

10/12

Améliorez votre CPC

10/12.1

Autoboot de CP/M sur CPC

Comme beaucoup de chefs de famille, vous avez délégué la gestion du budget familial à votre conjointe, plus experte et minutieuse que vous dans la tenue de la feuille de calcul sous Multiplan proposée en Partie 7, chapitre 2.1.5, pages 1 à 26, mais peu douée quant à la manipulation des disquettes et l'utilisation des commandes de lancement (:CPM - SUBMIT PROFILE.SUB...).

Ou alors êtes-vous de ceux qui utilisent souvent des logiciels fonctionnant sous CP/M, et qui en ont assez de frapper une commande Basic (:CPM) pour accéder à leur système d'exploitation favori. Vous jalousez probablement les possesseurs d'Amstrad PCW ou de compatibles PC qui peuvent lancer automatiquement un programme (on appelle cela BOO-TER sur un programme).

Ou tout simplement êtes-vous curieux et téméraire pour apporter des modifications sur votre CPC.

Nous nous sommes penchés sur ces problèmes et avons trouvé une solution simple et très bon marché. Aussi nous allons vous la livrer ainsi que

Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

toutes les études et réflexions qui nous ont conduits à ces solutions, ce afin de parfaire la connaissance de votre ordinateur.

Cette solution a été exploitée sur CPC-6128, mais doit fonctionner sans problème sur CPC-664 ainsi que sur CPC-464 muni de l'interface DDI-1.

Nous avons même poussé la performance jusqu'à permettre le démarrage de CPM automatiquement sur un lecteur additionnel type 5 pouces 1/4.

Vous trouverez aussi dans les lignes qui suivent des explications pour utiliser les nouvelles possibilités avec un lecteur VORTEX formatant sur 720 k-octets.

SOYONS TÊMÉRAIRE !

Pour résoudre ce problème (sur CPC-6128), nous fûmes obligés de prendre notre courage à pleine main, et de risquer la garantie de notre matériel.

Nous vous demanderons de faire de même, si le vôtre est encore sous garantie. Si celle-ci est passée, alors juste un peu de courage sera nécessaire : nous allons *désencapsuler* la carte principale pour y effectuer des modifications (couper des pistes, souder des fils, percer le boîtier, ajouter des interrupteurs).

La peur vous serre la gorge ? Rassurez-vous, nous vous donnerons suffisamment d'explications détaillées et de schémas pour assurer cette réalisation sans problèmes.

I. Le matériel

Nous allons ici faire l'inventaire de tout ce qui vous sera utile pour comprendre et réaliser notre modification.

POUR COMPRENDRE LE POURQUOI DE LA CHOSE

Vous trouverez ici le matériel nécessaire pour comprendre comment nous avons découvert ces nouvelles possibilités :

- le schéma structurel de l'unité centrale CPC 6128 (ou 664 ou 464) donné en Partie 2, chapitre 2.3, page 3 ;
- le schéma structurel de l'interface disquette donné en Partie 2, chapitre 3.5, page 5 ;
- quelques connaissances en assembleur Z80 pour pouvoir suivre les explications données ;
- quelques connaissances en électronique logique — celles de la Partie 13, chapitre 1.2, pages 1 à 22 devraient vous suffire. Nous ajouterons, quand cela s'avèrera nécessaire, quelques compléments d'information.

LE MATÉRIEL EN COMMUN POUR LES DEUX RÉALISATIONS

- Un tournevis cruciforme.
- Un cutter ou un petit couteau bien aiguisé.
- Un fer à souder 25 à 40 watts maximum.
- De la soudure à l'étain.
- Une perceuse et une mèche du diamètre des interrupteurs choisis (voir ci-dessous), ou, comme nous, quelques tournevis permettant de percer un trou dans le plastique tendre du boîtier de l'Amstrad.
- Une petite lime ronde pour le plastique.

LE MATÉRIEL POUR L'AUTOBOOT

- Deux fils de cuivre isolés de 60 centimètres de petite section (type fil téléphonique multibrin, si possible jointifs). Pour nos explications, nous supposons que l'un de ces fils est blanc et l'autre gris.
- Un interrupteur unipolaire petit modèle (si vous possédez, comme nous un inverseur, unipolaire ou bipolaire, pas de problème, nous vous précisons comment le câbler).

LE MATÉRIEL POUR L'AUTOBOOT SUR LECTEUR B

- Quatre fils de 60 centimètres de quatre couleurs différentes (préférez du câble en nappe). Nous avons utilisé les couleurs jaune, verte, bleue et violette — vous remplacerez dans le texte, par les couleurs de votre choix, celles que nous avons décrites.
- Deux fils d'environ 3 centimètres.
- Un inverseur bipolaire petit modèle.

II. Tout est dans le ROM WALK

Quel nom bizarre ! Il s'agit en effet d'un nom donné à une partie du système d'exploitation, qui effectue une recherche des différentes ROMs connectées autour de l'unité centrale (ROM WALK = littéralement promenade dans la Rom).

COMMENT CELA MARCHE ?

Vous savez certainement que votre AMSTRAD possède au moins *deux* ROMs :

- la ROM inférieure contenant le système d'exploitation qui permet essentiellement la communication avec le matériel (clavier, écran, ...)

Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

— la ROM supérieure contenant le programme INTERPRETEUR BASIC.

Aussi bizarre que cela paraisse, ces *deux* ROMs sont fictives, et toutes les données, qui permettent le bon fonctionnement de votre Amstrad, se trouvent dans le même composant ; charge à l'Amstrad de s'y retrouver.

Parmi ces *deux* ROMs, seule la ROM inférieure est connue lors de l'initialisation de votre CPC (à la mise sous tension). La ROM supérieure est ensuite sélectionnée, mais il est possible de faire croire au CPC que c'en est une autre que celle du Basic.

Il est ensuite effectué, si tout se passe comme prévu sur un Amstrad non modifié, une recherche de toutes les éventuelles ROMs connectées (notamment la ROM AMSDOS contenant une partie de CPM), et, procédé à leur initialisation.

Voyons un peu comment tout cela se déroule, en étudiant le désassemblage des parties intéressantes des données des ROMs du CPC-6128.

APPROFONDISSONS LE SUJET

Munissez-vous ici du schéma structurel de l'unité centrale, et repérez l'un des composants centraux : IC103, la ROM intégrée de 32 K BYTE, telle qu'elle est repérée (BYTE = octet, en anglais).

Lors de la mise sous tension, la ROM inférieure du système d'exploitation est automatiquement sélectionnée par le Video Gate Array : le signal ROMEN est au niveau logique zéro, ainsi que le signal présent sur la ligne ROMDIS, forcé par le composant résistif R112 connecté en rappel à la masse (0). Les broches OE et CS de la ROM IC103 sont donc à zéro, ce qui valide la ROM.

La réinitialisation de tous les composants effectuée, le Z80 commence à lire le programme qui y débute à partir de l'adresse &0000.

Une rapide programmation du Video Gate Array est effectuée, puis un saut à l'adresse &0591 poursuit le programme (toujours en ROM).

A partir de cette adresse sont initialisés les composants périphériques du Z80 (PIO 8255, Interface Centronics, le VDU 6845 par reconnaissance du type d'alimentation : secteur 50 ou 60 Hz).

Cette partie du programme se termine en &05C4, et on saute à l'adresse &05E5 où sont initialisés les registres DE et HL, et aussitôt, nouveau saut (encore ! Et ce n'est pas terminé !) à l'adresse &061F :

```
061F   F3           DI
0620   ED 56       IM    1
0622   01 00 DF   LD    BC,0DF00H
0626   ED 49       OUT   (C),C
.....
```

Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

Les interruptions sont inhibées (DI), le mode d'interruption M1 est initialisé (IM 1), BC est chargé avec la valeur &DF00 (LD BC,0DF00H) pour sortir la valeur 00 sur le port &DFXX (OUT (C),C).

Ces deux dernières instructions servent à sélectionner la ROM située sur le plan numéro 0, dans notre cas la ROM Basic. Pour sélectionner une ROM avec un numéro différent, il suffira d'effectuer un chargement de BC avec, par exemple, la valeur &FD06 pour la ROM numéro 6. Il est ainsi possible de sélectionner &FF = 255 ROMs supplémentaires en parallèles (l'Amstrad peut donc gérer en tout 256 ROMs supérieures, avec la ROM no 0).

La suite du programme permet l'initialisation de la configuration RAM, l'arrêt des moteurs des lecteurs de disquettes efface une partie de la mémoire, fixe le pointeur de pile, copie les vecteurs nécessaires en RAM (ceux qui vous sont énumérés Partie 4, chapitre 2.7).

Ensuite arrive un saut à l'adresse &0077 :

0077	7C	LD	A,H
0078	B5	OR	L
0079	79	LD	A,C
007A	20 04	JR	NZ,0080H
007C	7D	LD	A,L
007D	21 06 C0	LD	HL,0C006H
0080	32 D6 B8	LD	(0B8D6H),A
0083	32 D9 B8	LD	(0B8D9H),A
0086	22 D7 B8	LD	(0B8D7H),HL
0089	21 FF AB	LD	HL,0040H
008C	11 40 00	LD	DE,0040H
008F	11 FF B0	LD	BC,0B0FFH
0092	31 00 C0	LD	SP,0C000H
0095	DF	RST	18H
0096	D7	DEFB	0D7H
0097	B8	DEFB	0B8H
0098	C7	DEFB	0C7H
.....		

Suite à l'initialisation, lors du premier branchement à cette routine, le registre HL contient la valeur &0000, l'indicateur de zéro sera donc nul suite à l'opération LD A,H – OR L, on ne sautera donc pas à l'adresse &0080 (JR NZ,0080H), mais l'on poursuivra en &007C, suite au chargement de A par L (LD A,L) et de HL par &C006 (LD HL,0C006H), ainsi que des adresses mémoire RAM &B8D6 par A (LD (0B8D6H),A), &B8D9 (LD (0B8D9H),A), et &B8D7 par HL (LD (0B8D7),HL). On se retrouvera ainsi avec la configuration mémoire suivante :

B8D6	00	DEFB	00H
B8D7	06 C0	DEFW	0C006H
B8D9	00	DEFB	00H

Le contenu de &B8D6 indique le numéro de la ROM d'extension actuellement sélectionnée (ici 0).

Le contenu de &B8D7 et &B8D8 indique l'adresse à accéder dans la ROM sélectionnable (ici &C006), dont le numéro est contenu à l'adresse suivante.

Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

Le contenu de &B8D9 permettra la sélection de la ROM supérieure pour effectuer le saut à l'adresse précédemment citée (la ROM d'extension 0 est sélectionnée dans notre cas).

Dans la suite du programme, les registres BC, DE, et HL sont chargés avec différentes valeurs qui permettront entre autre de reconnaître la première et la dernière adresse de la mémoire libre pour les ROMs d'extensions ; le pointeur de pile est à nouveau initialisé à l'adresse &C000H.

Enfin, une instruction ReStart est effectuée (RST 18H). Cette instruction permet d'effectuer un saut en ROM ou en RAM, selon la configuration précisée par l'adresse pointée par les deux octets suivants. Les deux octets suivants contiennent les valeurs &D7 et &B8, ce qui nous permet de reconstituer l'adresse &B8D7. A partir de cette adresse, trois octets seront significatifs :

— les deux premiers indiquent l'adresse à laquelle il faut se brancher, dans notre cas : &C006 ;

— les deux suivants indiquent le status ROM/RAM pour permettre le saut. Une valeur comprise entre &00 et &FB (0 et 251) sélectionne la ROM supérieure dont le numéro est cité, et invalide la ROM inférieure. Dans notre cas : la ROM numéro 0.

