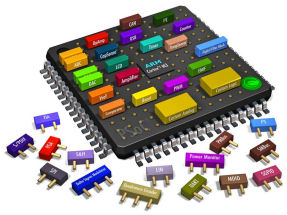
Prototypage électronique

Premiers pas avec PSoC 5LP



Objectifs : Être capable de développer des applications simples fonctionnant sur un PSoC 5LP en utilisant un kit de développement. Valider le fonctionnement de ces applications.

1. Présentation

Késaco PSoC ?

PSoC : Programmable System on Chip, est une famille de circuits intégrés introduits au début des années 2000 par Cypress. C'est un circuit intégré qui comprend un microcontrôleur et des fonctions **logiques** et **analogiques** configurables et inter-connectables entre eux.

Sur internet, vous pourrez trouver des tutoriels bien faits, certains en vidéo, en particulier sur le site de Cypress.

Tout au long de ce TP, vous travaillerez sur le kit de développement CY8CKIT-050 PSoC 5LP. Ce kit de développement comprends un circuit PSoC mais aussi de nombreux connecteurs, des boutons-poussoirs, des leds, etc. Ce kit permet de commencer très rapidement à développer et de tester des applications PSoC5. On utilise ce kit dans des phases de pré-étude ou de faisabilité.

Kit de développement CY8CKIT-050

Passée cette première étape de prise en main, il peut être intéressant de développer sur un kit de prototypage comme le CY8CKIT-059 afin de réaliser des projets plus « embarqués ». On utilisera ce type de kit pour du prototypage ou pour des produits développés en très petite série.

Kit de prototypage CY8CKIT-059

Enfin, pour le développement d'un produit industriel fabriqué en série, on utilisera un PSoC dans sa version composant, implanté sur une circuit imprimé, en y ajoutant des composants extérieurs.

Composant PSoC

Le but de ce TP est de vous présenter les possibilités de PSoC et vous montrer comment on peut très simplement réaliser rapidement des fonctions analogiques et numériques intégrées dans un circuit unique.

Vous allez réaliser une suite d'activités réalisant des fonctions de base et fonctionnant sur la carte d'évaluation.
Le premier tutoriel est très guidé, puis les tutoriels sont de moins en moins guidés pour arriver à des micro-projets où l'on ne donne que le cahier des charges. Pour chaque activité, vous indiquerez dans votre compte-rendu les broches utilisés, le câblage de la carte d'évaluation et vous fournirez une copie du schéma interne réalisé et du programme C. Les programmes C devront être correctement commentés. Vous détaillerez les manipulations vous permettant de valider la structure réalisée.

Présentation du poste de travail

Pour réaliser ces activités, vous disposez d'un ordinateur PC connecté à Internet et d'un kit CY8CKIT-050 PSoC 5LP. Le kit est connecté au PC par un câble USB. Sur le kit, c'est le connecteur USB en haut à droite qui est utilisé pour se connecter au PC.

Pour développer les applications PSoC, vous utiliserez le logiciel PSoC Creator 3.0.

Vous trouverez sur le poste de travail les fichiers suivants CY8CKIT-050B_Schematic.pdf (schéma structurel du kit) et PSOC_CREATOR_USER_GUIDE.pdf. (notice logiciel PsoC Creator).

Ces documents pourront vous être utiles au cours de ce TP.

2. Activité n°1 : tutoriel hello blinking world !

Objectif

C'est un grand classique lorsque on débute sur un microcontrôleur : on commence à faire clignoter une led. C'est très basique mais, si on y arrive, c'est qu'on est capable d'utiliser l'outil de développement, de charger un programme, de commander une sortie et de maîtriser le temps. C'est déjà bien ! Maintenant, certaines personnes d'ages certains vous diront qu'utiliser un super processeur (plus puissant que leur premier ordinateur) pour remplacer un NE555 (un petit circuit à 8 broches qui date des années 1970), c'est un peu du gâchis !

Cette activité (très tutoré) est inspirée de la vidéo suivante :

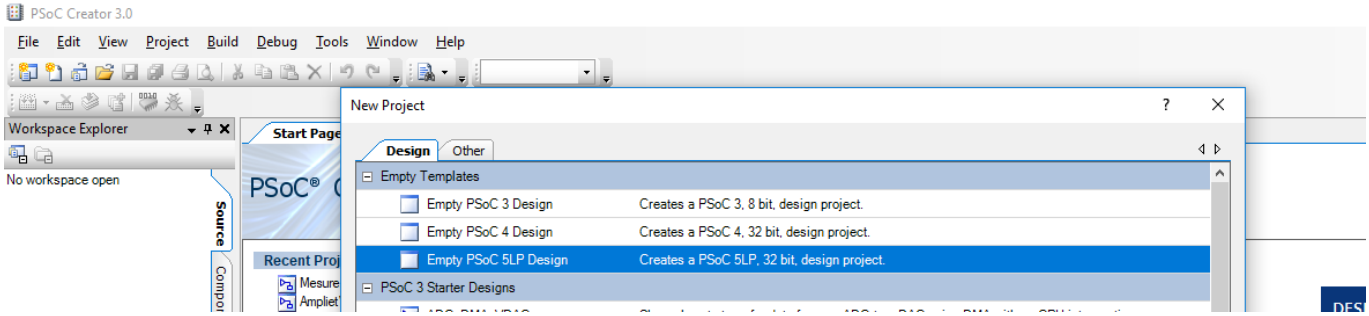
http://www.cypress.com/video-library/PSoC-Software/how-blink-led-using-pwm-component-psoc-5lp/108346

Document pdf disponible à cette adresse : http://www.cypress.com/file/137441/download

Tutoriel

Lancer PSoC Creator

Sélectionner **File > New> Project puis Empty PsoC 5LP Design** (cf figure).



