur, et deux moteurs pas à pas alimentés par des piles.

#### 2. Problématique :

Il s'agit ici d'éviter la collision du robot avec les obstacles présents dans l'environnement. On ne se soucie pas de la direction que prend le robot pour éviter l'obstacle ni de la mémorisation des obstacles rencontrés. Pas de construction de carte.

La définition d'un problème en logique floue se décompose en trois parties principales? Définition des variables : quelles sont les entrées, les sorties disponibles et pertinentes? Partitionnement des domaines de définitions de ces variables, on crée alors des sous-ensembles flous (obstacle près, très près ou éloigné).

#### 3. Cahier des charges

L'objectif est de construire un robot à bas coût et évolutif.

- reproductibilité : toutes les pièces utilisées doivent être faciles à trouver
- low-cost : le coût doit être acceptable pour une utilisation en nombre, de l'ordre de 800dh

- polyvalent: la carte Arduino peut être utilisée pour réaliser d'autres projets
  - programmation libre et open-source : code open-source et GPL
  - simplicité de réalisation
  - Il doit être facilement réalisable sans avoir de compétences particulières en mécanique ou bricolage
- [Schéma](#)

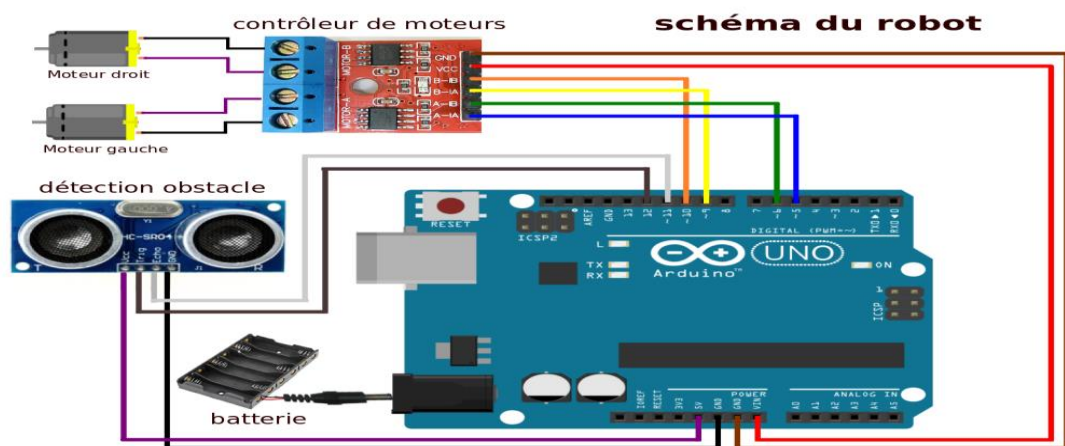


Figure 2.1 [Schéma du robot](#) :

#### 4. [Partie commande](#) :

##### 4.1 [les capteurs](#) :

Un capteur est un dispositif transformant l'état d'une grandeur physique observée en une grandeur utilisable, nous allons utiliser les capteurs ultrason pour la réalisation de notre robot :

##### 4.1.1 [Le capteur ultrasonique](#) :

Les capteurs ultrasons fonctionnent en mesurant le temps de retour d'une onde sonore inaudible par l'homme émise par le capteur. La vitesse du son étant à peu près stable, on en déduit la distance à l'obstacle. Comme il est décrit précédemment, on utilise ces capteurs pour localiser le robot par rapport à des obstacles statiques ou dynamiques, ce capteur repose sur l'idée de calculer le temps de vol d'un signal ultrasonique depuis son point de départ. Alors si la distance entre le transducteur et l'obstacle est « D » donc la distance parcourue par l'onde ultrasonique est « 2D », le temps du parcours est donné par :

Ou :  $T$  : le temps entre l'émission et la réception.

$D$  : distance entre le transducteur et l'obstacle.

