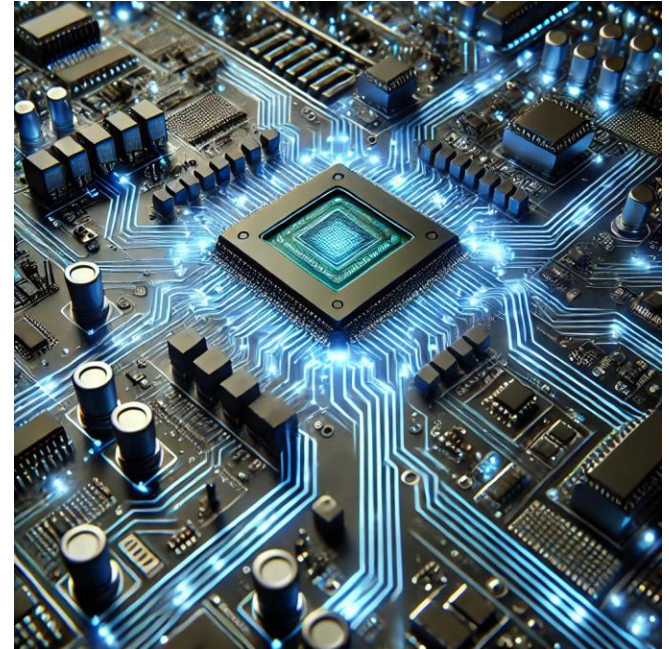


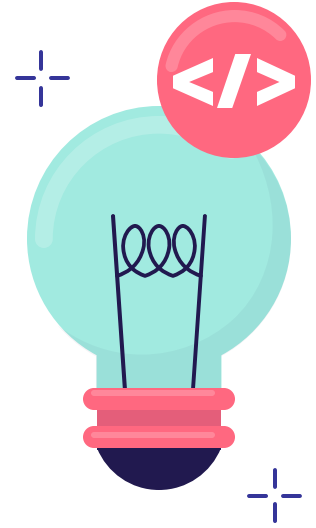
# Etude et réalisation d'un robot détecteur d'obstacle, contrôlé par une carte Arduino

Présenté par :  
Elkhoulati Yahya



# Plan

- 01 Introduction**
- 02 Fonctionnement du robot**
- 03 Matériel nécessaire**
- 04 Montage du robot**
- 05 Programmation et Implémentation**





# 01 Introduction

# Introduction :



Un véhicule **robot** est un véhicule automatisé qui utilise des algorithmes de contrôle et de navigation pour se déplacer sans intervention humaine. Les véhicules **robots** peuvent être utilisés pour des applications telles que la livraison de colis, la reconnaissance et l'inspection, le transport de personnes et la cartographie automatisée. Les véhicules **robots** peuvent utiliser des technologies telles que la vision par ordinateur, les capteurs GPS et LIDAR pour naviguer de manière autonome.

Les **robots** qui détectent les obstacles utilisent des capteurs pour percevoir leur environnement et éviter les obstacles en temps réel. Les capteurs couramment utilisés pour détecter les obstacles comprennent le LIDAR, la caméra, le radar et les capteurs ultrasons. Les informations collectées par ces capteurs sont utilisées pour alimenter un algorithme de contrôle du mouvement du **robot**, qui peut ajuster sa trajectoire en temps réel pour éviter les obstacles. Les **robots** peuvent également utiliser des techniques de cartographie en temps réel pour construire une carte de leur environnement et planifier une trajectoire sécuritaire pour atteindre leur destination.



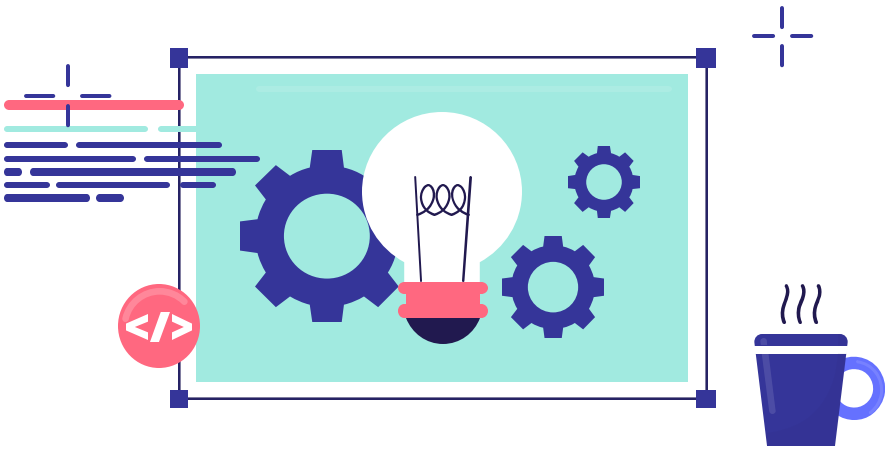
# Applications:

