JPC

NOVEMBRE 1993

NUMERO 91

Le numéro 40 F

A PROPOS DU CLUB

Le Bureau	Editorial	*1*
P. Courbis	Droit de réponse	*2*
	Courrier du Coeur	*4*

HP28

G. Toublanc	Quel Avenir pour la HP28 ?	*6*
G. Toublanc	Recherche de Chaînes	*6*
P. Jebeily	Graphismes et animations, correctif	*8*

HP48

G. Toublanc	Le kit de connection HP48GX	*10*
G. Toublanc	Présentation de *CAT-48*	*10*
R. Pulluard	Calcul du numéro de semaine	*11*
G. Toublanc	Calculs hyperboliques	*12*
G. Toublanc	Trucs et astuces	*13*
P. De Sacy	Points d'entrées correspondant SX & GX	*15*
P. De Sacy	Cartes TDS sur le port 2 (48GX)	*16*
P. De Sacy	Changement de répertoire (Acte II)	*18*

HP95

G. Merlier	La configuration du Drive95	*22*
J. Belin	Conversion d'images pour HP95	*25*
J. Belin	Programmation du graphisme, correctif	*28*

Le coin des codes *30*

EDITORIAL

Après quelques années d'attente, les cartes PCMCIA commencent enfin à faire leur entrées en force dans tous les domaines de l'informatique portable. En effet, suivant l'exemple du HP95, nous voyons d'un côté de plus en plus de portables acceptant ces cartes. De l'autre côté, alors que ces cartes se cantonnaient jusqu'à peu de temps aux simples mémoires, quasiment tous les types d'interfaces ont explosé en l'espace de quelques mois. D'abord les disques durs et les cartes Flash (au fait, on commence seulement à voir apparaitre le nom de SunDisk dans la presse dite spécialisée !) et maintenant tout ce qu'on peut imaginer : Modems, Contrôleurs SCSI ou cartes réseaux et même (avec un an d'avance) les cartes *Pager* permettant de recevoir ses message directement sur son HP ! Nous ne manquerons pas de tester quelques unes de ces nouvelles cartes, dès que nous pourrons en disposer. Que quelques sociétés aient l'idée d'en adapter quelques unes pour la HP48, et cela serait le Paradis....

Autre Chose. Certains auteurs tiennent compte de nos observations et n'hésitent pas à nous envoyer un exemplaire de leur nouvel ouvrage, alors que nous avons pas hésité à faire une critique relativement dure à propos de leur première édition (nous pensons tout particulièrement à Mathieu Cornillaut et sa nouvelle édition de *HP48 en Prépa*, que nous venons juste de recevoir et que nous vous présenterons dans le prochain numéro). D'autres se sentent personnellement attaqués lorsque nous faisons quelques observations sur leur livre, et ne comprennent pas que nous jugions logique d'être plus critiques pour un livre arrivant à sa quatrième version. Alors quelle attitude devons nous adopter ? Passer sous silence les points qui ne nous plaisent pas, afin de ne pas froisser l'auteur ou l'éditeur ? Ou faire de la critique un inventaire exhaustif de tous les défauts du livre ? Sans aller jusque là, l'experience prouve que ce n'est qu'en mettant au grand jour les défauts d'un produit qu'on incite l'auteur à les corriger. Au delà de cet exemple, c'est la liberté d'expression du bureau et de ses adhérents que l'on tente encore une fois de critiquer. Rassurez vous, il n'est pas dans notre intention de céder à quelque pression que ce soit. Après tout, à quoi cela servirait-il que nous faisions tout pour conserver notre indépendance financière (c'est à dire sans passer de publicité dans JPC) si ce n'est pour conserver cette totale liberté d'expression ?

Pour revenir à un sujet plus agréable et puisque nous parlons de livres, nous venons aussi de recevoir un exemplaire de *PC de Poche : HP100 & 95LX*, le premier livre Français consacré aux HP95 et HP100. Le temps de le lire et nous vous le présenterons aussi dans le prochain numéro.

Dans un futur un peu plus éloigné, si vous n'avez rien prévu pour votre week-end des 28 et 29 mai 1994, le club Hollandais Prompt-HPGC a pensé à vous. En effet, il organisera à cette date une conférence internationnale à Amsterdam, pour fêter ses dix ans. Bien sûr, nous projetons d'y aller. Si vous désirez vous joindre à nous, manifestez vous dès maintenant pour connaitre les modalités pratiques.

Enfin, pour revenir sur ce JPC, certains nouveaux adhérents ont manifesté leur étonnement dans le fait que le journal soit en retard. Pour eux, nous rapellerons encore une fois que ce retard est dû uniquement au fait que nous ne recevons pas suffisement d'articles provenant des adhérents. Pour les autres, nous ne pouvons que rapeller encore une fois que ce retard peut être éliminé dès la semaine prochaine si vous nous envoyez tout de suite vos articles.

CHOC EN RETOUR

"PROGRAMME BIZARRE ET DERIVES LITTERAIRES"

Lettre ouverte à Monsieur Guy TOUBLANC

Note de Jacques Belin : Paul Courbis s'est senti attaqué personellement par l'article de Guy Toublanc à propos du livre "Voyage au Centre de la HP48". Il a donc souhaité nous faire publier le droit de réponse suivant. Nous le diffusons dans son intégralité, tel que nous l'avons reçu. Je me permettrais seulement d'apporter quelques observations, précédées par la mention "NBJB".

Après avoir lu la violente diatribe de Monsieur Guy Toublanc (276) à l'encontre de *Voyage au centre de la HP48* (JPC 90, pages 14 à 18), il m'a paru opportun de répondre ici à ses nombreuses critiques.

Commençons par le commencement (les phrases extraites de l'article de Monsieur Guy Toublanc sont entre guillemets et en italiques) :

«Les éditions du livre Au centre de la HP48 *se suivent mais se ressemblent par la conservation de programmes qui ne sont pas des modèles du genre. Dans la dernière édition [...], on trouve le programme* R→Q√ *(page 402) qui est le même que celui de la troisième édition (page 358), donc en principe devrait être sans gros défauts.»*

Monsieur Guy Toublanc débute son article par une longue critique du programme R→Q√ (critique un peu rapidement généralisée à l'ensemble du livre). Il est vrai que les imperfections de ce programme m'avaient échappé et c'est pourquoi il n'a pas été modifié entre la troisième édition de *Voyage 48 S/SX* et *Voyage 48 G/GX*. Je m'étonne néanmoins de l'attitude qui consiste à attendre avec un malin plaisir la sortie du nouveau livre pour en "remarquer" les erreurs. Il aurait été plus profitable pour tous de me les signaler à temps : j'aurais été très heureux d'en tenir compte pour améliorer *Voyage*. D'ailleurs, toute personne découvrant des erreurs dans le livre m'aiderait grandement si elle me faisait parvenir un "bug-report" (par exemple, à l'adresse indiquée dans le livre dans la page de copyright). NDJB : Guy Toublanc a fait, dans ces mêmes colonnes, de nombreuses remarques ou améliorations sur des programmes parus dans Voyage (et dans d'autres livres d'ailleurs). JPC étant distribué chez Maubert, il te suffisait d'y jeter un coup d'oeil. De plus Guy est-il si bon que cela pour qu'il soit le seul à trouver les bugs dans des livres ?

«Tout d'abord je signale que cette rapide analyse est basée sur un exemplaire que j'ai acheté et qui n'a pas été envoyé au club (il y a des principes qui se perdent).»

Je m'interroge sur la signification de cette remarque faite dans le cadre d'une analyse qui se voudrait objective. ôtez-moi donc un doute : vos conclusions auraient-elles été différentes si le livre vous avait été offert ? NDJB : Pas du tout. Si tel était le cas, cela s'appellerait de la corruption. Le fait qu'un auteur nous envoie un article (ou tout autre objet) ne peut lui garantir qu'une chose : que nous en fassions une présentation dès le prochain JPC.

Quant aux "principes qui se perdent", je n'ai pas le souvenir d'avoir un jour reçu un exemplaire gratuit de JPC. Par contre Sébastien LALANDE et moi-même nous souvenons avoir été "virés comme des malpropres" d'une des réunions du club (en théorie ouvertes à tous ???), sous le prétexte que Sébastien n'avait pas payé sa cotisation. Cela ne m'a pas empêché de citer le club sans remarque désobligeante d'aucune sorte. NDJB : Nous n'avons jamais recu aucun livre de ta part (que nous te le demandions ou pas, d'ailleurs). Nous n'avions donc pas de raison de t'envoyer un exemplaire gratuit du journal, comme nous le faisons aux autres auteurs (Jean-Michel Ferrard peut en témoigner). Il ne faudrait pas mettre la charrue avant les boeufs. En ce qui concerne le fait que tu aurais été "viré" d'une réunion, cela s'est passé avant que je prenne la présidence du club et je n'ai pas de souvenir précis à ce sujet. Il est possible que la personne qui ait demandé à Sébastien de penser à régler sa cotisation s'y soit mal pris (en effet, certains ont une facheuse tendance à oublier, tout en étant de bonne foi). Depuis que le club organise des réunions, nous n'avons jamais seulement tenté de décourager une personne d'y venir, bien que pour certains "pique-assiette" ou "indésirables" cela nous ait souvent démangé. Indésirables dont vous ne faites absolument pas partie (Sébastin et toi), comme tu as pu le voir ensuite, les quelques fois où tu est revenu aux réunions.

«Qu'en est-il aujourd'hui avec cette 4ième édition actualisée pour les HP48 G/GX ?»

Je tiens à préciser que *Voyage 48 G/GX* ne remplace nullement *Voyage 48 S/SX* : il ne s'agit donc pas d'une quatrième édition, mais d'un nouveau livre (d'ailleurs, les deux ouvrages possèdent des numéros ISBN distincts). NDJB : Il ne faudrait tout de même pas jouer sur les mots. S'il est vrai de le livre consacré à la GX comporte de nouveaux points et que les tableaux comportent des adresses différentes, sa structure est quasiment la même que pour ceux de la SX. En fait, pour les parties communes, le texte est quasiment le même. Nous ne pouvons donc pas parler d'un nouveau livre, mais plutôt d'une mise à jour consacrée à une nouvelle machine .

«A l'époque j'avais trouvé assez scandaleux qu'un erratum n'ait pas été glissé dans les exemplaires de la première édition car les coquilles étaient connues dès les premières semaines de la sortie du livre. Ce qui avait été fait pour le livre sur la HP28 n'a pas été renouvelé et donc on était en régression du côté honnêteté.»

Concernant "l'honnêteté" de la deuxième édition : aucun erratum n'a été inclus dans la première édition, cela est vrai. Mais il est vrai aussi que, vu la quasi-inexistence de "bugs-reports" qui m'ont été adressés, ces imperfections ne devaient pas être si gênantes que cela. Vous sous-entendez que les "coquilles" vous étaient connues dès les premières semaines : il est regrettable que vous ne m'ayez jamais écrit à ce sujet (les moyens de le faire ne vous manquaient pas : mon adresse est connue du club et vous n'êtes pas sans fréquenter le magasin Maubert Electronic où je passe moi-même assez souvent). NDJB : Peu après la sortie de la première édition, Guy Toublanc m'a fait part d'une discussion (entre toi et un autre utilisateur) ayant eu lieu chez Maubert à propos de ces bugs, alors qu'il était lui-même dans le magasin. Pourquoi aurait-il dû t'ecrire, puisqu'il a obtenu ces informations en même temps que toi ?

«On peut constater que la liste des objets (avec les adresses) contenus dans la Rom a disparu par enchantement alors que celle des messages a été maintenue. Est-ce un oubli ou le temps manquait-il pour vérifier cette liste ou manque de place dans le livre ?»

Il ne s'agit ni d'un oubli, ni d'un manque de temps ou de place. Après enquête auprès de plusieurs utilisateurs du livre, il est apparu que cette liste ne présentait pas d'intérêt particulier et n'a donc pas été reproduite dans *Voyage 48 G/GX*. Il s'agissait d'autant moins d'un manque de temps que j'ai depuis longtemps réalisé une série de programmes générant ces listes de manière automatique (en langage C, pour station UNIX sous X.Window).

«Si les externals *ont enfin été utilisés dans les programmes en langage machine (dans la troisième édition on était encore à faire* DUP DROP *en début de programme) ce ne sont pas toujours les plus adéquats. Dans ces programmes les entiers système en Rom ne sont pas utilisés bien que leurs points d'entrée soient supportés par HP.»*

L'objet des différents *Voyages* est la programmation en langage-machine. Seuls les externals les plus utiles ont donc été retenus et vérifiés, principalement dans l'optique d'une utilisation avec des programmes en langage-machine (en particulier les externals de vérification d'arguments, de conversion de type...). Il était donc inutile d'encombrer le livre avec des milliers d'adresses ne servant presque jamais. D'ailleurs, il existe de très bons ouvrages sur le sujet qu'il aurait été stupide et inutile de plagier (*Les secrets de la HP48* de Jean-Michel Ferrard).

«Un parti pris vis-à-vis de la documentation HP et de ses outils de développement qui existent et que l'auteur ne peut pas ignorer car étant sur la liste officielle des développeurs HP, il a plus que la plupart de nous des facilités pour accéder à cette documentation.»

Je suis en effet "développeur officiel HP", mais depuis peu de temps puisque c'est grâce au livre *Voyage au centre de la HP48 S/SX* que j'ai pu le devenir. Les documentations auxquelles Monsieur Guy Toublanc fait allusion sont revêtues du "Copyright" HP et ne peuvent donc absolument pas être reproduite dans un livre. (NDJB : Le seul fait d'ecrire un livre sur les internals non rendus publics par des documents tels que les Goodies #4 (ou de publier des programmes les utilisant dans un journal tel que JPC) ne viole t-il pas déjà le copyright de HP ?). Elles ne présentent, en outre, que la programmation en System-RPL (qui, encore une fois, n'est absolument pas l'objet de *Voyage*) ainsi que la liste des instructions du Saturn (entachée de très nombreuses erreurs, car il s'agit de l'antique documentation du HP71, plutôt mal remise au goût du jour. Ainsi, tous les cycles-machine présentés sont faux, et ont dû être mesurés sur la HP48 pour *Voyage*, certaines instructions ne sont pas documentées...).

