Paul Hogenboom

FLEX

le Système d'Exploitation de Disquettes du EC-6809

Ailleurs dans ce numéro nous vous proposons EC-6809, un ordinateur FLEX à réaliser à l'aide de deux cartes au format europe, ordinateur au coeur duquel bat, comme l'indique son nom, un 6809, processeur de bonne maison s'il en est (son dernier "petit" frère, le 68000 fait des malheurs). Pour pouvoir remplir une quelconque fonction utile, un ordinateur ne peut se passer de logiciel. Le Système d'Exploitation de Disquettes (SED) le plus répandu pour le 6809 est FLEX-9. Il existe également un second SED pour le traitement en temps réel conçu tout spécialement à l'intention du 6809, OS-9. La logithèque de programmes écrits à l'intention du FLEX-09 est extrêmement impressionnante tant par sa diversité que par sa puissance. Comme il s'agit de matériel d'origine professionnelle, il existe en France plusieurs sociétés vendant des logiciels tournant sous FLEX. S'il vous fallait un logiciel très spécifique, non commercialisé en France, il vous resterait toujours la possibilité de le commander aux Etats-Unis (le dollar est devenu notablement plus abordable ces derniers mois).

'68' MICRO Journal
5900 Cassandra Smith Rd
Hixson, TN 37343
$24.50 par an + $12 port surface

Le SED FLEX a été conçu à l'origine pour les unités centrales (CPU) de la série 68XX (telles que 6800, 6802, 6809). Hormis Thomson, très rares sont les fabricants d'ordinateurs personnels à utiliser le 6809, ce qui explique sans doute que le terme FLEX soit, côté amateur, relativement inconnu en France. Dans le monde professionnel, le FLEX a eu son heure de gloire, sachant qu'il est à l'heure actuelle relevé par un descendant notablement plus puissant, UNI-FLEX. Rares sont cependant les amateurs pouvant se payer UNI-FLEX, et de nombreuses petites firmes devront se saigner aux quatre veines pour pouvoir acquérir ce SED. Les initiés prétendent que la puissance de UNI-FLEX, un logiciel multi-utilisateur, dépasse celle d'UNIX, ce qui, vous en conviendrez, n'est pas peu dire.

Revenons-en à FLEX. FLEX-09 (09 pour indiquer la version conçue à l'intention du 6809) devrait être le SED le plus puissant disponible pour 6809; en pratique, sur de nombreux points, FLEX-09 dépasse CP/M, un autre SED célèbre, qui tel un phoenix, ne cesse de renaître de ses cendres. A noter que s'il vous faut travailler en temps réel, il vous faudra opter pour OS-9.

S'il est dans vos intentions de vous lancer à fond dans les arcanes du SED FLEX, nous ne pouvons que vous recommander de prendre un abonnement à une revue américaine "68 Micro Journal" spécialisée tant dans les extensions matérielles que logicielles consacrées au 6809. Si de plus vous ne reculez pas devant l'aventure de commander une disquette de programmes aux Etats-Unis, c'est sûrement la revue qu'il vous faut. En effet, l'un des points forts de FLEX est de permettre à tout possesseur d'un SED FLEX de faire fonctionner tous les programmes FLEX sur son système.

Outre la firme américaine Technical Systems Consultants (TSC) qui possède les droits sur le FLEX, de nombreuses autres sociétés spécialisées dans le logiciel proposent des programmes pour FLEX. Il nous est impossible de vous proposer une description complète des logiciels tournant sous ce SED, un numéro complet d'Elektor, même "centenaire" n'y suffirait pas. Si EC-6809 devient l'ordinateur chéri de ceux

d'entre nos lecteurs qui envisagent de se lancer dans la construction d'un ordinateur personnel, même si financièrement une telle construction est délicate à justifier et que nous soyons submergés sous leurs lettres, il se pourrait que nous revenions à l'occasion à parler de FLEX. Quoi qu'il en soit, il est bon de savoir qu'il ne faut rien de plus que les deux cartes décrites dans l'article consacré au EC-6809, pour pouvoir se lancer dans une grande aventure de découverte. Pour vous donner les premiers points de repère, nous allons maintenant passer en revue les différents sous-programmes de FLEX. Bien qu'incomplet, ce tour d'horizon devrait vous donner une bonne idée sur les possibilités du FLEX.

