

## SPECIFICATIONS DTMF

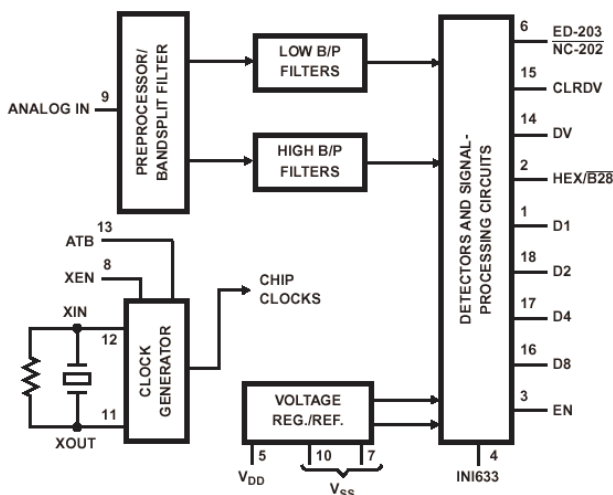
CCITT Recommendations Q.23 et Q.24 (valeurs AT&T) :

1. Fréquences
  - a. Groupe bas : 697, 770, 852, 941 Hz
  - b. Groupe haut : 1209, 1336, 1477, 1633 Hz
2. Tolérance sur la fréquence
  - a. ON : 1.5%, si une fréquence est détectée à mieux que +/- 1.5% elle est valide
  - b. OFF : 3.5%, si une fréquence est détectée au-delà de +/- 3.5% elle est invalide
3. Durées
  - a. Durée signal détecté ON : >40 ms
  - b. Durée signal détecté OFF: <23 ms
  - c. Durée de la pause : 40 ms max.
  - d. Interruption du signal : 10 ms min.
  - e. Rapidité de numérotation : 93ms/digit min.
4. Différence d'amplitude ; H/L = +4 dB max / - 8 dB min
5. Force du signal
  - a. Rapport signal sur bruit : 15 dB minimum (Amplitude du signal 5.6 fois plus forte que celle du bruit)
  - b. Puissance du signal : -26 dBm minimum (38 mVeff)

## METHODES DE DETECTION DTMF

### Détection avec un circuit spécialisé

Le plus direct et le plus sûr, consiste à utiliser un décodeur intégré.



On utilise un circuit SSI202P, ou un CD22202,

Si le microcontrôleur peut être cadencé à 3.5795 MHz, on pourra partager l'horloge.

La détection est directe, il suffit de surveiller le signal DV Data Valid et de lire les 4 bits D1 à D4 lorsque DV passe de 0 à 1.

Il n'est pas nécessaire d'utiliser le signal CLRDRV.

Les 4 bits sortis sont décodés, mais une anomalie est présente sur le 0 et les codes A,B,C,D. Le microcontrôleur devra faire un transcodage pour retrouver le code ASCII du code reçu, et donc directement affichable.

Code reçu	sortie décodeur 202	
	Hexa (Hex=1)	2 sur 8 (Hex=0)
1	1	0
2	2	1
3	3	2
4	4	4
5	5	5
6	6	6
7	7	8
8	8	9
9	9	A
0	A	D
*	B	C
#	C	E
A	D	3
B	E	7
C	F	B
D	0	F

## Détection par traitement de signal

### Transformées de Fourier Discrettes (DFT)

Considérant une séquence de N échantillons, la DFT échantillonne uniformément l'espace fréquentiel avec N raies, équidistantes, localisées aux fréquences :  $kF_{\text{sampling}}/N$  avec k entier de 0 jusqu'à N-1. Pour avoir suffisamment de précision en fréquence, il faudrait que N soit très grand (plusieurs milliers), et ceci est totalement incompatible avec l'exigence d'une détection en quelques dizaines de ms.