Nous effectuons ainsi un saut à l'adresse &C006 de la ROM Basic. Qu'à cela ne tienne, désassemblons cette partie de la ROM supérieure :

C006	31 00 C0	LD	SP,0C000H
C009	CD CB BC	CALL	0BCCBH
C00C	CD 3F F5	CALL	0F53FH
C00F	DA 00 00	JP	C,0000H

Le pointeur de pile est une fois de plus initialisé, et l'on effectue un saut à l'adresse &BCCB qui se trouve en RAM (l'espace ROM ne peut-être compris qu'entre &0000-&3FFF ou &C000-&FFFF).

Les vecteurs ayant été précédemment initialisés, il suffit de se reporter à la Partie 4, chapitre 2.7, page 53, pour découvrir que ce vecteur permet d'effectuer l'initialisation des différentes ROMs d'extension : notre fameux ROM WALK (on retrouve l'utilisation des registres DE et HL pour connaître les différentes adresses libres utilisées par les ROMs pour gérer leurs données).

Ce vecteur effectue un saut à l'adresse &0326 de la ROM inférieure du système d'exploitation. Les différentes ROMs sont ensuite activées dans l'ordre décroissant.

Pour ne pas vous embrouiller, nous allons écourter ici notre description ; nous commençons à en connaître assez. Sachez tout de même que la ROM de l'AMSDOS est considérée comme ayant le numéro 7 et qu'elle est activée par la sortie sur le port &FDXX de la valeur &07, ce que nous allons prouver ci-dessous.

DÉCORTIQUONS L'INTERFACE DISQUETTE

Munissez-vous maintenant du schéma structurel de l'interface du lecteur de disquette.

Auparavant, sachez qu'il existe depuis quelque temps une norme spécifiant la façon de représenter les composants intégrés. Nous allons donc l'utiliser pour l'étude d'une partie du circuit, et vous trouverez en figure 1 la représentation de cette nouvelle norme (avec un rappel de l'ancienne au-dessous), pour chacun des composants qui vont nous intéresser dans l'interface. Nous y avons joint les différentes tables de vérité permettant d'en comprendre le fonctionnement dans les cas que nous étudierons.

Le fonctionnement particulier d'un composant nous sera important : il s'agit du « OU EXCLUSIF » à 2 entrées, aussi vous trouverez, en figure 2, ses deux modèles possibles. Lorsque sa sortie doit être considérée comme étant à l'état logique « 0 », elle se comporte comme si elle était connectée à la masse. Par contre, à l'état logique « 1 », la sortie est déconnectée, ce qui en fait ne prouvera rien sur la tension effectivement mesurée sur la broche : c'est le montage extérieur qui décidera de l'état présent.

Voyons, maintenant, ce qui se déroule lors de la sortie sur le port &DFXX de la valeur &07, qui se traduira par le programme suivant :

```
01 07 DF    LD    BC,007DFH
ED 49      OUT   (C),C
```

La valeur binaire 1101 1111 0000 0111, correspondant à la valeur &DF07, est placée sur le bus d'adresses de la manière suivante :

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
1	1	0	1	1	1	1	1	0	0	0	0	0	1	1	1

La valeur binaire 0000 0111, correspondant à la valeur hexadécimale &07, est placée sur le bus de données de la façon suivante :

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	1	1	1

Les signaux $\overline{\text{IORQ}}$ et $\overline{\text{WR}}$ sont activés et $\overline{\text{RD}}$ inactif :

- $\overline{\text{IORQ}} = 0$
- $\overline{\text{WR}} = 0$
- $\overline{\text{RD}} = 1$

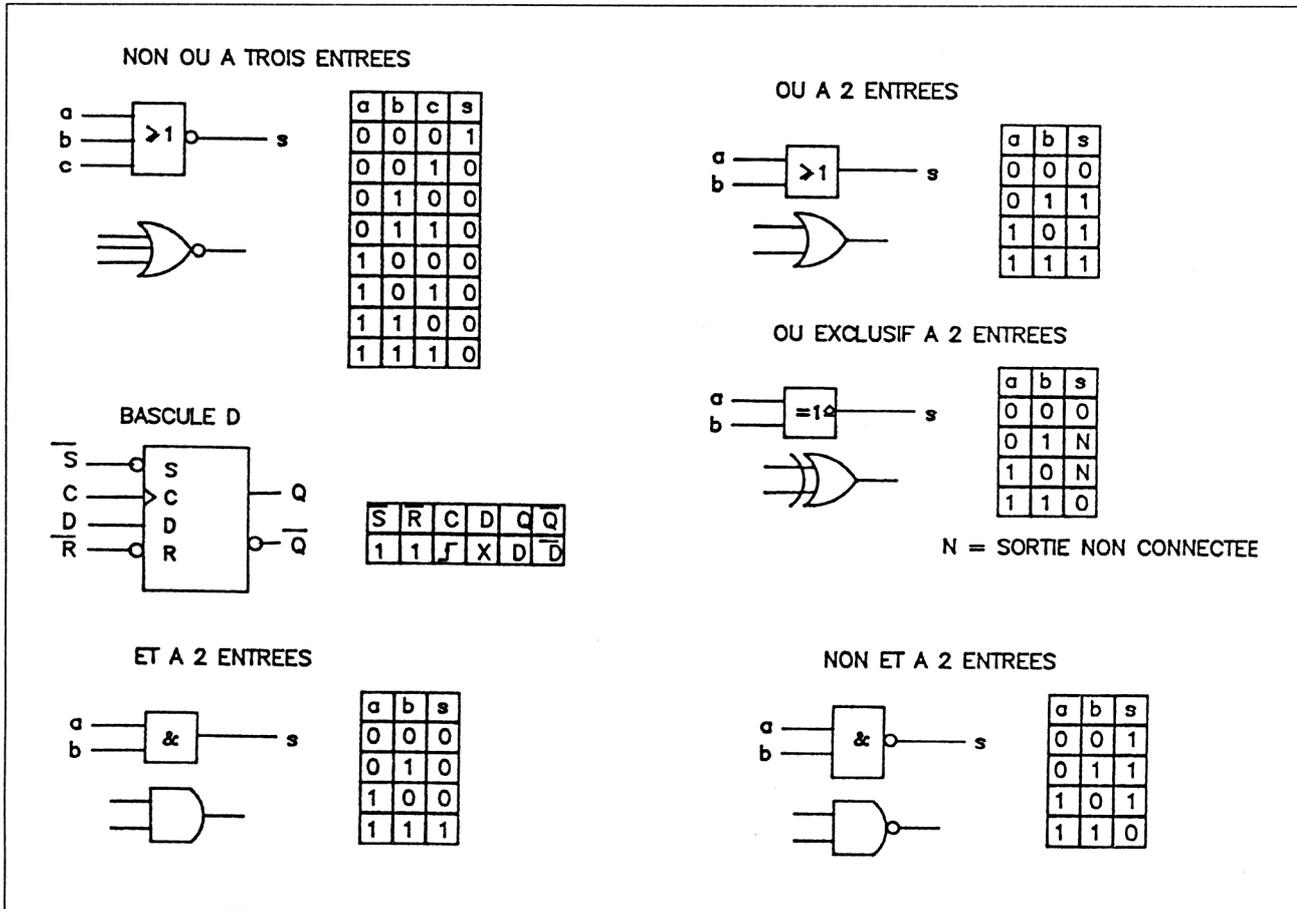


Fig. 1

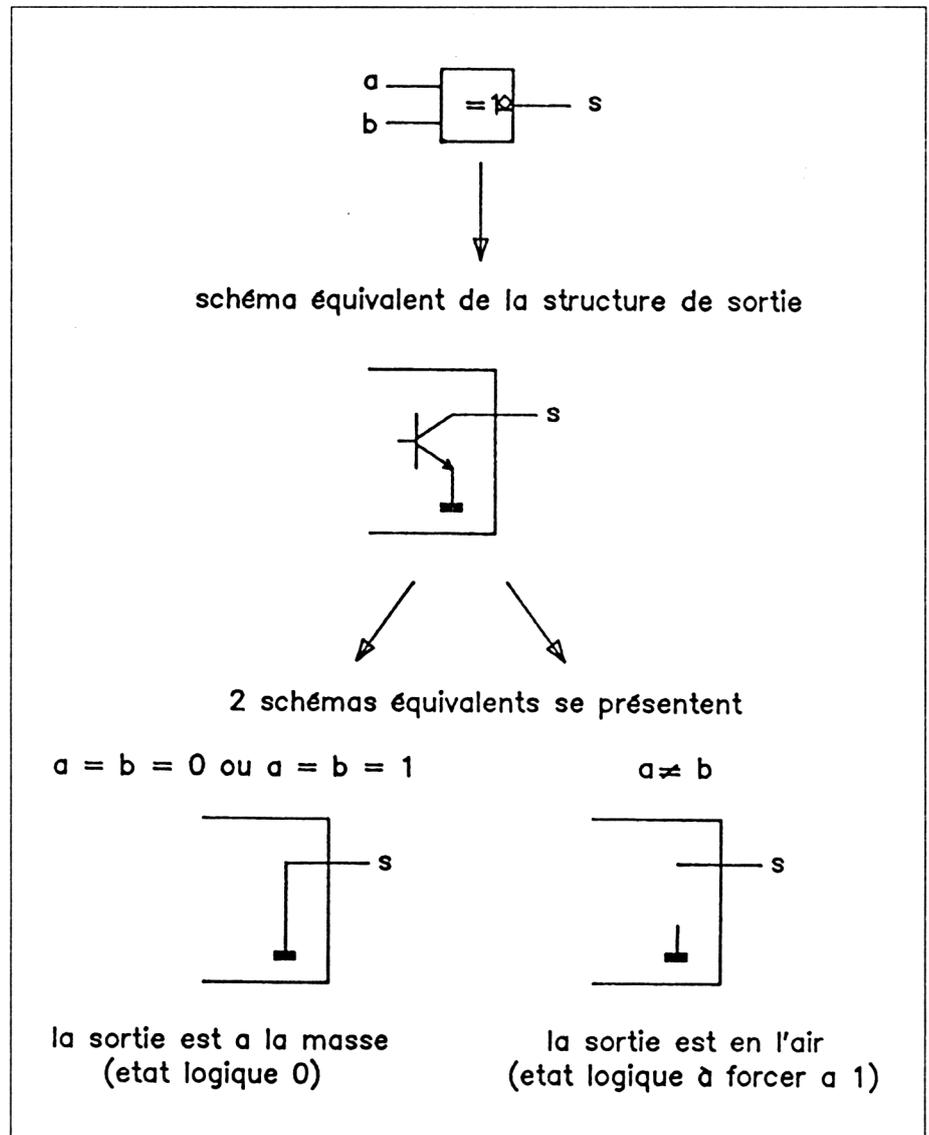


Fig. 2

Reportez-vous maintenant à la figure 3 qui représente le décodage de la valeur $\&07$ sur le bus de données (nous avons représenté aux normes la structure réalisée par les composants IC111-1, IC111-2, IC110-3, IC109-1, IC109-2, IC109-3, IC112-1, R111 et R112).

Prenons le cas du composant IC109-3 : l'une de ces entrées est placée à « 0 » (la masse) par l'intermédiaire de LK1 (LK1 est une liaison, qui a été repérée, sur le câblage imprimé de la carte d'interfaçage disquette). D0 est au niveau logique « 1 », donc la sortie de IC109-3 est placée en l'air.