Création du projet

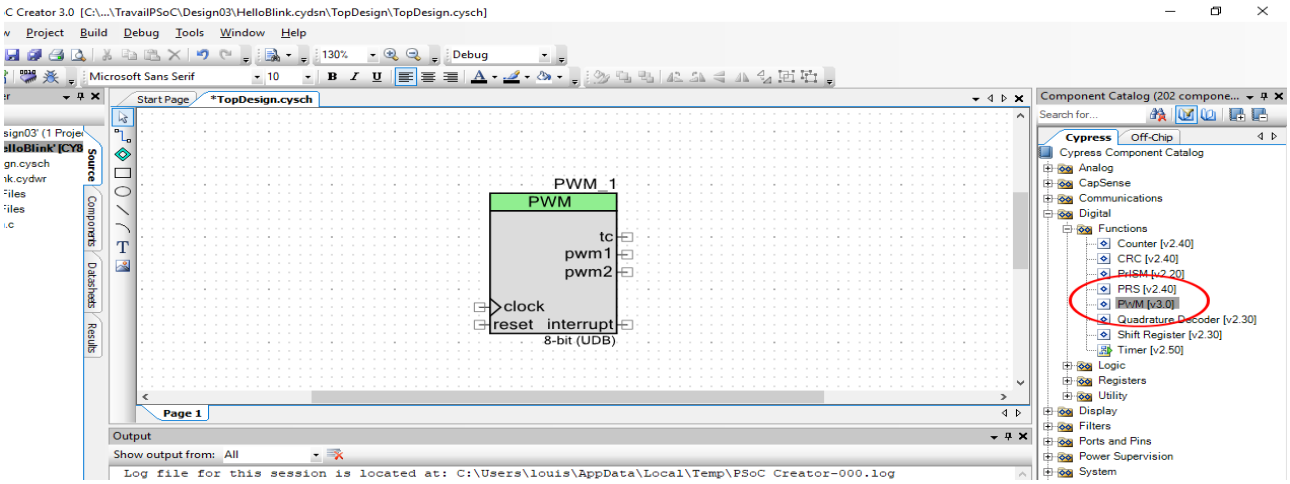
Puis donner un nom au projet.

Nous allons maintenant commencer à dessiner la structure de notre application.

Pour cela, on dépose des composants dans l'éditeur de schéma. Les composants sont disponibles dans la fenêtre : « **Component Catalog** »

Le premier composant à placer s'appelle PWM et est disponible dans **Digital>Functions**

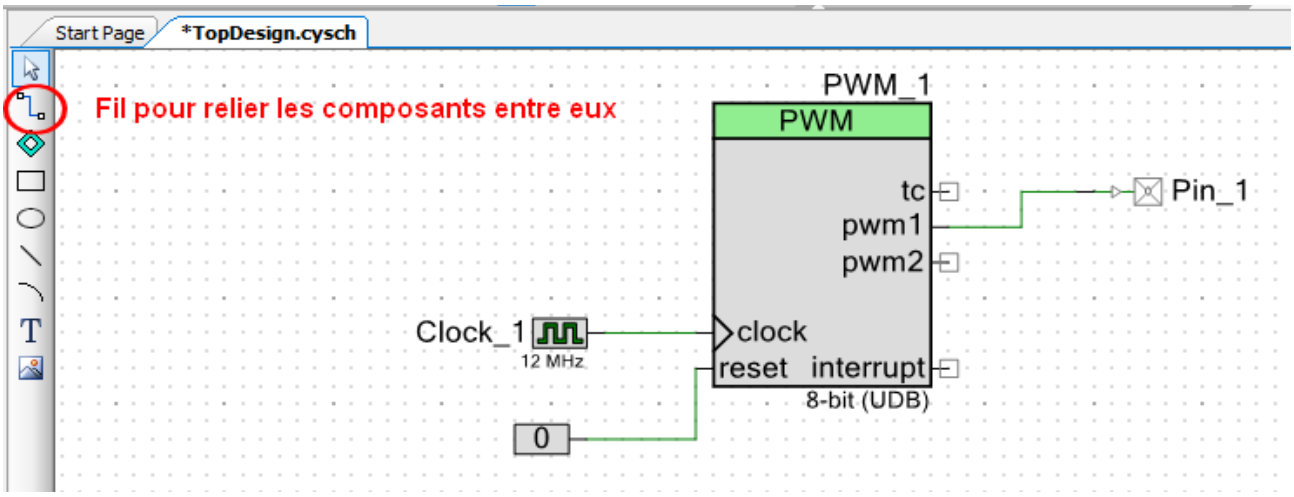
Avec un glissé-déposé, on place le composant au centre de la feuille.



