

## A PROPOS DU CLUB

Le Bureau  
J. Belin

Editorial 1  
Une conférence aux Pays-Bas 2

## HP28

G. Toublanc  
G. Toublanc

Le nouveau Coin des Codes HP28 4  
Réhabilitation 6

## HP48

G. Toublanc  
J.F. Garnier  
J.F. Garnier  
G. Toublanc  
P. de Sacy  
G. Toublanc

Littérature HP48 8  
La nouvelle disquette HP48 9  
Numéro de semaine (acte III bis) 10  
Conventions d'écriture des programmes 11  
Tracé de l'ensemble de Mandelbrot 12  
Le nouveau coin des codes HP48 18

## HP95 / HP100

J. Belin  
J. Belin  
J. Belin

Nouveaux produits 22  
La troisième disquette pour HP95 / HP100 22  
Les fonctions internes des palmtops 24

Le coin des codes 31

## EDITORIAL

Encore une fois, ce JPC est encore un peu en retard. Nous espérons que vous nous en excuserez lorsque vous en connaîtrez la raison. En effet, ce retard est dû au fait que nous venons de passer un mois un peu fou, pendant lequel nous avons récupéré, trié et sélectionné l'équivalent de 16 Mo de programmes et fichiers en tous genres, destinés aussi bien aux HP48 qu'aux HP95/HP100 ! Bien sûr, si nous avons récupéré ces fichiers, c'est pour vous en faire profiter. Et puisque nous savons que vous êtes toujours impatients d'obtenir de nouveaux programmes pour vos machines préférées, nous commençons la distribution dès ce mois ci. D'autres disquettes devraient suivre le mois prochain, mais laissez nous travailler !

L'autre grande information de ce mois est notre récente adhésion au réseau CompuServe. Hormis le fait de pouvoir récupérer des programmes très récents (certains inclus dans une des disquettes auront moins de quinze jours au moment où vous lirez ces lignes), cette adhésion nous permet enfin de renouer des contacts plus directs avec les principaux acteurs du monde HP, dont bien entendu certains membres des équipes de développement de nos machines préférées. La seule mauvaise surprise (non, une des deux mauvaises surprises dira notre cher trésorier) étant que si l'on désire prendre connaissance de tous les messages, et grappiller quelques informations intéressantes, il nous faudra probablement se réserver au moins une heure de lecture par jour...

Tout ceci expliqué, vous comprendrez que nous n'avons pas vraiment eu le temps de nous consacrer à fond à nos articles. Il vous faudra donc attendre un petit mois avant d'avoir des précisions à propos de certaines informations dont nous avons eu connaissance, comme la possibilité de tripler la fréquence des HP48 ou l'annonce d'un disque dur externe de 80 Mo, à la norme PCMCIA version 1.0 (donc adapté aux HP95), proposé aux environs de 1500 F...

En attendant, bonne lecture !

## UNE CONFERENCE

### AUX PAYS BAS

Comme certains d'entre vous le savent déjà, le club Hollandais Prompt-HPGC organise les 28 et 29 mai (à Amsterdam) une conférence internationale.

Parmi les sujets traités, différents thèmes seront plus particulièrement abordés :

- Les calculateurs et l'Education,
- Les interfaces Graphiques sur HP48,
- Les relations entre les clubs utilisateurs.

Il y aura aussi des tutoriels sur la programmation des HP48 et HP95/HP100 et (très probablement) des présentations de nouveaux matériels.

Parmi les personnalités ayant déjà donné leur accord, on pourra noter la présence de Diana Byrne, qui a occupé le poste de "Project Manager" pendant le développement des HP48G/GX...

Enfin, Hewlett-Packard a annoncé sa décision de sponsoriser le dernier événement du week-end : Pour chaque inscription au dîner d'adieu (environ 150F), ils ajoutent la même somme !

Bien sûr, nous espérons nous y rendre. Si certains d'entre vous désirent se joindre à nous, qu'ils nous contactent le plus vite possible.

Jacques Belin (123)

## **HP28**

G. Toublanc  
G. Toublanc

Le nouveau Coin des Codes HP28	4
Réhabilitation	6
<b>Le coin des codes</b>	<b>31</b>

## LE COIN DES CODES HP28S

### NEW LOOK

#### Le confort pour moins cher

La méthode d'entrée des codes des programmes créés en System Rpl ou assembleur et fournie dans le coin des codes, apporte confort et sécurité. Aussi, dès la relance de la programmation dans ces langages, depuis le JPC 84 de mars 93, mon premier souci avait été de fournir un utilitaire à l'instar de ce qui se faisait pour les autres machines. C'est ainsi qu'apparaissait ASSCOD28 et son sous-programme INPUT28 pour 1409 octets. L'actuelle version du coin des codes (depuis JPC 88) a été améliorée et occupe 1364 octets pour le corps des programmes. Cela me semble encore trop pour les 32Ko de Ram de nos HP28S, aussi je vous propose deux nouvelles versions :

- En User Rpl avec un seul programme et pour 1032.5 octets, donc un gain de plus de 340 octets.
- En System Rpl et assembleur pour 705.5 octets et aussi en un seul programme avec la rapidité en plus.

#### Les programmes

Pour les principes de la programmation, pour ne pas me répéter je vous renvoie à la rubrique HP48 de ce JPC. Je signalerai seulement que le sous-programme intégré au début de ASSCOD28 est utilisé aussi à la place de la commande INPUT de la HP48 mais inexistante sur nos HP28. On remarquera que ce sous-programme de saisie est plus simple que celui pour HP48 car la commande KEY renvoie le nom de la touche et non son numéro.

Le programme ASSCOD28 en User Rpl :

Par rapport au versions précédentes et pour réduire l'encombrement mémoire, l'assemblage de la chaîne des codes est différent :

- La chaîne est découpée en tronçons de 16 caractères, ces sous-chaînes sont inversées et converties en entiers binaires. Ceux-ci sont concaténés via SYSEVAL # 03B82h (mnémonique &\$ en System Rpl). Le SYSEVAL # 20238h extrait la partie 'données' de l'entier binaire contenant tous les codes (c'est l'équivalent en Sytem Rpl de: TWO NTHCOMPDR0P). Enfin le programme est créé proprement en mémoire via le SYSEVAL # 04F3D (TOTEMPOB en System Rpl).

Remarques:

Si cette version de ASSCOD28 est plus courte que les précédentes, elle est un peu moins rapide concernant la phase d'assemblage proprement dite. L'encombrement mémoire aurait pu être encore réduit mais cela aurait posé quelques problèmes pour le listage en parallèle de ASSCOD28 et ASSCOD48 dans le coin des codes. Comme il vous est proposé une deuxième version en System Rpl et assembleur beaucoup plus courte et rapide, on ne pourra considérer la version en User Rpl qu'à titre provisoire.

Voici donc ASSCOD28 en User Rpl et allégé.

ASSCOD28 1032.5 octets  
cksum # 55CEh

```

« RCLF
« CLLCD 3 DISP 1 DISP ""
DO DUP 4 DISP
DO UNTIL KEY END
IF DUP "ENTER" ==
THEN DROP 1
ELSE
IF DUP "BACK" ==
THEN DROP 1 OVER SIZE 1 - SUB
ELSE + DUP SIZE 1 + 5 MOD NOT 1 FC? AND
IF
THEN " " +
END
END 0
END UNTIL END
»
HEX 64 STWS 1 SF "nombre d'octets ?" ""
3 PICK EVAL 1 CF STR+ 2 * 16 DUP2 / IP
3 ROLLD MOD DUP2 0 > + SWAP 1 + # 0h "" 1 5 ROLL
FOR i
DO "ligne " i 1 - R-B # 1000h + -STR 4 6 SUB + DUP
" @ NEWLINE
codes ?" + "---- ---- ---- ----" 7 PICK i <
IF
THEN 6 PICK DUP 4 / IP + 1 SWAP OVER - SUB
END
1 FS?C
IF
THEN DROP2 SWAP CLMF HALT
ELSE 8 PICK EVAL
END DUP
WHILE DUP " " POS DUP
REPEAT DUP2 1 SWAP 1 - SUB 3 ROLLD 1 + 25 SUB +
END DROP 0 OVER SIZE 1 SWAP
FOR j OVER j DUP SUB NUM j * +
NEXT
6 PICK + DUP # FFFh AND
" @ NEWLINE
somme de controle ?" 6 ROLL SWAP + "===="
11 PICK EVAL "#" SWAP + STR+ ==
IF

```

```

THEN 1
ELSE DROP2 1000 1 BEEP 1 SF 0
END
UNTIL
END ROT 5 ROLL DROP2 3 ROLLD +
NEXT 5 ROLLD 4 DROPN
"" SWAP 1 OVER SIZE
FOR j DUP j DUP 15 + SUB "" OVER SIZE 1
FOR i OVER i DUP SUB + -1
STEP
SWAP DROP STR→ ROT SWAP # 3B82h SYSEVAL SWAP 16
STEP DROP # 20238h SYSEVAL # 4F3Dh SYSEVAL SWAP STOF
CLMF
»

```

Le programme ASSCOD28 en System Rpl et assembleur :

Ce programme suit les mêmes principes que celui en User Rpl pour la saisie des codes. Puis la chaîne est convertie en code hexadécimal par une séquence en assembleur, donc très rapidement.

La présentation du listing est identique à celle que l'on trouverait dans la rubrique HP48.

Voici donc ce programme ASSCOD28 qui n'a rien à envier à son homologue pour HP48.

### ASSCOD28

705.5 octets cksum # 9498h

```

::
XRCLF
'
:: DOCLLCD DISPROW3 DISPROW2 DISPROW1
NULL$
BEGIN DUP DISPROW4
BEGIN XKEY %0= WHILE REPEAT
$_ENTER OVER EQUAL ITE DROPTURE
:: $_BACK OVER EQUAL
ITE :: DROP ONE OVER LENS$ #1- SUB$ ;
:: &$ DUP LENS$ #1+ FIVE #/ DROP
#0<> ONE TestUserFlag OR
?SKIP APPEND_SPACE ;
FALSE
;
UNTIL
;
DOHEX SIXTYFOUR dostws ONE SetUserFlag
$ "nombre d'octets : " NULL$ DUP 4PICK EVAL
ONE ClrUserFlag
DOSTR> %2 %* COERCE SIXTEEN #/
SWAP 2DUP #0= ?SKIP #1+
UNROT #1+ 4ROLL
ZERO NULL$ TOTEMPOB
NULLLAM DUP 2DUP DUP FIVE {}N BIND
#1+_ONE_DO (DO)

```

```

BEGIN $ "ligne : "
INDEX@ #FFF #+ UNCOERCE %># HXS>$s-
FOUR SIX SUB$ &$ DUP
$ "codes : " $ "-----"
5GETLAM INDEX@ #<
IT :: 4GETLAM DUP FOUR #/ SWAPDROP
#+ ONE SWAP #1- SUB$ ;
ONE TestUserFlag ONE ClrUserFlag
ITE :: 4ROLL 2DROP ROT xHALT SWAPDROP ;
:: 3GETLAM EVAL ;
SPACE$ OVER
BEGIN 2DUP SWAP POS$>% COERCE DUP#0<>
WHILE 2DUP ONE SWAP #1- SUB$ UNROT
#1+ OVER LENS$ SUB$ &$
REPEAT
ROT 2DROP ZERO OVER LENS$
#1+_ONE_DO (DO) OVER INDEX@ DUP
SUB$ CAR$ CHR># INDEX@ #* #+
LOOP
2GETLAM #+ DUP # 1000 #/ DROP
$ "somme de controle : sm" 6ROLL SWAP "===="
3GETLAM EVAL
$_# SWAP &$ DOSTR> HXS>#
#=# ITE TRUE
:: DROP DROPBEEP FALS ONE SetUserFlag ;
UNTIL 2PUTLAM SWAPDROP 1GETLAM SWAP &$ 1PUTLAM
LOOP
XSTOF 1GETLAM DUP LENS$ NULL$ OVER TEN #- EXPAND
SWAP
CODE
gosbvl =POP#
gosbvl =SAVPTR
a=a-1 a
b=a a * compteur
a=dat1 a * adresse mémoire
d0=a * réservée
d1=d1+ 5
a=dat1 a * adresse chaîne
d1=a * de codes
d1=d1+ 10 * début des codes
lchex 39
loop a=dat1 2 * conversion
?a<=c b * ASCII -> HEXA
goyes inf10 *
a=a-con b,7 *
inf10 dat0=a 1 *
d1=d1+ 2 *
d0=d0+ 1 *
b=b-1 a *
gonc loop *
govlng =GETPTRLOOP
ENDCODE
SWAPDROP ABND
;

```

Jean-François Garnier nous avait donné une liste de points d'entrée pour HP28S avec les mnémoniques HP pour HP48 (JPC 87-88) ce qui est très pratique car cela permet d'exploiter la documentation HP pour HP48 et aussi d'utiliser avec le même confort les outils HP de développement ainsi que le décompilateur MON28 de Jean-François Garnier. Voici quelques points d'entrée utilisés ici et non donnés dans la liste citée :

#*	#07022 ✓	DISPROW2	#1BBED
#-	#06F51 ✓	DISPROW3	#1BC00
#/	#07057 ✓	DISPROW4	#1BC13
#0<>	#06E38 ✓	DOCLLCD	#0112A
#<	#06E55 ✓	DOHEX	#31680
\$_#	#3F089	DROPBEEPALS	#3FB38
\$_BACK	#1CD41	DROPTTRUE	#3F687 ✓
\$_ENTER	#19CB1	DUP#0<>	#3E9AE
%*	#11D48 ✓	EXPAND	#3DF66 *
&\$	#03B82 *	GETPTRLOOP	#125E5
1PUTLAM	#3DAAA	HXS>\$s-	#31B28
2PUTLAM	#3DABA	ITE	#3DCA3 *
4GETLAM	#3D962	NULLLAM	#3E0EA
4PICK	#3D815 ✓	POS\$>%	#2283A
4ROLL	#3D65C *	REPEAT	#056B6 *
5GETLAM	#3D986	SIXTYFOUR	#115DC
6ROLL	#3D6B8 ✓	SPACE\$	#1DFE5
APPEND_SPACE	#3E9EB	SetUserFlag	#0C8A7 *
CAR\$	#03AC8	UNROT	#3D636
ClrUserFlag	#0C8C8 *	WHILE	#056BF *
DISPROW1	#1BBA3	dostws	#316F7

Dans l'article *A la recherche des trésors cachés* (JPC 91 page 7) je faisais valoir la rapidité de FIND (pour HP28) par rapport à la fonction de recherche de code du décompilateur MON28 pour HP28. Cette affirmation n'est plus exacte car Jean-François Garnier à reprogrammé MON28 et la rapidité de la fonction est excellente maintenant. Donc ma fonction FIND n'est utile qu'à ceux qui ne peuvent scruter la Rom HP28 que dans la machine elle-même.

