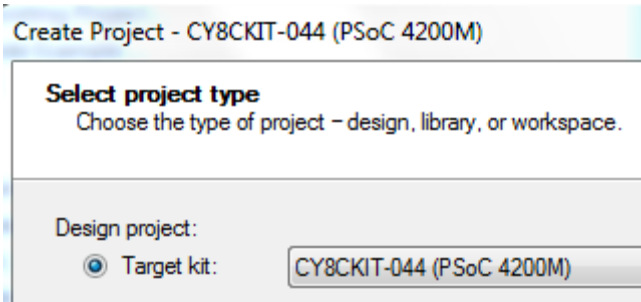
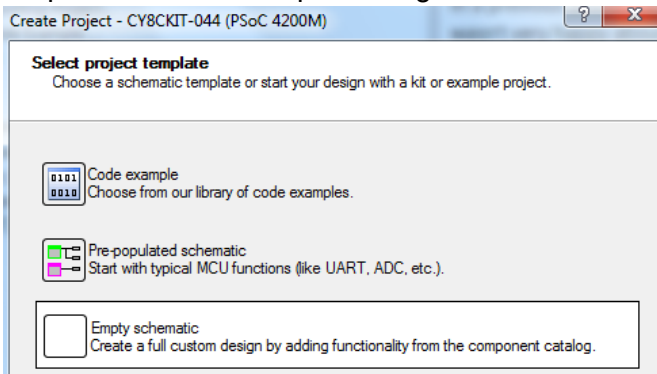




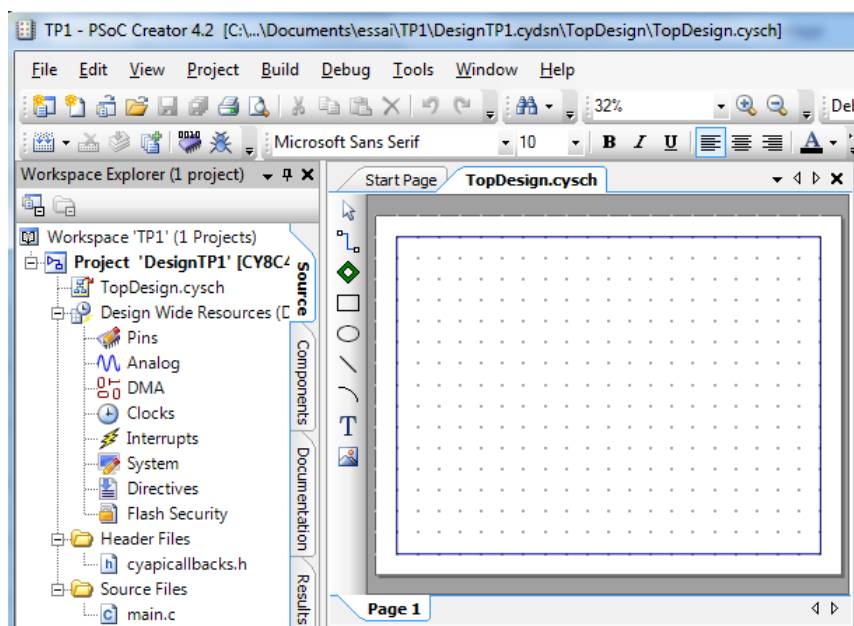
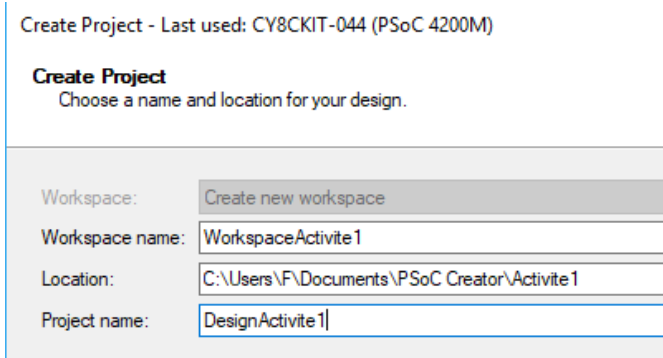
- Lancement de l'Environnement de Design Intégré :
- Création d'un nouveau projet : *File > New > Project...*
- La cible du projet est la carte du Kit CY8CKIT-044 (PSoC 4200M) :



