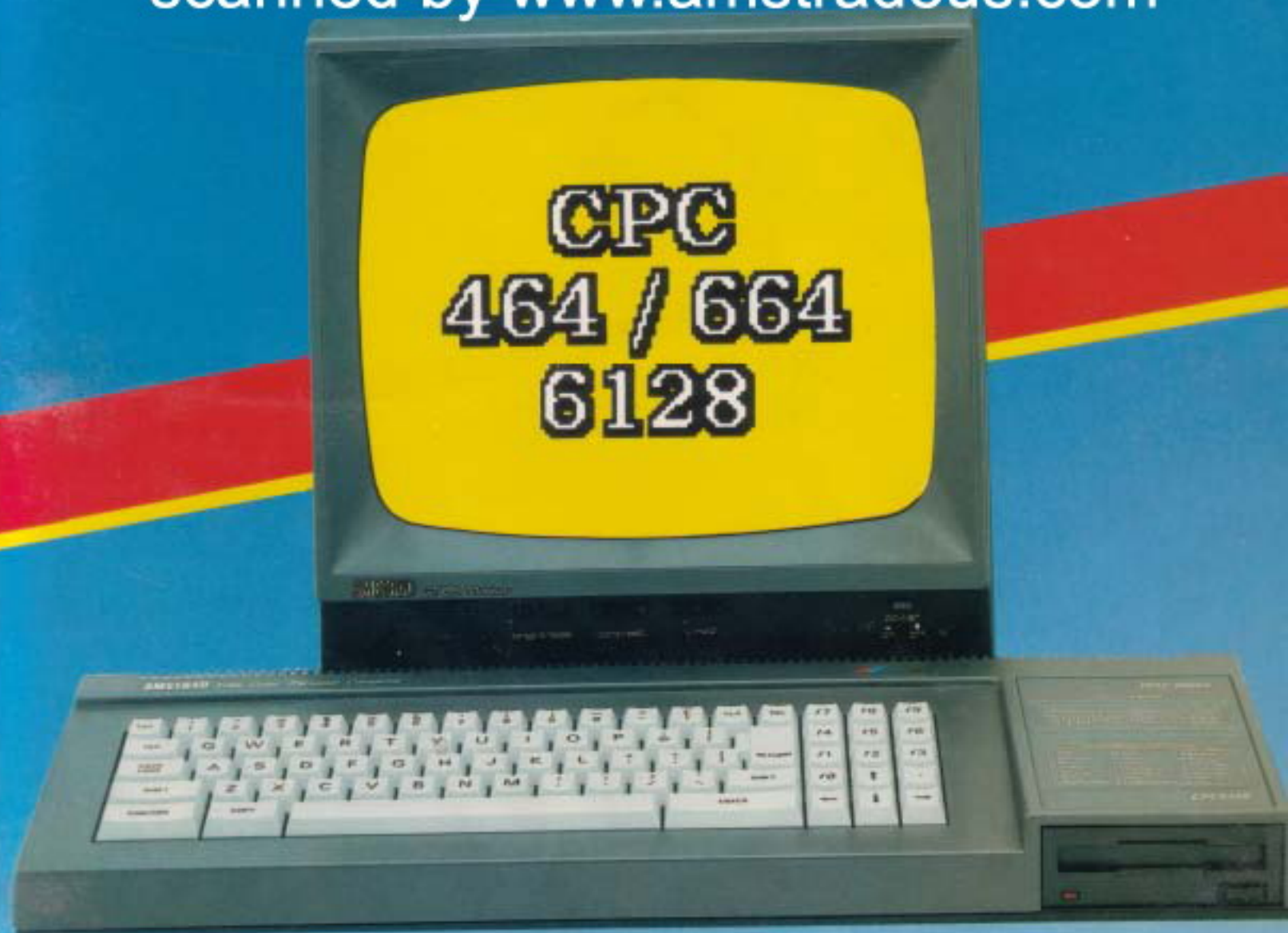


Comment exploiter
toutes les ressources
et augmenter les
performances de votre

AMSTRAD

scanned by www.amstradeus.com



8/0

Table des matières

8/1	Les connecteurs de l'AMSTRAD
8/1.1	La prise d'extension
8/1.2	La prise « unité de disquette 2 »
8/2	Module péritel pour téléviseur couleur
8/2.1	Module commercialisé
8/2.2	Comment construire soi-même un module ?
8/2.3	Quelques branchements vidéo
8/3	Imprimantes
8/3.1	Quelle imprimante choisir ?
8/3.1.1	L'imprimante Citizen 120-D
8/3.2	L'interface CENTRONICS
8/3.3	Le câble de liaison : achat ou réalisation
8/4	La souris AMX mouse
8/5	Systèmes et méthodes de transfert de fichiers
8/5.1	Copies et transmissions de cassettes
8/5.2	L'interface RS-232
8/5.2.1	La liaison RS-232
8/5.2.2	Le code ASCII
8/5.2.3	La RS-232 Amstrad
	I. Les fonctions de base
	II. Les fonctions de transferts de chaînes de caractères
	III. Transferts de fichiers
	IV. Instruction spécifique de ligne
	V. Fin de transmission
	VI. Utilisation d'une imprimante série
	VII. Emulation de terminal
	VIII. Le Vidéotex anglo-saxon
8/5.2.4	Connexions et programmes
8/5.2.4.1	Connexion réduite
8/5.2.4.2	Connexion à un Minitel
	I. Montage du boîtier
	A. Description du montage
	B. Essai préalable
	C. Maintenance superficielle

- II. Logiciels pratiques d'utilisation Minitel
 - A. Sauvegarde de pages Minitel
 - B. Lecture d'une page Minitel sauvegardée
 - C. Communication de données par ligne téléphonique
 - D. Echanges de programme par ligne téléphonique
 - E. Un répondeur télématique

8/5.3 Les modems

8/5.3.1 Le modem Digitec DTL 2000

- I. Connexion à l'Amstrad
- II. L'émulateur Minitel
- III. Programmation avancée

8/5.4 Réalisation d'un serveur télématique

8/5.4.1 La structure matérielle

8/5.4.2 La structure logicielle

- I. Conseils de base
- II. Organisation du serveur
- III. Restrictions d'utilisation

8/5.4.3 Les routines de base

- I. L'initialisation
- II. La connexion
- III. Emission d'un caractère
- IV. Emission d'une série de données
- V. Emission d'une chaîne alphanumérique
- VI. Routine de réception d'un caractère

8/5.4.4 Les essais préliminaires

8/5.4.5 Le programme du serveur

8/5.5 Les ports de sortie

8/6 Les mémoires de masse

8/6.1 Le lecteur de disquettes 5 1/4 Vortex en lecteur additionnel

- I. Présentation
- II. Installation
- III. Mise en place
- IV. Utilitaires Vortex sous CP/M
- V. Le Basic V-DOS
- VI. Le moniteur V-DOS

8/6.2 Multiface II, sauvegarde mémoire

8/6.3 Brancher le lecteur du CPC 464 sur le CPC 6128

8/1

Les connecteurs de l'AMSTRAD

Selon les modèles (464, 664 ou 6128), le nombre de connecteurs disponibles sur le clavier n'est pas le même.

Sur CPC 464, nous disposons des ports suivants :

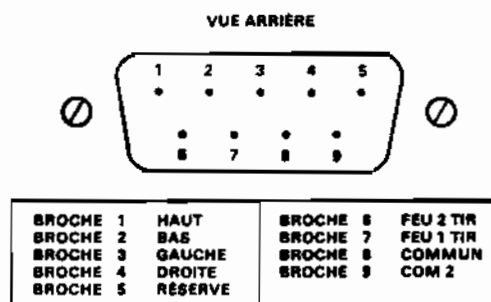
- interface joystick (appelée USER PORT),
- interface CENTRONICS pour imprimante (appelée PRINTER),
- interface pour lecteur de disquettes (appelée FLOPPY DISC),
- alimentation 5 volts (référéncée 5V DC),
- prise pour le moniteur (appelée MONITOR),
- bouton pour ajuster le volume sonore (appelé VOLUME).

Sur CPC 664 et CPC 6128, nous disposons des ports suivants :

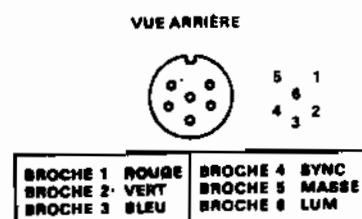
- interface joystick (appelée JOYSTICK),
- interface CENTRONICS pour imprimante (appelée PRINTER),
- interface pour lecteur de disquettes supplémentaire (appelée DISC DRIVE 2),
- alimentation 5 Volts (référéncée 5V DC),
- alimentation 12 Volts (référéncée 12V DC),
- prise pour le moniteur (appelée MONITOR),
- bouton pour ajuster le volume sonore (appelé VOLUME),
- port d'extension (appelé EXPANSION),
- prise pour connecter un magnétophone à cassettes (appelée TAPE),
- prise de sortie sonore stéréo (appelée STEREO).

Le schéma des différents connecteurs est le suivant :

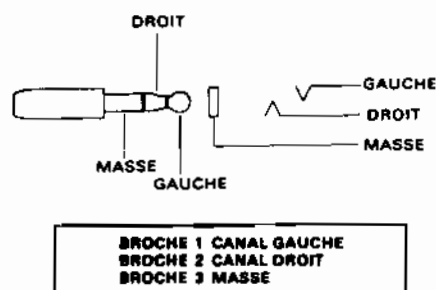
Prise de manette de jeu



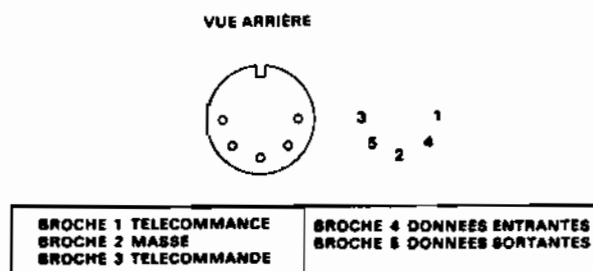
Prise moniteur



Prise stéréo

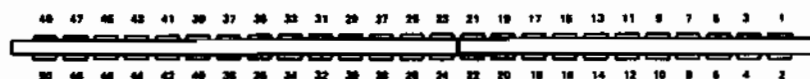


Prise de lecteur de cassette



Prise d'extension

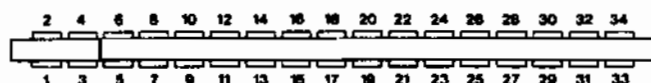
VUE ARRIÈRE



PIN 1	SOUND	PIN 18	A0	PIN 35	INT
PIN 2	GND	PIN 19	D7	PIN 36	NMI
PIN 3	A15	PIN 20	D6	PIN 37	BUSR2
PIN 4	A14	PIN 21	D5	PIN 38	BUSAR
PIN 5	A13	PIN 22	D4	PIN 39	READY
PIN 6	A12	PIN 23	D3	PIN 40	BUS RESET
PIN 7	A11	PIN 24	D2	PIN 41	RESET
PIN 8	A10	PIN 25	D1	PIN 42	ROMEN
PIN 9	A9	PIN 26	D0	PIN 43	ROMDIS
PIN 10	A8	PIN 27	+5V	PIN 44	RAMRD
PIN 11	A7	PIN 28	MREQ	PIN 45	RAMDIS
PIN 12	A6	PIN 29	M1	PIN 46	CURSOR
PIN 13	A5	PIN 30	RFSH	PIN 47	L PEN
PIN 14	A4	PIN 31	IORQ	PIN 48	EXP
PIN 15	A3	PIN 32	RD	PIN 49	GND
PIN 16	A2	PIN 33	WR	PIN 50	0
PIN 17	A1	PIN 34	MALT		

Prise de l'unité de disquette 2

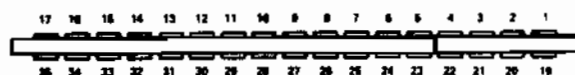
VUE ARRIÈRE



PIN 1	READY	PIN 18	GND
PIN 2	GND	PIN 19	MOTOR ON
PIN 3	MODE SELECT	PIN 20	GND
PIN 4	GND	PIN 21	N/C
PIN 5	READ DATA	PIN 22	GND
PIN 6	GND	PIN 23	DRIVE SELECT 1
PIN 7	WRITE PROTECT	PIN 24	GND
PIN 8	GND	PIN 25	N/C
PIN 9	TRACK 0	PIN 26	GND
PIN 10	GND	PIN 27	INDEX
PIN 11	WRITE GATE	PIN 28	GND
PIN 12	GND	PIN 29	N/C
PIN 13	WRITE DATA	PIN 30	GND
PIN 14	GND	PIN 31	N/C
PIN 15	STEP	PIN 32	GND
PIN 16	GND	PIN 33	N/C
PIN 17	DIRECTION SELECT	PIN 34	GND

Porte pour imprimante

VUE ARRIÈRE



PIN 1	STROBE	PIN 19	GND
PIN 2	D0	PIN 20	GND
PIN 3	D1	PIN 21	GND
PIN 4	D2	PIN 22	GND
PIN 5	D3	PIN 23	GND
PIN 6	D4	PIN 24	GND
PIN 7	D5	PIN 25	GND
PIN 8	D6	PIN 26	GND
PIN 9	GND	PIN 28	GND
PIN 11	BUSY	PIN 33	GND
PIN 14	GND		
PIN 16	GND		

Autres broches : NC

Reportez-vous aux schémas des carte mère et carte du driver de disque (partie 2) pour identifier les différentes connexions physiques entre les connecteurs et les circuits intégrés, en particulier pour la prise d'extension, pour la prise d'unité de disquette 1 ou 2, et pour le port imprimante.

8/1.1

La prise d'extension¹⁾

La plus grande des prises du panneau arrière de l'AMSTRAD (dont l'appellation exacte peut varier selon les modèles) est la *prise d'extension* (Fig. 1).

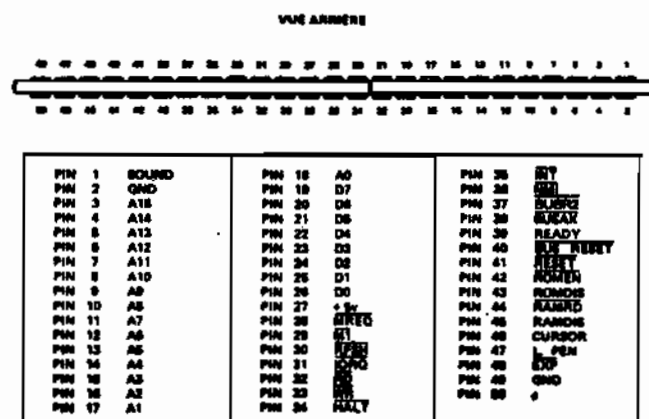


Fig. 1

C'est ce connecteur qui offre les plus grandes possibilités à l'amateur imaginatif, mais c'est aussi celui qui peut causer les plus gros dégâts en cas de fausse manœuvre.

Outre quelques connexions qui ne servent que dans des cas très précis, cette prise donne en effet accès aux *bus du microprocesseur Z 80*.

On sait que ce circuit (petit boîtier en plastique muni de 40 broches) est le cœur de l'ordinateur : c'est lui qui coordonne le fonctionnement de tous les autres circuits de la machine, en exécutant en permanence le programme enregistré dans la mémoire ROM ou dans une cartouche extérieure.

Pour remplir ce rôle, le microprocesseur doit être électriquement relié à tous ses « périphériques ».

Pour limiter à un niveau acceptable le nombre de ces connexions, il a été imaginé qu'un certain nombre de lignes desservent à la fois (en « parallèle ») tous les circuits devant communiquer avec le Z 80. Ces lignes,

1) Chapitre rédigé par Patrick Gueulle.

véritables « autoroutes internes » sont les *bus*. En envoyant certains signaux sur ces bus, le Z 80 peut informer tel ou tel périphérique ou telle ou telle mémoire qu'il désire établir un contact. Le périphérique se met alors en état de recevoir des données, ou envoie au microprocesseur les données réclamées, strictement à son tour.

Trois bus distincts sont nécessaires pour cela :

1°) *Le bus de données est bidirectionnel* : ses huit lignes (D0 à D7) peuvent véhiculer des octets (groupes de huit bits) du Z 80 vers les périphériques, ou des périphériques vers le microprocesseur. Il transporte les informations devant être enregistrées ou lues en mémoire, ou entrant et sortant par les périphériques (écran, clavier, imprimante, disquette, cassette, manettes, etc.)

Outre les échanges imposés par le programme en cours d'exécution, ce bus amène au microprocesseur, depuis la mémoire ROM, les *instructions* qu'il exécutera immédiatement et fidèlement.

2°) *Le bus d'adresses, lui, ne véhicule ses informations que du microprocesseur vers des circuits extérieurs* (mémoires, périphériques, etc.). C'est par lui que le Z 80 avertit les autres circuits qu'il désire communiquer avec eux. Comme il possède seize lignes, il peut distinguer 2^{16} « adresses » différentes, c'est-à-dire 65536 (on parle volontiers de 64 K, 1 K valant 1024).

Pour dépasser 64 K de mémoire, il est donc nécessaire de recourir à des artifices d'agencement de ces mémoires sur les bus.

A côté des mémoires, le Z 80 peut aussi sélectionner des « ports » d'entrée-sortie, servant à échanger des octets non plus avec des mémoires, mais avec des périphériques (circuits de son, imprimante, manettes de jeu, etc.).

3°) *Le bus de commande* permet au microprocesseur d'indiquer à ses circuits associés s'il s'adresse à une mémoire ou à un port, s'il désire recevoir des octets ou au contraire s'il va en envoyer. Il gère également des fonctions très particulières comme les *interruptions* ou le *rafraîchissement* des mémoires dynamiques.

La figure 2 résume cette architecture selon laquelle sont conçus tous les systèmes à microprocesseurs, malgré des variantes de détail.

Apparaissent en même temps les principales broches du connecteur d'extension de l'AMSTRAD qui, on le voit, sont reliées au cœur même de la machine.

En plus de tous les signaux des bus, on remarque des signaux spécifiques AMSTRAD, élaborés par les *circuits de sélection* de la machine : ROMEN, ROMDIS, RAMRD, RAMDIS sont notamment utilisés pour piloter les mémoires ROM et RAM internes de l'ordinateur, et pourront contribuer à simplifier l'étude des *circuits de sélection* d'éventuelles mémoires externes.

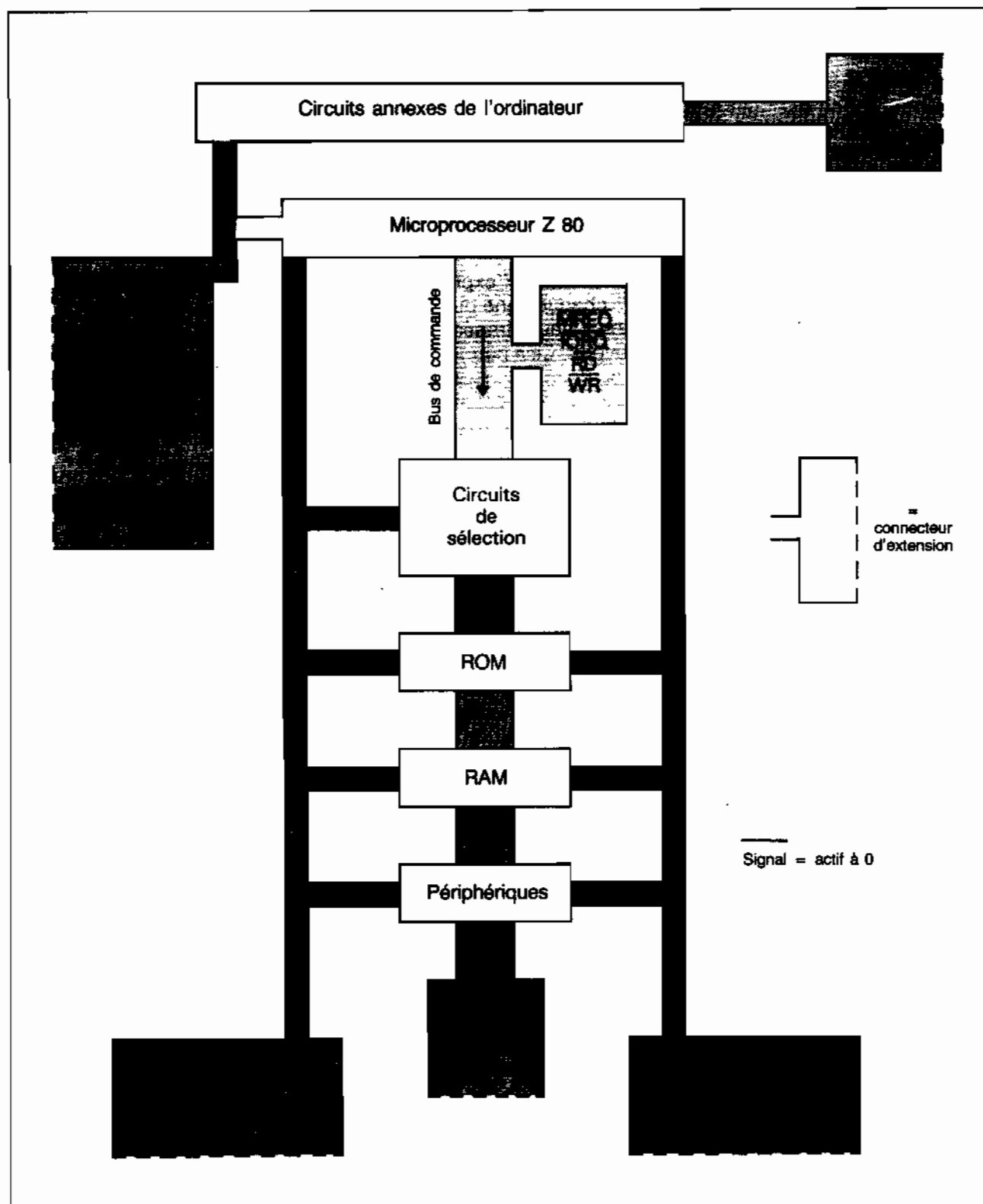


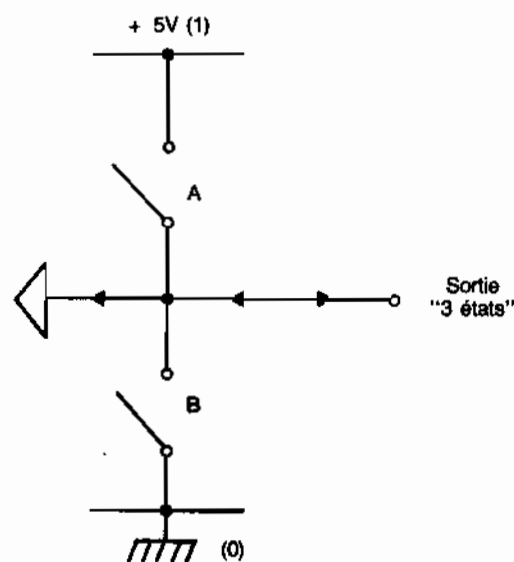
Fig. 2

Un tel *circuit de sélection* est, en effet, indispensable pour *interfacer* une mémoire ou un périphérique sur les bus : en réalisant une *combinaison logique* entre certaines lignes d'adresse et des signaux de commande tels que $\overline{\text{MREQ}}$ (accès à la mémoire), $\overline{\text{IORQ}}$ (accès aux ports), RD (lecture) ou $\overline{\text{WR}}$ (écriture), il élabore le signal destiné à la *broche de sélection* du circuit de mémoire ou de port. C'est ce processus de *décodage* des bus qui permet de placer un boîtier de mémoire ou un périphérique à l'endroit voulu dans la *carte de la mémoire* de la machine.

Pourtant, tous ces boîtiers sont bel et bien branchés ensemble sur les bus ! Comment est-ce donc possible ?

La *figure 3* donne l'explication ; le bus de données est du type « trois états » : chaque ligne du bus arrivant sur un boîtier rencontre normalement une « haute impédance », les « interrupteurs » A et B étant tous deux ouverts. Cela permet au circuit de *lire* les informations circulant sur le bus, ou de rester inactif. Sur demande expresse du microprocesseur, via son circuit de sélection, il peut cependant fermer l'un ou l'autre des interrupteurs A et B, mais évidemment pas les deux à la fois : il impose alors un état 1 ou 0 à la ligne de bus.

Evidemment, il faut à tout prix éviter que l'interrupteur A d'un circuit relié au bus puisse se fermer en même temps que l'interrupteur B d'un autre : il y aurait court-circuit et très vraisemblablement destruction d'un ou plusieurs composants. C'est pourquoi l'adjonction de compléments de mémoire (RAM ou ROM) ou de périphériques (cartes d'entrée-sortie, etc.) exige une très sérieuse étude théorique : brancher les bus d'adresse et de données est très simple, mais élaborer le signal de sélection correct est une toute autre affaire.



A	B	Sortie
fermé	fermé	cas interdit
ouvert	ouvert	"haute impédance"
ouvert	fermé	0
fermé	ouvert	1

Fig. 3

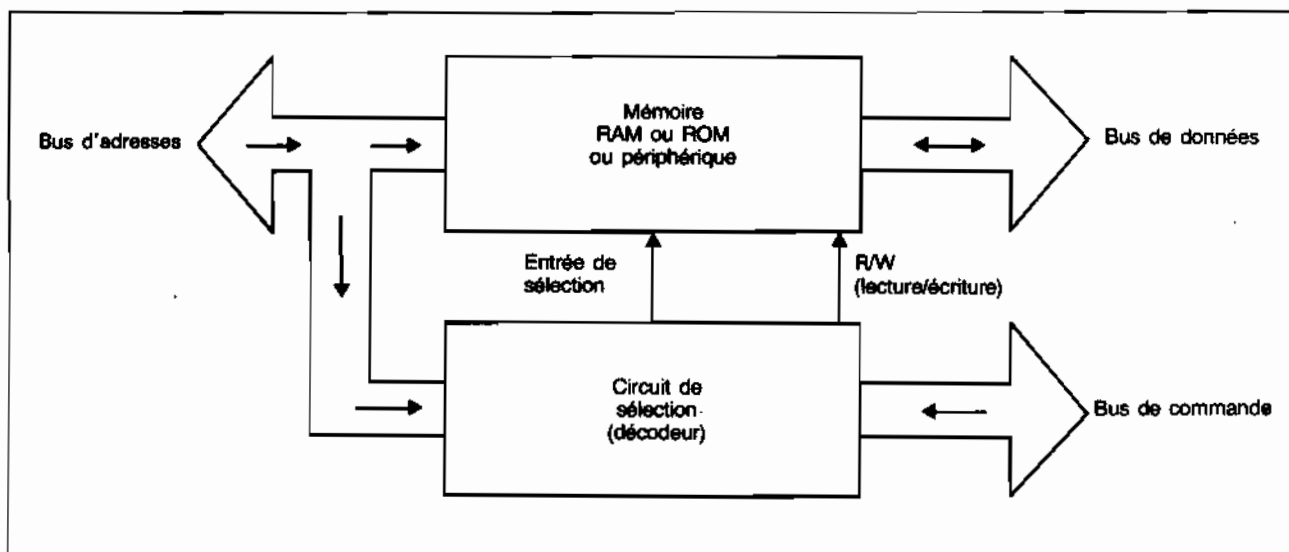


Fig. 4

Il faut en premier lieu décider de la zone d'adresses dans laquelle se fera l'implantation, soit en trouvant une zone libre dans le « plan de mémoire », soit en en dégagant une de force. Cela fait, il faudra concevoir un circuit *décodeur* qui, combinant les signaux des bus de commande et d'adresse, ne fera quitter l'état « haute impédance » à notre extension qu'à bon escient. La *figure 4* résume schématiquement l'organisation matérielle d'une telle extension (RAM additionnelle, « cartouche » de ROM, périphérique, etc.)

Le lecteur aura compris que ce genre d'adaptation nécessite de solides connaissances en micro-informatique et en électronique, associées à une documentation complète sur l'architecture du système à modifier : toutes choses qu'il trouvera dans les divers chapitres de cet ouvrage !

Une autre approche consiste à faire confiance aux spécialistes en la matière, et à réaliser l'un des montages pratiques proposés en quelque sorte « clé en main ».

8/1.2

La prise « unité de disquette 2 »

Le circuit contrôleur de disques utilisé dans les CPC 664 et 6128 ainsi que dans l'interface pour CPC 464 est capable, comme beaucoup de ses semblables, de prendre en charge deux lecteurs distincts.

Il est donc possible de brancher un second lecteur (très logiquement baptisé lecteur 2 !) sur la prise d'extension équipant les CPC déjà munis d'un drive, ou sur le boîtier d'interface DDI1 pour CPC 464.

Dans les deux cas, le schéma du contrôleur est le même : on en trouvera une description détaillée en Partie 2, chapitre 3.5.

Il importe de bien connaître le rôle de chacun des multiples signaux transitant entre le contrôleur et les lecteurs qu'il pilote : en effet, ceux-ci sont normalisés et donc communs à tous les lecteurs de disquette sans distinction de format ni de marque. Cela signifie qu'il est matériellement possible de brancher sur un CPC, un lecteur 3' 1/2 ou 5' 1/4 de provenance quelconque, quitte à faire quelques adaptations logicielles pour qu'il soit correctement reconnu et utilisé par le système d'exploitation AMSTRAD.

D'un autre côté, ces connaissances pourront se révéler utiles en cas de panne : quelques tests simples permettront alors de déterminer si le problème provient du lecteur (généralement irréparable par l'amateur), de son contrôleur, ou de l'unité centrale, à moins qu'il ne s'agisse tout simplement que d'un problème d'alimentation...

Si l'on excepte les connexions d'alimentation (masse, + 5V et éventuellement + 12V) réunies dans une prise séparée, les signaux échangés entre un lecteur et son contrôleur sont des niveaux TTL en logique négative. Certains d'entre eux transmettent des ordres du contrôleur au lecteur, les autres rendant compte au contrôleur de l'état du lecteur.

INDEX : cette ligne est mise à zéro volt lorsque le trou d'index de la disquette est en face du capteur chargé de le détecter.

DRIVE SELECT 0 ET DRIVE SELECT 1 :

le contrôleur met à zéro volt la ligne correspondant au lecteur avec lequel il veut communiquer, d'après les instructions transmises par le système d'exploitation. Un cavalier ou un interrupteur incorporé à chaque lecteur

permet de lui assigner le numéro 0 ou 1 : il est convenu que le premier lecteur porte le n° 0 et l'éventuel lecteur supplémentaire le n° 1.

MOTOR ON : le contrôleur met cette ligne à zéro volt lorsqu'il doit faire tourner le moteur d'entraînement de la disquette. Mis à la masse sur un lecteur débranché mais alimenté, ce fil permet de tester le moteur et son circuit de commande.

DIRECTION SELECT : cette ligne permet au contrôleur de spécifier la direction dans laquelle la tête devra se déplacer lorsqu'une impulsion sera appliquée au moteur pas à pas qui l'actionne.

STEP : un bref niveau zéro volt sur cette ligne permet au contrôleur de faire avancer ou reculer la tête (selon l'état de **DIRECTION SELECT**) d'un cran, par l'intermédiaire d'un moteur pas à pas (*Stepper Motor*). Plusieurs de ces impulsions sont en général nécessaires pour amener la tête là où elle doit aller.

WRITE DATA : sur cette ligne transitent les données destinées à être enregistrées sur la disquette.

WRITE GATE : cette ligne véhicule la commande de validation des données circulant sur la ligne **WRITE DATA**.

TRACK 00 : cette ligne est mise à zéro volt par le lecteur lorsque la tête est positionnée en regard de la piste n° 0 de la disquette : elle est en fait commandée par un capteur de fin de course monté sur la glissière de la tête.

WRITE PROTECT : cette ligne est mise à zéro volt par le lecteur si celui-ci contient une disquette protégée en écriture : recevant cette information, le contrôleur devient incapable d'enregistrer quoi que ce soit, même si le système d'exploitation lui en donne l'ordre. Toutefois, la tête peut se déplacer normalement sur le disque : une disquette protégée n'est donc pas nécessairement à l'abri des divagations d'un contrôleur défectueux...

READ DATA : cette ligne rapatrie vers le contrôleur les données lues sur la disquette.

SIDE 1 SELECT : selon l'état de cette ligne, les opérations de lecture et d'écriture interviennent sur l'une ou l'autre face de la disquette, à condition bien sûr que le lecteur soit du type double face. Les lecteurs AMSTRAD étant du type 3 pouces simple face, cette ligne reste normalement inutilisée, mais on peut songer à s'en servir si on adapte un lecteur double face 3 1/2 ou 5 1/4.

READY : cette ligne est mise à zéro volt par le lecteur lorsqu'il est en état d'exécuter les ordres du contrôleur : si READY est à +5V, le contrôleur suspend l'émission d'ordres jusqu'à ce que le lecteur se déclare prêt, mais déclenche une erreur au niveau du système d'exploitation si cette attente dure trop longtemps.

Tous ces signaux circulent sur un câble plat appelé limande, dont la longueur ne doit pas excéder quelques dizaines de centimètres, et dans lequel chaque fil de signal est séparé de son voisin par un fil de masse. Le brochage, normalisé, du connecteur du lecteur est le suivant :

1	masse	2	INDEX
3	masse	4	DRIVE SELECT 0
5	masse	6	DRIVE SELECT 1
7	masse	8	MOTOR ON
9	masse	10	DIRECTION SELECT
11	masse	12	STEP
13	masse	14	WRITE DATA
15	masse	16	WRITE GATE
17	masse	18	TRACK 00
19	masse	20	WRITE PROTECT
21	masse	22	READ DATA
23	masse	24	SIDE 1 SELECT
25	masse	26	READY

8/2

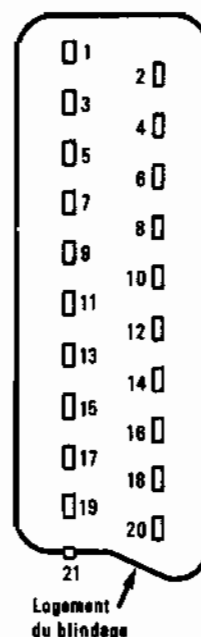
Module péritel pour téléviseur couleur

Ce dispositif est installé sur tous les téléviseurs couleurs commercialisés en France depuis 1981. Il permet d'utiliser un téléviseur couleur comme un moniteur. La plupart des micro-ordinateurs possèdent les signaux de synchronisation nécessaires au pilotage de la prise péritel. L'AMSTRAD fait partie de ceux-là. De plus, la qualité obtenue sur le téléviseur en utilisant la prise péritel est supérieure à celle que l'on aurait en utilisant un modulateur UHF.

Nous voyons donc qu'il peut être intéressant de connecter l'AMSTRAD sur votre téléviseur couleur, surtout si vous possédez un CPC avec écran monochrome.

La prise PERITEL implantée sur le téléviseur comporte 21 broches implantées de la façon suivante :

VUE CÔTÉ CÂBLAGE



Le pas entre deux contacts est de 3.81 mm et la distance entre les deux rang es de 5.08 mm.

Les signaux qui arrivent ou partent de chaque broche sont les suivants :

- 1 Sortie audio voie B
- 2 Entr e audio voie B
- 3 Sortie audio voie A
- 4 Masse audio
- 5 Masse vid o
- 6 Entr e audio voie A
- 7 Entr e composante bleu
- 8 Entr e commutation lente (imp dance d'entr e > 10 k )
- 9 Masse de la composante vert
- 10 Ligne omnibus de donn es
- 11 Entr e composante vert
- 12 Ligne omnibus de donn es
- 13 Masse de la composante rouge
- 14 Masse des lignes omnibus
- 15 Entr e composante rouge
- 16 Entr e commutation rapide
- 17 Masse vid o
- 18 Masse commutation rapide
- 19 Sortie vid o composite positive
- 20 Entr e vid o composite ou synchro (imp dance 75 )

8/2.1

Module commercialisé

L'AMSTRAD CPC génère tous les signaux nécessaires pour piloter la prise PERITEL. Nous pouvons donc penser qu'il est simple de le connecter à un téléviseur PERITEL.

Effectivement, la connexion est simple. Il subsiste cependant un petit problème : le moniteur AMSTRAD contient une alimentation stabilisée de 5 V pour alimenter le clavier, et une alimentation stabilisée de 12 V sur les modèles CPC 664 et CPC 6128 pour alimenter le lecteur de disquettes.

Le module PERITEL présenté dans le commerce est donc constitué :

- d'un câble de liaison entre la sortie vidéo de l'AMSTRAD et la prise PERITEL,
- de deux alimentations stabilisées (5 volts et 12 volts).

8/2.2

Comment construire soi-même un module ?

Mis à part le problème de connectique entre ordinateur et téléviseur, il faut fabriquer deux alimentations stabilisées. L'examen des cartes mères des CPC (voir Partie 2 chap. 2) nous montre que l'alimentation de 5 volts

8/2.3

Quelques branchements vidéo

L'organisation des systèmes Amstrad en un ensemble indivisible clavier-moniteur ouvre la porte à toute une variété d'adaptations de matériels vidéo de toutes sortes : utilisation des images composées par l'Amstrad sur des magnétoscopes ou des moniteurs « non-Amstrad », mais aussi utilisation des moniteurs Amstrad avec des sources d'images autres que le clavier d'origine.

Les branchements nécessaires ne sont pas compliqués, mais encore faut-il savoir ce qui transite par les connecteurs qu'il va falloir utiliser !

Le principe même de la vidéo est de transmettre des images animées (25 images fixes par seconde) sur un câble ou une liaison radio, ou de les enregistrer sur bande magnétique. Toute image est donc représentée par des signaux électriques que l'on nomme *signaux vidéo*.



Les connecteurs des moniteurs CPC.

Un signal vidéo noir et blanc se compose d'une tension continuellement variable représentant la « blancheur » plus ou moins forte des points de l'image « balayés » ligne par ligne (625 lignes par image). S'y ajoutent des impulsions dites « tops de synchro » indispensables pour que le balayage de l'écran puisse se synchroniser avec celui de la caméra ou de la source d'images (ordinateur, par exemple).

En couleur, c'est plus compliqué puisqu'il ne suffit pas de transmettre la luminance : trois signaux séparés sont nécessaires pour véhiculer les niveaux de rouge, de vert, et de bleu, couleurs primaires par mélange desquelles l'écran peut reproduire n'importe quelle teinte.

En fait, pour des raisons de compatibilité avec les matériels noir et blanc, on transmet aussi la luminance, quitte à supprimer l'une des composantes de couleur puisque la luminance est évidemment la somme des trois couleurs.

Des mélanges sont possibles entre ces différents signaux, qui permettent de limiter le nombre de conducteurs (ou de voies radio) nécessaires à leur transmission : le mélange de la luminance et des tops de synchro donne ce qu'il est convenu d'appeler une *Vidéo Composite Noir et Blanc* pouvant circuler sur un câble coaxial unique ou sur une voie radio simple.

En couleur, le simple mélange est impossible, sous peine de retomber sur un signal noir et blanc : un *codage* est nécessaire, tout comme pour transmettre de la stéréo à la radio. C'est là que différents standards peuvent être distingués : le SECAM, le PAL, le NTSC, et maintenant le D2MAC utilisent des artifices différents (et incompatibles !) pour véhiculer à la fois la synchro, la luminance, et les couleurs sur un même câble ou une même voie radio. Codeurs et décodeurs sont des appareils assez complexes et donc coûteux.

Pour des liaisons à courte distance (et notamment du clavier au moniteur d'un Amstrad), il est inutile de compliquer les choses par un tel codage : la figure 1 montre qu'on utilise carrément un fil pour chaque couleur, un pour la synchro, et évidemment un (le blindage) pour la masse.

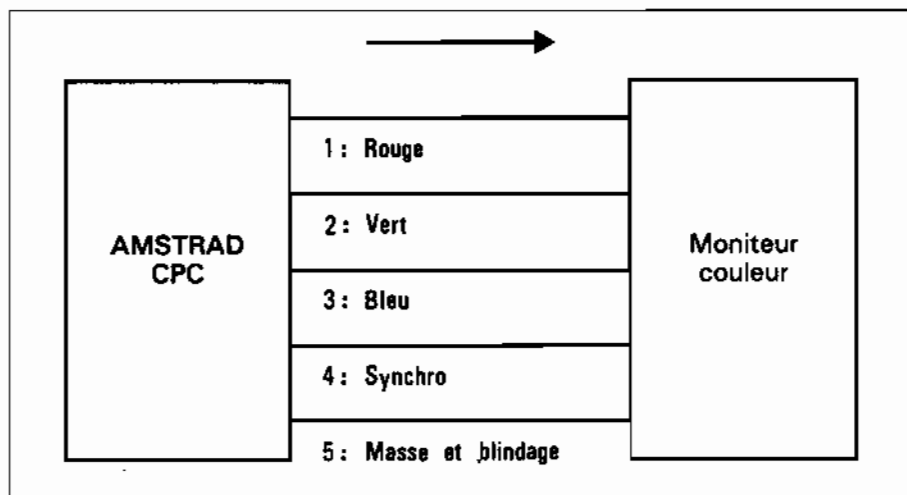


Fig. 1 : Connexion clavier / moniteur couleur.

En noir et blanc, la figure 2 montre que trois fils suffisent, luminance et synchro étant transmis séparément.

Toutefois, la figure 3 permet de constater que le circuit de sortie vidéo de l'Amstrad mélange un peu de synchro à la luminance : sur la broche 6 de la prise DIN vidéo (dont le brochage est donné à la figure 4), c'est donc en fait une véritable *vidéo composite* dont dispose l'utilisateur.

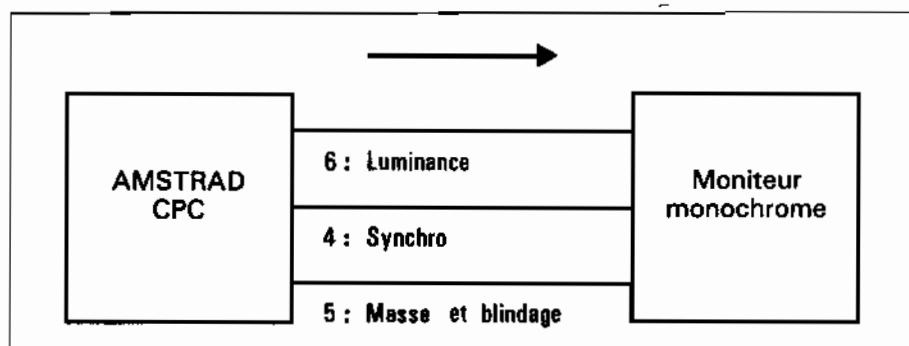


Fig. 2 : Connexion clavier / moniteur monochrome.

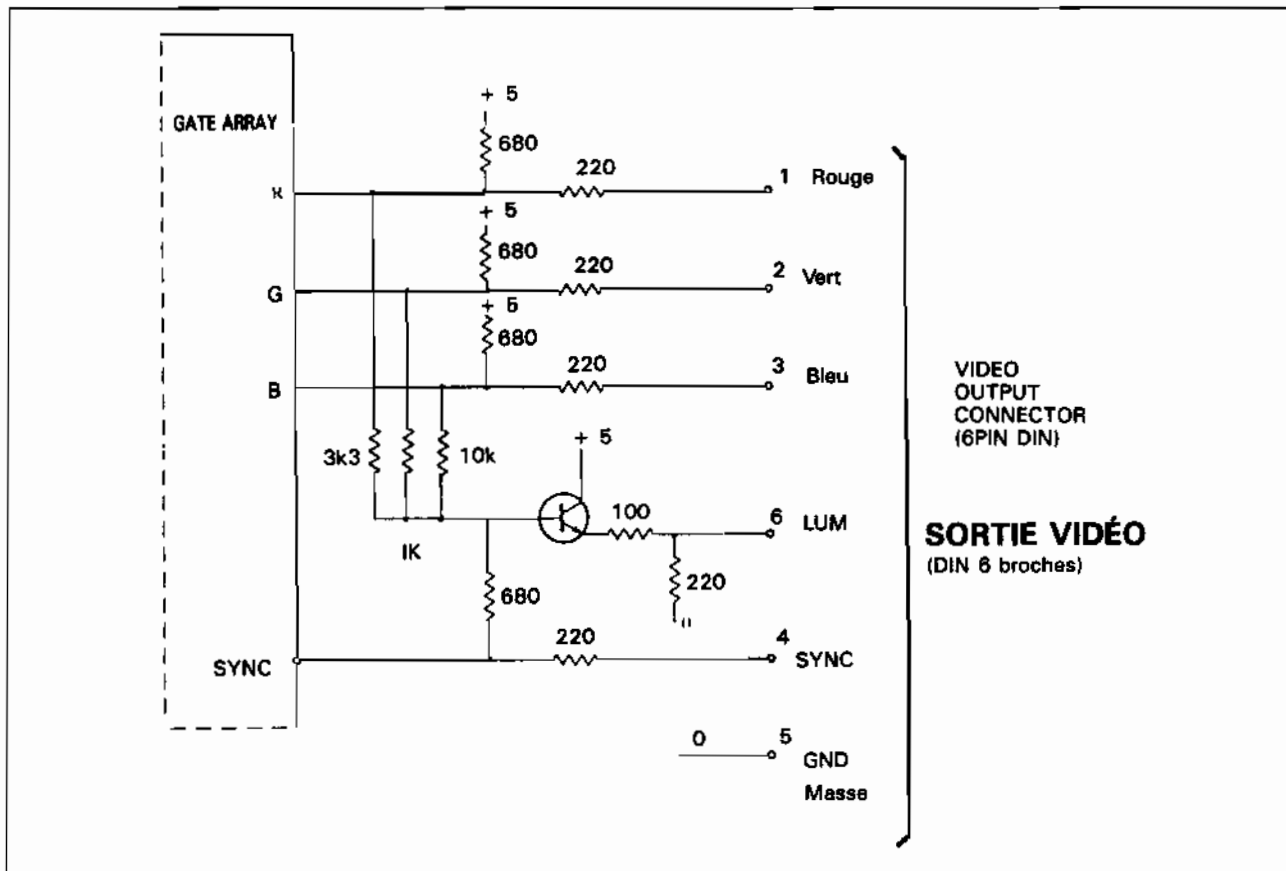


Fig. 3 : Circuit de sortie vidéo des CPC.

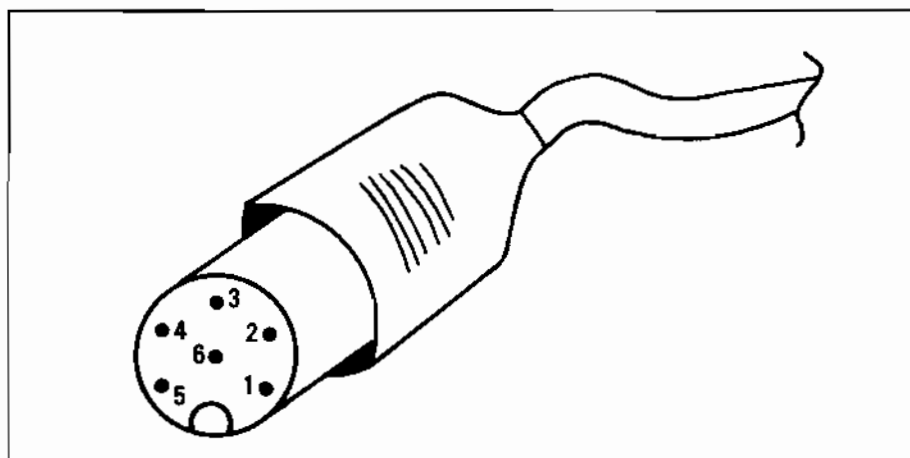


Fig. 4 : Prise DIN vidéo du moniteur.

Rien de plus facile donc que de brancher l'Amstrad par un simple câble coaxial à la prise VIDEO IN (entrée vidéo) d'un moniteur monochrome non-Amstrad, ou même d'un magnétoscope : toute image produite par l'Amstrad (qui continue à être alimenté par son moniteur même si la fiche vidéo est débranchée) peut donc être enregistrée, *mais en noir et blanc*.

C'est toutefois déjà très intéressant, par exemple pour réaliser des titrages de films vidéo personnels, mais avec une qualité quasi-professionnelle (il existe d'excellents logiciels de composition d'images).

La figure 5 détaille le branchement à réaliser, tandis que la figure 6 indique ce qu'il faudrait pour obtenir une image couleur sur l'enregistrement : le codeur Secam nécessaire existe dans le commerce, mais ce n'est pas bon marché...

L'opération inverse est évidemment tout aussi envisageable : la figure 7 montre comment on peut arriver à se servir d'un moniteur monochrome Amstrad pour visualiser une image provenant d'un magnétoscope, d'une caméra, ou de toute source vidéo composite (prise coaxiale VIDEO OUT, c'est-à-dire sortie vidéo). Le même signal composite doit être appliqué à la fois sur l'entrée « luminance » et sur l'entrée « synchro » du moniteur.

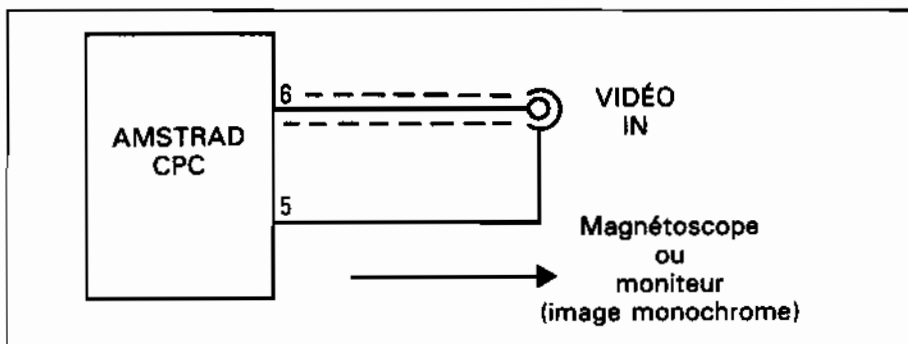


Fig. 5 : Connexion du CPC sur un « récepteur vidéo ».

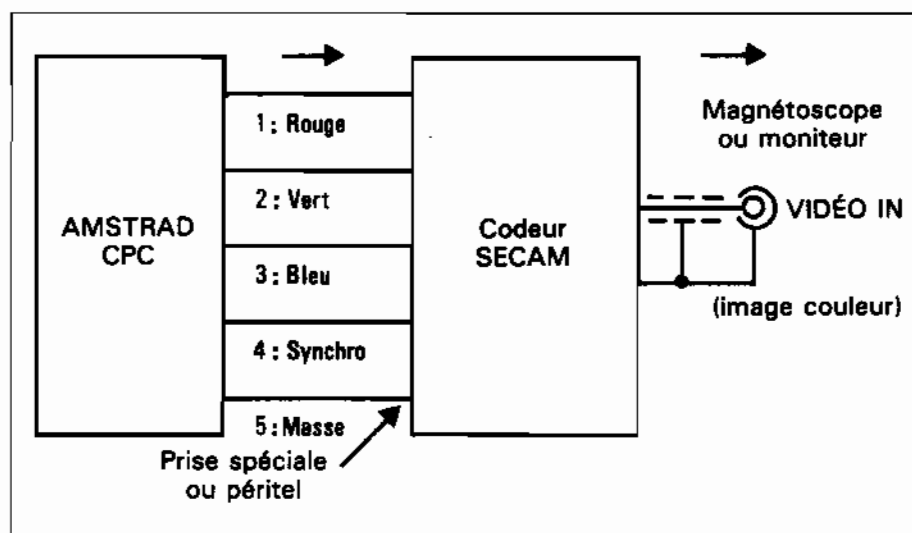


Fig. 6 : Connexion d'un codeur Secam sur le CPC.

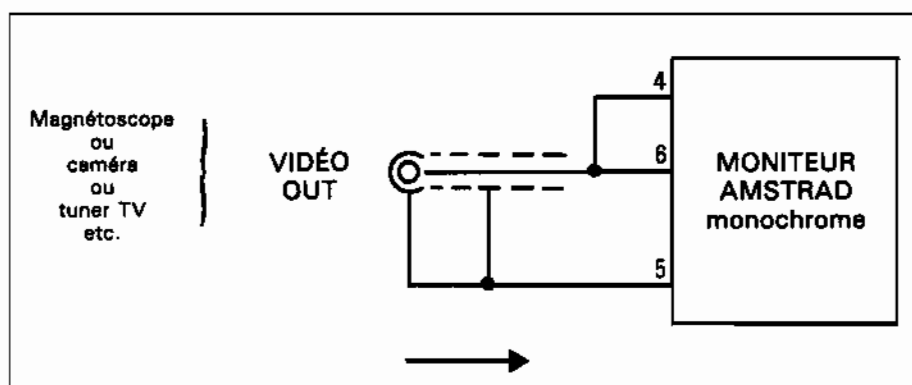


Fig. 7 : Utilisation d'un moniteur Amstrad monochrome comme « récepteur vidéo ».

Bien sûr, l'image est de la couleur de l'écran (vert ou ambre) et donc moins contrastée que sur un véritable écran TV, mais très correcte pour de la surveillance ou du contrôle.

Le même branchement peut se faire sur un moniteur couleur Amstrad (figure 8 en page 7), mais en utilisant cette fois l'une des trois entrées couleur, voire plusieurs en parallèle : l'image obtenue sera alors reproduite dans la couleur correspondant à l'entrée utilisée, comme si un écran noir et blanc était regardé à travers un filtre !

Pour obtenir une véritable image couleur, il faut que le signal vidéo soit disponible sous la forme « RVB-S » (rouge vert bleu synchro), comme indiqué sur le tableau 1. Le plus courant est une fiche Péritel, par exemple celle d'un ordinateur d'une autre marque ou d'une console de jeu, mais il peut aussi s'agir d'un autre type de connecteur multibroche (adaptation à étudier).

Tableau 1 : Connexion d'un CPC sur une source RVB.

Moniteur couleur Amstrad	Source RVB-S péritel
1	15
2	11
3	7
4	20
5	17, 13, 9, 5
6 non reliée	

Un cas particulier intéressant est le raccordement d'un « tuner » télévision : pour peu que celui-ci sorte bien en « RVB-S » (comme celui proposé par Amstrad), les émissions de télévision pourront être regardées en couleurs sur l'écran du moniteur lorsqu'il ne sert pas à l'usage informatique !

Inversement, le tableau 2 montre comment le clavier d'un Amstrad (alimenté par son moniteur ou par un bloc-secteur séparé) peut envoyer une image couleur sur tout téléviseur équipé d'une prise Péritel. Il faudra en général actionner une touche particulière du récepteur, ou envoyer du +12 V (par rapport à la masse de la broche 18) sur la broche 8 de la Péritel (voir figure 9) pour le commuter en « moniteur ».

Rappelons que toutes ces liaisons sont à faire en fil blindé (petit coaxial 75 Ω), ou mieux en câble « spécial Péritel », les blindages devant tous être mis à la masse.

Il est en général conseillé de limiter la longueur de ces câbles, bien que les signaux vidéo puissent voyager facilement sur plusieurs dizaines de mètres si nécessaire.

Tableau 2 : Connexion d'un CPC sur un téléviseur couleur.

Amstrad CPC	Téléviseur couleur
1	15
2	11
3	7
4	20
5	17, 13, 9, 5
6 non reliée	

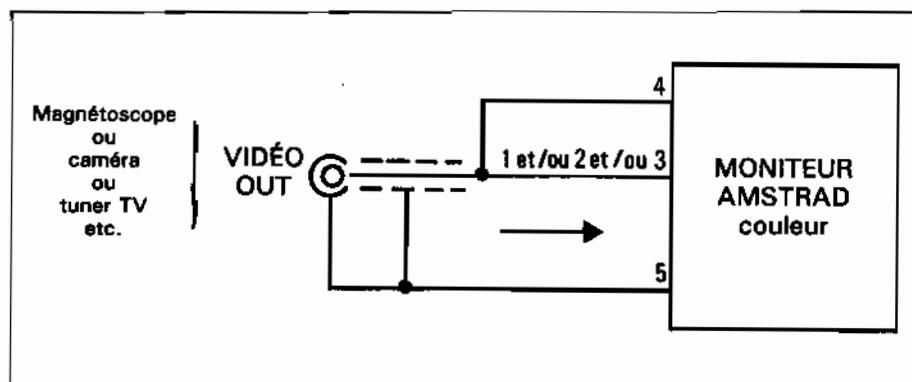


Fig. 8 : Utilisation d'un moniteur Amstrad couleur comme « récepteur vidéo ».

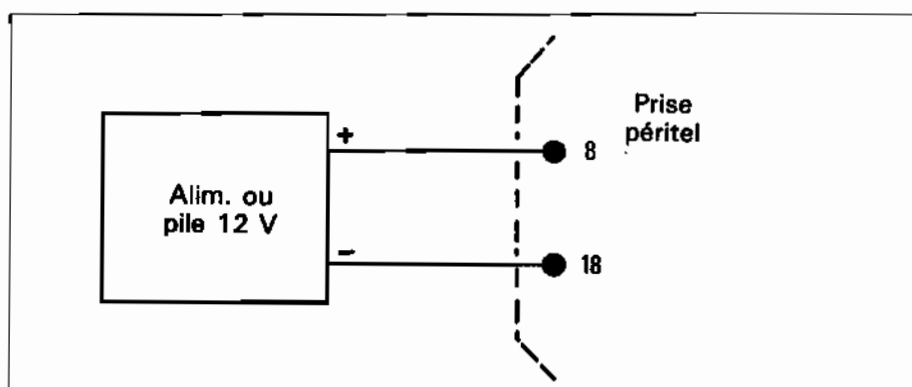


Fig. 9 : Alimentation d'une prise Péritel (commutation en moniteur).

8/3

Imprimantes

Ce périphérique vous sera bien utile si :

- vous programmez,
- vous utilisez :
 - un programme de traitement de textes,
 - un gestionnaire de fichiers,
 - un tableur,
 - un programme de tracé graphique.

Théoriquement, vous n'avez pas besoin d'imprimante pour mettre au point des programmes si vous utilisez les concepts de programmation hiérarchisée structurée définis dans cet ouvrage. En effet, le programme est divisé en plusieurs modules élémentaires qui, individuellement, ne demandent pas de phase de mise au point car ils manipulent peu de concepts différents. En pratique, si le programme que vous mettez au point a une longueur importante, ou si les divers modules qui le composent ont une longueur importante, il pourra être utile de lister sur imprimante le programme en question. D'une manière générale, même si les supports cassettes ou disquettes sont relativement sûrs, vous aurez intérêt à lister sur papier un programme que vous avez mis au point, surtout si celui-ci a une longueur importante.

De plus, certains programmes rendent obligatoire l'utilisation d'une imprimante. Ainsi, un programme d'édition de texte ou de mailing n'est vraiment utile que si vous possédez une imprimante, de qualité courrier si possible. De même, il pourra être intéressant de sortir le résultat d'une grille constituée par un tableur sur papier.

8/3.1

Quelle imprimante choisir ?

Si vous avez décidé d'acheter une imprimante, nous vous conseillons de lire attentivement ce qui suit avant de vous précipiter chez le marchand.

Les ordinateurs AMSTRAD peuvent gérer n'importe quelle imprimante parallèle à interface de type CENTRONICS fonctionnant sur un bus de 7 ou 8 bits de données. Si l'imprimante peut fonctionner sur 8 bits, 7 seulement seront utilisés par l'AMSTRAD pour imprimer des codes ASCII compris entre 0 et 127, car un bus de 7 bits seulement est disponible sur la prise imprimante AMSTRAD.

Le budget nécessaire va dépendre du type d'imprimante que vous choisirez :

- jusqu'à 1 500 francs pour une imprimante thermique,
- de 1 500 à 4 000 francs pour une imprimante matricielle, selon la qualité et les possibilités d'impression (gras, italique, souligné, double largeur, etc.),
- de 5 000 à 8 000 francs pour une imprimante à marguerites,
- de 5 000 à 20 000 francs pour une table traçante,
- plus de 10 000 francs pour une imprimante à jet d'encre ou laser.

Le type d'imprimante dont vous avez besoin va dépendre de l'utilisation que vous comptez en faire.

IMPRIMANTE THERMIQUE

Ces imprimantes ont tendance à disparaître du marché. Vous choisirez ce type d'imprimante si vous pensez l'utiliser occasionnellement. En effet, une imprimante thermique est assez peu chère (1 500 francs au maximum). Mais, d'un autre côté, le papier utilisé est de type spécial et, lui, d'un prix assez élevé. De plus, la vitesse d'impression de telles imprimantes est souvent très lente (jusqu'à 60 caractères par seconde au maximum) et pourra être un facteur déterminant si vous désirez beaucoup l'utiliser.

IMPRIMANTE MATRICIELLE

Ces imprimantes très répandues offrent une qualité d'impression raisonnable et un rapport qualité/prix très correct. Leur prix est très variable, en fonction de la qualité d'impression désirée, mais la plupart de ces imprimantes sont vendues aux alentours de 2 000 francs.

Leur principe de fonctionnement est très simple : la tête d'impression renferme plusieurs aiguilles (souvent huit) disposées les unes sous les autres. Pour imprimer un caractère, elles seront activées n fois, et la tête se déplacera d'un pixel vers la droite à chaque activation (n est le nombre de pixels qu'occupe un caractère en largeur). Ceci explique la dénomination « matricielle ». Elles sont parfois également qualifiées d'imprimantes à impact, ceci pour caractériser leur mode d'impression : n aiguilles sont poussées vers le papier à travers un ruban encreur.

La vitesse d'impression classique en qualité courante est de 80 à 100 caractères par seconde sur une imprimante bas de gamme et 400 à 500 caractères par seconde sur une imprimante haut de gamme.

Les imprimantes matricielles possèdent très souvent un mode d'impression « qualité courrier » qui consiste à imprimer les lignes une première fois, puis une seconde fois en décalant légèrement la tête d'impression pour obtenir une qualité d'impression supérieure.

Très souvent, des micro-interrupteurs (appelés *switches*) situées derrière l'imprimante permettent de sélectionner l'écriture dans plusieurs langues, comme, par exemple USA, GB, France, Allemagne, Danemark, Suède, Italie, Espagne. Reportez-vous à la notice qui accompagne l'imprimante pour connaître les modalités d'utilisation de ces switches, et n'hésitez pas à demander au vendeur si ces switches existent sur l'imprimante que vous comptez acheter. D'autres switches souvent présents permettent de :

- détecter la présence/l'absence du papier,
- définir la longueur du papier,
- définir le mode d'impression par défaut (normal, gras, italique, etc.).

Enfin, certaines imprimantes matricielles permettent d'accéder au pixel, et ainsi d'imprimer des graphismes. Cette possibilité n'est pas générale à toutes les imprimantes matricielles ; aussi, renseignez-vous sur cette possibilité avant de l'acquérir.

IMPRIMANTE A MARGUERITES

Si vous recherchez une grande qualité d'impression pour un prix inférieur à celui des imprimantes à jet d'encre ou à laser, nous vous conseillons d'opter pour une imprimante à marguerites. Ces imprimantes fonctionnent sur le même principe que les machines à écrire classiques : une boule sur laquelle se trouve un jeu de caractères est frappée par un marteau ; elle entre en contact avec un ruban encreur, puis enfin avec le papier. La qualité d'impression dépend de la qualité de la boule de

caractères qui, généralement, est très correcte. Ce type d'imprimantes a cependant plusieurs inconvénients :

- le premier, et non le moindre, est le bruit qu'elles produisent ;
- le second est la vitesse d'impression qui dépasse rarement 40 caractères par seconde ;
- enfin, le troisième est le suivant : une seule police de caractères peut être utilisée en même temps. Si vous désirez utiliser plusieurs polices de caractères, il vous faudra changer de boule d'impression. A ce sujet, il existe de nombreux types de boules pour générer des caractères pica, élite, italiques, etc..

TABLES TRAÇANTES

Il n'existe aucune table traçante dédiée à l'AMSTRAD, mais plusieurs tables peuvent être interfacées assez aisément par soft.

L'utilisation d'une table traçante se justifie si vous désirez réaliser des dessins en haute définition et/ou en couleurs sur un format classique (A4) ou plus grand (A2, A1 ou même A0).

Vu le prix d'une table traçante (2 à 3 fois celui de l'ordinateur), son acquisition ne sera justifiée que dans certains cas particuliers, souvent en utilisation professionnelle, comme pour reproduire des cartes ou des plans, ou tracer des pièces à réaliser dans un atelier de tournage ou de fraisage.

IMPRIMANTE À JET D'ENCRE OU À LASER

Ces imprimantes sont comparables aux imprimantes matricielles à impact citées précédemment, à ceci près qu'elles n'utilisent pas d'aiguilles, mais un jet d'encre pour les imprimantes à jet d'encre et un rayon laser pour les imprimantes laser.

Ces imprimantes sont utilisées par les professionnels de l'édition de livres ou de revues, et, à moins que vous ne disposiez d'un très gros budget, il faudra vous contenter d'une imprimante thermique, ou mieux, d'une imprimante matricielle.

Un dernier conseil : demandez toujours une démonstration des possibilités de l'imprimante devant vous. En effet, il n'est pas rare que le vendeur vous parle de possibilités qui n'existent pas sur l'imprimante qu'il essaie de vous vendre. En particulier, regardez si :

- le câble d'alimentation de l'imprimante,
 - le câble de liaison de l'imprimante à l'AMSTRAD,
 - le ruban encreur,
- sont fournis.

8/3.1.1

L'imprimante Citizen 120-D

Vos logiciels vous sont chers, et vous désirez en garder la trace sur papier, ou vous venez de vous procurer un logiciel de DAO (Dessin Assisté par Ordinateur), à moins que ce ne soit un traitement de texte.

Vous voici donc confronté au choix d'une imprimante de qualité si possible, mais aussi adaptée à votre budget. Hélas l'éventail du choix est énorme, et varie d'un prix inférieur à 2 000 francs à plusieurs milliers de francs.

Dans les prix les plus bas, outre les imprimantes AMSTRAD DMP-2000 supplantée par la DMP-2160 (conçues à l'origine pour les CPCs, dont le ruban est peu pratique à installer, et par périodes introuvable chez les revendeurs de matériel micro-informatique), il existe l'imprimante CITIZEN 120-D, très répandue car compatible avec différents types de micro-ordinateurs, et d'un coût équivalent à la DMP, pour une qualité supérieure.

Présentation

De conception sobre, pour une taille minime ($H \times L \times P = 90,5 \times 370 \times 240$ mm) et un poids plume (3,7 kg), elle n'en est pas moins robuste.

Sa tête d'impression supporte 9 aiguilles (bien que l'AMSTRAD ne puisse en commander que 7), et permet une vitesse d'écriture de 120 caractères par seconde en mode normal, et 24 caractères par seconde en qualité courrier.

Elle est configurable selon trois formats EPSON : EPSON FX-#1, EPSON FX-#2, EPSON FX-#3, selon la police de caractères désirée, plus le format IBM.

Son jeu de caractères est composé de 96 caractères standard, et 12 caractères pour 11 pays différents.

Les formats classiques s'y trouvent : Pica, Double Frappe, Gras, Italique, Qualité Courrier (NLQ), Condensé, Elargie, Elite, et même Double Hauteur et Négatif (ou encore inversion de fond, c'est-à-dire caractères Blancs sur fond noir). Le mode Pica est établi à la mise sous tension. La plupart de ces formats se retrouvent au tableau 2. On remarquera qu'il est possible de sélectionner les formats directement à partir du panneau de contrôle en face avant.

L'interligne est sélectionnable en 1/6, 1/8, 7/72^e de pouce, ou encore paramétrable en n/72, n/144, n/126^e de pouce.

La qualité d'impression en graphique est elle aussi paramétrable en densité selon le nombre d'aiguilles utilisées : 480, 576, 720, 960, 1152, 1920 points par ligne (voir différents graphismes obtenus en figures 1 et 2).

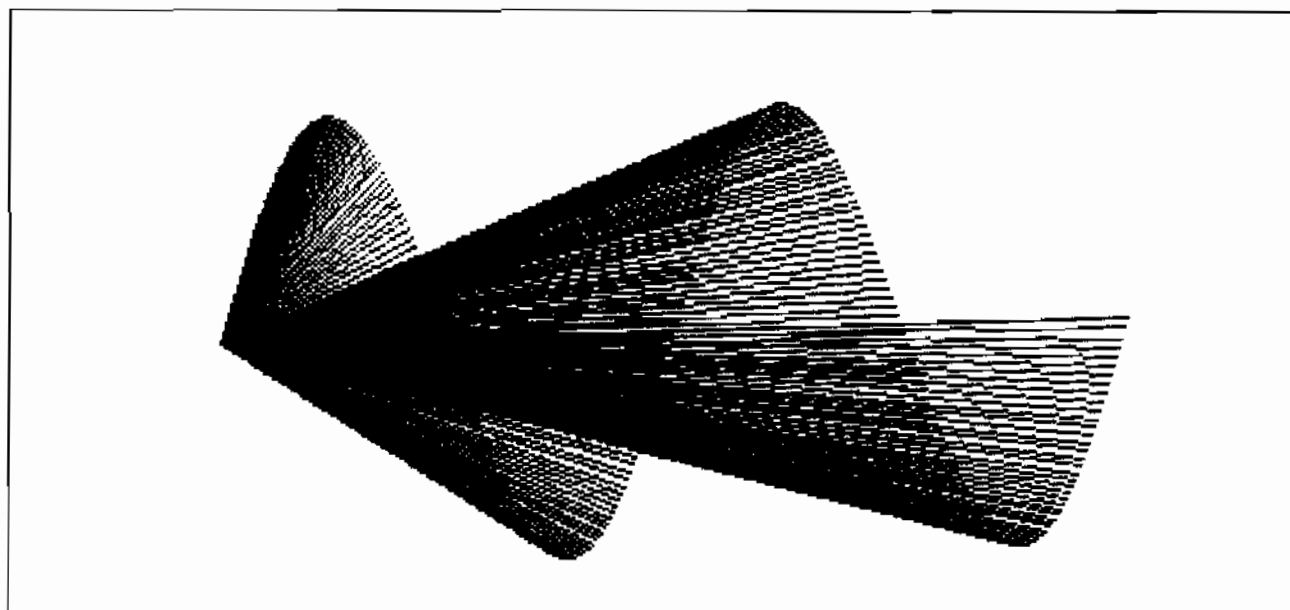


Fig. 1 : Exemple de copie graphique sur la Citizen.

