EUPHORIC ET FTDOS JASMIN

Par Roger Barbier, Club Europe Oric

Grâce à l'émulateur EUPHORIC*, il est désormais possible à chacun d'exécuter sur PC des logiciels écrits soit sous SEDORIC pour le lecteur de disquette Microdisc, soit sous FTDOS pour le lecteur Jasmin. Ce dernier étant bien moins répandu, les utilisateurs habitués au SEDORIC ont parfois quelques difficultés avec la syntaxe et les particularités du FTDOS. Il m'a donc semblé utile de mettre à la portée de tous le chapitre II du manuel d'utilisation de ce dernier.

Le lecteur Jasmin 2, double tête, est prévu pour utiliser des disquettes 3", double face, double densité.

Le Dos d'origine FTDOS, est dans l'ensemble, moins souple et moins rapide que le Dos SEDORIC et n'offre pas les extensions Basic de ce dernier. Par contre, il permet de disposer d'instructions spécifiques pour l'écriture et la lecture de matrices de nombres ou tableaux de chaînes Basic. Ceci pose quelques problèmes pour la conversion des programmes de l'un vers l'autre système.

Pour ceux qui ne veulent que lancer l'exécution d'un programme sous FTDOS Jasmin, voici quelques informations utiles.

-Le répertoire contenant EUPHORIC.EXE doit contenir une image disque incluant le FTDOS ou l'une de ses versions dérivées (Je conseille celle appelée NEWDOS, qui évite l'interminable présentation du logo de la Société TRAN).

-Pour le lancement d'EUPHORIC, un fichier batch est bien utile. Il peut-être directement exécuté en fenêtre sous Windows avec toutes ses facilités, icône, etc..

Exemple de fichier batch à installer dans le même répertoire:

```
euphoric -j jasmin.dsk
```

-Après lancement d'Euphoric, presser la touche F6 pour "booter" le FTDOS.

-Il vaut mieux installer sur le lecteur 1 (appelé A sous Sedoric) la disquette contenant le logiciel à lancer car il n'est pas possible de changer le lecteur par défaut, ce qui amène

quelques problèmes avec certains logiciels. Utiliser la touche de configuration F1.

-Dans ce qui suit N=Numéro de lecteur Ext=Extension
Les " en fin d'instruction sont facultatifs.

-Lire le catalogue !CAT" " pour le lecteur 1 (par défaut)
ou bien !CAT"N." (ne pas oublier le point après le N°)

-Chargement sans lancer !LOAD"N.nomfich.Ext" N facultatif si N=1
Ext facultatif si = .BAS

-Charger et exécuter ! " N.nomfich.Ext" idem

Alors, maintenant, plus d'excuses pour ne pas essayer les nombreux logiciels écrits pour Jasmin.

ORIC, SEDORIC, FTDOS, MICRODISC, JASMIN sont des marques déposées.

* Grand merci à Fabrice FRANCÈS

II - UTILISATION DU FTDOS

(Chapitre II et annexes de la notice JASMIN 2, avec quelques corrections)

SOMMAIRE

GÉNÉRALITÉS

D'entrée nous allons vous donner les quelques règles à respecter pour la syntaxe ainsi que les conventions prises pour décrire chaque commande.

1- Chaque commande T.DOS commence impérativement par le caractère !
2- Les noms de fichiers commencent obligatoirement par un caractère alphabétique "A à Z" et ne doivent pas comporter d'espace en leur milieu.
!SAVE "1.TURLUTU" message correct
!SAVE "1.4TUR LU" message incorrect
3- De même les commandes syntaxiquement groupées ne doivent pas comporter d'espace.
!SAVE "1.TITI.BAS" message correct
!SAVE "1.TITI. BAS" message incorrect.
Une commande syntaxiquement groupée est une commande dont les paramètres sont séparés par des points.

Règle de description syntaxique
Les éléments situés entre parenthèses sont optionnels et peuvent prendre une valeur par défaut qui est toujours clairement indiquée.

Règle sur les noms de fichiers
T.DOS accepte des fichiers ayant le même nom à condition qu'ils n'aient pas le même type ou qu'ils ne résident pas sur la même unité de disque.
1.PAPA.BIN et PAPA.DAT sont deux fichiers différents.
Il en est de même pour
1.PAPA.BIN et 2.PAPA.BIN

OPÉRATIONS SUR LES DISQUETTES

-1 Système mono disquette et multidisquettes

Montez maintenant la disquette T.DOS dans le lecteur, face A vers le haut et appuyez sur le bouton "BOOT".
Le système T.DOS est chargé. La disquette que vous avez dans le lecteur est la disquette MAÎTRE (MASTER>. Elle a pour nom de volume FT.DOS et quelques utilitaires (pour vérifier, frapper la commande !"CAT")
A partir de là, vous pouvez commencer à travailler si vous le désirez (c'est à dire créer des fichiers et les utiliser)
.Vous pouvez d'abord opérer comme suit
Formatez, Initialisez, et donnez un nom à une autre disquette.

Voici les commandes générales.
Dans le cas d'un monolecteur. le n° DISQ est bien évidemment égal à 1.

-2 Formatage. Commande !"FORMAT"

*Syntaxe "!FORMAT"
C'est un utilitaire Basic. C'est un programme conversationnel, suivez ce qu'il vous demande, et n'oubliez pas de changer la disquette.
* Action : Formatage de l'unité de disque et ensuite le programme INITsi vous le désirez.
Remarque : lorsque vous tapez !"FORMAT", la disquette Maitre doit être dans le lecteur.

-3 Initialisation. Commande INIT

*Syntaxe !INIT "(N0 DISQ.) NOMVOLUME"
Paramètres N° DISQ. = Numéro de l'unité disque (de 1 à 4)
NOMVOLUME Nom de volume (de la disquette) que vous voulez initialiser (1 à 8 caractères alpha-numériques). Si vous mettez plus de 8 caractères, le nom est tronqué à droite à 8 caractères.
* Action :Cette commande initialise le catalogue.
ATTENTION : Dans le cas d'une disquette contenant des fichiers, il y a perte de ces fichiers.
Toutes vos disquettes formatées et initialisées peuvent être 'Bootstrappées" c'est à dire contenir T.DOS.

-4 Copie du T.DOS. Commande !MASTER

Syntaxe: ! MASTER "(N° DISQ.) NOMDOS"
* Paramètres : N° DISQ = Numéro de l'unité disque (1 à 4)
NOMDOS : Nom que l'on donne au système d'exploitation disque 1 à 8 caractères alphanumériques. Si vous mettez plus de 8 caractères, le nom est tronqué à 8 caractères.
* Action : Cette commande permet de rendre une disquette MAÎTRESSE, en recopiant

dessus le système T.DOS. Cette disquette devient alors "Bootstrappable".
ATTENTION: La disquette d'origine livrée avec l'appareil contient deux systèmes d'exploitation, l'un pour ORIC l'autre pour ATMOS.
Bien sûr cela prend deux fois plus de place sur la disquette.
Lors BOOT, le système sait reconnaître si c'est l'ORIC 1 ou l'ATMOS qui est branché et charge le "DOS" correspondant.

