ATARI BASIC

REFERENCE GUIDE

LEITFADEN

MANUAL DE REFERENCIA

MANUALE D'USO

MANUEL DE REFERENCE

HANDLEIDING

ATARI®

MANUEL DE REFERENCE

POUR PROGRAMMEURS EXPERIMENTES

Apprendre le BASIC, c'est un peu comme apprendre une langue étrangère: il faut y consacrer un peu de temps et quelques efforts, mais la récompense en vaut la peine. Ce manuel vous donnera un certain nombre de renseignements sur le BASIC ATARI — un langage simple et puissant —destiné à ceux qui désirent programmer en BASIC.

Ce manuel est un guide de référence. Il ne présente pas une série complète d'exemples de programmation et ne constitue pas un manuel d'instruction complet à l'usage du débutant. Nous conseillons aux lecteurs non familiarisés avec le langage BASIC de se procurer un manuel destiné aux débutants; il en existe beaucoup d'excellents sur le marché.

INDEX

CONTRÔLES	PAGE NUMERO
ABS	49
ADR	50
AND	44
ASC	50
ATN	50
BYE	45
CLOAD	45
CHAR$	50
CLOG	49
CLOSE	47
CLR	48
COLOR	48
COM	48
CONT	45
COS	50
CSAVE	45
DATA	48
DEG	50
DIM	48
DOS	45
DRAWTO	49
MISE EN FORME	51
END	46
ENTER	45
CODE DES ERREURS	51
EXP	49
FOR	46
FRE	50
GET	47
GOSUB	46
GOTO	46
GRAPHIQUES/GRAPHICS	48
IF	46
INPUT	47
INT	49
LEN	50
LET	48
LIST	45
LOAD	45
LOCATE	49
LOG	49
LPRINT	47
NEW	45
NEXT	46
NOT	44
NOTE	47
ON	46
OPEN	47
OPERATEURS	44
OR	44
PADDLE	50
PEEK	50
PLOT	49
POINT	47
POKE	48
POP	46
POSITION	49
PRINT	47
PTRIG	50
PUT	47
RAD	50
READ	48
REM	48
RESTORE	48
RETURN	46
RND	49
RUN	45
SAVE	45
SETCOLOR	48
SGN	49
SIN	50
SOUND	45
SQR	49
STATUS	47
STICK	50
STRIG	50
STOP	46
STR$	50
THEN	46
TOUCHES DE FONCTIONS SPECIALES	51
TO	46
TRAP	46
USR	50
VAL	50

MANUEL DE REFERENCE

ORDRE DE PRIORITE DES OPERATEURS

Les opérations qui sont insérées dans les parenthèses les plus à l'intérieur doivent être exécutées en premier. Lorsqu' un jeu de parenthèses est compris à l'intérieur d'un autre jeu de parenthèses, on parle de parenthèses "imbriquées". Les opérations qui sont au même niveau s'exécutent dans l'ordre suivant:

DE LA PRIORITE LA PLUS ELEVEE A LA PRIORITE LA PLUS BASSE

<, >, =, <=, >=, <>
Opérateurs relationnels utilisés dans les expressions chaîne de caractères. Ils ont le même ordre de priorité et sont exécutés de gauche à droite.

-
Signe moins unaire, z. B.: x=-1

Elévation à une puissance

***, /**
Multiplication et division ont le même ordre de priorité et s'exécutent de gauche à droite.

+, -
Addition et soustraction ont le même ordre de priorité et s'exécutent de gauche à droite.

<, >, =, <=, >=, <>
Opérateurs relationnels utilisés dans les expressions numériques

NOT
NON logique

AND
AND (ET) logique

OR
OR (OU) logique

(Abréviations permises entre parenthèses)
Les mots qui suivent ne peuvent pas être utilisés comme noms de variables. La majorité de ces instructions peut être tapée directement au clavier ou être utilisée à l'intérieur d'un programme.

COMMANDE DU SYSTEME

BYE (B.) — Sort du mode BASIC pour passer en mode SELF TEST.

DOS (DO.) — Affiche le menu DOS (à utiliser avec une unité de disquette seulement)

CSAVE (CS.) — Sauvegarde le programme sur cassette.

CLOAD (CLOA.) — Charge le programme dans la mémoire de l'ordinateur à partir d'une cassette.

SAVE (S.) — Sauvegarde un programme en BASIC, sur une disquette ou tout autre périphérique.
Ex.: SAVE "D:FICHIER.BAS"

LOAD (LO.) — Charge un programme dans la mémoire de l'ordinateur à partir d'un périphérique d'entrée.
Ex.: LOAD "D:FICHIER.BAS"

LIST (L.) — Liste un programme à l'écran ou sur un périphérique.
Ex.: LIST (liste la totalité du programme).
LIST 10 (liste la ligne 10 à l'écran)
LIST 10,20 (liste tout ce qui se trouve de la ligne 10 à la ligne 20 incluse)
LIST "P:" (liste à l'imprimante)
LIST "P:", 10, 20 (les lignes 10 à 20 sont listées sur l'imprimante)
LIST "D FICHIER" (liste à un fichier sur disquette)
LIST "D: FICHIER.LST" 10,20 (liste les lignes 10 à 20 dans un fichier sur disquette).
LIST "C:" (liste sur cassette).

