



Sciences et technologies de l'Industrie et du développement durable

Bac STI2D Formation des enseignants

Jean-François LIEBAUT – Denis PENARD

SIN 63 : Prototypage d'un traitement de l'information
analogique et numérique (PSoC)

TP introductif PSoC
Gestion du port GPIO

1. POINTS ABORDES

Etude du port GPIO avec la mise en place des entrées-sorties numériques.

Implémentation d'un générateur MLI (PWM) à 10 kHz.

1.1. CREATION DU PROJET « BOUTON-POUSOIR »



Un tutoriel vidéo pour la création d'un projet PSoC est disponible dans les ressources sous le nom « tutoriel_video ».

Lancer l'environnement de développement PSoC Designer 5.1 et créer un nouveau projet :

- nom du projet : gestion_bp
- langage de programmation : langage C

Microcontrôleur (Device) : CY8C24966-24PXI (remplacer éventuellement par la référence de votre microcontrôleur).

Une fois le projet créé, vous allez devoir configurer le port GPIO.

1.2. MISE EN PLACE DU BOUTON POUSOIR ET DE LA LED

Vous devez mettre en place un bouton poussoir sur une broche du PSoC pour piloter une LED en sortie.

Rechercher dans la documentation « DataSheet_C8Y29x66_E.pdf » quelles broches peuvent être utilisées en entrée et en sortie numérique (chapitre « Pinout »).

Choisir une broche sur laquelle vous allez câbler le bouton poussoir : _____

Choisir une broche sur laquelle vous allez câbler la LED : _____

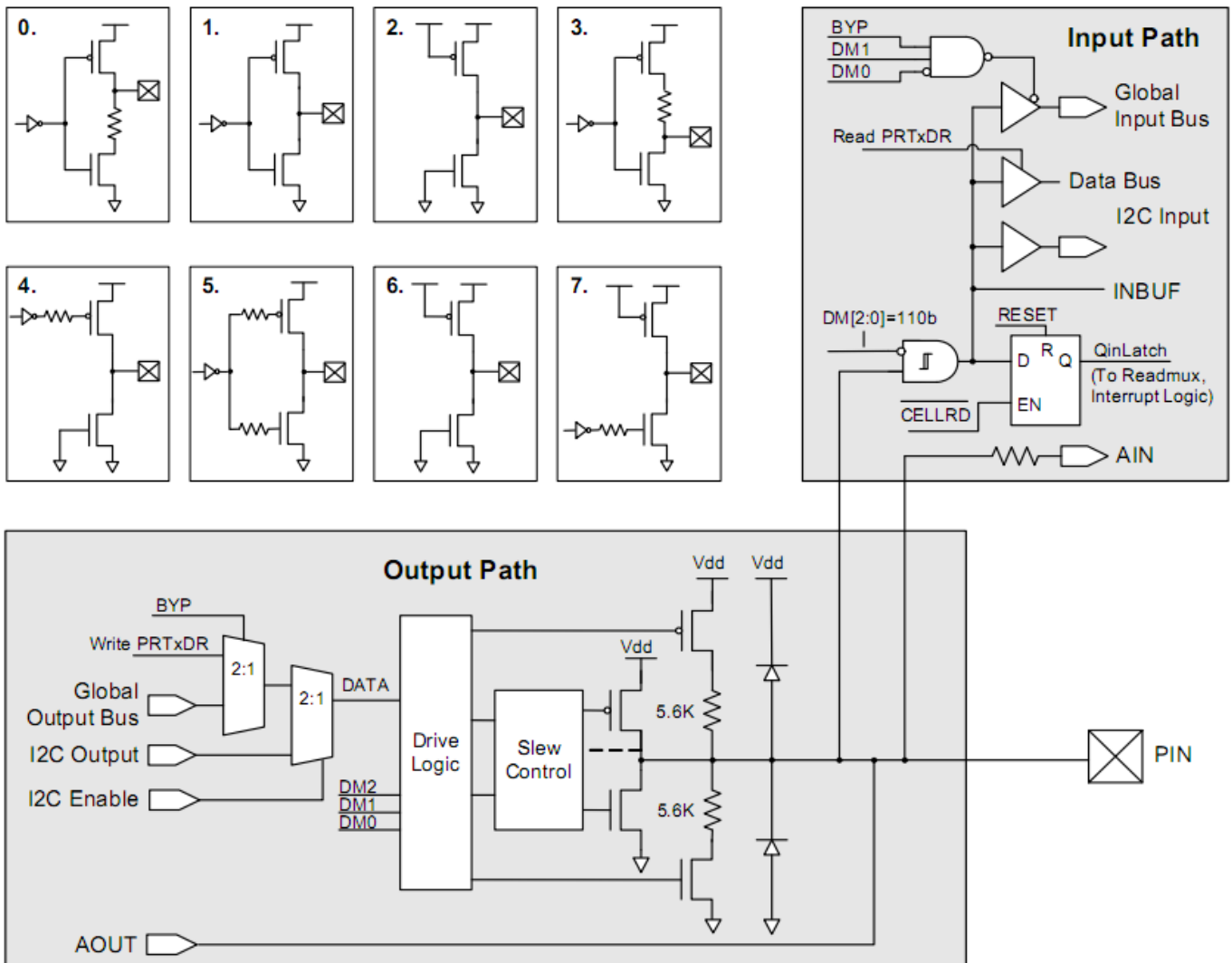
Analyser le schéma de la carte CY3210 « PSoCEval1_PDC-9286A_Schematic.pdf » et déterminer quel type de résistance de rappel il faut prévoir pour le bouton poussoir (S1) (pull-up ou pull-down) : _____