Le DOS de l'ATMOS occupe la première place sur la disquette et celui de l'ORIC1 la seconde.
La commande !MASTER sauvegarde le DOS de l'ORIC1 ou de l'ATMOS à la première place, selon que l'ORIC1 ou ATMOS est branché.Ainsi lorsque vous utilisez !MASTER sur la disquette d'origine un seul DOS reste utilisable.
Si vous voulez créer une disquette MAÎTRE directement à partir d'une disquette nouvellement formatées, il vous suffit de faire !INIT suivi de !MASTER. Un seul DOS est alors sauvegardé.
Quand vous créez une disquette MAÎTRE à partir d'une disquette dupliquée de l'originale, par un "!BKP", utilisez l'utilitaire !"AUTODOS".
Cet utilitaire réalise dans l'ordre !MASTER puis efface le DOS de la deuxième place.
IMPORTANT : Cet utilitaire ne peut être utilisé qu'une fois. Car une seconde fois il effacerait les fichiers logés dans la seconde place rendue disponible.

-5 Nomination. Commande !DNAME
* Syntaxe : !DNAME "(N0DISQ.) NOMVOLUME"
* Paramètres : N°DISQ = Numéro de l'unité disque (i à 4). Valeur par défaut 1
NOMVOLUME Nom que vous voulez donner à votre disquette. 1 à 8 caractères alphanumériques. Si vous mettez plus de 8 caractères, le nom est tronqué à droite à 8 caractères. Le premier caractère doit être alphabétique (A à Z).

* Action Cette commande permet de donner un nom à une disquette ou de renommer une disquette (changement de nom).
Exemple : !DNAME "ESCLAVE"

-6 Liste du catalogue sur écran !CAT
* Syntaxe :
!CAT "(N°DISQ.)"
!CAT " ". pour le disque 1
!CAT "2." pour le disque 2
* Fonction : liste du catalogue complet des fichiers contenus sur la disquette.
Vous verrez par exemple
VOLUME : NOMVOLUME

LTDOS 1-2	.SYS S	62 SECTORS
U FORMAT	.BAS S	13 SECTORS
U FICHES	.DAT D	22 SECTORS

355 SECTORS FREE

La première lettre L ou U indique si le fichier est respectivement verrouillé ou non par la fonction !LOCK. Ensuite suivent le nom du fichier et son type ou extension.
Après l'extension, une lettre isolée S ou D indique si le fichier est séquentiel ou à accès direct.
La taille du fichier est exprimée en nombre de secteurs. Chaque secteur contient 256 octets.
A la fin vous avez l'espace disponible du disque.
Vous pouvez interrompre momentanément le listage en appuyant sur la barre "espace". Ensuite à chaque pression, vous imprimez une ligne. Pour arrêter définitivement cette instruction, tapez "RETURN".

-7 Liste du catalogue sur imprimante !LCAT
Fonctionne comme !CAT mais liste sur imprimante.

-8 Copie d'une disquette entière !"BKP"
* Syntaxe !"BKP"
* Action Copie intégrale d'une disquette sur une autre disquette.
(C'est un utilitaire conversationnel qui se trouve sur la disquette "MAÎTRE" et qui se charge en mémoire à chaque appel>.
Répondez aux questions que vous pose le programme.
La disquette "Emettrice" est celle que l'on peut dupliquer.
La disquette "Réceptrice" est celle qui va devenir le duplicata.
Lorsque vous avez deux lecteurs, le transfert se fait sans manipulation. Si vous n'avez qu'un seul lecteur, vous pouvez suivre scrupuleusement les indications du programme. La taille mémoire de l'ordinateur étant plus petite que celle d'une disquette, l'opération se fera en plusieurs fois. Vous aurez à permuter plusieurs fois les disquettes.
Veillez à ne pas vous tromper entre la disquette "émettrice" et "réceptrice".

-9 Montage d'un volume !MOUNT
Vous êtes en possession de plusieurs unités de lecteurs (maximum 4), d'une disquette supportant le système T.DOS et d'autres disquettes neuves. Vous effectuez tous vos branchements, la disquette T.DOS dans le lecteur N° 1, les disquettes formatées et initialisées dans les autres unités. "Bootstrappez" la disquette T.DOS pour charger le système en mémoire.
Si votre système n'a pas été informé pour être utilisé en multidisquette, il faut le prévenir par l'instruction !MOUNT.
Comme pour le système mono-lecteur, vous pouvez recopier le T.DOS sur d'autres disquettes. La procédure est identique sauf que vous n'avez pas à effectuer de changement de disquette (attention au paramètre N° DISQ.).
Lorsque vous avez configuré votre système, c'est à dire "Monté" tous les lecteurs, refaites un !MASTER, ainsi la configuration définie reste valable chaque fois que vous refaites

un "BOOT".
* Syntaxe !MOUNT " N° DISQ."
* Paramètre N° DISQ. = Numéro de l'unité disque (1 à 4)
* Action Cette commande a pour but de faire connaitre au T.DOS en mémoire, qu'il existe des unités de lecture connectées ou "montées" autres que le n° 1.
Cette fonction vous permet de configurer votre système.

-10 Démontage d'un volume !DEMOUNT
Cette commande indique au T.DOS en mémoire que la configuration est modifiée et qu'il y a démontage d'une unité de lecture.

-11 Recopie de fichiers. Commande !COPY
Un système T.DOS serait incomplet s'il ne permettait pas la duplication de fichier. Ainsi T.DOS a une commande pour cet usage.
Syntaxe : !COPY "(N° DISQ1.)NOMA(.TYP1) = (N°DISQ2.)NOMB(.TYP2)"
* Paramètres
N°DISQ1 Numéro de l'unité disque émetteur. Valeur par défaut 1
NOMA Nom de fichier émetteur ou à recopier. 1 à 8 caractères alphanumériques. Si vous mettez plus de 8 caractères, le nom est tronqué à droite à 8 caractères. Si le nom existe, le fichier est écrasé, sinon il est créé.
TYP1:
BIN
BAS
TXT
ARY
DAT
CMD
Valeur de TYP1 par défaut : BAS
N°DISQ2 Numéro de l'unité réceptrice (de 1 à 4). Valeur par défaut 1.
NOMB Nom du fichier récepteur, 1 à 8 caractères alphanumériques. Si vous mettez plus de 8 caractères, le nom est tronqué à droite à 8 caractères.
TYP2 :
BIN
BAS
TXT
ARY
DAT
CMD
Valeur de TYP2 par défaut : TYP1
Remarque
Lors de la recherche du fichier à recopier, si le paramètre TYP1 est présent, les fichiers ayant pour nom NOMA mais dont le type ne correspond pas à TYP1, ne seront pas considérés comme le fichier à recopier.