### Algorithme de Goerzel

Le principe est de calculer le coefficient DFT sur une raie particulière au lieu de calculer les N raies. C'est l'algorithme de Goerzel qui est basé sur un filtre à réponse impulsionnelle infinie (IIR), donc avec un algorithme à boucle récursive.

$$S_k(n) = x(n) + 2 \cos\left(\frac{2\pi k}{N}\right) S_k(n-1) - S_k(n-2) \quad \text{Avec } S_k(-2) = 0, S_k(-1) = 0$$

Tel quelle, et pour les mêmes raisons que la DFT, la précision fréquentielle est médiocre pour un nombre d'échantillons faible ; par contre en forçant  $k = N \times F_{\text{dtmf}} / F_{\text{sampling}}$  non entier, on force le filtre IIR à travailler sur une fréquence précise. En faisant cela, la DFT devient « non uniforme », la détection de la fréquence sera parfaite, mais attention, la largeur de la raie « k » vaut toujours  $F_{\text{sampling}}/N$ . A chaque échantillon acquis on calculera :

$$S_k(n) = x(n) + 2 \cos(2\pi \times F_{\text{dtmf}} / F_{\text{sampling}}) S_k(n-1) - S_k(n-2) \quad \text{Avec } S_k(-2) = 0, S_k(-1) = 0$$

Lorsqu'on a acquis N échantillons, l'algorithme calcule la puissance de la seule raie « k » avec l'expression :

$$|Y(k)|^2 = S_k^2(N) - 2 \cos(2\pi \times F_{\text{dtmf}} / F_{\text{sampling}}) S_k(N) S_k(N-1) + S_k^2(N-1)$$

Le nombre d'échantillons  $N$  doit être pris bien plus petit que la durée valide DTMF (40 ms, soit 320 échantillons), mais il conditionne aussi la largeur de l'analyse  $F_s/N$ .

D'abord il faut voir que 1.5% de la plus basse fréquence représente 10Hz, alors que 3.5% de la plus haute représente 57Hz, la gamme est large, pas facile de trouver un  $F_{sampling}/N$  correct.

Mais aussi, il faut pouvoir accepter un  $1633\pm 24\text{Hz}$  ( $\pm 1.5\%$ ) en même temps qu'on rejette un  $697\pm 24\text{Hz}$  ( $\pm 3.5\%$ ).

Dans l'idéal il faudrait faire varier la longueur  $N$  de calcul pour chaque fréquence traitée.

Certains prennent 2 fois plus d'échantillons pour le groupe bas par rapport au groupe haut.

	1,50%	3,50%
697	10	24
770	12	27
852	13	30
941	14	33
1209	18	42
1336	20	47
1477	22	52
1633	24	57

### **Détection par corrélation directe**

Puisqu'on connaît les fréquences à détecter, et que celles ci sont assez précises, on peut utiliser une corrélation directe, qui consiste à réaliser l'opération :

$$\int_0^T f(t) \cdot A_i \cdot \sin(\omega_i t + \varphi) \cdot dt \quad \text{avec} \quad f(t) = A_a \cdot \sin(\omega_a t) + A_b \cdot \sin(\omega_b t)$$

Si le signal d'entrée contient la pulsation  $\omega$ , alors la sortie est un signal continu qui vaut  $A_i A_b \cos(\varphi)/2$ ,  $\varphi$  étant le déphasage entre le signal de corrélation et la pulsation à détecter. Si on fait la même opération avec le cosinus, on obtient en sortie un signal continu qui vaut  $A_i A_b \sin(\varphi)/2$ . Il suffit de faire la somme des carrés de ces 2 sorties pour obtenir une valeur indépendante des phases  $A_i^2 A_b^2 / 4$  à condition que la période d'intégration soit grande devant les pulsations.

La méthode est insensible à la composante continue du signal à l'entrée.

La somme des carrés est évidemment proportionnelle au carré de l'amplitude du signal à l'entrée. Et pour détecter les groupes de fréquences, on aura intérêt à résonner en logarithme.

Si la pulsation recherchée n'est pas présente dans le signal, la sortie est nulle, en théorie.

Si la fréquence de référence est un peu différente de la fréquence recherchée, le signal de sortie

cosinus vaut  $A_i A_a / 2 \cdot \int_0^T \cos((\omega - \omega_a)t + \varphi) \cdot dt$  et cette expression est assez vite dépendante de la période d'intégration.

La période d'intégration doit évidemment être grande devant la période des fréquences cherchés, mais pas trop pour respecter la norme DTMF (durée de détection entre 23ms et 40ms) et afin que la sélectivité ne soit pas trop forte. 32 ms semble un bon compromis, cela correspond à 256 échantillons à 8kHz et à une résolution de l'ordre de 16Hz.

