

وزارة التعليم العالي والبحث العلمي

BADJIMOKHTAR-ANNABAUNIVERSITY

UNIVERSITE BADJI MOKHTAR ANNABA



جامعة باجي مختار - عنابة

Année : 2018/2019

Faculté: Sciences de l'Ingéniorat

Département: Electronique

MEMOIRE

Présenté en vue de l'obtention du diplôme de : LICENCE

Intitulé

Robot détecteur et éviteur d'obstacles

Domaine : Sciences et Technologie

Filière : Télécommunication

Par : Semichi Achwak
Taba Lina

DEVANT Le JURY

Directeur de mémoire : Madame Harize

Examineur : Monsieur A.Redjati

Remerciements :

Au terme de ce travail, nous tenons tout d'abord à remercier Madame Harize qui en tant que directeur de mémoire, s'est toujours montré à l'écoute et très disponible tout au long de la réalisation de notre projet, ainsi pour l'inspiration, l'aide et le temps qu'elle a bien voulu nous consacrer. Merci encore de la part de tous vos élèves.

Nos plus vifs remerciements s'adressent aussi au chef département d'électronique Monsieur A.Redjati pour sa grande disponibilité, ses précieux conseils et ses encouragements.

Nous tenions aussi à remercier tous les enseignants qui ont contribué à notre formation universitaire durant nos trois années d'études.

Enfin nous adressons nos plus sincères remerciements à toutes les personnes qui ont participé de près ou de loin à l'élaboration de ce mémoire ainsi qu'à la réussite de cette formidable année

universitaire.

Résumé :

Robot détecteur d'obstacles

Notre projet de fin de cycle consiste en la réalisation d'un robot détecteur d'obstacles à base d'Arduino et un capteur à ultrasons. Un tel robot est complètement autonome et est capable d'éviter tout obstacle rencontré lors de son déplacement. Lorsque le robot rencontre un obstacle, il s'arrête d'avancer et fait une petite marche arrière. Puis, regarde dans à droite et à gauche et prend la meilleure direction possible. Ce type de robot est très utile et est la base de grands projets tels que les voitures automatiques, les robots utilisés dans les usines et même en assistance lors de catastrophes telles que séismes, incendies, effondrement, ... etc.

Abstract:

Obstacles detector robot

Our end-of-the-cycle project consists of the creation of an obstacle detection robot at the Arduino base and an ultrasonic sensor. Such a robot is completely autonomous and is able to avoid any obstacle encountered during its movement. When the robot encounters an obstacle, it stops moving forward and makes a small reverse. Then, look in to the right and left and take the best direction possible. This type of robot is very useful and is the basis of large projects such as automatic cars, robots used in factories and even assistance during disasters such as earthquakes, fires, collapse ... etc.

الخلاصة:

روبوت كاشف للعقبات

مشروعنا لنهاية الدورة يتمثل في إنشاء روبوت كاشف للعقبات وهو مؤسس من لوح آردوينو و جهاز إستشعار للموجات فوق الصوتية. هذا الروبوت يعتبر مستقل تماما ويمكنه تجنب أي عقبة تواجهه أثناء حركته دون أي تدخل خارجي، عندما يواجه الروبوت عقبة، يتوقف عن التقدم للأمام ثم يعكس إتجاهه إلى الوراء قليلا، ثم يستدير إلى اليمين ثم اليسار ثم يقوم بإتخاذ أفضل المسار الممكن. هذا النوع من الروبوتات مفيد للغاية و هو أساس المشروعات الكبيرة مثل السيارات الآلية و الروبوتات المستخدمة في المصانع وحتى المساعدة أثناء الكوارث الطبيعية كالزلازل و الحرائق و الإنهيارات

Table des figures :

Figure -1- ARDUINO Méga 2560	3
Figure -2- Les composants d'un ARDUINO Méga 2560	3
Figure -3- CAPTEUR ULTRASON « HC-SR04 »	4
Figure -4- Fonctionnement du capteur	4
Figure -5- Les parties du capteur	4
Figure -6- Diagramme polaire	5
Figure -7- Montage avec l'Arduino	5
Figure -8- Adafruit Motor Shield l293d	6
Figure -9- Les blocks d'un moteur Shield.....	6
Figure -10- Servo Motor Tower Pro 9g	6
Figure -11- Le branchement de servomoteur	6
Figure -12- Diagramme polaire du fonctionnement d'un servomoteur.....	7
Figure -13- Moteur DC.....	7
Figure -14- montage sur Shield	8
Figure -15- Photo du robot de détection et évitement d'obstacles	8
Figure -16- Organigramme de fonctionnement.....	9
Figure -17- Sens des roues dans les différentes marches.....	9
Figure -18- Fenêtre de logiciel de programmation d'Arduino.....	10

Table de matière :

- Remerciements
- Résumés
- Liste des figures

INTRODUCTION GENERALE	1
Chapitre 1.Présentation du projet	
1.1 Problématique.....	2
1.2 Objectif du projet et ses caractéristiques.....	2
1.3 Cahier des charges.....	2
Chapitre 2. Choix des composants	
2.1 ArduinoMega2560.....	3
2.2 Capteur Ultrason « HC-SR04 ».....	4
2.3 Motor Shield « ADAFRUIT L293D ».....	5
2.4 Servo moteur « Tower Pro 9g ».....	6
2.5 Moteurs DC.....	7
Chapitre 3. Réalisation du projet	
3.1 Les étapes de montage.....	8
3.2 La programmation.....	10
3.3 Les difficultés rencontrées.....	13
CONCLUSION GENERALE	14
Bibliographie	15

INTRODUCTION GENERALE:

La Robotique fait partie des sciences des objets et des systèmes artificiels. Elle peut être vue comme la science de la perception, du mouvement et de leur intégration en une machine physique, mécanique et informatique.

