

Guido de Cuyper

EC-6809 — Flex

un micro-ordinateur professionnel pour pas bien cher

EC-6809 est, en dépit de son faible encombrement, un système informatique très puissant. De par cette puissance et le nombre de logiciels disponibles, il est certain que de nombreux professionnels pourront être intéressés par un tel système. EC-6809 utilise un SED (système d'exploitation des disquettes) puissant: Flex-9 (pour Flexibilité 6809).

Le titre le trahit, le SED d'origine du EC-6809 est Flex. Sachant que fort peu d'entre nos lecteurs connaissent ce système, nous lui consacrerons un article spécifique. Comme nous le signalions dans l'introduction il existe une très grande variété de logiciels fonctionnant sous Flex, traitements de texte, tableurs, et autres programmes du même acabit.

Le matériel

Tout ordinateur, quel qu'il soit, peut, du point de vue du matériel, être subdivisé en plusieurs sous-ensembles fonctionnels, tous indispensables

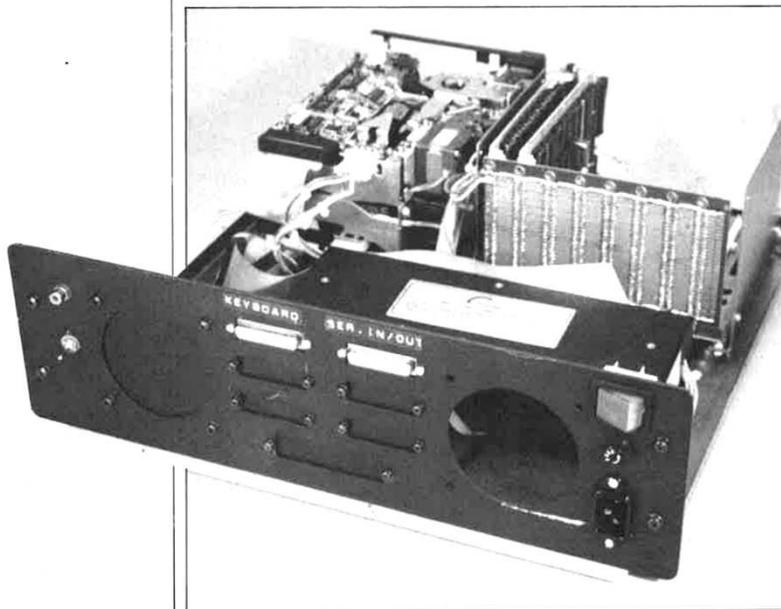
au fonctionnement du système. Il n'en va pas différemment dans le cas du EC-6809. Il comporte une unité centrale (CPU), de la mémoire, une interface vidéo et une interface de commande d'unité(s) à disquettes (floppy). Une simple estimation aura vite fait de vous donner à penser qu'il ne vous faudra pas moins de quatre cartes du format "europe" pour réaliser un tel ordinateur. Erreur. Nous avons réussi à diviser ce nombre par deux de sorte que l'ensemble du matériel tient sur deux eurocartes seulement: la première comporte l'unité centrale et la RAM dynamique, la seconde les

la vidéo et l'unité de disquettes).

Le synoptique des éléments constituant EC-6809 est donné en **figure 1**; cette dernière comporte en outre la dénomination des principaux circuits intégrés utilisés.

Deux cartes d'électronique ne peuvent bien évidemment pas, à elles seules, constituer un ordinateur complet; il faut leur adjoindre quelques périphériques reliés à l'électronique centrale par l'intermédiaire d'interfaces répondant au standard utilisé dans l'industrie. Pour constituer un système prêt à fonctionner, il nous faut en outre:

- Un clavier parallèle (ASCII)
- Un moniteur vidéo (composite ou pas) à entrée 75 ohms ayant une bande passante de 16 MHz au minimum.
- Au moins un lecteur de disquettes 40 ou 80 pistes dont le connecteur réponde au standard du bus Shugart (le plus fréquent). A noter qu'il faut que ce lecteur de disquettes soit du type double face (donc deux têtes de lecture face à face), sachant que la disquette système comporte des programmes système sur ses deux faces.
- Une alimentation fournissant 5 V sous 3 A, + 12 V sous 2 A (pour le (ou les) lecteur(s) de disquettes) et - 12 V sous 250 mA (pour l'interface RS232). Si votre alimentation possède des caractéristiques supérieures à celles que nous venons de définir, pas de problème, bien au contraire, elle sera en mesure de faire face à d'éventuelles extensions.
- Une platine de bus dotée de plu-



sieurs connecteurs femelles à 64 broches, l'Omnibus décrit en décembre 1983 par exemple.

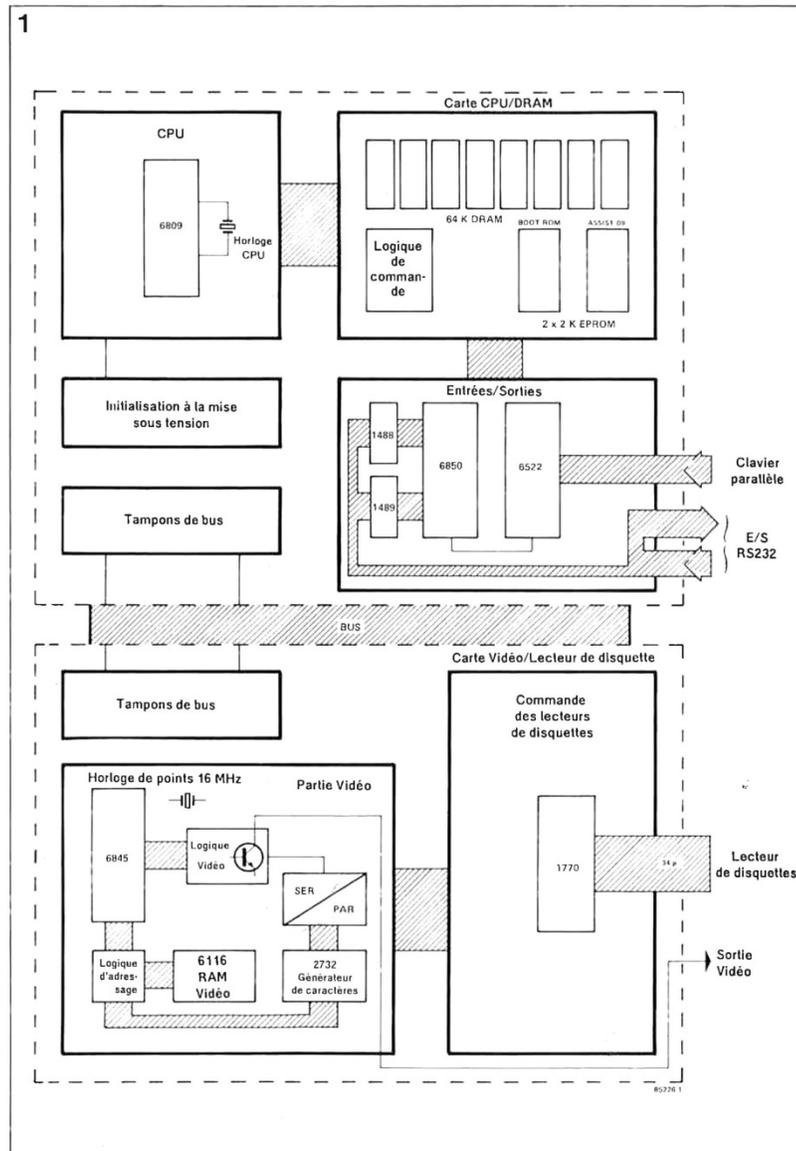
En résumé on peut dire qu'il s'agit d'un ordinateur construit comme la plupart de ceux basés sur des cartes de format europe. Les photos d'illustration montre que nous avons placé tout ce beau monde dans un boîtier "type IBM".

L'unité centrale (CPU)

Les figures 2 à 4 donnent les schémas de la carte de CPU/RAM dynamique, la figure 2 se concentrant plus particulièrement sur la CPU et les circuits connexes.

Le coeur d'un système est son microprocesseur. Dans EC-6809 bat un 6809 (d'où son nom), car comme vous n'êtes pas sans le savoir, le Flex a été écrit pour le 6800 puis adapté au 6809. La version écrite pour ce dernier a été baptisée Flex-9. Le progiciel de EC-6809 (Assist-09 et Bootrom) est contenu dans deux EPROM du type 2716 (IC13 et IC14 de la figure 2). Pour des raisons de prix de revient des circuits intégrés, nous avons adopté une fréquence d'horloge du système de 4 MHz, 3,6864 MHz très exactement, cette fréquence mettant à disposition un certain nombre de taux de transmission pour l'ACIA (nous y reviendrons un peu plus loin). EC-6809 peut également fonctionner à 8 MHz: il faudra dans ce cas remplacer non seulement le quartz X1 par un quartz de 8 MHz, mais aussi modifier les valeurs des condensateurs C1 et C2, remplacer les circuits intégrés de la famille 68XX par des circuits identiques mais du type B, et utiliser des circuits de RAM dynamique plus rapides. Il faudra en outre modifier la chronologie des signaux de commande des RAM. Comme on peut le constater, cette tâche loin d'être herculéenne, n'en demande pas moins une expérience certaine quant à la mise en oeuvre des instruments de mesure et la définition d'une nouvelle chronologie des signaux de commande des RAM. Comme indiqué précédemment, l'ensemble des circuits intégrés, le matériel donc, n'occupe pas plus de deux cartes "europe". Le dessin des circuits imprimés est basé sur le bus défini en figure 7, ceci de manière à simplifier plus tard l'adjonction d'éventuelles extensions.

