

memento

**CLEFS  
POUR LE  
ZX SPECTRUM  
ET TIMEX 2000**

Jean-François Séhan

Editions du 

**CLEFS POUR  
LE ZX SPECTRUM  
ET TIMEX 2000**

La collection « **MEMENTOS** » est constituée de recueils de données sur un type de matériels.  
Dix titres sont actuellement disponibles dans cette collection :

- Clefs pour le TRS-80 — tomes 1 et 2 — Rémy Pineau
- Le Basic de A à Z — Jacques Boigontier
- Clefs pour l'Apple II — Nicole Bréaud-Pouliquen
- CP/M mot par mot — Yvon Dargery
- Clefs pour le ZX-81 — Jean-François Séhan
- Clefs pour le CBM — Daniel-Jean David
- Clefs pour le VIC — Daniel-Jean David
- PC/DOS mot par mot — Yvon Dargery
- Clefs pour le ZX-spectrum — Jean-François Séhan

D'autres ouvrages relatifs au ZX-81 :

- La pratique du ZX-spectrum — tome 1 — Xavier Linant de Bellefonds  
Collection « **MATERIELS** »  
série bleue
- La pratique du ZX-spectrum — tome 2 — Marcel Henrot  
Collection « **MATERIELS** »  
série rouge
- 102 programmes pour ZX-81 et spectrum — Jacques Deconchat  
Collection « **PROGRAMMES** » — Série verte
- Le petit livre du Spectrum — Trevor Toms — traduit par Allan Keil

#### RAPPELS

##### Les séries :

En fait, il faudrait parler de niveaux, puisque la couleur attachée à chaque ouvrage permet de situer la « force » de celui-ci selon le code suivant :

- Série VERTE* : ouvrage d'initiation ne nécessitant que des connaissances de base.
- Série BLEUE* : suppose une connaissance élémentaire du sujet traité.
- Série ROUGE* : ouvrage d'approfondissement, niveau de complexité moyen.
- Série NOIRE* : ouvrage d'approfondissement, niveau de complexité élevé.

##### Les collections :

Les ouvrages d'Édition du P.S.I., actuellement au nombre de 100, sont répartis en collections :

« **LANGAGES** », « **MATERIELS** », « **PROGRAMMES** », « **GUIDES PRATIQUES** », « **MEMENTOS** », « **UTILISATIONS DE L'ORDINATEUR** », « **LOGIGUIDE** » et pour l'initiation, outre quelques livres hors collection, « ... **POUR TOUS** ».

La loi du 11 mars 1957 n'autorisant, aux termes des alinéas 2 et 3 de l'article 41, d'une part, que les « copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective », et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toute représentation ou reproduction intégrale, ou partielle, faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause, est illicite » (alinéa 1<sup>er</sup> de l'article 40).

Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles 425 et suivants du Code Pénal.

© Editions du P.S.I. Parc industriel nord, Bâtiment 9, 77200 Torcy Marne-la-Vallée 1983  
ISBN : 2-86595-117-0

# CLEFS POUR LE ZX SPECTRUM ET TIMEX 2000

par  
Jean-François Séhan



Editions du P.S.I.  
1983

**Du même auteur :**

- Variations pour PC-1211
- Etudes pour ZX-81 - tomes 1 et 2
- Suites pour PC-1500
- Le TI-99/4A à l'affiche
- Le VIC à l'affiche
- Clefs pour le ZX81 et Timex 1000

## PRESENTATION

Ce livre est destiné aux nombreux utilisateurs de ZX SPECTRUM perdus parmi la vaste documentation mise à leur disposition. Pas de parlote inutile, mais des renseignements, et encore des renseignements !

Ce mémento permet d'accéder rapidement aux données techniques dont vous aurez besoin. Vous y trouverez la liste des instructions Basic commentées, les mnémoniques de l'assembleur Z80 et leurs codes objets, les points d'entrées variables système et les caractéristiques des principales extensions, ainsi que les codes clavier et écran, les codes erreur et le brochage des circuits intégrés.

Une liste d'astuces pour mieux utiliser l'écran, les cassettes et les programmes en langage machine complète cet ouvrage.

Et, par souci d'efficacité un index, placé en fin d'ouvrage, vous indique où trouver telle instruction Basic ou telle variable système.

# SOMMAIRE

	pages
<b>COMMANDES</b>	9
Fonctions Basic	9
Commandes Basic	12
Opérateurs logiques	18
Codes des mots-clés	19
<b>CODES ERREUR</b>	21
Codes erreur	21
Liste par mots-clés	24
<b>ECRAN-CLAVIER</b>	25
Jeu de caractères	25
Grille d'écran	29
<b>LANGAGE MACHINE</b>	31
Organisation interne du Z80	31
Registres du Z80	32
Schéma simplifié du Spectrum	33
Jeu d'instructions du Z80	34
Codes des instructions du Z80	40
Tableaux de désassemblage	55
<b>CONNECTEURS</b>	59
Connecteurs du Spectrum	59
Brochage des circuits intégrés	60

<b>ADRESSES</b>	65
Carte mémoires du Spectrum	65
Adresses de la ROM Basic du Spectrum	66
Variables système du Spectrum	86
<b>LES TRUCS</b>	97
Trucs et astuces	97
<b>ANNEXES</b>	103
Liste des attributs vidéo	103
Conversion décimal/binaire	105
Conversion décimal/hexadécimal	106
<b>INDEX DES MOTS CLES</b>	109

## FONCTIONS BASIC

### Fonctions mathématiques

Mot-clé	Définition, exemples
ABS	Donne la valeur absolue de l'argument : PRINT ABS -5 → 5
ACS	Donne l'arccosinus de l'argument (en radian) : PRINT ACS 0.5 → 1.0471976
ASN	Donne l'arcsinus de l'argument (en radian) : PRINT ASN 0.6 → 0.64350111
ATN	Donne l'arctangente de l'argument (en radian) : PRINT ATN 0.7 → 0.61072596
ATTR	Donne les attributs d'un caractère particulier sur l'écran. Cette valeur se décompose ainsi : Bit 8 = 1 si clignotement ; Bit 7 = 1 si surintensité lumineuse ; Bit 6 à 4 = couleur du papier Bit 3 à 1 = couleur de l'encre. 10 PRINT AT 10,10;FLASH 1;PAPER 5;INK 7;20 PRINT ATTR(10,10) → 175 (128+8*5+7)
BIN	Nombre dont la valeur est donnée en notation binaire (utilisé surtout pour les caractères programmables) : 10 LET A=BIN 01010101
COS	Donne le cosinus de l'argument (en radian) : PRINT COS 1.6 → 0.029199522
EXP	Fonction exponentielle : PRINT EXP 10 → 22026.466
INT	Donne la valeur entière de l'argument : PRINT INT 2.5 → 2 PRINT INT -5.4 → -6
LN	Fonction logarithme naturel (base e) : PRINT LN 5 → 1.6094379
PI	Constante de valeur 3.141592653. Seuls les huit premiers chiffres sont affichés à l'écran (3.1415927).

Mot-clé	Définition, exemples
SGN	Signe de l'argument. -1 si négatif, 0 si nul et 1 si positif.
SIN	Donne le sinus de l'argument (en radian) : PRINT SIN 1 → 0.84147099
SQR	Donne la racine carrée de l'argument : PRINT SQR 25 → 5
TAN	Donne la tangente de l'argument (en radian) : PRINT TAN 1.5 → 14.10142

Fonctions chaînes de caractères

Mot-clé	Définition, exemples
CHR\$	Donne le caractère de l'argument (numérique) : PRINT CHR\$ 38 → "A"
CODE	Donne le code du premier caractère d'une chaîne, ou 0 si la chaîne est vide : 10 LET A\$ = "SPECTRUM" 20 PRINT CODE A\$ donne 83 (code de 5).
LEN	Longueur d'une chaîne : 10 LET A\$ = "SPECTRUM" 20 PRINT LEN A\$ → 8
STR\$	Convertit un nombre en chaîne de caractères : PRINT STR\$ 12.5 "12.5"
TO	Extrait une sous-chaîne d'une chaîne de caractères : LET A\$ = "SPECTRUM" PRINT A\$( TO 4) → "SPEC" PRINT A\$(3 TO 5) → "ECT" PRINT A\$(5 to ) → "TRUM"
VAL	Donne la valeur numérique d'une chaîne : PRINT VAL "12.5" → 12.5
VAL\$	Supprime d'une chaîne les deux doubles quotes (""). Le résultat est toujours une chaîne : PRINT VAL\$ ""SPECTRUM"" → "SPECTRUM"

Fonctions langage machine

Mot-clé	Définition, exemples
IN	Donne la valeur d'un port d'entrée/sortie (clavier, imprimante, micro-drives, etc...) spécifié : 10 LET A=IN 65278
OUT	Place dans le port d'entrée/sortie (clavier, imprimante, micro-drive, etc...) la valeur spécifiée : 10 OUT 251,128
PEEK	Restitue l'octet contenu dans la mémoire spécifiée : 10 PRINT PEEK 16398 20 LET A=PEEK 16398+256*PEEK 16399
POKE	Place dans l'octet désigné le nombre spécifié : 10 POKE 16442,4
USR	Appelle un sous-programme écrit en langage machine dans la mémoire vive ou la mémoire morte du Basic : 10 PRINT USR 42514 100 A=USR 16514 Au retour du sous-programme en langage machine, USR donne la valeur de la paire de registres BC du Z80.
USR " "	Définit un caractère utilisateur. Celui-ci doit correspondre à une lettre de l'alphabet entre A et U (soit 21 caractères au total). Le caractère étant dessiné dans une matrice 8x8, on débute par USR "A" jusqu'à USR "A"+7 pour définir A : 10 POKE USR "A" BIN 00001111 20 POKE USR "A"+1, BIN 11110001 30 ... etc... 80 POKE USR "A"+7, BIN 00001110

D : instruction utilisable uniquement en mode direct.  
 P : instruction utilisable uniquement dans un programme.

Mot-clé	Définition, exemples
<b>BEEP</b>	Emet sur le haut-parleur un beep sonore de durée T secondes et de F demi-tons au dessus du UT de référence (ou en dessous si F est négatif) : 10 LET T=2:LET F=5 20 BEEP T,F
<b>BORDER</b>	Définit la couleur du pourtour de l'écran et le fond de la fenêtre texte : 10 BORDER 2
<b>BRIGHT</b>	Tous les caractères affichés après cette instruction seront en brillance normale (0), brillance forte (1) ou transparente (8) : 10 BRIGHT 1
<b>CIRCLE</b>	Trace un cercle sur l'écran dont le centre a pour coordonnées X,Y et de rayon Z : 10 CIRCLE X,Y,Z
<b>CLEAR</b>	Supprime les variables déjà déclarées et libère l'espace qu'elles occupaient.
<b>CLS</b>	Efface l'écran en plaçant dans le fichier d'affichage 25 newlines.
D <b>CONT</b>	Continue l'exécution du programme après une instruction STOP ou un BREAK (commande CONTINUE sur l'écran).
<b>COPY</b>	Recopie le contenu de l'écran sur l'imprimante si elle est connectée.
P <b>DATA</b>	Liste de données numériques ou alphanumériques lues par la fonction READ : 10 DATA 83,"SPECTRUM"
P <b>DEF FN</b>	Déclaration d'une fonction utilisateur dont l'argument est une variable : 10 DEF FN F(X)=X+(X*2)+(X*3)
D <b>DELETE</b>	Supprime le dernier caractère tapé au clavier.
P <b>DIM</b>	Dimensionne un tableau (fixe le nombre et les valeurs maximum des indices). Pour les chaînes, le dernier indice indique la longueur des chaînes du tableau : 10 DIM A(50) 20 DIM A(20,10) 30 DIM A\$(30,15)

Mot-clé	Définition, exemple
<b>DRAW</b>	Trace à l'écran une ligne de la position courante aux coordonnées X,Y en effectuant une rotation Z (ligne droite si Z=0 ou arc de cercle si Z=PI) : 10 DRAW 10,10,PI
D <b>EDIT</b>	Place la ligne pointée par le curseur en mode édition.
<b>FLASH</b>	Tous les caractères affichés après cette instruction clignoteront (1), ne clignoteront plus (0) ou resteront sans changement (8) : 10 FLASH 1
P <b>FN</b>	Appelle une fonction utilisateur définie par DEF FN : 10 LET A=FN F(X)
P <b>FOR TO STEP</b>	Introduit une boucle : toutes les instructions comprises entre FOR I= A TO B STEP C et le NEXT I correspondant seront répétées pour toutes les valeurs de I allant de A à B, de C en C. Si C n'est pas spécifiée, elle prend, par défaut, la valeur 1 : 10 FOR I=1 TO 20 STEP 2 20 PRINT I,I*2 30 NEXT I
<b>GOSUB</b>	Appelle le sous-programme de la ligne spécifiée : 100 GO SUB 100
<b>GOTO</b>	Effectue un saut à la ligne spécifiée : 100 GO TO 150
<b>IF THEN</b>	Test logique de la forme IF condition THEN instruction : 10 IF A=10 THEN GO TO 100 100 IF A\$="OUI" THEN PRINT "FAUX" 200 IF A<0 THEN LET A=1
<b>INK</b>	Donne la couleur de l'encre avec laquelle les caractères seront écrits : 10 INK 5
P <b>INKEY\$</b>	Donne le caractère tapé au clavier ou une chaîne vide si aucune touche n'est enfoncée : 10 A\$=INKEY\$ 20 IF INKEY\$="" THEN GO TO 20

Mot-clé	Définition, exemples
P INPUT	Place dans la variable spécifiée la valeur ou les caractères tapés au clavier : 10 INPUT A 20 INPUT "VOTRE NOM ?";A\$
INVERSE	Inverse les couleurs des caractères qui vont être écrits sur l'écran. La couleur de l'encre devient celle du papier et vice-versa pour la valeur 1 et reste en vidéo normale pour 0 : 10 INVERSE 1
LET	Affecte à une variable le résultat de l'expression à droite du égal : 10 LET A\$="SPECTRUM" 20 LET B=5+(2*3)
P LINE	Après l'instruction INPUT d'une chaîne de caractères, supprime les deux doubles quotes : 10 INPUT LINE A\$ Dans une instruction SAVE, indique le numéro de la première ligne à exécuter après l'instruction LOAD : SAVE "PROGRAMME" LINE 200
D LIST	Liste le programme sur l'écran à partir du début ou de la ligne spécifiée : LIST LIST 100
D LLIST	Liste le programme sur l'imprimante si elle est connectée.
LOAD	Charge un programme du magnétophone. Si aucun nom n'est indiqué, LOAD charge le premier programme de la bande : LOAD "PROGRAMME" LOAD ""
LOAD DATA	Identique à LOAD, mais charge en mémoire un tableau : LOAD "TABLEAU" DATA A\$()
LOAD CODE	Identique à LOAD, mais charge en mémoire un programme en langage machine : LOAD "LM" CODE 16384,6912 où 16384 est l'adresse de début et 6912 le nombre d'octets.

Mot-clé	Définition, exemples
LOAD SCREEN\$	Identique à LOAD, mais charge en mémoire un écran : LOAD "ECRAN" SCREEN\$
LPRINT	Identique à PRINT, mais sur l'imprimante.
MERGE	Identique à LOAD, mais n'effectue pas de NEW avant le chargement du programme. Si des numéros de ligne sont identiques, ils sont remplacés par ceux du nouveau programme : 10 MERGE "PROGRAMME"
D NEW	Efface de la mémoire vive le programme et les variables.
P NEXT	Incrémente la variable correspondant au FOR (voir FOR TO STEP).
OVER	Superpose deux caractères écrits sur la même position d'écran (1) ou le remplace (0) : 10 OVER 1
PAPER	Définit la couleur du fond des prochains caractères écrits sur l'écran : 10 PAPER 6
POINT	Indique si un point de l'écran est affiché (1 pour la couleur de l'encre et 0 pour la couleur du papier) : 10 POINT (10,10)
P PAUSE	Arrête l'exécution du programme en cours, pendant n trames de la télévision. N est compris entre 1 (0.02 seconde) et 65535 (environ 22 minutes). Si N=0, le programme attend qu'une touche soit enfoncée : 10 PAUSE 1000 250 PAUSE 40000
PLOT	Affiche à l'écran l'élément C,L (où C est le numéro de colonne et L la ligne ; voir chapitre Caractères) : 10 PLOT 12,24
PRINT	Affiche à l'écran des valeurs numériques ou des chaînes de caractères : PRINT A,A\$;"SPECTRUM"
PRINT AT	Identique à PRINT, mais à l'adresse X,Y désignée : PRINT AT 10,10;"ZX-84".

Mot-clé	Définition, exemples
<b>PRINT TAB</b>	Identique à PRINT, mais en partant de la colonne désignée : 10 PRINT TAB 10;"BONJOUR"
<b>RANDOMIZE</b>	Fixe la séquence des nombres pseudo-aléatoires à partir de N, N étant compris entre 0 et 65536 : 10 RANDOMIZE 2000 20 PRINT RND
P <b>READ</b>	Lit dans le programme les données définies par l'instruction DATA : 10 DATA 83,"SPECTRUM" 20 READ A,A\$
P <b>REM</b>	Début une ligne de commentaires dans un programme : 10 REM DEBUT DU PROGRAMME
P <b>RETURN</b>	Retour au programme principal après l'appel d'un sous-programme par GOSUB.
P <b>RESTORE</b>	Définit la ligne à laquelle doit commencer la lecture des DATAs. Si aucun numéro n'est spécifié, la lecture commence à la première ligne contenant des DATAs : 10 RESTORE 100 RESTORE 50
<b>RND</b>	Donne un nombre pseudo-aléatoire : 10 PRINT RND 20 LET A=RND*10
<b>RUN</b>	Exécute le programme en mémoire à partir de la première ligne ou du numéro de ligne spécifiée : RUN RUN 100 Cette commande effectue d'abord un CLS.
<b>SAVE</b>	Sauve sur le magnétophone le programme en mémoire : SAVE "PROGRAMME"
<b>SAVE LINE</b>	Identique à SAVE, mais la commande de chargement LOAD exécute automatiquement le programme à la ligne donnée : SAVE "PROGRAMME" LINE 1000
<b>SAVE DATA</b>	Identique à SAVE, mais pour un tableau : SAVE "TABLEAU" DATA A\$()

Mot-clé	Définition, exemples
<b>SAVE CODE</b>	Identique à SAVE, mais pour les programmes en langage machine dont on spécifie l'adresse de début et le nombre d'octets : SAVE "LM" CODE 16384,6912
<b>SAVE SCREEN\$</b>	Identique à SAVE, mais pour les écrans uniquement : SAVE "ECRAN" SCREEN\$
<b>SCREEN\$</b>	Donne le caractère de l'écran dont on spécifie le numéro de ligne et de colonne : 10 LET A\$=SCREEN\$(10,20)
P <b>STOP</b>	Arrête l'exécution du programme en cours. On peut alors visualiser des variables et repartir par CONT ou CONTINUE.
<b>VERIFY</b>	Compare le programme sur bande avec celui en mémoire. Cette vérification permet de déceler les mauvais enregistrements : VERIFY "PROGRAMME"
<b>VERIFY DATA</b>	Identique à VERIFY, mais vérifie un tableau sauvé sur cassette : VERIFY "TABLEAU" DATA A\$()
<b>VERIFY CODE</b>	Identique à VERIFY, mais vérifie un programme en langage machine sauvé sur cassette : VERIFY "LM" CODE 16384,6912
<b>VERIFY SCREEN\$</b>	Identique à VERIFY, mais vérifie un écran sauvé sur cassette : VERIFY "ECRAN" SCREEN\$



LISTE DES CODES ERREUR

<i>Code</i>	<i>Signification</i>	<i>Cas de :</i>
0	Le programme s'arrête normalement après l'exécution de la dernière ligne. Aucune erreur n'a été rencontrée.	
1	NEXT sans FOR. La variable de contrôle d'un NEXT n'a jamais été définie par FOR, mais existe cependant dans le programme comme variable normale.	NEXT
2	Variable indéfinie. La variable utilisée n'a pas été définie préalablement, soit par une instruction LET ou INPUT, soit par l'instruction DIM s'il s'agit d'une variable indiquée.	
3	Indice hors intervalle. La valeur de l'indice ou le nombre d'indices d'un tableau dépasse la taille définie par l'instruction DIM.	
4	Mémoire remplie. Toute la mémoire vive du SPECTRUM est prise par les lignes de programme, les variables et le fichier d'affichage.	LET, INPUT, DIM, FOR, GOSUB
5	Hors écran. Une instruction INPUT a tenté de générer une 24ème ligne dans la partie inférieure de l'écran. Cette erreur peut aussi résulter d'un PRINT AT 22,...	INPUT, PRINT AT
6	Nombre trop grand. Le résultat d'une opération donne un nombre supérieur de 10 puissance 38.	
7	RETURN sans GOSUB. Le programme a rencontré une instruction RETURN sans qu'il n'y ait jamais eu de GOSUB.	RETURN
8	Fin d'un fichier sur micro-drive.	

