JPC

MAI 1993

NUMERO 86

Le numéro 40 F

A PROPOS DU CLUB

HP28

HP48

HP95 / HP100

EDITORIAL

Avant de parler de ce JPC, revenons un peu sur le numéro précédent. Nos anciens adhérents savent qu'un doux délire nous prend vers le troisième numéro de chaque année. En effet, il nous arrive de passer à cette époque une ou deux informations totalement fausses. Toutefois, cette année nous vous avons tout de même laissé un petit indice... En effet, ceux qui ont été étonnés de n'avoir jamais entendu parler de Roland Sivipos se sont peut-être aperçus qu'il s'agissait de l'anagramme de *Poisson d'Avril* ! Il n'y a donc pas de possibilité de modifier vos machines comme cela a été indiqué. Enfin, pas encore, à notre connaissance... Pour ceux qui ont cru qu'il s'agissait du *Mode Exam* sur les prochaines HP, sachez que nous avons récupéré cette information étonante dans le numéro de mars du journal du club Australien, qui est pourtant bien informé... Enfin, et là c'était une erreur de frappe, il n'existe pas encore de tome 3 de *Voyage...*, mais seulement une troisième édition. Excusez nous pour cette étourderie.

Parlons de ce numéro maintenant. Vous y trouverez plus de texte et moins de programmes qu'habituellement. Cela est dû au hazard des arrivées des articles. Par ce même hasard, nous avons recu deux articles (écrits par Guy Toublanc et Christophe Vaillant) traitant du même sujet, la réutilisation des routine internes de la Rom de la HP48. Toutefois, ces articles étant très diférents par l'approche et les outils utilisés, nous avons décidé de les passer ensemble. A vous de choisir celui qui vous conviendra le mieux.

Mais le grand événement de ce mois est l'annonce officielle du HP100LX (dont vous avez déjà entendu parler sous le nom de Cougar). Bien sûr, vous trouverez dans ce journal une présentation de cette nouvelle machine. Vous avez bien sûr remarqué que la numérotation est passée à trois digits. On peut évidement y voir un symbole de la liaison entre les calculateurs de poches (avec une numérotation inférieure à 99) et les micro ordinateurs MS-DOS portables HP-110 et HP-150 d'autrefois. Toutefois cela nous amène à nous poser la question : pourquoi le HP95 ne s'est il pas appelé tout de suite HP100 ? A vous de trouver la réponse en lisant ce journal...

Sachant que la première présentation officielle du HP100 (et de la nouvelle gamme micro) a été programmée pour le 25 mai à l'hippodrome de Vincennes, que va donc faire Hewlett-Packard le 14 juin à EuroDisney ? Réponse dans le prochain numéro, bien sûr !

Le bureau

LIBRES PROPOS

Cet article a pour but de poser quelques questions qui appelleront, je l'espère, des réponses de la part d'autres lecteurs ou de la rédaction (*NDLR : Pourquoi pas ?*). Quant à moi je fournis mes réponses car il faut être honnête et que ce ne soit pas toujours les mêmes qui répondent ou se sentent attaqués par le biais de questions justement sans réponses. Je cherche juste ici à faire progresser le fonctionnement du club qui peut-être est resté un peu trop le même depuis dix ans. Il contient aussi quelques réflexions qui si elles ne sont pas neuves auront le mérite de venir de la part d'un lecteur assidu de JPC.

- Ou en HEX ? La rédaction se plaint à juste titre d'ailleurs du manque d'articles du journal et qu'elle doit faire beaucoup de travail au dernier moment. Pourquoi alors ne pas insérer 2 ou 3 articles des clubs d'HEX, puisque si j'ai bien compris tous les clubs formant ce groupement s'envoient mutuellement leurs parutions. Nous savons bien que l'anglais est la langue utilisée pour ce type d'échange mais je ne crois pas que cela ferait peur à nos lecteurs pour qui se mettre ou se remettre à la langue de Shakespeare ne serait pas si mal (quoique je parle peut-être pour moi). En tous les cas les programmes seraient tous dans les mêmes langages : user-RPL, système-RPL ou LM. Et là pas de problèmes ...

Reponse : HEX n'existe actuellement que par les échanges de journaux. Ils sont disponibles à la lecture lors de nos réunions mensuelles. Si nous sommes tout à fait disposés à passer certains de leurs articles dans JPC, il faut savoir que (pour les membres du bureau), traduire et taper un article provenant de ces journaux prendrait autant de temps que de créer un article personnel. Quant à la langue utilisée, nous ne recevons qu'un seul journal qui soit rédigé en Anglais : DATAFILE. Tous les autres sont écrits dans d'autres langues : Allemand, Hollandais, Flamand, Danois... Si ils arrivent tous avec un "English Summary", j'ai donc tout de même un petit peu de mal à comprendre les articles, surtout ceux écrits en Finlandais ! Encore une fois, si vous connaissez des personnes pouvant nous aider... Un des autres problèmes est que les programmes sont rarement accompagnés des sources, ce qui nous demanderait soit de contacter les auteurs, ce qui peut être très long, soit désassembler les programmes, ce qui est également long et peut aussi aller à l'encontre des souhaits de l'auteur.

- Qu'en est-il aujourd'hui des numéros 73/75 de JPC qui contenaient, parait-il pas mal d'articles pour la HP48 ? Si ces numéros sont difficilement retirables à cause du prix de revient, pourquoi ne pas les proposer dans la boutique du club par correspondance comme cela est fait pour la documentation du kit de développement ? Cela limiterait les frais pour le club et permettrait aux adhérents d'avoir des infos supplémentaires. De plus seules les personnes vraiment intéressées commanderaient ces numéros et on éviterait ainsi au club de grever son budget en envoyant à tout le monde ces journaux.

Reponse : Comme toi, de nombreux adhérents n'ont pas reçu ce numéro (qui ne contenait, en fait, que très peu d'articles sur HP48). Comme tu le sais, c'est une partie des éléments faisant partie du litige nous opposant à Jean-Paul Nalin. Pour diverses raisons, nous avons obtenu un report de l'audience, qui ce tiendra le 22 Septembre. Ce n'est donc qu'après conclusion de cette affaire que nous pourons décider de la façon dont nous distribuerons les très rares numéros qui sont en notre possession. Mais, entre temps, nous avons déjà remporté une première victoire, puisque nous venons de récupérer notre Laserjet (qui nous a permis d'imprimer la maquette de ce numéro).

- Pierre S. de Sacy (572) nous parle dans JPC 84 de la librairie RECOVER de HPeed. Mes questions sont : quelle est cette librairie ? D'ou vient elle ? Comment a t'il pu se la procurer ? D'une façon plus générale, certains adhérents ont surement des librairies ou des programmes recueillis à droite et à gauche et qui pourraient satisfaire d'autres adhérents. Un bon moyen d'échange pourrait se faire toujours par le club, qui recueillant de cette façon des librairies ou des programmes, les stockerait sur disquette et les redistribuerait via la boutique. Je comptais donner l'exemple en envoyant une librairie dénommée ASM+ (n°804) de J.Stark. Cette librairie était un assembleur tournant sur HP48 donc, avec des mnémoniques HP. De plus aucune librairie de ce type ne se trouve sur les Goodies Disks. Je sais en outre qu'il existe une librairie faite par HPNinja (je crois ; mais ne me demandez pas qui c'est !) qui était aussi un assembleur, mais là mon propos revient à ma question précédente : c'est une librairie qui circule et qui pourrait surement profiter à d'autres personnes si ceux qui en avaient un exemplaire (ou ses concepteurs) l'envoyaient au club. Mais revenons à ASM+; il avait quelques défauts du type : non compatible avec le désassembleur DBUG (librairie 1210) du même auteur + C.Bourgeois (qui se trouve sur le Goodies Disk n°7), il n'acceptait pas les étiquettes ... Mais il avait le mérite d'exister. J'avais donc l'intention de l'envoyer mais je pense ce n'est plus la peine depuis la sortie des utilitaires de Detlef Mueller !!!

Reponse : Nous distribuons déjà les Goodies Disks, qui représentent la grande majorité des programmes disponibles sur HP48. Cependant, il est vrai que de

nombreux adhérents possèdent, échangent et utilisent des programmes et des librairies provenant d'autres auteurs. Le bureau essaye de récupérer ces programmes afin de les redistribuer, après avoir établi une sélection des meilleurs dans chaque catégorie. Nous connaissons les librairies dont tu nous parle mais, comme de nombreuses autres, elles présentent quelques défauts (qui n'ont rien à voir avec la qualité de programmation). Par exemple, la librairie ASMFLASH *de HPNinja (en fait Christophe Nguyen) utilise certaines mnémoniques assembleur différentes de celle définies par HP. Or, pour le journal, nous préférons éviter d'utiliser un trop grand nombre d'outils différents car cela ne ferait que compliquer le travail des lecteurs qui souhaitent étudier et modifier les sources. Un de nos problèmes est aussi qu'il nous arrive d'avoir des programmes qui ne sont pas accompagnés d'une documentation (car elle est perdue, ou n'a jamais existé). C'est principalement pour ces raisons que nous n'avons pas encore la possibilité de distribuer une disquette totalement remplie de programmes HP48 externes au club (hors Goodies). En ce qui concerne le HP95, nous avons déjà une importante bibliothèque de programmes, mais elle doit être organisée avant de vous en faire profiter. Ceci devrait être fait dans les prochaines semaines et devrait bénéficier du même type de distribution que les goodies Disks.*

Pourquoi les horaires de réunion du club (qui a lieu tous les premiers samedi de chaque mois) sont-ils de 16 à 19h ? S'ils sont adaptés aux parisiens, il est impossible pour un provincial de venir y assister. En effet, il faut aujourd'hui pour venir de province par le train, une moyenne de 4h, que ce soit de Marseille (TGV), Brest (TGV), Bordeaux (TGV), Strasbourg... En prenant le premier train vers disons 8h du matin, un provincial arriverai à midi à Paris. Là il lui faudrait attendre jusqu'à 16h pour la réunion et il devrait repartir vers 17h pour attraper le dernier train vers 18h, ce qui le ferait arriver vers 22h chez lui. Alors rester 1h en réunion, il est bien évident qu'aucun provincial ne fera jamais le voyage et le sacrifice ! Ce qu'il serait par contre possible de faire c'est de déplacer les horaires de 14 à 17h, quitte pour ceux qui le peuvent, à rester après. Cela permettrait ainsi aux provinciaux d'assister à une réunion complète.

Réponse : Il est vrai que de très nombreux adhérents ne peuvent pas venir aux réunions car ils habitent en province. Mais tu te trompe en disant qu'aucun provincial ne fait le voyage pour les réunions. Par exemple, Guy Toublanc, qui habite à Lille, fait ce genre de déplacement depuis de nombreuses années (donc avant le nouveau TGV !), un mois sur deux en moyenne). Il pourra te confirmer qu'il est possible de s'occuper entre midi et 16h. Par exemple en faisant la tournée des librairies du quartier Latin ! En ce qui concerne le retour, a tu pensé à prendre un train de nuit ? Ceci te permettrais donc en plus de d'avoir un petit aperçu des folles nuits Parisiennes ! Plus serieusement, il est évident que les horaires ne pourront jamais convenir à tout le monde, quoi qu'on y fasse. Un des autres problèmes est que toutes les salles du centre Verdier sont utilisées le samedi après midi. Notre salle est utilisée, avant nous, par un autre Club. Je ne sais donc pas si il serait possible de changer des horaires qui ne dépendent pas que de nous. Mais enfin, si vous êtes nombreux à demander un changement, nous pouvons toujours étudier la question.

C'est ainsi que j'en arrive tout naturellement à mes réflexions qui méritent qu'on s'y attarde pour l'avenir du club.

Pourquoi le club ne reçoit plus beaucoup d'articles ? Une solution réside je crois dans l'évolution du comportement des programmeurs ou plutôt des utilisateurs des machines programmables. Aujourd'hui les gens préfèrent consommer des programmes plutôt que d'en faire, c'est une constatation. L'époque ou l'on bidouillait sur des machines est assez loin et repose (dans le monde HP) sur une seule machine ou presque : la HP-41. Cette machine avait l'avantage d'être accessible autant par son prix que par ses possibilités. Ses limitations initiales, faible mémoire, petit écran, ROM de faible capacité, devenaient aussi des avantages par le fait qu'on en arrivait facilement, du moins finalement, à bout. Tout le monde écrivait quelque chose car tout le monde pouvait le faire. Mais aujourd'hui qui peut me dire qu'il connait les 2100 fonctions de la HP 48, qu'il arrive à remplir sa mémoire (32 K) d'un programme de son crû ? Qui ne sent pas ridicule devant les programmes fait par des professionnels qui tiennent sur des cartes de 128K ?

On peut faire facilement le rapprochement avec le monde des PC. Au début des années 80, certains programmeurs arrivaient seuls à faire des programmes et il ne fallait pas être informaticien! Qui aujourd'hui fait un programme seul ? Non, se sont des équipes entières, l'une pour le graphisme, l'une pour le son ... Comment le petit programmeur indépendant peut lutter ? Pourquoi se dit il finalement " m'embéter " à faire tel programme alors qu'il existe déjà sur le marché, qu'il est surement mieux fini que je ne pourrais le faire et surtout que je peux avoir de suite ? Ce raisonnement est à rapprocher du monde des HPistes. Quand je regarde ce qui existe déjà et ce que les pros font, je me demande ce que moi je vais bien pouvoir faire.

Voila le problème auquel sont confrontés les gens de clubs comme le notre. Nous vivons dans un monde de consommation ou tout nous est apporté à condition

qu'on y mette le prix, mais comme c'est plus facile que de se creuser la tête ... Alors y a t'il une solution ?

Oui je crois. Elle est simple et tient en un mot : la PASSION. Il faut être passionné pour continuer à y croire quand tout se retourne contre vous, pour continuer à travailler et à faire partager ses programmes, trouvailles ... aux autres quand plus personne ne répond. Mais la PASSION a ses limites et, c'est pourquoi, quand j'ai appris que le club envisageait de ne plus accepter la réhadésion des personnes qui n'enverraient pas un programme dans l'année, je me suis dit que c'était peut-être la seule solution pour que le club s'en sorte et retourne à ses sources (les programmeurs qui ont fait sont succès) afin de repartir sur des bases saines. Un club ou le bénévolat est de mise ne peut vivre que par et grâce à ses adhérents ; il faut donc retrouver ceux qui en veulent, ceux qui sont prêts à fournir un peu d'effort pour que la PASSION ne s'éteigne pas.

Malgré ceci, je ne suis pas réfractaire au progrès (je crois l'avoir montré plus haut) et pour que tout le monde n'aie plus d'excuses, je demande à ce que vous lisiez mes deux articles de ce mois ci !

A bientôt au plus grand nombre dans ce journal !

Réponse finale : Tu as pu constater dans le numéro précédent (que tu n'avais pas vu quand tu as écrit cet article) ainsi que dans ce numéro, que notre action commence à porter ses fruits. Le danger, maintenant, est que les adhérents pourraient croire que la situation est déjà revenue à la normale et n'envoient pas les articles qu'ils auraient envoyé si la situation s'était agravée. Rappelez vous que si notre retard de parution est stabilisé, il nous faut beaucoup plus d'articles pour assurer une sortie régulière et un contenu suffisement varié.

Christophe Vaillant (523)
Jacques Belin (123)

UN NOUVEAU CLUB EN ESPAGNE

Si la plupart des pays Européens ont eu au moins un club HP connu à l'étranger, l'Espagne manquait encore à l'appel. Aujourd'hui cette "erreur" est corrigée, puisque depuis environ six mois la cité de Barcelone accueille un club reconnu sous le nom de *HPUC* (HP User's Club).

Travaillant principalement sur HP48, il édite mensuellement une revue (au format A5, d'une vingtaine de pages) appelée *Forum HP*. Doté d'une excellente qualité d'impression, il se particularise par une très belle couverture quadrichrome montrant nos machines préférées. Cependant, les coûts d'impression en offset étant ce qu'ils sont, les différents numéros que nous avont reçu bénéficient de cette unique couverture. On ne peut pas tout avoir !

Comme pour les autres clubs étrangers, nous avons bien sûr conclu un accord d'échange des journaux. Si vous désirez les rejoindre personnellement, vous pouvez les contacter à l'adresse suivante :

HPUC Barcelona
Apdo 10.080
08031 Barcelona
Espagne

Dernier détail : ce journal est bien sûr écrit en Espagnol !

Jacques Belin (123)

CA BOUGE CHEZ HP !

Et pour une fois, ce n'est pas une image, puisque les services de HP France situés Porte de Champeret ont déménagé à Boulogne Billancourt (juste à côté du "donjon" d'une certaine chaine de télévision privée...).