ENTER (E.) — Entre le programme dans l'ordinateur.
Ex.: ENTER "C:"
ENTER "D: FICHIER.LST"
Note: Les lignes de même numéro d'un programme déjà chargé seront remplacées.

NEW — Efface tout programme se trouvant dans la mémoire de l'ordinateur.

RUN (RU.) — Lance l'exécution d'un programme en BASIC. Le programme peut se trouver en mémoire ou bien être chargé à partir d'une disquette ou d'une cassette. Initialise les variables en les remettant à zéro et supprime les tableaux et les chaînes de caractères.
Ex.: RUN (exécute le programme en mémoire).
RUN "D: FICHIER.BAS" (CHARGE le programme à partir de la disquette et l'exécute).

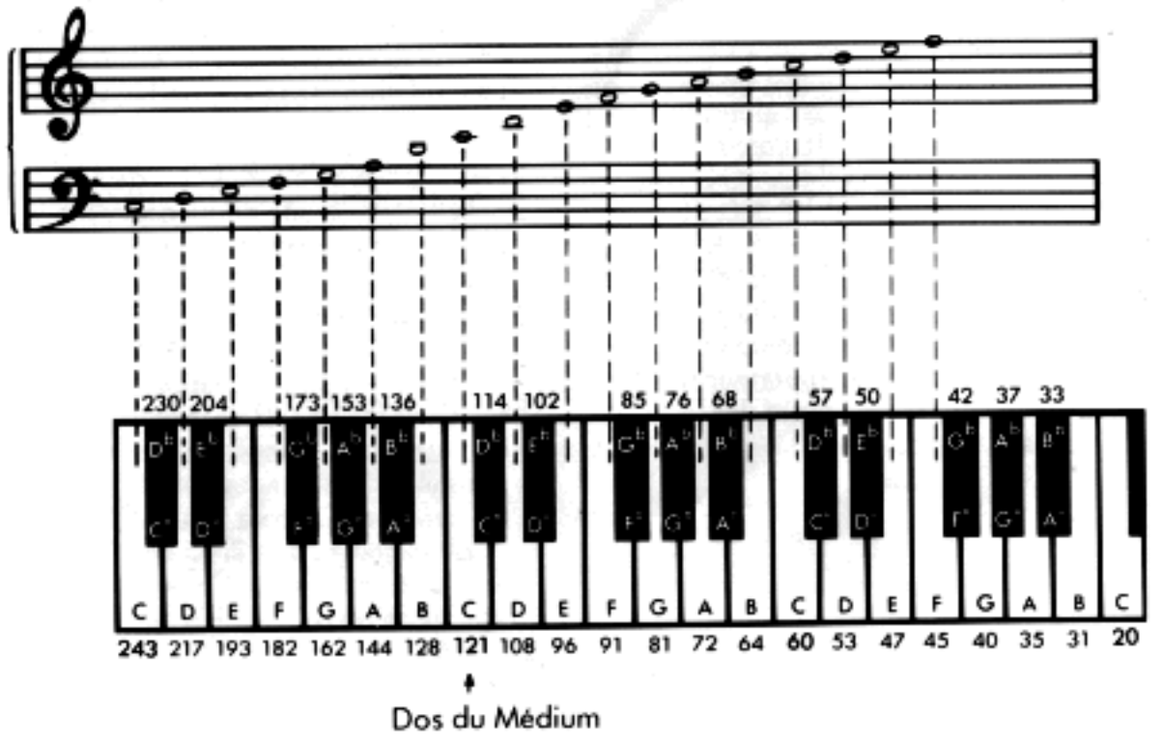
CONT — Reprend l'exécution du programme après que la touche BREAK ait été pressée ou que le programme ait exécuté une instruction STOP ou END. S'il existe d'autres instructions de programme sur la même ligne, celles-ci ne sont pas exécutées. L'exécution du programme reprend à la ligne numérotée suivante.

INSTRUCTIONS SONORES

SOUND (SO.) — Provoque l'émission d'un son. L'ordinateur ATARI peut jouer jusqu'à quatre sons simultanément. On parle alors de voix. Cela correspond à l'intérieur de l'ordinateur à quatre synthétiseurs indépendants. Le son dure jusqu'à ce qu'une autre instruction SOUND soit adressée à la même voix, ou qu'une instruction END, RUN ou NEW soit exécutée. Les voix sont programmées indépendamment et peuvent toutes fonctionner simultanément. Cette instruction doit être suivie par quatre valeurs (nombres, variables ou expressions).

Ex.: SOUND A, B, C, D où:
A = Numéro de canal (0 à 3)
B = Période (0 à 255). Plus la période est grande, plus la fréquence est basse. Fréquence = 31960/PERIODE + 1). Voir le tableau pour les équivalences musicales.
C = Distorsion (0 à 14, valeurs paires seulement). 10 et 14 sont des tons "purs". Les autres nombres donnent d'autres sonorités.
D = Volume (0 à 15). Plus le nombre est grand, plus le volume sonore est fort. Le chiffre 0 ne produit aucun son. Il se peut que le son soit très fortement distordu si le volume total des quatre voix dépasse 32.

