EXECUTION COMMANDE SEDORIC LINPUT

Saisie formatée (fenêtre) de texte dans une variable alphanumérique. Cette commande LINPUT, dont l'entrée est en EC94, analyse la syntaxe et les options demandées, fait appel au s/p XLINPU, qui est en fait la partie principale de LINPUT et qui altère en sortie les adresses suivantes:

- 12/13 adresse de la ligne du curseur TEXT.
- 1F/20 calcul de l'adresse de la ligne du curseur TEXT.
- 28 flag numérique/alphanumérique (#00 si numérique et #FF si chaîne).
- 91/92 adresse de la chaîne pointée par TXTPTR.
- B8/B9 et B6/B7 pointent sur le descripteur de chaîne.
- D0 longueur de la chaîne saisie.
- D1/D2 adresse de la chaîne saisie.
- F2 nombre de caractères à saisir.
- F3 indique quels paramètres ont été sélectionnés (E, S, C, J et K).
- F4 index lié au dernier paramètre: b3 pour E, b4 pour K, b5 pour J, b6 pour C et b7 pour S; puis, à la fin, contient le mode de sortie.
- F5 nombre de colonnes actives selon 026A (indicateur état console), puis à la fin constitue une réplique de F2.
- 0268 nE de la ligne du curseur (ordonnée y, de 1 à 27).
- 0269 et 02F8 le nE de la colonne du curseur (abscisse x, de 0 à 39).
- C075 caractère à utiliser pour remplir la fenêtre de saisie.

Enfin, la variable OM (Output Mode) contient le mode de sortie:
0 (si RETURN), 1 (si ESC), 2 à 5 (si flèches gauche, droite, bas et haut respectivement) et 6 (si sortie automatique).

Rappel de la syntaxe de LINPUT. Il s'agit en fait d'une fenêtre avec éditeur de type pleine page. Sont valides: CTRL/T (min/MAJ), CTRL/N (effacement ligne), CTRL/Z (ESC pour attributs vidéo), DEL, ESC (sortie), RETURN (sortie) et flèches (déplacement et sortie).

LINPUT(@x,y,)(car,)long;VA(,E)(,S)(,C)(,J)(,K)

@x,y, permet d'effectuer la saisie à la position xy de l'écran (en théorie, car cette fonction est boguée, ne pas l'utiliser, voir plus loin).

car, en fait 1er caractère d'une expression alphanumérique. Sert à matérialiser la place des caractères demandés ("." par défaut). Est conservé d'un LINPUT à l'autre (tant que pas modifié). Après modification, pour revenir au ".", il faut re-indiquer "." ou faire RESET!

long nombre de caractères à saisir (1 à 255), paramètre obligatoire.

;VA variable alphanumérique pour chaîne saisie, paramètre obligatoire.

Si aucun des 5 paramètres qui suivent n'est indiqué, le curseur, arrivé en fin de fenêtre, rebouclera au début de la fenêtre. Pour sortir, il faudra utiliser les flèches, ESC ou RETURN.

,E Le manuel indique "permet de ne pas effacer la fenêtre avant l'entrée du texte". En fait, ce paramètre permet de ne pas afficher le caractère de remplissage ("matérialisation" de la fenêtre, qui est toujours effectuée par défaut). NB: On n'efface vraiment la fenêtre qu'en indiquant " " comme caractère de remplissage.

,S Contrairement à ce qui est indiqué dans le manuel (**bogue**), interdit de sortir avec les flèches de déplacement (mode par défaut).

,C sortie automatique lorsque le curseur atteint la fin de la fenêtre. Sinon, par défaut, le curseur revient au début de la fenêtre.

,K "justifie" le texte entré à l'écran (mais pas la variable) en remplaçant les caractères de remplissage par des espaces.

,J inversement, "justifie la variable", sans affecter l'affichage. La combinaison ,J,K permet de "justifier" l'écran et la variable.

La bogue de LINPUT

LINPUT présente un grave problème de gestion du curseur. Après un LINPUT, le positionnement du curseur est complètement faussé, rendant impossible l'utilisation de certaines autres commandes, par exemple un PRINT@ ou un autre LINPUT@. Ceci apparait aussi lorsque la longueur de la chaîne demandée dépasse 38 caractères. Les facéties du curseur sont quasi imprévisibles et rendent impossible l'utilisation, à coup sûr, du paramètre "@x,y,". Ceci vient du fait qu'il ne suffit pas d'initaliser correctement les variables impliquées dans la gestion du curseur (comme le fait très bien LINPUT), il faut les valider par un affichage réel.

Voici la solution la plus simple, valable non seulement pour LINPUT, mais aussi dans tous les cas où il y a un problème de ce type. Elle consiste à repositionner soit-même le curseur avec un PRINT@x,y;CHR$(18); (ne pas oublier le ";"). CHR$(18) correspond à CTRL/R et n'est pas utilisé par ORIC (il faut se rappeler **R** comme **R**epositionner).

