

# Commande PWM pour LED-RGB

## Objectifs :

Nous souhaitons réaliser une commande d'une LED-RGB permettant de contrôler l'intensité et la couleur via des signaux binaires. Nous allons utiliser le principe du PWM pour contrôler l'intensité.

Le principe de la modulation de largeur d'impulsion PWM (Pulse Width Modulation) est de contrôler la valeur moyenne d'un signal en modulant la largeur d'activation d'un signal binaire ('1' / '0') pendant une période définie. Dans la partie test, nous pourrions expérimenter le principe du mélange additif des couleurs « RGB »

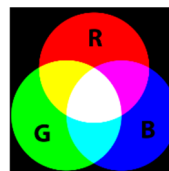
### Principe physique :

L'affichage couleur d'un point lumineux est basé sur le principe du mélange additif des couleurs primaires ( Rouge « R », Vert « G », Bleu « B » ).

Soit X : une lumière de couleur quelconque, alors :

$$X = I_r . R + I_g . G + I_b . B$$

où (  $I_r$ ,  $I_g$ ,  $I_b$  ) sont les intensités lumineuses de chacune des lumières de couleurs primaires (RGB)



### Application pratique du principe physique :

- Un écran couleur, est constitué des milliers de points lumineux. L'intensité et la couleur de chaque point sont contrôlées par le niveau des signaux vidéo (RGB) à afficher
- L'affichage LCD d'un téléphone mobile de la nouvelle génération est rétro-illuminé par 3 LED (RGB) afin de pouvoir régler la couleur de cette lumière (Back-Light)
- Des éléments décoratifs sont actuellement en vente (LED Tower, RGB Led projector)



### Solutions pour contrôler la luminosité :

Pour contrôler ces 3 LED (RGB), afin de pouvoir définir la couleur voulue, deux méthodes sont possibles :

- **Analogique** à l'aide de 3 signaux analogiques (r, g, b), le cas des écrans à Tube à Rayon Cathodique (CRT)
- **Numérique** : Solution robuste, simple, et à rendement énergétique très élevé !

### Principe de fonctionnement d'un PWM :

Un PWM permet de contrôler via un signal binaire une information analogique.

La modulation de largeur d'impulsions (PWM) est une technique couramment utilisée pour synthétiser des signaux continus à l'aide d'un signal tout ou rien numérique ('1' et '0'). La variation de la tension continue est obtenue en modifiant le rapport cyclique entre la durée à '1' et à '0' du signal digital.