CORRESPONDANCE ENTRE LE CLAVIER DU PIAN ET LA GAMME MUSICALE



COMMANDES DES PROGRAMMES

GOTO (G.) — L'exécution du programme se fait à partir du numéro de ligne qui a été spécifié.
Ex.: GOTO 30 (exécute le programme à partir de la ligne 30).
GOTO A + 10 (autorisé, mais peut être difficile à déterminer).

ON. . .GOTO — L'exécution du programme passe à l'un des numéros de ligne spécifié après le mot GOTO. Le choix s'effectue selon la valeur de l'expression suivant ON.
Ex.: ON A GOTO 10, 300, 50 (IF A = 1 THEN GOTO 10; IF A = 2 THEN GOTO 300; IF A = 3 THEN GOTO 50).

GOSUB (GOS.) — L'exécution du programme continue à la ligne spécifiée. Une instruction RETURN fait revenir le programme à l'instruction qui suit ON . . . GOSUB.
Ex.: GOSUB 30 (executer sous-programme à la ligna 30)
GOSUB A + 10 (légal, mais peut s'avérer tres difficile à mettre au point).

ON . . . GOSUB — L'exécution du programme continue au numéro de ligne indiqué par une expression. Une pression de RETURN fera revenir la commande à l'instruction qui suit le ON . . . GOSUB
Ex.: ON A + 1 GOSUB 10, 300, 50
(IF A + 1 = 1 THEN GOSUB 10;
IF A + 1 = 2, THEN GOSUB 300;
I F A + 1 = 3, THEN GOSUB 50).
Attention: si l'expression évalue un nombre inférieur à 1 ou supérieur au numéro de ligne le plus élevé, les résultats sont imprévisibles.

RETURN (RET.) — Termine un sous-programme et ramène le programme à l'instruction qui suit immédiatement la dernière instruction GOSUB exécutée.
Ex.: RETURN

FOR (F.) — Cette instruction fixe les valeurs de départ et de fin d'une variable d'index, ainsi que la valeur qui doit lui être ajoutée chaque fois qu'une boucle FOR. . .NEXT s'exécute. La valeur ajoutée est 1 à moins qu'il n'en soit spécifié autrement par une instruction STEP. Une instruction NEXT fera se rèpèter l'instruction entre FOR et NEXT.
Ex.: FOR A = 1 TO 10 (A commencera à 1 et augmentera chaque fois de 1 pour s'arrêter à 10).
FOR A = 10 TO 1 STEP -2 (la valeur du pas vaut -2).
FOR A = B/T TO B*T STEP X (les valeurs calculées sont également autorisées).

NEXT (N.) — Termine une boucle FOR . . . NEXT. Vérifie que la valeur d'index n'a pas dépassé la valeur finale, incrémente la valeur d'index de la valeur STEP et reprend l'exécution après FOR. Si l'index dépasse la valeur finale, le programme passe à l'instruction suivant NEXT.
Ex.: NEXT A (A est la variable d'index)

POP — Enlève l'information de retour stockée dans la pile et qui concerne la dernière instruction FOR ou GOSUB exécutée. Pratique pour sortir prématurément d'une boucle FOR—NEXT, ou pour sortir d'un sous-programme sans exécuter RETURN.
Ex.: POP: GOTO 10

IF. . .THEN — L'instruction qui suit THEN est exécutée lorsque la condition qui se trouve entre IF et THEN est vraie. Sinon, passe à la ligne suivante du programme.
Ex.: IF A ≠ C THEN GOTO 300
IF A + B THEN PRINT "A = B": PRINT "FICHTRE!": LET A = 5: GOTO 20
IF A THEN PRINT "A est différent de 0" (une expression différente de 0 es vraie)

TRAP (T.) — Lors d'une erreur, TRAP déroute l'exécution à la ligne spécifiée. TRAP reste valide jusqu'à ce qu'une erreur se présente ou bien jusqu'à la prochaine instruction TRAP.
PEEK (195) renvoie le numéro d'erreur.
Ex.: TRAP 30 (en cas d'erreur, aller à la ligne 30)
TRAP 40000 (les numéros de ligne plus élevés que 32767 mettent TRAP hors fonction).

STOP (STO.) — Arrête le programme. Imprime le numéro de ligne. Ne ferme pas les fichiers ni n'arrête le son. CONT. permet de relancer un programme.
Ex.: IF A=B THEN STOP

END — Arrête le programme, ferme tous les fichiers ouverts, arrête le son. Le programme peut être relancé avec CONT.

