

Compte-rendu du TP de Robotique

Etudiants: *OURO-AKONDO M Barak, SY Oumou Kalsoum, Béatrice Diome & NYNA Darlène*

Animateur(s): *M. Arsène COULIBALY & M. Aymar TCHOTA*

1 Objectifs

L'objectif de ce TP est de réaliser un robot roulant éviteur d'obstacles.

2 Composition

2.1 2 moteurs électriques :

Les moteurs électriques sont responsables de la propulsion du robot. Ils convertissent l'énergie électrique en mouvement mécanique pour permettre au robot de se déplacer. Ils sont connectés aux deux roues par un système d'engrenage, pour fournir une traction et permettre au robot de se déplacer vers l'avant, l'arrière et de tourner.

2.2 Driver :

Le driver est un composant électronique qui agit comme un pont entre l'arduino et les moteurs. Il reçoit des signaux électriques de l'Arduino et les amplifie pour fournir la puissance nécessaire aux moteurs. Le driver contrôle également la vitesse et la direction des moteurs en fonction des signaux envoyés par l'Arduino.

2.3 Le capteur ultrason

Le capteur envoie des ultrasons qui reviennent après avoir rencontré un obstacle. Le capteur permettra de déterminer la distance qui sépare le robot de l'obstacle et ainsi de prendre une décision.


2.4 Carte Arduino :

La carte Arduino est le cerveau du robot. Le capteur est connecté à la carte Arduino pour détecter les obstacles et prendre des décisions en conséquence. En fonction des informations des capteurs, l'Arduino envoie des signaux au driver pour contrôler les moteurs et éviter les obstacles.

3 Programme

avance_arriere | Arduino 1.8.19

Fichier Édition Croquis Outils Aide



```
// Moteur 1
int ENA = 2;
int IN1 = 3;
int IN2 = 4;

// Moteur 2
int ENB = 8;
int IN3 = 6;
int IN4 = 5;

int Trig = 11;
int Echo = 13;

float dist;
float temps;
int d;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  pinMode(ENA, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(ENB, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);

  pinMode(Trig, OUTPUT);
  pinMode(Echo, INPUT);
}
```

```

void loop() {
    // put your main code here, to run repeatedly
    Avance(100);
    delay(5000);
    arret();
    delay (2000);
    Arriere (100);
    delay (5000);
    sonic();

}

```

```

void M1Control(int vitesse, bool sens){
    digitalWrite(ENA, vitesse);
    digitalWrite(IN1, sens);
    digitalWrite(IN2, !sens);
}

```

```

void M2Control(int vitesse, bool sens){
    digitalWrite(ENB, vitesse);
    digitalWrite(IN3, sens);
    digitalWrite(IN4, !sens);
}

```

```

void Avance (int vitesse){
    M1Control(100, true);
    M2Control(100, true);
}

```

```

void Arriere (int vitesse){
    M1Control(100, false);

```

```

void arret(){
    M1Control(0, true);
    M2Control(0, true);
}

```

```

void sonic(){
    digitalWrite(Trig, LOW);
    delayMicroseconds(10);
    digitalWrite(Trig, HIGH);
    delayMicroseconds(10);
    digitalWrite(Trig, LOW);
    temps = pulseIn(Echo,HIGH);
    d = (0.034*temps)/2;
    Serial.print("Distance=");
    Serial.print (d);
    Serial.println("cm");
    //return d;

}

```

Le code commande le robot à l'aide de deux moteurs (Moteur 1 et Moteur 2) et utilise un capteur de distance à ultrasons pour mesurer la distance des obstacles.

1. Déclaration des broches : Les premières lignes déclarent les broches utilisées pour connecter les différents composants. Les moteurs sont connectés aux broches ENA, IN1, IN2 (pour le Moteur 1) et ENB, IN3, IN4 (pour le Moteur 2). Le capteur à ultrasons est connecté aux broches Trig (pour l'émission du signal) et Echo (pour la réception du signal).
2. Configuration initiale (setup) : La fonction `setup()` est exécutée une fois au démarrage. Dans cette fonction, la communication série est initialisée avec une vitesse de 9600 bauds à l'aide de `Serial.begin(9600)`. Ensuite, les broches sont configurées en tant que sorties (OUTPUT) ou entrées (INPUT) à l'aide de la fonction `pinMode()`.
3. Boucle principale (loop) : La fonction `loop()` est exécutée en boucle continue après la configuration initiale. Voici ce qu'elle fait :
 - Appel à la fonction `Avance(100)` : Cela permet de faire avancer le robot avec une vitesse de 100 (la vitesse est ajustable).
 - Attente de 5 secondes (`delay(5000)`).
 - Appel à la fonction `arret()` : Cela arrête les moteurs.
 - Attente de 2 secondes (`delay(2000)`).
 - Appel à la fonction `Arriere(100)` : Cela permet de faire reculer le robot avec une vitesse de 100.
 - Attente de 5 secondes (`delay(5000)`).
 - Appel à la fonction `sonic()` : Cela active le capteur à ultrasons pour mesurer la distance des obstacles.
4. Fonctions de contrôle des moteurs : Les fonctions `M1Control()` et `M2Control()` sont utilisées pour contrôler les moteurs individuellement. Elles prennent en paramètre la vitesse du moteur (de 0 à 255) et le sens (`true` pour l'avant, `false` pour l'arrière). Ces fonctions ajustent les signaux des broches ENA, IN1, IN2 (pour le Moteur 1) et ENB, IN3, IN4 (pour le Moteur 2) en conséquence.
5. Autres fonctions :
 - La fonction `Avance()` appelle les fonctions `M1Control()` et `M2Control()` avec la vitesse spécifiée pour faire avancer le robot dans la même direction.
 - La fonction `Arriere()` appelle les fonctions `M1Control()` et `M2Control()` avec la vitesse spécifiée pour faire reculer le robot dans la même direction.
 - La fonction `arret()` appelle les fonctions `M1Control()` et `M2Control()` avec une vitesse de 0 pour arrêter les moteurs.

- La fonction `sonic()` contrôle le capteur à ultrasons pour mesurer la distance des obstacles. Elle envoie un signal ultrasonique (Trig) et mesure le temps que met le signal à revenir (Echo). En utilisant la formule de la vitesse du son dans l'air (34 cm/ms), elle calcule la distance et l