Un robot détecteur d'obstacles est utilisé dans les secteurs industriels, pour des tâches de surveillance ou d'exploration, ainsi que dans la robotique de service (robots aspirateurs, tondeuses automatiques, etc.). Ces robots sont également essentiels pour les véhicules autonomes qui doivent naviguer en toute sécurité dans des environnements complexes.



02

# Fonctionnement du robot



## Fonctionnement du robot commandé par la carte Arduino qui détecte et évite les obstacles:

Un robot détecteur d'obstacles commandé par **Arduino UNO** avec un capteur ultrasonique **HC-SR04** et un **servomoteur** est conçu pour détecter des obstacles et ajuster sa trajectoire pour les éviter. L'ajout d'un servomoteur au capteur ultrasonique permet de déplacer le capteur sur un angle donné (gauche, centre, droite) pour scanner l'environnement. Cela permet au robot d'analyser plusieurs directions et de prendre des décisions plus précises pour contourner les obstacles.

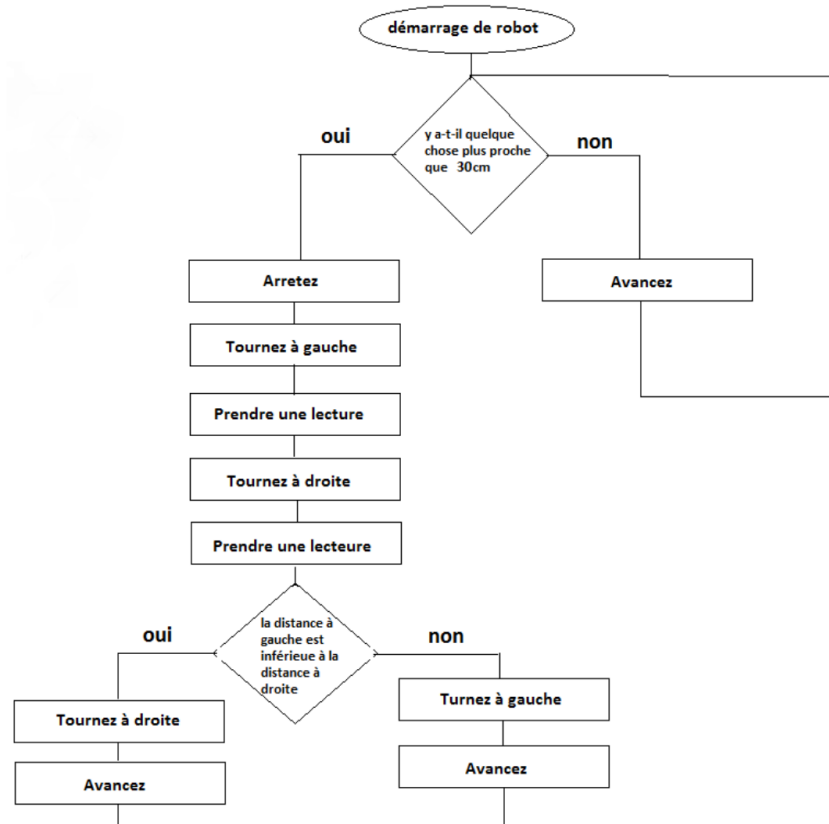


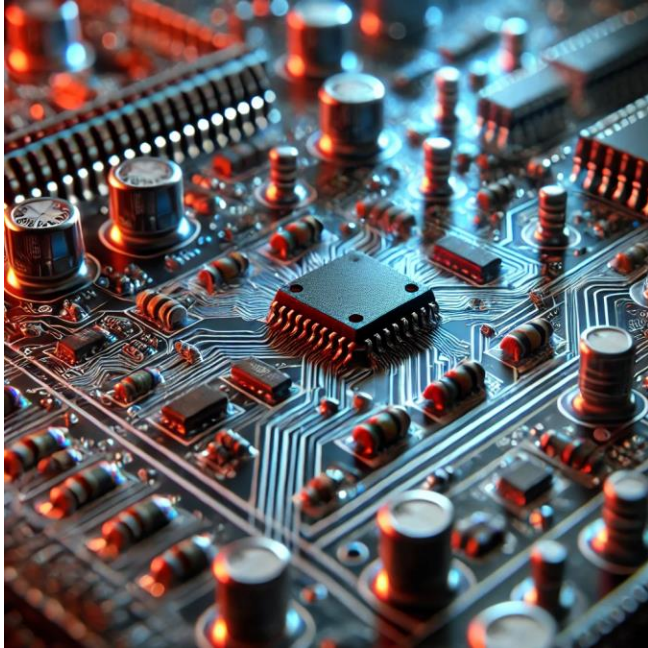
# Les étapes du montage :

- **Initialisation** : le **robot** est allumé et les différents composants sont initialisés, notamment les capteurs, les moteurs, et la carte arduino.
- **Collecte de données** : le capteurs HC-SR04 collecte des informations sur l'environnement du **robot**, telles que la distance à des obstacles.
- **Analyse des données** : la carte **Arduino** analyse les données collectées par les capteurs pour déterminer la présence et la distance d'un obstacle.
