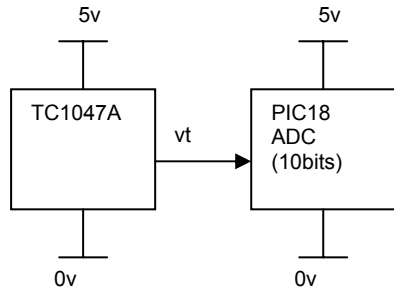


## 1. Le problème :

Les fonctions devant effectuer une conversion analogique numérique sur une grande plage avec une grande résolution nécessitent des convertisseurs analogiques numériques souvent coûteux.

### Un exemple : mesure de température

On souhaite mesurer une température entre 0°C et 50°C avec une résolution de 0,1°C, le  $\mu\text{C}$  utilisé est un PIC18 MICROCHIP équipé d'un CAN 10bits avec  $V_{\text{ref}}=5\text{v}$ .



Le capteur Microchip TC1047A mesure des températures de -40°C à +125 °C  
vt évolue linéairement : de 10mV/°C avec un offset de 500mV

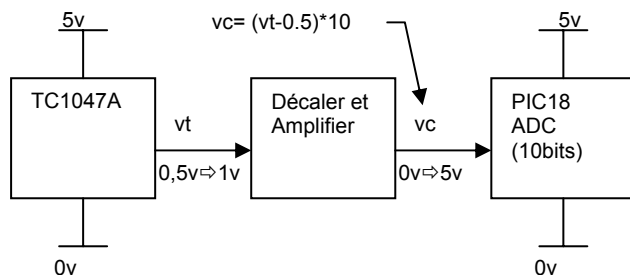
100mV à -40°C  
500mV à 0°C  
750mV à +25°C  
1750mV à +125°C

L'étalonnage sera effectué par une correction logicielle, la résolution dépendant exclusivement du CAN. Le quantum du CAN est  $q = \frac{V_{\text{ref}}}{2^n}$  soit ici 4.88mv (n représente le nombre de bits du CAN)

Pour une mesure de température entre 0°C et 50°C, vt évolue de 500mV à 1v. La résolution de la mesure serait donc de  $\frac{500\text{mV}}{4,88\text{mV}} = 102\text{pts}$  soit  $\frac{50^\circ\text{C}}{102\text{pts}} = 0,49^\circ\text{C}$ , insuffisante ici ,

Une solution consiste à réduire  $V_{\text{ref}}$ . Pour obtenir une résolution de 0,1°C il faut  $q \leq 1\text{mV}$  d'où  $V_{\text{ref}} \leq 2^{10} * 1\text{mV}$  donc  $V_{\text{ref}} \leq 1,024\text{v}$ . Le data sheet du PIC18F452 donne  $V_{\text{refmin}}=1,8\text{v}$  !

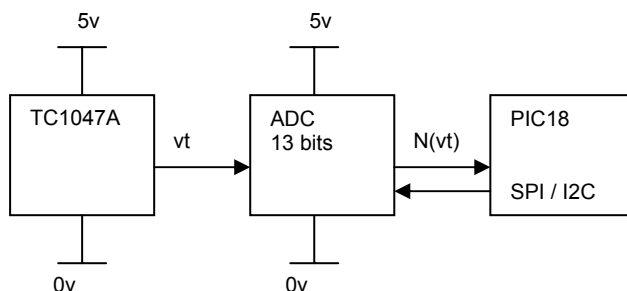
Il est possible d'amplifier et décaler cette tension afin qu'elle entre dans la plage maximale du CAN (0v-5v) mais cela au prix d'une électronique de traitement de signal complexe, onéreuse, analogique donc avec une tolérance sur les composants discrets.



Une autre solution consiste à utiliser un CAN avec une plus grande résolution mais donc plus onéreux et surtout extérieur au micron contrôleur, augmentant ainsi la consommation et réduisant la fiabilité.

En faisant le calcul inverse :  $\frac{50^\circ\text{C}}{0,1^\circ\text{C}} = 500\text{pts}$  donc un quantum  $q = \frac{500\text{mV}}{500\text{pts}} = 1\text{mV}$

$2^n = \frac{V_{\text{ref}}}{q}$  d'où  $n = \frac{\log\left(\frac{V_{\text{ref}}}{q}\right)}{\log(2)} = 12,28$  pratiquement il faudrait un convertisseur 13 bits



Une troisième solution consiste à augmenter le nombre de bits effectif (Effective Number Of Bits ou ENOB) par suréchantillonnage, c'est cette technologie qui est présentée ici.

