



# *ROBOT MADD DRH*





# Réalisation technique

- Fabrication d'un Robot qui évite les obstacles grâce a des capteurs de sons
- Temps Réel





# Liste du matériel

- Arduino
- Wemos D1 Wifi
- Piles
- Voiture Télécommandé inutilisable
- Capteur de son
- Fer a souder

CREATIVITE  
ET  
FABRICATION  
NUMERIQUE



# Programme sous ARDUINO



```

/*****
 *
 * Robot éviteur d'obstacle muni d'un capteur à ultrasons HC-SR04.
 *
 * Comportement: Le robot avance en ligne droite, sauf s'il
 * rencontre un obstacle, auquel cas il tourne sur place jusqu'à
 * ce qu'il n'y ait plus d'obstacle devant lui.
 *
 * http://electroniqueamateur.blogspot.ca/
 *
 *****/
#include <Servo.h> //on importe la bibliothèque Servo
// définition des pins de l'Arduino qui contrôlent le 1er moteur
#define pinIN1 2 // Marche avant du premier moteur
#define pinIN2 3 // Marche arrière du premier moteur
#define pinENA 9 // doit être une pin PWM

// 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
|
int periode=20000;// période entre chaque début d'impulsion en microsecondes

Servo myservo;

/*int pinServo=7; // variable pour stocker le pin pour la commande */

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);

```



```
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(pinIN1, OUTPUT);
pinMode(pinIN2, OUTPUT);
pinMode(pinENA, OUTPUT);

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

/* pinMode(pinServo,OUTPUT); // on prépare le pin en mode OUTPUT
digitalWrite(pinServo,LOW); // on l'initialise à l'état bas */
myservo.attach(7);

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

}

void loop() {

Mspeed = 200; // vitesse du moteur 0 à 1023

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



```
/* for (int angle=0;angle<=60;angle+=20){//on fait varier l'angle de 0 à 60° par tranche de 20°
  setAngle(angle); }// on appelle la fonction setAngle définie plus bas

//fonction setAngle pour envoyer les impulsions
void setAngle(int a){
  int duree=map(a,0,179,1000,2000);// on transforme l'angle en microsecondes et on stocke dans la variable duree
  digitalWrite(pinServo,LOW); }//on met le pin à l'état bas */

if (distance > seuil){

  // marche avant:

  analogWrite(pinENA, Mspeed);
  digitalWrite(pinIN1, HIGH);
  digitalWrite(pinIN2, LOW);
  delay(2000);
  myservo.write(90);
  delay(100);
}

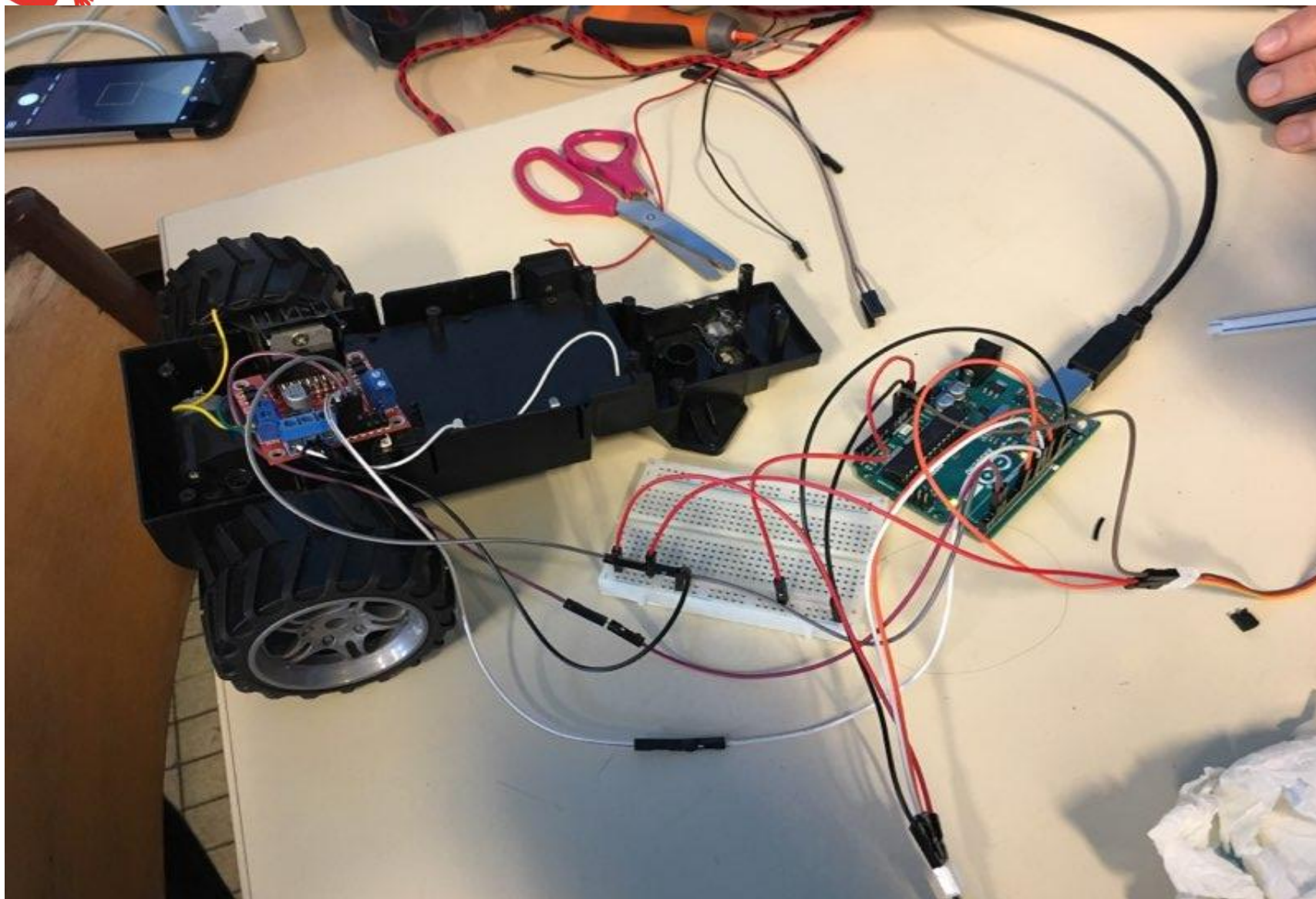
else { // on a détecté un obstacle
  // nouvelle vérification pour éviter les faux positifs
  look();
  if (distance <= seuil){
    // obstacle confirmé, on tourne sur place
    analogWrite(pinENA, Mspeed);
    digitalWrite(pinIN1, sensRotation);
    digitalWrite(pinIN2, !(sensRotation));

    delay(1000);
    myservo.write(140);

  }
}
}
```



# Câblage et Assemblage







# Cadre pédagogique

- S'adresse à un public a partir de 14ans
- A pour objectifs pédagogique : connaître le monde des robots et leurs fonctionnement a travers un simple exemple et apprendre les base du logiciel ARDUINO.