

Objectif : Démarrer la programmation des moteurs et de la détection d'obstacle

En attente de la livraison de notre châssis composés de 4 moteurs 12V CC, on réalise les branchements qu'on réalisera sur nos 4 moteurs à partir du moteur fourni pour actionner notre tourelle (moteur CC).

Pour ce faire on doit déjà s'assurer du fait que nos 6 capteurs ultrasons pour la détection d'obstacle placés sur notre robot militaire fonctionnent correctement, c'est à dire qu'ils nous renvoient la bonne distance à un obstacle:

I-Détection d'obstacle

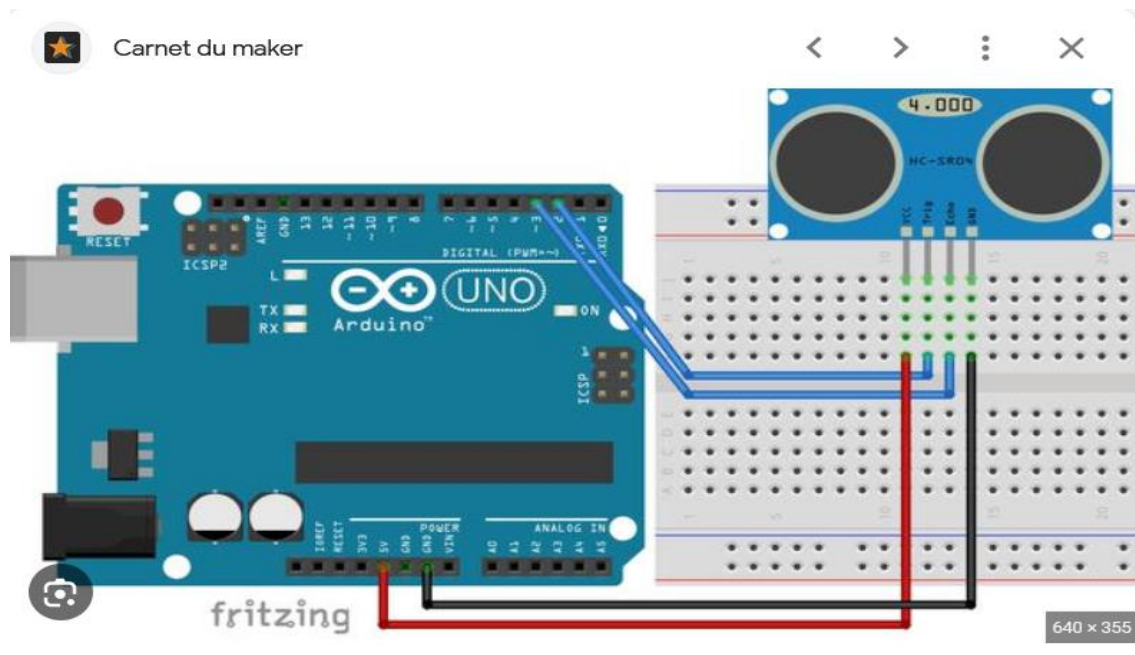
La programmation du code permettant de donner la distance entre un capteur ultrasonore et un obstacle est donné ici : celui-ci nous permettra de donner une condition d'activation ou de désactivation des moteurs en fonction d'une distance minimale donnée entre le capteur ultrasonore et l'obstacle.

Sonar_hcsr04.ino

```
1
2  int lecture;
3  int distance;
4  int trig=9;
5  int echo=8;
6
7 > void setup() { ...
8
9  Serial.begin(9600);
10  pinMode(trig,OUTPUT);
11  pinMode(echo,INPUT);
12
13
14  }
15
16  void loop() {
17    // put your main code here, to run repeatedly:
18    digitalWrite(trig,LOW);
19    delayMicroseconds(2);
20    digitalWrite(trig,HIGH);
21    delayMicroseconds(10);
22    digitalWrite(trig,LOW);
23    lecture=pulseIn(echo,HIGH);
24    long distance=lecture*0.017;
25    Serial.println(distance);
26    delay(60);
27
28  }
```

Le branchement réalisé pour le montage est le suivant : le programme nous renvoie la distance entre le détecteur d'obstacle et l'objet détecté jusqu'à une distance d'environ 5 mètres.

Le programme a été testé précédemment en cours et fonctionne.



II-Programmation d'un moteur sur Arduino

Avec le kit Moteur + Carte de Contrôle initialement destiné à la rotation de notre tourelle, nous avons fait des recherches dans le but de faire fonctionner ce moteur.

D'une part j'ai téléchargé puis analysé la datasheet de la carte de contrôle du moteur fourni avec, ce qui nous a permis de brancher correctement le moteur et de le faire fonctionner tout en faisant varier sa vitesse à l'aide d'un potentiomètre.

Voir vidéo dans le github.

Une nouvelle problématique m'est venue à l'esprit lors du fonctionnement du moteur CC destiné à la rotation de la tourelle :

Comment changer le sens de rotation du moteur sans avoir à inverser les fils connectés au moteur ? (utilité pour passer de la marche avant à la marche arrière ou encore pour faire tourner la tourelle)

Pour ce faire, on doit se servir d'un pont en H :

Pont H

Un **transistor** est un composant électronique qui fonctionne comme un **interrupteur**, que l'on peut commander à l'aide d'une tension électrique. Le microcontrôleur n'est pas capable de fournir la puissance nécessaire à faire tourner un moteur, mais il peut contrôler un transistor.

