

Exercice révision chapitres 5 à 9

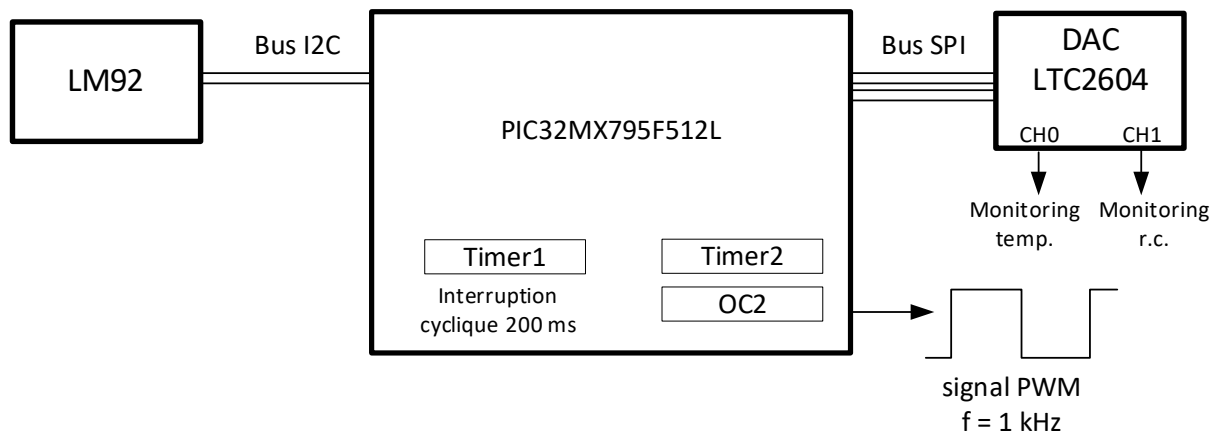
Cette révision porte sur les chapitres 5 à 9 du cours "programmation des PIC32MX".

Le principe de cet exercice est de réaliser pratiquement une application avec MPLABX & Harmony, en réalisant la préparation du projet en répondant aux questions théoriques.

SYSTEME A REALISER

Le système prévu est un convertisseur température en signal PWM, avec monitoring des valeurs de température et rapport cyclique PWM sur les sorties DAC.

Voici le schéma de principe du système demandé :



L'interruption du timer 1, cycle 200 ms, active l'application toutes les 1000 ms.

Lorsque qu'elle est active, l'application effectue les actions suivantes :

- Lecture de la température du LM92.
- Il faut calculer le rapport cyclique du PWM en fonction de la température :
 - pour -20 °C, rapport cyclique = 5%
 - pour 40 °C, rapport cyclique = 95%.
- Monitoring de la température sur le DAC canal 0 : 10 deg <=> 1 V
 - à - 20 °C, on doit obtenir -2 V
 - à 40 °C, 4 V.
- Monitoring du rapport cyclique PWM sur le DAC canal 1 :
 - 0 à 100 % <=> 0 V à 10V.
- Monitoring sur les LED :
 - LED_0 : toggle dans interruption du timer 1 pour vérifier la période.

QUESTIONS

a) A quelles broches du PIC32MX795F512L (noms complets et no) correspondent les 4 éléments suivants :

OC2 :
SPI-SCK :
SPI-MOSI :
SPI-CS_DA :

b) Quelle doit être la configuration du timer 1 pour obtenir une interruption cyclique de 200 ms (période et prescaler) ?

Prescaler minimum :
.....
Choix de prescaler :
.....
Période :
.....

c) Quelle doit être la configuration du timer 2 pour obtenir un signal PWM d'une fréquence de 1 kHz (période et prescaler) ?

Prescaler minimum :
.....
Choix de prescaler :
.....
Période :
.....

d) Pour la configuration de l'OC2 avec le MHC, quelles valeurs devez-vous donner aux éléments suivants (avec timer 2 et un PWM à 50 %) ?