ENTREE/SORTIE

DESIGNATION DES PERIPHERIQUES

Chaque appareil de la gamme ATARI a une désignation qui lui est propre. L'unité de disquette A 810 et les ports série du module d'interface A 850™ (interface RS 232) doivent être numérotés de 1 à 4. Les unités de disquette doivent aussi recevoir un nom de fichier. Ces noms dovient être placés dans des chaînes de caractères ou dans des variables chaînes de caractères. En voici quelques exemples:

K:	Clavier. Entrée seulement.
P:	Imprimante. Sortie seulement.
C:	Cassette. Entrée et Sortie.
S:	Ecran (TV). Sortie seulement.
E:	Edition à l'écran (clavier et écran combinés). Entrée et Sortie.
R:	Interface RS 232. Port série du module d'interface A 850. Entrée et Sortie.
D:NOM.EXT	Exécute le fichier sur l'unité de disquette n° 1. NOM.EXT placé dans la disquette n° 1.
D2:NOM.EXT	Même fichier NOM.EXT sur l'unité de disquette ° 2.

MANUEL DE REFERENCE

INSTRUCTIONS D'ENTREES/SORTIES

Les noms de fichiers mis sur disquette commencent par une lettre de l'alphabet et peuvent avoir une longueur allant jusqu'à 8 caractères. La désignation du fichier peut se terminer par une extension optionnelle (un point suivi de 1 à 3 caractères). Vous pouvez utiliser n'importe quelle combinaison de 3 lettres ou chiffres à votre gré. Voici quelques extensions utiles possibles:
.BAS = Sauvegarde des programmes BASIC (identique à SAVE).
.LST = Listage des programmes BASIC (identique à LIST).
.DAT = Fichier de données
.OBJ =Fichier en langage machine (code objet)
.TXT =Fichier de texte

OPEN (O.) — Ouvre un canal et lui associe un buffer en vue d'une opération d'entrée /sortie. Les valeurs possibles de canaux vont de 1 à 7. Utilisez les codes suivants:

TYPE	CODE	OPERATION
Lecture	4	Lecture seulement
Ecriture	8	Ecriture seulement
Ecriture & Lecture	12	Lecture et écriture
Ajout	9	Ajout en fin de fichier
Catalogue	6	Catalogue de la disquette considérée

Ex.: OPEN # 1, 4, 0, "K:" (Ouverture du canal n° 1 pour le clavier (lecture))
OPEN #2, 8, 0, "P:" (Ouverture du canal n° 2 pour l'imprimante (écriture))
OPEN # 1, 12, 0, "D:FICHIER.DAT" (Ouverture du canal n° 1 pour le fichier FICHIER DAT placé sur disquette (écriture et lecture))
OPEN # 1, 6, 0, "D:*.*" (Ouverture du canal n° 1 pour lire le catalogue de la disquette placée dans l'unité 1)

CLOSE (CL.) — Ferme un canal et après l'opération d'entrée ou de sortie et desalloue le IOCB. Peut être exécutée lorsqu'aucun canal n'a été ouvert. Les numéros IECO sont 1-7, comme dans OPEN.
Ex.: CLOSE# (Ferme le canal ouvert sur IOCB No. 1 et désalloue le IOCB.)

INPUT (I.) — Reçoit une ligne de caractères par un canal. La chaîne de caractères doit être terminée par un caractère "fin de ligne" (EOL).
Ex.: INPUT A (Entrée d'une valeur numérique (sous-entendue au clavier)).
INPUT A, B, C, (Entrée de trois valeurs numériques séparées par des virgules et placement dans les trois variables A, B, C)
INPUT A $ (Entrée d'une ligne de caractères. La placer dans la variable chaîne A $; A $ ne contient pas de caractère RETURN)
INPUT # 1; A $ (Entrée d'une ligne de caractères par le canal n° 1 et mise en place de ces caractères dans la nouvelle chaîne A$ and B)

PRINT (PR.) ou **(?)** — Envoie des données sur l'écran ou sur autre équipment.
Ex.: PRINT (envoie une ligne vierge).
PRINT "Le numéro est" ; A (Imprime texte et numéro à l'écran.)
PRINT "Le numéro est", A (La virgule entraín l'impression de A dans une colonne séparée. POKE 201 avec largeur de colonne désirée.)
PRINT A $; (Le point-virgule empêche le RETURN D'être envoyé à la fin de la line.)
PRINT #1 ; A $ (Envoie A$ aux périphériques ouverts sur IOCB No.1.)

LPRINT (LP.) — Envoie des messages ou des résultats vers l'imprimante.
L'ouverture et la fermeture d'un canal ne sont pas nécessaires. Le point-virgule placé à la fin de l'instruction provoque toutefois un saut de ligne.
Ex.: LPRINT Provoque un saut de ligne.
LPRINT A$ (A$ seront imprimés)
LPRINT A$;B(A$ et B seront sur la même ligne.)
LPRINT A$,B (La virgule entraín l'impression de B dans une colonne séparée. POKE 201 avec largeur de désirée.)

GET — Attend un octet en provenance du périphérique spécifié par le numéro de canal et le place dans la variable indiquée.
Ex.: GET # 1,A (Attend un octet du périphérique relié au canal n° 1 et le place dans la variable A).

PUT — Envoie un octet à destination du périphérique spécifié.
Ex.: PUT 1, A (Insére l'octet qui est dans A, au cnal ouvert sur IOCB 1.)

