

---

# **RS232C**

## **INTERFACE SERIE**

### **GUIDE DE L'UTILISATEUR**

---

---

# **RS232C AMSTRAD**

**INTERFACE SERIE AVEC  
LOGICIEL ROM ET  
ALIMENTATION ELECTRIQUE**

**AMSOFT**

Un département de

**AMSTRAD**

© Copyright 1985 AMSOFT, AMSTRAD SARL, AMSTRAD Consumer Electronics plc

Ni l'information contenue aux présentes, ni le produit décrit dans ce manuel, ne peuvent être modifiés ou reproduits totalement ou partiellement, en tout ou partie, sous quelque forme que ce soit, sans l'accord préalable d'Amstrad SARL.

---

AMSOFT et AMSTRAD accepteront volontiers vos suggestions à propos du produit et de ce guide

Toute correspondance doit être adressée à

AMSTRAD FRANCE  
72-78 Grande rue  
92310 Sèvres

Toute maintenance et service après vente concernant le produit doivent être effectués obligatoirement par les revendeurs Amsoft agréés. Ni Amsoft ni Amstrad ne seront responsables, de quelque façon que ce soit, de toute perte ou dommage causé par une maintenance ou service effectué par des personnes non-agrèés.

Ce guide est seulement destiné à faciliter l'utilisation du produit par le lecteur et, par conséquent, ni Amsoft ni Amstrad ne seront responsables de toute perte ou dommage quelconque qui pourrait résulter de l'utilisation de toutes informations, renseignements, erreurs ou omissions contenus dans ce guide, ainsi que toute utilisation impropre du produit.

CP/M est une marque déposée de Digital Research Inc.  
Z80 est une marque déposée de Zilog Inc.  
AMSDOS, CPC464, CPC664, CPC6128, DDI-1, et FD-1 sont des marques déposées d'AMSTRAD  
Première publication 1985

Ecrit par Roland Perry  
Traduit par Laurence Ollivry  
Illustré par Alexander Martin

Electronique et SIO drivers de MEJ Electronics

Programmation d'AMSOFT:

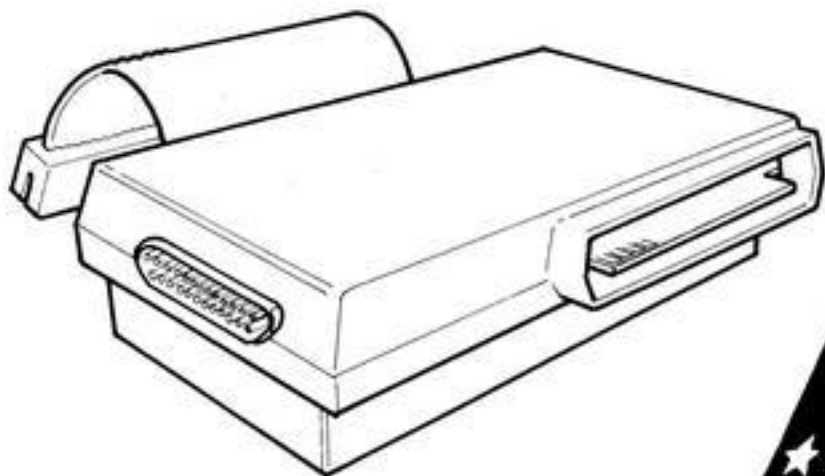
Imprimante, PRESTEL, et TERMINAL par David Radisic  
Transferts de dossiers par Vik Olliver

Concepts de Roland Perry  
Liaison et service ROM par Cliff Lawson  
Compilé par Ivor Spital

Publié par AMSTRAD  
Composé par KAMSET typesetting graphics (Brentwood)

AMSTRAD est une marque déposée de AMSTRAD Consumers Electronics plc. L'emploi de la marque ou du nom AMSTRAD sans autorisation préalable est strictement interdit.

**Bienvenu dans le monde  
de la magie de la RS232C  
AMSTRAD**



L'installation de la RS232C AMSTRAD sur votre ordinateur, vous permettra de communiquer avec les imprimantes, les modems et les autres ordinateurs. La réalisation d'un tel lien est souvent regardée comme étant compliquée et obscure, particulièrement à cause de la flexibilité et la versatilité qu'offre l'interface série. Pour faciliter l'utilisation de votre interface série RS232C AMSTRAD, nous avons produit des instructions sous forme de 'tours de magie'. Il vous suffira de sélectionner les tours, expliqués par des dessins, se rapportant à votre application.

Toutes les explications techniques vous sont fournies dans les appendices. Les exemples de 'tours' ne démontrent pas l'entière possibilité de la RS232C, mais représentent, cependant, la majorité des applications rencontrées.

L'interface série RS232C AMSTRAD et ces instructions, peuvent être utilisées, avec la même efficacité, sur les ordinateurs CPC464, CPC664 et CPC6128 d'AMSTRAD. Certaines applications mentionnées marchent mieux avec un système à disquette, la disquette CP/M 2.2 devra être utilisée (elle est fournie avec le CPC664, et le DDI-1 du CPC464) ou la CP/M Plus (fournie avec le CPC6128). Vous pourrez aussi avoir recours à la cassette, comme alternative.

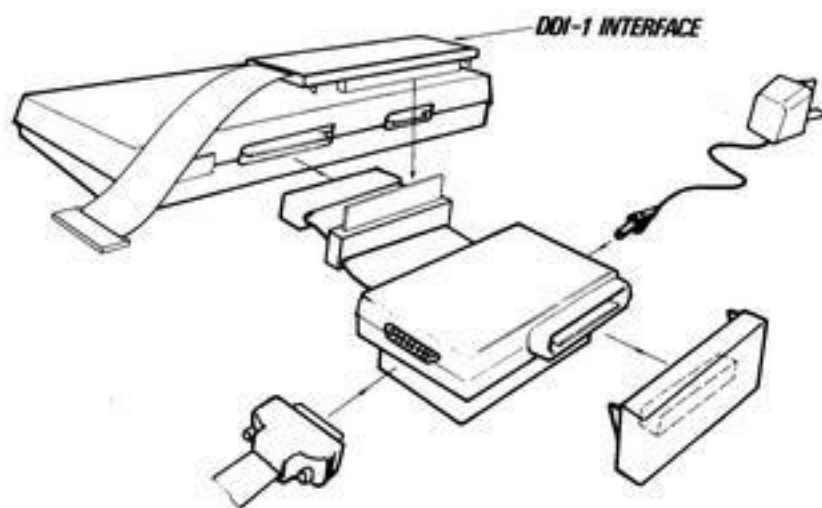
Si vous utilisez le BASIC, les commandes supplémentaires sont programmées dans un ROM (mémoire morte) se trouvant à l'intérieur de la RS232C et sont accédées par des mots clés commençant par une barre |, qui est obtenue en appuyant sur [SHIFT]a.

Vous trouverez les commandes CP/M mentionnées sur votre disquette système CP/M.



L'interface série vous permet de connecter le matériel à l'aide de cables simples, d'à peu près 15 mètres de long. A l'aide d'un modem, on peut convertir les signaux de ces cables de manière à les envoyer, à n'importe quelle distance, vers un autre modem, par des lignes téléphoniques ordinaires. La plupart des fabricants se sont mis d'accord sur les détails de liaison d'un certain standard, appelé RS232C. Il est aussi possible de brancher des dispositifs se conformant à la 'RS423'.

## Tour No 1 Brancher la RS232C sur votre ordinateur



Eteignez l'ordinateur et débranchez tout ce qui est marqué 'EXPANSION' ou 'FLOPPY DISC'.



Enfoncez la prise se trouvant au bout du câble plat de la RS232C dans le connecteur du port de l'ordinateur marqué 'EXPANSION' ou 'FLOPPY DISC'.

Si nécessaire, rebranchez les autres interfaces dans le bus d'extension de la RS232C.

Branchez le fil électrique de la RS232C dans la petite prise se trouvant sur le côté de la RS232C.

**ATTENTION - RISQUE DE DECHARGE ELECTRIQUE**

**NE JAMAIS ESSAYER D'UTILISER LE BOITIER D'ALIMENTATION DE LA RS232C, S'IL EST OUVERT.**

Allumez toutes les autres interfaces d'extension, branchez la prise d'alimentation de la RS232C dans la prise de courant, et allumez votre ordinateur, suivez cet ordre. Un message 'réveil', de ce genre, apparaîtra sur votre écran:-

**Amstrad 64K Microcomputer (v2)**

©1984 Amstrad Consumer Electronics plc  
and Locomotive Software Ltd.

**Amstrad RS232C Serial Interface (v1)**

©1985 Amstrad Consumer Electronics plc

Ready

Il est important d'éteindre dans le sens inverse de l'allumage, avant de débrancher la RS232C (et les interfaces d'extension) de l'ordinateur.

(Toujours débrancher la RS232C quand elle n'est pas utilisée)

Le câble de sortie séquentielle (le connecteur à 25 plots), peut être connecté ou déconnecté à tous moments, même si la RS232C est branchée.

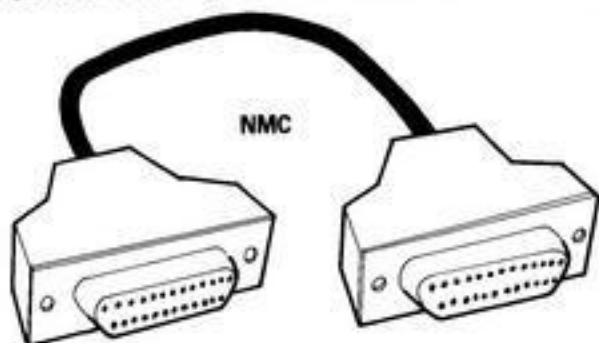




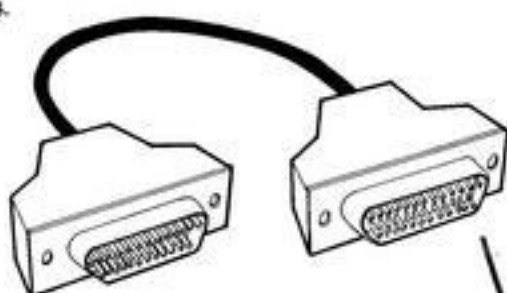
## Tour No 2

### Câbles pour connecter une imprimante (ou table traçante)

Le câble qui est utilisé pour le branchement d'une imprimante (ou tout autre dispositif de sortie) est appelé un câble modem-nul (NMC). Vous trouverez des explications techniques et un schéma du branchement dans l'appendice 1.



Un câble modem-nul a une prise à chaque extrémité pour brancher les connecteurs de la RS232C et l'imprimante. Si votre imprimante a une prise femelle, vous aurez besoin d'un câble convertisseur à deux connecteurs mâles.



## **Tour No 3**

### **Régler la vitesse de votre imprimante**

Il est possible de modifier la vitesse de transmission des caractères venant de la RS232C, sur l'imprimante. Cette vitesse est mesurée en bauds et est à peu près dix fois le nombre de caractères par seconde.

Vous devrez vous assurer que la RS232C transmet à la même vitesse que l'imprimante reçoit. Ne confondez pas la vitesse de cette interface-série avec la vitesse à laquelle l'imprimante imprime les caractères sur papier. Si l'imprimante ne peut pas suivre la vitesse de l'arrivée des caractères de l'interface-série, elle renverra des signaux spéciaux à la RS232C pour arrêter l'envoi et pouvoir combler le retard. Ce procédé s'appelle le 'contrôle de déroulement'.

Votre imprimante devrait avoir des boutons (se trouvant parfois à l'intérieur du boîtier) servant à régler la vitesse de transmission. La bonne vitesse à choisir est 9600 bauds (à peu près 960 caractères par seconde) qui est la vitesse par défaut de la RS232C.



## Tour No 4

### Régler la vitesse de la RS232C

La vitesse normale de la RS232C est 9600 Bauds.

**BASIC AMSTRAD:** Utilisez la commande  
`!SETSIO , -vitesse de transmission-`

**CP/M 2.2:** La vitesse de transmission est réglée durant le chargement de CP/M. Se reporter à l'Appendice 2 (`SETUP.COM`) pour voir comment modifier cette configuration de mise sous tension.

**CP/M Plus:** La vitesse de transmission peut être changée par la commande:

`SETSIO , -vitesse de transmission-`

ou

`DEVICE SIO (-vitesse de transmission-)`

Exemples:-

**BASIC AMSTRAD:** `!SETSIO ,1200` pour régler sur 1200 bauds

**CP/M Plus:** `SETSIO 1200` ou `DEVICE SIO(1200)`

Si vous devez changer la vitesse de transmission, vous devrez la régler à chaque remise à zéro de l'ordinateur ou chaque fois que vous passez du BASIC au CP/M, ou inversement. Avec CP/M Plus, il est possible de programmer la vitesse de transmission désirée comme faisant parti du procédé de chargement, en incorporant la commande donnée ci-dessus, dans le dossier 'PROFILE.SUB'.