Rechercher à partir de la figure ci-après qu'elle est la configuration (Drive Mode) à appliquer à la broche reliée au bouton poussoir. : _____

Faire de même pour la broche reliée à la LED : _____

Drive Modes

DM2	DM1	DM0	Drive Mode	Diagram Number	Data = 0	Data = 1
0	0	0	Resistive Pull Down	0	Resistive	Strong
0	0	1	Strong Drive	1	Strong	Strong
0	1	0	High Impedance	2	High Z	High Z
0	1	1	Resistive Pull Up	3	Strong	Resistive
1	0	0	Open Drain, Drives High	4	High Z	Strong (Slow)
1	0	1	Slow Strong Drive	5	Strong (Slow)	Strong (Slow)
1	1	0	High Impedance Analog	6	High Z	High Z
1	1	1	Open Drain, Drives Low	7	Strong (Slow)	High Z



Détails d'un port d'entrée-sortie GPIO

Assigner la configuration dans votre projet dans la section Pinout de PSoC Designer
5.1. pour les broches d'entrée et sortie choisies.

Pinout - avirer2	
<input checked="" type="checkbox"/> P0[0]	Port_0_0, StdCPU, High Z Analog, Disablelr
Name	Port_0_0
Port	P0[0]
Select	StdCPU
Drive	High Z Analog
Interrupt	DisableInt
InitialValue	0
<input checked="" type="checkbox"/> P0[1]	Port_0_1, StdCPU, High Z Analog, Disablelr
<input checked="" type="checkbox"/> P0[2]	Port_0_2, StdCPU, High Z Analog, Disablelr



Vous pouvez renommer une broche (ici Port_0_0) par un nom de votre choix afin d'améliorer la compréhension du projet.

Terminer cette étape en choisissant - **Generate Configuration files for 'gestion_bp' Project** - du menu Build ou bien en cliquant sur l'icône associée.

Par la suite, vous allez travailler dans le programme en C du fichier main.c.

1.3. PILOTAGE DE LA LED AVEC LE BOUTON POUSSOIR

Vous devez allumer la led à chaque appui sur le bouton poussoir.

Le registre permettant d'accéder en lecture ou écriture au port 0 est :

PORT0DR

Une lecture du port 0 qu'on stocke dans la variable valeur_port0 donne :

```
valeur_port0 = PORT0DR ;
```

Une écriture sur le port 0 s'écrit :

```
PORT0DR = 0x01 ; //si on veut mettre à 1 le bit 0 (les autres bits seront mis à 0).
```

On constate que l'accès aux ports GPIO est classique. La méthode des masques pour isoler le bit lu et le bit écrit est plus élégante.

- Ouvrir le fichier main.c en le sélectionnant dans l'arborescence (**Workspace Explorer** -> **gestion_bp /Sources Files/main.c**) et rajouter une boucle dans la fonction main() qui effectue une lecture du bouton poussoir et qui allume ou éteint la LED en fonction de la valeur lue.

1.4. PROGRAMMATION DU PSoC ET TEST

- Compiler et linker le projet
- Choisir la commande " **Build 'gestion_bp ' Project F7** " du menu Build.
- Choisir " **Program Part...CTRL+F10** " du menu Program.

Pour la programmation, connecter le programmeur miniprogram à la carte d'un côté et au PC par l'USB de l'autre. Il y a deux manières de programmer :

- Mode Reset : le programmeur utilise la broche XRES du PSoC. A la fin de la programmation, le PSoC reste alimenté.
- Mode Power : le programmeur utilise une séquence sur la broche d'alimentation pour faire passer le PSoC en mode programmation. Le PSoC ne reste pas alimenté à la fin de la programmation.

