

MICRO REVUE

LA REVUE DE L'INFORMATIQUE PORTABLE

DOSSIER FORTH



© 1984 Bimestriel

ÉDITÉ PAR PPC-T

N° 2 Sept.-Oct. 1984

40 F

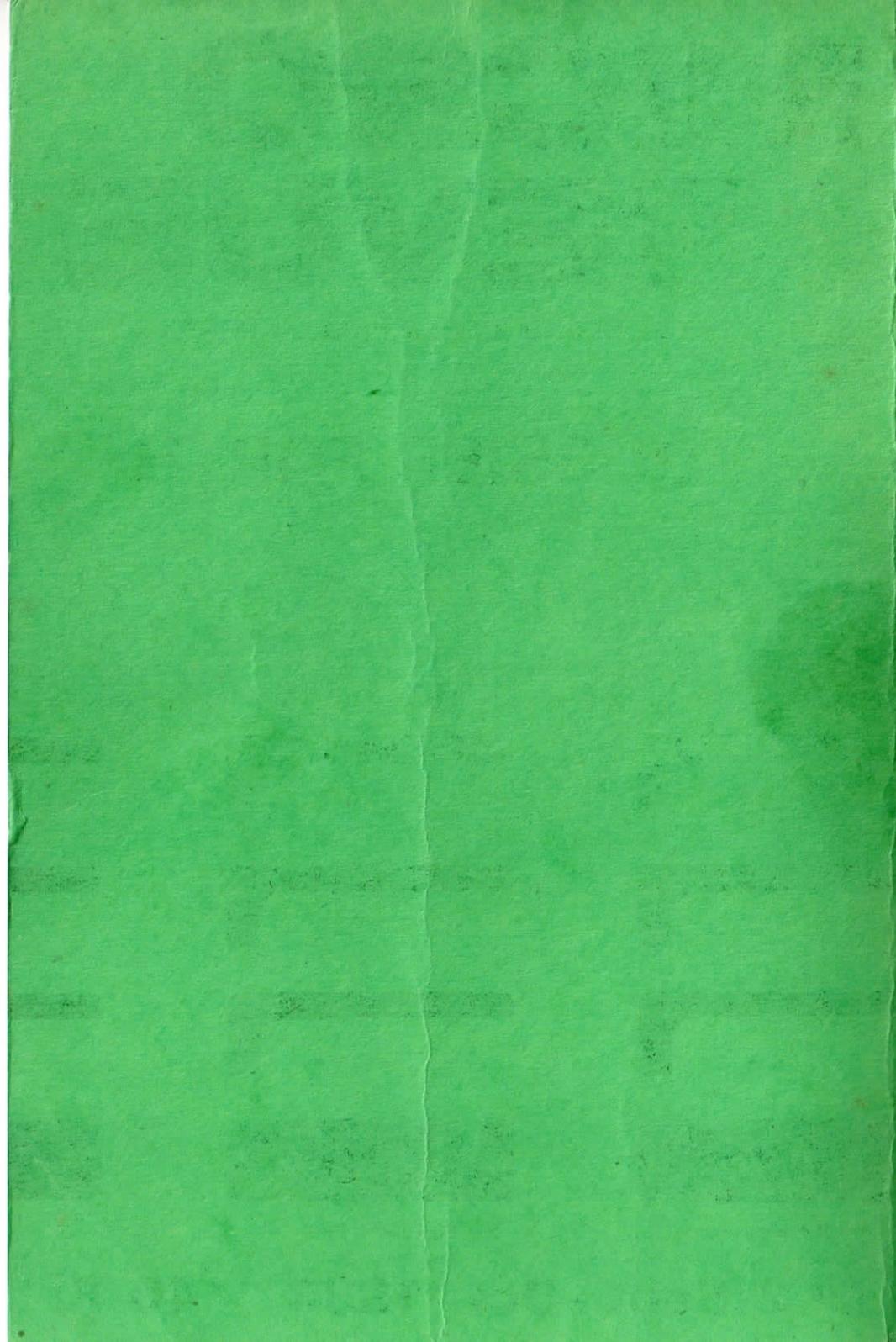


TABLE DES MATIERES

Page	Contenu
5	Editorial
	<u>DOSSIER FORTH</u>
7	Dieu le Père, une interview de Charles Moore.
12	Concepts, Interpréteur, Compilateur,....
17	HP-71BF, où comment le FORTH vint au HP-71.
23	Utilitaires, chaîne alpha, éditeur,...
28	Jedi, la FORTH est avec nous aujourd'hui.
31	Utilitaires, DUMP, LISTING toutes choses utiles pour commencer.
35	Calcul flottant
38	Mastermind
38	Remarque sur le standard
21	FORTH HP-41 !
	<u>TESTAMENT?</u>
43	Il n'est pas mort, bien heureusement, mais Richard Nelson nous a quitté de façon un peu triste.
	<u>PROGRAMMES HP-41</u>
55	Isola
62	Décode prgm
65	Config
69	FLAS, fichiers ASCII.
70	Vérité, elle vient de loin, en est-elle meilleure ?
	<u>APPLICATIONS</u>
78	PPC44
80	Structure des cassettes HP-IL.
84	Le XF mis à nu.
86	Variables symboliques.
89	W"
92	XROM
	<u>MICROCODE</u>
96	Analyse
98	Assembleur, Langage machine ou Microcode ?
99	Call
	<u>HP-1x</u>
103	Faux billets !
	<u>HP-71B</u>
106	Graphique
110	Le Math ROM.
	<u>HP-75</u>
113	Dessin
115	Trucs, des fonctions cachées, Wow!
118	Help
	<u>MAGAZINE PPC-T, LE COIN DU CLUB</u>
121	L'organisation de PPC-T
122	Les Journaux, les petites annonces.
124	SOS
125	Nouvelles brèves.
126	Erreur, Sur Le Marché.
129	Les livres
130	Programmathèque

ADRESSES :

MICRO-REVUE, PPC-T, 77 rue du Cagire 31100 Toulouse.

EDITION :

Édité par le club de Programmation Personnelle sur Calculatrice de Toulouse (PPC-T).

REDACTION-ADMINISTRATION :

Rédacteur en chef, directeur de la publication : Jean-Daniel Dodin, Secrétaire de rédaction : Gilles Barret et Jean-Yves Pasquier, Administration : Jean-François Sibille.

VENTE ET DISTRIBUTION :

Vente en librairie et par correspondance. Diffusion par l'éditeur. De plus MICRO-REVUE est disponible à un tarif très avantageux pour les autres clubs et associations qui le désirent, ainsi que pour les professionnels et les libraires qui souhaitent le revendre.

PARUTION :

Bimestriel, paraît en début de bimestre civil (Janvier, Mars, Mai, Juillet, Septembre, Novembre), 6 numéros par an.

TARIFS (franco de port) :

Prix de vente au numéro : 40F (50F pour envoi par avion).

Abonnement individuel : abonnement annuel 200F (au lieu de 240F, prix de revient au numéro), 250F pour envoi par avion.

Soumission d'article :

Tous les membres du club sont invités à fournir des articles pour MICRO-REVUE. L'original à nous faire parvenir doit être tapé à la machine avec un ruban bien noir, recto seul, simple interligne, avec une largeur maximale de 15 cm.

Des articles manuscrits sont acceptés s'ils sont calligraphiés sur papier blanc quadrillé 5x5, sans sauter de ligne, recto seul, au stylo noir.

Nous faire parvenir cartes ou cassettes (qui vous seront retournées), pour les programmes, ainsi qu'un listing.

Cet original sera utilisé tel que pour la publication, soignez-le ! Ceux qui prévoient à l'avance d'écrire des articles peuvent nous demander un KIT AUTEUR qui leur donnera les modèles et toutes les indications nécessaires.

© 1984 REPRODUCTION INTERDITE

En dérogation aux mentions ci-dessus, la reproduction des programmes est autorisée pour l'usage personnel des lecteurs, à l'exclusion de toute reproduction commerciale.

Ont collaboré à ce numéro :

Martin Ouverson, Charles Moore, J-D Dodin, W.C. Wickes, B. Jouve, Eric Aubourg, Marc Petreman "JEDI", Jean Thomas, R. J. Nelson, Bill Kolb, Gary Bachlund, Franck Lebastard, Robert Schwartz, François Laux, J. P. Magnan, Antoine Pierre, J. Boschhat, Olivier pougeon, Cyril Reigner, J. C. Vandenhakke, Jean-Christophe Benoit, Marc Chiffolleau, Alexis Kupier, Serge Vaudenay, Gilles Barret, Michel Paulus, Janick Taillandier, Marcel Trimborn, Damien Debril.

EDITORIAL



*Dernière minute : FORTH HP-41 p 21
La HP-41 est aban donnée au profit de la CV
forte hausse de tarif (x 20%)*

Chers amis,

Voici donc le deuxième numéro de MICRO-REVUE. Beaucoup de choses se sont passées pour moi pendant ces "vacances". En particulier j'ai mis en page à l'avance tout ce que vous m'aviez envoyé jusqu'à présent.

J'ai constaté ce faisant que sur certains sujets (applications, microcode, dans une moindre mesure programmes pour HP-41) la publication demande actuellement un an de délai. J'ai encore en stock certains articles datant de Aout 83 qui ne passeront pas dans le présent numéro, faute de place. Ceci est normal, les nouvelles machines ne s'introduisent que peu à peu parmi nos adhérents, et d'ailleurs la proportion d'articles HP-41 que je reçois diminue (à moins de dire que les autres augmentent?). Il n'y a pas de quoi s'inquiéter, mais si votre article n'est pas encore publié, prenez patience.

Actuellement tout article reçu au club fait l'objet d'un accusé de réception. Tout article non accepté est retourné avec le motif et des conseils. Si vous voulez être publiés à coup sûr, envoyez moi des articles sur les petites machines (HP-1x, c'est à dire 11, 12, 15, 16..., mais aussi "vieilles" comme HP-34C...).

Vous constaterez à la lecture de la rubrique "Sur Le Marché" que l'Europe est en pointe pour tous les produits adaptables HP-41 ou HP-11. Presque tout ce qui se fait en ce moment (4 modules client, 2 interfaces vidéo, 1 lecteur d'eprom, 2 MLDL...) vient d'Europe. Le plus beau est l'interface vidéo graphique (photo p. 3, détail p. 128). Enfin du graphique à pas trop cher (enfin, c'est relatif).

Le dossier de ce numéro est consacré au FORTH. J'attends vos réactions. Le prochain dossier est consacré aux débutants. Il n'y a encore que peu d'articles sur ce sujet essentiel. Ne vous demandez pas ce que je veux mettre dans ce dossier, mais faites un article sur ce que vous pensez qu'il faut y mettre. Tous à vos plumes...

Pour le numéro de Janvier, je pense à un dossier qui fait un peu suite à celui sur le FORTH et à celui sur les débutants, et qui pourrait s'intituler Les langages de programmation, vrai ou faux débat ? où "y a-t-il une programmation indépendante des langages, ou au contraire chaque langage entraîne-t-il une programmation particulière?". Nous traitons déjà dans ces colonnes de langage spécialisé/calculatrices, de BASIC, de FORTH, de langage machine HP-41, HP-75, HP-71. J'aimerais beaucoup que ceux d'entre vous qui ont eu l'occasion d'essayer d'autres langages nous fassent partager leur expérience.

Mais j'attends de vous d'autres sujets de dossiers. Certains demandent un dossier sur l'HP-11. Il y a le livre de Janick, qui tarde à écrire le deuxième tome, et il doit y avoir beaucoup à écrire là-dessus, pourquoi pas. Faites connaître vos idées.

Enfin, dernier point, je vous propose de personnaliser un peu MICRO-REVUE en publiant en tête d'article une photo d'identité de l'auteur (au moins la première fois qu'il publie). D'où mon portrait ci-dessus. Joignez une photo d'identité (qui ne vous sera pas retournée) si vous voulez qu'elle soit publiée.

Bonne lecture

J-D Dodin (T1, USA7226)

DIEU LE PERE

L'article ci-dessous est une traduction d'une interview parue dans Forth Dimensions Volume V, N°2, Juillet/Aout 1983.

Martin Ouverson, Forth Dimensions

L'auteur a trouvé la maison de Moore en se promenant à travers les rues de front de mer ensoleillées de la Californie du Sud jusqu'à ce qu'il trouve une femme arborant la douce et souriante voix de "Min" Moore. Les présentations faites sous son aile, chacun fut installé devant le grand panorama de l'Océan Pacifique et de l'île de Catalina. Les bandes de l'enregistrement sont hachées d'exclamations, de rires, d'observations et de spéculations de l'inventeur de FORTH.

Forth Dimension : Est-il vrai que vous avez développé FORTH à l'origine pour améliorer la productivité de votre programmation ?

Charles Moore : Amélioration de la productivité est une expression sévère - je ne pense pas que j'en attendais d'écrire davantage de programmes. J'en espérais probablement une simplification de ce que je faisais déjà. Le désagrément de la compilation d'un programme FORTRAN était ce que je cherchais d'abord à éviter : Charger encore et encore des paquets de cartes, les mettre en place, les enlever et ensuite imprimer des listages pour voir ce qui se passe.

Il est évident qu'ayant contourné le système d'exploitation, le travail était facilité et qu'il était possible d'en faire plus ; aussi je pense que ma productivité était améliorée. Mais à ce moment, j'avais accepté la notion "une année, un programme".

J'ai entendu plusieurs histoires au sujet de votre première application pratique de FORTH...

La première était pour le dessin de tapis, les dessins du tapis, en graphique haute résolution. Ce n'était pas une trame, c'était un système graphique vectoriel. Et même avec le type de FORTH que j'avais alors, je pouvais faire des figures mobiles complexes - pas vraiment adapté aux tapis. Je pouvais faire en 4K et 1/10ème de secondes ce qui autrement aurait pris plusieurs secondes. Vous ne pouvez pas effectuer de mouvements ainsi, et la place mémoire aurait été insuffisante.

Ainsi vous êtes passé du piéton à l'extra-terrestre en travaillant ensuite en astronomie et contrôle d'instruments ?

Bien, après les tapis, je me suis occupé d'un système d'exploitation de grande base de données. Là j'utilisais FORTH comme système d'exploitation et d'interface avec des modules COBOL. Tout le travail réel devait être fait en COBOL. Heureusement l'aspect système d'exploitation de FORTH avait été amélioré, et il faisait du bon travail. Les performances étaient améliorées de façon spectaculaire par rapport à toute autre méthode. A ce point j'ai commencé à penser que j'aurais pu coder les modules en FORTH plus facilement qu'en COBOL. Je n'ai pas insisté sur ce point, parce que c'était trop demander aux gens. "Ce n'est évidemment pas possible - sinon, à quoi servirait le COBOL ?" La réponse est (sourire), j'avais raison et il n'y avait aucun besoin de COBOL.

Mon application suivante était l'acquisition de données astronomiques, suivie par le contrôle de télescope. J'y ai travaillé plusieurs années ; c'était drôle. Si jamais je pouvais obtenir le bon équipement, j'aimerais l'installer sur mon propre télescope. Petit, 20 ou 25 cm, avec de bons contrôles et une vidéo, pour que je puisse mettre le télescope sur le toit, m'installer à l'aise dans ma salle de contrôle et observer juste comme le font les astronomes.

Certains disent que FORTH crée ou attire l'intérêt des fanatiques.

Nous en parlions l'an passé, non avec des utilisateurs de FORTH, mais avec des

industriels. Spéculant sur l'aspect surnaturel de l'inventeur à succès. Il ne correspond pas aux schémas courants de vie et de carrière. Le même genre de chose est vrai des programmeurs FORTH -probablement des informaticiens en général. Les gens qui aiment travailler avec des ordinateurs, que ce soit sur le matériel ou sur le logiciel, ne sont pas des gens ordinaires. Ils doivent être introvertis -des tas de termes psychiatriques pourraient être utilisés- mais en particulier ils sont du genre qui préfère traiter avec les objets plutôt qu'avec les gens.

C'est une caractéristique très particulière. Les gens qui pratiquent FORTH sont comme les amateurs qui aiment construire leur propre matériel. Ce n'est pas qu'ils soient insatisfaits des machines disponibles ; simplement ils veulent comprendre comment ça marche, en totalité, pour contrôler le matériel. Le seul moyen d'y arriver est de le construire pièce à pièce.

FORTH correspond aux préférences de ces personnes, parce que ce sont aussi mes préférences. C'est ce que je voulais dans un langage -quelque chose dont je puisse changer tout élément si j'en ai envie. Quand ce genre de personne trouve FORTH, elle a tendance à l'aimer car il lui donne la puissance voulue. Aucun autre langage ne le fait ; en assembler vous pouvez faire ce que vous voulez, mais la difficulté est très grande. Et la possibilité d'écrire de bons logiciels n'est pas une conclusion décédée d'avance.

Maintenant je considère que FORTH a résolu le problème logiciel. Vous pouvez le faire en FORTH et vous arrêter là. Maintenant le problème de matériel. Vous ne pouvez pas construire ou contrôler votre propre matériel jusqu'au point attractif. Aussi je suis en train de construire un ordinateur FORTH et je veux me donner la possibilité d'y descendre et de manipuler les bits et définir les registres et déplacer les données sans avoir personne dans les jambes à me dire ce que je dois faire.

Je pense qu'il y a beaucoup à faire pour les prochaines années : l'approche de construction par blocs du matériel, où vous achetez des pièces élémentaires et les assemblez simplement et avez des possibilités beaucoup plus grandes parce que vous avez adapté le matériel à votre problème.

Comme beaucoup de programmeurs FORTH, vous aimez entrer dans FORTH et l'améliorer ou changer FORTH lui-même. Que pensez-vous de la tension qui existe -et je pense que c'est une forte tension- entre le groupe de normalisation et les vendeurs, qui veulent une certaine stabilité dans la définition du langage.

C'est un problème très difficile. Nous travaillons avec le groupe de normalisation pour formuler un standard portable. Ainsi le programme d'une personne peut s'exécuter sur l'appareil d'une autre personne. Je pense que nous y sommes assez bien arrivés avec le FORTH-79. Je pense que FORTH-83 y arrive aussi bien. La vraie question est de savoir combien de gens utilisent des programmes d'autres personnes ? FORTH ne semble pas être un langage comme BASIC, où vous écrivez des tas de programmes, les distribuez largement et où beaucoup de gens utilisent les mêmes programmes. FORTH est plutôt un langage dans lequel vous écrivez des programmes qu'un langage avec lequel vous exécutez les programmes écrits par d'autres. On entend toujours dire que le degré de portabilité dont il faut discuter dépend de la portabilité réellement utilisée. Mais c'est l'histoire de l'oeuf et de la poule, si vous n'avez pas un langage commun, vous ne pourrez transmettre aucun programme et ainsi de suite.

Le groupe de normalisation, de fait, est constamment (et inévitablement) en train de discuter non des mots que tout le monde utilise, mais au sujet des mots que tout le monde devrait utiliser. Cela se transforme en discussion de promotion au lieu d'une recherche d'un accord. Ceci est beaucoup plus intéressant, mais ce n'est pas le but initialement prévu : quels mots devons nous agréer qui ne vous poseront pas de problème de portabilité. Malheureusement, une grande part de la portabilité d'un programme que vous voulez exécuter sur votre IBM PC ne vous intéresse que si elle utilise les graphiques, ou le clavier d'une façon intéressante... L'interface avec l'utilisateur devient très importante et le standard ne traite pas du tout de cet aspect, parce qu'il ne le peut pas. Il y a trop de variété dans ce domaine.

Je lisais "Twice shy" ("deux fois timide"), ce matin. Il traite des ordinateurs. Ils ont deux ordinateurs, un Grantley et un Harris. Je ne sais pas si ils existent

vraiment, je suspecte que non. Un programme est écrit pour l'un d'eux et doit être exécuté sur l'autre. Ils sont tous les deux dans BASIC -qui veut l'écrire ? Je ne pense pas que Dick Francis en sache tellement au sujet des ordinateurs, mais il discute le problème du désordre, du désordre et encore du désordre et il assume que les fabricants le font probablement délibérément. Ils veulent coincer leurs clients. Ils veulent rendre leurs systèmes uniques, sur des points peut-être élémentaires, juste pour empêcher leurs clients de changer de machine, ainsi ils se mettent également en dehors d'un marché. Il n'est pas évident qu'ils y gagnent quoi que ce soit. Il n'est même pas évident qu'ils le fassent exprès.

Parmi les choses que vous voyez faire aux gens avec FORTH, lesquelles trouvez-vous les plus encourageantes, les moins encourageantes ?

Le plus encourageant est les puces FORTH qui viennent de sortir. La raison de leur sortie semble être la raison correcte, en fait. Parce que FORTH est compact, on peut implémenter les primitives de FORTH sur une puce ; c'est un des avantages d'être compact. Le fait que vous ayez une quantité de mémoire théoriquement infinie ne signifie pas que vous avez toute la mémoire du monde dans chaque situation. Ada est grossièrement en violation de l'éthique de la mémoire, mais c'est un langage prévu pour le monde moderne, qui assume la disponibilité d'espaces mémoire de largeur arbitraire. Ce n'est pas le cas quand vous voulez mettre un ordinateur dans un obus. Vous avez peu de mémoire car vous avez peu de place disponible. Ada ne programmera jamais des obus à tête chercheuse à infra-rouge. Il ne peut simplement pas le faire. Ca c'est prometteur : reconnaître que la mémoire n'est pas toujours bon marché. La mémoire est parfois bon marché, parfois très chère.

L'aspect moins encourageant est qu'il n'y a pas d'effort d'éducation à grande échelle pour former des programmeurs FORTH. Il y a pénurie de programmeurs FORTH. Et leur qualité est indéfinie. Si vous trouvez quelqu'un qui prétend connaître FORTH, il est très difficile de savoir ce qu'il connaît exactement. Si il y avait une sorte de cours, parcouru par un grand nombre de gens, vous pourriez calibrer les gens en fonction du programme. Il n'y a simplement aucun effort significatif dans cette direction.

Ca a été plus dur dans le passé que maintenant. Ca n'avait pas de sens pour une université ou un lycée d'enseigner FORTH aux gens, car il n'y avait pas de marché porteur. Maintenant le marché existe, même s'il est petit, et chaque université doit probablement avoir un cours de FORTH, ou une partie de leur cours fondamental sur les ordinateurs doit inclure FORTH. J'aimerais établir un lieu -une université est sans doute un trop grand monde- où les gens pourraient venir une semaine ou deux, un mois ou autre pour apprendre sur les ordinateurs et FORTH. Si vous suivez un cours sur les ordinateurs qui est enseigné avec PASCAL, vous apprenez moins sur les ordinateurs que sur PASCAL. Vous apprenez comment une certaine sorte de gens pensent que les programmes doivent être écrits et les algorithmes définis. Mais vous n'apprenez pas grand chose sur ce que l'ordinateur représente sous le logiciel.

Suivez le même cours en FORTH. FORTH est plus transparent, et vous devez porter votre attention plus près du problème, sur la façon de le résoudre, que sur la théorie de la programmation, qui était supposé vous faciliter les choses. Il n'est pas évident qu'un environnement de formes prédéfinies simplifie le problème ; il change le problème d'une difficulté avec les bits en une difficulté avec les virgules flottantes. Mais ce n'est pas plus facile, le problème ne disparaît pas. Il ne fait que changer.

FORTH ne le change pas trop. Si vous voulez apprendre sur les signaux réels et la façon de les interfacer avec des mémoires et des convertisseurs A/D, FORTH vous permet de traiter directement ce problème au lieu de s'occuper de compilateurs et de sous routines qui n'ont pas un intérêt fondamental.

Que fait FORTH pour les langages à venir ?

C'est une spéculation fascinante. Etant donné que FORTH est ici mais ne résoud pas tous les problèmes, qu'est-ce qui doit suivre ?

Le prochain langage devra être parlé, car la reconnaissance vocale arrive très vite. Je ne crois pas que les gens veuillent taper sur un clavier ; les claviers sont des interfaces très flexibles. Mais ils n'en veulent pas, ils ne les auront pas. Ils

veulent parler à leurs ordinateurs. Je crois que FORTH peut être un bon langage parlé si nous éliminons quelques ambiguïtés qu'il comporte.

Le langage doit avoir un contexte. Il manque quelque chose à la discussion au sujet des robots. Je perçois un robot différemment de la plus part des gens actuels. Bien sûr, je prétends que j'ai raison et qu'ils ont tort. Un robot doit percevoir son environnement. Il doit voir des choses qu'une machine ne voit pas. Tout le monde en est plus ou moins conscient, mais ils n'en voient pas les conséquences profondes. Ce qui intéresse un robot n'est pas ce qui intéresse un humain. Il ne prends pas garde à la température, par exemple : les humains sont préoccupés par la température -il vaut mieux qu'elle soit entre 18° et 24° ou il y aura des commentaires. Mais un robot doit être mortellement préoccupé par l'état exact de la charge de ses batteries et la position de la prise de courant la plus proche.

Pour qu'un robot agisse d'une façon qui nous paraisse intelligente, il faut s'assurer qu'il comprends beaucoup de choses. Par exemple un tas de concepts simples pour les humains peuvent être délicats à programmer. Le concept "sur" ou "sous" ou "dans". Le fait que de petites choses peuvent entrer dans des plus grandes si les formes s'y prêtent ; que des objets légers peuvent être posés sur une table mais que des objets lourds ne le peuvent pas. Nous devons décrire le monde d'une façon adaptée aux ordinateurs. Après quoi nous n'avons plus rien à lui expliquer. Il comprend toute une classe de choses que vous ne comprenez pas, mais vous n'avez pas à en discuter avec lui. C'est automatique.

FORTH manipule une grande partie de ce contexte sur la pile de paramètres et la pile de retour. La pile de retour est un endroit merveilleux pour stocker l'état de la machine. Vous ne pouvez pas décrire l'état de la machine à un humain sans lister le contenu de la pile. Mais le chemin que vous avez suivi dans la chaîne des définitions pour arriver où vous êtes est une assez bonne description. Dans une moindre mesure, les nombres qui peuvent se trouver sur la pile de retour sont une description d'un autre aspect de l'état courant de la machine. Les deux sont invisibles pour le programmeur et manipulés automatiquement par le langage. Un nouveau langage doit manipuler le même contexte invisible. Par exemple, vous devez avoir des noms de référence pour les objets, pour que les humains s'y reconnaissent. Cela doit être aussi évident pour la machine. Les deux points de vue doivent être maintenus en accord.

Comme inventeur de FORTH, pensez-vous que vous êtes devenu une idole ?

Ouai, plus qu'il ne faudrait. C'est une croix que je suis heureux de porter, mais il y a deux choses qui valent d'être notées. D'abord j'ai bien inventé FORTH, mais c'était il y a environ 10 ans. Il y a longtemps. Il est approprié de dire "qu'avez-vous fait récemment ?". Ensuite, depuis lors, beaucoup d'autres gens ont contribué à FORTH. Le FORTH que nous avons maintenant est bien supérieur à celui avec lequel j'ai débuté. La seule raison qui fait que j'en garde le crédit est que le reste du crédit est diffus. Si vous voulez désigner du doigt, c'est moi que vous allez désigner, parce que les autres sont trop difficiles à identifier. A mon point de vue, FORTH est un outil que j'ai développé à mon propre usage, et c'est toujours ainsi que je le vois. Le FORTH qui m'intéresse est celui-là même que j'utilise. Je continue d'essayer de trouver un jeu de mots qui le rende encore plus commode.

Par quoi êtes-vous motivé ?

Faire les choses bien. C'est le cas de beaucoup de scientifiques. Quelqu'un a peut-être déjà étudié le problème précédemment, mais vous voyez un moyen d'aller une décimale au delà. La décimale supplémentaire est une justification suffisante.

J'aime faire les choses bien. Si je vois le traitement de texte de quelqu'un d'autre, j'ai coutume de dire : "Ouai, il y a une paire de jolies choses par ici, mais je vois comment je vais l'adapter à mon propre traitement de texte, et le résultat sera pas triste". Je suis motivé par ce genre de choses. En particulier toute bonne idée qui vient de FORML (Ndt : FORML est un groupe d'utilisateurs avancés du FORTH qui organise des séminaires réguliers) ou de la communauté FORTH, je l'emploie ou je l'ajoute à mon répertoire. Pas seulement une copie, ni l'addition d'un sous programme, mais une réflexion à son sujet et peut-être un peaufinage pour qu'il soit plus compatible avec mes habitudes.

Comment faites vous quand vous voulez oublier tout ça ?

Je vais faire une promenade en montagne. L'été dernier, nous avons fait une randonnée dans la Sierra. J'ai un bateau et je marche jusqu'au port. Je ne passe pas tout mon temps au clavier, principalement parce qu'il me faut du temps pour avoir des idées qui valent d'être implémentées. FORTH est bien à cet égard. Tout ce que je veux faire peut être fait dans l'après midi. Il n'arrive pratiquement jamais qu'un projet demande beaucoup de temps de réalisation.

J'aime conduire. Je peux prendre la voiture au pied levé et conduire pendant 6 heures -d'habitude c'est pour le travail, cependant. Je me déplace pour aller voir quelqu'un, mais c'est juste une excuse pour conduire, plutôt que de téléphoner ou de prendre l'avion.

Je m'échappe dans les livres, je suppose. Ceux de science fiction sont mes favoris -c'est une bonne année pour la science fiction : Asimov et Clark. Heinlein n'a rien fait depuis Le numéro de la bête. Je suspecte que quelque chose se prépare.

Je peux me plonger dans ce genre de livre toute une journée. Mais ils ne peuvent pas écrire leurs livres aussi vite que je les lis. Même avec tous les auteurs que j'ai maintenant dans ma collection, je ne peux pas passer trop de temps à lire. Ce n'est pas bon, vous vivez dans le monde fantastique de quelqu'un d'autre et le monde réel est lui-même d'un intérêt intimidant.

J'aurais aimé être davantage impliqué dans l'effort sur les programmes de pointe. Le programme spatial, la fusion, même quelque chose d'aussi terre à terre que le nouveau système de contrôle des nouveaux avions d'attaque, qui est supposé à la fois piloter et combattre pour le pilote. Mais aucun de ces projets n'utilise FORTH dans le sens où ils le devraient ou dans le sens où je voudrais qu'ils le fassent. Et je ne suis pas prêt à entamer des batailles -c'est seulement une question de boniment. FORTH magouille toujours dans les dessous de la technologie, et il prévaudra probablement. Mais il prévaudra non par décision supérieure mais par infiltration du dessous.

Votez-vous le concept de l'informatique personnelle transformer notre travail, ou nos vies en général ?

C'est un sujet très difficile. Je ne sais pas. Je n'ai pas d'ordinateur familial. J'ai l'usage de ce LSI-11, mais je n'ai jamais acheté d'ordinateur. Il n'y a rien que puisse faire un ordinateur pour moi. Il pourrait allumer la télévision au moment prescrit, mais je peux le faire aussi bien. Je ne peux pas savoir d'avance si je vais vouloir allumer la télévision. Comme instrument de liaison avec une banque de données, je ne suis pas engagé dans une profession qui nécessite l'accès à l'information, il y a une limite à l'information dont j'ai besoin sur le temps qu'il fait ou les cours de la bourse. Ce que j'aimerais qu'un ordinateur fasse pour moi, ce sont les appels téléphoniques. Et il le peut, bien sûr. Dans quelques années, je pourrai prononcer le nom de quelqu'un et attendre de l'ordinateur qu'il me l'amène à l'écouteur. Et si il ne peut pas le trouver, de faire une note pour essayer plus tard.

Les gens dans l'ensemble -je pense que les ordinateurs ont un effet important sur leur vie, mais pas un effet profond. Pas plus, disons, que la télévision. Il se peut que les ordinateurs aient une importance culturelle, que la société ait besoin des ordinateurs pour organiser ses affaires, mais les gens -les gens n'en ont pas besoin.

Ainsi vous n'approuvez pas nécessairement la vision de troisième vague de Toffler, de grands changements, que la technologie des ordinateurs peut aider à renverser l'effet de la révolution industrielle, c'est à dire la pollution, les migrations ?