**NOTEZ:** Si pendant l'utilisation de CP/M Plus, vous changez la vitesse de transmission à l'aide de `SETSIO`, la commande `DEVICE` ne montrera pas le changement.



## **Tour No 5** **Régler les bits de synchronisation** **(imprimante)**

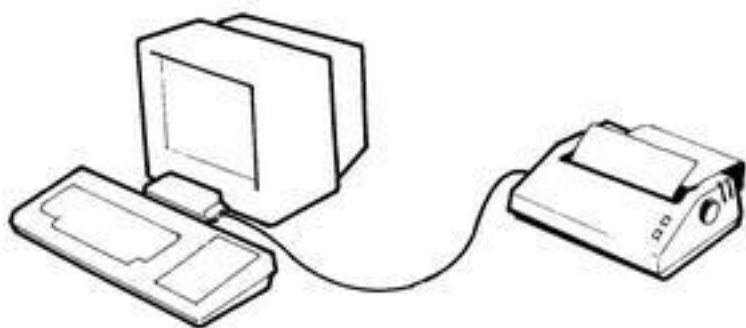
Les caractères envoyés par la RS232C ont d'autres propriétés que leur vitesse, notamment, le nombre de bits 'données', le nombre de bits 'arrêt', et le type de 'parité'. Il n'est pas important de comprendre la nature exacte de ces propriétés, mais de même qu'avec la vitesse de transmission, la RS232C et l'imprimante devront s'accorder.

La plupart des imprimantes ont des boutons de réglage, pour ces propriétés, il n'est pas essentiel que la RS232C et l'imprimante soient exactement pareilles; le test final est d'essayer une certaine configuration et de constater que tout marche comme prévu.

Le réglage par défaut de la RS232C est de 8 bits 'données', 1 bit 'arrêt', et aucune parité. Voir Appendice 2 pour CP/M, ou Appendice 3 pour BASIC qui vous donnent une description complète des commandes utilisées pour modifier les bits de synchronisation de la RS232C.



## Tour No 6 Renvoyer les sorties de l'imprimante par la RS232C



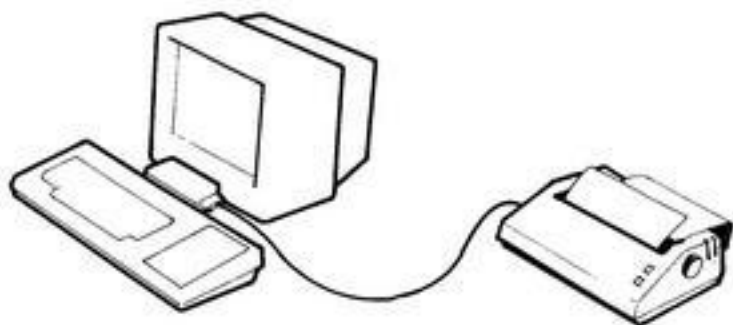
Normalement, les sorties de l'imprimante sont envoyées au port Centronics Parallèle, se trouvant derrière votre ordinateur. Mais si la RS232C est branchée et mise en marche, les commandes suivantes enverront toutes les sorties de l'imprimante par l'interface série:-

BASIC AMSTRAD:	ISERIAL
CP/M 2.2	STAT LST:=TTY:
CP/M Plus:	DEVICE LST:=SIO

Avec CP/M, il est possible de programmer ce changement de direction comme faisant parti du procédé de chargement. Pour CP/M Plus vous devrez incorporer la commande donnée ci-dessus, dans le dossier 'PROFILE.SUB'. Pour CP/M 2.2, utilisez SETUP.COM, décrit dans l'Appendice 2.



## Tour No 7 Restituer les sorties de l'imprimante au port parallèle



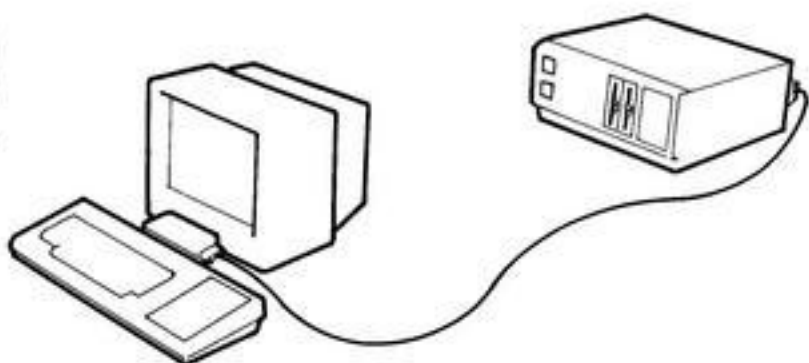
A chaque remise à zéro de l'ordinateur, ou chaque fois que vous passez du BASIC au CP/M ou inversement, les sorties seront rendues au port parallèle. Ceci peut aussi être achevé à l'aide des commandes:-

BASIC AMSTRAD:	IPARALLEL
CP/M 2.2:	STAT LST:=LPT:
CP/M Plus:	DEVICE LST:=LPT



## Tour No 8

### Câbles pour brancher l'amstrad comme terminal d'un mini-ordinateur

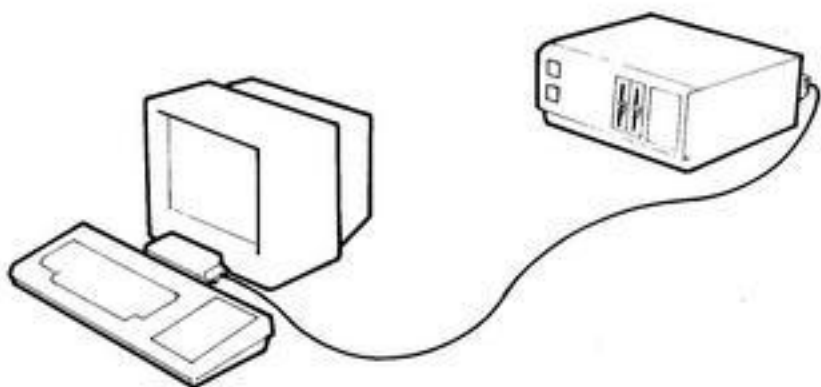


Dans ce manuel, un micro-ordinateur commercial, c'est à dire comportant un terminal, est parfois appelé un mini-ordinateur.

Il est possible de connecter votre ordinateur et sa RS232C, qui feront guise de terminal, directement à un mini-ordinateur. Le câble à utiliser dépendra du type du mini-ordinateur. Ce sera soit un câble modem-nul si le mini-ordinateur a une prise mâle, ou un câble modem s'il a une prise femelle. En cas de doute, reportez vous à l'Appendice 1 du guide du matériel du mini-ordinateur.



## Tour No 9 Attacher un émulateur de terminal à un mini-ordinateur



La vitesse de transmission et les paramètres de synchronisation de la RS232C et du mini-ordinateur devraient correspondre, comme indiqué auparavant pour les imprimantes.

Pour introduire l'émulateur de terminal, tapez:

**BASIC AMSTRAD:    I**TERMINAL

et pour en sortir, pressez sur **[CTRL][ESC]**





Notez que l'émulateur de terminal hérite des valeurs du clavier ayant été laissées par BASIC, et répond aux codes de fonction de l'écran décrits dans le guide de l'utilisateur de l'ordinateur AMSTRAD. Il est souvent utile de régler la touche [ESC] pour qu'elle renvoie toujours la même valeur, en entrant:

KEY DEF 66,B,27



## Tour No 10

### L'écho et les codes de fonction de l'émulateur de terminal

L'émulateur de terminal habituel envoie toutes les frappes au mini-ordinateur, affiche tous les caractères reçus, à imprimer, et obéit à tous les codes de fonction.

Normalement, le mini-ordinateur affiche simultanément, sur l'écran, les frappes qui lui sont envoyées, comme un écho. S'il ne peut pas faire fonctionner cet écho, on indiquera à l'émulateur de terminal de le faire à sa place, pour exécuter, tapez:

`IHALFDUPLEX`

Cet écho local peut être interrompu, en tapant:

`IFULLDUPLEX`

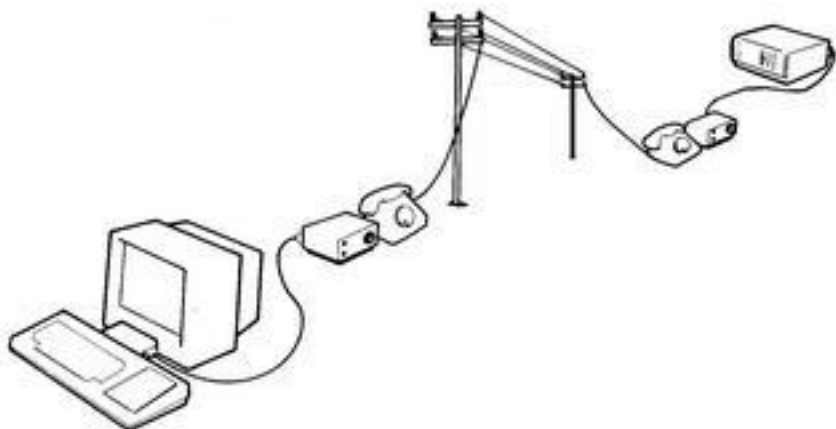
Il est parfois préférable que le terminal affiche (display) les codes de fonction plutôt que de les obéir, pour les diagnostics par exemple. L'émulateur de terminal peut être passé d'un mode à l'autre, à l'aide de ces commandes:

`ICTRLDISPLAY ...et... ICTRLACTION`



## Tour No 11

### Connecter à un autre ordinateur par un modem



Un modem est tout simplement une façon d'allonger la liaison série, entre deux ordinateurs, normalement par le réseau téléphonique. Le modem est connecté par un câble-modem (câble-prise femelle à câble-prise mâle) si la prise femelle est la bonne. Sinon, il faudra utiliser un câble spécial, voir Appendice 1 pour explications et exemples.



La vitesse de transmission et les paramètres de synchronisation de la RS232C doivent correspondre au modem et à l'ordinateur se trouvant à l'autre bout de la ligne.

Les vitesses habituelles des modems sont: 300 bauds (réception et transmission) ou 1200 bauds (réception)/75 bauds (transmission).

Les commandes à utiliser pour cela, sont:-

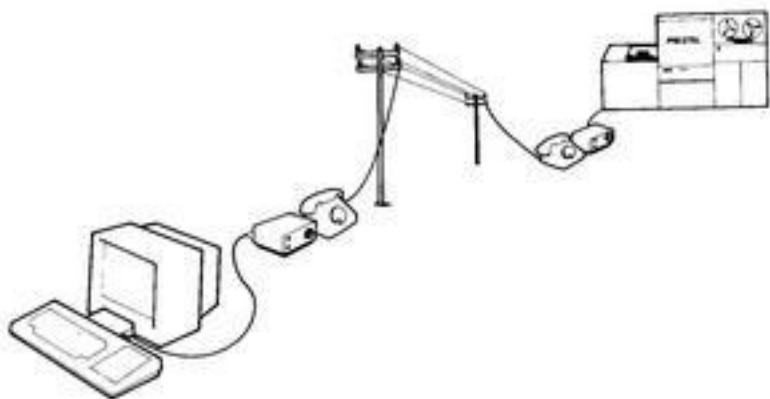
`ISETS10,300` qui actionne transmission et réception à 300 bauds

`ISETS10,75,1200,1,7,1,0` pour PRESTEL



## Tour No 12 Attacher un émulateur de terminal ou un émulateur PRESTEL

(PRESTEL est un service valable uniquement en Angleterre)



Après avoir réglé la vitesse de transmission correspondante, votre ordinateur + la RS232C peuvent être attachés au serveur, et être utilisés comme un terminal, à l'aide de la commande `!TERMINAL`. Reportez vous au branchement du mini-ordinateur pour de plus amples détails.



Un émulateur PRESTEL est aussi disponible, et est introduit par:

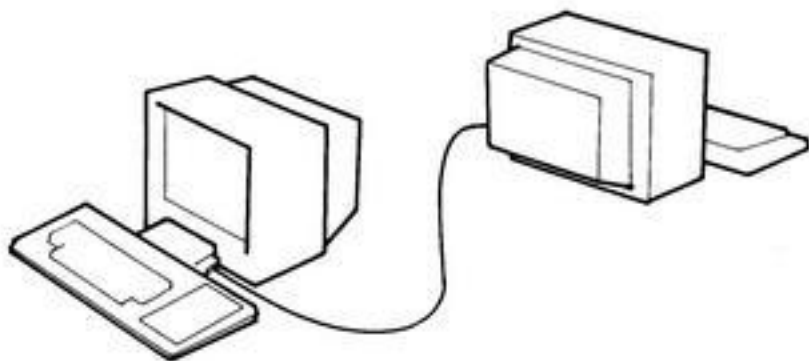
`I PRESTEL` et pour en sortir ... `[CTRL][ESC]`.

L'écran est exploité en **MODE 0** avec 16 couleurs, dont 8 intermittentes. Le nombre de pixels disponibles dans ce mode limite la résolution des caractères sur l'écran.

