

Voltmètre numérique

Le CAN du 68HC11

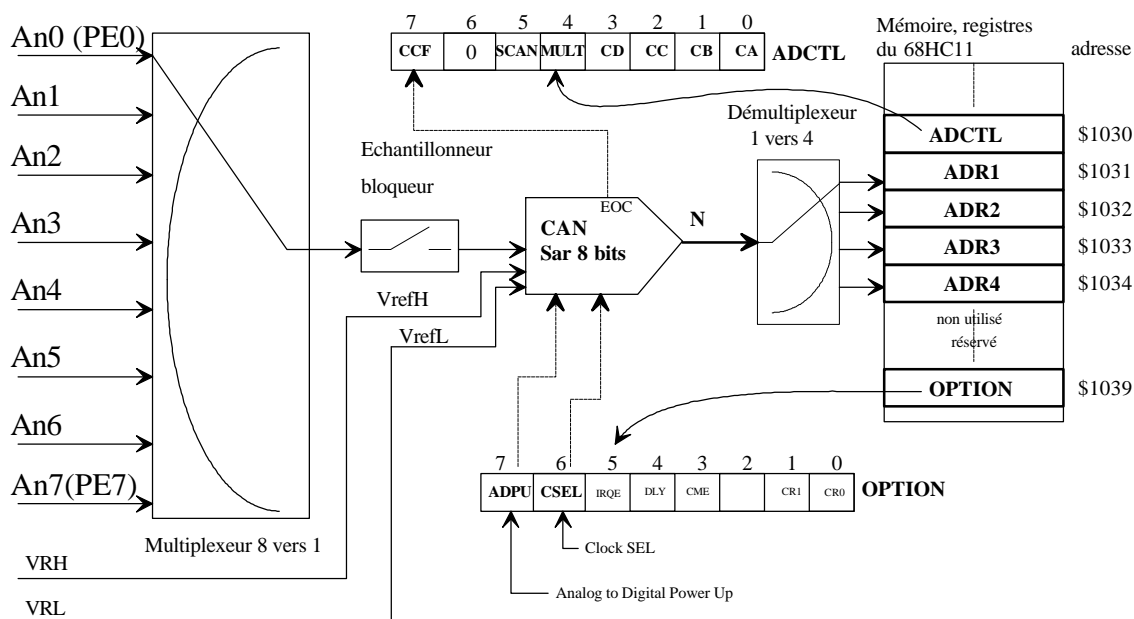
Mr COTTET

Présentation:

On désire mettre en oeuvre le convertisseur analogique numérique interne au 68HC11, pour réaliser un voltmètre numérique. On utilisera le simulateur UMPS en ajoutant la ressource nommée "A/D Slider" correspondant à un potentiomètre associé au convertisseur du HC11 (le potentiomètre permet d'imposer sur une entrée du HC11 une tension comprise entre V_{RL} et V_{RH} , assurant une valeur numérique entre \$00 et \$FF après conversion).

I) Rappel du convertisseur (CAN) interne au 68HC11:

Le convertisseur analogique numérique du HC11 correspond au schéma de principe suivant. Un CAN 8 bits à approximations successives est connecté au port E à l'aide d'un multiplexeur analogique 8 vers 1. Le résultat de la conversion est transmis par un démultiplexeur 1 vers 4 aux adresses ADR1 à ADR4. Les différents modes de fonctionnement sont configurable par l'utilisateur à l'aide de 2 registres "ADCTL" et "OPTION".



- 1) En vous aidant du livre *"Les microcontrôleurs HC11 et leur programmation"* de "Christian CAZAUBON" (Chapitre convertisseur analogique numérique), ou de la documentation constructeur du HC11, indiquez le rôle des 2 bits ADPU et CSEL du registre OPTION.
- 2) Indiquez le rôle des bits SCAN et MULT du registre de contrôle du CAN "ADCTL".
- 3) Indiquez le rôle du bit CCF.
- 4) Indiquez l'intérêt de la fonction échantillonneur-bloqueur.

5) Le 68HC11 comporte 2 broches nommées VRH et VRL . Indiquez le rôle de ces broches. Donnez l'expression de la tension V_i (Voisine de V_{in} à 1 LSB) après conversion, en fonction de b_7 à b_0 (bits de données du CAN) et de VRH et VRL.