NOTE (NO.) — Utilisé avec l'unité de disquette pour déterminer la position de l'octet suivant à lire ou à écrire dans l'ensemble du buffer.
Ex.: NOTE 1, SEC, BYTE (Place le numéro de secteur dans SEC et le numéro de l'octet à l'intérieur du secteur dans BYTE. Se réfère au fichier ouvert sur IOCB 1.)

POINT (P.) — Utilisé pour informer DOS de l'endroit où le prochain octet doit être pris et lu ou doit être adressé.
Ex.: POINT # 1, SEC, BYTE Positionne le pointeur sur le secteur numéro SEC, avec un offset d'une valeur BYTE.

STATUS (ST.) — Demande l'état du périphérique choisi.
Ex.: STATUS #1, A (Place l'état du périphérique relié au canal numéro 1 dans la variable A).

INSTRUCTIONS DE TRAITEMENT

LET (LE.) — Assigne des valeurs aux variables numériques ou de chaîne.
Ex.: LET A = B (valeur de B assignée à A)
LET A $ = "BONJOUR"
A = B: A$= "BONJOUR" (LET peut être omis)

POKE — Place un nombre entre 0 et 255 dans une case mémoire numérotée entre 0 et 65535. Les valeurs décimales sont arrondies. PEEK permet de lire ce qui se trouve en mémoire.
Ex.: POKE 82, 0 (Met 0 dans la case mémoire 82)
A = PEEK (82) (Lit le contenu de la case mémoire 82 et place le résultat dans la variable A)

DIM — Réserve une zone mémoire pour une chaîne de caractères ou des tableaux numériques. Chaque caractère occupe un octet et chaque élément d'un tableau numérique occupe six octets.
Ex.: DIM A$ (10) (On réserve la place pour une chaîne de caractères comprenant au minimum 10 caractères)
DIM B (10) (On crée un tableau numérique de 10 éléments à une dimension)
DIM B (10, 10) (On crée un tableau numérique de 100 valeurs à deux dimensions)
DIM A$ (10), B(10) (Séparez les différentes variables par des virgules)

COM — Voir DIM.

CLR — Efface les dimensions de n'importe quel tableau, vide les variables-chaînes, et réduit les variables numériques à zéro (inverse de DIM).
Ex.: CLR

DATA (D.) — Présente une liste de valeurs numériques et/ou de caractères alphanumériques qui seront lus séquentiellement par l'instruction READ.
Ex.: DATA 12, 27, BONJOUR, ATARI 600

READ (REA.) — Lit séquentiellement les données placées dans les lignes de DATA et les assigne à des variables. Il doit y avoir correspondance entre le type de la variable et le type de DATA (numérique ou alphanumérique).
Ex.: READ A (A sera le numéro suivant sun la note dans les instructions du DATA)
READ A$ (Sert également pour les chaînes de caractères).
READ A, A$, B, BS (Sépares les éléments mulitples par des virgules).

RESTORE — Positionne le pointeur utilisé par l'instruction READ sur une ligne de DATA particulière.
Ex.: RESTORE (L'oclet suivant sera le prémières de DATA).
RESTORE (10) (L'oclet suivant sera le prémier élément des instructions prémières de DATA à la ligne 10).

REM (R.) ou ([ESPACE].) — Permet l'introduction de commentaires dans le corps du programme. Tous les caractères ou instructions placés sur une ligne après une instruction REM sont ignorés par le programme.

GRAPHIQUES

GRAPHICS (GR.) — Sélectionne le mode graphique. Le fait d'ajouter 16 au numéro du mode graphique sélectionne le plein écran. En ajoutant 32, on supprime l'effacement automatique de l'écran.
Ex.: GRAPHIQUES 8 (Graphiques Made 8 avec écran partage)
GRAPHIQUE 8 8+16 (Modes, plein écran)
GRAPHIQUES 8+32 (N'effacera pas l'écran)
GRAPHIQUES 8+16+32 (Une combinaison des deux options)

SETCOLOR (SE.) — Définit la teinte et la luminance pour un registre couleur particulier.
Ex.: SETCOLOR 1, 2, 4 (Attribue au registre couleur 1, 1a couleur 2 et la luminance 4. Les registres vont de 0 à 3; les teintes de 0 à 15; les luminances de 0 à 14, numéros pairs seulement).
(L'exception est modes GTIA, qui peuvent utiliser des nombres pairs et impairs jusqu'à 15.)

COLOR (C.) — Dans les modes graphiques 3 à 11, cette instruction sélectionne le registre couleur qui sera utilisé dans l'instruction PLOT.
Ex.: COLOR 2 (Sélectionne le deuxième registre couleur dans les modes 0-2, sélectionne la caractère ASCII, ayant une valeur de deux pour PLOT).