- **Prise de décision** : en fonction des informations collectées et analysées, la carte **Arduino** décide de la meilleure façon d'éviter l'obstacle.
- **Contrôle des moteurs** : la carte **Arduino** envoie des commandes aux moteurs pour effectuer les mouvements nécessaires pour éviter l'obstacle.



- **Boucle Continue :**





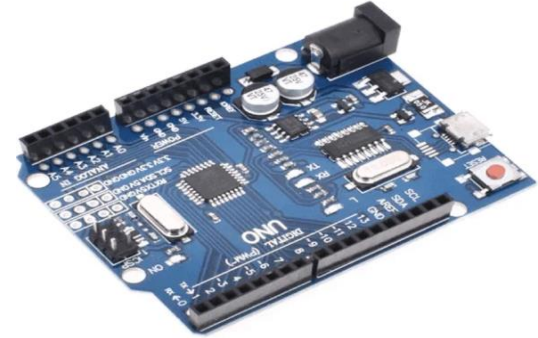
# 03

## Matériel nécessaire

# Les composants

- Carte Arduino Uno :

- L'Arduino UNO est la carte de contrôle principale du robot. Elle est programmée pour lire les données des capteurs et contrôler les moteurs en fonction des informations reçues.



- Châssis de robot 4 roues (4WD) :

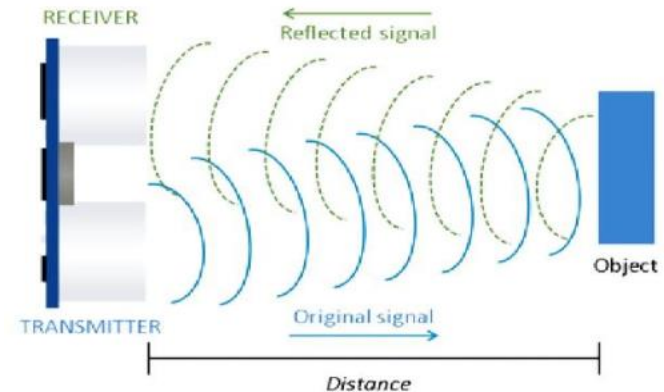
- Le robot aura besoin de moteurs pour se déplacer et de roues pour assurer la mobilité. Les moteurs peuvent être de types différents en fonction de la conception du robot, mais ils doivent être compatibles avec le module L298N.

- Le châssis est la structure du robot sur laquelle sont montés les autres composants. Il peut être fabriqué à partir de divers matériaux tels que le plastique, le bois ou le métal, et sa conception dépendra de la taille et de la forme souhaitées pour le robot



## • Capteurs HC-SR04 :

-Ces capteurs à ultrasons sont utilisés pour détecter la distance entre le robot et l'objet à suivre. Chaque capteur est composé d'un émetteur et d'un récepteur ultrasonique. L'émetteur envoie un signal ultrasonique qui rebondit sur l'objet et revient au récepteur. En mesurant le temps que prend le signal pour revenir, le capteur peut calculer la distance entre le robot et l'objet.