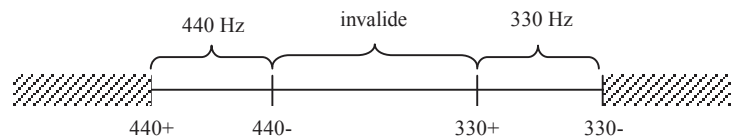
## DETECTION DES TONALITES

### *Détection par circuit spécialisé*

Bien sûr, on pense rapidement à l'utilisation d'une PLL, genre NE567, mais si ces circuits fonctionnent très bien, ils consomment beaucoup et c'est un peu encombrant.

### *Détecter par mesure de période*

L'algorithme de détection est très simple. On mesure N fois la période, puis on compare le résultat à un gabarit.



Le logiciel utilise une routine de détection des fronts identique à celle du V23 par mesure de temps, mais le programme est plus simple. On prendra des seuils avec des tolérances sont assez resserrées, par exemple +/- 5%.

Le test se fera sur la période entière, c'est à dire sur deux mesures consécutives de demi périodes.

On ajustera le nombre de périodes seuil (pour décider d'une détection) via un délai de réaction, par exemple 50ms soit 24 périodes consécutives de 440 Hz, ou 18 périodes consécutives de 330 Hz

### *Détecter par corrélation*

Puisqu'on connaît exactement la fréquence à détecter, on peut utiliser une corrélation, comme pour détecter des signaux DTMF.

## DEVELOPPEMENT D'UNE CORRELATION DIRECTE

### Algorithmme

Le programme principal est une boucle activée chaque fois qu'un échantillon est disponible, tout les  $1/F_s$  seconde (125  $\mu$ s à 8 keps). On peut aussi imaginer acquérir N échantillons puis les traiter ensuite, si on dispose de la mémoire nécessaire.

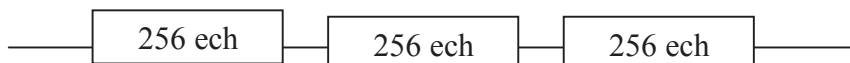
Pour chaque fréquence discrète F à détecter :

- Calculer la nouvelle phase  $\varphi_{i,j+1} = \varphi_{i,j} + \delta\varphi_i$  ; l'incrément de phase  $\delta\varphi_i = \text{fullscale} * \frac{F}{F_s}$  est une constante. Il semble important de travailler sur une phase précise, sur 16 bits par exemple. Le fullscale vaudra donc 65536, la phase parcourt le domaine 0x0000:0xFFFF. Par convention, on dit que 0x0000 sera la phase nulle, et 0x10000 sera 360°. Donc 90° ou  $\pi/2$  sera représenté par 0x4000
- Index = high(phase) en annulant le bit fort pour rester le domaine 0-pi
- X = Sinus[Index] ; on récupère un octet de 0 à 127
- Si le bit 15 de la phase est 1, X=-X
- CumulSin[i] = CumulSin [i] + X \* echantillon
- Index = high(phase+0x4000) en annulant le bit fort pour rester le domaine 0-pi
- X = Sinus[Index] ; on récupère un octet de 0 à 127
- Si le bit 15 de la phase est 1, X=-X
- CumulCos[i] = CumulCos [i] + X \* echantillon

Au bout des N échantillons :

Les cumul ont pour valeur maximale  $Nech * X_{max} * Sin_{Max}/2 = 256 * 127 * 127/2 = 0x1F8080$  (en fait 0x1F8080 ou 0xE07F80)

- Prendre la valeur absolue de Cumul pour ne plus s'embêter avec le signe
- Normer décalant de 3 bits vers la gauche (la valeur max devient FC0400) et ne conserver que l'octet poids fort.
- Calculer Force[i] = CumulCos[i]\*CumulCos[i] + CumulSin[i]\*CumulSin[i]  
la valeur maximale de Force[i] est le carré du max du CumulCos ou CumulSin (0xFC) donc = 0x00F810, déjà remarquablement normalisé sur le deuxième octet.
- Calculer le log\_Force[i], par un simple décalage logique pour trouver la position du bit de poids fort. On aura 16 pour une valeur  $\geq 0x8000$ , 15 pour ]0x8000:0x4000], 14 pour ]0x4000:0x2000] etc jusqu'à 1 pour ]0x0002:0x00001], et enfin 0 pour une valeur nulle. Finalement, un niveau log correspond exactement à 3dB sur le signal d'entrée. C'est magique.
- Chercher Max\_low, maximum des 4 fréquences basses
  - Vérifier qu'il est supérieur au niveau mini
  - Vérifier que l'écart avec le 2ème niveau est supérieur au contraste mini
- Calculer Max\_high, maximum des 4 fréquences hautes, supérieur au niveau mini
  - Vérifier qu'il est supérieur au niveau mini
  - Vérifier que l'écart avec le 2ème niveau est supérieur au contraste mini
- Vérifier Max\_low et Max\_high sont à moins de 2 niveaux d'écart (6dB) ; en Anglais, on parle du twist entre les 2 fréquences, les specs des circuits spécialisés sont à +/-8dB à +/- 10dB
  - Décider code\_on[i,j] = 1 si log\_Force[i] = Max\_grp\_low et log\_Force[j] = Max\_grp\_high



