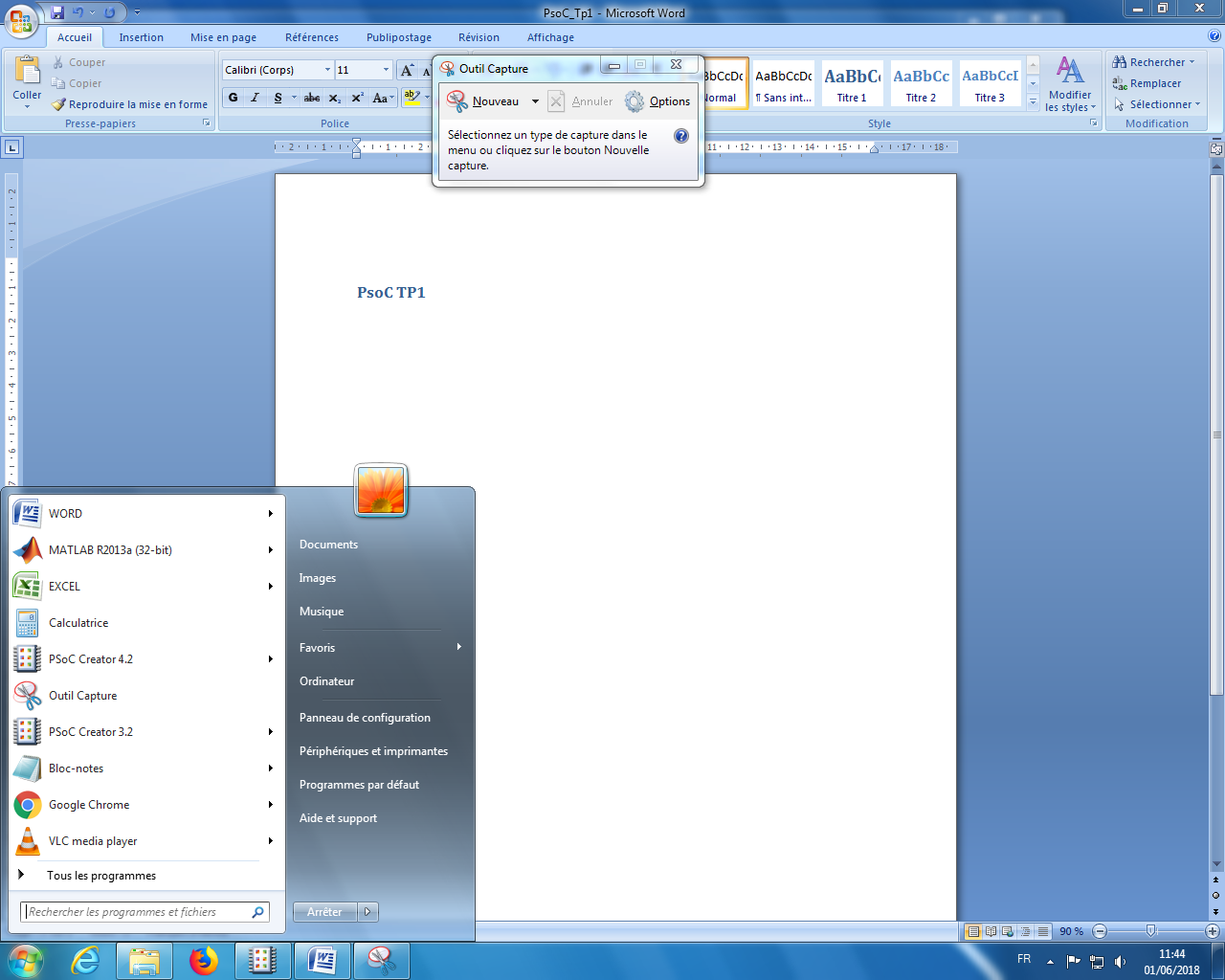
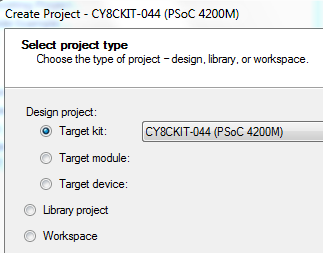
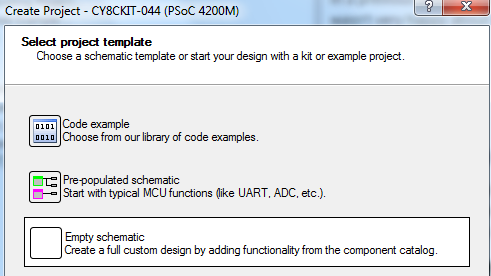
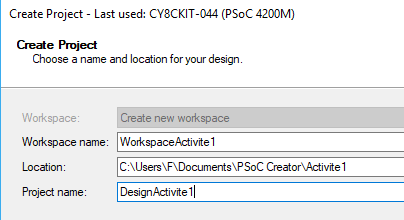
* Lancement de l’Environnement de Design Intégré :
* Création d’un nouveau projet : *File > New > Project…*
* La cible du projet est la carte du Kit CY8CKIT-044 (PsoC 4200M) :

