

TéléBASIC

par Joël BERNARD



Avertissement

L'Association CLUB APPLE se réserve le droit d'apporter des améliorations au produit décrit dans ce manuel à tout moment et sans préavis.

Limites de garanties et de responsabilité

Bien que le CLUB APPLE et Joël BERNARD aient testé les logiciels décrits dans ce manuel et revu son contenu ni le CLUB APPLE, ni Joël BERNARD n'offrent de garantie, explicite ou implicite, en ce qui concerne le présent manuel ou les logiciels décrits dans ce manuel quant à leur qualité, leurs performances, leur négociabilité, ou leur aptitude à un usage particulier.

Les logiciels du CLUB APPLE sont vendus "en l'état". L'intégralité du risque, quant à leurs qualités et leurs performances, est à la charge de l'acheteur. Il appartient donc à l'utilisateur de s'assurer par lui-même qu'ils correspondent bien à ses besoins, et le CLUB APPLE ne saurait apporter aucune garantie sur ce point.

En conséquence, ce manuel et ces logiciels sont vendus tels quels et l'acheteur supporte tous les risques en ce qui concerne leur qualité et leur fonctionnement. Le CLUB APPLE, et le concepteur du logiciel Joël BERNARD, ne pourront, en aucun cas, être tenus pour responsables des préjudices directs ou indirects, de quelque nature que ce soit, résultant d'une imperfection dans le manuel ou les programmes, même s'ils ont été avisés de la possibilité que de tels préjudices se produisent. En particulier, ils ne pourront encourir aucune responsabilité du fait des programmes ou données mémorisés, exploités ou développés grâce au logiciel TélÉBASIC, y compris pour les coûts de récupération de ces programmes ou données.

Certains pays n'acceptent pas l'exclusion ou la limitation des garanties implicites ou des responsabilités pour des dommages accidentels ou par voie de conséquence, de tel sorte que la limitation ou exclusion ci-dessus ne pourront pas être appliquées.

Le CLUB APPLE est propriétaire de tous les droits relatifs au logiciel TélÉBASIC.

Ce manuel et le programme qu'il décrit ont été déposés tout droit réservé. Aux termes de la législation relative aux droits d'auteur, ce manuel et le programme ne peuvent être copiés, en tout ou partie, sans le consentement écrit du CLUB APPLE. Toute copie, duplication ou communication, par quelque procédé que ce soit, effectuées sans autorisation écrite du CLUB APPLE constitueraient une violation de la loi et exposeraient l'auteur à des poursuites judiciaires. Aux termes de la législation, copie signifie également traduction dans une autre langue ou format.

© Joël BERNARD, 1984

© CLUB APPLE, 1985

43, avenue de la Grande Armée
75016 PARIS

Le logo Apple et les mots Apple et Applesoft sont des marques déposées d'Apple Computer, Inc.

TéléBASIC

(Version 1.0)

Manuel de référence

Remerciements

Je remercie particulièrement Patrick Potier et Roland Moréno pour leur contribution sans laquelle TéléBASIC n'existerait pas.

TABLE DES MATIERES

DEFINITIONS	4
PRESENTATION	5
1) TéléBASIC	5
2) Configuration nécessaire	6
3) Comment démarrer TéléBASIC	6
4) Exemples de programmation	7
5) Droits attachés au BASIC Applesoft	7
GENERALITES	8
1) Principales différences avec le BASIC Applesoft	8
2) Mots-clés générés par la touche CTRL	10
3) Le partage écran	10
4) Apple II Plus	11
5) Interprétation des commandes BASIC	11
6) Commandes DOS	11
7) Arithmétique hexadécimale	11
8) Implantation de TéléBASIC	12
COMMANDES SYSTEME	13
COMMANDES D'EDITION, COMMANDES DIVERSES	16
COMMANDES D'ENTREES-SORTIES	19
INSTRUCTIONS DE BRANCHEMENT	22
INSTRUCTIONS DE COMMUNICATION	26
INSTRUCTIONS LIEES A L'AFFICHAGE EN MODE VIDEOTEX	37
EXEMPLES	39
UTILISATION DE LA PAGE ZERO	42
CONCESSION DE LICENCE DE LOGICIEL	43

Définitions

CONVENTIONS UTILISEES DANS CE MANUEL

Afin de définir la syntaxe des instructions de TéléBASIC, nous avons utilisé les symboles suivants :

- < > symboles utilisés pour délimiter des expressions optionnelles,
- | symbole utilisé pour séparer différentes options,
- { } symboles utilisés pour séparer des expressions pouvant être répétées.

linnum représente un numéro de ligne.

expression représente une expression, au sens du terme défini dans le manuel du BASIC Applesoft.

MODES D'EXECUTION

imm : mode d'exécution immédiat (l'instruction tapée au terminal est immédiatement exécutée),

dif : mode d'exécution différé, ou encore **mode programme** (l'instruction doit être précédée d'un numéro de ligne, afin d'être exécutée ultérieurement).

PRESENTATION

1) TélÉBASIC

TélÉBASIC est un interpréteur BASIC.

A la différence cependant des interpréteurs classiques que l'on rencontre habituellement sur les micro-ordinateurs, y compris l'Apple II, TélÉBASIC est en plus un outil de communication permettant d'utiliser le modem **Apple-Tell** ou le port série de l'Apple IIc (ou de l'Apple IIe muni d'une carte **Super Serial Card**) relié à un modem RS-232 et de concevoir ainsi des applications entièrement orientées vers les télécommunications et la télématique de façon générale : serveurs, messageries, terminaux automatiques, redirecteurs, répondeurs vidéotex, etc.

Dérivé de l'**Applesoft™**, TélÉBASIC est en fait un interpréteur BASIC complet, possédant certaines des fonctions (if..then..else, print using, etc.) qui étaient considérées généralement comme faisant défaut à l'Applesoft, ainsi que tout un ensemble d'outils de communication totalement intégrés au langage BASIC.

La réalisation de TélÉBASIC correspond au souci de disposer d'un langage de programmation permettant d'écrire **facilement** (quel langage plus simple que BASIC ?) des applications faisant appel aux télécommunications, sur la base des ressources complètes, et intégrées, fournies en particulier par la carte **Apple-Tell** : modem universel à appel et réponse automatiques.

Les applications de TélÉBASIC sont nombreuses :

- serveur Vidéotex 1200/75 bauds (les terminaux de consultation étant des terminaux Minitel ou d'autres ordinateurs en mode émulation Minitel).
TélÉBASIC possède pour cela des commandes particulières à ce mode de transmission. Les applications sont, là aussi, nombreuses : messageries, journaux cycliques, prise de commandes, distributeurs d'informations...
- serveur 300 bauds monoporte (c'est-à-dire supportant un seul appel à la fois) normes CCITT ou BELL (messageries, panneau d'annonces, distributeurs d'informations diverses, etc...).
- ante-serveur (petit frontal pour mini-ordinateurs) chargé de gérer la ligne PTT et la couche vidéotex d'une application tournant sur le mini.
- terminal 300 bauds (CCITT et BELL) ou même 1200/75 bauds (CCITT) intégré. Il n'est pas nécessaire de quitter TélÉBASIC et de charger un logiciel spécialisé pour se connecter à des serveurs 300 bauds ou 1200/75 comme le réseau Calvados par exemple.
- outil de communication entre ordinateurs (300 ou 1200/75 bauds). Dans le cas de deux Apple II on peut très bien imaginer des jeux à distance (jeux d'échecs, othello...) dont l'interface entre les deux Apple serait réalisée en BASIC.

- terminal de consultation intelligent, permettant l'accès automatisé à des serveurs ainsi que la gestion et le traitement local des informations reçues.
- "serveur-servi" c'est-à-dire serveur diffusant une information **enrichie** par l'Apple lui-même, sur la base de données qu'il est allé consulter automatiquement chez d'autres serveurs ou auprès de banques de données. Exemple : revue de presse, cours de valeurs mobilières, jurisprudences, etc.

2) Configuration nécessaire

TéléBASIC nécessite :

- soit un Apple IIc,
- soit un Apple IIe,
- soit un Apple II Plus 64 K (c'est-à-dire avec une extension mémoire 16 K).