- On partira d'une conception vierge :



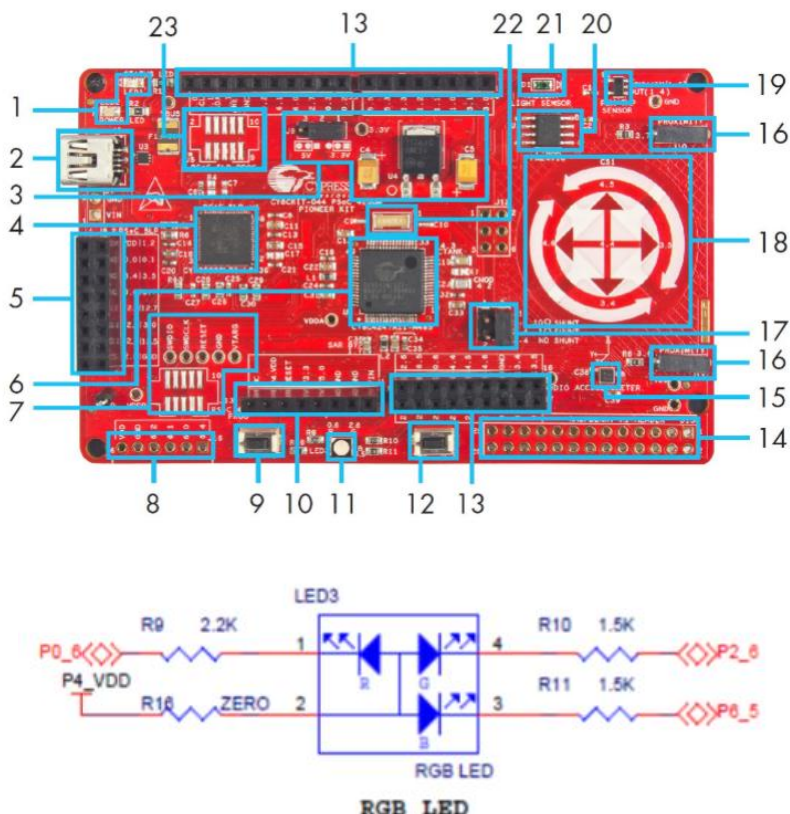
- Renseigner le nom du projet et l'emplacement des fichiers :



L'environnement est prêt...

Clignotement d'une LED

La carte comprend une LED multicolore RGB montée en surface (repère 11 ci-dessous) :

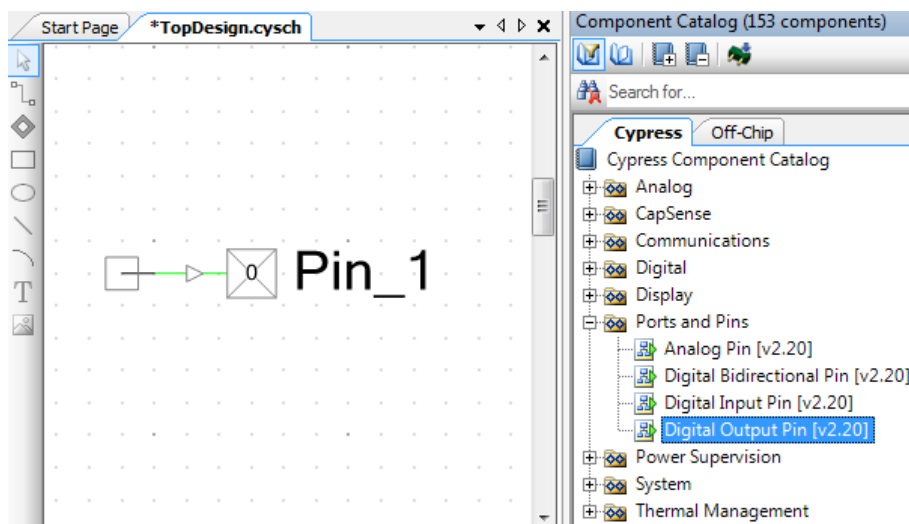


Les trois LED ont une anode commune reliée à l'alimentation VDD de la carte.

