

Quincaillerie

Pour rendre compatible le Junior Computer et le système d'exploitation de disques souples, il a fallu faire quelques aménagements. Rassurez-vous! Il n'y a pas la moindre piste à couper et l'intervention reste bénigne. L'un des circuits intégrés de décodage d'adresses est remplacé par un autre. Il y a d'autre part quelques composants supplémentaires pour l'interface pour imprimante EPSON (ligne BUSY); il nous a paru raisonnable de nous contenter d'un montage "volant", plutôt que de rajouter encore un (micro) circuit supplémentaire. Ceci dit, rien n'empêchera le lecteur de procéder à sa façon...

G. de Cuyper

Sur la carte d'interface, le décodage est le suivant:

VIA (IC1) du type 6522:

\$F800...\$F9FF (Paperware!)

RAM (IC2 et IC3) du type 2114:

\$E400...\$E7FF

EPROM (IC4 et IC5) du type 2716:

\$E800...\$F7FF

La deuxième modification consiste à aménager une interface spéciale pour la ligne BUSY d'une imprimante EPSON. La figure 2 en illustre le schéma. Le relais Re1 pourra être omis, et la LED D4 servira d'indicateur "BUSY".

Les figures 3 et 4 indiquent comment nous nous y sommes pris pour effectuer ces modifications.

interface pour unités à disques souples

du logiciel pour le Junior Computer et autres systèmes à 6502

Au cours de ce second (et dernier) article consacré à l'interface pour unités à disquettes du Junior Computer, nous indiquons quelles sont les (petites) modifications à effectuer sur le système actuel du Junior Computer pour le rendre compatible avec le logiciel d'Ohio Scientific. Il faut notamment une nouvelle EPROM, qui lors de l'initialisation de la machine, assure le chargement du programme d'exploitation sur la disquette. Le logiciel de ce programme moniteur est disponible en "Paperware" d'Elektor.

2ème partie

Replongeons nous un instant dans le schéma du circuit d'interface du Junior Computer reproduit sur la figure 1. Les portes N33/N34 sont remplacées par une porte NAND: la ligne 8K0 (ou EX) n'est plus active pour l'espace mémoire \$0000...\$1FFF, mais de \$E000...\$FFFF. Ce qui va avoir les conséquences suivantes sur le décodage d'adresses:

— de \$0000 à \$BFFF, on aura 48 K de RAM dynamique!

Il suffit de trois cartes de mémoire dynamique comme celle que nous avons publiée en Avril 1982, Elektor n° 46, page 4-48, pour couvrir cet espace mémoire. La consommation électrique reste modérée... le prix aussi!

— le décodeur d'adresses de la carte principale du Junior Computer (IC6) décode l'espace mémoire compris entre \$E000 et \$FFFF. L'adressage sur cette carte sera donc le suivant:

EPROM (IC2) du type 2708:

\$FC00...\$FFFF

PIA, RAM, temporisateur du type 6532:

\$FA00...\$FBFF (Paperware!)

RAM (IC4 et IC5) du type 2114:

\$E000...\$E3FF

Une fois que tout est en place (et dûment vérifié, il reste à mettre la nouvelle EPROM du type 2708 (ESS 515) à la place de l'EPROM originale de la carte principale du Junior Computer. Les EPROM de la carte d'interface (PM et TM) ne sont plus nécessaires à présent, puisque les routines d'entrée et sortie pour terminal vidéo et imprimante sont logées dans la nouvelle EPROM 2708 de la carte principale. Par conséquent, IC4 et IC5 pourront être mis à contribution pour des programmes résidents spécifiques à l'utilisateur.

Ne pas omettre la liaison entre les points R et S sur la carte d'interface (WITH).

Ainsi fait! Le Junior Computer est prêt maintenant pour recevoir l'introduction... Si vous ne disposez que de deux cartes de RAM dynamique (soit 32K), ce n'est pas dramatique... mais le système n'est vraiment complet qu'avec 48 K.

Le décodage d'adresse des cartes de mémoire vive est le suivant:

carte RAM 1: U - 0

V - 1

X - 2

Y - 3

carte RAM 2: U - 4

V - 5

X - 6

Y - 7

carte RAM 3: U - 8

V - 9

X - A

Y - B

Avant de mettre l'interface pour unités à disques souples sur le bus du Junior Computer, mettez-le sous tension et actionnez la touche "RST" du clavier

1

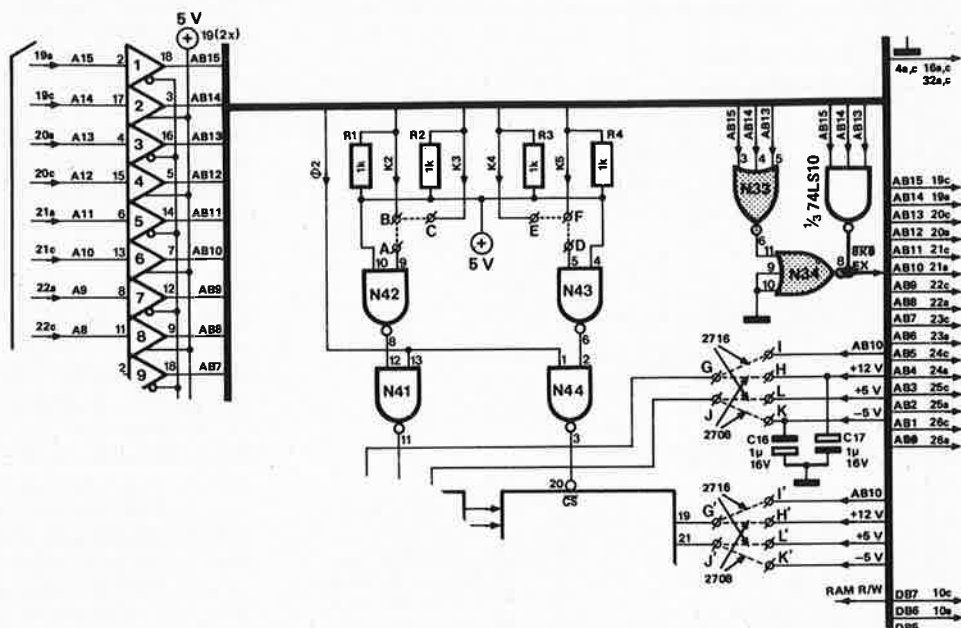


Figure 1. Le schéma de la carte d'interface du Junior Computer a été modifié: les portes N33 et N34 sont remplacées par une porte NAND 74LS10 afin de modifier le décodage d'adresses.

hexadécimal de la carte principale. Les afficheurs s'illuminent, les fonctions des touches "AD", "DA", "+" et "GO" sont restées les mêmes. Seule la touche "PC" a changé... mais nous reviendrons là-dessus lorsque le moment sera venu.

Le tableau 1 vous permettra de naviguer à travers la mémoire du Junior-DOS ou de tout autre système à 6502 associé à la carte d'interface pour disques souples d'Elektor publiée le mois dernier. Les adresses \$C000...\$FBFF peuvent être organisées différemment selon les systèmes; l'essentiel reste que la première moitié de la zone mémoire consiste en (au moins) 32 K de RAM sans discontinuité. Si l'espace mémoire de \$FC00 à \$FFFF devait être indisponible sur l'un ou l'autre système, il faudra traduire le programme nommé "Bootstrap". Ceci ne devrait poser aucun problème grâce aux listings fournis en "Paperware".

Le logiciel du Junior-DOS

L'orientation du logiciel du Junior-DOS est celle des systèmes les plus avancés: à savoir, le moins possible de mémoire morte résidente et un maximum de mémoire vive. Les avantages de ce choix sont évidents:

Il suffit de fractions de secondes pour le chargement de programmes (BASIC, FORTH, Assembleur, Traitement de textes, etc) depuis les disquettes. Pourquoi s'embarasserait-on d'encombrantes ROM/EPROM résidentes?

Il suffit d'un programme "minimal" permettant la gestion de l'affichage et du clavier hexadécimal, de l'interface pour l'Elektterminal, et... l'amorçage du système d'exploitation des unités à disques souples, bien sûr! Il s'agit en fait de charger en mémoire la piste 0 de la disquette.