Quant aux "outils" fournis par HP, ils sont bien loin de valoir ceux auxquels il est fait référence dans *Voyage* (HPDS, Areuh, Satass.) et sont, de plus, frappés eux aussi du copyright HP. NDJB : Ces outils portent peut-être le Copyright HP, mais une version d'entre eux est disponible et libre d'utilisation. C'est celle qui est diffusée sous le nom de Goodies 4, et que nous distribuons dans notre "Kit de développement". Même si ils ne sont pas les meilleurs, ce sont les outils et documentations de référence utilisés par la grande majorité des programmeurs au niveau mondial. Ce qui n'est pas le cas pour les programmes que tu cites.

«Des programmes à revoir qu'on ne devrait pas retrouver sous la même forme d'une édition à la suivante [...]. J'ai fait allusion à de nouveaux programmes faits par d'autres. Certains ont des faiblesses de forme de programmation et je me garderai bien de critiquer leurs auteurs [...]. Ce que je reproche c'est au professionnel *de ne pas avoir aidé les auteurs à redresser leurs maladresses ce qui aurait permis au lecteur d'avoir des modèles à imiter.»*

Vous critiquez à nouveau les programmes de *Voyage*. Encore une fois, j'aurais préféré recevoir à temps des suggestions constructives plutôt que des critiques un peu trop tardives. Quant aux programmes qui m'ont été très aimablement fournis par d'autres personnes, j'ai pris le parti de respecter l'originalité des ouvres qui m'ont été envoyées. Il est vrai que j'aurai pu réécrire plus ou moins partiellement ces programmes et me les attribuer. Mais, pour moi, ça, c'est du vol. NDJB : Tu pouvais pourtant très bien le faire en mentionnant le nom de l'auteur original. A l'époque où tu faisais partie du club (et maintenant encore), cela a toujours été la règle. Et tout le monde (y compris l'auteur) était d'accord à ce sujet, car le plus

important était d'avoir le meilleur programme possible. Plus que de savoir quels étaient les apports de l'un ou l'autre sur le programme. Il est regrettable que tu aie oublié cette philosophie et que tu ne l'aie pas répercuté autour de toi.

«Un gonflement excessif du volume du livre par des artifices typographiques : trop de pages avec peu de texte»

Il est vrai que quelques bas de pages sont vides. Mais il faut savoir choisir entre une présentation claire et lisible et un texte tassé sans structuration. Quant aux blancs qui subsistent parfois, ils permettent d'insérer des ajouts dans des éditions futures, sans que cela nécessite un retirage complet de l'ouvrage (trop coûteux pour être accepté par un éditeur). Préféreriez-vous donc un ouvrage figé dans ses imperfections et dans ses lacunes ? C'est malheureusement ce qui a été le cas pour *Voyage 28 c/s*, mais il est vrai que vous ne vous en êtes jamais plaint.

«Les chapitres du début n'ont pas leur place dans un ouvrage s'adressant à des lecteurs qui veulent aller au-delà de ce que le constructeur met à leur disposition. Je ne suis pas le premier à le dire.»

Ils sont là pour permettre à un utilisateur novice d'utiliser les programmes de *Voyage* sans même les comprendre, et, donc, de se familiariser avec la machine. Devait-on réserver *Voyage* à une élite jalouse de le rester ?

La conclusion de Monsieur Guy Toublanc est dure : *Voyage* n'est qu'*«une bonne compilation d'informations, utile et nécessaire dans ce domaine car actuellement non concurencée ailleurs mais l'auteur qui se transforme progressivement en homme d'affaire (qui marche bien) pourrait lasser ses fidèles lecteurs par un cumul de pis aller et un manque de fini.»*

A vous entendre, on pourrait croire que *Voyage* ne contient aucune information originale, mais uniquement des idées et des programmes recopiés avec une multitude d'erreurs. Ce n'est pas vraiment le cas, et c'est même plutôt le contraire qui s'est produit : nombreux sont ceux qui ont pillé le livre sans vergogne... NDJB : Guy n'a jamais dit "Voyage N'EST QU'UNE bonne compilation" mais "Voyage EST UNE BONNE compilation". C'est toi qui déforme ses propos. La preuve en est que par le passé, il a souvent présenté tes livres comme étant la référence incontournable pour ceux qui voulaient mieux connaitre leur machine. L'article que tu mets en cause ayant seulement pour but de dire qu'il regrettait que tu n'aie pas profité de la nouvelle edition pour corriger les quelques erreurs présentes et présenter de nouvelles versions de programmes qui n'ont jamais évolué. Quant à être un "homme d'affaire", je préfère de pas répondre à ce genre d'insinuations : ceux qui me connaissent vraiment auront rectifié d'eux-mêmes...

Pour finir, je voudrais juste m'étonner d'une telle animosité de la part d'un membre de PPC Paris : 4 pages de critiques acerbes et seulement quelques mots flatteurs... pour l'article de Wlodek Mier-Jedrzejowicz. NDJB : Bien sûr nous (ainsi que tous les autres clubs) te remercions pour la présence de cette liste des clubs. Mais il est vrai que par le passé, toutes les personnes écrivant un livre sur HP mentionnaient au moins les coordonnées des clubs nationaux. Ce n'est qu'une bonne habitude reprise, et c'est ce qu'a voulu dire Guy. Sans plus. D'autre part, aurait-il été normal que nous changions d'opinion sur un livre pour la seule raison que nous y soyons cités ? Il est en outre étrange qu'un "critique littéraire" (NDJB : Guy n'a jamais prétendu en être un !) ne cite pas les référence des livres mis en cause, ce que je répare ici :

Voyage au centre de la HP48 S/SX par Paul COURBIS et Sébastien LALANDE, 3ème édition, Editions Angkor, Paris. ISBN 2-87-892-003-1, 518 pages, prix : 260 Francs.

Voyage au centre de la HP48 G/GX par Paul COURBIS, 1ère édition, Editions Angkor, Paris. ISBN 2-87-892-006-6, 618 pages, prix : 265 Francs.

Ces deux livres sont disponibles chez Maubert-Electronic (Paris), dans les Fnac (en général au rayon librairie), ainsi que dans diverses librairies spécialisées.

Paul Courbis (ex-392)

COURRIER DU COEUR

Christian Rapello
146, rue de Charenton
75012 PARIS
Tel : (1) 43 47 59 74 (soir)

Vend :

HP95LX 1Mo (très peu servi); etui cuir; Alimentation. Prix : 2800F

HP28

G. Toublanc	Quel Avenir pour la HP28 ?	*6*
G. Toublanc	Recherche de Chaine	*6*
P. Jebeily	Graphismes et animations, correctif	*8*
	Le coin des codes	*30*

LA HP28S ? ENCORE !

La rubrique HP-28 aura été présente 12 fois sur 16 dans JPC depuis mai 1992. Les HP-28 ne sont donc pas ignorées de PPC Paris même si une plus forte participation de lecteurs aurait été la bienvenue.

Qu'en est-il de l'avenir des HP-28S ?

Certains étudiants s'en contentent, car dans leurs études une HP48 ne leur apporterait rien de plus. Ceux qui font des mathématiques pures ignorent les résultats numériques au profit des démontrations, et là, nos chères machines n'ont pas été conçues pour cela. D'autre part une partie des étudiants se voit interdire l'usage des calculatrices lors des examens. Dans ces conditions une HP-28S peut paraître suffisante d'autant plus que le format et le double clavier sont appréciables.

Etant données les limites de la Ram, il me semble que le travail essentiel consiste à réaliser le maximum de programmes tenant le minimum de place. Pour cela il faut que chacun se mette au travail *quelque soit son niveau de programmeur et son langage de programmation*. Ce qui compte c'est de montrer qu'il existe des centres d'intérêt pour chacun et que le club n'est pas un ensemble de fantômes. Alors manifestez vous comme l'a fait Paul Jebeily qui a peut-être moins de temps que certains (car pris par ses études) mais qui sait faire don de ses créations et cela sans moyens matériels confortables.

Le club peut fournir des utilitaires, à la fois pour HP-28S et PC, de communication à ceux qui ont une HP-28S connectable à un PC travaillant sous DOS. Une liste d'adresses compilées par mon ami Mathieu Clabaut peut également être fournie sur disquette. Et n'oublions pas les listes de mnémoniques et les décompilateurs de J.F Garnier qui a mis généreusement tout cela à la disposition de chacun.

Guy Toublanc (276)



A LA RECHERCHE DES TRESORS CACHES

Après avoir fourni un ensemble d'outils à la fois classiques (mais actualisés) et nouveaux pour HP-28S voici le tout dernier `FIND` qui nous permettra de trouver en ROM ou en RAM l'adresse d'une suite de codes.

Il existe déjà des solutions:

- `SEARCH` du livre *Au centre...* mais en *User Rpl*, donc trop lent.
- `FINDMEM`, pour HP48, de Laurent Grand (JPC 85 pages 18-19). Réalisé en assembleur ce programme est donc plus rapide.
- La méthode américaine qui consiste à mettre dans un entier binaire le code mais inversé. On peut y placer jusqu'à 15 codes car le dernier chiffre de l'entier binaire est réservé au nombre de codes moins un. Si le code a été trouvé le programme renvoie l'adresse + 1. Cette méthode présente des avantages. La programmation en assembleur est très simple et très courte car ne nécessitant pas de conversion ascii→hexa. Les occurences suivantes se trouveront en activant simplement la commande car l'adresse de départ (adresse trouvée + 1) est déjà sur la pile.

C'est en adaptant cette dernière méthode que je vous propose `FIND` qui est plus simple d'emploi que son inspirateur.

Syntaxe

niveau 2 : la chaîne des codes (maximum 16 codes) et dans le même ordre qu'en mémoire.
niveau 1 : l'adresse de départ

`FIND` → adresse + 1 si trouvé, sinon # 1h

Exemple : recherche de programmes. On sait que le prologue est 02C67 donc en mémoire 76C20 :

```
2:"76C20"
1:# 0h

FIND → # F0Ah  donc # 00F09h
FIND → # 11CBh  donc # 011CBh
```

etc..

Certains rétorqueront qu'un maximum de 16 codes est trop restrictif. Je justifierai ceci par les raisons suivantes:

- Mon objectif est de rendre la recherche la plus rapide possible. N'oublions pas que nous aurons, parfois, à parcourir une grande partie de la Rom ou de la Ram. Le code cherché est stocké dans un seul registre, rendant les tests très rapides.
- Il est souvent illusoire de chercher de longues séquences de code pour retrouver des programmes identiques à ceux d'une autre machine (HP-71, 48). En effet s'il est encore possible de trouver des sauts courts avec les mêmes offsets, en ce qui concerne les autres sauts ou appels de routines il est exceptionnel d'avoir du code identique, mon expérience dans ce dommaine est là pour le prouver. Il vaut mieux à partir des adresses fournies par FIND faire une analyse plus étendue mais plus nuancée en désassemblant le code.

Le décompilateur MON28 de Jean-François Garnier, qui fonctionne sur PC, possède une commande de recherche de code. Il faut fournir les adresses de début et de fin de recherche ainsi que la suite des codes. Avec mon PC qui n'est qu'un 286 cadencé à 16 Mhz, la recherche des codes 137135144 entre les adresses # 0h et # 40000h de la Rom se fait en 137 secondes alors que le programme ci-dessous LFIND qui fait la recherche sur toute la mémoire, donc une plage 4 fois plus importante, me donne le résultat en 30 secondes. Aussi je préfère observer le code donné par le décompilateur et aux adresses fournies par FIND.

Voici le fichier source (SPEED est intégré dans le programme):

FIND
cksum # E0C5h 126 octets

```
ASSEMBLE
       con(5)  #02c67   * DOCOL
       con(5)  #0c3c4   * CK2NOLASTWD
       con(5)  #04f3d   * TOTEMPOB

       con(5)  #02c96   * DOCODE
       rel(5)  fin

       gosbvl  #5081    * SAVPTR
       d0=(5)  #fff00   * pour adr. speed
       lchex   f        * pour speed maxi
       dat0=c  1        * speed maxi
       a=dat1  a        * adresse 1er niveau
       ad1ex            * @ entier binaire
       d0=a             * @ 1er niveau
       lchex   02a70    * prologue entier binaire
       a=dat1  a
       ?a=c    a        * entier binaire ?
       goyes   binary
errtyp lchex   0202     * erreur : bad type
       govlng  #111b9   * a=c a sethex GETPTR
*                         Errjmp
```

```
binary
       d1=d1+  10       * @ valeur entier binaire
       c=dat1  a        * adresse départ recherche
       cd0ex            * @ position départ recherche
       cd1ex
       rstk=c           * sauve adresse 1er niveau
       d1=d1+  5        * passe au 2ième niveau
       a=dat1  a        * adresse chaîne
       d1=a             * @ sur chaîne
       a=dat1  a        * prologue
       lchex   02a4e    * prologue chaîne
       ?a#c    a        * différent de chaîne ?
       goyes   errtyp
       d1=d1+  5        * @ longueur chaîne
       a=dat1  a        * long. chaîne + 5
       d1=d1+  5        * @ début chaîne
       a=a-con a,5      * long. chaîne en nibs
       lc(4)   32       * maxi long. chaîne
       ?c<=a   a        * long. = min(len,32)
       goyes   ok       *
       c=a     a        *
ok     ad1ex            * adresse début caractères
       a=a+c   a        * adresse après dernier car
       d1=a             * @ après dernier caract.
       csrb             * nombre de caractères
       c=c-1   a        * pour compteur
       d=c     a        * compteur caractères
       csrc             * len - 1 -> c(s)
       lcasc   '9'

loop   bsl     w        * conversion
       d1=d1-  2        * ASCII -> HEXA
       a=dat1  b        * de la chaîne
       ?a<=c   b        * avec 1er code
       goyes   inf10    * en B(0) pour
       a=a-con b,7      * même ordre
inf10  b=a     p        * qu'en mémoire
       d=d-1   b        *
       gonc    loop
       p=c     15       * pour WP

loop2  a=dat0  b
       ?a#b    b        * teste si les 2 1ers nibs
       goyes   nxt      * en mémoire sont conformes
       a=dat0  wp       * si oui test si l'ensemble
       ?a=b    wp       * en mémoire est conforme
       goyes   found    * si oui alors trouvé
nxt    d0=d0+  1        * sinon adresse suivante
       gonc    loop2

found  ad0ex
       a=a+1   a        * adresse trouvée + 1
       c=rstk           * adresse 1er niveau
       d1=c             * restaure d1
       dat1=a  a        * valeur position + 1
       govlng  #125e1   * GETPTRLOOP
fin
       con(5)  #02f90   * SEMI
```

Voici un petit programme permettant de rechercher toutes les occurences d'une suite de codes dans la mémoire:

LFIND

```
«
  HEX { } SWAP #0h
  WHILE DUP # 1h ≠
  REPEAT FIND ROT OVER
    1 - + 3 ROLLD
  END
»
```

Il suffit de founir la chaîne des codes à chercher et le programme renvoie la liste des éventuelles adresses (adresse # 0h à éliminer).

