

SEDORIC? DO IT YOURSELF! (20)

Correction de la Gestion de PB5 : le PATCH.002

Notre Atmos est décidément une petite machine merveilleuse, simple et ouverte : Il est possible d'en comprendre chaque détail. A force de scruter cette petite chose, nombre d'entre nous ont pu constater qu'une des pattes du VIA 6522 est non connectée d'origine. Il s'agit de la ligne PB5 (Port B, bit n°5). C'est bête de laisser non utilisée quelque chose d'aussi précieux qu'une ligne d'entrée/sortie ! Encore faut-il que Sédoric sache respecter cette ligne PB5 dont il n'avait que faire jusqu'ici. Nous allons lui apprendre.

NATURE DU PROBLEME

PB5 n'ayant jamais été connecté, personne ne s'en est soucié. Résultat, beaucoup de programmes, massacrent PB5. Je veux dire que lorsqu'un programme écrit sur le Port B, il modifie de manière erratique l'état de PB5. Pour ceux qui voudraient enfin utiliser PB5, il devient donc nécessaire de disposer d'une recette universelle pour corriger les programmes existants, y compris Sédoric.

Rappelons que le Port A est utilisé pour l'imprimante, le son et le clavier. Le Port B est impliqué dans d'autres tâches : PB0 à PB3 pour le clavier, PB4 pour le STROBE de l'imprimante, PB6 pour le "Remote control" du lecteur de K7 et enfin PB7 pour l'entrée/sortie des data K7. PB5 est resté

inutilisé. Deux registres sont utilisés pour chaque port : un registre de direction des échanges (entrée ou sortie) et un registre de data (là où il faut lire ou écrire sur le port). Pour le Port B, ces registres sont respectivement accessibles aux adresses #0302 et #300. En fait, à chacune des 8 lignes d'un port correspond un bit dans ces registres. Par exemple pour mettre PB5 en sortie il faut poker #20 (0010 0000) en #0302. Et pour tirer PB5 au +5V (haut logique), il faut poker #20 en #0300. En pratique, ce n'est pas si simple, car il ne faut toucher qu'au bit n°5 (la numérotation commence au bit n°0). Dans l'exemple ci-dessus, nous avons non seulement mis PB5 en sortie, mais aussi forcé les autres lignes en entrée ! Le protocole à utiliser pour programmer correctement est indiqué en annexe.

PRINCIPE DE LA CORRECTION

La commande fautive étant toujours un STA 0302 (qui occupe 3 octets), il suffira : 1) De la remplacer dans le code à corriger par un JSR XXXX (qui occupe lui aussi 3 octets). 2) D'installer à l'adresse XXXX une petite routine qui lira le contenu du registre #0300, modifiera le bit n°5 sans changer la valeur des autres bits et re-écrira le résultat en #0300.

CORRECTION DE SEDORIC

C'est bien sûr par là qu'il faut commencer. Toutes les versions de Sédoric sont affectées, mais nous ne corrigerons que la version 3.0. Je vous propose de fabriquer une petite rustine, le PATCH.002, qui viendra se coller sur la partie fautive de Sédoric, en RAM overlay. Comme précédemment avec le PATCH.001 qui corrigeait le "INSERT MASTER DISC IN DRIVE" (voir "DO IT YOURSELF" n°17), il faudra insérer PATCH.002 dans la commande

I N I S T ,

afin que la correction prenne effet dès le boot. Attention, notez que seule la RAM overlay sera modifiée et non votre disquette master Sédoric.

Nous allons assembler (ou poker directement les octets indiqués ci-dessous, pour ceux qui n'ont pas d'assembleur) nos deux modifications en RAM, les sauver, changer les adresses des 2 fichiers sauvés avec la commande STATUS (pour qu'ils soient ensuite chargés directement à la bonne place dans la RAM overlay) et enfin les merger dans le fichier PATCH.002

1) Assemblez (ou pokez) la routine corrective à l'adresse 981E :

981E	48	PHA	sauve la valeur "V" qui était destinée au Port B
981F	A9 20	LDA #20	soit masque 0010 0000 pour lire l'état actuel "X" de PB5 en 0300
9821	2D 00 03	AND 0300	résultat : l'accumulateur contient maintenant 00X0 0000
9824	8D 2B EA	STA EA2B	qui met à jour le #00 dans la routine elle-même (un peu plus loin)
9827	68	PLA	recupère la valeur "V" d'origine à écrire dans le Port B
9828	29 DF	AND #DF	soit le masque 1101 1111 qui force à 0 le bit 5 de "V" puis le
982A	09 00	ORA #00	remplace par le bit 5 d'origine en gardant les autres bits de "V"
982C	8D 00 03	STA 0300	et enfin écrit le résultat dans le Port B
982F	60	RTS	avant de retourner au point d'appel

SAVE"P1",A#981E,E#982F, puis STATUS"P1",A#EA1E, et enfin CHKSUM"P1", qui doit vous indiquer les adresses en RAM overlay (EA1E à EA2F) et la CHKSUM #054C. Si ce n'est pas le cas... corrigez!

2) Assemblez (ou pokez) en 983A l'appel à cette routine qui sera patchée en RAM overlay à l'endroit où se trouve le STA 0300 fautif dans Sédoric, c'est à dire en D83A.