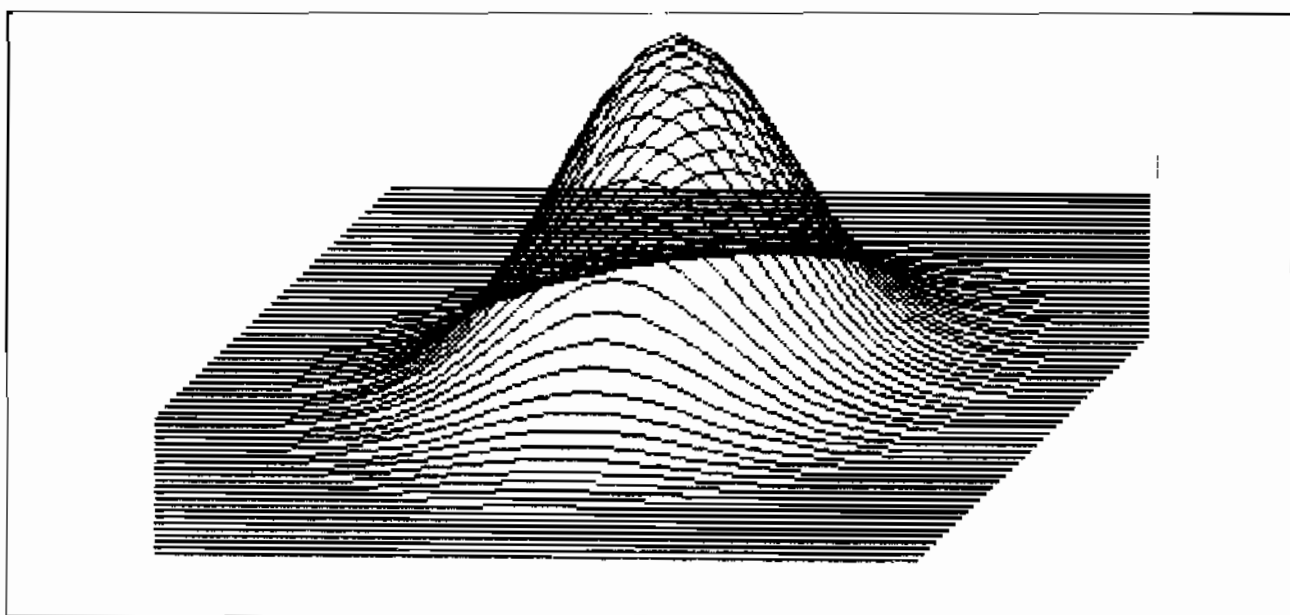


Fig. 2 : Exemple de copie graphique sur la Citizen.

Selon le type d'alimentation en papier (feuille à feuille ou listing continu), un guide et un tracteur à picot sont fournis. Précisons que l'alimentation peut s'effectuer par le dessous de l'imprimante.

En feuille à feuille, le panneau de commande possède une touche permettant le positionnement automatique du papier (« AUTO SHEET LOAD »), fini les feuilles de travers.

Autre avantage fourni avec cette imprimante : une documentation conséquente et détaillée, qui plus est, en vrai français, avec quand même l'inconvénient de s'y perdre un peu, c'est pour cela que nous allons vous donner quelques conseils pour la mise en œuvre de la Citizen 120-D avec l'Amstrad-CPC.

Installation matérielle

Un point noir avec cette imprimante : le câble de liaison.

Lors de l'achat, nous vous conseillons de spécifier que cette imprimante est prévue pour être connectée à un CPC, donc que le câble soit adapté.

Après vérification, s'il vous est impossible d'obtenir le câble adéquat de la part du revendeur, et que l'imprimante vous intéresse absolument, nous vous proposons de réaliser le câble vous-même, avec nos conseils, car la documentation est relativement légère sur ce point.

Vous trouverez tous les détails du câblage en figure 3.

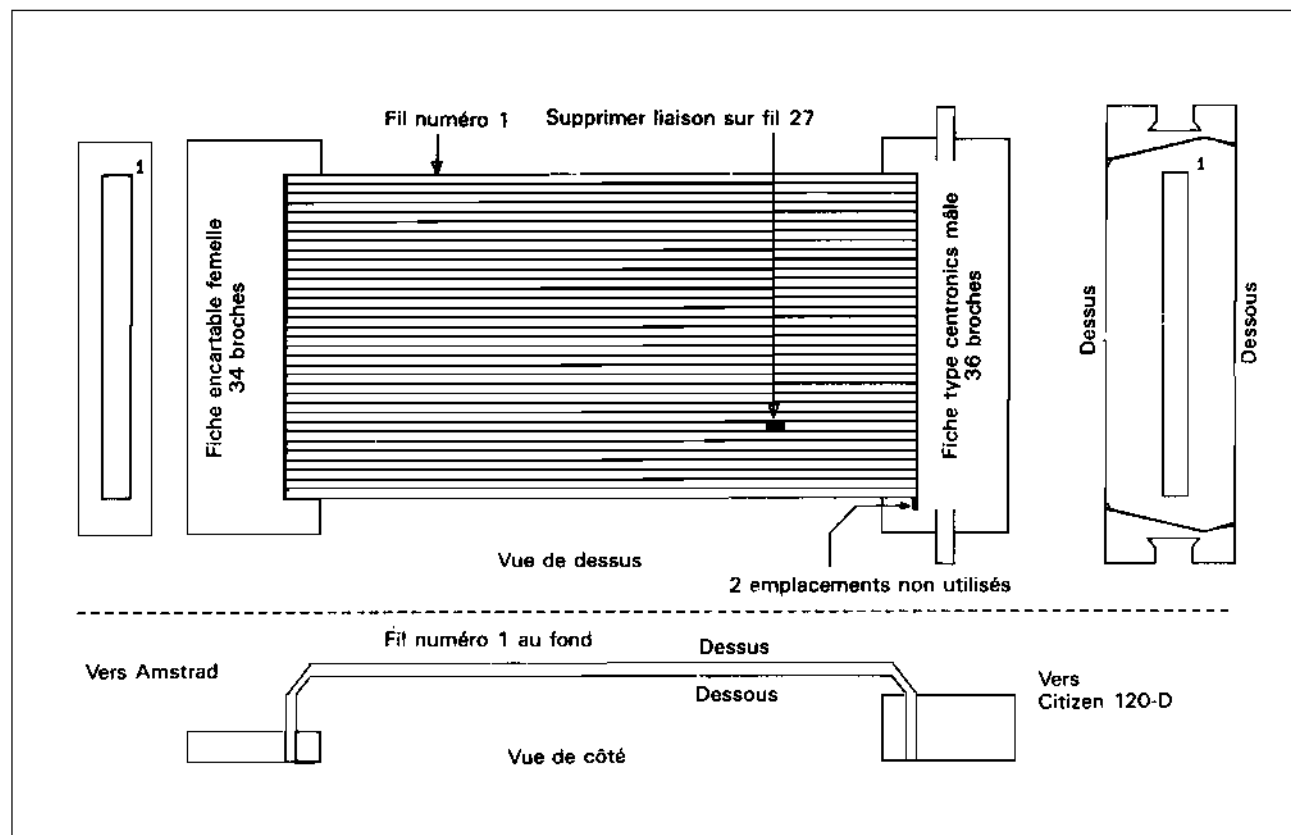


Fig. 3 : Schéma de connexion pour la réalisation du câble de liaison Amstrad CPC-Citizen 120-D.

MATÉRIEL NÉCESSAIRE

- Un connecteur femelle encartable sur circuit imprimé 2 fois 17 broches (ou un connecteur type centronics mâle 2 fois 18 broches pour les versions plus récentes des CPCs) pour l'AMSTRAD.
- Un connecteur mâle de type centronics 2 fois 18 broches pour l'imprimante.
- 1 à 2 mètres (vous choisirez) de câble plat 36 conducteurs plat (50 conducteurs convient aussi, il suffira d'en enlever 14).

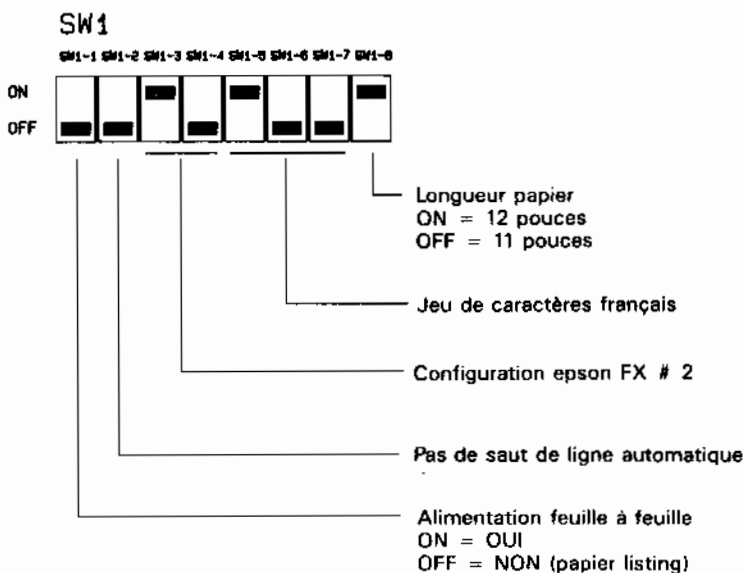
Un conseil, ne conservez que 34 conducteurs si vous possédez une fiche encartable femelle, il restera 2 emplacements libres sur la fiche côté imprimante, prenez soin surtout de respecter le sens des fiches et le repérage de la broche 1.

Vous devrez aussi éliminer la connexion sur le 27^e fil, avec beaucoup de précaution et à l'aide d'un cutter de préférence, sinon l'imprimante générera un saut de ligne supplémentaire automatiquement.

Configuration

Comme nous vous l'avons indiqué précédemment, trois types de format EPSON, et un IBM sont possibles. Nous vous proposons ici d'éviter de fouiller la documentation pour utiliser de suite l'imprimante, en vous fournissant l'état des interrupteurs de l'unique boîtier de la carte d'interface (celle qui se trouve sur le côté, où s'enfiche le câble de liaison).

Ce positionnement fixe l'imprimante en fonctionnement EPSON FX-#2, ce qui garantit une grande facilité d'usage avec les logiciels professionnels (SEWORD, MULTIPLAN...).



Utilisation avec SEMWORD

Le traitement de texte le plus économique sur CPC est le logiciel SEMWORD de la société SEMAPHORE Logiciel, ce qui nous incite à vous proposer les différents codes de contrôle à insérer dans le programme.

Dans le choix C (Changer Programme), nous vous incitons à modifier les paramètres tels que présentés dans le tableau 1 et qui permettront d'obtenir les styles de caractères du tableau 2.

Tableau 1

	27	69		27	70
	27	112 1		27	112 0
	15			18	
	27	71		27	72
	27	77		27	80
	12			12	
	27	83 0		27	84
	27	83 1		27	84
	27	4		27	5
	27	45 1		27	45 0
	15	14		20	18
	14			20	
	27	50		27	50
	27	48		27	50
	27	49		27	50
	27	104		27	117
	27	64		27	64
	27	114		27	116
	27	120 1		27	120 0

Tableau 2

A	Cette ligne est en caractères gras
B	Cette ligne est en caractères proportionnels
C	Cette ligne est en caractères condensés
D	Cette ligne est imprimée en double frappe
E	Cette ligne est en caractères Elite
F	ce code correspond à un saut de page
G	Cette ligne est en caractères surélevés
H	Cette ligne est en caractères abaissés
I	Cette ligne est en caractères italiques
J	<u>Cette ligne est entièrement soulignée</u>
K	Cette ligne est en Condensé-Agrandi
L	Cette ligne est agrandie
M	interligne de 1/6
N	interligne de 1/8
O	interligne de 7/72
G	interligne à 1 initialisation
P	Cette ligne est en double hauteur
Q	ce code ré-initialise l'imprimante
R	Ce code inverse le fond de la ligne
S	
T	Cette ligne est en qualité courrier

N'oubliez pas de sauvegarder SEMWORD pour retrouver la configuration au lancement.

Utilitaire

Nous vous proposons ci-après une extension de plusieurs commandes BASICs permettant de placer directement l'imprimante sous différents types de formats de caractères, ce qui vous permettra d'améliorer la lisibilité de votre programme en supprimant les lignes du type :

```
PRINT #8, CHR$(27);CHR$(120);CHR$(1);CHR$(27);CHR$(45);CHR$(1)
```

Il est en effet plus parlant de frapper :

```
ùNLQ:ùSOU...
```

pour initialiser l'imprimante en qualité courrier soulignée, plutôt que la chaîne ASCII précédente.

Les différentes commandes ajoutées sont :

- ùRST : ré-initialise l'imprimante
- ùNLQ : qualité courrier
- ùELI : élite
- ùDBL : double largeur
- ùCOM : compressé
- ùGRA : gras
- ùITA : italique
- ùSOU : souligné
- ùINV : inversion
- ùSUR : surélevé
- ùABB : abaissé
- ùDBH : double hauteur

Le programme assembleur

Nous vous fournissons le programme assembleur largement commenté.

```

1      ;*****
2      ;* UTILITAIRE D'IMPRESSION *
3      ;*   SUR CITIZEN 120-D   *
4      ;*****
5      ;
6      ;
7      ;*** ORIGINE D'ASSEMBLAGE ***
8      ;
9      ;           ORG 0A000H
10     ;
11     ;*** ORIGINE DE CHARGEMENT ***
12     ;
13     ;           LOAD 0A000H
14     ;
15     ;*** TABLE DES EQUIVALENCES ***
16     ;
17     KLOGTX:    EQU 0BCD1H      ;KL-LOG-EXT
18     BUSY:      EQU 0BD2EH      ;TESTE BUSY IMP
19     LPRINT:    EQU 0BD2BH      ;ENVOI CARACT.
20     ;
21     ;
22     ;*** INSTALLATION INSTRUCTIONS ***
23     ;
24     A000 215AA0  DEBUT:        LD    HL,KERNEL
25     A003 010FA0          LD    BC,VECTEU
26     A006 CDD1BC          CALL KLOGTX
27     A009 3EC9          LD    A,0C9H      ;POUR EVITER UN
28     A00B 3200A0          LD    (DEBUT),A  ;NOUVEL ACCES
29     A00E C9          RET
30     ;
31     A00F 35A0  VECTEU:        DEFW TABLE
32     ;*** TABLE DES SAUTS ***
33     ;
34     A011 C35EA0          JP    RSI      ;REINITIALISATION IMP
35     A014 C364A0          JP    NLQ      ;QUALITE COURRIER
36     A017 C36AA0          JP    ELI      ;ELITE
37     A01A C370A0          JP    DBL      ;DOUBLE LARGEUR
38     A01D C376A0          JP    COM      ;COMPRESSE
39     A020 C37CA0          JP    GRA      ;GRAS
40     A023 C382A0          JP    ITA      ;ITALIQUE
41     A026 C388A0          JP    SOU      ;SOULIGNE
42     A029 C3BEA0          JP    INV      ;INVERSION
43     A02C C394A0          JP    SUR      ;SURELEVE
44     A02F C39AA0          JP    ABB      ;ABAISSSE
45     A032 C3A0A0          JP    DBH      ;DOUBLE HAUTEUR
46     ;
47     ;*** TABLE DES INSTRUCTIONS ***
48     ;
49     TABLE:    EQU $
50     A035 5253          DEFB "RS"
51     A037 D4          DEFB "I"+0B0H
52     A038 4E4C          DEFB "NL"
53     A03A D1          DEFB "Q"+0B0H
54     A03B 454C          DEFB "EL"
55     A03D C9          DEFB "I"+0B0H
56     A03E 4442          DEFB "DB"
57     A040 CC          DEFB "L"+0B0H
58     A041 434F          DEFB "CD"
59     A043 CD          DEFB "M"+0B0H
60     A044 4752          DEFB "GR"

```

```

61 A046 C1 DEF B "A"+080H
62 A047 4954 DEF B "IT"
63 A049 C1 DEF B "A"+080H
64 A04A 534F DEF B "SQ"
65 A04C D5 DEF B "U"+080H
66 A04D 494E DEF B "IN"
67 A04F D6 DEF B "V"+080H
68 A050 5355 DEF B "SU"
69 A052 D2 DEF B "R"+080H
70 A053 4142 DEF B "AB"
71 A055 C2 DEF B "B"+080H
72 A056 4442 DEF B "DB"
73 A058 C8 DEF B "H"+080H
74 A059 00 DEF B 00
75
76 A05A 00000000 KERNEL: DEF B 00,00,00,00
77
78
79
80
81 A05E 21B5A0 RST: LD HL,CRST
82 A061 C3A6A0 JP ENVOI
83 A064 21B8A0 NLQ: LD HL,CNLQ
84 A067 C3A6A0 JP ENVOI
85 A06A 21BCA0 ELI: LD HL,CELI
86 A06D C3A6A0 JP ENVOI
87 A070 21BFA0 DBL: LD HL,CDBL
88 A073 C3A6A0 JP ENVOI
89 A076 21C3A0 COM: LD HL,CCOM
90 A079 C3A6A0 JP ENVOI
91 A07C 21C5A0 GRA: LD HL,CGRA
92 A07F C3A6A0 JP ENVOI
93 A082 21C8A0 ITA: LD HL,CITA
94 A085 C3A6A0 JP ENVOI
95 A088 21CBA0 SOU: LD HL,CSDU
96 A08B C3A6A0 JP ENVOI
97 A08E 21CFA0 INV: LD HL,CINV
98 A091 C3A6A0 JP ENVOI
99 A094 21D4A0 SUR: LD HL,CSUR
100 A097 C3A6A0 JP ENVOI
101 A09A 21D8A0 ABB: LD HL,CABB
102 A09D C3A6A0 JP ENVOI
103 A0A0 21DCA0 DBH: LD HL,CDBH
104 A0A3 C3A6A0 JP ENVOI
105
106
107
108
109 A0A6 7E ENVOI: LD A,(HL)
110 A0A7 FEFF CP OFFH (PLUS DE CARACTERES?)
111 A0A9 C8 RET Z (OUI ALORS FIN)
112 A0AA 23 INC HL (POINTE SUIVANT)
113 A0AB CD2EBD TENTE: CALL BUSY (TESTE IMP LIBRE?)
114 A0AE 38FB JR C,TENTE (NON ALORS RE-ESSAI)
115 A0B0 CD28BD CALL LPRINT (OUI ENVOI CARACT.)
116 A0B3 1BF1 JR ENVOI (AU SUIVANT)
117
118
119
120

```

```

121
122 AOB5 1B40FF CRST:      DEFB 01BH,040H,0FFH
123 AOB8 1B7B01FF CNLQ:    DEFB 01BH,07BH,01H,0FFH
124 AOB0 1B4DFH  CEL1:     DEFB 01BH,04DH,0FFH
125 AOB5 1B5701FF CDBL:    DEFB 01BH,057H,01H,0FFH
126 AOC3 0FFF      CCUM:    DEFB 0FH,0FFH
127 AOC5 1B45FF  CGRA:     DEFB 01BH,045H,0FFH
128 AOC8 1B34FF  C11A:     DEFB 01BH,034H,0FFH
129 AOCB 1B2D01FF CSOU:    DEFB 01BH,02DH,01H,0FFH
130 AOCF 1B7E3201 CINV:    DEFB 01BH,07EH,032H,01H
130 AOD3 FF
131 AOD4 1B5300FF CSUR:    DEFB 01BH,053H,00H,0FFH
132 AODB 1B5301FF CABB:    DEFB 01BH,053H,01H,0FFH
133 AODC 1B68FF  CDBH:     DEFB 01BH,068H,0FFH
134
135                                END

```

Le chargeur Basic

```

10 REM *****
20 REM * CREATION D'INSTRUCTIONS *
30 REM * DE CONTROLE D'IMPRIMANTE *
40 REM *****
50 REM
60 REM *** CHARGEUR BASIC ***
70 ADR = &A000: I = 0
80 RESTORE 280
90 READ A$
100 IF A$ = "XX" THEN 180
110 B$ = "&" + A$
120 B = VAL(B$)
130 POKE ADR,B
140 ADR = ADR + 1
150 I = I + 1
160 GOTO 90
170 REM
180 REM *** SAUVEGARDE ***
190 PRINT "POUR SAUVEGARDER LA ROUTINE E
N BINAIRE"
200 PRINT "SAVE ";CHR$(34);"ROUTINE.BIN"
;CHR$(34);",B,&A000,";RIGHT$(STR$(I),LEN
(STR$(I))-1)
210 PRINT
220 PRINT "MEMORY &9FFF"
230 PRINT "CHARGEMENT PAR LOAD ";CHR$(34)
;"ROUTINE.BIN";CHR$(34);",&A000"
240 PRINT
250 PRINT "INITIALISATION PAR CALL &A000"
"
260 STOP
270 REM
280 REM *** ROUTINE L.M. ***

```

```
290 DATA 21,5A,A0,01,0F,A0,CD,D1
300 DATA BC,3E,C9,32,00,A0,C9,35
310 DATA A0,C3,5E,A0,C3,64,A0,C3
320 DATA 6A,A0,C3,70,A0,C3,76,A0
330 DATA C3,7C,A0,C3,82,A0,C3,88
340 DATA A0,C3,8E,A0,C3,94,A0,C3
350 DATA 9A,A0,C3,A0,A0,52,53,D4
360 DATA 4E,4C,D1,45,4C,C9,44,42
370 DATA CC,43,4F,CD,47,52,C1,49
380 DATA 54,C1,53,4F,D5,49,4E,D6
390 DATA 53,55,D2,41,42,C2,44,42
400 DATA CB,00,00,00,00,00,21,B5
410 DATA A0,C3,A6,A0,21,B8,A0,C3
420 DATA A6,A0,21,BC,A0,C3,A6,A0
430 DATA 21,BF,A0,C3,A6,A0,21,C3
440 DATA A0,C3,A6,A0,21,C5,A0,C3
450 DATA A6,A0,21,CB,A0,C3,A6,A0
460 DATA 21,CB,A0,C3,A6,A0,21,CF
470 DATA A0,C3,A6,A0,21,D4,A0,C3
480 DATA A6,A0,21,DB,A0,C3,A6,A0
490 DATA 21,DC,A0,C3,A6,A0,7E,FE
500 DATA FF,CB,23,CD,2E,BD,3B,FB
510 DATA CD,2B,BD,1B,F1,1B,40,FF
520 DATA 1B,7B,01,FF,1B,4D,FF,1B
530 DATA 57,01,FF,0F,FF,1B,45,FF
540 DATA 1B,34,FF,1B,2D,01,FF,1B
550 DATA 7E,32,01,FF,1B,53,00,FF
560 DATA 1B,53,01,FF,1B,6B,FF
570 REM *** FIN DE DONNEES ***
580 DATA XX
590 END
```

8/3.2

L'interface CENTRONICS

L'interface parallèle CENTRONICS est une des plus utilisées en micro-informatique pour assurer la communication entre le micro et une imprimante parallèle.

Contrairement à l'interface série (de type RS 232, par exemple), les imprimantes de type parallèle ne nécessitent pas de bit start et stop pour assurer la synchronisation entre l'émetteur et le récepteur. Des caractères spéciaux sont utilisés à cette fin.

I. La transmission synchrone

Il existe deux types de protocoles de communication synchrone :

- Protocoles orientés caractères (les données sont des caractères de longueur fixe) : protocole **BSC**.
- Protocoles orientés bits (les données n'ont pas forcément une longueur fixe) : protocole **HDLC**.

PROTOCOLE BSC (BiSync) :

Ce protocole est utilisé pour transmettre des caractères de longueur fixe.

Les données à transmettre à l'imprimante sont insérées dans des blocs appelés *trames*. Une trame doit commencer par les caractères de synchronisation PAD et SYN comme suit :

P	P	S	S	S		D	S		D	I		D	S		D	E		P
A	A	Y	Y	O	Entête	L	T	Données	L	T	BCC	L	T	Données	L	T	BCC	A
D	D	N	N	H		E	X		E	B		E	X		E	B		D

Les caractères de contrôle qui sont employés sur une liaison CENTRONICS sont les suivants :

Caractère	Valeur hexadécimale	Description
SYN	32	Boucle de synchronisation
PAD	55	Début de trame
PAD	FF	Fin de trame
DLE	10	Escape
ENQ	2D	ENQuiry (demande d'état)
SOH	01	Début d'entête
STX	02	Début de texte
ITB	1F	Fin de bloc intermédiaire
ETB	26	Fin de transmission
ETX	03	Fin de texte

PROTOCOLE SDLC (Synchronous Data Link Control) :

Ce protocole est utilisé pour transmettre des données de longueur variable.

Les caractères de contrôle du protocole BSC ne sont plus utilisés pour définir le début et la fin des trames. SDLC utilise des séquences binaires à cette fin. Une trame SDLC typique a la structure suivante :

Drapeau de départ	Champ d'adresse	Champ de contrôle	Données	Vérification cyclique	Vérification cyclique	Drapeau de fin
-------------------	-----------------	-------------------	---------	-----------------------	-----------------------	----------------

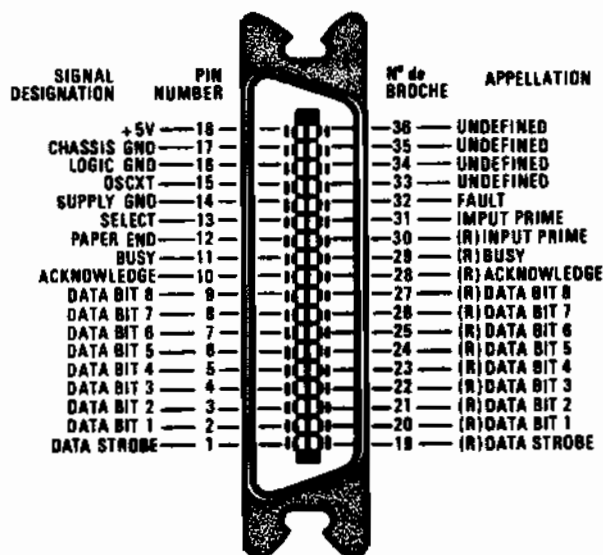
II. Brochage de l'interface CENTRONICS

L'interface CENTRONICS comporte 36 broches dont un bus parallèle de 8 bits de données.

L'ordinateur peut contrôler le flux des données grâce au signal **STROBE**. L'imprimante fait savoir qu'une trame a bien été reçue par le signal **ACKNOWLEDGE**, et fait savoir qu'un caractère est en cours d'impression par le signal **BUSY**.

Les signaux de l'interface CENTRONICS suivent le standard TTL.

Connecteur parallèle (Centronics)



Ⓜ INDICATES SIGNAL GROUND RETURN

BROCHE	NOM	SENS	BROCHE	NOM	SENS
1	STROBE	→	19	Masse	—
2	DATA 1	→	20	"	—
3	DATA 2	→	21	"	—
4	DATA 3	→	22	"	—
5	DATA 4	→	23	"	—
6	DATA 5	→	24	"	—
7	DATA 6	→	25	"	—
8	DATA 7	→	26	"	—
9	DATA 8	→	27	"	—
10	ACK	←	28	"	—
11	BUSY	←	29	"	—
12	PE	←	30	"	—
13	SEL	←	31	INIT	→
14	Masse	—	32	ERROR	←
15	?	—	33	Masse	—
16	Masse	—	34	CLK	←
17	CHASSIS	—	35	TEST	→
18	+ 5 V	←	36	?	—

? : utilisation qui dépend de l'imprimante → : de l'ordinateur vers l'imprimante ← : de l'imprimante vers l'ordinateur

8/3.3

Le câble de liaison : achat ou réalisation

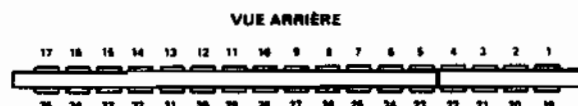
Si les quelques deux cent cinquante francs nécessaires pour l'achat du cordon de liaison entre CPC et l'imprimante vous semblent un peu trop élevés, et si vous êtes un peu bricoleur (bricoleuse), vous pouvez réaliser assez simplement ce câble.

Pour cela, il faut vous munir de :

- un connecteur femelle 34 broches au pas de 2.54 mm pour la connexion côté AMSTRAD,
- un connecteur mâle de type CENTRONICS pour la connexion côté imprimante,
- un câble plat de 10 fils de la longueur que vous jugez nécessaire (jusqu'à 4 m).

Le schéma du connecteur côté AMSTRAD est le suivant :

Porte pour imprimante



PIN 1	STROBE	PIN 19	GND
PIN 2	D0	PIN 20	GND
PIN 3	D1	PIN 21	GND
PIN 4	D2	PIN 22	GND
PIN 5	D3	PIN 23	GND
PIN 6	D4	PIN 24	GND
PIN 7	D5	PIN 25	GND
PIN 8	D6	PIN 26	GND
PIN 9	GND	PIN 27	GND
PIN 11	BUSY	PIN 33	GND
PIN 14	GND		
PIN 16	GND		
Autres broches : NC			

Le schéma du connecteur côté imprimante est le suivant :
(voir schéma du Connecteur parallèle du chapitre précédent 8/3.2)

8/4

La souris AMX mouse

Ce produit proposé par ADVANCED MEMORY SYSTEMS LTD comprend :

- une souris équipée de trois boutons (exécution de la commande, annulation de la commande et déplacement sans tracé),
- une interface entre la souris et le port JOYSTICK de l'AMSTRAD alimentée par la prise 5 volts qui sort du moniteur.
- une cassette sur laquelle se trouvent les logiciels de gestion de la souris,
- un manuel d'utilisation de la souris, des logiciels associés et des commandes spécifiques.

La manipulation de la souris est agréable, et les logiciels qui sont fournis sur la cassette sont de bonne qualité. De plus, il est possible d'utiliser la souris :

- dans un programme écrit en BASIC, grâce à des instructions RSX dont la liste est donnée dans le manuel d'utilisation,
- dans un programme écrit en ASSEMBLEUR en se servant des RSX dont les points d'entrée et les conditions d'entrée sont donnés dans le manuel d'utilisation.

Nous pouvons donc dire que AMX MOUSE est un bon produit qui apporte un plus à la série des CPC. Malheureusement, comme tous les utilisateurs de CPC ne possèdent pas de souris, les logiciels qui ont été développés dans le commerce ne font pas très souvent appel aux possibilités supplémentaires qu'offre l'utilisation d'une souris...

Le logiciel fourni avec AMX MOUSE se compose de :

- AMX CONTROL,
- AMX ART,
- ICON DESIGNER,
- PATTERN DESIGNER.

AMX CONTROL

Il contient l'ensemble des commandes RSX (mots-clés incorporés au BASIC précédés d'une barre verticale) qu'un programmeur BASIC ou ASSEMBLEUR pourra activer très simplement.

Les RSX définis dans AMX CONTROL sont les suivants :

- redéfinition des codes produits par la souris,
- définition de la couleur de fond d'écran,
- position graphique de la souris,
- élimination d'un pointeur d'écran et rétablissement du contenu de l'écran sous le pointeur,
- affichage d'icônes sur l'écran,
- chargement d'icônes stockés sur disque et définis par ICON DESIGNER,
- transformation des mouvements de la souris en codes curseur plutôt qu'en coordonnées graphiques ou inversement,
- activation ou désactivation de la souris,
- déplacement du pointeur sur l'écran,
- changement de la forme du pointeur sur l'écran,
- définition du pas de déplacement de la souris,
- affichage du pointeur aux coordonnées courantes de la souris,
- déplacement du pointeur à des coordonnées spécifiées,
- gestion de fenêtres du même type que celles utilisées dans ICON DESIGNER.

AMX ART

C'est un utilitaire de dessin perfectionné qui utilise pleinement les possibilités de la souris.

Lorsque ce logiciel est en mémoire, vous pouvez distinguer 4 parties différentes sur l'écran : la barre de menu, les modes d'utilisation de la souris, la zone des motifs et enfin l'écran.

Première partie :

Une **barre de menu**, en haut de l'écran, qui comporte les options suivantes :

- **FILE**

- **LOAD PIC**

Permet de charger une image stockée sur support magnétique (cassette ou disquette) en mémoire d'écran.

- **SAVE PIC**
Permet de sauvegarder l'image qui se trouve sur l'écran sur un support magnétique cassette ou disquette.
- **EPSON**
Permet de faire une « hard-copy » de l'écran graphique.
- **LOAD FILLS**
Permet de charger un ensemble de motifs définis par PATTERN DESIGNER dans la zone réservée aux motifs.
- **DIRECTORY**
Donne la liste de tous les fichiers présents sur la disquette courante.

- **OPTIONS**

- **CLEAR**
Permet d'effacer l'écran.
- **GRIDBLOCK**
Permet de définir le pas de progression de la souris sur l'écran pour certaines options.
- **ITAPE**
Pour sélectionner les lectures/écritures de fichiers sur cassette.
- **IDISC**
Pour sélectionner les lectures/écritures de fichiers sur disquette.
- **COPY**
Permet de dupliquer une zone de l'écran.
- **FAST**
La souris se déplace rapidement.
- **MEDIUM**
La souris se déplace à la vitesse moyenne.
- **SLOW**
La souris se déplace lentement.
- **PAINT**
Permet de peindre une zone de l'écran délimitée par un contour fermé.
- **ZOOM**
Permet de retoucher une partie du dessin pixel par pixel.

- **TEXT**

- **PLAIN**
Caractères ordinaires.

- **BOLD**
Caractères gras.
- **V. BOLD**
Caractères très gras.
- **ITALIC**
Caractères en italique, ordinaires, gras ou très gras.

• **LINES**

- **BLACK**
Les traits et encadrés sont tracés en noir.
- **INVERT**
Les traits ou encadrés remplis sont inversés.
- **WHITE**
Les traits ou encadrés sont tracés en blanc.
- **PEN 2**
La couleur PEN 2 est utilisée pour les traits ou encadrés.
- **PEN 3**
La couleur PEN 3 est utilisée pour les traits ou encadrés.

Deuxième partie :

Différents **modes d'utilisation** possibles de la souris.

Ces modes sont :

- « crayon » pour dessiner des traits,
- « pistolet » pour pulvériser un motif sur l'écran,
- « gomme » pour effacer une partie du dessin sur l'écran,
- « rouleau » simule le passage d'un rouleau à peinture sur l'écran,
- « remplissage » permet de remplir une zone de l'écran avec un des motifs prédéfinis dans la fenêtre des motifs graphiques,
- « encadré » permet de définir un rectangle non rempli,
- « encadré rempli » permet de définir un rectangle rempli avec l'encre courante (en NOIR, en mode INVERSE, en BLANC, PEN 2 ou PEN 3),
- « cercle » permet de définir des cercles de dimension quelconque.
- « texte » pour afficher du texte sur l'écran. Ce texte peut être écrit avec les lettres normales, en gras ou en italique.

Troisième partie :

Zone des motifs : cette zone contient les motifs qui seront utilisés par les options de remplissage, de pistolage, de rouleau et d'encadré rempli.

Quatrième partie :

L'**écran** sur lequel vous pourrez dessiner, charger des dessins, des icônes, écrire du texte, etc.

ICON DESIGNER

C'est un programme utilitaire qui permet de dessiner des icônes utilisables par un programme écrit par vous, en BASIC ou en ASSEMBLEUR.

L'écran ICON DESIGNER est composé de huit parties différentes :

- La zone de dessin qui représente l'icône courant en gros plan (grille 16 × 16 pixels),
- La zone de mise en fichier qui représente les icônes qui seront sauvegardés dans le fichier d'icônes,
- La zone de visualisation des symboles qui permet de voir l'icône dans sa grandeur réelle,
- La zone de mémoire de travail qui permet de manipuler 25 icônes,
- La provenance de l'icône en cours d'édition,
- Le numéro du symbole courant qui donne le numéro de l'icône actuellement en cours de traitement dans la zone de dessin,
- La zone des messages active lorsque les entrées/sorties sur cassette ou disquette sont sélectionnées,
- Les icônes de fonction qui permettent :
 - de faire des entrées/sorties sur cassette ou disquette,
 - de définir jusqu'à 320 icônes,
 - de vider la grille de dessin,
 - d'accéder à des commandes d'entrée/sortie supplémentaires telles que I DISC.IN, ERASE, etc.,
 - d'effacer des icônes chargés par erreur.

PATTERN DESIGNER

C'est un programme utilitaire qui permet de dessiner des icônes utilisables par AMX ART. Il est en tout point semblable à ICON DESIGNER.

8/5

Systèmes et méthodes de transfert de fichiers

La multiplicité des standards d'ordinateur et de mémoires de masse entraîne souvent des difficultés énormes de transfert de fichiers d'un ordinateur vers un autre.

D'autre part, nombre de systèmes de banques de données se sont créés depuis peu et l'accès à ceux-ci pose certains problèmes qu'il est bon, au minimum de connaître, à défaut de ne savoir les résoudre.

L'objet de ce chapitre est précisément de réunir toutes informations utiles à la solution de ces problèmes, que ce soient des informations sur les matériels à mettre en œuvre, les normes régissant ce type d'opérations ou les méthodes, astuces et difficultés particulières liées à ces transferts.

La norme la plus courante dans ce domaine est la norme RS-232, (voir Partie 8, chap. 5.2). Les modems, maillons indispensables de la chaîne télématique, seront également décrits, d'abord de façon générale puis dans le détail, au travers d'études réalisées sur divers modems du commerce.

8/5.1

Copies et transmissions de cassettes

Caractéristiques

Comme beaucoup d'autres ordinateurs de la même catégorie, les Amstrad enregistrent leurs cassettes selon le procédé « FSK » (*Frequency Shift Keying*) : c'est-à-dire modulation par saut de fréquence).

Pour tout programme ou fichier se présentant dans la mémoire de la machine sous la forme d'une longue liste de « 1 » et de « 0 », il a été imaginé d'affecter une tonalité de fréquence F_0 au zéro, et une autre de fréquence F_1 au un.

Pour peu que ces fréquences ne soient ni trop basses ni trop hautes, elles peuvent facilement être enregistrées et relues par des magnétophones de qualité très moyenne.

Bien que l'écoute d'une cassette « informatique » sur un magnétophone « audio » ne soit guère agréable à l'oreille, le contenu de la bande ne diffère pas techniquement d'un enregistrement de musique ou de paroles, ce qui permet d'envisager des opérations du type copie ou transmission par téléphone ou radio.

Attention, il vous faudra résister à la tentation de copier des logiciels du commerce dûment protégés, et faire preuve de beaucoup de minutie dans vos manipulations !

N'espérez pas cependant échanger des logiciels ou des fichiers sur cassettes avec des utilisateurs équipés d'ordinateurs d'autres marques : il existe presque autant de « formats » d'enregistrement que de types de « micros », même si quelques formes de standardisation ont été définies (Kansas-City ou Basicode, par exemple).

A part les questions purement logicielles de signification des 1 et des 0 (organisation des données enregistrées), deux caractéristiques fondamentales sont attachées à chaque standard d'enregistrement :

- fréquences audio correspondant respectivement au 0 et au 1.
- vitesse de succession des 1 et des 0, exprimée en bauds : à la vitesse de 1000 bauds, par exemple, mille 1 ou 0 sont enregistrés ou lus chaque seconde !

Le cas de l'Amstrad est intéressant à approfondir, du fait de ses particularités : sauf ordre contraire (SPEED WRITE 1), l'enregistrement se fait à 1 000 bauds à l'aide de fréquence d'à peu près 500 et 1 000 Hz.

En vitesse rapide, tout est doublé : 2 000 bauds et des fréquences d'environ 1 000 à 2 000 Hz, exactement comme si l'on « passait la bande » à vitesse double.

En lecture, l'ordinateur reconnaît lui-même la vitesse utilisée lors de l'enregistrement, ce qui nous a donné l'idée suivante : nous avons recopié une cassette sur une bande magnétique en bobine tournant à 9,5 cm/s, avant de la copier à nouveau sur une cassette, mais en lisant la bande intermédiaire à 19 cm/s. Le résultat obtenu est riche d'applications : la cassette ainsi « accélérée » se charge deux fois plus vite que l'originale, et peut être « produite » en série (intéressant si vous envisagez de commercialiser vos meilleurs logiciels...).

Encouragé par ce premier succès, nous avons tenté de quadrupler la vitesse en faisant tourner la bande mère à 38 cm/s : là c'est un peu juste, mais faisable.

Autre tentative intéressante, essayer des vitesses intermédiaires : le programme se charge tout aussi bien si on accélère la bande de 30 %, par exemple !

En fait, l'Amstrad accepte de charger des cassettes enregistrées à pratiquement n'importe quelle vitesse comprise entre deux limites qui dépendent surtout de la qualité du matériel utilisé. Cela signifie qu'il n'y a aucune difficulté particulière à copier des cassettes informatiques par voie « audio », du moins avec du matériel de bonne qualité (c'est d'ailleurs ainsi que sont fabriquées les cassettes logicielles du commerce...).

Paradoxalement, la qualité est souvent meilleure en accéléré qu'à vitesse normale : 1 000 et 2 000 Hz sont des fréquences plus proches de la meilleure zone de fonctionnement des magnétophones que 500 et 1 000 Hz.

L'important est de travailler avec du matériel de bonne qualité et si possible monophonique. Transiter par un magnétophone à bobines est l'idéal, mais on peut aussi accoupler deux bonnes platines à cassette : il faudra alors dans certains cas intercaler un « égaliseur » (ou « equalizer ») afin que les deux fréquences soient reproduites exactement au même niveau (contrôle au VU-mètre ou mieux à l'oscilloscope).

La figure 1 résume les branchements à effectuer pour le procédé avec « bande-mère », le plus « professionnel » et aussi le plus recommandable pour tous travaux de duplication en nombre.

La figure 2 décrit la solution « amateur » utilisant deux platines cassette pour opérer une copie en une seule étape.

Plus encore que lors de copies de cassettes musicales, un grand soin est de rigueur : propreté et bon état des appareils, bande de bonne qualité, réglage rigoureux des niveaux (juste avant la « zone rouge »), mais surtout pas dedans.

Ne perdons pas de vue, en effet, qu'à une vitesse de bande de 4,75 cm/s dans la cassette et pour un débit d'informations de 2 000 bauds, un « bit » (1 ou 0) ne dure que 0,5 ms et occupe une longueur de bande de ...24 μ (millièmes de millimètre) ! Le moindre défaut, à peine perceptible sur de la musique, peut modifier l'état d'un bit et compromettre le chargement du programme !

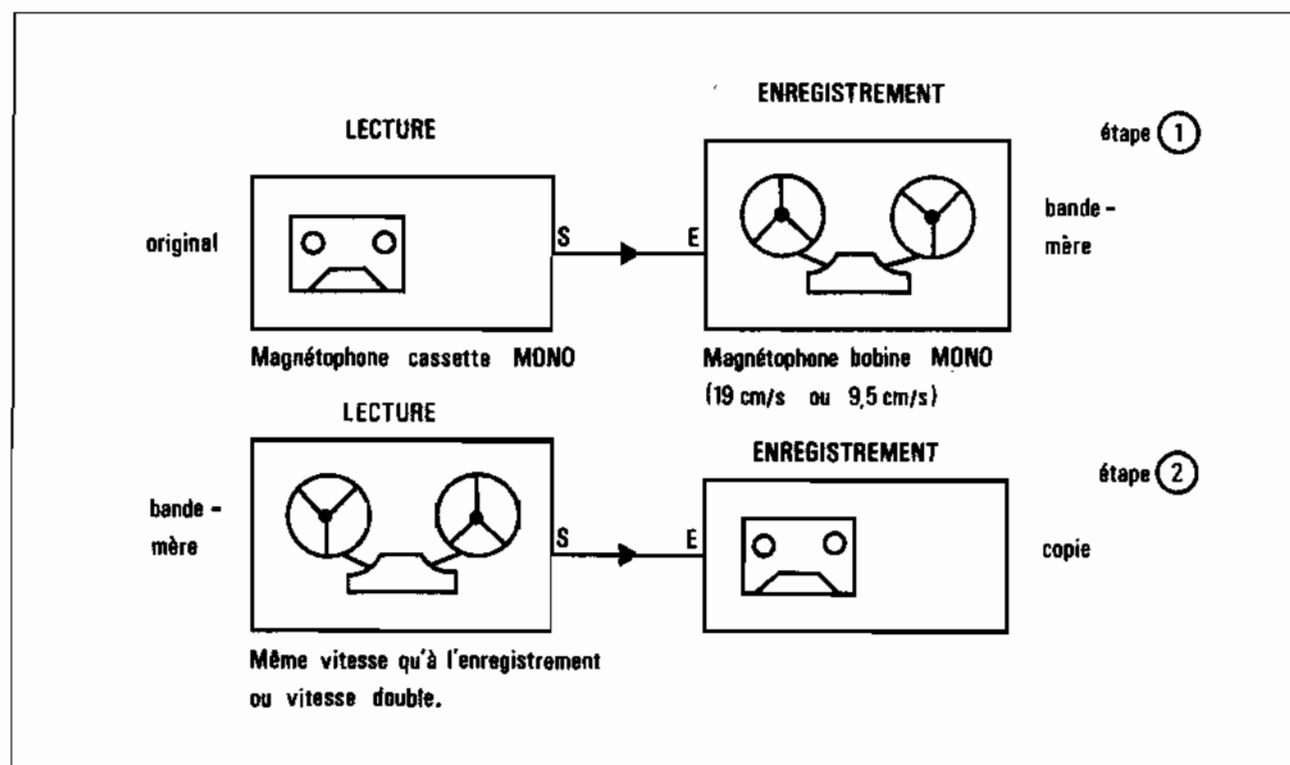


Fig. 1

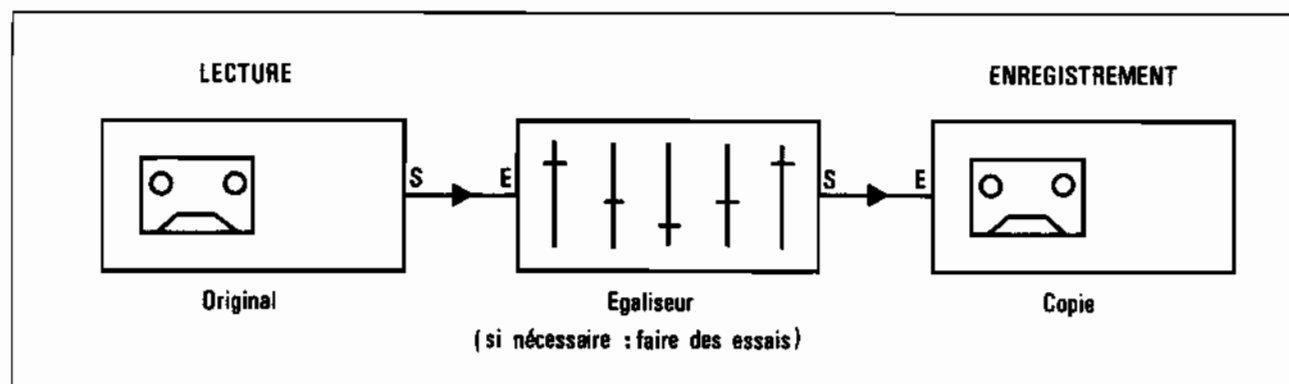


Fig. 2

Moyens de transmission

Puisque les données informatiques enregistrées sur les cassettes d'Amstrad peuvent être manipulées comme du son, il n'est pas interdit de songer à les transmettre par les divers moyens de télécommunication qui s'offrent à nous.

Les transmissions directes entre ordinateurs sont courantes grâce aux modems, appareils utilisant précisément la technique « FSK » et des vitesses de l'ordre de 1 200 bauds. Pour notre part, nous envisagerons plutôt la transmission de cassettes logicielles par des moyens plus rapides que le courrier, et permettant éventuellement la distribution simultanée d'un même programme à de multiples destinataires.

Des expériences très positives ont d'ailleurs été faites par la station de radio hollandaise NOS, qui a diffusé sur son antenne « ondes moyennes » des logiciels qui ont été reçus par des milliers d'auditeurs dans toute l'Europe de l'Ouest et jusqu'en Algérie ! Tout cela pour montrer qu'il est très faisable de copier une cassette à des dizaines ou centaines de kilomètres de distance soit par radio, soit par téléphone.

La transmission de programmes informatiques est une nouvelle activité qui intéresse beaucoup les radio-amateurs, les cibistes et même certains responsables de radios locales privées (FM). Une cassette informatique d'Amstrad peut être transmise exactement comme une cassette de musique ou de paroles : simplement, il faut que la liaison soit excellente, sans parasite, c'est-à-dire à relativement courte distance et avec du bon matériel d'émission et de réception.

La figure 3 donne un exemple d'installation permettant, avec un minimum de soins, la transmission de programmes à l'aide d'émetteurs-récepteurs « CB » : en fait, de simples atténuateurs à résistances suffisent pour adapter les niveaux de signal rencontrés sur les « radios » et sur les magnétophones !

A la figure 4, nous découvrons un procédé rustique mais utile permettant de transmettre des programmes par téléphone. Bien évidemment, la qualité de la liaison devra être bonne, ce qui est plus ou moins fréquent (ou rare !) selon les régions.

Le principe de l'opération consiste à démonter l'écouteur supplémentaire de chacun des postes téléphoniques ordinaires utilisés, ce qui libère deux fils à chaque bout de la ligne.

Sous réserve qu'un silence rigoureux règne au voisinage du combiné décroché (le calfeutrer si nécessaire !), tout signal audio injecté dans ces fils à un bout de la liaison pourra être récupéré à l'autre bout s'il ne se compose que de fréquences comprises entre 300 et 3 400 Hz (la « grande vitesse » est donc parfaitement utilisable pour diviser la note par deux !)

Le niveau reçu est souvent un peu fort pour une entrée MICRO de magnéto-cassette, mais un atténuateur très simple suffit pour que tout rentre dans l'ordre. Un branchement direct peut par contre être effectué sur une entrée « ligne » (LINE IN).

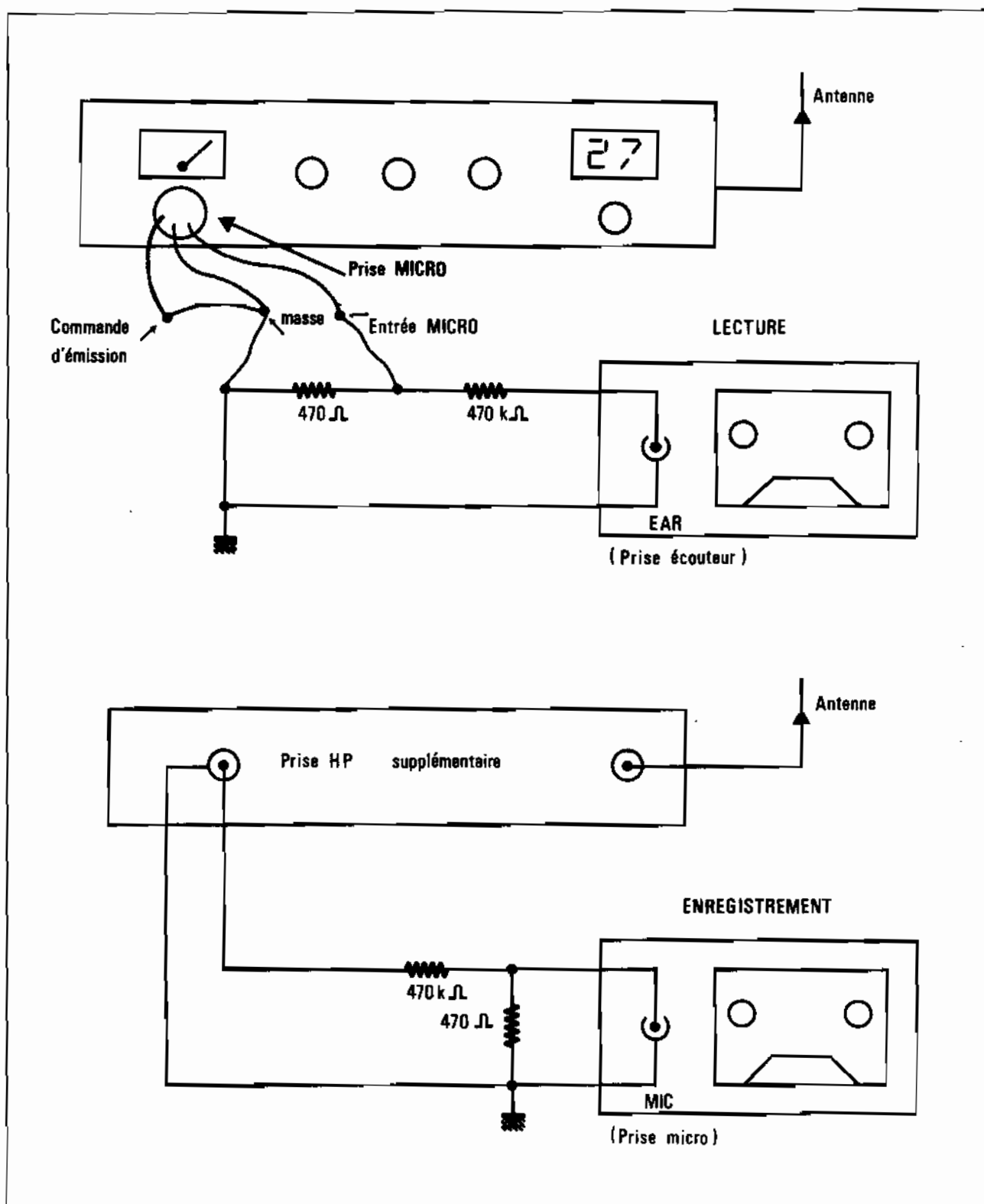


Fig. 3

Il importe de régler le niveau de lecture du magnétophone transmetteur pour que le son perçu dans le combiné ne soit ni plus fort, ni plus faible que pour une conversation normale. Du point de vue de la réception, il faudra régler avec soin le niveau d'enregistrement, sauf s'il est automatique.

Précisons bien que toutes ces idées, même si elles ont été soigneusement vérifiées « sur le terrain », ne sont pas à considérer comme des recettes « clés en main » : les conditions locales sont trop variables, tant en radio qu'en téléphonie.

Elles doivent être donc considérées comme des axes de recherche qu'il faut adapter à votre cas personnel. C'est précisément le plus passionnant...

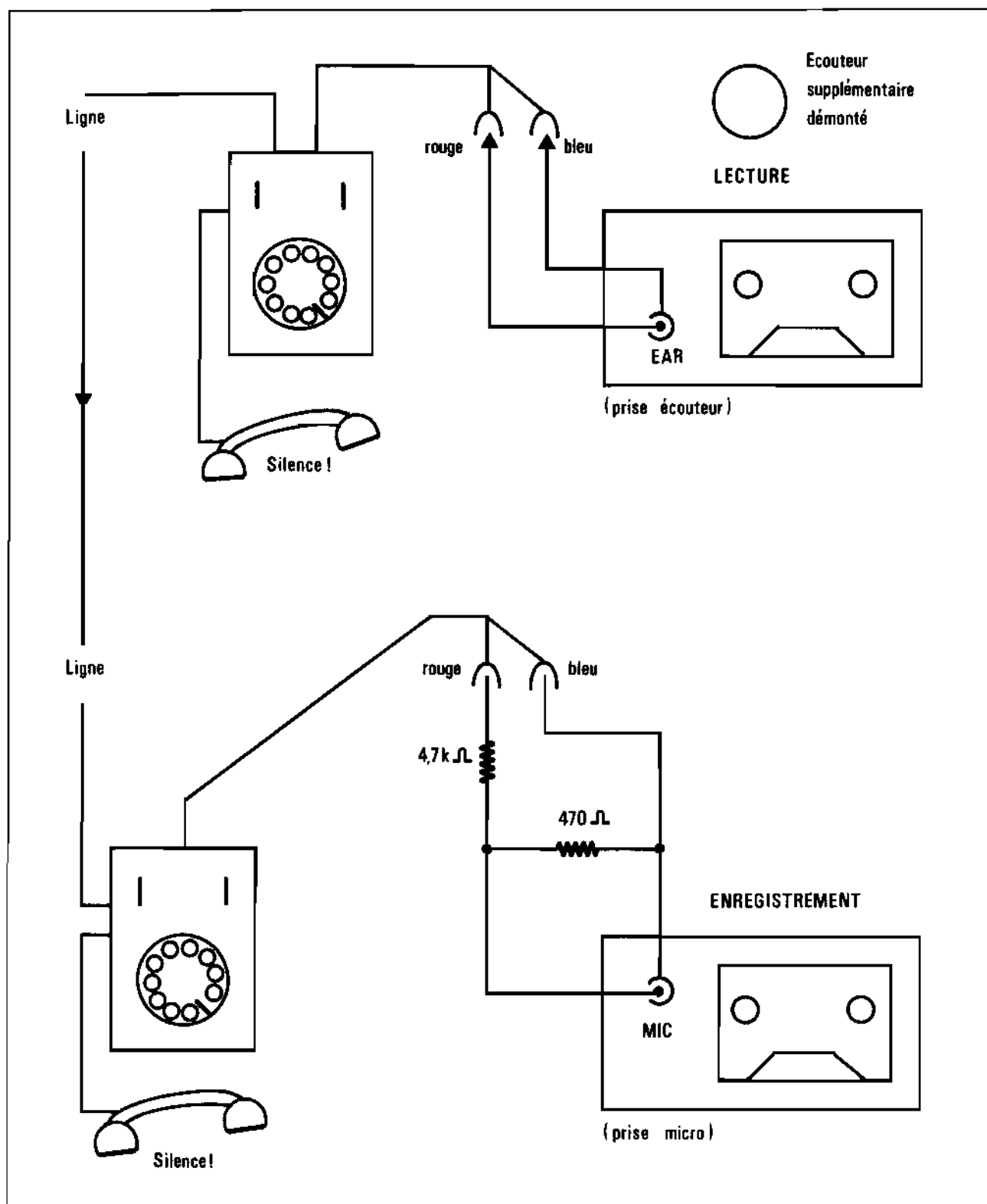


Fig. 4

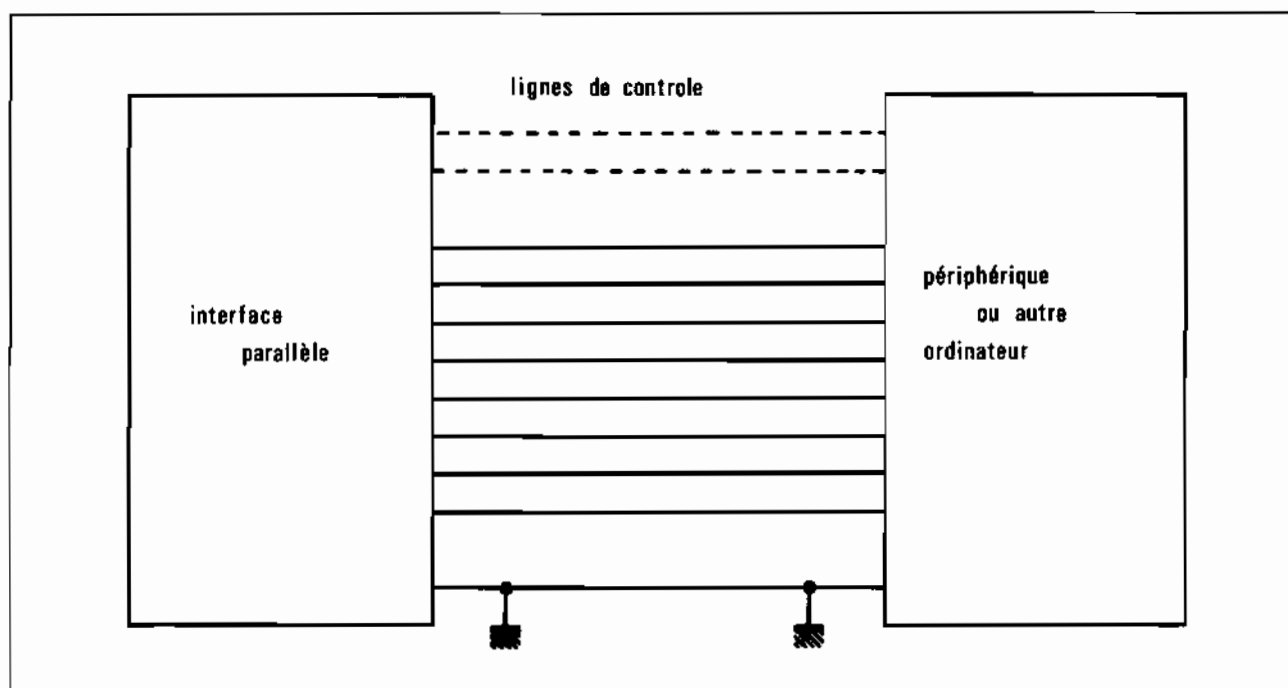
8/5.2

L'interface RS-232

En informatique les échanges entre un ordinateur et un périphérique ou un autre ordinateur s'effectuent à l'aide de mots de huit bits.

Ces bits sont caractérisés par un niveau de tension : en logique dite positive, un niveau haut sera équivalent à un « 1 » logique, et un niveau bas (en général la masse) pour un « 0 » logique.

Dans une transmission parallèle, on est obligé d'utiliser autant de fils qu'il y a de bits par mots, plus un fil pour la référence de masse, ce qui nous en fait neuf pour la transmission en huit bits.



La liaison de type parallèle permet de véhiculer tous les bits d'un même mot en une seule fois, on transmettra les mots en série.

Plusieurs types de liaisons existent, chacune possédant sa norme de transmission : HP-IB, IEEE-488, Multibus Intel, STD Prolog, Centronics...

Aux fils de liaisons des données, il faut de plus ajouter des connexions permettant le contrôle du transfert.

Lors d'une liaison sur une grande distance d'un organe émetteur et d'un organe récepteur, le coût du câblage devient vite très important, il fut nécessaire de créer une liaison de type série.

8/5.2.1

La liaison RS-232

Plusieurs types de liaisons sont apparues. Nous nous intéresserons plus particulièrement à la liaison dénommée RS-232.

Cette liaison fut à l'origine conçue dans le but de relier un terminal à un modem. L'utilisation de la RS-232 est tout de même étendue à la liaison d'un micro-ordinateur avec un périphérique ou un autre micro-ordinateur.

Principe de la RS-232

La RS-232 permet de connecter deux ensembles informatiques à l'aide d'un nombre minimal de fils (trois suffisent, plus des lignes de contrôle si besoin est).

Ces trois fils sont composés de :

- la référence des signaux (masse) ;
- la ligne émission de données ;
- la ligne réception de données.

Un mot de huit bits (ou caractère) à émettre est envoyé par le micro-ordinateur à l'interface RS-232 qui le « découpe » bit par bit et en assure l'émission en série sur le seul fil émission de données. Un bit indiquant le début du mot et un ou deux bits indiquant la fin l'accompagnent. On pourra ajouter au mot un bit de parité permettant de vérifier la bonne transmission du mot par le récepteur.

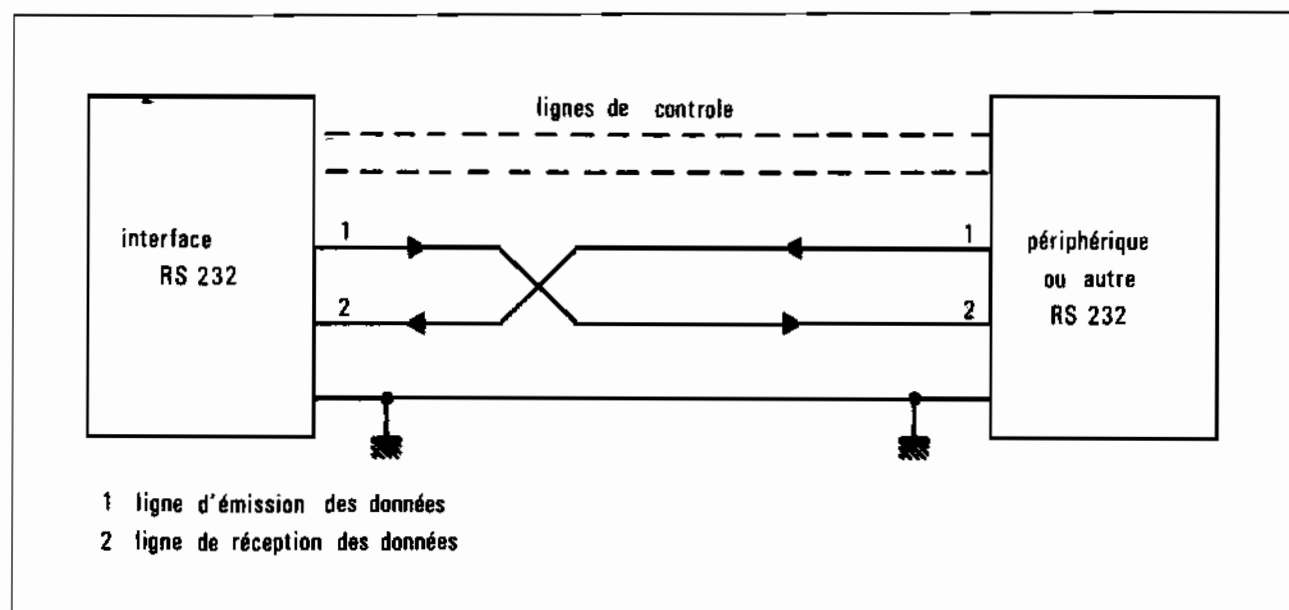
Par le principe inverse, le récepteur lorsqu'il reçoit le bit lui indiquant l'arrivée d'un mot (sur sa ligne réception de données) va acquérir un à un les bits du mot jusqu'au(x) bit(s) lui indiquant la fin, puis le reconstituer en un mot parallèle.

Niveaux de tensions utilisés

Le niveau de tension définissant le niveau logique « 0 » est compris entre +3 et +15V.

Le niveau de tension définissant le niveau logique « 1 » est compris entre -3 et -15V.

Si des lignes de contrôle sont utilisées pour assurer la liaison, elles sont en général considérées comme actives lorsqu'elles sont placées au niveau logique « 0 ».

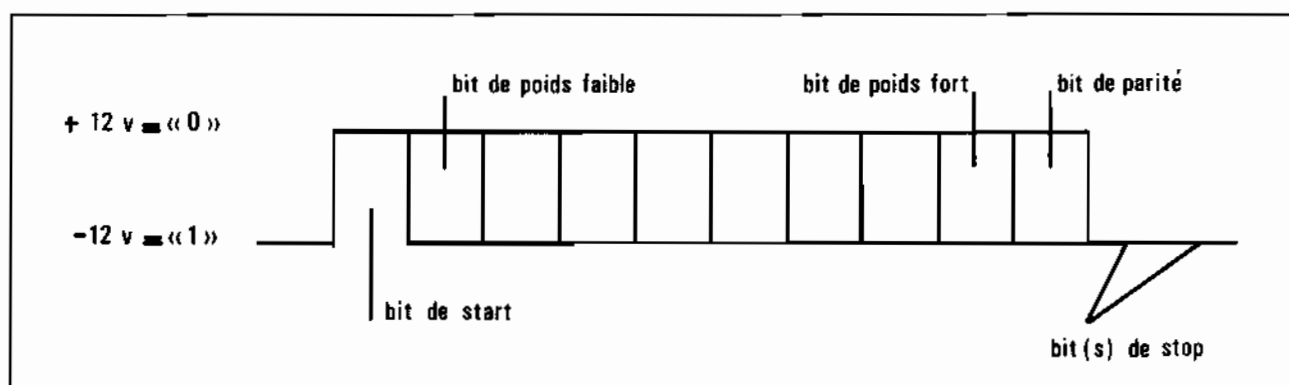


Format de la transmission

Une ligne de transmission est considérée au repos lorsqu'elle est placée à l'état logique « 1 ».

La transmission d'un caractère commence lors d'une transition de l'état logique « 1 » à l'état logique « 0 » (appelé bit de START).

Après la transmission de sept ou huit bits composant le mot, un temps mort de un ou deux bits (appelé(s) bit(s) de STOP) est nécessaire pour permettre au récepteur de reconstituer le mot.



Protocoles d'échanges

Un protocole d'échanges de données est un « accord » entre les deux organes informatiques afin de s'échanger les données.

Trois protocoles d'échanges sont généralement reconnus :

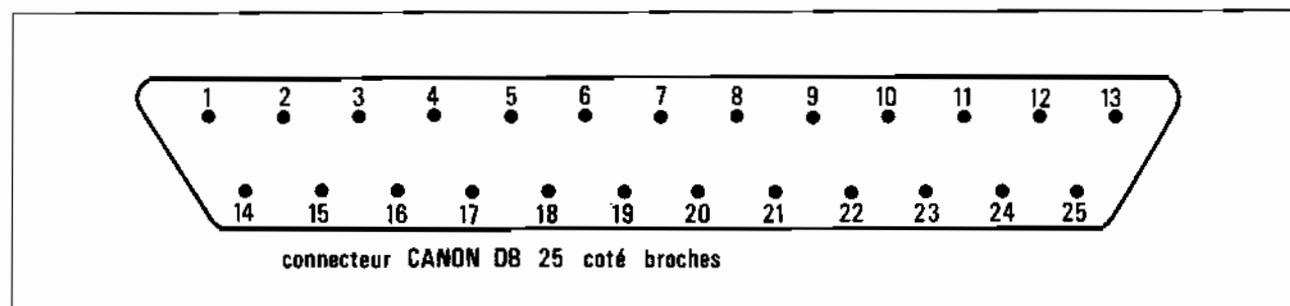
- en mode semi-duplex, l'émetteur envoie un lot de données (par exemple 128 mots) puis attend un ordre du récepteur pour reprendre l'émission avec le lot suivant. En mode ETX/ACK, le mot ETX (\$03) est envoyé après chaque lot, puis l'émetteur attend le mot ACK (\$06) pour émettre le lot suivant ;
- en mode duplex XON/XOFF, le récepteur ordonne à l'émetteur d'arrêter la transmission par l'envoi du caractère XOFF (\$13), la transmission ne reprendra que lors de la réception du caractère XON (\$11) par l'émetteur ;
- un contrôle de la transmission peut s'effectuer au niveau matériel par l'utilisation de lignes de contrôle supplémentaires (en général CTS et DTR, expliquées plus loin) pour bloquer ou rétablir la transmission.

Vous pouvez si vous le désirez utiliser d'autres protocoles d'échanges que vous définirez par vous-même ; mais lors de l'utilisation d'un matériel commercialisé, vous serez obligé de vous soumettre au protocole du constructeur (pour une imprimante série par exemple).