Dans le cas d'un Apple IIe ou d'un Apple II Plus, les commandes relatives à l'utilisation d'un modem ne sont évidemment disponibles que si l'on possède soit une **carte Apple-Tell**, soit une **carte Super Serial Card** et un modem RS-232 comme le **modem universel Apple**.

Les cartes **Apple-Tell** et **Super Serial Card (SSC)** peuvent être installées dans tout slot (connecteur) libre de l'Apple, TéléBASIC détectant le slot occupé par la carte. Si les deux cartes sont présentes lors du chargement de TéléBASIC, priorité est donnée au modem **Apple-Tell**, seules seront alors disponibles les commandes relatives à cette carte.

Dans le cas d'un Apple IIc, le port série de communications sera utilisé. Ce port correspond au slot 2 de l'Apple IIc.

3) Comment démarrer TéléBASIC

Avec ce manuel, vous avez trouvé deux disquettes TéléBASIC. Elles sont identiques. Vous utiliserez en général une de ces disquettes. La deuxième sera la disquette de sauvegarde. Nous vous conseillons très fortement de la mettre à l'abri de toutes les sources de poussières ou de chaleur, champs magnétiques, etc... en un mot de tous les ennemis héréditaires des disquettes. elle vous servira en cas d'incident sur votre disquette habituelle. Une procédure de remplacement est prévue en cas d'ennuis. Vous trouverez dans cet emballage toutes les instructions nécessaires pour les échanges ou réparations de disquette.

La mise en route de TéléBASIC est très simple. Il suffit de mettre la face 1 de la disquette TéléBASIC dans le lecteur de disquette puis de mettre l'unité centrale sous tension ou de procéder à une réinitialisation de la machine. Si vous ne savez pas comment procéder, nous vous conseillons de vous reporter au manuel d'utilisation de votre unité centrale Apple. L'interpréteur va alors se charger en mémoire centrale. Lorsque le message "**Veillez retirer votre disquette TéléBASIC du lecteur...**" apparaîtra à l'écran, vous devez retirer la disquette TéléBASIC de votre lecteur et insérer une disquette vierge ou initialisée DOS. Elle servira à stocker vos fichiers BASIC.

4) Exemples de programmation TéléBASIC

La face 2 de la disquette TéléBASIC (verso) contient plusieurs programmes écrits en TéléBASIC et quelques fichiers d'image vidéotex qui seront utilisés par le programme de démonstration du serveur vidéotex. Contrairement à la face 1 TéléBASIC, cette face n'est pas protégée contre la copie. Nous vous conseillons donc d'en faire une sauvegarde selon la procédure habituelle (COPYA).

5) Droits attachés à l'interpréteur BASIC Applesoft

TéléBASIC ne peut fonctionner, **ni même se charger** dans la mémoire de l'Apple, que si l'interpréteur BASIC Applesoft est déjà implanté en ROM sur la carte-mère de l'appareil. Ainsi sont garantis les droits de propriété intellectuelle détenus par les auteurs de l'Applesoft.

IMPORTANT

L'utilisation de ce manuel implique une connaissance préalable du langage BASIC, et en particulier de l'Applesoft. Nous vous recommandons la lecture préalable du "Manuel de référence du BASIC Applesoft, volume 1 et 2" (n° de produit Apple : A2L 2008F).

GENERALITES

1) Principales différences avec le BASIC Applesoft

Beaucoup de mots-clés du TéléBASIC sont communs avec ceux de l'Applesoft, ainsi que les codes correspondants, ceci afin de garder un degré maximum de compatibilité.

Seules les instructions relatives au mode **graphique basse résolution** de l'Apple II et à la gestion du lecteur de cassette (largement obsolètes fin 1984 !) ne sont plus présentes dans TéléBASIC.

Une exception cependant : les commandes HTAB et VTAB sont remplacées par la commande unique CURSOR.

Mots-clés de l'Applesoft inconnus de TéléBASIC

GR	instructions relatives au mode basse résolution
SCRN	
HLIN	
VLIN	
PLOT	
LOAD	instructions relatives à l'utilisation d'un lecteur de cassette
SAVE	
SHLOAD	
STORE	
RECALL	
NOTRACE	TRACE et NOTRACE ont été remplacés par TRACE ON et TRACE OFF
IN #	ce sont des commandes du DOS
PR#	
XDRAW	

Mots-clés de TélÉBASIC inconnus de l'Applesoft

AUTO	
BEEP	
BYE	
ELSE	
LINPUT	
MENU	
OFF	utilisé dans TRACE OFF, AUTO OFF
ON	utilisé dans TRACE ON et ON GOTO, GOSUB
STATUS	
SUBSTR	
SWAP	
USING	utilisé dans PRINT USING, SEND USING
WINDOW=	

instructions de télécommunications :

BCOLOR=	
COMMUNICATE	
CHRSIZE=	
ECHO=	
FCOLOR=	
HANGUP	
LINE	utilisé dans les instructions GET LINE, HOME LINE, etc.
LOWERCASE	
RDCHR	
RECEIVE	
SEND	
TERMINAL	
TIMEOUT=	

Plusieurs instructions TélÉBASIC concernent le mode vidéotex. Ces instructions sont étroitement liées aux possibilités de la carte interface correspondante. Elles ne seront donc disponibles que pour les Apple II équipés de la carte **Apple-Tell**.

2) Mots-clés générés par l'utilisation de la touche CTRL

L'appui simultané de la touche CTRL (Control) et de l'une des touches suivantes provoque l'affichage et la mise dans le "buffer" d'entrée du mot-clé correspondant, ce qui accélère la frappe (souvent fastidieuse) des programmes BASIC.

A	ASC
B	USING
C	CURSOR
D	DATA
E	ELSE
F	FOR
G	GOSUB
I	INPUT
J	LEFT\$
K	RIGHT\$
N	NEXT
P	PEEK
Q	POKE
R	RETURN
S	TERMINAL
T	THEN
W	WAIT
Z	RESTORE

Il est néanmoins possible d'insérer dans un programme des caractères de contrôle, en tapant au préalable CTRL-O.

exemple :

si l'on veut ajouter un CTRL-J dans une ligne Basic afin d'améliorer la mise en page d'un programme, il suffit de taper la suite de touches : CTRL-O CTRL-J.

3) Le partage écran

La touche *CTRL-W* (pour WINDOW), tapée en début de ligne BASIC, agit comme une bascule pour définir rapidement une fenêtre d'écran. Ce dispositif s'avère pratique à l'usage, dans le cas où l'on désire visionner une partie de programme tout en modifiant celui-ci.

La première frappe de la touche CTRL-W définit la partie supérieure de l'écran comme fenêtre active, la seconde la partie inférieure de l'écran, et la troisième revient à l'état initial.

4) Apple II Plus

TéléBASIC fonctionne également sur l'Apple II Plus équipé d'une extension mémoire de 16 Kilo-octets (appelée également carte langage).

Ce matériel ne disposant pas d'origine de caractères minuscules, TéléBASIC convertit en caractères majuscules tous les caractères avant de les afficher à l'écran. Attention cependant que la conversion n'ait lieu qu'à l'affichage, les caractères en mémoire pouvant être des caractères minuscules ainsi que les caractères reçus via modem (utilisez *LOWERCASE* pour ne pas avoir de surprises...)

5) Interprétation des commandes BASIC

Une commande BASIC non précédée d'un numéro de ligne est immédiatement exécutée : il s'agit du mode d'exécution immédiat noté *imm* dans la suite de ce manuel. Une commande précédée d'un numéro de ligne est enregistrée en mémoire pour être ultérieurement exécutée : il s'agit du mode d'exécution différé noté *dif* par la suite.

Seule différence avec le BASIC Applesoft, **les commandes entrées en caractères minuscules sont acceptées.**

6) Commandes DOS

Le système d'exploitation de disquette est le DOS 3.3, celui-ci présentant cependant à l'usage quelques différences avec la version habituellement pratiquée :

- les commandes INT, FP ne sont plus disponibles,
- **les commandes en caractères minuscules sont acceptées.**

7) Arithmétique hexadécimale

TéléBASIC permet la manipulation de **nombre hexadécimaux** codés sur 2 octets (0-\$FFFF), ces nombres étant signés (\$FFFF = -1). Par convention, tout nombre hexadécimal doit débiter par le signe \$.

exemples :

- | | |
|--------------------------------|--------------------------------------|
| • ? \$100 | affiche le nombre 256 |
| • x = peek (\$C000) | permet la lecture du clavier |
| • poke \$300,\$60 : call \$300 | est licite (\$60 est le code de RTS) |
| • ? \$100 *4 | affiche 1024 |

8) Implantation de TéléBASIC

TéléBASIC est implanté dans la carte 16K, dite "carte langage", et utilise également la zone mémoire \$8D00-\$9D00, la plus haute adresse mémoire disponible étant donc \$8600 avec MAXFILES initialisé à 3 (valeur par défaut).

