

# pom's

ÉDITEUR : ÉDITIONS  
ÉDITEUR : ÉDITIONS

## Sommaire

Editorial		3
Overlay dynamique	<i>Guy Mathias</i>	9
<b>VISICALC et ARPLESOFT</b>	<i>Bruno Rives</i>	17
<b>Un programme aide-mémoire</b>	<i>Gérard Michel</i>	23
Des instructions en une lettre	<i>Christian Guérin</i>	29
Graphiques : de l'ITT 2020 à l'Apple	<i>Luis Nobre</i>	32
Les adresses du Graphique	<i>Jean-François Duvivier</i>	35
<b>P.L.E. : Le Program Line Editor</b>	<i>Hervé Thiriez</i>	41
<b>C.R.A.E. : Co-Resident Applesoft Editor</b>	<i>Hervé Thiriez</i>	46
Des vers dans la pomme...	<i>Alexandre Duback</i>	48
Inverseur DOS 3.2 - DOS 3.3	<i>Jean-François Duvivier</i>	52
Déplacement du programme en assembleur	<i>Jean-François Duvivier</i>	55
Changez votre poignée de jeu	<i>Gilles Mauffrey</i>	59
Programmer en Pascal?	<i>Christian Colmant</i>	62
Trucs, astuces et informations		8-47-54-65-66

143, Av. Félix-Faure 75015 Paris - Tél. : 544.97.48

86, Bd Magenta 75010 Paris - Tél. : 201.94.68



# EDITORIAL

Le numéro un de Pom's est la concrétisation d'un rêve vieux de plus d'un an. Comme beaucoup d'utilisateurs d'Apple, j'ai souvent souffert du manque d'information. J'ai fréquemment eu l'impression de réinventer ce que d'autres avaient déjà fait plus tôt et mieux que moi !

En plusieurs mois de préparation, une équipe regroupant des spécialistes français de matériels Apple parmi les meilleurs a été constituée. Cette équipe est entièrement disposée à accueillir ceux d'entre vous qui désirent soumettre leurs contributions. Une analyse approfondie des autres publications s'adressant aux utilisateurs d'Apple a été réalisée, afin de ne perdre aucune occasion de vous transmettre des informations utiles.

Nous vous proposons ainsi une revue de 64 pages dont la publication est pour le moment prévue trimestriellement. Si la demande se révèle suffisamment forte, nous passerons ultérieurement à une publication bimestrielle.

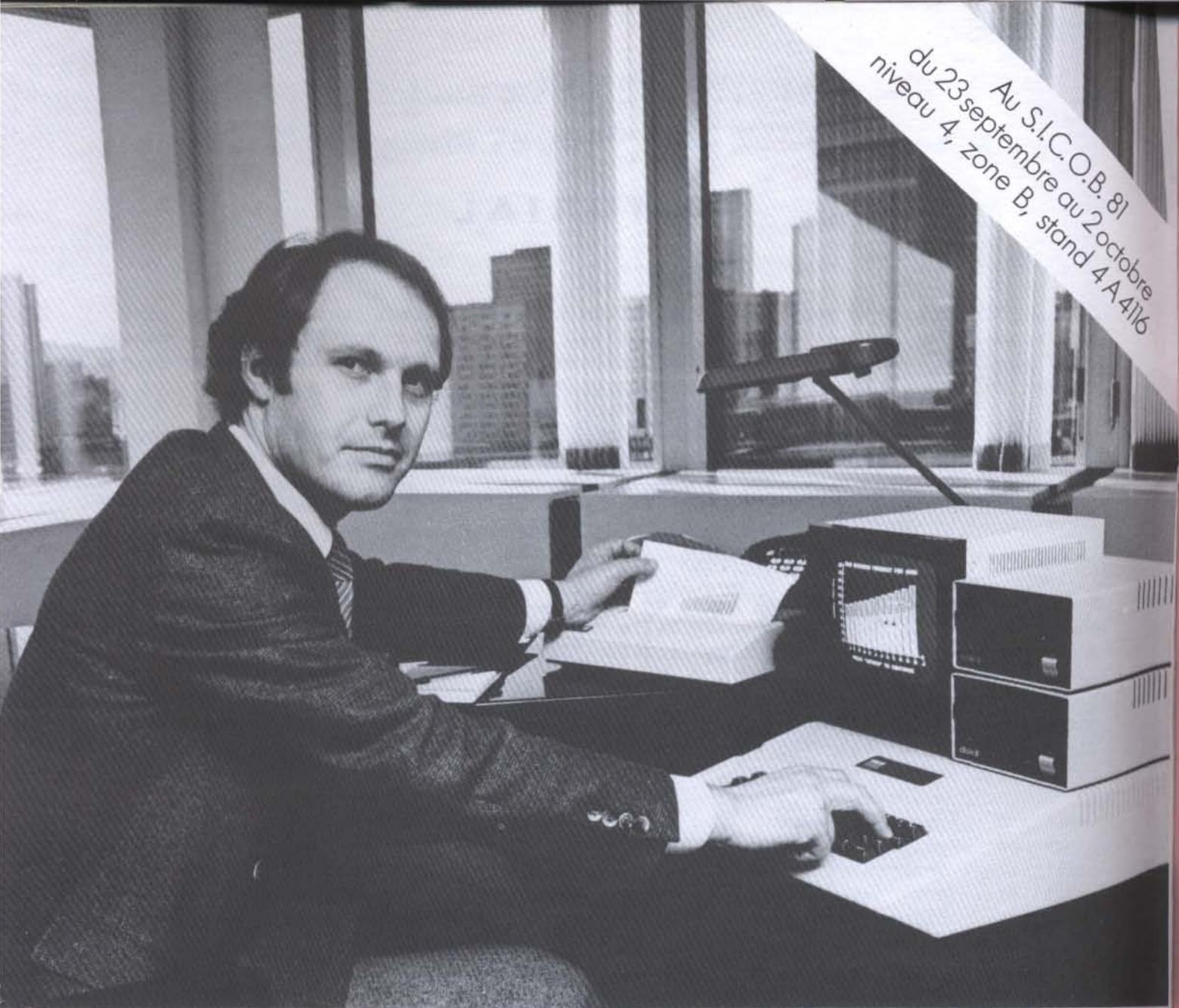
L'objectif de cette revue est de rendre service à tous les utilisateurs. A ce titre, nous essaierons de vous offrir régulièrement des articles de niveaux variés, et sur une grande diversité de sujets. Dans un souci de qualité, tous les articles ont été testés avant impression.

Certaines rubriques donneront lieu à des articles dans chaque numéro. Ainsi, vous trouverez régulièrement des articles dans chacune des rubriques suivantes : utilitaires, analyse ou modification de matériel, bugs, programmes d'application ou progiciels, jeux ou applications graphiques. Dans le prochain numéro, nous mettrons en place les rubriques suivantes : problèmes à résoudre, communication des clubs et courrier des lecteurs.

Hervé Thiriez

*Directeur de la publication - rédacteur en chef : Hervé Thiriez - Siège social : Editions MEV - 49, rue Lamartine - 78000 Versailles - Tél. (3) 955.14.14 - Rédaction et abonnements : 59, bd de Glatigny - 78000 Versailles - Tél. (3) 950.48.99 ou 918.13.07 - Régie publicitaire : Forcé 7 - 41, rue de la Grange-aux-Belles - 75483 Paris Cedex 10 - Tél. (1) 238.66.10 - Diffusion auprès des boutiques informatiques et libraires : Editions du PSI - 41-51, rue Jacquard - BP 86 - 77400 Lagny - Tél. (6) 007.59.31. Ont collaboré à ce numéro : Christian Colmant, Alexandre Duback, Jean-François Duvivier, Christian Guérin, Guy Mathias, Gilles Mauffrey, Gérard Michel, Luis Nobre, Bruno Rives, Hervé Thiriez.*

Au S.I.C.O.B. 81  
du 23 septembre au 2 octobre  
niveau 4, zone B, stand 4A416



## L'ORDINATEUR PERSONNEL DE M<sup>R</sup> LE DIRECTEUR DES VENTES.

Apple, c'est d'abord un nouveau concept. Celui de la relation individuelle et directe entre l'homme et sa machine. Un concept qui ouvre une troisième voie entre rien du tout et l'informatique de millionnaire. Apple, c'est un outil personnel aussi facile à utiliser qu'une voiture, aussi léger et peu encombrant qu'une simple machine à écrire et aussi peu coûteux qu'une photocopieuse. Un outil qui vous simplifie la vie en diminuant la paperasse qui vous encombre, efficace et rapide dès qu'une décision s'impose.


Apple vous aide à répondre à des questions de type "qu'arriverait-il si ?", traite vos problèmes de gestion (planification, prévisions, budget) et met en forme et imprime vos lettres et vos rapports. Apple est un outil versatile qui fonctionne sur une vaste gamme de programmes que vous soyez avocat ou médecin, responsable des ventes ou du marketing, comptable ou financier.

Intéressé, un peu sceptique ou dubitatif... Lorsque votre revendeur Apple local aura résolu devant vous quelques-uns de vos problèmes, ou vous aura fait utiliser vous-même un programme de planification ou une analyse sous forme de graphique couleur, vous aurez compris l'intérêt d'avoir à portée de la main la puissance de l'ordinateur personnel.

Et vous commencerez à penser à votre Apple.

Pour obtenir l'adresse de votre revendeur local Apple, contactez :

- Sonotec : 41-45, rue Galilée, 75116 Paris, France, Tél. : 723.78.56; Bell Telephone Mfg. Co. Microcomputer Division, Luitenant Lippenslaan 44, 2200 Borgerhout, Belgique - Téléphone : 32-31 - 35.75.72; Industrade AG Thur-gauerstrasse 72, 8050 Zürich, Suisse.

 **apple computer**



**29 rue de Clichy 75009 Paris**

**64, Av. du Prado - 13008 MARSEILLE** Tél. 37.25.03

## Garantie FLASH

Il y a garantie et garantie. Avant d'acquérir une machine chez un distributeur, assurez-vous d'abord si celui-ci dépanne lui-même en ses ateliers, ou bien s'il renvoie ledit matériel à l'importateur. Bien sur, le résultat sera le même, mais non pas le délai.

Pour raccourcir ce délai, et sur certains produits sélectionnés, INTERNATIONAL COMPUTER a décidé de pratiquer une GARANTIE FLASH. Cette garantie, gratuite les 6 premiers mois, permet la réparation ou l'échange de la machine défectueuse, DANS UN DELAI MAXIMAL DE 9 HEURES OUVRABLES.

Cette garantie est valable sur les produits marqués d'un label "GARANTIE FLASH" dans nos publicités.

**BELFORT** 7, r. Commandos-d'Afrique - Cravanche Tél. 28.23.29

REPRISE OU DEPOT - VENTE DE VOTRE ANCIEN ORDINATEUR  
DEMONSTRATION A DOMICILE SUR RV - EXPEDITIONS RAPIDES SUR  
TOUTE LA FRANCE ET LA PLUPART DES PAYS - DETAXE A L'EXPORTATION  
POUR RESIDENTS ETRANGERS - CONTRATS D'ENTRETIEN-REPARATIONS

Microordinateurs - Miniordinateurs - Interfaces - Floppies - Disques durs - Disques souples - Imprimantes - Moniteurs - TV - Téléprojecteurs pour conférences - Meubles pour ordinateurs - Supports magnétiques - Cassettes - Papier - Librairie - Programmes Composants - Terminaux - Consoles de visualisation - Systèmes "Clé en main"  
OCCASIONS - MATERIELS DE DEMONSTRATION - MATERIELS LOGICIELS

**NOUS SOMMES HEUREUX  
D'ACCEPTER  
LES BONS DE COMMANDE  
DE L'ADMINISTRATION**

**CLUBS, COLLECTIVITES  
COMITES D'ENTREPRISES  
ADMINISTRATIONS  
UNIVERSITES**  
CONTACTEZ  
NOTRE DEPARTEMENT  
COLLECTIVITES

## LA "HOT LINE" I.C. ou COMMENT ACHETER MOINS CHER

Grand de la distribution microinformatique, INTERNATIONAL COMPUTER souhaite également être le moins cher.

S'il arrivait que dans les pages de votre magazine préféré, vous trouviez une publicité présentant le ou les produits que vous désirez acquérir, moins cher que chez nous, alors appelez sans tarder la HOT LINE IC (285.24.55, lignes groupées) et nous nous efforcerons d'être encore moins cher si c'est un produit que nous commercialisons.

Pourquoi ceci ? Parce que la recherche du meilleur prix d'achat peut nous amener à pouvoir modifier A LA BAISSSE nos prix de vente et à en faire ainsi profiter nos clients.

### CONDITIONS DE VENTE PAR CORRESPONDANCE

- 1) Le matériel est expédié en porteur pour les imprimantes et la librairie France au dessus de 200 F TTC
- 2) Pour un paiement comptant vous joignez à votre bon de commande le règlement spinal du paiement de votre achat. Il vous sera alors adressé votre facture par retour du courrier.
- 3) Pour un paiement à crédit joignez à votre bon de commande 20% du montant total de votre achat plus 30 F pour les frais de dossier de crédit. Nous vous renverrons alors un dossier de crédit que vous nous renverrez rempli et signé.
- 4) Pour un leasing, spécifiez nous votre commune. Nous vous enverrons un dossier que vous nous renverrez rempli et signé.
- 5) Pour une demande de documentation joignez 3 F en timbres.

Bon de Commande à renvoyer à INTERNATIONAL COMPUTER, 29, rue de Clichy - 75009 PARIS

Je, soussigné M \_\_\_\_\_ Prénom \_\_\_\_\_ Adresse \_\_\_\_\_

Code Postal \_\_\_\_\_ Ville \_\_\_\_\_ Tél. (bur.) \_\_\_\_\_ (dom.) \_\_\_\_\_

commande le matériel suivant : microordinateur \_\_\_\_\_

périphériques \_\_\_\_\_ accessoires \_\_\_\_\_

librairie \_\_\_\_\_ programmes \_\_\_\_\_

TOTAL T.T.C. \_\_\_\_\_

Ci-joint la somme de \_\_\_\_\_ F

en chèque bancaire  CCP

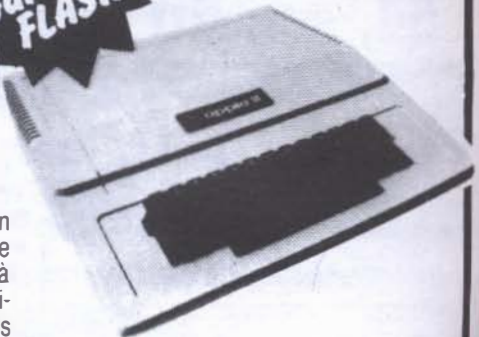
Date \_\_\_\_\_ Signature \_\_\_\_\_



## Apple II<sup>TM</sup> PLUS

**Garantie FLASH**

### L'UNITÉ CENTRALE



Depuis les utilisations industrielles et académiques (comptabilité de processus, acquisition de données, etc.) en passant par la gestion jusqu'aux applications domestiques, peu de choses échappent à l'APPLE. Sa conception robuste mais très sophistiquée peut être qualifiée d'ouverte. Par exemple, si l'on branche sur un Apple à disque 2 imprimantes, 1 magnétophone, 1 crayon optique, il vous restera encore 4 connecteurs disponibles. C'est donc, par sa "expandabilité", un appareil indémodable et qui pourra toujours s'adapter aux techniques nouvelles. La preuve, son langage PASCAL est le même que celui installé sur des machines bien plus importantes en INTERNATIONAL COMPUTER peut vous proposer en option un disque dur de 10 mega-Octets (10 000 000 de caractères).

**CARACTÉRISTIQUES DE L'APPLE**  
 Sa carrosserie en matière synthétique moulée est d'une bonne apparence et est pratiquement indestructible. Son clavier "QWERTY" est à dire amical, est d'un toucher particulièrement agréable sans rebonds intempestifs, quant à sa habitude... un des programmes médicaux que nous distribuons à nécessite l'entrée de plus de 3 000 000 de caractères sur le même APPLE et il s'en porte toujours très bien.

**PROCESSEUR**: c'est un 6802 avec une horloge à 1 MHz. C'est l'un des microprocesseurs les plus puissants actuellement. Grâce notamment à la richesse de ses modes d'adressage.

**MÉMOIRES**: Unité en 16, 32 ou 48 K octets de mémoire RAM ou vive si peut, grâce à la carte PASCAL, être portée à 64 K (1 K octet = 1000 octets = 8000 bits). Transformer un Apple 16 K en 48 est une opération extrêmement simple qui ne nécessite même pas un tournevis.

**LANGUAGES**: L'APPLE II reçoit en série le basic APPLESOFT<sup>®</sup> interprète II possède 9 chiffres significatifs, ON ERR GOTO, etc. Au sujet des chiffres significatifs, une remarque est importante: certains autres basics peuvent avoir, en mode double précision sur un 16 chiffres. Mais cette double précision n'est disponible que pour les opérations arithmétiques. C'est à dire que si l'on calcule un SIN ou un LOG, il n'y aura que 6 à 7 chiffres, alors que l'APPLE vous donnera imperméablement ses 9 chiffres.

**CARACTÉRISTIQUES GÉNÉRALES**  
**AFFICHAGE**: 40 caractères par ligne, 24 lignes, caractères normaux, inverses, clignotants.

**COULEURS**: 15 en basse résolution (40 H x 48 V), 6 en haute résolution (280 x 192). Pour apprécier l'efficacité de la haute résolution APPLE, nous vous recommandons les programmes SARGON II (check) ou SUPER INVADER (bataille contre les monstres).

**SON**: haut-parleur incorporé.  
 En résumé, l'APPLE est vraiment le grand classique du microordinateur car depuis le 16 K relié à une TV et un magnétocassette, jusqu'au 64 K Pascal, relié à un terminal, une imprimante rapide et des disques durs de 10 MOctets, c'est toujours la même unité centrale, le même APPLE.

### Un APPLE II avec les caractères français à l'écran!!!

Nous proposons pour APPLE une option "Traitement de Texte" qui se compose de deux cartes hard, permettant de visualiser à l'écran non seulement les majuscules/minuscules, obtenues par Shift, mais aussi les caractères français auxquels des touches ont été attribuées. Une modification de l'imprimante est nécessaire.

**3900 F HT                      4586 F TTC**

16 K	<b>7195 F HT</b>	8461.32 TTC
32 K	<b>7595 F HT</b>	8931.72 TTC
48 K	<b>7995 F HT</b>	9402.12 TTC

### ENSEMBLES BUDGET

1 Apple 16 K	1 Apple 32 K	1 Apple 32 K	1 Apple 48 K
1 modulateur NB	1 Floppy DOS 3.3	1 Floppy DOS 3.3	1 Floppy DOS 3.3
1 lecteur de K7	1 Modulateur NB	1 moniteur NB IC	1 moniteur NB IC
<b>7600 F TTC</b>	1 Carte CLUB	1 Carte CLUB	1 imprimante
1 Apple 16 K	<b>12495 F TTC</b>	<b>13905 F TTC</b>	Seikosha GP 80
1 moniteur NB IC			1 Interface pour imprimante
<b>8400 F TTC</b>			1 Carte CLUB
			<b>16495 F TTC</b>

## LES PERIPHERIQUES SPECIFIQUES APPLE II

### MINIDISK II

#### DISK II

C'est le complément idéal de votre APPLE. Vous pourrez ainsi accéder à la manipulation de fichiers, charger tous vos programmes en quelques secondes, faire des copies, etc... Il se manipule avec des séries d'instruction ajoutées au basic, telles que LOAD, SAVE, OPEN, WRITE, RENAME, etc... L'accès peut être séquentiel ou direct, vous pourrez ainsi chaîner vos programmes, les renumérotter, et profiter vraiment de toutes les possibilités de votre APPLE.

La documentation américaine fournie est copieuse (178 pages), mais il n'est pas nécessaire de la posséder à fond pour commencer à se servir du disque. Capacité d'une disquette = 116 K octets. Directement alimentée par APPLE (jusqu'à 14 drives). Possibilité d'utilisation en langage machine. Temps d'accès moyen = 200 MS. Vitesse de transfert des données = 156 K-bits par seconde.



**AVEC CONTROLEUR DOS 3.3**  
**3895 F HT**  
**4580,52 F TTC**  
**SANS CONTROLEUR**  
**2895 F HT**  
**3404,52 F TTC**

### PASCAL LANGUAGE CARD

#### LE PASCAL APPLE II

C'est un langage très puissant, et qui est compilé, et non totalement interprété comme pour le basic. Il y a donc un gain de temps et de mémoire. Les problèmes que peut traiter PASCAL, dans les mêmes conditions de mémoire et de temps sont 5 à 10 fois plus importants que ceux que peut traiter basic. C'est un langage structuré qui rend les programmes modulaires (variables locales) quand un élément du programme est défectueux, on le change sans toucher au reste du programme. PASCAL APPLE II est un véritable logiciel de base, comprenant un langage, un système de traitement de fichiers, un système de traitement de texte. Avec PASCAL, APPLE II devient un véritable mini système informatique à la portée de tous. En outre, les possibilités de votre APPLE sont augmentées, puisque sa mémoire RAM passe à 64 K, la capacité de sa disquette à 148 K et permet de travailler avec des consoles de visualisation de 80 caractères par ligne.

**2795 F HT**  
 (il faut 48 K et 1 disk II)  
**3286,92 F TTC**



## LES AUTRES PRODUITS APPLE

Cartes 80 colonnes VEEDEX	2500 F HT	Moniteur 12" SANYO vert	1950 F HT
Interface parallèle pour imprimante	1195 F HT	Music System Mountain Hardware (16 voies)	3300 F HT
Interface série pour imprimante	1195 F HT	Alf Music Synthesizer	2000 F HT
Carte de communication RS 232, 110/300 Bauds	1195 F HT	Micro Music	1200 F HT
Langage FORTRAN	1195 F HT	Graphic input tablet	4150 F HT
CARTE HORLOGE CCS	1195 F HT	Table traçante A3 WATANABE Miplot	9500 F HT
CARTE Z80 MICROSOFT	2395 F HT	Carte logique/analogique	950 F HT
KIT DOS 3.3	570 F HT	Carte analyseur d'images vidéo Digisector	2850 F HT
EXTENSION 16 K RAM en KIT	400 F HT	Clavier numérique	790 F HT
CARTE SECAM	940 F HT	Disque CORVUS 9,5 M octets	28500 F HT
CARTE RVB avec prise Pérétélevision	940 F HT	Floppy MEGASTOR 2x512 K 8"	20620 F HT
Moniteur Couleur Thomson et RVB	3600 F HT	Programme APPLEWRITER	450 F HT
Moniteur Vidéo 100 N/B	1190 F HT	Programme APPLEPOST	290 F HT
Moniteur Vidéo 100 vert	1365 F HT	Programme APPLEPLOT	350 F HT
Moniteur 9" NEC professionnel	1650 F HT		

POUR TOUTE CONFIGURATION  
**APPLE**  
 +  
**PERIPHERIQUE**  
**1400,00 F HT**  
 (1646,40 F TTC)  
**24 mois de garantie totale**  
 comprenant : pièces main d'œuvre et ...  
**2 révisions générales**



## Faites patienter durant FRE(0)

Que vous le demandiez ou non, votre Apple doit de temps en temps réorganiser sa mémoire et éliminer les chaînes alphanumériques.

Il vaut mieux le demander vous-même à un endroit de passage fréquent, par exemple le menu principal. Vous pouvez alors envoyer un message à l'utilisateur pour l'aider à prendre son mal en patience.

La ligne suivante satisfait cet objectif :

```
1000 IF PEEK(112) - PEEK(110) < 4  
THEN PRINT "PATIENCE ..." :  
X=FRE(0)
```

Dans un prochain numéro, nous vous montrerons encore mieux, comment réellement accélérer ce processus de réorganisation. C'est plus satisfaisant que de faire attendre l'utilisateur. Mais en attendant ...

# E.A.O.

Enseignement Assisté par Ordinateur pour Ordinateurs  
MOPPE D'ANFICO - SILEX - ITT 2020 - APPLE II - APPLE II +

COURS EN FRANÇAIS

de B A S I C	<ul style="list-style-type: none"><li>• APPLIESOFT - APPLE II ( + Carte APPLIESOFT )</li><li>• APPLE II +</li><li>• APPLE III</li><li>• MOPPE D'ANFICO ( + APPLIESOFT en ROM )</li><li>• PALSOFIT - ITT 2020 ( + Carte PALSOFIT )</li><li>• SILROM - SILEX ( de Léanord )</li></ul>
--------------	---

( Sur DISQUETTES 5 1/4 Pouches - DOS = DOS 3.3 - Mémoire = 48 K. )

**1/ COURS 1 ( BASIC ) - En Français**

- TRES PROGRESSIF - Ne nécessite aucune connaissance préalable en informatique.
- Four débutants et non débutants. TOUT Y EST EXPLIQUÉ.
- 20 Leçons - Environ 10 à 12 Heures de cours.
- 80 Exercices commentés, expliqués, résolus, exécutés.
- 140 Questions notées sur 20, par groupes. - Réponses aux questions.
- GRAPHISME BASSE et HAUTE RESOLUTION.
- Défilement automatique du Cours avec arrêts et reprises possibles en cours de leçon.
- De nombreux exercices peuvent être réexécutés autant de fois que vous le souhaitez. Ainsi, vous pouvez obtenir les mêmes résultats ou des résultats différents en faisant varier les données d'entrée.