6) On appelle N la valeur numérique sur 8 bits délivrée par le convertisseur CAN.

$$N = 2^7 \cdot b_7 + 2^6 \cdot b_6 + 2^5 \cdot b_5 + 2^4 \cdot b_4 + 2^3 \cdot b_3 + 2^2 \cdot b_2 + 2^1 \cdot b_1 + 2^0 \cdot b_0$$

Faire apparaître le nombre N dans l'expression de V_i .

On suppose $VRL = V_{ss} = 0V$, et $VRH = V_{ref}$, en déduire l'expression générale de V_i en fonction de V_{ref} , N, et n le nombre de bits du convertisseur.

7) Pour toute la suite, on suppose $VRL = 0V$ et $VRH = 5V$. Déduisez-en le quantum du convertisseur (quantum= valeur analogique élémentaire correspondant à $N=1$).

8) Calculez alors la dynamique d'entrée (ou FSR = Full Scale Range). Rem: la dynamique d'entrée est la différence entre la valeur maximale et la valeur minimale de la grandeur analogique à l'entrée du convertisseur pour obtenir respectivement N_{max} puis N_{min} .

9) Calculez à l'aide du quantum, la valeur de N en décimal et en hexadécimal correspondant à une tension V_{in} de 2,578V. (Rem: On prendra 3 chiffres significatifs pour le quantum).

10) Calculez à nouveau la valeur de N si l'on arrondi le quantum à 2 chiffres significatifs. Conclure sur la précision des nombres par rapport au nombre de bits d'un convertisseur.

11) On suppose que le quartz associé au 68HC11 a une fréquence de 8MHz. Le bit CSEL du registre OPTION permet de choisir le signal E du HC11, ou un oscillateur à circuit RC interne, comme horloge du CAN. Indiquez alors en recherchant dans la documentation l'état logique de ce bit pour sélectionner E comme horloge, et donnez alors le temps de conversion d'une voie. en déduire la fréquence d'échantillonnage maximale possible (dans le cas d'une voie).

12) On suppose que l'on souhaite convertir les 8 entrées analogiques, en utilisant le signal E comme horloge. Expliquez sous forme algorithmique, la procédure à suivre (écriture ou lecture des registres OPTION et PACTL), pour recopier les valeurs converties dans 8 cases mémoires appelées Mem0 à Mem7. En déduire alors la fréquence d'échantillonnage maximale possible (on négligera dans une première approximation, le temps des instructions par rapport aux temps de conversion du convertisseur).

II) Etude d'un programme voltmètre utilisant le CAN interne au 68HC11:

On désire convertir la tension présente sur PE0 ($0V < V_{in} < 5V$), et afficher la valeur en volt sur l'afficheur LCD.

1) Indiquez le contenu des registres OPTION et PACTL, pour obtenir cette conversion. On indiquera la définition des bits de chaque registre sur la première ligne, et la valeur nécessaire pour obtenir le fonctionnement désiré sur la deuxième ligne.