Placement d'un PWM

On va ensuite chercher une horloge : **Clock** dans **System**, puis une borne de sortie : **Digital Output Pin** dans **Port and Pins** et un niveau 0 dans **Logic Level 0** dans **Digital > Logic**

On effectue ensuite le câblage conformément au schéma ci-joint :



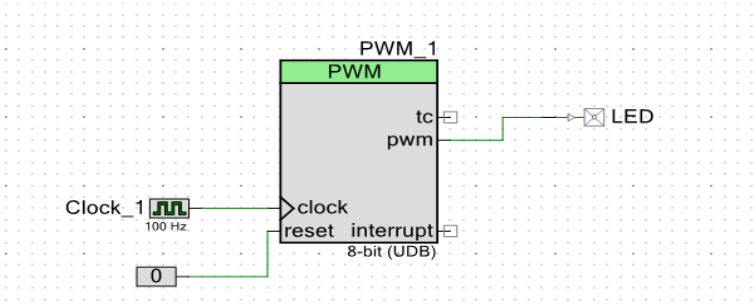
Câblage

On configure ensuite l'horloge et le PWM en double-cliquant dessus.

On configure d'abord l'**horloge** à une fréquence de **100 Hz** puis le **PWM** avec une période de **100** unités et une durée à l'état haut de **50** unités. On ne sélectionne qu'**une sortie** au PWM.

On modifie le nom de la broche de sortie pour l'appeler LED.

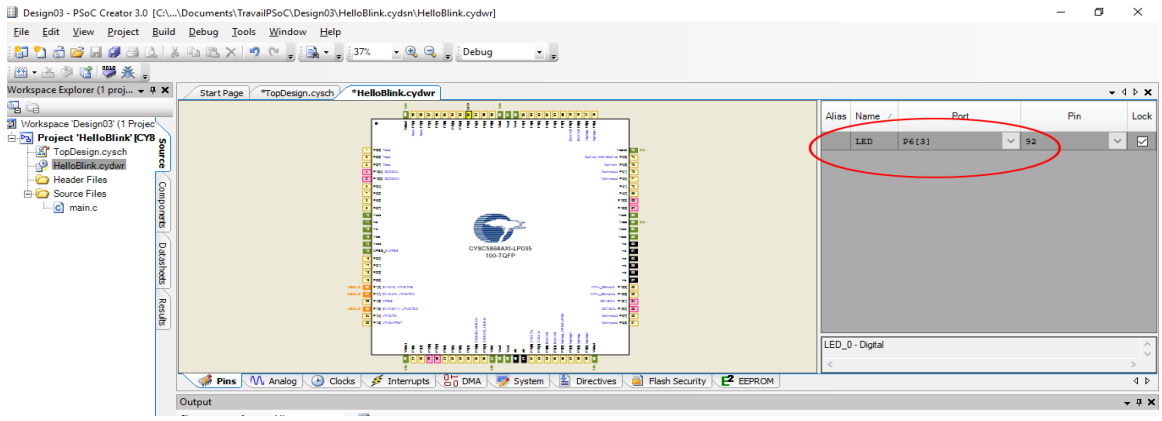
A ce stade, on doit obtenir une figure qui ressemble à la suivante :



PWM câblé et configuré

Remarque : pour accéder à la documentation d'un composant, il suffit d'un clic droit puis « Open datasheet ».

Il faut maintenant affecter les broches du circuit. Pour cela, on ouvre le fichier .cydwr et on affecte à la pin LED la broche P6[3] conformément à la figure suivante :

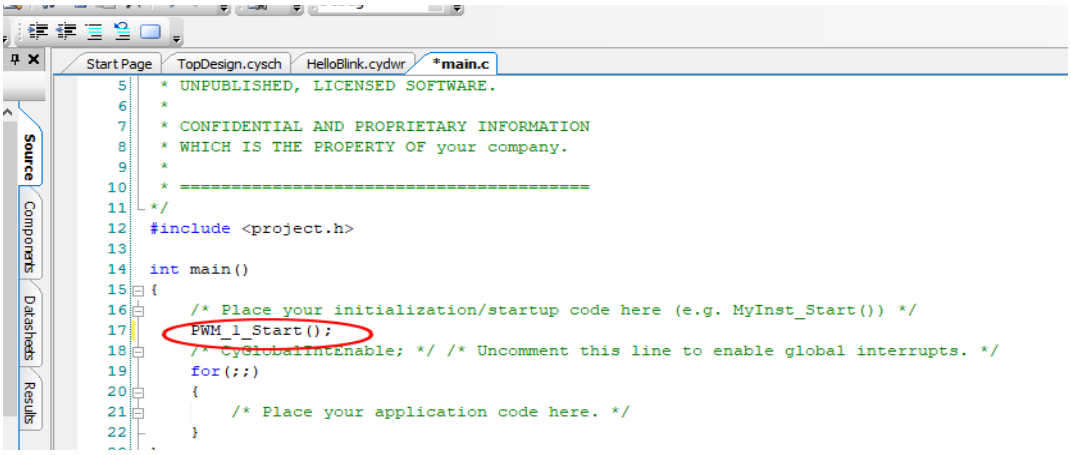


Affectation des broches

Remarque : sur la carte d'évaluation, la broche P6[3] du PSoC est reliée à la led LED4 (en bas au milieu de la carte d'évaluation).

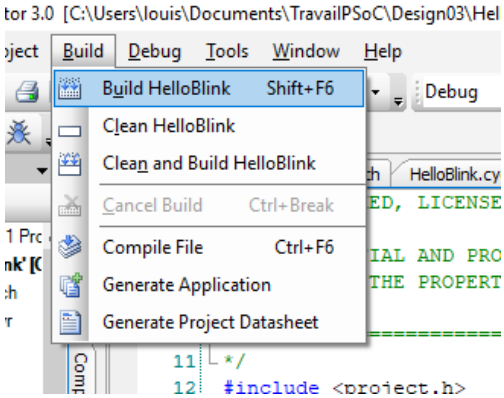
C'est presque terminé, il ne reste qu'à démarrer chaque composant (ici juste le PWM appelé PWM_1) dans le programme principal.

