

MANUEL DEVELOPPEUR
TELESTRAT

Par Fabrice BROCHE

Version Avril 1987

© ORIC INTERNATIONAL

INTRODUCTION

Ceci n'est que l'ébauche d'un dossier développeur complet sur le TELESTRAT et les logiciels qui l'accompagnent. Il devrait permettre de commencer à travailler en fournissant un minimum d'information. Un dossier exhaustif est en cours de réalisation.

RECOMMANDATIONS

- Il serait préférable d'utiliser au maximum les appels systèmes et les variables standards, et donc aussi les mnémoniques standards pour assurer la compatibilité avec les versions différentes du logiciel ou du matériel. En particulier, tout appel direct à une routine par un autre moyen que les vecteurs se soldera par une non compatibilité certaine.
- Il est de même recommandé, sauf application spécifique, d'éviter de travailler directement avec la mémoire écran ou avec la prise minitel, les extensions 80 colonnes, modem, clavier etc ... rendraient le logiciel inutilisable.
- Enfin, il est préférable d'utiliser au maximum des variables telles que SCRDX etc ... pour des données qui ne sont pas forcément constantes comme la taille écran, la largeur d'impression.
- Chaque fois que des données sont paramétrées, il est recommandé d'en faire usage.

REMARQUE

- Le TELESTRAT se présente au niveau Hardware comme un noyau identique à l'atmos (vidéo, clavier, imprimante etc...) pourvu de capacités supplémentaires (une ACIA, un VIA et un contrôleur de disque intégrés, entre autres).
- Les développeurs qui ne seraient pas familiarisés avec l'oric-1 ou l'atmos trouveront tous les renseignements concernant la gestion de l'écran, du clavier etc... dans le livre l'ORIC A NU des éditions Soracom (disponible chez ORIC International), en attendant bien sur le dossier développeur complet.

Fabrice BROCHE

24 Avril 1987

CARTE MEMOIRE TELESTRAT

0000 005F	variables systèmes TELEMOM
0060 008B	variables systèmes TELEMOM, usages spécifiques (flottant ...)
008C 00CF	variables systèmes HYPER-BASIC ou langage
00D0 00FF	variables systèmes TELEMATIC ou application
*	
0100 01FF	pile 6502
*	
0200 02FF	variables systèmes moniteur
*	
0300 032F	E/S internes
0330 037F	E/S extensions ORIC
0380 03FF	E/S utilisateur
*	
0400 04FF	routines moniteur
0500 055F	variables systèmes STRATSED
0560 058F	variables systèmes HYPER-BASIC ou langage
0590 05FF	buffer éditeur de texte
*	
0600 07FF	buffers, piles... pour langage ou application
*	
0800 97FF	(HIRES)
0800 B3FF	(TEXT) zone utilisateur
9800 9FFF	(HIRES) caractères
A000 BFDF	(HIRES) écran
B400 BB7F	(TEXT) caractères
BB80 BFDF	(TEXT) écran
*	
C000 FFFF	banque 0 RAM interne
C000 FFFF	banque 4-7 port cartouche gauche (reset sur le 7)
C000 FFFF	banque 1-4 port cartouche droit

- Les banques peuvent être occupées indifféremment par n'importe quoi (RAM, ROM ...) sauf la banque 7 (qui contient le TELEMOM), la banque 0 (qui contient le STRATSED) et la banque 3, car la TELEMATIC, lorsqu'elle est utilisée, doit impérativement aller se placer en banque 3.
- Les langages et applications doivent être conçus pour fonctionner sur n'importe quelle banque, ce qui est le cas de TELE-FORTH, HYPER-BASIC, TELE-ASS entre autres. Il suffit pour ce faire de lire les numéros de banque dans V2DRA, les 3 bits de poids faible.

REPARTITION DES BANQUES MEMOIRES

Cartouche HYPER-BASIC :

banque 7 : TELEMONT

banque 6 : HYPER-BASIC

Cartouche TELE-FORTH :

banque 7 : TELEMONT

banque 6 : TELE-FORTH

Cartouche TELEMATIC :

banque 3 : TELEMATIC

Cartouche STRATORIC :

banque 7 : SEDORIC+BOOT

banque 6 : ORIC BASIC V1.1

banque 5 : ORIC BASIC V1.0

Cartouche TELE-ASS droite :

banque 3 : TELEMATIC

banque 2 : TELE-ASS

Cartouche TELE-ASS gauche :

banque 7 : TELEMONT

banque 6 : HYPER-BASIC

banque 5 : TELE-ASS

Cartouche RAM 64 Ko

banque 4 : RAM

banque 3 : RAM

banque 2 : RAM

banque 1 : RAM

STRUCTURE D'UNE BANQUE MEMOIRE

Les 8 octets (FFF8 a FFFF) de chaque banque occupée contiennent un entête qui permet au TELEMOM de reconnaître, et / ou de lancer un logiciel qui y est placé.

FFF8-9	adresse du message copyright (si b7 de FFF8) il doit être terminé par un Ø et peut avoir une longueur quelconque.
FFFA	numéro de la version du logiciel, en BCD version 1.4 --> \$14
FFFB	octet de status de la banque mémoire b7 1 si il y a un copyright a afficher (adresse en FFF8) b6 1 si le programme doit se lancer automatiquement (adresse en FFFC-FFFD) b5 1 si ROM, Ø si RAM b4 1 si il faut ignorer la banque bØ-b3 nombre de Ko-1 exemple: 7 pour 8Ko, 15 pour 16 Ko
FFFC-D	adresse d'exécution (si b6 de FFFB)
FFFE-F	adresse de traitement des IRQ sauf traitement particulier, doit contenir Ø2FA

NB : L'interruption non masquable NMI n'est pas cablée l'action sur le poussoir RESET passe automatiquement le TELESTRAT sur la banque 7, celle du moniteur, qui a donc une structure un peu spéciale.

LES BUFFERS INTEGRES

Le TELEMOM gère quatre ressources par l'intermédiaire de buffer (pile FIFO, premier entré, premier sorti) : le clavier, la sortie centronics, la liaison série entrée et la liaison série sortie.

D'autres buffers peuvent être utilisés par le programmeur.

A chaque buffer est associée une définition de 12 octets. Les différents paramètres des buffers sont modifiables par les vecteurs systèmes ou directement (à proscrire !).

Les buffers sont repérés par un index, de 12 en 12 :

- 00 pour le buffer clavier
- 12 pour l'ACIA entrée
- 24 pour l'ACIA sortie
- 36 pour l'imprimante
- 48, 60 etc... pour les buffers utilisateurs (maximum 6)

L'adresse de base des définitions est \$C080 (banque 0)

Signification des 12 octets de la définition :

00-01	adresse de début du buffer
02-03	adresse de fin (exclue)
04-05	pointeur d'écriture
06-07	pointeur de lecture
08-09	longueur utilisée
10-11	longueur du buffer

Ex : pour le buffer ACIA entrée adresse de base = \$C080+12 = \$C08C

\$C08C	adresse de début
\$C08E	adresse de fin
\$C090	pointeur écriture
\$C092	pointeur de lecture
\$C094	longueur utilisée
\$C096	longueur du buffer

UTILISATION DE LA RS 232

BROCHAGE :	NOM	No	
	TX	2	émission des données
	RX	3	réception des données
	CTS	5	contrôle de l'émission
	DTR	2Ø	ne pas utiliser, utiliser DCD contrôle Telestrat à Ø à la mise sous tension à 1 après un SPRINT si mode imprimante RS 232 à Ø si pas d'impression à 1 pendant impression revient à Ø, 2s (96ØØ bds) après l'émission du dernier octet
	DCD	8	NB: DTR n'est pas géré en réception contrôle émission à Ø, l'émission est stoppée à 1, l'émission est autorisée
	RTS	4	idem à DTR mode imprimante RS 232 attention, à Ø pendant accès disques.
	GND	7	référence signaux

ATTENTION :

Le Telestrat ne peut pas signaler en réception que son buffer est plein. C'est à l'émetteur de s'adapter à sa vitesse, si possible. Sinon, ne pas hésiter à agrandir le buffer de réception série.

Le Telestrat ne reçoit pas correctement à 192ØØ bauds, sauf si on inhibe toutes les interruptions autres que l'ACIA.(POKE #3ØE,127:POKE #32E,127)

CABLAGE MINIMUM :

	TX	2	s'il s'agit d'émettre
ou	RX	3	s'il s'agit de recevoir
	GND	7	

CABLAGE IMPRIMANTE RS 232 :

TELESTRAT			IMPRIMANTE	
TX	2	->	RX	3
DCD	8	<-	DTR	2Ø
GND	7	--	GND	7

RAJOUTER UNE ENTREE/SORTIE

- Le telestrat gère en standard un certain nombre d'E/S, repérées par leur numéro.
- Pour en rajouter de nouvelles, il suffit de savoir que les entrées ont des numéros de #80 à #87 les sorties des numéros supérieurs à #88 (0 à 7 et supérieurs à 8 pour le BASIC).
- Ensuite, il faut écrire une routine qui se conforme au format standard et enfin placer l'adresse de la routine dans la table des vecteurs.

FORMAT D'UNE ROUTINE D'E/S

- Il y a une seule routine pour les 3 opérations : lecture/écriture, ouverture, fermeture, repérées par les flags à l'entrée de la routine :

N=Ø	C=x	écriture de A ou lecture dans A
N=1	C=Ø	ouvrir le canal
N=1	C=1	fermer le canal

- les entrées doivent retourner un flag dans C : C=1 si pas de caractère à fournir, C=0 si la routine retourne un caractère. Dans ce dernier cas, A contient le code de l'octet fourni. (en effet, les entrées sont consultées sans qu'elles ait nécessairement de caractères à fournir).
- les sorties reçoivent le caractère à traiter dans A.
- il est inutile de sauvegarder X, Y et A
- utiliser de préférence USx
- ne jamais utiliser de commandes BRK dans une définition d'E/S, plantage immédiat.

TABLE DES ADRESSES DES VECTEURS

2BE:KBD	2C6:MIE	2CE:SCR	2D6:MIS	2DE:RSS	2E6:US3
2CØ	2C8:USA	2DØ:SC1	2D8	2EØ:VDT	2E8:US4
2C2:MDE	2CA:US2	2D2:SC2	2DA:LPR	2E2	2EA:US5
2C4:RSE	2CC:US3	2D4:SC3	2DC:MDE	2E4	2EC:US6

EXEMPLE :

- pour créer une entrée utilisée sous le numéro #86 (6 en BASIC) la routine commencera par:

US1	BPL LIRE	si N=0, il s'agit de lire un caractère
	BCC OUVRE	si N=1 et C=0, il faut ouvrir l'E/S
	BCS FERME	si N=1 et C=1, il faut fermer l'E/S
LIRE	...	test d'un caractère à fournir
	LDA CODE	lire le caractère
	CLC	indiquer caractère valide
	RTS	
	SEC	indiquer pas de caractère à fournir
	RTS	

- et il faut mettre l'adresse US1 en #2CA.

