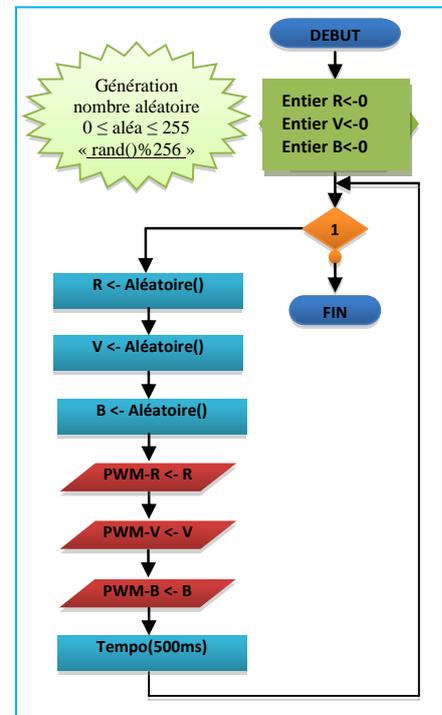
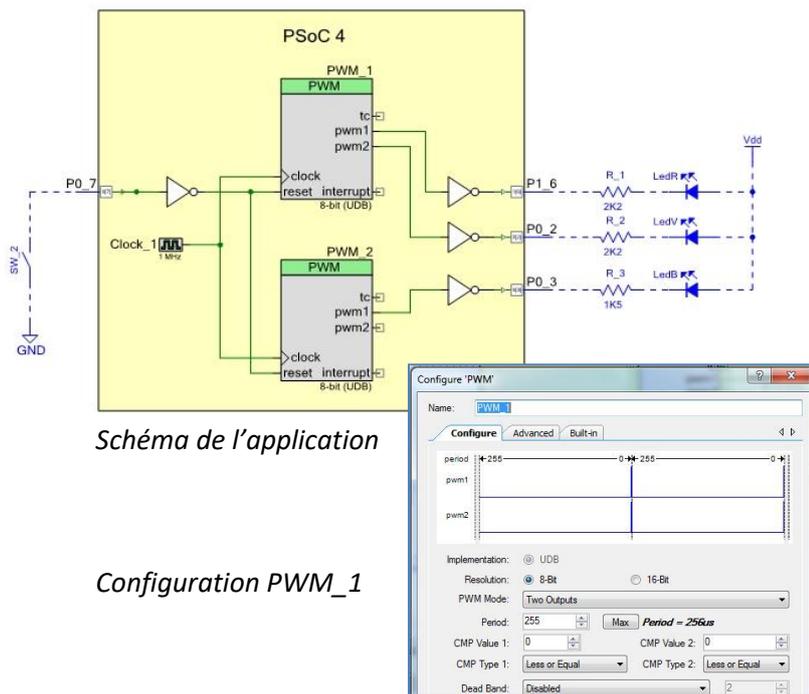
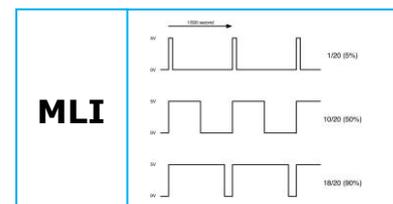


## 1 VARIATION DE COULEUR D'UNE LED RGB EN PWM

Ce projet permet de faire varier aléatoirement la couleur de la led RGB de la carte de développement, en modulant par PWM les 3 composantes Rouge, Vert et Bleu. La remise à zéro des compteurs se fait par appui sur le bouton poussoir de test SW2.



Organigramme de l'application



### Rappel :

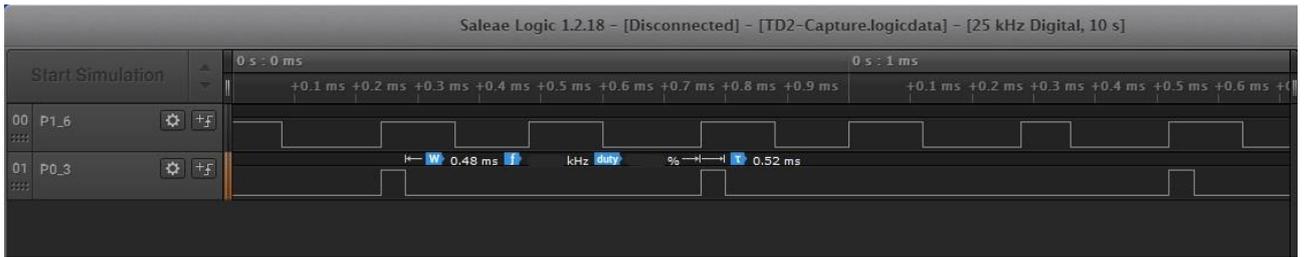
La modulation de largeur d'impulsions (**MLI**) permet de faire varier la puissance d'un dispositif électrique. Ici on utilisera la fonction PWM pour faire varier l'intensité lumineuse de chaque led de couleur.

