# Trucs et Astuces Programmation (3)

## Le BRK de l'Oric

*par André C.*

Nous avons entrevu dans un précédent article ("La commande Break, mnémonique BRK, code #00"), les problèmes engendrés par la gestion du BRK propre à l'Oric. En effet, le registre PC empilé lorsque le processeur rencontre un BRK pointe alors sur l'adresse BRK+2, comme si l'instruction BRK occupait deux octets.

Dans ce qui suit, je traite du BRK de l'Oric (nom générique pour désigner l'Oric-1 et l'Atmos dans la configuration de base, sans lecteur de disquette). Les routines de gestion de IRQ/BRK sont identiques pour l'Oric-1 et pour l'Atmos., quoique leur adresse en Rom soient différentes. Pour simplifier mon propos, je ne donnerai donc que les adresses mémoire de l'Atmos.

**Comment planter votre Oric**

Il "suffit" d'insérer un BRK n'importe comment ! Mais sauf si vous travaillez sur le fichier source (listing assembleur), il est difficile, voire impossible, d'insérer votre BRK, d'autant plus qu'il serait prudent d'insérer non pas un, mais deux octets, soit un BRK et un NOP (#00,#EA), comme nous l'a conseillé Yann L. (CEO-mag n°324 page20, avril 2017), ou un BRK et un opérande (cf. voir l'excellent article de Jérôme D. dans le CEO-mag n°323, pages 10 et 11).

Sauf si on le fait en parfaite connaissance de cause, insérer un seul octet (c'est-à-dire #00) conduira à un plantage garanti. Le cas général consistera donc, non pas à insérer un BRK, mais à remplacer un octet du programme par un BRK. Reste à choisir quel(s) octet(s) écraser.

Tout d'abord l'octet #00 (BRK) ne doit pas être mis à la place d'un opérande. Par exemple, avec STA BB82,Y (soit #99,#82,#BB) mettre le BRK à la place du poids fort ou à celle du poids faible de l'adresse produira à coup sûr autre chose que l'effet escompté. Dans certains cas, un plantage est certain.

Il faut donc mettre le BRK à la place d'un opérateur (une commande langage machine). Dans l'exemple ci-dessus on aurait #00,#82,#BB à la place de #99,#82,#BB. Mais ce n'est pas tout.

Dans le cas de l'Oric, le processeur retourne à l'adresse empilé initialement, à savoir celle de BRK+2. Et là, c'est potentiellement le plantage, car le BRK occupe un seul octet et le processeur reprend deux octets plus loin, c'est-à-dire potentiellement au milieu d'une instruction.

Le plantage va se produire sauf si l'instruction suivante ne comporte qu'un octet. Par exemple, si on avait la séquence INY SEI STA BB82,Y (soit #C8,#78,99,#82,#BB), remplacer le INY par BRK (soit #00,#78,99,#82,#BB) n'entrainerai pas de plantage, sauf que l'instruction SEI ne serait pas exécutée. Attention à ne pas altérer un RTS, sinon vous réabonner au plantage !

Il est donc nécessaire de faire suivre ce BRK par un ou plusieurs NOP (No Operation) pour éviter le plantage du programme. Remplacer un octet par un BRK dans un programme est dangereux, car l'octet suivant sera sauté et l'exécution reprendra probablement là où il ne faut pas, ce qui produira un plantage.

**Règles pratiques**

Pas de panique, je me suis étendu sur la question pour que vous compreniez bien les 3 règles qui suivent et qu'il suffit d'appliquer bêtement :

- Si on veut insérer un BRK, il faut insérer deux octets (par ex BRK+NOP).
- Si l'on veut remplacer un octet, il faut remplacer l'ensemble opérateur + opérande par un nombre d'octets équivalent (BRK+NOPs).
- Si cet ensemble se réduit à un octet (par ex INY), il faut remplacer cet octet par BRK et tous les octets de l'ensemble suivant par un nombre de NOPs équivalent.

Par exemple remplacer les 3 octets de STA BB82,Y (soit #99,#82,#BB) par les 3 octets BRK NOP NOP (soit #00,#EA,#EA). En réalité #00,#82,#EA marcherait aussi car l'octet #82 est sauté par le processeur. Mais cela ferait moins propre.

**Travaux pratiques**

Pour vous montrer que je ne tire pas tout ça de mon chapeau, je vous invite à faire les petites expériences suivantes sur un Atmos ou un Oric-1 sans lecteur de disquette. Vous pouvez utiliser un moniteur sur cassette, comme MONASM de Vismo. Pour ma part, je me suis contenté de quelques POKEs.