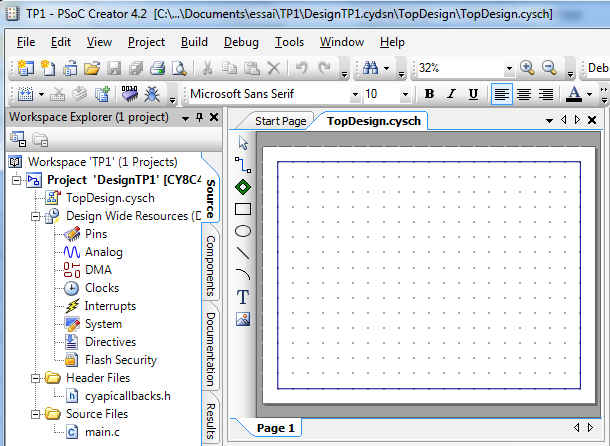


* On partira d’une conception vierge :



* Renseigner le nom du projet et l’emplacement des fichiers :

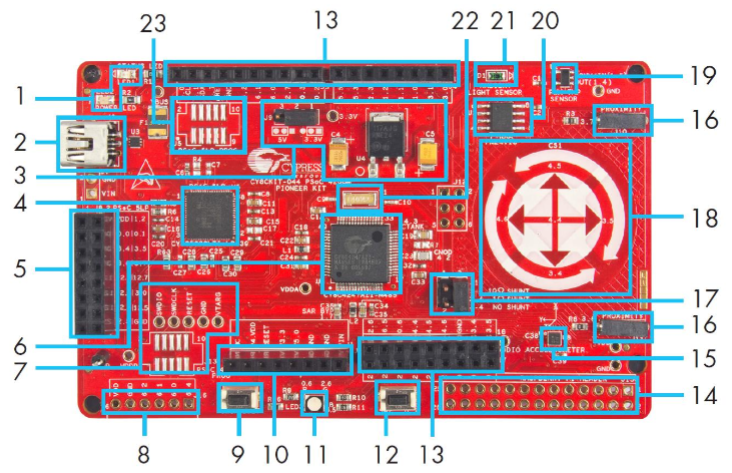


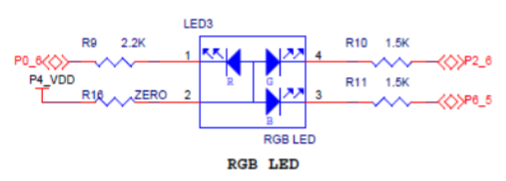


L’environnement est prêt…

## Clignotement d’une LED

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

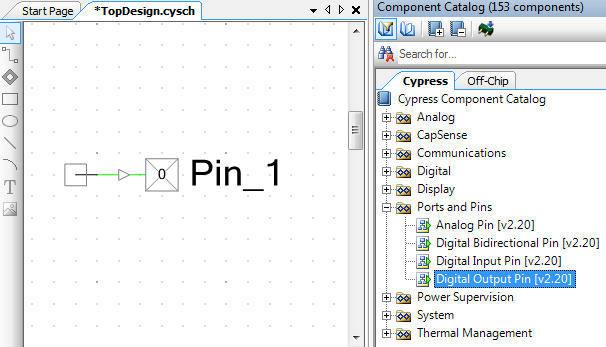




Les trois LED ont une anode commune reliée à l’alimentation VDD de la carte.

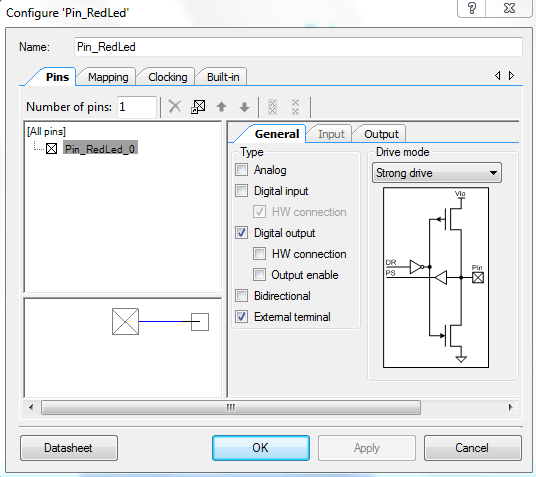
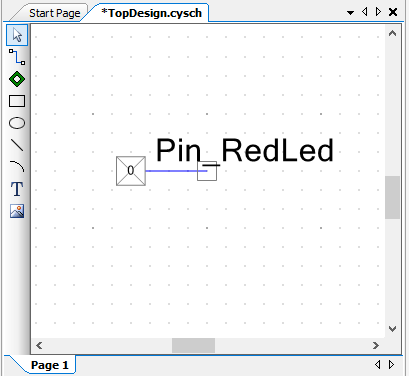
Pour allumer une LED, il faut donc relier sa cathode à la masse GND.