VECTEURS SYSTEMES

La banque 7 du TELESTRAT est toujours occupée par le TELEMOM, qui est en fait en ensemble de routines gérant la plupart des ressources du TELESTRAT.

Pour utiliser ces routines, la solution de l'instruction du BRK suivie d'un paramètre a été retenue, en raison de son indépendance vis à vis de la gestion des banques, vis à vis des versions successives du logiciel, et enfin de sa compacité, 2 octets suffisant à appeler une routine système contre 3 pour un JSR ou un JMP.

La liste exhaustive des routines est donnée ci-après, avec les différents paramètres à passer.

Tous les vecteurs et les variables systèmes utilisés seront directement compris par l'assembleur TELE-ASS. Pour les autres assembleurs, il faudra initialiser une table d'équivalence.

Pour faciliter la rédaction des programmes, la plupart des assembleurs sur TELESTRAT utilisent l'adressage 'PAGE ZERO' pour l'instruction BRK, le numéro de la routine correspondant à l'argument.

Il est très vivement conseillé d'utiliser les noms donnés par le constructeur, pour faciliter la lisibilité des programmes.

EXEMPLE:

1000	A9 5Ø	LDA # "P"	charger dans A le code ascii de P
1002	ØØ 1Ø	BRK XWRØ	et l'envoyer sur le canal Ø

Dans ce chapitre, les vecteurs sont regroupés par thème, le numéro d'ordre des vecteurs ne suivant pas toujours cette logique !

- 1- La gestion des canaux : ouverture, fermeture, lecture, écriture ...
- 2- La gestion de la mémoire : décalages, remplissages ...
- 3- Les modes d'écrans : texte, hires, jeux de caractères ...
- 4- Les calculs : additions, multiplications, divisions ...
- 5- Les sorties SCR, SC1, SC2, SC3 : fenêtres, codes gérés ...
- 6- La sortie LPR : sauts de lignes, test de présence ...

GESTION DES CANAUX

Pour travailler sur les entrées-sorties, le TELEMOM utilise de manière banalisée quatre canaux. Un canal peut contenir au maximum 4 E/S. La liste des E/S est donnée ci-après.

Les opérations sont donc de deux types :

- ouvrir ou fermer une E/S dans un canal
- lire ou écrire dans le canal

A la mise sous tension, le TELEMOM initialise le canal Ø (le plus usité) avec deux E/S :

- le clavier XKBD qui va servir à saisir les données on s'en doute
- l'écran, fenêtre Ø XSCR pour les affichages.

Lors d'une opération de lecture, le TELEMOM demande successivement à toutes les entrées ouvertes si elles peuvent fournir un octet. Si oui, il est retourné à l'utilisateur, sinon, il passe à la suivante. Dans le cas qui nous préoccupe, seul XKBD est une entrée.

Lors d'une opération d'écriture, le TELEMOM envoie l'octet à émettre sur toutes les sorties ouvertes. Dans notre cas, il s'agira seulement de XSCR.

Supposons que nous voulions utiliser simultanément un clavier interfacé RS 232. Il suffit d'ouvrir sur le canal Ø l'E/S RS 232 entrée, XRSE. Les opérations de lecture feront référence à XKBD puis XRSE.

De même, pour sortir simultanément à l'écran et à l'imprimante, il suffit d'ouvrir l'E/S XLPR et le tour est joué. A ce stade, le canal Ø ne peut plus recevoir d'E/S, puisqu'il est occupé par 4 E/S : XKBD, XSCR, XRSE, XLPR. Bien sûr, on peut à tout instant fermer une des E/S pour récupérer de la place dans le canal.

L'avantage de la gestion des canaux est que le programme peut être transparent vis à vis de l'E/S utilisée. Par exemple, dans la cartouche TELEMATIC en mode serveur, le canal Ø sert toujours au commentaires et le canal 1 à la gestion de la communication, les canaux étant ouverts préalablement pour gérer correctement le mode serveur ou le mode borne etc...

Voici la liste des entrées sorties disponibles sur le Téléstrat. Les numéros sont les mêmes que sous HYPER BASIC, avec un offset de 128.

Les entrées ont forcément des numéros inférieurs à #88.

Les entrées-sorties référencées MIDI IN et MIDI OUT ne sont pas gérées par le TELEMOM 2.x, mais sont réservées pour les extensions midis.

Les utilisateurs peuvent utiliser à leur propres fins les E/S référencées "utilisateur".

UTILISATION DU TELEMON : VECTEURS SYSTEMES

XKBD	numéro #80	CLAVIER
	numéro #81	réservé constructeur
XMDE	numéro #82	MINITEL ENTREE
XRSE	numéro #83	RS232 ENTREE
XMIE	numéro #84	MIDI IN
XUS1	numéro #86	utilisateur entrée
XUS2	numéro #87	utilisateur entrée
XSCR	numéro #88	ECRAN, fenetre 0
XSC1	numéro #89	" 1
XSC2	numéro #8A	" 2
XSC3	numéro #8B	" 3
XMIS	numéro #8C	MIDI OUT
	numéro #8D	réservé constructeur
XLPR	numéro #8E	SORTIE CENTRONICS
XMDS	numéro #8F	MINITEL SORTIE
XRSS	numéro #90	RS 232 SORTIE
XVDT	numéro #91	VIDEOTEK
	numéro #92	réservé constructeur
	numéro #93	réservé constructeur
XUS3	numéro #94	utilisateur sortie
XUS4	numéro #95	utilisateur sortie
XUS6	numéro #96	utilisateur sortie
XUS7	numéro #97	utilisateur sortie

OUVRIR UNE VOIE E/S SUR UN CANAL

Ces quatre routines permettent d'ouvrir l'E/S A sur le canal considéré.

Si on tente d'ouvrir deux fois la même entrée-sortie sur un canal, ou si le canal contient déjà quatre E/S valides, la commande est simplement ignorée.

XOP0	numéro #00
XOP1	numéro #01
XOP2	numéro #02
XOP3	numéro #03

EXEMPLE: ouvrir l'imprimante sur le canal 1

```
1000  A9 8E          LDA #XLPR
1002  00 01          BRK XOP1          ouvrir l'imprimante sur le canal 1
```

FERMER UNE E/S SUR UN CANAL

Ces quatre routines permettent de fermer l'entrée sortie A sur le canal considéré.

Si on tente de fermer une E/S qui n'est pas ouverte dans le canal, la commande est simplement ignorée.

XCL0	numéro #04
XCL1	numéro #05
XCL2	numéro #06
XCL3	numéro #07

TESTER LA PRESENCE D'UN CARACTERE SUR UN CANAL

Ces quatre routines permettent de tester la présence d'un caractère sur l'un des quatre canaux.

- au retour, C=1 si aucun caractère n'est présent
- C=0 si un caractère est présent, son code ASCII est alors dans A
- dans tous les cas, X et Y sont inchangés.

```

XRD0    numéro #08
XRD1    numéro #09
XRD2    numéro #0A
XRD3    numéro #0B
    
```

EXEMPLE:

```

1000    00 08          BRK XRD0
1002                                BCS $1000    si pas de caractère, attendre
1004            02          STA RESB          sauver le caractère reçu
1006    60              RTS
    
```

ATTENDRE UN CARACTERE SUR UN CANAL

Ces quatre routines permettent d'attendre qu'un caractère se présente sur un canal.

- au retour, C=0 et A contient le code ASCII.
- X et Y sont inchangés.

```

XRDW0   numéro #0C
XRDW1   numéro #0D
XRDW2   numéro #0E
XRDW3   numéro #0F
    
```

EXEMPLE:

```

1000    00 0C          BRK XRDW1    attendre un caractère sur le canal 1
1002            02          STA RESB    et le sauver en RESB
1004    60              RTS
    
```

ECRIRE UN CARACTERE SUR UN CANAL

Ces quatre routines permettent d'envoyer un caractère dont le code est contenu dans A sur le canal désigné. Il sera envoyé sur chaque sortie ouverte dans le canal.

- en sortie, X et Y sont inchangés.

```

XWR0    numéro #10
XWR1    numéro #11
XWR2    numéro #12
XWR3    numéro #13
    
```

EXEMPLE:

```

1000    A9 42          LDA #"B"        mettre dans A le code ASCII de B
1002    00 10          BRK XWR0        et l'envoyer sur le canal 0 (écran en standard)
    
```

ECRIRE UNE CHAINE SUR UN CANAL

Ces quatre routines permettent d'envoyer sur un canal une suite d'octets, une "chaîne".
 La chaîne doit être placée à l'adresse AY, et doit se terminer par un zéro. Elle peut être d'une longueur quelconque, même supérieure à 256.
 Ces routines se ramènent en fait à une boucle de XWRx.

XWSTR0 numéro #14
 XWSTR1 numéro #15
 XWSTR2 numéro #16
 XWSTR3 numéro #17

EXEMPLE:

```
1000  A9 07      LDA #<CHAINE  mettre dans A le poids faible de l'adresse
1002  A5 10      LDY #>CHAINE  et dans Y le poids fort
1004  00 14      BRK XWSTR0  et afficher la chaîne sur le canal 0
1006  60         RTS
1007  42 6F ...  BYT "Bonjour !",0
```

ALLER A LA LIGNE SUR LE CANAL 0

Cette commande permet d'envoyer sur le canal 0 les codes RC (retour chariot, 13) et LF (line feed, 10) sur le canal 0.

X et Y ne sont pas affectés.

Tout à fait équivalent à LDA #10 / BRK XWR0 / LDA #13 / BRK XWR0

XCRLF numéro #25

GESTION DE LA MEMOIRE**DECALER UN BLOC MEMOIRE**

Cette routine permet de décaler un bloc mémoire compris entre l'adresse DECDEB et l'adresse DECFIN vers la nouvelle adresse DECCIB (qui sera la nouvelle adresse basse du bloc).

La routine ajuste le sens de décalage vers le haut ou vers le bas.

Si DECFIN est plus petit que DECDEB, la routine ne fait rien.

Le décalage se fait toujours sur la Banque 7 (TELEMON) au delà de #C000).

Les registres A,X et Y sont inchangés, N et Z sont initialisés selon A.

XDECAL numéro #18

EXEMPLE: Scroller d'une ligne vers le bas l'écran HIRES

```

1000  A9 00      LDA #<$A000
1002                LDY #>$A000
1004                STA DECDEB
1006                STY DECDEB+1
1008  A9 40      LDA #<$BF40
100A                LDY #>$BF40
100C                STA DECFIN
100E                STY DECFIN+1
1010  A9 28      LDA #<$A028
1012                LDY #>$A028
1014                STA DECCIB
1016                STY DECCIB+1
1018  00 18      BRK XDECAL
101A  60          RTS

```

REEMPLIR UNE ZONE MEMOIRE

Cette routine permet de remplir la zone mémoire de RES à YX avec la valeur contenue dans A.