COMMANDES SYSTEME

1) RUN imm & dif

run < *expression* >

Exécute le programme en mémoire, après avoir effacé toutes les variables.

run exécute le programme en mémoire.

run *expression* exécute le programme en mémoire, à partir du numéro de ligne évalué dans *expression*. Si la ligne n'existe pas, on obtient le message *UNDEF'D STATEMENT ERROR*.

exemple :

• **A = 10 : run 100 + A** provoque un début d'exécution en ligne 110.

Remarque :

En mode immédiat, *RUN A* ne provoquerait pas l'exécution du programme à partir de la ligne 10, mais l'exécution du programme "A" contenu sur disquette; ceci est dû au DOS, qui capture l'ordre *RUN* comme un ordre disque, ne laissant pas s'exécuter l'ordre BASIC. Il est possible de pallier à cette particularité du DOS en ajoutant un ":" avant la commande *RUN* .

2) TRACE ON imm & dif

Provoque la mise en fonction du mode de correction d'erreurs.

Cette commande remplace la commande Applesoft *TRACE* .

3) TRACE OFF imm & dif

Mise hors fonction du mode TRACE.

Remplace la commande Applesoft *NOTRACE* .

4) WAIT imm & dif

wait < / > expression

Provoque une pause dont la durée en dixièmes de secondes est approximativement égale à la valeur d'*expression*. L'option "/" permet d'interrompre cette pause en pressant une touche. Le clavier est remis à zéro à la fin de l'instruction.

exemples :

- **wait 20** pause de 2 secondes (ininterrompible)
- **wait A** pause de A * 0,1 secondes environ
- **wait / 100** pause de 10 secondes environ (pouvant être interrompue par l'appui d'une touche quelconque du clavier).

5) CTRL-C dif

ctrl-c arrête l'exécution du programme en mémoire. Peut être contrôlé par la commande *ONERR GOTO*. Cette commande est également prise en compte lors de l'attente de réception de caractères (commandes *RECEIVE* et *GET LINE*).

Remarque :

Cette fonction est inopérante lors de l'exécution de la fonction menu.

6) BYE imm & dif

bye est équivalent à l'instruction *CALL-151* qui appelle le moniteur. Le retour au BASIC se fait par les instructions :

- ctrl-b* retour à froid (programme et variables effacés),
- ctrl-c* retour à chaud (programme et variables sauvegardés).

7) STATUS imm

status affiche le nombre d'octets occupés par le programme en mémoire ainsi que le nombre d'octets disponibles. Si l'on est en communication, le message *OFF HOOK* est affiché; dans le cas contraire le message *ON HOOK* est affiché. L'emploi de cette commande en mode différé provoque l'affichage de *ILLEGAL DEFERRED ERROR*.

8) AUTO imm & dif

auto llnum1 < ,llnum2 >

Numérotation automatique des lignes de programme :

- *llnum1* est le numéro de ligne initial
- *llnum2* est l'incrément entre les lignes BASIC. Par défaut, *Llnnum2* = 100.

L'appel du numéro de ligne s'effectue en appuyant sur la barre d'espacement en début de ligne.

exemple :

- **auto 100,10** définit comme première ligne BASIC la ligne 100. L'appui sur la barre d'espacement affichera les numéros de ligne 110, 120, 130...

Remarques :

- *auto 0* provoque le message *syntax error*.
- Si *llnum2* est nul, on obtient le message *ILLEGAL QUANTITY ERROR*.

9) AUTO OFF imm & dif

auto off : mise hors fonction de la commande *auto*. La barre d'espacement retrouve sa fonction usuelle.

10) REMOTE imm & dif

remote : mode émulation Apple II . Permet la maintenance de programmes à distance.

Remarques :

- en mode Vidéotex les touches spéciales du terminal Minitel (*Envoi, correction* ...) ne sont pas prises en compte. Utilisez la touche spéciale afin d'obtenir un vrai retour chariot.
- CTRL-S et CTRL-C n'arrêtent plus le déroulement d'un listing.

11) LOCAL imm & dif

local : rends la main à l'Apple IIe. A utiliser après la commande *REMOTE* .

COMMANDES D'EDITION , COMMANDES DIVERSES

1) LIST imm & dif

list < *linnum1*, > < *linnum2* >

liste tout ou partie du programme en mémoire.

list	liste tout le programme
list <i>linnum1</i>	liste la ligne <i>linnum1</i>
list <i>linnum1</i>,	liste le programme à partir de la ligne <i>linnum1</i> jusqu'à la fin
list <i>linnum1</i>,<i>linnum2</i>	liste de la ligne <i>linnum1</i> jusqu'à la ligne <i>linnum2</i>
list,<i>linnum2</i>	liste le programme du début jusqu'à la ligne <i>linnum2</i> .

ESC provoque l'arrêt définitif du déroulement du listing. Tout autre caractère provoque un arrêt momentané du déroulement du listing, arrêt s'effectuant au prochain caractère correspondant à la touche appuyée. (L'appui sur la touche *RETURN* permet de lister ligne à ligne et la barre d'espacement permet de lister mot à mot.)

exemples :

• list 10,100	liste le programme de la ligne 10 à la ligne 100
• list ,100	liste le programme du début jusqu'à la ligne 100
• list 100,	liste le programme de la ligne 100 jusqu'à la fin.

2) DEL imm & dif

del < *linnum1*, > < *linnum2* >

efface tout ou partie du programme en mémoire, remet à 0 toutes les variables numériques et initialise toutes les variables chaîne à la chaîne nulle.

del <i>linnum1</i>	efface la ligne <i>linnum1</i>
del <i>linnum1</i>,	efface le programme à partir de la ligne <i>linnum1</i> jusqu'à la fin
del <i>linnum1</i>, <i>linnum2</i>	efface de la ligne <i>linnum1</i> jusqu'à la ligne <i>linnum2</i>
del ,<i>linnum2</i>	efface le programme du début jusqu'à la ligne <i>linnum2</i>

exemples :

• del 10,100	efface les lignes 10 à 100
• del ,100	efface le programme du début jusqu'à la ligne 100
• del 100,	efface le programme de la ligne 100 jusqu'à la fin.

Remarques :

- **list, del, :** liste et efface respectivement tout le programme.

3) CURSOR imm & dif

cursor < expression1 > < ,expression2 >

Positionne le curseur aux coordonnées *expression1, expression2*.

Expression1 doit être compris entre 0 et 79 et *expression2* entre 0 et 23 sinon on obtient le message *ILLEGAL QUANTITY ERROR*.

exemples :

- **cursor 10** est équivalent à l'instruction Applesoft *HTAB 10*
- **cursor 30, 20** est équivalent aux instructions Applesoft *VTAB 20: HTAB 30*
- **cursor, 20** est équivalent à l'instruction Applesoft *VTAB 20*.

Remarque :

L'utilisation d'une carte 80 colonnes est facilitée par le fait que *expression1* peut être supérieur à 40, ce qui évite l'utilisation du *POKE 36, expression1* comme c'était le cas dans le BASIC Applesoft.

4) CLEAR imm & dif

clear < variable1 > <{ ,variable2 }>

Efface tout ou partie des variables du programme BASIC en mémoire.

clear : efface toutes les variables du programmes

clear variable : réinitialise la variable dimensionnée *variable* à 0 s'il s'agit d'une variable numérique ou à la chaîne nulle s'il s'agit d'une variable chaîne. Permet de récupérer la place utilisée par des variables locales.

exemples :

- **clear** efface toutes les variables
- **clear A** réinitialise le tableau A à zéro
- **clear A,B,A\$** réinitialise les tableaux A et B à zéro, et le tableau A\$ à la chaîne nulle.