Le bus complet (bus d'adresse, de données et de commande) est tamponné par les circuits tampons IC11,



IC12 et IC13 qui constituent en quelque sorte une interface de liaison entre les deux cartes. L'implantation des ponts et la combinaison des portes logiques de la figure 2 mérite quelques explications. Les ponts permettent de redéfinir IC13 et d'adapter le montage à une RAM CMOS 6116 au cas où l'on désirerait utiliser la carte pour une application différente de celle prévue ici (du point de vue purement technique, rien n'interdit d'utiliser ces deux cartes pour réaliser un terminal). Dans le cas de l'ordinateur EC-6809, IC13 est une EPROM 2716 et les interconnexions réalisées sont celles indiquées sur le schéma de la figure 2. Lors de la description du progiciel

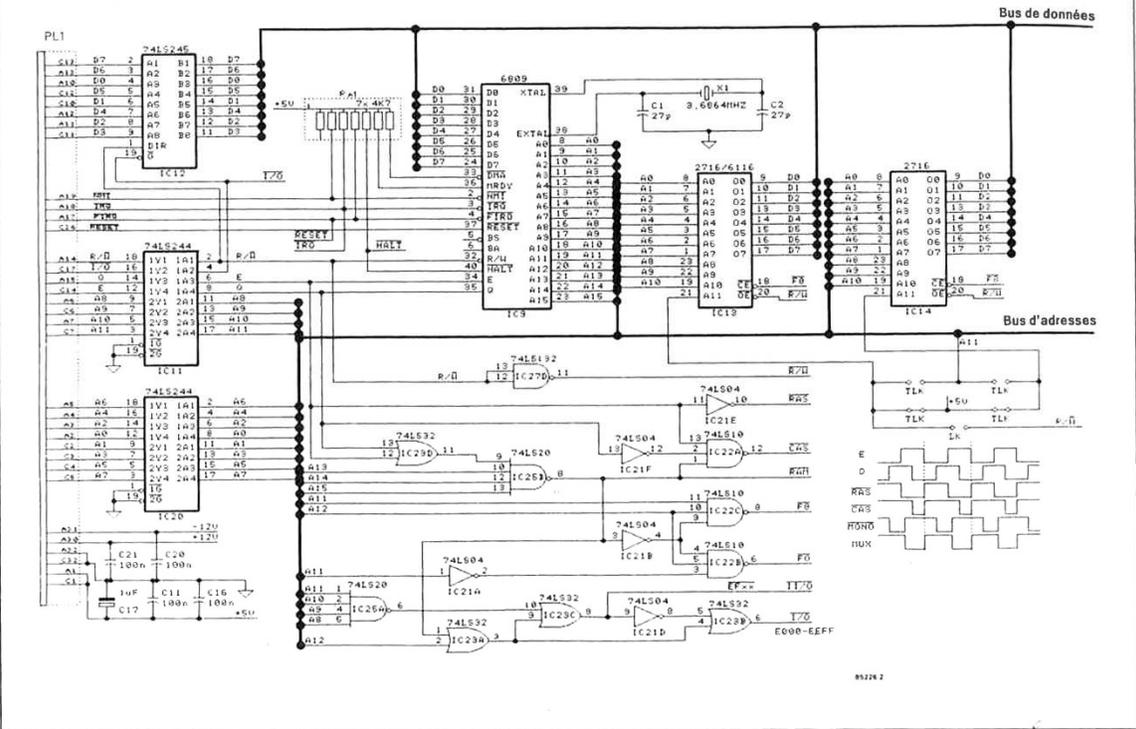
nous reviendrons sur le contenu des deux EPROM (IC13 et IC14). La fonction de la combinaison des portes présentes dans la partie centrale basse du schéma de la figure 2 est de générer les divers signaux de sélection de circuits tels que les signaux RAS (Row Address Select) et CAS (Column Address Select) nécessaires aux circuits de RAM dynamique.

Mémoire et Entrées/Sorties

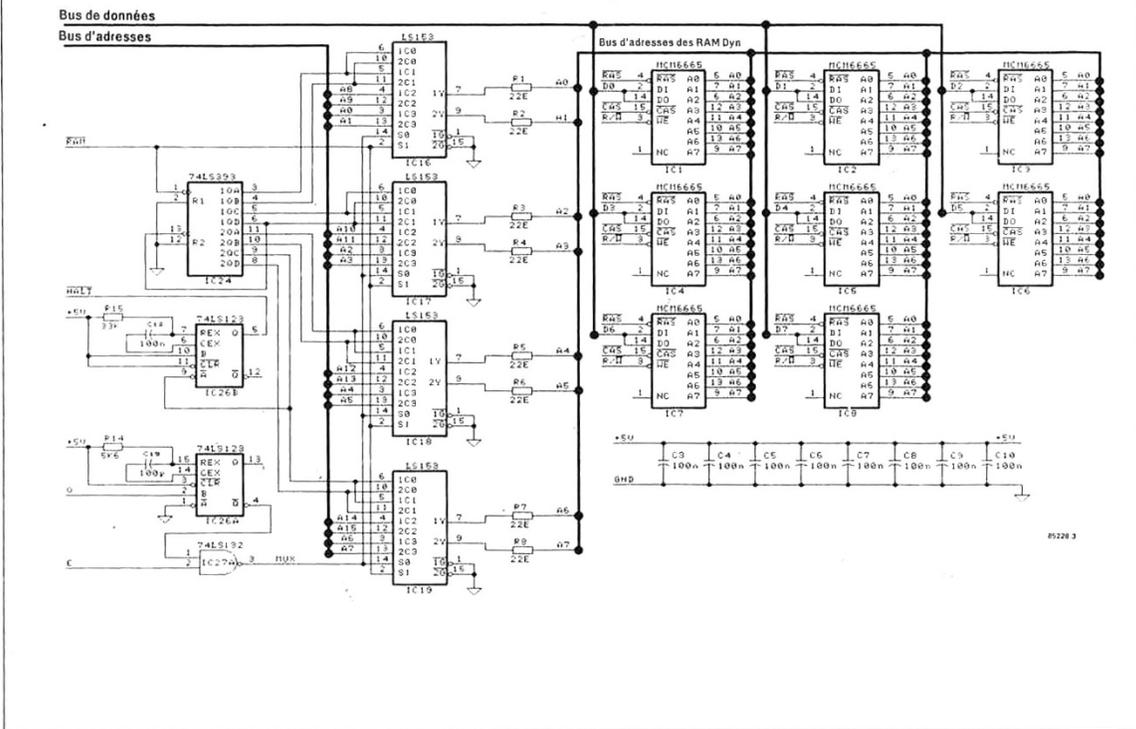
Comme l'illustre le schéma de la figure 3, la mémoire de notre ordina-

Figure 1. Synoptique donnant les divers sous ensembles fonctionnels de l'EC-6809 et leur répartition sur les deux cartes de format europe.

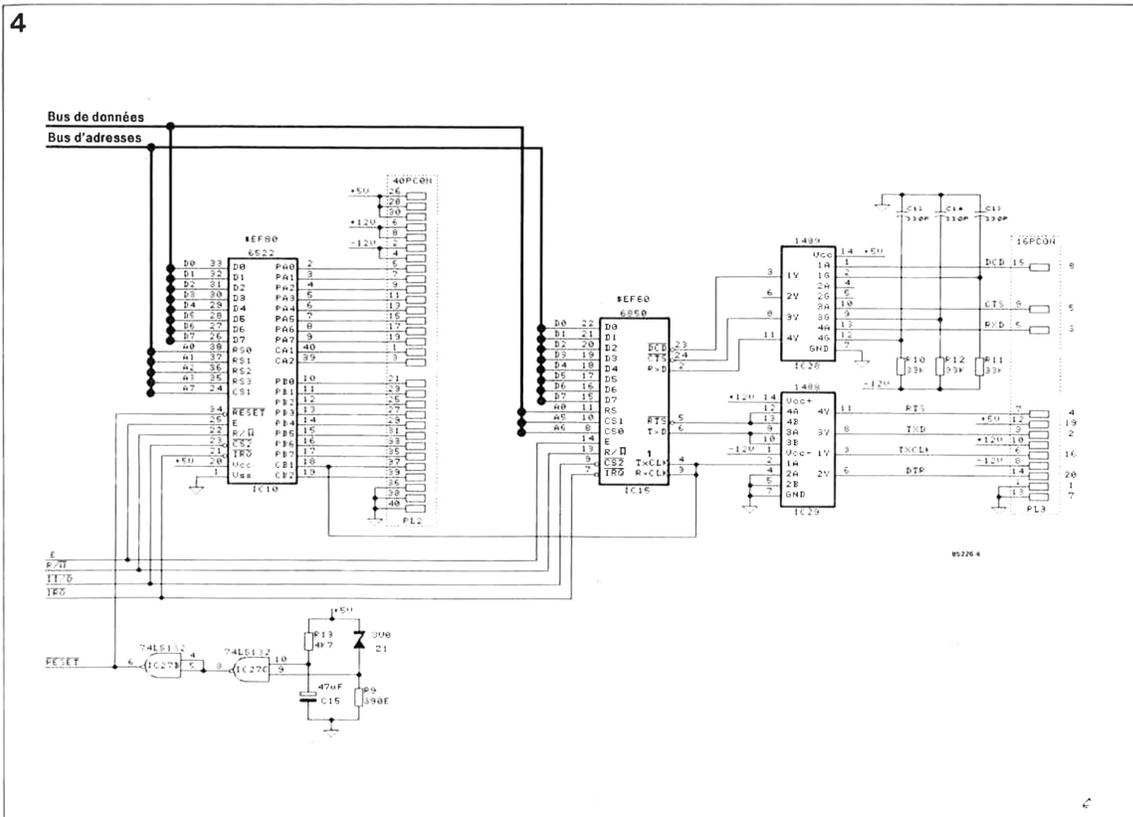
2



3



4



teur Flex est constituée de 8 circuits de mémoire du type 6164 (64 K x 1 bit), IC1...IC8, les autres circuits intégrés servant à la commande des précédents. En bas à droite de la figure 2 est donné le chronodiagramme des signaux de commande. Outre les sous-ensembles mentionnés précédemment, nous trouvons sur cette première platine les circuits assurant la liaison avec le monde extérieur. Un coup d'oeil à la figure 4 nous apprend qu'il s'agit d'une paire de circuits d'Entrées/Sorties (E/S ou I/O pour Input/Output), IC10 et IC15, et d'une paire de circuits d'adaptation de niveau (IC28 et IC29). Le matériel du système est conçu de manière à permettre l'émission ou la réception de données soit en mode parallèle soit en mode série. L'échange de données en mode parallèle, avec le clavier ou une imprimante à interface Centronics par exemple est pris en compte par un VIA du type 6522 (IC10). Un ACIA modèle 6850 associé aux circuits d'adaptation de niveau (IC28/IC29) constitue une interface série type RS232. L'ordinateur reçoit les commandes et autres informations fournies par l'utilisateur par l'intermédiaire d'un clavier à sortie parallèle. L'interface nécessaire est fournie par l'un des ports du 6522 (le

port A de IC10 de la figure 4 en l'occurrence), le port non utilisé pouvant (avec le logiciel adéquat), servir, par exemple, de sortie Centronics. L'interface série RS232 peut être utilisée pour bien des applications, telle que la transmission de données par modem. La disquette système comporte un programme de modem (MODEM.COM). Un coup d'oeil aux entrées d'horloge du 6850 (broches 3 et 4 de IC15) explique la raison du choix d'une fréquence de quartz aussi "biscornue". Le progiciel programme le port CB1 du VIA (broche 18 de IC10) en diviseur de fréquence qui abaisse la fréquence d'horloge du système à la valeur du signal d'horloge nécessaire à l'ACIA. Comme le 6809 procède à une division par quatre interne de la fréquence d'horloge du système et que cette dernière doit être aussi proche que possible de 4 Mz, la fréquence de quartz adoptée est de 3,6864 MHz. Cette solution "généialement simple" permet de se passer de générateur d'horloge et d'étages de division pour l'ACIA. Ce procédé ne fait pas qu'écraser les coûts; en effet, c'est grâce à lui et aux cinq ponts qu'il faudra monter sur le côté pistes (par opposition à côté composants) de cette carte (nous y reviendrons) que