7	6	5	4	3	2	1	0

OPTION

7	6	5	4	3	2	1	0

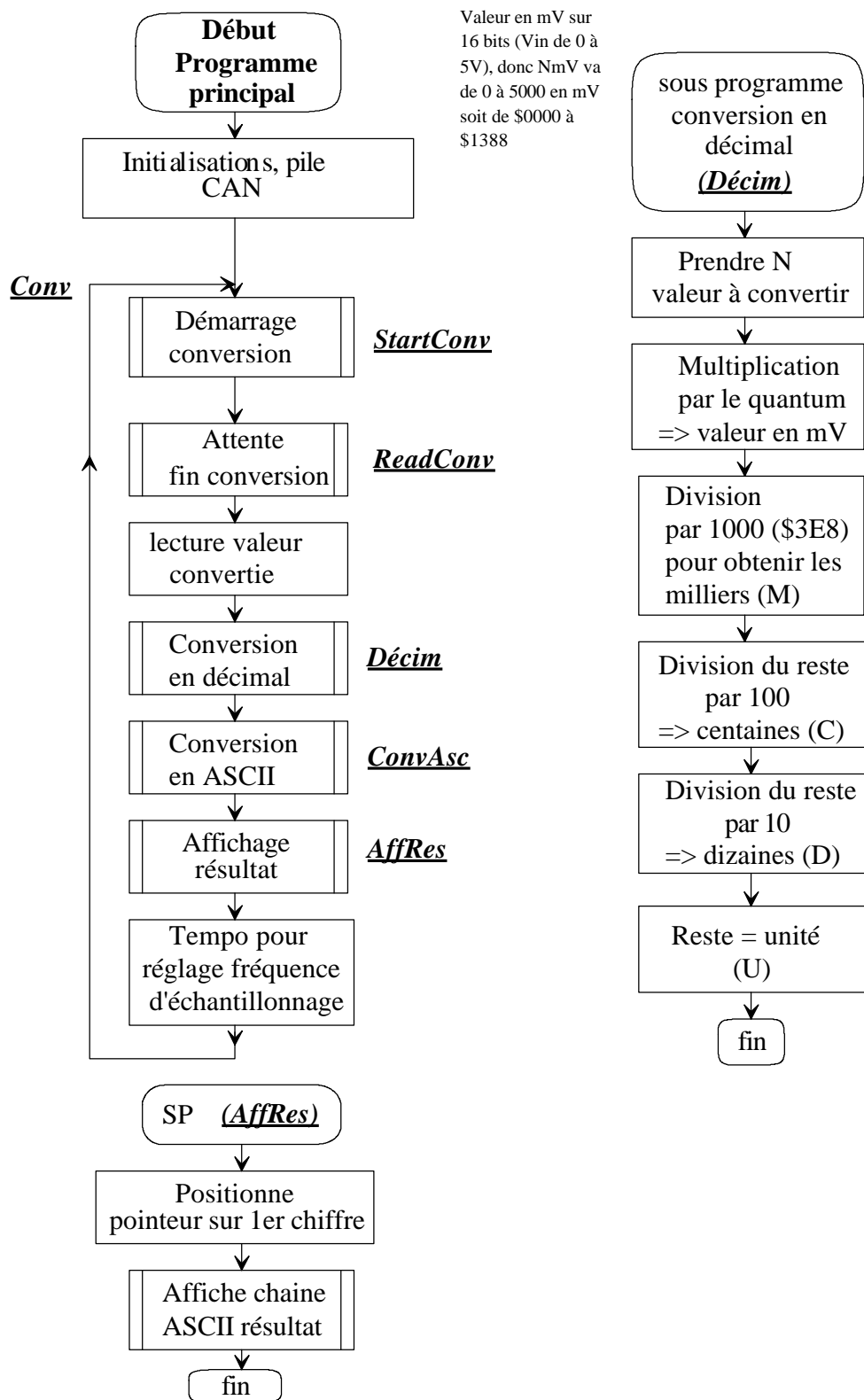
PACTL

2) Traduire en assembleur l'ordinogramme suivant. (Complétez le programme fourni).

On remarquera la présence des étiquettes lors des sauts et des noms des sous programmes, que l'on respectera.

Il sera inutile d'écrire le sous programme AffRes et la tempo pour le réglage de la fréquence d'échantillonnage.

