

# RÉALISER UN MICRO-ORDINATEUR "HAUT DE GAMME" C'EST TRÈS SIMPLE : AUJOURD'HUI, LA TECHNOLOGIE LE PERMET

# Vegas 6809

## VII. LE "X BASIC"

**Si divers langages sont disponibles en option pour Vegas (Forth, Fortran, Cobol...), le Basic fait partie intégrante de la version de base du système (il ne pouvait, évidemment, en être autrement). Il s'agit d'ailleurs d'un Basic particulièrement évolué, très rapide et complet, connu sous le nom de XBasic. Le XBasic comprend les commandes classiques de ce langage. Il est, de ce fait, compatible avec ceux régulièrement diffusés. Cette version est particulièrement adaptée aux applications de gestion et aux calculs scientifiques, c'est-à-dire à des domaines où une grande précision est requise : XBasic met en effet en œuvre 16 chiffres significatifs...**

**Toutes ces particularités impliquent un interpréteur de 19 Ko ! Bien que cette taille mémoire paraisse importante, que le lecteur ne soit pas inquiet : Vegas, rappelons-le, dispose d'une mémoire vive de 48 Ko.**

**A**u cours de cet article qui se veut être une description détaillée du Basic de Vegas, nous supposons le lecteur déjà familiarisé avec les notions élémentaires de programmation.

Si tel n'était pas le cas, nous ne saurions trop lui recommander la lecture de la série d'initiation au Basic, actuellement développée dans la revue *Telesoft*.

Notre propos se limitera donc à une description des différentes commandes, instructions et fonctions intrinsèques du XBasic.

À ce sujet, savez-vous comment taper un programme Basic et l'exécuter tandis que Vegas est dans sa phase d'initialisation, ne pouvant alors comprendre et exécuter que des commandes FLEX (décrites dans notre précédent numéro) ? La réponse à cette question est très simple : il suffit, tout simplement, à partir du FLEX (les symboles +++ apparaissant à l'écran) de taper au clavier l'ordre XBasic suivi d'un appui sur la touche « retour chariot ». L'interpréteur Basic est dès lors chargé en mémoire vive et l'écran de Vegas affiche : PRET (à exécuter les commandes du XBasic).

### Les commandes du XBasic

Une commande, à la différence d'une instruction, n'apparaît pas dans un programme. Le XBasic est pourvu des commandes suivantes :

**CLEAR** : est utilisée pour mettre à zéro toutes les variables d'un programme.

**COMPILE** : entraîne la sauvegarde d'un programme sous une forme condensée, c'est-à-dire, prenant le moins de place possible en langage Basic (pour les informaticiens, ce mot est traître, car cette commande ne transforme pas le programme en code machine).

**CONT** (continue) : permet de poursuivre l'exécution d'un programme après qu'il ait été interrompu par ordre STOP ou CTRL C.

**EXIT** : provoque le retour sous moniteur.

**FLEX** : permet de revenir au VFlex lorsque le système est sous Basic.

**LIST** : permet d'afficher à l'écran soit la totalité du pro-

## UNE ETUDE ASSISTEE PAR MICRO-SYSTEMES

# VISMO

Vente Informations Services Micro-Ordinateurs

22 Bd de Reuilly, 75012 PARIS

Tél. (1) 628.28.00

VENTE ET DEMONSTRATION DE 14 H à 21 H SAUF LUNDI

Métros : Daumesnil ou Dugommier

**VISMO EXPRESS :** Livraison dans toute la France. Nous encaissons vos chèques à l'expédition de votre commande, jamais à la réception de vos ordres.