J'ai été fasciné par cette partie de ses thèses. En première lecture, j'ai en quelque sorte hoché la tête en disant "c'est juste". Mais je ne pense pas que les gens soient aussi malléables. Il y aura des gens qui s'installeront pour travailler à la maison. Comme moi. Mais je ne le fais pas à travers un réseau de communication à travers un ordinateur situé au centre de Los Angeles, je fais tout moi-même. Je suis hésitant sur la place des ordinateurs dans la société. Je n'aime pas vraiment les ordinateurs, comme on peut aimer une voiture. Je déteste surtout les ordinateurs parce qu'ils sont brutaux et maladroits à l'usage. Ils ne répondent jamais à mes désirs. Je pense à un usage des ordinateurs, tenez. Dans la Sierra il y avait des moustiques. Je verrais bien un déquilleur de moustiques à énergie solaire et à laser,

porté sur la tête. Tout moustique passant à moins d'un mètre de moi est mort !

Etes vous sûr qu'il fasse la différence entre les moustiques et les gens ?

Je pense que c'est tout à fait faisable, surtout si l'ordinateur est assez précis pour, disons, toucher l'aile droite juste à l'articulation. Cela ne prendrait pas beaucoup d'énergie mais mettrait hors circuit le moustique qui ne serait plus alors une nuisance. Je préférerais avoir un ordinateur faisant ça que descendant des ICBM (Ndt : InterContinental Ballistic Missile, missile balistique intercontinental). C'est beaucoup plus personnel, mais ce n'est pas le genre de chose que les gens ont en tête.

Un commentaire général ?

Vous devez descendre dans le monde du logiciel, le monde de l'imagination ; et vous créez votre propre monde. Vous créez vos propres problèmes et la quantité de puissance qu vous obtenez est assez effrayante. J'ai toujours rêvé ce genre de choses -j'ai toujours préféré le logiciel au matériel. Le matériel est le fléau que vous devez assembler pour pouvoir faire du logiciel. Si vous voulez voir un aspect religieux dans FORTH, c'est de dire : "avez vous besoin du matériel, pouvez-vous concevoir un moyen de représenter ces idées, de construire ces châteaux en l'air sans aucune particule en fondation ?" Je suis sûr que cette puce devrait descendre au niveau de la molécule, mais il y aura toujours cette matière au coeur des choses. Pouvez-vous prendre des champs d'énergie et, tant bien que mal, faire des logiciels ? Peut-être est-ce cela que font les couches et les couches de système d'exploitation, vous tenant si éloignés du matériel que vous en oubliez l'existence.

Traduit par J-D Dodin (T1)

CONCEPTS

Chers amis,

Comme vous pouvez le voir, ce texte n'est pas tapé sur mon traitement de texte habituel. En effet je suis en vacances, mais même là je ne vous oublie pas. J'espère que vous voudrez bien pardonner les défauts de la frappe.

Mon objet aujourd'hui est de vous parler de FORTH. Ce langage est très intéressant à enseigner car il réuni à lui tout seul les principaux concepts utilisés dans les autres langages. Nous allons essayer de rendre clairs les concepts de INTERPRÉTATION, COMPILATION, EXECUTION qui sont à la base de tous les langages.

Disons d'abord que, chaque livre ou article destiné aux débutants insiste sur le fait qu'il ne faut jamais programmer de boucle infinie. Une boucle infinie est du genre LRL OO GTO OO ou IO GOTO IO. Sur beaucoup d'ordinateurs, le seul moyen de sortir de ce genre de boucle est de faire RRESET ce qui efface souvent le programme en mémoire. Même si les appareils présentant ce problème sont maintenant rares, la recommandation est toujours valable.

Pourtant tout langage d'ordinateur repose pour l'essentiel sur une boucle infinie. En effet ce que nous appelons langage d'ordinateur est un programme qui se doit de prévoir toutes les réactions de l'utilisateur pour leur donner une réponse adéquate, même si cette réponse est un message d'erreur. Le seul moyen de ne pas manquer une réaction de l'utilisateur est de ne jamais dormir, et donc de ne jamais s'arrêter.

Tous les utilisateurs de HP-41 (en tous cas tous les membres du club qui ont une 41) savent que leur machine tombe en "sommeil léger" entre chaque pression de touche et en "sommeil profond" quand elle est éteinte. Et bien même dans cet état de "sommeil profond", la HP-41 continue à travailler! Elle teste des drapeaux internes et boucle pour savoir si on lui demande de se réveiller.

Le seul cas où la 41 s'arrête est celui où les piles ont été retirées pendant plusieurs jours. Dans ce cas le contenu des registres est également perturbé et il s'en suit le fameux MEMORY LOST.

Comme tous les langages d'ordinateurs, FORTH ne s'arrête que si l'alimentation vient à manquer. Donc FORTH est à la base une boucle infinie, comme les copains.

Installez-vous devant votre clavier favori et demandez-vous quelles grandes catégories de tâches vous pouvez y accomplir. Vous devez constater qu'il y en a trois : Ecrire un programme, exécuter un programme ou effectuer un calcul. De la façon avec laquelle votre ordinateur va réagir à ces trois tâches dépend sa classification. Vous remarquerez que j'ai le défaut fréquent de ne pas distinguer entre la machine -l'ordinateur- et son langage. C'est un défaut dû au fait que la plupart des ordinateurs que j'utilise ne possèdent qu'un seul langage. Strictement parlant, l'ordinateur ne fait rien et le langage fait tout, c'est donc le langage que nous allons classer.

La première grande catégorie de langages, certainement la plus répandue dans la pratique est la catégorie des COMPILATEURS. Curieusement, un compilateur ne fait vraiment aucune des choses que vous attendez de lui et que nous avons évoquées plus haut. Pas question de demander à un compilateur de calculer, pas question de lui demander d'exécuter un programme, pas même question de lui demander d'écrire un programme ! En fait cette dernière tâche est bien celle d'un compilateur, mais il faut lui mâcher le travail en lui proposant un programme déjà écrit et rangé sur une disquette (ou autre support). L'écriture (la frappe) du programme doit se faire avec un "utilitaire" genre traitement de texte appelé par anglicisme "Editeur de texte". Bien sûr vous n'aurez garde d'oublier de faire des fautes de frappe pendant cette étape. Ensuite vous mettez en route le compilateur en lui donnant le nom du fichier.

Le compilateur est un programme qui a pour but unique de traduire vos élocubrations en langage compréhensible par la machine, le fameux "langage machine". Tous ceux qui ont déjà essayé d'écrire à la main du langage machine savent que pour faire un programme court et rapide il faut utiliser un grand nombre d'astuces de programmation. C'est ce qui explique qu'il y a une grande variété de compilateurs. Le compilateur le plus simple est l'assembleur. Ce langage vous laisse faire presque tout le travail à la main et se contente de traduire du décimal en binaire. Mais les assembleurs perfectionnés ne se différencient pas des compilateurs spécialisés.

Un "petit" compilateur occupe 500 000 octets et donne des programmes très gros et -relativement- lents. Un compilateur commercial

peut occuper des milliards d'octets et mettre plusieurs jours à finir son travail, mais le résultat est joli, même si il ne vaut pas le travail fait main.

Le compilateur que vous pouvez vous offrir sur un ordinateur personnel ne peut-être que petit et mauvais (relativement parlant). Pour être précis, un compilateur de programme BASIC donne en général un programme "objet" (le résultat) plus gros et à peine plus rapide que le programme "source" (le programme BASIC habituel). disons deux fois plus gros et - quand même - 5 fois plus rapide. Mais ceci est obtenu au prix d'une utilisation pénible.

Nous avons vu qu'il faut d'abord taper le programme avec un éditeur de texte. Ensuite le compilateur lit ce texte et essaie de le déchiffrer. A chaque faute de frappe le compilateur vous annonce "ERROR LINE xxx" et essaie de continuer, sans grand succès en général.

Après 1/4 d'heure d'essais infructueux, le compilateur vous rend la main dans un soupir à fendre l'âme. Vous devez alors effacer le compilateur, recharger le traitement de texte, relire votre programme et essayer de comprendre où sont les erreurs. Ceux d'entre vous qui travaillent avec l'éditeur de texte CP/M savent ce que corriger veut dire.

Quand vous avez corrigé l'erreur, il est temps d'aller dîner. Il n'y a plus qu'à tenter le lendemain une nouvelle compilation. Une procédure à dégouter les programmeurs les plus endurcis, pas étonnant qu'il y ai encore tant de gens qui travaillent le langage machine à la main.

FORTH est un langage compilé. Allons nous donc être astreint à l'odieuse mécanique de la compilation ? Oui et non. Oui parce que les programmes FORTH sont tapés avec un éditeur de texte, sur le mien (FORTH HP-75) j'utilise l'éditeur de texte incorporé, qui est un des meilleurs du marché.

Oui parce que quand vous avez tapé votre programme, votre "écran" vous devez activer FORTH (en tapant simplement FORTH sur le 75 si il est chargé en mémoire), puis charger votre programme par "n°d'écran LOAD".

Non parce que sur la plupart des FORTH il y a un éditeur incorporé qui permet de taper un écran à partir de FORTH, non parce que FORTH est ultra modulaire et qu'en cas de problème de chargement FORTH vous indique quel est le "MOT" qui est incorrect. Les écrans FORTH étant courts, et les définitions qu'ils contiennent encore plus courtes, la correction est relativement facilitée. Notons cependant que les "debuggers", programmes permettant l'analyse des erreurs fleurissent dans les revues spécialisées, et que ce n'est pas sans raison.

Disons en résumé que FORTH compile le programme source par petits morceaux bien définis et que cela rend le processus de compilation-édition moins désagréable que l'on pouvait le craindre. La vitesse de FORTH est sans doute la meilleure des compilateurs disponibles sur microordinateurs, tout en restant bien inférieure à celle du langage machine ou celle obtenue avec de gros compilateurs. Par contre la place en mémoire occupée par FORTH est très faible, bien que ce soit un langage compilé. Ceci tient à sa structure, et non pas au fait que ce langage est compilé et c'est en dehors de mon sujet actuel.

La deuxième grande catégorie de programmes "langages" est la catégorie des langages INTERPRETES, dont le BASIC est l'exemple le mieux connu.

Le BASIC comporte deux parties : un Editeur de texte qui ne dit en général pas son nom et qui permet de taper des programmes et un INTERPRETEUR qui se charge de les exécuter.

Si le BASIC est dit interprété, c'est qu'il ne compile pas (Lapalisse nous voilà !). Dans la plupart des cas il ne regarde qu'à peine ce que vous tapez, juste pour voir si il y a un numéro devant (le numéro de ligne). C'est à l'exécution que le BASIC essaie de traduire en langage machine vos instructions. En fait il n'y a pas réellement traduction, à chaque instruction le BASIC se contente d'exécuter la partie correspondante de ses propres instructions.

C'est à ce moment que les erreurs sont détectées. Le Basic peut alors vous dire "SYNTAX ERROR LINE xxx" et vous afficher la ligne pour correction. C'est bien plus commode que le compilateur. C'est même bien plus commode que FORTH, mais si votre programme est long, l'erreur peut se produire après une demi-heure, ou même jamais si elle est sur une banche que vous oubliez de vérifier.

Il faut quand même dire ici qu'il y a grosso-modo deux types d'erreurs possibles : celles qui peuvent être décelées par l'ordinateur (instruction inexistante, par exemple), c'est de celles-là que je cause en ce moment, et celles qui ne peuvent pas l'être, par exemple un 2 au lieu d'un 3 dans une formule. Ce dernier type d'erreur existe dans tous les programmes, et leur correction dépasse le cadre de cet article.

Donc l'interpréteur permet la correction des erreurs plus facilement que le compilateur. Mais cette commodité se paie puisque à chaque exécution l'interpréteur effectue cette recherche des erreurs, même si elle n'est plus utile. D'où une grande perte de temps.

L'idéal serait d'avoir un langage interprété, BASIC par exemple, utilisé pour la mise au point, et un compilateur qui, après cette mise au point viendrait compiler ce programme. Cela n'existe pas chez les professionnels, mais il existe des compilateurs BASIC sur certains micro-ordinateurs. L'ennui c'est que les deux types de programmes n'ont pas exactement les mêmes besoins et que certaines instructions facilement interprétées sont difficilement compilables.

Où est FORTH là-dedans ? Et bien la structure de FORTH lui permet d'accepter des instructions de toute sortes au clavier et de les exécuter immédiatement, sans compilation. FORTH comporte donc un interpréteur, qui est d'ailleurs matérialisé par un mot d'instruction : INTERPRET (?). Ainsi il vous est loisible d'essayer au clavier une définition instruction par instruction pour en vérifier la validité. Il y a bien quelques limitations, mais elles sont minimes. Cette faculté permet à la fois d'essayer des séquences d'instructions nouvelles et de réaliser des calculs directement au clavier sans problème. Il est assez facile de disposer FORTH, par exemple, pour qu'il fonctionne comme une calculatrice HP. C'est particulièrement évident avec le FORTH du HP-71B. Une fois que l'on a testé une définition, il suffit de la réécrire avec : devant, un nom, et ; derrière pour que FORTH la compile dans son dictionnaire et qu'elle devienne immédiatement utilisable. Il est donc possible de programmer sans passer par des écrans.

A l'utilisation, au moment de l'exécution, compilateur et interpréteur se conduisent très différemment. Un programme compilé n'est rien d'autre qu'un programme en langage machine. La plupart des ordinateurs savent exécuter un programme en langage machine directement. La présence du compilateur n'est pas nécessaire à l'exécution. Du coup, même si le programme est plus volumineux

qu'un programme interprété, il y a un gros gain de place. Un programme interprété ne peut être exécuté que si l'interpréteur est présent dans la mémoire de l'ordinateur. Comme un bon BASIC occupe facilement 30 000 octets, le plus petit programme BASIC occupe au moins 30 000 octets plus les siens propres.

FORTH est un compromis entre les deux. Dans sa structure la plus évoluée, FORTH se compose d'un noyau ("Kernel") qui sera toujours indispensable et de "vocabulaires" spécialisés correspondant à peu près à nos programmes habituels. Si une application particulière nécessite une grande économie de mémoire, il est toujours possible (à condition que FORTH soit intégralement en mémoire vive) de ne conserver que les mots qui sont directement utiles dans l'application envisagée. Si vous n'avez pas de calcul de sinus, inutile d'encombrer la mémoire avec les codes de calcul du sinus, comme le fait un bête BASIC.

Les programmes FORTH peuvent être sauvegardés de deux manières : directement sous forme compilée en copiant le FORTH sur disquette avec tous les mots nouveaux. Il y a aussi de copiés tous les mots du noyau. Si votre programme est court, il y a gaspillage de place. Par contre le programme peut s'exécuter dès chargement sans aucune perte de temps. Les programmes peuvent aussi être sauves en sauvant les écrans, donc sous forme de programmes source qu'il faut

recompiler au moment de l'utilisation. Pour les petits programmes, c'est certainement la façon de faire la plus économique. Avec les FORTH évolués, il est possible de sauver uniquement des vocabulaires sous forme compilée, mais cette sauvegarde n'est pas évidente à réaliser.

En résumé, FORTH admet les trois modes de travail : compilation ("compile time"), interprétation ("INTERPRET") et exécution ("run time"). Ceci fait que quand vous voulez utiliser toute la puissance de FORTH en créant de nouvelles catégories d'instructions (ce qui est courant en FORTH), vous devez prévoir le fonctionnement de ces instructions dans les trois situations possibles.

Le débutant a certainement intérêt à utiliser un FORTH bien fourni en instructions, et à cet égard le FORTH HP-71R rends ridicule tous ses concurrents. Par contre un programmeur entraîné préférera sans doute se construire lui-même un dictionnaire minimum, avec des mots disposés par ordre de fréquence d'utilisation pour minimiser les temps d'exécution. Mais là, cela relève du libre arbitre.

Je vous laisse vous perdre dans les arcanes des modes de travail, tenez-moi au courant de vos découvertes.

Jean-Daniel Dodin (TJ)
77 rue du Cagire
31100 Toulouse
France

HP-71 BF

Maintenant que j'ai attiré votre attention avec ce titre provocateur, je vais être correct et vous dire que cet article concerne le HP-71BF. "HP-71BF" est la désignation abrégée du HP-71 avec le Module FORTH/Assembleur en place. L'idée générale est que le HP-71BF est, au moins pour sa partie interne, tout à fait le genre de machine dont les programmeurs style PPC ont rêvé dès que la première agitation autour de la nouveauté HP-41C a diminué. J'utilise la précision "partie interne" afin d'éviter le problème des caractéristiques extérieures du HP-71; on pourrait discuter à vie sur le format, le clavier et l'affichage (et aussi sur le prix) sans produire quoi que ce soit d'utile.

Je ne vais pas parler beaucoup du BASIC HP-71 non plus. La plupart d'entre vous connaissent au moins les rudiments du BASIC et sont sûrement conscients du fait que c'est très pratique pour rédiger rapidement un programme avec des E/S "conviviales", mais pas si génial pour écrire un code structuré aisément modifiable. Il est suffisant de dire que le BASIC HP-71 est une mise en oeuvre très riche du langage. Cette mise en oeuvre dépasse de loin la puissance, précision et flexibilité de la plupart des ordinateurs de table, sans parler des portables. L'héritage des calculatrices est visible dans certaines caractéristiques du BASIC 71 telles que les variables globales, les drapeaux et les formats d'affichage.

Mis à part les beautés du BASIC 71, le BASIC restera le BASIC, et le passage de la Polonaise Inverse de la HP-41 au BASIC HP-71 n'est pas évident. Nous retrouvons ce saut dans l'existence de deux revues PPC, une pour les calculatrices, (RPN), et l'autre pour les ordinateurs, (BASIC). Les idées de base et les techniques de programmation des deux langages sont suffisamment différentes pour qu'il soit dur de trouver des points communs. BASIC est un langage de haut niveau agréable qui isole le programmeur de la structure de base de l'ordinateur. La HP-41 dispose de caractéristiques de haut niveau, en particulier dans la prévention des erreurs, mais elle laisse le programmeur plus proche des bases de la machine: registres et non variables, pas de programmes, octets et autres. Son charme vient de l'aisance avec laquelle un calcul au clavier, pour lequel la RPN n'a pas d'égal, peut être transféré dans un programme sans que le programmeur ait à reprendre l'approche du problème.

Pour revenir au titre, que veut dire "programmation synthétique"? (Du fait que j'ai inventé l'expression, je pense être aussi qualifié qu'un autre pour donner une définition.) Pour la HP-41, cela signifiait au départ quelque chose de spécifique, plus précisément d'utiliser des défauts du système pour créer de "nouvelles" fonctions non prévues dans le jeu d'origine. Ces fonctions n'étaient pas nouvelles en ce sens qu'elles étendaient le jeu d'origine par l'utilisation du microcode. (La différence est que, en programmation synthétique, on prend ce qu'on trouve - si une fonction est utile, tant mieux. Aucun programme synthétique n'a donné une fonction SIZE programmable par exemple. Je reçois encore des lettres de gens qui n'ont pas saisi ce point.)

Au fur et à mesure de l'évolution de la PS sur HP-41, PS a commencé à avoir une signification plus générale: un style de programmation où les programmeurs n'acceptent ni règles ni conventions si cela leur permet d'arriver à un résultat utile. Le fait que STO M n'est pas prévu dans la HP-41 est hors de propos. C'est une fonction si utile que les programmeurs synthétiques l'ont utilisée au point qu'elle est difficilement distinguable des autres fonctions de la 41. Finalement,

l'esprit de la PS a conduit à une dissection complète de la HP-41, ayant pour résultat le développement de la programmation en langage machine chez les utilisateurs. Et il est raisonnable de dire que les résultats positifs pour HP de ce travail ont contribué à la décision de faire du HP 71 une machine "ouverte".

Le module HP-71 FORTH/Assembleur fournit un pont naturel entre la HP-41 et le HP-71. Afin de passer d'une machine à l'autre, un programmeur aimerait avoir deux choses: d'abord, un langage d'origine dans le 71 qui garde une partie du style de la RPN; et ensuite quelque chose de l'esprit de la PS, c'est-à-dire la possibilité de faire faire à un programme ce que le programmeur veut et non ce que le langage impose. Le module FORTH/assembleur fournit les deux, et à la pelle.

Voyons d'abord les généralités du module. Les deux premiers schémas donnent une vue générale des mises en oeuvre dans le module de FORTH et de l'assembleur. Les deux schémas suivants montrent de quelle manière le module et la zone mémoire vive (RAMMEV) de FORTH s'incorporent dans la mémoire du HP-71.

Pour le programmeur expérimenté en FORTH, HP-71 FORTH apparaîtra comme une jolie mise en oeuvre du standard international FORTH '83, avec des extensions pour le calcul en virgule flottante, les chaînes de caractères et le contrôle de HP-IL en E/S. Le fait que les données et adresses manipulées par le 71 FORTH soient de 20 bits au lieu des 16 habituels peut donner à penser à certains puristes, mais c'est la seule mise en oeuvre sensée vu l'adressage sur 20 bits du 71. Quelques possibilités seront nouvelles pour les amateurs de FORTH. En particulier, l'interface BASIC/FORTH, qui permet aux programmes écrits dans un langage d'avoir accès aux données et aux programmes écrits dans l'autre. De même, le traitement des virgules flottantes est nouveau, tant du point de vue de la structure de la pile virgule flottante (séparée de la pile normale), que de celui de la taille et la précision disponibles. La coexistence d'un système FORTH avec une mémoire permanente, flexible et multi-fichiers est aussi un apport significatif de ce module.

La première passerelle de la 41 au 71 est fournie par FORTH lui-même. FORTH est fondamentalement un langage RPN, plus profondément que la RPN HP-41 qui abonde en fonctions postfixées telles que STO, RCL, etc.. Les calculs arithmétiques sont faits sur une pile de taille arbitraire, en RPN pure: $3*(1+2)$ est effectué par $3\ 1\ 2\ +\ *$, la seule différence avec la 41 étant le remplacement de ENTER par un espace qui sépare les entrées numériques pour les calculs au clavier (il n'y a pas de blocage des mouvements de pile en FORTH).

L'étape suivante du passage 41-71 est l'extension virgule flottante. Il n'y a pas de standard FORTH pour l'arithmétique en virgule flottante (en fait, certains fanatiques vont jusqu'à dire que l'arithmétique en virgule flottante viole la pureté de FORTH). Nous avons donc choisi de fournir cette possibilité d'une manière qui fait intentionnellement la liaison avec les calculatrices HP. Le 71 FORTH utilise une pile VF à 4+1 registres identique à celle de la HP-41. Les 5 registres sont appelés X, Y, Z, T et L. Les nombres en VF (qui se distinguent des entiers FORTH par la présence d'une virgule) sont introduits dans le registre X en montant la pile (perte du contenu de T). Les fonctions binaires combinent X et Y dans X en descendant la pile (recopie de T dans Z). Ça vous dit quelque chose? Voici la liste des fonctions VF d'origine, dont les équivalents HP-41 sont évidents:

F+, F-, F*, F/ (Le F fait la distinction entre ces fonctions et leur équivalent arithmétique)

RDN, RUP, X<>Y, FENTER

STO, RCL, LASTX

X2, 10X, 1/X, EX, LN, LGT, YX, SQRT

Ceci ne veut pas dire que le 71BF est une 41. Il ya des différences fondamentales dans les suites de touches utilisées pour faire des calculs au clavier. FORTH s'interface avec l'utilisateur par l'intermédiaire de "l'interprète extérieur" qui attend une ligne complète d'entrée utilisateur avant l'exécution. L'équivalent HP-41 interprète chaque touche individuellement, ce qui est plus efficace que le FORTH au point de vue des appuis de touche, mais moins flexible. Par exemple, les séquences de touches pour le calcul de $\text{SIN}(1.23+45.1)/2$ seraient:

HP-41

(1)(.) (2)(3)
 (ENTER)
 (4)(5)(.) (1)
 (ENTER)
 (+)
 (SIN)
 (2)
 (/)

HP-71BF

(1)(.) (2)(3)
 (SPC)
 (4)(5)(.) (1)
 (SPC)
 (SPC)
 (F)(+)
 (SPC)
 (S)(1)(N)
 (SPC)
 (2)(.)
 (F)(/)
 (F)(.)
 (ENDLINE)

Chaque () représente un appui de touche. Cet exemple montre le 71 sous son plus mauvais jour: Je n'ai pas tiré parti des assignations de touche, ce qui fait que chaque fonction est entièrement épelée, comme (S)(I)N(.). La HP-41 est encore pire si l'on choisit des fonctions n'étant pas au clavier telle que MOD par exemple: il faudrait presser (XEQ)(ALPHA)(M)(O)(D)(ALPHA) alors que (M)(O)(D) suffit sur le 71. L'assignation de touche "+", "F+F." fait fonctionner la touche (+) de la même manière en FORTH et sur la HP-41. La similitude peut être plus forte moyennant un peu de travail.

Il y a aussi de fortes ressemblances entre la 41 et le 71BF en programmation. On peut remarquer que dans la 41 une fonction est une routine en assembleur appelée par son nom alors qu'une routine en langage utilisateur est appelée par XEQ. En FORTH, il n'y a pas de distinction entre les routines utilisateur et les fonctions: elles sont toutes appelées par leur nom. Si l'on fait abstraction de la présence ou de l'absence de XEQ, les programmes se ressemblent dans les deux langages: une suite d'appels de routines mélangées avec des entrées de données. Et ce, jusqu'à la fonction HP-41 GTO. Il n'y a pas d'équivalent direct de GTO en FORTH (FORTH se rachète par des boucles et des tests plus puissants). Si vous pouvez vous imaginer la programmation d'une 41 sans GTO, alors vous êtes murs pour programmer en FORTH. L'absence de GTO reflète la philosophie de FORTH: chaque élément d'un programme doit être un tout en lui-même, qui attend un état défini du système (configuration de pile) et fini dans un état bien déterminé. Cela oblige à une discipline de programmation qui paye en aisance de chaînage de programmes.

La plus grande partie de l'effort de PPC pour maîtriser la HP-41 a été concentrée sur l'extension du jeu de fonctions, d'abord avec des modules enfichables HP, ensuite en utilisant la programmation synthétique pour utiliser des possibilités cachées et finalement en développant un langage d'assemblage permettant de créer de vraies nouvelles fonctions. Sur le HP-71BF, la bataille est gagnée avant de commencer. Voyez ce dont vous disposez:

* Un langage dont le style fondamental est la création de nouvelles fonctions indifférenciables des fonctions ROM (70% de FORTH est écrit en FORTH)...

* Un assembleur pleine puissance, avec accès à toutes les fonctions de l'unité centrale (CPU). Le code objet est chargé automatiquement, soit chaîné dans le dictionnaire FORTH, soit en binaire HP-71, ou en fichiers d'extension de langage...

* Une machine ouverte pour laquelle toute la documentation, tous les listings du système opératoire sont disponibles aux utilisateurs.

Le langage FORTH est idéal pour la programmation synthétique. Bien que le langage ait, d'origine, certaines structures et conventions, le jeu d'outils est la pour que les programmeurs les plus ambitieux puissent faire ce qui leur plaît. Vous avez besoin d'une nouvelle fonction? Créez-la. Une nouvelle structure de données? Construisez-la. Vous avez le contrôle de la mémoire du HP-71 non seulement jusqu'au niveau du nibble (= nybble = 4 bits = 1/2 octet), mais aussi sur le comportement du langage lui-même. Vous ne pouvez pas vivre sans GTO? OK, faites vous un GTO. Voici une manière: : GTO >R :

lorsque GTO sera exécuté, l'exécution sera transférée à l'adresse sur la pile. Une fonction peu élégante, mais qui marche. L'important est de noter que le langage ne place pas de restriction à votre créativité.

Il y a plus à dire dans cette histoire: j'espère simplement maintenir la discussion PPC en marche. Pour terminer, je vais vous donner quelques indicateurs de performance pour les langages qui ont été discutés ici. Toutes ces comparaisons sont à analyser avec délicatesse, car le travail d'un certain langage se passe plus ou moins bien suivant, jusqu'à un certain point, la bonne adaptation entre le problème à tester et les possibilités du langage. Pour ce que cela vaut calculons:

$$\sum_{n=1}^{100} n(n+1)$$

** Pour la HP-41:

```

01 LBL "SUMF"
02 1.1
03 0
04 LBL 01
05 1
06 RCL Z
07 INT
08 +
09 LASTX
10 *
11 +
12 ISG Y
13 GTO 01
14 END

```

Mémoire : 29 octets

Temps d'exécution : 30 secondes

** Pour le BASIC HP-71:

```

10 X=0 @ FOR I=1 TO 100 @ X=X + I*(I+1) @ NEXT I

```

Mémoire (avec entête) : 98 octets

** Pour le FORTH HP-71:

```

: SUMI 0 101 1 DO I DUP 1+ * + LOOP ;

```

Mémoire : 42.5 octets

Temps d'exécution : 0.45 secondes

** Si vous voulez une version virgule flottante:

```

: SUMF 0. 101 1 DO I ITOF FENTER 1. F+ F* F+ LOOP ;

```

Mémoire : 63.5 octets

Temps d'exécution : 2.36 secondes

L'avantage de temps de FORTH sur BASIC vient d'un accès plus rapide aux variables et d'un déroulement de programme plus rapide. Si l'application fait beaucoup appel à la virgule flottante, il y aura peu de différence dans la vitesse des deux langages du fait que la plus grande partie du temps passé l'est dans les routines assembleurs faisant les calculs. Cela se monte en modifiant les programmes précédents pour calculer :

$$\sum_{n=1}^{100} \sin(n(n+1))$$

FORTH (en VF) prend 7.48 secondes, BASIC, 8,01 secondes. La différence de 0,53 secondes est presque la même que pour les programmes de départ. Le programme HP-41 arrive à 119.23 secondes.

Article de Bill Wickes traduit par P. Jousset (T502)

FORTH

HP-41

Je travaille depuis maintenant 5 mois sur un module permettant au HP-41 de parler en FORTH. Ce formidable module est fini.

Le CAT 2 de ce module ne donne que 'FORTH 41' qui est le nom du module et 'FORTH' qui est la seule fonction de ces 4K. Mise à part cette particularité, ce module est "classique": il ne fait que 4K mots et se place dans un quelconque des logements du 41 comme un vulgaire module de math...

Après un XEQ 'FORTH', le 41 affiche 'OK' et se place en mode ALPHA. Le clavier a été redéfini de manière à avoir accès à tous les caractères indispensables en FORTH comme le # ou le @. La sortie du mode ALPHA à pour effet de shifter automatiquement les touches situées au dessous de ENTER de manière à introduire facilement des données hexadécimales. La touche PRGM fait office de BREAK et la touche R/S sert de RETURN. La touche USER (ainsi que SHIFT USER) peut être redéfinie au gré de l'utilisateur. Pour continuer dans le même sens d'exploration du clavier, je signale que la touche ON sert toujours à éteindre le 41, puis à le rallumer en mode FORTH (pour revenir en mode RPN, il faut lancer la primitive 'HP-41').

Pour écrire des textes et des ordres à donner en forth (40 caractères au maximum), vous avez un éditeur classique: SST et BST déplace le curseur, CLX fait une insertion et la touche d'effacement une suppression. toutes les touches non shiftées sont à répétition (plus le BST qui ramène le curseur au début de la ligne s'il est maintenu).

Après avoir exploré le clavier, passons maintenant à la programmation.

Le jeu de primitives que je vais décrire paraîtra bien restreint à un habitué du FORTH, mais je rappelle que ce module ne fait que 4K. J'écrirai peut-être 4 autres kmots de manière à étendre le FORTH 41 en microcodes, mais chacun peut définir les mots qui lui manquent.