1

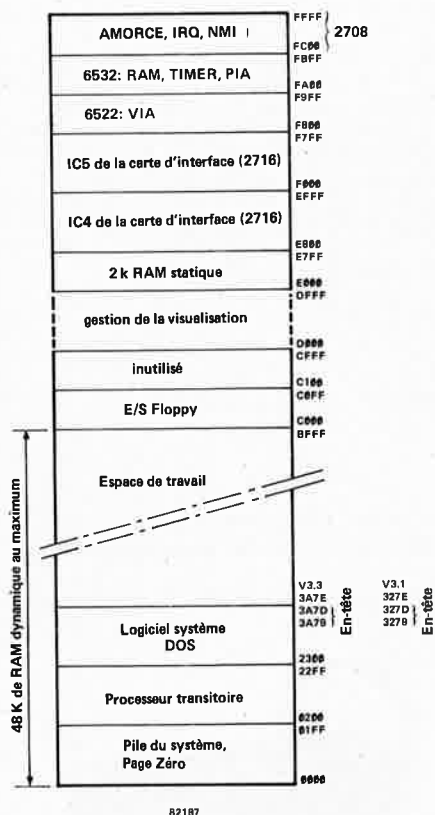


Tableau 1. La cartographie de la mémoire du Junior Computer est quelque peu bouleversée. On notera plus particulièrement que le processeur transitoire surcharge une partie du DOS: il s'agit de la zone qui s'étend de \$2200 à \$22FF.

Le grand avantage des disques souples est donc de nous offrir un confort d'utilisation inconnu jusqu'ici; ce qui se traduit par une rapidité et une souplesse étonnantes.

Mais revenons au tableau 1 et à la cartographie de la mémoire de notre machine. La page 0 et la pile restent en \$0000...\$01FF, on pourrait difficilement faire autre chose! De \$0200... à \$2FFF on trouve une zone mémoire attribuée au "TRANSIENT PROCESSOR"; il s'agit de ce que l'on pourrait appeler le "PROCESSEUR TRANSITOIRE" en français, c'est à dire le programme avec lequel la machine est en train de fonctionner. Vous travaillez en BASIC? Alors le processeur transitoire est l'interpréteur BASIC... Vous êtes en train d'assembler un programme en langage machine? C'est l'Assembleur qui est le processeur transitoire. Ou peut-être rédigez-vous une lettre pour féliciter la rédaction d'Elektor de l'excellente tenue de son magazine? Le processeur transitoire est le programme de traitement de textes! Et ainsi de suite.

Le programme d'exploitation des unités à disques et de gestion de l'imprimante, ainsi que le "Memory Mapped Video Display Unit" c'est à dire une zone mémoire attribuée à l'écran de visualisation et son contenu, occupent environ 4 K de mémoire vive. Dans la version OS-65D V3.1, le logiciel d'exploitation de disques magnétiques s'étend de \$2300 à \$3278. C'est à l'adresse \$327E que commence la mémoire de travail, c'est à dire la zone attribuée aux programmes de l'utilisateur. Lorsque vous mémorisez un programme en BASIC, c'est en \$327E qu'il sera placé. Les données écrites sur disquette sont lues à partir de cette même adresse. Inversement, les données lues sur dis-

4

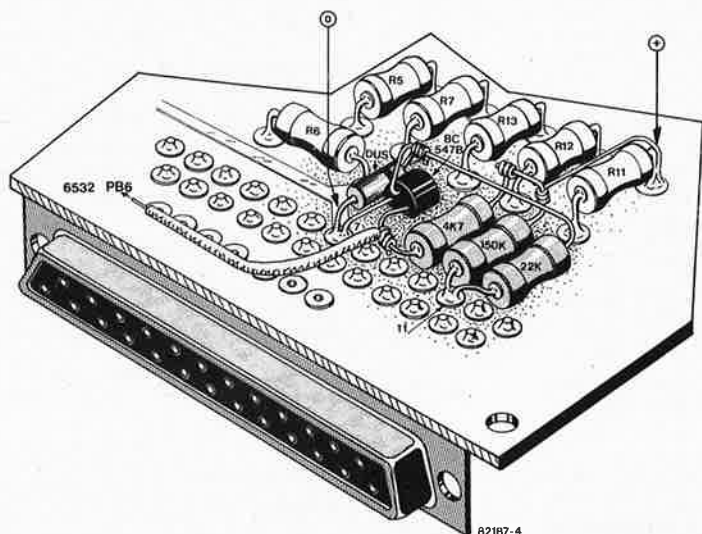


Figure 4. Croquis illustrant la mise en place du montage volant appelé "interface EPSON" sur la carte d'interface du Junior Computer.

du clavier de l'Elekterminal).

Une fois que l'interpréteur BASIC est chargé en mémoire et que l'ordinateur l'a signalé sur l'écran (ou l'imprimante), le processeur transitoire peut être interrompu à tout moment. Pendant une opération d'impression, on peut actionner la touche (BREAK). S'il s'agissait d'une opération de listage d'un fichier, le processeur imprime le message "Break". S'il s'agissait de l'exécution d'un programme (déclenchée par l'instruction RUN), l'interpréteur indique "BREAK IN LINE X". Toutes les variables du programme BASIC ainsi que les pointeurs sont sauvegardés sur la pile. Après une interruption d'exécution, on peut relancer le programme avec l'instruction CONT. Le vecteur-saut indirect pour la fonction BREAK est positionné automatiquement lors du lancement du BASIC via l'adresse \$FF17. Le vecteur BREAK se trouve aux adresses \$FA7C et \$FA7D.

3. Chargement de programmes associés au logiciel d'exploitation de disques magnétiques (nous continuerons de l'appeler DOS... c'est plus commode!). La séquence suivante:

(AD) FF34 (GO) (RUBOUT)

permet de charger des programmes comme ceux que nous proposerons dans l'avenir pour le Junior Computer. L'adresse RESDOS ne convient donc pas pour le logiciel d'Ohio Scientific.

4. Adaptation d'une disquette Ohio Scientific OS-65D V3.1.

La séquence suivante:

(AD) FFE2 (GO) (RUBOUT)

permet d'adapter le logiciel OS-65D V3.1 au Junior Computer. Lorsque l'on exécute le programme à partir de l'adresse VONE en suivant la procédure ci-dessus, il se produit les événements suivants:

- Le processeur place la tête de l'unité au-dessus de la piste 0.
- Le processeur charge les données de la piste 0 et les range à l'adresse \$2200 et suivantes en mémoire vive.
- Le processeur met la tête de lecture sur la piste 1 et en charge les données qu'il place à l'adresse \$2A00 et suivantes. Ainsi la mémoire contient la totalité du logiciel d'exploitation de la version 3.1. C'est à l'utilisateur de procéder aux modifications qui lui permettront de rendre ce programme compatible avec le Junior Computer (à l'aide du clavier hexadécimal).
- Une fois que le contenu des pistes 0 et 1 a été chargé, le processeur revient au moniteur hexadécimal après avoir

émis le message *Track 0&1*.

5. Adaptation du logiciel de la disquette OS-65D V3.3 Tutorial Disk 5. Pour le chargement des pistes 0 et 1 de cette disquette, tout se passe comme sous 4, à ceci près que l'adresse de lancement du programme de chargement est VTHREE*: \$FFE8.

Une fois que les deux premières pistes sont chargées, les événements se déroulent comme indiqué ci-après:

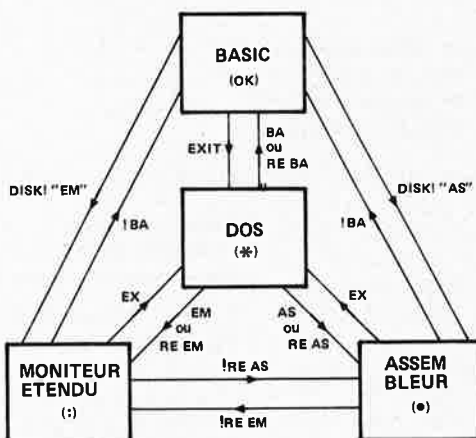
- Le processeur place la tête sur la piste 6 et charge les données du secteur 2, et les range à partir de l'adresse \$3200.
- Le processeur charge les données de la piste 6/secteur 3 et les place à l'adresse \$0000 et suivantes (le secteur 3 comporte 6 pages).
- Le processeur place la tête sur la piste 13 dont il charge le secteur 1: l'adresse de destination des données est \$3A79. Ce secteur compte 8 pages. Une fois que toutes ces opérations ont été effectuées, le processeur émet le message *TRACK 0&1* et retourne au moniteur hexadécimal.

6. La touche "PC". Sur le clavier hexadécimal, la touche "PC" permet de quitter le moniteur hexadécimal pour accéder à l'interpréteur de directives du DOS. La configuration d'entrée/sortie pour le terminal est réinitialisée, mais on n'effectue pas de mesure de la durée de bit (pour déterminer le taux de transmission). Le processeur émet le message A* ou B*.