Exemple
!C0PY "TOTO.BIN = 2.TATA" copie de l'unité 1 à 2
!COPY "TOTO.BIN = TITI.CMD" copie de la même unité.
C'est un utilitaire conversationnel qui vous permet de recopier un fichier binaire sur une autre disquette, dans le cas où vous n'auriez qu'un seul lecteur

TRAITEMENT SUR TOUT FICHIER

-1 Sauvegarde !SAVE

De la même manière que vous pouvez effectuer des sauvegardes sur cassettes, T.DOS donne cette possibilité. La fonctionnalité est identique.
*Syntaxe : ! SAVE "(N°DISQ.)(NOMFICH(.TYP)(,AD,AF)
*Paramètres :
N°DISQ:: Numéro de l'unité disque de 1 à 4. Valeur par défaut 1
NOMFICH : Nom du fichier sur 8 caractères alphanumériques obtenus par troncature à droite. Le premier caractère doit être obligatoirement alphabétique (A à Z).
TYP: BAS indique la sauvegarde d'un programme écrit en Basic.
BIN indique la sauvegarde en binaire d'une zone de mémoire
délimitée par les adresses AD et AF. Valeur de TYP par défaut : BAS.
Si TYP:
BIN }
TXT: }
DAT: } AD, AF Adresse de début, adresse Fin.
CMD: }
SCR }

Les valeurs peuvent être en décimal ou en hexadécimal précédées du caractère "E", AD et AF peuvent être des variables BASIC.
Exemple:

```
10 AD = #6000
20 AF = #8000
30 A$ = "TOT.BIN,AD,AF"
40 !SAVE A$  (ou SAVE "TOT.BIN, #6000, #8000")
```

Action : Sauvegarde par la disquette portée par l'unité N0DISQ du programme BASIC en mémoire si TYP = BAS ou de la zone nécessaire définie par AD et AF incluses si TYP = BIN,TXT,DAT,SCR,CMD. Le nom du fichier ainsi créé est NOMFICH.
Si un fichier de même nom et de même type existe déjà dans le catalogue, il y a écrasement du précédent fichier par le nouveau, sauf si le fichier est protégé par !LOCK.

-2 Chargement !LOAD

* Syntaxe !LOAD "(N°DISQ.)NOMFICH(.TYP>(,AD)"
* Paramètres :
N°DISQ Numéro de l'unité disque de 1 à 4. Valeur par défaut :1.
NOMFICH : Nom du fichier à charger sur 8 caractères obtenus par troncature à droite. Le premier caractère doit être obligatoirement alphabétique (de A à Z)
TYP .BIN Indique le chargement d'une zone mémoire. Si le paramètre AD est présent, le chargement s'effectue à partir de cette adresse, dans le cas contraire l'adresse du début

est celle donnée lors de la commande !SAVE.

SCR: }
TXT: }
DAT: } comme BIN
CMD: }

Valeur de TYP par défaut : BAS
Si TYP = BIN : AD Adresse à partir de laquelle doit être chargée la zone mémoire contenue dans le fichier.
La valeur décimale ou héxadécimale AD peut être une variable BASIC.
* Action : Chargement à partir de la disquette portée par l'unité N0DISQ. du programme basic si TYP = BAS, ou de la zone mémoire si
TYP = BIN,TXT,DAT ou CMD.
Dans ces derniers cas, si le paramètre AD est présent le chargement s'effectue à partir de cette adresse. (possibilité de décalage de zones).
Après exécution de cette instruction l'adresse de début se trouve en '#48D et #48E et l'adresse de fin en
#48F et #490.

-3 Chargement et exécution
* Syntaxe !"(N°DISQ.)NOMFICH(.TYP)".
* Paramètres
N°DISQ. = Numéro de l'unité disque de 1 à 4. Valeur par défaut 1.
NOMFICH Nom de fichier à charger sur 8 caractères obtenus par troncature à droite. Le premier caractère doit être obligatoirement alphabétique (de A à Z).
TYP BAS Indique le chargement d'un programme écrit en basic.
BIN ou CMD Indique le chargement d'un programme écrit en langage machine, le chargement s'effectue à partir de l'adresse du début donnée lors de la commande ! SAVE.
Valeur de TYP par défaut : BAS.
* Action Chargement et exécution à partir de la disquette portée par l'unité N°DISQ du programme Basic, si TYP = BAS, ou du programme en code machine si TYP = BIN ou CMD.
Cette instruction permet aussi des lancements automatiques d'un programme à partir d'un autre programme.
Exemple
!"FORMAT" ou ! "TKD" ou ! "TOT.CMD"

-4 Transfert d'un fichier de cassette à disquette sans déprotéger !"TKD"
* Syntaxe !"TKD"
* Action Transférer un fichier d'une cassette sur la disquette. C'est un utilitaire conversationnel qui se charge en mémoire à chaque appel. Suivez les indications du programme.

Si vous ne connaissez pas le nom du programme sur cassette, mettez " "
Dans ce cas, le premier fichier rencontré est transféré
Lorsque le programme vous demande le nom de fichier sur disquette, vous mettez le nom que vous voulez sur 8 caractères maximum, commençant par un caractère alphabétique. Si le nom sur cassette commence par un caractère autre qu'alphabétique, ou avec un "blanc" de séparation, vous devez changer le nom sur disquette, sinon vous ne pourrez plus y accéder.
Cet utilitaire est créé dans le but de faciliter la sauvegarde de vos programmes ou fichiers de cassette à disquette et non de pirater les logiciels protégés.
Il lit la cassette selon le codage standard d'ORIC. Si votre fichier a été codé d'une autre manière, cet utilitaire ne pourra le lire. Si votre programme a été enregistré en plusieurs tronçons, le transfert se fera tronçon par tronçon. C'est à vous d'assurer ensuite le lien entre ces troncons. Pour arrêter l'opération en cours, faites un reset à chaud par le bouton situé sous de 'ORICi ou ATMOS, faites un RESET général.

-5 Recherche de fichier !SEARCH
* Syntaxe !SEARCH "(N0DISQ.) NOMFICH.TYP"
* Action Rechercher dans le catalogue le fichier de nom NOMFICH.TYP. Le type est obligatoire. En mode "commande", si le fichier existe, le message "EXISTING FILE!" est imprimé, sinon "FILE NOT FOUND". En mode programmé, l'impression des messages interrompt le programme comme si c'était une erreur. Cette instruction peut être utilisée en combinaison avec !ERSET. Dans ce cas, se trouve à l'adresse #489, la valeur 1 si le fichier existe sinon 7.
Exemple:

```
5 REM RECHERCHE DE FICHIER
10 ERSET
20 INPUT "NOM DE FICHIER" ;A$
30 B$ = A$ + ".BAS"
40 !SEARCH B$
50 IF PEEK (#489) = 1 THEN PRINT B$, "EXISTE"
60 B$ = A$ = ".BIN"
70 !SEARCH B$
80 IF PEEK (#489) = 1 THEN PRINT BS, "EXISTE" etc...
```

-6 Changement de nom d'un fichier : !RENAME
* Syntaxe: !RENAME "(N°DISQ.)NOMFICA(.TYPE) = NOMFICB(.TYPE)"
Paramètres :
N0DISQ. = Numéro de l'unité qui supporte le fichier à renommer. Valeur de 1 à 4. Valeur par défaut 1.
NOMFICA Nom du fichier à renommer de 1 à 8 caractères (avec troncature à droite pour les tailles supérieures). Le premier caractère est obligatoirement alphabétique.
TYPE:
ARY

BAS
BIN
TXT
DAT
CMD
Valeur de TYP par défaut : BAS
NOMFICB : Nouveau nom du fichier. De 1 8 caractères alphanumériques troncature à droite pour plus de 8 caractères. Le premier caractère est obligatoirement alphabétique.
* Action Cette commande permet de changer le nom (pas le type) d'un fichier et ce pour tous les fichiers sauf les fichiers systèmes.