## SPÉCIAL NOËL...

### VISMO... C'EST UN SERVICE DE QUALITÉ

DÉMONSTRATION PERMANENTE PAR DE VÉRITABLES INFORMATIENS  
TOUT MATÉRIEL TESTÉ DEVANT VOUS AVANT ACHAT  
ÉCHANGE IMMÉDIAT DU MATÉRIEL SOUS GARANTIE

**NOUVEAU CHEZ VISMO**  
Augmentez la taille de votre ZX-81 avec :

- le boîtier VISMO (forme Apple)
- le clavier mécanique (VISMO)
- l'inverseur TV/Video (VISMO)
- le moniteur Zénith 12p. écran vert (se pose sur le boîtier)
- l'interface imprimante ZP 82

**CRÉDIT  
GRATUIT**  
Nous consulter

**L'ENSEMBLE  
AVEC LE ZX**

2.500 F  
3.000 F

COMPAREZ LES PRIX...  
VOUS JUGEREZ LA QUALITÉ SUR L'EFFICACITÉ DU SERVICE

Demandez les produits Vismo chez votre revendeur habituel.

### BON DE COMMANDE

à retourner à Vismo, 68 rue Albert 75013 Paris

Nom : \_\_\_\_\_ Prénom : \_\_\_\_\_

Adresse : \_\_\_\_\_

Code Postal : \_\_\_\_\_ Ville : \_\_\_\_\_

Date : \_\_\_\_\_ Tél. : \_\_\_\_\_

Signature : \_\_\_\_\_

Désire recevoir catalogue  
20 F (remboursable à la 1<sup>re</sup> commande)

Quantité	Désignation	Prix unit. TTC	Prix total TTC

MS

MODE DE REGLEMENT	
Chèque bancaire joint	<input type="checkbox"/>
CCP joint	<input type="checkbox"/>
Mandat-lettre joint	<input type="checkbox"/>
Contre-remboursement	<input type="checkbox"/>

Participation frais de port et d'emballage + 30 F
Port gratuit pour + de 3000 F d'achat sauf Sernam.
Contre-remboursement + 30 F.

A la différence d'une instruction, une commande n'apparaît jamais dans un programme.

gramme (LIST), soit une ligne (LIST n), soit encore une série de lignes (LIST u,v).

**LOAD** : cette commande charge un fichier de la disquette en mémoire vive.

**NEW** : permet d'effacer le programme résident en mémoire vive. Un nouveau programme Basic peut alors prendre sa place.

**RUN** : provoque l'exécution du programme stocké en mémoire vive. Dans le cas d'un programme compilé, il faut utiliser la commande : RUN « PROGRAM » (le nom du programme figure entre guillemets).

**SAVE** : permet de stocker un programme sur disquette.

**SCALE** : indique au Basic le nombre de chiffres qu'il faut conserver à droite de la virgule (maximum 6).

**TRON** : met en place la fonction TRACE utilisée pour la recherche des erreurs existant éventuellement dans les programmes.

**TROFF** : supprime la fonction TRACE.

**+** : permet de revenir momentanément comme nous l'avons signalé. Ces commandes, prises en compte dès leur frappe sur le clavier, ne sont pas utilisables à l'intérieur des programmes Basic qui sont constitués d'instructions.

### Les instructions du XBasic

Différents types d'instructions sont rencontrés. Ainsi, il existe les assignations, les transferts, les branchements, les entrées/sorties, les boucles, les ordres de fin d'exécution et quelques instructions diverses.

#### Les assignations du XBasic

**DATA** : contient des données qui seront lues, par la suite, au cours du programme.

**LET** : assigne une valeur à une variable. Cette instruction

est facultative (LET C = 0 est identique à C = 0).

**READ** : permet de lire des données à partir d'une instruction DATA.

**RESTORE** : remet à zéro les données figurant au niveau des lignes DATA.

### Les instructions de transfert de programmes

Ces instructions permettent le passage d'un programme (ou sous-programme) à un autre.

**GOSUB x** : transfère le contrôle au sous-programme spécifié à la ligne x.

**GOTO x** : permet le branchement du programme à la ligne x.

**ON <EXPRESSION> GOSUB x,y,z...** : permet d'appeler un sous-programme parmi plusieurs. En effet, selon la valeur de <expression> le sous-programme appelé sera celui débutant à la ligne x, y, z...

