Trucs et Astuces Programmation (4)

Personnalisation du BRK de l'Oric *par André C.*

Nous avons entrevu dans un précédent article ("Trucs et Astuces Programmation (3) : Le BRK de l'Oric"), comment planter un Oric. Oups, comment placer un BRK sans planter l'Oric.
Je rappelle que dans ce contexte j'utilise le mot Oric pour désigner un Atmos ou un Oric-1 nu (sans lecteur de disquette). Pour simplifier mon propos, les adresses que j'indiquerai sont principalement celles de l'Atmos.
Mais "L'Oric à nu" de Fabrice Broche vous permettra facilement de trouver la correspondance pour l'Oric-1.

Ouh ! Le vilain !

Je ne sais pas si vous vous en êtes rendu compte, mais j'ai un peu triché. En effet, l'exemple de programme en langage machine que j'ai utilisé dans l'article précédent, était appelé par un CALL et le RTS n'était pas loin derrière l'introduction du BRK. Cela garantissait un retour au prompt du Basic, sauf plantage de la machine.
La réalité était donc un peu voilée. A savoir, l'Oric ne rend pas la main, il saute le BRK+son pseudo opérande (l'octet qui suit) et tente de reprendre le fil du programme, si c'est possible, sinon il se vautre.
Comme je l'ai signalé dans l'article "La commande Break, mnémonique BRK, code #00", tous les ordinateurs ne sont pas configurés comme l'Oric. La plupart des machines retournent simplement au "prompt" du Basic ou du DOS, affichant généralement un message du genre "BREAK ON BYTE"+adresse.
Aujourd'hui, nous allons reconfigurer l'Oric pour qu'il s'arrête net sur le BRK et rende proprement la main. Pour ce faire, nous allons profiter d'un des deux passages en Ram que comporte le processus de gestion du BRK.

Résumé de la partie du processus propre à l'Oric :

Après que le processeur ait initialisé le registre PC avec l'adresse #FFFE, il saute à l'adresse indiquée par ce vecteur.
Dans le cas de l'Atmos, l'adresse indiquée en #FFFE est #0244 (en Ram), qui elle-même renvoie en #EE22 (en Rom, routine de gestion des IRQ/BRK). Cette routine teste le drapeau B du SR (Status Register).
S'il s'agit d'un BRK, elle renvoie en #024A (à nouveau en Ram) pour exécuter un simple RTI (ReTurn from Interrupt). Ce RTI remet en place les registres SR et PC qui avaient été empilés au début de la procédure et reprend donc le programme en exécutant (ou en tentant d'exécuter) le 2e octet après BRK.

Stratégie

Comme déjà indiqué, il se produit 2 passages en Ram, en #0244 (#0228 pour l'Oric-1) et en #024A (#0230 pour l'Oric-1).
Nous laisserons l'Oric effectuer ce qui était prévu lors du 1er passage en Ram (un appel à la routine de gestion des IRQ/BRK en Rom). Cette routine, qui doit pouvoir continuer à gérer les IRQ normalement, se contente en cas de BRK d'aller sur le RTI en Ram. Aucune gêne pour nous donc.
C'est au 2ème passage en Ram que nous allons nous attaquer. Ce passage en Ram consiste en un simple RTI. Coup de chance, juste après se trouvent de 2 octets non utilisés, de quoi remplacer ce RTI par un JMP+adresse. Ça s'appelle un détournement de fonds. Oups, je veux dire que le détournement est une stratégie de fond pour tout bricoleur de software.
En fait, tout cela n'est pas de la chance : Les concepteurs de la Rom de l'Oric auraient pu tout mettre en Rom (la taille du code en Ram est très faible). Mais ils ont délibérément laissé ouverte une possibilité de "customisation". Les deux octets de libre après le RTI ne sont pas là par hasard, mais pour pouvoir appliquer un patch. Nous allons donc profiter de cette opportunité.

Où trouver un peu de place pour cacher nos fonds.

Ah ! Zut, j'y tiens ! Je voulais dire "Un peu de place pour mettre notre code additionnel" (notre patch). Si la page zéro de la mémoire est bourrée, par contre il reste de la place en page deux. "L'Oric à nu" montre par exemple une belle rangée de "inutilisé" de #02C4 à #02DE, soit 27 octets de libres ! Nous en avons besoin de 16, c'est Byzance !