A LA FIN DE CE COURS, VOUS SAUREZ PROGRAMMER.

**2/ COURS 2 ( BASIC + ) - En Français**

- Philosophie générale identique à celle du COURS 1 ( BASIC ).
- 25 Leçons - 12 Heures de cours minimum.
- 120 Exercices commentés, expliqués, résolus, exécutés.
- 160 Questions notées sur 20, par groupes - Réponses aux questions.

**3/ CONTRATS-LOCATION du COURS BASIC pour:**

- Etablissements d'Enseignement.
- Etablissements de Formation payante.
- Centre de Recherches, Laboratoires, Centre d'Essais,....
- Comité d'Entreprises.

<p>PRIX ( T.V.A. comprise )</p> <ul style="list-style-type: none"><li>• COURS 1 ( BASIC ) 690 FF.</li><li>• COURS 2 ( BASIC + ) 890 FF.</li></ul> <p>Prix donnés à titre indicatif, pouvant être modifiés sans préavis.</p>	<p style="text-align: center;">ANDRÉ F. FINOT 8 Allée BUFFON 91000 EVRY-COURCOURONNES</p>
---	---

R E V E N D E U R S ,  
C O N T A C T E Z - N ' O U S .

# Logma

- Vous savez programmer en Basic
- Vous voulez tirer le maximum de votre Apple
- Vous voulez gagner du temps dans la création et la mise au point de vos programmes

Inscrivez-vous au séminaire :

## "Plus et mieux avec l'Apple"

à La-Celle-St-Cloud,

les 22 et 23 octobre 1981 ou les 9 et 10 décembre 1981

- PROGRAMME :
- l'optimisation de programmes
  - la structuration de programmes
  - les structures d'overlay
  - l'exploitation des utilitaires.

Travaux en petits groupes - Frais d'inscription 3000 F H.T.

Une informatique de gestion adaptée aux besoins des gestionnaires et réalisée par des gestionnaires.

- Etudes d'application
- Conception de logiciel spécialisé
- Installation et formation
- Harmonisation avec les méthodes de gestion.

**Logma** Centre La Châtaigneraie - 29, avenue de Versailles - 78190 La-Celle-St-Cloud - Tél. : 918.13.07



# Overlay Dynamique

Guy Mathias

## Overlay dynamique : principes généraux.

Dans un souci de clarté, cet article est écrit de façon très détaillée. Les experts sont priés, de faire preuve de tolérance si certaines explications leur semblent superflues.

Un overlay est une technique consistant à découper un programme en morceaux de façon à ne jamais garder plus de deux morceaux dans la mémoire en même temps. Cette technique permet d'éviter de stocker en mémoire des programmes trop volumineux tout en donnant à chaque morceau, ou module, un rôle propre.

Deux approches sont possibles lors de la conception d'un overlay :

- . Soit un seul module réside en mémoire à un moment donné. Quand il le faut, il se fait remplacer par un autre module.
- . Soit un module appelé noyau réside en permanence dans la mémoire. Il gère les appels des autres modules, chacun d'entre eux cohabitant à un moment donné avec le noyau.

Nous avons choisi la seconde solution ; en effet, les programmeurs expérimentés bâtissent des programmes utilisant abondamment des sous-programmes d'intérêt général. Ceux-ci étant nécessaires à la plupart des modules, il est préférable de les regrouper dans le noyau afin d'éviter le déplacement de modules de taille importante.

L'objectif de l'overlay est double : permettre des traitements importants, et autoriser des chaînages beaucoup plus rapides que ceux proposés par le DOS : chargement du programme CHAIN, déplacement des variables, chargement du programme appelé et raccordement aux variables.

Analysons en détail la construction d'un programme utilisant l'overlay dynamique, en prenant l'exemple d'un programme de gestion des stocks, MASTOCK. Sous forme monolithique, la structure du programme se présenterait comme nous pouvons le voir dans la page suivante.

Elle comporterait douze blocs qui sont effectivement ceux ayant servi à la conception d'un progiciel de gestion des stocks. Bien entendu, la structure d'overlay s'applique à tout programme de taille importante, particulièrement quand il existe un bloc de sous-programmes communs conséquent.

1	DIMENSIONS
2	LECTURE DES PARAMETRES DE FONCTIONNEMENT
3	CHARGEMENT DES DONNEES (DATA)
4	MENU PRINCIPAL
5	ZONE DE SOUS-PROGRAMMES COMMUNS
6	MISE A JOUR DU FICHER ARTICLES
7	GESTION DES COMMANDES ET DES FOURNISSEURS
8	FACTURATION
9	ETATS COMMERCIAUX
10	INVENTAIRES
11	ETATS DIVERS
12	DONNEES (DATA)

Fort probablement, les diverses parties font appel aux mêmes sous-programmes et aux mêmes données. Formons donc le noyau avec ces éléments communs; on obtient ainsi par exemple :

Noyau : formé des parties 4 et 5  
 Module 0 : parties 1, 2, 3 et 12  
 Module 1 : partie 6  
 Module 2 : partie 7  
 ..... etc.

Le lecteur remarquera que le module 0 doit être exécuté en premier car il réalise l'initialisation. Il faut en outre le concevoir de façon à ce qu'il soit le plus long de tous les modules, afin que les données ne soient pas détruites en cours d'exécution par le chargement d'un module plus long.

On forme ainsi un premier programme constitué par le noyau et MOD0. Il faut maintenant créer un système qui remplace, à la demande du noyau, le module 0 par le module correspondant au choix de la fonction demandée par l'utilisateur. Pour atteindre cet objectif, écrivons un programme assembleur qui sache repérer MOD0 et le remplacer par le module choisi.

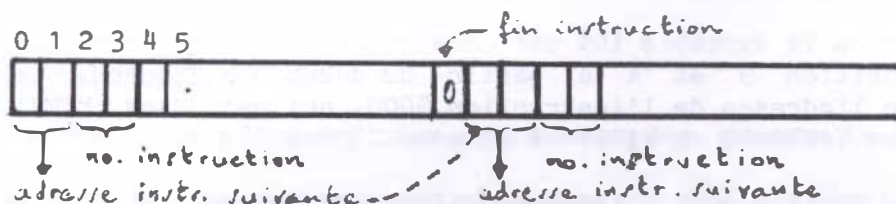
#### Description du programme OVERL.

Ce programme se décompose en deux parties distinctes: le chargement du module demandé, et le recadrage des adresses de début et fin du programme.

Nous avons choisi le numéro 5000 comme premier numéro d'instruction de chaque module. Le noyau est donc formé d'instructions étiquettées entre 1 et 4999. Pour charger le nouveau module, il faut d'abord rechercher en assembleur l'instruction 5000.

Afin de comprendre le principe de cette recherche, il est préférable de fournir quelques explications sur la représentation en mémoire d'un programme BASIC. L'adresse du début du programme est contenue dans les mémoires \$67 et \$68. A

partir de cette adresse, les contenus mémoire s'expliquent comme suit :



La partie du programme assembleur correspondant à cette recherche est la séquence comprise entre les lignes 0280 et 1000.

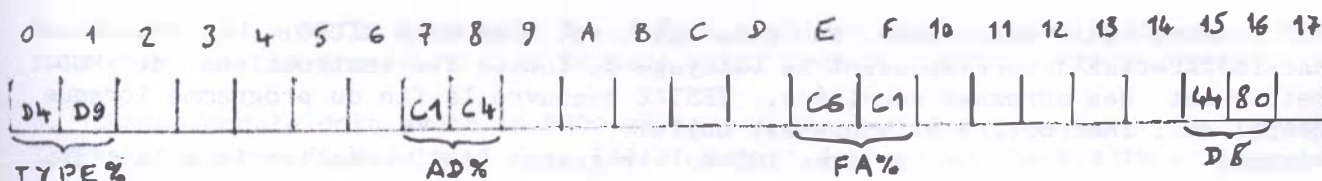
Commentaires :

- le sous-programme RECHLI met l'adresse début de programme dans MEMVI
- lignes 0380-0440 : le numéro de l'instruction correspond-il à \$13 (poids fort de 5000 en hexadécimal) ?
  - . si c'est le cas, voir si le poids faible du numéro d'instruction correspond à la valeur \$88 recherchée : lignes 0600 à 0660
  - . dans le cas contraire, prendre l'adresse de l'instruction suivante pour la mettre dans MEMVI : lignes 0460 à 0580
- une fois l'instruction 5000 trouvée, la séquence TROUV! est abordée (0700). Cette séquence place dans la variable AD% du programme BASIC la valeur hexadécimale de l'adresse de cette instruction.

En effet, l'interpréteur BASIC range les variables simples, ici TYPE% et AD%, au fur et à mesure de leur utilisation première dans le programme. Chaque variable occupe 7 octets, leur contenu étant expliqué dans le prochain paragraphe. A titre d'exemple, nous reproduisons ici le programme BASIC, suivi de l'analyse mémoire des variables.

```

1  REM  PROGRAMME MASTOCK ,AUTEUR GUY MATHIAS
5  HIMEM: 36550
10 TYPE% = 0:AD% = 0:FA% = 0:DB% = CHR$(4)
20 PRINT D$"LOAD OVERL"
30 GOSUB 5000
40 IF TYPE% = 0 THEN 5030: REM  EXECUTION MODULE 0
50 REM  B% CONTIENT LA FONCTION SOLHAITEE A PARTIR DU MENU
60 ON B% GOTO 70,80,90: REM  ETC...
70 IF TYPE% < > 1 THEN A$ = "MOD1": GOSUB 100:TYPE% = 1: GOTO 95
80 IF TYPE% < > 2 THEN A$ = "MOD2": GOSUB 100:TYPE% = 2: GOTO 95
90 REM  ... ET AINSI DE SUITE.
95 FA% = 0: ON B% GOTO 5030,5030,5030
100 CALL 36560: PRINT D$"LOAD"A$,A$AD%: CALL 36563: RETURN : REM  3656
    0=BED0
5000 REM  MODULE 0
5010 TYPE% = 0: RETURN
5020 REM  SUITE MODULE 0
    
```



Les variables entières sont archivées en deux octets, les trois suivants étant à

zéro. Les variables réelles prennent cinq octets pour leur représentation en flottant. Les variables chaînes ont leur longueur dans le premier octet, leur adresse dans les deux suivants, et les trois derniers octets à zéro.

Dans notre exemple, la valeur de la variable AD% est constituée par les contenus des mémoires rangées en position 9 et A à partir du début des données. Le programme assembleur met donc l'adresse de l'instruction 5000, qui est dans MEMVI et MEMVI+01, dans les mémoires VARIA+09 et VARIA+0A (lignes 0700 à 0800).

La séquence de 0840 à 0980, inutile pour l'instant, calcule la longueur du module qui était en mémoire (MODO, dans le cas présent), et la met en FA%.

Connaissant maintenant l'adresse de début, on peut charger le module, ce qui est réalisé par l'instruction 100 du programme BASIC. Dans celle-ci, CALL 36560 calcule l'adresse de l'instruction 5000 ; A\$ contient le nom du module demandé.

Notre nouveau module, MOD1 par exemple, a maintenant remplacé MODO. Tout n'est pas terminé, car il faut calculer la nouvelle adresse fin de programme et, surtout, recalculer dans le module que l'on a chargé toutes les adresses des instructions.

Expliquons-en la raison : quand MOD1 a été archivé, il était associé à un noyau qui n'était pas forcément le même que celui résidant en ce moment en mémoire. La moindre modification d'un programme BASIC décale tout son contenu en mémoire.

Quand un module est chargé par BLOAD, le DOS sait où ce module se trouvait et quelle était sa longueur au moment où il avait été sauvegardé sur disquette. La routine assembleur CADRAG recherche ces informations et, compte tenu de la nouvelle adresse début de programme (AD%), recalcule toutes les adresses de MOD1.

Commentaires sur CADRAG :

- 1160 : remise à zéro de la zone de travail AD600
- 1200 : on cherche dans le DOS la longueur de MOD1 et on l'ajoute à celle de l'instruction 5000 (MEMVI). Le résultat est la nouvelle adresse fin de programme, ADRFIN.
- 1320 : début du calcul des nouvelles adresses. Rappel : chaque instruction commence par l'adresse de la suivante, puis le numéro, et se termine par 00.
- 1340-1420 : recherche de la position réelle de l'instruction suivant la 5000 dans MOD1.
- 1460 : la position relative de cette adresse par rapport au début de programme est placée en AD600.
- 1500-1780 : calcul de la différence entre l'adresse réelle de l'instruction suivante avec l'adresse annoncée dans le début de MOD1.
  - . 1600 : AD600 et AD600+01 contient l'adresse réelle
  - . 1680 : soustraction de l'adresse annoncée

La différence, qui correspond au décalage, est mise dans MICRO. Les séquences CADRA2-TESTZE-CADRA1 correspondent au balayage de toutes les instructions de MOD1 et retournent des adresses corrigées. TESTZE découvre la fin du programme lorsque l'adresse de l'instruction suivante est nulle.

En résumé, le CALL 36563 de la ligne 100 du programme BASIC réalise le calcul des

adresses du module venant d'être chargé. On peut maintenant exécuter le programme BASIC avec le nouveau module. Il nous reste à voir comment les modules avaient été créés.

### Création des modules

Comment nous y prendre pour sauvegarder par exemple le module MOD1 ?

Il suffit d'interrompre son exécution par RESET : Frapper alors BLOAD OVERL, puis CALL 36560 pour chercher la longueur du module et l'adresse de l'instruction 5000. Afficher alors AD% et FA%, puis taper BSAVE MOD1, AAD%, LFA%.

Ce procédé est complexe et risque d'introduire des erreurs. Nous proposons donc le fonctionnement suivant :

- mettre dans le programme BASIC le chargement de OVERL avec un HIMEM (voir notre exemple) pour protéger OVERL.

- utiliser une procédure EXEC qui affichera à l'écran l'instruction BSAVE MOD1, A...,L... Il suffira alors de faire traverser cette ligne au curseur.

Le petit programme ci-dessous crée le fichier EXEC nécessaire.

```
10 D$=CHR$(4) : PRINT D$"OPEN SS" : PRINT D$"WRITE SS" : PRINT"CALL 36350" :  
PRINT"CALL 36566" : PRINT"PRINT TYPE% SPC(2) AD% SPC(2) FA%" : PRINT D$"CLOSE"
```

Il suffit d'exécuter ce programme pour créer le fichier EXEC, à la première utilisation. Par la suite, il restera à taper EXEC SS.

Remarque : en même temps que l'ordre BSAVE sont affichées les valeurs TYPE%, AD%=adresse instruction 5000, FA%=longueur du module, afin que le programmeur puisse suivre l'évolution de ses modules.

Comme on peut le voir, la procédure EXEC appelle un nouveau programme assembleur, le dernier, qui affiche l'expression BSAVE et les valeurs de AD% et FA% calculées par CALL 36560. Nous laissons au lecteur le soin d'analyser ce dernier programme.

### Remarques

1. Le noyau et le module 0 doivent rester sur la disquette en forme BASIC.
2. Le module 0 doit absolument être le plus grand de tous. Il doit aussi être archivé sous forme binaire.
3. Quand on modifie le noyau, il suffit d'archiver le programme BASIC contenant le noyau et le module 0.
4. Quand on modifie un module quelconque, il faut lancer le programme par RUN pour charger les variables, arrêter par RESET, puis lancer la procédure EXEC SS.
5. Quand on modifie le module 0, il faut archiver le programme BASIC noyau+MOD0, puis effectuer la procédure EXEC SS pour aussi archiver MODO sous forme binaire.
6. Chaque module doit avoir en 5000 un sous-programme chargeant dans la variable TYPE% le numéro du module ; dans l'état actuel du programme assembleur, ce numéro doit se limiter à un chiffre.


		0010		ORG BECU
		0040	ADRDEB	EQU 67
		0060	ADRFIN	EQU AF
		0080	AD600	EQU 06
		0100	MEMVI	EQU FD
		0120	LONDOS	EQU AA60
		0140	VARIA	EQU 69
		0160	COU7	EQU FDEC
		0180	PRNTAX	EQU F941
		0200	MICRO	EQU 08
8ED0	4C20EF	0220	ENTRE1	JMP SAVE
8ED3	4C288F	0240	ENTRE2	JMP CADRAG
8ED6	4CB48F	0260	ENTRE3	JMP AFFICH
8ED9	A567	0280	RECHLI	LDA ADRDEB
8EDB	85FD	0300		STA MEMVI
8EDD	A568	0320		LDA ADRDEB+0
			1	
8EDF	85FE	0340		STA MEMVI+01
8EE1	60	0360		RTS
8EE2	A003	0380	PASSE	LDY ##03
8EE4	B1FD	0400		LDA (MEMVI),
			Y	
8EE6	C913	0420		CMP ##13
8EE8	F010	0440		BEG TROUV
8EEA	A001	0460	PASSE1	LDY ##01
8EEC	B1FD	0480		LDA (MEMVI),
			Y	
8EEE	48	0500		PHA
8EEF	88	0520		DEY
8EF0	B1FD	0540		LDA (MEMVI),
			Y	
8EF2	85FD	0560		STA MEMVI
8EF4	68	0570		PLA
8EF5	85FE	0575		STA MEMVI+01
8EF7	4CE2BE	0580		JMP PASSE
8EFA	88	0600	TROUV	DEY
8EFB	B1FD	0620		LDA (MEMVI),
			Y	
8EFD	C988	0640		CMP ##88
8EFF	F003	0660		BEG TROUV1
8F01	4CEA8E	0680		JMP PASSE1
8F04	A009	0700	TROUV1	LDY ##09
8F06	A5FE	0720		LDA MEMVI+01
8F08	9169	0740		STA (VARIA),
			Y	
8F0A	A5FD	0760		LDA MEMVI
8F0C	C8	0780		INY
8F0D	9169	0800		STA (VARIA),
			Y	
8F0F	A011	0820		LDY ##11
8F11	A5AF	0840		LDA ADRFIN
8F13	38	0860		SEC
8F14	E5FD	0880		SEC MEMVI
8F16	9169	0900		STA (VARIA),
			Y	
8F18	A5E0	0920		LDA ADRFIN+0
			1	
8F1A	E5FE	0940		SBC MEMVI+01
8F1C	88	0960		DEY
8F1D	9169	0980		STA (VARIA),
			Y	
8F1F	60	1000		RTS
8F20	DB	1020	SAVE	CLD

LABEL TABLE

ADRDEB	0067
ADRFIN	00AF
AD600	0006
MEMVI	00FD
LONDOS	AA60
VARIA	0069
COU7	FDEC
PRNTAX	F941
MICRO	0008
ENTRE1	8ED0
ENTRE2	8ED3
ENTRE3	8ED6
RECHLI	8ED9
PASSE	8EE2
PASSE1	8EEA
TROUV	8EFA
TROUV1	8F04
SAVE	8F20
CADRAG	8F28
RECHZE	8F40
TROUZE	8F4A
CADRA2	8F6A
TESTZE	8F75
CADRA1	8F7C
FINI	8F9B
TABSAU	8F9C
AFFICH	8FB4
ECRIT	8FC0
ADRESS	8FCF

8F21	20D98E	1040		JSR	RECHLI
8F24	20E28E	1060		JSR	PASSE
8F27	60	1080		RTS	
8F28	D8	1100	CADRAG	CLD	
8F29	18	1120		CLC	
8F2A	A900	1140		LDA	##00
8F2C	8506	1160		STA	AD600
8F2E	8507	1180		STA	AD600+01
8F30	AD60AA	1200		LDA	LONDOS
8F33	65FD	1220		ADC	MEMVI
8F35	85AF	1240		STA	ADRFIN
8F37	AD61AA	1260		LDA	LONDOS+0
1					
8F3A	65FE	1280		ADC	MEMVI+01
8F3C	8580	1300		STA	ADRFIN+0
1					
8F3E	A003	1320		LDY	##03
8F40	B1FD	1340	RECHZE	LDA	(MEMVI),
Y					
8F42	C900	1360		CMF	##00
8F44	F004	1380		BEG	TRGUZE
8F46	C8	1400		INY	
8F47	4C408F	1420		JMP	RECHZE
8F4A	C8	1440	TROUZE	INY	
8F4E	8406	1460		STY	AD600
8F4D	18	1480		CLC	
8F4E	A5FD	1500		LDA	MEMVI
8F50	6506	1520		ADC	AD600
8F52	8506	1540		STA	AD600
8F54	A900	1560		LDA	##00
8F56	65FE	1580		ADC	MEMVI+01
8F58	8507	1600		STA	AD600+01
8F5A	38	1620		SEC	
8F5B	A506	1640		LDA	AD600
8F5D	A000	1660		LDY	##00
8F5F	F1FD	1680		SEC	(MEMVI),
Y					
8F61	8508	1700		STA	MICRO
8F63	C8	1720		INY	
8F64	A507	1740		LDA	AD600+01
8F66	F1FD	1760		SEC	(MEMVI),
Y					
8F68	8509	1780		STA	MICRO+01
8F6A	A000	1800	CADRA2	LDY	##00
8F6C	B1FD	1820		LDA	(MEMVI),
Y					
8F6E	C900	1840		CMF	##00
8F70	F003	1860		BEG	TESTZE
8F72	4C708F	1880		JMP	CADRA1
8F75	C8	1900	TESTZE	INY	
8F76	B1FD	1920		LDA	(MEMVI),
Y					
8F78	C900	1940		CMF	##00
8F7A	F01F	1960		BEG	FINI
8F7C	18	1980	CADRA1	CLC	
8F7D	A000	2000		LDY	##00
8F7F	B1FD	2020		LDA	(MEMVI),
Y					
8F81	6508	2040		ADC	MICRO
8F83	91FD	2060		STA	(MEMVI),
Y					
8F85	C8	2080		INY	
8F86	B1FD	2100		LDA	(MEMVI),
Y					

Sicob-boutique 81  
Stand 172-174  
Face au bar



# IMAGOL

## informatique

vous apporte

### ASSISTANCE et CONSEIL

pour la

### MISE EN SERVICE

de votre

### 1<sup>er</sup> MICRO-ORDINATEUR

- Réalisation de logiciels spécifiques
- Aide au choix de logiciels standards
- Organisation de sessions d'initiation à l'ordinateur et de formation à l'utilisation des programmes


Date	Heures	Sujet	Prix H.T.
6 oct.	18 h-20 h	Les Bases de Données	100 F
8 oct.	9h-17h	VISICALC	1 000 F
13 oct.	18 h-20 h	SUPER-GENEFICHE	100F
15 oct.	9h-17h	LeM/DOS	1 000 F
20 oct.	18h-20h	VISICALC	100 F
22 oct.	9 h-17 h	Les Bases de Données	1 000 F
27 oct.	18 h-20 h	Le Traitement de texte	100 F
29 oct.	9h-17h	Le Traitement de texte	1 000 F

Toutes nos sessions ont lieu rue Gutenberg et peuvent faire l'objet d'une convention de formation.