## 2. Approche intuitive

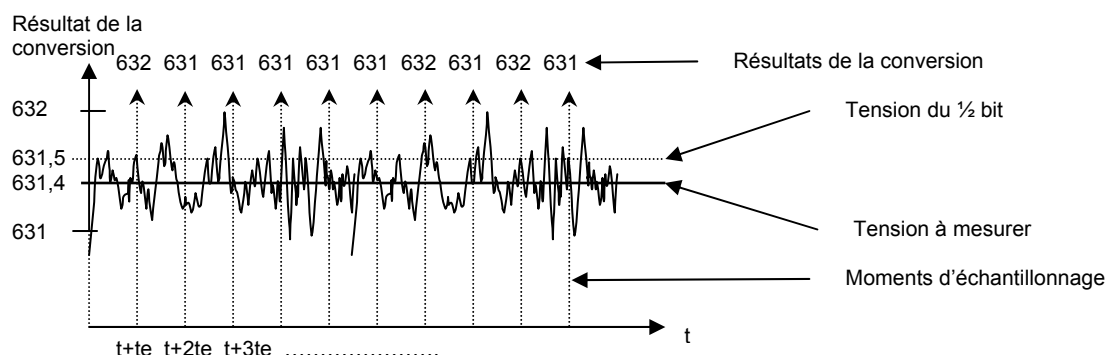
La conversion analogique numérique introduit un bruit de conversion lié aux paliers du convertisseur. Le résultat est toujours une approximation à  $\frac{1}{2}$  bit près (ici à  $\pm 2,44\text{mV}$ ). Généralement on peut considérer que ce bruit s'approche d'un bruit blanc. On démontre que le rapport signal bruit d'un CAN est au maximum pour un signal d'entrée sinusoïdal

$$SNR = 6.02 * N + 1,77\text{dB}$$

L'incertitude sur le bit de poids faible le rend inexploitable en présence de bruit.

Le SNR augmente donc avec le nombre de bits. Le bruit de quantification est créé par l'erreur de quantification sur le bit de poids faible. Comme le bruit est blanc, l'erreur est purement aléatoire, si la tension à mesurer est inférieure au  $\frac{1}{2}$  bit, il y aura probablement plus d'échantillons mesurés aux bits inférieurs qu'au supérieur et vice versa si la tension à mesurer est supérieure au  $\frac{1}{2}$  bit. Un suréchantillonnage suivi d'un moyennage devrait augmenter la résolution (donc le nombre de bits effectifs).

### Exemple de suréchantillonnage par 8 :



Moyenne des 8 premiers échantillons : 631,375 pour une tension correspondant à 631,4 (attention c'est un nombre réel, on recherche un résultat sous forme d'un nombre entier). Il est donc possible par suréchantillonnage et moyennage d'augmenter le nombre de bits effectifs.

Ex : la somme des huit mesures précédentes fait 5051 pour une excursion comprise entre 0 et 8192 ( $8 \text{ fois } 2^{10}$ )

	10 bits	Somme des 8 échantillons
N max	1024	8192
Mesure max	632	5051
Mesure min	631	Pas de mini
N min	0	0

La somme des huit échantillons est proportionnelle au résultat d'un CAN 13 bits ( $0 \leq N \leq 8191$ )

Il faut 13 bits pour coder un nombre entre 0 et 8192. Le nombre effectif de bits du CAN est donc maintenant de 13.

**Attention**, ceci n'est qu'une première approche, ce résultat n'est pas correct. Le paragraphe suivant décrit une approche mathématique de la réduction de bruit par suréchantillonnage. **Il faut en réalité  $4^3$  échantillons (64) pour augmenter le nombre de bits de 3 !**

### 3. Approche théorique de la réduction du bruit par suréchantillonnage.

Le critère de Nyquist impose une fréquence d'échantillonnage  $f_e$  au moins le double de la plus grande fréquence du signal à convertir.

La technique du suréchantillonnage est basée sur l'approximation que le bruit total de quantification

d'un CAN peut s'écrire  $\sigma = \frac{q^2}{12}$  avec  $q$  : quantum.