Voici donc ce que je vous propose :

En #02C4, implanter le code suivant :

```
02C4 STA 0B   Sauvegarde une copie de l'accumulateur en #0B, page zéro
02C6 PLA      Récupère dans l'accumulateur l'octet sur la pile
02C7 PHA      Et le remet en place (donc en prend une copie)
02C8 AND #10  N'en garde que le bit 4
02CA BEQ 02D1 S'il est à 0, saute jusqu'en #02D1
02CC SEC      Sinon, force Carry à 1
02CD CLI      Autorise les interruptions
02CE JMP C976 Exécute la commande END
02D1 LDA 0B   Récupère la valeur d'origine de l'accumulateur
02D3 RTI      Remet en place le RTI qui avait été détourné
```

En #024A (#0230 pour l'Oric-1), remplacer le RTI par JMP#02C4 pour lancer le patch ci-dessus.
Notez que pour notre patch, les seules différences entre l'Oric-1 et l'Atmos sont l'adresse dans la commande END, soit #C976 pour l'Atmos et #C944 pour l'Oric-1 et l'adresse du RTI à modifier, soit #024A pour l'Atmos et #0230 pour l'Oric-1.

Explications

La commande END met de l'ordre dans les pointeurs et registres, avant d'afficher le message "BREAK" et de rendre la main au prompt du Basic. Quelle est l'information qui se trouvait sur la pile et dont on a testé le bit 4 ? La réponse se trouve dans le code exécuté avant l'intervention de notre détournement, à savoir la routine de gestion des IRQ/BRK en #EE22 en Rom (#ED09 pour l'Oric-1). Il s'agit bien sûr du Status Register, dont le bit 4 est le fameux drapeau B. Notre code additif exécute END si c'est un BRK et RTI si c'est une IRQ. Simple non ? Le déroulé complet de la procédure de gestion du BRK étant un peu compliqué, le plus simple était de ne toucher à rien, simplement lire le drapeau B et :

- soit remettre tout comme c'était, dans le cas d'une IRQ,
- soit exécuter la commande END, dans le cas d'un BRK.

Je suis arrivé sur la commande END en cherchant dans la Rom une routine qui affiche le message BREAK (le "BREAK ON BYTE" appartient à Sedoric, dommage…). Après quelques essais, la recette finale s'est révélée très simple.
Attention toutefois, en #024A se trouve normalement un RTI suivit de deux #00. Si on POKE en premier un JMP à la place du RTI, on aura (pour quelques secondes) un JMP#0000. Or les interruptions pleuvent sans arrêt (à l'échelle humaine) pour la gestion du clavier, du clignotement du curseur, etc. Je me suis fait piéger : mon beau greffon plantait 3 fois sur 4 (mais pas à tous les coups) ! Il faut donc commencer par implanter les zones mémoires inutilisées, à savoir de #02C4 à #02D3 et de #024B à #024C. Cela restera sans effet tant que l'on n'aura pas changé, en tout dernier, le RTI en JMP. D'où l'ordre un peu baroque de mon programme.

```
 Salut les gars !                                  CAPS
ORIC EXTENDED BASIC V1.1fr
1983 TANGERINE

 37631 BYTES FREE

Ready
 100 FOR I=0 TO 15:READ D:POKE#02C4+I,
D:NEXT
 110 DATA #85,#0B,#68,#48,#29,#10,#F0,
#05
 120 DATA #38,#58,#4C,#76,#C9,#A5,#0B,
#40
 130 DOKE#024B,#02C4:POKE#024A,#4C
CSAVE"BRKENDBAS"

Ready
RUN

Ready
CLOAD"SALUTBIN"

Ready
CALL#9812

Ready
```

Fig 1

Au travail maintenant

Vous pouvez utiliser un moniteur sur cassette, comme MONASM de Vismo ou, mieux, le mini-moniteur d'André Chénière qui est autorelogeable et que vous trouverez sur :
<http://andre.cheramy.net/telechargement/Programmes/choix.htm>
Le fichier Chénière André.zip fait 31 Ko. Les manuels d'André Chénière sont disponibles sur :
<http://andre.cheramy.net/telechargement/Librairie/wxcv.htm>

- Atmos-Oric1, manuel de référence 1 de André Chénière (11.4 Mo).
- Atmos-Oric1, manuel de référence 2, travaux pratiques de André Chénière (19.6 Mo).

Pour ma part, je me suis contenté de fabriquer le petit chargeur Basic suivant :

```
100 FOR I=0 TO 15:READ D:POKE#02C4+I,D:NEXT
110 DATA #85,#0B,#68,#48,#29,#10,#F0,#05
120 DATA #38,#58,#4C,#76,#C9,#A5,#0B,#40
```

La 1ère chose à faire après avoir tapé ce listing est de le sauver, par exemple sous le nom BRKENDBAS (figure 1, page précédente). Puis il faut mettre le code en place avec un RUN. Enfin charger le programme de test utilisé dans le dernier article : SALUTBIN.