7. La routine de réception de données émises par le terminal. Cette routine reçoit le caractère émis par le terminal et assure la mise à zéro du bit 7. Le contenu des registres d'index n'est pas altéré, c'est l'accumulateur qui contient le caractère lorsque l'on quitte cette routine. L'adresse de début est:

RECCHA*: \$FE1B.

5



Transfert des Directives de Commande 82187 - 5

Figure 5. Ce schéma illustre le transfert de contrôle d'un processeur transitoire à l'autre en passant par "l'organe central" qu'est le DOS.

8. La routine d'émission de données vers le terminal. Cette routine assure l'émission sérielle du caractère contenu dans l'accumulateur au moment où il y est fait appel.

Le format utilisé est le suivant:

- un bit de départ
- sept bits de données
- pas de bit de parité
- deux bits de fin.

Le contenu des registres d'index n'est pas altéré par cette routine dont l'adresse de début est:

PRCHA* \$FEA3.

Comment fonctionne l'amorçage?

Maintenant que nous connaissons les parties constituant l'EPROM d'amorçage, il nous reste à en élucider le fonctionnement.

Après la séquence suivante:

```
(RST)
(AD) FF17
(GO)
```

il se produit un certain nombre d'événements que nous allons détailler:

1. Le processeur positionne le vecteur-saut indirect pour la fonction BREAK du terminal. Le vecteur NMI conduit au moniteur hexadécimal.

2. Le processeur initialise les entrées/sorties sur la carte d'interface pour unités à disques magnétiques (6821 et 6850) et sur la carte d'interface pour terminal de visualisation (6532). Après quoi le processeur attend le caractère RUBOUT pour opérer la mesure de la durée de bit (taux de transmission) dont la valeur sera placée en \$FA5A... \$FA5D (voir Paperware!).

3. Une fois que le taux de transmission a été déterminé, le processeur assure le chargement du contenu de la piste 0 (2 K de logiciel en langage machine). Pour cela, la marche à suivre comporte plusieurs étapes:

— Placer la tête d'enregistrement/reproduction sur la piste 0. Un opto-coupleur (ou un interrupteur mécanique) signale au processeur que la tête est dans la bonne position;

— Le processeur émet l'impulsion de chargement de la tête, et attend le signal d'index (l'orifice permettant de repérer le début des pistes) — émis par un opto-coupleur.

— Aussitôt après l'extinction de l'impulsion d'index, le processeur initialise le registre de commande de l'ACIA (6850). Le format de transmission est le suivant: un bit de départ, huit bits de données, un bit de parité et un bit d'arrêt.

— Le processeur lit le premier octet sur la disquette; il s'agit de l'octet d'adresse de poids fort du début de la zone mémoire dans laquelle il faut ranger le contenu de la piste 0 (\$22). Le second octet de la piste 0 est l'octet de poids faible (\$00). Ces deux octets constituent l'indicateur de chargement (appelé "bump pointer") qui désignera donc l'adresse \$2200. A présent l'ordinateur lit le troisième octet sur la dis-

2

```
<RST>
<AD> FFE2
<GO>
```

```
(RUBOUT)
```

```
*TRACK 0&1*
```

```
<AD> 2217
<DA> 4C
<+> 40
<+> 22
```

```
<AD> 2245
<DA> 4C
<+> 76
<+> 22
```

```
<AD> 2283
<DA> 4C
<+> A6
<+> 22
```

```
<PC>
```

```
A*CA 0200=13,1
```

Tableau 2. Procédure d'adaptation d'une disquette OS-65D V3.1 au Junior Computer avec une seule unité à disques souples (1ère partie).

4

A*CA 4000=10,1	A*CA 4000=18,1	A*CA 4000=26,1
A*CA 4800=11,1	A*CA 4800=19,1	A*CA 4800=27,1
A*CA 5000=12,1	A*CA 5000=20,1	A*CA 5000=28,1
A*CA 5100=12,2	A*CA 5800=21,1	A*CA 5800=29,1
A*CA 5200=12,3	A*CA 6000=22,1	A*CA 6000=30,1
A*CA 5300=12,4	A*CA 6800=23,1	A*CA 6800=31,1
A*CA 5800=13,1	A*CA 7000=24,1	A*CA 7000=32,1
A*CA 6000=14,1	A*CA 7800=25,1	A*SA 26,1=4000/8
A*CA 6800=15,1	A*SA 18,1=4000/8	A*SA 27,1=4800/8
A*CA 7000=16,1	A*SA 19,1=4800/8	A*SA 28,1=5000/8
A*CA 7800=17,1	A*SA 20,1=5000/8	A*SA 29,1=5800/8
A*SA 10,1=4000/8	A*SA 21,1=5800/8	A*SA 30,1=6000/8
A*SA 11,1=4800/8	A*SA 22,1=6000/8	A*SA 31,1=6800/8
A*SA 12,1=5000/1	A*SA 23,1=6800/8	A*SA 32,1=7000/8
A*SA 12,2=5100/1	A*SA 24,1=7000/8	A*
A*SA 12,3=5200/1	A*SA 25,1=7800/8	
A*SA 12,4=5300/1		
A*SA 13,1=5800/5		
A*SA 14,1=6000/8		
A*SA 15,1=6800/8		
A*SA 16,1=7000/8		
A*SA 17,1=7800/8		

Tableau 4. Suite et fin de la procédure d'adaptation d'une disquette OS-65D V3.1 au Junior Computer (3ème partie).

quette: celui-ci indique le nombre de pages contenues dans la piste 0 (\$08 dans ce cas).

— Une fois que le contenu de la piste 0 a été chargé en mémoire, le processeur décharge la tête de lecture/écriture, et effectue un saut à l'adresse \$2200. A partir de là il trouve les indications nécessaires au chargement d'autres pistes et secteurs (avec leurs adresses de destination). Normalement, les deux K de logiciel de la piste 1 sont chargés à l'adresse \$2A00 (... \$31FF). Les pistes

3

```
A*IN
ARE YOU SURE?Y
A*GO 0200
```

```
- DISKETTE UTILITIES -
```

```
SELECT ONE:
1) COPIER
2) TRACK 0 READ/WRITE
? 2
```

```
- TRACK ZERO READ/WRITE UTILITY -
```

```
COMMANDS:
Rnnnn - READ INTO LOCATION nnnn.
Wnnnn/gggg,p - WRITE FROM nnnn FOR p PAGES
WITH gggg AS THE LOAD VECTOR
E - EXIT TO OS-65D
```

```
COMMAND? W2200/2200,8
```

```
- TRACK ZERO READ/WRITE UTILITY -
```

```
COMMANDS:
Rnnnn - READ INTO LOCATION nnnn.
Wnnnn/gggg,p - WRITE FROM nnnn FOR p PAGES
WITH gggg AS THE LOAD VECTOR
E - EXIT TO OS-65D
```

```
COMMAND? E
```

```
A*HOME
```

```
A*SA 01,1=2A00/8
```

Tableau 3. Suite de la procédure d'adaptation d'une disquette OS-65D V3.1 (2ème partie).

```
<RST>
<AD> FF17
<GO>

(RUBOUT)

OS-65D V3.0
OSI 9 DIGIT BASIC
COPYRIGHT 1977 BY MICROSOFT
36225 BYTES FREE
```

0 et 1 comportent en tout quatre K de logiciel en langage machine, lequel permet la mise en route du système d'exploitation de disques magnétiques.

— Aussitôt après, c'est le tour de l'interpréteur BASIC d'être chargé. Dans la version OS-65D V3.1, il est logé sur les pistes 2...4, tandis que dans la version OS-65D V3.3 il occupe non seulement ces pistes-là, mais aussi les pistes 6 et 13 (avec quelques extensions de l'éditeur). Dès l'achèvement du chargement de l'interpréteur BASIC, le processeur en

5

RUN"BEEXEC"

BASIC EXECUTIVE FOR
OS-65D V3.1

JUNE 25, 1980 RELEASE

FUNCTIONS AVAILABLE:

CHANGE- ALTER WORK-
SPACE LIMITS

DIR- PRINT DIRECTORY

UNLOCK- UNLOCK SYSTEM
FOR END USER MODI-
FICATIONS

FUNCTION? UNLOCK

SYSTEM OPEN

OK

RUN

BASIC EXECUTIVE FOR
OS-65D V3.1

JUNE 25, 1980 RELEASE

FUNCTIONS AVAILABLE:

CHANGE- ALTER WORK-
SPACE LIMITS

DIR- PRINT DIRECTORY

UNLOCK- UNLOCK SYSTEM
FOR END USER MODI-
FICATIONS

FUNCTION? DIR

LIST ON LINEPRINTER INSTEAD OF DEVICE # 1 ? NO

OS-65D VERSION 3.0
-- DIRECTORY --

FILE NAME	TRACK RANGE
OS65D3	0 - 12
BEEXEC*	14 - 14
CHANGE	15 - 16
CREATE	17 - 19
DELETE	20 - 20
DIR	21 - 21
DIRSRT	22 - 22
RANLST	23 - 24
RENAME	25 - 25
SECDIR	26 - 26
SEQLST	27 - 28
TRACE	29 - 29
ZERO	30 - 31
ASAMPL	32 - 32

50 ENTRIES FREE OUT OF 64

OK

Tableau 5. Procédure d'essai de la version 3.1. Le programme utilitaire BEEXEC* est exécuté après le chargement de l'interpréteur BASIC.

effectue un lancement à froid (Cold Start Entry) à l'adresse \$20E4. L'utilisateur a accès au système dès l'apparition du message "OK".

