
RÉALISEZ VOS CARTOUCHES PB5 (8)

par André Chéramy, Claude Sittler et Dominique Pessan

Aujourd'hui : Retour sur la gestion de PB5

Nous avons précédemment montré comment utiliser le signal PB5 dans quelques applications basées sur une double ROM, la fameuse "cartouche PB5". Cependant plusieurs lecteurs ont exprimé le souhait d'avoir plus d'explications sur ce qu'est PB5, sur la manière de gérer ce signal et sur la possibilité de l'utiliser pour d'autres applications.

QU'EST-CE QUE PB5 ?

Pour communiquer avec l'extérieur (clavier, K7, son), l'Oric-1/Atmos utilise une puce, le VIA 6522. Ce VIA possède deux ports A et B, chacun d'eux étant un registre de 8 bits (numérotés de b0 à b7) qui correspondent à 8 entrées/sorties indépendantes. Pour s'y retrouver, les entrées/sorties du Port A sont numérotées de PA0 à PA7 et celles du Port B de PB0 à PB7. Chaque port comporte un registre de direction de 8 bits qui définit le sens de chacune des 8 entrées/sorties et un registre de data où sera lu (entrée) ou écrit (sortie) le niveau électrique de chacune des 8 entrées/sorties. Pour le port B qui nous intéresse aujourd'hui les registres de direction et de data ont respectivement pour adresse 0302 et 0300. Par simplification, lorsqu'on parle de PB5, il faut comprendre le bit 5 de l'octet d'adresse 0300 (registre data du port B).

Ces 16 entrées/sorties sont connectées à 16 broches du VIA 6522 et peuvent être soit à la

masse (niveau bas) soit au +5V (niveau haut). Nous n'entrerons pas dans le détail de l'utilisation de ces entrées/sorties. Mais sachez **par exemple** que PB6 (broche 16 du 6522) est connecté au relais du moteur de, quelle tentation d'utiliser ce signal pour piloter de nouveaux périphériques ou magnétophone. Lorsque l'on fait un CSAVE, PB6 est mis à 1 et la broche 16 du 6522 tirée au +5V, ce qui colle le relais et qui met en marche le magnétophone (si la prise de commande du moteur est connectée).

Dans l'Oric, toutes les entrées/sorties du VIA sont utilisées, **sauf PB5** : la broche 15 n'est pas connectée, elle est restée "en l'air". Les concepteurs de l'Oric n'ont pas trouvé d'utilité à cette entrée/sortie ! Pour les Oriciens d'aujourd'hui, quelle tentation d'utiliser ce signal pour piloter de nouveaux périphériques ou ajouter de la mémoire à leur machine !

POUR PILOTER PB5

Le principe est ultra simple : il suffit de mettre le bit 5 de 0300 (registre data) à zéro ou à un, sans toucher les autres bits, puis de mettre PB5 en sortie sans toucher la direction des autres entrées/sorties. Cette deuxième étape se fait en mettant sélectivement à un le bit 5 de 0302 (registre de direction). **Notez qu'il faut respecter cet ordre** sous peine de voir PB5 prendre brièvement une valeur erratique s'il était précédemment en entrée.

Les personnes allergiques à la logique booléenne peuvent sauter l'encadré qui suit. Les autres vont se régaler. A toutes fins utiles, les versions en langage machine et en BASIC sont indiquées :

1) Mettre sélectivement PB5 au niveau bas sans altérer les autres entrées/sorties

LDA 0300	ou	V = PEEK(#300)	on lit la valeur actuelle du port B
AND #DF		V = V AND #DF	le masque 1101 1111 force le bit 5 à zéro
STA 0300		POKE#300,V	la nouvelle valeur est remise en place
LDA 0302		D = PEEK(#302)	on lit le registre de direction des entrées/sorties
ORA #20		D = D OR #20	le masque 0010 0000 force le bit 5 à un (sortie)
STA 0302		POKE #302,D	la nouvelle direction est remise en place
RTS		RETURN	fin de la procédure

2) Mettre sélectivement PB5 au niveau haut sans altérer les autres entrées/sorties

LDA 0300	ou	V = PEEK(#300)	on lit la valeur actuelle du port B
ORA #20		V = V OR #20	le masque 0010 0000 force le bit 5 à un
STA 0300		POKE#300,V	la nouvelle valeur est remise en place
LDA 0302		D = PEEK(#302)	on lit le registre de direction des entrées/sorties
ORA #20		D = D OR #20	le masque 0010 0000 force le bit 5 à un (sortie)
STA 0302		POKE #302,D	la nouvelle direction est remise en place
RTS		RETURN	fin de la procédure

3) Inverser sélectivement le niveau de sortie de PB5 sans altérer les autres ent./sorties

LDA 0300	ou	V = PEEK(#300)	on lit la valeur actuelle du port B
EOR #20		V = (NOT(VAND#20))AND(VOR#20)	inverse seulement le b5
STA 0300		POKE#300,V	la nouvelle valeur est remise en place
LDA 0302		D = PEEK(#302)	on lit le registre de direction des entrées/sorties
ORA #20		D = D OR #20	le masque 0010 0000 force le bit 5 à un (sortie)
STA 0302		POKE #302,D	la nouvelle direction est remise en place
RTS		RETURN	fin de la procédure

PROBLÈMES “SOFT” ASSEZ COTON

Si la situation est claire du côté “hard” (il suffit de repérer le 6522 et de relier sa broche 15 au périphérique à piloter), il n’en va pas de même du côté “soft”. En effet, à l’origine, PB5 étant “en l’air”, les concepteurs ne s’en sont pas souciés, en conséquence PB5 est souvent “bousculé”.