Guy Toublanc (276)

CHOC EN RETOUR
FONCTIONS GRAPHIQUES

NDLR : Le gros article de Paul Jebeily "Graphisme et animations", paru dans le précédent numéro, nous était parvenu sous forme manuscrite. Il était donc malheureusement prévisible que quelques erreurs ou fautes de frappe nous aient échappé, même après vérifications. Voici donc un petit correctif :

Dans MENUER du répertoire HOME :
Remplacer MAITK par WAITK

Dans le répertoire OBJETS du chemin { HOME GRAPH OBJETS } :

- Dans INIT : 2ème ligne, remplacer
  64 CF par 64 SF

- Dans FAM1ROT : 7ème ligne, remplacer
  + im0 LCD→ par im0 →LCD

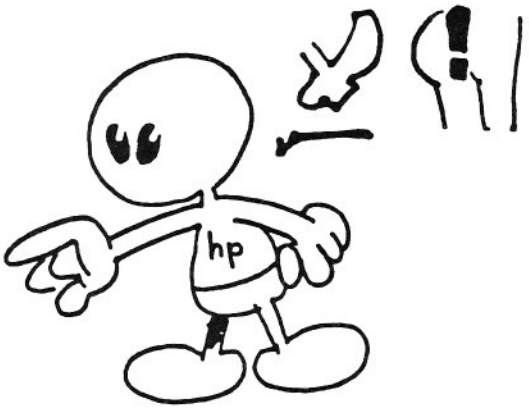
- Dans LBLOCK : 6ème et 11ème ligne, remplacer
  FOR u u par FOR u
(ou supprimer un "u" aux ligne 6 et 11).

- Dans PARP : 1ère ligne, remplacer
  DUP 2 ≥ par DUP2 ≥

- Dans TEST2 : 1ère ligne, remplacer
  « 4 SPEED ... 5 ROLLD SWAP 6 par
  « SPEED ... 5 ROLLD SWAP DROP 2 SWAP 6

Les listes VARS (et non VARS:) sont obtenues par l'instruction VARS. Elles ne sont pas à stocker.

Paul Jebeily (571)



HP48

G. Toublanc	Le kit de connection HP48GX	*10*
G. Toublanc	Présentation de *CAT-48*	*10*
R. Pulluard	Calcul du numéro de semaine	*11*
G. Toublanc	Calculs hyperboliques	*12*
G. Toublanc	Trucs et astuces	*13*
P. De Sacy	Points d'entrées correspondant SX & GX	*15*
P. De Sacy	Cartes TDS sur le port 2 (48GX)	*16*
P. De Sacy	Changement de répertoire (Acte II)	*18*
	Le coin des codes	*30*

NOUVEAU KIT DE CONNECTION

Comme cela était annoncé, HP a livré aux acquéreurs de HP48G/X (achetées entre le 1/9/93 et le 30/10/93) le kit de connection :

HP 82208C Serial Interface Kit
HP 48 to DOS-compatible PCs

Ce kit se compose comme les précédents de deux parties :

- Côté matériel: un cable de connection qui est en fait celui du HP100LX avec un adaptateur pour HP48 ce qui peut être une chose appréciable pour les futurs possesseurs de HP100LX.

- Coté logiciel: une disquette qui se différencie de celle des précédents kits par un ensemble de programmes pour PC et permettant d'utiliser sous Windows des objets graphiques HP48 et seulement dans le sens HP48→PC. Ceci peut être utile par exemple pour des publications relatives aux HP48.

Pour le reste, pas de nouveautés si ce n'est que les utilitaires HP (INPRT, USAG etc..) sont bien compatibles pour les HP48 séries S et G, donc une très bonne chose.

Pour les communications HP48←→PC ce sont toujours les programmes avec le protocole Kermit, donc rien pour Xmodem.

Si ce kit a vu le jour au moment de promouvoir la vente des HP48G/X, il reste aussi utile pour les HP48S/X. D'ailleurs le programme d'installation propose deux options S ou G. L'option S permet de charger des programmes utilisables sur HP48S/X mais devenus inutiles avec les toutes nouvelles instructions des HP48G/X.

Pour ceux qui possèdent les anciennes disquettes et en particulier celle de la précédente opération *Back to School* et qui n'ont pas la nécessité d'utiliser sur PC des objets graphiques HP48, je ne vois un intérêt que pour les utilitaires HP devenus compatibles HP48G/X.

Guy Toublanc (276)

LITTERATURE POUR HP48

CAT-48

Notre ami Robert Pulluard est un lecteur assidu des journaux de liaison des clubs HP et aussi un auteur idéal pour répertorier les programmes parus dans ces journaux. Il a donc réalisé un catalogue de programmes HP48 et cela depuis avril 90 (mon exemplaire contient 528 entrées). Chaque programme est décrit méthodiquement avec les références qui s'imposent. Les publications sont essentiellement (ou presque) européennes mais vont de la Suède à l'Espagne. Cela permet d'avoir un aperçu de ce qui se fait parmi les clubs qui publient et jouent donc un rôle pédagogique dans le milieu HP48.

C'est un catalogue rédigé en anglais et réalisé avec tout le sérieux de Robert. Aux descriptions on peut ajouter :

- Un index des auteurs.
- Un index d'équivalences lorsque les programmes ont été publiés par plus d'un club.
- Une liste d'adresses des publications.
- Un index alphabétique et détaillé par thèmes.
- Un index regroupant les programmes par thèmes généraux.

A partir de ce catalogue on pourrait établir des statistiques sur les thèmes abordés et une dure réalité : le taux de participation dans les clubs.

On peut se procurer cette publication contre un chèque de 40 F français adressé à :

Robert Pulluard
Lijsterlaan 31
2665 TH BLEISWIJK
Pays-Bas

Ce catalogue est actualisé périodiquement et cette entreprise mérite notre reconnaissance.

Guy Toublanc (276)

NUMERO DE SEMAINE

Le petit programme pour HP48 présenté ici a pour but de fournir, à partir d'une date introduite au niveau 1 de la pile opérationnelle, le numéro de la semaine correspondante dans l'année, selon les normes ISO (Organisation Internationale de Standardisation), et notamment le document ISO 8601, section 3.17, qui précise que la semaine commence un lundi et que la première semaine de l'année (semaine No.1) est celle qui contient un jeudi.

Ce numéro de semaine dans l'année se rencontre dans certains agendas (pas toujours selon les normes ISO), et est utilisé dans certains pays, tels que la Suède. Il peut donc s'avérer intéressant de mettre à profit les fonctions sur les dates, dont est pourvue la HP48, pour disposer de cette information d'une manière simple et rapide.

Le programme NSEM accepte comme argument :
- soit une date au format "européen" (jj,mmaaaa), si l'indicateur binaire -42 est armé ("set"),
- soit une date au format américain (mm,jjaaaa), si l'indicateur binaire -42 est désarmé ("clear"),
- soit 0 (zéro), auquel cas c'est la date courante, prise dans l'horloge interne de la HP48, qui est utilisée comme argument.

L'algorithme utilisé est relativement simple : à partir de la date argument on détermine d'abord la position du jour dans l'année. C'est le rôle des trois premières lignes dans le listage du programme (pour être rigoureux il faut tout de même préciser que la valeur calculée est n+2, n étant ici le numéro du jour en question). C'est dans cette partie qu'est détectée la présence éventuelle d'une valeur nulle comme argument (commande IFT au lieu d'une structure [IF...THEN...END]) : dans ce cas, l'instruction DATE prend la date courante dans l'horloge interne de la HP48.

Les trois lignes suivantes déterminent, par accès à la chaîne de temps ("timestring"), le jour de semaine correspondant au premier janvier de l'année. C'est en fait le numéro *p* du jour dans la semaine qui nous intéresse ici, et la méthode utilisée consiste à convertir bêtement le nom anglosaxon de ce jour, tel que fourni par la chaîne, au moyen d'une liste de correspondance.

Ceci fait, il ne reste plus qu'à combiner les résultats partiels obtenus, à savoir *(n+2)* et *p*, en calculant :

r = (p-4) MOD 7

puis en prenant la partie entière de:

s = ((n+2) + r)/7

C'est ce que fait la dernière ligne du listage.

NSEM
Cksum: # 9300h 167 octets

```
«
  DUP  NOT  { DROP  DATE }  IFT
  -2  ALOG  MOD  LASTARG  ROT  +
  1  +  DUP  ROT  DDAYS  3  +
  { "MON"  "TUE"  "WED"  "THU"
    "FRI"  "SAT"  "SUN" }  ROT
  0  TSTR  1  3  SUB  POS
  4  -  7  MOD  +  7  /  FLOOR
»
```

Utilisation du programme NSEM :

```
1:date        → NSEM →      1:semaine
1:0           → NSEM →      1:sem_courante
```

Exemple:

```
1:6,111993    → NSEM →      1:44
```

Quelques remarques additionnelles :

Les trois premières lignes du programme ont été optimisées et sont donc quelque peu cryptiques : le mieux est de les exécuter pas à pas pour comprendre leur fonctionnement exact. Le reste du programme n'a pas été optimisé, cet exercice paraissant illusoire quand on est confronté au gros morceau que représente la liste des noms de jour de semaine. Pour raccourcir encore ce programme, la seule solution me semble être de se débarrasser de cette liste et de faire appel à une autre méthode pour déterminer la valeur "*p*" du premier jour de l'année. Une possiblité est évidemment de faire tout simplement appel à une routine DOW (Day Of Week) si vous en avez une à votre disposition ou, moins direct, de passer par la date julienne (si vous avez des programmes d'astronomie, vous devez alors avoir une telle routine parmi vos utilitaires).

Quoiqu'il en soit, si un lecteur connaît le moyen de résoudre ce problème du jour de semaine de manière simple et élégante, sans faire appel à un sous-programme extérieur, qu'il n'hésite pas à le faire connaître !

Robert PULLUARD (561)

SOS :

41 APPELLE 48

Il est courant que des utilisateurs de machines HP soient plus ou moins désarmés en passant d'un modèle à un autre plus évolué. Ici c'est le cas d'un utilisateur de machine qui est passé d'une machine HP antérieure au modèle HP-41, à une HP-41 puis une HP48. Pour lui venir en aide, il nous demande de traduire le programme HP-41 ci-dessous en programme HP48 :

```
01 LBL HYPERB     16 X^2          31 STOP
02 RCL 03         17 RCL 04       32 RCL 04
03 X^2            18 *            33 1
04 RCL 01         19 RCL 03       34 +
05 X^2            20 X^2          35 SQRT
06 -              21 -            36 RCL 05
07 RCL 02         22 RCL 04       37 *
08 X^2            23 /            38 STO 07
09 RCL 00         24 SQRT         39 STOP
10 X^2            25 STO 05       40 RCL 07
11 -              26 STOP         41 2
12 /              27 RCL 05       42 *
13 STO 04         28 2            43 STO 08
14 STOP           29 *            44 STOP
15 RCL 02         30 STO 06       45 END
```

Celui-ci doit permettre de tracer des hyperboles pour la réalisation d'un cadran solaire.

Au vu de ce listing on voit que les possibilités d'une HP-41 ne sont pas utilisées de manière très optimale, qu'il s'agit plutôt d'une programmation utilisée sur une machine plus ancienne où les commandes d'entrée et de sortie n'existaient pas car les données doivent être stockées dans les registres avant le lancement du programme et les résultat sont lus simplement à chaque arrêt du programme qu'il faut relancer par séquences.

Il s'agit donc à partir des données :

x_1 y_1 x_2 y_2

de trouver les valeurs:

$$K^2 \text{ tel que } K^2 = \frac{y_2^2 - y_1^2}{x_2^2 - x_1^2}$$

$$a \text{ tel que } a = \sqrt{((K^2 x_2^2 - y_2^2)/K^2)}$$

$$c \text{ tel que } c = a\sqrt{(1 + K^2)}$$

a et *c* étant respectivement la distance de l'origine des axes au sommet de l'hyperbole et la distance du foyer à cette origine.

Devant les difficultés d'adaptation de notre programmeur la méthode à adopter est celle d'une programmation de type algébrique en reproduisant les formules. Ici je ne chercherai pas à optimiser mais à rester très simple en traduisant le programme HP-41. Je simplifierai quand même la formule de *a* :

$$a = \sqrt{(x_2^2 - y_2^2/K^2)}$$

Voici le programme HP-48 :

HYPERB
Cksum #E293h 394 octets

```
« "x1,y1,x2,y2 ?"   arguments demandés
 ""                   chaine vide pour l'entrée
                      par défaut
 INPUT                demande d'entrée de valeurs
 OBJ→               extrait les objets entrés
                      c'est-à-dire x1,y1,x2,y2
 0                    valeur pour K^2
 0                    valeur pour a
 0                    valeur pour c
                      ces 3 valeurs ne sont
                      utilisées que pour la
                      création de varables locales
 RCLF                 rapelle la liste contenant
                      les flags système, liste qui
                      sera sauvegardée dans 'F'
         ces 8 valeurs de x1 à RCFL vont être
         sauvegardées dans les 8 variables locales
         ci-dessous : x1 à F et cmprises entre →
         qui indique que les valeurs avant sont
         stockées dans les variables correspon-
         dantes (dans le même ordre) et '«' qui
         marque le début de la séquence de
         programme avec ces variables locales
→ x1 y1 y2 y2 K2 a c F
  « '(y2^2-y1^2)/(x2^2-x1^2)' EVAL
         formule de K^2 puis valeur numérique
         fournie par EVAL
    'K2' STO          sauvegarde de K^2 dans la
                      variable locale (v.l) 'K2'
    '√(x2^2-y2^2/K2)' EVAL
         formule de a puis valeur numérique
    'a' STO           sauvegarde de a dans la v.l 'a'
    'a*√(1+K2)' EVAL
         formule de c puis valeur numérique
    'c' STO           sauvegarde de a dans la v.l 'c'
    4 FIX             pour l'affichage des résultats
    "K2 = " K2 +      chaine "K2 = n.nnnn" pour
                      affichage
    "                 chaine commençant par NEWLINE
    a  = " a +        avant 'a' : "a = n.nnnn"
```

```
   +                ajoute les 2 chaînes qui sont
                    séparées par NEWLINE
   "
   c  = " c +       même principe pour "c = n.nnnn"
   +                somme des 3 chaines avec 2
                    NEWLINE
   "
   2a = "
   a 2 * +          "2a = n.nnnn"
   +                somme des 4 chaines et 3 NEWL.
   "
   2c = "
   c 2 * +          "2c = n.nnnn"
   +                ici nous avons 5 chaines
                    assemblées avec 4 NEWLINE
                    intercalés
   F STOF           on restaure les flags système
 »                  fin de l'environnement des
                    variables locales qui sont
                    détruites
 CLLCD              efface l'écran
 1 DISP             affiche à partir de la ligne
                    1 la chaine qui va donner 5
                    ligne de résultats
 7 FREEZE           gèle l'affichage de tout
                    l'écran
»
```

Exemple:

pour

x_1 = 40,52
y_1 = 29,31
x_2 = 127,72
y_2 = 246,76

lancer HYPERB

A la demande des arguments, taper sur une seule ligne ces 4 valeurs séparées par un espace et valider avec [ENTER]

l'affichage fera apparaître :

```
K2 = 4.0920
a  = 37.8408
c  = 85.3894
2a = 75.6817
2c = 170.7787
```

On voit qu'avec ce type de programmation cela est même plus simple qu'avec le programme HP-41 et aussi plus compréhensible car les formules sont traduites naturellement. Nous n'avons pas utilisé de mouvements de pile, nous contentant de stocker les résultats dans des variables locales qui sont ici l'équivalent des registres de la HP-41 mais rendent la programmation plus facile car nous pouvons leur donner des noms plus évocateurs que R00, R01 etc... Le rappel des flags système est destiné à pouvoir restaurer les modes de la HP-48 avant le lancement du programme (par exemple si l'on était en mode STD au lieu de FIX 4). Cela évite de le faire manuellement.

