

**sinclair**

# ZX SPECTRUM

**Manuel du microdrive  
et de l'interface 1**

**IMPORTANT**

**NE JAMAIS CONNECTER OU DÉCONNECTER UN  
PÉRIPHÉRIQUE LORSQUE LE MICRO-ORDINATEUR  
SE TROUVE ALIMENTÉ.**

**DIRECO INTERNATIONAL DÉCLINE TOUTE  
RESPONSABILITÉ EN CAS DE NON RESPECT  
DE CETTE INSTRUCTION.**

**sinclair**

**ZX INTERFACE 1**

et

**ZX MICRODRIVE**

de Cambridge Communication Limited

**Remarque :**

Les noms Microdrive ZX et Interface ZX 1 sont des marques commerciales de  
Sinclair Research Limited  
Stanhope Road  
Camberley  
Surrey  
GU15 3PS  
Tél : Camberley (0276) 685311

Première édition 1983  
© Sinclair Research Limited  
Couverture de John Harris de Young Artists

L'interface ZX 1 ...

assure les trois fonctions de contrôleur de Microdrive, réseau local et interface RS232. Branchez-la à votre Spectrum et vous pourrez alors contrôler jusqu'à huit Microdrives, communiquer avec d'autres ordinateurs et alimenter une grande diversité d'imprimantes.

Le MICRODRIVE ZX ...

vous assure un accès rapide à une mémoire importante. Chaque Microdrive peut contenir jusqu'à 100K octets avec une seule cartouche interchangeable.

CE MANUEL...

présente l'utilisation du BASIC comme **système d'exploitation** et comme **langage de programmation**. Ainsi, on peut utiliser le BASIC pour établir des liaisons de communication, pour envoyer et recevoir des données et pour manipuler des fichiers. Cela correspond à un développement considérable de sa puissance pour exécuter des tâches où :

- des données sont stockées et extraites,
- des données ou des programmes sont envoyés vers des périphériques,
- des liaisons de communication sont établies avec d'autres ordinateurs Spectrum ou avec d'autres types d'ordinateurs.

La double utilisation du BASIC en tant que système d'exploitation et langage de programmation donne au BASIC la puissance, la souplesse et la facilité d'utilisation qu'on ne trouvait jusqu'ici qu'avec des ordinateurs familiaux beaucoup plus onéreux.

Nous vous recommandons de lire les chapitres relatifs au Microdrive même si vous n'en possédez pas dans la mesure où ils présentent divers concepts (concernant les **canaux** et les **voies** ainsi que leur utilisation) dont la compréhension vous sera utile.

TABLE DES MATIERES

Chapitre 1 Installation de l'Interface ZX 1	5
Chapitre 2 Installation du Microdrive	9
Utilisation des cartouches	
Protection des programmes et des données stockées sur cartouches	
Durée de vie des cartouches	
Chapitre 3 Démarrage du Microdrive	15
Exécution automatique	
Catalogue	
Chargement du programme	
Chapitre 4 Les programmes et le Microdrive	17
Sauvegarde, vérification, chargement et fusion de programmes	
Effacement de programmes	
Formatage et identification (nom) d'une cartouche vierge	
Installation de votre propre fonction d'exécution automatique	
Chapitre 5 Données, canaux et voies	21
Chapitre 6 Les données et le Microdrive	23
Ouverture et identification d'un fichier	
Introduction de données	
Fermeture d'un fichier	
Lecture d'un fichier	
Notes relatives à PRINT et INPUT	
- <u>séparateurs</u>	
- <u>changement de voie</u>	
- <u>détermination des couleurs</u>	
Lecture du catalogue des fichiers	
Protection d'un fichier	
Extension d'un fichier	
Chapitre 7 Le réseau local	29
Etablissement d'un réseau	
Les programmes et le réseau	
Les données et le réseau	
Diffusion	

Chapitre 8 Utilisation de l'interface RS232	35
Connexion des périphériques à l'Interface RS232	
Canaux t et b	
Le canal t	
Le canal b	
Envoi des codes de contrôle	
Chapitre 9 L'instruction MOVE	39
Le programme "serveur d'imprimantes"	
Annexe 1 Le jeu du réseau (net game)	41
Annexe 2 Les variables système	45
Annexe 3 Les canaux de Microdrive et de réseau	47
Annexe 4 Les connexions RS232	49
Annexe 5 Les messages d'erreur	51
Annexe 6 BASIC étendu	55
Index	59

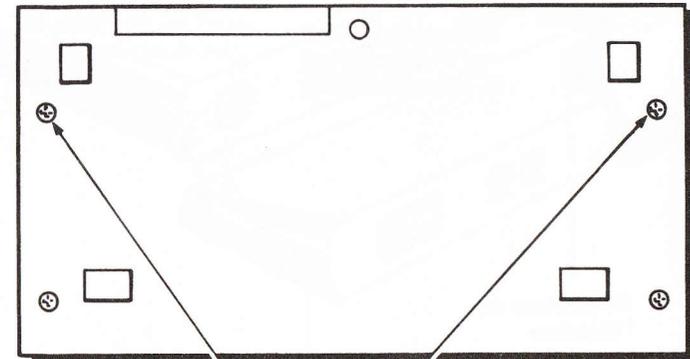
### INSTALLATION DE L'INTERFACE ZX 1

Au déballage, vous devez avoir :

- le présent manuel,
- l'interface elle-même (avec deux vis captives sur la face inférieure),
- un câble plat d'environ 8cm de longueur (pour la connexion d'une interface au Microdrive),
- un câble normal avec une prise jack à chaque extrémité (pour l'établissement d'un réseau)

Pour brancher l'interface sur votre ordinateur, vous devez :

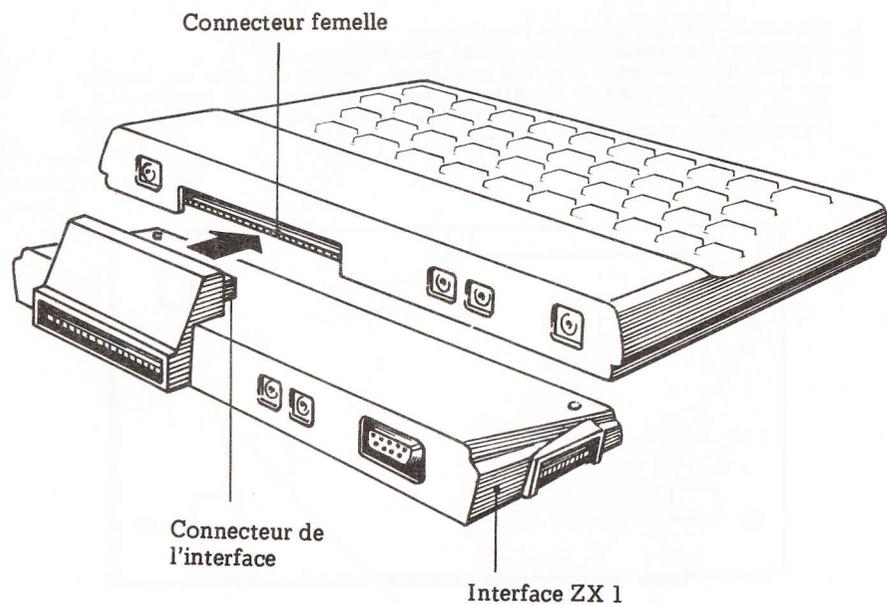
1. Débrancher l'alimentation de l'ordinateur.
2. Débrancher l'ordinateur de ses unités périphériques.
3. A l'aide d'un tournevis à pointe cruciforme, retirer les deux vis situées sur la face inférieure de l'ordinateur comme indiqué sur le croquis ci-dessous (vous ne réutiliserez ces vis que si vous débranchez ultérieurement l'interface de l'ordinateur.)



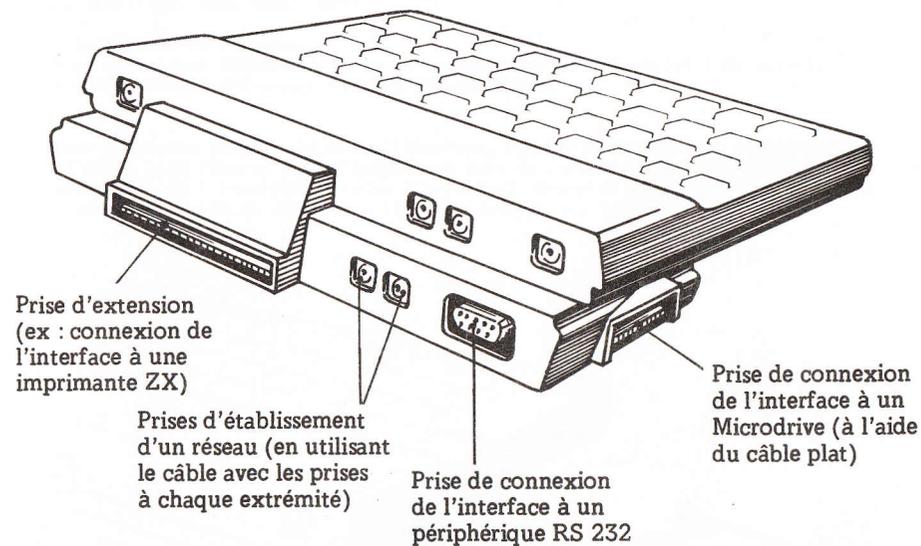
retirez ces deux vis mais  
laissez les autres en place

4. Introduire à fond le connecteur de l'interface dans le connecteur femelle à l'arrière de l'ordinateur.
5. Fixer l'interface sous l'ordinateur en utilisant les deux vis captives fournies.

Vous pouvez maintenant rebrancher l'ordinateur à ses unités périphériques et le brancher sur l'alimentation.



L'illustration ci-dessous présente la destination des diverses prises équipant l'interface.



Maintenant, vous pouvez soit :

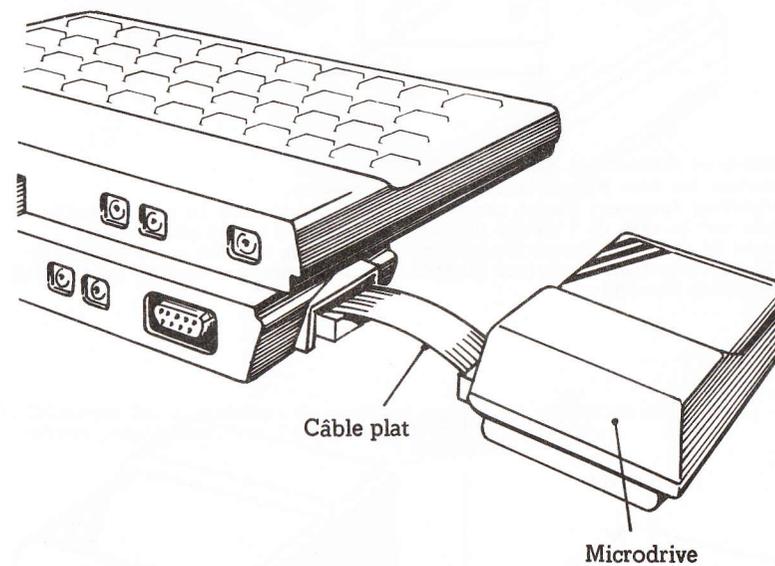
- installer un Microdrive voir page 9
- ou
- établir un réseau voir page 29
- ou
- connecter l'interface à un périphérique RS232 voir page 35

### Installation du Microdrive

Au déballage, vous devez avoir :

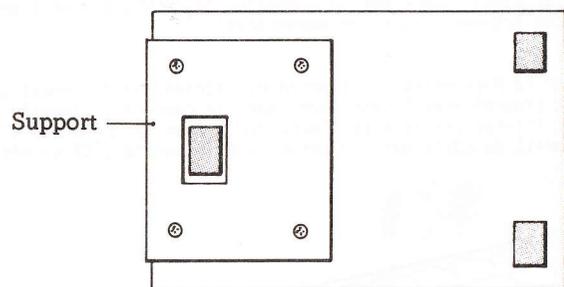
- le Microdrive lui-même,
- un connecteur électrique (pour la connexion de ce Microdrive à un autre),
- une cartouche pré-enregistrée de démonstration.