**ON <EXPRESSION> GO TO x,y,z** : est identique à l'instruction précédente, mais pour un branchement à une ligne.

**ON ERROR GOTO** : permet à l'utilisateur de constituer ses propres messages d'erreurs en imposant, par exemple, un branchement à une ligne comportant une instruction PRINT.

**RESUME** : redonne le contrôle au programme principal après qu'une routine d'erreur ait été exécutée.

**RETURN** indique à Vegas la fin du sous-programme en cours d'exécution et provoque un branchement à la ligne du programme principal suivant l'ordre GOSUB.

### Les branchements conditionnels

Dans cette catégorie, nous rencontrons l'ensemble des instructions provoquant un branchement dépendant d'une condition.

**IF <CONDITION> GOTO x** : selon le résultat de la condition, le programme saute à

la ligne x (condition vérifiée) ou poursuit son déroulement séquentiel (condition non vérifiée).

**IF <CONDITION> THEN <EXPRESSION>** : est identique à l'instruction précédente, mais sans obligation de branchement à une ligne. Dans ce cas, si la condition est vérifiée, l'expression qui suit THEN est prise en compte.

**IF <CONDITION> THEN <EXPRESSION 1> ELSE <EXPRESSION 2>** : est identique à la précédente, mais, si la condition n'est pas vérifiée, ce qui suit le mot ELSE est pris en compte.

### Les ordres d'entrées-sorties

Comme leur nom l'indique, ces instructions gèrent les échanges de données entre périphériques et mémoire centrale.

**INPUT** : permet d'affecter une valeur à une variable.

**INPUT LINE** : est utilisée afin d'affecter une ligne complète à une variable de type « chaîne ».

**PRINT** : permet d'afficher à l'écran une variable ou une chaîne de caractères.

**PRINT USING** : permet de choisir le format de l'écriture. Par exemple :

10 PRINT USING "# # # # . # #" indiquera la présentation d'un nombre à deux décimales. Différentes possibilités sont offertes par cette instruction très performante telles que :

- la création d'un champ d'un seul caractère alphanumérique (signalé par le signe !),

- la définition d'un champ de 2 caractères ou plus (signalé par le signe //)

- la définition du nombre de chiffres dans un nombre (signe #). Ceci est très appréciable au niveau des applications de gestion pour, par exemple, aligner les points décimaux (virgules) dans un tableau de chiffres.

## Boucles et fin de programme

Les boucles de programme sont obtenues de manière classique grâce à l'instruction FOR... NEXT.

**FOR** : permet l'exécution de toutes les instructions qui le suivent jusqu'à ce que le mot NEXT soit atteint.

Les instructions de fin de programme provoquent une interruption de l'exécution du programme, définitive ou provisoire.

**END** : interrompt purement et simplement l'exécution d'un programme.

**STOP** : est identique à la précédente, mais un message d'interruption apparaît à l'écran. L'exécution du programme peut être reprise grâce à l'instruction **CONT** (CONTInue).

## Les ordres divers

**DEF** : permet de définir des fonctions simples (par exemple  $DEF FN(x) = 3x + 3$ ).

**DIM** : réserve l'espace mémoire pour les variables matricielles.

**POKE** et **DPOKE** : permettent l'écriture des données à des emplacements mémoire spécifiés.

**DPEEK** et **PEEK** : sont des fonctions opposées aux ordres Basic **DPOKE** et **POKE**.

**REM** : permet de placer remarques et commentaires dans un programme. Ceux-ci ne sont pas pris en compte par l'interpréteur.

**DIGITS** : assure la définition du nombre de chiffres, après la virgule que devra posséder tout nombre.

**SNAP** : échange les valeurs de deux variables entre elles.

## Les fonctions intrinsèques du XBasic

Les fonctions intrinsèques du XBasic sont constituées des fonctions mathématiques et trigonométriques, des opérateurs agissant sur les chaînes de caractères et de certaines fonctions d'entrées-sorties.