-7 Protection d'écriture !LOCK
* Syntaxe: !LOCK "(N°DISQ.)NOMFICH.TYP"
* Paramètres : habituels.
* Action : Interdire l'écriture dans un fichier désigné, de même que son effacement (!DEL) ou son écrasement (!SAVE)
Par sécurité le type ne peut être omis.

-8 Déprotection d'écriture !UNLOCK
* Syntaxe : !UNLOCK "(N°DISQ.)NOMFICH.TYP"
* Paramètres : habituels.
* Action : Enlever l'interdiction d'écriture ou d'effacement dans le fichier désigné.
Fonction inverse de !LOCK
Par sécurité le type ne peut être omis.

-9 Concaténation d'un fichier Basic à la suite d'un autre fichier Basic !MERGE
* Syntaxe : !MERGE "(N°DISQ.)NOMFICH"
* Action : Adjonction du fichier NOMFICH.BAS à la suite du fichier BASIC existant en mémoire, tout en gardant les variables existant en mémoire. Ce qui permet d'effectuer des chainages de programmes en continuant leur exécution.
Le programme appelé doit être en BASIC.
Vous devez veiller à ce que tous ces numéros de lignes soient supérieurs à ceux existant en mémoire centrale.
Nous ne garantissons pas du résultat autrement.
Le type du fichier peut être omis, car il est d'office .BAS.

-10 Destruction de fichier !DEL
*Syntaxe: !DEL "(N0DISQ.)NOMFICH.TYP
* Action : Effacer le fichier NOMFICH.TYP, l'enlever du catalogue et libérer la place qu'il occupait sur la disquette.
Attention pour éviter l'effacement par erreur, le type du fichier est obligatoire.
Exemple : !DEL "TOTO.BAS"

-11 Effacement à partir de la ligne numéro N !CUT N

*Syntaxe : !CUT N

* Paramètre : N : Numéro de ligne

* Action : Efface les lignes de programme de numéro N à la fin.

Cette instruction, utilisée avec ! MERGE, vous permet de faire tourner des programmes qui ne tiendraient pas dans les 42k octets utilisateur. Vous pouvez garder un morceau de programme fixe en mémoire centrale, appelé "Noyau", les autres morceaux seront appelés à tour de rôle par !MERGE et effacés pour libérer la place par ICUT. Cette instruction conserve la valeur des variables en cours.

TRAITEMENT D'ERREUR

-1 Empêche l'interruption en cas d'erreur du DOS !ERSET

* L'excution du programme en cours ne s'interrompt pas à la rencontre d'une erreur d'utilisation du T.DOS, et continue sur la prochaine instruction.
A la place du message d'erreur, le code d'erreur est mis en #489, que vous pouvez consulter par un PEEK (#489) par exemple. Cette instruction est très intéressante lorsqu'elle est utilisée avec !ERR GOTO, voir un exemple au chapitre correspondant.

-2 Autorise l'interruption normale en cas d'erreur d'utilisation du DOS !EROFF

*Syntaxe : !EROFF.
* Action : Autorise l'interruption d'un programme et l'impression du message d'erreur en cas d'erreur d'utilisation du T.DOS. C'est la fonction inverse de !ERSET.

-3 En cas d'erreur d'utilisation du DOS, continue à la ligne N !ERR GOTO NN

*Syntaxe . !ERR GOTO NN
*Action : C'est l'équivalent de "ON ERROR GOTO" de certain BASIC, à utiliser avec !ERSET.
Quand le programme arrive à cette instruction, il regarde s'il n'y avait pas une erreur d'utilisation du T.DOS, si c'est le cas, il va à la ligne NN, sinon il continue à l'instruction suivante.
Exemple

```
10 REM chargement d'un FICHIER
20 !ERSET
30 INPUT "NOM DE FICHIER" A$
40 !LOADA$
50 !ERR GOTO 100
60 PRINT "FICHIER" ;A$; "CHARGE"
70 INPUT "POUR CONTINUER TAPEZ O :" ;0$
80 IF O$ = "O" THEN 30
90 END
100 ER = PEEK (#489)
110 IF ER = 7THEN 200
120 PRINT "RECOMMENCER"
130 GOTO 30
200 PRINT "CE FICHIER N'EXISTE PAS"
210 GOTO 120
```

Remarque
Le code d'erreur 7 correspond à "FILE NOT FOUND". Ce traitement d'erreur ne s'occupe pas des erreurs du BASIC. La table des codes d'erreur est donnée en annexe A4.

FICHIERS DE DONNÉES DE TYPE DAT

-1 Généralités

Maintenant que vous savez préparer vos disquettes, sauvegarder et charger vos programmes, il est temps de passer à un stade supérieur l'utilisation des fichiers autres que les fichiers binaires, basics ou commandes.
Ce sont des fichiers de données, caractérisés par le type .DAT (pour DATA). Ils peuvent être à accès séquentiel ou direct.
L'utilisation de tels fichiers suit toujours le même protocole.

1- Création : Si le fichier n'existe pas, il faut le créer et l'ouvrir par l'instruction !CREATE.

2- Ouverture : Si le fichier existe déjà sur disquette, il faut l'ouvrir. C'est comme si l'on ouvrait un compte, un dossier, un livre etc...

Chaque fois que vous créez et ouvrez ou simplement ouvrez un fichier, vous devez lui donner en même temps un numéro, appel numéro logique du fichier (NLU).
C'est comme si on donnait un numéro de compte ou de dossier. Après, jusqu'à la fermeture du fichier, il n'est connu que sous son numéro logique.
Vous créez par exemple un fichier en lui donnant le nom "ESSAI". Il sera sauvegardé sur disque sous le nom "ESSAI.DAT", mais dans l'ordinateur, il sera ouvert et connu par le numéro logique que vous lui aurez donné.
Exemple :!CREATE "ESSAI,S = 5"
Vous avez créé un fichier ESSAI.
Il est séquentiel, vous l'avez indiqué par S. Et il a le numéro logique 5.
Vous pouvez avoir jusqu'à 16 fichiers ouverts en même temps.
Le numéro logique est un entier de 1 à 25.
Le numéro zéro est interdit. Il a une signification bien particulière pour les instructions !CLOSE, !REWIND, !APND.

3- Lecture et écriture Une fois qu'un fichier est ouvert, vous pouvez écrire dedans et relire ce qui a été écrit.
C'est là que se distinguent deux types de fichiers
- Le fichier à accès séquentiel
- Le fichier à accès direct.