Code	Signification	Cas de :
9	Point d'arrêt. Le programme a rencontré une instruction STOP. Le programme peut continuer en utilisant CONTINUE, mais ne tente pas l'exécution d'une nouvelle instruction STOP.	STOP
A	Argument erroné. Une fonction mathématique a pour argument une chaîne de caractères.	SQR, LN, ASN ...
B	Entier hors intervalle. L'argument requis est supérieur à 65535 ou négatif.	RUN, GOTO, DIM, USR...
C	L'argument de VAL (chaîne) n'est pas une expression numérique autorisée.	VAL
D	Une commande a été interrompue par la touche BREAK pendant l'utilisation d'un périphérique. La commande CONTINUE répète la dernière instruction.	LOAD, SAVE, VERIFY, MERGE, LPRINT, LLIST, COPY
E	Fin des données. Il n'y a plus de DATAs à lire par l'instruction READ, ou aucune ligne de DATAs n'existe dans le programme.	READ
F	Commande SAVE erronée. L'argument de SAVE est une chaîne vide ou de plus de 10 caractères.	
G	Fin de mémoire programme. Il n'y a plus de place pour stocker de nouvelles lignes de programme.	
H	STOP dans INPUT. L'instruction STOP a été tapée pendant l'entrée de données par INPUT. La commande CONTINUE répète la dernière instruction INPUT.	INPUT
I	FOR sans NEXT. Une boucle FOR qui n'est pas exécutée (exemple, FOR I=1 TO 0) n'a pas de NEXT correspondant.	FOR
J	Erreur d'entrée/sortie sur un micro-drive.	
K	Couleur interdite. L'argument spécifié n'est pas une valeur autorisée.	INK, PAPER, BORDER, FLASH, BRIGHT, OVER, INVERSE
L	Le programme a été interrompu par la touche BREAK. La commande CONTINUE saute à l'instruction suivante.	

Code	Signification	Cas de :
M	Pointeur haut de mémoire (RAMTOP) incorrect. La valeur spécifiée pour RAMTOP est trop grande (pointe une adresse qui n'a pas de mémoire) ou trop petite (pointe la ROM ou les variables système).	CLEAR
N	Instruction non trouvée. On effectue un saut à une instruction qui n'existe plus.	RETURN, NEXT, CONTINUE
O	Commande interdite sur un micro-drive.	
P	FN sans DEF. On utilise une instruction FN qui n'a pas été définie par un DEF FN.	FN
Q	Erreur d'argument. Le nombre ou le type d'arguments utilisés sont erronés (chaîne à la place d'un numérique ou vice-versa).	FN
R	Erreur sur cassette. Le fichier demandé est présent sur la bande, mais des erreurs de lecture sont apparues.	VERIFY, LOAD, MERGE

CODES ERREUR

EUR  
:  
EXT,  
AD,

Clé	Codes erreurs	Clé	Codes erreurs	Clé	Codes erreurs
*	6	DIM	4,B	OVER	K
+	6	DRAW	B	PAPER	K
-		ERASE	0	PAUSE	B
/		EXP	6	PEEK	B
<=		FLASH	K	PI	
<>		FN	P,Q	PLOT	B
<		FOR	4,I	POINT	B
=		FORMAT	0	POKE	B
>=		GO SUB	4,B	PRINT	
>		GO TO	B	RANDOMIZE	B
ABS		IF		READ	E
ACS	6,A	INK	K	REM	
AND		INKEY\$		RESTORE	B
ASN	6,A	INPUT	4,5,H	RETURN	7,N
ATN		INT		RND	
ATTR	B	INVERSE	K	RUN	B,M
AT	5,B	IN	B	SAVE	D,F
BEEP	B	LEN		SCREEN\$	B
BTN		LET	4	SGN	
BORDER	B,K	LINE	B	SIN	
BRIGHT	K	LIST	B	SQR	A
CAT	0	LLIST	B,D	STEP	
CHR\$	B	LN	A	STOP	9
CIRCLE	B	LOAD	4,D,R	STR\$	
CLEAR	M	LPRINT	D	TAB	B
CLOSE #	B	MERGE	4,D,R	TAN	
CLS		MOVE		THEN	B
CODE		NEW		TO	3
CONTINUE	N	NEXT	1,2,N	USR	A,B
COPY	D	NOT		VAL\$	C
COS		OPEN #	F	VAL	C
DATA		OR		VERIFY	D,R
DEF FN		OUT	B	↑	6

Remarque : toutes les expressions utilisant une ou plusieurs variables non définies donnent un code erreur 2.

# ECRAN-CLAVIER

E  
C  
R  
A  
N  
-  
C  
L  
A  
V  
I  
E  
R

## JEU DE CARACTERES

Tous les caractères de ce tableau sont affichables à l'écran (même les instructions), à l'exception des commandes comme EDIT ou GRAPHICS. On peut les obtenir par le programme suivant :

```
10 FOR I=32 TO 255
20 PRINT CHR$ I ;" ";
30 NEXT I
```

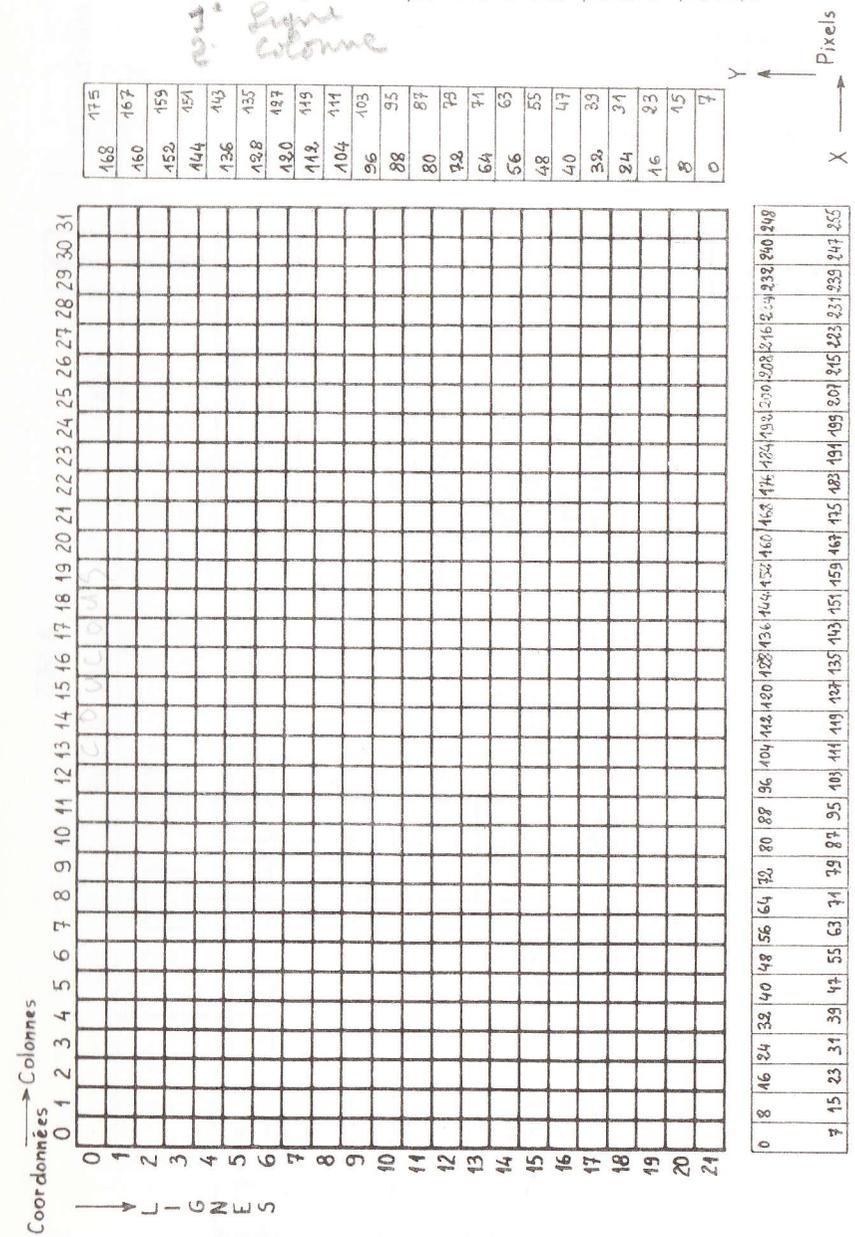
Dec.	Hexa.	Caractères	Dec.	Hexa.	Caractères
0	00	] inutilisés	30	1E	inutilisés
1	01		31	1F	inutilisés
2	02		32	20	espace
3	03	] inutilisés	33	21	!
4	04		34	22	"
5	05		35	23	#
6	06	PRINT virgule	36	24	\$
7	07	EDIT	37	25	%
8	08	curseur gauche	38	26	&
9	09	curseur droite	39	27	'
10	0A	curseur bas	40	28	(
11	0B	curseur haut	41	29	)
12	0C	DELETE	42	2A	*
13	0D	ENTER	43	2B	+
14	0E	numérique	44	2C	,
15	0F	inutilisés	45	2D	-
16	10	INK commande	46	2E	.
17	11	PAPER commande	47	2F	/
18	12	FLASH commande	48	30	0
19	13	BRIGHT commande	49	31	1
20	14	INVERSE commande	50	32	2
21	15	OVER commande	51	33	3
22	16	AT commande	52	34	4
23	17	TAB commande	53	35	5
24	18	] inutilisés	54	36	6
25	19		55	37	7
26	1A		56	38	8
27	1B	] inutilisés	57	39	9
28	1C		59	3A	:
29	1D		59	3B	;

Dec.	Hexa.	Caractères	Dec.	Hexa.	Caractères
60	3C	<	107	6B	k
61	3D	=	108	6C	l
62	3E	>	109	6D	m
63	3F	?	110	6E	n
64	40	@	111	6F	o
65	41	A	112	70	p
66	42	B	113	71	q
67	43	C	114	72	r
68	44	D	115	73	s
69	45	E	116	74	t
70	46	F	117	75	u
71	47	G	118	76	v
72	48	H	119	77	w
73	49	I	120	78	x
74	4A	J	121	79	y
75	4B	K	122	7A	z
76	4C	L	123	7B	{
77	4D	M	124	7C	}
78	4E	N	125	7D	~
79	4F	O	126	7E	-
80	50	P	127	7F	©
81	51	Q			☐
82	52	R	128	80	☐
83	53	S	129	81	☐
84	54	T			☐
85	55	U	130	82	☐
86	56	V	131	83	☐
87	57	W			☐
88	58	X	132	84	☐
89	59	Y	133	85	☐
90	5A	Z			☐
91	5B	[	134	86	☐
92	5C	/	135	87	☐
93	5D	]			☐
94	5E	↑	136	88	☐
95	5F	⎵	137	89	☐
96	60	£			☐
97	61	a	138	8A	☐
98	62	b	139	8B	☐
99	63	c			☐
100	64	d	140	8C	☐
101	65	e	141	8D	☐
102	66	f			☐
103	67	g	142	8E	☐
104	68	h			☐
105	69	i	143	8F	☐
106	6A	j			☐

Dec.	Hexa.	Caractères	Dec.	Hexa.	Caractères
144	90	(a)	190	BE	PEEK
145	91	(b)	191	BF	IN
146	92	(c)	192	C0	USR
147	93	(d)	193	C1	STR\$
148	94	(e)	194	C2	CHR\$
149	95	(f)	195	C3	NOT
150	96	(g)	196	C4	BIN
151	97	(h)	197	C5	OR
152	98	(i)	198	C6	AND
153	99	(j)	199	C7	< =
154	9A	(k)	200	C8	> =
155	9B	(l)	201	C9	<>
156	9C	(m)	202	CA	LINE
157	9D	(n)	203	CB	THEN
158	9E	(o)	204	CC	TO
159	9F	(p)	205	CD	STEP
160	A0	(q)	206	CE	DEF FN
161	A1	(r)	207	CF	CAT
162	A2	(s)	208	D0	FORMAT
163	A3	(t)	209	D1	MOVE
164	A4	(u)	210	D2	ERASE
165	A5	RND	211	D3	OPEN #
166	A6	INKEY\$	212	D4	CLOSE #
167	A7	PI	213	D5	MERGE
168	A8	FN	214	D6	VERIFY
169	A9	POINT	215	D7	BEEP
170	AA	SCREEN\$	216	D8	CIRCLE
171	AB	ATTR	217	D9	INK
172	AC	AT	218	DA	PAPER
173	AD	TAB	219	DB	FLASH
174	AE	VAL\$	220	DC	BRIGHT
175	AF	CODE	221	DD	INVERSE
176	B0	VAL	222	DE	OVER
177	B1	LEN	223	DF	OUT
178	B2	SIN	224	E0	LPRINT
179	B3	COS	225	E1	LLIST
180	B4	TAN	226	E2	STOP
181	B5	ASN	227	E3	READ
182	B6	ACS	228	E4	DATA
183	B7	ATN	229	E5	RESTORE
184	B8	LN	230	E6	NEW
185	B9	EXP	231	E7	BORDER
186	BA	INT	232	E8	CONTINUE
187	BB	SQR	233	E9	DIM
188	BC	SGN	234	EA	REM
189	BD	ABS	235	EB	FOR

Dec.	Hexa.	Caractères
236	EC	GO TO
237	ED	GO SUB
238	EE	INPUT
239	EF	LOAD
240	F0	LIST
241	F1	LET
242	F2	PAUSE
243	F3	NEXT
244	F4	POKE
245	F5	PRINT
246	F6	PLOT
247	F7	RUN
248	F8	SAVE
249	F9	RANDOMIZE
250	FA	IF
251	FB	CLS
252	FC	DRAW
253	FD	CLEAR
254	FE	RETURN
255	FF	COPY

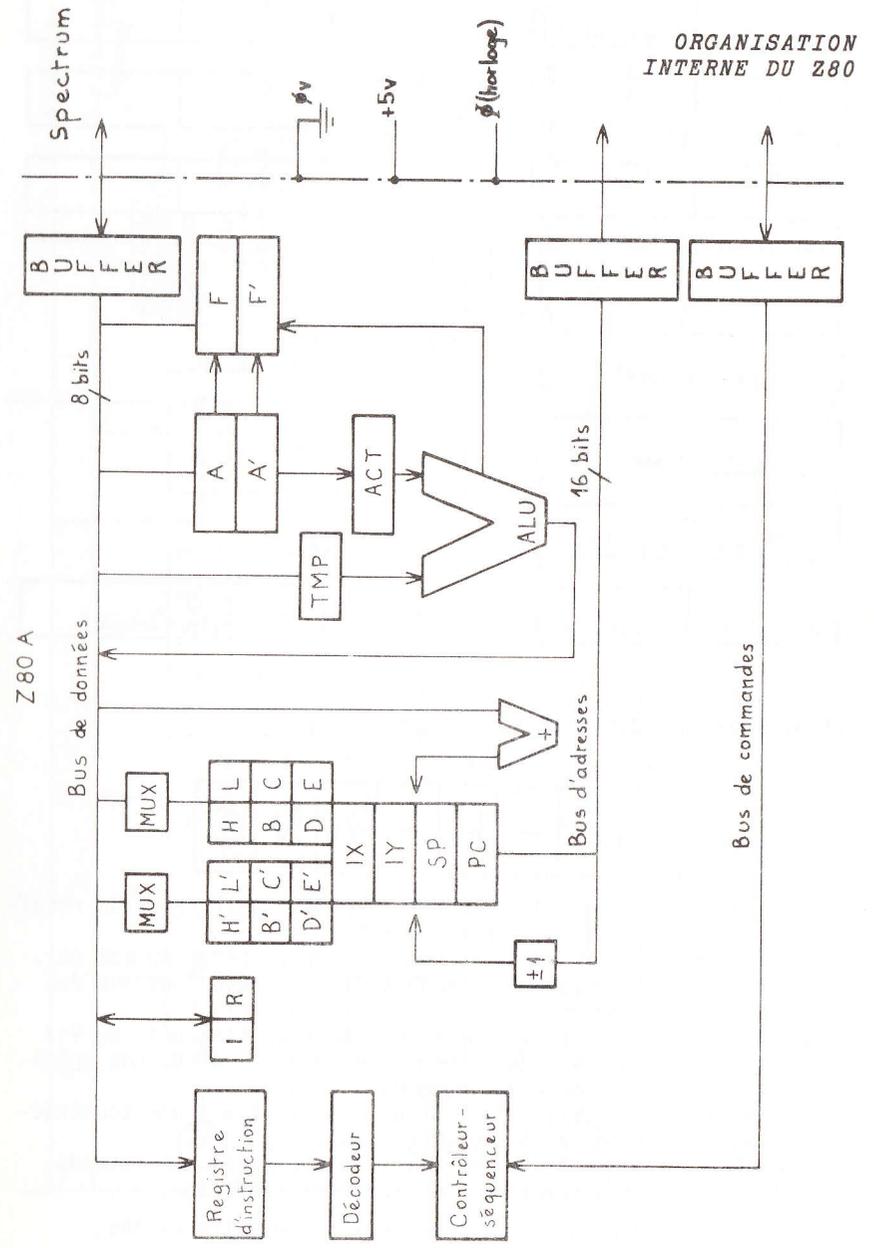
La grille ci-dessous vous donne la position des caractères sur l'écran (PRINT AT) ainsi que celle des pixels (PLOT).

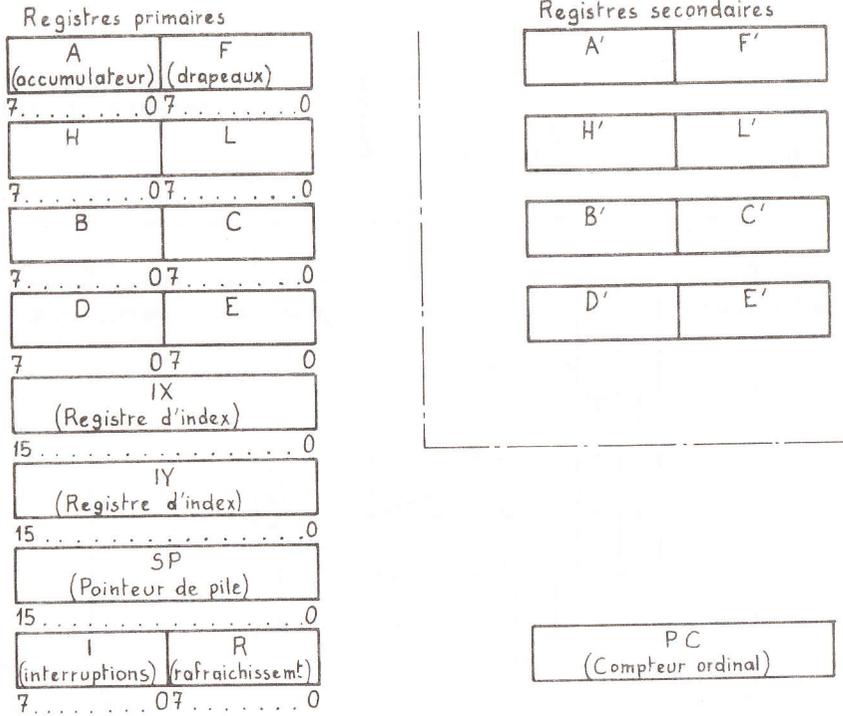


SCHEMA SIMPLIFIE DU SPECTRUM

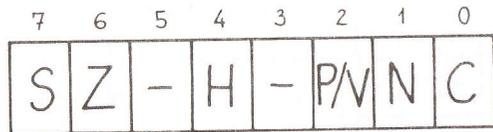
# LANGAGE MACHINE

# LANGAGE MACHINE



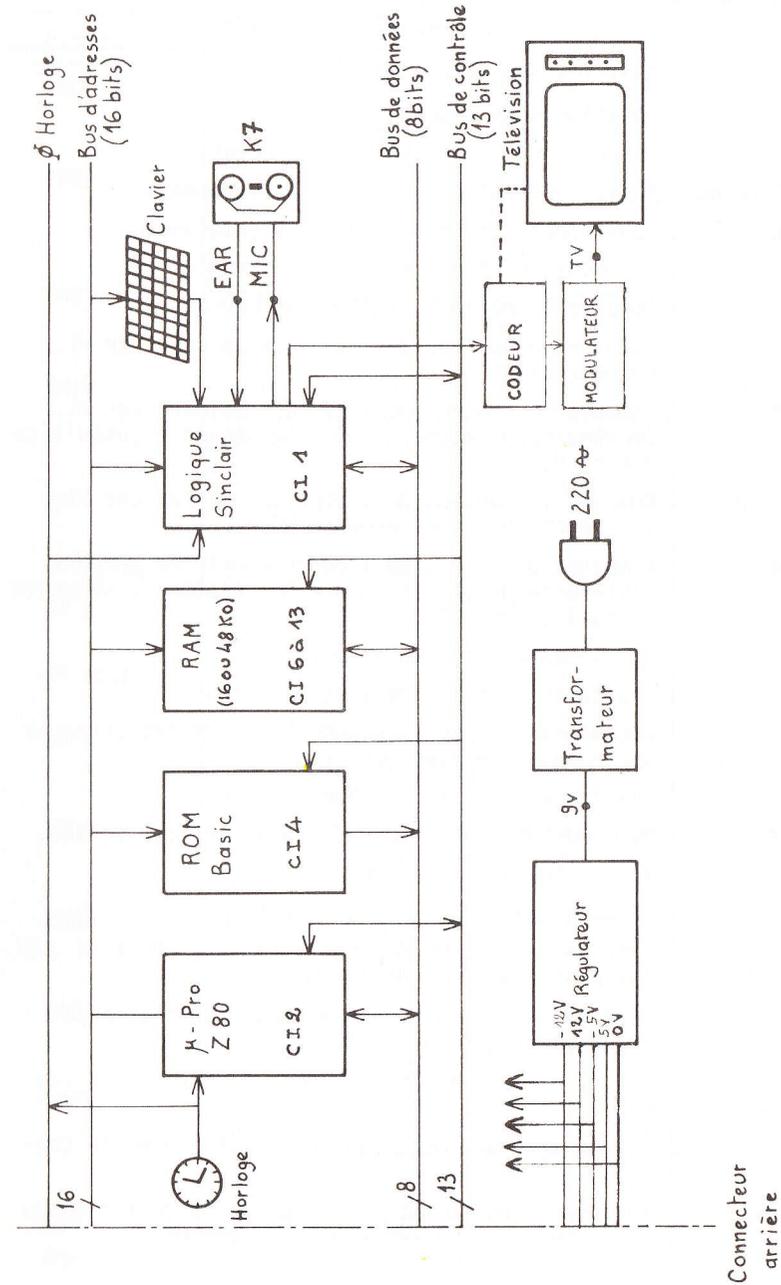


Détails du registre F (flag = drapeau)



- S = Signe** : passe à 1 si le bit le plus significatif du résultat d'une opération est 1.
- Z = Zéro** : passe à 1 si le résultat de l'opération est nul.
- H = Demi** : identique à C, mais pour les opérations sur des demi-octets.
- P/V = Parité/dépassement** : P=1 s'il y a un nombre pair de bits à 1, ou V=1 s'il y a un dépassement de capacité après une opération avec des nombres signés.
- N = Opération précédente** : N=1 si l'opération précédente était une soustraction, et N=0 pour une addition.
- C = Retenue** : passe à 1 si le résultat nécessite une retenue (soustraction) ou un report (addition).

