

## Compte-rendu de TP

Etudiants :

*Jacintho MPETEYE, Oussama EDDOUG, ZINE Bahaeddine, Iwan MBOULOUNGOU*

Matière : Robotique

Animateur : M. Tadjou TCHOTA, M. COULIBALY Arsène

---

### **I- But du travail pratique**

Le but de ce travail pratique était principalement de mettre en application les notions théoriques vues en cours de robotique et d'implémenter les parties acquisition, traitement et actionnement dans un robot. Nous avons donc à cet effet eu à travaillé avec un capteur, des actionneurs et une carte électronique pour le traitement.

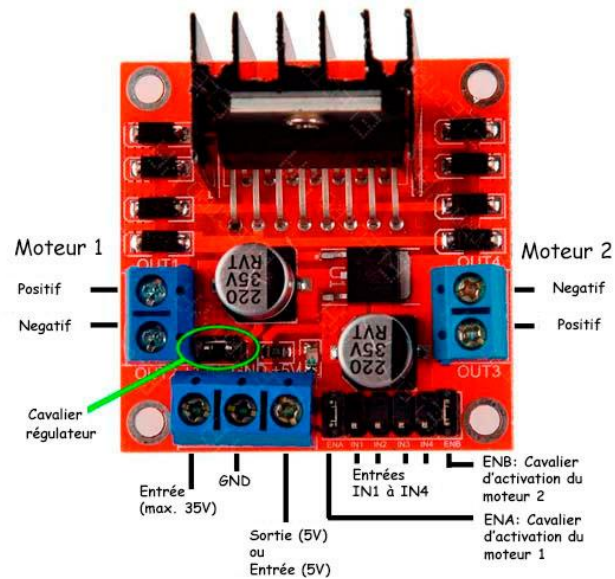
### **II- Composants du circuit**

Composants utilisés :

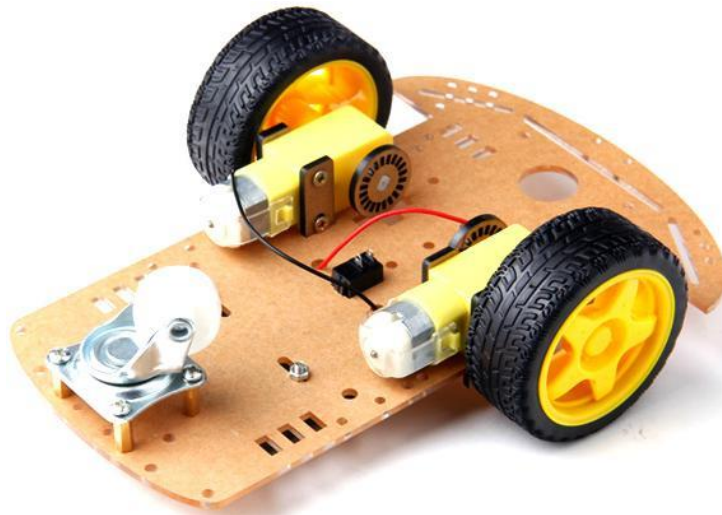
1. Arduino Méga : L'Arduino Méga est une carte de développement basée sur microcontrôleur qui offre une puissance de traitement et des capacités d'entrée/sortie étendues. Il a été utilisé comme cerveau du robot pour gérer la logique de contrôle et l'interfaçage avec les autres composants.



2. Drivers L298N : Les drivers L298N sont des circuits intégrés de contrôle de moteur utilisés pour contrôler les moteurs à courant continu. Ils fournissent un contrôle bidirectionnel et une puissance suffisante pour alimenter les moteurs.



3. Moteurs à courant continu monté de part et d'autre du châssis de la voiture : Les moteurs à courant continu sont les actionneurs principaux du robot. Ils ont été utilisés pour le mouvement du robot dans différentes directions. :



4. Capteur ultrasons : Le capteur ultrasons est un capteur de distance utilisant des ondes sonores pour détecter les objets environnants. Il a été utilisé pour détecter les obstacles et permettre au robot de les contourner.