**Compensation de la valeur moyenne et contrôle de gain**

L’algorithme de corrélation n’est pas sensible à la composante continue, et on peut disposer d’un convertisseur mieux résolu que la précision nécessaire.

Donc il serait idiot de ne pas profiter de cette sur-résolution pour avoir une sorte de gain automatique autour de la valeur moyenne.

Il faut d’abord pouvoir mesurer **la valeur moyenne**, la stocker, la retrancher de toutes les mesures, avant d’appliquer un gain. Le plus simple est de faire cette mesure à un moment où il y a peut de chance d’avoir un signal valide, voire forcer un mode calibration en amont de la chaîne analogique.

- Mode mesure : ici les opérations de corrélation sont activées. La valeur moyenne doit être connue, ainsi que le gain à appliquer. Il doit y avoir un signal à l’entrée, et si on calcule la moyenne ici, on risque de commettre une erreur qui sera d’autant plus grande que la période du signal utile sera faible.
- Mode attente : ici les opérations de corrélation ne sont pas activées. On n’attend pas de signal particulier à l’entrée. On peut mesurer une valeur moyenne et un écart max-min en temps réel sur 256 échantillons. Ceux-ci doivent ensuite satisfaire des critères de vraisemblance :
  - La moyenne doit se situer entre Vmin et Vmax, bornes données par la tolérance design, par exemple de 1.80 à 2.50 Volts.
  - L’ écart max-min doit être inférieur au bruit nominal, par exemple 20 mV.
  - La moyenne doit se situer à moins de dV de la précédente, delta donné par la dérive maximale, par exemple 100 mV. Cette dernière condition est difficile à satisfaire pour les transitoires comme la mise en route. De plus elle pourrait être redondante avec la première.

Il faut ensuite pouvoir détecter que le gain n’est pas bon, et qu’il faut le changer. Il faudrait pouvoir mesurer une valeur alternative (RMS ou max-min) en étant sûr qu’on est en présence d’un signal utile et pas d’un parasite. Pour cela, la meilleure façon est encore la corrélation elle-même.

L’algorithme pour augmenter le gain est simple ; dès qu’un signal valide est détecté, on regarde l’amplitude de la corrélation et on décide du gain pour l’acquisition suivante (exemple avec 3 gains 1-2-4 ) :

	Mesure basse ( < 1/5 PEC)	Mesure moyenne	Mesure haute ( > 2/5 PEC)
Gain mini (1)	Gain → 2	OK	Techniquement impossible
Gain moyen (2)	Gain → 4	OK	Gain → 1
Gain haut (4)	Signal trop faible	OK	Gain → 2

PEC : Pleine échelle crête

Là encore, on pourra mettre des critères de vraisemblance :

- Ne changer de gain que si plusieurs acquisitions successives (au moins 2 !) ont donné la même conclusion.

### Implantation réelle

Dans le cas d'une implantation temps réel, l'algorithme doit être réalisé en moins de  $1/F_s$  seconde, soit  $125\mu\text{s}$  @8kHz. Si on veut détecter 8+1, il faudrait faire 9 fois (DTMF et tonalité) les opérations de base (sin ET cos). Ceci laisse moins de  $13\mu\text{s}$  par opération, et c'est peu. Impossible d'implanter cela avec une instruction par  $\mu\text{s}$ , il faudra aller beaucoup plus vite.

La multiplication signée est nécessaire, au moins pour faire l'opération finale

$A^2\cos^2(\varphi)+A^2\sin^2(\varphi)=A^2$  mais aussi pour faire l'opération de corrélation. On pourrait s'en passer sur cette dernière en limitant le nombre de pas des signaux sin et cos, par exemple à +/-2 niveaux (-2, -1, 0, +1, +2). L'algorithme marche encore assez bien.