Remarque : les drapeaux H et N ne peuvent pas être testés.



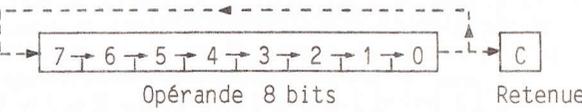
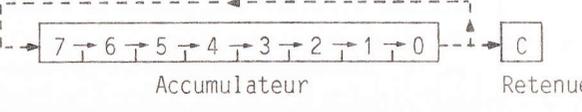
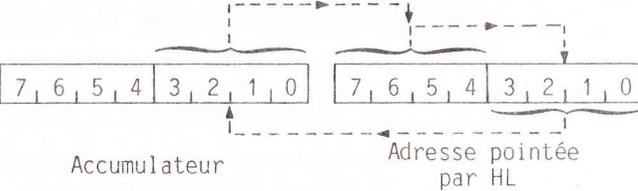
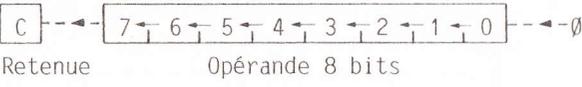
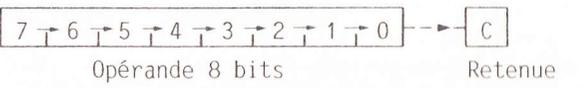
LANGAGE MACHINE

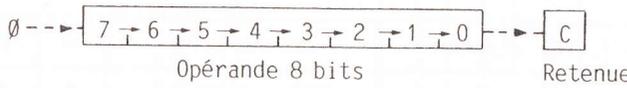
Mnémonique	Opération effectuée
ADC	Addition avec retenue.
ADD	Addition sans retenue.
BIT	Teste un bit particulier d'un octet.
CALL cc,nn	Appel conditionnel d'un sous-programme.
CALL	Appel inconditionnel d'un sous-programme.
CCF	Complémente l'indicateur de retenue.
CP	Compare l'opérande et l'accumulateur.
CPD	Compare le contenu de l'adresse pointée par HL. Décrémte HL et BC.
CPDR	Compare le contenu de l'adresse pointée par HL. Décrémte HL et BC. Répète la séquence jusqu'à ce que BC=0.
CPI	Compare le contenu de l'adresse pointée par HL. Incrémte HL et décrémte BC.
CPIR	Compare le contenu de l'adresse pointée par HL. Incrémte HL et décrémte BC. Répète la séquence jusqu'à ce que BC=0.
CPL	Complémente l'accumulateur.
DAA	Ajustement décimal de l'accumulateur.
DEC	Décrémte un registre, une paire de registres ou une adresse pointée par HL.
DI	Désactive les interruptions.
DJNZ	Décrémte B et effectue un saut relatif si B≠0.
EI	Active les interruptions.
EX	Echange les contenus des registres.
EXX	Echange le contenu des registres BC, DE et HL avec les registres BC', DE' et HL'.
HALT	Place le microprocesseur en position d'attente d'une interruption ou d'un reset.
IM	Positionne un des trois modes d'interruption (de 0 à 2).
IN	Charge l'accumulateur ou un registre avec le contenu d'un port d'entrées/sorties.
INC	Incrémte un registre, une paire de registres ou le contenu de l'adresse pointée par HL.

Mnémonique	Opération effectuée
IND	Charge l'adresse pointée par HL avec le contenu du port d'entrées/sorties pointé par C, et décrémte HL et B.
INDR	Charge l'adresse pointée par HL avec le contenu du port d'entrées/sorties pointé par C, et décrémte HL et B. Répète la séquence jusqu'à ce que B=0.
INI	Charge l'adresse pointée par HL avec le contenu du port d'entrées/sorties pointé par C, et incrémente HL et décrémte B.
INIR	Charge l'adresse pointée par HL avec le contenu du port d'entrées/sorties pointé par C, et incrémente HL et décrémte B. Répète la séquence jusqu'à ce que B=0.
JP	Saut inconditionnel à l'adresse donnée ou à celle pointée par HL, IX et IY.
JP cc,aa	Saut conditionnel (cc) à l'adresse donnée (aa).
JR e	Saut inconditionnel relatif à PC plus déplacement (e).
JR cc,e	Saut conditionnel (cc) relatif à PC plus déplacement (e).
LD	Charge l'accumulateur, un registre ou une adresse avec le contenu de l'accumulateur, un registre ou une adresse.
LDD	Charge l'adresse pointée par HL avec le contenu de l'adresse pointée par DE, puis décrémte DE, HL et BC.
LDDR	Charge l'adresse pointée par HL avec le contenu de l'adresse pointée par DE, puis décrémte DE, HL et BC. Répète la séquence jusqu'à ce que BC=0.
LDI	Charge l'adresse pointée par HL avec le contenu de l'adresse pointée par DE, puis incrémente DE et HL et décrémte BC.
LDIR	Charge l'adresse pointée par HL avec le contenu de l'adresse pointée par DE, puis incrémente DE et HL et décrémte BC. Répète la séquence jusqu'à ce que BC=0.
NEG	Inverse le signe de l'accumulateur.
NOP	Le Z80 n'effectue pas d'instruction.

Mnémonique	Opération effectuée
OR	"Ou" logique entre l'opérande et l'accumulateur.
OTDR	Charge le port d'entrées/sorties pointé par C avec le contenu de l'adresse pointée par HL, puis décrémente HL et B. Répète la séquence jusqu'à ce que B=0.
OTIR	Charge le port d'entrées/sorties pointé par C avec le contenu de l'adresse pointée par HL, puis incrémente HL et décrémente B. Répète la séquence jusqu'à ce que B=0.
OUT	Charge le port d'entrées/sorties spécifié avec le contenu de l'accumulateur.
OUTD	Charge le port d'entrées/sorties pointé par C avec le contenu de l'adresse pointée par HL, puis décrémente HL et B.
OUTI	Charge le port d'entrées/sorties pointé par C avec le contenu de l'adresse pointée par HL, puis incrémente HL et décrémente B.
POP	Charge une paire de registres ou un index avec la dernière valeur de la pile (pointée par SP).
PUSH	Place le contenu d'une paire de registres ou d'un index dans la pile (pointée par SP).
RES	Place 0 dans le bit spécifié de l'opérande.
RET	Retour d'un sous-programme.
RETI	Retour d'un sous-programme d'interruption.
RETN	Retour d'un sous-programme d'interruption non masquable.
RL	Rotation à gauche à travers la retenue de l'opérande.
RLA	Rotation à gauche à travers la retenue de l'accumulateur.

Mnémonique	Opération effectuée
RLC	Rotation circulaire à gauche du contenu d'un registre ou d'une adresse pointée par HL ou par IX et IY plus déplacement.
RLCA	Rotation circulaire à gauche de l'accumulateur.
RLD	Rotation BCD (binaire codé décimal) de 4 bits à gauche entre l'accumulateur et l'adresse pointée par HL.
RR	Rotation à droite à travers la retenue de l'opérande.
RRA	Rotation à droite à travers la retenue de l'accumulateur.

Mnémonique	Opération effectuée
RRC	Rotation circulaire à droite de l'opérande. 
RCA	Rotation circulaire à droite de l'accumulateur. 
RRD	Rotation BCD à droite de 4 bits entre l'accumulateur et le contenu de l'adresse pointée par HL. 
RST	Saut à une adresse sur un octet.
SBC	Soustraction avec retenue entre l'accumulateur et l'opérande ou HL et une paire de registres.
SCF	Met à 1 le drapeau de retenue.
SET	Met à 1 un bit particulier d'un registre ou d'une adresse pointée par HL ou IX et IY plus déplacement.
SLA	Décalage arithmétique à gauche de l'opérande. 
SRA	Décalage arithmétique à droite de l'opérande.  <p>NB : le bit 7 reste inchangé.</p>

Mnémonique	Opération effectuée
SRL	Décalage logique à droite de l'opérande. 
SUB	Soustraction à l'accumulateur de l'opérande.
XOR	"Ou" exclusif entre l'opérande et l'accumulateur.

- d = donnée sur 8 bits
- dd = donnée sur 16 bits
- aa = adresse sur 16 bits
- ● = drapeau modifié
- 0 = drapeau à 0
- 1 = drapeau à 1

Code objet	Instruction	S	Z	P/V	C
8E	ADC A, (HL)	●	●	●	●
DD8Ed	ADC A, (IX+d)	●	●	●	●
FD8Ed	ADC A, (IY+d)	●	●	●	●
8F	ADC A,A	●	●	●	●
88	ADC A,B	●	●	●	●
89	ADC A,C	●	●	●	●
8A	ADC A,D	●	●	●	●
8B	ADC A,E	●	●	●	●
8C	ADC A,H	●	●	●	●
8D	ADC A,L	●	●	●	●
CEd	ADC A,d	●	●	●	●
ED4A	ADC HL,BC	●	●	●	●
ED5A	ADC HL,DE	●	●	●	●
ED6A	ADC HL,HL	●	●	●	●
ED7A	ADC HL,SP	●	●	●	●
86	ADD A, (HL)	●	●	●	●
DD86d	ADD A, (IX+d)	●	●	●	●
FD86d	ADD A, (IY+d)	●	●	●	●
87	ADD A,A	●	●	●	●
80	ADD A,B	●	●	●	●
81	ADD A,C	●	●	●	●
82	ADD A,D	●	●	●	●
83	ADD A,E	●	●	●	●
84	ADD A,H	●	●	●	●
85	ADD A,L	●	●	●	●
C6d	ADD A,d	●	●	●	●
09	ADD HL,BC	●	●	●	●
19	ADD HL,DE	●	●	●	●
29	ADD HL,HL	●	●	●	●
39	ADD HL,SP	●	●	●	●
DD09	ADD IX,BC	●	●	●	●
DD19	ADD IX,DE	●	●	●	●
DD29	ADD IX,IX	●	●	●	●
DD39	ADD IX,SP	●	●	●	●
FD09	ADD IY,BC	●	●	●	●
FD19	ADD IY,DE	●	●	●	●
FD29	ADD IY,IY	●	●	●	●
FD39	ADD IY,SP	●	●	●	●
A6	AND (HL)	●	●	●	0
DDA6d	AND (IX+d)	●	●	●	0
FDA6d	AND (IY+d)	●	●	●	0
A7	AND A	●	●	●	0

Code objet	Instruction	S	Z	P/V	C
A0	AND B	●	●	●	0
A1	AND C	●	●	●	0
A2	AND D	●	●	●	0
A3	AND E	●	●	●	0
A4	AND H	●	●	●	0
A5	AND L	●	●	●	0
E6d	AND d	●	●	●	0
CB46	BIT 0, (HL)	●	●	●	0
DDCBd46	BIT 0, (IX+d)	●	●	●	0
FDCBd46	BIT 0, (IY+d)	●	●	●	0
CB47	BIT 0,A	●	●	●	0
CB40	BIT 0,B	●	●	●	0
CB41	BIT 0,C	●	●	●	0
CB42	BIT 0,D	●	●	●	0
CB43	BIT 0,E	●	●	●	0
CB44	BIT 0,H	●	●	●	0
CB45	BIT 0,L	●	●	●	0
CB4E	BIT 1 (HL)	●	●	●	0
DDCBd4E	BIT 1, (IX+d)	●	●	●	0
FDCBd4E	BIT 1, (IY+d)	●	●	●	0
CB4F	BIT 1,A	●	●	●	0
CB48	BIT 1,B	●	●	●	0
CB49	BIT 1,C	●	●	●	0
CB4A	BIT 1,D	●	●	●	0
CB4B	BIT 1,E	●	●	●	0
CB4C	BIT 1,H	●	●	●	0
CB4D	BIT 1,L	●	●	●	0
CB56	BIT 2, (HL)	●	●	●	0
DDCBd56	BIT 2, (IX+d)	●	●	●	0
FDCBd56	BIT 2, (IY+d)	●	●	●	0
CB57	BIT 2,A	●	●	●	0
CB50	BIT 2,B	●	●	●	0
CB51	BIT 2,C	●	●	●	0
CB52	BIT 2,D	●	●	●	0
CB53	BIT 2,E	●	●	●	0
CB54	BIT 2,H	●	●	●	0
CB55	BIT 2,L	●	●	●	0
CB5E	BIT 3, (HL)	●	●	●	0
DDCBd5E	BIT 3, (IX+d)	●	●	●	0
FDCBd5E	BIT 3, (IY+d)	●	●	●	0
CB5F	BIT 3,A	●	●	●	0
CB58	BIT 3,B	●	●	●	0
CB59	BIT 3,C	●	●	●	0
CB5A	BIT 3,D	●	●	●	0
CB5B	BIT 3,E	●	●	●	0
CB5C	BIT 3,H	●	●	●	0
CB5D	BIT 3,L	●	●	●	0

Code objet	Instruction	S	Z	P/V	C
CB66	BIT 4,(HL)	•	•	•	
DDCBd66	BIT 4,(IX+d)	•	•	•	
FDCBd66	BIT 4,(IY+d)	•	•	•	
CB67	BIT 4,A	•	•	•	
CB60	BIT 4,B	•	•	•	
CB61	BIT 4,C	•	•	•	
CB62	BIT 4,D	•	•	•	
CB63	BIT 4,E	•	•	•	
CB64	BIT 4,H	•	•	•	
CB65	BIT 4,L	•	•	•	
CB6E	BIT 5,(HL)	•	•	•	
DDCBd6E	BIT 5,(IX+d)	•	•	•	
FDCBd6E	BIT 5,(IY+d)	•	•	•	
CB6F	BIT 5,A	•	•	•	
CB68	BIT 5,B	•	•	•	
CB69	BIT 5,C	•	•	•	
C86A	BIT 5,D	•	•	•	
CB6B	BIT 5,E	•	•	•	
CB6C	BIT 5,H	•	•	•	
CB6D	BIT 5,L	•	•	•	
DB76	BIT 6,(HL)	•	•	•	
DDCBd76	BIT 6,(IX+d)	•	•	•	
FDCBd76	BIT 6,(IY+d)	•	•	•	
CB77	BIT 6,A	•	•	•	
CB70	BIT 6,B	•	•	•	
CB71	BIT 6,C	•	•	•	
CB72	BIT 6,D	•	•	•	
CB73	BIT 6,E	•	•	•	
CB74	BIT 6,H	•	•	•	
CB75	BIT 6,L	•	•	•	
CB7E	BIT 7,(HL)	•	•	•	
DDCBd7E	BIT 7,(IX+d)	•	•	•	
FDCBd7E	BIT 7,(IY+d)	•	•	•	
CB7F	BIT 7,A	•	•	•	
CB78	BIT 7,B	•	•	•	
CB79	BIT 7,C	•	•	•	
CB7A	BIT 7,D	•	•	•	
CB7B	BIT 7,E	•	•	•	
CB7C	BIT 7,H	•	•	•	
CB7D	BIT 7,L	•	•	•	
DCaa	CALL C,aa				
FCaa	CALL M,aa				
D4aa	CALL NC,aa				
C4aa	CALL NZ,aa				
F4aa	CALL P,aa				
ECaa	CALL PE,aa				

Code objet	Instruction	S	Z	P/V	C
E4aa	CALL PO,aa				
CCaa	CALL Z;aa				
CDaa	CALL aa				
3F	CCF				•
BE	CP (HL)	•	•	•	•
DDBBEd	CP (IX+d)	•	•	•	•
FDBEd	CP (IY+d)	•	•	•	•
BF	CP A	•	•	•	•
B8	CP B	•	•	•	•
B9	CP C	•	•	•	•
BA	CP D	•	•	•	•
BB	CP E	•	•	•	•
BC	CP H	•	•	•	•
BD	CP L	•	•	•	•
FEd	CP d	•	•	•	•
EDA9	CPD	•	•	•	•
EDB9	CPDR	•	•	•	•
EDB1	CPDR	•	•	•	•
EDA1	CPI	•	•	•	•
2F	CPL				•
27	DAA	•	•	•	•
35	DEC (HL)	•	•	•	•
DD35d	DEC (IX+d)	•	•	•	•
FD35d	DEC (IY+d)	•	•	•	•
3D	DEC A	•	•	•	•
05	DEC B	•	•	•	•
0B	DEC BC	•	•	•	•
0D	DEC C	•	•	•	•
15	DEC D	•	•	•	•
1B	DEC DE	•	•	•	•
1D	DEC E	•	•	•	•
25	DEC H	•	•	•	•
2B	DEC HL	•	•	•	•
DD2B	DEC IX	•	•	•	•
FD2B	DEC IY	•	•	•	•
2D	DEC L	•	•	•	•
3B	DEC SP	•	•	•	•
F3	DI				•
10d	DJNZ d				•
FB	EI				•
E3	EX (SP),HL				•
DDE3	EX (SP),IX				•
FDE3	EX (SP),IY				•
08	EX AF,AF'				•
EB	EX DE,HL				•
D9	EXX				•

LANGAGE MACHINE

Code objet	Instruction	S	Z	P/V	C
76	HALT				
ED46	IM 0				
ED56	IM 1				
ED5E	IM 2				
ED78	IN A,(C)	•	•	•	
ED40	IN B,(C)	•	•	•	
ED48	IN C,(C)	•	•	•	
ED50	IN D,(C)	•	•	•	
ED58	IN E,(C)	•	•	•	
ED60	IN H,(C)	•	•	•	
ED68	IN L,(C)	•	•	•	
DBd	IN A,(d)				
34	INC (HL)	•	•	•	
DD34d	INC (IX+d)	•	•	•	
FD34d	INC (IY+d)	•	•	•	
3C	INC A	•	•	•	
04	INC B	•	•	•	
03	INC BC				
0C	INC C	•	•	•	
14	INC D	•	•	•	
13	INC DE				
1C	INC E	•	•	•	
24	INC H	•	•	•	
23	INC HL				
DD23	INC IX				
FD23	INC IY				
2C	INC L	•	•	•	
33	INC SP				
EDAA	IND	•	•	•	
EDBA	INDR	•	•	•	
EDA2	INI	•	•	•	
EDB2	INIR	•	•	•	
C3aa	JP aa				
E9	JP (HL)				
DDE9	JP (IX)				
FDE9	JP (IY)				
DAdd	JP C,aa				
FAaa	JP M,aa				
D2aa	JP NC,aa				
C2aa	JP NZ,aa				
F2aa	JP P,aa				
EAaa	JP PE,aa				
E2aa	JP PO,aa				
CAaa	JP Z,aa				
38d	JR C,d				
30d	JR NC,d				
20d	JR NZ,d				

Code objet	Instruction	S	Z	P/V	C
28d	JR Z,d				
18d	JR d				
02	LD (BC),A				
12	LD (DE),A				
77	LD (HL),A				
70	LD (HL),B				
71	LD (HL),C				
72	LD (HL),D				
73	LD (HL),E				
74	LD (HL),H				
75	LD (HL),L				
36d	LD (HL),d				
DD77d	LD (IX+d),A				
DD70d	LD (IX+d),B				
DD71d	LD (IX+d),C				
DD72d	LD (IX+d),D				
DD73d	LD (IX+d),E				
DD74d	LD (IX+d),H				
DD75d	LD (IX+d),L				
DD36d20	LD (IX+d),d				
FD77d	LD (IY+d),A				
FD70d	LD (IY+d),B				
FD71d	LD (IY+d),C				
FD72d	LD (IY+d),D				
FD73d	LD (IY+d),E				
FD74d	LD (IY+d),H				
FD75d	LD (IY+d),L				
FD36d20	LD (IY+d),d				
32dd	LD (dd),A				
ED43dd	LD (dd),BC				
ED53dd	LD (dd),DE				
22dd	LD (dd),HL				
DD22dd	LD (dd),IX				
FD22dd	LD (dd),IY				
ED73dd	LD (dd),SP				
0A	LD A,(BC)				
1A	LD A,(DE)				
7E	LD A,(HL)				
DD7Ed	LD A,(IX+d)				
FD7Ed	LD A,(IY+d)				
3Add	LD A,(dd)				
7F	LD A,A				
78	LD A,B				
79	LD A,C				
7A	LD A,D				
7B	LD A,E				
7C	LD A,H				

Code objet	Instruction	S	Z	P/V	C
ED57	LD A,I	•	•	•	
7D	LD A,L				
3E	LD A,d				
ED5F	LD A,R	•	•	•	
46	LD B,(HL)				
DD46d	LD B,(IX+d)				
FD46d	LD B,(IY+d)				
47	LD B,A				
40	LD B,B				
41	LD B,C				
42	LD B,D				
43	LD B,E				
44	LD B,H				
45	LD B,L				
06d	LD B,d				
ED4Bdd	LD BC,(dd)				
01dd	LD BC,dd				
4E	LD C,(HL)				
DD4Ed	LD C,(IX+d)				
FD4Ed	LD C,(IY+d)				
4F	LD C,A				
48	LD C,B				
49	LD C,C				
4A	LD C,D				
4B	LD C,E				
4C	LD C,H				
4D	LD C,L				
0Ed	LD C,d				
56	LD D,(HL)				
DD56d	LD D,(IX+d)				
FD56d	LD D,(IY+d)				
57	LD D,A				
50	LD D,B				
51	LD D,C				
52	LD D,D				
53	LD D,E				
54	LD D,H				
55	LD D,L				
16d	LD D,d				
ED5Bdd	LD DE,(dd)				
11dd	LD DE,dd				
5E	LD E,(HL)				
DD5Ed	LD E,(IX+d)				
FD5Ed	LD E,(IY+d)				
5F	LD E,A				
58	LD E,B				
59	LD E,C				