Guy Toublanc (276)

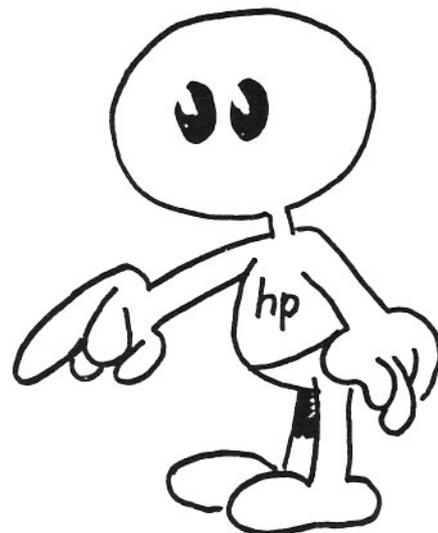
\* : adresses qui m'ont été communiquées par Jean-François Garnier

**précisions sur le mode d'emploi**

Lorsque ASSCOD28 est lancé, répondre au différents messages, les différentes entrées se validant avec la touche [ENTER].

Lors de l'entrée des codes le programme placera un espace après chaque groupe de 4 caractères. Si vous revenez en arrière (avec la touche [-]) pour une correction et que vous éliminez un espace, lors de la réintroduction de caractères il pourra y avoir un décalage dans la position des espaces mais cela est sans conséquences pour la suite des opérations.

Si, à la suite d'une erreur, le programme demande de corriger les codes d'une ligne, ceux-ci apparaîtront sur la pile à l'arrêt du programme. Il suffira de faire [EDIT] et corriger, puis de relancer avec [CONT]. Ceci constitue une différence avec ASSCOD48 pour lequel la commande INPUT de la HP48 permet d'opérer autrement.



## HP48

G. Toublanc	Littérature HP48	8
J.F. Garnier	La nouvelle disquette HP48	9
J.F. Garnier	Numéro de semaine (acte III bis)	10
G. Toublanc	Conventions d'écriture des programmes	11
P. de Sacy	Tracé de l'ensemble de Mandelbrot	12
G. Toublanc	Le nouveau coin des codes HP48	18
	<b>Le coin des codes</b>	<b>31</b>

## LITTERATURE HP48

Pour clore cette série d'articles sur les ouvrages pour HP48 series G, nous en arrivons avec le livre de référence produit par HP.

Les manuels accompagnant les calculatrices HP sont en général moins sommaires que ceux des machines des autres marques. Tout d'abord parce que une machine HP offre plus de possibilités et d'autre part les manuels doivent être à la hauteur de la réputation de la marque qui demande à l'acheteur un sacrifice financier plus important. La sortie des HP-48 séries G s'est accompagnée de deux phénomènes :

- Dès le lancement de ces séries G et GX, un prix plus bas pour des machines aux possibilités étendues, avec une Rom deux fois plus importante et pour les GX une Ram quadruplée.

- Paradoxalement des manuels moins importants où les bases de la programmation ont pratiquement disparu. Pour les séries S nous avons un manuel en deux tomes et représentant 885 pages auquel il faut ajouter un petit aide-mémoire. Avec les séries G le manuel d'utilisation représente 591 pages auxquelles 110 pages s'ajoutent pour le guide d'initiation, au total 701 pages. Ce rétrécissement aboutit par exemple à la non-documentation de 183 fonctions ou commandes sur les 640 listées en fin d'ouvrage, et parmi elles des fonctions apparues avec ces nouvelles machines. La solution HP : voyez le manuel de référence (en option). Ceci peut choquer les utilisateurs qui comptent programmer avec leurs machines et devront avoir recours au manuel de référence non vendu avec la machine et en anglais. La justification de ceci repose sur un constat : le faible pourcentage d'utilisateurs qui programment. Alors pourquoi faire payer des pages qui ne seront pas lues par la grande majorité des utilisateurs ?

Voyons ce que contient ce :

**HP 48G Series**  
Advanced User's  
Reference Manual

Hewlett Packard  
Part Number 00048-90136  
Edition 1

Le corps de l'ouvrage se répartit en :

*1 : Programming (82 pages)*

Ceci complète les 22 pages du manuel d'utilisation qui renvoie au manuel de référence. Cela donne des bases qui seront mises en pratique dans le chapitre suivant.

*2 : Programming Examples (58 pages)*

Ce sont 24 programmes avec listings commentés et introduction qui permettront au futur programmeur de mettre en application les principes exposés dans le chapitre 1. Tout n'est pas dit mais cela constitue un bon départ.

*3 : Command Reference (425 pages)*

C'est le dictionnaire des fonctions avec leur mode d'emploi et la documentation relative aux multiples utilisations. Ceci constitue la partie principale de l'ouvrage.

*4 : Equation Reference (82 pages)*

Ceci concerne donc une partie de la carte d'équations qui a été intégrée dans la Rom. C'est principalement la reproduction des différentes équations avec les éventuels graphiques qui leur sont relatifs. Tout cela est contenu dans la Rom et affichable depuis la HP48. Etait-il nécessaire d'en disposer sous forme imprimée si ce n'est que pour avoir rapidement une vue d'ensemble de ces équations et pouvoir suivre les exemples numériques donnés ici?

Puis viennent les annexes.

*A : Error and Status Messages (17 pages)*

*B : Table of Units (4 pages)*

*C : System Flags (6 pages)*

*D : Reserved Variables (16 pages)*

*E : New Commands in the HP 48G Series (7 pages)*

Liste alphabétique des nouvelles commandes avec une à trois lignes de description.

*F : Technical Reference (9 pages)*

Il s'agit d'informations concernant :

- L'occupation mémoire des objets
- Les règles de simplification utilisées par la HP48
- Les correspondances utilisées par la HP48 entre dérivée et primitive pour l'intégration symbolique

- Le développement des fonctions trigonométriques pour : -DEF TRG\* -TRG.

*G : Parallel Processing with Lists (7 pages)*

Exposé des nouvelles possibilités d'opérations sur les listes.

Donc un ouvrage pour utilisateur très sérieux.

Le prix: je l'avais acheté 160 francs, je ne pourrais préciser son prix actuel fonction des fluctuations du dollar.

Guy Toublanc (276)

---

## UNE NOUVELLE DISQUETTE POUR HP48

Une nouvelle disquette est disponible au club. Elle comprend la *Goodies Disk #9* de J.K. Horn, ainsi que HPS (un utilitaire pour PC d'origine française), et divers programmes récupérés récemment sur CompuServe.

### La Goodies Disk #9.

On retrouve sur cette disquette les rubriques habituelles :

#### - ANIMS :

Quelques animations, dont le principal intérêt est de montrer comment écrire une animation valable pour SX ou GX, tout en tirant parti de la fonction ANIMATE de cette dernière.

#### - FRANCE :

Des jeux et quelques utilitaires, mais avec peu ou pas de documentation, je n'ai donc pas approfondi. A étudier.

#### - HACKER :

On y trouve des utilitaires de conversion `directory<->library`, ainsi que la version 1.06 du MLDL de Jan Brittonson (la version 1.01 se trouvait sur la Goodies #3) pour S(X) uniquement. Ce *Machine Language Development Library* n'utilise toujours pas le jeu de mnémoniques HP, ce que je trouve bien dommage bien que soit fourni une table de correspondance entre des 2 jeux de mnémoniques.

#### - HP :

De nouvelles versions des utilitaires HP bien connus EPSPRINT, PCLPRINT, INPRT, STPWATCH et USAG valables pour 48S(X) et G(X). USAG s'applique maintenant aux librairies comme aux mots standards. Pour les G et GX, HP nous fournit gracieusement l'application table périodique PTLIB qui avait été "oubliée" lors de l'intégration de la carte HPSolve dans la G/GX.

#### - MATH

Rubrique particulièrement bien fournie cette fois. On y traite de polynômes, de matrices, de transformées de Laplace, et bien d'autres sujets. Certaines réalisations sont sous forme de librairie, nous n'avons donc plus accès aux sources.

#### - POSTING

Divers articles toujours aussi intéressants, sur des aspects peu connus ou des bugs de nos chères machines.

#### - RPL48

Le morceau le plus intéressant de cette Goodies Disk, il s'agit de la nouvelle version des utilitaires de développement en System RPL de Detlef Mueller et Raymond Hellstern, toujours en GiftWare (alias FreeWare, pas d'obligation d'enregistrement). La nouveauté est que cette version tourne sur les SX comme sur les GX. Je rappelle qu'il faut une carte RAM de 128K en port indépendant (port 1 ou 2 pour la SX, port 1 pour la GX).

Pas de révolution donc, mais de petites modifications de détails. On trouve les nouveaux -LIB-, -RPL-, et la table des points d'entrées RPL.TAB valable pour les S et G. Il n'y a pas de nouvelle version du debugger qui était sur la goodies 8, la GX en est donc privée.

#### - UTILS

On y trouve des utilitaires déjà connus ou non :

- COMPRESS : nouvelle version du compacteur RF, pour S et G.

RFP, RFU : autre compacteur, pour G uniquement.

- XYMODEM2 : nouvelle version 2.2 du protocole XMODEM pour S(X) (voir GD7).

- PCT : Le PC Tools Library venant de France, sans doc.

- DBASE : gestion de base de données pour G(X).

- FONEMAN, PIM100, TELE32 : gestions de carnet d'adresse / téléphone.

- SCH48 : gestion d'alarmes
- SMTK : nouvelle version de cet utilitaire d'assignation de touche (voir GD7)

L'intérêt de cette GD9 me semble donc en diminution, en effet on trouve beaucoup de nouvelles versions de produits déjà connus nécessitées en particulier par l'arrivée de la 48G/GX. Mais peut-être est-ce moi qui suis un peu blasé, le nouvel utilisateur de 48G(X) y trouvera probablement une mine d'informations et d'outils qui le satisferont pleinement.

### HPS 1.0 : High Performance Shell For HP48.

Ce programme FreeWare (pour PC) permet de développer pour la HP48 et plus précisément d'éditer des programmes (User RPL) sur le PC avec le jeu de caractères de la 48 (accents, symboles spéciaux), et de transférer par Kermit sans quitter HPS.

Ce logiciel est très bien réalisé, l'éditeur est au standard Borland, multi-fenêtre, avec couper-coller.

L'aide en ligne est conséquente, avec un index, mais ne concerne que l'utilisation de HPS, et non pas la HP48.

Il est possible de faire apparaître une fenêtre avec la table ASCII de la 48 afin d'insérer facilement les caractères spéciaux de celle-ci. Inconvénient, les fichiers générés contiennent les caractères HP48, et non pas les séquences \<<, \>>, ..., ce qui peut poser des problèmes pour l'impression ou la publication (dans JPC par exemple !).

Petit regret, il ne semble pas possible de configurer HPS en mode 43 ou 50 lignes.

HPS est donc une alternative intéressante à la solution classique Editeur + Kermit.

Mon souhait: un shell pour programmer aussi simplement en System RPL !

### Fichiers CompuServe :

Jacques Belin a récupéré sur CompuServe plus de 3 Mo de fichiers divers (NDLR : la plupart étant compressés, il faut en fait compter environ 6 Mo). Un tri s'est imposé, car un certain nombre de fichiers sont soit obsolètes, soit déjà diffusés par les Goodies Disks sous d'autres noms. J'en ai extrait les applications tournant sur HP48, en mettant de côté les textes et les programmes pour PC (ils feront l'objet d'une prochaine disquette).