Pour utiliser correctement PB5, il ne suffit donc pas de fixer son niveau et de le mettre en sortie. Il faut aussi s’assurer que la gestion de PB5 est correctement traitée par ailleurs.

Il y a trois niveaux “soft” dans l’Oric : la ROM (V1.0 ou 1.1), le DOS (par exemple SEDORIC) et le programme d’application. Chacun de ces trois niveaux est susceptible de

perturber PB5 de manière erratique. Nous supposons que votre programme d’application, nouvellement conçu pour utiliser PB5, sera correctement codé et respectera PB5. Nous avons déjà corrigé la version 3.0 de SEDORIC à l’aide du **PATCH.002** que vous trouverez dans toutes les disquettes trimestrielles récentes. Reste donc la ROM qui pose problème. Nous allons voir ça.

Si l’on branche une diode électroluminescente (LED) sur PB5, on peut la voir papillonner à qui mieux mieux ! A défaut de LED, voici un petit programme BASIC à faire tourner de préférence sur un Oric nu (sans lecteur de disquette) :

100 FOR J = 0 TO 1 :D=J*32	pour tester les 2 valeurs de direction possibles pour pb5
110 E=0:POKE#302,D	compteur d'erreurs, mise en place valeur de direction
120 PRINT"DIRECTION"J,	impression de la valeur de direction initiale
130 FOR I = 0 TO 255	pour tester toutes les valeurs de data du port B
140 POKE#300,I:V=PEEK(#300)	mise en place de la valeur data et relecture immédiate
150 A=I AND #20:B=V AND #20	ne garde que le bit 5, les autres bits sont mis à zéro
160 IF A<>B THEN E=E+1	erreur si PB5 a varié entre mise en place et relecture
170 NEXT I	valeur suivante de data
180 PRINT E,PEEK(#302)	impression nombre d'erreurs et valeur finale direction
190 NEXT J	valeur suivante de direction

Ce programme permet d'observer que la valeur pokée en #302 (registre de direction) reste stable, c'est déjà rassurant.

Quand PB5 est en entrée (en lecture), on peut toujours essayer d'écrire (sortie) quelque chose dedans, ça ne marche pas, elle garde toujours la même valeur, 0 ou 1 qui dépend de ce qui arrive sur la broche 15 du 6522, ici probablement 0 tout le temps, puisque cette broche 15 est « en l'air »... d'ou les 128 erreurs (les 128 "1", c'est à dire une fois sur deux) que l'on peut observer. Ceci démontre la stabilité de la broche 15 du VIA qui, bien qu'elle soit « en l'air » conserve une valeur stable.

Lorsque la direction assignée est la sortie, la valeur donnée à PB5 (POKE en #0300 dans le registre de data) est parfois perdue. En effet PB5 diffère entre le POKE et le PEEK, **qui sont pourtant immédiatement consécutifs**.

Le port B est impliqué dans les entrées/sorties du clavier, de l'imprimante et du magnétophone. Voyons par exemple ce qui se passe si nous inhibons le clavier. Pour se faire, ajoutez deux lignes au programme ci-dessus : KEY OFF au début et KEY SET à la fin et lancez le sur un Atmos avec Microdisc et SEDORIC patché (voir plus haut). Vous pourrez constater qu'il n'y a plus aucune erreur.

Conclusion, **il faut placer PB5 en sortie et inhiber le clavier**. Pour ce faire sans passer par SEDORIC, voici la recette utilisée par Fabrice Broche : LDA#00 STA030B RTS Au contraire, pour rétablir le clavier : LDA0307 STA0305 LDA#40 STA030B RTS (voir SEDORIC 3.0 À NU, page 296). Ca marche aussi en BASIC, mais attention à ce que vous faites !

RECHERCHES DANS LA ROM V1.1

Clavier, imprimante, magnétophone... mieux vaut corriger la ROM plutôt que d'accumuler les contraintes. Nous avons vu que pour altérer PB5, il faut modifier les octets en 0300 et 0302. Il suffit donc, à l'aide d'un éditeur hexadécimal de rechercher les deux suites d'octets 00 03 et 02 03 et d'examiner si nos trouvailles sont pertinentes. Voici ce que ça donne avec la V1.1:

1) Configuration pour travail K7 :

En **E776**, on trouve un STA0300,X. L'analyse de la routine révèle que la valeur #FF est placée en 0302 et donc que l'ensemble du port B est mis en **sortie**. C'est parfait à ceci près que le niveau de ces 8 sorties est **indéfini** pendant un court laps de temps (qui peut cependant être lourd de conséquences) avant d'être forcé vers le bas juste après (voir ci-dessous).