Fonction	Rôle
<code>uint8 PWM_Start(void);</code>	Initialisation du timer avec les valeurs saisies dans la boîte de configuration
<code>void PWM_WriteCompare1(uint8 value);</code>	Définition de la valeur du rapport cyclique de la sortie PWM1
<code>void PWM_WriteCompare2(uint8 value);</code>	Définition de la valeur du rapport cyclique de la sortie PWM2

### Travail demandé :

- 1) Créer un nouveau projet « Activite04 » - architecture du type PSoC 4.
- 2) Saisir le schéma de l'application.
- 3) Configurer les 2 Timers en mode 8 bits ainsi que l'entrée P0\_7 en mode « Pull-up ».
- 4) Définir les affectations des broches d'entrées / sorties.
- 5) Ecrire le programme en langage C correspondant à l'organigramme de l'application
- 6) Compiler le projet puis transférer le programme dans le PSoC.
- 7) Vérifier le bon fonctionnement

On a capturé à un instant donné, les signaux générés sur les leds rouge et bleue de la carte de développement :



- 8) Quelle est la fréquence et le rapport cyclique du signal généré sur la led bleue ?
- 9) Déterminer la valeur du registre de comparaison du module PWM\_2.

Notes :

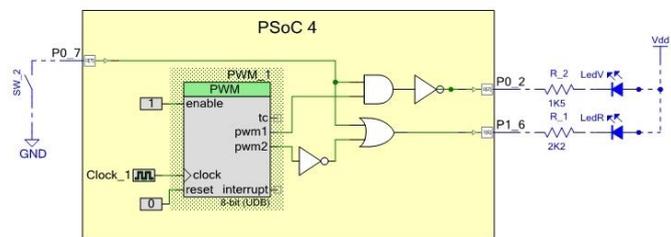
## 2 COMMANDE DE 2 LEDS EN PWM

Cette application permet d'indiquer à un opérateur le bon ou mauvais fonctionnement d'un système à l'aide de 2 leds (verte et rouge).

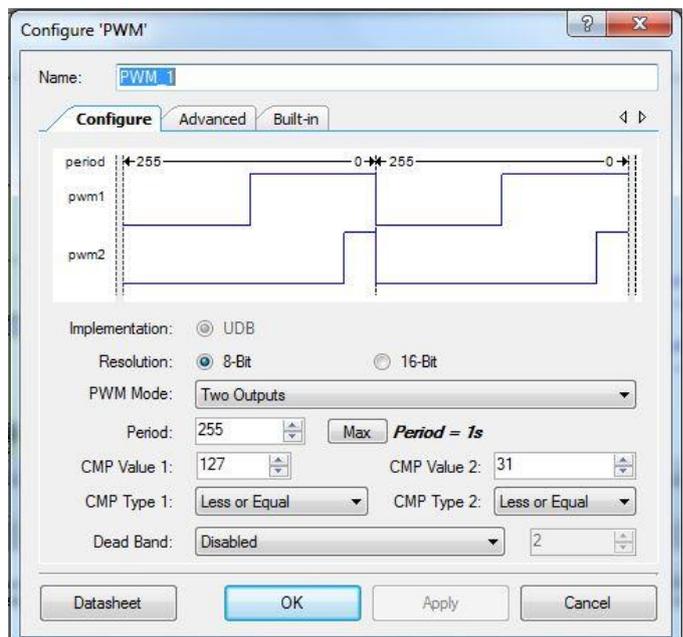
La led verte indique le bon fonctionnement en clignotant toutes les secondes avec une durée d'allumage de 0,5s.

La led rouge indique un défaut en clignotant toutes les secondes également, mais avec une durée d'allumage de 125ms.

On simulera un défaut en appuyant sur le bouton poussoir de test SW2.