— Ces accès est encore limité (les instructions LIST, CONT, etc) ne sont pas encore acceptées. A vrai dire, il n'est pas même possible de créer un fichier en BASIC! La seule instruction acceptée et correctement exécutée est RUN. Et notamment pour le programme BEEXEC*, que l'on lance en faisant:

RUN"BEEXEC"

Après quoi le processeur présente le menu. Avec la version OS-65D V3.1, on optera pour "UNLOCK", tandis qu'avec la version OS-65D V3.3 ce sera l'option "9".

La réponse sera:

"SYSTEM OPEN"

6

OK
LIST

10 REM DIRECTORY UTILITY FOR OS-65D VERSION 3.0
20 REM
30 NF=0
40 PN=11897
50 DEF FNA(X)=10*INT(X/16)+X-16*INT(X/16)

BREAK

OK

NEW

OK

RUN"DIR"

LIST ON LINEPRINTER INSTEAD OF DEVICE # 1 ? NO

OS-65D VERSION 3.0
-- DIRECTORY --

FILE NAME	TRACK RANGE
OS65D3	0 - 12
BEEXEC*	14 - 14

BREAK IN 11110

OK

CONT

CHANGE 15 - 16

CREATE 17 - 19

DELETE 20 - 20

DIR 21 -

BREAK IN 11110

OK

Tableau 6. Vérification du fonctionnement de la touche BREAK avec une instruction LIST et pendant l'exécution d'un programme (RUN).

7

A*CA 0200=13,1

A*GO 0200

- DISKETTE UTILITIES -

SELECT ONE:

1) COPIER

2) TRACK 0 READ/WRITE

? 1

- DISKETTE COPIER -

FROM DRIVE (A/B/C/D)? A

TO DRIVE (A/B/C/D)? B

STARTING TRACK? 2

ENDING TRACK (INCLUSIVE)? 32

READY (Y/N)? Y

(suite ci-contre, en
haut à droite)

ANOTHER (Y/N)? N

A*

Tableau 7. Lorsque l'on dispose de deux unités (A et B), la procédure d'adaptation de la version OS-65D V3.1 est beaucoup plus simple qu'avec une seule unité. La copie est faite automatiquement d'une unité à l'autre.

A quoi on ajoutera l'instruction "NEW"...

A présent, l'ordinateur est prêt à recevoir un fichier en BASIC de l'utilisateur.

Le programme d'exploitation des disques souples

Le DOS d'Ohio Scientific est logé sur les pistes 0 et 1. Grâce à lui, on pourra changer facilement de processeur transitoire sur le Junior Computer. La figure 5 illustre les relations entre le DOS et les processeurs transitoires. Si par exemple le DOS est lancé à l'adresse \$FF17, le Junior Computer devient "BASIC Computer". Pour passer à l'Assembleur, il faut quitter le BASIC via l'instruction "EXIT". Le DOS émet le message A* ou B*; à quoi l'utilisateur répond

par AS ou ASSEMBLER. Le Junior Computer devient aussitôt "Assembler-Computer", permettant ainsi à l'utilisateur de créer des fichiers en langage assembleur, de mettre ces fichiers source sur disquette et de générer des codes-objets; ces derniers pourront être mis sur EPROM très facilement grâce à l'Eprogrammateur d'Elektor. On trouvera des explications détaillées quant à l'utilisation de l'Assembleur dans le manuel d'Ohio Scientific.

Attention! Dans la version OS-65D V3.1, il n'y a ni moniteur étendu ni assembleur sur disquette. Seule la version OS-65D V3.3 comporte l'assembleur et le moniteur étendu (EM).

Modification de la disquette OS-65D V3.1 avec une seule unité à disques souples

Si vous ne possédez qu'une seule unité à disques, l'adaptation de la version V3.1 au Junior Computer se traduira par une série de manipulations astreignantes. Les tableaux 2, 3 et 4 indiquent comment procéder.

A l'adresse \$FFE2 commence un programme qui permet le chargement du contenu de la disquette OS-65D V3.1 dans la mémoire du Junior Computer. Une fois que la touche (RUBOUT) a été actionnée, le processeur charge les pistes 0 et 1 et émet le message "TRACK 0&1*", puis les afficheurs à sept segments de la carte principale du Junior Computer s'allument. Une fois que l'utilisateur a modifié le contenu des adresses \$2217, \$2245 et \$2283 conformément aux indications du tableau 2, il lui faut retourner dans le système DOS en actionnant la touche (PC). Le processeur répond par le message A*.

L'instruction CA 0200 = 13,1 effectue le chargement de la piste 13 (1er secteur); les données sont rangées à l'adresse \$0200. Le fichier ainsi chargé est un copieur de disquettes assorti d'une option utilitaire pour la lecture et l'écriture de données de et sur la piste 0. Cette dernière permet notamment de réécrire le DOS modifié sur la piste 0. Mais avant cela, il y a lieu d'effectuer l'initialisation de la disquette vierge. Pour cela, il faut extraire la disquette d'Ohio Scientific de l'unité et y introduire la disquette vierge. Fermez la porte... et suivez la procédure du haut du tableau 3:

L'instruction IN d'abord (pour Initialisation); le processeur s'assure de votre santé mentale: ARE YOU SURE? (en êtes-vous certain?); la réponse est OUI, c'est à dire YES ou Y... Ne vous affolez pas... car aussitôt après votre unité à disques se met à faire un tintamarre effrayant, et ceci jusqu'à ce que l'initialisation des 40 pistes de la disquette soit achevée. Une fois que les données de formatage figurent sur chaque piste de la disquette vierge, celle-ci est prête à recevoir les données, con-

