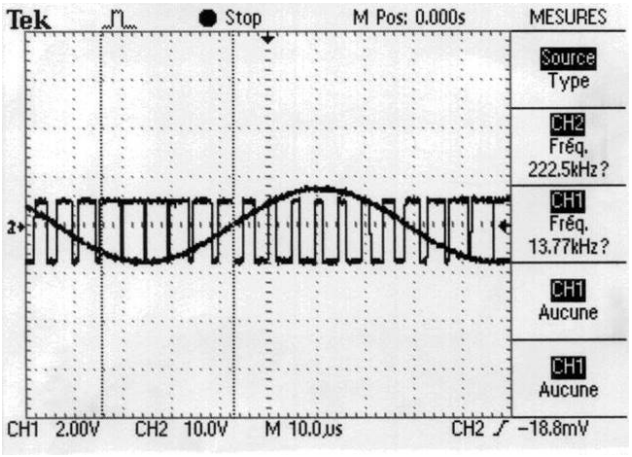
STM32 TIMER

Objectifs : Mises en œuvre des PWM (Pulse Width Modulation)

Matériel : Ce TP utilise une NUCLEO-F411RE, mais n'importe quelle autre carte NUCLEO convient.
Logiciel : MBED

Le signal PWM (MLI, Modulation de largeur d'impulsion) est un signal de fréquence constante et de rapport cyclique variable. Il est mis en œuvre dans des fonctions telles que :

La synthèse vocale ou associée à un filtre passe bas permet la synthèse de signaux audios.
La commande en vitesse d'un moteur à courant continu, ou ce dernier fait naturellement office de filtre passe bas.
La commande en position d'un servomoteur
La génération de signaux aléatoires ou périodiques, pour un onduleur par exemple.



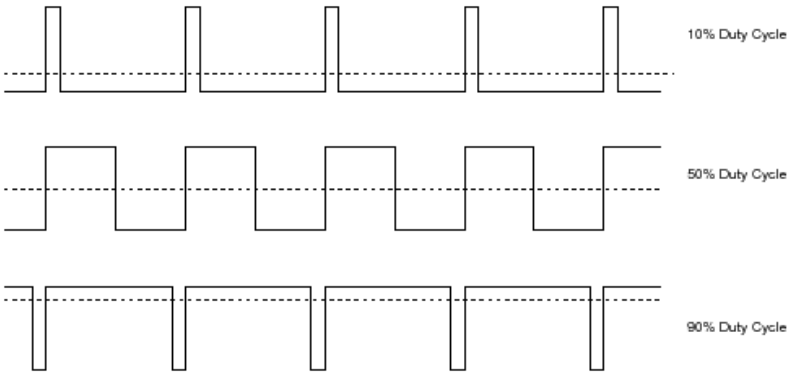
Courbe PWM dont la valeur moyenne est une courbe sinusoïdale.
La valeur moyenne est récupérée simplement par un filtre passe bas.

Valeur moyenne d'un signal rectangulaire :

La valeur moyenne d'un signal rectangulaire dépend du rapport cyclique (duty Cycle) ainsi que de la tension maximum (Vmax)

Le rapport cyclique η=th/T

th représente la durée de l'état haut et T la période.
Si th=0, η=0
Si th=T, η=1
donc 0≤η≤1



Valeur moyenne :

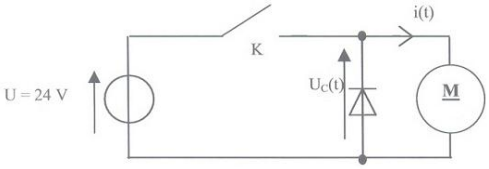
Vm=Vmax.th/T = Vmax. η

En retirant la fréquence porteuse F=1/T avec un filtre passe bas, il reste la valeur moyenne du signal.
Le filtre passe bas peut être électronique ou mécanique (cas d'un moteur et des son inertie l'inertie) ou optique (cas de l'œil humain qui filtrera les fréquence au delà de 30Hz).

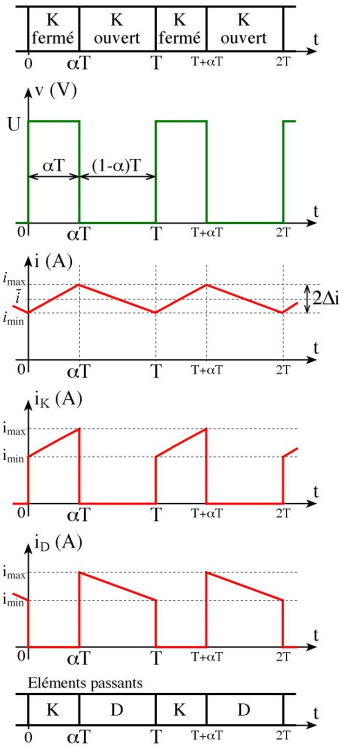
La génération d'un signal par PWM est particulièrement avantageux du point de vue de la consommation de la commande. En effet il sera produit pas un transistor MOSFET qui ne consomme pratique rien en mode triode (le transistor se comporte comme une résistance de très faible valeur) et rien du tout en mode bloqué.

