TP I2C

Objectifs : Mettre en œuvre un composant communiquant sur bus I2C, compléter un programme en C18 utilisant la bibliothèque I2C.h

A partir de :
cours bus I2C
MPLAB + C18 + ISIS
Schéma ISIS pour la simulation : TP_MCP3421.DSN
Projet MPLAB : TP_MCP3421.mcp
Documentation C18 et bibliotheque Microchip I2C.h
datasheet : LTC2631 et MCP3421
Information sonde PT100 :
http://fr.wikipedia.org/wiki/Thermom%C3%A8tre_%C3%A0_r%C3%A9sistance_de_platine
Doc bibliothèque I2C.h : C:\mcc18\doc\periph-lib\i2c.htm

Rédiger un compte rendu

La fonction init_UART ne sera pas étudiée, elle configure les communications asynchrones du PIC afin de visualiser sur un terminal les messages ASCII transmis par le PIC.
La fonction init_horloge_interne , place la PIC en mode horloge interne Fosc=32MHz , Fcy=8MHz

Durée 6h

Préparation :

1) Décrivez la condition de début de communication : « START CONDITION »
2) Décrivez la condition de fin de communication : « STOP CONDITION »
3) Quelle est la fréquence maximum de transfert du bus I2C ?
4) Quelle est la fonction et la signification de SDA, de SCL ?
5) Pourquoi le bus I2C est dit « SYNCHRONE » ? Qui génère l'horloge de communication ?
6) Qu'est-ce qu'un bit récessif ? Un bit dominant ? Quels sont ces bits sur un bus I2C ?
7) Pourquoi faut-il placer des résistances de pull up sur SCL et SDA

TP :
Partie 1 : étude des communications entre un PIC18F2620 et ADC MCP3421

1) Quelle est la fonction du MCP3421 ?
2) Quelle est la valeur de sa tension de référence ?
3) Quel est son quantum en mode 16 bits ?
4) On désire fonctionner en mode « One-Shot » 16 bits, avec un gain de 4V/V indiquer en binaire la valeur à placer dans le registre « CONFIGURATION REGISTER »
5) Quelle est l'adresse I2C de ce composant en écriture et en lecture (en binaire et hexadécimal)
6) Le modèle mathématique retenu pour la sonde PT100 est celui du deuxième ordre : donnezr l'équation reliant Rt à la température. Tracer RT = f(t°) entre 0° et 100°C.
7) Donner l'équation reliant la tension Vin+ - Vin- à la température.
8) En déduire l'équation reliant le nombre N produit par l'ADC en fonction de la température.
9) En déduite t°=f(N)

Tester le programme TP_MCP3421.C. Vérifier sur l'analyseur I2C et l'oscilloscope le déroulement des échanges I2C entre le microcontrôleur et l'ADC.

10) Tracer l'algorigramme du programme principal.
11) Quels sont les rôles des fonctions :
    a. OpenI2C(MASTER, SLEW_ON);
    b. SOC
    c. lit_MCP3421
    d. calcTemp

12) Montrer en relation avec le datasheet que la fonction **SOC** lance une conversion unique sur 16bits avec PGA 4V/V. Vérifier sur la data sheet la séquencement des fonctions de pilotage du bus I2C.
    Pourquoi la fonction AckI2C() ; n'est pas appelée ?
13) Montrer en relation avec le datasheet que la fonction **litMPC3421** effectue une lecture de l'ADC. Vérifier sur le data sheet puis sur l'analyseur de protocoles le séquencement des fonctions de pilotage du bus I2C.

Pourquoi est il maintenant nécessaire d'appeler la fonction AckI2C() ;

14) Vérifier que la ligne **Rt=(n*q)/(iPT100*gain);** renvoie bien la valeur en Ohms de la PT100.
15) Vérifier que la fonction **clacTemp** renvoie bien la température en degré centigrades. Pourquoi la variable temp doit-elle être de type float ?

Partie 2 : Réalisation d'un programme gérant un DAC LTC2631-HZ12
Travail à réaliser :

On se propose maintenant à partir de l'étude précédente de réaliser un programme de gestion d'un DAC (Digital to Analog Converter) LTC2631.
Shéma TP_LTC2631.DSN
Programme à compléter : TP_LTC2631_a_completer.c

16) Quelle est la tension de référence interne du DAC LTC2631-HZ12?
17) Quelle est le nombre de bits du convertisseur ?
18) Quel est son quantum ?
19) Compléter le programme **TP_LTC2631_a_completer.c** générant des rampes de tensions de 0v à 4,095 V
20) Vérifier les communications sur l'analyseur de protocoles, mesurer le quantum, les tensions min et max en sortie à l'aide de l'oscilloscope.
21) **Bonus** : proposer en vous aidant d'un tableau limité à une dizaine de valeurs environ un programme générant une fonction sinusoïdale sur ce principe

Déclaration d'un tableau de valeurs correspondantes à une sinusoïde (exemple)

```
unsigned int tabSin[ ]={0,200,300,400,500,1000,3000…..} ;
```

Envoie des valeurs de ce tableau vers le DAC

```
ecritLTC2631(writeLTC2632,tabSin[i++]);  // genere une sinusoide
if (i>taille_du_tableau) i=0;
```