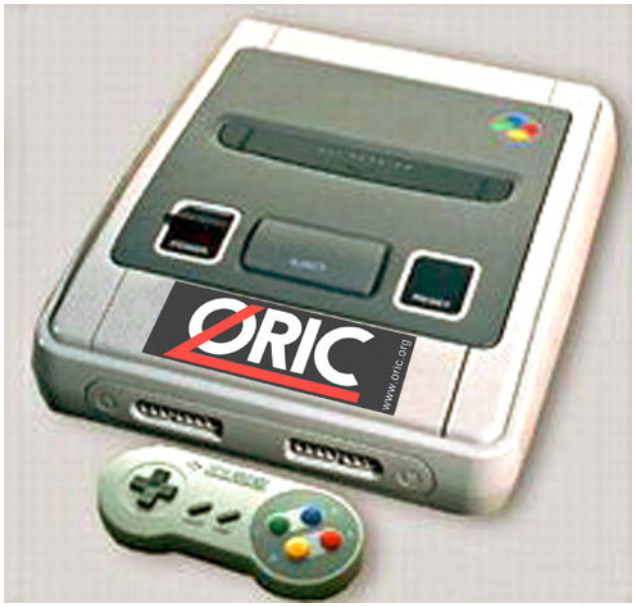
Mise au Point Software pour Cartouche Super-Oric (5)

L'écran Texte

par André C.



RESUME DE LA SITUATION

Dans les épisodes précédents, nous avons vu comment insérer un petit programme Basic dans une cartouche Super-Oric (pour afficher "Vive le Super-Oric"), comment détecter les appuis sur le Joypad (12 possibilités différentes), comment confectionner et afficher un écran Hires (composé des fichiers "Palette" et "image") et enfin quelques notions sur la structure de l'écran Hires (pour le plaisir, car les commandes Basic nous dispensent de connaître ces détails).

La structure d'une cartouche Super-Oric est très simple. Il suffit de concaténer (mettre bout à bout) une 'base' de Fabrice (#8000 octets, cette 'base' est une adaptation du Basic 1.1 à la console Snes) et le ou les fichiers au format Tap de type Euphoric (programme Basic, écrans Texte et/ou Hires, caractères redéfinis etc.).

QUELQUES PRECISIONS IMPORTANTES CONCERNANT LA SYNTAXE

Il y a une double complication avec la syntaxe des quatre commandes destinées respectivement à forcer l'affichage, charger/sauver un programme et tester le joypad :

1) Si on passe par Euphoric (en configuration Atmos et donc avec la Rom Basic 1.1 classique), certains mots clés ne sont pas encodés correctement.

2) Il existe 2 variantes de cartouches Super-Oric.

Commençons par cette histoire de variantes de cartouches. Il existe en effet deux 'bases' dans les cartouches Super-Oric. La plus ancienne (que j'ai baptisé 'Base1' et que l'on trouve dans les cartouches Superoric.swc et Supersoko.swc) utilise les mots clefs REFRESH, LOAD, SAVE et JOY. La plus récente (que j'ai baptisé 'Base2' et qui se trouve dans la flash 29F010, prêtée par Fabrice) utilise respectivement RELEASE, CLOAD, CSAVE et '&'. Ceci appelle deux petits commentaires :

a) La commande REFRESH s'appelle SHOW dans l'article de Fabrice (Ceo-Mag n°151 pages 14-17), mais correspond bien au même token (#A0) et à la même fonction.

b) Après avoir restauré le mot-clef et le code original de la commande '&', Fabrice a finalement utilisé la commande USR(0) pour la lecture du Joypad. En conséquence, la lecture du Joypad s'effectue avec le token de '&' (soit #DD) dans la première version et avec le token de USR (soit #D9) dans la deuxième version.

Selon que l'on part de la 'Base1' ou de la 'Base2', il faudrait utiliser les mots-clefs REFRESH, LOAD, SAVE et JOY ou RELEASE, CLOAD, CSAVE et USR(0), mais…

Ici intervient la deuxième complication dont je parlais : Il est crucial de se rappeler que nous introduisons un biais en utilisant la Rom 1.1 classique pour préparer les programmes Basic destinés au Super-Oric. Ce qui compte ce sont les tokens qui seront rencontrés dans le programme pendant l'exécution avec le Super-Oric. En effet, dans les programmes Basic (et donc les fichiers Tap) les commandes ont été remplacées par des tokens. Par exemple, à la place du mot 'RELEASE' (fichier texte) on a le token #A0 (fichier Tap).