Un premier coup d'oeil rapide m'a permis de repérer quelques sujets intéressants :

- MATH : une librairie concernant les calculs (numériques et symboliques) sur les polynômes et les matrices.
- ASTRO : une application ... d'astronomie bien sûr.
- Des jeux : dont un StartTrek faisant dans sa version préliminaire plus de 35 Ko (que sera la version définitive ?).
- Des utilitaires : base de données, éditeur de mémoire, solutions à certains bugs de la G(X), gestion de l'infra-rouge, bloc-notes, outils graphiques, exemples en system RPL, etc ...

Tout cela nous est livré en vrac, et vous demandera un peu de travail d'exploration, cependant lorsque vous lirez ces lignes, nous aurons eu le temps de organiser et de décrire un minimum les sujets sur la disquette.

Jean-François GARNIER (242)

---

## NUMERO DE SEMAINE (ACTE III bis)

Guy Toublanc m'ayant signalé que tous les programmes de calcul du numéro de semaine publiés par Robert Pulluad, Guy et moi-même (JPC91, 92 et 93) souffraient du même défaut, à savoir un résultat incorrect pour les années commençant par un jeudi (1987, 1998, ...), je me suis plongé dans les algorithmes utilisés (ce que je n'avais pas fait précédemment, me limitant à l'optimisation du code), et ai finalement trouvé l'origine du problème.

Si la semaine commence le lundi, et si la semaine numéro 1 est la première contenant un jeudi, alors les formules correctes pour déterminer le numéro de la semaine  $s$  sont :

$$s = (n + 3 + r) / 7$$

avec :

$$r = (p-5) \text{ MOD } 7$$

$n$  = numéro du jour dans l'année considéré,

$p$  = numéro du jour dans la semaine (0=dimanche, 1=lundi, ..., 6= samedi) correspondant au 1er janvier de l'année considérée.

Mon programme NSEM3 corrigé ne change pas de taille et devient :

### NSEM3B

Checksum: #B615h 78 octets

«

```
DUP DATE IFTE
-2 ALOG      @ jj.mmaaaa 0.01
DUP2 MOD + 1 + @ jj.mmaaaa 1.01aaaa
DUP ROT      @ 1.01aaaa 1.01aaaa jj.mmaaaa
DDAYS 4 +    @ 1.01aaaa (n+3)
3.031989 ROT @ (n+3) 3.031989 1.01aaaa
DDAYS 7 MOD  @ (n+3) r
+ 7 / FLOOR  @ résultat !
```

»

Jean-François GARNIER (242)

## BIEN SE COMPRENDRE

Si j'ai pu constater, avec plaisir, que le cercle des programmeurs en System Rpl et assembleur publiant dans JPC, venait de s'agrandir, je me suis aussi rendu compte que les outils de développement n'étaient pas les mêmes pour tous. Si cela n'a aucune incidence sur le résultat, la présentation des listings et des explications peuvent dérouter des lecteurs utilisant un seul type d'outils. Ceci a pu éventuellement se produire avec les articles de Arthur Ripoll dans le JPC 94. Ces articles ont le mérite de donner d'utiles informations mais l'auteur aurait dû préciser, dès le début, les outils ou librairies qu'il emploie.

Voyons d'abord quelques points relatifs à l'article *L'insertion de datas en assembleur* :

Arthur utilise l'assembleur ASM-FLASH qui est une très bonne librairie, d'exécution rapide, avec différentes options et qui fut, jusqu'à l'arrivée de la librairie -RPL- de Detlef Mueller et Raymond Hellstern, sans concurrente pour l'assembleur sur HP48. Pour ma part j'utilise, sur HP48, plus souvent cette dernière car elle est très largement compatible avec les outils de développement HP, alors que ASM-FLASH ne l'est pas.

ASM-FLASH peut-être préférée par ceux qui n'ont pas beaucoup de place dans leur machine, en particulier les propriétaires de HP-48 sans ports d'extensions.

Revenons aux différences que l'on peut trouver dans la deuxième méthode concernant les datas. La séquence pour ASM-FLASH :

```
A=PC
GOINC Data1
A=A+C A
.....
*Data1
$hhhhh...
```

où hhhh... sont les données en hexadécimal.

est équivalente, pour les autres compilateurs cités, à :

```
A=PC
ici LC(5) (Data1)-(ici)
A=A+C A
.....
Data1 NIBHEX hhhh...
```

De même pour des données aaaa... en ASCII, au lieu de :

```
caaaa...
```

on aura:

```
NIBASC aaaa....
```

Passons à l'article suivant sur la vérification des arguments dans une librairie.

Arthur donne les adresses de routines vérifiant le nombre d'arguments, et cela avec les mnémoniques du livre *Voyage 48* :

```
LCHK0 ..... LCHK5
```

dont les équivalents sont chez HP :

```
CK0 ..... CK5
```

Mais les routines de vérification du nombre et des types d'arguments étant ignorées dans le livre cité, notre ami nous les précise avec des mnémoniques dont j'ignore la provenance :

```
LCHKT1 ..... LCHKT5
```

correspondant chez HP à :

```
CK1&Dispatch ..... CK5&Dispatch
```

Il aurait été plus simple que tout le monde parle le même langage. Nous ne pouvons en rendre responsable, ici, notre ami, mais tout simplement celui qui est à l'origine de cette belle pagaille.

Puis vient un petit programme-exemple d'utilisation de LCHKT1 ou Ck1&Dispatch. Le listing correspond à celui qui pourrait être compilé par la commande \$→ de la librairie DEVLIB :

```
PRG
$ 18ECE
□ 00033
.....
END
END
```

qui traduit avec le langage HP donnerait :

```
::
CK1&Dispatch
#33
.....
;
;
```

Par ces simples exemples on voit qu'il y a quelques problèmes qui se posent pour une publication telle JPC dont l'objectif n'est pas seulement de faire profiter ses lecteurs de programmes mais aussi d'aider par des articles explicatifs ou des listings qui sont des exemples de programmation. L'utilisation d'outils divers peut se faire naturellement mais peut-être avec quelques précautions en précisant les librairies ou compilateurs utilisés, l'origine des mnémoniques.

Il pourrait être souhaitable que chacun fasse profiter les autres de son expérience comme utilisateur de compilateurs. Une synthèse de cela serait réalisée en s'appuyant sur des exemples. Comme je l'ai dit la librairie ASM-FLASH pourra être préférée par des programmeurs avec des machines peu riches en mémoire et donc ces derniers doivent pouvoir publier leurs travaux qui seront toujours les bienvenus dans JPC.

Une table d'équivalences serait à établir pour ces outils et cela en réunissant les acquis de chacun.

De temps en temps, et pour des programmes assez courts, les listings utilisables par deux types de compilateurs seraient publiés en parallèle comme j'ai eu l'occasion de le faire dans 48Sxtant, le journal de notre ami Robert Pulluad. Cela permettrait aux habitués d'un type de compilateur de comprendre le langage des autres.

Comme je ne me contente pas de prêcher la bonne parole, vous devriez trouver dans ce JPC, un article relatif au *coin des codes* avec un listing où sont précisés les compilateurs utilisables en System Rpl pour ce type de listing.

En attendant que les petits problèmes ci-dessus se résolvent, je souhaite que notre ami Arthur Ripoll nous rejoigne encore souvent et crée des émules.

Nota :

Ci-dessus j'ai rectifié quelques erreurs d'impression qui s'étaient glissées dans les articles d'Arthur :

- Les mnémoniques de l'assembleur doivent être en majuscules pour être acceptées par ASM-FLASH, ce qui n'est pas le cas de l'assembleur SASM de HP qui accepte indifféremment majuscules et minuscules.

- Les caractères spéciaux HP48 étaient mal transcrits.

Guy Toublanc (276)

---

## VOYAGE AU CENTRE... DES FRACTALES !

Voici un petit programme pour tracer des ensembles de Mandelbrot sur votre HP48 préférée.

Voyons d'abord très rapidement comment l'on détermine un ensemble de Mandelbrot. De manière théorique, on peut dire que si l'on prend chaque point du plan complexe et qu'on les soumet indéfiniment au calcul suivant :

- A chaque complexe on fait correspondre son carré et on lui additionne le complexe initial (les coordonnées du point).

- Si le module du résultat tend vers l'infini, le point initial n'appartient pas à l'ensemble de Mandelbrot, s'il reste fini, le point initial appartient à cet ensemble.

- Comme il n'est pas possible de poursuivre les calculs à l'infini, on fixe une profondeur de calcul au début du programme. De plus, pour tester le résultat, il faut se fixer une fontière entre nombre infini et nombre fini. Il apparaît qu'au delà d'un module égale à 2, les nombres complexes tendent assez rapidement vers de fortes valeurs. Donc pour faire partie de l'ensemble, il faudra que le résultat soit inférieur à 2.

Ce qui nous donne en UserRPL :

```

«
→ depth
«
  ERASE -1.2 1.2
  FOR Y
    -1.5 1
    FOR X
      X Y R-C DUP 1 depth
      FOR Z
        SQ OVER +
      NEXT
      IF ABS 2 <
        THEN PIXON
        ELSE DROP
      END
    .1 STEP
  .1 STEP
»
»

```

Le programme que j'ai écrit étant en assembleur, il permet d'obtenir un tracé plus rapide.

Pour les calculs sur les nombres réels, j'ai choisi de les coder en nombres réels à virgule fixe ce qui permet des calculs plus rapides. Ils sont codés sur 64 bits (sur le champ W). Les 20 bits de poids faibles représentent la partie fractionnaire du nombre, les 43 bits suivants représentent la partie entière du nombre, le bit de poids fort est le bit de signe.

Toujours pour accélérer le programme j'ai essayé de réduire la lecture et l'écriture en mémoire, et pour ne pas avoir à faire de calculs avec les pointeurs D0 et D1 lorsque l'on accède à la mémoire, le programme est copié en RAM puis les données lui sont concaténées et enfin le programme commence par une écriture directe des adresses des données à tous les endroits où l'on charge D0 et D1.

Je m'explique. Chaque fois que l'on a :

```

CON(2) #B1 * D0=(5)
Dx
REL(5) Label_donnee

```

À la compilation "REL(5) Label\_donnee" est remplacé par un offset sur la donnée. Lorsque le programme est lancé et recopié en mémoire, il remplace cet offset par l'adresse à laquelle est la donnée. Ceci est en fait effectué par le deuxième code qui n'est pas recopié en mémoire, ce qui permet de n'utiliser qu'un minimum de place.

Il est possible d'effectuer une alternance de "couleurs" en remplaçant :

C=0 W	par	C=0 W
LCHEX 400000		LCHEX 400000
?C<A W		?C<A W
GOYES PixOff		GOYES In
D=D-1 A		D=D-1 A
GOC In		GOC In
GOTO Boucle.Z		GOTO Boucle.Z
In		In
* C=D A		C=D A
* ?CBIT=0 0		?CBIT=0 0
* GOYES PixOff		GOYES PixOff
GOSUB PIXON		GOSUB PIXON

On obtient ainsi une alternance de noirs et de blancs en fonction de la parité des profondeurs de calcul. Il est aussi possible de tester un autre bit que le 0 pour obtenir une alternance plus espacée.

