

# Cours

## Filtrage numérique d'un signal d'évolution lente avec un microcontrôleur

Version 1.2.2.

### 1- Contexte

### 2- La chaîne de mesure

### 3- Le filtre numérique

- 3-1- Définitions générales
- 3-2- Filtre numérique à « moyenne glissante »
- 3-3- Réponse en fréquence d'un filtre numérique
- 3-4- Le problème du repliement du spectre de fréquence
- 3-5- Le filtre anti-repliement
- 3-6- Théorème de Shannon
- 3-7- Réponse en fréquence du filtre numérique à « moyenne glissante »
  - 3-7-1- Etude mathématique de la réponse en fréquence
  - 3-7-2- Influence du nombre d'échantillons
  - 3-7-3- Influence de la fréquence d'échantillonnage
- 3-8- Réjection de mode commun (élimination du bruit du secteur)

### 4- Exemple concret :

#### Carte d'acquisition à 7 entrées analogiques à microcontrôleur PIC 16F88 (Microchip)

- 4-1- Présentation
- 4-2- Schéma électrique de la carte
- 4-3- Principe de fonctionnement de la carte
  - 4-3-1- Les filtres anti-repléments
  - 4-3-2- Le convertisseur analogique - numérique (ADC)
  - 4-3-3- Le filtrage numérique
- 4-4- La liaison RS232
  - 4-4-1- Le câble de communication entre l'ordinateur et la carte
  - 4-4-2- Protocole de communication entre l'ordinateur et la carte

## 1- Contexte

On suppose que le signal à échantillonner est un **signal d'évolution lente** (de période très supérieure à la seconde).

Ce pourra être le signal provenant d'un capteur de température analogique (LM335) ou d'un capteur de pression (MPX2200).

On s'intéressera tout particulièrement au **filtrage numérique**, qui sera réalisé par un **microcontrôleur** (de manière purement logicielle).

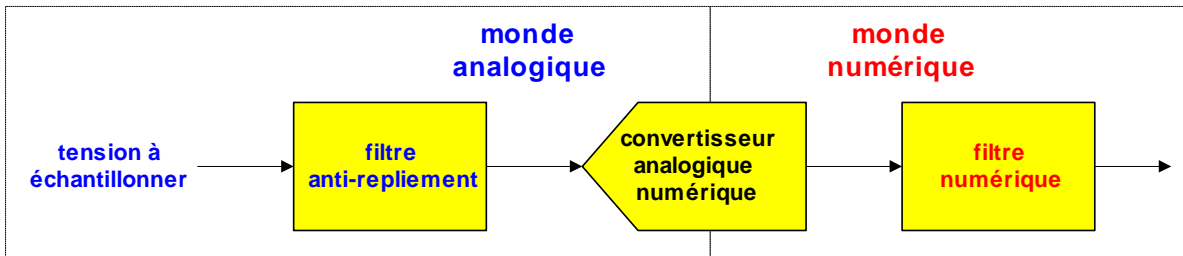
On choisira une technique de filtrage numérique élémentaire, qui consiste à répéter  $n$  fois la mesure, et d'en calculer la moyenne arithmétique (ce sera le travail du microcontrôleur, à partir des résultats du convertisseur analogique - numérique).

Cela revient à déterminer la composante continue du signal, le filtre numérique est donc du type passe-bas.

D'autre part, nous verrons qu'en choisissant intelligemment la fréquence d'échantillonnage du convertisseur analogique - numérique, notre filtre numérique pourra **éliminer complètement le bruit du secteur (50 Hz) ainsi que ses harmoniques**.

## 2- La chaîne de mesure

Structure générale :



La tension à échantillonner est souvent une tension issue d'un capteur analogique. Le filtre anti-repliement est un filtre passe-bas analogique dont la fréquence de coupure est liée à la fréquence d'échantillonnage du convertisseur analogique - numérique. Le filtrage numérique est ici réalisé par un microcontrôleur (de façon logicielle).

## 3- Le filtre numérique

### 3-1- Définitions générales

Il existe plusieurs types de filtres numériques linéaires :

- Filtre numérique non récursif (ou filtre à réponse impulsionnelle finie RIF)
- Filtre numérique récursif (ou filtre à réponse impulsionnelle infinie RII)

Nous ne nous intéresserons qu'aux filtres numériques « non récursifs », dont voici l'expression générale :

$$y_k = \sum_{\ell=0}^{n-1} h_{\ell} x_{k-\ell}$$

$x_{k-\ell}$  représente la valeur numérique fournie par le convertisseur analogique - numérique à l'instant  $t = kT_e - \ell T_e$

$y_k$  représente la valeur numérique en sortie du filtre numérique à l'instant  $t = kT_e$

Un filtre numérique est caractérisé par :

- $n$  : le nombre d'échantillons (c'est un nombre entier, supérieur à 1)
- $h_0, h_1, \dots, h_{n-1}$  :  $n$  coefficients réels

$T_e$  est la période d'échantillonnage du convertisseur analogique - numérique.

$F_e = 1 / T_e$  est la fréquence d'échantillonnage.

Par définition, la fonction de transfert en  $z$  du filtre s'écrit :

$$H(z) = \sum_{\ell=0}^{n-1} h_{\ell} z^{-\ell}$$

### 3-2- Filtre numérique à « moyenne glissante »

Pour des raisons de simplicité, nous choisirons un filtre numérique non récursif avec des coefficients identiques, tels que :

$$h_{\ell} = \frac{1}{n}$$

d'où :

$$y_k = \sum_{\ell=0}^{n-1} \frac{1}{n} x_{k-\ell}$$

$$H(z) = \frac{1 + z^{-1} + z^{-2} + \dots + z^{-(n-1)}}{n}$$

En fait, **ce filtre numérique calcule la valeur moyenne des n derniers échantillons**.  
On parle aussi de **filtre à « moyenne glissante »**.

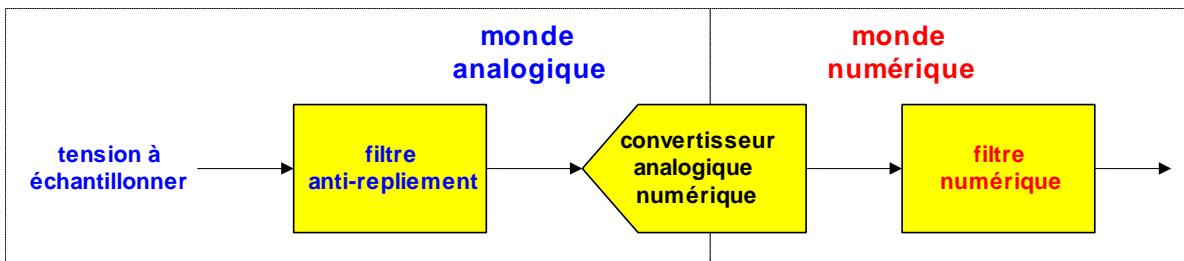
Toujours pour des raisons de simplicité, le nombre d'échantillons n sera une puissance de 2.

➤ Exemple avec :  $n = 2^3 = 8$

$$y_k = \frac{x_k + x_{k-1} + x_{k-2} + x_{k-3} + x_{k-4} + x_{k-5} + x_{k-6} + x_{k-7}}{8}$$

$$H(z) = \frac{1 + z^{-1} + z^{-2} + \dots + z^{-7}}{8}$$

On suppose que le convertisseur analogique – numérique a une résolution de 10 bits (nombre binaire de 10 bits), avec une fréquence d'échantillonnage de 1 kHz (c'est-à-dire une conversion toutes les 1 ms).



Voici les 8 dernières conversions de ce convertisseur :

		<b>Binaire naturel</b>	<b>Décimal</b>
$x_{k-7}$	mesure à l'instant $t - 7$ ms	01 11111110	510
$x_{k-6}$	mesure à l'instant $t - 6$ ms	10 00001001	521
$x_{k-5}$	mesure à l'instant $t - 5$ ms	01 11111101	509
$x_{k-4}$	mesure à l'instant $t - 4$ ms	10 00000011	515
$x_{k-3}$	mesure à l'instant $t - 3$ ms	10 00000000	512
$x_{k-2}$	mesure à l'instant $t - 2$ ms	10 00001010	522
$x_{k-1}$	Avant dernière mesure (à l'instant $t - 1$ ms)	01 11111011	507
$x_k$	Dernière mesure (à l'instant $t$ )	01 11111111	511

La dispersion des mesures est en partie due à la présence de bruit.

Un microcontrôleur sait très bien faire les additions (en binaire).

Par exemple le microcontrôleur Microchip PIC 16F88, possède l'instruction **addwf** qui lui permet d'additionner 2 nombres de 8 bits, en un cycle d'horloge (soit 0,2  $\mu$ s avec une horloge à quartz de 20 MHz).

La somme de 8 nombres de 10 bits donne un résultat sur  $10 + 3 = 13$  bits :

$$\begin{aligned}
 & x_k + x_{k-1} + x_{k-2} + x_{k-3} + x_{k-4} + x_{k-5} + x_{k-6} + x_{k-7} \\
 & = 10000\ 00001011 \text{ (en binaire naturel)} \\
 & = 4017 \text{ (en décimal)}
 \end{aligned}$$

Il faut maintenant faire une division pour obtenir la valeur moyenne.

Un microcontrôleur standard, comme le PIC 16F88, ne possède pas d'instruction de division (ni d'instruction de multiplication).

Le PIC 16F88 doit reconstruire la division uniquement avec des instructions d'additions, ce qui donne lieu à une routine assez complexe (et lente).

Heureusement, la division devient triviale quand on divise par une puissance de 2.

Pour diviser par  $2^n$ , on décale la virgule de  $n$  bits vers la gauche.

(C'est la même chose en décimal : pour diviser par  $10^n$ , on décale la virgule de  $n$  chiffres vers la gauche :  $6594 / 1000 = 6,594$ ).

Ainsi, pour diviser par 8, il suffit de décaler la virgule de 3 bits vers la gauche :

$$10000\ 00001011 \rightarrow 10\ 00000001,011$$

Ce qui donne pour la partie entière :

$$\begin{aligned}
 & 10\ 00000001 \\
 & = 513 \text{ en décimal}
 \end{aligned}$$

Et pour le reste :

$$011 = 0,5 \times 0 + 0,25 \times 1 + 0,125 \times 1 = 0,375$$

Finalement :

$$y_k = 513,375$$

$$(4107 / 8 = 513,375)$$

On remarquera que la valeur moyenne sur 8 conversions donne un nombre de 13 bits.  
On augmente ainsi, de manière logicielle, la résolution du convertisseur analogique – numérique (on passe de 10 à 13 bits).

### 3-3- Réponse en fréquence d'un filtre numérique

Voici l'expression générale de la fonction de transfert :

$$H(f) = \sum_{\ell=0}^{n-1} h_{\ell} e^{-2\pi j f \ell T_e}$$

$f$  : fréquence en hertz du signal échantillonné (supposé sinusoïdal)

Avec notre filtre à moyenne glissante ( $h_{\ell} = 1/n$ ) :

$$H(f) = \sum_{\ell=0}^{n-1} \frac{1}{n} e^{-2\pi j f \ell T_e}$$

que l'on peut écrire plus simplement :

$$H(f) = \frac{1}{n} \frac{\sin(2\pi f \cdot \frac{nT_e}{2})}{\sin(2\pi f \cdot \frac{T_e}{2})} e^{-2\pi j f \frac{(n-1)T_e}{2}}$$

En effet (vive les maths !) :

$$\begin{aligned}
H(f) - H(f)e^{-2\pi jfT_e} &= \sum_{\ell=0}^{n-1} \frac{1}{n} e^{-2\pi jf\ell T_e} - \sum_{\ell=0}^{n-1} \frac{1}{n} e^{-2\pi jf(\ell+1)T_e} = \sum_{\ell=0}^{n-1} \frac{1}{n} e^{-2\pi jf\ell T_e} - \sum_{\ell=1}^n \frac{1}{n} e^{-2\pi jf\ell T_e} \\
&= \frac{1}{n} (1 - e^{-2\pi jfnT_e}) \\
H(f) &= \frac{1}{n} \frac{1 - e^{-2\pi jfnT_e}}{1 - e^{-2\pi jfT_e}} \\
&= \frac{1}{n} \frac{e^{+\pi jfnT_e} - e^{-\pi jfnT_e}}{e^{+\pi jfT_e} - e^{-\pi jfT_e}} \frac{e^{-\pi jfnT_e}}{e^{-\pi jfT_e}} \\
&= \frac{1}{n} \frac{\sin(2\pi f \cdot \frac{nT_e}{2})}{\sin(2\pi f \cdot \frac{T_e}{2})} e^{-2\pi jf \frac{(n-1)T_e}{2}}
\end{aligned}$$

**En module, la fonction de transfert donne l'amplification du filtre numérique, rapport entre le niveau de sortie et le niveau d'entrée d'un signal sinusoïdal de fréquence f :**

$$A(f) = \left| \frac{1}{n} \frac{\sin(2\pi f \cdot \frac{nT_e}{2})}{\sin(2\pi f \cdot \frac{T_e}{2})} \right|$$

On en déduit le gain du filtre numérique (en dB) :

$$G(f) = 20 \log_{10} A(f)$$

En argument, la fonction de transfert donne le déphasage (en radians) entre la sortie et l'entrée du filtre numérique :

$$-2\pi f \frac{n-1}{2} T_e$$

On notera que le déphasage est linéaire.