- Location de matériel courte ou longue durée


Configuration	Week-end	1 <sup>re</sup> Semaine	Semaine sup.
Apple II + 48 K + Floppy	300 F.H.T.	600 F.H.T.	300 F.H.T.
Apple II 48 K + 2 Floppy	400 F.H.T.	750 F.H.T.	375 F.H.T.
Apple II Pascal 2 Floppy	500 F.H.T.	1 000 F.H.T.	500 F.H.T.
Imprimantes	-	5 % du P. Vte	3 % du P. Vte

- Contrats d'entretien de matériel



## apple et goupil 2

**Attention, Nouvelle Adresse :**



# IMAGOL

## informatique

1 à 5, rue Gutenberg  
75015 PARIS (M<sup>o</sup> Javel)  
tél. : (1) 577-59-39

et toujours :

**LA BOUTIQUE NOIRE du CENTRE BEAUGRENELLE**  
16, rue Linois (Niv. 1)  
75015 PARIS  
tél. : (1)575-5996

Le Spécialiste  
des nouveautés  
électroniques

BF88	6509	2120		ADC MICRO+01
BF8A	91FD	2140		STA (MEMVI),
			Y	
BF8C	A000	2160		LDY ##00
BF8E	B1FD	2180		LDA (MEMVI),
			Y	
BF90	AA	2200		TAX
BF91	C8	2220		INY
BF92	B1FD	2240		LDA (MEMVI),
			Y	
BF94	85FE	2260		STA MEMVI+01
BF96	B6FD	2280		STX MEMVI
BF98	4C6A8F	2300		JMP CADRAZ
BF9B	60	2320	FINI	RTS
BF9C	C2D3C1	2340	TABSAV	DFD *BSAVE M
	D6C5A0		OD2,A#0000,L#0000	
	CDCFC4			
	BZACC1			
	A4E0E0			
	B0E0AC			
	CCA4B0			
	B0B0B0			
BF8:4	A003	2360	AFFICH	LDY ##03
BF86	B169	2380		LDA (VARIA),
			Y	
BF88	18	2400		CLC
BF89	69B0	2420		ADC ##80
BF8B	BDA58F	2440		STA TABSAV+0
			9	
BF8E	A200	2460		LDX ##00
BF90	BD9C8F	2480	ECRIT	LCA TABSAV,X
BF93	20EDFD	2500		JSR COLT
BF96	E8	2520		INX
BF97	8A	2540		TXA
BF98	C90D	2560		CMP ##0D
BF9A	F003	2580		BEG ADRESS
BF9C	4CC08F	2600		JMP ECRIT
BF9F	A00A	2620	ADRESS	LDY ##0A
BFD1	B169	2640		LDA (VARIA),
			Y	
BFD3	AA	2660		TAX
BFD4	88	2680		DEY
BFD5	B169	2700		LDA (VARIA),
			Y	
BFD7	2041F9	2720		JSR FRNTAX
BFDA	ADAD8F	2740		LDA TAESAV+1
			1	
BFDD	20EDFD	2760		JSR CUUT
BFDE	ADAE8F	2780		LDA TABSAV+1
			2	
BFEB	20EDFD	2800		JSR COUT
BFED	ADAF8F	2820		LDA TABSAV+1
			3	
BFEB	20EDFD	2840		JSR COUT
BFED	A011	2860		LDY ##11
BFEE	B169	2880		LDA (VARIA),
			Y	
BFF0	AA	2900		TAX
BFF1	88	2920		DEY
BFF2	B169	2940		LDA (VARIA),
			Y	
BFF4	2041F9	2960		JSR FRNTAX
BFF7	60	2980		RTS

### RECAPITULATION

\*B2D0, BFF7

B2D0-	4C	20	8F	4C	26	6F	4C	B4
B2D8-	8F	A5	67	85	FD	A5	68	85
B2E0-	FE	60	A0	03	E1	FD	C9	13
B2E8-	F0	10	A0	01	E1	FD	48	88
B2F0-	E1	FC	85	FD	68	85	FC	4C
B2F8-	E2	8E	88	B1	FD	C9	88	F0
BF00-	03	4C	EA	8E	A0	09	A5	FE
BF08-	91	69	A5	FD	C8	91	69	AC
BF10-	11	A5	AF	38	E5	FD	91	69
BF18-	A5	80	E5	FE	88	91	69	60
BF20-	D8	20	D9	8E	20	E2	8E	60
BF28-	D8	18	A9	00	85	06	65	07
BF30-	AD	60	AA	65	FD	85	AF	AD
BF38-	61	AA	65	FE	85	B0	A0	03
BF40-	B1	FD	C9	00	F0	04	C8	4C
BF48-	40	8F	C8	84	06	18	A5	FD
BF50-	65	06	85	06	A9	00	65	FE
BF58-	85	07	38	A5	06	A0	00	F1
BF60-	FD	85	08	C8	A5	07	F1	FD
BF68-	85	09	A0	00	B1	FD	C9	00
BF70-	F0	03	4C	7C	8F	C8	B1	FD
BF78-	C9	00	F0	1F	16	A0	00	B1
BF80-	FD	65	08	91	FD	C8	B1	FD
BF88-	65	09	91	FD	A0	00	B1	FD
BF90-	AA	C8	B1	FD	85	FE	B6	FD
BF98-	4C	6A	8F	60	C2	03	C1	D6
BFA0-	C5	A0	CD	CF	C4	B2	AC	C1
BFA8-	A4	B0	B0	B0	B0	AC	CC	A4
BFB0-	B0	E0	B0	B0	A0	03	B1	69
BFB8-	18	69	B0	8D	A5	8F	A2	00
BFC0-	BD	9C	8F	20	ED	FD	E6	8A
BFC8-	C9	0D	F0	03	4C	C0	8F	A0
BFD0-	0A	B1	69	AA	88	B1	69	20
BFD8-	41	F9	AD	AD	8F	20	ED	FD
BFE0-	AD	AE	8F	20	ED	FD	AD	AF
BFE8-	8F	20	ED	FD	A0	11	B1	69
BFF0-	AA	88	B1	69	20	41	F9	60



# VISICALC et APPLESOFT

**Bruno Rives**

VISICALC est un programme conçu et réalisé par Dan Bricklin et Bob Frankston de la société Software Arts dans le Massachussets et distribué exclusivement par Personal Software en Californie, ces deux sociétés proposant des logiciels pour micro-ordinateurs de grande qualité.

Ce programme est né de l'observation que beaucoup de problèmes sont résolus à l'aide d'une feuille de papier, d'un crayon et d'une calculatrice, des outils universels. VISICALC allie ces outils à la performance et aux caractéristiques du micro-ordinateur, et notamment de l'Apple II. Pour l'utiliser il faut, au minimum, posséder un Apple II avec 32 K de mémoire centrale, un écran de visualisation et un lecteur de disquettes.

## Que fait VISICALC ?

VISICALC simule l'usage d'une grande feuille de papier, d'un crayon, d'une gomme et d'une calculatrice. La feuille de papier et le crayon servent à inscrire du texte, des nombres et des formules dans tous les sens. La calculatrice permet à l'utilisateur d'effectuer les calculs dont il a besoin, du plus simple (addition, soustraction, multiplication, division) au plus compliqué (moyenne, actualisation...). Enfin, la gomme laisse l'utilisateur libre d'effacer toute partie ou tout chiffre pour écrire autre chose. Mais, à la différence d'une utilisation manuelle, VISICALC gère immédiatement toute modification, suppression ou insertion en en tenant compte dans les calculs.

La feuille est organisée en 63 colonnes (identifiées de "A" à "BK") et 254 lignes (numérotées de 1 à 254). L'intersection d'une ligne et d'une colonne définit une case à laquelle on peut affecter une étiquette appelée "label" (titre ou commentaire), une valeur numérique (constante), ou enfin une formule, combinant d'autres cases, de la plus simple à la plus compliquée : c'est ce qui fait la puissance du modèle.

Un curseur permet de se déplacer très rapidement, à travers la feuille électronique dont seule une partie apparait à l'écran. Un "scrolling" très performant simule un déplacement sur la totalité de la feuille. Pour aider l'utilisateur, certaines parties du plan de travail peuvent être fixées par des commandes qui maintiennent les titres sur l'écran, ou qui divisent l'écran en deux parties totalement indépendantes, l'utilisateur pouvant se promener indifféremment dans l'une ou l'autre des deux parties.

## Les principales caractéristiques

VISICALC est très facile à utiliser : nul besoin d'être un informaticien ou de recevoir une formation spéciale pour s'en servir. Il peut être assimilé rapidement et procure des résultats immédiats : les utilisateurs ne modifient pas leur manière de travailler pour s'adapter au système. VISICALC représente bien la

philosophie des nouveaux logiciels que l'on trouve maintenant sur micro-ordinateur et qui sont totalement tournés vers l'utilisateur final. C'est la disparition totale de l'intermédiaire informaticien ; l'utilisateur achète un système (machine + logiciel) qui répond au mieux à son besoin, et il attend du système un service complet et immédiat. VISICALC a un format souple et peut donc être utilisé pour de nombreuses applications différentes : fini le temps où il fallait demander au centre informatique de la société de modifier un programme pour rajouter une ligne ou pour changer la taille des colonnes. Avec VISICALC, il est possible de changer rapidement la disposition du tableau, la largeur des colonnes, d'insérer ou supprimer des lignes ou des colonnes, sans modifier la logique du modèle.

Les possibilités de calcul font de VISICALC un outil très varié puisque l'utilisateur peut définir dans une case aussi bien des calculs courants que des formules très complexes, scientifiques ou financières : sinus, cosinus, minimum, maximum, mais aussi taux actuariel brut ...

La commande de répétition et duplication permet instantanément de copier tout ou partie de l'information du modèle, qu'elle soit sous la forme d'étiquettes, de valeurs numériques ou de formules.

Enfin, le calcul automatique donne à VISICALC sa véritable puissance. A partir du moment où l'on modifie le modèle, de quelque façon que ce soit, toutes les formules et calculs sont automatiquement mis à jour.

A ces caractéristiques, il faut ajouter toutes les commandes de formatage d'écran et de données, d'impression sur toute imprimante, de modification et de sauvegarde sur disque.

### Les applications

VISICALC possède un domaine d'utilisation très étendu puisqu'il offre la possibilité de mettre en forme des données alphabétiques et numériques (sous forme de lignes, de colonnes ou de tableaux), et de pouvoir les sauver, les éditer ou les modifier à tout moment.

Il s'adresse donc au départ aux professionnels : service des budgets et de leur suivi, départements financiers pour la construction de modèles, bureaux de planification, de simulation et de prévision, services marketing, etc... et il s'adresse a fortiori à toute personne susceptible de mettre en forme des tableaux de données : comptabilité et budgets personnels, simulation (répondre à la question : que se passerait-il si...), et même pourquoi pas l'établissement de simples "balances carrées".

L'utilisation de VISICALC couplée à celle de programmes graphiques comme APPLEPLOT, ou bien à celle de systèmes de gestion de base de données tels que CCA DMS (ces liaisons étant prévues en standard), apporte une dimension supplémentaire au programme.

### Les limites

Les principales limites que l'on peut trouver au programme VISICALC sont les suivantes :

- La limite de taille, inhérente au micro-ordinateur : on peut traiter sur une même feuille l'équivalent de trois pages dactylographiées d'étiquettes, chiffres et calculs. Il faut noter que le système tient compte de la place mémoire apportée par la carte d'extension Pascal.

- L'impossibilité de faire de la consolidation ou du rapprochement de fichiers. C'est pour cette raison que nous donnons plus loin la méthode pour analyser et construire des fichiers VISICALC à partir de programmes BASIC Applesoft.

- La non-existence de tests logiques et de branchements conditionnels ou non.

- L'impossibilité de cumuler des valeurs numériques à celles déjà inscrites. On ne peut faire l'équivalent du  $X=X+Y$  en BASIC.

La connaissance de l'organisation des fichiers VISICALC peut apporter beaucoup : elle offre d'une part la possibilité en BASIC Applesoft d'exploiter ou de modifier des fichiers existants pour agglomérer plusieurs fichiers, pour faire de la consolidation. Elle permet d'autre part de créer de toute pièces des fichiers VISICALC à partir d'un programme BASIC, afin d'exploiter par VISICALC des données venant d'un autre programme.

Un fichier VISICALC est tout d'abord un fichier séquentiel. Le programme dont le listing est donné ci-après permet de lire un fichier VISICALC et de le sortir sur imprimante :

```
10 D$=CHR$(4)
20 DIM B$(100)
30 ONERR GOTO 100
40 PRINT D$"OPEN FICHIER VISICALC"
50 PRINT D$"READ FICHIER VISICALC"
60 GET A$
70 B$(I)=B$(I)+A$
80 IF A$=CHR$(13) THEN I=I+1
90 GOTO 50
100 PRINT D$"CLOSE"
110 PRINT D$"PR#1"
120 FOR K=0 TO I
130 PRINT B$(K)
140 NEXT K
150 PRINT D$"PR#0"
160 END
```

Si l'on prend le fichier VISICALC suivant :

	A	B	C	D
1		80	81	TOT
2				
3	VENTES	1000	1500	2500
4				
5	COUTS	600	900	1500
6				
7	MARGE	400	600	1000
8				
9				

-où  $X5 = X3 \cdot 0,6$  et  $X7 = X3 - X5$  (répété sur B et C)  
et  $D3 = @SUM(B3..C3)$

voici ce que donne la lecture de ce fichier sauvé par VISICALC (/SS nom du fichier VISICALC) :

```
>D7:+D3-D5                (suite)
>C7:+C3-C5
->B7:+B3-B5                >A3:"VENTES
>A7:"MARGE                 >D1:/FR"TOT
>D5:+D3*.6                 >C1:/FR81
>C5:+C3*.6                 >B1:/FR80
>A5:"COUTS                 /W1
>D3:@SUM(B3...C3)         /GOC
>C3:1500                   /GRA
>B3:1000                   /GC9
...                          /X>A1:>A1:
```

Il faut noter d'abord l'ordre "GOTO" (>) avant chaque case du modèle. Ce fichier contient les ordres tels qu'ils ont été tapés sous VISICALC. Les données sont sauvées de la dernière ligne - dernière colonne à la première ligne - première colonne, suivant l'ordre choisi pour le calcul (en ligne ou en colonne).

Tout d'abord :

```
> GOTO case
XY coordonnées de la case
: séparateur
contenu : label
      XXX : numérique
      + XX * YY ou @SUM : formule
```

RETURN

ceci autant de fois que l'on a de cases à remplir.

Ensuite les constantes de format :

```
W1 window
GOC ordre de récapitulation
GRA recalculation (ici automatique)
GC9 taille des colonnes
```

et enfin la case sur laquelle on veut se trouver au départ, A1 par exemple.

Il est donc très facile d'exploiter ces données, ainsi que de créer des fichiers VISICALC, en procédant de manière inverse et en écrivant des chaînes alphanumériques sur un fichier séquentiel.

Si l'on veut uniquement exploiter des données numériques ou des labels venant de VISICALC, sans s'occuper des formules ou des types particuliers de formatage, le plus simple est de sauver le fichier par un ordre PRINT sur la disquette (/PD nom du fichier VISICALC). Le contenu de toutes les cases est alors sauvé en séquentiel, chaque position de la case (même vide) comptant pour un caractère, et chaque fin de ligne se caractérisant par un RETURN (exactement comme une impression sur imprimante). Voici le contenu du fichier décrit précédemment, mais cette fois-ci sauvé par l'ordre PRINT sur disquette et lu avec le programme donné plus

haut :	80	81	TOT
VENTES	1000	1500	2500
COUTS	600	900	1500
MARGE	400	600	1000

Chaque ligne ici comporte 37 caractères : 4 colonnes de 9 caractères plus un RETURN (les lignes vides comportant aussi 37 caractères, 36 espaces plus un RETURN).

Pour toute information complémentaire ou toute précision que vous pourriez apporter, écrivez au journal qui transmettra.

### Le nouveau VISICALC 16 secteurs (DOS 3.3)

La nouvelle version de VISICALC comporte des caractéristiques qui apportent au système des avantages très intéressants :

- un éditeur de ligne permet de modifier les données, labels ou formules précédents très rapidement sans avoir à retaper toute la ligne.
- il est désormais possible d'imprimer le modèle tel qu'il a été rentré, c'est à dire avec ses formules.
- la commande /SL pour appeler un fichier du disque a été modifiée pour ne plus écraser la feuille actuelle, ce qui autorise une certaine forme de consolidation (malgré tout limitée).
- l'introduction des tests apporte véritablement le choix du traitement à effectuer : la fonction @IF permet l'équivalent du IF...THEN...ELSE.
- la fonction @CHOOSE, comme son nom l'indique, choisit le nième argument dans une liste précisée par l'utilisateur.
- enfin, on peut désormais quitter le programme VISICALC sans être obligé d'éteindre l'Apple à l'aide de la commande QUIT.

Pour ceux qui voudraient utiliser leurs anciens fichiers VISICALC 13 secteurs, il suffit de les transformer par l'utilitaire MUFFIN du DOS 3.3.

## LES DISQUES SOUPLES 8" SIBDISC POUR



**LEANORD**  
ISA.CREUSOT LOIRE



**Lille :**  
236, rue Sadi Carnot - 59320 HAUBOURDIN  
Tél.: (20) 07.30.55 - Télex: 810910 F



**SICOB Stand 3E-3511**

**Paris :**  
15, rue Guyton de Morveau - 75013 PARIS  
Tél.: (1) 589.73.33 - Télex: 201368 F



EB 10/010 983

@marque déposée  
Apple Computer Inc

# MASTOCK-II-A

## LOGICIEL DE GESTION DE STOCK ET DE FACTURATION

**MASTOCK-II-A** est un logiciel bien adapté aux entreprises désirant maîtriser la gestion de leur stock (2000 articles). Ce logiciel permet les inventaires valorisés, le suivi des approvisionnements, la facturation des ventes, les statistiques de ventes, etc.

Avec **MASTOCK-II-A** et un ordinateur APPLE II vous disposerez d'un outil de gestion performant.

3000 FF  
TTC

# MICROGÈS

Société spécialisée dans la diffusion  
de logiciels de gestion.

Demandez aux revendeurs de matériels  
APPLE II de vous présenter nos produits

30 Bd de Glatigny  
78000 VERSAILLES  
Tél. (3) 955.30.23

3000 FF  
TTC

## LOGICIEL INTERACTIF DE PAYE

Avec **MAPAYE-I-A** et un ordinateur APPLE II vous pourrez calculer la paye de vos salariés mensuels ou horaires.

Ce logiciel permet à une PME ou PMI de gérer jusqu'à 150 bulletins de salaires et 200 rubriques et d'éditer à la demande les états fiscaux, comptables ou de gestion.

# MAPAYE-I-A

# Un programme aide-mémoire

Gérard Michel

## Description générale du programme

Le programme dont la liste apparait ci-dessous est un petit programme aide-mémoire sans prétention. Le lecteur pourra à loisir l'améliorer et l'adapter à ses propres besoins.

Le programme mémorise pour chaque information de l'aide-mémoire un code de référence ayant un maximum de six caractères. Ces codes sont en permanence classés en mémoire, selon l'ordre des codes alphanumériques.

De cette façon, on retrouve très rapidement n'importe quelle information par l'appel d'une séquence alphanumérique initiale du code : il n'est pas nécessaire de reproduire la totalité du code. Ainsi, en demandant une recherche à partir du code 'ABC', on obtiendra toutes les informations dont le code commence par la séquence 'ABC'.

Il est également possible d'avoir plusieurs informations ayant des codes identiques. Il faut cependant se méfier dans ce cas : quand on souhaite annuler l'information relative à un code donné, le programme élimine la première information qu'il trouve avec ce code. Or, le programme utilise une procédure de recherche dichotomique : la première occurrence du code qu'il trouvera ne sera pas nécessairement celle correspondant à la première information ayant ce code dans la liste.

Dès l'appel du programme débute la lecture de la table de références, c'est-à-dire des codes et des indices des zones libres sur disquette. L'utilisateur a ensuite le choix entre insérer, rechercher ou lister des données. Dans ce dernier cas, la liste est faite au choix à l'écran ou sur imprimante (code en ligne 980).

Pour annuler une information, il faut d'abord la trouver grâce à la recherche, fonction dans laquelle les options possibles sont :

- suite des informations, s'il y en a trop avec ce code pour tenir sur un écran.
- annulation : le programme demande le code exact à annuler. En effet, la recherche a pu se faire avec un sous-ensemble du code.
- retour au menu principal.

## Description des fichiers

Le fichier séquentiel, MEMOR (MEMO, table de Référence), comporte deux parties. En premier lieu, nous trouvons la liste de la table de référence : pour chaque information, le code auquel est accolé le numéro de l'enregistrement sur disquette occupé par l'information (augmenté de 100, pour toujours utiliser trois caractères).

La seconde partie de MEMOR indique le nombre d'enregistrements sur disquette

libérés par des annulations antérieures et restés libres. Ce nombre est suivi de la liste des enregistrements ainsi libérés, lus par le programme dans le tableau IND.

Le fichier à accès direct, MEMOT (MEMO, Texte), mémorise le texte d'information associé à chaque code. Il a été choisi de limiter le texte à la plus grande taille permettant de faire tenir code et texte lisiblement au rythme d'une information par ligne.

#### Quelques points intéressants du programme

Afin d'améliorer la rapidité de fonctionnement, toutes les routines ont été placées au début du programme. La plus grande est la routine 50-57 réalisant la recherche dichotomique d'une information à partir du code.

La routine en 120-123 donne un menu en une ligne en bas d'écran, ce qui est très pratique quand on souhaite poser une question à réponses multiples sans perdre de place.

#### Modifications et améliorations possibles

Les règles de recherche du code peuvent être changées. Il est possible d'exiger que le code soit fourni intégralement ou que les caractères fournis soient une séquence quelconque à l'intérieur du code. On peut aussi vouloir rechercher telle ou telle chaîne à une position donnée du code, par exemple, rechercher toutes les informations dont le code possède un 'A' en colonne 4 et un '7' en colonne 6. On conviendrait, par exemple, de rentrer '\$\$\$A\$7' comme code, le signe '\$' étant choisi comme caractère neutre vis-à-vis de la recherche.

Il est de même possible d'augmenter la taille de l'information, ou de l'éclater en un champ alphanumérique et un champ numérique. Si l'on se laissait aller à toutes ces améliorations, on finirait par écrire un programme "base de données", mais il y en a déjà de bons sur le marché !



# LIBRAIRIE LA NACELLE

INFORMATIQUE • ÉLECTRONIQUE • AUTOMATISME • MICROPROCESSEUR

**TOUS OUVRAGES ET ABONNEMENTS  
FRANÇAIS ET ÉTRANGERS**

Tous les ouvrages français ou étrangers signalés dans cette revue peuvent être obtenus ou commandés à La Nacelle

**2, rue Campagne-Première 75014 PARIS - Tél. 322 56 46**

Métro Raspail - Parking à la hauteur du 120 bd du Montparnasse

ouvert tous les jours lundi compris, sans interruption de 9 h 30 à 18 h 50, samedi fermeture à 17 h 50.



```

1 REM *****
  *
  * PROGRAMME *
  * AIDE-MEMOIRE *
  *
  *****
4 REM
5 REM *****
  * DEFINITION DES *
  * VARIABLES *
  *
  * NOM$ : INDEX EN *
  * MEMOIRE *
6 REM * IND : ADRESSES *
  * LIBRES EN MEMOIRE *
  *****
10 PRINT CHR$(4);"CLOSE": PRINT
  CHR$(4);"MAXFILES 4";D$ =
  CHR$(4);PO$ = "....."
  .":BL$ = " ":PO$ =
  PO$ + PO$ + PO$:BL$ = BL$ +
  BL$
20 ONERR GOTO 9900
30 DIM NOM$(200),IND(100)
35 GOSUB 9000: GOTO 400
44 REM *****
  * RECHERCHE DES *
  * CARACTERES NON *
  * BLANCS DANS LE *
  * CODE *
  *****
45 ZX$ = "":L = LEN (ZZ$): FOR I
  = 1 TO L:Z$ = MID$(ZZ$,I,
  1): IF Z$ < > " " THEN ZX$ =
  ZX$ + Z$
46 NEXT :ZZ$ = ZX$:LC = LEN (ZZ
  $): RETURN
49 REM *****
  * RECHERCHE DU CODEX
  *****
50 K1 = 1:K2 = NZ:Q = LC: IF Z =
  1 OR Z = 9 THEN Q = 6
51 K = INT ((K1 + K2) / 2):A$ =
  LEFT$(NOM$(K),Q):B$ = LEFT$(
  R$,Q): IF B$ = A$ THEN K1 =
  - 1: RETURN
52 IF B$ < A$ AND K2 > K1 + 1 THEN
  K2 = K: GOTO 51
53 IF B$ < A$ THEN RETURN
54 IF K2 > K1 + 1 THEN K1 = K: GOTO
  51
55 K = K + 1: IF B$ > LEFT$(NOM
  $(K2),Q) THEN K = K2 + 1
56 IF B$ = LEFT$(NOM$(K2),Q) THEN
  K1 = - 1:K = K2: RETURN
57 RETURN
58 REM
60 VTAB V: HTAB 1: CALL - 868: HTAB
  20 - LEN (Z$) / 2: INVERSE
  : PRINT Z$: NORMAL : RETURN

```

```

69 REM *****
  * IMPRESSION DE *
  * MESSAGE ET *
  * DEMANDE DE *
  * REPONSE *
  *****
70 VTAB 20: HTAB 1: CALL - 868:
  INVERSE : PRINT ME$(Z): NORMAL
  : INPUT " ? ";Z$: FOR ZZ = 1
  TO 1000: NEXT : POKE 34,ZZ:
  HOME : POKE 34,5:Z$ = LEFT$(
  Z$,1): IF Z$ = "0" OR Z$ =
  "N" THEN RETURN
71 GOTO 70
75 FOR L = 1 TO 1500: NEXT : RETURN
80 VTAB V: HTAB 1: CALL - 868: INVERSE
81 PRINT Z$: NORMAL : RETURN
89 REM *****
  * TITRE CENTRE *
  *****
90 HOME : HTAB 20 - LEN (Z$) /
  2: INVERSE : PRINT Z$: NORMAL
  : RETURN
119 REM *****
  * IMPRESSION *
  * DU MENU *
  *****
120 VTAB V: HTAB 1: CALL - 868:
  FOR ZZ = 1 TO TT$(Y):Z$ = T
  T$(TIX(Y) + ZZ): INVERSE : PRINT
  LEFT$(Z$,1): NORMAL : PRINT
  RIGHT$(Z$, LEN (Z$) - 1) SPC(
  ZY): NEXT
121 INPUT " ";Z$: FOR ZZ = 1 TO
  TT$(Y): IF Z$ = LEFT$(TT$(
  TIX(Y) + ZZ),1) THEN 123
122 NEXT : GOTO 120
123 RETURN
149 REM *****
  * ENTREE DE *
  * DONNEES AU *
  * CLAVIER *
  *****
150 VTAB V: HTAB H: PRINT LEFT$(
  (PO$,LO)" ":ZZ$ = "": VTAB: V
  : HTAB H: INPUT " ";ZZ$: IF LEN
  (ZZ$) > LO THEN Z$ = ME$(5):
  GOSUB 250: POKE 34,V: HOME
  : POKE 34,5: GOTO 150
160 RETURN
245 FOR LY = 1 TO 50:LL = PEEK
  (- 16336): NEXT : RETURN
250 VTAB 24: HTAB 1: CALL - 868
  : INVERSE : GOSUB 245: PRINT
  Z$: NORMAL : FOR LY = 1 TO
  1500: NEXT : HTAB 1: CALL -
  868: RETURN

