

# CONVERSION A/N SUR µC

Les microcontrôleurs faible coût sont souvent dépourvus de convertisseurs

Analogique / Digital. De nombreux fabricants proposent des circuits qui

remplissent efficacement cette fonction. Quand le coût du convertisseur est

élevé pour l'application développée, on cherchera à réaliser la fonction de conversion sans utiliser de circuits spécialisés.

**L**a technique de conversion proposée ici peut être mise en œuvre notamment pour mesurer des valeurs issues de potentiomètres, de CTN ou CTP (pour la mesure de température), de LDR (mesure d'éclairement), ou tout autre conditionnement de capteur résistif dont la valeur évolue lentement, ce pour un très faible coût. La technique la plus simple pour mesurer une résistance consiste à mesurer le temps de charge d'une capacité à travers la résistance dont on désire déterminer la valeur. Si la capacité est complètement déchargée, le temps de charge de la capacité pour atteindre la valeur  $V_{ref}$  est de ( $R_1$  connectée à  $V_{lim}$ ) :

$$t_1 = R_1 \cdot C \cdot \ln(V_{lim} - V_{ref}) / V_{lim}$$

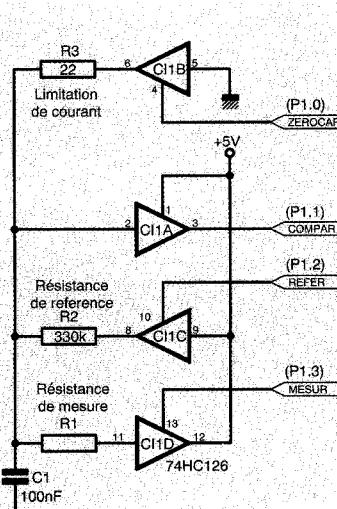
On peut ainsi déduire la valeur de la résistance  $R_1$  si la tension  $V_{lim}$  et  $V_{ref}$  ainsi que la capacité  $C$  sont connues. Les tolérances sur ces différents paramètres n'étant souvent pas bien contrôlées, cette méthode directe n'est pas facilement utilisable. En revanche, on peut répéter la même opération en chargeant la capacité  $C$  ; cette fois à travers une résistance  $R_2$ , on obtiendra :

$$t_2 = R_2 \cdot C \cdot \ln(V_{lim} - V_{ref}) / V_{lim}$$

Puis en effectuant le rapport des deux temps :

$$t_1/t_2 = R_1/R_2$$

On obtient ainsi une mesure relative de résistance indépendante des fluctuations des tensions d'alimentation et de référence. On pourra même effectuer ces mesures avec un processeur dont la fréquence de l'oscillateur est déterminée par un oscillateur RC. La même méthode pourra être utilisée pour traiter des capteurs capacitifs. Le listing 1 concrétise cette méthode en utilisant un processeur de la famille 805X. Avec cette famille de processeurs, il est nécessaire d'ajouter des composants d'interface entre les composants analogiques et le microprocesseur. En effet, la structure des ports d'entrée / sortie est telle que l'on ne peut pas mettre les lignes de ports en haute impédance (contrairement aux microprocesseurs Motorola ou Micro-



■ Figure 1 : schéma à mettre en œuvre.

chip par exemple). Nous avons donc utilisé un 74HC126 qui est un quadruple buffer à commande de haute impédance. Celui-ci assure la remise à zéro de la capacité, la charge de la capacité via une des deux résistances et pour finir détecte la charge de la capacité au-delà d'une tension  $V_{ref}$  (approximativement égale à  $V_{cc}/2$ ). Suivant la structure matérielle de l'application réalisée, on pourrait substituer au buffer de mise à zéro un transistor ou un MOS ; de même, à la place des buffer commandant la charge, on aurait pu utiliser des commutateurs analogiques (405X ou 4016), et remplacer le détecteur de charge de la capacité par un comparateur ou un amplificateur opérationnel.

La résolution de la mesure dépend du temps de charge de la capacité à travers la résistance de référence. Avec un microprocesseur cadencé à 12 MHz, la résolution temporelle sur le temps de mesure est de 4 microsecondes. Ainsi avec une résistance de 330 k et une capacité de 100 nF, le temps de mesure sera approximativement de :  $330 \text{ k} \times 100 \text{ nF} \times \ln(1/2) = 22 \text{ ms}$ .