Pour des adresses au-dessus de #C000, elle agit sur le TELEMON, c'est donc dire qu'elle ne fait rien.

L'adresse de fin est exclue, c'est à dire qu'elle ne sera pas affectée par le remplissage.

XFILLM numéro #1C

EXEMPLE: Remplir l'écran texte avec le caractère "A"

```

1000                LDA #<$BBA8
1002                LDY #>$BBA8
1004                STA RES
1006                STY RES+1
1008                LDY #<$BFE0
100A                LDX #>$BFE0
100C                LDA #"A"
100E                BRK XFILLM
1010                RTS

```

CHANGEMENTS DE MODE D'ECRAN

```

XTEXT  numéro #19      passage TEXT
XHIRES  numéro #1A     passage HIRES
XEFFHI  numéro #1B     effacer HIRES
XSCRNE  numéro #39

ZADCHA  numéro #1D     trouver l'adresse d'un caractere
XTSTLP  numéro #1E     tester la presence de l'imprimante
XMINMA  numéro #1F     convertit A en maj. s'il y a lieu XY inchanges
*
*      calcul divers
*
XMUL40  numéro #20     AY = A*40
XMULT   numéro #21     AY * RES --> TR0-1-2-3
XADRES  numéro #22     AY = AY+RES
XDIVIS  numéro #23     division
*      RES/AY --> RES (quotient) RESB (reste)
XNOMFI  numéro #24     (AY), longueur dans X --> BUFNOM
*      tient compte de l'extension par default...
*      --> X=0 si longueur nulle
*      X=1 si ras, C=1 si jokers
*      X=2 si modification lecteur par default
*      X>127 si nom incorrect
XDECAY  numéro #26     nombre decimal a l'adresse AY --> AY, longueur X
*      ex: 258 (ascii) a l'adresse AY --> AY=#102, X=3
*
*      conversions numeriques
*
XBINDX  numéro #28     conversion decimale
*      nombre AY, nbre chiffre-2 dans X, DEFAFF=justif
*      --> nombre stocke a l'adresse TR5
XDECIM  numéro #29     idem a BINDX mais affichage sur canal 0
XHEXA   numéro #2A     conversion A --> YA
XA1AFF  numéro #2B     afficher ACC1 sur canal 0
*
*      macros commandes
*
XMENU   numéro #2C     gerer un menu AY
*      ADMEN = adresse de la table des choix
*      choix separes par des 0, plus 1 a la fin
*      MENDDY numero premiere ligne ecran dans la fenetre
*      X      numero derniere ligne ecran dans la fenetre
*      MENDDX premier caractere dans la ligne
*      MENLX  largeur de la barre en video inverse
*      A      ler choix a afficher
*      Y      No choix ou placer la barre
*      --> X=numero du choix
*      A=mode de sortie (13,27,32,03)

```

UTILISATION DU TELEMON : VECTEURS SYSTEMES

XEDT numéro #2D edition pleine page (travaille sur le canal 0)
 * A=longueur (maxi =110)
 * Y b7=1 si retour au debut de la ligne
 * X nombre de caractere a se deplacer a droite
 * --> A=mode de sortie (13,3,11,10)
 * Y= nombre d'espace bidon au debut de la ligne
 * X= debut reel de la ligne (No ligne saute)
 * RES= numero de ligne
 * BUFEDT= ligne en clair, terminee par un 0
 XINSER numéro #2E insertion de ligne
 * numero dans RES
 * A=longueur utile
 * TR0-1 =adresse du buffer a inserer
 * --> ligne inseree selon SCEDEB,SCEFIN
 * difference dans TR3-4 et AY
 XSCELG numéro #2F recherche de ligne RES.
 * C=1 si trouvee, adresse dans RESB

XEDTIN numéro #32
 XECRPR numéro #33 ecrire le prompt
 XCOSCR numéro #34 effacer le curseur, X=No fenetre
 XCSSCR numéro #35 afficher le curseur, X=No fenetre
 XSCRSE numéro #36 initialiser une fenetre
 XSCROH numéro #37 scroll vers le haut
 XSCROB numéro #38 scroll vers le bas

* travail CLK
 *

XCLCL numéro #3D fermer l'horloge
 XWRCLK numéro #3E afficher l'horloge a l'adresse AY

*
 * travail PSG
 *

XSONPS numéro #40 14 parametres XY
 XEPSG numéro #41 X --> registre No A du 8912
 XOUPS numéro #42 OUPS
 XPLAY numéro #43 PLAY param dans HRS1,2 ...
 XSOUND numéro #44 SOUND
 XMUSIC numéro #45 MUSIC
 XZAP numéro #46 ZAP
 XSHOOT numéro #47 SHOOT

*
 * travail LPR
 *

XLPRBI numéro #48 ecrire directement A sur l'imprimante
 XLPCRL numéro #49 aller a la ligne
 XHCSCR numéro #4A hard copy texte (SCRNB = numero fenetre)
 XHCVDT numéro #4B hard copy videotex texte
 XHCHRS numéro #4C hard copy hires

* travail KBD

UTILISATION DU TELEMON : VECTEURS SYSTEMES

```

*
XALLKB  numéro #50      balayer le clavier --> KBDCOL
XKB DAS  numéro #51      conversion en ASCII
XGOKBD  numéro #52      changer de type de clavier (type dans A)
*          00 = QWERTY          01 = AZERTY
*          02 = FRENCH          03 = bwana
*          04 = ACCENT SET      05 = ACCENT OFF

*
*      travail buffers
*
XECRBU  numéro #54      ecrire A dans buffer X=index (00,12,24,36 ...)
XLISBU  numéro #55      lire A (X=index) --> (C=1 si vide)
XTSTBU  numéro #56      tester si vide (X=index) --> (C=1 si vide)
XVIDBU  numéro #57      vider (X=index)
XINIBU  numéro #58      initialiser taille RES = debut, RESB = fin (exclue)
XDEFBU  numéro #59      rendre valeur par default (X=index)
XBUSY   numéro #5A      tester si au moins un non vide (C=1 si tous vides)

*
*      vecteurs buffers
*
*      ORG H'00'
*
XKBDBU  RESB 12          clavier
XRSEBU  RESB 12          serie entree (ACIA 6551-1)
XRSSBU  RESB 12          serie sortie (ACIA 6551-1)
XLPRBU  RESB 12          parallele
XUS0BU  RESB 12          utilisateur      reserve ORIC
XUS1BU  RESB 12          utilisateur      reserve ORIC
XUS2BU  RESB 12          uti

```

UTILISATION DU TELEMON : VECTEURS SYSTEMES

```

*
*   liaisons series, minitel, gestion ligne telefonique
*
XSDUMP  numéro #5C      dump entree RS 232
XCONSO  numéro #5D      transforme en terminal
XSLOAD  numéro #5E      lire un fichier de RS 232
      C=1, DESALO pour SLOADA
      C=Ø pour SLOAD
XSSAVE  numéro #5F      ecrire un fichier sur RS 232, DESALO,
FISALO ...
XMLOAD  numéro #6Ø      lire un fichier du minitel
      CF SLOAD
XMSAVE  numéro #61      ecrire un fichier sur minitel
XRING   numéro #62      detection sonnerie, --> C=0 si detectee
XWCXFI  numéro #63      attente connexion du correspondant
XLIGNE  numéro #64      prise de ligne (retournement et connexion)
XDECON  numéro #65      deconnexion minitel
XMOUT   numéro #66      A --> minitel
XSOUT   numéro #67      A --> RS232
*
*   travail flottant
*
XA1DEC  numéro #68      ACC1 --> decimal (BUFTRV)
XDECA1  numéro #69      decimal, binaire (%), hexa (#) (adresse AY) --> ACC1
XA1PA2  numéro #6A      A1+A2 --> A1
XA2NA1  numéro #6B      A2-A1 --> A1
XA1MA2  numéro #6C      A1*A2 --> A1
XA2DA1  numéro #6D      A2/A1 --> A1
XA2EA1  numéro #6E      A2iA1 --> A1
XNA1    numéro #6F      -A1 --> A1
*
XSIN    numéro #70      SIN(A1) --> A1
XCOS    numéro #71      COS(A1) --> A1
XTAN    numéro #72      TAN(A1) --> A1
XATN    numéro #73      ATN(A1) --> A1
XEXP    numéro #74      EXP(A1) --> A1
XLN     numéro #75      LN(A1) --> A1
XLOG    numéro #76      LOG(A1) --> A1
XRND    numéro #77      RND(A1) --> A1
XSQR    numéro #78      SQR(A1) --> A1
XRAD    numéro #79      RAD(A1) --> A1
XDEG    numéro #7A      DEG(A1) --> A1
XINT    numéro #7B      INT(A1) --> A1
XPI     numéro #7C      PI --> A1
XRAND   numéro #7D      RAND(A1) --> A1
*
XA1A2   numéro #7E      A1 --> A2
XA2A1   numéro #7F      A2 --> A1
XIYAA1  numéro #80      AY --> A1
XAYA1   numéro #81      (AY) --> A1
XA1IAY  numéro #82      A1 --> AY
XA1XY   numéro #83      A1 --> (XY)
XAA1    numéro #84      AA1 --> A1
XADNXT  numéro #85      (AY)+A1 --> (AY)
XINTEG  numéro #86      1 si A1 entier
*
*   travail hires

```

UTILISATION DU TELEMON : VECTEURS SYSTEMES

*

XHRSSE	numéro #8C	placer en X, Y
XDRAWA	numéro #8D	ADRAW HRS1,HRS2 ... dans l'ordre du BASIC
*		sur 2 octets signes.
XDRAWR	numéro #8E	DRAW le FB code est dans b7-b6 de HSRFB
XCIRCL	numéro #8F	CIRCLE
XCURSE	numéro #90	CURSET
XCURMO	numéro #91	CURMOV
XPAPER	numéro #92	PAPER ù X = No fenetre si TEXT, 128 si HIRES
XINK	numéro #93	INK ù A = valeur (0-7 ou 16-23)
XBOX	numéro #94	BOX
XABOX	numéro #95	ABOX
XFILL	numéro #96	FILL
XCHAR	numéro #97	CHAR
XSCHAR	numéro #98	SCHAR adresse chaine AY, longueur X
XEXPLO	numéro #9C	EXPLODE
XPING	numéro #9D	PING