```

```

255 VTAB 24: HTAB 1: CALL - 86E
: INVERSE : GOSUB 245: PRINT
Z$: NORMAL : GET Z$: HTAB 1
: CALL - 86E: RETURN
399 REM *****
* *
* DEBUT DU *
* PROGRAMME *
* *
*****
400 TEXT :Z$ = "PROGRAMME AIDE-M
EMOIRE": GOSUB 90: VTAB 10: PRINT
"LECTURE DE LA TABLE DE REFE
RENCES ..."
405 FOR L = 1 TO 1000: NEXT
408 REM *****
* MEMOT = FICHER *
* TEXTE *
* MEMOR = FICHER *
* INDEX *
*****
410 POKE 34,22: PRINT : PRINT D$
:"OPEN MEMOT,L41": PRINT D$:
"PR#0"
420 CE = 1: PRINT : PRINT D$:"OPE
N MEMOR": PRINT D$:"READ MEM
OR": INPUT NZ: FOR L = 1 TO
NZ: INPUT NOM$(L): NEXT : INPUT
NL: IF NL > 0 THEN FOR L =
1 TO NL: INPUT IND(L): NEXT
430 PRINT D$:"CLOSE MEMOR":CE =
0
440 POKE 34,2: HOME :V = 4: PRINT
"CODE TEXTE DE REFER
ENCE ": PRINT "-----
-----": POKE 34,4: GOTO 510
479 REM *****
* ROUTINE D'ENTREE*
* DU CODE AU *
* CLAVIER *
*****
480 V = UX:H = HX:LO = 6: GOSUB 1
50: GOSUB 45: IF ZZ$ = "" THEN
9999
490 R$ = ZZ$: IF LEN (R$) < 6 THEN
R$ = R$ + LEFT$ (BL$,6 - LEN
(R$))
500 GOSUB 50: RETURN
510 HOME :UX = 7:HX = 1:V = 19:Z
Y = 2:Y = 1: GOSUB 120: IF Z
$ = "I" THEN Z = 1: GOTO 600
550 IF Z$ = "R" THEN 700
570 IF Z$ = "L" THEN 950
580 IF Z$ = "S" THEN 9999

```

```

599 REM *****
* INSERTION DE *
* REFERENCES *
*****
600 GOSUB 480:V = 7:H = 8:LO = 3
2: GOSUB 150: IF ZZ$ = "" THEN
Z$ = ME$(4): GOSUB 250: GOTO
510
620 T$ = ZZ$: VTAB 24: HTAB 1: CALL
- 86E:U$ = ZZ$: GOSUB 70: IF
Z$ < > "0" THEN 600
638 REM *****
* DEFINITION DE *
* L'ADRESSE DE LA *
* NOUVELLE *
* REFERENCE *
*****
639 REM *****
* DECALAGE DE *
* L'INDEX *
*****
640 FOR L = NZ TO K STEP - 1:NO
M$(L + 1) = NOM$(L): NEXT :N
Z = NZ + 1: IF NL < 1 THEN I
= NZ: GOSUB 660: GOTO 510
650 I = IND(NL):NL = NL - 1: GOSUB
660: HOME : GOTO 510
660 NOM$(K) = R$ + STR$ (100 + I
): POKE 34,23: PRINT : PRINT
D$:"WRITE MEMOT,R":I: PRINT
R$: PRINT T$: PRINT D$:"PR#0
"
661 PRINT : PRINT D$:"OPEN MEMOR
": PRINT D$:"WRITE MEMOR": PRINT
NZ: FOR L = 1 TO NZ: PRINT N
OM$(L): NEXT
662 PRINT NL: IF NL > 0 THEN FOR
L = 1 TO NL: PRINT IND(L): NEXT
663 PRINT D$:"PR#0": POKE 34,5: RETURN
699 REM *****
* RECHERCHE DES *
* REFERENCES SUR *
* UNE PARTIE DU *
* CODE DONNEE *
*****
700 Z = 2: GOSUB 480: IF K1 > -
1 THEN Z$ = ME$(6): GOSUB 25
0: GOTO 510
720 JK = K - 1: FOR L = JK TO 0 STEP
- 1: IF LEFT$ (NOM$(L),LC)
< > B$ THEN L = 0: GOTO 74
0
730 K = K - 1
740 NEXT
760 JK = K:JJ = 0: FOR J = JK TO
NZ: IF LEFT$ (NOM$(J),LC) <
> LEFT$ (R$,LC) THEN J = N
Z: GOTO 840

```

```

780 I = VAL ( RIGHT$ (NOM$(J),3)
) - 100: POKE 34,22: PRINT :
PRINT D$;"READ MEMOT,R";I: INPUT
R$,T$: PRINT : PRINT D$;"PR#
0": POKE 34,5: VTAB 7 + JJ: CALL
- 868: PRINT R$ "T$:JJ = J
J + 1: IF JJ < 12 THEN 840
789 REM *****
* MENU : SUITE, *
* RETOUR DU *
* ANNULLATION *
*****
790 V = 19:ZY = 4:Y = 2: GOSUB 12
0
800 IF Z$ = 'S' THEN HOME :JJ =
0: GOTO 840
810 IF Z$ = 'R' THEN 510
820 Z = 9: VTAB 22: HTAB 1: CALL
- 868: HTAB 1: PRINT "CODE
A ANNULER :.....":VX = 22:K
X = 17: GOSUB 480: IF K1 > -
1 THEN Z$ = ME$(6): GOSUB 25
0: GOTO 790
830 Z = 9: GOSUB 70: IF Z$ < > "
0" THEN 790
834 NL = NL + 1
835 IN$(NL) = VAL ( RIGHT$ (NOM$(
(K),3)) - 100:NZ = NZ - 1: FOR
L = K TO NZ:NOM$(L) = NOM$(L
+ 1): NEXT : POKE 34,23: GOSUB
661:Z = 2: GOTO 510
840 NEXT : VTAB 22: HTAB 1: CALL
- 868: GOSUB 245: INVERSE :
PRINT "IL N'Y A PLUS RIEN A
CE CODE-LA ...": NORMAL : GOSUB
75: HTAB 1: CALL - 868: GOTO
790
899 REM *****
* IMPRESSION DE *
* LISTE A L'ECRAN *
*****
900 HOME : VTAB 7: HTAB 1: FOR K
= 1 TO NZ:I = VAL ( RIGHT$(
NOM$(K),3)) - 100: PRINT D$
;"READ MEMOT,R";I: INPUT R$,
T$
920 PRINT "R$ "T$: IF 16 * ( INT
(K / 16)) = K THEN V = 24:Z$
= ME$(8): GOSUB 80: PRINT D
$;"IN#0": VTAB 23: HTAB 39: GET
Z$: PRINT : HOME : VTAB 7: HTAB
1: IF ASC (Z$) = 27 THEN 51
0

```

```

940 NEXT :V = 24:Z$ = ME$(8): GOSUB
80: PRINT D$;"IN#0": VTAB 23
: HTAB 39: GET Z$: HOME : GOTO
510
949 REM *****
* IMPRESSION DE *
* LISTE SUR *
* L'IMPRIMANTE *
* (SLOT 4) *
*****
950 Z = 7: GOSUB 70: IF Z$ < > "
0" THEN HOME : GOTO 900
960 GOSUB 245:Z = 10: GOSUB 70: IF
Z$ < > "0" THEN HOME : GOTO
510
960 PRINT D$;"PR#4": PRINT SPC(
9) CHR$(27); CHR$(14)"AIDE
-MEMOIRE": PRINT : FOR K = 1
TO NZ:I = VAL ( RIGHT$ (NO
M$(K),3)) - 100: PRINT D$;"RE
AD MEMOT,R";I: INPUT R$,T$: PRINT
" ";
1000 PRINT R$ SPC( 10 - LEN (R$
))T$: NEXT : PRINT D$;"IN#0"
: PRINT D$;"PR#0": HOME : GOTO
510
9000 REM *****
* LECTURE DES *
* LIBELLES *
*****
9080 DATA ENREGISTREMENT C
ONFIRME,MODIFICATION CONFIRM
EE,"APPUYER SUR UNE TOUCHE,
'ESC' POUR STOP","ALORS, JE
NE PEUX RIEN FAIRE ..."
9085 DATA INFORMATION TROP LONG
UE,AUCUN ENREGISTREMENT NE P
OSSEDE CE CODE,VOULEZ-VOUS U
NE LISTE IMPRIMEE,APPUYER SU
R UNE TOUCHE APRES LECTURE,A
NNULATION CONFIRMEE,IMPRIMAN
TE BIEN CONNECTEE
9090 FOR L = 1 TO 10: READ ME$(L
): NEXT
9100 DATA 4,INSERTION,RECHERC
HE,LISTES,STOP
9120 DATA 3,SUITE,ANNULATION,R
ETOUR
9200 L = - 1: FOR Y = 1 TO 2: READ
TT$(Y): FOR LL = 1 TO TT$(Y)
: READ TT$(L + LL): NEXT :TI
$(Y) = L:L = L + TT$(Y): NEXT
9205 RETURN
9900 IF PEEK (222) = 5 AND DE =
1 THEN 430
9999 PRINT : PRINT D$;"CLOSE": TEXT
: HOME : ENC

```

# APPLE II PLUS. CINQ RAISONS POUR ALLER LE VOIR DE PLUS PRES.

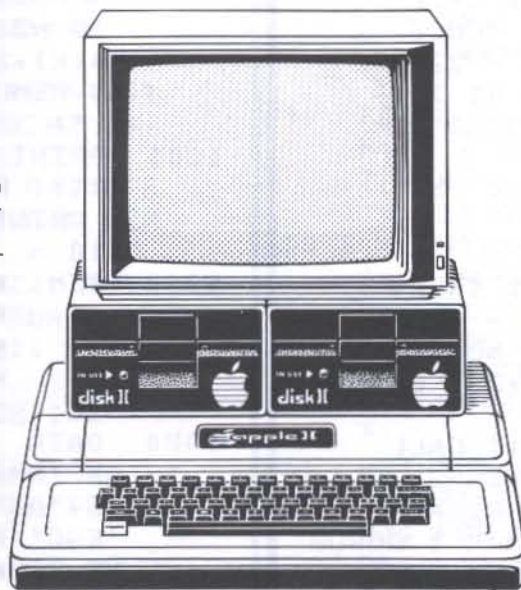
**I** Première raison: Apple II Plus est un ordinateur personnel étonnant. L'ordinateur personnel, c'est un nouveau concept: la relation individuelle et directe entre l'homme et sa machine. Apple II Plus a la taille d'une machine à écrire, la simplicité d'une règle à calcul, les capacités d'un ordinateur, et le coût d'une photocopieuse.

**II** Deuxième raison: Apple II Plus intéresse autant un chef d'entreprise qu'un avocat, un chercheur qu'un homme d'affaires. Apple II Plus vous aidera à résoudre aussi bien des problèmes d'ordre administratif, de gestion ou de marketing que de recherche et d'ingénierie.

**III** Troisième raison: Apple II Plus dispose d'une bibliothèque très vaste de programmes très performants mais d'utilisation simple qui couvrent la quasi-totalité des branches d'activité.

**IV** Quatrième raison: Apple II Plus, c'est un réseau de revendeurs agréés et soutenus par Apple. Grâce à une connaissance pratique de l'informatique et de ses derniers développements, chacun d'eux saura vous conseiller efficacement.

**V** Cinquième raison: les revendeurs agréés Apple forment un réseau après-vente entièrement à votre disposition pour les très rares fois où vous pourriez avoir besoin d'eux.



 **apple computer**

Apple vient d'introduire sur le marché plus de 30 nouveaux programmes qui ont été spécialement conçus pour Apple II Plus. En voici quelques-uns:

- **PLAN 80:** Un programme élaboré de planification financière.
- **APPLE PROJECT MANAGER:** Permet de suivre plus de 2000 tâches d'un même projet.
- **REDEFINABLE DATA BASE:** Programme de gestion de base de données.
- **PERSONAL FINANCE MANAGER:** Permet le contrôle complet de budgets personnels au d'entreprise.
- **VISITREND/VISIPLOT™:** Un système complet de prévision et d'analyse statistique.
- **VISIPILOT™:** Pour la production - en couleur - de graphes, tableaux, etc...
- **VISICALC™:** Le "Best-seller" parmi les programmes de modèles d'analyse financière.
- **APPLE WRITER:** Traitement de texte.
- **APPLE PLOT:** Création de graphiques et tableaux.
- **APPLE PILOT:** Programme d'enseignement.
- **APPLE GRAPHICS:** Un système graphique en 3 dimensions.

™ : Marques de Personal Software, Inc.

**Au SICOB 81 du 28 Septembre au 2 Octobre, niveau 4, zone B, stand n° 4A4116**

Si vous souhaitez vous informer sur l'ensemble de la gamme Apple, remplissez ce coupon et retournez-le à:

**SONOTEC S.A.** 41-45, rue Galilée - 75016 Paris - Tél. : 723 78 56.

**Pour l'Europe :**

**APPLE COMPUTER INTERNATIONAL** 7, rue de Chartres - 92200 Neuilly - Tél. : 624 21 13.

Nom \_\_\_\_\_

Société \_\_\_\_\_ Activité \_\_\_\_\_

Adresse \_\_\_\_\_

Code postal \_\_\_\_\_ Tél. \_\_\_\_\_

# Des instructions en une lettre

Christian Guérin

Un ordinateur aide normalement à travailler plus vite et mieux. Cependant, l'aspect rapidité n'est pas toujours évident quand il faut taper de nombreux caractères pour donner un ordre fréquent, à l'intérieur d'un vocabulaire d'ordres restreint. Ainsi, pourquoi taper 'CATALOG' au lieu de simplement 'C', ou 'SAVE' au lieu de 'S'.

Le programme en assembleur que je vous propose vous permettra de remplacer les ordres du DOS par des ordres à une seule lettre. L'objet de ce programme est de remplacer certaines lettres choisies, quand il leur arrive d'être la première lettre d'un message envoyé à l'ordinateur, par des instructions complètes en DOS.

Ainsi, au lieu de taper 'SAVE BIDULE', il vous suffira de taper 'S BIDULE', le 'S' que vous aurez tapé se transformera sur l'écran en SAVE immédiatement. De cette façon, tout 'S' tapé en première lettre devient un 'SAVE', tout 'C' devient un 'CATALOG',...

Une fois que vous aurez compris le fonctionnement, vous pourrez modifier les codes de votre propre chef et introduire les instructions en une lettre qui vous conviennent le mieux.

Il vous suffit pour cela de regarder comment est constituée la table en fin de programme et de la modifier à loisir. Vous pouvez ainsi constater que, dans le cas du ';' qui remplace l'instruction 'LIST', le programme fournit automatiquement le RETURN. Vous avez alors remplacé 'LIST' suivi de RETURN par un ';'.

Dans le cas des autres instructions, le RETURN doit suivre la lettre code que vous avez tapée, mais vous pouvez aussi court-circuiter ces RETURN en procédant de la même manière que pour le remplacement du 'LIST'.

Récapitulatif des codes :

C remplace CATALOG  
S remplace SAVE  
L remplace LOAD  
R remplace RUN  
; remplace LIST (RETURN)

Il vous reste une précaution à prendre : tant que le programme assembleur reste actif, la première lettre est toujours transformée si elle correspond à l'un de vos codes. Si, à l'intérieur d'un programme BASIC, un INPUT vous demande votre prénom et que vous répondez 'CHARLES', l'ordinateur comprendra 'CATALOGCHARLES' !

```

1      NLS
2      ;
3      KSWL   EPZ $38
4      BASL   EPZ $28
5      ;
6      KEYIN  EQU $FD16
7      ;
8      ;
9      ORG   $300
10     ;
11     LDA  #PROG
12     STA  KSWL
13     LDA  /PROG
14     STA  KSWL+1
15     ;
16     LDA  #0
17     STA  FLAG
18     ;
19     RTS
20     ;
21     ;
22     ;
23     ;
24     FLAG   BYT 0
25     SAVE.Y BYT 0
26     ;
27     ;
28     ;
29     PROG   STY SAVE.Y
30           LDY FLAG
31           BNE P6
32     P1     LDY SAVE.Y
33           JSR KEYIN
34           CPX #0
35           BNE FLAG=0
36           LDY #$$FF
37     P2     INY
38           CPY #FTABLE
39           BNE P3
40     FLAG=0 LDY #0
41           BEQ P5
42     P3     CMP TABLE,Y
43           BNE P2
44     P4     INY
45     P5     STY FLAG
46           LDY SAVE.Y
47           RTS
48     ;
49     P6     PHA
50           LDA TABLE,Y
51           BPL P7
52           PLA
53           JMP P1
54     ;
55     P7     JSR P4
56           STA SAVE.Y
57           PLA
58           STA (BASL),Y

```

```

59      LDA SAVE.Y
60      ORA #$80
61      RTS
62 ;
63 ;
64 TABLE ASC 'C'
65         ASC 'ATALOG'
66         ASC 'S'
67         ASC 'AVE '
68         ASC 'L'
69         ASC 'OAD '
70         ASC 'R'
71         ASC 'UN '
72         ASC ';'
73         BYT 'H'-$40
74         ASC 'LIST'
75         BYT 'M'-$40
76         ASC ' '
77 ;
78 ;
79 FTABLE EQU *-TABLE-1
80 ;
81 ;
82 FIN    EQU *
83 ;
84 ;
85        DCM 'BSAVE B:CLAV,A$300,L$80'
86 ;
87        END

```

\*300.380

```

0300- A9 10 85 38 A9 03 85 39
0308- A9 00 8D 0E 03 60 00 00
0310- 8C 0F 03 AC 0E 03 D0 22
0318- AC 0F 03 20 1B FD E0 00
0320- D0 07 A0 FF C8 C0 1E D0
0328- 04 A0 00 F0 06 D9 53 03
0330- D0 F2 C8 8C 0E 03 AC 0F
0338- 03 60 48 B9 53 03 10 04
0340- 68 4C 18 03 20 32 03 8C
0348- 0F 03 68 91 28 AD 0F 03
0350- 09 80 60 C3 41 54 41 4C
0358- 4F 47 D3 41 56 45 20 CC
0360- 4F 41 44 20 D2 55 4E 20
0368- BB 08 4C 49 53 54 0D A0
0370- 00 A5 0D 85 01 38 D8 A5
0378- 00 E5 02 85 00 A5 01 E5
0380- A0

```

# Graphiques : de l'ITT2020 à l'Apple

Luis Nobre

L'article paru dans l'Ordinateur Individuel numéro 28, pp 97 à 99 donnait le détail d'une modification possible afin de rendre compatible au point de vue graphique, l'Apple II et l'ITT 2020. Quelques explications s'avèrent nécessaires pour comprendre l'utilisation du graphique sur ITT 2020 et les limitations de cette modification.

## Ecran mémoire

La définition du graphique haute résolution de l'ITT 2020 est de 360 points en horizontal, soit 40 fois 9 bits. A chaque adresse de la mémoire écran haute résolution correspondent 9 points haute résolution consécutifs, situés sur une même ligne.

La définition verticale est de 192 points en graphique et de 160 points lorsque le mode mixte est utilisé.

On utilise donc  $192 \times 360/9 = 7680$  adresses mémoires pour afficher la haute résolution.

L'espace mémoire total réservé est situé de 8192 à 16383, soit 8192 adresses. Il y a donc 512 adresses qui restent libres et sont sans effet sur l'affichage, les mêmes que pour l'Apple.

## Ecriture des 9 bits

Il n'est pas possible de faire l'opération en une seule fois puisque le microprocesseur 6502 travaille avec des mots de 8 bits. L'ITT 2020 emploie un "inverseur logiciel" qui dans un cas permet l'écriture de 8 bits et dans l'autre du 9ème bit en plus. Cet inverseur est constitué par les adresses C05E et C05F. L'adresse C05E autorise l'écriture du 9ème bit qui est alors recopié à partir du 8ème bit de la donnée. L'adresse C05F déconnecte l'écriture du 9ème bit. Celui-ci garde alors la valeur précédemment mémorisée.

L'exemple fourni dans le tableau 1 vous permettra de mieux comprendre le mécanisme de l'écriture en suivant pas à pas l'effet des commandes sur l'écran.

## Lecture des 9 bits

L'approche est légèrement différente de celle décrite ci-dessus pour l'écriture. La lecture d'une adresse mémoire donne les huit premiers bits. Le 9ème bit est également lu lors de cet appel et sa valeur est placée dans le huitième bit (de poids fort) de l'adresse C063 (les sept autres bits ne sont pas significatifs). Le tableau 2 donne un exemple de lecture de la page composée par les



instructions du tableau 1.

### Portée de la modification

Le chargement, à partir d'une disquette, d'une page graphique prévue pour l'Apple ne positionne pas le 9ème bit supplémentaire. Il s'ensuit que l'on observe sur l'écran des raies verticales et discontinues (suivant l'état du 9ème bit) disposées tous les 9 points. La modification décrite dans l'O.I. permettait de ne plus visualiser ce 9ème bit.

Le problème n'est cependant pas totalement résolu. En effet, l'ITT visualise les 8 bits d'un mot, tandis que l'Apple n'en visualise que 7, le dernier étant réservé pour une information couleur. L'ITT incorporera donc tous les 8 points soit un point allumé (bit couleur à 1), soit un espace (bit couleur à 0).

Il ne parait pas possible pour l'instant d'effectuer une modification matérielle permettant de ne plus tracer ce huitième bit et de revenir donc dans une configuration totalement compatible avec l'Apple.

### Quelques autres problèmes

En raison du double adressage nécessaire tant en lecture qu'en écriture, toutes les routines haute résolution sont plus lentes avec l'ITT.

D'autre part, la façon dont le neuvième bit est adressé ne permet pas le stockage de pages haute résolution par un BSAVE. Il faut dans ce cas un programme spécial.

Enfin, la ROM DO de l'ITT n'est pas la même que celle de l'Apple. Elle a été modifiée afin d'adapter toutes les routines haute résolution à la configuration particulière de l'ITT.

L'une des différences les plus flagrantes concerne le compteur de collisions utilisé avec les "shape". La valeur contenue dans ce compteur est élaborée de façon très différente et donne des résultats surprenants avec certains jeux.

En raison de toutes ces particularités, il semble très difficile de pouvoir rendre l'ITT totalement compatible avec l'Apple. Si vous voulez pouvoir exécuter tous les programmes, notamment les jeux, le meilleur moyen semble être de revendre votre ITT pour acheter un Apple ! ➔

# SOFTCARD Z 80 & RAMCARD 16K

**POUR**  
 **apple™**



**LEANORD**  
ISA.CREUSOT-LOIRE



**MICROSOFT**  
CONSUMER PRODUCTS

**SICOB Stand 3E-3511**

**Lille :**  
236, rue Sadi Carnot - 59320 HAUBOURDIN  
Tél. : (20) 07.30.55 - Télex : 810910 F

**Paris :**  
15, rue Guyton de Morveau - 75013 PARIS  
Tél. : (1) 589.73.33 - Télex : 201368 F

● marque déposée  
Apple Computer Inc

88 hypapub 187

TABLEAU 1

SOUS MONITEUR	SOUS BASIC	FONCTION
* C050	Poke - 16304,0	} passage en haute résolution mode mixte
* C053	Poke - 16301,0	
* C057	Poke - 16297,0	
* C05E	Poke - 16290,0	autorisation écriture 9ème bit
* 2000 : FF	Poke - 8192,255	FF:11111111
* 2001 : 7F	Poke - 8193,127	7F:01111111
* 2002 : 80	Poke - 8194,128	80:10000000
* C05F	Poke - 16289,0	} écriture de 9 bits le 8ème et le 9ème étant identiques (1)
* 2000 : 0F	Poke - 8192,15	
* 2001 : FF	Poke - 8193,255	
* 2002 : 00	Poke - 8194,0	00:0000 0000
		} écriture de 8 bits le 9ème étant inchangé (2)

Les 27 premiers points sur l'écran sont à l'état suivant :

points	1 . . . . . 8	9	10 . . . . . 17	18	19 . . . . . 26	27
(1)	B B B B B B B B	B	B B B B B B B N	N	N N N N N N N B	B
(2)	B B B B N N N N	B	B B B B B B B B	N	N N N N N N N N	B

TABLEAU 2

SOUS MONITEUR	SOUS BASIC	RESULTAT OBTENU
* 2000	Peek (8192)	15 = \$0F = 0000 1111 lecture des 8 bits
* C063	Peek (-16285)	>127 =>\$7F = 1xxx xxxx lecture du 9ème bit
* 2001	Peek (8193)	255 = FF = 1111 1111 lecture 8 bits
* C063	Peek (-16285)	<128 =<\$80 = 0xxx xxxx lecture du 9ème bit
* 2002	Peek (8194)	0 = \$00 = 0000 0000 lecture 8 bits
* C063	Peek (-16285)	>127 =>\$7F = 1xxx xxxx Lecture 9ème bit

Les résultats ci-dessus sont obtenus en lisant la page graphique composée d'après les instructions du tableau 1.