|  |  |
| --- | --- |
| **LED** | **Port cathode** |
| Rouge (R) | P0\_6 |
| Vert (G) | P2\_6 |
| Bleu (B) | P6\_5 |

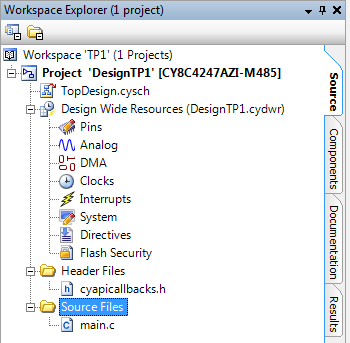


Depuis le catalogue des composants à droite, faire glisser le composant *Digital Output Pin* dans la fenêtre de conception *TopDesign* 🡪

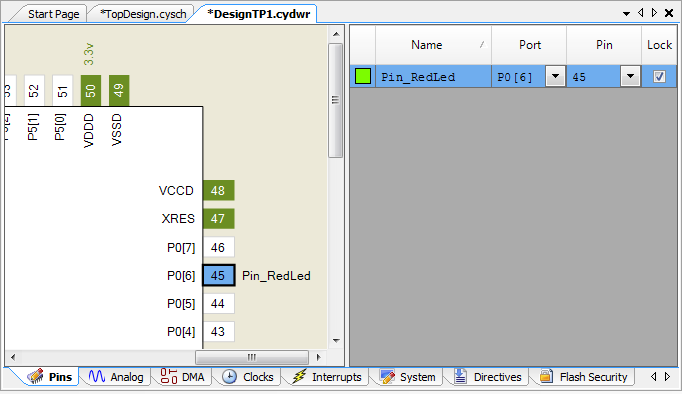
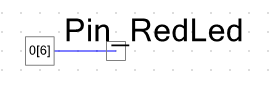
Double-cliquer sur le composant pour le configurer :



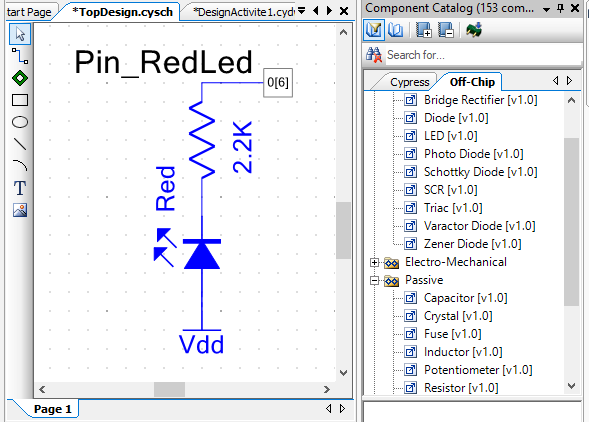
Dans l’explorateur, double-cliquer sur *Pins* :



Diriger le composant vers le port P0\_6, c’est-à-dire la cathode de la LED rouge :

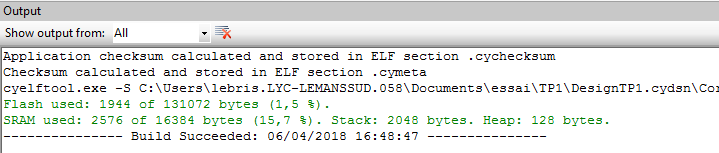


On peut agrémenter le schéma et faciliter sa lisibilité grâce au catalogue de composants, onglet *Off-Chip* :

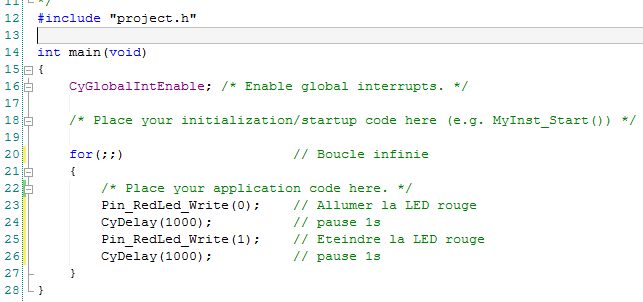


Outil de câblage

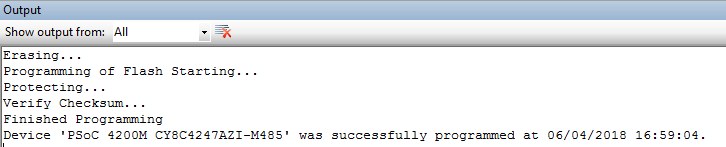
Générer le projet, menu *Build* (icône ) :