Le Support Technique Téléphonique Calculateurs a suivi ce déménagement. Il vous faudra donc noter son nouveau numéro de téléphone :

(1) 46 10 18 69

où Bernard Challiel se fera un plaisir de répondre à toutes vos questions, ou presque !

Jacques Belin (123)

HP28

G. Toublanc — Réponses et factorielles — *6*

QUESTIONS ..... REPONSES

A propos de son article *traitement de données statistiques* (JPC 84 page 4), Paul Jebeily lançait un appel pour une commande LINE en assembleur. Dans un tout prochain JPC je proposerai la réponse à cet appel.

Dans ce même JPC 84 je faisais appel, à mon tour, pour une utilisation de la routine d'adresse #1E9BA qui m'avait valu beaucoup de plantages. Avec un peu de temps j'ai trouvé la solution.

Ce programme assembleur réserve un espace mémoire pour une chaîne en précisant en C(A) l'espace à réserver c'est-à-dire le nombre de caractères * 2 et auquel on ajoute 10 (5 pour le prologue et 5 pour la longueur). La routine appellera SAVTPTR pour sauver les pointeurs D0, D1 et B(A) et C(D). Au retour D0 pointe le premier futur caractère, R0(A) contient l'adresse du prologue et R1 celle de la longueur. Si le prologue chaîne a bien été stocké le programmeur doit préciser la longueur, contrairement à la routine MAKE$N de la HP48.

Aussi je suis en mesure de donner une version du programme FFACT28S en utilisant cette routine. Cela réduit la longueur du programme. J'en ai profité pour intégrer le SPEED. Je ne donne que le listing de la partie modifiée en précisant que l'utilisation des registres R0 et R3 est inversée.

```
gosbvl  #04f27      * pop#
cd0ex
rstk=c              * sauve D0
d0=(5)  #fff00      * pour
lchex   f           * le
dat0=c  1           * SPEED
c=rstk
d0=c                * restitue D0
c=a     a           * longueur de n!
r3=c                * sauve longueur
c=c+c   a           * nombre de quartets
c=c+con a,5         * ajoute 5 pour long.
r2=c                * sauve long. + 5
c=c+con a,5         * + 5 -> total réservé
gosbvl  #1e9ba      * res_str
d0=d0-  5           * pointe longueur
a=r2                * restitue la longueur
dat0=a  a           * et stocke
d0=d0+  5           * avance sur le 1er car
c=r3                * nombre caract. de n!
rstk=c              * et sauvevegarde
cd0ex
rstk=c              * sauve position départ
r1=c                * idem
d1=c                * pointe 1ère position
c=r3                * nombre de caractères
c=c+c   a           * nombre de quartets
gosbvl  #5042       * pour write_zeros
```

Vous trouverez la chaîne de codes à assembler dans le *coin des codes*.

Guy Toublanc (276)

HP48

LA HP48 MENE GRAND TRAIN

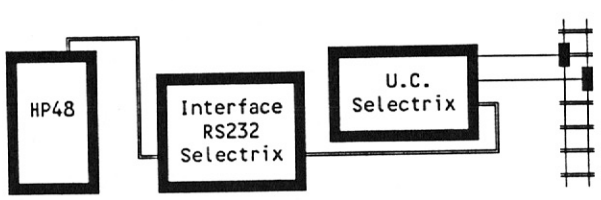
La HP48 vous permet de faire beaucoup de choses, à part calculer des fonctions mathématiques ésotériques. Comme ceux qui étaient présents à la Réunion Anniversaire de janvier dernier ont pu le voir, elle peut même faire marcher un train électrique... Si on lui ajoute quelques accessoires supplémentaires, bien sûr. Je vais essayer de vous donner ici un aperçu du comment afin de permettre aux adeptes de ces deux sports (HP48 et train électrique) de les pratiquer simultanément.

Le matériel

Tout d'abord, il convient de préciser qu'il n'y a aucune modification à apporter à la HP48 : tout passe par l'intermédiaire de la sortie série, le câble série étant dès lors un accessoire indispensable. Côté "train" il faut un réseau adapté à la commande numérique "Selectrix" de la firme *Trix*, les éléments minimaux étant : l'unité centrale (Réf.66801), l'interface RS232 (Réf.66824), et un ou deux décodeurs (Réf.66826 pour l'échelle HO - 1/87 - ou Réf.66827 pour l'échelle N - 1/160 -) montés sur locomotives. Tout cela coûte à peu près autant qu'une HP48SX (le prix des locomotives elles-mêmes non compris).

Le montage de l'installation est des plus simples : outre la prise secteur, l'unité centrale comporte deux sorties à relier directement aux rails du réseau (16 Volts alternatif) et deux prises DIN sur l'une desquelles (n'importe laquelle d'ailleurs puisqu'elles sont en parallèle et équivalentes) doit être branchée l'interface RS232.

Il suffit alors de brancher le câble série de votre HP48 à cette interface, la seule précaution à prendre pour obtenir une transmission correcte étant de croiser les connexions 2 et 3, la seule autre connexion utilisée étant la 7. Le montage global est illustré ci-dessous :



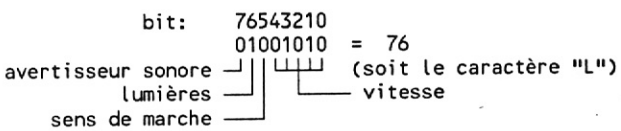
Codage des commandes

Une fois le matériel en place et la HP48 connectée, il ne reste plus qu'à envoyer les commandes sous forme appropriée pour obtenir un résultat, soit le déplacement d'un train, soit l'actionnement d'un accessoire. Là encore, tout comme avec le matériel et son branchement, Trix a joué la carte de la simplicité : les messages acceptés par l'interface sont toujours sous la forme de deux octets complets.

Le premier octet représente l'adresse de l'élément auquel est destinée la commande transmise. Cet élément peut être aussi bien une locomotive qu'un groupe d'accessoires (quatre aiguillages, ou relais, lumières, etc...). Le nombre maximum d'éléments adressables est de 127: le codage de l'adresse doit être effectué en binaire, utilisant les bits 0 à 6 de l'octet. Le bit 7 détermine s'il s'agit d'un message de commande (il est alors à "1"), ou d'interrogation (il est dans ce cas à "0").

Dans le cas des commandes proprement dites, c'est-à-dire celui où on désire envoyer un ordre à l'élément en question, cela signifie que l'adresse de cet élément devra être systématiquement augmentée de la valeur "128" : ainsi, pour envoyer un ordre à la locomotive d'adresse "9", le premier octet du message devra avoir la valeur binaire "137". Sur la HP48, il suffit d'introduire cette dernière valeur puis d'exécuter la commande CHR qui fournit au niveau 1 de la pile le caractère correspondant, à savoir "≤", sous forme de chaîne à un seul caractère. Il faut également noter que l'adresse "0" correspond à l'unité centrale et permet, au moyen d'une commande à deux octets nuls, d'effectuer une immobilisation immédiate et générale de tous les trains: elle correspond au bouton "panique" disponible sur l'unité centrale elle-même.

Le deuxième octet constitue la commande proprement dite. Sa structure est sensiblement plus complexe et sera différente selon qu'il s'agit d'une commande pour locomotive ou pour accessoire (tel que des aiguillages). L'octet de commande pour une locomotive est défini selon la structure suivante :



Les bits 6 et 7 doivent être mis à "1" pour enclencher le fonctionnement du dispositif correspondant (avertisseur sonore/sifflet ou lumières allumées), ou être mis à "0" pour arrêter leur fonctionnement. Le bit 5 détermine le sens de marche : "0" pour marche avant, "1" pour marche arrière. Il convient de préciser qu'une locomotive pourvue d'un décodeur Selectrix est "orientée", c'est-à-dire qu'elle possède un avant et un arrière, même s'il s'agit d'une machine moderne (électrique ou diesel) a priori symétrique : ceci est tout à fait différent de l'alimentation en 12 V

continu/deux rails, où le sens de marche est indépendant de l'orientation de la locomotive.

Les bits 0 à 4 déterminent la vitesse à laquelle circule le train parmi 31 valeurs distinctes ("00000" correspond à l'arrêt, "11111" = 31 à la vitesse maximale). Ainsi, à titre d'exemple, pour donner l'ordre à la locomotive d'adresse 5 de rouler en avant à vitesse 10, lumières allumées, le message à envoyer sera : 10000101-01001010 , soit: 133-74 , ou encore la chaîne de caractères: "ΣJ".

Dans le cas d'une commande pour décodeur d'accessoires, tels que des aiguillages, chaque bit de l'octet de commande est affecté à un accessoire: "0" correspondant à l'accessoire "éteint" (ou l'aiguillage dans une première position), "1" à l'accessoire allumé (ou l'aiguillage dans l'autre position). Un décodeur déterminé ne peut commander que quatre accessoires, utilisant quatre des huit bits disponibles. Il est ainsi possible de mettre deux décodeurs sous la même adresse, l'un recevant les bits 0 à 4, l'autre les bits 5 à 7. Globalement chaque adresse peut être attribuée indifféremment à une locomotive ou à un groupe d'accessoires: cela permet de commander par exemple 99 trains et 224 accessoires, ou 50 locomotives et 616 accessoires. Plus que confortable...

Programme pour commande de locomotives

Le programme KRM1 ci-après est un exemple simple permettant d'illustrer toutes ces explications, et de montrer la simplicité du procédé de commande à partir de la HP48.

KRM1 :
Cksum: #EB7Bh ; 121 octets

```
«
  SWAP  128  MOD  128  +  CHR  OVER
  IF  0  <
  THEN  96
  ELSE  64
  END
  ROT  ABS  32  MOD  +  CHR  +
  XMIT  DROP  CLOSEIO
»
```

Avec: IOPAR = { 9600 0 0 0 0 0 } dans le même répertoire.

L'utilisation de ce programme nécessite deux paramètres: l'adresse de la locomotive ou motrice au niveau 2 de la pile, et la vitesse désirée au niveau 1, cette vitesse étant positive pour la marche avant, négative pour la marche arrière. Les lumières, lorsque disponibles, sont systématiquement allumées (dans le cas de feux avant-arrière / blancs-rouges réversibles, le décodeur se charge de l'inversion automatique).

L'exécution du programme ne donne aucun résultat sur la pile qui descend donc de deux niveaux. Cela a l'avantage de permettre une programmation de commandes successives, nous en verrons un exemple plus loin.

Eventuellement (et ce serait sans doute plus dans la logique de la HP48), on peut inverser la position de ces deux paramètres (vitesse en premier au niveau 2 et adresse ensuite au niveau 1) : il suffit alors de supprimer le SWAP tout au début du programme. La séquence 128 MOD a pour seul rôle d'éviter toute adresse au-delà de 127. Cela fait, la valeur 128 est ajoutée à l'adresse et le résultat converti en octet de caractère pour la première partie du message à envoyer.

La vitesse est ensuite examinée, un test sur son signe déterminant le sens de marche et ajoutant soit 96 ("01100000": feux allumés, marche arrière) soit 64 ("01000000": feux allumés, marche avant) à la valeur de la vitesse, celle-ci étant auparavant ramenée à l'intervalle 0..31 par la séquence ABS 32 MOD (on peut faire mieux, mais après tout ceci n'est qu'un programme de démonstration). Quoiqu'il en soit, la valeur ainsi obtenue est transformée en caractère et simplement ajoutée au caractère d'adresse pour fournir une chaîne à deux caractères. Il suffit ensuite d'envoyer cette chaîne par l'interface série au moyen de la commande XMIT : cette commande évite le protocole Kermit qui ne doit pas être utilisé dans le cas présent. L'interface du système Selectrix n'accepte que les messages à deux octets en binaire pur.

La transmission retourne alors une valeur "1" si elle est réussie (ou "0" dans le cas contraire) : ce résultat est ici simplement ignoré et éliminé par un DROP. Une procédure de détection d'erreur peut être ajoutée, pour signaler le "non-envoi" par un message éventuel mais, encore une fois, je n'ai pas jugé utile d'alourdir le programme avec une telle procédure. La toute dernière instruction du programme, CLOSEIO, ferme la sortie série pour éviter toute consommation d'énergie superflue.

Pour résumer l'emploi de KRM1, la séquence :

```
5  12  KRM1
```

fait avancer la locomotive "5" à vitesse "12".

Une séquence telle que :

```
15  9  22  -4  KRM1  KRM1
```

entraîne une marche arrière à vitesse "4" de la loco "22" et une marche avant à vitesse "9" de la loco "15"; noter l'ordre inversé dans l'exécution des commandes lorsque les paramètres successifs sont introduits avant l'exécution de KRM1. Il est tout à fait possible d'utiliser KRM1 en sous-programme et d'écrire un programme exécutant une séquence complexe commandant plusieurs trains simultanément, quelque chose dans le genre :

```
«
  2  3  KRM1  4  WAIT
  5  2  KRM1  2  WAIT
  2  0  KRM1  6  WAIT
  5  0  KRM1  2  -6  KRM1  8  WAIT
  2  0  KRM1
»
```

Cette séquence dure environ vingt secondes et peut être schématisée ainsi :

```
          0....5....10....15....20 secondes
Loco 2 : +3-----0      -6---------0
Loco 5 :      +2--------0
```

Il est même possible de programmer une accélération ou un freinage progressif, mais pour ce genre de simulation il est préférable d'écrire des programmes mieux adaptés.

L'interrogation

Il est parfois intéressant voire nécessaire de connaître la situation d'un élément du réseau (locomotive ou accessoire). C'est tout à fait possible en envoyant au système Selectrix un message dont l'octet d'adresse est inférieur à 128 (bit 7 à "0"). Le deuxième octet (nécessaire) du message peut avoir n'importe quelle valeur: il n'en est pas tenu compte. La réponse du système Selectrix consiste en un octet, structuré selon le mode du deuxième octet dans les messages de commande tel que décrits plus haut. Le programme LCK ("Loco CheckK") permettant cette interrogation est simple et très court :

LCK :
Cksum: #9250h ; 37,5 octets

```
« CHR  0  CHR  +  XMIT  SRECV  DROP  NUM »
```

Il suffit d'introduire au niveau 1 de la pile l'adresse de l'élément à contrôler et d'exécuter LCK qui remplace cette adresse par une valeur numérique correspondant à la situation de cet élément. Ainsi, pour la locomotive d'adresse "12" circulant en marche avant à la vitesse "7", feux allumés, on obtiendra :

```
1:    12   -[LCK]->   1:    71
```

la valeur "71" correspondant à l'octet "01000111".

Le rôle essentiel de LCK est d'être utilisé comme sous-programme au sein d'applications plus avancées, aussi s'agit-il d'un programme "minimal", sans vérifications ni détection d'erreurs. Il se contente de convertir l'adresse en caractère (sans ajouter 128 cette fois-ci), de lui adjoindre un octet nul pour former une chaîne de deux caractères, puis de transmettre cette chaîne à l'interface Selectrix par la commande XMIT, suivie immédiatement de SRECV pour lire l'octet que renvoie cette interface. Il convient de noter ici que le résultat de la transmission ("1" ou "0" laissé par XMIT au niveau 1 de la pile) est estimé à "1" et sert de paramètre pour SRECV, indiquant qu'un seul octet doit être reçu.

SRECV fournit par ailleurs un "1" ou un "0" au niveau 1 de la pile pour signaler le succès de la réception: ce résultat est ignoré et éliminé par un DROP et le caractère reçu du système Selectrix est converti en valeur numérique pour interprétation et utilisation ultérieure.

Un mot à propos de la valeur logique fournie par XMIT et SRECV dans les programmes décrits ici. Il peut paraître assez cavalier de les ignorer comme je le fais : en fait l'expérience a montré jusqu'ici que cela ne pose guère de problème. Depuis un an et demi que j'utilise ces programmes (ou plutôt des programmes analogues) je n'ai jamais eu un seul "raté". Mais même si cela se produisait, l'influence sur l'exécution des programmes et sur la HP48 est de toute façon mineure et sans conséquence. Quant à l'effet sur les trains (songez que les collisions entre convois sont tout à fait possibles avec le système Selectrix, comme dans la réalité), si un train ne répond plus aux commandes qui lui sont envoyées, le bouton "panique" est toujours disponible...

Pour terminer il n'est sans doute pas inutile de préciser, pour ceux qui seraient intéressés par un tel projet mais qui auraient des difficultés à obtenir le système Selectrix, qu'il existe d'autres systèmes de commandes numériques pour trains électriques, tels que le FMZ de *Fleischmann* ou le *Märklin-Digital*, également susceptibles d'être pilotés par une HP48 selon des méthodes analogues.

