TP3 Microcontrôleur

# Partie 1 : Récupérer la valeur d’un ADC

* Use of A/D converter and display a binary encoded value using LEDs
* Play with the accuracy of A/D conversion accuracy (reference voltage) and resolution (nb. of bits of output word)

Récupérée sur RA0 reliée avec un potentiomètre

On met une interruption sur la modification de la valeur de RA0, on déclare RA0 en analog input, on initialise le PORTC en sortie, la routine d’interruption envoie la valeur de RA0 dans le PORTC et dans la boucle principale. On peut également faire des décalages à droite pour que la valeur de l’ADC soit plus compréhensible.

Résultat : les leds du PORTC affiche la valeur binaire du courant traversant le potentiometre AO

# Partie 2 : Afficher la valeur sur un 7-segments

* Play with the accuracy of A/D conversion accuracy (reference voltage) and resolution (nb. of bits of output word)
* Display a number on a 7-SEG display

On doit lire la valeur de l’ADC RB0 car RA0 est utilisé pour le 7-segments. Il est nécessaire d’affiner la valeur binaire de la lecture pour un affichage sur le 7-segments. Il est nécessaire de jouer sur la résolution de la mesure pour éviter que le nombre soit supérieur au nombre max affichable par le 7-segment. Pour le 7-segments les bits 0, 1, 2 et 3 du PORTA permet de sélectionner sur quel digit on veut écrire. Les PORTD de 0 à 7 permettent de choisir la valeur affichée sur le digit : chaque sortie correspond à un des segments.

# Partie 3 : Faire une clock avec le 7-segments

* Display a number on a 7-SEG display
* Real time clock implementation and display on a four-7-SEG display.

Pour cela on active des interruptions sur la clock, on déclare le PORTD comme sortie et chaque interruption de la clock va incrémenter la valeur de PORTD. Ce qui fait que naturellement, comme PORTD est lié à l’afficheur 7-segments on voit que la valeur prise par le 7-segment change à chaque interruption de clock.