

Mise au Point Software pour Cartouche Super-Oric

par André C. et Fabrice F.



De André C. :

Cher Fabrice, peux-tu venir à mon secours? J'essaie de mettre un programme Basic dans une cartouche Super-Oric, en utilisant Zsnes pour tester si ça marche. Mais ça ne marche pas, je ne suis même pas capable de faire PRINT«OK»! J'ai examiné les deux fichiers Superoric.swc et Supersoko.swc que tu m'avais envoyés, ainsi qu'un certain nombre d'autres .swc et .smc (quelle différence?) et le contenu de la Flash 29F010 [NDLR: Cartouche Super-Oric réelle prêtée par Fabrice].

Sauf pour la Flash, les fichiers .swc et .smc comportent une entête de #200 octets dont seuls les premiers sont différents de #00. Ces entêtes varient selon les fichiers et je n'ai pas réussi à les comprendre.

Les fichiers Superoric.swc et Supersoko.swc comportent ensuite un bloc de code de #231 octets (#28E pour la Flash), puis la Rom 1.1 modifiée (celle de la Flash a un code modifié, mais les mots clés sont normaux), puis des #00 jusqu'à l'offset #8200 (#8000 pour la Flash qui n'a pas les #200 octets d'entête) et enfin une série de fichiers .tap. Je croyais, qu'il suffirait de concaténer le programme .tap (que l'on a mis au point) à la suite des #8200 premiers octets, mais cela ne marche pas. De même si je colle l'entête de Superoric.swc et Supersoko.swc (identiques) devant le contenu de la Flash, ça ne marche pas. En fait la 1ere partie (encre rouge) marche, mais la suite (pastilles de couleurs sur fond cyan) plante.

Pourrais-tu me donner quelques indications, pour que je puisse continuer mes essais? Merci d'avance. Bien amicalement. André

Réponse de Fabrice F. : Cher André, tu vas dire que je change d'avis comme de chemise, mais apparemment les dernières versions de Zsnes marchent moins bien que celles d'avant, alors que Snes9X marche beaucoup mieux... je ne crois pas que ce soit un header mal interprété qui fait que la démo des pastilles de couleurs plante : Je crois plutôt que c'est la dernière version de Zsnes qui a un problème (j'ai constaté ton problème avec Zsnes, alors que ça marche avec l'autre émulateur Snes9x). En plus, Zsnes laisse des morceaux de droite bizarres dans la première partie de la démo...

Je vais néanmoins essayer de te répondre sur le sujet des headers SWC et SMC.

La première chose qu'il faut savoir, c'est que ces headers de 512 octets n'existent évidemment pas dans les cartouches SNES, mais sont apparus avec les copieurs de cartouches Super MagiComm (SMC) et Super Wild Card (SWC), pour que le copieur puisse se configurer en fonction de ce header. Comme les copieurs de cartouches rajoutaient en standard ces headers aux images des jeux copiés, les émulateurs sur PC ont ensuite gardé ces formats.

Bref, pour tes cartouches Flash, tu ne dois pas mettre d'entête avant le code, il suffit de concaténer tes fichiers .tap après les 32K (#8000).

J'admets que le flashage n'est pas très pratique puisqu'il faut découper le résultat en morceaux de 16K et les flasher dans le bon ordre avec la cartouche programmable pour Telestrat...

Pour les émulateurs, après avoir collé les fichiers .tap, il vaut mieux avoir un fichier dont la taille est une puissance de 2 (sans le header). En effet, les émulateurs se basent sur les informations données à la fin de la première banque de l'image de rom, plutôt que sur celles du header :

Pour une cartouche de type «LO-ROM» (banques de 32Ko), on a :

En 7FC0, le nom du jeu (21 octets).

En 7FD5, le type de cartouche : HIROM/LOROM, FASTROM/SLOWROM...

En 7FD6, le type de contenu (0 = ROM only).

En 7FD7, la taille de la rom (\$08 = 2 Mbit, \$09 = 4 Mbit, \$0A = 8 Mbit, \$0B = 16 Mbit, etc.). Le problème c'est qu'on fait ici des roms plus petites que la normale, et il me semble que mon copieur ne reconnaissait pas des tailles plus petites (j'ai un SWC DX2, comme Romu je crois). Il faudrait que je revérifie... tu remarqueras que j'ai indiqué une taille de 2 Mo dans la rom Super-Oric.

En 7FD8, la taille de la ram de sauvegarde (0 = 0 bits)

En 7FD9, le pays (ça joue pour le mode d'écran émulé,

NTSC ou PAL).

En 7FDA, le code de la licence Nintendo pour les sociétés agréées.

En 7FDB, un numéro de version du jeu.

En 7FDC-7FDD, le complément à 1 du checksum.