Voici donc la liste des primitives dans l'ordre chronologique, c'est à dire dans l'ordre inverse de VLIST (qui n'est pas une primitive, d'ailleurs):

F41 titre du dictionnaire des primitives;

HP-41 place le 41 en mode RPN;

CURRENT USER-variable;

CONTEXT " ;

BLK " ;

SCR " ;

BASE " ;

SO " ;

RO " ;

>IN " ;

PAD " ;

HERE " ;

ST " ;

(flags internes au FORTH);

IP " ;

RP " ;

SP " ;

@ rappel de 2 octets;

! stockage de 2 octets;

C@ rappel d'un octet;

C! stockage d'un octet;

>R mise dans la pile de retour;

R> prise dans la pile de retour;

FORTH

DUP duplication de la pile;
 DROP dépileage;
 SWAP échange dans la pile;
 OVER empilage de la seconde donnée de la pile;
 ROT rotation des 3 premières données dans la pile;
 + addition (sur 2 octets);
 - soustraction;
 * multiplication;
 / division entière;
 */ multiplication, puis division;
 MOD reste de la division;
 ABS valeur absolue;
 NEGATE changement de signe;
 = test d'égalité;
 < test d'infériorité;
 > test de supériorité;
 0= test-nul;
 0< test-négatif;
 0> test-positif;
 AND et;
 OR ou;
 NOT non;
 HEX mode hexadécimal;
 DEC mode décimal;
 EMIT émission d'un caractère à l'affichage (en ASCII);
 CR retour à la ligne;
 <# début d'une chaîne de formatage de l'affichage;
 # chiffre dans une chaîne de formatage;
 SIGN signe d'un nombre dans une chaîne de formatage;
 #S suite d'un nombre dans une chaîne de formatage;
 #> fin d'une chaîne de formatage;
 @TYPE envoi à l'affichage d'une chaîne formatée;
 @HOLD caractère ASCII dans une chaîne de formatage;
 ? affichage du contenu d'une cellule;
 @ affichage du sommet de la pile;
 ABORT réinitialisation du fichier FORTH;
 QUIT mise en mode sommeil léger;
 FIND recherche dans le dictionnaire;
 BRANCH saut relatif;
 OBRANCH saut relatif conditionnel;
 ' recherche de l'adresse d'un mot du dictionnaire;
 KB@ appel du code de la touche pressée;
 @ stockage dans le dictionnaire et modification de HERE;
 ." affichage d'un texte;
 C, comme ", sur 1 octet;
 TONE fonction musicale (utilisation analogue à TYPE).

Ces 68 fonctions font du FORTH 41 un FORTH assez léger, mais tout y est pour en faire un FORTH 79, au minimum...

Un mot sur la structure de la RAM FORTH: le module FORTH 41 prend systématiquement toute la place disponible en mémoire centrale (255 registre au maximum) et toutes les X-MEMORY (en 3 blocs dans le meilleur des cas).

Mais cela fait 5,8koctets si je compte bien...

Je termine donc actuellement (25 aout 1984) ce module pour qu'il soit plus performant et plus complet (il fera sans doute 8K). Cette ROM sera peut être sous forme de module enfichable, mais je ne peut pas encore me prononcer.

T270 SERGE VAUDENAY

Stockage de masque par WATA a WALL d'imagier. Quittons ceux qui seraient intéressés par une EPPON (E module 400F) ou un module client (à environ 1000F) n'importe sans envoyer lajust mais très vite. Voir aussi "Sur le Marché"

J. D. Dany

UTILITAIRES

Eric AUBOURG (T366)

12 rue des 4 vents

79000 NIORT

Chers membres de PP2-T

Voici mes dernières créations en matière d'écrans FORTH :
Tout d'abord un traitement de chaînes alpha. Il permet la
définition de variables de type chaînes et leur manipulation. Trois
mots non présents sur tous les forths sont nécessaires : PICK, F DUP et
DROVE. J'en ai donné la définition. DROVE est l'équivalent de CMOVE, mais
en commençant par la fin et en descendant. Il est nécessaire lorsque le bloc
d'arrivée a une adresse supérieure au bloc de départ et le cherche.

Les noms utilisés sont les mêmes que les équivalents en Pascal UCSB.
Les commentaires et surtout l'exemple sont je pense suffisants pour comprendre
leur usage. Je tiens néanmoins à préciser au sujet de "\$" qui permet
d'utiliser une constante alpha qu'une seule constante est accessible
à la fois : Si vous faites "\$ C1" "\$ C2" \$. \$. vous obtiendrez C2 C2.
Pour les intéressés, on aurait pu concevoir cette unité uniquement avec
"\$" et un mot \$! au lieu de "\$=", et utiliser les bloc buffers au lieu
de PAD pour avoir plusieurs constantes. Mais c'est une autre histoire...

La structure d'une variable chaîne est la suivante : le PFA contient
en LSB (octet de poids faible) la longueur maximale et en MSB la
longueur effective. Les caractères sont stockés à partir de PFA+2.

Les deux autres utilitaires sont :

- une extension de l'éditeur, U, qui autonumérote et évite de P...LP...etc
Il faut valider par ENTER l'effacement de l'écran courant après "SCR#n", puis
taper les lignes. ENTER seul stoppe l'autonumérotation.

- Un mot de décompilation, pour les curieux : Décompile "nom" affichera
la définition de "nom". Inpartant : c'est vous, en appuyant sur une touche,
qui arrêtez la décompilation (faites le dès que vous voyez des
caractères bizarres.) c'est un pis-aller mais rien ne permet à priori
de déterminer la fin d'un mot : on peut tester ;S mais il peut
apparaître en plein milieu d'un mot, et certains mots se terminent
par un BRANCH... 2 astuces utilisées méritent une explication :

- [LATEST PFA CFA @] LITERAL compile la valeur du CFA du dernier mot, DECOMPILER, défini par ":" : pas question de décompiler une variable!
- LIT nom compile comme un nombre le CFA de nom : à l'exécution, l'interpréteur considère le nombre qui suit LIT comme un nombre à empiler et non comme un CFA à exécuter.

Heuseuse Programmation

T 356

Subow

FORTH/HP-75

Juste une petite remarque au sujet du HP75 et de certains mots prétendument manquants (voir MICRO-REVUE n°1 p 56) : en fait, les mots NEXT, POP, PUSH et PUT entre autres NE SONT PAS des mots FORTH : citons le manuel d'installation : "Ils ne sont pas directement exécutables, mais sont des points de levée FORTH après du code machine". En clair : leurs noms sont factices, ils n'apparaissent pas au dictionnaire, mais existent sur tous les FORTHS, y compris celui du HP75.

T 366

* MOTS NON STANDARDS UTILISÉS

: PICK 2 * SPB + @ ;

(.. n → .. cellule n)

: FDUP 3 PICK 3 PICK 3 PICK ;

(e b a → a b c a b c)

: DMOVE

(d a l) transfert de bloc décroissant

>R R + 1 - SWAP R + 1 - SWAP @ R >

(d+1 a-1 @ l)

DO

OVER @ OVER C! 1 - SWAP 1 - SWAP

transfert d'un octet et décrémentation octet suivant

-1 + LOOP

;

* UNITÉ "STRING"

VOCABULARY STRING IMMEDIATE

A, B... désignent une chaîne définition du vocabulaire

STRING DEFINITIONS

: DEFSTR <BUILD6 DUP C, 1+ ALLOT DOES> ;

(long max) définition d'une chaîne

: STOS ROT 2DUP @ > B1 ?ERROR 1+ 2DUP

(A adx B) Utilitaire de stockage

C! 1+ ROT SWAP ROT CMOVE ;

: (\$=) R COUNT DUP 1+ R > + >R STOS ;

(A) Utilitaire d'affectation

: \$=" 34 STATE @

(A) affectation.

IF COMPILER (\$=) WORD HERE @ 1+ ALLOT

mode compilation
mode execution

ELSE WORD HERE COUNT STOS THEN

; IMMEDIATE

: \$, 1+ COUNT TYPE ;

(A) affichage

: \$CAR OVER 1+ @ OVER < B1 ?ERROR + 1+ ;

(A n → adx) accès caractère

: LENGTH 1+ @ ;

(A → l) longueur

: POS 1+ COUNT ROT 1+ COUNT DUP

(A B → n) cherche A dans B

>R MATCH R > - 1+ SWAP @ =

IF DROP @ THEN ;

retourne @ si pas trouvé

: CONCAT ROT 1+ COUNT DUP >R HERE ROT SWAP ROT (A BC) C ← A@B

CMOVE SWAP 1+ COUNT HERE R + ROT SWAP ROT Transfert de A après Here, B à la suite, et stockage en C

: COPY >R >R + 1+ R > R > ROT ROT STOS ;

(A p l B) extraction B ← A[p..p+l]

: DELETE FDUP + SWAP LENGTH 1+ > B1 ?ERROR

(A p l) suppression

ROT 1+ DUP @ DUP >R 3 PICK - DUP @ < B1

Tests de validité

?ERROR OVER C! 1+ ROT 1 - R > OVER - >R

longueur ← longueur - l

+ DUP >R + R > R > CMOVE ;

Décalage.

```

: INSERT ROT 2DUP DUP @ SWAP LENGTH ROT (A B) Insertion de B dans A
  LENGTH + DUP >R < B ! ?ERROR R> OVER 1+ à partir de p
  C! ROT 2DUP SWAP LENGTH SWAP - DUP @< Tests de validité
  B ! ?ERROR 1+ >R + 1+ 2DUP SWAP LENGTH +
  OVER R> SWAP >R MOVE 2+ R> OVER 1 - Décalage de A
  C@ MOVE; Transfert de B
: ($) PAD DUP 265 OVER C! ; utilisé par "$"
: "$" STATE @ (→ A) Constante
  IF COMPILÉ ($)
  ELSE ($)
  THEN [COMPILÉ] $="

```

; IMMEDIATE
FORTH DEFINITIONS

* EXEMPLE

Entrée

Affichage

STRING	OK
50 DEFSTR A	OK
A "\$" TRAITEMENT ALPHA	OK
A \$.	TRAITEMENT ALPHA OK
A LENGTH .	16 OK
A 6 \$CAR @ EMIT	6 OK
" ENT" A POS .	8 OK
" ZZZ" A POS .	0 OK
A 12 "\$" DE CHAINES " INSERT A \$.	TRAITEMENT DE CHAINES ALPHA OK
A "\$" NUMERIQUES" A CONCAT A \$.	TRAITEMENT DE CHAINES ALPHANUMERIQUES OK
A 12 11 DELETE A 26 1 DELETE A \$.	TRAITEMENT ALPHANUMERIQUE OK

UTILITAIRES



Extension de l'éditeur

Pour l'éditeur, voir par exemple le FIG-FORTH INSTALLATION MANUAL.

EDITOR DEFINITIONS DECIMAL

: U

Autonumerotation (écran courant).

FORTH ." SCR # " SCR ? KEY 13 - . Validation.

IF ;S THEN ≠ ENTER → arrêt.

SCR @ CLEAR 16 @ efface l'écran.

DO CR I. QUERY TIB @ @ @ = entrée de la ligne.

IF CR R> R= DROP DROP ;S THEN ENTER seul → arrêt.

I EDITOR P Stockage de la ligne.

LOOP ligne suivante.

; FORTH DEFINITIONS

Décompilation

: DECOMPILE

CR [COMPILE] ' DUP CFA @ Recherche du mot.

[LATEST PFA CFA @] LITERAL - 80 ?ERROR Défini par ":" ?

BEGIN Affichage du mot.

DUP @ DUP 2+ NFA ID. mot avec paramètre

DUP LIT (Loop) = OVER LIT BRANCH = OR numérique ?

OVER LIT OBRANCH = OR OVER LIT LIT = OR

OVER LIT' (+ Loop) = OR oui → affichage du paramètre.

IF DROP 2+ DUP ? ELSE mot avec paramètre

LIT (.") = IF 2+ COUNT 2DUP + CFA ROT alpha? oui → aff. change.

ROT TYPE SPACE mot suivant

THEN THEN 2+ fin ?

?TERMINAL

UNTIL CR DROP

JEDI

Monsieur,

Notre association a le plaisir de vous envoyer quelques programmes pour votre numéro "SPECIAL FORTH".

Le premier programme est une fonction "CATALOGUE" qui reprend le fonctionnement de VLIST, mais en améliorant la présentation. La définition a été étudiée sur ORIC 1 et fonctionne pour toutes machines. Les mots apparaîtront alignés verticalement, chaque mot étant précédé d'un numéro. Le premier numéro concernera le dernier mot créé dans le vocabulaire de contexte, le second numéro concernera l'avantdernier et ainsi de suite. Une borne, sous forme de constante, permet en outre de limiter l'édition du vocabulaire à un mot spécifique, permettant ainsi des listages partiels.

```
' LIT LFA CONSTANT BORNE
```

```
: CATALOGUE ( --- )
```

```
1 CONTEXT @ @
```

```
BEGIN
```

```
CR SWAP DUP
```

```
3 .R 1+ SWAP
```

```
DUP SPACE ID.
```

```
PFA LFA DUP @
```

```
SWAP BORNE =
```

```
?TERMINAL OR
```

```
UNTIL
```

```
DROP DROP ;
```

Le listage complet du vocabulaire s'obtient en initialisant la constante BORNE en y implantant le LFA du 1er mot du dictionnaire Forth, c'est à dire LIT. C'est ce qui est réalisé dans la 1ère ligne du listing. Pour limiter l'édition du vocabulaire jusqu'à un mot spécifique, il suffit d'implanter le LFA de ce mot dans le PFA de la constante BORNE. Le PFA d'un mot est obtenu grâce au mot " ". On désire arrêter l'édition du vocabulaire à DUP, il faut donc initialiser la valeur de BORNE avec le LFA de DUP comme suit:

```
' DUP LFA ' BORNE !
```

L'astuce décrite ci-dessus permet de modifier le contenu d'une constante. Il est fortement déconseillé de modifier les valeurs de 0, 1, et 2, constantes prédéfinies dans Forth, car celles-ci sont utilisées dans le système. Exemple de modification de valeur de constante:

```
1983 CONSTANT ANNEE
```

peut être modifiée le 1er janvier de l'année suivante en tapant:

```
1984 ' ANNEE !
```

Exemple d'exécution de CATALOGUE:

```
1 JOUR
2 ENDCASE
3 ENDOF
4 OF
5 CASE
6 TASK ...etc...
```

Le second programme proposé est la fonction **DUMP**, fonction disponible d'origine sur de nombreux systèmes professionnels. Cette fonction permet la visualisation des octets contenus en mémoire, et ce, grâce à une matrice de chiffres hexadécimaux et de caractères.

```

: LIGNE ( adr --- adr c1 ... c8 )
DUP 8 + SWAP DUP 0
HEX CR <# # # # # #> TYPE SPACE
DO
  I C @ S->D
  <# # # #> TYPE SPACE
LOOP ;

: CARACTERES ( adr --- c0 ... c7 )
DUP 8 + SWAP
DO
  I C @ DUP 31 >
  OVER 127 < AND
  IF
    EMIT
  ELSE
    DROP 46 EMIT
  ENDIF
LOOP ;

: DUMP ( adr --- 20 lignes affichées )
DUP 160 + SWAP CLS
DO
  I DUP LIGNE CARACTERES
  8 +LOOP
CR ;

```

Le mot **DUMP** s'emploie en déposant sur la pile de données l'adresse du début du bloc à **DUMPER**. Exemple: 2360 **DUMP**, exécutera la fonction **DUMP** en affichant sur 20 lignes le contenu de la mémoire à partir de l'adresse 2360. Chaque ligne d'affichage commence par l'adresse du premier octet affiché sur la ligne correspondante, suivie des huit octets en hexadécimal, et des huit caractères correspondants aux octets. Cependant, un test est effectué afin de déterminer si les codes des caractères sont en dehors de l'intervalle 32 à 127, auquel cas seul un point est affiché. Si on tente d'afficher l'ensemble des caractères, on risque de perturber quelque peu l'affichage des données.

Le troisième programme permet de diviser deux nombres et d'afficher le nombre de décimales voulues.

```

: DEC ( n1 n2 n3 --- n4,nnn(n3) )
ROT ROT DUP >R /MOD
S->D <# #S #> TYPE 46 EMIT SWAP R>
SWAP 0
DO
  10 * OVER
  /MOD S->D <# #S #> TYPE SWAP
LOOP
DROP DROP ;

```

Les paramètres à déposer sur la pile avant d'effectuer **DIV** sont les suivants:

- n1 dividende
- n2 diviseur
- n3 nombre de décimales à afficher.

Exemple: 22 7 100 **DIV** affichera le résultat de 22 divisé par 7 suivi de 100 décimales.

Le dernier programme est en fait une extension au langage Forth permettant d'utiliser une structure de contrôle équivalente à celle disponible en langage PASCAL grâce aux mots CASE, OF, ENDOF, et ENDCASE.

```

: CASE
?COMP CSP @ !CSP 6 ; IMMEDIATE
: OF
6 ?PAIRS COMPILE OVER COMPILE = COMPILE
OBRANCH HERE 0 , COMPILE DROP 7 ; IMMEDIATE
: ENDOF
7 ?PAIRS COMPILE BRANCH HERE 0 , SNAP 2 [COMPILE]
ENDIF 6 ; IMMEDIATE
: ENDCASE
6 ?PAIRS COMPILE DROP
BEGIN
SP@ CSP @ = 0=
WHILE 2 [COMPILE] ENDFI
REPEAT
CSP ! ; IMMEDIATE

```

Ces structures, bien que non-standards, sont disponibles d'origine sur les machines suivantes; THOMSON T07, HECTOR HRX. Pour les autres machines, ces définitions ne sont pas toujours disponibles en options, ou alors ne sont pas portables (cas de l'ORIC 1). Les définitions ci-dessus sont intégralement portables.

Exemple d'utilisation de CASE...OF...ENDOF...ENDCASE :

```

: SEMAINE
CASE
1 OF ." LUNDI" CR ENDOF
2 OF ." MARDI" CR ENDOF
3 OF ." MERCREDI" CR ENDOF
4 OF ." JEUDI" CR ENDOF
5 OF ." VENDREDI" CR ENDOF
6 OF ." SAMEDI" CR ENDOF
7 OF ." DIMANCHE" CR ENDOF
ENDCASE

```

Le mot SEMAINE s'emploie en déposant un nombre sur la pile de données, nombre compris entre 1 et 7 et en tapant SEMAINE. Exemple: 3 SEMAINE affichera MERCREDI.

L'avantage de la structure CASE...ENDCASE est de réaliser des tests de la même manière que la structure IF...ENDIF, mais ne poursuit par les tests si une condition est vraie. Dans l'exemple précédent, les tests ont été effectués jusqu'à ce que la condition correspondante à 3 OF...ENDOF soit exécutée, puis le programme s'est poursuivi en passant directement à ENDCASE sans effectuer les autres tests. Il s'ensuit un gain de temps considérable ainsi qu'une simplification d'écriture évident (pour la compréhension également).

L'Association JEDI (que le Forth soit avec vous) publie une brochure mensuelle. Les buts de l'association sont la promotion des nouveaux langages informatiques.

Les cotisations sont de 80 Fr pour 6 mois et de 150 Fr pour un an, abonnement à la revue JEDI inclus, ainsi que la possibilité de commander des logiciels ou des matériels à prix club. Pour tout renseignement ou inscription, écrire à JEDI, 8 Rue Poirier de Narçay 75014 PARIS.

Jean THOMAS
Ecole de Garçons
30700 - UZES
T446

UTILITAIRES

Je ne présenterai pas le langage FORTH. Les notions de base peuvent être facilement acquises par la lecture de la dizaine d'ouvrages disponibles en français.

Je propose plutôt de montrer quelques facettes de FORTH en examinant de petits programmes:

- cinq programmes "maison" de listage d'un contenu mémoire. (DUMP - DUMPB - LISTING - LISTINGB - LISTINGT)
 - un court programme test donnant un aperçu des performances de FORTH dans les calculs en virgule flottante. (selon une version personnelle).
- Egalement, il sera mis en évidence le comportement curieux qu'a, parfois, la fonction SINUS.
- et pour terminer, un jeu de Master Mind adapté d'un article américain paru dans FORTH DIMENSIONS.

** DUMP - LISTING **

Les programmes DUMP, DUMPB, LISTING, LISTINGB, LISTINGT réalisent des vidages mémoire.

Ils affichent adresses et données en octal (langage naturel du microprocesseur LSI 11/23). Ils pourraient être adaptés à l'hexadécimal.

Ils présentent quelques particularités:

- les nombres sont justifiés à droite.
- l'affichage des adresses commence toujours à ...00,
- ...20, ...40, ...60 (octal) quelle que soit l'adresse
- entrée. Il est donc aisé de se retrouver dans les colonnes. (voir les exemples donnés pour DUMP et DUMPB)
- ils bénéficient de la fonction PAUSE.
- DUMP et DUMPB ne se terminent pas par "OK", quelquefois peu souhaité sur l'imprimante. Il suffit pour atteindre ce résultat, de sortir de DUMP ou DUMPB par QUIT.

* LISTING et DUMP affichent des mots de 16 bits.

* LISTINGB et DUMPB affichent des mots de 8 bits.

* LISTINGT traduit les octets > 32 (base 10) en ASCII.

*** La fonction PAUSE.

Elle permet d'interrompre un listage puis de le poursuivre ou l'arrêter.

Sur la frappe d'une touche quelconque, le déroulement du listage s'interrompt. Sur la frappe suivante, il repart mais sur 2 frappes rapprochées, il s'arrête définitivement.

PAUSE suivi de ?TERMINAL fournit un drapeau "faux" sur le 2ème appui isolé et un drapeau "vrai" sur les 2ème et 3ème appuis rapprochés.

Voici un petit programme permettant de tester PAUSE:

```

: TEST 1001 0 DO I . PAUSE ?TERMINAL
      IF LEAVE
      ENDIF
LOOP ;

```

(on affiche simplement les nombres de 0 à 1000 - et non 1001)

En cas d'ennui, il faut suspecter ?TERMINAL, défini en langage machine et variant selon les versions. Dans la version que j'utilise (FIG PDP-11 John S. JAMES) ?TERMINAL fournit:

- un drapeau "faux" (=0) si aucune touche n'a été enfoncée.
- un drapeau "vrai" (=ASCII) sur une touche enfoncée.
- un drapeau "vrai" égal à 120000 (base 8) sur la touche BREAK enfoncée.
- il vide le port d'entrée du clavier (le caractère est en pile et sert de drapeau).

En cas de difficulté, voici une version PAUSE extraite de "All about Forth" de Haydon Vol.1 (chez Mountain View Press, l'éditeur et le fournisseur US N°1 de documents Forth):

```

: PAUSE ?TERMINAL IF KEY DROP 1000 0 DO LOOP
  BEGIN ?TERMINAL
  UNTIL KEY DROP 2000 0 DO LOOP
  ENDIF ;

```

Ici, ?TERMINAL fournit un drapeau mais ne vide pas le port d'entrée du clavier: le caractère sera récupéré par KEY.

```

*****
*  Du l'on inclut PAUSE dans des mots appartenant au dictionnaire.  *
*****

```

Des mots tels que INDEX, VLIST produisent des listages qui, certes, peuvent être interrompus, mais seulement de manière définitive. Dans ces conditions, il serait agréable de bénéficier de la fonction PAUSE. Un tel souhait est aisément réalisable, ce qui montre bien les possibilités d'adaptation de Forth.

Les mots à utiliser sont PATCH-P et WRITE-PATCH figurant dans le "screen" N°93. Les mots INDEX et VLIST contiennent d'origine ?TERMINAL, qu'il suffira de dérouter vers PATCH-P qui lui, contient PAUSE suivi de ?TERMINAL. Dans sa propre version, il faudra relever l'adresse de ?TERMINAL figurant dans INDEX et VLIST et calculer l'offset avec les CFA respectifs de ces deux mots (24 et 54 dans ma version). WRITE-PATCH se charge de calculer les adresses résultantes et remplace l'adresse de ?TERMINAL par celle de PATCH-P, réalisant ainsi le détournement.

Si l'on se contente de redéfinir un mot, seuls les futurs programmes utiliseront la nouvelle version (la recherche dans le dictionnaire se faisant en commençant par les derniers mots entrés). Les anciens mots déjà compilés et appartenant au vocabulaire de base, continueront à employer l'ancienne version.

Par le procédé du détournement, tous les mots, anciens et nouveaux, feront appel à la version "patchée".

A vrai dire, INDEX et VLIST sont plutôt utilisés en commande directe et ne font, généralement, pas partie d'un programme. Dans ce cas particulier, une redéfinition serait toujours opérante en commande directe.

```

SCR # 90
0 ( DUMP )
1 0 VARIABLE V-JUST      0 VARIABLE V-SPACES      OCTAL
2 0 VARIABLE BASE-OLD
3 : A-LINE CR DUP 6 U.R 2 SPACES V-SPACES @ SPACES V-JUST @ +
4     20 V-JUST @ DO DUP @ 7 U.R 2+ 2DUP SWAP U>
5     IF 2DROP CR BASE-OLD @ BASE ! QUIT
6     ENDIF
7     I 6 = IF 2 SPACES ENDIF 2 /LOOP ;
8 : DUMP BASE @ BASE-OLD ! OCTAL =CELLS SWAP =CELLS
9     DUP 77 AND 20 MOD DUP DUP V-JUST !
10    IF 2 / 7 * DUP 32 > IF 2+ ENDIF
11    ENDIF V-SPACES ! V-JUST @ -
12    BEGIN A-LINE 0 V-JUST ! 0 V-SPACES ! PAUSE ?TERMINAL
13    UNTIL 2DROP ;
14
15    DECIMAL -->

```

```

SCR # 91
0 ( DUMPB )
1
2 : A-LINEB CR DUP 6 U.R 2 SPACES V-SPACES @ SPACES V-JUST @ +
3     20 V-JUST @ DO DUP C@ 4 U.R 1+ 2DUP SWAP U>
4     IF 2DROP CR BASE-OLD @ BASE ! QUIT
5     ENDIF
6     I 7 = IF 2 SPACES ENDIF 1 /LOOP ;
7 : DUMPB BASE @ BASE-OLD ! OCTAL =CELLS 1+ SWAP =CELLS
8     DUP 77 AND 20 MOD DUP DUP V-JUST !
9     IF 4 * DUP 36 > IF 2+ ENDIF
10    ENDIF V-SPACES ! V-JUST @ -
11    BEGIN A-LINEB 0 V-JUST ! 0 V-SPACES ! PAUSE ?TERMINAL
12    UNTIL 2DROP ;
13
14
15    DECIMAL -->

```

```

SCR # 92
0 ( LISTING LISTINGB LISTINGT )
1 : LISTING CR BASE @ >R OCTAL 2+ =CELLS SWAP =CELLS
2     DO I 6 U.R I @ 9 U.R
3     PAUSE ?TERMINAL IF LEAVE ENDIF CR 2 /LOOP R> BASE ! CR ;
4 : LISTINGB CR BASE @ >R OCTAL 2+ =CELLS SWAP =CELLS
5     DO I 6 U.R I 1+ C@ 6 U.R I C@ 5 U.R
6     PAUSE ?TERMINAL IF LEAVE ENDIF CR 2 /LOOP R> BASE ! CR ;
7 : ?NON-PRINTING DUP 32 < SWAP 126 > OR ;
8 : LISTINGT CR BASE @ >R OCTAL 2+ =CELLS SWAP =CELLS
9     DO I 6 U.R I 1+ C@ 6 U.R I C@ 5 U.R
10    3 SPACES I 1+ C@ DUP ?NON-PRINTING
11    IF DROP 32 ENDIF EMIT
12    3 SPACES I C@ DUP ?NON-PRINTING
13    IF DROP 32 ENDIF EMIT
14    PAUSE ?TERMINAL IF LEAVE ENDIF CR 2 /LOOP R> BASE ! CR ;
15
16 OK

```

SCR # 93

```
0 ( DUMP LISTING - mots à définir )
1 : PAUSE ?TERMINAL IF KEY DROP
2 30000 0 DO LOOP
3 ENDIF ;
4 : 2DUP OVER OVER ;
5 : 2DROP DROP DROP ;
6 : U> SWAP U< ;
7 : U.R 0 SWAP D.R ; justification à droite nbr 16 bits non signé
8 -CELLS: à supprimer (spécifique du microprocesseur LSI 11/23)
9
10 *****
11 Pour inclure la fonction PAUSE dans les mots INDEX et
12 VLIST du vocabulaire de base:
13 : PATCH-P PAUSE ?TERMINAL ;
14 : WRITE-PATCH ' PATCH-P CFA/DUP ' INDEX 24 + ! ' VLIST 54 + ! ;
15 (Il faut adapter 24 et 54 à sa propre version - voir texte)
OK
```

OCTAL	101446	101512	LISTINGB	101446	101512	LISTINGT
101446	40	75		101446	40	75
101450	115	40		101450	115	40 M
101452	101	40		101452	101	40 A
101454	123	40		101454	123	40 S
101456	124	40		101456	124	40 T
101460	105	40		101460	105	40 E
101462	122	40		101462	122	40 R
101464	40	40		101464	40	40
101466	40	115		101466	40	115 M
101470	40	111		101470	40	111 I
101472	40	116		101472	40	116 N
101474	40	104		101474	40	104 D
101476	75	40		101476	75	40 =
101500	75	75		101500	75	75 =
101502	40	75		101502	40	75 =
101504	5	46		101504	5	46 &
101506	104	206		101506	104	206 D
101510	123	101		101510	123	101 S A
101512	105	110		101512	105	110 E R H

OK

OK

DECIMAL 33574 33610 LISTING

101446	20075
101450	46440
101452	40440
101454	51440
101456	52040
101460	42440
101462	51040
101464	20040
101466	20115
101470	20111
101472	20116
101474	20104
101476	36440
101500	36475
101502	20075
101504	2446
101506	42206
101510	51501
101512	42510

OK

```

OCTAL 101446 101512 DUMP
101440                                     20075 46440 40440 51440 52040
101460 42440 51040 20040 20115 20111 20116 20104 36440
101500 36475 20075 2446 42206 51501 42510

101446 101512 DUMPB
101440                                     75 40 40 115 40 101 40 123 40 124
101460 40 105 40 122 40 40 115 40 111 40 116 40 104 40 40 75
101500 75 75 75 40 46 5 206 104 101 123 110 105

```

```

*****
** FORTH ET LE CALCUL FLOTTANT **
*****

```

On lit quelquefois cette assertion: "Forth n'est pas destiné au calcul flottant". Il est exact que ce mode de calcul ne fait pas partie du noyau de base. D'ailleurs, beaucoup de calculs complexes peuvent se faire à partir des nombres entiers, double précision - ou plus. Voir à ce propos, le chapitre 5 de l'ouvrage "Débutez en Forth" de L. Brodie. On gagne certainement en temps d'exécution.

Toutefois, rien n'interdit de concevoir un calcul flottant en Forth. Personnellement, j'ai implémenté une version sur 64 bits soit environ 17 chiffres décimaux dont 15 exacts dans le pire des cas. Cette précision s'étend aux fonctions SIN, LOG etc, alors que souvent la double précision n'est effective que sur les quatre opérations.

A titre de démonstration, calculons le SIN d'un angle de 1 radian puis avec cette valeur, l'ARCSIN, de façon à retrouver l'angle d'origine. (TEST-1 screen 450).
Calculons également PI au carré (par les puissances et non par PI*PI) puis extrayons la racine carrée pour tenter de retrouver PI (TEST-2):

```

SCR # 450
0 ( Essais en calcul flottant )
1
2 : TEST-1 CR 16 SPACES FDUP ." x =" 20 TAB F.
3 CR 9 SPACES SIN FDUP ." y=SIN(x) =" 20 TAB F.
4 CR 10 SPACES ASIN ." ASIN(y) =" 20 TAB F. CR ;
5
6 : TEST-2 CR 15 SPACES PI FDUP ." PI =" 20 TAB F.
7 CR 13 SPACES F2 Y^X FDUP ." PI^2 =" 20 TAB F.
8 CR 5 SPACES SQR ." RAC. de PI^2 =" 20 TAB F. CR ;
9
10
11

```

FI TEST-1

```

      x = 1
y=SIN(x) = 0.841 470 984 807 896 4
      ASIN(y) = 1

```

OK

TEST-2

```

      PI = 3.141 592 653 589 792
      PI^2 = 9.869 604 401 089 356
RAC. de PI^2 = 3.141 592 653 589 792

```

OK

On peut constater que ce petit jeu de va-et-vient ne diminue pas la précision qui reste de 15 chiffres. Le 16ème est faux: 5 au lieu de 4 pour le SIN, 8 au lieu de 6 pour le carré de PI et 3 au lieu de 2 pour la racine carrée de PI².