*Schéma de l'application*



*Configuration du module PWM\_1*

### Travail demandé :

- 1) Créer un nouveau projet « Activite05 » - architecture du type PSoC 4.
- 2) Saisir le schéma de l'application.
- 3) Déterminer la valeur du signal d'horloge appliqué au module PWM, les valeurs de comparaison de chaque sortie afin de répondre au cahier des charges.
- 4) Configurer le Timer « PWM\_1 » en mode 8 bits ainsi que l'entrée P0\_7 en mode « Pull-up ».
- 5) Définir les affectations des broches d'entrées / sorties.
- 6) Appeler la fonction PWM\_1\_Start() dans la fonction main() du programme pour initialiser les timers et générer les signaux MLI.
- 7) Compiler le projet puis transférer le programme dans le PSoC.
- 8) Vérifier le bon fonctionnement

### Notes :

---

## 3 COMMANDE DE MOTEUR PAS A PAS

---

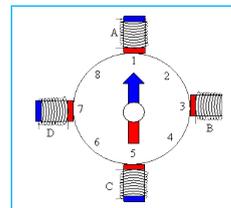
### Principe de fonctionnement :

Ci-contre la représentation simplifiée d'un moteur à 4 pas (4 enroulements).

Le principe pour faire tourner le rotor dans le sens horaire est d'alimenter successivement les bobines dans l'ordre A, B, C, D.

Pour le sens antihoraire, c'est l'ordre inverse.

En réalité, il existe beaucoup plus d'enroulements à l'intérieur d'un moteur pas à pas.



Position	A	B	C	D
1	1	0	0	0
3	0	1	0	0
5	0	0	1	0
7	0	0	0	1

Position	A	B	C	D
2	1	1	0	0
4	0	1	1	0
6	0	0	1	1
8	1	0	0	1

Le système que l'on se propose de réaliser dans cette activité est le positionnement d'une mini antenne parabolique sur le plan azimutal.

Le système doit être capable de positionner la parabole juste en lui indiquant la valeur de l'angle en degré. Le nord étant la référence (0°).

Le moteur 28BYJ48 utilisé dans cette application est un petit moteur unipolaire à 4 phases, alimenté en 5V et comportant 64 pas couplé à un motoréducteur de rapport 1/64 (voir documentation).

La commande du moteur pas à pas consiste donc à lui appliquer une succession de signaux dans un ordre bien déterminé

(voir le tableau des commandes ci-contre).

Le temps minimal d'une commande entre 2 pas est de 1ms (temps minimum pour la stabilisation).

**Half-Step Switching Sequence**

Lead Wire Color	--> CW Direction (1-2 Phase)							
	1	2	3	4	5	6	7	8
4 Orange	-	-						
3 Yellow			-	-				
2 Pink					-	-		
1 Blue							-	-

Commande en 1/2 pas

L'interface de puissance pour commander le moteur est basé sur le composant ULN2003 (voir documentation).

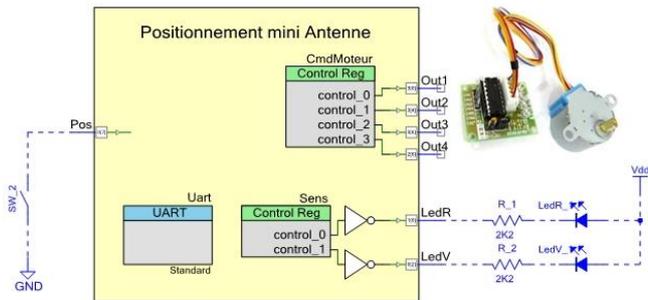


Schéma de l'application

Name	Port	Pin
\Console:rx\	P0 [4]	28
\Console:tx\	P0 [5]	29
LedR	P1 [6]	43
LedV	P0 [2]	26
Out1	P3 [0]	11
Out2	P3 [4]	15
Out3	P3 [6]	17
Out4	P2 [6]	8
Pos	P0 [7]	31

### Travail demandé :

- 1) Etant donné les caractéristiques du moteur et du rapport de moto réduction :
  - ↳ quelle est la valeur de l'angle correspondant à un pas du moteur ?
  - ↳ En combien de temps obtient-on une rotation complète de la parabole, si la durée de commande d'un pas est de 2 ms ?
- 2) Ajouter le projet « Activite06 » à l'espace de travail.
- 3) Configurer l'UART en 9600 bits/s, 8 bits de données, pas de parité, 1 bit de stop.
- 4) Etablir en langage C, le programme permettant de :
  - saisir l'angle de rotation dans un terminal
  - mettre le moteur en fonctionnement lorsque l'on appuie le bouton poussoir de test SW2
  - arrêter le moteur à la position souhaitée
  - allumer la led verte dans le sens horaire de rotation et la led rouge dans l'autre sens

Consigne : La durée de la commande d'un pas est fixée à 2 ms
- 5) Compiler le projet puis transférer le programme dans le PSoC.
- 6) Vérifier le bon fonctionnement