Robert Pulluard (561)

RECURSIVITE ET GRAINS DE SABLE

Pour les personnes ayant déjà des connaissances ou ayant déjà écrit des programmes à bases récursives, cet article ne sera pas une révélation, mais pour les autres personnes...

Avant tout, qu'est-ce qu'un programme récursif, ou plutôt une routine récursive ? C'est une routine qui s'appelle elle-même. Pour que cela puisse fonctionner, il faut à cette routine un ou plusieurs paramètre(s) d'entrée et une condition de sortie, et un ou plusieurs paramètres de sortie.

Prenons pour exemple le calcul d'une factorielle. Comme dans bien des cas, il s'agit de correctement poser le problème:

$n! = (n-1)!.n$, pour tout n>0.

Ecrit sous une autre forme :

$fact(n) = fact(n-1)*n$

Voilà résumé en une ligne la partie principale du programme récursif.

En Pascal cela donne, une fois écrit dans le détail, le programme suivant :

```
function fact(n:longint): longint;
begin
  if n:=0
    then fact:=1
    else fact:=fact(n-1)*n;
end;
```

Un simple appel par `writeln(fact(5));` donnera 120.

Et pour la HP-48 :

```
'Facto'
« → n
  « n
    IF 0 ==
    THEN 1
    ELSE n DUP 1 - Facto *
    END
  »
»
```

Bien sûr, pour calculer quelque chose d'aussi simple qu'une factorielle, la méthode est lourde, puisque l'on connait les bornes de début et de fin, il suffit de faire une boucle simple avec un compteur. Mais il y a des problèmes que l'on ne peut résoudre qu'avec un programme récursif, comme les tours de Hanoï, ou la courbe de Von Koch (fractal). Le problème des tours de Hanoï est une utilisation typique de la récursivité: on ne connait pas le nombre de coups pour le résoudre, mais on connait la méthode pour y arriver.

Et que tombent les pixels...

A titre d'exemple, Voici un programme qui n'a d'autre but que d'être un petit divertissement graphique, où l'on observe passivement "tomber" des grains de sable dans les deux dimensions de l'afficheur de la HP48. En fait ces grains de sable sont symbolisés par des pixels qui ne seront affichées que lorsque ceux-ci seront tombés et ne bougeront plus. Les grains de sable observent les quelques règles suivantes :

`X` représente un pixel déjà allumé, et `O` la position actuelle du grain de sable.

Le grain peut descendre d'un cran, s'il n'y a pas d'obstacle en dessous.

```
..O..
.....
.XXX.
```

Le grain ne peut pas descendre plus bas, mais vers la droite.

```
.....    .....
..O..    .XO..
.XX..    ..X..
```

Le grain ne peut pas descendre plus bas, mais vers la gauche.

```
.....    .....
..O..    ..OX.
..XX.    ..X..
```

Le grain ne peut plus descendre du tout : il sera affiché à l'écran.

```
.....    .....    .....    .....
..O..    ..OX.    .XO..    .XOX.
.XXX.    .XX..    ..XX.    ..X..
```

Le programme utilise une boucle récursive (`Tst`), qui a pour paramètres d'entrée les coordonnées actuelles du pixel, et comme condition de sortie : le pixel ne peut plus tomber. Le programme se lance avec "Principal", et au début l'écran va se remplir d'obstacles de façon aléatoire. Les grains de sable commencent leur chute à partir du coin gauche, pour remplir l'écran vers la droite (dans la mesure du possible).

Bien sûr, après avoir essayé ce programme, vous le trouverez un peu lent, et il serait plus rapide écrit d'une manière non récursive, mais j'ai trouvé cette solution plus élégante.

```
DIR
  Principal
    « Init Ligne Tomber Fin »

  Init
    «
      { (1;64) (131;1) X 0 (0;0) FUNCTION Y }
      'PPAR' STO ERASE
      { # 0d # 0d }
      PVIEW (1;0) (131;64) BOX
    »

  Ligne
    «
      1 130
      FOR i
        i 57 RAND * 2 + R→C DUP
        (0;8) + LINE 130 RAND * i 2 / 1 +
        R→C DUP (8;0) + LINE
      NEXT
    »

  Tomber
    «
      { # 0d # 0d }
      PVIEW 2 130
      FOR i
        i 1 R→C Tst
      NEXT
    »

  Tst
    «
      0 → p f
      «
        DO
          IF p (0;1) + DUP PIX? NOT
          THEN
            Tst
          ELSE
            DROP
            IF p (1;1) + DUP PIX? p 1 + PIX? OR NOT
            THEN
              Tst
            ELSE
              DROP
              IF p (-1;1) + DUP PIX? p 1 - PIX? OR NOT
              THEN
                Tst
              ELSE
                DROP 1 'f' STO
              END
            END
          END
        UNTIL f
        END
        p PIXON
      »
    »

  Fin
    « LCD→ 'PPAR' PURGE »
END
```

Philippe LECKLER (434)

Rappel sur les tours de Hanoï : Les tours de Hanoï se joue avec trois piquets, sur l'un desquels sont empilés plusieurs anneaux de taille décroissante. Il s'agit de déplacer, en un minimum de coups, des anneaux d'un piquet à un autre en utilisant le troisième comme piquet intermédiaire. Mais il faut suivre les règles suivantes: ne déplacer les anneaux que un par un et il faut toujours mettre un anneau sur un autre plus grand. Le nombre de coups va dépendre du nombre d'anneaux à déplacer.

RPL48 TOOLKIT

Le directory le plus passionnant dans les Goodies Disks 8 est sans aucun doute celui dénommé DETLEF, car il offre aux programmeurs des outils exceptionnels. Ce sont 3 librairies (*pour et sur* HP48) qui contiennent dans l'ordre : un assembleur/désassembleur system RPL et un assembleur/désassembleur LM, un débugueur système- RPL, un assembleur/désassembleur de librairies. Ces outils vont ôter toutes les tâches ingrates pour certains, ou les rendre possible pour d'autres : qui n'a jamais râlé devant le fait qu'il fallait allumer son PC pour travailler (et qu'il ne pouvait pas emmener avec soi), qui n'a jamais désespéré de ne jamais faire un bon programme en system RPL par exemple parce qu'il n'avait simplement pas de PC !

Heureusement, tout est aujourd'hui possible avec ces 3 programmes (ecrits par Detlef Mueller et Raymond Hellstern) qui occupent tout de même 60k environ sur la HP48 (car il y a aussi une table d'adresses de 25k). Ces 3 programmes utilisables immédiatement (bien que distribué en Shareware pour 30 Deutsche Marks soit environ 100F) sont les équivalents de la disquette n°4 des Goodies Disks distribuée au club sous le nom d'IDS 48. Si ces IDS 48 ne pouvaient profiter qu'à des possesseurs de PC, le mal est aujourd'hui réparé avec cette disquette n°8. Tout le monde va pouvoir écrire des programmes directement sur sa HP, en s'épargnant le travail ingrat (recherche des codes à la main, assemblage manuel,

recherche des adresses dans divers livres ...). Tout est ici sous la main, en mémoire. Tout le monde va pouvoir recopier les programmes system RPL ou LM de Guy Toublanc ou d'autres facilement. La fin des limitations matérielles et le début de la créativité spirituelle !

Mais détaillons un peu plus ces programmes (que je n'avais pas encore en vous proposant dans ce même numéro mon article *Détournements Mineurs*.

la librairie RPL.LIB (n°1234) contient :

- →RPL : un compilateur système RPL
- RPL→ : un décompilateur système RPL
- →COD : un assembleur LM en mnémo HP
- COD→ : un désassembleur LM en mnémo HP
- DAn : désassemble n instructions
- DA1 : " 1 "
- DAXY : " de X à Y
- $→ : convertit une chaîne en hexdump
- →$: convertit un hexdump en chaîne
- EC : entre dans le Catalogue des Entrées. Donne tous les points d'entrée officiels de chez HP; permet de rechercher une adresse donnée
- E←→A : donne le nom correspondant pour une adresse officielle (c'est pourquoi pas mal de SYSEVALs de J.M. Ferrard ne s'y trouvent pas puisqu'ils ne sont pas des points d'entrée officiels) et vice-versa
- CC : Catalogue des Caractères sur la HP

Je vais vous donner un exemple pour les commandes RPL : reprenez mon 1° exemple de mon article *Détournements...* Vous avez la chaîne suivante à assembler par ASC→ ou ASS : D9D20D004011920FFFFF33750B2130. Là il n'y a pas nécessité de posséder ces programmes car vous avez tout. Vous faites d'abord $→ et vous obtenez : <3h> <FFFFFh> External, ce qui est normal. Vous lancez ensuite RPL→ et vous obtenez la traduction directe en system RPL :

```
::
  THREE
  # FFFFF
  SUB$
;
```

Un autre exemple ? Facile, construisez la liste suivante avec la commande SWAP à l'intérieur : « SWAP ». Faites 1 GET et vous devez obtenir au niveau 1 de la pile la commande SWAP. Lancez RPL→ et vous obtenez la chaîne system RPL suivante :

```
::
  CK2
  SWAP
;
```

Cela veut dire que la commande User Rpl SWAP se décompose en System Rpl en deux codes, le premier vérifiant la présence de 2 éléments dans la pile, le deuxième effectuant le véritable SWAP en LM. Vous recompilez ensuite avec →RPL et vous obtenez 2 Externals que vous pouvez transformer en chaîne de codes avec →$. Vous obtenez alors D9D2008A8132230B2130. On en déduit les 2 adresses dont vous pouvez vérifier l'appellation en faisant #3223h E←→A puis #18A80h E←→A (la première adresse étant le SWAP en LM dont vous pouvez vérifier l'adresse dans le livre de J.M- Ferrard). Vous pouvez bien entendu vérifier de la même manière les adresses #2D9Dh qui donne docol et #312Bh qui donne semi. On comprend maintenant pourquoi le system RPL est plus rapide : dans cet exemple avec SWAP il ne vérifie pas la présence d'éléments dans la pile. D'ou plus de rapidité mais plus de dangers aussi.

Nous allons maintenant utiliser le débugueur.

Cette librairie contient :

- SDBG : lance le débugage avec un programme system RPL au niveau 1 de la pile
- →SST : pas à pas d'une instruction
- →IN : idem ci-dessus même dans les sous-programmes
- SNXT : montre la prochaine instruction
- DBRK : met un point d'arrêt

Un exemple, facile. Essayons la commande User RPL DISP. Ecrivons 1 1 « DISP » 1 get et nous obtenons aux niveaux 2 et 3 le chiffre 1 et au niveau 1 la commande DISP. Passons un court instant dans la librairie *RPL* et décompilons la avec RPL→ (on obtient ::CK2&Dispatch ONE DODISP ;), recompilons la avec →RPL pour passer en système RPL (on obtient External <1h> External <0h> External... (à ne pas éditer surtout pour voir la suite car risque de plantage)) et revenons à DEBUG. Lançons SDBG : on voit nos 1 passer aux niveaux 1 et 2 de la pile et devenir %1. Appuyons sur →SST et la première commande CK2&Dispatch s'inscrit (il faut 2 éléments). Réappuyons et ONE DODISP s'inscrit et s'éxécute. Notre 1 est affiché en haut à gauche, normal. Appuyons une dernière fois sur →SST le mot 'Finished' s'écrit un court instant et on revient au début.

Refaisons les mêmes opérations mais au lieu d'employer →SST, utilisons →IN. CK2&Dispatch s'inscrit puis après un nouvel appui sur →IN, il s'inscrit Into: DODISP et on rentre dans le sous-programme et ainsi de suite. Si on continue avec →IN, nous allons

parcourir tous les sous- programmes composant la routine. Vraiment super ! Plus rien ne peut nous échapper, les entrailles de la machine sont à nous. C'est vraiment fantastique.

Enfin un dernier mot sur la librairie LIB. Celle-ci se charge de construire une librairie à partir d'un directory ou le contraire, à savoir 'casse' une librairie existante et en construit un directory. Ses commandes sont nombreuses et entre autres :

- D→LIB : construit une librairie
- LIB→D : 'casse' une librairie
- RNLIB : renomme une librairie
- CHLID : change le numéro d'une librairie
- RLIB : rappelle une librairie
- PGLIB : purge une librairie
- LIBP : donne une carte de la librairie.

Voila, vous savez maintenant tout et je voudrais pour terminer faire quelques remarques :

- Ces outils sont les plus puissants qui existent d'après moi sur la 48. Qu'ils soient distribués en Shareware n'est pas un problème car il est très facile d'envoyer un mandat international vers l'Allemagne et de s'acquitter moralement et financièrement (100F c'est peu) de sa dette envers celui qui a fait ces programmes. On peut ainsi se désister des autres outils tels ASC→, ASS ... mais que leurs auteurs en soient remerciés car ce sont eux qui avaient montré le chemin.

- ces outils ayant été mis au point par la même personne, ils échangent parfaitement entre eux des données system RPL ou Langage Machine (ce qui n'était pas le cas avec des assembleurs et des désassembleurs provenant des sources différentes.

- Ces outils sont identiques à ceux mis au point par HP et que l'on trouve sur la disquette n°4 des Goodies Disks (ou IDS 48 du club). Tout le monde va donc parler le même langage : ceux qui ont un PC et ceux qui n'en ont pas. Il permettent une programmation aisée même sur la HP48 notamment avec l'assembleur dont je n'ai pas parlé : il utilise tout ce qui est possible à savoir étiquettes, symbolisme, utilisation des pseudo-codes etc... (à titre de comparaison tout ceci était impossible avec la librairie ASM+).

- Sur cette disquette n°8 il n'y a pas de documentation sur la programmation en system RPL ni sur l'assembleur (mnémoniques HP légèrement différentes des mnémos utilisées par P.Courbis et S.Lalande dans leur livre Voyage au Centre de la HP48, utilisation des étiquettes...). Par conséquent il vous faudra commander au club au moins la disquette IDS 48 pour ceux qui ont un PC et une imprimante et pour les autres la disquette plus la documentation.

- Je prévoie enfin ceux qui vont me dire qu'il n'ont pas de PC et qu'ils ne pourront jamais charger ces programmes dans leur HP. Je leur répondrai que c'est un fausse excuse car moi-même au début je n'avais pas de PC (j'envoyais mes articles sur papier et la rédaction les recopiait) et je me suis fait mon propre câble HP-PC pour moins de 100F, fer à souder compris ! Ensuite, comme je n'avais pas de copains qui pouvaient me prêter un PC portable, je suis allé chez un revendeur de PC (exactement la librairie Charlemagne (NDLR : Tiens le nom de code de la HP48 !) à Toulon : qu'elle en soit ici remerciée) et après accord avec le vendeur (qui vendait aussi des HP et donc me comprenait plus facilement) je me suis branché sur un PC et j'ai chargé tout ce qui m'intéressait ; j'y suis même retourné 2 fois ! Vous voyez, il n'y a pas de problèmes pour un passionné.

J'espère maintenant voir dans les colonnes de JPC plein de programmes en system RPL ou en assembleur : pas besoin qu'ils soient immenses ... au début !

Christophe Vaillant (523)

DETOURNEMENTS MINEURS

Et ne lisez pas ce que je n'ai pas écrit ! Plus sérieusement, cet article s'adresse aux personnes qui aimeraient faire des programmes rapides et puissants, sans pour autant passer par le langage machine, car elles ne le maîtrisent pas assez bien. Il y a une solution pourtant : l'emploi des SYSEVALS.

Pour accéder à ceux-ci rien de mieux que le livre de J.M. Ferrard : Les Secrets de la HP48. Celui-ci contient en effet plusieurs milliers de SYSEVALS, propres à créer des programmes plus rapides que le RPL, plus puissants aussi car offrant des 'instructions' à la fois plus précises et plus nombreuses. C'est en fait le paradis de celui qui veut programmer vite, fort, précis, sans pour autant passer par le LM.

Malgré la multitude de possibilités offertes par ces SYSEVALS, il se peut que l'on ne trouve pas exactement la fonction qu'on aurait aimé. Il y en bien une voire plusieurs qui se rapproche de ce qu'on aurait voulu mais ...

C'est pourquoi, à partir d'exemples simples, je vous propose de faire un détournement mineur de ces SYSEVALS, afin qu'ils correspondent à votre besoin.