Pour connecter le Microdrive à l'interface, introduisez fermement un connecteur du câble plat (fourni avec l'interface) dans le connecteur femelle situé sur le côté droit de l'interface (vue de l'arrière). Branchez ensuite le connecteur de l'autre extrémité du câble dans la prise femelle sur le côté du Microdrive.

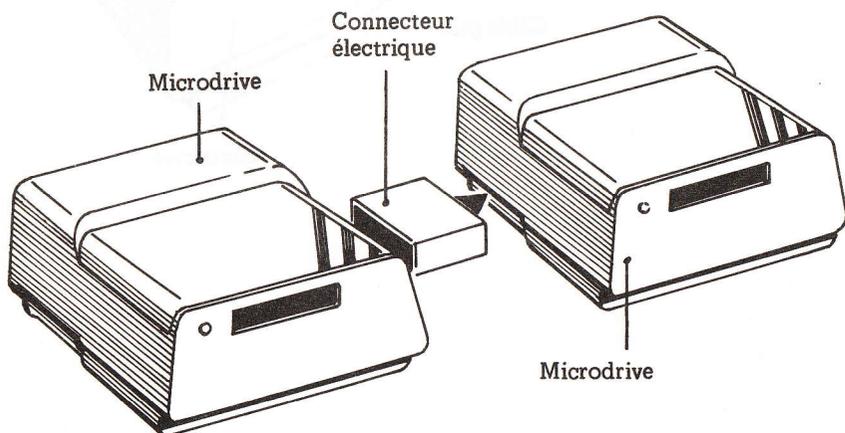


Si vous possédez plusieurs Microdrives, le deuxième peut être connecté en procédant comme suit :

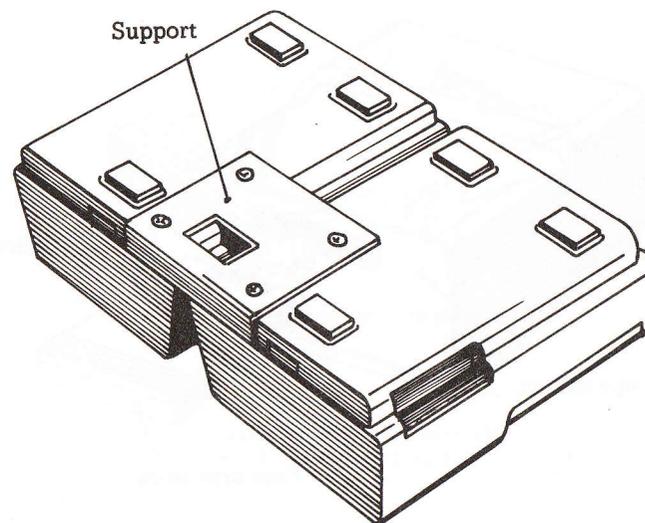
1. Débranchez le Microdrive de l'ordinateur.
2. Retournez les Microdrives. Chacun doit posséder un support vissé sur la face inférieure.



3. Dévissez et retirez les deux supports.
4. Retournez les deux Microdrives en position normale.
5. Introduisez fermement un des connecteurs électriques dans la prise femelle située sur le côté de l'un des Microdrives (comme illustré ci-dessous).
6. Appuyez le deuxième Microdrive contre le premier de manière que l'extrémité libre du connecteur électrique pénètre dans le connecteur femelle sur le côté du deuxième Microdrive.



7. Avec délicatesse, retournez encore une fois les Microdrives, et procédez au remontage de l'un des deux supports. Cette fois, le support doit établir la liaison mécanique entre les deux Microdrives (le connecteur électrique, le support et les vis restant seront conservés pour l'attachement d'une unité Microdrive supplémentaire.)



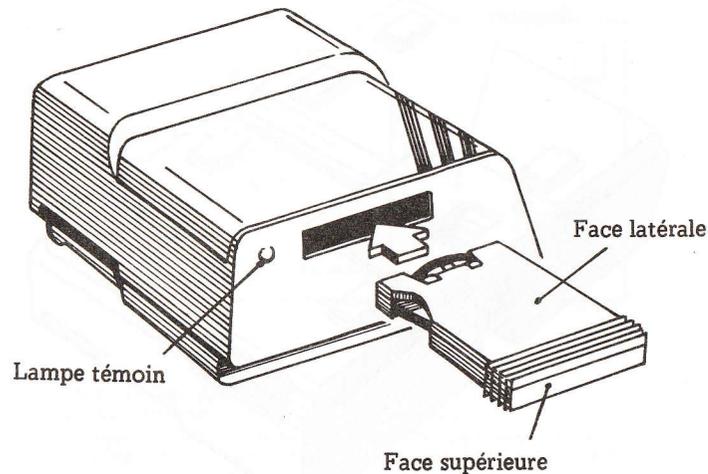
8. Retournez les Microdrives en position normale et rebranchez en un comme décrit précédemment sur l'ensemble ordinateur-interface.

Notez dès maintenant que le Microdrive se trouvant le plus près de l'ordinateur est toujours appelé Microdrive 1, le suivant Microdrive 2 et ainsi de suite. Remarquez aussi qu'à l'avant de chaque Microdrive se trouve une lampe-témoin. Celle-ci s'allume à chaque fois que le Microdrive entre en fonction.

**Utilisation des cartouches**

Chaque cartouche est livrée dans une boîte de protection que vous devrez toujours conserver pour ranger la cartouche non utilisée.

Lorsque vous sortez une cartouche de sa boîte (en veillant à ne pas toucher la bande magnétique), vous constaterez qu'elle possède une étiquette sur la face supérieure et une autre sur la face latérale.



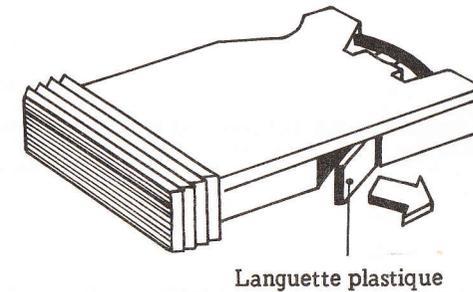
Introduisez à fond la cartouche dans la fente du Microdrive en veillant à ce que l'étiquette de la face latérale soit orientée vers le haut. En procédant de cette façon, la face supérieure de la cartouche doit dépasser d'environ 1/2 cm. Le Microdrive avec la cartouche est prêt à fonctionner.

Lorsque vous n'avez plus besoin de la cartouche, sortez-la sans brutalité du Microdrive et rangez-la immédiatement dans sa boîte. Mais attention...

NE JAMAIS SORTIR LA CARTOUCHE HORS DU MICRODRIVE LORSQUE LE TÉMOIN EST ALLUMÉ.  
NE JAMAIS BRANCHER OU DEBRANCHER L'ALIMENTATION QUAND UNE CARTOUCHE SE TROUVE DANS LE MICRODRIVE.

**Protection des programmes et des données stockés sur cartouche**

Si vous désirez protéger les programmes et les données mémorisés sur une cartouche (c'est-à-dire éviter les risques de sur-écriture ou d'effacement), il vous suffit de casser la languette plastique située sur le côté de la cartouche comme le montre l'illustration ci-dessous :



(Comme avec une cassette de magnétophone, vous obtiendrez l'effet inverse en plaçant un bout d'auto-collant à l'endroit où se trouvait la languette)

**Durée de vie des cartouches**

Les cartouches pour Microdrive ne sont pas inusables et il sera parfois nécessaire de les remplacer. Le premier symptôme de vieillissement d'une cartouche est le temps de plus en plus long que met l'ordinateur à trouver un programme ou un fichier avant de le charger. L'attitude la plus sage consiste donc à mettre de côté des copies de sauvegarde des programmes et des fichiers importants soit sur une autre cartouche soit sur une cassette.

### Démarrage du Microdrive

#### Exécution automatique

L'interface ZX 1 et le Microdrive étant installés, c'est l'instant idéal pour prendre connaissance des programmes qui ont été enregistrés sur la cassette de démonstration. Pour ce faire, introduisez cette cartouche dans le Microdrive (ou dans le Microdrive 1, si vous en avez plusieurs), et introduisez :

**NEW**

suivi de :

**RUN (et ENTER)**

Par cette commande, vous chargez et exécutez automatiquement le premier programme de la cartouche. Prenez connaissance de ce qui suit quand vous aurez vu ce qui se passe avec ce programme.

#### Le catalogue

Pour reconnaître les autres programmes mémorisés sur cette cartouche de démonstration, introduisez l'instruction CAT pour catalogue.

**CAT 1**

Le 1 identifie le numéro de l'unité Microdrive que vous utilisez.

Sept secondes après environ, l'écran affichera :

- le nom de la cartouche,
- un catalogue alphabétique des noms de tous les fichiers mémorisés sur la cartouche,
- la quantité de mémoire restant disponible dans la cartouche (en kilo-octets).

## Chargement des programmes

L'étape suivante consiste à charger (LOAD) le programme que vous voulez voir ensuite. Choisissez tout d'abord votre programme puis introduisez :

```
LOAD * "m";1;"nom"
```

\* : L'astérisque indique à l'ordinateur que vous utilisez un Microdrive et non un magnétophone à cassette ordinaire.

"m";1; : identifie le Microdrive que vous utilisez

nom : correspond au nom du programme que vous avez choisi

Après quelques instants, l'écran affiche le message OK (qui n'est pas le nom du programme). C'est à partir de ce moment que vous pouvez exécuter le programme.

## Les programmes et le Microdrive

## Sauvegarde, vérification, chargement et fusion de programmes

En lisant le livret d'introduction au ZX Spectrum, vous avez appris comment sauvegarder (SAVE) des programmes sur une cassette. La sauvegarde de programmes sur des cartouches est tout aussi facile. Là encore, nous prendrons un programme appelé Squares (carrés), ressemblant beaucoup à celui du livret d'introduction :

```
10 REM Carrés
20 FOR n = 1 TO 10
  PRINT n,n*n
NEXT n
```

Pour sauvegarder (SAVE) ce programme sur une cassette, vous devez introduire :

```
SAVE "Carrés"
```

Pour sauvegarder sur la cartouche du Microdrive 1, vous devez introduire :

```
SAVE * "m";1;"Carrés"
```

Après six secondes environ, le bord de l'image va se mettre à clignoter, ce qui signifiera que le programme aura été sauvegardé. Essayez pour voir.

(Les noms des programmes mémorisés sur une cartouche, comme ceux des programmes chargés sur une cassette, peuvent avoir jusqu'à dix caractères.)