Le logiciel-système

Les divers fichiers nécessaires au démarrage de l'ordinateur, le "boot" de nos amis d'outre-Manche et Atlantique, se trouvent sur la disquette-système.
Prenons-les dans un ordre arbitraire.

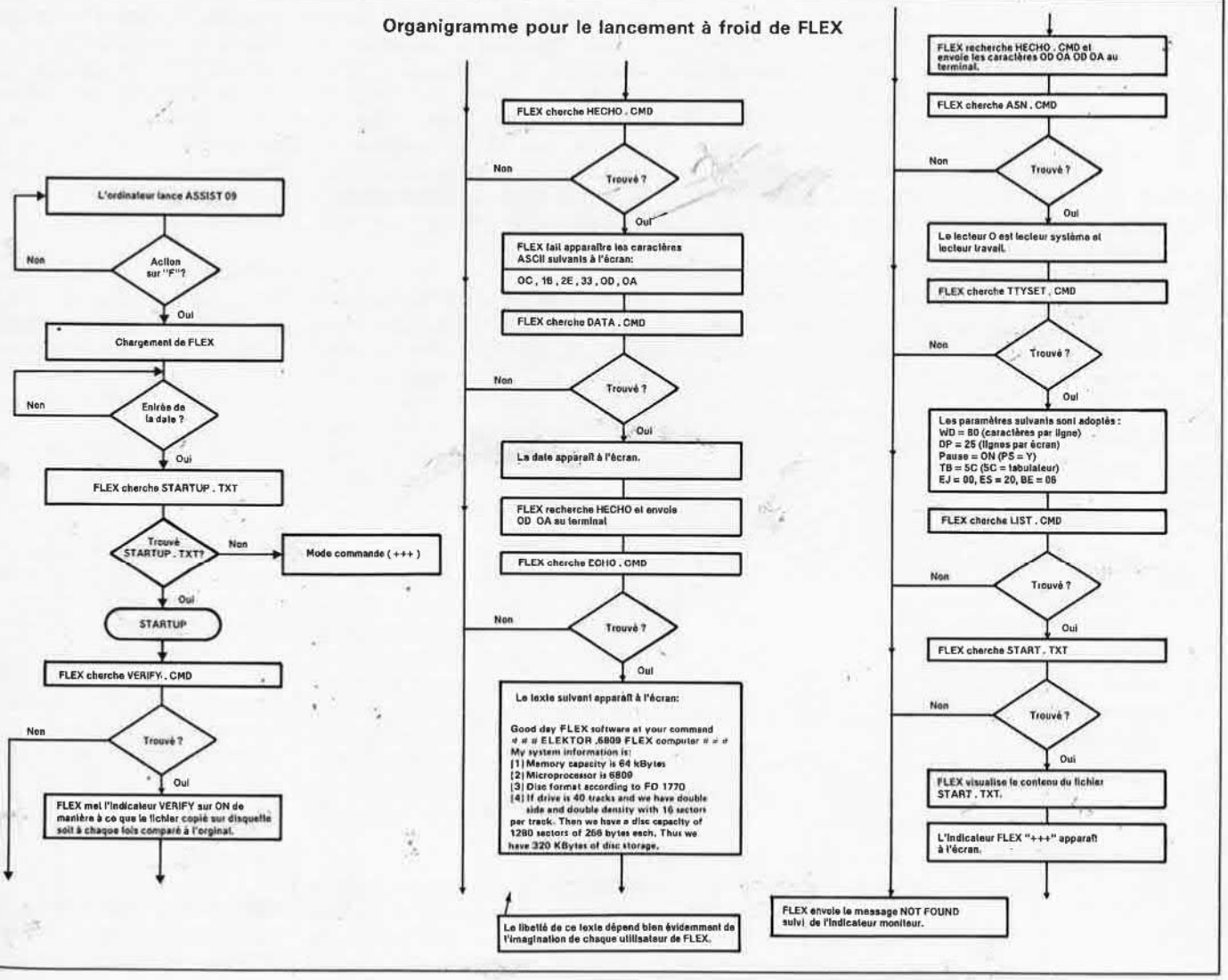
1

Dans les trois exemples, le texte introduit par l'utilisateur est souligné.

```
+++ LIST STARTUP.TXT
0.VERIFY IS ON
0.HECHO 0C
0.HECHO 1B 2E 33
0.HECHO 0D 0A
0.DATE
0.HECHO 0D 0A
0.ECHO Good day FLEX software at your command
0.ECHO     E L E K T O R  6809 Flex computer
0.ECHO My system information is:
0.ECHO  1  Memory capacity is 64 KBytes
0.ECHO  2  Microprocessor is 6809
0.ECHO  3  Disc format according to FD 1770
0.ECHO  4  If drive is 40 tracks and we have double side
0.ECHO     and double density with 16 sectors per track.
0.ECHO     Then we have a disc capacity of 1280 sectors
0.ECHO     of 256 bytes each. Thus we have 320 KBytes of
0.ECHO     disc storage.
0.HECHO 0D 0A
0.HECHO 0D 0A
0.ASN W=0 S=0
0.TTYSET WD=80 DP=25 PS=Y
0.TTYSET TB=5C EJ=00 ES=20 BE=08
0.LIST 0.START1.TXT
+++
```

2

Organigramme pour le lancement à froid de FLEX



A tout seigneur tout honneur, il nous faut bien évidemment commencer par FLEX lui-même (nom de baptême FLEX.SYS sur la disquette). Pour simplifier la vie de l'utilisateur FLEX est accompagné d'un fichier système qui génère les divers messages d'erreur (ERRORS.SYS).
Après avoir chargé le SED de la disquette système, FLEX se met à la recherche un fichier de commande (suffixe: .CMD) baptisé EXEC (de execute): il s'agit de EXEC.CMD, bien évidemment. FLEX exécute en effet la ligne de commandes contenue dans un fichier Texte répondant au doux