# Les adresses du graphique

Jean-Francois Duvivier

Pour ceux d'entre vous qui désirent explorer dans le détail les entrailles de leur Apple, nous avons rassemblé dans cet article un certain nombre d'adresses utiles relatives au mode graphique. Le tableau 1 vous montre comment, sous moniteur ou en BASIC, passer en mode graphique, aller d'une page à l'autre, ...

Le second tableau indique les zones mémoire utilisées par les divers modes de fonctionnement graphique. Ces zones utilisées par les pages 1 et 2 ne sont pas totalement exploitées par le système. Afin de rendre l'adressage plus simple et plus rapide, le système a laissé quelques trous dont vous pouvez faire ce que vous voulez.

Ces adresses libres sont fournies, pour les graphiques en basse résolution, dans le tableau 3. On peut constater que les huit octets situés après le dernier octet de chaque ligne sont ainsi libres et utilisables à loisir. Une aubaine pour ceux qui sont en permanence à la recherche de place !

Le quatrième tableau montre, pour chaque ligne en basse résolution de la page 1, la localisation de l'octet représentant la première, la 21ème et la dernière colonne de chaque ligne.

Ces adresses apparaissent en hexadécimal dans la partie gauche et en décimal dans la partie droite. L'analyse des adresses permet de constater que l'écran est découpé en trois parties horizontales, et que les adresses cyclent sur ces trois parties. A l'écran, il est d'ailleurs visible que l'apparition d'un graphique se réalise en parallèle, ou presque, sur trois tiers d'écran.

Le tableau 5(A) donne les adresses décimales du début et de la fin de chaque ligne de la page 1 en graphique haute résolution. Comme l'indique le découpage en traits gras, nous retrouvons le regroupement en trois tiers d'écran. A l'intérieur de chaque tiers, ainsi que l'illustrent les traits fins, on remarque des blocs de huit lignes, cette taille ayant bien entendu été choisie pour des raisons d'efficacité d'adressage.

Le lecteur pourra par lui-même identifier les zones mémoire laissées libres par le graphique haute résolution, telles que 8312-8319, ...

Pour la basse comme pour la haute résolution, une simple translation permet de passer des adresses de la page 1 à celles de la page 2.

TABLEAU 1

SOUS MONITEUR	SOUS BASIC	FONCTION
* C050	Poke - 16304,0	Mode graphique
* C051	Poke - 16303,0	Mode Texte
* C052	Poke - 16302,0	Graphique total
* C053	Poke - 16301,0	Graphique Mixte
* C054	Poke - 16300,0	Page 1
* C055	Poke - 16299,0	Page 2
* C056	Poke - 16298,0	Graphique Basse Résolution
* C057	Poke - 16297,0	Graphique Haute Résolution

TABLEAU 2

ADRESSES HEXA	ADRESSES DECIMALES	FONCTION
400 à 7FF	1024 à 2047	Page 1 texte ou graphique B R
800 à BFF	2048 à 3071	Page 2 " "
2000 à 3FFF	8192 à 16383	Page 1 Graphique haute résolution
4000 à 5FFF	16384 à 24575	Page 2 " "

TABLEAU 3

ADRESSES HEXADECIMALES	ADRESSES DECIMALES		
478 - 47F	1144 - 1151	Adresses libres en page 1	
4F8 - 4FF	1272 - 1279		
578 - 57F	1400 - 1407		
5F8 - 5FF	1528 - 1535		
678 - 67F	1656 - 1663		
6F8 - 6FF	1784 - 1791		
778 - 77F	1912 - 1919		
7F8 - 7FF	2040 - 2047		
878 - 87F	2168 - 2175		Adresses libres en page 2
8F8 - 8FF	2296 - 2303		
978 - 97F	2424 - 2431		
9F8 - 9FF	2552 - 2559		
A78 - A7F	2680 - 2687		
AF8 - AFF	2808 - 2815		
B78 - B7F	2936 - 2943		
BF8 - BFF	3064 - 3071		

Tableau 4	COLONNE 1		COLONNE 21		COLONNE 40	
	LIGNES	ADR. HEXA	ADR. DEC	ADR. HEXA	ADR. DEC	ADR. HEXA
0-1	400	1024	414	1044	427	1063
2-3	480	1152	494	1172	4A7	1191
4-5	500	1280	514	1300	527	1319
6-7	580	1408	594	1428	5A7	1447
8-9	600	1536	614	1556	627	1575
10-11	680	1664	694	1684	6A7	1703
12-13	700	1792	714	1812	727	1831
14-15	780	1920	794	1940	7A7	1959
16-17	428	1064	43C	1084	44F	1103
18-19	4A8	1192	48C	1212	4CF	1231
20-21	528	1320	53C	1340	54F	1359
22-23	5A8	1448	58C	1468	5CF	1487
24-25	628	1576	63C	1596	64F	1615
26-27	6A8	1704	68C	1724	6CF	1743
28-29	728	1832	73C	1852	74F	1871
30-31	7A8	1960	78C	1980	7CF	1999
32-33	450	1104	464	1124	477	1143
34-35	400	1232	4E4	1252	4F7	1271
36-37	550	1360	564	1380	577	1399
38-39	500	1488	5E4	1508	5F7	1527
40-41	650	1616	664	1636	677	1655
42-43	600	1744	6E4	1764	6F7	1783
44-45	750	1872	764	1892	777	1911
46-47	700	2000	7E4	2020	7F7	2039



# A TOULOUSE

**APPLE**  


LE PLUS MODULAIRE  
DU DISQUE 5P AU DISQUE DUR  
LE PLUS CONNU

**MICRO ORDINATEUR SHOP  
SOUBIRON SA.**  
9 RUE KENNEDY TOULOUSE  
TEL. 21.64.39./21.04.57.  
TELEX. LPS.INF 521075 F

**LOGICIELS**

PET 2001  
CBM3001  
CBM8001

STOCKS  
PAYE  
FACTURATION  
TRAITEMENT DE TEXTE  
COMPTABILITE GENERALE  
GESTION DES COMMANDES  
BIJOUTIER. MEDICAL. DENTISTE  
GESTION DE FICHIERS  
ASSURANCES

**FORMATION**

STAGES D'INITIATION  
BASIC. PASCAL. FORTRAN

PARTICULIER  
SCIENTIFIQUE  
GESTION

**HP 85**  


DU SCIENTIFIQUE  
AU BUREAU  
D'ETUDE

**VIDEO GENIE  
EG 3003**

DU GENIE  
POUR PETITS  
ET GRANDS

**PCC 2000**  
**ISTC 5000**

- LIBRAIRIE  
- SPECIALISE  
- INFORMATIQUE

**100M<sup>2</sup> D'EXPOSITION**

**S.A.V.**  
DEPANNAGE SOUS 48 H  
REGION MIDI PYRENEES

**SHARP**  
PC 1211  
MZ 80

DU MOINS  
CHER AU  
PARTICULIER

## CHOIX EN MICRO INFORMATIQUE

TABLEAU 5 (A)

ADRESSES DECIMALES

0 8192- 8231	48 8960- 8999	96 8744- 8783	144 8528- 8567
1 9216- 9255	49 9984-10023	97 9768- 9807	145 9552- 9591
2 10240-10279	50 11008-11047	98 10792-10831	146 10576-10615
3 11264-11303	51 12032-12071	99 11816-11855	147 11600-11639
4 12288-12327	52 13056-13095	100 12840-12879	148 12624-12663
5 13312-13351	53 14080-14119	101 13864-13903	149 13648-13687
6 14336-14375	54 15104-15143	102 14888-14927	150 14672-14711
7 15360-15399	55 16128-16167	103 15912-15951	151 15696-15735
8 8320- 8359	56 9088- 9127	104 8872- 8911	152 8656- 8695
9 9344- 9383	57 10112-10151	105 9896- 9935	153 9680- 9719
10 10368-10407	58 11136-11175	106 10920-10959	154 10704-10743
11 11392-11431	59 12160-12199	107 11944-11983	155 11728-11767
12 12416-12455	60 13184-13223	108 12968-13007	156 12752-12791
13 13440-13479	61 14208-14247	109 13992-14031	157 13776-13815
14 14464-14503	62 15232-15271	110 15016-15055	158 14800-14839
15 15488-15527	63 16256-16295	111 16040-16079	159 15824-15863
16 8448- 8487	64 8232- 8271	112 9000- 9039	160 8784- 8823
17 9472- 9511	65 9256- 9295	113 10024-10063	161 9808- 9847
18 10496-10535	66 10280-10319	114 11048-11087	162 10832-10871
19 11520-11559	67 11304-11343	115 12072-12111	163 11856-11895
20 12544-12583	68 12328-12367	116 13096-13135	164 12880-12919
21 13568-13607	69 13352-13391	117 14120-14159	165 13904-13943
22 14592-14631	70 14376-14415	118 15144-15183	166 14928-14967
23 15616-15655	71 15400-15439	119 16168-16207	167 15952-15991
24 8576- 8615	72 8360- 8399	120 9120- 9167	168 8912- 8951
25 9600- 9639	73 9384- 9423	121 10152-10191	169 9936- 9975
26 10624-10663	74 10408-10447	122 11176-11215	170 10960-10999
27 11648-11687	75 11432-11471	123 12200-12239	171 11984-12023
28 12672-12711	76 12456-12495	124 13224-13263	172 13008-13047
29 13696-13735	77 13480-13519	125 14248-14287	173 14032-14071
30 14720-14759	78 14504-14543	126 15272-15311	174 15056-15095
31 15744-15783	79 15528-15567	127 16296-16335	175 16080-16119
32 8704- 8743	80 8488- 8527	128 8272- 8311	176 9040- 9079
33 9728- 9767	81 9512- 9551	129 9296- 9335	177 10064-10103
34 10752-10791	82 10536-10575	130 10320-10359	178 11088-11127
35 11776-11815	83 11560-11599	131 11344-11383	179 12112-12151
36 12800-12839	84 12584-12623	132 12368-12407	180 13136-13175
37 13824-13863	85 13608-13647	133 13392-13431	181 14160-14199
38 14848-14887	86 14632-14671	134 14416-14455	182 15184-15223
39 15872-15911	87 15656-15695	135 15440-15479	183 16208-16247
40 8852- 8871	88 8616- 8655	136 8400- 8439	184 9168- 9207
41 9856- 9895	89 9640- 9679	137 9424- 9463	185 10192-10231
42 10880-10919	90 10664-10703	138 10448-10487	186 11216-11255
43 11904-11943	91 11688-11727	139 11472-11511	187 12240-12279
44 12928-12967	92 12712-12751	140 12496-12535	188 13264-13303
45 13952-13991	93 13736-13775	141 13520-13559	189 14288-14327
46 14976-15015	94 14760-14799	142 14544-14583	190 15312-15351
47 16000-16039	95 15784-15823	143 15568-15607	191 16336-16375

TABLEAU 5 (B)

ADRESSES HEXADECIMALES

0	2000-2027	48	2300-2327	96	2228-224F	144	2150-2177
1	2400-2427	49	2700-2727	97	2628-264F	145	2550-2577
2	2800-2827	50	2B00-2B27	98	2A28-2A4F	146	2950-2977
3	2C00-2C27	51	2F00-2F27	99	2E28-2E4F	147	2D50-2D77
4	3000-3027	52	3300-3327	100	322E-324F	148	3150-3177
5	3400-3427	53	3700-3727	101	3628-364F	149	3550-3577
6	3800-3827	54	3B00-3B27	102	3A28-3A4F	150	3950-3977
7	3C00-3C27	55	3F00-3F27	103	3E2E-3E4F	151	3D50-3D77
8	2080-20A7	56	2380-23A7	104	22A8-22CF	152	2100-21F7
9	2480-24A7	57	2780-27A7	105	26A8-26CF	153	2500-25F7
10	2880-28A7	58	2B80-2BA7	106	2AA8-2ACF	154	2900-29F7
11	2C80-2CA7	59	2F80-2FA7	107	2EA8-2ECF	155	2D00-2DF7
12	3080-30A7	60	3380-33A7	108	32A8-32CF	156	3100-31F7
13	3480-34A7	61	3780-37A7	109	36A8-36CF	157	3500-35F7
14	3880-38A7	62	3B80-3BA7	110	3AA8-3ACF	158	3900-39F7
15	3C80-3CA7	63	3F80-3FA7	111	3EA8-3ECF	159	3D00-3DF7
16	2100-2127	64	2028-204F	112	2328-234F	160	2250-2277
17	2500-2527	65	2428-244F	113	2728-274F	161	2650-2677
18	2900-2927	66	2828-284F	114	2B28-2B4F	162	2A50-2A77
19	2D00-2D27	67	2C28-2C4F	115	2F28-2F4F	163	2E50-2E77
20	3100-3127	68	3028-304F	116	3328-334F	164	3250-3277
21	3500-3527	69	3428-344F	117	3728-374F	165	3650-3677
22	3900-3927	70	3828-384F	118	3B28-3B4F	166	3A50-3A77
23	3D00-3D27	71	3C28-3C4F	119	3F28-3F4F	167	3E50-3E77
24	2180-21A7	72	20A8-20CF	120	23A8-23CF	168	2200-22F7
25	2580-25A7	73	24A8-24CF	121	27A8-27CF	169	2600-26F7
26	2980-29A7	74	28A8-28CF	122	2BA8-2BCF	170	2A00-2AF7
27	2D80-2DA7	75	2CA8-2CCF	123	2FA8-2FCF	171	2E00-2EF7
28	3180-31A7	76	30A8-30CF	124	33A8-33CF	172	3200-32F7
29	3580-35A7	77	34A8-34CF	125	37A8-37CF	173	3600-36F7
30	3980-39A7	78	38A8-38CF	126	3BA8-3BCF	174	3A00-3AF7
31	3D80-3DA7	79	3CA8-3CCF	127	3FA8-3FCF	175	3E00-3EF7
32	2200-2227	80	2128-214F	128	2050-2077	176	2350-2377
33	2600-2627	81	2528-254F	129	2450-2477	177	2750-2777
34	2A00-2A27	82	2928-294F	130	2850-2877	178	2B50-2B77
35	2E00-2E27	83	2D28-2D4F	131	2C50-2C77	179	2F50-2F77
36	3200-3227	84	3128-314F	132	3050-3077	180	3350-3377
37	3600-3627	85	3528-354F	133	3450-3477	181	3750-3777
38	3A00-3A27	86	3928-394F	134	3850-3877	182	3B50-3B77
39	3E00-3E27	87	3D28-3D4F	135	3C50-3C77	183	3F50-3F77
40	2280-22A7	88	21A8-21CF	136	20D0-20F7	184	2300-23F7
41	2680-26A7	89	25A8-25CF	137	24D0-24F7	185	2700-27F7
42	2A80-2AA7	90	29A8-29CF	138	28D0-28F7	186	2B00-2BF7
43	2E80-2EA7	91	2CA8-2CCF	139	2CD0-2CF7	187	2FD0-2FF7
44	3280-32A7	92	31A8-31CF	140	30D0-30F7	188	3300-33F7
45	3680-36A7	93	35A8-35CF	141	34D0-34F7	189	3700-37F7
46	3A80-3AA7	94	39A8-39CF	142	38C0-38F7	190	3B00-3BF7
47	3E80-3EA7	95	3DA8-3DCF	143	3CD0-3CF7	191	3FD0-3FF7

# Des livres pour votre Apple II



## La découverte de l'APPLESOFT

par Dominique Schraen et Frédéric Lévy

Cet ouvrage d'initiation s'adresse aussi bien aux futurs utilisateurs d'Apple voulant apprendre la programmation en Basic Applesoft, qu'à l'Appleophile chevronné, sollicité par ses proches curieux de "voir un peu comment ça marche". D'approche progressive, il est illustré de nombreux exemples et exercices.

Série verte  
**128 Pages - 65 FF**

## La pratique de l'APPLE II

(Volume III)  
par Nicole Bréaud-Pouliquen et Daniel-Jean David

Ce volume est une initiation à la programmation en langage machine 6502, dont le jeu d'instruction est expliqué et utilisé. L'assembleur symbolique et ses logiciels connexes y sont décrits. L'interaction avec le Basic et avec le système y sont étudiés.

Série noire  
**176 Pages - 75 FF**

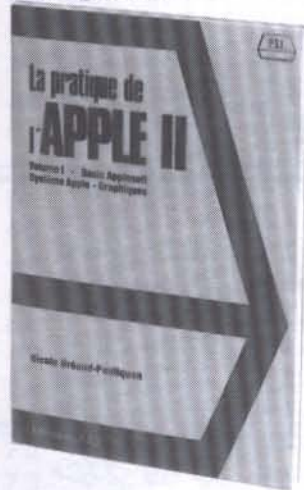


## La pratique de l'APPLE II

(Volume I)  
par Nicole Bréaud-Pouliquen

Cet ouvrage présente les spécificités du Basic Applesoft à partir d'une description du matériel et du logiciel du système Apple. Les techniques de programmation, de composition et d'animation de dessins et graphiques colorés y sont expliquées à l'aide d'exemples illustratifs et d'exercices résolus.

Série bleue  
**128 Pages - 65 FF**

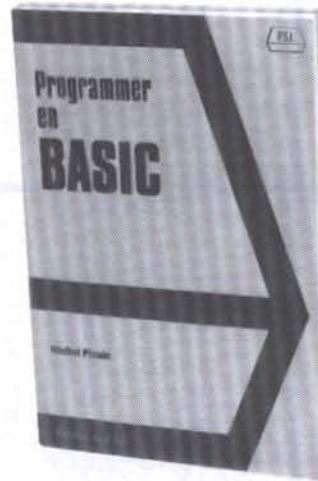


## La pratique de l'APPLE II

(Volume II)  
par Nicole Bréaud-Pouliquen

Ce second volume de la Pratique de l'Apple II est consacré au système d'exploitation disque, à la gestion des fichiers, à l'impression et aux imprimantes, à la carte horloge Applock. De nombreux exemples de programmes illustrent les fonctions et les commandes décrites.

Série rouge  
**120 Pages - 65 FF**



## Programmer en BASIC

par Michel Plouin

Ce livre a été écrit pour les utilisateurs d'ordinateurs individuels en particulier d'Apple II, TRS-80 et PET/CBM. Un répertoire Basic rend son utilisation très pratique et facilite la transposition d'un programme écrit pour un P.S.I. sur un autre.

Série verte  
**132 Pages - 65 FF**

## Mathématiques et statistiques

par Hervé Haut

Cet ouvrage est un recueil de logiciels de base (niveau supérieur) tant en mathématiques qu'en statistiques. Chaque problème traité comporte une introduction numérique, un exposé de la technique de programmation utilisée, un organigramme détaillé et un programme complet en Basic suivi d'un exemple d'utilisation.

Série rouge  
**272 Pages - 85 FF**



**EDITIONS DU P.S.I.**

### BON DE COMMANDE

NOM \_\_\_\_\_  
 RUE \_\_\_\_\_  
 CODE POSTAL \_\_\_\_\_  
 VILLE \_\_\_\_\_

APP 1

Envoyer ce bon accompagné de votre règlement à ÉDITIONS DU P.S.I.  
 41-51, rue Jacquard BP86  
 77400 Lagny s/Marne Tél. (6) 007.59.31.

DÉSIGNATION	NOMBRE	PRIX
Les prix sont : taxes, emballage et port compris (envoi par avion + 5 FF par ouvrage).	<b>TOTAL</b>	





# PLE : le Program Line Editor

Hervé Thiriez

Tous ceux d'entre nous qui suent encore sang et eau pour écrire ou modifier un programme BASIC sur leur Apple peuvent se réjouir : avec le Program Line Editor de Synergistic Software, leur tâche se trouvera considérablement allégée.

## A. PROBLEMES RELATIFS A L'EDITION DE PROGRAMMES EN BASIC.

Analysons tout d'abord les divers problèmes qui engendrent si facilement la frustration lors de l'écriture ou de la modification des programmes :

### 1. Le non-remplissage de l'écran.

Quand une ligne en BASIC est listée sur l'écran, ce qui se passe entre la 34<sup>ème</sup> et la 40<sup>ème</sup> colonne ne répond pas à une logique évidente. Quand on repasse à travers une ligne au-delà de la 33<sup>ème</sup> colonne, on risque de récupérer des blancs imprévus qui dénatureront complètement l'instruction.

### 2. La transparence des fonctions ESC.

Pour modifier une instruction, l'utilisation de la touche ESC, en conjonction avec les touches A-B-C-D ou les touches I-J-K-M est transparente. C'est-à-dire qu'il n'y a pas de différence visible entre les caractères traversés de façon inactive avec la fonction ESC, et ceux écrits ou traversés de façon active avec la flèche à droite.

### 3. Les caractères de contrôle.

Quand on liste un programme à l'écran comme sur l'imprimante, les caractères de contrôle n'apparaissent pas. Il est par conséquent difficile de réaliser des vérifications ou modifications de programmes comportant de tels caractères.

### 4. La longueur des modifications.

Supposez qu'il faille modifier un caractère au début d'une longue instruction. Comment devons-nous procéder ? Il faut lister l'instruction, se déplacer en ESC au début de l'instruction, atteindre en déplacement actif le caractère recherché, le modifier, puis se déplacer de façon active jusqu'à la fin de l'instruction grâce à la flèche de droite et à la touche REPT. C'est long ...

En outre, pour insérer quelques caractères, il faut aussi se positionner au bon endroit, sortir de la ligne avec ESC, taper les caractères à insérer sans oublier que le premier caractère sera écrasé si l'on utilise I-J-K-M, revenir en ESC à la continuation de l'instruction, et enfin se déplacer de façon active jusqu'à la fin de l'instruction. C'est encore plus long ...

### 5. La difficulté du contrôle.

Après avoir modifié une instruction, afin de vérifier si la modification s'est

effectuée comme prévu, il faut de nouveau lister cette instruction, ce qui ne résoud d'ailleurs pas le problème des caractères de contrôle.

#### 6. Les écrasements intempestifs.

Qui d'entre nous n'a jamais, en essayant par exemple de modifier l'instruction 1250, écrasé l'instruction 250 ? En effet, pour peu que le positionnement en ESC ait abouti au "2" et non au "1" du numéro de l'instruction (ce qui est naturel quand on remonte directement à partir du signe " "), un moment d'inattention suffit pour commettre l'irréparable.

Ceci s'aggrave encore si l'on a oublié de lister l'instruction après modification : l'erreur n'apparaîtra alors qu'au moment de l'exécution, et il faudra quelquefois passer un certain temps avant d'en retrouver l'origine.

#### 7. La lourdeur de l'écriture des commentaires.

Pour inclure dans le programme des commentaires se détachant bien, l'idéal est de les faire apparaître en décalage à l'impression, comme le montrent les quelques lignes ci-dessous. Le saut de ligne s'obtient grâce à CTRL-M, mais il faut à chaque commentaire insérer les CTRL-M, et frapper REM et les \*\*\*, tout en positionnant le début des remarques toujours dans la même colonne de façon à obtenir un alignement des remarques ; l'esthétique se joint alors à la lisibilité. L'écriture des remarques en respectant toutes ces contraintes est elle aussi assez longue !

```
1000 REM
```

```
    *** LECTURE DU FICHER ***
```

```
1010 PRINT D$;"OPEN FIC,L60":PRINT D$;"READ FIC,R";I:FOR L = 1 TO 10:INPUT  
A$(I):NEXT
```

```
1020 REM
```

```
    *** FERMETURE DU FICHER ***
```

```
1030 PRINT D$;"CLOSE FIC"
```

#### 8. CAATLOG ou CATALOG

Il arrive souvent que l'on souhaite, en cours d'écriture ou de modification de programme, obtenir le catalogue de la disquette du drive 1 ou du drive 2. Pour ma part, à force de vouloir taper vite sur le clavier, je fais une faute de frappe une fois sur deux quand j'essaye de taper CATALOG. Si vous avez le même problème, vous serez content de disposer d'un moyen simplifié d'obtenir ces catalogues. Cette difficulté peut aussi être tournée grâce à un programme en binaire (voir l'article de C. Guérin) permettant de remplacer l'instruction CATALOG par un caractère isolé tel que "&" ou "C".

