JPC
JUILLET/AOUT 1994
NUMERO 98
Le numéro 40 F

A PROPOS DU CLUB

Le Bureau	Editorial	*1*
	Courrier du Coeur	*2*
G. Toublanc	Libres Propos	*2*
J. Belin	Programmation : Mythes et Réalités	*6*

HP28

G. Toublanc	Remontée de directory	*8*
J.F. Garnier	Compression des Sysevals (Acte III)	*8*
G. Toublanc	Graphisme sur HP28	*9*

HP48

P. Basset	Diagramme de Bode	*22*
J.F. Garnier	Précision étendue	*22*
G. Toublanc	Réduction de programmes (Acte II)	*25*

HP95 / HP100 / HP200

J. Belin	Le HP200LX	*28*
G. Toublanc	Calcul des moments d'inertie	*30*
	Le coin des codes	*34*

EDITORIAL

Il est très dur, pour un éditeur de journal de décider si il doit retarder ou non sa sortie, lorsque il est dans l'attente imminente de l'arrivée d'informations très importantes. Puis lorsque ces informations se mettent à a arriver à un rythme important, de décider à quel moment il en a suffisement pour publier l'article attendu. Ce mois ci, nous avons décidé d'attendre. En effet ce qui n'était en ces derniers mois qu'une rumeur s'est révélé tout à fait fondé, puisque le mois d'août verra la sortie du HP200. Tout ce que HP a révélé, plus le reste, dans ce JPC !

Récement, un adhérent nous a demandé de lui donner des idées d'articles. Ne sachant pas quels étaient ses domaines d'intérets, il nous a été impossible de lui répondre (déjà que nous avons du mal à trouver des idées pour nos propres articles !). Heureusement, deux articles arrivés ce mois ci, parlant d'électronique ou de mécanique, nous ont rappelé qu'il ne fallait pas chercher très loin pour trouver des sujets d'article. Que ce soit dans le cadre de votre travail ou celui de vos études, nous sommes sûrs que vous avez dans la mémoire de votre machine un petit programme ou un fichier (Lotus ou autre) dont vous vous servez tous les jours, et que vous jugez indispensable. Puisque il vous semble utile, il est très possible qu'il puisse rendre aussi de grands services à d'autres utilisateurs. En effet, les activités des adhérents est suffisament variée pour que des articles de mécanique, électricité, chimie ou autres trouve une audience suffisante.

Ce mois ci, les utilisateur de HP48 et HP100 trouverons peut-être leur rubrique préférée un peu courte, et celle concernant la HP28 trop importante. Rapelez vous que nous traitons à égalité toutes les machines, et que même les HP28 ont le droit de d'envahir le journal. Ceci dit (et actualité HP200 oblige), nous avons décidé de reporter la deuxième partie de l'article sur l'affichage d'animations sur HP95/HP100 au mois prochain, afin de ne pas surprendre notre imprimeur (et notre trésorier !) avec un numéro de plus de 50 pages...

Vous trouverez page suivante un article que certains trouverons trop élogieux, et d'autres trop vindicatif (et pour la plupart, les deux en même temps !). Bien que Guy Toublanc fasse partie du bureau, rapelez vous bien que, comme tous les *Libres Propos* écrits par d'autres adhérents, il ne s'agit ici que de réflexions personnelles, et non de la position officielle du Club. A vous de juger, et rapelez vous que nous sommes bien sûr tout a fait disposés à publier tout avis contradictoire.

Pour terminer sur le contenu de ce JPC, notez bien que les articles de Jean-François Garnier concernent aussi bien les HP28 que les HP48.

Maintenant, une bonne nouvelle. Nous avons décidé de réduire de façon très significative le prix de nos disquettes. Cela est principalement dû au désir de respecter les souhaits légitimes de certains auteurs refusant que d'autres fassent quelque bénéfice que ce soit avec leurs programmes (ne vous inquiétez pas, nous n'aurions jamais fait fortune !). Une autre raison, de cette baisse est que lorsque nous avions fixé le prix, nous nous étions basés subjectivement sur celui proposé par certains distributeurs de programmes freeware (nous faisions cependant beaucoup moins d'argent, puisque nous ne faisions pas payer le port, et que nous utilisions des disquettes de marque). Depuis, les prix des revendeurs (ainsi que celui des disquettes pour une moindre part) ayant baissé, nous ne pouvions que suivre le rythme...

Une bonne nouvelle ne venant jamais seule, les Goodies Disks HP48 numéro 10 sont disponibles. Le temps de faire une sélection de nos derniers arrivages, afin de remplir une disquette, et nous devrions pouvoir vous en faire une présentation le mois prochain.

En attendant, essayez de ne pas rayer l'écran de votre palmtop si vous l'ammenez sur la plage !

COURRIER DU COEUR

Emmanuel TARTU
75, rue Jeanne Labourbe
37700 SAINT PIERRE DES CORPS
Tel : (16) 47 32 07 71

Vend :

HP100LX 1Mo; Carte Flash 2.5/5Mo ACE; Connectivity Pack. Sous Garantie (reste 7 mois). Prix : 5500 F.

LIBRES PROPOS

Nous lui devons beaucoup

Depuis la reprise en main du club par notre président Jacques Belin, les numéros de JPC se sont succédés (No 76 de mai 92 au présent No 98) de façon continue et cela, il faut le reconnaître, essentiellement grâce à Jacques qui s'est dépensé sans compter pour remettre sur pieds un club qui pratiquement n'existait plus. Ce redressement s'est fait avec beaucoup de mérite et d'opiniâtreté car en réalité tout repose sur les épaules d'une seule personne : le président (*NDJB : Pas tout à fait quand même. Même si il n'apparait pas dans ces colonnes, Bernard Poughon, notre trésorier, assure parfaitement sa tâche. Qu'il en soit remercié*). Jacques assure avec méthode et rigueur la gestion du club et à lui seul réalise aussi bien ce que faisait autrefois une équipe d'au moins deux personnes. Si certains ne voient l'activité du club que par la production du journal JPC, ils simplifient beaucoup les choses car Jacques remplit avec sérieux de nombreuses tâches pour faire vivre le club. Tout ce travail est fait dans le bénévolat le plus total car notre association n'est ni une entreprise commerciale, ni un pseudo-club servant à canaliser un type d'usagers dans un magasin. Non seulement il n'était pas aisé de refaire fonctionner le club mais il a fallu faire face aux multiples entraves et attaques destructrices de la part de gens qui ont fait le maximum, avec tous les moyens à leur disposition, pour accélérer la disparition de notre association non pas parce qu'elle les gêne mais parce que leur raison de vivre est de détruire.

Pour que le miracle continue il faudrait que les éternels passifs fassent au moins un petit effort de participation au sein du club pour encourager Jacques et le récompenser pour les sacrifices personnels consentis.

Je pense qu'il était l'heure de rendre hommage à celui qui fait le maximum pour notre association.

A chacun sa façon d'être

S'il existe un certain nombre de clubs pour usagers de calculatrices et palmtops HP, ils se distinguent les uns des autres en de nombreux points et sont souvent complémentaires. Pour le nôtre, sil n'est pas sans défauts, il se caractérise par :

- Une totale indépendance vis-à-vis de la marque, des distributeurs ou d'un quelconque magasin. Cela nous coûte cher mais nous permet aussi de diffuser des informations sans contraintes commerciales alors que d'autres clubs sont trop près du négoce pour se payer ce luxe.

- La possession d'une salle de réunions mensuelles à Paris, ce que certains peuvent nous envier, là aussi ce n'est pas totalement gratuit mais cela donne le loisir de discuter largement pendant trois heures et sans les contraintes qu'imposent des rencontres dans un magasin exigu.

- La publication d'un journal de liaison dont la périodicité (10 fois par an) est exceptionnelle parmi ses homologues puisque la mode est plutôt à une périodicité bimestrielle, trimestrielle quand ce n'est pas tous les 4, 5, 6 mois etc...! Journal pouvant recevoir, sans contraintes de longueur, des articles de nature très diverse, y compris les fichiers sources et non pas seulement des listes de codes contrairement à ce qui se pratique très souvent mais qui n'est pas forcément acceptable pour ceux qui ne se contentent pas d'être de simples consommateurs. Le club n'ayant pas de contraintes commerciales, les rubriques sont ouvertes aux différents types de machines et non pas au dernier type sous prétexte qu'un magasin n'a plus d'intérêts commerciaux à tirer des modèles moins récents. La disponibilité des numéros de JPC sans devoir se lamenter pendant des semaines et des mois pour les recevoir.

- La diffusion sur support magnétique de logiciels ou documentations du domaine public ce qui est beaucoup moins onéreux que la diffusion par les serveurs surtout lorsqu'on est pas à proximité de ceux-ci (problème des provinciaux éloignés de la capitale).

- Un interlocuteur (Jacques) sérieux, compétent, permanent et non pas fantomatique.

- Un club qui a un passé avec des méthodes de gestion qui ont fait leurs preuves (lorqu'il y a quelqu'un pour les appliquer évidemment).

Avec tous ces atouts que manque-t-il à notre club pour être dans la catégorie des heureux?

La réponse est la même que celle pour tous les autres clubs qui vivent du bénévolat : moins d'adhérents passifs donc plus d'actifs.

A propos de réflexions

Parmi les désagréments, pour celui qui essaie de faire vivre un club, il en est un qui paralyse littéralement : celui provoqué par l'indifférence. Ne pas savoir si ce que l'on (quiconque participe) fait intéresse au moins une personne est le pire des maux. Aussi je remercie Robert Amram d'avoir réagi en nous livrant ses réflexions dans le JPC 97.

Si Jacques a apporté des éléments de réponses aux questions posées par Robert, je voudrais préciser quelques points. Je suis un vieil adepte des petites machines HP (HP45, 55, 97, 25, 41, 71, 28, 48) et je n'ai pas connu tout de suite de club HP. J'ai appris à programmer avec les seuls manuels HP pendant une assez longue période d'isolement. Puis j'ai découvert le club PPC Toulouse qui fut un excellent club pour HP41. Jusque là je ne connaissais pas les langages de bas niveau (assembleur, forth). Enfin PPC Paris fut la révélation après avoir acquis un HP71. C'est en lisant dans JPC les fichiers-sources des programmes en langage forth ou assembleur que je me suis initié à ce type de programmation que je pratique depuis. A l'époque il n'existait pas de livres tels que ceux de Paul Courbis ou Jean-Michel Ferrard, donc les programmeurs en assembleur de PPC Paris étaient mes maîtres à penser.

Si j'ai fait ce petit historique c'est pour justifier ma façon de concevoir mes articles. J'ai appliqué une pédagogie par l'exemple pour ce qui me semble le moinss accessible pensant que les programmeurs en herbe éprouveraient les mêmes types de difficultés que moi-même à mes débuts. Il est possible que mon évolution de programmeur ne se répète pas de la même manière chez d'autres. Si je n'ai jamais écrit, dans JPC, d'articles d'initiation à l'assembleur sur Saturn c'est qu'il existe dans l'ensemble des numéros de ce journal une abondance d'exemples de programmes dans ce langage. Par contre j'ai amorcé une telle initiation dans le journal 48Sxtant où le terrain était totalement vierge.

D'un autre côté, si le plus gros de mes articles a trait à des programmes en assembleur et largement commentés (comme le fait Jacques côté DOS) il me semble que j'ai abordé différents domaines dans JPC, y compris certains articles qui ont un caractère d'initiation (optimisation ou comparaison des différentes programmations en User Rpl).

Pour ma part ce que je peux suggérer, suite aux remarques de Robert, c'est que des gens désireux de débuter en assembleur sur Saturn proposent un petit problème pour le programmer dans ce langage et en donnant les toutes premières explications. Cela aurait l'avantage de révéler les candidatures et pour une fois de savoir qui veut quoi, car ce qui intéresse l'un peut laisser indifférent l'autre. Par exemple Robert qui s'est intéressé aux traitement de caractères ne sera peut-être pas tenté par le graphisme qui n'est pas, à mon avis, le premier thème à aborder car pas des plus faciles.

Ou si Robert Pulluard est d'accord, les quelques pages d'initiation de ma main et parues dans son journal pourraient être reprises ici. Dans tous les cas il faudrait que les gens intéressés se manifestent.

Passé ce petit coup de pousse pour le départ, il ne me semble pas possible de faire mieux que de commenter les fichiers-sources des programmes importants et de demander aux lecteurs un peu d'attention et de ténacité. S'il n'y a pas de recettes miracles, la programmation en assembleur ne requiert pas des capacités intellectuelles exceptionnelles et je connais des gens, hors de notre club, qui pratiquent l'assembleur sur Saturn et pourtant font preuve d'un niveau culturel lamentable et sont bien incapables de réaliser un programme correct en User Rpl, langage qu'ils dénigrent certainement parce qu'il fournit des programmes directement éditables et lisibles par chacun.

Diversité et passivivité

Nos calculatrices ont d'abord été conçues pour faire des calculs et certains semblent ignorer cette simple évidence. Le constructeur a étendu les possibilités d'utilisation qui en fait sont limitées par la vitesse et les capacités mémoire. Actuellement les motifs d'utilisation sont très divers et de là les niveaux des utilisateurs puisque l'on pourra arriver au plus bas avec des gens qui ne connaîtront que quelques touches permettant d'utiliser un jeu conçu très astucieusement par un autre mais qui n'ont pas la moindre idée de ce qu'est un programme ni de ce qu'il y a dans les différents menus HP.

Cette diversité d'utilisation devrait avoir pour conséquence une participiation aussi diverse au niveau des articles et programmes. Ce que l'on constate est l'opposé :

- Trop d'adhérents pratiquent l'attentisme : un autre fera bien ce qu'ils devraient faire. Le résultat est que l'on ignore les centres d'intérêts des uns et des autres et l'entraide associative est réduite à néant.

- Certains savent programmer pour leurs besoins personnels mais pensent que cela ne peut servir qu'à eux-mêmes. Egoïsme ou peur de consacrer un peu de temps à écrire quelques lignes explicatives pour les éventuels utilisateurs de leurs programmes ? Là encore le sens associatif est mis à mal.

- Sans programmer on peut être un bon utilisateur de sa machine ainsi que de ses accessoires et bien connaître quelques logiciels. Faire profiter les autres de ses propres expériences permettrait d'élargir le cercle des bons utilisateurs.

- Il y a les peureux qui n'osent pas publier parce qu'ils ont des scrupules. Ils ont totalement tort car leur travail serait profitable à d'autres qui à leur tour pourraient faire avancer les choses. Par exemple le petit programme de conversion de caractères accentués, de Robert Amram, aurait pu ne jamais être publié sous prétexte que Robert jugeait que son programme n'avait pas un degré de sophistication suffisant pour avoir l'honneur de paraître dans les colonnes de JPC. Cela aurait été une grosse erreur car son travail m'a révélé un besoin auquel je n'avais pas pensé et m'a incité à aller un peu plus loin que Robert. Le résultat : nous sommes au moins deux à utiliser le même type de programme car celui-ci m'est très utile. J'aurais été très heureux que des utilisateurs nous révèlent d'autres applications. Contrairement à ce que devrait être un livre fait par un spécialiste chevronné, JPC n'est pas un recueil de programmes parfaits mais au contraire de choses perfectibles grâce au concours des uns et des autres comme cela l'était par le passé. La périodicité de parution de JPC devrait être un atout dans le sens évolutif et l'émulation, à l'opposé d'un ouvrage qui reste statique pendant de nombreux mois, sinon des années.

- Enfin il y a les gens qui se réservent pour la rentabilité. Il y est tout à fait naturel que ceux qui ont réalisé des travaux importants soient enclin à commercialiser le fruit de leur travail. Mais lorsque ces personnes profitent de ce qui se fait dans un club il semblerait naturel qu'en retour elles fasse don au club de quelque pour-cent de leur savoir sans mettre en péril leurs produits commerciaux. Si nous avons l'honneur de compter deux auteurs d'ouvrages pour HP48, parmi nos adhérents, nous n'avons jusqu'ici pas pu lire une ligne de leur prose alors que l'un d'eux a contribué à une revue, mais qui se vend en kiosque donc certainement beaucoup plus publicitaire qu'un journal comme JPC. J'espère qu'ils sauront rendre caduques ces dernières lignes.