Pour cela, on ouvre le fichier main.c et l'on écrit la ligne suivante en début de programme :



Démarrage du composant PWM

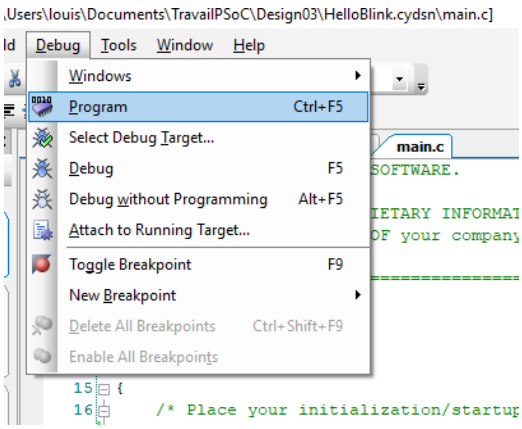
On compile maintenant les fichiers créés. Ceci ce fait avec la commande **Build**.



« Build » du projet

Remarque : il peut être parfois intéressant de compiler le programme avant d'écrire le code, cela permet de disposer de l'auto-complétion de l'éditeur C du PSoC.

On relie maintenant la carte d'évaluation au PC à l'aide de la prise USB située en haut à droite puis on télécharge l'application dans le PSoC avec la commande : **Debug>Program**



Programmation du composant

Si tout c'est bien passé, vous devez voir la LED4 clignoter à la fréquence d'une impulsion par seconde !

Modifications du projet

Vous allez effectuer quelques modifications sur le circuit **sans modifier la fréquence de l'horloge**.

Modification n°1 : Modifier la séquence d'allumage de la led pour qu'elle soit allumée pendant 1s , éteinte pendant 1s.

Modification n°2 : modifier la séquence d'allumage de la led pour qu'elle soit allumée pendant 0,1s éteinte pendant 1,9s

Modification n°3 : C'est maintenant la LED3 qui doit clignoter ! Indice : pour connaître la broche qui est connectée à la LED3, se reporter aux indications présentes sur la sérigraphie de la carte.

3. Activité n°2 : tutoriel lire et écrire un bit de 2 façons !

Lire une entrée et commander une sortie. C'est encore une opération très basique mais nous allons voir comment la faire de deux manières différentes.

Cahier des charges : Il s'agit dans ce projet d'allumer la LED4 lorsque on appui sur le bouton-poussoir SW2.

On donne ici le schéma de la zone « espace de prototypage », comprenant entre autres, l'interrupteur SW2 et la LED4 qui seront utilisés dans ce TP.

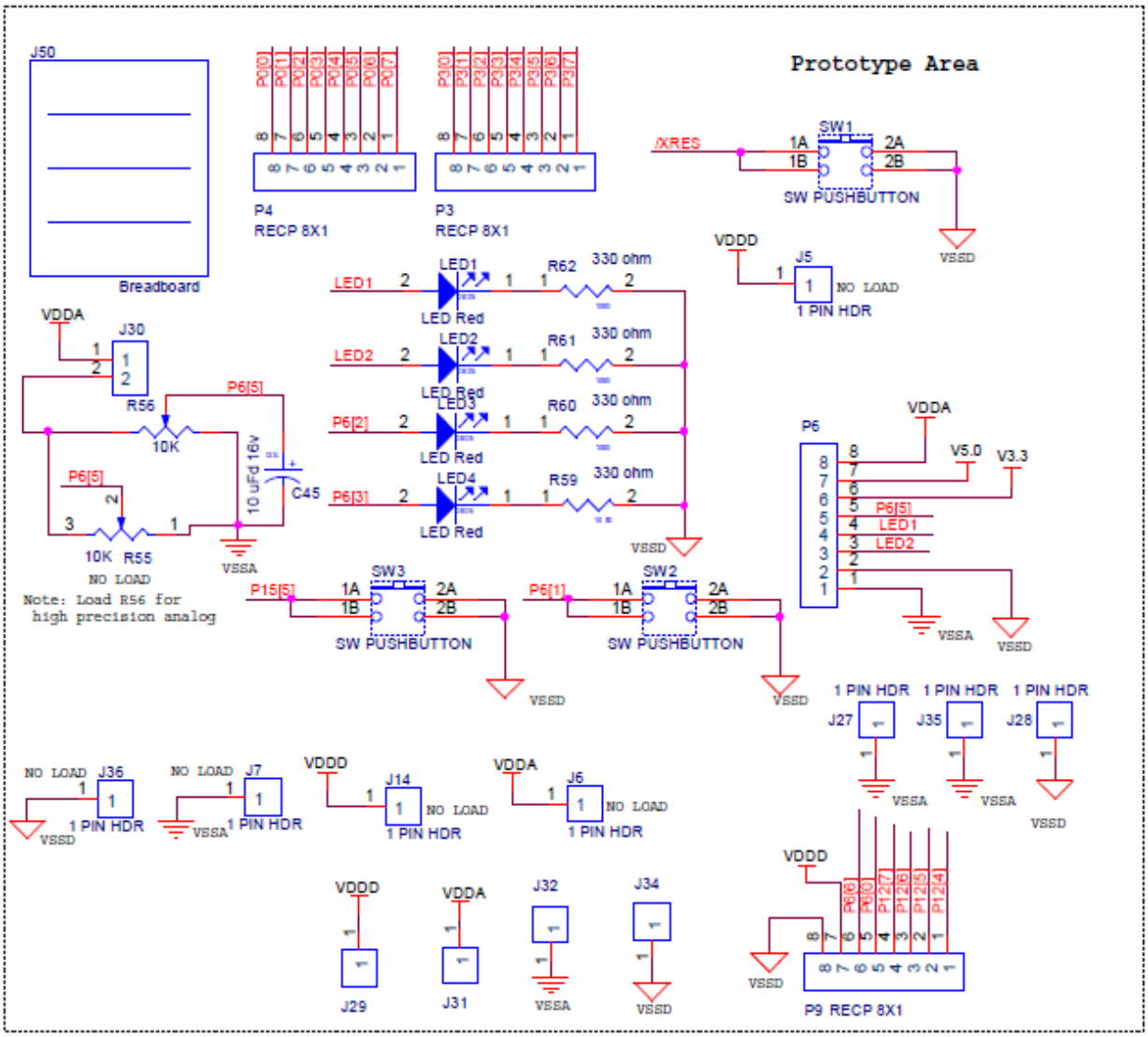
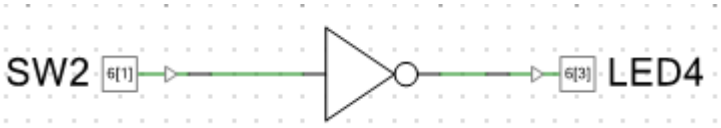