Naturellement, vous pouvez vérifier maintenant que le programme a bien été sauvegardé en introduisant :

```
VERIFY * "m";1;"Carrés"
```

L'écran affichera ensuite le message OK.

Vous pouvez maintenant charger (LOAD) le programme Carrés en introduisant :

```
NEW
```

suivi de :

```
LOAD * "m";1;"Carrés"
```

puis :

```
ENTER
```

Si vous désirez que le programme s'exécute automatiquement, essayez d'introduire :

```
SAVE * "m";1;"Carrés 2" LINE 10
```

puis :

```
NEW
```

puis :

```
LOAD * "m";1;"Carrés 2"
```

On peut aussi fusionner des programmes en utilisant un Microdrive. Introduisez :

```
NEW
```

suivi de :

```
100 REM autres carrés
110 FOR n = 11 TO 20
120 PRINT n,n*m
130 NEXT n
```

Introduisez ensuite :

```
MERGE*"m";1;"Carrés
```

puis :

```
ENTER
```

Le programme Carrés sera ajouté à la liste.

En bref, comme vous avez déjà pu le constater probablement, toute la syntaxe utilisée dans une interface de cassette ordinaire (expliquée au chapitre 20 du Cours de Programmation BASIC) s'applique aussi au Microdrive.

#### Effacement de programmes

Supposons que vous veniez de terminer le programme Carrés et que vous ne vouliez plus l'utiliser. Pour l'effacer, introduisez :

```
ERASE "m";1;"Carrés"
```

(Comme précédemment, "m";1 indique simplement le Microdrive que vous utilisez).

La bordure de l'écran clignote pendant tout le processus d'effacement.

Si vous envoyez une interruption (BREAK) pendant la sauvegarde d'un programme, vous aurez alors un fichier non fermé dans la cartouche. Il vous est impossible de charger (LOAD) un fichier non fermé et toute tentative de chargement se traduira par l'affichage de file **not found** (fichier non trouvé). L'instruction ERASE permet de supprimer un fichier non fermé mais l'opération durera trente secondes environ parce que l'ordinateur doit vérifier le contenu de la cartouche plusieurs fois pour s'assurer que le fichier n'est pas fermé (END).

#### Formatage et identification des cartouches vierges

Pour utiliser une cartouche neuve, vous devez tout d'abord introduire dans un Microdrive (le Microdrive 1 en l'occurrence) et taper :

```
FORMAT "m";1;"nom"
```

"m";1 : identifie le Microdrive que vous utilisez (numéro 1, dans le cas présent)

nom : le nom que vous voulez voir apparaître dans le catalogue. Ce nom peut avoir jusqu'à dix caractères.

L'ordinateur met environ trente secondes pour formater une cartouche. Pendant le processus, le bord de l'écran se met tout d'abord à clignoter, s'efface puis clignote à nouveau et affiche finalement le message OK. Dans ce processus, l'ordinateur recherche toutes les zones de la cartouche qui ne peuvent être écrites ou lues et les marque d'un signe de reconnaissance.

Il est inutile de procéder plusieurs fois au formatage. Notez à l'occasion que le formatage d'une cartouche efface toutes les informations qu'elle peut contenir. Ainsi, si vous avez acheté plusieurs Microdrives, vous pouvez formater les cartouches de démonstration et les utiliser comme des cartouches vierges.

Tapez ensuite :

```
CAT 1
```

1 : identifie le numéro du Microdrive que vous utilisez.

Quelques secondes après, l'écran affichera le nom de la cartouche et sa capacité de mémoire. Cette capacité peut varier d'une cartouche à l'autre mais ne doit jamais être inférieure à 85 Ko.

### Installation de votre propre fonction d'exécution automatique

Au tout début, vous avez utilisé sans le savoir la fonction d'exécution automatique existant sur la cartouche de démonstration. Avec un programme que vous voulez exécuter souvent, vous pouvez définir votre propre fonction d'exécution automatique afin de n'avoir pas à retaper sans cesse les instructions **LOAD** et **RUN**. Pour cela ;

- il faut donner au programme le nom **run** ;
- la cartouche doit être introduite dans le Microdrive 1 ;
- la fonction automatique doit être utilisée soit immédiatement après la mise sous tension soit immédiatement après l'introduction d'un **NEW**.

En conséquence, introduisez le programme désiré puis :

```
10 SAVE * "m";1;"run" LINE numéro
```

**run** : doit être tapé en entier. Ne vous contentez pas d'appuyer sur la touche **RUN**.

numéro : vous devez introduire ici le numéro de ligne approprié.

Introduisez ensuite :

```
NEW
```

puis :

```
RUN (et ENTER)
```

**RUN** : est ici le mot-clé et non le nom de fichier.

Notez bien ici toutefois que l'instruction **MERGE** ne fonctionne pas en présence d'un quelconque programme sauvegardé avec l'instruction **SAVE\*...LINE...** (Cette disposition a été adoptée afin de protéger ces programmes). Toute tentative de fusion (**MERGE**) produira le message d'erreur "**Merge**" error.

### DONNEES, CANAUX ET VOIES

Un programme est une succession d'opérations qui seront exécutées dès que vous taperez **RUN**. Les données représentent un ensemble de lettres, de chiffres ou de symboles sur la base desquels un programme peut agir. Les chiffres de 1 à 10 avec leurs carrés en sont un exemple.

Les données peuvent circuler à destination et en provenance de divers éléments d'un système informatique. Ces éléments sont appelés des **canaux**. Les canaux vers lesquels vous pouvez envoyer des données sont :

- votre écran
- une imprimante ZX
- un fichier Microdrive pour les stocker
- un autre ordinateur ZX Spectrum, si les deux ordinateurs sont connectés à un réseau
- l'interface RS232 et à partir de celle-ci, par exemple, vers un modem ou une imprimante.

Les canaux d'où les données peuvent provenir sont :

- le clavier
- un fichier Microdrive
- un autre ordinateur ZX Spectrum, si les deux ordinateurs sont connectés à un réseau
- l'interface RS232, un modem ou un terminal.

Les axes de circulation des données en provenance et à destination de ces canaux sont appelés des **voies**. Dans le système informatique Spectrum, le nombre de ces voies est fixé à 16. Elles sont numérotées de 0 à 15 et les numéros de voies doivent toujours être précédés du signe # (symbole du mot numéro).

Quatre de ces voies sont déjà liées à des canaux :

voie # 0 **sortie** de donnée vers la partie inférieure de l'écran et  
voie # 1 **entrée** à partir du clavier

voie # 2 **sortie** vers la partie supérieure de l'écran, mais pas  
d'**entrée** ;

voie # 3 **sortie** via l'imprimante ZX ; mais pas d'**entrée**.

Chaque instruction donnant lieu à une entrée ou une sortie utilise automatiquement une de ces voies. Par exemple, l'instruction PRINT utilise la voie #2 et l'instruction LPRINT utilise la voie #3. Ainsi, quand vous entrez :

```
PRINT "C'est un ordinateur Spectrum"
```

Cette ligne est en fait une version abrégée de :

```
PRINT # 2;"C'est un ordinateur Spectrum"
```

Essayez d'introduire les deux pour voir ce qu'il se passe. Vous pouvez toutefois faire en sorte qu'une instruction quelconque utilise une voie différente en tapant # suivi du numéro de la nouvelle voie. Essayez d'introduire :

```
LPRINT # 2;"C'est un ordinateur Spectrum"
```

La ligne n'est plus imprimée sur l'imprimante ZX mais apparaît quand même à l'écran.

De même que vous pouvez utiliser les voies/canaux pré-établis, vous pouvez en créer de votre propre chef. Les voies #4 à #15 sont destinées à cet effet et plusieurs **spécificateurs de canal** (lettres réservées) vous permettent d'indiquer le périphérique que vous voulez. En voici quelques-uns :

```
"K" pour le clavier
"S" pour l'écran
"P" pour l'imprimante ZX
```

(Les autres vous seront présentés plus loin.)

Notez dès maintenant que les canaux K, S, P sont des canaux pré-établis. Vous devez obligatoirement utiliser des virgules (,) comme séparateurs dans les instructions OPEN #. Avec les autres canaux par contre, vous pouvez utiliser soit la virgule soit le point-virgule (;).

Pour créer vos propres ensembles voies-canaux, vous utiliserez l'instruction OPEN#. Par exemple, introduisez :

```
10 OPEN # 4,"S"
```

Ainsi, vous ouvrez la voie 4 et la liez au canal S. Introduisez maintenant :

```
20 PRINT # 4;"C'est un ordinateur Spectrum"
```

Là encore, la ligne apparaîtra à l'écran.

(En raison des résultats que l'on obtiendra, il est déconseillé d'ouvrir sur les voies 0, 1 ou 2.)

## LES DONNEES ET LE MICRODRIVE

### Ouverture et identification d'un fichier de données

A chaque fois que vous stockez des informations sur une cartouche, vous les conservez dans un fichier. Vous devez donner un nom à ce fichier de manière à pouvoir le retrouver par la suite. L'instruction qui vous permet d'ouvrir et de donner un nom (identification) à un fichier de données prend toujours la même forme. Par exemple, introduisez :

```
OPEN # 4;"m";1;"numéros"
```

4 : est le numéro de voie. Ce numéro peut être compris entre 0 et 15

"m";1 : identifie le Microdrive que vous utilisez.

"numéros" : est le nom du fichier. Le nom quelconque de votre choix peut avoir jusqu'à dix caractères de longueur.

Cette instruction remplit deux rôles :

- elle établit un nouveau canal : "m";1;"numéros"
- elle affecte ce nouveau canal à la voie # 4

Cette procédure prend sept secondes environ pendant lesquelles l'ordinateur explore la cartouche à la recherche d'un fichier appelé "numéros". Comme il n'en existe pas, il ouvre un fichier pour l'écriture. (Toutefois, s'il en trouve un appelé "numéros", il l'ouvrira pour la lecture. Et s'il trouve un programme "numéros", il produira le message **Wrong file type** (mauvais type de fichier).

### Introduction de données

Vous pouvez introduire des données dès que vous avez ouvert un fichier. Supposons que vous vouliez mémoriser les nombres de 1 à 10 avec leurs carrés. Introduisez et exécutez :

```
10 FOR n =1 TO 10
20 PRINT # 4,n'n*n
30 NEXT n
```

Vous pouvez penser que tous les nombres ont été stockés sur la cartouche. En fait, l'ordinateur ne transfère pas automatiquement chaque élément vers la cartouche tant qu'une certaine quantité d'informations n'a pas été amassée, laquelle est ensuite transférée en une seule fois. C'est ce qu'on appelle la **mise en mémoire-tampon** (ou tamponnage). Un tampon de Microdrive contient 512 octets (ou caractères).

Pour stocker sur la cartouche les données que vous avez introduites, vous devez fermer (CLOSE) le fichier. Vous ne pourrez rien lire du fichier tant que vous n'aurez pas fait cela.

## Fermeture de fichier

La procédure de fermeture d'un fichier permet d'assurer que le fichier est correctement sauvegardé sur la cartouche. Elle permet aussi de fermer le canal (dans le cas présent "m";1;"numéros") et laisse la voie (dans le cas présent, #4) libérée de tout canal. Pour fermer (CLOSE) un fichier, il vous suffit de fermer la voie appropriée. Ainsi, introduisez :

```
CLOSE # 4
```

La bordure de l'écran se mettra à clignoter pour indiquer que quelque chose est en cours de stockage sur la cartouche. (Remarquez que, comme l'instruction OPEN, l'instruction CLOSE est suivie automatiquement de #.)