Un robot est donc un système matériel possédant des capacités de perception, d'action, de décision et de communication, parfois capable d'améliorer ses propres performances par apprentissage automatique ou supervisé par des humains.

Si la Robotique non manufacturière a été longtemps considérée comme un champ scientifique très prospectif les applications, en dehors de la conquête spatiale, semblaient futuristes, les progrès scientifiques et technologique importants intervenus dans les grandes disciplines invoquées par la Robotique (mécanique, électronique, informatique, automatique) ont considérablement ouvert le spectre de ses applications, de plus en plus tangibles et crédibles.

Le mémoire est organisé en trois chapitres.

Dans le premier chapitre, l'objectif principal, les différentes caractéristiques de notre projet et l'acquisition des composants nécessaires ont été présentés.

Le chapitre 2 décrit en détail les différents composants utilisés.

Dans le troisième chapitre, les différentes étapes de réalisation du projet et la partie de programmation sont expliquées. Les tests finaux sont aussi présentés.

Chapitre 1. Présentation du projet

1.1 Problématique :

Notre étude a pour objectif la détection et l'évitement d'obstacles par un robot autonome qui élabore des lois de contrôles et prend des décisions à partir de la connaissance de son environnement. Pour assurer la sécurité du robot, il est nécessaire qu'il navigue sans collision avec les objets qui l'entourent. Comment alors peut-on réaliser ce robot? Et Comment peut-on le faire avec les moyens disponibles au niveau du marché algérien d'une part, et d'autre part avec un budget personnel assez limité ?

On va voir tout ceci étape par étape d'après nos connaissances personnelles et les recherches documentaires dans les différentes sources.

1.2 Objectifs du projet et ses caractéristiques :

L'objectif au terme de ce projet est de réaliser un robot éducatif, facile à monter et bon marché. Il devra rassembler des éléments permettant de réaliser des expériences amusantes de découvertes de la robotique, l'électronique et l'informatique.

Nous voulons donc concevoir un robot capable d'interagir avec son environnement au moyen d'un capteur ultrason. Ce robot va être capable d'éviter les obstacles quand il les retrouve dans son chemin. Il sera commandé à l'aide de différents programmes C.

On peut citer quelques points caractéristiques de ce projet :

- Toutes les pièces utilisées doivent être faciles à trouver. Le coût doit être acceptable pour une utilisation en nombre.
- La carte Arduino peut être utilisée pour réaliser d'autres projets.
- Code open-source et 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

1.3 Cahier des charges :

Premièrement, nous avons décidé d'approfondir nos connaissances par des recherches et des études dans ce cas.

Une première recherche web nous a conduits sur le bon choix du capteur « capteur ultrasons » car généralement un système de navigation autonome utilise ce type de capteur qui peut fournir une information très importante au robot, à savoir la distance le séparant des obstacles autour de lui.

La seconde partie de la recherche a été sur le reste de matériel nécessaire. Nous avons constaté que nous avons besoin d'un : driver, des moteurs avec leurs roues, un servomoteur et une carte Arduino.

Ensuite, nous avons effectué des recherches sur la disponibilité du matériel et nous avons fini par acheter le matériel chez les étudiants de la promotion qui nous a précédées.

Enfin, dans la troisième partie, nous nous sommes concentrés sur la programmation qui permet de contrôler notre robot.

Chapitre 2. Choix des composants

2.1 Arduino Mega2560 :

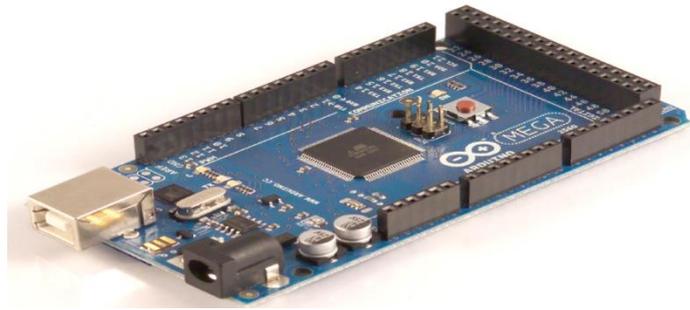


Figure -1- ARDUINO Méga 2560 -

Un Arduino Méga 2560 est une carte à base du microcontrôleur l'ATmega2560 qui a besoin d'être programmé et cela introduit un élément logiciel sous la forme d'un environnement de développement intégré (IDE) fonctionnant sur des ordinateurs personnels. Avec celui-ci, les utilisateurs écrivent des programmes (appelés sketches) à l'aide de C ou C++ qui sont des langages de programmation.

Sur l'arduino se trouve les composants présentés dans la figure ci-dessous :

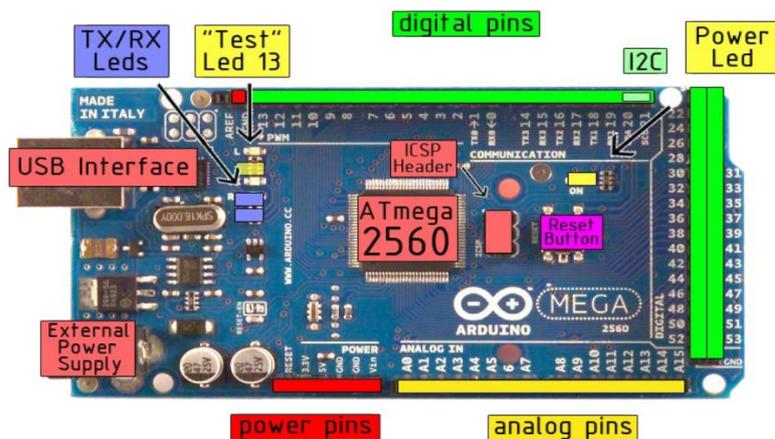


Figure -2- les composants d'un ARDUINO Méga 2560 -

Le microcontrôleur :

Le microcontrôleur peut être programmé pour analyser et produire des signaux électriques de manière à effectuer des tâches très diverses comme le contrôle des appareils et le pilotage d'un robot.