Les gloutons

Dans le précédent éditorial Jacques a révélé la fin de qualité de dépositaire de JPC d'un magasin qui avait assuré ce rôle pendant de nombreuses années comme le faisait un autre magasin aujourd'hui disparu et qui m'a appris l'existence de JPC : La Règle à calcul. A une certaine époque JPC figurait en bonne place à la devanture de ces deux magasins. Puis après la disparition de la RAC, comme on l'appelait, l'autre magasin a profité du manque de concurrence pour occuper une place plus importante dans le négoce des petites machines HP et cela sous l'impulsion d'un vendeur compétent et très dynamique. Celui-ci a dû changer d'employeur, malgré ses compétences, car ne pouvant passer vendeur à part entière après avoir fait largement ses preuves (on en prend un autre et on repart à zéro utilisant à plein une certaine loi sur l'emploi). C'est alors que s'est infiltrée dans ce magasin une espèces particulière : les gloutons. Ce sont des gens qui ne supportent pas que d'autres existent, donc détruisent avec tous les moyens en leur possession, entrainant derrière eux les faibles qui ne savent pas juger par eux-mêmes. Spécialistes de la contre-publicité et du dénigrement ils sont arrivés à leurs fins : éjecter JPC du magasin en question. Le responsable de ceci est passé de vendeur d'appoint à vendeur permanent et semble confondre négoce et club car membre du bureau d'un club qui produit de façon aléatoire un journal au contenu informel et de valeur nullement en rapport avec les prétentions affichées (six numéros péniblement sortis en plus de deux ans, dont la périodicité actuelle est de six mois ne me semble pas un exploit). Je n'entrerai pas dans les détails qui sont peu reluisants mais il m'a semblé que nous ne pouvions pas recevoir indéfiniment des coups de la part de gamins ayant la prétention de faire la pluie et le beau temps dans le monde HP48 (nous ne sommes pas les seules victimes).

Les librairies miracles pour HP48

Certains ne jurent que par les librairies. Le gros avantage de celles-ci est que leurs fonctions ou commandes peuvent être appelables directement quel que soit le répertoire actif. Faut-il pour cela faire des librairies avec tout? Non, cela ressort de l'absurdité pour les raisons suivantes :

- Chaque librairie a un numéro d'identification et ses fonctions ou commandes ont un numéro d'ordre dans la librairie. Le problème est celui des librairies dont les auteurs s'attribuent sauvagement des numéros sans se préoccuper de ce qui existe déjà, ce qui amène à une belle collection de désagréments lorsque le même numéro est utilisé par plusieurs librairies. Le problème s'est encore aggravé avec la sortie des HP48G/GX et les programmeurs qui ont doublé les

numéros des librairies existantes pour les séries S. Il y a aussi ceux qui utilisent des mots de fonctions de la rom, leurs fonctions deviennent prioritaires et sèment une belle pagaille (exemple SYSEVAL de Pac Math).

- Lorsqu'il y a beaucoup de librairies dans une HP48GX équipée de cartes à forte capacité (carte de 1Mo par exemple) le temps de disponibilité de la machine après l'allumage est loin d'être négligeable et même désagréable.

Je ne vois aucun intérêt de mettre en librairies tous les jeux. Celles-ci doivent être réservées aux utilitaires dont on se sert très souvent. Si les jeux se présentent en France si souvent sous forme de librairies c'est pour les protéger des copieurs et aussi pour mystifier un peu les naïfs.

La mise en librairie permet aussi de cacher, des regards indiscrets, un contenu subtil dont l'auteur désire garder le secret ou à l'opposé n'a pas été créé en grande partie par un pseudo-auteur.

PPC Paris est-il incapable de produire une librairie pour HP48 alors qu'il avait réalisé JPC Rom pour HP71, module tout en assembleur et même commercialisé aux Etats-Unis. Tout d'abord il y a la librairie Config Picker publiée dans JPC 92 par Pierre de Sacy. Lu Vy Huynh a réalisé une librairie permettant de compiler ou décompiler des programmes écrits en System Rpl avec les mnémoniques de Jean-Michel Ferrard. Cette librairie remise au club est très pratique. D'autre part de nombreux outils pour fabriquer des librairies existent aussi bien sur HP48 que sur PC et la réalisation de celles-ci ne constitue pas obligatoirement un acte de bravoure comme certains le prétendent, du moins au niveau le plus élémentaire. Il aurait été facile de créer celle de PPC Paris en mettant dans un même répertoire ce qui a été publié dans JPC et en ajoutant des compléments pris ailleurs comme certains le font. Je pense qu'il était plus raisonnable de mettre à la disposition des lecteurs de JPC quelques fonctions, pas trop mal programmées et constituant un service à la carte suivant les besoins de chacun et surtout les capacités mémoire des machines sans possibilités d'extensions. Par ailleurs il existe quelques bons utilitaires qui ont été paufinés version après version et qu'il était inutile de copier. Leur seul défaut est qu'ils ne peuvent pas tenir à deux dans une HP48 G ou S. Dans un autre contexte (plus de programmeurs) il n'est pas impossible de concevoir une librairie spécifique du club.

Les serveurs bons serviteurs

Les serveurs ont rendu et rendent des services que nul ne peut sous-estimer mais parfois à quel prix. Certains ne sont utilisables pratiquement que par les proches qui ne sont pas sousmis aux tarifs dissuasifs. Au point de vue téléchargement ce qu'ils proposent n'est pas toujours filtré et documenté. Aussi, si par exemple pour HP48 la liste des logiciels paraît impressionnante, au premier abord, il faut tenir compte de ceux qui font double, triple emploi et même plus ou des différentes versions dont la plus performante devrait seule subsister. D'autre part il y a les les librairies qui sont des réarrangements de choses prises chez différents auteurs. Et pour couronner le tout, une abondance de logiciels non documentés donc non utilisables à cent pour cent. Quant aux forums ils sont souvent envahis par les questions de ceux qui n'ont pas lu les dix premières pages du manuel de leur machine et lassent les gens compétents qui ne peuvent pas rabâcher continuellement les mêmes réponses. Aussi, si l'on fait le compte de ce qui reste de correct après plusieurs mois de suivi, c'est assez décevant. Je n'insisterai pas sur la formulation des messages où trop souvent en France les principes élémentaires de l'éducation sont bien bafoués ni sur l'esprit sectaire de certains propriétaires de serveurs et de leurs émules. Enfin les serveurs ne sont pas obligatoirement très favorables à la mémoire : de bonnes réponses données il y a quelques mois auront disparu car les serveurs n'ont pas des capacités illimitées de stockage.

L'éternel recommencement

Très souvent les clubs, pour machines telles que les nôtres, naissent à partir d'un groupe de jeunes à la fois enthousiastes et généreux et qui doivent se partager entre leurs études et leur passion. Leurs créations n'ont parfois rien à envier aux productions commerciales et font le bonheur des autres adhérents. Puis ces groupes se disloquent car leurs membres sont entrés dans la vie professionnelle et leurs motivations pour leurs chères petites machines se sont évanouies. D'autre part les modèles de machines changent et certains renoncent à réinvestir surtout lorsqu'ils leur en a coûté beaucoup en temps et en argent. Dans ces conditions les clubs ne peuvent subsister que par le renouvellement des groupes créateurs et la persistance de certains piliers qui en assurent la continuité. Certains esprits tels que les *gloutons* feraient bien de méditer sur ce que sera leur cercle (qui a déjà subi des modifications) d'ici cinq ou six ans.

Il y aurait bien d'autres thèmes à aborder dans ces libres propos mais je me suis limité à quelques uns relatifs a la vie de notre club. Cela fera peut-être naître des réactions et prendre conscience qu'une association ne vit et ne peut survivre que par la contribution de chacun sous toutes ses formes.

Guy Toublanc (276)

PROGRAMMATION : MYTHES ET REALITES

Régulièrement, de nombreux adhérents nous expriment leur désir d'apprendre à programmer, mais disent avoir abandonné car ils ne comprenaient rien. En fait, après une petite discussion, je me rends souvent compte qu'ils ont abandonné dès la première difficulté. Cet article a donc pour but de casser quelques idées reçues, afin de leur faire voir que ces dificultées sont plus "culturelles" que réelles (nous avons toujours entendu dire que la programmation était compliquée, et l'assembleur encore plus).

Mais tout d'abord, je pense qu'il est impératif de faire la distinction entre les deux grandes notions *totalement indépendantes* de la programmation, que l'algorithmique, et les langages :
- L'algorithmique traite uniquement de la façon dont doit être formulé un problème. Elle n'est basée que sur quelques concepts simples : opérations (mathématiques ou actions), tests, boucles et sauts. C'est l'arangement de ces concepts qui font le programme, pas le langage.
- Tous les langages ne font qu'utiliser ces concepts. Ce qui fait leur différence, c'est la syntaxe les décrivant, la façon dont sont déclarées et manipulées les données, et la richesse des bibliotèques de fonctions.

Ceci dit, passons à quelques réflexions :

- Si vous ne comprenez pas un listing, demandez vous tout d'abord si c'est le langage que vous ne comprenez pas, ou l'agorithme. En effet, il semble que beaucoup de personnes soient tout à fait capables de décoder les instructions séparément, mais incapables de comprendre la logique qui les unit. Donc, si vous ne comprenez pas ce listing, n'accusez pas le langage, et reportez vous sur un livre mathématique (par exemple), pas sur un livre de programmation.

- Un bon programmeur n'est pas celui qui maitrise parfaitement l'un et/ou l'autre des domaines, mais celui qui sait faire le bon choix du langage à utiliser, afin de mettre en oeuvre son algorithme dans les meilleurs conditions (facilité et temps de développement, performances du programme). Que ce langage soit évolué ou non n'a aucune importance.

- L'assembleur n'est pas plus difficile qu'un autre langage. Il demande seulement plus de rigueur (car il n'y a plus les "gardes fous" existant dans les langages évolués) et plus de travail (car si on ne dispose pas d'une bibliothèque de fonctions, il faut la créer).

- Si vous savez déjà programmer dans un langage, il n'y a aucune raison que vous ne puissiez pas en apprendre un autre. Cependant, il semble que le grand problème des programmeurs expérimentés est qu'ils pensent que la difficulté de changement de programme est inversement proportionnelle à l'expérience. Cela est totalement faux. En fait, mon expérience personnelle m'a montré que tout nouveau langage, même si il parait plus simple que les précédents, implique le réapprentissage TOTAL de la façon dont sont gérées les instructions. Par exemple, si un test est toujours un test, il est toujours accompagné de conditions de mises en oeuvre totalement différentes entre chaque langage. C'est cela que vous devez apprendre et pas ce qu'est un test. Et cela prend *TOUJOURS* un certain temps.

- Contrairement à ce que beaucoup croient, un livre d'initiation n'est pas la panacée. Pour paraphraser, je dirais qu'il est plus efficace d'aller un mois en Angleterre pour apprendre la langue, que de passer six mois à lire des livres d'Anglais. En revanche, ce qui est indispensable, c'est un livre de référence vous indiquant la syntaxe exacte de toutes les fonctions, ainsi que leurs effets réels (par exemple la doc de SASM dans les Goodies Disks 4).

- Même si l'analyse des sources des autres représente un des meilleurs moyens d'apprentissage (c'est bien pour cela que nous publions tous ces sources dans JPC), ce n'est pas en étudiant les programmes des autres que vous apprendrez le plus, mais en corrigeant vos propres erreurs. Ce n'est pas non plus lorsque vous réussissez à écrire un programme du premier coup que vous apprenez. C'est toujours lorsque vous cherchez à corriger vos propres erreurs.

En résumé, le seul moyen d'apprendre la programmation (ou un langage) est de se mettre à son clavier et de persévérer. Et si les premiers jours vous ne comprenez rien, c'est normal !

Jacques Belin (123)

HP28

G. Toublanc	Remontée de directory	*8*
J.F. Garnier	Compression des Sysevals (Acte III)	*8*
G. Toublanc	Graphisme sur HP28	*9*
	Le coin des codes	*34*

REMONTEES LEGERES

Pour remonter de directory en directory, si nos HP-28S ne possèdent pas la commandeUP des HP-48, beaucoup utilisent le programme du même nom du livre *Voyage .... HP28*. Paul Jebeily nous a donné aussi sa version dans JPC 90. Toujours soucieux de loger le maximmum de programmes dans ma HP28S, j'ai fait ma version, en fait une traduction en System Rpl du programme du livre cité et qui est en tout banal User Rpl. L'avantage: de 37,5 octets on passe à 22,5 octets, c'est toujours cela de gagné.

UP
22,5 octets cksum : # D9FEh

```
::
  PATH
  DUP
  LENCOMP
  #1-
  NTHELCOMP
  IT EVAL
;
```

Voir la liste des codes dans *le coin des codes*.

Guy Toublanc (276)

COMPRESSION DES SYSEVALS (Acte III)

La programmation à base de SYSEVAL souffre de défauts rééls, dont un encombrement mémoire relativement important. Il exite différents utilitaires destinés à compacter les séquences #nnnnnh SYSEVAL, en les remplaçant par l'objet correspondant (voir par exemple JPC 96), cependant ils ont l'inconvénient d'interdire une édition ultérieure du programme. Mes réalisations étant rarement figées mais plutôt en constante évolution, cette solution ne me satisfait pas, car m'oblige à conserver à la fois la version source avec les SYSEVAL, et la version compactée, d'où une perte de place à l'inverse du but recherché. Ceci est particulierement vrai pour la HP28 : comme pour la grande majorité des propriétaires de cette machine, je ne dispose pas de possibilité de sauvegarde/restauration sur PC (ceci nécessite une modification matérielle de la machine, que je n'ai jamais entreprise).

Je vous propose une autre solution, qui bien que moins efficace, permet de conserver la possibilité de modification des programmes compactés. Il s'agit de jouer sur la taille des entiers binaires grâce aux fonctions STWS et RCWS. En passant d'une taille de 64 à 20 bits, on gagne 5.5 octets par SYSEVAL. Si vous utilisez de façon intensive cette fonction, par exemple pour réaliser des calculs sur les Long Real (voir mon article Précision étendue sur HP28/48), ce gain n'est pas négligable.

La première idée qui vient à l'esprit est de fixer la taille à 20 bits avant de saisir le programme contenant les SYSEVAL. Malheureusement, cela ne marche pas, car la taille des entiers binaires spécifiée par STWS n'est utilisée par la 28 (ou la 48) que lors de la génération d'un résultat par la machine. Les entiers binaires placés par l'utilisateur sur la pile ou dans un programme sont toujours codés sur 64 bits. Il faut donc faire appel à un programme.

Le programme PACKSE (Pack SysEval) réalise cette fonction. Il utilise les fonctions COM→ et →PRG qui décomposent et reconstruisent un programme (comme LIST→ et →LIST pour une liste). Ces fonctions pour HP28 et HP48 ont déjà été publiées dans JPC 87, avec d'autres utilitaires.

HP28S :
COM→

```
« # 3F41h SYSEVAL   # C610h SYSEVAL »
```

→PRG

```
« # C53Bh SYSEVAL   # 3EEBh SYSEVAL »
```

HP48S/G :
COM→

```
« # 54AFh SYSEVAL   # 18DBFh SYSEVAL »
```

→PRG

```
« # 18CEAh SYSEVAL  # 5445h SYSEVAL »
```

PACKSE

```
«
  RCWS          @ rappel de la taille courante
  20 STWS       @ fixe la taille à 20 bits
  SWAP
  COM→          @ décompose le programme
  → n           @ n : nombre d'éléments du prog
  «
    1 n
    START
      n ROLL    @ rappel de l'élément suivant
      DUP       @ test si c'est un entier binaire
      IF TYPE 10 ==
      @ oui, ajoute 0 pour le tronquer à 20 bits
      THEN 0 + END
```

```
    NEXT
    n →PRG      @ reconstruit le programme
  »
  SWAP
  STWS          @ restaure la taille initiale
»
```

PACKSE a une limitation : il ne traite pas les objets programmes inclus tels que ceux délimités par "«" et "»". Sur la HP48, il serait facile de tester le type programme (8) et d'appeler récursivement PACKSE dans ce cas. Malheureusement, l'instruction TYPE de la HP28S ne différencie pas le type programme du type commande interne, et je n'ai pas trouvé de façon simple de résoudre ce problème.