VARIABLES SYSTEMES TELEMON

```

RES      adresse #00      usage general...
RESB     adresse #02
*
DECDEB   adresse #04      parametres decalage
DECFIN   adresse #06
DECCIB   adresse #08
DECTRV   adresse #0A
*
TR0      adresse #0C      usage general
TR1      adresse #0D
TR2      adresse #0E
TR3      adresse #0F
TR4      adresse #10
TR5      adresse #11
TR6      adresse #12
TR7      adresse #13
*
DEFAFF   adresse #14      valeur par defaut conversion decimale

IRQSVA   adresse #21      sauvegarde des registres en entree d'IRQ
IRQSVX   adresse #22
IRQSVY   adresse #23
*
* variables SCR
*
ADSCR    adresse #26      adresse debut ligne d'ecran
SCRNB    adresse #28      numero de la fenetre d'ecran
*
* variables KBD
*
ADKBD    adresse #2A      adresse de la table de conversion ASCII
*
* variables VDT
*
ADVDT    adresse #2C      adresse ecran hires
ADASC    adresse #2E      adresse table ASCII
ADATR    adresse #30      adresse table des couleurs
VDTPAR   adresse #32      b7      : G0 attribut a valider
*                          b6-b4 : G0-G1 couleur de fond
*                          b2      : G0 souligne
VDTASC   adresse #33      b7      : delimitateur
*                          b6-b0 : G0 code ascii (G2 : 00-1F)
*                          b6      : G1 non disjoint
*                          b5-b0 : G1 motif
VDTATR   adresse #34      b7      : G1
*                          b6      : G0 video inverse
*                          b5      : G0 double largeur
*                          b4      : G0 double hauteur
*                          b3      : G0-G1 clignotement
*                          b2-b0 : G0-G1 couleur de texte

VDTFT    adresse #35      b7: SS2
*                          b6: SEP
VDTX     adresse #38      coordonnee X
VDTY     adresse #39      coordonnee Y
VDTGX    adresse #3A      idem en graphique          (0 - 1)

```

VARIABLES SYSTEMES TELEMON

VDTGY	adresse #3B	idem en graphique	(0 - 2)
FLGVD0	adresse #3C	b7: sequence en cours	
*		b6: ESC	
*		b4: US	
*		b3: REP	
*		b1-b0: nombre de caractere de la sequence	
FLGVD1	adresse #3D	b7: curseur ON	
*		b6: mode graphique	
*		b1: mode trace / efface	
*		b0: N/B	
*	variables horloge		
*			
ADCLK	adresse #40	adresse d'affichage de l'horloge	
TIMEUS	adresse #42	decompteur utilisateur en seconde	
TIMEUD	adresse #44	decompteur utilisateur en dixieme de secondes	
*			
*	variables HIRES		
*			
HRSX	adresse #46	position X	
HRSY	adresse #47	position Y	
HRSX40	adresse #49	X / 6	
HRSX6	adresse #4A	reste de X / 6	
ADHRS	adresse #4B	adresse ecran HIRES	
HRS1	adresse #4D	param 1	
HRS2	adresse #4F	param 2	
HRS3	adresse #51	param 3	
HRS4	adresse #53	param 4	
HRS5	adresse #55	param 5	
HRSFB	adresse #57	b7-b6	
*			
*	variables RS232 C		
*			
RS232T	adresse #59	b0-b3: vitesse	
*		b4 : 0 horloge externe 1 horloge interne	
*		b6-b5: 00 8 bits	
*		01 7 bits	
*		10 6 bits	
*		11 5 bits	
*		b7 : 0 un stop, 1 2 ou 1.5 stops.	
RS232C	adresse #5A	b0-b3 a 0	
*		b4 : 1 si echo	
*		b5 : 1 si parite	
*		b7-b6: 00 parite impaire emission/reception	
*		01 parite paire emission/reception	
*		10 parite emise, reception non testee	
*		11 SPACE emis, reception non testee	
SCEDEB	adresse #5C		
SCEFIN	adresse #5E		

VARIABLES SYSTEMES TELEMON

```

*
* variables a usage particulier
*
VARMNB  adresse #60          (48 octets)
VARLNG  adresse #8C          (68 octets) variables systemes language
VARAPL  adresse #D0          (48 octets) variables systemes application

MEN      adresse #60
MENDDX  adresse #61
MENDDY  adresse #62
MENDFY  adresse #63
MENX    adresse #64
MENLX   adresse #65
MENDY   adresse #66
MENFY   adresse #67
FLGMEN  adresse #68
ADMEN   adresse #69

ACC1E   adresse #60
ACC1M   adresse #61
ACC1S   adresse #65
ACC1EX  adresse #66
ACC1J   adresse #67          doit valoir 0, sinon probleme !!!
*
ACC2E   adresse #68
ACC2M   adresse #69
ACC2S   adresse #6D
ACCPS   adresse #6E
*
ACC3    adresse #6F
*
ACC4E   adresse #73
ACC4M   adresse #74
*
ACC5E   adresse #78
ACC5M   adresse #79
*
FLTR0   adresse #7D
FLTR1   adresse #7E

ACC6    adresse #80
*
FLPOLP  adresse #85
FLPO0   adresse #87
*
FLINT   adresse #88
FLSVS   adresse #89
*
FLSGN   adresse #8A
FLERR   adresse #8B
*
FLDT0   EQU ACC4M
FLDT1   EQU ACC4M+1
FLDT2   EQU ACC4M+2
FLSVY   EQU ACC4M+3

```

VARIABLES SYSTEMES TELEMON

```

* etats des banques memoires
*
BNKST   RESB 8   200   valeur des octets $FFFB de chaque banque
*
* lecteurs connectes
*
TABDRV  adresse #208   activation des lecteurs (0 si non connecte)
*
DRVDEF  adresse #20C   sinon, nb pistes, b7=1 si double face
*
*                               numero (0-3) lecteur par default
*
FLGTEL  adresse #20D   b7: 1, haute resolution
*
*                               b6: 1, mode minitel
*
*                               b5: 1, mode degre 0, calculs en radian
*
*                               b2: 1, BONJOUR.COM existe
*
*                               b1: 1, imprimante CENTRONICS detectee
*
*                               b0: 1, STRATSED absent
KOROM   adresse #20E   Ko de ROM totale
KORAM   adresse #20F   Ko de RAM totale
*
* variables IRQ
*
TIMED   adresse #210   horloge 1/10
TIMES   adresse #211   horloge seconde
TIMEM   adresse #212   horloge minutes
TIMEH   adresse #213   horloge heures
FLGCLK  adresse #214   b7: afficher l'horloge toutes les secondes
*
* variables SCR
*
ADSCRL  adresse #218
ADSCRH  adresse #21C   adresse courante de chaque fenetre
SCRX    adresse #220   coordonnee X
SCRY    adresse #224   coordonnee Y
SCRDX   adresse #228   debut de la fenetre (inclus)
SCRFX   adresse #22C   fin de la fenetre (inclus)
SCRDY   adresse #230   debut de la fenetre (inclus)
SCRFY   adresse #234   fin de la fenetre (inclus)
SCRBAL  adresse #238   poids faible
SCRBAH  adresse #23C   poids fort adresse de base de la fenetre
SCRCT   adresse #240   couleur de texte
SCRCF   adresse #244   couleur de fond
FLGSCR  adresse #248   b7: afficher le curseur
*
*                               b6: curseur fixe
*
*                               b5: video inverse
*
*                               b4: 38/40
*
*                               b3: escape
*
*                               b2: US
*
*                               b1: double hauteur
*
*                               b0: compteur pour US ...
CURSCR  adresse #24C   caractere sous le curseur
LPRVEC  adresse #24E   execution hard copy hires
*
*                               30
*
SCRTXT  RESB 6   256   caracteristiques fenetre texte
SCRHIR  RESB 6   25C   caracteristiques fenetre hires
SCRTRA  RESB 6   262   caracteristiques fenetre trace

```

VARIABLES SYSTEMES TELEMON

```

*
* variables KBD
*
KBDCOL RESB 8 268 image du clavier 8x8
KBDVRR adresse #272 nb avant repetition
KBDVRL adresse #273 diviseur repetition
FLGKBD adresse #275 b7: 1 si majuscule
* b6: 1 si son
* b5: 1 si ESC = iT
* b2-b1: 0 0 si qwerty
* 0 1 si azerty
* 1 0 si french
* 1 1 si bwana
* b0: 1 si gerer touches de fonction
*
KBDFACT adresse #276 adresse de gestion des touches de fonction
KBDSSH adresse #278 b7: 1 si Ctrl
* b6: 1 si Funct
* b5: 1 si joystick gauche
* b4: 1 si joystick droit
* b3: 1 si souris
* b0: 1 si shift
*
KBDKEY adresse #279 valeur ASCII de la touche
KBDCTC adresse #27E b7 : 1 si Ctrl C frappe type: ASL KBDCTC/BCS ...
* b6-b0 : 0

* variables VDT
*
VDTFIL adresse #282 valeur des parametres PROx
*
* variable imprimante
*
LPRX adresse #286 position dans la ligne
LPRY adresse #287 position dans la page → ineffective
LPRFX adresse #288 largeur d'impression
LPRFY adresse #289 longueur d'une page → ineffective
FLGLPR adresse #28A b7: 1 si prete
* b6: 1 si pas gerer CRLF
* b2: 1 si RS 232 au lieu de centronics
* b1: 1 si echo SCR
* b0: 1 si pas LF apres CR
LPRSY adresse #28B position qui doit declencher un saut de page → ineffective
*
* variables joystick/souris
*
FLGJCK adresse #28C b7: 1 si joystick droit Faux
* b6: 1 si joystick gauche
* b0: 1 si souris / gauche joystick gauche
JCGVAL adresse #28D valeur joystick droit (H B FG G D)
JCDVAL adresse #28E valeur joystick gauche dist (H B FG G D FC FD)
JCKTAB adresse #29D valeur par defaut: 0B 0A 20 08 09 0A 03