L'émulateur **PRESTEL** reçoit des caractères simultanément, venant du modem et fait une mise à jour, par étapes, de l'écran.



## Tour No 13 Connecter deux ordinateurs pour les transferts de dossiers



Il est utile de pouvoir transférer des dossiers d'un ordinateur à l'autre par une liaison série, surtout s'ils n'utilisent pas le même format, disquette ou cassette.



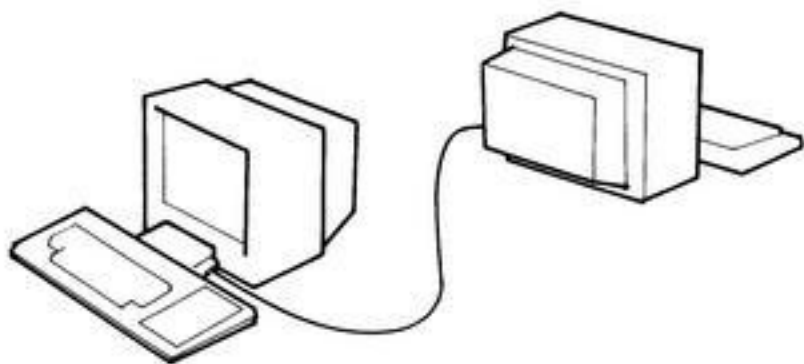
On assume que vous désirez recevoir des dossiers sur votre ordinateur AMSTRAD, venant d'un autre ordinateur. Si vous voulez programmer un autre ordinateur pour qu'il puisse recevoir des dossiers, venant d'un ordinateur AMSTRAD, vous trouverez le protocole en Appendice 3. Un programme de transmission sera aussi nécessaire, si votre ordinateur ne fait pas parti des exemples qui suivent.

Suivez les recommandations pour connecter votre ordinateur à l'ordinateur de transmission ou au modem. Vous trouverez des types de câbles de connexion parmi les exemples suivants. Assurez vous que la vitesse de transmission et les paramètres de synchronisation correspondent. Vous devrez utiliser les bits de 'données' 8 par défaut, il est aussi utile de régler la parité paire ou impaire (plutôt qu'aucune) afin d'obtenir l'intégrité des données.





## Tour No 14 Réception du dossier sur votre ordinateur AMSTRAD



Après avoir sélectionné la cassette ou la disquette comme destination du dossier, tapez:

`!SUCK, nom du dossier ...d'une des façons suivantes:`

Pour toutes les versions de BASIC (CPC464, CPC664 et CPC6128):

`a$="FILE.TYP": !SUCK,aa$`

ou

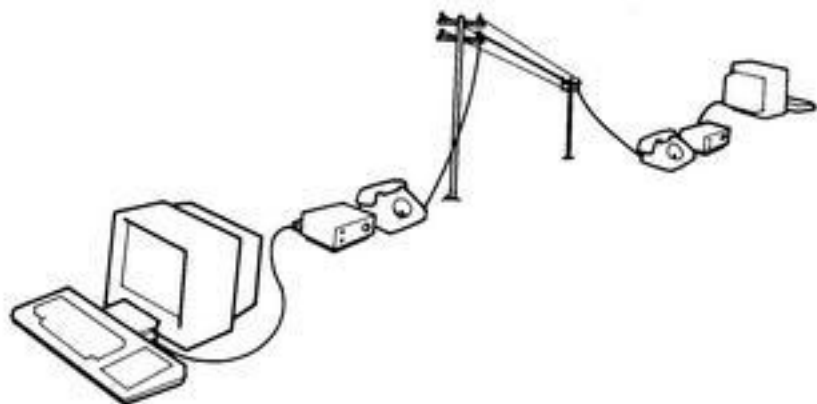
`!SUCK,"FILE.TYP" (pour CPC664 et CPC6128 seulement)`



Le dossier sera mis en mémoire sur disquette ou cassette sous forme de dossier ASCII, et pourra être lu par un programme BASIC ou CPM. La section de votre Guide de l'Utilisateur (CPC664, CPC6128 ou DDI-1) décrivant les transferts de disquette à cassette, contient toutes les informations sur ce dossier ASCII.



## Tour No 15 Transmission du dossier de votre ordinateur AMSTRAD



Ceci est surtout utile quand deux ordinateurs AMSTRAD distants l'un de l'autre, sont connectés par modem, rendant un simple transfert physique de cassette ou disquette, impossible. Le programme de réception vérifie l'arrivée des données pour empêcher les erreurs, et demande la retransmission, si nécessaire.



Après avoir sélectionner la cassette ou disquette comme source du dossier, tapez:-

`IBLOW, <nom du dossier> ... de la façon suivante:`

Pour toutes les versions de BASIC (CPC464, CPC664 et CPC6128):

```
a$="FILE.TYP": IBLOW,@a$
```

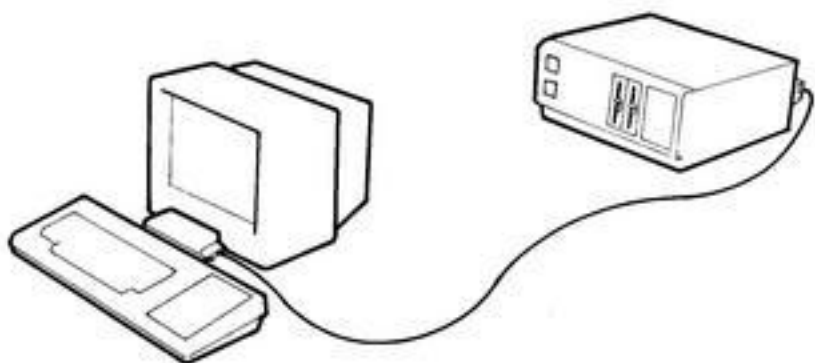
ou

```
IBLOW, "FILE. TYP" (pour CPC664 et CPC6128 seulement)
```

Encore une fois, le dossier doit être en ASCII (n'oubliez pas que AMSDOS considère tous les dossiers CP/M comme des ASCII), et il y a ici aussi, des similitudes avec la procédure de transfert de cassette à disquette.



**Tour No 16:  
Transmission du dossier d'un ordinateur CP/M**



Entrez le vidage HEX suivant, dans votre ordinateur CP/M à l'aide d'un éditeur de texte ou de PIP, par exemple:-

PIP BLOW.HEX=CON:

Maintenant, entrez le vidage HEX suivant (ligne par ligne), et pressez [ENTER] à la fin de chaque ligne, suivi de [CTRL]J. A la dernière ligne, pressez en plus [CTRL]Z, pour terminer l'opération:

```
:180100003A5D00FE20CA0502115C000E0FC00500FEFFCAB0E02C026013A
:180118000E10CD0500118E020E09CD0500C721FFFF22A5021E020E0474
:18013000CD0500C08101022C012AA5022322A502C09C01C0B001C0D055
:1801480001B7C26A01C0B001C0C001C0C801C0D101380DEC09C01C0B0C4
:1801600001C34001C09C01C0B0011E000E04C0050021000022A702C0D2
:18017800C801C08101026401C90E03C00500FE03C29601F1F1116A020B
:180190000E09C00500C9FE0637C83FC9215C007EF6405F0E04E5C00541
:1801A80000E123060FC3F50121A5020602C3F5011E800E04C00500C999
:1801C0002180000680C3F50121A7020602C3F50121000022A7020E14AE
:1801D800115C00CD05002180000680E516005E2AA7021922A702E12395
:1801F00005C2E301C9E5C55E0E04C00500C1E12305C2F501C911170222
:180208000E09C00500C71142020E09C00500C70A074E6F2066696C659B
:18022000207370656369666965642E0D0A0A5472616E736665722061E5
:18023800626F727465640D0A0A240A0746696C65206E6F74206666F757D
:180250006E642E0D0A0A5472616E736665722061626F727465640D0A18
:180268000A240A075472616E736665722041626F7274656420627920FE
:180280006F7468657220656E642E0D0A0A240A5472616E7366657220B
:11029800636F6D706C6574652E0D0A0A240000000089
:0000000000
```

Pour convertir le dossier .HEX en dossier .COM, utilisez la commande LOAD, exemple:-

LOAD BLOW

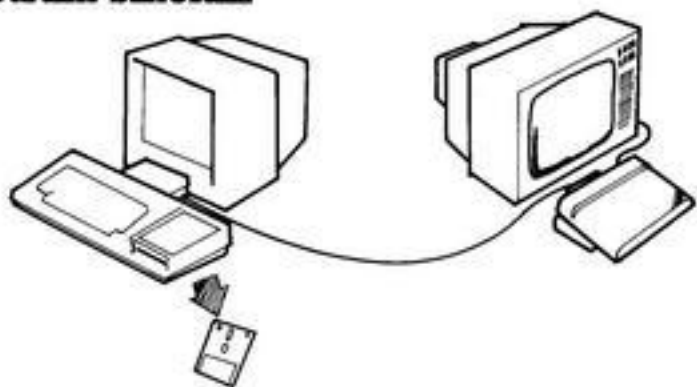
Le programme assume que l'entrée-sortie séquentielle de l'ordinateur CP/M est configurée suivant les systèmes de cartes perforées

Exécutez le programme en tapant:-

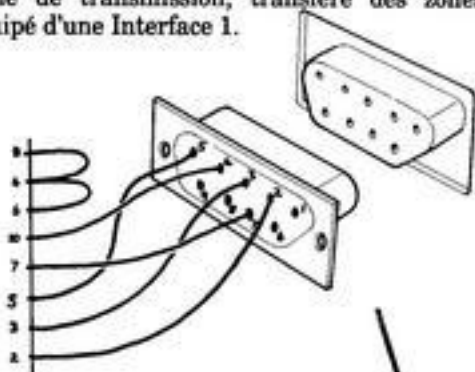
BLOW FILE.TYP



## Tour No 17 Transmission du dossier d'un Spectrum Sinclair



Ce programme de transmission, transfère des zones de mémoire d'un Spectrum, équipé d'une Interface 1.



Les données à transmettre devraient être chargées comme des données binaires, dans la zone de mémoire la plus élevée. Les petites erreurs de transmission seront automatiquement repérées.

```
5 INPUT "vitesse de transmission : ";baud
10 FORMAT "b";baud
20 OPEN #4;"b"
25 OPEN #3;"b"
30 INPUT "Tapez debut du texte en memoire : ";mstart
40 LET initl=mstart
50 INPUT "longueur du texte : ";mlen
55 INPUT "nom du dossier";f$
56 IF LEN f$>16 THEN GO TO 55
57 IF LEN f$<16 THEN LET f$=f$+" ": GO TO 57
60 PRINT "commencer le transfert"
100 LET blkno=0
105 LPRINT CHR$(2);
110 LET a$=INKEY$#4: IF a$="" THEN GO TO 110
120 IF a$=CHR$(3) THEN PRINT "TRANSFERT ABANDONNE":
    BEEP 1,5: GO TO 9000
130 IF a$<>CHR$(6) THEN GO TO 105
140 LPRINT f$;
150 LET bkh=INT (blkno/256)
160 LET bkl=blkno-bkh*256
170 LPRINT CHR$(bkl);CHR$(bkh);
180 LET chksm=0
200 LET ftn=mlen+initl-mstart
220 IF ftn>=128 THEN LET ftn=128
260 LPRINT CHR$(ftn);
265 IF ftn=0 THEN GO TO 300
270 PRINT ftn: FOR t=1 TO ftn
280 LET mloc=PEEK (mstart): LPRINT CHR$(loc);: LET
    chksm=chksm+mloc
```

(continue à la page suivante)

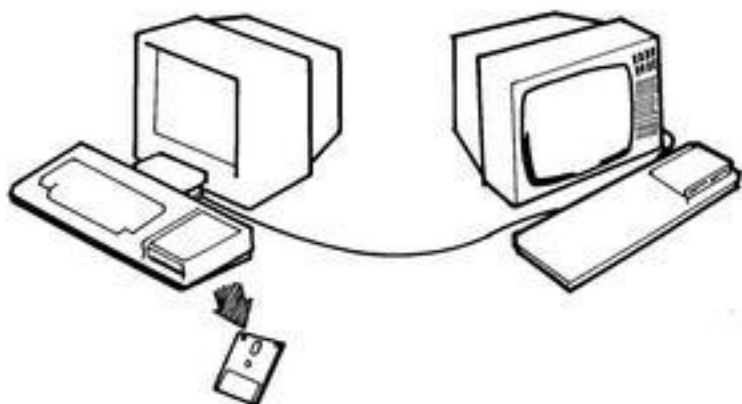




```
290 LET mstart=mstart+1: NEXT t
300 LET ckh=INT (chksm/256)
310 LET ckl=chksm-(ckh*256)
320 LPRINT CHR$ (ckl); CHR$ (ckh);
330 LET a$=INKEY$#4
335 IF a$=CHR$ (6) AND ftn=0 THEN GO TO 500
340 IF a$=CHR$ (6) THEN LET blkno=blkno+1: GO TO 140
350 IF a$=CHR$ (21) THEN LET mstart=mstart-ftn: GO TO
  140
360 IF a$<>CHR$ (3) THEN GO TO 330
370 GO TO 120
500 PRINT "TRANSFERT COMPLET"
510 BEEP 1,1
9000 CLOSE #3: CLOSE #4
```