FORMATS D'ECRAN ET MODES GRAPHIQUES

Mode Graphique	Type	Nombre de Colonnes	Nombre de Lignes* Ecran Partagé	Nombre de Lignes Plein Ecran	Nombre de Couleurs	Espace Mémoire Nécessaire Ecran Partagé	Espace Mémoire Nécessaire Plein Ecran
0	TEXTE	40	—	24	1,5	—	992
1	TEXTE	20	20	24	5	674	672
2	TEXTE	20	10	12	5	424	420
3	GRAPHIQUE	40	20	24	4	434	432
4	GRAPHIQUE	80	40	48	2	694	696
5	GRAPHIQUE	80	40	48	4	1174	1176
6	GRAPHIQUE	160	80	96	2	2174	2184
7	GRAPHIQUE	160	80	96	4	4190	4200
8	GRAPHIQUE	320	160	192	1-1/2	8112	8138
9	GRAPHIQUE	80	—	192	1	—	8138
10	GRAPHIQUE	80	—	192	9	—	8138
11	GRAPHIQUE	80	—	192	16	—	8138
12	GRAPHIQUE	40	20	24	5	1154	1152
13	GRAPHIQUE	40	10	12	5	664	660
14	GRAPHIQUE	160	160	192	2	4270	4296
15	GRAPHIQUE	160	160	192	4	8112	8138

PLOT (PL.) — Affiche un point ou un caractère à l'écran aux coordonnées X-Y spécifiées.
Ex.: PLOT 3, 5 (troisième colonne, cinquième ligne)

POSITION (POS.) — Positionne le curseur sur l'écran à un emplacement de coordonnées X et Y. Mais ne trace rien.
Ex.: POSITION X, Y

LOCATE (LOC.) — Permet de relire dans la mémoire-écran le contenu d'une case spécifiée. Dans les modes 0 à 2, on relit un caractère. Dans les modes 3 à 11, on lit le numéro de couleur.
Ex.: LOCATE X Y D (Avance à X, Y et met des données dans D).

DRAWTO (DR.) — Tire une ligne entre la dernière position du curseur et le point de coordonnées X et Y. Le curseur s'arrête aux nouveaux coordonnées. (Tire une ligne au point X, Y de la position actuelle du curseur).
Ex.: DRAWTO X-Y

TABLEAU DE CORRESPONDANCE ENTRE LES COULEURS ET LES VALEURS NUMERIQUES (INSTRUCTION SET COLOR)

Couleurs	Werte Argument Numéro 2
GRIS	0
ORANGE CLAIR (OR)	1
ORANGE	2
ORANGE-ROUGE	3
ROSE	4
VIOLET	5
BLEU-VIOLET	6
BLEU	7
BLEU	8
BLEU CLAIR	9
TURQUOISE	10
BLEU-VERT	11
VERT	12
VERT-JAUNE	13
VERT-ORANGE	14
ORANGE CLAIR	15

Remarque: Les couleurs peuvent varier selon le type et le réglage du téléviseur ou du moniteur utilisé.

VALEURS PAR DEFAUT DES COULEURS A LA MISE SOUS TENSION

Numéro du Registre Couleur	Couleur Par Défaut	Luminance Par Defaut	Resultat
0	2	8	ORANGE
1	12	10	VERT
2	9	4	BLEU FONCE
3	4	6	ROSE OU ROUGE
4	0	0	NOIR

Il y a "DEFAUT" si aucune indication de REGISTRE DE COULEUR n'est utilisée.

Remarque: Les couleurs peuvent varier selon le type de moniteur de télévision et l'état et réglage de ce dernier.

FONCTIONS

Une fonction accepte un ou plusieurs arguments pour donner finalement une nouvelle valeur. L'ordinateur ATARI possède des fonctions arithmétiques, des fonctions trigonométriques, des fonctions chaînes de caractères et quelques fonctions spéciales.

MANUEL DE REFERENCE

FONCTIONS ARITHMETIQUES

ABS — Donne la valeur absolue (sans signe) d'un nombre.
Ex.: A = ABS (B)

CLOG — Donne le logarithme base 10.
Ex.: A = CLOG (B)

EXP — Donne la valeur e (2,718 approximativement) élevée à la puissance spécifiée. Dans certains cas, la précision d'EXP ne s'étend que sur 6 digits significatifs. Inverse de LOG.
Ex.: A = EXP (B)

INT — Donne la partie entière de la valeur.
Ex.: A = INT (B)

LOG — Donne le logarithme naturel d'une valeur. Inverse de EXP.

RND — Tire un nombre aléatoire entre 0 et 1. Ne donne jamais 1.
Ex.: A = RND (0) (A = un nombre égal à 0 ou plus petit que 1)
A = RND (0) x 8 (A = un nombre égal ou supérieur à 0, et plus petit que 8)

SGN — Donne — 1 si la valeur est négative, 0 si la valeur est nulle et 1 si la valeur est positive.
Ex.: A = SGN (A)

SQR — Donne la racine carrée positive d'une valeur positive.
Ex.: A = SQR (B)

FONCTIONS TRIGONOMETRIQUES

ATN — Donne l'arctangente d'une valeur, en radians et degrés.
Ex.: A = ATN (B)

SIN — Donne le sinus de la valeur.
Ex.: A = SIN (B)

COS — Donne le cosinus d'une valeur.
Ex.: A = COS (B)

DEG — Toutes les fonctions trigonométriques seront exprimées en degrés.
Ex.: DEG

RAD — Toutes les fonctions trigonométriques seront exprimées en radians. A la mise sous tension, l'ordinateur travaille en radians.