J'ai formaté les nombres par tranches de 3 chiffres ce qui facilite la lecture des nombres longs. La même possibilité existe à l'entrée des nombres en format flottant, l'espace n'étant pas considéré comme fin du nombre. Bien entendu, en dehors de ce cas particulier qui ne s'applique qu'à ma version de flottant, Forth exige toujours la présence d'au moins un espace entre les mots.

** SINUS ET ANOMALIE **

On sait que, pour des angles très petits, la valeur du SIN est égale au nombre qui mesure l'angle en radians.

Essayons le programme BASIC ci-dessous:

```
10 X= 1/3
20 FOR K=1 TO 41
30 PRINT X TAB(20) SIN(X)
40 X=X/10
50 NEXT K
OK
```

Note de l'édition :
le matériel HP me
tombe pas dans le
pneuveau - c'est sa
la classe -

```
RUN
.333333333      .327194697
.033333333      .0333271608
3.3333333E-03   3.33332716E-03
3.3333333E-04   3.3333327E-04
3.3333333E-05   3.3333313E-05
3.3333334E-06   3.3333313E-06
3.3333334E-07   3.3331027E-07
3.3333334E-08   3.3332817E-08
3.3333334E-09   3.33156734E-09
3.3333334E-10   0
3.3333334E-11   0
3.3333334E-12   0
3.3333333E-13   0
3.3333334E-14   0
BREAK IN 30
OK
```

Pour de petits angles, les résultats ne sont pas conformes à la théorie mathématique: à la puissance -09, 3 chiffres seulement sont exacts et aucun à la puissance -10.

Ecrivons le même programme en Forth et exécutons-le:

```
F1 F3 F/ FVARIABLE X
: XX CR 42 1 DO X F@ FDUP F. 35 TAB SIN F. CR
  FIU-OFF X F@ F10 F/ X F!
LOOP;
```

```

XX
0.333 333 333 333 333 3      0.327 194 696 796 152 2
0.333 333 333 333 333 3 E-01  0.333 271 608 367 536 1 E-01
0.333 333 333 333 333 3 E-02  0.333 332 716 049 725 6 E-02
0.333 333 333 333 333 3 E-03  0.333 333 327 160 493 8 E-03
0.333 333 333 333 333 3 E-04  0.333 333 333 271 604 9 E-04
0.333 333 333 333 333 3 E-05  0.333 333 333 332 716 0 E-05
0.333 333 333 333 333 3 E-06  0.333 333 333 333 327 1 E-06
0.333 333 333 333 333 3 E-07  0.333 333 333 333 333 2 E-07
0.333 333 333 333 333 3 E-08  0.333 333 333 333 333 3 E-08
0.333 333 333 333 333 3 E-09  0.333 333 333 333 333 3 E-
0.333 333 333 333 333 3 E-10  0.333 333 333 333 333 3
0.333 333 333 333 333 3 E-11  0.333 333 333 333 33
0.333 333 333 333 333 3 E-12  0.333 333 333 33
0.333 333 333 333 333 3 E-13  0.333 333 333
0.333 333 333 333 333 3 E-14  0.333 333
0.333 333 333 333 333 3 E-15
0.333 333 333 333 333 3 E-
0.333 333 333 333 333
0.333 333 333 333 333
0.333 333 333 333
0.333 333 333 333
0.333 333 333 333 2 E-35
0.333 333 333 333 2 E-36
0.333 333 333 333 2 E-37
0.333 333 333 333 2 E-38
0
0
OK

```

Ici, nous sommes en accord avec la théorie. A tout moment, pour de petits angles, angles et SIN sont représentés par des nombres identiques. Les quantités ne sont assimilées à zéro que lorsque l'on atteint la limite inférieure de capacité (underflow), soit 10 puissance -38.

L'inconvénient? - assimiler à zéro un nombre inférieur à 10 puissance -10 dans un résultat final n'est pas une grande erreur. Mais que penser si un tel nombre se glisse dans un calcul itératif? - ou s'il prend place comme dénominateur d'une fraction avec arrêt brutal des calculs? (division par 0).

Le remède BASIC est simple:

```

100 IF X < 1E-05 THEN Y=X : GOTO 120
110 Y=SIN(X)
120 suite

```

Dans le programme Forth XX, on trouve un mot spécial FIU-OFF. Sur un "underflow", le microprocesseur produit une interruption vectorisée qui conduit à une routine réajustant le résultat du calcul (ce qui peut aboutir à un traitement spécifique, fonction du problème). Le mot FIU-OFF invalide l'interruption et le résultat est simplement assimilé à zéro. FIU-ON produit produit l'effet inverse.

Bien évidemment, il ne faudrait pas mettre en cause le langage BASIC lui-même. Seul l'algorithme interne utilisé pour évaluer SIN porte l'entière responsabilité de cette anomalie. D'ailleurs, certains BASIC ne commettent pas cette faute.

Virgule flottante et FORTH? - pourquoi pas?

** MASTER MIND **

Ce programme a été tiré d'un article paru dans FORTH DIMENSIONS, la revue de l'association US, Forth Interest Group. (VOL. III/5)

Il a été modifié et adapté à la console M19 de HEALTHKIT. Tout autre modèle peut être utilisé, dans la mesure où il possède l'adressage du curseur.

La fonction chrono (qui peut se supprimer) fait appel à une interruption qui se produit tous les 50èmes de seconde. Le nombre qui exprime ces 50èmes sert également à initialiser la racine de RANDOM (ce qui réalise un "RANDOMIZE" automatique).

Il faut noter que cette interruption est étroitement liée au microprocesseur utilisé (LSI 11/23 de DIGITAL EQUIPMENT).

J. THOMAS
F8RW - T466

REMARQUE

J'ai longuement insisté sur l'importance que j'attache à la standardisation, et au fait de ne publier que des programmes dont tous les "mots" soient connus des lecteurs. Mais je me rends bien compte que je n'ai pas le temps moi-même d'effectuer cette vérification. Je demande donc aux auteurs d'y apporter la plus grande attention, et aux lecteurs de me signaler **immédiatement** les mots qui leur posent problème. Si vous êtes assez rapides, cela me donnera le temps de demander des détails aux auteurs. Vous pouvez même envoyer un double de votre lettre à l'auteur. Mais envoyez-moi toujours la lettre, au cas où l'auteur ne répondrait pas assez vite. Faites-moi également des propositions quand au format des fiches pratiques de définition de mots. Bien sûr seuls les mots d'intérêt général sont concernés par ces fiches.

J-D Dodin

SCR # 228

```
0 ( Master Mind - Cursor tracking definitions )
1 : X YLOC @ ; : Y YLOC @ ;
2 : XBUMP X 52 = IF 23 DUP CUR.COL ! XLOC !
3 ELSE 1 XLOC +! X CUR.COL @ 8 + =
4 IF X CUR.COL ! ENDIF
5 THEN ;
6 : UNBUMPX X 23 = IF 52 XLOC ! ELSE -1 XLOC +! THEN ;
7 : TB. CUR.COL @ 47 = IF 23 CUR.COL !
8 ELSE 8 CUR.COL +!
9 THEN CUR.COL @ XLOC ! DROP Y X CURS. ;
10 : BS X.CUR.COL @ = IF DROP
11 UNBUMPX Y X CURS. SPACE Y X CURS.
12 DROP 32 COLORS X + 23 - C!
13 THEN ;
14
15 -->
```

SCR # 229

```
0 ( Master Mind - Character collection/editing guess )
1 : PROCESS DUP EMIT COLORS X + 23 - C! XBUMP Y X CURS. ;
2 : GET.CHAR KEY DUP 8 = IF BS ELSE DUP 9 =
3 IF TB. ELSE DUP 13 =
4 IF 1 DONE ! DROP ELSE PROCESS
5 THEN THEN THEN ;
6 : INITIAL 2 * 3 + DUP YLOC ! CUR.ROW ! 23 23 XLOC !
7 CUR.COL ! Y X CURS.
8 30 0 DO 32 I COLORS + C! LOOP ;
9 : GET.COLORS INITIAL 0 DONE ! BEGIN GET.CHAR DONE @ UNTIL ;
10 : PARSE.GUESS 4 0 DO I 8 * COLORS + C@
11 I 8 * COLORS 1+ + C@
12 I 8 * COLORS 2+ + C@
13 + + I GUESS + C!
14 LOOP ;
15 -->
```

SCR # 230

```
0 ( Master Mind - Clue generation )
1 : CLUE.CHECK 0 BLACKER ! 0 WHITER !
2 4 0 DO SECRET I + C@ GUESS I + C@ =
3 IF 1 BLACKER +! 0 I GUESS + C!
4 ENDIF
5 LOOP
6 4 0 DO GUESS I + C@ 0 >
7 IF 4 0 DO GUESS I + C@ SECRET J + C@ =
8 IF 1 WHITER +! 1 I GUESS + C!
9 LEAVE
10 ENDIF
11 LOOP
12 ENDIF
13 LOOP ;
14
15 -->
```

OK

SCR # 231

```

0 ( Master Mind - Present clues )
1 : GIVE.CLUES PARSE.GUESS CLUE.CHECK
2       Y 1 CURS. BLACKER @ . ." BLACK "
3       WHITER @ . ." WHITE " ;
4 : UNMASK 3 23 CURS. 4 0 DO I SCODE + C@ COLOR.FIND I 3 =
5       IF ." " ELSE ." " THEN
6       LOOP 23 1 CURS. ;
7 : ?AGAIN 20 58 CURS. ." TYPE MASTER TO"
8       21 58 CURS. ." PLAY AGAIN." UNMASK 23 1 CURS. ;
9 : LOSER 16 58 CURS. ." NICE TRY BUT"
10      17 58 CURS. ." NO CIGAR." 1 BEEP 1 TIME-INV ?AGAIN ;
11 : WINNER 16 58 CURS. ." PRECISELY:"
12      17 60 CURS. #ATTEMPTS ? ." TRIES."
13      1 BEEP 1 TIME-INV ?AGAIN ;
14
15

```

-->

SCR # 232

```

0 ( Master Mind - Directions to player )
1 : DIRECTIONS CLS CR CR CR CR CR CR
2       ." Welcome to Master Mind." CR CR
3 CR ." The object of Master Mind is to break the secret code."
4 CR ." The computer will pick the secret code, and you must"
5 CR ." figure it out. Two kinds of clues are given:" CR
6 CR ." (n) BLACK means that you have (n) pegs correct"
7 CR ." in both color and position." CR
8 CR ." (m) WHITE means that you have (m) pegs of the"
9 CR ." correct color that are incorrectly"
10 CR ." placed." CR
11 CR ." Be sure to spell the colors correctly. You may TAB among"
12 CR ." the 4 positions until you've make your best guess." CR
13 CR ." Type <RETURN> to receive clues."
14 CR ." Type any key to exit Directions." ;
15

```

-->

SCR # 233

```

0 ( Master Mind - Final )
1 : MASTER CLS 0 0 0 HMS 1 BEEP 20 TIME-INV 1 BEEP
2       1 15 CURS. ." DIRECTIONS WANTED? < Y or N > "
3       $INPUT 1 $LEFT " Y" $= IF DIRECTIONS KEY DROP ENDIF
4       $DROP CR CR 0 #ATTEMPTS !
5       ASK.FOR.LEVEL ASK.FOR.RANDOM
6       DISPLAY.BOARD TRANSLATE.CODE 0 0 0 HMS
7       0 10 DO 1 #ATTEMPTS +!
8       PARSE.GUESS I GET.COLORS GIVE.CLUES
9       1 BEEP 1 TIME-INV
10      BLACKER @ 4 = IF WINNER LEAVE
11      ELSE THEN
12      -1 +LOOP
13      BLACKER @ 4 < IF LOSER
14      ELSE THEN 23 60 CURS. ;
15

```

OK

SCR # 234

```
0 ( Master Mind - ADAPTATION )
1
2 # 226: supprimer la ligne 1
3 # 227: remplacer les caractères semi-graphiques par:
4 : BAR-1 33 EMIT ;
5 : BAR-2 33 EMIT ;
6 : LINE-1 61 EMIT ;
7 # 231: supprimer le chronométrage et la sonnette dans les
8 lignes 10 et 13:
9 : BEEP 1 TIME-INV
10 # 233: supprimer la ligne 9
11 : supprimer dans la ligne 1:
12 : 0 0 0 HMS 1 BEEP 20 TIME-INV 1 BEEP
13 : supprimer, éventuellement, CLS (Clear Screen)
14 : supprimer dans la ligne 6: 0 0 0 HMS (RAZ du chrono)
15
```

SCR # 235

```
0 ( Master Mind - ADAPTATION suite )
1 # 226: : CURS. HDCA ; ==> adressage du curseur en absolu
2 Pour la console H19 de HEATHKIT:
3 : HDCA 31 + SWAP 31 + 89 HESC EMIT EMIT ;
4 : HESC 27 EMIT ;
5 Si l'on veut ajouter BEEP:
6 : BEEP 10 * 0 DO 7 EMIT LOOP ;
7 ( n BEEP avec n allant de 1 à 100 - zéro exclu)
8 Pour H19: : CLS HOME HCD ;
9 : HOME 72 HESC ; : HCD 69 HESC ;
10 # 233: ajouter RANDOMIZE à la ligne 1 et définir:
11 1001 VARIABLE SEED
12 : RAND SEED @ 2725 U* 13947 S->D D+ DROP DUP SEED ! ;
13 : RANDOM RAND U* SWAP DROP ;
14 : RANDOMIZE CR ." PLEASE, DEPRESS ANY KEY: "
15 KEY DUP SEED @ * + SEED ! ;
```

SCR # 236

```
0 ( Master Mind - ADAPTATION suite et fin )
1 Si l'on ne possède pas les fonctions chaînes de
2 caractères, il faut supprimer en # 233:
3 $INPUT 1 $LEFT " Y" $=
4 $DROP
5 et remplacer par un mot unique: Y/N qu'il faudra définir:
6
7 : Y/N 1 KEY DUP EMIT 78 = IF 1- ENDIF ;
8
9 Ce mot place en pile un drapeau "faux" (=0) pour "N" entré
10 et un drapeau "vrai" (= ASCII du caractère entré) pour un appui
11 sur Y ou tout autre caractère.
12 En FORTH: faux = 0
13 vrai = toute valeur <> 0
14 donc, "vrai" ne correspond pas obligatoirement
15 à 1 (ce peut être une valeur ASCII ...)
OK
```

TESTAMENT ?

Au mois d'Avril dernier, Richard J. Nelson, Président de PPC-USA, donnait sa démission de son poste de Président et de celui d'éditeur des journaux du club. Les conditions de cette démission, survenue presque à la date du 10ème anniversaire de PPC-USA sont longtemps restées obscures. La situation est maintenant éclaircie.

Pour connaître de longtemps ce genre de situation, je n'ai pas de peine à en résumer le principal : Au fil des années Richard est venu à considérer PPC comme son enfant, ce qui n'est d'ailleurs pas faux. Mais les enfants grandissent. Richard n'a pas vu grandir le club et/ou n'a pas voulu partager la responsabilité. Il a dû accepter la constitution d'un conseil d'administration lorsque PPC est devenu une corporation officielle, mais a toujours gardé l'essentiel du travail -et donc du pouvoir- pour lui, au détriment de sa santé et de la santé de PPC. Le concept : PPC ne fait que ce que je suis en mesure de faire, et si je m'arrête, PPC s'arrête, ne pouvait être accepté par les autres membres du Conseil d'Administration. Quand, pris d'un moment de dépression, Richard a déposé sa démission, et demandé que PPC s'arrête (il a même envisagé la dissolution de PPC dans sa lettre de démission), les membres du bureau ont relevé le défi et fait le nécessaire pour que PPC continue.

Richard a alors manqué de la force d'esprit nécessaire pour voir grandir loin de lui son rejeton et s'est risqué à l'obstruction systématique. Je n'insisterai pas sur cet épisode peu glorieux, bien qu'excusable.

Je préfère garder en mémoire le Richard J. Nelson à qui nous devons le plus ancien et le plus grand club de micro-informatique du monde, avec ses 7000 adhérents, et, pour bien marquer la reconnaissance que nous lui devons et adoucir un peu son amertume, je vous propose d'envoyer, chacun de vous, une carte postale à "Richard J. Nelson, 2545 Camden Place, Santa Anna, California 92704 USA", disant à peu près ceci : "Hi, Richard, I won't ever forget you and the work you did for PPC !" et vous signez. Cela lui fera sûrement plaisir (attention à l'adresse, le 2545 est chez lui, le 2541 est le siège de PPC, mais il ne faut plus y envoyer de courrier, l'adresse actuelle de PPC est : PPC, POB 9799, Fountain Valley, California, 92728-9599 USA.

Ci-joint quelques commentaires sur l'affaire et le dernier article (publié récemment par PPC).

(traduction de la lettre : "Salut, Richard, je ne t'oublierai jamais, et je n'oublierai jamais ce que tu as fait pour PPC !")

Cher Jean-Daniel

A ce jour, tu as du recevoir le dernier numéro du PPC Journal et savoir que Richard Nelson n'est plus l'éditeur, ni le président, ni le responsable du chapitre de Santa Ana. La plupart d'entre nous sont déçus et en colère du fait de la décision du Conseil(1). J'ai beaucoup parlé avec Richard Nelson et Henry Horn a propos des évènements qui ont conduit a cette décision. J'ai aussi beaucoup parlé avec David White, qui a pris la place d'éditeur. En me basant sur ce que j'ai appris, je pense que le Conseil a agi de manière irresponsable et injuste. Je ne pense pas que PPC, tel que nous l'avons connu, survivra plus d'un an avec la nouvelle direction. Emmett Ingram nous a dit qu'il ne publierait rien de négatif ou de préjudiciable au Conseil. Il ne publiera même pas le dernier éditorial de Richard! Au moins Richard publiait les opinions des deux côtés lorsqu'il n'était pas d'accord.

Tu trouveras ci-joint une copie du dernier éditorial de Richard ainsi qu'une évaluation très objective du travail du Conseil depuis le dépôt des statuts de PPC, Inc. Ces points ont été discutés a notre dernière réunion de Chapitre le 14 juin. Je suis peu disposé a exprimer l'opinion de notre Chapitre en écrivant au Conseil. Nous avons toujours été et nous serons toujours des individus dans PPC. Nous disons ce que nous pensons librement et individuellement. C'est ce qui nous lie. C'est l'esprit que Richard a infusé dans le journal et dans PPC. Pour cette raison, la seule action que j'ai recommandée a ceux qui ont des idées précises sur l'action du Conseil est d'écrire au Conseil et d'exprimer individuellement leurs opinions. Si quelqu'un n'apprécie pas l'action du Conseil, il n'a pas a soutenir PPC au moment du renouvellement de sa cotisation. Notre Chapitre a toujours été ouvert a tous, gratuitement, membres de PPC ou non. Nous encourageons le libre échange des idées et du savoir et nous survivrons aussi longtemps qu'il y aura des individus qui voudront venir partager ce qu'ils ont avec les autres.

Bill KOLB (265)

(1) N.d.T. Comme vous le verrez plus loin, l'équivalent US de la loi 1901 n'évite rien d'autre qu'une Société de type commercial dont les statuts prévoient la clause de non-bénéfice. L'organisation de cette société est celle de toute autre société: Conseil d'administration (Board of directors) avec Président, Président Directeur Général (Chairman of the Board). (Les correspondances sont approximatives)

Une proposition d'article pour PPC Calculator Journal, 24 avril 1984.

UNE EXPERIENCE S'ACHEVE

Il y a dix ans, vers Mars 1984, J'ai eu affaire a un nouveau type de matériel Hewlett Packard. Pendant de nombreuses années, j'ai été habitué aux générateurs de signaux, oscilloscopes et instruments de test de HP. L'instrument mis sur le marché en 1974 était la Calculatrice Scientifique de Poche HP-65. En se référant aux définitions habituelles, la HP-65 était en fait un ordinateur, mais une bonne stratégie de marketing imposait que HP la baptise calculatrice.

La puissance énorme d'un système bien pensé, le stockage off-line des programmes et données, une vitesse raisonnable et des branchements sur test faisaient de la HP-65 un outil de calcul incroyable. Le prix de vente de 795 US\$ la mettait toutefois hors de portée du consommateur moyen. Soit on disposait d'assez d'argent, soit on mettait l'achat de la machine très haut sur sa liste de priorités. Ce qui surprit Hewlett Packard fut le nombre de personnes qui achetèrent la machine pour leur usage personnel. Il y avait peut-être une raison professionnelle pour l'achat, mais la vraie raison était le sentiment que la machine pouvait faire à peu près n'importe quoi si l'utilisateur était suffisamment astucieux ou créatif.

Il ne me fallut pas longtemps pour tomber amoureux de cette machine. Après quelques mois, je rencontrai d'autres utilisateurs et je commençai à réaliser que les utilisateurs de HP-65 avaient en commun des traits de personnalité dans lesquels je me reconnaissais. En Juin 1974, j'ai commencé à publier une lettre appelée 65 NOTES. J'ai financé les premiers numéros moi-même et je travaillais à cette lettre le soir. La réalisation de la lettre devint relativement facile après la mise en place d'un système basé sur une dactylo salariée et une photocopieuse XEROX 7000 du bureau. L'activité HP-65 était très forte et il apparut qu'une lettre mensuelle n'était pas l'idéal. Il était évident qu'il fallait la poster 'First Class' et comme cela était une activité de 'loisirs' il n'était pas question de salaires ou de bénéfices. Le prix demandé aux 'abonnés' était le minimum nécessaire pour couvrir les frais d'impression et de port. Toutefois, le problème vint du fait de la croissance de la taille de la lettre, chaque abonné étant un auteur potentiel d'articles.

Il devint immédiatement évident que ce qu'il fallait n'était pas une lettre, mais une organisation d'utilisateurs et le HP-65 Users Club naquit officiellement le 1^{er} Juin 1974. En plus de la lettre, un réseau d'utilisateurs personnels se développa. Il y eut des réunions au cours desquelles les utilisateurs les plus avancés partageaient leurs connaissances avec les autres. La chose la plus surprenante de l'affaire fut les personnes dans le coup. Je ne suis pas quelqu'un très 'social' (je suppose que la plupart des 'techniciens' sont dans ce cas), mais je me suis aperçu que tant que quelqu'un servait de point de ralliement, d'autres aideraient volontiers à remplir des enveloppes, écrire des articles et programmes, etc....

Le temps a passé, la HP-65 aussi. D'autres fabricants et groupes ont apparu et disparu, mais les utilisateurs HP ont une certaine loyauté et le Club devint une organisation internationale connue aujourd'hui sous le nom de Personal Programming Center, ou sous sa marque déposée PPC.

PPC a précédé l'industrie des ordinateurs personnels de six mois. Pendant les dix dernières années, j'ai vu une croissance formidable dans l'industrie et dans les nombreux clubs qui ont apparu pour supporter des machines particulières. PPC était toutefois fait d'un type d'utilisateurs différent et j'ai été continuellement émerveillé par le caractère de ces gens. J'ai voyagé dans le monde entier pour visiter des chapitres PPC et j'ai trouvé un noyau commun de traits de personnalités et de systèmes de valeur. Pendant des années, j'ai écrit à propos de ces gens et de leur engouement pour ce que j'appelle le véritable ordinateur personnel.

Au fur et à mesure que le temps et les années passaient, ma participation dépassa quelques heures le soir. Les heures passées à travailler avec les machines furent dépassées par les heures passées avec les gens, l'échange d'information, l'étude du marché, les contacts avec HP et la réalisation de la publication. Avec la croissance, les revenus augmentèrent, de même que les besoins en temps. Toutefois, la croissance était lente et continue et je pus, dans la plupart des cas, trouver de nouvelles méthodes pour faire face. C'était intéressant, mais cela devenait plus dur physiquement.

En moins de trois ans, j'arrivai à un style de vie structuré de façon à réserver toutes mes heures en dehors du travail à des activités liées à PPC, ceci parce que le défi était devenu un but: Est-ce que ces clubs d'utilisateurs d'ordinateurs personnels pouvaient devenir autonomes financièrement parlant? Je savais que cela était impossible d'un point de vue commercial, il suffisait de regarder le monde autour de PPC. Les publications apparaissaient et disparaissaient les unes après les autres, mais PPC continuait car il faisait tout lui-même. Il ne payait personne pour faire quelque chose. Je sentis que le problème était d'avoir plus de temps dans mes journées. Ceci devint très clair quand je m'aperçus que je m'endormais aux feux rouges en rentrant du travail.

À partir de là, j'adoptai une autre approche. Comme je suis un de ceux qui ne peuvent pas fonctionner continuellement avec seulement quelques heures de sommeil par nuit, je changeai de système. J'avais remarqué que mon cycle naturel de sommeil était d'environ 1 heure et demie. Quand j'arrivais à la maison après le travail, je mangeais et dormais aussitôt pendant 1 heure et demie. De cette façon, je pouvais me lever et travailler de 20h à 2h les jours de semaine et être au travail de 8h à 17h. Plusieurs fois, je préférais dormir plutôt que manger le midi et je me mettais dans ma voiture, dans une rue calme près de l'usine avec un réveil. Il n'y avait pas de problème à faire ceci car c'était un défi que de développer une organisation unique dans sa branche. Après trois ans à ce rythme, je sus que quelque chose de plus devait être fait.

Je décidai alors que j'avais besoin de plus de temps car 55 heures par semaine pour PPC ne suffisaient pas. Mon travail prenait 45 à 50 heures chaque semaine et je cherchais une manière d'utiliser ces heures. Naturellement, tout cela était motivé par un intérêt fantastique dans les gens et les techniques de ces petites machines si personnelles que nous supportions. Chaque année, les rentrées d'argent s'approchaient de ce qu'il fallait pour arriver à la stabilité financière. En 1979, j'ai commencé à réfléchir à des méthodes me permettant de travailler sur les projets PPC et être payé pour cela en même temps. Pourquoi ne pas gagner ma vie comme consultant en utilisant le savoir spécialisé accumulé en 5 ans d'expérience? Dans ce cas, je pourrais aider d'autres personnes à résoudre leurs problèmes et utiliser leur problèmes comme source d'articles le journal. Je quittai donc mon job à STATEK et devins consultant. Je continuai à consulter pour STATEK pendant un temps, mais seulement pour les dépanner et pouvoir faire la transition vers des jobs uniquement à 'double' but.

À cette époque, PPC avait son propre bureau et les coups de téléphone familiaux ne dérangeaient plus. Mes trois enfants étaient en 'High school' et avaient leurs propres activités. Je fis en sorte que le bureau soit à trois minutes de la maison et je plongeai dans un emploi du temps encore plus bousculé. Ma femme, Paz, était super efficace pour enregistrer les nouveaux membres, envoyer les vieux numéros, etc... Elle arriva à connaître de nombreux membres de PPC et savait toujours où j'étais et ce que je faisais. Elle supporta que je ne passe pas beaucoup de temps avec elle. Dans ces conditions, je pus être maître de mon temps et dormir suffisamment.

Depuis 1979, mon horaire est à peu près midi-4h du matin sept jours par semaine. La croissance de PPC et des projets comme PPC RDM demandaient beaucoup, mais, au moins, j'étais maître de mon temps. Tout mon matériel personnel, outils, équipement, livres et 23 ans de littérature et notes techniques furent amenés dans le bureau. Paz utilisait le garage comme bureau quand elle travaillait pour PPC le soir. Je m'arrangeais pour bloquer les activités d'impression le soir, à un moment où le temps de trajet passait de 30 à 10 minutes. J'étais disponible pour les membres du club 14 à 16 heures par jour, et grâce à une rotation de 4 heures de ma journée, j'avais généralement de minuit à 4h du matin pour me concentrer avec un minimum d'interruptions. Grâce à l'aide de ma femme et de ma famille, et en laissant tomber le reste, je sentis que mon but était plus proche, mais toujours insaisissable.

Quand le projet PPC ROM atteint les 250.000 \$, il devint évident que PPC n'était plus une opération menée par une personne, à temps perdu. HP avait des problèmes à traiter cette situation: combien de clients leur remettent des chèques individuels pour 10.000 \$ à la commande? Je me mis d'accord avec les gens concernés et deux d'entre nous cherchèrent des membres de PPC pour former un Conseil d'Administration. Après une année de discussions, un Conseil fut formé et nous déposâmes les statuts dans le style PPC: en faisant tout nous-mêmes. Ce dépôt de statuts fut rendu possible par le projet PPC ROM.

Peu de temps après le début de la vente de la deuxième série de PPC ROM, le bénéfice de ces ROM me permit d'avoir un salaire. PPC avait embauché Mary un an plus tôt et je devins l'employé n°2. Avec les salaires et autres dépenses, le coût opératoire de PPC approchait 18.000\$ par mois. Cela signifie que si PPC n'a pas au moins 216.000 \$ à la banque, il ne peut pas durer un an. Toutefois, nous étions éloignés de la stabilité financière autre que celle d'un magazine 'normal'. Cela était difficile, en particulier à cause des lois fédérales et locales sur les impôts des sociétés.

La société était une société sans profits (NdT: =association loi de 1901) sans exonération des impôts. Obtenir cette exonération était dur à cause des années précédentes où nous avions opéré sans statuts. Le projet ROM avait amené PPC du monde du club à celui des affaires. Le travail nécessaire pour faire tourner PPC augmenta énormément du fait de ces statuts. Nous n'avions pas ajouté de nouveaux projets pendant quelques années, c'était un effort continu sur tous les fronts. L'ajout de Computer Journal, la croissance lente, mais continue, et le nombre croissant de personnes dans le coup rendirent mon travail beaucoup plus complexe.

En 1983, j'ai commencé à penser que je ne pouvais plus être aussi efficace que les années précédentes. La qualité du Journal en souffrait évidemment. Dans le passé, je pouvais avoir dans ma poche de chemise l'ensemble des transactions financières de deux ans. Quand il fallait payer une facture, je plongeai la main dans ma poche, écrivais un chèque et reprenais mon travail. C'est la méthode la plus efficace pour travailler, sauf si plus de gens sont concernés. Les règles de l'art des affaires imposent d'avoir des livres comptables et si l'on veut bénéficier des réductions d'impôts et gagner des intérêts sur son argent, il faut suivre les règles de la comptabilité. Très peu de tout ce travail est directement lié à l'activité de l'organisation, mais après le dépôt des statuts, il fallait respecter les lois sur les impôts. Il était évident que cela m'intéressait moins que le reste et que c'était le travail des trésoriers de s'en occuper. Le Conseil de PPC s'occupa de ces problèmes opérationnels, et, petit à petit, je m'aperçus que je ne sentais plus battre le cœur de PPC. Ce n'était de la faute de personne, c'était simplement le manque de gens et de talents pour travailler de la façon voulue par le Conseil.

Le 10 Avril, je reçus une convocation pour une réunion du Conseil. J'étais fatigué et je savais que je ne pouvais plus me concentrer sur ce que je faisais. Depuis plusieurs mois, j'avais pris l'habitude de me réserver du temps le samedi et le dimanche pour voir des films et sortir de mon esprit un travail continuellement en expansion. Aucun problème n'était trop compliqué, c'était simplement trop pendant trop longtemps. Quand je lus l'ordre du jour de cette réunion du 21 Avril, je fus choqué par ce que je vis. J'eus l'impression que certains des sujets posaient la question même de ma présence. Je n'ai même pas fini la page. J'ai marché dans tous les sens pendant une heure en me demandant "Pourquoi fais-je cela?".

Du fait de la situation et de ce que je ressentais comme un gros manque de confiance de la part du Conseil, j'ai pensé qu'il était mieux de donner à ce dernier toute latitude pour résoudre les problèmes. J'ai donc envoyé un memorandum aux six autres directeurs leur expliquant la situation. Ce memo était accompagné d'une copie de ma démission en tant

que Président et Editeur. Mercredi après-midi, en arrivant au bureau, je demandai à Mary de bloquer le Journal, d'appeler Emmett, le "P.D.G." (Chairman of the Board) et je lui lus ma letter et mon memo. Je voulais du temps libre et pensais que le Conseil devait se réunir sans moi.

J'appelai Mary vendredi pour voir s'il y avait des problèmes. En même temps, j'eus une discussion avec Emmett qui m'apprit qu'une réunion de Conseil était prévue. Le Conseil se réunit le dimanche suivant, Emmett fut élu Président. Il avait déjà pris le poste de manager quatre jours avant. Ma demande de repos sans modification majeure de l'organisation était rejetée. Emmett, avec le support complet du Conseil, faisait marcher PPC. J'assistai à une partie de la réunion et renouvelai ma demande.