A ce propos, ajoutons qu'il ne vous est pas possible de fermer (CLOSE) les voies # 0, # 1, # 2 ou # 3. Si vous le faites, les voies # 0 et # 1 prendront par défaut la valeur du canal K ; la voie # 2 la valeur du canal S ; et la voie # 3 le canal P (voir page 22).

## Lecture d'un fichier

Pour lire le fichier "numéros" exécutez le programme suivant :

```
10 OPEN # 4;"m";1;"numéros"
20 FOR b = 1 TO 10
30 INPUT # 4;m;n
40 PRINT"Le carré de ;m; "est"; n
50 NEXT b
60 CLOSE # 4
RUN
```

(Notez à ce niveau qu'en raison de l'existence du fichier "numéros", le canal "m";1;"numéros" est ouvert pour l'entrée et que toute tentative de sortie donnera lieu à une erreur.)

Vous pouvez également utiliser l'instruction INKEY\$ pour lire un fichier (cette instruction donne toujours le caractère suivant du fichier). Essayez le programme suivant :

```
10 OPEN # 11;"m";1;"liste"
20 LIST # 11
30 CLOSE # 11
40 OPEN # 12;"m";1;"liste"
50 PRINT INKEY$ # 12
60 GO TO 50
```

Ce programme se terminera sur la production du message End of file (fin du fichier).

## Notes relatives à PRINT et INPUT

Du fait que les instructions PRINT et INPUT sont conçues principalement pour être utilisées avec l'écran et le clavier, vous devez faire attention lorsque vous les utilisez avec des fichiers.

## Séparateurs

L'instruction PRINT utilise trois séparateurs différents :

- le point-virgule (;) n'imprime rien,
- la virgule (,) vous amène au début de la demi-ligne suivante,
- l'apostrophe (') donne une nouvelle ligne (le code ENTER)

L'instruction INPUT suppose toujours que vous allez taper ENTER après un nombre ou une chaîne. Ainsi, lorsque vous imprimez vers un fichier à partir duquel vous prévoyez d'introduire des données par une instruction INPUT, vous devez soit :

- imprimer les éléments séparément, exemple :

```
10 PRINT # 4;2
20 PRINT # 4;3
```

soit

- les séparer avec une apostrophe, exemple :

```
10PRINT # 4;2'3
```

Il faut aussi faire très attention lorsqu'on utilise des séparateurs dans une instruction INPUT. Comme vous l'avez déjà vu, l'instruction INPUT peut imprimer vers la partie inférieure de l'écran tout ce que vous pouvez introduire dans une instruction PRINT. Mais lorsque vous introduisez (INPUT) des données à partir d'un fichier, le fichier n'est ouvert que pour la lecture. Donc, si vous ajoutez quoi que ce soit qui puisse être imprimé lorsque vous utilisez l'écran, vous obtiendrez le message d'erreur Writing to a 'read' file (écriture vers un fichier en 'lecture') ; ce message signifiera que des éléments contenus dans l'instruction INPUT doivent être séparés par un point-virgule. Exemple :

```
10 INPUT # 4;a;b
```

Vous devez aussi faire très attention lorsque vous introduisez (INPUT) une chaîne contenant des guillemets (") parce que l'instruction INPUT considèrera que le " marque la fin d'une chaîne. Pour contourner ce problème, il vous suffit de remplacer, par exemple :

```
10 INPUT # 4;a$
```

par

```
10 INPUT # 4;LINE a$
```

Changement de voies

Les instructions **PRINT** peuvent aussi contenir des informations pour plusieurs voies à la fois. Le programme ci-dessous imprimera "un" sur l'écran, "deux" vers un fichier Microdrive appelé "chiffres" dans le Microdrive 1, "trois" vers la station 1 d'un réseau (reportez-vous au chapitre 7), et "quatre" vers la ligne suivante à l'écran.

```
10 OPEN # 4;"m";1;"chiffres"
20 OPEN # 5;"n";1
30 PRINT"un";#4;"deux";#5;"trois"#"2"quatre"
40 CLOSE # 4
50 CLOSE # 5
```

Détermination des couleurs

Après l'utilisation d'un canal autre que l'écran, vous pourrez constater que les instructions **PAPER** et **INK** restent sans effet. Pour éviter cela, introduisez :

```
PRINT;
```

avant d'introduire les instructions **PAPER** et **INK**.

Pour plus d'informations concernant **PRINT** et **INPUT**, reportez-vous au chapitre 15 du Cours de Programmation BASIC.

Lecture du catalogue des fichiers

Les fichiers que vous introduisez dans une cartouche sont catalogués automatiquement. Ainsi, lorsque vous voulez connaître les fichiers contenus dans une cartouche, il vous suffit d'introduire la cartouche dans un Microdrive et d'entrer l'instruction **CATALOGUE**. Par exemple, introduisez :

```
CAT 1
```

L'écran affichera :

- le nom de la cartouche,
- les noms de fichiers,
- la quantité (en kilo-octets) de mémoire restant disponible sur la cartouche.

Vous pouvez aussi envoyer la sortie de **CAT** vers une voie en introduisant :

```
CAT # numéro;numéro
      |
      v
introduisez ici le numéro de voie approprié
introduisez ici le numéro de Microdrive
```

Vous pouvez de cette façon envoyer le catalogue vers une imprimante ou vers un fichier pour l'exploiter avec le programme.

Protection d'un fichier

Si vous voulez qu'un nom de fichier n'apparaisse pas sur le catalogue, vous pouvez le protéger en lui donnant un nom commençant par le caractère dont le code est 0. Introduisez ce qui suit :

```
10 OPEN # 4,"m";1;CHR$0+"Résultats"
20 FOR n=1 TO 15
30 PRINT # 4;n,n*n
40 NEXT n
50 CLOSE # 4
```

Introduisez ensuite :

```
CAT 1
```

Le nom du fichier n'apparaîtra pas. Par conséquent, à chaque fois que vous voulez créer un fichier protégé, n'oubliez pas de noter son nom quelque part au cas où vous l'oublieriez plus tard !

## Extension d'un fichier

Supposons que vous vouliez étendre le fichier "Numéros" pour inclure les carrés des nombres de 1 à 20 au lieu de 1 à 10 seulement. Vous ne pouvez pas réouvrir un fichier pour l'écriture ; vous devez donc :

- créer une nouvelle version avec un nom différent,
- transférer l'ancien fichier vers la nouvelle version,
- ajouter les données supplémentaires,
- supprimer l'ancien fichier.

Voici comment il faut procéder :

Tout d'abord, exécutez (RUN) ce programme :

```
10 OPEN # 4;"m";1;"Numéros":REM pour lecture
20 OPEN # 4;"m";1;"Numéros 1":REM pour écriture
30 FOR f=1 TO 10
40 INPUT # 4;m;n
50 PRINT # 5;m'n
60 NEXT f
70 FOR n=11 TO 20
80 PRINT # 5;n'n*n
90 NEXT n
100 CLOSE # 4:CLOSE # 5
```

Maintenant, vérifiez que vous possédez bien deux fichiers "Numéros" et "Numéros 1" ; introduisez :

CAT 1

Ensuite, pour supprimer l'ancien fichier, introduisez :

ERASE "m";1;"Numéros"

Pour vérifier qu'il a bien été supprimé, introduisez :

CAT 1

Le nom de fichier "Numéros" doit avoir disparu du catalogue tandis que le nouveau fichier "Numéros 1" contient maintenant les nombres de 1 à 20.

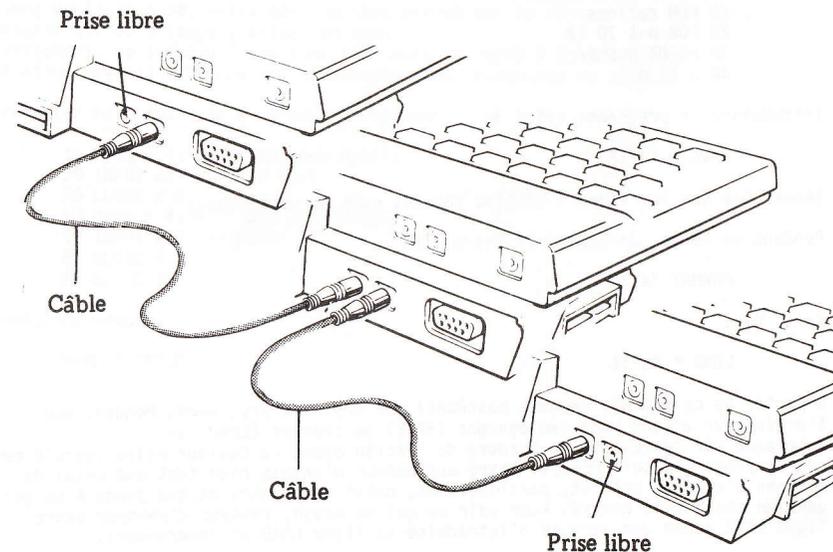
## LE RESEAU LOCAL

## Etablissement d'un réseau

Le réseau local (ou appelé simplement réseau) vous permet de jouer à des jeux informatiques avec vos amis et d'échanger avec eux des programmes et des données. Cela signifie, par exemple, qu'un seul d'entre vous peut se charger de la saisie au clavier d'un programme dont tous vos amis pourront profiter. Un réseau est aussi particulièrement utile si un seul d'entre vous possède un Microdrive.

A l'aide du câble fourni avec chaque interface, vous pouvez relier de 2 à un maximum de 64 ordinateurs Spectrums de la façon montrée ci-dessous.

Notez toutefois qu'il ne faut absolument pas constituer une boucle d'ordinateur : les ordinateurs à chaque extrémité du réseau ne doivent jamais être connectés ensemble. Ils doivent tous les deux être laissés avec une prise de réseau libre.



NE JAMAIS METTRE SOUS NI HORS TENSION UN SPECTRUM CONNECTE A UN RESEAU LORSQU'UNE COMMUNICATION EST EN COURS. Toutefois, un Spectrum peut très bien rester hors tension sur le réseau ; bien évidemment, vous pouvez aussi mettre en ou hors tension des Spectrums connectés à un réseau à condition qu'aucune communication ne soit en cours.

Lorsque vous avez établi un réseau, chaque ordinateur (ou **station**) doit posséder son propre numéro d'identification. Il faut donc tout d'abord décider ensemble le numéro de station devant correspondre à chacun après quoi chacun de vous devra introduire :

```
FORMAT "n";numéro
```

numéro : correspond au numéro de station choisi

Notez à ce propos que s'il n'y a que deux ordinateurs sur le réseau, les deux peuvent avoir le même numéro de station et, puisque des deux ordinateurs le premier mis sous tension devient automatiquement la station 1, il devient inutile d'utiliser l'instruction **FORMAT**.

### Les programmes et le réseau

Supposons que vous et votre ami Jacques ayez chacun un ordinateur branché sur le même réseau. Votre numéro de station est 1 et celui de Jacques est 2.

Supposons maintenant que vous vouliez envoyer le programme suivant à Jacques :

```
10 REM racines
20 FOR n=1 TO 10
30 PRINT n,SQRn
40 NEXT n
```

Introduisez le programme suivi de :

```
SAVE * "n";2
```

(Remarquez que le réseau n'utilise pas les noms de programme.)