Aux possesseurs de HP-41 et HP48 (ou HP28) je conseille de traduire leurs anciens programmes et de ne pas se contenter d'utiliser ceux-ci sur une HP48 avec émulateur HP-41. Leur effort sera payé par un gain de rapidité et de confort.

Ces adaptations seront les bienvenues dans les colonnes de JPC car les programmes résultants résoudront certainement des problèmes pratiques à la portée de beaucoup.

Guy Toublanc (276)

TRUCS, ASTUCES ET PETITS UTILITAIRES

Je ne comprends pas pourquoi cette rubrique n'a été alimentée que par moi. Elle devrait permettre à beaucoup de participer car elle ne doit recevoir que des petits travaux, parfois même que des idées. Elle est accessible à tous, une simple feuille de papier et une dizaine de lignes écrites à la main sont un moyen souvent suffisant pour faire part de vos...
...trouvailles simples mais utiles que ce soit côté matériel ou logiciel.
...éclairs de génie ou pas qui simplifient la vie.
...corrections des petites erreurs des autres.
...découvertes de sysevals.
...petits utilitaires en User Rpl et même en System Rpl ou assembleur (pourquoi pas ?).

Ici on ne reçoit pas que les programmes. Les idées comptent autant.

Avec l'arrivée des HP48G(X) il y a tout un terrain qui est vierge. Les menus déroulants et boites de dialogues c'est bien, mais je préfère un raccourci clavier. Donc de ce côté il y a des découvertes à faire. De même il serait utile de signaler les changements entre HP-48G(X) et HP48S(X) au niveau de l'accès aux commandes de façon à éviter les tatonnements pour ceux qui avaient des habitudes et qui ont des problèmes d'adaptation. Il y encore beaucoup de petites questions qui demandent des réponses simples.

La rubrique des lecteurs est à la disposition de tous pour poser des problèmes. S'il y a réponse cela pourrait très bien se faire ici.

Correction de 2 erreurs

Auto-correction de deux de mes bétises.

Pour récupérer les mnémoniques HP correspondant à des adreses en Rom je proposais un mini programme MNEMO dans JPC 88. Je m'étonne de ne pas avoir provoqué de réactions car ce programme est inutile. En effet la librairie ←RPL→ fait cela avec E⟷A et comme l'indique le label dans les sens mnémonique → adresse et adresse → mnémonique. Donc une grosse faute d'étourderie de ma part pour ne pas avoir lu complètement le mode d'emploi de *RPL Toolkit*. Débarassez vous de MNEMO.

Deuxième faute à propos de l'article *Transferts plus rapides sur HP-48G(X)* (JPC 89 page 8) il y a plus simple que ce que j'avais indiqué. Côté HP48 pour l'envoi sous protocole XMODEM il suffit que le IOPAR soit configuré avec 9600 Bauds. XSEND ou XRECEV ne servant qu'en mode XMODEM pour envoyer ou recevoir des objets HP48 en mode binaire. Donc il est inutile de passer par le menu déroulant pour configurer et si l'on fait une transmission par cable à 9600 Bauds on peut alternativement utiliser XMODEM et KERMIT, ce dernier protocole s'utilisant avec SEND et RECEV. La connaissace de la boite de dialogue n'est peut-être pas complètement inutile pour connaître les paramètres de XMODEM sur HP48 mais elle ne doit être considérée que comme un pense-bête en cas de perte de mémoire.

Compatibilité Kermit et Xmodem

Un objet transféré depuis une HP48G(X) sous un des protocoles Kermit ou Xmodem est-il récupérable dans un transfert en sens inverse et sous l'autre protocole ? Le problème se pose si vous récupérez des objets sauvegardés par un tiers sur disquette et dont vous ignorez le protocole utilisé.

Tout d'abord si on transfère en mode binaire et sous les deux protocoles un objet HP48 on constate que l'occupation dans le PC n'est pas la même.

Ici, pour le protocole Kermit, n'est examiné que le transfert en mode binaire.

- HP48/kermit → PC
  PC/Xmodem → HP-48 : compatibilité

- HP48/Xmodem → PC
  PC/Kermit → HP-48 : incompatibilité apparente

Dans ce dernier cas par exemple un objet binaire de 819 octets et autre qu'une chaîne revient sous forme de chaîne de 901 octets avec l'entête Kermit HPHP-M. Donc il y a un problème. Dans ce cas il existe deux solutions:

- Réitérer le transfert mais sous Xmodem si vous en avez encore la possibilité.
- Extraire de la chaîne l'objet d'origine. C'est ce que fait le petit programme ci-dessous.

syntaxe:

niveau 1 : chaîne XMOD→KER → objet récupéré

XMOD→KER
62.5 octets cksum # BB7Bh

```
::
 CK1NoBlame CK&DISPATCH0 str
 ::
CODE
     gosbvl   =SAVPTR
     a=dat1   a            * adresse chaîne
     d1=a                   * @ sur chaîne
     d1=d1+   5            * @ sur longueur
     c=dat1   a            * longueur totale
     d1=d1+   5            * départ codes chaîne
     c=c-con  a,6          * longueur en quartets
     d=c      a            * compteur quartets ch.
     lc(5)    =PRLG        * début à l'adr. prolog.
     goc      out          * sortie si chaîne nulle
loop a=dat1   a            * pope 5 quartets
     d0=a                  * @ sur adr en Rom ?
     a=dat0   a            * pope PRLG si adr prol.
     ?a#c     a            * est-ce PRLG ?
     goyes    cont         * non alors on continue
     ad1ex                 * oui alors adr. objet
     gonc     out          * et sortie
cont d1=d1+   1            * @ quartet suivant
     d=d-1    a            * actualise compteur
     gonc     loop         * non fin: + 1 tour
out  govlng   =GPOverWrALp * GTPTR + dat1=a + Loop

ENDCODE
  TOTEMPOB
 ;
;
```

XMOD→KER utile aussi pour HP-48S(X) et encore pour HP48G(X)

Si un objet binaire a été transféré dans votre HP48 sous Kermit et en mode Ascii vous récupérerez une chaîne. Il se peut très bien que vous n'ayez plus la possibilité de refaire le transfert (par exemple le copain étourdi qui vous a gratifié de ce cadeau bizarre n'est plus là) alors XMOD→KER est tout indiqué pour vous

restituer l'objet perdu dans la chaîne (voir ci-dessus la syntaxe).

Xmodem pour HP-48S(X)

Je signale que les possesseurs de HP48S(X) ont la possibilité de transférer des objets entre HP48 et PC à l'aide de la librairie XYMODEM.LIB des goodies disk #7 (répertoire UTILS). Une documentation est présente. Pour ma part je l'utilise avec le setup par défaut (de la librairie) et de la même manière qu'avec Xmodem de la HP48G(X) à la différence que l'objet est sur la pile, soit à l'envoi où à la réception. Voir l'article *transferts de fichiers* (JPC-89) et ci-dessus pour la rectification.

C'est tout pours cette fois, en espérant que mon appel du début de cet article aura été bien entendu et compris.

Guy Toublanc (276)

POINTS D'ENTREES SX OU GX ?

Vous voulez écrire des programmes assembleur compatible entre GX et SX... voici de quoi vous aider.

La liste qui suit se compose de quatre colonnes :

- la première contient les diverses informations des 48 que l'on trouve en RAM (Le libellé étant celui que nous pouvons trouver dans *Voyage au centre de la HP48*).

- les deuxième et troisième contiennent les adresses en RAM pour, respectivement, la SX et la GX de ces informations.

- la quatrième contient les adresse dans la ROM des deux machines ou l'on trouve l'adresse des colonnes deux et trois en fonction du modèle.

Il suffit donc de remplacer, par exemple :

```
DO=(5) #7050E
A=DATO A          pour la SX

DO=(5) #8068D
A=DATO A          pour la GX
```

par

```
DO=(5) #12D21
A=DATO A
DO=A
A=DATO A          pour la SX et la GX
```

Toutes les adresses de la quatrième colonne pourraient changer dans une future version de la 48, je ne peux rien garantir !!!!

Toutefois n'ayant pas changé entre les version D et M (versions de mes 48), il y a peu de chance qu'elles change tout d'un coup.

Tout programme écrit de cette facon a plus de chance de fonctionner dans les versions future des GX... car si le bugs de la GX faisant que le port 33 d'une carte 4Mo n'est pas accessible est corrigée, la plupart des adresses de la GX risquent d'etre décalées...

Libellé	SX	GX	ROM SX&GX
CMOS word	70000	80000	01192
préparation affichage	7008C	80092	
cartes enfichées	700DB	800E1	
données buffer	701FC	80319	
Infos carte 1	7042C	8052F	09B84
Infos carte 2	70437	80531	
adr. fin port 0	7044D	80536	
adr. fin port 1	70452	8053B	
adr. fin port 2	70457	80540	
sauvegarde pour les INT	7045C	805DB	sur(4) 00064
masque de sortie clavier	704C3	80642	0188F
vitesse machine	704D6	80655	017CA
keystart	704EA	80669	047DF
keyend	704EB	8066A	04725
buffer clavier	704EC	8066B	
adr. bitmap ecran	7050E	8068D	12D21
marge droite	70513	80692	12D46
adr. bitmap menu	70516	80695	sur(4) 01CA3
marge gauche	7051B	8069A	12F73
adr. GROB menu	70551	806D0	1264C
adr. GROB pile	70556	806D5	12516
adr. GROB courant	7055B	806DA	01F79
adr. GROB non visible	70560	806DF	12F11
adr. GROB graphique	70565	806E4	1266C

adr. debut objet temp.	7056A	806E9	060CB
adr. fin objet temp.	7056F	806EE	060DC
adr. deb.mem libre (sauv B)	70574	806F3	05FBC
adr. fin mem libre (sauvD1)	70579	806F8	034F0
adr. ligne de commande	7057E	806FD	03159
adr. pile d'undo&var locale	70583	80702	04E68
adr. données boucles int.	70588	80707	04E73
adr. environment temp.	7058D	8070C	04B17
adr. répertoire HOME	70592	80711	08DA6
adr. port 0	70597	80716	0A0B9
adr. dir courant	7059C	8071B	086D9
adr. stock dir ptr	705A1	80720	07A2E
adr. touches utilisateur	705A6	80725	09739
adr. alarmes	705AB	8072A	01372
adr. objet à executer(s.D0)	705B0	8072F	05FE1
pile de last	705BA	80739	112F3
adr. grand entier	705D3	80752	15ADE
pile de commande	705DD	8075C	18249
adr. message d'erreur	70600	8077F	04E0E
adr. app display	70605	80784	38C0F
adr. app key	7060A	80789	38C3F
adr. app exit cond	7060F	8078E	38C6F
adr. app error	70614	80793	38C9F
menu courant	7061E	8079D	4189B
dernier menu	70623	807A2	419EB
unshift menu key rout	70637	807B6	4191B
left shift menu key rout	7063C	807BB	4194B
right shift menu key rout	70641	807C0	4196B
ReviewKey	70646	807C5	4198B
adr. last menu VAR	7064B	807CA	41A0B
exit action	70655	807D4	419CB
numéro de commande	7065A	807D9	18814
dernier token RPL	7065F	807DE	40A76
adr. fin de RAM	70669	807E8	0A472
mémoire libre (sauveg. D)	7066E	807ED	0553C
prochaine erreur à aff.	70673	807F2	04CF3
flag ATTN	70679	807F7	01888
FIRSTPROC	7067E	807FC	
drapeaux internes	70683	80801	53B4A
taille de la pile	7069F	8081D	113C9
racine nombre aleatoire	706A4	80822	2AFD2
indicateurs	706C3	80841	01426
drapeaux systeme	706C5	80843	537E3
drapeaux utilisateur	706D5	80853	537D7
caractere à gauche	706E7	80865	11439
nbr de lignes	706EC	8086A	11429
N° 1ere ligne	706F6	80874	113E9

dernier numéro d'erreur	706FF	8087D	13FF9
pos du curseur ds ligne	70704	80882	111B9
pos Vert. à l'ecran	70709	80887	13F4E
pos Horiz. à l'ecran	7070E	8088C	13C22
curseur visible	70710	8088E	13D72
code ascii car sous curseur	70711	8088F	13BE2
GROB car sous curseur	70713	80891	13D64
pos Horiz curseur en pixels	7073B	808B9	13C36
pos Vert curseur en pixels	70740	808BE	13C18
sauvegardes pour garb coll	70750	808CE	06142
indicateur mode Alpha	70793	80911	11508
indicateur mode Prg	70794	80912	11518
offset menu courant	707C9	80947	4184F
dernier offset menu	707D3	8099E	41875
nb de lib presentes	707D9	809A3	0843C

Si certains de vos anciens programmes sur SX utilisent par exemple les drapeaux de la 48 comme suit :

```
D0=(5)  #706D4  (@ des drapeaux -60 à -64)
A=DAT0  B
?ABIT=1 0
GOYES   [Alpha]lock
```

L'adresse #706D4 n'ayant pas de pointeur en Rom, il est possible de faire :

```
D0=(5)  #537E3  (@ de l'@ des drapeaux système)
A=DAT0  A
D0=A
D0=D0+  15      (@ des drapeaux -60 à -64)
A=DAT0  B
?ABIT=1 0
GOYES   [Alpha]lock
```

Pierre Silvestre de Sacy (572)

EXTENSION DU PORT 2

Ce programme permet d'utiliser les cartes TDS de 256 et 512 ko dans le port 2 de la HP48GX.

