

# Euphoric : Le Debugger

par André Chéramy, révisé et complété par Fabrice Francès

Vous ne le savez peut-être pas, mais un Moniteur est incorporé à Euphoric. Fabrice m'a indiqué : «J'ai dû parler des fonctions du debugger dans le fichier historique.txt, mais c'est vrai que je n'ai pas trop fait de publicité, pas le temps... :-) Je me sers constamment du debugger, c'est le genre de truc que j'aurais dû faire en premier, j'aurais gagné du temps par la suite...».

Et en effet, il n'en fait pas mention dans le mode d'emploi 'Manuel.htm'. Toutefois, la touche F11 est belle et bien signalée dans le menu F1.

Voici donc les lignes que Fabrice consacre au 'debugger' dans son 'Historique.txt' : «Les débuts d'un débogueur, pas encore stable... il n'est pas fini, il y a des bugs, alors vous êtes prévenus... la suite viendra avec la version suivante (oui, oui, Mike, il y aura le support pour les symboles utilisateurs ;-). Pour l'instant, vous pouvez entrer/sortir du mode débogueur avec la touche F11, faire du pas-à-pas (F2), ou du pas-à-pas sans rentrer dans les routines (F3), basculer vers/depuis l'écran Oric (F4) (pas très fiable...), lancer l'exécution (F5), stopper l'exécution (F12), positionner une adresse de point d'arrêt avec B xxxx (une seule pour l'instant), visualiser des données mémoire (D xxxx), scroller la fenêtre page zéro (flèches haut/bas) ou la fenêtre données mémoire (PgUp/PgDn)...».

A ces informations, Fabrice avait ajouté à mon intention : «Pour surveiller un boot, il y a aussi une nouvelle option -v pour la ligne de commande. F5 lance l'exécution jusqu'au point d'arrêt défini par B adr. F6, F7, F8 font les resets habituels.» Et plus récemment : «Il y a juste un petit problème que je n'ai pas élucidé, je pense que les passages dans le Bios y sont pour quelque chose: l'utilisation répétée de la touche F4 pour afficher l'écran de l'Oric provoque souvent des blocages, dans la pratique j'évite de m'en servir. Il faudra que je regarde de plus près quand j'aurais un moment, parce que les entrées et sorties dans le debugger avec F11 ne posent pas le même problème... Il y a une fonctionnalité que tu auras peut-être plus de mal à tester, mais elle est très pratique pour les développeurs, c'est le mode symbolique: si tu

fournis un fichier de symboles (couples symboles/valeurs), les désassemblages utilisent ces symboles et tu peux utiliser un symbole à la place des adresses pour les commandes D, U, B, etc.»

## MODE D'EMPLOI

J'utilise depuis longtemps ce debugger, mais un peu au pif. Comme je le trouve pratique, je l'ai récemment exploré de façon plus systématique afin de pouvoir vous en parler et vous inciter à l'utiliser vous aussi. Lancez Euphoric et entrez dans le debugger en pressant la touche F11. Voici ce que vous voyez (avec en rouge la dernière valeur qui a été modifiée):

1) Le cadre '6502' vous informe sur l'état des registres du microprocesseur (accumulateur A, index X, index Y, compteur de programme PC, indicateur d'état P et pointeur de pile S).

Dans l'exemple PC=C5E8, c'est l'adresse du code à exécuter (ici la routine en Rom 'Prendre un caractère au clavier'). Vous pouvez d'ailleurs voir ce code dans la fenêtre 'Code' en haut à droite.

Pour l'indicateur d'états P on peut lire '-VIB—Z-', soit 8 caractères, correspondant aux 8 drapeaux (à zéro si représentés par un tiret et à un si représentés par une lettre) : N (Négatif), V (oVerflow = débordement), I (non significatif), B (Break = interruption), D (Décimal), I (Inhibition d'interruption), Z (Zéro) et C (Carry = retenue).