## • Module L298N :

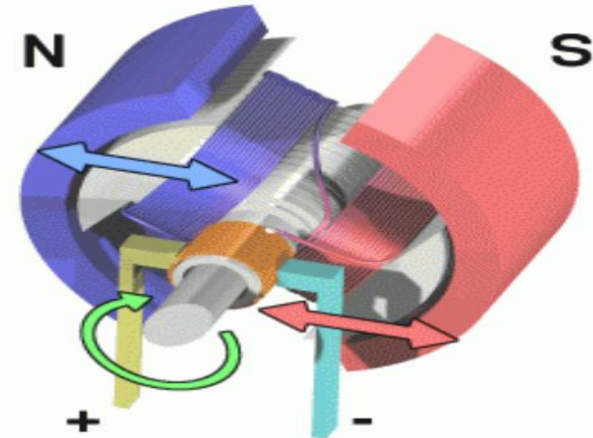
- Le module **L298N** est un module de commande de moteur bidirectionnel utilisé pour contrôler la vitesse et la direction de rotation de moteurs DC ou de moteurs à courant alternatif à faible puissance. Il est basé sur le circuit intégré L298N, qui est un contrôleur de moteur bidirectionnel double pont en H à haute tension et à haute intensité.
- Le module **L298N** est généralement utilisé avec des microcontrôleurs ou des cartes de développement comme Arduino pour contrôler les moteurs dans les projets de robotique ou d'automatisation. Il possède deux entrées de commande pour contrôler la direction de rotation des moteurs, ainsi que deux sorties de commande pour contrôler la vitesse des moteurs. Il peut également être utilisé pour contrôler la vitesse et la direction de rotation de moteurs pas à pas.
- Le module **L298N** est facile à utiliser et peut être alimenté avec une tension de 4,5 V à 46 V. Il est équipé de bornes de connexion pour le moteur, de bornes de connexion pour l'alimentation et de bornes de connexion pour la commande. Il est également équipé de dissipateurs thermiques pour protéger le circuit intégré L298N des surchauffes.



- **Moteurs DC:**

– Un **moteur DC de 5 volts** est un moteur électrique qui fonctionne à une tension de 5 volts. Les moteurs DC sont des moteurs qui utilisent un courant continu pour fonctionner et qui peuvent être alimentés par une variété de sources, notamment des piles, des panneaux solaires et des alimentations électriques.

Les **moteurs DC de 5 volts** sont couramment utilisés dans les applications où une faible tension est requise, telles que les projets de bricolage et de loisir, les jouets et les appareils électroniques portables. Ils sont également souvent utilisés dans les projets de robotique et de domotique, où ils peuvent être utilisés pour contrôler les mouvements des robots et des dispositifs automatisés.



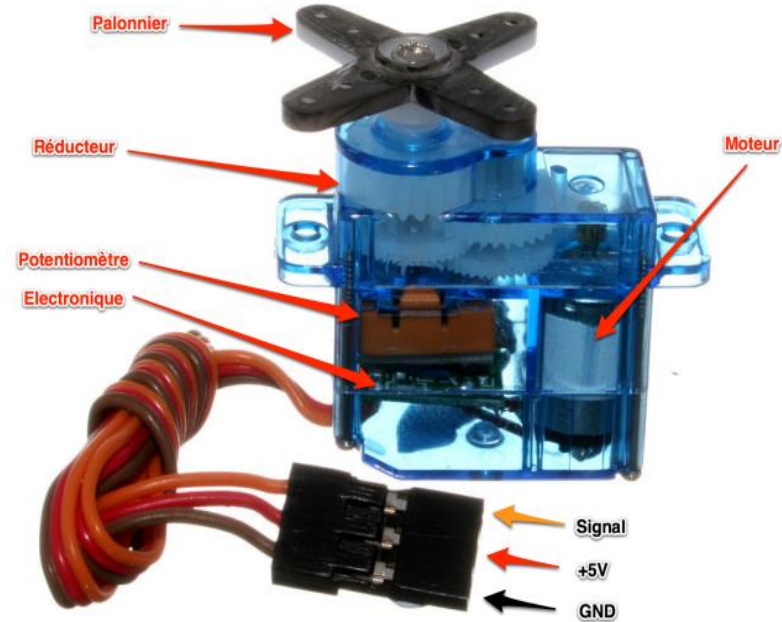
## • Servomoteur:

-Les **servomoteurs** sont souvent utilisés dans les projets de robotique pour contrôler la position et la vitesse de mouvement des objets. Ils peuvent être utilisés pour faire bouger des bras robotiques, des roues, des caméras, etc.