```

*1 par défaut
0 si défini*

*#28D 0 0 0 H B F G D
#28E 0 FC FD H B F G D*

VARIABLES SYSTEMES TELEMOM

```

*
*      HIRES
*
HRSPAT  adresse #2AA      motif
HRSEER  adresse #2AB      indication d'erreur dans les parametres
*
*      table d'ouverture des canaux
*      contient le numero de l'E/S (b7=1)
*      ou rien (b7=0)
*
IOTAB0  adresse #2AE      activation canal 0
IOTAB1  adresse #2B2      1
IOTAB2  adresse #2B6      2
IOTAB3  adresse #2BA      3
*
*      cette table contient les adresses des routines des drivers d'E/S
*      reperes par XOPx, XCLx et OPCH/CLCH du basic.
*      ces routines sont appelees par les ecritures/lectures dans
*      les canaux.

ADIOB   RESB 48 2BE      adresse de gestion des E/S
FLGRST  RESB 1
*
CSRND   adresse #2EF      valeur courante generateur aleatoire
*
* VECTEURS
*

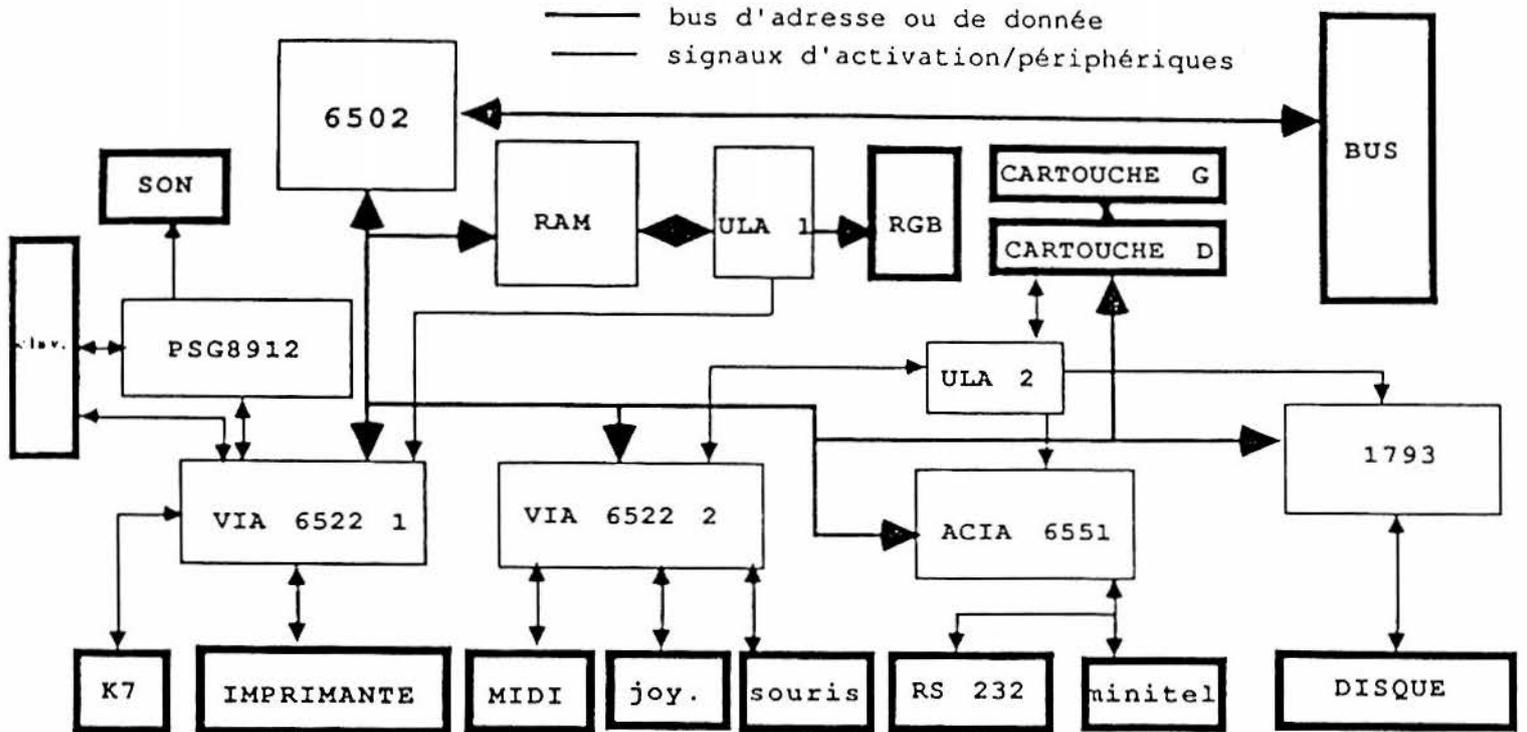
VNMI    adresse #2F4      excution reset a chaud (No banque, adresse)
VIRQ    adresse #2FA      execution IRQ
VAPLIC  adresse #2FD      (No banque, adresse) TELEMATIC --> LANGAGE

EXBNK   adresse #40C      execution d'une routine d'une autre banque
VEXBNK  adresse #414
BNKCIB  adresse #417

BUFTRV  adresse #0100      buffer travail divers
*
DOSVAR  adresse #0500      adresse variables DOS
EXTDEF  adresse #055D      extension par defaut
VARLNB  adresse #0560      variables language
BUFEDT  adresse #0590      buffer editeur de texte
*
PROGRM  adresse #0800      zone programme
*
BARBRE  adresse #4000      buffer arborescence (1000 pages environ)
BUFTXT  adresse #8000      buffer code videotext (3 Ko maxi)
BVDTAS  adresse #9000      buffer attribut
BVDTAT  adresse #9400      buffer couleur
*
VDTDES  adresse #9C00      travail et G2
TABDBH  adresse #9C80
BUFBUF  adresse #C080      definition buffers
BUFROU  adresse #C500      routines gestion buffers

```

SCHEMA SYNOPTIQUE DU TELESTRAT



UTILISATION DU TELEMONT : ADRESSES D'ENTREES/SORTIES

*
* adresses du VIA 6522 1

*
V1DRB adresse #0300'
V1DRA adresse #0301'
V1DDR adresse #0302'
V1DDRA adresse #0303'
V1T1 adresse #0304'
V1T1L adresse #0306'
V1T2 adresse #0308'
V1ACR adresse #030B'
V1PCR adresse #030C'
V1IFR adresse #030D'
V1IER adresse #030E'
V1DRAB adresse #030F'

*
* adresses du FDC 8912

*
FDCCR adresse #0310'
FDCTR adresse #0311'
FDCSR adresse #0312'
FDCDR adresse #0313'

*
* adresses du controleur de disquette

*
CDRIVE adresse #314'
FDCDRQ adresse #318'

*
* adresses de l'ACIA 6551

*
ACIADR adresse #031C' registre de donnees
ACIASR adresse #031D' registre d'etat
ACIACR adresse #031E' registre de commande
ACIACT adresse #031F' registre de controle

UTILISATION DU TELEMOM : ADRESSES D'ENTREES/SORTIES

```

*
* adresses du VIA 6522 2
*
*
*      CA1      in      MIDI      clock IN
*      CB1      in      MINITEL   detection d'appel
*      CA2      out     MINITEL   emission en ligne
*      CB2      out     MIDI      MIDI OUT
*      PA0      out     BANK      !
*      PA1      out     BANK      !      No banque (0-7)
*      PA2      out     BANK      !
*      PA3      in      MIDI      MIDI start/stop
*      PA4      out     RS232/MIN  0: RS232 1: MINITEL
*      PA5      in      SOURIS     bouton droit
*      PA6      in      MIDI      MIDI IN
*      PA7      in      SOURIS     bouton central
*      PB0      in      JOYSTICK   droit
*      PB1      in      JOYSTICK   gauche
*      PB2      in      JOYSTICK   feu ! ou bouton gauche souris
*      PB3      in      JOYSTICK   bas
*      PB4      in      JOYSTICK   haut
*      PB5      out     SWITCH
*      PB6      out     SOURIS     selection port droit
*      PB7      out     JOYSTICK   selection port gauche
*
V2DRB  adresse #0320'
V2DRA  adresse #0321'
V2DDR  adresse #0322'
V2DDRA adresse #0323'
V2T1   adresse #0324'
V2T1L  adresse #0326'
V2T2   adresse #0328'
V2ACR  adresse #032B'
V2PCR  adresse #032C'
V2IFR  adresse #032D'
V2IER  adresse #032E'
V2DRAB adresse #032F'
*
* adresses de l'extension clavier IBM / carte 80 colonnes
*
SGDPAR  adresse #0340'      registre d'adresse
*
SGDPDR  adresse #0341'      registre de donnees
KBDSR   adresse #0342'      b7 (1) IRQ clavier
*
*      b0 (0) presence clavier
KBDDR   adresse #0343'
KBDCR   adresse #0344'      b7 (1) autorisation IRQ clavier
*
*      b0 (1) pas d'incrustation
HTR     adresse #0360'

```

UTILISATION DU STRATSED

Tous les vecteurs du STRATSED étant en RAM, il faut une routine spéciale pour y accéder, dont un modèle est donné ci-dessous.

NB : le numéro du lecteur (0-3) doit être placé dans DRIVE

*
 * routine standard
 * attention, cette routine modifie A, et P au retour
 * ce qui peut gener certaines routines du DOS.
 *
 * en entree, A doit contenir le poids faible de l'adresse du vecteur
 * vise.
 *

```

EXEDOS  STA VEXBNK+1      stocker poids faible adresse
        LDA #H'FF'
        STA VEXBNK+2      et poids fort toujours #FF
        LDA FLGTEL
        LSR A
        BCS pas de stratsed  tester presence stratsed
        LDA #00
        STA BNKCIB         indiquer execution sur banque 0
        STA ERRNB         indiquer pas d'erreur dos pour l'instant
        PHP               sauver I car le dos risque de le modifier
        TSX
        DEX
        DEX
        DEX
        DEX               sauter 2 adresses de retour
        STX SAVES         initialiser pointeur pile en cas d'erreur
        JSR EXBNK         executer la routine
        PLP
        LDA ERRNB
        BNE erreur du dos  tester erreur
        RTS               tout est ok !
  
```

UTILISATION DES FICHIERS DIRECT/SEQUENTIELS

I/ INITIALISATIONS GENERALES

La gestion des fichiers utilise un buffer de travail dont la longueur est $\#300+\#260*n$, n etant le nombre de fichiers ouverts simultanement.

Il faut donc en premier lieu initialiser le nombre de fichier ouverts simultanement (variable NBFIC, 549) et l'adresse du buffer (pointeur TAMPFC, 542).

Ensuite, pour toute operation, FICNUM (548) contiendra le numero logique du fichier.

II/ TRAVAIL EN SEQUENTIEL

La gestion de fichiers sequentiel est simple:

- OUVRIER le fichier (vecteur XOPEN), en initialisant BUFNOM, FICNUM et type=#10 dans FTYPE
s'il n'existe pas, il est cree.
- ECRIRE dans le fichier a la position courante (vecteur XSPUT)
ACC1S contient le type (b7=1 si chaine, 0 si flottant)
 - si chaine ACC1E=longueur
ACC1M=adresse
 - si flottant, nombre dans ACC1
- LIRE dans le fichier a la position courante (vecteur XSTAKE)
ACC1S contient le type, memes convention que pour SPUT
- MODIFIER LE POINTEUR de fichier
 - vecteur XREWIND
 - vecteur XAPPEND
 - vecteur XJUMP, nombre d'enregistrement a sauter dans DESALO
- TESTER LA FIN DU FICHER
vecteur XFST, resultat dans ACC1M (1 si fin, 0 sinon)
- FERMER LE FICHER (vecteur XCLOSE)

III/ TRAVAIL EN ACCES DIRECT

1/ OUVRIR LE FICHIER

- initialiser le vecteur XFIELD (voir routine plus loin)
XFIELD = No banque (0-7)
XFIELD+1 = adresse
- Tester si le fichier existe deja (vecteur XTRVNM)

LE FICHIER EXISTE DEJA

- placer BUFNOM, FICNUM, FTYPE=8
- envoyer le vecteur XOPEN

LE FICHIER DOIT ETRE CREE

- placer BUFNOM, FICNUM, FTYPE=8
- initialiser DESALO, DESALO+1 a 0 (nombre de fiche au debut)
- longueur de la fiche dans FISALO (maxi 1-512 ou 768)
- envoyer le vecteur XOPEN