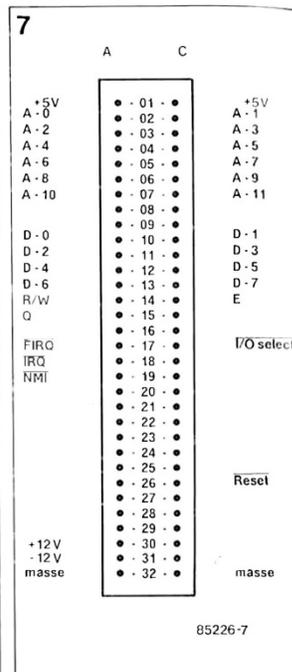
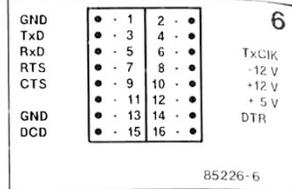
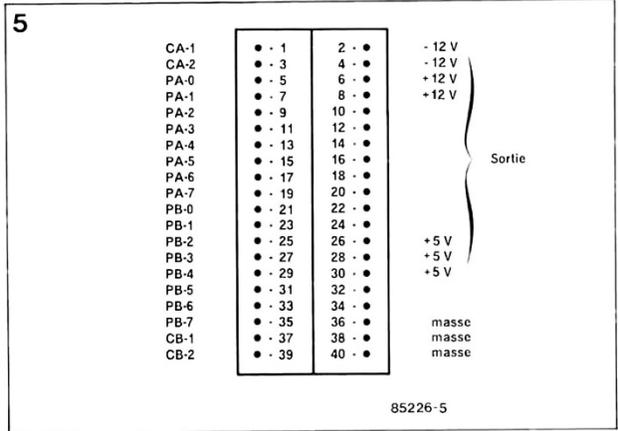
nous sommes arrivés à concentrer autant de circuits sur une si faible surface. La figure 5 définit le brochage du connecteur à 40 broches auquel arrivent les ports parallèles pour le clavier et l'imprimante. La figure 6 donne quant à elle le brochage du connecteur à 16 broches prévu pour l'interface série, le schéma de la figure 4 indiquant en outre la numérotation correspondante des broches d'un connecteur à 25 broches type Sub-D répondant aux normes RS232. Comme indiqué précédemment, la figure 7 donne le brochage des connecteurs de bus assurant l'interconnexion des deux cartes. Lors de la réalisation de cartes d'extension pour EC-6809 il faudra bien évidemment veiller à le respecter. La paire de portes IC27B et IC27C de la figure 4 constitue un dispositif d'initialisation à la mise sous tension (Power On Reset) initialisant l'ordinateur lors de la mise sous tension, ce qui rend inutile tout bouton de remise à zéro. Bien qu'il ne soit pas prévu, il est facile d'en implanter un entre la broche Reset (26c) du connecteur et la masse (32a ou 32c). Il évite de devoir couper la tension d'alimentation lors d'un "plantage" du système, (ce qui ne manque pas d'arriver en particulier lors de la conception de

Figures 2...4. Il ne faut pas moins de trois schémas (en DAO s.v.p.) pour restituer la complexité plus apparente que réelle de la carte CPU/RAM dyn. On ne sera donc pas trop surpris de se trouver confronté à une implantation des composants un peu plus dense que d'habitude.

Figure 5. Brochage du connecteur d'E/S de l'interface parallèle. Le port parallèle PA sert d'entrée aux informations fournies par le clavier. Le bit 0 de ce dernier arrive à PA 0, le bit 1 à PA-1, etc. Le signal d'échantillonnage (strobe) nécessaire à la prise en compte des données (lors de l'arrivée du premier flanc de ce signal) est appliqué à CA-1. L'ACIA (6850) reçoit son signal d'horloge par l'intermédiaire de CB-1, ce qui explique que cette broche soit également reliée aux broches 3 et 4 du 6850. Le port PB pourra être utilisé en port parallèle (interface Centronics) pour une imprimante par exemple.

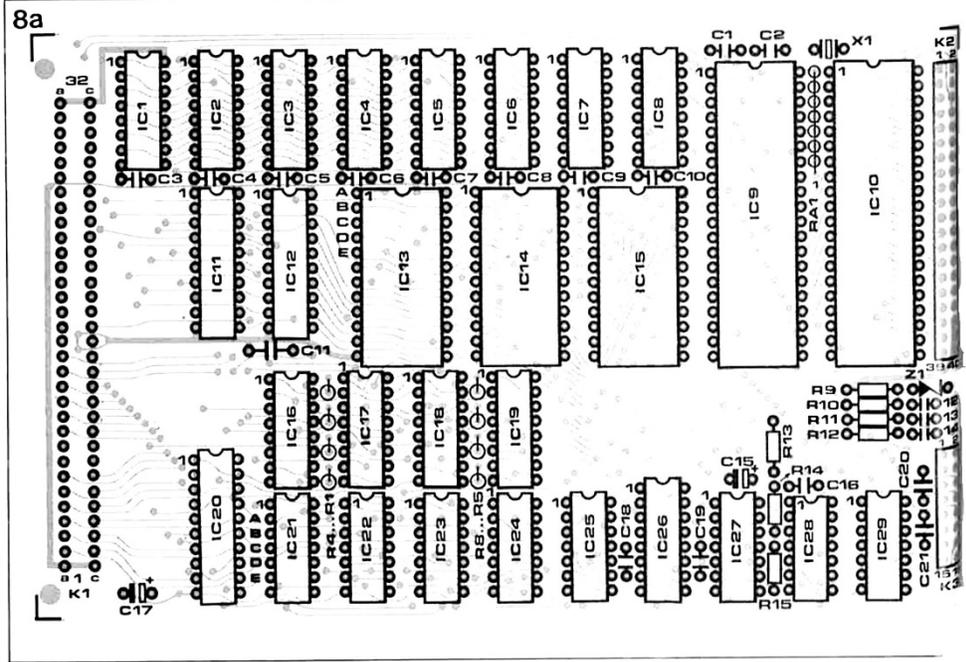
Figure 6. Brochage du connecteur de sortie de l'interface série. Il correspond à celui utilisé avec EC-68. On peut transformer ce connecteur en sortie répondant aux normes RS232 en reliant ce connecteur à un connecteur châssis Sub D à 25 broches en respectant la numérotation donnée à droite de la figure 4.

Figure 7. Brochage du connecteur de bus de EC-6809. L'ordinateur Flex d'Elektor. (Attention, il diffère notablement de celui respecté par les différents

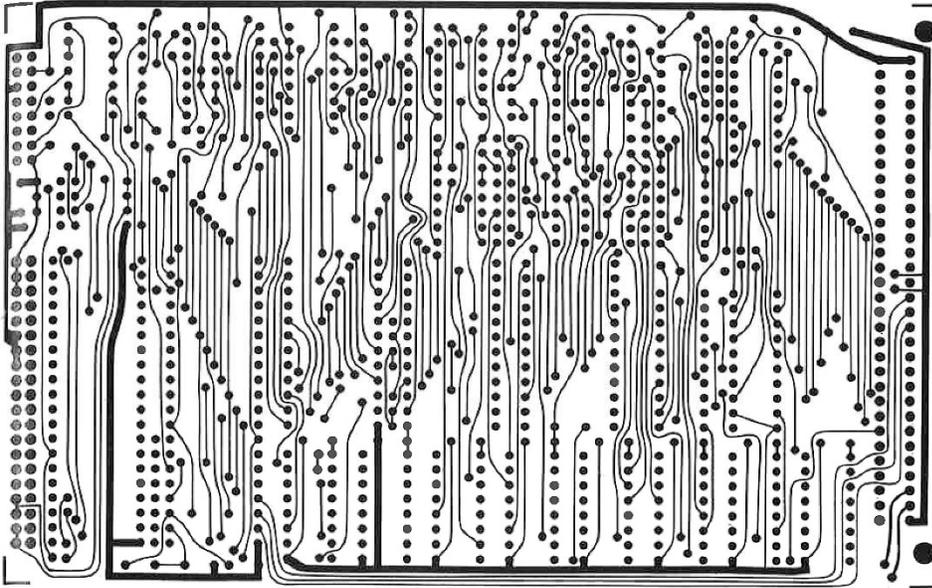


logiciels personnels). La réalisation de la carte CPU ne devrait pas poser de gros problème, pour peu que l'on veuille à effectuer une implantation (et une soudure) correcte des composants. Vous ne serez guère surpris qu'avec une implantation aussi dense, le dessin des pistes ne soit pas du type de celui auquel vous avez habitué Elektor: la dure réalité nous a empêché d'utiliser nos filots de soudure octogonaux. Lors de l'implantation des composants (dont la sérigraphie est illustrée en figure 8), il ne faudra pas oublier d'effectuer les 5 liaisons qui consistent tout simplement à interconnecter deux à deux les points A, B, C, D et E. Côté EPROM, on soudera les straps aux broches corres-

pondantes du support prévu pour ce circuit intégré. La seconde série de points se trouve entre les circuits intégrés IC20 et IC21. Attention, seulement 5 des 6 points présents à cet endroit sont utilisés: si l'on tient la platine en regardant les pistes et que le connecteur à 64 broches pointe vers la droite, les points A...E sont les 5 points les plus bas. Le point situé au-dessus de ces derniers sert uniquement d'interconnexion entre les deux faces du circuit imprimé double face à trous métallisés et il n'est pas question de le doter d'un strap. La solution la plus simple consiste à effectuer ces interconnexions côté pistes. Le désir de concentrer cet ordinateur sur deux cartes seule-



8b



ordinateurs décrits les années précédentes). Il conviendra de respecter ce brochage en cas de réalisation d'extensions ultérieures.

Figure 8a. Représentation de la sérigraphie de l'implantation des composants de la carte CPU/RAM dyn. Elle correspond bien évidemment aux schémas des figures 2...4 et ne comporte pas moins de 29 circuits intégrés (dont 5 "gros") et quelques composants connexes.

Figure 8b. Un exemple de ce que permet la C.A.O. Comme on le constate à l'évidence, le résultat est très différent de ce que nous vous proposons d'habitude dans Elektor!

13

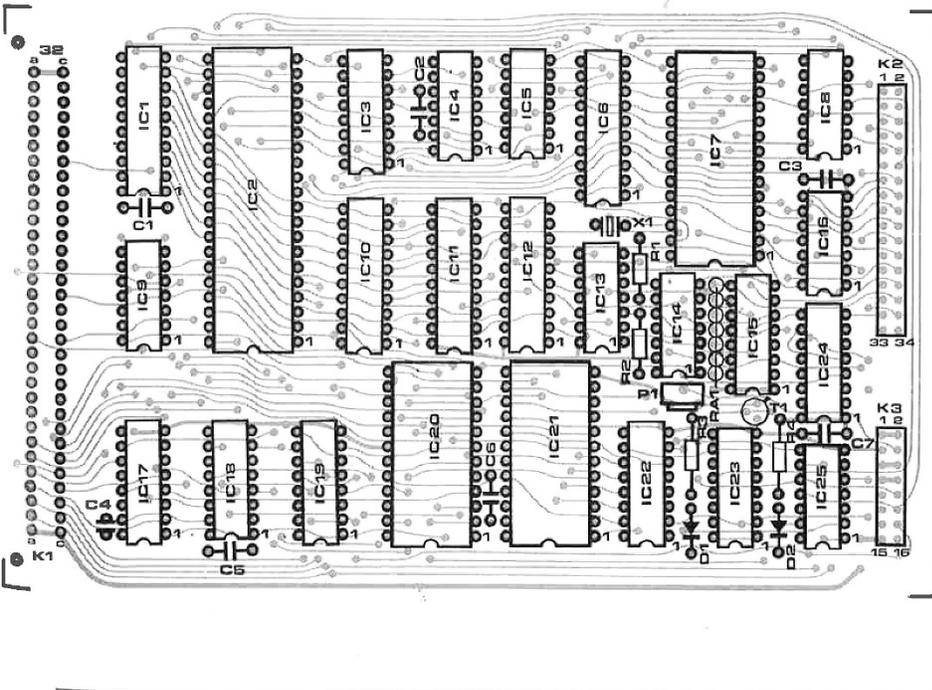


Figure 13. Représentation de la sérigraphie de l'implantation des composants de la carte CRTC/FDC. Cette dernière ne comporte "que" 25 circuits intégrés, ce qui explique que la densité de l'implantation des composants soit notablement moins importante que celle de la platine CPU/RAM dyn.