Code objet	Instruction	S	Z	P/V	C
5A	LD E,D				
5B	LD E,E				
5C	LD E,H				
5D	LD E,L				
1E20	LD E,n				
66	LD H,(HL)				
DD66d	LD H,(IX+d)				
FD66d	LD H,(IY+d)				
67	LD H,A				
60	LD H,B				
61	LD H,C				
62	LD H,D				
63	LD H,E				
64	LD H,H				
65	LD H,L				
26d	LD H,d				
2Add	LD HL,(dd)				
21dd	LD HL,dd				
ED47	LD I,A				
DD2Add	LD IX,(dd)				
DD21dd	LD IX,dd				
FD2Add	LD IY,(dd)				
FD21dd	LD IY,dd				
6E	LD L,(HL)				
DD6Ed	LD L,(IX+d)				
FD6Ed	LD L,(IY+d)				
6F	LD L,A				
68	LD L,B				
69	LD L,C				
6A	LD L,D				
6B	LD L,E				
6C	LD L,H				
6D	LD L,L				
2Ed	LD L,d				
ED4F	LD R,A				
ED7Bdd	LD SP,(dd)				
F9	LD SP,HL				
DDF9	LD SP,IX				
PDF9	LD SP,IY				
31dd	LD SP,dd				
EDA8	LDD				•
EDB8	LDDR				•
EDA0	LDI				•
EDB0	LDIR				•
ED44	NEG				
00	NOP				
B6	OR (HL)	•	•	•	•

Code objet	Instruction	S	Z	P/V	C
DDB6d	OR (IX+d)	•	•	•	0
FDB6d	OR (IY+d)	•	•	•	0
B7	OR A	•	•	•	0
B0	OR B	•	•	•	0
B1	OR C	•	•	•	0
B2	OR D	•	•	•	0
B3	OR E	•	•	•	0
B4	OR H	•	•	•	0
B5	OR L	•	•	•	0
F6d	OR d	•	•	•	0
EDBB	OTDR	•	•	•	
EDB3	OTIR	•	•	•	
ED79	OUT (C),A				
ED41	OUT (C),B				
ED49	OUT (C),C				
ED51	OUT (C),D				
ED59	OUT (C),E				
ED61	OUT (C),H				
ED69	OUT (C),L				
D3d	OUT (d),A				
EDAB	OUTD	•	•	•	
EDA3	OUTI	•	•	•	
F1	POP AF				
C1	POP BC				
D1	POP DE				
E1	POP HL				
DDE1	POP IX				
FDE1	POP IY				
F5	PUSH AF				
C5	PUSH BC				
D5	PUSH DE				
E5	PUSH HL				
DDE5	PUSH IX				
FDE5	PUSH IY				
CB86	RES 0, (HL)				
DDCBd86	RES 0, (IX+d)				
FDCBd86	RES 0, (IY+d)				
CB87	RES 0,A				
CB80	RES 0,B				
CB81	RES 0,C				
CB82	RES 0,D				
CB83	RES 0,E				
CB84	RES 0,H				
CB85	RES 0,L				
CB8E	RES 1, (HL)				
DDCBd8E	RES 1, (IX+d)				

Code objet	Instruction	S	Z	P/V	C
FDCBd8E	RES 1, (IY+d)				
CB8F	RES 1,A				
CB88	RES 1,B				
CB89	RES 1,C				
CB8A	RES 1,D				
CB8B	RES 1,E				
CB8C	RES 1,H				
CB8D	RES 1,L				
CB96	RES 2, (HL)				
DDCBd96	RES 2, (IX+d)				
FDCBd96	RES 2, (IY+d)				
CB97	RES 2,A				
CB90	RES 2,B				
CB91	RES 2,C				
CB92	RES 2,D				
CB93	RES 2,E				
CB94	RES 2,H				
CB95	RES 2,L				
CB9E	RES 3, (HL)				
DDCBd9E	RES 3, (IX+d)				
FDCBd9E	RES 3, (IY+d)				
CB9F	RES 3,A				
CB98	RES 3,B				
CB99	RES 3,C				
CB9A	RES 3,D				
CB9B	RES 3,E				
CB9C	RES 3,H				
CB9D	RES 3,L				
CBA6	RES 4, (HL)				
DDCBdA6	RES 4, (IX+d)				
FDCBdA7	RES 4, (IY+d)				
CBA7	RES 4,A				
CBA0	RES 4,B				
CBA1	RES 4,C				
CBA2	RES 4,D				
DBA3	RES 4,E				
CBA4	RES 4,H				
CBA5	RES 4,L				
CBAE	RES 5, (HL)				
DDCBdAE	RES 5, (IX+d)				
FDCBdAE	RES 5, (IY+d)				
CBAF	RES 5,A				
CBA8	RES 5,B				
CBA9	RES 5,C				
CBAA	RES 5,D				
CBAB	RES 5,E				
CBAC	RES 5,H				

Code objet	Instruction	S	Z	P/V	C
CBAD	RES 5,L				
CBB6	RES 6,(HL)				
DDCBdB6	RES 6,(IX+d)				
FDCBdB6	RES 6,(IY+d)				
CBB7	RES 6,A				
CBB0	RES 6,B				
CBB1	RES 6,C				
CBB2	RES 6,D				
CBB3	RES 6,E				
CBB4	RES 6,H				
CBB5	RES 6,L				
CBBE	RES 7,(HL)				
DDCBdBE	RES 7,(IX+d)				
FDCBdBE	RES 7,(IY+d)				
CBBF	RES 7,A				
CBB8	RES 7,B				
CBB9	RES 7,C				
CBBA	RES 7,D				
CBBB	RES 7,E				
CBBC	RES 7,H				
CBBD	RES 7,L				
C9	RET				
D8	RET C				
F8	RET M				
DO	RET NC				
CO	RET NZ				
FO	RET P				
E8	RET PE				
EO	RET PO				
C8	RET Z				
ED4D	RETI				
ED45	RETN				
CB16	RL (HL)	•	•	•	•
DDCBd16	RL (IX+d)	•	•	•	•
FDCBd16	RL (IY+d)	•	•	•	•
CB17	RL A	•	•	•	•
CB10	RL B	•	•	•	•
CB11	RL C	•	•	•	•
CB12	RL D	•	•	•	•
CB13	RL E	•	•	•	•
CB14	RL H	•	•	•	•
CB15	RL L	•	•	•	•
17	RLA				
CB06	RLC (HL)	•	•	•	•
DDCBd06	RLC (IX+d)	•	•	•	•
FDCBd06	RLC (IY+d)	•	•	•	•

Code objet	Instruction	S	Z	P/V	C
CB07	RLC A	•	•	•	•
CB00	RLC B	•	•	•	•
CB01	RLC C	•	•	•	•
CB02	RLC D	•	•	•	•
CB03	RLC E	•	•	•	•
CB04	RLC H	•	•	•	•
CB05	RLC L	•	•	•	•
07	RLCA				•
ED6F	RLD	•	•	•	
CB1E	RR (HL)	•	•	•	•
DDCBd1E	RR (IX+d)	•	•	•	•
FDCBd1E	RR (IY+d)	•	•	•	•
CB1F	RR A	•	•	•	•
CB18	RR B	•	•	•	•
CB19	RR C	•	•	•	•
CB1A	RR D	•	•	•	•
CB1B	RR E	•	•	•	•
CB1C	RR H	•	•	•	•
CB1D	RR L	•	•	•	•
1F	RRA				•
CB0E	RRC (HL)	•	•	•	•
DDCBd0E	RRC (IX+d)	•	•	•	•
FDCBd0E	RRC (IY+d)	•	•	•	•
CB0F	RRC A	•	•	•	•
CB08	RRC B	•	•	•	•
CB09	RRC C	•	•	•	•
CB0A	RRC D	•	•	•	•
CB0B	RRC E	•	•	•	•
CB0C	RRC H	•	•	•	•
CB0D	RRC L	•	•	•	•
0F	RRCA				•
ED67	RRD	•	•	•	
C7	RST 00H				
CF	RST 08H				
D7	RST 10H				
DF	RST 18H				
E7	RST 20H				
EF	RST 28H				
F7	RST 30H				
FF	RST 38H				
DEd	SBC A,d	•	•	•	•
9E	SBC A,(HL)	•	•	•	•
DD9Ed	SBC A,(IX+d)	•	•	•	•
FD9Ed	SBC A,(IY+d)	•	•	•	•
9F	SBC A,A	•	•	•	•
98	SBC A,B	•	•	•	•

Code objet	Instruction	S	Z	P/V	C
99	SBC A,C	●	●	●	●
9A	SBC A,D	●	●	●	●
9B	SBC A,E	●	●	●	●
9C	SBC A,H	●	●	●	●
9D	SBC A,L	●	●	●	●
ED42	SBC HL,BC	●	●	●	●
ED52	SBC HL,DE	●	●	●	●
ED62	SBC HL,HL*	●	●	●	●
ED72	SBC HL,SP	●	●	●	●
37	SCF				1
CBC6	SET 0,(HL)				
DDCBdC6	SET 0,(IX+d)				
FDCBdC6	SET 0,(IY+d)				
CBC7	SET 0,A				
CBC0	SET 0,B				
CBC1	SET 0,C				
CBC2	SET 0,D				
CBC3	SET 0,E				
CBC4	SET 0,H				
CBC5	SET 0,L				
CBCE	SET 1,(HL)				
ddCBdCE	SET 1,(IX+d)				
FDCBdCE	SET 1,(IY+d)				
CBCF	SET 1,A				
CBC8	SET 1,B				
CBC9	SET 1,C				
CBCA	SET 1,D				
CBCB	SET 1,E				
CBCC	SET 1,H				
CBCD	SET 1,L				
CBD6	SET 2,(HL)				
DDCBdD6	SET 2,(IX+d)				
FDCBdD6	SET 2,(IY+d)				
CBD7	SET 2,A				
CBD0	SET 2,B				
CBD1	SET 2,C				
CBD2	SET 2,D				
CBD3	SET 2,E				
CBD4	SET 2,H				
CBD5	SET 2,L				
CBDE	SET 3,(HL)				
DDCBdDE	SET 3,(IX+d)				
FDCBdDE	SET 3,(IY+d)				
CBDF	SET 3,A				
CBD8	SET 3,B				
CBD9	SET 3,C				
CBDA	SET 3,D				

Code objet	Instruction	S	Z	P/V	C
CBDB	SET 3,E				
CBDC	SET 3,H				
CBDD	SET 3,L				
CBE6	SET 4,(HL)				
DDCBdE6	SET 4,(IX+d)				
FDCBdE6	SET 4,(IY+d)				
CBE7	SET 4,A				
CBE0	SET 4,B				
CBE1	SET 4,C				
CBE2	SET 4,D				
CBE3	SET 4,E				
CBE4	SET 4,H				
CBE5	SET 4,L				
CBEE	SET 5,(HL)				
DDCBdEE	SET 5,(IX+d)				
FDCBdEE	SET 5,(IY+d)				
CBEF	SET 5,A				
CBE8	SET 5,B				
CBE9	SET 5,C				
CBEA	SET 5,D				
CBEB	SET 5,E				
CBEC	SET 5,H				
CBED	SET 5,L				
CBF6	SET 6,(HL)				
DDCBdF6	SET 6,(IX+d)				
FDCBdF6	SET 6,(IY+d)				
CBF7	SET 6,A				
CBF0	SET 6,B				
CBF1	SET 6,C				
CBF2	SET 6,D				
CBF3	SET 6,E				
CBF4	SET 6,H				
CBF5	SET 6,L				
CBFE	SET 7,(HL)				
DDCBdFE	SET 7,(IX+d)				
FDCBdFE	SET 7,(IY+d)				
CBFF	SET 7,A				
CBF8	SET 7,B				
CBF9	SET 7,C				
CBFA	SET 7,D				
CBFB	SET 7,E				
CBFC	SET 7,H				
CBFD	SET 7,L				
CB26	SLA (HL)	●	●	●	●
DDCBd26	SLA (IX+d)	●	●	●	●
FDCBd26	SLA (IY+d)	●	●	●	●
CB27	SLA A	●	●	●	●
CB20	SLA B	●	●	●	●

LANGAGE MACHINE

Code objet	Instruction	S	Z	P/V	C
CB21	SLA C	●	●	●	●
CB22	SLA D	●	●	●	●
CB23	SLA E	●	●	●	●
CB24	SLA H	●	●	●	●
CB25	SLA L	●	●	●	●
CB2E	SRA (HL)	●	●	●	●
DDCb2E	SRA (IX+d)	●	●	●	●
FDCBd2E	SRA (IY+d)	●	●	●	●
CB2F	SRA A	●	●	●	●
CB28	SRA B	●	●	●	●
CB29	SRA C	●	●	●	●
CB2A	SRA D	●	●	●	●
CB2B	SRA E	●	●	●	●
CB2C	SRA H	●	●	●	●
CB2D	SRA L	●	●	●	●
CB3E	SRL (HL)	●	●	●	●
DDCb3E	SRL (IX+d)	●	●	●	●
FDCBd3E	SRL (IY+d)	●	●	●	●
CB3F	SRL A	●	●	●	●
CB38	SRL B	●	●	●	●
CB39	SRL C	●	●	●	●
CB3A	SRL D	●	●	●	●
CB3B	SRL E	●	●	●	●
CB3C	SRL H	●	●	●	●
CB3D	SRL L	●	●	●	●
96	SUB (HL)	●	●	●	●
DD96d	SUB (IX+d)	●	●	●	●
FD96d	SUB (IY+d)	●	●	●	●
97	SUB A	●	●	●	●
90	SUB B	●	●	●	●
91	SUB C	●	●	●	●
92	SUB D	●	●	●	●
93	SUB E	●	●	●	●
94	SUB H	●	●	●	●
95	SUB L	●	●	●	●
D6d	SUB d	●	●	●	●
AE	XOR (HL)	●	●	●	0
DDAEd	XOR (IX+d)	●	●	●	0
FDAEd	XOR (IY+d)	●	●	●	0
AF	XOR A	●	●	●	0
A8	XOR B	●	●	●	0
A9	XOR C	●	●	●	0
AA	XOR D	●	●	●	0
AB	XOR E	●	●	●	0
AC	XOR H	●	●	●	0
AD	XOR L	●	●	●	0
EEd	XOR d	●	●	●	0

Instructions sans préfixes

- n : octet (8 bits, de 0 à 255) ;
- nn : double octet (16 bits, de 0 à 65535) ;
- d : déplacement pour l'adressage relatif (8 bits).

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	LD BC,nn	LD (BC),A	INC BC	INC B	DEC B	LD B,n	RLCA	EX AF,AF'	ADD HL,BC	LD A,(BC)	DEC BC	INC C	DEC C	LD C,n	RRCA
1	DJNZ d	LD DE,nn	LD (DE),A	INC DE	INC D	DEC D	LD D,n	RLA	JR d	ADD HL,DE	LD A,(DE)	DEC DE	INC E	DEC E	LD E,n	RRR
2	JR NZ,d	LD HL,nn	LD (nn),HL	INC HL	INC H	DEC H	LD H,n	DAA	JR Z,d	ADD HL,HL	LD HL,(nn)	DEC HL	INC L	DEC L	LD L,n	CPL
3	JR NC,d	LD SP,nn	LD (nn),A	INC SP	INC (HL)	DEC (HL)	LD (HL),n	SCF	JR C,d	ADD HL,SP	LD A,(nn)	DEC SP	INC A	DEC A	LD A,n	CCF
4	LD B,B	LD B,C	LD B,D	LD B,E	LD B,H	LD B,L	LD B,(HL)	LD B,A	LD C,B	LD C,C	LD C,D	LD C,E	LD C,H	LD C,L	LD C,(HL)	LD C,A
5	LD D,B	LD D,C	LD D,D	LD D,E	LD D,H	LD D,L	LD D,(HL)	LD D,A	LD E,B	LD E,C	LD E,D	LD E,E	LD E,H	LD E,L	LD E,(HL)	LD E,A
6	LD H,B	LD H,C	LD H,D	LD H,E	LD H,H	LD H,L	LD H,(HL)	LD H,A	LD L,B	LD L,C	LD L,D	LD L,E	LD L,H	LD L,L	LD L,(HL)	LD L,A
7	LD (HL),B	LD (HL),C	LD (HL),D	LD (HL),E	LD (HL),H	LD (HL),L	HALT	LD (HL),A	LD A,B	LD A,C	LD A,D	LD A,E	LD A,H	LD A,L	LD A,(HL)	LD A,A
8	ADD A,B	ADD A,C	ADD A,D	ADD A,E	ADD A,H	ADD A,L	ADD A,(HL)	ADD A,A	ADC A,B	ADC A,C	ADC A,D	ADC A,E	ADC A,H	ADC A,L	ADC A,(HL)	ADC A,A
9	SUB B	SUB C	SUB D	SUB E	SUB H	SUB L	SUB (HL)	SUB A	SBC A,B	SBC A,C	SBC A,D	SBC A,E	SBC A,H	SBC A,L	SBC A,(HL)	SBC A,A
A	AND B	AND C	AND D	AND E	AND H	AND L	AND (HL)	AND A	XOR B	XOR C	XOR D	XOR E	XOR H	XOR L	XOR (HL)	XOR A
B	OR B	OR C	OR D	OR E	OR H	OR L	OR (HL)	OR A	CP B	CP C	CP D	CP E	CP H	CP L	CP (HL)	CP A
C	RET NZ	POP BC	JP NZ,nn	JP nn	CALL NZ,nn	PUSH BC	ADD A,n	RST 0	RET Z	RET	JP Z,nn	CALL Z,nn	CALL nn	ADC A,n	RST 8	
D	RET NC	POP DE	JP NC,nn	OUT (n),A	CALL NC,nn	PUSH DE	SUB n	RST 16	RET C	EXX	JP C,nn	IN A,(n)	CALL C,nn	SBC A,n	RST 24	
E	RET PO	POP HL	JP PO,nn	EX (SP),HL	CALL PO,nn	PUSH HL	AND n	RST 32	RET DE	RET (HL)	JP PE,nn	EX DE,HL	CALL PE,nn	XOR n	RST 40	
F	RET P	POP AF	JP P,nn	DI	CALL P,nn	PUSH AF	OR n	RST 48	RET M	LD SP,HL	JP M,nn	EI	CALL M,nn	CP n	RST 56	

LANGAGE MACHINE

Instructions avec le préfixe CB

Toutes les instructions de ce tableau doivent être précédées du préfixe CB.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	RLC B	RLC C	RLC D	RLC E	RLC H	RLC L	RLC (HL)	RLC A	RRC B	RRC C	RRC D	RRC E	RRC H	RRC L	RRC (HL)	RRC A
1	RL B	RL C	RL D	RL E	RL H	RL L	RL (HL)	RL A	RR B	RR C	RR D	RR E	RR H	RR L	RR (HL)	RR A
2	SLA B	SLA C	SLA D	SLA E	SLA H	SLA L	SLA (HL)	SLA A	SRA B	SRA C	SRA D	SRA E	SRA H	SRA L	SRA (HL)	SRA A
3									SRL B	SRL C	SRL D	SRL E	SRL H	SRL L	SRL (HL)	SRL A
4	BIT 0,B	BIT 0,C	BIT 0,D	BIT 0,E	BIT 0,H	BIT 0,L	BIT 0,(HL)	BIT 0,A	BIT 1,B	BIT 1,C	BIT 1,D	BIT 1,E	BIT 1,H	BIT 1,L	BIT 1,(HL)	BIT 1,A
5	BIT 2,B	BIT 2,C	BIT 2,D	BIT 2,E	BIT 2,H	BIT 2,L	BIT 2,(HL)	BIT 2,A	BIT 3,B	BIT 3,C	BIT 3,D	BIT 3,E	BIT 3,H	BIT 3,L	BIT 3,(HL)	BIT 3,A
6	BIT 4,B	BIT 4,C	BIT 4,D	BIT 4,E	BIT 4,H	BIT 4,L	BIT 4,(HL)	BIT 4,A	BIT 5,B	BIT 5,C	BIT 5,D	BIT 5,E	BIT 5,H	BIT 5,L	BIT 5,(HL)	BIT 5,A
7	BIT 6,B	BIT 6,C	BIT 6,D	BIT 6,E	BIT 6,H	BIT 6,L	BIT 6,(HL)	BIT 6,A	BIT 7,B	BIT 7,C	BIT 7,D	BIT 7,E	BIT 7,H	BIT 7,L	BIT 7,(HL)	BIT 7,A
8	RES 0,B	RES 0,C	RES 0,D	RES 0,E	RES 0,H	RES 0,L	RES 0,(HL)	RES 0,A	RES 1,B	RES 1,C	RES 1,D	RES 1,E	RES 1,H	RES 1,L	RES 1,(HL)	RES 1,A
9	RES 2,B	RES 2,C	RES 2,D	RES 2,E	RES 2,H	RES 2,L	RES 2,(HL)	RES 2,A	RES 3,B	RES 3,C	RES 3,D	RES 3,E	RES 3,H	RES 3,L	RES 3,(HL)	RES 3,A
A	RES 4,B	RES 4,C	RES 4,D	RES 4,E	RES 4,H	RES 4,L	RES 4,(HL)	RES 4,A	RES 5,B	RES 5,C	RES 5,D	RES 5,E	RES 5,H	RES 5,L	RES 5,(HL)	RES 5,A
B	RES 6,B	RES 6,C	RES 6,D	RES 6,E	RES 6,H	RES 6,L	RES 6,(HL)	RES 6,A	RES 7,B	RES 7,C	RES 7,D	RES 7,E	RES 7,H	RES 7,L	RES 7,(HL)	RES 7,A
C	SET 0,B	SET 0,C	SET 0,D	SET 0,E	SET 0,H	SET 0,L	SET 0,(HL)	SET 0,A	SET 1,B	SET 1,C	SET 1,D	SET 1,E	SET 1,H	SET 1,L	SET 1,(HL)	SET 1,A
D	SET 2,B	SET 2,C	SET 2,D	SET 2,E	SET 2,H	SET 2,L	SET 2,(HL)	SET 2,A	SET 3,B	SET 3,C	SET 3,D	SET 3,E	SET 3,H	SET 3,L	SET 3,(HL)	SET 3,A
E	SET 4,B	SET 4,C	SET 4,D	SET 4,E	SET 4,H	SET 4,L	SET 4,(HL)	SET 4,A	SET 5,B	SET 5,C	SET 5,D	SET 5,E	SET 5,H	SET 5,L	SET 5,(HL)	SET 5,A
F	SET 6,B	SET 6,C	SET 6,D	SET 6,E	SET 6,H	SET 6,L	SET 6,(HL)	SET 6,A	SET 7,B	SET 7,C	SET 7,D	SET 7,E	SET 7,H	SET 7,L	SET 7,(HL)	SET 7,A

Instructions avec préfixe ED

Toutes les instructions de ce tableau doivent être précédées du préfixe ED.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1																
2																
3																
4	IN 0,(C)	OUT (C),B	SBC HL,BC	ID (nn)BC	NEG	RETN	IM 0	LD I,A	IN C,(C)	OUT (C),C	ADC HL,BC	LD BC(nn)		RETI		LD R,A
5	IN 0,(C)	OUT (C),D	SBC HL,DE	LD (nn)DE			IM 1	LD A,I	IN E,(C)	OUT (C),E	ADC HL,DE	LD DE(nn)			IM 2	LD A,R
6	IN H,(C)	OUT (C),H	SBC HL,HL	LD (nn)HL				RRD	IN L,(C)	OUT (C),L	ADC HL,HL	LD HL(nn)				RLD
7	IN F,(C)		SBC HL,SP	LD (nn)SP					IN A,(C)	OUT (C),A	ADC HL,SP	LD SP(nn)				
8																
9																
A	LDI	CPI	INI	OUTI					LDD	CPD	IND	OUTD				
B	LDIR	CPIR	INIR	OTIR					LDDR	CPDR	INDR	OTDR				
C																
D																
E																
F																

Instructions indexées

Toutes les instructions de ce tableau doivent être précédées par le préfixe DD, dans le cas du registre d'index IX, et de FD, pour le registre IY.

