# TP 4 : Utilisation du convertisseur analogique numérique du 9s12

Objectifs :
- Apprendre à utiliser un sous-ensemble du microcontrôleur en lisant la documentation constructeur.
- Construire un programme C de démonstration des fonctionnalités offertes.

## Introduction

Dans ce TP, vous mettrez en oeuvre le convertisseur A/N intégré du 9s12, afin de comprendre comment on peut récupérer et traiter des informations de nature analogique. En préalable, il est indispensable de prendre le temps de lire la documentation annexe (poly de référence) qui décrit l'utilisation de ces convertisseurs.

## Configuration du convertisseur

Nous ferons travailler le convertisseur en mode "8 bits". Il sera en mode "séquence de conversion unique", et à chaque fois, il faudra par programme relancer une nouvelle conversion, et attendre qu'elle se termine avant de lire le résultat. Nous utiliserons "l'alignement à droite", permettant d'avoir la valeur convertie dans le poids faible du registre résultat, en représentation non-signée.

# 1 Préparation

Sur la carte de TP, le potentiomètre ajustable RT1 est connecté sur une des entrées de l'un des 2 convertisseurs. En recherchant sur le schéma de la carte de TP (poly.), répondez aux questions suivantes :

1. Sur quelle entrée : _____ Sur lequel des 2 convertisseurs : 0 - 1
2. Par conséquent, dans quel registre faudra-t-il venir lire le résultat de la conversion : _____
3. Lisez attentivement la documentation fournie (poly.), et déterminez les valeurs à écrire dans les registres de configuration du convertisseur (en cas de doute, mettre la valeur par défaut indiquée dans la doc).

ATD_CTL2

| ADPU | AFFC | AWAI | ETRIGLE | ETRIGP | ETRIGE | ASCIE | ASCIF |
|---|---|---|---|---|---|---|---|
| | | | | | | | |

ATD_CTL3

| 0 | S8C | S4C | S2C | S1C | FIFO | FRZ1 | FRZ0 |
|---|---|---|---|---|---|---|---|
| | | | | | | | |

ATD_CTL4

| SRES8 | SMP1 | SMP0 | PRS4 | PRS3 | PRS2 | PRS1 | PRS0 |
|---|---|---|---|---|---|---|---|
| | | | | | | | |

ATD_CTL5

| DJM | DSGN | SCAN | MULT | 0 | CC | CB | CA |
|---|---|---|---|---|---|---|---|
| | | | | | | | |

# 2 Travail à effectuer

## 2.1 Programme 1

Dans un 1er temps, on souhaite simplement visualiser sur la barre de dels (Port B) la valeur binaire lue sur le convertisseur. Comme nous disposons de 8 dels, nous ferons travailler le convertisseur en mode "8 bits". Il sera en mode "séquence de conversion unique", et à chaque boucle, il faudra relancer une nouvelle conversion, et attendre qu'elle se termine avant de lire le résultat et d'afficher la valeur sur les Dels. Le programme aura l'architecture suivante. Le saisir et l'enregistrer dans un fichier **tp4_1.c**, en utilisant les valeurs déterminées lors de la préparation.

```
1  int main()
2  {
3      char a;
4  // A - initialisations
5      DDRB    = 0xff;
6      ATDXCTL2 = ...;
7      ATDXCTL3 = ...;
8      ATDXCTL4 = ...;
9  // B - conversion et affichage en boucle
10     while( 1 ) {
11         ATDXCTL5 = ..     // 1 - lancement d'une nouvelle conversion
12         while(  .....  ) // 2 - attente de la fin de conversion
13             ;
14         a =  .....  ;     // 3 - lecture dans le registre -résultat
15         PORTB = ~a;       // 4 - écriture sur le Port B (complémentée)
16     }
17 }
```

Note : l'attente de la fin de la conversion sera réalisée par une structure 'while()', en testant le bit signalant la fin de la séquence de conversion.