Il fallait, en effet, trouver quelque chose à afficher qui ne modifie pas l'affichage, l'idéal étant PRINT@x,y;"" qui hélas ne valide pas les variables impliquées dans la gestion du curseur. Par chance, le PRINT@x,y;CHR$(18); effectue tout ce qu'il faut: après consultation de la table des codes de contrôles en F5E2/ROM, il ne fait rien de spécial, cette fonction n'étant pas affectée, et continue comme s'il avait affiché quelque chose!

Voici un petit exemple de ce qu'il faut faire:

```
10 PRINT@10,10;CHR$(18); :' Positionne le curseur au début de la fenêtre
20 LINPUT 40;A$          :' Pour saisir 40 caractères à la position 10,10
30 PRINT@10,20;CHR$(18); :' Repositionne le curseur pour le PRINT@ suivant
40 PRINT@10,20;A$        :' Seul moyen d'afficher avec PRINT@ après LINPUT
```

Début de l'analyse de la commande LINPUT

EC94- AA TAX 1er caract après commande LINPUT (paramètre)
EC95- AD 6A 02 LDA 026A empile mode console Oric-1/Atmos à l'entrée
EC98- 48 PHA notamment affichage curseur qui sera changé
EC99- E0 C6 CPX #C6 1er car est-il "@"? préfixe de coordonnées xy
EC9B- D0 1E BNE ECBB si non, saute la gestion des coordonnées xy

Gestion des coordonnées xy

EC9D- 20 98 D3 JSR D398 XCRGET: Saisit dans A le caractère suivant. Ceci afin de positionner convenablement TXTPTR pour la routine DA22/ROM.
ECA0- 20 40 D7 JSR D740 "CURSEUR OFF" (curseur caché = vidéo normale)
ECA3- 20 92 D2 JSR D292 JSR DA22/ROM Prend 2 coordonnées à TXTPTR. Au retour, X contient le nE de ligne (y), 02F8 contient le nE de colonne (x) et 1F/20 contient l'adresse LLHH de la ligne (A contient aussi LL de adr)
ECA6- A4 20 LDY 20 HH poids fort de l'adresse de la ligne cible
ECA8- 85 12 STA 12 copie en 12/13 l'adresse de la ligne où
ECAA- 84 13 STY 13 devra agir LINPUT pour la saisie formatée
ECAC- 8E 68 02 STX 0268 copie ordonnée y en 0268 (nE de ligne cible)
ECAF- AE F8 02 LDX 02F8 copie abscisse x en 0269 (nE colonne cible)
ECB2- 8E 69 02 STX 0269
ECB5- 20 3E D7 JSR D73E "CURSEUR ON" (curseur visible = vidéo inverse)
ECB8- 20 2C D2 JSR D22C JSR D067/ROM demande "," lit car suiv, conv MAJ

Gestion du caractère de remplissage

ECBB- 20 24 D2 JSR D224 JSR CF17/ROM Evaluer une expression à TXTPTR, retourne avec valeur numérique dans ACC1 ou adresse de chaîne dans D3/D4
ECBE- 24 28 BIT 28 valeur numérique si #00 dans 28, chaîne si #FF
ECC0- 10 15 BPL ECD7 si numérique, saute le s/p de gestion du caract à afficher et continue au s/p d'évaluation du nombre de caract à saisir. Si aucune chaîne n'est spécifiée, le programme poursuit en EDC7 et le caractère de remplissage sera celui présent en C075 ("." lors du boot).
ECC2- 20 77 D2 JSR D277 JSR D7D0/ROM (longueur de la chaîne dans A avec Z selon cette longueur et adresse de la chaîne dans XY et 91/92)
ECC5- F0 05 BEQ ECCC branche en ECCC si longueur de la chaîne nulle
ECC7- A0 00 LDY #00 Y = premier caractère de cette chaîne qui
ECC9- B1 91 LDA (91),Y permettra de remplir ultérieurement la fenêtre
ECCB- 2C A9 2E BIT 2EA9 et continue en ECCE
ECCC- A9 2E LDA #2E sinon A = "." (pris par défaut)
ECCE- 8D 75 C0 STA C075 le caractère de remplissage A sera gardé en C075 d'un LINPUT à l'autre, tant que pas de nouvelle chaîne spécifiée
ECD1- 20 2C D2 JSR D22C JSR D067/ROM (demander une ",")
ECD4- 20 24 D2 JSR D224 JSR CF17/ROM Evalue expression à TXTPTR, retour avec val num dans ACC1 (#00 ds 28) ou adr chaîne dans D3/D4 (#FF dans 28)

Evaluation du nombre de caractères à saisir

ECD7- 20 19 D2 JSR D219 Vérifie que l'expression évaluée à TXTPTR est bien numérique. Retourne avec la valeur dans ACC1 ou "TYPE MISMATCH ERROR". Le paramètre "nombre de caractères à saisir" est obligatoire.
ECDA- 20 82 D2 JSR D282 JSR D8CB/ROM Prend un entier dans ACC1 et le retourne dans X (nombre de caractères à saisir).
ECDD- 8A TXA teste ce nombre de caractères (Z = 1 si nul)
ECDE- F0 4B BEQ ED2B si X = 0, erreur nE9 "ILLEGAL QUANTITY"
ECE0- 86 F2 STX F2 sauve le nombre de caractères à saisir dans F2