$V_{son}$  : la vitesse de déplacement des ultrasons dans l'air.

Généralement, l'air est la porteuse de l'onde ultrasonique, cette dernière se propage, à la même vitesse que le son.

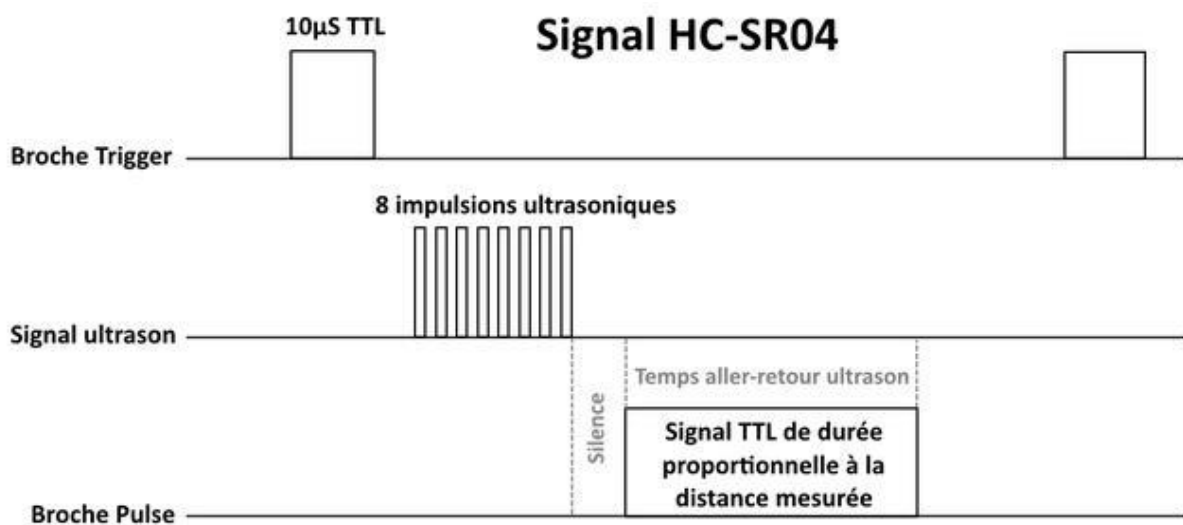
Le capteur utilisé dans notre projet est Le [capteur HC-SR04](#) qui est un capteur à ultrason low cost. Ce capteur fonctionne avec une tension d'alimentation de 5 volts, dispose d'un angle de

mesure de 15° environ et permet de faire des mesures de distance entre 2 centimètres et 4 mètres avec une précision de 3mm.

Principe de fonctionnement :

1. On envoie une impulsion **HIGH** de 10µs sur la broche **TRIGGER** du capteur.
2. Le capteur envoie alors une série de 8 impulsions ultrasoniques à 40KHz (inaudible pour l'être humain, c'est quand plus agréable qu'un biiiiiiiip).
3. Les ultrasons se propagent dans l'air jusqu'à toucher un obstacle et retourne dans l'autre sens vers le capteur.
4. Le capteur détecte l'écho et clôture la prise de mesure.

Le signal sur la broche **ECHO** du capteur reste à **HIGH** durant les étapes 3 et 4, ce qui permet de mesurer la durée de l'aller-retour des ultrasons et donc de déterminer la distance.



[Schéma synoptique du capteur à ultrason :](#)

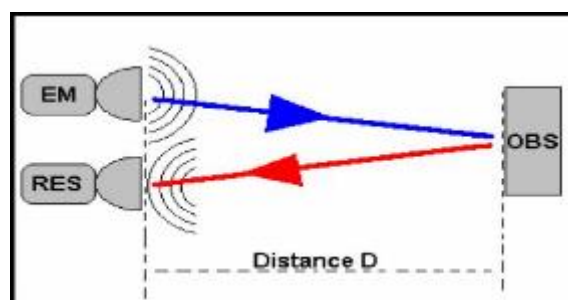


Figure 2.2



#### 4.2 les microcontrôleurs :

Le microcontrôleur est un objet technique, intégrant de l'électronique, fait souvent apparaître des fonctions ayant pour rôle le traitement d'information : opérations arithmétiques (Addition, multiplication...) ou logiques (ET, OU...) entre plusieurs signaux d'entrée permettant de générer des signaux de sortie.