Schéma zone prototypage de la carte d'évaluation

Tutoriel 2A : Lire et écrire un bit avec du « hard »

Pour ce faire, il n'est pas besoin d'écrire du logiciel mais simplement de saisir la structure suivante.



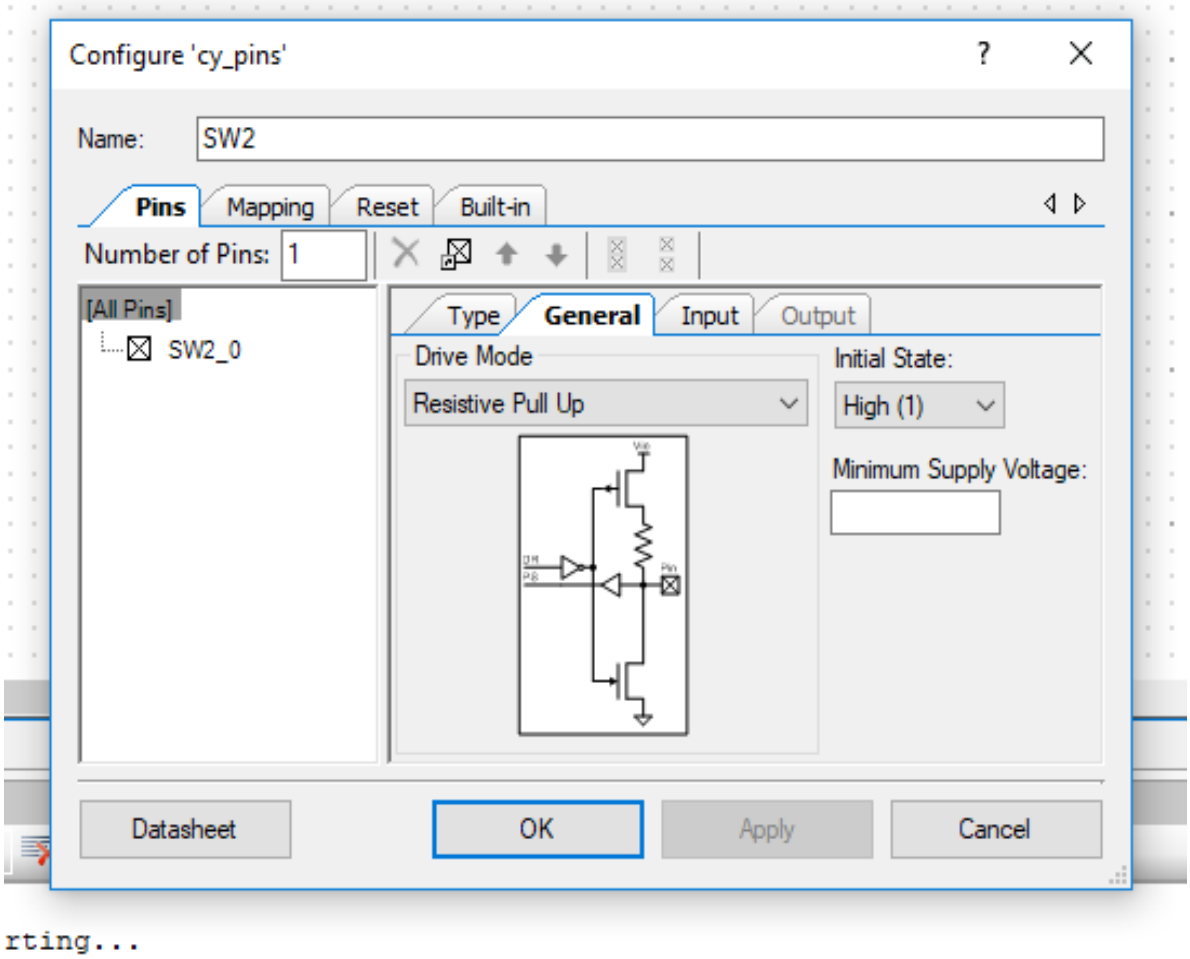
Lire et écrire un bit avec du « hard »

SW2 est définie comme une entrée logique (Digital Input Pin) et LED4 comme une sortie logique (Digital Output Pin).