Ordinogramme du programme Voltmètre



Programme (partiel) de gestion du CAN avec conversion en ASCII: CAN3.asm

```
;*****
;* Lycée Maurice GENEVOIX (INGRE)
;* Auteur: Mr COTTET
;* Date: 28/10/99
;* Programme CAN3.asm
;* Essai du port E lecture du convertisseur CAN
;* avec affichage de la valeur en volt
;* 00 a FF => 0.000V a +5.000V en ASCII
;* En RAM à partir de $0000
;*
;***** Déclaration des constantes *****
;*OPTION equ $1039 ;ADPU,CSEL,IRQE,DLY,CME,-,CR1,CR0
;*ADCTL equ $1030 ;CCF,-,SCAN,MULT,CD,CC,CB,CA
;*ADR1 equ ADCTL+1
;*ADR2 equ ADCTL+2
;*ADR3 equ ADCTL+3
;*ADR4 equ ADCTL+4

;***** Variables RAM *****
ORG 0
ValAscii Rmb 8 ; Résultat en ASCII
M Rmb 1 ; Milliers
C Rmb 1 ; Centaines
D Rmb 1 ; Dizaines
U Rmb 1 ; Unités (valeur en Volt)
N Rmb 1 ; Valeur du CAN en Hexa
Res0 Rmb 1 ; 4 valeurs résultats du convertisseur
Res1 Rmb 1 ; correspondant aux 4 voies AN0 ... AN3
Res2 Rmb 1
Res3 Rmb 1
VAL Rmb 2 ; Calcul sur 16 bits

;***** programme principal *****
ORG $F800

Debut lds #$00FF ; (ou $00C0 en mode Bootstrap)
jsr InitCAN

Conv jsr ; conversion de PE0 ... PE3
jsr ; Recopie des valeurs converties
ldaa ; Prend le résultat
STAA N ; Valeur en Hexa
JSR ; CONVERSION EN DECIMAL
JSR ; Affiche RESULTAT en ASCII
bra

;***** Sous programmes *****
InitCAN ldaa OPTION
oraa ;ADPU (analog to digital power UP)
anda ;CSEL=0 =>horloge interne E (sinon RC)
staa OPTION
jsr Tempo ;Tempo > 100uS pour alim CAN stable
ldaa ;SCAN=0 MULT=1 (conversion non continue)
staa ADCTL ;des 4 voies AN0 ... AN4)
rts

;***** Démarre conversion CAN *****
StartConv ldaa ;SCAN=0 MULT=1 (monocoup et 4 voies AN0 à AN3)
staa ; Relance conversion
rts

;*****
ReadConv ldaa ADCTL
;carry = CCF (1 indique fin conv)
bcc ReadConv
Ldd ADR1
std Res0
ldd ADR3
std Res2
rts

;***** CONVERSION EN DECIMAL *****
DECIM LDAA N ;N= Nombre à convertir
Ldab #39 ;multiplication par 39/2=19.5mV (quantum)
```

```

mul          ;résultat*2 en mV sur 16 bits dans D
lsrd         ;division par 2 pour le quantum de 19.5mV
CLR M        ;chiffre des milliers (en mV => V)
CLR C        ;idem centaines
CLR D        ;dizaines
CLR U        ;unités

SUI1  STD VAL      ;*** RECHERCHE DU NOMBRE EN DECIMAL, 4 CHIFFRES ***
      LDX VAL
      CPX #1000      ; Nombre de paquets de 1000
      BMI CENT
      INC M
      SUBD #1000
      BRA SUI1

CENT  STD VAL
      LDX VAL
      _____ ; Nombre de centaines
      _____
      _____
      _____

DIX   STD VAL
      LDX VAL
      CPX #10        ; Nombre de dizaines
      BMI UNIT
      INC D
      SUBD #10
      BRA DIX

UNIT  STAB U
      RTS

;***** AFFICHAGE DU RESULTAT en ASCII *****
ConvAsc ldx #ValAscii
      LDAA M          ;affiche les Volts (Milliers en mV)
      ADDA #$30
      staa 0,x
      inx
      LDAA #','      ;affiche virgule
      staa 0,x
      inx
      LDAA C          ;centaines de mV
      _____
      _____

      LDAA D          ;dizaines de mV
      ADDA #$30
      staa 0,x
      inx
      LDAA U          ;affiche les mV
      ADDA #$30
      staa 0,x
      inx
      LDAA #'V'      ;affiche le V de Volt
      staa 0,x
      inx
      RTS

;***** Tempo 100 us pour CAN *****
Tempo  ldx #FFFF      ; Valeur à calculer
temp2  dex
      bne temp2
      rts

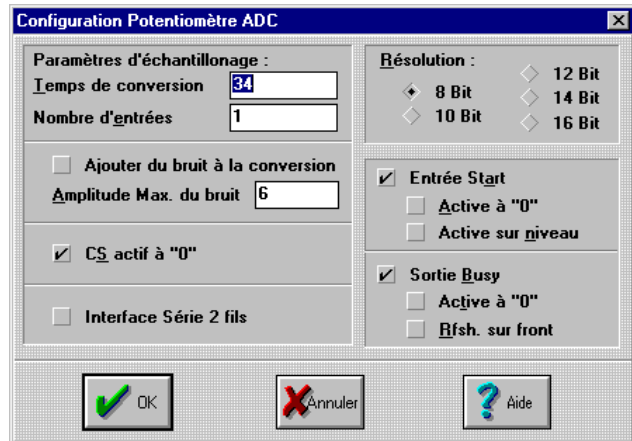
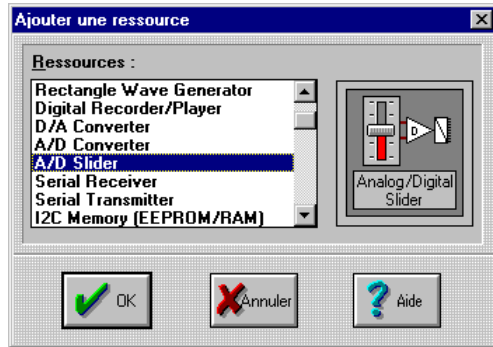
;*----- Vecteurs d'interruptions -----
      org $fffe
      fdb Debut

      end