Si la numérisation est faite de 0 à N-1, avec une valeur au repos de N/2,

si les signaux de référence sin et cos sont numérisés entre 0 et M-1, avec une valeur nulle à M/2,

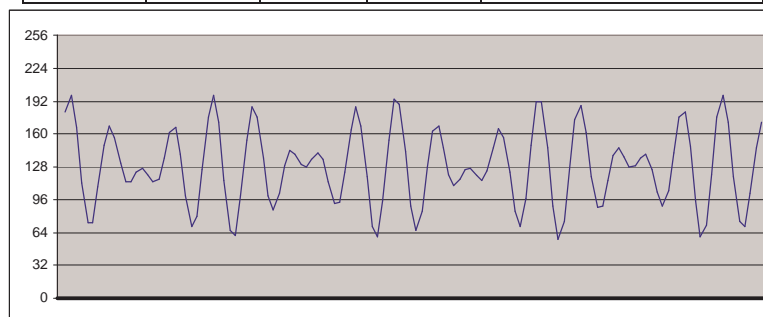
- chaque produit élémentaire signé va de  $-N.M/4$  à  $N.M/4$ , avec une valeur moyenne à 0.
- chaque produit élémentaire non signé va de 0 à N.M, avec une valeur moyenne à  $N.M/4$ . Attention ce n'est plus symétrique. La fonction de corrélation ne marche plus.

La phase sera directement sur une largeur compatible de la table sinus, sans décalage.

### Cas du PIC18F :

Ce  $\mu$ contrôleur est équipé avec un convertisseur ADC 10bits, déclenché par le module CCCP2, et qui génère une interruption en fin de conversion. Le transfert est direct entre 0 et 5 Volts. Le point repos est polarisé à  $V_{dd}/2$ . On dispose donc de 10 bits....

Tension	0.000V	5/4096	2.500V	5.000V x 4095/4096
Code	0x000	0x001	0x200	0x3FF



Ce  $\mu$ contrôleur est équipé avec une multiplication codée non signée

```
MOVWF ARG1,W ;
MULWF ARG2 ; ARG1 * ARG2 -> PRODH:PRODL
```

Elle peut être signée si besoin :

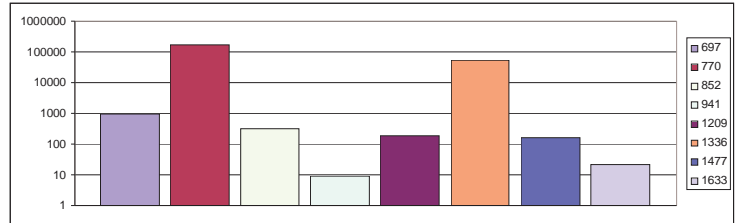
```
MOVWF ARG1,W
MULWF ARG2 ; ARG1 * ARG2 -> PRODH:PRODL
BTFSC ARG2,SB ; Test Sign Bit
SUBWF PRODH, F ; PRODH = PRODH - ARG1
MOVWF ARG2,W
BTFSC ARG1,SB ; Test Sign Bit
SUBWF PRODH,F ; PRODH = PRODH - ARG2
```



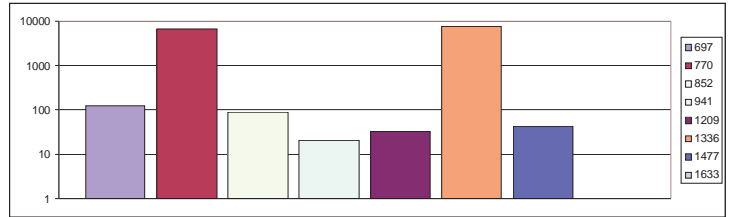
**Essais et mise au point**

**Essai 1, simulation**

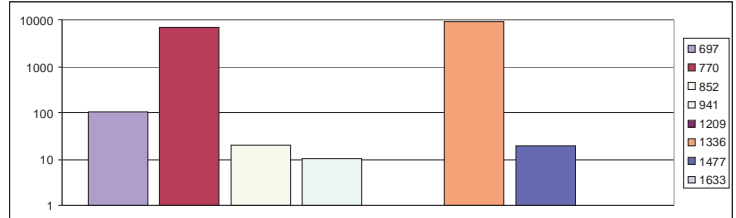
Décodage d'un signal DTMF "6", enregistré en WAV 16kHz, 16bits, puis traité sous EXCEL avec une corrélation sin/cos sur tous les échantillons (1200). Les deux raies concernées (770 et 1336) sont bien visibles.