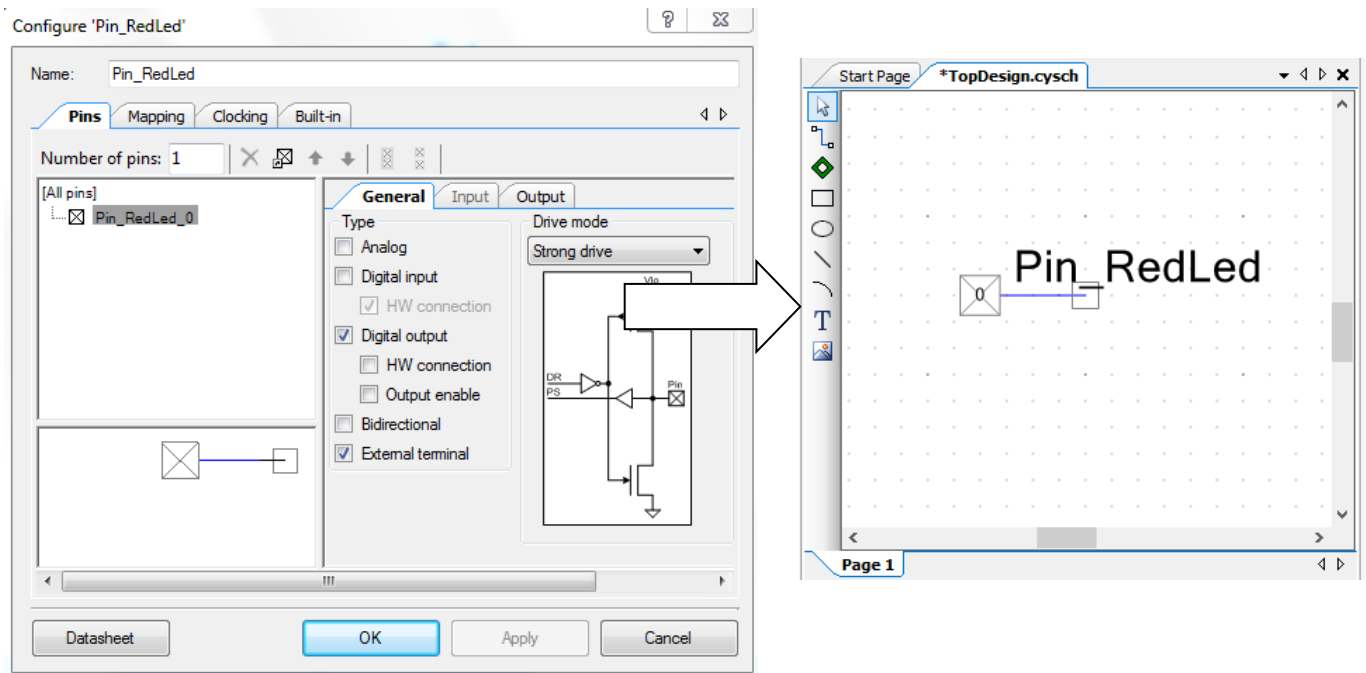
Pour allumer une LED, il faut donc relier sa cathode à la masse GND.

LED	Port cathode
Rouge (R)	P0_6
Vert (G)	P2_6
Bleu (B)	P6_5

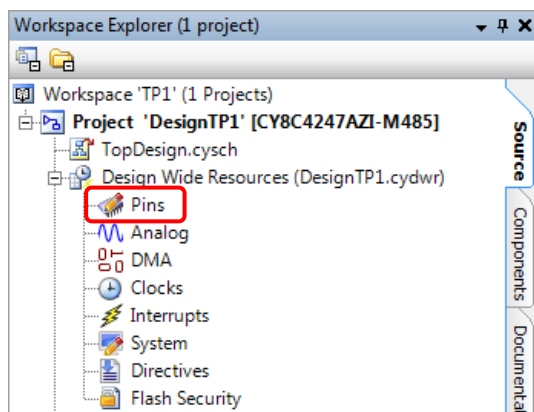
Depuis le catalogue des composants à droite, faire glisser le composant *Digital Output Pin* dans la fenêtre de conception *TopDesign* →