Figures 9 et 10. A nouveau des schémas "au dessin assisté par ordinateur"; ils donnent respectivement les sous-ensembles vidéo CRTIC (exception faite du 6845 que l'on aurait dû retrouver ici également) et commande de lecteur de disquette FDC (floppy). L'utilisation du contrôleur 1770 a permis d'adopter une densité d'implantation plus aérée que celle racontée sur la carte CPU/RAM dyn

Liste des composants de la carte CPU/DRAM

Résistances:

- R1...R8 = 22 Ω
- R9 = 390 Ω
- R10...R12, R15 = 33 k
- R13 = 4k7
- R14 = 5k6
- Ra1 = 7 x 4k7 (simples ou réseau SIL)

Condensateurs:

- C1, C2 = 27 p
- C3...C11, C16, C18, C20,

- C21 = 100 n
- C12...C14 = 330 p
- C15 = 47 μ/16 V tantale
- C17 = 1 μ/16 V tantale
- C19 = 100 p céramique

Semiconducteurs:

- Z1 = diode zener 3V0/400 mA
- IC1...IC8 = 4164 (200 ns (ou moins) pour version standard
- IC9 = 6809 (Motorola)
- IC10 = 6522
- IC11, IC20 = 74LS244
- IC12 = 74LS245
- IC13, IC14 = 2716

- IC15 = 6850
- IC16...IC19 = 74LS153
- IC21 = 74LS04
- IC22 = 74LS10
- IC23 = 74LS32
- IC24 = 74LS393
- IC25 = 74LS20
- IC26 = 74LS123
- IC27 = 74LS132
- IC28 = 1489 (75189)
- IC29 = 1488 (75188)

Divers:

- X1 = quartz 3,6864 MHz (petit boîtier)
- PL1 = connecteur 64 broches en équerre

- mâle DIN 41612
- PL2 = connecteur mâle type Bergh 40 broches en deux rangées avec détrompeur + connecteur femelle correspondant (par exemple Molex 5342-40-GS1 & 5320-40-BGS1)
- PL3 = idem que K2 avec 16 broches en deux rangées + connecteur femelle correspondant (par exemple Molex 5342-16-GS1 & 5320-16-BGS1)

Liste des composants de la carte CRTIC/FDC

Résistances:

- R1, R2 = 560 Ω
- R3 = 1 k
- R4 = 2k7
- P1 = 470 (500) Ω ajustable à positionnement vertical
- Ra1 = 7 x 220 Ω (individuelles ou en réseau SIL)

Condensateurs:

- C1...C7 = 100 n
- C8
- C9
- C10
- C11
- C12
- C13
- C14
- C15
- C16
- C17
- C18
- C19
- C20
- C21
- C22
- C23
- C24
- C25
- C26
- C27
- C28
- C29
- C30
- C31
- C32
- C33
- C34
- C35
- C36
- C37
- C38
- C39
- C40
- C41
- C42
- C43
- C44
- C45
- C46
- C47
- C48
- C49
- C50
- C51
- C52
- C53
- C54
- C55
- C56
- C57
- C58
- C59
- C60
- C61
- C62
- C63
- C64
- C65
- C66
- C67
- C68
- C69
- C70
- C71
- C72
- C73
- C74
- C75
- C76
- C77
- C78
- C79
- C80
- C81
- C82
- C83
- C84
- C85
- C86
- C87
- C88
- C89
- C90
- C91
- C92
- C93
- C94
- C95
- C96
- C97
- C98
- C99
- C100

Semiconducteurs:

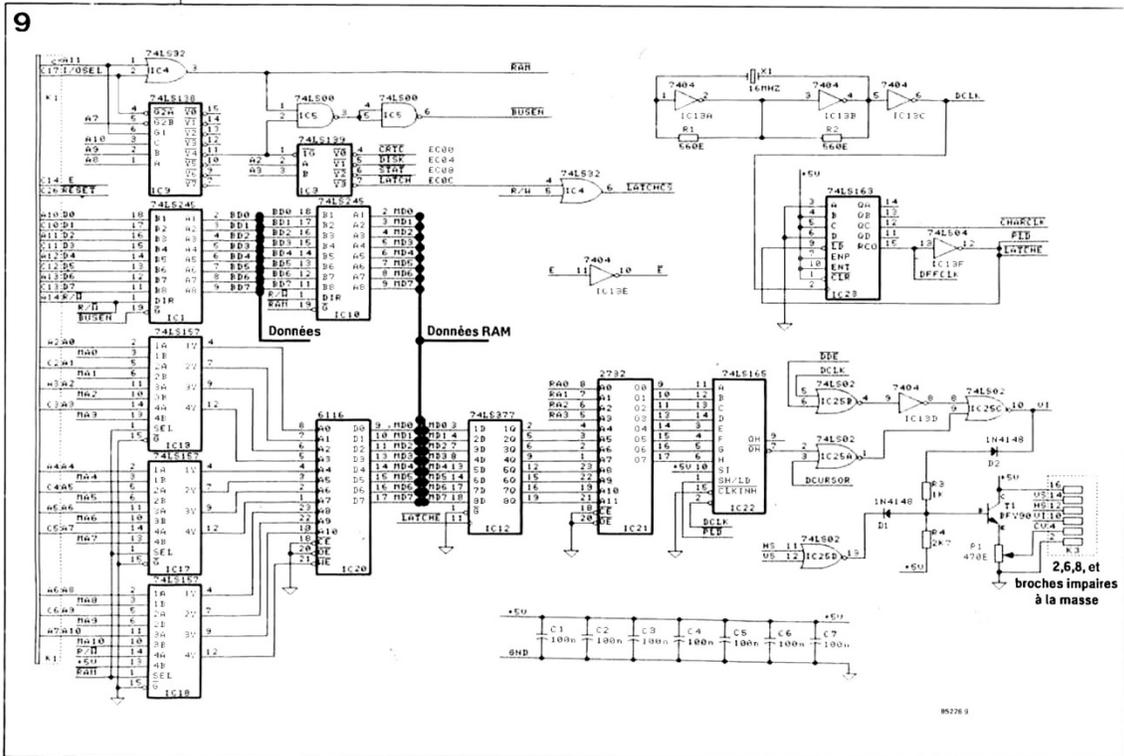
- D1, D2 = 1N4148
- T1 = BFY90
- IC1, IC10 = 74LS245
- IC2 = 6845
- IC3 = 74LS139
- IC4 = 74LS32
- IC5 = 74LS00
- IC6, IC12 = 74LS377
- IC7 = 1770 (Western Digital)
- IC8, IC16 = 74LS06
- IC9 = 74LS138
- IC11 = 74LS244
- IC13 = 74LS04
- IC14 = 74LS74
- IC15 = 4050
- IC17...IC19 = 74LS157

- IC20 = 6116
- IC21 = 2732
- IC22 = 74LS165
- IC23 = 74LS163
- IC24 = 74LS175
- IC25 = 74LS02

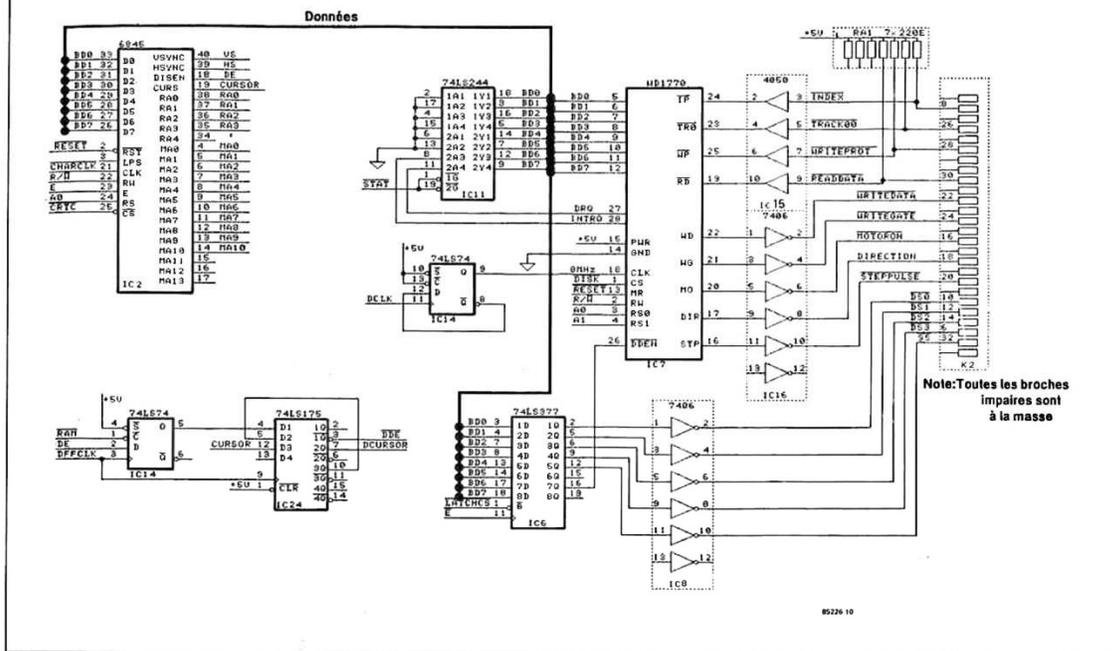
Divers:

- X1 = quartz 16 MHz (petit boîtier)
- K1 = connecteur 64 broches en équerre mâle DIN 41612
- K2 = connecteur mâle type Bergh 34 broches en deux rangées avec détrompeur + connecteur femelle

- correspondant (par exemple Molex 5342-34-GS1 & 5320-34-BGS1)
- K3 = idem que K2 avec 16 broches en deux rangées + connecteur femelle correspondant (par exemple Molex 5342-16-GS1 & 5320-16-BGS1)
- Connecteur femelle pour bus Shugart (floppy 5"1/4) (par exemple Molex 8173-34a)



2,6,8, et broches impaires à la masse



travail d'orfèvre dont nous ne doutons pas un instant qu'il soit parfaitement à votre portée. Nous vous recommandons instamment d'utiliser de bons supports pour les circuits intégrés et avant d'implanter ces derniers, de vérifier la présence des tensions d'alimentation aux broches correspondantes des supports. Ceci fait, on pourra implanter les circuits dans leurs supports respectifs en veillant à ne pas commettre d'erreur de polarité. Pour que l'ordinateur puisse travailler il faut bien évidemment que les EPROM soient programmées avec le bon progiciel (voir à ce sujet le paragraphe "circuits imprimés et progiciel: sources").