-Un **servomoteur** peut être utilisé pour orienter la direction de détection des obstacles dans un robot détecteur d'obstacles. Le **servomoteur** peut être relié à un capteur ultrasonique tel que le HC-SR04 pour déterminer la distance à un objet.

-Cela permet de couvrir une zone plus large et de fournir des informations plus complètes sur l'environnement de l'objet.

-Lorsque le robot se déplace, le **servomoteur** peut être programmé pour faire pivoter le capteur ultrasonique dans différentes directions pour détecter les obstacles dans son environnement.



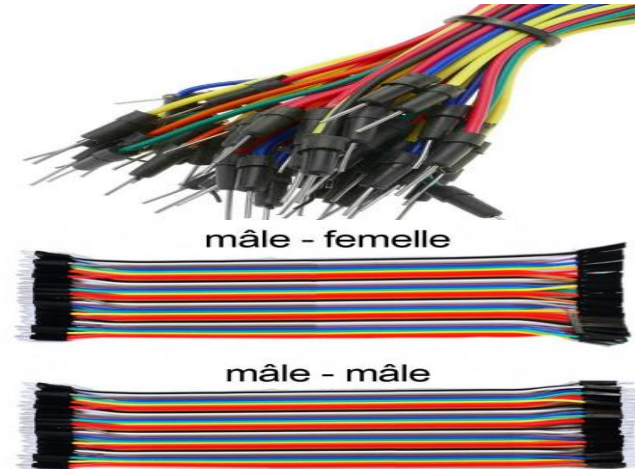


## • Fils de connexion:

─ Les fils de connexion sont des fils électriques utilisés pour connecter des composants électroniques à la carte Arduino. Ils sont généralement utilisés pour connecter des capteurs, des actionneurs, des afficheurs et d'autres composants à la carte Arduino afin de créer des circuits électroniques.

– Il existe deux types de fils de connexion: les fils de connexion mâle-mâle et les fils de connexion mâle-femelle. Les fils de connexion mâle-mâle sont utilisés pour connecter des composants qui ont tous deux des broches mâles, tandis que les fils de connexion mâle-femelle sont utilisés pour connecter des composants avec une broche mâle et une broche femelle.

– Les fils de connexion sont généralement fabriqués en cuivre ou en alliage de cuivre et sont revêtus d'un isolant en plastique pour protéger les fils électriques et empêcher les courts-circuits. Ils sont disponibles dans une variété de couleurs pour aider à identifier et organiser les différents fils dans un circuit.



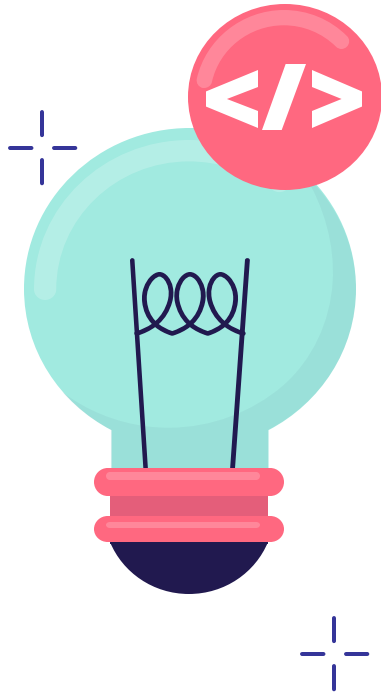


- Batterie Li-ion 18650:

- Le robot aura besoin d'une source d'alimentation électrique pour fonctionner. Cela peut être une batterie rechargeable ou des piles, en fonction de la portabilité et de la durée d'utilisation souhaitées.