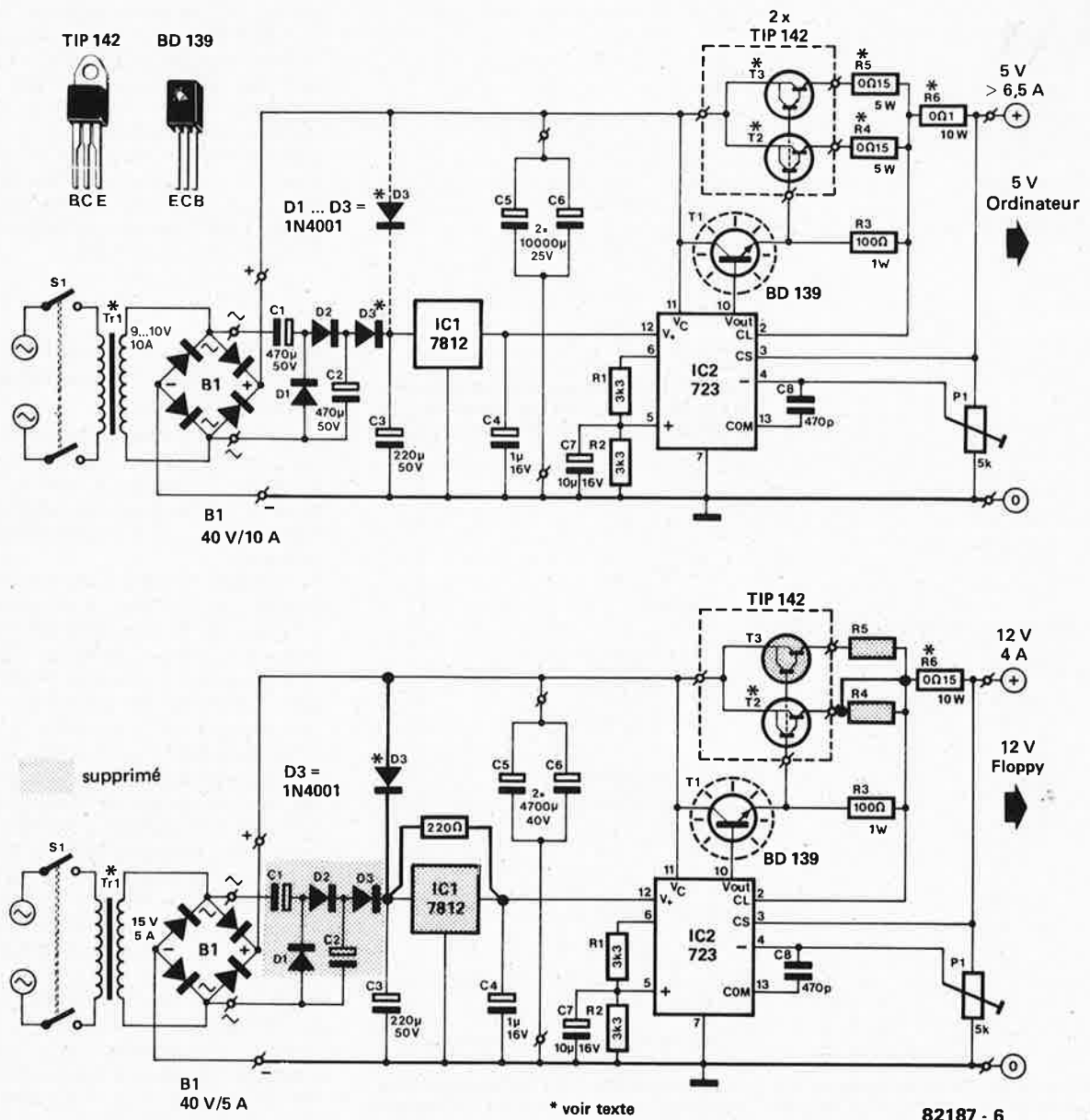


Figure 6. L'alimentation du Junior DOS Computer doit offrir de solides garanties. Avec les schémas proposés ici il est possible d'alimenter un système complet, avec quatre unités à disques souples.

formément à la procédure décrite au tableau 3. On commence par l'instruction W2200,2200,8, qui assure le chargement de huit pages sur la piste 0. L'adresse de début du fichier transféré est \$2200, tandis que le vecteur de chargement pour l'amorce est également \$2200. De sorte qu'avec cette manipulation on aura mis sur disquette vierge la première moitié du DOS adapté au Junior Computer.

L'instruction SA 01,1 = 2A00/8 permet de mettre sur le secteur 1 de la piste 1 un bloc de données de 8 pages, commençant à l'adresse \$2A00. C'est ainsi

que l'on sauvegarde sur disquette la seconde moitié du DOS. Malgré l'option de copiage d'unité à unité disponible avec la version V3.1, nous ne disposons que d'une unité et il faudra donc effectuer la copie piste à piste et à la main. Le tableau 4 indique la procédure à suivre. Il faut noter qu'avec l'instruction "CA" (pour CALL), c'est la disquette d'Ohio Scientific qui se trouve dans l'unité A, alors qu'avec l'instruction "SA" (pour SAVE) c'est la disquette de l'utilisateur. Une fois que les pistes 2...32 de la disquette d'Ohio Scientific auront été copiées, il ne reste plus qu'à

passer aux essais... On commence par couper la tension d'alimentation que l'on rétablit quelques secondes après; puis on introduit la nouvelle disquette dans l'unité A; l'adresse de lancement de l'amorce est \$FF17 sur le clavier hexadécimal du Junior Computer, comme l'indique le tableau 4. Le système se déclare et annonce le nombre d'octets disponibles (avec 48 K, vous disposez de 36225 octets).

L'instruction RUN"BEXEC*" permet d'obtenir le chargement et l'exécution du programme utilitaire BEXEC*, écrit en BASIC. Parmi les options proposées,

8

```
<RST>
<AD> FFE8
<GO>
```

```
(RUBOUT)
```

```
*TRACK 0&1*
```

```
<PC>
```

```
A*GO 2276
```

```
OS-65D V3.0
```

```
OSI 9 DIGIT BASIC
```

```
COPYRIGHT 1977 BY MICROSOFT
```

```
33921 BYTES FREE
```

```
Ok
```

```
RUN "BEXEC"
```

```
OS-65D Tutorial disk five - Sept. 16, 1981
```

```
1 > Directory
```

```
2 > Create a new file
```

```
3 > Change a file name
```

```
4 > Delete file from diskette
```

```
5 > Create blank data diskette
```

```
6 > Create data diskette with files
```

```
7 > Create buffer space for data files
```

```
8 > Single or dual disk drive copier
```

```
9 > Enter OS-65D system
```

```
Type the number of your selection
and depress RETURN ? 8
```

```
- Diskette copier -
```

```
Copy from which drive (A/B/C/D) ? A
```

```
Copy to which drive (A/B/C/D) ? A
```

```
What is the last track to be copied (Inclusive) <0-39> ? 39
```

```
Are you ready to start copying (Y/N) ? Y
```

```
Insert master diskette -- press <RETURN> ?
```

```
Reading --
```

```
Insert blank diskette -- press <RETURN> ?
```

```
Initializing --
```

```
Track 01 - 01/08
```

```
Track 02 - 01/08
```

```
Track 03 - 01/08
```

```
Track 04 - 01/08
```

```
Track 05 - 01/08
```

```
Track 06 - 01/01 - 02/01 - 03/01 - 04/02
```

```
Track 07 - 01/08
```

```
Track 08 - 01/08
```

```
Track 09 - 01/08
```

```
Track 10 - 01/08
```

```
Track 11 - 01/01 - 02/01 - 03/01 - 04/01 - 05/01 - 06/01 - 07/01
```

```
Track 12 - 01/01 - 02/01 - 03/01 - 04/01
```

```
Track 13 - 01/08
```

```
Track 14 - 01/08
```

```
Insert master diskette -- press <RETURN> ?
```

```
Reading --
```

```
Insert blank diskette -- press <RETURN> ?
```

```
Track 15 - 01/08
```

```
Track 16 - 01/08
```

```
Track 17 - 01/08
```

```
Track 18 - 01/08
```

```
Track 19 - 01/08
```

```
Track 20 - 01/08
```

```
Track 21 - 01/08
```

```
Track 22 - 01/08
```

```
Track 23 - 01/08
```

```
Track 24 - 01/08
```

```
Track 25 - 01/08
```

```
Track 26 - 01/08
```

```
Track 27 - 01/08
```

```
Track 28 - 01/08
```

```
Insert master diskette -- press <RETURN> ?
```

```
Reading --
```

```
Insert blank diskette -- press <RETURN> ?
```

```
Track 29 - 01/08
```

```
Track 30 - 01/08
```

```
Track 31 - 01/08
```

```
Track 32 - 01/08
```

```
Track 33 - 01/08
```

```
Track 34 - 01/08
```

```
Track 35 - 01/08
```

```
Track 36 - 01/08
```

```
Track 37 - 01/08
```

```
Track 38 - 01/08
```

```
Track 39 - 01/05 - 02/02
```

```
Insert master diskette -- press <RETURN> ?
```

```
Please, put the tutorial disk in drive A and depress <RETURN>.
```

```
<RST>
```

```
<AD> FFE8
```

```
<GO>
```

```
(RUBOUT)
```