nom de STARTUP.TXT. Le plus souvent, cette ligne s'écrit EXEC 0.STARTUP.TXT. Pour cette raison, il est nécessaire que la disquette système comporte un fichier "STARTUP.TXT", fichier que l'utilisateur pourra écrire lui-même en s'aidant d'un logiciel de traitement de texte. Ce fichier comporte plusieurs lignes de commandes exécutées séquentiellement, c'est-à-dire à la succession l'une de l'autre. L'instruction EXEC 0.START.TXT signifie mot à mot: exécute l'une après l'autre les différentes commandes contenues dans le fichier Texte START.TXT. La **figure 1** donne un exemple type d'un tel fichier, la **figure 2** donnant quant à elle l'ordinogramme de la chronologie du processus. Ces deux illustrations indiquent très clairement la fonction remplie par le fichier STARTUP.TXT sur l'ensemble de la chronologie de démarrage de l'ordinateur, Comme le montre l'étude de la figure 1 il ne faut mettre qu'une seule commande par ligne.
Remarque générale, comme tous ces fichiers sont du type Commande, .CMD, il n'est pas nécessaire de doter chacun des fichiers explicitement de ce suffixe lors de son appel.

Entrons maintenant dans le détail des différents fichiers.

VERIFY ON, la première commande de STARTUP.TXT, force FLEX à charger le fichier de commande **VERIFY.CMD** qui a pour fonction de vérifier la correction de la procédure lors de l'écriture sur la disquette d'un fichier, par comparaison entre cette version et celle présente à cet instant dans la mémoire de l'ordinateur. Ce contrôle, pour extrêmement utile qu'il soit, coûte du temps, ce qui explique que cette ligne soit bien souvent absente du fichier STARTUP de nombreux utilisateurs.