Redessiner le schéma en y faisant figurer le bouton-poussoir SW2 et la del LED4. Indiquer quel niveau permet d'allumer LED4 et justifier la nécessité de placer une résistance de tirage sur SW2.

Pour que la structure fonctionne, il est nécessaire de placer une résistance de tirage (interne) sur l'entrée SW2 !

Cela se fait en double-cliquant sur la broche SW2.



Configurer un pull-up sur une entrée

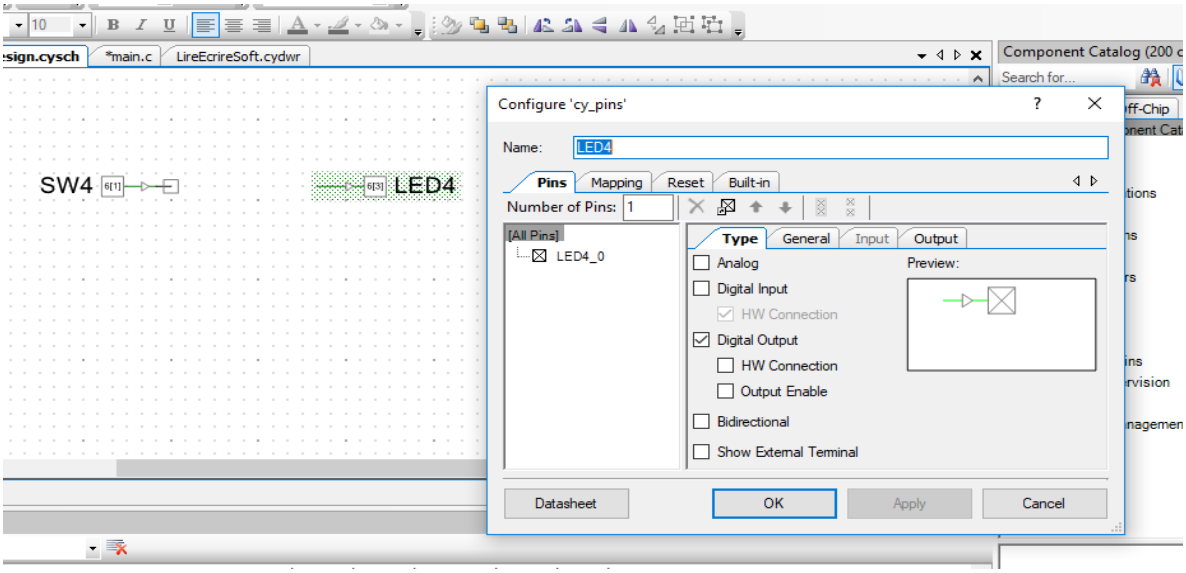
Travail demandé : Réaliser puis tester le projet ! Justifier la structure utilisée.

Tutoriel 2B : Lire et écrire un bit avec du « soft »

La structure est simplement la suivante :

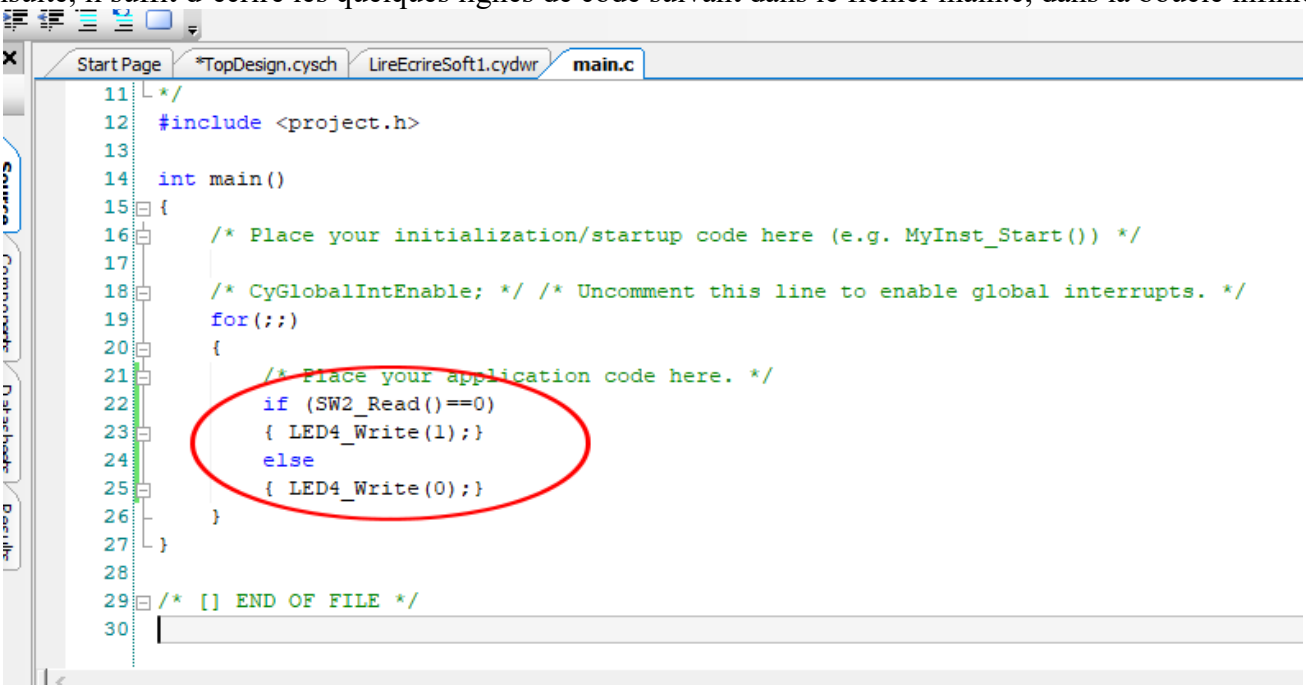
Structure pour lire et écrire un bit avec du « soft »