5) SWAP imm & dif

swap variable1 , variable2

Echange le contenu des variables *variable1* et *variable2* . Les variables doivent être de même type (chaîne ou numérique), sinon on obtient le message *TYPE MISMATCH ERROR*.

exemples :

- swap A,B , swap A%,B et swap A\$,B\$ sont corrects, alors que swap A\$,B n'est pas légal.

6) BEEP imm & dif

beep

Emet un petit bip sonore.
[Remplace CALL 64477.]

7) SUBSTR imm & dif

substr (expression chaîne , variable chaîne)

Cette fonction retourne le rang de la dernière occurrence de *variable chaîne* dans *expression chaîne*.
Si *variable chaîne* n'a pas été trouvée, la valeur retournée est 0.

exemples :

- A\$="envoi" : ? substr ("Tapez envoi pour continuer" , A\$) retourne la valeur 7
- ? substr ("", A\$) donne 0
- A\$="" : ? substr ("hello" , A\$) retourne 1.

COMMANDES D'ENTREES-SORTIES

1) INPUT dif

Input < chaîne; > variable <{ ,variable }>

Permet l'entrée de variables numériques ou de variables chaînes à partir du terminal.
Cette commande est identique à la commande Applesoft.

2) LINPUT dif

Linput < chaîne; > variable chaîne

Permet l'entrée d'une variable chaîne à partir du terminal.

Si *chaîne* n'est pas présent, un point d'interrogation est affiché, et l'entrée de la variable est effectuée.

L'entrée d'une seule variable chaîne est permise .

La commande *linput* permet l'entrée de tous les caractères alphanumériques (y compris les caractères ":" et ";"). Seul le retour chariot détermine la fin de la chaîne.

Ctrl-c arrête l'instruction *LINPUT* s'il est le premier caractère entré.

Utiliser l'ordre *LINPUT* en mode direct provoque le message *ILLEGAL DIRECT ERROR*.

Si la variable n'est pas une variable de type chaîne, on obtient le message *TYPE MISMATCH ERROR*.

3) GET dif

Get variable : permet la saisie d'un caractère mis dans *variable*.

Dans le cas d'une variable chaîne, tous les caractères sont acceptés, y compris le retour chariot.

Pour une variable numérique, seuls les caractères numériques (0 à 9) sont acceptés; l'entrée d'un caractère non numérique est ignorée, et ne provoque pas de message d'erreur.

L'instruction *GET* en mode immédiat provoque le message *ILLEGAL DIRECT ERROR*.

4) PRINT imm & dif

Print <{ [*expression numérique*] *expression chaîne* | *expression* } <{ , | ; <{ *expression numérique*] *expression chaîne* | *expression* }> }> > < , | ; >

Identique à l'instruction correspondante du BASIC Applesoft, avec en plus la possibilité de répéter une chaîne de caractères. L'instruction *print [expression1] expression2* avec *expression2* de type numérique provoque le message *TYPE MISMATCH ERROR* .

exemple :

L 'instruction **print [9]" - ";" Télébasic ";[9]" -"** est équivalente à l'instruction **print "-----";" Télébasic ";"-----"**

5) PRINT USING imm & dif

Print using expression chaîne ; expression numérique <{ , | ; < *expression numérique* }> > < ; >

Imprime le ou les expressions numériques selon le format indiqué par l'*expression chaîne*.

Si les expressions numériques sont séparées par un point-virgule, on n'imprime pas de retour chariot.

L'image selon laquelle un nombre résultant d'une expression numérique sera imprimé est fonction des caractéristiques du format :

- le nombre est placé dans le champ défini par le format avec justification à droite
- le premier point à partir de la droite définit la position du point décimal (et donc le nombre de décimales)
- le nombre affiché est arrondi
- le caractère # indique les emplacements qui seront remplacés par des chiffres. Tout autre caractère (excepté le point décimal) est non-remplaçable
- le nombre affiché peut comporter jusqu'à neuf chiffres significatifs
- si le nombre est trop long (ou si le champ est trop petit), le champ est entièrement rempli de ?

exemples :

• **print using "##.##";4/3**

affiche la chaîne " 1.33"

• **A\$= "montant = #####.## FF" : print using A\$; 100**

affiche la chaîne

"montant = 100.00 FF"

6) READ , DATA

Instructions identiques aux instructions du BASIC Applesoft.

7) RESTORE imm & dif

Restore < < next > expression > : déplace le pointeur de *DATA*

Restore : en début de programme

Restore expression : au début de la ligne dont le numéro est évalué par *expression* . Si la ligne n'existe pas on obtient le message *UNDEF'D STATEMENT ERROR*.

Restore next expression : au début de la première ligne dont le numéro est supérieur ou égal à la valeur donnée par *expression*.

exemples :

- *restore 100* déplace le pointeur de data à la ligne 100
- *A = 100 : restore A* est équivalent.

8) WINDOW imm & dif

Window = (expression1, expression2, expression3, expression4)

avec $0 \leq \text{expression1} \leq 79$
 $0 < \text{expression1} + \text{expression2} \leq 79$
 $0 \leq \text{expression3} < \text{expression4} \leq 23$

Modifie la taille de la fenêtre d'écran.

expression1 : bord gauche de la fenêtre

expression2 : largeur de la fenêtre

expression3 : bord haut de la fenêtre

expression4 : bord bas de la fenêtre.

Si les inégalités précédentes ne sont pas vérifiées, on obtient le message *ILLEGAL QUANTITY ERROR*.

exemple :

- *window = (10, 20, 10, 20)* définit une fenêtre d'écran de 20 x 10 caractères, dont le bord supérieur gauche est aux coordonnées 10, 10.

INSTRUCTIONS DE BRANCHEMENT

1) GOTO imm & dif

goto < next > expression numérique

goto expression numérique : modifie le pointeur de prochaine ligne à exécuter par la valeur de *expression*. Si la ligne cherchée n'existe pas, on obtient le message *UNDEF'D STATEMENT ERROR*.

goto next expression numérique : modifie le pointeur de prochaine ligne à exécuter par le numéro de la première ligne ayant un numéro supérieur ou égal à la valeur évaluée dans *expression*. S'il n'y en a pas, on obtient le message *UNDEF'D STATEMENT ERROR*.

exemples :

- ***goto A * 100*** provoque un branchement à la ligne 100 si A = 1, à la ligne 200 si A = 2, etc.
- ***gotomenu (2,0) *100*** provoque un branchement à la ligne 100 si *menu (2,0)* = 1, à la ligne 200 si A = 2, à la ligne 0 si A = 0 (voir fonction *menu*).

2) GOSUB..RETURN imm & dif

gosub < next > expression numérique

gosub expression numérique : exécute le sous-programme débutant au numéro de ligne évalué dans *expression*. Si la ligne cherchée n'existe pas, on obtient le message *UNDEF'D STATEMENT ERROR*.

gosub next expression numérique : exécute le sous-programme débutant au premier numéro de ligne supérieur ou égal au numéro évalué dans *expression*. Si la ligne cherchée n'existe pas, on obtient le message *UNDEF'D STATEMENT ERROR*.

exemple :

- ***gosub menu (2,0) * 100*** exécute le sous programme débutant à la ligne 100 si *menu (2,0)* = 1, à la ligne 200 si a = 2, à la ligne 0 si a = 0 (voir fonction *menu*).

3) FOR..NEXT imm & dif

Identiques aux instructions Applesoft, excepté que si plus de sept instructions *FOR..NEXT* sont imbriquées, on obtient le message *OUT OF MEMORY ERROR*.

4) ON..GOTO, ON..GOSUB dif

on expression goto {< , *expression numérique* >}
on expression gosub {< , *expression numérique* >}

Identiques aux instructions Applesoft, excepté que le numéro de ligne où est effectué le branchement peut être évalué dans une expression numérique .

Si *expression* = 0 ou supérieure au nombre d'*expressions numériques* , l'instruction *ON GOTO* ou *ON GOSUB* est ignorée.

exemple :

• *on menu (2,0) goto 100,200* est équivalent à *goto menu (2,0) * 100* pour *menu (2,0)* valant 1 ou 2. Pour *menu (2,0) = 0*, l'instruction *on menu (2,0) goto 100, 200* provoque un branchement à l'instruction suivante, alors que l'instruction *goto menu (2,0) * 100* provoque un branchement à la ligne 0.