La librarie SWITCH marche pour les GX lorsque la carte et en port 1, mais il n'est pas possible de l'utiliser dans le port 2.

Pour pouvoir changer de banc, les cartes TDS comportent un circuit qui controle les accès en lecture de la HP sur la carte et qui est chargée de changer le banc accessible.

Quand on lit sur le champ B aux adresses :

```
#xFFFA #xFFFE #xFFFE #xFFFE
#xFFFC #xFFFE #xFFFC #xFFFC #x0010
```
→ on passe sur le banc 1

```
#xFFFA #xFFFE #xFFFE #xFFFE
#xFFFC #xFFFE #xFFFC #xFFFE #x0010
```
→ on passe sur le banc 2

```
#xFFFA #xFFFE #xFFFE #xFFFE
#xFFFC #xFFFE #xFFFE #xFFFE #xFFFC #x0010
```
→ on passe sur le banc 3

```
#xFFFA #xFFFE #xFFFE #xFFFE
#xFFFC #xFFFE #xFFFE #xFFFC #x0010
```
→ on passe sur le banc 4

Où le x correspond au quartet de poids fort de l'adresse de fin de carte (x=B pour une carte allant de #80000 à #BFFFF, x=F pour une carte allant de #C0000 à #FFFFF).

Ce programme marche comme le BANKN de la librairie SWITCH, sauf qu'il fait d'office un arrêt systeme apres le changement de banc.

De plus il peut être stocké dans la carte sur laquelle s'effectue le Switch car la programme est recopié en RAM avant d'être exécuté... Attention, il vaut mieux avoir une copie de ce programme dans chaque banc, ou l'on risque de ne plus pouvoir revenir dans les autres bancs.

BANKN2
(2: numero de port, 1: numero de banc)

```
::
  CK2&Dispatch
  # 11
   ::
     COERCESWAP
     COERCESWAP
     '
      ::
        OVER
        PORTSTATUS
        5UNROLL
        3DROP
        NOT
        IT
          ::
            3DROP
            #1=
            ITE
            $ "1"
            $ "2"
            $ "No card in port "
            SWAP
            &$
            DOCLLCD
            DO$EXIT
          ;
CODE
        D1=D1+  10
        GOSBVL  =SAVPTR
        D1=D1-  10
        A=DAT1  A       * @ Extended Pointer
        D0=A
        D0=D0+  5
        A=0 W
        A=DAT0  10
        ?A#0    W       * si nul > pas de carte
        GOYES   CART
        GOTO    FIN

CART    R1=A.F  A       * @ debut port
        D0=D0+  5
        A=DAT0  A       * @ routine deconfig
        ST=0 0
        ?A=0 A
        GOYES   CARTE.PORT1 * si nul > port 1
        ST=1    0       * port 2
CARTE.PORT1
        D1=D1+  5
        A=DAT1  A
        D1=A
        D1=D1+  5
        A=DAT1  A       * numero banc
        R2=A.F  A
        GOSUB   SAUTE.PROG
                        * programme
                        * de switching
        GOSBVL  #01115  * interdit interruptions
        ?ST=0   0
        GOYES   NODECALAGE
        P= 0
        GOSBVL  #7073F  * si port 2 déconfiguré
NODECALAGE

        D0=(5)  #FFFFA  * première
        C=DAT0  B       * partie
        D0=D0+  4       * de switching
        C=DAT0  B       * commune
        C=DAT0  B       * a tous
        C=DAT0  B       * les bancs
        D0=D0-  2
        C=DAT0  B
        D0=D0+  2
        C=DAT0  B

        A=R2.F  A
        A=A-1   A
        A=A-1   A
        GONC    BANC2
```

```
       D0=D0-  2       * banc 1
       C=DAT0  B
       C=DAT0  B
       GOTO    FIN_SWITCH

BANC2  A=A-1   A
       GONC    BANC3
       D0=D0-  2       * banc 2
       C=DAT0  B
       D0=D0+  2
       C=DAT0  B
       GOTO    FIN_SWITCH

BANC3  A=A-1   A
       GONC    BANC4
       C=DAT0  B       * banc 3
       C=DAT0  B
       D0=D0-  2
       C=DAT0  B
       GOTO    FIN_SWITCH

BANC4  A=A-1   A
       GONC    FIN.B   * banc incorrect
       C=DAT0  B       * banc 4
       D0=D0-  2
       C=DAT0  B
FIN_SWITCH
       D0=(4)  #0010
       C=DAT0  B
FIN.B
       ?ST=0   0
       GOYES   NOINSTALL
       P= 1            * si port 2 reconfiguré
       GOSBVL  #7073F
NOINSTALL
       GOSBVL  #010E5  * autorise interruptions
FIN
       GOVLNG  =GETPTRLOOP * retour RPL

SAUTE.PROG
       D1=(5)  #800F5  * recopie le
       C=RSTK          * prog dans buffer
       D0=C            * de transfert
       LCHEX   000CE
       GOSBVL  =MOVEDOWN (#0670C)
       GOVLNG  #800F5  * puis y saute

ENDCODE
        DROP
        PTR    21B4E   * Arrêt Système
      ;
    TOTEMPOB           * Copie en RAM
    EVAL
  ;
;
```

Pierre Silvestre de Sacy (572)

GO : LE RETOUR !

Voici une nouvelle version du programme GO paru dans le JPC du mois de Février.

Je rapelle que ce programme recherche dans l'arborescence une variable. Si elle la trouve, le répertoire contenant cette variable devient courant, sauf si cette variable est un répertoire auquel cas c'est lui qui devient le répertoire courant...

Cette version fonctionne sur SX et sur GX, de plus elle ne bugge plus comme la première qui ne fonctionnait pas quand la variable recherchée était dans le repertoire HOME. Maintenant le recherche s'effectue en partant du répertoire courant et en remontant les répertoires tant que l'on n'a pas trouvé la variable dans tous les répertoires fils de celui dans lequel on est, jusqu'a ce que l'on ait atteint le répertoire HOME.

```
::
  CK1&Dispatch
  SIX
    ::
CODE
       GOSBVL  =SAVPTR
       A=DAT1  A
       D1=A
       D1=D1+  5
       A=DAT1  W       * debut nom recherché
       R1=A
       C=0     A
       R2=C    A       * nombre de sous-repertoires
       D0=(5)  #086D9  * @ @ rep. courant
       A=DAT0  A
       D0=A
       A=DAT0  A
       R0=A.F  A       * @ rep. courant
BOUCLE D=0     A
       D=D+CON A,16    * constante
       D0=(5)  #08DA6  * @ @ rep. HOME
       C=DAT0  A
       D0=C
       C=DAT0  A       * @ rep. HOME
       D0=A
       D0=D0+  5
       ?C#A    A
       GOYES   NOTHOME
       ST=1 0          * on est dans HOME
       A=0 A
       A=DAT0  X
       B=A A           * saute
       A=A+A   A       * libs
       A=A+A   A       * attachées
       C=A A
```

```
        A=A+A   A
        A=A+C   A
        A=A+B   A
        A=A+CON A,3
        CDOEX
        A=A+C   A
        DO=A
        GOTO    CTHOME

NOTHOME                 * on est pas dans HOME
        ST=0    0
        A=A+CON A,8     * saute LA lib attachée
        DO=A

CTHOME  C=DATO  A       * offset dernier objet
        ?C#0    A
        GOYES   NOFIN
        GOTO    FIN     * plus d'autres objets

NOFIN   C=C+A   A

B1      DO=C
        A=0     A
        A=DATO  B       * taille du nom
        C=R1    B
        ?A#C    B
        GOYES   MV      * si les tailles sont #
        GOTO    BON?    * est-ce le bon ?

MV      A=A+A   A
        ?A=0    A       * si nom vide
        GOYES   S1
        A=A+CON A,2

S1                      * alors la taille n'est comptée
                        *  qu'une fois
        A=A+CON A,2
        CDOEX
        A=A+C   A
        DO=A
        A=DATO  A       * prologue de l'objet
        B=C     A
        LCHEX   02A96
        ?A=C    A
        GOYES   DIR     * si c'est un rep.
NOTADIR
        C=B     A

NF      DO=C
        DO=DO-  5
        A=DATO  A       * prochain objet ?
        ?A=0    A
        GOYES   FIN?
        CDOEX
        C=C-A   A
        GOTO    B1      * on recommence l'analyse

DIR     C=R2    A
        C=C+1   A       * on descend dedans
        R2=C    A
        DO=DO+  8
        A=DATO  A
        ?A=0    A       * rep. vide
        GOYES   NOTADIR
        CDOEX
        C=C+A   A
        GOTO    B1      * on analyse ce rep.

FIN?    C=R2    A
        ?C=0    A
        GOYES   FIN     * si dans rep. d'origine
        C=C-1   A       * sinon
        R2=C    A       * ce ne peut pas etre
        DO=DO-  15      * HOME donc
        A=0     A       * une seule lib attachée
        A=DATO  B
        A=A+A   A
        ?A=0    A       * nom vide
        GOYES   S3
        A=A+CON A,2

S3                      * taille une seule fois
        CDOEX
        C=C-A   A
        GOTO    NF      * on continue

FIN     A=RO.F  A       * rep. d'origine
        GOSBVL  #083D1  * UPDIR
        C=RO.F  A
        ?A#C    A       * si inchanger > HOME
        GOYES   UPDIR
        LCHEX   00204   * nom pas trouver
        A=C     A       * => Erreur
        GOSBVL  =GETPTR
        GOVLNG  =Errjmp

UPDIR   RO=A.F  A       * rep. d'au dessus
        GOTO    BOUCLE  * on recommence

BON?    ?A#0    B
        GOYES   PBN
        GOTO    BON     * si nom vide > trouve

PBN     ACEX    A
        C=C+C   A
        C=C+CON A,2
        ?C>D    A       * taille > 7
        GOYES   +7

-7      C=C-1   B
        P=C     0       * p = nombre de quartets du nom
        A=DATO  WP
        C=DAT1  WP
        ?A=C    WP
        GOYES   BON     * si meme
        A=0     A
```

```
        A=DAT0  B
        P=      0
        GOTO    MV      * ce n'etait pas celui-la

+7      AD0EX
        R3=A    A       * sauvegarde D0
        D0=A
        AD1EX
        R4=A    A       * sauvegarde D1
        D1=A

B3      C=C-CON A,16    * taille - 8
        B=C     A
        C=DAT0  W
        A=DAT1  W
        ?A#C    W
        GOYES   MV2.    * debut pareil ?
        D0=D0+  16
        D1=D1+  16
        C=B A
        ?C<=D   A       * toujours taille > 8 ?
        GOYES   B4
        GOTO    B3      * si oui

MV2.    GOTO    MV2     * ( saut trop long )

BON     P=      0
        A=0     A
        A=DAT0  B
        A=A+A   A
        ?A=0    A       * si nom vide
        GOYES   S2
        A=A+CON A,2

S2                      * taille une seule fois
        A=A+CON A,2
        CD0EX
        A=A+C   A
        D0=A
        A=DAT0  A
        B=C     A
        LCHEX   02A96
        ?A#C    A
        GOYES   MVD     * est-ce un rep.

DF      CD0EX           * rep. trouvé
        D0=(5)  #086D9
        A=DAT0  A
        D0=A
        DAT0=C  A       * devient rep. courant
        GOSBVL  =GETPTRLOOP * retour RPL
B4      C=C-1   A
        P=C     0       * p = nombre de quartets du nom
        A=DAT0  WP
        C=R3    A       * restaure D0
        D0=C
        C=DAT1  WP
        ?A=C    WP      * si bon nom
        GOYES   BON
MV2     C=R4    A
        D1=C            * restaure D1
        P=      0
        A=0     A
        A=DAT0  B
        GOTO    MV      * on continue

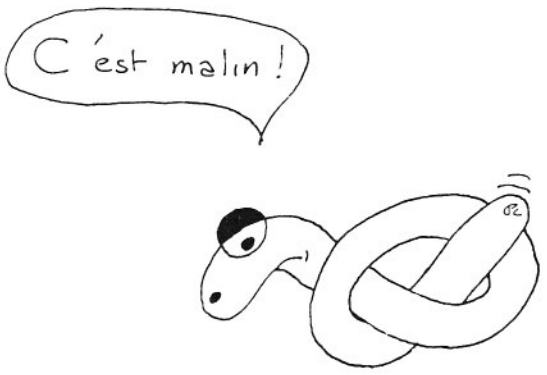
MVD     C=B     A
        D0=C
        C=R2    A
        ?C#0    A
        GOYES   NBDR    * on est pas dans le rep.
                        *  courant (celui pointé par
                        *  R0)
        ?ST=0   0
        GOYES   NBDR    * on est pas dans HOME
        D0=(5)  #08DA6
        A=DAT0  A
        D0=A
        A=DAT0  A
        D0=A            * c'est HOME
        GOTO    DF

NBDR    D0=D0-  5
        A=DAT0  A
        ?A=0    A       * derniere objet du rep. ?
        GOYES   BDR
        CD0EX
        C=C-A   A
        CD0EX           * prochain objet
        GOTO    NBDR

BDR     D0=D0-  13      * @ debut rep.
        GOTO    DF

ENDCODE
    DROP
  ;
;
```

Pierre Silvestre de Sacy (572)



HP95

G. Merlier	La configuration du Drive95	*22*
J. Belin	Conversion d'images pour HP95	*25*
J. Belin	Programmation du graphisme, correctif	*28*
	Le coin des codes	*30*

ESSAI DU DRIVE95 SUR LE HP95LX

NDLR : Bien que le DRIVE95 soit présent sur le marché (chez Maubert) depuis une assez longue période, ce produit revient dans l'actualité puisque il est commercialisé depuis peu de temps dans les FNAC. Cela est donc une bonne occasion d'en faire une petite présentation, puisque nous avions omis d'en parler en détail lors de sa sortie. Cet article peut paraitre un peu fastidieux pour les "pros" de l'IBM PC, mais nous l'avons reproduit tel quel, car ce produit est suceptible d'interesser des personnes n'ayant pas une grande experience du monde IBM : ceux qui n'ont pas d'autre ordinateur, ou ceux qui travaillent sur MacIntosh.