Pendant ce temps, Jacques doit introduire :

```
FORMAT "n";2
```

suivi de :

```
LOAD * "n";1
```

A partir de ce moment, Jacques possèdera une copie du programme. Pendant que l'ordinateur attend pour sauvegarder (**SAVE**) ou charger (**LOAD**) un programme sur le réseau, la bordure de l'écran prend la couleur noire jusqu'à ce que le programme soit envoyé. Votre ordinateur n'envoie rien tant que celui de Jacques n'est pas prêt et, parallèlement, celui de Jacques attend jusqu'à ce que quelque chose soit envoyé. Pour voir ce qui se passe, essayez d'envoyer votre ligne **SAVE** avant que Jacques n'introduise sa ligne **LOAD** et inversement.

Pour s'assurer qu'il possède bien une copie de votre programme, Jacques doit introduire :

```
VERIFY * "n";1
```

Tandis que vous répétez l'envoi de votre programme en introduisant :

```
SAVE * "n";2
```

Il se passe en fait que l'instruction **SAVE** est la seule qui envoie des programmes sur le réseau. Les instructions **LOAD**, **VERIFY** et **MERGE** ne sont que des moyens différents de recevoir des programmes.

Le jeu de réseau présenté à l'annexe 1 est un bon exemple de la façon d'utiliser des programmes avec un réseau.

### Les données et le réseau

Supposons que vous vouliez maintenant envoyer des données à Jacques.

L'instruction **OPEN # 4;"n";2** ouvre un canal vers la station 2 sur le réseau ("**n**") et affecte la voie #4 à celui-ci ; quand vous envoyez votre sortie le long de la voie #4, votre message sera envoyé sur le réseau avec une note indiquant que ce message provient de vous.

(Inversement, en supposant que vous introduisiez **INPUT # 4;"n";2**, votre ordinateur devrait attendre des informations en provenance de Jacques.)

Introduisez maintenant le programme ci-dessous :

```
10 OPEN # 4;"n";2:REM pour sortie
20 INPUT a$: PRINT # 4;a$
30 CLOSE # 4
40 OPEN # 4;"n";2:REM pour entrée
50 INPUT # 4; b$:PRINT b$
60 CLOSE # 4
70 GO TO 10
```

Introduisez ensuite :

```
SAVE * "n";2
```

Demandez maintenant à Jacques d'introduire :

```
FORMAT "n";2
```

puis :

```
LOAD * "n";1
```

Vous pourriez alors exécuter votre programme et Jacques éditerait les lignes 10 et 40 du programme pour qu'elles fassent référence à la station 1 et non à la station 2. Il devrait introduire ensuite :

```
GO TO 40
```

Vous seriez à partir de cet instant prêt à commencer une conversation. Mais prenez connaissance des trois remarques suivantes avant de passer à ce niveau.

- Votre sortie par la voie # 4 est **tamponnée** : en d'autres termes, elle n'est pas envoyée automatiquement sur le réseau tant qu'une certaine quantité de données en sortie n'a pas été accumulée. Donc, à chaque fois que vous terminez une sortie de données ou d'informations, vous pouvez fermer (CLOSE) la voie. Ceci permettra d'envoyer le contenu du tampon même s'il n'est pas plein (les tampons ont une capacité de 255 octets ou caractères.)

- Votre sortie contient une marque spécifiant qu'elle provient de **vous**, de sorte que si Jacques procède à l'introduction de données ou attend des informations d'un canal différent du réseau, votre message sera ignoré. Si votre message a été ignoré, votre écran n'affichera pas le message OK et la bordure de l'écran ne deviendra pas noire tant que Jacques n'indiquera pas qu'il est prêt à recevoir le message.

- Bien que l'instruction **INPUT** attend simplement l'envoi de quelque chose, l'instruction **INKEY\$** peut être utilisée pour lire le réseau. Elle sera ensuite renvoyée avec le premier octet de n'importe quelle donnée ayant été envoyée ou en attente d'envoi. (Sinon, elle sera renvoyée avec la chaîne vide.) Cette procédure s'appelle le polling) ou initiation à émettre. (En fait, **INKEY\$** se comporte exactement de la même façon avec le réseau qu'avec le clavier.)

Le programme ci-dessous imprimera tout ce qui lui sera envoyé par la station 1 :

```
10 OPEN # 8;"n";1
20 PRINT INKEY$ # 8
30 GO TO 20
```

(Pour plus d'informations sur **INKEY\$**, voyez le chapitre 18 du Cours de Programmation BASIC).

### Diffusion

Il existe aussi un canal de réseau spécial appelé **diffusion** dont le spécificateur de canal est "n";0. Lorsque vous introduisez des informations à partir de ce canal, vous "attrapperez" n'importe quel message en cours de diffusion. Et, quand vous envoyez un message, celui-ci peut être lu par quiconque envoie ses informations à partir du canal "n";0. Cette disposition peut être très utile, à l'école par exemple, si chaque élève d'une classe possède un ordinateur Spectrum mais que seul le professeur possède un Microdrive.

Supposons le cas où le professeur veut diffuser un programme. Tout d'abord, les élèves doivent introduire :

```
LOAD * "n";0
```

Les élèves se mettront ainsi en attente pour recevoir le programme. Le professeur **diffusera ensuite le programme en introduisant :**

```
SAVE * "n";0
```

Comme vous pouvez le supposer, les messages ou programmes diffusés (à la différence des messages privés) sont envoyés tout d'une fois et n'attendent pas que les autres ordinateurs soient prêts à les recevoir.

Notez à ce propos que lorsque vous envoyez une diffusion, l'ordinateur ne vous permet pas de savoir si chacun a reçu le message.

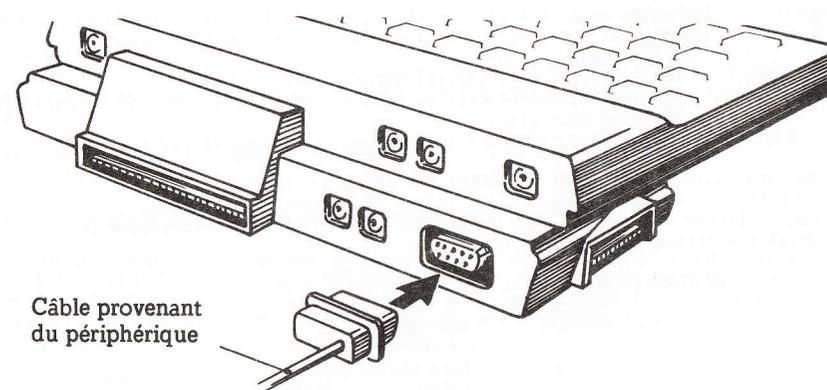
Notez également que **INKEY\$** ne permet pas d'appeler (poll) le canal de diffusion. Comme **INPUT**, il attendra simplement que quelque chose soit envoyé.

UTILISATION DE L'INTERFACE RS232

Connexions des périphériques à l'Interface RS232

Comme vous le savez, le jeu de caractère du Spectrum comprend à la fois des symboles uniques (lettres, chiffres, etc.) et des caractères spéciaux composites (mots-clés, noms de fonction, etc). Tous ces caractères peuvent être envoyés et reçus par l'intermédiaire de l'interface RS232 à destination et en provenance de n'importe quel périphérique série compatible pouvant être, par exemple, une imprimante, un modem ou une autre interface RS232 connectés à un type d'ordinateur différent.

Pour connecter un de ces périphériques à la RS232, vous devez utiliser un câble possédant à une extrémité une prise D à 9 broches et une prise adaptée à votre périphérique à l'autre extrémité. Vous devez introduire la prise D à l'arrière de l'interface RS232 comme montré ci-dessous. (Pour plus de détails de câblage, voir l'annexe 4, page 49.)



Ensuite, avant de pouvoir utiliser la RS232, vous devrez procéder à quelques réglages sur le périphérique. Ce peut être :

- le saut de ligne automatique qui doit être normalement coupé. (Le Spectrum envoie un retour chariot et un saut de ligne sur un canal "t" mais un retour chariot uniquement sur un canal "b". Ces canaux t et b sont expliqués plus loin).
- la parité qui doit être forcée à 0,
- le nombre de bits puisque le Spectrum utilise 8 bits,
- le nombre de bits d'arrêt puisque le Spectrum en met un.

Chapitre 8

- le débit en **bauds** (c'est-à-dire le nombre de bits par seconde). Le Spectrum peut fonctionner à tous les débits standard : 50, 110, 300, 600, 1200, 2400, 4800, 9600, et 19200.

Vous devez régler votre périphérique pour utiliser le débit le plus élevé dont il est capable (vous verrez ultérieurement comment régler le Spectrum pour utiliser le même débit.)

Vous trouverez l'explication des réglages dans le livret d'instruction fourni avec votre périphérique.

Les canaux t et b

La RS232 utilise deux canaux différents : Le canal t et le canal b.

Le canal t (pour texte) est normalement utilisé pour envoyer des listages. Le canal t a l'effet suivant sur le jeu de caractères :

- 0-31 (les codes de contrôle) ne sont pas envoyés à l'exception du 13 (retour chariot) qui est envoyé sous la forme 13 suivi de 10 (retour chariot et saut de ligne).
- 32-127 (les codes ASCII) sont envoyés normalement.
- 128-164 (les caractères graphiques) ne sont pas envoyés. Le message ? (code 63) est envoyé à leur place.
- 165-255 (les caractères spéciaux) sont étendus aux codes 32-127 ASCII.

Sur les instructions **INPUT** et **INKEY\$**, le canal t attend des caractères à 7 bits de sorte qu'il ignore le huitième s'il existe. Pour utiliser le canal t, vous devez tout d'abord dire à l'ordinateur le débit à utiliser. Introduisez donc :

```
10 FORMAT "t";débit en bauds
      |
      | introduisez ici
      | le débit que vous
      | avez choisi pour
      | votre
      | périphérique.
```

Maintenant, pour ouvrir (**OPEN**) une voie vers celui-ci, introduisez :

```
20 OPEN # 3;"t"
30 LLIST
```

La bordure de l'écran clignote et le listage est envoyé vers le périphérique. (notez que **LLIST** est l'annotation abrégée de **LIST # 3**). Introduisez maintenant :

```
LPRINT "ceci est un message court"
```

Ce message sera aussi envoyé vers le périphérique.

Si votre interface ZX1 est connectée à un terminal ou à un périphérique RS232 capable d'envoyer des informations, vous pouvez alors envoyer (**INPUT**) des informations à partir du terminal ou du périphérique. Introduisez ce qui suit :

```
10 FORMAT "t";débit en bauds
20 OPEN # 4;"t"
30 PRINT INKEY$ # 4;
40 GO TO 30
```

Introduisez ici le débit que vous avez choisi pour votre périphérique.

A partir de cet instant, tout ce que vous taperez sur le terminal ou le périphérique apparaîtra sur votre écran.

Le canal b (pour binaire) envoie les codes complets à huit bits utilisés par le Spectrum et vous permet d'envoyer des codes de contrôle vers des imprimantes, etc.

Avec les instructions **INPUT** et **INKEY\$**, le canal b attend des caractères à huit bits.

Les instructions **SAVE** et **LOAD** ne marchent qu'avec le canal b.