### 3-4- Le problème du repliement du spectre de fréquence

Reprenons notre filtre à moyenne glissante :

$$A(f) = \left| \frac{1}{n} \frac{\sin(2\pi f \cdot \frac{nT_e}{2})}{\sin(2\pi f \cdot \frac{T_e}{2})} \right|$$

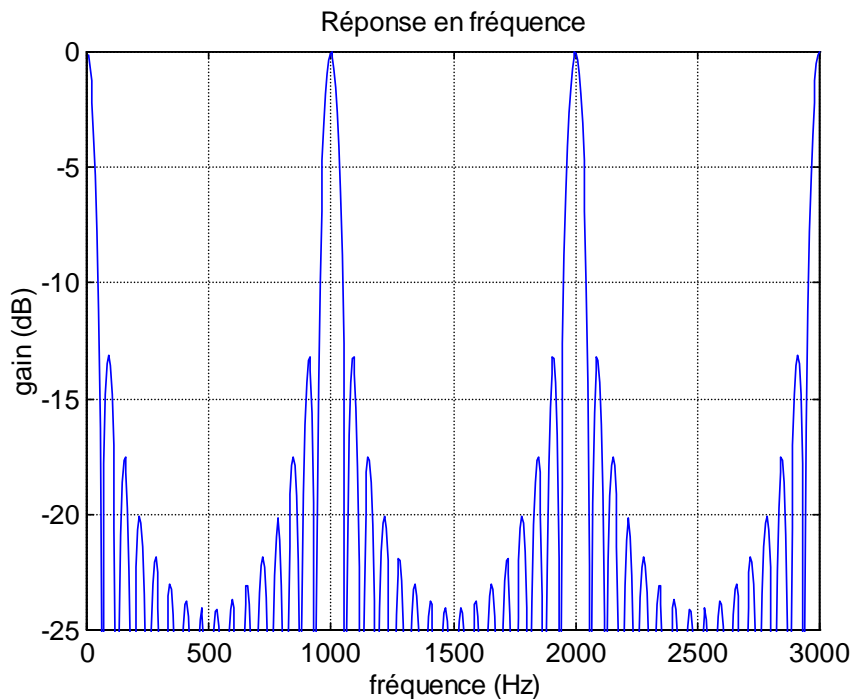
On montre que :

$$A(kF_e \pm f) = A(f)$$

avec k entier

- Application numérique avec  $F_e = 1 \text{ kHz}$  ( $T_e = 1 \text{ ms}$ ) et  $n = 16$  échantillons :

$$\left\{ \begin{array}{l} A(f) = \left| \frac{1}{16} \frac{\sin(2\pi f \cdot \frac{n \cdot 10^{-3}}{2})}{\sin(2\pi f \cdot \frac{10^{-3}}{2})} \right| \\ G(f) = 20 \cdot \log_{10} A(f) \end{array} \right.$$



N.B. Les courbes ont été tracées avec Matlab.

Pour en savoir plus sur Matlab :

[http://perso.orange.fr/fabrice.sincere/matlab/matlab\\_index.htm](http://perso.orange.fr/fabrice.sincere/matlab/matlab_index.htm)

Le spectre de fréquence est périodique (période  $F_e$ ).

**On dit que le spectre de fréquence se « replie ».**

Ainsi, les fréquences suivantes donnent la même réponse :

100 Hz	
900 Hz	(1000 – 100)
1100 Hz	(1000 + 100)
1900 Hz	(2×1000 – 100)
2100 Hz	(2×1000 + 100)
2900 Hz	(3×1000 – 100)
3100 Hz	(3×1000 + 100)
3900 Hz	
4100 Hz ...	

Un signal de 1 kHz n'est absolument pas filtré (gain 0 dB).

Cela s'explique par le fait que le signal est échantillonné exactement une fois par période ( $f = F_e$ ). Ce signal est vu par le filtre numérique comme un signal constant (fréquence nulle).

C'est exactement la même chose avec un stroboscope qui éclaire un disque en rotation : si la fréquence des flashes du stroboscope coïncide exactement avec la fréquence de rotation du disque, le disque semble immobile (le disque fait exactement un tour entre 2 flashes, ce qui donne l'illusion d'immobilité).

Il en est de même pour les signaux de fréquences :

2 kHz (1 échantillon toutes les 2 périodes exactement)

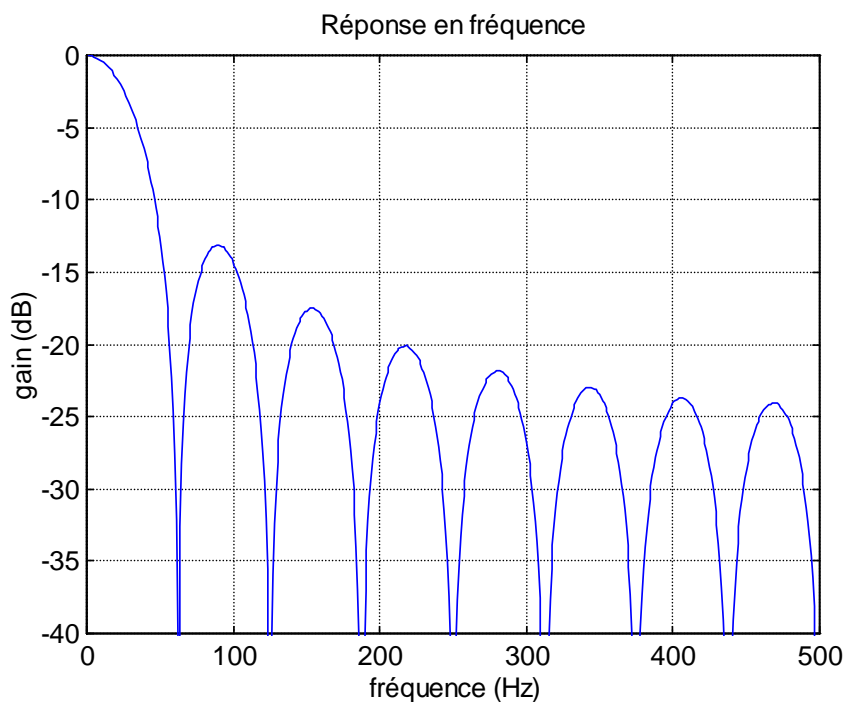
3 kHz (1 échantillon toutes les 3 périodes exactement)

4 kHz et tous les multiples entiers de 1 kHz (= fréquence d'échantillonnage)

**C'est embêtant pour un filtre numérique « passe-bas » ...**

**La solution à ce problème est l'utilisation, en amont, d'un filtre « anti-repliement ».**

Le filtre anti-repliement permet de travailler dans la zone « utile » du filtre numérique, c'est-à-dire dans la bande de fréquences située entre 0 Hz et 500 Hz ( $F_e / 2$ ) :



Entre 0 Hz et  $F_e / 2$ , il y a une diminution globale du gain avec la fréquence, caractéristique d'un filtre passe-bas.

Vous noterez cependant des discontinuités pour certaines fréquences (62,5 Hz ; 125 Hz etc...) où le gain du filtre numérique tend vers  $-\infty$ .

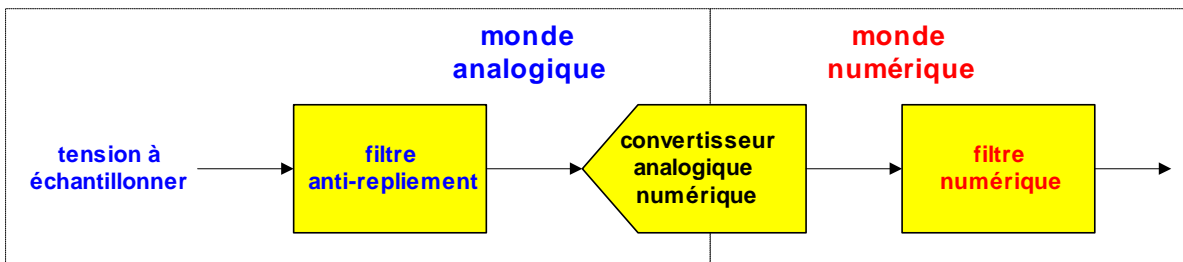
Le filtre élimine totalement ces fréquences (filtre réjecteur).

Dans la suite, nous mettrons à profit cette propriété pour supprimer le bruit de ronflement du secteur (50 Hz).



Ce que nous venons de voir dans ce cas particulier se généralise.  
Quel que soit le filtre numérique, le spectre se replie.

### 3-5- Le filtre anti-repliement



Dans le paragraphe précédent, nous avons montré qu'il est indispensable d'intercaler un filtre « anti-repliement » avant le convertisseur analogique – numérique.  
C'est un filtre analogique passe-bas de fréquence de coupure  $F_e / 2$  (dans l'idéal).

En pratique, le filtre passe-bas parfait n'existe pas et le repliement du filtre numérique ( $f > F_e / 2$ ) est partiellement éliminé.  
La qualité se paie par l'utilisation d'un filtre anti-repliement élaboré (ordre élevé).

Dans notre cas, on cherche simplement à extraire la valeur moyenne (fréquence nulle) du signal (et non pas la bande de fréquences de 0 Hz à  $F_e / 2$ , comme ce serait le cas pour un signal audio).

On peut se contenter d'un filtre passe bas du premier ordre de fréquence de coupure très inférieure à  $F_e / 2$  (mais pas trop basse car le temps de réponse du filtre deviendrait important).

### 3-6- Théorème de Shannon

Le théorème de Shannon dit que pour pouvoir reconstituer un signal analogique à partir des ses échantillons, il faut que la fréquence d'échantillonnage  $F_e$  soit égale ou supérieure à deux fois la fréquence maximale du spectre du signal analogique :

$$f < \frac{F_e}{2} \quad \text{ou} \quad f < \frac{1}{2T_e} \quad \text{ou} \quad f \cdot T_e < \frac{1}{2}$$

Dans notre cas, le signal à échantillonner est un **signal d'évolution lente** (de période très supérieure à la seconde).

Si on choisit une fréquence d'échantillonnage d'au moins 1 Hz (au moins 1 mesure par seconde), le théorème de Shannon est bien évidemment respecté.

### 3-7- Réponse en fréquence du filtre numérique à « moyenne glissante »

#### 3-7-1- Etude mathématique de la réponse en fréquence

$$A(f) = \frac{1 \cdot \sin(2\pi f \cdot \frac{nT_e}{2})}{n \cdot \sin(2\pi f \cdot \frac{T_e}{2})}$$

Nous allons étudier cette fonction dans l'intervalle  $0 \leq f \leq \frac{F_e}{2}$

On suppose que le nombre d'échantillons (n) est une puissance de 2 (n = 2, 4, 8 ...)

a) f = 0 Hz

$$A(f \rightarrow 0) = 1 \text{ (ou } G = 0 \text{ dB, gain nul)}$$

C'est un maximum

b) f =  $F_e / 2$

$$A(f = \frac{F_e}{2}) = \left| \frac{1 \cdot \sin(n \frac{\pi}{2})}{n \cdot \sin(\frac{\pi}{2})} \right| = \left| \frac{1 \cdot \sin(n \frac{\pi}{2})}{n} \right| = 0 \quad (\text{gain} - \infty)$$

C'est un minimum

c) Détermination des fréquences où l'amplification est maximum

$$\left| \sin(2\pi f \cdot \frac{nT_e}{2}) \right| = 1$$

$$\Rightarrow 2\pi f \cdot \frac{nT_e}{2} = (2k - 1) \frac{\pi}{2}$$

k est un nombre entier  $\geq 1$

$$f = (2k - 1) \frac{F_e}{2n}$$

Dans l'intervalle  $0 < f < \frac{F_e}{2}$  :

$$k < \frac{n-1}{2}$$

$$\Rightarrow \left( \frac{n}{2} - 1 \right) \text{ maximum}$$

En tenant compte du maximum à f = 0 Hz, cela donne  $\left( \frac{n}{2} \right)$  maximum.

Pour les fréquences où l'amplification est maximum :

$$A = \frac{1}{n \sin(2\pi f \cdot \frac{T_e}{2})}$$

Quand f augmente A diminue : c'est normal, c'est un filtre numérique passe-bas.

