

DMA et pilotage de la LED RGB de la carte MyLab

Labo PRO

Semestre hiver 2016-2017

Introduction

Ce laboratoire va vous permettre d'utiliser la carte d'extension **MyLab** avec la carte processeur **LPCXpresso** ainsi que mettre en oeuvre la LED RGB à l'aide du DMA.

Aucun projet à importer ne vous est donné. Pour ce laboratoire, nous vous demandons de créer un nouveau projet de type C pour LPC1769 à l'aide de l'assistant de LPCXpresso. (Note : Il sera important de choisir le CMSIS que vous avez déjà dans votre workspace.)

1 Carte MyLab2

Les LEDS RGB WS2812 sont connectées au GPIO P2.8 du micro-contrôleur.

Attention : Pour des raisons techniques la sortie du micro-contrôleur est inversée avant d'arriver sur l'entrée DIN de la première LED. Vous devez faire attention à bien générer les signaux inversés par rapport à la documentation technique de la LED WS2812. En cas de doutes prenez un oscilloscope ou analyseur logique.

2 Exercice

Exercice 1: Contrôle d'une LED RGB

La LED RGB sur la carte **MyLab** intègre un contrôleur PWM (voir documentation technique WS2812). Il est extrêmement complexe de la contrôler sans DMA car il faut garantir des délais précis de durée inférieure à la microseconde. L'utilisation de la mémoire et de la copie vers le GPIO par DMA permet de résoudre ce problème.

Méthodologie pour contrôler une LED WS2812 :

1. Préparer un tableau `uint8_t tabRGB[97]` ; avec les données RGB à afficher. Nous vous recommandons de réaliser une fonction ayant pour paramètre `R, G, B`.
note : Il est important de garantir que la ligne soit à l'état 0 avant le transfert du premier

bit dans le but que le premier flanc montant soit correctement détecté par la LED. Il est ainsi recommandé de mettre en place un tableau de 96 octets plus 1 ; le premier élément étant égal à 0.

2. Activer un évènement périodique TIMER toute les 300 ns. Il n'est pas nécessaire d'activer les interruptions. Pour régler l'horloge PCLK à 100 MHz utilisez le registre PCLKSEL.

Selon le TIMER choisi, modifiez le registre PCLKSEL0 afin de mettre PCLK/1 au lieu de PCLK/4.

Exemple : Pour utiliser le TIMER0 à 100 MHz

```
PCLKSEL0 = 1<<2;
```

3. Configurer le transfert DMA afin d'écrire le contenu du tableau vers le registre FIOPIN correspondant à la LED RGB.
 - (a) Activer l'alimentation électrique pour le DMA (PCONP)
 - (b) Activer le contrôleur DMA (DMACConfig). L'endianness doit garder la valeur par défaut.
 - (c) Choisir un canal DMA (parmi les 8 à disposition) et régler la source sur l'adresse de début du tableau tabRGB (DMACCxSrcAddr)
 - (d) Régler l'adresse de destination du DMA sur le registre correspondant à la fonction FIOxPIN (DMACCxDestAddr). Attention à l'existence des variantes du registre FIOxPIN_y où **y** permet de faire des accès par octet ou mots de 16 bits. Ceci vous sera utile afin que le DMA accède uniquement à l'octet contenant la sortie à manipuler et non pas aux 32 bits du port.
 - (e) Configurer correctement le registre (DMACCxControl). A savoir :
 - Spécifier la taille du transfert, à savoir ici le nombre de bytes que le DMA devra copier.
 - Nous ne ferons pas de transferts de type BURST. Ceux-ci s'appliquent surtout aux copies rapides mémoire à mémoire.
 - Nous allons transférer des octets et non pas des mots de 32 bits.
 - Spécifier si après chaque transfert les adresses de source et de destination doivent être incrémentées ou non.
 - Nous garderons les valeurs par défaut aux Protx.
 - Nous n'utiliserons pas d'interruption DMA dans cet exercice. Celles-ci peuvent être utilisées pour les exercices suivant pour, par exemple, générer une interruption à la fin du transfert DMA.
 - (f) Configurer le registre (DMACCxConfig).
 - SrcPeripheral et DestPeripheral ne permettent pas de spécifier le périphérique de source ou de destination (comme on pourrait l'imaginer sans lire la description) mais servent à dire quel périphérique va déclencher le transfert DMA suivant. Donc il s'agira de spécifier le Match du TIMER sélectionné (précédemment au point 2) comme destination de contrôle. Au préalable, sélectionner la source TIMER et non pas UART dans le registre DMAReqSel.
 - Spécifiez le type de transfert
 - Les valeurs par défaut peuvent être gardées pour les autres bits.
 - (g) Lancez le transfert DMA en activant le bit Channel enable du registre (DMACCxConfig).

note : Ne pas oublier de régler en sortie le GPIO correspondant à la LED RGB (une fonction d'initialisation spécifique est la bienvenue). Un masque FIOMASK devra être utilisé, étant donné que nous ne souhaitons modifier que l'état d'une seule sortie.

en cas de problème : Demandez un analyseur logique ou oscilloscope pour vérifier votre signal en entrée de la première LED...

Exercice 2: Contrôle de plusieurs LED

Adaptez votre code pour contrôler les 2 LEDS de la carte MyLab2. Vous pouvez réfléchir à une solution générique pouvant s'adapter à n LED.

Exercice 3: Effets lumineux

Activer un autre TIMER générant un événement toutes les ms. Préparer un nouveau contenu de tableau avec un algorithme propre à vous. Le but étant de générer des effets lumineux sur les LED.