

Le SHARP MZ-80K

Description.

Ecran monochrome 40 caractères x 25 lignes

RAM 48 Ko

Clavier 78 touches réparties en 3 groupes :

- Grises : caractères alphanumériques
- Jaunes : touches de contrôle
- Bleues : caractères graphiques

Lecteur-enregistreur de cassettes

Haut-parleur : musique programmable 3 octaves, 7 tempi

Langages : Assembleur Z80, Basic, Forth

L'écran possède une deuxième définition : 80 x 50 affichant uniquement des points 2 x 2 pixels. Les deux modes écran peuvent être utilisés en même temps.

Le clavier virtuel.



Un appui sur la touche suivi de la saisie de 3 chiffres sur le clavier virtuel permet d'obtenir des caractères graphiques ne figurant pas au clavier. L'effet Shift est obtenu par le bouton droit de la souris.



CLR efface l'écran et réinitialise les variables programme. HOME renvoie le curseur dans le coin supérieur gauche.



INST insère un caractère à la position du curseur. DEL supprime le caractère sous le curseur.



Affiche un espace.



Déplacement vertical du curseur.



Déplacement horizontal du curseur



Quitte le programme Basic en cours d'exécution.



Entrée (Carriage Return)



Un appui sur cette touche permet d'accéder au troisième caractère de certaines touches (en bas à droite). Cela donne également accès aux minuscules pour les caractères alphabétiques.

Le voyant à droite de l'écran passe au rouge quand la touche est activée. Pour désactiver la fonction, un simple clic sur la touche fait l'affaire.

Autres touches.



Le caractère du haut est obtenu en cliquant à l'aide du bouton droit.
Pour les minuscules, activer la touche SLM/CAP.



Le caractère en bas à droite est obtenu en activant la touche SML/CAP
comme pour les minuscules.



Les caractères graphiques permettent, par assemblage, de réaliser
des cadres, des dessins...

Le clavier virtuel s'utilise donc avec la souris. On également se servir du clavier de l'ordinateur, mais cet usage est limité aux lettres, aux chiffres, flèches de direction, touches Home, Inser, Suppr et Entrée.

Le lecteur de cassettes.

Bien entendu, il est virtuel. Il permet d'enregistrer et de charger rapidement des programmes. Le nom de ces fichiers est standard : "**PGM**" suivi d'un numéro d'ordre attribué automatiquement.

La commande **NEW** (voir plus loin) génère automatiquement un nouveau nom que l'on pourra utiliser...ou pas.

Le Moniteur.

Gère l'environnement du programme et comporte les commandes suivantes :

LOAD "name" : Permet de charger un programme en mémoire.

SAVE "name" : Sauvegarde d'un programme sur disque.

CLR : Réinitialise les variables en mémoire.

RUN : Lance l'exécution d'un programme.

LIST : Affiche le programme en cours

LIST n : Affiche le programme à partir de la ligne **n**

Pour faire défiler la liste, amener le curseur première ou dernière ligne,
un appui supplémentaire de la touche de direction, reculera ou avancera
le listing d'une demi-page.

NEW : Efface le programme en mémoire ainsi que ses variables.

BYE : Permet de quitter le programme MZ80.

Le Basic.

C'est un micro-basic fortement simplifié du Basic standard fourni avec l'engin.

Les numéros de lignes comportent trois chiffres, de 001 à 999.

Les différents éléments d'une instruction sont séparés par un espace, parfois une virgule.

Plusieurs instructions par ligne séparées par  (^ du clavier ordi).

Certaines instructions tel que affectation de variables, affichages, peuvent être exécutées en mode direct. Il suffit de ne pas mettre de numéro de ligne.
Pratique pour faire des essais.

Tous le termes (instructions, variables, ...) doivent être écrits en majuscules. Seuls les libellés (entre double quote) peuvent être en minuscules ainsi que le commentaires.

Variables numériques.

Le nom de ces variables commence obligatoirement par une lettre et peut comporter des chiffres.