**Pour un filtrage plus efficace, on a intérêt à choisir un grand nombre d'échantillons (n).**

d) Détermination des fréquences où l'amplification est minimum (nulle)

$$\sin(2\pi f \cdot \frac{nT_e}{2}) = 0$$

$$\Rightarrow 2\pi f \cdot \frac{nT_e}{2} = k\pi$$

k est un nombre entier  $\geq 1$

$$f = k \frac{F_e}{n}$$

Dans l'intervalle  $0 < f < \frac{F_e}{2}$

$$k < \frac{n}{2}$$

$$\Rightarrow \left(\frac{n}{2} - 1\right) \text{ minimum}$$

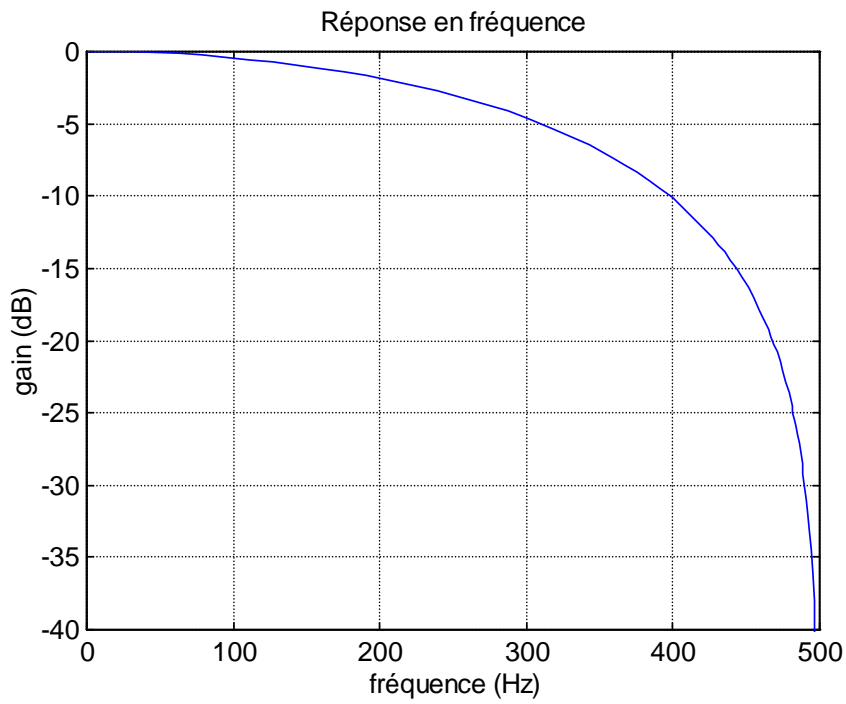
On tenant compte du minimum à  $f = \frac{F_e}{2}$ , cela donne  $\left(\frac{n}{2}\right)$  minimum.

En définitive :

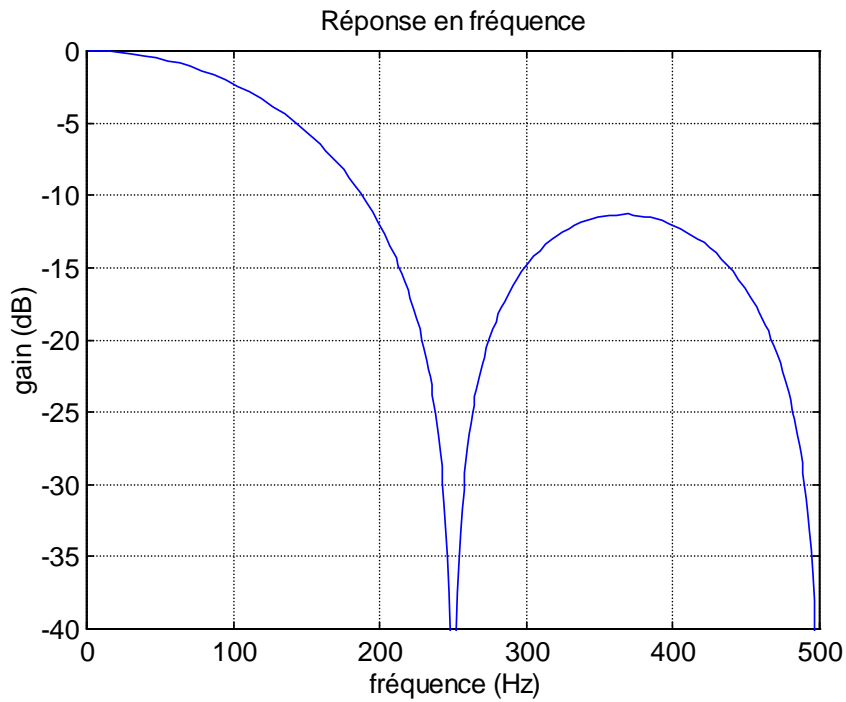
n (nombre d'échantillons)	Nombre de fréquences rejetées	Fréquences rejetées dans l'intervalle $0 \leq f \leq \frac{F_e}{2}$
2	1	$F_e/2$
4	2	$F_e/4 ; F_e/2$
8	4	$F_e/8 ; F_e/4 ; 3 F_e/8 ; F_e/2$
16	8	$F_e/16 ; F_e/8 ; 3 F_e/16 ; F_e/4 ; 5 F_e/16 ; 3 F_e/8 ; 7 F_e/16 ; F_e/2$
32	16	$F_e/32 ; F_e/16 ; 3 F_e/32 ; \dots ; 31 F_e/32 ; F_e/2$
$2^n$	$n/2$	<b><math>k F_e/n</math> avec <math>k = 1, 2, 3, \dots, n/2</math></b>

### 3-7-2- Influence du nombre d'échantillons (n)

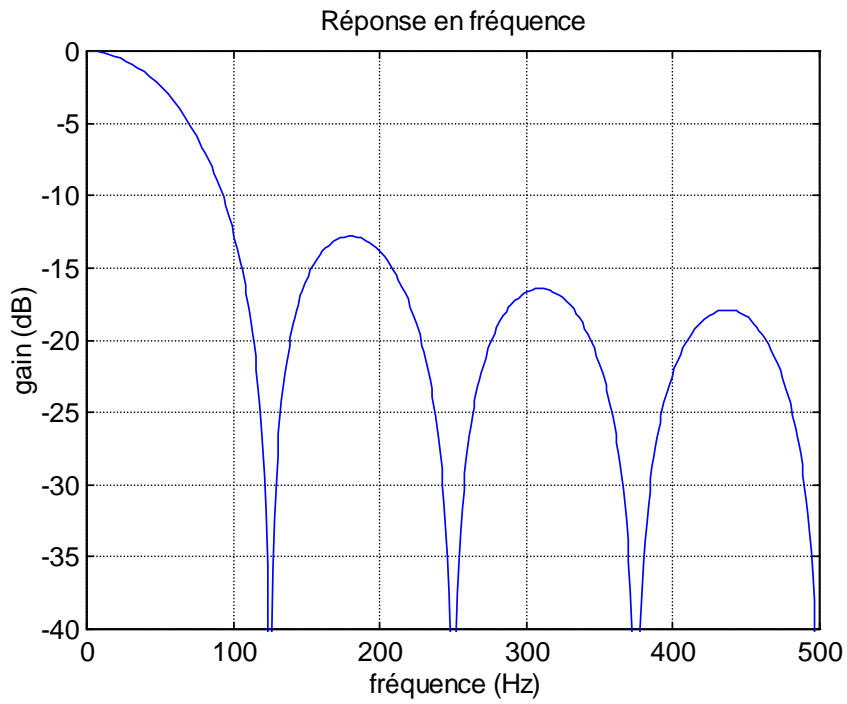
➤  $F_e = 1 \text{ kHz}$      $n = 2$  échantillons



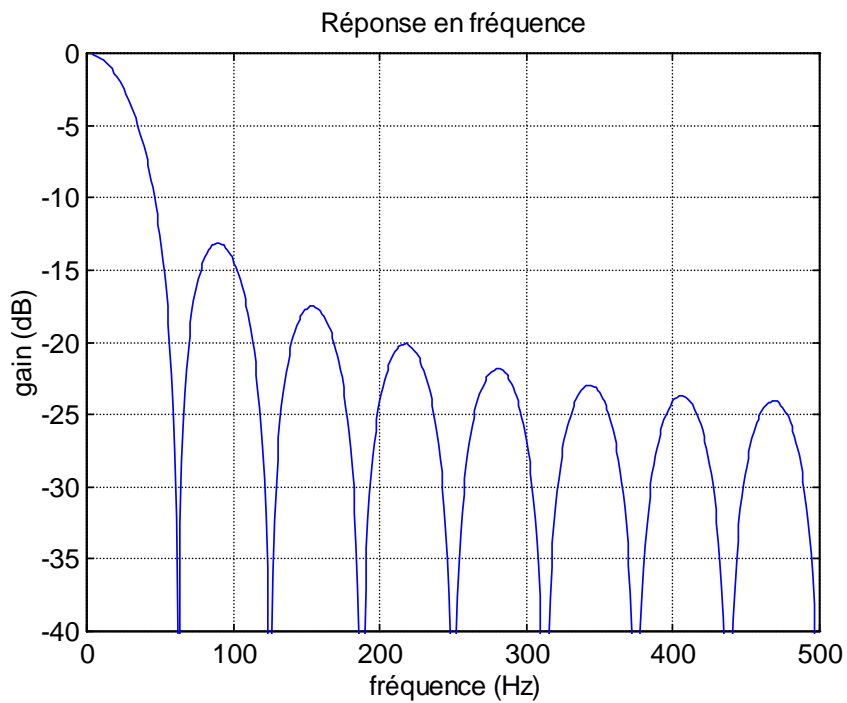
➤  $F_e = 1 \text{ kHz}$      $n = 4$  échantillons



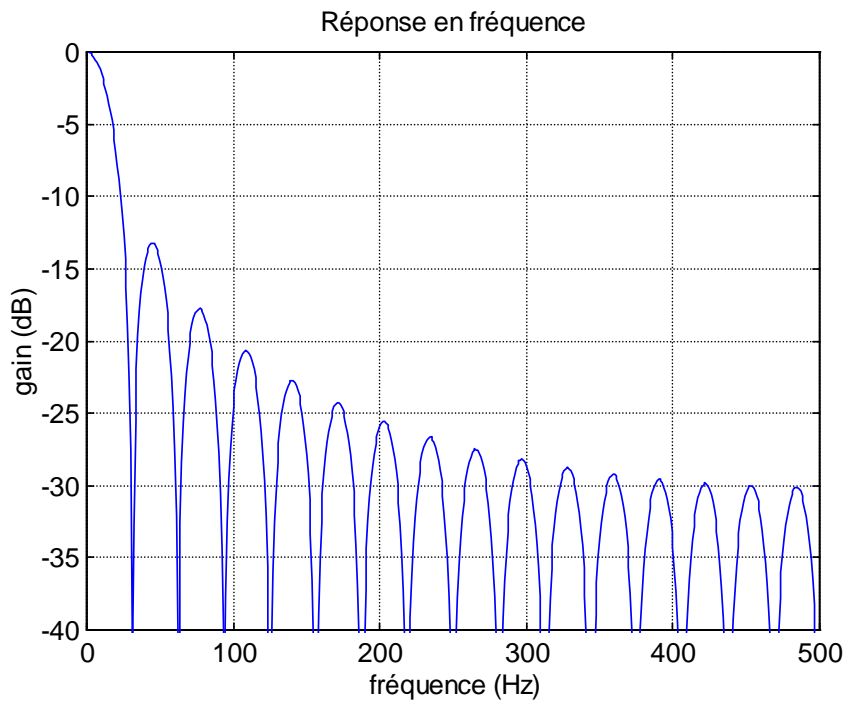
➤  $F_e = 1 \text{ kHz}$     $n = 8$  échantillons



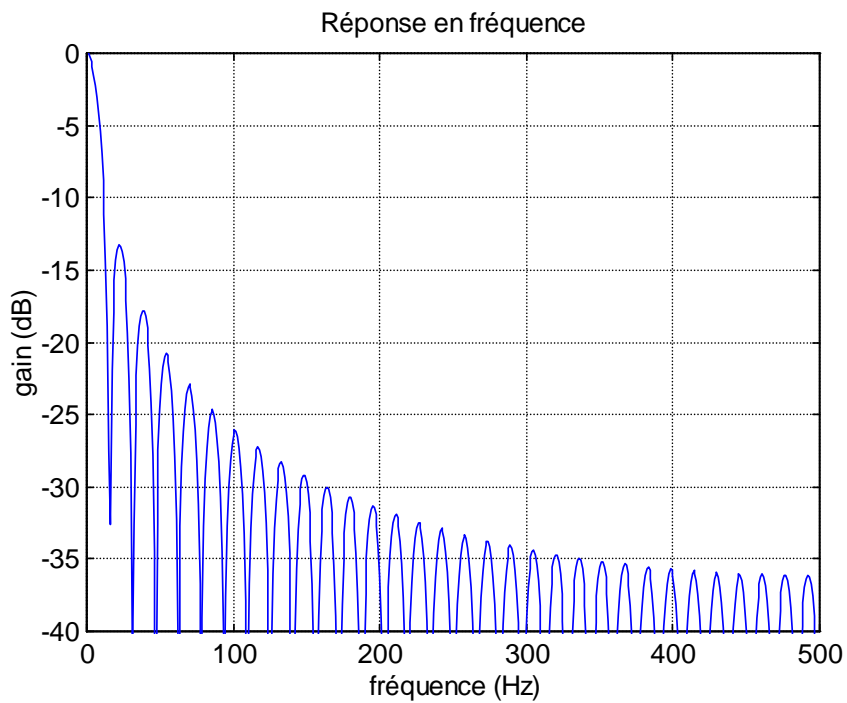
➤  $F_e = 1 \text{ kHz}$     $n = 16$  échantillons