Les caractéristiques techniques d'une carte Arduino Méga2560 sont les suivants :

- ❖ Microcontrôleur : ATmega2560
- ❖ Tension de fonctionnement : 5V
- ❖ Tension d'entrée (recommandée) : 7 à 12V
- ❖ Tension d'entrée (limites) : 6 à 20V
- ❖ Digital I/O Pins : 54 (dont 14 fournissent une sortie PWM« pulse width modulation »)
- ❖ Broches d'entrée analogiques : 16

- ❖ Courant CC par broches I/O : 40 mA
- ❖ Courant DC pour 3.3V Pin : 50 mA
- ❖ Mémoire Flash : 256 Ko dont 8 Ko utilisés par le chargeur de démarrage
- ❖ SRAM : 8 KB
- ❖ EEPROM : 4KB
- ❖ Vitesse d'horloge : 16 MHz

2.2 Capteur Ultrason « HC-SR04 » :



Les Ultrasons sont des ondes infra-acoustique qui oscillent à des fréquences supérieures au seuil acoustique (40kHz et vitesse de 340 m/s). Ces fréquences n'interfèrent pas avec les bruits audibles et elles sont très précises, rapides et peuvent être personnalisées.

Figure -3- Capteur à ultrason -

Les capteurs Ultrasons sont utilisés pour effectuer des mesures de distance et/ou détecter les objets en état statique ou en mouvement. Ils fonctionnent en mesurant le temps de retour d'une onde sonore inaudible émise par le capteur. La vitesse du son étant à peu près stable, donc il reste à déduire la distance à l'obstacle.

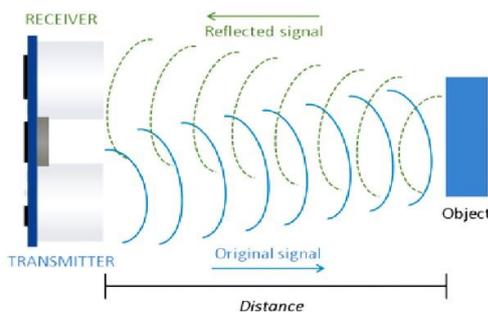


Figure -4- fonctionnement du capteur -

Les caractéristiques techniques du module sont les suivantes :

- Alimentation : 5v.
- Consommation en utilisation : 15 mA.
- Large gamme de mesure de quelques mm à plus de 5 mètres.
- Résolution : 3 mm
- Angle de mesure : < 15°.

Un capteur ultrason a deux parties : émetteur (VSS, Trig) et récepteur (Echo, GND) :

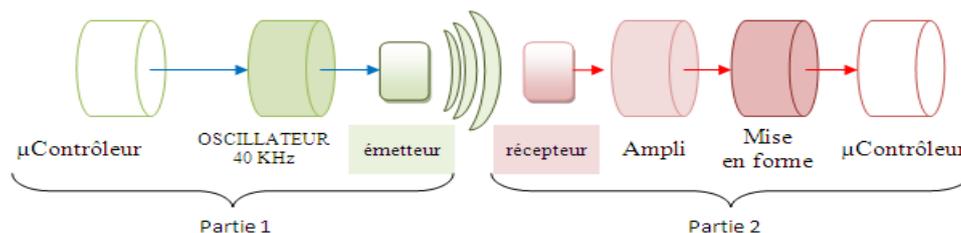


Figure -5- les parties du capteur

Principe de fonctionnement :

Le capteur à ultrasons HC-SR04 est utilisé par le système sonar pour déterminer la distance à un objet. Il offre une excellente détection d'un intervalle (sans contact) d'environ **2 cm à 400 cm**. Initialement, en mettant la broche **Trig** à l'état haut (de 0 à + 5v) pour envoyer un court signal (une onde ultrason) pendant au moins de 10 us, on adonc déclencher le **Ping** pour la mesure. S'il détecte un objet, alors il réfléchit le signal d'écho en retour qui est pris comme entrée dans le capteur par le biais de la broche Echo. Le **Ping** va donc repasser à 0 quand l'écho est revenu au récepteur. La broche **Echo** va fournir une impulsion de + 5v dont la durée est proportionnelle à la distance de l'objet. On utilise la fonction **Pulse In** pour calculer le temps. Chaque fois la fonction attend que la broche **Echo** soit à l'état haut pour commencer le chronométrage.

Afin de pouvoir calculer la distance en cm, on utilise la formule suivante :

$$\text{Distance} = (\text{durée de l'impulsion (en } \mu\text{s)}) / 58$$

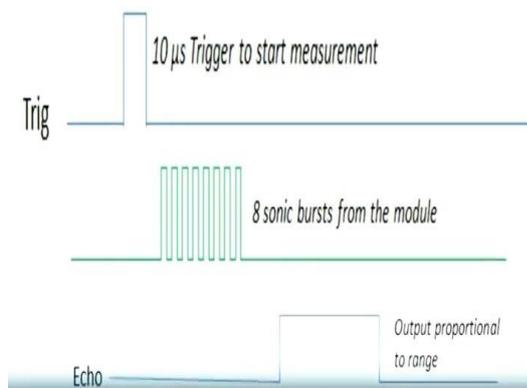
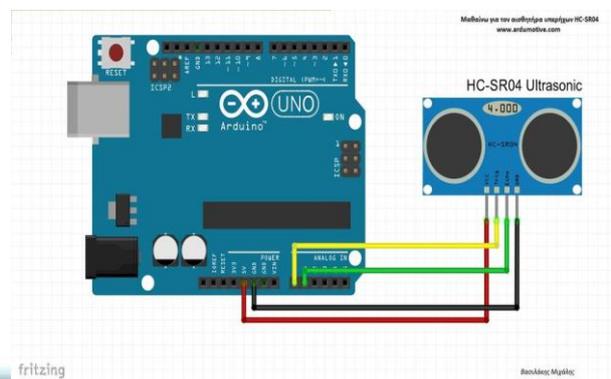