On remarquera que le broche de sortie n'est pas reliée à un petit carré. Cela se fait en décochant l'option HW Connection dans la configuration de la broche, conformément à la figure suivante.



Configuration de la sortie LED4 pour commande par « soft »

Ensuite, il suffit d'écrire les quelques lignes de code suivant dans le fichier main.c, dans la boucle infinie :



Code permettant de lire une entrée et écrire une sortie

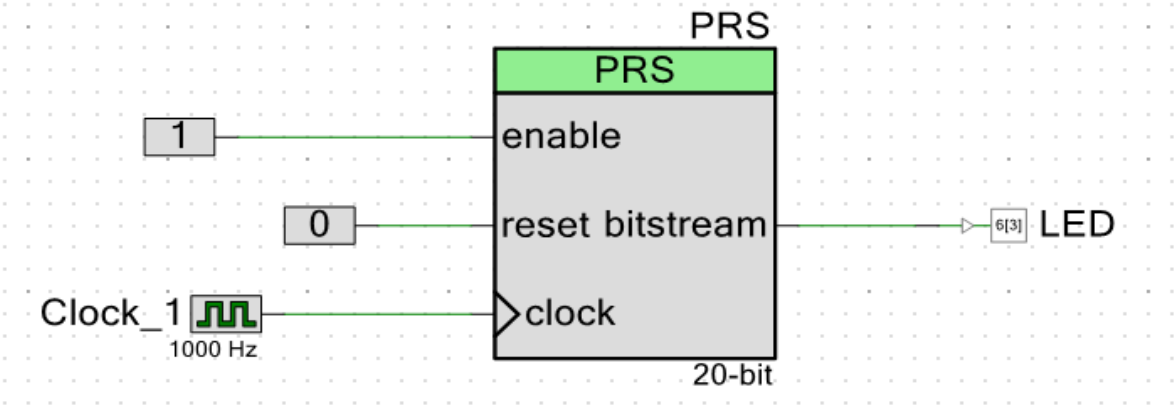
Travail demandé : Réaliser puis tester le projet ! Justifier la structure logicielle utilisée.

4. Activité n°3 : tutoriel « une bougie virtuelle »

Après avoir fait clignoter une led, allumer puis éteindre une led, que faire de plus avec une led ? On se propose de faire ici une led qui scintille à la manière d'une bougie virtuelle !

Pour cela, on va commander la led par la sortie d'un générateur pseudo-aléatoire. Un générateur pseudo-aléatoire est un circuit qui délivre à chaque période d'horloge un signal 0 ou 1 de manière (presque) aléatoire.

La structure est donnée :



Structure « bougie virtuelle »

Travail demandé : faire fonctionner le projet et le tester en visualisant l'état de le led.

Conseil : ne pas oublier de démarrer le générateur pseudo-aléatoire PRS !

5. Activité n°4 : micro-projet « une bascule RS »

Le cahier des charges est le suivant :
Entrée : Boutons-poussoirs SW2 et SW3
Sortie : LED 4
Un appui sur SW3 doit allumer la led, elle reste allumée si on relâche SW3.
Un appui sur SW2 doit éteindre la led, elle reste éteinte si on relâche SW2.

Travail demandé :
On demande la réalisation de ce projet de 2 façons différentes.
1°) En utilisant du logiciel uniquement
2°) En utilisant un composant S R flip-flop avec une horloge de fréquence 1 kHz.

Conseils : bien regarder la documentation du composant SR flip-flop ! Penser aux pull-ups !

Bon courage !

6. Activité n°5 : micro-projet hello lcd world bilingue

Créer un nouveau projet, placer dans la structure un afficheur LCD « Character LCD ».
Affecter les broches du LCD à P2[6 :0]
Insérer dans le fichier main.c les deux lignes qui permettent :
- de démarrer l'afficheur,
- d'afficher la phrase « Hello World ! »

Pour cela, vous devrez consulter la « data sheet » du composant afficheur LCD.

Travail demandé : Implanter le projet et tester le bon fonctionnement.