Voici le source du programme :

**MANDEL**  
733 octets Cksum # 8AA6h

```

::
CK3&Dispatch
#221
::
'
::
% 1048576 %* DUP
%0 %< case
::
%CHS %># HXS 1 0
SWAP bit-
;
%>#
;
1LAMBIND %ABS
%># SWAP C%>%
1GETLAM EVAL SWAP
1GETLAM EVAL &HXS
&HXS SWAP C%>%
1GETLAM EVAL SWAP
1GETLAM EVAL &HXS
&HXS SIXTYEIGHT
EXPAND ABND
TURNMENUOFF
DOCLLCD
'
CODE
Deb
ST=0 5
GOSBVL =SAVPTR

```

	LCHEX 1E	A=C W
	B=C B	GOSUB A*C
	CON(2) #B1 * D0=(5)	R2=A
D2		C=R1
	REL(5) C.Y	A=C W
	A=DAT0 W	GOSUB A*C
	CON(2) #F1 * D1=(5)	C=R2
D3		ACEX W
	REL(5) ScLY	A=A-C W
	C=DAT1 W	C=R4
B1		GOSUB A+C
	GOSUB A+C	R2=A
	B=B-1 B	
	GONC B1	A=R0
	DAT0=A W	C=A W
	R0=A	GOSUB A+C
	D0=D0+ 16	C=R1
	D1=D1+ 16	GOSUB A*C
	LCHEX 40	C=R3
	B=C B	GOSUB A+C
	A=DAT0 W	R1=A
	C=DAT1 W	
B2		C=A W
	A=A-C W	GOSUB A*C
	B=B-1 B	R0=A
	GONC B2	A=R2
	DAT0=A W	C=A W
	D0=D0+ 16	GOSUB A*C
	DAT0=A W	C=R0
	D0=D0+ 16	GOSUB A+C
	C=R0	
	DAT0=C W	C=0 W
	D0=D0+ 16	LCHEX 400000
	DAT0=A W	?C<A W
		GOYES PixOff
Boucle.I		
		D=D-1 A
Boucle.J		GOC In
	CON(2) #F1 * D1=(5)	GOTO Boucle.Z
D4		
	REL(5) Itter	In
	C=DAT1 A	* C=D A
	D=C A	* ?CBIT=0 0
		* GOYES PixOff
	CON(2) #B1 * D0=(5)	GOSUB PIXON
D5		
	REL(5) Cur.Y	PixOff
	A=DAT0 W	
	R1=A	CON(2) #B1 * D0=(5)
	R3=A	D7
	D0=D0+ 16	REL(5) J
	C=DAT0 W	A=DAT0 A
	R2=C	LCHEX 00082
	R4=C	?A=C A
		GOYES NEXT1
Boucle.Z		A=A+1 A
	C=R2	DAT0=A A
	R0=C	CON(2) #B1 * D0=(5)

D8	REL(5) ScIX CON(2) #F1 * D1=(5)	DAT0=A A D0=D0+ 5 A=0 A DAT0=A A GOTO Boucle.I
D9	REL(5) Cur.X A=DAT1 W C=DAT0 W GOSUB A+C DAT1=A W	A+C B=0 S CON(6) #714818 * B=B+CON S,8 B=B&A S ?B#0 S GOYES Neg1 B=0 S CON(6) #714818 * B=B+CON S,8 B=B&C S ?B#0 S GOYES Neg1 A=A+C W B=0 S CON(6) #714818 * B=B+CON S,8 B=B&A S ?B=0 S RTNYES A=0 W A=-A W ASRB W RTN
D16	CON(2) #B1 * D0=(5) REL(5) Pix.X A=DAT0 A A=A+1 A DAT0=A A GOTO Boucle.J	Neg1 A=A+C W RTN
Fin	GOSBVL =GETPTRLOOP	A*C ST=0 0 ASLC ?ABIT=0 3 GOYES Pos2 ASRC A=-A W ST=1 0 ASLC
NEXT1	CON(2) #B1 * D0=(5)	Pos2 ASRC CSLC ?CBIT=0 3 GOYES Pos3 CSRC C=-C W CSLC ?ST=1 0 GOYES Neg ST=1 0 GONC Pos3
D10	REL(5) Orig.X A=DAT0 W D0=D0+ 16 D0=D0+ 16 DAT0=A W D0=D0+ 16 D0=D0+ 15 A=0 A DAT0=A A	Neg A=A+C W RTN
D11	CON(2) #B1 * D0=(5) REL(5) I A=DAT0 A LCHEX 0003F ?A=C A GOYES Fin A=A+1 A DAT0=A A CON(2) #B1 * D0=(5)	A*C ST=0 0 ASLC ?ABIT=0 3 GOYES Pos2 ASRC A=-A W ST=1 0 ASLC
D12	REL(5) ScLY CON(2) #F1 * D1=(5)	Pos2 ASRC CSLC ?CBIT=0 3 GOYES Pos3 CSRC C=-C W CSLC ?ST=1 0 GOYES Neg ST=1 0 GONC Pos3
D13	REL(5) Cur.Y A=DAT1 W C=DAT0 W A=A-C W DAT1=A W	Neg ST=0 0
D17	CON(2) #B1 * D0=(5) REL(5) Pix.Y A=DAT0 A A=A+1 A	Pos3 CSRC ?C>A W GOYES NoExchange

ACEX W	CON(8) 0
NoExchange	C.X
?A=0 W	CON(8) 0
GOYES FB	CON(8) 0
	Orig.X
P= 0	CON(8) 0
B=0 W	CON(8) 0
ABEX W	Cur.Y
Boucle*	CON(8) 0
B=B-1 P	CON(8) 0
GOC NoRetenu	Cur.X
A=A+C W	CON(8) 0
GOC Big	CON(8) 0
GOTO Boucle*	Pix.Y
NoRetenu	CON(5) 0
CSL W	Pix.X
P=P+1	CON(5) 0
GONC Boucle*	I
P= 0	CON(5) 0
FB	J
GOSBVL #0D5E5	CON(5) 0
Retour*	ENDCODE
	#1
?ST=0 0	ASSEMBLE
RTNYES	CON(5) #02911
A=-A W	REL(5) Deb
RTN	RPL
Big	# FFF62
A=0 W	SWAP
A=A-1 W	#-
ASRB W	SUBHXS
GOTO Retour*	SWAP
	&HXS
PIXON	CODE
CON(2) #B1 * D0=(5)	A=DAT1 A
D14	D1=D1+ 5
REL(5) Pix.Y	D=D+1 A
C=DAT0 A	GOSBVL =SAVPTR
D0=D0+ 5	D1=A
A=DAT0 A	R1=A
R0=C	LCHEX 02DCC
ST=0 0	DAT1=C A
ST=1 1	GOSUB DATA
GOSBVL #138F2	REL(5) Deb
GOVLNG #138CA	REL(5) D2
Data	REL(5) D3
Itter	REL(5) D4
CON(8) 0	REL(5) D5
CON(8) 0	REL(5) D7
ScLY	REL(5) D8
CON(8) 0	REL(5) D9
CON(8) 0	REL(5) D10
ScLX	REL(5) D11
CON(8) 0	REL(5) D12
CON(8) 0	REL(5) D13
C.Y	
CON(8) 0	

```

REL(5) D14
REL(5) D16
REL(5) D17
CON(5) 0
DATA
C=RSTK
B=C A
D0=C
A=DAT0 A
A=A+B A
D1=D1+ 10
CD1EX
C=C-A A
R0=C
BBKK
B=B+CON A,5
D0=D0+ 5
A=DAT0 A
?A=0 A
GOYES FIN
C=R0
A=A+C A
A=A+B A
D1=A
C=DAT1 A
A=A+C A
DAT1=A A
GOTO BBKK
FIN
GOSBVL =GETPTR
A=R1
PC=(A)
ENDCODE
# 40 # 83 MAKEGROB
ABUFF OVER ZEROZERO
GROB!
;
;

```

Pour utiliser ce programme, entrer dans la pile :

```

3:   coordonnees du centre du grob (complexe)
2:   distance entre pixel (complexe)
1:   profondeur (reel)

```

Par exemple (-.5,0) (.03,.03) 20 pour une vue générale de l'ensemble de Mandelbrot.

Voici un petit programme en UserRPL pour pouvoir ce déplacer dans l'ensemble de Mandelbrot :

```

«
DUP PICT STO
{ } PVIEW
C-R ROT C→ ROT
DUP2
IF <
THEN SWAP

```

```

END 4 ROLL 4 ROLL
DUP2
IF <
THEN SWAP
END ROT SWAP R→C
ROT ROT R\→C SWAP
→ coin1 coin2
«
coin1 coin2 + 2 /
coin1 coin2 - C-R
64 / ABS SWAP 131 /
ABS SWAP R→C
'PPAR' 1 coin1 PUT
'PPAR' 2 coin2 PUT
»
»

```

Il prend en entrée un objet graphique et le PPAR doit être modifié en fonction de l'échelle de l'objet graphique.

Par exemple, je veux tracer la vue générale :

```
(-.5,0) (.03,.03) 20 Mandel
```

Je modifie le PPAR comme suit:

```
{ (-2.465,-.96) (1.465,.96)
X 0 (0,0) FUNCTION Y }
```

Je lance le programme ci-dessus. Il me montre la vue générale.

Je fait [Shift-Gauche] [Fleche-Gauche].

Là, je positionne le curseur dans un coin du carré que je veux sélectionner et j'appuie sur [ENTER], puis je vais dans le coin opposé et de nouveau j'appuie sur [ENTER] puis sur [ON]. Le programme a calculera les deux premiers arguments qui vont me servir à zoomer sur la zone sélectionnée.

Je n'est plus qu'à choisir la profondeur du calcul et à relancer Mandelbrot.

Attention! : A chaque nouveau dessin correspond un PPAR. Si vous voulez zoomer sur deux zones d'un meme dessin, sauvegardez le PPAR avant les zoom,.

Pierre Silvestre de Sacy (572)

# LE COIN DES CODES HP48

## NEW LOOK

### Rétrospective

Le coin des codes HP48 est apparu pour la première fois dans le JPC 78 de juillet 92, mais avec un simple assembleur. Depuis novembre 92 (JPC 84) cette rubrique s'est enrichie avec un assembleur plus évolué et évitant de vérifier l'ensemble des codes entrés, en cas d'erreur de frappe, technique déjà utilisée pour le HP71 mais dont l'algorithme de calcul des sommes de contrôle a dû être revu.

Depuis mars 93 la rubrique HP28 a été dotée aussi de son coin des codes. L'absence de la commande INPUT pour HP28 m'avait obligé à en créer une spécialement adaptée au programme ASSCOD28, ce qui, tout compte fait, était un avantage car offrant la possibilité de gérer automatiquement l'introduction des espaces entre groupes de codes. On en était arrivé au paradoxe que ASSCOD48 était moins pratique que ASSCOD28. D'autre part ces deux programmes, présentés séparément dans le coin des codes, prenaient beaucoup d'espace. Pour toutes ces raisons, une nouvelle présentation voyait le jour dans le JPC 88, avec les deux programmes listés ensemble et un INPUT48. Ainsi ASSCOD48 et son sous-programme INPUT48 occupent 1405 octets pour les corps des programmes.

Si ASSCOD48 est à la fois d'utilisation pratique et sécurisante, il me semblait encore trop gourmand en mémoire. Aussi je vous propose deux nouvelles formules plus économes :

- En User Rpl, un seul programme de 1114.5 octets donc plus de 290 octets économisés et tout en un. Cette version est utilisable par tous et les affichages sont plus rapides qu'avec les versions précédentes.

- En System Rpl pour 699 octets, en un seul programme avec la rapidité en plus. Cette version ne peut donc être introduite que par ASSCOD48 en User Rpl.

### Les programmes

Si les techniques de programmation de ASSCOD48, dans ses différentes versions, sont passées inaperçues car non commentées, il m'a semblé utile d'apporter, à ce stade, quelques précisions.

Voyons d'abord les principes de ASSCOD48 en User Rpl.

Pour gagner des octets, le programme n'utilise pas de variables mais seulement la pile. Après la sauvegarde de l'état des indicateurs, un programme de saisie des codes est déposé sur la pile sans être évalué. Ce dernier initialise l'affichage puis attend la frappe d'une touche.

-Si touche [ENTER] alors sortie du programme de saisie.

-Sinon si touche [-] alors efface le dernier caractère.

-Sinon convertit la position de la touche en caractère :  
A à F 7 à 9 4 à 6 1 à 3 et 0

Après chaque groupe de 4 caractères ajoute 1 espace.

Revenons au programme principal après le '»' du sous programme. Après l'introduction du nombre d'octets du programme à créer calcul du nombre de lignes de 16 codes à entrer et du résidu éventuel. Puis une boucle correspondant au nombre de lignes à entrer. Prépare l'affichage pour le sous-programme de saisie et en fonction de la longueur de ligne à entrer. Le flag 1 repère une répétition d'entrée de codes à la suite d'une erreur, dans ce cas l'entrée se fait simplement par [INPUT] car les codes à corriger sont réaffichés. Sinon l'entrée se fait par le sous-programme de saisie avec [EVAL]. Une copie de la ligne de codes est allégée des espaces et une somme de contrôle est calculée pour cette ligne et ajoutée à celle des lignes précédentes pour être comparée à celle qui est demandée. Cette dernière est entrée par le sous-programme de saisie. S'il y a erreur un BEEP est émis avec demande de correction. Si tout est correct, la ligne est ajoutée à la chaîne des codes et le checksum général est mis à jour. Après la dernière ligne entrée puis vérifiée on ne garde que :

- au niveau 2 : l'état des indicateurs.

- au niveau 1 : la chaîne de l'ensemble des codes.

Celle-ci est transformée en objet graphique en la faisant précéder du prologue et des dimensions et via STR→ (méthode du livre *voyage...hp48*). La partie 'données' de l'objet graphique est extraite avec les SYSEVALS équivalents à la séquence en System Rpl :

FOUR NTHCOMPDROP (donc un Syseval en moins que dans *voyage*). Puis le programme est créé proprement via un NEWOB et les indicateurs sont restaurés.

