



Séance 2

Au cours de la séance précédente, vous avez pris connaissance du fonctionnement d'un moteur à courant continu ainsi que de la constitution et du fonctionnement d'un pont en H et vu aussi la nécessité d'un étage de puissance pour le moteur.

But de la séance

Vous aurez à effectuer le branchement du moteur à courant continu étudié au pont en H et à le piloter par programmation au moyen de l'Arduino UNO afin de le faire tourner dans les deux sens.

Vous devrez aussi brancher et programmer un détecteur d'obstacles. Pour cela, vous utiliserez la carte Arduino UNO virtuellement. La réalisation du montage, la programmation et la vérification du fonctionnement se fera grâce au logiciel de conception assistée par ordinateur (CAO) TINKERCAD.

1.Prise en main du logiciel TINKERCAD

Voici le lien pour simuler en ligne le fonctionnement de notre système. Le logiciel permet de faire tourner le moteur CC sans utiliser un étage de puissance.

<https://www.tinkercad.com/learn/circuits>

2.Génération du signal PWM et raccordement du moteur

En vous aidant du guide de référence Arduino et des liens donnés :

<https://www.arduino.cc/reference/en/>

<https://www.locoduino.org/>

- a- Expliquez comment fonctionne l'instruction `analogWrite(...)` ? Générez alors le signal PWM permettant de faire varier la vitesse du moteur CC.
- b- En vous basant sur la documentation du constructeur, trouvez la fréquence de ce signal. Quel type de filtre faut-il utiliser pour ne conserver de ce signal que sa partie continue ? Réalisez ce filtre.
- c- Faire un schéma du montage (incluant la carte, le pont et le moteur) pour montrer où vous branchez ce signal PWM filtré.

3-Détecteur d'obstacles

1-Expliquer l'utilité d'un capteur dans un système embarqué. Puis l'intérêt d'un détecteur d'obstacles pour notre projet .

2-Etudiez le principe de fonctionnement de la détection de distance par ultrasons.

3- En vous basant sur la datasheet du HC SR04, dites :

a-quel est le rôle de la pin Trigger ? Quel signal doit recevoir cette pin ?

b-Que retourne l'instruction pulseIn()? Quels sont ses arguments ?

4- Réalisation matérielle et logicielle

Dans un premier temps, juste pour vous assurer que vos branchements sont corrects, vous pourrez programmer en utilisant les instructions Arduino et le moteur à courant continu simple. Puis, pour la partie programmation vous passerez au langage C (bas niveau), pour la configuration des ports puis pour l'instruction analogwrite. Pour ce premier bloc, vous pourrez conserver l'instruction gérant le temps : delay.

Pour la partie branchement, et en vous aidant de la datasheet du moteur (référence FIT0520), vous utiliserez le moteur à roue codeuse. Vous utiliserez alors une autre alimentation continue pour simuler l'étage de puissance.

Revoyez la structure d'un programme Arduino et la déclaration des variables pour l'écriture d'un sketch.

4.1Ecrivez l'algorithme d'un programme mesurant la distance d'un obstacle avec un HC SR04 et l'affichant sur la console.

4.2.La conversion analogique numérique et le quantum étant des notions déjà vues, implémentez le programme nécessaire et câblez le HC SR04 afin de réaliser le détecteur d'obstacles. Vérifiez son bon fonctionnement par l'affichage des distances sur la console Arduino.

4.3 Donnez l'algorithme pour la rotation du moteur dans les deux sens avec un temps de pose de 2s entre chaque rotation.

4.4 Réalisez le branchement du montage permettant de piloter la rotation du moteur CC. Implémentez le programme de pilotage et vérifiez son bon fonctionnement.