Certains ex-utilisateurs des calculatrices HP-41 ou de HP71 se sont trouvés attirés par la HP95LX, mais ne disposent pas à domicile d'un équipement important. Il se pose alors rapidement à eux le problème suivant : comment transférer sur le disque C: de la machine divers fichiers contenus sur des disquettes DOS, si l'on ne dispose pas d'un PC ?

Une solution peut être apportée grace à l'utilisation d'un lecteur de disquettes externe tel que le Drive95 de la société SPARCOM.

C'est un lecteur de disquettes 3"1/2 alimenté soit par batteries soit par adaptateur secteur et acceptant les disquettes de 720K et 1,44MB. Ses dimensions aproximatives sont : 18×13×5 cm. Il possède de plus d'avoir le protocole Kermit intégré dans sa Rom Interne. La vitesse de transfert maxi est de 9600 Bauds. Un cable série, fourni avec le lecteur, assure la liaison matérielle avec le HP95 (fiche femelle 4 broches coté HP95, fiche femelle 9 broches coté lecteur). Il est également fourni une disquette comprenant les logiciels nécessaires pour communiquer soit avec un HP95 soit avec une HP48, via Kermit. Son prix est de 2500F environ.

Une fois les connexions effectuées, la première tâche à accomplir consistera à installer un fichier nommé DRIVE95.SYS sur le disque C: du HP95LX. Pour ce faire, il suffira de suivre pas à pas les instructions du manuel livré avec le lecteur. Il est de plus indispensable de modifier (ou de créer) le fichier CONFIG.SYS de votre HP95, en lui incorporant la ligne suivante :

```
DEVICE=C:\DRIVE95.SYS
```

Une fois cet ensemble installé, le lecteur sera reconnu par le HP95 comme l'unité G:

Les lignes qui suivent permettront, du moins je l'espère, de réduire au minimum les tatonnements voire les plantages plutot décourageants pour le néophyte.

Installation du driver DRIVE95.SYS.

Les détenteurs de HP95 version US se contenteront de suivre strictement les séquences énoncées dans le manuel fourni avec le lecteur. Les séquences indiquées ci-dessous sont valables pour un clavier francisé. Supposons que la liaison matérielle entre HP95 et lecteur soit assurée; supposons également que toutes les données à conserver sont sauvegardées et que toutes les applications intégrées sont closes (`Menu,Quitte..Quitte...` etc.)

1- Activer l'application "Communication de données" (seconde touche bleue à partir de la gauche).

2- Presser la séquence de touches :

`[MENU] [P] [C] [I] [V] [5]`
(`Menu, Paramètres, Configuration, Interface, Vitesse, 9600 Bauds`)

La première colonne de paramètres de l'écran doit alors indiquer :

```
Interface :
  S-Comm. 1 (COM1)
  Vitesse 5 (9600)
  B-Arrêt 1
  Parité  Sans
  Long    8
```

Si tel est le cas:

3- Presser `[ESC]` deux fois, puis la séquence `[T] [K] [C]` (`Transfer, Kermit, Charge`)

A la demande :

```
Fichier(s) à charger avec Kermit
```

répondre en frappant : `DRIVE95.SYS`

Presser ensuite la touche `[ENTREE]`. Le lecteur est alors mis en action, puis l'écran du HP95 présente dans un encadré le message suivant :

```
Fin de transaction. Presser une touche pour continuer.
```

Obtempérer à l'aide de la touche `[ENTREE]`.

Recherche du fichier "CONFIG.SYS"

Un fichier CONFIG.SYS contient des commandes particulières qui se trouvent exécutées à chaque réinitialisation du HP95 (par exemple en pressant simultanément les trois touches : [CRTL] [ALT] [SUP]). Il s'agit maintenant de s'assurer de l'existence d'un tel fichier. Procéder comme suit :

- Refermer l'application "communication de données" (Menu, Quitte)

- Ouvrir le gestionnaire de fichiers en pressant la touche bleue située le plus à gauche. Le répertoire du disque C: apparait à l'affichage; si le fichier recherché ne s'y trouve pas, il ne reste plus qu'à explorer le répertoire racine de l'unité A: (ceci suppose bien entendu la présence d'une carte Ram dans son logement). Presser successivement les touches suivantes :

```
[F5] A: [ENTREE]
```

Si le fichier CONFIG.SYS n'existe dans aucune unité, il nous faudra transférer celui de la disquette fournie avec le lecteur.

Transfert du fichier CONFIG.SYS depuis la disquette.

Commencer par clore toute application. Ensuite répéter les points 1, 2 et 3 décrits ci-dessus; pour le dernier point, désigner CONFIG.SYS comme fichier à charger puis presser [ENTREE]. Finalement, sortir de l'application COMM (presser [ESC] deux fois, puis QUITTE) avant de réinitialiser le HP95 (pression simultanée de [CRTL] [ALT] [SUP]).

Le lecteur sera désormais reconnu comme l'unité G: et l'on peut y accéder de la mème façon que pour les autres unités (presser [F5] G: [ENTREE]).

Modification du fichier CONFIG.SYS.

On suppose ici qu'un fichier CONFIG.SYS a été trouvé au cours de l'avant-dernière étape.

- Sortir du gestionnaire de fichiers (Menu, Quitte).

- Ouvrir l'application MEMO (cinquième touche bleue depuis la gauche) et frapper la séquence: [MENU] [T] [O] (Menu, Transfert, Ouvre). L'écran doit mentionner :

```
Fichier à charger :
C:\_Dat\*.TXT
```

Effacer cette ligne par pressions répétées sur la touche ←.

- Presser ensuite : A:*.* ou bien C:*.* selon où se situe CONFIG.SYS.

- Presser [ENTREE] et contraster CONFIG.SYS dans le répertoire affiché; ensuite presser à nouveau [ENTREE].

- Sur la première ligne disponible frapper :

```
DEVICE=DRIVE95.SYS
```

- Presser [ENTREE] puis [MENU] [T] [S] (Menu, Transfert, Sauve). L'écran indiquera :

```
Fichier à sauvegarder :
```

Répondre en pressant: A:\Config.sys ou C:\Config.sys selon le cas.

- Presser [ENTREE]; à la demande :

```
Remplacer le fichier en cours? O ou N
```

Repondre [O], puis [MENU] [Q] (Oui, Menu, Quitte).

- Finalement, réinitialiser comme vu précédemment.

Lancement d'un programme à partir d'une disquette

Par précaution, placer le cran de sureté de la disquette en position "lecture". A partir du gestionnaire de fichiers, presser [F5] G: [ENTREE].

Le contenu de la disquette est alors affiché. A l'aide des touches fléchées, contraster le fichier à lancer puis presser [F4] (lance).

Recopie de disquettes.

Le processus à suivre peut se résumer ainsi:

1- Insertion et formatage d'une carte Ram (éventuellement).
2- Insertion de la disquette à recopier, protégée en écriture.
3- Détermination de son contenu.
4- Recopie du contenu de cette disquette sur carte-ram ou sur le disque C: .
5- Substitution d'une disquette vierge à celle qui vient d'être lue et formatage à l'aide du logiciel FORMAT95.
6- Duplication du contenu de la carte-ram ou du disque C: sur la disquette formatée.

Mise en place et formatage d'une carte Ram.

Vérifier que la carte Ram n'est pas protégée en écriture avant de l'insérer dans son logement.

- Placer le HP95 en mode configuration.
- Dans le menu, sélectionner l'option "Format" et suivre les instructions affichées à l'écran.
- Sortir du mode Configuration en sélectionnant Quitte.

Insertion de la disquette à recopier

La protection en écriture s'effectue en mettant en position haute le cran de sécurité situé dans le coin supérieur droit de la disquette.

Contenu de la disquette.

- Passer en DOS (Menu, Système).
- Frapper DIR G: puis [ENTREE].
- Effectuer la somme des nombres de la troisième colonne qui donnent la taille de chaque fichier.

Une autre alternative, beaucoup plus simple, permet d'obtenir le mème résultat : revenir au gestionnaire de fichiers (frapper EXIT, presser [ENTREE]). Ensuite presser [F5] (Va) puis G: [ENTREE].

Enfin des indications plus précises sont obtenues comme suit :

Presser [MENU] [R] [S] (Menu, Répertoire, Statut)
Presser [ENTREE] pour lire sur l'écran :

```
          Statut du répertoire
G:\
Sous-répertoire dans le disque:   nn
Fichiers du répertoire:           nn
Octets dans le répertoire:     nnnnnn
Octets disponibles sur disque:nnnnnn
Nom du volume:                 nnnnnn
```

Recopie du contenu de la disquette sur carte Ram ou sur le disque C:

Presser simultanément [ALT] [F5] pour sélectionner la totalité des fichiers de la disquette. Ensuite, presser :

- [F2] (Copie), A: [ENTREE] pour transférer sur carte.
- [F2]] C: [ENTREE] pour transférer sur le disque C:

Le lecteur se met en marche et l'écran indique, dans un encadré, l'action en cours.

Formatage d'une disquette vierge.

Remplacer la disquette originale par une disquette vierge. Le formatage s'effectue à l'aide du logiciel FORMAT95, comme indiqué précédemment, qu'il ne faut pas confondre avec la fonction FORMAT de l'application CONFIG; celle-ci est en effet réservée au formatage des cartes Ram.

ATTENTION : prendre soin de vérifier l'existence du fichier FORMAT95.EXE en C: avant de poursuivre, sous peine de plantage...

Si ce fichier n'existe pas :

- Remplacer la disquette vierge par celle livrée avec le lecteur.
- Presser [F5] G: contraster FORMAT95.EXE; presser [F2] C: [ENTREE].
- Une fois le transfert opéré, remplacer la disquette-logiciel par la disquette à formater; celle-ci devant bien entendu être mise en position lecture-écriture.
- Presser [F5] C: [ENTREE]
- Contraster FORMAT95.EXE, puis presser [F4] (lance).
- A la demande :

```
Format 720K or 1.44M?
  (default 1.44M)
```

presser 720 ou 1.44 selon le cas, puis presser [ENTREE].

- A la demande :

```
Volume name :,
```

Frapper le nom retenu pour la disquette, puis [ENTREE]; ce nom pouvant comporter jusqu'à 11 caractères.

- A la demande :

```
Verify disk after format ?
```

Presser Y ou N selon l'option choisie. Un formatage avec vérification durera environ 11 minutes pour une disquette 1.44MB et 15 minutes pour une disquette 720K.

- Enfin une dernière demande :

```
"Format a 1.44 (ou 720K,selon le cas) disk with
     Volume name: (nom du support) ? (Y/N)"
```

Requiert une confirmation globale.

Il est à remarquer que cette opération est particulièrement vorace. Aussi mieux vaut-il éviter d'entreprendre un formatage en absence de l'adaptateur secteur. Il semble également audacieux d'enchainer le formatage de plusieurs disquettes. Enfin il faut aussi noter que l'alarme Battery low s'affiche fugitivement et n'est pas maintenue. Toutefois, si cette alarme est restée ignorée, le processus s'interrompra sur une alarme sonore et un affichage d'erreur. Cela montre à l'évidence qu'il importe d'être très économe dans l'utilisation du

lecteur. C'est sans aucun doute la raison de la présence d'un programme nommé POWEROFF.EXE sur la disquette jointe au lecteur. Il est indispensable de charger ce programme sur le disque C: et de le lancer chaque fois que le lecteur n'est plus utile.

Transfert du contenu de A: EN G:

- Se placer dans l'unité A:([F5] A: [ENTREE]).
- Sélectionner tous les fichiers de A: ([ALT][F5]), presser [F2] (copie), puis G: [ENTREE]. Le lecteur doit se mettre en marche; un encadré au centre de l'écran indique alors la nature de l'opération en cours.

Contrôle des résultats

- Presser [F7] (Divise): l'écran se divise en deux zones. Affecter celle de gauche à l'unité A: de la manière suivante :

presser ← (flèche gauche) A: [ENTREE]

Affecter celle de droite à l'unité G:

presser → (flèche droite) G: [ENTREE].

Le contrôle des deux répertoires peut alors être effectué pas à pas. Il est également possible de comparer la taille de chaque fichier en pressant à nouveau [F7] (Remplit), puis successivement [F5] A: [F5] G:.

EN CONCLUSION

Grâce à l'utilisation de ce lecteur de disquettes, j'ai appris à employer mon HP95 plus intensément. Souscripteur de la revue "HP Palmtop Paper", après quelques tentatives à l'aide de la disquette annuelle adressée aux abonnés, je me suis résolu à souscrire au "HP Palmtop on Disk"; ceci peut être une alternative à l'utilisation de messageries telles que Compuserve sur notre continent. Je peux aussi disposer d'un ensemble de disquettes de sauvegarde des fichiers qui encombraient le disque C: auparavant; économie par rapport à l'emploi de cartes Ram.

Il est en outre fort probable que cet appareil puisse intéresser nombre de possesseurs de HP48 (en particulier pour l'emploi des "Goodies"). Voilà sans aucun doute matière à de nouveaux articles que des chevronnés du HP48 pourraient rédiger au profit de leurs cadets.

Gérard Merlier (540)

DES IMAGES SUR LE HP95

Il est de plus en plus courant de récupérer des images contenues dans des fichiers graphiques. Si leur visualisation ne pose souvent pas de problèmes sur IBM PC, comment le faire sur le HP95 ?

Cet article a pour but d'y répondre, et pourrait représenter la première partie d'une série d'articles consacrés à ce domaine.