Nous utiliserons notre programme fétiche (voir "Trucs et Astuces Programmation (2)") dont je rappelle l'explication du code :

```
9801 53 61 6C 75 74 20 6C 65 73 20 67 61 72 73 20 21 00 "Salut les gars !"
9812 A0 00    LDY #00    mettre la valeur zéro dans le registre Y
9814 B9 01 98 LDA 9801,Y lire l'octet présent à l'adresse 9801+Y
9817 F0 06    BEQ 981F   si c'est 0, fini, on termine en 981F
9819 99 82 BB STA BB82,Y sinon on copie l'octet en BB82+Y
981C C8       INY Y=Y+1  pour indexer l'octet suivant
981D D0 F5    BNE 9814   reboucle tant que Y ne repasse pas à zéro
981F 60       RTS        retourne au point d'appel
```

Nous expérimentons sur un Atmos ou un Oric-1 sans lecteur de disquette. Donc, commençons par sauver ce programme sur cassette. La figure 1 montre le chargeur Basic à taper, puis à sauvegarder par exemple sous le nom SALUTBAS. Lançons ensuite ce chargeur de DATA et sauvegardons le résultat avec un CSAVE"SALUTBIN",A#9801,E#981F.

Nous allons maintenant procéder à une série de petits essais. Pour chaque essai, il faudra rebooter de frais la machine et recharger SALUTBIN.

```
                                                    CAPS
ORIC EXTENDED BASIC V1.1fr
1983 TANGERINE

 37631 BYTES FREE

Ready
 100 FOR I=0 TO 30
 110 READ D:POKE #9801+I,D
 120 NEXT
 130 DATA #53,#61,#6C,#75,#74,#20
 140 DATA #6C,#65,#73,#20,#67,#61
 150 DATA #72,#73,#20,#21,#00
 160 DATA #A0,#00,#B9,#01,#98
 170 DATA #F0,#06,#99,#82,#BB
 180 DATA #C8,#D0,#F5,#60

CSAVE"SALUTBAS"

Ready
RUN

Ready
CSAVE"SALUTBIN",A#9801,E#981F

Ready
```

Fig 1

```
S                                                   CAPS
ORIC EXTENDED BASIC V1.1fr
1983 TANGERINE

 37631 BYTES FREE

Ready
CLOAD"SALUTBIN"

Ready
POKE#981D,#00

Ready
CALL#9812

Ready
```

Fig 2

## TEST 1 :

Voyons ce qui se passe si on remplace l'octet #D0 (BNE) en #981D par #00 (BRK). Soit :

```
981D D0 F5        (BNE#9814)        →     981D 00 F5        (BRK+ex#F5)
```

POKE#981D,#00 puis CALL#9812. La figure 2 montre que seule la première lettre du message a bien été affichée sur la ligne service et donc que l'exécution a bien été arrêtée en #981D. Le processeur a sauté l'octet qui suit le BRK et le programme a été relancé en #981F sur le RTS. Retour au prompt du Basic comme le montre le Ready. La machine n'est pas plantée. Nous avons commencé par un bon test.

## TEST 2 :

Voyons ce qui se passe si on remplace l'octet #C8 (INY) en #981C par #00 (BRK). Soit :

```
981C C8           (INY)             ->    981C 00           (BRK)
```

POKE# 981C,#00 puis CALL#9812. La figure 3 montre que la seule première lettre du message a bien été affichée sur la ligne service et donc que l'exécution a bien été arrêtée en #981C, mais qu'ensuite la machine s'est plantée. Comme lors du premier test, un seul octet BRK implanté, mais le résultat est bien différent !

## TEST 3 :

Remplaçons donc également l'octet qui suit le BRK par un NOP, comme nous le conseille Yann L. Soit :

```
981C C8           (INY)             ->    981C 00           (BRK)
981D D0 F5        (BNE#9814)        ->    981D EA F5        (NOP+ex#F5)
```

POKE# 981C,#00 puis POKE#981D,#EA et enfin CALL#9812. La figure 4 ne montre guère de différence avec la précédente. Certes la lettre "S" s'est bien affichée, mais la machine est toujours bloquée. En effet, le processeur a sauté l'octet qui suit le BRK et le programme a été relancé en #981E sur l'octet #F5, qui ne correspond à aucune commande langage machine. Il faut donc neutraliser ce #F5.

```
S                                                   CAPS
ORIC EXTENDED BASIC V1.1fr
1983 TANGERINE

 37631 BYTES FREE

Ready
CLOAD"SALUTBIN"

Ready
POKE#981C,#00

Ready
CALL#9812
```

Fig 3

```
S                                                   CAPS
ORIC EXTENDED BASIC V1.1fr
1983 TANGERINE

 37631 BYTES FREE

Ready
CLOAD"SALUTBIN"

Ready
POKE#981C,#00

Ready
POKE#981D,#EA

Ready
CALL#9812
```

Fig 4

## TEST 4 :

Voyons ce qui se passe si on remplace l'octet #C8 (INY) en #981C par un BRK et les 2 suivants par des NOPs. Soit :

```
981C C8          (INY)          ->    981C 00          (BRK)
981D D0 F5       (BNE#9814)     ->    981D EA EA       (NOP NOP)
```

POKE# 981C,#00 puis POKE#981D,#EA puis POKE#981E et enfin CALL#9812. La figure 5 montre que la première lettre du message a été affichée sur la ligne service, comme précédemment, mais que cette fois la machine n'est pas plantée. Le processeur a sauté l'octet qui suit le BRK et le programme a été relancé en #981E sur le 2ème NOP. Le RTS a ensuite été exécuté. Retour au prompt du Basic comme le montre le Ready.