Si vous utilisez des modems pour connecter votre ordinateur à un autre Spectrum par l'intermédiaire d'une ligne téléphonique ou si vous stockez vos programmes sur un ordinateur différent possédant aussi une interface RS232, vous pouvez très bien sauvegarder (**SAVE**) et charger (**LOAD**) par l'intermédiaire de l'interface RS232. Pour ce faire, introduisez :

```
FORMAT"b";débit en bauds
      |
      | introduisez ici le débit que
      | vous avez choisi pour votre
      | périphérique.
```

A présent, par exemple, introduisez :

```
10 REM chiffres
20 FOR n=1 TO 10
30 PRINT n,n*RND
40 NEXT n
```

suivi de :

```
SAVE * "b"
```

Si ce programme est envoyé par l'intermédiaire d'un modem, quelqu'un à l'autre extrémité doit introduire :

```
LOAD * "b"
```

Les extensions habituelles telles que ci-dessous sont également autorisées :

```
SAVE * "b";SCREEN $
```

et

```
SAVE * "b";LINE numéro
```

#### Envoi de codes de contrôle

De nombreuses imprimantes vous permettent d'envoyer des instructions pour diverses opérations telles que l'impression en double largeur, etc. Pour les envoyer, vous devez utiliser le canal b. Toutefois, avec le canal b, le retour chariot ne comprend pas de saut de ligne. Vous devrez donc soit changer les réglages de votre imprimante soit imprimer le code de saut de ligne : 10. Par exemple, supposons que le code de contrôle pour l'impression en double largeur soit 14. Introduisez :

```
OPEN # 4;"b"
20 PRINT # 4;"largeur normale
30 PRINT # 4;CHR$14;"double largeur
40 CLOSE # 4
```

(Pour savoir les codes de contrôle à utiliser, reportez-vous au livre d'instruction de votre imprimante.)

Vous pouvez avoir les deux canaux b et t ouverts en même temps. Essayez d'introduire le programme suivant :

```
10 OPEN # 5;"b"
20 OPEN # 6;"t"
30 PRINT # 5;CHR$ 14;
40 LIST # 6
50 CLOSE # 5;CLOSE # 6
```

Ce programme doit vous donner un listage en double largeur.

#### L'INSTRUCTION MOVE

Jusqu'ici, le seul déplacement que vous ayez pu faire était de transférer des données d'un programme vers un canal ou inversement. Toutefois, l'instruction **MOVE** vous permet de déplacer des données d'un canal vers un autre. Par exemple, pour déplacer des données du clavier vers l'écran, introduisez :

```
10 MOVE # 1 TO # 2
```

puis :

```
RUN
```

Tout ce que vous tapez au clavier apparaîtra maintenant à l'écran. Cependant, vous découvrirez que quand vous appuyez sur **BREAK**, vous n'obtenez qu'un espace à l'écran. Pour éviter ce piège, appuyez sur **ENTER** jusqu'à ce que la position d'impression atteigne le bas de l'écran. Vous appuierez ensuite sur **BREAK** lorsque l'ordinateur demandera **scroll?** (défilement?). (A ce propos, il résulte que vous devez éviter de déplacer des données du clavier vers une quelconque autre voie puisque vous risquez d'être incapable d'interrompre (**BREAK**) un tel mode.

En utilisant l'instruction **MOVE**, vous pouvez aussi examiner des fichiers enregistrés sur cartouche. Par exemple, établissez le fichier "Numéros" (voir page 23) puis, pour vérifier son contenu, introduisez :

```
10 MOVE "m";1;"Numéros" TO # 2
```

(Remarquez que vous n'avez plus besoin d'ouvrir (**OPEN**) ou de fermer (**CLOSE**) le fichier vous-même puisque **MOVE** le fait à votre place)

De même, pour faire une copie du fichier "Numéros", introduisez :

```
10 MOVE "m";1;"Numéros" TO "m";1;"Numéros 2"
```

Dans ce cas, **MOVE** ouvre une voie pour la lecture du fichier existant ("Numéros") et une autre pour l'écriture vers le nouveau fichier ("Numéros 2"). Elle lit ensuite les données dans "Numéros" et les écrit dans "Numéros 2". Elle ferme ensuite les deux voies.

L'instruction **MOVE** fonctionne avec les numéros de voies (tels que # 4) et avec les spécificateurs de canal (tels que "m";1;"Numéros"). Notez toutefois que les voies établies # 1 à # 3 peuvent ne pas être spécifiées par les spécificateurs de canal K, S ou P.

Si vous avez un deuxième Microdrive, vous pouvez utiliser l'instruction **MOVE** pour faire des copies de sauvegarde de données sur une autre cartouche. Introduisez :

```
10 MOVE "m";1;"Numéros" TO "m";2;"Numéros 2"
```

(Notez que l'instruction **MOVE** ne fonctionne qu'avec des fichiers de données. Si voulez une copie de sauvegarde d'un programme, vous devez charger (**LOAD**) le programme puis le sauvegarder (**SAVE**).)

## Chapitre 9

L'instruction **MOVE** peut aussi envoyer des fichiers vers une imprimante. Ainsi si vous possédez une imprimante ZX, introduisez :

```
10 MOVE "m";1;"Numéros" TO # 3
```

## Le programme "serveur d'imprimante".

Ce programme permet à un Spectrum connecté à un réseau de contrôler une imprimante RS232. L'imprimante peut alors être utilisée par tous les autres ordinateurs du réseau. Cette solution est pratique, par exemple, si des personnes utilisant des Spectrums n'ont qu'une seule imprimante qualité courrier qu'elles souhaitent partager. Le programme présente aussi une utilisation puissante de l'instruction **MOVE**.

L'ordinateur serveur d'imprimante doit toujours être la station 64 et doit toujours se connecter avec une station 62 (qui est une station établissant le contact). Ainsi, l'expéditeur utilise temporairement la station 62 et lui envoie ensuite son numéro réel de station à partir de laquelle il transfère ensuite un fichier vers le canal t. Pour établir une station "serveur d'imprimante", utilisez le programme ci-dessous :

```
10 FORMAT "n";64
20 OPEN # 4;"n";62: INPUT # 4;a$:CLOSE # 4
30 MOVE "n":CODE a$ TO "t"
40 OPEN #4:"b":PRINT # 4;CHR$ 12: CLOSE # 4: RUN
```

(La ligne 40 envoie un saut de page.)

Le programme présenté ci-dessous est l'un de ceux utilisé ensuite par l'expéditeur. En premier lieu, la station de l'expéditeur est temporairement affectée du numéro 62. En deuxième lieu, le numéro réel de station de l'expéditeur est envoyé. En troisième lieu, la station de l'expéditeur reprend elle-même son numéro réel. Enfin, la ligne 60 envoie les données à imprimer (dans le cas présent, le listing).

```
10 LET station=numéro
      |
      | introduire ici le numéro
      | réel de station de
      | l'expéditeur
20 FORMAT "n";62
30 OPEN # 4;"n";64:PRINT # 4 CHR$ station: CLOSE # 4
40 FORMAT "n";station
50 OPEN # 4"n";64
60 LIST # 4
70 CLOSE # 4
```

## LE JEU DU RESEAU

Une copie de ce jeu figure sur la cartouche de démonstration fournie avec le Microdrive. Son nom de fichier est "**net game**". Ce programme est un bon exemple de la façon dont on peut utiliser le réseau. Vous pourrez utilement intégrer des parties de celui-ci dans des programmes que vous aurez créés.

## Le jeu

Pour jouer à ce jeu, vous et votre adversaire devez chacun penser à un nombre compris entre 1 et 100. Le gagnant du jeu est celui qui parvient le premier à deviner le nombre choisi par l'adversaire. A chaque proposition que vous faites, l'ordinateur vous dit le degré de proximité par rapport au nombre à deviner.

## Le programme

Le sous-programme à la ligne 500 décide qui est l'utilisateur 1 et qui est l'utilisateur 2. Cette disposition a été prise afin que, lorsque vous échangez des hypothèses, l'un de vous utilise le sous-programme à la ligne 1100 et l'autre utilise le sous-programme à la ligne 1200 ; donc l'utilisateur 1 envoie son hypothèse en premier et l'utilisateur 2 reçoit l'hypothèse le premier.

Votre ordinateur décide qui est l'utilisateur 1 en envoyant à l'ordinateur de votre adversaire le message "1" et attend une réponse. S'il reçoit en retour un "1", cela signifie que l'ordinateur de votre adversaire a été mis sous tension après le vôtre. Votre ordinateur envoie par conséquent un "2" à l'ordinateur de votre adversaire et devient lui-même l'utilisateur 1. (Si d'autre part votre ordinateur reçoit en retour un "2", cela signifie que l'ordinateur de votre adversaire était déjà mis sous tension et à l'écoute lorsque votre ordinateur a envoyé le "1". Votre ordinateur deviendra donc de lui-même l'utilisateur "2".)

Si les deux ordinateurs démarrent en même temps, ils entreront en conflit et il sera alors nécessaire d'envoyer une interruption (**BREAK**) et de recommencer.

Le corps du programme est consacré à l'échange des noms, l'entrée du nombre secret à deviner (lequel n'est pas envoyé) et à la comparaison des hypothèses. Les hypothèses sont envoyées en premier, les réponses en second.

A partir de la ligne 190, les lignes détectent une victoire (une hypothèse correcte), la signalent de façon appropriée puis proposent une autre partie.

```

10 GO SUB 500
20 PRINT;:BORDER 1:PAPER 1:INK 7:CLS
30 PRINT " JEU DU NOMBRE A DEVINER""ENTREZ TOUT D'ABORD VOTRE NOMBRE.
  SECRET, PUIS DEVINER CELUI DE VOTRE ADVERSAIRE"
40 INPUT "COMMENT VOUS APPELEZ-VOUS?";a$
50 PRINT""HELLO";a$
60 GO SUB 1000+100*user
70 PRINT "A VOUS DE JOUER";b$
75 PRINT 'a$,b$
80 INPUT"PENSEZ A UN NOMBRE (ENTRE 1 ET 100)";a
90 IF a 1 OR a 100 OR a INT a THEN GO TO 80
130 INPUT"Faites une hypothèse";b
140 LET a$=STR$b: GO SUB 1000+100 user
150 LET c=ABS(a-VALb$)
160 IF c=0 THEN LET a$="Exact":GO TO 170
161 IF c<4 THEN LET a$="Très très proche":GO TO 170
162 IF c<10 THEN LET a$="Très proche":GO TO 170
163 IF c<20 THEN LET a$="Proche":GO TO 170
164 IF c<40 THEN LET a$="Assez proche":GO TO 170
165 IF c<60 THEN LET a$="Pas très proche":GO TO 170
166 LET a$="Vous en êtes loin"
170 GO SUB 1000+100*user
180 PRINT b$,a$
190 IF c=0 OR b$="Exact" THEN GO TO 210
200 GO TO 130
210 IF b$="Exact"THEN PRINT FLASH 1;"Victoire":FOR n=0 TO 7:
  BORDER n:BEEP.1,n:BEEP.1,n+16:NEXT nGO TO 230
220 PRINT,"Echec":FOR n=7 TO 0 STEP-1:BORDER n:
  BEEP.2,n:NEXT n
230 BORDER 1:INPUT"Une autre partie?(O/N)";a$
240 IF a$="O"THEN RUN 20
499 STOP
500 OPEN#4;"n";0
510 PRINT#4;"1"
520 CLOSE#4
530 OPEN#4;"n";0
540 INPUT #4;a$
545 CLOSE#4
550 LET a=4:IF a$="1"THEN OPEN#a;"n";0:PAUSE 5:PRINT#4;"2":LET user=2
560 IF a$="2" THEN LET user=2

```

```

570 CLOSE#4
580 FORMAT"n";user:RETURN
1100 OPEN#4;"n";3-user
1110 PRINT#4;a$
1120 CLOSE#4
1130 OPEN#4;"n";3-user
1140 INPUT#4;b$
1150 CLOSE#4
1160 RETURN
1200 OPEN#4;"n";3-user
1210 INPUT#4;b$
1220 CLOSE#4;"n";3-user
1240 PRINT#4;a$
1250 CLOSE#4
1260 RETURN

```

## LES VARIABLES SYSTEME

En plus des variables système données au chapitre 25 du Cours de Programmation BASIC, le Microdrive, le réseau local et le logiciel RS232 utilisent les variables système ci-dessous.