Le programme ainsi compacté peut être édité librement, cependant le compactage est alors perdu et il faut exécuter à nouveau PACKSE.

Jean-François Garnier (242)

POUR LE PLAISIR DES YEUX

LINE & Cie

Je dédie cet article à Paul Jebeily qui avait publié (JPC 90) un ensemble très intéressant de programmes pour réaliser du graphisme animé ou non.

Dans son article Paul déplorait la lenteur du graphisme sur HP-28. Pour y remédier partiellement je vous propose quelques commandes en assembleur ou en Sytem Rpl.

LINE

Cette commande LINE est réalisée en assembleur et remplace avantageusement le programme SEGMT utilisé par Paul. En effet LINE est d'exécution 600 fois plus rapide que SEGMT, ce dernier étant appelé souvent par d'autre programmes il en résulte une augmentation de rapidité appréciable de l'ensemble et l'exécution des deux programmes de démonstration d'animation graphique de Paul (TEST1 et TEST2) devient très agéable.

Programmer LINE en assembleur sur HP-28 est plus difficile que sur HP-48 car la mémoire occupée par l'écran est divisée en deux parties disjointes correspondant aux moitiés gauche et droite de l'écran. En plus, les pixels de l'écran ne sont pas adressés en continu comme sur les HP-48 mais par colonnes de huit pixels. Ceci a pour conséquence d'allonger la programmation. Aussi j'ai essayé de réduire le plus possible l'enconbrement mémoire pour arriver à 280,5 octets. Il est possible, avec un surplus d'octets d'augmenter la rapidité, en particulier avec les instructions d'échange entre registres de calculs et registres de sauvegarde sur les champs et non avec les registres entiers. J'ai préféré me contenter de la vitesse annoncée mais avec moins d'octets car je pense avoir obtenu un compromis très acceptable.

Voyons les particularités d'emploi de LINE :

il faut fournir les coordonnées des extrémités du segment à tracer, soit :

(x_1,y_1) et (x_2,y_2)

le système de coordonnées repose sur la disposition suivante et est indépendant de PPAR :

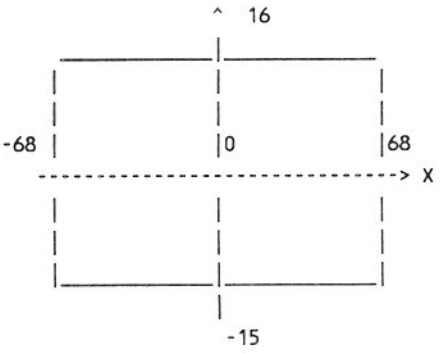
Ce choix se justifie pour les raisons suivantes :

- Trois autres commandes utilisent de façon naturelle ce système. Pour éviter les confusions il est donc nécessaire de s'en tenir à un seul sytème.

- Il m'a semblé que ces commandes seront utilsées plutôt pour du graphisme de *divertissement* que pour des tracés purement mathématiques. Paul nous en a fourni de bons exemples. Cela n'exclut pas que LINE sera utile, par exemple, pour tracer des asymptotes.

Pour gagner de la rapidité et des octets il sera bon de construire les programmes de graphisme en pensant au système de coordonnées de LINE. Sinon on pourra toujours faire des transformations de coordonnées.

Si le système de coordonnées est le suivant :



Pour utiliser LINE il faudra faire :

```
« (68,16) SVAP CONJ OVER +
  ROT CONJ ROT - LINE DROP2 »
```

Les programmes de Paul Jebeily fonctionnent avec les PMIN (1,1) et PMAX (137,32) donc le système suivant :

Ainsi en remplaçant la variable SEGMT par :

```
« (-1,32) SVAP CONJ OVER +
  ROT CONJ ROT + LINE DROP2 »
```

Sans modifier les autres programmes, les démos vont à merveille. Le DROP2 élimine les coordonnées après le traçage car celles-ci sont laissées sur la pile par LINE. Cette dernière possibilité permet de les réutiliser éventuellement pour d'autres tracés (par exemples carés, rectangles etc...).

Le plus rationnel serait de travailler directement avec le système de coordonnées de LINE pour éviter les conversions et ainsi gagner en octets et vitesse.

Enfin, pour la disposition suivante, il suffira de remplacer dans les programmes ci-dessus le nombre complexe, au début, par (0,31).

En armant le flag 9 on peut effacer une ligne au lieu de la tracer si ce flag est désarmé. Donc les portes de l'animation graphique sont ouvertes avec des polygones de toutes natures. J'ai choisi le flag 9 pour raison mnémotechnique : 9 (neuf) pour mise à neuf ce qui est simple à retenir.

LINE met la vitesse maximale de la HP-28 comme la commande SPEED donc pour revenir à la vitesse normale il faudra utiliser la commande SPOFF. Si vous ne l'avez pas, voici ma version en 17 octets au lieu de 20 octets pour les versions habituelles.

SPOFF
17 octets checksum : # 8FF9h

```
CODE
        ad0ex
        d0=(5) #fff00
        lchex  8
        dat0=c 1
        govlng #03801
ENDCODE
```

Revenons au programme LINE. Pour qu'il soit rapide il faut employer un algorithme de calculs en nombres entiers. Comme pour les tracés de cercles c'est celui de Bresenham qui convient. Sans entrer dans tous les détails voici le principe :

Les extrémités du segment de droite à tracer étant :

(x_1,y_1) et (x_2,y_2)

on a :

$x_i = x_1$
$y_i = y_1$
$\Delta x = (x_2 - x_1)$
$\Delta y = (y_2 - y_1)$
s_1 = signe de $(x_2 - x_1)$
s_2 = signe de $(y_2 - y_1)$

si $\Delta y > \Delta x$ alors échange Δy <-> Δx
drapeau = 1
ou bien drapeau = 0

err = 2*Δy - Δx

pour i = 1 à Δx
    allume (x,y)
    tant que err positive
        si drapeau = 1 alors
            $x_{i+1} = x_i + s_1$
        ou bien
            $y_{i+1} = y_i + s_2$
        err = err - 2*Δx
    fin de tant que

si drapeau = 1 alors

$y_{i+1} = y_i + s_2$

ou bien

$x_{i+1} = x_i + s_1$

err = err + 2*Δy

Pour des explications plus approfondies vous pourrez consulter par exemple l'ouvrage de David F.Rogers : *Algorithmes pour l'infographie* (éditions McGraw-Hill)

Du fait de la possibilité d'avoir des coordonnées négatives et celles-ci étant converties en nombres hexadécimaux, il faut que ces derniers soit signés ce qui sera reconnu par le bit de poids fort du champ A dans les registres. Aussi les coordonnées dont la valeur absolue est supérieure ou égale à 10000 sont divisées par 10 donc le maximum accepté est 9999 ce qui par rapport aux dimensions de l'écran est réaliste.

Voici le programme miracle :

```
************************************************
*                LINE pour HP28S               *
*          280,5 octets  cksum # A943h         *
*                                              *
*    auteur Guy Toublanc          le 26/06/1994 *
*                                              *
*   syntaxe :                                  *
*     niveau 2 : ( X1 , Y1 ) origine du segment *
*     niveau 1 : ( X2 , Y2 ) extrémité du segment *
*                                              *
*                                              *
* X1, Y1, X2, Y2 : réels                       *
* Les extrémités P1 et P2 quelconques dans le plan *
* complexe                                     *
*                                              *
*  sytème de coordonnées absolues :            *
*                                              *
*       0  ------------------------> 136       *
*          |                                   *
*          |                                   *
*          |                                   *
*       31 V                                   *
*                                              *
*  flag 9 désarmé : ligne tracée               *
*  flag 9 armé    : ligne effacée              *
*                                              *
* fichier source compilable sur PC par RPLCOMP de HP*
************************************************

CODE
        gosbvl  =SAVPTR
        c=0     w       * pour initaialiser
        c=c-1   a       * c(a) := FFFFFh
        c=0     b       * c(a) := FFF00h
        d0=c            * @ speed
        dat0=c  xs      * pour speed maxi
        d0=(5)  #c0111  * @ flag 9
        a=dat0  1       * flags 12 à 9
        clrst           * intilialise st 2 et 3
        ?abit=0 0       * 9 FC?
        goyes   lineon  * oui alors on allumera
        st=1    3       * pour ligne à effacer
lineon
        c=dat1  a       * adresse complexe niv.1
        gosub   pop&conv * X2 : conversion en hexa
        r4=c            * sauve X2 -> R4
        gosub   popim   * conversion Y2 en hexa
        r0=c            * sauve Y2 -> R0
        d0=d0+  5       * @ 2ième niveau
        c=dat0  a       * adresse 1er complexe
        gosub   pop&conv * X1 : conversion
        r2=c            * sauve X1 -> R2
        gosub   popim   * conversion Y1 en hexa
        r3=c            * sauve Y1 -> R3
        a=r0            * Y2
        ?a#c    a       * Y2 <> Y1 ?
        goyes   okline
okline  c=r4            * X2
        a=r2            * X1
        goc     line    * si test okline
        ?a#c    a       * X1 <> X2 ?
        goyes   line    * oui ligne traçable
        ar3ex
        goto    point   * non alors 1 point

line    p=      4       * pour test des signes
        gosub   XY      * 2|X2-X1| signe -> A(S)
        d=c     w       * sauvegarde pour index
        ar3ex            * sauvegarde 2|X2-X1|
        c=r0            * Y2    a := Y1
        gosub   XY      * 2|Y2-Y1| signe -> A(S)
        ?d<c    a       * delta X < delta Y
        goyes   dx<dy
        cdex    a       * non alors permutation
        st=1    2       * et repère avec ST 2
        ar3ex           * et échange des deltas
dx<dy
        r4=c            * sauve 2DX (ou 2DY)
        r2=a            * sauve X1 (ou Y1)
        cdex    a       * échange 2DX<->2DY
        r0=c            * sauve 2DY (ou 2DX)
        dsrb            * |X2-X1| (ou |Y2-Y1|)
        c=c-d   a       * err= 2DY - DX
        b=c     a       * sauve err
        d=d-1   a       * départ pour compteur

************************************************
*                                              *
* Boucle de tracé des points de la ligne sauf le *
* dernier point.                               *
* suivant que la pente est inférieure ou       *
* à 45° le tracé se fait suivant les abscisses *
* ou les ordonnées                             *
*                                              *
************************************************
```

```
loop    gosub   plot        * trace le point Xi,Yi
        p=      4           * pour test de signes
        ?b#0    p           *  while err >= 0
        goyes   tst         *
        a=r4                * 2DX
        c=r3                * Xi

loop2   c=c+1   a           * Xi + 1
        ?c=0    s           * X2 >= X1 ?
        goyes   x2>x1       * oui Xi+1 = Xi + 1
        c=c-con a,2         * non : Xi+1 = Xi - 1
x2>x1
        b=b-a   a           * err = err - 2DX
        gonc    loop2
        r3=c                * sauve Xi+1

tst     a=r2                * end while A(A) := Yi
        a=a+1   a           * Yi + 1
        ?a=0    s           * Y2 >= Y1 ?
        goyes   y2>y1       * oui Yi+1 = Yi + 1
        a=a-con a,2         * non : Yi+1 = Yi - 1
y2>y1
        r2=a                * sauve Yi+1
        c=r0                * 2DY
        b=b+c   a           * err = err + 2 DY
        d=d-1   a           * actualise index
        gonc    loop        * point suivant

point   gosub   plot        * dernier point et sortie
        govlng  =GETPTRLOOP

plot    p=      0           * A(A) := Yi (ou Xi)
        c=r3                * Xi (ou Yi)
        ?st=1   2           * DY >= DX ?
        goyes   norm
        acex    a           * non : permutation
norm
        d1=c                * sauve Yi
        lc(5)   136
        ?c<a    a           * Xi > 136 ?
        rtnyes              * oui on fait rien
        lc(2)   67
        ?c>=a   b           * Xi <= 67 ?
        goyes   left        * oui demi écran gauche
        a=a-c   a           * Xi pour 1/2 écr. droit
left
        lchex   ffbf8       * adresse 1/2 écr. droit
        gonc    right       * si Xi > 67 saut écr-dr
        lchex   840         * FF840 pour 1/2 écr-g
right
        a=a+a   a
        a=a+a   a
        a=a+a   a           * colonne pour Xi
        a=a+c   a           * adresse colonne
        ad1ex               * @ colonne  A(A) := Yi
        d0=a                * sauve Yi
        lc(5)   31
        ?c<a    a           * Yi > 31 ?
        rtnyes              * oui on ne fait rien
        asrb
        asrb                * Yi en quartets
        cd1ex               * adresse colonne
        c=c+a   a           * adresse quartet pour Yi
        cd1ex               * @ quartet Yi
        ad0ex               * Yi
        lchex   3
        a=a&c   p           * Yi modulo 4
        csr     a           * récupère 1 de 1F (31d)
        gonc    decr        * départ pour position

loop4   c=c+c   p           * position pixel dans nib
decr    a=a-1   p           * compteur
        gonc    loop4

        a=dat1  b           * quartet du pixel en mem
        ?st=0   3           * PIXON ?
        goyes   pixon
        a=-a-1  p           * inversion pour PIXOFF
pixon
        a=a!c   p           * OR pour PIXON ou PIXOFF
        goc     pixon2      * saut pour PIXON
        a=-a-1  p           * inversion pour PIXOFF
pixon2
        dat1=a  1           * quartet modifié
        rtn

pop&conv cd1ex              * @ prologue
        d0=c                * sauve D1
        d1=d1+  4

popim   d1=d1+  1           * @ exposant
        a=0     a
        a=dat1  x           * exposant
        lchex   500
        ?c>a    x           * exposant positif ?
        goyes   exppos
        b=0     a           * 0,.... arrondi à 0
        d1=d1+  15          * pour partie imag.
        gonc    convrtn
exppos
        lchex   003         * si exposant > 3
        ?c>=a   x           * alors exposant = 3
        goyes   exp<4
        a=c     x
exp<4
        d1=d1+  14          * @ digit de poids fort
        ad1ex               * A(A) := D1
        cd1ex               * C(A) := exposant
        p=c     0           * P := exposant
        c=a-c   a           * adresse digit de poids
        d1=c                * faible (mantisse)
        a=0     a
        a=dat1  wp          * mantisse
        b=a     a           * sauvegarde
        d=0     a           * initialisation
        c+p+1               * adresse après mantisse
```

```
        d1=c            * @ le signe
        p=      0
        c=0     a
        lchex   6       * 3e cste de conversion
        rstk=c          * sauvegarde
        lchex   9c      * 2e cste de conversion
        rstk=c          * sauvegarde
        lchex   c18     * 1e cste de conversion
        p=      3       * compteur pour loop5

loop5   a=b     p       * et pointeur de digits
        gonc    decr3   * départ conversion

loop3   d=d+c   a
decr3   a=a-1   p
        gonc    loop3

        p=p-1           * position suivante
        ?p=     0       * plus de digits ?
        goyes   cont    * alors sortie de loop5
        c=rstk          * dépile constante suiv.
        gonc    loop5   * tour suivant de boucle

cont    c=d     a
        b=b-c   a       * résultat en hexa
        a=dat1  1       * signe
        ?a=0    p       * positif ?
        goyes   pos
        b=-b    a       * si non entier signé
pos
convrtn c=b     a       * nombre converti
        rtn

XY      c=c-a   a       * ici P=4   A(S) := 0
        ?c=0    p       * différence positive ?
        goyes   xy2>xy1
        a=a+1   s       * sinon A(S) := 1
        c=-c    a       * et valeur absolue
xy2>xy1
        c=c+c   a       * 2*ABS(différence)
        rtn
ENDCODE
```

Certains programmes de Paul peuvent être simplifiés ou même supprimés. Par exemple VERTIC et HORIZ ne se justifient plus. Le programme SEGMT ci-dessus ou tout simplement LINE, suivant le cas, feront la même chose que VERTIC ou HORIZ mais nettement plus vite. LINE nous fait gagner plus de 140 octets sur la somme des 3 programmes SEGMT + VERTIC + HORIZ.

Remarques :

Le programme LINE en 280,5 octets ne vérifie pas la présence et le type des arguments pour ne pas augmenter l'encombrement mémoire. Cela ne présente que le risque de faire des graphiques inattendus si les 2 nombres complexes nécessaires ne sont pas sur la pile. Vous serez prévenus.

