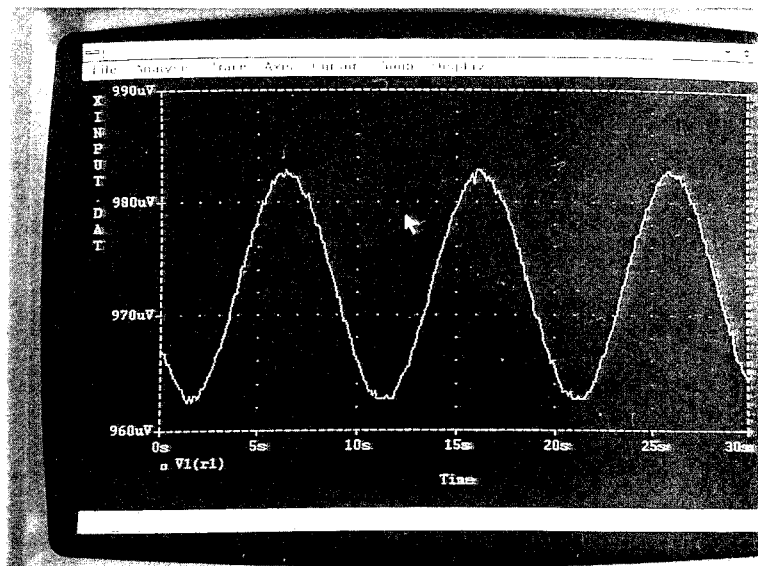


GÉNÉRATION DE SIGNAUX PÉRIODIQUES PAR MICROCONTRÔLEUR

La génération de signaux périodiques est habituellement une tâche qui est dévolue à un processeur ou un périphérique spécialisé. Contrairement à ce que l'on pourrait penser, les microprocesseurs modernes se prêtent bien à la génération de signaux et ce à des fréquences pouvant aller jusqu'à quelques kHz.



L'idée qui vient immédiatement à l'esprit quand on désire générer un signal (pour simplifier, supposons-le rectangulaire) est d'utiliser des compteurs.

On utilise ainsi un compteur que l'on charge avec une valeur proportionnelle à la période que l'on désire obtenir, on décrémente ce compteur jusqu'à ce qu'il atteigne zéro, à ce moment-là on bascule la ligne de sortie de signal, on recharge le compteur et on recommence.

Le temps de décompte étant proportionnel à la valeur initialement chargée, on obtiendra un signal dont la période est variable et dépend de la valeur de chargement. Les microprocesseurs ont souvent des périphériques (Timers et autres Time Processus Unit) qui réalisent la fonction de décompte et de rechargement automatiquement, et qui permettent donc de réaliser ces fonctions très simplement.

Pour simple qu'elle soit, cette technique n'est souvent pas appropriée et ce pour plusieurs raisons :

- Le signal créé a une période proportionnelle à une valeur donnée, ce qui veut dire que l'on aura une bonne précision relative pour les périodes importantes, mais pour les périodes courtes la précision s'effondrera.
- Le temps de boucle dépendant de la valeur de la période, il est donc difficile de générer une salve dont la fréquence pourra être variable mais d'une durée fixe.

Une alternative existe qui permet de résoudre ces problèmes.

Sur la **figure 1**, un additionneur est rebouclé sur lui-même sur une de ses entrées via une mémoire, sur l'autre entrée est appliquée une valeur proportionnelle à la fréquence que l'on veut générer. Supposons que le nombre de bits P de l'additionneur soit 8 et que (pour simplifier les calculs) $F_c = 25,6$ kHz. Supposons encore que la valeur initiale de la mémoire soit 0.

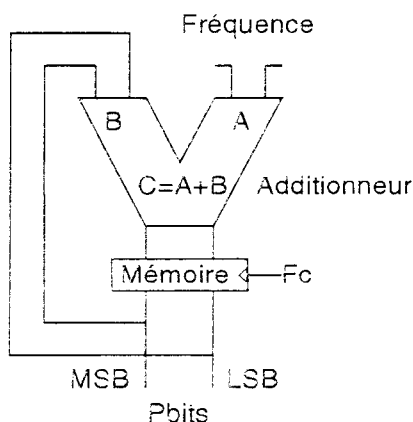


Figure 1 : la technique exploitée avec un additionneur.