Le pont H est un montage de quatre transistors, qui permettent de connecter le moteur à l'alimentation de diverses façons.

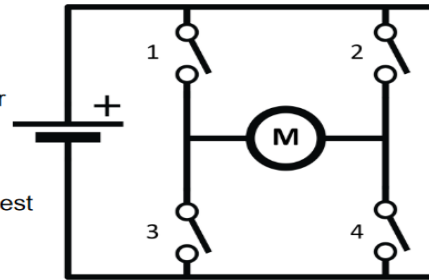
Voici deux exemples :

Marche avant

En fermant les interrupteurs 1 et 4, le circuit passe par le moteur dans un sens.

Marche arrière

En fermant les interrupteurs 2 et 3, le circuit est maintenant connecté dans l'autre sens.

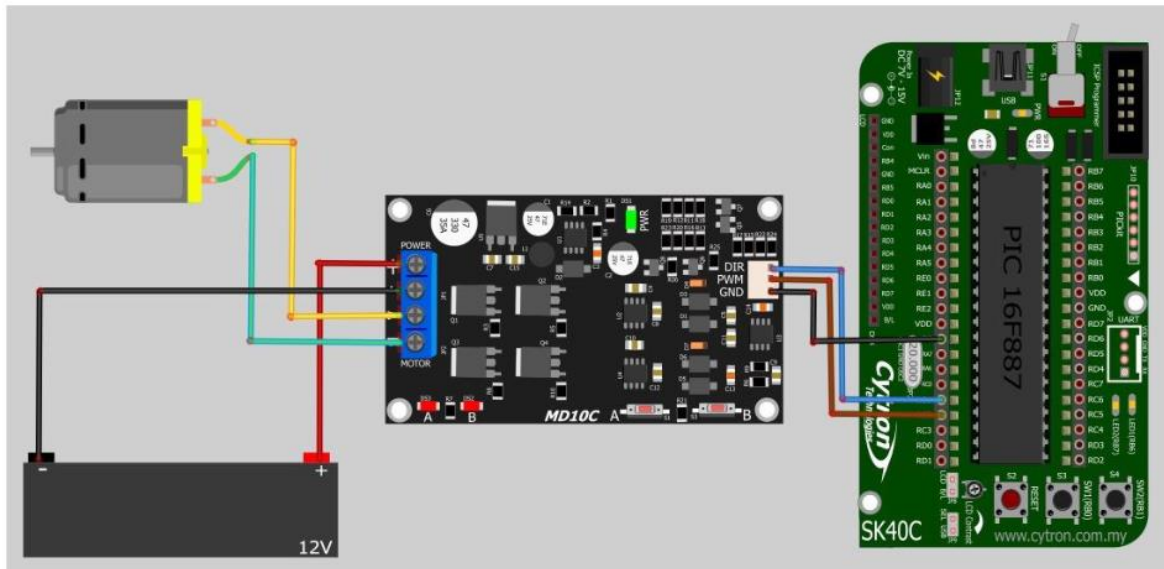


A partir du pont en H donné destiné à la carte arduino UNO, donné en cours, nommé Cytron MD10C, j'ai téléchargé et analysé sa datasheet, dans le but de construire un circuit sur Arduino capable de changer le sens de rotation du moteur.

Après analyse, de la datasheet c'est le le pin nommé DIR du pont en H qui en fonction de s'il reçoit une entrée HIGH ou LOW, va faire tourner le moteur dans le sens horaire ou anti-horaire.

On place également un potentiomètre dans le but de faire varier la vitesse de rotation du moteur.

Montage réalisé:



Voici le code Arduino permettant de réaliser ce qui a été expliqué précédemment.

```
1  int potentio = A0;
2  int moteur = 3;
3  int dir = 10;
4
5
6  void setup() {
7
8    Serial.begin(9600);
9    pinMode(potentio,INPUT);
10   pinMode(moteur,OUTPUT);
11   pinMode(dir,OUTPUT);
12   digitalWrite(dir,LOW) //on met un LOW pour faire tourner le moteur en sens horaire et en HIGH pour le sens anti-horaire
13
14 }
15
16 void loop() {
17
18   int valeurpotentio= analogRead(potentio);
19   Serial.println(valeurpotentio);
20   int valeurmoteur= map(valeurpotentio,0,1023,0,255);
21   analogWrite(moteur,valeurmoteur);
22
23 }
24
```

III-soudure

Pendant cette séance, nous avons rencontré un premier problème : une des broches de notre moteur (entrée +) s'est cassée, on a donc réalisé une soudure à l'étain en reliant les bornes + et – du moteur à des fils arduino (sans succès : le moteur ne tournait pas, la connexion à la borne lors de la soudure devait être mal réalisée).

On a alors emprunté un châssis d'un projet précédent qui possédait un moteur 12V CC dans le but de tester notre programme.

Voir Vidéo dans le github.

IV- Objectifs pour l'étape suivante :

Créer un programme capable d'activer un moteurs ou de le désactiver après avoir détecté un obstacle.