Les actions du Conseil étaient-elles celles que j'attendais? Je ne sais pas, je savais seulement que je ne pouvais pas continuer sans repos. J'étais complètement vidé de toute énergie intellectuelle. J'avais besoin de compréhension et j'étais jugé. La seule chose que le Conseil a pris en compte était que je n'allais pas leur montrer qu'ils ne pouvaient pas faire marcher PPC. Il est maintenant clair qu'ils ont acquis l'esprit de société, cet esprit qui permet d'imposer aux gens ce qu'ils doivent faire sans ces considérations "humaines" si chères à l'esprit de PPC. Des phrases comme Si Machin ne peut pas faire le travail, j'en trouverai un autre.... ne sont pas très bien reçues par des bénévoles.

C'est si triste pour moi, maintenant, de lire des mots qui font si mal. Ils n'avaient pas ce sens et mis à part une question à laquelle j'ai eu les réponses plus tard, ces mots ne concernaient pas les activités qui me concernaient tant. Je ne peux pas critiquer le Conseil, ils ne peuvent résoudre les problèmes qu'avec les moyens dont ils disposent. Ce qui était clair pour moi, c'était que j'avais travaillé dix ans pour quelque chose qui, apparemment, n'était pas leur idéal.

Est-ce que l'expérience est un échec? Si on définit un échec comme une non-réussite, alors, on conclura peut-être à l'échec. Il est certain que les objectifs idéaux pour lesquels je travaillais auraient de grandes difficultés à survivre dans des conditions commerciales normales. Je crois que j'ai failli parce que je n'ai pas expliqué dans le détail les idées, pratiques et procédures qui m'étaient si claires, de façon à ce que d'autres puissent comprendre la nature du système PPC. Je ne peux pas le faire ici par manque de temps et d'espace, mais je vais détailler certains des points principaux.

OUVERTURE. Le système d'échange d'informations doit permettre la libre expression des membres sur tous les sujets présentant un intérêt pour eux ou pour l'organisation. Dans les limites définies par les lois applicables, le bon goût dans le langage et les attaques contre un groupe défini, chaque membre devrait pouvoir demander que ses soumissions soient publiées. Je pense que pendant les dix ans où j'ai été responsable de ce processus, j'ai pratiqué l'ouverture vis à vis de tous les membres, en réunion ou par écrit. J'ai même publié des lettres contenant des attaques personnelles dans un esprit de compréhension. Les soumissions des membres ont toujours été publiées "telles quelles", c'était une règle.

AMATEUR. C'est un point que certains membres du Conseil ont du mal à comprendre. Un professionnel effectue un travail suivant un programme, d'une manière dite professionnelle pour un certain prix. Un amateur a le droit de faire des erreurs. Il a le droit de dépasser son calendrier. Une grosse différence entre les deux réside au niveau des attentes et de l'importance. Quelque chose de si personnel et si amusant ne devrait pas être si sérieux. Il est important de se souvenir aussi que l'évolution est à sens unique. Un amateur peut passer professionnel, mais il est rare que le professionnel retourne au statut d'amateur. Les performances d'un amateur peuvent atteindre et dépasser le niveau professionnel, l'inverse est inacceptable. La démarche professionnelle est plus rigide et inflexible. PPC, ce sont des gens, pas des produits informatiques.

L'ASPECT COMMERCIAL. Un groupe d'utilisateurs réussit s'il survit. Il n'a pas à rétribuer un investissement. Si chaque projet ou chaque activité de PPC était basée sur des pratiques commerciales, il y aurait peu de participation. C'est un autre point sur lequel le Conseil est en désaccord avec moi. Ils se sentent obligés de chiffrer en termes financiers chaque activité de PPC. Je pense qu'ils oublient les objectifs et les raisons d'être (de PPC) quand ils se mettent à penser en termes commerciaux. Quand les membres de PPC veulent que quelque chose soit fait, ils le font. Certaines règles et certains niveaux minimum ont été fixés afin de pouvoir exister, mais tout le reste se fait suivant les nécessités du moment. La continuation de PPC est le résultat de la volonté des membres, non le résultat d'une analyse commerciale.

PPC a toujours vécu au rythme émotionnel de ses membres, et, sur une base de un an, ses performances ont toujours été exceptionnelles.

Les membres doivent diriger PPC. leur travail, leurs idées, leur moral, haut ou bas, doivent décider le contenu, la raison et le délai d'exécution des projets lorsqu'ils sont définis, mis en chantier et terminés.

Ceci est ce en quoi j'ai toujours cru.

PERSONNEL. C'est un concept fondamental de la manière dont PPC a fonctionné. Si une erreur fondamentale a été faite dans les prévisions pour l'année et que les coffres sont vides, il est possible de demander de l'aide - et de l'obtenir - auprès des membres s'ils se sentent personnellement impliqués dans l'organisation. Pouvez-vous imaginer BYTE Magazine demandant à chacun de ses lecteurs un dollar car il est un peu à cours de liquidités? Le management doit avoir une foi absolue et un contact personnel avec les membres. Une attitude professionnelle n'aboutira pas à cela.

DECENTRALISATION. PPC entretient un réseau d'information au travers de ses chapitres, mais n'essaie pas de les contrôler. L'esprit de PPC, s'il existe, convient aux activités par chapitres. PPC fournit une ligne de conduite par l'exemple.

COMPATIBILITE COMMERCIALE. PPC doit s'efforcer d'être le sol sur lequel peuvent croître des produits productifs et différents. Le résultat passé, en ce qui concerne la création de nouvelles activités et le soutien est excellent. Si PPC n'existait pas, combien de millions de dollars de chiffre d'affaire pour HP et les vendeurs n'auraient pas existé? Il est possible d'être compatible avec le monde du commerce et des affaires sans en être partie prenante.

L'aspect le plus important de PPC est que c'est un système bien rodé, développé et inspiré par des centaines de personnes de bonne volonté dans le monde entier pendant dix ans. Que l'on change trop de composantes et le système sera dérégulé. Ceci ne veut pas dire que PPC ne pourra plus fonctionner et laisser son empreinte. Cela veut simplement dire que quand certaines conceptions de base changent, les méthodes de travail doivent aussi changer.

Et pour le futur? L'inertie que PPC a acquise le maintiendra pendant un moment. Je crois que maintenant qu'il a été décidé d'avoir une approche professionnelle, tout le monde sera content que les horaires et prévisions soient respectés. Il faudra quelqu'un qui de motivé, qui ait du temps et qui adore les produits que PPC soutient pour guider PPC à long terme. Il n'y a actuellement personne de tel au Conseil.

Et moi? J'ai très mal, je suis déçu. Ce qui me chagrine le plus, c'est de n'avoir pas pu faire passer chez d'autres une idée qui est si claire pour moi. Si je me suis trompé, il me semble que c'est parce que

j'ai supposé que les autres me comprenaient. Une idée insuffisamment expliquée est juste une ébauche de pensée. J'ai essayé. Après un peu de repos, qui sait ce qui va m'intéresser. Après tout, l'ordinateur personnel de mes rêves reste encore à construire. Je crois toujours qu'il le sera, un jour.

Merci à tous pour dix bonnes années.

Richard J. NELSON (1)
2545 W. Camden Place
Santa Ana, CA 92704 USA

Le 30 Mai 1984

De: Gary Bachlund

A: Membres du Conseil PPC

Emmett Ingram
John Kennedy
Fernando Lopez-Lopez
Doug Meacham
Bruce Murdock
Richard Nelson
Richard Schwartz

Sujet: Une opinion sur les affaires en cours

Chers amis, il y a eu récemment quelques événements qui ont projeté PPC, en tant que société, dans ce qui peut-être qualifié soit de petit problème, soit de crise majeure. Ayant agi en tant que "sage-femme" pendant la naissance et la première jeunesse de la société en tant que telle, je me sens prêt, comme ami de celle-ci et de vous tous à proposer quelques idées basées sur mes conversations récentes avec la plupart d'entre vous et sur ma position unique en tant qu'observateur extérieur pendant la conception de PPC du point de vue légal.

Premièrement, il y a l'évidence de la démission par écrit de Richard Nelson de ses responsabilités de Président et d'éditeur. Soumis à des tensions à la fois publiques et privées, il avait atteint les limites de ses capacités à remplir ces positions au moment de cette démission.

Au cours de discussions ultérieures, je l'ai trouvé plein de regrets et gêné par cette démission et les conflits avec d'autres membres du Conseil qui ont découlé d'un "empressement" apparent à accepter cette démission.

Oui, je sais que l'article 5-4 des statuts précise le processus à suivre pour accepter une démission, article étant éventuellement annulé par "des conditions opposées dans un contrat approuvé ou ratifié par le Conseil".

Y aurait-il des conditions conflictuelles à prendre en compte, peut-être sur une base historique ou éthique en ce qui concerne le travail de Nelson au club et à la société qui en a découlé? Vous me connaissez tous comme un médiateur et un réconciliateur. De mes conversations avec vous, évidemment différentes d'une personne à l'autre, j'ai retiré l'impression d'un accord général sur le fait que Nelson devrait rester avec PPC, bien que la mise en oeuvre de cette idée pose de gros problèmes.

À propos de ce premier sujet, sa démission et votre désir de le réincorporer dans la société comme employé, je ne suis pas sûr de la manière à employer, bien que je sois convaincu que chaque "joueur" devra faire des concessions pour arriver à un résultat.

Deuxièmement, il y a le fait évident que la tenue des documents comptables sous le contrôle de Nelson était loin d'être exemplaire. Il le reconnaît lui-même, et vos comptables vous demandent sûrement d'accomplir les actions nécessaires pour corriger cela, afin de tenir en main la fiscalité de PPC. Les livres ne sont pas encore apurés.

Après tout, le seul véritable but de la société était d'agir comme un abri public pour tous les membres, en leur permettant d'avoir une exemption d'impôts pour tout ce qui se rapportait au partage d'information représenté par PPC et le projet ROM. Réunir, préserver et distribuer au public les informations concernant l'histoire, l'architecture, la programmation et les utilisations des systèmes de calcul personnel programmables.

Et, troisièmement et à mon avis le plus important, avec Richard Nelson comme président et éditeur, et, je dois aussi le souligner, avec Emmett Ingram comme Chairman de tout le Conseil, PPC n'a pas commencé à remplir les papiers pour obtenir l'exemption d'impôts prévue dans les statuts, statuts qui portent toutes vos signatures sauf celles de Nelson, Noble et Lopez-Lopez.

Mon avis est que Nelson, en tant qu'officiel de PPC, Ingram, en tant que Chairman et visiblement "boss" de Nelson, et tous les autres membres du Conseil sont partiellement coupables de 41 mois de délais en n'ayant pas apuré les comptes de la société et en n'ayant pas rempli ces papiers. Je suis peut-être aussi responsable parce que je ne vous ai pas aidé à commencer ce processus plus efficacement qu'avec des conseils pendant la formation du Conseil.

Mais chaque membre du Conseil, Nelson, Ingram et les autres, était membre du comité de supervision chargé de contrôler le fonctionnement de PPC, et chacun, de par sa fonction de directeur est chargé d'effectuer les changements nécessaires pour faire progresser la société vers son objectif. Chacun de vous aurait dû être le catalyseur pour faire arriver PPC à l'exemption d'impôts prévue dans les statuts, Nelson, en tant que président aurait du. Naturellement, le Chairman, Ingram l'aurait aussi du. Chacun semble avoir espéré que quelqu'un d'autre aurait pris la tête, aurait fait le travail, aurait été l'homme-clé.

Au cours d'une conversation récente avec Emmett Ingram, j'ai senti renaître une volonté de remplir ces formalités complètement: papperasse et mise à jour des comptes. Est-ce qu'il fallait la démission de Nelson pour arriver à ce résultat? Était-il si fortement opposé à cette demande que les autres membres du Conseil ne pouvaient pas le faire? Emmett avait l'air de dire que, après le départ de Richard de la présidence, le travail de mise à jour des livres pouvait être fait. L'idée est que six hommes vont faire le travail que sept n'ont pas fait. Est-ce que Emmett Ingram était Chairman pendant 41 mois, oui ou non? Est-ce que lui et le Conseil ne sont pas autant responsables que Richard de la non demande d'exemption par PPC?

Je répète, et je m'excuse pour le manque de précautions, que le Conseil en entier semble coupable pour tous les problèmes que j'ai entendus rejeter sur Nelson. Et Nelson lui-même est aussi responsable, en tant que directeur.

En tant que corps gouvernant, le conseil aurait pu, à tout moment, améliorer la fiscalité pratiquée par Nelson, et il ne l'a pas fait, mis à part la rédaction de memos demandant cette amélioration. Le conseil, en permanence, avait le pouvoir de modifier certaines opérations au jour le jour que Nelson avait instituées, et il ne l'a pas fait. Mon impression est que le marécage dans lequel Nelson s'était embourbé avant sa démission était, partiellement, dû aux autres membres du conseil, aussi bien qu'à lui-même.

Quoiqu'il en soit, Nelson est présenté par certains membres du conseil comme le seul responsable de la position actuelle de PPC. Ceci est purement théorique, car chaque membre du conseil était (est) responsable des fonds remis à PPC par ses membres et ses clients "ROM" et est autorisé par les statuts à "superviser tous les officiels... de la société pour s'assurer que leurs fonctions sont correctement remplies."

A n'importe quel moment, le conseil aurait pu prendre des actions allant jusqu'au renvoi de Nelson de la présidence, "avec ou sans cause", pour citer les statuts, soit en se basant sur l'inefficacité de Nelson, soit sur son opposition aux désirs du conseil, et il ne l'a pas fait. Apparemment, le conseil, pour des raisons de son cru, a été extrêmement tolérant pendant 41 mois et semble maintenant fermement décidé à résoudre les problèmes que Nelson semble avoir créés. Toutefois, la trace des responsabilités ne s'arrête pas devant Nelson, mais passe devant chaque membre du conseil.

Chacun est responsable des problèmes et des réussites des 41 derniers mois, et le retour de Nelson dans la société est de bon sens. Vous avez tous été proposés au départ comme équipe pour propulser PPC vers l'avant. L'équipe est responsable de l'état actuel des choses. C'est la responsabilité de l'équipe tout entière de maintenir PPC comme organisme vivant et murissant.

Le faux problème de savoir qui est le chef ne fait que cacher les vrais problèmes et leurs solutions potentielles. Le faux problème du bouc émissaire ne fait que créer des vagues entre vieux amis. Dans mon esprit, soit tous les membres sont le chef, soit personne ne l'est. Dans mon esprit, les solutions demandent un petit effort, mais sont aisément réalisables.

D'abord, trouvez un moyen de garder l'équipe unie. La mégalomanie n'a rien à faire dans une organisation bénévole, d'intérêt public telle que l'incroyable PPC. Il n'y a aucun besoin de conflits de personnalités, de rivalités et d'affrontement aux réunions de conseil, de gardes du corps et de ces bruits de couloir qui ont caractérisé les dernières semaines. Si le conflit entre Emmett et Richard a une origine personnelle, je recommande de placer John Kennedy ou Bruce Murdock à la barre, au moins pour l'instant. Définissez plus précisément les fonctions et responsabilités de chaque directeur si ce directeur doit aussi agir en tant qu'officiel. Gardez cela simple et concis.

Deuxièmement, faites un effort pour faire aboutir cette demande d'exemption. Après tout, chacun de vous est responsable des fonds remis par les membres de PPC. Et, les statuts le demandent de chacun de vous, sans exception.

Troisièmement, trouvez des méthodes pour apurer la comptabilité et la garder en ordre. Ceci n'est pas du ressort des autres membres du Conseil, mais de chacun de vous. Essayez de faciliter ce travail. Si Nelson revient comme éditeur, rendez-lui la tâche facile pour ce travail. Aidez-le, même si cela veut dire travailler cote à cote pour y arriver. Faites faire un rapport de trésorerie, rapport très en retard d'après les statuts.

Restez dans les limites de ce que John Kennedy appelait "les disputes normales" et interdisez-vous les attaques personnelles. Un conseil évident.

La plupart des problèmes que chacun de vous m'a racontés n'a pas une, mais plusieurs solutions. Si, comme le dit Richard Schwartz, Nelson "s'opposait aux consignes du conseil pour la comptabilité", pourquoi ne pas travailler avec lui pour appliquer ces consignes. Ne considérez pas l'un d'entre vous comme subalterne dans son rôle d'officiel de la société. Je suis sûr qu'il était souvent plus facile de considérer Nelson comme "un pion faisant la volonté du conseil", plutôt que comme un collègue se débattant avec trop de travail.

Quelque soit le "chef" a la fin, dans un groupe d'intérêt public tel que PPC, il doit aussi être le principal "servant" du point de vue du temps et de l'effort, car le club ne peut pas s'offrir le luxe d'embaucher et de débaucher arbitrairement des talents tels que les vôtres. Ce n'est pas une question de nombre critique d'adhérents ou de financement pour aborder cette approche type "business". PPC a commencé grace au volontariat et ne vit que comme cela, que ce soit pout une "bouffe" dans le conté d'Orange ou pour un membre créant un article ou un programme a publier. La survie de PPC ne viendra pas d'un "autoritarisme", qu'il soit celui de Nelson, a sa façon, de Ingram, a la sienne, ou de "directives" du conseil, mais plutot de chacun de vous trouvant un moyen d'aller dans le même sens, en équipe. Soyez une équipe communicative. Résolez vos problèmes en équipe.

PPC est une magie. C'est un aiguillon pour l'industrie qui tend vers certains types de secrets et de satisfaction. C'est un réseau pour échanger des informations et des connaissances sur la programmation et l'utilisation de vos "jouets". C'est un rassemblement d'enthousiastes qui partagent pour le plaisir. C'est une société d'intérêt public, sans but lucratif pour abriter cette magie. Pour la garder et la nourrir. C'est Richard Nelson. Et Emmett, John, Bruce et les autres. Ce sont des chapitres et des membres indépendants. Et des taupes fouillant dans l'industrie pour voir le futur. PPC est vivant, en danger. L'ordonnance du docteur pour le tuer: des disputes. Une engendrera la suivante, tôt ou tard.

Pourquoi ne pas travailler a soigner? A rapprocher? A écouter plus qu'a ne décréter? A coopérer plutot qu'a diriger? C'est pour cela que j'ai recommandé plusieurs d'entre vous pour être directeurs de PPC. C'est ce que les Statuts impliquent. En fin de compte, c'est votre devoir. Pensez-y.

Meilleures salutations

copies: T. Tarasoff (HP N. Hollywood)
D. Terpak (HP Corvallis)

gary Bachlund

Traduit par
B. Soure
(TS02)

HP-41

PROGRAMMES

HP-41

ISOLA

Franck Lebastard
9672T35P43
10, rue de la Jalousie
35600 Redon

Redon, le 19 nov 1983

Isola.

Qui encore ne connaît pas le jeu isola ? Peu de monde, je pense. Quoiqu'il en soit, je vais pour ceux-là rappeler rapidement les règles de ce jeu passionnant que vous allez pouvoir essayer sur votre calculatrice (terme non néjoratif!!!) préférée grâce au programme que je vous propose.

Isola se dispute sur un terrain carré de huit cases de côté et oppose deux adversaires. Chacun d'entre eux dispose d'un pion qu'il met à son gré d'une case à chaque tour sur le jeu. De même, à chaque tour, après qu'il se soit déplacé, un joueur doit détruire la case de son choix, et ce, dans le but de bloquer l'adversaire. Le premier n'ayant plus de degré de liberté a perdu.

Prenons un exemple de partie, il sera plus parlant :

C'est au joueur 1 de jouer. A cet instant de la partie, il se trouve en B6 tandis que son adversaire occupe la case F4. Comme déplacement, il a plusieurs possibilités, quatre, et qui sont A6, A7, B5, C7. Il peut choisir C7 qui semble ici le meilleur coup. Il doit ensuite détruire la case vide de son choix. Les meilleurs tirs semblent être E3 et G5 car ils contribuent bien à l'isolement de 2. Mais bien entendu, rien ne vous empêche de détruire F3, c'est à vous de voir. Quand vous aurez joué, ce sera au joueur 2, et ainsi de suite.

	A	B	C	D	E	F	G	H	
1									
2									
3									
4						"2"			
5									
6		"1"							
7									
8									

Le programme que je vous propose occupe 1218 octets et 567 lignes, et nécessite un size 038 et un X-functions. Après avoir rentré le prgm en mémoire, faites r/s

- A la question "Alea ?" donnez un nombre, une graine, puis r/s
- Affichage successivement de la position du prgm sur le jeu puis de la votre.
- Question "DEBUTEZ O=1" à laquelle il faut répondre comme suit :

Si vous voulez commencer à jouer,
faites Enter puis r/s
Si vous laissez le prgm commencer,
faites simplement r/s

Dans ce second cas, au bout d'un certain temps, la machine BEEPE, puis s'éteint. Quand vous la rallumez, elle vous indique son déplacement puis son tir. Puis elle vous interroge sur votre coup comme dans le premier cas :

Si vous aviez opté pour la première possibilité en souhaitant commencer, après vous avoir indiqué le nombre de coup (déplacement) auquel vous avez le choix, elle vous pose les deux questions suivantes :

- "Votre Dep ?". Pour un même choix de déplacement (par exemple aller de B6 en C7), vous pouvez le rentrer de trois manières différentes :
soit B6-67
soit B687
ou encore tout simplement 67
puis r/s
- "Votre tir ?". Indiquez simplement la case que vous détruisez puis r/s
Le prgm fera le reste.

Bien entendu, si à l'une de vos réponses, vous essayiez de tricher, histoire de voir comment cette chère petite allait réagir par exemple, vous en seriez pour vos frais car cette chère petite refuserait tout simplement votre entrée et vous renverrait la question.

Vous allez voir que mon prgm n'est pas tron mauvais à ce jeu (...) mais qu'il présente un énoooooorme défaut, il est très lent. En début de partie (heureusement plus on avance plus il va vite), il peut mettre facilement une demi-heure à jouer son coup (...). Peut-être des courageux parmi vous en feront-ils une version optimisée quant à la vitesse; je les encourage alors à nous la livrer : un prgm rapide vaut toujours mieux que le même lent...

En espérant pour vous de folles parties,

Heureuse Programmation

Franck Lebastard

01*LBL *ISOLA*
ALEA ? PROMPT STO 00
ISOLA. FL: V. I AVIEM
FIX 0 CF 28 CF 29 38
PSIZE

12*LBL 23
RCL 00 CLRG STO 00
IS SF 25 PURFL
CF 25 15 CRFLAS
****** APPCHR
*g*****g 8

26*LBL 04
APPCHR DSE X GTO 04
****** APPCHR E
STO 05 CHS STO 01 E1
STO 08 CHS STO 04 +
STO 03 CHS STO 07 9
STO 02 CHS STO 06
SF 00

49*LBL 05
RCL 00 R-D FRC STO 00
00 * INT E1 +
STO 33 E3 / STO 09
SEEKPT GETREC E
ST+ 33 ATOX X?Y?
GTO 05 *JE SUIS EN *
FC? 00 *VS ETES EN *
XEQ 25 STO 32 FS? 00
STO 21 TONE 9 AVIEM
X FS? 00 *+* RCL 09
*SEEKPT E DELCHR RDN
INSCR FS?C 00 GTO 05
/ STO 09 X?Y
DEBUTEZ 0+ PROMPT
X=Y? GTO 22

97*LBL 17
SF 00 SF 11 55 FS? 55
XROM -IF*

103*LBL 06
/ SEEKPT *X* FC? 00
+ POSFL E3 * E
+ FS? 00 STO 32
FC? 00 STO 21 FS?C 00
GTO 06

120*LBL 24
1.008 STO 35

123*LBL 18
RCL IND 35 RCL 21 +
E - E3 / SEEKPT

GETREC ATOX E X?Y
GTO 02 1.008 STO 36

139*LBL 08
RCL IND 35 RCL IND 36
CHS X=Y? GTO 01 -
RCL 21 + E - E3 /
SEEKPT GETREC ATOX 3
X?Y? GTO 01 1.008
STO a -

160*LBL 09
RCL IND 36 RCL IND a
CHS X=Y? GTO 00 -
RCL IND 35 RCL 21 + +

E - E3 / SEEKPT GETREC ATOX E X<Y? GTO 00 RCL 35 E1 + E ST+ IND Y	298*LBL 00 RCL IND 19 RCL 21 + E - E3 / SEEKPT GETREC ATOX E X<Y? GTO 12 GTO 20	X<=0? AVIEW X<=0? STOP E X=Y? GTO 00
186*LBL 00 ISG a GTO 09	313*LBL 19 11.022011 REGSMAP FS?C 01 GTO 00 SF 01 GTO 24	441*LBL 15 FS? 03 GTO 26 "PARMI LES " ARCL 30 "1 COUPS JOUABL" AVIEW "ES, VOTRE DEP ?" GTO 16
189*LBL 01 ISG 36 GTO 08 GTO 00	320*LBL 00 RCL 30 X=0? GTO 01 1.008 STO a	450*LBL 00 RCL 34 STO 33 "VOUS NE POUVEZ" AVIEW "JOUER QUE " XEQ 25 AVIEW GTO 00
193*LBL 02 RCL 35 E1 + - E STO IND Y	326*LBL 13 RCL IND a RCL 32 + E - E3 / SEEKPT GETREC ATOX E X<Y? GTO 00 RCL a INT STO 30	459*LBL 16 ADM AVIEW STOP AOFF PRA "1" -3 AROT 7 ATOX 65 - X<0? GTO 15 X<Y? GTO 15 2 + STO 34 8 ANUM X<Y? GTO 15 E X<Y? GTO 15 X<Y E1 * ST+ 34 RCL 34
199*LBL 00 ISG 35 GTO 18 STO 19 STO 20 11.018 STO I	343*LBL 00 ISG a GTO 13	491*LBL 00 E - E3 / SEEKPT GETREC ATOX E X<Y? GTO 15 FS?C 03 GTO 01 RCL 34 RCL 32 - STO 09 11 X<Y? GTO 15 CHS X<Y? GTO 15 , STO 33 1.008 STO I
207*LBL 10 RCL 20 RCL IND I X<Y? GTO 00 STO 20 RCL I INT E1 - STO 19	346*LBL 01 RCL IND 30 RCL 32 + STO 33 E - E3 / SEEKPT E DELCHR RDM "0" INSCR BEEP OFF "JE SENS MA FIN " "1PROCHE..." FS? 03 AVIEW FS?C 03 PSE "MON DEPLAC. : " ARCL 37 TONE 9 AVIEW GETKEY 5 RCL 31 "JE SENS VOTRE F" "1IN PROCHE" TONE 9 X<Y? AVIEW X<Y? PSE "JE DETRUIS " SF 02	518*LBL 03 RCL 09 RCL IND I X<Y? GTO 00 E ST+ 33
218*LBL 00 , STO IND I ISG I GTO 10 FS? 01 GTO 19 5 RCL 20 X<=Y? SF 03 RCL 19 X=0? GTO 21	385*LBL 25 RCL 33 2 - RCL X E1 ST/ Z MOD 65 + XTOR RDM INT ARCL X RCL 33 FC?C 02 RTN AVIEW	525*LBL 00 ISG I GTO 03 RCL 33 X=0? GTO 15 RCL 34 E ST+ 10 - E3 / SEEKPT E DELCHR RDM "X" INSCR RCL 32 E - E3 / SEEKPT E DELCHR RDM "X" INSCR
232*LBL 20 RCL 21 STO 33 E - E3 / SEEKPT E DELCHR RDM "X" INSCR CLA XEQ 25 "1" RCL IND 19 INT RCL 21 + STO 33 XEQ 25 ASTO 37 E - E3 / SEEKPT E DELCHR RDM "X" INSCR GTO 19	403*LBL 22 , STO 30 STO 34 1.008 STO a	
266*LBL 12 E ST+ 19 RCL 19 8 X<Y? GTO 00 ST- 19	409*LBL 14 RCL IND a RCL 32 + RCL X E - E3 / SEEKPT GETREC ATOX E X<Y? GTO 00 ST+ 30 R1 STO 34	554*LBL 26 TONE 9 "VOTRE TIR ?" SF 03 GTO 16
274*LBL 00 RCL 19 RCL 09 X<Y? GTO 00 BEEP OFF "J'AI PERDU EN " ARCL 10 "1 COUPS..." AVIEW STOP	427*LBL 00 ISG a GTO 14 RCL 30 "J'AI GAGNE EN " ARCL 10 "1 COUPS"	559*LBL 01 RCL 34 E - E3 / SEEKPT E DELCHR RDM "0" INSCR GTO 17 .END.
286*LBL 21 RCL 00 R-D FRC STO 00 8 * E + INT STO 19 STO 09		

prgm "ISOLA"

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24

prgm "ISOLA"

"AJOSI" prgm

25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48

prgm "ISOLA"

"AJOSI" prgm

- 49
- 50
- 51
- 52
- 53
- 54
- 55
- 56
- 57
- 58
- 59
- 60
- 61
- 62
- 63
- 64
- 65
- 66
- 67
- 68
- 69
- 70
- 71
- 72

DECODE PRGM

prgm "ISOLA"

- 73
- 74
- 75
- 76
- 77
- 78
- 79
- 80
- 81
- 82
- 83
- 84
- 85
- 86
- 87
- 88
- 89
- 90
- 91
- 92
- 93
- 94

rs

The program has been decoded
by the author of the program
(1978) 2

DECODE PRGM

DECODE PRGM

Le PRGM "DCPGdV" (DéCode PrGm en dca. sur Vidéo) ci-joint donne les codes DEC de tout PRGM (319 REG au maximum ?). L'affichage des codes sur écran Vidéo se fait sous forme de tableau par 8 colonnes de 7 codes, chaque code ayant systématiquement 3 chiffres (complété éventuellement par 1 ou 2 zéros, en tête): 8 fois 3 = 24 + 7 SPACE = 31, ce qui occupe la quasi-totalité de la largeur de l'écran (32 colonnes) piloté par l'interface HP 82163. Signalons que ce PRGM peut être modifié facilement pour être utilisé avec l'interface 80 colonnes.

Ce PRGM met en évidence l'utilité d'une partie des fonctions de la ROM "PANAME" écrite par J.-J. DHENIN et St. BARIÉZEN, fonctions servant à piloter une interface Vidéo. La fonction 'XTOAL' incorporée dans la ROM précitée, est également utilisée ainsi que "DECODE" qui est issue de la ROM (EPROM) "ML1H" de St. BARIÉZEN mais qui peut être remplacée par "NH" de la PPC-ROM ou par tout PRGM semblable ("DECODE" se trouve également dans la "TOULROM"). La fonction "H-D" est également issue de la "ML1H" mais ceux qui n'ont pas la chance de disposer de cette fonction (ou d'une fonction semblable), pourront utiliser le PRGM "b-D" de Bruno TREDEZ (MR N1P31).

Pour rappeler en MEM centrale les codes d'un PRGM situé en IM, il suffit de faire 'croire' à notre machine qu'il s'agit d'un fichier de données. Il faut donc savoir où et comment est indiqué le type de fichier et le modifier en conséquence. Pour ce faire, la PS nous permet de placer le pointeur sur le REG adéquat (adresse absolue = DEC 190) dans lequel est introduit le code "qui-va-bien". Signalons au passage que ce REG 190 contient, lorsqu'il s'agit d'un fichier de données, un code de la forme :

```
20 BE 00 00 00 OR rr
```

où rr est en HEX le nbre de REG du fichier mais comme sur 2 DIGITS il n'est pas possible de dépasser 255, le multiple entier de 255 (nbre de REG) est indiqué en 'R' (cf lignes 20 à 25). Les autres DIGITS sont des constantes (en HEX).

Ceux qui disposent de la merveilleuse "ML1H" pourront remplacer les lignes 28 à 33 par -190 NRCL -495 NSTO RCL M -190 NSTO et ajouter à la fin du PRGM RTN LBL 03 BEEP -495 NRCL -190 NSTO (le END se trouve alors à la ligne 100). Le 'OFF' de la ligne 46 sera remplacé par GOTO 03. Dans ce dernier cas, le fichier sera remplacé dans son état initial. Si des débutant en PS son intéressés, je pourrais développer comment "travaille" ce PRGM.