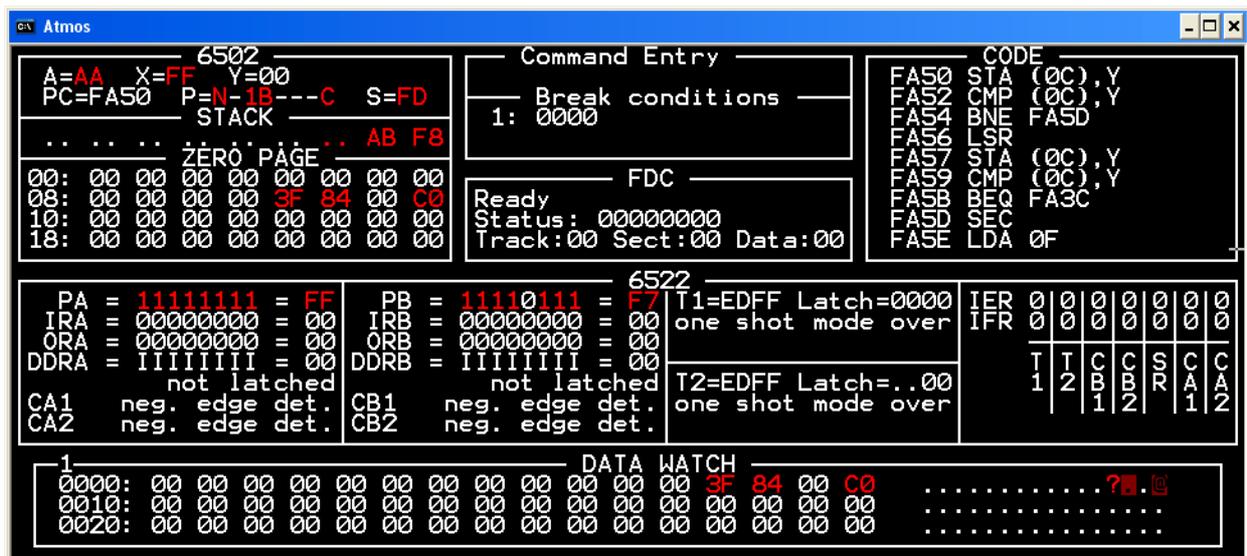
2) Le cadre 'Stack' montre l'état de la pile au pointeur S.

3) Le cadre 'Zéro Page' affiche la page zéro, que l'on peut faire défiler dans la fenêtre avec les flèches haut & bas.

4) Le cadre 'Commande Entry' vous invite à taper une commande. Les valeurs doivent être fournies en hexadécimal sans # ou \$ par-devant. Les lettres minuscules sont acceptées. J'ai repéré les syntaxes suivantes:

a) B xxxx fixe un Break à l'adresse indiquée. Cette adresse est rappelée dans le cadre 'Break Conditions'.

b) D xxxx, dump la mémoire à partir de l'adresse indiquée. Le dump s'affiche dans le cadre 'Data



Watch' et peut défiler avec les touches 'Page Up' et 'Page Down'.

Si cette adresse est en Rom (C000 à FFFF), les adresses sont curieusement affichées entre 0000 et 3FFF. Je suppose qu'il est possible de dumper la Ram Overlay, lorsque celle-ci est validée.

c) N, V, B, D, I, Z et C force l'état du drapeau correspondant (la touche bascule entre zéro et un). Le résultat est immédiatement visible dans le cadre '6502'.

d) A, X ou Y plus un octet, force la valeur du registre correspondant. Particulièrement pratique pour tester une routine. La nouvelle valeur s'affiche dans le cadre '6502'.

e) PC plus deux octets d'adresse, fixe un nouveau point d'exécution. Cette adresse est visible dans le cadre '6502' et le listing correspondant dans le cadre 'Code'.

f) S plus un octet, fixe une nouvelle valeur au pointeur de pile. Cette valeur est visible dans le cadre '6502' et la zone correspondante de la pile dans le cadre 'Stack'.

g) U plus deux octets d'adresse, désassemble le code présent à l'adresse indiquée. Le résultat est immédiatement visible dans le cadre 'Code'.

Notez que le point d'exécution n'est pas affecté. Si après avoir lu le code désassemblé dans le cadre 'Code', vous pressez F2, l'exécution se fera non à l'adresse indiquée par U, mais à celle indiquée par PC dans le cadre '6502'.

h) E xxxx zz, Ecrit l'octet zz à l'adresse xxxx. Le résultat apparaît dans le cadre 'Data Watch', correctement positionné par une commande D appropriée.