- il faut creer la structure des champs.
le secteur de structure contient 250 octets utiles, soit
25 definition de champs.

un champ occupe 10 octets :

00 : 0 si numerique #80 si alphanumerique
+ nombre de caractere du nom du champ
(si nom=NOM\$, la valeur est #84)

01 : longueur pour un champ alpha, rien sinon

02-09 : nom du champ, termine par \$ si alpha + espaces.

le dernier champ a son type egal a #FF

exemple, si le fichier contient le champ TOTO\$ de 20 octets et

le champ numerique NOMBRE :

85 14 54 4F 54 4F 24 20 20 20 06 00 4E 4F 4D 42 52 45 20 20 FF

l'ecriture de la structure se fait par:

- placer la table des champs a l'adresse pointee par ADTMP
- il faut aussi sauver la structure dans le 1er secteur du fichier :
mettre RWBUF a la valeur de ADTMP
- la piste est en (ADPS)+12, le secteur est en (ADPS)+13
- sauver par le vecteur XSAY (A=piste, Y=secteur)

2/ ECRIRE UNE FICHE

- numero de fiche dans DESALO, numero logique dans FICNUM
 - vecteur XPUT
- pour chaque champ, passage par XFIELD
- b7 de TD4 = 1 car ecriture
 - type (b7) et longueur du nom du champ dans ACC1S
 - determiner de quel champ il s'agit grace a l'adresse du nom du champ dans ADTMP (1er caractere du nom)
 - placer la valeur, memes convention que SPUT

3/ LIRE UNE FICHE

- numero de fiche dans DESALO, numero logique dans FICNUM
 - vecteur XTAKE
- pour chaque champ, passage par XFIELD
- b7 de TD4 = 0 car lecture
 - type (b7) et longueur du nom du champ dans ACC1S
 - determiner de quel champ il s'agit grace a l'adresse du nom du champ dans ADTMP (1er caractere du nom)
 - lire la valeur, memes convention qu SPUT

4/ SAVOIR LE NOMBRE TOTAL DE FICHE

- numero logique dans FICNUM, vecteur XFST
- DESALO contient le nombre de fiches

5/ FERMER LE FICHER

- numero logique (ou 0) dans FICNUM vecteur XCLOSE

USAGE DES RWTS

RWTS est l'abréviation de Read Write Track Sector, et on regroupe sous ce nom générique le driver d'écriture ou lecture sur disque.

Les RWTS comprennent des ordres du style "lire un secteur" déplacer la tête sur la piste 0 etc...

En fait, ce nombre de commande est limité.

Chaque fois qu'une routine du STRATSED doit effectuer une quelconque opération sur le disque, elle passe par la routine RWTS. pour permettre facilement la création d'autres supports organisés en disques virtuel (notament RAM disques), un vecteur RWTS a été crée en RAM, pour permettre de modifier le comportement de cette routine.

Le vecteur RWTS est placé à l'adresse #054F, où il y a normalement un JMP #D400, #D400 étant l'adresse normale de gestion des RWTS.

Il suffit de remplacer ce JMP par un JMP sur sa propre routine pour jongler avec les RWTS.

Voici les caractéristiques d'une routine RWTS :

- conserver le registre Y
- en sortie, positionner le registre d'erreur ERRFDC (#50E) et positionner comme suit le registre d'état :
 - ne pas toucher à I
 - Z=1 (et ERRFDC=Ø) si pas d'erreur, Ø si erreur déclenchée
 - N=1 (et b7 de ERRFDC =1) si erreur d'E/S
 - V=1 (et b6 de ERRFDC=1) si protection en écriture.
 (si ERRFDC est correctement positionné, la séquence LDA #\$FF / BIT ERRFDC position correctement Z N et V)
- finir par JMP #D400 pour utiliser les RWTS normales. (ou, pour plus de rigueur, il faudrait avoir sauver l'ancienne valeur du vecteur, afin de s'affranchur de STRATSED spéciaux).
- utiliser les variables systèmes
- traiter les ordres (normalement destinés au FDC 1793) suivants :
 - (l'ordre se trouve dans le registre X en arrivant dans les RWTS)
 - #Ø8 retour sur la piste Ø (sur le lecteur DRIVE)
 - #FØ formatage de la piste PISTE sur le lecteur DRIVE, le format de la piste (standard 1793 est à l'adresse RWBUF).
 - #88 lecture du secteur (256 octets) PISTE SECTEU sur le lecteur DRIVE à l'adresse RWBUF.
 - #A8 écriture du secteur (256 octets) PISTE SECTEU sur le lecteur DRIVE à partir de l'adresse RWBUF.
 - toutes les autres commandes du FDC 1793 peuvent être gérées, pour obtenir un maximum de transparence vis à vis des programmes utilisateurs . Toutefois, le STRATSED n'utilise que 08, F0, 88 et A8. Bien entendu, selon le type de disque virtuel utilisé, les ordres 08 ou F0 seront ignorés.

VARIABLES DU STRATSED

BUF1	adresse #C100'	buffer de travail
BUF2	adresse #C200'	! buffers de bitmap de la disquette
BUF2b	adresse #C300'	!
BUF3	adresse #C400'	buffer de catalogue
DRIVE	adresse #500	No lecteur (0-3)
PISTE	adresse #501	No piste (b7=1 si face B)
SECTEU	adresse #502	No secteur
RWBUF	adresse #503	adresse secteur/piste a lire ou ecrire
ERRFDC	adresse #50E	
ERRFLG	adresse #50F	autorisation erreur
ERRVEC	adresse #510	adresse de gestion des erreurs
ERRNB	adresse #512	numero de l'erreur
*		00 pas d'erreur
*		01 fichier inexistant
*		02 I/O
*		03 fichier existant
*		04 plus assez de place
*		05 disquette protegee
*		06 erreur de type du fichier
*		07 format inconnu (pas STRATSED ou SEDORIC)
*		08
*		09
*		10
*		11 fichier deja ouvert
*		12 fichier fermer
*		13 fin de fichier (sequentiel)
*		14 mauvais type d'expression
*		15 mauvais numero de fiche
SAVES	adresse #513	
*		
POSNMP	adresse #514	piste: position d'un nom dans le catalogue
POSNMS	adresse #515	secteur
POSNMX	adresse #516	index
*		
BUFNOM	adresse #517	nom (numero lecteur, 12 caracteres du nom)
VSALO0	adresse #528	flag lecture / ecriture
VSALO1	adresse #529	idem
LOSALO	adresse #52A	
FTYPE	adresse #52C	type du fichier lu / a ecrire
DESALO	adresse #52D	debut du fichier a ecrire
FISALO	adresse #52F	fin du fichier a ecrire
EXSALO	adresse #531	adresse d'execution
LOCRE	adresse #533	
LOCREB	adresse #535	
DECREP	adresse #537	
DECRES	adresse #538	
NBCRE	adresse #539	
VCRE0	adresse #53A	
BUFC0	adresse #53B	

UTILISATION DU TELEMOM : USAGE DU STRATSEED

```

TAMPFC  adresse #542
BITMFC  adresse #544
INDFCB  adresse #546
FICTYP  adresse #547
FICNUM  adresse #548
NBFIC   adresse #549
FICLON  adresse #54A
XFIELD  adresse #54C
RWTS    adresse #54F          vecteur détournement RWTS
*
TD0     EQU HRS1              (#4D)
TD1     EQU HRS1+1
TD2     EQU HRS2
TD3     EQU HRS2+1
TD4     EQU HRS3
TD5     EQU HRS3+1
TD6     EQU HRS4
TD7     EQU HRS4+1
INDIC0  EQU HRS5
INDIC1  EQU HRSPAB
INDIC2  EQU HRSFB
INPIS   EQU DESALO          nombre de pistes pour initialisation
INSEC   EQU DESALO+1      ' ' secteurs ' '
PARPIS  EQU FISALO         80 si pistes précisées, 0 sinon
PARSEC  EQU FISALO+1      ' ' secteurs ' '
*
*      pointeurs fichiers
*
ADENT   EQU DECDEB
ADTMP   EQU DECFIN
ADPS    EQU DECCIB
ADTR    EQU DECTRV

XMERGE  adresse #FF0E'      special BASIC !
XFST    adresse #FF11'      prendre infos fichier selon FICNUM
XSPUT   adresse #FF14'      ecrire sequentiel
XSTAKE  adresse #FF17'      lire sequentiel
XOPEN   adresse #FF1A'      ouvre fichier nom ds BUFNOM numero log FICNUM
XCLOSE  adresse #FF1D'      ferme fichier de numero FICNUM
XTAKE   adresse #FF20'      prendre enr. ds fichier FICNUM
XPUT    adresse #FF23'      ecrire ' ' ' ' ' '
XAPPEN  adresse #FF26'      placer index sequ. FICNUM en fin de fichierâA
XREWIN  adresse #FF29'      ' ' ' ' ' ' debut
XJUMP   adresse #FF2C'      sauter enreg. fichier sequ. FICNUM
XLGBUF  adresse #FF2F'      calcul lg tampon fichiers (selon FICNUM)
XERVEC  adresse #FF32'      restaure le vecteur d'erreur
XESAVE  adresse #FF35'      sauve l'ecran texte ou graphique (selon FLGTEL)
XCOPY   adresse #FF38'      copie les fichiers
*
*      source dans BUFTRV cible dans BUFNOM
*
*      VSALO0: b7=O b6=M
*      VSALO1: b7=N b6=C
XDNAME  adresse #FF3B'      change le nom de la disquette
*
*      longueur dans ACC1E, adresse dans ACC1M