Voici donc le listing de cette nouvelle mouture plus discrète :

### ASSCOD48

1114.5 octets cksum # 621Eh

```

« RCLF
« CLLCD 3 DISP 1 DISP ""
DO DUP 4 DISP
DO UNTIL KEY END
IF DUP 51 == THEN DROP 1
ELSE IF DUP 55 ==
THEN DROP 1 OVER SIZE 1 - SUB
ELSE
CASE DUP 17 < THEN 54 + END
DUP 66 < THEN 7 - END
DUP 76 < THEN 20 - END
DUP 86 < THEN 33 - END
DROP 48 END
CHR + DUP SIZE 1 + 5 MOD NOT 1 FC? AND
IF THEN " " + END
END 0
END
UNTIL
END
»
HEX 64 STWS "nombre d'octets ?" "" INPUT
1 CF STR→ 2 * 16 DUP2 / IP 3 ROLL MOD
DUP2 0 > + SWAP 1 + # 0h "" 1 5 ROLL
FOR i
DO "ligne " i 1 - R→B # 1000h + →STR 4 6 SUB + DUP
" @NEWLINE
codes ?" + "-----" 7 PICK i <
IF
THEN 6 PICK DUP 4 / IP + 1 SWAP OVER - SUB
END 1 FS?C
IF
THEN DROP2 SWAP OVER " a corriger" +
{ -1 } ROT + INPUT
ELSE 8 PICK EVAL
END DUP
WHILE DUP " " POS DUP
REPEAT DUP2 1 SWAP 1 - SUB 3 ROLL 1 + 25 SUB +
END DROP 0 OVER SIZE 1 SWAP
FOR j OVER j DUP SUB NUM j * + NEXT
6 PICK + DUP # FFFh AND
" @NEWLINE
somme de controle ?" 6 ROLL SWAP + "=="
11 PICK EVAL "#" SWAP + STR→ ==
IF
THEN 1
ELSE DROP2 1000 1 BEEP 1 SF 0
END
UNTIL
END ROT 5 ROLL DROP2 3 ROLL +
NEXT 5 ROLL 4 DROPN
"GROB 8 " OVER SIZE 2 / " " + + SWAP + STR→
# 4017h SYSEVAL # 62B9Ch SYSEVAL NEWOB SWAP STOF
»

```

Passons à ASSCOD48 en System Rpl.

Les principes de programmation sont identiques à ceux ci-dessus. Les caractères accentués *ô* et *à* ont été introduits dans les messages pour les puristes et la présentation des affichages est améliorée. On pourra remarquer le sous-programme de saisie, non évalué, au début avec la numérotation des touches, différente de celle en User Rpl. Le gain d'octets et de rapidité sont à l'avantage de ce langage où l'on utilise des objets en Rom et des instructions plus spécifiques ou synthétiques (DISPROW7 pour 7 DISP etc..). Ici sont utilisées 5 variables au nom de longueur nulle ce qui facilite la programmation tout en étant économe en mémoire contrairement à ce que l'on obtient en User Rpl avec les variables globales ou locales.

Voici donc le listing que vous pourrez comparer à celui de ASSCOD48 en User Rpl.

### ASSCOD48

699 octets cksum # 420Fh

```

*****
* fichier source compilable: *
* sur compatible IBMPC avec: *
* RPLCOMP du Kit de développement HP pour *
* compatible IBMPC, programmes et documentation *
* disponibles dans Goodies Disk #4 ou auprès du *
* club : Kit de développement pour HP48. *
* sur HP48 avec: *
* →RPL du Kit RPL48 de Detlef Mueller & Raymond *
* Hellstern, programmes et documentation *
* disponibles dans Goodies Disk #8 pour HP48S/X *
* et Goodies Disk #9 pour HP48S/X et HP48G/X, ou *
* auprès du club. *
*****
::
OLASTOWDOB! xRCLF
'
:: DOCLLCD DUP DISPROW5 DISPROW7
ONE THREE Disp5x7 NULL$
BEGIN DUP DISPROW6
WaitForKey DROP
:: TWENTYFIVE OVER#=case DROPTTRUE
TWENTYNINE OVER#=case
:: DROP DUPLen$ 1_#1-SUB$ FALSE ;
:: SEVEN OVER#> case :: SIXTYFOUR #+ ;
THIRTYFOUR OVER#> case :: TWENTYFOUR #+ ;
THIRTYNINE OVER#> case :: SIXTEEN #+ ;
FORTYFOUR OVER#> case #8+
DROP FORTYEIGHT
;
#>CHR >T$ DUPLen$ #1+ FIVE #/ DROP
#0= ONE TestUserFlag NOTAND
IT APPEND_SPACE
FALSE
;

```

```

UNTIL
;

DOHEX SIXTYFOUR dostws
$ "nombre d'octets : " NULL$TEMP xINPUT
ONE ClrUserFlag
DOSTR> %2 %* COERCE SIXTEEN #/
SWAP2DUP #0=?SKIP #1+
UNROT #1+ 4ROLL
ZERO NULL$TEMP
4NULLLAM() NULLLAM >TCOMP BIND
#1+_ONE_DO (DO)
BEGIN $ "ligne : "
  INDEX@ #1- #1000 #+ #>HXS HXS>$
  FOUR SIX SUB$ &$ DUP
  $ "\0a\0acodes : " &$
  $ "-----"
5GETLAM INDEX@ #<
IT :: 4GETLAM DUP FOUR #/ SWAPDROP #+ 1_#1-SUB$ ;
ONE TestUserFlag ONE ClrUserFlag
ITE
  :: 2DROP SWAPOVER $ " \e0 corriger" &$
  %-1 ROT TWO()N xINPUT ;
  :: 3GETLAM EVAL ;
SPACE$ OVER
BEGIN 2DUPSWAP ONE POS$ DUP#0<>
WHILE 2DUP 1_#1-SUB$ UNROT #1+ OVERLENS$ SUB$ &$
REPEAT
ROT2DROP ZERO OVERLENS$
#1+_ONE_DO (DO) OVER INDEX@ SUB$1# INDEX@ #* #+
  LOOP
2GETLAM #+DUP # FFF #AND
$ "\0a\0asomme de contr\4le : sm" 6ROLL SWAP&$
"===="
3GETLAM EVAL
CHR_# >H$ DOSTR> HXS>#
#=ITE TRUE
  :: 2DROP %100 %10* %1 DOBEEP
  ONE SetUserFlag FALSE ;
UNTIL 2PUTLAM SWAPDROP 1GETLAM SWAP&$ 1PUTLAM
LOOP
xSTOF $ "GROB 8 " 1GETLAM LENS$ #1+ #2/ #>$
APPEND_SPACE &$ 1GETLAM &$
ABND
DOSTR> FOUR NTHCOMPDROP TOTEMPOB
;

```

Vous trouverez la liste des codes dans le coin des codes. L'assemblage se fera avec l'une des versions de ASSCOD48, puis vous pourrez ne conserver que cette version en System Rpl qui se fera plus discrète dans votre HP48.

#### Précisions sur le mode d'emploi.

Lorsque ASSCOD48 est lancé, répondre aux différents messages, les différentes entrées se validant avec la touche [ENTER].

Lors de l'entrée des codes le programme placera un espace après chaque groupe de 4 caractères. Si vous revenez en arrière (avec la touche [-]) pour une correction et que vous éliminez un espace, lors de la réintroduction de caractères il pourra y avoir un décalage dans la position des espaces mais cela est sans conséquences pour la suite des opérations.

Si, à la suite d'une erreur, le programme demande de corriger les codes d'une ligne, ceux-ci s'afficheront via INPUT et donc la gestion des espaces ne se fera pas dans ce cas et il n'y a aucune obligation de les introduire. Il suffit de corriger la ligne de codes et de valider avec [ENTER].

Guy Toublanc (276)



## HP95 / HP100

J. Belin  
J. Belin  
J. Belin

Nouveaux produits	22
La troisième disquette pour HP95 / HP100	22
Les fonctions internes des palmtops	24

## DES NOUVELLES VENUES DU CIEL

### Du côté des étoiles...

A partir du 4 Avril, Hewlett-Packard introduit (en collaboration avec Skytel) un nouveau service de messagerie radio destinés aux utilisateurs de HP100LX.

Baptisé *Starlink*, ce service permet d'assurer les fonctions suivantes :

- Réception d'Email.
- Reception de message vocaux, retranscrits en texte par une opératrice de Skytel.
- Reception d'informations générales, financières, météo...

Le logiciel intégré permet de mettre à jour automatiquement l'agenda et le repertoire téléphonique du HP100.

L'interface matérielle est assurée par la *NewsCard*, une carte PCMCIA fabriquée par Motorola. Elle permet de recevoir les messages, même quand la carte n'est pas incluse dans le HP100. Une Led et un petit buzzer permettent d'indiquer qu'il y a des messages en attente. Les données sont stockées dans une Ram interne de 128 Ko (dont environ 90 sont cependant utilisés pour stocker le logiciel du HP100).

Fiancièrement, il faut compter 229\$ pour la carte, Puis un abonnement dont le prix dépend des différentes options choisies : taille de la zone de réception (locale, régionale, mondiale), taille maximum des messages reçus, nombre de messages reçus dans le mois, réception des informations générales...

Comme pour le NewStream, déjà opérationnel pour le HP95, un service équivalent devrait probablement bientôt être disponible en France.

### Et du coté du Soleil...

La société SunDisk vient d'annoncer une nouvelle carte Flash pour HP100, avec une capacité de 40 Mo (80 Mo, pour ceux qui veulent utiliser Stacker). Je n'ai pas confirmation du prix, mais il devrait être compris entre 1200\$ et 2000\$.

Jacques Belin (123)

## LA NOUVELLE DISQUETTE HP95 / HP100

Comme annoncé dans l'éditorial, nous avons récemment récupéré un nombre important de fichiers destinés aux HP95 et HP100.

Après une rapide sélection, nous avons décidé de répartir ces programmes en plusieurs groupes :

- Une disquette contenant des programmes fonctionnant sur HP95 uniquement.
- Une disquette contenant des programmes fonctionnant sur HP100 et PC, mais ne fonctionnant pas sur HP95.
- Une disquette de programmes fonctionnant sur les deux machines. Cela inclus de nombreux programmes System Manager HP95, qui fonctionnent parfaitement sur le HP100.
- Enfin, si la demande existe, une quatrième disquette plus destinée aux programmeurs, contenant, entre autres, des informations sur les internes des palmtops ainsi que de nombreux sources.

Nous commençons ce mois-ci avec la troisième disquette. Les utilisateurs de HP100 vont peut-être m'en vouloir un peu de ne pas leur donner tout de suite des utilitaires totalement dédiés à leur machine, mais il leur faudra attendre encore un peu, car le contenu de cette dernière disquette n'est pas encore totalement défini... D'autre part, j'ai récupéré ces derniers jours des programmes HP100 très récents (certains datant de moins d'une semaine au moment où j'écris ces lignes), mais présentant encore quelques défauts de jeunesse. Je préfère donc retarder un peu leur mise en circulation, le temps qu'une nouvelle version apparaisse. De toutes façons, je leur rappelle que si les deux premières disquettes sont étiquetées "HP95", elles contiennent une majorité de programmes fonctionnant sur le HP100.

Comme d'habitude, toutes les applications présentées sur cette disquette ont leurs différents fichiers (programme, documentations, etc...) regroupés et compressés dans des fichiers ZIP distincts. Ceci nous permet d'une part d'en mettre plus sur une seule disquette, mais aussi de vous inciter à ne redistribuer que les archives complètes, et non seulement les exécutables, comme cela est trop souvent le cas. De plus, à l'intérieur de ces archives, j'ai aussi compressé les exécutables à l'aide de PKLITE. Ainsi, ils prendront moins de place sur le disque de votre palmtop. Veuillez cependant à les décompresser avant de faire une procédure d'installation modifiant des données à l'intérieur même du programme.

Voici donc une liste des fichiers contenus dans cette nouvelle disquette. Notez cependant qu'elle n'est pas exhaustive et qu'il est possible qu'au moment où vous lirez ces lignes, de nouveaux fichiers auront été rajoutés. Dans ce cas, nous vous en tiendrons informés dans le prochain JPC.

### Directory UTILS

Si le HP100 possède une application permettant de créer des bases de données mono-fichier, celle-ci possède cependant le défaut de ne pas être compatible avec les autres programmes tournant sur PC. En effet, il est nécessaire d'obtenir le *Connectivity Pack* du HP100 pour pouvoir les lire sur PC. C'est probablement pour cela que de nombreux utilisateurs ont préféré installer Dbase II ou III sur leur machine. Cependant, une solution plus économique existe maintenant avec EFMS. Il s'agit d'un générateur de base de données mono-fichier permettant de créer de façon libre ses propres écrans de saisie. Le gros avantage est qu'il tourne sur PC et les deux palmtops et qu'il n'est pas nécessaire de recourir à des utilitaires de conversion.

Un des habitués de nos réunions mensuelles à déclaré ne pas être intéressé par le HP95, car on ne peut pas y faire tourner AutoCad ! Malheureusement, si je suis au regret de vous dire que cela n'est toujours pas possible, il existe tout de même un utilitaire permettant d'afficher les fichiers DXF sur l'écran du HP95 ! Une chose à essayer absolument est sa fonction de zoom, permettant de visualiser sur toute la taille de l'écran un texte qui n'apparaissait au début que sous la forme d'un pixel... Attention cependant à ne pas utiliser des fichiers trop complexes, car les deux palmtops ne sont pas équipés de coprocesseurs...

Un des autres utilitaires présents sur la disquette est un générateur d'hypertexte. La conversion des fichiers originaux est assez simple et le programme admet deux formats de fichiers, dont un compressé.

Dans la première disquette pour HP95, nous vous avons présenté un programme permettant de convertir (au cas où vous ne savez pas quoi en faire !) votre HP95 en télécommande. Nous vous présentons ici une nouvelle version fonctionnant parfaitement sur les deux machines.