5) IF..THEN..ELSE imm & dif

if expression then <*instruction* > <{ : *instruction* }> < : *else* <*instruction* >
<{ : *instruction* }> >

Lorsque *expression* est vraie, toutes les instructions entre *IF* et *ELSE* sont exécutées, et celles après *ELSE* sont ignorées. Lorsqu'elle est fausse, seules les instructions suivant *ELSE* sont exécutées.

exemple :

• *if a < b then print "a plus petit que b" : else print "a plus grand que b"*

if expression then linnum < : else < linnum > >

Idem avec *instruction* = *goto*.

exemples :

- *10 if a < b then 100 : else 200*
- *100 print "a plus petit que b" : end*
- *200 print "a plus grand que b" : end*

Remarques :

- *IF* et *ELSE* doivent être compris dans la même ligne de programme.
- Une instruction *ELSE* non précédée par *IF* est équivalente à une instruction *REM*.
- Plusieurs instructions *IF..THEN..ELSE* ne peuvent être emboîtées, alors que plusieurs *IF..THEN* le peuvent.

6) MENU imm & dif

menu (expression1 , expression2)

Cette fonction fournit la valeur choisie dans un menu présentant *expression1* choix. La fonction *menu* commence par écrire en mode vidéo inverse la première ligne de la fenêtre d'écran, et attend l'appui d'une touche du clavier. Les seules touches validées sont :

- la touche *ESC* (qui fournit 0 comme résultat)
- les quatre flèches directionnelles (qui déplacent la barre en mode vidéo inverse)
- la barre d'espacement qui déplace la barre en mode vidéo inverse vers le bas
- *RETURN* (qui valide le choix)
- les touches 1 à *expression1* (qui déplacent la barre directement à la valeur correspondante); si *expression1* est supérieur à 9 on utilise les touches A, B, C ...

expression2 détermine le nombre de lignes sautées à chaque déplacement de la barre en mode vidéo inverse.

expression1 doit être compris entre 1 et 24 sinon on obtient le message *ILLEGAL QUANTITY ERROR*.

La fonction *menu* est compatible avec la carte 80 colonnes Apple et reconnaît lorsque celle-ci est validée.

Remarques :

- Les lettres minuscules, ainsi que les caractères (en mode vidéo inverse et mode flash) ne sont pas modifiés par la commande *MENU* (en mode 40 colonnes uniquement).
- L'emplacement et la largeur de la barre en mode vidéo inverse sont déterminés par la fenêtre d'écran précédemment définie. (Utiliser la commande *WINDOW* pour la modifier).

exemple :

L'instruction *A = MENU (3,0)* propose un menu comportant 3 choix. La valeur de la variable *A* après cette instruction sera :

- 1, 2 ou 3 si la touche *RETURN* a été appuyée
- 0 si *ESC* a été appuyée.

7) **ONERR GOTO ,RESUME** dif

onerr goto expression numérique

Lors d'une erreur, *ONERR GOTO* provoque un branchement au numéro de ligne évalué dans *expression* . Comme pour le BASIC Applesoft, le code de l'erreur est alors obtenu en lisant le contenu de la mémoire 222 . Notez que les valeurs diffèrent.

CODE (décimal)	MESSAGE D'ERREUR
128	NEXT without FOR
129	Syntax
130	RETURN without GOSUB
131	Out of DATA
132	Illegal quantity
133	Overflow
134	Out of memory
135	Undef'd statement
136	Bad subscript
137	Redim'd array
138	Division by zero
139	Illegal direct
140	Type mismatch
141	String too long
142	Illegal deferred
143	Can't continue
144	Undef'd function
145	No communication
146	Line busy
147	No carrier
148	Not Vidéotex mode
149	Time Out
150	Connexion-Fin
255	Ctrl-C

ONERR GOTO doit être exécuté avant la première occurrence d'une erreur, afin d'éviter l'interruption du programme.

POKE 216 , 0 annule les effets de l'instruction *ONERR GOTO*.

resume : identique à l'instruction Applesoft.

INSTRUCTIONS DE COMMUNICATION

Les instructions de communication étant étroitement liées aux possibilités du modem utilisé, leur syntaxe diffère légèrement suivant que l'on utilise le modem **Apple-Tell** ou le port série de l'Apple IIc ou de la SSC. Dans la suite de ce chapitre, le sigle ¶ sera utilisé chaque fois que la syntaxe d'une instruction dépendra du modem utilisé.

1) COMMUNICATE imm & dif

communicate <, < *F* | *V* | *U* > < *A* | *O* > < *R* | *L* > < *7* | *8* > < *Y* | *N* > < *E* | *D* | *I* > < *1* | *2* > < *T* > > < # *variable chaîne* > < , *variable numérique* >

La commande *communicate*, utilisée avec différents paramètres, permet d'initialiser la communication via modem soit en mode serveur, soit en mode servi, ceci sous différentes normes de transmission. Lorsque la communication est établie, un petit bip est émis.

¶ Modem Apple-Tell

L'instruction *communicate* permet d'utiliser toutes les possibilités du modem universel **Apple-Tell** à savoir l'appel et la réponse automatique, les modes 300 bauds (CCITT et BELL) et 1200/75 bauds .

Détaillons les différents paramètres :

<i>FVU</i>	<i>F</i>	communication 300 bauds full duplex norme Francaise (norme CCITT V.21)
	<i>V</i>	communication 1200/75 bauds (duplex avec canal de retour), plus couramment appelée mode Vidéotex (norme CCITT V.23 mode 2)
	<i>U</i>	communication 300 bauds full duplex norme US (norme Bell 103)
<i>AO</i>	<i>A</i>	communication en mode serveur (Answer)
	<i>O</i>	communication en mode terminal (Originate)
<i>RL</i>	<i>R</i>	écho distant (Remote echo)
	<i>L</i>	écho local (Local echo)
<i>78</i>	<i>7</i>	7 bits de données
	<i>8</i>	8 bits de données

YN	Y	bit 8 forcé à un
	N	Bit 8 forcé à zéro
EDI	E	parité paire (Even)
	D	parité impaire (odd)
	I	absente
I2	1	1 bit de stop
	2	2 bits de stop
T		passage en mode terminal une fois la communication établie
0	0	pas de LineFeed après le retour chariot.

L'ordre des paramètres est indifférent.

Les paramètres par défaut sont *FOR7YE1* soit une communication en 300 bauds norme française (CCITT), mode terminal, écho distant, 7 bits de données avec bit 8 mis à 1, parité paire, retour à l'interpréteur BASIC lorsque la communication est établie et envoi du code LineFeed après chaque retour chariot.

a) mode terminal (mode servi)

L'instruction *communicate* en mode terminal doit être suivie du caractère # (numéro) et d'une variable chaîne qui doit contenir le numéro d'appel. Tous les caractères non numériques de la chaîne sont considérés comme une temporisation lors de l'appel. Si la variable n'est pas de type chaîne, on obtient le message *TYPE MISMATCH ERROR*.

Une fois le numéro composé, l'Apple II (ou plus exactement la carte Apple-Tell) essaie de détecter une onde porteuse. Si la temporisation de 50 secondes est dépassée, le numéro est recomposé, ceci 3 fois au maximum. Si la communication n'est pas obtenue, on obtient le message *NO CARRIER ERROR*. La temporisation d'attente de porteuse dont la valeur par défaut est égale à 50 secondes peut être modifiée en suivant le numéro à composer d'une variable,

exemple :

- **communicate,"2222890",100** représente une temporisation d'environ 50 secondes (c'est la valeur par défaut). A noter que l'option ,O (pour Originate) n'est point nécessaire, le mode servi étant le mode par défaut.
- **A=150 : communicate,"2222890",A** correspond à une temporisation d'environ deux minutes.

La valeur de la variable de temporisation doit être comprise entre 1 et 255. Dans le cas contraire, l'interpréteur émet le message *ILLEGAL QUANTITY ERROR*.

Lors de l'attente de détection de l'onde porteuse, appuyer une fois sur la touche *ESC* force la recomposition du numéro.