Notes	Adresse	Nom	Signification
X1	23734	FLAGS 3	Drapeaux
X2	23735	VECTOR	Adresse utilisée pour étendre l'interpréteur BASIC
X10	23737	SBRT	Sous-programme d'appel de mémoire ROM
2	23747	BAUD	Nombre de deux octets déterminant le débit en bauds calculé comme suit : $BAUD = (3500000 / (26 * \text{débit})) - 2$ Vous pouvez utiliser cette formule pour définir des débits non standard.
1	23749	NTSTAT	Numéro de station sur réseau
1	23750	IOBORD	Couleur de bordure utilisée pendant une E/S Vous pouvez choisir (POKE) la couleur que vous voulez
N2	23751	SER_FL	Zone de travail de 2 octets utilisée par RS232
N2	23753	SECTOR	Zone de travail de 2 octets utilisée par Microdrive
N2	23755	CHADD	Stockage temporaire pour CH ADD
1	23757	NTRESP	Stockage du code de réponse du réseau
1	23758	NTDEST	Le début du tampon du réseau contient les numéros 0-64 des stations de destination
1	23759	NTSRCE	Numéro de station source
X2	23760	NTNUMB	Numéros 0-65535 de bloc sur réseau
N1	23762	NTTYPE	Code du type d'en-tête
X1	23763	NTLEN	Longueur 0-255 du bloc de donnée
N1	23764	NTDCS	Total de contrôle du bloc de donnée
N1	23765	NTHCS	Total de contrôle du bloc d'en-tête
N2	23766	D_STR1	Début du spécificateur de fichier à 8 octets Numéro d'unité à 2 octets (1-8)
N1	23768	S_STR1	Numéro de voie (1-15)
N1	23769	L_STR1	Type de périphérique... "m", "n", "t" ou "b"
N2	23770	N_STR1	Longueur du nom de fichier
N2	23772		Début du nom de fichier
N8	23774	D_STR2	Deuxième spécificateur du fichier à 8 bits utilisé par les commandes MOVE et LOAD
N1	23782	HD_00	Début de la zone de travail pour SAVE, LOAD, VERIFY et MERGE (Sauvegarder, charger, vérifier et fusionner) le code de type de donnée.

Notes	Adresses	Nom	Spécifications
N2	23783	HD_0B	Longueur de données (0-65535)
N2	23785	HD_0D	Début des données (0-65535)
N2	23787	HD_0F	Longueur de programme (0-65535)
N2	23789	HD_11	Numéro de ligne
1	23791	COPIES	Nombre de copies effectuées avec SAVE
	23792		Début des MAP ou CHANS de Microdrive.

## AVERTISSEMENT

1. L'ouverture d'une voie vers le Microdrive ou vers le réseau occupe une certaine quantité de mémoire afin de créer un canal. Un canal Microdrive occupe 595 octets et un canal de réseau 276 octets. Ces canaux seront créés soit avec OPEN#, soit avec SAVE/LOAD/VERIFY/MERGE. En d'autres termes, un programme existant avec un espace mémoire insuffisant au-dessous de RAMTOP produira le message **Out of memory** (mémoire insuffisante) pour une quelconque de ces opérations.

2. La création de ces tampons a pour autre effet de transférer le code machine stocké dans une instruction REM, ce qui peut être à l'origine de problème. Il faut donc toujours placer les programmes en code machine au-dessus du RAMTOP.

3. Il est déconseillé d'envoyer une interruption (BREAK) pendant une opération d'écriture Microdrive (une pendant laquelle la bordure clignote) puisque vous risquez d'aboutir avec un fichier non fermé. L'instruction ERASE détruira les fichiers non fermés ; mais cette opération prendra environ trente secondes car l'ordinateur vérifie la cartouche plusieurs fois pour s'assurer que le fichier n'a pas de fermeture.

## LE CANAL MICRODRIVE

A chaque fois qu'un fichier est ouvert, une zone appelée CANAL est créée dans la zone désignée CHANS dans le Cours de Programmation en BASIC. Cette zone est habituellement adressée par le registre IX dans le logiciel. Le canal a une longueur de 595 octets et contient le tampon de 512 octets.

Le contenu du canal est le suivant :

0		Adresse 8
2		Adresse 8
4		'M'
5		Adresse du sous-programme de sortie en ROM
7		Adresse du sous-programme d'entrée en ROM
9		Adresse 595
11	CHBYTE	Compteur d'octet courant indiquant l'octet devant être ajouté à ou retiré de la zone de données dans la gamme 0-512 incluse
13	CHREC	Numéro d'enregistrement. Il indique la position de l'enregistrement dans la gamme de fichiers 0-255
14	CHNAME	Nom de fichier à 10 octets avec espaces de fin.
24	CHFLAG	Bit 0 d'octet drapeau forcé à 1...ouvert pour l'écriture Forcé à 0 ...ouvert pour la lecture Bits 1-7...inutilisés
25	CHDRIV	Numéro d'unité (0-7)
26	CHMAP	Adresse de la topographie (MAP) pour ce Microdrive
28	CHMAP	Préambule d'en-tête à 12 octets...marque le début de la zone de travail d'en-tête
40	HDFLAG	Bit 0 d'octet drapeau mis à 1 Bits 1-7...inutilisés
41	HDNUMB	Numéro de secteur dans la gamme 0-255
42		Inutilisé
44	HDNAME	Nom de cartouche et espaces de fin
54	HDCHK	Total de contrôle d'en-tête
55		Préambule de bloc de données de 12 octets...marque le début de la zone de travail des données
67	RECFLG	Bit 0 d'octet drapeau=0 Bit 2 =fichier non PRINT BITS 3-7 =inutilisés
68	RECNUM	Numéro de cet enregistrement dans la gamme 0-255

69	RECLEM	Nombre d'octets de données dans cet enregistrement (0-512)
71	RECNAM	Nom de fichier avec espaces de fin
81	DESCHK	Total de contrôle des 14 octets précédents
82	CHDATA	512 octets de données
594	DCHK	Total de contrôle des 512 octets précédents

## MAP

Pour chaque Microdrive contenant un fichier ouvert, il existe une zone appelée MAP créée dans la zone appelée "Microdrive maps" (topographies Microdrive) dans le Cours de Programmation BASIC. La zone MAP contient 32 octets. Chaque bit correspond à un secteur du Microdrive correspondant. Si ce secteur contient des données ou s'il est inutilisable, le bit est forcé à 1. Les bits sont numérotés comme suit : bit 0 octet 0 = secteur 0, bit 1 octet 0 = secteur 1, bit 0 octet 1 = secteur 8, etc.

## LE CANAL RESEAU

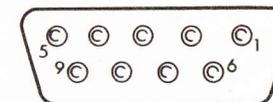
Quand une voie est ouverte vers le réseau, un canal est créé dans la zone désignée par CHANS dans le Cours de Programmation BASIC. Cette zone est habituellement adressée par le registre IX du logiciel. Le canal a une longueur de 276 octets et contient un tampon de 255 octets.

Le contenu du canal est décrit comme suit :

0		Adresse 8
2		Adresse 8
4		"N"
5		Adresse du sous-programme de sortie en ROM
7		Adresse du sous-programme d'entrée en ROM
9		Adresse 276
11	NCIRIS	Le numéro de station de destination
12	NCSELF	Le numéro de station de ce Spectrum
13	NCNUMB	Le numéro de bloc
15	NCATYPE	Le code de type de paquets...0 données, 1EOF
16	NCOBL	Nombre d'octets du bloc de données
17	NCDCS	Total de contrôle de données
18	NCHCS	Total de contrôle d'en-tête
19	NCCUR	Position du dernier caractère sorti du tampon
20	NCIBL	Nombre d'octets dans le tampon d'entrée
21	NCB	Un tampon de données a 255 octets

## LES CONNEXIONS RS232

Le connecteur RS232 est câblé comme suit :



1. Pas de connexion
2. Transfert de données (entrée)
3. Réception de données (sortie)
4. Signal DTR (entrée). Doit être haut quand le terminal est prêt
5. Signal CTS (sortie). RAZ avant transmission. Doit être haut quand le terminal est prêt
6. Sans connexion
7. Masse (aval)
8. Sans connexion
9. +9V (amont)

Le câble RS232 peut être obtenu auprès de Sinclair Research ; il associe le connecteur D à 9 broches à une prise D à 25 broches (les connecteurs D à 25 broches sont communs sur les périphériques RS232). Se reporter au catalogue des périphériques et du logiciel fourni avec l'interface ZX 1 où l'on trouvera de plus amples informations sur la manière de se procurer ce câble. Les conducteurs de ce câble sont affectés comme suit :

2. Transfert des données
3. Réception des données
5. Signal de RAZ avant transmission
6. +9V (normalement DSR : Data Set Ready)
7. Masse
20. DTR (Data Terminal Ready). Terminal prêt à envoyer.

## LES MESSAGES D'ERREURS

Maintenant que vous avez raccordé une interface ZX 1 à votre ordinateur, vos programmes peuvent donner lieu à des messages qui ne sont pas décrits dans l'annexe B du Cours de Programmation en BASIC. Ces messages seront suivis du numéro de ligne et du numéro d'instruction où le programme s'est arrêté.

Ces nouveaux messages sont présentés ci-dessous (en ordre alphabétique).

### Débit en bauds manquant (missing baud rate)

Le débit n'a pas été spécifié.

### Description invalide de périphérique (invalid device expression)

Un périphérique a été spécifié autrement qu'avec **s**, **p**, **k**, **m**, **n**, **t** ou **b**. Le même message peut être produit si un point-virgule a été utilisé au lieu d'une virgule avec l'un des canaux **s**, **p** ou **k**.

### Echec de vérification (verification has failed)

Un fichier sauvegardé ne concorde pas avec le programme, les données ou le code présents dans l'ordinateur.

### Ecriture vers un fichier en lecture (writing to a 'read' file)

Vous avez essayé d'envoyer des données vers un fichier existant. Le fichier existant doit être effacé s'il est devenu inutile. Sinon, un nouveau fichier doit être créé.

### Erreur de code (code error)

Vous avez essayé de charger (**LOAD**) un bloc de code plus grand que la zone de destination spécifiée par votre instruction **LOAD**.

### Erreur fusion (merge fusion)

Vous avez essayé de fusionner (**MERGE**) des données ou des codes. Vous ne pouvez fusionner que des programmes. Ce message apparaît également si vous essayez de fusionner un programme sauvegardé avec l'instruction **SAVE... LINE....**

### Fichier non trouvé (file not found)

Ceci signifie soit que vous avez essayé de charger (**LOAD**) à partir d'un fichier qui n'existe pas soit qu'une partie du fichier ne peut être trouvée (ceci peut être dû soit à la non fermeture du fichier soit à l'endommagement du fichier au moment de la mise en ou hors tension alors que la cartouche se trouvait dans un Microdrive).