En **E77E**, la valeur #40 (0100 0000) est placée dans le registre data (0300) dont le bit 6 passe au niveau haut (active le moteur du magnétophone). Ce faisant on force toutes les autres sorties dont PB5 au niveau bas ! Cette bogue rend incompatible l'utilisation de PB5 et d'une K7.

Correction (réservée aux amateurs de code machine) :

Si l'usage de ces commandes était nécessaire, il faudrait inverser l'ordre des opérations : placer la valeur, puis fixer la direction.

En **E770**, le code serait alors : LDA#70 STA 0300 qui corrige aussi la bogue du strobe inopiné.

En **E775**, on retrouvera l'ancien code de LDX E782,Y à BPL E775 (avec adresse adaptée).

2) Gestion du clavier :

En **F570**, une valeur de la forme 1011 1xxx est placée en 0300 (xxx étant le numéro de ligne du clavier à scruter). Dans cette valeur le bit 5 est toujours à un donc PB5 est toujours forcé au niveau haut. C'est cette bogue qui rendait nécessaire l'inactivation du clavier.

Correction (réservée aux amateurs de code machine) :

La seule solution est de remettre en place dans le bit 5 la valeur qu'il avait initialement. La routine incriminée commence en F561 par un PHA intouchable. Mais on peut intervenir juste après :

En **F562**, remplaçons le LDA#0E JSR F590 par **NOP NOP JSR EDB4**, adresse de notre greffon. Il nous faut en effet trouver de la place ailleurs en ROM, par exemple à l'adresse EDB4, dans le message de copyright, (qu'il faudra donc couper un peu).

En **EDB4**, on aura alors le code suivant :

LDA 0300 AND#20 (isolation du bit 5, on a alors une valeur de la forme 00x0 0000)

ORA#98 STA 00 (qui donne la forme 10x1 1000 et mise en réserve à l'adresse 00)

LDA#0E (remplacé précédemment par des NOP dans la routine mère)

JMP F590 (remplacé précédemment par JSR EDB4 dans la routine mère)

NOP (pour nettoyer la fin du copyright).

La routine reprend alors son cours normal en F567.

En **F56E**, il faut encore remplacer le ORA#B8 (1011 1000) par **ORA 00** pour que la valeur initiale du bit 5 soit prise en compte, ainsi que les bits initialement requis par la routine (b3, b4 et b7).

A titre d'exemple, voici le nouveau message de copyright que nous avons placé de ED96 à EDB3 : "ATMOS/PB5 V1.500D0A© DAC 2000D0A00" soit 30 octets au lieu des 45 initiaux. Notre greffon occupe les 15 octets restants. Nous avons aussi changé les couleurs d'encre en F915 (#04 bleu au lieu de #00 noir) et de fond en F91A (#13 jaune au lieu de #17 blanc).

3) Gestion de l'imprimante :

En **F5CB**, un strobe est généré en forçant sélectivement le bit 4 à zéro. Bravo, rien à dire.

En **F5D3**, remise sélective du bit 4 à un. Voilà du bon travail.

4) Initialisation du VIA :

En **F9B1**, la valeur #F7 (1111 0111) est placée dans le registre de direction (0302). Le port B est donc forcé en sortie sauf PB3 qui est mis en entrée (détection clavier). E776 et F9B1 sont apparemment les deux seuls endroits de la ROM V1.1 où le registre de direction est touché. Dans les deux cas PB5 est forcé en sortie, ce qui est parfait. Il y a donc tout lieu de croire qu'au boot et après chaque reset PB5 est forcé en sortie et y reste.

En **F9B6**, la valeur #B7 (1011 0111) est placée dans le registre data du port B (0300). On remarque que PB5 est forcé au niveau haut (il est d'autre part en sortie, voir ci-dessus). Ceci ne pose pas vraiment de problème, sauf qu'au boot et après un reset, PB5 sera par défaut au niveau haut. Dans une cartouche PB5 par exemple, c'est la moitié supérieure de la ROM qui sera active, car la ligne d'adresse A14 commandée par PB5 sera au niveau haut.

Il est toujours possible, au début de votre application, de remettre PB5 au niveau bas, si cela était nécessaire. Sinon, il faudrait remplacer en F9B6 dans la ROM la valeur #B7 (1011 0111) par #97 (1001 0111). Mais nous ne ferons pas cette correction, pour rester compatible avec les cartouches PB5 déjà élaborées.

CONCLUSION

La ROM que nous avons élaborée, respecte complètement PB5 et va donc nous servir de référence. Elle s'appelle ATMOSPB5.ROM, elle a une checksum de #1C09 et est à votre disposition. Vous avez probablement encore beaucoup de questions, notamment sur l'utilité de PB5. Patience, nous n'avons pas dit notre dernier mot. Une surprise est en préparation...