Figure -6- Diagramme polaire



-Figure -7- Montage avec l'Arduino

2.3 Motor Shield « ADAFRUIT L293D » :

Les Shields sont des cartes conçues pour se poser sur l'Arduino. Il permet avec une carte Arduino de contrôler au maximum 2 servomoteurs, 4 moteurs DC ou 2 moteurs bipolaires. Un Shield propose aussi une série bien pratique de trois connecteurs SIL-6 reprenant les ports analogiques A0~A5 de l'Arduino ainsi que VCC et GND, le tout dans l'optique de connecter des capteurs ou d'autres servomoteurs (avec contrôle software).

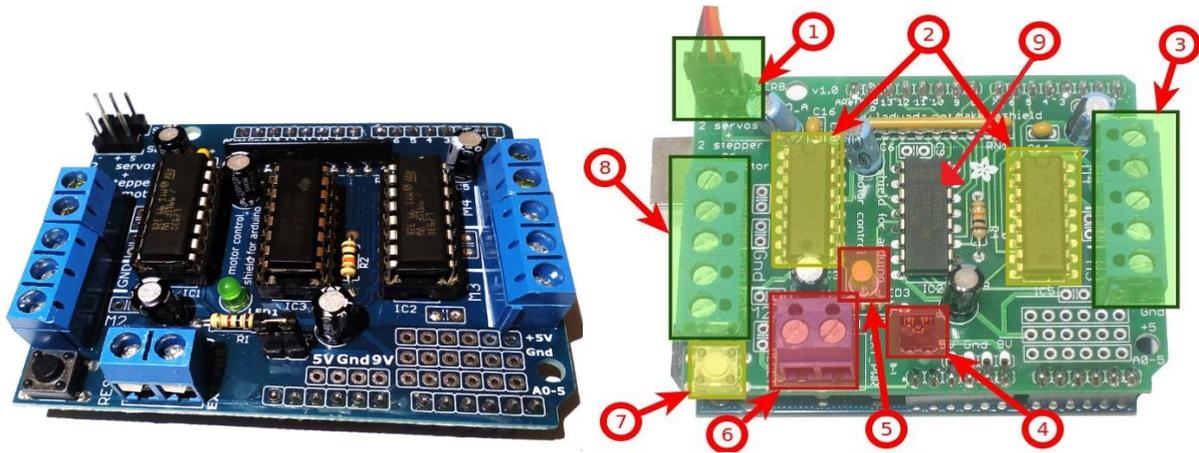


Figure -8- Adafruit motor shield L293d -Figure -9- Les blocks d'un moteur Shield -

1. Connecteurs pour deux Servo moteurs.
2. 2x Double Pont-H L293D : Ils sont des circuits quadruples demi-H à courant élevé. Ils sont conçus pour fournir des courants de commande bidirectionnels allant jusqu'à 600 mA à des tensions comprises entre 4,5V et 36V.
3. Blocks connecteurs pour 2 moteurs à courant continu ou un moteur stepper.
4. Clavier de sélection entre alimentation moteur externe ou prise alimentation Arduino.
5. Indicateur d'alimentation moteur (LED).
6. Block d'alimentation moteur externe
7. Bouton Reset.
8. Blocks connecteurs pour 2 moteurs à courant continu ou un moteur stepper.
9. Le 74HC595 comprend un registre à décalage de 8 bits et un registre de stockage avec des sorties parallèles à trois états. Il convertit l'entrée série en sortie parallèle afin que nous puissions enregistrer les ports IO d'un MCU.

2.4 Servo motor « Tower Pro 9g »:



Figure -10- Servo motor Tower Pro 9g -

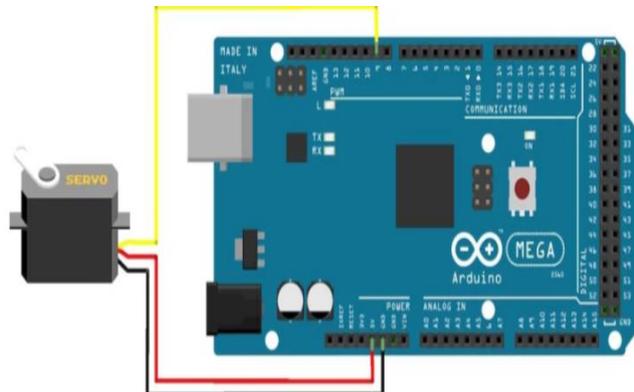
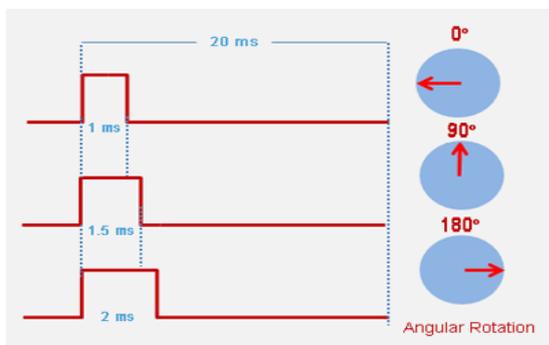


Figure -11- Le branchement de servomoteur-

Un servomoteur est un appareil électrique capable de pousser ou de faire pivoter un objet à des angles ou à des distances spécifiques avec une grande précision. Il est composé d'un moteur simple qui fonctionne à travers un mécanisme d'avertissement c-à-dire qu'il peut alimenter en courant continu ou bien alternatif, un potentiomètre, un ensemble à engrenages et un circuit de contrôle. Ce moteur à trois fils, un fil de GND (marron), fil de VCC (rouge) et un fil de contrôle (jaune)

Le servomoteur est contrôlé par des impulsions avec modulation (PWM) fourni par le fil de contrôle, il existe une impulsion minimal, maximal et un taux de répétition. Il peut tourner à 90 degrés dans les deux sens de puis sa position neutre.