J'indique aux collègues éventuellement intéressés, que j'ai également fait un PRGM "DGP/H" qui n'a que 47 lignes (il tient sur 1/2 carte) et qui permet d'imprimer (IMprimante non 'IL' suffisante) les codes HEX. Seul le module XF est nécessaire.....

N'en profitez pas pour trop DECODer !

hS (T178)

```

01*LBL "DCP 55 STO IND          PRP .. PRGM "DCPGdV
GdY" L
02 CF 27 56 DSE L               01*LBL "DCPG/H"          C6 00 F7 31 44 43 50
03 CF 15 57 GTO 09             CF 27 E PSIZE ASTO L      47 64 56 A9 18 A9 0F
04 CF 16 58 14                 FLsize STO 00 "PRGM" ..  A9 10 11 17 A6 5E 9C
05 17 59 ABS                   SF 12 ARCL L "I-" PRA   00 A9 1D A1 4F A1 54
06 PSIZE 60*LBL 08            CF 12 256 / " ***"     9A 74 A6 4F 83 F6 50
07 FIX 0 61 CLA                XTOA RCL 00 XTOA        52 47 4D 20 22 A0 0C
08 CF 29 62 RCL IND           RCL I "A"              98 74 F2 7F 22 A7 48
09 CLEAR L                     *iA                    A9 0C 12 15 16 43 F5
10 CSROFF 63 XTOA              RCL I                  20 BE 00 00 00 A6 6F
11 ASTO L 64 DSE L             X() c X()Y X() 00      75 A6 6F 90 75 F7 00
12 FLsize 65 RCL IND          X()Y STO c , SEEKPT   00 01 69 00 E0 00 90
13 ENTER↑ L                    75 CE 7D 71 CE 00 71
14 "PRGM" "" 66 XTOA           91 7D 1A 19 19 19 10
15 SF 12 67 H-D                30*LBL 01                14 3F 1A A6 6A 03 18
16 ARCL L 68 CLA               GETX DECODE 7          91 10 22 BF A8 19 A6
17 "I-" 69 ARCL X              34*LBL 00                57 AD 19 8D A1 0A 11
18 PRA 70 XEQ 13                ATOX XTOA RDM ATOX     14 61 0A A6 47 91 F4
19 CF 12 71 XEQ 13             XTOA RDM "I" DSE X     97 74 A6 47 91 F4 97
20 256 72 RCL 15               GTO 00 PRA DSE 00      74 BA 00 11 14 61 09
21 / 73 CSRHX                  GTO 01 END              87 90 F4 A6 6F 97 74
22 " ***" 74 PRA               GTO 01 END              90 F4 A6 6F A1 1F 87
23 XTOA 75 DSE L               UTILISATION:            98 73 E0 00 0D E0 00
24 RDM 76 GTO 08               PLACEK EN XM LE PRGM à  00 2F A1 52 A7 48 97
25 XTOA 77 -8                  DECODER ET SON NOM EN  74 09 00 1C 18 A1 57
26 RCL I 78 CSRvX              ALPHA ET XEQ.          96 0F 97 10 B2 00 18
27 "A" 79 ISG 15               A1 57 1A 19 19 19 10
* iA 80 DSE 16                 14 3F B3 00 0E A6 41
81 GTO 01                       13 8 85 14 18 A2 7C
82 8                               DCPGdV                  C0 1C 09 C7 (00 00 00)
28 RCL I 83 CSRvX
29 X<> c 84 ,99904
30 X<>Y 85 STO 15
31 X<> 00 86 GTO 02
32 X<>Y 87*LBL 13
33 STO c 88 ALENG
34 ,99904 89 3
35 STO 15 90 X=Y?
36 , 91 RTN
37 SEEKPT 92 48
38*LBL 02 93 XTOAL
39 8 94 END
40 STO 16
41*LBL 01 LBL"DCPG/H" CAT 1
42 ADV END 109 BYTES
43 SF 25 LBL"DCPGdV"
44 GETX END 199 BYTES
45 FC? 25 .END. 07 BYTES
46 OFF XROM "VK"
47 DECODE
48 14 USER KEYS:
49 ABS 14 "DCPGdV"
50*LBL 09
51 ATOX CODES DEC. :
52 STO IND
L L22: 245 032 190 000 000
53 DSE L 000
54 ATOX L27: 247 011 224 001 105
011 224 176

```

SAVEP
ENDIR

DCPGdV P029

.93

ATTENTION !
LE PRGM à DÉCODER DOIT
SE TROUVER EN TÊTE DE "M"

```

01*LBL "DC"
02 E
03 PSIZE
04 510
05 ENTER↑
06 "N REG"
07 PROMPT
08 -
09 E3
10 /
11 +
12 STO 00
13*LBL 00
14 RCL 00
15 CHS
16 NRCL
17 DECODE
18 PRA
19 DSE 00
20 GTO 00
21 END

```

UTILISATION :
PLACER LE PRGM à DECODER EN TÊTE DE MEN CENTRALE ET XEQ "DC" PUIS INTRODUIRE, à LA DEMANDE, LE NOMBRE DE REG OCCUPÉS PAR LE PRGM à DÉCODER.

prgm "DCPGdV"

1 [REDACTED]
2 [REDACTED]
3 [REDACTED]
4 [REDACTED]
5 [REDACTED]
6 [REDACTED]
7 [REDACTED]
8 [REDACTED]
9 [REDACTED]
10 [REDACTED]
11 [REDACTED]
12 [REDACTED]
13 [REDACTED]
14 [REDACTED]
15 [REDACTED]
16 [REDACTED] rs

prgm "DC"

1 [REDACTED]
2 [REDACTED]
3 [REDACTED]
4 [REDACTED] rs

prgm "DCPG/H"

1

2

3

4

5

6

7

8

9

~5

CONFIG

Le PRGM "CONFIG" permet d'obtenir la CONFIGURATION dans laquelle doivent se trouver les petits interrupteurs du lecteur d'EPROM de la SCIP pour l'adressage aux pages désirées des jeux d'EPROM 4, 8 ou 16k. Précisons que ce PRGM est principalement destiné aux débutants car il contient quelques "combines" qui me semblent appréciables et il n'est pas indispensable de posséder le lecteur d'EPROM précité pour utiliser et analyser ce programme.

Le PRGM initial est de Stéphane BARIZIEN qui, comme l'a écrit fort judicieusement notre Président, est "vite devenu un virtuose du microcode" (AU FOND 4ème édition P97). Je me suis contenté, suivant mes goûts et manies personnels, de modifier légèrement ce PRGM et, en particulier, d'y ajouter des "rambardees" interdisant d'introduire des données farfelues.

Analyse du PRGM

Le CF 27 du Pas 02 efface le mode USER et permet donc, si "CONFIG" est ASN à la touche de votre choix, d'éviter d'avoir des surprises désagréables lorsque vous avez de nombreuses ASN. Je conseille cette "précaution" en tête de tout PRGM.

Les Pas 04 & 05 baissent les DRAPEaux \emptyset à 7 (le module XFUNCTIONS est donc nécessaire mais ceux qui n'en disposent pas, pourront remplacer ces 2 Pas par :

,007 LBL 99 CF IND X ISG X GTO 99 ou :
7 LBL 99 CF IND X DSE X GTO 99 CF 00

L'adresse de la "1ère PAGE" demandée aux Pas 06 à 10 doit être indiquée en DEC et Les Pas 11 à 12 précédés obligatoirement de CF 22 (Pas 07), interdisent de rentrer autre chose qu'une valeur numérique (de l'ALPHA/HEX). Ne jamais oublier de placer CF 22 avant l'affichage (PROMPT, par exemple) car ce DRAPEau ne se baisse pas automatiquement, ce qui, à mon avis, est fort peu judicieux.

Les Pas 08 à 13-14 interdisent de "choisir" des Pages supérieures à 15. Notons que dans ce cas, la valeur 15 doit être introduite avant l'affichage.

Dans Les 2 cas de "barrage", Le retour au LBL 00 permet de redemander l'adresse de la 1ère Page.

Le GTO IND X du Pas 15 dirige vers un des LBL 04 à 15 ou à un des LBL 00 à 03 (Pas 94 à 97); dans ce dernier cas, Le TONE 8 du Pas 98 indique l'erreur et le retour au Pas 03 (qui se trouve à 4 Pas !) est effectué. J'espère que vous savez pourquoi Les Pages 8 à 3 ne peuvent pas être utilisées (?). L'adressage à la Page 4 m'a semblé utile pour ceux qui disposent d'une copie du module de service (sur EPROM).

Les Pas 16 à 48 lèvent les DRAPEaux correspondant à la Page choisie et permettent d'obtenir la valeur BIN sur 4 BITS (DRAP 00 à 03) écrits de Droite à Gauche. Cette disposition (ainsi que la suivante) a été imposée pour simplifier le câblage du Circuit Imprimé. Si vous désirez vérifier ces valeurs BIN, ajouter après le Pas 48 GTO 80 et à la fin du PRGM LBL 80 ,003 CLA LBL 81 FS? IND X '1' FC? IND X '10' AVIEW ISG X GTO 81 et essayez avec Les Pages de votre choix.....

Les Pas 49 à 51 demandent le nombre de kMOTs. Le AOFF a son utilité dans le cas où de l'ALPHA aurait été introduit, le GTO 28 du Pas 60 étant alors actif.

Le nbre de k introduit en 'x' est placé en ALPHA (Pas 52 à 55) puis remplacé en 'x' mais sous la forme alphanumérique de façon à permettre un branchement INDIRECT à un des LBL "4", "8" ou "16", Les LBL numériques correspondants étant déjà utilisés. Le SF 25 du Pas 57 est destiné à permettre de sauter le Pas suivant dans le cas d'une introduction différente de 4, 8 ou 16. En effet, si un LBL alphanumérique n'est pas trouvé, le fameux NONEXISTENT serait affiché et le PRGM serait STOPpé après un "tour" en l'absence de ce SF 25. Les Pas 57, 59 à 80 constituent donc une "rambarde" efficace contre les introductions incohérentes.

Les instructions des Pas 61 à 80 sont destinées à lever les DRAPEaux 4 à 7 correspondant au nbre de k choisi. Les Pas 65 à 68 et 72 à 75 sont destinés à interdire d'essayer d'introduire de la ROM au-dessus de la Page 15.

Les Pas 81 à 93 affichent la CONFIGURATION dans laquelle doivent être placés les interrupteurs prêtés. La boucle (Pas 83 à 92) ventile les résultats obtenus caractérisés par la situation (l'état) des DRAPEaux 8 à 7. Vous conviendrez que le procédé utilisé est fort astucieux (il est de Stéphane).

Il est certainement possible d'optimiser ce PRGM et j'invite les spécialistes à se pencher sur le problème, pas trop !

J'ai fait rapidement un PRGM "PRCONF" qui imprime la liste de toutes les CONFIGURATIONS possibles (listing ci-joint).

Indiquez-moi si je dois poursuivre dans cette voie.

MS (T178)

```

51 PROMPT          01+LBL "CON          4 K
52 CLA              FIG"                P4: *-----*--
53 FIX 0            02 CF 27            P5: *-----*--
54 CF 29            03+LBL 00          P6: *-----*--
55 ARCL X           04 .                P7: *-----*--
56 ASTO X           05 X<>F            P8: *-----*--
57 SF 25            06 "1eRe PA        P9: *-----*--
58 GTO IND          GE"                 P10: *-----*--
X                   07 CF 22            P11: *-----*--
59 TONE 6           08 15              P12: *-----*--
60 GTO 26           09 AOFF            P13: *-----*--
61+LBL "4"         10 PROMPT           P14: *-----*--
62 SF 07           11 FC?C 22         P15: *-----*--
63 GTO 25           12 GTO 00
64+LBL "16"        13 X>Y?                8 K
65 RDN              14 GTO 00          P4: *-----*--
66 12                15 GTO IND        P5: *-----*--
67 X<Y?             X                  P6: *-----*--
68 GTO 00           16+LBL 06          P7: *-----*--
69 SF 04            17 SF 01          P8: *-----*--
70 GTO 98           18 SF 02          P9: *-----*--
71+LBL "8"         19 GTO 26          P10: *-----*--
72 RDN              20+LBL 07          P11: *-----*--
73 15                21 SF 01          P12: *-----*--
74 X=Y?             22+LBL 05          P13: *-----*--
75 GTO 00           23 SF 00          P14: *-----*--
76 SF 07            24+LBL 04
77+LBL 98           25 SF 02                16 K
78 SF 05            26 GTO 26          P4: *-----*--
79 SF 06            27+LBL 11          P5: *-----*--
80+LBL 25           28 SF 01          P6: *-----*--
81 " "              29+LBL 09          P7: *-----*--
82 2.635074        30 SF 00          P8: *-----*--
1                   31 GTO 08          P9: *-----*--
83+LBL 24           32+LBL 14          P10: *-----*--
84 FS? IND          33 SF 01          P11: *-----*--
X                   34+LBL 12          P12: *-----*--
85 "1*"            35 SF 02
86 FC? IND          36 SF 03
X                   37 GTO 26
87 "1-"            38+LBL 10          2
88 FRC              39 SF 01
89 E1               40 GTO 08
90 *                41+LBL 15
91 X=0?             42 SF 01
92 GTO 24           43+LBL 13
93 PROMPT           44 SF 00
94+LBL 01           45 SF 02
95+LBL 02           46+LBL 08
96+LBL 03           47 SF 03
97+LBL 00           48+LBL 26
98 TONE 6           49 "4, 8 OU
99 GTO 00           16 K"
100 END             50 AOFF

```

```

← 1 CARTE
CAT 1
LBL'CONFIG
LBL'4
LBL'16
LBL'8
END 224 BYTES
LBL'CLXW
END 20 BYTES
.END. 03 BYTES
PRP "CLXW"
01+LBL "CLXW" 0
-64 NSTO ENDIR END

```

prgm "CONFIG"

1 [REDACTED]

2 [REDACTED]

3 [REDACTED]

4 [REDACTED]

5 [REDACTED]

6 [REDACTED]

7 [REDACTED]

8 [REDACTED]

9 [REDACTED]

10 [REDACTED]

11 [REDACTED]

12 [REDACTED]

13 [REDACTED]

14 [REDACTED]

15 [REDACTED]

16 [REDACTED]

17 [REDACTED]

18 [REDACTED] *rs*

```

      00 0T0 00
      01 00 00
      02 00 00
      03 00 00
      04 00 00
      05 00 00
      06 00 00
      07 00 00
      08 00 00
      09 00 00
      10 00 00
      11 00 00
      12 00 00
      13 00 00
      14 00 00
      15 00 00
      16 00 00
      17 00 00
      18 00 00
    
```

prgm "CLXM"

1 [REDACTED]

2 [REDACTED] *rs*

01*LBL "WFLAS"

- 02 SIZE?
- 03 FLsize
- 04 XYY?
- 05 Psize
- 06 STO 00
- 07 2
- 08 STO 01
- 09 ,
- 10 SEEKPTA
- 11 CLA
- 12 SF 25

- 13*LBL 00
- 14 SIZE?
- 15 RCL 01
- 16 E1
- 17 +
- 18 XYY?
- 19 Psize
- 20 ARCLREC
- 21 FC? 25
- 22 GTO 02
- 23 ALENG
- 24 6
- 25 /
- 26 INT
- 27 X=0?
- 28 "I="
- 29 X=0?
- 30 GTO 00

- 31*LBL 01
- 32 ASTO IND 01
- 33 ASHF
- 34 ISC 01
- 35 --
- 36 DSE X
- 37 GTO 01
- 38 FC? 17
- 39 "I="
- 40 GTO 00

- 41*LBL 02
- 42 E
- 43 CHS
- 44 AROT
- 45 ATOX
- 46 ASTO IND 01
- 47 CLA
- 48 RCL 01
- 49 E3
- 50 /
- 51 STO 01
- 52 TONE 9
- 53 XROM 30,00
- 54 RTN

- 55*LBL "RFLAS"
- 56 TONE 9
- 57 XROM 30,02
- 58 RCL 00
- 59 CRFLAS
- 60 E
- 61 ST+ 01

- 62*LBL 03
- 63 ISC 01
- 64 FS? 30
- 65 RTN
- 66 CLA
- 67 ARCL IND 01
- 68 FS? 11
- 69 APPREC
- 70 FC? 11
- 71 APPCHR

- 72*LBL 04
- 73 E
- 74 POSA
- 75 X=0?
- 76 GTO 03
- 77 RCLPT
- 78 INT
- 79 SEEKPT
- 80 --
- 81 POSFL
- 82 999

- 83 DELCHR
- 84 CLA
- 85 ARCL IND 01
- 86 Rt
- 87 E
- 88 +

- 89*LBL 05
- 90 ATOX
- 91 RDN
- 92 DSE X
- 93 GTO 05
- 94 ASTO IND 01
- 95 ALENG
- 96 X=0?
- 97 SF 11
- 98 APPREC
- 99 GTO 04

- 100*LBL "PRFLAS"
- 101 SF 21
- 102

- 103 SEEKPTA
- 104 SF 25
- 105*LBL 06
- 106 GETREC
- 107 PRA
- 108 CLA
- 109 FS? 25
- 110 GTO 06
- 111 .END.

VERITE

FLAS

Monsieur François LAUX (T 207)
 Chemin des Fontaines
 FEROLLES
 77580 CRECY la CHAPELLE

Voici trois routines oubliées par H.P. :

WFLAS Stock bout à bout dans les registres de données le fichier ASCII dont le nom est en Alpha. Chaque fin de "Record" est "balisée" par "x" (code 1 en décimal). A la fin de l'opération "WFLAS" vous demande la ou les cartes nécessaires à l'écriture de ce fichier. Vitesse: environ 4 caractères/sec.

RFLAS fait l'inverse. Mettez en Alpha le nom du fichier à lire et appelez RFLAS qui vous demandera d'introduire les cartes de données. Vitesse: environ 1,3 caractères/sec.

PRFLAS imprime un fichier ASCII dont le nom est en Alpha.

R 00 contient la taille enregistrée du fichier ASCII

R 01 pointeur du nombre de registres de données
 R 02 à Rxx fichier.

VERITE

Ce programme permet de dresser la table de vérité d'une fonction logique à 2 ou 3 variables, utilisant les opérateurs ET et OU (non limitatif), cette fonction étant entrée en clair en alpha.

Matériel nécessaire : 41C standard
+ X Fonctions

SIZE 023

CLOG: 24 octets

FLOG: 132 octets

J. P. Magnan (#359)

Ecole du NORD
Laboumboumas MAPOU
Mauritius

} fichiers "AS" 10 registres
(ascii)

Utilisation :

Soit à écrire la table de vérité de la fonction F :

$$F(A, B, C) = ((A \text{ ET } B) \text{ OU } (\text{NON } B \text{ ET } (A \text{ OU } C)))$$

Pour des raisons de rapidité d'exécution, une première partie du travail va être faite très simplement à la main.

- Les variables NON (V) vont être entrées comme étant indépendantes, c'est à dire sous un autre nom, selon la table suivante :

$$\begin{cases} (\text{NON } A) = D \\ (\text{NON } B) = E \\ (\text{NON } C) = F \end{cases}$$

- Les opérateurs ET et OU seront entrés comme suit :

$$\begin{cases} \text{ET} : * \\ \text{OU} : + \end{cases}$$

La fonction précédente sera donc entrée en alpha sous la forme suivante :

$$\langle \langle A * B \rangle + \langle E * \langle A + C \rangle \rangle \rangle$$

Remarques : le nombre de parenthèses "<" et ">" doit être égal.
• 3 variables ne doivent pas être au même niveau (A+B+C)

fichier ascii "AS" pour pouvoir retourner à notre aise le registre alpha sans perdre la fonction d'origine.

La boucle 10 recherche si la 1^{ère} parenthèse "L" trouvée correspond au début d'une parenthèse simple (de la forme $LVOV_L$). Sinon elle décale alpha d'1 caractère et retourne en 10.

Dans l'affirmatif elle commence le traitement de cette parenthèse simple

Chaque variable correspond à un registre de MEV :

A: 06	D: 09
B: 07	E: 10
C: 08	F: 11

A l'intérieur de ce registre se trouve le code de cette variable (0.255) sur huit bits. Les variables de A à F sont prédéfinies pour traiter tous les cas possibles à 3 variables :

A = 240 D = 15 (= 255 - A)

B = 204 E = 51

C = 170 F = 85

Le calcul de (V_1, OV_2) se fait à partir de la ligne 70 :

XEQ IND Z

en Z on a le code de l'opérateur

00: 43 = CH "+"
ET: 42 = CH "*"

en X et Y on a les 2 variables à traiter (leurs valeurs)

En retour on a une nouvelle valeur qu'on va affecter au premier registre libre (pointeur en 05)

La parenthèse simple $LVOV_L$ va être remplacée par cette nouvelle variable (la 1^{ère} disponible sera donc G)

Après une remise en ordre en alpha et restockage dans le fichier "AS" on retourne à la boucle 10 si le drapeau est brisé c'est à dire si il faut continuer l'interprétation.

Dans le cas contraire, le programme a terminé l'interprétation et se branche au programme "RES" pour publication des résultats.

Seule la longueur d'alpha limite la longueur de la chaîne (et bien entendu la MEV mais je doute qu'on utilise 320 variables logiques en une seule fois). On pourrait traiter directement en X MEM mais cela suppose un + grand nombre d'opérations (c'est-à-dire, en microcode ...)

Mais :

Possibilités supplémentaires pour l'entrée de la fonction en alpha et évaluer des expressions trop longues :

1. L'écriture $((A+B) \times (A+B))$ est équivalente à $((A+B) \times G)$
En effet, dans le traitement, la machine commence par évaluer $(A+B)$ (1^{ère} parenthèse simple). Elle va lui attribuer la variable G (1^{ère} disponible après F). On aura donc bien $((A+B) \times G) = ((A+B) \times (A+B))$
On peut ainsi entrer des expressions très complexes et plus longues que 24 caractères.

2. Il est possible de scinder une expression en deux.
Chaque variable calculée est sauvee à la fin du traitement dans la variable =, qui n'est pas effacée.

Il est donc possible d'évaluer une première partie puis la fonction complète en remplaçant la 1^{ère} partie par =

ex : Soit $F = (((A+B) \times (C+D)) + (D+E)) \times (C+(D+F))$

$$\begin{aligned} \text{on va évaluer } F_1 &= (((A+B) \times (C+D)) + (D+E)) \\ &= ((G \times (C+D)) + (D+E)) \\ &= ((G \times H) + (D+E)) \\ &= (I + (D+E)) = (I+J) = K \end{aligned}$$

K va être sauvee dans = Il suffit donc de faire maintenant :

$$\text{NEP "BOL"} \quad " (= \times (C+(D+F))) \quad "$$

$$F_2 = (= \times (C+G))$$

$$F_2 = (= \times H)$$

$$F_2 = I$$

La complexité atteinte peut être très grande : Calculer :

$$F_1 = ((A+B) \times ((A \times D) \times F))$$

$$\text{puis } F_2 = ((= \times (=)) \times (G+G))$$

revient à calculer : $F_2 = \underbrace{(((A+B) \times ((A \times D) \times F)) \times ((A+B) \times ((A \times D) \times F))) \times (Y+Y)}$

FLOG: catalogue de Fonctions LOGiques.

Ce catalogue ne comprend actuellement que 2 fonctions ET et OU. Mais on peut très facilement l'augmenter. C'est pourquoi j'ai décrit son fonctionnement.

Il comporte quelques techniques que je crois originales. Ma 1^{ère} version était plus classique et beaucoup plus lente. Elle se rapprochait de celle publiée dans PPCJ V10N1 (JAN/FEB 83)

Sous-Programme "XY": Les 2 nombres, en X et Y en entrée sont placés dans le registre d en position 0-7 et 8-15 respectivement. Les octets de poids fort étant en 0 et 8.
Les drapeaux 7, 8 et 16 sont levés en entrée pour éviter le problème des octets nuls

Il ne reste plus ensuite qu'à tester ces drapeaux avec les fonctions habituelles FS? FS?C pour exécuter les fonctions désirées. Le résultat est mis en sortie dans les drapeaux [0-7] de d et la routine $\Phi 2$ permet de mettre l'octet correspondant en X

DB: C'est un programme de conversion décimal - binaire pour visualiser les résultats sous leur forme habituelle (0-1)

Mémoires utilisées

$\Phi 5$: pointeur de variable	$\Phi 0$:
$\Phi 6$: $A = 240$	$\Phi 1$:
$\Phi 7$: $B = 204$	$\Phi 2$: variable =
$\Phi 8$: $C = 170$	$\Phi 3$:
$\Phi 9$: $\bar{A} = D = 15$	$\Phi 4$: pointeur de début de α
$\Phi 10$: $\bar{B} = E = 51$	
$\Phi 11$: $\bar{C} = F = 100$	

Flags utilisés:

- [0-15], utilisés par FLOG (peuvent être utilisés en dehors)
- 20 : indicateur de fin de traitement
- 25 : erreur

01 LBL^T FLOG

LBL^T ET

XEQ^T XY

7 E-3

LBL 00

STO Y

FS? C IND X

XEQ 01

ISG X

010 GTO 00

GTO 02

LBL^T OU

XEQ^T X1

7 E-3

LBL 03

STO Y

FC? IND X

XEQ 01

ISG X

020 GTO 03

LBL 02

RCL d

STO M

ATOX

RTN

LBL 01

8

+

FS? IND X

030 SF IND Y

RDN

RTN

LBL^T XY

SF 07

SF 08

SF 1G

RCL d

STO M

RDN

040 XTOA

RDN

XTOA

CLX

STO N

-2

AROT

RCL M

STO d

RTN

050 LBL^T DB

FIX 0

T.

LBL 05

STO Y

2

MOD

ARCL X

RDN

LASTX

060 /

INT

X ≠ 0?

GTO 05

END

LBL T CLOG
 22 PSIZE
 11 STO 05
 240 STO 06
 204 STO 07
 170 STO 08
 15 STO 09
 51 STO 10
 85 STO 11
 SF 25
 E1 "AS"
 CLFL
 FC? 25 CRFLAS
 CF 20
 0 SEEKPTA STO 04
 ADN TONE 7
 T FONCTION ?
 PROMPT AOFF
 AVIEW APPREC

LBL 10
 ALENG
 5 x=y? SF 20
 62 PCSA
 67 PCSA
 STC Z -
 4 x=y? GTC 01
 RCL Z 1 +
 ST + 04 AROT
 GTO 10

LBL 01
 60 POSA
 ST + 04 AROT
 ATOX ATOX ATOX

X <> Y ATOX
 DELREC APPREC
 XEQ IND Z
 0 SEEKPT
 GETREC
 ATOX 1 ST + 05
 RCL T STO IND 05
 RCL 05 59 + XTOA
 RCL 04
 1 + CHS
 AROT AVIEW
 DELREC APPREC
 0 STO 04
 FC? 20 GTO 10
~~0~~ RCL IND 05
 STO 02
 BEEP STOP
 GTO T RES

LBL 09
 59 - X <> Y
 LASTX -
 RCL IND X
 RCL IND Z RTN

LBL 43
 XEQ 09
 XEQ T 0U RTN

LBL 42
 XEQ 03
 XEQ T ET RTN

LBL T RES
 RCL 06 XEQ 00

RCL 07 XEQ 00
 RCL 08 XEQ 00
 RCL IND 05

LBL 00
 XEQ T DB
 TONE 8
 PROMPT
 ENC

Phase I partie II

La première partie a été soumise dans PPC Journal de mai 1987. La PPC-44 a reçu un bon accueil de la part de nos lecteurs. Certains commentent de leur temps, du matériel et même de l'argent pour que le projet se réalise. Les ordinateurs ne sont pas si compliqués et si coûteux qu'il y a quelques années. Les ordinateurs des entreprises y ont contribué de leur côté. Les entreprises qui réussissent ont souvent des programmes généraux.

Voici quelques recommandations.

A. Ne faites pas trop de recommandations. Les recommandations doivent être utiles et réalisables. Les recommandations doivent être utiles et réalisables. Les recommandations doivent être utiles et réalisables.

B. Les idées financières et techniques pour les programmes doivent être utiles et réalisables.

APPLICATIONS

financier du projet.

Les applications sont les suivantes :

Phase	Description	nb.
1.	Établissement et publication des listes de noms des utilisateurs.	Quatre à cinq
2.	Algorithme des listes et publication des algorithmes existants de référence.	Quatre à cinq
3.	Commentaire des algorithmes existants et leur utilisation.	Quatre à cinq
4.	Évaluation des besoins de matériel pour les listes de noms des utilisateurs.	Quatre à cinq
5.	Évaluation du coût de l'investissement matériel.	Quatre à cinq
6.	Recherche des logiciels, l'achat des nouvelles sociétés.	Départ de l'investissement.
7.	Plan de travail technique et matériel.	24 mois
8.	Conduite des travaux de l'investissement.	36 mois

PPC-44

Phase 1 partie II

La première partie a été publiée dans PPC Journal de nov. 1983. La PPC-44 a reçu un bon accueil de la part de nos membres. Certains proposent de leur temps, du matériel et même de l'argent pour que le projet se réalise. Les ordinateurs aujourd'hui sont si compliqués et si sophistiqués qu'il est nécessaire que les utilisateurs y mettent du leur pour faire une machine qui répondra aux besoins des prochaines générations.

Voici quelques propositions.

A. HP étudie actuellement une troisième génération de calculateurs. La société américaine s'engage fortement dans le marché du calculateur et il y a de l'espoir pour qu'apparaisse chez eux, pour la première fois depuis plus de deux ans, une nouvelle génération de machine.

B. Des aides financières et logicielles pour une première étude de la machine sont possibles grâce aux ressources des membres PPC.

C. D'éventuels capitaux pour une autre société importante sont possibles si différents membres consentent à nous aider en établissant les contacts nécessaires, etc.

D. Le plus grand problème est la conduite et l'engagement financier du projet.

Les différentes phases du projet PPC-44

Phase	Description	nb.
1.	Rassemblement et publication des idées de tous les utilisateurs	Ouvert à tous
2.	Affiner les idées et publier les spécifications externes de référence.	Ouverts à tous les participants
3.	Commencer les développements logiciels pour faire une maquette de démonstration.	Ouvert à tous les participants
4.	Faire une étude de marché pour vérifier le projet et son intérêt.	Ouvert à tous
5.	Etablir un plan de travail, former une équipe.	Deviens privé
6.	Chercher les capitaux, fonder une nouvelle société.	Départ de l'horloge.
7.	Finir le design logiciel et matériel.	24 mois
8.	Conduire une revue de l'utilisateur.	3 mois

9. Débuter la construction (production) 6 mois
10. Distribution du produit. Prix spéciaux pr PPC

Certains membres trouveront difficile de participer au projet, qui deviendra éventuellement privé mais qui, s'il est couronné de succès, sera profitable à ceux qui auront "tenté le coup".

Si un membre était contre cette manière de voir, il ou elle devrait garder toute idée et projet secrets, et ne pas se laisser entraîner. Tout le projet est basé sur l'idée que toute nouvelle idée est nécessaire pour faire une machine réellement universelle, pratique, bref, un vrai ordinateur personnel. Ceux qui travaillent sur le projet méritent toutes les récompenses qu'ils obtiennent. Nous voulons tous une super machine, et nous avons tous nos propres idées. Si elles ne sont jamais activement exprimées, elles ne seront probablement jamais utilisées. Abordez une société avec une nouvelle idée de produit et vous trouverez un mur légal. Vous perdrez un temps fou à obtenir d'eux ce que vous voulez, avec toutes les procédures légales que les grosses sociétés doivent suivre. Ce sera donc difficile. Si vous voulez une machine qui répond à vos besoins, cette manière de voir apparaît être comme l'une de celles qui sont possibles. Je peux vous assurer que vous fournirez d'excellents renseignements à HP ou quelques autres fabricants si vos idées sont publiées en un article. Si vous explicitiez clairement vos idées, vous connaîtrez le moment où elles seront utilisées. Vous devez donner vos idées librement si vous en attendez des progrès.

Il n'y a aucune garantie que nous puissions amener ce projet à terme, jusqu'à la phase 10, mais je suis confiant quant aux phases 1 à 4. La phase 1 est toute bien, et il y a beaucoup de gens qui sont intéressés pour aller jusqu'au bout.