## Tour No 18 Transmission du dossier d'un QL Sinclair



Ce programme de transmission envoie des données du port SER2 du QL, et nécessite un câble RS232 QL et un câble modem-nul. Les données à transmettre doivent d'abord être mise en mémoire à l'aide de la commande LBYTES. Les petites erreurs de transmission produiront automatiquement un message.



```

100 CCLC
110 INPUT "vitesse de transmission: ";b
120 BAUD b
130 OPEN #3, ser2
140 INPUT "Tapez debut de texte en memoire: ";mstart
150 initl=mstart
160 INPUT "Tapez la longueur du texte" : ";mlen
170 INPUT "Nom du dossier : ";fs
180 IF LEN (fs)>16 THEN GO TO 170
190 IF LEN (fs)<16 THEN fs=fs&" " : GO TO 190
200 PRINT "commencer le transfert"
210 blkno=0
220 PRINT #3,CHR$(2);
230 a$=INKEY$(#3): IF a$="" THEN GO TO 230
240 IF a$=CHR$(3) THEN PRINT "TRANSFERT ABANDONNE!!":
    BEEP 1000,50: GO TO 490
250 IF a$<>CHR$(6) THEN GO TO 220
260 PRINT#3,f$;
270 bkh=INT (blkno/256)
280 bkl=blkno-bkh*256
290 PRINT#3,CHR$(bkl);CHR$(bkh);
300 chksm=0
310 ftn=mlen+initl-mstart
320 IF ftn>=128 THEN ftn=128
330 PRINT#3,CHR$(ftn);
340 IF ftn=0 THEN GO TO 380
350 FOR t=1 TO ftn
360 mloc=PEEK(mstart): PRINT#3, CHR$(mloc);:
    chksm=chksm+mloc
370 mstart=mstart+1: NEXT t
380 ckh=INT (chksm/256)
390 ckl=chksm-(ckh*256)

```

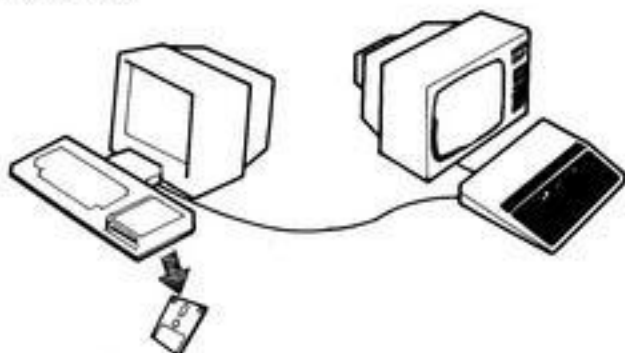
(continue à la page suivante)



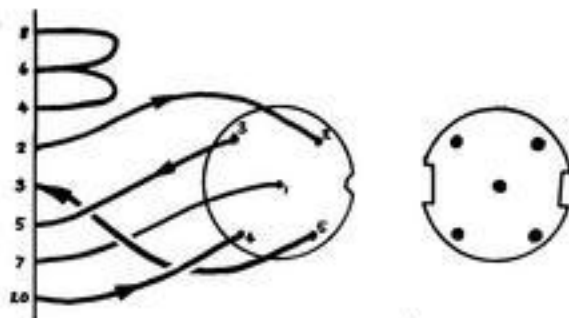
```
400 PRINT#3,CHR$(ckl);CHR$(ckh);
410 a$=INKEY$(#3)
420 IF a$=CHR$(6) AND ftn=0 THEN GO TO 470
430 IF a$=CHR$(6) THEN blkno=blkno+1: GO TO 260
440 IF a$=CHR$(21) THEN mstart=mstart-ftn : GO TO 260
450 IF a$<>CHR(3) THEN GO TO 410
460 GO TO 240
470 PRINT "transfert termine"
480 BEEP 6000,12
490 CLOSE #3
```



## Tour No 19 Transmission du dossier d'un micro BBC Acorn



Ce programme de transmission nécessite l'utilisation de OS1.1 (ou versions plus récentes). Les données d'un dossier particulier sont transmises à travers le port RS423.



```
100 PRINTCHR$(12)''
110 INPUT"dossier a transferer";FS
120 X=OPENIN FS
130 *FX3,7
140 REPEAT: FS=FS+" ": UNTIL LEN (FS)=16
150 BN=-1
160 REPEAT
170 PRINT CHR$(2);
180 PROCwait
190 UNTIL W<>0
200 REPEAT
210 BN=BN+1
220 PROCgetblock
230 REPEAT
240 PRINT FS;
250 PROCdbyte (BN)
260 PRINT CHR$(N);
270 PRINT AS;
280 PROCdbyte(CK)
290 PROCwait
300 UNTIL W<>0
310 UNTIL EOF#X
320 BN=BN+1
330 REPEAT
340 PRINT FS;
350 PROCdbyte (BN)
360 PRINT CHR$(0);
370 PROCdbyte (0)
380 PROCwait
390 UNTIL W<>0
400 *FX3,0
410 PRINT""TRANSFERT COMPLET""
420 *FX2,0
```



```
430 CLOSE#X: END
440 DEFPROCgetblock
450 ASQ="": N=0: CK=0
460 REPEAT
470 N=N+1
480 A=BGET#X
490 CK=CK+A
500 AS=AS+CHRS(A)
510 UNTIL (N=128)OR(EOF#X)
520 ENDPROC
530 DEFPROCdbyte(BX)
540 J=BX DIV 256
550 BX=BX-J*256
560 PRINT CHRS(BX);CHRS(J);
570 ENDPROC
580 DEFPROCwait
590 *FX2,1
600 K=GET
610 IF K=3 THEN 650
620 IF K=6 THEN W=6 ELSE W=0
630 *FX2,0
640 ENDPROC
650 *FX3,0
660 PRINT CHRS(7);"TRANSFERT ABANDONNE": GOTO 420
```



# Appendice 1

## Connecter la RS232C

---

Pour comprendre exactement ce qui est nécessaire à la connexion entre la RS232C et le monde extérieur, il est important de savoir que tous les dispositifs ayant une interface série peuvent être classifiés comme modem ou terminal. Les modems sont tout simplement une façon d'allonger la connexion (souvent par un câble téléphonique), le schéma 1 (ci-dessous) montre une connexion simplifiée et idéalisée, entre deux terminaux.



Connexion idéalisée entre terminaux

### Schéma 1

Le connecteur utilisé pour les interfaces série a 25 plots, bien que le plus souvent, 7 seulement, sont utilisés. Pour connecter un terminal à un modem, on utilise un câble 'un à un', c'est à dire plot 1 vers plot 1, plot 2 vers plot 2 et ainsi de suite. Si ces câbles sont utilisés, les données seront transférées de cette façon:

Après le trajet du signal de gauche à droite, les caractères tapés sur le clavier sortent du plot 2 du terminal gauche et vont vers le plot 2 du modem (appelé 'transmission de données'). Ensuite, le modem gauche envoie les caractères, par la ligne téléphonique, au modem de droite. Les caractères sont reçus par le plot 3 du modem de droite (appelé 'réception de données'), qui les envoie au plot 3 du terminal de droite. Après réception de ces données, le terminal de droite les affichera sur son écran.

Notez que les noms des connexions 'transmission de données' et 'réception de données' ont été choisis en fonction du terminal et non pas du modem.



Le trajet des données, de gauche à droite, que nous venons de décrire, est exactement le même que le trajet des données de droite à gauche, avec les mêmes numéros de connexions, c'est-à-dire, le plot 2 du terminal vers le modem (transmission) et ensuite le plot 3 du modem vers le terminal (réception). Ce format est bien symétrique et évite toute confusion dans l'utilisation des plots, ou la direction des transferts de données.

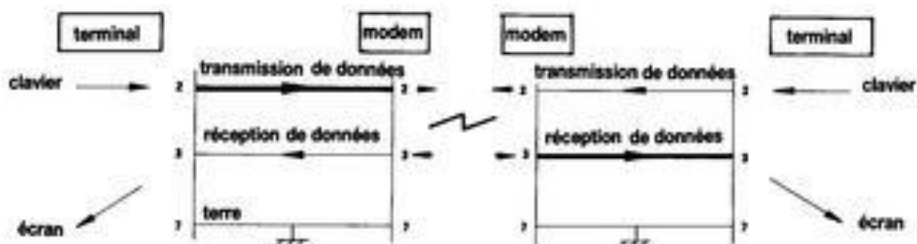


Schéma 2

Cependant, vous rencontrerez des problèmes de définition lorsque vous essaieriez de connecter deux terminaux ensemble, localement, sans l'aide d'une paire de modems. On ne pourra pas connecter le plot 2 au plot 2 parce que les deux claviers transmettront en face à face, et les écrans ne seront pas connectés pour envoyer. La meilleure solution serait de croiser les plots 2 et 3 afin que le plot de transmission de chaque terminal soit connecté au plot de réception de l'autre. Le câble utilisé pour ce genre de connexion est appelé un câble 'Modem nul' car il peut remplacer deux modems de même sexe.

Le plot de terre (plot 7) est commun au deux terminaux utilisant cet arrangement.

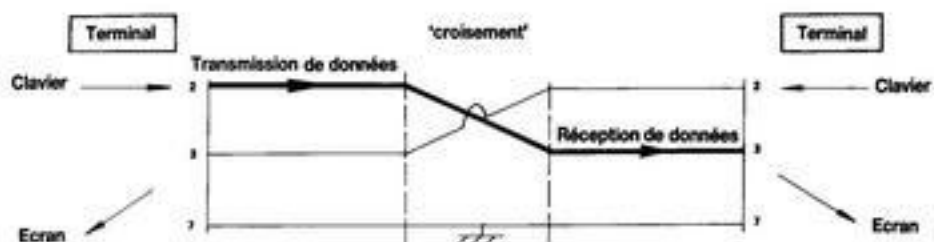


Schéma 3

Comme un ordinateur Amstrad + une RS232C peuvent servir de terminal, pour les connecter à un modem, il suffira d'utiliser un simple câble 'un à un' (par exemple, pour accéder à une base de données téléphonique).

On devra utiliser un câble modem-nul pour connecter aux autres terminaux. Ce que nous entendons par 'autres terminaux' peut être, un deuxième ordinateur Amstrad + RS232C, un terminal de visualisation ordinaire (VDU), une imprimante avec une interface série, ou même un ordinateur de bureau nécessitant un VDU.

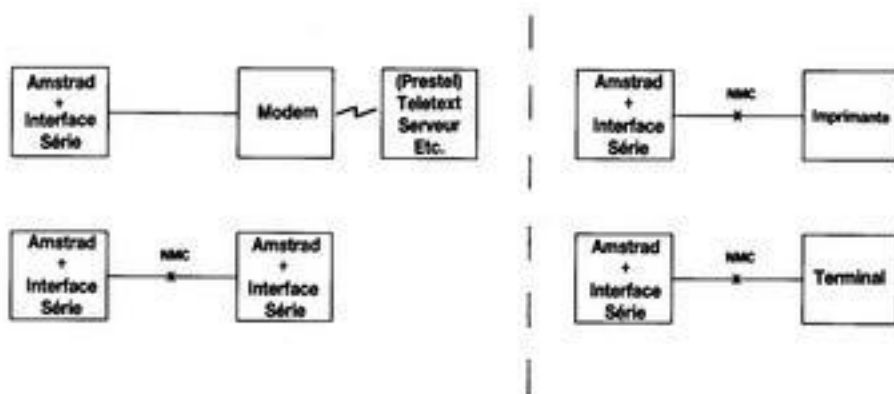


Schéma 4

Il faut noter que certains fabricants d'ordinateurs de bureau branchent l'interface série (d'un VDU ou d'une imprimante) comme un modem plutôt qu'un terminal, ceci parce qu'ils pensent faciliter les choses, en permettant aux imprimantes et VDUs d'être connectés à cet ordinateur par des câbles 'un à un'.



Schéma 5

---

Dans un monde de perfection, il serait possible d'identifier quels dispositifs série se comportent en modems et lesquels se comportent en terminaux, en examinant le 'sexe' du connecteur à 25 plots; les terminaux auraient un connecteur mâle, et les modems un connecteur femelle. Ceci n'est, malheureusement, pas aussi simple, car certains fabricants de terminaux et d'imprimantes les équipent de connecteurs femelle, par mesure de sécurité.

En cas de doute, le test ultime est d'examiner le manuel de l'utilisateur et de déterminer la fonction de PLOT 2; si la description comporte le mot 'TRANSMISSION', le matériel doit être branché comme un terminal, et s'il comporte le mot 'RECEPTION', il est branché comme un modem.