Le servomoteur s'attend à voir une impulsion toutes les 20 millisecondes et la durée de l'impulsion déterminera le degré de rotation.



Les caractéristiques techniques :

- Tension d'alimentation : 4,8 à 6 v.
- Vitesse et couple : 0,12s /60° et 1,2 kg /cm sous 4,8 v.
- amplitude : de 0 à 180°.

On va utiliser le servomoteur pour la rotation du capteur ultrasons.

Figure -12- Diagramme polaire –

2.5 Moteurs DC :



Figure -13- Moteur DC -

Le moteur à courant continu est très simple à utiliser. Câblé dans un sens, il tournera dans un sens, câblé dans le sens contraire il tournera dans le sens contraire. Du fait qu'il est alimenté par une tension continue, on peut faire varier sa vitesse par PWM (ou manuellement avec une alimentation variable).

Chapitre 3. Réalisation du projet

3.1 Les étapes du montage :

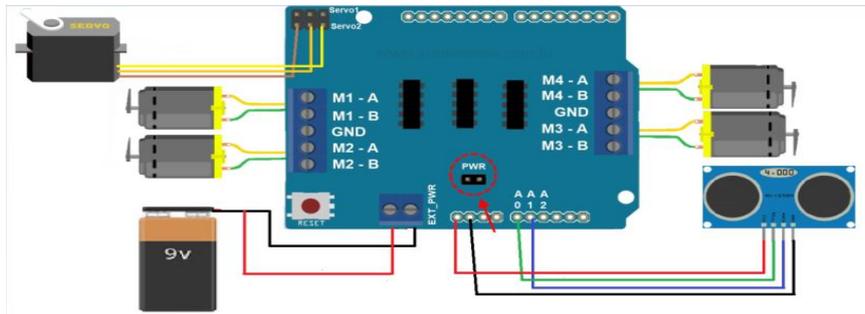


Figure -14- montage sur Shield -

- **1^{er} étape** : Nous avons commencé par l'assemblage sur le châssis des moteurs avec leurs roues.
- **2^{ème} étape** : On a installé la carte Arduino sur le châssis puis on branche le Shield sur l'Arduino.
- **3^{ème} étape** : On relie les câbles des moteurs DC au Shield dans les blocks M1, M2, M3 et M4 (si les moteurs tournent dans le sens opposé, il suffit d'inverser le bronchement des fils)
- **4^{ème} étape** : On a connecté le servomoteur avec le Shield puis on a monté le capteur à ultrason sur le servomoteur avec support pour permettre la rotation du capteur.
- **5^{ème} étape** : On a installé le capteur à ultrason sur le Shield où on va choisir le port A0 pour contrôler la broche TRIG, A1 pour la broche ECHO et bien sûr connecter le VCC et GND avec les ports 5V et GND du Shield.
- **6^{ème} étape** : On branche le câble USB sur l'Arduino pour télécharger le programme.
- **7^{ème} étape** : On a utilisé deux batteries de 9V pour alimenter l'Arduino ainsi que le Shield. Pour séparer l'alimentation de la plaque il faut enlever le fil qui permet d'alimenter le Shield à partir d'Arduino.