Ces fonctions peuvent être réalisées par des circuits analogiques ou logiques. Mais, lorsque l'objet technique devient complexe, et qu'il est alors nécessaire de réaliser un ensemble important de traitements d'informations, il devient plus simple de faire appel à une structure à base de microcontrôleur.

##### 4.2.1 La carte arduino uno :

C'est un circuit imprimé comportant tous les composants électroniques nécessaires pour faire fonctionner un microcontrôleur (Atmega 328) associé à une interface USB lui permettant de communiquer avec un ordinateur. La carte "ARDUNO uno" que nous allons utiliser dispose de 14 broches ("pin" en anglais, numérotées de 0 à 13) qui peuvent être configurées en "entrées digitales" ou en "sorties digitales" susceptibles de délivrer une intensité maximale de 40mA sous une tension égale à 0V ou 5V. Certaines de ces broches peuvent être configurées en "sorties PWM" (Pulse Width Modulation ou modulation de largeur d'impulsion). Elle possède également 6 entrées analogiques (notées A0 à A5) permettant de mesurer des tensions comprises entre 0V et 5V grâce à un convertisseur A/N à 10 bits. Les broches pin0 et pin1 ne sont pas disponibles pour l'utilisateur de la carte car elles sont utilisées pour communiquer avec l'ordinateur via le port USB, et sont notées RX (réception) et TX (transmission) sur la carte. Le port USB de la carte ARDUINO est géré par l'ordinateur comme un port série virtuel (COMx) et il est nécessaire d'installer les drivers correspondants avant d'établir la liaison ordinateur-Arduino.

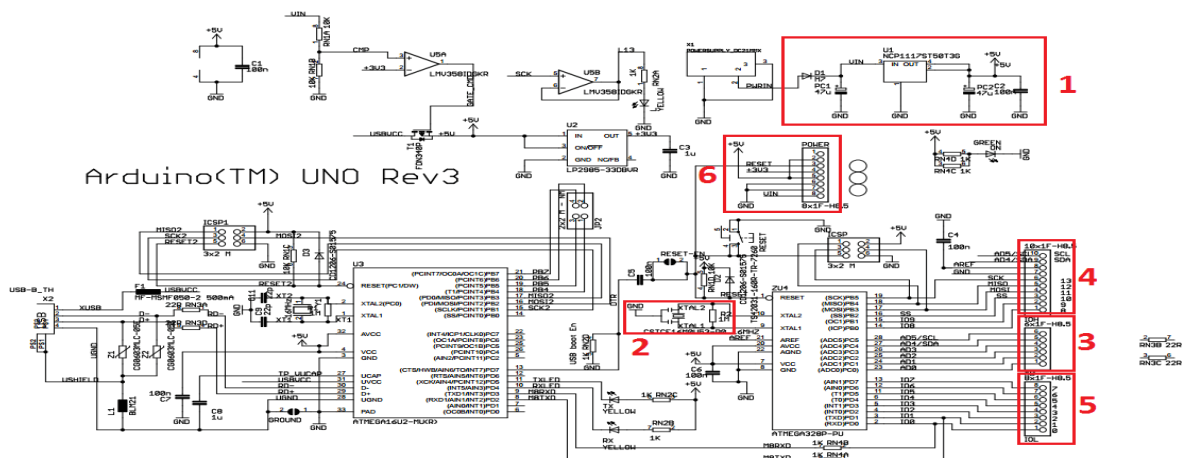


Figure 2.3 Schéma électronique de la carte arduino





Carte arduino uno

#### 4.3 Alimentation :

Nous avons besoin de deux alimentations continues, une de 5V pour alimenter le capteur, et une autre de 9V pour la carte arduino et les deux moteurs. Pour faire marcher les deux roues Nous avons utilisé 4 batteries de 1.5V en série. L'ensemble de tension égale à 6V.