Le connecteur RS-232

Le connecteur standard

Le connecteur standard d'une interface RS-232 est une fiche type CANON 25 broches (DB25) sur laquelle sont présents les signaux suivants :

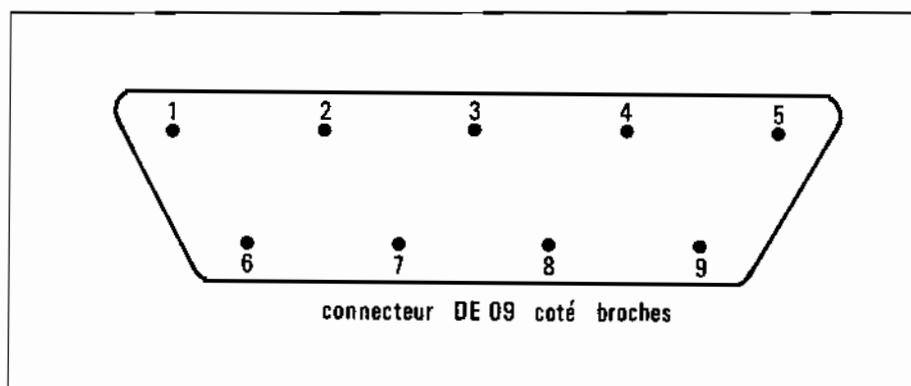


- | | |
|---|---|
| 1 : Masse mécanique (101) (terre). | 13 : Deuxième prêt à émettre (SCTS, Second CTS). |
| 2 : Emission de donnée (103) (TD, Transmit Data). Ligne où sont transmises les données séries. | 14 : Deuxième émission de données (STD). |
| 3 : Réception de données (104) (RD, Receive Data). Ligne de réception des données séries. | 15 : Horloge d'émission (114) (TC, Transmit Clock). Horloge de transmission pour synchroniser le récepteur avec l'émetteur lors d'une transmission synchrone. |
| 4 : Demande d'émission (105) (RTS, Request To Send). Ce signal permet d'informer le Modem qu'une donnée est prête à être émise. Le modem doit répondre par le signal CTS pour que la donnée soit émise. | 16 : Deuxième réception de données (SRD). |
| 5 : Prêt à émettre (106) (CTS, Clear To Send). Lorsque l'émetteur reçoit CTS, il émet une donnée sur TD. | 17 : Horloge de réception (115) (RC, Receive Clock). Horloge de synchronisation de l'émetteur et du récepteur lors de la réception de données synchrone. |
| 6 : Modem prêt (107) (DSR, Data Set Ready). Ce signal permet au modem de signaler à l'émetteur qu'il est en ligne. | 18 : Non connectée. |
| 7 : Masse Electrique (102). | 19 : Deuxième demande d'émission (SRTS). |
| 8 : Détection de porteuse (109) (DCD, Data Carrier Detect). Le modem indique au système informatique qu'une porteuse a été détectée. | 20 : Terminal prêt (DTR, Data Terminal Ready). Ce signal est actif lorsque le système est en ligne. Il sert de commande de connexion du modem. |
| 9 et 10 : non connectés. | 21 : Qualité du signal (SQ, Signal Quality). |
| 11 : Sélection de fréquence d'émission (126). Rarement utilisée. | 22 : Indicateur d'appel (125) (RI, Ring Indicator). Ce signal permet au modem de signaler au système informatique qu'une communication est demandée. |
| 12 : Deuxième détection de porteuse (SDCD, Second DCD). Deuxième ligne DCD pour la transmission en full duplex. | 23 : Non connectée. |
| | 24 : Horloge d'émission externe (113) (ETC, External Transmit Clock). |

Les nombres portés entre parenthèses (101, 102,...) correspondent à une notation des signaux selon la norme V24 émise par le CCITT (*Comité consultatif international pour le télégraphe et le téléphone*) - norme européenne de la liaison RS-232).

Le connecteur simplifié

Tous les signaux énoncés ci-dessus ne se trouvent pas toujours câblés sur les interfaces RS-232. Il existe de plus un connecteur simplifié (DE 09), réunissant un nombre minimal de signaux pour piloter un modem, dont le brochage est le suivant :



- 1 : DCD
- 2 : RD
- 3 : TD
- 4 : DTR
- 5 : GND
- 6 : DSR
- 7 : RTS
- 8 : CTS
- 9 : RI

Les vitesses de transmission

La vitesse de transmission de la liaison RS-232 est en fait la vitesse de sérialisation des bits du mot à transmettre. Elle s'exprime en bauds ou encore dans le cas de la RS-232 en bits par secondes.

Cette vitesse peut varier de 0 à 20 kilo-bauds, mais les valeurs standard admises sont 50, 75, 150, 300, 600, 1 200, 2 400, 4 800, 9 600, ou 19 600 bauds.

8/5.2.2

Le code ASCII

Pour permettre la communication en différents organes informatiques, des codes concernant les caractères se sont multipliés. Le code ASCII (*American Standard Code for International Interchange*) est actuellement le plus répandu et admis par presque toutes les sociétés.

C'est un code à sept bits, le plus souvent utilisé avec un bit supplémentaire (bit de parité) permettant de détecter d'éventuelles erreurs.

position des bits 7, 6 et 5

		000	001	010	011	100	101	110	111	
		0	1	2	3	4	5	6	7	
position des bits 4, 3, 2 et 1	0000	0	NUL	DLE	SP	0	␣	P	`	p
	0001	1	SOH	DC1	!	1	A	Q	a	q
	0010	2	STX	DC2	"	2	B	R	b	r
	0011	3	ETX	DC3	#	3	C	S	c	s
	0100	4	EOT	DC4	\$	4	D	T	d	t
	0101	5	ENQ	NAK	%	5	E	U	e	u
	0110	6	ACK	SYN	&	6	F	V	f	v
	0111	7	BEL	ETB	'	7	G	W	g	w
	1000	8	BS	CAN	(	8	H	X	h	x
	1001	9	HT	EM	)	9	I	Y	i	y
	1010	A	LF	SUB	*	:	J	Z	j	z
	1011	B	VT	ESC	+	;	K	[	k	{
	1100	C	FF	FS	,	<	L		l	l
	1101	D	CR	GS	-	=	M	]	m	}
	1110	E	SO	RS	.	>	N	^	n	~
	1111	F	SI	US	/	?	O	—	o	DEL

LE CODE ASCII

NUL : Absence de caractère.

SOH : Début d'en-tête (*Start Of Heading*).

STX : Début du texte (*Start Of Text*).

ETX : Fin du texte (*End Of Text*).

EOT : Fin de transmission (*End Of Transmission*).

ENQ : Demande d'identification (*Enquiry*).

ACK : Accusé réception (*Acknowledge*).

BEL : Sonnette (*Bell*).

BS : Retour arrière d'un caractère (*Back Space*).

HT : Tabulation horizontale (*Horizontal Tabulation*).

LF : Passe à la ligne suivante (*Line Feed*).

VT : Tabulation verticale (*Vertical Tabulation*).

FF : Passe à la page suivante (*Form Feed*).

CR : Retour chariot (*Carriage Return*).

SO : Passage dans un jeu de caractères spéciaux (*Shift Out*).

SI : Retour en caractères standards (*Shift In*).

DLE : Change la signification du caractère suivant (*Data Link Escape*).

DC1, DC2, DC3, DC4 : Caractères de commande de signaux ou de périphériques (*Device Control 1, 2, 3, 4*).

NAK : Réponse négative à une demande (*Negative Acknowledge*).

SYN : Caractère de synchronisation (*Synchronus*).

ETB : Fin de transmission d'un bloc de données (*End of Transmission Block*).

CAN : Annulation du caractère précédent (*Cancel*).

EM : Fin physique du support utilisé (*End of Medium*).

SUB : Remplacement (*Substitute*).

ESC : Caractère de contrôle permettant un codage (*Escape*).

FS, GS, RS, US : caractères de séparation organisés hiérarchiquement.

SP : Espace (*Space*).

DEL : Suppression (*Delete*).

Certains de ces caractères peuvent avoir une signification spéciale selon le type de système informatique.

8/5.2.3

La RS-232 Amstrad

L'interface RS-232 Amstrad se présente sous la forme d'un boîtier se connectant sur la prise d'extension 2 × 25 contacts de votre Amstrad CPC 464, 664 ou 6128. Il est possible d'adjoindre l'extension lecteur de disquette DDI-1 sur un raccord prévu à cet effet. Vous pouvez, de plus, ajouter d'autres extensions, le BUS étant disponible. L'alimentation de l'interface est réalisée par un boîtier supplémentaire.

Vous disposez en sortie d'un connecteur DB-25 sur lequel sont disponibles les signaux apparaissant aux broches 2 (TD), 3 (RD), 4 (RTS), 5 (CTS), 7 (GND), 8 (DCD), 20 (DTR), 22 (RI) [voir Partie 8, chap. 5.2.1]. Les autres broches ne sont pas utilisables.

Dans l'interface se trouve un logiciel en ROM qui vous permet de programmer et d'utiliser la RS-232 sous le langage Basic grâce aux instructions RSX.

Ce logiciel est placé sous le numéro de ROM 6 dont vous pouvez lister toutes les commandes grâce à l'instruction nouvelle : IROMCAT,6¹⁾.

Si vous possédez une autre extension munie d'un logiciel en ROM, vous pouvez aussi accéder à la liste de ses commandes en tapant IROMCAT,n° de l'extension.

Si vous ne connaissez pas le numéro de votre extension, tapez tout simplement IROMCAT et vous obtiendrez le nom de toutes vos extensions connectées.

1) Les caractères @ (aérobis) et I dont les codes ASCII sont respectivement 64 et 124 sont obtenus sur Amstrad par la frappe des touches : SHIFT + à et SHIFT + ù. Sur l'imprimante Amstrad ces caractères deviennent respectivement : à et ù.

ROM	Ver#	Title
15	1.20	BASIC
14	1.20	BASIC
13	1.20	BASIC
12	1.20	BASIC
11	1.20	BASIC
10	1.20	BASIC
9	1.20	BASIC
8	1.20	BASIC
7	0.50	CPM ROM
6	1.01	SIO DRIVER
5	1.20	BASIC
4	1.20	BASIC
3	1.20	BASIC
2	1.20	BASIC
1	0.32	V ROM
0	1.20	BASIC

On s'aperçoit, sur cette copie d'écran, que la ROM CPM est la ROM n° 7, la ROM V-DOS (si vous la possédez) est la ROM n° 1...

Si vous désirez ne plus utiliser vos ROMs durant une session de travail, vous pouvez déconnecter toutes vos ROMs (HORS TENSION !), soit taper l'instruction IROMOFF. Attention, à cet instant vous perdez toutes les instructions précédemment installées en RSX — mais vous récupérez de la mémoire vive (RAM). Seule une réinitialisation de votre Amstrad réinstallera les ROMs.

I - Les fonctions de base

ISETSIO

Cette fonction permet d'initialiser l'interface RS-232, c'est-à-dire lui fournir tous les paramètres pour l'utiliser dans une configuration donnée.

La syntaxe est la suivante :

**ISETSIO, vitesse émission, vitesse réception, contrôle,
nombre de bits, parité, nombre de bits d'arrêt**

- Vitesse d'émission : c'est la vitesse (en bauds) à laquelle seront émises les données. Les valeurs peuvent être : 50, 75, 110, 150, 200, 300,

600, 1 200, 1 800, 2 000, 2 400, 3 600, 4 800, 9 600, 19 200. La valeur par défaut étant 9 600.

- **Vitesse de réception** : de même que la vitesse d'émission mais pour la réception des données.
- **Contrôle** : ce paramètre, lorsqu'il est non nul, permet de valider le contrôle du protocole d'échange par les lignes CTS, DTR et DCD. Si vous donnez à ce paramètre la valeur 0, le protocole d'échange matériel est dévalidé, ce qui peut être utile lorsque vous n'utilisez pas les lignes de contrôle de la RS-232.
- **Nombre de bits** : c'est le nombre de bits par caractère qu'il faut transmettre, ce nombre peut être compris entre 5 et 8.
- **Parité** : si vous désirez que la RS-232 génère un bit de parité à chaque mot émis, afin de vérifier la validité du caractère lors de la réception, il faudra fixer ce paramètre à 2 pour une parité impaire, à 1 pour une parité paire. La valeur 0 impose à l'interface de ne pas générer de bit de parité, ni de vérifier la validité du caractère. La parité est liée au nombre de bits mis à 1 dans un même octet. Si ce nombre est toujours pair, on dira que la parité est toujours paire (even en anglais) une parité impaire consiste en un nombre impair de bits à 1 (odd en anglais), aucune parité particulière (none en anglais) correspondra à des nombres indifféremment pairs ou impairs.

Lorsqu'on déclare un échange en parité paire par exemple, le système comptera les bits à 1 de chaque octet et mettra le huitième à 1 ou à zéro de façon à ce que le nombre total soit toujours pair. Le récepteur recevra alors des octets à parité paire uniquement. La perte d'un seul bit dans la transmission rendra l'octet impair et donc générera une erreur à la réception. Selon le protocole, soit cet octet sera ignoré ou le récepteur demandera sa réémission.

- **Nombre de bits d'arrêt** : vous pouvez spécifier ici le nombre de bits d'arrêt à la fin du caractère transmis :

0 = un bit d'arrêt

1 = un bit et demi

2 = deux bits d'arrêt

Exemple :

Il est possible d'échanger des données avec un minitel, par l'intermédiaire de sa prise péri-informatique et d'une RS-232 (voir Partie 8, chap. 5.2.4.2).

Voici l'instruction d'initialisation de l'interface RS-232 afin de coordonner les vitesses de transfert et d'adapter les formats des caractères :

ISETSIO, 1200, 1200, 0, 7, 1, 0 — vitesses d'émission et de réception de 1 200 bauds
— pas de gestion matérielle du protocole

- 7 bits par caractère
- parité paire
- 1 bit de stop

ISETTIMEOUT

Cette fonction vous permet de modifier le temps après lequel une commande sera abandonnée en cas de non-réussite.

Syntaxe : **ISETTIMEOUT,temps**

La valeur du temps sera donnée en millisecondes.

Par exemple l'instruction **ISETTIMEOUT,20000** imposera un temps d'attente de 20 secondes avant d'abandonner une tentative de communication.

ISIO

Cette commande vous permet de connaître l'état actuel de l'interface RS-232.

Syntaxe : **ISIO,@a%**

Il vous faudra tout d'abord définir la variable a% (par l'instruction a% = 0 par exemple).

Après exécution, la variable a% contiendra un entier codé sur 16 bits dont chacun aura la signification suivante :

- | | |
|--------|---|
| bit 15 | toujours égal à 0. |
| bit 14 | indique une erreur de synchronisation lorsqu'il est égal à 1, sinon 0. |
| bit 13 | indique qu'il y a eu surcharge lors de la réception donc des caractères perdus lorsqu'il est à 1. |
| bit 12 | indique une erreur de parité sur le caractère reçu lorsqu'il est à 1. |
| bit 11 | toujours égal à 0. |
| bit 10 | toujours égal à 0. |
| bit 9 | toujours égal à 0. |
| bit 8 | indique, lorsqu'il est à 1, que le dernier caractère est envoyé. |
| bit 7 | indique, par un 1, qu'une coupure a été reçue. |
| bit 6 | toujours égal à 0. |
| bit 5 | état de la ligne CTS (0 si prêt à émettre). |
| bit 4 | état de la ligne RI (1 si détection de sonnerie). |
| bit 3 | état de la ligne DCD. |
| bit 2 | indique, lorsqu'il est à 1, que le registre d'émission est vide. |
| bit 1 | toujours égal à 0. |
| bit 0 | indique lorsqu'il est à 1 qu'une donnée reçue est disponible. |

Exemple :

attente de la réception d'une donnée

```

10  a% = 0
20  WHILE (a% AND 1) < > 1
30  ùSIO,@a%
30  WEND
40  REM une donnée est reçue lorsque l'on
50  REM sort de la boucle
60  REM il faut maintenant la lire

```

IINCHAR

Cette instruction permet de lire un caractère reçu par la RS-232.

La syntaxe est la suivante : **IINCHAR,@r%,a**

La variable r% devra préalablement avoir été définie par l'instruction r% = 0, par exemple, car c'est son adresse qui est utilisée par le sous-programme de traitement de l'instruction ; de même que pour la variable a.

Après exécution, la variable r% contiendra :

- 0 : si un caractère était présent.
- 256 : si aucun caractère n'a été spécifié.
- 512 : s'il y a eu arrêt avant qu'un caractère ne soit lu.
- 768 : si une coupure en ligne a été détectée.
- 1024 : si un caractère a été lu, mais une erreur de synchronisation s'est produite (le caractère n'est pas valable).
- 1280 : si un caractère a été lu mais il y a eu saturation de caractères à la réception (on a perdu des caractères).
- 1536 : si un caractère a été lu, mais avec une erreur sur la parité.

Afin de récupérer le caractère reçu, on pourra lire sa valeur ASCII dans une variable (x par exemple), en exécutant la ligne de programme suivante :

x = peek(a)

Si cette valeur est un caractère affichable, l'instruction x\$ = str\$(x) vous la rendra.

IOUTCHAR

Cette instruction permet d'envoyer un caractère à la RS232. Celle-ci attend que son buffer d'émission soit vide avant de le transmettre.

Syntaxe : **IOUTCHAR,@r%,caractère(ou variable)**

Après exécution, r% contient :

- 0 : si le caractère a bien été envoyé.
- 256 : si aucun caractère n'a été spécifié.
- 512 : s'il y a eu arrêt avant que le caractère n'ait pu être envoyé.

II - Les fonctions de transferts de chaînes de caractères

ISETBLOCKEND

Il est possible de spécifier le caractère indiquant la fin de la chaîne à transmettre en exécutant l'instruction :

ISETBLOCKEND,caractère

IINBLOCK

Cette instruction permet de recevoir une chaîne de caractère qui s'achèvera par un arrêt sur la ligne, le caractère de fin de chaîne spécifié précédemment ou lorsque la variable chaîne a été remplie (255 caractères maximum).

Syntaxe : **IINBLOCK,@r%,a\$**

Après exécution r% contient :

- 0 : si la chaîne a été remplie.
- n : si n caractères ont été reçus en plus de la fin de chaîne.
- 256 : si aucune chaîne n'a été reçue.
- 512 + n : si le caractère n + 1 n'a pas été reçu et que l'on attendait plus de n caractères.
- 768 + n : si une coupure de ligne a été reçue au n° caractère.
- 1024 + n : si une erreur de synchronisation a été reçue avec une erreur de synchronisation.
- 1280 + n : si le n° caractère a été lu avec saturation du buffer de réception.
- 1536 + n : si le n° caractère est entaché d'une erreur de parité.

La chaîne a\$ contiendra la chaîne de caractères reçue.

IOUTBLOCK

Cette instruction permet d'envoyer une chaîne de caractères via la RS-232.

Syntaxe : **IOUTBLOCK,@r%,a\$(ou "chaîne de caractère")**

Après exécution r% contient :

- 0 : la chaîne a bien été envoyée.
- 256 : aucune chaîne n'a été spécifiée ou la chaîne était vide.
- 256 + n : arrêt avant la fin d'émission de la chaîne, il reste les n caractères de la chaîne à envoyer.

III - Transferts de fichiers

Deux types d'instruction permettent le transfert de fichiers par la RS-232.

Si vous utilisez une liaison minimale (vous ne connectez que les liaisons RX, TX et la Masse), alors vous utiliserez les instructions ISETFILEEND ; IINFILE ; IOUTFILE.

Si vous utilisez d'une liaison complète en connectant RTS, CTS, DTR, DSR et DCD en plus de RX, TX et la Masse, vous pouvez alors travailler avec des instructions plus « intelligentes » qui contrôlent le déroulement matériel du transfert (IBLOW ; et ISUCK).

ISETFILEEND

Cette instruction spécifie le caractère de fin du fichier.

Le caractère par défaut est CONTROLE Z (appui simultané sur la touche CTRL et sur la touche Z).

Syntaxe : ISETFILEEND,caractère

IINFILE

Cette instruction permet de recevoir un fichier ASCII et l'enregistre sur le support par défaut (cassette ou disquette).

Syntaxe : IINFILE,nom du fichier

Le transfert s'achève lorsque le caractère de fin de fichier est lu. Les éventuels messages d'erreurs sont affichés à l'écran.

IOUTFILE

Cette instruction envoie un fichier stocké sur disquette ou cassette via la RS-232.

Syntaxe : IOUTFILE,nom du fichier

Les messages d'erreurs sont affichés à l'écran

IBLOW

Cette instruction a la même fonction que l'instruction IOUTFILE mais permet un transfert contrôlé.

Syntaxe : IBLOW,nom du fichier

Un seul message d'erreur peut être envoyé à l'écran : TRANSFERT ABORTED (= transfert interrompu), si une erreur fatale se produit.

ISUCK

Cette instruction fonctionne de la même façon que IINFILE mais contrôle le déroulement matériel.

Syntaxe : ISUCK,nom du fichier

IV - Instruction spécifique de ligne

IBREAKSEND

Cette instruction permet d'envoyer un « Break » sur la ligne durant un temps déterminé. Le break n'est envoyé que lorsque le buffer d'émission est vide (break = arrêt, interruption).

Syntaxe : IBREAKSEND,@r%,temps de coupure

La variable r% contiendra, suite à l'exécution

- 0 si la coupure a bien été envoyée.
- 512 si le break n'a pas pu être envoyé, la ligne d'émission n'ayant pu être libérée.

Le temps de coupure est exprimé en millisecondes et ne pourra être compris qu'entre 0 et 65535.

V - Fin de transmission

ICLOSESIO

Cette commande provoque la fermeture de la liaison RS232 et dévalide les signaux de contrôle de déroulement matériel RTS et DTR. Elle n'est exécutée que lorsque le buffer d'émission est vide.

Syntaxe : ICLOSESIO,@r%

La variable r% contient après exécution :

- 0 si la RS232 est bien déconnectée.
- 512 si tout n'a pas été envoyé durant le temps d'attente spécifié.

VI - Utilisation d'une imprimante série

La plupart des imprimantes sont des imprimantes « parallèles » ou « centronics », communiquant avec l'ordinateur selon la norme CENTRONICS (voir Partie 8, chap. 3.2). Mais d'autres types de transmission peuvent être utilisés pour les imprimantes ou les tables traçantes, en particulier la transmission RS232.

Quatre commandes internes à la ROM de la RS-232 Amstrad permettent la gestion de dialogue avec une imprimante série.

ISERIAL

Cette commande frappée, toute donnée envoyée au port d'imprimante Centronics est détournée vers l'interface RS-232C. Celle-ci sera donc considérée comme étant le canal #8.

Par défaut, la RS-232C utilise le protocole XON/XOFF ainsi que le contrôle de déroulement matériel pour l'échange des données.

INOXON

Cette commande annule le contrôle de déroulement logiciel XON/XOFF. Tous les caractères reçus seront donc ignorés par la RS232C (Voir Partie 8, chap. 5.2).

IXON

Cette instruction rétablit le protocole XON/XOFF précédemment annulé.

IPARALLEL

Afin de ramener le canal #8 sur le port imprimante Centronics, il suffira de frapper PARALLEL.

VII - Emulation de terminal

Votre Amstrad peut se comporter comme un terminal informatique, si vous le connectez par exemple à un ordinateur « maître ».

ITERMINAL

Ordonne à l'Amstrad de se comporter comme un terminal informatique. Le clavier est directement connecté sur la ligne d'émission de données et l'écran sur la ligne de réception de données. C'est l'ordinateur maître qui devra renvoyer le caractère frappé pour affichage, car, comme le minitel en mode connecté, il n'y a pas d'écho local.

Pour reprendre la main il suffira de frapper sur <ESC> ou <CONTROL> <ESC>.

IHALFDUPLEX

Permet de connecter localement le clavier à l'écran si l'ordinateur maître ne renvoie pas les caractères frappés.

IFULLDUPLEX

Ramène à l'état antérieur, sans écho local.

ICTRLDISPLAY

Cette commande permet d'afficher les codes de fonction reçus au lieu d'y obéir.

ICTRLACTION

Ramène le terminal en état d'exécuter les codes de fonction reçus.

VIII - Le Vidéotex anglo-saxon

Si votre facture téléphonique ne vous effraie pas, vous pouvez vous connecter sur le réseau télématique anglais.

Les commandes, toutes spécifiques aux serveurs anglais, ont la syntaxe suivante :

IPRESTEL, (caractere)

ISAVEPRESTEL, nom du fichier

ILOADPRESTEL, nom du fichier

ICURSOR, no de colonne, no de ligne

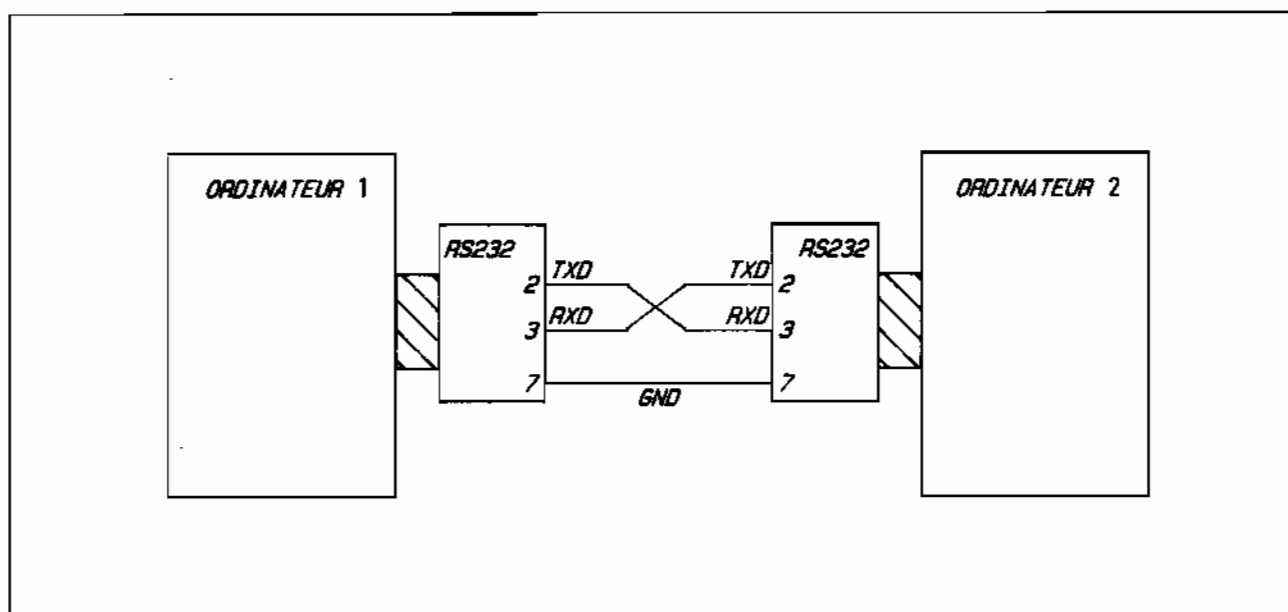
IREFRESH, no de ligne

8/5.2.4

Connexions et programmes

8/5.2.4.1

Connexion réduite



On remarquera sur la figure ci-dessus qu'il est nécessaire de croiser les liaisons RX et TX pour assurer la transmission correctement. En effet, la ligne d'émission d'un des organes est la ligne de réception de l'autre, et vice versa.

Discussion par clavier interposé : si vous possédez deux Amstrad, ou avez un ami proche de chez vous, vous pouvez, grâce à l'interface RS-232C, discuter avec lui par l'intermédiaire de vos claviers et écrans sur une distance pouvant aller jusqu'à cinquante mètres sans trop de problème, à condition d'utiliser du câble blindé. Si vous n'avez que quelques mètres à faire parcourir à votre discussion, un câble classique suffira.

Nous vous présentons tout d'abord l'algorithme du programme afin de se familiariser avec la programmation structurée.

- **Algorithme du programme**

Début

Initialiser l'interface RS-232

CO

Programmer les vitesses d'émission
et réception, la taille des caractères
et le format (bits de stop)

FIN CO

Effacer l'écran

Créer une fenêtre de réception

Créer une fenêtre d'émission

TANT QUE l'on désire communiquer

Prendre un caractère clavier à la volée

SI un caractère a été frappé

ALORS

Afficher le caractère dans la fenêtre émission

TANT QUE la RS-232 n'est pas libre

Attendre

FIN TANT QUE

Emettre le caractère

FIN SI

SI un caractère a été reçu par la RS-232

ALORS

lire le caractère

Afficher le caractère dans la fenêtre de réception

FIN SI

FIN TANT QUE

FIN

On remarquera les termes de la programmation structurée notés en majuscules soulignées, et les verbes d'action en minuscules soulignées, ainsi que l'indentation (décalage) lorsque l'on entre dans une structure de traitement suite à un terme d'algorithmique (SI ... ALORS ... FIN SI ; TANT QUE ... FIN TANT QUE).

La structure CO ... FIN CO est une structure dite de commentaire, c'est-à-dire qu'elle permet d'ajouter une explication sur un traitement. Elle ne modifie en rien le déroulement de l'algorithme et est optionnelle.

En algorithmique, tout programme possède un début et une fin, ce qui nous impose de ne pas introduire une structure qui bouclerait en permanence. Le remède peut être l'introduction d'une structure itérative du type :

TANT QUE l'ordinateur est connecté

...

FIN TANT QUE

Nous avons préféré la structure :

TANT QUE l'on désire communiquer

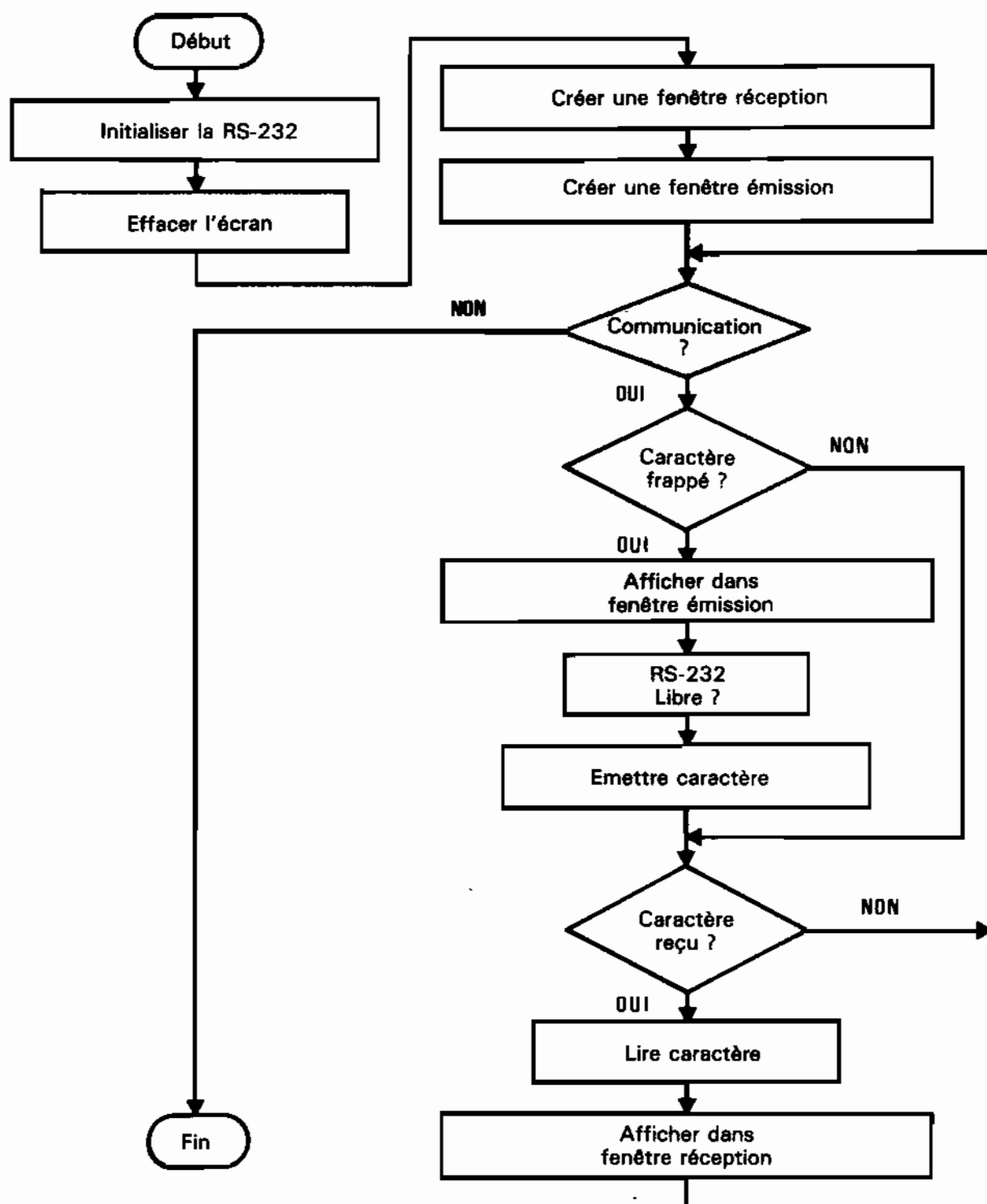
...

FIN TANT QUE

que le basic Amstrad résout plus ou moins rigoureusement par la ligne de commande : ON BREAK GOSUB...

- **Algorithme**

C'est la représentation symbolisée selon une norme de l'algorithme.



• Programme

Chacun des deux interlocuteurs lancera le programme. Deux fenêtres seront alors créées :

- la fenêtre EMISSION qui affichera les caractères frappés au clavier. Ces caractères seront transmis immédiatement à l'interface RS-232C ;
- la fenêtre RECEPTION qui affichera les caractères reçus par la RS-232, donc émis par le deuxième interlocuteur.

```

10 REM *** COMMUNICATION SERIE ***
20 REM *** ENTRE DEUX AMSTRADs ***
30 REM *****
40 REM *** INITIALISATION RS-232 ***
50 ùSETSID,1200,1200,0,7,1,0:REM *
60 REM *****
70 REM *** INITIALISATION ECRAN ****
80 MODE 2:REM *
90 WINDOW #4,1,80,1,12:REM *
100 INK 3,3:INK 2,26:REM *
110 PAPER #4,3:PEN #4,2:CLS #4:REM *
120 PRINT #4,"EMISSION":REM *
130 WINDOW #5,1,80,13,24:REM *
140 PAPER #5,2:PEN #5,3:CLS #5:REM *
150 PRINT #5,"RECEPTION":REM *
160 REM *****
170 s%=0:r%=0:r%=0
180 REM *** CARACTERE A EMETTRE ***
190 a$=INKEY$:REM *
200 IF a$="" THEN 290:REM *
210 PRINT #4,a$;:REM *
220 REM *****
230 REM *** EMISSION CARACTERE *****
***
240 ùSID,àR%:REM
*
250 IF (r% AND 4) <> 4 THEN 240:REM
*
260 ùOUTCHAR,às%,ASC(a$):REM
*
270 REM *****
***
280 REM *** RECEPTION CARACTERE ? *****
***
290 ùSID,àr%:REM
*
300 IF (r% AND 1) <> 1 THEN 190:REM
*
```

```
310 REM *****
***
320 REM *** RECEPTION/AFFICHAGE *****
***
330 @INCHAR,@s%,@r:REM
*
340 PRINT #5,CHR$(PEEK(r));:REM
*
350 GOTO 190:REM
*
360 REM *****
***
370 END
```

- Ligne 50 : Un compromis entre la vitesse et la qualité de la communication est ici choisi. Nous aurions pu utiliser une vitesse de communication supérieure (19 200 bauds), mais la qualité des signaux circulant sur les liaisons en aurait souffert. De plus si le milieu dans lequel vous expérimenterez le programme est générateur de parasites, ou si vous utilisez une liaison assez longue, vous pourrez modifier les valeurs des vitesses de transmission de façon à obtenir une qualité satisfaisante.
- Lignes 190-200 : On teste si un caractère a été frappé au clavier, mais on ne l'attend pas, ceci afin de ne pas perdre de caractère reçu par la RS232C.
- Ligne 240 : On relève ici l'état de la liaison.
- Ligne 250 : On teste si la ligne d'émission TX est libérée pour émettre le caractère en Ligne 260.
- Lignes 290-300 : Test de l'arrivée d'un caractère dans le buffer de réception de la RS232. Ce test est fait « à la volée » afin de pouvoir émettre le caractère suivant frappé au clavier.

8/5.2.4.2

Connexion à un minitel

La connexion d'une interface RS-232 à un minitel peut s'effectuer relativement facilement moyennant quelques points de soudure et l'achat de trois composants.

Le format des signaux délivré par la fiche péri-informatique du minitel (fiche DIN 5 broches à l'arrière [voir Partie 10, chap. 1.1]) est le suivant pour un caractère (Fig. 1).

On remarquera qu'il est simulable par programmation de la RS-232 (1 200 bauds [réception et émission], 7 bits, 1 bit de parité paire, 1 bit de stop).

Le problème est posé par les niveaux logiques ; en effet un niveau 0V du minitel devra correspondre à un +12V pour la RS-232 et un 5V (niveau logique « 1 ») à un niveau -12V.

Cette adaptation sera réalisée par deux composants : les MC1488 et MC1489 classiques dans ce type de montage. Ces composants modifient la tension du signal à leur entrée, tout en les inversant. Une résistance dite de Pull-up sera connectée au +5V et à la sortie TX du minitel (cette sortie étant de type collecteur-ouvert).

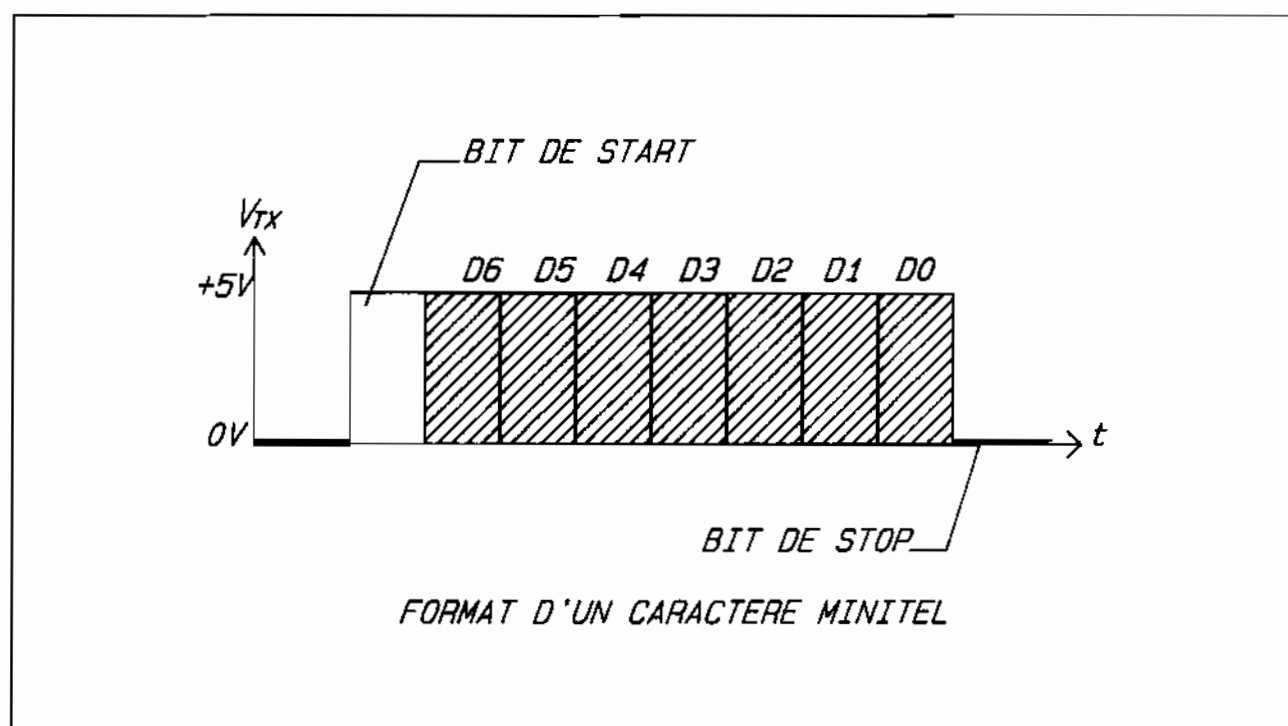
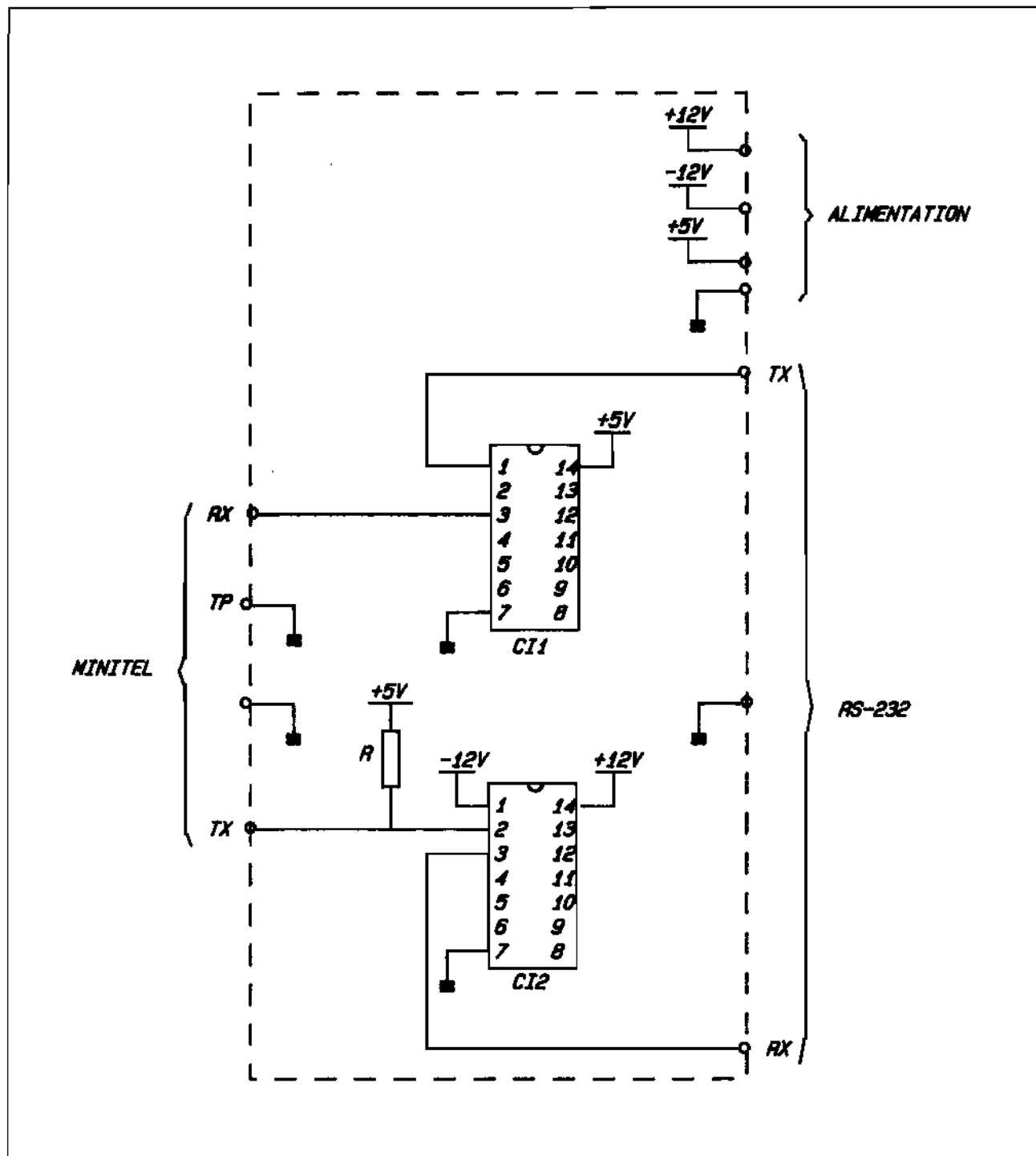


Fig. 1

Schéma structurel



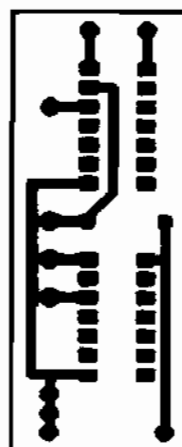


Fig. 2 : Câblage imprimé de la carte d'adaptation vue côté cuivre.

7^e Complément

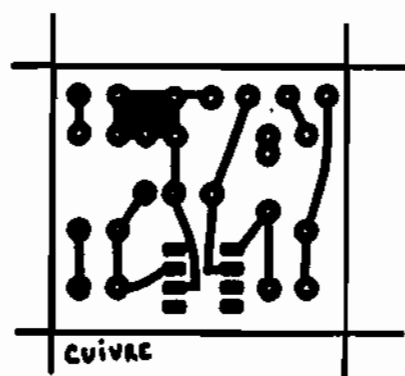


Fig. 3 : Circuit imprimé, côté soudure.

7^e Complément

I - Montage du boîtier

Nomenclature : C11 : MC 1489

C12 : MC1488

R : 10k Ω 1/4W

La réalisation pourra se faire par un câblage imprimé ou sur une plaque de type « Board à souder » vue le faible nombre de composants (Fig. 2).

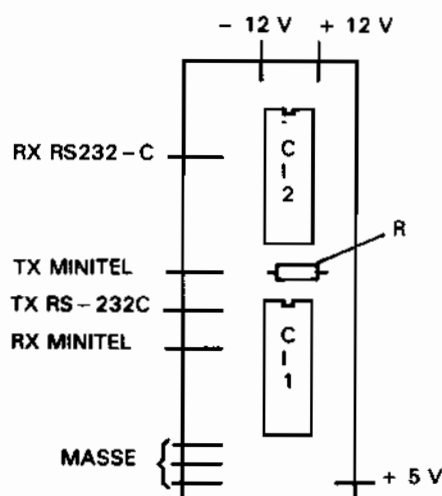


Fig. 3

A. DESCRIPTION DU MONTAGE

Vous pourrez connecter les fils d'alimentation, sur une alimentation spécialement conçue pour l'adaptation (délivrant les trois tensions +5V, +12V, -12V, référencées par rapport à la masse commune 0V – **ATTENTION** : Le +5V ne devra pas être commun avec le +5V de votre micro-ordinateur sous peine de surprises désagréables quant à la durée de vie de ce dernier).

Le plus économique et le plus rapide est d'intégrer le montage dans le boîtier de l'interface RS-232 et de monter une prise DIN femelle type châssis sur le boîtier.

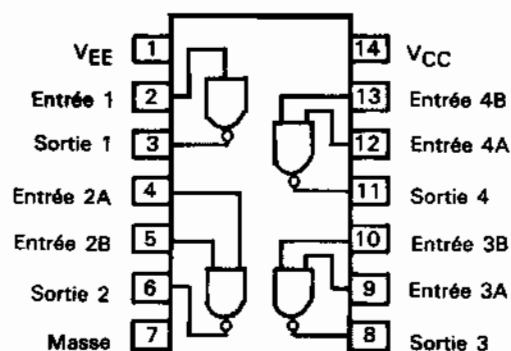
Un bon fer à souder et une technique de soudage correcte sera utile (attention à ne pas trop chauffer).

Tout d'abord, vous dévisserez le boîtier de la RS232, qui doit s'enlever sans problème. Deux autres vis maintiennent la carte de circuit imprimé ainsi que le connecteur Cannon DB-25. Une fois soulevés, vous aurez largement la place pour installer votre montage.

Caractéristiques des composants

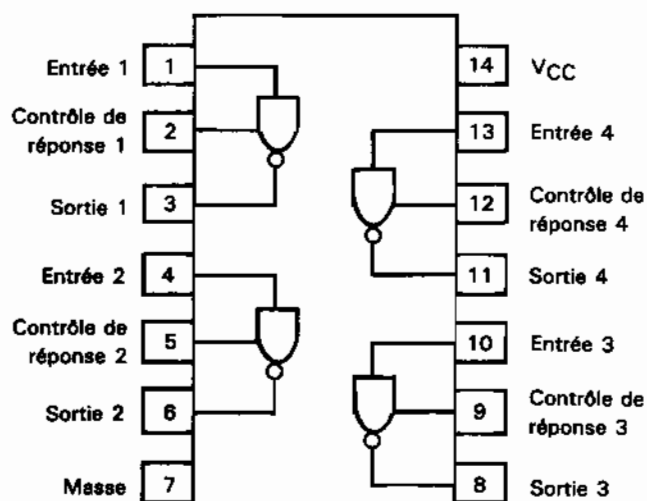
Configuration des broches

MC 1488



VUE DE DESSUS

MC 1489



VUE DE DESSUS

Vous remarquerez sur la carte imprimée deux composants intégrés MC1488 et MC1489. N'hésitez pas à souder — avec précaution tout de même ! — les fils d'alimentation sur les pattes vues côté cuivre (prenez soin de bien repérer les broches) dont la masse sur la broche 7 du MC1488, le -12V sur la 1 et le +12V sur la 14, une seule soudure sur la broche 14 du MC1489 pour connecter le +5V.

Le fil TX côté RS-232 sera soudé sur la broche 11 du MC1488 et le fil RX sur la broche 4 du MC1489.

Vous pourrez percer à l'arrière la partie inférieure du boîtier au diamètre de votre fiche DIN femelle et vous soudez le fil TX côté minitel sur la broche repérée 1 de la fiche DIN, le fil RX sur la broche 2 et la masse sur la broche centrale repérée 3 (Fig. 4).

Avant de remonter le tout, découpez une feuille de papier à la dimension du circuit imprimé de l'interface, que vous placerez entre celui-ci et le vôtre pour éviter tout court-circuit. Vous pouvez maintenant refixer le circuit imprimé et le boîtier en veillant à ne pas pincer les fils de liaisons.

Un câble volant vous sera nécessaire pour connecter votre minitel sur l'interface. On préférera le fabriquer soi-même à l'aide de trois fils (un câble blindé serait encore mieux, mais pas expressément recommandé sur une liaison relativement courte) que vous connecterez broche à broche sur les n°s 1, 2 et 3 de deux fiches DIN mâles. Méfiez-vous des câbles du commerce, certains possèdent des croisements dans les liaisons qui mettraient en danger votre montage.

De la patience, de la méthode et un travail méticuleux vous permettront de réaliser un montage fiable qui va vous permettre d'utiliser les ressources du minitel dans les programmes suivants.

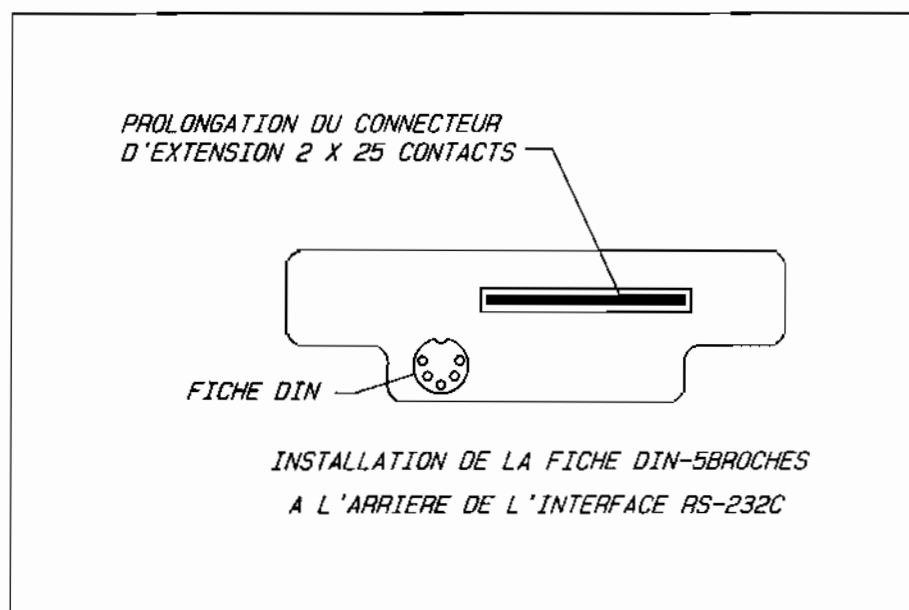


Fig. 4

B. ESSAI PRÉALABLE

Vous pourrez essayer votre montage à l'aide du programme précédent. Il vous suffira de modifier la ligne 50 par la ligne suivante :

50 SETSIO,1200,1200,0,7,1,0

Le minitel ne possédant pas de possibilité de fenêtrage, vous verrez sur celui-ci s'afficher les caractères émis à côté des caractères reçus, mais vous aurez une idée, si le cas se produit, de la partie de votre montage, qui ne fonctionne pas.

C. MAINTENANCE SUPERFICIELLE

- premièrement vérifiez votre programme ;
- deuxièmement vérifiez les alimentations des composants (si un composant est brûlant, il y a un court-circuit, il faudra très certainement le changer) ;
- si aucun caractère frappé sur l'ordinateur ne s'affiche sur le minitel, vérifiez la partie fléchée de droite à gauche sur le schéma structurel.
- si aucun caractère frappé sur le minitel ne s'affiche sur l'écran de votre Amstrad, vérifiez la partie fléchée de gauche à droite sur le schéma structurel.

Lorsque tout fonctionnera correctement, vous pourrez passer aux programmes suivants.

II. Logiciels pratiques d'utilisation minitel

A. SAUVEGARDE DE PAGES MINITEL

Lorsque vous explorez l'univers attirant du réseau télématique français, et, qui plus est, le passionnant kiosque (TELETEL3 ou encore 3615), il faut vous attendre à une facture téléphonique élevée si vous passez un temps non négligeable à lire les écrans proposés. Vous utilisez probablement des boîtes aux lettres télématiques, ou consultez les horaires ferroviaires, ou encore vos relevés bancaires... Voilà un moyen d'alléger vos factures :

Un programme de sauvegarde de pages écrans sur vos disquettes, ou cassettes, vous permettra de rester juste le temps nécessaire à l'affichage de la page.

Le programme présenté enregistre les pages en mémoire centrale dans un tableau pré-défini de 3 000 caractères (dimensionnables dans la limite de votre mémoire disponible).

Cette taille doit cependant être la taille minimale pour une page, car certains serveurs vous proposent des pages animées très avides de caractères.

• Algorithme du programmeDEBUTInitialiser la taille du fichier temporaireInitialiser la RS-232Spécifier un temps maximal d'attente de caractèreInitialiser les variables de communicationAfficher le menuAttendre un nom de fichierTANT QUE l'utilisateur ne désire pas enregistrerAttendreFIN TANT QUETANT QUE l'on désire enregistrerTANT QUE un caractère n'est pas reçuAttendre ce caractère le temps spécifiéFIN TANT QUESI aucun caractère n'a été reçuALORS ALLER A la procédure de sauvegarde du fichierFIN SISauvegarder temporairement le caractèrePointer la variable suivanteSORTIR SI le fichier temporaire est pleinFIN TANT QUEOuvrir le fichier de sauvegardeCO

Procédure de sauvegarde du fichier

FIN COPOUR I ALLANT DU début du fichier A la finSauvegarder séquentiellement les valeurs
contenues dans le fichierFIN POURFermer le fichier de sauvegardeFIN

• Programme

Après avoir entré et sauvegardé ce programme, vous pourrez l'exécuter ; toutes les instructions sont affichées à l'écran. Nous vous conseillons d'effectuer un essai préalable en frappant des caractères à enregistrer sur le clavier du minitel. Si vous n'avez pas fait d'erreur de frappe et que votre interface RS232 ainsi que le montage précédent fonctionne correctement, vous pourrez relire la page enregistrée avec le programme proposé ensuite.

```

10 REM *** PROGRAMME DE SAUVEGARDE ***
20 REM *** DE PAGES MINITEL ***
30 REM *****
40 REM *** INITIALISATION ***
50 DIM fich(3000):REM *
60 ùSETSID,1200,1200,0,7,1,0:REM *
70 ùSETTIMEOUT,10000:REM *
80 r%=0:a=0:REM *
90 REM *** MENU DE PRESENTA
TION ***
100 MODE 2:REM *
110 PRINT"SAUVEGARDE DE PAGES MINITEL":R
EM *
120 PRINT:PRINT:REM *
130 INPUT"Nom de la page a sauvegarder";
fich$:REM *
140 PRINT:PRINT:REM *
150 PRINT"Appuyer sur <ESPACE> pour comm
encer":REM *
160 a$=INKEY$:REM *
170 IF a$="" THEN 160:REM *
180 PRINT CHR$(11);CHR$(13);:REM *
190 PRINT"Enregistrement de la page en c
ours...":REM *
200 PRINT"Appuyez deux fois sur <ESC> po
ur interrompre":REM *
210 REM *****

```

```
*****
220 k=0
230 ON BREAK GOSUB 350
240 i=0
250 REM *** RECEPTION CARACTERES ***
260 &INCHAR, &r%, a: REM *
270 IF r%=512 THEN 400: REM *
280 fich(i)=PEEK(a): REM *
290 IF k=1 THEN 400: REM *
300 i=i+1: REM *
310 IF i>3000 THEN 380: REM *
320 GOTO 260: REM *
330 REM *****
340 REM *** INTERRUPTION ***
350 k=1: REM *
360 RETURN: REM *
370 REM *****
375 REM *** TROP DE PAGES ECRAN
***
380 PRINT CHR$(7); "FICHIER REMPLI !!": RE
M *
385 REM *****
***
390 REM *** SAUVEGARDE SUR DISQUETTE *
**
400 j=i: REM
*
410 PRINT: PRINT: REM
*
420 PRINT "Sauvegarde en cours...": REM
*
430 OPENOUT fich$: REM
*
440 FOR i=0 TO j: REM
*
450 WRITE#9, fich(i): REM
*
460 NEXT i: REM
*
470 CLOSEOUT: REM
*
480 PRINT CHR$(11); CHR$(13);: REM
*
490 PRINT "Sauvegarde effectuee ": REM
*
500 REM *****
**
510 END
```

— Ligne 70 : On spécifie une durée d'attente maximale de caractères de 10 000 millisecondes au cas où vous utiliseriez ce programme durant une période d'affluence sur le serveur, ce qui a pour fâcheuse incidence de ralentir considérablement la communication, voire la bloquer.

— Ligne 270 : Si un caractère n'est pas reçu dans le délai imparti, $r\% = 512$, on quitte la sauvegarde temporaire et on va à la sauvegarde sur support.

Lorsque la sauvegarde sur disquette (ou cassette) est en cours, vous pouvez sans problème déconnecter votre minitel de la ligne. Vous pouvez même effectuer cette manœuvre avant, ceci aura pour effet d'attendre 10 secondes avant de procéder à l'enregistrement.

Votre page minitel, une fois sauvegardée, il ne vous reste plus qu'à la lire.

B. LECTURE D'UNE PAGE MINITEL SAUVEGARDÉE

La lecture s'effectuera en mode non connecté, le fichier de sauvegarde sera envoyé séquentiellement sur le minitel par l'intermédiaire de la prise péri-informatique. La visualisation sera celle de la ou des page(s) que vous aurez enregistrées précédemment. Aucune action particulière ne sera nécessaire à partir du clavier du minitel.

- Algorithme du programme

DEBUT

Initialiser la RS-232C

Initialiser les variables de communication

Dimensionner la taille maximale du fichier

Afficher le menu

Attendre un nom de fichier

Ouvrir le fichier sur disquette en lecture

TANT QUE l'on n'a pas atteint la fin du fichier

Transférer les données au fur et à mesure
dans le tableau

FIN TANT QUE

Fermer le fichier

POUR I ALLANT DU début du tableau A la fin

Transmettre séquentiellement les valeurs du tableau

FIN POUR

FIN

• Programme

```

10 REM *** PROGRAMME DE LECTURE ***
20 REM *** DE PAGES ENREGISTREES ***
30 REM *** SUR UN MINTEL ***
40 REM *****
50 REM *** INITIALISATION ***
60 @SETSID,1200,1200,0,7,1,0:REM *
70 R%=0:REM *
80 DIM FICH(3000):REM *
90 REM *****
100 REM *** MENU ***
110 MODE 2:REM *
120 INPUT "NOM DU FICHIER A VISUALISER";F
    ICH$:REM *
130 REM *****
140 REM *** LECTURE FICHIER **
150 OPENIN FICH$:REM *
160 I=0:REM *
170 WHILE NOT EOF:REM *
180 INPUT#9,FICH(I):REM *
190 I=I+1:REM *
200 WEND:REM *
210 CLOSEIN:REM *
220 REM *****
230 REM *** ENVOI FICHIER ***
240 FOR J=0 TO I:REM *
250 @SID,@R%:REM *
260 IF (R% AND 4) <> 4 THEN 250:REM *
270 @OUTCHAR,@R%,FICH(J):REM *
280 NEXT J:REM *
290 REM *****
300 END

```

— Le fichier est lu entièrement en lignes 140 à 210 puis envoyé sur le minitel caractère par caractère en lignes 230 à 280. On vérifie en ligne 260 que le caractère précédent a bien été envoyé avant d'envoyer le suivant.

— S'il se produit des erreurs lors de la lecture des pages écran minitel, par exemple, s'il manque des caractères, ou si les graphismes ne sont pas conformes à ceux normalement enregistrés, alors modifiez les lignes 260 à 290 du programme SAUVEGARDE DE PAGES MINTEL, page 11, et remplacez-les par les lignes qui suivent.

260 INCHAR,@r%,a

270 fich(i) = PEEK(a)

280 i=i+1

290 GOTO 260

— Les lignes 300 à 320 seront effacées. Vous effacerez aussi les lignes 350 à 385.

— Cette modification vous évitera de « rater » des caractères, notamment pour le cas des pages écran très fournies en graphismes et animation, où un seul affichage peut demander l'envoi de plusieurs codes ASCII; mais l'inconvénient est qu'il faut surveiller le défilement de la page afin de pallier un éventuel blocage du serveur lors de l'enregistrement. On frappera alors deux fois sur la touche <ESC> pour sauvegarder le fichier enregistré.

C. COMMUNICATION DE DONNEES PAR LIGNE TELEPHONIQUE

Votre Amstrad connecté sur une ligne téléphonique, voilà l'occasion rêvée de faire profiter vos amis éloignés de vos fichiers précieusement sauvegardés sur disquette.

Les deux programmes qui suivent — l'un pour envoyer un fichier, l'autre pour le recevoir — vont vous permettre de vous échanger vos fichiers numériques par l'intermédiaire des télécommunications (Fig. 4).

Un petit problème subsiste : si les vitesses d'émission et de réception sont les mêmes sur les liaisons RX et TX de la fiche péri-informatique du minitel (1 200 bauds par défaut), il n'en est pas de même pour le modem qui transmet sur la ligne téléphonique. En effet, celui-ci émet à 75 bauds et reçoit à 1 200 bauds. On ne peut donc pas connecter deux minitels ensemble et dialoguer ainsi.

Si vous êtes un des nombreux possesseur de minitel à modem retournable (vous le saurez en lisant la fiche signalétique sous l'appareil : il est indiqué la mention Modem Retournable) vous pourrez pallier cet inconvénient par programmation du minitel au travers de la prise péri-informatique, donc de votre RS-232. On préférera faire émettre à 1 200 bauds le système envoyant le flot de caractères le plus important.

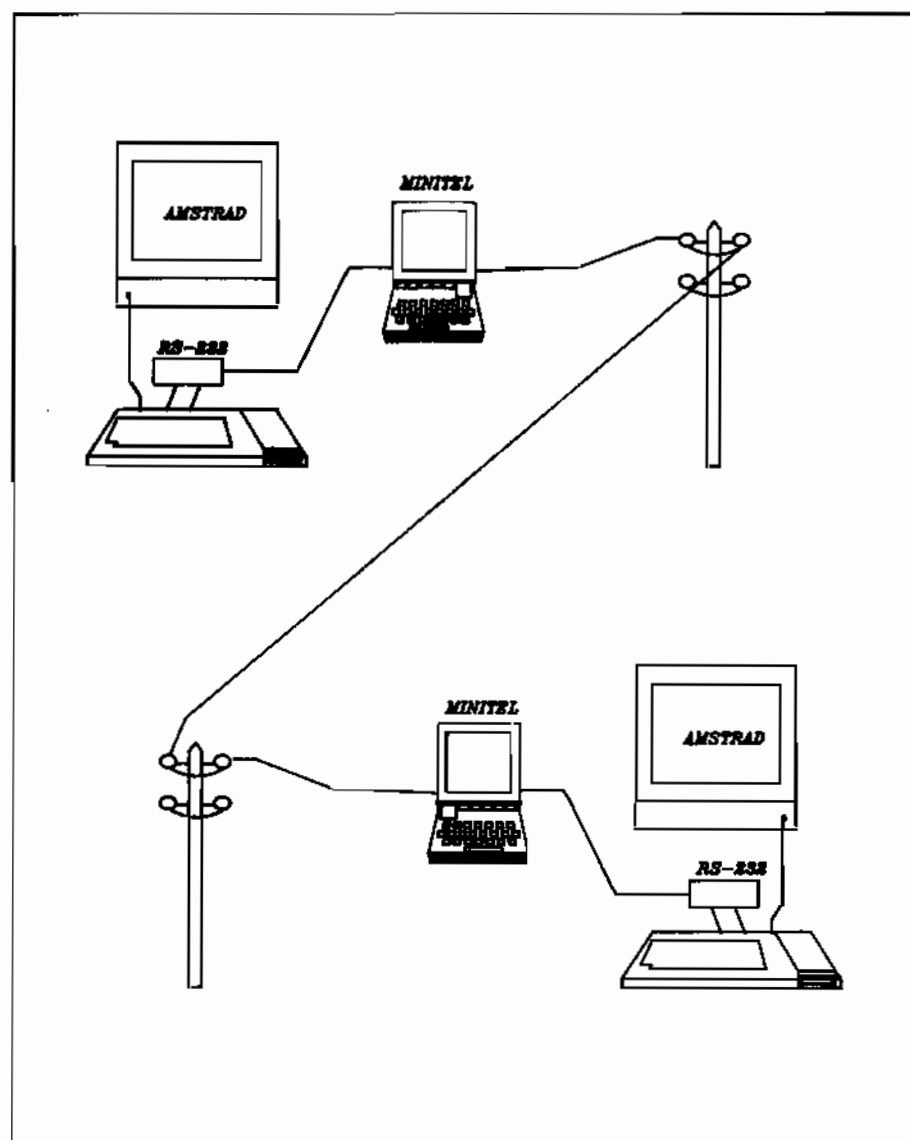


Fig. 4

- **Programme Emission**

On relira avec profit la Partie 10, chapitre 1.2.1 « Programmation du minitel ».

```
10 REM *** ECHANGE DE DONNEE NUMERIQUE *  
**  
20 REM *** ENTRE DEUX AMSTRADs *  
**  
30 REM *** PAR UNE LIGNE TELEPHONIQUE *  
**  
40 REM *****  
**  
50 REM *** INITIALISATION ***  
60 @SETSID,1200,1200,0,7,1,0  
70 @SETTIMEOUT,5000  
80 A=0:R%=0  
90 REM *** AFFECTATION CODES TRANSMISSIO  
N ***  
100 ENQ=&5:ETB=&17:EOT=&4:ACK=&6  
110 REM *****  
*****  
120 REM *** MENU ***  
130 MODE 2  
140 PRINT"ENVOI D'UN FICHIER DE DONNEES  
A UN AUTRE AMSTRAD"  
150 PRINT:PRINT  
160 PRINT"VOTRE FICHIER EST-IL BIEN UN F  
ICHER NUMERIQUE?"  
170 A$=INKEY$  
180 IF A$="" THEN 170  
190 IF A$="D" OR A$="o" THEN 210  
200 STOP  
210 PRINT:PRINT  
220 INPUT"NOM DU FICHIER A TRANSMETTRE";  
fich$  
230 PRINT:PRINT  
240 PRINT"APPUYER SUR UNE TOUCHE POUR EN  
VOYER LA PORTEUSE"  
250 A$=INKEY$  
260 IF A$="" THEN 250  
270 REM *** ENVOI DE LA PORTEUSE DU MINI  
TEL ***  
280 REM *** POUR BASCULEMENT EN 1200 BAU  
DS ***  
290 REM *** EMISSION ET 75 BAUDS RECEPTI  
ON ***  
300 REM *** PUIS CONNEXION A LA LIGNE  
***  
310 RESTORE 380  
320 FOR I=1 TO 6  
330 READ A  
340 @SID,@R%  
350 IF (R% AND 4) <> 4 THEN 340  
360 @OUTCHAR,@R%,A
```

```
370 NEXT I
380 DATA &1B,&39,&6F,&1B,&39,&6B
390 REM *****
*****
400 REM *** TEMPORISATION A LA CONNEXION
***
410 FOR I = 1 TO 1000
420 ̑CLOSESID
430 ̑SETSID,1200,1200,0,7,1,0
440 NEXT I
450 REM ***** FIN DE TEMPORISATION ***
****
460 REM *** DEMANDE DE PRESENCE AU DESTI
NATAIRE ***
470 ̑OUTCHAR,̑R%,ACK
480 REM *****
*****
490 REM *** ATTENTE REPONSE ***
500 ̑INCHAR,̑R%,A
510 IF R% = 512 THEN 1050
520 REM *****
530 REM *** PREPARATION DU FICHIER ***
540 OPENIN FICH$
550 REM *****
560 REM *** ENVOI FICHIER ***
570 PRINT:PRINT
580 PRINT"ENVOI DU FICHIER EN COURS:";
590 WHILE NOT EOF
600 INPUT#9,A
610 A$=STR$(A)
620 REM *** ENVOI DE LA LONGUEUR ***
630 REM ***          DU NOMBRE          ***
640 L=LEN(A$)+32
650 ̑SID,̑R%
660 IF (R% AND 4) <> 4 THEN 660
670 ̑OUTCHAR,̑R%,L
680 ̑INCHAR,̑R%,A
690 IF R%=512 THEN 1040
700 REM *** ENVOI DU PREMIER NOMBRE ***
710 REM ***          CHIFFRE PAR CHIFFRE          ***
720 FOR I=1 TO L
730 ̑SID,̑R%
740 IF (R% AND 4) <> 4 THEN 730
750 B$=MID$(A$,I,1)
760 A=ASC(B$)
770 REM *** ENVOI D'UN CHIFFRE ***
780 ̑OUTCHAR,̑R%,A
790 REM *** VISUALISATION DE LA TRANSMIS
SION D'UN CARACTERE SUR L'ECRAN ***
800 PRINT". ";
```

```
810 NEXT I
820  SIO, R%
830 IF (R% AND 4) <> 4 THEN 820
840 REM *** ENVOI DE LA FIN DU PREMIER B
LOCK ***
850  OUTCHAR, R%,EOT
860 REM ***. ATTENTE D'UN ACCUSE RECEPTIO
N ***
870  INCHAR, R%,A
880 IF R% = 512 THEN 1000
890 X=PEEK(A)
900 IF X <> ACK THEN 1010
910 WEND
920 REM *** FIN D'ENVOI DU FICHIER ***
930 REM *** ENVOI DU CARACTERE DE FIN DE
TRANSMISSION ***
940  SIO, R%
950 IF (R% AND 4) <> 4 THEN 940
960  OUTCHAR, R%,EOT
970 CLOSEIN
980 GOTO 1080
990 REM FIN DE TRANSMISSION *****
*****
1000 REM *** INCIDENT EN TRANSMISSION **
*
1010 PRINT:PRINT
1020 PRINT"PROBLEME DE TRANSMISSION !! "
1030 GOTO 1080
1040 REM *****
*
1050 REM *** DESTINATAIRE NON CONNECTE *
**
1060 PRINT"L'AMSTRAD RECEPTEUR NE REpond
PAS !!"
1070 REM *****
*
1080 REM *** DECONNEXION DU MINITEL ***
1090 REM *** RETOUR EN MODE NORMAL ***
1100 REM *****
1110 RESTORE 1180
1120 FOR I=1 TO 3
1130 READ A
1140  SIO, R%
1150 IF (R% AND 4) <> 4 THEN 1140
1160  OUTCHAR, R%,A
1170 NEXT I
1180 DATA &1B,&39,&67
1190 REM *****
1200 END
```

- Ligne 100 : On essaiera, dans les programmes de transmission de données, de définir les valeurs ASCII des caractères de contrôle par des variables.
- Ligne 160 : L'ordinateur s'assure que le fichier à émettre est bien un fichier numérique.
- Lignes 240-380 : Le modem est mis en opposabilité (envoi de la séquence &1B, &39, &6F) pour émettre à 1 200 bauds, puis il est connecté à la ligne (séquence &1B, &39, &68) pour émettre la porteuse qui permettra au minitel distant de se connecter.
- Lignes 400-440 : Cette temporisation laisse le temps au minitel distant de se connecter à la ligne. Une astuce est employée pour vider le buffer de réception d'éventuels caractères reçus durant ce temps (caractères d'acquittement du protocole du minitel) : on ferme et on ouvre la liaison RS-232 dans cette boucle.
- Lignes 460-520 : L'émetteur interroge le récepteur en envoyant ACK, puis attend un caractère en réponse. Si ce caractère n'arrive pas avant 5 secondes (cf ligne 70) alors R% = 512, on considère que le récepteur n'est pas présent (aller en ligne 1050).
- Ligne 590 : TANT QUE la fin du fichier n'est pas atteinte.
- Lignes 620-670 : On émet la longueur du nombre (transformé en chaîne de caractères car le minitel ne peut envoyer que des caractères ASCII). En fait on va émettre un caractère ASCII, mais qui doit être supérieur à &20 (= 32) car les caractères ASCII compris entre 0 et &1F sont des caractères de contrôle. On enverra donc la longueur plus 32 (la longueur maximale du nombre sera égale à &7F - &20 = 127 - 32 = 95 chiffres, ce qui est plus que vraisemblable). Le récepteur retranchera 32 à la valeur ASCII reçue pour reconstituer la longueur du nombre.
- Lignes 680,690 : Attente d'un accusé de réception avant d'envoyer le nombre.
- Lignes 700-810 : Le nombre est envoyé chiffre par chiffre, en n'oubliant pas le signe ou l'exposant puisque l'on émet la valeur ASCII de chacun des caractères de la chaîne composant le nombre. Un point est affiché à l'écran pour chaque caractère émis.
- Ligne 850 : Envoi du caractère EOT (End Of Transmission = fin de transmission) pour signaler au récepteur qu'un nombre a été envoyé complètement.
- Lignes 860-900 : Attente d'un acquittement de la part du récepteur. Si l'acquittement n'est pas reçu dans les 5 secondes, la communication est coupée.
- Ligne 910 : FIN TANT QUE
- Ligne 960 : Envoi d'un deuxième EOT signalant la fin de la transmission totale.