## **Contrôle de déroulement du matériel**

La connexion simplifiée qui a été décrite jusque là, ne permet pas le contrôle du déroulement des données. Souvent, nous aimerions que le dispositif de réception puisse contrôler le dispositif de transmission, ce qui empêcherait le dispositif de réception d'être submergé (s'il ne suit pas la vitesse d'arrivée des entrées). De plus, si le dispositif de transmission n'a pas confiance dans les données qu'il envoie, il devrait avoir un moyen de mettre le dispositif de réception, hors service.

Dans le cas d'une connexion d'un modem à un terminal, si le terminal est satisfait qu'il peut transmettre, il active le plot 4 - le RTS (Request To Send = demande d'envoi). Si le modem est prêt à recevoir, il active le plot 5 - le CTS (Clear To Send = Prêt à recevoir). Le terminal enverra seulement si le CTS est activé. De cette façon, le modem peut contrôler la vitesse du déroulement à l'aide de CTS.

Quand le modem a décidé que les données qu'il s'apprête à envoyer sont les bonnes, il active le plot 8 - le DCD (Data Carrier Detect = détection de support de données). Si le terminal est prêt à recevoir des entrées, il active le plot 20 - le DTR (Data Terminal Ready = terminal prêt). Le modem ne transmettra que lorsque le DTR est activé. Ainsi, le terminal peut contrôler la vitesse de déroulement à l'aide de DTR.

Il reste deux signaux à introduire, l'un d'eux est le plot 22 - l'Indicateur de Sonnerie, qui tout simplement indique au modem de faire savoir au terminal que le téléphone sonne! (à ce moment, on s'attend au réveil du logiciel se trouvant dans le terminal). L'autre signal est sur le plot 6 -DSR (Data Set Ready = Poste de données prêt). Ce signal est ignoré par le côté transmission de la RS232C; le modem activera ce signal à peu près en même temps que le DCD, de cette façon aucune fonction ne sera perdue.

CONNEXIONS D'UN MODEM

INTERFACE SERIE

MODEM

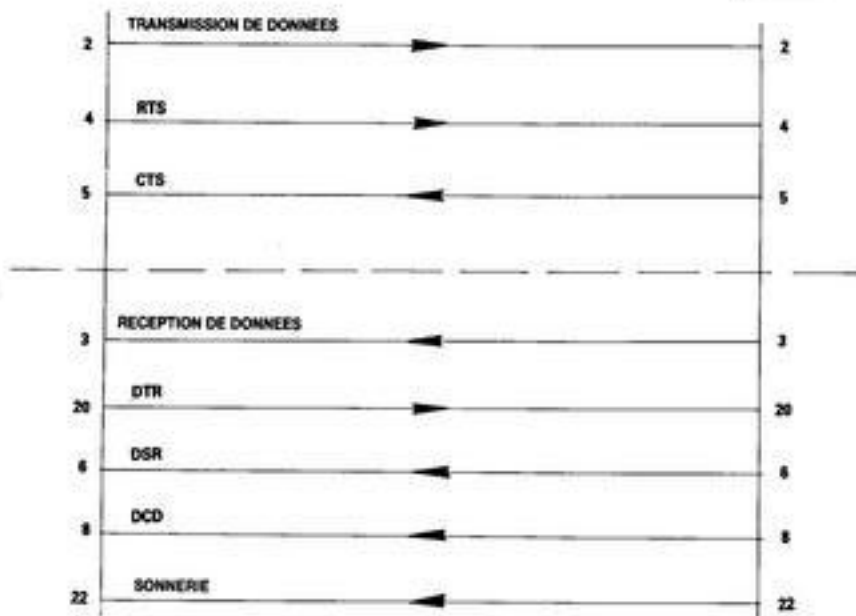


Schéma 6

Pour les connexions entre terminaux, le câble modem-nul doit être utilisé, avec les connexions des plots 2, 3 et 7. Le câble modem-nul échange les plots 4 et 8 : les signaux RTS/DCD, et les plots 20 et 5 : les signaux DTR/CTS. Par précautions, le plot 6 (DSR) est connecté au plot 8 (DCD), au cas où ce câble serait connecté à un terminal méticuleux qui demanderait les deux.

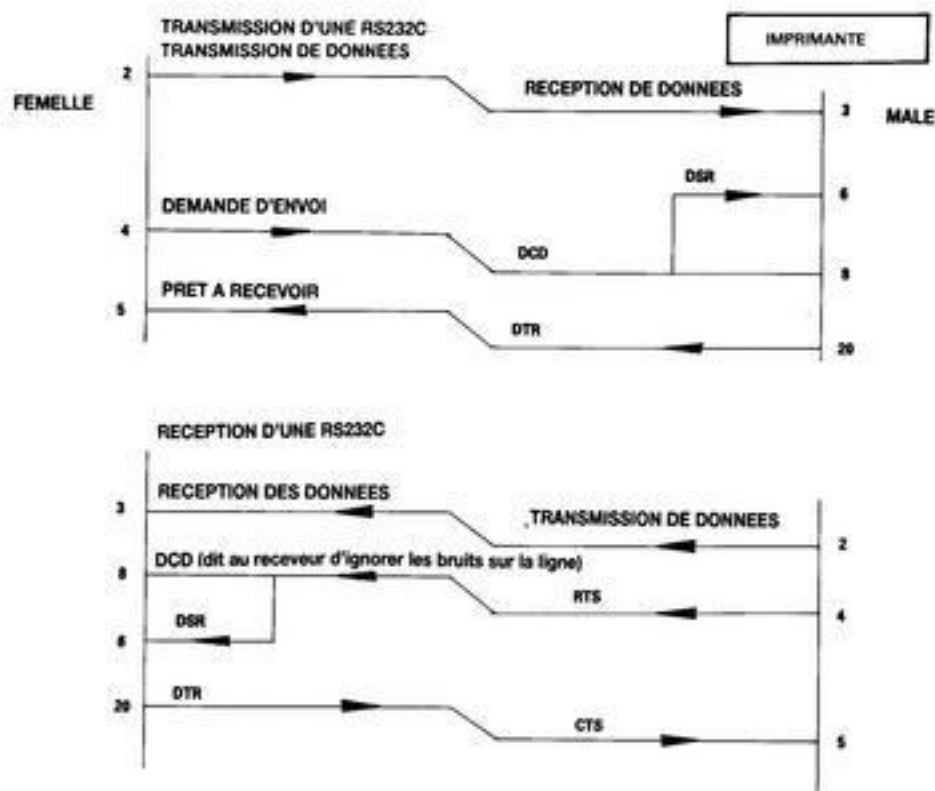


Schéma 7

L'idée générale est que le modem-nul, contrairement aux deux modems qu'il remplace, est TOUJOURS 'satisfait d'envoyer'. Il est donc raisonnable de générer DCD (et DSR) en permanence. Pour cela, il suffit de les connecter au RTS, à la même extrémité du câble, au lieu d'avoir le RTS à l'autre bout.

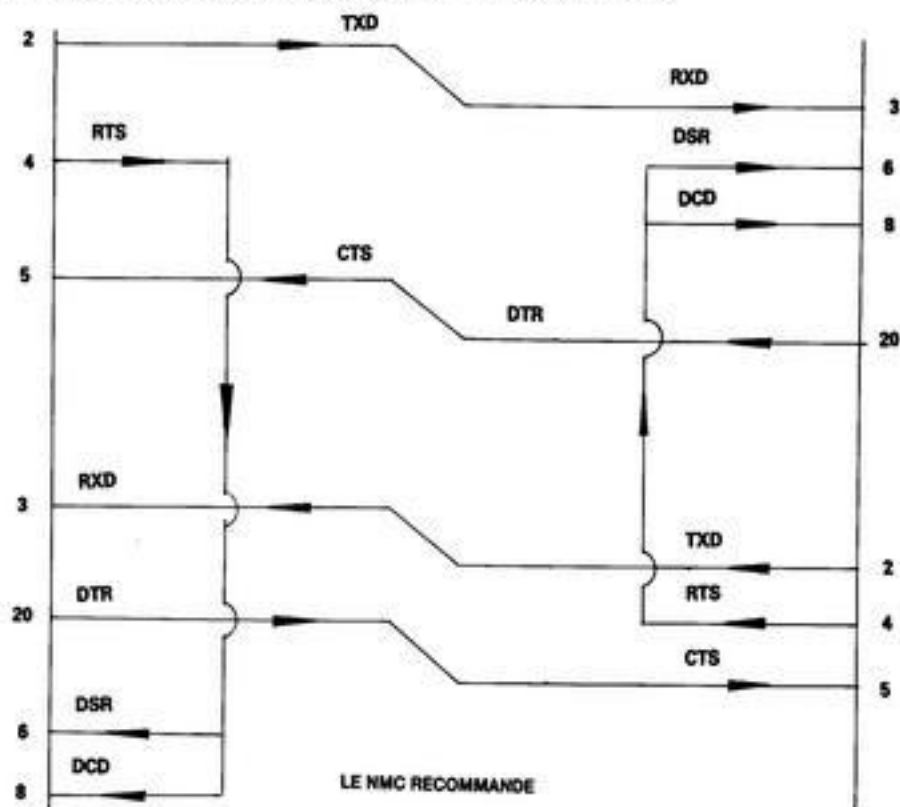


Schéma 8

Finalement, si la vitesse de transmission d'un des deux terminaux ne peut pas être arrêtée (par exemple, si quelqu'un tape sur le clavier), ou si elle est très lente (ex. les caractères d'échange de données du logiciel, 'XON, XOFF' sont envoyés par l'imprimante) et qu'il n'y a aucun danger de dépasser la vitesse de réception, il sera permis de valider, définitivement, la transmission en reliant le plot 5 (CTS) au plot 4 (RTS), c'est-à-dire, qu'il faudra toujours envoyer si c'est prêt. Il est quelquefois plus facile, pour les terminaux de transmission, d'ignorer l'état du CTS.

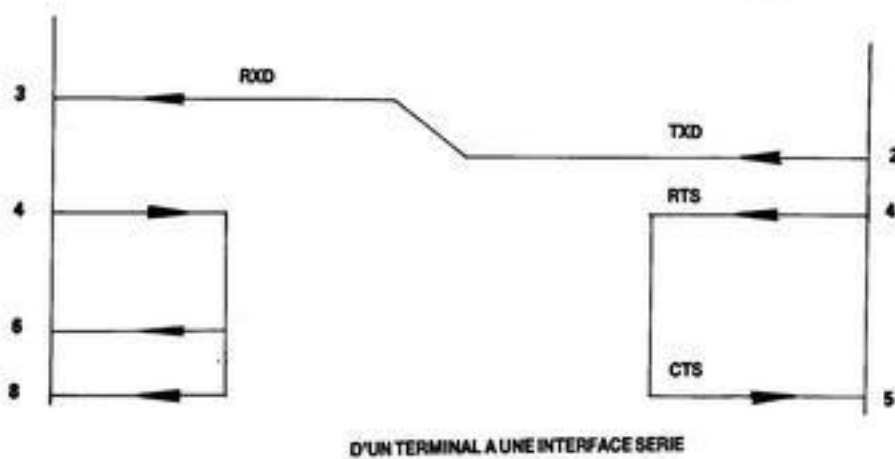
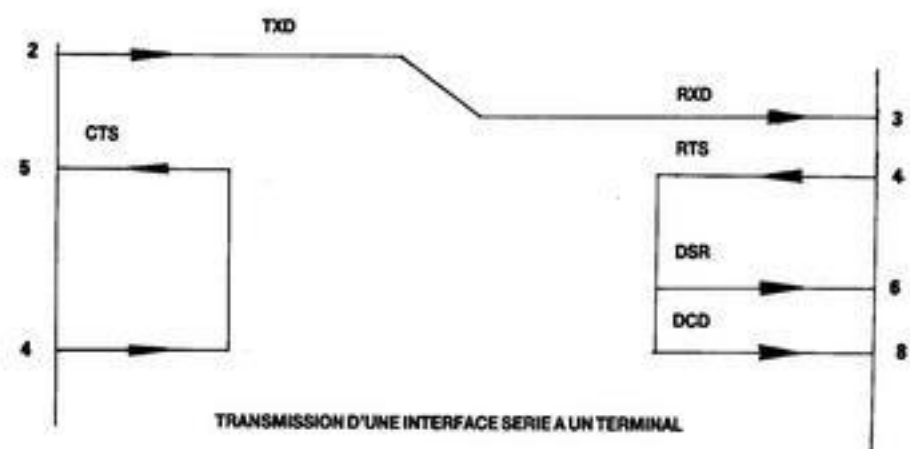


Schéma 9



## Appendice 2

# Utiliser la RS232C avec CP/M

---

Le programme ROM fourni avec la RS232C n'est pas utilisé par CP/M. Il est possible, mais improbable, qu'un programme CP/M accède le ROM de la RS232C en utilisant le firmware. Les possibilités décrites ci-dessous sont fournies avec le CP/M pour le DDI-1, le CPC664, ou le CPC6128.

NOTEZ:- Les utilisateurs de CP/M Plus version 1.0 du CPC6128, devront incorporer la petite modification encadrée, se trouvant à la fin de cette appendice.