Si on met la valeur 1 à l'entrée, on aura en sortie de la mémoire successivement les valeurs 0, 1, 2, ...255, 0, 1...

On fera donc «le tour complet» en 256 coups d'horloge, et donc à $25,6 \text{ kHz} / 256 = 100 \text{ Hz}$.

Si on met la valeur 2 : 0, 2, 4, 6... 254, 0... le tour complet sera donc effectué en $25,6 \text{ kHz} \times 2/256$ soit 200 Hz, et si on met la valeur n :

$$F = 25,6 \text{ kHz} \times n/256 \text{ soit } F = n \times 100 \text{ Hz.}$$

Si on s'intéresse au dernier bit de l'additionneur, on voit qu'il vaut 0 pour les valeurs inférieures à 128 et 1 pour les valeurs supérieures. Il passe donc alternativement de la valeur 0 à 1 à une fréquence proportionnelle à la valeur entrée sur l'additionneur. De la même manière, si on prend l'ensemble des bits disponibles en sortie de l'additionneur, on pourra se servir de la sortie de celui-ci pour servir d'index dans un tableau dans lequel on aura préalablement stocké la forme d'onde que l'on désire générer. Si $P = 16$, le tableau aura 65536 éléments, ce qui dépasse les capacités d'adressage de microprocesseurs 8 bits, aussi on utilisera seulement une partie de la sortie de l'additionneur (les bits de poids forts) comme index dans le tableau. Si on

prend uniquement le dernier bit, on ne pourra générer que des formes rectangulaires, deux bits permettront de définir un signal comportant quatre valeurs différentes, etc.

Nous savons que l'on ne peut pas générer de fréquence supérieure à la moitié de la fréquence d'échantillonnage F_c , les valeurs d'entrée seront donc obligatoirement inférieures ou égales à $2^{(P-1)}$.

La fréquence F_c est donnée par l'inverse du temps nécessaire pour effectuer le calcul, ici (listing 1) 45 microcycles soit une fréquence d'échantillonnage de 20,48 kHz. La résolution sera donnée suivant le nombre de bits de l'additionneur par la formule suivante : $r = F_c / 2^p$.

Il suffit donc de prendre l'additionneur de taille suffisante pour le couple résolution/fréquence maximum que l'on désire. Dans l'exemple précédent, si on prend un additionneur sur 16 bits, on aura une résolution : $F_c / 2^p = 0,3125$ Hz

Le LA 3 (440 Hz) correspondra donc à la valeur 1408 présentée à l'entrée de l'additionneur. Le listing 1 montre la réalisation de cet algorithme avec un processeur de type 8051. La forme d'onde générée est ici un sinus. Nous n'avons pas mis le tableau complet des valeurs du sinus (256 valeurs), aussi il faudra le compléter avant d'exécuter le sous-programme. Bien entendu, on pourra utiliser d'autres formes d'ondes ou bien même modifier celle-ci de manière dynamique... Un exemple complet de générateur de mélodie est disponible sur le serveur ERP...

J.-L VERN

Listing 1 ►

Erratum

Plusieurs petites erreurs se sont glissées dans l'article sur l'étude des filtres en double T du N°552 (novembre). Nous vous livrons ci-après les rectifications avec toutes nos excuses.

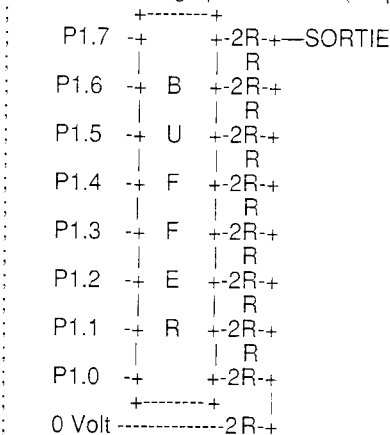
La figure 14 est donnée deux fois, la bonne est celle du bas, mais le commentaire est bon.