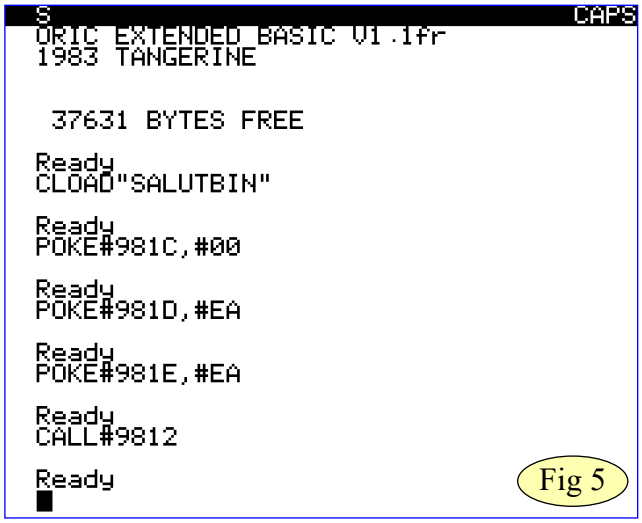

Fig 5

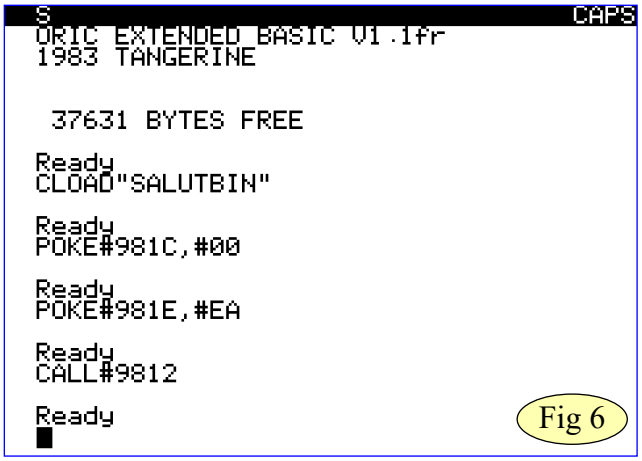

Fig 6

## TEST 5 :

Pour en finir avec le remplacement de l'octet #C8 (INY) par un BRK, essayons le test de paresseux. Pourquoi toucher à l'octet qui suit le BRK, puisque de toute façon il est sauté par le processeur. Ne remplaçons que l'octet #F5 qui cause le plantage. Soit :

```
981C C8          (INY)          ->    981C 00          (BRK)
981D D0 F5       (BNE#9814)     ->    981D D0 EA       (exD0+NOP)
```

POKE# 981C,#00 puis POKE#981E,#EA et enfin CALL#9812. Que l'octet #D0 ait été touché ou pas, la figure 6 ne montre aucune différence . Le processeur saute par-dessus sans s'en occuper…

## TEST 6 :

Remontons encore dans le programme et implantons notre BRK sur le STA BB82,Y. Soit :

```
9819 99 82 BB    (STA#BB82,Y)   ->    9819 00 82 BB    (BRK+ex#82&#BB)
```

Le résultat est couru d'avance. Mais bon, autant faire les choses à fond. POKE# 9819,#00 puis CALL#9812. La figure 7 montre que pour une fois la lettre "S" ne s'est pas affichée sur la ligne service et c'est normal puisque l'instruction STA BB82,Y n'a pas été exécutée. La machine est bloquée. Le contraire aurait été surprenant.