Il est également possible de décrocher la ligne et d'envoyer une onde porteuse sans avoir à composer un numéro téléphonique en omettant le numéro ou en donnant une valeur égale à la chaîne nulle.

exemples :

- **communicate** aura pour effet d'envoyer une onde porteuse, sans composition préalable d'un numéro.
- **communicate,#""** lui est équivalent.

b) mode serveur

En mode serveur l'instruction *communicate* doit être suivie de *A* (pour Answer) et d'autres paramètres éventuellement. L'Apple se met alors en attente d'appel; lorsque la sonnerie retentit, une onde porteuse est générée, et l'Apple essaie ensuite de détecter une onde porteuse; si celle-ci n'est pas détectée, l'Apple retourne en attente d'appel, l'instruction suivante n'étant exécutée que lorsque la communication est établie. L'appui de la touche *ESC* force l'Apple à envoyer une onde porteuse et à passer à l'instruction suivante.

exemples :

- **communicate,T,#"16019100"** appelle TRANSPAC en mode 300 bauds, en mode non serveur avec passage en mode terminal écran une fois la communication établie.
- **communicate,VA** transforme l'Apple en serveur Vidéotex.

Remarques :

la "tokenisation" des lignes BASIC (c'est-à-dire la traduction des mots-clés BASIC en codes prédéfinis) peut poser des problèmes lors de l'emploi des paramètres de *communicate* (utilisez par exemple OT plutôt que TO qui est un mot-clé du BASIC).

¶ port série

L'instruction *communicate* permet de communiquer via la sortie RS-232 sous différentes vitesses de transmission (75, 300 et 600 bauds). Le mode 1200/75 n'est pas disponible.

Détaillons ici les différents paramètres :

<i>UVW</i>	<i>U</i>	communication 75 bauds
	<i>V</i>	communication 300 bauds, c'est le mode par défaut
	<i>W</i>	communication 600 bauds
<i>AO</i>	<i>A</i>	communication en mode serveur (Answer)
	<i>O</i>	communication en mode terminal (Originate)
<i>RL</i>	<i>R</i>	écho distant (Remote echo)
	<i>L</i>	écho local (Local echo)
<i>5678</i>	<i>5</i>	5 bits de données
	<i>6</i>	6 bits de données
	<i>7</i>	7 bits de données
	<i>8</i>	8 bits de données
<i>EDIMS</i>	<i>E</i>	parité paire (Even)
	<i>D</i>	parité impaire (odd)
	<i>I</i>	absente
	<i>M</i>	bit 8 mis à 1 (Mark parity)
	<i>S</i>	bit 8 mis à 0 (Space parity)
<i>I2</i>	<i>1</i>	1 bit de stop
	<i>2</i>	2 bits de stop
<i>T</i>	passage en mode terminal une fois la communication établie	
<i>0</i>	<i>0</i>	pas de LineFeed après le retour chariot

L'ordre des paramètres est indifférent.

Les paramètres par défaut sont *VOR7I1* soit une communication en 300 bauds, mode terminal, écho distant, 7 bits de données avec bit 8 mis à 1, pas de parité, retour à l'interpréteur BASIC lorsque la communication est établie et envoi du code LineFeed après chaque retour chariot.

L'instruction *communicate* peut être interrompue par la frappe de CTRL-C.

2) HANGUP imm & dif

hangup

Cette commande permet de raccrocher si une communication est établie. Pas de paramètres.
Si la ligne n'est pas occupée, on obtient le message *NO COMMUNICATION ERROR*.

Remarque :

L'appui sur la touche RESET ou CTRL-RESET a également pour effet (entre autres) de mettre un terme à la communication.

3) TERMINAL imm & dif

terminal

L'instruction *terminal* permet de passer en mode terminal écran (on quitte alors l'interpréteur BASIC), terminal semblable à celui obtenu en tapant la commande IN # s (où s est le numéro du slot contenant la carte Apple-Tell).

Des différences cependant :

- la touche DEL n'est plus inopérante, celle-ci est équivalente à la touche <-
- en mode Vidéotex, l'appui de la touche RETURN génère un line-feed suivi d'un retour chariot (ce qui permet de communiquer directement avec un terminal Minitel par exemple)
- la touche -> est ignorée, afin de ne pas désactiver la carte 80 colonnes Apple IIe (le cas échéant).

Le retour à l'interpréteur se fait au moyen de la touche *ESC* ; si l'on est en mode programme, l'instruction suivante est exécutée.

4) SEND imm & dif

send <{ [*expression numérique*] *expression chaîne* | *expression* } <{ , / ; <{ *expression numérique*] *expression chaîne* | *expression* }> }> > < , | ; >

L'instruction *SEND* est similaire à l'instruction *PRINT*, excepté que le résultat n'est pas visualisé mais transmis via le modem de la carte Apple-Tell.

La syntaxe de l'instruction *SEND* est identique à celle de l'instruction *PRINT*.

A noter qu'en mode Vidéotex l'instruction *SEND* (lorsqu'elle n'est pas terminée par le caractère ";") envoie les codes linefeed et retour chariot, alors qu'en mode 300 bauds seul le retour chariot est envoyé.

En mode Vidéotex, la commande *SEND [x] expression chaîne* génère les codes de répétition particuliers à ce mode de transmission lorsque les conditions suivantes sont réunies :

- la longueur de l'expression chaîne est égale à 1
- *x* est inférieur ou égal à 62

Si l'une de ces conditions n'est pas vérifiée, la chaîne de caractères est émise *x* fois.

exemples de commandes autorisées :

- | | |
|-------------------------------------|--|
| • send "hello" | envoie la chaîne hello suivie d'un retour chariot |
| • send "hello"; | identique, mais sans retour chariot |
| • send [9]"-";"hello";[9]"-" | envoie la chaîne ----- hello ----- |
| • send A,B | envoie les nombres A et B séparés par un retour chariot (c'est la seule différence avec l'instruction <i>PRINT</i>) |
| • send sin(1) | envoie la chaîne 0.841470985 . |

Note : l'instruction *SPEED* permet de modifier le débit des caractères envoyés.

5) *SEND USING* imm & dif

send using expression chaîne ; expression numérique <{ , | ; < expression numérique }> > < ; >

Identique à l'instruction *PRINT USING*, à cette différence près que l'argument est envoyé vers la ligne téléphonique.

exemple :

- **send using "sin(1)=#.####";sin(1)** envoie la chaîne "sin(1) = .8415"

6) RECEIVE dif

receive < [*expression chaîne*] > < / > *variable chaîne*

Cette instruction est similaire à l'instruction *INPUT* ou plutôt *LINPUT*, mais orientée télécommunications (le résultat provenant de l'autre terminal en ligne).

Seules les variables de type chaîne sont permises. L'emploi d'une variable numérique provoque le message *TYPE MISMATCH ERROR*.

L'utilisation de *RECEIVE* en mode direct donne le message *ILLEGAL DIRECT ERROR*.

L'instruction *RECEIVE* ["envoi"] A\$ attend l'arrivée d'une chaîne de caractères terminée par le séparateur "envoi" qui est une *expression chaîne*, la chaîne étant ensuite placée dans la variable A\$. Si le séparateur n'est pas de type chaîne, le message *TYPE MISMATCH ERROR* est émis. Seuls les 255 premiers caractères sont acceptés. La valeur par défaut du séparateur est fonction du mode de communication, il est égal :

- au retour chariot (\$8D) en mode 300 bauds ou Vidéotex
- au codage des commandes émises par le terminal minitel en mode serveur Vidéotex :

\$13 \$41 touche Envoi
\$13 \$42 touche Retour
\$13 \$43 touche Répétition
\$13 \$44 touche Guide
\$13 \$46 touche Sommaire
\$13 \$48 touche Suite.

Après réception d'une chaîne de caractères en mode Vidéotex (donc terminée par l'une des commandes ci-dessus), il est possible de connaître la touche appuyée en lisant le contenu de l'adresse \$F9, qui est égal au second code envoyé par l'appui de la touche.

exemple : si la touche Retour a été appuyée, le contenu de l'adresse mémoire \$F9 = 249 est égal à \$42 = 66. On peut donc, à partir d'un programme BASIC, décider de l'action à effectuer en exécutant la suite d'instructions :

RE = 66 : receive A\$: if peek (249) = RE then

L'option "/" permet de ne pas effacer le ou les caractères reçus avant le début de l'instruction *RECEIVE*.