#### 9. Lister avec CTRL-S.

La façon normale d'interrompre temporairement une liste se déroulant sur l'écran consiste à taper sur CTRL-S, qui sert dans ce cas d'interrupteur. La nécessité de taper simultanément sur les deux touches n'est pas très pratique. Le Program Line Editor a tourné cette difficulté en donnant cette fonction à la touche ESC lors d'une liste sur écran. De même, il suffit de taper sur ESC et L pour obtenir la

liste complète d'un programme. Il s'agit d'une légère amélioration par rapport à l'instruction LIST.

## B. LE FONCTIONNEMENT DU PROGRAM LINE EDITOR

### 1. Appel d'une instruction

Le fonctionnement de base du P.L.E. consiste à appeler une instruction à l'écran afin de pouvoir ensuite l'éditer. Les principales fonctions d'appel sont les suivantes :

. CTRL-Ennn (Edit) : appel de la ligne nnn pour édition. Cette ligne apparaît sur l'écran, le curseur se positionnant automatiquement sur le premier caractère suivant le numéro de la ligne.

. CTRL-E. (suivi d'un point et bien entendu de RETURN) : appel de la dernière instruction ayant été éditée.

. CTRL-R (Restore) : permet de rééditer l'instruction en cours, en ignorant les modifications que l'on vient d'y apporter.

. CTRL-W : frappé durant l'écriture d'une instruction, réécrit l'instruction et entre dans le mode édition.

Dans tous les cas précédents, les lignes trop longues sont automatiquement resserrées, ce qui permet d'éditer des instructions qui ne seraient plus modifiables en édition normale sans P.L.E.

### 2. Modification d'une instruction

. CTRL-B (Beginning) : positionne le curseur au début de l'instruction.

. CTRL-N (eNd) : positionne le curseur après le dernier caractère de l'instruction.

. déplacement normal grâce aux flèches gauche et droite, qui fonctionnent aussi avec la touche REPT.

. CTRL-Fx (Find) : positionne le curseur à la première occurrence du caractère x. CTRL-Fxxx positionne selon le même principe à la troisième occurrence de ce caractère.

. CTRL-I (Insert) : entrée en mode insertion. Les caractères suivants sont insérés à l'endroit où se trouve le curseur. Le reste de l'instruction se déplace automatiquement.

. CTRL-O (Override) : fonctionne comme CTRL-I, mais accepte comme premier caractère un caractère de contrôle. . CTRL-D (Delete) : efface le caractère couvert par le curseur. Ce mode fonctionne aussi avec la touche REPT.

. CTRL-Zx (Zap) : efface tous les caractères jusqu'à la première occurrence du caractère x. Comme pour CTRL-F, peut être suivi de plusieurs x, afin de tout effacer jusqu'à la nième occurrence.

. CTRL-P (Pack) : resserre toute l'instruction en compactant les blancs non

inclus dans des guillemets.

### 3. Sortie du mode éditeur

Trois façons de sortir du mode éditeur sont offertes :

. CTRL-X : garde ici sa signification habituelle en rendant nulles et non avenues les modifications de la ligne en cours d'édition. Fait apparaître une barre transversale à la fin de l'instruction pour souligner cette annulation, comme en BASIC standard.

. CTRL-M ou RETURN : enregistre l'instruction telle qu'elle apparaît sur l'écran, y compris la partie de l'instruction située éventuellement au-delà du curseur.

. CTRL-Q (Quit) : enregistre l'instruction telle qu'elle apparaît sur l'écran jusqu'à la position du curseur non comprise. Le reste de l'instruction disparaît alors de l'écran.

Remarque : il faut noter que, lorsque l'instruction n'est pas annulée, elle est toujours mémorisée telle qu'elle apparaît sur l'écran. Beaucoup d'erreurs habituelles dues à l'édition courante d'instructions BASIC sont ainsi évitées.

### 4. Les fonctions ESC préprogrammées

Le P.L.E. a programmé un certain nombre de fonctions spécifiques, obtenues en appuyant sur la touche ESC, suivie d'une touche donnée. Ces fonctions simplifient le travail de l'utilisateur en lui donnant rapidement accès à des résultats qui auraient normalement requis une frappe plus longue. Nous en apportons par la suite quelques exemples.

ESC !, A-G, I, J, K, M : ont dans le P.L.E. la même fonction d'édition que d'habitude.

ESC 0 : fait apparaître CALL -936 dans le texte à l'endroit du curseur.

ESC 1 : catalogue de la disquette placée dans le drive 1.

ESC 2 : même fonction, pour le second drive.

ESC / : fait apparaître PRINT dans le texte à l'endroit du curseur.

"ESC :" : revient à faire CALL -151, c'est-à-dire entrer en mode moniteur.

ESC L : revient à frapper LIST, suivi de RETURN.

ESC T : frappe TEXT suivi de RETURN, puis POKE -16300,0 suivi de RETURN. Cette fonction opère donc un retour à la page 1 du texte.

ESC Q : donne le contenu de deux positions mémoires adjacentes commençant dans la position A. Cette fonction revient grosso modo à faire PRINT PEEK(A) + 256\*PEEK(A+1). Ainsi, il faudra par exemple fixer la valeur de A à 105 (resp. 73) préalablement à la fonction ESC Q pour avoir l'adresse de LOMEM (resp. HIMEM) en Applesoft. En INTEGER BASIC, il faudra utiliser A=74 pour LOMEM et A=76 pour HIMEM.

ESC W : fournit l'adresse de début et la longueur du dernier programme BLOADE sur un système 48K. Pour des systèmes plus petits, il faut modifier les paramètres de

cette fonction grâce à la programmation de fonctions ESC (voir section suivante).

#### 5. La programmation de fonctions ESC

Chacune des fonctions ESC ci-dessus peut être reprogrammée à loisir. En outre, l'utilisateur peut inventer les fonctions ESC qu'il désire en association avec n'importe quel autre caractère.

On pourra ainsi faire intégrer dans une instruction les PEEKs, POKEs et CALLs dont on se sert régulièrement. Des fonctions plus complexes sont aussi possibles, telles que programmer ainsi le lancement de l'imprimante, la fusion en APPEND, la sauvegarde du programme sur une disquette, ...

Cette présentation n'avait pas l'exhaustivité pour objectif et nous avons volontairement exclu certains éléments afin de mieux nous concentrer sur l'essentiel. Nous conseillons au lecteur de bien étudier la documentation fournie avec le programme avant de l'utiliser.

L'impression générale que l'on retient, une fois familiarisé avec le Program Line Editor est la suivante : "Mais comment donc pouvais-je me débrouiller sans utiliser le P.L.E. ". Bien entendu, ce programme n'est pas la panacée universelle ; il s'agit d'un éditeur de ligne et non pas d'un éditeur de programme, c'est-à-dire un éditeur de texte dont la sortie serait un programme exécutable. Mais, dans le cadre de l'objectif qu'il s'est fixé, c'est un programme bien conçu, bien réalisé et bien documenté.

#### C. LE CHARGEMENT DU P.L.E. EN MEMOIRE

Les utilitaires P.L.E., C.R.A.E. (voir pages 46 et 47), et FID, dont nous parlerons ultérieurement, sont tous compatibles en mémoire. La distribution des zones mémoire en Applesoft, quand ces trois utilitaires sont chargés simultanément, est la suivante :

49151 (\$BFFF)	DOS
40192 (\$9D00)	Program Line Editor
38656 (\$9700)	Buffers du DOS
36864 (\$9000)	C.R.A.E.
29696 (\$7400)	Programme et variables
16384 (\$4000)	Haute résolution page 1
08192 (\$2000)	1454 octets disponibles
06738 (\$1A51)	F.I.D.
02048 (\$0800)	Basse résolution et écran texte
00000 (\$0000)	

# CRAE : Co-Resident Applesoft Editor

Hervé Thiriez

Ce programme est un produit récent (1980) de la société Highlands Computer Services. Il s'agit, comme pour le P.L.E., d'un programme d'assistance à l'édition de programmes. Il peut fonctionner indépendamment de celui-ci, ou conjointement avec lui. Nous limiterons notre analyse à cette dernière hypothèse, puisque nous avons précédemment reconnu la grande utilité du P.L.E.

C.R.A.E. apporte un complément utile aux fonctions du P.L.E., en autorisant en particulier la recherche ou la modification d'une chaîne de caractères donnée, la numérotation automatique et le dump hexadécimal/ASCII d'une zone de mémoire sélectionnée. Pour rentrer en mode C.R.A.E., il faut taper le signe "&", suivi de RETURN. Le prompt " " est alors remplacé par " ". En frappant RETURN à vide, on récupère le prompt Applesoft, indispensable pour l'écriture en Applesoft comme pour l'utilisation des instructions du P.L.E.

## Liste des instructions

A : APPEND d'un programme sur disquette au programme en mémoire. Le second programme se met à la suite du premier, ce qui pose des problèmes si les numéros d'instructions du premier programme ne sont pas tous inférieurs à ceux du second. Il paraît plus raisonnable d'utiliser à cet égard le RENUMBER fourni avec le système Apple. Cela prend un peu plus de temps, mais offre plus de sécurité.

C/ch1/ch2/n1,n2 : remplace la chaîne ch1 par la chaîne ch2 à la première occurrence dans chaque ligne de n1 à n2. Ce remplacement s'effectue seulement dans les cas où la chaîne ch1 n'est pas placée entre guillemets.

C/ch1/ch2/An1,n2 : le remplacement s'effectue dans ce cas pour toutes les chaînes ch1 non incluses entre guillemets dans les lignes de n1 à n2.

D : le programme demande l'adresse de début et l'adresse de fin du dump. Il fournit un dump hexadécimal formaté en matrice avec écho ASCII, pour la zone mémoire placée entre les deux adresses.

F/ch1/n1,n2 : toutes les instructions entre n1 et n2 et contenant la chaîne ch1 hors guillemets sont listées à l'écran. Là, comme dans tout les cas où C.R.A.E. liste des instructions, celles-ci sont resserrées pour éliminer les blancs, ce qui ne simplifie ni la lecture, ni le contrôle syntaxique.

Ln1,n2 : liste les instructions allant de n1 à n2.

Mnnn : permet de modifier l'instruction nnn. Ceci ne présente pas d'intérêt pour nous ici, les possibilités du P.L.E. en matière d'édition d'instruction étant nettement plus avantageuses.

Nn1,i : numérotation automatique en débutant par le numéro d'instruction n1 et en utilisant un incrément de i.

Qn1,n2,n3,i : duplique le bloc d'instructions comprises entre n1 et n2 à partir de n3 avec un incrément de i. Tous les numéros d'instructions cités dans des GOTOs ou des GOSUBs dans le bloc de départ gardent leur valeur initiale dans le bloc d'arrivée. Il ne semble pas y avoir de nombreuses utilisations pour une fonction de ce type.

Rn1,n2,n3 : renumérote les instructions de n1 à n2 avec un incrément de i et en commençant par le numéro n3. Il est regrettable que des syntaxes différentes aient été utilisées pour les fonctions Q et R ; cela augmente de risque d'erreur dans la manipulation.

QUICK AND DIRTY : sous ce nom sont regroupées cinq fonctions simples.

\* : des instructions moniteur peuvent être introduites à la suite du signe \*.

! : imprime le nombre d'octets disponibles entre la fin du programme et HIMEM.

% : renvoie la longueur en décimal et l'adresse de début du dernier programme ayant fait l'objet d'un BLOAD. Cette fonction est redondante avec la fonction ESC W du P.L.E.

!XXXX : convertit en décimal le nombre hexadécimal XXXX (0000-FFFF).

%XXXX : convertit en hexadécimal le nombre décimal XXXX (0-65535).

### Conclusions

C.R.A.E. est un honnête programme de complément pour le P.L.E. Il faut toutefois noter que les deux fonctions les plus intéressantes pour la programmation en BASIC, la recherche et la modification de chaînes, existent et sont mieux traitées dans le programme Apple Doc, dont nous ferons l'analyse dans le prochain numéro.

L'avantage du C.R.A.E. , relativement à Apple Doc, tient au fait qu'il réside en mémoire pendant que l'on met un programme au point. Toutes ses fonctions sont par conséquent immédiatement disponibles.

En conclusion, C.R.A.E. est un complément intéressant du P.L.E., mais il ne représente pas un aussi grand caractère de nécessité. Pour ma part, la fonction que j'utilise le plus souvent dans la mise au point de programmes en BASIC est la recherche de chaînes.

---

### Obtenez le mode moniteur grâce au RESET

En venant du BASIC, vous devez faire CALL -151, suivi de RETURN, pour entrer en mode moniteur. Si vous souhaitez gagner du temps, mettez-vous en mode moniteur et tapez : 3F2 : 69 FF 5A <RETURN>

Il suffira alors de taper sur RESET pour obtenir le mode moniteur. Le nouveau Reference Manual vous montre (pp. 36 et 37) comment vous brancher sur votre programme assembleur avec un RESET, pour ceux bien entendu qui possèdent Applesoft en ROM.

## Des vers dans la pomme ...

Alexandre Duback

### DES BOUCLES A S'ARRACHER LES CHEVEUX !

Essayez de faire fonctionner le programme suivant; vous constaterez qu'il fonctionne parfaitement bien, puisqu'il ne fait rien de mal et qu'on ne lui en demande pas plus.

```
10 FOR I=1 TO 3 : IF I=2 THEN 30
20 NEXT I
30 REM DEBUT DES BOUCLES DE TEST
40 FOR I=1 TO 3 : FOR J=1 TO 2 : NEXT J : NEXT I
50 FOR J=1 TO 2 : FOR I=1 TO 3 : NEXT I : NEXT J
```

Otez maintenant l'instruction 40, et relancez le programme. Vous vous apercevrez alors avec stupeur que votre Apple favori vous envoie une injure à la figure :  
NEXT WITHOUT FOR ERROR IN 50

Cela provient d'un bug rencontré dans la plupart des interpréteurs BASIC de Microsoft: cette erreur survient quand une sortie de boucle sur un indice (ici, I=2) est suivie d'une boucle imbriquée dans laquelle cet indice est interne.

Conclusion : si vous aimez les sorties de boucles, modifiez-les de la façon dont nous le faisons ci-dessous pour la ligne 10 :

```
10 FOR I=1 TO 3 : IF I=2 THEN I=3 : NEXT I : GOTO 30
```

### REPREZ VOTRE APPEND, SI CE N'EST DEJA FAIT (DOS 3.2 ET 3.2.1)

Dans les versions 3.2 et 3.2.1, le DOS ne met pas de marqueur de fin de fichier sur une disquette quand il ferme un fichier. Il se contente de remplir les nouveaux secteurs de marqueurs de fin de fichier ; de cette façon, une information jointe par APPEND est normalement suivie d'un tel marqueur. Par contre, quand le dernier caractère se trouve en fin de secteur, le DOS ne remplit pas le secteur à suivre de marqueurs. A l'APPEND suivant, n'en trouvant pas, il recommence par défaut au début de fichier et écrase ainsi les informations antérieures.

La correction consiste à ajouter un marqueur après chaque ordre d'écriture, comme le fait le programme en cinq octets indiqué ci-dessous. Si vous utilisez déjà les adresses de 768 à 772, vous pouvez bien entendu déplacer ce programme à loisir.

```
10 POKE 768,169 : POKE 769,0 : POKE 770,76 : POKE 771,237 : POKE 772,253
20 REM VOICI COMMENT IL FAUT UTILISER CE PROGRAMME
30 D$=CHR$(4) : PRINT D$"APPEND nom" : PRINT D$"WRITE nom"
40 PRINT "données"
50 CALL 768 : REM ADJONCTION DU MARQUEUR
60 PRINT D$"CLOSE nom"
```

Cette correction est indiquée dans le Apple Answer Book.



SIDEG

# SIDEG INFORMATIQUE

170, rue Saint-Charles, 75015 PARIS. Tél. : 557.79.12

Ouvert du lundi au samedi de 9 h 30 à 18 h 30 sans interruption

Vente sur place et par correspondance — Commande par téléphone — Crédit — Leasing —  
Détaxe à l'exportation — Carte Bleue ou Visa

## Déménagement

A compter du 8 septembre 1981, nous vous accueillerons 170, rue Saint-Charles à PARIS 15<sup>e</sup>.  
A cette occasion, nous vous offrons une remise exceptionnelle de 5 % pour toute commande de matériel : ordinateur ou périphérique informatique, à valoir sur l'achat de logiciels, librairie ou fournitures diverses.  
Valable du 8 septembre au 7 novembre 1981.

# APPLE II plus

8495 F T.T.C.  
1 APPLE II - Plus 16 K  
1 Modulateur T.V.  
10 cassettes

9495 F T.T.C.  
1 APPLE II - Plus 16 k  
1 Moniteur  
10 cassettes

13995 F T.T.C.  
1 APPLE II - Plus 32 K  
1 Modulateur T.V.  
Mini-Disk II avec contrôleur

13995 F T.T.C.  
1 APPLE II - Plus 32 K  
1 Moniteur  
1 Mini-Disk II avec contrôleur

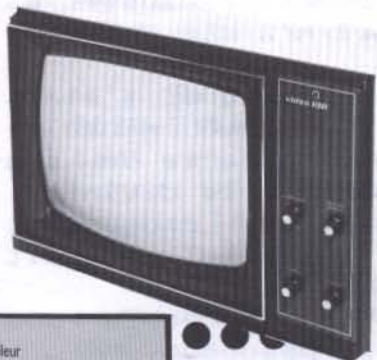
14495 F T.T.C.  
1 APPLE II - Plus 48 k  
1 Moniteur  
1 Mini-Disk avec contrôleur

19495 F T.T.C.  
1 APPLE II - Plus 48 K  
1 Moniteur  
1 Mini-Disk II avec contrôleur  
1 Imprimante OKI 80  
1 Interface parallèle APPLE

24495 F T.T.C.  
1 APPLE II - Plus 48 K  
1 Moniteur  
1 Mini-Disk II avec contrôleur  
1 Mini-Disk II sans contrôleur  
1 Imprimante MX-80 FT  
1 Interface parallèle APPLE



Exemple d'achat à crédit pour un montant de :  
**9 495 F T.T.C.**  
versement comptant 1 945 F  
+ 24 mensualités de : 421,16 F T.T.C.



## Moniteurs - Vidéo

Vidéo 100 écran N 18 31 cm	1 450 F T.T.C.
Vidéo 100 écran vert 31 cm	1 000 F T.T.C.
Vidéo OPC écran vert 9 pouces	1 250 F T.T.C.
Vidéo Tono écran vert 31 cm	1 950 F T.T.C.
Vidéo SANNIO écran vert 31 cm	2 300 F T.T.C.
Téléviseur couleur équipé prise P émette (42 cm)	4 000 F T.T.C.

## Extensions spécialisées

Barcode	1 820 F T.T.C.
Joystick	550 F T.T.C.
Clavier numérique	850 F T.T.C.
Carte Pascal	3 195 F T.T.C.
Carte Integer	1 400 F T.T.C.
Carte Parallel	1 400 F T.T.C.
Z-80 Software Microsoft	2 995 F T.T.C.
Ramcard 16 K Microsoft	1 850 F T.T.C.
Carte R.Y.B. 16 couleurs et H.R.	1 535 F T.T.C.
Eprom Programmer	1 100 F T.T.C.
ROM Plus	1 000 F T.T.C.
ETC.	

## Logiciels

Apple Writer	650 F T.T.C.
Visicalc	1 100 F T.T.C.
CCA Data Management System	600 F T.T.C.
Fortran Microsoft	1 050 F T.T.C.
Cobol Microsoft	4 950 F T.T.C.
Apple Plot	1 400 F T.T.C.
List 1.0	405 F T.T.C.
Mystery House	250 F T.T.C.
Raster Blaster	275 F T.T.C.
Space Eggs	175 F T.T.C.
Autobahn	175 F T.T.C.
FSJ Flight Simulator	180 F T.T.C.
ETC.	

## Librairie

La découverte de l'Apple II	65 F T.T.C.
La pratique de l'Apple III (Vol. 1)	65 F T.T.C.
La pratique de l'Apple II (Vol. 2)	65 F T.T.C.
Pascal sur Apple II	100 F T.T.C.
Basic Apple-soft	100 F T.T.C.
Programmation de 6502	98 F T.T.C.
Applications de 6502	98 F T.T.C.
Call Apple Magazine	30 F T.T.C.
Nibble Magazine	30 F T.T.C.
Micro 6502 Magazine	30 F T.T.C.
ETC.	

Mini-Disk II avec contrôleur  
DOS 3.3 ..... 4 600 F T.T.C.  
Mini-Disk sans contrôleur  
DOS 3.3 ..... 3 400 F T.T.C.  
Double lecteur de disque avec contrôleur  
DOS 3.3 ..... 7 995 F T.T.C.



## Imprimantes

Imprimante SEIKO GP-80M avec interface Apple ..... 3 500 F T.T.C.  
Imprimante OKI Microline 80 avec interface ..... 6 000 F T.T.C.  
Imprimante OKI Microline B3 avec interface ..... 10 700 F T.T.C.  
Imprimante EPSON MX-80 avec interface ..... 6 000 F T.T.C.  
Imprimante EPSON MX-80 FT avec interface ..... 7 000 F T.T.C.  
Imprimante Diablo 630 avec introducteur frontal et interface ..... 19 000 F T.T.C.  
ETC...



## Dossier CRÉDIT

- 1) Précisez le montant total T.T.C. de votre commande.
- 2) Spécifiez le versement comptant que vous voulez effectuer (minimum 20 %).
- 3) Indiquez le montant approximatif des mensualités que vous souhaitez régler.

« Le respect de ces formalités nous permettra de mieux vous servir et d'accélérer l'acceptation de votre dossier. »

**Conditions :** Être salarié depuis plus d'un an.

Les prix sont donnés à titre indicatif et peuvent être modifiés sans préavis.

## Notre nouveau catalogue général 1981/82.

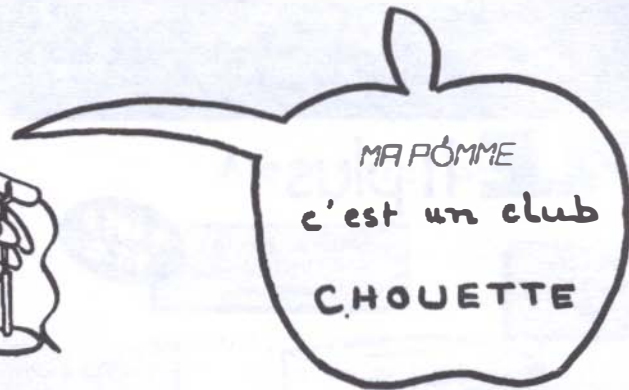
Je désire recevoir gratuitement le catalogue SIDEG 81-82 :

Nom ..... Prénom .....

Adresse complète .....



concept : 302.49.65



# MA PÔMME

Plus Jeune, j'ai toujours été stupéfait et très impressionné par le comportement passionné, exclusif et religieux de quelqu'un et quelqu'une qu'il m'a été donné de côtoyer.

J'ai en mémoire le souvenir de cet être qui s'était payé un train électrique, et croyez-moi, pas un minable, pas une copie ! Passion dévorante, toute une aile de son appartement était consacrée à l'encensement de la divinité. Il y avait des rails partout et les convois se croisaient dans un chaos dément qu'Astaroth et lui seul pouvait orchestrer.

J'ai beaucoup plaint cet être ; pourtant, depuis que j'ai acquis mon Apple II, je lui ressemble beaucoup.

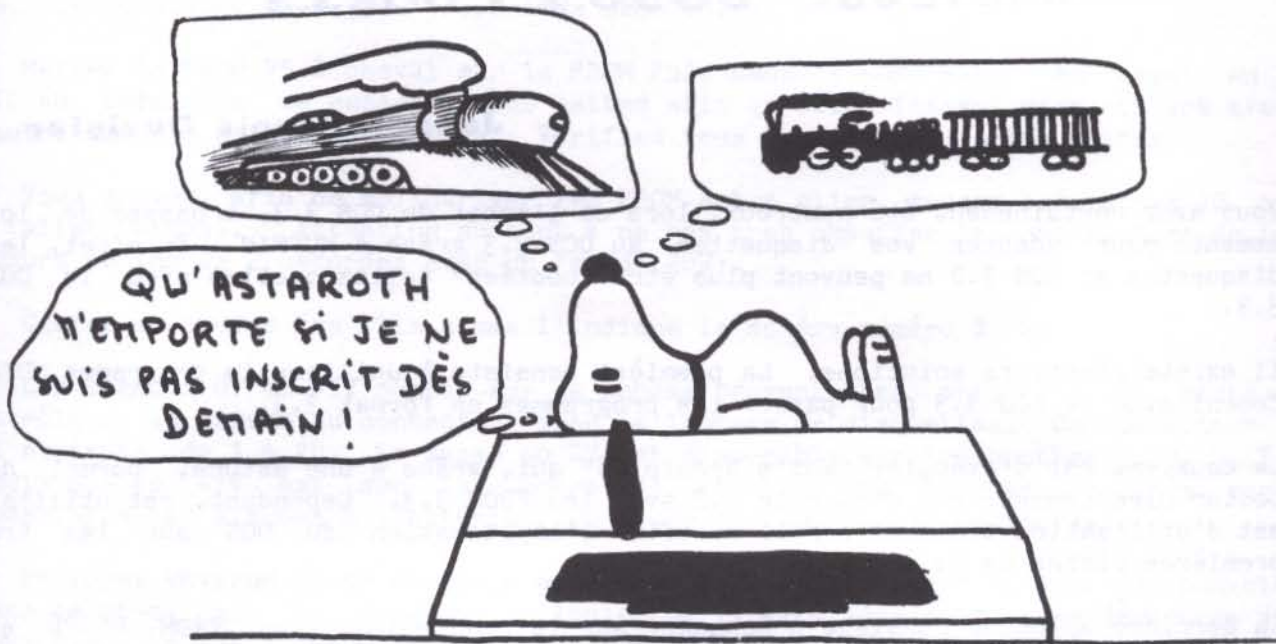
Mais, ne vous y trompez pas, bien que déesse Micro-informatique soit plus exigeante que ces anciennes divinités sauvages de notre préhistoire, réclamant temps et développement, je suis désormais comme un Faust moderne, mais moi, je n'ai pas eu à vendre mon âme au diable.