La figure 15 et son commentaire manquent.

Il y a deux figures 17, en fait la deuxième est la figure 20 avec son commentaire comme tout le monde l'aura compris.

Enfin la figure 30 de la page 67 reproduit la figure 28 mais le commentaire est bon (cette figure manque donc).

: La valeur binaire du signal de sortie est envoyée sur le port P1.
: Un convertisseur D/A en sortie permettra de transformer le signal digital en signal analogique. On pourra utiliser soit un convertisseur du commerce du type AD7533 ou autre ou bien plus simplement un ensemble de réseau de résistances. Dans ce cas, il faudra mettre entre le réseau de résistances et le port, des buffers CMOS de type 74HC541 ou 74HC241.
: On prendra par exemple pour la valeur 2R 22kohms, la valeur R étant obtenue en mettant deux résistances de 22k en parallèle. 6 réseaux identiques seront nécessaires, 2 pour les 8 résistances 2R, et 4 pour les 7 résistances R et la 9ième 2R. L'impédance de sortie étant élevée, il est souhaitable de faire suivre le montage par un buffer (Ampli op en suiveur ou collecteur commun).



```

RSEG SEGMENT_DATA
FREQ: DS 2
CUMUL: DS 2
DUREE: DS 2
DELAI: DS 2
MSB EQU 0
LSB EQU 1

```

: tableau des 256 valeurs de sinus obtenues de manière suivante :

: $V_i = (127 \times \sin(2 \times \pi \times i / 256)) + 128$

: les 256 valeurs du sinus sont calculées avec la valeur 128 pour 0. 255 pour 1 et 1 pour -1.

SINUS: DB 128+0.128+3.128+6.128+9.128+12.128+16.128+19.128+22....

: Génère pendant le temps DUREE la note de fréquence FREQ

JOUE: CLR A

: si la phase est remise à 0 entre chaque notes, il y a un claquement à chaque changement de notes...

: MOV CUMUL+MSB,A : commence avec la phase à 0

: MOV CUMUL+LSB,A

: MOV DUREE+LSB,A : et une durée de 0.4 sec par note

: MOV DUREE+MSB,#20H : 8192 * 48.83 us = 0.4

: MOV DPTR,#SINUS : table des sinus

JOUEF0: NOP : égalisation du temps de boucle

JOUEFR: MOV A,FREQ+LSB

: ADD A,CUMUL+LSB : fréquence 1

: MOV CUMUL+LSB,A

: MOV A,FREQ+MSB : calcule sur 16 bits

: ADDC A,CUMUL+MSB

: MOV CUMUL+MSB,A

: MOVC A,@A+DPTR : prend la valeur de la sinusoïde

: MOV P1,A

: boucle d'attente de 32 cycles (peut être remplacée par la gestion d'un autre

: oscillateur ou par la gestion de l'enveloppe du signal ...). Cette boucle

: d'attente amène la durée de la boucle de calcul de la forme d'onde à

: 45 cycles, soit, en utilisant un quartz de 11.0592 MHz, une fréquence

: d'échantillonnage de $11.0592 \text{ MHz} / (12 * 45) = 20.48 \text{ kHz}$ (un cycle demande

: 12 périodes d'horloge). On peut augmenter ou abaisser la fréquence

: d'échantillonnage en diminuant ou augmentant la durée de la temporisation.

: La fréquence de 20.48kHz permet d'avoir le LA 440 Hz juste avec une valeur

: de FREQ = 1408. ($1408 \times 0.3125 \text{ Hz} = 440 \text{ Hz}$)

: MOV R0,#15 : 1

: NOP : 1

DELAY: DJNZ R0,DELAY : 2 * 15

: décrémente le compteur (sur 16 bits) de durée de la note.

: la boucle met DUREE / 20.48 kHz = DUREE * 48.83us pour s'exécuter.

: DJNZ DUREE+LSB,JOUEF0

: DJNZ DUREE+MSB,JOUEFR

: 45 cycles @ 11.0592 MHz => $F_c = 20.48 \text{ kHz}$. Résolution : $F_c / (2^{16})$