Premier exemple :

Vous avez une chaîne de caractères au niveau 1 de la pile dont vous voulez enlever les 2 premiers caractères. Soit par exemple au niveau 1 de la pile :

```
1 : "abcdef"
                →enlève2caract. →
1 : "cdef" .
```

La première chose que vous faites est de regarder dans les manuels de la HP48 pour y constater qu'une telle fonction n'existe pas. Vous vous précipitez donc sur le livre cité plus haut, rubrique *Chaînes de Caractères* p.173, et là, de même, vous constatez malheureusement que le Syseval espéré n'existe pas. Seul le Syseval dénommé del1 (voir p.175) s'approche mais il ne détruit que le premier caractère de la chaîne : or vous en auriez voulu deux.

Qu'à cela ne tienne, nous allons détourner ce Syseval pour le mettre à notre goût. En fait détourner est un bien grand mot : nous allons seulement voir comment a été composée cette routine puis la modifier pour qu'elle corresponde à notre besoin. Il ne faut pas oublier en effet que cette routine est en Rom et que nous pouvons juste la lire ; après l'avoir lue nous la transfèrerons en Ram et la modifierons. Le tour sera alors joué !

L'adresse de del1 est #0516Ch. Nous allons nous y rendre par le mini-moniteur. Tapons donc la séquence de touches suivantes :

```
_ ON-D
_ CLR
_ ENTER
_ touche de direction HAUT 15 fois
_ + jusqu'à arriver à l'adresse désirée
```

On lit alors le code suivant :

```
_ D9D20 (#0516Ch)
_ 30040
_ 11920
_ FFFFF
_ 33750
_ B2130 (#05185h)
```

On s'arrête ici car on a tout de suite reconnu les codes D9D20 et B2130 qui indiquent le début et la fin d'un programme : ici détruire le premier caractère d'une chaîne. Il nous faut donc déterminer les codes qui sont au milieu. Première opération : les retourner, puisque le microprocesseur de la HP48 les retourne avant de les lire. Cela nous donne :

```
_ 04003
_ 02911
_ FFFFF
_ 05733
```

Coup de chance, on reconnait la dernière adresse 05733, tout simplement car on l'a vue un peu plus haut en cherchant dans les routines : elle se trouve 7 lignes plus haut que notre del1 et elle se dénomme : sub(st,2s). Avec la syntaxe employée par J-M.F. on comprend tout de suite que l'on doit trouver en amont de cette adresse 2 entiers système : vérifions.

L'adresse 04003 correspond à <2h> : voir pour cela le chapitre *Entiers Système* p.41. L'adresse pour désigner <FFFFFh> existe bien (p.43) mais les programmeurs de la Rom ont décidé de faire autrement en codant cet entier système par la voie normale, à savoir : prologue 02911 puis valeur FFFFF.

On s'aperçoit donc en fait que le Syseval del1 d'adresse #0516Ch fait appel à un autre lSyseval sub(st,2s) d'adresse #05733h, en fixant les 2 entiers système, paramêtres de cette routine (attention : mon but ici n'est pas de vous expliquer comment marche ces routines, J-M.F. le fait bien mieux que moi).

Pour nous, il nous faut reloger cette routine en Ram en modifiant finalement qu'une adresse : celle de l'entier système <2h>, par celle de l'entier système <3h> (voir chapitre Entiers système p.41) soit #0400Dh (puisqu'on veut enlever 2 caractères).

Notre routine que l'on pourra nommer del2 s'écrira donc :

```
_ D9D20
_ D0040  ← code remplacé
_ 11920
_ FFFFF
_ 33750
_ B2130
```

Vous mettez ces codes dans une chaîne que vous assemblez par ASC→ de JPC 79, ou par ASS du livre *Au Centre de la HP48* par P.Courbis et S.Lalande, ou par ASM48 de la librairie TOOLS que l'on trouve sur un des Goodies Disks (je ne me rappelle plus lequel) disponible au club maintenant.

Après assemblage vous devez obtenir au niveau 1 de la pile : <3h> <FFFFFh> External. Vous mettez ceci sous le nom del2 et vous n'avez plus qu'à l'essayer en mettant une chaîne d'au moins 2 caractères dans la pile.

Voila, c'était un exemple bien simple mais qui repose sur 2 choses : que l'on connaisse bien comment la HP48 code ses objets tels les entiers système ou les réels etc ... et soit que l'on connaisse les adresses contenues dans les routines (il n'y en a que quelques milliers !) soit que l'on ait un peu de flair ou de chance pour se dire : j'ai déjà vu cette routine quelque part, ou, vu ce que je demande, cette routine a de grandes chances de se trouver dans tel chapitre.

Deuxième exemple :

Vous avez une chaîne au niveau 2 de la pile et vous voulez la couper en deux à un endroit bien précis que vous indiquez au niveau 1. Soit par exemple :

```
2 : "abcdef"
1 : 3
```

Si vous dénommez la fonction Coupe, celle-ci devrait donner :

```
2 : "abc"
1 : "def"
```

Vous regardez alors si une fonction RPL ou si un Syseval existe, ce qui n'est pas le cas. Vous envisagez alors de faire le programme correspondant mais vous avez peur qu'il ne soit trop long, en temps surtout. Vous fouillez dans les Sysevals du chapitre *Chaînes de Caractères* et vous en trouvez un qui ressemble pas mal à ce que vous désirez, c'est `CutRC(st)` d'adresse `#127A7h`. Mais ce Syseval a 3 défauts : il coupe au niveau d'un Retour Chariot et toujours au premier rencontré, il retourne les 2 chaînes résultantes, et de plus il enlève le RC (faites un essai pour voir).

Vous rentrez dans le mini-moniteur et vous vous rendez à l'adresse ci-dessus et vous trouvez (j'indique des numéros de ligne pour qu'on puisse s'y retrouver plus facilement après) :

```
1  _ D9D20
2  _ 88130
3  _ A6656
4  _ 9FF30
5  _ 1B546
6  _ 66226
7  _ CB916
8  _ D9D20
9  _ 7A726
10 _ 63650
11 _ FED30
12 _ B2130
13 _ CA130
14 _ E0E30
15 _ 76E26
16 _ 33750
17 _ D0040
18 _ E9330
19 _ FED30
20 _ 22650
21 _ D6D26
22 _ B2130
```

Assurément cette routine est bien plus compliquée que la précédente, et pourtant il suffit d'un peu de courage et d'habitude pour s'y retrouver. Allons-y ensemble.

Que remarquons nous : tout d'abord qu'un autre programme est imbriqué (lignes 8 à 12) et que deux adresses nous sont connues (l. 16,17). Quant au reste c'est un puzzle qu'il nous faut reconstituer.

Nous savons que cette routine opère sur une chaîne située au niveau 1 de la pile. Avec un peu de chance elle va donc effectuer sur celle-ci quelques traitements 'pilaires' avant tout autre chose. Regardons le chapitre *Opérations sur la Pile* de notre livre (Les Secrets...). La chance aidant, le deuxième Syseval d'adresse `#03188h` correspond bien à notre ligne 2 (après retournement) : nous avons là un `DUP`. Nous trouvons encore 1 page plus loin que l'adresse `#627A7h` de notre ligne 9 correspond à un `DROP;DUP`. Hélas c'est tout ce que nous voyons pour le moment. Mais nous avons un autre indication : cette routine agit sur le premier Retour Chariot. Page 50, chapitre `Caractères` nous regardons l'adresse du Retour Chariot (code ASCII 10) et nous en trouvons un en `#6566Ah`. Comme par hasard, nous retrouvons ce code en ligne 3 de notre routine. Et là, que remarquons nous ? Et bien que nous avons en fait déjà vu le code de la ligne 4 : il correspond à l'entier système `<1h>` dont l'adresse précède juste celle que l'on a vue dans l'exemple précédent (celle du `<2h>`). Et oui, il faut de la curiosité pour regarder à chaque fois un peu plus que l'adresse qui nous intéresse momentanément et de la mémoire !

Nous avons déterminé maintenant les 4 premières lignes de notre routine et nous nous demandons ce qu'elle peut bien effectuer après. C'est là qu'un peu de flair (et vraiment qu'un peu car c'est quasiment évident) nous vient en aide : que peut faire cette routine avec deux chaînes , un caractère (le RC) et l'entier `<1h>` (je vous rappelle qu'elle doit couper la chaîne en deux au niveau du RC) ? Et bien un `POS` évidemment ! Tous les `POS` se trouvent au chapitre *Chaînes de Caractères* p.175 et qu'est-ce qu'on y trouve : que l'adresse `#645B1h` (ligne 5) correspond à `SPOS(2st,s)`. Celle-ci demande en effet 2 chaînes , un caractère et une position à partir de laquelle il faut commencer la recherche ; le résultat étant un autre entier système. Que faisons nous donc ? Et bien nous allons regarder au chapitre *Entiers Système*. Et nous y

AH ! VOUS ECRIVEZ

Vous vous sentez en verve, mais vous ne savez pas sous quelle forme "l'équipe de rédaction" souhaite recevoir votre prose. C'est ici que se trouvent les réponses à vos questions.

Dans la mesure du possible, vous devez nous envoyer vos écrits sur support magnétique (carte, cassette ou disquette) dans le format lisible directement pour votre machine. Vous pouvez taper vos articles sur IBM PC, mais n'utilisez pas de traitement de texte (WORD...). Utilisez plutôt un éditeur simple manipulant des fichiers en ASCII pur, sans codes d'enrichissement. Soyez sans crainte, nous vous retournerons vos biens après copie.

Si vous n'avez pas accès à un IBM, et seulement dans ce cas là, vous pouvez à la rigueur nous envoyer des disquettes provenant de MacIntosh.

Si vous ne pouvez pas utiliser de support magnétique, ou ne pouvez vous rendre aux réunions, alors et alors seulement faites le sur papier.

Que ce soit sur une feuille de papier, ou sur support magnétique, ne dépassez pas 50 caractères par ligne.

Pour nous épargner du travail, insérez dans votre texte les commandes de formattage suivantes (et non les commandes du formatteur HP-71) :

"^" centre un titre, par exemple :
^TITRE

"\" (CHR$(92)) marque le début et la fin d'un paragraphe. Par exemple :

\Début de paragraphe exprimant le contenu de vos idées qui, même si vous en doutez, intéressera certains des membres du Club. Surtout si vous vous sentez débutant. Les articles pour débutants écrits par des débutants sont ceux qui manquent le plus. Fin de paragraphe.\

Pour écrire une accolade ({ ou}), il faut doubler l'accolade, une accolade simple ayant une signification spéciale dans l'édition de JPC.

Les utilisateurs de HP-71 utiliseront le fichier CHARLEX, qui a été souvent listé dans le coin des Lhex, pour utiliser les caractères accentués du jeu Roman8.

Les utilisateurs de HP-48 doivent nous envoyer les programmes RPL sous deux formes : Binary, plus ASCII avec *Translate Code* : 3. Les programmes composés de plusieurs variables doivent être dans un répertoire à transférer sur disquette et dans les deux modes. Si vous insérez dans le corps des articles le texte des programmes, veuillez faire précéder ce texte de `{rpl` et le faire suivre de `{endrpl`.

Les utilisateurs de HP-95 et MS-DOS nous transmettront les sources de leurs programmes accompagés des fichiers compilés. Il nous indiquerons le nom et la version du compilateur ou de l'interpréteur qu'ils ont utilisé.

Le Bureau

PPC PARIS SE REUNIT
UNE FOIS PAR MOIS

Comme vous le savez peut être déjà, PPC Paris se réunit une fois par mois, en plein coeur de Paris. Amenez votre matériel, votre bonne volonté et vos idées ! Plus vous en apporterez, et plus vous en trouverez chez vos collègues de PPC.

Ces réunions se déroulent de manière très libre, aucun ordre du jour, discussion ou autre n'étant imposé. Un membre du bureau est toujours présent. Ainsi, si vous désirez remettre votre article tout frais au Journal, si vous avez des suggestions à faire, si vous voulez vous procurer des anciens numéros de JPC, ce sera en principe toujours possible.

Si donc cela vous intéresse, n'hésitez plus un seul instant, venez nous rejoindre tous les premiers samedis de chaque mois (sauf en période de vacances scolaires) au :

Centre de Jeunesse et de Loisirs Jean Verdier
11 rue de Lancry
75010 Paris

et en montant au deuxième étage, vous entendrez des éclats de rire et des discussions passionnées vers la salle 215. Attention, toutefois, de venir entre 16 et 19h.

Pour l'accès en métro, trois possibilités s'offrent à vous :
- Métro Strasbourg Saint Denis :
Sortie porte St Martin / Bd St Denis, coté pairs
- Métro République :
Sortie Bd St Martin, coté pairs
- Métro Jacques Bonsergent :
Sortie Bd Magenta, coté impairs.

Ah, j'oubliais ! JPC est (souvent) distribué en avant première lors de ces réunions... A bon entendeur, salut !

Les dates des prochaines réunions sont :

Samedi 6 Mars 1993
Samedi 3 Avril 1993
Samedi 5 Juin 1993

NOUS EN AVONS

La coopérative du Club vous propose :

- Des **anciens numéros** de JPC (Préciser les numéros).
- Des **années complètes** de numéros de JPC (février à janvier).
- La **Programathèque HP-71**, regroupant tous les Lex et programmes pour HP-71 et HP-75 parus à ce jour dans *JPC*. Elle vous est livrée avec un catalogue décrivant brièvement tous les programmes.
- La **JPC Rom** pour HP-71, comportant plus d'une centaine de fonctions en assembleur (mathématiques, programmation structurée, utilitaires systèmes...). Joindre une Eprom vierge (64 Ko) à votre règlement. Il est possible d'y ajouter vos propres programmes.
- Les **Goodies Disks HP48**, disquettes compressées : Disk1 = G1 + G2 + G3, Disk2 = G5 + G6, Disk3 = G7 + G8.
- Les **utilitaires de développement** pour HP-48 fonctionnant sur IBM PC. La disquette contient les utilitaires développés mais non supportés par HP connus aussi sous le nom de **goodies 4**. C'est à dire un assembleur et un Linker pour le microprocesseur Saturn, un compilateur RPL et un générateur de librairies ainsi qu'une très abondante documentation sur le fonctionnement en détail de la HP48. La disquette contient aussi un compilateur écrit par Laurent Grand, permettant de mixer de l'*User Rpl* et de l'assembleur. Le dernier programme est un désassembleur/explorateur de la Rom de la HP48, écrit par Jean-François Garnier. Vous pouvez obtenir la disquette avec la documentation imprimée et reliée par nos soins.

Le port, et eventuellement la disquette (précisez le format), sont inclus dans les prix.

Si vous souhaitez des renseignements complémentaires, n'hésitez pas à nous contacter.

VOUS EN VOULEZ

Nom :
Prénom :
No d'adhérent :
Adresse :

	Quantité	Prix Unitaire Adhérent	Autre	Prix Total
Anciens numéros de JPC		x 40 FF	45 FF	
Année complète de JPC		x 300 FF	350 FF	
Programmathèque HP-71		x 60 FF	75 FF	
JPC Rom + Manuel		x 500 FF	600 FF	
JPC Rom + Manuel + vos propres programmes		x 700 FF	800 FF	
Actualisation Eprom		x 150 FF	200 FF	
Goodies Disks HP48 1, 2 et 3 compressés sur disque 1		x 40 FF	50 FF	
Goodies Disks HP48 5 et 6 compressés sur disque 2		x 40 FF	50 FF	
Goodies Disks HP48 7 et 8 compressés sur disque 3		x 40 FF	50 FF	
Kit de développement pour HP48 (disquette seule)		x 40 FF	50 FF	
Kit de développement pour HP48 (disquette + doc imprimée)		x 170 FF	200 FF	
			Total	FF

Préciser éventuellement les numéros de JPC commandés :

PPC PARIS

Association régie par la loi de 1901, enregistrée
à Paris le 2 décembre 1982 sous le numéro 82/3240

BULLETIN D'ADHESION

Nom |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|
Prénom |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_| Date de naissance |_|_|/|_|_|/|_|_|
Adresse |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|
_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_
_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_
_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_	_
Code Postal |_|_|_|_|_| Ville |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|
Téléphone Domicile |_|_|_|_|_|_|_|_|_|_|_|_|_| Bureau |_|_|_|_|_|_|_|_|_|_|_|_|

Profession ______________________________
Intérêts ______________________________
____
Matériel HP en votre possession ______________________________
____
Autre matériel informatique ______________________________
____

Comment avez-vous connu PPC Paris ? ______________________________
____

Que recherchez-vous au sein de PPC Paris ? ______________________________
____
____

La Loi No 78-17 du 6 janvier 1978 relative à l'informatique, aux fichiers et aux libertés, garantit à toute personne justifiant de son identité un droit d'accès et de rectification auprès des services ou organismes chargés de mettre en oeuvre des traitements informatiques comportant des informations nominatives le concernant.