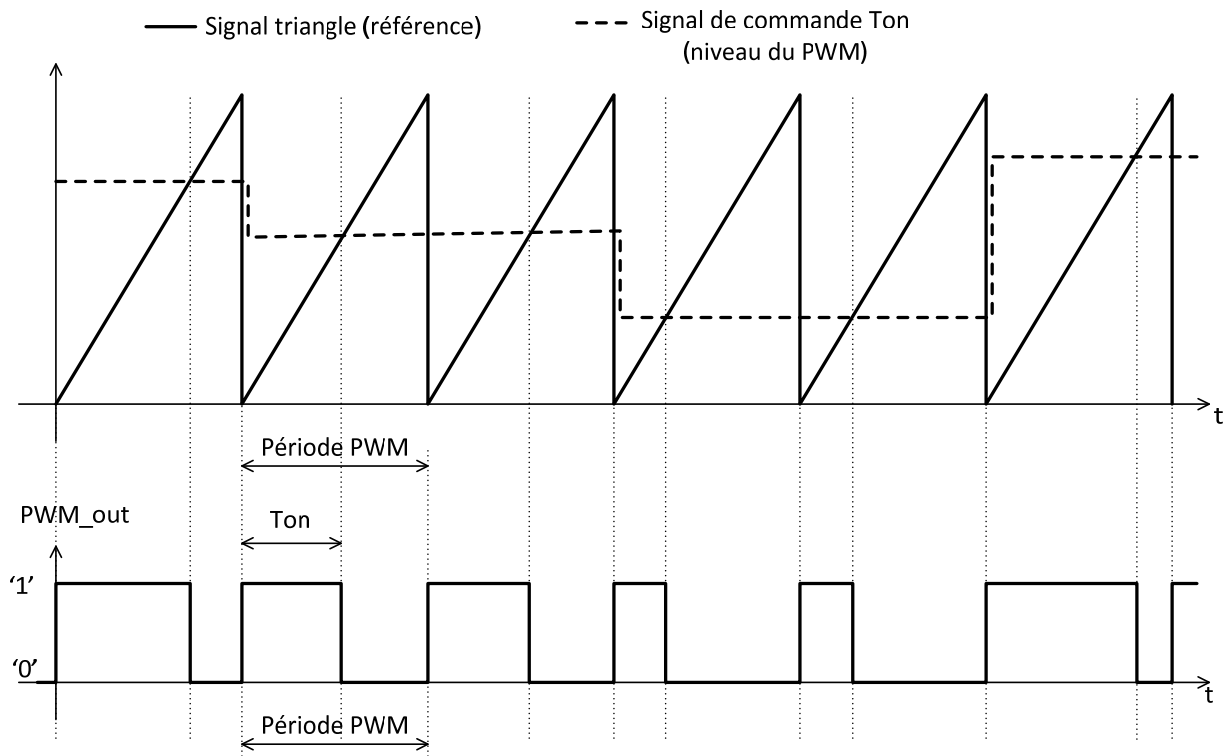


Figure: Principe de fonctionnement du PWM

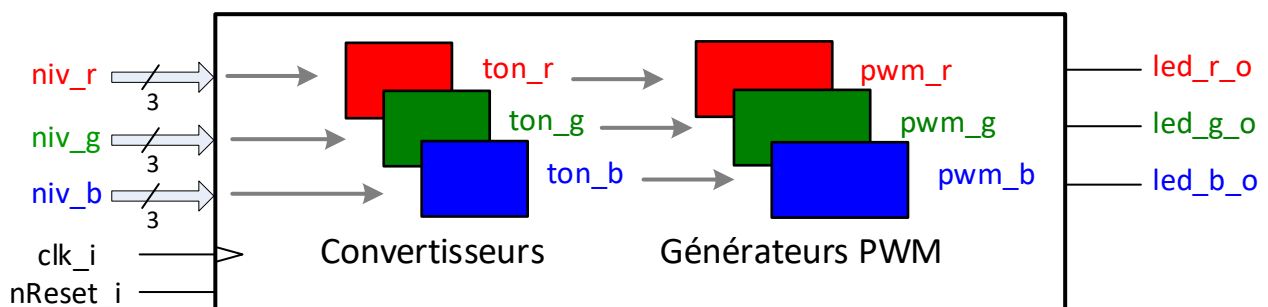
Nous pouvons définir 2 paramètres qui spécifient un générateur PWM, soit:

- La fréquence du PWM (période PWM).
- La résolution du PWM, soit le nombre de bits du signal de commande du niveau (Ton).

### Spécification du système de commande d'une Led RGB :

Le système comprend 2 blocs, soit les convertisseurs et les PWM. L'intensité des leds n'étant pas linéaire en fonction de ton, nous allons utiliser des convertisseurs pour obtenir 8 niveaux d'intensité régulièrement répartis. Le second bloc comprend les 3 générateurs PWM, soit un pour chaque couleur.

Voici le schéma bloc du système :



Entrées/sorties du système :

signal	direction	taille	Description
clock_i	entrée	1	Horloge du système synchrone
nReset_i	entrée	1	Signal de remise à zéro asynchrone
m_blanc	entrée	1	Sélection du mode blanc où les 3 leds sont pilotées par la même intensité, soit <b>niv_g_i</b>
<b>niv_r_i</b> <b>niv_g_i</b> <b>niv_b_i</b>	entrée	3	Niveau d'intensité souhaité pour les 3 leds R- G- B
<b>led_r_o</b>	sortie	1	Signal de commande PWM de la <b>led red</b>
<b>led_g_o</b>	sortie	1	Signal de commande PWM de la <b>led green</b>
<b>led_b_o</b>	sortie	1	Signal de commande PWM de la <b>led blue</b>
cycle_pwm_o	sortie	1	Signal ayant une fréquence identique au PWM. La forme de celui-ci est libre. Utilisé pour mesurer $F_{PWM}$ à l'aide d'un oscilloscope

Description du fonctionnement :

- L'intensité lumineuse ( **niv\_r**, **niv\_g**, **niv\_b**) de chaque couleur est définie par une valeur sur 3 bits de 0 à 7.
- Les convertisseurs permettent de déterminer la valeur de ton pour les 8 niveaux d'intensité. Une table sera utilisée pour convertir les niveaux d'intensité (codée sur 3 bits) en une largeur d'impulsion codée sur 8 bits (table donnée ci-après).
- Les trois sorties **led\_r**, **led\_g** et **led\_b** sont commandées par un signal modulé en largeur d'impulsion (PWM) codé sur 8 bits. La largeur de l'impulsion peut varier de 0 à 255.