➤  $F_e = 1 \text{ kHz}$      $n = 32$  échantillons



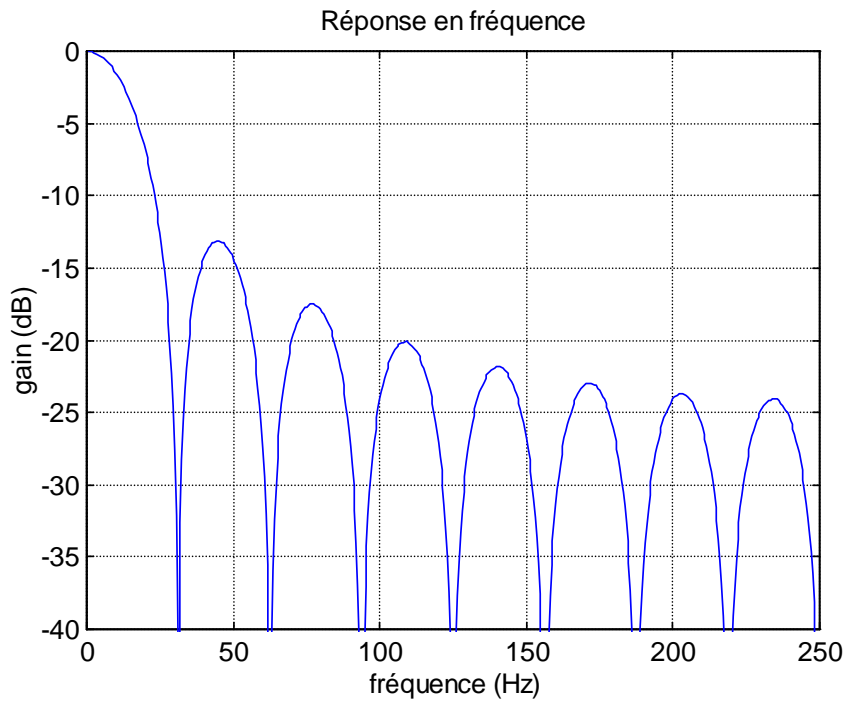
➤  $F_e = 1 \text{ kHz}$      $n = 64$  échantillons



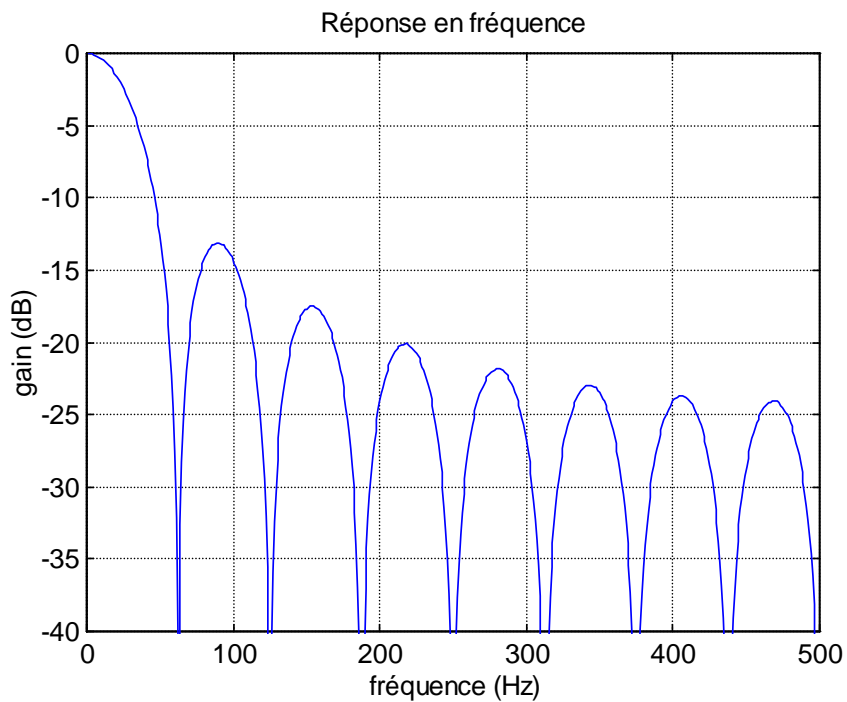
**Pour un filtrage plus efficace, on a donc intérêt à choisir un grand nombre d'échantillons (n).**

### 3-7-3- Influence de la fréquence d'échantillonnage

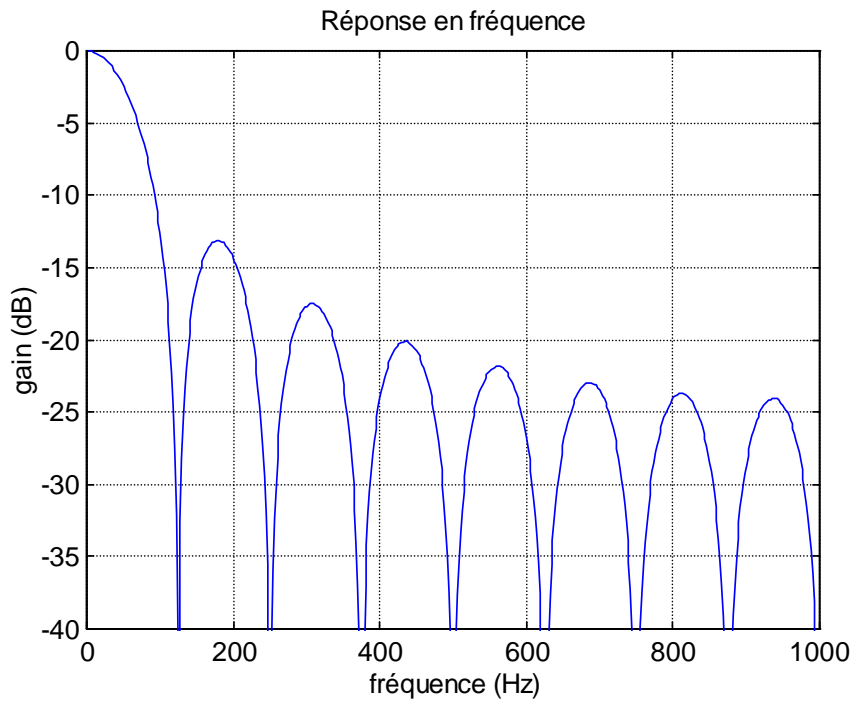
➤  $F_e = 500 \text{ Hz}$   $n = 16$  échantillons



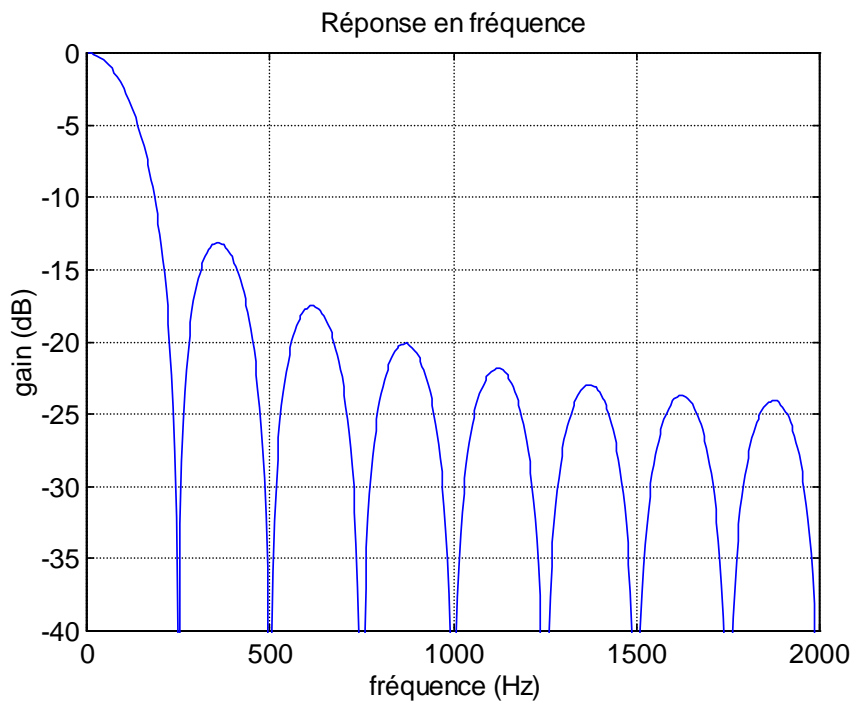
➤  $F_e = 1 \text{ kHz}$   $n = 16$  échantillons



➤  $F_e = 2 \text{ kHz}$     $n = 16$  échantillons



➤  $F_e = 4 \text{ kHz}$     $n = 16$  échantillons





### 3-8- Réjection de mode commun (élimination du bruit du secteur)

L'un des principaux bruits qui perturbe le signal de mesure est le ronflement du secteur (fréquence nominale 50 Hz).

Avec notre filtre numérique à moyenne glissante, il est facile d'éliminer le fondamental du secteur (50 Hz), ainsi que ses harmoniques (100, 150, 200, 250 Hz ...).

Nous avons montré précédemment que notre filtre rejète un certain nombre de fréquences.

Ces fréquences sont données par la relation :

$$f = k \frac{F_e}{n}$$

avec k un nombre entier.

$$\text{Dans l'intervalle: } 0 < f \leq \frac{F_e}{2}$$

$$1 \leq k \leq \frac{n}{2}$$

Déterminons les conditions pour rejeter la fréquence de 50 Hz :

$$50 = k \frac{F_e}{n}$$

$$F_e = 50 \frac{n}{k}$$

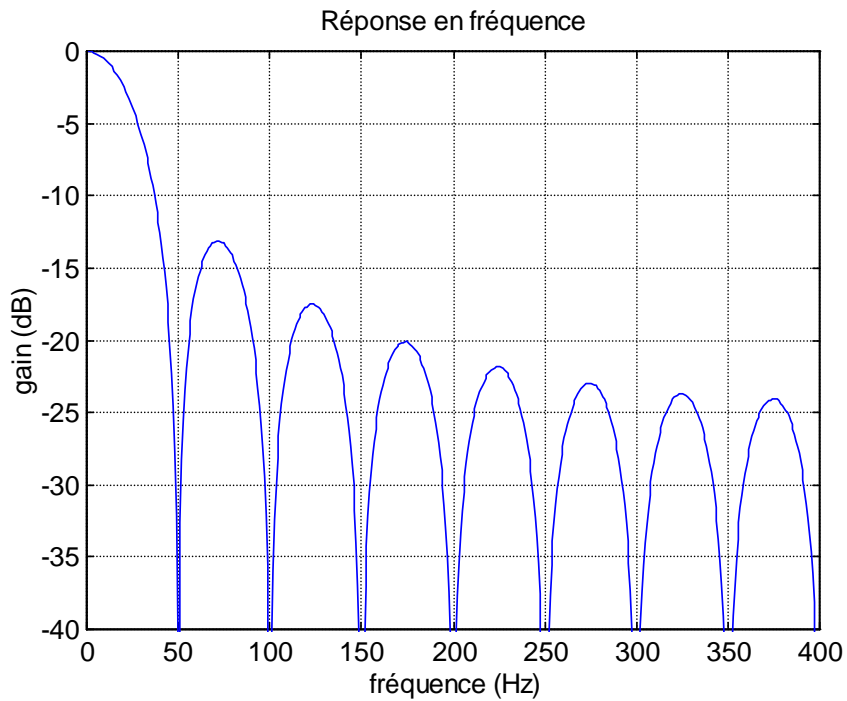
avec k un nombre entier tel que :  $1 \leq k \leq \frac{n}{2}$

Nombre d'échantillons n	Fréquence d'échantillonnage $F_e$	Fréquences rejetées (Hz) (dans l'intervalle $0 < f \leq \frac{F_e}{2}$ )	Intervalle de mesure $n T_e$
2	100 Hz	50 Hz	20 ms
4	200 Hz	50, 100	20 ms
4	100 Hz	25, 50	40 ms
8	400 Hz	50, 100, 150, 200	20 ms
8	200 Hz	25, 50, 75, 100	40 ms
8	133,3 Hz	16,7 33,3 50 66,7	60 ms
8	100 Hz	12,5 25 37,5 50	80 ms
16	800 Hz	50, 100, 150, 200, 250, 300, 350, 400	20 ms
16	400 Hz	25, 50, 75, 100, 125, 150, 175, 200	40 ms
16	266,7 Hz	16,7 33,3 50 66,7 83,3 100 116,7 133,3	60 ms
16	200 Hz	12,5 25 37,5 50 62,5 75 87,5 100	80 ms
16	160 Hz	10 20 30 40 50 60 70 80	100 ms
16	133,3 Hz	8,3 16,7 25 33,3 41,7 50 58,3 66,7	120 ms
16	114,3 Hz	7,1 14,3 21,4 28,6 35,7 42,9 50 57,1	140 ms
16	100 Hz	6,25 12,5 18,75 25 31,25 37,5 43,75 50	160 ms
...			