HECHO.CMD (pour **H**exadécimal **ECHO**) est le fichier qui envoie une chaîne de caractères hexadécimaux vers le terminal. Ainsi l'exemple adopté HECHO 0C signifie: envoie le caractère ASCII de rang hexadécimal 0C à l'écran. 0C est le rang du caractère (de l'instruction en fait) Clear Screen = Effacement de l'écran. Pour commencer une nouvelle ligne on donnera l'instruction HECHO 0D 0A (0D = Retour Chariot, abbrégé CR = Carriage Return) et 0A = Saut de ligne, LF = line Feed).

DATE.CMD fait apparaître à l'écran la date entrée par le clavier lors du lancement de l'ordinateur. Cette commande permet en outre de modifier la date. Cette routine est très dépendante du matériel, de sorte qu'il arrive assez souvent qu'il faille modifier le fichier DATE.TXT et le réassembler avant qu'il ne fonctionne correctement.

ECHO.CMD est un fichier très proche de HECHO dont nous avons parlé plus haut. ECHO fait apparaître à l'écran des caractères alphanumériques qui peuvent par exemple être des messages système, des messages d'erreur. Chaque ligne du message, quelle que soit sa longueur, doit commencer par 0.ECHO.

ASN.CMD, (de to ASsigN = attribuer), définit quel est le lecteur de disquette maître (ou système (System) ou 0) et quel est le lecteur esclave (ou travail (Work) ou 1). Si l'ordinateur ne comporte qu'un unique lecteur de disquettes, la commande ASN prend la forme suivante: 0.ASN W=0 S=0, ce qui indique au système que le drive 0 est tout à la fois lecteur système et lecteur de travail.

TTYSET.CMD remplit une fonction importante, puisque c'est par son intermédiaire que sont définis les différents paramètres caractérisant le terminal. Parmi les plus importants citons: le nombre de lignes par écran, (DP = 25: 25 lignes par écran) et la sélection d'un caractère ASCII pour la fonction de tabulation TAB (5C dans le cas présent).

LIST.CMD est la dernière instruction de la **figure 1**. Elle produit l'apparition du fichier à l'écran; ainsi, 0.LIST 0.START1.TXT visualise à l'écran le contenu du fichier START1.TXT. Comme il est possible de passer du fichier START1.TXT à un logiciel de traitement de texte (à l'aide d'une instruction du genre 0.SCR DEMO.TXT par exemple), le fichier START1.TXT peut fort bien servir de programme d'introduction ou d'explication pour un logiciel plus complexe. Ces explications pourraient concerner FLEX en donnant un résumé des fonctions remplies par les instructions les plus importantes. Sur une seconde disquette, réservée à la création et déverminage de programmes en BASIC, un fichier que nous baptiserons START2.TXT pourra, par exemple, indiquer à l'utilisateur qu'immédiatement après avoir chargé le BASIC Etendu (Extended BASIC, par l'instruction 0.XBASIC), il peut taper les premières lignes de son programme. Il va sans dire que si l'on veut que les choses se déroulent sans accroc, il faut que tous les fichiers de commande appelés par START.TXT soient présent sur la disquette système (si l'on travaille avec un seul lecteur); si tel n'était pas le cas, FLEX reviendra à l'écran avec le message d'erreur "NOT FOUND".

LINK.CMD est un autre fichier presque aussi important que FLEX.SYS; ce fichier est indispensable pour effectuer le lien (link) entre FLEX et le logiciel de démarrage. La syntaxe à respecter est: LINK FLEX.