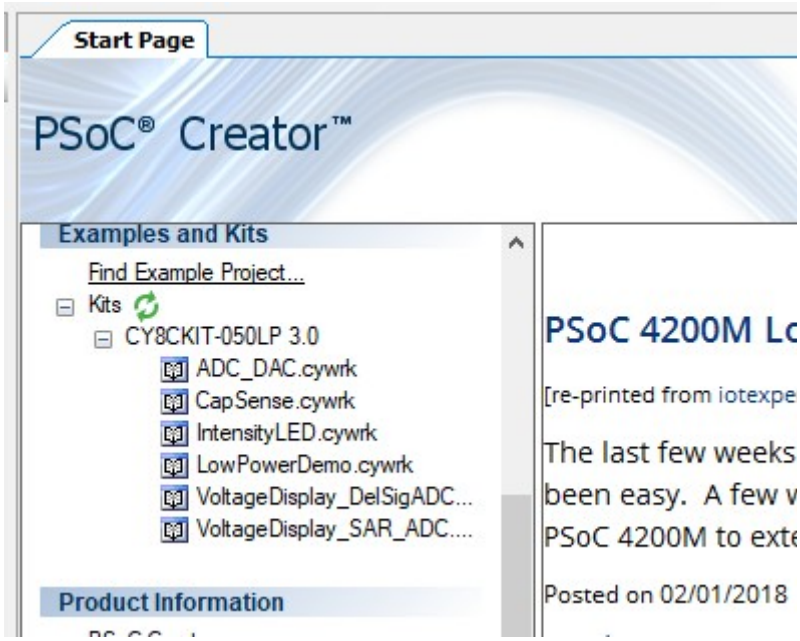
Modification : Modifier le projet pour que l'afficheur affiche « Bonjour monde ! » et allume LED4 lorsque l'on appui sur SW2. Un appui sur SW3 permet d'éteindre la led et de retrouver l'affichage « Hello World ! ».

7. Activité n°6 : mise en œuvre d'un convertisseur analogique-numérique

Utilisation d'un fichier exemple

Au cours de cette activité, vous allez modifier un fichier d'exemple donné par Cypress.

A partir de la « Start Page », sélectionner le projet exemple « VoltageDisplay_SAR_ADC ».



Projets exemples

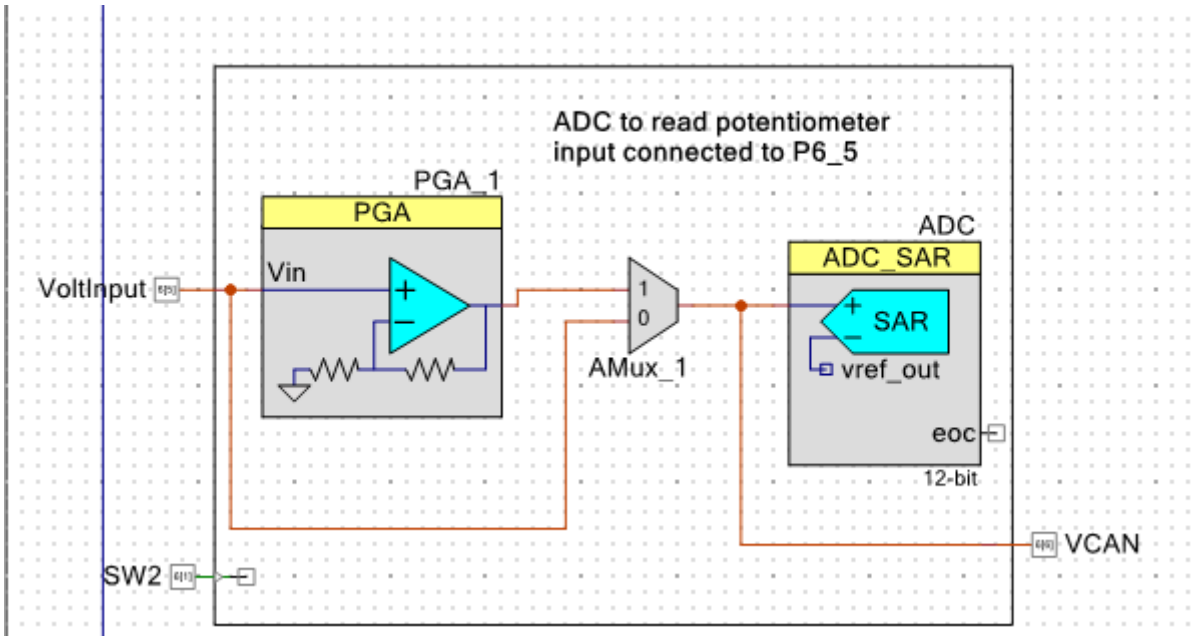
Travail demandé :

Analyser la structure et le fichier main.c.

Construire et programmer le projet. Valider le fonctionnement. Indiquer le rôle de cette structure.

Modification : ajout d'un amplificateur

On se propose de modifier la structure conformément au schéma joint.



Modification de la structure

On a rajouté sur cette structure :

- un amplificateur programmable PGA (amplificateur non-inverseur) de gain 2.

- un multiplexeur analogique qui permet d'aiguiller 2 signaux analogiques. On envoie sur l'entrée du convertisseur soit la tension à l'entrée de l'amplificateur, soit la tension à la sortie de l'amplificateur.

- une broche VCAN qui permet de mesurer avec un voltmètre par exemple la tension mesurée par le convertisseur analogique-numérique.

Travail demandé :

- écrire le programme qui permet en appuyant (ou pas) sur SW2 de mesurer et d'afficher sur l'afficheur LCD, la tension d'entrée de l'amplificateur ou la tension de sortie.

- tracer à l'aide de la carte d'évaluation la caractéristique de transfert de l'amplificateur sur tableur Vs = f(Ve). Conclure