- Lignes 990-1070 : Traitement des problèmes de communication.
- Lignes 1080-1200 : Déconnexion du minitel.

• **Programme Réception**

```
10 REM *** RECEPTION D'UN FICHIER DE DON
NEES ***
20 REM ***      PAR UNE LIGNE TELEPHONIQUE
E      ***
30 REM *****
*****
40 REM *** INITIALISATION ***
50 DIM FICH(50)
60 ꝀSETSID,1200,1200,0,7,1,0
70 ꝀSETTIMEOUT,50000
80 A=0:R%=0
90 ETB=&17:ÉDT=&4:ACK=6:ENQ=5
100 REM *** MENU ***
110 MODE 2
120 PRINT"RECEPTION D'UN FICHIER DE DONN
EES"
130 PRINT:PRINT
140 INPUT "NOM DU FICHIER";FICH$
150 OPENOUT FICH$
160 REM *** ATTENTE DE CONNEXION DU MINI
TEL SUR LA LIGNE TELEPHONIQUE ***
170 PRINT:PRINT
180 PRINT"ATTENDEZ LA PORTEUSE"
190 PRINT"PUIS APPUYEZ SUR LA TOUCHE CON
NEXION/FIN DU MINITEL"
200 PRINT"APPUYEZ ENSUITE SUR LA TOUCHE
<ESPACE> DE VOTRE AMSTRAD"
210 A$=INKEY$
220 ꝀCLOSESID,àR%
230 ꝀSETSID,1200,1200,0,7,1,0
240 IF A$="" THEN 210
250 REM *** ACCORD AVEC L'EMETTEUR POUR
DEBUTER LA TRANSMISSION ***
260 ꝀSID,àR%
270 IF (R% AND 1) <> 1 THEN 260
280 ꝀINCHAR,àR%,A
290 IF PEEK(A) <> ACK THEN 730
300 ꝀSID,àR%
310 IF (R% AND 4) <> 4 THEN 300
320 PRINT"TRANSFERT EN COURS ... "
330 ꝀOUTCHAR,àR%,ACK
340 REM *** RECEPTION LONGUEUR DU NOMBRE
***
```

```
350 @SID,@R%
360 IF (R% AND 1) <> 1 THEN 350
370 @INCHAR,@R%,A
380 J=PEEK(A)-32
390 REM *** ACCUSE RECEPTION ***
400 @SID,@R%
410 IF (R% AND 4) <> 4 THEN 400
420 @OUTCHAR,@R%,ACK
430 REM *** RECEPTION NOMBRE ***
440 FOR I=1 TO J
450 @INCHAR,@R%,A
460 FICH(I)=PEEK(A)
470 NEXT I
480 REM *** RECEPTION FIN DU NOMBRE ***
490 @INCHAR,@R%,A
500 IF R%=512 THEN 730
510 IF PEEK(A) <> EOT THEN 730
520 REM *** RECONSTITUTION DU NOMBRE ***
530 FICH(0)=0:FICH$(0)="
540 FOR I=1 TO J
550 FICH$(0)=FICH$(0)+CHR$(FICH(I))
560 NEXT I
570 FICH(0)=VAL(FICH$(0))
580 REM *** AFFICHAGE ET SAUVEGARDE ***
590 PRINT FICH(0)
600 WRITE #9,FICH(0)
610 REM *** ENVOI PRET POUR LE NOMBRE SU
IVANT ***
620 @SID,@R%
630 IF (R% AND 4) <> 4 THEN 620
640 @OUTCHAR,@R%,ACK
650 @INCHAR,@R%,A
660 IF R%=512 THEN 730
670 IF PEEK(A) <> EOT THEN 380
680 REM *** FICHIER COMPLET - SAUVEGARDE
DU FICHIER ***
690 CLOSEOUT
700 PRINT"FIN DE TRANSMISSION"
710 PRINT"FICHIER RECU ET SAUVEGARDE"
720 END
730 REM *** ERREUR DANS LA TRANSMISSION
***
740 PRINT:PRINT
750 PRINT"ERREUR FATALE EN COURS DE TRAN
SMISSION !!!!!"
760 PRINT"FICHIER INUTILISABLE"
770 STOP
```

D. ECHANGES DE PROGRAMME PAR LIGNE TELEPHONIQUE

Vous avez écrit des programmes BASIC originaux et intéressants, pourquoi, entre Amstradistes chevronnés, ne pas créer un réseau d'échanges par ligne téléphonique.

Voici deux programmes (Emission et Réception) qui vous permettront les échanges avec une fiabilité remarquable.

Vous pourrez d'ailleurs, à titre d'entraînement, utiliser la méthode de transfert pour l'échange de données.

Contrairement aux fichiers de données, un programme Basic possède des caractères de contrôle (essayez la commande CPM: TYPE PROGRAM.BAS ; vous aurez quelques surprises à l'écran).

Il vous faudra donc sauvegarder le programme sous sa forme ASCII par l'instruction :

SAVE 'PROGRAM',A

De plus les caractères , et " sont considérés comme des séparateurs pour l'instruction

INPUT #9, variable aphanumérique

aussi il nous faudra utiliser

LINE INPUT #9, variable alphanumérique

qui rangera une ligne entière de programme dans la variable.

Afin d'améliorer la fiabilité de l'échange, le programme d'émission, après avoir envoyé un caractère, attend le retour de ce même caractère. Si celui-ci est différent, on le renvoie, jusqu'à ce qu'il soit correct. Une fois correct, un accusé réception est envoyé pour signaler que l'on passe au caractère suivant.

Un accusé réception est envoyé à chaque fin de ligne, puis on attend ACK pour passer à la ligne suivante.

EOT est envoyé en fin de programme, puis la communication est rompue. Le programme récepteur gère le processus inverse, comme nous l'analyserons dans les deux programmes suivants.

```
420 IF (R% AND 4) <> 4 THEN 410
430 OUTCHAR,AR%,B
440 REM *** VERIFICATION DU CARACTERE **
*
450 INCHAR,AR%,A
460 IF R%=512 THEN 840
470 C=PEEK(A)
480 IF C <> B THEN 410
490 REM *** CARACTERE OK ***
500 OUTCHAR,AR%
510 IF (R% AND 4) <> 4 THEN 830
520 OUTCHAR,AR%,ACK
530 NEXT I
540 REM *** FIN D'ENVOI D'UNE LIGNE DE P
ROGRAMME ***
550 OUTCHAR,AR%
560 IF (R% AND 4) <> 4 THEN 550
570 OUTCHAR,AR%,ACK
580 INCHAR,AR%,A
590 IF R% = 512 THEN 840
600 C=PEEK(A)
610 IF C <> ACK THEN 840
620 WEND
630 REM *** PROGRAMME ENVOYE ***
640 OUTCHAR,AR%
650 IF (R% AND 4) <> 4 THEN 640
660 OUTCHAR,AR%,EOT
670 PRINT:PRINT
680 PRINT"TRANSFERT TERMINE"
690 CLOSEIN
700 GOTO 740
710 REM *** CORRESPONDANT NON CONNECTE O
U PAS PRET ***
720 PRINT:PRINT
730 PRINT"VOTRE CORRESPONDANT NE REpond
PAS"
740 RESTORE 810
750 FOR I=1 TO 3
760 READ A
770 OUTCHAR,AR%
780 IF (R% AND 4) <> 4 THEN 770
790 OUTCHAR,AR%,A
800 NEXT I
810 DATA &1B,&39,&67
820 STOP
830 REM *** COMMUNICATION COUPEE OU MAUV
AISE ***
840 PRINT:PRINT
850 PRINT"ERREUR DURANT LE TRANSFERT"
860 GOTO 740
```

- Lignes 310-330 : Le programme émet ACK pour signaler qu'il est prêt et attend durant 10 secondes un caractère signalant la présence du destinataire.
- Ligne 300 : Début du programme d'envoi du fichier ASCII représentant le programme à transmettre.
- Lignes 450-480 : On attend que le caractère précédemment envoyé soit reçu. Si le caractère reçu est erroné, on le renvoie (THEN 410).
- Lignes 500-520 : Envoi de ACK pour signaler que le caractère est bon.
- Lignes 550-570 : Envoi ACK pour signaler que la ligne de programme est terminée.
- Lignes 580-620 : Attente de ACK pour passer à la ligne de programme suivante.
- Lignes 630-700 : Envoi de EOT pour signaler la fin de la transmission, qui se termine par une déconnexion en lignes 740 à 820.
- Lignes 720-730 et lignes 830-860 : Affichage d'une information d'erreur durant la transmission puis déconnexion du minitel à la ligne.

• Programme Réception

```

10 REM *** PROGRAMME DE RECEPTION ***
20 REM *** D'UN PROGRAMME ASCII ***
30 REM *****
40 REM *** INITIALISATION ***
50 MODE 2
60 ꞤSETSID,1200,1200,0,7,1,0
70 ꞤSETTIMEOUT, 10000
80 R%=0:A=0
90 EOT=&4:ACK=&6
100 DIM LIGNE(255)
110 INPUT"NOM DU PROGRAMME";PGM$
120 OPENOUT PGM$
130 PRINT:PRINT
140 PRINT"APPUYER SUR UNE TOUCHE POUR DE
BUTER LE TRANSFERT"
150 A$=INKEY$
160 IF A$="" THEN 150
170 REM *** SIGNALEMENT PRESENCE ***
180 ꞤINCHAR,ꞤR%,A
190 IF R%=512 THEN 570
200 B=PEEK(A)
210 IF B <> ACK THEN 610

```

```
220 @SID,@R%
230 IF (R% AND 4) <> 4 THEN 220
240 @OUTCHAR,@R%,ACK
250 REM *** RECEPTION D'UNE LIGNE DE PRO
GRAMME ***
260 I=0
270 @INCHAR,@R%,A
280 IF R%=512 THEN 610
290 B=PEEK(A)
300 IF B=ACK THEN 450
310 IF B=EOT THEN 530
320 REM *** RENVOI DU CARACTERE ***
330 @SID,@R%
340 IF (R% AND 4) <> 4 THEN 330
350 @OUTCHAR,@R%,B
360 REM *** VERIFICATION DE LA VALIDITE
***
370 @INCHAR,@R%,A
380 C=PEEK(A)
390 IF C <> ACK THEN 290
400 LIGNE(I)=B
410 REM *** CARACTERE SUIVANT ***
420 I=I+1
430 GOTO 270
440 REM *** RECONSTITUTION LIGNE DE PROG
RAMME ***
450 LIGNE$=""
460 FOR J=0 TO I
470 B$=CHR$(LIGNE(J))
480 LIGNE$=LIGNE$+B$
490 NEXT J
500 PRINT#9,LIGNE$
510 GOTO 260
520 REM *** FERMETURE DU FICHIER PROGRAM
ME ***
530 CLOSEOUT
540 PRINT:PRINT
550 PRINT"PROGRAMME TRANSFERE"
560 END
570 PRINT:PRINT
580 PRINT"L'EMETTEUR NE S'EST PAS MANIFE
STE"
590 CLOSEOUT
600 STOP
610 PRINT:PRINT
620 PRINT"ERREUR DANS LA TRANSMISSION"
630 PRINT"PROGRAMME INUTILISABLE"
640 CLOSEOUT
650 STOP
```

- Ligne 100 : Dimensionnement d'un tableau qui recevra caractère par caractère une ligne de programme pouvant aller jusqu'à 255, maximum admis.
- Lignes 110-120 : Le fichier est préparé de suite afin de gagner du temps lors de la transmission.
- Lignes 170-240 : Après réception de ACK lui demandant de se signaler, le programme envoie ACK.
- Ligne 260 : On pointe la première variable du fichier LIGNE(I) puis la retransmission débute.
- Ligne 300 : Si on a reçu ACK, il faut aller traiter la fin de ligne entrée.
- Ligne 310 : Sur réception de EOT, le programme est totalement reçu.
- Lignes 320-390 : Le caractère reçu est renvoyé aussitôt, et sur réception de ACK il est sauvé (ligne 400) sinon il faut aller le relire (THEN 290).
- Lignes 440-500 : Une ligne de programme peut être reconstituée, et elle est de suite sauvegardée. Cette méthode permet de transférer des programmes sans limitation de taille (tout dépendra de la place qu'il reste sur votre disquette, méfiez-vous du « DISK FULL »).
- Lignes 530-560 : Fermeture du fichier et fin du programme.
- Lignes 570-650 : Affichage des éventuelles erreurs de transmission pouvant intervenir.

E. UN REPONDEUR TELEMATIQUE

Qui n'a pas envie de posséder un répondeur/enregistreur chez lui afin de prendre les communications importantes qui pourraient lui être transmises.

Le minitel étant de plus en plus répandu, votre correspondant reconnaîtra automatiquement la tonalité aiguë de la porteuse, l'invitant à brancher son minitel et à se connecter.

Votre Amstrad s'occupera alors de prendre les éventuels messages que vous pourrez relire ultérieurement.

Si vous êtes présent, vous verrez le nom de votre correspondant s'inscrire sur votre écran, et vous pourrez couper la communication télématique pour reprendre oralement (Fig. 5).

Pour réaliser votre répondeur : un AMSTRAD avec lecteur de disquette, une interface RS-232C avec le montage précédent, un minitel connecté à l'interface, le montage du détecteur de sonnerie (voir la Partie 10, chap. 4.3) et les programmes qui suivent.

Le montage détecteur de sonnerie sera connecté sur la fiche manette de jeu de votre Amstrad. Une vérification de son fonctionnement pourra être visuelle. Lorsque la sonnerie retentira, vous devrez voir apparaître sur votre écran une des flèches de direction ou le caractère X. Si ce n'est pas le cas, vérifiez les raccordements.

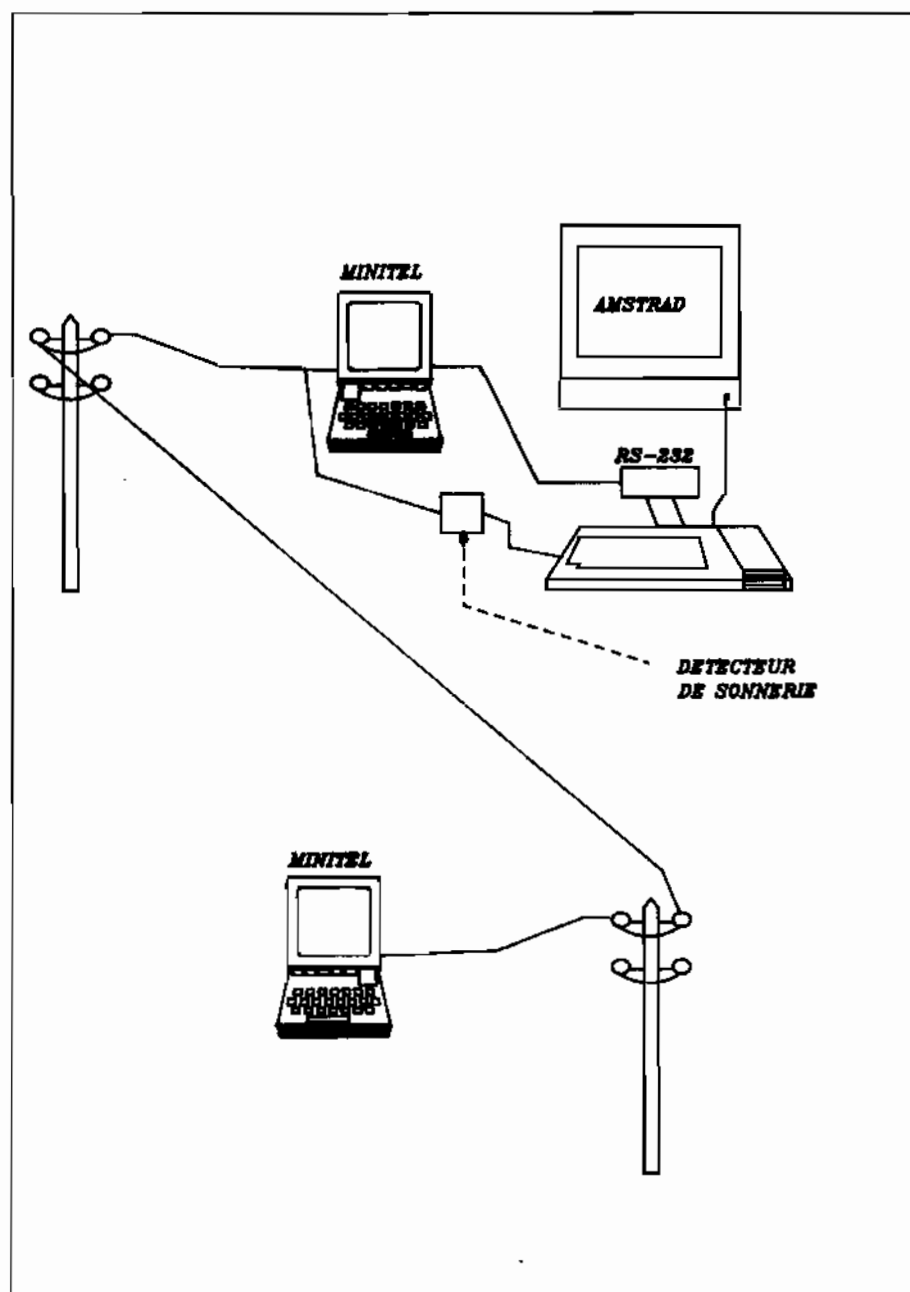


Fig. 5

- **Programme d'initialisation du fichier messages**

Ce programme permet de créer un fichier nommé « MESSAGE. » qui sauvegardera tous les messages créés par vos correspondants.

Il vous permettra aussi d'effectuer tous les messages que vous aurez préalablement lus.

```

10 REM *** PROGRAMME PREPARANT UN FICH
IER DE ***
20 REM *** MESSAGES POUR LE REPONDEUR TE
LEMATIQUE ***
30 REM *****
*****
40 MODE 1
50 PRINT"ATTENTION CE PROGRAMME EFFACE T
OUS LES"
60 PRINT"MESSAGES DU FICHIER, MEME S'IL
N'ONT"
70 PRINT"PAS ETE ENCORE LUS"
80 PRINT:PRINT
90 INPUT"FRAPPER 2 FOIS <ESC> POUR ABAND
ONNER SINON <RETURN>";A
100 PRINT"FICHIER EN PREPARATION ..."
110 REM *** ECRIT UN MESSAGE PERMETTANT
***
120 REM *** LA CREATION DU FICHIER
***
130 OPENOUT"MESSAGE"
140 PRINT#9,"XXXXXX"
150 PRINT#9,"PREMIERE PARTIE DU MESSAGE
PERMETTANT"
160 PRINT#9,"DE CREER LE FICHIER MESSAGE
"
170 CLOSEOUT
180 PRINT:PRINT
190 PRINT"VOTRE FICHIER EST PRET A ETRE
UTILISE"
200 PRINT:PRINT
210 PRINT"LANCEZ LE PROGRAMME REPONDEUR"
220 END

```

Il est conseillé de sauvegarder ce programme sous le nom « INI-MES.BAS », car il est appelé par le programme de lecture des messages.

• Programme du répondeur

```

10 REM *** PROGRAMME REPONDEUR/ENREGISTR
EUR **
20 REM *** DE MESSAGES TELEMATIQUES
***
30 REM *****
*****
40 REM *** INITIALISATION ***
50 MODE 1
60 &SETSID,1200,1200,0,7,1,0
70 &SETTIMEOUT,5000
80 RZ=0:A=0

```

```
90 DIM MES(400),NDM(20)
100 FOR I=1 TO 3
110 READ A
120 USIO,AR%
130 IF (R% AND 4) <> 4 THEN 120
140 OUTCHAR,AR%,A
150 NEXT I
160 DATA &1B,&39,&67
170 REM *** ATTENTE APPEL ***
180 PRINT"COMMUNICATION EN ATTENTE..."
190 IF JOY(0) = 0 THEN 190
200 REM *** ENVOI PORTEUSE ***
210 RESTORE 280
220 FOR I=1 TO 6
230 READ A
240 USIO,AR%
250 IF (R% AND 4) <> 4 THEN 240
260 OUTCHAR,AR%,A
270 NEXT I
280 DATA &1B,&39,&6F,&1B,&39,&68
290 REM *** TEMPO A LA CONNEXION ***
300 CLS:PRINT"CORRESPONDANT EN LIGNE"
310 FOR I=1 TO 600
320 CLOSESIO,AR%
330 SETSIO,1200,1200,0,7,1,0
340 CLEAR INPUT
350 NEXT I
360 REM *** PRESENTATION DU REPONDEUR **
*
370 RESTORE 420
380 J=3
390 GOSUB 2270
400 J=1
410 GOSUB 2350
420 DATA &1F,&40,&41
430 DATA VEUILLEZ PATIENTER...SVP
440 RESTORE 470
450 J=8
460 GOSUB 2270
470 DATA &1F,&40,&41,&20,&12,&67
480 DATA &0C,&11
490 RESTORE 580
500 J=21
510 GOSUB 2270
520 J=1
530 GOSUB 2350
540 J=5
550 GOSUB 2270
560 J=1
570 GOSUB 2350
```

```
580 DATA &1F,&41,&41,&1B,&52,&20,&12,&67
590 DATA &1F,&42,&41,&1B,&52,&20,&12,&67
600 DATA &1F,&41,&42,&1B,&40
610 DATA "Bonjour, vous etes connecte sur le"
620 DATA &1F,&42,&42,&1B,&40
630 DATA repondeur minitel de Michel RIO
UART
640 REM *** DEMANDE DU NOM ***
650 RESTORE 760
660 J=3
670 GOSUB 2270
680 J=1
690 GOSUB 2350
700 J=2
710 GOSUB 2270
720 J=1
730 GOSUB 2350
740 J=7
750 GOSUB 2270
760 DATA &1F,&44,&41
770 DATA Votre NOM(20 caracteres maxi)+
780 DATA &1b,&5D
790 DATA ENVOI
800 DATA &1F,&45,&41,&2E,&12,&54,&0D
810 AFTER 20000,1 GOSUB 1370
820 M=1
830 FOR i=1 TO 20
840 K=I
850 @SID,@R%
860 IF (R% AND 1) <> 1 THEN 850
870 @INCHAR,@R%,A
880 IF PEEK(A)=&13 THEN GOTO 1840
890 NOM(I)=PEEK(A)
900 @OUTCHAR,@R%,PEEK(A)
910 NEXT I
920 REM *** RECONSTITUTION DU NOM DU COR
RESPONDANT ***
930 I=20
940 GOSUB 1960
950 PRINT NOM$
960 REM *** POSSIBILITE DE LAISSER UN ME
SSAGE DE 400 CARACTERES ***
970 RESTORE 1020
980 J=3
990 GOSUB 2270
1000 J=2
1010 GOSUB 2350
1020 DATA &1F,&47,&41
1030 DATA "Vous disposez de 10 lignes e
```

```
t de 5 minu-"  
1040 DATA "tes pour laisser un message"  
1050 J=3  
1060 GOSUB 2270  
1070 DATA &1F,&49,&41  
1080 FOR Z=1 TO 10  
1090 RESTORE 1120  
1100 J=3  
1110 GOSUB 2270  
1120 NEXT Z  
1130 DATA &5F,&12,&67  
1140 J=3  
1150 GOSUB 2270  
1160 J=3  
1170 GOSUB 2350  
1180 DATA &1F,&54,&41  
1190 DATA "ATTENTION, seules les touches  
<ENVOI> et"  
1200 DATA "<CORRECTION> sont autorisees,  
vous cou- "  
1210 DATA "periez la communication sinon  
!"  
1220 J=3  
1230 GOSUB 2270  
1240 DATA &1F,&49,&41  
1250 M=2  
1260 FOR I=1 TO 400  
1270 K=I  
1280 @SIO,@R%  
1290 IF (R% AND 1) <> 1 THEN 1280  
1300 @INCHAR,@R%,A  
1310 IF PEEK(A)=&13 THEN GOTO 2040  
1320 MES(I)=PEEK(A)  
1330 @OUTCHAR,@R%,PEEK(A)  
1340 PRINT CHR$(PEEK(A));  
1350 NEXT I  
1360 REM *** RECONSTITUTION DU MESSAGE *  
**  
1370 GOSUB 2160  
1380 REM *** DECONNEXION DU CORRESPONDAN  
T ***  
1390 DI  
1400 RESTORE 1550  
1410 j=4  
1420 GOSUB 2270  
1430 j=1  
1440 GOSUB 2350  
1450 j=3  
1460 GOSUB 2270  
1470 j=1
```

```
1480 GOSUB 2350
1490 j=3
1500 GOSUB 2270
1510 FOR I=1 TO 1000
1520 NEXT I
1530 J=6
1540 GOSUB 2270
1550 DATA &0C,&1F,&45,&43
1560 DATA VOTRE MESSAGE EST ENREGISTRE
1570 DATA &1F,&47,&43
1580 DATA MERCI DE VOTRE APPEL....AU REV
DIR
1590 DATA &1B,&39,&67,&1B,&39,&6F,&1B,&3
9,&68
1600 REM *** ENREGISTREMENT DU MESSAGE *
**
1610 REM *** A LA SUITE DES PRECEDENTS *
**
1620 REM *** DANS LE FICHIER SUR DISQUET
TE ***
1630 IF M=1 THEN RUN
1640 OPENIN "MESSAGE"
1650 OPENOUT "MESSAGE.TEM"
1660 WHILE NOT EOF
1670 LINE INPUT#9, NOM2$
1680 PRINT#9,NOM2$
1690 LINE INPUT#9, MES4$
1700 PRINT#9,MES4$
1710 LINE INPUT#9, MES4$
1720 PRINT#9,MES4$
1730 WEND
1740 PRINT#9,NOM$
1750 PRINT#9,MES1$
1760 PRINT#9,MES2$
1770 CLOSEIN
1780 CLOSEOUT
1790 UERA,"MESSAGE"
1800 UREN,"MESSAGE","MESSAGE.TEM"
1810 RUN
1820 REM *** TRAITEMENT TOUCHE FONCTION
***
1830 REM *** DURANT ENTREE DU NOM ***
1840 USIO,AR%
1850 IF (R% AND 1) <> 1 THEN 1840
1860 UINCHAR,AR%,A
1870 IF PEEK(A) <> &47 THEN 1920
1880 I=I-2
1890 POKE A,&8
1900 IF I<0 THEN I=0:POKE A,&0
1910 GOTO 900
```

```
1920 I=20
1930 POKE A,&21
1940 GOTO 900
1950 REM *** RECONSTITUTION NOM ***
1960 NOM$=""
1970 FOR J=1 TO I
1980 IF NOM(J)=0 THEN NOM(J)=&2E
1990 NOM$=NOM$+CHR$(NOM(J))
2000 NEXT J
2010 RETURN
2020 REM *** TRAITEMENT TOUCHE FONCTION
***
2030 REM *** DURANT ENTREE DU MESSAGE
***
2040 @SIO,@R%
2050 IF (R% AND 1) <> 1 THEN 2040
2060 @INCHAR,@R%,A
2070 IF PEEK(A) <> &47 THEN 2120
2080 I=I-2
2090 POKE A,&8
2100 IF I<0 THEN I=0:POKE A,&0
2110 GOTO 1330
2120 I=400
2130 POKE A,&21
2140 GOTO 1330
2150 REM *** RECONSTITUTION DU MESSAGE *
**
2160 MES1$="":MES2$=""
2170 FOR J=1 TO 200
2180 IF MES(J)=0 THEN MES(J)=&2E
2190 MES1$=MES1$+CHR$(MES(J))
2200 NEXT J
2210 FOR J=201 TO 400
2220 IF MES(J)=0 THEN MES(J)=&2E
2230 MES2$=MES2$+CHR$(MES(J))
2240 NEXT J
2250 RETURN
2260 REM *** ENVOI D'UN LOT DE CODES ASC
II ***
2270 FOR I=1 TO J
2280 READ A
2290 @SIO,@R%
2300 IF (R% AND 4) <> 4 THEN 2290
2310 @OUTCHAR,@R%,A
2320 NEXT I
2330 RETURN
2340 REM *** ENVOI D'UNE CHAINE DE CARAC
TERES ***
2350 FOR I=1 TO J
2360 READ A$
```

```
2370 FOR K=1 TO LEN(A$)
2380 B$=MID$(A$,K,1)
2390 B=ASC(B$)
2400 USIO,AR%
2410 IF (R% AND 4) <> 4 THEN 2400
2420 OUTCHAR,AR%,B
2430 NEXT K
2440 NEXT I
2450 RETURN
```

Après initialisation et dimensionnement des variables MES(I) et NOM(I) respectivement utilisées pour stocker les valeurs ASCII des caractères du Message et du nom du correspondant, voici les grandes lignes du programme.

— Lignes 100-160 : L'Amstrad assure la déconnexion du minitel qui est effectuée en lignes 1530-1540.

— Ligne 190 : Tant que le port Joystick ne répond pas (détecteur de sonnerie), on attend.

— Lignes 200-280 : Le minitel est placé en mode d'opposabilité (il émettra à 1 200 bauds et recevra à 75 bauds) puis sera connecté à la ligne téléphonique pour mettre sa porteuse.

— Lignes 300-350 : Un message est envoyé sur l'écran de l'Amstrad pour signaler la présence d'un interlocuteur. Une attente d'environ une dizaine de secondes permettra au correspondant d'allumer son minitel et de se connecter.

— Lignes 370-800 : Le répondeur vous présente (à ce sujet, n'oubliez pas de modifier la ligne 630) et invite le correspondant à s'identifier sur une ligne de 20 caractères.

— Ligne 810 : Un temps limite de communication est fixé, ce temps se calcule par la valeur du timer activé, ici le timer 1, multipliée par 0.02 s, on a donc un temps de $20000 \times 0.02 = 400$ s, environ 6 mn et demi (contrairement aux 5 minutes annoncées sur le minitel pour presser un peu les paresseux du clavier). Ces 6,5 mn écoulées, le programme se rendra en ligne 1350. Cette interruption ne sert pas uniquement à presser votre correspondant, mais aussi à déconnecter votre minitel de la ligne au cas où votre correspondant se serait intempestivement volatilisé. En effet le minitel utilisé en émission à 1 200 bauds ne reconnaît pas la perte de la communication, ce qui aurait pour fâcheuse incidence de faire boucler le programme du répondeur et de garder la ligne occupée jusqu'à votre retour.

— Lignes 820-910 : Acquisition des 20 caractères du nom. Si le code &13 est reçu — cela signifie que votre correspondant a appuyé sur une des touches de fonction de son minitel (<ENVOI> <SUITE> <RETOUR> <CORRECTION>...) — on saute à une « procédure » de traitement.

On remarquera que chaque caractère reçu est aussitôt renvoyé au minitel appelant ; en effet en mode concerné, le minitel ne visualise pas les caractères frappés, c'est le correspondant qui doit lui renvoyer (en mode non connecté, c'est le modem interne du minitel qui boucle sur lui-même pour renvoyer le caractère à l'écran).

— Lignes 920-950 : Le nom du correspondant est reconstitué puis affiché sur l'écran de l'Amstrad, pour que vous puissiez, si vous êtes présent, prendre la conversation oralement.

— Lignes 960-1240 : On invite le correspondant à laisser un message de 10 lignes de 40 caractères. Ces dix lignes sont affichées à l'écran (lignes 1080-1130).

— Lignes 1240-1350 : Attente d'un message de 400 caractères. Comme pour la boucle d'attente du nom, une touche fonction appuyée est reconnue et un branchement s'effectue pour le traitement.

— Ligne 1370 : Saut à un sous-programme de reconstitution du message reçu.

— Lignes 1380-1500 : Votre minitel est déconnecté de la ligne, ce qui a pour incidence de déconnecter le minitel de votre correspondant (si par exemple celui-ci avait dépassé le temps imparti, ou avait oublié son minitel pour une raison plus importante que votre message : sachez que les services Teletel font de même quand vous n'appuyez pas sur une touche durant 5 mn).

— Lignes 1510-1540 : La ligne est de nouveau mise en occupation par connexion du minitel afin de pouvoir sauvegarder le message sur disquette, ce qui prend un temps non négligeable car le fichier « MESSAGE. » est un fichier à accès séquentiel.

— Lignes 1600-1800 : Le nouveau message doit être enregistré à la suite des précédents messages. Pour cela le fichier « MESSAGE. » est relu dans des variables alphanumériques et sauvé au fur et à mesure dans un nouveau fichier temporaire appelé « MESSAGE.TEM ». Le nouveau message est enfin sauvegardé à la suite. L'ancien fichier « MESSAGE. » est effacé, puis « MESSAGE.TEM » est renommé sous le nom « MESSAGE. »

Cette méthode est intéressante car elle permet de ne pas occuper trop de place en mémoire en emmagasinant tous les messages (si vous vous absentez quelques jours, les messages risquent d'affluer). De plus, si une coupure de secteur EDF intervenait au moment de la sauvegarde, vous aurez plus de chance de récupérer votre fichier.

— Lignes 1820-1940 : On teste ici quelle est la touche fonction appuyée durant l'entrée du nom. Si cette touche est différente de la touche <CORRECTION> (code &47) alors on considère que l'utilisateur a terminé d'inscrire son identification. Sinon, on pointe deux variables plus haut dans le tableau NOM(1), puis on prépare l'envoi d'un caractère BS (&08, retour d'un caractère en arrière qui sera renvoyé au minitel).

- Lignes 2020-2140 : Traitement de la frappe d'une touche fonction durant la réception du message.
- Lignes 2150-2250 : Reconstitution du message, mais dans deux variables car celui-ci peut contenir jusqu'à 400 caractères. Les variables nulles du tableau MES(I) seront remplacées par des points (&2E).
- Lignes 2260-2330 : Ce sous-programme permet d'envoyer un nombre J de caractères contenus dans une ligne de DATAs.
- Lignes 2340-2450 : Ce sous-programme envoie caractère par caractère un nombre J de chaînes alphanumériques contenues en DATA. Remarque : On prendra soin d'entourer la chaîne de caractères avec des guillemets si une virgule est contenue dans celle-ci, sinon le basic considère qu'il a affaire à deux chaînes distinctes et ignore la virgule qui est un séparateur de données.

• Programme de lecture des messages

De retour chez vous, votre empressement sera de relire les messages reçus. Ce petit programme vous permettra de l'effectuer sur l'écran de votre Amstrad.

```

10 REM *** PROGRAMME DE LECTURE DE MESSA
GES ***
20 REM *****
*****
30 MODE 1
40 OPENIN"MESSAGE"
50 REM *** ANNULATION DU PREMIER MESSAGE
***
60 REM *** QUI EST LE MESSAGE DE CREATIO
N **
70 REM ***      DU FICHIER MESSAGE
***
80 FOR I=1 TO 3
90 LINE INPUT#9,A$
100 NEXT I
110 WHILE NOT EOF
120 CLS
130 LINE INPUT#9,A$
140 PRINT"MESSAGE DE ";A$
150 PRINT
160 LINE INPUT#9,A$
170 PRINT A$;
180 LINE INPUT#9,A$
190 PRINT A$
200 PRINT:PRINT
210 INPUT"APPUYEZ SUR <RETURN> POUR LIRE
LE      MESSAGE SUIVANT";A
220 WEND

```

```
230 PRINT:PRINT
240 PRINT"PLUS DE MESSAGES"
250 PRINT:PRINT
260 INPUT"DESIREZ VOUS EFFACER LE FICHIE
R DE      MESSAGES (O/N) ";A$
270 A$=UPPER$(A$)
280 IF A$<>"O" THEN STOP
290 RUN"INIMES"
```

— Lignes 50-100 : Les trois premières chaînes alphanumériques du fichier sont celles qui nous ont permis de le créer, elles sont donc ignorées à l'affichage.

— Lignes 110-220 : TANT QUE la fin du fichier n'est pas atteinte

Afficher le message suivant

FIN TANT QUE

— Lignes 260-290 : Le programme vous demande confirmation pour lancer le programme « INIMES.BAS » afin d'effacer tous les messages.

• Essais du programme

Si vous désirez essayer le programme de répondeur, il faudra que vous empruntiez un minitel à un ami. Vous le connecterez sur une deuxième prise si vous en possédez une chez vous — sinon vous reliez borne à borne à l'aide de fils les bornes 1 et 3 des conjoncteurs mâles (voir Partie 10, chap. 4.3 page 5).

Un joystick remplacera le détecteur de sonnerie.

Vous lancez le programme puis actionnez le joystick. Un claquement discret doit se faire entendre dans le minitel connecté à la RS232-C. Il est temps de connecter le minitel emprunté. Devant celui-ci, vous serez le correspondant vous appelant et vous n'aurez qu'à suivre les directives.

Prendre la conversation en cours

Lorsque l'on vous appellera, si vous désirez prendre la conversation en cours, rien de plus simple. Frappez deux fois <ESC>, puis installez-vous devant votre minitel. Vous inviterez alors votre correspondant à décrocher le combiné de son téléphone par l'intermédiaire du clavier de votre minitel ; un seul petit problème, vous frapperez vos caractères à l'aveugle car le minitel appelant ne renverra pas votre texte. Toujours de cette façon, vous l'inviterez à frapper deux fois sur <CONNEXION/FIN>. De votre côté, faites de même : la conversation orale peut débuter.

Si vous n'appréciez pas de frapper à l'aveugle, ce petit programme vous permettra d'envoyer les directives à votre correspondant. Il est à entrer à la suite du programme de répondeur. Après avoir fait deux fois <ESC>, vous frapperez « GOTO 5000 ». Quelques secondes plus tard (laissez votre correspondant réagir) vous pourrez prendre votre combiné téléphonique et déconnecter votre minitel.

5000 RESTORE 6000

5010 J=3

5020 GOSUB 2270

5030 J=2

5040 GOSUB 2350

5050 STOP

6000 DATA &OC,&OC,&OC

6010 DATA Veuillez s'il vous plaît prendre l'ecou-

6020 DATA teur et appuyer sur CONNEXION/FIN

8/5.3

Les modems

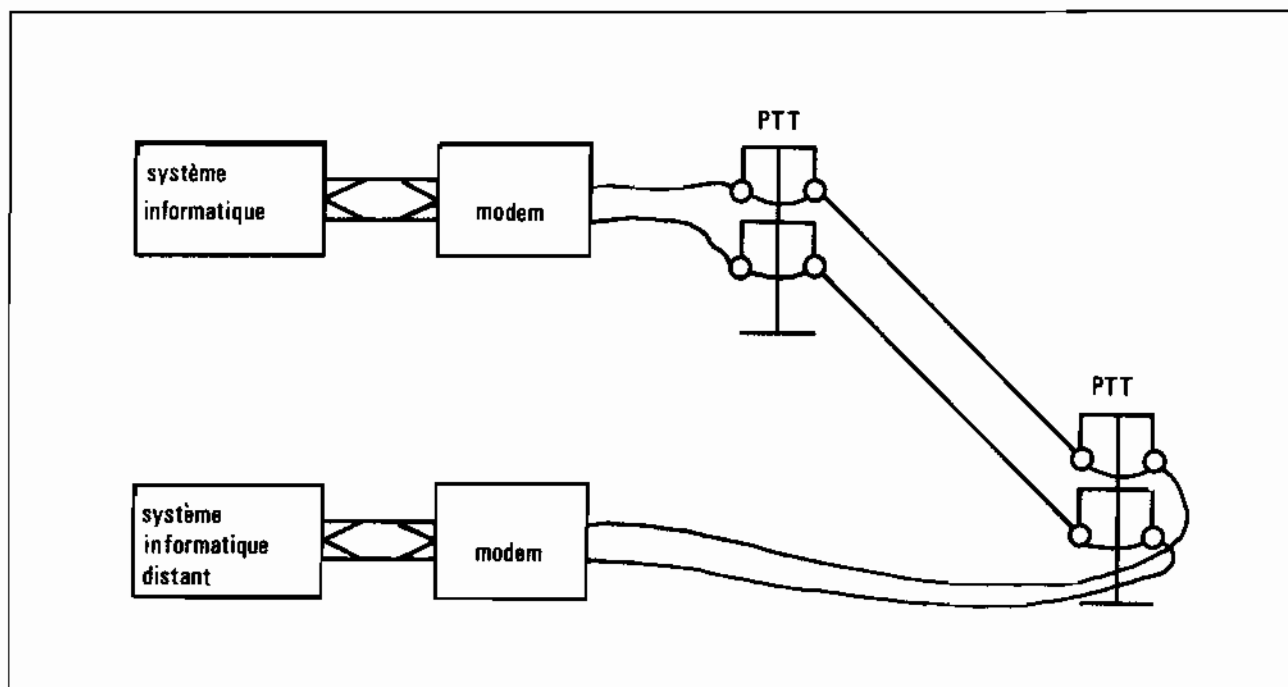
Pourquoi un modem ?

Le nom modem vient de la contraction du terme « modulateur-démodulateur ».

Pour transmettre des informations sur des distances très importantes, le plus simple est d'utiliser une ligne téléphonique.

Les lignes disponibles par l'utilisateur ne sont pas prévues pour accepter directement les informations en binaires, les fréquences admises se situant entre 300 Hz et 3,3 kHz environ.

On a donc recours à un modem qui va moduler le signal lors de l'émission pour l'adapter à la ligne téléphonique, et qui va le démoduler à la réception.

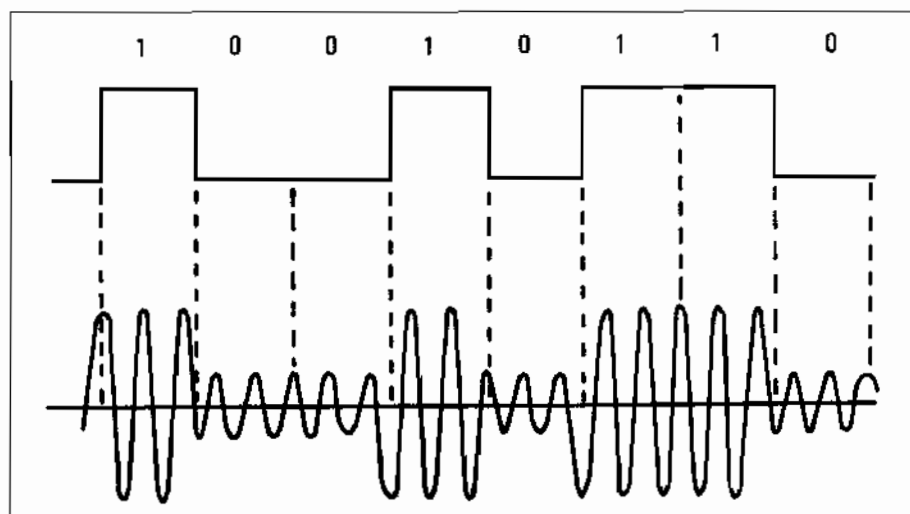


Les modems peuvent entre autres posséder un système d'appel automatique (il compose le numéro du demandé : structure utile pour interroger une banque de données éloignée la nuit, en tarif réduit), ou un système de réponse automatique (lors de la réception d'un appel : principe du serveur).

Les vitesses de transmission généralement utilisées sont de 133, 200, 300, 600, 1 200, 2 400, 4 800, 9 600 bits par seconde, et ce, dans différentes techniques de modulation.

La modulation d'amplitude

Le signal envoyé par le modem est une onde de fréquence fixe mais dont le niveau varie selon la valeur du bit transmis.



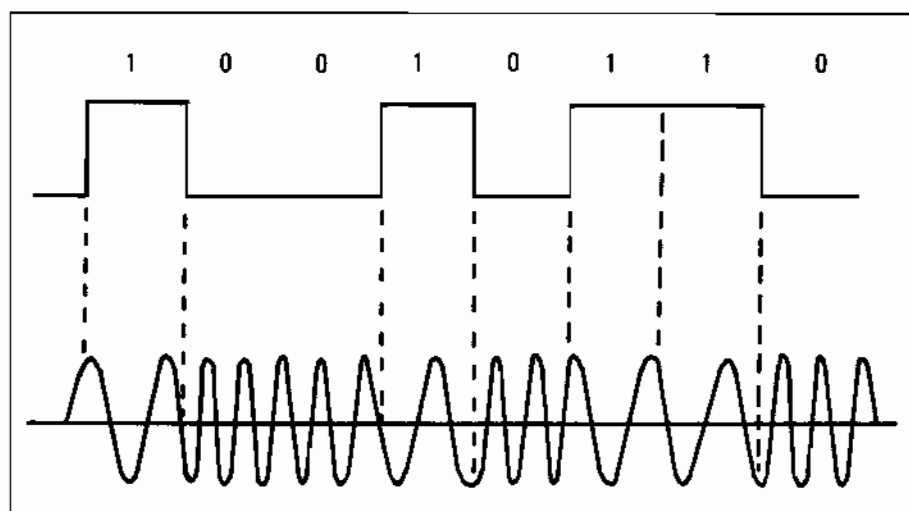
Deux techniques de modulation d'amplitude (AM) peuvent être employées : l'ASK (*Amplitude Shift Keying*) et la BLU (*Bande Latérale Unique*).

La modulation de fréquence

C'est le principe de codage le plus utilisé actuellement pour les modems car peu sensible aux parasites, et de fabrication économique.

Dans ce type de modulation, l'amplitude du signal émis reste fixe, c'est la fréquence du signal qui varie selon l'état du bit à transmettre.

Deux fréquences différentes seront utilisées. Dans le cas du minitel, en réception à 1 200 bauds, la fréquence du « 1 » logique est 1 300 Hz, celle du « 0 » est 2 100 Hz.

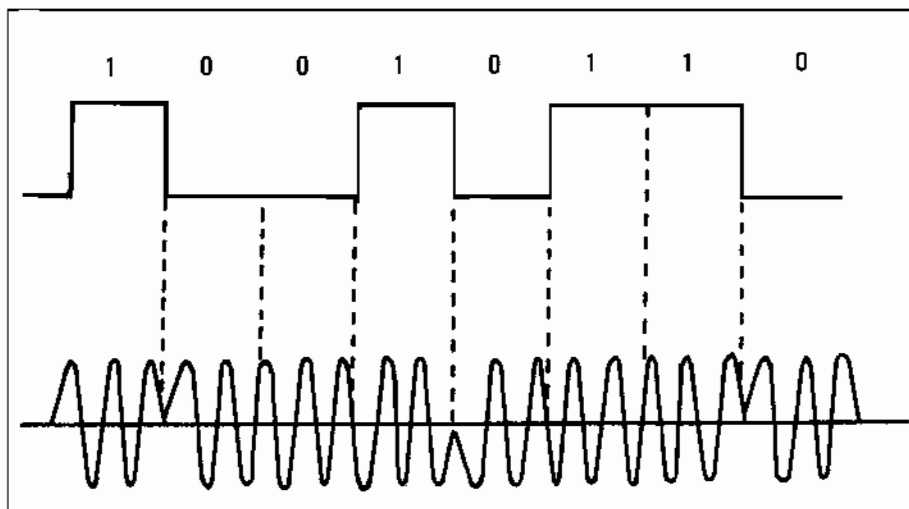


Ce type particulier de modulation de fréquence est appelé modulation FSK (*Frequency Shift Keying*), car les deux fréquences sont situées autour d'une fréquence porteuse de 1 700 Hz, à laquelle est retranchée ou ajoutée ce que l'on appelle une excursion de fréquence de 400 Hz pour obtenir les deux précédentes fréquences. Ce principe de codage est défini par ce que l'on appelle l'avis V23 du CCITT.

La modulation de phase

Ce type de modulation tend à s'imposer sur les modems demandant une très grande fiabilité dans la transmission de l'information numérique car très peu sensible aux parasites, donc utilisable pour la transmission à très grande vitesse.

Appelée DPSK (*Differential Phase Shift Keying*), ce type de modulation produit un retard de la fréquence porteuse pour un bit à « 1 » par rapport à la fréquence émise pour un bit à « 0 ».



Le couplage sur la ligne

Deux méthodes peuvent être utilisées pour envoyer le signal sur la ligne téléphonique :

- le signal est envoyé directement sur la ligne par l'intermédiaire d'un adaptateur. Cette méthode exige l'utilisation d'un matériel agréé par les PTT ;
- un coupleur acoustique sur lequel on vient poser le combiné téléphonique donne aussi de bons résultats et ne nécessite pas l'agrément PTT. Généralement ce type de coupleur émet en modulation de fréquence.

Quel modem pour l'Amstrad ?

Deux cas se présentent :

- Vous possédez une interface RS-232, dans ce cas vous pouvez utiliser n'importe quel modem se connectant à celle-ci ; votre choix se situera au niveau des fréquences de porteuse et au type de modulation désirés. Pour des transmissions fiables par la ligne téléphonique, vous devrez vous limiter à 4 800 bauds au maximum, sinon il vous faudra louer votre propre ligne spécialisée.

Un modem très économique est le modem du minitel, puisqu'il est gratuit, mais vous serez contraint à une vitesse d'émission 75 bauds (1 200 bauds en réception), ou l'inverse, si vous avez un minitel retournable. Quelques astuces vous permettront d'émettre et de recevoir à 1 200 bauds, mais vous serez cette fois-ci contraint à une transmission dans un seul sens à la fois.

- Vous ne possédez pas d'interface RS-232, ou celle que vous possédez n'est pas complète et ne permet pas de piloter un modem : il existe certains modems possédant leur propre interface série qui se connecte directement sur le bus d'extensions disponible derrière votre ordinateur.

8/5.3.1

Le modem Digitelec DTL 2000

L'univers de la communication est à votre portée grâce aux lignes téléphoniques, votre ordinateur et un modem. La société *Digitel* fabrique différents types de modems se connectant sur votre Amstrad : les séries DTL 2000 et DTL 2100 sont intéressantes pour l'amateur.

Comme vous trouverez en partie 10, chapitre 4.3 un montage permettant à votre CPC de détecter un appel sur votre ligne téléphonique nous ne nous intéresserons pas de ce fait au DTL 2100 qui possède son détecteur intégré.

Le modem qui nous intéressera tout particulièrement sera le DTL 2000+, qui a la particularité de pouvoir fonctionner en mode V23 Réponse Full Duplex, c'est-à-dire qu'il peut émettre sur la ligne à la vitesse de 1200 bauds et recevoir à 75 bauds. Nous pouvons donc déduire que ce modem pourra se connecter à un minitel via la ligne téléphonique.

Toutes les possibilités du DTL 2000+ sont résumées ci-dessous :

- V23 appel en Full-Duplex, c'est-à-dire que le modem se comporte comme le modem du minitel (émission à 75 bauds et réception à 1200 bauds), donc que l'on pourrait émuler le fonctionnement d'un minitel grâce à l'Amstrad.
- V23 réponse en Full-Duplex (voir ci-dessus).
- V23 Half Duplex, permet de communiquer rapidement avec un autre modem, configuré de la même façon, à 1200 bauds en émission et réception.
- V21 appel en Full-Duplex, permet d'entrer directement sur le réseau Transpac (émission et réception à 300 bauds).
- V21 réponse en Full-Duplex (émission et réception à 300 bauds sur Transpac).

I. Connexion à l'Amstrad

Le modem DTL2000+ a la particularité de posséder une interface série simplifiée, donc d'éviter le recours à une interface RS232, ce qui lui permet de se connecter directement sur le bus d'extension de votre Amstrad. En prenant soin de manipuler délicatement le connecteur 2 x 25 broches, l'enfichage à l'arrière du CPC ne doit poser aucun problème.

Un autre câble sur lequel est montée une fiche gigogne sera connecté sur la ligne téléphonique, le poste venant ensuite dessus.

Enfin, et seulement maintenant, vous pouvez brancher la prise d'alimentation 220 V. Vos périphériques, si vous en avez, seront ensuite mis sous tension, et en dernier lieu, votre ordinateur.

Le modem est prêt à fonctionner, vous pouvez charger le logiciel d'émulation minitel livré avec, ou créer vos propres programmes.

II. L'émulateur minitel

UTILISATION

Le modem est livré avec un programme d'émulation de terminal minitel, somme toute assez sommaire, mais suffisant si vous désirez utiliser l'annuaire électronique. Le programme est livré sur cassette et ne pose aucun problème pour être transféré sur disquette, moyen tout de même plus rapide pour ce type d'utilisation.

Une fois lancé par RUN « DTL 2000 », vous vous trouverez face à une page d'accueil vous permettant de choisir entre :

- faire connaissance avec les touches de fonctionnement propres au minitel (celles-ci sont résumées sur la figure à la page 4),
- procéder à un appel V23, ou un appel V21.

L'annuaire électronique sera choisi par l'option appel en V23. Une fois la connexion établie et une demande effectuée, vous vous trouverez face à des pages écran minitel telles qu'elles sont représentées sur les copies graphiques suivantes :

```

Connexion établie                                     9
0.. WEKA
PARIS PARIS                                           1
-----
Editions Weka
1 -12 cour St Eloi      (1) 43 07 60 50
  12e
2 -poste de télécopie T(1) 43 46 06 16
  même adresse

SUIVEZ LA FLECHE DU 11
Vous trouverez:
plans d'accès, horaires,
produits, services...

page suivante → SUITE
  
```

```

fin de communication
norme CEPT 2
T1 36 13
T3 36 15
T3P 36 16
T4 36 17
F/mn TTC serveur télécom
t20 0,37 0,37
t2 0,37 0,37
tarif réduit du téléphone

Minitel Guide des Services,
tapez:MGS

code
du service:"0.....

Copr 83 DGT
fin Envoi
affichage du prix Cx/Fin Sommaire

```

Comme vous pourrez le remarquer, la qualité graphique n'est pas excellente, mais pour une utilisation en mode texte, cet émulateur est largement suffisant.

SAUVEGARDE DE PAGES MINITEL

Lors de la déconnexion, la page minitel affichée sur le moniteur de l'Amstrad est effacée pour laisser place au menu, aussi une sauvegarde éventuelle est-elle intéressante pour relire ultérieurement.

Le logiciel ne possédant pas d'option permettant la sauvegarde de pages écran, nous vous proposons donc une façon de procéder pour l'y insérer.

Lancez le programme et effectuez une connexion sur l'annuaire électronique. Une fois la page de présentation affichée, appuyez sur la touche <COPY> (commande de déconnexion), et aussitôt sur <ESC> que vous maintiendrez enfoncée plusieurs fois de suite. Vous verrez apparaître le message **Break in XXXX** (sur notre logiciel XXXX = 2110). Insérez alors la ligne de sauvegarde de page écran entre le numéro de ligne affiché et le précédent ; sur notre logiciel :

2105 SAVE « PAGEECRAN.BIN », B,&C000,&4000

Attention, ne faites pas d'erreur, les touches et flèche de retour arrière ne sont plus disponibles ; vous validerez par la touche <ENTER> (et non <RETURN> qui n'est plus active). Frappez ensuite **GOTO XXXX** <ENTER> (dans notre cas : GOTO 2110) et le logiciel est prêt à sauvegarder la page minitel que vous aurez devant les yeux lors de la déconnexion. Vous pouvez aussi insérer les lignes suivantes, si vous voulez sauvegarder plusieurs pages minitel avant relecture : (programme sur page 5).

<u>TOUCHES MINITEL</u>	<u>TOUCHES AMSTRAD</u>
ENVOI	RETURN
SOMMAIRE	↑
RETOUR	←
SUITE	→
REPETITION	↓
ANNULATION	CLR
CORRECTION	DEL
GUIDE	TAB
CONNEXION / FIN	COPY
<u>FONCTION SUPPLEMENTAIRE</u>	
FIN DE COMMUNICATION	CONTROL

Tables de correspondance. Touches CPC/Fonction minitel

```
2102 SAVE "TEMPORAI.BIN",B,&C000,&4000
2104 MODE 2
2106 INPUT "NOM DE LA PAGE (8 LETTRES), PUIS <ENTER>";A$
2108 LOAD "TEMPORAI.BIN"
2109 SAVE A$,B,&C000,&4000
```

Les lignes 2108 et 2109 auraient pu être remplacées par l'instruction `ÛREN, A$, « TEMPORAI.BIN »` dans un autre programme, mais ce logiciel redéfinissant les touches, la fonction `ÛREN` n'est plus disponible.

Pour une utilisation ultérieure du logiciel ainsi modifié, vous pouvez le sauvegarder sous le nom « DTL » par exemple (de préférence sur disquette — sinon il faudra le sauvegarder sur une cassette vierge, puis sauvegarder dans l'ordre les logiciels suivants de la cassette originale).

LECTURE DES PAGES ENREGISTREES

La lecture des pages enregistrées grâce au programme modifié, peut être effectuée par l'instruction :

LOAD « NOMPAGE.BIN »

Le petit programme suivant vous permettra de relire une parmi plusieurs pages enregistrées sur une disquette, et vous permettra d'en faire une copie sur imprimante :

```
10 MODE 1
20 INPUT "NOM DE LA PAGE A LIRE";A$
30 LOAD A$:REM CHARGEMENT DE LA PAGE
40 A$ = INKEYS:REM ATTENTE TOUCHE SANS AFFICHAGE
50 IF A$ = "1" THEN 80
60 IF A$ = "2" THEN 10
70 GOTO 40
80 REM ***** PLACEZ ICI LA COPIE D'ECRAN *****
```

Vous pouvez placer après la ligne 80 la copie d'écran graphique proposée en Partie 9 chapitre 8.1. Si vous ne désirez imprimer que le texte de la page minitel (numéros de téléphone, relevé de compte bancaire,...) nous vous proposons le programme de copie page suivante, qui a l'avantage d'être beaucoup plus rapide car il fonctionne en mode texte.

```
80 PRINT#8
90 FOR I = 1 TO 25:REM LECTURE PAR LIGNE
100 FOR J = 1 TO 40:REM LECTURE PAR COLONNE
110 LOCATE J,I
120 AS$ = COPYCHR$(#0):REM LECTURE CARACTERE SUR ECRAN
130 IF AS$ = "" THEN AS$ = " "
140 IF ASC(AS$) > 127 THEN AS$ = " "
150 PRINT#8,AS$;
160 NEXT J
170 PRINT#8
180 NEXT I
190 GOTO 10
```

La ligne 140 élimine tout caractère graphique lu à l'écran, et la ligne 130 évite l'éventuelle erreur « TYPE MISMATCH ».

III. Programmation avancée

La programmation du Modem DTL2000+ requiert une connaissance de la structure interne du modem, sur laquelle le manuel est assez discret. Trois fonctions principales permettent de réaliser la communication de données grâce au DTL2000+ : la modulation/démodulation des données à transmettre, la mémorisation des caractéristiques de modulation, et la conversion des données parallèle/série.

Les structures réalisant ces fonctions sont respectivement : le MODEM, un P.I.A. (port d'entrées/sorties parallèles, le composant EF 6821 de chez Motorola), et un A.C.I.A. (le composant EF 6850 de chez Motorola). L'organisation du système vous permettant de communiquer est donnée sur la figure suivante, l'unité centrale étant composée par votre CPC. Nous laisserons de côté le composant MODEM, sa programmation s'effectuant par l'intermédiaire du PIA 6821.

LE PIA 6821 (PIA : *Peripheral interface Adaptor*)

Ce composant se trouve placé aux adresses hexadécimales &F8F8 et &F8F9 par un décodage approprié. Ces adresses correspondent respectivement à deux registres A et B qui sont les registres de données (DDR/OR) et de contrôle (CR) du PIA.

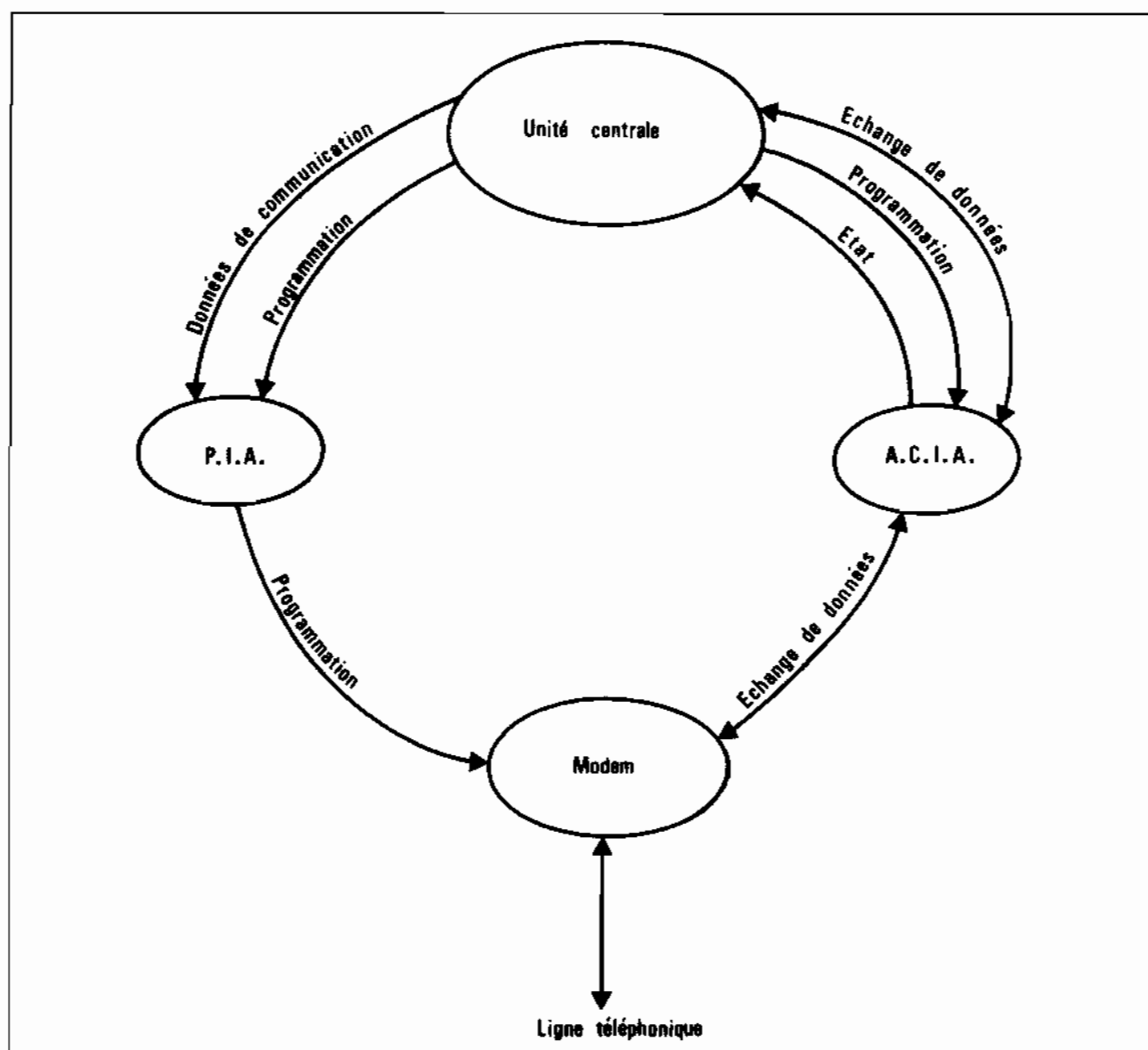


Schéma fonctionnel de la transmission par modem

Le registre de contrôle

Ce registre permet de valider diverses caractéristiques de fonctionnement du PIA. Il est composé de huit bits dont seul le bit CR2 est utilisé.

CR7 CR6 CR5 CR4 CR3 CR2 CR1 CR0 B

Ce bit permet lorsqu'il est à 0 de choisir le registre DDR, et lorsqu'il est à 1 de choisir le registre OR (nous verrons ci-dessous l'explication de ces deux registres). Tous les autres bits seront placés à 0

OUT &F8F9,&00 → choix de DDR

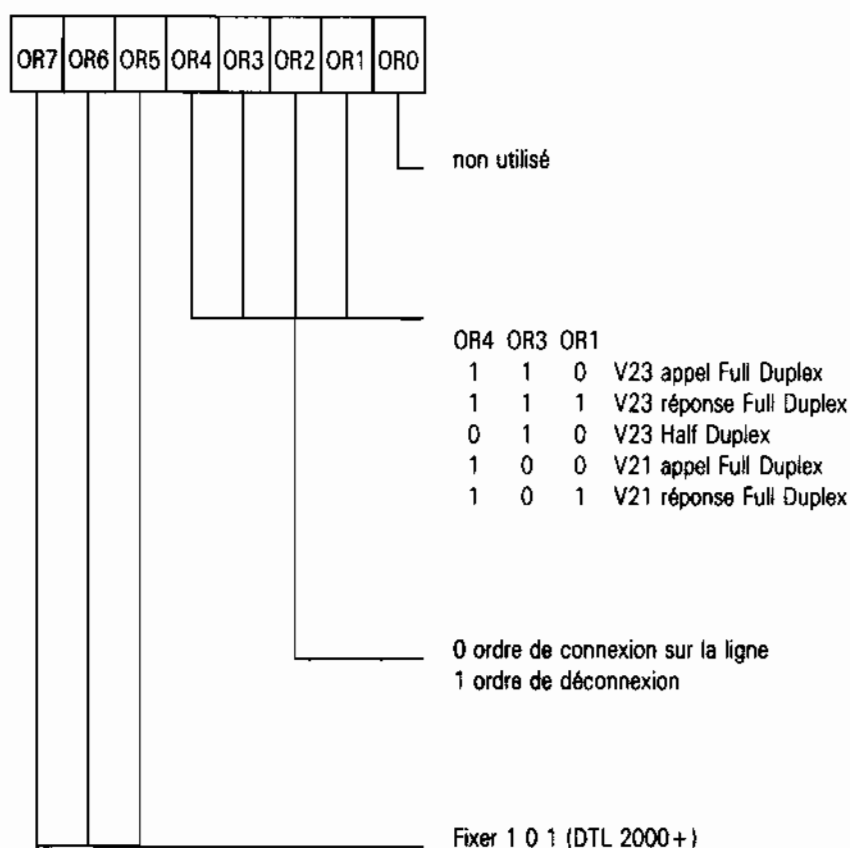
OUT &F8F9,&04 → choix de OR

Le registre double DDR/OR

Le registre placé à l'adresse &F8F8 est un registre double, dont le choix est conditionné par le registre CR comme indiqué ci-dessus.

La première opération à effectuer est de choisir le DDR qui est le registre de direction de données. Ce registre est composé de huit bits (DDR7 à DDR0) qui indiquent le sens du transfert des données. Par exemple si le bit DDR7 est à 1 et le bit DDR5 à 0, le PIA pourra émettre des données sur la ligne numéro 7 (on dira qu'elle est en sortie) et en recevoir sur la ligne numéro 5 (on dira qu'elle est en entrée). La programmation des directions pour le DTL2000+ est : toutes les lignes en sorties sauf la ligne 0 en entrée ; donc on écrira dans DDR la valeur &FE par l'instruction : **OUT &F8F8,&FE**

La deuxième opération est de sélectionner le registre OR qui permet d'écrire ou de lire des données sur le PIA (comme indiqué pour le registre de contrôle). Ce registre est un registre huit bits dont les significations suivantes sont données.



Registre de données du PIA

Si nous voulons émuler un minitel connecté sur la ligne téléphonique nous programmerons : **OUT &F8F8,&B8**

Il faut remarquer que lorsqu'il y a ordre de connexion sur la ligne, le modem attend 45 secondes, puis il se déconnecte automatiquement en cas de perte de porteur du modem distant. Pour une nouvelle connexion, il faut mettre OR2 à 1 puis de nouveau à 0.

L'ACIA 6850 (*Asynchronous communication interface adaptor*)

Ce composant se trouve placé aux adresses hexadécimales &F8FC et &F8FD (registres C et D). Le registre C est le registre de contrôle et d'état des données et le registre D, celui de transmission.

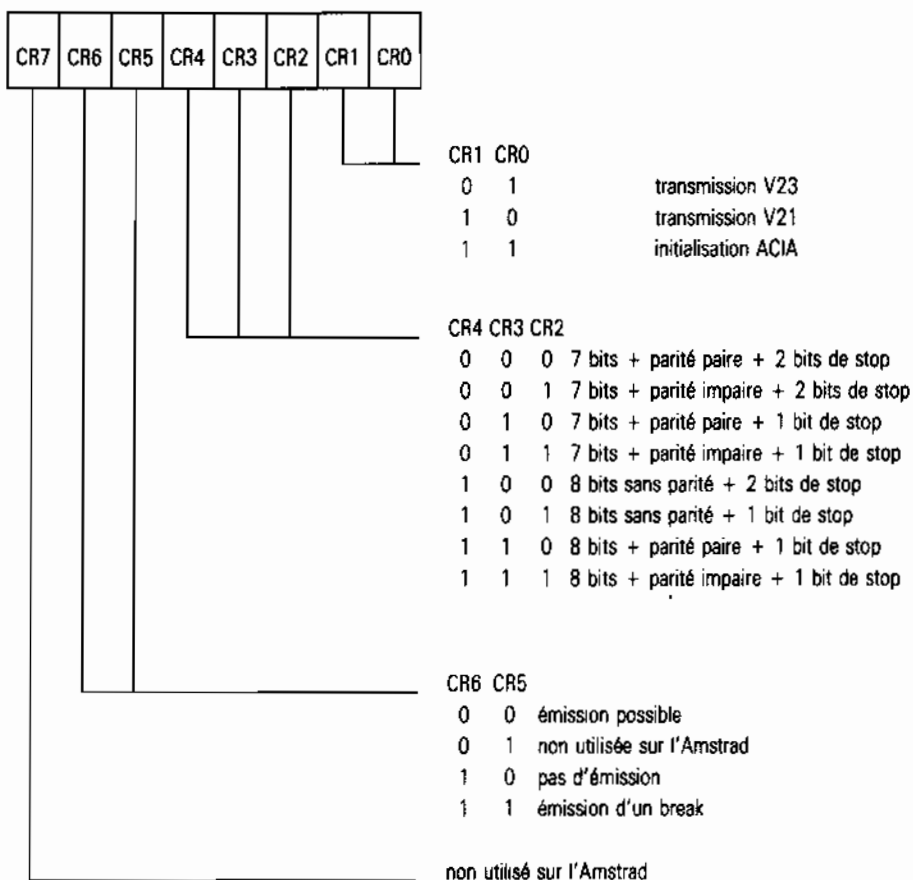
Le registre de contrôle et d'état

Ce registre est un registre double qui peut être lu ou écrit.