```

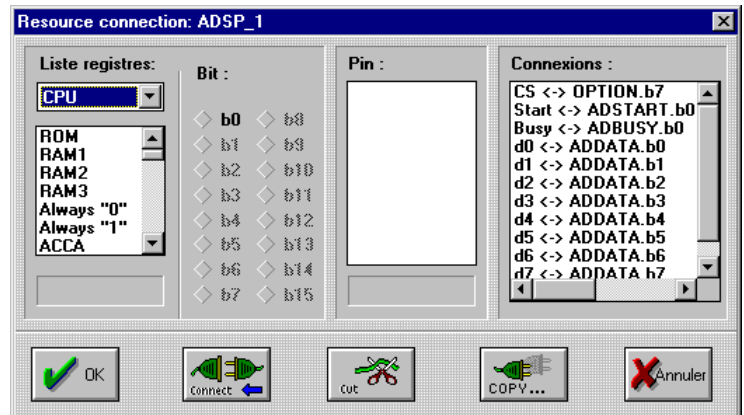
III) Simulation du programme voltmètre avec UMPS:

On désire utiliser la ressource nommée "A/D Slider" correspondant à un potentiomètre associé au convertisseur du HC11 (le potentiomètre permet d'imposer sur une entrée du HC11 une tension comprise entre V_{RL} et V_{RH} , assurant une valeur numérique entre \$00 et \$FF après conversion).



Le convertisseur dispose de trois broches qui déterminent son fonctionnement:

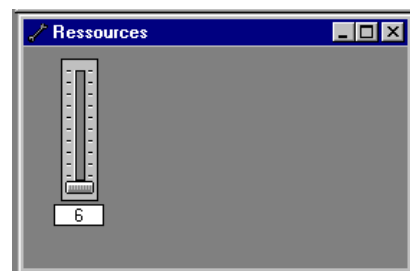
- CS : Active le convertisseur sur niveau haut (ou bas), à connecter à ADPU (OPTION bit7).
- START : Démarre une conversion. Un bit spécial a été prévu dans le modèle UMPS du HC11 (ADSTART.b0).
- BUSY : Lorsque cette ligne est à l'état 1, le convertisseur est occupé. De même un bit spécial a été prévu dans le modèle UMPS du HC11 (ADBUSY.b0)



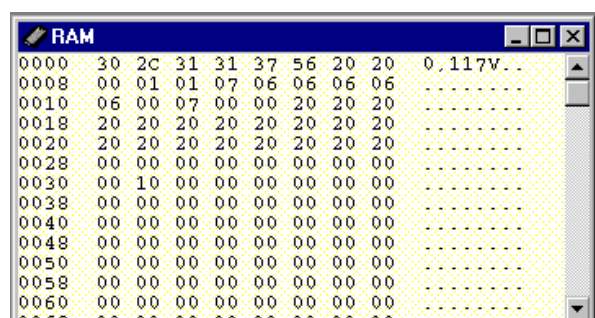
Cette ressource simule le fonctionnement d'un convertisseur Analogique/Digital à N bits connecté à un ou plusieurs potentiomètres. Son fonctionnement est en tout point identique au convertisseur A/D, avec en plus la possibilité de générer un bruit sur la conversion.

Le temps de conversion est réglable en nombre de cycles CPU (32 cycles pour le HC11).

Lorsque le potentiomètre est configuré sauvegardez le projet et les différentes fenêtres. La valeur numérique correspondant à la position apparaît en bas. Cette valeur peut être changée soit à la souris en glissant le curseur, ou en double cliquant sur la valeur.



De même configurez la RAM en demandant d'afficher par groupe de 8 octets pour visualiser les variables. La visualisation montre les valeurs en hexa et en ASCII. Ainsi on peut vérifier que pour N=6, le quantum étant de 19,5mV, le résultat est de 0,117V.