## 2.2 Programme 2

On souhaite maintenant un programme qui "balade" une del allumée de gauche à droite selon la position du potentiomètre. A cet effet, on prévoira un tableau de `char`[1], initialisé avec les 8 valeurs correspondant aux différentes positions des dels :

```
char tab[] = { 0x80, 0x40, ... };
```

Il faudra ensuite calculer un indice de tableau (de type 'char' ou 'int') fonction de la valeur convertie, puis envoyer sur le Port B la valeur du tableau correspondante (sans oublier de faire l'opération "complément à 1") :

```
PORTB = ~ tab[indice];
```

La variable `indice` est calculée à partir de la valeur convertie : la plage de valeurs $[0-255]$ doit être ramenée sur l'intervalle $[0-7]$ (8 valeurs).

1. Proposer une expression donnant l'`indice` du tableau en fonction de la valeur convertie n.
2. Sauvegarder le fichier précédent en **tp4_2.c**, et le modifier pour répondre à ce cahier des charges. La structure générale du programme sera conservée.

## 2.3 Programme 3

On souhaite réaliser la gestion d'un système de chauffage. Le système sera doté d'une sonde de température, simulée ici par le potentiomètre, et de deux résistances de chauffage, R1 et R2, simulées ici par les dels PB1 et PB2. Le fonctionnement est décrit par le tableau ci-dessous :

| Température | R1 | R2 | Rem. |
|---|---|---|---|
| T $\leq$ 10 ° | Allumée | Allumée | Pleine puissance |
| 10°C T $\leq$ 20 °C | Allumée | Eteinte | Demi puissance |
| 20 °C $\leq$ T | Eteinte | Eteinte | |

Le capteur de température est linéaire, son électronique de traitement associée permet d'obtenir les relations suivantes :
T = 0 °C $\Rightarrow$ Vs = 0 V
T = 20° C $\Rightarrow$ Vs = 4 V

On considère par ailleurs que le convertisseur est étalonné sur le tableau suivant :

| Ve | N |
|---|---|
| 0 V | $00 |
| 5V | $ff |

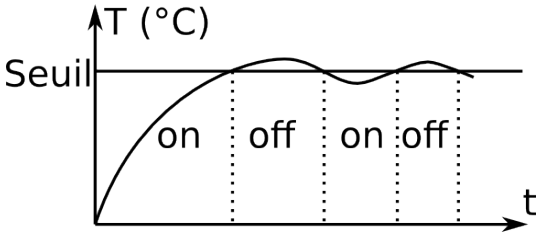
1. le type `char` correspond à un octet en C.

1. Calculer les valeurs numériques obtenues pour les 2 seuils à considérer :

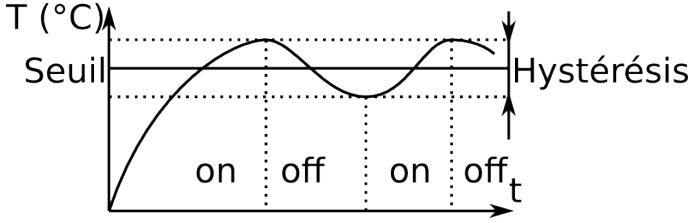
| T (°C) | N |
|---|---|
| 10 | |
| 20 | |

2. Sauvegarder le fichier précédent en **tp4_3.c**, et y définir en tête du fichier 2 constantes `SEUIL_10` et `SEUIL_20` (avec la directive `#define`) correspondant à ces 2 valeurs.
3. Implémenter le cahier des charges décrit, et le tester.

Ce type de régulation est en pratique inadéquat : dès l'arrêt du chauffage, la température redescend, et dès qu'on franchit à nouveau le seuil, la résistance se réactive, et ainsi de suite : il y a **instabilité**.




Pour éviter ce problème, il est indispensable d'ajouter un **hystérésis** sur les seuils de commutation. Ceci se fait simplement en effectuant deux tests, un pour déterminer si on est au dessus du seuil haut, un autre pour déterminer si on est au dessous du seuil bas. Ces deux seuils étant définis comme la valeur nominale à laquelle on ajoute ou on retranche un delta, faible devant la valeur nominale.




4. Modifier le programme en introduisant un hystérésis de 2 degrés (+/- 1°C) sur les seuils. On définira dans l'en-tête une constante `DELTA` avec la valeur adéquate.