#### 5. Partie mécanique :

La partie mécanique est composée de deux types de moteur :

##### 5.1 types de moteurs utilisés :

###### 5.1.1 les moteurs DC:

Un **moteur DC** est un convertisseur électromécanique permettant la conversion bidirectionnelle d'énergie entre une installation électrique parcourue par un courant continu et un dispositif mécanique. **Les moteurs DC** ont ainsi la particularité de pouvoir fonctionner dans les 2 sens, suivant la manière dont le courant lui est soumis. Les moteurs DC sont pilotés par un contrôleur moteur. Les contrôleurs de moteurs électriques permettent de piloter un moteur en vitesse, en position ou en couple en fournissant la puissance électrique et le signal adéquat.



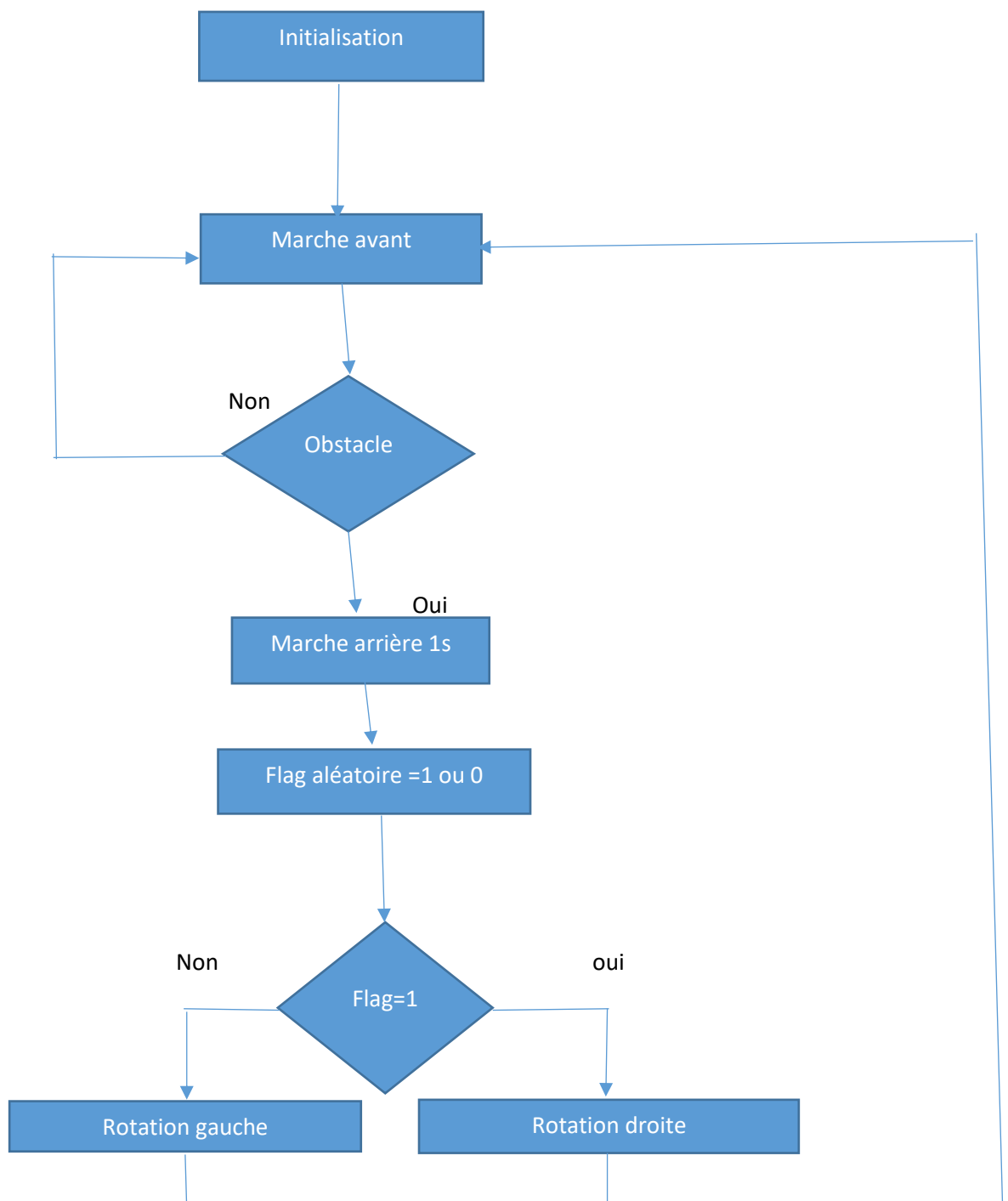
Figure 2.4 moteur DC

6. partie logiciel et programmation :

6.1 logiciel de programmation arduino :

Ce logiciel permettra d'écrire le code source, de le compiler et de l'envoyer vers la carte qui l'exécutera ensuite à l'aide de son microcontrôleur. Grâce aux microcontrôleurs et à la carte Arduino les développeurs pourront alors programmer toutes sortes d'interaction.

7. Organigramme principal du mouvement de notre robot :



### 7.1 Sous-programme détection d'obstacle :

```
Const int trigPin=12;  
Const int echoPin=11;  
Const int buzzer = 13;  
  
Long duration;  
Int distance;  
Int safetyDistance;
```