FONCTIONS SPECIALES

ADR — Donne l'adresse mémoire décimale du début d'une chaîne.
Ex.: A = ADR (B$)
A = ADR ("CETTE CHAINE")

FRE (O) — Donne le nombre d'octets de mémoire vive utilisateur qui restent. (0) est une variable "fictive" exigée.
Ex.: A = FRE (0)
PRINT FRE (0) (Ce qui vous permettra de savoir combien il reste de mémoire libre).

PEEK — Donne le nombre stocké à un endroit spécifique de la mémoire. L'adresse est un nombre compris entre 0 et 65535. Le nombre lu sera compris entre 0 et 255.
Ex.: A = PEEK (B)

USR — Appel des sous-programmes en langage machine à partir de BASIC. USR(ADDR, P1, P2, . . . , PN), par exemple, pousse les arguments (P1 à PN) dans la pile dans l'ordre inverse. Le dernier argument "PN" est poussé dans la pile avant le premier, "P1".
Le nombre d'arguments — représenté par un seul octet — est alors poussé dans la pile. Si la fonction ne spécifie pas d'arguments, c'est zéro qui est poussé dans la pile.
Le sous-programme en langage machine à l'adresse (ADDR) est alors appelé. Si le programme en langage machine doit donner une valeur en BASIC, les octets de poids fort et de poids faible doivent être stockés aux adresses de mémoire $D4 et $D5 respectivement.
Le programme doit éliminer le nombre d'arguments et les arguments avant de revenir à BASIC, sinon le système sera complètement déréglé.
Ex.: A = USR (B, C, D)
(Le programme en B sera appelé, et les paramètres en C et D passeront au sous-programme par l'intermédiaire de la pile).

FONCTIONS DE CHAINE

ASC — Donne le numéro de code ATASCII pour le premier caractère d'une chaîne.
Ex.: A = ASC ("A") (A vaudra 65)
A = ASC (B$) (On peut utiliser des variables de chaînes)

CHR$ — Donne le caractère représenté par le numéro de code ATASCII qui a été spécifié. Réciproque de ASC.
Ex.: A$ = CHR$ (65) (A$ sera "A")

LEN — Donne un nombre qui représente la longueur d'une variable de chaîne.
Ex.: A = LEN (A$)

STR$ — Donne une chaîne représentant une valeur spécifiée (traduit un nombre en une chaîne).
Ex.: A$ = STR$ (65) (A$ sera égal à "65", qui est une chaîne)

VAL — Donne un nombre représentant une chaîne spécifique (traduit une chaîne en un nombre)
Ex.: A = VAL ("100") (A sera égal au nombre 100)

MANIPULATION DE CHAINES

Le BASIC ATARI n'emploie pas un format en tableau de chaîne pour la manipulation des chaînes. Une puissante commande d'extraction de chaîne suffit pour toutes les applications.
Exemples:
CHAÎNE EXTRAITE
50 A$ = "DAVEMARKGRETCHEN"
60 B$ = A$(9, 16) (B$ est "GRETCHEN")

CONCATENATION
50 A$ = "HI"
60 B$ = "FRED"
70 A$ (LEN (A$) + 1) = B$ (A$ est "HI FRED")

RECHERCHE D'UNE CHAINE
50 FOR Z = 1 TO LEN (A$)
60 IF A$ (Z, Z) = "E" THEN PRINT "AN EZ"
70 NEXT Z

FONCTIONS DE COMMANDE DE JEU

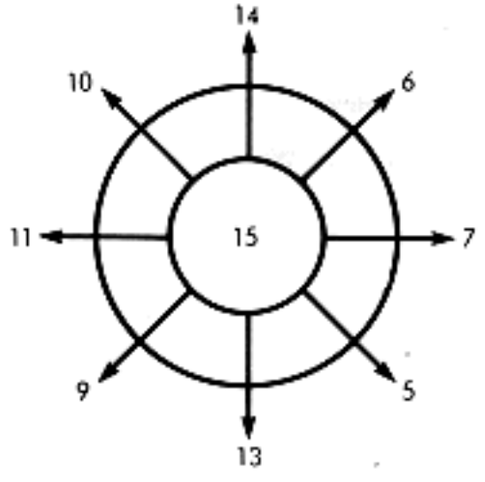
PADDLE — Donne la position du bouton d'une commande à molette spécifique. Les commandes sont numérotées de 0 à 3, de l'avant vers l'arrière. Le nombre donné se situe entre 1 et 228, augmentant au fur et à mesure que la molette est tournée vers la gauche (sens inverse des aiguilles d'une montre).
Ex.: A = PADDLE (0)

PTRIG — Le résultat est 0 si l'on presse le bouton rouge de la commande à molette correspondante, 1 dans le cas contraire. Les commandes sont numérotées de 1 à 3.
Ex.: A = PTRIG (0)

STICK — Donne l'état d'une commande à levier spécifique. Les commandes à levier sont numérotées 0 et 1, de l'avant vers l'arrière. Pour les détails, voir le diagramme.
Ex.: A = STICK (0)

STRIG — Donne un 0 si l'on presse le bouton rouge de la commande à levier correspondante, et 1 dans le cas contraire. Les commandes à levier sont numérotées 0 et l.
Ex.: A = STRIG (0)
Commande 1 = STICK (0)
Commande 2 = STICK (1)

Le diagramme qui suit donne les valeurs qui sont renvoyées lorsque la commande à levier est déplacée dans une direction quelconque.



