

# Le Journal du Soft ( 9 ) : la ROM CMB

par André Chéramy et Claude Sittler

Dans le "Journal du Hard" n°9, nous vous avons exposé le support hard de cette ROM CMB: un simple fil entre la broche n°15 (PB5) du VIA 6522 et la broche n°27 d'une EPROM 27256 remplaçant la 27128 habituelle.

Tout ceci avait été nécessité à l'origine par un cahier des charges bien précis: il fallait transformer l'Atmos en régie de télévision portable. Pour échanger des images d'un bout de la France à l'autre, il fallait disposer non seulement d'un émetteur de télévision, mais aussi d'un système de génération d'images. Tout électronicien éclairé peut fabriquer bien des choses, mais pas une

caméra. L'Oric a donc été chargé de générer des images et sa sortie vidéo a été connectée à l'émetteur TV. Pour améliorer la portée de cette émission, il était nécessaire de grimper sur un point haut, d'où la nécessité de disposer d'un système portable. C'était courageux de la part d'un lycéen (François Sittler), ce fut une remarquable réussite.

Nous ne vous donnerons pas le listing complet de la ROM CMB. Ça prendra plus de 20 pages. D'ailleurs peu d'entre vous disposent d'un émetteur de télévision. Toutefois, nous mettons à la disposition de tous ceux qui nous le demanderont une EPROM CMB à mettre dans votre Oric (voir les modalités d'implantation dans le "Le Journal du Hard" n°9). Nous allons quand même décrire en quoi consiste ce système et surtout donner des indications précises pour que ceux d'entre vous qui se sentiront stimulés puissent écrire d'autres "Cartouches PB5" (jeux, utilitaires, etc.).

## EN QUOI CONSISTE LA ROM CMB ?

C'est en fait une double ROM. La partie haute ou ROM1 (en service quand PB5 est mis à 1) est une ROMV1.1 légèrement modifiée. La partie basse ou ROM0 (en service lorsque PB5 est à 0) contient tout un "package" qui sera recopié en RAM:

1) Un gros programme BASIC, situé de C000 à D8B9 en ROM0, qui sera copié de 0501 à 1DBA en RAM par une routine de la ROM1 lors du boot.

2) Un programme en LM, situé de D900 à DFFF en ROM0, qui sera copié de 4B90 à 528F en RAM lors de l'exécution du programme BASIC ci-dessus, lors du boot.

3) Enfin, une mire, située de E000 à FFFF en ROM0, qui sera copiée de A000 à BFFF en RAM, c'est à dire dans l'écran HIRÉS, lorsque la fonction "Z" sera activée.

Sur le haut de la montagne, pas besoin de lecteur de K7 ou de disquette: tout est inclus dans la machine. Dès la mise sous tension (sur batterie) le programme BASIC est copié en RAM, puis exécuté. Le 1<sup>er</sup> écran de CMB apparaît sans autre intervention, il n'y a plus qu'à se laisser guider par les menus. Vous trouverez ci-joints quelques exemples des écrans auxquels on accède. Le menu principal intitulé "COMMANDES DISPONIBLES" liste tout ce que l'on peut faire. Certains items sont propres aux radio-amateurs: indicatif, locator, numéro de contrôle etc. Tout ce qui apparaît sur l'écran est envoyé par l'émetteur TV: initialisation du contact, mires, messages scrollés (et oui, ce ne sont donc pas uniquement des images fixes). Lorsqu'une des commandes est en cours d'exécution, il est possible de changer la couleur du fond par les touches Ø et la couleur de l'encre par les touches ù. Après exécution de chaque commande, on peut revenir au menu principal en tapant sur <espace> (ou sur <funct> dans le cas de la mire "Z"). Tout cela peut être édité, notamment les messages qui vont défiler sur l'écran. C'est du joli travail!