Tableau 8. Procédure d'adaptation d'une disquette OS-65D V3.3 au Junior Computer. L'interpréteur BASIC est chargé via une directive du DOS (GO 2276), et copié grâce à une option du programme utilitaire BEXEC*. Pour finir, on procède aux essais.

```
*TRACK 0&1*
```

```
<AD> 2217
```

```
<DA> 4C
```

```
<+> 40
```

```
<+> 22
```

```
<AD> 2245
```

```
<DA> 4C
```

```
<+> 76
```

```
<+> 22
```

```
<AD> 2285
```

```
<DA> 8E
```

```
<+> C6
```

```
<+> 2A
```

```
<+> 4C
```

```
<+> B3
```

```
<+> 22
```

```
<AD> 2E84
```

```
<DA> 4C
```

```
<+> B0
```

```
<+> 2E
```

```
<PC>
```

```
A*CA 0200=06,4
```

```
A*GO 0200
```

```
- TRACK ZERO READ/WRITE UTILITY -
```

```
COMMANDS:
```

```
Rnnnn - READ INTO LOCATION nnnn.
```

```
Wnnnn/gggg,p - WRITE FROM nnnn FOR p PAGES
WITH gggg AS THE LOAD VECTOR
```

```
E - EXIT TO OS-65D
```

```
COMMAND? W2200/2200,8
```

```
- TRACK ZERO READ/WRITE UTILITY >
```

```
COMMANDS:
```

```
Rnnnn - READ INTO LOCATION nnnn.
```

```
Wnnnn/gggg,p - WRITE FROM nnnn FOR p PAGES
WITH gggg AS THE LOAD VECTOR
```

```
E - EXIT TO OS-65D
```

```
COMMAND? E
```

```
A*SA 01,1=2A00/8
```

```
A*
```

```
<RST>
```

```
<AD> FF17
```

```
<GO>
```

```
(RUBOUT)
```

```
OS-65D V3.0
```

```
OSI 9 DIGIT BASIC
```

```
COPYRIGHT 1977 BY MICROSOFT
```

```
34177 BYTES FREE
```

```
Ok
```

```
RUN "BEXEC"
```

```
OS-65D Tutorial disk five - Sept. 16, 1981
```

```
1 > Directory
```

```
2 > Create a new file
```

```
3 > Change a file name
```

```
4 > Delete file from diskette
```

```
5 > Create blank data diskette
```

```
6 > Create data diskette with files
```

```
7 > Create buffer space for data files
```

```
8 > Single or dual disk drive copier
```

```
9 > Enter OS-65D system
```

```
Type the number of your selection
and depress RETURN ? 1
```

```
Directory utility
```

```
Directory of which drive ?
```

```
Type A,B,C or D and depress RETURN <A> ?
```

```
Do you want to list the directory
to the printer (Yes or No) <No> ?
```

```
-- Directory --
```

```
File name Track range
```

```
OS65D3 0 - 13
```

```
BEXEC* 14 - 16
```

```
COPIER 17 - 18
```

```
CHANGE 19 - 20
```

```
CREATE 21 - 22
```

```
DELETE 23 - 23
```

```
DIR 24 - 24
```

```
RANLST 25 - 26
```

```
RENAME 27 - 27
```

```
SECDIR 28 - 28
```

```
SEQLST 29 - 30
```

```
TRACE 31 - 31
```

```
ZERO 32 - 33
```

```
ASAMPL 34 - 34
```

```
ATNENB 35 - 35
```

```
COLORS 36 - 36
```

```
MODEM 37 - 38
```

```
COMPAR 39 - 39
```

```
46 Entries free out of 64
```

```
Depress RETURN to continue ?
```


choisissez UNLOCK. A présent, la machine est prête à fonctionner en BASIC.

La suite est décrite par le tableau 5. Le programme DIRECTORY permet d'obtenir un listing du répertoire. Profitons-en pour vérifier le fonctionnement de la touche (BREAK) (tableau 6). On commence par interrompre un listing, puis l'exécution d'un programme. L'instruction CONT permet de relancer l'exécution du programme interrompu.

Modification d'une disquette OS-65D V3.1 à l'aide de deux unités

L'adaptation de la version 3.1 au Junior Computer est bien plus simple lorsque l'on dispose de deux unités à disques. La modification des pistes 0 et 1 se fait conformément aux indications données par les tableaux 2 et 3. Une fois que c'est chose faite, extrayez votre disquette de l'unité A et introduisez-la dans l'unité B. Remettez la disquette originale d'Ohio Scientific dans l'unité A et suivez la procédure du tableau 7. Cette fois la copie est automatique, de la piste 2 jusqu'à la piste 32 depuis l'unité A jusqu'à l'unité B, sans que vous ayez à faire autre chose que d'attendre.

Modification d'une disquette OS-65D V3.3

La modification de la version V3.3 est plus simple que celle de la version V3.1. Il suffit d'une seule unité. Le tableau 8 indique comment procéder. On commence par charger les pistes 0 et 1. Une fois que le DOS est en mémoire, il nous est permis de charger l'interpréteur BASIC à l'aide de l'instruction (GO) 2276. Le message OK indique que la manœuvre est réussie. L'étape suivante consiste à charger et à lancer le programme utilitaire BEXEC* à l'aide de l'instruction RUN "BEXEC*". Parmi les options proposées, on choisira "8", c'est à dire le programme copieur. Une fois que la copie est effectuée, il reste quelques modifications à effectuer dans le DOS. Actionner la touche (RST) et lancer le programme à l'adresse \$FFE8 après avoir introduit la nouvelle disquette (copie).

Les pistes 0 et 1 sont à nouveau chargées en mémoire. Toutefois sans que l'interpréteur BASIC ne vienne surcharger l'espace mémoire \$2200...\$22FF. Le tableau 8 indique les modifications à apporter au contenu des adresses \$2217, \$2245, \$2285 et \$2E84. L'étape suivante consiste à charger le programme de lecture et d'écriture sur la piste 0, que l'on trouve sur le secteur 4 de la piste 6. Ce logiciel utilitaire permet de remettre sur la disquette le contenu modifié de la piste 0.

L'instruction SA 01,1 = 2A00/8 assure la sauvegarde sur la nouvelle disquette de huit pages de logiciel DOS sur la piste 1. Vous disposez ainsi d'une dis-

7

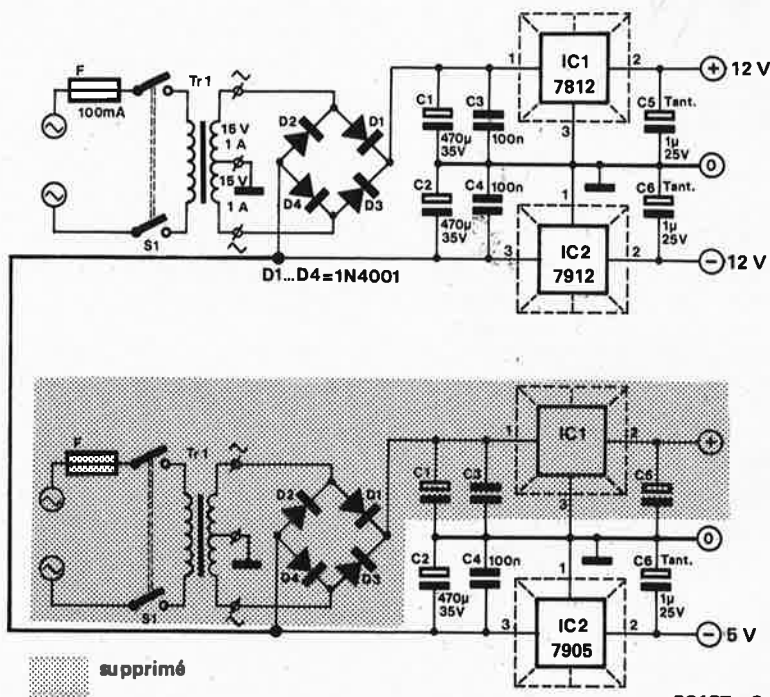


Figure 7. Pour les tensions "spéciales", nous recommandons le circuit ci-dessus, qui a déjà fait ses preuves maintes fois dans nos colonnes.

quette V3.3 adaptée au Junior Computer. L'adresse \$FF17 permet le lancement du système BASIC: le chargement du DOS et de l'interpréteur BASIC est assuré automatiquement dès que vous actionnez la touche (RUBOUT) (voir tableau 8).

L'interpréteur de directives du DOS

Le DOS dispose d'un interpréteur de directives dont nous allons décrire les plus importantes. Une manipulation fautive lors de l'introduction d'une directive est sanctionnée par un message d'erreur.

Chaque fois que le processeur a émis le message A* ou B*, il est prêt à recevoir une directive dont il ne retient d'ailleurs que les deux premiers caractères (en majuscules, s.v.p.); de sorte que pour la directive SAVE par exemple, il suffit d'indiquer SA.

La directive AS ou ASM

Lorsqu'il reçoit cette directive, le processeur entreprend aussitôt le chargement de l'assembleur et du moniteur étendu depuis l'unité sélectionnée à ce moment-là. Après quoi il effectue un saut vers l'entrée à froid de l'assembleur (celui-ci est doté d'un éditeur).

La directive EM

Le chargement est le même que ci-dessus, mais cette fois le saut est effectué à l'adresse de lancement du moniteur étendu. Celui-ci est écrit en langage machine, et permet la manipulation et le déverminage de programmes eux-mêmes en langage machine. Ce moniteur est doté d'un jeu d'instructions particulières, dont les plus importantes sont:

● ISTRING

Où STRING est une directive que le moniteur envoie à l'interpréteur de directives du DOS.