L'affectation à ces variables peut être une variable, un littéral ou une expression.

Les expressions comportant les signes +, -, * ou / sont écrites sans intervalles.

Par exemple : **A = B+12**

A sera créé, s'il n'existe pas, mais B doit exister.

Par souci de simplification, ces variables sont de type integer.

Variables alphanumériques.

Le nom des variables est constitué comme les numériques avec en plus le signe \$ en terminaison.

En affectation, seule l'addition est autorisée (concaténation).

Exemple : **A\$ = A\$+" perdu!"**

Les instructions.

GOTO : 100 GOTO 200 Poursuit le traitement à la ligne 200

GOSUB : 100 GOSUB 700 Branchement à une routine en ligne 700

RETURN : Termine un routine et reprend le traitement immédiatement après le GOSUB correspondant.

Pas de GOSUB imbriqués ni de sortie par GOTO

ON ... GOTO : 20 ON A GOTO 50,60,80 En fonction des valeurs 1,2,ou 3 que peut prendre A, le saut se fera en ligne 50,60,ou 80.

ON...GOSUB: Comme ON...GOTO mais branchement à des routines.

IF ... THEN : Teste une condition et en fonction du résultat
IF A = 20 THEN 100 Saut à la ligne 100
IF B < 3 THEN B = B+3
IF C\$ = "" THEN C\$ = "vide"

IF ... PRINT Affichage conditionnel

IF ... GOTO Saut conditionnel

IF ... GOSUB Exécution conditionnelle d'une routine

FOR ... NEXT **FOR A = 1 TO 9 STEP 2** Boucle classique
PRINT A STEP est facultatif et peut
NEXT A avoir une valeur négative.

Il est permis d'imbriquer des boucles FOR...NEXT

DATA : **DATA 12,-3,100** Liste de données numériques

READ : **READ A,B** Lecture des DATA : A = 12 et B = -3

READ C C prendra la valeur 100

RESTORE : Repositionne le READ en début des DATA
 Plusieurs lignes de DATA peuvent être programmées, elles
 seront lues dans l'ordre consécutivement.

SET X,Y : Cette instruction utilise l'écran en mode 80x50 et trace un point
 de 2x2 pixels dans un carré 4x4..

RESET X,Y : Permet d'effacer un point tracé par SET.

WOPEN : Ouvre un fichier de données en écriture.
 Ex : **WOPEN "Oseille.txt"**

ROPEN : Ouvre un fichier de données en lecture.
 Ex : **ROPEN "Oseille.txt"**

CLOSE : Ferme un fichier ouvert avec WOPEN ou ROPEN.

INPUT : Permet la saisie de données. On termine par Entrée.
 Ex : **INPUT A**
 INPUT A\$
 INPUT "Montant :" A

INPUT/T : Lecture d'un enregistrement après un ROPEN.
 EX : **INPUT/T A\$**

GET : Saisie d'un caractère numérique ou alphanumérique.
 Le programme continue dès que le caractère est frappé.

PRINT Affichage de données à l'écran
 Ex : **PRINT A**
 PRINT A\$
 PRINT "Montant :" A

PRINT/T : Ecriture dans un fichier ouvert par WOPEN
 Ex : **PRINT/T A\$**

TI\$: Donne l'heure sous la forme "hh:mm:ss"
 Ex : **PRINT TI\$**
 A\$ = TI\$

TAB(X) : Déplace le curseur de X caractères vers la droite.
 Ex : **PRINT TAB(5) A\$**

SPC(X) : Affiche X caractères espace.
 Ex : **PRINT SPC(5) A\$**

END : Fin du programme

REM : Commentaire

PEEK : Permet de lire un code caractère dans la mémoire écran.
Ex : **A = PEEK(adr)**
adr = 40 * ligne + colonne (de 0 à 999).
Pour mémoire : Ligne et colonne débutent à zéro.