Dans un des plus gros fichiers ZIP contenu dans cette disquette (il fait 91 Ko) se trouve TERSE, un des plus petits éditeurs de textes. Il ne fait que 4096 octets. La taille de ce fichier ZIP est justifiée par la présence de plusieurs versions du programme (avec des possibilités étendues) accompagnées chacune de ses propres programmes de configuration. Ainsi que d'une

doc faisant plus de dix fois la taille du programme lui-même.

### Directory CALC

Dans ce directory ne se trouvent que deux programmes émulant des calculatrices, mais extrêmement puissants par rapport aux programmes du même type sortis auparavant sur nos palmtops.

Le premier, nommé EXCALC, est très orienté vers le scientifique et l'analyse des fonctions. Un peut comme pour les HP28 et HP48 (mais beaucoup moins bien, tout de même !), il est possible d'évaluer des expressions symboliques. Parmi ses autres possibilités, on peut citer en vrac : intégration numérique, dérivation, statistiques, Fonctions d'Euler8, conversions d'unités, tracé de fonctions... Si ces capacités sont très aléchantes, on peut cependant regretter un certain manque d'ergonomie, non corrigé par une doc pas toujours très explicite.

Quant au deuxième programme, son absence sur cette disquette aurait pu me valoir (et à juste titre) ma destitution de mon poste de président de PPC Paris ! En effet, si nous le lançons (sous un nom tout à fait commun : PSCALC95), nous voyons apparaître un affichage ressemblant furieusement à un écran de HP48SX... Il s'agit en effet d'un programme émulant le fonctionnement de la machine, y compris la programmation en User RPL ! Bien entendu, le programme ne faisant que (?) 80 ko, il ne contient qu'une petite partie des fonctions de la machine originale et ne gère ni le graphisme ni le calcul symbolique ni la possibilité de stocker et lire les programmes sur disque. De plus, il est bien sûr impossible d'y faire tourner des programmes System RPL ou assembleur. Mais malgré ces manques, ce programme permet une première approche du fonctionnement de la machine, avec ses principales fonctions et quelques types d'objets (variables, programmes, directories...). Il est aussi à noter que la version incluse dans la disquette ne sauvegarde pas le contenu de sa mémoire (programmes et données) sur le disque interne du HP95. Pour obtenir cette fonction, il vous faudra acquérir votre licence d'utilisation.

### Directory TEXTES

Quelques petits fichiers textes (en Anglais, of course !), dont une discussion à propos des cartes PCMCIA. Elle date de 1991, c'est à dire avant l'apparition de la version 2.0, mais certains points sont toujours d'actualité.

## Directory PROGRAMR

Un seul fichier dans ce répertoire. Il s'agit d'un petit langage mixant le Basic et Le Forth. Il est peut sembler plus difficile à apprendre que le Swift!Basic, mais la doc est très complète et il est accompagné de nombreux exemples de sources. Et il est beaucoup moins cher... Bien entendu, vous serez tout à fait invités à publier vos propres programmes dans JPC !

## Directory APPLICS

Trois programmes d'application dans ce répertoire. Le premier (plutôt destiné aux consultants et travailleurs indépendants) permet de comptabiliser jour par jour les heures de travail concernant chaque affaire, de les totaliser puis d'établir automatiquement les factures correspondantes.

Le deuxième programme permet d'effectuer de calculer avec une grande précision la distance entre deux points sur une sphere, est aussi sur une certaine autre ayant la particularité d'être légèrement aplatie : la Terre.

Dans le même esprit, le dernier programme permet de calculer les heures de lever et de coucher la lune. Utile s'il y a des loup-garous parmi les membres du club...

## Directory JEUX

Rien d'extraordinaire : des jeux d'aventures en mode texte, des petits jeux d'arcade et une simulation du *Spirographe* de notre enfance. Un petit conseil cependant à ceux qui projettent d'utiliser ces programmes pendant les cours ou les réunions : n'oubliez pas de couper le son !

## Directory DIVERS

Comme pour toute disquette de compilation, il y a toujours quelques programmes inclassables : ce ne sont pas des utilitaires, ni des jeux, ni tout autre chose.

Tout d'abord, pour se reposer de l'effort intellectuel causé par l'utilisation des programmes du directory précédent, j'ai inclus quelques programmes faisant fonction d'économiseurs d'écran, bien que pour le LCD le problème ne se pose pas vraiment....

Enfin, certains d'entre vous ont peut être pu voir, dans des expositions ou des salons, des "tableaux" ne semblant rien représenter, si ce n'est des minuscules points de couleurs disposés de façon aléatoire ou rythmique. Il s'agit de l'exploitation d'un phénomène découvert par Bela Julesz, montrant que si on focalise

son attention sur un point fictif situé au delà du plan du tableau, on perçoit une sensation de relief... Pour ma part, et malgré de nombreux essais, j'ai été incapable d'y voir quoi que ce soit (oui, je sais, certaines mauvaises langues diront que le simple fait d'être président de ce club prouve que l'on n'est pas normal !), mais je vous laisse tenter votre chance. Vous pouvez me tenir au courant de vos résultats, mais si vous m'annoncez que vous y avez vu l'image du successeur du HP100, j'aurais quand même un petit doute !

Jacques Belin (123)

---

## HP95 ET HP100 : LES FONCTIONS INTERNES

Les personnes qui on pu prendre le temps de lire les sources des programmes que j'ai publié dans de précédents numéros (je pense plus particulièrement à celui de 95PANEL) ont dû s'apercevoir que j'utilisais de temps en temps des fonctions non documentées dans les ouvrages traitant des IBM PC.

Ceci est dû au fait que si nos palmtops possèdent la plupart des fonctions présentes sur les ordinateurs de bureau, ils possèdent aussi de nombreuses fonctions inédites, traitant par exemple, du contraste de l'écran ou la gestion des piles. Et je ne parle pas du "gros morceau" sur les HP95 et HP100 : les fonctions permettant de programmer des programmes System Manager.

Malheureusement, la politique de Hewlett-Packard ayant radicalement changé depuis le temps des HP-71, les seules personnes ayant accès aux renseignements concernant ces fonctions sont celles qui se sont fait enregistrer, dans le but de développer des applications commerciales. Les autres doivent se contenter des quelques fichiers d'informations provenant de CompuServe, qui si ils décrivent bien en détail les fonctions graphiques ou la programmation System Manager, ne font absolument aucune référence directe à certains aspects des fonctions internes, telles que l'utilisation des ports d'entrées-sortie. Pire, un autre aspect capital (les fonctions internes regroupées sous l'interruption 15h) sont totalement occultées, puisqu'il ne me semble même pas y avoir vu une seule référence à ce sujet.

Cependant, si l'on prend la peine d'examiner en détail certains programmes et certaines portions de la Rom des Palmtops, il est tout à fait possible de découvrir les informations qui nous intéressent.

Cet article représente l'état de mes recherches. Bien que je n'ai pas encore totalement décrypté l'ensemble des fonctions, je pense qu'il pourra déjà intéresser certaines personnes.

Mais avant de commencer, je tiens à rappeler qu'étant donné le sujet (et les méthodes que j'ai utilisé pour décoder ces fonctions), ni moi ni PPC Paris ne peuvent être tenus responsables de problèmes pouvant survenir suite à l'utilisation de ces fonctions dans vos programmes. Vous voilà prévenus !

### Remarques générales

Toutes ces fonctions sont regroupées sous l'interruption 15h, avec des numéros de fonctions compris entre 41h et 62h (certains d'entre eux ne sont pas utilisés).

En plus de la méthode officielle (l'accès par les interruptions), je vous donnerais aussi les méthodes plus "directes" permettant de d'écrire ou lire les valeurs courantes. Ce sont souvent celles-ci que j'ai utilisé pour 95PANEL.

La raison en est que si certaines fonctions permettent de sélectionner une valeur, il n'est souvent pas possible de lire la valeur courante (voir ma remarque à propos du contrôle du volume).

Cependant, les HP95 et HP100 étant totalement différents au niveau matériel, l'utilisation de ces méthodes directes rendront fatalement le programme incompatible avec l'une des machines. Elles ne sont donc à utiliser qu'en dernier recours.

### Programmation des ports

C'est ici que nous voyons que ces différences. Alors que sur le HP95, les ressources matérielles sont accessibles par différents ports, le HP100 n'accède au matériel spécifique que par l'intermédiaire de deux ports (22h et 23h) adressant plusieurs registres matériels.

Pratiquement cela se fait comme ceci :

Lecture d'un port HP95 :

```
mov dx,port
in  al,dx  → al contient la valeur courante
```

Lecture d'un registre HP100 :

```
mov al,registre
out 22h,al
in  al,23h  → al contient la valeur courante
```

Ecriture d'un port HP95 :

```
mov dx,port
mov al,valeur
out dx,al
```

Ecriture d'un registre HP100 :

```
mov al,registre
out 22h,al
mov al,valeur
out 23h,al
```

### Conventions d'écriture

Il n'est peut-être pas inutile de rappeler ici quelques conventions d'écriture !

- Le terme "mot" indique un groupe de deux octets (Un "double mot" indique un groupe de quatre octets). Sur IBM PC, rappelez vous qu'ils sont stockés de façon inverse à leur ordre dans les registres du microprocesseur. C'est à dire que dans la mémoire, si ils sont stockés sous la forme "1234h", ils seront lus apparaîtront dans les registres sous la forme "3412".
- A l'intérieur d'un octet, les bits seront numérotés de 0 à 7. 0 représentant le bit de poids faible et 7 celui de poids fort.

## LISTE DES FONCTIONS

### Gestion d'événements

Disponibilité : HP95 et HP100

Bien que non spécifique au HP95 (elle est implantée dans de nombreux Bios dont le Phoenix) cette fonction est très peu documentée. Je n'ai pas eu l'occasion de développer des programmes l'utilisant, et ne peut donc pas vous donner beaucoup de détails à ce sujet.

Elle permet d'interrompre l'exécution du programme jusqu'à ce qu'une condition externe intervienne (valeur dans un port ou dans une adresse mémoire) ou qu'un intervalle de temps donné se soit écoulé.

Disponible sur HP95 et HP100.

Valeurs d'entrée :

AH = 41h

AL = condition

bits 0-2: condition à attendre

0 tout événement externe

1 compare et retourne si égal

2 compare et retourne si non égal

3 teste et retourne si non zéro

4 teste et retourne si zéro

bit 3: réservé

bit 4: 1=port, 0=octet mémoire

bits 5-7: réservé

BH = valeur de comparaison ou valeur de masque

BL = valeur de timeout en ticks de 55 ms

00h si pas de timeout

DX = numéro de port, si bit 4 de AL = 1

ES:DI -> octet mémoire, si bit 4 de AL = 0

### Extinction de la machine

Cette fonction permet à un programme d'éteindre automatiquement la machine.

Disponible sur HP95 et HP100.

Valeurs d'entrée :

AH = 42h

### Contrôles de la fenêtre d'affichage.

Cette fonction gère les fonctions *Cursor Tracking* et *Alt Arrows*, permettant contrôler respectivement le déplacement de la fenêtre lorsque le curseur quitte celle-ci et ce même déplacement grâce à l'action des touches [Alt]+[Flèches] sur le HP95 et [Menu]+[Flèches] sur le HP100.

Disponible sur les HP95 et HP100.

Valeurs d'entrée :

AH = 45h

AL = combinaison des deux flags :

	Cursor Tracking	Alt Arrows
00 :	désactivé	désactivé
01 :	désactivé	activé
02 :	activé	désactivé
03 :	activé	activé

Accès direct :

HP95 et HP100 : octet situé en 40h:93h  
(format identique au paramètre).

### Contrôle du Timeout

Cette fonction permet de sélectionner le temps d'auto-extinction de la machine.

Disponible sur HP95 et HP100.

Valeurs d'entrée :

AH = 46h

BX = Temps, en Ticks de 55 ms

Une valeur de zéro désactive le timeout.

Valeur maxi = 65536 ticks, soit une heure.

La valeur par défaut (réinitialisée après chaque [Ctrl][Alt][Del]) est fixée à CCCh (3 minutes).

Accès direct

HP95 et HP100 : mot situé en 40h:A3h.

### Contrôle du contraste

Cette fonction permet de régler le contraste de l'écran.

Elle est disponible sur les deux machines, mais le HP100 possède en plus une deuxième fonction parallèle à la première permettant de d'obtenir 32 niveaux de contraste, au lieu de 16 sur le HP95.

Sélection

HP95 et HP100 :

Valeurs d'entrée :

AH = 47h

AL = 00

BL = Valeur du contraste (0 à 15)

HP100 uniquement :

Valeurs d'entrée :

AH = 62h

AL = 00

BL = Valeur du contraste (0 à 31)

Lecture

HP95 et HP100 :

Valeurs d'entrée :

AH = 47h

AL = 01

Valeurs de sortie :

AL = Contraste courant

HP100 uniquement :

Valeurs d'entrée :

AH = 62h

AL = 01

Valeurs de sortie :

AL = Contraste courant

#### *Accès direct*

HP95 : bits 4-7 du port E309h

HP100 : Suivant les modes d'écriture et de lecture, le HP100 semble travailler sur 3 registres différents :

- Lorsqu'il utilise la fonction 47h en lecture, ce sont les bits 4-1 du registre 23h qui sont lus.