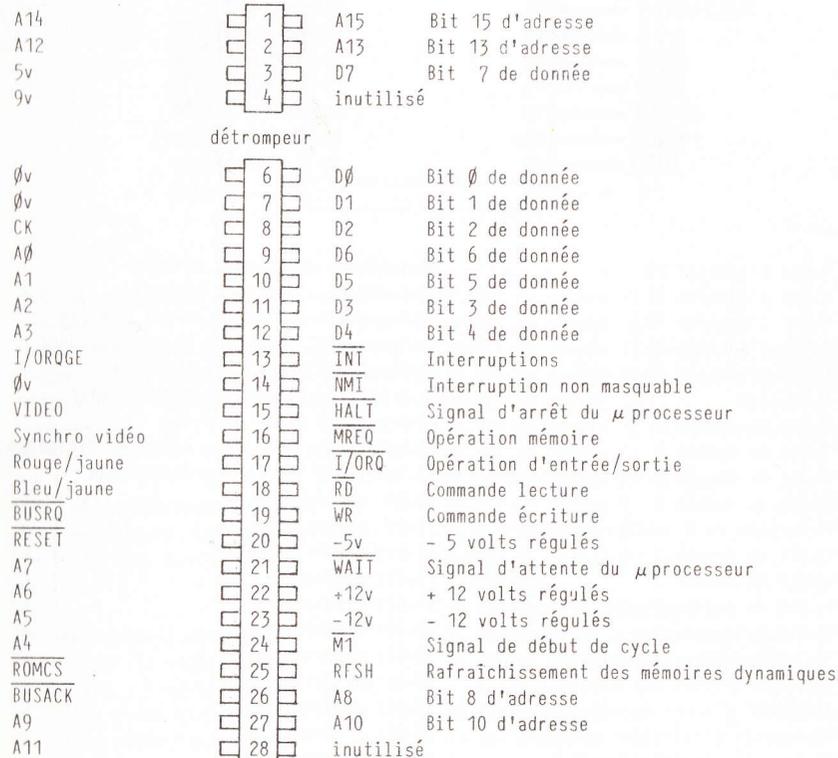
Code	Mnémonique	Code	Mnémonique
09	ADD IX,BC	CB d 0E	RRC (IX+d)
19	ADD IX,DE	CB d 16	RL (IX+d)
21	LD IX,nn	CB d 1E	RR (IX+d)
22	LD (nn),IX	CB d 26	SLA (IX+d)
23	INC IX	CB d 2E	SRA (IX+d)
29	ADD IX,IX	CB d 3E	SRL (IX+d)
2A	LD IX,(nn)	CB d 46	BIT 0,(IX+d)
2B	DEC IX	CB d 4E	BIT 1,(IX+d)
34	INC (IX+d)	CB d 56	BIT 2,(IX+d)
35	DEC (IX+d)	CB d 5E	BIT 3,(IX+d)
36	LD (IX+d),nn	CB d 66	BIT 4,(IX+d)
39	ADD IX,SP	CB d 6E	BIT 5,(IX+d)
46	LD B,(IX+d)	CB d 76	BIT 6,(IX+d)
4E	LD C,(IX+d)	CB d 7E	BIT 7,(IX+d)
56	LD D,(IX+d)	CB d 86	RES 0,(IX+d)
5E	LD E,(IX+d)	CB d 8E	RES 1,(IX+d)
66	LD H,(IX+d)	CB d 96	RES 2,(IX+d)
6E	LD L,(IX+d)	CB d 9E	RES 3,(IX+d)
70	LD (IX+d),B	CB d A6	RES 4,(IX+d)
71	LD (IX+d),C	CB d AE	RES 5,(IX+d)
72	LD (IX+d),D	CB d B6	RES 6,(IX+d)
73	LD (IX+d),E	CB d BE	RES 7,(IX+d)
74	LD (IX+d),H	CB d C6	SET 0,(IX+d)
75	LD (IX+d),L	CB d CE	SET 1,(IX+d)
77	LD (IX+d),A	CB d D6	SET 2,(IX+d)
7E	LD 1,(IX+d)	CB d DE	SET 3,(IX+d)
86	ADD A,(IX+d)	CB d E6	SET 4,(IX+d)
8E	ADC A,(IX+d)	CB d EE	SET 5,(IX+d)
96	SUB (IX+d)	CB d F6	SET 6,(IX+d)
9E	SBC A,(IX+d)	CB d FE	SET 7,(IX+d)
A6	AND (IX+d)	E1	POP IX
AE	XOR (IX+d)	E3	EX (SP),IX
B6	OR (IX+d)	E5	PUSH IX
BE	CP (IX+d)	E9	JP (IX)
CB d 06	RLC (IX+d)	F9	LD SP,IX

# CONNECTEURS

## CONNECTEUR ARRIERE DU SPECTRUM

Côté soudure

Côté composants

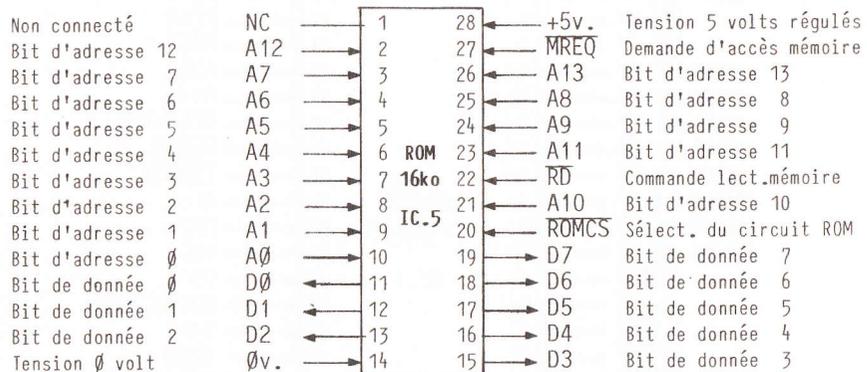


CONNECTEURS

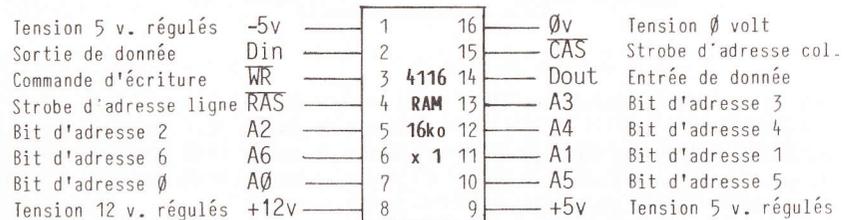


BROCHAGE DES CIRCUITS INTEGRES

Mémoire morte



Mémoire vive 16 ko



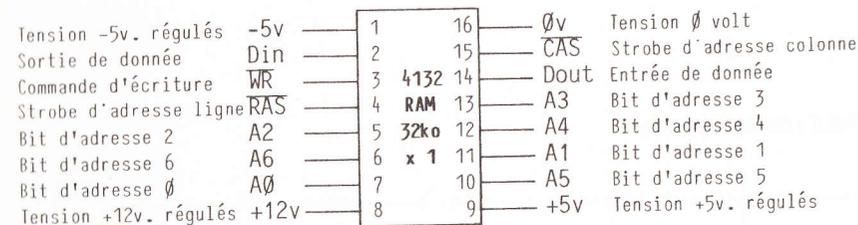
IC7 à IC14

IC7	IC8	IC9	IC10	IC11	IC12	IC13	IC14
D1	D2	D0	D3	D4	D5	D6	D7

Répartition des bits de données

BROCHAGE DES CIRCUITS INTEGRES

Mémoire vive 32ko



IC3 à IC10

IC3	IC4	IC5	IC6	IC7	IC8	IC9	IC10
D2	D6	D2	D0	D5	D2	D4	D3

Répartition des bits de données

CONNEXIONS

## CARTE MEMOIRES

P.RAMT		
RAM-TOP	Graphiques utilisateur	<p>RAM 1600 48 Ko</p> <p>ROM 16 ko</p>
STKEND	Pile de GO SUB	
STKBOT	Pile du calculateur	
WORKSP	Espace de travail temporaire	
E-LINE	Ligne en cours d'édition	
VAR5	Variables Basic	
PROG	Programme Basic	
CHANS	Informations sur les différents canaux	
23734	Réservé aux microdrives	
23552	Variables système	
23296	Tampon imprimante	
22528	Attributs vidéo	
16384	Fichier écran	
Ø	BASIC	

ADRESSES

Ces emplacements mémoires sont des points d'entrées utilisables par vos programmes en langage machine.

### Routine RST

- 00 Initialisation. C'est la première instruction exécutée par le microprocesseur après un Reset. La routine appelle ensuite le programme en 11CB pour initialiser la mémoire, l'écran et les différentes variables.
- 08 Routine d'erreur. L'octet qui suit RST08 donne le type d'erreur. Le programme retourne automatiquement au Basic.
- 10 Routine d'affichage d'un caractère. Ce dernier doit être placé dans l'accumulateur avant RST10.
- 18 Place dans l'accumulateur le prochain caractère à interpréter par le Basic (pointé par la variable système CH-ADD) et vérifie s'il est affichable à l'écran.
- 20 Identique à RST18, mais pour le caractère suivant (variable système CH-ADD +1).
- 28 Routine de calculateur en virgule flottante. Cette routine appelle celle placée en 335B.
- 30 Réserve de la place pour les variables. Le registre BC doit contenir le nombre d'octets que l'on veut ainsi récupérer.
- 38 Routine d'interruption masquable. Cette routine génère une ligne sur l'écran à chaque interruption.
- 0053 Routine d'interprétation des erreurs.
- 0066 Routine des interruptions non-masquables.
- 0074 Incrémente le pointeur CH-ADD.
- 007D Teste si la valeur contenue dans le registre A est affichable (routine appelée par RST0018).
- 0095 Table des instructions. Le septième bit du dernier caractère de chaque instruction est mis à un.

### Table des touches du clavier

- 0205 Table des touches en mode L.
- 022C Table des fonctions avec "CAPS SHIFT" (READ, BIN, etc...).
- 0246 Table des fonctions avec "CAPS SHIFT" + "SYMBOL SHIFT" (trois rangées du bas).
- 0260 Table des codes de contrôle (DELETE, EDIT, etc...).

- 026A Table des fonctions avec "SYMBOL SHIFT" (STOP, \*, etc...).
- 0284 Table des fonctions avec "CAPS SHIFT" + "SYMBOL SHIFT" (rangée du haut).

### Routines clavier

- 028E Routine de scrutation du clavier.
- 02BF Routine clavier appelée toutes les 20 ms par l'interruption masquable.
- 0310 Routine de répétition d'une touche quand le délai donné par les variables REPDEL et REPPER est passé.
- 031E Routine de test de la touche. Revient normalement si aucune touche n'est appuyée ou seulement SHIFT. Sinon le registre HL pointe la table des touches du clavier.
- 0333 Routine de décodage. Le registre E doit contenir le code de la table de base, le registre D la valeur de FLAGS, le registre C le mode et le registre B le mode shift. La routine place alors dans le registre A le code final.

### Routines BUZZER

- 03B5 Routine d'exécution d'un beep sonore. La paire de registres DE doit contenir les valeurs de F et T (fréquence et temps) et HL celle du temps de base de la boucle.
- 03F8 Routine de la commande BEEP.
- 046E Table des fréquences des douze demi-tons d'une octave.

Adresse	Fréquence Hz	Note
046E	261.63	D0
0473	277.18	D0 $\sharp$
0478	293.66	RE
047D	311.13	RE $\sharp$
0482	329.63	MI
0487	349.23	FA
048C	369.99	FA $\sharp$
0491	392.00	SOL
0496	415.30	SOL $\sharp$
049B	440.00	LA
04A0	466.16	LA $\sharp$
04A5	493.88	SI

Routines cassette

- 04C2** Routine de sauvegarde des informations d'identification du fichier (nom et nombre de blocs).
- 053F** Routine de modification de la couleur du fond de l'écran dans les instructions SAVE et LOAD.
- 0556** Routine de chargement par l'instruction LOAD ou VERIFY des informations d'identification de fichier (nom et nombre de blocs).
- 05E3** Routine de test de pré-chargement. Le registre B doit contenir la constante de temps et le registre C les couleurs du fond de l'écran. La routine met le drapeau de retenue à 1 si la synchronisation est correcte.
- 0605** Routine des commandes SAVE, LOAD, VERIFY et MERGE. Le contenu de la variable T-ADDR permet de distinguer les quatre commandes.
- 07CB** Routine de contrôle de la commande VERIFY.
- 0802** Routine de chargement d'un bloc des commandes LOAD, VERIFY et MERGE.
- 0808** Routine de contrôle de la commande LOAD.
- 08B6** Routine de contrôle de la commande MERGE.
- 092C** Routine de chargement d'une ligne Basic ou d'une variable par la commande MERGE. Le drapeau de retenue est à 0 pour une variable. Le drapeau de zéro est à 0 pour ajouter une ligne ou une variable et à 1 pour un remplacement. La paire de registres HL pointe l'emplacement de la nouvelle entrée ou DE celui du merge.
- 0970** Routine de contrôle de la commande SAVE.
- 09A1** Liste des messages cassettes. Le dernier caractère du message est inversé (on ajoute 80 hex. à sa valeur ASCII).

Routines écran et imprimante

- 09F4** Routine d'initialisation. Le registre A doit contenir le caractère à afficher.
- 0A11** Table des caractères de contrôle.

Adresse	Offset	Caractère	Adresse	Offset	Caractère
0A11	4E	PRINT	0A1A	4F	pas utilisé
0A12	57	EDIT	0A1B	5F	INK ctrl
0A13	10	curseur gauche	0A1C	5E	PAPER ctrl
0A14	29	curseur droite	0A1D	5D	FLASH ctrl
0A15	54	curseur bas	0A1E	5C	BRIGHT ctrl
0A16	53	curseur haut	0A1F	5B	INVERSE ctrl
0A17	52	DELETE	0A20	5A	OVER ctrl
0A18	37	ENTER	0A21	*54	AT ctrl
0A19	50	pas utilisé	0A22	53	TAB ctrl

- 0A23** Routine de déplacement du curseur vers la gauche.
- 0A3D** Routine de déplacement du curseur vers la droite.
- 0A4F** Routine du retour chariot.
- 0A5F** Routine d'affichage de la virgule.
- 0A69** Routine d'affichage du point d'interrogation.
- 0A6D** Routine des caractères de contrôle avec opérande (INK, PAPER, etc...).
- 0AD9** Routine des caractères "affichables" (appel de la routine en 0B24).
- 0ADC** Routine de modification des variables système avec la nouvelle position curseur et l'adresse du pixel.
- 0B03** Routine de recherche de la position actuelle du curseur dans les variables système.
- 0B24** Routine d'affichage de n'importe quel caractère.
- 0B7F** Affiche tous les caractères à partir de leur matrice 8x8.
- 0BDB** Routine de recherche de l'attribut. Ce dernier est mélangé avec l'ancienne valeur en fonction des variables système ATTR-T, MASK-T et P-FLAG.
- 0COA** Routine d'affichage d'un message. Le registre A doit contenir la longueur du message et la paire de registres DE l'adresse de la table.
- 0C3B** Routine d'affichage d'un caractère avec sauvegarde des registres DE, HL et BC.
- 0C41** Place dans le registre DE l'adresse de début de la table à afficher.
- 0C55** Routine de test du scroll. Cette routine est appelée lors de l'utilisation d'un retour chariot, d'une fonction PRINT AT ou de l'affichage d'un caractère sur une ligne complète.

- 0CF8** Liste des caractères du message "scroll?". Le point d'interrogation est inversé (on ajoute 80 hex. à sa valeur ASCII soit BF hex.).
- 0D4D** Routine de sauvegarde de la couleur dans la variable système ATTR-T.
- 0D6B** Routine de la commande CLS. Tous les pixels sont mis à zéro et les attributs prennent la valeur de la variable système ATTR-P (attribut permanent).
- 0DAF** Routine d'effacement de la zone d'affichage. Cette routine est appelée par la commande CLS, la commande RUN et la commande LIST.
- 0DD9** Donne l'adresse du caractère dont le numéro de ligne et de colonne se trouve dans la paire de registres BC ou le numéro de colonne dans le registre C s'il s'agit de l'imprimante.
- 0DFE** Routine du scroll.
- 0E44** Routine d'effacement de lignes. La routine efface B (registre) lignes à partir du bas de l'écran.
- 0E88** Place dans la paire de registres DE l'adresse de l'attribut correspondant à l'adresse pointée par HL dans le fichier d'affichage.
- 0E9B** Place dans la paire de registres HL l'adresse du fichier d'affichage à partir du nombre de lignes contenues dans le registre B.
- 0EAC** Routine de la commande COPY.
- 0ECD** Copie la ligne qui doit être imprimée dans le buffer.
- 0EDF** Efface le contenu du buffer d'imprimante.
- 0EF4** Routine de copie d'une ligne à imprimer. La paire de registres HL doit contenir l'adresse des 33 caractères et le registre B le numéro du pixel dans cette ligne.
- 0F2C** Routine de l'éditeur. Cette routine est utilisée lors de l'entrée d'une ligne Basic ou par la commande INPUT.
- 0F81** Routine d'ajout d'un caractère lors de l'entrée d'une ligne Basic ou d'un INPUT.
- OFA0** Table des touches utilisées lors d'une édition.

Adresse	Offset	Caractère	Adresse	Offset	Caractère
OFA0	09	EDIT	OFA5	70	DELETE
OFA1	66	curseur gauche	OFA6	7E	ENTER
OFA2	6A	curseur droite	OFA7	CF	SYMBOL SHIFT
OFA3	50	curseur bas	OFA8	D4	GRAPHICS
OFA4	B5	curseur haut			

- OFA9** Routine de la touche EDIT.
- OFF3** Routine de la touche curseur bas lors d'une édition.
- 1007** Routine de la touche curseur gauche lors d'une édition.
- 100C** Routine de la touche curseur droit lors d'une édition.
- 1015** Routine de la touche DELETE lors d'une édition.
- 101E** Teste deux fois le clavier avant une édition et ignore les caractères qui auraient pu être tapés.
- 1024** Routine de la touche ENTER lors d'une édition.
- 1031** La paire de registre HL contenant l'adresse du curseur, on la décréméte pour que le curseur soit placé en début en ligne.
- 1059** Routine de la touche curseur haut lors de l'édition.
- 1076** Routine des touches "SYMBOL" ou "GRAPHICS" lors de l'édition.
- 107F** Routine de gestion d'erreur lors d'une édition.
- 1097** Routine d'effacement de la zone écran d'édition.
- 10A8** Donne la valeur de la dernière touche appuyée.
- 111D** Réécrit la ligne en cours d'édition en bas de l'écran.
- 1190** Place dans la paire de registres HL l'adresse de début de la zone mémoire utilisée par l'éditeur et dans DE l'adresse de fin de cette même zone.
- 11A7** Place dans la pile tous les nombres en virgule flottante d'une ligne Basic en cours d'interprétation.

