



# TP n°2bis

## Convertisseur analogique/numérique



### Objectifs :

**Savoir utiliser les entrées analogiques du PIC.**

---

### 1/ Le convertisseur analogique/numérique

**On s'interdit dans ce tp d'utiliser les fonctions `adc_on()` et `read_adc_8b()` de biosdem.**

On souhaite simplement afficher la valeur résultant de la conversion de la tension présente sur AN0 sur l'afficheur lcd.

- *Configuration des broches en mode analogique/logique/référence*

En analysant le schéma de la carte biosdem, déterminer quelle est la seule combinaison possible des bits PCFG3:PCFG0 qui permette une utilisation simultanée du module lcd et l'entrée analogique AN0.

- *Choix de l'horloge de conversion*

Sachant que l'oscillateur de la carte picdem2+ est un quartz de 4Mhz, indiquer les deux choix possibles pour  $T_{AD}$ .

Retrouver par un calcul simple la valeur de 5Mhz pour la fréquence maximum lorsque  $T_{AD} = 8T_{OSC}$ .

- *Choix de la présentation des résultats dans ADRESH et ADRESL*

Si on souhaite ne conserver que les 8 bits de poids forts quel est le mode le plus pratique. Quel bit de quel registre SFR permet cette configuration ?

☺ En suivant la méthode suggérée par la documentation du PIC, afficher en permanence en décimal et en hexadécimal sur le lcd les 8 bits de poids forts de la valeur obtenue lors de la conversion de l'entrée AN0.

☺ Ecrire une fonction `adc_init_8b()` qui regroupe la séquence d'initialisation du module convertisseur et une fonction `adc_read_8b1()` qui retourne les 8 bits de poids forts du résultat de la conversion d'un canal quelconque (0-7) passé en paramètre. (Sur la carte picdem cette fonction ne pourra pour l'instant être utilisée qu'avec la valeur 0 en paramètre).

☺ Ecrire une fonction `adc_init_10b()` qui regroupe la séquence d'initialisation du module convertisseur et une fonction `adc_read_10b()` qui retourne les 10 bits du résultat de la conversion d'un canal quelconque (0-7) passé en paramètre. (Sur la carte picdem cette fonction ne pourra pour l'instant être utilisée qu'avec la valeur 0 en paramètre).

# TP n°2bis

## 2/ Voltmètre numérique

☺ En utilisant la fonction `adc_read_8b1()`, afficher la tension en Volt présente en RA0 et la masse sur l'écran lcd. Pour cela il suffit d'inverser l'équation  $N = 255.V/5$ . Les nombres réels en sont pas nécessaires.

☺ En utilisant la fonction `adc_read_10b()`, afficher la tension en Volt comme précédemment mais avec une meilleure résolution. Dans ce cas les calculs en nombres réels sont bien utiles mais pour l'affichage il faudra mettre la partie entière et la partie décimale dans des variables de type `int` car la fonction `lcd_printf()` ne supporte pas le format `%f`.

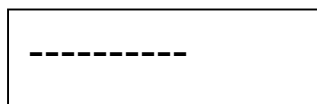
## 3/ Affichage " BarGraph "

☺ Afficher un "bargraph" composé de caractères '-' évoluant en fonction de la position du potentiomètre R16 sur la première ligne de l'écran lcd.

Pour avoir un bon rendu de l'affichage, il ne faut pas réafficher tous les caractères à chaque tour de boucle mais il faut ajouter ou enlever des caractères '-' en fonction de ce qui était affiché au tour de boucle précédent.

Ex :

pour passer de :

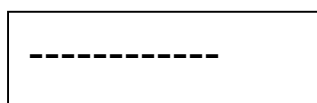


à

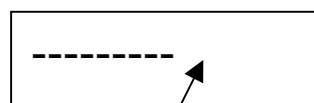


On ajoute trois -

pour passer de :

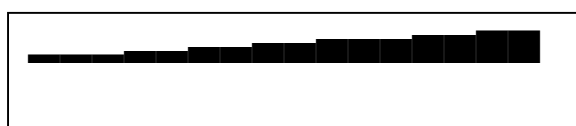


à



On ajoute trois espace

☺ Améliorer l'affichage en affichant le bargraph sous la forme ci-dessous :



█ : 0x80 , █ : 0x81 , █ : 0x82 , █ : 0x83 , █ : 0x84 , █ : 0x85 , █ : 0x86.