Or, le Basic 1.1 ne connaît pas les mots clefs REFRESH, LOAD, SAVE et JOY et ne sait pas les encoder (les remplacer par le bon token). C'est pourquoi Fabrice est revenu à des noms de commandes normaux (RELEASE, CLOAD, CSAVE et USR) dans sa deuxième version. Le Basic 1.1 classique les remplace respectivement

par les tokens #A0, #B6, #B7 et #D9 qui vont bien avec l'interpréteur du Super-Oric (deuxième mouture). Alors, pour encoder correctement un programme destiné au Super-Oric première mouture, il faut tricher et y écrire les commandes RELEASE, CLOAD, CSAVE et '&' (à la place de REFRESH, LOAD, SAVE et JOY). Le Basic 1.1 classique va remplacer ces mots-clés par les tokens corrects #A0, #B6, #B7 et #DD.

On voit donc, qu'en pratique, il faut seulement faire attention à utiliser '&' pour la 'Base1' et USR pour la 'Base2'. Le reste marche pareil, tant qu'on respecte la syntaxe RELEASE, CLOAD et CSAVE si on passe par le Basic 1.1 classique pour fabriquer les programmes destinés au Basic 1.1 Super-Oric.

Note provisoirement pessimiste : Il existe pour l'instant une troisième complication. Une bogue empêche le fonctionnement de la commande RELEASE avec la 'Base2' et donc il faut s'en tenir à la 'Base1', si on veut afficher un écran Hires.

UTILISATION DES ECRANS 'TEXTE'

L'écran texte des Oric classiques comporte 28 lignes de 40 caractères, ce qui représente 28x40=1120 octets, situés en Ram de #BB80 à #BFDF, chaque caractère étant codé sur un octet. L'écran texte du Super-Oric comporte 28 lignes de 32 caractères, mais chaque caractère est codé sur 2 octets, ce qui permet de choisir parmi 1024 caractères, de nombreuses couleurs etc. Ceci représente 28x32x2=1792 octets (#700 octets). Mais ces octets ne sont pas situés dans la Ram habituelle. En effet, le Super-Oric est doté de deux banques de 64 Ko et c'est la deuxième banque qui abrite les écrans Texte et Hires, la palette de couleurs et les caractères redéfinis. Ceci libère beaucoup de place dans la première banque où se trouve la Ram habituelle, qui reçoit les programmes (48 vrais Ko) et la Rom 1.1 (16 Ko). L'écran texte se trouve donc dans la deuxième banque de #F000 à #F6FF. Ceci a deux conséquences :

1) Pour charger un écran texte, il ne faut plus indiquer CLOAD"ECRAN" (le fichier tap correspondant ayant les adresses #BB80 à #BFDF), mais CLOADTEXT,"ECRAN" (le fichier Tap correspondant ayant les adresses # F000 à #F6FF). La commande CLOAD 'tout court' enverrait votre écran dans la première banque de F000 à F6FF. Dans le meilleur des cas rien ne serait affiché et dans le pire votre programme en Ram serait écrasé !