Je souhaite adhérer au club PPC Paris conformément aux statuts de l'Association. Au mieux de ma connaissance, je déclare avoir le droit de fournir tous les programmes et informations que je vous enverrai (sans enfreindre des obligations de secret à l'égard d'autres personnes ou organismes) pour publication dans le Journal de liaison, sans obligations ni responsabilité d'aucune sorte (en cas d'utilisation frauduleuse) de la part des dirigeants de PPC-Paris.

Date |_|_|/|_|_|/19|_|_|
Signature, précédée de la mention "Lu et Approuvé" ____________________

Le montant de la cotisation s'élève à 350.00 F pour un an.
Etudiants: 300.00 F (justificatif indispensable)

Paiement à l'ordre de "PPC Paris" par chèque bancaire ou virement postal (CCP No 18 823 40 C à Paris). Ne pas utiliser d'Eurochèques.

Veuillez envoyer toute correspondance à :
PPC Paris, BP 604, 75028 Paris Cedex 01, France

trouvons p.140 l'adresse recherchée (ligne 6) : #62266h → DUP;=0(s)?

Ce Syseval nous retourne TRUE ou FALSE (indiqué comme un External au niveau 1 de la pile) et la ligne 7 doit sûrement effectuer un traitement booléen : elle l'effectue bien car l'External disparait de la pile, mais je suis incapable de vous dire quoi exactement (c'est la seule adresse que je n'ai pas trouvée : vous pouvez essayer de comprendre ce qu'elle fait en la désassemblant mais attention c'est du LM). De plus le sous-programme qui suit est surement effectué ou pas à la suite de ce traitement (qui dépend lui même de la présence ou non du RC).

Pour déterminer les autres adresses, je vais partir maintenant de la ligne 16 qui m'est connue : SUB(st,2s). Elle nécessite 2 entiers système donc je devrai trouver en ligne 15 un entier système : c'est le cas mais il faut chercher un peu p.142, ce qui nous donne <1h>;Swap.

De même ligne 14 je dois trouver un entier système ou un traitement s'y rapportant : on trouve p.139 que #03E0Eh (retournée) correspond à 1-(s). La ligne 13 est en sortie du sous-programme des lignes 8 à 12, cette ligne doit donc surement effectuer un traitement quelconque. Le premier Chapitre qu'il nous vient à l'esprit d'aller consulter est celui des *Opérations sur la Pile* : pas de miracle, on trouve que #031ACh correspond à DUP2. Ligne 11 c'est obligatoirement une opération sur un entier système puisque la ligne 14 lui soustrait 1, et on ne peut faire des opérations que sur des nombres de même nature. Gagné la ligne 11 donne 1+(s). Même si on n'avait pas pensé à çà, il était quasiment obligatoire qu'on l'eusse vu auparavant en recherchant la ligne 14 (même chapitre, même page, l'un est en dessous de l'autre et vice-versa).

En cherchant dans le chapitre *Chaîne de Caractères*, on ne peut pas avoir remarqué l'adresse de la ligne 10 : #05636h. Elle se situe p.174 et donne Ssize(st). De même quand nous avons regardé dans les Opérations sur la Pile, il nous semblait bien que les lignes 9 et 18 nous disaient quelque chose : elles signifient respectivement Drop;Dup et Rolld(s). La ligne 19 a été trouvée plus haut, il ne nous manque plus que les lignes 20 et 21. Or pour le moment nous n'avons fait qu'un SUB de la chaîne afin d'en extraire un premier morceau (ligne 16) ; il nous faudrait peut-être en faire un deuxième pour obtenir le 2 morceau. En cherchant p.175 on trouve facilement que la ligne 21 (#62D6Dh) correspond à Sub(st,2s);Swap. Enfin p.174 on trouve aussi facilement la ligne 20 (#05622h) qui donne Over;Ssize(st) car en cherchant la ligne 10 on l'avait déjà remarqué.

Ca y est, nous avons décodé toutes nos adresses, sauf 1, ce qui veut dire qu'on a compris ou presque comment fonctionnait la routine et donc qu'on allait facilement pouvoir la modifier, tout en en gardant la structure. Car il ne faut pas oublier que ces routines ont été programmées par les ingénieurs HP et qu'elles ont été réalisées avec le plus grand souci d'efficacité.

C'est là qu'il faut se rappeler ce que l'on veut : couper une chaîne en deux, à un endroit précis que l'on indiquera, sans perdre le caractère ou va s'effectuer la coupe, et sans retourner les chaînes résultantes.

En fait, dans le Syseval que nous avons décortiqué, une bonne partie ne nous intéresse pas, à savoir celle qui traite du Retour Chariot : du début du programme à la fin du sous-programme. On va donc shunter toute cette partie. Je vous rappelle que nous pouvons le faire car nous indiquons une position bien précise de coupe alors que dans le Syseval #127A7h, il faut déterminer ou se trouve le RC (et s'il s'y trouve vraiment).

De plus nous ne voulons pas perdre le caractère situé au niveau de la coupe, nous pouvons donc enlever l'instruction #03E0Eh (ligne 14) qui soustrait 1 à un entier système. Enfin nous ne voulons pas que les chaînes soient inversées dans la pile. Il nous faut donc remplacer l'adresse #62D6Dh (ligne 21) par un Sub(st,2s) à savoir tout simplement #05733h. Dernière remarque : notre position de coupe que l'on indique au niveau 1 de la pile est un réel ; il faut donc le convertir en entier système puisque tous les calculs se font avec ce type de nombre. On regarde une dernière fois dans le livre de J-M.F. p.177 chapitre *Conversions de Type*, et on emploiera l'adresse #18CEAh qui se charge de la transformation.

Finalement, on a le programme (beaucoup plus simple) :

```
_ D9D20
_ AEC81
_ CA130
_ 76E26
_ 33750
_ D0040
_ E9330
_ FED30
_ 22650
_ 33750
_ B2130
```

Comme précédemment il ne reste plus qu'à l'assembler et à essayer.

Voilà c'est fini. J'espère, à défaut de vous avoir appris quelque chose, vous avoir convaincu que les détournements (de routines 'sysevalesques' exclusivement) pouvaient vous apporter beaucoup !

Christophe Vaillant (523)

SYSTEM RPL FACILE & UNLINE

La ROM contient beaucoup de richesses qu'il est facile de récupérer pour étendre les commandes de la HP-48. Par exemple si la commande LINE est à la disposition de l'utilisateur il n'en est pas de même de UNLINE. Et pourtant tous les ingrédients, pour cette commande, existent bel et bien dans la ROM. Voici la méthode employée pour créer UNLINE :

Utiliser un décompilateur RPL, si possible donnant, les mnémoniques HP pour le RPL : par exemple MON48 de Jean François Garnier si vous avez un PC ou RPL48 Toolkit de Detlef Mueller & Raymond Hellstern, si vous n'avez qu'une HP-48.

Décompilation de la commande LINE avec MON48 :

on sait que cette commande User Rpl a la mnémonique xLINE pour la distinguer de LINE en System Rpl.

```
MON48
```

1ère étape

o =xLINE

```
1E398 02D9D Program
1E39D 18EDF CK2&Dispatch
1E3A2 64C66 2LIST
1E3A7 4F525 External (prog)
1E3AC 04143 THIRTYFOUR
1E3B1 4F584 External (prog)
1E3B6 0312B (End of Prog)
```

2ième étape: décompilation des programmes appelés par LINE (j'abrège les lignes de décompilation)

o =4F525

```
Program
TRUE
External (prog)
(End of Prog)
```

o =4F584

```
Program
TRUE
External (prog)
(End of Prog)
```

Avec un peu d'intuition on peut imaginer que si ces deux programmes, en utilisant le booléen TRUE, nous tracent une ligne, en remplaçant TRUE par FALSE ces mêmes programmes effaceront une ligne. Il suffira donc de refaire les deux programmes avec FALSE au lieu de TRUE.

Voici le fichier source de UNLINE à compiler avec RPLCOMP du kit de développement HP :

```
RPL
::
CK2&Dispatch
2LIST
 ::
  FALSE
ASSEMBLE
    con(5)  #4F54D
RPL
 ;
THIRTYFOUR
 ::
  FALSE
ASSEMBLE
    con(5)  #4F5AC
RPL
 ;
;
```

Avec une HP48 et RPL48 nous procéderions de la manière suivante:

la commande LINE se trouve à l'adresse 1E398 et la décompilation se fera en pas à pas avec DCADR :

#1E938h DCADR

```
::
CK2&Dispatch
2LIST
PTR 4F525
THIRTYFOUR
PTR 4F525
;
```

#4F525h DCADR

```
::
TRUE
PTR 4F54D
;
```

#4F584h DCADR

```
::
TRUE
PTR 4F5AC
;
```

En remplaçant TRUE par FALSE finalement on obtient le fichier source de UNLINE à compiler avec →RPL:

```
::
 CK2&Dispatch
 2LIST
 ::
  FALSE
  PTR 4F54D
 ;
 THIRTYFOUR
 ::
  FALSE
  PTR 4F5AC
 ;
;
```

UNLINE ayant évidemment la même syntaxe que LINE

Ceux qui n'ont pas de compilateur trouveront la chaîne du code à assembler dans le *Coin des codes*.

Conclusion :

Les décompilateurs usant les mnémoniques HP sont des outils apportant un confort très apprécié pour comprendre la programmation en System Rpl et en assembleur. Parfois, grâce à eux, de nouvelles commandes peuvent se réaliser sans effort comme ci-dessus. Avis aux amateurs !

Guy Toublanc (276)

NOUVEAUX POINTS D'ENTREE SYSTEM RPL

Depuis la sortie des utilitaires de développement HP, la liste des points d'entrées supportés s'est enrichie de quelques 200 adresses. J'en ai eu connaissance de façon indirecte, en fait par le système de développement RPL de Detlef Mueller (voir le volume 8 des Goodies Disks).

Le club disposera probablement rapidement de la dernière version des outils HP, cependant je pense que nombre d'entre vous sont interessés dès maintenant par l'étude de ces nouvelles adresses, aussi je les mets à disposition du club. J'ai donc transmis les fichiers nommées ENTR48.ADD et ENTR48.TYP classés respectivement par adresse et par type de points d'entrées. J'y ai inclus les nouveaux points d'entrées qui suivent cet article, ainsi que les adresses des objets user RPL. Ceux-ci sont en effet implicitement supportés, car on voit mal un programme utilisateur ne plus tourner sur les nouvelles versions de ROM. De plus j'ai indiqué en commentaire le type d'objet défini (prog, primitive, assembly, user RPL, ...).

Cette nouvelle liste est utilisable par les outils HP, ainsi que (pour ceux d'entre vous qui se l'ont procuré au club) par mon utilitaire d'exploration MON48 tournant sur PC.

Nouveaux points d'entrées :