Capacité: Nominal 4000mAh  
Tension nominale 3,6 V - 3,7 V  
Tension de fin de cycle de charge  $4.20V \pm 0.05V$   
Courant de décharge max: 10A  
Tension de fin (coupure) 2,5 V  
Pôle + : dépasse  
Composition: Li-ion  
Circuit de protection : Oui  
Diamètre : 18,4 mm  
Hauteur : 69,2 mm





# 04

## Montage du robot



# Montage du robot

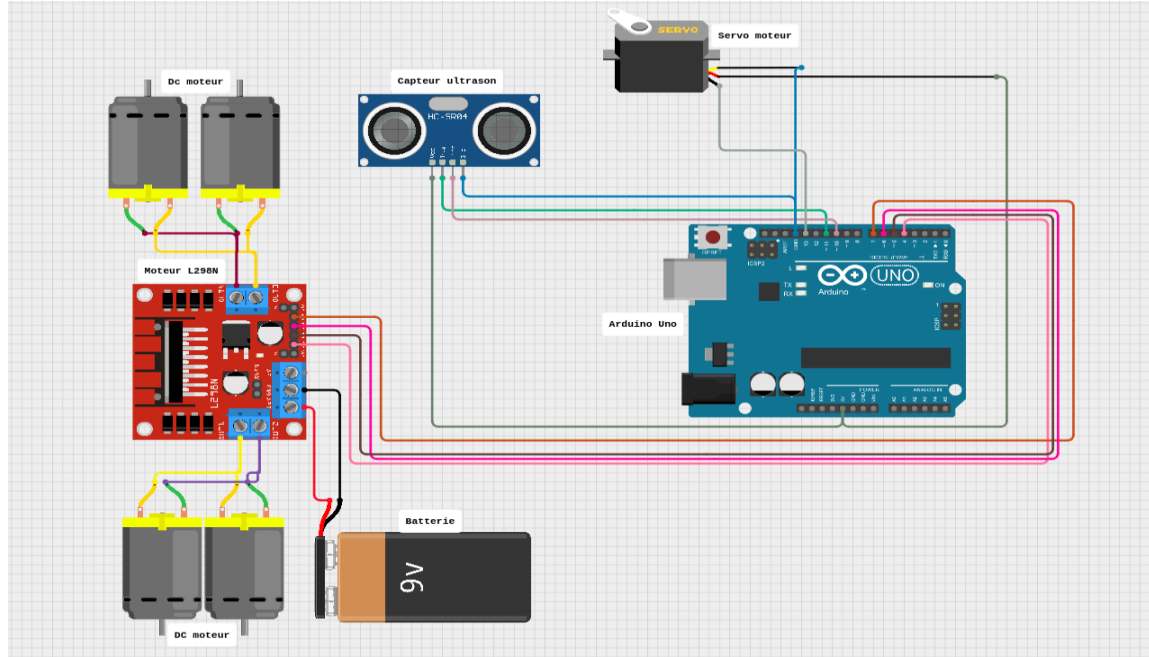


Schéma électronique du robot



## 1.Connexion du module L298N et des moteurs DC

- .Reliez les **moteurs DC** aux sorties **OUT1, OUT2, OUT3, OUT4** du L298N.
- .Connectez **IN1, IN2, IN3, IN4** du L298N aux **pins 7, 6, 5, 4** de l'Arduino.
- .Branchez la **batterie 9V** au L298N et reliez son **GND** au GND de l'Arduino.

## 2.Connexion des capteurs et actionneurs

- .**HC-SR04** : VCC → 5V, GND → GND, Trig → Pin 8, Echo → Pin 9.
- .**Servo moteur** : VCC → 5V, GND → GND, Signal → Pin 3.

## 3.Alimentation et vérifications

- .Reliez le **GND du L298N** au **GND de l'Arduino** pour un circuit commun.
- .Vérifiez toutes les connexions avant d'alimenter le circuit.
- .Testez chaque composant individuellement avant d'intégrer le code final.