- Lorsqu'il utilise la fonction 62h en lecture, ce sont les bits 4-0 de ce même registre qui sont utilisés.

- Lorsqu'il utilise la fonction 47h en écriture, ce sont les bits 4-1 du registre 0Fh qui sont modifiés.

- Lorsqu'il utilise la fonction 62h en écriture, ce sont les bits 4-0 du registre 1Fh qui sont modifiés.

#### **Contrôle du Volume Sonore**

Disponible sur HP95 et HP100.

#### *Sélection*

Valeurs d'entrée :

AH = 48h

AL = Volume (entre 0 et 3)

#### *Lecture*

Alors qu'il est possible de lire la valeur du contraste, il n'est pas possible de lire la valeur courante du volume. Pourtant on peut considérer que ce sont deux fonctions soeurs. Je trouve que cela manque un peu de cohérence...

#### *Accès direct*

HP95 : bits 3-2 du port E309h

HP100 : bits 7-6 du registre 1Eh

#### **Contrôle du port de sortie**

Cette fonction permet de sélectionner le port de sortie des communications : Interface série ou RS-232.

Disponible sur HP95 et HP100.

Valeurs d'entrée :

AH = 49h

AL = port de sortie

0 : RS-232

1 : Infra-Rouge

#### *Accès direct*

HP95 : bit 2 du port E30Ah

#### **Activation de la liaison série**

Cette fonction permet d'activer ou désactiver la liaison série.

Disponible sur HP95 et HP100.

Valeurs d'entrée :

AH = 4Ah

AL =

0 : Désactivation

1 : Activation

#### *Accès direct*

HP95 : bit 2 du port E301h

HP100 : bit 2 du registre 1Eh

#### **Synchronisation du timer**

Cette fonction permet de remettre le timer interne (double mot situé à l'adresse 40h:6Ch et remis à jour tous les 18.2 secondes environ.) à l'heure par rapport à l'horloge temps réel. Cette fonction doit être utilisée en fin d'exécution d'un programme modifiant la fréquence de mise à jour du timer (le programme devant avoir auparavant restauré la fréquence initiale).

Disponible sur HP95 et HP100.

Valeurs d'entrée :

AH = 4Bh

#### **Contrôle du Clavier**

Cette fonction permet de contrôler la langue du clavier : AZERTY ou QWERTY.

Disponible sur HP95 et HP100, mais utilisation restreinte sur la version Française du HP100.

#### *Sélection*

Valeurs d'entrée :

AH = 4Ch

AL = 00

BX = Code langue du clavier

0001h : Belge

0002h : Canadien Français

0004h : Danois

0008h : Finlandais (???)

0010h : Français

0020h : Finlandais (???)

0040h : Italien  
0080h : Hollandais  
0100h : Norvégien  
0200h : Portugais  
0400h : Espagnol  
0800h : Suédois  
1000h : Suisse Francophone  
2000h : Suisse Germanophone  
4000h : Royaume uni  
8000h : USA (QWERTY)

#### *Lecture de la valeur courante*

Valeurs d'entrée :

AH = 4Ch  
AL = 01

Valeurs de sortie :

AX = Code langue (voir ci dessus)

#### *Accès direct*

HP95 : mot situé en 40h:96h  
HP100 : mot situé en 40h:9Dh

#### **Lecture des valeurs admises par la Rom**

Valeurs d'entrée :

AH = 4Ch  
AL = 02

Valeurs de sortie :

AX = Combinaison des Codes ci-dessus

La version Française du HP95 retourne la valeur 8010h, indiquant bien que l'on peut sélectionner cette machine en QWERTY et en AZERTY.

Par contre, la version Française du HP100 ne peut pas être configurée en QWERTY. Ce qui est confirmé par le fait que cette fonction retourne la valeur 0010h.

#### **Controle de la langue de travail**

Cette fonction permet de contrôler la langue d'affichage des messages DOS et des écrans System Manager.

Disponible sur HP95 et HP100, mais utilisation restreinte sur la version Française du HP100, qui ne travaille qu'en Français.

#### *Sélection*

Valeurs d'entrée :

AH = 4Ch  
AL = 10h

BL = Code langue

00h : Anglais

01h : Français

#### *Lecture de la valeur courante*

Valeurs d'entrée :

AH = 4Ch

AL = 11h

Valeurs de sortie :

AL = Code langue (voir ci dessus)

#### *Accès direct*

HP95 et HP100 : octet situé en 40h:92h

#### **Controle du type de la machine**

Cette fonction permet au programme de vérifier que l'on travaille bien sur un HP95 ou un HP100 (une autre machine retournant des valeurs incohérentes).

Disponible sur HP95 et HP100.

Valeurs d'entrée :

AH = 4Dh

AL = D4h

Valeurs de sortie :

BX = 4850h : codes ASCII des caractères "HP"

CH = 1 (pour les HP95 et 100)

CL = type de la machine

1 : HP95

2 : HP100

DL = Numéro de version de la Rom

#### **Controle du Light Sleep**

Cette fonction permet d'activer ou désactiver l'auto-extinction de la machine, après un temps de non-utilisation défini par la fonction 46h (voir ci-dessus).

Disponible sur HP95 et HP100.

Valeurs d'entrée :

AH = 4Eh

AL = Valeur

0 : Désactivé

1 : Activé

#### *Accès direct*

HP95 et HP100 : octet situé en 40h:D6h

## Lecture de la tension des batteries

Cette fonction permet de lire la tension des batteries internes de la machine : Alimentation principale (piles 1.5V) et Backup (pile "bouton"). Elle ne permet pas de lire la tension de la pile interne des cartes Ram PCMCIA.

Comme pour le contraste, le HP100 possède une fonction parallèle. Celle ci donne les résultats sous une forme différente.

*Version pour HP95 et HP100.*

Valeurs d'entrée :

AH = 50h

AL = Batterie

0 : Piles Principales

1 : Pile de Backup

Valeurs de sortie :

AX = Valeur de tension  
(si FFFFh, erreur)

La valeur de la tension est donnée dans un code particulier. Afin de la connaître en volts, vous pouvez utiliser le code suivant :

```
mov ah,50h
mov al,pile
int 15h
and ax,00ffh
mov dl,05h
div dl
dec ax
cmp al,19
jnb suite
inc ah
cmp al,14
jnb suite
inc ah
label: ...
```

Ceci place en AL la tension exprimée en dixièmes de volts. Je n'ai pas fait de mesure exacte quant à ce résultat, mais il semble assez précis aux environs de la limite fatidique des 2 Volts !

*Version pour HP100 uniquement.*

Valeurs d'entrée :

AH = 60h

AL = Batterie

00 : Piles Principales

01 : Pile de Backup

Valeurs de sortie :

AX = Valeur de tension  
(si FFFFh, erreur)

Comme pour la fonction précédente, il est nécessaire de recourir à quelques lignes de codes afin d'obtenir une valeur plus facilement exploitable, cependant la méthode est ici plus simple, puisqu'il suffit d'effectuer l'opération suivante :

Contenu de AL

Valeur en Volts = 1.6 + -----  
37.333

Les valeurs 1.6 et 37.333 sont relativement empiriques, mais donnent un résultat juste au dixième de volt près.

Afin d'avoir le même type de résultat que pour la fonction précédente (c'est à dire exprimé en dixièmes de volts), vous pouvez utiliser la routine suivante :

```
mov ah,60h
mov al,pile
int 15h
mov dl,10
mul dl
mov dl,37
div dl
add al,16
```

## Sélection du type de piles

Cette fonction permet de sélectionner le type de piles installées dans la machine : Alkaline ou Cadmium-Nickel.

Disponible sur HP100 uniquement.

Valeurs d'entrée :

AH = 60h

AL = 03h

BL = Code type de piles

00h : Alkalines

01h : Cadmium-Nickel

*Accès direct*

bit 7 de octet situé en 40h:D4h

## Activation recharge batteries

Cette fonction permet d'activer ou de désactiver la recharge des piles lorsque l'adaptateur secteur est installé.

Disponible sur HP100 uniquement.

Valeurs d'entrée :

AH = 60h

AL = 04h

BL =

00h : Désactivation

01h : Activation

*Accès direct*

bit 3 de octet situé en 40h:D4h

### Lecture état des flags batterie

Cette fonction permet de lire l'état des deux informations précédentes : Type de piles et activation de la recharge ainsi que le temps de recharge des batteries.

Disponible sur HP100 uniquement.

Valeurs d'entrée :

AH = 60h

AL = 02h

Valeurs de sortie :

AL = valeurs courantes

bit 7 : type de pile

bit 3 : flag recharge

D'autre part, la fonction retourne deux valeurs en CX et DX indiquant le temps de recharge des batteries. N'ayant pas eu l'occasion de faire des tests avec l'adaptateur secteur, je ne connais donc ni la signification exacte des données (temps écoulé ou temps restant ?) ni leur format. La seule chose certaine est que ces résultats sont contenu dans les mots situés en 9000:F848 et 9000:F84A.

### Autres fonctions :

Si j'ai pu trouver les informations sur les fonctions précédentes, certaines me posent encore des problèmes. A titre indicatif en voici une liste succincte (elles concernent toutes le HP100 uniquement) :

- Fonction 51h : Aucune information précise, si ce n'est qu'elle semble concerner les batteries, puisque son code est situé entre les fonctions 50h et 60h. Elle admet un paramètre en AL

- Fonction 60h / sous-fonction 05 : Charge en 40:D0 et 40:D2 un paramètre situé en BX. La valeur située en 40:D0 est décrétementée 18.2 fois/seconde. Valeur initiale 10Eh=15 secondes. Je vous rappelle que la fonction 60 à trait aux batteries.

- Fonction 60h / sous-fonction 06 : Lit la valeur du bit 2 de l'octet situé en 40:CD.

- Fonction 61h : Lorsque le bit 5 de AL est désarmé, charge en AL les bits 6 et 7 de l'octet situé en 40:EB. Lorsque le bit 5 est armé, charge les bits 6 et 7 de AL dans cet octet. Cette fonction est insérée entre les deux fonctions traitant du contraste, mais l'octet situé en 40:EB est situé en plein milieu d'une zone concernant uniquement le clavier. Aucune action visible.

Je compte continuer mes recherches à propos de ces fonctions ce mois-ci. Mais si certains d'entre-vous veulent se joindre à moi dans l'exploration de ces dernières fonctions, ils sont bien entendu les bienvenus ! Affaire à suivre...

Jacques Belin (123)

## LE COIN DES CODES

Voir les articles de Guy Toublanc, parus ce mois ci, à propos des programmes ASSCOD28 et ASSCOD48.