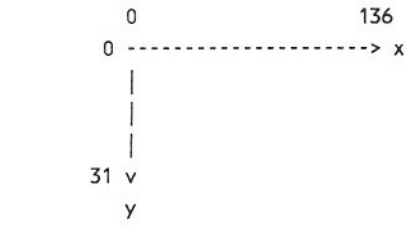
LINE travaille avec des nombres complexes dont les réels les composant sont arrondis par défaut en nombres entiers (donc toujours dans le même sens). Un arrondi au plus près améliore parfois l'aspect des tracés mais pas systématiquement. C'est ainsi que les démos de Paul ont une partie des graphiques plus agréable à l'oeil en modifiant le programme SEGMT ci-dessus de la manière suivante :

```
« (1,32) SVAP CONJ OVER + 0 FIX RND ROT
  CONJ ROT + RND LINE 2 FIX DROP2 »
```

après arrondi le 2 FIX est restauré pour les autres programmes appelés pour les démos. Peut-être qu'il y a une solution pour éviter ces changements de format.

LES TROIS MOUSQUETAIRES

Nos HP-28 ne connaissent pas les fonctions des HP-48 : PIXOFF et PIX? permettant d'éteindre un point de l'écran ou de connaître son état (allumé ou éteint). Paul Jebeily dans son travail cité ci-dessus avait, je pense, essayé de remédier à cette lacune. En ce qui concerne PIXOFF on trouve dans le livre *Voyage ..... HP28* le programme UNPIXEL pour 139 octets. Je vous propose PIXOFF pour 22,5 octets mais dont les coordonnées sont 2 réels pour le système suivant qui est celui utilisé par les programmes graphiques internes de la machine :



Ce système sera aussi celui utilisé par PIXON, PIX? donc pour ceux qui travaillent avec la disposition habituelle de l'origine des coordonnées au centre de l'écran, il faudra transformer les coordonnées de la manière suivante :

```
X = x - 66
Y = 16 - x
```

qui est identique à ce qui a été exposé ci-dessus pour LINE.

Ces programmes sont relativement rapides. Ainsi PIXON est 11 fois plus rapide que PIXEL. Pour PIXON remplaçant PIXEL dans le programme HORIZ, de Paul, le tracé d'une horizontale sur toute la largeur de

l'écran se fait 2,2 fois plus vite avec PIXON qu'avec PIXEL.

Ces fonctions nécessitent toutes deux arguments :

```
niveau 2 : X   1 réel
niveau 1 : Y   1 réel
```

dans le système de coordonnées défini ci-dessus.

PIXON fait donc la même chose que PIXEL. Voici son fichier source.

PIXON
22.5 octets checksum : # 6D07h

```
::
  CK2NOLASTWD
  Dispatch1
  SEVENTEEN
  ::
    COERCE2
    PIXON
  ;
;
```

PIXOFF a l'effet inverse de PIXON et sera utile à ceux qui feront de l'animation graphique avec des créations et disparitions de motifs graphiques, par exemple le train pour HP-48 paru dans JPC 97.

PIXOFF
22.5 octets checksum : # C314h

```
::
  CK2NOLASTWD
  Dispatch1
  SEVENTEEN
  ::
    COERCE2
    PIXOFF
  ;
;
```

Enfin le troisième compère PIX? pour tester l'état d'un pixel : si allumé 1 est renvoyé, sinon 0.

PIX?
25 octets checksum : # D2FEh

```
::
  CK2NOLASTWD
  Dispatch1
  SEVENTEEN
  ::
    COERCE2
    PIX?
    COERCEFLAG
  ;
;
```

Avec ces trois nouvelles fonctions les gens amateurs d'animations graphiques et de jeux devraient pouvoir nous concocter quelques programmes pour le plaisir des yeux ou le divertissement.

Sauf erreur de ma part, je pense que ces trois fonctions (pour 22.5*2 + 25 = 70 octets) devraient pratiquement remplacer l'ensemble des variables du répertoire GRAPH (hormis le sous-répertoire OBJETS) de notre ami Paul et publié dans JPC 90. Si cela était le cas l'économie d'octets et le gain de rapidité seraient très appréciables car ces fonctions opérant directement sur l'écran elles évitent les conversions via LCD→ et →LCD et le travail sur des chaînes. Peut-être des articles en perspective?

Remarque :

Les quatre commandes ci-dessus sont indépendantes de PPAR, c'est donc à vous de gérer les problèmes d'échelle si vous ne travaillez pas directement avec leur système de coordonnées.

Voici les adresses nouvelles par rapport à ce qui est paru précédemment dans JPC :

```
PIXON   # 1ADFEh
PIXOFF  # 1ADF2h
PIX?    # 1AEC8h
```

POLYGONES tous azimuts

Pour ne pas perdre trop de rapidité en traçant des polygones voici un programme en majorité en System Rpl. En fait ce sont trois commandes en une seule et relatives aux polygones réguliers inscrits dans un cercle. C'est l'équivalent des programmes POLYG et PPOLYG de Paul Jebeily, avec d'autres possibilités et ne faisant appel qu'au programme LINE. Ce qu'il fait :

- traçage d'un contour polygonal.
- traçage d'une surface polygonale.
- traçage de la surface polygonale comprise entre deux plygones semblables et concentriques.

Lorsque le flag 9 est désarmé il y a traçage sinon avec le flag 9 armé les polygones ou les surfaces polygonales sont effacées.

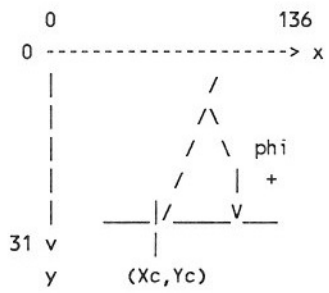
Le système de coordonnées est celui défini pour les autres commandes ci-dessus.

Syntaxe :

```
niveau 3 : ( Xc , Yc ) centre du polygone
niveau 2 : ( R  , r  ) rayons des cercles
```

```
                        circonscrits
    niveau 1 : ( n , phi ) n nombre d'arêtes
                        phi angle de déphasage
```

- Xc, Yc : réels quelconques
- R : rayon du cercle circonscrit au contour extérieur de la surface polygonale.
- r : rayon du cercle circonscrit au contour intérieur de la surface polygonale.
- R, r : réels positifs \- n : nombre d'arêtes des contours polygonaux.
- phi : angle de déphasage en radians.



Si R et r sont inversés, le remplissage entre les contours polygonaux sera aussi inversé donc de l'intérieur vers l'extérieur (voir les exemples).

On pourra obtenir :

- avec R = r : un simple polygone.
- avec R < > 0 et r = 0 : un polygone plein. remplissage vers le centre.
- avec R = 0 et r < > 0 : même polygone mais remplissage du centre vers l'extérieur.
- avec R < > 0 et r < > 0 : surface polygonale évidé si R > r : remplissage vers le centre sinon l'inverse.

On voit qu'il est possible de faire des fondus enchaînés et autres effets sachant qu'en manipulant le flag 9 on peut créer des images progressivement construites puis effacées. Dans certains cas on pourra faire directement des animations graphiques. Pour les autres cas il faudra utiliser une méthode comme celle de Paul, par exemple.

Voici donc ce programme relativement compact et rapide.

POLYG
157,5 octets cksum : # 417Ch

```
::
  xC>R
 %PI %2 %* 3PICK %/
 ROT COERCE 4UNROLL ROT
 xC>R 2DUP %> ITE %-1 %1 5UNROLL OVER %- %ABS
 COERCE 4UNROLL 4UNROLL
 { NULLLAM NULLLAM } BIND
 #1+
 ZERO_DO (DO)
   DUP 2GETLAM %>C% xP>R 5PICK x+ C%RND
   4PICK
   ZERO_DO (DO)
     OVER
     2GETLAM 1GETLAM %+ DUP 2PUTLAM
     %>C% xP>R 6PICK x+ C%RND
     ID LINE SWAPDROP
   LOOP
   DROP OVER %+
 LOOP
 4DROP
 ABND
;
```

Exemples d'utilisation de POLYG :

Tracé d'un simple hexagone sur la hauteur de l'écran.

```
« (68,16) (18,18) (6,0)  CLLCD 9 CF POLYG »
```

Tracé d'un carré plein par remplissage des bords vers le centre puis effaçage du centre vers les bords. Ce carré est centré et horizontal. Sa hauteur est égale à celle de l'écran (32 pixels).

```
« (68,16) 4 π →NUM 4 / R→C DUP2 (22,0)
  SWAP CLLCD
  9 CF POLYG (0,22) SWAP
  9 SF POLYG
  9 CF CLMF
»
```

Puis même déroulement mais pour un pentagone.

```
« (68,15) (5,.28) DUP2 (18,0)
  SWAP CLLCD
  9 CF POLYG (0,18) SWAP
  9 SF POLYG
  9 CF CLMF
»
```

Cette fois, l'effaçage se fait des bords vers le centre.

```
« (68,15) (18,0) (5,.28) 3 DUPN CLLCD
  9 CF POLYG
  9 SF POLYG
  9 CF CLMF
»
```

Même déroulement mais pour un hexagone non totalement plein.

```
« (68,16) (18,6) (6,0) 3 DUPN CLLCD
  9 CF POLYG
  9 SF POLYG
```

```
  9 CF CLMF
»
```

Animation par rotation d'un pentagone inscrit dans un autre pentagone. Arrêt du programme par appui sur une touche autre que ATTN

ANIM
206 octets cksum : # F00Ch

```
« (68,15) DUP (18,18) (5,.28) CLLCD
  9 CF POLYG
  (13.6,13.6) (5,0)
  DO
    0 π →NUM 7 / R→C + 3 DUPN
    POLYG .25 WAIT 3 DUPN
    9 SF POLYG 9 CF
  UNTIL
    KEY
  END
  4 DROPN CLMF
»
```

Programme de démonstration TEST1 de Paul Jebeily.

Il suffit de faire la transformation suivante :

```
« « 4 DUPN DROP2 (4,0) POLYG 0 SWAP R→C + POLYG »
  { (68,15) (15,15) (6,0) } 2 π * 11 / 10 CLLCD
  FAM1ROT ROTIMG »
```

En supposant le flag 9 désarmé, la démo TEST1 (189 octets cksum : #AE07h) s'exécute rapidement et pour le plaisir des yeux.

Programme de démonstration TEST2 de Paul Jebeily.

Il suffit de faire la transformation suivante :

```
« « SPEED 4 DUPN 4 DUPN 4 ROLL 30 - 4 ROLLD
  SWAP DROP 2 SWAP 3 * NEG  R→C POLYG 4 ROLL
  30 + 4 ROLLD ROT C→R DROP 0 R→C 3 ROLLD SWAP
  DROP 4 SWAP 6 * 4 / NEG R→C POLYG R→C POLYG »
  { (68,15) (15,15) 6 } π 3 / 11 / 10 CLLCD
  FAM1ROT ROTIMG »
```

Ce programme TEST2 (275,5 octets cksum : # AEFBh) va aussi à merveille.

CERCLES, ANNEAUX, DISQUES sans peine

Produire des cercles ne se fait pas rapidement avec le langage utilisateur des HP28. Pour y remédier voici RING, un programme polyvalent permettant de générer et d'effacer des cercles, anneaux et disques avec les mêmes fonctionnalités que POLYG ci-dessus. Il peut remplacer les programmes CERCL, PCERCL et DCERCL de Paul.