**Lecture d'un fichier en écriture** (reading a 'write' file)

Vous avez essayé d'entrer des données à partir d'un fichier inexistant ou ouvert pour recevoir des données.

**Microdrive absent** (Microdrive not present)

Vous avez essayé d'utiliser soit un Microdrive non connecté à votre ordinateur soit un Microdrive ne contenant pas de cartouche soit un Microdrive contenant une cartouche non formatée.

**Microdrive pleine** (Microdrive full)

Vous avez essayé d'écrire des données sur une cartouche ne contenant plus d'espace disponible. Vous devez donc réexécuter votre programme ou votre commande avec une cartouche non pleine. Pour ce faire, vous pouvez effacer des fichiers anciens de la cartouche existante. Un fichier ouvert pour l'écriture sur la cartouche pleine ne peut pas être fermé. Il doit être effacé. Toutefois, cette opération peut prendre environ trente secondes du fait que l'ordinateur vérifie à plusieurs reprises la cartouche pour s'assurer que le fichier n'est pas fermé.

**Nom invalide** (invalid name)

Un nom de fichier a été spécifié comme une chaîne vide ou une chaîne possédant plus de 10 caractères.

**Nom manquant** (missing name)

Le nom de fichier n'a pas été spécifié.

**Numéro invalide d'unité** (invalid drive number)

Un numéro de Microdrive a été spécifié hors de la gamme 1 à 8

**Numéro de station invalide** (invalid station number)

Une station du réseau a été spécifiée hors de la gamme 0 à 64 (ou hors de la gamme 1 à 64 pour des instructions FORMAT).

**Numéro de station manquant** (missing station number)

Le numéro de station du réseau n'a pas été spécifié.

**Numéro d'unité manquant** (missing drive number)

Le numéro de Microdrive n'a pas été spécifié.

**Numéro de voie invalide** (invalid stream number)

Une voie a été spécifiée hors de la gamme de 0 à 15.

**Programme terminé** (program finished)

Vous avez essayé d'exécuter une ligne au-delà de la dernière existante. Ce message apparaîtra également si une instruction GO TO est suivie d'un numéro supérieur à celui de la dernière ligne. Il apparaîtra encore si l'instruction RUN est tapée en l'absence de programme.

**Type incorrect de fichier** (wrong file type)

Vous avez essayé soit d'entrer (INPUT) ou de transférer (MOVE) un fichier sauvegardé soit de charger (LOAD), vérifier (VERIFY) ou fusionner (MERGE) un fichier du type PRINT soit de charger (LOAD) un fichier de codes ou de données comme un programme (ou inversement).

Si vous utilisez l'instruction INPUT, vous devez aussi utiliser l'instruction LOAD. Si vous utilisez LOAD, vous devez utiliser soit les options CODE ou DATA soit utiliser INPUT.

**Unité protégée en écriture** (drive 'write' protected)

Vous avez essayé d'écrire des données sur le Microdrive contenant une cartouche protégée (c'est-à-dire que la languette plastique a été cassée).

**Voie déjà ouverte** (stream already open)

Vous avez essayé d'ouvrir (OPEN) une voie qui a déjà été ouverte vers un nouveau canal (m, n, t ou b). La voie ne peut être ouverte que si elle a d'abord été fermée.

## LE BASIC ETENDU

L'interface ZX 1 étend les capacités BASIC existantes du Spectrum. Les extensions et les suppléments sont résumés ci-dessous.

### Voies

Les voies sont spécifiées par # n où n est un nombre compris entre 1 et 15. Les voies 1, 2 et 3 sont habituellement utilisées par BASIC. Le caractère # est une partie du mot-clé pour les instructions **OPEN #** et **CLOSE #**.

### Canaux

Il existe sept types différents de canal dans le BASIC étendu ; le clavier (k), l'écran (s), l'imprimante (p), l'interface de texte RS232 (t), l'interface binaire RS232 (b), le réseau (n) et le Microdrive (m).

Chaque type de canal est spécifié par sa lettre qui peut être en majuscule ou en minuscule. Le réseau et le Microdrive nécessitent des informations supplémentaires pour spécifier totalement le canal.

Un canal de réseau exige un numéro de station de sorte qu'un canal de réseau est spécifié par "n";x où x est un numéro de station compris entre 0 et 64.

Un canal Microdrive nécessite un numéro de Microdrive et un nom de fichier ; ainsi, un canal Microdrive est spécifié par "m";y;"nom" où y est le numéro de Microdrive compris entre 1 et 8 "nom" est une chaîne de 1 à 10 caractères.

### Instructions

#### CAT y

Donne une liste de tous les fichiers de la cartouche dans le Microdrive y. La liste est présentée en ordre alphabétique et est précédée du nom de la cartouche et suivie de la capacité en kilo-octets restant disponible.

#### CAT #z;y

Envoie le catalogue de la cartouche dans le Microdrive y comme décrit ci-dessus vers la voie z.

#### CLOSE#voie

Détache tout canal pouvant être lié à la voie spécifiée. Si des données sont conservées en tampon, celles-ci sont soit transmises (sur le réseau) soit enregistrées (sur le Microdrive).

**Instructions**

<b>ERASE "m";y;"nom"</b>	Efface de la cartouche dans le Microdrive y le fichier ayant le nom spécifié.
<b>FORMAT "m";y;"nom"</b>	Prépare une cartouche vierge dans le Microdrive y pour son utilisation avec le BASIC. Le "nom" est donné à la cartouche et apparaîtra dans les catalogues.
<b>FORMAT "n";x</b>	Donne la valeur x au numéro de station du réseau.
<b>FORMAT "t";x</b> <b>FORMAT "b";x</b>	Donne la valeur x au débit en bauds pour l'interface RS232 (x doit être choisi parmi les valeurs 50, 110, 300, 600, 1200, 2400, 4800, 9600, 19200).
<b>INKEY\$ # voie</b>	Renvoie un caractère unique sous forme d'une chaîne si un caractère est disponible et renvoie la chaîne nulle " " si aucun caractère n'est disponible sur la voie. Cette instruction ne prend son sens que si la voie est liée au réseau ou à l'interface RS232.
<b>INPUT # voie;variable</b>	Entre la variable à partir de la voie spécifiée. La voie doit avoir été préalablement ouverte vers un canal d'entrée. Il convient de noter qu'aucun élément d'impression dans l'instruction <b>INPUT</b> ne sera sorti sur la voie. Ceci n'est habituellement requis qu'à l'occasion d'un entrée à partir du clavier. Il faut noter aussi que le séparateur "," sort un caractère. L'option <b>LINE</b> reste disponible comme auparavant.
<b>LOAD*options de canal</b>	Charge le programme, les données ou les codes à partir du canal spécifié. Seuls les canaux "b", "n", ou "m" peuvent être utilisés. Toutes les options disponibles avec <b>LOAD</b> , sont disponibles avec <b>LOAD*</b> .
<b>MERGE*options</b>	Même commentaire que pour <b>LOAD</b> ci-dessus à l'exception que l'instruction ne supprime pas de ligne ou de variable de l'ancien programme sauf pour faire place pour de nouvelles en utilisant le même numéro de ligne ou le même nom.

**Instructions**

<b>MOVE source TO destination</b>	Transfère des données de la source vers la destination. La source et la destination peuvent être soit des numéros de voie ou des canaux. La commande n'est terminée que lorsqu'un marqueur de fichier est rencontré dans la source : ceci ne peut arriver que si la source est soit un canal de réseau ou de Microdrive soit une voie liée à un canal. Si la source ou la destination est un canal, le canal est tout d'abord ouvert puis fermé.
<b>OPEN # voie, canal</b>	Lie le canal spécifié à la voie spécifiée pour permettre au BASIC d'envoyer une entrée ou une sortie vers ce canal. La voie doit être préalablement fermée ou ouverte vers k, s ou p.
<b>PRINT # voie...</b>	Sort la séquence d'impression (...) vers la voie spécifiée. La voie doit avoir été préalablement ouverte vers un canal de sortie. La séquence d'impression a la même syntaxe que précédemment et doit contenir des # supplémentaires.
<b>SAVE * options canal</b>	Sauvegarde le programme, les données ou les codes vers le canal spécifié. Seuls les canaux "b", "n" ou "m" peuvent être utilisés. Toutes les options disponibles avec <b>SAVE</b> sont également disponibles avec <b>SAVE*</b> .
<b>VERIFY*options canal</b>	Comme pour <b>LOAD</b> ci-dessus à la différence que les données ne sont pas chargées mais comparées avec les données existantes.

## Index

Le présent index contient les touches correspondant au clavier et comment les obtenir

apostrophe (')		25
ASCII		36
binaire, voir canal b		
bit d'arrêt, numéro sorti par spectrum		35
bits, nombre sorti par Spectrum		35
BREAK	CAPS SHIFT et SPACE	18, 39, 46
canal b		37, 38, 55
canal k		22, 24, 39, 55
canal m		16ff, 55
canal n		30ff, 55
canal p		22, 24, 39, 55
canal s		22, 24, 39, 55
canal t		36ff, 40, 55
canaux		21ff, 36ff, 39, 46, 48, 55
caractères graphiques		36
caractères spéciaux		36
CATalogue	E, SYMBOL SHIFT 9.	15, 19, 27, 28, 55
CLOSE #	E, SYMBOL SHIFT 5.	23, 24, 32, 39, 55
codes de contrôle		36, 37, 38
connecteurs, sur l'interface Zx		7
couleurs, changement de		26
diffusion		33
débit en bauds		36
ERASE	E, SYMBOL SHIFT 7.	18, 28, 46, 52, 56
exécution automatique		15, 20
FORMAT	E, SYMBOL SHIFT 0.	19, 30, 36, 37, 56
guillemets (")		25
identification des cartouches vierges		19
identification des fichiers de données		23, 27
identification des programmes		17
INK	E, X	26
INKEY\$	E, sur N	24, 32, 33, 36, 37, 56
INPUT	K, sur l	25, 26, 31, 36, 37, 56
jeu de caractères		36
listes, listings		36
LLIST	E, sur V	37
LOAD	K, sur J	16, 17, 18, 30, 31, 33, 37, 38, 39, 46, 51, 56
LPRINT	E, sur C	22, 37
MERGE	E, T	18, 20, 31, 46, 56
MOVE	E, SYMBOL SHIFT 6.	39, 40, 46, 56
numéros de station		30
OPEN#	E, SYMBOL SHIFT 4.	22, 23, 31, 37, 39, 46, 57
PAPER	E, C	26
parité ON/OFF		35
point-virgule (;)		22, 25
POKE	K, sur O	45
PRINT	K, sur P	22, 25, 26, 57
RAMTOP		46

saut de ligne automatique		
SAVE	K, sur S	35
SAVE*...LINE...		17, 30, 31, 33, 37, 38,
scroll? défilement		20
spécificateurs de canal		39
séparateurs		22, 39
tampons		22, 25
VERIFY	E, R	23, 32
virgule (,)		17, 31, 46, 57
voies		22, 26
		21ff, 26, 39, 46, 48, 55

' voir apostrophe  
, voir virgule  
" voir guillemets  
; voir point-virgule