## LA ROM1 OU PARTIE HAUTE DE LA ROM CMB

Nous la décrivons in extenso, car c'est le coeur du système à "Cartouche PB5". Pour créer vos propres cartouches, placez votre propre "package" dans la ROM0 et accolez-y une ROM1 où vous aurez simplement modifié en F855 la taille du fichier à copier de la ROM0 vers la RAM. Si vous désirez lancer un programme BASIC au boot, celui-ci devra commencer par la ligne suivante: 0 DOKE#9C,#1234:DOKE#9E,#1234:HIMEM#97FF:RESTORE. Après avoir sauvé votre BASIC, il faudra remplacer l'adresse fictive #1234 par l'adresse réelle de l'octet qui suit les 3 #00 marquant la fin du programme BASIC. Aidez-vous d'un moniteur.

### **1) "CONFIGURER LE VIA POUR LE TRAVAIL K7"**

En E77D, la valeur #40 (0100 0000) doit être remplacée par #60 (0110 0000). En effet, cette valeur #40 est pokée en 0300 afin de fermer le relais de télécommande K7 (mise à "1" du b6 de 0300, qui tire la sortie PB6 à +5V). (Rappel: les bits d'un octet sont numérotés de b0 à b7). Mais ce faisant, le b5 de 0300 est aussi mis à "0", ce qui génère une mise à la masse intempestive de PB5. Il faut donc poker #60 au lieu de #40 pour rester sur la ROM1.

### **2) INSERER UNE ROUTINE SUPPLÉMENTAIRE**

A la fin du processus de boot (routine COLD START située de ECC8 à ED85), se trouve un JMP C4A8 qui rend la main à l'interpréteur BASIC. En ED83, Ce JMP C4A8 (#4C, #A8 et #C4) est remplacé par JMP F844 (#4C, #44 et #F8). La nouvelle routine en F844 se charge de copier en RAM le programme BASIC de la ROM0 et d'en lancer l'exécution.

### **3) REMPLACER "GÉNÉRER LES CARACT. ALTERNÉS"**

Il fallait bien trouver une place pour implanter la nouvelle routine. Les caractères alternés n'étant pas utilisés par le programme CMB,

```
@ F6EKT & TCS

LORSQUE LE PROGRAMME TOURNE,
IL EST POSSIBLE DE CHANGER:
LA COULEUR DU FOND PAR LES TOUCHES:
- CURSEUR DROITE ET CURSEUR GAUCHE.
LA COULEUR DE L'ENCRE PAR LES TOUCHES:
- CURSEUR HAUT ET CURSEUR BAS.

PATIENTEZ UN MOMENT SUP...
```

```
BONJOUR

VOTRE INDICATIF : F6CMB
VOTRE NOMBRE DE CONTROLE : 2512
VOTRE QRA LOCATOR : ? HENRI
QTR (HH-MM-SS) : ? 12-00-00

TAPER UN ESPACE POUR CONTINUER
12 00 03
```

```
COMMANDES DISPONIBLES :
MIRE F6CMB/ATV      Z
FIN DE QSO :       F
INDICATIF :        I
LOCATOR :          L
MIRE COULEUR :     M
NUMERO DE CONTROLE : N
INDICATIF QSO :    Q
73' DE F6CMB :     S
BOITE A MESSAGE :  B
ACQUISITION :     A
COMMANDES :        " "
```

12 00 12

le sous-programme "générer les caractères alternés" qui se trouvait de F816 à F864 a donc été remplacé par les éléments suivants:

**a) Neutralisation du sous-programme "Générer les caractères alternés"**

**F816** 60 RTS qui est donc réduit au minimum!

**b) "Mise en place en page 4 du sous-programme de copie"**

**F817** A6 04 LDX 04 nombre de pages à copier  
**F819** 78 SEI interdire les interruptions  
**F81A** A0 1A LDY #1A index pour copier #1B = 27 octets  
**F81C** B9 29 F8 LDA F829,Y lit octets F829 à F843 en ROM1  
**F81F** 99 00 04 STA 0400,Y les copie de 0400 à 041A en RAM  
**F822** 88 DEY vise octet suivant  
**F823** 10 F7 BPL F81C reboucle tant qu'il en reste  
**F825** 20 00 04 JSR 0400 continue en page 4, avec  
**F828** 60 RTS X = nombre de pages à copier