```

UTILISATION DU TELEMOM : USAGE DU STRATSED

XSTATU	adresse #FF3E'	non implemente dans STRATSED 2.0x
XUPDAT	adresse #FF41'	recopie le DOS en memoire
*		parametres specifiques)
XFORMA	adresse #FF44'	formate la disquette ds DRIVE selon BUF2+9 (pistes)
*		et BUF2+7 (secteurs ,disp. apres un pmap)
XREN	adresse #FF47'	changement de nom
*		source dans BUFTRV (lecteur+9 carac) cible dans BUFNOM
XDELBK	adresse #FF4A'	DELBAK
XDELN	adresse #FF4D'	DEL
XPROT	adresse #FF50'	PROT
XUNPRO	adresse #FF53'	UNPROT
XDIRN	adresse #FF56'	catalogue (nom --> BUFNOM)
XBKP	adresse #FF59'	BUFNOM=No lecteur source, BUFNOM+1=No lecteur cible
XINITI	adresse #FF5C'	initialiser, lecteur dans DRIVE
*		b7 PARPIS=0, nb piste par defect. =1, nb pistes INPIS
*		b7 PARSEC=0, 17 secteurs =1, nb secteurs INSEC
XERREU	adresse #FF5F'	generer erreur
*		X = numero de l'erreur
XLOAD	adresse #FF62'	charger fichier dans BUFNOM
*		VSALO0: 00 rien de special
*		40 afficher status
*		80 forcer ne pas executer
*		VSALO1: 00 rien de special
*		80 charger a adresse DESALO
XDEFSA	adresse #FF65'	valeurs par defect SAVE: type = LM,
SAVEU		
XDEFLO	adresse #FF68'	valeurs par defect LOAD: rien de special
XSAVE	adresse #FF6B'	sauver fichier selon BUFNOM
*		DESALO,FISALO,FTYPE positionnes
*		VSALO0: 00 SAVEO
*		40 SAVEM
*		80 SAVE
*		C0 SAVEU
XNOMDE	adresse #FF6E'	detruit le fichier dans BUFNOM
XCREAY	adresse #FF71'	creer une table P/S de AY secteurs
XDETSE	adresse #FF74'	occupe le secteur Y piste A
XLIBSE	adresse #FF77'	libere le secteur Y piste A
XTRVCA	adresse #FF7A'	
XTRVNM	adresse #FF7D'	cherche le nom dans BUFNOM
*		retour Z=1 si trouve
XTRVNX	adresse #FF80'	cherche a partir de la position courante
XCABU	adresse #FF83'	transfert catalogue --> BUFNOM
XBUCA	adresse #FF86'	transfert BUFNOM --> catalogue
XVBUF1	adresse #FF89'	remplit de 0 le buffer 1
XSVSEC	adresse #FF8C'	ecrit un secteur selon DRIVE,PISTE,SECTEU,RWBUF
XSAY	adresse #FF8F'	ecrit le secteur Y piste A selon DRIVE et RWBUF
XSBUF1	adresse #FF92'	ecrit BUF1 dans le secteur Y piste A
XSBUF2	adresse #FF95'	BUF2
XSBUF3	adresse #FF98'	BUF3
XSCAT	adresse #FF9B'	ecrit le secteur de catalogue selon POSNMP,POSNMS
XPRSEC	adresse #FFA1'	lit secteur selon DRIVE,PISTE,SECTEU,RWBUF
XPBUF1	adresse #FFA4'	lit secteur Y piste A dans BUF1
XPMAP	adresse #FFA7'	lit le secteur de bitmap

DE 94
D 923
D 908
D 856

? ←

UTILISATION DU TELEMON : USAGE DU STRATSED

XRWTS adresse #FFAA' envoie la commande X au FDC 1793 (selon DRIVE etc...)

STRUCTURE DISQUETTE STRATSED

NB : la structure d'une disquette STRATSED ou SEDORIC a été créée par Fabrice BROCHE et Denis SEBBAG.

Occupation disquette:

PISTE 0 secteur 1 SECTEUR DE BOOT

00 : #80 si stratsed (et donc charger le STRATSED)
01 : nombre de secteur par piste
02 : nombre de secteur a charger
03-04 : adresse de chargement
05-07 : RTS ou JMP ou ... : sous programme appelé après le chargement
08-FF : libre

PISTE 0 secteur 2 à .. STRATSED

00-FF : STRATSED, selon sa longueur (44 secteurs en standard)

PISTE 20 secteur 1 NOM DE LA DISQUETTE

00-08 : inutilisé
09-1C : nom de la disquette (20 caractères)
1D-FF : inutilisé

PISTE 20 secteur 2 SECTEUR DE STATUS

00-01 : FF 00 indicateurs SED/Stratsed
02-03 : nombre secteurs libres
04-05 : nombre de fichiers du catalogue
06 : nombre de pistes
07 : nombre de secteurs par piste
08 : nombre de secteurs du catalogue
09 : indicateur simple double face (b7) et nbre de pistes
0A-0F : inutilisé
10-FF : bit map (bit à 1 si secteur libre, 0 si occupé)

PISTE 20 secteur 3 SECTEUR DE STATUS (bis)

00-0F : inutilisé
10-FF : bit map soit $2*240*8 = 3840$ secteurs = 960 Ko

STRUCTURE FICHIERS DIRECTS/SEQUENTIELS

variables fichiers:

TAMPFC : adresse de début des tampons fichiers
(#542-3)
BITMFC : bit map des tampons logiques
(#544-5) même convention bit map disquette
FICLON : longueur d'une fiche en direct
(#54A-B)
FICTYP : type du fichier : 0=séquentiel #80=direct
(#547)
FICNUM : numéro logique du fichier
(#548)
NBFIC : nombre de fichiers autorisés par FILE
(#549)
INDFCB : index dans la liste des secteurs de FCB du secteur FCB courant
(#546)

pointeurs:

ADENT : adresse de l'entête du tampon de no FICNUM
(4-5)
ADTMP : adresse du tampon secteur séquentiel
(6-7) définition des champs direct
ADPS : adresse du tampon FCB courant
(8-9)
ADTR : adresse du tampon de 768 octets (idem TAMPFC)
(#A-B)

UTILISATION DU TELEMON : USAGE DU STRATSED

Structure tampons fichiers: (tampons logiques)

entête:

00-0F : nom du fichier, lecteur, entête du catalogue
10 : index du FCB courant dans la liste des FCB
11 : indicateur d'écriture : 80 si écriture a eu lieu, 0 sinon
12 : index dans le secteur séquentiel de l'enreg. courant
13-1B : recopie de l'entête du FCB du fichier
1C : index du secteur courant dans le FCB courant
1D : index du dernier secteur de FCB de la liste

020-05F : liste des secteurs de FCB du fichier
060-15F : secteur de FCB courant (adresse dans ADPS)
160-25F : tampon secteur de données séquentiel ou
définition des champs en direct

Structure des fichiers sur la disquette:

-secteurs de données direct:

1er secteur : définition des champs:

25 champs maximum de 10 octets chacun:

00 : type 80=alpha, 0=flottant
01 : longueur si alpha
02-09 : nom du champ + '\$' si alpha

secteurs suivants : données en vrac

-secteurs de données séquentiel:

données en vrac: chaque enreg. séparé par type+longueur de l'enreg.
type vaut FF pour indiquer la fin du fichier

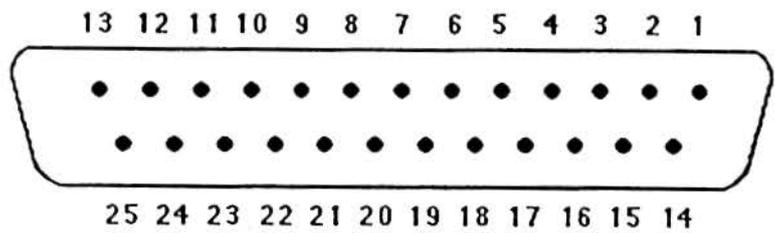
Les connecteurs du TELESTRAT

Schémas de brochage.

Tous les schémas sont indiqués en vue de l'extérieur du boîtier.

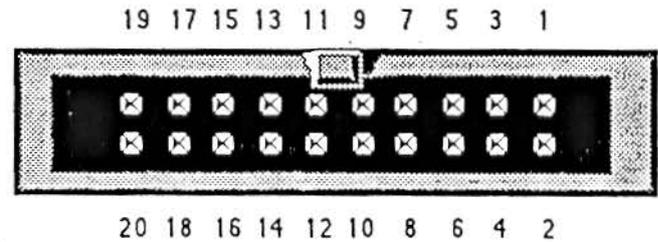
Port RS 232

- 2. Données émises (TX)
- 3. Données reçues (RX)
- 4. Demande d'émission (RTS)
- 5. Prêt pour émission (CTS)
- 7. Masse signaux
- 8. Détection de porteuse (DCD)
- 20. Terminal prêt (DTR)



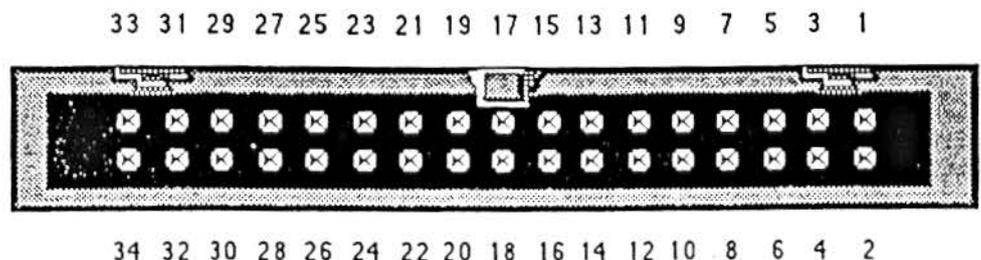
Port imprimante Parallèle

- 1. Strobe ($\overline{\text{STR}}$)
- 5. D1
- 9. D3
- 13. D5
- 17. D7
- 2, 4, 6 ... 20 : toutes bornes paires reliées à la masse.
- 3. D0
- 7. D2
- 11. D4
- 15. D6
- 19. Acknowledge ($\overline{\text{ACK}}$)



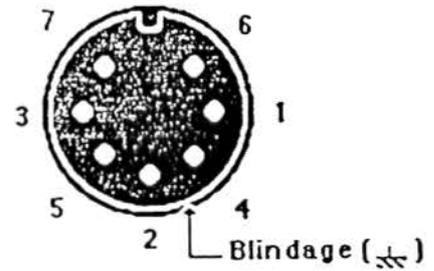
Port d'extension

- | | | | | |
|---------|-----------|----------|-----------|---------|
| 1. MAP | 2. ROMDIS | 3. Ø2 | 4. RESET | 5. I/O |
| 6. N.C. | 7. RW | 8. IRQ | 9. D2 | 10. D0 |
| 11. A3 | 12. D1 | 13. AO | 14. D6 | 15. A1 |
| 16. D3 | 17. A2 | 18. D4 | 19. D5 | 20. A4 |
| 21. A5 | 22. D7 | 23. A6 | 24. A15 | 25. A7 |
| 26. A14 | 27. A8 | 28. A13 | 29. A9 | 30. A12 |
| 31. A10 | 32. A11 | 33. +5V. | 34. Masse | |



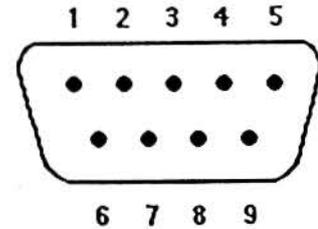
Cassette

1. Sortie des sonnées
2. Masse
3. Entrée des données
- 4 - 5 : Son (reliées intérieurement)
- 6 - 7 : Relai télécommande cassette
- Blindage = Masse



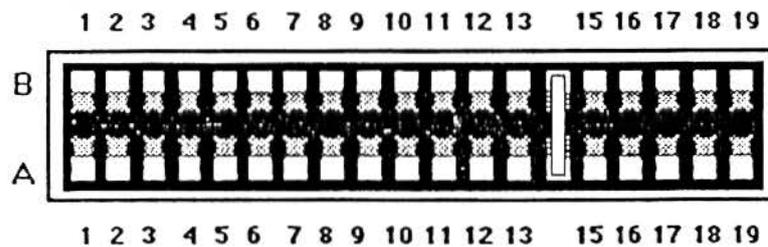
Prise Joystick

- | | |
|----------------------|----------------------|
| 1. ↑ | 6. Souris bouton n°1 |
| 2. ↓ | 7. + 5 V. |
| 3. ← | 8. Masse |
| 4. → | 9. Souris bouton n°2 |
| 5. Souris bouton n°3 | |