Les fonctions mathématiques, calculées sur 16 chiffres, proposées par le XBasic sont les suivantes : l'exponentielle : **EXP(X)**, le logarithme naturel : **LOG(X)**, et la fonction racine carrée **SQR(X)**.

Les fonctions trigonométriques ont une précision dépendant de la valeur de l'argument et, en général, de 13 chiffres.

Les fonctions arctangentes : **ATN(X)**, cosinus **COS(X)**, sinus **SIN(X)** et tangente **TAN(X)** sont incluses dans le XBasic.

Ensuite, on rencontre les instructions agissant sur les chaînes de caractères : la fonction **ASC(XS)** permet d'obtenir la valeur numérique ASCII du 1<sup>er</sup> caractère d'une chaîne de caractères baptisée **XS**.

La fonction **CHR\$(I)** : fournit le caractère ASCII de I (si I est inférieur à 127).

La fonction **HEX** convertit une chaîne de caractères hexadécimale en sa valeur décimale.

La fonction **INCH\$(I %)** est très spécifique au XBasic : elle permet d'attendre et de prendre en compte un caractère frappé au clavier.

**INSTR(I %, SS, PS)** : permet la recherche d'une chaîne de caractères **PS** à l'intérieur de la chaîne **SS**, à partir du 1<sup>er</sup> caractère.

**LEFT\$(XS, I %)** : sélectionne les I caractères à gauche de **XS**.

**LEN(XS)** : compte le nombre de caractères contenus dans la chaîne **XS**.

**MID\$(XS, I %, J)** : sélectionnent une sous-chaîne de caractères dans la chaîne **XS**.

**RIGHT\$(AS, I)** : reconnaît les I caractères les plus à droite de la chaîne **XS**.

**STR\$(X)** : engendre une chaîne de caractères qui représente l'expression numérique de X.

**VAL(XS)** : est la fonction inverse de la précédente.

**POS(I %)** : détermine la position de la colonne du canal I.

**SPC(I %)** : permet un espacement de I caractères lors de l'édition.

**TAB(I)** : cette fonction déplace le curseur de I positions horizontales sur l'écran, ou la tête de l'imprimante.

## Fonctions diverses

**ABS(X)** : donne la valeur absolue de X.

**INT(X)** : restitue la partie entière de X.

**PI** : donne la valeur de « pi » = 3,1415926535897933.

**PTR <nom de la variable>** : fournit l'adresse de la variable donnée en argument.

**RND(X)** : génère un nombre aléatoire dont la valeur est comprise entre 0 et 1.

**SGN(X)** : est la fonction signe ( $SGN(X) = 1$  si  $X > 0$ ,  $SGN(X) = 0$  si  $X \leq 0$ ). ■

N. HUTIN D. HABERT

## Comment réaliser Vegas 6809 ?

L'ensemble des éléments nécessaires à la construction de Vegas :

- kit de base (carte « mère » avec ses composants, lecteur de disquettes, clavier Qwerty, système d'exploitation Flex et XBasic) ;

- le circuit imprimé ;

- les composants ;

- le (ou les) lecteur(s) de disquettes ;

- le clavier...

est disponible par correspondance chez son concepteur, Microkit, B.P. 46, 91302 Massy Cedex. Tél. : (6) 013.39.21.

Vous pouvez également voir Vegas chez :

- **SOS Computer**, 78, rue de Dunkerque, 75009 Paris. Tél. : (1) 281.03.73.

- **Vismo**, 22, boulevard de Reuilly, 75012 Paris. Tél. : 628.28.00.

- **Lens Buro**, 73, boulevard Basly, 42300 Lens. Tél. : (21) 28.39.43.

Vegas est une marque déposée 3D International, 2, rue de l'Armée-Patton, 91640 Briis-sous-Forges. Tél. : (1) 594.61.36.