● @NNNN

Donne accès à un emplacement mémoire pour l'exécution des manipulations suivantes:

- (LF) : donne accès à l'adresse suivante
- (CR) : supprime l'accès à l'emplacement actuellement adressé
- (D) (D): données à mettre à l'emplacement adressé
- (") : impression du caractère ASCII de l'emplacement adressé
- (/) : prépare l'emplacement adressé à recevoir une donnée
- (^ ou (^) : donne accès à l'emplacement mémoire précédent

● BN,LLLL

Placer l'instruction BREAK numérotée N à l'adresse LLLL; les instructions

BREAK peuvent être numérotées de 1 à 8.

- EN

Elimine l'instruction BREAK numérotée N.

- A

Impression du contenu de l'accumulateur tel qu'il était lors de l'exécution de la dernière instruction BREAK.

- C

Exécution du programme à partir de la dernière instruction BREAK.

- DNNNN,MMMM

Vidage mémoire hexadécimale de l'adresse NNNN à l'adresse MMMM (excluse).

- EX

Retour au DOS.

- FNNNN,MMMM=DD

Placer la donnée DD de l'adresse NNNN à l'adresse MMMM-1.

- GNNNN

Effectuer un saut à l'adresse NNNN en

vue de l'exécution du programme qui s'y trouve.

- HNNNN,MMM(OP)

Permet d'effectuer des calculs hexadécimaux; NNNN et MMMM sont les données hexadécimales tandis qu'(OP) est l'opérateur: soit +, -, x ou /. Ce qui permet d'effectuer des additions, des soustractions, des multiplications et des divisions hexadécimales.

- MNNNN=MMMM,LLLL

Copie le contenu de MMMM à LLLL-1 (ou MMMM et LLLL sont des adresses) à l'adresse NNNN et suivantes.

- RMMMM=NNNN,LLLL

Relocation de programmes en langage machine par correction de toutes les adresses absolues d'origine (NNNN... LLLL-1) en adresses absolues de destination (à partir de l'adresse MMMM).

Le moniteur étendu dispose encore d'autres instructions que nous ne détaillerons pas ici malgré leur intérêt indiscutable. Nous renvoyons le lecteur à la documentation fournie par Ohio Scientific.

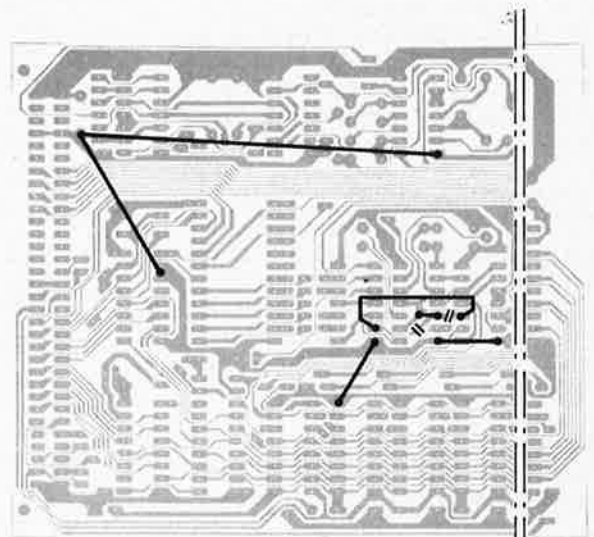
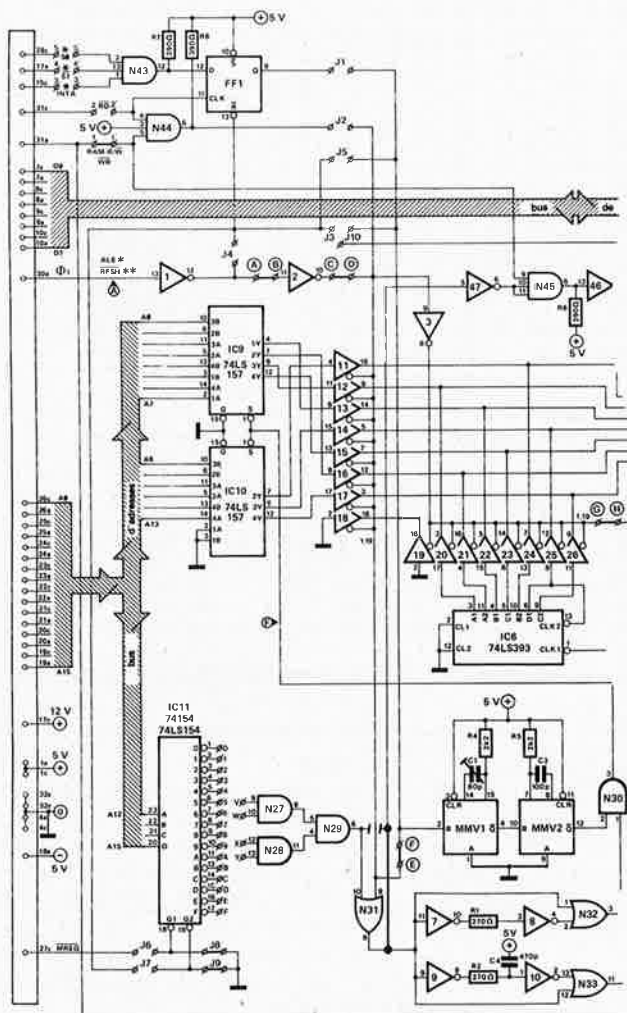
La directive BA

Le processeur charge l'interpréteur BASIC depuis l'unité sélectionnée à ce moment-là. Après quoi il effectue un saut vers l'entrée à froid de l'interpréteur: il indique le nombre d'octets disponibles et émet le message OK.

La directive CA NNNN=TT,S

... ou CALL NNNN=TT,S. Le processeur charge les données qu'il trouve dans le secteur S de la piste TT et les dépose à l'adresse NNNN et suivantes. Les numéros valides pour TT sont compris entre 01 et 39, tandis que pour S ils vont de 1 à 8.

8



82187 - 11

Figure 8. Les modifications indiquées ci-dessus permettent de tirer meilleur parti des cartes 16 K RAM dynamiques rétives. On remarque l'interruption de la liaison entre la sortie de N29 et l'entrée de N47. Cette dernière est reliée maintenant avec la sortie de N31. Par ailleurs, C3 peut être omis et C1 remplacé par un ajustable de 80 p.

La directive SA TT,S=NNNN/P

... ou SAVE TT,S=NNNN/P. Le processeur charge P pages de données à partir de l'adresse NNNN, et les sauvegarde sur disquette dans le secteur S de la piste TT (TT = 01...39; S = 1...8 et P = 1...8).

La directive DI TT ou DIR TT

Cette directive permet de connaître l'organisation (nombre de secteurs) de la piste TT (TT = 01...39).

La directive IN ou INIT

Le processeur assure l'initialisation d'une disquette vierge.

La directive IN TT

Le processeur initialise la piste TT.

La directive SE X

... ou SELECT X. Cette directive permet de changer la sélection des unités (X = A, B, C ou D).

La directive LO FILE

... ou LOAD FILE. Si le nom "FILE" est disponible dans le répertoire, le processeur charge le fichier portant ce nom et le place dans la mémoire de travail. Le nom d'un fichier doit nécessairement commencer par une lettre (A...Z) et ne peut comporter que 6 lettres au maximum.

La directive PU FILE

... ou PUT FILE. Le fichier présent dans la mémoire de travail est baptisé FILE est sauvegardé sur disquette, à condition que le répertoire de la disquette comporte ce nom auquel s'applique d'ailleurs les mêmes règles que ci-dessus.

Les directives PU TT et LO TT

Grâce à ces directives, il est possible de charger/sauvegarder un fichier (à condition toutefois qu'il n'exède pas 2 K) sans lui attribuer de nom. TT est le numéro de la piste. Nous déconseillons fortement l'usage de ces deux directives... périlleuses!

La directive RE

... ou RETURN. La directive RETURN permet de quitter le DOS pour retourner à l'assembleur, au BASIC ou au moniteur étendu; les formulations correspondantes sont RE AS, RE BA et RE EM.

On trouvera encore bien d'autres informations complémentaires dans le manuel fourni par Ohio Scientific.

Remarques

Une système à microprocesseur doté d'un DOS nécessite une alimentation robuste et fiable. Si vous n'en disposez pas encore, nous vous proposons les schémas des figures 6 et 7 que nous utilisons nous-mêmes au laboratoire d'Elektor.