C'est très rapide et même si rapide qu'il faudra parfois utiliser des temporisations pour certains types d'animations. L'algorithme utilisé est encore celui de Bresenham et la programmation est adaptée de celles des programmes CIRCLE et RING que j'avais écrits pour HP-48 (JPC 78).

```
*************************************************
*                  RING pour HP28S              *
*           321,5 octets  cksum # 5D77h         *
*                                               *
*    auteur Guy Toublanc          le 28/06/1994 *
*                                               *
* 1ère syntaxe :                                *
*      niveau 2 : R rayon du cercle             *
*      niveau 1 : ( Xc , Yc ) centre du cercle  *
*                                               *
* 2ème syntaxe :                                *
*      niveau 2 : ( R , r ) rayons de l'anneau  *
*      niveau 1 : ( Xc , Yc ) centre du cercle  *
*                                               *
* si R > r tracé de l'extérieur vers le centre  *
* si R < r tracé du centre vers l'extérieur     *
*                                               *
* Xc, Yc : réels quelconques                    *
* R , r  : réels positifs                       *
*                                               *
*  sytème de coordonnées absolues :             *
*                                               *
*        0  ------------------------> 136       *
*           |                                   *
*           |                                   *
*           |                                   *
*        31 V                                   *
*                                               *
*   9 CF : cercle ou anneau tracé               *
*   9 SF : cercle ou anneau effacé              *
*                                               *
* fichier source compilable sur PC par RPLCOMP de HP*
*************************************************
```

```
CODE
        gosbvl  =SAVPTR
        c=0     w           * pour initaialiser
        c=c-1   a           * c(a) := FFFFFh
        c=0     b           * c(a) := FFF00h
        d0=c                * @ speed
        dat0=c  xs          * pour speed maxi
        clrst               * intilialise st ....
        d0=(5)  #c0111      * @ flag 9
        a=dat0  1           * flags 12 à 9
        ?abit=0 0           * 9 FC?
        goyes   circleon    * oui alors on allumera
        st=1    3           * pour cercle à effacer
circleon
        c=dat1  a           * adresse complexe niv.1
        gosub   pop&conv    * Xc : conversion en hexa
        r3=c                * sauve Xc -> R3
```

```
        gosub   popim       * conversion Yc en hexa
        r0=c                * sauve Yc -> R0
        d0=d0+  5           * @ 2ième niveau
        c=dat0  a           * adresse R
        gosub   pop&conv    * R : conversion
        r2=c                * sauve Y = R -> R2
        ?st=0   0           * R seul ?
        goyes   Ronly
        gosub   popim       * conversion r
Ronly
        a=r2                * R
        c=a-c   a           * R - r
        gonc    R>r
        st=1    1           * pour R < r
        c=-c    a           * valeur absolue
R>r

*****************************************************
*                                                   *
*       Boucle de tracé des cercles successifs      *
*                                                   *
*  en entrée :                                      *
*           C(A) := nombre de cercle à tracer       *
*           A(A) := R2(A) := R courant              *
*           R0   := Yc                              *
*           R3   := Xc                              *
*           ST1 = 0  : R > r                        *
*           ST0 = 1  : R < r                        *
*           ST3 = 0  : pixels allumés               *
*           ST3 = 1  : pixels éteints               *
*****************************************************

gloop
        rstk=c              * sauve |R-r| et index
        r4=a                * sauve R  A(A) := Y = R
        a=a+a   a
        c=0     a
        r1=c                * départ X=0 sauve X
        lchex   3
        c=c-a   a           * d = 3 - 2*r
        d=c     a           * sauve d
        c=r1                * C(A):=X
        goto    start       * départ tracé

*****************************************************
*                                                   *
*         Boucle de tracé d'un cercle               *
*           algorithme de Bresenham                 *
*     avec un point calculé (X,Y), les symétries    *
*     permettent de tracer huit points              *
*                                                   *
*****************************************************

loop    gosub   plot        * tracé de 8 points
        p=      4
        c=r1                * X
        ?d#0    p
        goyes   dnegat
        cr2ex               * Y
        c=c-1   a           * Y - 1
        b=c     a           * sauve Y
        cr2ex               * sauve Y     C:=X
        c=c-b   a           * X := X - Y
dnegat  c=c+c   a
        c=c+c   a           * 4X
        p=      0
        d=d+c   a
        d=d+con a,6         * d =: d + 4X + 6
        c=r1                * X
        c=c+1   a           * X := X + 1
        r1=c                * sauve X
start   a=r2                * Y
        c=c-a   a           * X - Y
        goc     loop        * X < Y ? alors loop
        c=r1                * X
        ?c#a    a           * X <> Y ?
        goyes   next        * alors sortie
        gosub   plot        * encore tracé
next    c=rstk              * épaisseur anneau
        c=c-1   a           * restante
        goc     out         * nulle alors sortie
        a=r4                * R
        a=a-1   a           * pour le cercle suivant
        ?st=0   1           * R > r
        goyes   R>>r        * oui alors c'est bon
        a=a+con a,2         * sinon incrémentation
R>>r
        r2=a                * sauve Y := R
        goto    gloop
out     govlng  =GETPTRLOOP

negyy   c=r1                * ces divers segments
        c=-c    a           * de programme
        r1=c                * effectuent les
        goto    negy        * permutations entre X
xyexx   c=r1                * et Y et changements
        c=-c    a           * de signe pour obtenir
        r1=c                * par symétrie 8 points
        goto    xyex

plot    st=0    0           * marque 1er tour
loop2   gosub   plotxy      *
        goc     negyy
        gosub   lxx
negy    c=r2                * Y
        c=-c    a           * Y := Y
        r2=c                * sauve Y
        gosub   plotxy2     *
        goc     xyexx
        gosub   lxx
xyex    c=r1                * X
        cr2ex               * Y := X
        c=-c    a           * X := -Y
        r1=c                * sauve X
        ?st=1   0           * 2e tour ?
        rtnyes              * alors retour
```

```
        st=1    0       * marque 2e tour
        gonc    loop2
lxx     c=r1            * encore symétrie
        c=-c    a
        r1=c
plotxy
        c=r2            * Y
plotxy2 a=r0            * Yc
        a=a-c   a       * Yi
        lc(5)   31
        ?c<a    a       * Yi > 31 ?
        rtnyes          * oui on ne fait rien
        d1=a            * sauve Yi
        a=r1            * X
        c=r3            * Xc
        a=a+c   a       * Xi
        lc(5)   137     * ex 136
        ?a<c    a       * Xi < 137 ?
        goyes   Xok
        rtncc
Xok     lc(2)   67
        ?c>=a   b       * Xi <= 67 ?
        goyes   left    * oui demi écran gauche
        a=a-c   a       * Xi pour 1/2 écr. droit
left
        lchex   ffbf8   * adresse 1/2 écr. droit
        gonc    right   * si Xi > 67 saut écr-dr
        lchex   840     * FF840 pour 1/2 écr-g
right
        a=a+a   a
        a=a+a   a
        a=a+a   a       * colonne pour Xi
        a=a+c   a       * adresse colonne
        ad1ex           * @ colonne  a(A) =: Yi
        d0=a            * sauve Yi
        asrb
        asrb            * Yi en quartets
        cd1ex           * adresse colonne
        c=c+a   a       * adresse quartet pour Yi
        cd1ex           * @ quartet Yi
        ad0ex           * Yi
        lchex   3
        a=a&c   p       * Yi modulo 4
        lchex   1       * récupère 1 de 1F (31d)
        gonc    decr    * départ pour position

loop4   c=c+c   p       * position pixel dans nib
decr    a=a-1   p       * compteur
        gonc    loop4

        a=dat1  b       * quartet du pixel en mem
        ?st=0   3       * PIXON ?
        goyes   pixon
        a=-a-1  p       * inversion pour PIXOFF
pixon
        a=a!c   p       * OR pour PIXON ou PIXOFF
        goc     pixon2  * saut pour PIXON
        a=-a-1  p       * inversion pour PIXOFF

pixon2
        dat1=a  1       * quartet modifié
        rtncc

pop&conv cd1ex          * @ prologue
        d0=c            * sauve D1
        a=dat1  a
        a=a+a   a
        a=a+a   b
        st=0    0       * pour réel
        gonc    real
        st=1    0       * pour complexe
real
        d1=d1+  4

popim   d1=d1+  1       * @ exposant
        a=0     a
        a=dat1  x       * exposant
        lchex   500
        ?c>a    x       * exposant positif ?
        goyes   exppos
        b=0     a       * 0,.... arrondi à 0
        d1=d1+  15      * pour partie imag.
        gonc    convrtn
exppos
        lchex   003     * si exposant > 3
        ?c>=a   x       * alors exposant = 3
        goyes   exp<4
        a=c     x
exp<4
        d1=d1+  14      * @ digit de poids fort
        ad1ex           * a(A) =: D1
        cd1ex           * C(A) =: exposant
        p=c     0       * P =: exposant
        c=a-c   a       * adresse digit de poids
        d1=c            * faible (mantisse)
        a=0     a
        a=dat1  wp      * mantisse
        b=a     a       * sauvegarde
        d=0     a       * initialisation
        c+p+1           * adresse après mantisse
        d1=c            * @ le signe
        p=      0
        c=0     a
        lchex   6       * 3e cste de conversion
        rstk=c          * sauvegarde
        lchex   9c      * 2e cste de conversion
        rstk=c          * sauvegarde
        lchex   c18     * 1e cste de conversion
        p=      3       * compteur pour loop5

loop5   a=b     p       * et pointeur de digits
        gonc    decr3   * départ conversion

loop3   d=d+c   a
decr3   a=a-1   p
        gonc    loop3
```

```
        p=p-1           * position suivante
        ?p=     0       * plus de digits ?
        goyes   cont    * alors sortie de loop5
        c=rstk          * dépile constante suiv.
        gonc    loop5   * tour suivant de boucle

cont    c=d     a
        b=b-c   a       * résultat en hexa
        a=dat1  1       * signe
        ?a=0    p       * positif ?
        goyes   pos
        b=-b    a       * si non entier signé
pos
convrtn c=b     a       * nombre converti
        rtn
ENDCODE
```

A titre d'exemple, voici deux petites démos montrant les possibilités de RING.

DEMO1 trace puis efface deux familles de cercles symétriques. Le sous-programme contenu dans DEMO1 est évalué une première fois pour le tracé des cercles avec le flag 9 levé, puis est évalué une seconde fois avec le flag 9 baissé pour l'effaçage de ce qui a été tracé précédemment. C'est très rapide et cette démo pourrait être améliorée avec des temporisations. Il suffit d'activer DEMO1 pour voir cette animation.

DEMO1
130,5 octets cksum : # 730Bh

```
« CLLCD
  « -68 68
    FOR i
      i ABS i 0 R→C (68,15) +
      RING DROP2 3
    STEP
  »
  9 CF DUP EVAL
  9 SF EVAL
  9 CF CLMF
»
```

DEMO2 est similaire mais trace et efface des anneaux au lieu de cercles.

DEMO2
145,5 octets cksum : # A2CDh

```
« CLLCD
  « -68 68
    FOR i
      i ABS DUP 2 - 0 MAX R→C
      i 0 R→C (68,15) +
      RING DROP2 4
    STEP
  »
  9 CF DUP EVAL
  9 SF EVAL
  9 CF CLMF
»
```

Vous trouverez les listes de codes des programmes dans *le coin des codes*.

J'espère que quelques utilisateurs de HP-28 sauront nous faire profiter de leurs créations en matière de graphisme et d'animation à l'instar de Paul Jebeily qui aura peut-être avancé ses projets dans ce domaine.

Remarques importantes :

On aura toujours à l'esprit le système de coordonnées avec lequel fonctionnent toutes ces fonctions. La mise à l'échelle, éventuellement, sera à gérer par vous. Tous les programmes ci-dessus fonctionnent sans rapport avec PPAR. Il faudra aussi se rappeler le rôle important du flag 9 (commun à LINE POLYG et RING). On vérifiera avec soin le nombre et le type des arguments fournis,car dans un souci d'allègement de programmation ces vérifications n'ont pas toujours été incluses dans les programmes ou seulement partiellement. Vous aurez été prévenus.

Guy Toublanc (276)

Annexes

Avec les programmes ci-dessus je pense que certains des programmes de Paul Jebeily (JPC 90) seraient à modifier pour bénéficier de la plus grande rapidité des nouveaux venus. Cela ne peut se faire que par quelqu'un qui en a bien saisi le mécanisme. Pour ma part je me suis contenté de traduire bêtement, en en mixant System Rpl et User Rpl, quelques-uns des programmes pour réduire l'encombrement mémoire. Voici à titre d'exemple le programme ROTIMG qui regroupe ROTIMG, LPGROT et LTCHROT, faisant gagner environ 500 octets et légèrement de rapidité.

```
************************************************
*                   ROTIMG                     *
*            638 octets   cksum  #FA36h        *
*  remplace ROTIMG LPGROT LTCHROT avec OFF intégré *
*  pour ceux-ci voir l'article de Paul Jebeily dans*
*  JPC 90 pages 7 et 8                        *
************************************************

ASSEMBLE
=OFF equ #18e58
RPL
::
 xDUP xSIZE DUP %0=
 IT :: 2DROP xABORT ;
```

```
BEGIN
 2DUP ID PWAIT xSIGN DUP xNOT %+ %*
 '
 :: DO>LCD ID PWAIT %ABS xWAIT ;
 ID LSTX xKEY %0<>
 IT
  ::
   { "0" $_1 $_2 $_3 $_4 $_5 $_6 $_7 $_8 $_9
     $_. $_= $_+ $_- $_* $_/ $_CHS "S" "Q" "O" }
   SWAP xPOS %1 %+
   ' ID PWAIT ' ID STP.W
   { :: 2DROP 2.20000000000E002 1.00000000000E-01
  2DUP DOBEEP 2DUP SWAP DUP %+ SWAP DOBEEP DOBEEP ;
     :: 1.00000000000E-03 UNROTSWAP DROP xSTO ;
     :: 1.00000000000E-01 UNROTSWAP DROP xSTO ;
     :: 2.00000000000E-01 UNROTSWAP DROP xSTO ;
     :: 3.00000000000E-01 UNROTSWAP DROP xSTO ;
     :: 4.00000000000E-01 UNROTSWAP DROP xSTO ;
     :: 5.00000000000E-01 UNROTSWAP DROP xSTO ;
     :: 6.00000000000E-01 UNROTSWAP DROP xSTO ;
     :: 7.00000000000E-01 UNROTSWAP DROP xSTO ;
     :: 8.00000000000E-01 UNROTSWAP DROP xSTO ;
     :: 9.00000000000E-01 UNROTSWAP DROP xSTO ;
     :: 1.00000000000E001 UNROTSWAP DROP xSTO/ ;
     :: xRCL SWAP xSTO ;
     :: xRCL SWAP xSTO+ ;
     :: xRCL SWAP xSTO- ;
     :: xRCL SWAP xSTO* ;
     :: xRCL SWAP xSTO/ ;
     :: DROP DUP xRCL xNEG SWAP xSTO ;
     :: 2DROP ID SPEED ;
     :: xCLEAR xABORT ;
     :: 2DROP OFF ; }
    4ROLL xGET xEVAL
 ;
 FALSE
 UNTIL
;
```

HP48

P. Basset
J.F. Garnier
G. Toublanc

Diagramme de Bode *22*
Précision étendue *22*
Réduction de programmes (Acte II) *25*

Le coin des codes *34*

DIAGRAME DE BODE

Je vous envoie un petit programme qui m'a été très utile dans mes études (je sors d'un IUT en électronique). Il s'agit de tracer sur la HP48 le diagramme de Bode en gain et en phase d'une fonction complexe.

Tout d'abord, il faut préciser qu'il s'agit d'un programme "de débutant pour débutant" et que s'il il n'intéresse pas les amateurs chevronnés des HP, il satisfera peut-être ceux qui, comme moi, sont incapables de comprendre les pages qui suivent l'éditorial de JPC.

Il s'agit en fait de deux petits programmes complémentaires dont la structure est inspirée de certains listings du livre de Jean-Michel Ferrard *Programmation et applications de la HP48.*

Le premier d'entre eux, que j'ai appelé BODE, demande lasaisie d'une fonction sous la forme d'un quotient de polynômes de Laplace, la pulsation de départ, celle d'arrivée, et le pas désiré. Les points des courbes sont ensuite calculés puis stockés dans deux matrices : NORM pour la courbe des gains et ARGU pour la courbe des phases.

Le deuxième programme, appelé COURBE, trace une courbe à partir des points stockés dans la matrice statistique en cours. Il reste donc à faire un menu qui stocke la matrice ARGU ou la matrice NORM dans la matrice statistique avant d'appeler le programme COURBE.

BODE

```
«
  " Saisir: 'equation en P' début fin pas "
  "" INPUT OBJ→
  CLΣ 0 → a b c d e
  «
    0 0 1 2 2 →LIST →ARRY
    DUP 'norm' STO
    'argu' STO b →NUM c →NUM
    FOR i
      IFERR
        0 1 R→C 'e' STO
        i a EVAL DUP2
        ABS LOG 20 *
        2 →ARRY 'NORM' STOΣ Σ+
        ARG 2 →ARRY 'ARGU' STOΣ Σ+
      THEN
        CLEAR
      END
    d STEP
  »
»
```

COURBE

```
«
  ERASE SCARPLOT DRAX
  'Σdat' EVAL OBJ→ DROP
  'Σdat' EVAL SIZE LIST DROP
  0 → a b c
  «
    b a
    FOR i
      R→C DUP
      IF c 0 <> THEN
        c SWAP LINE
        ELSE PIXON
      END
      'c' STO
    1 STEP
  »
»
```

Philippe Basset (582)

PRECISION ETENDUE POUR HP28/48

Il existe toute une classe de calculs où la précision est primordiale : il s'agit des calculs itératifs pour lesquels les erreurs d'arrondi se cumulent progressivement.

Considérez l'itération suivante : prenez le nombre 1.0000001 (1 + 1e-7), et élevez le 27 fois au carré :

IT1

```
«
  1.0000001
  1 27 START
    SQ
  NEXT
»
```

Les HP28 et 48 donnent comme résultat 674514.86877 alors que la valeur exacte avec 10 chiffres significatifs est 674530.4707 comme l'indique le calcul direct de 1.0000001 ^ (2 ^ 27).

(Cet exemple est tiré de l'excellent ouvrage *Récréations informatiques* dans la série *Pour La Science* malheureusement plus disponible. On le trouve cependant encore dans les bibliothèques, je vous encourage à le consulter si vous en avez la possibilité).

Je vous propose une façon simple d'augmenter la précision de votre machine. Il s'agit d'utiliser le type interne Long Real, et d'appeler les routines de manipulation par l'intermédiaire de SYSEVAL.

Le type Real Utilisateur est constitué d'une mantisse de 12 chiffres, et d'un exposant variant de -499 à +499.

Par contre le type Long Real voit sa mantisse passer à 15 chiffres, tandis que l'exposant s'étend de -49999 à +49999, bien au delà des besoins usuels.

Les Long Real sont affichés en tant que 'System Object' sur la HP28S, et comme 'Long Real' sur la HP48, il faut donc disposer de routines de conversions pour l'introduction et l'affichage.

Avec les routines suivantes:

%>%% : conversion d'un Real en Long Real
%%>% : conversion d'un Long Real en Real
%%* : multiplication de 2 Long Real

Les versions pour HP28 et HP48 de IT1 en précision étendue s'écrivent ainsi (attention aux adresses des SYSEVAL !!) :

IT2 POUR HP28 !!

```
«
  1.0000001
  # 1129Bh SYSEVAL      @ %>%%
  1 27 START
    DUP
    # 11D2Dh SYSEVAL    @ %%*
  NEXT
  # 11280h SYSEVAL      @ %%>%
»
```

IT2 POUR HP48 !!

```
«
  1.0000001
  # 2A5C1h SYSEVAL      @ %>%%
  1 27 START
    DUP
    # 2A99Ah SYSEVAL    @ %%*
  NEXT
  # 2A5B0h SYSEVAL      @ %%>%
»
```

Cette fois, les HP28 et HP48 fournissent 674530.4678 avec 7 chiffres exacts, au lieu de 4 précédemment.

Vous trouverez peut être que cet exemple n'est pas convaincant, à cause du choix très particulier de la valeur de départ (1.0000001), et du fait que l'on peut calculer directement le résultat. Cependant, un tel raccourci n'existe généralement pas. Par exemple, le résultat de n itérations de $x=abs(ln(x))$ ne peut être obtenu qu'en calculant effectivement tous les termes.

Comparez le résultat du programme pour HP28/48 :

IT3

```
«
  0.5
  1 40 START
    LN ABS
  NEXT
»
```

Avec les résultats obtenus sur d'autres machines:

HP41C : 0.387230709
HP75/71/28/48 : 0.768875183625
PC Turbo Pascal : 0.829369170184
PC Quick-Basic : 0.829369170184

(note: les résultats sur PC ont été obtenus en double précision, avec coprocesseur).

Quel est le résultat exact ?