Pour plus de détails sur les possibilités et programmes décrits ci-dessous, reportez vous aux manuels de firmware et de CP/M de votre ordinateur ou système de disquette.

### **CP/M 2.2 (fournit avec le DDI/1, le CPC664 et, pour assurer la compatibilité, avec le CPC6128):**

Tous les paramètres associés à la RS232C sont mis en route au moment où le CP/M 2.2 est chargé. Chaque disquette système comprend une 'zone configuration' qui contient les paramètres demandés. L'utilitaire **SETUP** permet d'éditer cette zone configuration.

L'AMSTRAD BIOS 2.2 exécute la RS232C comme le dispositif physique **TTY:**. **SETUP** et **STAT** vous permettent d'attribuer ce dispositif physique à une variété de dispositifs logiques et ainsi vous donne la possibilité de consulter la RS232C pendant l'exploitation de l'utilitaire **PIP**, ou quand vous écrivez des programmes d'assemblage qui ont accès à l'interface **BDOS (CALL 5 = appel 5)**. Le BIOS contient aussi un 'jumpblock' (saut de bloc de mémoire) étendu, qui permet au programmeur d'assemblage, de se servir directement de la RS232C.

**STAT** permet d'assigner à nouveau, les dispositifs physiques **TTY:**, **CRT:**, et **LPT:** aux dispositifs logiques **CON:**, **RDR:**, **PUN:**, et **LST:**.

La console (**CON:**) est normalement attribuée au clavier/écran (**CRT:**).

L'imprimante (**LST:**) est normalement assignée au port de l'imprimante parallèle (**LPT:**).

Le dispositif d'entrée (**RDR:**) et le dispositif de sortie (**PUN:**) sont normalement assignés à la RS232C (**TTY:**).

Les assignements actuels peuvent être interrogés par la commande:

```
STAT DEV:
```

---

et peuvent être changés de cette façon:-

STAT -dispositif logique- = -dispositif physique-

ex. STAT LST:=TTY:

qui assigne la RS232C à envoyer des sorties vers l'imprimante.

Si un assignement de sorties est attribué à un dispositif physique, constamment occupé ou qui n'existe pas, vous devrez taper **[CTRL]C** au clavier pour empêcher un 'verrouillage'.

PIP est un programme qui copie des dossiers, et traite les dispositifs logiques d'entrées et de sorties comme des dossiers. De cette façon...

PIP PUN:=FILE.TYP	enverra un dossier à la RS232C
PIP FILE.TYP=RDR:	recevra un dossier venant de la RS232C
PIP PUN:=CON:	envoie les entrées du clavier à la RS232C
PIP CON:=RDR:	envoie les entrées de la RS232C à l'écran

si les assignements par défaut sont en vigueur. Les transferts du clavier ou de la RS232C se terminent généralement par un **[CTRL]Z**.

SETUP vous permet, entre autre, de changer les paramètres de la RS232C. La configuration comprend le support d'une interface série à deux canaux - la RS232C est le canal A. Les défauts sont:-

- 9600 bauds
- 8 bits de données
- aucune parité
- 1 bit arrêt
- RDR: et PUN: sont assignés à la RS232C

Insérez une copie de la disquette système CP/M 2.2, mettez en route le CP/M, et dès que le prompt A> apparaît à l'écran, tapez:

SETUP

Passez aux questions suivantes en répondant Y (pour yes) jusqu'à la sixième section (qui est le réglage des octets IO par défaut).

Cette section permet de changer les assignements des dispositifs physiques et logiques (voir STAT). L'octet IO est un marqueur spécial storé à l'adresse 0003, qui donne les allocations des dispositifs logiques aux physiques. Les nouveaux assignements sont indiqués par des commandes telles que LST:=TTY: qui attribue l'imprimante à la RS232C.

---

La treizième section (réglage du canal A du Z80 SIO) permet un changement aux paramètres, qui doivent être affichés dans l'ordre suivant:-

Transmission (tx) de la vitesse en bauds (doit être un de ces chiffres, 19200, 9600, 4800, 3600, 2400, 2000, 1800, 1200, 600, 300, 200, 150, 110, 75 ou 50).

Reception (rx) de la vitesse en bauds (doit être un des chiffres ci-dessus).

Les bits 'données' (doivent être un de ces chiffres 5,6,7 ou 8).

La parité (doit être un de ces mots, ODD, EVEN ou NONE)

Les bits 'arrêt' (doivent être un de ces chiffres, 1, 1.5 ou 2).

Par exemple:

9600 9600 8 NONE 1

... qui correspondent aux réglages par défauts.

Vous avez aussi la possibilité de faire une mise à jour de la disquette système avec la nouvelle configuration. A ce moment, il vous sera donné le choix de recommencer avec CP/M (ce qui invoquera les nouveaux réglages de configuration de la RS232C).

## **CP/M Plus (fournit avec le CPC6128):**

Les paramètres de la RS232C sont mis aux valeurs par défaut quand le CP/M Plus est mis en route. Les paramètres peuvent être modifiés à l'aide du programme SETSIO. L'intention est que SETSIO sera invoqué en même temps que l'ouverture du dossier PROFILE.SUB.

L'AMSTRAD BIOS du CP/M Plus exécute la RS232C comme étant le dispositif physique SIO. L'utilitaire DEVICE vous permet de réassigner ce dispositif physique (qui se place par défaut au dispositif logique AUX:) et ainsi, vous permet de consulter la RS232C durant l'opération de l'utilitaire PIP ou pendant l'écriture de programmes d'assemblage qui ont accès à l'interface BDOS (CALL 5). Le BIOS contient aussi un 'jumpblock' qui permet au programmeur d'assemblage, d'obtenir le support direct de la RS232C.

DEVICE fournit la possibilité d'assigner à nouveau les dispositifs physiques SIO, CRT, et LPT aux dispositifs logiques CON: (qui peuvent être sub-divisés en AUXIN: et AUXOUT:), et LST:.

---

La console (CON:) est normalement attribuée à l'écran/clavier (CRT).

L'imprimante (LST:) est normalement attribuée au port de l'imprimante parallèle (LPT).

Le dispositif auxiliaire (AUX:) est normalement attribué à la RS232C (SIO).

Les assignements actuels peuvent être interrogés par la commande:

```
DEVICE
```

... et peuvent être changés de cette façon:

```
DEVICE -dispositif logique = -dispositif physique
```

ex.

```
DEVICE LST:=SIO
```

... qui assigne la RS232C à envoyer les sorties à l'imprimante, ou:

```
DEVICE CON:=SIO
```

... qui permet d'exploiter l'ordinateur, d'un terminal attaché à la RS232C (si vous tapez `DEVICE CON:=CRT` sur le terminal déporté, ceci renverra l'exploitation au clavier et à l'écran local).

Si un assignement de sorties est donné à un dispositif physique constamment occupé ou non-existant, l'ordinateur enverra un message afin d'empêcher un 'verrouillage'.

PIP est un programme qui copie des dossiers et traite les dispositifs logiques d'entrée et de sortie comme des dossiers.

Ainsi...

```
PIP AUX:=FILE.TYP      enverra un dossier à la RS232C
PIP FILE.TIP=AUX:      recevra un dossier d'un RS232C
```

... si les assignements par défaut sont en vigueur. Les transferts venant du clavier ou de la RS232C sont généralement terminés par un **[CTRL]Z**.

SETSIO affiche et modifie les paramètres de la RS232C. Les paramètres peut être changés en ajoutant les clauses suivantes, dans n'importe quel ordre:

par exemple -

```
SETSIO TX 75 RX 1200 BITS 7 PARITY ODD
```

---

La lettre initiale du premier mot seulement, est nécessaire, le reste de ce mot est facultatif. Une clause illégale produira un message d'erreur, si une clause est spécifiée deux fois, la deuxième sera utilisée.

Si le nom de la vitesse de transmission est omise (TX ou RX), les deux seront mise en marche. Si la vitesse de transmission est changée et que le nombre de bits 'arrêt' ne l'est pas, il sera mis à 1 (si la vitesse de transmission est supérieure à 110), sinon il sera mis à 2.

Les clauses existantes sont:

**TX** <vitesse de transmission> - par défaut 9600 - règle l'émetteur de vitesse de transmission

**RX** <vitesse de transmission> - par défaut 9600 - règle le receveur de vitesse de transmission.

... la vitesse de transmission doit être un de ces chiffres, 50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2400, 3600, 4800, 7200, 9600 ou 19200.

**BITS** <n> - par défaut 8 - met le nombre de bits 'données'

... <n> doit être un de ces nombres, 5, 6, 7 ou 8.

**PARITY** <p> - par défaut NONE - met le type de parité.

... <p> doit être un de ces mots EVEN, ODD ou NONE.

**STOP** <n> - par défaut 1 - met le nombre de bits 'arrêt'

... <n> doit être un de ces nombres, 1, 1.5, ou 2.

**XNON** <x> - par défaut OFF - invoque ou annule le contrôle du déroulement du logiciel

... <x> doit être un de ces mots, ON ou OFF.

## CP/MPlus - version 1.0

Le 'raccord' suivant doit être appliqué à la version 1.0 du CP/M Plus si vous voulez l'utiliser avec la RS232C. Deux disquettes seront nécessaires:

La disquette insérée dans l'unité B, doit être la face 2 de la disquette système fournie avec le CPC6128.

La disquette insérée dans l'unité A, devrait être une COPIE validée pour l'écriture de la face 1 de la disquette système (ou de la disquette utilisée pour charger le CP/M Plus).

Maintenant mettez en route le CP/M Plus, et tapez SEULEMENT les caractères normaux, imprimés ci-dessous en vous assurant que l'ordinateur répond comme indiqué, par des caractères en **blobs inverses**.

```
A>B:SID C10CPM3.EMS
CP/M 3 SID - Version 3.0
NEXT MS2E PC END
6500 6500 0100DAFF
# S4EF
04EF B7 57
04F0 28 ED
04F1 08 78
04F2 ED 1F
04F3 78 38
04F4 1F 0A
04F5 38 7A
04F6 08 B7
04F7 CD C4
04F8 8B .
# WC10CPM3.EMS
00C8h record(s) written.
# [CTRL]C
A> REN P11CPM3.EMS=C10CPM3.EMS
A>
```

## APPENDICE 3

# Le ROM de la RS232C en BASIC et en code machine

---

Le ROM fournit dans votre RS232C contient un programme qui ajoute, automatiquement un nombre de commandes 'externes' au BASIC. Ces commandes sont identifiées par le symbole '@' (barre) (obtenu en appuyant sur [SHIFT]@). Les programmes en code machine peuvent accéder ces commandes, à l'aide du firmware KL\_FIND\_COMMAND et en préparant des blocs paramètres appropriés.

Reportez vous à la Spécification Concise du Firmware de votre ordinateur AMSTRAD.

Le programme ROM réserve, automatiquement, une allocation pour RAM, réduisant ainsi, la place réservée à vos programmes en BASIC. Les ordinateurs équipés de la RS232C ont à peu près 40K de disponible, alors que ceux qui ont aussi une unité de disquette en auront 39K. (CP/M ne crée aucune perte de mémoire disponible, lors de son utilisation avec la RS232C).

Les commandes externes de la RS232C sont groupées en catégories pour pouvoir s'y référer facilement. Les conventions suivantes sont suivies lors de la description des commandes:

1. Les paramètres entre [parenthèses carrées] sont facultatifs.
2. «état» est l'adresse d'une variable entière (à laquelle on aura, auparavant donné une valeur artificielle) dans laquelle un code, représentant le succès ou l'échec de la commande, est placé. L'adresse d'une variable entière est calculée par BASIC si le nom de la variable est précédé d'un symbole @.

par exemple:       SX=@: IRINGWAIT,@SX,60

3. «chaîne d'entrée» représente l'adresse d'une variable de chaîne, à laquelle on aura donné auparavant, une chaîne artificielle contenant au moins, autant de caractères que vous devriez recevoir durant l'exécution de la commande. L'adresse d'une variable de chaîne est calculée par BASIC si la chaîne est précédée du symbole @.

par exemple:

SX=@: RCVD\$=STRING\$(100," "): IINBLOCK,@SX,@RCVD\$

4. «chaîne de sortie» représente l'adresse de la chaîne qui doit être envoyée par la commande. Dans la version 1 du BASIC (CPC464), cette adresse doit être celle d'une variable de chaîne, qui est indiquée par le symbole @ la précédent. Dans les versions plus récentes du BASIC (CPC664 et CPC6128), le symbole @ est facultatif et les chaînes constantes peuvent être spécifiées.

---

par exemple:

```
SX=0: TRANSS="une version du BASIC": IOUTBLOCK,@SX,@TRANSS  
SX=0: TRANSS="Version 2 ou plus recente": IOUTBLOCK,@SX,TRANSS
```

... ou ...

```
SX=0: IOUTBLOCK,@SX,"Version 2 ou plus recente"
```

5. *·caractère d'entrée·* représente l'adresse d'un entier dans lequel la valeur ASCII d'un seul caractère d'entrée, sera placée. On aura donné, auparavant, une valeur artificielle à l'entier. L'adresse d'une variable entière est calculée par BASIC si le nom de la variable est précédé d'un symbole @.