## Commentaire:

Notre impression de luminosité n'est pas linéaire par rapport à la tension moyenne appliquée à une Led. Nous allons donc introduire, pour la réalisation, un convertisseur qui va permettre de fournir une suite de largeur d'impulsion, Ton pour le PWM, non linéaire. Cela permettra d'attribuer 8 valeurs de largeur d'impulsion permettant une progression plus ou moins linéaire de l'impression de luminosité de la LED.

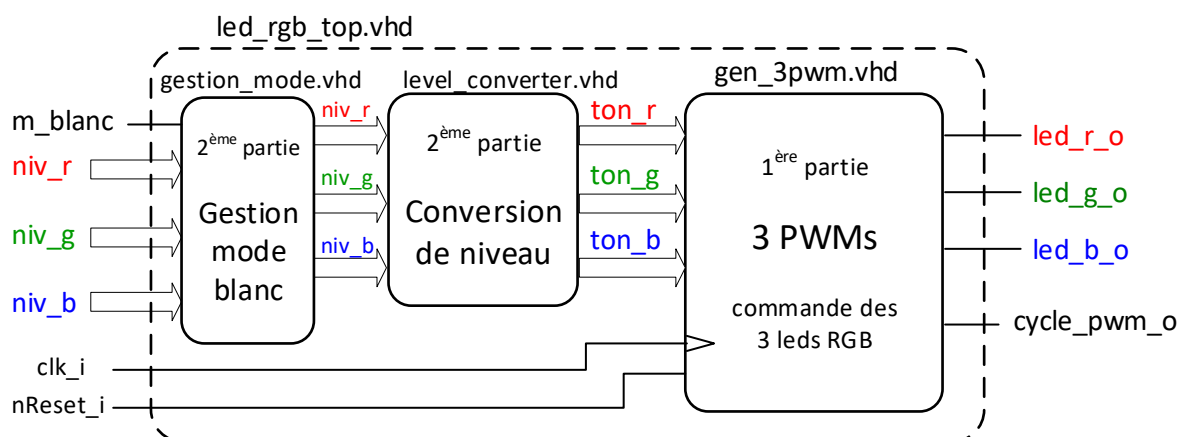
Décomposition du système :

Pour la réalisation pratique, nous allons décomposer le système en deux modules, soit:

- Un module de conversion des niveaux d'intensité en une valeur de Ton appliquée à la Led via le PWM.
- Un module composé de 3 PWMs.

Pour la réalisation, nous allons travailler en deux parties, soit:

- 1<sup>ère</sup> partie: conception et réalisation du module comprenant trois PWMs, un par Led (**gen\_3pwm.vhd**).
- 2<sup>ème</sup> partie: Conception et réalisation du module de conversion des niveaux d'intensité en une valeur de Ton et gestion du mode « blanc ».



## 1<sup>ère</sup> partie : Réalisation des 3 générateurs PWM

Voici les spécifications des générateurs PWM à réaliser, soit :

- PWM à 256 niveaux, le signal de consigne, temps d'activation Ton, est sur 8 bits.
- Période du PWM est de 1.024 ms soit une fréquence proche de 1 KHz (976 Hz). Vous disposez d'une fréquence système à 1 MHz.

Vous devez **optimiser** la quantité de logique nécessaire à votre solution. Celle-ci doit être optimisée dans notre cas, soit la réalisation de 3 PWMs fonctionnant à la **même** fréquence mais avec des niveaux différents.

## 2<sup>ème</sup> partie : Réalisation du module de conversion

Pour cette partie, il y a 2 blocs à réaliser, soit :

- Conception et réalisation du module de conversion des niveaux d'intensité en une valeur de Ton. La table de conversion vous est donnée ci-dessous.
- Réalisation d'un bloc pour la gestion du mode blanc, soit :

m\_blanc actif ('1') Les trois leds sont pilotées par le niveau d'intensité vert (niv\_g). Ainsi la led RGB a une couleur blanche

m\_blanc inactif ('0') Les trois leds sont pilotées par leur niveau d'intensité respectif. Ainsi la led RGB a toute la palette de couleur possible.