Comme leur nom l'indique, ils sont différents par leur façon d'accéder aux informations sauvegardées sur disquette. Ces deux types de fichiers mémorisent des chaînes de caractères représentés sous leur forme ASCII.
Dans un fichier à accès séquentiel, les informations que vous écrivez sont mises l'une à la suite de l'autre.
Vous relisez l'une derrière l'autre les informations dans l'ordre où vous les avez écrites.
Une bande magnétique est un très bon exemple de fichier séquentiel. Si vous n'avez pas repéré à l'avance où se trouve enregistrée votre information sur la bande, vous êtes obligé

de lire la bande depuis le début, jusqu'au moment où vous trouvez votre information.
Un tel fichier est simple, mais devient très vite peu pratique, lorsque vous avez des informations que vous voulez changer souvent, surtout quand elles ne sont pas situées l'une derrière l'autre.
Le fichier à accès direct aléatoire est plus structuré et va donc vous permettre d'accéder directement à une zone d'informations voulue, sans avoir à lire tout ce qui se trouve avant. Chaque zone d'informations est appelée "enregistrement".
Chaque enregistrement peut être composé de plusieurs rubriques.
Ce fichier est, en fait, une suite d'information (en l'occurrence des mots). Chaque élément d'information est dénommé "Rubrique".
Chaque rubrique commence par un octet qui contient sa longueur en nombre d'octets, suivi par le contenu et fini par un octet "# FF". La longueur maximale d'une rubrique est de 256 octets. La longueur maximale réelle de l'information est alors de 254 octets.
Un fichier séquentiel est donc formé d'une suite de rubriques. Ces rubriques peuvent être de longueurs variables.
Un fichier à accès direct aléatoire est composé d'enregistrement de longueur fixe. Chaque enregistrement est en fait un petit fichier séquentiel qui comporte des rubriques de longueur variable. La longueur maximale d'un enregistrement est 256 octets. Il va de soi que la somme des longueurs des rubriques d'un enregistrement ne doit pas dépasser la longueur de ce dernier.
Pour une variable réelle, le système réserve automatiquement 17 octets et pour une variable entière (%), 8 octets.
Pour lire ou écrire dans un fichier à accès direct, il vous suffit de préciser le numéro logique du fichier concerné, le numéro d'enregistrement dans lequel vous voulez travailler.
Ce type de fichier pratique d'utilisation est indispensable pour des applications où l'on a besoin d'accéder, rapidement et dans un ordre quelconque, aux informations.

4- Fermeture

Quand vous classez une affaire, vous fermez le dossier. Quand vous avez fini de lire un roman, vous fermez le livre.
Pour un fichier, c'est pareil, quand vous n'avez plus besoin d'un fichier, vous devez le fermer. Ce qui a pour effet de sauver certains paramètres du fichier et libérer le numéro logique correspondant.
Attention :Les fichiers à accès séquentiel ou à accès direct ont des paramètres qui sont bien particuliers à la configuration du moment de la disquette, ils ne peuvent donc pas être copiés par fonction ! COPY. Vous devez soit utiliser la fonction !BKP pour recopier tout le disque dans la même configuration, soit recopier par un programme, enregistrement par enregistrement ou rubrique par rubrique.

-2 Protection d'écriture !WL

Syntaxe : !WL NLU
Paramètre NLU = Numéro logique d'unité du fichier déjà ouvert.

1 < ou = NLU < ou = 255 NLU = 0 EST INTERDIT.
*Action : Protège contre toute tentative d'écriture sur le fichier.
Cette protection ne s'étend pas jusqu'à l'effacement ou l'écrasement du fichier par des instructions !DEL, !SAVE etc...
Pour protéger contre ces dernières, il faut utiliser !LOCK.

-3 Déprotection d'écriture !WUL
Syntaxe: !WUL NLU
Paramètre NLU = Numéro logique du fichier déjà ouvert.
1 < ou = NLU < ou = 255 NLU = 0 est interdit.
Action fonction inverse de WL, permet l'écriture sur le fichier NLU

-4 Ouverture de fichier !OPEN
Syntaxe !OPEN "(N°DISQ.)NOMFICH = NLU"
Paramètres
NLU = Numéro logique qui sera affecté au fichier de nom NOMFICH.DAT. c'est une valeur décimale, hexadécimale ou une variable.
1 < ou = NLU < ou = 255
TYPE Le type du fichier sera mis d'office à .DAT, quelque soit le type que vous ayez indiqué.
Action Cette commande ouvre le fichier existant NOMFICH.DAT et lui associe le numéro logique NLU.

-5 Fermeture du fichier !CLOSE
Syntaxe !CLOSE NLU
Paramètre NLU Numéro logique du fichier à fermer, valeur décimale, héxadécimale ou variable;
1 < ou = NLU < ou = 255
si NLU = 0, tous les fichiers ouverts sont concernés.
Action : Fermeture du fichier de numéro logique NLU ou fermeture de tout fichier ouvert si NLU = O. A partir de l'exécution de cette commande, l'utilisateur n'a plus accès aux informations contenues dans le fichier par le numéro logique NLU

FICHIERS A ACCÈS SÉQUENTIEL

-1 Création d'un fichier !CREATE

Syntaxe !CREATE "(N0DISQ.)NOMFICH,FTYPE =NLU"
Paramètres :
NLU = Numéro logique qui sera affecté au fichier. C'est une valeur décimale, hexadécimale ou une variable.
1 < ou = NLU < ou = 255
Type ; Le type du fichier sera mis d'office à .DAT, quelque soit le type que vous ayez indiqué.
FTYPE = S pour fichier à accès séquentiel.
*Action Cette commande crée le fichier NOMFICH en accès séquentiel sur disque et l'ouvre en mémoire centrale en lui associant le numéro logique NLU

-2 Ouverture de fichier !OPEN

Syntaxe !OPEN "(N°DISQ.)NOMFICH=NLU"
Paramètre NLU = Numéro logique qui sera affecté au fichier de nom NOM FIC H . DAT.
C'est une valeur décimale, hexadécimale ou une variable.
1 < ou = NLU < ou = 255
Type : Le type du fichier sera mis d'office à DAT, quel que soit le type
que vous ayez indiqué.
* Action. Cette commande ouvre le fichier existant NOMFICH.DAT et lui associe le numéro logique NLU.

-3 Fermeture du fichier !CLOSE

Syntaxe !CLOSE NLU
Paramètre : NLU = Numéro logique du fichier à fermer. Valeur décimale ou hexadécimale ou variable.
1 < = NLU < = 255
Si NLU = 0, tous les fichiers ouverts sont concernés.
* Action Fermeture du fichier de numéro logique NLU ou ferme tout fichier ouvert si NLU = 0. A partir de l'exécution de cette commande, l'utilisateur n'a plus accès aux informations contenues dans le fichier, par le numéro logique NLU.

-4 Ecriture !WRITE

*Syntaxe : !WRITE NLU' "liste des variables"
ou !WRITE NLU REM "liste des variables" pour version ORIC 1
Paramètres NLU = numéro logique de fichier
1 < = NLU < = 255
Liste des variables : Suite de variables séparées par des ",". Tous les types de variables sont acceptés sauf la variable chaîne en tableau comme A$(5,4).

ATTENTION Lorsqu'une variable chaîne est utilisée dans la liste des variables, elle doit être utilisée ou avoir existé précédemment.
Sinon il faut la définir par exemple par A$ =" " avant d'utiliser cette instruction.
* Action Cette commande permet l'écriture dans un fichier si ce dernier n'est pas verrouillé par !WL.
Une variable flottante occupe 17 octets.
Une variable entière occupe 8 octets.
Une variable chaîne a une longueur variable.
Exemple
A$="" :I=5 :NLU = 120 :B%=400 :C$= "TOTO"
!WRITE NLU l,A,B%,C$.
Chaque variable occupe une rubrique.
Vous avez la possibilité de réécrire sur une rubrique existante, donc de la modifier. Mais prenez garde, une fois qu'une rubrique d'un fichier séquentiel a été créée par une première écriture, sa longueur est fixée. Tout ce que vous allez écrire dessus va garder automatiquement la même longueur. C'est le T.DOS qui s'occupe de cette opération et vous évite d'écraser par mégarde la rubrique suivante.
Résultat : Lorsque votre nouvelle rubrique est trop longue, elle est tronquée par la droite. Si elle est plus courte, la suite est remplie par des "espaces". Chaque fois que vous avez écrit une rubrique, vous êtes positionné pour accéder à la rubrique suivante.