On commence le code avec quatre constantes : deux constantes pour les broches **TRIGGER** et **ECHO** du capteur, une constante pour le buzzer, une constante qui servira de durée pour la prise de mesure et une constante pour définir la distance qui nous sépare de l'obstacle du son.

```
Void setup () {
```

```
    PinMode (trigPin, OUTPUT);  
    DigitalWrite (trigPin, LOW) ;  
    PinMode (echoPin, INPUT);  
    PinMode (buzzer, OUTPUT);
```

```
    Serial.begin(9600);
```

```
}
```

La fonction **setup ()** initialise le port série, met la broche **TRIGGER** du capteur en sortie et à **LOW**, et met la broche **ECHO** du capteur en entrée.

```
Void loop () {
```

```
    DigitalWrite (trigPin, HIGH);  
    DelayMicroseconds (10);  
    DigitalWrite (trigPin, LOW);  
    Duration = pulseIn (echoPin, HIGH);  
    Distance = duration *0.034/2;
```

```
    SafetyDistance = distance;  
    If (safetyDistance<=10)  
    {DigitalWrite (buzzer, HIGH);  
    }  
    Else {  
        DigitalWrite (buzzer, LOW);  
    }  
}
```

```
}
```

La fonction `loop ()` s'occupe de la mesure et de l'affichage.

Elle génère d'abord l'impulsion `HIGH` de 10µs qui déclenche la prise de mesure. Elle mesure ensuite le temps nécessaire pour un aller-retour du signal ultrason avec la fonction `pulseIn ()`. Pour finir, elle calcule la distance avant de l'afficher sur le port série.