Demande la variable alphanumérique requise

ECE2- A9 3B LDA #3B A = ";" (marqueur situé devant la variable)
ECE4- 20 2E D2 JSR D22E JSR D067/ROM Demande ";" à TXTPTR, lit le caractère suivant en continuant au s/p CHARGET en E2 (qui saute les espaces), puis à l'interpréteur Sédoric en 0400, puis au s/p ECB9/ROM et finalement convertit ce caract en MAJUSCULE (s/p D3A1/RAMOV). Au cours de ce périple, le sort de Y est assez indéfini, mais semble finir avec Y = 0.
ECE7- 84 F3 STY F3 supposons que Y = #00 (sauf contre indication!)
ECE9- 20 2E ED JSR ED2E prendre adresse de la variable à TXTPTR (B8/B9)
ECEC- 20 1B D2 JSR D21B SEC & JSR CF09/ROM Vérifier si alphanumérique
ECEF- 20 9E D3 JSR D39E XCRGOT Lit dans A le car courant et conv en MAJ
ECF2- F0 25 BEQ ED19 Si nul, fin des paramètres, continue à XLINPU

Analyse des paramètres de fin de commande, lorsqu'ils existent

ECF4- 20 2C D2 JSR D22C Si pas nul, il y a encore des paramètres.
JSR D067/ROM demande une "," lit le caract suiv et le convertit en MAJ
ECF7- 20 A1 D3 JSR D3A1 re-convertit en MAJUSCULE (pour être bien sûr!)
ECFA- A2 04 LDX #04 X = 4 pour lire 5 paramètres dans une table
ECFC- 86 F4 STX F4 sauve cet index dans F4 (seul bit b2 est à 1)
ECFE- 06 F4 ASL F4 décale vers la gauche le bit qui est à 1
ED00- DD BA CD CMP CDBA,X cherche si le paramètre visé est l'une de ces 5 lettres suivantes: **E** (en CDBE avec X=#04), **K** (en CDBD avec X=#04), **J** (en CDBC avec X=#04), **C** (en CDBB avec X=#04) ou **S** (en CDBA avec X=#04).

128	64	32	16	8	4	2	1
0	0	0	0	0	1	0	0
7	6	5	4	3	2	1	0
S	C	J	K	E			

ED03- F0 05 BEQ ED0A si oui, branche en ED0A
ED05- CA DEX si non, indexe la lettre précédente dans table
ED06- 10 F6 BPL ECFE et reboucle en ECFE (suite de la recherche)
ED08- 30 1E BMI ED28 "SYNTAX ERROR": le paramètre situé après la virgule n'est pas l'un de ceux qui sont autorisés par la syntaxe de LINPUT
ED0A- A5 F4 LDA F4 porte le flag du paramètre trouvé:
b3 à 1 si **E**, b4 à 1 si **K**, b5 à 1 si **J**, b6 à 1 si **C** ou b7 à 1 si **S**
ED0C- 45 F3 EOR F3 A = F3 plus copie du bit qui est à 1 dans F4 (5 paramètres et 5 places de b3 à b7), mais "efface" tout bit qui serait déjà à 1 dans F3 (doublon dans commande), ce qui entraîne comme résultat A < F3
ED0E- C5 F3 CMP F3 compare le résultat avec F3
ED10- 90 16 BCC ED28 si A < F3 branche vers "SYNTAX ERROR", ce qui se produit si un paramètre figure en double dans la commande!
ED12- 85 F3 STA F3 sinon, sauve A dans F3 qui garde donc trace de tous les paramètres qui ont été trouvés et sera utilisé par XLINPU.
ED14- 20 98 D3 JSR D398 XCRGET: Lit dans A le caract suiv, conv en MAJ
ED17- D0 DB BNE ECF4 reboucle s'il reste des paramètres, sinon...
ED19- 20 36 ED JSR ED36 XLINPU: Appel de la routine LINPUT proprem dite
ED1C- 20 8E EE JSR EE8E au retour, met la longueur et l'adresse de la chaîne dans la variable alphanumérique indiquée dans la ligne de commande
ED1F- 68 PLA
ED20- 8D 6A 02 STA 026A restaure le mode console Oric-1/Atmos initial
ED23- A5 F4 LDA F4 copie nE du mode de sortie
ED25- 4C D8 D7 JMP D7D8 dans la variable OM et retourne.

ED28- 4C 23 DE JMP DE23 simple relai vers "SYNTAX ERROR"
ED2B- 4C 20 DE JMP DE20 simple relai vers erreur nE9 "ILLEGAL QUANTITY"

Prend l'adresse de la valeur de la variable à TXTPTR

ED2E- 20 38 D2 JSR D238 JSR D188/ROM Place cette adr dans AY et D3/D4
ED31- 85 B8 STA B8 et la copie aussi dans B8/B9
ED33- 84 B9 STY B9
ED35- 60 RTS

André Chéramy, 54, rue de Sours, 28000 CHARTRES et Yann Legrand, 608, rue de l'Eglise, "Les Templiers", 62610 LANDRETHUN LES ADRES
à suivre...