Pour terminer le passage en revue du SED, il nous reste à énumérer quelques fichiers moins fréquemment utilisés:
P et **PRINT.SYS** servent lorsque l'on désire envoyer un fichier non pas à l'écran, mais vers une imprimante sérielle. Au lieu d'ordonner LIST Nom du fichier.Type du fichier, on fera P LIST Nom du fichier.Type du fichier. Il est également possible de connecter une imprimante parallèle (sortie Centronics) au EC-6809 (par utilisation éventuelle d'un des ports du PIA (ou adjonction d'une carte supplémentaire) + le logiciel adéquat).

CAT.CMD et **DIR.CMD** visualisent sur l'écran le nom de tous les fichiers présents sur la disquette, sous forme

3

```
+++ newdisk
SCRATCH DISK IN DRIVE 0? Y
80 TRACKS (N=40 TRACKS)? Y
DOUBLE SIDED DISK? Y
DOUBLE DENSITY DISK? Y
VOLUME NAME? ELEKTOR
VOLUME NUMBER? 1
ARE YOU SURE? Y

FORMATTING COMPLETE
TOTAL SECTORS = 2560
+++
```

ra[illegible]ourcie[illegible]taloguée) dans le pre-[illegible]er ca[illegible]ne instruction P DIR 1 e[illegible] l'imprimante les noms de tous les fichiers existants sur la disquette se trouvant dans le lecteur 1. DIR fournit des informations plus complètes que CAT.

APPEND.CMD effectue la concaténation de plusieurs fichiers. Syntaxe: APPEND,Nom du fichier 1.Type du fichier,Nom du fichier 2.Type du fichier. . .etc. . .Nom du dernier fichier.Type du fichier.

BUILD.CMD est un programme relativement simple, que l'on utilise pour la création de fichiers texte. Syntaxe: BUILD,Nom du fichier. Inutile d'ajouter le Type du fichier, sachant que ce programme ajoute automatiquement le suffixe .TXT au fichier qu'il créé. L'une des applications principales de BUILD est la création de fichiers tels que STARTUP.TXT que nous avons mentionné plus haut.

COPY.CMD sert bien évidemment à recopier un fichier d'une disquette sur une autre.

DELETE.CMD permet de faire de la place sur une disquette en effaçant des fichiers périmés. Avant que n'ait lieu l'effacement proprement dit, le fichier DELETE pose la question de confiance "ARE YOU SURE" (ce qui signifie en quelque sorte "êtes-vous conscient des conséquences de votre action sur la touche CR!!!"). Une action sur la touche Y (pour yes = oui) provoque l'effacement des informations de piste et de secteur de la liste des programmes du catalogue ou du répertoire (directory), les informations proprement dites sont conservées, mais les pistes et secteurs en question sont libérés pour une future utilisation.

NEWDISC.CMD est sans doute le premier programme dont vous aurez à vous servir pratiquement puisque vous l'utiliserez pour formater une disquette vierge. Le programme NEWDISC pose un certain nombre de questions à l'utilisateur quant au nombre de pistes, à la densité d'écriture (simple = single ou double), au nom de baptême de la disquette et à son numéro. Après avoir terminé le formatage proprement dit, NEWDISC vérifie l'intégrité de chaque secteur et signale (le cas échéant) la présence de secteurs non utilisables par le message "BAD SECTORS". La **figure 3** donne un exemple du déroulement de NEWDISC.

O.CMD écrit sur disquette au lieu de les visualiser à l'écran les informations générées par l'exécution d'une autre commande (catalogue par exemple). La syntaxe correcte sera dans ce cas: O.DIR. On retrouvera alors sur la disquette le catalogue sous le nom DIR.OUT. O. . . (de "Output") génère donc un fichier destination (de sortie).

RENAME.CMD est le dernier fichier décrit dans cet article. Ce sous-programme permet de rebaptiser (nom et type) un fichier. Il faudra par exemple rebaptiser DEMO.TXT, un programme de démonstration de BASIC écrit à l'aide d'un logiciel de traitement de texte, avant de pouvoir l'exécuter. En effet, XBASIC, l'interpréteur BASIC est incapable de charger ou d'exécuter un logiciel du type .TXT. La commande RENAME DEMO.TXT DEMO.BAS transforme d'un coup de baguette magique un fichier .TXT en fichier .BAS (Code Source BASIC) que XBASIC sera en mesure d'exécuter.