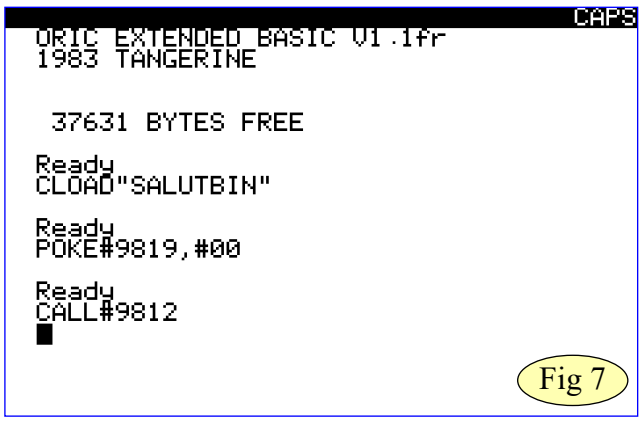

Fig 7

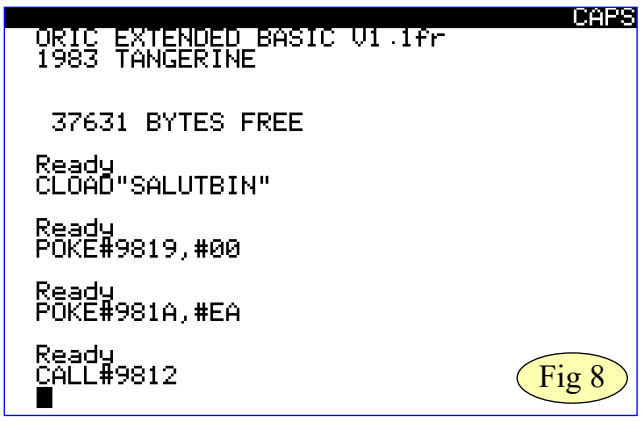

Fig 8

## TEST 7 :

Poursuivons en remplaçant l'octet suivant le BRK par un NOP. Soit :

```
9819 99 82 BB    (STA#BB82,Y)   ->    9819 00 EA BB    (BRK+NOP+ex#BB)
```

Normalement ça ne devrait rien changer au plantage, mais on teste systématiquement, non? POKE# 9819,#00 puis POKE#981A,#EA et enfin CALL#9812. La figure 8 montre comme précédemment et sans surprise, que le seul résultat visible est un beau plantage !

Fig 9

Fig 10

## TEST 8 :

Arrêtons de tourner autour du pot et neutralisons enfin le responsable du plantage. Dans les 2 tests précédents, comme dans le test 2, le processeur a sauté l'octet qui suit le BRK et le programme a été relancé en #981B sur l'octet #BB, qui ne correspond à aucune commande langage machine. Il faut évidemment neutraliser ce #BB.

```
9819 99 82 BB    (STA#BB82,Y)     ->    9819 00 EA EA    (BRK+NOP+NOP)
```

POKE# 9819,#00 puis POKE#981A,#EA puis POKE#981B,#EA et enfin CALL#9812. La figure 9 ne montre pas grand-chose de plus, puisque de toute façon le caractère "S" ne peut pas être affiché, sauf que cette fois-ci la machine ne s'est pas plantée. C'était bien l'octet #BB le responsable, CQFD (oui, enfin, pas de quoi pavoiser, c'était couru) !

## TEST 9 :

Le test du paresseux : Comme pour le test 5, pourquoi toucher à l'octet qui suit le BRK, puisque de toute façon il est sauté par le processeur. Ne remplaçons que l'octet #BB qui cause le plantage. Soit :

```
9819 99 82 BB    (STA#BB82,Y)     ->    9819 00 82 EA    (BRK+ex#82+NOP)
```

POKE# 9819,#00 puis POKE#981B,#EA et enfin CALL#9812. La figure 10 ne montre aucune différence avec la précédente. Le curseur au Ready clignote toujours vaillamment. Que l'octet #82 ait été touché ou pas, le processeur saute allégrement par-dessus sans s'en soucier !

**Un BRK pour quoi faire ?**

M'enfin, à quoi bon mettre un BRK, puisque l'Oric le saute (avec plus ou moins de succès) et reprend (ou tente de reprendre) le programme comme si de rien n'était ?

Si l'on agit en toute connaissance de cause, en respectant les trois règles édictées au début, il n'y a rien d'aléatoire à placer un BRK. De ce point de vue, nous devrions être à l'aise après tous les tests exhaustifs que nous avons effectués. On peut éventuellement ajouter un RTS pour forcer le retour au Basic si ce programme a été appelé par un CALL.

Cela permet d'arrêter un programme en langage machine récalcitrant et d'examiner ce qui cloche, au moins avec quelques commandes PRINT PEEK, avant d'envisager une arme plus lourde, telle qu'un moniteur dont les adresses d'implantation doivent être compatibles avec le programme à déboguer.

Je vous conseille le mini-moniteur d'André Chénière qui est auto-relogeable 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).

**Alors, c'est tout ?**

Certes non ! Les possibilités du BRK sont beaucoup plus vastes que nous ne le pensons généralement. En effet, il serait très intéressant de profiter des deux passages en Ram, lors du processus de gestion du BRK, pour dérouter celui-ci et lui faire faire autre chose, plus adapté à nos besoins.

Ce sera le but du prochain article "Trucs et Astuces Programmation (4)".

A bientôt donc pour de nouvelles aventures !