NDLR : Jean-François a établi cette liste à partir d'une liste de points d'entrées antérieure aux Goodies #4. Certains adhérents possédant la première version, nous la passons en intégralité, mais après avoir noté d'un 'N' les points réellements nouveaux. A vous de faire le tri suivant la liste que vous possédez déjà.

```
=portnotaverr EQU #0000A =  (assembly)   N
=CRC          EQU #00104 =  (assembly)     N
=RCS          EQU #00111 =  (assembly)     N
=CRER         EQU #00113 =  (assembly)     N
=RBR          EQU #00114 =  (assembly)     N
=LCR          EQU #0011C =  (assembly)     N
=TIMER1       EQU #00137 =  (assembly)
=SavPtrTime*  EQU #01307 *  (assembly)     N
=OffNoBlush   EQU #01D44 *  (assembly)     N
=DOSYMB       EQU #02AB8 =  (assembly)     N
=GETTEMP      EQU #039BE *  (assembly)     N
=ECUSER       EQU #039EF *  (assembly)     N
=EXITMSGSTO   EQU #04E37 *  (primitive)    N
=ATTNFLG@     EQU #05040 *  (primitive)
=>HCOMP       EQU #052C6 *  (prog)         N
=SYMBN        EQU #0546D *  (prog)
```

```
=MOVERSD       EQU #06992 *  (assembly)     N
=MOVEDSU       EQU #069C5 *  (assembly)     N
=MOVEDSD       EQU #06A1D *  (assembly)     N
=MOVERSU       EQU #06A53 *  (assembly)     N
=DOBIND        EQU #074E4 *  (primitive)    N
=BAKNAME       EQU #081D9 *  (primitive)    N
=BAK>OB        EQU #0948E *  (primitive)    N
=SAFESKIPOB    EQU #0A532 *  (assembly)     N
=NEXTLIBBAK    EQU #0AB82 *  (primitive)    N
=Date>d$       EQU #0CFD9 *  (primitive)
=TOD>t$        EQU #0D06A *  (primitive)
=mpop1%        EQU #0D8AE *  (assembly)     N
=CKTIME        EQU #0D9C7 *  (assembly)     N
=U>nbr         EQU #10047 *  (prog)
=GPErrjmpC     EQU #10F40 *  (assembly)     N
=SYNTAXERR     EQU #10F86 *  (primitive)    N
=PrgmEntry?    EQU #11511 *  (primitive)    N
=SetPrgmEntry  EQU #11533 *  (primitive)    N
=makegrob      EQU #115B3 *  (assembly)
=DISPROW8      EQU #124CB *  (primitive)    N
=WINDOWUP      EQU #131C8 *  (primitive)    N
=WINDOWDOWN    EQU #13220 *  (primitive)    N
=WINDOWLEFT    EQU #134E4 *  (primitive)    N
=WINDOWRIGHT   EQU #1357F *  (primitive)    N
=WINDOWXY      EQU #13679 *  (primitive)    N
=ALGeq?        EQU #1568F *  (prog)
=DOSTOE        EQU #15717 *  (prog)
=EQUATION      EQU #15744 *  (prog)
=EVALCRUNCH    EQU #1583C *  (prog)
=CRUNCHNoBlam  EQU #15941 *  (prog)
=1stkdecomp$w  EQU #15978 *  (prog)
=stkdecomp$w   EQU #159EB *  (primitive)
=ederr         EQU #15A40 *  (prog)
=rstfmt1       EQU #15A60 *  (primitive)
=savefmt1      EQU #15A8B *  (primitive)
=NDROPFALSE    EQU #169A5 *  (prog)
=!DcompWidth   EQU #1795A *  (primitive)
=VLM           EQU #17B86 *  (prog)         N
=?PURGE_HERE   EQU #1854F *  (prog)
=OLASTOWDOB!   EQU #1884D *  (primitive)
=OCRC%         EQU #1A1FC *  (prog)         N
=xFCNAPPLY     EQU #1F640 *  (prog)
=%9600         EQU #22391 *  (real)         N
=xIFEND        EQU #22FD5 *  (prog)         N
=xALG->        EQU #22FEB *  (prog)         N
=xSILENT'      EQU #2349C *  (prog)         N
=xRPN->        EQU #234C1 *  (prog)         N
=x>>ABND       EQU #235FE *  (prog)         N
=x>>           EQU #23639 *  (prog)         N
=xENDTIC       EQU #23679 *  (prog)         N
=xWHILEEND     EQU #23694 *  (prog)         N
=xENDDO        EQU #236B9 *  (prog)         N
=xERRTHEN      EQU #2371F *  (prog)         N
=xTHENCASE     EQU #237A8 *  (prog)         N
=tok->         EQU #25446 *  (string)
=?OKINALG      EQU #26A2D *  (prog)
=%9            EQU #2A371 *  (real)         N
=STATCLST      EQU #2C22F *  (prog)         N
=STATSADD%     EQU #2C2D9 *  (prog)         N
=STATN         EQU #2C535 *  (prog)         N
=STATSMAX      EQU #2C558 *  (prog)         N
=STATMEAN      EQU #2C571 *  (prog)         N
=STATSMIN      EQU #2C58A *  (prog)         N
=STATSTDEV     EQU #2C5A3 *  (prog)         N
=STATTOT       EQU #2C5BC *  (prog)         N
=STATVAR       EQU #2C5D5 *  (prog)         N
=LAMPACKET     EQU #2D3B1 *  (locname)      N
=LAMKP         EQU #2D3EE *  (locname)      N
=LAMOBJ        EQU #2D40E *  (locname)      N
=LAMOPOS       EQU #2D41D *  (locname)      N
=LAMKLIST      EQU #2D45A *  (locname)      N
=EXCHINITPK    EQU #2D517 *  (prog)         N
=SetServMode   EQU #2D9A1 *  (primitive)
=DropSysErr$   EQU #2DDC4 *  (prog)         N
=APNDCRLF      EQU #2E4DC *  (prog)         N
=GetStrLenC    EQU #2FFB7 *  (assembly)     N
=GETKP         EQU #307E2 *  (prog)         N
=CLOSEUART     EQU #315C6 *  (prog)
=CloseUart     EQU #315F9 *  (assembly)     N
=DOCR          EQU #31868 *  (prog)         N
=SetEcma94     EQU #3252B *  (primitive)
=InitSysUI     EQU #385E8 *  (prog)         N
=SysErrorTrap  EQU #38728 *  (prog)         N
=AppDisplay!   EQU #38C08 *  (primitive)
=AppKeys!      EQU #38C38 *  (primitive)
=AppExitCond!  EQU #38C68 *  (primitive)
=AppError!     EQU #38C98 *  (primitive)
=AppMode?      EQU #38CFB *  (primitive)    N
=SetAppMode    EQU #38D09 *  (primitive)
=SetNAppKeyOK  EQU #38D33 *  (primitive)
=SetDoStdKeys  EQU #38D5D *  (primitive)
=ClrAppSuspOK  EQU #38D9B *  (primitive)
=DA1OK?        EQU #38DAC *  (prog)
=DA1OK?NOTIT   EQU #38F28 *  (prog)
=DA2aOK?NOTIT  EQU #38F41 *  (prog)
=DA2bOK?NOTIT  EQU #38F5A *  (prog)
=DA3OK?NOTIT   EQU #38F73 *  (prog)
=SetDA2aValid  EQU #38FEB *  (prog)
=MENP&FixDA12  EQU #390B3 *  (prog)         N
=ClrDA1OK      EQU #390CC *  (prog)
=ClrDA2OK      EQU #39117 *  (prog)         N
=SetDA2Valid   EQU #3915D *  (prog)         N
=SetDA12NoCh   EQU #3919E *  (prog)         N
=SetDA123NoCh  EQU #391EE *  (prog)         N
=SetDA2OKTemp  EQU #39207 *  (prog)         N
=SetDA1Bad     EQU #3947B *  (primitive)    N
=DA2aBad?      EQU #39497 *  (primitive)    N
=SetDA2aBad    EQU #394A5 *  (primitive)    N
=ClrDA2aBad    EQU #394B3 *  (primitive)    N
=SetDA1IsStat  EQU #39523 *  (primitive)    N
=SetNoRollDA2  EQU #3957A *  (primitive)    N
=?DispStatus   EQU #3959C *  (prog)         N
=DispStatus    EQU #395BA *  (prog)         N
=Blank&GROB!   EQU #39632 *  (prog)         N
=AngleStatus   EQU #39673 *  (prog)         N
=ComVecStatus  EQU #396C8 *  (prog)         N
```

```
=UserFlagStat EQU #39748 *  (prog)        N
=UserKeysStat EQU #397BB *  (prog)        N
=AlgEntryStat EQU #3981B *  (prog)        N
=PrgmEntrStat EQU #39853 *  (prog)        N
=DispStsBound EQU #39B0A *  (prog)        N
=?DispStack   EQU #39B85 *  (prog)        N
=DispEditLine EQU #3A00D *  (prog)
=?DispMenu    EQU #3A1CA *  (prog)
=BlankDA1     EQU #3A546 *  (prog)
=BlankDA2a    EQU #3A591 *  (prog)
=TurnOffKey   EQU #3A9CE *  (prog)
=1A/LockA     EQU #3AA0A *  (prog)
=DoPlotMenu   EQU #3ADED *  (prog)        N
=EditMenu     EQU #3BDFA *  (list)
=IStackKey    EQU #3E5CD *  (list)
=NoExitAction EQU #3EC85 *  (prog)
=Box/StdLbl:  EQU #3ECB2 *  (prog)
=Key>U/SKeyOb EQU #3FA57 *  (prog)        N
=2DropBadKey  EQU #3FDBD *  (prog)        N
=DropBadKey   EQU #3FDC7 *  (prog)        N
=KeyOb!       EQU #40A6F *  (primitive)   N
=KeyOb@       EQU #40A82 *  (primitive)   N
=Parse.1      EQU #40AD9 *  (prog)
=ParseFail    EQU #40B2E *  (prog)
=StartMenu    EQU #41008 *  (prog)
=SaveLastMenu EQU #4139B *  (prog)        N
=InitTrack:   EQU #41741 *  (prog)        N
=SetRebuild   EQU #417F3 *  (primitive)   N
=MenuRow!     EQU #41848 *  (primitive)   N
=LastMenuRow! EQU #4186E *  (primitive)   N
=LastMenuRow@ EQU #41881 *  (primitive)   N
=LastMenuDef! EQU #419E4 *  (primitive)   N
=LastMenuDef@ EQU #419F4 *  (primitive)   N
=GetUserKeys  EQU #41F3F *  (primitive)   N
=GetKeyOb     EQU #4203C *  (prog)        N
=EchoChrKey   EQU #42AE4 *  (prog)
=InputLAttn   EQU #43200 *  (prog)
=SetCursor    EQU #4325A *  (prog)
=InitEd&Modes EQU #44277 *  (prog)
=InitEdLine   EQU #4428B *  (prog)
=InitEdModes  EQU #44394 *  (prog)
=EditString   EQU #443CB *  (prog)
=EDITLINE$    EQU #44683 *  (primitive)
=EDITPARTS    EQU #44730 *  (primitive)
=?TogU/LCase  EQU #448C1 *  (prog)
=DoNewMatrix  EQU #44C31 *  (prog)        N
=DoOldMatrix  EQU #44FE7 *  (prog)        N
='IDX         EQU #4744F *  (prog)
=SORTASLOW    EQU #48FF9 *  (prog)
=AUTOSCALE    EQU #491D5 *  (prog)
=PointDerivUt EQU #49AD3 *  (prog)
=FcnUtilEnd   EQU #49BA5 *  (prog)
=RootUtil     EQU #49BD2 *  (prog)
=CkEQUtil     EQU #49C54 *  (prog)
=PointMoveCur EQU #49F06 *  (prog)
=DISPCOORD2   EQU #4A055 *  (prog)
=GetEqN       EQU #4A0AA *  (prog)
=ICMPDRPRTDRP EQU #4A95A *  (prog)
=CHECKPVARS   EQU #4A9AF *  (prog)
=MAKEPVARS    EQU #4AAEA *  (prog)
=GETSCALE     EQU #4ADB0 *  (prog)        N
=PUTSCALE     EQU #4AE3C *  (prog)        N
=GETINDEP     EQU #4AF63 *  (prog)        N
=PUTINDEP     EQU #4AF77 *  (prog)
=PUTINDEPLIST EQU #4AF8B *  (prog)
=GETRES       EQU #4AFDB *  (prog)
=PUTRES       EQU #4B012 *  (prog)
=GETPTYPE     EQU #4B062 *  (prog)
=PUTPTYPE     EQU #4B076 *  (prog)
=GETPMIN&MAX  EQU #4B0DA *  (prog)
=GETXMIN      EQU #4B10C *  (prog)
=GETYMIN      EQU #4B120 *  (prog)
=GETXMAX      EQU #4B139 *  (prog)
=GETYMAX      EQU #4B14D *  (prog)
=PUTXMIN      EQU #4B166 *  (prog)
=PUTYMIN      EQU #4B189 *  (prog)
=PUTXMAX      EQU #4B1AC *  (prog)
=PUTYMAX      EQU #4B1CF *  (prog)
=MAKEPICT#    EQU #4B323 *  (prog)
=GETPARAM     EQU #4B364 *  (prog)        N
=VSCALE       EQU #4B553 *  (prog)
=HSCALE       EQU #4B5AD *  (prog)
=PLOTERR      EQU #4B6D9 *  (prog)
=RESETDEPTH   EQU #4B710 *  (prog)
=PLOTPREP     EQU #4B765 *  (prog)
=GetRes       EQU #4B7D8 *  (prog)
=NEXTSTEP     EQU #4BFAE *  (prog)
=NEWINDEP     EQU #4C09B *  (primitive)
=drax         EQU #4C639 *  (prog)
=EXITFCNsto   EQU #4CE4C *  (prog)
=GraphicExit  EQU #4CEE7 *  (prog)
=GDISPCENTER  EQU #4CF05 *  (prog)
=SETLOOPENV   EQU #4CF41 *  (prog)
=ExitFcn      EQU #4CF68 *  (locname)
=DROPDEADTRUE EQU #4D11E *  (prog)
=CROSS_HAIRS  EQU #4DA0D *  (prog)
=CROSS_OFF    EQU #4DA76 *  (prog)
=CHECKMENU    EQU #4E266 *  (prog)
=MENUOFF      EQU #4E2AC *  (prog)
=MENUOFF?     EQU #4E360 *  (prog)
=CURRENTMARK? EQU #4E442 *  (prog)
=EQCURSOR?    EQU #4E46A *  (prog)
=PREMARKON    EQU #4E497 *  (prog)
=NEWMARK      EQU #4E4B0 *  (prog)
=DispCoord1   EQU #4E6EF *  (prog)
=Z-BOX        EQU #4E776 *  (prog)
=CROSSMARKON  EQU #4ECAD *  (prog)
=C%>#         EQU #4F408 *  (prog)        N
=GETXPOS      EQU #505C6 *  (prog)
=getxpos      EQU #505E4 *  (primitive)
=GETYPOS      EQU #5068D *  (prog)
=getypos      EQU #506AB *  (primitive)
=TOGGLELINE#3 EQU #5072B *  (prog)
=DRAWLINE#3   EQU #50758 *  (prog)
=LASTPT?      EQU #50D78 *  (prog)
=PlotOneMore? EQU #50DA5 *  (prog)
```

```
=!#1+IF<dim-1 EQU #50E59 *  (primitive)
=!#1-IF>0     EQU #50EA5 *  (primitive)
=INDEPVAR     EQU #510AD *  (prog)
=RECORDX&YC%  EQU #510D5 *  (prog)
=CLEARMENU    EQU #51125 *  (prog)
='IDFUNCTION  EQU #5129C *  (prog)
='IDPOLAR     EQU #512C4 *  (prog)
='IDPARAMETER EQU #512D8 *  (prog)
=PtoR         EQU #5133C *  (prog)
=GETRHS       EQU #514AF *  (prog)
=1REV         EQU #514DC *  (prog)
=TOPROW       EQU #515A0 *  (prog)
=BOTROW       EQU #515B4 *  (prog)
=LEFTCOL      EQU #515FA *  (prog)
=RIGHTCOL     EQU #5160E *  (prog)
=WINDOWTOP?   EQU #5162C *  (prog)
=WINDOWBOT?   EQU #51645 *  (prog)
=WINDOWLEFT?  EQU #5165E *  (prog)
=WINDOWRIGHT? EQU #51677 *  (prog)
=ISTOP-INDEX  EQU #5182F *  (prog)
=4NULLLAM{}   EQU #52D26 *  (list)
=HISTON       EQU #5386E *  (primitive)    N
=UNDO_ON?     EQU #538C0 *  (primitive)    N
=UNDO_ON      EQU #538CE *  (primitive)    N
=UNDO_OFF     EQU #538DC *  (primitive)    N
=AlgEntry?    EQU #53968 *  (primitive)    N
=SetAlgEntry  EQU #53976 *  (primitive)
=EditLExists? EQU #53A4A *  (primitive)    N
=RAD?         EQU #53BDD *  (prog)
=SYMBWHERE    EQU #547B5 *  (prog)
=1GETLAMSWP1+ EQU #55288 *  (prog)         N
=xssgeneral   EQU #560ED *  (prog)         N
=xnsgeneral   EQU #56101 *  (prog)         N
=xsngeneral   EQU #5611F *  (prog)         N
=SYMSHOW      EQU #58D75 *  (prog)
=RDROPFALSE   EQU #5DE55 *  (prog)         N
=psh1top&     EQU #5E401 *  (prog)         N
=ATTNERR      EQU #64FC2 *  (systbin)
=tok<<        EQU #651D6 *  (string)
=tok'         EQU #65284 *  (string)
=UserFlags    EQU #706D5 =  (assembly)     N
```

Jean-François GARNIER (242)

MISE EN MAJUSCULES

Voici un petit programme qui convertit une chaîne de caractères en majuscules. J'ai écrit ce programme lors de la création d'un nouvel agenda téléphonique, ce qui m'a permis d'ignorer les majuscules / minuscules lors de la recherche de noms.

Le corps du programme est en assembleur. Cela permet de gagner un facteur d'environ soixante par rapport à un programme écrit en RPL ou en system RPL.

Je ne pense pas que ce programme nécessite d'autres commentaires. A vous d'en profiter.

```
 ; Programme TOUPPER
 ; Utilisation : 1:$ --> 1:$
 ; ===========
RPL
::
CK1NOLASTWD
CK&DISPATCH0
str
* Test des arguments
::
TOTEMPOB
ASSEMBLE
    include  addr-48.as
    nibhex   begin_code
beg rel(5)   end - beg
    gosbvl   save_reg
    a=dat1   a
    d0=a
    d0=d0+5
    a=dat0   a
    a=a-con  a,5
    asrb.f   a          a = long (chaine)
    d0=d0+3
    lchex    #61
    b=c      b
    lchex    #7A
    d=c      b
lp  ?a=0     a          boucle principale
    goyes    fin
    a=a-1    a
    d0=d0+2
    c=dat0   b
    ?c<b     b
    goyes    lp
    ?c>d     b
    goyes    lp
    c=c-con  b,16
    c=c-con  b,16
    dat0=c   b
    goto     lp
fin gosbvl   load_reg
    a=dat0   a
    d0=d0+5
    pc=(a)
end ENDCODE
;
;
```

Vous trouverez le listing hexadécimal de ce programme dans le *Coin des Codes*, en fin du journal.

Laurent Grand (516)

HP95 / HP100

LE NOUVEAU HP100LX

Deux ans après la sortie du HP95, Hewlett-Packard nous offre enfin un successeur à cette machine. Cet article a pour but d'en montrer les principales évolutions.

PRESENTATION PHYSIQUE

La nouvelle machine bénéficie du même boitier que le HP95LX. De loin, la seule différence visible est la couleur des touches, qui sont devenues blanches, à l'exception du pavé numérique.

Hormis le '100LX', les marquages entourant l'écran ont légèrement changé. Tout d'abord, le sigle '123' de Lotus est maintenant complèté par la mention cc:Mail. Mais le plus important ajout est sûrement la mention 'Palmtop PC', située à coté de '1MB RAM'. Ceci ayant peut-être pour but de se démarquer ostensiblement de tous les *Organizers* lui ressemblant un peut trop...

L'écran est très légèrement plus large (environ 8 mm). Mais ce qui le caractérise le plus est sa nouvelle résolution : 640×200 (au lieu de 240×128 pour le 95). Il est à la norme CGA (avec apparament 8 ou 16 niveaux de gris). Cette hausse de la résolution, alors que l'écran ne change pas de taille, peut rendre l'affichage difficilement lisible pour certains. Ceci est corrigé par une faculté de zoom à trois niveaux, affichant cycliquement les caractères en 80, 64 et 40 colonnes.

Le clavier, comme je l'ai dit plus haut, a changé de couleur. Cependant, d'autres modifications ont été effectuées. La plus importante concerne le bloc des touches d'applications (toujours bleues), qui ont été décalées d'un cran vers la droite, afin de laisser la place pour la touche TAB. La touche bleue DATACOM est remplacée par un nouveau logo : une enveloppe. Ceci permet l'accès direct à cc:Mail (voir plus bas). Une nouvelle touche bleue a été ajoutée : More, représentée par "&...". Elle permet d'accéder à un nouvel écran, que nous verrons plus tard. Ces changement provoquent la disparition des touches '(' et ')'. Parmi les autres caractères qui ont changé, Les autres touches principales qui ont changé sont :

TAB qui devient '\'
CHAR qui devient 'FN' (accès aux fonctions spéciales)
'@' qui devient '.'

D'autres caractères *shiftés* ont été déplacés, ainsi que l'accès à certaines fonctions spéciales. Cependant la machine que j'ai eu entre les mains étant un modèle de pré-série (avec un clavier US), il faudra attendre les modèles définitifs pour connaitre les différences exactes et définitives.

Le port PCMCIA est toujours présent. Il accepte maintenant les cartes à la version 2 de la norme. Ceci permettra d'accueillir de nouveaux types de périphériques (voir la fin de l'article pour une pré-liste). Les cartes RAM standard peuvent atteindre maintenant la capacité de 32 Mo.

Le port série bénéficie à présent d'un connecteur spécifique à 10 broches, gérant les signaux standard des connecteurs 9 broches des PC. Ceci permettra d'y connecter directement des périphériques tels que des modems ou une souris !

L'interface infrarouge permet maintenant de supporter un débit allant jusqu'à 115 kbauds, soit la même vitesse que le port série.