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



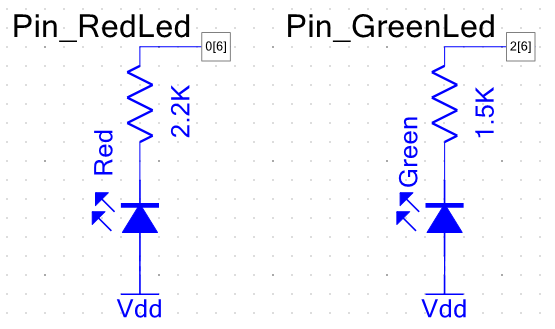
Connecter la carte PsoC au port USB du PC. Compiler et transférer le programme dans la puce en cliquant sur l’icône *Program*  :



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

## Exercice

Compléter le design  pour obtenir le schéma ci-dessous :

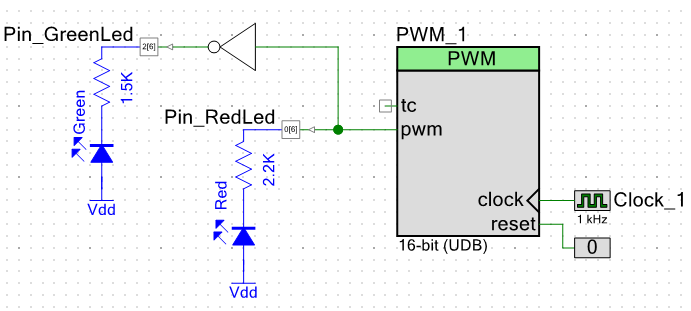


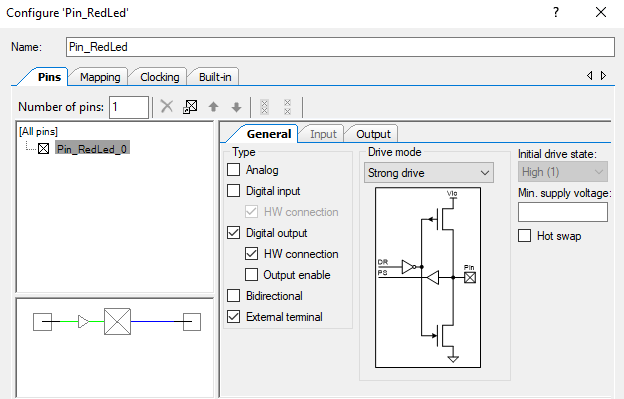
Modifier le programme pour faire clignoter la LED multicolore en rouge et en vert : la LED s’allume en rouge pendant 1 s, puis elle s’allume en vert pendant 1 s, puis à nouveau en rouge pendant 1 s, etc.

# Pwm 5steps.gifPour aller plus loin…

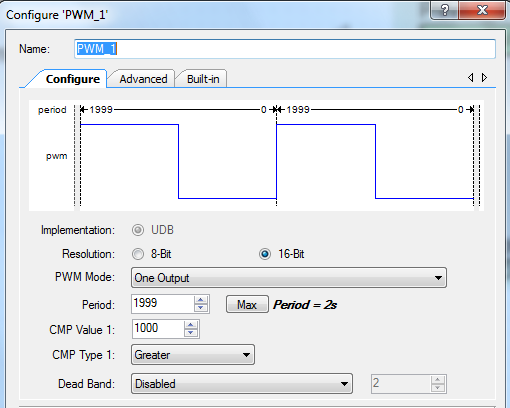
Le clignotement sera cette fois produit au niveau matériel grâce à un générateur de signaux PWM (*Pulse Width Modulation* ou Modulation en Largeur d’Impulsion).

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





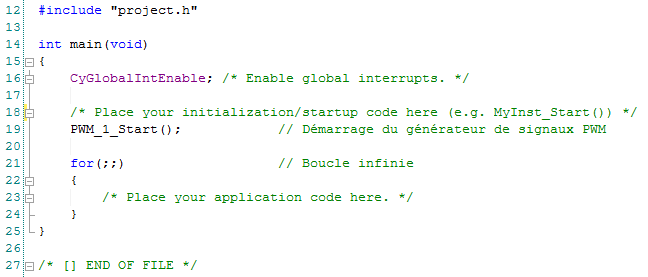
Pour relier les ports 0[6] et 2[6] au générateur PWM, il faut autoriser cette connexion en cochant   
 **HW connection** dans la boîte de configuration.

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 décroit 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 :



**Exercice supplémentaire :**

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