En écriture, il permet de programmer les caractéristiques de transfert des données vers le composant Modem — il est nommé registre de contrôle.

En lecture, il permet de connaître l'état de l'ACIA (porteur présente, donnée reçue...) — il est nommé registre d'état.

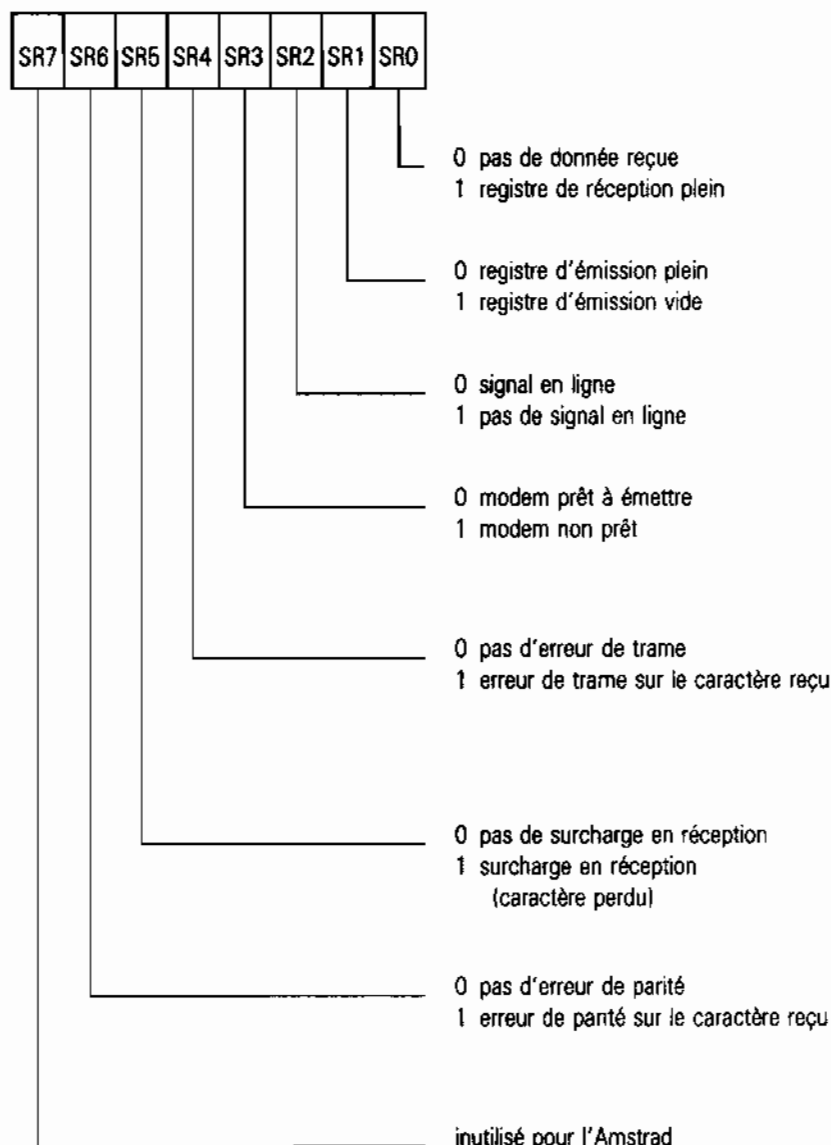
a) Le registre de contrôle (en écriture)



Programmation du registre de contrôle de l'ACIA

Lors de la programmation de l'ACIA, une phase d'initialisation doit toujours être prévue avant toute modification par l'instruction : **OUT &F8FC,&03**

b) Le registre d'état (en lecture)



Signification des bits du registre d'état de l'ACIA

Ce registre permet par lecture de connaître l'état de la ligne ou du registre de transmission (donnée reçue, registre vide pour un nouveau caractère, erreur sur un caractère,...)

Le registre de transmission

Ce registre permet, lorsque toutes les phases de programmation précédentes sont correctes, de communiquer sur la ligne. Une donnée sera émise lorsque l'on y effectuera une écriture, ou pourra être lue par lecture de ce même registre.

OUT &F8FD,&41 → Emet le code ASCII du caractère A

X = INP(&F8FD) Place dans la variable X la valeur hexadécimale reçue.

ORGANIGRAMMES ET PROGRAMMES D'UTILISATIONS

Initialisation

Avant toute programmation du modem DTL2000+ , une initialisation est nécessaire et sera la même quelle que soit l'utilisation sur la ligne :

```
10 OUT &F8F9,&00:REM SELECTION DDR(PIA)
20 OUT &F8F8,&FE:REM OR1 A OR7 EN SORTIE
30 OUT &F8F9,&04:REM SELECTION OR
40 OUT &F8F8,&00:REM INITIALISATION MODEM
50 OUT &F8FC,&03:REM INITIALISATION ACIA
60 OUT &F8FC,&40:REM PAS D'EMISSION
```

Par expérience, il est parfois nécessaire d'ajouter entre chacune des instructions une boucle d'attente du genre

FOR I = 1 TO 1:NEXT I

qui laissera au modem DTL2000+ le temps de « digérer » les valeurs envoyées.

Emulation d'un minitel

Suite à l'initialisation précédente, le DTL2000+ peut être programmé pour émuler un minitel, par exemple. En se référant à la figure page 8, le minitel fonctionnant en mode V23 appel Full Duplex, nous aurons donc pour le registre A, les valeurs OR4 OR3 OR1 = 110.

Si nous supposons que le minitel ne doit pas être connecté de suite, nous aurons OR2 = 1, l'octet à écrire dans A sera : 10111100 = &BC.

En se référant au registre de contrôle de la page 9, il reste l'ACIA à configurer. Nous aurons CR1 CR0 = 01 (mode V23) ; CR4 CR3 CR2 = 010

(le minitel communique au format 7 bits, parité paire et 1 bit de stop), et CR6 CR5 = 10 (pas d'émission) ; ce qui donne 01001001 = &49 à écrire dans le registre de contrôle de l'ACIA.

Vous pourrez composer le numéro du service télématique sur le téléphone et, lorsque la porteuse se fera entendre, il suffira de mettre le bit OR2 à 0 (ce qui donnera 10111000 = &B8 dans A) et le bit CR6 du registre de contrôle de l'ACIA à 0 (00001001 = &09 dans C) :

```

70 OUT &F8F8,&BC:REM CONFIGURATION MODEM

80 OUT &F8FC,&49:REM CONFIGURATION ACIA

90 REM *** ATTENTE DE PORTEUSE ***

100 OUT &F8F8,&B8:REM CONNEXION A LA LIGNE

110 OUT &F8FC,&09:REM ORDRE D'ENVOYER LA PORTEUSE

```

Comme précédemment, il sera utile d'insérer des boucles de temporisation, entre chaque instruction.

DISCUSSION PAR AMSTRAD INTERPOSE

Le programme que nous vous proposons de réaliser permettra, grâce à la même configuration matérielle (un Amstrad, un modem DTL2000+ et une ligne téléphonique pour chacun des partenaires), de discuter l'un et l'autre grâce au clavier et deux fenêtres sur l'écran de la visu.

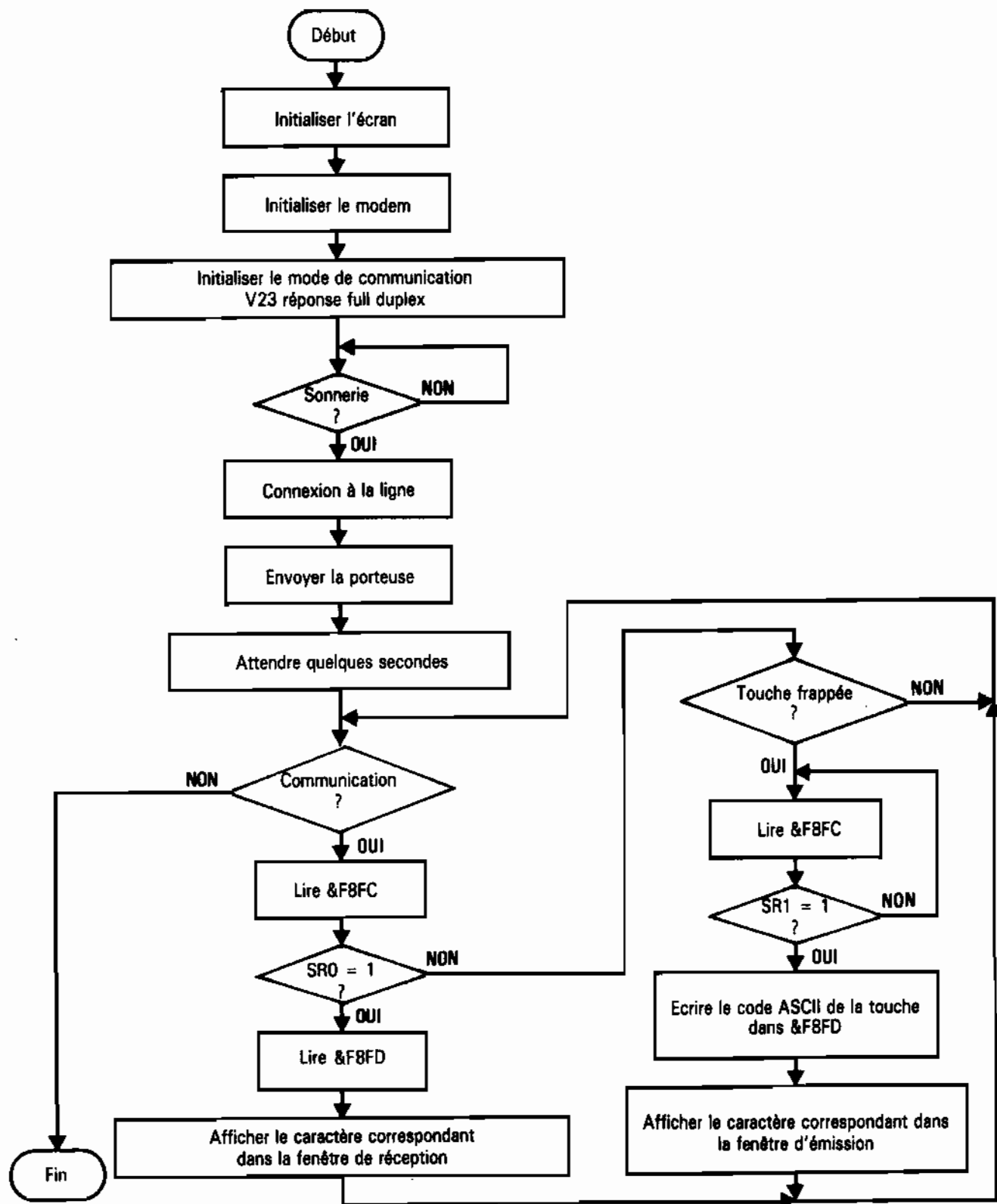
L'algorithme gérant la procédure de communication vous est donné page suivante.

Cet organigramme est valable pour l'un des interlocuteurs (que nous appellerons A) et demandera quelques modifications pour le deuxième (B). Nous allons l'analyser pour vous permettre de le modifier afin de le rendre opérationnel de chaque côté de la ligne.

Une première initialisation concernera l'écran de l'Amstrad pour une présentation sur deux fenêtres de chacun des partenaires, vient ensuite la communication effective.

Tout d'abord, la phase d'initialisation du modem sera la même.

Ensuite nous trouvons une phase d'initialisation du mode de communication : nous avons choisi de programmer le modem de l'interlocuteur A en Mode V23 réponse Full Duplex, B devra donc programmer le sien en mode V23 appel Full Duplex. Vient ensuite un test de sonnerie : ce test pourra s'appliquer sur le port joystick si l'on y a connecté le détecteur de sonnerie, ou pourra être manuel par la détection d'appui sur une touche quelconque du clavier. La réponse à ce test sera oui pour l'interlocuteur A dès que la sonnerie se fera entendre (ou dès que l'accord entre A et B aura été établi oralement), B ne répondra oui à ce test (appui



L'Algorithme

sur une touche) que lorsqu'il entendra un sifflement aigu dans le combiné, indiquant que la porteuse de A est présente sur la ligne. Suite à la détection de sonnerie, A se connecte à la ligne puis envoie sa porteuse. Il attend quelques secondes que B réagisse et fasse de même.

Le test suivant est là pour permettre à l'algorithme d'avoir une fin et aura pour réponse NON si la communication est rompue (par programmation, manuellement ou par arrêt de l'un des ordinateurs).

Le reste de l'algorithme est ensuite commun pour les deux partenaires. Il faut d'abord effectuer une lecture du registre d'état de l'ACIA en &F8FC. Un test sur le bit 0 permettra de savoir si un caractère a été reçu. Si c'est le cas, il faut lire le registre de réception de l'ACIA (&F8FD), le caractère reçu sera ensuite affiché dans la zone écran prévue à cet effet.

Si aucun caractère n'a été reçu, on testera si une touche a été enfoncée. Si NON, le cycle recommence. Si OUI, on relira le registre d'état de l'ACIA afin de tester le bit 1. Tant que ce bit n'est pas à 1, un caractère est encore en instance d'être envoyé, il faut donc relire et tester à nouveau ce bit, jusqu'à ce qu'il soit à 1 (le registre d'émission sera alors vide). On pourra alors écrire le code ASCII du caractère dans &F8FD et l'afficher à l'écran dans la zone d'émission. On bouclera ensuite pour la suite de la communication.

Ce programme peut être écrit en langage Basic sans problème particulier. Les deux fenêtres seront créées par l'instruction **WINDOW #...** Les instructions de lectures sont :

X = INP (&F8FC)

pour lire dans la variable X le contenu du registre d'état de l'ACIA, est :

Y = INP (&F8FD)

pour lire dans Y le code ASCII d'un caractère reçu. En écriture, l'instruction d'envoi d'un caractère est :

OUT &F8FD,Z

avec Z étant le code ASCII du caractère à émettre. Le test $SR0 = 1$ se traduit par :

**IF (X AND 1) = 1 THEN GOTO (réponse OUI)
ELSE GOTO (réponse NON)**

Le test $SR1 = 1$ sera effectué par :

**IF (X AND 2) = 2 THEN GOTO (réponse OUI)
ELSE GOTO (réponse NON)**

Vous êtes ainsi en possession de toutes les informations nécessaires pour écrire ce programme. Nous vous conseillons tout de même d'effectuer les essais étape par étape et de réfléchir posément en cas de problème. La première étape concernera par exemple un essai de connexion, la deuxième consistera à recevoir un caractère et l'afficher, etc. Cette façon de procéder vous permettra de mieux comprendre le fonctionnement de votre modem et de détecter plus facilement les erreurs éventuelles.

Un dernier conseil : lorsque vous travaillerez avec un des registres (&F8F8, &F8F9, &F8FC), tâchez de transcrire sur papier les valeurs à y écrire en binaire et reportez-vous aux explications du paragraphe 3 pour anticiper le fonctionnement.

ECHANGE DE PROGRAMMES

L'échange de programmes a été étudié lors de l'utilisation de la RS232 Amstrad conjointement au modem du minitel (voir partie 8/5.2.3). Cet échange ne pouvait s'effectuer que sur 7 bits car le minitel ne communique que sous ce format. Le modem DTL2000+ peut lui communiquer en 8 bits, ce qui nous permettra d'envoyer n'importe quel programme ou fichier en mémoire centrale possédant des codes de contrôle et un jeu de caractères étendus sur 8 bits (un fichier CPM, un programme Basic en RAM).

Nous vous proposons l'algorithme de la page 17 que votre perspicacité traduira en programme.

Ici encore, l'un des interlocuteurs sera en mode Appel et l'autre en mode Réponse.

Afin d'améliorer les performances, vous pourrez écrire ce programme en assembleur, mais le Basic Amstrad traitera correctement la communication.

Vous pourrez modifier cet algorithme pour envoyer un fichier contenu sur disquette comme donné par exemple dans ces algorithmes d'émission et de réception :

EMISSION

DEBUT

- Initialiser le Modem
- Initialiser les variables
- Etablir la communication
- Ordonner l'émission
- Ouvrir le fichier programme en lecture
- TANTQUE la fin du fichier n'est pas atteinte
 - Lire un enregistrement

- CO

L'incréméntation s'effectue
automatiquement

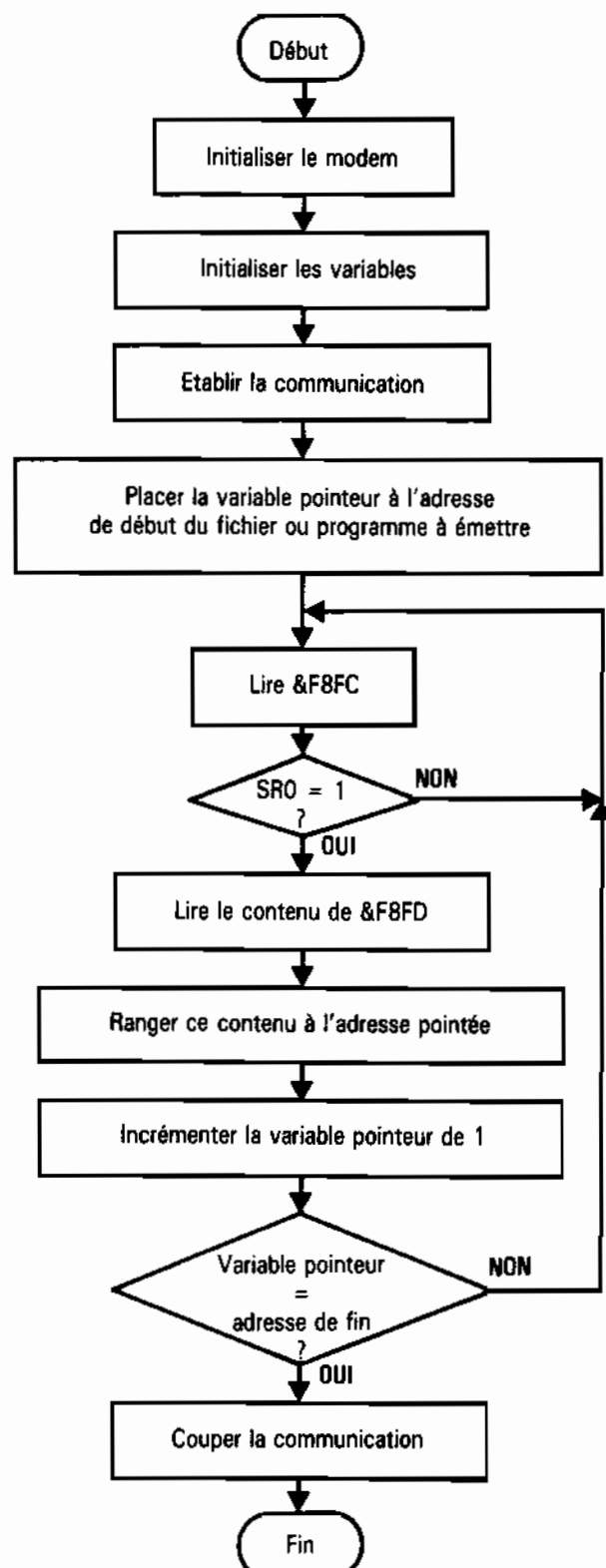
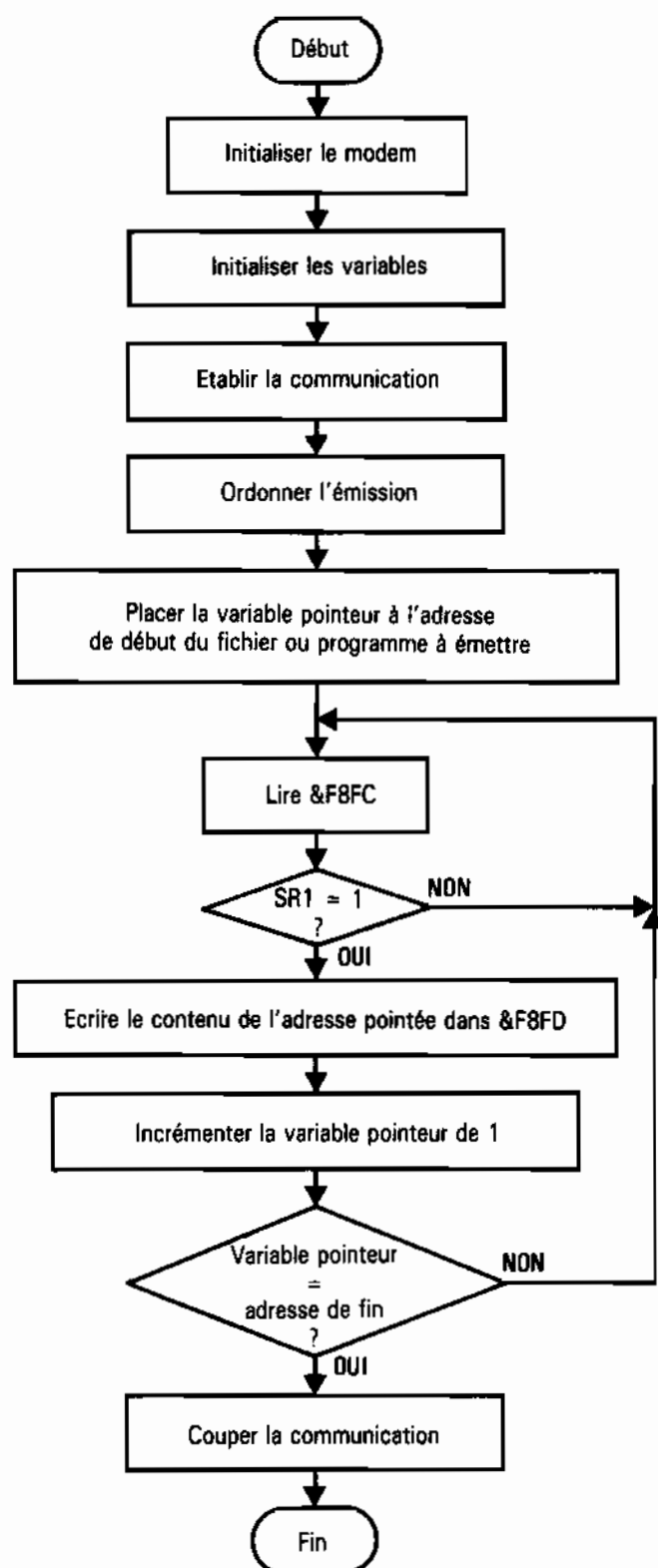
- FINCO

- POUR I allant DE 1 A la fin de l'enregistrement
 - Prendre le caractère de la 1^{ème} position
 - Traduire ce caractère en ASCII
 - REPETER
 - Lire le registre en &F8FC
 - JUSQU'A ce que le bit SR1 soit à 1
 - Ecrire la valeur ASCII précédente dans le registre d'émission &F8FD
 - FINPOUR
 - FIN TANTQUE
 - Fermer le fichier
 - Romp la communication
- FIN

RECEPTION

DEBUT

- Initialiser le Modem
 - Initialiser les variables
 - Etablir la communication
 - Ouvrir le fichier programme en écriture
 - TANTQUE la communication n'est pas rompue
 - REPETER
 - Lire le registre en &F8FC
 - JUSQU'A ce que le bit SR soit à 1
 - Lire la valeur recue dans le registre d'émission &F8FD
 - Ecrire le caractère correspondant dans le fichier programme
 - FIN TANTQUE
 - Fermer le fichier
- FIN



8/5.4

Réalisation d'un serveur télématique

Qu'est-ce qu'un serveur télématique ?

Dans ce chapitre, nous vous proposons de créer un serveur Minitel. Le serveur télématique qui nous intéresse est composé d'une structure matérielle et logicielle proposant un service par l'intermédiaire du Minitel. Le service peut être de n'importe quelle nature : téléchargement de programmes, cours d'informatique, banques de données divers, discussions simultanées entre plusieurs utilisateurs, liste des nouveaux films et critique, etc.

La *structure matérielle* devra comporter une unité mémoire contenant les données du ou des services à rendre, le logiciel permettant de restituer ces données à l'utilisateur, et une unité d'adaptation des données à la ligne téléphonique et au Minitel.

La *structure logicielle* sera constituée d'adaptations des différentes données aux normes vidéotex (norme du Minitel), et proposera un enchaînement des différents services, ainsi que la gestion de ces services (mémorisation de messages, gestion de jeux, ..., etc.).

8/5.4.1

La structure matérielle

La structure matérielle que nous utiliserons se composera d'un CPC 6128, dont l'unité de disquette servira d'unité mémoire des différents services et données à restituer à un éventuel utilisateur. Vous pourrez adapter

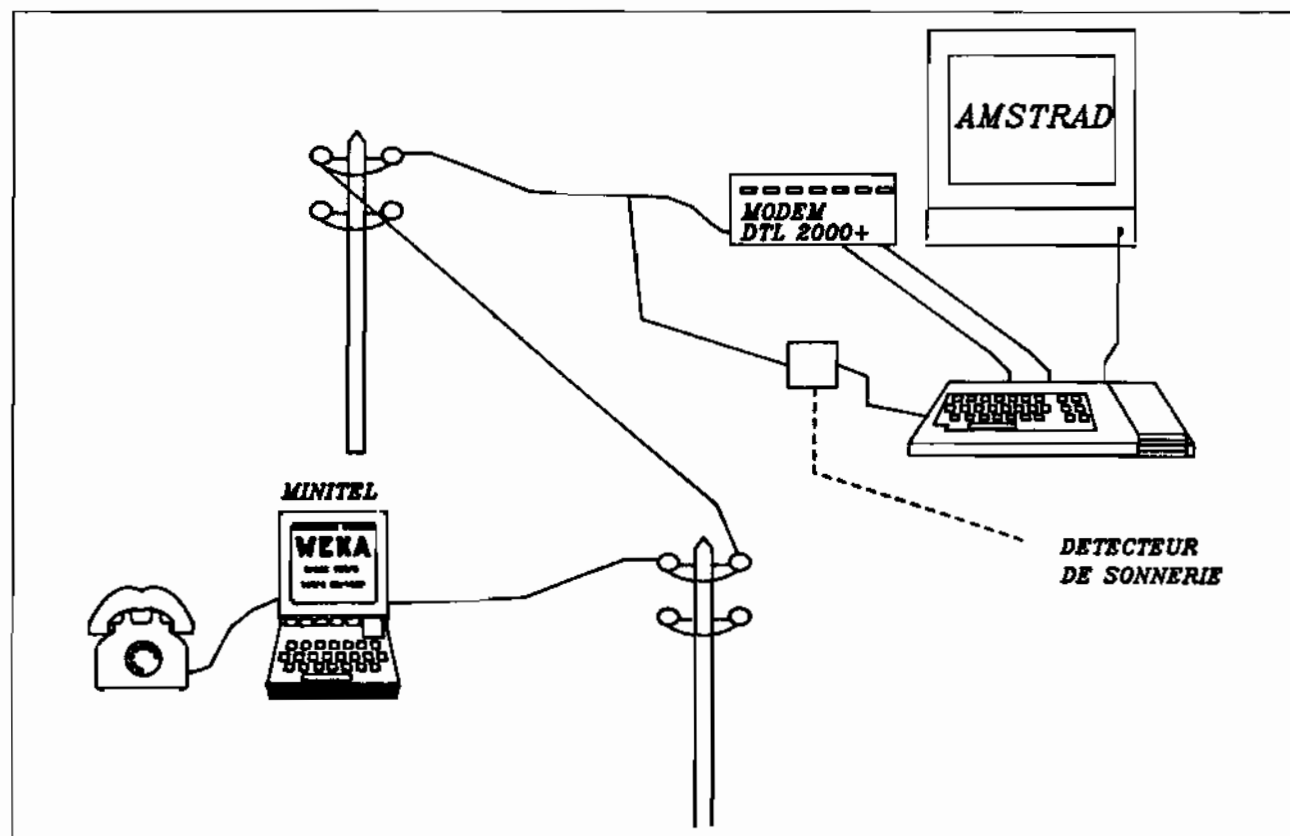
le logiciel sans problème à un CPC 664 ou CPC 464 muni d'un lecteur de disquette. Le programme de gestion du serveur sera en mémoire centrale.

Nous aurons aussi besoin d'un modem afin de transmettre les données par la ligne téléphonique : le modem utilisé est le modem Digitelec DTL 2000 plus, qui se connecte directement sur le connecteur d'extension 2 x 50 broches de votre CPC, et possède sa propre interface série, évitant le recours à une RS-232.

Nous utiliserons, pour commander la connexion du modem à la ligne, le détecteur de sonnerie décrit Partie 10 chapitre 4.3, mais vous pourrez effectuer cette connexion manuellement (si vous décidez de ne mettre votre serveur à disposition que peu de temps chaque jour), en utilisant un joystick que vous actionnerez lorsque vous entendrez la sonnerie du téléphone.

Vous aurez bien entendu besoin d'une ligne téléphonique, réservée totalement ou non à votre serveur (vous pouvez décider d'une plage horaire quotidienne durant laquelle vos amis pourront profiter de votre serveur).

La structure matérielle complète est représentée ci-dessous avec, connecté à distance, un abonné qui visualisera les informations proposées sur son propre Minitel.



La structure matérielle.

8/5.4.2

La structure logicielle

I - Conseils de base

Le Minitel est un outil de communication relativement performant et peu onéreux lorsqu'il est utilisé intelligemment tant pour les loisirs que pour les renseignements utiles. Il possède, de plus, un attrait mystérieux qui en fait plus qu'un objet : nombre d'utilisateurs ont l'impression de dialoguer avec un objet vivant.

La qualité de votre service devra préserver cet aspect par une présentation graphique agréable : animation, utilisation de couleurs (qui se traduiront par des niveaux de gris sur un Minitel noir et blanc).

Le Minitel présente par contre le défaut d'être lent. Les écrans ne devront pas durer plus de quelques secondes sous peine de lasser l'utilisateur qui ne consultera plus votre serveur. Une seule page pourra prendre quelque temps, mais elle ne devra pas réapparaître lors de la suite de la consultation (ce sera la page de présentation du serveur). L'utilisation abusive des couleurs et des caractères graphiques allonge considérablement les temps d'affichage, aussi ne les utilisera-t-on qu'à bon escient.

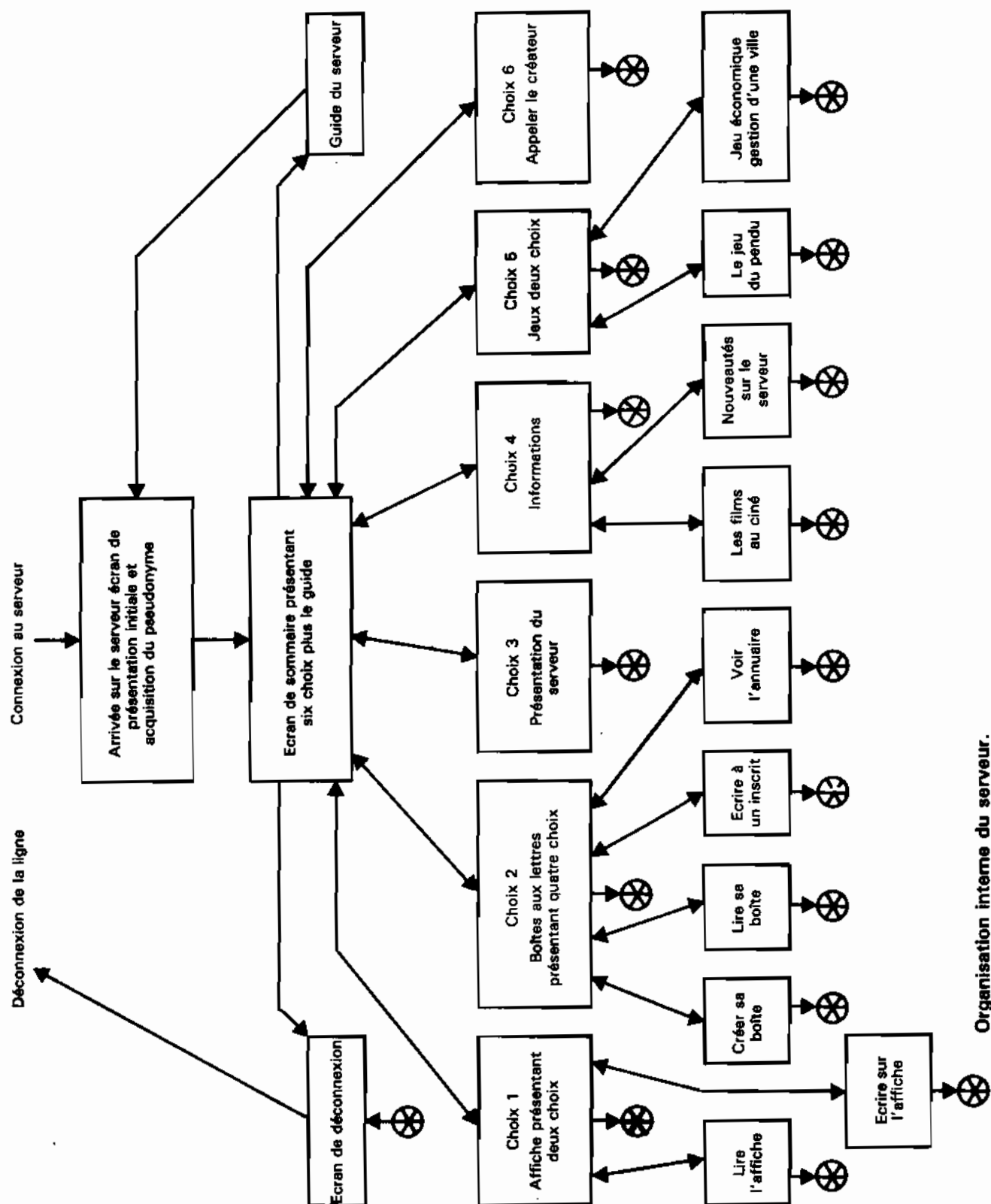
Beaucoup d'utilisateurs risquent de se lasser si vos pages de présentation ne sont pas renouvelées de temps à autre. Vous pourrez même ajouter de nouveaux services et éliminer ceux qui ne sont plus ou peu consultés.

En résumé : rapidité, originalité et nouveautés sont appréciées par les utilisateurs du Minitel. Vous pourrez modifier à votre gré le logiciel que nous vous proposons pour le rendre plus personnel et attractif.

II - Organisation du serveur

La structure logicielle d'un serveur dépend avant toute chose de son organisation. Nous choisirons une organisation de type arborescente, qui s'adapte très bien aux structures de choix disponibles dans un service.

Présentée simplement, l'arborescence est une structure logicielle ressemblant à un arbre (l'arbre généalogique en est un exemple). Lorsque l'on se trouve à un certain niveau de l'arbre, nous pouvons atteindre un autre niveau situé plus bas si le choix en est permis, ou remonter au niveau précédent (celui d'où l'on vient), sauf si l'on se trouvait au niveau initial bien sûr.



Organisation interne du serveur.

L'arborescence se remarque à partir de l'écran de sommaire qui permet de descendre vers six niveaux différents. A partir de certains de ces niveaux, nous pouvons descendre vers un niveau inférieur ou remonter au sommaire.

Une seule exception à l'arborescence, la déconnexion du serveur (représentée par une étoile entourée d'un cercle qu'il faut traiter comme un choix à part et qui mène à l'attente d'une nouvelle connexion — le serveur est en veille).

Toute création d'un serveur commencera donc par la définition des services proposés puis par leur agencement.

L'écran de présentation

Nous trouvons d'abord dans cet exemple la connexion au serveur qui interviendra dès le premier coup de sonnerie reçu.

Il sera proposé au correspondant un écran de présentation qui pourra être animé au mieux pour retenir son attention. Cet écran n'apparaîtra qu'une fois dans la consultation et pourra être renouvelé régulièrement : c'est lui qui présente votre serveur, donc, qui le met en valeur. Le nom ou pseudonyme de votre correspondant pourra éventuellement être demandé pour le reconnaître dans la suite de la consultation.

L'écran de sommaire

C'est l'écran par lequel l'utilisateur devra passer souvent pour se « promener » dans le serveur. Nous proposons ici six choix, mais vous pourrez modifier ce nombre selon votre imagination dans les services proposés. Il est à remarquer que plus il y aura de choix, plus la page sera longue à afficher, et plus l'utilisateur devra réfléchir à la signification de ceux-ci. Un maximum de neuf choix est conseillé. Un choix supplémentaire sera le guide.

Le guide

Votre serveur pourra proposer un guide (mais ce n'est pas obligatoire) qui aiguillera votre correspondant sur certains « trucs » (fonctions spéciales) ou certaines recommandations (utilisation des boîtes aux lettres, particularités de certains jeux, ... etc.).

L'affiche

Nous avons choisi de proposer « deux murs » permettant à chacun de laisser un message de deux lignes ou de lire tous les messages déjà écrits. Le nombre de messages sera limité à seize et le message le plus ancien sera éliminé lorsque l'on écrira un nouveau message. La séparation des différents messages s'effectuera par une différence de couleur de fond.

Les boîtes aux lettres

Ce choix permettra d'utiliser ce que l'on appelle une boîte aux lettres électronique. Vous pourrez limiter le nombre d'utilisateurs de ces boîtes aux lettres en effectuant des restrictions de création (vos meilleurs amis, vos collègues...).

Dans le choix **Créer sa boîte** pourront être demandées les coordonnées exactes de l'utilisateur, que vous recontacterez pour vérification, puis pour qui vous ouvrirez la boîte aux lettres. Il sera nécessaire de limiter le nombre des inscrits et si vous ne possédez qu'un lecteur de disquette ; par contre, si vous possédez un lecteur de disquette supplémentaire de grande capacité (700 Ko), vous pourrez laisser les utilisateurs du Minitel s'inscrire eux-mêmes, vous effacerez périodiquement des boîtes non utilisées depuis un certain temps.

Le choix **Ecrire à un inscrit** permettra de laisser des messages dans une boîte aux lettres ouverte à un nom existant.

Si vous êtes inscrit, vous pourrez lire les messages reçus grâce au choix **Lire sa boîte**. Le nombre de messages dans chaque boîte aux lettres sera limité par exemple à trois messages de 255 caractères (ce qui fait un peu plus de six lignes sur un écran de Minitel M1). Un mot de passe sera nécessaire pour lire une boîte aux lettres afin d'assurer la confidentialité des messages.

L'annuaire permettra de connaître les noms des inscrits en boîte aux lettres.

Présentation du serveur

Vous pourrez grâce à ce choix informer l'utilisateur sur l'organisation matérielle de votre serveur, sur le logiciel, des trucs d'utilisation...

Informations

Dans cette rubrique, vous pourrez par exemple présenter les derniers films à l'affiche du cinéma du quartier ainsi que leur critique. Vous pourrez ajouter une rubrique présentant les dernières nouveautés en matière de service télématique, et notamment sur votre serveur.

Les informations contenues dans cette rubrique devront être facilement modifiables par vous-même, et souvent modifiées pour que ce choix ne soit pas délaissé.

Les jeux

Ce choix même, s'il vous paraît futile et peu intéressant à cause de la lenteur d'affichage d'un écran Minitel (n'espérez pas transposer votre jeu d'arcade favori de votre CPC sur le serveur), se doit d'être présent afin de délasser l'utilisateur et lui montrer votre originalité en matière de jeu et de graphisme.

Cette rubrique comportera ici sur notre serveur deux jeux dont un jeu classique de pendu et un jeu économique vous permettant de gérer la population d'une ville. Vous pourrez au gré de votre imagination proposer autant de jeux que vous le désirez. Nous vous conseillons un jeu nouveau tous les deux mois environ, ainsi que l'épuration de jeux délaissés. Pourquoi ne pas proposer de jeu primé si vous désirez une fréquentation assidue.

Appel du maître

Sur votre serveur, vous serez considéré bien souvent, et surtout par les nouveaux utilisateurs, comme le maître momentané du Minitel par la personne connectée (un nom presque magique et très souvent utilisé : SYSOP qui est la contraction de SYStem OPérateur). Vous ne décevrez pas l'utilisateur si, lorsqu'il entrera dans cette rubrique, vous répondez présent et engagez un dialogue (bien souvent enrichissant : critiques, idées nouvelles, trucs techniques, ...). Vous serez peut-être surpris de constater que cette rubrique sera presque systématiquement utilisée à chaque appel, ce qui rendra votre serveur plus convivial.

Pensez tout de même à laisser un message d'excuse au cas où vous vous absentez.

Ecran de déconnexion

Le logiciel devra détecter la frappe sur la touche **CONNEXION/FIN** du Minitel et présentera une page avant la déconnexion totale qui invitera l'utilisateur à revenir consulter votre serveur.

III - Restrictions d'utilisation

La structure matérielle que nous utilisons pour réaliser notre serveur ne nous permet pas de mémoriser une touche à la volée, aussi il vous faudra prévenir l'utilisateur (dans le *guide* ou la *rubrique* informations) pour qu'il attende la fin de l'affichage d'une page avant d'effectuer son choix, sous peine de ne pas être pris en compte. Cette restriction ne sera pas contraignante dans la mesure où vos pages seront rapides. Elle sera de plus compensée par la validation immédiate du choix (l'utilisateur ne sera pas obligé de frapper la touche **<ENVOI>** de son Minitel).

8/5.4.3

Les routines de base

Avant de nous lancer dans le logiciel de gestion des rubriques, il faut créer des routines de base qui permettront d'initialiser le modem, d'envoyer un caractère, d'émettre une série de caractères placés en DATA dans le programme, d'émettre une chaîne alphanumérique, de recevoir un caractère.

I - L'initialisation

L'initialisation du modem sera effectuée grâce aux informations décrites dans le chapitre traitant du modem Digitelec DTL2000 + . (Partie 8, Chap. 5.3.1).

```
OUT &F8F9,&00:FOR I = 1 TO 10:NEXT I
OUT &F8F8,&FE:FOR I = 1 TO 10:NEXT I
OUT &F8F9,&04:FOR I = 1 TO 10:NEXT I
OUT &F8F8,&00:FOR I = 1 TO 10:NEXT I
OUT &F8FC,&03:FOR I = 1 TO 10:NEXT I
OUT &F8FC,&40:FOR I = 1 TO 10:NEXT I
OUT &F8F8,&BE:FOR I = 1 TO 10:NEXT I
OUT &F8FC,&49:FOR I = 1 TO 10:NEXT I

RETURN
```

La boucle de temporisation FOR I = 1 to 10 : NEXT I a été réécrite à chaque ligne afin que cette routine puisse être implantée à n'importe quel endroit du programme. Nous ne devons donc pas faire appel à un numéro de ligne particulier dans cette même routine, aussi avons-nous été obligés de la répéter, ce qui est concevable sur des routines de faible longueur.

II. La connexion

Lorsqu'un appel sera détecté sur la prise joystick, le logiciel ordonnera au modem de se connecter selon la procédure décrite ci-dessous :

```
110 OUT &F8F8,&BA:FOR I = 1 TO 10:NEXT I
120 OUT &F8FC,&09:FOR I = 1 TO 10:NEXT I
130 RETURN
```

III. Emission d'un caractère

Supposons que nous ayons besoin d'émettre le code ASCII d'un caractère. Nous allons créer une routine qui émettra la valeur contenue dans une variable (nous utiliserons la variable **a**). Il faudra d'abord s'assurer que le registre d'émission est bien vide avant d'émettre la valeur : cette vérification s'effectue par le test du bit 1 du registre de contrôle &F8FC de l'ACIA.

Nous allons de plus ajouter à cette routine un test permettant de détecter une éventuelle déconnexion du Minitel afin de quitter la routine (test sur le bit 2).

Voici la forme algorithmique que prendra cette routine :

DEBUT

TANT QUE l'on ne peut pas émettre de caractère

CO

test du bit 1 de &F8FC

FINCO

SI la communication est rompue

CO

test du bit 2 de &F8FC

FINCO

ALORS Sortir de la routine

FIN SI

FIN TANT QUE

Emettre le caractère

CO

sur le port &F8FD

FINCO

FIN

On se référera avec profit en Partie 8, chapitre 5.2.4.1 page 3 afin d'obtenir quelques explications sur les structures algorithmiques.

On remarquera que, dans la routine basic qui suit, les structures algorithmiques sont quasiment réécrites telles quelles aux instructions basic près (il en aurait été de même avec un autre langage tel que Turbo-Pascal d'où l'intérêt de la forme algorithmique).

```
WHILE ( INP(&F8FC) AND 2 ) <> 2
    IF ( INP(&F8FC) AND 4 ) = 4 THEN RETURN
WEND
OUT &F8FD,a:REM emission du caractère
RETURN
```

IV - Emission d'une série de données

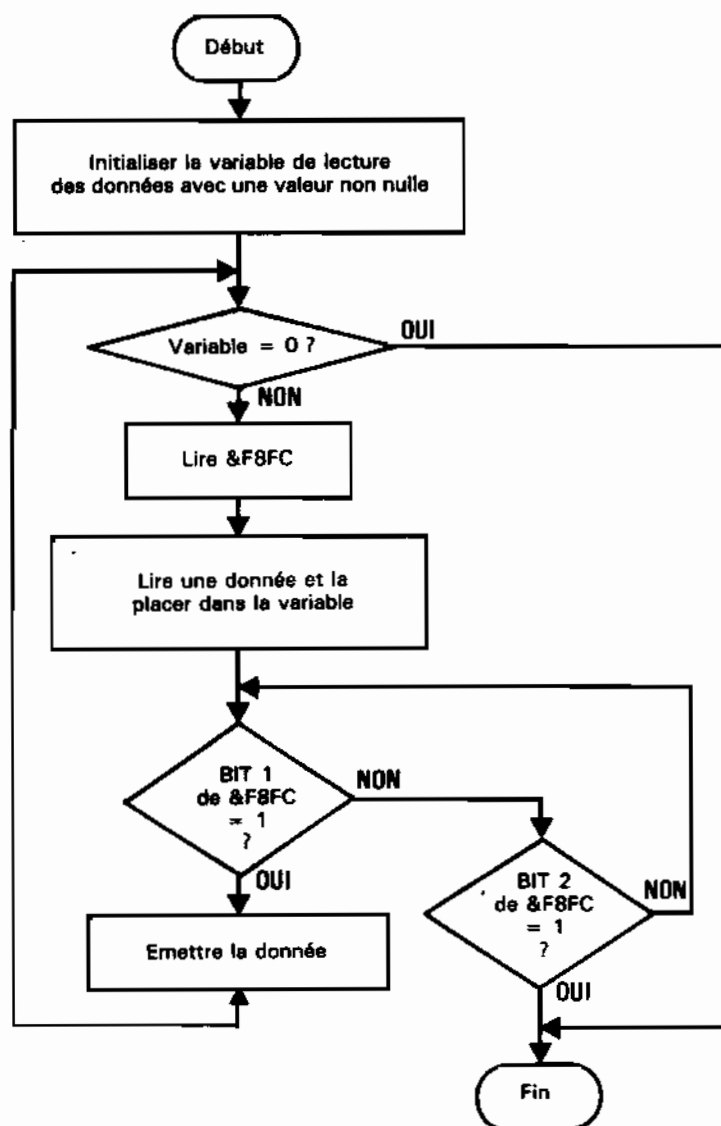
Il nous faut maintenant créer une routine permettant d'émettre une série de données placées en DATA dans le programme. Afin d'éviter un comptage de ces données, et pour permettre des modifications ultérieures plus aisées, nous allons utiliser la valeur 0 pour délimiter la fin de la chaîne de données. Voir l'organigramme de la page 4.

Nous pourrions utiliser le sous-programme précédent pour émettre la donnée. Mais comme dans ce cas le Basic serait trop lent, le programme d'émission de donnée sera donc inclus dans la routine :

```
a = 1
WHILE a <> 0
    READ A
    WHILE ( INP(&F8FC) AND 2 ) <> 2
        IF ( INP(&F8FC) AND 4 ) = 4 THEN RETURN
    WEND
    OUT &F8FD,a
WEND
RETURN
```

N'oubliez pas qu'un bloc de données à émettre par cette routine devra être écrite sous la forme suivante

DATA donnée1,donnée2,donnée3,...,donnée_n,0

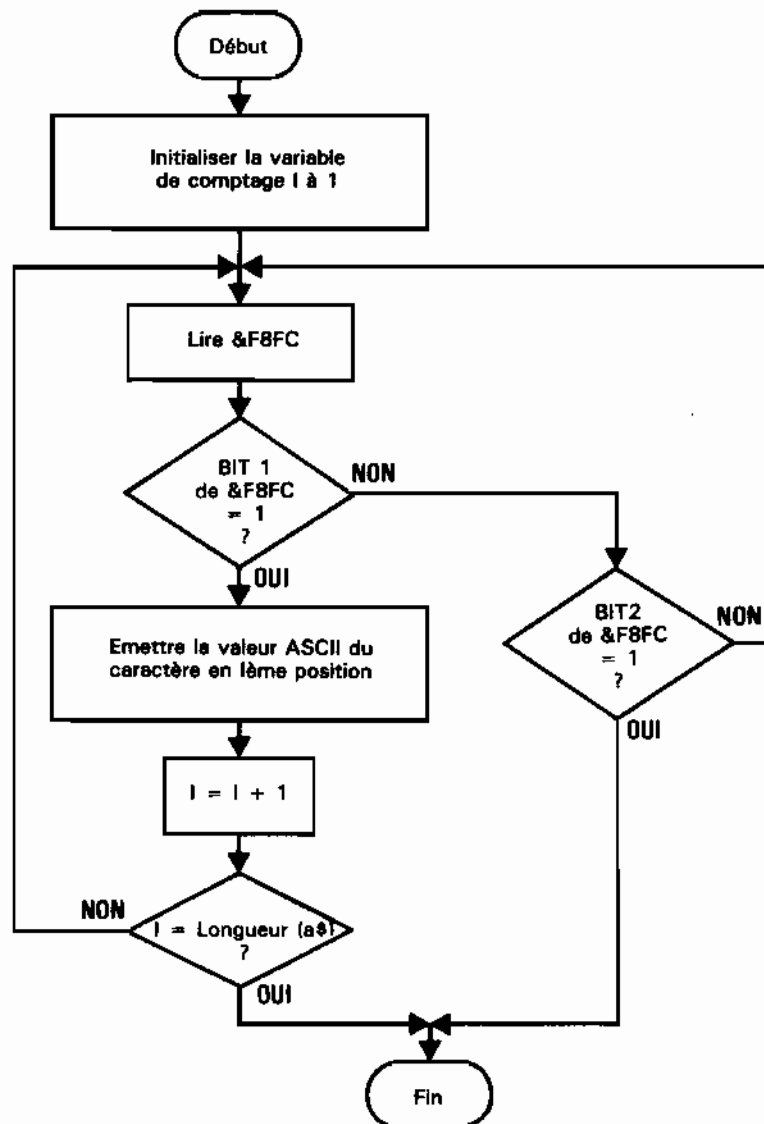


V - Emission d'une chaîne alphanumérique

Il nous sera utile, pour des raisons de place mémoire, d'envoyer des chaînes de caractères alphanumériques plutôt que de placer leurs valeurs ASCII en chaînes de DATA qui seront plus gourmandes en octets.

Afin d'envoyer une telle chaîne, il faudra la placer dans une variable `a$` qui sera décomposée caractère par caractère et dont la valeur ASCII de chacun sera émise tour à tour.

L'organigramme et le programme vous sont donnés ci-dessous :



```

FOR I = 1 TO LEN (a$)
    WHILE ( INP(&F8FC) AND 2 ) <> 2
        IF ( INP(&F8FC) AND 4 ) = 4 THEN RETURN
    WEND
    OUT &F8FD,ASC(MID$(a$,I))
NEXT I
RETURN
  
```

Une fois de plus nous retrouvons le test de déconnexion sur le bit 2 dans la boucle d'émission.

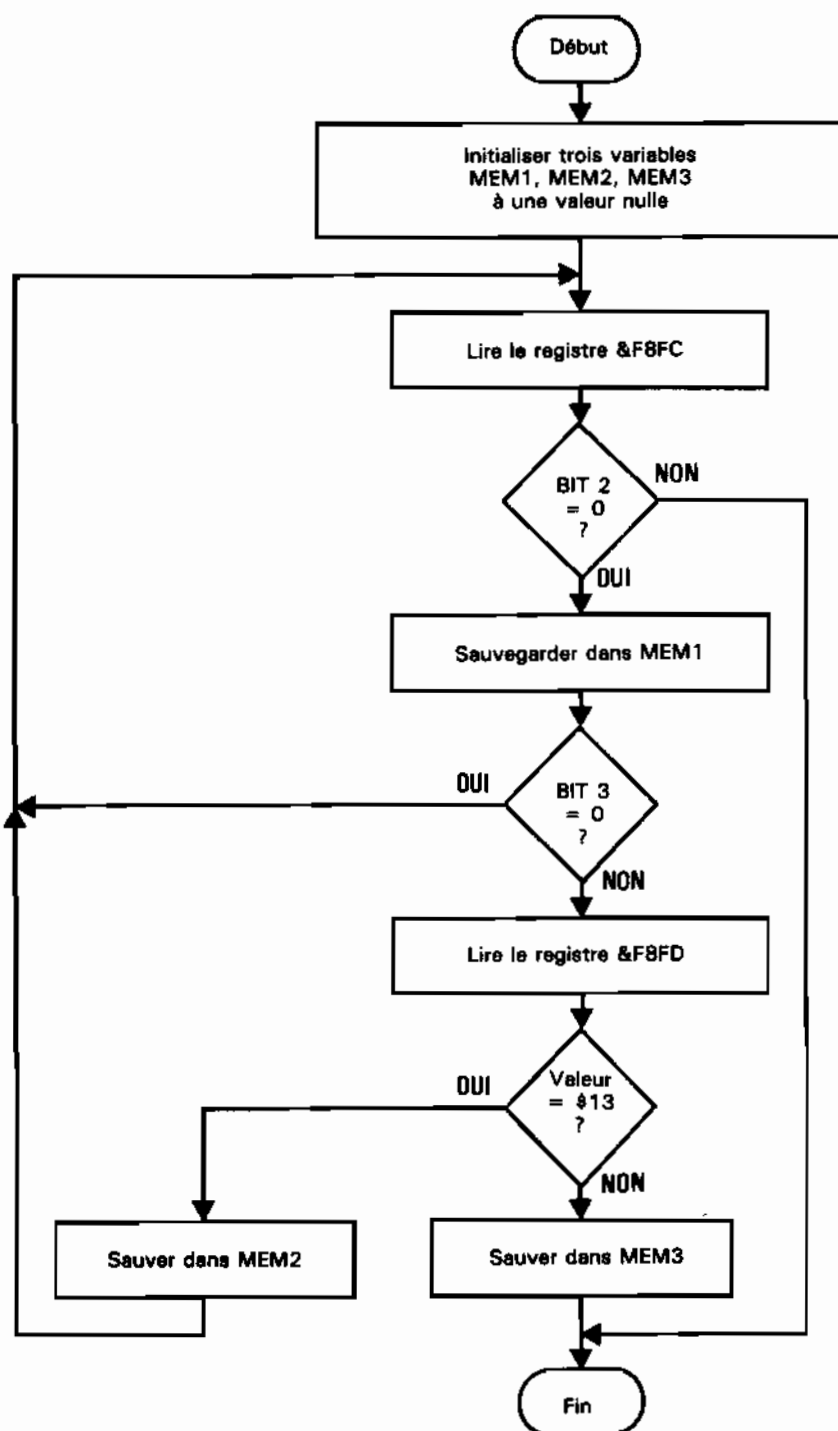
VI - Routine de réception d'un caractère

Avant d'écrire une routine d'acquisition d'un caractère reçu par le modem, il faut déterminer quel type de caractères pourrait être envoyé par le Minitel : soit l'utilisateur frappe une touche alphanumérique, et dans ce cas le caractère reçu est un code ASCII, soit la touche frappée correspond à une touche fonction (<SOMMAIRE>, <SUITE>, <RETOUR>, <GUIDE>, ...) et dans ce cas, le modem reçoit une séquence de deux codes dont le premier est la valeur hexadécimale &13. Il faudra donc tester le code reçu puis, si ce code est égal à &13, attendre l'arrivée du code suivant. Chacune des valeurs reçues sera sauvegardée pour un traitement ultérieur.

Un avantage du modem DTL2000+ est qu'il se comportera comme s'il avait reçu un caractère au cas où le Minitel appelant se déconnecterait, ce qui permettra de sortir de la routine d'attente de caractère.

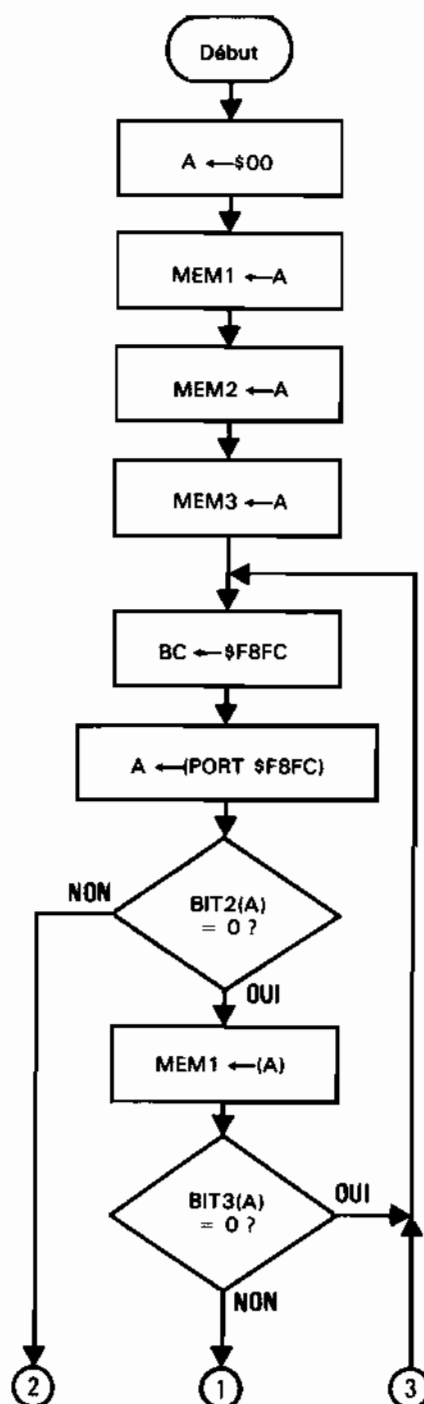
Nous allons écrire le programme Basic correspondant à l'algorithme de la page suivante :

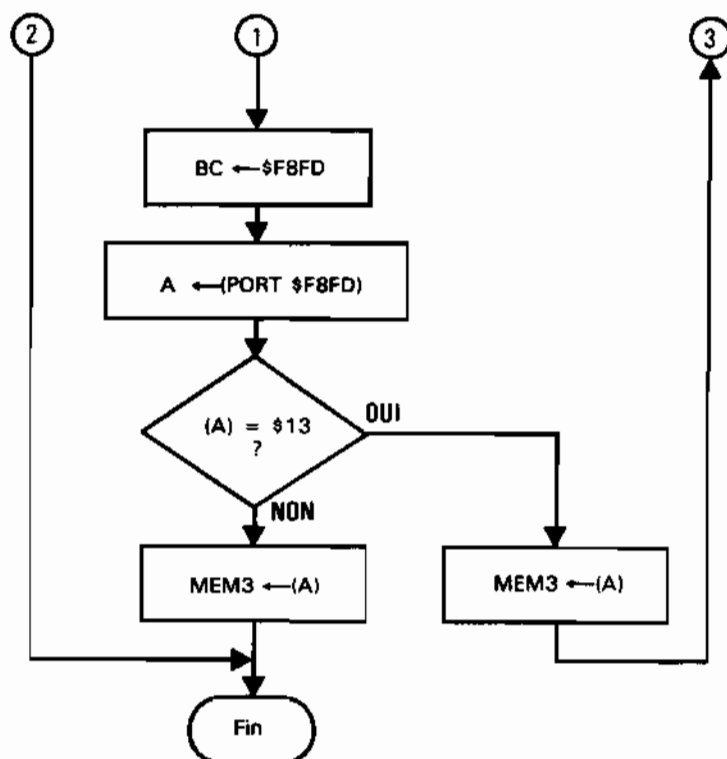
```
10 MEM1 = 0:MEM2 = 0:MEM3 = 0
20 IF ( INP(&F8FC) AND 4 ) = 4 THEN RETURN
30 MEM1 = INP(&F8FC)
40 IF ( INP(&F8FC) AND 8 ) <> 8 THEN 20
50 a = INP(&F8FD)
60 IF a = &13 THEN 90
70 MEM3 = a
80 RETURN
90 MEM2 = a
100 GOTO 20
```



Ce programme devrait fonctionner sans problème, mais par expérience, nous nous sommes aperçus qu'il sautait parfois des caractères, surtout lors de l'appui sur une touche fonction qui envoie deux codes se succédant. Aussi avons-nous décidé d'écrire cette routine en assembleur, puis de la traduire en code machine.

Pour cela, nous pouvons réutiliser l'algorithme ci-dessus et le perfectionner en construisant un ordiogramme, qui est l'amélioration de l'algorithme en le rendant spécifique au matériel utilisé : ici le Z-80 et son code machine.





Nous trouvons dans ce second ordinoigramme un symbole non encore rencontré : un chiffre entouré d'un cercle.

Ce symbole est spécifique d'un renvoi : lorsqu'il n'y a pas assez de place pour représenter un algorithme (ou un ordinoigramme) sur une même feuille, on place ce type de repère qui renvoie au même repère sur une autre feuille (dans le sens de cheminement de l'algorithme).

Nous pouvons maintenant traduire l'ordinoigramme directement en code machine, en choisissant d'implanter le programme à partir de l'adresse &A000 (ou \$A000 - si vous travaillez avec un assembleur, le caractère \$ doit remplacer le caractère & du Basic qui indique une valeur hexadécimale — Il sera utilisé indifféremment \$ ou & dans la suite des explications, sauf dans les instructions Basic où le caractère & est obligatoire).

Vous trouverez le programme traduit en assembleur et en code machine. A gauche, on remarquera le champ d'adresses, c'est-à-dire l'emplacement mémoire où se trouveront les valeurs hexadécimales, vient ensuite le champ étiquettes qui permet d'installer des « Labels » pour les sauts, puis l'assembleur (mnémoniques) et la traduction en code machine de cet assembleur.

Adresse étiquette	Assembleur	Code machine
A000	XOR A	AF
A001	LD MEM1,A	32 2E A0
A004	LD MEM2,A	32 2F A0

Adresse étiquette	Assembleur	Code machine
A007	LD MEM3,A	32 30 A0
A00A boucl1	LD BC,\$F8FC	01 FC F8
A00D	IN A,(C)	ED 78
A00F	BIT 2,A	CB 57
A011	RET NZ	C0
A012	LD MEM1,A	32 2E A0
A015	BIT 3,A	CB 47
A017	JP Z boucl1	CA 0A A0
A01A	LD BC,\$F8FD	01 FD F8
A01D	IN A,(C)	ED 78
A01F	CP \$13	FE 13
A021	JP NZ boucl2	C2 2A A0
A024	LD MEM2,A	32 2F A0
A027	JP boucl1	C3 0A A0
A02A boucl2	LD MEM3,A	32 30 A0
A02D	RET	C9
A02E MEM1		
A02F MEM2		
A030 MEM3		

Programme : 46 octets de \$A000 A \$A02D

3 octets réservés aux variables : \$A02E, \$A02F, \$A030

• Explications du programme

A l'adresse \$A000 nous trouvons l'instruction **XOR A** qui effectue un OU EXCLUSIF entre le registre A et lui-même (le registre A peut être comparé à une variable A du Basic). Cette instruction est un « truc » qui permet de mettre \$00 dans A.

De l'adresse \$A001 à \$A009, les trois instructions **LD MEMX,A** permettent de remplir les trois adresses \$A02E, \$A02F, \$A030 avec la valeur \$00 contenue dans A.

En \$A00A on charge les registres B et C avec la valeur \$F8FC qui adressera le port correspondant (LD BC,\$F8FC).

IN A,(C) à l'adresse \$A00D charge dans A le contenu du port dont l'adresse est contenue dans B et C, ici \$F8FC.

On teste ensuite le bit 2 du registre A afin de savoir si la communication a été rompue (BIT 2,A).

Si ce bit est non nul, la communication est rompue, il faut donc sortir de la routine (**RET NZ**).

En \$A012, le contenu de A est sauvegardé à l'adresse \$A02E pour traitement ultérieur (quand on sortira de la routine). En fait, on cherchera à savoir si le contenu de cette adresse est nul ou non (si l'on est sorti de la routine avant, il est nul, sinon il contient un nombre non nul).

Ensuite on teste le bit 3 de A (BIT 3,A) afin de savoir si un caractère a été reçu.

En \$A017 on effectue un saut en \$A00A qui recommencera les tests de \$F8FC si le bit 3 est nul, aucun caractère n'a été reçu.

Si un caractère a été reçu, on le lit sur le port d'adresse \$F8FD (lignes d'adresses \$A01A à \$A01E).

Le caractère reçu est comparé à la valeur \$13 (CP \$13) qui indique qu'une touche fonction du Minitel distant a été frappée.

Si ce n'est pas le cas, on effectue un branchement à l'adresse \$A02A (JP NZ boucl2).

Si le caractère est bien égal à \$13 alors on le sauvegarde ligne \$A024 à l'emplacement mémoire \$A02F (LD MEM2,A) puis on effectue un branchement inconditionnel à l'adresse \$A00A (JP boucl1) (équivalent au GOTO du Basic).

Adresse \$A02A : le caractère reçu n'est pas égal à \$13, on le sauvegarde donc à l'emplacement mémoire \$A030 (LD \$A030,A).

Puis on sort de la boucle par RET.

Les emplacements mémoire A02E à \$A030 sont réservés à la sauvegarde des valeurs précédemment citées (MEM1, MEM2, MEM3).

Vous pourrez écrire cette routine à l'aide du logiciel d'assemblage décrit Partie 9 chapitre 2.2, ou tout autre logiciel d'assemblage en votre possession. Lors de l'utilisation, il sera alors chargé directement en binaire par l'instruction LOAD. Une protection par l'instruction MEMORY &9FFF sera nécessaire avant le lancement du serveur.

Cette routine en assembleur pourra être appelée à partir du programme écrit en Basic par l'instruction CALL &A000.

Les différentes valeurs seront récupérées par l'instruction PEEK (adresse).

— A = PEEK(&A02E) chargera le contenu de l'adresse \$A02E et permettra de vérifier si le Minitel est toujours connecté sur le serveur (si c'est le cas, la valeur 0 sera lue à cette adresse).

— A = PEEK(&A02F) permettra de vérifier la réception du caractère \$13 si l'utilisateur a frappé une touche de fonction de son Minitel. Si c'est le cas, on connaîtra la touche de fonction en testant le contenu de l'adresse \$A030 IF PEEK(&A030) = code ...

sinon, le caractère lu en \$A030 sera un caractère du code ASCII standard.

Si vous ne possédez pas de logiciel d'assemblage, le petit programme Basic suivant vous permettra d'implanter la routine très simplement :

```
10 MEMORY &9FFF
20 RESTORE 100
30 FOR A = &A000 TO &A02D
40 READ C$
```

```
50 C$ = "&" + C$
60 C = VAL(C$)
70 POKE A,C
80 NEXT A
90 STOP

100 DATA AF,32,2E,A0,32,2F,A0,32,30,A0,01,FC,F8
110 DATA ED,78,CB,57,C0,32,2E,A0,CB,47,CA,0A,A0
120 DATA 01,FD,F8,ED,78,FE,13,C2,2A,A0,32,2F,A0
130 DATA C3,0A,A0,32,30,A0,C9
```

Avant d'utiliser cette routine en langage machine, nous ne saurions trop vous conseiller de vérifier très soigneusement les données en ligne de DATA et de sauvegarder sur disquette ce petit programme sous peine, lors de l'utilisation, de devoir effectuer une ré-initialisation complète de l'ordinateur (**RESET** par action de l'interrupteur **MARCHE/ARRET**).

VII - Essais préliminaires

Essayons les programmes précédents afin de vérifier le bon fonctionnement de votre modem DTL2000+.

Au lieu de connecter le modem sur la ligne téléphonique, vous pouvez connecter en fil à fil les points 1 et 3 (voir Partie 10 chapitre 4.3 page 5) des connecteurs mâles du modem et du Minitel.

Essai de connexion

Vous devez lancer le programme d'initialisation du modem suivi du programme de connexion (paragraphe I et II). Si vous avez connecté un poste téléphonique sur le connecteur femelle à l'arrière du Minitel, vous pourrez entendre un faible sifflement indiquant la présence de la portuse. Appuyez alors sur la touche **<CONNEXION/FIN>** du Minitel qui aura pour effet d'inscrire le caractère **F** en vidéo inversée en haut et à droite de l'écran, indiquant que le Minitel est connecté sur votre Amstrad via le modem Digitelec.

Essai d'émission

Vous devrez frapper le programme d'émission d'une série de données du paragraphe IV à partir de la ligne 100 par exemple, puis insérer le petit programme suivant :

```
10 RESTORE 40
20 GOSUB 100
30 STOP
40 DATA &0C,&1F,&49,&49,&1B,&4F,&57,&45,&4B,&41,0
```

Le Minitel toujours connecté, vous pouvez lancer le programme et vous devez voir affiché sur l'écran : **WEKA** inscrit en caractères double grandeur.

Vous pourrez essayer la routine d'émission d'une chaîne alphanumérique en chargeant une variable **a\$** (par exemple : **a\$ = "WEKA"**) et en effectuant un **GOTO** le numéro de ligne où se trouvera implantée la routine du paragraphe V.

Essai en réception

Une fois la routine assembleur implantée et le Minitel connecté, frappez l'instruction : **CALL &A000 + <RETURN>** sur votre Amstrad.

Le curseur doit disparaître et l'Amstrad être indisponible. S'il y a réinitialisation, vous devrez vérifier le programme, vous avez certainement effectué une erreur de frappe. Si le programme vous redonne la main, relancez-le, vous avez certainement frappé une touche du Minitel par inadvertance. Sinon, frappez la touche **A** du Minitel. L'Amstrad va alors vous redonner la main ; frappez alors (sur le clavier de l'Amstrad) : **X = PEEK(&A030)**, puis **PRINT X** et vous devez obtenir l'affichage de 65 qui est la valeur décimale du code ASCII du caractère **A** majuscule.

Si l'Amstrad ne vous a pas redonné la main, vous devrez vérifier le programme, une erreur de frappe y est probablement insérée.

8/5.4.4

Le programme du serveur

Nous aurons besoin pour écrire le programme complet du serveur de l'étude menée précédemment, ainsi que de la connaissance des codes Vidéotex utilisés par le minitel. Vous pouvez pour cela vous procurer la notice des *Spécifications techniques d'utilisation du minitel (STUM)* auprès des Télécommunications, ou vous reporter à la Partie 10, chapitre 1.