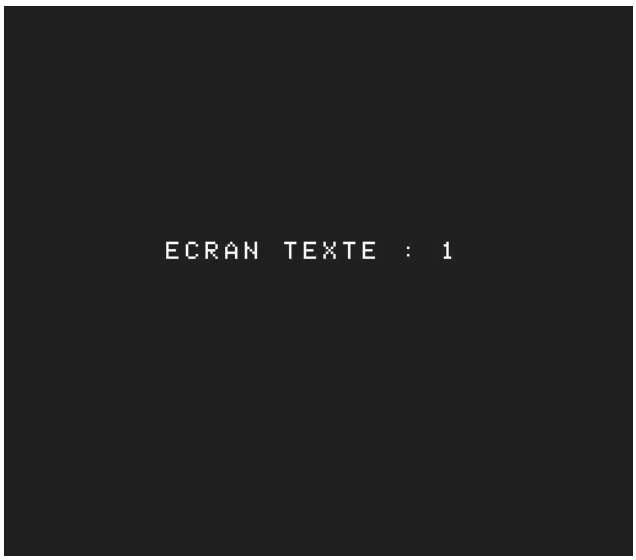
2) Finis les PEEK et POKE dans l'écran texte. Ces deux commandes agissent uniquement dans la première banque. Alors, il faut ruser. Et en fait, c'est beaucoup plus simple et pratique, car il faut indiquer des coordonnées X,Y au lieu d'une adresse. A la place de PEEK, il faut utiliser SCRN(X,Y). Et à la place de POKE, il faut utiliser PLOT X,Y, WORD où WORD est une valeur sur deux octets. SCRN renvoie aussi une valeur sur deux octets. La structure des 16 bits de cette valeur qui correspond en fait aux deux octets définissant chaque caractère de l'écran a été indiquée par Fabrice dans l'article déjà cité. Nous reviendrons dessus le moment voulu.

CONFECTION D'UN ECRAN TEXTE

Pour tester le chargement d'un écran texte avec la commande CLOADTEXT,"" il faut d'abord fabriquer un fichier Tap contenant un tel écran. Rassurez-vous cet écran sera réduit à sa plus simple expression : Le message 'ECRAN TEXTE : 1' sera visible au beau milieu et c'est tout !

Pour ce faire, la méthode habituelle consistant à passer par Euphoric avec la Rom Basic 1.1 classique est trop fastidieuse, parce que dans un deuxième temps il faut mouliner l'écran produit avec une routine qui le re-formatera au standard Super-Oric (32 caractères par ligne au lieu de 40). Si en outre, vous voulez convertir les attributs 'série' de l'écran texte classique en attributs 'parallèle' (individuel à chaque caractère), ce sera un peu trop compliqué pour aujourd'hui. C'est faisable mais nous verrons ça une autre fois.

Donc nous optons pour la méthode manuelle. A l'aide d'un éditeur hexadécimal créer un bloc de #700 fois l'octet #20. Pour les curieux, cela correspond à des espaces. Pour les très curieux, cela correspond à des espaces en priorité 1 (voir l'article de Fabrice déjà cité). Ensuite, à l'offset #310 tapez le message 'ECRAN TEXTE : 1'. Comme un caractère correspond à deux octets, il faut taper un octet sur deux en laissant inchangé le #20 entre chaque lettre. Le '1' doit donc finalement se trouver à l'offset #32C.



Si votre éditeur hexadécimal ne permet pas d'entrer des caractères Ascii (ce dont je doute), entrez les octets suivants en hexadécimal (on peut en voir la signification à la ligne en dessous) :

```
4520432052204120 4E20202054204520
E  C  R  A  N  .  T  E
5820542045202020 3A20202031 20
X  T  E  .  :  .  1
```

Explication réservée aux curieux : A chaque ligne de l'écran correspondent 32 caractères soit 64 octets (#40). Pour faire simple, la 12e ligne de l'écran se trouve à l'offset 12x64=768 (#300) et pour centrer le message (qui fait 15 caractères) sur la ligne de 32 caractères, il faut décaler de 8 caractères soit 16 octets (#10). On commence donc à écrire à l'offset #310.

Enfin, devant ce bloc de #700 octets, il faut ajouter une entête cassette, avec les bonnes adresses #F000 à F6FF et le titre 'ECRTXT1' soit :

```
16161624000080 00F6FFF00000454352
. . . . . fin déb . E C R
5458543100
T X T 1 .
```

(également en hexadécimal avec signification simplifiée à la ligne en dessous).

Sauvez-le tout sous le nom 'ECRTXT1.TAP'. Pendant que vous y êtes faites 3 autres écrans en changeant seulement le n°1 par 2, 3 et 4. Attention il y a deux changements à faire pour chaque fichier Tap : Dans le corps de l'écran qui sera affiché et dans l'entête (nom du fichier). Sauvez respectivement sous les noms 'ECRTXT2.TAP', 'ECRTXT3.TAP' et 'ECRTXT4.TAP'.