Mais tout d'abord, pour pouvoir visualiser des images dans de bonnes conditions, il est cependant nécessaire d'effectuer diverses conversions :
- Tout d'abord, l'image doit être au format PCX. En effet, il n'existe pas (à ma connaissance tout au moins) d'utilitaires HP95 affichant d'autres types de fichiers graphiques.
- Ensuite, il faut que l'image soit au format de l'ecran : 240×128.
- Les pixels de l'écran du HP95 n'étant pas carrés (il sont d'un raport 4/3 environ). Il est nécessaire d'anamorphoser l'image, exactement comme pour un film Cinémascope.
- Le HP95 possédant un écran monochrome uniquement (sans niveau de gris), il est nécessaire de filtrer le fichier de facon à ce que les plages de couleur soient replacées par des plages de pixels plus ou moins allumés, de facon à faire apparaitre des zones plus ou moins grises. C'est ce que les Anglophones appellent le *dithering*.
- Enfin, les fichiers PCX monochromes étant prévus pour être affichés sur un écran vidéo (pixels blancs sur fond noir), il faut inverser l'image (la rendre en "négatif"), afin d'être utilisée sur l'écran du HP95 (pixels noirs sur fond blanc).

Toutes ces étapes dégradant de façon (très !) significative l'image originale, n'essayez donc pas de convertir des images présentant de trop fins détails ou des couleurs trop proches les unes des autres...

Afin d'effectuer toutes ces étapes, il est bien sûr préférable de travailler sur PC et de ne transférer que l'image finale sur le HP95. Surtout si l'on sait que certaines images originales peuvent dépasser largement le Méga-octet !

De nombreux logiciels permettent d'assurer ces conversions. Si vous ne voulez pas dépenser une fortune pour en acquérir un, vous pourrez en trouver de très bons dans le domaine public ou le Shareware (PICLAB, IMPROCESS...).

Pour ma part, j'utilise GRAPHIC WORKSHOP, qui permet de visualiser et convertir la plupart des types de fichiers graphiques (TGA 24 bits, PCX, BMP, GIF....). C'est ce programme qui va nous servir tout au long de cet article.

Si vous ne possédez pas ce logiciel, vous pouvez vous le procurer chez mon fournisseur de Sharewares habituel : DP Tool Club (BP 745, 59657 Villeneuve d'Ascq). La disquette porte la référence n°1710 et coute 20F (plus 16 Francs de port par commande).

Une fois que vous aurez recu et décompacté la disquette sur le disque dur, il ne fautra pas oublier de configurer le logiciel avec l'utilitaire GWSINSTAL. Cette étape dépendant de votre matériel, je ne peux évidement pas vous aider dans le cadre de cet article. Cependant, si vous avez des problèmes, n'hésitez pas à me contacter.

Maintenant passons aux différentes étapes de la conversion.

Conversion initiale

Le programme pouvant accepter une grande variété de fichiers en entrée, jusqu'au TGA 24 bits (16 milions de couleurs). La première étape est donc de se fixer un format de travail, qui soit le même pour toutes les conversions. Mon choix s'est porté sur un des formats les plus répandus : le GIF. Il s'agit d'un format de fichier compressé en 256 couleurs (8 bits).

Si vous utilisez déjà un fichier en 256 couleurs, utilisez la fonction Convert, accessible par la touche [F2], et selectionner le format GIF.

Si vous utilisez un fichier possédant plus de 256 couleurs (TGA 24 bits par exemple), il est necessaire de passer par la fonction Effects, accessible par la touche [F9]. A partir de là, selectionnez les sous-menus suivants : Colour reduction, Remap, 256 Colours Quantized, puis le format de fichier GIF.

En sortie, vous obtiendrez un fichier GIF en 256 couleurs. Si vous avez utilisé la seconde méthode, il portera le même nom que le fichier d'entrée, mais précédé des caractères X_. Ceci permet de garder le fichier original, quelque soit son type, et donc de revenir en arrrière si vous n'êtes pas satisfaits du résultat. Cependant, toutes les étapes suivantes transformant le nom du fichier de cette façon, il faut savoir qu'au bout de quatre étapes, le nom du fichier original aura complètement disparu. Pensez donc à sortir du programme et à renomer tout de suite le fichier final si vous avez plusieurs images à convertir.

Conversion en image anamorphosée, 240×128

La seconde étape consiste à convertir le format dans le résolution de l'écran (240×128). Cependant cette étape devra être coupée en deux parties : réduction de la résolution avec anamorphose, puis élimination des parties haute et basse de l'image. En effet, l'écran du HP95 est au format 10/4 alors qu'il nous faudrait un écran 4/3. Si nous désirons obtenir une image qui ne soit pas déformée, il nous faudra donc sacrifier les parties haute et basse de l'écran. Ceci est indépendant du problème du format des pixels, qui demande une anamorphose et dont nous allons nous occuper tout de suite.

Conversion en image anamorphosée, 240 colonnes

Cette première partie demande d'abord une petite partie de réflexion : Si nous savons que nous voulons obtenir un image de 240 colonnes, comment déterminer le nombre de ligne de l'image ? Ce n'est pas 128, puisque nous devrons couper l'image finale à cette valeur, comme je l'ai expliqué plus tôt. Mais alors quelle est la valeur à déterminer ?

En fait cette valeur dépend de deux paramètres : tout d'abord, il faut obtenir une image intermédiaire respectant le format initial de l'image. Ensuite il faut déformer l'image en fonction du format des pixels.

Le premier paramêtre est une simple règle de trois : il suffit de diviser le nombre de colones par le format de l'image initiale. Le second paramètre est le rapport entre les formats des pixels des deux écrans.

Ce qui nous permet d'obtenir l'équation suivante :

$$lignes = 240 * \frac{nb_lignes}{nb_colonnes} * \frac{4/3}{format_pixel}$$

où nb_lignes et nb_colonnes sont le format de l'image initiale et format_pixel est le format du pixel de l'écran pour lequel a été défini l'image initiale. En fait cette valeur est toujours à 1, sauf pour les images qui ont été prévues pour un écran CGA, pour laquelle cette valeur est de 4/3. Le résultat (lignes) est donc la valeur correspondant au nombre de lignes de l'image intérmédiaire.

Pour vous éviter quelques calculs, voici les valeurs correspondant aux formats d'images les plus répandus (pour un écran VGA) :

320×200 → image de 240×200
640×400 → image de 240×200
640×480 → image de 240×240
800×600 → image de 240×240

Pratiquement, sur Graphic Workshop, nous utiliserons la fonction SCALE, accessible par la touche [F8]. Celle ci permet trois modes de conversion : proportionnel, anamorphique (tous les deux demandant un pourcentage d'élongation) et par taille, permettant de fixer la taille de l'image en valeur absolue.

Pour arriver à ce mode (puisque c'est celui qui nous intéresse), il nous faudra appuyer encore deux fois sur [F8] de facon à obtenir l'écran suivant :

```
Set Scale Factor :
     WIDTH : 100
     DEPTH : 100
```

Il nous suffira de remplacer les deux valeurs numériques par 240 pour WIDTH et 200 ou 240 (suivant le format de l'image initiale) pour DEPTH.

Après ceci, le programme nous demandera le format du fichier de sortie (nous répondrons GIF) et sauvera le fichier avec un nom commencant par S_.

Avant de terminer cette partie, je ne ferais que vous conseiller d'utiliser une image initiale de grande résolution. En effet, si vous utilisez une image d'un format proche de celle de destination, vous risquez d'obtenir un effet de moirage peu esthétique.

Conversion en image 240×128

Une fois cette première phase effectuée, il faut mainenant supprimer les parties hautes et basse de l'image. Cela est fait en utilisant la fonction Crop, accessible par la touche [F5].

Le programme affiche alors l'image (en noir et blanc). En utilisant les touches [Shift Droit] et [Shift Gauche] en coordination avec les touches fléchées, nous pouvons redéfinir les nouvelles dimensions de l'image. Puisque nous avons déjà défini la largeur de l'image (240 colonnes) nous ne devons modifier que la hauteur en cadrant la partie intéressante.

Une fois une zone de 128 ligne sélectionnée, appuyer sur F10 pour sauvegarder la nouvelle image au format GIF. Ce nouveau fichier portera le préfixe C_.

Conversion en image monochrome

Si les précédentes étapes demandaient peu de réflexion, cette étape est probablement celle qui risque de demander le plus d'essais avant de récupérer une image qui restitue le plus de détails de l'original.

Tout d'abord, vous avez remarqué que cette étape se situe vers la fin du processus et que nous avons toujours une image couleur. Cela est du au fait que si nous étions passé dès le début en monochrome, nous aurions vite perdu de nombreux pixels significatifs de l'image. De plus, le fait d'avoir conservé une image couleur nous permet de traiter les parties les plus complexes du processus sans avoir a reprendre toutes les étapes.

La première partie consiste à préparer l'image de façon à accentuer tous les détails importants de l'image, par rapport aux éléments qui les entourent. Ceci est fait en travaillant sur le contraste et la couleur. Par exemple, si nous avons une image relativement sombre, avec une partie de l'objet principal dans l'ombre, le fait de diminuer le contraste permet de "sortir" l'élement sombre de façon à le rendre plus visible.

Pratiquement, sur Graphic WorkShop, cela se fait en affichant l'image et en utilisant les touches [R], [G] et [B] pour les couleurs primaires, [C] pour le contraste et [I] pour la lumière. (La touche [Shift] permettant de diminuer ces valeurs). Une fois le résultat désiré obtenu, on sauve l'image (qui portera le préfixe A_) à l'aide de la combinaison de touches [Shift] [ESC].

L'image étant maintenant prête à passer en monochrome, il nous faut selectionner la fonction Dithering à l'aide de la touche [F3]. Le programme nous demande alors différentes méthodes, dont Floyd, Burkes et Stucki. Théoriquement, la troisième méthode est la meilleure, mais étant donné la faible résolution de l'image, certaines images peuvent être plus fidèles en utilisant une des deux autres méthodes. Il vous faudra tester les trois et prendre l'image qui vous semblera la meilleure.

A partir de cette étape, vous pourrez stocker l'image résultante au format PCX, qui sera préfixée D_ après cette opération.

Inversion de l'image

La dernière étape est très simple par rapport à ce que nous venons de faire, puisque il s'agit de selectionner la fonction Inverse avec la touche [F6] et de sauver le résultat dans un nouveau fichier PCX portant le préfixe R_.

Visualisation de l'image sur le HP95

Après avoir renomé le fichier avec un nom moins barbare que celui résultant des étapes précédentes. Vous pouvez le transférer sur le HP95.

Pour le visualiser sur cette machine, vous avez maintenant deux solutions :

- Utiliser un utilitaire tel que PCXVIEW (qui est inclus dans notre disquette n°1).

- Ou mieux, en profiter pour remplacer le "*TOPCARD*", c'est à dire l'image affichée par le System Manager lorsque toutes les applications sont fermées. Pour cela, la méthode la plus simple est de copier le fichier dans le directory _DAT en le renomant TOPCARD.PCX. Notez que pour que l'image ne soit pas surchargée par les informations parasites (Nom de l'utilisateur, Compagnie...) vous devrez les effacer en utilisant le sous-menu Owner du Setup. Il peut en être de même pour l'horloge et le message Press Any Key présent en bas à gauche de l'écran. Pour cela, il vous vaudra (toujours dans le sous-menu Owner) spécifier votre fichier graphique en l'appelant autrement que _DAT\TOPCARD.PCX.

Jacques Belin (123)

PROGRAMMATION GRAPHIQUE : LES POINTS NOIRS

Il est des fois où certaines erreurs ont la dent dure. En effet, ce n'est qu'après avoir fait plusieurs programmes graphiques et repris l'article présenté dans JPC 84 pour *Poche Computers*, que je me suis aperçu que j'avais fait deux lamentables gaffes...

La première était présente dans les articles eux mêmes et non dans le listing du fichier grafx95.c. Elle concerne la fonction d'entrée en mode graphique (que j'avais jugé bon de simplifier par rapport aux fichiers de description) et ou je mentionnais le code de sous-fonction 20h au lieu de 80h. Le résultat était simple : Plantage de la machine en entrée du mode graphique lorsque le programme exécuté est une application DOS.

La deuxième erreur ne s'est manifesté que sur le HP100LX. En effet, si le HP95 accepte très bien de revenir en mode texte avec la fonction 0 / sous-fonction 7 de l'interruption 10h, le HP100 impose absolument de passer par l'interruption 5Fh pour revenir en mode texte. Pour que le programme fonctionne bien sur les deux machines, il est donc conseillé de n'utiliser que l'interruption 5Fh.

Je vous donne donc le texte exact de la fonction concernée (toujours sur l'interruption 5Fh) :

AH = 0h
AL = mode :
  87h : texte
  A0h : graphique
  07h : texte (system manager)
  20h : graphique (system manager)

Les sous fonctions 07h et 20h ne doivent être employées que quand le programme est une application System Manager. Elles permettent de l'informer que le mode graphique a changé et le contenu de l'écran détruit. Elles ne doivent pas être utilisées dans un programme purement DOS, sous peine de plantage.

Jacques Belin (123)

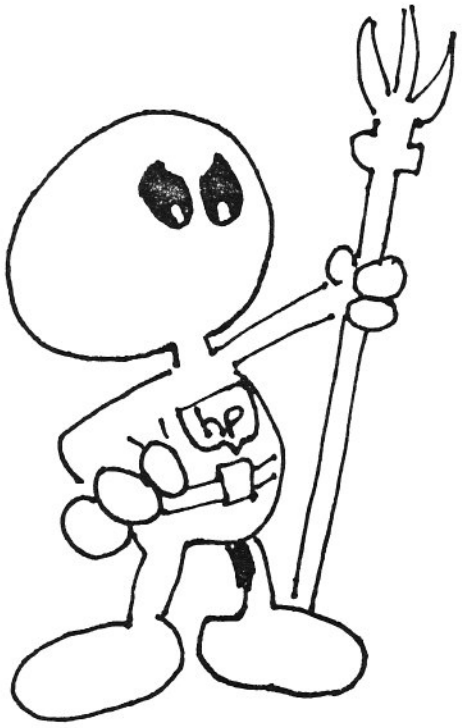
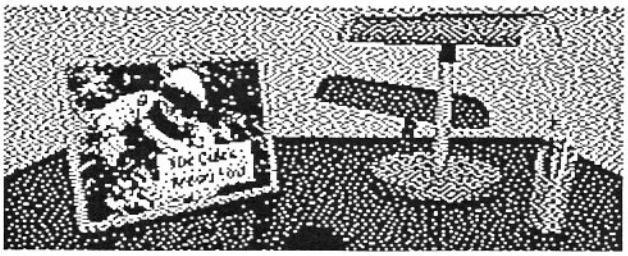




Image Originale (640*480, 256 couleurs) convertie en Noir et Blanc

(Image de synthèse de Tom Price, réalisée avec
le logiciel Persistence Of Vision).



La même, telle qu'elle apparaitra sur le HP95.