Vous trouverez dans les paragraphes suivants, les différents programmes, organisés en modules, vous permettant de mettre en œuvre le serveur.

Ces programmes pourront être frappés, et enregistrés séparément, avec le nom d'enregistrement que nous vous proposerons sur votre disquette. Ils seront ensuite concaténés grâce à l'instruction **MERGE**. Les trois premiers programmes devront être frappés entièrement avant tout essai, les suivants venant s'insérer aux endroits prévus pour leur implantation.

Nous vous conseillons, de plus, de conserver ces programmes séparément, dans le cas où vous désireriez modifier certaines rubriques. Il vous suffira alors de ne concaténer que les services proposés durant les travaux. Le choix ne disparaît pas du menu général, mais si l'utilisateur le demande, il est prévu un message signalant le défaut.

Le programme DEBUT

Ce programme assure l'installation des codes machines hexadécimaux pour l'acquisition des caractères du minitel, et le lancement du programme SERV, qui est le logiciel gérant le service proprement dit.

Le programme, en codes machines Z 80, assurant la réception des caractères, sera résidant, le programme d'installation disparaîtra par contre après l'appel de SERV.

```
10 REM *** IMPLANTATION ROUTINE MACHINE
***
20 REM *** DE RECEPTION DE CARACTERES
***
30 REM *****
***
40 REM *** RESERVATION MEMOIRE ***
50 MEMORY &9FFF
60 REM *****
70 REM *** ENTREE CODE MACHINE ***
80 i=0
```

```
90 FOR a=&A000 TO &A02D
100 READ c$
110 c$="&" + c$
120 c=VAL(c$)
130 i=i+c
140 POKE a,c
150 NEXT a
160 IF i <> 5797 THEN 300
170 REM *****
180 REM *** LANCEMENT PROGRAMME INITIAL
***
190 MODE 2:PRINT"ATTENTE CONNEXION":RUN"
serv"
200 REM *****
***
210 REM *** CODES MACHINE ***
220 DATA af,32,2e,a0,32,2f,a0,32,30
230 DATA a0,01,fc,f8,ed,78,cb,57,c0
240 DATA 32,2e,a0,cb,47,ca,0a,a0,01
250 DATA fd,f8,ed,78,fe,13,c2,2a,a0
260 DATA 32,2f,a0,c3,0a,a0,32,30,a0
270 DATA c9
280 REM *****
290 REM *** ERREUR !!! ***
300 PRINT"Erreur dans les DATAs"
310 PRINT"Verifiez votre programme"
```

Nous ne nous attarderons pas sur les explications concernant ce listing, celles-ci ayant été fournies lors de la réflexion sur l'acquisition d'un caractère envoyé par le minitel (voir Partie 10, chapitre 1.2). Signalons uniquement le lancement du programme SERV en ligne 190, au cas où vous désireriez modifier le nom de celui-ci, ainsi que l'affichage "ATTENTE CONNEXION" signalant la libération de la ligne.

Le programme INIT

Ce programme est constitué des différentes routines de gestion de la communication avec le minitel, et de la mise en veille du Modem en l'attente d'une connexion.

Les différents sous-programmes le constituant seront ainsi appelés, chaque fois que vous désirerez émettre un caractère ou une chaîne de caractères, ou encore recevoir une chaîne de caractères du minitel.

```

10 GOTO 900
100 WHILE (INP(&F8FC) AND 2) <> 2
110 IF (INP(&F8FC) AND 4) =4 THEN RETURN
120 WEND
130 OUT &F8FD,a
140 RETURN
200 a=1
210 WHILE a <> 0
220 READ a
230 WHILE (INP(&F8FC) AND 2) <> 2
240 IF (INP(&F8FC) AND 4) =4 THEN RETURN
250 WEND
260 OUT &F8FD,a
270 WEND
280 RETURN
300 FOR i=1 TO LEN (a$)
310 WHILE (INP(&F8FC) AND 2) <> 2
320 IF (INP(&F8FC) AND 4) = 4 THEN RETURN
330 WEND
340 OUT &F8FD,ASC(MID$(a$,i))
350 NEXT i
360 RETURN
400 texte$=""
410 DATA 17,27,73,0:RESTORE 410:GOSUB 200
420 CALL &A000:z = PEEK(&A030):z1=PEEK(&A02F)
430 IF PEEK(&A02E)=0 THEN RUN"debut"
440 IF z1 = 19 AND z = 65 THEN GOTO 530
450 IF z1 = 19 AND z = 71 THEN GOSUB 640
460 IF z1 = 19 AND z = 69 THEN GOSUB 550
470 IF z1 = 19 THEN GOTO 420
480 IF z < 32 THEN GOTO 420
490 IF LEN(texte$) = x THEN a$ = CHR$(7)
:GOSUB 300:GOTO 420
500 a$=CHR$(z):GOSUB 300
510 texte$=texte$+a$
520 GOTO 420
530 DATA 20,0:RESTORE 530:GOSUB 200
540 RETURN
550 IF LEN(texte$) = 0 THEN RETURN

```

```

560 a$ = "":FOR i = 1 TO LEN(texte$):a$=
a$+CHR$(8):NEXT i
570 GOSUB 300
580 a$ = "":FOR i = 1 TO LEN(texte$):a$=
a$+CHR$(95):NEXT i
590 GOSUB 300
600 a$ = "":FOR i = 1 TO LEN(texte$):a$=
a$+CHR$(8):NEXT i
610 GOSUB 300
620 texte$=""
630 RETURN
640 IF LEN(texte$) = 0 THEN RETURN
650 a$ = CHR$(8) + CHR$(95) + CHR$(8):GO
SUB 300
660 texte$ = LEFT$(texte$,LEN(texte$)-1)
670 RETURN

```

Une fois frappé, ce programme sera sauvegardé sous le nom :

SAVE "INIT.BAS"

pour être concaténé ultérieurement.

- La ligne 10 : Réalisation d'un branchement direct en ligne 900, afin de ne pas exécuter les sous-programmes.
- Les lignes 100 à 140 : Sous-programme d'émission d'un caractère ASCII contenu dans la variable **a**.
- Les lignes 200 à 280 : Sous-programme d'émission d'une série de données placées en ligne de DATA, et se terminant par un zéro.
- Les lignes 300 à 360 : Sous-programme d'émission d'une chaîne de caractères contenue dans la variable **a\$**.

Tous les sous-programmes précédents ont été expliqués lors de l'analyse du programme du serveur.

Nous trouvons ensuite un sous-programme permettant d'acquérir une chaîne de caractères frappés à partir du minitel, et se terminant par la touche **<ENVOI>**. Le nombre de caractères maximal à recevoir aura préalablement été fixé à une valeur inférieure ou égale à 255 dans la variable **x**, la chaîne de caractère, se trouve en fin de programme dans la variable **TEXTE**.

- Ligne 400 : Préparation de la variable **TEXTE** destinée à mémoriser le texte frappé.
- Ligne 410 : Visualisation du curseur clignotant et changement de couleur pour le texte frappé.
- Ligne 420 : Attente d'un caractère.
- Ligne 430 : Si déconnexion inopportune, on relance tout.
- Ligne 440 : Test de la touche **<ENVOI>**.

- Ligne 450 : Test de la touche **<CORRECTION>**.
- Ligne 460 : Test de la touche **<ANNULATION>**.
- Ligne 470 : Test autre touche **<FONCTION>**.
- Ligne 480 : Test si caractère ASCII visualisable.
- Ligne 490 : Test si longueur maximale de la chaîne atteinte.
- Ligne 500 à 520 : Prise en compte du caractère reçu, renvoi en écho au minitel et ajout à la variable **TEXTE**.
- Lignes 530 et 540 : Fin de réception du texte, inhibition du curseur et retour au programme initial.
- Lignes 550 à 630 : L'utilisateur a appuyé sur la touche **<ANNULATION>**. On revient au début de la ligne de texte (code 8 autant de fois que de caractères frappés depuis le début). Le texte affiché sur l'écran du minitel est remplacé par le nombre correspondant de traits de soulignements (code décimal 95) puis retour au début du texte, et le contenu de la variable **TEXTE** est annulé. Cette façon de procéder n'est sûrement pas la plus rapide, ni la plus esthétique, mais elle permet d'être utilisée dans n'importe quel cas et à partir de n'importe quelle position sur l'écran.
- Lignes 640 à 670 : L'utilisateur a appuyé sur la touche **<CORRECTION>**. Retour d'une position en arrière (code 8), affichage d'un caractère de soulignement et de nouveau retour en arrière. On retire un caractère de la variable **TEXTE**.

Programme ATTENTE

Le listing suivant permet, lors de son exécution, de programmer le modem afin de le préparer à la communication lors de l'appel.

```
900 OUT &F8F9,&0:FOR i = 1 TO 10:NEXT
910 OUT &F8F8,&FE:FOR i = 1 TO 10:NEXT
920 OUT &F8F9,&4:FOR i = 1 TO 10:NEXT
930 OUT &F8F8,&0:FOR i = 1 TO 10:NEXT
940 OUT &F8FC,&3:FOR i = 1 TO 10:NEXT
950 OUT &F8FC,&40:FOR i = 1 TO 10:NEXT
960 OUT &F8F8,&BC:FOR i = 1 TO 10:NEXT
970 OUT &F8FC,&49:FOR i = 1 TO 10:NEXT
```

Vous aurez soin de sauvegarder cette partie du programme sous le nom :

SAVE "ATTENTE.BAS"

Programme CONNECTE

Ce programme permet d'attendre un appel sur la ligne téléphonique et d'effectuer la connexion le cas échéant.

```

1000 IF (JOY(0) = 0) AND (JOY(1) = 0) TH
EN 1000
1010 MODE 2:PRINT CHR$(7);"CONNEXION EN
COURS":a$ = INKEY$
1020 OUT &F8F8,&BA:FOR i=1 TO 10:NEXT i
1030 OUT &F8FC,&9:FOR i=1 TO 10:NEXT i
1040 FOR j = 1 TO 10000:NEXT j
1050 FOR j = 1 TO 10
1060 a$ = CHR$(12) + "Connexion Etablie"
1070 GOSUB 300
1080 NEXT j

```

Ce petit programme sera sauvé sous le nom :

SAVE "CONNECTE.BAS"

— Ligne 1000 : Attente du signallement d'un appel par le détecteur de sonnerie. Lors de la simulation, vous pourrez connecter un Joystick sur votre Amstrad, qui simulera un appel par l'action sur celui-ci. Si vous ne possédez pas de joystick, remplacez cette ligne par la ligne suivante :

1000 a\$ = INKEYS : IF a\$ = " " THEN 1000

Il vous suffira d'appuyer sur une touche pour effectuer la connexion. Vous pouvez même utiliser cette possibilité si vous ne possédez pas de détecteur de sonnerie, mais vous serez astreint, ou une personne de votre entourage, à être près de votre Amstrad lors de la mise en service du serveur.

— Ligne 1010 : Signalisation par un BIP et un message la prise de la ligne pour répondre à un appel.

— Lignes 1020 et 1030 : Connexion et envoi de la porteuse par le Modem, avec les temporisations appropriées.

— Ligne 1040 : Attente permettant à votre correspondant de se connecter, elle est suffisamment longue pour laisser réagir un correspondant (il pensera à allumer son minitel et à se connecter) ignorant la présence de votre serveur.

— Lignes 1050 à 1080 : Signalent au correspondant que la connexion est correctement établie. Vous pouvez d'ailleurs modifier sans problème le message de la ligne 1060 en remplaçant celui-ci. Le code décimal 12 assure l'effacement de l'écran du minitel appelant.

Premier essai intermédiaire

Avant d'aller plus loin dans la frappe du logiciel, nous vous proposons de suivre une procédure d'appel.

Vos programmes étant sauvegardés sous les noms préconisés, nous allons créer le programme SERV à partir des commandes suivantes :

NEW

MERGE "INIT.BAS"

MERGE "ATTENTE.BAS"

MERGE "CONNECTE.BAS"

SAVE "SERV.BAS"

Vous suivrez ensuite les explications concernant les branchements à effectuer pour la procédure d'essai au paragraphe *Essais préliminaires*, hors tension (Voir Partie 8, chapitre 5.4.3) et lancerez enfin le programme :

RUN "DEBUT"

Lors de la simulation de la connexion, votre minitel doit afficher le caractère **C** en vidéo inversée, puis quelques secondes plus tard, le message **Connexion établie** doit apparaître en clignotant 20 fois. Si vous estimez que le temps de connexion est trop long, vous pouvez raccourcir la durée de la boucle FOR-NEXT de la ligne 1040 dans le programme CONNECTE.

Tout doit fonctionner au premier essai, sinon nous vous conseillons de vérifier les branchements ainsi que la frappe des programmes.

Le programme PRESENTE

Cette partie du programme du serveur permet de présenter agréablement celui-ci par une page dessinée et une petite animation.

Elle est ici entièrement remplie en ligne de DATAs afin d'améliorer la rapidité d'affichage.

Nous l'avons créée au nom de WEKA, mais avec tous les renseignements mis à votre disposition dans notre ouvrage, vous serez à même de créer votre page de présentation seul.

Si vous ne désirez pas utiliser cette page, vous pouvez éliminer les lignes 2100 à 2380 par l'instruction : **DELETE 2100-2380** mais nous ne saurions trop vous conseiller de l'étudier si vous voulez vous en inspirer.

Ce programme assure aussi l'acquisition du nom du correspondant.

```

2000 PRINT CHR$(7);"***** ";CHR$(24
);"PAGE DE PRESENTATION";CHR$(24);" ****
*****"
2010 DATA 12,31,64,65,32,18,127,20,31,64
,65,27,81,32,76,79,71,73,67,73,69,76,32,
77,46,82,73,81,85,65,82,84,32,0
2020 RESTORE 2000:GOSUB 200
2030 DATA 31,64,65,32,0,8,27,87,32,0,8,2
7,84,32,0
2040 FOR i=1 TO 3:RESTORE 2030:FOR k=1 T
O 3:GOSUB 200:FOR j=1 TO 50:NEXT j:NEXT
k:NEXT i
2050 DATA 27,58,105,67,27,58,105,69,0
2060 RESTORE 2050:GOSUB 200
2070 DATA 31,64,65,32,18,127,0
2080 RESTORE 2070:GOSUB 200
2090 REM *** ECRAN D'ARRIVEE SUR LE SERV
EUR ***
2100 RESTORE 2120:GOSUB 200
2110 REM --- LETTRE W ---
2120 DATA 12,31,66,66,14,27,90,95,31,66,
72,14,27,90,95,31,67,66,14,27,90,95,95,3
2,32,32,95,95,31,68,66,14,27,90,95,95,32
,32,32,95,95
,31,69,66,14,27,90,95,95,32,95,32,95,95.
31,70,66,14,27,90,95,95,95,32,95,95,95,3
1,71,67,14,27,90,95,32,32,32,95
2130 REM --- LETTRE E ---
2140 DATA 31,66,74,14,27,90,95,18,68,31,
67,74,14,27,90,95,95,31,68,74,14,27,90,9
5,95,95,31,69,74,14,27,90,95,95,31,70,74
,14,27,90,95
,18,68,31,71,74,14,27,90,95,18,68
2150 REM --- LETTRE K ---
2160 DATA 31,66,80,14,27,90,95,95,32,32,
95,31,67,80,14,27,90,95,95,32,95,31,68,8
0,14,27,90,95,95,95,31,69,80,14,27,90,95
,18,67,31,70
,80,14,27,90,95,95,32,95,95,31,71,80,14,
27,90,95,95,32,95,95
2170 REM --- LETTRE A ---
2180 DATA 31,71,86,14,27,90,95,95,32,18,
66,95,95,31,70,86,14,27,90,95,18,70,31,6
9,87,14,27,90,95,95,32,95,95,31,68,87,14
,27,90,95,95
,32,95,95,31,67,88,14,27,90,95,95,95,31,
66,89,14,27,90,95,0
2190 REM --- SECOURSSES ---
2200 DATA 31,88,65,10,31,65,65,11,0,11,0
2210 FOR i=1 TO 20:RESTORE 2200:GOSUB 20
0:NEXT i:GOSUB 200

```

```

2220 REM --- ADRESSE ---
2230 DATA 31,74,68,27,70,27,77,27,65,0
2240 RESTORE 2230:GOSUB 200
2250 a$="82,rue Curial":GOSUB 300
2260 DATA 31,76,67,27,79,27,70,0
2270 RESTORE 2260:GOSUB 200
2280 a$="75019 PARIS":GOSUB 300
2290 DATA 31,68,94,27,87,27,64,27,72,27,
84,0
2300 RESTORE 2290:GOSUB 200
2310 a$=" Telephone:":GOSUB 300
2320 DATA 31,70,94,27,80,27,70,27,93,0
2330 RESTORE 2320:GOSUB 200
2340 a$="40 37 01 00":GOSUB 300
2350 FOR i=1 TO 20:RESTORE 2200:GOSUB 20
0:NEXT i
2360 DATA 31,81,65,27,68,27,77,0
2370 RESTORE 2360:GOSUB 200
2380 FOR i = 1 TO 500:NEXT i
2390 DATA 31,84,65,27,84,27,67,32,18,99,
31,84,66,0
2400 RESTORE 2390:GOSUB 200
2410 a$="Donnez moi votre nom ou un pseu
donyme: ":GOSUB 300
2420 DATA 31,85,70,95,18,71,0
2430 RESTORE 2420:GOSUB 200
2440 a$="+"+CHR$(27)+CHR$(93)+"ENVOI":GO
SUB 300
2450 DATA 31,85,70,0
2460 RESTORE 2450:GOSUB 200
2470 x=8:GOSUB 400
2480 nom$=texte$
2490 PRINT nom$+" est connecte sur le se
rveur"

```

Vous sauvegarderez ce programme sous le nom :

SAVE "PRESENTE.BAS"

— Ligne 2000 : Affichage sur l'écran de votre Amstrad du niveau de consultation dans lequel se trouve votre correspondant, ici la page de présentation.

— Ligne 2050 : Données permettant un affichage sur le minitel en mode scrolling.

— Lignes 2100 à 2180 : Envoi des codes permettant l'affichage de la première partie de la page de présentation. Un texte y est inséré mais grâce aux caractères semi-graphiques télételet (que l'on obtient à l'aide du code décimal 14).

- Lignes 2190 à 2210 : Obtention d'un effet de secousse du texte affiché grâce à une astuce consistant à écrire alternativement un caractère *line feed* (passage à la ligne du dessous) dans le bas de l'écran (code 10 décimal) et un caractère *vertical tabulation* (tabulation verticale - code décimal 11) dans le haut de l'écran, et en répétant cette séquence plusieurs fois.
- Lignes 2220 à 2340 : Suite de l'affichage de la page de présentation.
- Ligne 2350 : Nouvelles secousses, en réutilisant les données présentes en ligne 2200.
- Lignes 2360 à 2460 : Affichage de la requête du nom de l'utilisateur du serveur, accompagné de la méthode de validation.
- Lignes 2470 à 2480 : Acquisition d'une chaîne de 8 caractères.
- Ligne 2490 : Signalisation du nom de la personne connectée sur l'écran de l'Amstrad.

Les détails donnés ci-dessus, concernant l'organisation de cette partie du logiciel verneur, accompagné de connaissances sur les codes du Vidéo-text, vous permettront de modifier aisément tout ou une partie de cette page de présentation afin de l'adapter à votre propre serveur.

Le programme SOMMAIRE

Après avoir présenté votre serveur, il faut maintenant proposer le menu que nous appellerons SOMMAIRE.

Dans cette partie apparaissent neuf choix, plus le GUIDE.

Nous vous avons réservé des choix supplémentaires qui seront dénommés *En travaux*, et que vous pourrez compléter à votre guise, selon votre imagination,

Dans le présent développement, ils répondront par le message **Choix indisponible**, mais les explications fournies tout au long de notre étude vous permettront de les compléter.

```

3000 PRINT "*****"; CHR$(24); " SOMMAI
RE "; CHR$(24); "*****"
3010 DATA 20,31,88,65,0
3020 RESTORE 3010:GOSUB 200
3030 DATA 10,92,0:FOR i=1 TO 50:RESTORE
3030:GOSUB 200:NEXT
3040 DATA 31,65,65,14,27,68,96,112,112,4
8,31,66,65,27,83,27,64,32,49,32,14,27,68
,27,80,37,15,0," Affiche"
3050 RESTORE 3040:GOSUB 200:READ a$:GOSU
B 300
3060 DATA 31,67,67,14,27,68,96,112,112,4
8,31,68,67,27,83,27,64,32,50,32,14,27,68
,27,80,37,15,0," Boites Personnelles"

```

```
3070 RESTORE 3060:GOSUB 200:READ a$:GOSU
B 300
3080 DATA 31,69,69,14,27,68,96,112,112,4
8,31,70,69,27,83,27,64,32,51,32,14,27,68
,27,80,37,15,0," Presentation"
3090 RESTORE 3080:GOSUB 200:READ a$:GOSU
B 300
3100 DATA 31,71,71,14,27,68,96,112,112,4
8,31,72,71,27,83,27,64,32,52,32,14,27,68
,27,80,37,15,0," Informations"
3110 RESTORE 3100:GOSUB 200:READ a$:GOSU
B 300
3120 DATA 31,73,73,14,27,68,96,112,112,4
8,31,74,73,27,83,27,64,32,53,32,14,27,68
,27,80,37,15,0," Jeux"
3130 RESTORE 3120:GOSUB 200:READ a$:GOSU
B 300
3140 DATA 31,75,75,14,27,68,96,112,112,4
8,31,76,75,27,83,27,64,32,54,32,14,27,68
,27,80,37,15,0," Appel Sysop"
3150 RESTORE 3140:GOSUB 200:READ a$:GOSU
B 300
3160 DATA 31,77,77,14,27,68,96,112,112,4
8,31,78,77,27,83,27,64,32,55,32,14,27,68
,27,80,37,15,0," En travaux"
3170 RESTORE 3160:GOSUB 200:READ a$:GOSU
B 300
3180 DATA 31,79,79,14,27,68,96,112,112,4
8,31,80,79,27,83,27,64,32,56,32,14,27,68
,27,80,37,15,0," En travaux"
3190 RESTORE 3180:GOSUB 200:READ a$:GOSU
B 300
3200 DATA 31,82,75,14,27,68,96,112,112,1
12,112,112,112,48,31,83,75,27,83,27,64,2
7,72,32,71,85,73,68,69,32,27,73,14,27,68
,27,80,37,15
,27,71,0," Si vous ne connaissez"
3210 RESTORE 3200:GOSUB 200:READ a$:GOSU
B 300
3220 DATA 31,84,78,0,"pas le fonctionnem
ent de ce      ç      serveur"
3230 RESTORE 3220:GOSUB 200:READ a$:GOSU
B 300
3240 DATA 31,88,66,27,72,27,85,0," Appuy
er sur la touche de votre choix "
3250 RESTORE 3240:GOSUB 200:READ a$:GOSU
B 300
3260 CALL &A000:z=PEEK(&A030):z1=PEEK(&A
02F)
3270 IF z1=19 AND z=70 THEN 3010
3280 IF PEEK(&A02E)=0 THEN RUN"debut"
```

```
3290 IF z1=19 AND z=68 THEN a$="10":GOTO
3320
3300 IF z1=19 THEN a$ = CHR$(7):GOSUB 30
0:GOTO 3260
3310 a$=CHR$(z)
3320 IF a$="1" THEN 5000
3330 IF a$="2" THEN 8000
3340 IF a$="3" THEN 11000
3350 IF a$="4" THEN 14000
3360 IF a$="5" THEN 17000
3370 IF a$="6" THEN 20000
3380 IF a$="7" THEN 23000
3390 IF a$="8" THEN 26000
3400 IF a$="9" THEN 29000
3410 IF a$="10" THEN 32000
3420 a$ = CHR$(7):GOSUB 300:GOTO 3260
5000 DATA 12,31,75,68,27,85,32,18,96,31,
75,69,27,77,0
5010 RESTORE 5000:GOSUB 200
5020 a$="Ce choix n'est pas encore actif
":GOSUB 300
5030 FOR i=1 TO 1000:NEXT i
5040 GOTO 3010
8000 DATA 12,31,75,68,27,85,32,18,96,31,
75,69,27,77,0
8010 RESTORE 8000:GOSUB 200
8020 a$="Ce choix n'est pas encore actif
":GOSUB 300
8030 FOR i=1 TO 1000:NEXT i
8040 GOTO 3010
11000 DATA 12,31,75,68,27,85,32,18,96,31
,75,69,27,77,0
11010 RESTORE 11000:GOSUB 200
11020 a$="Ce choix n'est pas encore acti
f":GOSUB 300
11030 FOR i=1 TO 1000:NEXT i
11040 GOTO 3010
14000 DATA 12,31,75,68,27,85,32,18,96,31
,75,69,27,77,0
14010 RESTORE 14000:GOSUB 200
14020 a$="Ce choix n'est pas encore acti
f":GOSUB 300
14030 FOR i=1 TO 1000:NEXT i
14040 GOTO 3010
17000 DATA 12,31,75,68,27,85,32,18,96,31
,75,69,27,77,0
17010 RESTORE 17000:GOSUB 200
17020 a$="Ce choix n'est pas encore acti
f":GOSUB 300
17030 FOR i=1 TO 1000:NEXT i
17040 GOTO 3010
```

```
20000 DATA 12,31,75,68,27,85,32,18,96,31
,75,69,27,77,0
20010 RESTORE 20000:GOSUB 200
20020 a$="Ce choix n'est pas encore acti
f":GOSUB 300
20030 FOR i=1 TO 1000:NEXT i
20040 GOTO 3010
23000 DATA 12,31,75,68,27,85,32,18,96,31
,75,69,27,77,0
23010 RESTORE 23000:GOSUB 200
23020 a$="Ce choix n'est pas encore acti
f":GOSUB 300
23030 FOR i=1 TO 1000:NEXT i
23040 GOTO 3010
26000 DATA 12,31,75,68,27,85,32,18,96,31
,75,69,27,77,0
26010 RESTORE 26000:GOSUB 200
26020 a$="Ce choix n'est pas encore acti
f":GOSUB 300
26030 FOR i=1 TO 1000:NEXT i
26040 GOTO 3010
29000 DATA 12,31,75,68,27,85,32,18,96,31
,75,69,27,77,0
29010 RESTORE 29000:GOSUB 200
29020 a$="Ce choix n'est pas encore acti
f":GOSUB 300
29030 FOR i=1 TO 1000:NEXT i
29040 GOTO 3010
32000 DATA 12,31,75,68,27,85,32,18,96,31
,75,69,27,77,0
32010 RESTORE 32000:GOSUB 200
32020 a$="Ce choix n'est pas encore acti
f":GOSUB 300
32030 FOR i=1 TO 1000:NEXT i
32040 GOTO 3010
```

Vous sauvegarderez ce programme sur disquette sous le nom :

SAVE "SOMMAIRE.BAS"

- Ligne 3000 : Affichage sur l'écran du CPC du niveau de consultation du correspondant.
- Lignes 3010 à 3030 : Scrolling vertical avec affichage de deux lignes diagonales. Cette façon de procéder assure un effacement de l'écran tout en réalisant une animation agréable.
- Lignes 3040 et 3050 : Affichage du numéro 1 ainsi que le choix proposé avec ce numéro.
- Lignes 3060-70, 3080-90, 3100-10, 3120-30, 3140-50, 3160-70, 3180-90 : Affichage des autres possibilités du sommaire, sur la même présentation que le premier choix, mais légèrement décalées.

- Lignes 3200 à 3230 : Affichage de la possibilité d'utiliser la touche **GUIDE** pour informations.
- Lignes 3240 et 3250 : Affichage de l'attente d'un choix.
- Ligne 3260 : Attente de l'acquisition d'un caractère frappé au minitel. Cette acquisition s'effectue en une seule fois, c'est-à-dire que l'utilisateur ne sera pas obligé de frapper la touche **< ENVOI >** pour valider son choix, celui-ci sera automatiquement pris en compte.
- Lignes 3270 à 3410 : Différents tests sur la valeur du code ASCII reçu, permettent d'aiguiller le programme selon le choix de l'utilisateur. On remarquera en ligne 3280 la détection d'une éventuelle perte du correspondant, qui relance le programme début.
- Ligne 3420 : Si aucun des choix proposés n'est choisi, le caractère **Cloche** est envoyé afin d'émettre un BIP sur le minitel pour signaler à l'utilisateur son erreur.
- Lignes 5000 à 5040 : Cette partie du programme ne concernant que le menu, aucune des rubriques contenant le traitement du choix n'est ici développée. Afin de ne pas perturber le bon fonctionnement du serveur, nous avons inséré ici l'envoi du message **Ce choix n'est pas encore actif** en réponse au choix numéro 1.
- Lignes 8000 à 8040, 11000-11040, 14000-14040, 17000-17040, 20000-20040, 23000-23040, 26000-26040, 29000-29040, 32000-32040 : de même que pour les lignes 8000 à 8040, l'indisponibilité du choix est affichée. On fera attention lors de la frappe à ne pas recopier lignes à lignes, car les instructions **RESTORE** ne sont pas affectées aux mêmes numéros de lignes. Le message est visualisé quelques secondes puis le menu est ré-affiché pour un nouveau choix.

Deuxième essai intermédiaire

Nous vous conseillons d'effectuer maintenant un nouvel essai de tous les programmes qui ont été proposés.

Ensuite, vous serez capable d'effectuer les essais suivants au fur et à mesure du développement de notre serveur.

Vous frapperez les commandes suivantes :

NEW

MERGE "INIT.BAS"

MERGE "ATTENTE.BAS"

MERGE "CONNECTE.BAS"

MERGE "PRESENTE.BAS"

MERGE "SOMMAIRE.BAS"

SAVE "SERV.BAS"

En vous reportant à la procédure expliquée lors du premier essai, vous pourrez suivre le déroulement du programme, et corriger vos éventuelles erreurs.

Le programme AFFICHE

Nous allons maintenant découvrir un des choix les plus intéressants du serveur : l'affiche sur laquelle vos correspondants pourront laisser des messages de deux lignes.

INITIALISATION DU FICHIER D'AFFICHAGE

Les messages laissés lors d'une consultation devront être sauvegardés au fur et à mesure sur disquette, afin de ne pas être perdus lorsque vous effectuerez des modifications sur le serveur, ou lors d'une panne de courant.

Il nous faut pour cela préparer ce fichier, et, avant toute utilisation, l'initialiser avec des messages vides.

Nous vous proposons pour cela le programme suivant :

```
10 DIM I$(21)
20 FOR i = 1 TO 20
30 I$(i) = SPACE$(80)
40 NEXT i
50 OPENOUT "affich."
60 FOR i = 1 TO 20
70 WRITE #9,I$(i)
80 NEXT i
90 CLOSEOUT
100 END
```

Il chargera le fichier affich (sans e) avec des messages blancs de 80 caractères espaces.

LE CHOIX DE L'AFFICHE

Dans ce premier choix, l'utilisateur sera invité à patienter quelques instants, afin de laisser le logiciel charger les vingt messages du fichier.

Les dix premiers messages sont ensuite affichés chacun dans une couleur de fond différente du précédent, afin de le différencier.

Votre correspondant aura ensuite le choix entre visualiser la page suivante de l'affiche (touche <SUITE>, retourner au sommaire (touche <SOMMAIRE>), ou laisser un message (touche <E> pour Ecrire).

Lors de la consultation de la deuxième page, il peut revenir à la page précédente grâce à la touche <RETOUR>.

Le programme est conçu de telle sorte que lors de l'appui sur la touche sommaire, les messages sont sauvegardés, mais vous pourrez installer un *flag* (variable servant de drapeau) qui pourra vous indiquer s'il est utile de sauvegarder le fichier de message (action, bien sûr, inutile si aucun message n'a été écrit).

```

5000 PRINT"*****"+CHR$(24)+"AFFICHE"+
CHR$(24)+"*****"
5010 DATA 12,31,64,65,32,18,103,12,31,70
,65,14,27,90,127,18,99,31,70,79,27,77,0
5020 RESTORE 5010:GOSUB 200
5030 a$=" L'affiche ":GOSUB 300
5040 DATA 12,27,58,106,71,32,20,31,64,65
,32,18,99,12,0
5050 FOR i=1 TO 1000:NEXT i
5060 RESTORE 5040:GOSUB 200
5070 DATA 31,85,69,27,64,27,84,27,77,0
5080 RESTORE 5070:GOSUB 200
5090 a$=" Veuillez patienter...":GOSUB 3
00
5100 CLEAR:DIM l$(50):i=1
5110 OPENIN"affich"
5120 WHILE NOT EOF
5130 INPUT#9,l$(i)
5140 i=i+1
5150 WEND
5160 CLOSEIN
5170 RESTORE 5040:GOSUB 200
5180 DATA 27,70,27,93,0
5190 DATA 27,69,27,93,0
5200 FOR j=1 TO 10 STEP 2
5210 RESTORE 5180
5220 a$=l$(j):GOSUB 200:GOSUB 300
5230 a$=l$(j+1):GOSUB 200:GOSUB 300
5240 NEXT j
5250 DATA 31,88,74,27,93,0,"suite"
5260 DATA 27,92,0,","
5270 DATA 27,93,0,"e"
5280 DATA 27,92,0,"crime,"
5290 DATA 27,93,0,"sommaire"
5300 RESTORE 5250
5310 FOR j=1 TO 5
5320 GOSUB 200:READ a$:GOSUB 300
5330 NEXT j
5340 CALL &A000
5350 IF PEEK(&A02E)=0 THEN RUN "debut"
5360 IF PEEK(&A02F)=19 THEN 5390
5370 IF (PEEK(&A030)=101 OR PEEK(&A030)=
69) THEN 5510
5380 a$=CHR$(7):GOSUB 300:GOTO 5340
5390 IF PEEK(&A030)=72 THEN 5600

```

```

5400 IF PEEK(&A030)=66 THEN 5170
5410 IF PEEK(&A030)=70 THEN 5430
5420 a$=CHR$(7):GOSUB 300:GOTO 5250
5430 RESTORE 5250:GOSUB 200
5440 a$="      Je sauve les messages      ":
GOSUB 300
5450 OPENOUT"affich"
5460 FOR i=1 TO 20
5470 WRITE#9,1$(i)
5480 NEXT i
5490 CLOSEOUT
5500 GOTO 3000
5510 DATA 31,88,73,27,72,27,84,0,"      S
i fini : -> ENVOI      "
5520 RESTORE 5510:GOSUB 200
5530 READ a$:GOSUB 300
5540 DATA 31,86,65,27,93,95,18,103,18,10
4,11,11,17,0
5550 RESTORE 5540:GOSUB 200
5560 x=80
5570 GOSUB 400
5580 FOR i=0 TO 19:1$(i)=1$(i+1):NEXT i
5590 1$(20) = texte$ + SPACE$(80-LEN(tex
te$))
5600 RESTORE 5040:GOSUB 200
5610 FOR j=11 TO 20 STEP 2
5620 RESTORE 5180
5630 a$=1$(j):GOSUB 200:GOSUB 300:a$=1$(
j+1):GOSUB 200:GOSUB 300
5640 NEXT j
5650 DATA 31,88,74,27,93,0,"retour",27,9
2,0,"",27,93,0,"e",27,92,0,"crire",27,
93,0,"sommaire":RESTORE 5650:FOR j=1 TO
5:GOSUB 200:
READ a$:GOSUB 300:NEXT j:CALL &A000
5660 GOTO 5350

```

- Ligne 5000 : Affichage signalant que l'utilisateur se trouve dans l'option AFFICHE.
- Lignes 5010 à 5030 : Affichage du choix sur l'écran du minitel.
- Lignes 5040 à 5090 : Invitation à la patience durant le temps de chargement du fichier.
- Lignes 5100 à 5160 : Chargement du fichier. Le nombre de chaînes de caractères à remplir n'est heureusement pas important, le temps d'attente sera de quelques secondes.
- Lignes 5180 et 5190 : Données servant à l'affichage des fonds de couleurs alternativement différentes.

- Lignes 5200 à 5240 : Affichage des dix premiers messages. On remarquera que dans la boucle, deux messages sont affichés, ce qui nécessite l'emploi de l'instruction **STEP 2** en ligne 5200.
- Lignes 5250 à 5330 : Propositions des options possibles à l'utilisateur, lors de l'affichage de cette première page.
- Lignes 5340 : Attente du choix.
- Lignes 5350 à 5430 : Test des différents choix possibles, ainsi que de l'éventuelle déconnexion, ou perte de porteuse.
- Lignes 5440 à 5500 : Traitement de l'appui sur la touche **<SOMMAIRE>** par la sauvegarde des messages (signalée sur le minitel afin de faire patienter le correspondant), et le retour au menu du sommaire.
- Lignes 5510 à 5550 : Traitement de l'appui sur la touche **Ecrire** par l'affichage d'une zone de deux lignes de 40 caractères, et de l'invitation à frapper le message.
- Lignes 5560 à 5570 : Acquisition du message.
- Lignes 5580 à 5590 : Validation du message dans les variables du fichier par un effet de pile qui élimine le message le plus ancien.
- Lignes 5600 à fin : Affichage de la deuxième page du fichier de message. Elle permet de vérifier que le message est bien inscrit, ou est appelée par le choix **<SUITE>**. Cette partie du programme propose aussi le choix **<RETOUR>** pour accéder à la page précédente. Bien sûr le traitement de ce choix est effectué à partir de la ligne 5170.

Le programme GUIDE

Ce programme, bien que commençant à la ligne 32000 peut être écrit à n'importe quel moment de la création du serveur, l'instruction **MERGE** pouvant s'effectuer sur n'importe quel programme basic, sans tenir compte de la numérotation.

Notre guide indique, ici, les références du serveur ainsi que les différentes rubriques et leur mode d'utilisation.

```

32000 PRINT "***** "+CHR$(24)+" GUIDE
      "+CHR$(24)+" *****"
32010 DATA 12,31,88,65,0,"Le logiciel de
      ce serveur a été créé par M. RIQUART pour
      les Editions WEKA"
32020 RESTORE 32010
32030 GOSUB 200:READ a$:GOSUB 300
32040 DATA 13,10,0,"      82, rue Curial 75
      019 PARIS"
32050 GOSUB 200:READ a$:GOSUB 300

```

```
32060 DATA 13,10,0,"Telephone: (1) 40 3
7 01 00"
32070 GOSUB 200:READ a$:GOSUB 300
32080 DATA 13,10,0,"Telecopieur: (1) 40
37 01 00"
32090 GOSUB 200:READ a$:GOSUB 300
32100 DATA 13,10,0,"Telex: 210 504"
32110 GOSUB 200:READ a$:GOSUB 300
32120 DATA 13,10,96,18,103,0,"Les touche
s a frapper sont indiquees par des choix
,ou des touches speciales ("
32130 GOSUB 200:READ a$:GOSUB 300
32140 DATA 27,93,0,"SOMMAIRE"
32150 GOSUB 200:READ a$:GOSUB 300
32160 DATA 27,92,0,""
32170 GOSUB 200:READ a$:GOSUB 300
32180 DATA 27,93,0,"SUITE"
32190 GOSUB 200:READ a$:GOSUB 300
32200 DATA 27,92,0,""
32210 GOSUB 200:READ a$:GOSUB 300
32220 DATA 27,93,0,"RETOUR"
32230 GOSUB 200:READ a$:GOSUB 300
32240 DATA 27,92,0,"..,par exemple)"
32250 GOSUB 200:READ a$:GOSUB 300
32260 DATA 13,10,96,18,103,27,90,0," Aff
iche"
32270 GOSUB 200:READ a$:GOSUB 300
32280 DATA 27,89,0," :cette rubrique vou
s permet de laisser des messages pour le
s prochains visiteurs sur le serveur"
32290 GOSUB 200:READ a$:GOSUB 300
32300 DATA 13,10,96,18,103,27,90,0," Boi
tes Personnelles"
32310 GOSUB 200:READ a$:GOSUB 300
32320 DATA 27,89,0," :cette rubrique per
met de laisser des messages personnels
aux inscrits permanents sur le serveur"
32330 GOSUB 200:READ a$:GOSUB 300
32340 DATA 13,10,96,18,103,27,90,0," Pre
sentation"
32350 GOSUB 200:READ a$:GOSUB 300
32360 DATA 27,89,0," :cette rubrique vou
s renseigne sur la composition du s
erveur"
32370 GOSUB 200:READ a$:GOSUB 300
32380 DATA 13,10,96,18,103,32,18,8,27,81
,27,72,0," Appuyez sur une touche..."
32390 GOSUB 200:READ a$:GOSUB 300
32400 CALL &A000
32410 DATA 31,88,65,27,90,0," Informatio
ns"
```

```

32420 GOSUB 200:READ a$:GOSUB 300
32430 DATA 27,89,0," :cette rubrique vous
s donne toute information concernant votre
quartier, les nouveautés sur le serveur ... "
32440 GOSUB 200:READ a$:GOSUB 300
32450 DATA 13,10,96,18,103,27,90,0," Jeux "
32460 GOSUB 200:READ a$:GOSUB 300
32470 DATA 27,89,0," :sans commentaire"
32480 GOSUB 200:READ a$:GOSUB 300
32490 DATA 13,10,96,18,103,27,90,0," Appel du Sysop"
32500 GOSUB 200:READ a$:GOSUB 300
32510 DATA 27,89,0," :vous permet de discuter
en direct avec l'opérateur système de l'amstrad
(s'il est là!)"
32520 GOSUB 200:READ a$:GOSUB 300
32530 DATA 13,10,96,18,103,32,18,8,27,31,
27,72,0," Appuyez sur une touche..."
32540 GOSUB 200:READ a$:GOSUB 300
32550 CALL &A000
32560 GOTO 3000

```

Vous prendrez soin de sauvegarder ce programme sous le nom : **SAVE "GUIDE.BAS"**

— Ligne 32000 : Signale au programmeur que le correspondant consulte le guide.

— Lignes 32010 à 32370 : Ces lignes permettent l'affichage d'une page d'explications en mode scrolling, permettant à l'utilisateur de la consulter au fur et à mesure. On remarquera entre les lignes 32140 à 32250, l'alternance de codes d'inversion vidéo et de retour en affichage normal.

— Lignes 32380 à 32400 : Notre guide comportant plusieurs pages, il fut nécessaire de laisser un message indiquant l'attente de l'appui sur une touche du minitel pour permettre la visualisation de la page suivante. Cet appui étant indifférent quant au caractère frappé, on ne teste pas ce dernier.

— Lignes 32410 à 32520 : Nouvelle page d'explications.

— Lignes 32530 à 32550 : Nouvelle attente d'appui sur une touche.

— Ligne 32560 : Retour à la page de SOMMAIRE mais nous pouvons effectuer ce retour à la page d'accueil, en frappant la ligne basic suivante : **32560 GOTO 2000**

Si vous désirez laisser plus d'explications, il vous suffira d'insérer de nouvelles lignes de données et de commandes permettant d'émettre votre texte sur le minitel. Vous n'oublierez pas d'insérer les lignes d'attente d'appuis sur une touche, pour passer à la page suivante.

Le programme SYSOP

Nous avons décidé de donner la possibilité à l'utilisateur du serveur d'appeler le fantôme du serveur, qui ne sera autre que vous, si vous êtes présent bien sûr !

Lors de la sélection de la rubrique **Appel SYSOP** (= *SYStème OPérateur*), votre Amstrad émettra, si vous n'avez pas réglé le son du haut-parleur au minimum, un BIP strident, vous signalant l'appel.

L'utilisateur sera prévenu de votre recherche dans le serveur.

Si après quelques dizaines de secondes vous n'avez pas répondu par la frappe d'une touche du CPC, le minitelliste sera prévenu de votre absence, et le menu du SOMMAIRE lui sera affiché.

La discussion s'effectuera tour à tour, le passage au partenaire suivant s'effectuera suite à la commande **<ENVOI>** pour le minitelliste, et par la touche **<RETURN>** pour le Sysop.

Attention, la seule commande de correction pour le Sysop est la touche **DEL** ; toute autre touche, telles les flèches de déplacement, est prohibée, elle produirait l'envoi d'un caractère vidéotex que votre correspondant ne comprendrait pas.

```

20000 PRINT"***** "+CHR$(24)+" APPEL
      SYSOP "+CHR$(24)+" *****"
20010 x = 0: AFTER 1000,2 GOSUB 20470
20020 DATA 12,31,70,66,27,81,32,18,101,3
      1,70,67,27,67,27,72,27,77,0
20030 RESTORE 20020:GOSUB 200
20040 a$ = "Veuillez patienter, je l'app
      elle ..."
20050 GOSUB 300
20060 FOR i = 1 TO 20:PRINT CHR$(7);:NEX
      T i
20070 a$=INKEY$
20080 IF x = 1 THEN GOTO 3000
20090 IF a$="" THEN GOTO 20070
20100 IF x = 1 THEN GOTO 3000
20110 DATA 12,27,68,27,86,0
20120 RESTORE 20110:GOSUB 200
20130 x= REMAIN(2)
20140 a$ = " Vous etes en direct avec le
      Sysop "
20150 GOSUB 300
20160 DATA 31,66,65,0
20170 RESTORE 20160:GOSUB 200
20180 a$ = "Frappez votre texte puis <EN
      VOI>"
20190 GOSUB 300
20200 DATA 13,10,96,18,103,0
20210 RESTORE 20200:GOSUB 200
20220 PRINT:PRINT"C'est au minitelliste:
      "

```

```
20230 a$="C'est a "+nom$+" : "+CHR$(13)+CHR$(10):GOSUB 300
20240 CALL &A000
20250 IF PEEK(&A02E) = 0 THEN RUN "debut
"
20260 IF PEEK(&A02F) = 19 AND PEEK (&A030) = 70 THEN GOTO 3000
20270 IF PEEK(&A02F) = 19 AND PEEK (&A030) = 71 THEN GOTO 20340
20280 IF PEEK(&A02F) = 19 AND PEEK (&A030) = 65 THEN GOTO 20370
20290 IF PEEK(&A02F) = 19 THEN GOTO 20240
20300 IF PEEK(&A030) < 32 THEN GOTO 20240
20310 a$ = CHR$(PEEK(&A030)):PRINT a$;
20320 GOSUB 300
20330 GOTO 20240
20340 a$=CHR$(8)+CHR$(32)+CHR$(8):GOSUB 300
20350 PRINT a$;
20360 GOTO 20240
20370 DATA 13,10,96,18,103,0
20380 RESTORE 20370:GOSUB 200
20390 a$="C'est au Sysop: "+CHR$(13)+CHR$(10):GOSUB 300
20400 PRINT:PRINT a$;
20410 a$=INKEY$
20420 IF a$ = "" THEN 20410
20430 IF ASC(a$) = 127 THEN a$=CHR$(8)+CHR$(32)+CHR$(8):PRINT a$;:GOSUB 300:GOTO 20410
20440 IF ASC(a$) = 13 THEN GOTO 20210
20450 PRINT a$;:GOSUB 300
20460 GOTO 20410
20470 x=REMAIN(2):x=1
20480 a$=CHR$(12)+"Desole, il ne semble pas etre present"
20490 GOSUB 300
20500 FOR i=1 TO 2000:NEXT i
20510 RETURN
```

Le programme ainsi frappé sera sauvegardé sous le nom :
SAVE "SYSOP.BAS".

Le programme SYSOP

Nous avons décidé de donner la possibilité à l'utilisateur du serveur d'appeler le fantôme du serveur, qui ne sera autre que vous, si vous êtes présent bien sûr !

Lors de la sélection de la rubrique **Appel SYSOP** (= *SYStème OPera-tor*), votre Amstrad émettra, si vous n'avez pas réglé le son du haut-parleur au minimum, un BIP strident, vous signalant l'appel.

L'utilisateur sera prévenu de votre recherche dans le serveur.

Si après quelques dizaines de secondes vous n'avez pas répondu par la frappe d'une touche du CPC, le minitelliste sera prévenu de votre absence, et le menu du SOMMAIRE lui sera affiché.

La discussion s'effectuera tour à tour, le passage au partenaire suivant s'effectuera suite à la commande <ENVOI> pour le minitelliste, et par la touche <RETURN> pour le Sysop.

Attention, la seule commande de correction pour le Sysop est la touche **DEL** ; toute autre touche, telles les flèches de déplacement, est prohibée, elle produirait l'envoi d'un caractère vidéotex que votre correspondant ne comprendrait pas.

```

20000 PRINT"***** "+CHR$(24)+" APPEL
      SYSOP "+CHR$(24)+" *****"
20010 x = 0: AFTER 1000,2 GOSUB 20470
20020 DATA 12,31,70,66,27,81,32,18,101,3
      1,70,67,27,67,27,72,27,77,0
20030 RESTORE 20020:GOSUB 200
20040 a$ = "Veuillez patienter, je l'app
      elle ..."
20050 GOSUB 300
20060 FOR i = 1 TO 20:PRINT CHR$(7);:NEX
      T i
20070 a$=INKEY$
20080 IF x = 1 THEN GOTO 3000
20090 IF a$="" THEN GOTO 20070
20100 IF x = 1 THEN GOTO 3000
20110 DATA 12,27,68,27,86,0
20120 RESTORE 20110:GOSUB 200
20130 x= REMAIN(2)
20140 a$ = " Vous etes en direct avec le
      Sysop "
20150 GOSUB 300
20160 DATA 31,66,65,0
20170 RESTORE 20160:GOSUB 200
20180 a$ = "Frappez votre texte puis <EN
      VOI>"
20190 GOSUB 300
20200 DATA 13,10,96,18,103,0
20210 RESTORE 20200:GOSUB 200
20220 PRINT:PRINT"C'est au minitelliste:
      "

```

```
20230 a$="C'est a "+nom$+": "+CHR$(13)+CHR$(10):GOSUB 300
20240 CALL &A000
20250 IF PEEK(&A02E) = 0 THEN RUN "debut
"
20260 IF PEEK(&A02F) = 19 AND PEEK (&A030) = 70 THEN GOTO 3000
20270 IF PEEK(&A02F) = 19 AND PEEK (&A030) = 71 THEN GOTO 20340
20280 IF PEEK(&A02F) = 19 AND PEEK (&A030) = 65 THEN GOTO 20370
20290 IF PEEK(&A02F) = 19 THEN GOTO 20240
20300 IF PEEK(&A030) < 32 THEN GOTO 20240
20310 a$ = CHR$(PEEK(&A030)):PRINT a$;
20320 GOSUB 300
20330 GOTO 20240
20340 a$=CHR$(8)+CHR$(32)+CHR$(8):GOSUB 300
20350 PRINT a$;
20360 GOTO 20240
20370 DATA 13,10,96,18,103,0
20380 RESTORE 20370:GOSUB 200
20390 a$="C'est au Sysop: "+CHR$(13)+CHR$(10):GOSUB 300
20400 PRINT:PRINT a$:
20410 a$=INKEY$
20420 IF a$ = "" THEN 20410
20430 IF ASC(a$) = 127 THEN a$=CHR$(8)+CHR$(32)+CHR$(8):PRINT a$;:GOSUB 300:GOTO 20410
20440 IF ASC(a$) = 13 THEN GOTO 20210
20450 PRINT a$;:GOSUB 300
20460 GOTO 20410
20470 x=REMAIN(2):x=1
20480 a$=CHR$(12)+"Desole, il ne semble pas etre present"
20490 GOSUB 300
20500 FOR i=1 TO 2000:NEXT i
20510 RETURN
```

Le programme ainsi frappé sera sauvegardé sous le nom :
SAVE "SYSOP.BAS".

- Ligne 20000 : Affichage de la demande de discussion avec le maître des lieux sur l'écran de l'Amstrad.
- Ligne 20010 : Initialisation d'une période d'attente de 1 000 fois 50 millisecondes = 50 secondes sur le chronomètre numéro 2, pour laisser le temps au Sysop de se placer devant le clavier de l'Amstrad. Si aucune touche du CPC n'est frappée durant ce temps, le programme se branche en ligne 20470. La variable **x** sert de drapeau pour signaler si une touche a été frappée.
- Lignes 20020 à 20050 : Affichage d'un message d'attente sur le minitel.
- Ligne 20060 : Emission d'un BIP d'appel sur le CPC.
- Lignes 20070 à 21000 : Attente de la touche frappée et test de l'état de la variable **x**. Si **x** = 1, cela signifie que le temps imparti est écoulé, et on retourne au menu en ligne 3000.
- Lignes 20110 à 20150 : Affichage sur le minitel de la présence du programmeur. De plus, le chronomètre est annulé par l'instruction :
x = REMAIN (2).
- Lignes 20160 à 20190 : Invitation à frapper le message.
- Lignes 21200 à 21230 : Affichage d'une ligne de séparation sur le minitel, et du nom du correspondant sur l'écran du CPC.
- Ligne 20240 : Attente de la frappe d'une touche.
- Lignes 20250 à 20330 : Test de la validité du caractère frappé. Si touche **<SOMMAIRE>** : retour au menu en ligne 3000. Si touche **<CORRECTION>** : envoi de la séquence d'effacement en ligne 20340. Si touche **<ENVOI>** : le programme passe la main au Sysop en ligne 20370. Pour toute autre touche de fonction (premier code égal à 19 décimal), ne pas tenir compte et attendre de nouveau un caractère. Si le caractère est un caractère ASCII visualisable, alors le renvoyer en écho pour affichage sur le minitel et l'imprimer sur l'écran du CPC, puis attente du caractère suivant.
- Lignes 20340 à 20360 : Traitement de la touche **<CORRECTION>** par envoi d'un caractère **BS** (Back Space = Retour Arrière, code 08), effacement par un Espace, puis retour arrière.
- Lignes 20370 à 20380 : Affichage d'une ligne de séparation.
- Lignes 20390 à 20400 : Affichage du passage de la « parole » au Sysop.
- Lignes 20410 à 20460 : Affichage sur l'écran du CPC et envoi au Minitel des caractères frappés sur le clavier de l'Amstrad, avec gestion de la touche **DEL** (test du code décimal 127 en ligne 20430), puis nouveau passage de « parole » au minitelliste lors de l'appui sur la touche **RETURN** (de code décimal 13, testé en ligne 20440).
- Lignes 20470 à 20510 : Sous-programme appelé lorsque le Sysop n'a pas répondu à l'appel du correspondant. Le chronomètre est inhibé, et l'absence signalée au minitelliste.

Le programme JEU

Ce programme permettra uniquement l'affichage du sous-menu JEUX, mais ne comportera pas les programmes de ces jeux. Ceux-ci seront développés séparément pour vous permettre de remplacer un de nos jeux proposés par un de votre propre création.

Un message de non activité des jeux est par contre prévu si le correspondant choisi un jeu non encore installé.

```

17000 PRINT"***** "+CHR$(24)+" JE
UX "+CHR$(24)+" *****"
17010 DATA 12,20,31,73,66,27,79,0,"J"
17020 DATA 31,75,66,27,79,0,"E"
17030 DATA 31,77,66,27,79,0,"U"
17040 DATA 31,79,66,27,79,0,"X"
17050 DATA 31,73,95,27,79,0,"J"
17060 DATA 31,75,95,27,79,0,"E"
17070 DATA 31,77,95,27,79,0,"U"
17080 DATA 31,79,95,27,79,0,"X"
17090 RESTORE 17010
17100 FOR j=1 TO 8:GOSUB 200:READ a$:GOS
UB 300:NEXT j
17110 DATA 31,72,70,14,27,68,96,112,112,
48,31,73,70,27,83,27,64,32,49,32,14,27,6
8,27,80,37,15,0," Pendu"
17120 RESTORE 17110
17130 GOSUB 200:READ a$:GOSUB 300
17140 DATA 31,78,70,14,27,68,96,112,112,
48,31,79,70,27,83,27,64,32,50,32,14,27,6
8,27,80,37,15,0," Gestion economique"
17150 RESTORE 17140:GOSUB 200
17160 READ a$:GOSUB 300
17170 CALL &A000:z=PEEK(&A030):z1=PEEK(&
A02F)
17180 IF z1=19 AND z=70 THEN GOTO 3000
17190 IF z1=19 THEN a$=CHR$(7):GOSUB 300
:GOTO 17170
17200 a$=CHR$(z)
17210 IF a$="1" THEN 18000
17220 IF a$="2" THEN 19000
17230 a$=CHR$(7):GOSUB 300:GOTO 17170
18000 DATA 12,31,75,68,27,85,32,18,96,31
,75,69,27,77,0
18010 RESTORE 18000:GOSUB 200
18020 a$="Ce choix n'est pas encore acti
f":GOSUB 300
18030 FOR i=1 TO 1000:NEXT i
18040 GOTO 17000
19000 DATA 12,31,75,68,27,85,32,18,96,31
,75,69,27,77,0

```

```

19010 RESTORE 19000:GOSUB 200
19020 a$="Ce choix n'est pas encore acti
f":GOSUB 300
19030 FOR i=1 TO 1000:NEXT i
19040 GOTO 17000

```

Vous sauvegarderez ce programme sous le nom :

SAVE "JEUX.BAS"

- Ligne 17000 : Signale au Sysop le choix effectué.
- Lignes 17010 à 17160 : Affichage du sous-menu des jeux, présentant deux choix :
 - Choix 1 : Le Pendu
 - Choix 2 : Gestion Economique
- Lignes 17170 à 17220 : Attente du choix et aiguillage correspondant si celui-ci est valide. La touche <SOMMAIRE> ramène au menu principal. Le choix 1 branche au numéro de ligne 18000, le choix numéro 2 branche en 19000. Vous connaissez ainsi les numéros de lignes auxquelles vous devez vous reporter si vous devez écrire un autre jeu.
- Lignes 18000 à 18040 et 19000 à 19040 : Signalent à l'utilisateur du serveur que les jeux ne sont pas encore installés, mais avec le programme suivant, votre premier jeu pourra être installé par MERGE "JEU1.BAS".

LE PROGRAMME JEU1

Ce premier jeu proposé est une adaptation minitelliste du célèbre jeu du PENDU.

Vous pourrez y modifier les mots proposés ainsi que leur nombre et même le nombre de tentatives d'essais de lettres.

```

18000 PRINT"***** "+CHR$(24)+" PEND
U "+CHR$(24)+" *****"
18010 I=0
18020 DATA BONJOUR, COLLEGE, PHYSIQUE, RECL
AMER, VERBALISER, ORDINATEUR, CODAGE, PROFES
SEUR, COULEUR
18030 DATA CARACTERE, THEOREME, VOIR, DEBUT
, INFORMATIQUE, PHOTOGRAPHIE, IMAGE, HABITUD
E, ELECTRIQUE
18040 DATA GYMNASIQUE, MUSCLE, ABONNER, ME
TTRE, MOUCHOIR, LEVRE, LAPIN, PINCEAU, TABLE
18050 DATA OBTENIR, LITTERATURE, CLASSEUR,
EDITION, SAVOIR, CHERCHER, TROUVER, FEINDRE,
NVIGATION

```

```

18060 DATA HOROSCOPE,HABITABILITE,HEROS,
RECONNAISSANCE,COMPATIBLE,SYMETRIQUE,ESC
LAVE,TRANSPARENT,SURCHARGE
18070 DATA 12,31,68,68,27,79,27,84,0," L
E PENDU "
18080 RESTORE 18070:GOSUB 200
18090 READ a$:GOSUB 300
18100 i=INT(RND*45)+1:RESTORE 18020
18110 FOR j=1 TO i:a=RND:READ S$:NEXT J
18120 d$=STRING$(LEN(s$),".")
18130 DATA 31,79,90,0,"Mot a trouver:"
18140 RESTORE 18130:GOSUB 200
18150 READ a$:GOSUB 300
18160 DATA 31,80,90,0
18170 RESTORE 18160:GOSUB 200
18180 a#=d$:GOSUB 300
18190 IF s#=d$ THEN 18720
18200 DATA 31,88,70,17,0,"Quelle lettre
proposes-tu?"
18210 RESTORE 18200:GOSUB 200
18220 READ a$:GOSUB 300
18230 CALL &A000:z=PEEK(&A030):z1=PEEK(&
A02F)
18240 IF z1=19 AND z=70 THEN GOTO 3000
18250 IF z1=19 THEN a#=CHR$(7):GOSUB 300
:GOTO 18230
18260 a#=UPPER$(CHR$(z))
18270 IF ASC(a#)<65 OR ASC(a#)>90 THEN a
#=CHR$(7):GOSUB 300:GOTO 18200
18280 m=0:FOR h=1 TO LEN(s$)
18290 IF MID$(s$,h,1)=a# THEN MID$(d$,h,
1)=a#:m=1
18300 NEXT h
18310 IF m=1 THEN GOTO 18130
18320 l=l+1
18330 ON l GOSUB 18480,18500,18520,18540
,18560,18580,18600,18620,18640,18660,186
80,18700
18340 IF l<>12 THEN 18130
18350 DATA 31,88,65,0,"DESOLE VOUS ETES
PENDU,LA SOLUTION ETAIT:"
18360 RESTORE 18350:GOSUB 200
18370 READ a$:GOSUB 300
18380 a#=s$:GOSUB 300
18390 DATA 10,13,0,"UNE AUTRE PARTIE(C o
u N)?"
18400 RESTORE 18390:GOSUB 200
18410 READ a$:GOSUB 300
18420 CALL &A000:z=PEEK(&A030):z1=PEEK(&
A02F)
18430 IF z1=19 AND z=70 THEN GOTO 3000

```

```
18440 IF z1=19 THEN a$=CHR$(7):GOSUB 300
:GOTO 18420
18450 a$=UPPER$(CHR$(z))
18460 IF a$="Q" THEN 18010
18470 GOTO 3000
18480 DATA 31,84,67,126,126,126,126,126,
126,0
18490 RESTORE 18480:GOSUB 200:RETURN
18500 DATA 31,83,68,47,11,47,92,10,92,0
18510 RESTORE 18500:GOSUB 200:RETURN
18520 DATA 31,81,69,125,8,11,125,8,11,12
5,8,11,125,8,11,125,8,11,125,0
18530 RESTORE 18520:GOSUB 200:RETURN
18540 DATA 31,75,70,95,95,95,95,95,95,0
18550 RESTORE 18540:GOSUB 200:RETURN
18560 DATA 31,77,70,47,11,47,0
18570 RESTORE 18560:GOSUB 200:RETURN
18580 DATA 31,76,75,124,0
18590 RESTORE 18580:GOSUB 200:RETURN
18600 DATA 31,77,75,79,0
18610 RESTORE 18600:GOSUB 200:RETURN
18620 DATA 31,78,75,127,10,8,127,0
18630 RESTORE 18620:GOSUB 200:RETURN
18640 DATA 31,78,74,47,8,8,10,14,34,0
18650 RESTORE 18640:GOSUB 200:RETURN
18660 DATA 31,78,76,92,10,14,33,0
18670 RESTORE 18660:GOSUB 200:RETURN
18680 DATA 31,80,74,47,8,8,10,14,35,0
18690 RESTORE 18680:GOSUB 200:RETURN
18700 DATA 31,80,76,92,10,14,35,0
18710 RESTORE 18700:GOSUB 200:RETURN
18720 DATA 31,88,65,0,"BRAVO - UNE AUTR
E PARTIE (O ou N)?"
18730 RESTORE 18720:GOSUB 200
18740 READ a$:GOSUB 300
18750 GOTO 18420
```

Nous vous conseillons de sauvegarder ce programme sous le nom : **SAVE "JEU1.BAS"**

- Ligne 18000 : Signale le choix PENDU effectué.
- Ligne 18010 : Initialisation de la variable comptabilisant le nombre d'essais à zéro.
- Lignes 18020 à 18060 : Liste des noms possibles dans le pendu.
- Lignes 18070 à 18090 : Affichage sur le minitel de l'entrée dans le jeu.
- Lignes 18100 à 18120 : Recherche au hasard d'un nom possible, mais dans la variable s\$, et comptabilisation du nombre de lettres pour initialiser la variable d\$ avec autant de points. Le nombre de mots disponibles peut être modifié en ligne 18100, en remplaçant la valeur 45

par une autre valeur, selon le nombre de mots supplémentaires que vous introduirez en lignes de DATA (Attention, la mémoire de votre CPC n'est pas extensible au point de contenir tout le dictionnaire).

— Lignes 18030 à 18180 : Affichage des points sur le minitel.

— Lignes 18190 à 18330 : Acquisitions de lettres au fur et à mesure de leur frappe sur le minitel, vérification de leur validité, de leur appartenance au mot, et test si le mot est trouvé. En cas d'erreur, branchement à une routine traçant les étapes de la pendaison.

— Lignes 18340 à 18470 : Si plus de douze essais se sont révélés infructueux, le logiciel donne la solution et propose une autre partie.

— Lignes 18480 à 18710 : Ces lignes assurent l'affichage sur le minitel des différentes étapes de la pendaison et sont organisées en sous-programmes. Si vous voulez modifier le nombre d'essais autorisé, il vous faudra modifier ce nombre en ligne 18340, annuler un branchement (ou plusieurs branchements, selon le nombre d'essais en moins) en ligne 18330, et retirer le nombre d'instructions **RETURN** correspondantes dans les lignes 18480 à 18710.

— Lignes 18720 à 18750 : Félicitations du serveur, et proposition d'une autre partie, en cas de réussite au jeu.

Le programme de présentation du serveur

Cette rubrique, qui comporte les informations relatives à la conception du serveur, est élaborée très simplement par une succession de directives d'affichage et de textes, envoyés vers le minitel.

Le programme est qualifié de séquentiel : aucun test n'apparaît dans ce programme. Le correspondant est passif à la lecture du message, la seule initiative qui lui est laissée est de frapper sur une touche quelconque afin de remonter au menu, dès qu'il a terminé sa consultation.

Nous ne vous soumettons pas ici d'algorithme ni d'algorithme, à cause de la simplicité du programme.

Son insertion dans le serveur s'effectuera par l'instruction :

MERGE "PRESENTA"

s'il a été sauvegardé sous ce nom.

```

11000 PRINT"*****"+CHR$(24)+" PRESENT
ATION DU SERVEUR "+CHR$(24)+"*****"
11010 DATA 12,20,31,64,74,27,78,27,66,0,
" MATERIEL "
11020 RESTORE 11010
11030 GOSUB 200:READ a$:GOSUB 300
11040 DATA 31,67,65,27,84,32,18,103,31,6
7,71,27,77,, "Avis aux connaisseurs"
11050 DATA 31,69,65,0, "Ce serveur tourne
sur un Amstrad CPC6128 dont la memoire n
'est pas utilisee au maximum de ses po
ssibilites(m
ais nous y penserons), et fut conçu grac
e a la
11060 DATA 31,73,65,0, "revue: 'Comment ex
ploiter toutes les res-sources de votre
AMSTRAD' edite par WEKA"
11070 DATA 31,76,65,0, "La configuratin c
omplete comprend donc:"
11080 DATA 31,77,69,0, "-Un amstrad CPC 6
128"
11090 DATA 31,78,69,0, "-Un modem Digital
ec DTL2000+"
11100 DATA 31,79,69,0, "-Un detecteur de
sonnerie special"
11110 DATA 31,80,70,0, "WEKA"
11120 DATA 31,83,65,27,84,32,18,103,31,8
3,77,27,77,0, "Et c'est TOUT !"
11130 DATA 31,85,75,27,67,27,93,27,72,0,
" Frappez une touche "
11140 RESTORE 11040
11150 FOR x = 1 TO 10
11160 GOSUB 200:READ a$:GOSUB 300
11170 NEXT x
11180 LABEL &A000
11190 DATA 31,64,65,32,18,101,0
11200 RESTORE 11190
11210 GOSUB 200
11220 GOTO 3000

```

— Ligne 11000 : affichage sur l'écran du CPC de la rubrique où se trouve le correspondant.

— Lignes 11010 à 11130 : directives d'affichage et textes à afficher.

— Lignes 11140 à 11170 : programme permettant d'afficher les dix directives et leurs textes associés.

— Ligne 11180 : attente d'une touche frappée.

— Lignes 11190 à 11210 : effacement de la ligne 0 du minitel avant de retourner au menu en ligne 11220.

Le programme des Boîtes Aux Lettres (BAL)

Nous allons maintenant nous intéresser à la partie la plus délicate de notre serveur : les boîtes aux lettres.

ORGANISATION DES BAL

Avant de nous lancer dans la programmation de la gestion des boîtes aux lettres, il nous faut réfléchir à leur organisation.

Nous devrons pour cela posséder un lecteur de disquette, qui servira de BUREAU DE POSTE et effectuera le tri du courrier en POSTE RESTANTE.

Les boîtes aux lettres seront donc des fichiers au nom des correspondants qui se trouveront sur la disquette.

La capacité de la disquette étant limitée, il nous faudra limiter le nombre d'inscrits (certains petits malins seraient capables de saturer votre disquette en ouvrant sciemment un nombre très grand de boîtes aux lettres), ainsi que le nombre de messages dans une même boîte.

Nous limiterons manuellement le nombre d'inscrits, le programmeur ouvrira la boîte aux lettres grâce à un programme que nous vous fournirons plus loin. Le nombre maximal de messages sera arbitrairement à 3 messages de 240 caractères, mais vous pourrez modifier ce nombre.

Lorsqu'un minitelliste veut lire sa boîte aux lettres, il faut qu'il indique son nom, et c'est le programme qui recherche le fichier sur la disquette en l'ouvrant en lecture.

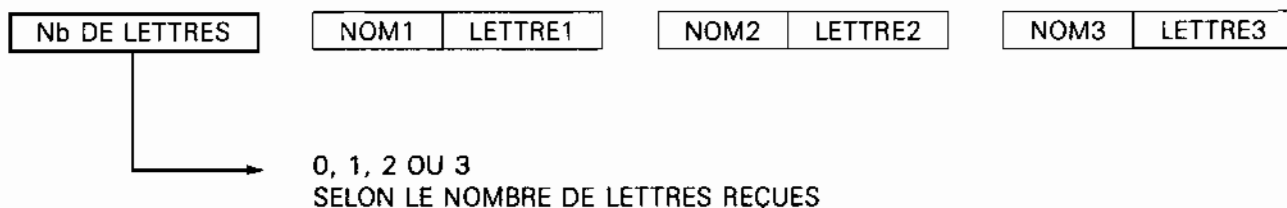
Si ce fichier n'existe pas, votre AMSTRAD indiquera une erreur disquette et s'arrêtera, ce qui serait préjudiciable à la popularité de votre serveur. Nous aurons donc besoin d'un fichier comportant tous les noms des inscrits afin d'effectuer les vérifications d'inscription, et nous y ajouterons les codes d'accès qui permettront d'ouvrir la boîte aux lettres.

Ce fichier sera un fichier séquentiel qui se présentera sous la forme ci-dessous :

NOM1	CODE1	NOM2	CODE2	NOM3	CODE3		NOMn-1	CODEn-1	NOMn	CODEn
------	-------	------	-------	------	-------	-------	--------	---------	------	-------

Nous trouverons les noms suivis du code correspondant, inscrit l'un après l'autre.

Chacun des noms inscrit aura son fichier propre qui se présentera sous la forme :



La première donnée du fichier sera le nombre de lettres inscrites. En fonction de ce nombre (compris entre 0 et 3), il y aura ensuite les différentes lettres, composées de deux champs : le nom de l'expéditeur, et sa lettre.