CONSTRUCTION DE LA CARTOUCHE

Sous Euphoric en configuration Atmos, tapez le court programme suivant :

```
100 CLOADTEXT,"ECRTXT1"
110 WAIT 100:END
```

Notez que le WAIT, c'est pour avoir le temps d'appuyer sur F12 (recopie d'écran), avant que le vilain "Ready" ne pollue notre bel écran Texte !

Sauvez ce petit programme avec CSAVE "ECRTXTA",AUTO. A l'aide de votre éditeur hexadécimal concaténez les trois fichiers Base2a.bin, Ecrtxta.tap et Ecrtxt1.tap et complétez la cartouche avec des #00 jusqu'à avoir une taille totale de #40000 octets. Sauvez sous le nom Ecrtxta.swc (qui sera sur la disquette trimestrielle de mars 2005). Rappel : Le fichier Base2a.bin indique une taille totale de 2Mbits (256 Ko) soit #40000 octets. Vous trouverez ce fichier dans la disquette trimestrielle de décembre 2004 (dans le fichier Vive_le_Super-Oric.zip). Si nécessaire, je peux aussi vous le fournir sur simple demande.

ESSAI D'AFFICHAGE DE L'ECRAN TEXTE

Lancer l'émulateur Snes9x (homepage <http://www.snes9x.com>) et charger la cartouche Ecrtxta.swc. Vous devez voir l'écran qui illustre la page précédente, bien modeste il est vrai, mais le but de l'opération était de montrer comment ça marche et pas de faire de l'art.

DESSIN ANIME

Si vous tentiez le programme suivant :

```
100 CLOADTEXT,"ECRTXT1"
110 CLOADTEXT,"ECRTXT2"
120 CLOADTEXT,"ECRTXT3"
130 CLOADTEXT,"ECRTXT4"
140 WAIT 100:END
```

Vous ne verrez rien, sauf le dernier écran bien sûr ! L'affichage du Super-Oric est tellement rapide que pour faire un petit dessin animé, il faudrait ajouter une brève temporisation à chaque ligne !

Voyons les deux solutions suivantes :

Premier programme, qui introduit une sorte de GET R$ adapté au joypad et où le WAIT 50 permet de calmer un peu les ardeurs du Super-Oric, trop rapide pour mes gros doigts:

```
100 CLOADTEXT,"ECRTXT1"
110 GOSUB 1000
120 CLOADTEXT,"ECRTXT2"
130 GOSUB 1000
140 CLOADTEXT,"ECRTXT3"
150 GOSUB 1000
160 CLOADTEXT,"ECRTXT4"
170 GOSUB 1000
180 END
1000 J=0:REPEAT
1010 J=USR(0)
1020 UNTIL J<>0
1030 WAIT 50:RETURN
```

Programme que vous sauvez sous le nom Ecrtxtb.tap (CSAVE"ECRTXTB",AUTO) et avec lequel vous fabriquez une cartouche Ecrtxtb.swc en concaténant Base2a.bin, Ecrtxtb.tap, Ecrtxt1.tap, Ecrtxt2.tap, Ecrtxt3.tap et Ecrtxt4.txt en procédant comme décrit pour Ecrtxta.swc.

Testez cette cartouche avec l'émulateur Snes9x et vous aurez tout loisir d'admirer chaque écran avant de passer au suivant en appuyant sur la barre d'espace.

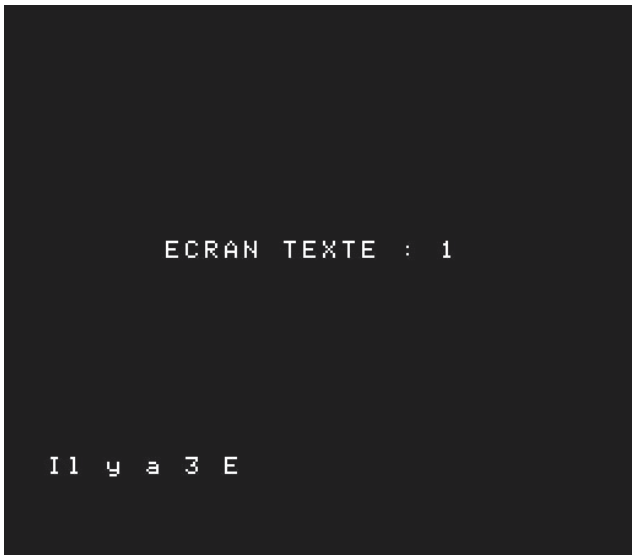
Deuxième programme, style dessin animé :

```
100 CLOADTEXT,"ECRTXT1"
110 WAIT 50
120 CLOADTEXT,"ECRTXT2"
130 WAIT 50
140 CLOADTEXT,"ECRTXT3"
150 WAIT 50
160 CLOADTEXT,"ECRTXT4"
```

```
170 WAIT 50
180 END
```