**c) Code du sous-programme de copie implanté en page 4**

**F829** A9 90 LDA #90 1001 0000 dans registre de sortie  
**F82B** 8D 00 03 STA 0300 PB5 niveau bas = ROM0 validée  
**F82E** A0 00 LDY #00 index de copie  
**F830** B1 00 LDA (00),Y lit un octet selon adresse en 00/01  
**F832** 91 02 STA (02),Y et l'écrit selon adresse en 02/03  
**F834** C8 INY vise l'octet suivant et reboucle  
**F835** D0 F9 BNE F830 tant que 256 octets pas copiés  
**F837** E6 01 INC 01 page suivante en lecture  
**F839** E6 03 INC 03 page suivante en écriture  
**F83B** CA DEX décrémente nombre pages à copier  
**F83C** D0 F2 BNE F830 reboucle tant qu'il en reste  
**F83E** A9 B0 LDA #B0 1011 0000 dans registre de sortie  
**F840** 8D 00 03 STA 0300 PB5 niveau haut = ROM1 validée  
**F843** 60 RTS et retourne

**d) Routine "Implantation du programme BASIC"**

**F844** A9 00 LDA #00 adresse en 00/01 = C000 =  
**F846** 85 00 STA 00 adresse du début de la ROM0,  
**F848** A9 C0 LDA #C0 valide si PB5 à la masse  
**F84A** 85 01 STA 01 initialise adresse en 02/03avec

**F84C** A9 01 LDA #01 0501, adresse du début du  
**F84E** 85 02 STA 02 BASIC en RAM, ce programme  
**F850** A9 05 LDA #05 fera #18B9 octets de long  
**F852** 85 03 STA 03 soit #19 "pages" en arrondissant  
**F854** A2 19 LDX #19 nombre de pages à copier  
**F856** 20 19 F8 JSR F819 place page 4 la routine de copie  
**F859** 58 CLI autorise les interruptions  
**F85A** 20 08 C7 JSR C708 initialise les pointeurs BASIC  
**F85D** 4C C1 C8 JMP C8C1 lance exécution du progr BASIC  
**F860** EA EA EA EA EA 5 NOPs pour faire propre