Certains codes sont filtrés lors de la réception des caractères :

- le code \$8A (Line feed) est ignoré
- en mode 300 bauds, le caractère BackSpace (\$88) permet l'effacement du dernier caractère frappé
- en mode Vidéotex, les codes \$13 \$45 et \$13 \$47 (correspondant aux touches Annulation et Correction) permettent respectivement l'effacement de tous les caractères frappés, et l'effacement du dernier caractère entré à partir du terminal Minitel.

exemple : • **receive A\$** attend l'arrivée d'une chaîne de caractères terminée par un retour chariot, et la met dans la variable A\$.

7) GET LINE dif

get line < / > variable chaîne

Permet de capturer un caractère provenant d'un terminal ou d'un système serveur; ce caractère est mis dans la variable mentionnée.

Contrairement à l'instruction *Receive*, tous les caractères sont acceptés (il n'y a pas attente d'un séparateur).

Etant donné qu'en mode Vidéotex deux codes sont envoyés lors de l'appui d'une touche de commande du terminal Minitel, le second code est stocké à l'adresse mémoire \$F9 = 249 afin de pouvoir identifier la touche appuyée [la variable chaîne sera telle que ASC(variable chaîne) = \$13 = 19].

Comme pour l'instruction *RECEIVE*, l'option "/" permet de ne pas effacer le caractère reçu précédemment.

exemple : • **get A\$** attend l'arrivée d'un caractère et le met dans la variable A\$.

8) SEND MEMORY imm & dif

send memory < / >, A expression numérique1 , L expression numérique2

Cette instruction permet d'envoyer tous les codes de la zone mémoire *expression numérique1* , *expression numérique1* + *expression numérique2* .

exemples :

- **send memory, A\$1000,L\$100** envoie tous les codes compris entre l'adresse mémoire \$1000 et l'adresse mémoire \$1100
- **send memory, A4096,L256** et **send memory,A\$1000,L\$100** sont équivalents.

Applications :

Envoi de pages Vidéotex en format Protex par exemple. Il suffit de charger en mémoire le fichier binaire correspondant, et d'utiliser la fonction *SEND MEMORY* :

```
bload E.EXEMPLE,A$7000  
send memory,A$7000,L peek($AA60)+peek($AA61)*256
```

les adresses \$AA60 et \$AA61 contenant la longueur du dernier fichier chargé par le DOS.

L'option / permet d'interrompre l'exécution de l'instruction *SEND MEMORY* si un caractère a été reçu, l'instruction suivante est alors exécutée. Cette option permet par exemple la possibilité d'interrompre l'affichage d'une page Vidéotex. Essayez l'exemple précédent avec l'option / .

9) TIME OUT imm & dif

time out = expression numérique

Cette instruction a pour but de définir le temps d'attente maximum lors de la réception de caractères au moyen de la commande *RECEIVE* ou *GET LINE*. Si ce laps de temps est dépassé, l'instruction suivante est exécutée.

Expression numérique doit être compris entre 0 et 255, sinon l'on obtient le message *ILLEGAL QUANTITY ERROR* .

L'instruction *RECEIVE* ou *GET LINE* une fois franchie, il est souvent intéressant de connaître la cause du franchissement :

- si le contenu de l'adresse mémoire \$F9 est égal à 1, la sortie de l'instruction *RECEIVE* ou *GET LINE* s'est faite par dépassement de temps.
- si le contenu de l'adresse mémoire \$F9 est égal à 2, la sortie de l'instruction *RECEIVE* ou *GET LINE* s'est faite par réception de 255 caractères ou plus (dépassement de capacité).
- si le contenu de l'adresse mémoire \$F9 est différent de 1 et de 2, la sortie des instructions *RECEIVE* et *GET LINE* ont été provoquées respectivement par la réception d'un séparateur et la réception d'un caractère.

Cette instruction est utile lors de l'écriture d'un système serveur, afin d'éviter que celui-ci ne reste bloqué en attente de réception de caractères. De même, cette commande peut être utilisée lors de consultations automatiques de systèmes serveurs.

D'une façon générale, en mode serveur, la sortie des instructions *GETLINE* et *RECEIVE* provoque le message-système *TIME OUT ERROR*. Cette disposition permet de contrôler efficacement, par de classiques *ONERR*, le raccrochage éventuel de la ligne en cas de mutisme prolongé du correspondant.

valeur de TIMEOUT	temps correspondant (environ)
0	0
1	3.5 secondes
2	15.5 secondes
..
n	12 (n-1) + 3.5 secondes
..

exemple : *timeout = 1* définit un temps d'attente maximum de 3.5 secondes.

10) ECHO imm & dif

echo = expression numérique

Permet de modifier sous communication le mode d'écho : local ou distant.

Expression numérique doit être compris entre 0 et 1, dans le cas contraire on obtient le message *ILLEGAL QUANTITY ERROR*. S'il n'y a pas d'autre terminal en ligne, on obtient le message *NO COMMUNICATION ERROR*.

En mode serveur :

Echo = 0, ne renvoie pas le caractère émis par l'autre terminal

Echo = 1, le renvoie. C'est le mode par défaut.

En mode non serveur :

N'a de sens que sous mode terminal.

Si Echo = 0, le caractère tapé au clavier de l'Apple II est affiché (Local echo).

Si Echo = 1, mode écho distant (Remote echo). C'est le mode par défaut.

11) HOME LINE imm & dif

home line

Envoie le code \$8C (form feed), qui efface l'écran du terminal en ligne.

12) LOWERCASE imm & dif

lowercase = variable numérique

Convertit en majuscules les caractères minuscules reçus par *RECEIVE* ou *GETLINE*.

- **Lowercase = 1** : aucune action (c'est le mode par défaut)
- **Lowercase = 0** : conversion autorisée.

13) RDCHR imm & dif

variable numérique = rdchr

Cette fonction permet de déterminer la réception ou la non-réception de caractères. Si aucun caractère n'a été reçu, la valeur retournée est inférieure à 128, dans le cas contraire elle est égale au code ASCII du dernier caractère reçu.

Note : tenir compte du fait que la carte Apple-Tell ne mémorise pas uniquement le dernier caractère reçu mais 3 caractères au total.

14) FILTER\$ imm & dif

Filter\$ = expression chaîne

Permet de filtrer tous les caractères de *expression chaîne* lors de l'utilisation de l'instruction *RECEIVE*. Par défaut, seul le code LineFeed est filtré : *Filter\$* = chr\$(10).

exemple :

- **filter\$ = chr\$(10) + chr\$(7)** les codes Linefeed et Bell sont ignorés.

INSTRUCTIONS LIEES A L'AFFICHAGE EN MODE VIDEOTEX

Important : ces instructions ne sont disponibles que si l'on dispose du modem Apple-Tell.

1) **CURSOR LINE** imm & dif

cursor line expression numérique1, expression numérique2

Définit la position du curseur sur un terminal Minitel.

Expression numérique1 et *expression numérique2* doivent vérifier les inégalités suivantes :

$$\begin{aligned} 1 &\leq \text{expression numérique1} \leq 40 \\ 0 &\leq \text{expression numérique2} \leq 24 \end{aligned}$$

sinon, le message *ILLEGAL QUANTITY ERROR* est affiché.

Remarque : seule la commande *CURSOR LINE x, 0* permet d'écrire dans la ligne de statut du terminal Minitel (à la coordonnée horizontale *x*).

exemple :

• *cursor line 10,10* positionne le curseur aux coordonnées 10, 10 sur le terminal Minitel en ligne.

2) **CHRSIZE** imm & dif

chrsiz = expression numérique

Définit la taille des caractères affichés sur le terminal Minitel en fonction de la valeur de *expression numérique* :

chrsiz = 0	taille standard
chrsiz = 1	double hauteur
chrsiz = 2	double largeur
chrsiz = 3	double grandeur

Expression numérique doit être compris entre 0 et 3, sinon l'on obtient le message *ILLEGAL QUANTITY ERROR*.