Je ne peux pas offrir mieux que le meilleur appui possible de la part de PPC. Je peux essayer d'apporter ma contribution à chaque phase, mais je dois considérer PPC comme un groupe d'utilisateurs totalement autonome et indépendant. Puisque nous progressons beaucoup en ce moment, je pense que nous devons considérer un certain nombre d'objectifs majeurs à réaliser avant de penser à quelque chose d'autre. Ceci est la volonté des dirigeants PPC et des membres qui supportent PPC. Je ne peux être qu'un conseiller volontaire dans ce projet. Et j'aiderai ainsi toutes les phases si c'est nécessaire.

La plus importante chose à avoir bien à l'esprit est qu'il faut conserver à chaque phase son objectif, et chacune doit être exécuter pas à pas. Nous devons absolument terminer la phase 2, ou tout ce travail ne sera qu'un simple exercice. Enfin, tout document publié pourra être utilisé par les participants et sera l'expression de leurs efforts.

Richard Nelson (1)

(traduction T35)

CASSETTES

CASSETTES HP: LE CATALOGUE

Toute cassette doit être initialisée pour y définir un catalogue et le format d'enregistrement. Cette opération est réalisée par la fonction programmable **NEW** du module **HP-IL**.

Le catalogue consiste en un nombre entier d'enregistrements (calculé par la HP suivant l'exemple p 32 du manuel HP-IL). Chaque enregistrement contient 8 fichiers-catalogue. Le dernier fichier-catalogue est réservé au système, de même que les deux premiers enregistrements de la cassette.

La position des enregistrements sur la bande sera notée:
REC Numéro de piste/Numéro d'enregistrement.
(0 ou 1) (0 à 255)

La position des octets sur la bande sera écrite sous la forme:
Numéro de piste/Numéro d'enregistrement/Numéro d'octet.
(0 ou 1) (0 à 255) (0 à 255)

Ex: REC 0/0 désigne l'ensemble des 256 octets stockés sur la piste 0 dans l'enregistrement 0; 0/0/19 est le 20ème octet de l'enregistrement 0 sur la piste 0 (voir manuel lecteur de cassettes, description technique). Toutes les valeurs sont écrites en base 10 sauf indications contraires. J'appelle "fichier-catalogue" ce qui contient le nom du fichier, sa position, etc et "fichier", les informations proprement dites: le programme, les assignations, etc...

Les REC 0/0 et 0/1: Bien que ne faisant pas partie du catalogue, ils sont souvent utilisés pendant les opérations d'écriture. Ils ont la structure suivantes:

REC 0/0 & REC 0/1							
POSITION	OCTETS						
0/0/0	: 128	0	0	0	2	10	0
0/0/7	: 128	0	0	0	2	10	0
0/0/14	: 0	0	0	0	0	5	0
0/0/21	: 0	0	0	0	0	0	0
0/0/28	: 0	0	0	0	0	0	0
etc	: "	"	"	"	"	"	"
0/1/249	: 0	0	0	0	0	0	0

L'octet 0/0/19 (encadré) est le nombre de REC réservés au catalogue. Ici le catalogue peut contenir 39 fichiers-catalogue parce qu'il occupe 5 REC. Quant aux 511 autres octets, je ne saurais vous dire à quoi ils servent (peut-être réglage de la sensibilité lecture... en définissant ainsi les bits nuls, tout autre signal au niveau des têtes de lecture sera pris comme un bit à 1...?).

Le catalogue: Il commence au REC 0/2.
Les 8 fichiers-catalogue par REC sont stockés les uns après les autres, ils contiennent chacun 32 octets (voir figure):

Les octets 0 à 6 contiennent le nom du fichier.

Les octets 7 à 9 ont tous pour valeur 32. Ils ne permettent pas d'identifier un fichier avec plus de 7 caractères avec la HP 41 et le module HP-IL.

Les octets 10 et 11 contiennent le type de fichier:

(*) Tous les fichiers suivants ainsi que celui-ci sont ignorés.

(**) Après PURGE, les octets 10 et 11 sont mis à zéro. (je détaillerai les fichiers annulés dans un prochain journal).

(***) Les informations contenues dans le fichier ne sont pas accessibles par le module HP-IL. Le module I/O ou un équivalent microcodes est nécessaire pour pouvoir lire ou enregistrer un tel fichier.

Les octets 12 et 13 sont des nuls (utilité ?).

Les octets 14 et 15 contiennent la position du fichier sur la bande:

Octet 14: octet de piste. En fait, il faut passer en Hexadécimal pour se rendre compte que le 1er digit n'est pas utilisé; ainsi 240 et 00 (FD et OD) désignent tous les deux la piste 0. Un numéro de piste supérieur à 1 déclenche une erreur de support (MEDM ERR) si on appelle le fichier (lecture).

Octet 15: numéro de l'enregistrement où commence le fichier (programme, ...)

Les octets 16 et 17 sont des nuls (utilité ?).

Les octets 18 et 19 donnent la longueur du fichier en REC. Utilisé pour calculer l'adresse (REC X/Y) d'un nouveau fichier. Longueur du fichier en REC = (octet 18) x 256 + octet 19.

Les octets 20 à 25 sont des nuls (utilité ?)

Les octets 26 et 27 contiennent respectivement 128 et 1 (pourquoi ?).

Les octets 28 et 29 donnent la longueur du fichier en octets. Utilisé pour savoir combien d'octets il faut transférer dans le fichier. Longueur du fichier en octets = (octet 28) x 256 + octet 29.

OCTETS 10 ET 11	TYPE DE FICHIER
224 208	: DA Données
224 64	: WA Sauvegarde
224 80	: KE Assignations
224 96	: ST Etat
224 128	: PR Programme
0 1	: AS ASCII (X mem.)
255 0 à 255	: Fin de catalogue (*)
0 0	: Fichier annulé (**)
Autres valeurs	: ?? Fichier de type inconnu (***)

L'octet 30 contient les particularités du fichier (S,P,A)

OCTET 30	PARTICULARITES
0	: Fichier ordinaire
1	: Fichier privé (P)
2	: Exécution automatique (A)
8	: Fichier protéger (S)
Autre	: Combinaison des particularités (on peut faire X<F et tester les drapeaux 0,1et3).

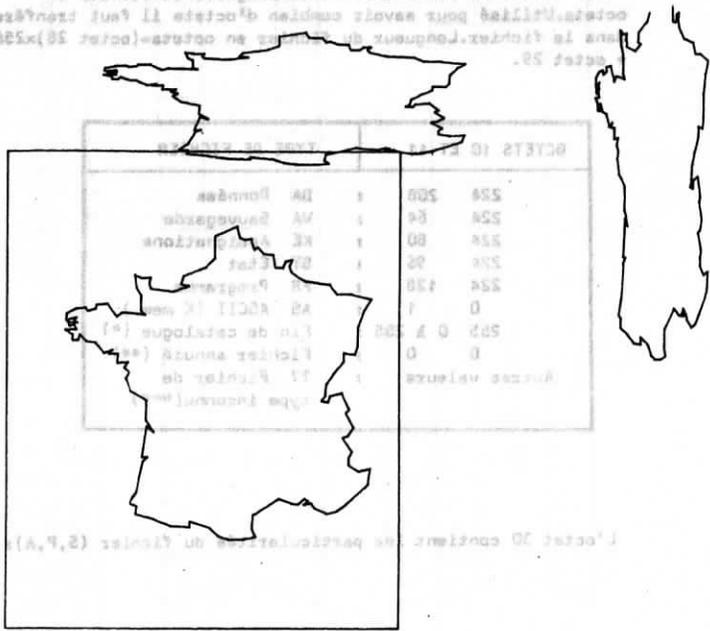
L'octet 31 a pour valeur 32 (pourquoi? séparateur de fichier?).

J'ai pu modifier tous les octets qui ne paraissaient pas avoir d'utilité (?) sans apparemment provoquer une erreur ou perturber le fonctionnement de la HP. Il y aurait en tout 16 octets qui ne serviraient à rien (ce qui pourrait permettre de porter à 16 fichiers le nombre de fichiers immédiatement accessibles au lieu de 8). A quoi servent les 2 premiers REC (mis à part l'octet 0/0/19)?

Je pense qu'un décodage du module HP-IL pourrait nous fixer les idées. En attendant, il faudrait voir du côté du HP 75 (avec le module "I/O utilities") si le catalogue à la même structure (10 caractères d'identification? mémoire tampon 1 de 8 fichiers?...)

* par fichier-catalogue

Pierre T. 384 P61



LE XF MIS A NU

Vous savez probablement tous maintenant ce que l'en-tête de chaque fichier contient (cf PPC-T N°6). Les informations présentes dans l'en-tête d'un fichier fournissent au HP-41 les renseignements utiles concernant le fichier. Toutefois, il restait à trouver comment le HP-41 pouvait reconnaître le fichier travail, et aussi comment était assurée la continuité entre les différentes zones de mémoire étendue!

Le premier point à connaître est que le X-FONCTIONS occupe les adresses 191 à 64 (inclus); un module X-MEMORY va occuper les adresses 1007 à 769 (inclus) s'il se trouve dans un port de numéro pair (2 ou 4) et les adresses 751 à 513 (inclus) s'il se trouve dans un port de numéro impair (1 ou 3). Vous comprenez donc pourquoi on ne doit pas mettre deux X-MEMORY dans des ports de même parité...

Si on fait un petit calcul, on découvre que le XF couvre une zone de 128 regs de mémoire vive, et chaque XM une zone de 239 regs. Or, ces modules ne fournissent (respectivement!) que 127 et 238 regs utilisables par l'utilisateur! De là à dire que belle mère HP a prévu un registre "d'état" à chacun de ces modules, il n'y a qu'un pas! En effet, les regs 64 (040), 513 (201) et 769 (301) ne sont pas comme les autres! Il est toutefois difficile de les décrypter, car lorsqu'on les rappelle (après avoir "tripatouillé" le rideau!) on obtient 0 en X, et on perd la mémoire étendue...

Méthode de décryptage:

La solution consiste à créer une adresse "artificielle". Cette adresse est utilisée conjointement avec le registre d'état b. Ceci permet de visualiser les registres sous forme de lignes de programmes! Le problème est qu'on ne peut, avec cette méthode, visualiser les Nuls. On risque aussi de quitter le bon registre sans s'en apercevoir! L'astuce consiste à placer 00 00 00 00 00 00 F7 dans le registre situé juste au dessus de celui que l'on veut décrypter: on pourra alors le visualiser sous la forme d'une ligne de texte: le tour est joué!

Registre d'adresse absolue 64(040):

Ce registre est de la forme:

HEX: 0 0 0 N N 0 0 0 J J J R R R

NN est le numéro du fichier travail (1er, 2em...). Comme une file occupe au minimum 3 regs, ce nombre est compris entre 0 (pas de fichier) et 200 (Hex 0C8). Deux digits (1/2 octets) suffisent donc pour le codage. Il semble que NN code le numéro de fichier lors d'un EMDIR, ce qui expliquerait pourquoi cette fonction transforme en fichier travail la file sur laquelle elle s'interrompt!

RRR contient à ma connaissance toujours 0EF, qui est l'adresse absolue du 1er reg d'en-tête du premier fichier (1er reg de la zone d'extension mémoire)

JJJ dépend de la configuration du système. Si le XF n'est pas accompagné de X-MEMORY, ces trois digits sont à 0. Dans le cas contraire, JJJ=2EF ou 3EF selon que la jonction se fait avec le registre d'adresse 2EF ou 3EF (ça dépend de la configuration du système).

Registres d'adresse absolue 769 et 513:

Le contenu dépend de la configuration du système! Supposons que la continuité entre les XM et XF se fasse dans l'ordre suivant: on passe de 040 à 3EF, puis de 301 à 2EF. Le registre 301(769) sera de la forme:

HEX: 0 0 0 0 0 0 4 0 J J J R R R

On aura R R R = 3 E F ici, et J J J = 2 E F (si le second XM n'est pas connecté on a J J J = 000)

Si le second XM est connecté, il occupera les adresses 751 à 513 et le registre 201(513) sera de la forme:

HEX: 0 0 0 0 0 D D D 0 0 0 R R R

Avec R R R = 2 E F et D D D = 301!

Les rôles des registres 769 et 513 peuvent être inversés si la continuité ne se fait pas dans cet ordre.

Si un seul XM est connecté, il en sera de même sauf que le registre 513 ou 769 sera:

HEX: 0 0 0 0 0 0 4 0 0 0 0 3 E F
ou 2 E F

CONCLUSION:

Nous savons désormais comment la HP traite les fichiers en mémoire étendue. La connaissance du contenu des registres "système" n'apportera pas grand chose en PS, dans la mesure ou le rappel est impossible(?). Les possibilités sont toutefois énormes en Microcode, et maintenant le listing MC du XF devrait être plus facile à analyser!

Je précise que les digits présentés à 0 dans cet article servent peut être à quelque chose, mais tous mes efforts pour le prouver ont été vains... Il est aussi à noter que si on enlève un module, la HP ne corrige pas immédiatement les contenus de ces registres systèmes.

J'espère que d'autres essaieront de continuer d'analyser le fonctionnement du XF, car bien des choses restent à découvrir...

PS: Le digit du reg système du XM qui fait la continuité avec le XF et qui est à 4 sur les exemples semble toujours conserver cette valeur(?)

J. BOSCHAT (T71+P51).



VARIABLES

POUGEON Olivier
T62P64

6 Bd Chanard
56170 QUIBERON Quiberon, Le 25/8/83

Dans la série : "Mais que peut on bien faire avec le
X-Fonctions ?" voici : LES VARIABLES SYMBOLIQUES.

D'accord , mais une variable symbolique c'est quoi ?
Pour les familiers des langages dits évolués (Basic , Fortran,
Pascal et autres Cobol , vous pouvez passer ce paragraphe.
Pour être schématique , disons que c'est une variable à laquelle
on accède par un nom au lieu d'y accéder par une adresse (c'est
comme cela que notre machine fonctionne). Ce que je vous
propose est donc d'appeler vos variables par un nom : c'est
plus clair !

À ma connaissance , aucun micro ordinateur de poche ne
peut disposer de noms de plus de 1 caractère c'est peu clair
et pas vraiment plus parlant que le système de la 41. Mais ,
se désespérer à ce sujet , c'est oublier que nous avons une
machine exceptionnelle , qui dispose de fichiers en X-Memory.
Nous allons donc les utiliser pour simuler ces variables symbo-
liques ; évidemment , ce que je propose prendra plus de place
en mémoire qu'un simple rappel de registres. Mais il est plus
pratique , à mon avis , d'avoir les constantes que vous utilisez
fréquemment (constante de Planck par exemple) en X-M sous
le nom "PLANCK" que d'être obligé de savoir qu'elle réside dans
le registre 00. En plus cette dernière situation force souvent
à modifier les programmes (géniaux , mais longs de préférence)
de vos camarades qui n'utilisent pas les mêmes constantes que
vous. Evidemment , il y a la solution de la carte de données
mais , que se passe t'il quand vous l'avez oubliée ?
Ma solution , la voici , quelques programmes qui vous permettent
de créer une variable , sous forme d'un fichier d'un registre.
D'en rappeler et d'y stocker une valeur et ceci en ne modifiant
que le registre L , presque avec la même facilité que par STO
ou RCL .

Les noms peuvent avoir jusqu'à 7 caractères (limite imposée
par H.P.) mais après tout c'est ce que permettaient les premiers
Fortrans donc , ne nous plaignons pas. !

CRV permet la création d'une variable d'un registre

```
01 LBL CRV
02 ABS
  R↓
  I
  CRFLD
  R↓
  LAST X
08 RTN
```

Le ABS du pas 02 n'est la que pour sauvegarder la valeur de X
dans L .

RT et ST permettent de (respectivement) rappeler et stocker
une valeur comme un STO ou un RCL , mis à part que SV change L.

```

01 LBL RV
O
SSEKPTA
R↓
GETX
06 RTN

01 LBL SV
X() L
CLX
SSEKPTA
X() L
SAVEX
07 RTN

```

```

TE 381 20
10 Q82
XVA2
RTR
10 381 20
W 11X
W 11X
W 11X
W 11X
1 072
1 R
WOW
W 11X 21
W 11X 11
W 11X
W 11X
1 11X
1 072

```

Le second intérêt du X-F à propos des variables symboliques est de permettre la création de tableaux à 2 dimensions. En fait on ne peut que simuler ces tableaux : les routines suivantes traitent un ensemble de fichiers de noms de la forme "NOM" numéro de colonne comme un tableau à 2 dimensions ; ainsi le tableau "TAB" de 10 X 10 registres sera représenté par les fichiers de données "TAB1" à "TAB10" comportant chacun 10 registres. CRT permet la création de tableaux de variables de N X M éléments . Pour cela mettre le nom du tableau en ALPHA N en X et M en Y , Puis XEQ CRT . Pour la destruction d'un tableau il suffit de mettre une valeur inférieure ou égale à 0 dans Y.

```

01 LBL CRT
E 3
/
1
+
FIX 0
ASTO Z
X() Y
LBL OO
CLA
ARCL Z
ARCL Y
X) O?
CRFLD
X) O?
PURFL
ISG Y
GTO OO
FIX 4
20 RTN

```

ST et RT font la même chose que SV et RV mais dans un tableau. La syntaxe est la suivante : nom du tableau auquel est collé le numéro de colonne que vous désirez traiter vous laissez ensuite 6 blancs (TRES IMPORTANT) puis vous mettez le numéro de ligne puis XEQ RT ou XEQ ST. Ici aussi le fonctionnement s'apparente à STO ou RCL avec simplement une perte de L.

```

01 LBL RT
XEQ O1
GETX
RTN

```

```

05 LBL ST
   XEQ 01
   SAVEX
   RTN
09 LBL 01
   X() N
   X() M
   X() N
   STO L
   R↓
   ANUM

```

```
16 X() N
```

```

17 X() M
   X() N
   SEEKPTA
   X() L
   STO L

```

```
22 RTN
```

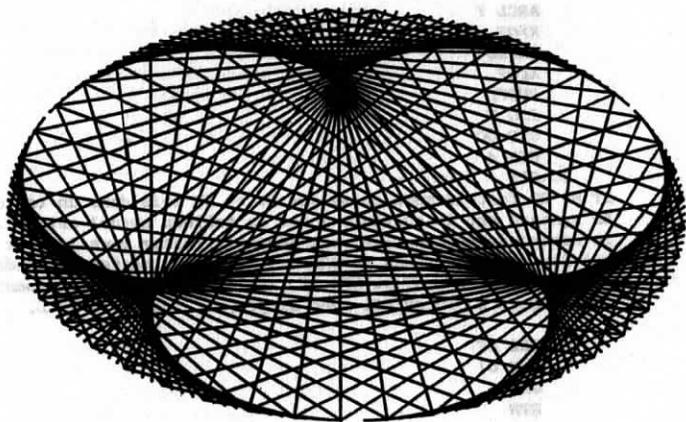
Le but des séquences X() N , X() M , X() N est d'intervertir les contenus des registres N et M pour permettre de récupérer le numéro de ligne (d'ou la nécessité des 6 blancs) , la deuxième séquence de ce type restaurant le registre ALPHA dans son état initial.

Voilà , l'avantage de ces différents programmes est qu'ils sont transparents , ils ne modifient que le registre L. L'intérêt des variables symboliques pour la HP est avant tout la clarté qu'elles apportent , ainsi que la structure de tableau. Dernier mais pas le moindre vient l'avantage de la X-M sur la mémoire normale dans le stockage des NNN: en effet d'après les essais que j'ai fait , les NNN ne sont pas normalisés quand ils sont stockés ou rappelés de X-M. On peut donc y stocker les NNN avec un nom qui correspond à leur origine (par exemple "RID" pour l'adresse du rideau avnt que vous n'y touchiez).

J'espère que vous trouverez d'autres applications à ce type de variables.

Heureuse Programmation à tous

O. Pougeon T62P64



DEMONSTRATION du TRACEUR HP 7470 A

W"

Monsieur Reigner Cyril(T282)
4,rue de Morlaix
44800 Saint-Herblain
Saint-Herblain,le 22 Août 1983.

Cher Jean Daniel,

Je t'envoie un article sur la fonction W",
decidement elle a fait et fera encore(j'espere),
couler beaucoup d'encre.

J'ai pris ma HP-41CV, qui a un TONE Z très
court donc une ROM 1 de type F(la dernière),et
j'ai un module X-FUNCTIONS dans le port 1 et le
PPC-ROM dans le port 2;l'ordre est important car
je n'ai pas eut les mêmes resultats dans l'ordre
inverse!

Faire memory lost et la sequence suivante:
31 ENTER! CHS XEQ"1K" puis passer en mode US-
ER (le pointeur est toujours en ROM) et faire
SHIFT SHIFT vous avez au milieu de l'afficheur
2___.Demande un nombre,pas d'alpha possible
mais un mode indirect par la pile ou un regi-
stre est possible,mais je ne sais pas si ce-
la change réellement quelquechose?

Tapez un nombre quelconque(ex 44).Afficha-
ge: 2 44 passe à l'extrême droite de l'affi-
cheur puis la 41 semble prendre un sommeil
profond....La touche ON reste inoperante?? il
faut donc effectuer un RESET(fleche arriere et
deux ON-cf AU FOND p 58-).

ENTREZ	VOIR
- "PRGM"	mode PRGM:.END.REG 45 nous sommes de retour en RAM???
-SHIFT SHIFT	02 2
- 0 0	01 W" (5 secondes, un V et une cloche)
-PRGM	mode RUN:0,0000
-SST	0,0000 passe à droite???
SST (2x)	GTO IND 00 puis après quelques secondes ALPHA DATA vous avez de nouveau la commande du cla- vier.

J'ai verifié la pile:vide,les registres(0,3
XEQ"BV")et voici leur contenu:

RO7:31,0000 postfixe
RO8:-31,0000 code de touche
RO9:176,2210 index pour le stockage
R10: R0 pour abaisser le rideau
R11: premier assignement

Ce sont les registres utilisés par 1K.
Au message nonexistent reprendre par XEQ"GE"

ENTREZ	VOIR
-CLΣ SST	GTO IND 00 puis 1,0000 en X
-SST (2x)	GTO IND 00 puis 2,0000 en X
-MEAN	0,5000 normal si on
-SDEV	0,7071 ($\frac{1}{\sqrt{2}}$) considère av- oir fait Σ+ avec 0 et 1!

Contenu des registres:

RO7,RO8,RO9 et R10:inchangés
R11:1,0000 (Σx) R14:0,0000 (Σy^2)
R12:1,0000 (Σx^2) R15:0,0000 (Σxy)
R13:0,0000 (Σy) R16:2,0000 (n)

On en deduit donc en toute logique que son action est $\Sigma+$ lorsqu'elle est programmée.

Faites SST deux fois et pendant l'affichage de GTO IND 00 appuyez sur ON jusqu'à voir $\Sigma-$, si vous relâchez avant le NULL la fonction sera bien exécutée. Si vous appuyez sur SHIFT au lieu de ON la 41 affiche l'indicateur SHIFT et attend la prochaine pression de touche, c'est à dire qu'elle exécute la fonction normalement shiftée du clavier en mode USER.

Mettez la HP en mode ALPHA puis SSTez deux fois et vous verrez après le GTO IND 00 habituel Σ (ou A et append)!

Plus curieux encore faites une boucle sur cette instruction en mode ALPHA (12 boucles sont nécessaires, en faire plus n'ajoute rien??pourquoi??) et vous pouvez voir GTO IND 0A, faites alors ALPHA deux fois et la 41 prendra ses vessies pour des lanternes car le "registre" alpha contient maintenant: AAAAAAAAAA. Amusant oui mais utile ???

Restez en mode ALPHA et SSTez deux fois puis pendant l'affichage de GTO IND 00 appuyez sur ON et vous ne serez pas surpris de voir Σ ...

Nota: si pendant un GTO IND 00 en mode RUN vous appuyez sur USER ou ALPHA la machine agit comme pour le cas de SHIFT tandis que sur PRGM provoque l'arrêt de la 41 en mode PRGM.

Je pense qu'il doit y avoir une relation avec les X-FUNCTIONS (RAM??) car refaite un memory lost et l'assignement sans le module X-F puis un double SHIFT suivi par 44 (comme la première fois) et vous aurez la joie de voir votre machine plantée pour combien de temps, mystère?? (je n'ai pas eu la patience d'attendre plus de dix minutes). J'ai personnellement fait un RESET puis éteint ma machine avant de remettre le X-F dans le port 1, vous pouvez alors ressaisir la fonction W la 41 ne vous en tiendra pas rigueur.

Echangez maintenant le PPC-ROM et le X-F (qui passe donc en port 2), rallumez exécutez W et la 41 est de nouveau bloquée mais pour un peu moins d'une minute et elle réaffiche le registre X.

Allons plus loin (âmes sensibles s'abstenir):

ENTREZ VOIR

GTO "MK", OFF, permutés (encore) les deux modules

ON
SHIFT (2x) 2
0 0 2 00 puis elle s'éteint
RESET? PRGM 01 +
avancez dans
le programme
par des SST

02 RCL 00
03 LBL 02
04 LBL 00

l'affichage s'éteint progressivement comme pour XROM 00,00 puis ap-

ciais??(#####)

J'ai effacé par ←

04 LBL 00
 05 STO 12
 06 LBL 03
 07 DEG
 08 LBL 07
 09 LBL 14
 10 XROM 00,0#1
 11 FRC
 12 LBL 13
 13 ""(FO)



là j'ai fait PRGM BST SST ATOX PRGM SST
 14 XEQ 00
 15 LN
 16 LBL 11

MOD 17 DEG (80)

18 E#x-1 X(58) exactement la fonction

19#R-P O(4F) appelée par XEQ-encore

20 CHS T(54) le registre Q-

21 - A(41)

*Si la Sequence
 Plus Haut n'est
 Pas exécutée.*

22 MOD LBL 04

LBL 01

X=0?

LBL 01

RCL 01

RCL 10

RCL 10

""(FO)

W"

LBL 00

FRC

LBL 00

XEQ N

RCL 02

O

LBL 01

ST-16

LBL 00

LBL 02

O

LBL 00

.END. REG 45

OFF et permutez les deux modules

ON,PRGM,BST + ""(AA Nuls)

""(FO)

.END. REG 45 c'est fini...

Qui pourrait fournir des explications à ces résultats?

Pourquoi W"" agit comme Σ+ et la touche devient l'équivalent de la touche 11?

Pourquoi l'action de W"" change avec l'emplacement du X-FUNCTIONS? Quel lien y a t'il?

Pourquoi a t'on ce programme (registres d'états)?

Qu'en dit le microcode et les résultats sont-ils identiques sur d'autres HP?

Je resumerais ces questions en deux mots:
Help Please

Reijer

XROM

T300

J.C. VANDENHEKKE
24 rue Jean Jaurès
59560 COMINES

Jean-Daniel DOBIN
PPC - T
77 rue du Cagire
31100 TOULOUSE

Cher ami,

La lecture de PPC - T m'amène aux disoutiens ci-après :

- * Les XROM : ils sont codés sur 16 bits (soit 2 octets), les 5 premiers bits désignent le préfixe, les 5 suivants la valeur du premier nombre qui désigne normalement un N° de périphérique, les 6 derniers donnent la valeur du second nombre qui désigne le N° d'ordre de la fonction désirée, ma remarque concerne essentiellement le préfixe.

- Dans un programme : le préfixe aura toujours la valeur $(10100)_2$ et correspond aux 8 premières colonnes de la ligne(A)₁₆ de la table des codes (A.F. Fig 9), une autre valeur entrainera une autre fonction définie par cette même table des codes

- Dans la table d'assignation : les fonctions sont codées sur 2 octets, si le premier digit du premier octet est égal à $(0)_{16}$, la fonction est définie par le second octet (fonction de 1 octet) ; si le premier digit est différent de $(0)_{16}$ nous avons affaire à une fonction de 2 octets et à un affichage XROM. Mais attention cet affichage XROM AA, BB ne donne aucune indication sur la valeur exacte du premier octet du code qui est à l'origine de cet affichage, le premier digit peut avoir toutes les valeurs comprises entre 1 et F, pour connaître la valeur de ce premier octet il faut aller voir directement dans la table d'assignation (bien que généralement nous créons nos assignations nous-mêmes !!)

- * Pointeur programme du registre b : ce pointeur 16 bits adresse des instructions de programme utilisateur, il peut se présenter sous 2 formes :

- bit 15 = 0 : programmes en mémoire vive (RAM)

$0 \ X_1 \ X_2 \ X_3 \ 0 \ 0 \ a_9 \ \dots \ a_{10}$

avec $X_1 \ X_2 \ X_3$: pointeur d'octet dans un registre (le code 111 est illégal car il n'y a que 7 octets)

$a_9 \ \dots \ a_{10}$: pointeur de registre sur 10 bits (peut adresser 2^{10} =

1024 registres de chacun 56 bits soit 7168 octets de mémoire vive)

- bit 15 = 1 : programme utilisateur en mémoire

morte sur l'un des connecteurs

$1 \ X_1 \ X_2 \ P \ a_{11} \ \dots \ a_{16}$

avec $X_1 \ X_2$: N° du connecteur

P : N° de page sur ce connecteur (il peut y avoir 2 mémoires mortes de 4K mots par connecteur)
 a₁₁ a₁₀ : pointeur de mots (chaque mot fait 10 bits),
 ($2^{12} = 4096$ mots)

* RCLFLAG: du module X Fonction: il rappelle dans le registre X l'état des drapeaux β à 43 sous forme de chaîne Alpha. Après exécution de RCLFLAG et décodage du contenu du registre X nous trouvons les codes décimaux suivants: 31 - 240 - β - β - 2 - 64 - 8 que nous pouvons écrire en Hexa : 1F - FX - XI - ... - XX, seuls les 3 premiers digits sont indépendants de l'état des drapeaux.

- 1^{er} digit (1)₁₆ : indique que le contenu du registre est une chaîne de caractères.

- 2^e et 3^e digit (FF)₁₆ : doivent permettre à STOFLAG de tester si la chaîne de caractères contenue dans X est bien le résultat d'une opération RCLFLAG

- digits suivants X ... X : représentent dans l'ordre l'état des drapeaux β à 43

Remarque : pour décoder le registre x avec le module X.FCT il ne faut pas faire Alpha CLA ARCL.X Alpha puis ATOX 7 fois, car dans ce cas le premier octet est perdu et ne sera pas décodé, d'autre part tous les octets nuls ne seront pas décodés la chaîne donnée ci-dessus aurait délivré successivement 240 - 2 - 64 - 8 - β - β ??

Personnellement j'utilise un petit programme (voir ci-dessous)

* Décodage et codage du registre x
 Ces 2 programmes sont très simples et utilisent le module X.FCT ainsi qu'un peu de programmation synthétique, ils n'occupent qu'une seule piste de carte magnétique

Si on fait IEQ "CDR" et qu'on entre successivement les valeurs 31 - 240 - β - β - 2 - 64 - 8, puis IEQ "STOFLAG", les drapeaux β à 43 sont initialisés, puis IEQ "RCLFLAG" et IEQ "DDR" nous retrouvons après chaque R/S la même chaîne de codes. Mais pour initialiser les drapeaux il est aussi possible de faire IEQ "CDR", d'entrer 7 valeurs décimales R/S puis STO d, de la même façon on peut initialiser n'importe quel "registre synthétique".