Programme (Complet) de gestion du CAN avec conversion en ASCII: CAN3.asm

```
;*****
;* Lycée Maurice GENEVOIX (INGRE)
;* Auteur: Mr COTTET
;* Date: 28/10/99
;* Programme CAN3.asm
;* Essai du port E lecture du convertisseur CAN
;* avec affichage de la valeur en volt
;* 00 a FF => 0.000V a +5.000V en ASCII
;* En RAM à partir de $0000
;*
;***** Déclaration des constantes *****
;*OPTION equ $1039 ;ADPU,CSEL,IRQE,DLY,CME,-,CR1,CR0
;*ADCTL equ $1030 ;CCF,-,SCAN,MULT,CD,CC,CB,CA
;*ADR1 equ ADCTL+1
;*ADR2 equ ADCTL+2
;*ADR3 equ ADCTL+3
;*ADR4 equ ADCTL+4

;***** Variables RAM *****
ORG 0
ValAscii Rmb 8 ; Résultat en ASCII
M Rmb 1 ; Milliers
C Rmb 1 ; Centaines
D Rmb 1 ; Dizaines
U Rmb 1 ; Unités (valeur en Volt)
N Rmb 1 ; Valeur du CAN en Hexa
Res0 Rmb 1 ; 4 valeurs résultats du convertisseur
Res1 Rmb 1 ; correspondant aux 4 voies AN0 ... AN3
Res2 Rmb 1
Res3 Rmb 1
VAL Rmb 2 ; Calcul sur 16 bits

;***** programme principal *****
ORG $F800

Debut lds #$00FF ; (ou $00C0 en mode Bootstrap)
jsr InitCAN

suite jsr StartConv ; conversion de PE0 ... PE3
jsr ReadConv ; Recopie des valeurs converties
ldaa Res0
STAA N ; Valeur en Hexa
JSR DECIM ; CONVERSION EN DECIMAL
JSR ConvAsc ; Affiche RESULTAT en ASCII
bra suite

;***** Sous programmes *****
InitCAN ldaa OPTION
oraa #$80 ;ADPU (analog to digital power UP)
anda #%10111111 ;CSEL=0 =>horloge interne E (sinon RC)
staa OPTION
jsr Tempo ;Tempo > 100uS pour alim CAN stable
ldaa #$10 ;SCAN=0 MULT=1 (conversion non continue)
staa ADCTL ;des 4 voies AN0 ... AN4)
rts

;***** Démarre conversion CAN *****
StartConv ldaa #$10 ;SCAN=0 MULT=1 (monocoup et 4 voies AN0 ... AN3)
staa ADCTL ;écriture dans ADCTL => relance conversion
rts

;*****
ReadConv ldaa ADCTL
Rola ;carry = CCF (1 indique fin conv)
bcc ReadConv
ldd ADR1
std Res0
ldd ADR3
std Res2
rts

;***** CONVERSION EN DECIMAL *****
DECIM LDAA N ;N= Nombre ... convertir
Ldab #39 ;multiplication par 39/2=19.5mV (quantum)
mul ;résultat*2 en mV sur 16 bits dans D
```

```

        lsrd                ;division par 2 pour le quantum de 19.5mV
        CLR M               ;chiffre des milliers (en mV => V)
        CLR C               ;idem centaines
        CLR D               ;dizaines
        CLR U               ;unités

SUI1   STD VAL              ;*** RECHERCHE DU NOMBRE EN DECIMAL, 4 CHIFFRES ***
        LDX VAL
        CPX #1000           ; Nombre de paquets de 1000
        BMI CENT
        INC M
        SUBD #1000
        BRA SUI1

CENT   STD VAL
        LDX VAL
        CPX #100           ; Nombre de centaines
        BMI DIX
        INC C
        SUBD #100
        BRA CENT

DIX    STD VAL
        LDX VAL
        CPX #10            ; Nombre de dizaines
        BMI UNIT
        INC D
        SUBD #10
        BRA DIX

UNIT   STAB U
        RTS

;***** AFFICHAGE DU RESULTAT en ASCII *****
ConvAsc ldx #ValAscii
        LDAA M              ;affiche les Volts (Milliers en mV)
        ADDA #$30
        staa 0,x
        inx
        LDAA #','          ;affiche virgule
        staa 0,x
        inx
        LDAA C              ;centaines de mV
        ADDA #$30
        staa 0,x
        inx
        LDAA D              ;dizaines de mV
        ADDA #$30
        staa 0,x
        inx
        LDAA U              ;affiche les mV
        ADDA #$30
        staa 0,x
        inx
        LDAA #'V'          ;affiche le V de Volt
        staa 0,x
        inx
        RTS

;***** Tempo 100 us pour CAN *****
Tempo   ldx    #$FFFF
temp2   dex
        bne    temp2
        rts

;*----- Vecteurs d'interruptions -----
        org $fffe
        fdb Debut

end

```