En ce qui concerne l'architecture interne, elle a considérablement évolué, puisque les deux circuits principaux (microprocesseur Nec V20 et controleur d'entrées sortie Intel/HP) ont été remplacés par un circuit unique. Le microprocesseur est maintenant un 80186, cadencé à 7.91 Mhz. La mémoire vive reste limitée à 1 Mo, mais la Rom est portée à 2 Mo.

Une autre modification concerne l'alimentation électrique. Le nouveau modèle accepte directement les accus Cadmium/Nickel, qui peuvent être rechargés directement dans la machine, grâce à l'adaptateur secteur. Le HP100 optimisant le niveau de charge afin d'accroitre la durée de vie des batteries.

PRESENTATION LOGICIELLE

Si vous avez été trop impressionnés par la première partie de cet article, je vous conseille de passer tout de suite à l'article suivant...

En effet, les évolutions du nouveau matériel ne concernent ne sont que des modifications mineures par rapport à l'évolution du logiciel, puisque une très grande partie de la Rom a été réécrite.

Le changement le plus flagrant concerne l'interface utilisateur du *System Manager*, puisqu'il offre maintenant une interface graphique avec menus déroulants et boites de dialogues, d'un niveau égal aux meilleurs logiciels tournant sur PC.

Toutes les applications internes bénéficient de cette nouvelle interface, à l'exception de Lotus qui garde l'ancienne configuration de menus.

De plus, certaines applications acceptent des raccourcis clavier, évitant de s'aventurer trop souvent dans les nombreux menus.

Le nombre d'applications *System Manager* étant devenu trop important, la touche 'More' (représentée par '&...') permet d'accéder à un *tableau de bord* (à ne pas confondre avec le "TopCard") où toutes les applications sont représentées par des icônes que l'on peut sélectionner (en se positionnant dessus ou en tapant une lettre repère) pour lancer le programme désiré. Il est possible d'installer de nouvelles applications System Manager, avec leurs icônes, dans ce tableau de bord. Cette methode sera donc plus simple que l'ancienne manière d'installation par l'intérmédiaire du fichier `APNAME.LST` et ses codes de touches dont personne n'a jamais réussi à se rappeler par coeur....

Malgré ces changements, il est bien entendu toujours possible d'utiliser les applications System Manager du HP95.

Hormis l'évolution de l'interface utilisateur, la plupart des applications existantes ont vu leur possibilités étendues. N'ayant pas eu le temps d'en faire le tour complet (il m'aurait fallu plus d'une journée pour cela !), je ne ferai qu'en donner un aperçu, ne donnant que les points les plus importants ou certaines "petites choses" qui améliorent l'utilisation par rapport au HP95.

Amélioration des applications existantes

Setup

Les principaux changements concernent certaines options de communication, ainsi que la possibilité de paramètrer certaines valeurs d'utilisation de la mémoire.

Il est à noter qu'il existe maintenant une possibilité d'imposer (au moment du boot) sur quelle unité (disque interne ou carte) doit se faire la réinitialisation.

Filer

La principale amélioration de cette application est sûrement qu'il est maintenant possible de transférer les fichier par le port infrarouge à la vitesse de 115 kBauds. Cela fera moins de moments d'attente pendant les réunions !

L'autre modification la plus notable est peut-être le fait que la touche `ENTER` n'affiche plus le contenu d'un fichier (comme sur le HP95), mais remplace la touche F4, en lancant l'exécution d'un programme, si il s'agit d'un fichier EXE, COM ou BAT. Personnelement, je n'avais jamais pu m'habituer à la touche F4...

Communications

Cette application n'est plus accessible directement, puisque la touche qui lui était allouée sur le HP95 est remplacée par l'accès à cc:Mail.

Cependant elle existe toujours, et sa principale amélioration est qu'elle supporte maintenant les protocoles Ymodem et Zmodem.

Agenda

Le changement le plus visible est que le calendrier du mois en cours est visible sur l'écran principal, en même temps que le planning de la journée pointée sur ce calendrier.

Répertoire téléphonique

De très nombreux champs ont été ajoutés : 2eme numéro de téléphone, indicateur du type de contact (personnel, professionnel...), un champ de format libre de deux lignes permettant de mettre des notes...

Editeur de texte

Considéré par certains comme étant la plus mauvaise application du HP95, l'éditeur a été totalement réécrit. Il se situe maintenant à la limite des traitements de textes, puisque il permet de gérer les caractères gras ou soulignés.

Une autre des grandes amélioration est la possibilité d'ouvrir plusieurs fichiers à la fois.

Lotus 1-2-3

Il s'agit maintenant de la dernière version, numérotée 2.4 (contre 2.1 sur le HP95), avec la totalité de ses possibilités graphiques et des macros prédéfinies.

Calculatrice

Les possibilités de lien avec Lotus ont été étendues.

Il est maintenant possible d'ajouter ses propres "mini-applications" dans les menus. Les figures sont affichées automatiquements avec le maximum de résolution.

Nouvelles applications

En plus des applications du HP95, la nouvelle machine comprend de nouvelles apllications. Elles sont principalement accessibles après avoir appuyé sur la touche 'More' (`&...`).

Notons que certaines applications, telles que le chronomètre ont été "sorties" de leur application d'origine et bénéficient de leur propre icône.

MS-DOS

A présent, le HP100LX utilise la version 5.0 de MS-DOS. Si toutes les commandes ne sont pas présentes, nous avons au moins la possibilité de récupérer sans modifications les commandes de la version PC, ce qui n'était pas possible avec le HP95.

Sur le HP95, l'accès à la ligne de commande de MS-DOS était possible uniquement en appelant le programme `COMMAND.COM`, à partir du Filer. Sur le HP100, par contre, le Dos possède sa propre icône. Il n'est plus obligatoire de fermer toutes les applications pour accéder à la ligne de commande. Cependant, la taille des programmes accessibles de cette façon est limitée par une valeur que l'on doit spécifier dans le Setup (par défaut 96 Ko). Autrement, il faudra procéder comme pour le HP95.

Base de données

Il s'agit d'une base de données de type mono-fichier. Il est bien sûr possible de créer ses propres écrans de saisie.

Le 'Note Taker'

Il s'agit d'un 'carnet de notes' permettant de créer des fiches et leur donner un titre. Un écran permet de récapituler et manipuler ces fiches en fonction de ce titre.

cc:Mail

cc:Mail est un logiciel de gestion de messagerie, développé par Lotus permettant d'envoyer ou recevoir des messages lorsque le HP100 est connecté à un modem ou à un autre ordinateur.

Produits associés

Un kit de connection spécifique au HP100 est bien sûr prévu. Il n'était pas encore disponible au moment du bouclage de ce numéro, ce qui implique que je n'en connait actuellement que le prix : 731 Francs HT (Prix Catalogue HP). Le cable seul est disponible 149 F HT.

L'evolution de version du slot PCMCIA, permet dès à présent d'y connecter de nouveaux types de périphériques : Controleurs SCSI, disques durs, la plupart des modems PCMCIA. On parle même d'une interface vidéo ! HP distribue dès à présent deux cartes Flash (équivalentes aux Sundisk), de 5 et 10 Mo aux prix respectifs de 3990 et 6990 Francs HT. Ces cartes sont allimentés en 12 V, et sont incompatibles avec le HP95 (cependant une version de 5 Mo est disponible pour cette machine, au prix de 4180 F)

Un logiciel PIM, regroupant certaines apllications du HP100 (Agenda, Base de données) est disponible pour les PC, sous Windows. Prix 422 F HT.

Enfin, Hewlett-Packard a annoncé en même temps que le HP100, une nouvelle série d'ordinateurs de bureau. Certains modèles sont équipés de lecteurs PCMCIA, et même d'intefaces infrarouge ! Il ne sera donc plus nécesaire d'effectuer de fastidieuses manipulations de cables pour échanger des fichiers avec les PC !

En conclusion, plus qu'une simple évolution, il s'agit vraiment d'une nouvelle machine. Le HP100LX a corrigé tous les défauts qui étaient donnés à son prédécesseur, et placera (encore une fois) Hewlett-Packard à l'abri de toute concurrence. Maintenant, non seulement il est un PC à part entière, mais grâce à ses application internes et à la puissance du System Manager, il est *plus* qu'un PC !

Il sera disponible début juin chez certains distributeurs, en version US. La version Française devrait apparaitre en septembre.

Mais avant que vous ne vous jetiez sur votre pauvre revendeur, il ne me reste plus qu'un renseignement à vous donner : le prix ! Le tarif officiel de HP étant de 6000 F HT, il est fort possible que vous le retrouviez, à ce prix là, mais en TTC. En regard du niveau de technologie de la machine et de la puissance des aplications intégrées, le prix est loin d'être excessif. Rappelez vous aussi que n'importe quel portable auquel on rajoute les aplication existantes dans la Rom (Lotus, Traitement de texte, Base de données, Laplink...) revient facilement deux ou trois fois plus cher, et que vous ne le transporterez jamais aussi facilement que le HP100 !

Jacques Belin (123)

JEUX DE CARTES

Divers lecteurs de cartes PCMCIA sont disponibles aux Etats-Unis, qui permettent de lire et écrire sur des cartes RAM à partir d'un PC; la plupart d'entre eux se branchent tout simplement sur le port d'imprimante, et donc ne nécessitent pas de carte interface spécialisée, ce qui est bien pratique avec les portables. L'article qui suit présente le TMD-550, de chez DataBook, qui offre deux "plus" par rapport aux autres, l'accès aux cartes SunDisk, "stackées" ou non, en lecture et en écriture, et la possibilité de "brûler" des cartes Flash.

Présentation, Prix et Disponibilité :

Le lecteur prend la forme d'un boîtier beige de dimensions réduites (14x18x3 cm), accompagné d'un bloc secteur (9 à 15 Volts CC, positif au centre) et d'un cordon DB-25 mâle-mâle blindé. Il comporte deux connecteurs DB 25, et permet donc l'utilisation simultanée d'une imprimante parallèle par déport du port d'imprimante du micro, auquel il est relié.

Il est livré avec un manuel très détaillé et une disquette d'installation des "drivers" et utilitaires. Ceux-ci lui permettent d'utiliser, outre les cartes de mémoire RAM classiques, des cartes "Flash" (type particulier d'EEPROMs) de divers formats. Attention à la version du logiciel, qui peut ne pas reconnaître correctement les cartes SunDisk; ce défaut est corrigé dans la dernière version, dénommée V2.15 et disponible chez le constructeur sur simple demande jointe à la carte d'enregistrement du produit.

Le lecteur est disponible comme d'habitude chez EduCALC, sous deux références : soit isolément (#TMD550, prix avoisinant $260 US), soit en offre promotionnelle accompagnant une carte SunDisk (#TMD550ZZ, prix baissant alors de moitié). Il existe aussi une version interne, se montant à la place d'un lecteur de disquettes, se connectant à une carte interface spécifique courte par une nappe, et avec des possibilités supplémentaires le destinant plutôt aux développeurs (notamment brûlage de cartes EPROM-UV). Un Manuel très complet et clair, commun aux 3 versions, accompagne le tout.

Particularités d'installation :

L'installation est très sûre et bien documentée pour l'ancienne version du logiciel, mais le Manuel n'a pas encore été mis à jour pour la V2.15, qui doit être installée un peu "au feeling". Cet appel au flair est encore plus important lorsqu'on doit installer STACKER et, là, c'est la doc de SunDisk qui se révèle légère... Voici donc quelques compléments indispensables, basés sur notre expérience de produits similaires (SuperStor, MS-DOS 6, etc.) et nos découvertes et déboires.

L'installation se déroule en deux étapes, à savoir (1) celle des "drivers" du lecteur et (2) celle de STACKER s'il y a lieu (cas des cartes SunDisk et ACE Double Card). Compte tenu des dernières évolutions en matière de MS-DOS et de diverses bizarreries relevées dans les fichiers batch d'installation, il est recommandé de faire cette installation à la main.

Le principe consiste, après avoir copié plusieurs fichiers dans un répertoire tel que C:\STACKER ou C:\DOS (le nom n'a guère d'importance, dès lors qu'il est dans le PATH), à créer dans `CONFIG.SYS` deux lignes `DEVICE`, dans cet ordre :

```
DEVICE=<unité:><chemin>TMD550.SYS /LPTn
DEVICE=<unité:><chemin>STACKER.COM /p:1 D: u:
```

En ce qui concerne le premier `DEVICE`, pas de problème notable; le "switch" final désigne le port parallèle où est branché le lecteur (en général, `LPT1:`). Pour le second, c'est un peu plus compliqué, et c'est là qu'intervient le "feeling".

Une carte SunDisk compactée apparaît, sur un PC (ou un 95!) sur lequel STACKER n'est pas activé, comme un disque (A: sur le 95, D: sur un PC -puisque la carte se place APRES toute autre unité de mémoire de masse de type disque dur-) qui contient un très gros fichier `STACVOL.DSK` (dont les deux attributs Système et Caché sont armés), et plusieurs autres petits, parmi lesquels `STACKER.DRV` et `STACKER.EXE` (que l'on retrouve à l'identique sur la disquette accompagnant la carte SunDisk). Passons sur leur rôle pour le moment, et notons simplement que l'on copiera `STACKER.DRV` dans <chemin> du disque dur, en en changeant l'extension de `DRV` en `COM`.

Notons aussi qu'il faut indiquer *deux* noms d'unités dans la ligne de lancement de `STACKER.COM` : l'unité "réelle" en premier (ici, D:), puis l'unité virtuelle (E: si vous n'avez pas déjà activé STACKER pour votre disque dur, G: si votre disque dur a lui-même été compacté).

Dès que les deux lignes `DEVICE` auront été créées, et les fichiers correspondants recopiés dans <chemin>, on devra rebooter, et le lecteur sera en mesure de reconnaître toute carte, RAM ou SunDisk, dès qu'elle sera placée dans la fente avant de l'appareil, et de faire tout seul la différence entre "volumes stackés" on non.

Prenons le cas d'une carte "stackée" : on y accède par les ordres et désignations d'unité standard du DOS, et notamment XCOPY (ou un "shell" quelconque : PCTools, XTree, DosShell, Norton Commander...), ce qui permet d'effectuer sauvegardes et restaurations à grande vitesse, d'utiliser un streamer, etc. La seule restriction, c'est que les fichiers non compactés ne sont plus visibles, et qu'il vaut mieux pour la santé de la partition compactée, éviter d'utiliser FORMAT <unité compactée:> (résultat garanti, d'autant que l'utilitaire STACPALM.EXE de la disquette SunDisk n'est, à dessein, pas suffisamment documenté pour permettre de réinstaller commodément une nouvelle partition compactée sans autre information. Conditions de licence et nécessités commerciales obligent!).

Après avoir "rebooté", une bonne surprise : non seulement les cartes SunDisk, mais toutes les cartes "stackées" (ACE Double-Cards, ou' autres) sont reconnues, et le lecteur fait de lui-même la différence entre celles-ci et les cartes ordinaires non compactées.

La disquette d'installation contient, par ailleurs, divers fichiers dont l'usage est à réserver aux spécialistes. Notamment, *Il NE FAUT PAS* utiliser celles des instructions dont le nom commence par TC, qui sont propres aux véritables cartes Flash (ce que les SunDisk ne sont pas, puisqu'ils contiennent un circuit auxiliaire générant la "tension de brûlage" pour l'écriture avec un HP-95 ou un ATARI Pocket, écriture d'ailleurs beaucoup plus rapide et souple que sur les cartes Flash...).

Mentionnons, pour clore le chapitre Installation, qu'il est possible d'utiliser une carte "stackée" sans avoir installé la ligne DRIVER=STACKER.COM... Il suffit de lancer le "résident" STACKER.EXE en tapant la ligne suivante :

```
[<chemin>]STACKER.EXE <lecteur:>
```

C'est une méthode qui permet de lire sur un PC non équipé une disquette ou un disque amovible "stackerisé", même sans disposer du logiciel. On se sert du fichier présent sur ladite disquette, qui est "visible" tant qu'il n'a pas été lancé, et "résident" ensuite. Il impose toutefois de "démonter" le volume en tapant EXIT en fin d'utilisation ou en cas de changement de volume dans le lecteur, et est *BEAUCOUP PLUS LENT* que la méthode normale (par contre, il fonctionne en lecture et en écriture, ce que ne font pas certains de ses concurrents...). Il est donc à réserver à des accès occasionnels sur un HP-95 non équipé (STACKER.EXE est présent et caché sur chaque carte SunDisk du modèle compressé).