Un CALL#9812 permet de s'assurer que tout fonctionne toujours normalement. La figure 1 (page précédente) montre que le message "Salut les gars !" s'affiche bien sur la ligne service.

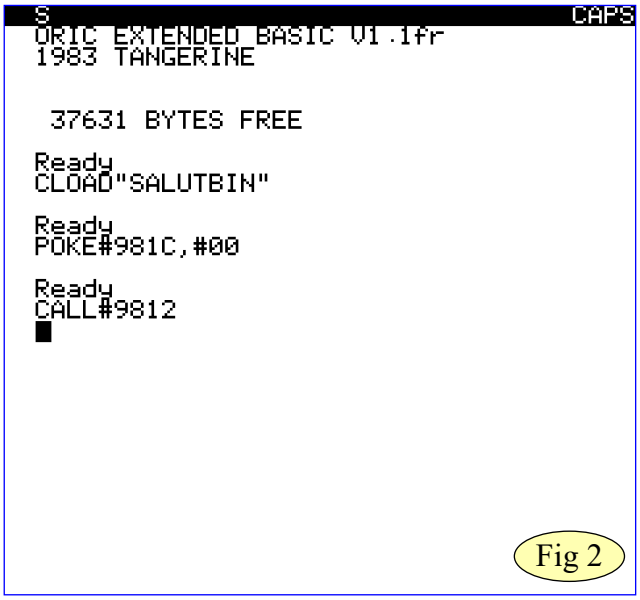

Fig 2

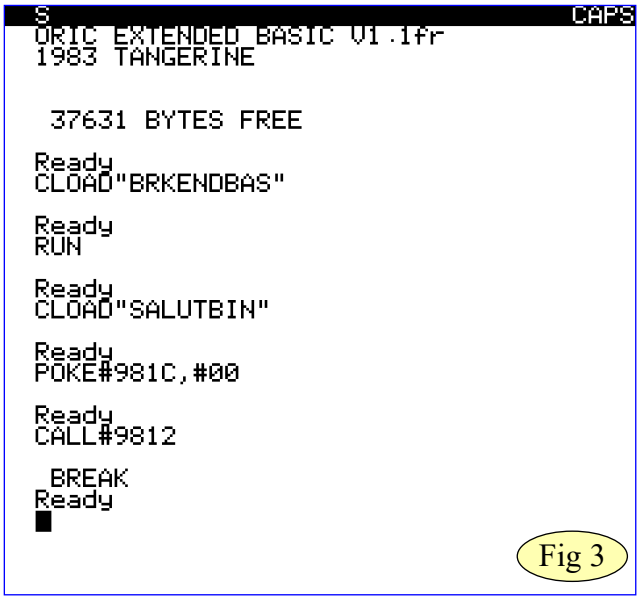

Fig 3

Passons maintenant aux choses sérieuses :

Nous allons tester ce qui se passe quand le processeur de notre Oric patché rencontre un BRK.

Pour cela nous allons placer le BRK en #981C, ce qui provoquait un beau plantage avec un Oric dans sa configuration d'origine (la machine poursuivait l'exécution du programme de test sur l'octet #F5, qui ne correspond à aucune commande langage machine) (figure 2, ci-dessus).

Comme à notre habitude, rebootons l'Oric de frais.

Les commandes CLOAD"BRKENDBAS" et RUN mettent le patch en place.

La commande CLOAD"SALUTBIN" charge le programme de test.

La commande POKE#981C,#00 met le BRK en place.

Un petit CALL#9812 et voilà ça marche, la première lettre "S" du message "Salut les gars !" s'est bien affichée sur la ligne service (figure 3, ci-dessus).

Et, beaucoup plus intéressant pour nous ici, un gentil retour au Ready s'est effectué sans plantage!

Notez que vous pouvez remplacer l'appel à la commande END par un appel à une routine de votre cru.

Essayer par exemple de récupérer l'octet qui suit le BRK (pseudo opérande qui pourrait être un numéro de routine) pour lancer la routine spécifique portant ce numéro (voir l'excellent article "Telemon" de Jérôme D. dans le CEO-mag n°323, pages 10 et 11).

Voilà, ce sera tout pour le BREAK pour l'instant.

En patchant votre Oric, vous aurez maintenant un comportement plus simple de votre machine et pourrez utiliser des BRK sans crainte.

N'hésitez pas à faire un break de temps en temps, c'est bon pour la santé !

Et bonnes découvertes…