

Retour sur la mise en œuvre du PSG AY3-8912

par André C.

Dans certaines de mes expérimentations musicales récentes (par exemple MereMichel2, qui utilise la commande SOUND), j'ai été amené à ajouter un CALL#C4B7 en fin de programme. Il s'agit d'une sorte de pseudo END qui permet de sortir du programme en évitant le "Ready" et surtout le parasite sonore qui l'accompagne (une sorte de "PEUP"). Impossible de trouver la cause de ce parasite.

En cherchant autre chose, je suis tombé sur la citation suivante :

"Le fait crucial suivant ne figure dans aucun manuel : Au boot et lors de chaque retour à l'interpréteur BASIC, le canal 1 est validé (mais ni le 2, ni le 3, ni le bruit). C'est comme si un PLAY 1,0,0,0 était exécuté. Ceci explique pourquoi les essais que l'on peut faire sur le canal 1 ne marchent pas toujours lors de leur mise en application ultérieure. Par exemple au "Ready", un MUSIC 2,3,10,9 ne produit rien, alors que MUSIC 1,3,10,9 produit bien un LA de l'octave 3. Si le bit 3 de #26A est à zéro (bruit clavier actif) alors PLAY 1,0,0,0 est envoyé à chaque "Ready". Si le bit 3 est à un (clavier muet) alors c'est le dernier PLAY qui reste valide (comme en mode programme) tant qu'on n'envoie pas un nouveau PLAY ou qu'on ne coupe pas l'alimentation électrique du PSG."

Ceci se trouve dans l'article "PLAY, MUSIC, SOUND, une mise au point effectuée par André C., Dominique P. et Claude S." (CEO-mag n°119, pages 25-39). En fait, Dominique est le père de cette observation étonnante, ce qui explique peut-être que je ne l'avais pas gardée en mémoire.

Comme ce phénomène me chiffonnait toujours, j'ai effectué quelques recherches. Voici les observations que j'ai pu faire :

1) Les sons parasites produits dépendent de la configuration. On peut bien sûr les entendre avec un Atmos seul (configuration 1), mais ils sont

plus importants dans la configuration Atmos+Microdisc+Sedoric (configuration 2). Ainsi, dans la configuration 2, si l'on remplace le CALL#C4B7 final par un END, le programme modifié produit un "PEUP" en rendant la main. Par contre avec la configuration 1, le même programme modifié de la même manière rend la main sans produire de parasite sonore. Curieux non?

2) Les sons parasites produits dépendent des commandes utilisées. Par exemple, les deux programmes MereMichel1 et MereMichel2 sont quasi identiques et produisent les mêmes sons, sauf que le premier utilise la commande MUSIC et reste silencieux après le retour au "Ready" (configuration 2), alors que le second utilise la commande SOUND et nécessite l'ajout du fameux CALL#C4B7 pour bloquer le disgracieux "PEUP"(toujours avec la configuration 2).

3) La manière dont on termine le programme joue aussi. Curieusement, ajouter une commande END en fin de programme est souvent toxique. Si nécessaire, mieux vaut sortir avec un CALL#C4B7 ou alors effectuer un reset des registres du PSG (voir plus loin la routine RSETPSG).

4) Si on utilise un Oric réel ou un émulateur.

Je ne suis pas allé plus loin dans la recherche des causes du phénomène, découragé par l'hétérogénéité des conditions dans lesquelles il se produit.

Mais **le problème vient sans doute en grande partie du fait que les 14 registres du PSG sont latchés (verrouillés)**, ce qui veut dire que la valeur qu'on y écrit y reste jusqu'à ce qu'on en écrive une autre ou que l'on coupe le courant.

