

LA MODULATION A LARGEUR D'IMPULSION

Nous allons nous intéresser à la **Modulation de Largeur d'Impulsion** (MLI ou PWM en anglais). Les réponses seront rédigées sur un compte rendu manuscrit. Les codes Arduino seront à rendre en fin de séance dans le dossier « Mes devoirs > MLI Arduino »

1. **Donner** la définition de ce qu'est une MLI.
2. **Rédiger** une explication succincte (5-10 lignes et un graphique) permettant de **décrire** ce qu'est une MLI, un rapport cyclique (que l'on notera α (alpha)) et une porteuse.
3. **Comparer** l'intérêt d'une MLI par rapport à une sortie digitale.

ETUDE DE LA MLI

La carte Arduino dispose d'entrées/sorties numériques spécifiques pour délivrer une tension modulée par une MLI.

4. **Repérer** le sigle correspondant à la MLI sur la carte Arduino.

Pour comprendre le fonctionnement de la MLI, nous allons commencer avec le code proposé en exemple dans le logiciel Arduino. Ouvrir le programme : « Fichier > Exemples > Analog > Fading ».

5. Le programme effectue une double boucle « for » paramétrée par « fadeValue ». La variable est ensuite écrite sur la sortie numérique 9, laquelle alimente une LED. **Proposer** un schéma de câblage permettant de connecter la LED sachant que :

- le courant doit être inférieur à 30 mA
- la LED entraîne une chute de tension de 1.4 V

Faire vérifier le schéma par le professeur

6. **Déterminer**, à l'aide du programme, la fréquence de la MLI (la fonction delay() peut être utilisée pour approcher la fréquence recherchée).
7. **Réaliser** le montage permettant l'allumage de la LED.

Faire vérifier le montage par le professeur

8. **Téléverser** et **afficher** sur le traceur série l'évolution de la sortie numérique 9 (pin sur laquelle est connectée la LED).
9. D'après le résultat obtenu à la question 4, **modifier** le programme afin que la MLI fonctionne aux alentours de 800Hz. **Afficher** sur le traceur série la sortie numérique 9. **Commenter** les éventuelles différences avec le résultat obtenu à la question 4.

FILTRAGE ANALOGIQUE

Nous allons générer un signal sinusoïdal de fréquence 50Hz à l'aide de la sortie MLI (pin 9) de l'Arduino. Le rapport cyclique sera modifié par le biais de la fonction analogWrite() décrite ici :

- <https://www.arduino.cc/en/Tutorial/PWM>.

10. Pour une fréquence d'échantillonnage de 800Hz, **calculer** le nombre d'échantillon d'une période sinusoïdale.

Pour répertorier l'ensemble des valeurs angulaires puis décimale de notre rapport cyclique, nous allons remplir un tableau comme ci-dessous :

Echantillon	1	2	3	4	5	Etc...
Angle (°)	0
Rapport cyclique	0

11. **Renseigner**, après avoir recopié le tableau sur votre compte-rendu, les valeurs angulaires puis le rapport cyclique correspondantes aux différents échantillons.
12. **Créer** une boucle « for » permettant d'exploiter l'ensemble des échantillons. On utilisera la fonction sin() pour convertir un angle en son sinus (la fonction retourne un double).
13. A l'aide du cours vu en classe et de vos connaissances, **proposer** un filtre du premier ordre permettant de n'observer que les basses fréquences.
14. **Déterminer** la fréquence de coupure afin que l'on observe uniquement le signal sinusoïdal à 50 Hz.
15. **Proposer** un couple RC (d'après le montage issu du cours) permettant d'atteindre la fréquence de coupure désirée. (NB : s'il n'y a plus de condensateur disponible, choisissez un autre couple RC).
16. **Câbler** le montage et proposer un point de mesure à l'oscilloscope permettant d'observer le signal filtré à l'oscilloscope.
Appeler le professeur pour faire valider le montage et le point de mesure.
17. **Téléverser** et **observer** le résultat du signal filtré obtenu ainsi que le signal brut et les reporter sur le papier 1/2 millimétré à votre disposition. **Comparer** les résultats obtenus entre le signal brut et le signal filtré. **Conclure** sur l'intérêt d'un filtre de ce type.