Ce temps de mesure correspond à une valeur de comptage de :

$$22 \text{ ms} / 4 \mu\text{s} = 5500$$

Cette valeur correspond à la résolution maximum que l'on peut obtenir avec cet ensemble capacité, résistance de référence.

On peut augmenter cette résolution en augmentant la vitesse d'exécution de la boucle de comptage, soit en ayant une vitesse d'horloge plus élevée, soit en réalisant le comptage avec un des timers disponibles.

En fait, la résolution utile est limitée par l'influence de perturbations extérieures et par les erreurs sur le seuil de détection de la tension de référence, ainsi que diverses non-linéarités introduites dans le système (courant de sortie des buffers...). On peut néanmoins tabler sur une répétitivité d'environ 1/1000 soit une résolution correspondant à un A/D 10 bits. Cette résolution est très nettement suffisante pour acquérir la position d'un potentiomètre (5 à 7 bits suffisent amplement) ou pour mesurer la variation de résistance d'une CTN.

Si on souhaite augmenter encore la résolution, on aura recours au filtrage des valeurs mesurées. Ce filtrage permettra aussi de stabiliser une mesure issue de capteurs légèrement fluctuants. La première solution pour filtrer ces mesures est de faire la moyenne des  $n$  précédentes mesures. Les microcontrôleurs disposant de peu de mémoire vive, cette solution n'est utilisable qu'avec des valeurs  $n$  faibles.

Pour obtenir un filtrage efficace tout en utilisant peu de mémoire, on réalisera un filtre passe-bas récursif de premier ordre, qui réalise une fonction appelée parfois moyenage exponentiel. Cette fonction numérique est l'équivalent direct du simple réseau RC que l'on utilise dans le monde analogique. La variable CONDEN joue le rôle du condensateur dans le réseau RC. À chaque itération les opérations suivantes sont effectuées :

CONDEN = CONDEN - CONDEN /  $n$  + Valeur Mesurée

Valeur Filtrée = CONDEN /  $n$ .

Plus  $n$  est grand, plus le filtrage est ef-

```
ZEROCAP EQU 90H ; P1.0 mise à zéro de la capacité
COMPAR EQU 91H ; P1.1 entrée indiquant le dépassement de Vref
REFER EQU 92H ; P1.2 charge de la résistance référence
MESUR EQU 93H ; P1.3 charge de la résistance de mesure
```

```
?PR?_div32?DIV32 SEGMENT CODE
?DT?_div32?DIV32 SEGMENT DATA
RSEG ?DT?_div32?DIV32
ACCU32: DS 4 ; registre 32 bits externe
RSEG ?PR?_div32?DIV32
```