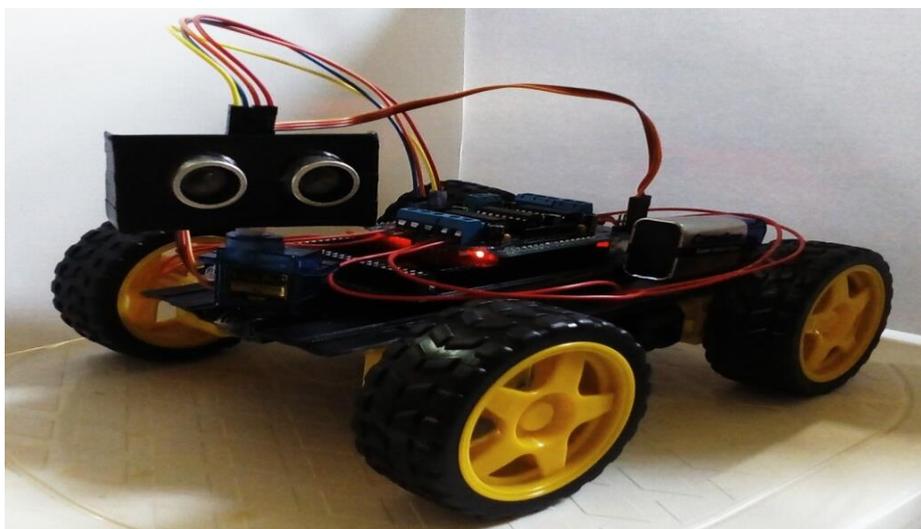


Figure -15- Photo du robot de détection et évitement d'obstacles -

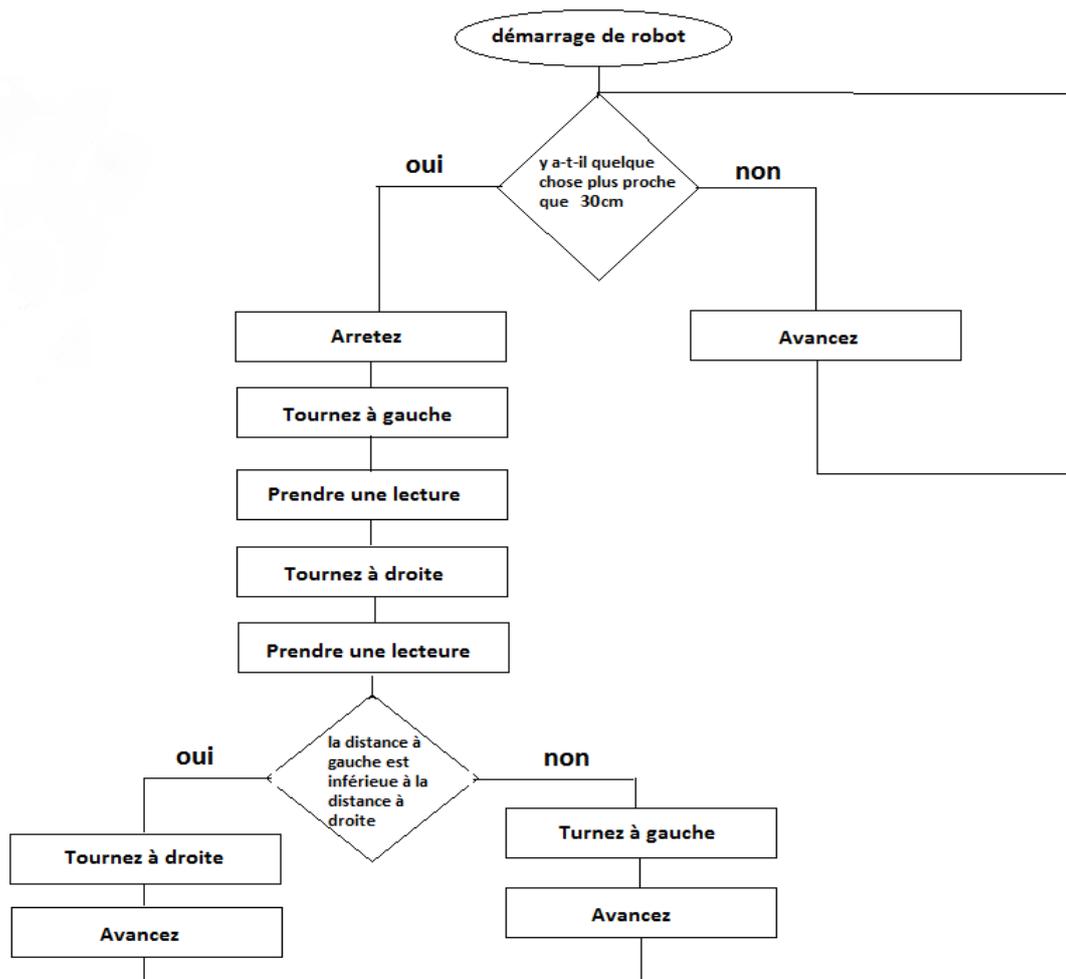


Figure -16- Organigramme de fonctionnement -

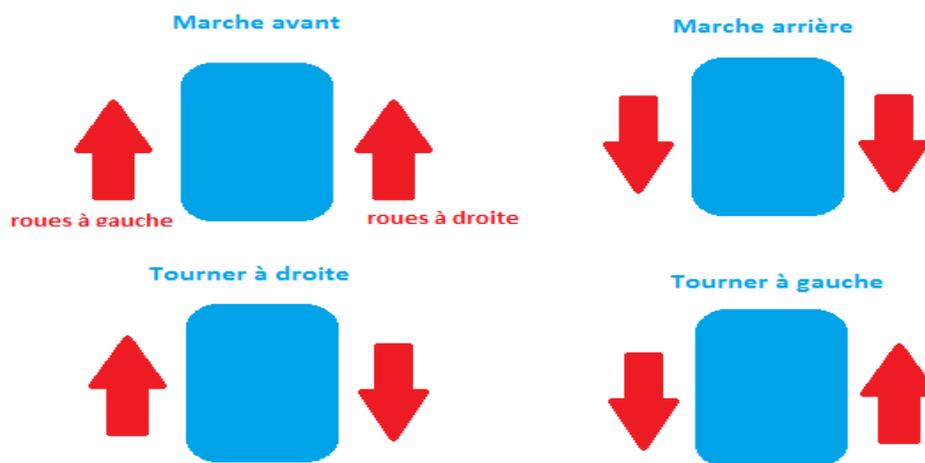


Figure -17- Sens des roues dans les différentes marches -

3.2 La programmation :

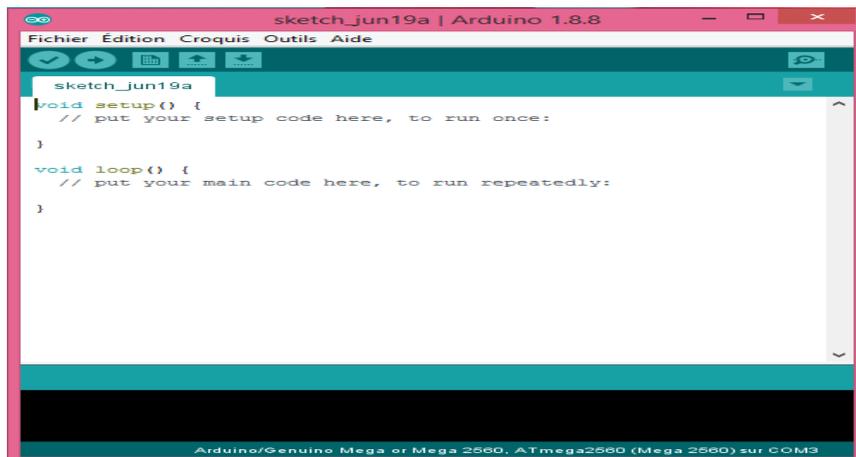


Figure -18- Fenêtre de logiciel de programmation d'Arduino -

- Il faut premièrement installer le logiciel programmation d'Arduino.
- On branche notre Arduino avec l'ordinateur, après on ouvre notre logiciel qui est représenté dans la figure 18, la première chose qu'on a fait et de choisir le port de communication de « **Outils** » de menu « **Tools** », si les ports n'existent pas donc votre ordinateur n'a pas reconnu l'Arduino et vous devez installer le pilote de votre carte, après on choisit le type de carte à utiliser. Pour tester votre Arduino il suffit que vous cliquiez dans le sous-menu **Fichier** » **Exemples** » **basics** après le sketch **Blink** pour allumer une LED.
- Pour faire notre programme il faut premièrement installer les bibliothèques nécessaires, on décompresse les fichiers ZIP sur notre ordinateur de bureau. La bibliothèque de moteurs **Shield Adafruit Motor Shield** et de capteur à ultrason **New Ping**, pour qu'il n'y ait pas d'erreur dans notre programme vous devez renommer la bibliothèque de moteurs Shield à **AFMotor**, on ajoute maintenant ces bibliothèques à travers le menu **Croquis** » **Inclure une bibliothèque** » **Ajouter la bibliothèque ZIP...**
- Maintenant nous sommes prêts à écrire le programme.

PROGRAMME:

```
#include <AFMotor.h> //add Adafruit Motor Shield library
#include <Servo.h> //add Servo Motor library
#include <NewPing.h> //add Ultrasonic sensor library
#define TRIG_PIN A0 // Pin A0 on the Motor Drive Shield soldered to the ultrasonic sensor
#define ECHO_PIN A1 // Pin A1 on the Motor Drive Shield soldered to the ultrasonic sensor
#define MAX_DISTANCE 300 // sets maximum useable sensor measuring distance to 300cm
#define MAX_SPEED 160 // sets speed of DC traction motors to 150/250 or about 70% of
full speed - to get power drain down.
#define MAX_SPEED_OFFSET 40 // this sets offset to allow for differences between the two
DC traction motors
#define COLL_DIST 30 // sets distance at which robot stops and reverses to 30cm
#define TURN_DIST COLL_DIST+20 // sets distance at which robot veers away from object
```

```

NewPing sonar(TRIG_PIN, ECHO_PIN, MAX_DISTANCE); // sets up sensor library to use
the correct pins to measure distance.
AF_DCMotor leftMotor1(1, MOTOR12_1KHZ); // create motor #1 using M1 output on
Motor Drive Shield, set to 1kHz PWM frequency
AF_DCMotor leftMotor2(2, MOTOR12_1KHZ); // create motor #2, using M2 output, set to
1kHz PWM frequency
AF_DCMotor rightMotor1(3, MOTOR34_1KHZ); // create motor #3, using M3 output, set to
1kHz PWM frequency
AF_DCMotor rightMotor2(4, MOTOR34_1KHZ); // create motor #4, using M4 output, set to
1kHz PWM frequency
Servo myservo; // create servo object to control a servo
int leftDistance, rightDistance; //distances on either side