Par le même raisonnement, les sorties de IC109-2 et IC109-1 sont en l'air.

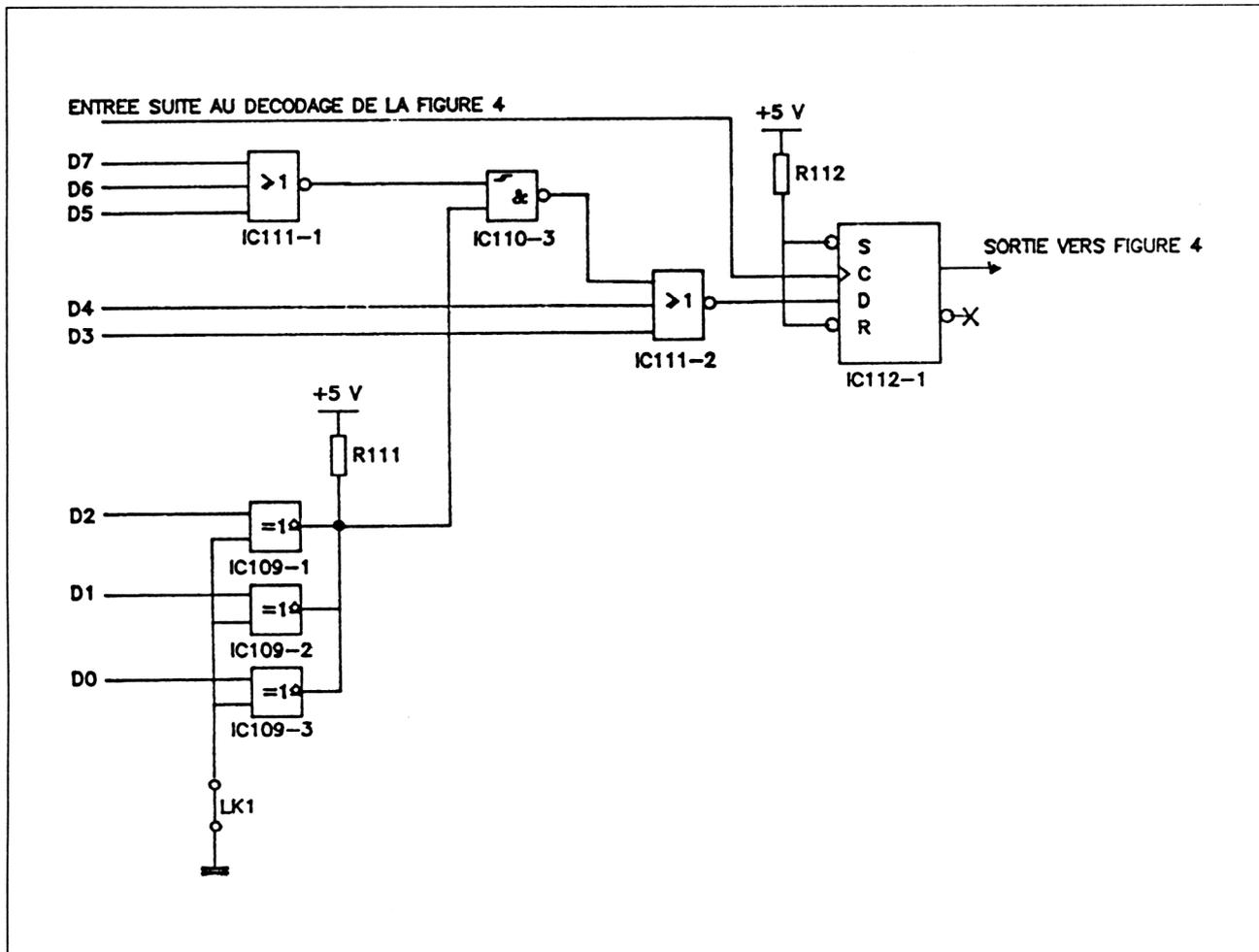


Fig. 3

Or, il a été placé le composant résistif R111 permettant d'amener une tension de +5 V au point de connexion des sorties, et comme les composants intégrés sont alimentés en 5 volts, un niveau logique « 1 » est présent sur l'entrée inférieure de IC110-3.

Occupons-nous maintenant de D7, D6, D5 qui sont tous trois à « 0 ». La sortie de IC111 se trouve ainsi à l'état logique « 1 ».

Deux états logiques 1 sur les entrées de IC110-3 assurent ainsi un état « 0 » sur sa sortie, qui est connectée sur une entrée de IC111-2.

IC111-2 possède ainsi toutes ses entrées à l'état 0, car D4 et D3 sont à « 0 », donc sa sortie est à l'état logique « 1 ».

Ce dernier état est donc présent sur l'entrée de la bascule D du composant IC112-1.

Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

Une première partie de la sélection de la ROM AMSDOS est ainsi réalisée en plaçant l'entrée D de la bascule au niveau logique « 1 ».

Sur cette bascule, les entrées R (Reset = forçage à « 0 ») et S (Set = forçage à « 1 »), sont rendues inactives par l'intermédiaire du composant R112, qui effectue un rappel au niveau « 1 ».

Le déclenchement de la bascule ne dépend plus que de l'entrée C (Clock = horloge), que nous allons étudier sur le schéma de la figure 4.

Sur ce schéma est reprise la bascule (IC112) et sont ajoutés les composants qui nous intéressent : IC107-1, IC107-2, IC107-3, IC110-1, IC105-1.

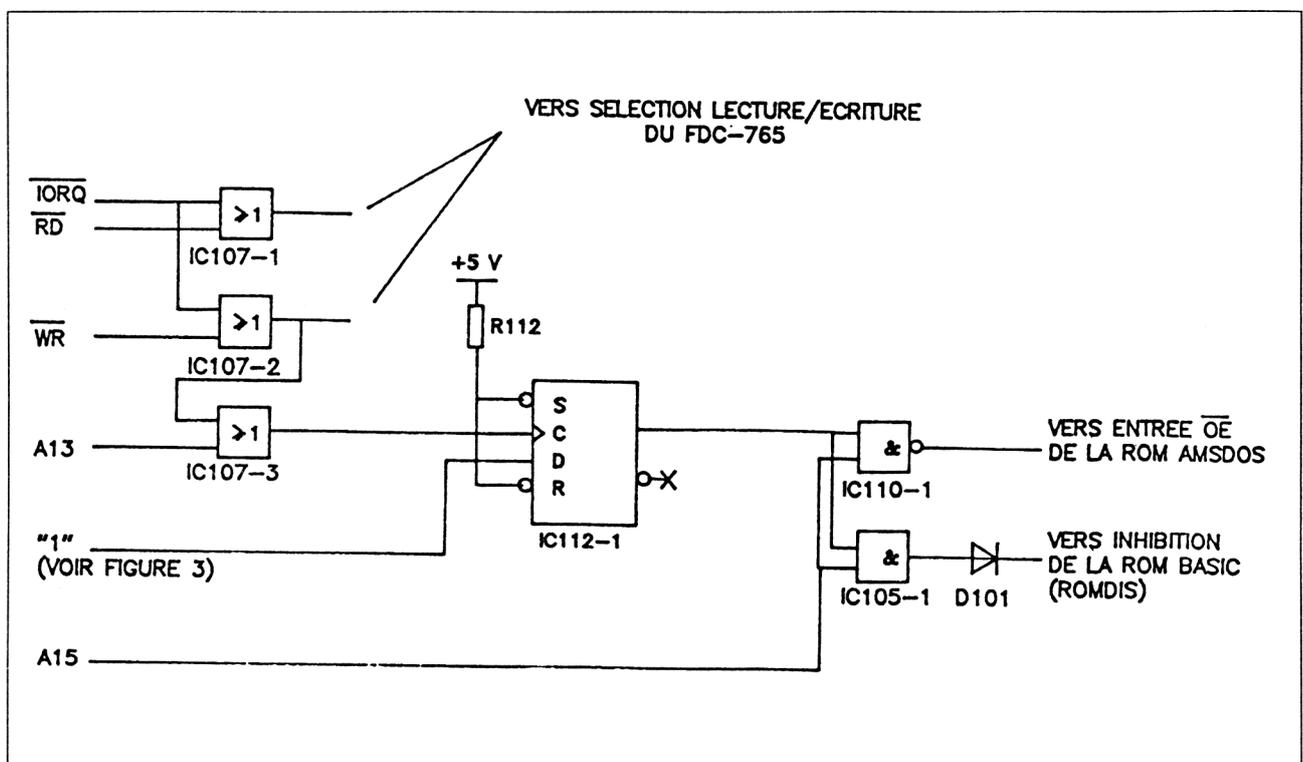


Fig. 4

Lors de l'écriture de la valeur &07 sur le port &FDXX, $\overline{\text{IORQ}}$ est actif, ainsi que $\overline{\text{WR}}$, donc tous deux à 0 ; ainsi, la sortie du OU à 2 entrées IC107-2 est positionnée à « 0 », niveau que l'on retrouve sur une des deux entrées du OU IC107-3.

Or, à cet instant, la deuxième entrée est positionnée aussi à « 0 » (A13 = « 0 »), condition idéale pour qu'apparaisse un niveau logique « 0 » en sortie, que l'on retrouve sur l'entrée d'horloge C de la bascule.

Si nous observons la table de vérité fournie en figure 1, la bascule n'est pas activée, car il lui faut un front montant sur C (ce qui est indiqué par le symbole de la flèche ascendante), c'est-à-dire une transition d'un niveau logique « 0 » au niveau logique « 1 », or nous venons de placer un niveau « 0 ».

Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

Par contre, lors de l'exécution de l'instruction suivante, le microprocesseur doit lire le code opération en mémoire, donc positionner au moins \overline{WR} au niveau logique « 1 » (et même \overline{IORQ}), ce qui placera un « 1 » logique en sortie de IC107-2, ainsi qu'en IC107-3 ; la bascule sera activée, et le niveau 1 de l'entrée D se retrouvera en sortie Q de IC112-1.

L'état de D peut maintenant être modifié, le niveau « 1 » de la sortie est mémorisé, sauf si l'on vient une nouvelle fois écrire sur l'adresse &FDXX, avec une autre valeur que &07.

Nous retrouvons donc ce niveau aux entrées des composants IC110-1 et IC105-1, en permanence, à la condition précédemment citée.

Ceci ne sélectionne pas encore la ROM AMSDOS, mais prépare cette sélection.

Supposons maintenant que le Z80 veuille lire (puisqu'on ne peut que lire en ROM) la ROM AMSDOS, pour y exécuter un programme : il doit d'abord placer le signal \overline{ROMEN} à l'état 0 (ROM ENable = Validation de ROM ; qui est connecté à toutes les ROMs en fonctionnement, et qui leur signale une éventuelle sélection). Ce signal arrive ainsi sur l'entrée de validation \overline{CE} de la ROM (ROM 40015, IC104 sur le schéma structurel de l'interface). Mais cela ne valide pas encore la possibilité de lire son contenu.

Toutes les ROMs supplémentaires doivent être placées de façon telle que leurs adresses soient comprises entre &C000 et &FFFF. On remarquera que pour ces adresses, les bits A14 et A15 sont tous deux à « 1 ». En reprenant le schéma, A15 (= « 1 ») est placé sur la deuxième entrée de IC110-1, ce qui donnera en sortie un niveau logique « 0 », donc validera l'autorisation, pour la ROM AMSDOS, de fournir le contenu d'une de ses adresses (activation de \overline{OE} = *Output Enable* = validation de sortie).