Le fait que plusieurs machines obtiennent le même résultat n'indique pas qu'il s'agit du bon, mais simplement qu'elles utilisent les mêmes algorithmes et les mêmes représentations internes des nombres !

La comparaison avec une version en précision étendue permet de se rendre compte de ce qui se passe :

IT4 POUR HP28 !!!

```
«
  0.5
  # 1129Bh SYSEVAL      @ %>%%
  1 40 START
    # 11EB5h SYSEVAL    @ %%LN
    # 11C74h SYSEVAL    @ %%ABS
  NEXT
  # 11280h SYSEVAL      @ %%>%
»
```

IT4 POUR HP48 !!!

```
«
  0.5
  # 2A5C1h SYSEVAL      @ %>%%
  1 40 START
    # 2AB5Bh SYSEVAL    @ %%LN
    # 2A8F0h SYSEVAL    @ %%ABS
  NEXT
  # 2A5B0h SYSEVAL      @ %%>%
»
```

Ces versions fournissent le résultat 0.829372768402 en bon accord avec les résultats sur PC, ce qui nous met sur la piste. Voici le résultat du déroulement de l'itération $x=abs(ln(x))$ sur HP28/48 :

précision standard	précision étendue
0.69314718056	0.69314718056
0.366512920582	0.366512920582
1.0037215043	1.0037215043
0.0037145966355	0.00371459663781
5.59548518396	5.59548518334
1.72196005542	1.72196005531
0.543463209116	0.543463209054
0.609793267231	0.609793267345
0.494635285435	0.494635285248
0.703934585083	0.703934585461
0.351069846058	0.351069845521
1.04677008369	1.04677008522
0.0457093124443	0.0457093139106
3.08545322844	3.08545319636
1.12669856016	1.12669854977
0.11929172826	0.119291719035
2.12618328791	2.12618336524
0.754328488767	0.754328525138
0.281927344352	0.281927296136
1.26610588538	1.26610605641
0.235945957965	0.235946093045
1.44415249185	1.44415191935
0.367522638669	0.367522242241
1.00097036041	1.00097143905
0.000969889914679	0.000970967513409
6.93832798293	6.93721754707
1.93706082084	1.93690076431
0.661171783436	0.66108915147
0.413741588725	0.413866574595
0.882513681863	0.882211640689
0.124980986767	0.12532329633
2.07959365911	2.07685850994
0.732172518432	0.730856420254
0.311739111871	0.313538254046
1.16558862087	1.15983390388
0.153226213386	0.148276808554
1.87583993066	1.90867442404
0.629056522136	0.646408982288
0.463534166007	0.436322876005
0.768875183625	0.829372768402

On constate que l'erreur atteint 1% vers la 37ème itération, pour s'amplifier ensuite rapidement. Au dela de la 40ème itération, les deux résultats n'ont plus rien à voir, et sont aussi faux l'un que l'autre. De plus, alors que les erreurs d'arrondis de l'exemple d'introduction augmentaient de façon géométrique (elles doublent à chaque itération), elles peuvent ici augmenter catastrophiquement en une itération. Par exemple, à l'itération 25, on passe de 1.00097036041 (erreur relative 1E-6) à 9.69889914679E-4 (erreur relative 1E-3 !!).

L'interêt de l'utilisation de la précision étendue reside dans la possibilité de vérifier expérimentalement la limite de validité d'un calcul itératif. Ce type d'itération intervient dans les études des fractales comme l'ensemble de Mandelbrot, ou encore les simulations de comportements chaotiques.

Cet article aura atteint son but s'il vous a convaincu que l'on peut, à peu de frais, étendre les possibilités de nos chères machines, sans pour cela recourir à une programmation complexe. Vous trouverez ci-dessous une liste de SYSEVAL de manipulation des Long Real pour HP48S/G toutes versions et pour HP28S.

Pour terminer, je rappelle aux utilisateurs de HP28S s'intéressant à la programmation par SYSEVAL ou en System RPL que je mets regulièrement à jour une liste de points d'entrées compatibles 48. Depuis sa publication dans JPC 87-88, le nombre d'adresses est passé de 500 à plus de 670. Vous pouvez vous la procurer au club, ou directement en m'envoyant une disquette PC formattée (avec l'enveloppe de retour) à :

Jean-François GARNIER
47 allée de la Dent de Crolles
38420 DOMENE

MNEMONIQUE	HP48	HP28S	COMMENTAIRE
Constantes Long Real			
%%.1	#2A562	#11562	

%%.4	#2B3DD	#1290F	
%%.5	#2A57C	#1157C	
%%0	#2A4C6	#114C6	
%%1	#2A4E0	#114E0	
%%2	#2A4FA	#114FA	
%%3	#2A514	#11514	
%%4	#2A52E	#1152E	
%%5	#2A548	#11548	
%%7	#2B1FF	#12731	
%%10	#2A596		sur 28S ??
%%12	#2B2DC	#1280E	
%%60	#2B300	#12832	
%%2PI	#0F688		(48 seulement)
%%PI		#11458	(28S seulement)
PI/180	#2A62C	#1190A	

Conversions

%%>%	#2A5B0	#11280	%%a --> %a
%>%%	#2A5C1	#1129B	%a --> %%a
C%>%%	#519CB	#30248	(a,b) -> %%a %%b
%%>C%	#51A07	#3027A	%%a %%b --> (a,b)

Fonctions d'un Long Real

%%MAX	#2A6DC	#119DE	
%%ABS	#2A8F0	#11C74	
%%CHS	#2A910	#11C94	
%%1/	#2AA92		sur 28S ??
%%SQRT	#2AAEA	#11E43	
%%EXP	#2AB1C	#11E74	
%%LN	#2AB5B	#11EB5	
%%LNP1	#2AB94	#11EF0	
%%SIN	#2AC06	#11F4E	avec mode courant
%%SINDEG	#2AC17	#11F77	argument degrés
%%COS	#2AC57	#11F9D	avec mode courant
%%COSDEG	#2AC68	#11FC6	argument degrés
%%COSRAD	#2AC78	#11FD9	argument radians
%%TANRAD	#2ACA8		argument radians
%%ASINRAD	#2ACD8	#12020	résultat radians
%%ACOSRAD	#2AD08	#1204D	résultat radians
%%SINH	#2AD95	#120E8	
%%COSH	#2ADC7	#12111	
%%FLOOR	#2AF99	#124C5	

Fonctions de deux Long Real

%%+	#2A943	#11CCB	
%%-	#2A94F	#11D01	
%%*	#2A99A	#11D2D	
%%/	#2A9E8	#11D73	
%%^	#2AA5F	#11DDF	
%%ANGLE	#2AD4F	#1208D	a b --> atan(b/a)
%%ANGLEDEG	#2AD6C	#120C2	résultat degrés
%%ANGLERAD	#2AD7C	#120D5	résultat radians

Jean-François Garnier (242)

REDUCTIONS TROP RADICALES

Si je m'étais contenté d'une simple compilation de SYSEVALs et d'une élimination des délimiteurs de programme pour des programmes écrits en User Rpl (JPC 96: allègements en douceur), notre ami Lu Vy Huynh est allé plus loin avec son programme STRIP (JPC 97 page 11) en éliminant d'autres instructions inutiles. Son programme est très intéressant et j'ai été tenté de le mettre à l'épreuve. S'il fonctionne très bien en réduisant certains programmes, il est mis en échec dans le cas ou le programme à réduire contient des séquences consécutives de tests comme par exemple:

```
« DUP
  IF 5 > THEN 1 - END
  IF DUP 3 > THEN .5 - END
»
```

Ce programme donne pour :

```
10  → 8.5
 3.5 → 3
 2   → 2
```

et compacté avec STRIP les résultats sont :

```
10  → 10
 3.5 →  - Error: Too Few Arguments
 2   →                    (rien n'est renvoyé)
```

On voit donc que si le premier test est positif, le résultat est correct, sinon cela plante dans tous les autres cas.

Le programme ASSCOD48, que vous trouvez dans la rubrique *le coin des codes*, après avoir été traité par STRIP, plante aussi. Il est possible que là, la structure CASE soit en cause, car traitée par STRIP de la même manière que IF THEN END.

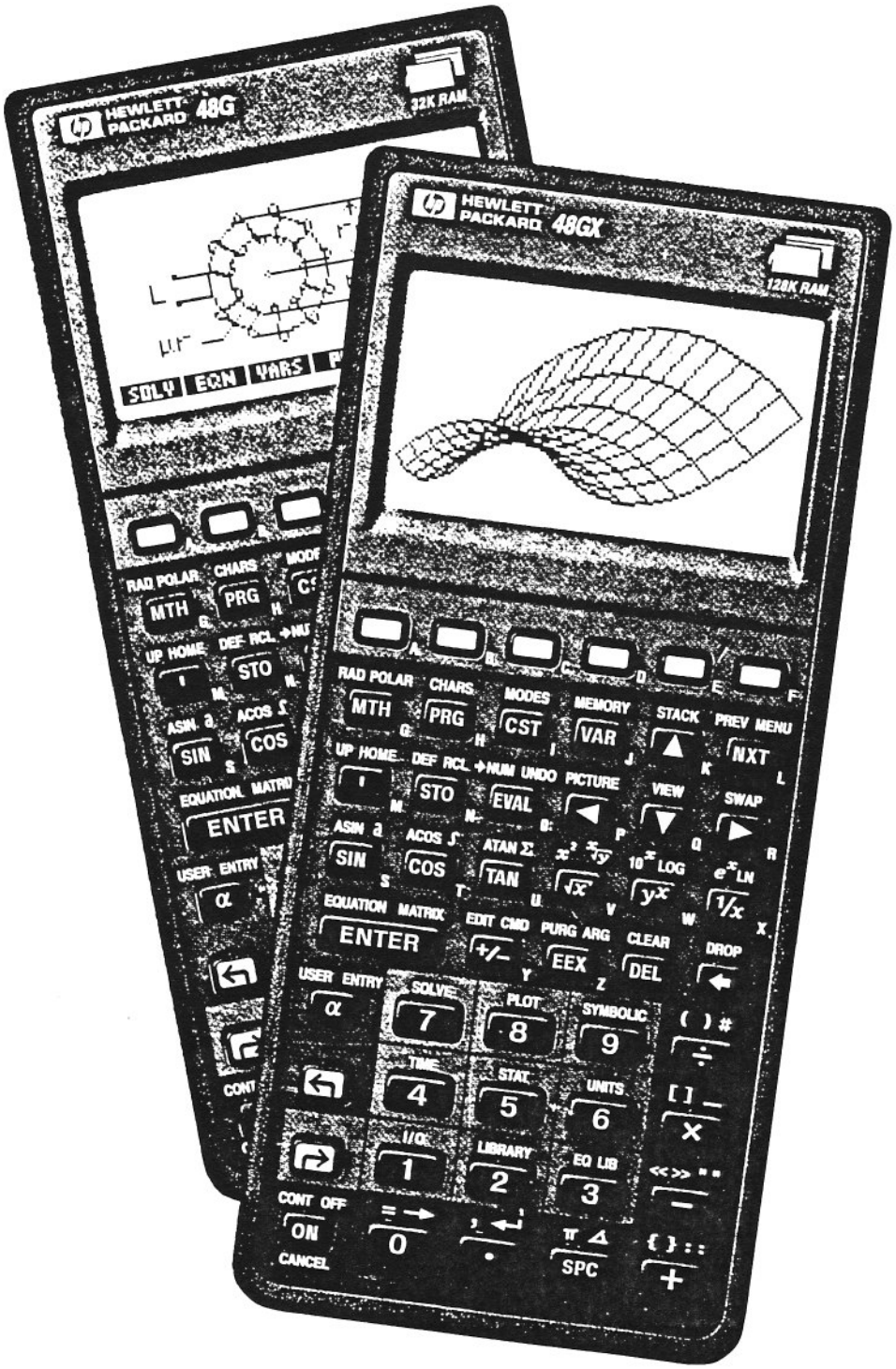
Un programme avec la structure suivante :

```
DO ..... IFERR ..... THEN .... END .... UNTIL
```

ne fonctionne pas non plus après compactage.

Je ne suis pas allé plus loin dans cette mise à l'épreuve et je laisse notre ami faire les retouches à son programme qui est par ailleurs très intéressant du point de vue programmation.

Guy Toublanc (276)


HEWLETT PACKARD 48G
32K RAM
SOLV EQN VARS
RAD POLAR CHARS MODES MEMORY STACK PREV MENU
MTH PRG CST VAR NXT
UP HOME DEF RCL →NUM UNDO PICTURE VIEW SWAP
STO EVAL
ASIN ACOS ATAN
SIN COS TAN
EQUATION MATRIX EDIT CMD PURG ARG CLEAR DROP
ENTER +/- EEX DEL
USER ENTRY SOLVE PLOT SYMBOLIC
TIME STAT UNITS
I/O LIBRARY EQ LIB
CONT OFF ON CANCEL
SPC
HEWLETT PACKARD 48GX
128K RAM

HP95 / HP100 / HP200

J. Belin	Le HP200LX	*28*
G. Toublanc	Calcul des moments d'inertie	*30*

LE HP200LX

Comme je vous le laissais entendre dans le dernier JPC, Hewlett-Packard à décidé de sortir de ses cartons un nouveau palmtop, destiné à *remplacer* le HP100LX.

Cette présentation est axée principalement sur les différences existant entre les deux machine.

Notez cependant que la plupart des peronnes directement concernées par la sortie de la machine (responsable des ventes, attaché de presse...) ayant eu la mauvaise idée de prendre leurs vacances au moment même où elle était annoncée, il ne m'a pas été possible d'en avoir un exemplaire en main, ni d'obtenir des photos exploitables. Cet article est donc basé sur des renseignements obtenue par différentes sources.

Tout d'abord, après avoir choisi des noms de grands félins pour codes de dévelopement des HP95 et HP100 (*Jaguar* et *Cougar*, respectivement), Corvallis a légèrement changé de politique, en choisissant des personnages félins de dessins animés ou de bandes dessinées. En effet, le HP200 1Mo à reçu le nom de *Felix* et la version 2Mo, *Garfield*. Le Connectivity Pack à été baptisé *Toonce*, nom d'un chat apparaissant dans l'émission satirique de la télévision américaine *Saturday Night Live*.

Aspect Général

Si il possède le même boitier que celui des précédents palmtops HP, celui du HP200 se distingue au premier coup d'oeil par un boitier gris foncé légèrement bleuté et des changements dans le couleur de certaines touches. En fait, il s'agit des mêmes couleurs que pour la HP48GX.

La touche `[Fnt]` (ainsi que les fonctions correspondantes) est maintenant mauve (ou violette, ou magenta suivant les utilisateurs...) légèrement plus claire que pour la HP48GX. Les touches d'applications (autrefois bleues) sont devenues vertes, et présentent une nouvelle organisation :

```
FILER  APPT  PHONE MEMO  QUICKEN LOTUS  CALC  &...
```

au lieu de :

```
FILER ccMAIL APPT  PHONE  MEMO   LOTUS  CALC  &...
```

Le marquage des touches est légèrement incliné et déplacé vers le haut à gauche. Probablement afin de donner l'impression que le clavier est plus grand.

L'écran est du même type que celui des derniers HP100, plus contrasté.

Le Hardware

Le Hardware est strictement le même que pour le HP100 (microprocesseur 80186 à 7.91 MHz, écran CGA, 1 connecteur PCMCIA version 2.0...). La seule différence est que la Rom est passée de 2 Mo à 3 Mo.

Le software

Mis à par deux nouvelles applications, et quelques utilitaires, peu de modifications majeures sont à noter. La version de MS-Dos est toujours la 5.00.

Le System Manager

A l'allumage du palmtop, un nouvel écran *Topcard* apparait. Il semble ne pas faire l'unanimité...

Les icônes ont été aussi modifiées.

La plupart des fichiers macros semblent fonctionner, à l'exception des certaines faisant appel à Lotus.

cc:Mail et Communications

Contrairement à ce que nous disions dans le précédent numéro, cc:Mail est toujours présent. Il n'y a pas d'améliorations notable, mais un nouvel utilitaire (`"cc:MADE"`) permetant des échanges automatiques des fichiers entre répertoires.

La vitesse d'affichage en mode émulation de terminal est enfin acceptable...