int curDist = 0;
String motorSet = "";
int speedSet = 0;
//----- SETUP LOOP -----
void setup() {
  myservo.attach(10); // attaches the servo on pin 10 (SERVO_1 on the Motor Drive Shield to
the servo object
  myservo.write(90); // tells the servo to position at 90-degrees ie. facing forward.
  delay(1000); // delay for one seconds }
//----- MAIN LOOP -----
void loop() {
  myservo.write(90); // move eyes forward
  delay(90);
  curDist = readPing(); // read distance
  if (curDist < COLL_DIST) {changePath();} // if forward is blocked change direction
  moveForward(); // move forward
  delay(500); }
//-----
void changePath() {
  moveStop(); // stop forward movement
  myservo.write(36); // check distance to the right
  delay(500);
  rightDistance = readPing(); //set right distance
  delay(500);
  myservo.write(144); // check distace to the left
  delay(700);
  leftDistance = readPing(); //set left distance
  delay(500);
  myservo.write(90); //return to center
  delay(100);
  compareDistance();}

void compareDistance() // find the longest distance
{ if (leftDistance>rightDistance) //if left is less obstructed
  { turnLeft();}
  else if (rightDistance>leftDistance) //if right is less obstructed

```

```

    { turnRight();}
else//if they are equally obstructed
    { turnAround();}}
//-----
int readPing() { // read the ultrasonic sensor distance
    delay(70);
    unsigned int uS = sonar.ping();
    int cm = uS/US_ROUNDTRIP_CM;
    return cm;
}
//-----
void moveStop() { leftMotor1.run(RELEASE); leftMotor2.run(RELEASE);
rightMotor1.run(RELEASE); rightMotor2.run(RELEASE);} // stop the motors.
//-----
void moveForward() {
    motorSet = "FORWARD";
    leftMotor1.run(FORWARD); // turn it on going forward
    leftMotor2.run(FORWARD); // turn it on going forward
    rightMotor1.run(FORWARD); // turn it on going forward
    rightMotor2.run(FORWARD); // turn it on going forward
    for (speedSet = 0; speedSet < MAX_SPEED; speedSet +=2) // slowly bring the speed up to
    avoid loading down the batteries too quickly
    { leftMotor1.setSpeed(speedSet);
    leftMotor2.setSpeed(speedSet);
    rightMotor1.setSpeed(speedSet);
    rightMotor2.setSpeed(speedSet);
    delay(5);}
}
//-----
void moveBackward() {
    motorSet = "BACKWARD";
    leftMotor1.run(BACKWARD); // turn it on going backward
    leftMotor2.run(BACKWARD); // turn it on going backward
    rightMotor1.run(BACKWARD); // turn it on going backward
    rightMotor2.run(BACKWARD); // turn it on going backward
    for (speedSet = 0; speedSet < MAX_SPEED; speedSet +=2) // slowly bring the speed up to
    avoid loading down the batteries too quickly
    { leftMotor1.setSpeed(speedSet);
    leftMotor2.setSpeed(speedSet);
    rightMotor1.setSpeed(speedSet);
    rightMotor2.setSpeed(speedSet);
    delay(5);} }
}
//-----
void turnRight() {
    motorSet = "RIGHT";
    leftMotor1.run(FORWARD); // turn motor 1 forward
    leftMotor2.run(FORWARD); // turn motor 2 forward
    rightMotor1.run(BACKWARD); // turn motor 3 backward
    rightMotor2.run(BACKWARD); // turn motor 4 backward
    rightMotor1.setSpeed(speedSet+MAX_SPEED_OFFSET);
}