Les interfaces vidéo et floppy

Lorsque l'on désire visualiser des caractères sur un écran, il ne suffit pas de disposer d'un CRTIC (Cathode Ray Tube Controller = circuit de commande de tube cathodique). Un coup d'oeil aux figures 9 et 10, le prouve car outre le 6845, CRTIC fameux s'il en est, on découvre plusieurs circuits additionnels au numéro prestigieux. La vidéo est conçue de manière à visualiser 24 lignes de 80 caractères sur l'écran, ce qui ne nous en donne pas moins de $24 \times 80 = 1920$; ceci

explique la présence d'une RAM CMOS du type 6116 constituant une mémoire d'écran (mémoire vidéo) de 2 K (IC20 de la figure 9). Par l'intermédiaire de IC1 et IC10, les caractères arrivent dans la mémoire sous la forme de codes ASCII. L'adressage est pris en compte par IC17...IC19. Le transfert des codes ASCII des caractères dans la RAM d'écran ne suffit pas à en permettre la visualisation cohérente. Il faut que chaque code ASCII soit converti en un ensemble de points reconnaissable comme tel (le caractère). Cette tâche est prise en compte par le générateur de caractères IC21 (une EPROM 2732 de 4 Koctets) associé à IC12 (un 74LS377), circuit faisant office de tampon intermédiaire. Le code ASCII du caractère à visualiser sert pratiquement d'adresse de la case de mémoire de la 2732 dans laquelle est stocké l'ensemble des points qui constituent le caractère concerné. Les octets présents en mode parallèle à la sortie du générateur de caractères sont convertis en mode série (mis à la queue-leu-leu) par IC22 (un 74LS165) avant d'être mélangés à divers signaux et d'être appliqués au transistor T1 qui amène le signal ainsi obtenu à un niveau lui permettant d'attaquer avec succès l'étage d'entrée d'un moniteur. A noter au passage que la fréquence du signal d'horloge de points (dot frequency) est de 16 MHz, fré-

quence générée par l'association des portes IC13A et IC13B et d'un quartz. IC2, le CRTIC prend la part la plus importante dans la génération du signal vidéo; ce circuit coordonne la chronologie de l'ensemble du processus et génère en outre les signaux de synchronisation (horizontale et verticale) pour le moniteur. Le CRTIC est à son tour piloté par la CPU (6809) car n'étant pas un contrôleur intelligent, il est incapable de fonctionner de manière autonome (sans CPU donc). L'unité centrale l'adresse comme n'importe quel autre circuit intégré périphérique (VIA ou ACIA), adressage se faisant par l'intermédiaire de IC3 et IC9, circuits situés en haut à gauche de la figure 9. Contrairement à la partie vidéo que nous venons de décrire, sous-ensemble relativement complexe, l'ensemble de commande d'une (ou plusieurs) unité(s) de lecture de disquettes (floppy drive) ne comporte que fort peu de composants rassemblés autour du contrôleur de lecteur de disquette (floppy disk controller = FDC), IC17 de la figure 10. Ce circuit spécialisé, un 1770 de Western Digital est en mesure d'assurer à lui seul la quasi totalité des fonctions garantissant un fonctionnement impeccable du lecteur de disquettes (ce qui explique peut-être que son prix ne soit pas particulièrement abordable. Disparus les potentiomètres de réglage, car même le sépara-

Figure 11. Le brochage du connecteur du lecteur de disquettes répond au standard Shugart. Il peut de se fait se voir connecter tout lecteur de disquettes standard. Rappel important: ce lecteur doit être à double face (40 ou 80 pistes).

Figure 12. Brochage du connecteur de sortie vidéo. Il est en mesure d'attacher différents types de moniteurs. Si ce dernier possède une entrée Vidéo Composite il suffira de relier la broche CV et l'une de broches de masse à cette entrée par l'intermédiaire d'un morceau de câble coaxial (75 ohms) ou d'une faible longueur de fil blindé flexible. Ce connecteur met également à disposition le signal vidéo inversé et les deux signaux de synchronisation (à l'intention de certains moniteurs hautes performances). Remarque importante: le moniteur utilisé doit avoir une bande passante de 16 MHz au moins, 20 MHz n'est pas du luxe, (si l'on veut qu'il soit en mesure de visualiser l'ensemble des informations sur l'écran).

teur de données est intégré dans le 1770.

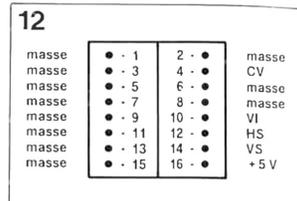
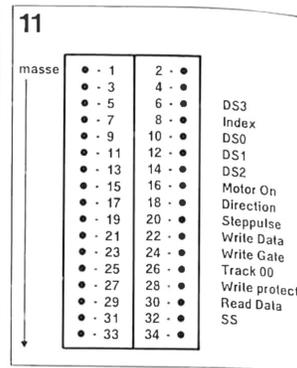
Nous ne pouvons nous permettre d'entrer dans les détails du fonctionnement du FDC, (le lecteur intéressé pourra tenter de se procurer la fiche de caractéristiques auprès du fabricant). L'essentiel est de savoir que IC17 écrit les données en mode série sur la disquette (et les lit de la même manière bien évidemment). IC8, IC9 et IC10 ont deux fonctions: ils tamponnent les signaux de commande et évitent une destruction du 1770 à la suite d'une inversion du connecteur du lecteur de disquettes (ou de tout autre court-circuit des sorties du FDC). Ce danger est en effet latent, sachant qu'un conducteur sur deux du câble de liaison entre l'ordinateur et le lecteur de disquettes est relié à la masse, ceci pour éviter que deux signaux véhiculés par des conducteurs adjacents puissent s'influencer l'un l'autre par l'intermédiaire de la capacité du câble. Pour la même raison, on veillera à ne pas allonger inutilement la longueur du câble de connexion au(x) lecteur(s) de disquettes (ne pas dépasser 40 cm).

Lorsque tous les conducteurs du câble en nappe ont trouvé leur place dans le connecteur (vérifier que le conducteur 1 est bien relié à la broche 1 du connecteur), il ne devrait plus y avoir le moindre problème, car le type de connecteur indiqué dans la liste des composants met à l'abri d'une inversion de polarité. La **figure 11** donne le brochage du (des) connecteur(s) du (des) lecteur(s) de disquettes. Comme on peut le déduire de l'examen de ce dernier, le 1770 est en mesure de contrôler un maximum de quatre lecteurs, les signaux DS0...DS3 (Drive Select) sélectionnant un lecteur chacun. La **figure 12** indique les divers signaux disponibles sur le connecteur vidéo (16 broches). Un morceau de câble coaxial relie ce connecteur au moniteur.

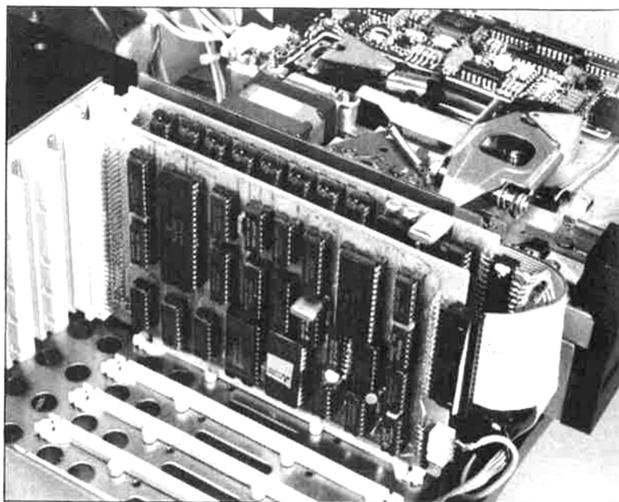
L'implantation des composants sur cette seconde carte étant effectuée (voir liste des composants et sérigraphie de l'implantation de ces derniers, **figure 13**), la partie matérielle de la réalisation du EC-6809 est terminée. A noter cependant que la sérigraphie de l'implantation des composants de cette seconde carte (**figure 13**) comporte une petite erreur: IC9 est un circuit intégré à 16 broches et non pas à 14 comme la sérigraphie de l'implantation des composants pourrait le faire croire.

ware dans la langue de Shakespeare) n'est d'aucune utilité (à la manière d'une voiture sans carburant). On peut subdiviser en deux catégories les programmes disponibles pour le EC-6809: les routines stockées dans les EPROM présentes sur le circuit imprimé (le progiciel proprement dit) d'une part et les programmes stockés sur disquettes (le logiciel) d'autre part. L'une des caractéristiques propres au SED Flex est que la quasi totalité du logiciel est chargé à partir de la disquette, ce qui explique la faible capacité des EPROM utilisées (pas de BASIC de 24 Koctets en ROM ou de DOS de 192 Koctets (!!!) comme pour l'Atari 1040). (Oui mais, diront certains, l'Atari a une mémoire d'1 Moctets!). Contrairement à ce qui se passe avec la majorité des micro-ordinateurs la capacité des EPROM a été réduite au strict minimum. L'avantage de cette solution est une très grande FLEXibilité, la modification d'un logiciel sur disquette est bien plus simple que celle d'un progiciel "grillé" dans une EPROM. Il faut noter en outre, qu'une très faible partie seulement du domaine d'adressage dont est capable le CPU est nécessaire au progiciel d'initialisation, car c'est de cela en fait qu'il s'agit. Ces avantages se paient bien évidemment: la vitesse de fonctionnement diminue légèrement: les choses étant ce qu'elles sont, la recherche de données stockées sur une disquette prend plus de temps qu'un accès à une EPROM.

Nous allons nous intéresser ici au progiciel contenu dans les trois EPROM. Etant donnée sa variété, nous réserverons un article complet au logiciel stocké sur la disquette système. La carte CPU comporte deux EPROM (2 x 2716), la carte CRTC/FDC en comporte une (2732).