-5 Lecture !TAKE
Syntaxe : !TAKE NLU'"Liste des variables" , ou !TAKE NLU : REM "Liste des variables" pour ORIC 1.

Paramètre : NLU = Numéro logique du fichier que l'on veut lire.
1 = ou < NLU < ou = 255
"Liste des variables" Suite des variables BASIC que l'on veut lire, séparées par des ",".
Tous types de variables sont acceptés sauf la variable chaîne en tableau comme A$(5,4).
ATTENTION Lorsqu'une chaîne est utilisée dans la liste des variables, elle doit être utilisée ou avoir existé auparavant. Sinon il faut la définir par A$ =" " avant d'utiliser cette instruction.
* Action : La commande !TAKE effectue la lecture d'une suite de rubriques suivant la liste des variables. Vous ne pouvez pas lire plus loin que ce que vous avez écrit. Chaque fois que vous avez lu une rubrique, vous êtes positionné pour accéder à la rubrique suivante.

-6 Positionnement en début de fichier !REWIND
Syntaxe : !REWIND NLU
Paramètre : NLU = Numéro logique du fichier à accès séquentiel déjà ouvert.
1= < NLU = < 255
NLU peut être une variable.
Lorsque NLU=0, l'opération s'effectue sur tous les fichiers séquentiels ouverts.

* Action : Positionnement en début du fichier séquentiel NLU. Lorsqu'un fichier vient d'être ouvert, il est positionné sur la rubrique où il était au moment de la fermeture précédente.
C'est une des particularités intéressantes du T.DOS. Ce qui vous permet d'interrompre une session et de la reprendre au point où vous l'avez laissée.
Si vous voulez tout reprendre depuis le début, n'oubliez pas de faire !REWIND 0 après avoir ouvert les fichiers.
Le terme REWIND qui veut dire rebobinage vient du fait qu'avant l'arrivée des disques, les fichiers étaient sur bandes.

-7 Positionnement en fin de fichier !APND

Syntaxe !APND NLU
*Paramètre NLU : Numéro logique du fichier à accès séquentiel déjà ouvert.
1 = < NLU = < 255
NLU peut être une variable.
Lorsque NLU = 0 l'opération s'effectue sur tous les fichiers séquentiels ouverts.
* Action : Positionnement en fin du fichier séquentiel NLU. Ce qui va vous permettre de rajouter d'autres rubriques à la suite.

Exemple

```
10    OPEN "ESSAI = 5"
20    !APND 5
30    A=55:C$ = "BONJOUR"
40    !WRITE 5'A,C$
```

Lorsque vous rajoutez les rubriques à la fin d'un fichier séquentiel. ces rubriques sont écrites pour la première fois, leur longueur est déterminée comme suit
Une variable flottante occupe 17 octets
Une variable entière occupe 8 octets
Une variable chaîne a une longueur variable.

-8 Indication du N° de rubrique courante du fichier et du nombre total de rubriques !WHERE

Syntaxe : !WHERE NLU
* Paramètre : NLU = Numéro logique du fichier ouvert
1 < ou = NLU < ou = 255
NLU = 0 est interdit.
* Action : Mettre à l'adresse #48D, le numéro de rubrique en cours du
fichier NLU et en #48F le nombre total de rubriques du fichier.
Ex : pour obtenir le nombre total de rubriques faites
A = DEEK (#48F)

-9 Incrémentation du pointeur de rubrique !JUMP

Syntaxe : !JUMP NLU,NI

Paramètres :

NLU = Numéro logique du fichier déjà ouvert.

1 < ou = NLU < ou = 255

NLU = 0 est interdit.

NI = Nombre d'incrémentation du pointeur

1< ou = NI< ou = #FFFF

*Action : Sauter NI rubriques et se positionner NI rubriques plus loin. Si vous êtes à la X ième rubrique, !JUMP NLU, NI vous positionne à la X+ NI ème rubrique.

FICHIERS A ACCÈS DIRECT

-1 Création d'un fichier !CREATE

Syntaxe : !CREATE "(N°DISQ.)NOMFICH.FTYPE =NLU,LR,NE"
Paramètres :
NLU = numéro logique qui sera affecté au fichier. C'est une valeur décimale, hexadécimale ou une variable.
1 < ou = NLU < ou = 255.
FTYPE = D pour fichier à accès direct.
TYPE = le type du fichier sera mis d'office à DAT, quel que soit le type que vous ayez indiqué.
LR = Largeur de l'enregistrement, compté en nombre d'octets. LR < ou = 255
Une variable flottante occupe 17 octets.
Une variable entière occupe 8 octets.
Une variable chaîne occupe la longueur de la chaîne + 2 octets
Exemple : Si C$ = "TOTO", la rubrique occupe 4 + 2 c'est à dire 6 octets.
Si vous voulez écrire A,B%, C$ dans un enregistrement, il doit avoir au minimum 17+8+6=31 octets.
LR = 31.
NE = Nombre d'enregistrements réservés à l'avance.
*Action : Créer le fichier NOMFICH en accès direct composé de NE enregistrement de LR octets chacun et lui associer le numéro logique NLU.

C'est à vous de savoir ce que vous allez mettre dans chaque enregistrement pour calculer sa taille.
Si vous avez réservé NE enregistrements, vous pouvez accéder directement à n'importe lequel de ceux-ci.

-2 Ecriture !WRITE

Syntaxe : !WRITE NLU,E'< Liste de variables>
ou !WRITE NLU,E REM <Liste de variables> pour la version ORIC 1.
*Paramètres :
NLU = Numéro logique du fichier sur lequel on écrit.
1 < ou = NLU < ou = 255 NLU=0 interdit.
E = Numéro de l'enregistrement sur lequel il va écrire. E peut être une variable.
1 < ou = E < ou = #FFFF
"Liste des variables" Suite de variables séparées par des ","
Tous les types de variables sont acceptés sauf la variable en tableau comme A$(3,3).
ATTENTION Lorsqu'une variable chaîne est utilisée dans la liste de variables, elle doit exister auparavant. Sinon il faut la définir par exemple par A$ =" ". avant d'utiliser cette instruction.

* Action : Ecrire dans le fichier à accès direct de numéro NLU, à1l'enregistrement E, la suite des valeurs correspondant à la liste des variables. E doit être inférieur ou égal au nombre d'enregistrements réservés; E peut être supérieur de 1 au nombre d'enregistrements total. A ce moment on rajoute un enregistrement au fichier.
Seule la taille de l'enregistrement est fixe. Contrairement aux fichiers séquentiels, la taille des rubriques à l'intérieur d'un enregistrement peut être remodifiée car chaque fois c'est l'enregistrement entier qui est écrit. Vous ne pouvez pas écrire ou modifier une rubrique particulière d'un enregistrement. La place restante d'un enregistrement est remplie par des "Espaces".
ATTENTION
Avec l'ATMOS, les deux syntaxes sont valables.
Avec l'ORIC i seule la syntaxe avec "REM" est conseillée.