```

```

rightMotor2.setSpeed(speedSet+MAX_SPEED_OFFSET);
delay(1500); // run motors this way for 1500
motorSet = "FORWARD";
leftMotor1.run(FORWARD); // set both motors back to forward
leftMotor2.run(FORWARD);
rightMotor1.run(FORWARD);
rightMotor2.run(FORWARD); }
//-----
void turnLeft() {
motorSet = "LEFT";
leftMotor1.run(BACKWARD); // turn motor 1 backward
leftMotor2.run(BACKWARD); // turn motor 2 backward
leftMotor1.setSpeed(speedSet+MAX_SPEED_OFFSET);
leftMotor2.setSpeed(speedSet+MAX_SPEED_OFFSET);
rightMotor1.run(FORWARD); // turn motor 3 forward
rightMotor2.run(FORWARD); // turn motor 4 forward
delay(1500); // run motors this way for 1500
motorSet = "FORWARD";
leftMotor1.run(FORWARD); // turn it on going forward
leftMotor2.run(FORWARD); // turn it on going forward
rightMotor1.run(FORWARD); // turn it on going forward
rightMotor2.run(FORWARD); // turn it on going forward}
//-----
Void turnAround() {
motorSet = "RIGHT";
leftMotor1.run(FORWARD); // turn motor 1 forward
leftMotor2.run(FORWARD); // turn motor 2 forward
rightMotor1.run(BACKWARD); // turn motor 3 backward
rightMotor2.run(BACKWARD); // turn motor 4 backward
rightMotor1.setSpeed(speedSet+MAX_SPEED_OFFSET);
rightMotor2.setSpeed(speedSet+MAX_SPEED_OFFSET);
delay(1700); // run motors this way for 1700
motorSet = "FORWARD";
leftMotor1.run(FORWARD); // set both motors back to forward
leftMotor2.run(FORWARD);
rightMotor1.run(FORWARD);
rightMotor2.run(FORWARD);}

```

3.3 Les difficultés rencontrées :

Nous avons rencontré quelques difficultés au cours de la réalisation pratique de notre projet, notamment le poids lourd auquel le robot a été exposé à cause des composants utilisés. Ce qui aggravé les choses sont les moteurs à faible puissance qui ne peuvent pas bien pousser le robot soit en avant ou en arrière, ni même à droite ou à gauche. Malheureusement, on n'a pas pu les changer à cause du manque du matérielles. Le robot ne peut fonctionne pas correctement à 100% et peut coincer lorsqu'il tourne soit à gauche ou à droite.

CONCLUSION GENERALE:

Durant ce projet nous avons pu acquérir une certaine expérience pour la conception et la réalisation des différentes parties d'un robot mobile. Nous avons avant tout, appris à bâtir un projet en respectant les objectifs cités.

Dans notre projet de fin d'étude, nous avons mis l'accent sur la partie perception, vu le rôle qu'elle joue dans l'autonomie et l'intelligence d'un robot mobile autonome.

Nous avons eu, dans notre projet, à faire une étude complète et une réalisation d'un robot mobile capable de détecter les obstacles et de les éviter. Ce robot utilise un capteur ultrason pour détecter les obstacles avant la collision, après cela il se déplace en marche arrière et tourne à droite ou à gauche pour éviter les obstacles.

Notre étude théorique est concrétisée par une conception sur trois plans :

- **Mécanique** : dimensionnement de la base mécanique, d'un système de fixation des moteurs ainsi que les roues.
- **Electrique** : conception d'une carte électronique qui répond aux exigences de l'application
- **Informatique** : élaboration d'un programme qui assure l'asservissement en vitesse des deux moteurs, et l'échange d'information avec le module ultrason.

Ce travail étant une œuvre humaine, n'est pas un modèle unique et parfait, c'est pourquoi nous restons ouverts à toutes les critiques et nous sommes prêtes à recevoir toutes les suggestions et remarques tendant à améliorer d'avantage cette étude. Nous croyons très fort que tout bon travail est toujours l'œuvre d'une équipe.

Bibliographie:

REF [1] : <https://www.arduino.cc/>

REF [2]: <http://www.microsonic.de/fr/support/capteurs-%C3%A0-ultrasons/les-modes-de-fonctionnement.htm>

REF [3]: <http://playground.arduino.cc/Main/AdafruitMotorShield>

REF [4]: <http://www.dzduino.com/store/>

REF [5]: <https://fr.wikipedia.org/wiki/Servomoteur>

Les sites web:

https://fr.wikipedia.org/wiki/Tondeuse_%C3%A0_gazon

<https://www.youtube.com/watch?v=AcAQm1KKbRk>

<https://www.arduino.cc/en/Guide/Libraries>

<http://playground.arduino.cc/Code/NewPing>