Allez, croissez et multipliez, annoncez ce nouvel Évangile à la terre entière et joignez

# MA PÔMME

6 rue Paul Saurière

75016 PARIS



MA FOMME est une association sans but lucratif, régie par la loi de Juillet 1901, qui a pour but de regrouper les utilisateurs d'informatique individuelle.

Sa section Apple (et matériels compatibles : ITT 2020, Silex...) offre de nombreux services, le plus important étant certainement les réunions régulières, où vous aurez peut-être l'occasion de rencontrer quelqu'un qui a résolu le problème sur lequel vous sechez depuis un mois !

Le bulletin interne, la documentation du club, la possibilité d'achats groupes, les conférences, les liens que vous pourrez créer avec d'autres membres sont autant d'arguments qui vous décideront, peut-être à venir nous voir.

Si vous êtes intéressé(e), remplissez le bulletin ci-dessous et envoyez-le-nous au plus vite !

Bulletin à recopier ou à découper  
et à renvoyer à MA FOMME

6 rue Paul Saunière - 75016 PARIS

Veuillez m'envoyer une documentation sur le club MA FOMME.

Veuillez m'indiquer la date de vos prochaines réunions.

NOM :

Prénom :

Adresse :

Matériel possédé :

# Inverseur DOS 3.2 - DOS 3.3

Jean-François Duvivier

Vous avez certainement été nombreux, lors de l'achat du DOS 3.3, à passer de longs moments pour adapter vos disquettes au DOS 3.3 grâce à MUFFIN. En effet, les disquettes en DOS 3.2 ne peuvent plus être "bootées" telles quelles avec le DOS 3.3.

Il existe plusieurs solutions. La première consiste à utiliser le programme MUFFIN fourni avec le DOS 3.3 pour passer les programmes en format 3.3.

La deuxième est d'exécuter "Woz's Update 16" qui, grâce à une astuce, permet de booter directement une disquette 3.2 avec les PROM 3.3. Cependant, cet utilitaire est d'utilisation délicate car il modifie l'implantation du DOS sur les trois premières pistes de la disquette.

La dernière solution consiste à échanger sur le contrôleur les PROM P5-P6 et P5A-P6A chaque fois que vous voulez changer de DOS. Cette solution vous a certainement fait sourire, tant elle paraît longue et peu souple !

Et pourtant, il est possible de le faire d'une façon simple, sans ouvrir l'Apple, en poussant seulement un bouton. La modification effectuée sur la carte contrôleur nécessite :

- les PROM P5, P5A et P6A (la PROM P6 n'est pas nécessaire).
- un inverseur double et du fil électrique de petit diamètre.
- un fer à souder de faible puissance.

Bien entendu, quand vous pratiquez une modification matérielle, vous perdez votre garantie. Alors, suivez attentivement le déroulement des opérations et ne commencez que si vous vous sentez capable de tout faire.

## Réalisation

1. Enlevez les 2 PROM actuellement sur votre contrôleur.
2. Sur les trois PROM (P5, P5A et P6A), relevez la patte numéro 15 de telle sorte qu'elle ne puisse plus faire contact avec le support, selon le schéma ci-dessous.

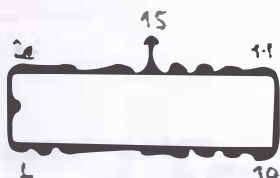


Schéma 1  
Brochage des PROM

3. Remplacez les PROM P5A et P6A en faisant attention au sens (tous les circuits du

contrôleur doivent avoir leur repère du même côté).

4. Mettez la PROM P5 à cheval sur la PROM P5A, dans le même sens, en ayant eu soin au préalable de resserrer ses pattes afin qu'elles fassent bien contact avec l'autre mémoire située en dessous. Vérifiez tous les contacts entre pattes.

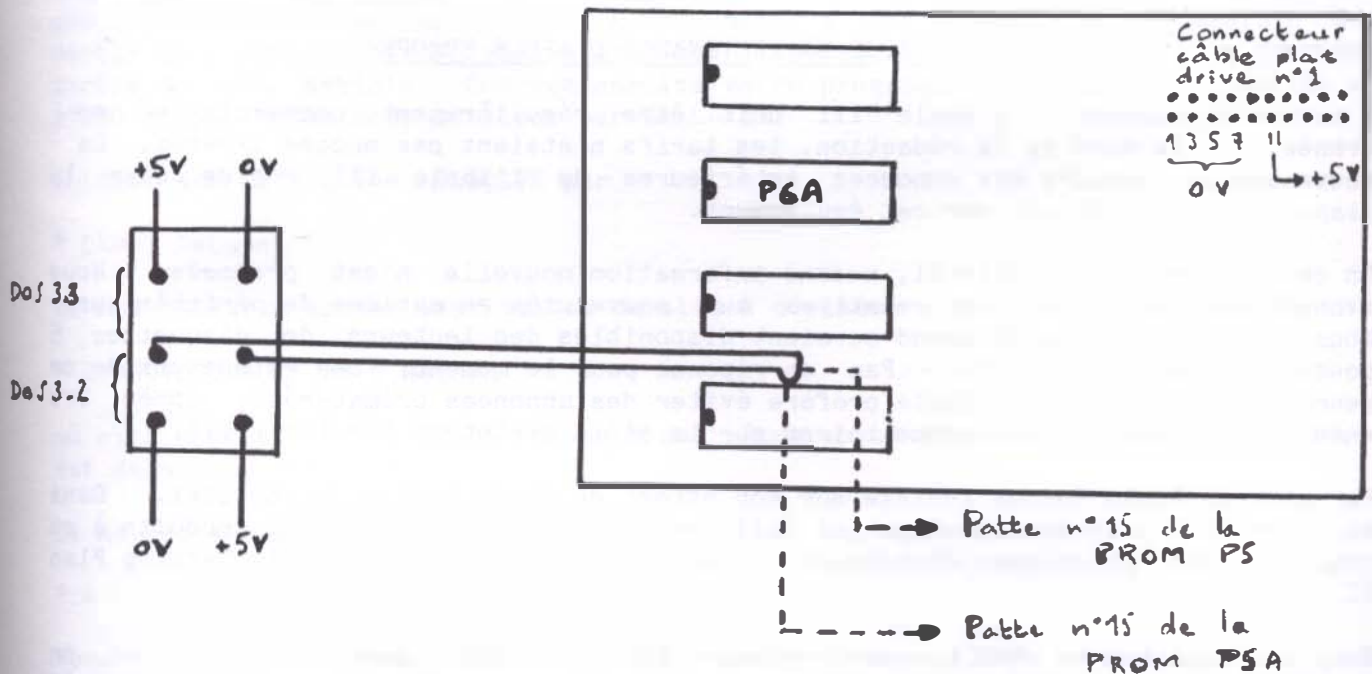
5. Vous pouvez, afin de solidariser les PROM entre elles, souder les pattes 10 et 20 (alimentation). Attention surtout à ne pas trop chauffer les pattes lors de la soudure (moins de 2 secondes avec un fer de 30W).

6. Cablez et soudez les fils comme l'indique le schéma numéro 2.

7. Les signaux 0V et +5V portés sur ce schéma peuvent être pris sur la carte contrôleur, au niveau du connecteur avec le lecteur de disquettes. Ce connecteur est numéroté de 1 à 20. La masse ou 0V est disponible sur les broches 1, 3, 5, 7, tandis que le +5V est sur la broche 11 (toutes les broches impaires sont accessibles car situées à l'extérieur du coude fait par le connecteur).

8. Prévoyez environ 20 cm de câble entre la carte et l'inverseur, afin de pouvoir fixer celui-ci sur le fond de l'Apple où il sera accessible sans démontage du couvercle. Le câble devra être bien fixé sur la carte afin d'éviter tout court-circuit.

Pour booter une disquette 3.3 ou 3.2, il suffira alors de placer l'inverseur dans la position correspondante.



**schéma n°2**  
**câblage**

## Remarques générales

1. Vous pouvez manoeuvrer l'inverseur lorsque l'appareil est allumé. Cependant, il faut bien entendu s'abstenir de le faire lorsque le lecteur de disquettes fonctionne.

2. Pour déterminer, à un moment donné, si vous êtes en DOS 3.2 ou DOS 3.3, mettez vous en mode moniteur, puis tapez C600L et RETURN. Vous faites ainsi apparaître les premiers octets du DOS, respectivement :

A2 20 A0 00 A9 03 85 3C 18 88 pour le DOS 3.2

A2 20 A0 00 A2 03 86 3C 8A 0A pour le DOS 3.3

## Remarques Techniques

1. La patte 15 des mémoires PROM 6309-1 (ou de ses équivalents : 74S471 ou 18S22N) est un chip select. Il est mis à zéro sur la carte contrôleur et donc la PROM est toujours sélectionnée.

2. Lorsque les PROM P5 et P5A sont installées l'une sur l'autre, la sélection est faite en appliquant un signal 0 sur le chip select de la PROM souhaitée, tandis que le chip select de l'autre PROM est mis à 1.

3. Le câblage sur l'inverseur se réalise de telle sorte que les PROM P5 et P5A ne puissent être sélectionnées simultanément. La PROM P6A est "universelle" et convient aussi bien pour le DOS 3.2 que pour le DOS 3.3. Il n'y a donc pas besoin d'utiliser la PROM P6.

---

## INFORMATIONS EN PROVENANCE D'APPLE EUROPE

A partir de septembre, l'Apple III doit être régulièrement commercialisé en France. A la date de la rédaction, les tarifs n'étaient pas encore arrêtés. La nouveauté par rapport aux annonces antérieures de l'Apple III réside dans la disponibilité du Pascal sur cet équipement.

En ce qui concerne l'Apple II, aucune information nouvelle n'est proposée. Nous avons posé des questions relatives aux nouveautés en matière de périphériques. Nous avons ainsi demandé quand seraient disponibles des lecteurs de disquettes 5 pouces à grande capacité. Pas de réponse pour le moment; des extensions de ce genre sont prévues, mais Apple préfère éviter des annonces prématurées, après les ennuis provoqués par des commentaires sur la série prototype des Apple III.

Par contre, Apple Europe insiste sur son effort de distribution de logiciel. Dans ce cadre sont proposés des "Special delivery" sous le nom d'Apple (26 produits à ce jour) et des programmes distribués par Apple, tels que Visicalc III, Desktop Plan II, ...

Nous espérons tester dans un avenir proche AGENDA FILES (gestion de carnet de rendez-vous), HAND HOLDING BASIC (apprentissage interactif du langage) et PLAN 80 qui doit, selon sa description, réunir et améliorer les possibilités conjointes de Visicalc et Desktop Plan.

Les utilisateurs désirant réaliser des progiciels sur Apple III trouveront auprès de Apple Europe une oreille attentive.

# Déplacement de programmes en assembleur

Jean-Francois Duvivier

Avez-vous déjà entendu parler de la ROM Programmer's Aid £1, fournie avec la carte Integer ? Il s'agit d'une ROM mise en \$D000-\$D800, zone laissée libre par l'Integer.

Elle rassemble un certain nombre d'utilitaires écrits en assembleur : graphique haute résolution, APPEND, renumérotation, test RAM, vérification cassette et surtout une routine RELOCATE qui m'a mis l'eau à la bouche dès que je l'ai aperçue. Voilà en effet le programme que je cherchais depuis plusieurs mois pour déplacer très facilement n'importe quel programme en assembleur.

En effet, la commande du moniteur MOVE permet de déplacer de l'assembleur mais ne change pas les adresses des JSR et des JMP rencontrés lorsqu'ils renvoient à l'intérieur du programme. Il faut donc remettre à jour le programme après le MOVE, ce qui se révèle un exercice redoutable pour les nerfs lorsque l'encombrement du programme se chiffre en kilo-octets !

## Utilisation de la routine RELOCATE

Supposons donc que vous vouliez déplacer un programme normalement situé de \$1000 à \$1DFE pour le faire débiter en \$2100. Rien de plus simple ! Assurez-vous d'abord que vous êtes bien en Integer car la Programmer's Aid n'est pas accessible à partir de l'Applesoft. Si vous n'avez que l'Applesoft, reportez-vous à la deuxième partie de cet article. Chargez ensuite votre programme en assembleur à partir de \$1000 et passez en mode moniteur par CALL-151.

Initialisez la routine RELOCATE par :

```
* D4D5G Return
```

Il faut ensuite donner les limites du bloc à déplacer ainsi que la destination :

```
* 2100<1000.1DFE cY * Return
```

où cY signifie control Y (la touche Y doit être appuyée alors que la touche CTRL est maintenue enfoncée).

Il ne reste plus à faire que la translation par la commande :

```
* 2100<1000.1DFE cY Return
```

Vous pouvez alors vérifier que le programme débute en \$2100. De plus, chaque JSR ou JMP qui avait dans le programme original une adresse comprise entre \$1000 et \$1DFE a été recalculé.

L'utilisation combinée de RELOCATE et du MOVE du moniteur permet également le

transfert de blocs en assembleur dans lesquels sont imbriquées des données. Supposons que le programme précédent comprenne deux zones de données, de \$12F1 à \$132E, et de \$1A4C à \$1B20.

Les zones de données ne doivent pas être analysées par RELOCATE mais simplement transférées telles quelles.

Les commandes sont alors les suivantes :

```
* D4D5G Return                Initialisation
* 2100<1000.1DFE cY * Return  Définition des limites
* 2100<1000.12FO cY Return
* 23F1<12F1.132E M Return      1ère zone de données
* 242F<132F.1A4B cY Return
* 2B4C<1A4C.1B20 M Return      2ème zone de données
* 2C21<1B21.1DFE cY Return
```

Il est possible d'utiliser des commandes abrégées, ce qui évite de calculer toutes les adresses ci-dessus. On obtient alors :

```
* D4D5G Return
* 2100<1000.1DFE cY * Return
* 2100<1000.12FO cY Return
* .132E M Return
* .1A4B cY Return
* .1B20 M Return
* .1DFE cY Return
```

Après ces quelques exemples, vous voici donc à même d'utiliser cet utilitaire. Cependant, votre programme déplacé ne tournera peut-être pas du premier coup, car il y a quelques limitations. La principale tient au fait que les adresses chargées en mode immédiat ne sont pas modifiées. Par exemple, le programme suivant :

```
900 : LDA £$0D
902 : STA $7FE
905 : LDA £$08
807 : STA $7FF
80A : JSR (7FE)
80D : Suite du programme
```

ne marchera pas après translation car le LDA £0D et le LDA £08 ne sont pas modifiés par RELOCATE. Il faudra donc les modifier à la main. Malgré cette limitation inévitable, RELOCATE peut vous rendre de nombreux services.

La routine RELOCATE est située en ROM et ceux d'entre vous qui ont un Apple II Plus, sans carte Integer ou carte Langage ne peuvent y avoir accès. Ma première solution a été de transférer la routine pour la mettre en RAM (en \$300 par exemple), afin de pouvoir l'utiliser sans ROM. Tous les appels au moniteur sont identiques entre le moniteur Integer et le moniteur Autostart de l'Apple II Plus. Pas de problème donc de ce côté. Malheureusement, la routine RELOCATE fait appel au Sweet16 qui est un pseudo-langage pour l'Apple et dont l'interpréteur n'est disponible qu'avec l'Integer Basic.

La solution finale a donc été de comprendre les fonctions réalisées par le code Sweet16 et de les réécrire en assembleur 6502. Ceux qui ne sont pas familiarisés avec le Sweet16 peuvent se reporter aux articles parus dans :

- BYTE de novembre 1977
- La rubrique l'Apple épluché de l'O.I. dans les numéros 24 et 25.



Le code Sweet16 utilisé est le suivant :

```

22      RO=R2
D6      Compare RO et R6
02 06   Branchement à la fin si RO < R6
26      RO = R6
B1      RO = RO - R1
02 02   Branchement à la fin si R1 > RO
A4      RO = RO + R4
36      R6 = RO
00      Fin et retour au programme 6502

```

Les registres R0 à R15 ont une longueur de 16 bits (2 octets consécutifs). Ils sont situés en page 0 aux adresses 00 à 0F.

Le tableau 1 liste la routine assembleur utilisée pour remplacer le Sweet16.

Le tableau 2 donne l'ensemble de la routine modifiée à partir de l'adresse \$300. L'initialisation de la routine se fait alors par la commande 300G. Le reste n'est pas modifié.

Les octets cerclés sont ceux que vous devrez modifier si vous déplacez la routine pour la mettre ailleurs qu'en \$300. A noter que si vous la déplacez en l'utilisant elle-même, les deux premiers octets devront cependant être modifiés car ils rentrent dans le cas vu en fin de la première partie, où les adresses sont chargées par des LDA.

```

0800      1 ; *****
0800      2 ; * *
0800      3 ; * ADDITIF A LA ROUTINE *
0800      4 ; * *
0800      5 ; * RELOCATE *
0800      6 ; * *
0800      7 ; * PAR J-F DUVIVIER *
0800      8 ; * *
0800      9 ; * 20 JUILLET 1981 *
0800     10 ; * *
0800     11 ; *****
0800     12 ;
0800     13 ;
0357     14      ORG $357
0357     15 R0L   EPZ $0
0357     16 R0H   EPZ $1
0357     17 R1L   EPZ $2
0357     18 R1H   EPZ $3
0357     19 R2L   EPZ $4
0357     20 R2H   EPZ $5
0357     21 R4L   EPZ $6
0357     22 R4H   EPZ $9
0357     23 R6L   EPZ $C
0357     24 R6H   EPZ $0
0357     25 ;
0357     26 ; CODE SW16 ; 22
0357     27 ; R0=R2
0357     28 ;
0357 A504     29      LDA R2L
0359 8500     30      STA R0L

```

```

035B A505      31          LDA R2H
035D 8501      32          STA R0H
035F           33          ;
035F           34          ; CODE SW16 : D6 02 06
035F           35          ; COMPARE R0 ET R6
035F           36          ; FIN SI R0<R6
035F           37          ;
035F A501      38          LDA R6H
0361 C500      39          CMP R6H
0363 9035      40          BCC FIN
0365 D006      41          BNE SUBE
0367 A500      42          LDA R0L
0369 C50C      43          CMP R6L
036B 902D      44          BCC FIN
036D           45          ;
036D           46          ; CODE SW16 : 26
036D           47          ; R0=R6
036D           48          ;
036D A50C      49          SUBE  LDA R6L
036F 8500      50          STA R0L
0371 A50D      51          LDA R6H
0373 8501      52          STA R0H
0375           53          ;
0375           54          ; CODE SW16 : E1 02 62
0375           55          ; R0=R0-R1
0375           56          ; FIN SI R1>R0
0375           57          ;
0375 38        58          SEC
0376 D8        59          CLD
0377 A500      60          LDA R0L
0379 E502      61          SEC R1L
037B 8500      62          STA R0L
037D A501      63          LDA R0H
037F E503      64          SEC R1H
0381 8501      65          STA R0H
0383 9015      66          BCC FIN
0385           67          ;
0385           68          ; CODE SW16 : A4
0385           69          ; R0=R0+R4
0385           70          ;
0385 18        71          CLC
0386 A500      72          LDA R0L
0388 6508      73          ADC R4L
038A 8500      74          STA R0L
038C A501      75          LDA R0H
038E 6509      76          ADC R4H
0390 8501      77          STA R0H
0392           78          ;
0392           79          ; CODE SW16 : 36
0392           80          ; R6=R6
0392           81          ;
0392 A501      82          LDA R0H
0394 850D      83          STA R6H
0396 A500      84          LDA R0L
0398 850C      85          STA R6L
039A           86          ;
039A           87          ; CODE SW16 : 00
039A           88          ; RETOUR
039A           89          ;
039A 60        90          FIN  RTS

```

port haut  
 octet bas  
 de l'adresse suivie  
 de l'adresse RTS

\*300.139F

0300-	A9	03	8D	FA	03	A9	10	8D
0308-	F9	03	A9	4C	8D	F8	03	60
0310-	A4	34	89	00	02	C9	AA	D0
0318-	0C	E6	34	A2	07	E5	3C	95
0320-	02	CA	10	F9	60	A0	02	E1
0328-	3C	99	0E	00	83	10	F8	20
0330-	8E	F8	A6	2F	CA	00	0C	A5
0338-	0B	29	0D	F0	09	29	08	F0
0340-	05	85	0D	20	57	03	A2	00
0348-	E5	0B	91	42	E8	20	B4	FC
0350-	C6	2F	10	F4	90	CF	60	A5
0358-	04	85	00	A5	65	85	01	A5
0360-	61	C5	00	90	35	00	06	A5
0368-	00	C5	0C	90	2D	A5	0C	85
0370-	00	A5	0D	85	01	38	D8	A5
0378-	00	E5	02	85	00	A5	01	E5
0380-	03	85	01	90	15	18	A5	00
0388-	65	08	85	00	A5	01	65	09
0390-	85	01	A5	01	85	0D	A5	00
0398-	85	0C	60	EA	EA	EA	EA	EA

JSR. o  
 l'adresse  
 suivante  
 de l'adresse  
 RTS

# Changez Votre Poignée de Jeu

Gilles Mauffrey

## CHANGEZ VOTRE POIGNEE DE JEUX

Tous les jeux n'utilisant qu'une seule poignée de jeu mettent à contribution la poignée 0. Cette poignée s'use donc beaucoup plus vite que la poignée 1 moins sollicitée ; il peut par conséquent s'avérer utile de mettre certains jeux sur la poignée 1, ou de disposer pour chaque jeu d'une version sur chaque poignée.

Il suffit pour réaliser ce transfert d'explorer systématiquement tout le programme assembleur, mais ce travail est fastidieux et mieux vaut le confier à un programme. Ma première idée fut d'écrire un programme d'exploration systématique du jeu en BASIC; malheureusement tous les jeux binaires que je possède utilisent directement les adresses du BASIC.

Il fallait donc écrire le programme en assembleur. Ne possédant pas d'assembleur au moment de la rédaction de cet article, j'ai directement écrit le programme en hexadécimal en commençant à l'adresse hexa \$8000, supérieure aux adresses utilisées habituellement par les jeux. Le lecteur pourra déplacer ce programme, en prenant la précaution de modifier tous les JMP, JSR et les LDA concernés.

L'idée de base du programme est de repérer tous les groupes de trois octets consécutifs formant l'instruction LDA \$C064, correspondant au potentiomètre de la poignée 0, puis tous les groupes de trois octets consécutifs formant l'instruction LDA \$C061, correspondant au bouton de la poignée 0 ; ces instructions sont modifiées respectivement en LDA \$C065 et LDA \$C062, correspondant à la poignée 1.

Pour des raisons de sécurité exposées plus loin, on imprime les adresses hexa correspondantes. Rappelons que LDA \$C064 s'écrit en hexa AD 64 C0 ; comme il est facile de rencontrer l'octet AD, le programme cherche tout d'abord deux octets consécutifs contenant 64 (ou 61) et C0 ; s'il les a trouvés, il remonte alors de deux octets pour vérifier que AD s'y trouve et procède dans ce cas à la transformation de 64 en 65 (ou de 61 en 62).

Pour entrer le programme, passer tout d'abord en moniteur par CALL -151, puis commencer à l'adresse \$8000 : AD 72 AA 85... Une fois le programme entré, le sauvegarder sur disquette par : BSAVE NOM,A\$8000,L\$C4.

Pour utiliser le programme, il faut le charger tout d'abord en mémoire par BLOAD NOM, charger ensuite le programme de jeu (BLOAD nom du programme) ; enfin, après s'être assuré qu'il n'y avait pas chevauchement des deux programmes, lancer l'exécution par CALL 32768. Il ne reste plus qu'à sauvegarder la nouvelle version du jeu. Je vous conseille de sauvegarder cette nouvelle version sous un autre nom; en effet, toutes les suites AD 64 C0 (resp. AD 61 C0) ayant été changées en AD 65 C0 (resp. AD 62 C0), il est alors possible que le programme ait modifié des octets ne correspondant pas à la poignée. C'est d'ailleurs pour cette raison que les adresses modifiées sont données en sortie. Toutefois, ce problème n'est apparu dans aucun des cas que j'ai traités.