### Tableaux des résultats

<u>Distance mm</u>	<u>mesure</u>	<u>Erreur</u>
<u>10</u>	<u>10.20</u>	<u>0.20</u>
<u>20</u>	<u>18.70</u>	<u>1.30</u>
<u>30</u>	<u>23.46</u>	<u>6.54</u>
<u>40</u>	<u>32.64</u>	<u>7.36</u>
<u>50</u>	<u>42.50</u>	<u>7.50</u>
<u>60</u>	<u>52.02</u>	<u>7.98</u>
<u>70</u>	<u>66.30</u>	<u>3.70</u>

#### 7.2 Sous-programme évitement d'obstacle :

```
#define motorPin1a 3 // Marche avant du premier moteur
#define motorPin1b 4 // Marche arrière du premier moteur
#define speedPin1 9 // L293D enable pin pour le premier moteur
#define motorPin2a 5 // Marche avant du deuxième moteur
#define motorPin2b 6 // Marche arrière du deuxième moteur
#define speedPin2 10 // L293D enable pin pour le deuxième moteur

// capteur à ultrasons:
int trigPin = 12;
int echoPin = 11;

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

long distance;
boolean sensRotation;

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

    long temps;

    // Nous envoyons un signal haut d'une durée de 10 microsecondes,
    en sandwich
    // entre deux signaux bas. Des 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 () {

    // réglage des broches à output
    pinMode (motorPin1a, 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);

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

void loop () {

    Mspeed = 700; // vitesse du moteur 0 à 1023

    Look (); // y a-t-il un obstacle devant?

    if (distance > seuil) {

        // marche avant:

        analogWrite (speedPin1, Mspeed);
        digitalWrite (motorPin1a, HIGH);
        digitalWrite (motorPin1b, LOW);

        AnalogWrite (speedPin2, Mspeed);
        DigitalWrite (motorPin2a, HIGH);
        DigitalWrite (motorPin2b, LOW);

        Delay(100);
    }

    else { // on a détecté un obstacle
        // Nouvelle vérification pour éviter les faux positifs
        Look ();
        If (distance <= seuil) {
            // Obstacle confirmé, on tourne sur place

            AnalogWrite (speedPin1, Mspeed);
            digitalWrite (motorPin1a, sensRotation);
            digitalWrite (motorPin1b, !(sensRotation));
        }
    }
}

```

```

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

        delay (100);
    }
}

```

## Conclusion

L'objectif de notre travail est la réalisation d'un robot mobile type voiture avec évitement d'obstacles en utilisant des capteurs de distances (ultrason) ainsi que l'application d'une intelligence artificielle, la logique floue dans notre cas. Notre robot se déplace sous une trajectoire programmée d'avance.

Pour réaliser ce travail, on est passé par différentes étapes :

On a utilisé un capteur ultrasonique HC-SR04 pour la détection des obstacles et le calcul de la distance entre le robot et l'obstacle. Le robot réalisé se déplace par 2 roues ; les deux roues s'occupent du mouvement par deux moteurs DC. Nous avons utilisé ce type de moteur pour la précision dans le déplacement et la rotation vers la cible.

Des circuits de commande sont réalisés afin de contrôler les moteurs associés. L'ensemble de système de perception et de déplacement est commandé par un microcontrôleur qui doit en utilisant les informations actuelles, décider l'action à prendre. Pour notre cas ; on a utilisé le microcontrôleur célèbre, la carte arduino uno, dont ses caractéristiques particulières nous ont aidé à faciliter les tâches surtout en ce qui concerne sa programmation.

Ce travail nous a permis de traiter des problèmes d'ordre pratique et de vérifier des connaissances théoriques acquises toute le long de notre formation.

Grâce au travail continu, on a pu atteindre notre but et satisfaire le cahier de charge, mais cela ne veut pas dire qu'il est complet, nous proposons que le robot réalisé soit la base de toute une série d'améliorations que nous n'avons pas eu la chance de les faire par manque de temps et de matériel. Le nombre d'améliorations que peuvent être ajoutées sont :

Utilisation de capteurs plus performants comme les capteurs laser ou les caméras et d'utiliser plus de capteurs ou un plateau de capteurs rotatif pour couvrir l'environnement pour choisir meilleur trajectoire? L'utilisation d'un microcontrôleur de nouvelle génération comme le uC Atmel (atmega)? L'utilisation des moteurs pas à pas avec plus de pas pour avoir une grande précision dans le mouvement du robot.