En 7FDE-7FDF, le checksum proprement dit, il est

```
0 - LOW BYTE OF 8K-BYTES PAGE COUNTS.
1 - HIGH BYTE OF 8K-BYTES PAGE COUNTS.
2 - EMULATION MODE SELECT.
   BIT 76543210 :
       X       : 1=RUN IN MODE 0. (JUMP $8000)
       X       : 0=LAST FILE OF THE GAME. (MULTI FILE LOADING)
       X       : 0=MODE 1, 1=MODE 2. (SRAM MAPPING)
       X       : 0=MODE 20, 1=MODE 21. (DRAM MAPPING)
       XX      : 0=SRAM OFF, 1=SRAM 16K, 2=SRAM 64K, 3=SRAM 256K
       X       : 0=RUN IN MODE 3, 1=RUN IN MODE 2. (JMP RESET)
       X       : 0=DISABLE, 1=ENABLE. (EXTERNAL CARTRIDGE MEMORY
                               IMAGE AT BANK 20-5F, A0-DF
                               IN SYSTEM MODE 2, 3).

3-7 - RESERVED. (SHOULD BE '00').
8 - FILE ID CODE 1. (SHOULD BE 'AA').
9 - FILE ID CODE 2. (SHOULD BE 'BB').
10 - CHECK THIS BYTE IF ID 1 AND 2 MATCH.
     '02': MAGIC GRIFFIN GAME FILE. (PC ENGINE)
     '03': MAGIC GRIFFIN SRAM DATA FAILE.
     '04': SWC&SMC GAME FILE. (SUPER MAGICOM)
     '05': SWC&SMC PASSWORD, SRAM DATA, SAVER DATA FILE.
     '06': SMD GAME FILE. (MEGA DRIVE)
     '07': SMD SRAM DATA FILE.
11-511 - RESERVED. (SHOULD BE '00').
```

Toutefois, j'ai déjà eu des déconvenues en suivant cette description pour mon émulateur Midway sur SNES...

Mais en fait, puisque tu n'utilises pas de copieur, tu n'as absolument pas besoin de ce header de 512 octets : les émulateurs arriveront à déterminer le type de cartouche d'après les informations en fin de la première banque, et tu n'en as pas besoin n'ont plus pour flasher sur tes cartouches Super-Oric... Si j'ai rajouté un header, c'est

vérifié par les émulateurs et il sera forcément incorrect de la façon dont on construit aujourd'hui les cartouches Super-Oric (par simple concaténation de fichiers .tap quelconques)...

Voici enfin ce que j'ai comme description du header SWC :

parce qu'au début, les émulateurs ne marchaient pas très bien avec la rom Super-Oric, et je testais plus rapidement sur la SNES réelle en mettant un fichier SWC sur disquette et en le chargeant sur mon copieur DX 2...

Voilà, voilà, donc en résumé : essaie donc Snes9x (j'utilise la version 1.39, il y en a peut-être une encore plus récente), et enlève ces headers :-)

Amitiés, Fabrice

Software

Quel processeur se trouve dans mon Oric ?

par Fabrice F.

Parfois je ne sais plus quel processeur se trouve dans mes Oric, alors plutôt que d'ouvrir le boîtier, j'utilise le petit programme suivant:

```
10 AD=#110:REPEAT:READ V$:POKE AD,VAL(«#»+V$):AD=AD+1:UNTIL V$=«»
20 CALL#110:ON PEEK(0) GOTO 100,200,300:END
100 PRINT«NMOS 6502 DETECTED»:END
200 PRINT«CMOS 6502 DETECTED»:END
300 PRINT«WDC 65802/65816 DETECTED»:END
999 DATA A2,01,A9,55,89,AA,D0,06,E8,E2,02,D0,01,E8,86,00,60,«»
```

La petite routine en assembleur utilise le fait que l'opcode \$89 n'est pas utilisé par le 6502 Nmos (celui qui équipe normalement les Oric), alors qu'il signifie BIT #immédiat sur le 6502 Cmos. Certains des opcodes non utilisés du 6502 Nmos bloquent le CPU (l'opcode \$02 par exemple), d'autres ont en fait un comportement non documenté, et l'opcode \$89 correspond à un NOP sur deux octets (l'octet suivant est récupéré mais non utilisé). Pour différencier les 6502 Cmos du 65816 de Western Design Center (bien que dans ce cas de figure je sais très bien où se trouve mon unique module 65816), j'utilise l'opcode \$E2 qui est non utilisé sur le 65C02 (les opcodes non utilisés du 65C02 ont tous un comportement de NOP), alors qu'il correspond à l'instruction SEP des 65802/65816. Le petit programme ci-dessus marche aussi avec Euphoric et affiche correctement le processeur émulé, mais en fait, il faudrait raffiner le programme pour tenir compte des variantes constructeur sur les processeurs 6502 Cmos.