Cette dernière n'est pas un progiciel à proprement parler puisqu'il s'agit du générateur de caractères de l'interface vidéo. Pour vous aider au cas où vous voudriez modifier le set de caractères contenu dans le générateur, il existe sur la disquette système Flex un fichier, CGTXT, qui donne les caractères programmés dans cette EPROM de 4 Koctets. Une commande LIST CGTXT permet de faire apparaître ce fichier à l'écran. Il est temps de nous intéresser aux EPROM de progiciel proprement dites, les 2 2716 présentes sur la carte deux CPU et baptisées respectivement "Assist-09" et "Boot-rom". Après mise sous tension, le processeur effectue un saut d'initialisation qui l'amène dans le domaine couvert par un programme moniteur, Assist-09, conçu par le fabricant du 6809



Le progiciel

Un ordinateur sans progiciel (firm-

(Motorola, comme vous le savez sans doute), logiciel mis à la disposition des utilisateurs du processeur en question. Pas de problème de piratage de ce côté-là. Le concepteur du EC-6809 s'est servi du moniteur Assist-09 comme base pour écrire une version plus élaborée adaptée à notre ordinateur Flex et stockée dans IC14. L'adresse de départ est déterminée par le vecteur d'initialisation (Reset vector) défini en logique câblée à l'intérieur du processeur; cette adresse, \$FFFE envoie le processeur en \$F800. Après mise sous tension, vous devriez voir apparaître sur votre écran le texte "ASSIST09" au-dessous duquel on trouve un signe représentant le mode moniteur (>). Assist-09 attend l'entrée d'une commande. A la suite d'une action sur la lettre "F" (pour "FLEX") du clavier suivie d'un retour du chariot (Return ou CR) le système quitte le moniteur Assist-09 pour charger, sous la houlette de "Boot-rom" (EPROM d'initialisation, le fameux Boot), le SED (système d'exploitation des disquettes) présent sur la disquette. Ceci

nous amènerait, et ceci sans passer du coq à l'âne, à parler du logiciel présent sur la disquette système, mais cela n'étant pas le propos de cet article, nous évoquerons quelques-unes des commandes du moniteur. Un "R" visualise à l'écran les contenus des registres du processeur. Un "D" suivi de deux adresses hexadécimales permet de visualiser le contenu d'une case mémoire donnée (définie par l'adresse en question) et de le modifier. Ces commandes qui se situent au niveau langage machine peuvent également être appelées par les logiciels utilitaires (Utilities) présents sur la disquette système. Sous le nom de baptême "Boot-rom", la seconde EPROM regroupe les sous-programmes spécifiques au matériel, routines indispensables à la communication avec le reste de l'ordinateur. Il s'agit principalement des routines vidéo, floppy et d'E/S qui contiennent des adresses spécifiques au système. L'examen des schémas permet déjà de découvrir quelques-unes des adresses les plus importantes de EC-6809. On y voit

par exemple que les données en provenance du clavier entrent dans l'ordinateur par le port du PIA se trouvant à l'adresse \$EF80, tandis que l'interface série (l'ACIA) occupe l'adresse \$EF60 et que l'interface floppy (personnalisée par le FDC) se situe en \$EC04, 5, 6 ou 7. Les diverses routines de commande des organes précités sont documentées très clairement sous la forme de fichiers .TXT présents sur la disquette système, ce qui nous permet de ne pas nous y attarder ici. Si vous désirez en savoir plus à leur sujet, il suffit de les visualiser à l'écran par la commande LIST ...TXT, les ... correspondant bien évidemment au nom du fichier concerné. Le **tableau 1** donne une "cartographie" complète de l'organisation de l'espace mémoire de EC-68, cartographie commençant aux adresses les plus élevées.

Circuits imprimés et logiciel: sources

Tableau 1. Récapitulation des adresses les plus importantes situées dans le domaine d'adressage et cartographie de la mémoire de l'ordinateur Flex EC-6809.

Tableau 1.

Cartographie de la mémoire de EC-6809.

0000 ... BFFF:	Ram utilisateur;
C000 ... DFFF:	Flex;
C000:	adresse de chargement du système d'exploitation Flex;
C000 ... C07F:	Zone de la pile (Stack Area), le pointeur de pile (SP) est initialisé à l'adresse C07F);
C080 ... C0FF:	Tampon d'entrée (Input Buffer);
C100 ... C6FF:	Zone des instructions des routines utilitaires (Utility Command Area);
C700 ... C83F:	agenda (Scheduler) et buffer logiciel pour imprimante (Spooler) (non adaptés à EC 6809);
C840 ... C97F:	FCB système (File Control Block = bloc d'exploitation des fichiers);
C980 ... CBFF:	Zone des fichiers système;
CC00 ... D3FF:	SED (Système d'exploitation des disquettes ou DOS = Disk Operating System);
D400 ... DDFF:	FMS (File Management System = système de traitement des fichiers);
DE00 ... DFFF:	Routines disquettes;
E000 ... E7FF:	Mémoire vidéo (d'écran);
E800 ... EBFF:	E/S utilisateur;
EC00 ... ECFF:	FDC, Vidéo;
EC00:	CRTC;
EC04:	FDC (registres d'état et d'instruction);
EC05:	RFC (registre de piste);
EC06:	FDC (registre de secteur);
EC07:	FDC (registre de données);
EF00 ... EFFF:	ACIA, VIA;
EF60:	ACIA (E/S série);
EF80:	VIA (E/S parallèle);
ED00 ... EFFF:	E/S utilisateur;
E800 ... EFFF:	E/S;
F000 ... F7FF:	Bootrom (EPROM);
F800 ... FFFF:	Assist-09 (EPROM);

Les platines pour EC-6809 devraient être disponibles auprès des revendeurs mentionnés dans la liste Publitronic ou directement auprès de cette société. Les revendeurs mentionnés plus haut devraient être en mesure de vous fournir les trois EPROM programmées: Bootrom (2716) possède le numéro ESS 540, Assist-09 (2716) possède le numéro ESS 541, et le générateur de caractère (2732), le numéro ESS 542.

La disquette système (Boot) comportant le système d'exploitation Flex, divers fichiers d'information + les fichiers TXT évoqués dans l'article et les manuels utilisateur de Flex sont disponibles, auprès de la société:

EDV-Systeme & Peripherie
Dipl.-Ing. Werner Schorstein
Pankratiusstr. 11
6100 Darmstadt
Tél. (06151) 74350

Lors de la commande, ne pas oublier de spécifier 40 ou 80 pistes. En outre, la société C.D.F. (voir en partie annonceurs) s'est attaquée à l'adaptation de son logiciel StarDos (Compatible FLEX) au EC-6809. Se renseigner auprès de ces deux sociétés sur la disponibilité et le prix. ■

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

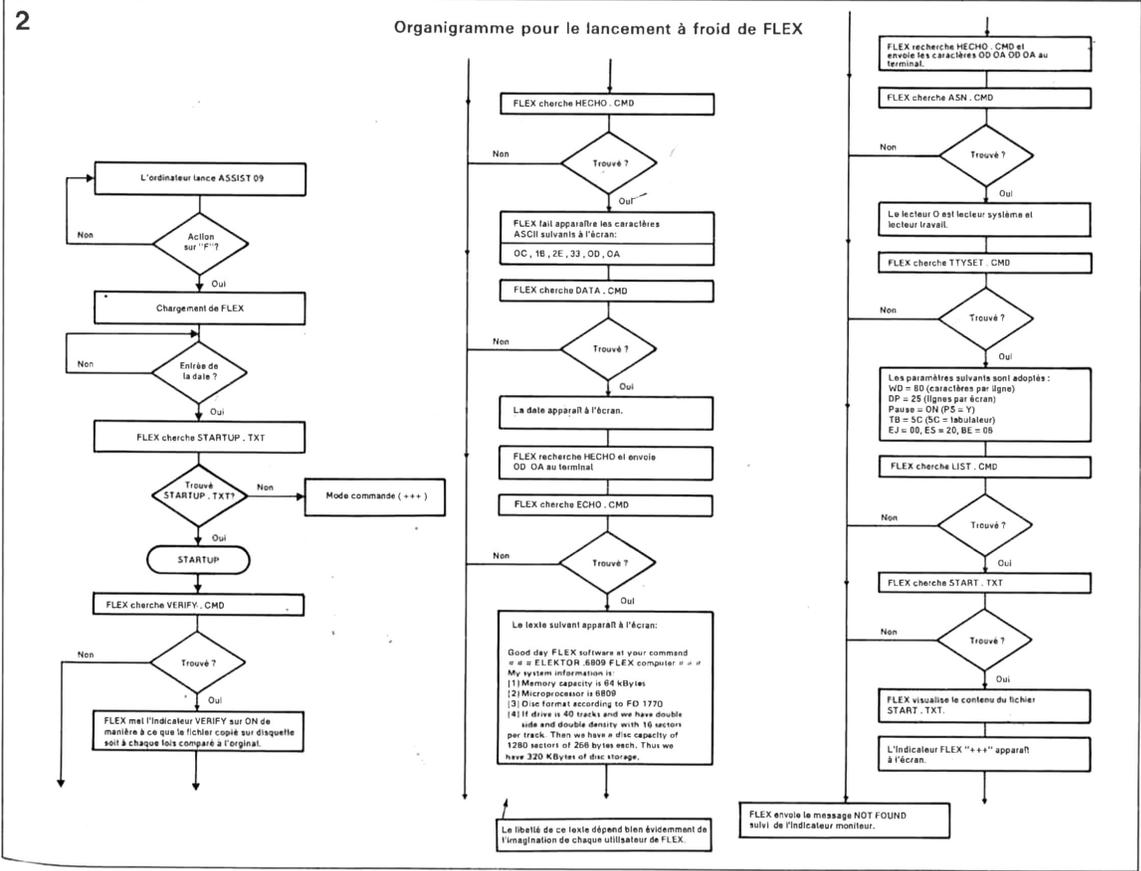
Les divers fichiers nécessaires au démarrage de l'ordinateur, le "boot" de nos amis d'outre-Manche et Atlantique, se trouvent sur la disquette-système. Prenons-les dans un ordre arbitraire.

1

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

```

+++ LIST_STARTUP.TXT
0.VERIFY IS ON
0.HECHO OC
0.HECHO 1B 2E 33
0.HECHO OD OA
0.DATE
0.HECHO OD OA
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 OD OA
0.HECHO OD OA
0.ASN W=0 S=0
0.TTYSET WD=90 DP=25 PS=Y
0.TTYSET TB=5C EJ=00 ES=20 BE=08
0.LIST 0.START1.TXT
+++
    
```



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 STARTUPTXT. Le plus souvent, cette ligne s'écrit EXEC 0.STARTUP.TXT. Pour cette raison, il est nécessaire que la disquette système comporte un fichier "STARTUPTXT", 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.STARTTXT signifie mot à mot: exécute l'une après l'autre les différentes commandes contenues dans le fichier Texte STARTTXT. La figure 1 donne un exemple type d'un tel fichier, la figure 2 donnant quant à elle l'ordonogramme de la chronologie du processus. Ces deux illustrations indiquent très clairement la fonction remplie par le fichier STARTUPTXT 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 STARTUPTXT, 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.

HECHOCMD (pour Hexadécimal **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, abrégé 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.

ECHOCMD 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.

ASNCMD, (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.

TTYSETCMD 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).

LISTCMD 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 DEMOTXT 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 STARTTXT 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

```

+++ nowdisk
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
+++
    
```

raccourcie (cataloguée) dans le premier cas. Une instruction P DIR l'envoie à 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.

BUILDCMD 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 crée. L'une des applications principales de BUILD est la création de fichiers tels que STARTUPTXT que nous avons mentionné plus haut.

COPYCMD 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.