Agenda

Chaque jour, au premier accès à l'agenda, un message de salutation est affiché, incluant les rendez vous et les choses à faire pour la journée.

Répertoire téléphonique

La principale innovation de cette application est l'apparition (optionnelle) d'un nouvel écran, présentant sur le coté gauche la liste des noms, alors que le côté droit présents une fiche mentionnant les coordonnées de la personne sélectionnée.

Il est à noter que les coordonées du `Palmtop Paper` sont maintenant mentionnées dans le fichier original !

Le Bloc Notes

Peu de choses à noter, si ce n'est que les épingles apparaissant sur l'écran graphique ont disparu.

Bases de données

Peu de modifications, semble-t-il. Y compris la vitesse qui est toujours aussi lente.

Nouvelles applications

Quicken

Quicken est un logiciel de gestion financière personnelle très en vogue chez les Anglo-Saxons. La version incluse dans le HP100 (*Pocket Quicken*) est bien sûr limitée par rapport à la version Dos ou Windows, mais possède les fonctionnalités suivantes :

- Entrée et édition des différents gains et dépenses, que ce soit par chèques, cartes de crédit, espèces, virements, pour différents comptes.
- Visualisation des balances.
- Création et impression de rapports.
- Echange des données avec les versions PC.

Laplink Remote

Le redirecteur n'est plus dans le menu de configuration. A la place, il y a Laplink Remote, un logiciel d'échange de fichiers bien connu des utilisateurs PC, qui fonctionne beaucoup mieux. Il y a bien sûr la possibilité de transférer une copie de Laplink sur un autre PC.

Nouveaux fichiers et utilitaires

Bien que la commande interne 100, soit conservée, un nouveau fichier 200.COM a été ajouté. Ceci est intéressant pour ceux qui utilisent des programmes tels que 4DOS.

Des versions adaptées au HP200 de HEXCALC et de l'éditeur d'icônes (présents dans notre disquette n° 5) ont aussi été incluses.

Deux jeux ont été ajoutés dans la Rom : *Hearts & Bones*, la version "pacifiée" du jeu du démineur (présent dans la version US du HP95), et *Lair of Squid*.

Des documentations supplémentaires, donnant quelques "trucs" d'utilisation, ont été incluses, au format du Bloc-Note. Vous trouverez aussi quelques informations intéressantes dans le fichier D:_DAT\APPTS.INI.

Parmi celles-ci vous pourrez noter le retour du "password" global, tel qu'il apparaissait sur le HP95.

Le Connectivity pack

Une nouvelle version de ce programme (référencé 1021B) est disponible. Il inclut bien sûr les routines de conversion de Quicken vers les versions desktop du proramme (le format des fichiers ayant été modifié afin d'économiser l'espace disque du HP200).

Il sera disponible à la fin du mois d'août.

Compatibilité avec d'autres programmes

Tous les programmes (System Manager ou non) tournant déjà sur le HP100 devraient fonctionner sur le HP200. Cependant, sur la version US, environ 50% des fonctions de 100Buddy semblent fonctionner correctement. Une nouvelle version de ce programme devrait sortir très prochainement.

Pour finir, je tiens à préciser que toute cette présentation concerne la version US du HP100. D'après Hewlett-Packard, il ne devrait pas y avoir de disparition de fichiers sur la version Française (comme cela s'était fait pour les HP95 et HP100). Mais la version Française n'étant pas disponible au moment ou j'écris ces lignes, je ne peux pas vous en dire plus.

Malgré le fait qu'il y ait de nouvelles applications, le prix des deux machines à baissé par rapport à ceux des HP100. La version 1 Mo coûte maintenant 3900 F HT et la 2 Mo 4730 F HT (prix catalogue HP).

En résumé, avec cette machine, Hewlett-Packard semble avoir répondu aux attentes des utilisateurs (en tous cas les Américains). Si il n'est peut être pas indispensable à un utilisateur de HP100, nous ne pouvons que le conseiller à tous les autres, puisque (même sans compter les nouvelles applications) il reste toujours sans concurence. Cependant, la baisse de prix de la version 2 Mo incitera peut-être les possesseurs de la version 1 Mo à sauter le pas !

Cependant, on peut s'interroger sur certaines incohérences (voulues par les services commerciaux ?) consistant à appeler HP200LX une machine ayant tout de même peu de différences fondementales par rapport au HP100LX, alors qu'il y a un an sortait un autre produit répondant à l'appelation HP48GX, qui possédait beaucoup plus de différences par rapport à la HP48SX. Peut être que *HP100QX* aurait été mieux adapté...

Jacques Belin (123)

CARACTERISTIQUES MECANIQUES DE SECTIONS

Venant juste d'acquérir un HP100 et n'ayant jamais utilisé un tableur, je me suis jeté à l'eau avec une feuille de calculs pour déterminer le moment d'inertie de sections planes ainsi que des caractéristiques s'y rapportant. Il s'agit essentiellement de sections décomposables en trapèzes ainsi que toutes figures géométriques en dérivant: triangles, parallélogrammes généraux ou particuliers (carrés, rectangles et losanges).

Voici comment se présente cette feuille pour la partie données et résultats, c'est-à-dire pour le champ A1 à F13. Pour des raisons de mise en page la colonne des résultats est rétrécie, ici, de deux caractères.

```
   caractéristiques de sections à base de trapèzes
==================================================
niv.|base sup|base inf| hauteur |       Résultats
--------------------------------------------------
  1 |   0,000    0,000    0,000 | I    =      0,0000
  2 |   0,000    0,000    0,000 | S    =      0,0000
  3 |   0,000    0,000    0,000 | Vs   =      0,0000
  4 |   0,000    0,000    0,000 | I/Vs =      0,0000
  5 |   0,000    0,000    0,000 | Vi   =      0,0000
  6 |   0,000    0,000    0,000 | I/Vi =      0,0000
  7 |   0,000    0,000    0,000 | rho  =      0,0000
  8 |   0,000    0,000    0,000 |====================
  9 |   0,000    0,000    0,000 |***** Init : /I *****
```

Le champ G1..K13 est occupé par les cellules des calculs intermédiaires. En E15 se trouve 0,000 servant à réinitialiser le champ des données. Enfin en F15 est logée une macro /re15~b5.d13~/cdb5.d~ de réinitialisation des données.

Exemple d'utilisation :

Section (imaginaire) composée de haut en bas de :

```
1 trapèze  : base supérieure 35
             base inférieure 30
             hauteur         25

1 trapèze  : base supérieure 20
             base inférieure 15
             hauteur         10

1 triangle : (base supérieure 0)
             base inférieure 30
             hauteur         35
```

```
   caractéristiques de sections à base de trapèzes
==================================================
niv.|base sup|base inf| hauteur |       Résultats
--------------------------------------------------
  1 |  35,000   30,000   25,000 | I    = 758823,3184
  2 |  20,000   15,000   10,000 | S    =   1512,5000
  3 |   0,000   30,000   35,000 | Vs   =     30,2342
  4 |   0,000    0,000    0,000 | I/Vs =  25098,2109
  5 |   0,000    0,000    0,000 | Vi   =     39,7658
  6 |   0,000    0,000    0,000 | I/Vi =  19082,2906
  7 |   0,000    0,000    0,000 | rho  =     22,3987
  8 |   0,000    0,000    0,000 |====================
  9 |   0,000    0,000    0,000 |**** Init : /I ****
```

Les résultats correspondants sont :

```
I    : momment d'inertie de la section par rapport à
       l'axe horizontal passant par le centre de
       gravité G de la section S
S    : aire de la section
Vs   : distance de l'axe horizontal, passant par G,
       aux fibres supérieures
I/Vs : module d'inertie supérieur
Vi   : distance de l'axe horizontal, passant par G,
       aux fibres inférieures
I/Vi : module d'inertie inférieur
rho  : rayon de giration
```

On peut déterminer les caractéristiques de sections creuses, il suffit de faire une décomposition et de regrouper éventuellement les parties pleines de mêmes niveaux. Pour une poutre, il est possible de calculer uniquement les semelles en entrant la hauteur de l'âme et 0 pour ses bases.

Il est bien évident que l'on peut utiliser cette feuille de calculs pour les caractéristiques par rapport à un axe vertical, il suffit d'effectuer une rotation de la figure.

La macro /I permet d'initialiser la feuille sous la forme présentée au début de cet article.

Voici la feuille de calculs de INER dans son intégralité. Pour des raisons de mise en page les lignes A1, E3, H10, H11, H12 E13 et H13 sont présentées ici sur deux lignes, mais sont à taper d'un seul tenant.

```
A1: [L5] '   caractéristiques de sections à base de
             trapèzes
A2: [L5] @REPETE("=";56)
A3: [L5] 'niv.|
B3: [L9] 'base sup|
C3: [L9] 'base inf|
D3: [L9] ' hauteur
E3: [L8] @CAR(124)&@REPETE(" ";7)&"Résultats"&@REPETE
         (" ";6)&"*"
A4: [L5] @REPETE("-";56)
```

```
A5: [L5] '  1 |
B5: (F3) D [L9] 0
C5: (F3) D [L9] 0
D5: (F3) D [L9] 0
E5: [L8] '| I    =
F5: (F4) [L15] @SOMME(K5..K13)-F6*F9^2
G5: (L) [L1] (B5+C5)*D5/2
H5: (L) [L1] @SI(G5=0;0;(2*B5+C5)*D5/3/(B5+C5))
I5: (L) [L1] @SOMME(D6..D13)+H5
J5: (L) [L1] +G5*I5
K5: (L) [L1] (3*B5+C5)*D5^3/12+G5*(I5^2-H5^2)
A6: [L5] '  2 |
B6: (F3) D [L9] 0
C6: (F3) D [L9] 0
D6: (F3) D [L9] 0
E6: [L8] '| S    =
F6: (F4) [L15] @SOMME(G5..G13)
G6: (L) [L1] (B6+C6)*D6/2
H6: (L) [L1] @SI(G6=0;0;(2*B6+C6)*D6/3/(B6+C6))
I6: (L) [L1] @SOMME(D7..D13)+H6
J6: (L) [L1] +G6*I6
K6: (L) [L1] (3*B6+C6)*D6^3/12+G6*(I6^2-H6^2)
A7: [L5] '  3 |
B7: (F3) D [L9] 0
C7: (F3) D [L9] 0
D7: (F3) D [L9] 0
E7: [L8] '| Vs   =
F7: (F4) [L15] @SOMME(D5..D13)-F9
G7: (L) [L1] (B7+C7)*D7/2
H7: (L) [L1] @SI(G7=0;0;(2*B7+C7)*D7/3/(B7+C7))
I7: (L) [L1] @SOMME(D8..D13)+H7
J7: (L) [L1] +G7*I7
K7: (L) [L1] (3*B7+C7)*D7^3/12+G7*(I7^2-H7^2)
A8: [L5] '  4 |
B8: (F3) D [L9] 0
C8: (F3) D [L9] 0
D8: (F3) D [L9] 0
E8: [L8] '| I/Vs =
F8: (F4) [L15] @SI(F7=0;0;+F5/F7)
G8: (L) [L1] (B8+C8)*D8/2
H8: (L) [L1] @SI(G8=0;0;(2*B8+C8)*D8/3/(B8+C8))
I8: (L) [L1] @SOMME(D9..D13)+H8
J8: (L) [L1] +G8*I8
K8: (L) [L1] (3*B8+C8)*D8^3/12+G8*(I8^2-H8^2)
A9: [L5] '  5 |
B9: (F3) D [L9] 0
C9: (F3) D [L9] 0
D9: (F3) D [L9] 0
E9: [L8] '| Vi   =
F9: (F4) [L15] @SI(F6=0;0;@SOMME(J5..J13)/F6)
G9: (L) [L1] (B9+C9)*D9/2
H9: (L) [L1] @SI(G9=0;0;(2*B9+C9)*D9/3/(B9+C9))
I9: (L) [L1] @SOMME(D10..D13)+H9
J9: (L) [L1] +G9*I9
K9: (L) [L1] (3*B9+C9)*D9^3/12+G9*(I9^2-H9^2)
A10: [L5] '  6 |
B10: (F3) D [L9] 0
C10: (F3) D [L9] 0
D10: (F3) D [L9] 0
E10: [L8] '| I/Vi =
F10: (F4) [L15] @SI(F9=0;0;+F5/F9)
G10: (L) [L1] (B10+C10)*D10/2
H10: (L) [L1] @SI(G10=0;0;(2*B10+C10)*D10/3/(B10
     +C10))
I10: (L) [L1] @SOMME(D11..D13)+H10
J10: (L) [L1] +G10*I10
K10: (L) [L1] (3*B10+C10)*D10^3/12+G10*(I10^2-H10^2)
A11: [L5] '  7 |
B11: (F3) D [L9] 0
C11: (F3) D [L9] 0
D11: (F3) D [L9] 0
E11: [L8] '| rho  =
F11: (F4) [L15] @SI(F6=0;0;@RACINE(F5/F6))
G11: (L) [L1] (B11+C11)*D11/2
H11: (L) [L1] @SI(G11=0;0;(2*B11+C11)*D11/3/(B11
     +C11))
I11: (L) [L1] @SOMME(D12..D13)+H11
J11: (L) [L1] +G11*I11
K11: (L) [L1] (3*B11+C11)*D11^3/12+G11*(I11^2-H11^2)
A12: [L5] '  8 |
B12: (F3) D [L9] 0
C12: (F3) D [L9] 0
D12: (F3) D [L9] 0
E12: [L8] @CAR(124)&@REPETE("=";22)
G12: (L) [L1] (B12+C12)*D12/2
H12: (L) [L1] @SI(G12=0;0;(2*B12+C12)*D12/3/(B12
     +C12))
I12: (L) [L1] @SI(G12=0;0;+D13+H12)
J12: (L) [L1] +G12*I12
K12: (L) [L1] (3*B12+C12)*D12^3/12+G12*(I12^2-H12^2)
A13: [L5] '  9 |
B13: (F3) D [L9] 0
C13: (F3) D [L9] 0
D13: (F3) D [L9] 0
E13: [L8] @CAR(124)&@REPETE("*";6)&" Init : /I "
     &@REPETE("*";5)
G13: (L) [L1] (B13+C13)*D13/2
H13: (L) [L1] @SI(G13=0;0;(2*B13+C13)*D13/3/(B13
     +C13))
I13: (L) [L1] @SI(G13=0;0;+H13)
J13: (L) [L1] +G13*I13
K13: (L) [L1] (3*B13+C13)*D13^3/12+G13*(I13^2-H13^2)
E15: (F3) [L8] 0
F15: [L15] '/re15~b5.d13~/cdb5.d13~
```

Pour taper cette feuille plus rapidement voici quelques indications:

-Largeur de colonnes :

A	B	C	D	E	F	G	H	I	J	K
5	9	9	9	8	15	1	1	1	1	1

-Cellules du champ `B5..D13` : format fix 3, déprotégées

-Cellules du champ `F5..F11` : format fix 4

-Cellule G5 : (B5+C5°*D5/2) à recopier en G6..G13

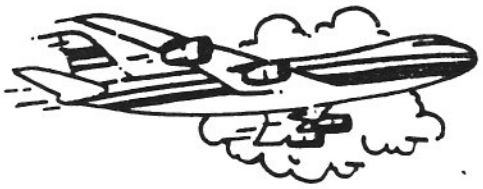
-Cellule H5 : (@SI(G5=0;0;(2*B5+C5)*D5/3/(B5+C5)))
à recopier en H6..H13

-Cellule I5 : (@SOMME(D6..D$13)+H5)
à recopier en I6..I11

-Pour I12 et I13, voir feuille

-Cellule J5 : (+G5*I5 à recopier en J6..J13)

-Cellule K5 : ((3*B5+C5)*D5^3/12+G5*(I5^2-H5^2))
à recopier en K6..K13



Pour toute la colonne A, la ligne 3, champs E5..F11, E12..E13, E15..F15, voir feuille.

Ne pas oublier d'enregistrer la macro /I en F15 et d'utiliser celle-ci pour réinitialiser la feuille.

Cette feuille représente 5516 octets, je l'ai créée avec pour toute documentation un aide-mémoire de la version 2.2 de Lotus, aussi il y a certainement des améliorations possibles ou même des maladresses à corriger. Suivant les besoins les champs A5..D13 et F5..K13 peuvent être allongés vers le bas. Je n'ai pas inclu d'unités car suivant les matériaux utilisés et les habitudes cela est variable. Le format des données et résultats n'est peut-être pas celui de tout le monde, donc des adaptations à faire.