### III- Explication du programme

Déclaration des ports pour la connexion du driver :

```
const int ENA = 10;
const int IN1 = 11;
const int IN2 = 12;
const int IN3 = 8;
const int IN4 = 9;
const int ENB = 7;
```

Déclaration des ports pour la connexion du capteur ultrasons et d'une variable pour stocker la distance mesurée par la fonction *sonic* que nous détaillerons par la suite :

```
const int pinTrig = 2;
const int pinEcho = 3;

float dist;
```

Fonction **Setup** configuration des modes entrées/sorties :

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

  pinMode(pinTrig, OUTPUT);
  pinMode(pinEcho, INPUT);

  // digitalWrite(pinTrig, LOW);

  Serial.begin(9600);
}
```

Le programme de la boucle principale est le suivant :

```
void loop() {  
  
    dist = sonic();  
    move_motor(70,true);  
    if (dist<5.00){  
        hold_down(2000);  
        move_motor(70,false);  
        delay(2000);  
        hold_down(2000);  
        run_motor1(70,true);  
        delay(2000);  
        hold_down(200);  
    }  
}
```

Comme on peut le voir tout commence par la lecture de la distance par le capteur. Par défaut le robot est censé aller droit devant lui. Lorsqu'un obstacle est détecté (distance inférieure à 5 cm), un autre bout de programme est mis en marche qui consiste à faire reculer le robot pendant un certain temps, à le faire tourner de côté pendant un autre moment. La fonction `hold_down` est utilisée pour temporiser le robot (l'immobiliser pendant un temps en gros).

```
void hold_down(int milli_sec){  
    stop_motor1();  
    stop_motor2();  
    delay(milli_sec);  
}
```

Fonction `run_motor<num>` contrôle le moteur numéro <num>.

```
void run_motor1(int vitesse, bool sens){  
    analogWrite(ENA,vitesse);  
    digitalWrite(IN1,sens);  
    digitalWrite(IN2,!sens);  
}  
  
void run_motor2(int vitesse, bool sens){  
    analogWrite(ENB,vitesse);  
    digitalWrite(IN3,sens);  
    digitalWrite(IN4,!sens);  
}
```

Fonction `stop_motor<num>` stoppe le moteur numéro <num>.

```

void stop_motor1() {
    analogWrite(ENA, 0);
    digitalWrite(IN1, false);
    digitalWrite(IN2, false);
}

void stop_motor2() {
    analogWrite(ENB, 0);
    digitalWrite(IN3, false);
    digitalWrite(IN4, false);
}

```

La fonction **move\_motor** fait soit avancer le robot soit le fait reculer selon la valeur de sens (true est pour l'avance).

```

void move_motor(int vitesse, bool sens) {
    run_motor1(vitesse, sens);
    run_motor2(vitesse, sens);
}

float sonic() {
    long temps;
    float distance;

    //Envoi du signal
    digitalWrite(pinTrig, HIGH);
    delayMicroseconds(10);
    digitalWrite(pinTrig, LOW);
    //Réception du signal
    temps = pulseIn(pinEcho, HIGH);

    //Gestion des erreurs de mesure
    if (temps > 25000) {
        Serial.println("Echec de la mesure");
        return 20;
    }

    else {
        //Détermination de la distance en cm
        temps = temps/2;
        distance = (temps*340)/10000.0;
        Serial.print("Distance: ");
        Serial.print(distance);
        Serial.println(" cm");
    }
    delay(2); //brève attente
    return distance; //retourner a valeur de la distance
}

```

#### **IV – Résultats et conclusion**

Le système a été capable de détecter les obstacles à l'aide du capteur ultrasons et d'effectuer des mouvements appropriés pour les éviter. Le robot a pu se déplacer de manière autonome dans un environnement avec succès.

Le travail pratique a démontré l'utilisation efficace de l'Arduino Mega, des drivers L298N, des moteurs à courant continu et du capteur ultrasons pour contrôler un robot. Le système développé a permis au robot de détecter les obstacles et de naviguer en conséquence, ce qui représente une application pratique dans le domaine de la robotique.