#### Routines générales

- 11B7** Routine de la commande NEW.
- 11CB** Initialise le registre I et met le bord de l'écran en blanc.
- 11DA** Recherche la taille de la mémoire vive et place le résultat dans RAMTOP.
- 12A2** Cette routine contrôle les éditions, les exécutions en mode direct et donne les comptes-rendus (de 12A2 à 15AE).
- 1391** Liste des comptes-rendus (ou messages d'erreurs).
- 1539** Copyright Sinclair research Ltd.
- 155D** Ajoute une nouvelle ligne de Basic au programme déjà existant en mémoire.

- 15AF Initialisation du clavier, de l'écran, de la place mémoire et de l'imprimante.
- 15C6 Tableau des données pour l'initialisation.
- 15D4 Routine de contrôle de la ligne éditée par INPUT.
- 15E6 Sauve les registres et place dans HL l'adresse de début de la zone d'édition pointée par CURCHL.
- 15EF Routine d'entrée/sortie du code contenu dans le registre A.
- 1601 Routine d'ouverture d'un canal.
- 1615 Initialisation des drapeaux des différents canaux (clavier, écran, etc...) dans la variable système FLAGS2.
- 162D Liste des routines d'initialisation des différents canaux.
- 1634 Routine d'initialisation du clavier.
- 1642 Routine d'initialisation de l'écran.
- 164D Routine d'initialisation de l'imprimante.
- 1652 Routine d'initialisation de l'espace mémoire.
- 1664 Modification de l'emplacement des pointeurs. La paire de registres BC doit contenir le nombre d'octets demandé et HL l'adresse juste avant la position.
- 168F Place dans la paire de registres DE le numéro de ligne pointé par HL.
- 169E Demande de place mémoire (routine appelée par RST 0030).
- 16B0 Initialisation des variables pointant la zone d'édition.
- 16DB Explore la table pointée par la paire de registres HL.
- 16E5 Routine de la commande CLOSE £.
- 1701 Fermeture des canaux spécifiés.
- 1716 Liste des routines de fermeture des différents canaux.
- 171E Place dans la paire de registres BC la donnée du canal spécifié par le registre A.
- 1736 Routine de la commande OPEN £.
- 177A Table des adresses des routines d'ouverture des canaux.
- 1793 Routine des commandes CAT, ERASE, FORMAT et MOVE.
- 1795 Routine des commandes LIST et LLIST.
- 17F5 Point d'entrée de la commande LLIST.
- 17F9 Point d'entrée de la commande LIST.
- 1855 Routine d'affichage d'une ligne Basic.

- 18B6 Routine de gestion des nombres en virgule flottante à l'intérieur d'une ligne Basic.
- 18C1 Routine d'affichage d'un caractère clignotant.
- 18E1 Routine d'affichage du curseur.
- 190F Modifie le contenu des variables système S-TOP et E-PPC avec le numéro de la ligne suivante. La paire de registres HL doit pointer une de ces deux variables.
- 1925 Routine d'affichage des caractères contenus dans une ligne Basic.
- 196E Recherche d'adresse d'une ligne Basic. Le numéro de cette ligne doit être placé dans la paire de registres HL. La routine place dans HL cette adresse et dans DE celle du premier caractère de la ligne précédente.
- 1980 Compare un numéro de ligne Basic avec une adresse mémoire.
- 1988 Recherche d'instruction dans une ligne Basic.
- 1988 Recherche l'adresse de la prochaine ligne ou de la prochaine variable.
- 19DD Place dans HL le nombre d'octets compris entre le premier caractère de la ligne et celui de la ligne suivante.
- 19E5 Déplace la zone mémoire pointée par la paire de registres DE dans celle pointée par HL et dont la longueur est spécifiée par BC.
- 19FB Routine de lecture du numéro de ligne Basic. Celui-ci est placé dans la paire de registres BC. S'il s'agit d'une commande en mode direct, BC est alors égale à zéro.
- 1A1B Affiche le compte-rendu et le numéro de ligne.

Interprétation des lignes Basic

Adresse	Offset	Commande	Adresse routine
1A48	B1	DEF FN	1AF9
1A49	CB	CAT	1B14
1A4A	BC	FORMAT	1B06
1A4B	BF	MOVE	1B0A
1A4C	C4	ERASE	1B10
1A4D	AF	OPEN £	1AFC
1A4E	B4	CLOSE £	1B02
1A4F	93	MERGE	1AE2
1A50	91	VERIFY	1AE1
1A51	92	BEEP	1AE3

Adresse	Offset	Commande	Adresse routine
1A52	95	CIRCLE	1AE7
1A53	98	INK	1AEB
1A54	98	PAPER	1AEC
1A55	98	FLASH	1AED
1A56	98	BRIGHT	1AEE
1A57	98	INVERSE	1AEF
1A58	98	OVER	1AF0
1A59	98	OUT	1AF1
1A5A	7F	LPRINT	1AD9
1A5B	81	LLIST	1ADC
1A5C	2E	STOP	1A8A
1A5D	6C	READ	1AC9
1A5E	6E	DATA	1ACC
1A5F	70	RESTORE	1ACF
1A60	48	NEW	1AA8
1A61	94	BORDER	1AF5
1A62	56	CONTINUE	1AB8
1A63	3F	DIM	1AA2
1A64	41	REM	1AA5
1A65	2B	FOR	1A90
1A66	17	GO TO	1A7D
1A67	1F	GO SUB	1A86
1A68	37	INPUT	1A9F
1A69	77	LOAD	1AE0
1A6A	44	LIST	1AAE
1A6B	0F	LET	1A7A
1A6C	59	PAUSE	1AC5
1A6D	2B	NEXT	1A98
1A6E	43	POKE	1AB1
1A6F	2D	PRINT	1A9C
1A70	51	PLOT	1AC1
1A71	3A	RUN	1AAB
1A72	6D	SAVE	1ADF
1A73	42	RANDOMIZE	1AB5
1A74	0D	IF	1A81
1A75	49	CLS	1ABE
1A76	5C	DRAW	1AD2
1A77	44	CLEAR	1ABB
1A78	15	RETURN	1A8D
1A79	5D	COPY	1AD6

1B17 Table des paramètres des commandes.

1B28 Boucle d'interprétation des commandes d'une ligne Basic jusqu'à la fin de cette dernière.

1B6F Affiche le message "Nonsense in BASIC" si le séparateur (deux points) n'est pas présent. Affiche seulement le point d'interrogation pour l'édition.

1B76 Retour après une interprétation correcte.

1B8A Met à 1 le bit 7 de la variable système FLAGS quand une ligne Basic à interpréter est dans la zone d'édition.

1B9E Recherche l'adresse de la prochaine ligne à interpréter.

1BB2 Routine de la commande REM.

1BB3 Place dans la paire de registres HL le numéro de la prochaine ligne Basic à interpréter (variable système NXTLIN).

1BBF Place dans la variable système PPC le numéro de la nouvelle ligne Basic et dans la paire de registres HL son adresse.

1BD1 Place dans la variable système NXTLIN le numéro de la prochaine "nouvelle" ligne Basic.

1BEE Recherche l'erreur de syntaxe d'une instruction et se place devant le caractère de l'instruction suivante.

1BF4 Donne une erreur de syntaxe si le caractère interprété n'est pas un retour chariot, l'instruction de la ligne suivante ou l'instruction qui suit les deux points.

1C01 Tableau de regroupement des types de commande.

1C22 Mis à jour des variables système DEST et STRLEN (adresse de la variable en cours et longueur s'il s'agit d'une chaîne de caractères).

1C96 Place la couleur temporaire (variable système ATTR-T) dans la couleur permanente (ATTR-P).

#### Routines des commandes

1CEE Routine de la commande STOP.

1CF0 Routine de la commande IF.

1D03 Routine de la commande FOR.

1D86 Recherche le dernier DATA, DEF FN ou NEXT. Le registre E doit contenir le code de l'instruction et la paire de registres HL doit pointer le début de la zone de recherche.

1DAB Routine de la commande NEXT.

1DDA Cette routine détermine la valeur limite de la boucle.

1DEC Routine de la commande READ.

1E27 Routine de la commande DATA.

- 1E42 Routine de la commande RESTORE.
- 1E4F Routine de la commande RANDOMIZE.
- 1E5F Routine de la commande CONTINUE.
- 1E67 Routine de la commande GO TO.
- 1E7A Routine de la commande OUT.
- 1E80 Routine de la commande POKE.
- 1E85 Vérifie les deux paramètres de POKE.
- 1E94 Comprime la dernière valeur de la pile du calculateur dans le registre A (routine 1E94) ou dans la paire de registres BC (routine 1E99).
- 1EA1 Routine de la commande RUN.
- 1EAC Routine de la commande CLEAR.
- 1EED Routine de la commande CLEAR.
- 1F05 Teste si la place est suffisante en mémoire pour les prochaines opérations.
- 1F1A Stocke dans la paire de registres BC le nombre d'octets libres en mémoire.
- 1F23 Routine de la commande RETURN.
- 1F3A Routine de la commande PAUSE.
- 1F54 Routine de test de la touche BREAK. Le programme n'effectue un arrêt que si les touches SHIFT et BREAK sont appuyées simultanément.
- 1F60 Routine de la commande DEF FN.
- 1FC9 Routine des commandes LPRINT.
- 1FCF Routine de la commande PRINT.
- 1FF5 Effectue un retour chariot
- 1FFC Identification des divers affichages (PRINT, LPRINT et INPUT).
- 2045 Met à 1 le drapeau de zéro pour indiquer la fin d'une impression.
- 204E Gestion des caractères de contrôle.
- 2089 Routine de la commande INPUT.
- 21D6 Met le drapeau Z à 1 si c'est le clavier qui est utilisé.
- 21E1 Routine des commandes PAPER, INK et INVERSE.
- 226C Routine des commandes FLASH et BRIGHT.
- 2294 Routine de la commande BORDER.

- 22AA Routine d'adressage d'un pixel. La paire de registre BC doit contenir l'adresse du pixel. La routine place alors dans HL l'adresse du pixel sur l'écran et dans le registre A sa position dans l'octet (routine utilisée par POINT et PLOT).
- 22CB Routine de la commande POINT.
- 22CD Routine de la commande PLOT.
- 2307 Place dans la paire de registres BC deux nombres en virgule flottante (paramètres compris entre 0 et FF pour l'instruction DRAW).
- 2314 Place dans le registre A le nombre en virgule flottante du haut de la pile (compris entre 0 et FF).
- 2320 Routine de la commande CIRCLE.
- 2382 Routine de la commande DRAW.
- 247D Initialisation des paramètres des commandes CIRCLE et DRAW.
- 24B7 Trace sur l'écran une ligne en suivant les paramètres donnés par DRAW.

Evaluation d'une expression

- 24FB Routine d'évaluation d'une expression numérique ou d'une chaîne de caractères. Le bit 6 de la variable système FLAGS est mis à 1 si le résultat est numérique.
- 2535 Routine d'évaluation des paramètres X et Y de la fonction SCREEN\$. Donne le caractère correspondant à ces coordonnées en utilisant la table pointée par la variable système CHARS.
- 2580 Routine d'évaluation des paramètres X et Y de la fonction ATTR. Donne l'attribut correspondant à ces coordonnées.
- 2596 Table des adresses d'évaluation de fonction.

Adresse	Evaluation de	Adresse routine
2596	double quotes	25B3
2598	crochets	25E8
259A	décimal	268D
259C	numérique 1 caractère	25AF
259E	fonction FN	
25A0	fonction RND	25F8
25A2	PI	2627
25A4	fonction INKEY\$	2634
25A6	BIN	268D
25A8	SCREEN\$	2668
25AA	ATTR	2672

**ADRESSES DE LA ROM BASIC**  
(de 1000 à 3CFF)

- 25AF** Evaluation d'un numérique à un seul caractère. On utilise seulement RST 0020.
- 25B3** Recherche de la position des quotes d'une chaîne de caractères.
- 26C9** Recherche de la valeur d'une variable quand celle-ci a été identifiée.
- 2795** Table des opérations.

Adresse	Opération	Code	Code opération
2795	+	2B	CF
2797	-	2D	C3
2799	*	2A	C4
279B	/	2F	C5
279D	^	5E	C6
279F	=	3D	CE
27A1	>	3E	CC
27A3	<	3C	CD
27A5	<=	C7	C9
27A7	>=	C8	CA
27A9	<>	C9	CB
27AB	OR	C5	C7
27AD	AND	C6	C8
27AF	Code de fin	00	

- 27B0** Table des priorités.

Adresse	Opération	Code
27B0	-	06
27B1	*	08
27B2	/	08
27B3	^	0A
27B4	OR	02
27B5	AND	03
27B6	<=	05
27B7	>=	05
27B8	<>	05
27B9	>	05
27BA	<	05
27BB	=	05
27BC	+	06

- 27BD** Routine d'évaluation d'une fonction utilisateur FN.

**ADRESSES DE LA ROM BASIC**  
(de 1000 à 3CFF)

- 28B2** Recherche les variables ou les arguments d'une fonction utilisateur déclarée par DEF FN. La variable système DEFADD donne l'adresse du premier caractère de la variable recherchée.
- 2951** Utilisé par la routine précédente pour rechercher les arguments du DEF FN avant de le faire dans la zone mémoire des variables.
- 2996** Recherche les paramètres d'une chaîne de caractères ou place, dans la paire de registres HL, l'adresse de début des éléments d'un tableau (utilisée par la commande DIM).
- 2A52** Routine de découpage d'une chaîne de caractères. Les paramètres de cette dernière doivent être placés dans le haut de la pile du calculateur et dans les registres A, B, C, D et E.
- 2AB6** Place le contenu des registres A, B, C, D et E dans la pile du calculateur.
- 2ACC** Place dans la paire de registres BC le résultat de la dernière expression sous forme d'un entier. S'il y a débordement, le drapeau de retenue est mis à 1 et le registre A contient alors FF.
- 2AEE** Place dans la paire de registres DE le contenu de l'adresse de DE+2.
- 2AF4** Place dans la paire de registres HL le résultat de l'opération HL\*DE.
- 2AFF** Routine de la commande LET.
- 2BA6** Transfert une valeur numérique de la pile du calculateur ou une chaîne de la zone de travail dans la zone mémoire des variables Basic.
- 2BAF** Suite de la routine de la commande LET.
- 2BC6** Transfert les paramètres d'une nouvelle chaîne de caractères en ayant, au préalable, réservé de la place mémoire pour celle-ci.
- 2B1F** Place, dans les registres A, B, C, D et E, le dernier nombre en virgule flottante de la pile du calculateur.
- 2C02** Routine de la commande DIM.
- 2C88** Met à 1 le drapeau de retenue si le registre A contient une lettre ou un chiffre.
- 2C8D** Met à 1 le drapeau de retenue si le registre A contient une lettre de l'alphabet.
- 2C9B** Convertit un nombre décimal d'une ligne Basic en nombre en virgule flottante utilisable par le calculateur. Le résultat est placé en haut de la pile du calculateur.

- 2D1B Met à 1 le drapeau de retenue si le registre A contient un numérique.
- 2D22 Sauve le nombre en virgule flottante dans la dernière valeur de la pile du calculateur si le registre A contient un numérique.
- 2D28 Donne le format virgule flottante à la valeur absolue du contenu du registre A.
- 2D2B Donne le format virgule flottante à la valeur absolue du contenu de la paire de registres BC.
- 2D3B Convertit un nombre entier d'une ligne Basic (ou un numéro de ligne) au format virgule flottante. Le résultat est placé dans la dernière valeur de la pile du calculateur.

#### Routines arithmétiques

- 2D4F Convertit un nombre sous forme exponentielle au format virgule flottante. Le nombre doit être placé en haut de la pile du calculateur et l'exposant dans le registre A. Le résultat est stocké dans la dernière valeur de la pile du calculateur.
- 2D7F Place dans la paire de registres DE le nombre entier (compris entre -65535 et +65535) pointé par la paire de registres HL (généralement le haut de la pile du calculateur).
- 2D8E Stocke le nombre entier contenu dans la paire de registres DE à l'adresse pointée par la paire de registres HL.
- 2DA2 Comprime le nombre en virgule flottante de la dernière valeur de la pile du calculateur dans la paire de registres BC. Si le résultat est trop grand (supérieur à 65535), le drapeau de retenue est mis à 1. L'octet bas du résultat est copié dans le registre A.
- 2DC1 Calcule le nombre de digits pour l'impression d'un nombre en virgule flottante sur l'écran. Le registre A doit contenir l'exposant.
- 2DD5 Comprime le nombre en virgule flottante de la pile du calculateur dans le registre A si cela est possible. Le drapeau de retenue est mis à 1 si le résultat est trop grand.
- 2DE3 Affiche à l'écran le dernier nombre en virgule flottante de la pile du calculateur. Cette routine est utilisée par les fonctions PRINT et STR\$.
- 2F8B Effectue l'opération  $CA=(10*A)+C$ .

- 2F9B Prépare aux routines d'addition, soustraction, multiplication et division en effectuant un complément à 2 pour les nombres négatifs. L'exposant est placé dans le registre A et le premier octet est 0 pour les nombres positifs et FF pour les négatifs.
- 2FBA Place les deux derniers nombres en virgule flottante de la pile du calculateur dans les registres et les registres primes pour l'addition, la multiplication et la division. La paire de registres HL doit pointer le premier octet du premier nombre et DE celui du second.
- 300F Routine de soustraction. Cette routine modifie le signe du nombre à soustraire et appelle ensuite la routine d'addition.
- 3014 Routine d'addition.
- 30A9 Effectue l'opération  $HL*DE$  et place le résultat dans la paire de registres HL.
- 30C0 Prépare les nombres en virgule flottante pour la multiplication et la division en mettant le drapeau de retenue à 1 si le nombre est nul et le signe du résultat dans le registre A.
- 30CA Routine de multiplication.
- 31AF Routine de division.
- 3214 Tronque un nombre pour tendre vers 0 (5.2 donne 5 et -5.2 donne -5).
- 3293 Sauve dans la pile deux nombres en virgule flottante pour l'addition, la multiplication et la division.
- 3297 Sauve dans la pile un nombre en virgule flottante pour les fonctions ARCTAN, EXP et LN.

#### Routines du calculateur en virgule flottante

- 3C25 Table des constantes utilisées par le calculateur au format virgule flottante.

Adresse	Valeurs
32C5	0
32C8	1
32CC	1/2
32CE	1/2 de PI
32D3	10

**32D7** Table des adresses des fonctions utilisant les routines en virgule flottante. Ces routines sont appelées par la fonction RST0028 ou la routine en 2734. Le nombre qui suit RST0028, dans le programme, donne le type d'opération à effectuer. Une liste d'opérations doit toujours se terminer par le code 38 (fin des calculs).

	Opération	Code après RST28	Adresse routine
<b>32D7</b>	Saut si exact	00	368F
<b>32D9</b>	Echange les opérandes	01	343C
<b>32DB</b>	Suppression	02	33A1
<b>32DD</b>	Soustraction	03	300F
<b>32DF</b>	Multiplication	04	30CA
<b>32E1</b>	Division	05	31AF
<b>32E3</b>	Puissance	06	3851
<b>32E5</b>	Or	07	351B
<b>32E7</b>	AND	08	3524
<b>32E9</b>	Comparaisons	09	353B
<b>32EB</b>	Comparaisons	0A	353B
<b>32ED</b>	Comparaisons	0B	353B
<b>32EF</b>	Comparaisons	0C	353B
<b>32F1</b>	Comparaisons	0D	353B
<b>32F3</b>	Comparaisons	0E	353B
<b>32F5</b>	Addition	0F	3014
<b>32F7</b>	AND entre chaîne et numérique	10	352D
<b>32F9</b>	Comparaisons	11	353B
<b>32FB</b>	Comparaisons	12	353B
<b>32FD</b>	Comparaisons	13	353B
<b>32FF</b>	Comparaisons	14	353B
<b>3301</b>	Comparaisons	15	353B
<b>3303</b>	Comparaisons	16	353B
<b>3305</b>	Concaténation de chaînes	17	359C
<b>3307</b>	VAL\$	18	35DE
<b>3309</b>	USR " "	19	34BC
<b>330B</b>	Lecture	1A	3645
<b>330D</b>	Négation	1B	346E
<b>330F</b>	CODE	1C	3669
<b>3311</b>	VAL	1D	35DE
<b>3313</b>	LEN	1E	3674
<b>3315</b>	SIN	1F	37B5
<b>3317</b>	COS	20	37AA
<b>3319</b>	TAN	21	37DA
<b>331B</b>	ASN	22	3833
<b>331D</b>	ACS	23	3843
<b>331F</b>	ATN	24	37E2
<b>3321</b>	LN	25	3713
<b>3323</b>	EXP	26	36C4
<b>3325</b>	INT	27	36AF

	Opération	Code après RST28	Adresse routine
<b>3327</b>	SQR	28	384A
<b>3329</b>	SGN	29	3492
<b>332B</b>	ABS	2A	346A
<b>332D</b>	PEEK	2B	34AC
<b>332F</b>	IN	2C	34A5
<b>3331</b>	USR ( )	2D	34B3
<b>3333</b>	STR\$	2E	361F
<b>3335</b>	CHR\$	2F	35C9
<b>3337</b>	NOT	30	3501
<b>3339</b>	Duplique un nombre	31	33C0
<b>333B</b>	Modulo	32	36A0
<b>333D</b>	Saut	33	3686
<b>333F</b>	Stocke des données	34	33C6
<b>3341</b>	Décrément et saut si non 0	35	367A
<b>3343</b>	<0	36	3506
<b>3345</b>	>0	37	34F9
<b>3347</b>	Fin des calculs	38	369B
<b>3349</b>	Donne l'argument	39	3783
<b>334B</b>	Tronque un nombre	3A	3214
<b>334D</b>	Opération simple	3B	33A2
<b>334F</b>	Conversion exponentielle	3C	2D4F
<b>3351</b>	Sauve un nombre en virgule flottante	3D	3297
<b>3353</b>	Générateur de séries	3E	3449
<b>3355</b>	Sauve une constante	3F	341B
<b>3357</b>	Sauve dans la mémoire	40	342D
<b>3359</b>	Charge dans la mémoire	41	340F

- 335B** Routine de calcul des nombres en virgule flottante.
- 33A1** Routine de suppression d'une opérande en virgule flottante de la pile du calculateur.
- 33A2** Utilisée par 2757 pour les opérations simples.
- 33A9** Teste s'il reste au moins 5 octets de libres dans la pile du calculateur pour stocker un nombre en virgule flottante.
- 33B4** Stocke dans la paire de registres DE la variable système STKEND, place le nombre en virgule flottante dans la pile du calculateur puis replace dans STKEND le contenu de DE.
- 33C0** Déplace un nombre en virgule flottante en haut de la pile du calculateur.
- 33F7** Routine de recherche d'une constante (voir tableau en 32C5). La paire de registres HL doit contenir l'adresse de début de la table et le registre A le numéro de la constante. La routine place alors, dans la paire de registres HL, l'adresse de la constante.