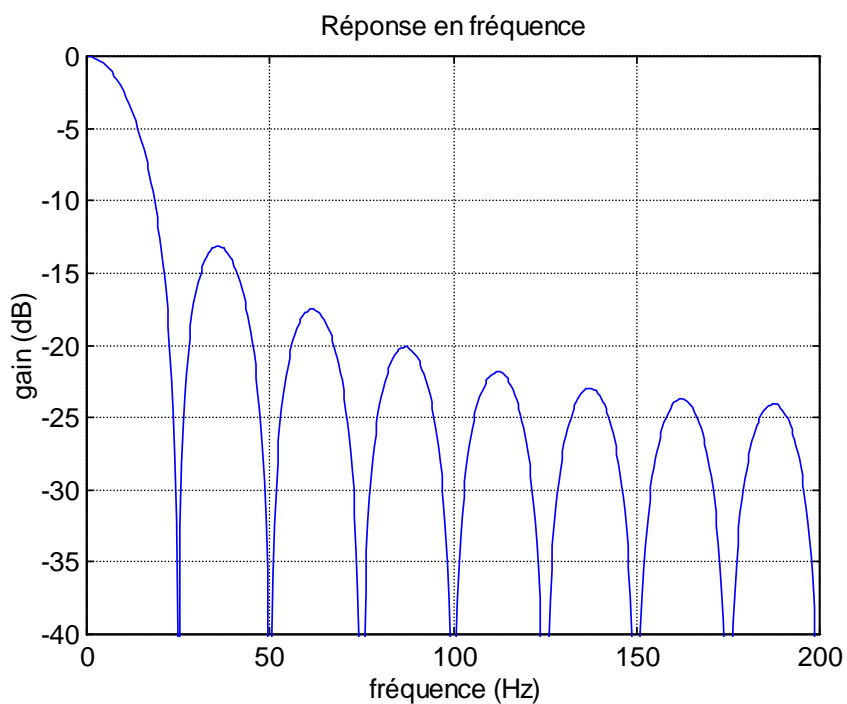
Remarque : ( $n T_e$ ) est un multiple entier de la période du secteur (20 ms).

- Exemple avec  $n = 16$  échantillons

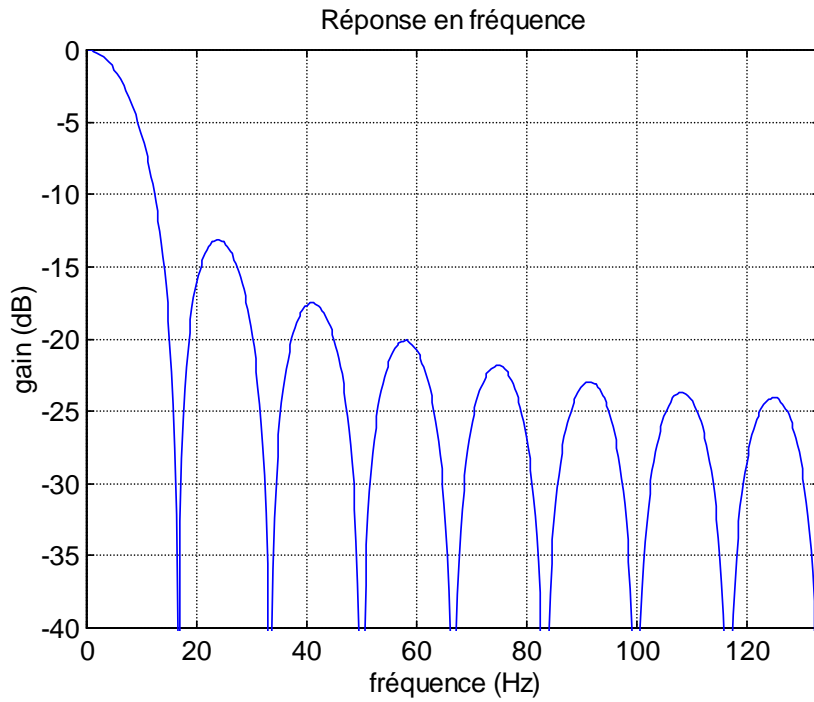
➤  $F_e = 800$  Hz       $n = 16$



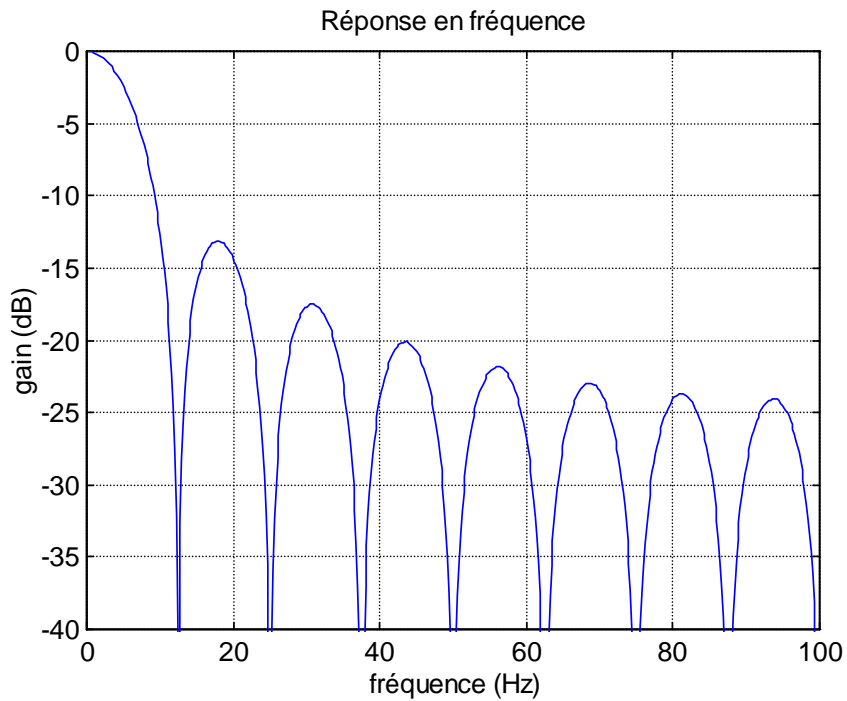
➤  $F_e = 400$  Hz       $n = 16$



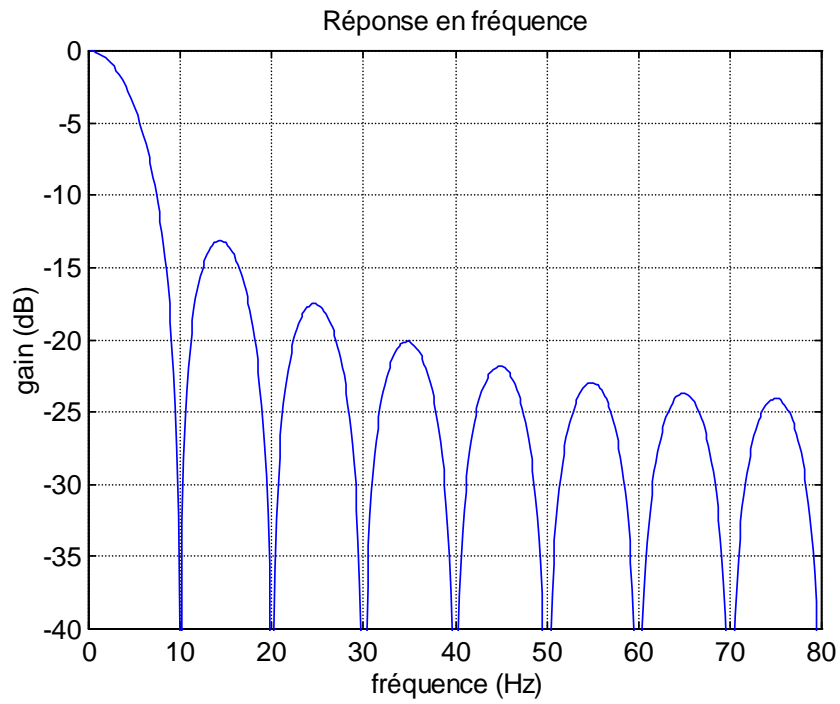
➤  $F_e = 266,67 \text{ Hz}$        $n = 16$



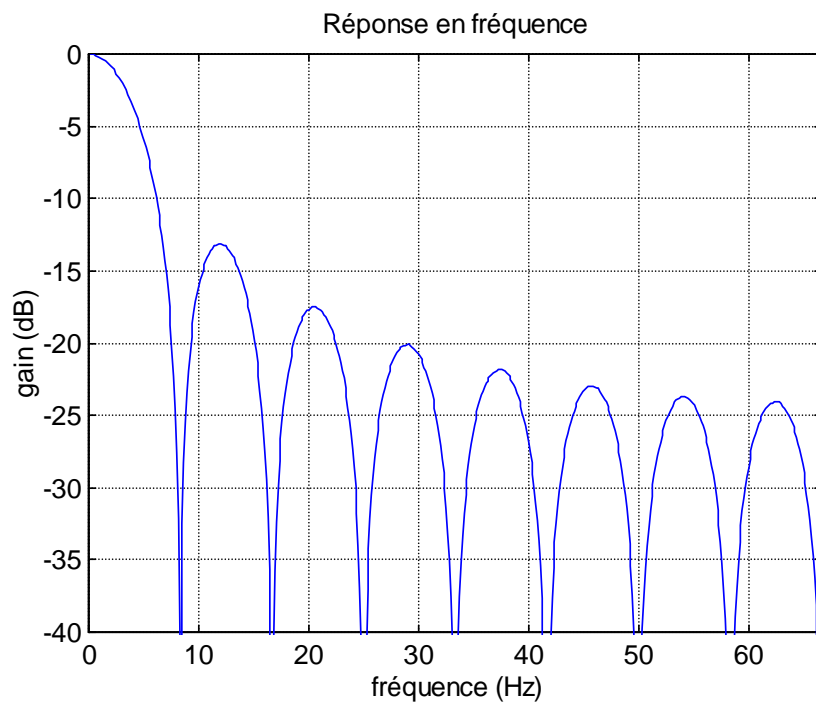
➤  $F_e = 200 \text{ Hz}$        $n = 16$



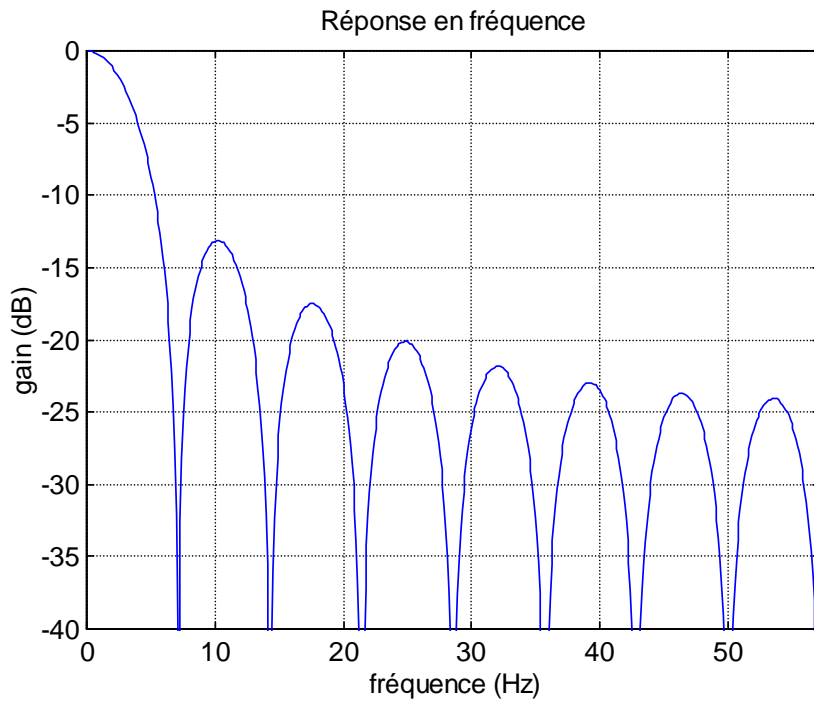
➤  $F_e = 160 \text{ Hz}$        $n = 16$