POKE : Ecrit un caractère en mémoire écran
Ex : **POKE adr,C**
adr : comme pour PEEK, C : code caractère
Utilisation particulière : Positionnement du curseur
POKE adr,0

TRAITEMENT DE CHAINES DE CARACTERES

LEFT\$: **A\$ = LEFT\$(X\$,N)** Extrait les N caractères de gauche de la chaîne X\$ et les place dans A\$.

RIGHT\$: **A\$ = RIGHT\$(X\$,N)** Extrait les N caractères de droite de la chaîne X\$.

MID\$: **A\$ = MID\$(X\$,P,N)** Extrait N caractères à partir de la position P de la chaîne X\$.

LEN : **LX = LEN(X\$)** Renvoie la longueur de la chaîne X\$

ASC : **A = ASC(X\$)** Renvoie le code ASCII du premier caractère de X\$.

CHR\$: **X1\$ = CHR\$(A)** Convertit le code ASCII A en caractère.

VAL : **K = VAL(N\$)** Convertit la chaîne N\$ en valeur numérique

STR\$: **N\$ = STR\$(K)** Convertit le nombre K en chaîne de caractères.

FONCTIONS.

SQR : **A = SQR(N)** Calcule la racine carrée de N.

RND : **A = RND(N)** Génère un nombre entre 0 et N -1.

OPERATEURS ARITHMETIQUES : =, +, -, *, /

OPERATEURS LOGIQUES : =, <, >, <=, >=, <>

AUTRES SYMBOLES.