Performances :

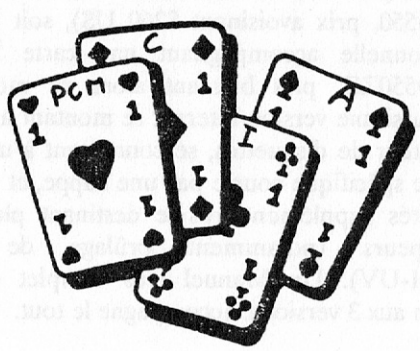
Les tests de rapidité et compatibilité ont été effectués sur deux machines assez différentes : un XT portable sous MS-DOS 3.30 (Panasonic CF-170, assez rapide dans sa catégorie) et un Dell 386SX portable sous MS-DOS 6.0 (ce dernier cas visant à tester le comportement des "drivers" avec un CONFIG.SYS "à géométrie variable", le lancement sous Windows, la possibilité de charger les "drivers" en HIMEM, et le freinage éventuel du PC par la lenteur des accès en écriture sur une carte Flash telle que la SunDisk.).

Il n'y a guère de restrictions à noter. Les accès se font à une vitesse intermédiaire entre celle du disque dur et celle d'une brave disquette, sauf lorsqu'il est fait usage de STACKER.EXE (donc, lorsqu'on n'installe pas à demeure le DEVICE...).

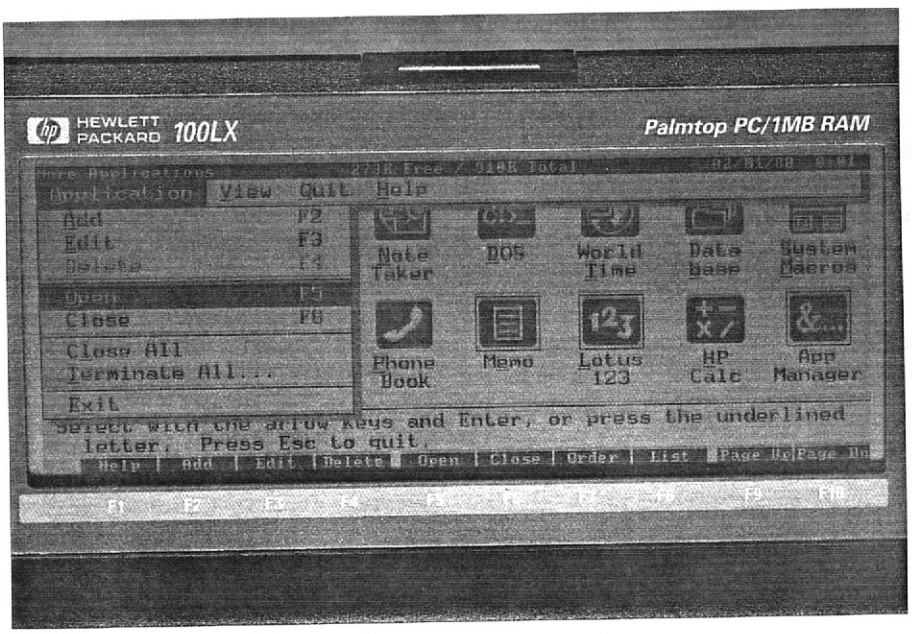
Une sauvegarde complète des 20 Mo de la carte SunDisk compactée, avec l'aide de XCOPY (ou de votre utilitaire préféré, Pctools, Xtree ou autres), ne vous prendra plus que quelques minutes - essayez d'en faire autant avec le câble série, tiens! - de même que la réinstallation des fichiers ainsi sauvegardés.

Un dernier mot, pour les possesseurs du lecteur PAMCO CD-3568, qui était commercialisé par EduCALC jusqu'il y a 3/4 mois à un prix d'environ $ 150 US : un autre éditeur, STEELE CREEK TECH., Inc., vient d'en remettre à niveau les "drivers", pour lui permettre de reconnaître les SunDisk. J'espère pouvoir vous en dire plus dans le prochain numéro...

Eric Gengoux (108)




HEWLETT PACKARD 100LX
Palmtop PC/1MB RAM


HEWLETT PACKARD 100LX
Palmtop PC/1MB RAM
Application View Quit Help
Add F2
Edit F3
Close F6
Close All
Terminate All...
Exit
Note Taker
DOS
World Time
Data Base
System Macros
Phone Book
Memo
Lotus 123
HP Calc
App Manager
letter. Press Esc to quit
Help | Add | Edit | Delete | Open | Close | Order | List | Page Up | Page Dn

LE COIN DES CODES

La compilation de certains programmes, tels ceux écrits en assembleur, nécessitent souvent un logiciel que ne possèdent pas tous nos lecteurs. Le *Coin des Codes* permet de résoudre ce problème.

Note importante :

Même si la présentation des listings est identique, le traitement de ceux-ci est différente suivant le programme d'entrée et la machine de destination. Chaque listing est prévu pour être entré sur sa machine de destination. Par exemple, ne tentez pas d'entrer un programme HP48 dans un fichier MS-DOS à l'aide du programme MAKEDOS. Vous obtiendrez un fichier que vous ne pourrez pas transférer dans la HP48.

Programmes HP28

Par rapport aux méthodes habituelles sur HP48, notre méthode effectuant un calcul local du checksum en fin de chaque ligne permet de faciliter la recherche d'erreurs, par rapport à une même recherche dans une chaîne de plusieurs centaines d'octets.

Tapez les deux programmes ASSCOD.28 et INPUT.28 avec la plus grande attention car leur mauvais fonctionnement peut entraîner un désordre fatal pour les objets contenus dans votre machine. La commande SPEED peut être incluse si vous possédez ce programme. La commande ASC→ peut remplacer les lignes avec @@ jusqu'à *EVAL*.

ASSCOD.28

```
« SPEED RCLF HEX 64 STWS 1 SF "" 'tmpcod' STO
"nombre d'octets                    @ chaine se
                                    @ terminant
"                                   @ 2 par NEWLINE
17 INPUT 1 CF STR→ 2 * 16 DUP2 / IP 3 ROLLD MOD
DUP2
  IF
  THEN 1 +
  END 3 ROLLD 1 + 4 ROLL # 0h DUP → n r f s o
  « 1 SWAP
    FOR i
      DO "ligne " "00" i 1 - R→B →STR 3 OVER
         SIZE 1 - SUB + DUP SIZE DUP 2 - SWAP SUB
         + DUP "                    @ chaine
               chaine               @ encadrée par
               " +                  @ 2 NEWLINE
               "---- ---- ---- ----"
               n i <
               IF
               THEN r DUP 4 / IP + SWAP OVER 1
                 SWAP OVER - SUB SWAP 18 +
               ELSE 38
               END 'o' STO +
               "                              @ NEWLINE
               " + 1 FS?C
               IF
               THEN ROT SWAP CLLCD 1 DISP HALT SPEED
               ELSE o INPUT
               END DUP
               WHILE DUP " " POS DUP
               REPEAT DUP2 1 SWAP 1 - SUB 3 ROLLD
                  1 + 25 SUB
                  IF DUP " " ==
                  THEN DROP
                  ELSE +
                  END
               END DROP 0 OVER SIZE 1 SWAP
               FOR j OVER j DUP SUB NUM j * +
               NEXT s + DUP # FFFh AND
               "                  @ NEWLINE
               somme de controle  @ NEWLINE
                ---               @ NEWLINE
               #"
               6 ROLL SWAP + 34 INPUT STR→ ==
               IF
               THEN SWAP DROP 1
               ELSE DROP2 1000 1 BEEP 1 SF 0
               END
      UNTIL
      END 's' STO
      WHILE DUP " " POS DUP
      REPEAT DUP2 1 SWAP 1 - SUB 3 ROLLD 1 +
         19 SUB +
      END DROP 'tmpcod' DUP RCL ROT + SWAP STO
    NEXT f STOF
  » tmpcod
  # 20204A04F3D02C67h # F80004F02C96040Ch     @@
  # 8DCC05081F804F27h # 313103190F818341h     @@
  # 681808AE91B51391h # 45DC061171085168h     @@
  # F3D3CEAA125E5D8Eh # 2F9004h 28 STWS #0 OR @@
  64 STWS 1 7                                 @@
     START # 3B82h SYSEVAL                    @@
     NEXT # 20238h SYSEVAL # 4F3Dh SYSEVAL    @@
  EVAL "fin" CLLCD 1 DISP                     @@
»
```

INPUT.28

```
« SWAP 1
  WHILE OVER CLLCD 1 DISP
  REPEAT
    DO
    UNTIL KEY
    END
    IF DUP "ENTER" ==
    THEN DROP 0
```

```
    ELSE
      IF DUP "BACK" ==
      THEN DROP 1 OVER SIZE 1 - SUB
      ELSE + DUP SIZE 3 PICK - 2 + 5 MOD NOT 1 FC?
        AND
        IF
        THEN " " +
        END
      END 1
    END
  END SWAP 60 SUB
»
```

Donc à partir de maintenant pour tout assemblage de chaîne de codes procédez de manière suivante:

1- lancez le programme ASSCOD.28
2- donnez le nombre d'octets (1/2 oct. compris) puis validez avec ENTER.
3- tapez chaque ligne de codes, correspondant au numéro de ligne à 3 chiffres, sans les espaces et validez.
4- tapez la somme de contrôle et validez. S'il y a erreur la ligne de codes sera demandée à nouveau après émission d'un BEEP. L'appui sur EDIT fera apparaître la ligne des codes qui pourra être corrigée. Relancez avec CONT.
5- stockez le programme assemblé dans la variable donnée en tête.
6- si tout s'est bien déroulé vous pouvez purger tmpcod qui contient la chaîne de codes.

Programmes HP48

Ceux qui n'ont pas encore de programme assembleur pourront procéder de la manière suivante :
- par mesure de sécurité sauvegardez vos programmes et fichiers, éventuellement verrouillez vos cartes RAM pour devenir ROMs.
- tapez le programme ASSCOD.

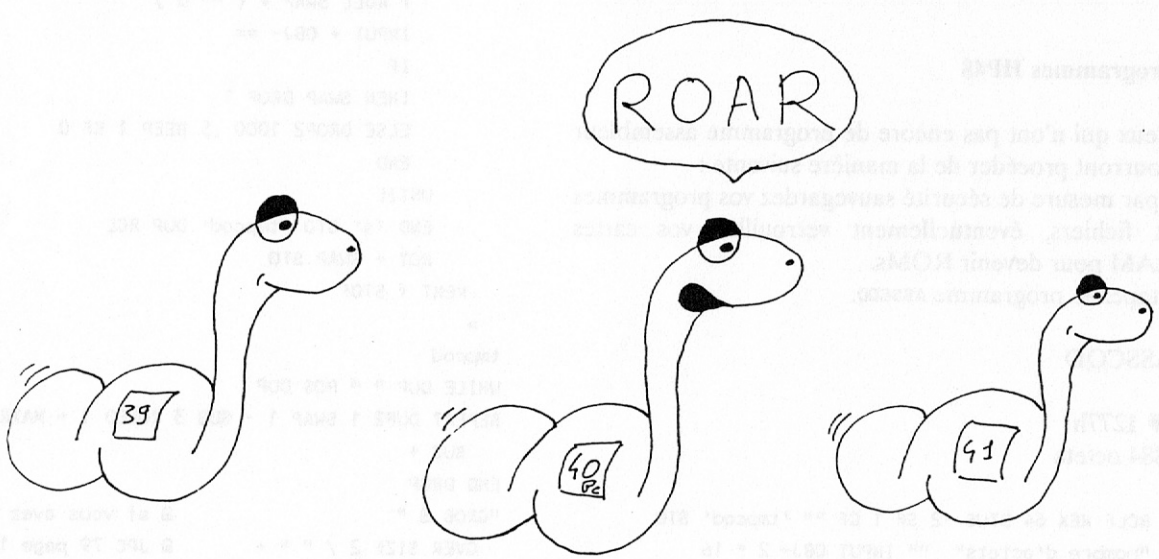
ASSCOD

1277h
884 octets

```
« RCLF HEX 64 STWS -2 SF 1 CF "" 'tmpcod' STO
  "nombre d'octets"  "" INPUT OBJ→ 2 * 16
  DUP2 / IP 3 ROLLD  MOD DUP2
  IF
  THEN 1 +
  END 3 ROLLD 1 + 4 ROLL
  # 0h → n r f s
    « 1 SWAP
      FOR i
        DO
          "ligne        "
          "00" i 1 - R→B →STR 3 OVER SIZE 1 -
          SUB + DUP SIZE DUP 2 - SWAP SUB + DUP
          "                    @ chaine commençant
          chaine" +            @ par ← (newline)
          "                    @ chaine commençant
          ___ ___ ___ ___"     @ par ← (newline)
                               @ et composée de 4
                               @ groupes de 4
                               @ CHR 95 obtenus par
                               @ α shift bleu ×
                               @ 1 espace séparant
                               @ 2 groupes
          n i <
          IF
          THEN 1 r DUP 4 / IP + SUB
          END + 1 FC?C
          IF
          THEN { "" α }
          ELSE ROT
          END INPUT 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 s + DUP # FFFh AND "#"
          "                  @ "somme de controle"
          somme de controle  @ précédée et suivie
          ___"               @ de ← (newline)
                             @ puis 3 CHR 95
                             @ (α shift bleu ×)
          7 ROLL SWAP + { "" α }
          INPUT + OBJ→ ==
          IF
          THEN SWAP DROP 1
          ELSE DROP2 1000 .5 BEEP 1 SF 0
          END
        UNTIL
        END 's' STO 'tmpcod' DUP RCL
        ROT + SWAP STO
    NEXT f STOF
  »
tmpcod
WHILE DUP " " POS DUP
REPEAT DUP2 1 SWAP 1 - SUB 3 ROLLD 1 + MAXR
   SUB +
END DROP
"GROB 8 "                    @ si vous avez ASC→
  OVER SIZE 2 / " " +        @ JPC 79 page 14
  + SWAP + STR→          @ vous pouvez remplacer
  # 4017h SYSEVAL            @ ces lignes
  # 56B6h SYSEVAL            @ par ASC→
  DROP NEWOB                 @ (plus rapide)
»
```

Le mode d'emploi est le même que pour la HP28.

```
FFACT28S.ARC                    (HP28)
# 0AC9h                276 octets

    0123 4567 89AB CDEF   sm

000: 76C2 0DDB 9038 C11A   E5E
001: 4020 3392 0100 0000   862
002: 0000 0051 062B 803A   4DC
003: CD37 6C20 6F52 1F34   269
004: 7109 F207 6C20 A402   F07
005: 0B35 C05E 020A 4020   B24
006: A402 0FFE E328 D11A   A0D
007: DE11 84D1 15E0 20A4   721
008: 020B EC11 3441 184D   44B
009: 11E5 E11A DE11 BEC1   3F9
00A: 1584 21B3 5C06 9C20   122
00B: E610 08F7 2F40 1360   D73
00C: 61B0 0FFF 30F1 5C00   B63
00D: 7134 D610 BC68 18F2   97D
00E: 410A 818F 248F AB9E   9A7
00F: 1184 1121 4016 411B   55B
010: 0613 6061 0913 511B   14D
011: C68F 2405 07AF 0143   E29
012: 1311 74D0 1031 4310   921
013: 2071 35D0 E42B 1511   5BA
014: 1191 3411 BD7D 2060   2AD
015: 5AF0 1521 11AA F182   02D
016: 281E 8325 0A78 A748   DD8
017: AECE 07A7 9154 116B   B50
018: A928 1281 2812 8120   683
019: 4CF5 9B8A AD01 4411   3D4
01A: 3E41 0311 ACE1 0A8A   272
01B: EF82 007C ED58 1ED7   207
01C: 119C 9134 0615 A015   E18
01D: F015 9015 C017 0180   9B1
01E: CF57 E071 35C9 1343   66A
01F: 1031 5F01 5C11 811C   354
020: 0CD5 FE7D 0011 0141   F78
021: 8DCE 5218 D8B0 5009   CBC
022: F200 9F20             455

TOUPPER                         (HP48)
# DCBFh                 78 octets

    0123 4567 89AB CDEF   sm

000: D9D2 02BA 81D9 F81D   F73
001: 0040 D9D2 0756 60CC   D95
002: D20F 6000 8FB9 7601   AB8
003: 4313 0164 1428 18F8   78F
004: 4819 F016 2311 6AE5   4FD
005: 31A7 AE78 A872 CC16   3D3
006: 114E 9E10 F9E7 BE81   35B
007: 86AF 8186 AF14 C69D   2D2
008: F8F2 D760 1421 6480   ECB
009: 8CB2 130B 2130        EBB
```



40eme rugissant.

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.

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