TOUCHES DE FONCTIONS SPECIALES

ESC — A pour effet d'afficher la touche suivante sous forme de caractère international ou de caractère graphique.

BREAK — Provoque l'arrêt d'un programme BASIC.

RESET — Arrête un programme en cours d'exécution, remet l'écran en mode graphique 0, efface l'écran et met fin à vos fichiers sans les fermer. N'efface pas votre programme.

SET-CLR-TAB — Déplace le curseur jusqu'à la prochaine position de tabulation prédéterminée.

SHIFT SEF-CLR-TAB — Elimine une tabulation.

CTRL 1 — Fige ou libère le défilement de l'affichage à l'écran.

CTRL 2 — Produit un son.

CTRL 3 — Indique la fin d'un fichier.

MISE AU POINT

SHIFT INSERT — Insère une ligne.

CTRL INSERT — Insère un caractère

DELETE BACK S — Supprime le caractère à gauche du curseur et fait reculer le curseur.

SHIFT DELETE BACK S — Supprime une ligne.

CTRL DELETE BACK S — Supprime le caractère qui se trouve au curseur et déplace le reste de la ligne pour remplir l'espace vide.

SHIFT CLEAR (ou) **CTRL CLEAR** — Efface l'écran.

CTRL ↑ Déplace le curseur vers le haut.

CTRL ↓ Déplace le curseur vers le bas.

CTRL ← Déplace le curseur vers la gauche.

CTRL → Déplace le curseur vers la droite.

MANUEL DE REFERENCE

CODE DES ERREURS

NUMERO DE CODE D'ERREUR	MESSAGES DES CODES D'ERREUR
2	Mémoire insuffisante.
3	Valeur erronée.
4	Trop de variables.
5	Erreur de longueur pour une chaîne de caractères.
6	Dépassement de données.
7	Nombre plus grand que 32767.
8	Erreur dans une instruction INPUT.
9	Erreur de dimension (DIM) de tableaux ou de chaînes de caractères.
10	Dépassement de pile.
11	Dépassement de capacité en virgule flottante.
12	Ligne inexistante.
13	Instruction NEXT sans FOR.
14	Ligne trop longue.
15	Suppression de l'instruction GOSUB ou FOR.
16	Erreur sur une instruction RETURN.
17	Erreur de mémoire.
18	Caractère non valide.
19	Chargement d'un programme (LOAD) trop long.
20	Erreur du numéro de périphérique.
21	Erreur de chargement. (LOAD)

REMARQUE: Vour trouverez ci-dessous les erreurs d'ENTREE/SORTIE découlant de l'utilisation des unités de disquettes, imprimantes ou autre équipement accessoire. De plus amples informations sont fournies avec le matériel (''hardware'') auxiliaire.

128	BREAK pendant une opération d'entrée /sortie.
129	IOCB déjà ouvert.
130	Périphérique inexistant.
131	Opération de lecture sur un périphérique ouvert en écriture seulement. (IOCB)
132	Commande invalide.
133	Périphérique ou fichier non ouvert.
134	Mauvais numéro d'IOCB.
135	Tentative d'écriture vers un périphérique ouvert en lecture seule. (IOCB)
136	Fin de fichier. (EOF)
137	Enregistrement incomplet.
138	Pas de réponse du péri phérique dans le temps normal.
139	Le périphérique ne répond pas correctement. (NAK)
140	Erreur sur le bus série en entrée.
141	Le curseur dépasse la gamme autorisée.
142	Erreur de transmission dans le bus série.
143	Erreur sur le bus série en sortie.
144	Erreur d'opération sur le périphérique.
145	Erreur de vérification après écriture sur disque.
146	Fonction non implémentée.
147	Mémoire insuffisante.
160	Erreur sur le numéro de disque.
161	Trop de fichiers ouverts (OPEN) simultanément.
162	Disquette pleine.
163	Erreur système en entrée/ sortie non corrigeable.
164	Erreur de chaînage entre les sectreurs du disque.
165	Erreur de longueur pour l'instruction POINT.
167	Fichier verrouillée.
168	Commande invalide.
169	Sommaire de disquette plein.
170	Fichier non trouvé.
171	Instruction POINT invalide.
172	Ajout interdit
173	Mauvais Format

Note: Voir page 11 (ERROR CODES) dans la version anglaise comment le message codé d'erreurs apparaitra sur l'écran de votre téléviseur.

Téléchargé sur
Le Vieux Manuel

Http://www.abandonware-manuels.org