Un problème subsiste : il faut dévalider la ROM Basic, car si elle fournit une donnée en même temps, il risque d'y avoir conflit sur le bus de données, et éventuellement détérioration de composants.

Nous pouvons constater que le composant IC105-1 possède aussi ses deux entrées à l'état logique « 1 », donc sa sortie sera placée au niveau « 1 », par l'intermédiaire de la diode D101 (qui sera passante, à moins qu'une autre ROM n'ait pris le contrôle, ce qui est fort peu probable à notre avis). On trouve donc un niveau logique « 1 » sur la liaison ROM-DIS (ROM DISable = Dévalidation de ROM), qui est connectée sur l'entrée \overline{OE} de la ROM, contenant le système d'exploitation et la Basic (voir IC103 32 K BYTE ROM, sur le schéma structurel de l'unité centrale), permettant l'inhibition de toute valeur qu'elle pourrait fournir.

COMMENT BOOTER AUTOMATIQUEMENT ?

Comme nous l'avons remarqué dans le désassemblage du début de la ROM Basic, l'initialisation des ROMs par le ROM WALK n'est effectuée que par un saut à un sous programme (CALL 0BCCBH à l'adresse &C009), le programme revient inéluctablement en &C00C pour continuer

Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

l'initialisation et l'activation de l'interpréteur Basic. Ce qui ne nous arrange pas.

L'intérêt serait qu'après initialisation du Basic, soit effectuée automatiquement la commande CPM qui charge en mémoire le secteur BOOT d'une disquette système et place l'ordinateur sous le système d'exploitation CP/M 2.2, ou CP/M+, selon la disquette utilisée.

Deux aspects négatifs se présentent :

1) Nous avons laissé s'initialiser inutilement l'interpréteur Basic, donc perte de temps. Mais ce n'est pas le pire.

2) Cela implique une modification de la ROM, impossible directement, car cela obligerait à un dessoudage de cette ROM, à une modification du programme qu'elle contient, à une sauvegarde de ce programme sur EPROM, et à un remplacement sur la carte principale par cette EPROM. Opération Oh combien ! délicate et dangereuse : si votre ROM est détruite par le dessoudage, pas d'alternative : votre Amstrad est bel et bien hors service.

Pour comprendre un peu plus le Boot de CP/M, nous avons désassemblé la ROM AMSDOS :

```
C000  01      DEFB  01H
C001  00      DEFB  00H
C002  05      DEFB  05H
C003  00      DEFB  00H
```

Comme pour toutes les ROMs additionnelles, ces quatre premières adresses indiquent les différentes caractéristiques de la ROM, dans l'ordre :

- 01 signale une ROM de premier plan,
- 00 indique le numéro de ROM,
- 05 signale le numéro de version,
- 00 indique le niveau de modification;

Les deux octets suivant contiennent l'adresse du nom de la ROM, qui est de plus l'adresse de la table des différents noms des RSXs utilisables sous Basic, lorsque AMSDOS est initialisée :

```
C004  72C0      DEFW  0C072H
```

A partir de l'adresse suivante (&C006), se trouve toute la table de sauts aux différentes routines de traitement des instructions résidentes et en lien avec la table des noms :

```
C006  C3  BC  C1  JP  0C1BCH
C009  C3  B2  C1  JP  0C1B2H
C00C  C3  D1  CC  JP  0CCD1H
.....
C02D  C3  8A  D4  JP  0D48AH
C030  C3  C4  D4  JP  0D4C4H
.....
```

Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

La table des noms se présente ainsi :

C072	43	50	4D	DEFB	« CPM »
C075	20	52	4F	DEFB	« RO »
C078	CD			DEFB	« M » + 080H ;CPM ROM
C079	43	50		DEFB	« CP »
C07B	CD			DEFB	« M » + 080H ;CPM
C07C	44	49	53	DEFB	« DIS »
C07F	C3			DEFB	« C » + 080H :DISC
.....				
C0B0	45	52		DEFB	« ER »
C0B2	C1			DEFB	« A » + 080H ;ERA
C0B3	52	45		DEFB	« RE »
C0B5	CE			DEFB	« N » + 080H ;REN
.....				

Un nom nous a surpris : « CPM ROM », qui n'est pas accessible sous Basic par la commande !CPM ROM qui a le fâcheux inconvénient de répondre par **syntax error**.

Qu'à cela ne tienne, nous avons procédé à son désassemblage qui fut fort intéressant.

Son traitement débute à l'adresse donnée par le premier saut de la table des vecteurs en &C006 (comme par hasard, une adresse dont on a déjà parlé dans le ROM Walk). Ce vecteur nous emmène en &C1BC :

C1BC	38	06		JR	C,0C1C4H
C1BE	CD	12	B9	CALL	0B912H
C1C1	B7			OR	A
C1C2	28	18		JR	Z,0C1DCH
C1C4	FD	E5		PUSH	IY

En &C1BC, l'indicateur de carry est testé (JR C,0C1C4H) pour initialisation de la ROM (à partir de &C1C4).

Dans le cas contraire, cet indicateur est nul. On effectue donc un branchement au vecteur RAM &B912 (CALL 0B912H). Ce vecteur (voir Partie 4, chap. 2.7, page 70) permet de connaître le numéro de la ROM actuellement utilisée. Au retour, il fournit dans A l'adresse de la ROM. Il n'effectue, en fait, qu'une lecture de la case mémoire &B8D6, qui contient le numéro de la précédente ROM sélectionnée.

Si cette ROM est zéro (on positionne l'indicateur grâce à OR A), alors sauter en 0C1DC, sinon continuer l'initialisation de la ROM (JR Z,0C1DCH).

Et comme par enchantement, le fait de sauter à l'adresse &C1DC permet de charger et lancer à froid CP/M.

Rappelons que le ROM Walk est effectué par un saut à l'adresse &C006 de la ROM numéro 0. Donc, si la ROM AMSDOS est positionnée comme ROM numéro 0, alors, CP/M est automatiquement Bootté.

Dernier problème : comment faire croire au Z80 que la ROM AMSDOS porte le numéro &00 ?

MODIFIONS LE SCHÉMA STRUCTUREL

Notre problème est de modifier l'adresse de sélection de la ROM AMS-DOS à la valeur 0.

Rappelons-nous que la sélection d'une ROM s'effectue grâce à la sortie du numéro de ROM sur le port &DFXX. Il faut donc que la ROM AMS-DOS soit sélectionnée lorsque l'on effectuera un OUT (C), C après une instruction LD BC,00DF..

Par rapport à la sélection de la ROM en tant que numéro 7, la différence joue sur les 3 bits de poids faibles du bus de données :

	D7	D6	D5	D4	D3	D2	D1	D0
00 =	0	0	0	0	0	0	0	0
07 =	0	0	0	0	0	1	1	1

Il faut donc que la ROM soit sélectionnée lorsque D2, D1, D0 sont tous trois au niveau logique zéro ; c'est la seule modification qui se produit, aussi bien sur le bus de données que sur le bus d'adresses où aucune modification n'intervient.

Reprenons le schéma structurel de la figure 3.

Si la valeur binaire D7D6D5D4D3D2D1D0 = 00000000 se trouve sur le bus de données, les trois opérateurs logiques IC109-1, IC109-2 et IC109-3 voient leurs sorties placées au niveau logique « 0 », ce qui ne sélectionne pas notre ROM.

Pour la sélection, il faut absolument que ces sorties soient considérées à « 1 » grâce à R111, donc que leurs modèles soient équivalents à celui de droite sur la figure 2, c'est-à-dire la sortie en l'air.

Or, que faut-il pour que cette situation se produise ? Il faut que les trois entrées, qui sont connectées à la masse par l'intermédiaire de LK1, soient plutôt au niveau logique « 1 ».

En étudiant le schéma structurel de l'unité centrale, nous avons remarqué que les trois entrées sont connectées à \overline{EXP} , lui-même relié au bit b5 du port B du 8255 (IC107 sur le schéma, en haut à gauche). Mais de plus, cette broche est amenée au +5 V (niveau « 1 » logique), par l'intermédiaire d'un des éléments résistifs d'un réseau de résistances repéré NR101. Ce rappel est nécessaire lorsque aucune extension n'est connectée sur le port (surtout dans le cas du CPC-464).

Soit, nous allons utiliser cet élément résistif pour amener un niveau « 1 » sur les trois entrées des OU EXCLUSIF : il suffit d'éliminer le rappel à la masse, donc de couper LK1.

LK1 éliminé, l'adresse de sélection de la ROM AMSDOS devient zéro, et, comme nous l'avons étudié précédemment, lorsque l'on voudra accéder à une adresse supérieure ou égale à &C000, la ROM Basic sera automatiquement dé-sélectionnée.

III. Le boot sur lecteur A - Réalisation pratique

Après toutes ces explications, qui vous ont initié à l'étude d'un schéma d'une réalisation micro-informatique, qui vous ont révélé l'entrelacement à la fois du logiciel (Software) et du matériel (Hardware), nous allons vous expliquer comment réaliser cette opération toute simple, mais tout de même délicate, qui est d'éliminer la liaison LK1.

D'ABORD OUVRIR L'AMSTRAD CPC-6128

Cette toute première opération est nécessaire puisque la modification doit s'effectuer sur la carte principale du CPC-6128.

Déconnectez les alimentations 5 volts et 12 volts ainsi que le connecteur vidéo de l'écran de visualisation.

Retournez le CPC, clavier contre votre plan de travail, que vous prévoirez large, et munissez-vous du tournevis cruciforme.

Dévissez et enlevez toutes les vis que vous apercevrez (normalement six).

Observez maintenant la face sur le côté du lecteur de disquettes, sur laquelle doivent se trouver deux vis qu'il vous faudra également ôter.

Vous pouvez maintenant séparer partiellement le clavier du fond, selon les indications de la figure 5, en veillant à ne pas déconnecter le clavier — le câble le reliant à l'unité centrale doit rester solidaire.