La puissance de densité spectrale du bruit (Power Spectral Density ou PSD) est

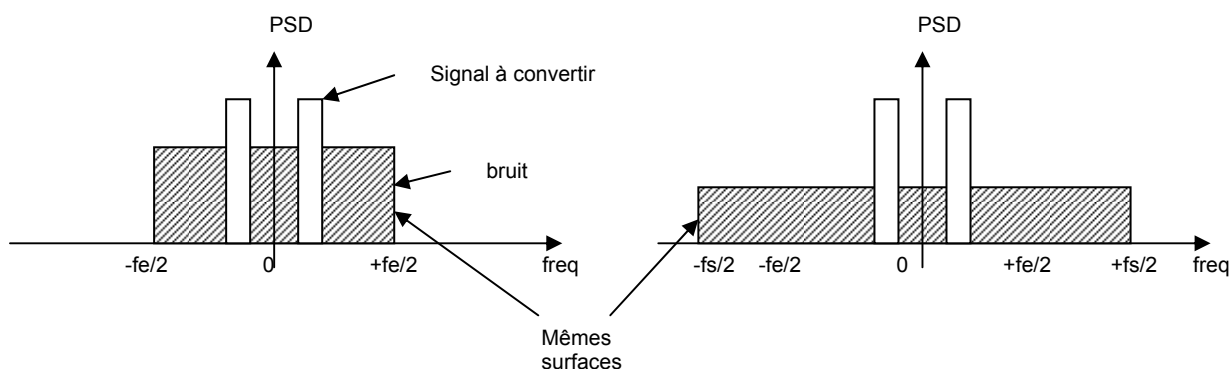
$$PSD = \frac{q^2}{12} * \frac{1}{f_e} = \frac{q^2}{12 f_e}$$

avec  $f_e$  la fréquence d'échantillonnage

Le PSD est inversement proportionnel à la fréquence d'échantillonnage. Augmenter  $f_e$  revient à diminuer le bruit donc augmenter le ENOB.

$f_e$  : fréquence d'échantillonnage initiale (souhaitée)

$f_s$  : fréquence de suréchantillonnage



On pose  $M = \frac{f_s}{f_e}$  le suréchantillonnage

Le gain du SNR est :  $SNR_g = 10 * \log(M)$

Le nombre de bit étant proportionnel au SNR :  $N = f(SNR/6)$  (d'après l'équation page 2)

$$ENOB = \frac{10 * \log(M)}{6} + N$$

( $N$  est le nombre de bits intrinsèque du CAN)

Un suréchantillonnage de 2 augmente le ENOB de 0,5.

Un suréchantillonnage de 4 augmente le ENOB de 1.

Un suréchantillonnage de 16 augmente le ENOB de 2.

**Le suréchantillonnage  $M = 4^k$  Avec  $k$  le nombre de bits supplémentaires souhaités**

#### Remarques :

Cette méthode ne fonctionne qu'en présence d'un bruit blanc d'amplitude moyenne supérieure au quantum. Si le niveau de bruit est insuffisant, on doit ajouter artificiellement du bruit au signal à convertir (dithering)

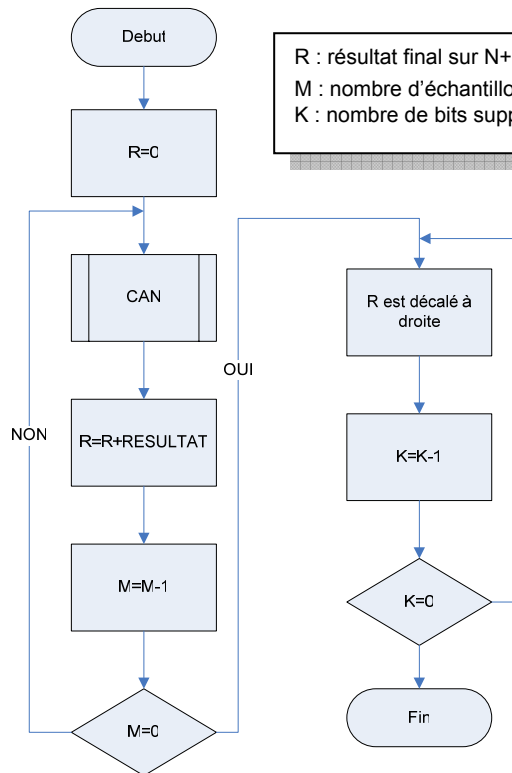
Le suréchantillonnage augmente le ENOB mais diminue la fréquence d'échantillonnage maximum.

Ex : Si la fréquence d'échantillonnage pratique d'un  $\mu C$  PIC18 est de 10KHz (échantillonnage plus calculs divers). Elle tombe à 156Hz pour une augmentation de 2 bits (suréchantillonnage fois 64) et à 39Hz pour une augmentation de 4 bits (suréchantillonnage fois 256).