Tous les fichiers évoqués jusqu'à présent, et bien d'autres font partie du logiciel de base de FLEX. Le (ou les) manuel(s) fourni(s) avec le lot de base, tel que "The FLEX Disc Operating System" ou l'ouvrage "Advanced Programmers Guide" en donne une description plus complète. Pour terminer cet article, nous donnons en **figure 4** un exemple de répertoire (directory) d'une disquette comportant les programmes que nous venons de mentionner. Avant d'en terminer, une remarque: FLEX est en mesure ce contrôler jusqu'à un maximum de quatre lecteurs de disquettes. Si l'on travaille avec un unique lecteur, ce lecteur 0 constitue et le lecteur système et le lecteur de travail (voir le programme ASN.CMD). Si l'on dispose de plusieurs lecteurs, il faut ajouter le numéro du lecteur concerné à la commande. Ainsi, pour connaître le contenu de la disquette présente dans le lecteur numéro 2, il faudra faire DIR 2. Avant toute commande, il faut attendre que le système soit en mesure d'exécuter une quelconque commande, ce que FLEX signale par l'affichage à l'écran d'un triple signe plus "+ + +".
La figure 4 donne un exemple de ce qui apparaît sur l'écran après exécution d'une commande DIR. On se trouve ici en présence d'un lecteur de disquette double face, double densité, de 40 pistes (ce qui nous donne un total de 1440 secteurs de 256 octets, soit 360 Koctets d'espace utilisable). ◼

4

```
+++ DIR
DIRECTORY OF DRIVE NUMBER 0
DISK: ELEKTOR  1   CREATED: 28-MAY-85

FILE#  NAME     TYPE    BEGIN  END    SIZE   DATE       PRT

   1  FLEX     .SYS    01-01  01-17   23   28-MAY-85
   2  ERRORS   .SYS    01-18  01-1B    4   28-MAY-85
   3  EXEC     .CMD    01-1C  01-1C    1   28-MAY-85
   4  STARTUP  .TXT    01-1D  01-1D    1   28-MAY-85
   5  VERIFY   .CMD    01-1E  01-1E    1   28-MAY-85
   6  ECHO     .CMD    01-1F  01-1F    1   28-MAY-85
   7  HECHO    .CMD    01-20  01-20    1   28-MAY-85
   8  DATE     .CMD    01-21  01-22    2   28-MAY-85
   9  ASN      .CMD    01-23  01-23    1   28-MAY-85
  10  TTYSET   .CMD    01-24  02-01    2   28-MAY-85
  11  LIST     .CMD    02-02  02-04    3   28-MAY-85
  12  LINK     .CMD    02-05  02-05    1   28-MAY-85
  13  P        .CMD    02-06  02-06    1   28-MAY-85
  14  PRINT    .SYS    02-07  02-07    1   28-MAY-85
  15  CAT      .CMD    02-08  02-0E    7   28-MAY-85
  16  DIR      .CMD    02-0F  02-13    5   28-MAY-85
  17  APPEND   .CMD    02-14  02-16    3   28-MAY-85
  18  BUILD    .CMD    02-17  02-17    1   28-MAY-85
  19  COPY     .CMD    02-18  02-1C    5   28-MAY-85
  20  DELETE   .CMD    02-1D  02-1E    2   28-MAY-85
  21  NEWDISC  .CMD    02-1F  03-01    7   28-MAY-85
  22  O        .CMD    03-02  03-03    2   28-MAY-85
  23  RENAME   .CMD    03-04  03-04    1   28-MAY-85

   FILES=23, SECTORS=76, LARGEST=23, FREE=1204
+++
```