ADRESSES

**ADRESSES DE LA ROM BASIC**  
(de 1000 à 3CFF)

- 3406 Routine de recherche de l'adresse d'un nombre en virgule flottante. Le résultat est placé dans la paire de registres HL.
- 340F Routine de déplacement d'un nombre en virgule flottante de la zone mémoire du calculateur dans la pile du calculateur (nouvelle valeur).
- 341B Recherche une constante et la sauve dans la dernière valeur de la pile du calculateur.
- 342D Routine identique à 340F, mais la source et la destination sont inversées.
- 343C Echange les valeurs des deux derniers nombres en virgule flottante de la pile du calculateur.
- 3449 Générateur de séries (polynômes de Chebyshev) utilisées pour les fonctions SIN, ATN, LN, EXP et les fonctions qui en découlent (COS, TAN, ASN, ↑ et SQR).
- 346A Met à 0 le bit de signe du dernier nombre en virgule flottante de la pile du calculateur (valeur absolue).
- 346E Change le bit de signe du dernier nombre en virgule flottante de la pile du calculateur (négation).
- 3492 Routine de la fonction SGN.
- 34A5 Routine de la fonction IN.
- 34AC Routine de la fonction PEEK.
- 34B3 Routine de la fonction USR suivie d'un nombre.
- 34BC Routine de la fonction USR suivie d'une chaîne de caractères.
- 34E9 Teste si un nombre en virgule flottante est nul. Si c'est le cas, la routine met à 1 le drapeau de retenue.
- 34F9 Routine de l'opération ">0".
- 3501 Routine de la fonction NOT.
- 3506 Routine de l'opération "<0".
- 350B Met la dernière valeur de la pile du calculateur à 0 si le drapeau de retenue est à 0 et à 1 s'il est à 1.
- 351B Routine de l'opération OR.
- 3524 Routine de l'opération AND entre deux nombres.
- 352D Routine de l'opération AND entre une chaîne de caractères et un nombre.
- 353B Routine des opérations de comparaisons.
- 359C Routine de concaténation de chaînes de caractères.

**ADRESSES DE LA ROM BASIC**  
(de 1000 à 3CFF)

- 35BF Place dans la paire de registres HL l'adresse du premier octet de la dernière valeur de la pile du calculateur (variable système STKEND-5) et dans DE celle de STKEND.
- 35C9 Routine de la fonction CHR\$.
- 35DE Routine des fonctions VAL et VAL\$.
- 361F Routine de la fonction STR\$.
- 3645 Routine de la fonction INKEY\$.
- 3669 Routine de la fonction CODE.
- 3674 Routine de la fonction LEN.
- 367A Effectue la fonction machine DJNZ, mais avec la variable système BERG (routine utilisée par le générateur de série).
- 3686 Effectue un saut inconditionnel (routine utilisée par 367A et 368F).
- 368F Effectue un saut conditionnel si la dernière valeur de la pile du calculateur (ou celle pointée par la paire de registres DE) est différente de zéro.
- 369B Fin de la liste de calculs effectuée par RST0028. Cette dernière a rencontré un code 38.
- 36A0 Routine de calcul d'un modulo pour les nombres aléatoires.
- 36AF Routine de la fonction INT.
- 36C4 Routine de la fonction EXP.
- 3713 Routine de la fonction LN.
- 3783 Réduit l'argument de SIN ou COS avant les calculs.
- 37AA Routine de la fonction COS.
- 37B5 Routine de la fonction SIN.
- 37DA Routine de la fonction TAN.
- 37E2 Routine de la fonction ATN.
- 3833 Routine de la fonction ASN.
- 3843 Routine de la fonction ACS.
- 384A Routine de la fonction SQR.
- 3851 Routine de la fonction ^.
- 3D00 Table de représentation graphique des caractères de l'espace (20) au copyright (7F). Ce tableau va de 3D00 à 3FFF.

VARIABLES SYSTEME (de 5C00 à 5CB5)

Ces variables sont utilisées en permanence par le Basic. Elles peuvent être modifiées par vos programmes en langage machine ou en Basic avec cependant quelques réserves :

- O : peut être modifiée par vos programmes en Basic ou L.M.
- N : ne peut être modifiée ("plante" votre Spectrum!).
- RIEN : le Spectrum change immédiatement cette valeur.

Adresse Hex.	Adresse Dec.	Nombre d'octets	Modif.	Nom de la variable	Définition
5C00	23552	8	N	KSTATE	Contient les différents états du clavier. Cette variable est utilisée par la routine de gestion du clavier (02BF) et celle de la commande NEW (11B7).
5C08	23560	1		LAST-K	Code de la dernière touche enfoncée. Cette variable est mise à jour par la routine de gestion du clavier (02BF).
5C09	23561	1	O	REPDEL	Durée maximum pendant laquelle une touche peut être appuyée sans qu'il y ait de répétition (environ 0./seconde). Cette variable est utilisée par la routine de gestion du clavier (02BF) et celle de la commande NEW (11B7).
5COA	23562	1	O	REPPER	Délai entre les répétitions quand une touche est maintenue appuyée (environ 0.1 seconde). Cette variable est utilisée par la routine de répétition d'une touche (0310).
5COB	23563	2		DEFADD	Adresse des arguments d'une fonction utilisateur. Cette variable est utilisée par la routine de la commande FN (27BD).
5COD	23565	1		K-DATA	Stocke le deuxième octet de contrôle des couleurs entrées au clavier. Cette variable est utilisée par la routine de la dernière touche appuyée (10A8).
5COE	23566	2		TVDATA	Stocke la couleur utilisée par les commandes AT et TAB. Cette variable est utilisée par la routine des caractères de contrôle avec opérande (0A60).

VARIABLES SYSTEME (de 5C00 à 5CB5)

Adresse Hex.	Adresse Dec.	Nombre d'octets	Modif.	Nom de la variable	Définition
5C10	23568	38	N	STRMS	Adresse des canaux.
5C36	23606	2	O	CHARS	Adresse (-255) des définitions graphiques de chaque caractère de l'espace au symbole copyright. Cette variable est mise à jour par la commande NEW (11B7) et contient alors 3C00. Vous pouvez la modifier pour pointer la mémoire vive et redessiner tout le jeu de caractères.
5C38	23608	1	O	RASP	Durée du signal sonore (mis à jour par la routine de la commande NEW en 11B7).
5C39	23609	1	O	PIP	Longueur du "clic" lors de la manœuvre d'une touche du clavier (mise à jour par la routine de la commande NEW en 11B7).
5C3A	23610	1	O	ERR-NR	Code de l'erreur rencontrée. Cette valeur correspond à 1 moins le code erreur. S'il n'y a pas d'erreur, on trouve dans cette variable 255d ou FF hex (PRINT PEEK(23610) donne toujours 255). Cette variable est utilisée par la routine de gestion d'erreur (0053) et celle de la commande INPUT (2089).
5C3B	23611	1	N	FLAGS	Cette variable contient les différents drapeaux utilisés par les routines en 02BF, 0A23, 0A4F, 0A6D, 0ADC, 0C55, 11B7, 1795, 1F3A, 1F60, 2089, 2AEE et 35DE.
5C3C	23612	1	N	TV-FLAGS	Drapeau de contrôle de l'écran. Cette variable est modifiée par les routines de contrôle de SAVE (0970), de sauvegarde du numéro de ligne et de colonne (0ADC), de test du Scroll (0C55), de sauvegarde de la couleur (0D4D), d'effacement d'une ligne d'écran (0E44), de la commande NEW (11B7), des commandes LIST et LLIST (1795) et de la commande INPUT (2089).

ADDRESSES

Adresse Hex.	Adresse Dec.	Nombre d'octets	Modif.	Nom de la variable	Définition
5C3D	23613	2	N	ERR-SP	Adresse de la pile machine utilisée en cas d'erreur. Cette variable est utilisée par les routines de l'éditeur (0F2C), de la commande NEW (11B7), de la commande CLEAR (1EAC), de la commande GO SUB (1EED), de la commande RETURN (1F23) et la commande INPUT (2089).
5C3F	23615	2		LIST-SP	Pointeur de l'adresse de retour après une liste automatique. Cette variable est utilisée par la routine de test du Scroll (0C55) et celle des commandes LIST et LLIST (1795).
5C41	23617	1		MODE	Type de curseur sur l'écran (E, C, K, L ou G). Cette variable est utilisée par les routines de gestion du clavier (02BF), de l'éditeur (0F2C), d'ajout d'un caractère dans une édition (0F81) et d'affichage du curseur (18E1).
5C42	23618	2	0	NEWPPC	Numéro de la ligne où doit être effectué un saut. Il est possible de modifier NEWPPC et NSPPC pour forcer un saut à une ligne donnée. Cette variable est utilisée par les routines de contrôle de LOAD (0808) de la commande FOR (1D03) et de la commande GO TO (1E67).
5C44	23620	1	0	NSPPC	Numéro d'instruction dans une ligne où doit être effectué un saut (les instructions dans une ligne sont différenciées par le sépareur ":"). Cette variable est utilisée par les routines de contrôle de LOAD (0808), de la commande FOR (1D03) et de la commande GO TO (1E67). Pour son utilisation, voir NEWPPC.

Adresse Hex.	Adresse Dec.	Nombre d'octets	Modif.	Nom de la variable	Définition
5C45	23621	2	0	PPC	Numéro de la ligne en cours d'exécution. Cette variable est utilisée par la routine de la commande FOR (1D03) et celle de la commande GO SUB (1EED).
5C47	23623	1	0	SUBPPC	Numéro de l'instruction dans la ligne en cours d'exécution. Cette variable est utilisée par la routine de la commande FOR (1D03) et celle de la commande GO SUB (1EED).
5C48	23624	1	0	BORDCR	Couleur (multipliée par 8) du bord et de la partie inférieure de l'écran. Cette variable est utilisée par les routines de modification des couleurs dans SAVE et LOAD (053F), de sauvegarde des couleurs (0D4D) et d'effacement d'une ligne d'écran (0E44).
5C49	23625	2	0	E-PPC	Numéro de la ligne pointée par le curseur. Cette variable est utilisée par la routine de gestion de la touche EDIT (0FA9) et celle de la commande LIST et LLIST (1795).
5C4B	23627	2	N	VARS	Pointeur de la table des variables utilisées par le Basic. Cette variable est utilisée par les routines des commandes SAVE et LOAD (0605), de contrôle de LOAD (0808), de contrôle de MERGE (08B6), de la commande NEW (11B7) et de la commande CLEAR (1EAC).
5C4D	23629	2		DEST	Adresse de la variable en cours d'affectation. Cette variable est modifiée par la routine de la commande LET (2AEE).
5C4F	23631	2	N	CHANS	Adresse des données des différents canaux. Cette variable est mise à jour par la routine de la commande NEW (11B7).

VARIABLES SYSTEME (de 5C00 à 5CB5)

Adresse Hex.	Adresse Dec.	Nombre d'octets	Modif.	Nom de la variable	Définition
5C51	23633	2	N	CURCHL	Adresse des informations utilisées pour les entrées/sorties. Cette variable est utilisée par les routines de gestion des caractères de contrôle avec opérandes (0A6D), de la commande CLS (0D6B), d'effacement d'écran (0DAF), de gestion de la touche EDIT (0FA9) et de la commande STR\$ (361F).
5C53	23635	2	N	PROG	Adresse du programme Basic. Cette variable est utilisée par les routines des commandes SAVE et LOAD (0605), de contrôle de la commande MERGE (08B6), de MERGE d'une ligne ou d'une variable (092C), de la commande NEW (11B7) et de recherche d'adresse d'une ligne Basic (196E).
5C55	23637	2	N	NXTLIN	Adresse de la prochaine ligne à exécuter. Cette variable est utilisée par la routine de la commande REM (1BB2) et celle de la commande FOR (1D03).
5C57	23639	2	N	DATADD	Adresse de fin de la dernière donnée (DATA). Cette variable est utilisée par les routines de la commande NEW (11B7), de la commande READ (1DEC) et de la commande RESTORE (1E42).
5C59	23641	2	N	E-LINE	Pointeur de l'adresse de la zone de travail de l'éditeur. Cette variable est utilisée par les routines des commandes SAVE et LOAD (0605), de contrôle de LOAD (0808), de la commande NEW (11B7), de lecture d'un numéro de ligne (19FB), de la commande LET (2AEE) et de la commande DIM (2C02).
5C5B	23643	2	0	K-CUR	Adresse du curseur sur l'écran. Cette variable est utilisée par les routines de l'éditeur (0F2C), d'ajout d'un caractère dans une

VARIABLES SYSTEME (de 5C00 à 5CB5)

Adresse Hex.	Adresse Dec.	Nombre d'octets	Modif.	Nom de la variable	Définition
5C5D	23645	2	N	CH-ADD	édition (0F81), d'affichage du curseur (18E1), de la commande INPUT (2089) et de la fonction STR\$ (361F).
5C5F	23647	2	0	X-PTR	Adresse du prochain caractère à interpréter. Cette variable est utilisée par les routines RST0008, RST0018, de recherche du prochain caractère (0074), de lecture d'un numéro de ligne (19FB), de la commande FOR (1D03), de la commande INPUT (2089) et des fonctions VAL et VAL\$ (35DE).
5C61	23649	2	N	WORKSP	Adresse du caractère précédant le symbole d'erreur de syntaxe (point d'interrogation) dans l'entrée d'une ligne Basic ou dans l'exécution de la commande INPUT. Cette variable est utilisée par les routines de contrôle de LOAD (0808), de MERGE d'une ligne ou d'une variable (092C), de la commande READ (1DEC) et de la commande INPUT (2089).
5C63	23651	2	N	STKBOT	Adresse de l'espace de travail temporaire. Cette variable est utilisée par les routines RST0030, de la commande NEW (11B7), de réservation mémoire (169E) et de la commande INPUT (2089).
5C65	23653	2	N	STKTOP	Pointeur de l'adresse de début de la pile du calculateur utilisée pour les calculs des nombres en virgule flottante. Cette variable est utilisée par les routines de la commande NEW (11B7), de réservation mémoire (169E) et de la commande INPUT (2089).

ADDRESSES

Adresse Hex.	Adresse Dec.	Nombre d'octets	Modif.	Nom de la variable	Définition
5C65	23653	2	N	STKEND	Pointeur de l'adresse de fin de pile du calculateur. Cette variable est utilisée par les routines de la commande NEW (11B7), de lecture d'un numéro de ligne (19FB), de la commande CLEAR (1EAC), de test de place mémoire (1F05) et de calcul en virgule flottante (335B).
5C67	23655	1		BREG	Sauvegarde du registre B par le calculateur dans l'exécution d'un pseudo DJNZ (voir codes machine). Cette variable est utilisée par la routine de calcul des nombres en virgule flottante (335B) et celle d'exécution d'un DJNZ (367A).
5C68	23656	2		MEM	Adresse de la pile spéciale du calculateur (généralement MEMBOT). Cette variable est utilisée par la routine de la commande FOR (1D03).
5C6A	23658	1	0	FLAGS2	Drapeau utilisé par les routines de décodage clavier (0333), d'effacement d'écran (ODAF), d'effacement du tampon d'imprimante (DEDF), des commandes LIST et LLIST (1795), d'affichage du curseur (18E1) et de la commande INPUT (2089).
5C6B	23659	1	N	DF-SZ	Nombre de lignes vides en dessous de l'écran (y compris les lignes vides). Cette variable est utilisée par les routines de test du Scroll (0C55), de la commande CLS (0D6B), de la commande NEW (11B7), des commandes LIST et LLIST (1795) et de la commande INPUT (2089).
5C6C	23660	2	0	S-TOP	Numéro de la ligne la plus haute dans une ligne automatique. Cette variable est utilisée par la routine des commandes LIST et LLIST (1795).

Adresse Hex.	Adresse Dec.	Nombre d'octets	Modif.	Nom de la variable	Définition
5C6E	23662	2	0	OLDPPC	Numéro de la ligne à laquelle l'instruction CONTINUE doit repartir. Cette variable est utilisée par la routine de la commande CONTINUE (15EF).
5C70	23664	1	0	OSPPC	Numéro de l'instruction dans une ligne à laquelle l'instruction CONTINUE doit repartir. Cette variable est utilisée par la routine de la commande CONTINUE (15EF).
5C71	23665	1		FLAGX	Drapeau utilisé par les routines de l'éditeur (0F2C), de la touche EDIT (0FA9), de la commande INPUT (2089) et de la commande !ET (2AEE).
5C72	23666	2		STRLEN	Nombre de caractères à interpréter. Cette variable est utilisée par la routine de la commande FOR (1D03) et celle de la commande LET (2AEE).
5C74	23668	2		T-ADDR	Adresse de la prochaine rubrique dans la table de syntaxe. Cette variable est utilisée par la routine des commandes SAVE et LOAD (0605).
5C76	23670	2	0	SEED	Base de la série des nombres aléatoires (fonction RND). Cette variable est déterminée par l'instruction RANDOMIZE (routine en 1E4F).
5C78	23672	3	0	FRAMES	Compteur de trames du téléviseur (incrémenté toutes les 20 ms). Cette variable est utilisée par la routine RST0038 et celle de la commande RANDOMIZE (1E4F).
5C7B	23675	2	0	UDG	Adresse du premier caractère graphique. Cette variable est mise à jour par la commande NEW (11B7).
5C7D	23677	1	0	COORDS	Coordonnées X du dernier pixel affiché. Cette variable est utilisée par les routines d'effacement d'écran (ODAF), de la commande

Adresse Hex.	Adresse Dec.	Nombre d'octets	Modif.	Nom de la variable	Définition
5C7E	23678	1	0	' '	PLOT (22DC), de la commande CIRCLE (2320) et de la commande DRAW (2382). Coordonnée Y du dernier pixel affiché.
5C7F	23679	1	0	P-POSN	Position de la tête d'impression de l'imprimante, c'est-à-dire un des 33 caractères du tampon. Cette variable est utilisée par la routine de sauvegarde ligne/colonne (OADC).
5C80	23680	1	0	PR-CC	Pointeur du prochain caractère à imprimer dans le tampon BRBUFF. Cette variable est utilisée par la routine de sauvegarde ligne/colonne (OADC) et celle d'effacement du tampon d'imprimante (OEDF).
5C81	23681	1	0	-----	Inutilisé par le Basic du Spectrum. Vous pouvez utiliser cet octet dans vos programmes en langage machine.
5C82	23682	2	0	ECHO-E	Numéro de la dernière colonne utilisée lors d'une entrée (sur la dernière ligne de l'écran). Cette variable est utilisée par la routine de sauvegarde ligne/colonne (OADC) et celle de la commande INPUT (2089).
5C84	23684	2	0	DF-CC	Adresse de la prochaine position d'écriture par l'instruction PRINT. Cette variable est utilisée par la routine de sauvegarde ligne/colonne (OADC).
5C86	23686	2	0	DFCCL	Identique à DF-CC, mais pour la partie inférieure de l'écran.
5C88	23688	1	N	S-POSN	Numéro de la colonne de la prochaine instruction PRINT. Cette variable est utilisée par la routine de la commande INPUT (2089).
5C89	23689	1	N	' '	Numéro de la ligne de la prochaine instruction PRINT.

Adresse Hex.	Adresse Dec.	Nombre d'octets	Modif.	Nom de la variable	Définition
5C8A	23690	2	N	SPOSNL	Identique à S-POSN, mais pour la partie inférieure de l'écran.
5C8C	23692	1	0	SCR-CT	Compteur du nombre de lignes à imprimer sur l'écran avant l'apparition du message "Scroll?". Cette variable est utilisée par les routines de test du Scroll (0C55), d'effacement d'écran (ODAF) et de la commande INPUT (2089).
5C8D	23693	1	0	ATTR-P	Numéro de la couleur permanente. Cette variable est utilisée par les routines de sauvegarde couleur (ODAD), de la commande CLS (0D6B), d'effacement de ligne (0E44) et de la commande NEW (11B7).
5C8E	23694	1	0	MASK-P	Utilisée pour les couleurs transparentes. Les bits à 1 spécifient que les bits d'attribut correspondants ne proviennent pas de la variable système ATTR-P, mais qu'ils sont déjà placés sur l'écran.
5C8F	23695	1		ATTR-T	Numéro de la couleur temporaire (à l'intérieur d'une instruction PRINT par exemple). Cette variable est utilisée par les routines de mise à jour de l'octet d'attribut (OBDB) de test du Scroll (0C55), de sauvegarde couleur (OD4D), de la commande NEW (11B7) et de gestion des caractères clignotants (18C1).
5C90	23696	1		MASK-T	Identique à MASK-P, mais pour les valeurs temporaires.
5C91	23697	1	0	P-FLAG	Drapeau utilisé par les routines de gestion du curseur droite (0A3D), de mise à jour de l'octet d'attribut (OBDB), de test du Scroll (0C55), de gestion des caractères clignotants (18C1) et de la commande PLOT (22DC).