3) FCOLOR LINE imm & dif

***fcolor* = K | R | G | Y | B | M | C | W**

Définit la couleur de forme (foreground) des caractères affichés sur le terminal Minitel.

fcolor = K	caractères noirs
fcolor = R	caractères rouges
fcolor = G	caractères verts
fcolor = Y	caractères jaunes
fcolor = B	caractères bleus
fcolor = M	caractères magenta
fcolor = C	caractères cyan
fcolor = W	caractères blancs

4) BCOLOR LINE imm & dif

***bcolor* = K | R | G | Y | B | M | C | W**

Définit la couleur de fond (background) des caractères affichés sur le terminal minitel.

bcolor = K	caractères noirs
bcolor = R	caractères rouges
bcolor = G	caractères verts
bcolor = Y	caractères jaunes
bcolor = B	caractères bleus
bcolor = M	caractères magenta
bcolor = C	caractères cyan
bcolor = W	caractères blancs

Les attributs de couleur de fond étant définis par zone, la séquence de définition est suivie du code Espace (\$A0). Ces attributs ne sont définis que pour une ligne.

EXEMPLES

Au verso de la disquette TélÉBASIC, et afin de guider vos premiers pas dans l'écriture de programmes TélÉBASIC, vous trouverez quelques exemples. Nous vous conseillons de les étudier avec la plus grande attention. Ils peuvent constituer le coeur d'applications beaucoup plus importantes.

1) Exemple d'un serveur minimum

Le programme suivant réalise un serveur interactif minimum : il attend une sonnerie, et permet le contrôle d'accès au serveur moyennant un test sur un mot de passe. Si le mot de passe est incorrect, le serveur raccroche.

Si l'utilisateur raccroche, le serveur est relancé.

```
10 Onerr goto 100
20 communicate,a
30 send "Bonjour, veuillez entrer le mot de passe "; : receive a$
40 if a$ <> "TELEBASIC" then send "mot de passe incorrect" : hangup : run 20

50 send "ok"
60 end : rem c'est sur cette ligne qu'un service moins ...minimum démarrerait
.....
100 if peek (222) = 145 or peek (222) = 147 then run 20 : else end
```

La ligne 20 initialise le serveur en **attente d'appel** (mode 300 bauds). Lorsque la connexion est établie, l'exécution du programme est reprise.

La ligne 30 affiche sur le terminal la chaîne de caractères entre apostrophes et attend l'entrée du mot de passe.

La ligne 40 permet le contrôle du mot de passe: la ligne est raccrochée si le mot de passe est incorrect.

La ligne 100 permet de relancer le serveur en cas de **perte de porteuse** (code erreur 147 ou 145).

Pour faire fonctionner ce programme sur un terminal **Minitel**, il suffit de changer la ligne 20 par

20 communicate, va .

2) Exemple d'un serveur Vidéotex fonctionnant avec le modem Apple-Tell et utilisant les touches de fonctions du terminal Minitel

Le programme suivant constitue un petit serveur Vidéotex illustrant l'utilisation des touches de fonctions du terminal Minitel afin de créer un embryon d'arborescence. Il commence par afficher une page Vidéotex et attend une commande. Si l'utilisateur appuie sur la touche **Suite**, la page suivante est affichée; s'il appuie sur la touche **Retour**, il s'agit de la page précédente. L'appui sur la touche Connexion-Fin provoque un retour à la page initiale.

Pour utiliser ce programme, il est nécessaire de recopier au préalable sur votre disquette les pages devant être servies (les pages utilisées dans l'exemple suivant proviennent de la disquette de démonstration fournie avec la carte Apple-Tell).

1 REM exemple de serveur Vidéotex utilisant les touches de fonctions du terminal Minitel.

10 ONERR GOTO 1000 : REM pour contrôler la perte de communication.

20 COMMUNICATE, VA : TIMEOUT = 3 : REM mise du serveur en attente d'appel.

25 FOR K=1 TO 3 : X = RDCHR : NEXT : HIMEM: \$7000 : REM nettoyage de la pile de la carte et sauvegarde d'une zone mémoire pour charger les pages Vidéotex

30 RESTORE 2000 : READ N : DIM PG\$(N) : FOR K=1 TO N : READ PG\$(K):
NEXT : I = 1 : REM N = nombre de pages à servir, I = page consultée.

40 PRINT CHR\$(4)"BLOAD E. "PG\$(I)",A\$7000" : HOME LINE : SEND
MEMORY , A\$7000, PEEK(\$AA60) + 256 * PEEK(\$AA61)

50 GET LINE / A\$

60 IF PEEK(249) = 200 THEN I = I + 1 : GOTO 100 : REM touche Suite.

70 IF PEEK(249) = 194 THEN I = I - 1 : GOTO 100 : REM touche Retour.

100 IF I < 1 THEN I = 1

110 IF I > N THEN I = N

120 GOTO 40

1000 IF PEEK (222) = 145 or PEEK (222) = 147 then RUN 20 : REM le serveur est relancé si la communication est coupée.

1010 IF PEEK (222) = 150 then RUN 30 : REM touche Connexion-Fin

1020 IF PEEK (222) = 149 then HANGUP : RUN 20 : REM le serveur est relancé si aucune réponse n'est reçue après trente secondes

1030 STOP

2000 DATA 12

2010 DATA POMPE A EAU, POMEROL, METEO, COUPE POMPE, SPRINGTIME, AUSTRALIE

2020 DATA BOURSE, CAMBOUIS, CHAPLIN

2030 DATA GALOP, MIRE, EVERY DAY'S LIFE

3) Les autres programmes de démonstration

D'autres programmes vous sont proposés. Nous vous laissons le soin de les étudier et de comprendre leur architecture. Nous vous recommandons de les lister sur imprimante selon la procédure habituelle (PR£1, LIST si votre carte interface imprimante est dans le connecteur d'extension 1).

UTILISATION DE LA PAGE ZERO

T adresse utilisée par **Télébasic**

D adresse utilisée par le **DOS**

M adresse utilisée par le **Moniteur**

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T
10	T	T	T	T	T	T	T	T	T						T	MT
20	M	M	M	M	M	M	MD	MD	M	M	MD	MD	MD	MD	MD	MD
30	M	M	M	M	M	MD	MD	MD	MD	MD	M	M	M	M	MD	MD
40	MD	MD	MD	MD	MD	MD	MD	MD	MD	M	D	D	D	D	M	M
50	M	MT	MT	MT	MT	MT	T	T	T	T	T	T	T	T	T	T
60	T	T	T	T	T	T	T	TD	TD	TD	TD	T	T	T	T	TD
70	TD	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T
80	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T
90	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T
A0	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	TD
B0	TD	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T
C0	T	T	T	T	T	T	T	T	T	T	TD	TD	TD	TD	T	T
D0	T	T	T	T	T	T	T	T	T	TD	T	T	T	T	T	T
E0	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T
F0	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T

CONCESSION DE LICENCE DE LOGICIEL

TéléBASIC est un interpréteur BASIC qui vous permet de développer des applications de télécommunication variées et très puissantes. Il est ainsi parfaitement possible de créer un serveur complet. La variété de ses instructions en font également un des interpréteurs BASIC les plus évolués du marché de l'informatique personnelle (présence d'instructions comme IF...THEN...ELSE, PRINT USING, CURSOR, WINDOW, etc...).

TéléBASIC est distribué par le CLUB APPLE sous forme d'une disquette contenant l'interpréteur. Il se charge une fois pour toute dans la mémoire vive de l'unité centrale et est opérationnel tant que l'appareil reste sous tension. Si vous éteignez l'unité centrale, vous devrez donc réamorcer la disquette.

Les programmes que vous écrirez avec l'interpréteur TéléBASIC nécessiteront le chargement préalable de l'interpréteur.

Si vous désirez commercialiser une application développée avec l'interpréteur TéléBASIC, deux possibilités s'offrent à vous :

- distribuer une disquette contenant votre programme TéléBASIC en indiquant à votre utilisateur qu'il doit se procurer l'interpréteur séparément en s'adressant au CLUB APPLE,
- contacter Joël BERNARD via le CLUB APPLE pour obtenir une version spéciale de TéléBASIC lui permettant d'installer sur une même disquette son application et l'interpréteur. Ce programme est alors cédé sous forme d'une concession de licence de logiciel.

Pour tout renseignement complémentaire, les personnes intéressées sont invitées à écrire au :

CLUB APPLE
Concession de licence TéléBASIC
43, avenue de la Grande Armée
75016 PARIS

(
2
"

))
"

12



Club Apple - 43, avenue de la Grande-Armée - 75116 Paris