Dans tous les cas, prendre son mal en patience... et réfléchir aux prochaines modifications à apporter à son programme !



Vous pouvez diminuer le temps de programmation en prenant un PSoC doté de moins de mémoire (prendre par exemple un 27443 au lieu d'un 29466).

1.5. FAQ

La LED reste allumée bien que le bouton poussoir soit relâché et que le test du bouton poussoir dans le programme soit correct !

L'analyse de la figure précédente détaillant une broche du GPIO fait apparaître dans la section INPUT PATH la présence d'un buffer. Avant de lire la broche, il faut écrire la valeur inverse sur le port pour « remettre à zéro » le buffer.

1.6. CREATION DU PROJET « GENERATEUR-PWM »

Lancer l'environnement de développement PSoC Designer 5.1 et créer un nouveau projet.

- nom du projet : generateur_pwm
- langage de programmation : langage C

Microcontrôleur (Device) : CY8C24966-24PXI (à remplacer éventuellement par la référence de votre microcontrôleur).

Une fois le projet créé, vous allez devoir configurer le user module PWM.

3 étapes seront nécessaires pour faire fonctionner le générateur PWM :

- Installer le user module
- Configurer le user module
- Utiliser le user module.



Il est fortement conseillé de lire la « datasheet » du user module que vous utilisez. Cette documentation contient toute la description du bloc, les routines en C qui permettent d'utiliser le module, ainsi qu'un exemple de code en langage assembleur et en C.

1.7. MISE EN PLACE DU MODULE PWM

Vous devez mettre en place un générateur de signal PWM sur une broche du PSoC. Le signal désiré à une fréquence de 10 kHz et un rapport cyclique de 50%.

Choisir " User Module Catalogue " du menu View ou bien sélectionner l'onglet " User Module " à droite de la fenêtre de PSoC Designer.

Choisir le module PWM de la section " PWMs " et le placer par un clique-droit -> place.

Choisir une broche sur laquelle vous allez « sortir » le signal PWM : _____

Pour choisir un module PWM (8 bits ou 16 bits), il faut déterminer la fréquence d'horloge à appliquer, ainsi que la valeur à stocker dans le « Period Register ». Plusieurs couples - horloge/valeur du diviseur du timer - sont possibles. Le choix peut dépendre des besoins en horloge des autres blocs utilisés dans le PSoC.

Calculer la fréquence de l'horloge à appliquer au bloc PWM ainsi que la valeur à stocker dans « Period » pour obtenir une fréquence sur sa sortie (TerminalCountOut de 10 kHz).

PWM8 bits ☐

PWM16 bits ☐

Parameters - PWM8_1	
Name	Pwm8_1
User Module	Pwm8
Version	2.60
Clock	
Enable	
CompareOut	
TerminalCountOut	
Period	0
Pulsewidth	0
CompareType	
InterruptType	
ClockSync	
InvertEnable	Normal

Terminer cette étape en choisissant - " **Generate Configuration files for "generateur_pwmjet" Project** - du menu Build ou bien en cliquant sur l'icône associée.

Par la suite, vous allez travailler dans le programme en C du fichier main.c.

Analyser le fichier PWM_1.h (si 'PWM_1' est le nom de votre module PWM). Ce fichier contient les prototypes des fonctions permettant de modifier par programmation la PWM.

Insérez dans votre programme principale du fichier main.c la fonction qui démarre le bloc PWM : **PWM_1_Start()** ;

Vous pouvez modifier dans votre programme le rapport cyclique en jouant sur la valeur du registre **PulseWidth** avec la fonction « qui va bien » (se référer à la documentation).



La valeur de PulseWidth doit être inférieure à celle de Period car la pseudo-période est une sous-division de la période.

1.7.1. Test du générateur PWM

Câbler en interne la sortie du module PWM (TerminalCountOut) sur une sortie disponible du PSoC et visualiser le signal à l'oscilloscope. Vérifier que le bloc PWM est bien configuré pour générer un signal à 10 kHz.

1.8. EVOLUTIONS POSSIBLES

Par la suite, vous pouvez piloter le rapport cyclique avec un potentiomètre.

Câbler le potentiomètre sur une entrée analogique du PSoC, convertir la tension issue du potentiomètre et appliquer le résultat de la conversion au registre PulseWidth. Un exemple est fourni dans les ressources analogiques.