Remarque : ce programme ne transforme que les jeux lisant directement à l'adresse de la poignée et ne faisant pas appel au sous-programme PREAD localisé en \$FB1E ; si tel est le cas, il n'est pas très difficile de modifier en conséquence le programme. Penser que le numéro de la poignée est alors placé dans le registre X qui a dû être déclaré antérieurement.

PROGRAMME PRINCIPAL

8000	AD 72 AA	LDA \$AA72	
8003	85 01	STA \$01	CHARGE EN PAGE ZERO L'ADRESSE
8005	AD 73 AA	LDA \$AA73	DE DEPART DU PROGRAMME JEU
8008	85 02	STA \$02	
800A	20 58 FC	JSR \$FC58	HOME
800D	20 8C 80	JSR \$808C	SOUS ROUTINE TITRE POTENTIOMETRE
8010	A9 64	LDA #\$64	STOCKAGE EN \$08 DE \$64 CORRESPONDANT
8012	85 08	STA \$08	A L'ADRESSE C064 DU POTENTIOMETRE
8014	20 30 80	JSR \$8030	APPEL SOUS PROGRAMME MODIFICATION
8017	AD 73 AA	LDA \$AA73	RESTAURE L'OCTET SUPERIEUR DE L'ADRESSE
801A	85 02	STA \$02	DE DEPART DU JEU
801C	20 8E FD	JSR \$FD8E	<RETURN>
801F	20 A1 80	JSR \$80A1	SOUS ROUTINE TITRE BOUTON
8022	A9 61	LDA #\$61	STOCKAGE EN \$08 DE \$61 CORRESPONDANT
8024	85 08	STA \$08	A L'ADRESSE C061 DU BOUTON (POIGNEE 0)
8026	20 30 80	JSR \$8030	APPEL SOUS PROGRAMME MODIFICATION
8029	60	RTS	RETOUR AU BASIC

SOUS PROGRAMME MODIFICATION

8030	AE 61 AA	LDX \$AA61	DANS X NOMBRE DE PAGES A EXPLORER
8033	A0 00	LDY #\$00	INITIALISATION DE Y
8035	B1 01	LDA (\$01),Y	LECTURE DE L'OCTET SUIVANT DU JEU
8037	C8	INY	INCREMENTATION DE Y
8038	C0 00	CHP #\$00	SI LA PAGE EST TERMINEE
803A	D0 07	BNE \$8043	INCREMENTER \$02 POUR
803C	E6 02	INC \$02	PASSER A LA PAGE SUIVANTE
803E	CA	DEX	ET DECREMENTER X
803F	E0 FF	CPX #\$FF	A-T-ON FINI
8041	F8 48	BEQ \$808B	SI OUI RETOUR
8043	C5 08	CHP \$08	COMPARAISON AVEC \$08 CONTENANT \$64 OU \$61
8045	D0 EE	BNE \$8035	SI NON EGAL RETOUR AU DEBUT SINON
8047	B1 01	LDA (\$01),Y	LECTURE DE L'OCTET SUIVANT
8049	C9 C0	CHP #\$C0	EST-IL EGAL A \$C0
804B	D0 EA	BNE \$8037	SI NON RETOUR POUR COMPARER A \$08
804D	A5 01	LDA \$01	
804F	85 07	STA \$07	LES ADRESSES \$07 ET \$08 SONT UTILISEES
8051	A5 02	LDA \$02	POUR LE CALCUL DE L'ADRESSE DE LDA \$C0--
8053	85 08	STA \$08	
8055	C0 02	CPY #\$02	
8057	B0 02	BCS \$805B	
8059	C6 07	DEC \$07	
805B	88	DEY	
805C	88	DEY	REMONTÉE DE 2 OCTETS POUR TROUVER LDA
805D	B1 06	LDA (\$06),Y	
805F	C9 AD	CHP \$AD	A-T-ON LDA
8061	F0 05	BEQ \$8068	SI OUI IMPRIMER ET MODIFIER
8063	C8	INY	
8064	C8	INY	REMISE A JOUR DE Y ET RETOUR
8065	4C 35 80	JMP \$8035	

8068	18	CLC	DEBUT DU CALCUL DE L'ADRESSE DE L'INSTRUCTION
8069	98	TYA	
806A	65 06	ADC \$06	
806C	85 06	STA \$06	
806E	90 02	BCC \$8072	
8070	E6 07	INC \$07	NE PAS OUBLIER LA RETENUE EVENTUELLE
8072	A5 07	LDA \$07	
8074	20 DA FD	JSR \$FDDA	IMPRESSION DE L'OCTET SUPERIEUR
8077	A5 06	LDA \$06	
8079	20 DA FD	JSR \$FDDA	IMPRESSION DE L'OCTET INFERIEUR
807C	20 8E FD	JSR \$FD8E	<RETURN>
807F	C8	INY	MISE SUR LA POIGNEE 1
8080	E6 08	INC \$08	
8082	A5 08	LDA \$08	
8084	91 01	STA (\$01),Y	STOCKAGE DE LA VALEUR (\$65 OU \$62)
8086	C6 08	DEC \$08	
8088	4C 63 80	JMP \$8063	RETOUR A LA SUITE DE L'INSPECTION
808B	60	RTS	

#### ECRITURE DU TITRE POTENTIO

808C	A2 08	LDX #\$08	EN X NOMBRE DE LETTRES
808E	A9 0A	LDA #\$0A	POSITIONNEMENT DU CURSEUR
8090	85 24	STA \$24	EN COLONNE 11
8092	E6 24	INC \$24	UN ESPACE ENTRE CHAQUE LETTRE
8094	BD B5 80	LDA \$80B5,X	ADRESSE DES LETTRES
8097	20 F0 FD	JSR \$FDFO	ECRITURE D'UNE LETTRE
809A	CA	DEX	
809B	D0 F5	BNE \$8092	
809D	20 8E FD	JSR \$FD8E	<RETURN>
80A0	60	RTS	

#### ECRITURE DU TITRE BOUTON

80A1	A2 06	LDX #\$06	
80A3	A9 0A	LDA #\$0A	
80A5	85 24	STA \$24	
80A7	E6 24	INC \$24	
80A9	BD BD 80	LDA \$80BD,X	
80AC	20 F0 FD	JSR \$FDFO	
80AF	CA	DEX	
80B0	D0 F5	BNE \$80A7	
80B2	20 8E FD	JSR \$FD8E	
80B5	60	RTS	

80B6	0F 09 14 0E 05 14 0F 10	
INVERSE	0 I T N E T 0 P	
80BE	0E 0F 14 15 0F 02	
INVERSE	N O T U O B	

# Programmer en Pascal ?

**Christian Colmant**

On parle beaucoup du Pascal, mais on le voit peu. On voudrait bien s'y essayer, mais on redoute un peu cette nouvelle expérience.

Faut-il ou ne faut-il pas programmer en Pascal ?

La réponse à cette question est une autre question : "Que voulez-vous programmer en Pascal ?". Et c'est à partir de ce moment là que l'on peut commencer à discuter. Alors, allons-y !

Si vous achetez un livre sur Pascal, qu'il soit en Français ou en Anglais, toutes les préfaces vous diront que c'est Niklaus WIRTH, de l'Université de Zürich, qui dès 1965 proposait un langage devant étendre le domaine de l'ALGOL au delà des calculs algébriques et numériques. En 1968, le Professeur WIRTH écrivit, en FORTRAN sur CDC 6000, un premier compilateur ayant pour base l'ALGOL 60, qu'il baptisa Pascal, en mémoire du philosophe et mathématicien du 17ème siècle.

L'enthousiasme ne fut pas unanime, mais cela n'arrêta pas le Pr WIRTH. Il réécrivit un nouveau compilateur Pascal, qu'il annonce en 1971, compilateur écrit en Pascal ! Comment ? Un Pascal écrit en Pascal ? Comment peut-on compiler un compilateur ? En fait, cela fonctionne parfaitement bien, et l'ère du Pascal est née. Depuis, des développements du Pascal se sont faits en Pascal. Nous pouvons noter le plus intéressant, réalisé par l'Université de Californie à SAN DIEGO : le Pascal UCSD, du Professeur Kenneth L. BOWLES. Ce produit est maintenant largement diffusé sur mini, micro-ordinateurs et plus particulièrement sur APPLE ou SILEX par la société SOFTECH aux U.S.A.

## Ecrire un langage de programmation sous Pascal

Ouf ! Ce sera tout pour l'historique. J'espère ne pas vous avoir trop ennuyé, mais nous avons déjà commencé à répondre à la question "Que programmer en Pascal ?" par un des raffinements les plus subtils du Pascal.

Si un Pascal peut être écrit en Pascal, un autre langage de programmation peut l'être également. Par exemple, le FORTRAN qui existe sous Pascal ou d'autres grands langages. Mais pourquoi ne pas développer des langages de programmation spécifiques sous Pascal, à l'usage d'utilisateurs non informaticiens. Un langage pour les gestionnaires, un langage pour les mathématiciens... A l'Institut de Programmation - Université PARIS VI, par exemple, il est envisagé d'écrire un langage de programmation en Français, sous Pascal, le LSE. C'est peut-être l'aspect le plus évolué des possibilités du Pascal.

Mais l'intérêt du Pascal ne se limite pas, et de loin, à la possibilité d'écriture de compilateurs en Pascal.

## L'analyse structurée de programme

Pour un programmeur averti, aborder Pascal sans aborder la notion de programmation structurée serait une hérésie. En fait, la structuration de programme réside dans

une méthode d'analyse qui n'est plus de haut en bas (organigramme qui commence en haut d'une page et finit en bas), mais qui se décrit sous forme de boîtes, dont chacune représente une fonction élémentaire, qui s'imbriquent les unes dans les autres pour constituer le programme. Il en est de même pour les données.

Cette structure est appelée "structure de blocs" ; elle remet en cause la méthode traditionnelle d'analyse des programmes et demande au programmeur de réviser ses méthodes.

Pascal est un langage particulièrement adapté à la programmation structurée ; là se trouve probablement son intérêt principal.

La programmation structurée, grâce au découpage par blocs, autorise l'analyse et la programmation d'un programme par raffinements successifs. Il faut diviser pour régner ... Le programmeur commence par mettre au point le niveau général, le programme principal ; il passe ensuite aux niveaux plus détaillés pour aboutir enfin aux instructions proprement dites.

L'approche de la programmation structurée réside donc dans la clarté, la lisibilité et la facilité de maintenance des programmes.

### Apprendre le langage Pascal

"Apprendre à programmer, c'est comme apprendre à rouler à bicyclette... Il faut avant tout pratiquer" écrit John COLIBRI dans son livre.

Pascal est en fait un langage comme les autres avec sa syntaxe et sa sémantique propres. Dans le cas du Pascal UCSD, vous disposez en plus d'un système d'exploitation disque "FILER" propre au Pascal, d'un éditeur, d'un assembleur..., ce qui fait sa puissance et sa souplesse d'utilisation.

Donc, apprendre Pascal ne pose a priori qu'un problème de temps, car ce langage est plus riche en instructions que BASIC, et, dans le cas du Pascal UCSD sur APPLE II ou SILEX, il réclame l'apprentissage du FILER et de l'éditeur. On ne saurait trop conseiller dans ce cas à ceux qui possèderaient déjà un kit Pascal UCSD et ne l'auraient pas encore utilisé, de se procurer le livre de John COLIBRI "Découvrez Pascal sur Apple II" et, chez LEANORD, "Vos premiers pas en Pascal UCSD".

### Avantages et inconvénients du Pascal

On pourrait dire que Pascal n'a que des avantages, par sa structure modulaire en blocs, sa définition des types de données, sa lisibilité (quand on sait le lire), par la lecture et modification des programmes, la puissance des instructions et les possibilités d'en créer, par l'efficacité d'exécution (sous forme d'un compilateur spécial) au point de vue vitesse et place mémoire.

Pour la réalisation de progiciels, Pascal apporte un avantage supplémentaire relativement au BASIC. Il devient possible de fournir à l'utilisateur une version compilée d'un programme, sans y joindre la liste source, ce qui réduit encore les risques de piratage.

Mais, car il y a un 'mais', Pascal étant un langage dit 'de haut niveau', son apprentissage demande un peu plus d'efforts que celui de BASIC, et plus d'efforts encore si l'on souhaite bien appréhender son concept de "Langage de programmation

structurée", afin de bien exploiter toutes ses ressources.

### Quand utiliser Pascal ?

Ceci dit, il devient impératif d'utiliser Pascal dès qu'un programme BASIC ou autre devient monstrueux et tentaculaire. Alors que pour écrire des programmes de jeux simples ou réaliser de petits calculs le langage BASIC est parfaitement adapté, dès qu'un programme grandit, vous arriverez très vite à saturer votre micro-ordinateur. A ce moment là, passez en Pascal ; vous découvrirez les avantages implicites de ce langage et vous conserverez le BASIC pour les démonstrations.

### De quoi sera fait l'avenir ?

Hier, c'était ALGOL (langage qui est resté universitaire et voué à l'enseignement de l'Algorithmique), aujourd'hui c'est Pascal et demain ce sera ADA (en hommage à la première programmeuse au monde), qui est un super-Pascal adapté et imposé par le Ministère de la Défense aux U.S.A., pour les grands projets informatiques.

En conclusion, si on programme juste pour se distraire, restons au BASIC ; c'est un langage simple à apprendre et suffisamment performant. Si le hobby devient passion, alors passionnez-vous aussi pour Pascal, et surtout, s'il devient métier, vous devez introduire Pascal dans votre culture informatique.

### Pour votre bibliothèque

- Découvrez Pascal sur Apple II, par John COLIBRI
- Programmer en Pascal, par Daniel Jean DAVID et Jean Luc DESCHAMPS (Edition du P.S.I.)
- Introduction au Pascal, par Pierre LE BEUX (SYBEX)
- Vos premiers pas en Pascal UCSD (LEANORD)
- Initiation à la programmation en Pascal, par G. GRIMOMPRES et JF PONS (DATA GESTION)
- Problem solving using Pascal, par KL BOWLES (SPRINGER Verlag)
- Pascal user and report, par WIRTH et K. JENSEN (SPRINGER Verlag - Special Printing for APPLE Computer Inc)
- Le langage Pascal par l'Université de Montréal (DUNOD)
- APPLE Pascal (reference manual chez APPLE Computer Inc)
- Le langage de programmation Pascal, par P. KRUCHTEN (EYROLLES)



## Trucs et astuces

Pour de nombreuses raisons, les utilisateurs de systèmes Apple n'utilisent pas toujours les mêmes slots pour mettre des cartes données. Il y a d'ailleurs plus de cartes disponibles sur le marché que de places pour les y installer.

Ainsi, quand on doit travailler sur un nouveau système, faut-il souvent inspecter les entrailles de l'appareil pour voir sur quel slot se trouve la carte de l'imprimante, et où se trouvent les autres cartes éventuelles. Faute de quoi, certains programmes mettent le système en panne parce que telle ou telle carte n'est pas placée comme prévu.

Le programme listé en page 66 vous permet de découvrir la nature de la carte placée dans chaque slot. Vous pouvez maintenant, dans vos programmes, intégrer une routine de recherche du slot de l'imprimante. Vous éviterez ainsi de poser la question à l'utilisateur du programme ou, ce qui arrive parfois dans des programmes commerciaux, de le coincer définitivement s'il n'a pas eu le bon goût d'utiliser ses slots de la même façon que vous.

RUN70 vous fournit le code associé à chaque slot : vous pourrez alors compléter le programme si vous avez accès à d'autres cartes que celles déjà étudiées.



# pom's

## **Pour son prochain numéro, POM'S se propose de vous présenter les articles suivants :**

- Banc d'essai des utilitaires : Dakin et Apple Doc.
- Les limites des éditeurs de texte sur Apple.
- Le Pascal - une première leçon.
- Apprentissage de l'assembleur-II
- Des applications de graphique haute résolution.
- Une incursion dans les mystères du DOS.
- Les cartes 80 colonnes.
- Faites le ménage dans la mémoire.
- Le jeu de la vie.

### **Plus comme dans chaque numéro :**

- Rubrique antibugs ■ Problèmes à résoudre ■ Information des clubs ■ Courrier des lecteurs.

5 REM

\*\*\* Programme d'analyse des slots - Copyright Pom's \*\*\*

```
10 TEXT : HOME : Z$="ANALYSE DES SLOTS" : HTAB 20-LEN(Z$)/2 : INVERSE : PRINT
Z$ : NORMAL : PRINT : NC=8 : FOR K= 0 TO NC : READ A$(K),A(K) : NEXT
20 DATA CONTROLEUR,828,APPLESOFT/LANGAGE,3341,DE COMMUNICATION,6200,HORLOGE,8392,
SERIE,14360,PARALLELE,18504,TABLETTE GRAPHIQUE,30975,SPEECHLAB,53199,KEYBOARD
FILTER,54048
30 S=49152 : FOR I=1 TO 7 : J = S + I*256 : PRINT "SLOT NUMERO "I" : ";
40 FOR K=0 TO NC : IF 256*PEEK(J+5) + PEEK(J+7) = A(K) THEN PRINT "CARTE "A$(K);:
GOTO 60
50 NEXT : PRINT "VIDE";
60 PRINT : PRINT : NEXT I : END
65 REM *** Recherche des codes des slots ***
70 TEXT : HOME : INVERSE : PRINT "IDENTIFICATION DES SLOTS :" : NORMAL :
S=49152
75 FOR I=1 TO 7 : PRINT : PRINT "SLOT "I" V="256*PEEK(S+5+256*I)+PEEK(S+7+256*I)
: NEXT
```

### ANALYSE DES PARAMETRES SYSTEME

Si vous souhaitez à un moment donné savoir quels sont les paramètres du système, il faut faire les PEEKs suivants (moins 16384 pour un 32K) :

Volume actuel	= PEEK(43622)	
Drive actuel	= PEEK(43624)	
Slot actuel	= PEEK(43626)	
Longueur d'enregistrement	= PEEK(43628)	
Numéro d'enregistrement	= PEEK(43630)	
Numéro de l'octet actuel	= PEEK(43632)	... Source Apple Answer Book

# pom's

## BULLETIN D'ABONNEMENT

Je désire recevoir le prochain numéro de POM'S

Avec disquette \_\_\_\_\_ 85 Francs TTC  
 Sans disquette \_\_\_\_\_ 35 Francs TTC

Je désire m'abonner pour 4 numéros

Avec disquette \_\_\_\_\_ 295 Francs TTC  
 Sans disquette \_\_\_\_\_ 120 Francs TTC

Nom \_\_\_\_\_

Adresse \_\_\_\_\_

Envoyez ce bon de commande et votre règlement à :

**Editions MEV - 49 rue Lamartine - 78000 Versailles**

**UN DOCUMENT  
ESSENTIEL**

**Avant  
d'acheter  
votre ordinateur  
lisez le**

**GUIDE 81-82**

**de  
L'ORDINATEUR  
INDIVIDUEL**

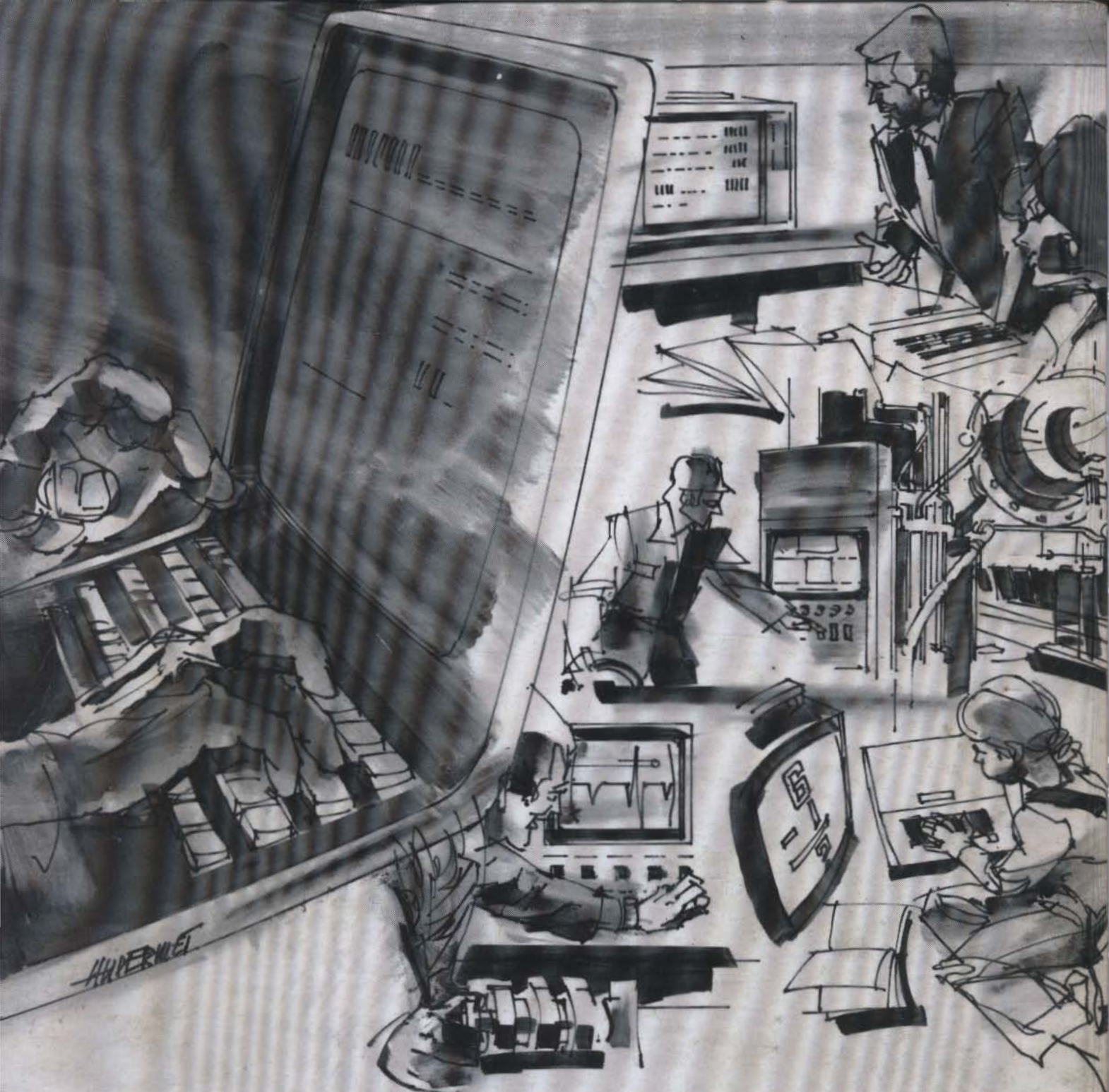
**au sommaire :**

- Panorama des ordinateurs valant entre 250 FF et 60 000 FF (plus de 80 matériels).
- Panorama des imprimantes valant moins de 10 000 FF (plus de 30 matériels).
- Réactualisation de 22 bancs d'essai parus dans l'OI depuis le N° 1.
- Annuaire des fournisseurs (plus de 500 adresses).
- Annuaire des clubs (plus de 200 adresses).
- Dictionnaire de l'informatique individuelle.
- Le point sur les nouveautés parues depuis l'été 1980.
- Et une série d'articles pour vous "guider" sur le chemin de votre informatique individuelle.

**25 FF\* chez votre marchand de journaux**

Pour recevoir, chez vous, le Guide 81-82,  
il vous suffit d'envoyer vos nom et adresse ainsi qu'un chèque de 25 FF\* à  
L'ORDINATEUR INDIVIDUEL (GUIDE 81-82) 41, rue de la Grange aux Belles - 75483 Paris Cedex 10

\* Etranger 30 FF



# concurrentiel votre micro-informatique!

Vous êtes industriel, chercheur, enseignant, particulier, membre de profession libérale.

Nous sommes une équipe d'ingénieurs ayant une position de leader sur le marché de la micro-informatique et nous avons une expérience unique en matière d'installation et maintenance de systèmes.

Nous sommes faits pour nous entendre.



Micro Informatique Diffusion

51 bis, AVENUE DE LA RÉPUBLIQUE, 75011 PARIS. TÉL : 357.83.20.

● **Micro-ordinateurs.** Apple Commodore Pertec.

● **Périphériques.**

Disquettes, disques durs, imprimantes, terminaux clavier-écran, tables traçantes, tables à digitaliser.

● **Interfaces.**

Pour terminaux de tous types (V 24 RS 232C, 8 bits parallèles). Entrées analogiques. Sorties analogiques. IEEE-488. Entrées BCD. Cartes base de temps horloge. Calcul rapide.

Ouvert de 9h à 12h et de 14h à 19h. Sauf le dimanche.