Adresse Hex.	Adresse Dec.	Nombre d'octets	Modif.	Nom de la variable	Définition
5C92	23698	30		MEMBOT	Zone de travail pour les nombres utilisés par le calculateur et qu'il ne peut pas placer dans sa pile.
5CB0	23728	2	0	-----	Inutilisé par le Basic du Spectrum. Vous pouvez utiliser ces deux octets pour vos programmes en langage machine.
5CB2	23730	2	0	RAMTOP	Adresse du dernier octet utilisé par le Basic. Cette variable peut être modifiée pour réserver de la place mémoire pour vos programmes en langage machine par l'instruction CLEAR suivie de la nouvelle valeur de RAMTOP. Cette variable est modifiée par la routine de la commande NEW (11B7) et celle de la commande CLEAR (1EAC).
5CB4	23732	2	0	P-RAMT	Adresse du tout dernier octet de la mémoire vive (32767 pour la version 16 ko et 65535 pour la version 48 ko). Cette variable est modifiée par la routine de la commande NEW (11B7) et celle de la commande CLEAR (1EAC).

# TRUCS ET ASTUCES

Desensibiliser la touche Break. Poke 23659,0

## CHAINAGE DE PROGRAMME

En utilisant la commande SAVE LINE, il est possible de chaîner plusieurs programmes (découpage d'un programme en plusieurs parties s'il ne tient pas entièrement en mémoire). Il suffit de sauver tous les morceaux de programme par SAVE "PROGRAMME" LINE et d'ajouter des instructions LOAD "PROGRAMME SUIVANT" pour charger et exécuter automatiquement la suite. Exemple :

### Programme 1

```
10 REM DEBUT
20 PRINT "EXECUTION PREMIERE PARTIE"
30 ...
...
1000 LOAD "PROGRAMME 2"
```

### Programme 2

```
10 REM DEBUT
20 PRINT "EXECUTION DEUXIEME PARTIE"
30 ...
...
1000 LOAD "PROGRAMME 3"
... ETC ...
```

## SAUVEGARDE RAPIDE

L'instruction SAVE sauvant les lignes de programme mais aussi les variables Basic, il peut être intéressant de taper directement au clavier l'instruction CLEAR avant d'effectuer la sauvegarde. CLEAR effaçant les variables et les tableaux, on gagne jusqu'à 50 % du temps de sauvegarde (surtout s'il existe des instructions comme DIM A(100,100) dans votre programme).

## INSTRUCTION VAL

Il est possible d'utiliser la fonction VAL d'une chaîne de caractères contenant une expression complexe ou utilisant des

variables Basic. On peut ainsi constituer des fonctions mathématiques entrées lors d'un INPUT pour les exécuter avec des variables déjà dans le programme. Pour obtenir les valeurs de 1 à 10 de la fonction  $Ax+B$ , il faut écrire le programme suivant :

```
10 LET A=10
20 LET B=5
30 FOR X=1 TO 10
40 PRINT A*X+B
50 NEXT X
```

Mais si vous désirez changer de fonction, il faut modifier la ligne 40. Avec l'instruction VAL il est possible d'écrire ceci :

```
10 INPUT A
20 INPUT B
30 INPUT FONCS$
40 FOR X=1 TO 10
50 PRINT VAL FONCS$
60 NEXT X
```

#### LANGAGE MACHINE DANS UNE INSTRUCTION REM

C'est la solution la plus simple pour les programmes en L.M. de petite taille. La première adresse de la routine sera 16514 puisque la mémoire vive utilisée par le Basic commence en 16509, le Basic prenant deux octets pour le numéro de ligne, deux octets pour sa longueur et un octet pour l'instruction REM. Vous devez placer après l'instruction REM autant de caractères qu'en comprend votre sous-programme en langage machine. Exemple :

```
10 REM 0123456789
20 LET A=PEEK 23635+256*PEEK 23636 +5
30 POKE A,201
40 LET L=USR A
```

(cet exemple effectue uniquement une instruction machine RET).

*Remarque* : L'instruction REM doit toujours être la première ligne du programme pour qu'elle ne soit pas déplacée par le Basic.

#### LANGAGE MACHINE EN HAUT DE MEMOIRE

Il suffit pour cela de modifier la variable système RAMTOP. Si vous avez besoin de 200 octets, vous devrez taper directement au clavier :

```
CLEAR (PEEK 23730+256*PEEK 23731)-200
```

*Remarque* : le sous-programme en langage machine ainsi établi n'est pas sauvé par l'instruction SAVE mais par SAVE "PGM LM" CODE adresse, longueur.

#### HORLOGE TEMPS REEL

La variable système FRAMES compte le nombre de trames affichées sur l'écran. Cette variable est donc incrémentée 50 fois par seconde (20 ms) et peut servir de compteur pour une horloge (précision du secteur) :

```
10 CLS
20 LET T=0
30 INPUT "H, M, S ?";H, M, S
40 POKE 23672,0
50 POKE 23673,0
60 POKE 23674,0
70 LET C=PEEK 23672+256*PEEK 23673+65536*PEEK 23674
80 IF T+50>C THEN GO TO 70
90 LET T=T+50
100 LET S=S+1
110 IF S<60 THEN GOTO 170
120 LET S=0:LET M=M+1
130 IF M<60 THEN GO TO 170
140 LET M=0:LET H=H+1
150 IF H<24 THEN GO TO 170
160 LET H=0
170 PRINT AT 10,10;H;" ":"M;" ":"S;" "
180 GO TO 70
```

#### NOMBRE D'OCTETS LIBRES

Pour connaître le nombre d'octets libres en mémoire vive, il faut taper en mode direct ou dans un programme Basic la ligne suivante :

```
PRINT 65535-USR 7962; "Octets libres" → 41572
```

#### LONGUEUR D'UN PROGRAMME

Pour connaître le nombre d'octets occupés par le programme Basic en mémoire, il faut taper en mode direct ou dans un programme Basic la ligne suivante :

```
PRINT (PEEK 23627+256*PEEK 23628)
-(PEEK 23635+256*PEEK 23636)
```

## EDITION CONTINUE

Il est toujours gênant d'avoir le message "Scroll?" au milieu d'une édition sur l'écran. La variable système SCR-CT donne au Basic le nombre de lignes à imprimer avant ce message. Le programme qui suit donne un listing continu quelque soit le nombre de lignes à imprimer :

```
10 FOR I=1 TO 1000
20 POKE 23692,255
30 PRINT I;" ";I*2;" ";I2
40 NEXT I
```

## CLIC DU CLAVIER

Il est possible de modifier le "clic" que le Spectrum fait entendre quand une touche est enfoncée. La variable système PIP définit la longueur de ce "clic". Initialement elle est à zéro mais, en lui donnant la valeur 128, le son est beaucoup plus audible :

```
POKE 23609,128
```

*Remarque* : toutes les valeurs entre 0 et 255 peuvent convenir, mais la saisie est ralentie quand on s'approche de 255.

## DELAI AVANT REPETITION

Il est possible de modifier le délai avant lequel une touche appuyée doit être répétée. C'est la variable système REPDEL qui définit ce délai. Initialement cette variable contient 35 soit 0.7 seconde :

```
POKE 23561,20 pour les as du clavier (0.4 seconde)
POKE 23561,75 pour les débutants (1.5 seconde)
```

Pour supprimer les répétitions du clavier, il faut taper :

```
POKE 23561,0
```

## DELAI ENTRE REPETITION

C'est la variable système REPPER qui définit le délai entre deux répétitions. Initialement elle contient la valeur 5, soit

0.1 seconde) :

```
POKE 23562,2 pour les as (0.04 seconde)
POKE 23562,15 pour les débutants (0.3 seconde)
```

## RENUMEROTATION

Le programme simple qui suit permet de renuméroter les lignes d'un programme Basic. Il est placé spécialement à partir de la ligne 9990 pour ne pas gêner le programme à renuméroter (il est à utiliser avec la commande MERGE). Pour l'exécution, taper RUN 9990 ou GO TO 9990 en mode direct :

```
9990 INPUT "NUMERO LIGNE DEBUT ?";D
9991 INPUT "INCREMENT ?";I
9992 LET A=PEEK 23635+256*PEEK 23636
9993 IF PEEK(A+1)+256*PEEK(A)=9990 THEN STOP
9994 POKE A,INT (D/256)
9995 POKE A+1,D-256*INT (D/256)
9996 LET D=D+I
9997 LET A=4+A+PEEK (A+2)+256*PEEK (X+3)
9999 GO TO 9993
```

*Remarque* : ce programme ne modifie pas les numéros de ligne des commandes GO TO et GO SUB.

## 117 CARACTERES UTILISATEUR

Tous les caractères utilisés par le Spectrum ont leur définition graphique à la fin de la ROM du Basic (de 3000 à 3FFF pour les caractères allant de l'espace au symbole copyright). La variable système CHARS pointe cette table en ROM (adresse de la table-256). Pour obtenir 96 caractères, définissables par l'utilisateur, il suffit que CHARS pointe un emplacement de la RAM (à partir de RAM-TOP, par exemple). Le programme suivant recopie le jeu de caractères dans la RAM et modifie en conséquence le pointeur CHARS. Pour démonstration, on place le dessin d'un carré pour le caractère espace et pour le graphique utilisateur "A" :

```
10 CLEAR (PEEK 23730+256*PEEK 23731)-770
20 LET A=PEEK 23730+256*PEEK 23731
30 FOR I=1 TO 768
40 POKE A+I,PEEK(15616+I)
50 NEXT I
60 LET A=A-155
```

Nouveau RAMTOP  
A = RAMTOP  
Copie le jeu de caractères  
CHARS=RAMTOP-256

70 POKE 23607,INT (A/256)  
 80 POKE 23606,A-256\*INT (A/256)  
 90 LET A=A+256  
 100 DATA 255,129,129,129,129,129,255  
 110 FOR I=0 TO 7  
 120 READ C:POKE A+I,C  
 130 POKE USR "A"+I,C  
 140 NEXT I

A=Adresse du jeu de caractères

L'espace devient un carré

Si le programme n'a pas d'erreur, vous devez avoir la ligne de rapport suivante :

Ø □ OK, □ 14Ø:1

Le graphique utilisateur A doit aussi vous donner un carré.

Maintenant vous disposez d'un jeu de 117 caractères programmables (les 96 du jeu initial plus les 21 graphiques utilisateur).

## LISTE DES ATTRIBUTS VIDEO

De 0 à 83 : brillance normale : sans clignotement

		PAPER (couleur du papier)							
		NOIR	BLEU	ROUGE	MAGENTA	VERT	TURQUOISE	JAUNE	BLANC
INK (couleur de l'encre)	NOIR	Ø	8	16	24	32	40	48	56
	BLEU	1	9	17	25	33	41	49	57
	ROUGE	2	10	18	26	34	42	50	58
	MAGENTA	3	11	19	27	35	43	51	59
	VERT	4	12	20	28	36	44	52	60
	TURQUOISE	5	13	21	29	37	45	53	61
	JAUNE	6	14	22	30	38	46	54	62
	BLANC	7	15	23	31	39	47	55	63

De 64 à 127 : brillance forte / sans clignotement

		PAPER (couleur du papier)							
		NOIR	BLEU	ROUGE	MAGENTA	VERT	TURQUOISE	JAUNE	BLANC
INK (couleur de l'encre)	NOIR	64	72	80	88	96	104	112	120
	BLEU	65	73	81	89	97	105	113	121
	ROUGE	66	74	82	90	98	106	114	122
	MAGENTA	67	75	83	91	99	107	115	123
	VERT	68	76	84	92	100	108	116	124
	TURQUOISE	69	77	85	93	101	109	117	125
	JAUNE	70	78	86	94	102	110	118	126
	BLANC	71	79	87	95	103	111	119	127

de 128 à 191 : brillance normale / avec clignotement

		PAPER (couleur du papier)							
		NOIR	BLEU	ROUGE	MAGENTA	VERT	TURQUOISE	JAUNE	BLANC
INK (couleur de l'encre)	NOIR	128	136	144	152	160	168	176	184
	BLEU	129	137	145	153	161	169	177	185
	ROUGE	130	138	146	154	162	170	178	186
	MAGENTA	131	139	147	155	163	171	179	187
	VERT	132	140	148	156	164	172	180	188
	TURQUOISE	133	141	149	157	165	173	181	189
	JAUNE	134	142	150	158	166	174	182	190
	BLANC	135	143	151	159	167	175	183	191

de 192 à 255 : brillance forte / avec clignotement

		PAPER (couleur du papier)							
		NOIR	BLEU	ROUGE	MAGENTA	VERT	TURQUOISE	JAUNE	BLANC
INK (couleur de l'encre)	NOIR	192	200	208	216	224	232	240	248
	BLEU	193	201	209	217	225	233	241	249
	ROUGE	194	202	210	218	226	234	242	250
	MAGENTA	195	203	211	219	227	235	243	251
	VERT	196	204	212	220	228	236	244	252
	TURQUOISE	197	205	213	221	229	237	245	253
	JAUNE	198	206	214	222	230	238	246	254
	BLANC	199	207	215	223	231	239	247	255

Binaire	Décimal	Hexadécimal	Octal
0000	0	0	0
0001	1	1	1
0010	2	2	2
0011	3	3	3
0100	4	4	4
0101	5	5	5
0110	6	6	6
0111	7	7	7
1000	8	8	10
1001	9	9	11
1010	10	A	12
1011	11	B	13
1100	12	C	14
1101	13	D	15
1110	14	E	16
1111	15	F	17

Exemple : D59 hexa → 1101 0101 1110 1001 binaire

Tableau 1 : octet bas (de 0 à 255)

Dec.	Hexa	Dec.	Hexa	Dec.	Hexa	Dec.	Hexa
0	00	44	2C	88	58	132	84
1	01	45	2D	89	59	133	85
2	02	46	2E	90	5A	134	86
3	03	47	2F	91	5B	135	87
4	04	48	30	92	5C	136	88
5	05	49	31	93	5D	137	89
6	06	50	32	94	5E	138	8A
7	07	51	33	95	5F	139	8B
8	08	52	34	96	60	140	8C
9	09	53	35	97	61	141	8D
10	0A	54	36	98	62	142	8E
11	0B	55	37	99	63	143	8F
12	0C	56	38	100	64	144	90
13	0D	57	39	101	65	145	91
14	0E	58	3A	102	66	146	92
15	0F	59	3B	103	67	147	93
16	10	60	3C	104	68	148	94
17	11	61	3D	105	69	149	95
18	12	62	3E	106	6A	150	96
19	13	63	3F	107	6B	151	97
20	14	64	40	108	6C	152	98
21	15	65	41	109	6D	153	99
22	16	66	42	110	6E	154	9A
23	17	67	43	111	6F	155	9B
24	18	68	44	112	70	156	9C
25	19	69	45	113	71	157	9D
26	1A	70	46	114	72	158	9E
27	1B	71	47	115	73	159	9F
28	1C	72	48	116	74	160	A0
29	1D	73	49	117	75	161	A1
30	1E	74	4A	118	76	162	A2
31	1F	75	4B	119	77	163	A3
32	20	76	4C	120	78	164	A4
33	21	77	4D	121	79	165	A5
34	22	78	4E	122	7A	166	A6
35	23	79	4F	123	7B	167	A7
36	24	80	50	124	7C	168	A8
37	25	81	51	125	7D	169	A9
38	26	82	52	126	7E	170	AA
39	27	83	53	127	7F	171	AB
40	28	84	54	128	80	172	AC
41	29	85	55	129	81	173	AD
42	2A	86	56	130	82	174	AE
43	2B	87	57	131	83	175	AF

Dec.	Hexa	Dec.	Hexa	Dec.	Hexa	Dec.	Hexa
176	B0	196	C4	216	D8	236	EC
177	B1	197	C5	217	D9	237	ED
178	B2	198	C6	218	DA	238	EE
179	B3	199	C7	219	DB	239	EF
180	B4	200	C8	220	DC	240	FO
181	B5	201	C9	221	DD	241	F1
182	B6	202	CA	222	DE	242	F2
183	B7	203	CB	223	DF	243	F3
184	B8	204	CC	224	E0	244	F4
185	B9	205	CD	225	E1	245	F5
186	BA	206	CE	226	E2	246	F6
187	BB	207	CF	227	E3	247	F7
188	BC	208	D0	228	E4	248	F8
189	BD	209	D1	229	E5	249	F9
190	BE	210	D2	230	E6	250	FA
191	BF	211	D3	231	E7	251	FB
192	C0	212	D4	232	E8	252	FC
193	C1	213	D5	233	E9	253	FD
194	C2	214	D6	234	EA	254	FE
195	C3	215	D7	235	EB	255	FF

Tableau 2 : octet haut (de 0 à 65280)

Dec.	Hexa	Dec.	Hexa	Dec.	Hexa	Dec.	Hexa
0	00	4864	13	9728	26	14592	39
256	01	5120	14	9984	27	14848	3A
512	02	5376	15	10240	28	15104	3B
768	03	5632	16	10496	29	15360	3C
1024	04	5888	17	10752	2A	15616	3D
1280	05	6144	18	11008	2B	15872	3E
1536	06	6400	19	11264	2C	16128	3F
1792	07	6656	1A	11520	2D	16384	40
2048	08	6912	1B	11776	2E	16640	41
2304	09	7168	1C	12032	2F	16896	42
2560	0A	7424	1D	12288	30	17152	43
2816	0B	7680	1E	12544	31	17408	44
3072	0C	7936	1F	12800	32	17664	45
3328	0D	8192	20	13056	33	17920	46
3584	0E	8448	21	13312	34	18176	47
3840	0F	8704	22	13568	35	18432	48
4096	10	8960	23	13824	36	18688	49
4352	11	9216	24	14080	37	18944	4A
4608	12	9472	25	14336	38	19200	4B



OVER	15,69
PAPER	15,69,76
PAUSE	15,76
PEEK	11,83,84
P-FLAG	69,95
PI	9,81
EXP	9,81,82,84,85
FLAGS	67,75,77,87
FLAGS2	72,92
FLAGSX	93
FLASH	13,69,76
FN	13,78
FOR	13,75
FORMAT	72
FRAMES	93,99
GO SUB	13
GO TO	13,76
IF	13,75
IN	11,83,84
INK	13,69,76
INKEY\$	13,85
INPUT	14,70,71,76
INT	9,82,85
INVERSE	14,69,76
K-CUR	90
K-DATA	86
K-STATE	86
LAST-K	86
LEN	10,82,85
LET	14,79
LINE	14
LIST	14,70,72
LIST-SP	88
LLIST	14,72
LN	9,81,82,84,85
LOAD	14,68,97
LPRINT	15,76
MASK-P	95
MASK-T	69,95
MEM	92
MEMBOT	96
MERGE	15,68
MODE	88
MOVE	72
NEW	15,71
NEW-PPC	88
NEXT	15,75
NSPPC	98
NOT	18,83,84

NTXLIN	75,90
OLDPPC	93
OPEN f	72
OR	18,78,82
OSPPC	93
OUT	11,76
PIP	87,100
PLOT	15,29,77
POINT	15,77
POKE	11,76
PPC	75,89
P-POSN	94
PR-CC	94
P-RAMT	65,96
PRINT	15,69,76,80
PROG	65,90
RAMTOP	65,71,96,101
RANDOMIZE	16,76
RASP	87
READ	16,75
REM	10,75,98
REPEL	67,86,100
REPPER	67,86,100
RESTORE	16,76
RETURN	16,76
RND	16
RUN	16,70,76
SAVE	16,68,97
SCR-CT	95,100
SCREEN\$	17,77
Scroll	69,70
SEED	93
SGN	10,83,84
SIN	10,82,84,85
S-POSN	94
SPOSNL	95
SQR	10,83,84,85
STEP	13
STKBOT	65,91
STKEND	65,83,85,92
STOP	17,75
S-TOP	73,92
STR\$	10,80,83,85
STRLEN	75,93
STRMS	87
SUBPPC	89
TAB	16,69
T-ADDR	68,93
TAN	10,82,84,85

THEN	13
TO	10
TVDATA	86
TV-FLAGS	87
UDG	93
USR	11,82,83,84
VAL	10,82,85,97
VAL\$	10,82,85
VARS	65,89
VERIFY	17,68
WORKSP	65,91
X-PTR	91

For I = debut To fin : Print i ; " " ;  
 Peek i ; " " ; CHR\$ Peek i AND  
 Peek i > 32 : Next I

poke 40000, 219  
 10000, 1, 254  
 2, 211  
 3, 254  
 4, 24  
 5, 250

Achévé d'imprimer en novembre 1983  
 sur les presses de l'imprimerie Laballery et C<sup>o</sup>  
 58500 Clamecy  
 Dépôt légal : novembre 1983

N° d'impression : 310010  
 N° d'édition : 86595-117-1  
 ISBN : 2-86595-117-0

m

Print Peek 23732 + 256 \* Peek 23733

65 535  
 - 23 732  
 -----  
 8 03

2048  
 8  
 -----  
 2056  
 Rand UBR 2056

836  
 768  
 -----  
 68  
 08 55

768  
 65  
 -----  
 832  
 Range 0 to 1

Poke (1 + Peek 23635 + 256 \*  
 Peek 23636), 1

*[Handwritten scribbles in green and red ink]*

# mémento

Comment gagner du temps?... Sans en perdre à glaner de-ci de-là tous les renseignements techniques dont vous avez besoin pour bien utiliser votre SINCLAIR. Les "Clefs" c'est : ● La liste des instructions Basic commentées ● Les mnémoniques de l'assembleur Z80 et leurs codes objets ● Les points d'entrées de la ROM Basic ● Des explications sur les variables système, et... ● Une liste d'astuces pour mieux utiliser l'écran, les cassettes et les programmes en langages machine.

## CLEFS POUR LE ZX SPECTRUM

Editions du P.S.I.  
B.P. 86  
77402 Lagny/Marne  
France

ISBN 2.86595.117.0