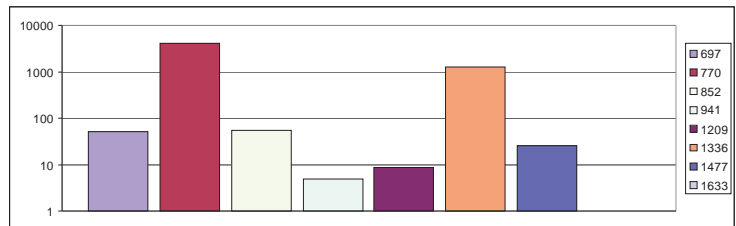
Le même signal en 8kHz, 8 bits, 256 échantillons (32ms). Les résultats sont bons. Le contraste entre la deuxième raie et la suivante est de 50.



Avec une fréquence d'analyse +1% C'est encore meilleur avec un contraste de plus de 60.



Avec une fréquence d'analyse -1% Le contraste tombe à 20.

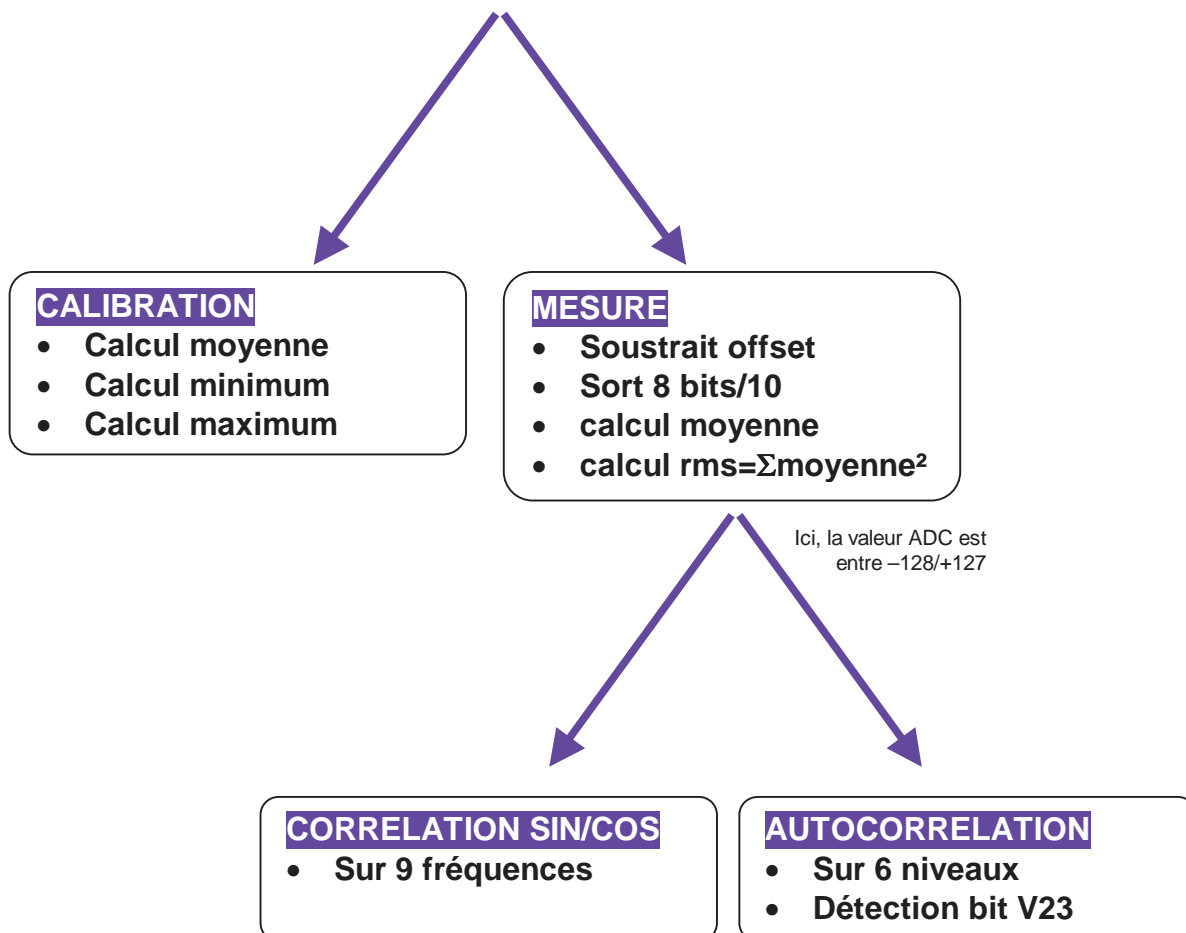


La sensibilité du contraste (moins forte du doublé sur plus forte de la suivante) au nombre d'échantillons est forte :