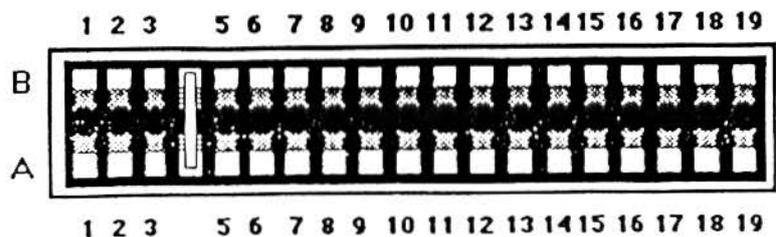
Port cartouche application (droite)

A1: CS5	A2: CS6	A3: CS3	A4: N.C.	A5: DØ
A6: D2	A7: D4	A8: D6	A9: Masse	A10: AØ
A11: A2	A12: A4	A13: A6	A14: Détrompeur	A15: A8
A16: A10	A17: A12	A18: N.C.	A19: + 5 V.	
B1: Ø ₂	B2: CS4	B3: RW	B4: N.C.	B5: D1
B6: D3	B7: D5	B8: D7	B9: N.C.	B10: A1
B11: A3	B12: A5	B13: A7	B14: Détrompeur	B15: A9
B16: A11	B17: A13	B18: NC	B19: NC	



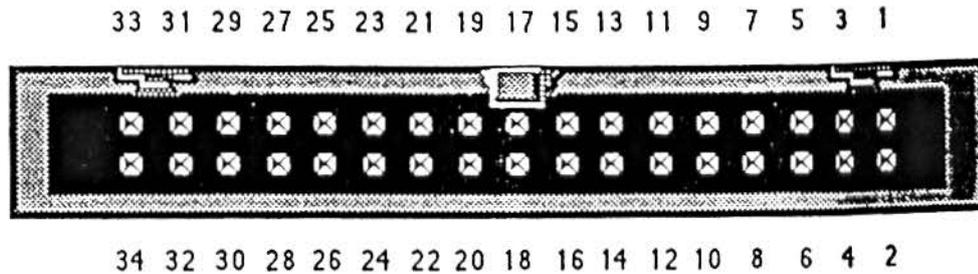
Port cartouche langage (gauche)

A1: CS1	A2: CS2	A3: CS3	A4: Détrompeur	A5: DØ
A6: D2	A7: D4	A8: D6	A9: Masse	A10: AØ
A11: A2	A12: A4	A13: A6	A14: N.C.	A15: A8
A16: A10	A17: A12	A18: N.C.	A19: + 5 V.	
B1: Ø ₂	B2: CSØ	B3: N.C.	B4: Détrompeur	B5: D1
B6: D3	B7: D5	B8: D7	B9: N.C.	B10: A1
B11: A3	B12: A5	B13: A7	B14: N.C.	B15: A9
B16: A11	B17: A13	B18: N.C.	B19: + 5 V.	



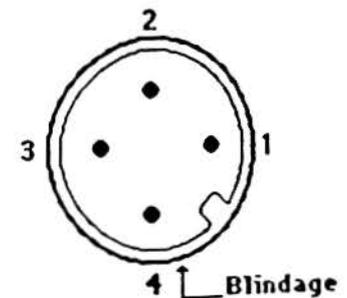
Port Disquettes

6. DS3	8. Index	10. DSØ	12. DS1	14. DS2
16. Motor ON	18. DIR	20. STEP	22. W Data	24. W Gate
26. TRACK ØØ	28. WPROT	30. R Data	32. SS	
2, 4, 34 : Non connecté		1, 3, 5, ... 33 : Masse		



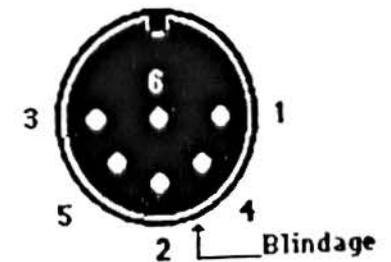
Alimentation

1. + 12 V
2. + 5 V
3. - 12 V (à partir du n° de série 1000)
4. Masse



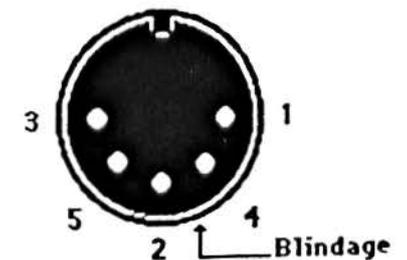
R.V.B. (pour télé ou moniteur couleur)

1. Rouge
 2. Vert
 3. Bleu
 4. Synchronisation
 5. Son
 6. +12 V (commutation Péritel)
- Blindage = Masse



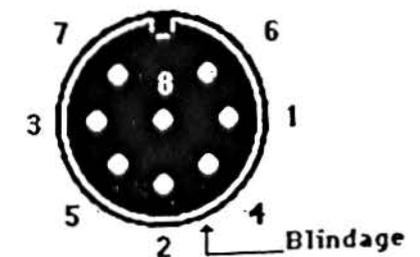
M.I.D.I.

1. CA1
 2. + 5 V.
 3. PA3
 4. CB2
 5. PA 6
- Blindage = Masse



MINITEL

1. Non Connecté
 2. Sortie des données
 3. Masse
 4. N.C.
 5. Entrée des données
 - 6, 7, 8 : N.C.
- Blindage : Masse



CODAGE D'UNE PAGE VIDEOTEX

I GENERALITES

Une page vidéotex au format Telestrat est en fait composée de deux parties :

- un entête qui sert à l'éditeur de page du Telestrat,
- la liste des codes vidéotex composant la page, terminée par un 0.

Seule la deuxième partie est obligatoire pour l'utilisation d'une page par l'arborescence. Il est ainsi possible d'utiliser des pages créées ailleurs simplement en plaçant les codes de la pages, terminés par un 0.

Il est en revanche indispensable que la première partie soit correctement initialisée pour que l'éditeur de page puisse prendre en compte les paramètres de codage (taille de la page, effacement ..).

II CODAGE D'UNE PAGE

nom	adresse	bits	rôle
CODCOU	7FFF	b0-b2	couleur de fond (*)
CODDX	8000		début de la fenêtre, X
CODFX	8001		fin de la fenêtre, X
CODDY	8002		début de la fenêtre, Y
CODFY	8003		fin de la fenêtre, Y
CODCX	8004		position du curseur, X
CODCY	8005		position du curseur, Y
FLGCOD	8006		status de la page
		b7	effacement de la page
		b6	affichage du curseur
		b5	masquage / démasquage (*)
		b4	codage accordéon (*)
		b3	inversion du sens (*)
		b2	codage colonne (*)
		b1	sens Y négatif (*)
		b0	sens X négatif (*)
	8007		début des codes vidéotex, terminés par un 0.

(*) Telestrat][uniquement.

Telestrat I

Voici la liste des caractères utilisés pour coder une page vidéotex :

nom	valeur hexa	rôle
NUL	00	fin de la page
FF	0C	effacement de l'écran
SO	0E	passage en G1
SI	0F	passage en G0
DC1	11	curseur ON
REP	12 xx	répétition de caractère
DC4	14	curseur OFF
SS2	19	accès ponctuel à G2
ESC	1B xx	attributs de visualisation
US	1F yy xx	accès direct à l'écran
	20-7F	G0
	20-5F	G1
	20-7F	G2

Le codage s'effectue de gauche à droite et de haut en bas. Les déplacements sont effectués avec US yy xx.

Telestrat II

Voici la liste des caractères utilisés pour coder une page vidéotex :

Nom	valeur hexa	rôle
NUL	00	fin de la page
BS	08	curseur gauche
HT	09	curseur droit
LF	0A	curseur bas
VT	0B	curseur haut
FF	0C	effacement de l'écran
CR	0D	retour au début de la ligne
SO	0E	passage en G1
SI	0F	passage en G0
DC1	11	curseur ON
REP	12 xx	répétition de caractère
DC4	14	curseur OFF
SS2	19	accès ponctuel à G2
ESC	1B 20 23 xx	masquage / démasquage
ESC	1B xx	attributs de visualisation
RS	1E	retour en haut à gauche de l'écran
US	1F yy xx	accès direct à l'écran
	20-7F	G0
	20-5F	G1
	20-7F	G2

Dans tous les types de codage choisis, le logiciel teste le plus court des deux "chemins", relatif (déplacement avec 08 09 0A 0B 0D) ou absolu (déplacement avec US yy xx). En effet, le déplacement à l'aide de US a pour caractéristique d'effacer tous les attributs (blanc sur fond noir non souligné non clignotant, G0), ce qui n'est pas forcément intéressant.

CODAGE D'UNE ARBORESCENCE

I STRUCTURE DE L'ARBORESCENCE

L'arborescence est constituée d'une suite de page repérées par leur numéro (de 0 à ...), reliées par leur longueur qui permet de passer de l'une à l'autre.

Lorsque une page est supprimée, sa longueur est simplement réduite à 1 pour ne pas avoir à recalculer toutes les références.

L'arborescence est stockée à partir de #4000, et monte au maximum jusqu'à #7FFE, ce qui donne un maximum théorique de 1024 pages.

Chaque page occupe 16 octets, ou 16 plus le double du nombre de choix pour les sommaires.

II STRUCTURE D'UNE PAGE

adresse relative	rôle
00	longueur de la page
01-07	nom de la page en ASCII
08-09	numéro de la page suivante (SUITE)
0A-0B	numéro de la page précédente (RETOUR)
0C	octet de status I
b7	sommaire
b6	pas de suite (fin de branche ou sommaire)
b5	pas de précédente (première page)
b4	page temporisée
b3	page langage
b2	journal cyclique (première page)
b1	page Fantome (pas de suite)
b0	BAL
0D	octet de status II
	si sommaire : nombre de choix
	si temporisée : temporisation en secondes
0E	poids faible page GUIDE (*)
0F	b6-b7 numéro du lecteur
	b0-b5 poids fort page GUIDE (*)
10-...	Sommaire : numéro des pages choix

(*) Telestrat][uniquement.

VECTEURS TELEMATIC

XTGET	FFB9	saisir un caractère dans A, déconnexion automatique. gère l'impression en mode borne de communication retourne code ascii dans A, de #A0 à #Ax pour SEP xx
XTINPU	FFBC	saisir une chaîne de longueur A, à la sortie, la chaîne est dans BUFEDT, A et 255 contiennent le code de sortie (#A0..)
XVDTEX	FFC2	lancer le codage de la page courante, avec les paramètres courants.
XSERVE	FFC5	lancer le serveur, paramètre dans X.

Scanné par Andrec