Sens des roues dans les différentes marches

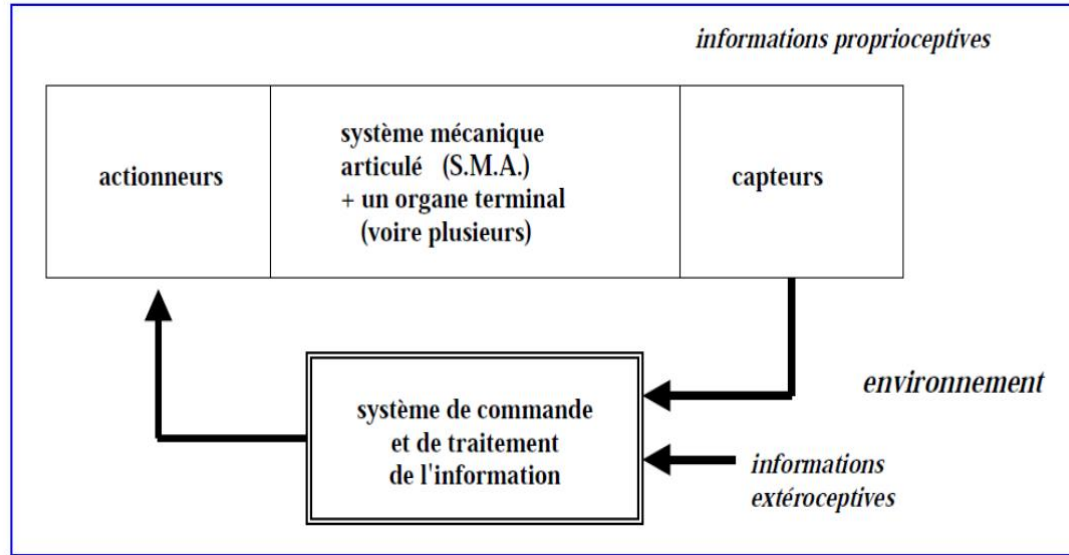
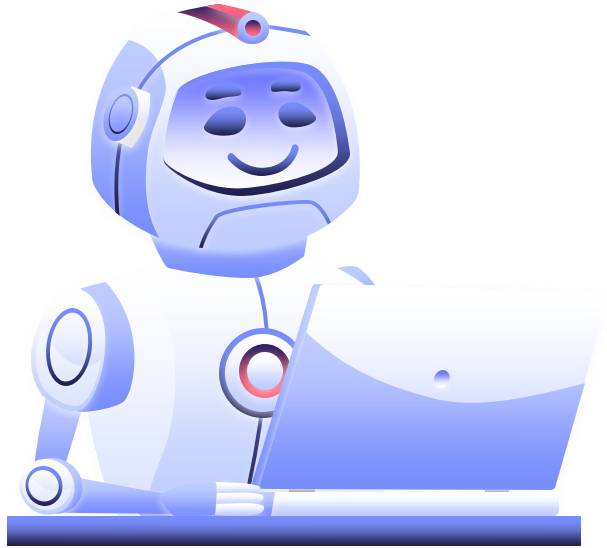


Schéma synoptique d'un robot



# 06

## Programmation et Implémentation



# La conception du logiciel sous l'environnement Arduino



La partie logicielle est la partie la plus complexe de ce projet. Le programme est évidemment écrit en langage Arduino.

Voici le programme du **robot commandé par la carte Arduino UNO** qui détecte et évite les obstacles en utilisant un capteur de distance à ultrasons.

Ce code utilise la bibliothèque **HCSR04** pour lire les données reçues par le capteur ultrason HC-SR04 et la bibliothèque **Servo** pour le Servomoteur





## Code :



```
robot_evitent_obstacle.ino
1  #include<HCSR04.h>
2  #include<Servo.h>
3
4  //Définition des broches de contrôle du L298N
5  const int ENA = 5; // Broche de vitesse pour moteur gauche
6  const int ENB = 6; // Broche de vitesse pour moteur droit
7  const int IN1 = 7; // Direction moteur gauche
8  const int IN2 = 8;
9  const int IN3 = 9; // Direction moteur droit
10 const int IN4 = 10;
11 //Définition des broches du capteur ultrason HC-SR04
12 const int brocheTrig = 3;
13 const int brocheEcho = 2;
14 //Déclaration du servomoteur
15 Servo servoTete;
16 int distanceObstacle, distanceDroite, distanceGauche;
17
18 void setup(){
19     Serial.begin(9600);
20     servoTete.attach(11);
21     servoTete.write(90); // Position initiale
22     // Configuration des broches des moteurs en sortie
23     pinMode(ENA, OUTPUT);
24     pinMode(ENB, OUTPUT);
25     pinMode(IN1, OUTPUT);
26     pinMode(IN2, OUTPUT);
27     pinMode(IN3, OUTPUT);
28     pinMode(IN4, OUTPUT);
29     // Activer les moteurs avec une vitesse initiale
30     analogWrite(ENA, 150); // Vitesse du moteur gauche (0-255)
31     analogWrite(ENB, 150); // Vitesse du moteur droit (0-255)
32 }
```