OCMD 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: ODIR. 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 DEMOTXT, 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 DEMOTXT DEMO.BAS transforme d'un coup de baguette magique un fichier .TXT en fichier .BAS (Codé 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 de 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 ASNCMD). 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
+++

```

les utilitaires

Paul Hogenboom

Une "boîte à outils" très complète!

Pour augmenter la puissance et les possibilités de son SED FLEX, Technical Systems Consultants (TSC) propose un certain nombre de programmes auxiliaires (les utilitaires). Parmi ceux-ci se trouvent quelques programmes auxiliaires dont la taille est inversement proportionnelle à leur utilité, à tel point que ce n'est pas sans raison que l'on peut affirmer qu'ils constituent une aide indispensable pour tout utilisateur de FLEX tant soit peu aventureux et sérieux. Etant donné l'espace disponible dans le cadre de cet article, nous ne pourrions nous pencher que sur quelques-uns des plus importants (les goûts et les couleurs se discutent). La documentation fournie avec la disquette comportant ces utilitaires est évidemment bien plus complète que nous ne pouvons l'être ici.

Pour la présentation de chaque programme nous adopterons un format commun qui sera le suivant dans l'ordre: description du programme et de son action, définition de la syntaxe, indication des valeurs ou adjonctions de défaut (la définition de ce terme sera donnée un peu plus loin), exemple, le résultat de l'exécution du programme. La syntaxe a pour fonction de définir très exactement les termes (ou caractères) à utiliser, et leur ordre pour demander l'exécution d'un programme (une instruction complexe en fait). Les termes de défaut peuvent être soit une valeur, soit un texte qu'il n'est pas nécessaire à l'utilisateur de préciser, le SED interprétant cette absence d'une manière bien définie. L'exemple type pour FLEX est l'extension .TXT (pour texte) que le logiciel ajoute "automatiquement" à tout fichier dont l'utilisateur n'a pas défini d'autre extension. Pour le reste de cet article, nous utiliserons dans nos exemples un fichier imaginaire que nous avons baptisé "PARAGR1.TXT".

Venons-en au fait:

FINDCMD

Cet utilitaire passe en revue un fichier texte pour y chercher une chaîne de caractères définie par l'utilisateur et lorsqu'il l'a trouvée, indique à l'écran ou sur imprimante, les lignes dans lesquelles il a trouvé la chaîne considérée. En fait, il s'agit là d'une instruction extrêmement utile lors de traitement de texte (ce type de logiciel étant dans la plupart des cas pourvu d'une telle fonction). Son existence hors programme de traitement de texte est très utile. On pourra utiliser cette instruction lorsqu'il faut rechercher dans un fichier texte (ou certain autres types de fichiers), un caractère, un mot, un chiffre ou même une phrase complète. Faire appel à FINDCMD prend sensiblement moins de temps que de charger un logiciel de traitement de texte de taille standard.

■ Syntaxe: FIND,Nom du fichier, chaîne de caractères à rechercher

■ Défaut(s): Le type du fichier est TXT

■ Exemple:
+++ FIND,PARAGR1,Disquette

■ Résultat: Il y a visualisation de toutes les lignes du fichier PARAGR1 dans lesquelles se trouve le mot "Disquette". On retrouve en outre le numéro d'ordre de chacune de ces lignes.

■ Produit:

```
+++ FIND,PARAGR1,Disquette
(Numéro de ligne)(1ère ligne dans laquelle se trouve "Disquette")
(Numéro de ligne)(2ème ligne dans laquelle se trouve "Disquette")
(etc. . .)
```

```
TOTAL STRING OCCURENCE IS XX
```

Comme l'illustre l'exemple, la liste des apparitions du mot recherché est suivie d'un message en indiquant le nombre d'occurrences.

WORDS

WORDSCMD est un autre utilitaire très pratique. Il ne nous est pas encore arrivé de rencontrer son équivalent dans un logiciel de traitement de texte sous FLEX. WORDS compte le nombre de mots et de lignes composant un texte, la connaissance de ces chiffres pouvant s'avérer extrêmement utile lorsqu'il faut respecter un nombre minimum (ou maximum) de mots (ou de lignes), ce dernier cas étant particulièrement intéressant pour la répartition du texte sur un certain nombre de pages.

■ Syntaxe: WORDS,Nom du fichier

■ Défaut(s): Le type de fichier est TXT

■ Exemple:

+++ WORDS,PARAGR1

■ Résultat: L'utilitaire affiche le nombre de mots et de lignes constituant le fichier (texte) cible: PARAGR1. Un espace séparant deux mots et le caractère "Retour à la ligne/ Curseur à gauche" sont les deux limiteurs de mots reconnus.

■ Produit:

```
+++ WORDS,PARAGR1
TOTAL WORD COUNT IS 1662
TOTAL LINE COUNT IS 144
```

TYPOS

TYPOSCMD est une version de luxe de l'instruction précédente. Elle indique d'une part le nombre de mots que comporte le fichier cible et génère en outre une liste indiquant la fréquence d'apparition de tous les mots constituant le fichier concerné. TYPOS permet ainsi de vérifier la bonne orthographe de tous les mots utilisés, de s'assurer de l'absence de termes techniquement obsolètes ou d'éviter les répétitions trop fréquentes. Comme certains mots (articles, pronoms et autres conjonctions) apparaissent bien plus fréquemment que d'autres, il faut indiquer le nombre maximal de répétition admis. En l'absence d'indication de "fréquence" maximale, le logiciel retient le chiffre "3" comme limite.

■ Syntaxe: TYPOS,Nom du fichier, fréquence maximale

■ Défaut(s): Le type de fichier est TXT; la fréquence de répétition maximale est 3 (en l'absence d'indication de limite, tous les mots présents dans le texte trois fois ou moins

sont indiqués).

■ Exemple: TYPOS,PARAGR1,5

■ Résultat: Une liste de tous les mots présents cinq, quatre, trois, deux ou une fois dans le fichier texte "PARAGR1". L'affichage indique la fréquence devant le mot considéré. Cette liste est classée par fréquence (et par ordre alphabétique dans chaque catégorie de fréquence).

■ Produit: Comme une telle liste est extrêmement longue, nous n'indiquerons pas ici le produit de l'instruction TYPOS. Vous pouvez sans doute vous imaginer ce qu'elle pourrait être.

SPLIT

C'est lorsque l'on travaille avec des textes de longueur importante que SPLITCMD prouve son utilité. Comme l'écran ne peut visualiser plus de 25 lignes, il est aisé de perdre le fil avec ce genre de textes. La division d'un tel texte en plusieurs petits morceaux peut faciliter la tâche de l'utilisateur. L'utilitaire SPLITCMD divise un fichier texte en deux parties, le point de séparation étant défini par l'utilisateur sous la forme d'un numéro de ligne.

■ Syntaxe: SPLIT,Nom du fichier d'origine,Nom du premier nouveau fichier,Nom du second nouveau fichier,Numéro de la ligne où doit avoir lieu la césure.

■ Défaut(s): Le type de fichier est TXT.

■ Exemple:
+++ SPLIT,PARAGR1,PARTIE1,
PARTIE2,100

■ Résultat: Le fichier d'origine PARAGR1 est découpé en deux nouveaux fichiers dont le premier sera baptisé PARTIE1 et comportera les lignes 1 à 99 incluse, le second PARTIE2 comprendra les lignes à compter de 100 jusqu'à la fin. Le fichier d'origine PARAGR1.TXT ne subit pas de modification au cours de cette opération.

LOW-UP, UP-LOW

LOW-UPCMD et UP-LOWCMD sont deux routines qui effectuent la conversion des minuscules en majuscules et inversement. Cette conversion peut être nécessaire par exemple, quand un programme écrit dans un BASIC admettant les deux types de lettres doit être interprété par un autre BASIC qui n'admet que les majuscules.

■ Syntaxe: LOW-UP,Nom du fichier d'origine avec majuscules et minuscules,Nom du nouveau fichier comportant des majuscules exclusivement.

■ Défaut(s): Le type de fichier est TXT.

■ Exemple:
+++ LOW-UP,MINUSC,MAJUSC

■ Résultat: A partir du fichier MINUSC qui comportait des majuscules et des minuscules est généré un fichier MAJUSC composé de majuscules uniquement.

L'exécution de la routine UP-LOW produit bien évidemment le résultat inverse.

DUMP

DUMPCMD visualise, à l'écran ou sur imprimante, le contenu d'un (ou de plusieurs) secteur(s) d'une disquette. Les premiers éléments affichés sont les adresses de la piste et du secteur, le contenu du secteur concerné étant ensuite visualisé à raison de 16 octets par ligne, tant en format hexadécimal qu'en format ASCII. Cette routine est extrêmement précieuse en cas de détérioration d'une disquette que l'on pourra ainsi réparer à condition de posséder un listing sur papier du secteur endommagé.

■ Syntaxe: DUMP,Nom du fichier, Type du fichier

■ Défaut(s): Le type de fichier est BIN (binaire).

■ Exemple:

+++ DUMP,SCRHELP.SYS

■ Résultat: On voit apparaître à l'écran l'adresse de début du fichier SCRHELP.SYS (numéros de la piste et du secteur en format hexadécimal). On aperçoit ensuite le contenu du secteur en 16 lignes de 16 octets (traduits en caractères ASCII sur la seconde partie de l'écran). Pour info, chaque secteur comprend 256 octets. Tous les secteurs faisant partie du fichier à visualiser apparaissent ainsi à l'écran.

■ Produit: Voir figure 1

CHECK

CHECKCMD compare les contenus de deux fichiers. Lorsque cette opération est terminée, le résultat de la comparaison est donné sous la forme de l'un des deux messages

suivants:

THE FILES CHECKED DO NOT MATCH ou
THE FILES CHECKED ARE IDENTICAL.

Cette routine est très pratique lorsque l'on se trouve en présence de deux fichiers de noms différents dont on suppose que l'un est la copie de l'autre.

■ Syntaxe: CHECK,Nom du fichier, Nom du fichier2

■ Défaut(s): Le type du fichier est TXT.

■ Exemple:

+++CHECK,PARAGRI.TXT,PARAGRI.BAK

■ Résultat: Le contenu du fichier PARAGRI est comparé à son fichier de sauvegarde (BAK = Back-up).

■ Produit:

+++CHECK,PARAGRI.TXT,PARAGRI.BAK

THE FILES CHECKED ARE IDENTICAL

MAP

MAPCMD est un programme permettant de retrouver les adresses de chargement et les domaines occupés par les fichiers de commande. Il indique à quelle adresse est implanté un fichier commande et à quelle adresse prend place son exécution, (cette dernière adresse est également appelée adresse de transfert). L'utilisation de MAP se justifie tout-particulièrement de concert avec l'instruction SAVE qui permet d'écrire sur disquette, sous n'importe quel nom, le contenu d'un bloc de mémoire choisi.

■ Syntaxe: MAP,Nom du fichier

■ Défaut(s): Le type du fichier est BIN.

■ Exemple:

+++MAP,MAPCMD

■ Résultat: On voit apparaître à l'écran les adresses de début et de fin du domaine mémoire dans lequel est chargé MAPCMD, suivies de l'adresse de transfert

■ Produit:

+++MAP,MAPCMD

C100-C126

C100-C1E8

C100

FREE

FREECMD indique la place disponible sur la disquette, place donnée en nombre de secteurs et en Koctets.

■ Syntaxe: FREE X

(X = numéro d'ordre du lecteur concerné)

■ Défaut(s): Le lecteur est le lecteur 0.

■ Exemple:

+++FREE

■ Résultat: On voit apparaître à l'écran le message suivant:
SECTORS REMAINING = X
APPROXIMATE KILOBYTES = YY

■ Produit:

+++FREE
SECTORS REMAINING = 243
APPROXIMATE KILOBYTES = 61

TEST

TESTCMD permet d'épargner de gros déboires à l'utilisateur de disquettes de moindre qualité. Tous les secteurs de la disquette sont passés en revue pour vérifier leur intégrité; si la disquette comporte de mauvais secteurs, le message BAD SECTOR est affiché à l'écran avec le numéro hexadécimal du (ou des) secteur(s) concerné(s).

■ Syntaxe: TEST,Numéro du lecteur

■ Défaut(s): Le numéro est celui du lecteur défini comme lecteur de travail par l'instruction ASN (voir l'article sur le SED FLEX).

■ Exemple: TEST,0

■ Résultat: La disquette présente dans le lecteur 0 subit un test rigoureux; la détection d'une erreur est signalée par le message BAD SECTOR AT... Si l'on demande de