6. *·caractère de sortie·* représente un entier ou une vraie variable, une expression constante ou numérique, qui prend la valeur ASCII du caractère (entre 0 et 255).

par exemple:        IOSTBLOCKEND,13 '    retour de chariot

7. *·nom du dossier·* représente le nom d'un dossier d'une cassette ou disquette. La version 1 du BASIC (CPC464) demande à ce que le nom du dossier soit spécifié comme étant l'adresse d'une variable de chaîne, qui est indiquée par le symbole @ la précédant. Dans les versions plus récentes du BASIC (CPC664 et CPC6128) le symbole @ est facultatif et les chaînes constantes peuvent être spécifiées.

8. *·autres paramètres, pas encore mentionnés·* représentent un entier ou une vraie variable, une expression constante ou numérique, qui prend la valeur du paramètre décrit.

## Les commandes pour le service ROM

### I ROMOFF

#### I ROMOFF

COMMANDE: initialise à nouveau le BASIC (et perd le programme actuellement chargé et les valeurs de toutes les variables) après avoir mis hors service le mécanisme qui enregistre automatiquement les commandes externes ROM. Ceci redonne la liberté maximum au RAM, et permet de pouvoir charger de très longs programmes à partir d'une cassette (la disquette ROM sera elle aussi mise hors service) ce qui mettrait normalement RAM en conflit avec la disquette, la RS232C et les autres ROMs.



---

## | ROMCAT

| ROMCAT ( , Nombre ROM )

COMMANDE: Affiche un catalogue (répertoire) d'un maximum de 16 ROMs externes, actuellement enregistrés. Si le paramètre facultatif est spécifié, le répertoire donne les commandes externes, disponibles à partir de ce ROM. Notez que les emplacements vides ont l'air d'être occupés par BASIC.

## Commandes pour le service RS232C

### | SETSIO

| SETSIO , vitesse de transmission ( , réception de la vitesse ( , contrôle de déroulement du matériel ( , bits de données ( , parité ( , bits d'arrêt ) ) ) ) )

COMMANDE: Modifie les paramètres fondamentaux d'exploitation de la RS232C. Après le premier paramètre, les suivants sont facultatifs. Si un paramètre invalide est spécifié, tous les paramètres suivants seront ignorés.

Les paramètres 'bits de données', 'parité', et 'bits d'arrêt' sont appelés, collectivement, les 'bits de synchronisation'.

Le paramètre de vitesse de transmission (9600 bauds par défaut) est un entier qui spécifie la valeur de la vitesse de transmission. Si une 'réception de vitesse de transmission' est spécifiée, le premier paramètre ne donnera que la transmission de vitesse en bauds.

Le paramètre 'réception de vitesse de transmission' (9600 bauds par défaut) est un entier qui spécifie la vitesse de transmission.

Le paramètre 'contrôle du déroulement du matériel' (par défaut ENABLED) est un entier qui valide ou met hors service le contrôle du déroulement du matériel. Le zéro met le contrôle du déroulement hors service, autre que zéro le valide.

Si le contrôle du déroulement du matériel est mis hors service, la RS232C transmettra, quelque soit l'état du contrôle du déroulement CTS sur le plot 5, et recevra quelque soit l'état du DCD sur le plot 8. Si le contrôle du déroulement du matériel est mis hors service, la RS232C activera en permanence le signal de contrôle du déroulement DTR sur le plot 20 jusqu'à ce que la commande | CLOSESIO soit reçue ou que le contrôle de déroulement du matériel soit revalidé.

---

La mise hors service du contrôle du déroulement est parfois utile si le câble ou le matériel connecté à la RS232C ne peut pas gérer les signaux du contrôle de déroulement. Un des désavantages de la mise hors service du contrôle de déroulement du matériel est que vous risquez de perdre des caractères si le dispositif de réception à un système de départ/arrêt plus lent que le dispositif de transmission.

Le paramètre `·bits de données·` est un entier entre 5 et 8, qui spécifie le nombre de bits 'données' en caractères transmis et reçus.

Le paramètre `·parité·` (par défaut, AUCUNE parité) est un entier entre 0 et 2, qui spécifie la parité à générer et à vérifier. 0 veut dire AUCUNE parité, 1 est parité ODD (impaire) et 2 est `·parité EVEN` (paire).

Le paramètre `·bits d'arrêt·` (par défaut, un bit 'arrêt') est un entier entre 0 et 2, qui spécifie le nombre de bits 'arrêt' en caractères transmis et reçus. 0 spécifie un bit 'arrêt', 1 est un bit et demi 'arrêt' et 2 est deux bits 'arrêt'.

Les vitesses de transmission permises sont: 19200, 9600, 4800, 3600, 2400, 2000, 1800, 1200, 600, 300, 200, 150, 110, 75, 50.

par exemple:

```
ISETS10,300 ' 300 bauds
ISETS10,9600,9600,0 ' met hors service le contrôle
                    du déroulement, 9600 bauds
ISETS10,75,1200,1,7,1,0 ' TX75 baud, RX1200 baud,
```

valide le contrôle du déroulement du matériel, 7 bits 'données', parité impaire, 1 bit 'arrêt'.

## **| SETTIMEOUT**

`| SETTIMEOUT, ·temps d'arrêt·`

COMMANDE: Modifie le temps d'arrêt après lequel, une coupure, ou un caractère ou une commande de transfert de bloc 'abandonnera' s'il n'est pas encore achevé.

Le paramètre `·temps d'arrêt·` (0mS par défaut) est un entier entre 1 et 65534, qui spécifie la longueur du temps d'arrêt en millisecondes. La valeur -1 met l'arrêt hors service, c'est-à-dire, rend l'arrêt infini.

par exemple:

```
ISSETTIMEOUT,1000 ' une seconde d'arrêt
```

---

## ISIO

ISIO, état

COMMANDE: Renvoie un état générale de l'interface série. Cette commande est pour les spécialistes seulement, car les autres commandes interrogent automatiquement les bits d'état, demandés durant leur exécution. L'état est renvoyé sous forme d'entier binaire à 16 bits:

Bit 15	(MSB)	0
Bit 14		Erreur de synchronisation
Bit 13		Embouteillage
Bit 12		Erreur de parité
Bit 11		0
Bit 10		0
Bit 9		0
Bit 8		Tout est envoyé
Bit 7		Coupure reçue
Bit 6		0
Bit 5		CTS
Bit 4		Détecteur de sonnerie
Bit 3		DCD
Bit 2		Transmetteur buffer est vide
Bit 1		0
Bit 0	(LSB)	Données reçues disponibles

Sous-programme d'exemple:

```
110 ' simule RINGWAIT,@SX,100
120 SX=0: T=TIME+(100*300)
130 WHILE TIME<T
140 ISIO,@SX
150 IF SX AND 16 THEN SX=0: RETURN
160 WEND
170 SX=1: RETURN
```

## IRINGWAIT

IRINGWAIT, état, temps d'arrêt

COMMANDE: Attend que le Détecteur de sonnerie soit vrai, ou que l'arrêt soit terminé. Cette commande est destinée à être utilisée dans une configuration de modems à réponse automatique. L'état est renvoyé sous forme d'entier, 0 indique que le Détecteur de sonnerie est vrai et 1 indique que l'arrêt a eu lieu.

---

Le paramètre `<temps d'arrêt>` est un entier entre 0 et 65535 qui règle le temps d'arrêt en secondes.

NOTEZ: Cet arrêt est indépendant de celui associé à la coupure, aux caractères et aux opérations de transferts de blocs.

## **| BREAKSEND**

`| BREAKSEND , <état> , <temps de coupure>`

COMMANDE: Attend que le buffer transmetteur soit vide ('tout est envoyé' état vrai) et ensuite envoie une 'coupure' pendant que la ligne continue à 'marquer'. Cette coupure est destinée à servir de signal et peut être détectée par le matériel connecté à la RS232C.

L'état renvoyé est:

- 0\*256- La coupure a bien été envoyée
- 2\*256- Arrêt, l'état 'tout est envoyé' n'a pas eu lieu pendant le temps d'arrêt spécifié auparavant, donc la coupure n'a pas été envoyée.

Le paramètre `<temps de coupure>` est un entier entre 0 et 65535 (0 veut dire 65536), qui spécifie pendant combien de millisecondes, la ligne marque.

## **| CLOSESIO**

`| CLOSESIO , <état>`

COMMANDE: Attend que le buffer transmetteur soit vide ('tout est envoyé' état vrai) et ferme la RS232C. La fermeture de la RS232C éteint en même temps les signaux RTS plot 4 et DTR plot 20 du contrôle du déroulement du matériel.

L'état renvoyé est:

- 0\*256- La RS232C est bien fermée
- 2\*256- Arrêt, l'état 'tout est envoyé' n'a pas eu lieu pendant le temps d'arrêt, spécifié auparavant, donc la RS232C n'a pas été fermée.

---

## Commandes pour les transferts de Caractères

### | INCHAR

| INCHAR, «état», «caractère d'entrée»

COMMANDE: Lit un seul caractère venant de la RS232C. Si le contrôle du déroulement du matériel est validé, le signal DCD du plot 8 doit être vrai, la RS232C activera alors, DTR du plot 20, s'il n'y a pas de données disponibles.

L'état renvoyé est:

- 0\*256- Un caractère a bien été lu
- 1\*256- Aucun «caractère d'entrée» a été spécifié
- 2\*256- Arrêt, aucun caractère n'a été lu pendant le temps d'arrêt spécifié auparavant
- 3\*256- Une coupure de ligne a été reçue
- 4\*256- Un caractère a été lu avec une erreur de synchronisation
- 5\*256- Un caractère a été lu avec un embouteillage
- 6\*256- Un caractère a été lu avec une erreur de parité.

S'il existe plus d'une erreur de matériel, (coupure, synchronisation, embouteillage et parité) seulement l'erreur portant le plus petit nombre sera reportée. Tous les bits d'état d'erreurs sont remis à zéro.

### | OUTCAR

| OUTCAR, «état», «caractère de sortie»

COMMANDE: Attend que le buffer transmetteur soit vide, ensuite envoie un seul caractère à la RS232C. Si le contrôle du déroulement du matériel est validé, le caractère ne sera envoyé que si le signal CTS du plot 5 est vrai. A l'allumage ou après avoir appelé |SIO, le signal RST du plot 4 est activé jusqu'à l'appel de |CLOSESIO.

L'état renvoyé est:

- 0\*256- Le caractère a bien été envoyé
- 1\*256- Aucun «caractère de sortie» n'a été spécifié
- 2\*256- Arrêt, Le caractère n'a pas pu être envoyé pendant le temps d'arrêt spécifié auparavant.

---

## Commandes pour les transferts de Blocs

### | SETBLOCKEND

| SETBLOCKEND , caractère de sortie

COMMANDE: Spécifie un caractère à la fin d'un marquage de bloc. Une valeur de 256 ou supérieure, met la fin du contrôle de bloc hors service (Le réglage par défaut est DISABLED).

### | INBLOCK

| INBLOCK , état , chaîne d'entrée

COMMANDE: Lit une chaîne de la RS232C. L'exploitation peut s'achever soit, lors d'un arrêt, à la fin de la lecture du marquage d'un bloc ou quand la chaîne de caractère est remplie. Si le contrôle du déroulement du matériel est validé, le signal DCD du plot 8 doit être vrai, la RS232C activera alors DTR du plot 20 s'il n'y a pas de données.

L'état renvoyé est:

- 0\*256+0 - Le caractère d'entrée a bien été rempli.
- 0\*256+N - Les caractères N et la fin du marquage de bloc, ont été reçus
- 1\*256+0 - Aucune chaîne d'entrée ou longueur zéro n'a été spécifiée
- 2\*256+N - Le caractère (N+1)ème n'a pas été reçu pendant le temps d'arrêt spécifié auparavant. Les caractères N ont bien été reçus.
- 3\*256+N - Le Nème caractère reçu a été une coupure de ligne
- 4\*256+N - Le Nème caractère a été lu avec une erreur de synchronisation.
- 5\*256+N - Le Nème caractère a été lu avec un embouteillage
- 6\*256+N - Le Nème caractère a été lu avec une erreur de parité.

S'il existe plus d'une erreur de matériel (coupure, synchronisation, embouteillage et parité), seule l'erreur portant le plus petit nombre, sera reportée. Tous les bits d'état d'erreurs sont remis à zéro.

### | OUTBLOCK

| OUTBLOCK , état , chaîne de sortie

---

COMMANDE: Envoie une chaîne à la RS232C. Si le contrôle du déroulement du matériel est validé, le caractère ne sera envoyé que si le signal CTS du plot 5 est vrai. A l'allumage ou après avoir appelé `!SIO`, le signal RTS du plot 4 est activé jusqu'à l'appel de `!CLOSESIO`.