Les deux transformateurs principaux

9

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
FC00:	A9	1E	8D	83	FA	A9	01	85	FD	A2	FF	9A	78	D8	20	7C
FC10:	FC	D0	FB	20	7C	FC	F0	FB	20	7C	FC	F0	F6	20	E1	FC
FC20:	C9	13	D0	07	A2	FF	9A	D8	6C	FA	00	C9	10	D0	06	A9
FC30:	01	85	FD	D0	14	C9	11	D0	06	A9	00	85	FD	F0	0A	C9
FC40:	12	D0	09	E6	FA	D0	02	E6	FB	4C	0D	FC	C9	14	D0	03
FC50:	4C	FD	FE	C9	15	10	F2	85	FF	A4	FD	D0	0D	B1	FA	0A
FC60:	0A	0A	0A	05	FF	91	FA	4C	49	FC	A2	04	06	FA	26	FB
FC70:	CA	D0	F9	A5	FA	05	FF	85	FA	4C	49	FC	A0	00	B1	FA
FC80:	85	F9	A9	7F	8D	81	FA	A2	08	A5	FB	20	B8	FC	A5	FA
FC90:	20	B8	FC	A5	F9	20	B8	FC	A9	00	8D	81	FA	A0	03	A2
FCA0:	00	A9	FF	8E	82	FA	E8	E8	2D	80	FA	88	D0	F5	A0	06
FCB0:	8C	82	FA	09	80	49	FF	60	48	84	FC	4A	4A	4A	4A	20
FCC0:	CB	FC	68	29	0F	20	CB	FC	A4	FC	60	A8	B9	08	FD	8D
FCD0:	80	FA	8E	82	FA	A0	FF	88	D0	FD	88	8C	82	FA	E8	E8
FCE0:	60	A2	21	A0	01	20	A1	FC	D0	07	E0	27	D0	F5	A9	15
FCF0:	60	A0	FF	0A	B0	03	C8	10	FA	8A	29	0F	0A	AA	98	10
FD00:	03	18	69	07	CA	D0	FA	60	40	79	24	30	19	12	02	78
FD10:	00	10	08	03	46	21	06	0E	20	1E	FD	6C	FD	00	20	28
FD20:	FD	20	CF	FD	20	4F	FD	60	A0	00	8C	01	C0	A9	40	8D
FD30:	00	C0	A9	04	8D	01	C0	A9	40	8D	00	C0	A2	04	8E	01
FD40:	C0	8C	03	C0	88	8C	02	C0	8E	03	C0	8C	02	C0	60	A9
FD50:	FB	D0	09	A9	02	2C	00	C0	F0	1C	A9	FF	8D	02	C0	20
FD60:	CE	FD	29	F7	8D	02	C0	20	CE	FD	09	08	8D	02	C0	A2
FD70:	18	20	BA	FD	F0	DD	A2	7F	8E	02	C0	20	D7	FC	AD	00
FD80:	C0	30	FB	AD	00	C0	10	FB	A9	03	8D	10	C0	A9	58	8D
FD90:	10	C0	20	C5	FD	85	FE	AA	20	C5	FD	85	FD	20	C5	FD
FDA0:	85	FF	A0	00	20	C5	FD	91	FD	C8	D0	F8	E6	FE	C6	FF
FDB0:	D0	F2	86	FE	A9	FF	8D	02	C0	60	A0	F8	88	D0	FD	55
FDC0:	FF	CA	D0	F6	60	AD	10	C0	4A	90	FA	AD	11	C0	60	D8
FDD0:	78	A9	67	8D	82	FA	A9	00	8D	80	FA	A2	FC	8E	5A	FA
FDE0:	A2	FF	8E	5B	FA	EA	A9	7F	8D	81	FA	4A	8D	83	FA	A2
FDF0:	03	8E	59	FA	2C	80	FA	30	FB	20	4F	FE	4E	5F	FA	6E
FE00:	5E	FA	AD	5E	FA	8D	5C	FA	AD	5F	FA	8D	5D	FA	A2	08
FE10:	20	72	FE	20	2B	FE	C9	7F	D0	B5	60	2C	80	FA	30	FB
FE20:	8E	61	FA	A2	08	20	72	FE	20	81	FE	2C	80	FA	10	09
FE30:	38	6E	62	FA	CA	D0	F1	F0	07	18	6E	62	FA	CA	D0	E8
FE40:	20	81	FE	AD	62	FA	29	7F	8D	63	23	AE	61	FA	60	18
FE50:	AD	5A	FA	69	01	8D	5A	FA	AD	5B	FA	69	00	8D	5B	FA
FE60:	2C	80	FA	10	EA	AD	5A	FA	AD	5E	FA	AD	5B	FA	8D	5F
FE70:	FA	60	AD	5C	FA	8D	5E	FA	AD	5D	FA	8D	5F	FA	38	B0
FE80:	0C	AD	5A	FA	8D	5E	FA	AD	5B	FA	8D	5F	FA	38	AD	5E
FE90:	FA	E9	01	8D	5E	FA	AD	5F	FA	E9	00	8D	5F	FA	EA	EA
FEA0:	B0	FE	60	8E	60	FA	8D	62	FA	AD	82	FA	29	40	D0	F9
FEB0:	AD	82	FA	29	FE	8D	82	FA	20	81	FE	A2	07	4E	62	FA
FEC0:	90	30	AD	82	FA	09	01	8D	82	FA	20	81	FE	CA	D0	ED
FED0:	AE	59	FA	AD	82	FA	09	01	8D	82	FA	20	81	FE	CA	D0
FEF0:	F2	2C	80	FA	10	04	AE	60	FA	60	2C	80	FA	10	FB	6C
FEF0:	7C	FA	AD	82	FA	29	FE	8D	82	FA	18	90	CD	20	03	FF
FF00:	4C	51	2A	A9	27	8D	82	FA	A9	00	8D	80	FA	A9	7F	8D
FF10:	81	FA	4A	8D	83	FA	60	A9	2E	8D	7C	FA	A9	FF	8D	7D
FF20:	FA	A9	00	8D	7A	FA	A9	FC	8D	7B	FA	4C	18	FD	A9	03
FF30:	8D	25	23	60	A9	51	8D	7C	FA	A9	2A	8D	7D	FA	A9	00
FF40:	8D	7A	FA	A9	FC	8D	7B	FA	4C	18	FD	6C	7A	FA	6C	7E
FF50:	FA	20	1E	FD	A9	28	8D	A3	26	A9	01	8D	5E	26	20	BC
FF60:	26	A9	2A	85	FF	20	54	27	86	FE	20	67	29	A9	01	8D
FF70:	21	23	8D	22	23	8D	C6	2A	20	C6	29	A9	1A	8D	01	23
FF80:	8D	03	23	A9	A2	8D	11	23	8D	13	23	A9	FE	8D	02	23
FF90:	8D	04	23	8D	12	23	8D	14	23	60	EE	5E	26	A9	06	20
FFA0:	BC	26	20	67	29	EE	5E	26	A9	00	85	FE	85	FF	20	67
FFB0:	29	A9	01	8D	5E	26	A9	13	20	BC	26	A9	32	85	FF	A9
FFC0:	74	85	FE	20	54	27	20	67	29	20	61	27	20	73	2D	0D
FFD0:	0A	2A	54	52	41	43	4B	20	30	26	31	2A	0D	0A	00	4C
FFE0:	00	FC	20	51	FF	4C	C9	FF	20	51	FF	4C	9A	FF	FF	FF
FFF0:	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	4B	FF	00	FC	4E	FF

Tableau 9. Vidage en format hexadécimal de l'EPROM amorce ESS 515. Le listing source est disponible en "Paperware".

seront de préférence toriques (9... 10 V/10 A et 15 V/4,5... 5 A), un petit transformateur ordinaire fournira deux fois 15 V/1 A pour l'alimentation des cartes RAM dynamiques. Le câblage de puissance sera effectué à l'aide de câble de section conséquente (au moins 1,5 mm²).

L'expérience a démenti les constatations enthousiastes que nous avions faites lors de la publication de la carte RAM dynamique: celle-ci ne fonctionne pas toujours comme il faut, notamment avec le 6809. C'est pourquoi nous avons procédé à quelques modifications que l'on trouvera réunies sur la figure 8, et qui assurent un fonctionnement désormais irréprochable à cette carte par ailleurs

très fiable.

Sur ce *mea culpa* nous refermons (provisoirement) le *grand livre* du Junior Computer en espérant que ce chapitre DOS n'aura pas trop irrité nos lecteurs peu enclins à la microphilie, et souhaitons bien du plaisir à tous ceux qui se lanceront dans cet aventure exaltante (mais peu reposante) du Junior DOS Computer. **M**

On pourra se procurer les disquettes et manuels des versions OS 65D-V3.1 et V3.3 auprès de la société:
Electronique JL
25, route du Pont Colbert
78 000 VERSAILLES
tél: 3/950 1354