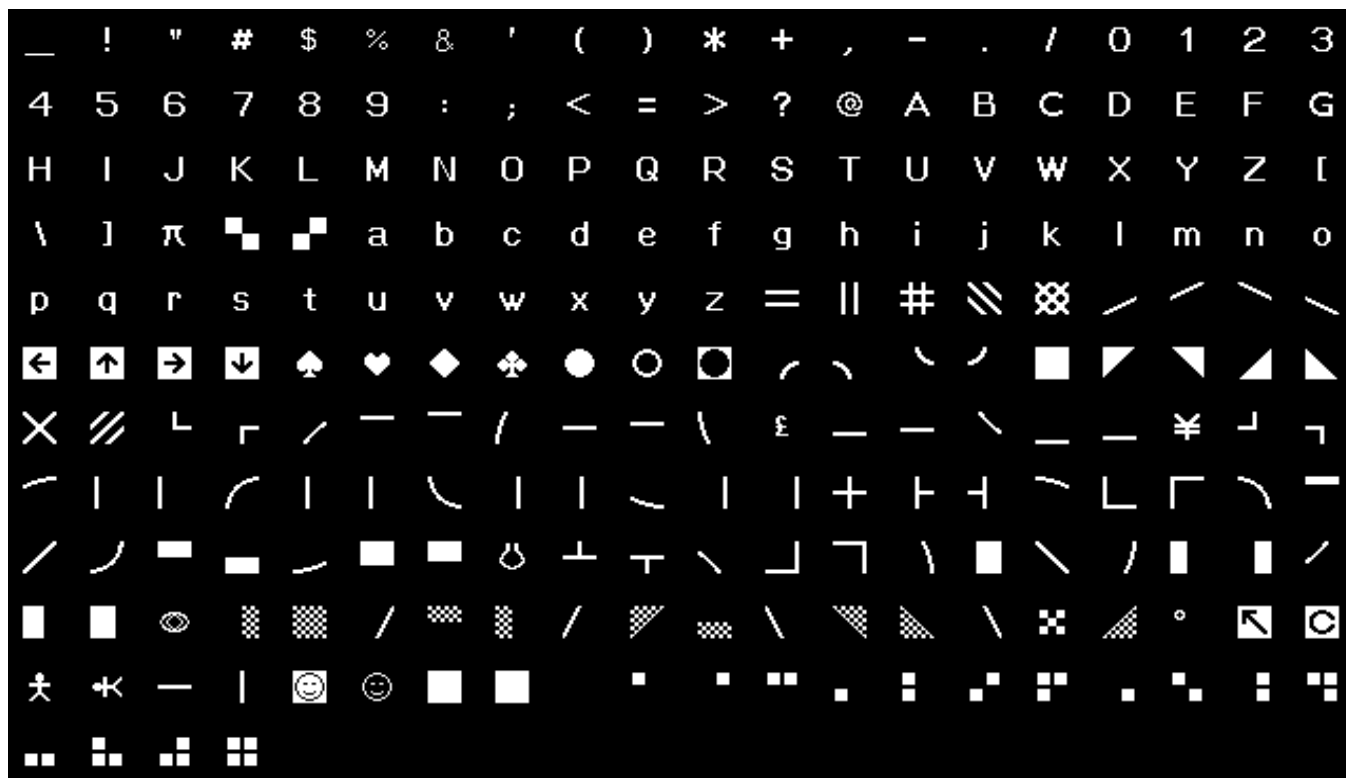
? Equivalence de l'instruction PRINT

; A la fin d'une instruction PRINT, annule le saut à la ligne.



" Délimite un libellé alphanumérique.

\$ Indique une variable chaîne.

LE JEU DE CARACTERES



Ces caractères dans un libellé de l'instruction PRINT permettent de déplacer le curseur.

De même,  positionne le curseur dans le coin supérieur gauche et  efface l'écran. Ces deux derniers ne figurent pas sur le clavier et ont respectivement les codes 230 et 231.

PROGRAMMES EXEMPLES.

- PGM1 :** Table des caractères (de code 32 à 255).
- PGM2 :** Test INPUT.
- PGM3 :** Page d'accueil d'une de mes réalisations de l'époque.
- PGM4 :** Démo SET / RESET
- PGM5 :** Création d'un fichier de grands chiffres.
- PGM6 :** Affichage des chiffres.
- PGM7 :** Horloge.
- PGM8 :** Jeu TAKE-DOWN fourni avec la machine.
- PGM9 :** Jeu des Allumettes.

PARTICULARITES ET DIFFICULTES.

Pour stocker et faciliter la recherche des commandes, instructions et symboles du Basic, je me suis souvenu de l'astuce de Caribensila : "Utiliser Case of avec des chaînes" c'est-à-dire utiliser des TStringList.

Idem pour les variables avec, associé à la StringList, une table d'integer pour les numériques et une table de string pour les alphanumériques.

Idem pour le texte du programme basic, stocké dans une ListBox (pour faciliter la mise au point), en association avec une StringList contenant les numéros de ligne nécessaires pour les branchements.

Pour les DIM() je n'ai rien trouvé de mieux que de stocké la table entière dans une chaîne, à condition que tous les éléments soit de même longueur.

Ce que j'ai fait dans le programme PGM5 où je crée de grands chiffres dans une chaîne qui est l'équivalent de "array [0..9,0..6] of string[5]"

Autre difficulté : La saisie de caractères par GET et INPUT... Là je stocke l'adresse de l'instruction suivante, je stoppe le programme. Comme l'écran est le même pour le programme et pour le moniteur, je fait ma saisie et je relance le programme au numéro d'instruction stockée.

Plus dur : Comment arrêter un programme qui tourne dans une boucle volontaire, alors que le clavier ne répond pas pendant l'exécution.

J'ai utilisé l'instruction STOP qui démarre un timer, arrête le programme à chaque boucle pendant 5 secondes puis continue. Ces quelques secondes sont suffisantes appuyer sur Echap ou cliquer sur Break. Voir PGM7.

Pour les calculs, j'ai fait appel à notre ami Barbichette et à son excellent programme "Evaluation-Expression-mathématiques" dont j'ai repris sans vegogne trois unités.

J'ai en outre ajouté trois instructions ne figurant pas dans le basic d'origine :

WAIT : nos bécanes vont trop vite.

TRON : Affiche le texte de l'instruction en cours dans le panel en bas du clavier.

TROF : Annule la précédente.