**4) MODIFIER "INITIALISER LE VIA 6522"**

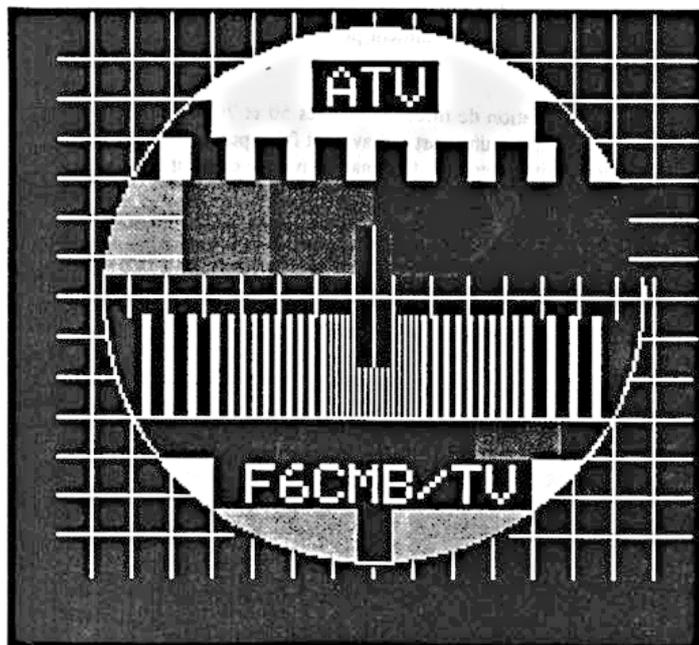
Ce sous-programme est situé de F9AA à F9C8. Lors d'un cold start, la broche n°34 est mise à la masse et tous les ports sont mis en entrée. C'est l'état de départ. Le port A est utilisé pour l'imprimante, le son et le clavier. Le port B est impliqué dans des tâches complémentaires: PB0 à PB5 pour le clavier, PB4 pour le strobe imprimante, PB6 pour la télécommande K7 et PB7 pour la sortie data K7. PB5 inutilisé servira pour gérer la ligne d'adressage supplémentaire de l'EPROM 27256, de telle manière que lorsque PB5 est à "0", la ROM0 soit active et que lorsque PB5 est à 1, la ROM1 soit active. Dans l'Oric-1/Atmos d'origine, PB5 n'était pas connecté. Son initialisation n'avait donc aucune importance. Maintenant, si l'on veut booter sur la ROM1, il faut que PB5 soit mis à "1" lors de l'initialisation du VIA 6522. On doit d'abord paker dans le registre data (0300) puis mettre le port en sortie (port 0302). En effet, si on mettait d'abord le port en sortie, PB5 prendrait une valeur imprévisible, selon l'état du registre 0300. Le code d'origine étant hélas inversé (initialisation du registre 0302, puis du registre 0300), il a fallu le mettre dans l'ordre souhaité.  
**F9AF** A9 B7 LDA #B7 1011 0111, le port B toujours  
**F9B1** 8D 00 03 STA 0300 en entrée ne bouge pas encore  
**F9B4** A9 F7 LDA #F7 1111 0111, configure toutes les  
**F9B6** 8D 02 03 STA 0302 lignes du port B en sortie sauf PB3 (ligne X clavier), PB5 (entre autres) passe alors au niveau haut.

**UTILISATION DE PB5 DANS UN PROGRAMME BASIC**

Le programme BASIC CMB fait appel à plusieurs reprises aux routines de la ROM1 pour copier une partie de la ROM0 en RAM. Voici un exemple, afin d'illustrer la méthode utilisée. Après appel de la commande "Z", les deux lignes suivantes copient la mire, que vous pouvez admirer ci-dessous, dans l'écran HIRES.

```
40000 PAPER0:INK7:HIRES
40005 DOKE0,#E000:DOKE2,#A000:POKE4,#20:CALL#F817
```

La méthode est simple: il suffit de doker l'adresse ROM0 de début du bloc à copier en 00, l'adresse RAM de destination en 02 et de paker le nombre de "pages" à copier en 04. Un CALL#F817 dans la ROM1 (c'est la ROM normalement validée) exécute le sous-programme décrit plus haut. Ce programme lit à l'adresse 04 le nombre de "pages" à copier, met en place en page 4 le programme de copie, lance ce programme, qui a son tour valide la ROM0, copie le bloc prévu selon les indications présentes en 00 et 02 et finalement valide à nouveau la ROM1.



La ROM0 est en fait utilisée comme un réservoir, un peu à la manière d'une K7 ou d'une disquette, où le programme principal puise les ressources dont il a besoin. On pourrait imaginer de mettre au point un ROM0 bourrée de routines LM supplémentaires telles que routines graphiques etc., un peu à la manière du MacIntosh.

Plusieurs remarques peuvent encore être faites (voir "Le Journal du Hard" n°9):

- 1) Le programme de "swap" doit être exécuté en RAM, mais pas forcément dans la page 4. Comme il est très petit (27 octets), il peut trouver sans problème un autre site en RAM, par exemple de #BFE0 à #BFFF.
- 2) Moyennant une programmation adaptée, il est possible d'exécuter directement des routines dans la ROM0. Mais il faut se rappeler qu'il est très délicat de faire appel directement de l'une des deux ROM aux routines de l'autre, sans passer par une zone d'exécution en RAM.