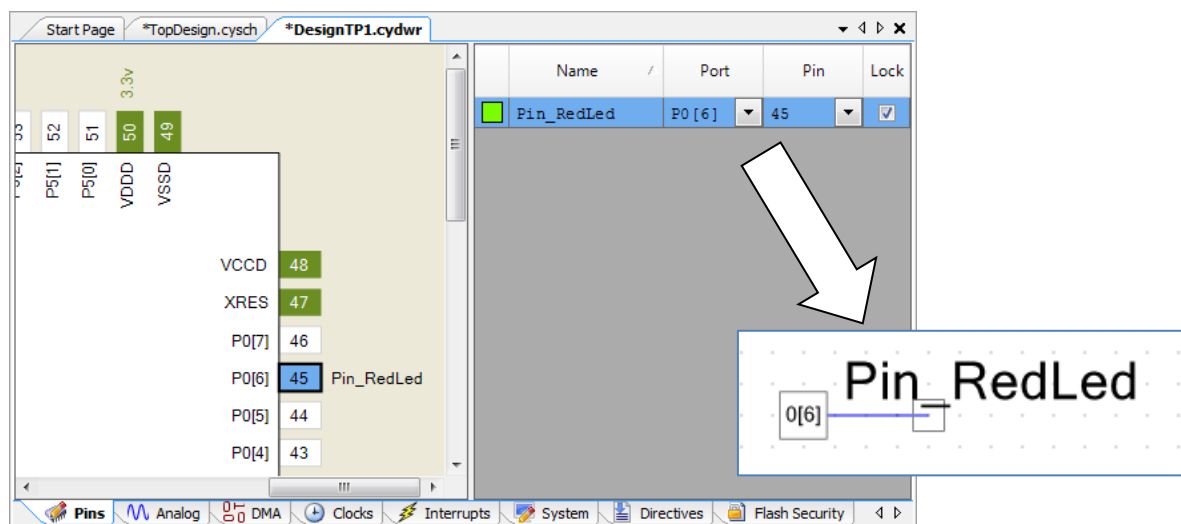
Double-cliquer sur le composant pour le configurer :



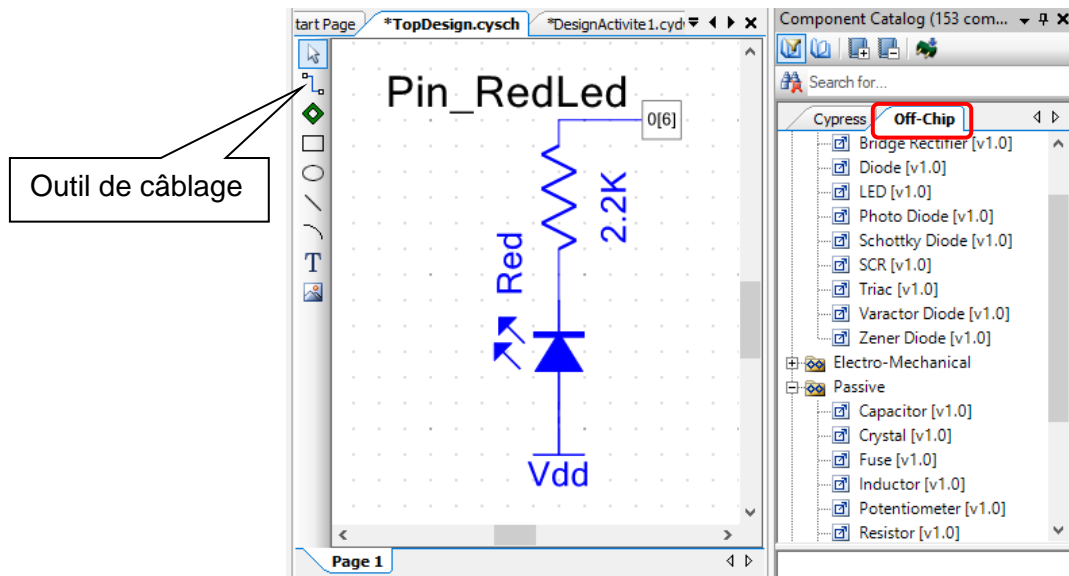
Dans l'explorateur, double-cliquer sur *Pins* :



Diriger le composant vers le port P0_6, c'est-à-dire la cathode de la LED rouge :



On peut agrémenter le schéma et faciliter sa lisibilité grâce au catalogue de composants, onglet *Off-Chip* :