983A 20 1E EA JSR EA1E qui occupe 3 octets comme le STA 0300 qu'il remplace
SAVE"P2",A#983A,E#983C.┘ puis STATUS"P2",A#D83A.┘ et enfin CHKSUM"P2"┘ qui doit vous indiquer les adresses en RAM overlay (D83A à D83C) et la CHKSUM #0128. Si ce n'est pas le cas... corrigez!

Si tout va bien, terminez avec COPYM "P?" TO "PATCH.002"┘ Vérifiez éventuellement avec un CHKSUM"PATCH.002"┘ et ajoutez PATCH.002:?"Sédoric est patché!" à votre INIST. Voilà, désormais Sédoric est prêt pour les nouvelles applications utilisant PB5.

Les lecteurs attentifs de cette rubrique se rendront peut-être compte que j'ai logé la routine corrective dans une zone de Sédoric 3.0 qui contient

des NOP et qui était donc en réserve pour ce genre d'opération. Toutefois dans le "DO IT YOURSELF" n°19, j'ai utilisé cette même zone pour créer des banques supplémentaires. Si vous voulez avoir le beurre et l'argent du beurre, il vous faudra déplacer une commande Sédoric du NOYAU vers une banque afin de libérer de la place. Si vous avez oublié comment faire, révisez vos anciens "DO IT YOURSELF".

CORRECTION DES PROGRAMMES BASIC ET LM

Comme bien sûr, Sédoric n'est pas le seul responsable et que beaucoup de programmes perturbent aussi PB5, en cas de besoin, il vous faudra rechercher le ou les STA 0300 (ou POKE#0300) et les remplacer par des JSR XXXX (ou CALL#XXXX). En RAM, à l'adresse XXXX de votre choix, devra se trouver une routine corrective analogue à celle décrite plus haut. Vous ne pouvez pas utiliser celle que vous avez patchée en RAM overlay car les JSR ou les CALL de votre correctif aboutiraient en ROM.

Prenons un exemple concret. Si vous implantez la routine corrective en 981E, il faudra changer le STA EA2B en STA 982B. C'est simple suivez le listing ci-dessus et modifiez seulement

9824 8D 2B 98 STA 982B qui auto-modifie la routine elle-même en RAM

Si vous optez pour un autre emplacement, il faudra ajuster le STA 982B de façon à écrire la valeur de l'accumulateur à l'endroit correspondant de votre routine.

ANNEXE

Protocoles à utiliser pour programmer correctement en BASIC (ou en LM) :

Pour mettre PB5 en entrée sans modifier la direction des autres lignes :

100 A=PEEK(#302)	(LDA 0302)	pour lire l'état actuel du registre de direction du Port B
110 A=A AND #DF	(AND #DF)	masque 1101 1111 pour forcer PB5 à zéro
120 POKE#302,A	(STA 0302)	les autres bits resteront tels quels.

Pour mettre PB5 en sortie sans modifier la direction des autres lignes :

100 A=PEEK(#302)	(LDA 0302)	pour lire l'état actuel du registre de direction du Port B
110 A=A OR #20	(ORA #20)	masque 0010 0000 pour forcer PB5 seulement à un
120 POKE#302,A	(STA 0302)	les autres bits resteront tels quels.

Lorsque PB5 est en entrée, pour lire sa valeur dans le registre data du Port B :

100 A=PEEK(#300)	(LDA 0300)	pour lire l'état actuel du registre de data du Port B
110 A=A AND #20	(AND #20)	le masque 0010 0000 force tous les bits à 0 sauf PB5
120 IF A=0 THEN...	(BEQ...)	A=0 lorsque PB5 est au niveau bas

Lorsque PB5 est en sortie, pour le mettre au niveau bas (à la masse) :

100 A=PEEK(#300)	(LDA 0300)	pour lire l'état actuel du registre de data du Port B
110 A=A AND #DF	(AND #DF)	masque 1101 1111 pour forcer PB5 à zéro
120 POKE#300,A	(STA 0300)	les autres bits resteront tels quels.

Lorsque PB5 est en sortie, pour le mettre au niveau haut (le tirer à +5V) :

100 A=PEEK(#300)	(LDA 0300)	pour lire l'état actuel du registre de data du Port B
110 A=A OR #20	(ORA #20)	masque 0010 0000 pour forcer PB5 à un
120 POKE#300,A	(STA 0300)	les autres bits resteront tels quels.

CONCLUSION...

Vous n'avez plus d'excuse maintenant pour ne pas développer une application originale basée sur l'exploitation de la ligne d'entrée/sortie PB5. Ce peut être la commande d'un relais pour votre train électrique (ce qui fait deux avec PB6). Ou pour commander l'allumage d'une LED. Où la détection d'un événement externe. Ce peut être aussi tout simplement l'utilisation de cartouches PB5 (voir le "Journal du Soft" n°9) qui devrait vous permettre de profiter de 16384 octets de ROM supplémentaires afin d'y installer les routines LM que demande le jeu que vous en train de développer ! N'hésitez pas à me contacter si vous avez besoin d'aide.

André Chéramy 54 rue de Sours 28000 CHARTRES