Table de conversion entre le niveau d'intensité et le Ton du PWM pour la led

Niveau intensité, 3bits	Ton (level PWM), 8 bits	
0	0	0x00
1	5	0x05
2	21	0x15
3	37	0x25
4	53	0x35
5	69	0x45
6	85	0x55
7	101	0x65

## Documents à rendre

Ce laboratoire se déroule par groupe de 2 étudiants. Vous devrez rendre à l'issue de ce travail de laboratoire:

- **UN SEUL fichier PDF** avec votre rapport de laboratoire comprenant l'ensemble de vos explications, étapes de conception, schémas, les preuves des vérifications réalisées, réponses aux différentes étapes et question demandées, ainsi que les descriptions VHDL synthétisable
- un fichier zip ou tar avec: scripts tcl, répertoires /src/... et /src\_tb/...

Ces documents sont à déposer sur Cyberlearn de la HES-SO (<https://cyberlearn.hes-so.ch/>), sur la page de votre unité.

## Marche à suivre

1<sup>ère</sup> partie :

1. Analyser le fonctionnement du sous-système comprenant les 3 générateurs PWM. Etablir une décomposition en optimisant la quantité de logique.
2. Etablir **un schéma bloc** de votre solution. Vous devez spécifier chaque bloc.
3. Etablir, pour chacun de vos blocs séquentiels, une table des fonctions synchrones.
4. Décrire chacun de vos composants à l'aide d'une description VHDL.
5. Regrouper l'ensemble de vos blocs par une description structurelle VHDL dans le fichier nommé *gen\_3pwm.vhd* (entité fournie).
6. Vérifier le fonctionnement du sous-système *gen\_3pwm.vhd* à l'aide d'une simulation manuelle interactive avec *console\_sim.vhd* et la console REDS.  
Utiliser le script : `run_ctrl_gen_3pwm_sim.tcl`
7. Réaliser la synthèse du fichier *gen\_3pwm.vhd* avec vos blocs afin de vérifier que l'ensemble est synthétisable. Noter la quantité de logique utilisée et **vérifier** que le **nombre de flip-flops** obtenu est correct (correspond à votre conception).

2<sup>ème</sup> partie :

8. Analyser le fonctionnement du module de conversion des niveaux d'intensité. Etablir une décomposition si cela est opportun. Réaliser une solution avec des tailles fixes, soit 3 bits pour le niveau d'intensité et 8 bits pour Ton du PWM.
9. Décrire votre composant à l'aide d'une description VHDL dans un fichier nommé *level\_converter.vhd*.
10. Réaliser la synthèse du fichier *level\_converter.vhd* afin de vérifier que ce composant est synthétisable. Analyser la quantité de logique obtenue.
11. Réaliser et décrire le composant *gestion\_mode.vhd* afin de contrôler les niveaux des 3 leds R-G-B selon la spécification.
12. Réaliser la synthèse du fichier *gestion\_mode.vhd* afin de vérifier que ce composant est synthétisable.

Vérification du système complet et intégration :

13. Vérifier le fonctionnement du système *led\_rgb\_top.vhd* avec une simulation automatique. Vous demanderez le fichier *test-bench* à votre professeur.  
Utiliser le script : `run_led_rgb_top_tb.tcl`
14. Faire la synthèse et le placement routage de votre solution pour le *led\_rgb\_top.vhd* à l'aide du logiciel Quartus. Vous utiliserez le top spécifique pour la carte Max-V fourni dans le projet, soit : *maxv\_top.vhd*.  
Vous analyserez la quantité de logique de composant *led\_rgb\_top* de votre solution (sans la logique du *maxv\_top*)  
La carte Max-V comprend un CPLD de type : 5M570ZF256C5  
Reprendre les étapes pratiquées lors du labo Intro\_Bin\_Lin.
15. Intégrer votre solution en programmant une maquette MaxV 80p-25p avec le fichier \*.pof généré lors du placement et routage
16. Tester votre solution du système de commande d'une Led RGB à l'aide d'une carte Max-V et d'une boîte de commande (EINEV 287). Faire des essais de différentes couleurs.  
Vérifier la fréquence de fonctionnement du PWM à l'aide de l'oscilloscope sur la pin 2 du connecteur 80 pôles
17. Vous devez faire valider votre montage final par le professeur ou l'assistant.