L'état renvoyé est:

- 0\*256+0 - La chaîne a bien été envoyée.
- 1\*256+0 - Aucune chaîne de sortie ou longueur zéro n'a été spécifiée.
- 2\*256+N - Arrêt, un des caractères de la chaîne n'a pas pu être envoyé pendant le temps d'arrêt spécifié, auparavant.  
Les caractères N de la chaîne restent à envoyer.

## Commandes pour les transferts de dossiers 'non-intelligents'

### **| SETFILEEND**

`| SETFILEEND`, caractère

COMMANDE: Spécifie un caractère comme fin de marquage de dossier. La valeur est prise du MOD 256 (par défaut, 27/[CTRL]Z).

### **| INFILE**

`| INFILE`, nom du dossier

COMMANDE: Reçoit un dossier ASCII et l'écrit sur la cassette ou disquette. Cette opération s'achève quand la fin du marquage du dossier est lu. Il n'y a pas d'arrêt. Les messages d'erreurs sont envoyés à l'écran si une erreur de classement se produit.

### **| OUTFILE**

`| OUTFILE`, nom du dossier

COMMANDE: Envoie un dossier ASCII (plus un caractère de marquage de fin de dossier) de la cassette ou disquette. Si le caractère de marquage de fin de dossier existe, il sera automatiquement envoyé, suivi du reste du dossier. Les messages d'erreurs sont envoyés à l'écran si une erreur de classement se produit.

---

## Commandes pour les transferts de dossiers 'intelligents'

### **I BLOW**

**I BLOW**, nom du dossier.

COMMANDE: Envoie un dossier ASCII de la cassette ou disquette, en utilisant un protocole de transfert spécial, qui assure la synchronisation avec le programme de réception et permet la détection d'erreurs. Le message d'erreur 'transfer aborted' (transfert interrompu) est envoyé à l'écran si le programme de réception annonce une erreur irréparable.

### **I SUCK**

**I SUCK**, nom du dossier.

COMMANDE: Reçoit un dossier ASCII et l'inscrit sur la cassette ou disquette, à l'aide d'un protocole de transfert spécial qui assure la synchronisation avec le programme d'envoi et permet la détection d'erreurs. Un message d'erreur approprié sera envoyé à l'écran si le programme de réception détecte une erreur irréparable.

Souvenez vous que les dossiers CP/M sont considérés comme des dossiers ASCII et peuvent donc être transférés à l'aide de ces commandes. Le BASIC et divers langages de programmation devraient être sauvegardés en ASCII avant d'être transférés.

Le protocole pour **I BLOW** et **I SUCK**, est:

Transmission (**I BLOW**):

- (1) Envoyer STX, attendre ACK
- (2) Envoyer un nom de dossier de 16 octets, un nombre de bloc de 2 octets, une longueur de bloc d'un octet (de 0 à 128), des données (de 0 à 128 octets), une somme de contrôle de 2 octets (la somme de tous les octets de données). Une longueur de bloc zéro indique une fin de dossier.
- (3) Attendre ETX, ACK ou NAK.
- (4) Si ETX est présent, abandonnez, si NAK est présent, allez à (2) et essayez à nouveau le même bloc, si ACK est présent, allez à (2) et envoyez le bloc suivant, ou terminez si ce bloc est le dernier.



---

### Réception (I S U C K):

- (1) Attendez STX, répondez avec ACK. NOTEZ: Cela veut dire que le programme de réception devrait commencer avant le programme de transmission, sinon le premier STX risque d'être oublié.
- (2) Réception du nom du dossier, du numéro de bloc, de la longueur de bloc, des données et de la somme de contrôle.
- (3) Cherchez le même nom de dossier que dans le bloc 1, et les numéros de blocs suivants, s'il y a une erreur, envoyez ETX et abandonnez.
- (4) Recherchez les erreurs de matériels ou de somme de contrôle; si tout est OK, envoyez ACK, sinon envoyez NAK.
- (5) Vérifiez la longueur du bloc, si c'est zéro finissez, autrement allez à (2).

(Pour votre information):

STX = [CTRL]B = ASCII 2  
ETX = [CTRL]C = ASCII 3  
ACK = [CTRL]F = ASCII 6  
NAK = [CTRL]U = ASCII 21

## Commandes pour réacheminer les données de sortie de l'imprimante

### I SERIAL

#### I SERIAL

COMMANDE: Envoie toutes les sorties du canal #8 à la RS232C au lieu de les envoyer au port de l'imprimante parallèle Centronics.

La RS232C, par défaut, répondra à un échange de données de logiciel XON/XOFF ainsi qu'à un échange de données de matériel.

Cela veut dire que si le dispositif connecté à la RS232C lui envoie un caractère XOFF, elle cessera de transmettre jusqu'à ce qu'elle reçoive un XON.

(Pour votre information):

XON = [CTRL]Q = ASCII 17  
XOFF = [CTRL]S = ASCII 19

---

## **| PARALLEL**

### **| PARALLEL**

COMMANDE: Ramène la sortie du canal 8 au port de l'imprimante parallèle Centronics.

## **| NOXON**

### **| NOXON**

COMMANDE: Annule les échange de données XON/XOFF du logiciel; la RS232C ignore les caractères qu'elle reçoit.

## **| XON**

### **| XON**

COMMANDE: Ramène l'échange de données XON/XOFF.

## **Commandes pour l'émulation d'un terminal VDU**

### **| TERMINAL**

#### **| TERMINAL**

COMMANDE: Connecte le clavier de l'ordinateur à la sortie de la RS232C, et l'entrée de la RS232C à l'écran. Le mode émulation du terminal VDU peut s'achever en pressant une touche ASCII & FC. Pour reprendre, utilisez **[CTRL][ESC]** (ou juste **[ESC]**, à moins que la valeur décimale 27 lui est été assigné).

Il n'existe pas d'écho local par défaut, de ce qui est tapé. Cet écho provient normalement du matériel connecté à la RS232C.

Les codes du clavier devraient être définis à l'aide de la commande **KEY DEF**, s'il est nécessaire de transmettre des codes n'étant pas décrits dans votre manuel de l'utilisateur.

L'écran obéira aux codes de fonctions par défaut qui lui sont envoyés, comme décrit dans votre manuel de l'utilisateur.

---

## | HALFDUPLEX

### | HALFDUPLEX

COMMANDE: Produit une connexion locale entre le clavier et l'écran en mode émulation, pour que vous puissiez voir ce qui est entrain d'être tapé (au cas ou le matériel connecté à la RS232C ne renvoie pas en écho vers le clavier).

## | FULLDUPLEX

### | FULLDUPLEX

COMMANDE: Coupe la connexion entre le clavier et l'écran qui est produit par |HALFDUPLEX.

## | CTRLDISPLAY

### | CTRLDISPLAY

COMMANDE: L'écran affiche les codes de fonctions (au lieu de les obéir) et reçoit en étant en mode émulation de terminal. Ceci est parfois utilisé pour les diagnostics. Vous pourrez identifier les symboles des codes de fonction affichés, à l'aide du programme ci-dessous:

```
10 MODE 1
20 FOR N=0 TO 30 STEP 3
30 FOR M=0 TO 2
40 IF N+M<32 THEN PRINT CHR$(1);CHR$(N+M);" = [CTRL]";
   CHR$(64+N+M),
50 NEXT M
60 PRINT
70 NEXT N
```

## | CTRLACTION

### | CTRLACTION

COMMANDE: Ramène le contrôle d'obéissance par défaut, des codes de fonction en mode émulation (au lieu de les afficher).

---

## Commandes pour l'émulation PRESTEL

Le texte suivant est réservé à ceux qui habitent l'Angleterre, mais il vous est possible de l'utiliser si vous arrivez à obtenir un des numéros serveurs (et si le prix des communications ne vous effraie pas)

### I PRESTEL

I PRESTEL, (caractère)

COMMANDE: Invoque un émulateur de PRESTEL. Cette émulateur affiche 24 lignes de 40 caractères en 16 couleurs (8 fixes, 8 intermittentes) et utilise des caractères spéciaux et des graphiques en Mode 0. L'émulateur obéit à tous les codes, y compris le code pour caractères double hauteur, le code graphique et le code d'activité du curseur. Ce manuel ne décrit pas l'entière possibilité de PRESTEL.

Il est important de vérifier que la vitesse de transmission, les données, et les bits de synchronisation sont correctes, ainsi que le mode de l'écran, avant de faire appel au service PRESTEL. Ceci est généralement 75TX, 1200RX, 7 bits de données, parité impaire, 1 bit d'arrêt, et le mode de l'écran est 0:

ISETS10,75,1200,1,7,1,0: MODE 0

Si le paramètre (caractère) facultatif est indiqué, il sera envoyé à l'émulateur de PRESTEL, et le contrôle sera renvoyé à BASIC (il sera alors nécessaire de faire appel à la commande (nombre de lignes), I REFRESH, afin de faire une mise à jour de l'écran après lui avoir envoyé le nouveau caractère).

Si le paramètre (caractère) n'est pas indiqué, les caractères seront lus, de la RS232C jusqu'à ce qu'une touche renvoyant ASCII & FC soit pressée. [CTRL][ESC] devrait permettre de renvoyer cette valeur.

Avec le mode continu, l'émulateur de PRESTEL reçoit des caractères et les inscrit simultanément dans un buffer intérieur. L'affichage de l'écran contenu dans ce buffer est, périodiquement, mis à jour. Le curseur est affiché à sa position habituel, sur le PRESTEL et les frappes seront transmises immédiatement.

Dans le mode continu, la touche 'point' du pavé numérique remplace temporairement, le caractère '\*', et la touche [ENTER] voisine du pavé, remplace le caractère de soulignement '\_' qui est l'enter de PRESTEL.

---

## | SAVEPRESTEL

| SAVEPRESTEL, <nom du dossier>

COMMANDE: Mémorise le contenu du buffer intérieur de l'émulateur de PRESTEL, sur une cassette ou disquette.

## | LOADPRESTEL

| LOADPRESTEL, <nom du dossier>

COMMANDE: Charge le buffer intérieur de l'émulateur de PRESTEL, d'un fichier de cassette ou disquette. Vous devrez faire appel à la commande | REFRESH pour faire une mise à jour de l'écran du buffer venant d'être chargé.

## | CURSOR

| CURSOR, <numéro de la colonne>, <numéro de la ligne>

COMMANDE: Ignore le curseur actuel, mis en position par l'émulateur de PRESTEL, si le <numéro de colonne> est un entier entre 1 et 40, et si le <numéro de ligne> est un entier entre 1 et 24.

## | REFRESH

| REFRESH(, <numéro de ligne>)

COMMANDE: Le buffer intérieur de l'émulateur de PRESTEL fait une mise à jour de l'écran. Il est aussi possible de faire une mise à jour d'une ligne seulement (entre 1 et 24). Les couleurs sont réglées de façon à être en accordance avec PRESTEL.

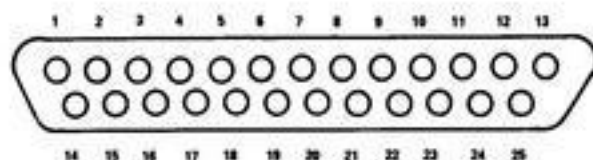
Parce que les effets spéciaux des caractères, couleur, hauteur, etc.. sont produit de gauche à droite et par ligne, il n'est pas recommandé de faire une mise à jour de moins d'une ligne entière.

## Appendice 4

# Le Materiel

---

Voici le détail des plots de la RS232C:



Plot 1 pas utilisé	Plot 14 pas utilisé
Plot 2 données transmises (TXD)	Plot 15 pas utilisé
Plot 3 données reçues (RXD)	Plot 16 pas utilisé
Plot 4 sortie RTS	Plot 17 pas utilisé
Plot 5 entrée CTS	Plot 18 pas utilisé
Plot 6 pas utilisé	Plot 19 pas utilisé
Plot 7 GND	Plot 20 sortie DTR
Plot 8 sortie DCD	Plot 21 pas utilisé
Plot 9 pas utilisé	Plot 22 sortie indic. sonnerie
Plot 10 pas utilisé	Plot 23 pas utilisé
Plot 11 pas utilisé	Plot 24 pas utilisé
Plot 12 pas utilisé	Plot 25 pas utilisé
Plot 13 pas utilisé	

La RS232C utilise le Canal A de la spécification de l'interface série, recommandée par AMSOFT. Les ports E/S utilisés sont les suivants:

Adresse	Sortie	Entrée
&FADC	Données DART	Données DART
&FADD	Contrôle DART	Contrôle DART
&FADE	*NE PAS UTILISER*	*NE PAS UTILISER*
&FADF	*NE PAS UTILISER*	*NE PAS UTILISER*
&FBDC	8253 compteur de chargement 0	8253 compteur de lecture 0
&FBDD	8253 compteur de chargement 1	8253 compteur de lecture 1
&FBDE	*NE PAS UTILISER*	*NE PAS UTILISER*
&FBDF	8253 mot en mode ecriture	*PAS UTILISE*

NOTEZ: Le logiciel ROM est le ROM numéro 6.