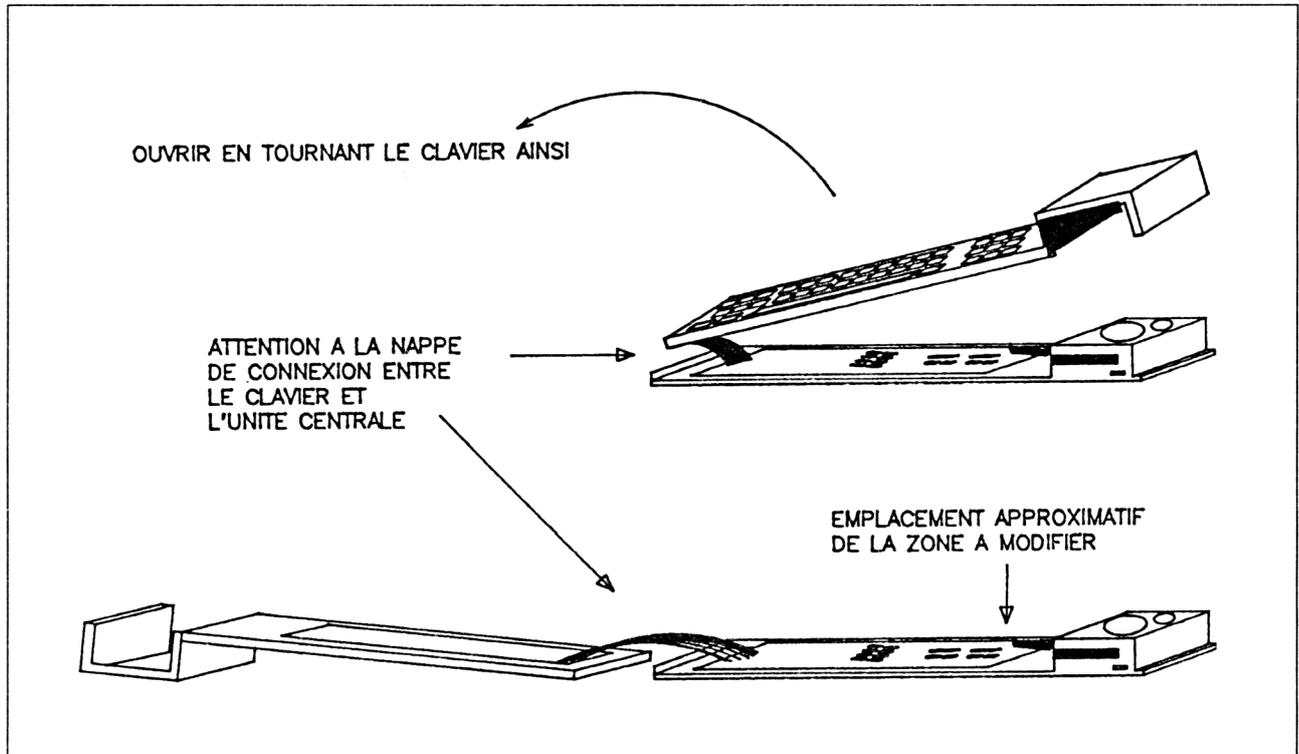


Fig. 5

ENSUITE MODIFIER LA CARTE PRINCIPALE

Observez maintenant la partie droite de la carte principale, celle la plus proche du lecteur de disquettes. Vous y découvrirez un câble en nappe qui se connecte à l'arrière du lecteur de disquettes.

Repérez-vous maintenant à l'aide de la figure 6.

Sous ce câble se trouve le composant intégré 2 fois 20 broches FDC-765, qui effectue la plus grande tâche de gestion des données à enregistrer sur la disquette, en compagnie du séparateur de données FDC-9216 (2 fois quatre broches) placé au-dessous à droite.

Au-dessous encore, vous trouverez deux composants intégrés (74LS161, deux fois huit broches, et à gauche : 74LS74, deux fois sept broches).

A gauche du 74LS74, qui n'est autre que la bascule D, se trouve un condensateur (il ressemble à une tête de punaise, de couleur marron, et possède deux pattes soudées sur la carte imprimée).

Juste à gauche de ce condensateur, vous pourrez apercevoir un petit trait blanc sérigraphié, avec légèrement sur le côté, le repère LK7. Ce repère correspond en fait à la liaison LK1 donnée sur le schéma structural de l'interface lecteur de disquettes. (Ce nom LK1 est probablement utilisé sur l'interface DDI-1 du CPC-464).

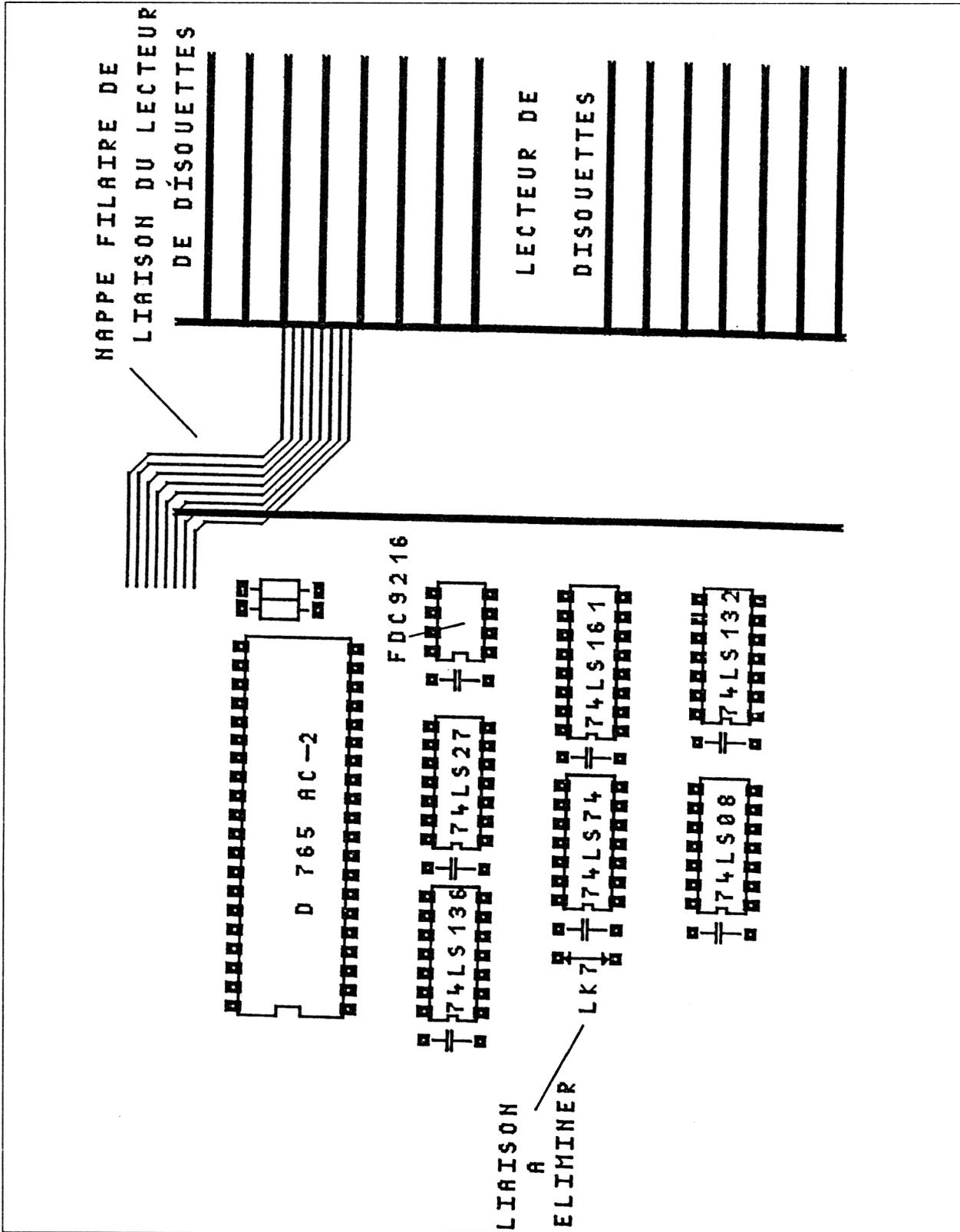


Fig. 6

Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

Sous le trait blanc, qui se trouve entre deux pastilles de circuit imprimé, se trouve une piste de cuivre (assez peu visible).

Munissez-vous de votre cutter et commencez à couper cette piste avec de grandes précautions.

Après avoir entaillé la piste (n'allez pas trop profondément), nous allons procéder à un essai.

Soufflez très fort, vers l'extérieur, à l'endroit où vous venez de travailler, afin d'écarter tout copeau de cuivre risquant d'occasionner un court-circuit.

Remplacez le clavier sur l'unité centrale sans remonter les vis, et effectuez les branchements à l'écran de visualisation. Mettez sous-tension, sans placer la disquette dans le lecteur.

Après quelques secondes, le message suivant doit s'afficher :

**Drive A: disc missing
Retry, Ignore or Cancel ?**

Si, par contre, vous voyez **BASIC 1.1**, cela signifie que la ROM Basic a été prise en compte, donc que vous n'avez pas entaillé suffisamment la piste. Débranchez l'unité centrale, ouvrez de nouveau et recommencez à couper la piste.

Dès que le message attendu s'inscrit, l'ordinateur est prêt pour Booter CP/M.

Mettez hors tension, placez votre disquette CP/M (une copie suffira si vous en avez), et mettez de nouveau sous tension.

Quelques instants plus tard, vous verrez apparaître le prompt **A>** signalant que vous êtes sous CP/M. Vous pouvez maintenant utiliser tout programme fonctionnant sous ce système d'exploitation.

IV. Retourner sous Basic

Vous pouvez légitimement vous poser la question : comment puis-je de nouveau retourner sous Basic ? (Car vous avez peut-être remarqué que même un RESET par <SHIFT> <CONTROL> <ESC> provoque un nouvel accès disquette).

Deux façons de retrouver le Basic vont donc se présenter : soit retourner sous Basic à partir de CP/M, soit ré-initialiser l'Amstrad après avoir déplacé la position d'un interrupteur que nous allons installer.

LE BASIC PAR CP/M

Si vous avez décidé que CP/M est le seul et vrai système d'exploitation de votre Amstrad (comme MS/DOS en est un pour les IBM-PC), la modification est alors définitive, et vous pensez que l'utilisation de Basic s'effectuera comme l'utilisation d'un Basic sur PC (GWbasic, par exemple), en le lançant à partir du système d'exploitation : le C-Basic est un Basic fonc-

Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

tionnant sous CP/M, performant et, de plus, compilable. A moins que vous ne préfériez l'utilisation du Basic Locomotive toujours présent en ROM.

Frappez donc la commande

A> AMSDOS <RETURN >

(si AMSDOS.COM se trouve sur votre copie), et vous retrouverez, après effacement de l'écran, le message Basic 1.1. Vous pouvez, de plus, retourner de nouveau sous CP/M grâce à la commande ;CPM.

Nous avons décidé de vous indiquer comment le programme AMSDOS.COM vous ramène sous Basic, puisque la ROM Basic n'a pas été reconnue, et ce à l'aide du désassemblage de ce court programme :

```

0100      0E 0C      LD      C,00CH
0102      CD 05 00  CALL    00005H
0105      7D        LD      A,L
0106      F3        DI
0107      01 8E 7F  LD      BC,07F8EH
010A      D9        EXX
010B      08        EX      AF,AF'
010C      AF        XOR     A
010D      08        EX      AF,AF'
010E      11 00 00  LD      DE,0000H
0111      FE 30     CP      030H
0113      38 0F     JR      C,0124H
0115      11 00 40  LD      DE,04000H
0118      01 C4 7F  LD      BC,07FC4H
011B      21 ED 49  LD      HL,049EDH
011E      22 22 41  LD      (04122H),HL
0121      C3 22 41  JP      04122H
0124      31 00 C0  LD      SP,0C000H
0127      3E 79     LD      A,0C9H
0129      32 38 00  LD      (0038H),A
012C      D5        PUSH   DE
012D      CD C8 BC  CALL    0BCC8H
0130      D1        POP    DE
0131      21 39 01  LD      HL,0139H
0134      19        ADD    HL,DE
0135      CD D4 BC  CALL    0BCD4H
0138      C7        RST    00H
0139      42 41 53 49  DEFB   "BASI"
013D      C3        DEFB   "C" + 080H

```

Ce programme, comme la plupart des programmes sous CP/M, débute dans la TPA à l'adresse &0100.

Le registre C y est chargé avec le numéro de la fonction 12 (LD C,0CH) qui est appelée dans le BDOS (CALL 0005H).

Cette fonction retourne le numéro de version que l'on place dans A (LD A,L).

Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

Les interruptions sont inhibées (DI), on prépare BC avec une configuration du *Vidéo Gate Array* (LD BC,07F8EH), que l'on sauvegarde par échange avec les registres auxiliaires (EXX et EX AF,AF'). A et les *Flags* sont initialisés à zéro (XOR A), et on les échange pour récupérer le numéro de version (EX AF,AF').

DE est initialisé à zéro (LD DE,0000H), et le numéro de version est testé par rapport à la version 3.0 (CP 030H). Si ce n'est pas le cas (c'est donc CP/M 2.2 qui est utilisé), on saute à l'adresse de traitement en &0124 (JR C,0124H).

Dans le cas de CP/M 3.0 (ou CP/M+), le registre DE est initialisé au début de la zone de RAM commutable (LD DE,04000H), BC avec une valeur permettant au *Gate Array* de commuter la RAM (LD BC,07FC4), et HL avec les deux octets d'une instruction de sortie sur le port adressé par BC (LD HL,049ED → ED 49 = OUT (C), C), que l'on place à l'adresse 04122H (LD (04122H), HL), et à laquelle on saute aussitôt (JP 04122H). A partir de cette adresse, la version de CP/M+, grâce à C10CPM3.EMS, s'occupe de ré-initialiser la ROM Basic et de s'y brancher.

A l'adresse &0124, commence la routine de traitement de l'initialisation de la ROM Basic par CP/M 2.2.

Le pointeur de pile est initialisé (LD SP,0C000H).

On place le code RET à l'adresse d'interruption RST 7 (LD A,0C9H - LD (0038H),A).

DE est sauvegardé dans la pile (PUSH DE), et l'on appelle le vecteur permettant de réinitialiser le KERNEL à sa valeur par défaut (CALL 0BCC8H → vous trouverez des explications sur ce vecteur Partie 4, chapitre 2.7).

On récupère la valeur de DE (POP DE → normalement &0000). HL est chargé avec l'adresse 0139H afin de pointer une zone contenant le nom de l'instruction Basic codée sous forme de RSX (LD HL,0139H - ADD HL,DE). Ceci permettant de rechercher dans toutes les ROMs l'instruction précédemment citée, et de quitter CP/M par un ReStart 0 (RST 00).

Nous voyons donc que pour quitter CP/M, cela est effectué par la recherche de l'instruction : BASIC du BASIC LOCOMOTIVE. Or, dans le cas du CPC-464, l'exécution de cette instruction initialise le Basic 1.0, ce qui fait perdre tous les vecteurs du lecteur de disquettes, et les remplace par ceux du lecteur de cassettes.

LE BASIC PAR LE MATÉRIEL

Nous allons vous expliquer ici comment vous pourrez retourner au Basic par une méthode beaucoup plus rapide que celle décrite précédemment, mais qui demande quelque dextérité quant au maniement du fer à souder.

Pour assurer le BOOT automatique de CP/M, nous avons coupé une piste de la carte imprimée. Aussi, pour retrouver votre CPC sous Basic, il suffit de rétablir le contact perdu, mais pas définitivement : à l'aide d'un interrupteur.

Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

Placez donc votre fer à souder sur son support et mettez-le en chauffe (rappelons que la puissance de celui-ci doit être de l'ordre d'une trentaine de watts maximum, sous peine de détériorer le circuit).

Pendant la chauffe, dénudez vos deux morceaux de fil d'environ 60 cm, et ouvrez de nouveau les entrailles de votre Amstrad-CPC.

Une fois le fer chaud, étamez-le et essuyez sa panne avec une éponge humide. Etamez aussi le cuivre apparent sur les câbles dénudés (au besoin recoupez-les pour ne laisser qu'environ 2 millimètres de cuivre).

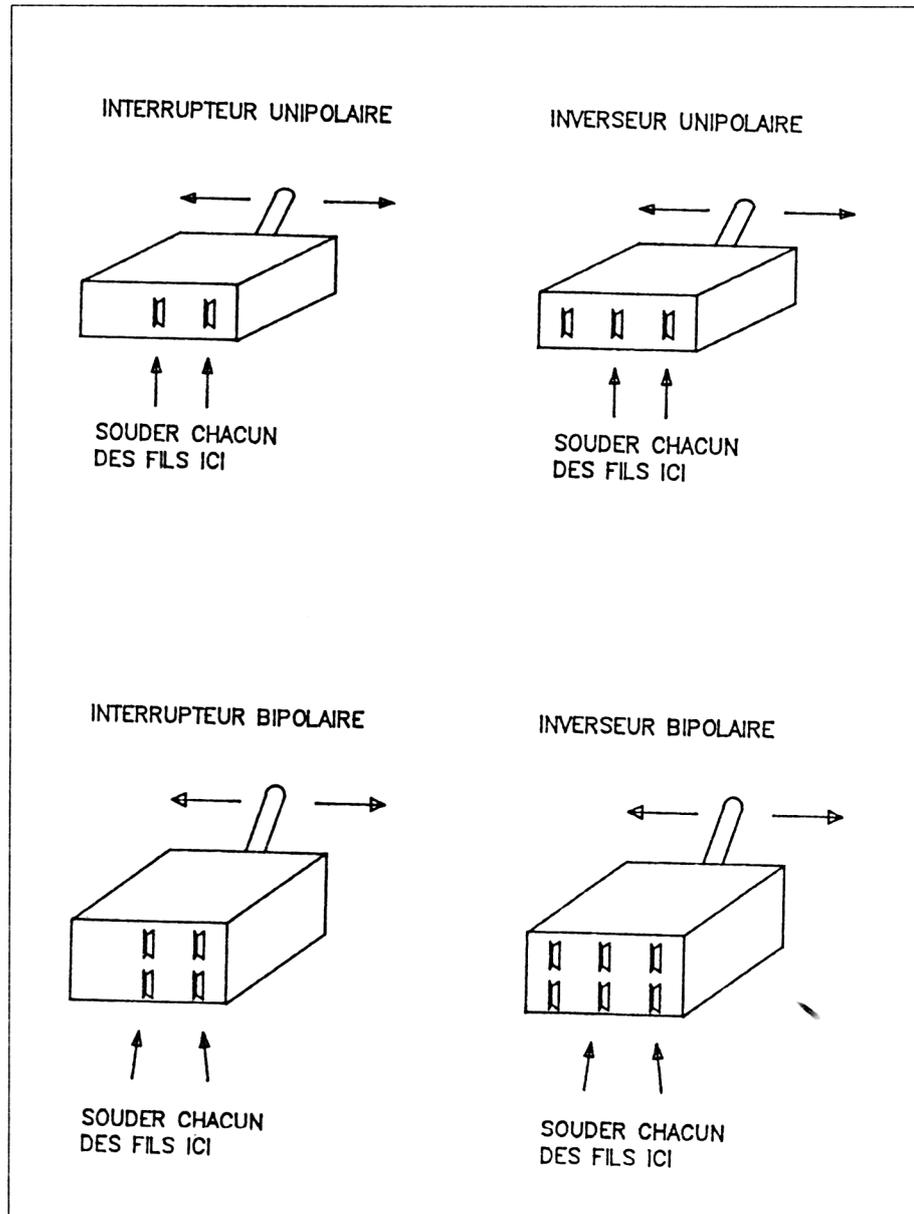


Fig. 7

Repérez de nouveau la liaison que vous avez éliminée. De chaque côté de celle-ci, vous trouverez une pastille métallisée (voir figure 6). Sur la pastille supérieure, vous soudez un côté du fil blanc, et sur la pastille inférieure, un côté du fil gris.

Vous soudez ensuite les autres extrémités, chacune sur une broche de l'interrupteur unipolaire (voir schéma de connexion selon différent type d'interrupteurs en figure 7).

Ecartez votre fer à souder pour éviter tout incident, et commencez à percer le capot plastique de l'Amstrad pour y implanter l'interrupteur. Choisissez un endroit où il n'y aura pas possibilité de court-circuit avec la plaque imprimée, de préférence vers le haut.

L'interrupteur fixé, refermez le boîtier sans le revisser, et procédez aux essais : selon la position de l'interrupteur, vous aurez soit le BOOT automatique, soit l'initialisation du Basic. Repérez aussitôt chaque position à l'aide d'une étiquette auto-collante sur le capot.

Si votre CPC s'obstine à BOOTer ou à initialiser le Basic, alors il vous faudra vérifier vos soudures ou la connexion sur l'interrupteur.

V. Lancer un programme automatiquement

AVEC CP/M 2.2

Formatez une disquette vierge au format SYSTEME avec l'utilitaire DISKIT2 de l'original de votre disquette CP/M 2.2. Profitez-en pour copier l'utilitaire SETUP.COM à l'aide de PIP.COM ou FILECOPY.COM, ainsi que le programme à exécuter.

Placez votre copie et effectuez un CONTROL C pour faire reconnaître la disquette, puis lancez SETUP.

A la toute première question :

***** Initial command buffer empty
Is this correct (Y/N)**

répondez **N(No)** et entrez le nom du programme à lancer, au message :

Enter new initial command buffer

Répondez ensuite **Y (Yes)** aux quatorze autres questions (à moins que vous ne désiriez y effectuer une modification — dans ce cas reportez-vous à la Partie 3, chapitre 3, pages 17 et suivantes, pour avoir plus d'explications).

Voilà, votre disquette est prête à exécuter automatiquement le programme de votre choix dès la mise sous tension (si vous n'avez pas oublié de basculer le commutateur en position BOOT automatique).

AVEC CP/M 3.0 OU CP/M +

Pour CP/M 3.0, la démarche est quelque peu différente.

Effectuez le formatage d'une disquette vierge au format SYSTEME. Copiez aussitôt, à l'aide de PIP.COM, les utilitaires C10CPM3.EMS, SUBMIT.COM et ED.COM.

Placez votre copie et lancez ED.

Au message :

Enter input file

répondez **PROFILE.SUB <RETURN>**, puis appuyez sur **<RETURN>** pour répondre à la question :

Enter output file

A l'apparition de l'étoile (*), frappez I pour voir apparaître le premier numéro de ligne (1:). Entrez alors le nom du programme à exécuter et validez. Frappez ensuite **<CONTROL> Z**, puis **E** suite à l'étoile pour quitter ED en sauvegardant le fichier PROFILE.SUB.

Vous voici en possession d'une disquette de lancement automatique du programme de votre choix sous CP/M +.

VI. Booter sur un lecteur additionnel

Ce paragraphe s'adresse à ceux d'entre vous qui ont fait l'acquisition, ou installé un lecteur supplémentaire sur DRIVE B, mais avec comme capacité, au formatage de 169 K-octets, ou 178 K-octets maximum.

En général, l'acquisition de ce type de lecteur s'est portée sur le choix 5 pouces 1/4, et il est pécuniairement avantageux d'avoir effectué les copies de sauvegarde sur ce format.

Le problème, dans le cas de CP/M, est que le secteur BOOT est recherché sur la piste 0 du lecteur A. Alors, comment lancer automatiquement CP/M sur le lecteur B ?

A moins de modifier la ROM, ce à quoi nous n'avons pas tenu pour l'instant, la possibilité que nous avons envisagée consistera à faire croire au CPC que le lecteur additionnel se trouve à la place du lecteur A ; le lecteur 3 pouces d'origine devenant ainsi lecteur B.

La modification que nous avons apportée se trouve donc au niveau matériel.

