

INFO2 : INFORMATIQUE EMBARQUEE

TP3 : Mise en œuvre du CAN

1 Objectifs

Il s'agit de mettre en œuvre le Convertisseur Analogique-Numérique du PIC18F4520 dans deux situations :

- en convertissant la tension aux bornes d'un potentiomètre dont le curseur est connecté à la broche RA3 du PIC ;
- en convertissant la tension issue d'un capteur de température LM35.

La configuration des registres contrôlant le CAN se fera d'abord en les écrivant directement puis en utilisant les fonctions de la bibliothèque dont le fichier d'en-tête est `adc.h`.

2 Projet TP3A : conversion d'une tension aux bornes d'un potentiomètre

2.1 Cahier des charges

Un signal analogique entre 0V et 5 V créé par le potentiomètre P1 est connecté à l'entrée RA3 du PIC18F4520.

On souhaite, le plus fréquemment possible, numériser ce signal sur 10 bits, puis, quand le résultat de la conversion est disponible, visualiser les 2 bits de poids fort du résultat sur les 2 leds connectées aux broches RC1 et RC0 et les 8 bits de poids faible du résultat sur les 8 leds connectées aux broches RD7 à RD0.

La disponibilité du résultat de la conversion sera détectée en scrutant (*polling*) le bit témoignant de la fin de conversion, pas par un mécanisme d'interruption.

Attention : sur le port C, seul RC1 et RC0 seront déclarées en sortie (pas tout le port) afin de ne pas interférer avec d'autres broches de ce port connectées au port série.

- Sur la carte EasyPic v7, repérez le potentiomètre P1. Vérifiez qu'un cavalier met en contact le curseur de P1 avec RA3.
- Quel canal ANx correspond à la broche RA3 ?
- Demander à l'enseignant de vérifier que RA3 n'est chargé ni par des leds ni par une validation d'affichage 7 segments.

2.2 Mise en œuvre et validation

- Ecrivez votre programme. Le résultat de la conversion étant sur 10 bits, il sera stocké dans une variable 16 bits signée. Ensuite, on « répartira » comme il faut cette variable sur les leds du port D et du port C.
- Validez-le. Notamment, relevez la valeur affichée sur les leds quand le potentiomètre est tourné au minimum puis au maximum. Vérifiez la cohérence de la progression des leds quand on passe du minimum au maximum du potentiomètre et dites si la **monotonie** du CAN est respectée (*i.e.* quand la tension analogique augmente, le résultat numérique augmente aussi sans jamais redescendre).

3 Projet TP3B : visualisation d'une température à l'aide du capteur LM35 et du CAN

3.1 Cahier des charges

Le capteur LM35 convertit une température (°C) en une tension (Volt).

La sortie de ce capteur est connectée à l'entrée RE2 du PIC18F4520.

On souhaite, le plus fréquemment possible, numériser ce signal sur 10 bits, puis, quand le résultat de la conversion est disponible, visualiser les 2 bits de poids fort du résultat sur les 2 leds connectées aux broches RC1 et RC0 et les 8 bits de poids faible du résultat sur les 8 leds connectées aux broches RD7 à RD0.

La disponibilité du résultat de la conversion sera détectée en scrutant (*polling*) le bit témoignant de la fin de conversion, pas par un mécanisme d'interruption.

Attention : sur le port C, seul RC1 et RC0 seront déclarées en sortie (pas tout le port) afin de ne pas interférer avec d'autres broches de ce port connectées au port série.

- Dans le répertoire **Ressources**, ouvrez la documentation du capteur LM35 et relevez son coefficient de proportionnalité. Relevez également sa précision (*i.e.* à combien de degrés près une température est annoncée par le capteur). Cette précision du capteur est-elle compatible avec le *quantum* du CAN utilisé ? En d'autres termes, la chaîne de mesure est-elle cohérente ?
- Sur la carte EasyPic v7, au niveau de J25, vérifiez qu'un cavalier jaune relie la broche du milieu du capteur LM35 en boîtier TO92 à la connexion marquée « RE2 ». Cela permet de relier la sortie du capteur vers la broche RE2 du PIC18F4520.
- Quel canal ANx correspond à la broche RE2 ?

3.2 Mise en œuvre et validation

- Ecrivez votre programme.
- Validez-le. Notamment, relevez la valeur affichée sur les leds et dites à quelle température ambiante cela correspond. Est-ce cohérent ?

NE PAS TOUCHER LE CAPTEUR POUR LE CHAUFFER ET FAIRE DES TESTS !

4 Utilisation des fonctions de la bibliothèque adc.h

Dans un but de meilleure lisibilité du code, nous allons réécrire le programme **TP3A.c** à l'aide des fonctions de la bibliothèque **adc.h**. Les lignes qui ne serviront plus seront mises en commentaire.

- Ouvrez **MPLAB_C18_Libraries_51297f.pdf**, la documentation des fonctions des bibliothèques du compilateur C18. Trouvez la partie relative à la conversion analogique-numérique : 2.2 A/D CONVERTER FUNCTIONS.
- Vos 4 lignes d'initialisation du CAN peuvent être remplacées par l'usage de la fonction **OpenADC ()** décrite pages 18/184 et 19/184 (« All Other Processors »). Pour le choix de la constante de « A/D voltage configuration », on prendra **ADC_REF_VDD_VSS** (bizarrement oubliée dans la documentation) qui permet de numériser une tension entre $V_{SS} = 0\text{ V}$ et $V_{DD} = 5\text{ V}$.
- Ensuite, dans la boucle infinie, en utilisant les fonctions de la bibliothèque : réécrivez l'ordre de départ de la conversion, l'attente de fin de conversion, l'affectation de la variable signée 16 bits par le résultat de la conversion. Affectez les leds correctement.
- Testez votre programme.