	Nombre d'échantillons		
	128 16ms	256 32ms	512 64ms
0.98	6	1	5
0.99	13	23	11
1.00	20	50	300
1.01	30	66	187
1.02	30	50	25

*Noter que les valeurs 0.98 sont très faibles car la fréquence recherchée était plutôt à 1.01*

En considérant 256 échantillons, on peut donc spécifier un contraste minimum de l'ordre de 20, soit 26dB.

**Implantation sur PIC18F****Automate ADC**

**Algorithme de corrélation sur 9 fréquences (440Hz et DTMF).**

Optimisation du code pour que les calculs soient les plus rapides possibles ; on arrive à 64.9µs de calcul dans l'interruption ADC cadencée à 125.0µs (8000 Hz). Soit une occupation processeur de 52%. Cela représente à peu près 72 cycles par fréquence recherchée.

Tests avec un générateur BF avec affichage du résultat sur LCD. On constate que la valeur de corrélation est bien proportionnelle au carré de l'amplitude à l'entrée. La pleine échelle est bien retrouvée.

Fréquence	Amplitude RMS	Valeur de corrélation
440	0.195	0x02C0, 00704
440	0.495	0x1260, 04704
440	0.707	0x2600, 09728
440	0.940	0x4400, 17408
440	1.710 (pleine échelle)	0xE400, 52308
431 (440-2%)	0.195	0x0270-0x0280
436 (440-1%)	0.195	0x02D0
444 (440+1%)	0.195	0x0270-0x0280

Test sur ligne réelle

Le tableau suivant donne les forces log2 des fréquences mesurées lors d'une numérotation.

La force du signal RMS est de 09.

La détection sur le groupe haut est très constante, avec un contraste toujours supérieur à 7 niveaux log, soit 21dB.

Code	Groupe bas		Groupe haut	
	Force	Contraste	Force	Contraste
0	00030108	5	02090201	7
0	0103010B	8	040C0402	8
1	0B010002	10	0B000301	8
1	0B010101	10	0B000201	9
2	0A050100	5	030A0000	7
2	09010101	8	01090001	8
4	040B0200	8	0B030001	8
4	05090401	4	0A030100	7
5	040B0503	6	040B0300	7
5	050B0401	6	030B0301	8
6	000A0000	10	03040A01	6
6	05090400	4	01020A00	8
7	00010B05	6	0B010201	9
7	04050904	4	09010000	8
9	04050A04	5	01000B04	7
9	01030B03	8	00030B00	8

## REFERENCES

Cours sur le traitement de signal :

1. [Techniques numériques pour le traitement du signal](#) ; LeRoux, nov 2000

Revue d'algorithmes :

2. [Performance Evaluation and Real-Time Implementation of Subspace, Adaptive, and DFT Algorithms for Multi-tone Detection](#), et aussi [ici](#) et aussi [ici](#) ; Arslany, Evans, Sakaryay, Pinoz, février 1996
3. [DTMF detection, efficient algorithm and implementation of a DTMF detector](#) ; Evans, Felder, mars 1997
4. [Efficient Dual-Tone Multi-Frequency Detection using the Non-Uniform Discrete Fourier Transform](#) ; Felder, Mason, Evans, avril 1998
5. [Embedded signal processing on microcontrollers](#) ; john, Evans, avril 1998
6. [Efficient decoding of digital DTMF and R2 tone signalization](#) ; Popović, dec 2002

Algorithme de Goerztel

7. [Telephone Message Watchdog](#); Zhang, Zhang, Ni, octobre 2004

Corrélation directe :

8. [http://www.parallax.com/dl/docs/prod/sx/simpfskgen1\\_03.zip](http://www.parallax.com/dl/docs/prod/sx/simpfskgen1_03.zip)
9. [Radu Constantinescu](#) qui dit que c'est un Goerztel, alors que c'est une corrélation directe