Générer le projet, menu *Build* (icône ) :

```

Output
Show output from: All
Application checksum calculated and stored in ELF section .cychecksum
Checksum calculated and stored in ELF section .cymeta
cyelftool.exe -S C:\Users\lebris.LYC-LEMANSSUD.058\Documents\essai\TP1\DesignTP1.cydsn\Co
Flash used: 1944 of 131072 bytes (1,5 %).
SRAM used: 2576 of 16384 bytes (15,7 %). Stack: 2048 bytes. Heap: 128 bytes.
----- Build Succeeded: 06/04/2018 16:48:47 -----

```


Dans l'explorateur, dans les fichiers source, cliquer sur *main.c* puis compléter le programme de clignotement de la LED rouge (fréquence = 1 s) :

```

11 //
12 #include "project.h"
13
14 int main(void)
15 {
16     CyGlobalIntEnable; /* Enable global interrupts. */
17
18     /* Place your initialization/startup code here (e.g. MyInst_Start()) */
19
20     for(;;)                // Boucle infinie
21     {
22         /* Place your application code here. */
23         Pin_RedLed_Write(0); // Allumer la LED rouge
24         CyDelay(1000);       // pause 1s
25         Pin_RedLed_Write(1); // Eteindre la LED rouge
26         CyDelay(1000);       // pause 1s
27     }
28 }

```

Connecter la carte PsoC au port USB du PC. Compiler et transférer le programme dans la puce en

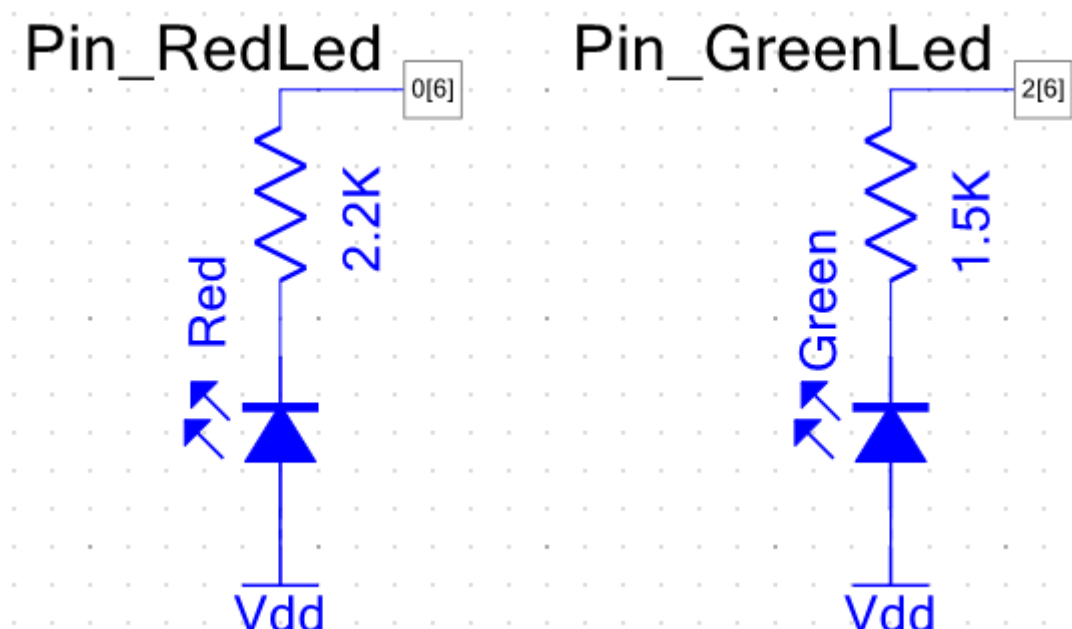
cliquant sur l'icône *Program*  :

```
Output
Show output from: All
Erasing...
Programming of Flash Starting...
Protecting...
Verify Checksum...
Finished Programming
Device 'PSoC 4200M CY8C4247AZI-M485' was successfully programmed at 06/04/2018 16:59:04.
```

Une fois programmée, la LED rouge devrait clignoter toutes les secondes...