5) Nous avons déjà vu que le cadre 'Break Conditions' affiche l'adresse du seul break disponible.

6) Le cadre 'FDC' (Floppy Disk Controlleur) nous informe sur l'état du lecteur de disquette, notamment sur la piste, le secteur et le registre de données.

7) Nous avons déjà vu que le cadre 'Code' affiche le listing au point d'exécution PC ou à l'adresse de désassemblage fixée par U.

8) Le gros cadre '6522' fournit toutes les informations possibles et imaginables sur l'état du VIA.

9) Et enfin, nous avons déjà parlé du cadre 'Data Watch' qui affiche le dump et peut être scrollé avec les flèches 'Page Up' et 'Page Down'.

## LES TOUCHES DE FONCTION.

F1 est inopérante.

F2 exécution pas à pas du code à l'adresse PC.

F3 idem, mais ne trace pas les sous-programmes et passe directement à l'instruction qui suit les JSR (attention aux routines qui ne reviennent pas, elles sont nombreuses dans les différents DOS de l'Oric. Idem pour les BRK du Telestrat).

F4 Bascule entre l'écran d'Euphoric et celui du debugger. Cette fonction est très pratique pour suivre l'effet des tests effectués sous debugger et je l'ai souvent utilisée dans une ancienne configuration. Mais actuellement avec Euphoric 1.0 sous Windows 98, ça plante sévère. On peut contourner avec F11, mais j'espère que Fabrice va revoir la question, car F11 laisse repartir l'exécution.

F5 Lancer l'exécution à partir de l'adresse PC. On arrête avec F12.

F6 Hard Reset habituel

F7 Soft Reset habituel

F8 Plante Euphoric

F9 Save State, marche bien, comme d'habitude.

F10 Quitte tout : debugger et Euphoric.

F11 Quitte le debugger et retourne à l'écran d'Euphoric.

F12 Arrête l'exécution lancée par F5

## RECOPIE DE L'ÉCRAN DU DEBUGGER

Il faut passer en mode fenêtre avec Alt/Enter, faire une copie de la fenêtre avec Alt/PrntScr et la copier dans un logiciel de retouche tel que Photoshop, Paint Shop Pro etc.

## LE MODE SYMBOLIQUE

Ce mode avait été introduit à la demande de Mickaël, pour les développeurs principalement, mais il peut aussi servir à ceux qui souhaitent mettre un nom sur des adresses en Rom ou dans Sedoric, etc.

Pour les développeurs, la plupart des assembleurs sont capables de générer une « cross-reference table » ou table des symboles. Même si l'on développe en C, le code intermédiaire est de l'assembleur, et donc la dernière passe d'assemblage permet d'obtenir les symboles utilisés dans le programme C.

Au démarrage d'Euphoric, le debugger cherche donc un fichier nommé symbols.txt contenant des

suites de symboles précédés de leur valeurs (hexadécimales), par exemple :

```
0035  BUFFER
0735  _printf
07EA  _main
0812  _message
```

Les valeurs et les symboles peuvent être séparés par n'importe quel caractère blanc (espaces, tabulations, sauts de lignes), certains assembleurs tels que le Frankenstein (qui est toujours utilisé dans certaines configurations du compilateur lcc) mettent plusieurs couples de valeurs/symboles par ligne, c'est parfaitement accepté.

Une fois chargé, ce fichier sert à la traduction des symboles vers les adresses correspondantes lors des entrées utilisateurs (par exemple « D\_message », « B\_main » ou « U\_printf »), et à la traduction inverse (adresses remplacées par des symboles) dans les désassemblages affichés dans le cadre de code.

## LE MODE ASSEMBLEUR

Oups, jamais content ! Ce mode ne semble pas encore implémenté. Mais puisqu'il y a un désassembleur, l'assembleur ne devrait pas être bien loin (même rustique, en mode immédiat pour corriger deux ou trois lignes). Je ne voudrais pas être à la place de Fabrice, avec tous ces Oriciens pendus à ses basques, qui aboient à qui mieux-mieux pour en avoir toujours plus...

*[NDLR : réponse de Fabrice : « Au moins, ça dénote un intérêt pour le produit ☺... Bon, et une commande Code, une... »].*