Programme que vous sauvez sous le nom Ecrtxtc.tap tap (CSAVE"ECRTXTC",AUTO) et avec lequel vous fabriquez une cartouche Ecrtxtc.swc en concaténant Base2a.bin, Ecrtxtc.tap, Ecrtxt1.tap, Ecrtxt2.tap, Ecrtxt3.tap et Ecrtxt4.txt en procédant comme d'habitude.

Testez cette cartouche et vous aurez à peine le temps de voir défiler les chiffres 1, 2, 3 et 4 attestant du chargement ultra rapide des écrans Texte. Lorsque nous aurons vu la redéfinition des caractères, cela donnera peut-être des idées aux amateurs d'animations…



LA COMMANDE SCRN(X,Y)

La commande SCRN(X,Y) classique renvoie le code Ascii du caractère de coordonnées XY, soit un octet. Dans l'écran Texte du Super-Oric, chaque caractère est caractérisé par deux octets. Le 'WORD' de 16 bits formé par ces deux octets a été décrit par Fabrice dans l'article cité. Sachez seulement ici que les 10 bits de poids faibles forment un code Ascii étendu, permettant de définir 1024 caractères. Les autres bits définissent divers attributs du caractère, notamment le numéro de palette qui lui est associé. La commande SCRN(X,Y) du Super-Oric renvoie donc la paire d'octets caractérisant le caractère de coordonnées XY.

Nous allons utiliser la commande SCRN(X,Y) pour compter le nombre de lettre 'E' présentes dans l'écran n°1. Faisons comme si nous ne connaissions pas les bits d'attributs et intéresserons-nous aux 10 bits du code Ascii en effectuant un AND #3FF, qui permet de ne retenir que ces 10 bits de poids faible. Voici notre petit programme, qui explore les 28 lignes de 32 caractères de l'écran :

```
100 CLOADTEXT,"ECRTXT1"
110 FOR Y=0 TO 27
120 FOR X=0 TO 31
130 CAR=SCRN(X,Y) AND #3FF
140 IF CAR=#45 THEN N=N+1
150 NEXT:NEXT
160 PRINT@2,22;"Il y a";N;"E"
170 WAIT 100:END
```

Sauvez avec CSAVE"ECRTXTD",AUTO et fabriquez la cartouche Ecrtxtd.swc en concaténant les fichiers Base2a.bin, Ecrtxtd.tap et Ecrtxt1.tap. Vous devez obtenir la même chose que ci-contre.

LA COMMANDE PLOT X,Y,WORD

Il y avait à l'origine deux syntaxes PLOT X,Y, A$ et PLOT X,Y,N. La première syntaxe fonctionne de manière inchangée. Mais l'expression numérique de la deuxième syntaxe doit maintenant être un WORD au lieu d'un octet. Nous allons illustrer ce fonctionnement en remplaçant le caractère '1' de notre écran Texte par le caractère '2'. Pour ce faire, il suffira de PLOTer le WORD de valeur #2032 à la place du WORD #2031. Voici notre petit programme qui nous affichera en outre les coordonnées du caractère '1' qu'il aura trouvé et remplacé par un '2' :

```
100 CLOADTEXT,"ECRTXT1"
110 FOR Y=0 TO 27
120 FOR X=0 TO 31
130 CAR=SCRN(X,Y) AND #3FF
140 IF CAR=#31 THEN X1=X:Y1=Y
150 NEXT:NEXT
160 PLOT X1,Y1,#2032
170 PRINT@2,22;"X=";X1;"et Y=";Y1
180 WAIT 100:END
```

Procédez comme ci-dessus pour fabriquer la cartouche Ecrtxte.swc et testez. Voici ce que vous obtiendrez. A bientôt…