Exercice

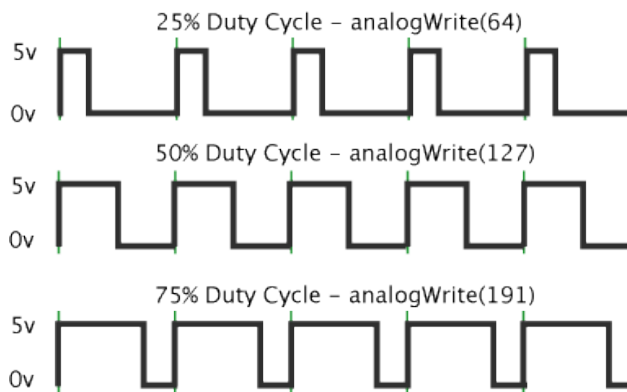
Compléter le design pour obtenir le schéma ci-dessous :



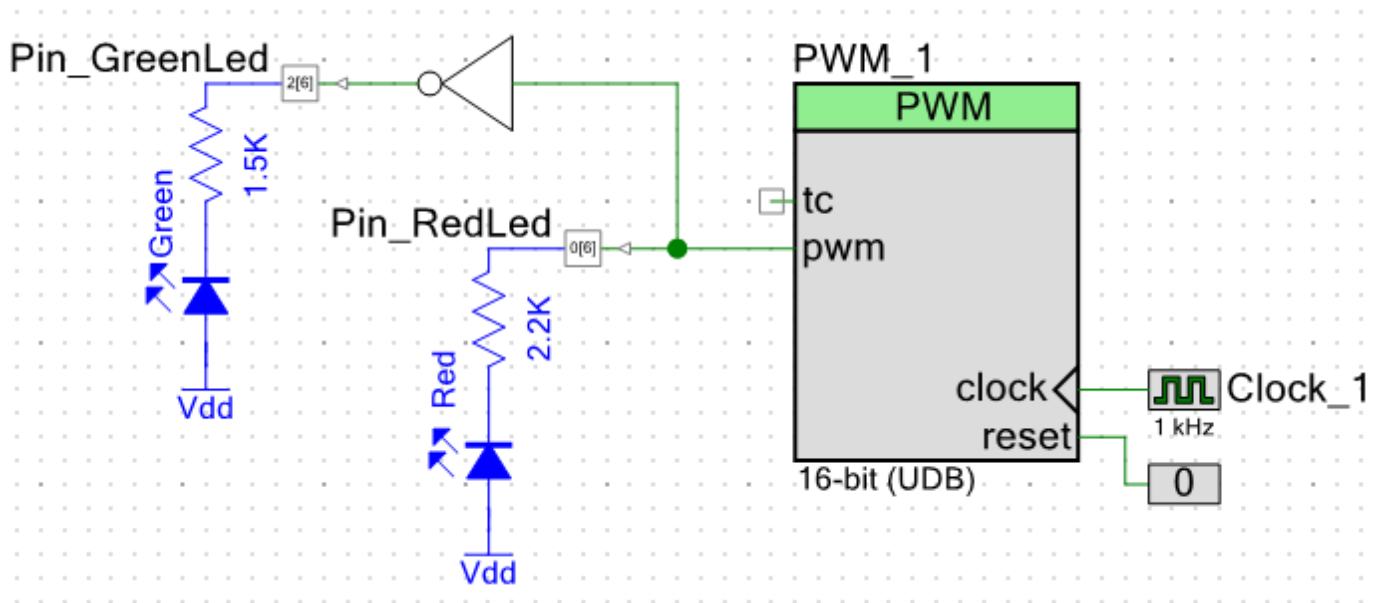
Modifier le programme pour faire clignoter la LED multicolore en rouge et en vert : la LED s'allume en rouge pendant 1 s, puis elle s'allume en vert pendant 1 s, puis à nouveau en rouge pendant 1 s, etc.

Pour aller plus loin...