ORGANISATION DU SERVICE DES BAL

Ce service comportera quatre rubriques :

- l'ouverture d'une boîte aux lettres qui se traduira par l'acquisition d'un message serveur précisant les NOM CODE et MOTIVATION, qui seront lus par le SYSOP. L'ouverture de la boîte aux lettres s'effectuera ainsi ultérieurement, selon votre bon vouloir. Nous aurons ainsi besoin d'un programme de lecture des messages reçus dans ce fichier, ainsi que d'un programme de création de BAL ;
- la visualisation de l'annuaire afin d'écrire à des minitellistes effectivement inscrits ;
- la possibilité de lire éventuellement son courrier ;
- la possibilité d'écrire à un minitelliste inscrit.

LE PROGRAMME DE LECTURE SERVEUR

Nous devons auparavant créer le fichier dans lequel seront inscrites les demandes d'ouverture de BAL.

Pour cela vous taperez en mode direct la séquence d'instructions suivantes :

```
OPENOUT "MESSAGE."  
WRITE #9," "  
CLOSEOUT
```

Le fichier étant créé, vous pourrez ultérieurement le consulter grâce au programme dont l'algorithme suit :

- DEBUT
 - Effacer l'écran en mode 2
 - Ouvrir le fichier « MESSAGE. » en lecture
 - Lire la première chaîne de caractères non utilisée
 - TANTQUE la fin du fichier n'est pas atteinte
 - Afficher une ligne de séparation
 - Lire le NOM
 - Afficher le NOM
 - Lire le CODE
 - Afficher le CODE
 - Lire le message
 - Afficher le message
 - Attendre l'appui d'une touche
- FINTANTQUE
- Fermer le fichier en lecture
- Ouvrir le fichier « MESSAGE » en écriture
- Sauver le message " "
- CO
 - le fichier est ré-initialisé
- FINCO
- Fermer le fichier en écriture
- FIN

```
10 REM ***** LECTURE DES MESSAGES
   DU SERVEUR *****
20 REM *****
   *****
30 REM
40 MODE 2
50 PRINT "LECTURE DU FICHIER MESSAGES"
60 OPENIN "message."
70 REM *** lecture ***
80 INPUT#9,a$
90 WHILE NOT EOF
100   PRINT:PRINT
105   PRINT STRING$(80,"-")
106   PRINT
110   INPUT#9,a$
120   PRINT "NOM: "+a$
130   PRINT
140   INPUT#9,a$
150   PRINT "CODE: "+a$
160   PRINT
170   INPUT#9,a$
180   PRINT "MESSAGE:"
190   PRINT a$
200   REM *** attente pour message suiv
   ant ***
210   a$=INKEY$
220   IF a$="" THEN 210
230 WEND
240 CLOSEIN
250 CLS
260 REM *** effacement du fichier ***
270 PRINT CHR$(24)+"EFFACEMENT DU FICHIE
R"+CHR$(24)+" en cours ..."
280 OPENOUT "message."
290 WRITE#9,""
300 CLOSEOUT
310 END
```

— Ligne 80 : lecture de la première chaîne de caractères, non significative.

— Lignes 90 à 230 : boucle de lecture du fichier messages jusqu'à sa ligne.

— Lignes 260 à 300 : réinitialisation du fichier.

Le programme sera sauvegardé sous le nom

SAVE "LECTURE.BAS"

et sera utilisé régulièrement afin d'ouvrir les éventuelles boîtes aux lettres.

LE PROGRAMME DE CRÉATION DE BAL

Après avoir consulté ses messages, le SYSOP doit créer les boîtes aux lettres réclamées. La mise à jour du fichier NOM.BAL, qui comportera les noms des inscrits à l'annuaire, et la création du fichier au nom du correspondant s'effectueront par le programme dont l'algorithme est décrit ci-après :

- DEBUT
 - REPETER
 - Effacer l'écran de l'AMSTRAD en mode 2
 - Demander le NOM du nouvel inscrit
 - Demander le CODE
 - Demander si le fichier NOM.BAL existe déjà
 - Ouvrir un fichier temporaire en écriture
 - SI NOM.BAL existe
 - Ouvrir le fichier NOM.BAL en lecture
 - TANTQUE la fin du fichier n'est pas atteinte
 - Lire une chaîne de caractères dans NOM.BAL
 - Ecrire cette chaîne dans le fichier temporaire
 - FINTANTQUE
 - Fermer le fichier NOM.BAL
 - Effacer le fichier NOM.BAL
 - FINSI
 - Ecrire le nouveau NOM dans le fichier temporaire
 - Ecrire le CODE correspondant
 - Fermer le fichier temporaire en écriture
 - Renommer le fichier temporaire en fichier NOM.BAL
 - Ouvrir un fichier au nouveau NOM en écriture
 - Ecrire le chiffre « 0 »
 - Fermer le fichier en écriture
 - JUSQU'A ce qu'il n'y ait plus de nom à entrer
- FIN

Le programme correspondant est donné par le listing ci-après :

```
10 REM ***** CREATION D'UNE BOITE A
UX LETTRES *****
20 REM *****
*****
30 MODE 2
40 REM *** acquisition coordonnees de **
*
50 REM *** la boîte aux lettres a ouvrir
***
60 CLS
70 PRINT CHR$(24)+"ATTENTION"+CHR$(24)+"
: les coordonnees de votre correspondant
ne doivent pas etre deja existante
s sous peine
d'une erreur disquette irremediable"
80 PRINT
90 REM *** securite ***
100 INPUT "Desirez vous continuer";reponse$
110 reponse$=LOWER$(reponse$)
120 IF reponse$ = "n" THEN STOP
130 INPUT "NOM de la boîte aux lettres a
ouvrir";nom$
140 nom$=LOWER$(nom$)
150 INPUT "CODE de la boîte aux lettres
a ouvrir";code$
160 code$=LOWER$(code$)
170 REM *****
180 INPUT "le fichier NOM.BAL existe il
deja";reponse$
190 reponse$=LOWER$(reponse$)
200 OPENOUT "temp."
210 IF reponse$ = "n" THEN 320
220 REM *** fichier deja cree ***
230 OPENIN "NOM.BAL"
240 REM *** sauvegarde temporaire des
noms et codes ***
250 WHILE NOT EOF
260 INPUT#9,a$
270 WRITE#9,a$
280 WEND
290 CLOSEIN
300 OPEN "NOM.BAL"
310 REM *** sauvegarde nouvelles coordon
nees ***
320 WRITE#9,nom$
330 WRITE#9,code$
340 CLOSEOUT
```

```

350 REM *** recuperation du fichier NOM.
BAL ***
360 GREN,"NOM.BAL","TEMP."
370 OPENOUT nom$
380 WRITE#9,0
390 CLOSEOUT
400 INPUT "Faut-il creer une autre Boite
    aux Lettres";reponse$
410 reponse$=LOWER$(reponse$)
420 REM *** autre traitement ***
430 IF reponse$ = "o" THEN 60
440 END

```

Le programme ainsi entré pourra être sauvé sous le nom

SAVE "GESBAL.BAS"

- Lignes 40 à 120 : prévention d'une éventuelle erreur d'inscription.
- Lignes 130 à 160 : entrée des coordonnées de la boîte aux lettres.
- Lignes 170 à 190 : faudra-t-il créer le fichier NOM.BAL ou l'utiliser ?
- Lignes 200 à 300 : sauvegarde des noms et codes déjà existant si le fichier existait déjà.
- Lignes 310 à 340 : sauvegarde des nouvelles coordonnées.
- Ligne 360 : récupération du fichier NOM.BAL en le renommant.
- Lignes 370 à 390 : création du fichier boîte aux lettres du minitelliste.
- Lignes 400 à fin : éventuelle création d'une autre BAL.

LE PROGRAMME DU MENU DES BAL

Les différents programmes de gestion des boîtes aux lettres étant maintenant conçus, nous pouvons maintenant créer la rubrique du serveur gérant le menu des boîtes aux lettres.

Cette partie du programme du serveur devra être étudiée avec soin, car relativement complexe et sujette à d'éventuelles erreurs, c'est pourquoi nous donnerons, avant les programmes, les algorithmes de ceux-ci, et leurs algorigrammes associés.

L'étude des algorithmes et algorigrammes vous permettra de modifier cette partie du serveur si vous désirez y apporter des améliorations. Rappelons tout de même les raisons qui nous ont inspiré la réflexion qui suivra (nous pouvons appeler ces raisons : *LE CAHIER DES CHARGES*).

Nous n'avons d'abord pas désiré que quiconque puisse ouvrir de lui-même une boîte aux lettres, mais donne sa motivation au programmeur pour l'ouverture de celle-ci ; la place nous étant comptée sur la disquette, nous avons réduit le nombre de messages à 3.

Après l'affichage des différents choix de ce menu, il nous faudra traiter chacun des choix, nous rencontrons ces contraintes dans l'algorithme suivant :

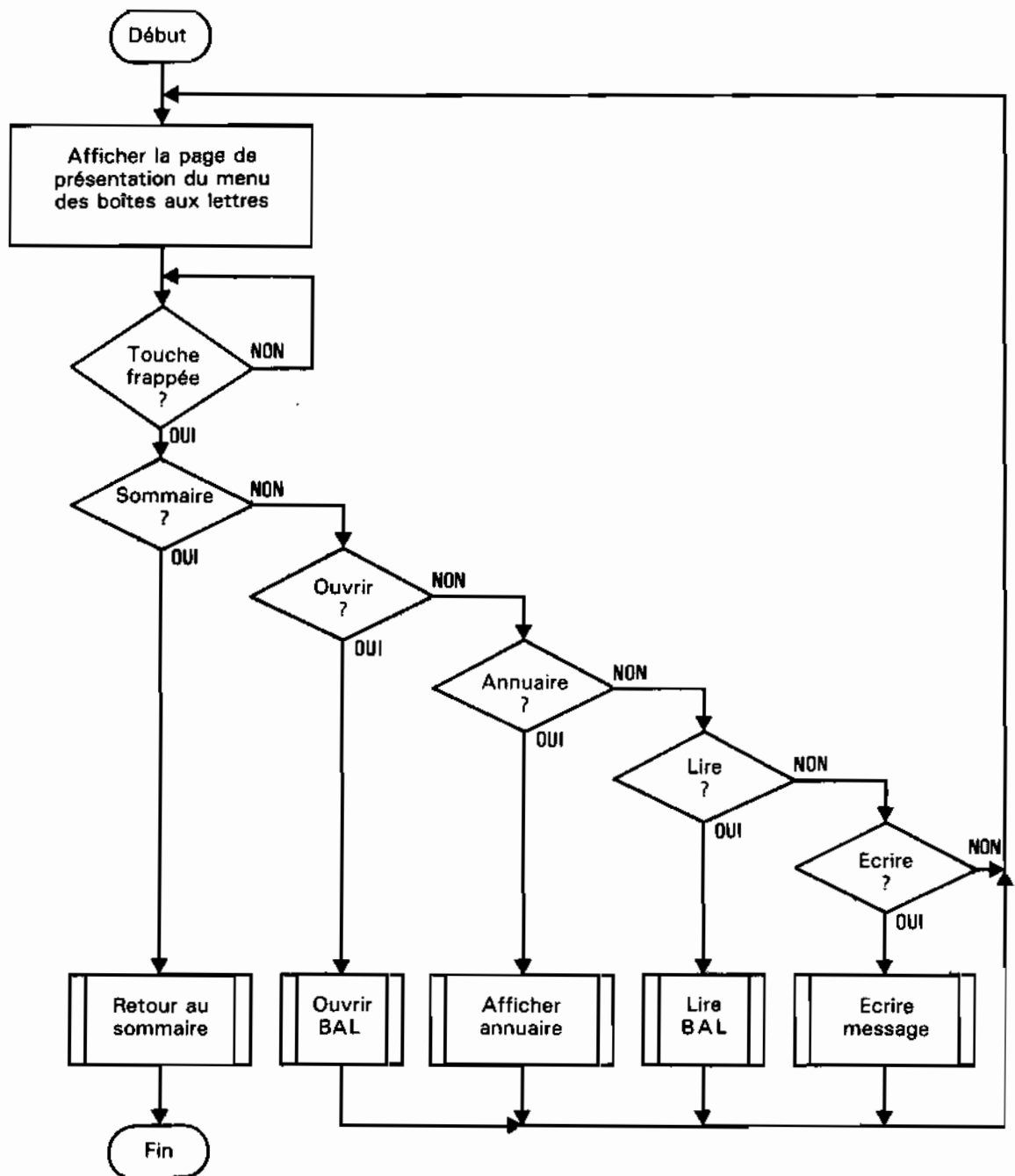
- DEBUT
 - Afficher la page de présentation du service des Boîtes Aux Lettres
 - REPETER
 - Attendre la frappe d'une touche indiquant le choix
 - SELONCAS
 - Cas : Choix SOMMAIRE
 - Remonter au menu du SOMMAIRE
 - CO

On sort de la boucle des boîtes aux lettres
 - FINCO
 - Cas : Choix INSCRIPTION
 - PROCEDURE INSCRIPTION
 - Cas : Choix ANNUAIRE
 - PROCEDURE ANNUAIRE
 - Cas : Choix LIRE son courrier
 - PROCEDURE LECTURE DE MESSAGES
 - Cas : Choix Ecriture d'un message
 - PROCEDURE ECRITURE D'UN MESSAGE
 - FINCAS
 - JUSQU'A ce que le choix soit valide
- FIN

Nous retrouvons dans cet algorithme une structure un peu particulière : **SELONCAS ... Cas1,Cas2,...Casn...FINCAS.**

Cette structure se présente dans les langages évolués tel Turbo-Pascal (la syntaxe étant : **CASE ... OF ... END**), et, même si elle n'existe pas en Basic, nous tenons à la faire apparaître ainsi, afin de conserver un maximum de rigueur dans la réflexion (vous pourrez même l'adapter quasi directement si vous décidez d'écrire le programme en Turbo-Pascal).

Nous simulerons cette structure de l'algorithmique par une série de tests sur les différents cas comme décrits par l'algorithme suivant :



Grâce à ces algorithmes et algorithmes, nous pouvons écrire sans problème le programme ci-dessous :

```

8000 PRINT"*****"+CHR$(24)+"    LES B.
A.L.s    "+CHR$(24)+"*****"
8010 REM caractere L
8020 DATA 12,31,66,71,47,10,8,8,47,10,8,
8,47,10,8,8,47,10,8,8,47,10,8,126,126,12
6,126,126
8030 REM caracteres es
8040 DATA 11,32,27,79,101,115,32,27,76
8050 REM caractere B
8060 DATA 47,11,47,11,47,11,47,11,47,126
,126,126,123,10,8,8,47,10,8,123,10,8,123
,10,8,8,47,10,8,8,8,8,8,8,126,126,126,
126,126,126,
11,11,11,8,8,8,126,126,126,126,10,10,9
8070 REM caractere A
8080 DATA 47,11,47,11,47,11,47,11,47,123
,10,8,123,10,8,123,10,8,123,10,8,123,11,
8,8,8,8,126,126,126,11,11,11,9,9,9,9,9,9
8090 REM caractere L
8100 DATA 47,10,8,8,47,10,8,8,47,10,8,8,
47,10,8,8,47,10,8,126,126,126,126,126
8110 DATA 11,9,27,79,115,0
8120 RESTORE 8020:GOSUB 200
8130 DATA 31,75,69,0,"Creer ma boite",32
,27,72,25,46,32,27,66,27,93,27,73,67,0
8140 DATA 31,77,76,0,"Annuaire boite",27
,72,27,93,27,64,27,87,32,25,46,27,92,27,
71,27,80,32,27,66,27,93,27,73,65,0
8150 DATA 31,81,68,0,"Ecrire en boite",3
2,27,72,25,46,32,27,66,27,93,27,73,69,0
8160 DATA 31,79,77,0,"Lire ma boite",27,
72,27,93,27,64,27,87,32,25,46,27,92,27,7
1,27,80,32,27,66,27,93,27,73,76,0
8170 RESTORE 8130
8180 FOR x = 1 TO 4
8190 GOSUB 200:READ a$:GOSUB 300:GOSUB 2
00
8200 NEXT x
8210 DATA 31,74,66,27,72,126,18,95,92,10
,8,125,10,8,125,10,8,125,10,8,125,10,8,1
25,10,8,125,10,8,125,10,8,47
8220 DATA 31,74,65,27,72,27,72,47,10,8,1
23,10,8,123,10,8,123,10,8,123,10,8,123,1
0,8,123,10,8,123,10,8,92
8230 DATA 95,18,95,0
8240 RESTORE 8210
8250 GOSUB 200

```

```

B260 CALL &A000:z=PEEK(&A030):z1=PEEK(&A
02F)
B270 IF z1=19 AND z=70 THEN GOTO 3000
B280 IF z1=19 THEN a$=CHR$(7):GOSUB 300:
GOTO B260
B290 a$=CHR$(z)
B300 IF a$="C" OR a$="c" THEN B350
B310 IF a$="A" OR a$="a" THEN B970
B320 IF a$="L" OR a$="l" THEN 9210
B330 IF a$="E" OR a$="e" THEN 9940
B340 a$=CHR$(7):GOSUB B340:GOTO B260

```

— Lignes 8000 à 8160 : (attention à ne pas confondre les nombres en ligne de DATAs avec les numéros de lignes), données de directives d'affichage et textes associés pour le minitel.

— Lignes 8170 à 8200 : transmission de ces données à l'aide des sous-programmes précédemment étudiés.

— Lignes 8210 à 8250 : autre directive d'affichage émise.

— Ligne 8260 : attente d'un choix.

— Ligne 8270 : retour au MENU général en cas d'appui sur **SOMMAIRE**.

— Lignes 8280 à 8340 : traitement des mauvais choix éventuels et des choix permis. On retrouve ici notre structure SELON CAS.

On veillera à sauvegarder provisoirement ce programme sous le nom :

SAVE "BAL.BAS"

que l'on réutilisera pour les 4 programmes qui vont suivre.

LE PROGRAMME D'INSCRIPTION EN BAL

Rappelons que le programme d'inscription est en fait l'accès à la boîte aux lettres de programmeur (donc vous-même) dans laquelle le miniteliste pourra laisser ses coordonnées et un message exposant ses motivations à utiliser votre serveur.

Dans la structure de l'algorithme nous appellerons cette partie du programme des boîtes aux lettres une PROCEDURE, bien que n'étant pas un sous-programme, mais se comportant comme tel, puisque retournant au menu des boîtes aux lettres, en fin de traitement.

— DEBUT

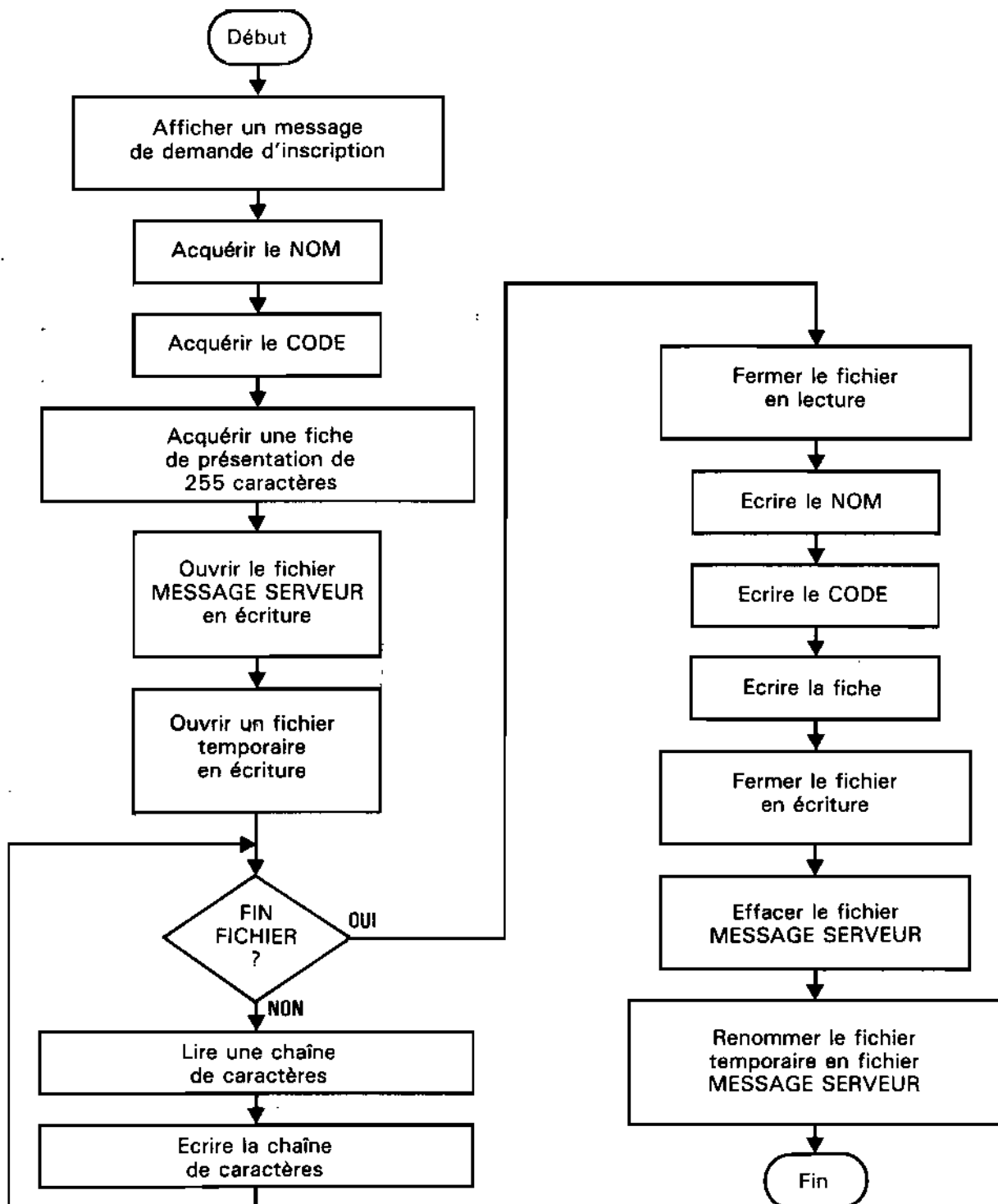
— Afficher le message de demande d'inscription

— Acquérir le NOM

— Acquérir le CODE

- Acquérir une fiche de présentation de 255 caractères
- Afficher un message invitant à patienter
- Ouvrir le fichier des messages SERVEUR en lecture
- Ouvrir un fichier temporaire en écriture
- TANTQUE la fin du fichier SERVEUR n'est pas atteinte
 - Lire une chaîne de caractères (dans le fichier messages SERVEUR)
 - Ecrire la chaîne de caractères (dans le fichier temporaire)
- FINTANTQUE
- Fermer le fichier en lecture (fichier SERVEUR)
- Ecrire le nom acquis
- Ecrire le code acquis
- Ecrire la fiche de présentation
- Fermer le fichier en écriture (fichier temporaire)
- Effacer le fichier messages SERVEUR
- Renommer le fichier temporaire en fichier Messages SERVEUR
- FIN

La boucle **TANTQUE ... FINTANTQUE** sert à sauvegarder les coordonnées et messages précédents avant de sauver les nouvelles coordonnées et messages.



Après étude de l'algorithme et de l'algorithme nous pouvons écrire le programme d'inscription suivant, qui sera à frapper suite au programme dénommé BAL.BAS :

```
8350 PRINT"*****"+CHR$(24)+"INSCRIPTI
ON"+CHR$(24)+"*****"
8360 DATA 12,0
8370 DATA 31,69,75,27,77,27,69,27,90,32,
0,"CREATION D'UNE B.A.L.",27,89,32,0
8380 DATA 31,69,75,27,77,27,70,27,90,32,
0,"CREATION D'UNE B.A.L.",27,89,32,0
8390 RESTORE 8350:GOSUB 200
8400 FOR x = 1 TO 4
8410 RESTORE 8370
8420 GOSUB 200:READ a$:GOSUB 300:GOSUB 2
00
8430 GOSUB 200:READ a$:GOSUB 300:GOSUB 2
00
8440 NEXT x
8450 DATA 31,71,65,0,"Si vous desirez vo
us inscrire en BAL, il suffit de laiss
er ci-dessous vos coordonnees..."
8460 DATA 31,82,65,0,"Dans quelques jour
s vous pourrez venir sur le serveur ave
c votre nouveau nom, dans la mesure o
u il est pos
sible de vous inscrire"
8470 DATA 31,75,65,0,"Nom ou Pseudonyme:
"
8480 DATA 95,18,71,0,"+"
8490 DATA 27,93,0,"ENVOI"
8500 DATA 31,75,84,0
8510 FOR x = 1 TO 5
8520 GOSUB 200:READ a$:GOSUB 300
8530 NEXT x
8540 GOSUB 200
8550 x=8:GOSUB 400
8560 nom$=texte$
8570 DATA 31,77,65,0,"Code: "
8580 DATA 95,18,69,0,"+"
8590 DATA 27,93,0,"ENVOI"
8600 RESTORE 8570
8610 DATA 31,77,71,0
8620 RESTORE 8570
8630 FOR x = 1 TO 3
8640 GOSUB 200:READ a$:GOSUB 300
8650 NEXT x
8660 code$=texte$
8670 GOSUB 200
8680 x=6:GOSUB 400
8690 code$=texte$
```

```

8700 DATA 31,79,65,0,"Fiche de presentat
ion:"
8710 RESTORE 8670
8720 DATA 31,80,65,95,18,127,18,127,18,1
27,18,114,0,"+"
8730 DATA 27,93,0,"ENVOI"
8740 DATA 31,80,65,0
8750 RESTORE 8700
8760 FOR x = 1 TO 3
8770 GOSUB 200:READ a$:GOSUB 300
8780 NEXT x
8790 GOSUB 200
8800 x=240:GOSUB 400
8810 DATA 31,88,75,27,72,0,"Patience ...
"
8820 RESTORE 8810
8830 GOSUB 200:READ a$:GOSUB 300
8840 OPENIN "message."
8850 OPENOUT "temp."
8860 WHILE NOT EOF
8870 INPUT#9,a$
8880 PRINT#9,a$
8890 WEND
8900 CLOSEIN
8910 WRITE#9,nom$,code$,texte$
8920 CLOSEOUT
8930 OPEN "message."
8940 OPEN "message.", "temp."
8950 GOTO 8000

```

- Lignes 8350 à 8440 : présentation du choix inscription.
- Lignes 8450 à 8460 : message d'information sur l'inscription.
- Lignes 8470 à 8560 : acquisition du NOM.
- Lignes 8570 à 8690 : acquisition CODE.
- Lignes 8700 à 8800 : acquisition du message de motivation qui reste dans la variable TEXTE afin d'économiser une variable, donc de la place mémoire.
- Lignes 8810 à 8830 : affichage d'un message invitant à patienter.
- Lignes 8840 à 8900 : sauvegarde temporaire des précédentes coordonnées et précédents messages.
- Lignes 8910 et 8920 : sauvegarde des nouvelles coordonnées.
- Lignes 8930 et 8940 : on renomme le fichier MESSAGE.
- Ligne 8950 : retour au menu des boîtes aux lettres.

LE PROGRAMME ANNUAIRE

Nous allons maintenant définir le programme permettant de donner l'annuaire des noms inscrits en BAL afin, pour le minitelliste, de vérifier l'existence d'un nom, avant d'établir une correspondance.

Nous n'avons pas établi d'algorithme ni d'algorithme pour ce programme. Sa structure est essentiellement séquentielle, il comporte une unique boucle de type **TANTQUE ... FINTANTQUE** permettant de lire le fichier NOM.BAL.

Par contre nous insisterons sur une fonction mathématique particulière utilisée : **MODULO** traduite en Basic par l'instruction **MOD**.

Dans notre programme ANNUAIRE, nous ne pourrions certainement pas inscrire l'annuaire complet après quelques jours de fonctionnement du serveur. Nous serons obligé de l'inscrire par page de plusieurs noms, par exemple 17 noms affichés au maximum dans chaque page. Cela veut dire qu'arrivé au dix-huitième nom, il faudra changer de page, ainsi que pour tous les multiples de 18.

Nous allons pour cela utiliser la fonction **MODULO**, qui opère entre deux nombres et donne comme résultat le reste de la division entière des deux nombres (on n'effectue pas le calcul après la virgule).

Exemple :

15 MOD 6 nous donne 3 car 15 divisé par 6 est égal à 2 et il reste 3.

Nous reconnaitrons ainsi un multiple A de 18 car le résultat de l'opération A MOD 18 donnera zéro.

Il nous suffira de tester ce résultat pour savoir si l'on doit changer de page ou non.

```

8970 PRINT "*****"+CHR$(24)+"ANNUAIRE"
+CHR$(24)+"*****"
8980 DATA 12,31,66,70,27,78,27,69,27,93,
0," * ANNUAIRE * "
8990 RESTORE 8980
9000 GOSUB 200:READ a$:GOSUB 300
9010 OPENIN "nom.bal"
9020 k=1
9030 WHILE NOT EOF
9040 INPUT#9,nomi$,codei$
9050 a$=CHR$(31)+CHR$(67+(k MOD 18))+CHR
$(67)+" ("+"RIGHT$( "..."+STR$(k),3)+" "
+CHR$(27)+CHR$(69)+CHR$(27)+CHR$(93)+"-"
+CHR$(27)+CH
R$(92)+nomi$+SPACE$(8-LEN(nomi$))+CHR$(2
7)+CHR$(93)+"-"
9060 GOSUB 300
9070 k=k+1
9080 IF (k MOD 18) <> 0 THEN 9150

```

```
9090 DATA 31,88,72,27,67,27,93,27,72,0,"
      Frapper une touche "
9100 RESTORE 9090
9110 GOSUB 200:READ a$:GOSUB 300
9120 CALL &A000
9130 RESTORE 8980
9140 GOSUB 200:READ a$:GOSUB 300
9150 WEND
9160 CLOSEIN
9170 RESTORE 9090
9180 GOSUB 200:READ a$:GOSUB 300
9190 CALL &A000
9200 GOTO 8000
```

- Lignes 8970 à 9000 : présentation du choix ANNUAIRE.
- Ligne 9010 : ouverture du fichier NOM.BAL en lecture.
- Ligne 9020 : initialisation de la variable de comptage et de positionnement à 1.
- Ligne 9030 : début de la boucle de lecture du fichier, associée avec le WEND de la ligne 9150.
- Ligne 9040 : lecture du NOM et du CODE d'un inscrit. Seul le nom sera retenu par la suite.
- Ligne 9050 : création de la variable d'affichage. Cette variable comporte toutes les coordonnées de positionnement du nom, en fonction de la variable k et de sa valeur par rapport à un multiple de 18, ainsi que le NOM d'un inscrit.
- Ligne 9060 : affichage de cette variable.
- Ligne 9070 : incrémentation de la variable k.
- Ligne 9080 : test de la valeur de k pour voir s'il faut changer de page.
- Lignes 9090 à 9120 : attente de la frappe d'une touche au cas où il faudrait changer de page.
- Lignes 9130 et 9140 : page suivante.
- Ligne 9160 : fermeture du fichier en lecture.
- Lignes 9170 à 9190 : attente de la frappe d'une touche avant de remonter au menu.
- Ligne 9200 : on remonte au menu des BAL.

LE PROGRAMME DE LECTURE EN BAL

Chaque inscrit, possédant donc une boîte aux lettres, pourra relire celle-ci grâce à cette partie du serveur.

Le programme, connaissant le NOM acquis lors de l'arrivée sur le serveur, vérifiera sa présence dans le fichier NOM.BAL, et demandera le CODE permettant l'accès à la lecture de la boîte.

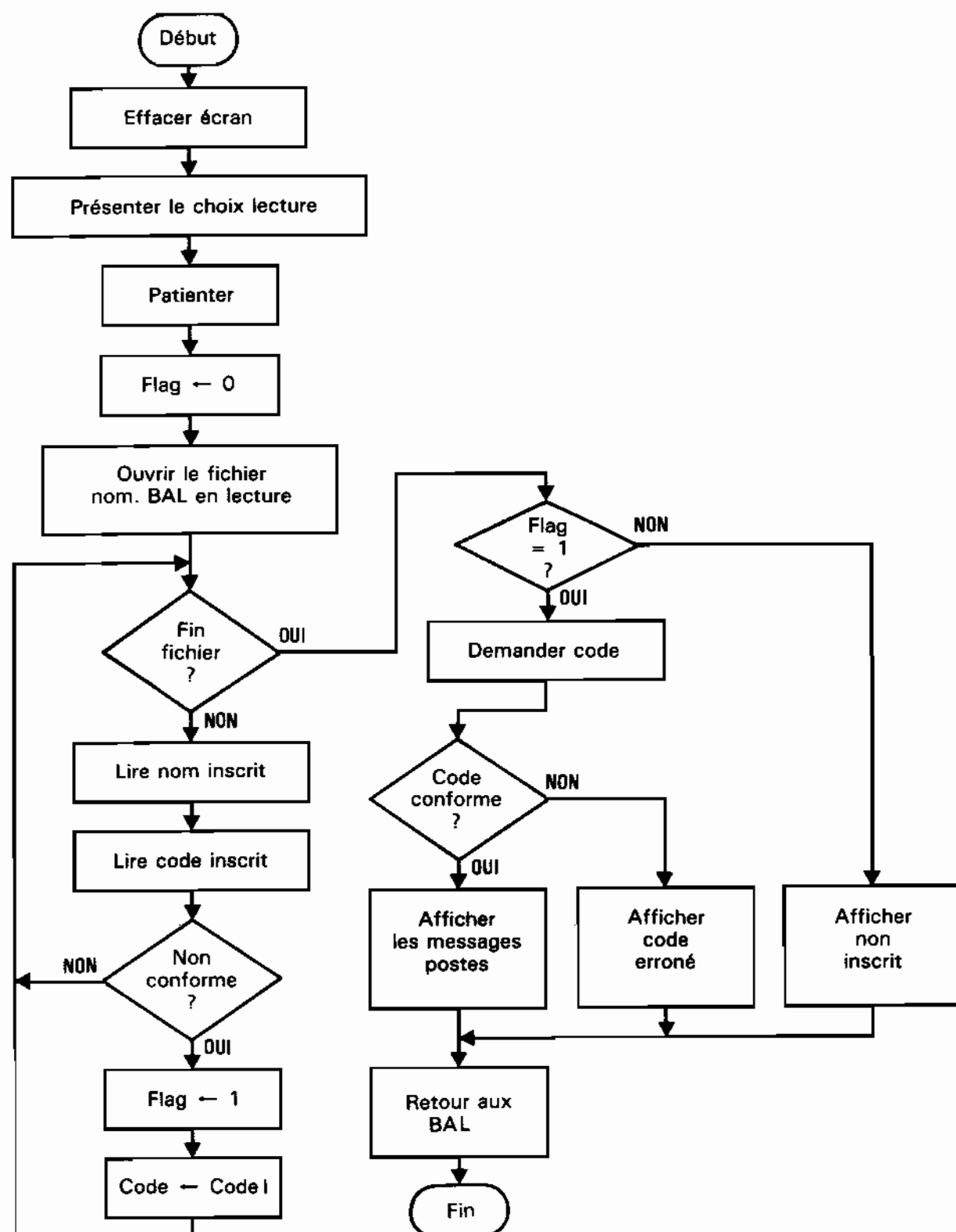
Une fois les messages affichés, le programme devra réinitialiser la boîte aux lettres, c'est-à-dire effacer les messages pour permettre d'autres enregistrements.

Vous trouverez ci-après l'algorithme de traitement de cette partie du programme.

— DEBUT

- Effacer l'écran
- Afficher la page de présentation en lecture avec le nom du correspondant
- Afficher un message invitant à patienter durant les vérifications
- Positionner un « flag » à zéro
- Ouvrir le fichier des noms (NOM.BAL) en lecture
- TANT QUE la fin du fichier n'est pas atteinte
 - Lire un nom inscrit
 - Lire le code correspondant
 - SI le nom lu est identique au nom du correspondant
 - ALORS
 - Positionner le « flag » à 1
 - Transférer le code inscrit (CODEI)
- FINSI
- FINTANTQUE
- Fermer le fichier
- SI le « flag » est à « 1 »
 - ALORS
 - Demander le code de vérification
 - SI le code est conforme
 - ALORS

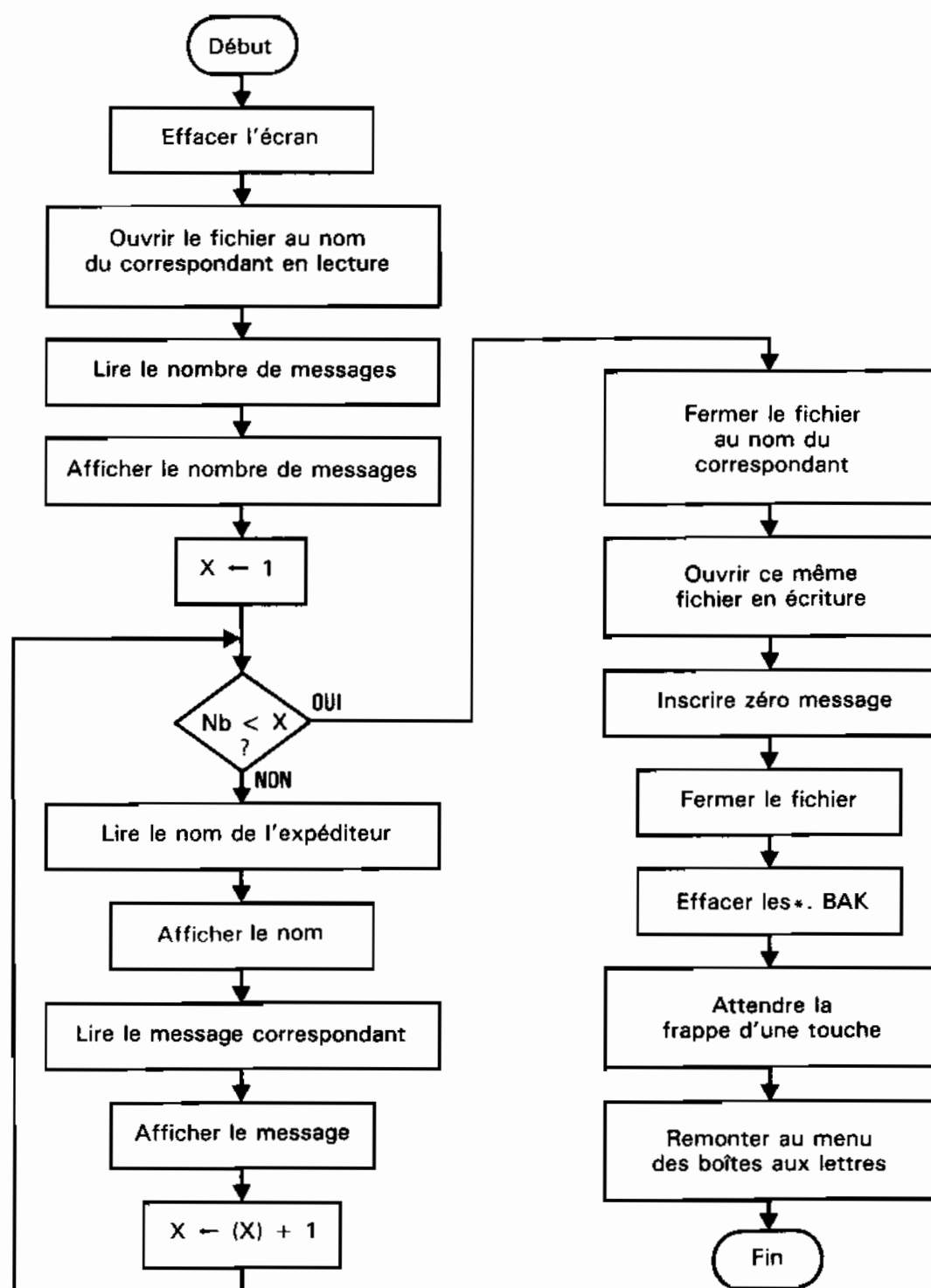
- PROCEDURE « AFFICHAGE DES MESSAGES »
 - SINON
 - Afficher l'illégalité du code
 - FINSI
 - SINON
 - Afficher la non-existence du nom dans l'annuaire
 - FINSI
 - Remonter au menu des boîtes aux lettres
- FIN



Nous trouvons dans cet algorithme le terme PROCEDURE, utilisé ici pour simplifier l'algorithme, et que l'on retrouve dans l'algorithme en tant que sous-programme.

Cette procédure est celle de l'affichage des messages que nous allons donc expliciter plus précisément dans l'algorithme et algorithme suivant :

- DEBUT
 - Efface l'écran
 - Ouvrir le fichier du nom inscrit en lecture
 - Lire le nombre de messages reçus
 - Afficher le nombre de messages
 - POUR X ALLANT DE 1 AU nombre de messages
 - Lire un nom
 - Afficher « message de » nom de l'expéditeur
 - Lire le message correspondant
 - Afficher ce message
 - FINPOUR
 - Fermer le fichier en lecture
 - Ouvrir le fichier du nom inscrit en écriture
 - Inscrire zéro messages dans le fichier
 - Fermer le fichier
 - Effacer les fichiers de copie *.BAL
 - Attendre la frappe d'une touche
 - Remonter au menu des BAL
- FIN



Ces deux parties du programme de lecture de messages vont se retrouver dans le programme suivant :

```

9210 PRINT"*****"+CHR$(24)+"LECTURE B
AL"+CHR$(24)+"*****"
9220 DATA 12,31,66,65,27,90,27,77,32,0,"
Lecture de la boîte de "
9230 RESTORE 9220
9240 GOSUB 200:READ a$:GOSUB 300
9250 a$=nom$:GOSUB 300
9260 DATA 31,72,65,27,72,0,"Patience, Ve
rification en cours ..."
9270 GOSUB 200:READ a$:GOSUB 300
9280 flag = 0
9290 code$=""
9300 OPENIN "nom.bai"
9310 WHILE NOT EOF
9320 INPUT#9,nomi$,codei$
9330 IF nom$=nomi$ THEN flag = 1:code$=c
odei$
9340 WEND
9350 CLOSEIN
9360 IF flag = 1 THEN 9440
9370 DATA 12,31,69,65,27,83,32,18,101,31
,70,65,27,86,32,18,101,31,70,68,27,77,27
,65,0
9380 RESTORE 9370
9390 GOSUB 200
9400 a$="Desole vous n'etes pas inscrit
!"
9410 GOSUB 300
9420 FOR i = 1 TO 1500:NEXT i
9430 GOTO 8000
9440 DATA 31,72,65,27,72,0,"Verification
du code d'accès ...",31,75,67,0,"Code:
",95,18,69,0,"+",27,93,0,"ENVOI"
9450 DATA 31,75,73,0
9460 RESTORE 9440
9470 FOR x = 1 TO 4
9480 GOSUB 200:READ a$:GOSUB 300
9490 NEXT x
9500 GOSUB 200
9510 x=6
9520 GOSUB 400
9530 IF code$ = texte$ THEN 9610
9540 DATA 12,31,70,65,27,85,32,18,101,31
,70,68,27,93,27,79,27,72,0
9550 RESTORE 9540
9560 GOSUB 200
9570 a$="CODE ERRONE !!!!"

```

```

9580 GOSUB 300
9590 FOR i = 1 TO 1500:NEXT i
9600 GOTO 8000
9610 DATA 13,10,10,10,0,"Ok, je depouill
e le courrier..."
9620 RESTORE 9610
9630 GOSUB 200:READ a$:GOSUB 300
9640 OPENIN nom$
9650 INPUT#9,nb
9660 DATA 12,27,90,27,69,32,0,"Vous avez
:"
9670 RESTORE 9660
9680 GOSUB 200:READ a$:GOSUB 300
9690 a$=STR$(nb)+" message(s) "
9700 GOSUB 300
9710 FOR x = 1 TO nb
9720 DATA 13,10,14,27,90,27,67,70,18,103
,15,27,66,0,"Message de: "
9730 RESTORE 9720
9740 GOSUB 200:READ a$:GOSUB 300
9750 INPUT#9,a$
9760 GOSUB 300
9770 DATA 13,10,27,70,0
9780 GOSUB 200
9790 INPUT#9,a$
9800 GOSUB 300
9810 NEXT x
9820 CLOSEIN
9830 OPENOUT nom$
9840 WRITE #9,0
9850 CLOSEOUT
9860 OPEN "*.bak"
9870 DATA 31,64,65,27,72,0,"appuyez sur
une touche"
9880 RESTORE 9870
9890 GOSUB 200:READ a$:GOSUB 300
9900 CALL &A000
9910 DATA 31,64,65,32,18,101,0
9920 GOSUB 200
9930 GOTO 8000

```

- Lignes 9210 à 9240 : présentation du choix de lecture.
- Ligne 9250 : on rappelle au miniteliste le nom (normalement le sien) de la boîte aux lettres qui va être lue.
- Lignes 9260 à 9350 : après affichage d'un message invitant à patienter, on vérifie si ce nom est bien inscrit dans l'annuaire (fichier NOM.BAL) et l'on positionne un drapeau (flag) à 1 si on l'a trouvé. Le code de la BAL est aussi récupéré pour vérification ultérieure.
- Ligne 9360 : test du flag.

- Lignes 9370 à 9430 : affichage du message signalant la non-inscription du nom dans l'annuaire, puis, après une PAUSE, on retourne au menu des boîtes aux lettres.
- Lignes 9440 à 9520 : on demande le code d'accès de la BAL que l'on récupère dans la variable **TEXTES**.
- Ligne 9530 : test du code d'accès.
- Lignes 9540 à 9600 : affichage du message signalant une erreur sur le code d'accès, puis, après une PAUSE, on retourne au menu des boîtes aux lettres.
- Lignes 9610 à 9820 : le fichier au nom du correspondant est ouvert en lecture et les différents messages sont affichés à l'écran du minitel.
- Lignes 9830 à 9860 : le fichier au nom du minitelliste est ré-initialisé, et le fichier *.BAK est effacé afin de récupérer de la place sur la disquette.
- Lignes 9870 à 9930 : on laisse le temps au minitelliste de lire ses messages en l'invitant à frapper sur une touche pour revenir au menu des BALs.

Ce programme est à écrire à la suite du programme BAL.BAS, puis sauvegardé, temporairement sous le même nom ; on pourra ensuite ajouter le dernier programme du paragraphe suivant.

LE PROGRAMME D'ÉCRITURE EN BAL

Cette partie du programme va autoriser vos correspondants à s'échanger des messages par boîtes aux lettres interposées.

Le programme demandera le NOM du destinataire, puis après vérification de l'inscription à l'annuaire, vous pourrez lui laisser un message de 240 caractères, si sa boîte aux lettres n'est pas saturée (trois messages maximum).

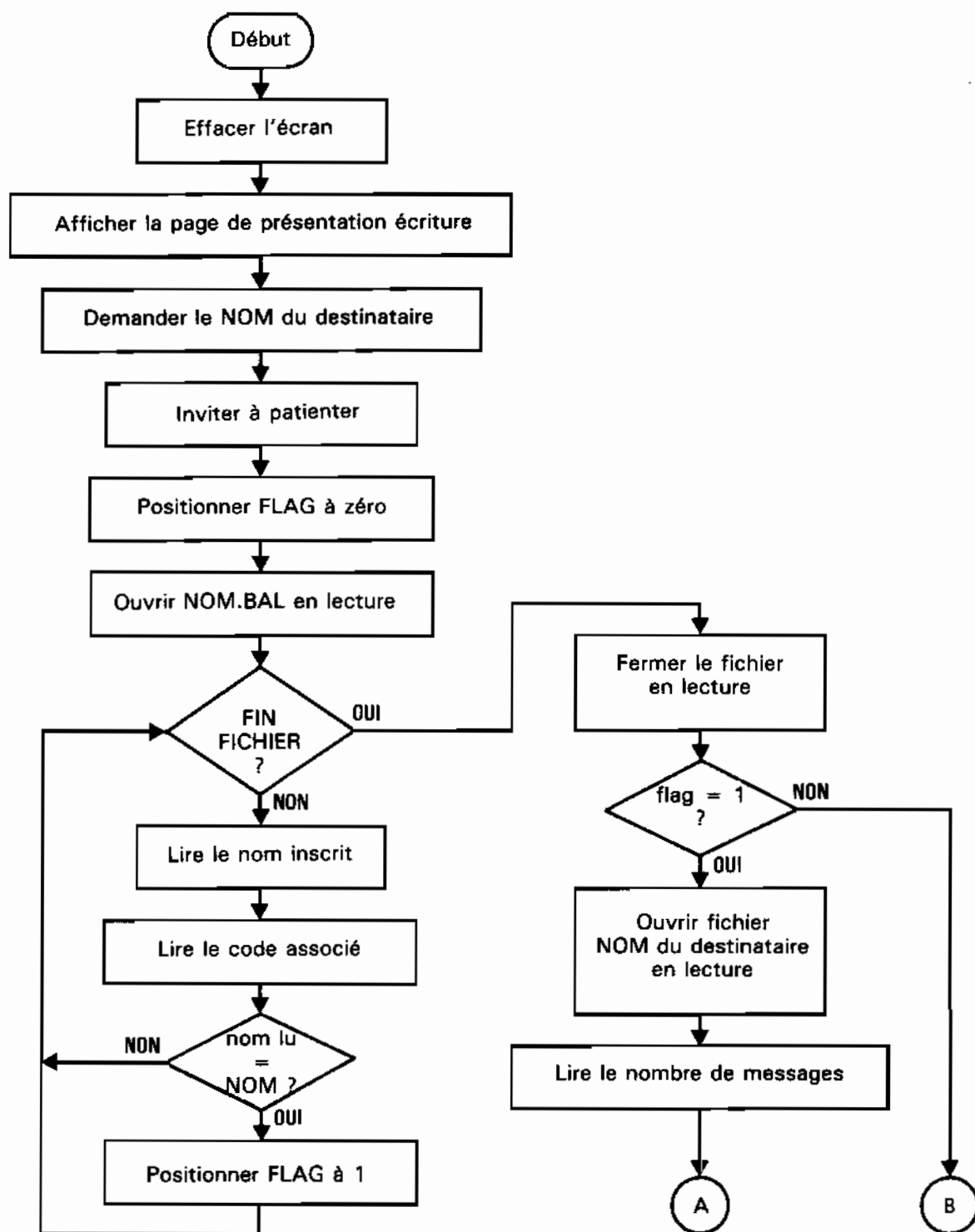
Vous trouverez l'algorithme du programme ci-dessous ainsi que son algorithme associé.

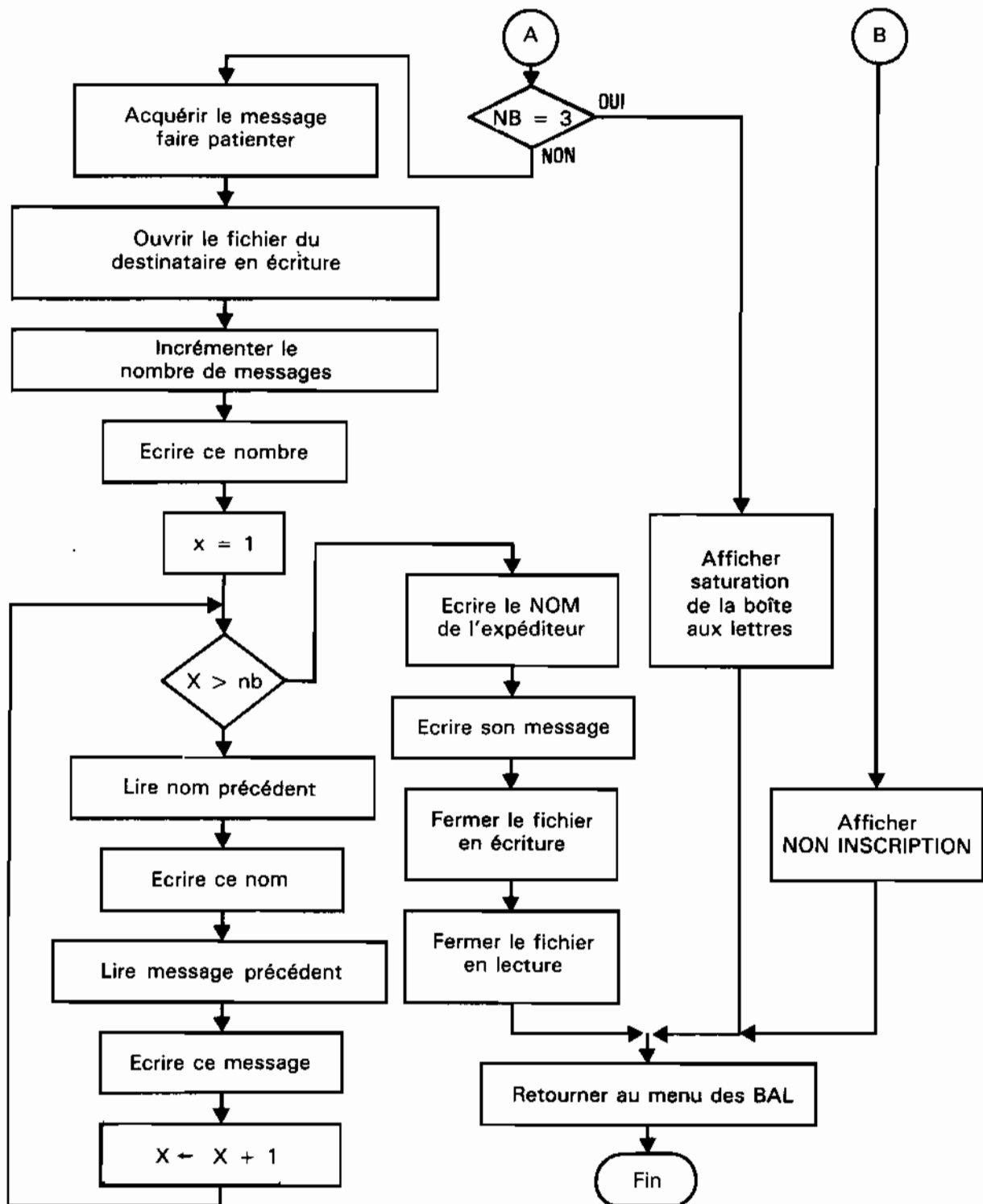
— DEBUT

- Effacer l'écran
- Afficher une page de présentation
- Demander le nom du destinataire
- Afficher un message invitant à patienter
- Positionner un « flag » à zéro
- Ouvrir le fichier des noms (NOM.BAL) en lecture
- TANTQUE la fin du fichier n'est pas atteinte
 - Lire le nom inscrit

- Lire le code associé
- SI le nom lu est identique au nom du destinataire
 - ALORS
 - Positionner le « flag » à « 1 »
- FINSI
- FINTANTQUE
- Fermer le fichier
- SI le « flag » est à « 1 »
 - ALORS
 - Ouvrir le fichier au nom du destinataire en lecture
 - Lire le nombre de messages
 - SI ce nombre est différent de 3
 - ALORS
 - Acquérir le message à poster
 - Inviter à patienter
 - Ouvrir le fichier au nom du destinataire en écriture
 - Incrémenter le nombre de messages
 - Ecrire le nouveau nombre dans le fichier
 - POUR X ALLANT DE 1 AU nombre de messages
 - Lire un nom d'expéditeur antérieur
 - Ecrire ce nom
 - Lire son message
 - Ecrire ce message
 - FINPOUR
 - Ecrire le nom de l'expéditeur actuel
 - Ecrire son message
 - Fermer le fichier en écriture
- SINON

- Afficher un message indiquant la saturation de la Boîte Aux Lettres du destinataire
 - FINSI
 - Fermer le fichier en lecture
- SINON
 - Afficher un message indiquant la non-inscription du destinataire à l'annuaire
- FINSI
- Retourner au menu des Boîtes Aux Lettres
- FIN





De cette étude algorithmique, on tire le programme suivant :

```
9940 PRINT"*****"+CHR$(24)+" ECRITURE
    EN BAL "+CHR$(24)+"*****"
9950 DATA 12,31,66,66,27,90,27,77,32,0,"
    Envoi d'un message"
9960 DATA 31,68,65,0,"A qui voulez vous
    ecrire?: "
9970 DATA 95,18,71,31,68,92,0
9980 RESTORE 9950
9990 GOSUB 200:READ a$:GOSUB 300
10000 GOSUB 200:READ a$:GOSUB 300
10010 x=8:GOSUB 400
10020 DATA 31,70,65,27,72,0,"Verificatio
    n en cours ..."
10030 RESTORE 10020
10040 GOSUB 200:READ a$:GOSUB 300
10050 flag = 0
10060 OPENIN "nom.bal"
10070 WHILE NOT EOF
10080 INPUT#9,nomi$,codei$
10090 IF texte$=nomi$ THEN flag = 1
10100 WEND
10110 CLOSEIN
10120 IF flag <> 1 THEN 10500
10130 nomi$=texte$
10140 OPENIN nomi$
10150 INPUT#9,nb
10160 IF nb = 3 THEN 10440
10170 DATA 12,31,66,65,0,"ecrit un mess
    age a "
10180 RESTORE 10170
10190 GOSUB 200
10200 a$=nom$:GOSUB 300
10210 READ a$:GOSUB 300
10220 a$=nomi$:GOSUB 300
10230 DATA 13,10,10,95,18,127,18,127,18,
    127,18,74,11,11,11,11,11,0
10240 RESTORE 10230
10250 GOSUB 200
10260 x=200:GOSUB 400
10270 DATA 13,10,10,10,10,10,0,"Veuillez
    patienter, enregistrement ..."
10280 RESTORE 10270
10290 GOSUB 200:READ a$:GOSUB 300
10300 OPENOUT nomi$
10310 nb=nb+1
10320 WRITE#9,nb
10330 FOR i = 1 TO nb-1
10340 INPUT #9,a$
```

```

10350 WRITE #9,a$
10360 INPUT #9,a$
10370 WRITE #9,a$
10380 NEXT i
10390 WRITE #9,nom$
10400 WRITE #9,texte$
10410 CLOSEOUT
10420 CLOSEIN
10430 GOTO 8000
10440 DATA 12,31,69,65,27,77,0,"Desole s
a B.A.L. est pleine"
10450 RESTORE 10440
10460 GOSUB 200:READ a$:GOSUB 300
10470 FOR i = 1 TO 1500:NEXT i
10480 CLOSEIN
10490 GOTO 8000
10500 DATA 12,31,69,65,27,77,0,"Desole c
e nom ne figure pas",13,10,10,31,27,77,0
,"a l'annuaire"
10510 RESTORE 10500
10520 GOSUB 200:READ a$:GOSUB 300
10530 GOSUB 200:READ a$:GOSUB 300
10540 FOR i = 1 TO 1500:NEXT i
10550 GOTO 8000

```

- Lignes 9940 et 9950 : présentation du choix ECRITURE.
- Lignes 9960 à 10010 : acquisition du nom du destinataire.
- Lignes 10020 à 10110 : vérification dans l'annuaire qui positionne la variable flag à 1 au cas où il y a inscription.
- Lignes 10130 à 10160 : ouverture du fichier du destinataire et lecture du nombre de messages inscrit pour éviter la saturation.
- Lignes 10170 à 10260 : acquisition du message en tenant compte du nom de l'expéditeur entré lors de l'arrivée sur le serveur.
- Lignes 10270 à 10380 : sauvegarde du nouveau nombre de messages dans le fichier du destinataire, puis de ses messages précédents.
- Lignes 10390 à 10420 : sauvegarde du nouveau message.
- Ligne 10430 : retour au menu des BAL.
- Lignes 10440 à 10490 : affichage du message de saturation de la boîte aux lettres, puis retour au menu des BAL.
- Lignes 10500 à 10550 : affichage de la non-inscription à l'annuaire, puis retour au menu des BAL.

En conclusion : après avoir écrit ce programme à la suite des autres programmes de la gestion des boîtes aux lettres, vous pouvez le sauvegarder définitivement sur disquette, et effectuer l'instruction

MERGE « BAL.BAS »

pour l'insérer au serveur.

Le programme INFORMATION

Cette rubrique INFORMATION vous permettra de laisser sur une page un certain nombre de messages que vous jugerez utiles et que vous pourrez modifier facilement sur votre AMSTRAD, grâce au logiciel de gestion que nous vous proposons ci-après.

LA CRÉATION DU FICHIER INFORMATION

Ce fichier information s'effectuera sur une page complète du minitel, c'est-à-dire 24 lignes de 40 caractères, qui correspondent à un écran CPC en mode 1, ce qui nous donne $24 \times 40 = 960$ caractères.

Nous allons donc d'abord créer le fichier par ce petit programme que vous rentrerez en mode direct :

```
10 OPENOUT "FIC.MTL"
20 FOR I = 1 TO 960
30 WRITE #9, " "
40 NEXT I
50 CLOSEOUT
60 END
```

Une fois ce programme exécuté, le fichier FIC.MTL sera créé, et rempli avec des caractères ESPACE. Il ne reste plus qu'à modifier ce fichier en y entrant le texte que vous désirez.

Pour cela vous utiliserez le programme suivant que vous pourrez sauvegarder sous le nom :

SAVE "GESFIC.BAS"

```
10 REM ***** MODIFICATION DU FICHIER
R INFORMATIONS *****
20 REM *****
*****
30 DIM FIC$(40,24)
40 MODE 1
50 REM *** lecture du fichier ***
60 PRINT "LECTURE DU FICHIER"
70 OPENIN "FIC.MTL"
80 FOR J=1 TO 24
90   FOR I= 1 TO 40
100     LOCATE 10,10
110     PRINT " ";J;SPACE$(6);I
120     INPUT #9,FIC$(I,J)
130   NEXT I
140 NEXT J
150 CLOSEIN
160 FOR i = 1 TO 960
170   PRINT ". ";
180 NEXT i
190 CLS
```

```
200 REM *** affichage du fichier sur CPC
***
210 FOR J=1 TO 24
220   FOR I=1 TO 40
230     LOCATE I,J
240     PRINT FIC$(I,J)
250   NEXT I
260 NEXT J
270 REM *** position initiale ***
280 I=1:J=1
290 LOCATE I,J
300 CURSOR 1
310 REM *** acquisition modification ***
320 WHILE A$ <> CHR$(13)
330   A$=""
340   WHILE A$ = ""
350     A$=INKEY$
360   WEND
370   IF ASC(A$) <> 127 THEN 430
380     IF I = 1 THEN GOTO 630
390     I = I-1
400     LOCATE I,J:PRINT " "
410     fic$(i,j) = " "
420     GOTO 630
430   IF ASC(A$) <> 240 THEN 460
440     IF J = 1 THEN GOTO 630
450     J = J-1:GOTO 630
460   IF ASC(A$) <> 241 THEN GOTO 490
470     IF J = 24 THEN GOTO 630
480     J = J+1:GOTO 630
490   IF ASC(A$) <> 242 THEN GOTO 520
500     IF I = 1 THEN GOTO 630
510     I = I-1:GOTO 630
520   IF ASC(A$) <> 243 THEN 550
530     IF I = 40 THEN GOTO 630
540     I = I+1:GOTO 630
550   IF ASC(A$) > 127 THEN 630
560     LOCATE I,J
570     PRINT A$;
580     fic$(i,j)=a$
590     I = I+1
600     IF I < 41 THEN 630
610     IF J = 24 THEN I = 1:J =
1:GOTO 630
620     I=1:J=J+1
630     LOCATE I,J
640 WEND
650 REM *** sauvegarde fichier ***
660 OPENOUT "fic.mtl"
670 FOR j = 1 TO 24
680   FOR i = 1 TO 40
```

```
690          IF fic$(1,j) = CHR$(34) THEN
   fic$(1,j) = " "
700          WRITE#9, fic$(i,j)
710          LOCATE 10,10
720          PRINT " ";j;SPACE$(6);i
730      NEXT i
740 NEXT j
750 CLOSE#9
760 END
```

Ce programme lit le fichier FIC.MTL, celui-ci étant relativement long, il est affiché en permanence l'état de la lecture (ligne, colonne).

Une fois le fichier affiché (vous verrez apparaître le curseur au bout de quelques dizaines de secondes), vous pourrez alors entrer votre texte.

Les touches à utiliser sont uniquement les touches de caractères affichables, ainsi que **** et les flèches de déplacement.

Vous pourrez valider le texte ainsi créé grâce à la touche **<COPY>**, ce qui provoquera la sauvegarde du texte dans le fichier FIC.MTL.

LE PROGRAMME INFORMATION

Ce sera un programme simple d'envoi du fichier FIC.MTL vers le minitel.

L'algorithme est le suivant :

- DEBUT
 - Afficher la page de présentation INFO en ligne zéro
 - Ouvrir le fichier « FIC.MTL » en lecture
 - TANTQUE la fin du fichier n'est pas atteinte
 - Lire un caractère
 - Emettre le caractère
 - FINTANTQUE
 - Afficher un message invitant à frapper une touche
 - Effacer la ligne zéro
 - Retourner au menu principal
- FIN

```
14000 PRINT"*****"+CHR$(24)+" INFORMAT
IONS "+CHR$(24)+"*****"
14010 DATA 12,31,64,65,32,18,101,31,64,7
0,27,67,27,83,27,93,27,78,0,"INFORMATION
S"
14020 RESTORE 14010
14030 GOSUB 200:READ a$:GOSUB 300
14040 DATA 31,65,65,0
14050 RESTORE 14040
14060 GOSUB 200
14070 OPENIN"fic.mtl"
14080 WHILE NOT EOF
14090 INPUT#9,a$
14100 GOSUB 300
14110 WEND
14120 CLOSEIN
14130 DATA 31,88,68,27,93,27,68,32,27,69
,32,27,65,32,27,66,32,27,67,32
14140 DATA 31,88,95,27,93,27,67,32,27,66
,32,27,65,32,27,69,32,27,68,32
14150 DATA 31,88,73,27,72,27,93,0,"APPUY
EZ SUR UNE TOUCHE"
14160 RESTORE 14130
14170 GOSUB 200:READ a$:GOSUB 300
14180 CALL &A000
14190 DATA 31,64,65,32,18,101,0
14200 RESTORE 14190
14210 GOSUB 200
14220 GOTO 13000
```

Ce programme sera sauvegardé sous le nom

SAVE "INFO.BAS"

et il suffira de l'insérer par **MERGE** dans le programme du serveur.

8/5.5

OUT et les ports de sortie

L'instruction Basic **OUT**, qui possède d'ailleurs un équivalent en Assembleur, permet d'exécuter des opérations de sortie sur ports.

Un port de sortie n'est pas forcément un connecteur accessible à l'arrière de l'ordinateur : ce peut bien sûr être, par exemple, la prise d'imprimante mais aussi l'interface purement interne entre le microprocesseur et le synthétiseur de sons ou le clavier.

En fait, un port est au départ une simple entité logique : une combinaison de niveaux logiques sur les bus du microprocesseur Z80. La façon dont ces signaux sont exploités par la circuiterie périphérique est une toute autre affaire, qui ne regarde finalement pas vraiment l'unité centrale.

L'exécution d'une instruction **OUT A,B** se traduit de la façon suivante au niveau des bus :

- la ligne $\overline{\text{IORQ}}$ du bus de commande est mise à zéro par le microprocesseur, exactement comme $\overline{\text{MREQ}}$ le serait en cas d'écriture en mémoire ;
- simultanément apparaît sur le bus d'adresse un ensemble de 16 niveaux logiques traduisant, en binaire, la valeur de **A** ;
- toujours en même temps apparaît sur le bus de données un ensemble de 8 niveaux logiques traduisant, en binaire, la valeur de **B**.

En pratique, ces états ne persistent que quelques centaines de nanosecondes (soit moins d'un millième de seconde) : c'est aux circuits périphériques du microprocesseur qu'il appartient de les intercepter au vol et, si nécessaire, de les mémoriser avant de les faire agir sur des dispositifs externes à temps de réponse plus élevé.

L'information véhiculée par le bus d'adresses quand $\overline{\text{IORQ}}$ est à zéro identifie tout simplement le périphérique destinataire de l'opération de sortie, tandis que celle présente sur le bus de données n'est autre que la représentation codée des états devant être pris par le périphérique en question.

Le tableau 1 explique comment faire correspondre l'argument **A** de **OUT A,B** aux états des lignes d'adresse : pendant l'exécution d'un **OUT 62719,B**, par exemple, toutes les lignes du bus d'adresse resteront à 1, sauf A8, A9 et A11 qui viendront à 0. Dans le cas d'un **OUT 61439,B** par contre, seule A12 viendra à 0.

Tableau 1

A ₀	A ₁	A ₂	A ₃	A ₄	A ₅	A ₆	A ₇	A ₈	A ₉	A ₁₀	A ₁₁	A ₁₂	A ₁₃	A ₁₄	A ₁₅	A de OUT A,B
1	2	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384	32768	
1	1	1	1	1	1	1	1	0	0	1	0	1	1	1	1	62719
1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	1	62975
1	1	1	1	1	1	1	1	0	1	1	0	1	1	1	1	63231
1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	63487
1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	61439

Le cas du bus de données est pour sa part traité au tableau 2 : compte tenu du poids de chaque ligne, **OUT A,32** déclenchera par exemple le passage à 0 de toutes les lignes de données à l'exception de D5 qui passera à 1.

Inversement, **OUT A,243** mettra D2 et D3 à 0, les autres lignes de données restant à 1.

Tableau 2

D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇	ligne de données
1	2	4	8	16	32	64	128	poids décimal

Physiquement, un port de sortie sera le plus souvent matérialisé par un groupe de huit bascules chargées de mémoriser les états du bus de données au moment où ceux-ci sont désignés comme significatifs par les bus d'adresse et de commande.

Pour faciliter ce travail, deux signaux internes au CPC sont élaborés à partir de IORQ, WR et RD en provenance du Z80 : la figure 1 permet de constater que IOWR à zéro indique une opération de sortie sur port, tandis que IORD passe à zéro en cas d'entrée sur port (par INP), ce qui ne nous concerne pas directement ici.

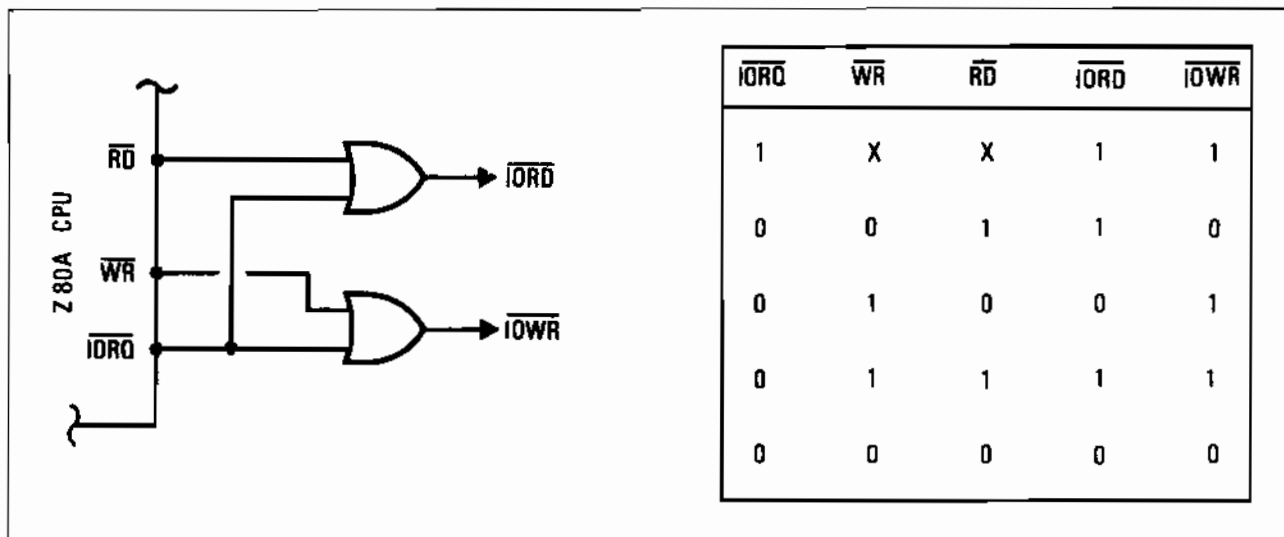


Fig. 1

Plusieurs ports remplissant des fonctions importantes sont matériellement regroupés dans le circuit intégré PIO (*Parallel Input Output*) 8255: on y accède par les adresses 62719, 62975, 63231 et 63487, grâce aux branchements décrits à la figure 2. Compte tenu de la relative complexité des modalités d'accès à ces ports, un chapitre entier est consacré à ce sujet (voir Partie 2, chapitre 3.4.1).