; Division 32 bits de ACCU32 par R4:R5:R6:R7. Le résultat est dans ACCU32  
; et le reste de la division dans R0:R1:R2:R3. R4:R5:R6:R7 sont inchangés,  
; B est modifié et vaut 0 en sortie de la division. ACCU32 est constitué de  
; 4 octets ACCU32+0:ACCU32+1:ACCU32+2:ACCU32+3 (ACCU32+0 est le MSB)  
; La division par 0 donne pour résultat FFFFH.  
; Taille de div32 : 67 octets  
; temps d'exécution :  $7 + (35 + [11]) \times 32 + 4$   
; soit dans le pire cas (FFFFH / 1) :  $46 \times 32 + 11 = 1483$  cycles.  
; Dans certains cas, on est sûr de n'avoir un résultat qui sur 16 bits et on  
; peut modifier le programme pour n'effectuer que 16 décalages et donc diviser  
; le temps par deux (par exemple quand le résultat est issu d'une règle de 3 de  
; type  $Y = X * n / p$  avec  $p > n$ ). Dans ce cas, on initialisera la division avec  
; le numérateur dans R2:R3:ACCU32+0:ACCU32+1, on supprimera l'initialisation  
; de R2 et R3 à 0, et on chargerà B avec 16 au lieu de 32.

```
_div32: CLR A ; initialise le MSB du numérateur
MOV R0,A
MOV R1,A
MOV R2,A ; à supprimer si on est sur d'avoir un
MOV R3,A ; résultat sur 16 bits (voir en-tête)
MOV B,#32 ; compteur de décalage
7 cycles
DIV320: CLR C ; multiplie le numérateur par 2 et laisse la
MOV A,ACCU32+3 ; place pour le résultat (voir plus bas)
RLC A
MOV ACCU32+3,A
MOV A,ACCU32+2
RLC A
MOV ACCU32+2,A
MOV A,ACCU32+1
RLC A
MOV ACCU32+1,A
MOV A,ACCU32+0
RLC A
MOV ACCU32+0,A ; et pousse les bits dans R0:R1:R2:R3 <- C
RLC A
MOV R3,A ; Dans une division sur papier, le numérateur
MOV A,R2 ; ne « bouge pas » et c'est le dénominateur qui
RLC A ; est divisé par 2 (ou par 10 si on est en
MOV R2,A ; en base 10). Ici, c'est le numérateur qui
MOV A,R1 ; « bouge » et est multiplié par 2 et le
RLC A ; dénominateur reste fixe. On pourrait faire
MOV R1,A ; les divisions sur papier exactement de la
MOV A,R0 ; même manière.
RLC A
MOV R0,A ; compare le numérateur et le dénominateur
SUBB A,R7 ; la retenue est propagée
MOV A,R2 ; vers le MSB
SUBB A,R6 ; (une manière de comparer sur le 8051 est de
MOV A,R1 ; faire une soustraction et de tester C.
SUBB A,R5 ; ici la comparaison est effectuée sur deux
MOV A,R0 ; mots de 32 bits : R0:R1:R2:R3 et R4:R5:R6:R7)
SUBB A,R4
JC DIV321 ; dénominateur > numérateur, résultat = 0.
35 cycles
; dénominateur <= numérateur, effectue la soustraction et résultat = 1
MOV R0,A ; sauve le nouveau MSB du numérateur
MOV A,R3 ; recalcule la soustraction
SUBB A,R7 ; numérateur - dénominateur
MOV R3,A ; en sauivant cette fois le résultat
MOV A,R2 ; (Les bits à droite de la partie qui sert à la
SUBB A,R6 ; comparaison du numérateur ne sont pas
MOV R2,A ; affectés par la soustraction. Dans une
MOV A,R1 ; division effectuée sur papier, on complète de
SUBB A,R5 ; manière sous-entendue le dénominateur avec
MOV R1,A ; des zéros, il en est de même ici).
; le numérateur était plus grand que le dénominateur, un 1 est donc poussé dans
; le résultat. Celui-ci prend place dans la place laissée libre par le
; numérateur lorsqu'il est décalé. A la fin de la division, le résultat occupe
; donc entièrement la place qui était prise initialement par le numérateur. Le
; décalage du résultat et du numérateur s'effectue donc en une seule même
; opération.
INC ACCU32+3 ; en fait met un 1 dans le LSB (voir au-dessus)
11 cycles
DIV321: DJNZ B,DIV320 ; passe ainsi les 32 bits
RET
4 cycles
```

; Mesure du rapport de deux résistances.
; On supposera que la résistance à mesurer est plus faible que la résistance
; de référence. La résistance de référence et la capacité doivent être
; choisies de manière à avoir une bonne résolution sur la durée de la mesure.
; Avec  $C = 150$  nF et  $Rref = 300$  k, la résolution maximum est d'environ
; 2500 points.
; Le résultat de la mesure est dans R6:R7. Si  $Rmesu = Rref$ ,  $R6:R7 = 0xFFFF$ 
; si  $Rmesu = Rref/2$ ,  $R6:R7 = 0x8000$  ...
; Si on désire une répétitivité de plus de 10 bits, il est nécessaire de
; moyenne la valeur mesurée en utilisant le sous programme FILTRE.

```
; initialise la division
MESURE: CLR A
MOV ACCU32+2,A ; met à 0 les bits de poids faibles
MOV ACCU32+3,A ; du numérateur
```

```
MOV R4,A ; ainsi que les bits de poids forts
MOV R5,A ; du dénominateur
; effectue la mesure du temps de charge de la capa à travers la résistance
; de mesure
;t1 = Rmesu X C X ln((Valim - Vref)/Valim)
; = Rmesu X C X ln(1/2) = Rmesu X C X 0.693
MOV DPTR,#0 ; initialise la mesure
CLR ZEROCAP ; laisse la capa se charger
SETB MESUR ; avec la résistance à mesurer
INC DPTR ; incrément le compteur 16 bits
JNB COMPAR,MESRES ; jusqu'au seuil
CLR MESUR ; puis arrête la charge
SETB ZEROCAP ; et vide la capa
MOV ACCU32+0,DPH ; transfère pour la division
MOV ACCU32+1,DPL ; dans le numérateur (multiplié par 65536)
```

; petite temporisation pour laisser la capacité se vider
CLR A
MOV R0,A
TMP: DJNZ R0,TMP