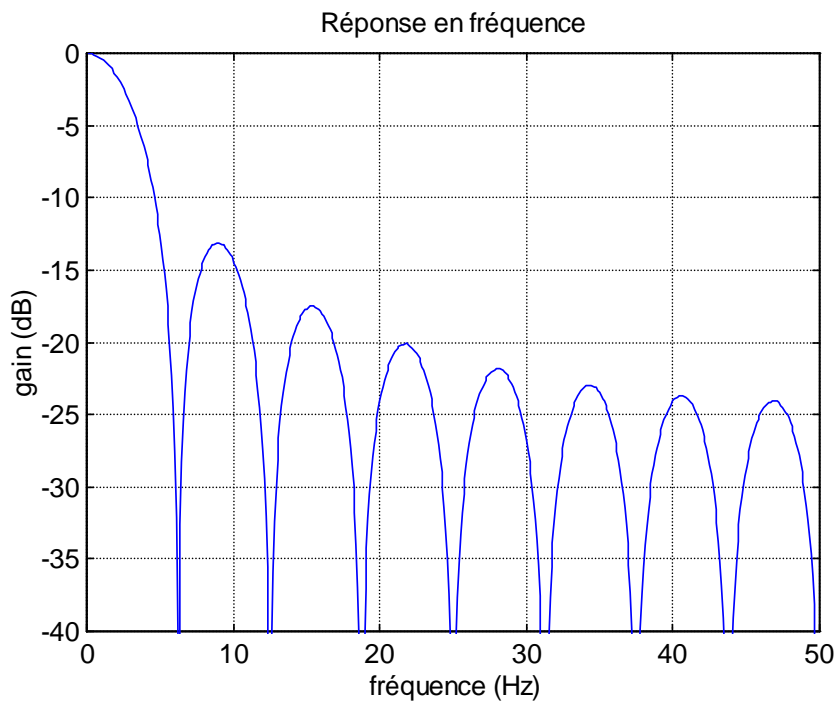
➤  $F_e = 133,33 \text{ Hz}$        $n = 16$



➤  $F_e = 114,29 \text{ Hz}$        $n = 16$



➤  $F_e = 100 \text{ Hz}$        $n = 16$



Pour en savoir plus sur la réjection de mode commun dans les multimètres numériques :

<http://perso.orange.fr/fabrice.sincere/electronique1.htm#5>

## 4- Exemple concret :

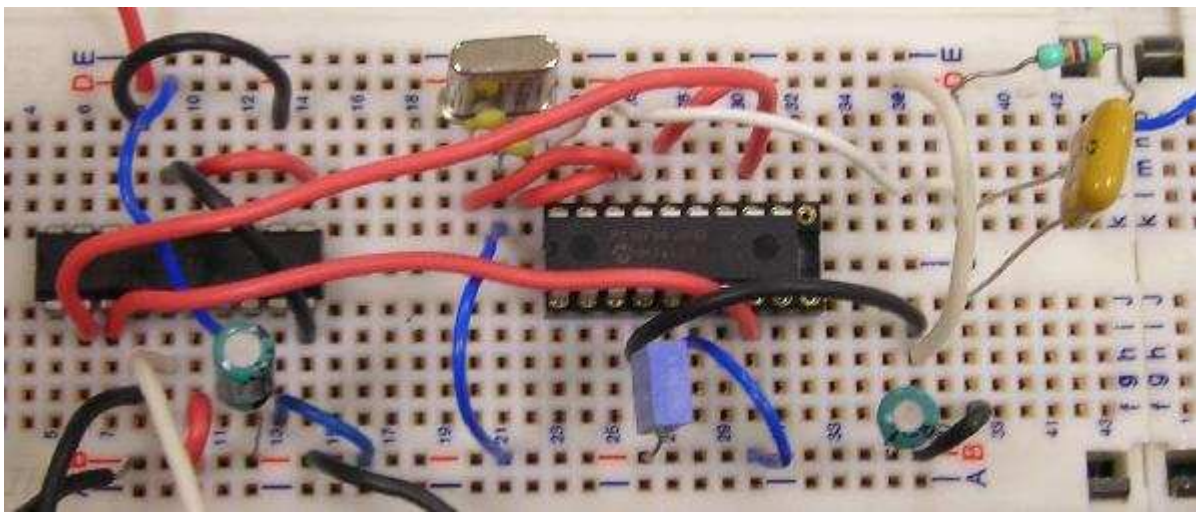
### Carte d'acquisition à 7 entrées analogiques à microcontrôleur PIC 16F88 (Microchip)

Lien vers le projet complet (avec code source du microcontrôleur) :

[http://pagesperso-orange.fr/fabrice.sincere/cm\\_electronique/projet\\_pic/carte%20acquisition%207%20voies/carte\\_acquisition7voies.htm](http://pagesperso-orange.fr/fabrice.sincere/cm_electronique/projet_pic/carte%20acquisition%207%20voies/carte_acquisition7voies.htm)

#### 4-1- Présentation

Il s'agit d'une carte d'acquisition 7 voies analogiques pour signaux lents, qui conviendra pour l'acquisition de températures, de pressions atmosphériques, etc ...



La fréquence d'acquisition est de 1 Hz maximum par voie (c'est-à-dire 60 acquisitions maximum par minute).

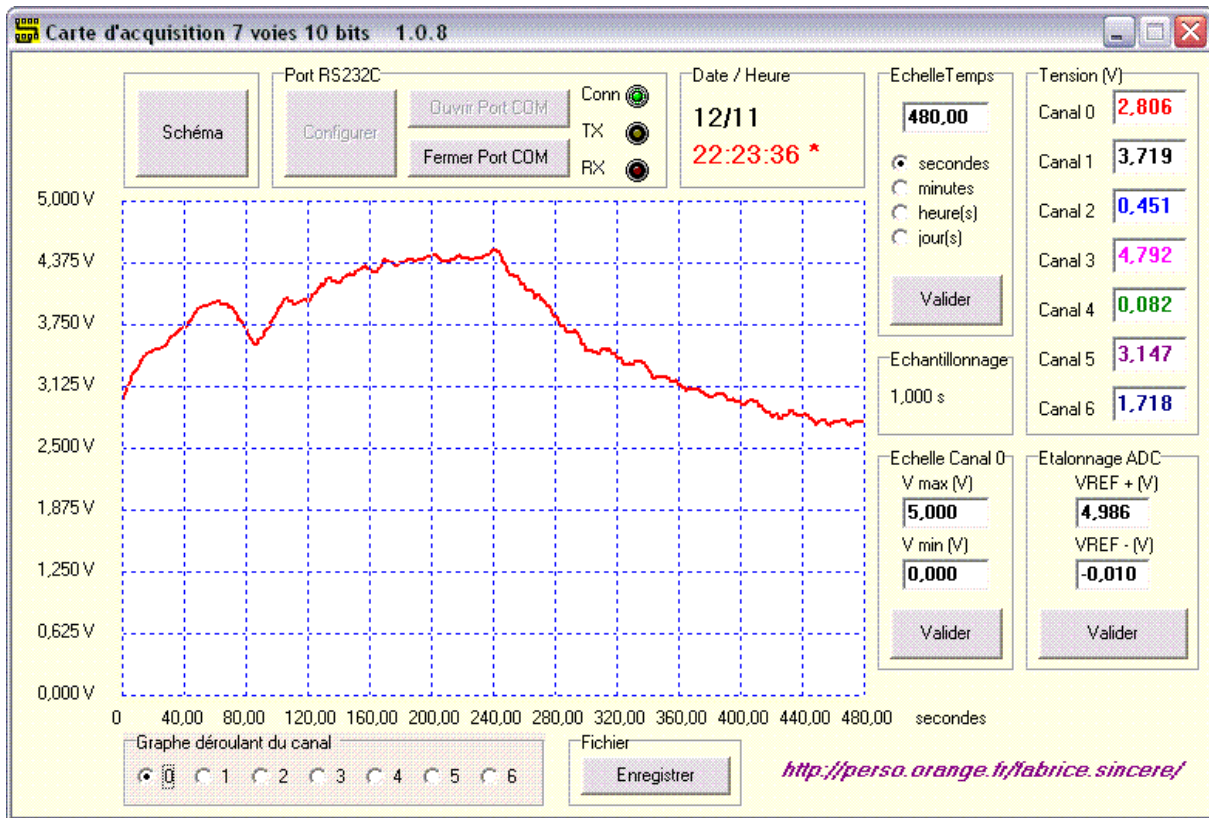
Le convertisseur analogique - numérique (module interne du microcontrôleur PIC 16F88) a les caractéristiques suivantes :

- Plage de tension analogique : 0,000 V à 5,000 V
- Résistance interne de la source de tension à échantillonner : 5 k $\Omega$  maximum
- Résolution numérique : 10 bits (5 mV)

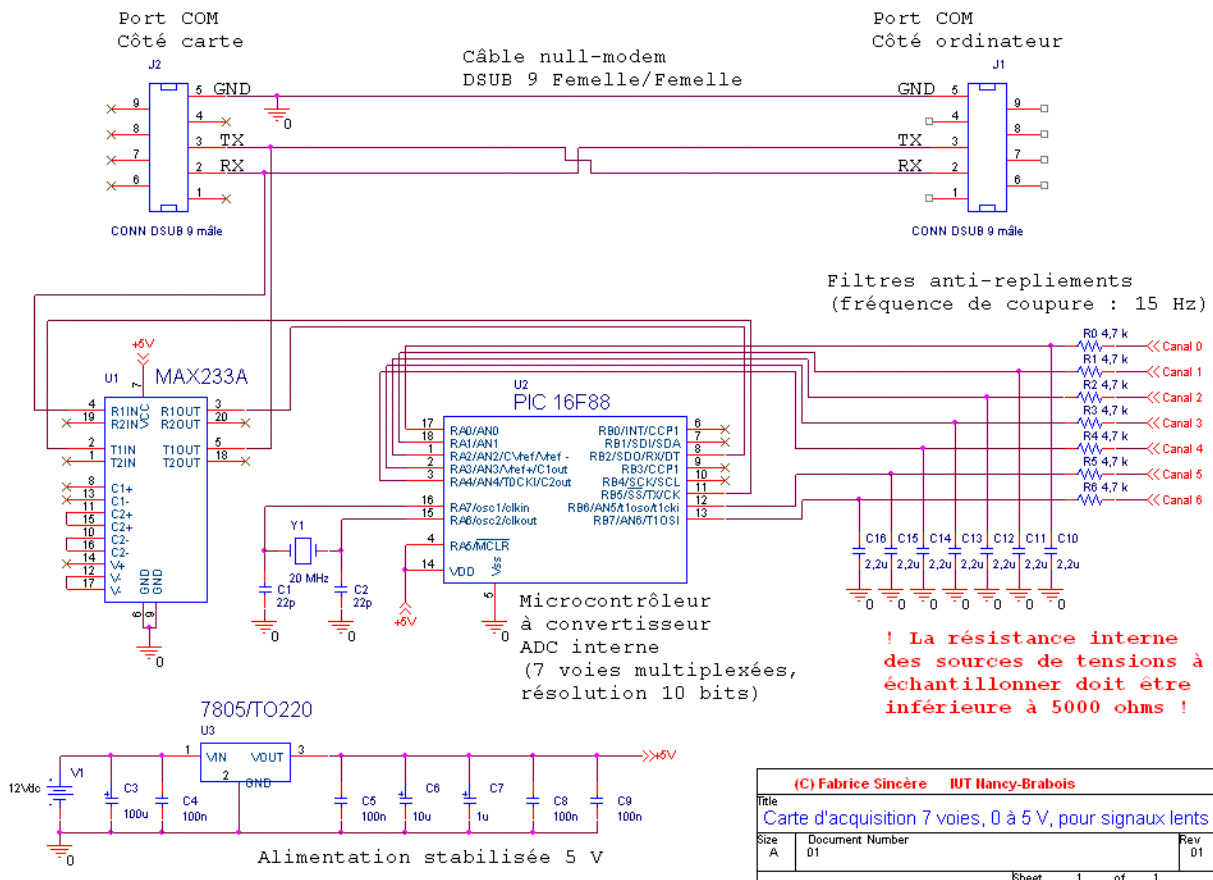
N.B. Le filtrage numérique améliore la résolution : moins de 1 mV en pratique.

La carte se connecte à un ordinateur (via une liaison RS232).

J'ai développé une application Windows qui affiche les données en temps réel et fournit un graphe déroulant (pour chaque voie) :



## 4-2- Schéma électrique de la carte



## 4-3- Principe de fonctionnement de la carte

### 4-3-1- Les filtres anti-repliements

Un filtre anti-repliement est intercalé entre la tension à mesurer (qui provient généralement d'un capteur) et le canal d'entrée de l'ADC du microcontrôleur PIC 16F88.