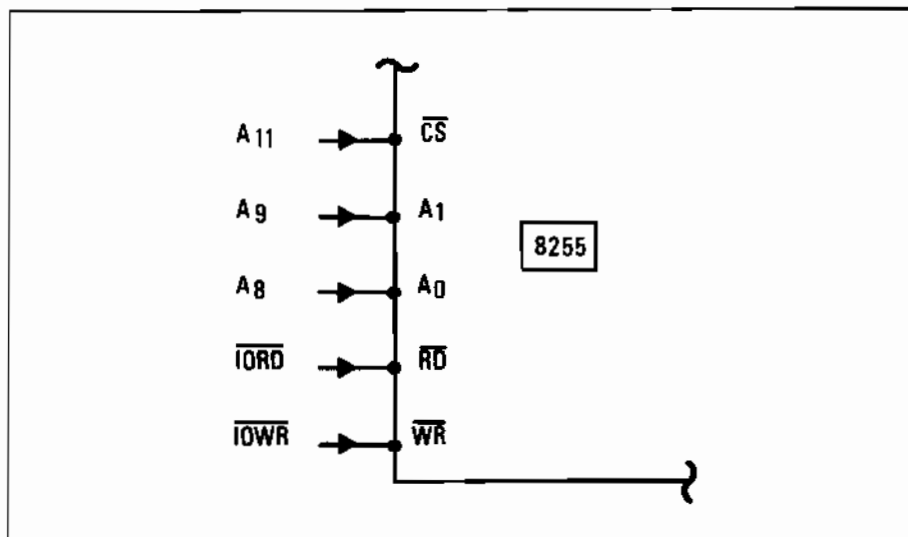


Fig. 2

Quelques ports sont réservés aux communications avec des périphériques spécialisés (*gate arrays* ou contrôleur d'écran) et nécessitent, pour être manipulés, une profonde connaissance de leur fonctionnement interne.

Le port d'imprimante parallèle, par contre, se prête bien à toutes sortes d'expérimentations.

Le port d'imprimante est matérialisé par un circuit intégré 74LS273 ou 74HC273, regroupant huit bascules D interposées chacune entre l'une des lignes du bus de données et l'une des sorties de la prise d'imprimante. Seule la ligne D7, véhiculant le STROBE, possède en plus un inverseur (74LS132).

La figure 3 montre que l'entrée d'horloge commune aux huit bascules est commandée par le résultat d'un OU logique entre les signaux $\overline{IOWR}$ et A12: les informations circulant sur le bus de données seront donc gelées sur les sorties des bascules au moment précis où une opération de sortie sera exécutée vers un port dont l'adresse est telle que la ligne A12 est à zéro (voir table de vérité ci-dessous).

A	B	A ou B
0	0	0
0	1	1
1	0	1
1	1	1

Afin d'éviter tout conflit avec d'autres ports, il est souhaitable que toutes les autres lignes d'adresse soient à 1 à ce moment-là, et donc d'utiliser l'adresse 61439 même si beaucoup d'autres peuvent sembler convenir.

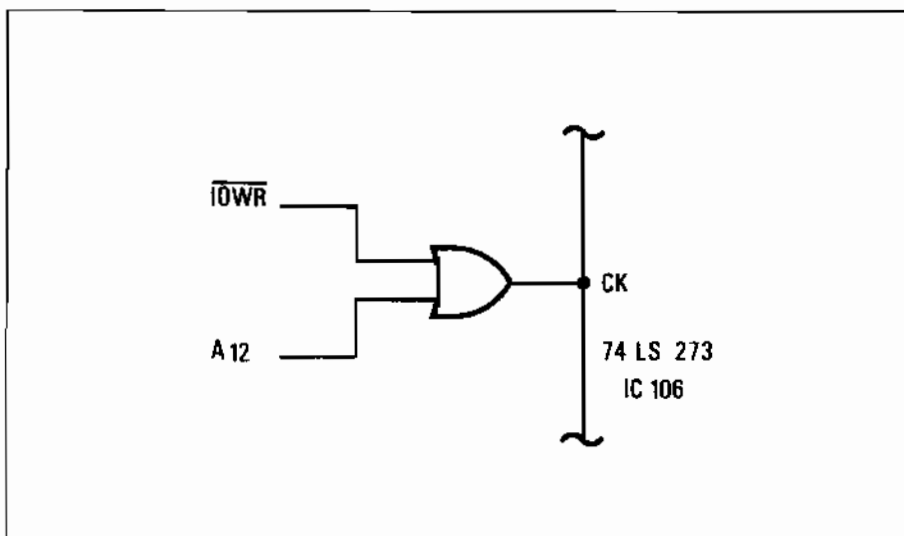


Fig. 3

Cela étant posé, le tableau 3 donne quelques exemples d'états pris par les broches de la prise d'imprimante en fonction de ceux du bus de données, eux-mêmes fonction de l'argument de l'instruction **OUT 61439**.

On notera que, conformément à la conception matérielle de ce port, le niveau présent sur la broche 1 du connecteur (STROBE) est inverse de celui de la ligne D7, mémorisé par la bascule correspondante.

Tableau 3

OUT 61439, ...	D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇		Lignes de données
	2	3	4	5	6	7	8	X	1	Broches prise imprimante
128	0	0	0	0	0	0	0	1	0	
0	0	0	0	0	0	0	0	0	1	
32	0	0	0	0	0	1	0	0	1	
160	0	0	0	0	0	1	0	1	0	
255	1	1	1	1	1	1	1	1	0	
254	0	1	1	1	1	1	1	1	0	
170	0	1	0	1	0	1	0	1	0	
4	0	0	1	0	0	0	0	0	1	

La figure 4 fournit un schéma très simple permettant la commande d'un petit relais « REED » 500 Ω 5V à partir de la ligne de STROBE, en l'absence d'imprimante bien sûr : pour le manœuvrer, il faudra faire alterner des OUT 61439,128 et des OUT 61439,0.

Signalons d'ailleurs que plusieurs applications pratiques décrites dans la Partie 10 exploitent cette facilité...

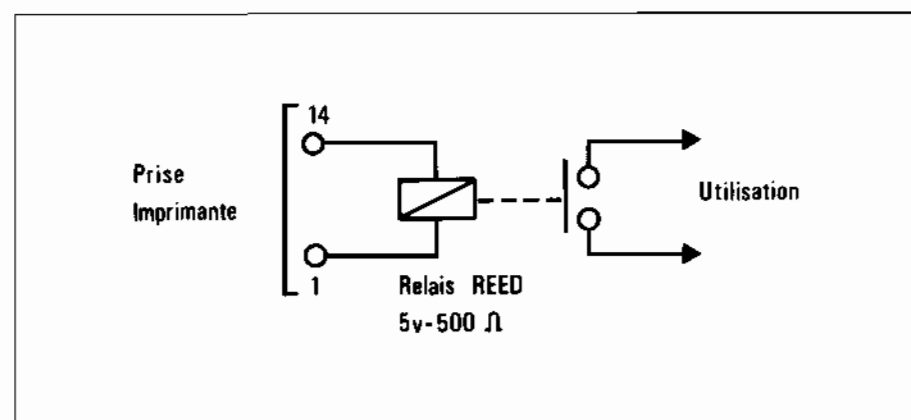


Fig. 4

8/6

Les mémoires de masse

8/6.1

Le lecteur de disquettes 5'1/4 Vortex en lecteur additionnel

Qui n'a pas un jour, suite à une sauvegarde, été confronté à l'affreux message d'erreur « Disc Full » ?

Qui ne désire pas payer moins cher ses disquettes, et en consommer environ quatre fois moins ?

La solution est très certainement l'achat d'un lecteur additionnel. Un lecteur 5'1/4 vous permettra de limiter les dépenses en disquettes, puisque la capacité de stockage est supérieure.

Le lecteur de disquettes 5'1/4 Vortex allie l'économie au confort de sauvegarde ainsi qu'à une grande souplesse d'utilisation logicielle : en effet il est livré avec un ROM supplémentaire remplaçant la ROM-AMSDOS et possédant de nombreuses instructions nouvelles (possibilité de choisir

son lecteur principal, possibilité de gérer simultanément jusqu'à 16 fichiers en accès direct, protection par mots de passe, protection lecture-écriture, possibilité de lancer CP/M de n'importe quel lecteur, utilisation de CP/M + en 708 k-octets, possibilité de formater directement sous basic...).

I. Présentation

Le lecteur de disquettes Vortex se présente sous la forme d'un boîtier d'environ 30 cm sur 15 cm pour une hauteur de 4 cm, d'une conception robuste. Un câble plat avec un connecteur 2 x 17 broches vous permettra de l'installer sur votre CPC 664 ou 6128. Pour le CPC 464, une petite manipulation sera nécessaire pour le connecter au contrôleur de disquette DDI-1.

Une extension ROM accompagne le lecteur (3 versions selon le type de CPC que vous possédez) ; celle-ci se connectant sur le bus d'extension 2 x 25 broches du micro-ordinateur.

Il est, de plus, livré avec une disquette contenant des utilitaires (formatage, copie, augmentation de la rapidité d'affichage...) fonctionnant sous CP/M, mais ne possédant pas les secteurs de « BOOT » (secteurs permettant le démarrage de CP/M).

Plusieurs versions du lecteur existent :

- Le F1-X : lecteur 5'1/4
- le M1-X : lecteur 3'1/2
- le F1-XRS : lecteur 5'1/4 avec en plus une interface RS-232 incorporée
- Le M1-XRS : identique au précédent mais en version 3'1/2.

Nous nous intéresserons plus particulièrement au lecteur F1-X, la version M1-X ne présentant pas de différence excepté le format des disquettes. La version RS-232 nous ayant posé quelques problèmes de compatibilité, notamment avec CP/M, nous ne l'aborderons pas.

II. Installation

F1-X VERSION CPC-464

Munissez-vous d'un tournevis cruciforme et armez-vous d'un peu de patience et de beaucoup de soin.

Tout d'abord, démontez le boîtier contenant le module V-DOS par ses quatre vis de maintien.

Ensuite déconnectez votre contrôleur de disquette DDI-A du CPC-464 et enfichez le connecteur femelle 2 x 25 contacts dans la partie mâle réservée à cet effet dans le module V-DOS (petite plaque de circuit imprimé libre de composants).

Vous pouvez remonter le boîtier du module V-DOS en veillant à ne pas coincer les câbles plats (un passage est réservé).

Vous pouvez maintenant connecter le module V-DOS sur le bus d'extension du CPC-464 (le dessus est repéré, sinon prenez le fil rouge du connecteur plat pour référence de la broche 1).

Attention : toute manipulation devra être effectuée hors tension !

Il vous incombe maintenant de connecter les deux lecteurs de disquettes sur le module compact que vous avez maintenant : placer le connecteur du DDI-1 sur le premier connecteur et le connecteur du lecteur F1-X sur le deuxième.

Voilà, tout est prêt pour fonctionner. L'ordre de mise sous tension sera le suivant :

- Lecteur DDI-1
- Lecteur F1-X
- CPC-464

VERSION CPC-664 ET CPC-6128

Le montage pour ces deux types de CPC est extrêmement simple.

Il suffit de connecter le câble plat 2 x 17 broches du lecteur de disquettes 5 1/4 sur le connecteur arrière de votre ordinateur dénommé « DISC DRIVE 2 ». On tâchera de bien repérer le sens de connection (repère pour le haut ou fil rouge pour la broche 1).

Le module V-DOS se connectera par l'intermédiaire du connecteur 2 x 25 broches sur le connecteur d'extension de votre Amstrad.

L'ordre de mise sous tension sera le suivant :

- Lecteur Vortex
- CPC-664 ou 6128

III. Mise en place

Etudiez tout particulièrement ce paragraphe pour installer la disquette livrée avec votre lecteur de disquettes, car celle-ci n'est pas prête à l'emploi sous CP/M ; et vous n'en avez qu'une !

CP/M n'a pas été installé sur cette disquette car c'est un système d'exploitation créé par Digital Research, sur lequel il faut payer des droits. La disquette est donc formatée sous un format spécial qui réserve les deux premières pistes pour l'installation de CP/M (c'est l'équivalent du formatage de type Vendor proposé par les utilitaires de la disquette CP/M).

Procédez donc à l'installation dans l'ordre suivant, même si vous possédez un CPC-6128, il vous faudra d'abord créer une disquette CP/M 2.2 pour le lecteur Vortex.

INSTALLATION DE CP/M 2.2

Préparez d'abord une copie de votre disquette CP/M 2.2 sur le lecteur 3', à l'aide de DISCKIT2 par exemple.

Toute l'installation suivante s'effectue avec le module V-DOS et le lecteur Vortex installés.

Après avoir placé la copie dans le lecteur et effectué un RESET, frappez l'ordre ICPM.

Il faut maintenant préparer votre copie à fonctionner sous les capacités nouvellement disponibles par le lecteur Vortex.

CP/M doit donc être déplacé afin de pouvoir traiter les 128 enregistrements possibles dans le catalogue.

Ce déplacement s'effectue par la commande :

A>MOVCPM 178 * <RETURN>

CP/M est abaissé de 256 octets et le message suivant apparaît :

CONSTRUCTING 44K CP/M vers 2.2

READY FOR « SYSGEN » OR

« SAVE 34 CPM44.COM »

puis le prompt réapparaît (le prompt est le nom souvent utilisé pour nommer le symbole « A> » indiquant que CP/M vous rend la main).

A partir de ce moment, le nouveau CP/M est en RAM, il faut donc le sauvegarder sur disquette et de plus l'installer. On frappera donc :

A>SYSGEN *

s'affichera ensuite :

SYSGEN V2.0

Please insert DESTINATION disc into drive A then press any key : _

Vous frapperez alors <RETURN> puis répondrez « N » à la question :

Do you wish to reconfigure another disc (Y/N) ? : _

SYSGEN V2.0 finished

Le nouveau CP/M 2.2 est maintenant installé sur votre disquette 3'.

Il reste à l'installer sur la disquette fournie avec le lecteur Vortex.

Effectuez une réinitialisation de votre lecteur de disquettes puis lancez maintenant la nouvelle version de CPM par la commande :

ICPM,1 <RETURN>

Vous constaterez la modification grâce au message de présentation suivant :

CP/M 2.2 - Amstrad Consumer Electronics plc

44 K CP/M vers. 2.2-03/86 vortex GmbH

A>

Vortex est venu ajouter son « plus » et vous disposez maintenant d'une capacité de 708 k-octets sur le lecteur 5'1/4.

Placez la disquette livrée dans le lecteur 5'1/4.

Frappez maintenant :

A>B : <RETURN>

B>SYSGEN <RETURN>

Les réponses à apporter aux différents messages sont indiquées en souligné dans les opérations décrites ci-dessous :

Disquette-source dans le lecteur A ou B ? A

Veillez insérer la disquette-source dans le lecteur A puis appuyez sur ENTER <RETURN>

Disquette-destination dans le lecteur A ou B ? B

Veillez insérer la disquette-destination dans le lecteur B puis appuyer sur ENTER <RETURN>

Une autre disquette O/N ? N

Insérez une disquette système dans le lecteur A puis appuyez sur ENTER RETURN

B>

Voilà votre disquette CP/M 5'1/4 version Vortex prête à être utilisée. Il ne vous reste plus qu'à copier les fichiers utilitaires de CP/M 2.2 que vous désirez avoir sur cette disquette à l'aide de la commande PIP (se trouvant sur le lecteur A actuellement) :

B>A:PIP B:=A:XXXXXXXX.YYY

XXXXXXXX.YYY représente le nom du fichier que vous désirez copier sur la disquette Vortex.

Il est inutile de copier les fichiers BOOTGEN.COM ; DISCKIT2.COM ; FORMAT.COM ; SETUP.COM ; SYSGEN.COM et tous les fichiers LOGO, ceux-ci ne fonctionnant pas depuis le lecteur Vortex.

N'ayez pas de regret pour DISCKIT2.COM et FILECOPY.COM, car Vortex a placé sur la disquette deux utilitaires dénommés DISCTOOL.COM et FILECOPY.COM très performants.

Vous pouvez de suite essayer de lancer CP/M 2.2 à partir du lecteur Vortex. Pour cela, effectuez une ré-initialisation de votre ordinateur puis frappez **ICPM,2 <RETURN>**.

CP/M 2.2 s'initialise depuis votre deuxième lecteur qui devient maintenant le lecteur principal ou lecteur A — à ne pas oublier dès ce moment, surtout lors de copies de disquettes ou de formatage !! — le lecteur B étant alors le lecteur 3'.

INSTALLATION DE CP/M 3.0 OU CP/M+

Si vous possédez au moins 128 k-octets de mémoire vive, vous pouvez installer la version 3.0 de CP/M qui vous ouvre : la bibliothèque d'utilitaires professionnels tels que DBaseII ou Multiplan.

Ré-initialisez votre ordinateur, placez une disquette vierge 5'1/4 dans le lecteur Vortex et formatez-la par la commande Basic V-DOS : **IFORMAT,2**(voir page 10)

Insérez les deux disquettes précédentes dans les lecteurs (les versions CP/M 2.2 configurées plus haut) et lancez votre nouvelle version de CP/M par la commande **ICPM,1**

Après le prompt, frappez la commande

A>PIP A: = B:XPATCH30.COM

Remplacez la disquette 5'1/4 par la disquette vierge puis lancer la commande :

A>XPATCH30

Suivez les instructions telles que décrites ci-dessous : vous répondrez **NON** à la question Sauvegarde sur 3'' (n souligné)

Vous voici maintenant en possession d'une disquette 5'1/4 sur laquelle vous pouvez effectuer les copies de vos logiciels fonctionnant sous CP/M+

Pensez tout de même à en effectuer des copies de sauvegarde auparavant.

xpatch30

```
*****
*
*      CP/M 3.0 (V1.1F-X) PROGRAMME-MASTER      *
*      (C)1986  vortex GmbH                      *
*
*****
```

(XPATCH30 modifie le fichier CP/M+ '.EMS' pour le rendre compatible avec votre lecteur-X)

Inserez une copie de la disquette 3" CP/M + dans le lecteur 3" puis appuyez sur ENTER

Chargement du fichier C10CPM3.EMS...
Sauvegarde sur 3" (O/N) ? n
Ecriture de CPM3.EMS...

Ecriture du secteur-boot...

5.25"/3.5" disquette-master CP/M Plus-X fut creee

Vous avez actuellement 3 possibilites de lancer CP/M plus:

- (1) Disquette CP/M plus master dans le lecteur 3":
Lancer CP/M a partir du lecteur 3" par @CPM<ENTER>.
Lecteur B sera considere comme un simple face 40 pistes avec 64 enregistrements dans le directory.
Attention: Ne jamais utiliser une disquette ecrite sous ----- VDOS ou un CP/M pouvant etre appele par CPM,1 ou CPM,2.
- (2) CP/M plus modifie dans le lecteur 3":
@CPM<ENTER> ou @CPM,1<ENTER> lance le CP/M a partir du lecteur 3".
Configuration: lecteur A=3", lecteur B=5.25" ou 3.5" avec 708k0 et 128 enregistrements dans le directory.
- (3) @CPM,2<ENTER> lance le CP/M plus a partir du lecteur 5.25" ou 3.5".
Configuration: lecteur A=5.25" ou 3.5" avec 708k0 et 128 enregistrements dans le directory; lecteur B=3".

Inserez la disquette CP/M 2.2 avec XPATCH30 dans le lecteur 3" puis appuyez sur ENTER.

IV. Utilitaires Vortex sous CP/M

DISCTOOL.COM

Cet utilitaire ne fonctionnant que sous CP/M 2.2 vous permettra de formater, d'effectuer des copies de disquettes, ou créer des disquettes systèmes. Il allie même la souplesse d'utilisation en proposant de se servir du lecteur 3' en tant qu'unité temporaire pour éviter les multiples manipulations de disquettes.

FILECOPY.COM

Cet utilitaire permet d'effectuer des copies de fichiers en les sélectionnant directement dans le catalogue. Ces copies peuvent être effectuées d'un lecteur vers l'autre, ou sur un même lecteur sans provoquer d'erreurs, même entre zones USERS. Cet utilitaire ne fonctionne aussi que sous CP/M 2.2.

Les présentations de ces deux utilitaires permettent une manipulation aisée et sont de plus très conviviaux, ce qui permet d'éviter les erreurs.

FAST.COM

Cet utilitaire n'est utilisable lui aussi que sous CP/M 2.2 (et c'est bien dommage !). Il permet de pratiquement doubler la vitesse d'affichage à l'écran et est compatible avec le fenêtrage. Mais gare aux graphismes, vous pourrez avoir quelques surprises d'affichage.

INSTALL.COM

Cet utilitaire vous permettra d'ajuster tous les paramètres de CP/M lors du « BOOT ».

Il est documenté en menus pour en faciliter l'utilisation, mais une sérieuse connaissance de CP/M vous sera nécessaire pour en connaître toutes les applications.

CASCOPY.COM

Vous pourrez effectuer très facilement vos copies de cassettes à disquettes et inversement grâce à cet utilitaire.

XMON.COM

Amateurs de langage machine, voici un utilitaire qui vous conviendra certainement si vous êtes passionnés par la programmation au plus près du cœur de votre ordinateur.

P22.COM et P30.COM

Ces deux logiciels permettront de modifier les paramètres de vitesses d'accès des lecteurs de disquettes pour les adapter respectivement sous CP/M 2.2 et CP/M+. Nous n'en avons pas encore eu l'utilité, nos logiciels fonctionnant pour l'instant sans problème avec le lecteur Vortex.

LA DOCUMENTATION

Le lecteur est livré avec une documentation très fournie, notamment sur les particularités de CP/M. Un seul regret : la traduction est parfois un peu difficile à comprendre, mais les initiés y trouveront tout de même de quoi passer d'agréables soirées...

V. Le Basic V-DOS

Le lecteur Vortex est livré avec un boîtier d'extension permettant de le « piloter », assurant notamment la gestion des 80 pistes de la disquette 5 1/4 (à cet usage, nous vous recommandons d'utiliser des disquettes double densité 96 TPI).

Mais ce petit boîtier possède une trentaine d'instructions de type RSX utilisables sous Basic.

La ROM contenue dans ce boîtier est une ROM venant remplacer la ROM de l'AMSDOS. On y retrouve des instructions AMSDOS telles quelles, certaines ayant la même fonction mais dont le nom a changé, et une vingtaine d'instructions supplémentaires.

Remarque : Dès lors que vous ne connectez pas l'extension V-DOS le lecteur Vortex se comporte comme un simple lecteur 5 1/4 de capacité maximale 178 k-octets. Les fichiers ne seront par contre concentrés que sur la moitié de la disquette, ce qui implique l'utilisation de disquettes double densité tout de même pour une sauvegarde fiable.

LES COMMANDES AMSDOS

Voici la liste des commandes dont la syntaxe n'a pas été modifiée par la ROM V-DOS et dont la fonction reste la même :

ICPM	Accès à CPM sur lecteur 3'
IDISC	Affectation par défaut du lecteur de disquettes en lecture/écriture
IDISC.IN	Affectation du lecteur de disquettes en lecture
IDISC.OUT	Affectation du lecteur de disquettes en écriture
IA	Le lecteur A est le lecteur par défaut
IB	Le lecteur B est le lecteur par défaut
IERA	Effacement d'un fichier sur la disquette
IREN	Modification du nom d'un fichier
IDIR	Catalogue d'une disquette ou cassette

LES COMMANDES RENOMMÉES

Certaines commandes de l'AMSDOS ont été renommées :

ICAS	Remplace la commande ITAPE qui définit le lecteur de cassette comme étant l'unité par défaut
ICAS.IN	Remplace ITAPE.IN qui affecte le lecteur de cassette en lecture
ICAS.OUT	Remplace ITAPE.OUT qui affecte le lecteur de cassette en écriture
ISELECT	Remplace l'instruction IUSER qui définit le numéro de la zone utilisée. Cette fois-ci, il y a une légère modification supplémentaire : on écrira ISELECT, « 1 » pour sélectionner la zone utilisateur N° 1 (au lieu de IUSER, 1).

Le nombre de zones est toujours fixé à 16 (de 0 à 15). Il est possible d'utiliser **ISELECT**, « B5 », pour sélectionner la zone 5 sur le disc B. Si vous utilisez cette commande à l'aide d'une variable, vous devrez affecter le caractère @ (a commercial) à la variable : **ISELECT**, @a\$ avec a\$ = « A3 » par exemple.

ICPM

La fonction **ICPM** a été légèrement modifiée pour permettre le passage à CP/M sur n'importe quel lecteur et en version 708 k-octets (voir paragraphe sur les utilitaires). **ICPM,1** appelle CP/M se trouvant sur le lecteur 3', celui-ci étant le lecteur A ; et **ICPM,2** charge CP/M à partir du lecteur Vortex qui devient le lecteur A par défaut.

LES NOUVELLES COMMANDES**IX**

Cette commande vous permet d'inverser à volonté les lecteurs A et B par défaut. Si lecteur A est le lecteur 3' ; après la commande IX, il deviendra le lecteur B tandis que le lecteur Vortex deviendra le lecteur A. IX une nouvelle fois rétablit l'ordre à l'initialisation.

IFORMAT

Cette commande vous permet de formater directement sous Basic une disquette.

IFORMAT,1 vous formatera une disquette du lecteur A

IFORMAT,2 vous formatera une disquette sur le lecteur B.

Lorsque vous formateriez sur le lecteur 3', il vous sera demandé le type de formatage désiré (Système ou Data), et la disquette sera formatée en 40 pistes 9 secteurs.

Lorsque vous formateriez sur le lecteur Vortex, la disquette 5 1/4 sera formatée en 80 pistes 9 secteurs et double face.

Dans tous les cas, assurez-vous que vous êtes sur le bon lecteur suite à la commande IX.

IATTRIBUT

Cette commande est similaire à la commande STAT de CP/M en attribuant aux fichiers les attributs suivants :

IATTRIBUT, « FICHIER.EXT », « S » le fichier est considéré comme fichier Système et ne sera pas visible au catalogue.

IATTRIBUT, « FICHIER.EXT », « D » annule la commande précédente, le fichier est de nouveau visible au catalogue.

IATTRIBUT, « FICHIER.EXT », « W » le fichier est affecté de l'attribut R/W (READ/WRITE), c'est-à-dire qu'il peut être lu et effacé.

IATTRIBUT, « FICHIER.EXT », « R » Le fichier est affecté de l'attribut R/O (READ ONLY), il est protégé en écriture donc ineffaçable ou modifiable.

CODE

Cette commande vous permet d'installer une clé d'accès à 65536 combinaisons sur un programme que vous désirez protéger. Ce code est un nombre de 5 chiffres signé ou non compris entre - 32768 et + 32767.

Le format de la commande est le suivant :

ICODE, 12345

save « PROGRAM.BAS »,P

Votre programme est ainsi protégé avec le code 12345. Après une ré-initialisation, il vous sera impossible de l'utiliser si vous ne reappelez pas la commande code suivie de votre nombre.

IRESET

Cette commande vous permet de changer la disquette de votre lecteur au cas où vous auriez ouvert des fichiers sur celle-ci. Elle ferme tous les fichiers et vous évite ainsi de perdre des données sur des fichiers ouverts en écriture. Nous ne saurions trop vous conseiller de prendre l'habitude de l'utiliser dans les programmes demandant un changement de disquette.

IDERROR

Cette commande permet d'éviter un arrêt du programme Basic dans le cas où une erreur surviendrait. Trois syntaxes existent :

IDERROR,0 active la commande qui n'affiche ainsi plus les messages d'erreurs et continue le traitement du programme Basic en cours.

IDERROR,1 désactive la commande, le programme s'arrêtera en cas d'erreur et celle-ci sera affichée.

IDERROR sans paramètre, crée une variable DERR qui contiendra le code d'erreur qui s'est produit pour un traitement ultérieur.

Les principaux codes d'erreurs sont les suivants :

10 : disquette changée sans fermeture du fichier en cours

14 : fichier non ouvert

15 : fin du fichier atteint

17 : imprimante non connectée

21 : disquette pleine

22 : directory plein

23 : fichier non présent sur la disquette

24 : fichier impossible à renommer, déjà existant

25 : fichier uniquement en lecture

33 : erreur de syntaxe

39 : champ trop long

40 : champ non défini

41 : champ non divisé

- 42 : canal déjà ouvert
- 43 : canal non ouvert
- 44 : numéro de canal erroné
- 45 : numéro d'enregistrement trop grand
- 53 : canal non défini
- 55 : enregistrement vide

IROMS

Cette commande vous donne la liste des ROMs connectées à votre CPC, ainsi que leur numéro et leur nom.

IROMOFF

Vous pouvez, si vous le désirez, déconnecter tout ou partie de vos ROMs à l'aide de cette commande.

IROMOFF,1 déconnecte la ROM N° 1

IROMOFF seul déconnecte toutes les ROMs

IAMS DOS

Cette commande vous permet de retourner au système d'exploitation AMSDOS. Le lecteur X redevient un lecteur simple 40 pistes.

Attention : les commandes **IROMOFF** et **IAMS DOS** vous font perdre toutes les données en mémoire centrale !

LES FICHIERS À ACCÈS DIRECT

Rappelons ce qu'est un fichier séquentiel.

Lorsque vous utilisez le Basic livré avec votre Amstrad vous pouvez gérer des fichiers grâce aux instructions OPENOUT, OPENIN, INPUT #9, PRINT #9, LINE INPUT #9, WRITE #9, PRINT #9, CLOSEIN ET CLOSEOUT. Ces fichiers sont appelés fichiers séquentiels car les données y sont rangées dans l'ordre et la lecture d'une donnée placée en n-ième position ne peut se faire qu'en ayant lu les n-1 données placées avant. De même pour modifier le fichier, vous devez lire tout le fichier, modifier la donnée et réenregistrer le fichier.

Sur des fichiers de petite taille, pas de problème, mais si vous vous lancez dans la gestion d'un fichier de taille très importante, alors vous risquez de perdre patience, et surtout de la mémoire.

Les fichiers à accès direct, (appelés aussi relatifs ou aléatoires), vous permettront de surmonter ce problème.

Vortex a même prévu la possibilité d'ouvrir en même temps 16 fichiers à accès direct, aussi bien en écriture qu'en lecture.

Une restriction est à appliquer à ce type de fichiers, vous ne pouvez enregistrer que des chaînes de caractères, mais les instructions STR\$ et VAL vous permettront un passage aisé aux valeurs numériques écrites ou lues.

Voyons maintenant comment organiser un fichier à accès direct, que nous appliquerons à une mini-gestion de stocks.

Vous devez d'abord définir la structure de vos enregistrements, c'est-à-dire le nom des variables qui seront stockées ainsi que leur contenu maximal en caractères.

Par exemple :

- * Article : 20 caractères → ART\$
- * Référence : 7 caractères → REF\$
- * Quantité : 5 caractères → QUA\$
- * Prix H.T. : 5 caractères → PRI\$
- TOTAL : 37 caractères

Les variables ainsi définies devront avant toute utilisation être définies en mémoire pour leur réserver la place par ce type d'instruction :

10 ART\$ = ''':REF\$ = ''':QUA\$ = ''':PRI\$ = '''

Voyons maintenant les instructions utilisables et leur application dans notre exemple.

IFILES

Cette instruction permet de réserver une ou plusieurs zones de travail en mémoire pour le stockage temporaire des enregistrements, que l'on appellera « buffers d'enregistrement ».

Chacun de ces buffers portera un numéro que l'on appellera canal (souvenez-vous que l'on peut ouvrir jusqu'à 16 fichiers).

La syntaxe est la suivante :

IFILES,canalx,taillex,canalz,taillez,canalc,...

canalx sera un numéro de canal compris entre 0 et 127 et **taillex** dépendra de votre fichier ainsi que de la mémoire disponible.

Pour notre exemple nous pourrions écrire :

IFILES,1,37

notre fichier sera ouvert sur le canal 1 et comportera 37 caractères maximum.

Précisons que toute nouvelle commande **IFILES** annule la précédente et qu'elle doit s'effectuer une fois tous les fichiers fermés.

IOPEN

Nous pouvons ensuite ouvrir les fichiers à utiliser par la commande

IOPEN,« FICHIER.REL »,long,canal.

FICHIER.REL est le nom de notre fichier.

long est la longueur du plus grand des enregistrements qui doit être inférieure ou égale à la longueur maximale définie précédemment.

canal est le numéro de canal sur lequel nous ouvrons notre fichier.

Dans notre exemple :

IOPEN,« GESTION.REL »,37,1

IFIELD

Il nous faut maintenant diviser notre fichier en zones appelées champs, dans lesquelles seront enregistrées ou lues les différentes variables, ceci grâce à l'instruction :

IFIELD,canal,longvar1,longvar2,longvar3...

canal est le canal sur lequel le fichier a été ouvert

longvarx est la longueur maximale de chacune des variables utilisées pour le traitement du fichier.

Pour notre exemple :

IFIELD,1,20,7,5,5

IPUT

Cette instruction permet de sauvegarder un champ de données dans un fichier à accès direct.

La syntaxe est la suivante :

IPUT,canal,place, @var1\$, @var2\$, @var3\$...

canal est le numéro du canal sur lequel a été ouvert le fichier.

place est appelée numéro d'enregistrement, c'est l'endroit où l'on va ranger les données.

var1\$,var2\$... sont les noms des différentes variables qu'il faut sauvegarder dans le champ précédemment défini.

Il est possible de ranger des chaînes de caractères directement avec cette commande.

IPUT,canal,place,« chaine1 »,« chaine2 »,...

Dans notre exemple, nous aurons :

IPUT,1,5,@ART\$,@REF\$,@QUA\$,@PRI\$

pour ranger dans l'enregistrement N° 5 un article, sa référence, la quantité en stock et son prix.

IGET

Cette commande vous permettra de relire les données sauvegardées préalablement dans un fichier à accès direct, en accédant directement à un numéro d'enregistrement.

La syntaxe est :

IGET,canal,place,@var1\$,@var2\$,@var3\$,...

On se reportera au paragraphe précédent pour la définition des paramètres.

Les différentes variables devront obligatoirement avoir été définies précédemment par une instruction du type :

var1\$ = ''':var2\$ = ''':var3\$ = ''':...

car l'instruction utilise l'adresse de la variable dans sa routine de traitement (ce qu'indique le caractère @).

Pour notre programme nous aurons par exemple :

IGET,1,89,@ART\$,@REF\$,@QUA\$,@PRI\$

si nous voulons relire les différentes caractéristiques du 89-ième article.

CLOSE

Lorsqu'un fichier n'est plus en prévision d'utilisation, vous pouvez fermer le canal qui lui a été affecté par l'instruction :

ICLOSE,N° de canal

PROPOSITIONS DE PROGRAMMES D'APPLICATIONS

Les programmes que nous vous présentons sont volontairement succincts, ce ne sont que des ébauches que vous pourriez développer plus en détail, mais qui vous permettront de vous faire une *petite* idée de la puissance des fichiers à accès direct.

De plus, grâce à ce type de fichiers, vous n'êtes plus limités par la mémoire de votre machine, seuls quelques centaines d'octets vous permettront de mettre à jour, de trier, lire et imprimer, ... etc., des fichiers qui pourront contenir plusieurs dizaines, voir centaines, de k-octets de données ; la limite étant la capacité de votre disquette (708 k-octets !!) et le nombre d'enregistrements (65535 !).

• Gestion de stock

Voici un exemple d'enregistrement de fiches concernant des articles, ainsi que le programme de relecture associé. Vous pourrez améliorer ces programmes en développant une présentation appropriée à la gestion. Les deux programmes pourront être fusionnés et des procédures de calculs y être ajoutées (attention à la numérotation des lignes).

```

10 REM *** GESTION D'UN FICHIER DE ***
20 REM *** STOCK D'ARTICLE EN ***
30 REM *** EN MAGASIN PAR FICHIER ***
40 REM *** DE TYPE ACCES DIRECT ***
50 REM *****
60 REM *** INITIALISATION ***
70 @FILES,1,37
80 REM *** OUVERTURE DU FICHIER ***
90 @OPEN,"ESSAI.REL",37,1
100 REM *** PARTITIONNEMENT DU FICHIER EN
    N PLUSIEURS CHAMPS ***
110 @FIELD,1,20,7,5,5
120 REM *** PROGRAMME ***
130 INPUT "Nombres d'articles a entrer";
    nb
140 PRINT
150 REM *** ENTREE DES CARACTERISTIQUES
    DES ARTICLES ***
160 FOR i=1 TO nb
170 INPUT "Designation de l'article";ART$
180 REM *** TANT QUE L'ON A PAS FINI ***
190 IF UPPER$(ART$)="FIN" THEN i=i-1: GO
    TO 280
200 INPUT "REFERENCE";REF$
210 INPUT "Quantite en stock";QUA$
220 INPUT "Prix unitaire";PRI$
230 REM *** SAUVEGARDE DANS LE TAMPON DU
    FICHIER ***

```

```

240 ùPUT,1,i,àART$,àREF$,àQUA$,àPRI$
250 PRINT
260 NEXT i: REM *** PROCHAIN ARTICLE ***
270 REM FERMETURE DU FICHIER ET SAUVEGAR
DE DU TAMPON ***
280 ùCLOSE,1
290 STOP
300 REM *** SUITE DU TRAITEMENT ***

```

Pour terminer l'entrée des articles, frappez « fin » à la place du nom d'un article.

```

10 REM *** PROGRAMME DE LECTURE D'UN ***
20 REM *** FICHIER DE GESTION DE ***
30 REM *** STOCK EN ACCES DIRECT ***
40 REM *****
50 REM *** INITIALISATION DES VARIABLES
***
60 ART$="":REF$="":QUA$="":PRI$=""
70 REM *** INITIALISATION DU FICHIER ***
80 ùFILES,1,37
90 REM *** OUVERTURE DU FICHIER ***
100 ùOPEN,"ESSAI.REL",37,1
110 REM *** PARTITIONNEMENT DU TAMPON **
*
120 ùFIELD,1,20,7,5,5
130 REM *** PROGRAMME ***
140 INPUT "Numero de l'article a recher
her";no
150 PRINT
160 REM *** TANT QUE L'ON DESIRE LIRE UN
ARTICLE ***
170 IF no=0 THEN 280
180 REM *** LECTURE DU NUMERO D'ENREGIST
REMENT DEMANDE ***
190 ùGET,1,no,àART$,àREF$,àQUA$,àPRI$
200 PRINT"Article: ";ART$
210 PRINT"Reference ";REF$
220 PRINT"Quantite disponible ";QUA$
230 PRINT"Prix: ";PRI$
240 PRINT
250 REM *** ARTICLE SUIVANT ***
260 GOTO 140
270 REM *** FERMETURE DU CANAL 1 ***
280 ùCLOSE,1
290 STOP
300 REM *** SUITE DU TRAITEMENT ***

```

- *Tri sur fichier*

Nous n'allons pas développer les différentes méthodes de tri, mais juste en appliquer une sur un fichier à accès direct.

Nous utiliserons le « tri à bulles », appelé ainsi car il s'agit de faire remonter les éléments les plus légers (nous choisirons un tri par ordre croissant, qui devra donc faire remonter la plus petite valeur du fichier).

Considérons un fichier de plusieurs nombres à trier, appelé « NOMBRES.REL ».

Nous devons lire les valeurs deux à deux (une valeur et la suivante) ; les comparer ; si la première est supérieure, il faudra les permuter ; ainsi de suite jusqu'à ce que le fichier soit complètement trié.

On remarquera tout de même que le tri n'est pas très rapide mais il existe d'autres méthodes. L'avantage est de ne pas charger le fichier entièrement en mémoire centrale.

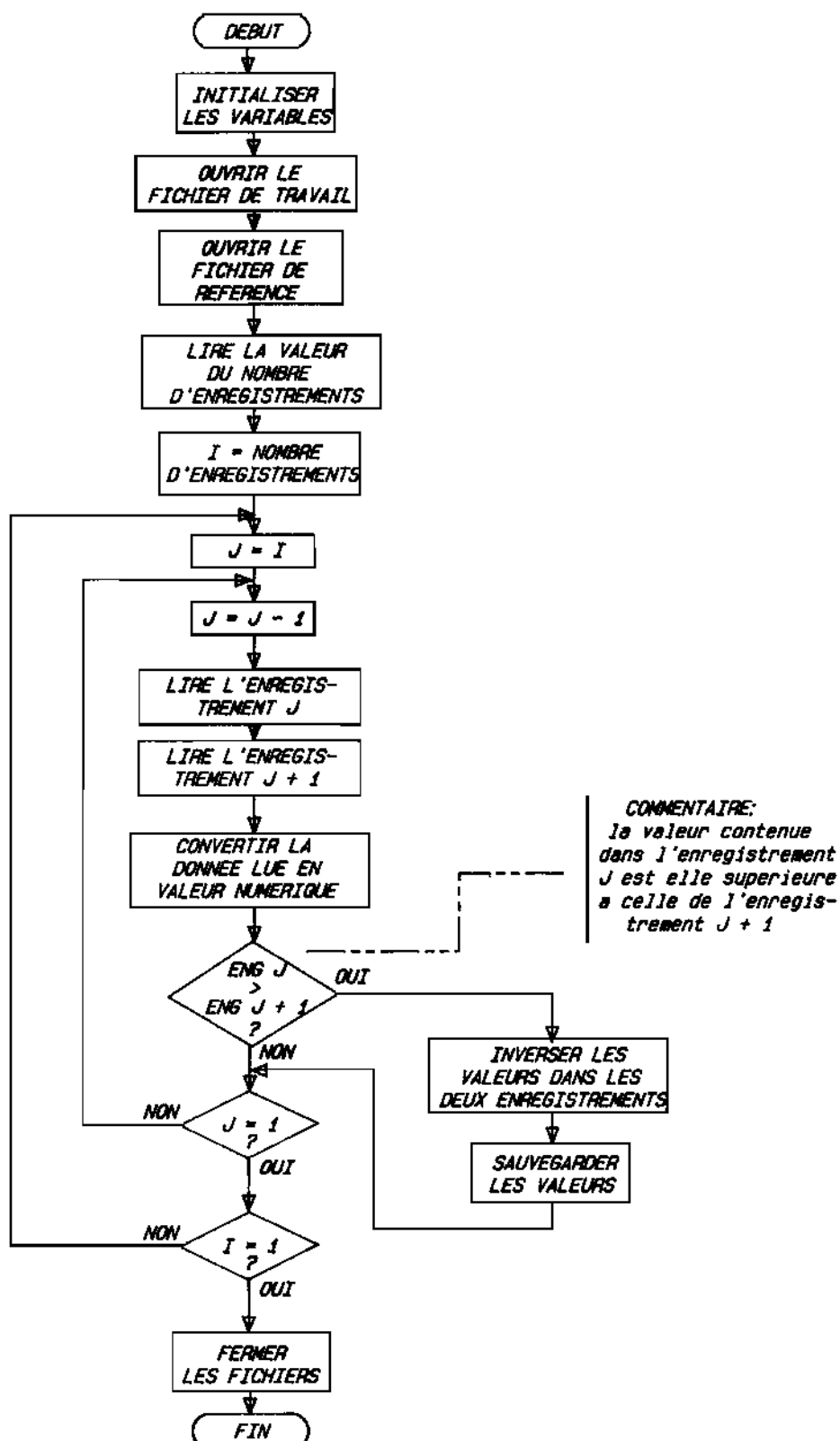
Vous pourrez tester le programme de tri en lançant ce programme qui génère aléatoirement un certain nombre de variables numériques. On remarquera que les variables sont transformées en chaîne de caractères avant sauvegarde.

```

10 REM ***   GENERATION DE VALEURS ALEATOIREMENT ***
20 REM ***   DANS UN FICHIER A ACCES DIRECT A TRIER ***
30 REM *****
40 REM ***   INITIALISATION ***
50 N$=""
60 ÛFILES,1,10
70 ÛOPEN,"FICHIER.REL",10,1
80 ÛFIELD,1,10
90 REM *** FICHIER DE REFERENCE CONTENANT LE NOMBRE DE VALEURS DU FICHIER A ACCES DIRECT ***
100 INPUT "Nombre de valeurs a generer";
Nb
110 OPENOUT "FICHIER.REF"
120 PRINT#9,Nb
130 CLOSEOUT
140 REM ***   GENERATION DES VALEURS ET SAUVEGARDE ***
150 FOR I=1 TO Nb
160 N=INT(RND*1000+1)
170 N$=STR$(N)
180 ÛPUT,1,I,N$
190 REM ***   VISUALISATION A L'ECRAN ***
200 PRINT N;" ";
210 NEXT I
220 ÛCLOSE,1
230 END

```

Algorithme du tri



Le programme de tri

```
10 REM *** TRI A BULLE SUR FICHER A ACC  
ES DIRECT ***  
20 REM *****  
*****  
30 REM *** INITIALISATION ET OUVERTURE D  
E FICHIERS ***  
40 n1$="":n2$=""  
50 ùFILES,1,10  
60 ùOPEN,"FICHER.REL",10,1  
70 ùFIELD,1,10  
80 OPENIN "FICHER.REF"  
90 REM *** LECTURE DU NOMBRE DE DONNEES  
***  
100 INPUT#9,nb  
110 REM *** CLASSEMENT ***  
120 REM *****  
130 FOR I=nb TO 1 STEP -1  
140 FOR J= nb-1 TO 1 STEP -1  
150 ùGET,1,J,àn1$  
160 ùGET,1,J+1,n2$  
170 n1=VAL(n1$)  
180 n2=VAL(n2$)  
190 REM *** TEST DE SUPERIORITE ***  
200 IF n1>n2 THEN GOSUB 290  
210 NEXT J  
220 NEXT I  
230 REM *** FIN DU TRI - FERMETURE DES F  
ICHIERS ***  
240 ùCLOSE,1  
250 CLOSEIN  
260 STOP  
270 REM *** SUITE DU PROGRAMME ***  
280 REM *** INVERSION DES DONNEES SI PAS  
DANS L'ORDRE ***  
290 n3=n2:n2=n1:n1=n3  
300 n1$=STR$(n1)  
310 n2$=STR$(n2)  
320 ùPUT,1,J,n1$  
330 ùPUT,1,j+1,n2$  
340 RETURN  
350 REM ***** FIN DE SOUS PROGRAM  
ME *****
```

- *Répertoire téléphonique*

Nous n'allons pas développer entièrement le programme de répertoire téléphonique, mais décrire la façon de créer votre répertoire et de le gérer.

D'abord, il nous faut définir les différentes données composant ce répertoire :

Nous aurons besoin d'enregistrer pour chaque personne inscrite, son nom, son prénom, son adresse, la ville, le code postal, et le numéro de téléphone. Il faut étudier le nombre d'octets que nous allons réserver, et les noms des variables :

nom	: NOM\$	15 octets
prénom	: ADR\$	25 octets
ville	: VIL\$	15 octets
code postal	: COD\$	5 octets
téléphone	: TEL\$	10 octets (pour les indicatifs)

Le fichier à ouvrir pourra s'appeler « REPERT.REL ».

Il serait peut-être intéressant de créer un fichier séquentiel contenant le nombre de personnes inscrites au répertoire, et si la mémoire ne vous manque pas, vous pourriez inscrire les noms et numéro d'enregistrement des inscrits afin d'augmenter la vitesse de recherche.

Le programme pourra aussi contenir une routine de tri en ordre alphabétique de vos correspondants potentiels.

Ce tri s'effectuera sur les noms et prénoms, vous devrez comparer une à une les lettres grâce à leur code ASCII, si deux lettres sont identiques, il vous faudra passer à la suivante et ainsi de suite jusqu'au prénom. Pensez à transformer toutes les chaînes de caractères en majuscule ou minuscule, car une différence de 32 sépare la valeur ASCII d'un caractère majuscule et celle d'une minuscule.

Remarques : en effectuant un catalogue de votre disquette 5 1/4, vous vous apercevrez que les fichiers ont une taille exprimée en multiples de 4, et au minimum 4 k-octets ; ceci est dû au fait que le module Vortex travaille avec des blocs supérieurs à ceux de l'AMSDOS.

Le catalogue par contre peut contenir jusqu'à 128 fichiers.

VI. Le moniteur V-DOS

Ce moniteur est accessible directement à partir du Basic en frappant l'instruction **IXMON** et **<RETURN>**

Lorsque vous serez sous XMON, l'AMSTRAD vous le signalera par le prompt « * ».

A ce moment, deux possibilités s'offrent à vous :

- travailler sous moniteur mémoire
- travailler sous moniteur disquette.

LE MONITEUR MEMOIRE

Cette partie de la description s'adresse aux fanatiques de la programmation en langage machine.

L'étude du langage machine dépasse l'objet de ce chapitre, aussi vous ne trouverez ci-dessous que la liste des commandes et leur signification.

A : permet d'entrer une chaîne de caractères.

Exemple :

A3000 <RETURN> puis chaîne de caractères à partir de l'adresse &3000 (hexadecimal).

B : montre les points d'arrêt (Break Point) et permet de les modifier.

Cadresse : xy permet de rechercher une séquence

Dadressesdébut,adressefin : permet de lister une zone mémoire comprise entre les adresses de début et de fin ainsi que le codage ASCII s'il est exécutable

Fadressedébut,adressefin,valeur : permet de remplir une zone mémoire avec une valeur

Gadresse : lance l'exécution d'un programme en code machine à partir de l'adresse

Iprogramme.bin : charge un programme en mémoire et vous indique l'adresse de chargement, la longueur et l'adresse de lancement

Ladressedebut,adressefin : désassemble une partie de la mémoire

Madressdébut, adressefin, adressearrivée : déplace une partie de la mémoire.

Oprogrammebin.,adressedebut,longueur,adressedelancement : sauvegarde une partie de la mémoire

P : agit comme CTRL P de CP/M en permettant l'impression sur imprimante

R : montre les registres et permet de les modifier

Sadresse : montre et permet de modifier le contenu de la mémoire. Presser <ESC> pour quitter cette possibilité.

Tadresse,nombre : permet d'exécuter un programme en langage machine au pas à pas

X : charge ou ne vide le firmware dans la zone &000-&3FFF

Y : sélectionne des extensions ROM

On quittera le moniteur mémoire par la frappe le <ESC> qui vous ramènera au basic.

LE MONITEUR DISQUETTE

Vous pouvez accéder à un intéressant moniteur disquette en frappant la commande @ (a commercial) à partir du moniteur mémoire.

Dans ce moniteur, voici les commandes auxquelles vous pouvez accéder :

L:xx : permet de changer le lecteur actif

S:xx : permet de se déplacer sur les secteurs (de &41 à &49 sur le lecteur 3'; de &01 à &09 sur 5'1/4)

T:xx : permet de changer de piste (« track ») (de &00 à &27 sur le 3' ; de &01 à &9F sur le 5'1/4)

B:adresse : sélectionne l'adresse du buffer

R : effectue la lecture de la zone désignée

W : écrit le contenu du buffer dans la zone disquette

Cadresse : permet de modifier

adresse : permet une chaîne de caractères

P : fonctionne comme pour le moniteur mémoire

Dadresse : effectue un « Dump » du contenu du buffer

Pour retourner au moniteur mémoire, il suffit de frapper

<ESC>

Voici un exemple d'utilisation de ce moniteur.

```
*à
L:00 T:00 S:01 B:A93E
>R
L:00 T:00 S:01 B:A93E
>D
A93E 11 00 03 21 00 00 06 04 CD 0B 02 30 34 10 F9 CD ...!.....04...
A94E C5 01 20 64 11 00 0C 06 10 7E B7 28 1A E5 C5 6F .. d.....".(....o
A95E 26 00 CD FC 01 30 1A C1 E1 23 10 ED 21 34 02 34 &....0...#...!4.4
A96E D5 CD C5 01 D1 2B E0 21 00 0C 0E FF CD 9B BE 16 .....(!.....
A97E BD 21 46 01 18 6B 0D 0A 44 69 73 63 65 74 74 65 .!F..k..Discette
A98E 20 72 65 61 64 20 66 61 69 6C 75 72 65 0D 0A 50 read failure..P
A99E 72 65 73 73 20 61 6E 79 20 6B 65 79 20 74 6F 20 res any key to
A9AE 72 65 73 74 61 72 74 FF 21 7D 01 1B 34 0D 0A 43 restart.!è..4..C
A9BE 61 6E 6E 6F 74 20 66 69 6E 64 20 22 2E 45 4D 53 annot find ".EMS
A9CE 22 20 66 69 6C 65 0D 0A 50 72 65 73 73 20 61 6E " file..Press an
A9DE 79 20 6B 65 79 20 74 6F 20 72 65 73 74 61 72 74 y key to restart
A9EE FF 7E 23 FE FF C4 5A BB 20 F7 CD 1B BB 3B FB CD .#...Z. ....8..
A9FE 1B BB CF 00 00 21 00 03 06 80 7E E6 F0 20 22 C5 .....!.....".
AA0E E5 11 0D 00 19 11 35 02 01 FF 0C 1B 1A 2B AE A1 .....5.....+..
AA1E 2B 09 1A FE 3F 20 0B 7E A1 12 AF 0E 7F 10 EC E1 (...? ..
AA2E C1 11 10 00 19 C8 19 10 D1 F6 FF C9 29 29 29 C5 .....))))).
AA3E 06 0B CD 0B 02 30 02 10 F9 C1 C9 C5 E5 01 F7 FF .....0.....
```

```

AA4E 3E 01 3C 09 38 FC 67 7D C6 0A 4F 2E 00 EB CD 89 >.<.B.gè..D.....
AA5E BE EB E1 C1 D0 14 14 23 C9 3F 3F 3F 3F 3F 3F .....#.???????
AA6E 3F 45 4D 53 01 CD EC 02 11 E4 08 CD EC 02 0E 01 ?EMS.....
AA7E CD 05 00 C3 00 00 00 43 31 30 43 50 4D 33 20 45 .....C10CPM3 E
AA8E 4D 53 01 00 80 48 12 13 14 15 16 17 18 19 1A 00 MS...H.....
AA9E 00 00 00 00 00 00 48 00 00 00 02 43 31 30 43 50 .....H....C10CP
AAAE 4D 33 20 45 4D 53 00 00 00 00 00 00 00 00 00 M3 EMS.....
AABE 00 00 00 00 00 00 00 00 00 00 00 00 00 02 43 .....C
AACE 50 4D 33 20 20 20 20 45 4D 53 01 00 00 48 01 02 PM3 EMS...H..
AADE 03 04 05 06 07 00 00 00 00 00 00 00 00 48 00 .....H.
AAEE 00 00 24 00 05 1F 03 B0 00 7F 00 80 00 20 00 02 ..$......
AAFE 00 02 03 01 50 09 01 00 02 2A 52 60 79 E6 FB F6 ....P....*R'y...
AB0E 20 C9 CD BB 3E E5 D5 C5 2B 2B 7E 21 D5 0A B7 3E ...>...+~!...>
AB1E 0C 28 02 3E 04 CD F2 0A 18 A3 0D 0A 0A 45 63 72 .(.>.....Ecr
AB2E 69 74 75 72 65 20 64 75 20 73 65 63 74 65 75 72 iture du secteur

```

```

L:00 T:00 S:01 B:A93E
>S
L:00 T:00 S:03 B:A93E
>P

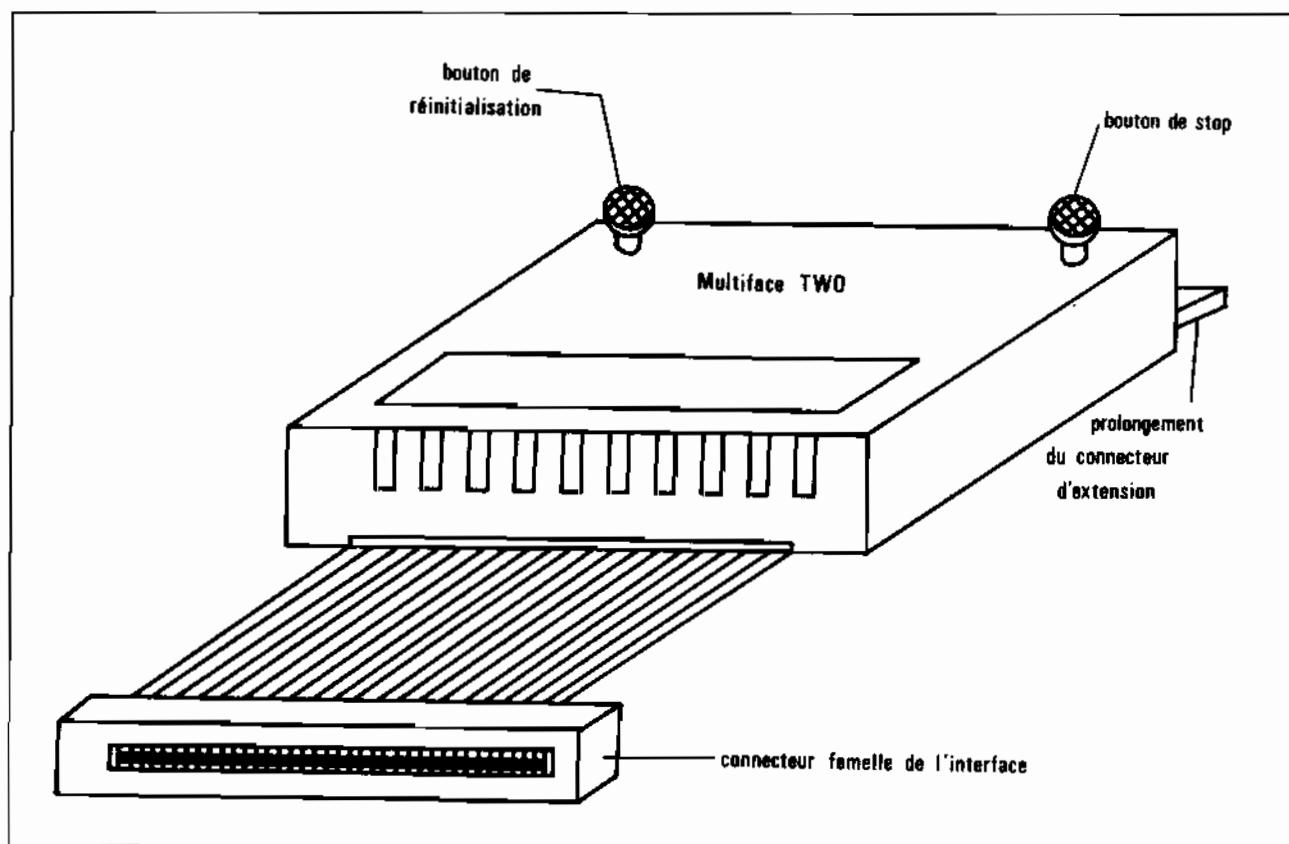
```

8/6.2

Multiface II, sauvegarde mémoire

Fabriquée par la société Romantic robot, l'interface Multiface II est un outil très intéressant à connecter sur votre Amstrad.

Conçue pour les trois types de CPC (464, 664 et 6128), elle permet d'effectuer automatiquement des sauvegardes mémoires sur cassette ou sur disquette, de « tricher » dans vos jeux passionnants, d'augmenter la capacité de votre CPC de 8 Ko de mémoire virtuelle, et de posséder un bouton de réinitialisation sans « bricoler » dans votre Amstrad ou encore de « fouiller » le système d'exploitation de votre ordinateur.



L'interface Multiface II et son connecteur.

INSTALLATION

L'installation de Multiface II ne devrait poser aucun problème particulier. Un connecteur femelle 2 x 50 broches s'enfiche sur le bus d'extension du CPC (dans le sens indiqué sur la figure).

Si vous possédez d'autres extensions, vous pouvez les connecter à l'arrière de Multiface II : Interface disquette pour CPC 464, interface RS-232C...

Attention : toutes ces manœuvres devront être effectuées hors tension sous peine de détériorer votre matériel.

UTILISATION

Dès la mise sous tension de votre ordinateur, Multiface II est active, mais complètement transparente vis-à-vis de vos logiciels.

Ré-initialisation

Qui ne s'est pas trouvé un jour confronté au problème de la ré-initialisation de son CPC, suite à un **POKE** malencontreux, ou encore pour passer d'un jeu à un autre, quand ce premier ne réagit pas à l'appui simultané sur **<SHIFT>** **<CONTROL>** et **<ESCAPE>** !

Le seul remède, brutal mais efficace, était de couper l'alimentation du CPC. Ce remède n'est pas sans mal, car il provoque des surtensions qui, à force de répétition, ne sont pas sans danger pour les composants.

Multiface II à l'arrière de votre Amstrad ? Dans ce cas, il suffit d'appuyer sur le bouton noté **RESET** pour retrouver devant soi la page d'accueil du Basic Locomotive.

Seul inconvénient à cette intervention : les 64 Ko de RAM sont effacés, par contre les 8 Ko de la mémoire propre à Multiface ne sont pas affectés.

Interruptions de programmes

Le deuxième bouton (noté **STOP**) de l'interface permet de stopper un programme en cours. Lors de l'appui sur celui-ci le menu suivant apparaît au bas de l'écran :

return save tool jump clear

Chacune des fonctions étant accessible par l'appui sur la touche ci-dessus soulignée.

A partir de ce moment, ne vous inquiétez pas si la page écran précédemment affichée est affectée, tant sur la forme des caractères que sur les couleurs, elle sera restituée correctement par la suite (si vous n'avez pas apporté de modifications à l'aide de l'utilitaire **tool**).

Return

Le choix de cette fonction permet de revenir au programme en cours, qui poursuit son déroulement comme si rien ne s'était passé.

Save

Supposons que vous soyez arrivé aux abords de la 23^e planète d'Alpha du Centaure, que vous ayez abattu 1752 vaisseaux ennemis, et ce avec un seul de vos trois croiseurs intergalactiques ; et paf ! vos trois vaisseaux se font désintégrer sous le feu ennemi sans que vous ayez eu le temps de parer à la manœuvre (ou plus sérieusement, si vous possédez un jeu d'échecs ne proposant pas le choix de sauvegarde en cours de partie). Seule alternative : relancer le jeu et recommencer la périlleuse aventure. Comment faire pour redémarrer au tableau où tout allait pour le mieux ?

Multiface II vous apporte la solution !

Appuyer sur le bouton STOP, puis choisir l'option **save**. Vous aurez alors à donner un nom de sept caractères au fichier, puis après avoir placé une disquette (de préférence libre et formatée), choisir **PROGRAM** puis **disc**.

Multiface se charge de compacter puis de sauvegarder toute la mémoire de votre CPC en plusieurs fichiers. Ces fichiers occuperont sur la disquette quelques dizaines de Ko.

Il est tout à fait possible de sauvegarder le contenu de la mémoire sur cassette (même en vitesse 3 000 bauds) en frappant sur **tape** ou **hypertape**.

Vous retournez ensuite au jeu par **return**.

Par la suite, pour reprendre une partie en cours, il vous suffira de lancer le programme par le nom que vous aurez donné à votre fichier de sauvegarde. Mais attention, il vous faudra laisser l'interface connectée.

Vous pourrez par contre effectuer des sauvegardes de pages écran réutilisables sans Multiface II. Pour la sauvegarde, pas de problème, suivez le menu en choisissant **SCREEN** mais pour récupérer l'image, il vous faudra frapper, en mode direct, ou dans un programme, les quelques instructions suivantes :

MEMORY &3FFF

LOAD "ECRAN.BIN"

CALL &8000

Clear

Si vous êtes possesseur d'un CPC 6128, vous pouvez utiliser cette option qui efface complètement les 64 Ko de mémoire supplémentaire afin de réduire la place occupée sur la disquette et améliorer la compression des programmes.

Attention si votre programme utilise cette deuxième « BANK » mémoire, ne pas utiliser cette option !

Jump

Cette option permet de se brancher à l'adresse d'une routine machine autre que le programme en cours.

Vous pourrez vous connecter aux adresses ROM ou RAM de l'Amstrad et même avoir accès aux 8 Ko de Multiface II.

Les adresses &2000 à &2003 contiendront dans l'ordre l'octet de poids faible puis de poids fort de l'adresse, la version ROM plus le mode écran, et la page RAM (uniquement pour CPC 6128, qui donne accès à quatre pages de 16 Ko de mémoire, éventuellement écran, commutables sur la page écran visible, de l'adresse &C000 à &FFFF).

Ce choix sera aussi bien sûr réservé aux passionnés de code machine qui sauront utiliser cette possibilité ainsi que la boîte à outils dont les explications suivent.

Tool

Le choix **tool** est réservé aux passionnés de code machine, le terme passionné n'étant pas de trop, car l'utilisation de cet utilitaire n'est pas très facile.

Ce choix offre le menu suivant :

<ESC> **RET**poke **SPC**addr**reg**hex**win**pal**info** **sel**

La touche <ESC> (ESCAPE) vous permettra de revenir au menu général de Multiface II.

La touche **hex** est une bascule permettant d'afficher les données en hexadécimal ou en décimal. La position initiale étant décimale.

Passons à la touche **SPC** qui correspond à la barre d'espace. Apparaît alors sur la dernière ligne le mot **adresse** vous invitant à entrer une adresse mémoire.

Sitôt cette adresse frappée (4 caractères hexadécimaux ou 5 décimaux), ou validée, son contenu est affiché. Vous pouvez soit modifier le contenu en frappant un nouveau, soit le valider par **RET**. **RET** seul provoque l'affichage de l'adresse suivante.

Vous pouvez visualiser les différents contenus d'adresses grâce à une fenêtre de 56 contenus (on appelle cela un « DUMP » mémoire), par la commande **win** qui agit comme une bascule (une nouvelle frappe provoquera l'effacement de la fenêtre). Il est même possible de se déplacer grâce aux touches curseur dans cette fenêtre pour effectuer des modifications.

reg permet lors de la frappe de **r** de visualiser les registres du microprocesseur Z-80 et/ou de les modifier. La touche **RETURN** permet le passage aux registres suivants. On pourra remarquer la valeur de l'adresse où ils sont temporairement stockés pour leur traitement par Multiface II : le premier registre débute en 3EE6 (Hexa).

sel permet de sélectionner les différentes bank mémoires du CPC 6128.

La commande **pal** permet d'afficher et/ou de modifier la palette des couleurs disponibles.

info offre des informations sur l'état actuel du système, et notamment sur certains états du contrôleur d'écran vidéo. Ces informations débutent à partir de l'adresse 3A80 et correspondent dans l'ordre à :

- 3A80 et 3A8F : référence CRT.
- 3A81 : nombre de caractères par lignes.

- 3A82 : synchronisation de la position horizontale.
- 3A8C et 3A8D : adresse de départ de l'écran.
- 3A91 et 3A92 : adresse écran pour le système des variables.
- 3A93 : version ROM et mode courant.
- 3A94 : page RAM (notée de C0 à C7).
- 3A95 : mode d'interruption (0 à 2).
- 3A96 : contient 0 si le mode d'interruption est désactivé, 1 sinon.
- 3A97 : type de ROM Basic.

Pour les novices, vous pouvez essayer de modifier le registre de l'adresse 3A81 en lui plaçant un nombre différent, puis quitter Multiface II par **<ESC>** et **return** pour admirer le résultat. Un conseil : si vous décidez d'explorer ces utilitaires prenez soin de retirer toute disquette du lecteur dont le contenu vous est précieux.

Multiface II est une interface pour Amstrad qui vous semblera intéressante si vous êtes un acharné du code machine, mais qui pourra vous rendre d'énormes services pour effectuer vos copies de sauvegarde, notamment en cours de jeux.

8/6.3

Brancher le lecteur du CPC 464 sur le CPC 6128

L'intérêt d'un deuxième lecteur de disquette sur le 6128 n'a pas à être démontré d'autant que certains logiciels ne fonctionnent pas de façon satisfaisante sans cette mémoire de masse supplémentaire, c'est par exemple le cas du compilateur dB COMPILER (voir Partie 7, Chap. 3.1.6).

La différence de prix entre le modèle FD1, destiné au 6128 et le modèle DDI-1 fonctionnant sur le 464 est suffisamment faible pour que l'on ait intérêt à acquérir ce dernier qui sera beaucoup plus facile à revendre par la suite. On peut aussi imaginer d'emprunter un lecteur de 464 à un ami équipé, le temps de quelques copies.

Le lecteur DDI-1 est effectivement un FD1 auquel on a ajouté un boîtier contenant la carte contrôleur qui fait défaut au CPC 464. Evidemment, cette carte contrôleur fait double emploi avec celle intégrée sur la carte mère du CPC 6128, et on ne peut pas relier le boîtier du contrôleur 464 sur un 6128 : il faut débrancher le contrôleur du lecteur DDI-1 et le remplacer par un câble de liaison allant directement au 6128.

Ce câble existe sous la référence DL2 (connecteur nez de carte) ou DL3 (connecteur Centronics présent sur certaines machines) et coûte une centaine de francs. Si on a des difficultés à se procurer le modèle DL2 et à condition d'être soigneux, on peut éventuellement le réaliser soi-même.

Il faut acheter :

- un connecteur type Multiflex HE10 2 fois 17 contacts femelle autodénudant équipé de deux détrompeurs mâles, qui sera branché au lecteur DDI-1 (par exemple référence AWP 34-7240 chez AM Distribution sur lequel il est nécessaire de supprimer l'unique détrompeur central à l'aide d'un cutter) ;
- un mètre de câble en nappe 34 conducteurs au pas de 1,27 mm (exemple : AWG 28-34/F) ;
- un connecteur encartable autodénudant au pas de 2,54 mm qui sera relié à la prise Disk Drive 2 du 6128 (exemple : AWDS 34-OF1).

On sertit le câble en nappe en veillant à ce que le contact 1 du lecteur soit bien relié au contact 1 de l'ordinateur, le 2 au 2, etc. On doit ensuite supprimer les contacts 21, 25, 29, 31 et 33 qui sont à + 5 V, il faut éviter de les relier puisque les deux appareils possèdent leur propre alimentation (voir Partie 8, Chap. 1, page 3 et Partie 2, Chap. 3.5, page 4) ; il suffit de couper les fils avec un cutter à 1 cm du connecteur après les avoir repérés.

On doit impérativement veiller à l'exactitude du sens d'insertion des deux connecteurs lors du branchement aux appareils, aussi est-il préférable de coller sur chacun une **étiquette** permettant par la suite le repérage facile du haut du connecteur.

Lors de l'utilisation, les voyants du lecteur interne et du lecteur externe s'allument alternativement et il y en a toujours un des deux qui reste allumé, même si la disquette n'est pas sollicitée, ce qui est normal.