```
03 20
03 21 00 01 43 6F 6D 6D 61 6E 64 73 20 69 6E 20 _!__Commands in
74 68 65 20 63 6F 6D 6D 61 6E 64 20 6D 6F 64 65 the command mode
3A 0D 0D 50 41 47 45 3D 4E 09 0B 53 65 74 20 6E :__PAGE=N__Set n
65 77 20 70 61 67 65 6C 65 6E 67 74 68 2E 0D 4C ew pagelength_L
49 4E 45 3D 4E 09 0B 53 65 74 20 6E 65 77 20 6C INE=N__Set new l
69 6E 65 6C 65 6E 67 74 68 2E 0D 53 54 41 54 55 inelength__STATU
53 3D 54 52 55 45 09 06 45 6E 61 62 6C 65 20 73 S=TRUE__Enable s
74 61 74 75 73 20 6C 69 6E 65 20 69 6E 20 74 65 tatus line in te
78 74 20 6D 6F 64 65 2E 0D 53 54 41 54 55 53 3D xt mode__STATUS=
46 41 4C 53 45 09 05 44 69 73 61 62 6C 65 20 73 FALSE_Disable s
74 61 74 75 73 20 6C 69 6E 65 2E 0D 49 4E 53 45 tatus line__INSE
52 54 3D 54 52 55 45 09 08 45 6E 61 62 6C 65 20 RT=TRUE__Enable
61 75 74 6F 20 66 69 65 6C 64 69 6E 67 2E 0D 49 auto fielding__I
4E 53 45 52 54 3D 46 41 4C 53 45 09 05 44 69 73 NSERT=FALSE__Dis
61 62 6C 65 20 61 75 74 6F 20 66 69 65 6C 64 69 able auto fieldi
6E 67 2E 0D 46 49 45 4C 44 3D 4E 09 0A 44 65 66 ng.__FIELD=N__Def
```

```
03 21
03 22 00 02 69 6E 65 20 66 69 65 6C 64 20 63 68 _"__ine field ch
61 72 61 63 74 65 72 20 61 73 20 64 65 63 69 6D aracter as decim
61 6C 20 6E 75 6D 62 65 72 2E 0D 46 49 45 4C 44 al number__FIELD
3D 27 43 09 09 44 65 66 69 6E 65 20 66 69 65 6C ='C__Define fiel
64 20 63 68 61 72 61 63 74 65 72 20 61 73 20 41 d character as A
53 43 49 49 20 63 68 61 72 2E 0D 43 4F 4D 4D 45 SCII char__COMME
4E 54 3D 2A 09 08 44 65 66 69 6E 65 20 63 6F 6D NT=*__Define com
6D 65 6E 74 20 63 68 61 72 61 63 74 65 72 2E 0D ment character__
57 49 4C 44 3D 3F 09 0B 44 65 66 69 6E 65 20 77 WILD=?__Define w
69 6C 64 20 73 65 61 72 63 68 20 63 68 61 72 61 ild search chara
63 74 65 72 2E 0D 46 4F 4C 44 3D 54 52 55 45 09 cter__FOLD=TRUE_
08 53 65 74 20 75 70 70 65 72 63 61 73 65 20 6F _Set uppercase o
6E 6C 79 2E 0D 46 4F 4C 44 3D 48 41 4C 53 45 09 nly__FOLD=FALSE_
07 55 70 70 65 72 2F 6C 6F 77 65 72 63 61 73 65 _Upper/lowercase
2E 0D 53 45 41 52 43 48 46 4F 4C 44 3D 54 52 55 _SEARCHFOLD=TRU
45 09 02 53 65 61 72 63 68 20 75 70 70 65 72 20 E__Search upper
```

```
03 22
03 23 00 03 6F 72 20 6C 6F 77 65 72 63 61 73 65 _£__or lowercase
2E 0D 53 45 41 52 43 48 46 4F 4C 44 3D 46 41 4C _SEARCHFOLD=FAL
53 45 20 53 65 61 72 63 68 20 66 6F 72 20 65 78 SE Search for ex
61 63 74 20 6D 61 74 63 68 2E 0D 41 55 54 4F 3D act match__AUTO=
54 52 55 45 09 08 41 75 74 6F 20 6E 75 6D 62 65 TRUE__Auto numbe
72 69 6E 67 20 6F 6E 2E 0D 41 55 54 4F 3D 46 41 ring on__AUTO=FA
4C 53 45 09 07 41 75 74 6F 20 6E 75 6D 62 65 72 LSE__Auto number
69 6E 87 20 6F 66 66 2E 5C 0D 0D 4F 6E 65 20 6C ing off.\__One l
65 74 74 65 72 20 63 6F 6D 6D 61 6E 64 73 3A 0D etter commands:_
0D 41 20 6F 72 20 51 09 0B 41 62 6F 72 74 20 6F _A or Q__Abort o
72 20 51 75 69 74 2E 20 4E 6F 20 73 61 76 65 2E r Quit. No save.
0D 53 09 10 53 61 76 65 20 61 6C 6C 20 74 65 78 _S__Save all tex
74 20 61 6E 64 20 72 65 74 75 72 6E 20 74 6F 20 t and return to
46 4C 45 58 2E 0D 47 09 10 47 6F 20 65 78 65 63 FLEX__G__Go exec
75 74 65 20 70 72 65 64 65 66 69 6E 65 64 20 47 ute predefined G
4F 20 63 6F 6D 6D 61 6E 64 2E 0D 4E 5B 6D 5D 09 O command__N[m]_
```

tester une disquette dans un lecteur n'existant pas, on aura affichage du message suivant: ILLEGAL DRIVE NUMBER.

■ Produit:
+++TEST,0
BAD SECTOR AT NN-MM
BAD SECTOR AT XX-YY
etc...

TEST COMPLETED

soit:

+++TEST,0
TEST COMPLETED

C'est bien évidemment ce second message que l'on souhaite voir s'afficher à l'écran: une disquette parfaite.

RPT

RPTCMD simplifie la tâche lorsqu'il faut qu'un programme soit exécuté à plusieurs reprises. Normalement, il faudrait à chaque fois entrer à nouveau la commande à exécuter. Cela est bien évidemment possible, mais l'efficacité en prend un coup (à quoi cela sert-il d'avoir un ordinateur?). RPT (de repeat, vous l'avez deviné) permet de faire exécuter une commande le nombre de fois que l'on veut.

■ Syntaxe: RPT,Nombre d'exécutions désiré,commande à exécuter

■ Défaut(s): sans objet

■ Exemple: RPT,3,P LIST PARAGR1.TXT

■ Résultat: Le contenu du fichier PARAGR1 est imprimé en triple exemplaire sur l'imprimante.

PDEL

PDELCMD est l'abréviation de "prompting delete", effacement par acquiescement en quelque sorte. Dans le cas de la commande d'effacement DELETE standard, il faut indiquer le nom du fichier à effacer. Ici les choses se passent différemment. Sur l'écran apparaissent les noms de tous les fichiers présents dans le répertoire de la disquette dans l'ordre correspondant à ce celui de ce dernier. Le nom de chaque fichier est suivi par un point d'interrogation auquel on peut répondre soit par Y (yes), N (no) ou

Retour à la ligne.

Un retour à la ligne fait sortir du programme PDEL; le système revient en mode commande (+++). Important: avant de procéder à l'effacement, PDEL ne pose pas la question de confiance "ARE YOU SURE" (contrairement à ce que fait DELETECMD), ce qui signifie qu'une action sur Y provoque une action immédiate! Il est possible de limiter à un certain type de fichier les fichiers apparaissant à l'écran (TXT par exemple). Il est possible de cette manière de filtrer, par combinaisons de lettres interposées les fichiers à effacer.

■ Syntaxe: PDEL,Numéro de lecteur(critères de sélection)

■ Défaut(s): sans objet

■ Exemple: PDEL,0,TXT

■ Résultat: Les noms de tous les fichiers texte de la disquette présente dans le lecteur 0 apparaissent l'un à la suite de l'autre à l'écran. Après chaque nom, il faut entrer soit un Y, soit un N, soit un Retour à la ligne. Après une action sur Y ou N, le curseur passe au fichier suivant, après un Retour à la ligne ou le dernier fichier, le système revient en mode commande (+++).

MEMDUMP

MEMDUMPCMD génère l'apparition à l'écran en blocs de 256 octets du contenu de la mémoire sous format hexadécimal. A la suite du premier bloc on voit apparaître trois lettres donnant le choix entre bloc suivant (F = forward), bloc précédent (B = backward) et sortie du programme (Retour à la ligne). Dans le premier cas, on voit apparaître le bloc de 256 octets suivant, dans le second le bloc précédent et dans le dernier cas on retourne en mode commande (+++).

■ Syntaxe: MEMDUMP,Adresse de début en hexadécimal

■ Défaut(s): En l'absence d'adresse de début, le programme suppose qu'il s'agit de 0000.

■ Exemple: MEMDUMP,F000

■ Résultat: Sur l'écran apparaît en format hexadécimal le contenu de la mémoire du domaine compris entre F000 et F0FF.

MEMTEST

MEMTESTCMD est un programme

permettant, comme l'indique d'ailleurs son nom, de tester le fonctionnement de la mémoire, une fonction extrêmement utile lorsque l'on a des doutes à ce sujet.

■ Syntaxe: MEMTEST,Adresse de début,Adresse de fin.

■ Défaut(s): sans objet

■ Exemple: MEMTEST,0,01FF

■ Résultat: Le système écrit un nombre aléatoire dans chacune des cellules de mémoire du domaine compris entre 0000 et 01FF; il procède ensuite à la lecture de ces nombres et les compare au nombre d'origine, cette procédure se répétant un certain nombre de fois. Le test d'un bloc de 4 Koctets dure quelque 60 minutes. Il n'est bien évidemment pas possible de tester le domaine mémoire dans lequel réside le programme MEMTEST sous peine de "planter" le système. Pour éviter ce problème, on pourra, à l'aide du programme MAP, localiser sans peine l'endroit où réside MEMTEST. A la fin du processus de test, si tout s'est bien passé, l'écran affiche un "I" victorieux. La seule solution pour interrompre l'exécution de MEMTEST est d'actionner la touche RESET.

Comme nous le signalions en début d'article, nous ne disposons pas de suffisamment d'espace pour passer en revue tous les programmes utilitaires de FLEX, surtout qu'il en naît des nouveaux chaque jour. Pour terminer cet article nous indiquons dans l'ordre alphabétique les différents utilitaires décrit dans cet article:

CHECK
DUMP
FIND
FREE
LOW-UP
MAP
MEMDUMP
MEMTEST
PDEL
RPT
SPLIT
TEST
UP-LOW
WORDS

Plusieurs de nos annonceurs proposent de nombreux logiciels tournant sous Flex. Voir entre autres aux pages 102 et 103.