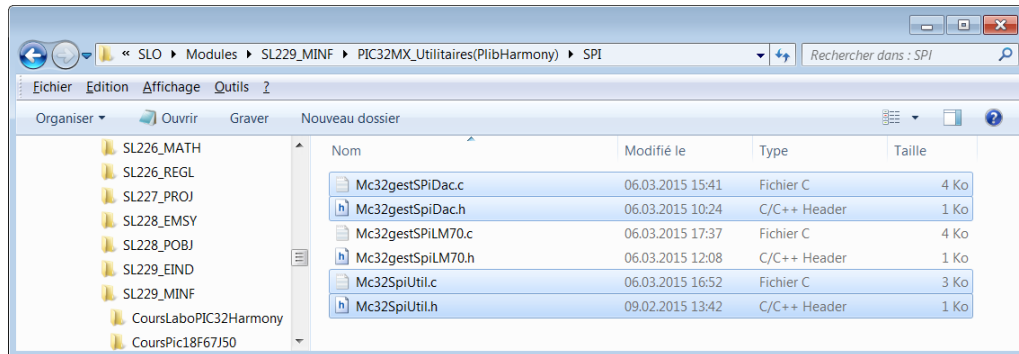
OC module ID :
OC Mode :
OC Buffer Size :
OC Timebase Timer Selection :
OC Pulse Width :

e) En supposant que vous avez oublié de cocher l'utilisation d'une interruption pour le timer 1, que devez-vous ajouter comme instructions ?

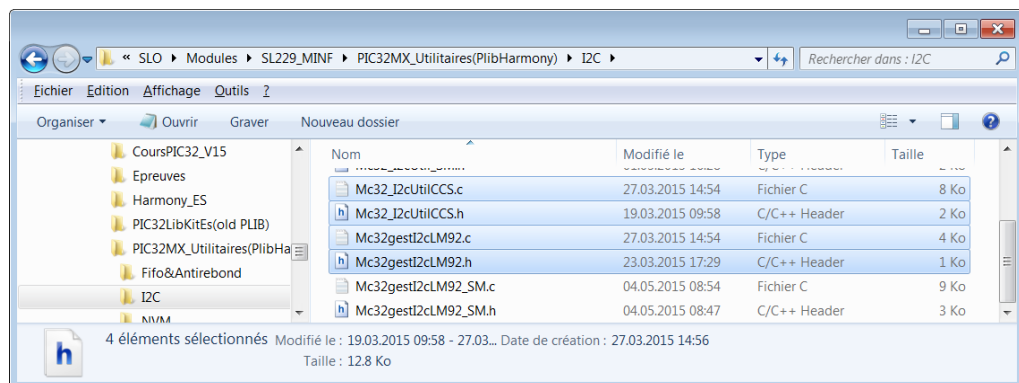
.....
.....
.....
.....
.....
.....
.....

REALISATION PRATIQUE

- Création d'un projet avec Harmony pour le starter-kit PIC32MX.
- Utilisation du timer 1 (driver statique) avec interruption, prendre niveau 3.
- Utilisation du timer 2 (driver statique) sans interruption.
- Utilisation de l'OC2 (driver statique), en mode PWM simple.
- Ajoutez les fichiers pour le SPI :



- Ajoutez les fichiers pour l'I2C :



MODIFICATION FICHIER SYSTEM_INTERRUPT.C

Compléter l'ISR du timer 1 avec l'inversion de la LED0, ainsi qu'un comptage de 5 cycles. Utilisez la fonction BSP_LEDToggle. Ajouter l'appel à la fonction APP_UpdateState.

MODIFICATION FICHIER APP.C

- Effectuer l'initialisation de l'afficheur LCD et afficher (section case INIT) :

Ex Rev chap 5 a 9
<Nom>

- Initialisation du I2C (LM92) et du SPI (LTC 2604)
- Il faut aussi appeler les fonctions pour lancer les timers et activer l'OC2.
- Dans la section case SERVICE_TASKS :
 - LED_1 ON
 - Lecture température du LM92
 - Affichage "Temp = ttt.t" sur ligne 3.
 - Calcul rapport cyclique PWM et action sur OC.
 - Affichage "r.c. = rr" sur ligne 4.
 - Calcul valeurs monitoring et envoi au DAC des 2 canaux.
 - LED_1 OFF
- Il faut aussi réaliser la classique fonction APP_UpdateState.

TEST DE FONCTIONNEMENT ET OBSERVATIONS

1^{ère} mesure :

Mesurez à l'oscilloscope la période du timer 1 qui doit être de 200 ms. Vérifiez que la période du signal sur la LED_1 est bien de 1000 ms.

2^{ème} mesure :

En utilisant le signal sur la LED_1, mesurer la durée du traitement de l'application.

En observant les 3 signaux du SPI (/CS, SCK, MOSI) indiquez quel est le temps de transmission des 2 valeurs de monitoring.

3^{ème} mesure :

Mesurez à l'oscilloscope le signal en OC2 et les 2 sorties du DAC.

Obtenez une copie d'oscilloscope pour une température approximative de -20, 10, 28 et 40 degrés. Notez la température affichée au moment où vous effectuez la copie de l'écran de l'oscilloscope.