Exemple de structure d'un hacheur permettant la commande d'un moteur à courant continu par PWM.

K est un interrupteur électronique, généralement un MOSFET.



K fermé le courant passe de l'alimentation vers le moteur.
K ouvert l'énergie emmagasinée dans l'inductance du moteur s'évacue sous forme d'un courant traversant la diode (dite de roue libre)



U représente le signal e commande PWM.

i représente le courant dans le moteur
il représente le courant dans le transistor
id représente le courant dans la diode

MBED propose une bibliothèque PWM très simple d'emploi, qui permet de définir la période et le rapport cyclique d'un signal PWM. *Attention tous les GPIO ne supportent pas le mode PWM.*

https://os.mbed.com/docs/v5.7/reference/pwmout.html

Pour tester le mode PWM créer un nouveau projet de avec comme modèle "Output a PWM signal"

Tester ce programme qui contrôle l'intensité lumineuse sur la LED verte

```
#include "mbed.h"
PwmOut mypwm(PWM_OUT);
DigitalOut myled(LED1);

int main() {
    mypwm.period_ms(10);
    mypwm.pulsewidth_ms(1);
    printf("pwm set to %.2f %%\n", mypwm.read() * 100);
    while(1) {
        myled = !myled;
        wait(1);
    }
}
```

STM32 TIMER

Exercice 1:

Réaliser un programme en C (C++), faisant varier l'intensité de la LED verte de +10% toutes les 100mS. La période PWM sera de 1mS.
La LED est allumée durant l'état haut de la sortie. La puissance lumineuse restituée est proportionnelle au rapport cyclique η=th/T (th= temps état haut, T est la période ici 1KHz). La persistance rétinienne fait office de filtre passe bas. (Il est admis que l'œil humain ne perçoit pas les variations de lumière inférieure à 30mS)

Exercice 2:

Un signal PWM peut permettre de faire clignoter automatiquement une LED , avec les mêmes structures que pour l'exercice , la fréquence est maintenant de 1Hz et le rapport cyclique est de ½ (th=T/2). Une fois le périphérique configuré et activé, le clignotement est automatique et n'est plus géré par le logiciel, le programme comportera une boucle infinie vide.

Exercice 3 :
La fonction PWM permet la synthèse de signaux, la valeur moyenne du signal PWM état directement proportionnelle au rapport cyclique. La valeur moyenne peut être récupérée par un simple filtre passe-bas. Ainsi il est possible de générer un signal analogique dont la fréquence est très inférieure à celle de la PWM, en filtrant la fréquence PWM on récupère les fréquences de modulation.
La sortie du signal PWM s'effectuera sur un connecteur Arduino et sera reliée à un filtre RC passe bas (prendre C=100nF) du premier ordre et de fréquence de coupure 100Hz . (On rappelle que ωc=1/RC), la fréquence PWM reste inchangée à 1KHz.

A l'aide d'un tableur on génère 20 valeurs comprises entre 0 et 1024 suivant une fonction sinusoïdale. Dans le programme, ces valeurs entières seront introduites dans un tableau qui servira à produire les PWM successives suivant la courbe ci-dessous.

Réaliser un programme générant une onde sinusoïdale de fréquence 50Hz. Vous vérifierez le résultat à l'aide d'un oscilloscope. Interpréter le résultat, comment l'améliorer.

angle radian	sin	PWM réelle	PWM entière
0	0	512	512
0,31	0,31	670,22	670
0,63	0,59	812,95	813
0,94	0,81	926,22	926
1,26	0,95	998,94	999
1,57	1	1024	1024
1,88	0,95	998,94	999
2,2	0,81	926,22	926
2,51	0,59	812,95	813
2,83	0,31	670,22	670
3,14	0	512	512
3,46	-0,31	353,78	354
3,77	-0,59	211,05	211
4,08	-0,81	97,78	98
4,4	-0,95	25,06	25
4,71	-1	0	0
5,03	-0,95	25,06	25
5,34	-0,81	97,78	98
5,65	-0,59	211,05	211
5,97	-0,31	353,78	354