TD n°2 Mise en oeuvre du CAN du AT90S8535

1- Sans interruption
2- Avec interruption

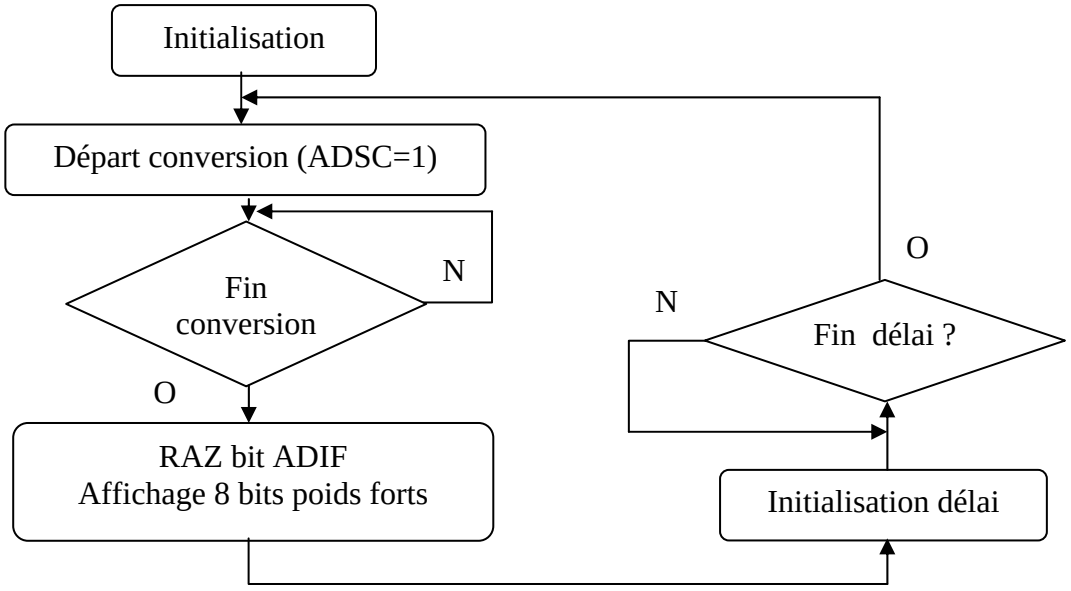
Objectifs :

- Configurer les registres d'initialisation du CAN pour l'acquisition d'une grandeur analogique sur l'entrée ADC0. Régler la fréquence d'acquisition à F/64 et interruption inhibée.
- Réaliser un programme en langage assembleur qui effectue la conversion toutes les « delai_ms » et affiche sur des leds les huit bits de poids forts.
- Même chose en mettant en œuvre l'interruption associée au CAN.

1) Utilisation sans interruption :

Voie ADC0 => ADMUX = 0
F/64, pas d'IT, monocoup => ADCSR = $86 (ADEN=1 , ADSC=0, ADFR=0, ADIF=X (0 ou 1), ADIE=0, ADPSi=110)

⇨ 1000 0110 = $86

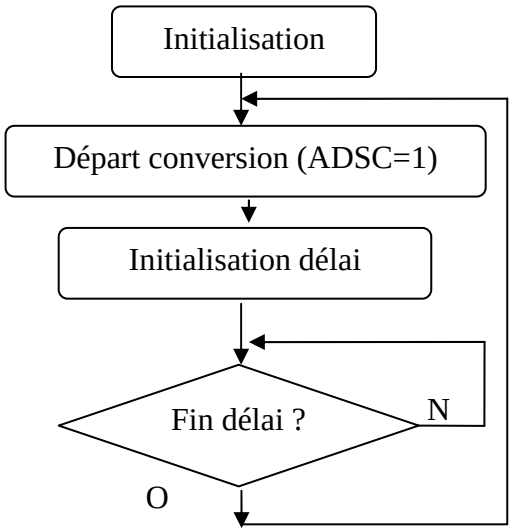


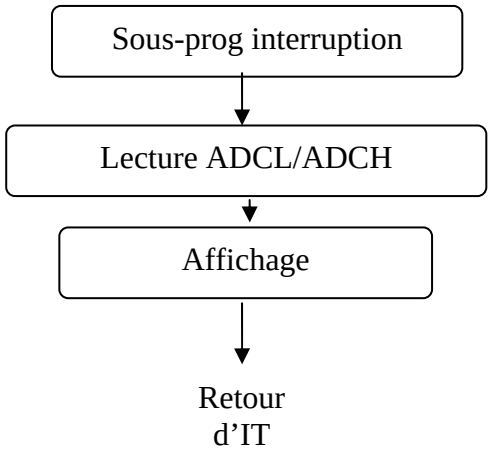
Voir programme adc.asm

2) Avec interruption

On ne teste plus la fin de la conversion (bit ADIF). C'est le convertisseur qui prévient le µC par une interruption.