LE PRINCIPE

Reprenez le schéma structurel de l'interface disquette. En haut, à droite, se trouvent deux opérateurs logiques IC106-1 et IC106-2 (74LS38 ou 7400) qui permettent, grâce à la sortie US0 du FDC-765, de sélectionner soit le lecteur A, soit le lecteur B, leurs sorties se retrouvant sur le connecteur en broches 23 et 25, appelées DRIVE SELECT1 et DRIVE SELECT2.

Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

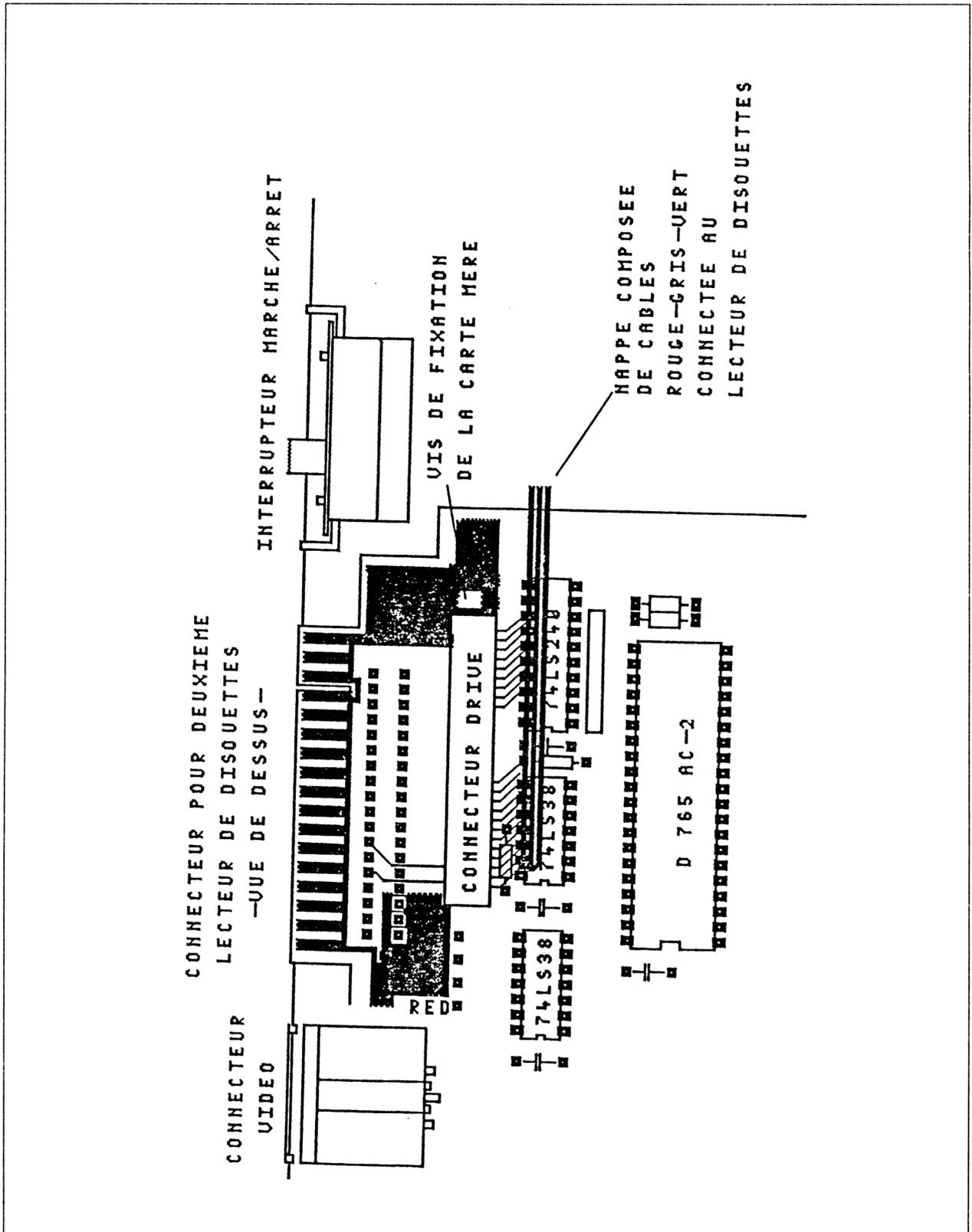


Fig. 8

Il suffira donc d'inverser ces sorties pour échanger les noms des lecteurs. Bien sûr, nous n'allons pas effectuer cette modification définitivement : nous vous proposerons un montage filaire à l'aide d'un inverseur bipolaire permettant de revenir à la configuration originale.

LA MODIFICATION

Cette fois-ci, la modification à effectuer demande plus de minutie, car plus de coupures de liaisons et de soudage. Si vous ne vous sentez pas à l'aise, préférez demander à un spécialiste de l'électronique.

Ouvrez de nouveau votre Amstrad et munissez-vous de la figure 8, pour repérer les composants.

Vous repérerez plus particulièrement les deux lignes de pastilles, près du connecteur disquette, le composant 74LS38 le plus à gauche, et le connecteur DRIVE d'où sort une nappe filaire de câbles colorés (le câble numéro 1 est le plus à gauche, ensuite on trouve un câble vert tous les cinq fils, tous les autres étant gris).

Munissez-vous maintenant de la figure 9 représentant les deux rangées de pastilles, ainsi que de votre cutter. Commencez à sectionner la piste que l'on voit se connecter sur la pastille numérotée 7.

Si votre fer à souder est prêt, vous pouvez souder le fil violet (préalablement dénudé !) sur la pastille ainsi isolée.

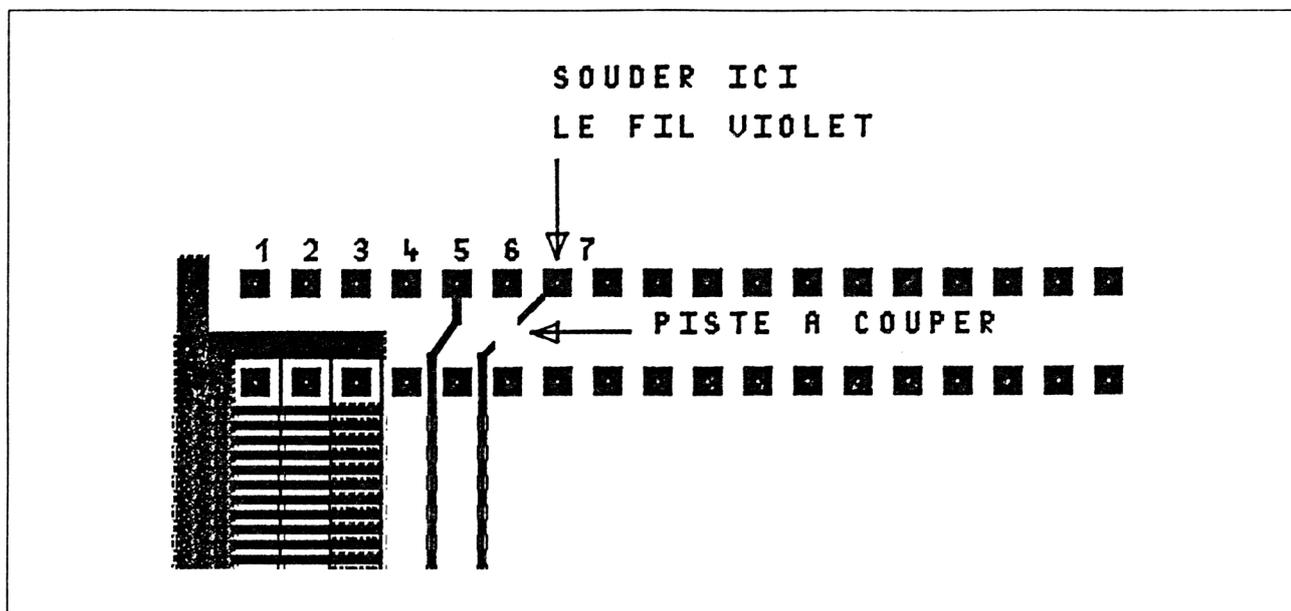


Fig. 9

La liaison que vous venez de supprimer se dirigeait en broche 8 du composant IC106-2, le 74LS38 ou 7400 (il s'agissait de DRIVE SELECT1).

A l'aide de la figure 10, repérez le composant 74LS38, le plus à gauche, et soudez le fil bleu sur la broche indiquée (cette opération est délicate,

Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

car il faut veiller à ne pas trop chauffer, sinon vous risqueriez de détériorer le composant intégré).

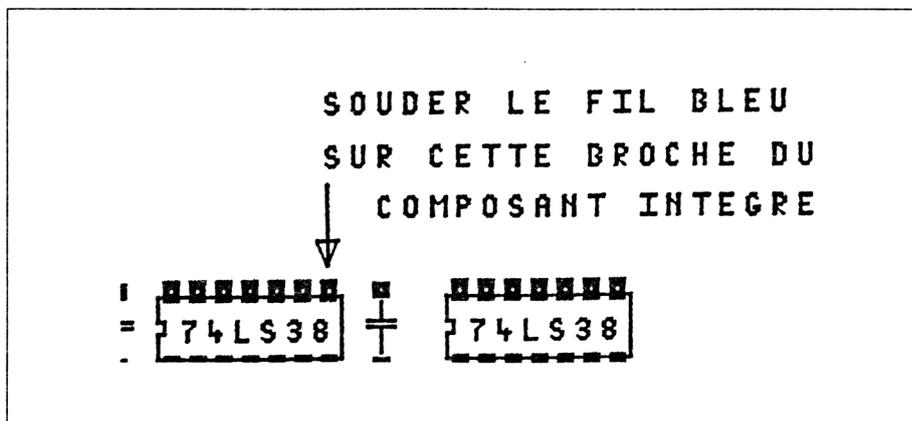


Fig. 10

Passons maintenant à la modification de la ligne de connexion DRIVE SELECT 0. Repérez le connecteur DRIVE d'où sort la nappe de fils (voir figure 11), et sectionnez à quelques centimètres le quatrième fil à partir de la gauche (normalement un fil gris situé juste avant un fil vert).