L'usage d'un tableur pour ce genre de problème m'a semblé approprié et en particulier pour un calcul de section en pré-dimensionnement.

Guy Toublanc (276)

HEWLETT PACKARD 95LX
PC
PARIS

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 différent 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 obtiendriez un fichier au transfert impossible dans la HP48.

Programmes HP48 et HP28S

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.

Par mesure de sécurité sauvegardez vos programmes et fichiers, éventuellement verrouillez vos cartes.
Tapez le programme correspondant à votre machine:
- ASSCOD48 pour HP-48 ou ASSCOD28 pour HP-28S
(*attention aux parties distinctes HP-48 ou HP-28S*)
cela avec la plus grande attention car un mauvais fonctionnement peut entraîner un désordre fatal pour les objets contenus dans votre machine. La commande SPEED (JPC-85 page 12) peut être incluse au début de ASSCOD28. NEWLINE (←) s'obtient sur HP-48 par [flèche bleue] puis [.].

Pour tout assemblage de chaîne de codes procédez de manière suivante :

1- lancez le programme ASSCOD48 ou ASSCOD28
2- donnez le nombre d'octets (1/2 octet compris) puis validez avec ENTER.
3- tapez chaque ligne de codes après son numéro à 3 chiffres et sans les espaces puis validez.
4- tapez la somme de contrôle sm, validez. S'il y a erreur la ligne de codes sera demandée à nouveau. Pour HP-28 l'appui sur EDIT fera apparaître la ligne des codes qui pourra être corrigée, ensuite relancez avec CONT.
5- lorsque l'objet apparaîtra sur la pile stockez le dans la variable donnée en tête.
Nota: des versions ASSCODxx plus performantes (System Rpl et Assembleur) existent dans JPC numéro 95.

ASSCOD48 1114.5 octets cksum # 621Eh **ASSCOD28** 1032.5 octets cksum # 55CEh

```
@ ******************** HP-48 et HP-28S ********************
« RCLF « CLLCD 3 DISP 1 DISP ""
DO DUP 4 DISP
 DO UNTIL KEY END
@ ************* HP-48 ************ @@ ******* HP-28S ******
 IF DUP 51 == THEN DROP 1        @@ IF DUP "ENTER" ==
 ELSE IF DUP 55 == THEN DROP     @@ THEN DROP 1
   1 OVER SIZE 1 - SUB           @@ ELSE
  ELSE                           @@  IF DUP "BACK" ==
   CASE DUP 17 < THEN 54 + END   @@  THEN DROP 1 OVER
        DUP 66 < THEN 7 - END    @@    SIZE 1 - SUB
        DUP 76 < THEN 20 - END   @@  ELSE
        DUP 86 < THEN 33 - END   @@
        DROP 48 END CHR          @@
@ ******************** HP-48 et HP-28S ********************
  + DUP SIZE  1 + 5 MOD NOT 1  FC? AND IF THEN " " +
    END END 0 END UNTIL END  »
HEX 64 STWS "nombre d'octets ?" ""
@ ************* HP-48 ************ @@ ******* HP-28S ******
INPUT                            @@ 1 SF 3 PICK EVAL
@ ******************** HP-48 et HP-28S ********************
1 CF STR→ 2 * 16 DUP2 / IP 3 ROLLD MOD
DUP2 0 > + SWAP 1 + # 0h ""  1 5 ROLL
FOR i
 DO "ligne " i 1 - R→B # 1000h + →STR 4 6 SUB + DUP
  "                                          @ NEWLINE
  codes ?" + "---- ---- ---- ----" 7 PICK i <
  IF THEN 6 PICK DUP 4 / IP + 1 SWAP OVER - SUB END
  1 FS?C IF THEN DROP2 SWAP
@ ************* HP-48 ************ @@ ******* HP-28S ******
  OVER " a corriger" +           @@
  { -1 } ROT + INPUT             @@ CLMF HALT
@ ******************** HP-48 et HP-28S ********************
  ELSE 8 PICK EVAL END DUP
  WHILE DUP " " POS DUP
  REPEAT DUP2 1 SWAP 1 - SUB 3 ROLLD 1 + 25 SUB +
  END DROP 0 OVER SIZE 1 SWAP
  FOR j OVER j DUP SUB NUM j * + NEXT
  6 PICK + DUP # FFFh AND
  "                                          @ NEWLINE
  somme de controle ?" 6 ROLL SWAP + "==="
  11 PICK EVAL "#" SWAP + STR→ ==
  IF THEN 1 ELSE DROP2 1000 1 BEEP 1 SF 0 END
 UNTIL END ROT 5 ROLL DROP2 3 ROLLD +
 NEXT 5 ROLLD 4 DROPN
@ **** HP-48 **** @@ *************** HP-28S ***************
"GROB 8 " OVER  @@ "" SWAP 1 OVER SIZE
SIZE 2 / " " +  @@ FOR j DUP j DUP 15 + SUB "#"
+ SWAP + STR→   @@  OVER SIZE 1
# 4017h SYSEVAL @@  FOR i OVER i DUP SUB + -1 STEP
# 62B9Ch        @@  SWAP DROP STR→ ROT SWAP
SYSEVAL NEWOB   @@  # 3B82h SYSEVAL  SWAP 16
SWAP STOF »     @@ STEP DROP # 20238h SYSEVAL
                @@ # 4F3Dh SYSEVAL SWAP STOF CLMF »
```

```
SPOFF                          (HP28)
# 8FF9h                    17 octets

     0123 4567 89AB CDEF    sm

000: 69C2 0D10 0013 21B0    BBD
001: 0FFF 3081 5C08 D108    92F
002: 30                     9C2


PIXOFF                         (HP28)
# C314h                  22.5 octets

     0123 4567 89AB CDEF    sm

000: 76C2 04C3 C05A 7C03    DCC
001: 5270 76C2 0DD1 C02F    C6B
002: DA10 9F20 09F2 0       FD7


PIXON                          (HP28)
# 6D07h                  22.5 octets

     0123 4567 89AB CDEF    sm

000: 76C2 04C3 C05A 7C03    DCC
001: 5270 76C2 0DD1 C0EF    D88
002: DA10 9F20 09F2 0       0F4


PIXX                           (HP28)
# D2FEh                    25 octets

     0123 4567 89AB CDEF    sm

000: 76C2 04C3 C05A 7C03    DCC
001: 5270 76C2 0DD1 C08C    C95
002: EA14 F9C0 09F2 009F    AD3
003: 20                     B65


POLYG                          (HP28)
# 417Ch                  157.5 octets

     0123 4567 89AB CDEF    sm

000: 76C2 00C8 7034 411E    CCA
001: D211 84D1 18F7 D328    AA8
002: D117 5120 B35C 0C27    7FD
003: D375 1200 C870 E602    49E
004: 00BB 113A CD36 8311    1D1
005: 9C21 1257 D348 120D    EBD
006: 1D11 38C1 1B35 C0C2    C87
007: 7D3C 27D3 69A2 0AE0    AEF
008: E3AE 0E30 9F20 5B95    8FB
009: 006F 6049 850A 4020    4FF
00A: 119D 37E5 40D7 390F    33A
00B: 38D3 EEE8 0E01 B251    10E
00C: 8D34 9850 4812 0119    C9B
00D: D37D 8D3B EC11 A402    AA6
00E: 0ABA D37E 540D 7390    862
00F: 268D 3EEE 80E0 1B22    696
010: 1D20 40C4 94E4 54C1    434
011: 6D35 0850 6012 0481    F6B
012: 20BE C115 0850 FF5D    E6F
013: 3379 5009 F20          C4C


LINE                           (HP28)
# 3B17h                  281 octets

     0123 4567 89AB CDEF    sm

000: 69C2 0C22 008F 1805    C96
001: 0AF2 CEAE 2134 1542    99A
002: 1B11 10C1 5A00 8808    607
003: 6050 8531 477D 4110    1F7
004: C7F4 1108 1641 4679    E33
005: 3110 A7B3 110B 1108    A56
006: A620 11C1 124E 08A6    7A5
007: 9012 36A7 0247 2A1A    4D6
008: F712 3118 7591 8B7A    29E
009: 0DF8 5212 310C 102D    F58
00A: F108 81FE BD5C F7A4    FC4
00B: 0249 0DD1 1141 1BE6    D73
00C: 94A8 0818 FA1E 850F    C53
00D: 10B1 12E4 9488 0818    902
00E: F811 0211 8C1C F5FB    88D
00F: 7700 8D5E 5212 011B    52E
010: 8724 0DE1 3534 8800    163
011: 08B6 0031 349E A40E    F65
012: A348 FBFF 5703 2048    CB9
013: C4C4 C4CA 1331 3034    909
014: F100 08B6 0081 C81C    689
015: 137C 2137 1323 030E    2A9
016: 06F6 550A 06A0 C59F    169
017: 14B8 6350 B8C0 E0E4    FEA
018: 50B8 C159 0011 3713    B74
019: 4173 170D 0153 3320    6BE
01A: 059B 2A0D 117E 5073    3F1
01B: 2300 9BA5 0ABA 17D1    298
01C: 3313 780D 0EE1 35D0    081
01D: 1531 D8D3 8091 3520    C93
01E: D230 6063 1C90 6328    8D4
01F: 1C23 A845 40C3 A0C5    6B9
020: AF0D 8907 0075 8EDB    5BF
021: E115 B090 840F 9D90    392
022: 1E29 0A70 B44F AC60    23D
023: 10                     2CE


ROTIMG                         (HP28)
# FA36h                  638 octets

     0123 4567 89AB CDEF    sm

000: 76C2 0CD5 70EE 870A    EF6
001: 4020 B8A1 149C D376    CF8
002: C20A 1120 AC3A 009F    B17
003: 2037 650E 6020 21D2    762
004: 0500 5751 4944 52E4    40C
005: 90A4 0202 F880 BEC1    29F
006: 184D 11F1 4507 6C20    F59
007: D815 221D 2050 0575    B19
008: 1494 4538 C11A 37A0    868
009: 09F2 021D 2040 C435    4E1
00A: 4585 877A 0FEA 1149    31C
00B: CD37 6C20 69A2 0E4A    16F
00C: 2070 0000 3CA2 F38B    F92
00D: 2F34 C2F3 0D2F 3CD2    ED9
00E: F38E 2F34 F2F3 003F    D21
00F: 3C03 F3ED 1F3E 32F3    C48
010: 622F 3232 F3E0 2F3A    AC6
011: 12F3 AE0A 1E4A 2070    84A
012: 0003 5E4A 2070 0001    380
013: 5E4A 2070 000F 409F    0DD
014: 205E 020C 6970 9C21    DF0
015: 1BEC 11F1 4502 1D20    A7E
016: 5005 7514 9445 F145    718
017: 021D 2050 3545 05E2    372
018: 7569 A207 6C20 A112    FF6
019: 0339 2020 0000 0000    9C1
01A: 0002 2033 9209 9900    544
01B: 0000 0000 010E 6020    036
01C: 54A2 2E60 205E 020A    D21
01D: 4020 BEC1 15E0 2054    9D7
01E: A225 4A22 09F2 076C    76E
01F: 2033 9207 9900 0000    221
020: 0000 0105 D5D3 6012    E32
021: 0022 A009 F207 6C20    B01
022: 3392 0999 0000 0000    56A
023: 0001 05D5 D360 1200    0FD
024: 22A0 09F2 076C 2033    D79
025: 9209 9900 0000 0000    78D
026: 0205 D5D3 6012 0022    2E2
027: A009 F207 6C20 3392    F4B
028: 0999 0000 0000 0003    94C
029: 05D5 D360 1200 22A0    518
02A: 09F2 076C 2033 9209    158
02B: 9900 0000 0000 0405    B7B
02C: D5D3 6012 0022 A009    731
02D: F207 6C20 3392 0999    3CE
02E: 0000 0000 0005 05D5    F4C
02F: D360 1200 22A0 09F2    BD2
030: 076C 2033 9209 9900    7B9
031: 0000 0000 0605 D5D3    457
032: 6012 0022 A009 F207    0B5
033: 6C20 3392 0999 0000    C00
034: 0000 0007 05D5 D360    88A
```

```
035: 1200 22A0 09F2 076C   5EC
036: 2033 9209 9900 0000   0AF
037: 0000 0805 D5D3 6012   CEA
038: 0022 A009 F207 6C20   9B9
039: 3392 0999 0000 0000   422
03A: 0009 05D5 D360 1200   FD5
03B: 22A0 09F2 076C 2033   C51
03C: 9201 0000 0000 0000   5E2
03D: 0105 D5D3 6012 0222   151
03E: B009 F207 6C20 FC1A   00A
03F: 05E0 2002 2A00 9F20   C70
040: 76C2 0FC1 A05E 0203   96D
041: 10B0 09F2 076C 20FC   804
042: 1A05 E020 811B 009F   54E
043: 2076 C20F C1A0 5E02   30F
044: 0F43 B009 F207 6C20   019
045: FC1A 05E0 2022 2B00   C23
046: 9F20 76C2 0601 20A4   85B
047: 020F C1A0 42D8 05E0   608
048: 2002 2A00 9F20 76C2   331
049: 0A11 2021 D205 0350   E73
04A: 5545 4440 9F20 76C2   B99
04B: 0337 70AC 3A00 9F20   902
04C: 76C2 0A11 2085 E810   581
04D: 9F20 09F2 0C56 D3A4   3C6
04E: 1808 14A0 09F2 033C   0F6
04F: 6099 6500 9F20        15A
```

```
RING                          (HP28)
# 7650h                  322 octets

          0123 4567 89AB CDEF   sm

000: 69C2 0E72 008F 1805   CC5
001: 0AF2 CEAE 2134 1542   9C9
002: 081B 1110 C15A 0808   65E
003: 6050 8531 477E 9110   29B
004: B71B 1108 1641 467A   F4F
005: 8110 A860 6078 9111   B10
006: 2EE5 7085 1FA0 6104   7CE
007: C4D2 1093 03E2 D711   4AB
008: 9633 07D7 0241 1990   0A1
009: FE01 2ACE D512 AE9C   0C5
00A: 6C62 0C38 18F3 5119   DAF
00B: E610 9112 E24A C119   B0A
00C: 8A66 07D3 007C E491   8BD
00D: 114C C861 8081 8F01   5B1
00E: 1026 78F8 D5E5 2111   2CD
00F: 9FA1 096D 1011 9FA1   091
010: 0964 2084 0783 040E   CF1
011: 7920 11AF A10A 7820   A1A
012: 49D7 6101 1912 AFA1   7DA
013: 0987 0008 505E C119   4E4
014: FA10 911A 110E A34F   313
015: 1000 8B60 0131 1111   DC3
016: 1BCA 3498 0008 B240   A2D
017: 0331 349E A40E A348   84E
```

```
018: FBFF 5703 2048 C4C4   627
019: C4CA 1331 3081 C81C   398
01A: 137C 2137 1323 030E   FB8
01B: 0630 1550 A06A 0C59   D1F
01C: F14B 8635 0B8C 0E0E   BFA
01D: 450B 8C15 9003 1371   7BE
01E: 3414 3C4A 6484 0550   3F8
01F: 8501 7317 0D01 5333   F9E
020: 2005 9B2A 0D11 7E50   D0E
021: 7323 009B A50A BA17   B6A
022: D133 1378 0D0E E135   8FD
023: D015 31D8 D380 9135   5B3
024: 20D2 3060 631C 9063   1F4
025: 281C 23A8 4540 C3A0   F4A
026: C5AF 0D89 0700 758E   D19
027: DBE1 15B0 9084 0F9D   B89
028: 9010                  D75
```

```
UP                            (HP28)
# D9FEh                 22.5 octets

          0123 4567 89AB CDEF   sm

000: 76C2 0760 82A4 0208   C0B
001: 0140 F7F6 0251 4049   852
002: CD38 E450 09F2 0      BCE
```

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** (**H**andhelds **E**uropean Clubs **EX**change), la fédération des principaux clubs Européens d'utilisateurs de calculateurs HP.

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

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

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

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

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

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

Imprimé par Paris Copie, 4 rue Linné, 75005 Paris