; effectue la mesure du temps de charge de la capa à travers la référence
;t2 = Rref X C X ln((Valim - Vref)/Valim)

```
MOV DPTR,#0 ; initialise la mesure
CLR ZEROCAP ; laisse la capa se charger
SETB REFER ; avec la référence
INC DPTR ; incrément le compteur 16 bits
JNB COMPAR,MESREF ; jusqu'au seuil
CLR ZEROCAP ; puis arrête la charge
SETB R6,DPH ; et vide la capa
MOV R7,DPL ; transfère pour la division
; dans le dénominateur
```

; calcule  $65536 \times t1 / t2 = 65536 \times Rmesu / Rref$ 
CALL \_div32 ; résultat dans ACCU32

; ACCU32 est une valeur de 32 bits qui contient le résultat de la mesure.
; Si  $Rmesu$  est inférieure à  $Rref$ ,  $ACCU32 < 65536$  (ou  $ACCU32 < 10000H$ ). Les
; seuls octets de poids faible sont donc significatifs
MOV A,ACCU32+0 ; teste si il n'y a pas
ORL A,ACCU32+1 ; de débordement
JZ MESOK ; ACCU32+2 = ACCU32+1 = 0,  $Rmesu < Rref$ 
MOV ACCU32+2,#0FFH ; si il y a un débordement
MOV ACCU32+3,#0FFH ; met la valeur maximum dans le résultat
MESOK: MOV R6,ACCU32+2 ; prend le résultat de la division
MOV R7,ACCU32+3 ; (talonni à 0xFFFF si dépassé)
RET

### ■ Listing 1

```
EFFICA EQU 2 ; efficacité du filtrage (de 1 à 8)
```

```
?DT?_filtre?FILTRE SEGMENT DATA
RSEG ?DT?_filtre?FILTRE
CONDEN: DS 3 ; variable associée au filtrage
```

; Divise CONDEN par  $2^n$  avec  $n = EFFICA$ . Le résultat est dans R1:R3
; Les calculs du filtre sont tels que le résultat est toujours sur 16 bits

```
DIVN: MOV R0,CONDEN+0 ; transfère CONDEN dans R0:R1:R2
      MOV R1,CONDEN+1
      MOV R2,CONDEN+2
      MOV R3,#EFFICA
DIVNO: MOV A,R0 ; décale R0:R1:R2
      RRC A
      MOV R0,A
      MOV A,R1
      RRC A
      MOV R1,A
      MOV A,R2
      RRC A
      MOV R2,A
      DINZ R3,DIVNO ; EFFICA fois
      RET ; soit une division par  $2^{\text{EFFICA}}$ 
```

; Filtrage de la valeur contenue dans R6:R7 résultat dans R6:R7
; La valeur CONDEN sur 3 octets constitue en quelque sorte l'équivalent
; numérique d'un condensateur dans lequel s'accumule les valeurs à filtrer.

```
FILTRE: ACALL DIVN ; divise CONDEN par  $2^n$ 
      CLR C ; effectue CONDEN - (CONDEN /  $2^n$ )
      MOV A,CONDEN+2 ; soustrait du LSB
      SUBB A,R2
      MOV CONDEN+2,A
      MOV A,CONDEN+1
      SUBB A,R1
      MOV CONDEN+1,A
      MOV A,CONDEN+0
      SUBB A,#0
      MOV CONDEN+0,A
      ADD A,R7
      MOV CONDEN+2,A
      MOV A,CONDEN+1
      ADDC A,R6
      MOV CONDEN+1,A
      MOV A,CONDEN+0
      ADDC A,#0
      MOV CONDEN+0,A
      ACALL DIVN ; propage la retenue dans le MSB
      MOV R6,AR1 ; CONDEN = CONDEN - ((CONDEN/2^n) + R6:R7)
      MOV R7,AR2 ; dans R6:R7, le résultat du filtrage
      RET
```

### ■ Listing 2

ficace. Dans le programme FILTRE (Listing 2) n est une puissance de 2 et est déterminé par la constante EFFICA. On peut faire varier cette constante de 1 à 8 pour une valeur n variant de 2 à 256. Pour mettre au point l'application qui utilise ces sous-programmes, le plus simple est d'essayer avec et sans un appel à FILTRE, avec différentes valeurs de

EFFICA et de juger du résultat obtenu. Un petit programme complet qui utilise ces sous-programmes est disponible sur le serveur ERP.