-3 Lecture !TAKE
Syntaxe : !TAKE NLU,E' "Liste de variables"
ou !TAKE NLU,E REM "Liste de variables" pour la version ORIC 1
*Paramètres :
NLU : Numéro logique du fichier sur lequel on lit.
1 <ou = NLU <ou = 255 NLU=0 est interdit.
E = Numéro de l'enregistrement sur lequel il va lire, E peut être une variable. 1 < ou = E < ou = #FFFF
"Liste de variable" suite de variables séparées par des ",". Tous les types de variables sont acceptés sauf la variable chaîne en tableau comme A$(13,3). Il ne doit donc pas y avoir d'espace derrière le signe $.
ATTENTION
Lorsqu'une variable chaîne est utilisée dans la liste de variables, elle doit avoir exister avant. Sinon il faut la définir par exemple par A$ avant d'utiliser cette instruction.
* Action : Lecture dans le fichier à accès direct de numéro NLU de l'enregistrement N0E. E doit être inférieur ou égal au nombre d'enregistrements total du fichier. A chaque lecture vous lisez depuis le début de l'enregistrement.

-4 Indication du nombre total d'enregistrements !WHERE
Cette instruction s'emploie comme en fichier à accès séquentiel. Voir § correspondant..

OPÉRATIONS SUR LES MATRICES

-1 Sauvegarde de la matrice et du tableau dans un fichier !MSAVE

Syntaxe: !MSAVE "(N0DISQ.)NOMFICH = NOMAT"
* Paramètres :
NOMFICH = Nom de fichier sur disquette.
TYPE = Le type de fichier sera d'office ARY, c'est à dire Matrice (ARRAY).
NOMAT = Nom de la variable tableau ou matrice du programme BASIC.
Si la matrice est A$ (5,5), NOMAT est A$, il ne faut plus mettre les parenthèses.
* Action : Sauvegarder la matrice de nom NOMAT du programme BASIC,
sur le disque en lui attribuant le nom A$
Une précaution particulière est à prendre lorsque vous utilisez des matrices de chaîne. Tous les éléments de la matrice doivent être définis avant la sauvegarde. Nous vous conseillons donc, juste après la déclaration de la matrice par DIM, de la remplir d'office par des blancs.
Exemple

```
DIM A$(5,5)
FOR I = 0 TO 5 : FOR J = 0 TO 5
AS (l,J> = "--"
NEXT J : NEXT I.
```

("--" veut dire "espace".)
*N'oubliez pas l'élément d'indice 0, par exemple A$(0,0), d'une matrice.

-2 Chargement de Matrice ou de tableau !MLOAD

Syntaxe: !MLOAD "(N°DISQ.)NOMFICH= NOMAT"
* Paramètres :
NOMFICH = Nom de fichier sur disquette
TYPE = Le type de fichier sera d'office ARY c'est à dire (ARrAY)
NOMAT = Nom de la variable tableau ou matrice du programme BASIC. Si la matrice est A$(5,5>, NOMAT est A$, il ne faut plus mettre les parenthèses.
*Action : Charger le fichier de nom NOMFICH.ARY dans la matrice NOMAT du programme BASIC.
Ilva de soit que la matrice était définie et dimensionnée.
Le système ne pourra charger un fichier contenant une matrice de chaîne de caractères dans une matrice de variable réel, et réciproquement. Il ne tient donc qu'à vous de bien gérer les types de matrice en mettant par exemple "$" dans le nom de fichier pour signaler une matrice chaîne. Si la dimension de la matrice à charger et celle de la matrice réceptrice ne sont pas identiques, vous ne retrouverez plus les valeurs au même endroit.

SAUVEGARDE DE L'ÉCRAN

-1 Sauvegarde en basse résolution !LSCR

Syntaxe : !LSCR "(N°DISQ.)NOMFICH"
*Paramètres :
NOMFICH = Nom de fichier sur disque
TYPE = Le type sera d'office SCR pour SCREEN c'est à dire ECRAN
*Action : Sauvegarder l'écran, c'est à dire l'espace mémoire de #BBBO à #BFDF, sur disque en lui attribuant le nom NOMFICH.SCR

-2 Sauvegarde en haute résolution !HSCR

Syntaxe : !HSCR"(N°DISQ.)NOMFICH"
*Paramètres :
NOMFICH = Nom de fichier sur disque TYPE = Le type sera d'office SCR pour SCREEN c'est à dire ECRAN.
* Action : Sauvegarder l'écran en haute résolution, c'est à dire l'espace mémoire de #AOOO à #BFDF sur disque en lui attribuant le nom NOM FICH.SCR.

EXÉCUTION AU DÉMARRAGE

-1 Assignation pour le démarrage automatique !START

Syntaxe: !START "NOMFICH(.TYP)"
*Paramètres :
NOMFICH = Nom de fichier en i seul caractère alphabétique
TYPE = .BAS, BIN, .CMD. BAS par défaut.
* Action : Prévenir le système que le programme NOMFICH.TYP sera lancé automatiquement après le chargement du système (BOOT). Avant de sortir d'un programme à lancement automatique, n'oubliez pas de remettre #CBED en #1B pour ORIC 1 ou #CCBO pour ATMOS. (ex : DOKE #1B, #CBED)
Exemple
Insérer dès le début du programme l'instruction suivante
Pour ATMOS : DOKE #1B, #CCBO
Pour ORIC 1 : DOKE #1B, #CBED
En langage machine n'oubliez pas d'interdire les interruptions et ensuite de leur rendre la main.
Une fois le programme assigné, faites : !MASTER "NOMSYS" pour figer la situation.

-2 Pas de démarrage automatique !UNSTART

Syntaxe : !UNSTART
*Action : Si vous ne voulez pas de démarrage automatique il suffit de faire !UNSTART puis !MASTER "NOMSYS".

APPEL D'ECRAN D'AIDE MÉMOIRE

-1 Appel d'écran d'aide mémoire !HELP

Syntaxe : !HELP "(N0DISQ.)NOMFICH"
* Paramètres :
N0DISQ : Numéro de l'unité disque de 1 à 4. valeur par défaut : 1.
NOMFICH : Nom du fichier sur 8 caractères alphanumériques maximum, obtenus pas troncature à droite. Le premier caractère doit être obligatoirement alphabétique (A à Z).
TYPE : Forcé à SCR, c'est à dire type d'écran. Vous n'êtes donc pas obligé de le mettre.
*Action : Charger la page d'écran de nom NOMFICH. C'est l'équivalent de !LOAD "NOMFICH.SCR". Vous pouvez donc appeler des pages d'aide mémoire sans écraser votre programme BASIC actuel.
Exemple !HELP "WS" vous affiche la page d'aide mémoire d'utilisation de la commande !WS, qui est sauvegardée sous le nom , "WS.SCR.". (voir aussi !HSCR et !LSCR).

ACCÈS DIRECT AUX SECTEURS

Vous pouvez lire ou écrire un bloc de 256 octets directement sur un secteur quelconque de la disquette. Un secteur est repéré par
-le numéro de lecteur (NLU) de 1 à 4, situé à l'adresse #48C
-le numéro de piste (NP) de 0 à 81, situé à l'adresse #48D
-le numéro de secteur (NS) de 1 à 17, situé à l'adresse #48E Lorsque vous lisez un secteur, il doit être transféré en mémoire centrale. Il faut donc indiquer l'adresse DB du début du bloc dans lequel les 256 octets du secteur doivent être transférés.
Lorsque vous écrivez sur un secteur, il faut indiquer l'adresse DB du début du bloc de 256 octets à transférer dans le secteur. Cette adresse DB se trouve en #48F et #490.