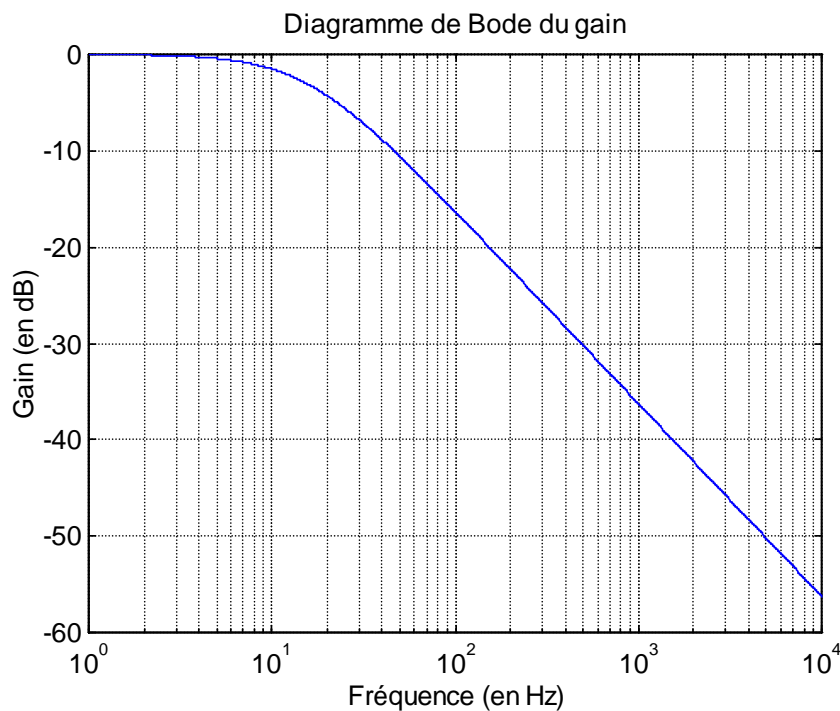
Il s'agit ici d'un simple filtre passif analogique passe-bas du premier ordre :

$$R = 4,7 \text{ k}\Omega$$

$$C = 2,2 \text{ }\mu\text{F}$$

Fréquence de coupure à  $-3 \text{ dB}$  :

$$f_c = \frac{1}{2\pi RC} \approx 15 \text{ Hz}$$



### 4-3-2- Le convertisseur analogique - numérique (ADC)

Le convertisseur analogique - numérique est situé dans un module interne du microcontrôleur PIC 16F88.

Pour plus d'informations sur le fonctionnement du convertisseur analogique - numérique du PIC 16F88, je vous invite à suivre ce lien :

[http://pagesperso-orange.fr/fabrice.sincere/cm\\_electronique/projet\\_pic/aidememoire/16F88\\_ADC/ADC\\_16F88.htm](http://pagesperso-orange.fr/fabrice.sincere/cm_electronique/projet_pic/aidememoire/16F88_ADC/ADC_16F88.htm)



### 4-3-3- Le filtrage numérique

Un filtrage numérique (passe-bas) est effectué de façon à enlever le maximum de bruit (en particulier le bruit de ronflement du secteur 50 Hz).

Ce filtrage numérique est réalisé de manière logicielle par le microcontrôleur PIC 16F88 (pendant la phase de traitement du résultat de la conversion de l'ADC).

- Caractéristiques du filtre numérique :

- Filtre numérique non récursif à moyenne glissante (sur 256 échantillons)
- Période d'échantillonnage nominale :  $T_e = 390,625 \mu\text{s}$
- Fréquence d'échantillonnage nominale :  $F_e = 1 / T_e = 2,560\ 000 \text{ kHz}$

- Choix de la fréquence d'échantillonnage du filtre :

La fréquence d'acquisition est de 1 Hz maximum par voie.  
Comme il y a 7 voies, cela laisse 140 ms entre deux acquisitions.

Il faut que  $nT_e$  soit un multiple entier de la période du secteur (20 ms), tout en restant inférieur à 140 ms :

On prendra :  $nT_e = 100 \text{ ms}$  (soit 5 fois la période du secteur).

On a intérêt à choisir n le plus grand possible.

$T_e$  est limitée par la rapidité du convertisseur ADC interne du microcontrôleur PIC 16F88.  
Dans notre cas, la durée de conversion est de 42  $\mu\text{s}$ .

On prendra :  $n = 256$  (puissance de 2)

D'où :

$$T_e = 100 \text{ ms} / 256 = 390,625 \mu\text{s} \text{ (ce qui est bien supérieur à } 42 \mu\text{s)}$$

$$F_e = 2560,000 \text{ Hz}$$

L'horloge du microcontrôleur PIC 16F88 est cadencée par un quartz de fréquence nominale 20 MHz : 1 cycle correspond donc à 0,2  $\mu\text{s}$ .

$$390,625 \mu\text{s} = 1953,125 \text{ cycles}$$

On arrondit à 1953 cycles :

$$T_e = 390,6 \mu\text{s}$$

$$F_e = 2560,16 \text{ Hz}$$

La tolérance sur le quartz est  $\pm 100 \text{ ppm}$ .

Finalement :  $F_e = (2560,16 \pm 0,26) \text{ Hz}$

La tolérance sur la fréquence d'échantillonnage peut être négligée.

Par la suite, on retiendra la valeur de 2560 Hz.

- Fréquences rejetées par le filtre numérique :

$$f = k \times 2560 / 256 \quad \text{avec } k \text{ entier}$$
$$f = k \times 10 \text{ Hz}$$

Tous les multiples entiers de 10 Hz sont complètement filtrés par le filtre numérique :

10, 20, 30, 40, **50**, 60, .... **100**, ...., **150** Hz ...

On élimine ainsi les harmoniques du secteur (50 Hz).

N.B. Les multiples de 60 Hz sont également éliminés (on rappelle que 60 Hz est la fréquence du secteur aux USA et en Grande Bretagne).

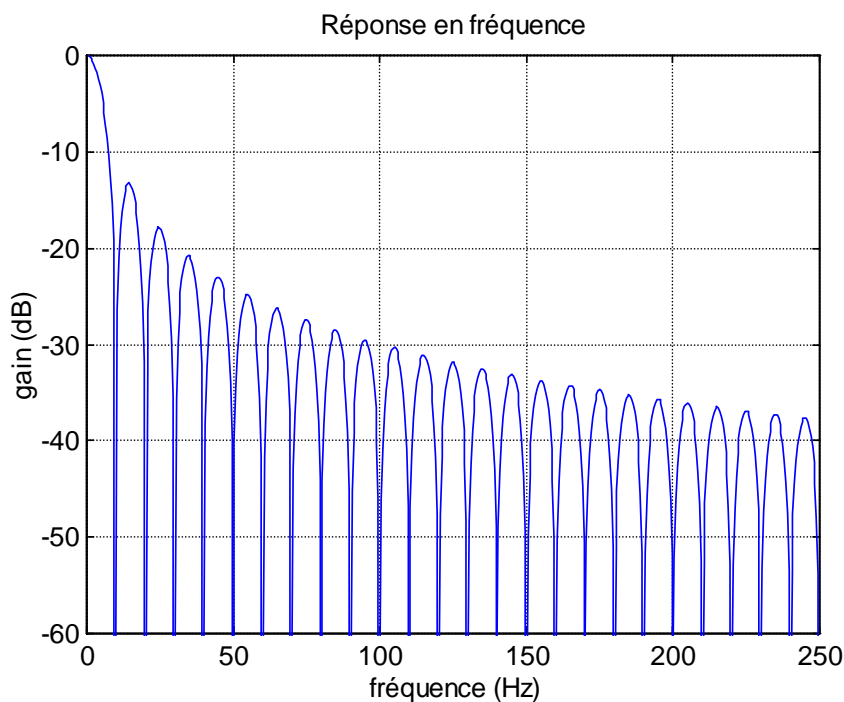
$$nT_e = 100 \text{ ms}$$

= 5 fois la période du secteur (50 Hz en France)  
= 6 fois la période du secteur (60 Hz aux USA)

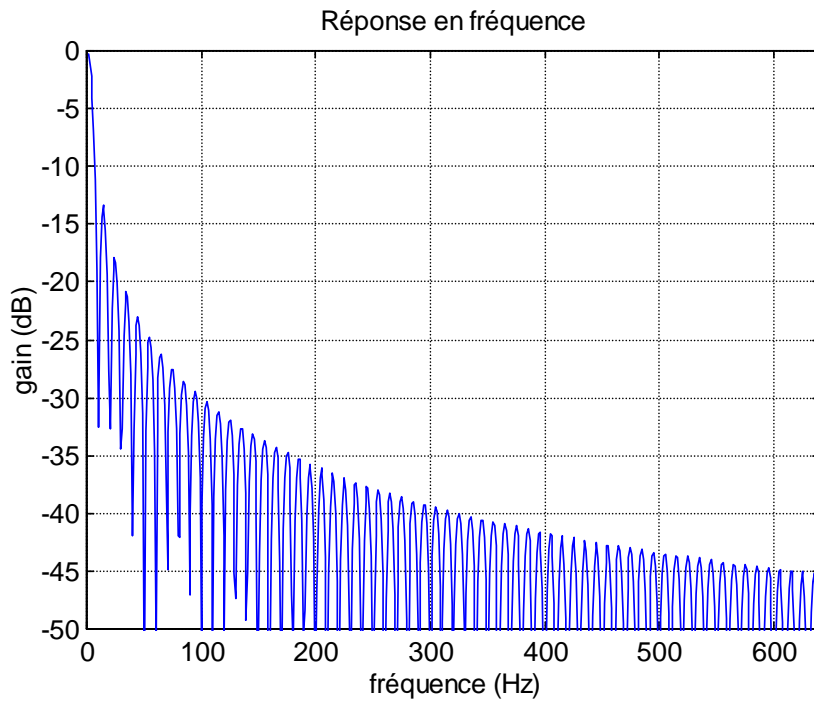
- Réponse en fréquence du filtre numérique

$$F_e = 2560 \text{ Hz} \quad n = 256 \text{ échantillons}$$

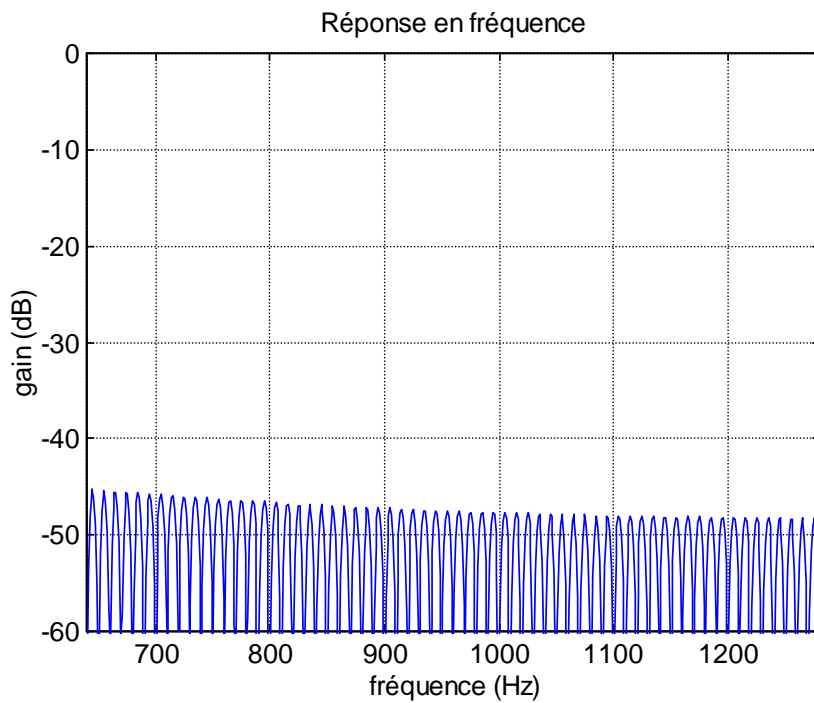
Zoom de la plage 0 à 250 Hz :



Plage 0 à 640 Hz :



Plage 640 Hz à  $F_c / 2 = 1280$  Hz :



- Résultats expérimentaux :

- Sans filtre anti-repliement

En entrée de l'ADC, on applique un signal sinusoïdal étalon de valeur moyenne 2,500 V et d'amplitude crête à crête 4,000 V.

➤  $f = 15 \text{ Hz}$

$$\text{Gain expérimental : } 20\log_{10}\frac{2,925-2,075}{4} = 20\log_{10}\frac{0,85}{4} = -13,5 \text{ dB}$$

En théorie : -13,2 dB (Cf. courbe ci-dessus)

➤  $f = 45 \text{ Hz}$

$$\text{Gain expérimental : } 20\log_{10}\frac{2,642-2,358}{4} = 20\log_{10}\frac{0,284}{4} = -23,0 \text{ dB}$$

En théorie : - 23,0 dB (Cf. courbe ci-dessus)

➤  $f = 55 \text{ Hz}$

$$\text{Gain expérimental : } 20\log_{10}\frac{2,616-2,384}{4} = 20\log_{10}\frac{0,232}{4} = -24,7 \text{ dB}$$

En théorie : - 24,7 dB (Cf. courbe ci-dessus)

Ca marche parfaitement !