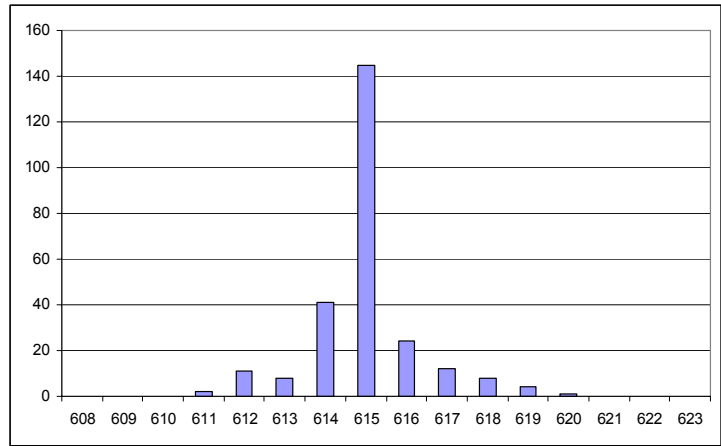
La pratique du suréchantillonnage est donc adaptée à la mesure de signaux évoluant lentement (cas de la température)

### 4. Récupérer le résultat, la décimation

Le suréchantillonnage offre la possibilité d'augmenter le ENOB de K. Pour récupérer le résultat on ajoute les échantillons et l'on ramène la somme au nombre de bits souhaités en divisant le résultat par  $2^K$  (décimation). (Diviser par 2 revient à effectuer un décalage à droite).



R : résultat final sur N+k bits  
 M : nombre d'échantillons à acquérir ( $M = 4^k$ )  
 K : nombre de bits supplémentaires souhaités



Répartition des 256 échantillons du programme page suivante. La courbe de Gauss peut être légèrement déséquilibrée si le bruit n'est pas blanc.

**Exemple :**

On désire augmenter le ENOB de 1 (passer de 10 bits à 11 bits), donc suréchantillonner par  $M = 4^1 = 4$

Sur 10 bits  $0 \leq N \leq 1024$ , avec  $V_{ref} = 5v$ ,  $q = 4,88mV$

Echantillons	Résultats binaires	Résultats décimaux	Tension équivalente sur 10bits
1	1101011001	857	4,1822v
2	1101011000	856	4,1773v
3	1101011000	856	4,1773v
4	1101011010	858	4,1870v
Somme	110101100011	Moyenne	4,1797v

Le résultat est décalé à droite du nombre de bits supplémentaires souhaités (1 bit) (décimation)  
 Résultat de la mesure = 11010110001 (1713d)  
 Sur 11 bits  $0 \leq N \leq 2048$   $q = 2,44mV$   
 Donc  $V_{mes} = N * q = 4,1797v$  qui correspond bien à la moyenne des mesures.

**Bibliographie:**

- Reducing ADC Quantization Noise : Richard Lyons & Randy Yates
- Understanding A/D Converter Performance Specification (MICROCHIP AD693) : Steve Bowling
- Improving ADC resolution by oversampling and averaging : Silicon Laboratories
- Enhancing ADC resolution by oversampling : ATMEL app note AVR121
- Demystifying Sigma-Delta ADCs : DALLAS-MAXIM app note 1870
- Sigma-Delta Measurement ADCs : James M. Bryant

MICROCHIP : bibliothèque (assembleur) de suréchantillonnage pour PIC18 avec l'utilitaire MAESTRO.

## 5. Un exemple de programme en C de suréchantillonnage et décimation sur PIC18.

Remarque : ce programme peut être testé sur PICDEM2+, il mesure la tension sur AN0 (potentiomètre)

```

/*
Programme adover.c (compilateur C18)
Augmenter la résolution d'un CAN par surechantillonnage et decimation
C.Dupaty 06/2006
*/
#define K 4          // nb de bits supplémentaires
#define M 256       // nb d'échantillons : 4 puissance K

#include <p18f452.h>

unsigned int resultat; // le résultat sur N+K bits
                    //N=10 pour le PIC18F452

void init_can(void)
{
    ADCON0=1;          // CAN on. CLOCK=FOSC/2. CANAL0 (RA)
    ADCON1=0x80;      // justification à droite, toutes les
entrées AN activées, VREF+=VDD VREF-=VSS
}

// effectue une conversion sur le canal spécifiée
unsigned int mesvolt(unsigned char canal)
{
    ADCON0=ADCON0|(canal<<3);
    ADCON0bits.GO_DONE=1; // SOC
    while(ADCON0bits.GO_DONE); // attend EOC
    return (ADRES);
}

// effectue une mesure de "entree" par surechantillonnage et decimation
unsigned int adover(unsigned char entree)
{
    unsigned int c;
    unsigned long r=0;
    for (c=0;c<M;c++) // somme des M echantillons
    {
        r+=mesvolt(entree);
    }
    r>>=K; // decimation
    return (r);
}

void main(void)
{
    init_can();
    resultat=adover(0);
    while(1); // le resultat sur 10+K bits est disponible
}

```