Selon la configuration, les commandes musicales utilisées et la manière dont on termine le programme, il est difficile de savoir ce qu'il y a dans ces 14 registres. Certes un PLAY 0,0,0,0 semble tout couper, mais en apparence seulement car seul le registre 7 est mis à jour, sans nettoyage des 13

autres registres. Ce registre 7 sert à valider les canaux actifs. Les bits 0, 1 et 2 contrôlent respectivement les canaux 1, 2 et 3 ; les bits 3, 4 et 5 mélangent le bruit blanc aux canaux 1, 2 et 3. Attention, pour valider un canal, il faut mettre le bit correspondant à 0. Par conséquent la commande PLAY 0,0,0,0 envoie la valeur #3F (111111 en

binaires) dans le registre 7 ce qui bloque toutes les sorties ! Je ne suis pas à l'origine de cette conception tordue et j'espère que vous avez suivi ! Quand on revalide un canal (ou quand on retourne au "Ready", ce qui revalide le canal 1), on peut alors entendre s'exprimer ce que le PSG a dans le ventre, même si ce n'est ni voulu, ni agréable !

Je ne vois qu'une seule solution à ce problème : neutraliser tous les registres du PSG !

Registre 0 et 1 :	Tonalité canal 1 (de #00 à #0FFF)	mettre à #0000
Registre 2 et 3 :	Tonalité canal 2 (de #00 à #0FFF)	mettre à #0000
Registre 4 et 5 :	Tonalité canal 3 (de #00 à #0FFF)	mettre à #0000
Registre 6 :	Tonalité bruit blanc (de 0 à \$1F)	mettre à #00
Registre 7 :	Contrôle des canaux en service (de #00 à #3F)	mettre à #3F
Registre 8 :	Volume canal 1 (de #00 à #10)	mettre à #00
Registre 9 :	Volume canal 2 (de #00 à #10)	mettre à #00
Registre 10 :	Volume canal 3 (de #00 à #10)	mettre à #00
Registre 11 et 12 :	Durée du son (de #00 à \$FFFF)	mettre à #0000
Registre 13 :	Forme de l'enveloppe (de #00 à #0F en fait à #07)	mettre à #00

Ceci peut se faire à l'aide de la petite routine en assembleur RSETPSG ci-dessous :

```

; Routine RSETPSG pour neutraliser les 14 registres du PSG AY3-8912
; Cette routine sera appelée par CALL STP (soit #980E si la routine est implantée en #9800)
org $9800 ; Exemple d'adresse d'implantation de la routine
DATA db $00, $00, $00, $00, $00, $00 ; Tonalité canaux A, B et C (de 0 à $0FFF)
db $00 ; Tonalité bruit blanc (de 0 à $1F)
db $3F ; Canaux C+br, B+br, A+br, C, B, A)
db $00, $00, $00 ; Volume canaux A, B et (0 à $0F, -> enveloppe si $10)
db $00, $00 ; Durée du son (de 0 à $FFFF)
db $00 ; Forme de l'enveloppe (de 0 à 7)
STP ldx #LOW DATA ; Adresse des 14 valeurs
ldy #HIGH DATA ; à envoyer au PSG
jmp $FA86 ; Routine en Rom "écrire dans les 14 registres du PSG"
EOF

```

Il est évidemment possible de POKER les octets correspondants avec un programme Basic :

```

100 FOR K=#9800 TO #9814:READ V:POKE K,V:NEXT
200 DATA 0,0,0,0,0,0,#3F,0,0,0,0,0,0,#A2,#00,#A0,#98,#4C,#86,#FA
300 SAVE"RSETPSG",A#9800,E#9814

```

La routine est relogeable si on assemble le programme source à une autre adresse ou si on met à jour manuellement les octets du LDX #00 (ici #A2,#00) et ceux du LDY#98 (ici #A0,#98) afin de cibler la nouvelle adresse d'implantation.

La différence entre ces 2 CALL est que le CALL#C4B7 (routine de la Rom) rend la main (sans afficher le "Ready"), alors que le CALL#980E (routine RSETPSG en Ram) permet de rester dans le programme et éventuellement d'en sortir plus proprement. Ces deux remèdes devraient résoudre vos éventuels problèmes. N'hésitez pas à faire connaître vos tribulations musicales...