Initialisation de ADCSR => ADCSR=$8E (ADIE=1)





Voir programme AD_IT.asm

```
.include "8535def.inc"
.def temp = r16
.def  r_conv_l = r17
.def r_conv_h = r18
;------------------------CAN sans utilisation IT-----------
.cseg
.org $000
       rjmp init                      ;Reset interrupt
-------------------initialisation-----------------------
.org $012
init:
       ldi     temp,low (RAMEND)
       out     SPL,temp        ; init pointeur pile poids faibles
       ldi     temp,high (RAMEND)
       out     SPH,temp        ; init pointeur pile poids forts
       ser     temp
       out     DDRB,temp       ; port B en sortie
       out     portb,temp      ; raz leds
       clr     temp
       out     ddra,temp       ; port A en entrée
       ldi     temp, 0x86      ; div 64 et can activé
       out     adcsr,temp
       clr     temp
       out     admux,temp      ; select voie 0 du CAN
; ------------------------conversion a=>n----------------
convert:
       sbi     adcsr,adsc      ; lance conversion
at_fin:
       sbis    adcsr,adif      ; lit bit fin_conv
       rjmp    at_fin          ; attente si pas fini
       in      r_conv_l, adcl  ; lit poids faibles
       in      r_conv_h, adch  ; lit poids forts
       sbi     adcsr,adif      ; raz drapeau fin_conv
       ror     r_conv_h
       ror     r_conv_l        ; conservation  des
       ror     r_conv_h        ; 8 bits de poids
       ror     r_conv_l        ; forts dans r_conv_l
       com     r_conv_l        ; complémente pour affichage leds
       out     portb, r_conv_l
       rcall   delai           ; tempo avant autre conversion
       rjmp    convert
;---------------boucle d'attente-------------------
delai:
       ldi     r21,$08
       ldi     r20,$FF
       ldi     r19,$FF
delai3:
       dec     r19
       brne    delai3
       dec     r20
       brne    delai3
       dec     r21
       brne    delai3
       ret
```

```
.include "8535def.inc"
.def temp = r16
.def  r_conv_l = r17
.def r_conv_h = r18
;----------------------------CAN avec IT----------------
.cseg
.org $000
       rjmp    init           ;Reset interrupt
.org $00E
       rjmp    adc_it         ;adresse it_adc
.org $012
       ;-------------------initialisation---------------------
init:
       ldi     temp,low (RAMEND)
       out     SPL,temp       ; init pointeur pile poids faibles
       ldi     temp,high (RAMEND)
       out     SPH,temp       ; init pointeur pile poids forts
       ser     temp
       out     DDRB,temp      ; port B en sortie
       out     portb,temp     ; raz leds
       clr     temp
       out     admux,temp     ; select voie 0 du CAN
       out     ddra,temp      ; port A en entrée
       ldi     temp, 0x8E     ; div 64, can et IT activés
       out     adcsr,temp
       sei                                    ;autorise IT
; ------------------------conversion a=>n-------------------
convert:
       sbi     adcsr,adsc     ; lance conversion
       rcall   delai          ; tempo avant autre conversion
       rjmp convert
;-------------------------------------------------------------
;---------------boucle d'attente-------------------
delai:
       ldi     r21,$08
       ldi     r20,$FF
       ldi     r19,$FF
delai3:
       dec     r19
       brne    delai3
       dec     r20
       brne    delai3
       dec     r21
       brne    delai3
       ret
;-----------------------------------------------------
;----------------------Sous-prog interruption-----------
adc_it:
       in      temp, sreg     ;sauvegarde de SREG
       push    temp
       in      r_conv_l, adcl ; lit poids faibles
       in      r_conv_h, adch ; lit poids forts
       ror     r_conv_h
       ror     r_conv_l       ; conservation  des
       ror     r_conv_h       ; 8 bits de poids
       ror     r_conv_l       ; forts dans r_conv_l
       com     r_conv_l       ; complémente pour affichage leds
       out     portb, r_conv_l
       pop     temp           ;récup SREG
       out      sreg, temp
       reti
```