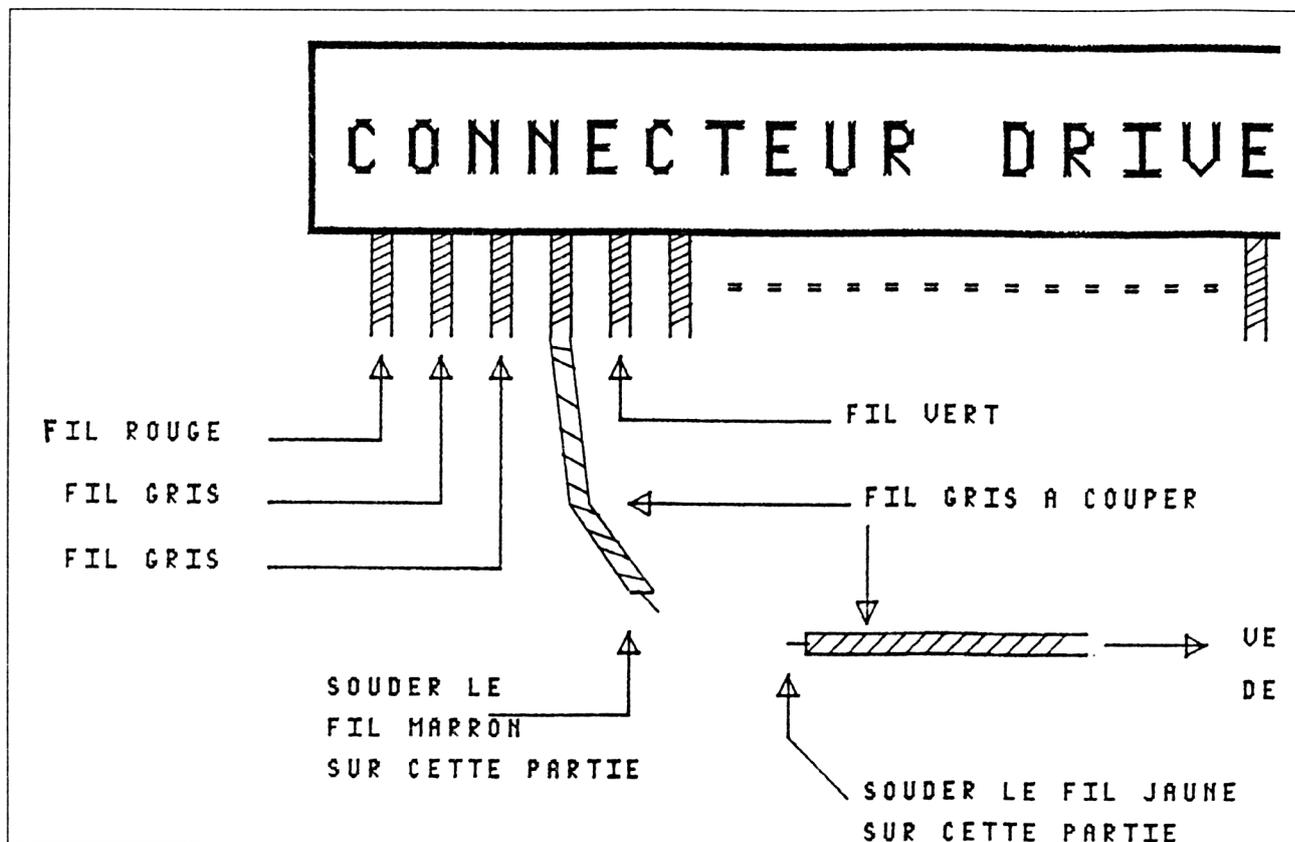


Fig. 11

Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

Sur les deux bouts que vous dénuderez, vous pourrez souder comme indiqué le fil jaune et le fil marron - soudures que nous vous conseillons d'isoler aussitôt. Si vous possédez des tout petits dominos, vous pouvez y fixer les fils plutôt que les souder.

Munissez-vous maintenant de l'inverseur bipolaire. Comme représenté en figure 12, commencez par souder une chute de fil préalablement dénudé entre les broches repérées 1 et 6 sur la figure, ainsi qu'une autre entre les broches 2 et 5. Vous pourrez ensuite souder les fils bleu, violet, marron et jaune successivement sur les broches 3, 5, 4 et 6.

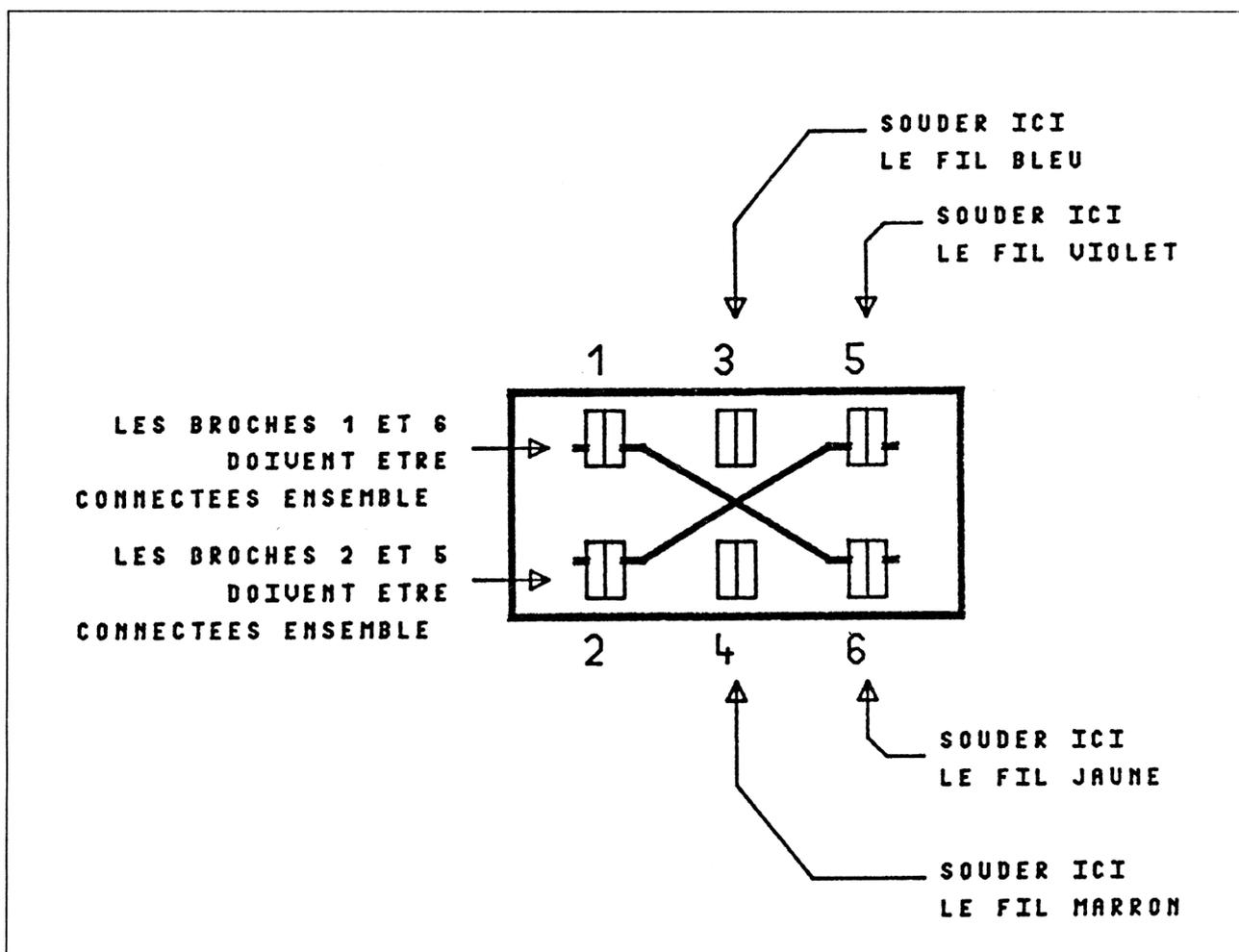


Fig. 12 : Connexions à réaliser sur l'interrupteur. Vue côté broches.

Après vérification de l'état de toutes vos soudures, monter l'interrupteur sur le capot de l'Amstrad en perçant à un endroit approprié (il reste de la place, soyez en sûr). Finissez à la lime fine.

Voyez si aucun court-circuit ne risque de se produire en refermant le capot. Soufflez bien fort sur la carte, et procédez au remontage du capot.

Mettez sous tension le deuxième lecteur et l'ordinateur, placez une disquette connue dans chacun des lecteurs, et effectuez une instruction CAT (le commutateur AUTOBOOT en position Basic).

Selon la disquette dont le catalogue sera affiché, repérez la position du commutateur à l'aide d'un morceau de papier autocollant sur lequel vous noterez le nom du lecteur actif. Procédez à une vérification en basculant l'interrupteur sur l'autre position.

Signalons que cette modification est aussi utilisable sans BOOT automatique. Par contre, si vous êtes munis d'un lecteur VORTEX, vous ne pourrez pas utiliser ce lecteur à ses pleines performances, c'est-à-dire en 720 K-octets formatable avec la ROM VORTEX. Il vous faudra déconnecter cette ROM.

Nous allons, dans le paragraphe suivant, vous indiquer la procédure à suivre pour BOTTER automatiquement sur ce lecteur en 720 K-octets.

VII. Utilisation avec lecteur Vortex

En démontant la ROM livrée avec le lecteur VORTEX pour CPC-6128, nous avons remarqué qu'il était possible de modifier, à l'aide de strap, le numéro de celle-ci. Mais tous les essais que nous avons effectués (en la plaçant d'office en ROM 0), se sont révélés infructueux : en effet, le vecteur placé en &C006 de cette ROM n'effectue pas automatiquement le démarrage à froid escompté.

Par contre, il est possible par une astuce logicielle de Booter automatiquement en faisant considérer à l'Amstrad que le lecteur VORTEX est le lecteur A, configuré en 720 K-octets, le lecteur 3 pouces étant lecteur B, avec une capacité de 169 ou 179 K-octets. Mais, revers de la médaille, nous n'avons réussi cet exploit que sous CP/M 3.0, ce qui n'est déjà pas si mal, tout de même !

Vous n'aurez à ce moment plus besoin du commutateur DRIVE A / DRIVE B, qui sera positionné sur DRIVE A, l'autre position étant formellement déconseillée (elle considérerait le lecteur 3 pouces comme étant double tête et pouvant formater en 720 K-octet, matériellement impossible).

Voyons auparavant comment BOOTER sous CP/M+ sur le lecteur 3 pouces considéré en lecteur A, ce qui peut-être utile avec certaines applications comprenant le programme sur lecteur 3 pouces, accédant à de très gros fichiers sur le lecteur B.

LECTEUR 3 POUCES = A – LECTEUR VORTEX = B

Vous allez créer une disquette 3 pouces BOOTable sous CPM+ en utilisant la procédure décrite en Partie 8, chapitre 6.1, page 6, à la différence que vous répondrez Oui à la question **Sauvegarde sur disquette 3 pouces**.

Partie 10 : Fabrication de circuits additionnels pour AMSTRAD

Vous serez alors en possession d'une disquette BOOTable au format VORTEX, sur laquelle vous pourrez installer les logiciels que vous désirerez et fonctionnant sous CP/M+.

Si vous voulez lancer automatiquement un programme, reportez-vous au paragraphe : *Avec CP/M 3.0 ou CP/M+*.

LECTEUR 3 POUCES = B - LECTEUR VORTEX = A

Pour que l'Amstrad considère que le lecteur VORTEX est le lecteur A, cette fois-ci la procédure est un peu plus complexe.

Placez-vous d'abord sous le Basic VDOS, et formatez une disquette 3 pouces au format système CP/M+ à l'aide de l'utilitaire DISCKIT3. Inutile de recopier le fichier C10CPM3.EMS, nous allons utiliser le fichier CPM3.EMS de la disquette VORTEX.

Utilisez ensuite dans le lecteur VORTEX une disquette BOOTable sous CPM/+ et contenant une copie de PIP.COM (voir Partie 8, chapitre 6.1, pages 6 et 7, comment réaliser cette disquette).

Frappez, toujours sous Basic, la commande :

```
:CPM,2
```

CP/M+ se charge et considère le lecteur VORTEX en tant que lecteur A. Lancez alors l'utilitaire PIP et, à l'apparition de l'étoile (*), frappez la commande suivante :

```
B = A:CPM3.EMS
```

qui effectuera la copie du fichier de configuration de CP/M 3.0, CPM3.EMS, sur la disquette 3 pouces (ce fichier configure le lecteur VORTEX en tant que lecteur A, et initialise la ROM VORTEX).

Vous voilà en possession d'une disquette 3 pouces qui vous permettra de lancer automatiquement un programme de lecteur VORTEX considéré comme lecteur A. Il vous faudra, hélas, posséder la disquette 3 pouces qui effectue le BOOT préalable, et qui renverra sur le lecteur supplémentaire !