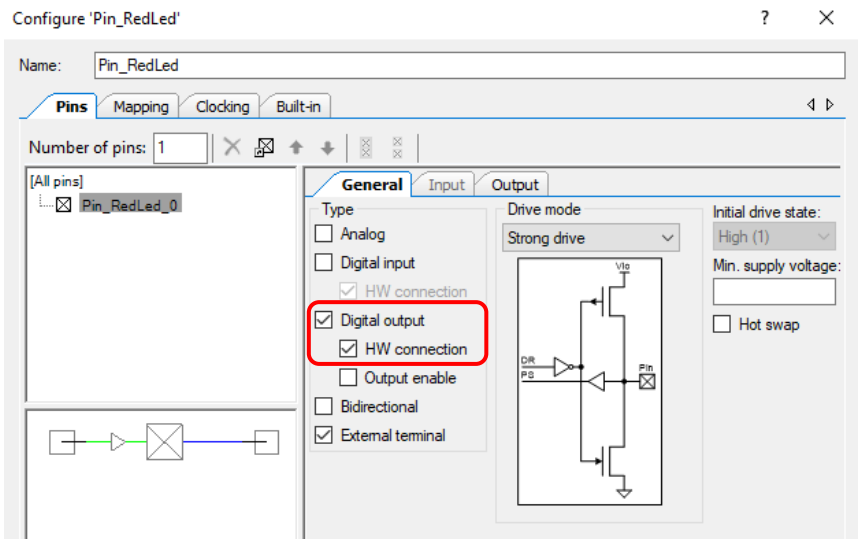
Le clignotement sera cette fois produit au niveau matériel grâce à un générateur de signaux PWM (*Pulse Width Modulation* ou *Modulation en Largeur d'Impulsion*).



La conception à réaliser dans l'EDI est la suivante :



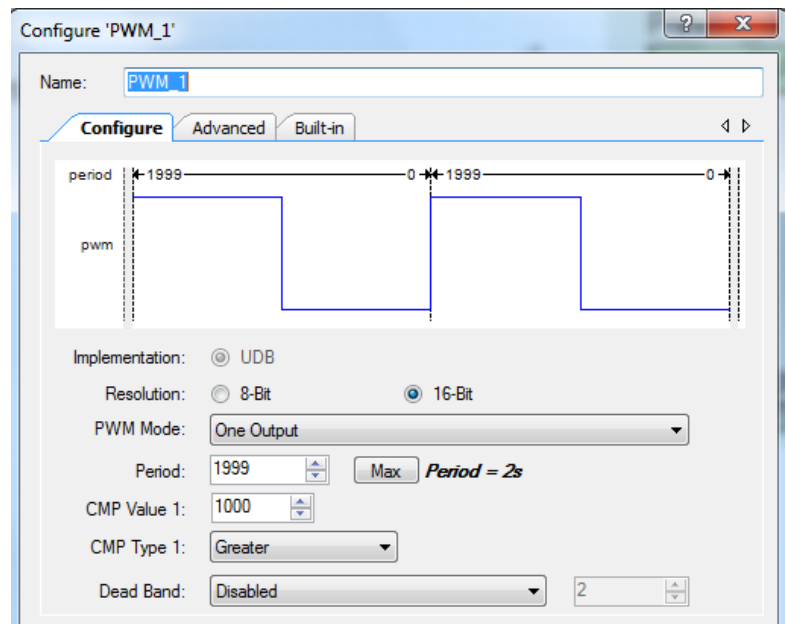
Pour relier les ports 0[6] et 2[6] au générateur PWM, il faut autoriser cette connexion en cochant **HW connection** dans la boîte de configuration.



Lorsque le signal en sortie du générateur PWM est à l'état haut, la LED rouge est éteinte alors que la LED verte est allumée grâce à la porte logique NON (Not). Et inversement lorsque le signal du générateur PWM est à l'état bas.

La configuration du générateur de signal PWM est la suivante →

Le compteur décroît de 1999 à 0, à la fréquence d'horloge (*Clock*) de 1 kHz, soit une période de 2 s. Lorsque le compteur atteint la moitié (c.à.d. 1000), le signal bascule de l'état haut à l'état bas. Une fois à zéro, le signal bascule à nouveau et le compteur repart à 1999.



Le programme principal *main.c* se contente de démarrer le générateur de signal PWM :

```

12  #include "project.h"
13
14  int main(void)
15  {
16      CyGlobalIntEnable; /* Enable global interrupts. */
17
18      /* Place your initialization/startup code here (e.g. MyInst_Start()) */
19      PWM_1_Start();      // Démarrage du générateur de signaux PWM
20
21      for(;;)             // Boucle infinie
22      {
23          /* Place your application code here. */
24      }
25  }
26
27  /* [] END OF FILE */

```

Exercice supplémentaire :

Modifier la configuration du générateur PWM de sorte que la LED rouge reste allumée deux fois plus longtemps que la verte...