## Code :



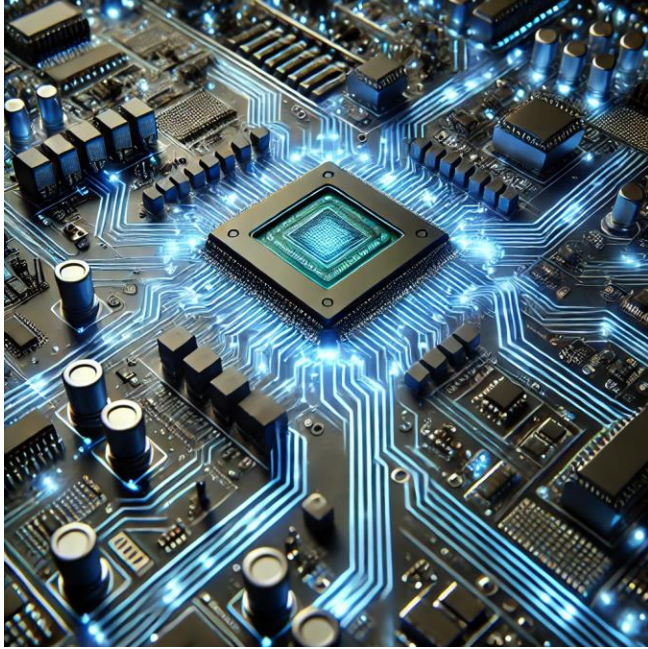
```
33 void loop(){
34     distanceObstacle = mesurerDistance();
35     Serial.print("Distance détectée : ");
36     Serial.println(distanceObstacle);
37     delay(50);
38     if(distanceObstacle > 0 && distanceObstacle < 15){
39         arreterMoteurs();
40         delay(500);
41         reculer();
42         delay(500);
43         arreterMoteurs();
44         // Scanner droite et gauche
45         distanceDroite = scannerDirection(10);
46         distanceGauche = scannerDirection(170);
47         servoTete.write(90);
48         delay(500);
49
50         if(distanceDroite > distanceGauche){
51             tournerDroite();
52         }else{
53             tournerGauche();
54         }
55     }else if(distanceObstacle > 0){
56         avancer();
57     }
58 }

59 //Fonction pour Mesurer la distance
60 int mesurerDistance(){
61     digitalWrite(brocheTrig, LOW);
62     delayMicroseconds(2);
63     digitalWrite(brocheTrig, HIGH);
64     delayMicroseconds(10);
65     digitalWrite(brocheTrig, LOW);
66
67     long duree = pulseIn(brocheEcho, HIGH);
68     return duree * 0.034 / 2; // Conversion en cm
69 }
70 //Fonction pour scanner une direction avec le capteur ultrason
71 int scannerDirection(int angle){
72     servoTete.write(angle);
73     delay(500);
74     return mesurerDistance();
75 }
```

## Code :



```
76 //Fonction pour avancer le robot
77 void avancer(){
78     digitalWrite(IN1, HIGH);
79     digitalWrite(IN2, LOW);
80     digitalWrite(IN3, HIGH);
81     digitalWrite(IN4, LOW);
82 }
83 //Fonction pour reculer le robot
84 void reculer(){
85     digitalWrite(IN1, LOW);
86     digitalWrite(IN2, HIGH);
87     digitalWrite(IN3, LOW);
88     digitalWrite(IN4, HIGH);
89 }
90 //Fonction pour tourner à gauche
91 void tournerGauche(){
92     digitalWrite(IN1, LOW);
93     digitalWrite(IN2, HIGH);
94     digitalWrite(IN3, HIGH);
95     digitalWrite(IN4, LOW);
96     delay(1000);
97     arreterMoteurs();
98 }
99 //Fonction pour tourner à droite
100 void tournerDroite(){
101     digitalWrite(IN1, HIGH);
102     digitalWrite(IN2, LOW);
103     digitalWrite(IN3, LOW);
104     digitalWrite(IN4, HIGH);
105     delay(1000);
106     arreterMoteurs();
107 }
108 //Fonction pour stopper le robot
109 void arreterMoteurs(){
110     digitalWrite(IN1, LOW);
111     digitalWrite(IN2, LOW);
112     digitalWrite(IN3, LOW);
113     digitalWrite(IN4, LOW);
114 }
115
```



# MERCI POUR VOTRE ATTENTION



Elkhoulati Yahya



elkhoulati.yahya@yahoo.com