-1 Recherche du premier secteur libre !FS.

Syntaxe : !FS.
*Action : charger dans NP et NS (#48D et #48E) le numéro de piste et le numéro de secteur du premier secteur libre de la disquette du lecteur désigné par NLU (#48C).
*Remarque : avant d'utiliser !FS, NLU doit être défini, par exemple par POKE #48C,1.

-2 Lire un secteur de la disquette !RS

Syntaxe : !RS
*Action : Transférer 256 octets d'un secteur de la disquette déterminé par le numéro de lecteur NS, dans un bloc en mémoire centrale, commencé à l'adresse DB.
*Remarque : avant d'utiliser !RS; NLU, NP et DB doivent être définis par exemple par des POKE et DOKE.

-3 Ecrire sur un secteur de la disquette !WS

Syntaxe : !WS.
*Action : Transférer 256 octets de la mémoire centrale dans un secteur de la disquette déterminé par le numéro de lecteur NLU, le numéro de piste NP, le numéro de secteur NS. le bloc de 256 octets en mémoire centrale est déterminé par son adresse de début de bloc DB.
Remarque : avant d'utiliser lRS; NLU, NS, NP et DB doivent être définis, par exemple par des POKE et DOKE.

Exemples

Transférer le secteur 5 de la piste 18 du lecteur 2 en mémoire centrale à partir de l'adresse #3000.

```
POKE #48C,2
POKE #48D,18
```

Transférer un bloc commençant en #3000 sur le lecteur 2, piste18, secteur 5

```
POKE #48C,2
POKE #48D,18
```

```
POKE #48E,5                    POKE #48E,5
DOKE #48F,#3000                DOKE #48F, #3000
!RS.                           !WS
```

Par !WS on peut écraser des informations déjà existantes sur la disquette donc il faut avant d'utiliser cette fonction, utiliser la fonction FS afin de connaître la première piste/secteur libre.
!WS écrit un secteur puis le réserve s'il ne l'a pas encore été, afin que lors d'un !SAVE ou !CREATE etc... ce secteur ne soit pas écrasé. Ce qui a été écrit par !WS n'apparaît pas au catalogue.

-4 Effacer un secteur de la disquette !DS.
Syntaxe : !DS
Action : enlever la réservation d'un secteur du disque. Ce secteur peut donc être utilisé lors d'un !SAVE ou !WRITE, etc...
Remarque : avant d'utiliser !DS; NLU, NS, et NP doivent être définis, par exemple par des POKE et DOKE.
Exemple : Effacer le secteur 5 de la piste 18 du lecteur 2.

```
POKE #48C,2
POKE #48D,18
POKE #48E,5
!DS.
```

ANNEXE 1 : OCCUPATION MÉMOIRE

#3F4
à Registres d'entrée/sortie de la carte contrôleur
#3FF

#400
à Lien entre les 16K RAM overlay et la ROM Basic
#4FF

#C000
à TDOS en RAM overlay
#FFFF

ANNEXE 2 : COMMENT ENTRER DANS LA MEV PARALLÈLE

La particularité de l'ORIC et de l'ATMOS est d'avoir 16k octets de mémoire vive (MEV) qui ont les mêmes adresses que la mémoire morte du BASIC, donc cachées par cette dernière et non utilisées.
Nous l'appelons la MEV (ou RAM) "OVERLAY" ou parallèle. C'est là que se charge le T.DOS, laissant ainsi libre le reste de la mémoire. Si, pour des applications particulières vous voulez accéder à la MEV ,il suffit de suivre le protocole suivant
1- Interdire toute interruption par SEl
2- Mettre 127 dans #30E pour interdire les interruptions de la VIA
3-Mettre 1 dans #3FA, à ce moment tout accès aux mémoires de #COOO à #FFFF est dans la RAM OVERLAY
Remarque
Il faut interdire les interruptions avant d'entrer dans la RAM OVERLAY si les adresses du sous programme de traitement d'interruption ne sont pas mises à jour dans la RAM OVERLAY.
Pour se remettre en situation normale (avec la ROM BASIC):
1- Mettre 0 dans #3FA.
2- Mettre 192 ou #C0 dans #30E.
3-Remettre les interruptions par CLI.

ANNEXE 3 : ADRESSES DU CONTRÔLEUR

#3F4 COMMAND/STATUS REGISTER
#3F5 TRACK REGISTER
#3F6 SECTOR REGISTER
#3F7 DATA REGISTER
#3F8 SIDE SELECT
#3F9 DCR, DISK CONTROLER RESET
#3FA ORMA, OVERLAY RAM ACCESS
#3FB ROMDIS
#3FC DISK 1 SELECT
#3FD DISK 2 SELECT
#3FE DISK 3 SELECT
#3FF DISK 4 SELECT

ANNEXE 4 : TABLE DES MESSAGES D'ERREUR

Messages	Code	Signification
EXITING FILE	1	Le fichier demandé existe sur disque.
DRIVE NOT IN LINE	2	Vous avez demandé une opération sur une unité de lecteur qui n'a pas été défini au système par !MOUNT.
PROGRAM TOO LARGE	3	Le fichier traité est trop long. Vérifier vos adresses de début et fin.
FILE TYPE MISM	4	Discordance de type fichier.
DISK FULL	5	La disquette est pleine.
I/O ERROR	6	Erreur de lecture ou d'écriture. Vérifier les connections. Vérifier si votre disquette n'est pas abîmée ou effacée. Si l'erreur persiste contactez le service après vente le plus proche.
FILE NO FOUND	7	Le fichier demandé n'est pas sur le disque.
WRITE PROTECT	8	L'enregistrement, le fichier ou la disquette est protégé en écriture.
RANGE ERROR	9	Vérifier la taille de vos paramètres.
SYSTEM ERROR	10	Erreur grave de T.DOS.
SYNTAX ERROR	11	Vérifier la syntaxe et le type de variables.
TOO MANY OPEN FILES	12	Trop de fichiers ouverts en même temps. (16 maximum).
FREE MEMORY UNDERFLOW	13	Il ne reste plus de mémoire disponible. Voir HIMEM et FREE.
MISSING VARIABLE	14	Vous avez fait appel à une variable chaîne qui n'a pas été définie.
END 0F FILE	15	Vous avez tenté une opération (lecture/écriture) en dehors du fichier existant.
OUT 0F DATA	16	Vous avez tenté une opération (lecture/écriture) en dehors de l'enregistrement en cours.

Le code correspond à la valeur que trouvez en #489.

ANNEXE 5 : EXEMPLES

Sur la disquette livrée avec l'appareil se trouvent des programmes d'exemples commentés.

TESTSEQU.BAS	- Ouverture d'un fichier séquentiel.
SEQU2.BAS	- Lecture et écriture sur fichier séquentiel à utiliser après TESTSEQU.BAS
TESTDIR.BAS	- Exemples sur des fichiers à accès direct.
MATEST.BAS	- Sauvegarde des Matrices.

Extrait du manuel d'utilisation JASMIN 2 publié par la Société TRAN.
JASMIN est une marque déposée.