<p>MNU (HP28) # B6BBh 55.5 octets</p> <p style="margin-left: 40px;">0123 4567 89AB CDEF sm</p> <p>000: 76C2 0E4A 20F1 0006 C66 001: 8124 262C 7A3B 3C8F BC5 002: 5E70 4C5B 1B22 70CA A35 003: 850A 4020 2F65 0A40 6C3 004: 20FC 140D E440 5E02 428 005: 0508 5060 120B 2270 FA9 006: FFE3 01A1 A009 F20 97A</p> <p>ASSCOD28 (HP28) # 9498h 705.5 octets</p> <p style="margin-left: 40px;">0123 4567 89AB CDEF sm</p> <p>000: 76C2 060B A0F1 4507 D09 001: 6C20 A211 000C B1DE B83 002: BB13 ABB1 F704 0376 8BE 003: 50A4 0203 1CB1 3765 576 004: 0877 A0B8 A11F B650 389 005: 76C2 009F 206B 6501 01C 006: BC91 4812 0A0D 603A D73 007: CD37 86F3 76C2 014D B3F 008: C148 120A 0D60 3ACD A15 009: 376C 2060 1203 B170 5DA 00A: 4812 02C0 40F7 F60F 446 00B: C140 09F2 076C 2028 0EB 00C: B30A 4020 2C04 006F DA3 00D: 60BD 1707 5070 6012 90D 00E: 083E 603B 170C D8C0 781 00F: 8EC6 0406 50BE 9E30 583 010: 9F20 33C6 009F 2099 2B0 011: 6500 9F20 0861 3CD5 04A 012: 117F 6133 B170 7A8C E75 013: 0E4A 2092 000E 6F6D CE0 014: 6262 7560 2467 2F63 997 015: 6475 6473 702A 302F 693 016: 7040 A402 0518 D38E 436 017: 4503 B170 8C8C 022A 1ED 018: 22ED 2118 4D11 B35C FEB 019: 0942 7075 0705 E020 BBE 01A: E602 071E 6040 6500 736 01B: 6F60 636D 306F 60C5 514 01C: 6D39 A170 F704 0D3F 35C 01D: 40AE 0E3A 4020 E602 041 01E: 0A40 20BD 170F FE30 EDF 01F: 5B95 0CA8 5037 650E C66</p>	<p>020: 4A20 5100 0C69 676E 9F7 021: 6560 2A30 22F6 5011 5DE 022: 920F FF00 D2F6 0016 326 023: C03C E138 2B13 1D17 06B 024: 05E1 70FC 1402 8B30 D67 025: A402 0E4A 2031 0003 899 026: 6F64 6563 702A 3E4A 6A5 027: 20B2 000D 2D2D 2D20 435 028: 2D2D 2D2D 202D 2D2D 30D 029: 2D20 2D2D 2D2D 2689 143 02A: D32F 6505 5E60 49CD FEF 02B: 376C 2026 9D3A 4020 C7E 02C: 1D17 0750 70C1 6D3D A82 02D: 2F60 3B17 05E0 20F7 7E9 02E: F60F C140 09F2 03B1 510 02F: 70CD 8C03 B170 8C8C 3E5 030: 03AC D376 C20C 56D3 255 031: A112 0751 2024 8E0C F5E 032: 16D3 09F2 076C 2083 C41 033: 9D38 E450 09F2 05EF B04 034: D148 1203 7650 E602 726 035: 05E0 20A3 822B 35C0 45C 036: EA9E 3FB6 5076 C20E 2E8 037: 6020 3B17 05E0 20F7 01B 038: F60F C140 636D 306F DFD 039: 6048 1202 C040 FC14 B1E 03A: 028B 3009 F206 B650 804 03B: 7512 0A11 209A 1704 405 03C: 8120 2C04 0CA8 5048 107 03D: 1202 F650 A402 0FC1 E83 03E: 408C A308 0440 2F65 B5E 03F: 0220 70D2 F605 0850 7A0 040: 119D 3D2F 60A4 0201 420 041: 1920 0001 0750 7060 EF3 042: 120E 4A20 1300 037F B6E 043: 6D6D 6560 2465 6023 738 044: 6F6E 6472 7F6C 6560 4F6 045: 2A30 237D 68B6 D35E 3ED 046: 020E 4A20 B000 0D3D 174 047: 3D38 39D3 8E45 0980 EA9 048: F35E 0202 8B30 22A2 B36 049: 2DA3 40A8 E603 ACD3 A4A 04A: 4FB6 076C 2060 1208 631 04B: 3BF3 3B17 07A8 C009 3C8 04C: F209 9650 ABAD 3C16 292 04D: D37D 8D35 E020 28B3 009 04E: 0AAA D350 8500 2BA0 D4A 04F: 7D8D 3A40 202C 040F A96 050: 7040 4812 0D02 7015 61E 051: F606 6FD3 5E02 069C 440 052: 2025 0008 F72F 408F 252</p>	<p>053: 1805 0CCD 8143 1301 E77 054: 7414 3131 1793 1931 9E9 055: 5B19 EA80 8186 8615 6CD 056: 8017 1160 CD54 E8D5 580 057: E521 C16D 3379 5009 21B 058: F20 355</p> <p>MANDEL (HP48) # 8AA6h 733 octets</p> <p style="margin-left: 40px;">0123 4567 89AB CDEF sm</p> <p>000: D9D2 00FE 8111 9201 C36 001: 2200 D9D2 079E 60D9 AA1 002: D203 3920 6000 0000 4CE 003: 6758 4010 CB9A 2881 24C 004: 304B 2A21 78A2 3991 F47 005: 6D9D 2002 9A29 F345 CCB 006: E4A2 0600 0003 2230 74E 007: 0BE3 5B21 309F 345B 528 008: 2130 FC43 6009 A29F 34B 009: 3453 2230 C2D5 06B3 085 00A: 16E8 F603 2230 6B31 CE9 00B: 6E8F 60A8 150A 8150 9A6 00C: 3223 0C2D 506B 316E 785 00D: 8F60 3223 06B3 16E8 4CE 00E: F60A 8150 A815 020C 181 00F: 46C1 C167 9470 FC2E 071 010: 4A64 0579 E60C CD20 EDA 011: 8730 0845 8FB9 7603 C22 012: 1E1A E51B CF20 0152 948 013: 71F1 D200 1577 77B1 63E 014: A6D5 8F15 0710 016F 31F 015: 17F3 104A E515 2715 FB6 016: 77B7 AA6D 59F1 5071 D61 017: 6F15 0716 F118 1547 9C5 018: 16F1 5071 FB62 0014 60D 019: 7D71 BFB2 0015 2710 213 01A: 1103 16F1 5671 0A10 E2C 01B: C11A 108A FA72 8110 B27 01C: 2119 AFA7 5711 1AAF A3E 01D: EB7A 11C7 C111 0211 61F 01E: 0AF6 7F01 1197 1511 206 01F: 1B71 0110 1AF6 7041 E65 020: 1001 12AF 6733 1118 A8D 021: 73E0 AF23 5000 0049 651 022: F6F0 CF46 0678 F7C9 57A 023: 11B0 5200 1423 4280 08B 024: 008A 2F3E 4140 1B6C E9F 025: 1001 FF02 0015 3715 A55</p>
---	---	--

026: 6771 9015 171B D020 6B2	058: 08C1 1920 0400 0119 1B3	027: 040B 2040 3375 0391 9C9
027: 0142 E414 06D0 F8F3 510	059: 2038 000F 8511 5562 DA0	028: 5088 130C 2A20 7100 570
028: 4150 1BEB 1001 5271 124	05A: 12C2 30CF 1469 7611 A57	029: 0A0A 036F 6465 6370 1FF
029: 6F16 F150 716F 16ED FF7	05B: B213 0B21 30 569	02A: 2A33 9150 C2A2 0B20 EBA
02A: 0140 1B8D 1001 4234 BA0		02B: 00D2 D2D2 D202 D2D2 CD6
02B: F300 08A2 5CE4 1401 84A		02C: D2D2 02D2 D2D2 D202 A56
02C: B341 001F C810 0153 438	ASSCOD48 (HP48)	02D: D2D2 D2D2 C541 6122 6ED
02D: 7156 7B7A 1517 1B69 1BD	# 420Fh 699 octets	02E: 704E C30C B916 D9D2 5FA
02E: 1001 42E4 1401 64D0 E04		02F: 0834 1688 1307 1040 108
02F: 1406 39EA C181 8417 B3D	0123 4567 89AB CDEF sm	030: 7FE3 0B9F 06CB D309 004
030: 0E44 94D4 3AC1 8184 8C4		031: 5236 B213 09FF 3087 D96
031: 170E 4194 D22A 7AAC 842	000: D9D2 0D48 8191 6C17 D56	032: 7359 FF30 5573 58DA C1A
032: 1818 4170 E449 4900 470	001: 9E60 D9D2 0A64 0588 AAB	033: 16D9 D208 5230 0831 7C6
033: AF0B F881 C01A 7A01 22E	002: 130B 9421 BB42 19FF 9A4	034: 6C2A 20B1 0000 20E0 3A1
034: 8408 1080 863E 0814 E7B	003: 30D0 040E C4A3 FD55 8A6	035: 236F 6272 7967 6562 03B
035: BF88 5081 0814 8128 A69	004: 02A1 7088 130B A421 53E	036: 7391 5068 3A25 9230 C3F
036: 08A3 9181 6BFA 8128 867	005: 56F1 4442 30D9 D209 280	037: 9B13 6AC4 22B2 130D 99F
037: 7080 8505 5084 0816 408	006: E040 C781 6301 2611 DD7	038: 9D20 E041 6E8F 60B2 7C7
038: 9F25 0AFE 9784 220A 1E4	007: 140C 7816 D9D2 0442 ACC	039: 1304 5256 2C23 02A1 40B
039: F1AF CA0D 4C0A 7A41 0AF	008: 30BB 7269 5236 0CA3 851	03A: 709F 1169 FF30 1B54 1D9
03A: 263F FBF2 0C5A E208 F90	009: 0B21 30D9 D205 3040 449	03B: 64D2 26EE 170D 9D20 FFE
03B: F5E5 D086 000B F801 CC5	00A: CD63 6399 16D9 D20A 296	03C: CA13 0952 36CA F06F EF7
03C: AF0A 7C81 C6CE F1B5 CD9	00B: DB46 CBD3 0B21 3034 F4B	03D: ED30 2265 0337 5039 AB6
03D: A000 1461 6414 2108 7FA	00C: 140C D636 3991 6D9D DFE	03E: 150B 2130 5E17 0627 6EE
03E: 8408 518F 2F83 18DA 6D4	00D: 20FD 040C BD30 B213 B92	03F: 26FE F302 2650 BD37 4C6
03F: C831 0000 0000 0000 084	00E: 0571 40CD 6363 9916 88A	040: 02C2 3012 2705 0803 FE5
040: 0000 0000 0000 0000 A04	00F: D9D2 0F80 40CB D30B 74B	041: 1227 02CE 30CB D304 DF7
041: 0000 0000 0000 0000 384	010: 2130 21B4 6CD6 3639 4D6	042: 3370 7E31 65D7 2611 A78
042: 0000 0000 0000 0000 D04	011: 916A B526 4423 0A3B 22E	043: 920F FF00 1BE3 0C2A 94F
043: 0000 0000 0000 0000 684	012: 46B2 1305 7A50 EE25 000	044: 2053 000A 0A03 7F6D 76B
044: 0000 0000 0000 0000 004	013: 0BB7 26FE D301 2040 C9A	045: 6D65 6024 6560 236F 433
045: 0000 0000 0000 0000 984	014: 7FE3 0442 306A C309 9B6	046: 6E64 7274 FC65 602A 203
046: 0000 0000 0000 0000 304	015: FF30 8773 555C 26CB 841	047: 3023 7D62 0016 FE22 F42
047: 0000 0000 0000 0000 C84	016: 9160 BB26 0CA3 0B21 5A9	048: 6C2A 20B0 000D 3D3D D5C
048: 0000 0000 9FF3 0119 8F4	017: 308C 170B 2130 73C3 25E	049: 3E04 16E8 F603 3456 A45
049: 2038 CFF1 1920 26FF 7B5	018: 5ADB 46AA C35C 2A20 0CA	04A: B525 0731 4130 A50D 6F4
04A: F322 300E D305 1850 3EC	019: 9200 0E6F 6D62 6275 E2A	04B: 2C26 18A3 0D9D 2085 47B
04B: 3223 0A81 50CC D204 16D	01A: 6024 672F 6364 7564 ACD	04C: 2301 F514 1FB2 69C2 286
04C: C000 1431 74E7 8FB9 016	01B: 7370 2A30 2F31 61AC 89A	04D: A2A5 1419 FF30 5273 F87
04D: 7601 3110 134C CD20 D2D	01C: 4229 FF30 5573 5731 503	04E: 50CA 30B2 1308 C170 C1B
04E: 1457 050E 2CFF C3CF E0F	01D: 41ED 2A2C B9A2 AEC8 59D	04F: 0F51 6B9F 066B 316F A98
04F: F24C FF39 CFFA 9CFF 0BE	01E: 1F80 407F E30C 6836 361	050: E226 0E51 6433 70F7 7B0
050: 33DF F94D FFB4 DFF7 2C5	01F: A333 6FED 30CA F06F 372	051: 6C1C 2A20 3100 0742 30C
051: 7DFF 29DF F8AD FFAA 5AC	020: ED30 BBF0 6FEF 30F3 33A	052: 5F42 4028 3026 B316 F2F
052: DFFC AEFB 44DF F1BD 7C1	021: 1616 2D25 03D4 3AF2 143	053: 6365 0FED 30E8 E304 D75
053: FF00 0000 7D51 3414 37B	022: 500D 470B D370 2A17 E8C	054: E761 0BB2 6391 506B AA6
054: 2C01 7913 7E21 0881 FCA	023: 0C2A 2051 000C 6967 B2B	055: 3163 9150 7947 0731 657
055: 8F14 1641 428A 8811 C31	024: 6E65 602A 3021 2270 691	056: 4171 040C 9B26 7566 364
056: 18CA C013 1147 CA14 984	025: E0E3 0119 2000 010C 216	057: 0B21 30 791
057: 16CD F8F2 D760 1118 6C8	026: BD30 CC95 0160 4571 E50	

Le Journal JPC est le bulletin de liaison entre les membres de l'Association "PPC Paris", régie par la loi de 1901. Le Club est éditeur de JPC, et son siège social est au 56, rue Jean-Jacques Rousseau, 75001 Paris.

PPC Paris est le représentant Français de HEX (Handhelds European Clubs EXchange), la fédération des principaux clubs Européens d'utilisateurs de calculateurs HP.

La maquette de ce numéro a été préparée et réalisée par Jacques Belin et Asdin Aoufi.

Les dessins sont de Jean-Jacques Dhénin et Paul Courbis.

Les informations et programmes parus dans ce journal sont publiées "Tels quels" et ne peuvent en aucun cas engager la responsabilité de Hewlett-Packard ou de PPC Paris. Hewlett-Packard se réserve le droit de ne pas répondre aux questions concernant le sujet de certains articles.

Les programmes publiés peuvent être utilisés librement. Cependant, ils ne peuvent être vendus ou fournis dans un ensemble commercialisé, sous quelque forme que ce soit, sans l'accord écrit de l'auteur ou de PPC Paris.

Directeur de la publication : Jacques Belin  
Numéro ISSN : 0762 - 381X

Veuillez adresser toute correspondance à :  
PPC Paris, BP 604, 75028 Paris Cedex 01.

Imprimé par Paris Copie, 4 rue Linné, 75005 Paris