➤  $f = 50 \text{ Hz}$

$$\text{Gain expérimental : } 20\log_{10}\frac{2,501-2,499}{4} = 20\log_{10}\frac{0,002}{4} = -66 \text{ dB}$$

En théorie : gain -  $\infty$

➤  $f = F_e = 2560 \text{ Hz}$

$$\text{Gain expérimental : } 20\log_{10}\frac{4,50-0,50}{4} = 20\log_{10}\frac{4}{4} = 0 \text{ dB}$$

En théorie : 0 dB (comme pour 0 Hz)

➤  $f = 2575 \text{ Hz}$

$$\text{Gain expérimental : } 20\log_{10}\frac{2,89-2,11}{4} = 20\log_{10}\frac{0,78}{4} = -14,2 \text{ dB}$$

En théorie : -13,2 dB (comme pour  $2575 - 2560 = 15 \text{ Hz}$ )

➤  $f = 2585 \text{ Hz}$

$$\text{Gain expérimental : } 20\log_{10}\frac{2,74-2,26}{4} = 20\log_{10}\frac{0,48}{4} = -18,4 \text{ dB}$$

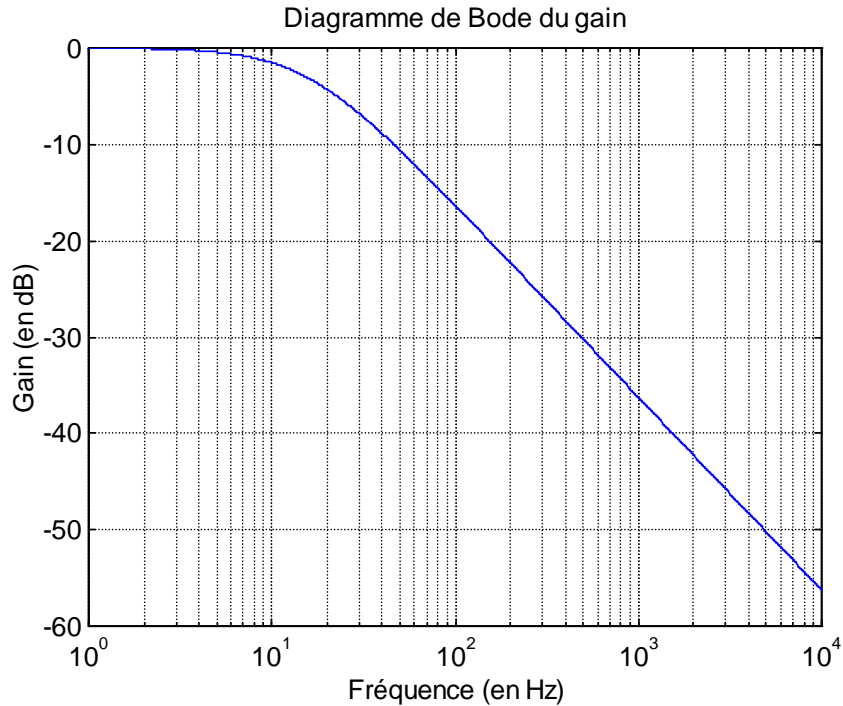
En théorie : -17,8 dB (comme pour  $2585 - 2560 = 25 \text{ Hz}$ )

- Avec filtre anti-repliement

La fréquence de coupure du filtre anti-repliement est de 15 Hz.  
 Cette fréquence a été choisie de la manière suivante :

>> 1 Hz (fréquence d'acquisition)

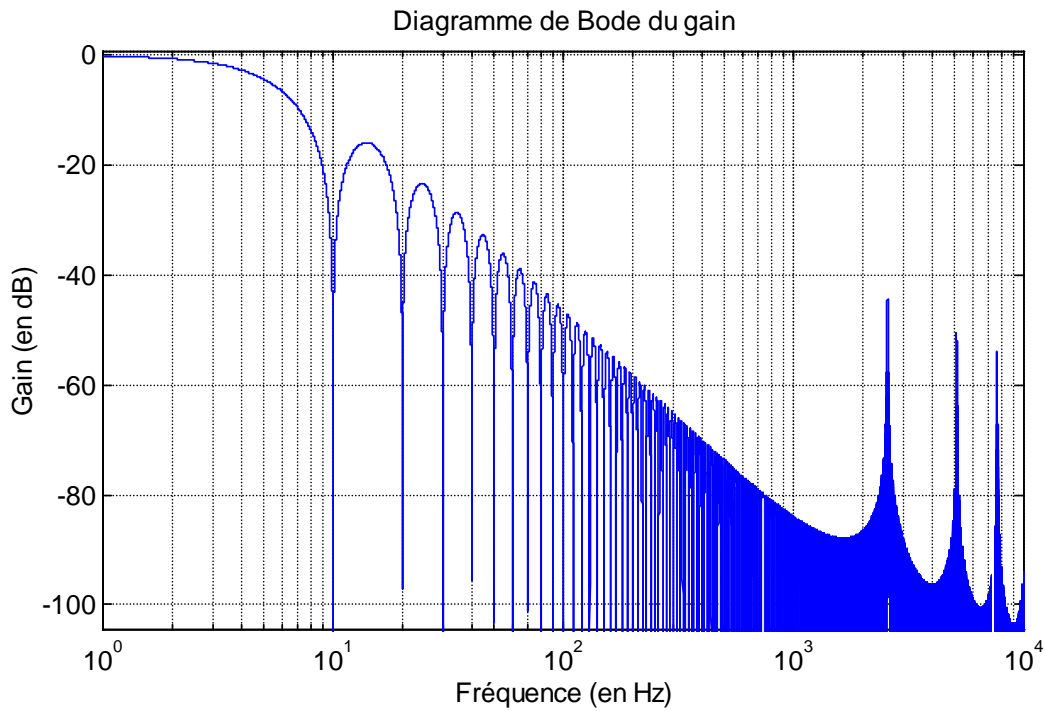
<< 1260 Hz ( $F_e / 2$ )



En théorie, le gain du filtre anti-repliement est :

- Pour  $f = F_e / 2 = 1280$  Hz : - 38,4 dB (Cf. courbe ci-dessus)
- $f = F_e = 2560$  Hz - 44,4 dB
- $f = 2 F_e = 5120$  Hz - 50,4 dB
- $f = 3 F_e = 7680$  Hz - 54,0 dB
- ...

Le gain total est égal à la somme du gain du filtre anti-repliement et du gain du filtre numérique, ce qui donne en théorie :



En pratique, on mesure :

➤  $f = 2560 \text{ Hz}$

$$\text{Gain expérimental : } 20\log_{10} \frac{2,511 - 2,487}{4} = 20\log_{10} \frac{0,024}{4} = -44,4 \text{ dB}$$

$$\text{En théorie : } -44,4 + 0 = -44,4 \text{ dB}$$

➤  $f = 5120 \text{ Hz}$

$$\text{Gain expérimental : } 20\log_{10} \frac{2,506 - 2,494}{4} = 20\log_{10} \frac{0,012}{4} = -50,5 \text{ dB}$$

$$\text{En théorie : } -50,4 + 0 = -50,4 \text{ dB}$$

Les résultats expérimentaux coïncident avec la théorie.

#### 4-4- La liaison RS232 (à titre d'information)

##### 4-4-1- Le câble de communication entre l'ordinateur et la carte

Il faut bien sûr un ordinateur qui possède un port COM, reconnaissable par son connecteur SubD 9 broches mâle.

Il faut brancher un câble "null-modem" (câble croisé) entre l'ordinateur et la carte.

Si vous n'en avez pas, vous pouvez facilement en faire un (il faut 3 fils et 2 connecteurs SubD 9 broches femelle).

#### 4-4-2- Protocole de communication entre l'ordinateur et la carte

Seul l'ordinateur prend la parole (maître).

Le microcontrôleur PIC 16F88 se contente de répondre (esclave).

L'ordinateur envoie 1 octet (via la liaison RS232) toute les 140 ms.

Cet octet contient le numéro du canal à échantillonner :

- o canal 0 -> 0x00
- o canal 1 -> 0x01
- o canal 2 -> 0x02
- o canal 3 -> 0x03
- o canal 4 -> 0x04
- o canal 5 -> 0x05
- o canal 6 -> 0x06

Le processus est cyclique : voie 0 puis voie 1 ... puis voie 6 puis voie 0 ...

Chaque voie est ainsi échantillonnée toutes les  $7 \times 140 = 980$  ms (d'où une fréquence d'acquisition d'environ 1 Hz).

Une fois l'octet reçu, le PIC 16F88 sélectionne le canal indiqué, et lance une série de 256 conversions (une toute les 390,6  $\mu$ s).

Cela prend :  $256 \times 390,6 \mu\text{s} = 100$  ms

Le microcontrôleur PIC effectue la moyenne des 256 conversions (filtrage numérique).

Sachant que la résolution du convertisseur ADC est 10 bits, la somme nécessite un nombre de 18 bits.

La moyenne correspond à la somme avec la virgule décalée de 8 bits vers la gauche.

Par exemple :

Somme brute (18 bits) : 10 10010010 01000101 = 168 517 (en décimal).

On décale la virgule de 8 positions :

10 10010010 , 01000101

Partie entière (10 bits) : 10 10010010 = 658 (en décimal)

01000101 =  $1/4 + 1/64 + 1/256 = 0,26953125$

En définitive :

$168\ 517 / 256 = 658,26953125$

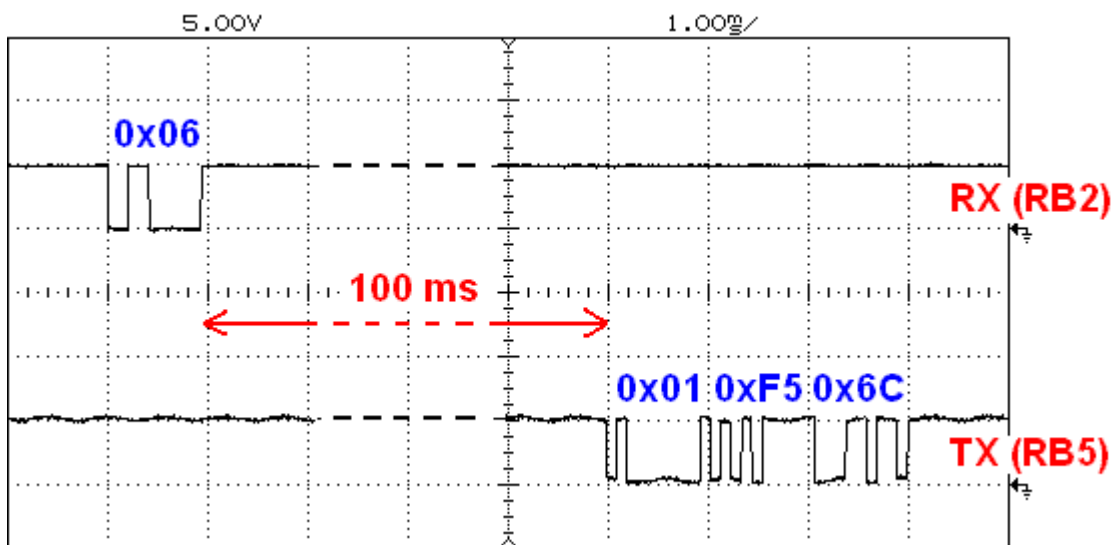
Notez que la résolution est améliorée : on passe de 10 bits à 18 bits.

On pratique, on restera prudent, et on considérera que la résolution est d'au moins 12 bits (soit une résolution de l'ordre de  $5 \text{ V} / 4096 \approx 1 \text{ mV}$ ).

La moyenne est envoyée telle quelle vers l'ordinateur sous la forme de 3 octets :

- 1er octet : (000000 D9 D8)
- 2ème octet : (D7 ... D0)
- 3ème octet : (D-1 ... D-8)

Exemple d'oscillogramme :



L'ordinateur envoie vers le PIC un octet :

0x06  $\Leftrightarrow$  voie 6

100 ms plus tard, le PIC renvoie 3 octets :

0x01 0xF5 0x6C  $\Leftrightarrow$  0x01F56C  $\Leftrightarrow$  128 364 (en décimal)

$128\,364 / 256 = 501,42 \Leftrightarrow (501,42 / 1023) \times 5 \text{ volts} = \mathbf{2,451 \text{ volts}}$

40 ms plus tard ( $140 - 100 = 40 \text{ ms}$ ), l'ordinateur envoie un nouvel octet (0x00) pour demander l'échantillonnage de la voie 0, et le cycle recommence ...

Lien vers le projet complet (avec code source du microcontrôleur) :

[http://pagesperso-orange.fr/fabrice.sincere/cm\\_electronique/projet\\_pic/carte%20acquisition%207%20voies/carte\\_acquisition7voies.htm](http://pagesperso-orange.fr/fabrice.sincere/cm_electronique/projet_pic/carte%20acquisition%207%20voies/carte_acquisition7voies.htm)

© *Fabrice Sincère*