LE COIN DES CODES

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

Note importante :

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

Programmes 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 les deux programmes correspondant à votre machine:
- pour HP-48 : ASSCOD48 et INPUT48
- pour HP-28S : ASSCOD28 et INPUT28

(*attention aux parties communes et spécifiques HP-48 ou HP-28S et aux commentaires*)
Ceci avec la plus grande attention car leur 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 [.].

ASSCOD48 996.5 octets
cksum # 5D1Bh

```
« RCLF HEX 64 STWS 1 SF "" 'tmpcod' STO
"nombre d'octets                          @ + NEWLINE
                                          @ NEWLINE
" 17 INPUT48 1 CF STR→ 2 * 16             @ INPUT28
DUP2 / IP 3 ROLLD MOD DUP2 0 > +          @ si HP-28S
3 ROLLD 1 + 4 ROLL # 0h DUP → n r f s o « 1 SWAP
FOR i
 DO "ligne " i 1 - R→B # 1000h + →STR 4 6 SUB + DUP
  "                                       @ NEWLINE
  chaine                                  @ + NEWLINE
  " + "---- ---- ---- ----" n i <
  IF THEN r DUP 4 / IP + SWAP OVER 1 SWAP OVER -
       SUB SWAP 18 + ELSE 38 END 'o' STO +
  "                                       @ NEWLINE
  " + 1 FS?C IF THEN ROT SWAP CLLCD 1 DISP HALT
             ELSE o INPUT48 END DUP       @ INPUT28
  WHILE DUP " " POS DUP                   @ si HP-28S
  REPEAT DUP2 1 SWAP 1 - SUB 3 ROLLD
   1 + 25 SUB IF DUP " " == THEN DROP ELSE + END
  END DROP 0 OVER SIZE 1 SWAP
  FOR j OVER j DUP SUB NUM j * + NEXT
  s + DUP # FFFh AND
  "                                       @ NEWLINE
  somme de controle                       @ + NEWLINE
   ---                                    @ + NEWLINE
  #" 6 ROLL SWAP + 34 INPUT48 STR→ ==     @ INPUT28
  IF THEN SWAP DROP 1                     @ si HP-28S
  ELSE DROP2 1000 1 BEEP 1 SF 0 END
 UNTIL END 's' STO
 WHILE DUP " " POS DUP
 REPEAT DUP2 1 SWAP 1 - SUB 3 ROLLD 1 + 19 SUB +
 END DROP 'tmpcod' DUP RCL ROT + SWAP STO
NEXT f STOF » tmpcod
@ * HP-48 * @@ ********** HP-28S **********
"GROB 8 " @@ # 20204A04F3D02C67h # F80004F02C96040Ch
OVER SIZE @@ # 8DCC05081F804F27h # 313103190F818341h
2 / " "   @@ # 681808AE91B51391h # 45DC061171085168h
+ + SWAP  @@ # F3D3CEAA125E5D8Eh # 2F9004h 28 STWS
+ STR→    @@ #0 OR 64 STWS 1 7 START # 3B82h SYSEVAL
# 4017h   @@ NEXT # 20238h SYSEVAL # 4F3Dh SYSEVAL
SYSEVAL   @@ EVAL
# 56B6h   @@ toutes ces lignes marquées @@ peuvent
SYSEVAL   @@ être remplacées par ASC→ :
DROP      @@ soit pour HP-48 (JPC-79 page 11)
NEWOB     @@ ou pour HP-28S (JPC-85 page 8)
"fin" CLLCD 1 DISP »                      @ HP-28S dito
```

INPUT48 412.5 octets
cksum # 15C7h

```
« SWAP 1                                  @ partie commune
WHILE OVER CLLCD 1 DISP                   @ HP-48 et HP-28
REPEAT                                    @ puis séquence
 DO UNTIL KEY END                         @ HP-48 ou HP-28
@ ********** HP-48 ********** @@ ****** HP-28S ******
 IF DUP 51 == THEN DROP 0       @@ IF DUP "ENTER" ==
 ELSE IF DUP 55 == THEN DROP    @@ THEN DROP 0
   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  @@ suite voir HP-48
           92 == THEN 48 END    @@
                 46 END CHR     @@
  + DUP SIZE 3 PICK - 2 + 5 MOD NOT @ partie
  1 FC? AND IF THEN " " + END END 1 @ commune
 END END SWAP 60 SUB »                    @ HP-48 et HP-28
```

INPUT28 228 octets
cksum # 9412h

Donc à partir de maintenant 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, correspondant au numéro de ligne à 3 chiffres, sans les espaces et validez.
4- tapez la somme de contrôle et validez. S'il y a erreur la ligne de codes sera demandée à nouveau après émission d'un BEEP. L'appui sur EDIT fera apparaître la ligne des codes qui pourra être corrigée. Alors relancez avec CONT.
5- après l'affichage de "fin" stockez le programme assemblé dans la variable donnée en tête.
6- si tout s'est bien déroulé vous pouvez purger tmpcod qui contient la chaîne de codes.

Programmes MS-DOS

Afin d'être utilisé par tous, ce programme est destiné à être écrit en GWBASIC. Il devrait cependant être facile de le convertir pour un autre programme (QBASIC, Turbo BASIC...).

Mode d'emploi :

1- Lancer le programme : GWBASIC MAKEDOS.BAS
2- Entrer le nom du fichier destination.
3- Entrer la taille du fichier.
4- Entrer les listes de codes puis le checksum (en prenant soin d'entrer les codes héxadécimaux en majuscules). En cas d'erreur corriger la ligne, en prenant soin de placer le curseur après le dernier caractère avant de taper sur la touche d'entrée.
5- Une fois que toutes les lignes sont entrées, sortir du GWBASIC en exécutant la commande SYSTEM. Le nouveau programme est immédiatement disponible.

Note : La taille du fichier résultant peut être supérieure d'un octet à ce qui est affiché dans le listing. Cela n'est pas un problème.

Programme MAKEDOS.BAS

```
 10 INPUT "Nom du fichier : ",NOM$ : INPUT "Nombre d'octets : ",N : N=N*2
 20 OPEN NOM$ FOR OUTPUT AS #1 : CLOSE #1 : KILL NOM$
 30 OPEN "bin.tmp" FOR OUTPUT AS #1 : S=0 : P$="---- ---- ---- ----"
 40 NLINES=N\16 : LENLAST=(N MOD 16)+((N MOD 16)\5)
 50 IF (N MOD 16)=0 THEN NLINES=NLINES-1 : LENLAST=LEN(P$)
 60 FOR X=0 TO NLINES
 70   IF X=NLINES THEN P$=LEFT$(P$, LENLAST)
 80   C$=P$
 90   X2$="00"+HEX$(X) : PRINT RIGHT$(X2$,3);":";
100   Y=CSRLIN : LOCATE Y,6 : PRINT C$; : LOCATE Y,6 : INPUT "",C$ : IF Y=24 THEN Y=Y-1
110   LOCATE Y,27 : PRINT " sm = ---" : LOCATE Y,33 : INPUT "",D$
120   M=S
130   FOR Z=1 TO LEN(C$)
140     IF MID$(C$,Z,1)<>" " THEN M=(M+((Z-(Z\5))*ASC(MID$(C$,Z,1)))) MOD 4096
150   NEXT Z
160   D2$="00"+HEX$(M) : D2$=RIGHT$(D2$,3)
170   IF D2$<>D$ THEN PRINT "Erreur de somme" : BEEP : GOTO 90
180   FOR Z=1 TO LEN(C$) STEP 2
190     IF MID$(C$,Z,1)=" " THEN Z=Z-1 : GOTO 230
200     CH=ASC(MID$(C$,Z,1))-48 : IF CH>9 THEN CH=CH-7
210     CL=ASC(MID$(C$,Z+1,1))-48 : IF CL>9 THEN CL=CL-7
220     PRINT#1,CHR$((16*CH)+CL);
230   NEXT Z
240   S=M
250 NEXT X
260 CLOSE #1 : NAME "bin.tmp" AS NOM$ : END
```

```
FIND28S                    (HP28)
# E0C5h              126 octets

     0123 4567 89AB CDEF   sm

000: 76C2 04C3 C0D3 F406   DEA
001: 9C20 3E00 08F1 8050   A2D
002: 1B00 FFF3 0F15 C014   7DF
003: 3133 1303 407A 2014   34E
004: 38A2 F033 2020 8D9B   106
005: 1111 7914 7136 1370   C62
006: 6174 1431 3114 334E   886
007: 4A20 8A6F C174 1431   54E
008: 7481 8F84 3302 008B   1E6
009: E40D 6133 CA13 181E   F84
00A: CED7 8163 193B F11C   DD0
00B: 114B 9EA8 0818 686A   BEC
00C: 88A6 F55E 80DF 14A9   B26
00D: 64B0 1521 9108 0160   646
00E: 5BE1 32E4 0713 5141   210
00F: 8D1E 5210 9F20        2A6


XMOD→KER                   (HP48)
# BB7Bh             62.5 octets

     0123 4567 89AB CDEF   sm

000: D9D2 0D29 51D9 F81D   F14
001: 0040 D9D2 0CCD 2005   CBF
002: 0008 FB97 6014 3131   893
003: 1741 4717 4818 FA5D   725
004: 734C F820 4E11 4313   360
005: 0142 8A68 0133 5A01   F7C
006: 70CF 56E8 D276 3075   CE1
007: 660B 2130 B213 0      F55


BANKN2                     (HP48)
# 2B15h            375.5 octets

     0123 4567 89AB CDEF   sm

000: D9D2 0FDE 8111 9201   CAC
001: 1000 D9D2 0B7E 26B7   B4C
002: E267 9E60 D9D2 02C2   943
003: 302B AA0B 9F06 12F0   747
004: 6CAF 0632 2302 FA30   448
005: CB91 6D9D 20B4 F067   283
006: A226 8DA1 6C2A 2070   FAE
007: 0001 3C2A 2070 0002   AC8
008: 3C2A 2052 000E 4F60   7B6
009: 2361 6274 6029 6E60   455
00A: 207F 6274 7023 2230   F9B
00B: 3915 0A64 059F F308   D5C
00C: 8130 EC4A 35E1 4195   AC7
00D: 124C B916 4423 0C2A   829
00E: 2013 0004 5572 7E60   479
00F: 2F46 4640 216E 6460   0D9
010: 2472 7970 2167 6169   D02
011: 6E69 33A1 B213 08DA   B19
012: 16D9 D20B 4F06 C2A2   980
013: 0130 0094 C6C6 5676   6C8
014: 16C6 0226 16E6 B602   3C2
015: 3777 9647 3686 96E6   184
016: 7693 3A1B 2130 CCD2   FD4
017: 0E21 0017 98FB 9760   D63
018: 1C91 4313 0164 AF01   A0B
019: 5A99 7C60 6ED0 81AF   969
01A: 0116 4142 8408 A850   5B4
01B: 8501 7414 3131 1741   0B9
01C: 4381 AF02 72B0 8F51   E58
01D: 1108 60B0 208F F370   BBA
01E: 71BA FFFF 14E1 6314   9E4
01F: E14E 14E1 8114 E161   6A9
020: 14E8 1AF1 2CCC C5F0   6AA
021: 1811 4E14 E6B3 0CC5   555
022: 2118 114E 1611 4E66   23B
023: 20CC 5211 4E14 E181   F4E
024: 14E6 110C C581 14E1   CA2
025: 8114 E1A0 1001 4E63   91E
026: 0086 0B02 18FF 3707   696
027: 8F5E 0108 D341 501F   3AD
028: 5F00 8071 3434 EC00   06E
029: 08FC 0760 8D5F 0084   DE7
02A: 4230 E4B1 2B21 3075   A47
02B: 660E 8F60 B213 0B21   712
02C: 3000 0000 0000 0000   095
02D: 0000 0000 0000 0000   A15
02E: 0000 0000 0000 000    095


GO                         (HP48)
# 866Ah            375.5 octets

     0123 4567 89AB CDEF   sm

000: D9D2 0ECE 81B2 040D   E3D
001: 9D20 CCD2 07A2 008F   C52
002: B976 0143 1311 7415   78D
003: 3710 1D21 0A1B 9D68   5A8
004: 0142 1301 4281 AF00   20A
005: D381 8F3F 1B6A D801   080
006: 4613 4146 1301 648A   CC6
007: 6B28 50D0 1523 D8C4   A81
008: C4D6 C4CA C081 8F02   8BF
009: 136C A130 6F00 8408   546
00A: 18F0 7130 1468 AE60   287
00B: 65A0 C213 4D01 4A11   F0D
00C: 9966 6068 C0C4 8A88   D54
00D: 0818 F018 18F0 1136   9AF
00E: CA13 0142 D534 69A2   6C7
00F: 08A2 B1D9 1341 8414   314
010: 28A8 7213 6E26 AAF1   1D2
011: 1AE6 10A1 6714 28A8   F10
012: 6D13 6C26 E8F1 1A8A   E29
013: A52C E10A 18ED 014A   C97
014: C48A 8808 18F0 1136   919
015: E265 AF81 AF10 8F1D   85C
016: 3808 1AF1 88A6 9134   5A7
017: 4020 0DA8 F2D7 608D   470
018: 3205 081A F006 4CE9   2E1
019: 6C60 6D60 DEC6 818F   24C
01A: 218B 712A 6E80 D015   FB2
01B: 2115 7191 294D 014A   C9F
01C: 2067 0F13 2103 1301   779
01D: 3310 4131 818F AFD5   67B
01E: 1567 1537 9763 116F   376
01F: 17FD 98BB 756D DF69   3C7
020: 6020 D014 AC48 A880   186
021: 818F 0181 8F01 136C   EB7
022: A130 142D 5346 9A20   B5E
023: 8A66 4136 1B9D 6801   86F
024: 4213 0144 8F34 150C   540
025: E80D 0152 111B 1341   0E7
026: 5719 12C9 11C1 3520   CFA
027: D014 A645 ED91 3411   9D0
028: A8AE E186 0911 B6AD   8BD
029: 8014 2130 1421 3066   3B1
02A: 9F18 4142 8A8E 0136   0B0
02B: E213 66CE F18C 687F   05A
02C: 4423 0B21 30B2 130C   CDA
02D: 0000 0000 0000 0000   65A
02E: 0000 0000 0000 000    CDA
```

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 EXchange), la fédération des principaux clubs Européens d'utilisateurs de calculateurs HP.

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

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

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

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

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

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

Imprimé par Paris Copie, 4 rue Linné, 75005 Paris