Décodage

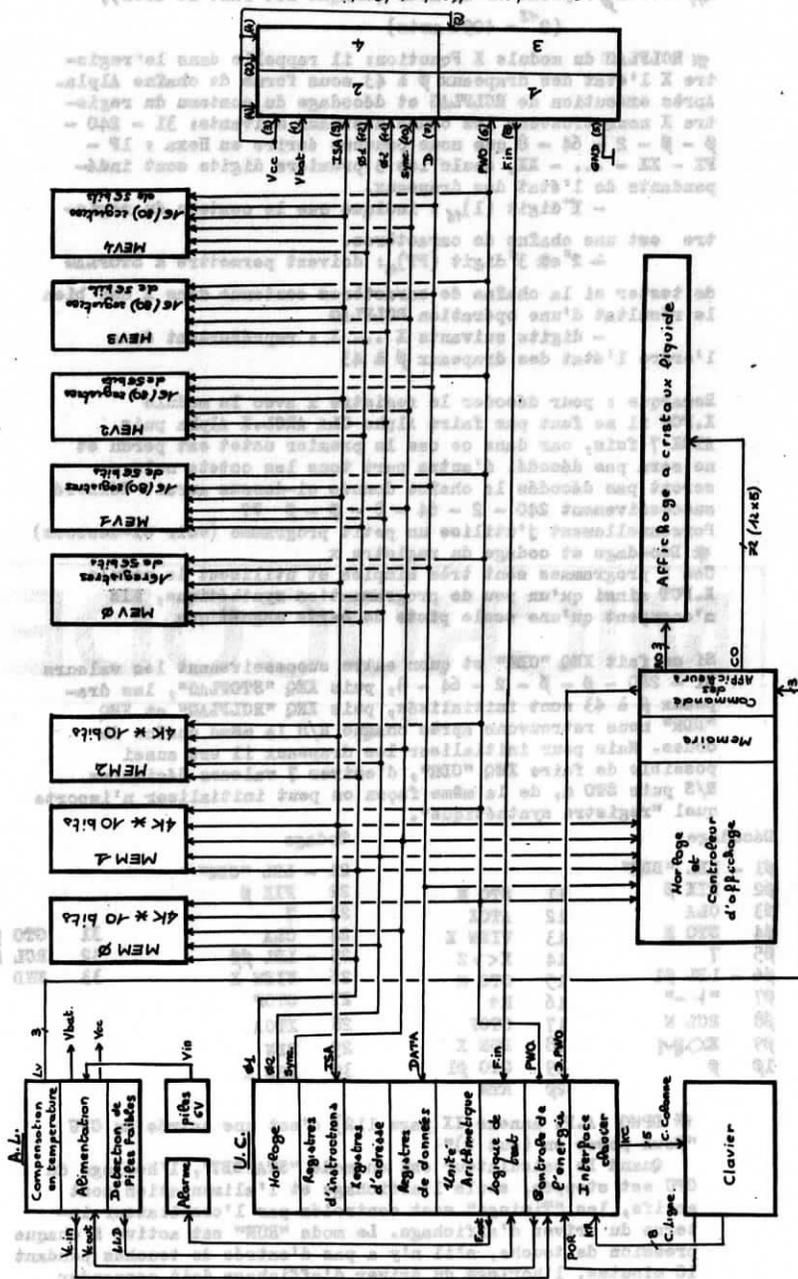
01 - LBL "DDR"	
02 - FIX β	11 STO N
03 - CLA	12 ATOX
04 - STO N	13 VIEW X
05 - 7	14 X<>Z
06 - LBL 01	15 STO N
07 - "└─"	16 R+
08 - RCL N	17 STOP
09 - E<>M	18 DSE X
10 - β	19 GTO 01
	20 RTN

Codage

21 - LBL "CDR"	
22 - FIX β	
23 - 7	
24 - CLA	31 GTO 00
25 - LBL 00	32 RCL N
26 - VIEW X	33 END
27 - STOP	
28 - XTOA	
29 - RIN	
30 - DSE X	

* DPWO (A.F. annexe IX page 112) c'est une entrée du CPU
 "Down power on (out ?)"

Quand le calculateur est en mode "STANDBY", l'horloge du CPU est stoppée, seuls l'affichage et l'alimentation sont actifs, les "Timings" sont contrôlés par l'oscillateur interne du driver d'affichage. Le mode "RUN" est activé à chaque pression de touche, s'il n'y a pas d'entrée de touches pendant 10 minutes, l'horloge du driver d'affichage doit commander l'arrêt du calculateur par l'intermédiaire de cette ligne de Bus : mode "SLEEP".



ANALYSE

12 juillet 1984

Jean-Louis BENOIST (1984)
40, rue de la République
93200 PANTIN

Cher Jean-Louis,

Enfin les vacances ! Et je pense au moment me consacrer à mes HP-71C. J'ai vu récemment l'HP-71C et je pense à son aspect et à son fonctionnement. J'ai vu récemment l'HP-71C et je pense à son aspect et à son fonctionnement. J'ai vu récemment l'HP-71C et je pense à son aspect et à son fonctionnement.

MICROCODE

Il est possible de modifier les instructions. En fait, il est possible de modifier les instructions. En fait, il est possible de modifier les instructions. En fait, il est possible de modifier les instructions.

Il est possible de modifier les instructions. En fait, il est possible de modifier les instructions. En fait, il est possible de modifier les instructions. En fait, il est possible de modifier les instructions.

12	11	10	9	8	7	6	5	4	3	2	1	0
12	11	10	9	8	7	6	5	4	3	2	1	0

Il est possible de modifier les instructions. En fait, il est possible de modifier les instructions. En fait, il est possible de modifier les instructions. En fait, il est possible de modifier les instructions.

ANALYSE ...

le 5 juillet 1983

Jean-Christophe BENDIST (T192)
40, rue de la République
77210 AVON

Cher Jean-Daniel,

Enfin les vacances ! Et je peux de nouveau me consacrer à ma HP-41C.
Dans ma dernière lettre, je te parlais de mon projet d'émulateur du
Processeur de la HP-41C en basic. Devant assurer mon passage en Mathématiques
spéciales, j'avais quelque peu délaissé tout cela ces derniers temps. Maintenant
c'est fait et j'ai repris l'affaire en main.

J'ai finalement été amené à débiter le basic pour deux raisons princi-
ales : - le basic est beaucoup trop lent, même sur TRS-80, et mon "prototype"
exécutait environ une instruction par seconde, ce qui est sans commune mesure avec
la rapidité réelle du langage machines HP.

- les instructions mettant en jeu les décalages, les levés de drapeaux etc... sont
très difficiles à faire en basic.

Mais j'ai repris espoir en recevant l'"éditeur assembleur plus" de Microsoft.
En effet, il est possible grâce à lui de fabriquer un espèce de "cross assembleur" grâce
à ses possibilités de macro-instructions. En gros, le principe est de créer pour chaque
instruction HP une série d'instruction en langage machine Z-80 (le microprocesseur de
TRS). Du coup, les deux problèmes que j'ai cités se trouvent résolus et je pense même
atteindre la même vitesse en émulateur qu'en langage réel ! J'ai déjà réalisé
quelques instructions sans difficultés particulières et je pense bien finir d'ici septem-
bre.

Mais revenons-en à l'objet principal de ma lettre : quelques éléments au sujet de
l'instruction STO 12 quand elle est assignée directement.

J'ai essayé d'aborder le problème par l'étude de la ROM HP, mais comme tu vas
le voir, cela ne fait qu'épaissir le mystère !

J'ai tout d'abord cherché à savoir où allait la HP quand on lui demandait d'exé-
cuter la fonction. La HP doit logiquement aboutir en 0091 hexa avec dans C [13,12]
le code de la fonction, c'est à dire 3C hexa. Cette routine est chargée de trouver
l'adresse de la fonction à partir de son code.

Voilà ce qui se passe :

Arrivée de la routine → C:

13 12 11 10 9 8 7 6 5 4 3 2 1 0	3 C	(ADR)	(S&X)
---------------------------------	-----	-------	-------

0091 LDI S&X → C:

3 C		0 1 4
-----	--	-------

0092 14 (hexa)

0093 RCR 9 → C:

	1 4 3 C	
--	---------	--

D'après la suite de la routine, le "poids faible" de l'adresse de "STO 12" se trouve
en 143C ce qui est intéressant puisque cette adresse tombe en zone programme et non
en zone donnée (on se retrouve de ce cas des TOUTES synthétiques). Continuons :

0084 FETCH S&X → C:

1	4	3	C	0	5	6
---	---	---	---	---	---	---

 Voilà! On a pris le code de C=β KS qui se trouve en 143C. Qu'est-ce que cela va donner?
 0085 LDR - 1 → C:

1	4	3	1	0	5	6
---	---	---	---	---	---	---

On met le "poids fort" de l'adresse au quartet n°3.

0086 RCR 11 → C:

ADR			S&X			
1	4	3	1	0	5	6

 Voilà l'adresse de notre fonction: 1056 hexa, en effet:
 0087 GTO ADR

Qu'il y a-t-il à cette adresse? On tombe un pas avant la routine "addition de deux nombres de 10 digits contenus ds A et dans C", ce pas n'a aucune importance puisqu'il s'agit d'un NOP:

1056 NOP
 1057 ?NC XA
 1058 → 1207 ← routine addition
 1059 ?NC 60
 1060 → 00DA ← fin routine concernant X et Y (descumbe pile, sauvegarde X dans L)

La fin semble confirmer les observations de ~~Patrice~~ Patrice Nolin (T263) ds le dernier n° de PPT. Mais que penser de la routine "addition" alors que le résultat fourni par cette fonction semble directement influencé par le registre d (flagi)?

En effet, un SF₀ ajoute un signe "moins" au résultat fourni, ainsi que le veut le format des registres HP.41. Baisser ou lever différents flags modifie de même le résultat.

Cependant le décodage d'un "RCL d" et du résultat montre que les deux n'ont rien à voir du point de vue structure... Mystère...

Autre chose: si on réinjecte la valeur obtenue dans le registre d, on obtient des choses très bizarres.

- Tout d'abord on passe en mode "alpha" avec "shift allumé"
- En revenant en mode normal, il n'y a plus d'erreur "OUT OF RANGE" quand on met un argument ≠ 0. Mais cette fonction semble transjérer X ds le registre d'affichage puis fait un CLR! Rien ne change à l'affichage mais X contient 0.

Bizarres quand on pense que tout cela est le fruit d'une addition!

Pro.: cette fonction recèle encore de nombreux mystères qu'il faut élucider!

En attendant: Heureuse Programmation.

✶ Fin ✶

ASS LANG OU M.C.

ASSEMBLEUR, LANGAGE MACHINE ou M.C. ?

Il peut être utile d'essayer de donner qq indications au sujet des 3 expressions citées dans le titre de ce "papier".

Rassurez-vous, je ne vais pas vous faire un exposé sur les micro-processeurs; une littérature abondante et très indigeste existe déjà sur le sujet.....

L'âme même d'un ORDI, donc de son micro-processeur, n'est composée que d'une multitude de "très petits éléments" pouvant être placés dans 2 états stables (bistables) et c'est à partir de cela que les concepteurs doivent bâtir un microprocesseur.

C'est donc l'utilisation de ces "oases" qui peuvent être activées ou désactivées (à '1' ou à '0') et qui sont groupées par "paquets", qui va permettre de faire des PRGM (et oui, déjà !) qui pourront être utilisés en adoptant leur LANGage qui est appelé LANGage MACHINE. On ne voit pas pourquoi ce LANG n'a pas été baptisé LANGage micro-processeur (?).

Il semblerait que le terme "MICROCODE" puisse être assimilé à "LANG MACH" mais signalons que les professionnels de l'informatique semblent ignorer le terme "MICROCODE".

Lorsque nous "lançons" notre PRGM préféré, notre '41' INTERPRETE en COMPILANT (d'où les noms: INTERPRETEUR ou COMPILATEUR) ce PRGM pour le traduire en LANG d'ASSEMBLAGE puis en LANG MACH, ce dernier étant le dernier maillon accessible de la chaîne.

Le MC serait donc le LANG (MACH ?) qui permettrait de "piloter" directement les Fonctions (PRGM) du micro-processeur en "shuntant" les opérations précitées.

Le LANG que nous utilisons pour écrire nos PRGM peut être assimilé à un ASSEMBLEUR "évolué" (°) mais il est cependant et tout de même, ASSEMBLÉ par le logiciel de notre machine. On peut signaler que le travail effectué à ce niveau est beaucoup moins important que pour un LANG dit "évolué" tel que le BASIC, par exemple.

Pour pouvoir utiliser des Fonctions ou des PRGM écrits en MC, il suffit d'accepter cette petite contrainte qu'est "le fil-à-la-patte" que constitue un LECTEUR d'EPROM mais ce dernier peut être placé dans un coffret très plat, de la grandeur de notre '41', ces deux éléments pouvant être introduits dans la sacoche fournie avec la machine. Je ne comprends pas pourquoi les lecteurs d'EPROM disponibles sur le marché n'ont pas rencontré le succès qu'ils méritaient. Ils présentent également l'avantage non négligeable, de permettre d'utiliser des PRGM écrits en LANG "utilisateur" (avec ou sans PS) sans encombrer la MEM centrale de notre machine. Dans ce dernier cas, il est nécessaire d'avoir recours à un MLDL et à un PRGMMEUR d'EPROM pour charger ces PRGM sur EPROM mais votre Club préféré peut se charger de ce travail.

Je crois savoir que je suis bien loin d'avoir répondu à la question posée dans le titre de ce "papier" mais je pense cependant avoir peut-être éclairé quelques collègues sur les "grandes lignes" du MC et de son utilisation qui offre des possibilités presque illimitées.

(°) On appelle LANGage "évolué" tout LANG qui facilite la PRGMmation et qui donc, se rapproche le plus possible de notre forme d'expression habituelle. Le BASIC qui est, pour l'instant, le LANG le plus répandu, est donc un LANG "évolué" et si je prends cela comme un postulat, j'en déduirais que plus le LANG est évolué, moins les utilisateurs ont besoin de l'être.... (à suivre).

rS (T178) Robert Schwartz 144 Av Carnot Bondy

CALL

MARC CHIFFOLEAU 9710-T189
8 RUE AUBER
59046 LILLE

PROGRAMME CALL

CETTE NOUVELLE FONCTION PERMET DE LANCER DIRECTEMENT AU CLAVIER OU PAR PROGRAMME UN PROGRAMME MICROCODE, EN MODE PROGRAMME, LA FONCTION CALL DOIT ETRE SUIVIE D'UNE CHAINE DE 4 CARACTERES DONNANT L'ADRESSE EN HEXADECIMAL DU PROGRAMME MICROCODE A LANCER. PAR EXEMPLE, POUR LANCER LE PROGRAMME "BEEP", ON FERA:

```
01 CALL  
02 "10B8"
```

POUR L'UTILISATION DIRECTE AU CLAVIER, LA FONCTION DEMANDE L'ENTREE DE 4 CHIFFRES HEXADECIMAUX:

```
CALL _ _ _ _
```

SEULES LES TOUCHES "0" A "9", ET "A" A "F" SONT VALIDES, AINSI QUE LA TOUCHE DE CORRECTION <- .

CETTE FONCTION PERMET AUSSI DE PASSER UN PARAMETRE AU PROGRAMME MICROCODE PAR L'INTERMEDIAIRE DE X, QUI EST TRANSFERE DIRECTEMENT EN C.

```
9D80 08C L :  
9D81 00C L : NOM DE LA FONCTION  
9D82 001 A :  
9D83 003 C :  
9D84 20C ?FSET 13  
9D85 005 : SAUT EN 9E01 SI SETF 13  
9D86 27B ?C GO 9E01 : --> PRGM RUNNING  
9D87 3C1  
9D88 0B0 ?NC XQ 2CF0 : EFFACE L'ECRAN  
9D89 3BD  
9D8A 01C ?NC XQ 07EF : EDITION D'UN MESSAGE  
9D8B 003 C  
9D8C 001 A  
9D8D 00C L  
9D8E 00C L  
9D8F 020  
9D90 21F : MESSAGE "CALL _"  
9D91 04E C=0 ALL  
9D92 058 G=C GR,+ : INITIALISATION DE G A 0  
9D93 12D  
9D94 038 ?NC XQ 0E4B : S/PRGM D'ENTREE DE PARAMETRES  
9D95 013 JNC +02 : RETOUR SI CORRECTION ( <- )  
9D96 019 JNC +03 : RETOUR SI AUTRE TOUCHE  
9D97 049  
9D98 036 ?NC GO 0D12 : FIN DE PRGM : FONCTION ANNULEE  
9D99 3B1
```

9D9A 274 ?NC XQ 9DEC : VERIFIE LA VALIDITE DE L'ENTREE
 9D9B 363 JNC -14 : RETOUR SI CARACTERE INVALIDE
 9D9C 04E C=0 ALL
 9D9D 058 G=C @R,+ : INITIALISATION DE G A 0
 9D9E 12D
 9D9F 038 ?NC XQ 0E4B : S/PRGM D'ENTREE DE PARAMETRES
 9DA0 013 JNC +02 : RETOUR SI CORRECTION (< -)
 9DA1 02B JNC +05 : RETOUR SI AUTRE TOUCHE
 9DA2 130 LDI S&X
 9DA3 020 32(d) : ECRITURE D' UN ESPACE A GAUCHE DE
 9DA4 3A8 WRIT 14(d) : L'ECRAN = EFFACE LE CHR DE DROITE
 9DA5 313 JNC -1E
 9DA6 3B1
 9DA7 274 ?NC XQ 9DEC : VERIFIE LA VALIDITE DE L'ENTREE
 9DA8 3A3 JNC -0C : RETOUR SI CARACTERE INVALIDE
 9DA9 04E C=0 ALL
 9DAA 058 G=C @R,+ : INITIALISATION DE G A 0
 9DAB 121
 9DAC 038 ?NC XQ 0E48 : S/PRGM D'ENTREE DE PARAMETRES
 9DAD 013 JNC +02 : RETOUR SI CORRECTION (< -)
 9DAE 02B JNC +05 : RETOUR SI AUTRE TOUCHE
 9DAF 130 LDI S&X
 9DB0 020 32(d) : ECRITURE D' UN ESPACE A GAUCHE DE
 9DB1 3A8 WRIT 14(d) : L'ECRAN = EFFACE LE CHR DE DROITE
 9DB2 353 JNC -16
 9DB3 3B1
 9DB4 274 ?NC XQ 9DEC : VERIFIE LA VALIDITE DE L'ENTREE
 9DB5 3A3 JNC -0C : RETOUR SI CARACTERE INVALIDE
 9DB6 04E C=0 ALL
 9DB7 058 G=C @R,+ : INITIALISATION DE G A 0
 9DB8 115
 9DB9 038 ?NC XQ 0E45 : S/PRGM D'ENTREE DE PARAMETRES
 9DBA 013 JNC +02 : RETOUR SI CORRECTION (< -)
 9DBB 02B JNC +05 : RETOUR SI AUTRE TOUCHE
 9DBC 130 LDI S&X
 9DBD 020 32(d) : ECRITURE D' UN ESPACE A GAUCHE DE
 9DBE 3A8 WRIT 14(d) : L'ECRAN = EFFACE LE CHR DE DROITE
 9DBF 353 JNC -16
 9DC0 3B1
 9DC1 274 ?NC XQ 9DEC : VERIFIE LA VALIDITE DE L'ENTREE
 9DC2 3A3 JNC -0C : RETOUR SI CARACTERE INVALIDE
 9DC3 30D
 9DC4 0AC ?NC XQ 2BF7 : JUSTIFICATION A GAUCHE
 9DC5 319
 9DC6 038 ?NC XQ 0EC6 : TESTE LE "NULL" DE TOUCHE
 9DC7 138 READ 4(L) : DECALLE L'AFFICHAGE DE 5 CARAC-
 9DC8 3F8 READ 15(e) : TERES VERS LA GAUCHE
 9DC9 130 LDI S&X
 9DCA 093 147(d) : MISE EN PLACE DE POINTEURS EN A
 9DCB 39C R=0
 9DCC 33C RCR 1
 9DCD 0AE A<>C ALL
 9DCE 3F8 READ 15(e) : LECTURE DU PREMIER CHIFFRE HEXA
 9DCF 358 ST=C
 9DD0 04C ?FSET 4 : SI SETF 4, ALORS < A
 9DD1 017 JC +02
 9DD2 202 C=C+A @R : SI >= A, ON LUI AJOUTE 9(d)
 9DD3 0E2 C<>B @R
 9DD4 198 C=M ALL
 9DD5 2FC RCR 13
 9DD6 0E2 C<>B @R
 9DD7 158 M=C ALL : SAUVEGARDE A DROITE DE M
 9DD8 1BE A=4-1 MS
 9DD9 3AB JNC -0B : CHIFFRE HEXA SUIVANT
 9DDA 3C1

```

9DD8 0B0 ?NC XQ 2CF0 : EFFACE L'ECRAN
9DDC 171
9DDD 01C ?NC XQ 075C : RESTITUE LES INDICATEURS
9DDE 211
9DDF 00C ?NC XQ 0384 : BAISSSE DIVERS FLAGS
9DE0 198 C=M ALL : RAPPELLE L'ADRESSE CALCULEE EN C
9DE1 046 C=0 S&X : RECHERCHE DU PREMIER MOT DU ROM
9DE2 1BC RCR 11 : CORRESPONDANT
9DE3 330 FETCH S&X
9DE4 2E6 ?C#0 S&X : SI = 0, IL N'Y A PAS DE ROM, DONC
9DE5 381
9DE6 00A ?NC GO 02E0 : ----> "NONEXISTENT" !!!!
9DE7 198 C=M ALL
9DE8 1BC RCR 11 : SI LE ROM EXISTE, ON POUSSE
9DE9 170 PUSH ADR : L'ADRESSE DANS LA PILE DE RETOUR
9DEA 0FB READ 3(X) : MET 3(X) DANS L'ACCUMULATEUR C
9DEB 3E0 RTN : BRANCHEMENT A L'ADRESSE DESIREE.

```

SOUS-PROGRAMME DE VERIFICATION

```

9DEC 00C ?FSET 3 : SI SETF 3, CHR COMPRIS ENTRE 0-9
9DED 067 JC +0C
9DEE 04C ?FSET 4 : SI SF4, TOUCHE "LOCALE", A-J
9DEF 3A0 ?NC RTN : SI CF3 ET CF4, INVALIDE => RTN
9DF0 35E ?A#0 MS
9DF1 3A0 ?NC RTN : TOUCHE J : INVALIDE => RTN
9DF2 130 LDI S&X
9DF3 007 7(d)
9DF4 33C RCR 1
9DF5 31E ?A<C MS : TOUCHE > F ?
9DF6 3A0 ?NC RTN : SI OUI, INVALIDE => RTN
9DF7 046 C=0 S&X : ON A UNE TOUCHE < A-F >
9DF8 01B JNC +03
9DF9 130 LDI S&X
9DFA 003 3(d) : POUR LES TOUCHES < 0-9 >, ON
9DFB 0BE A<>C MS : AJOUTE 30(H) POUR AVOIR LE CODE
9DFC 2FC RCR 13 : ECRAN
9DFD 3E8 WRIT 15(e) : ECRIT LE CARACTERE A L'ECRAN
9DFE 1B0 POP ADR
9DFF 23A C=C+1 M : RTN+1 =>
9E00 1E0 GOTO ADR : CARACTERE VALIDE.

```

CALL PROGRAMMABLE

```

9E01 0A0 SLCT P
9E02 01C R=3
9E03 141
9E04 0A4 ?NC XQ 2950 : S/PRGM DE LECTURE DU POINTEUR RAM
9E05 33D
9E06 0A4 ?NC XQ 29CF : DECREMENTE LE POINTEUR PRGM
9E07 2C1
9E08 0A4 ?NC XQ 29B0 : CHARGE EN C XP L'OCTET INDIQUE
9E09 056 C=0 XS : PAR LE POINTEUR PRGM
9E0A 06A A<>B R<-
9E0B 0A6 A<>C S&X,
9E0C 130 LDI S&X
9E0D 0F4 244(d) : VERIFIE QUE L'OCTET LU EST "F4"
9E0E 366 ?A#C S&X : CE QUI INDIQUE UNE CHAINE DE 4
9E0F 01B JNC +03 : CARACTERES
9E10 0B5 : SI L'OCTET EST DIFFERENT DE F4,
9E11 0A2 ?NC GO 282D : ----> "DATA ERROR"
9E12 0E0 SLCT Q : MISE EN PLACE DU POINTEUR DE
9E13 05C R=4 : COMPTAGE DES 4 CARACTERES
9E14 0A0 SLCT P
9E15 06A A<>B R<-

```

```

9E16 33D
9E17 0A4 ?NC XQ 29CF : DECREMENTE LE POINTEUR PRGM
9E18 2C1
9E19 0A4 ?NC XQ 29B0 : CHARGE EN C XP L'OCTET INDIQUE
9E1A 056 C=0 XS : PAR LE POINTEUR PRGM
9E1B 06A A<>B R<-
9E1C 0A6 A<>C S&X
9E1D 130 LDI S&X
9E1E 030 48(d) : SI L'OCTET EST INFERIEUR A 48(d)
9E1F 306 ?A<C S&X : ( CODE DU CARACTERE 0 )
9E20 387 JC -10 : ----> "DATA ERROR"
9E21 130 LDI S&X
9E22 047 71(d) : SI L'OCTET EST SUPERIEUR A 70(d)
9E23 306 ?A<C S&X : ( CODE DU CARACTERE F )
9E24 363 JNC -14 : ----> "DATA ERROR"
9E25 130 LDI S&X
9E26 03A 58(d)
9E27 306 ?A<C S&X : SI L'OCTET EST COMPRIS ENTRE
9E28 047 JC +08 : 57 ET 65
9E29 130 LDI S&X
9E2A 041 65(d) : CODE DU CARACTERE 9
9E2B 306 ?A<C S&X : ET DU CARACTERE A
9E2C 32F JC -1C : ----> "DATA ERROR"
9E2D 130 LDI S&X
9E2E 009 9(d)
9E2F 146 A=A+C S&X : DECODAGE DE L'OCTET
9E30 0A6 A<>C S&X
9E31 33C RCR 1
9E32 0BE A<>C MS
9E33 198 C=M ALL
9E34 0BE A<>C MS
9E35 2FC RCR 13
9E36 158 M=C ALL : STOCKAGE EN M
9E37 0E0 SLCT Q
9E38 3D4 R=R-1
9E39 394 ?R=0 : A-T-ON LU 4 OCTETS ? SI NON,
9E3A 2D3 JNC -26 : LECTURE DE L'OCTET SUIVANT
9E3B 198 C=M ALL : RAPPELLE L'ADRESSE CALCULEE EN C
9E3C 046 C=0 S&X : RECHERCHE DU PREMIER MOT DU ROM
9E3D 1BC RCR 11 : CORRESPONDANT
9E3E 330 FETCH S&X
9E3F 2E6 ?C#0 S&X : SI = 0, IL N'Y A PAS DE ROM, DONC
9E40 381
9E41 00A ?NC 80 02E0 : ----> "NONEXISTENT" !!!!
9E42 0A0 SLCT P
9E43 06A A<>B R<-
9E44 149
9E45 024 ?NC XQ 0952 : SELECTIONNE LES REGISTRES D'ETAT
9E46 0CC ?FSET 10 : SI SF 10, ON EST EN ROM
9E47 03F JC +07
9E48 09A A<>C M : CALCUL DU NOUVEAU POINTEUR RAM
9E49 1FA C=C+C M : A METTRE EN 12(b)
9E4A 1FA C=C+C M
9E4B 1FA C=C+C M
9E4C 33C RCR 1
9E4D 0BA A<>C M
9E4E 338 READ 12(b) : RESTITUE LE POINTEUR PROGRAMME
9E4F 0AA A<>C R<-
9E50 328 WRIT 12(b) : EN 12(b)
9E51 198 C=M ALL
9E52 1BC RCR 11 : SI LE ROM EXISTE, ON POUSSE
9E53 170 PUSH ADR : L'ADRESSE DANS LA PILE DE RETOUR
9E54 0F8 READ 3(X) : MET 3(X) DANS L'ACCUMULATEUR C
9E55 3E0 RTN : BRANCHEMENT A L'ADRESSE DESIREE

```

HP-1x

FAUX

T 386 Alexis Kupiec
2, avenue Kennedy Cuiry 59500 Douai
rubrique programmes
machines: HP11c HP15c

DETECTION DE FAUX BILLETS.

(convient pour les billets de 500, 100, 50, 10 Frs)
Le programme détermine le numéro central d'un billet
(celui qui est situé en haut et au centre) à partir des données
du bas:

1	LBL .1	25	RCL .1
	CF 0		2
	CF 1		.
	CF 6		5
5	STO I		EEX
	IF I	30	6
	R/S		X
	STO 0		STO .1
	GSB A		RCL 0
10	1		1
	STO .0	35	EEX
	R/S		5
	STO .1		X
	1		STO +.1
15	STO -.1		RCL .2
	2 Z Y	40	STO +.1
	R/S		RCL .1
	STO .2		F? G
	LOG		GSB B
20	5		PSE
	2 Z Y?	45	RTN
	IF 6		LBL A
	F? G		RCL 0
	GSB B		9

50 $\alpha \leq y?$
 GSB B
 RCL .0
 2
 2
 F \div 5
 55 GSB C
 LBL B
 1
 F?A.
 STO-.0
 60 RTN
 LBL B
 1
 STO-.0
 RTN
 65 LBL C
 2
 5
 STO.0
 RTN
 70 LBL 7
 RCL.2
 STO.3
 1
 0
 75 \div
 INT
 STO.2
 RCL.3
 1
 80 0
 \div
 FRAC

1
 0
 85 X
 STO.3
 RTN
 LBL 8
 RCL.1
 90 1
 0
 X
 RCL.3
 +
 95 RTN



XUAN

Execution: On introduit:
 - la catégorie du billet: 1 s'il est ancien, 0 s'il appartient aux nouvelles séries;
 - le rang de la lettre dans l'alphabet (ex A=1 B=2... Z=26);
 - le numéro qui lui est juxtaposé;
 - l'autre numéro situé dans le bas.

Exemple: avec un billet de 10 fcs à l'effigie de Berlioz portant les indications en bas:

590814	Z. 292.
catégorie: Ancienne	1 R/s
Z: 26 ^{ème} rang	26 R/s
	292 R/s
	59 0814 R/s

la machine affiche 7298530814 le nombre devrait figurer (normalement) en haut et au centre.

Note: Si le rang de la lettre dans l'alphabet est compris entre 1 et 22, l'introduction de la catégorie est inutile par contre si ce rang est compris entre 23 et 26, elle est nécessaire. ³

*** PPC-T ***

GRAPHIQUE

Ce programme est destiné à la création de graphismes sur l'afficheur de l'HP-71. A la mise sous tension, écrire "RUNEDITEUR" puis presser 'ENDLINE'. L'HP-71 attend que l'utilisateur ait sélectionné une option dans le menu; pour ce faire, on utilise la touche flèche vers le bas ou flèche vers le haut. Sont présents les choix suivants :

- Effacement des lutins.
- Creation de lutins.
- Stockage des lutins.
- Rappels des lutins.
- Liste des lutins.
- Visualisation ASCII.
- Assignations.
- Assignations-series.
- Assignations globales.
- Assignations-liste.
- Assignations blanches.
- Codage des lutins.
- Codage global.
- Décodage global.
- Mode HP-71.



Par lutins on signifie un caractère graphique (code supérieur à 127), défini par l'instruction 'CHARSET'.

Pour sortir du menu il y a 2 solutions : valider une option par 'ENDLINE' lorsque celle-ci est affichée, appuyer sur 'Z' pour mettre fin à ce programme.

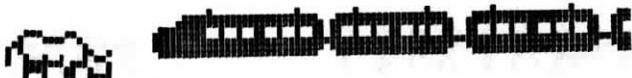
CREATION DES LUTINS.

Si, préalablement, nous avons déjà créé des caractères graphiques, la chaîne "CHARSET" est affichée aux fins de modifications ou d'ajouts. Dans l'hypothèse contraire l'affichage est réduit à un point clignotant en haut à gauche; on considèrera ce petit point comme un curseur de positionnement. Les touches '1' et '0' sont destinées à la validation du point à l'emplacement du curseur. C'est à dire qu'en appuyant sur '1', le point noir restera visible à l'écran et le curseur se déplacera d'un cran vers le bas; la pression du '0' provoquera le déplacement du curseur en laissant un espace vide. Lorsque le curseur parvient en bas d'une colonne il poursuit en haut de la colonne suivante. D'autre part, les 8 flèches de déplacement sont disponibles pour la manipulation du curseur,

- 'I/R' insère une colonne,
- '-BACK' efface la colonne courante,
- 'ENDLINE' provoque le retour au menu.

STOCKAGE ET RAPPEL DES LUTINS

Après la pression de 'ENDLINE', apparait une interrogation concernant le nom du fichier à créer, ou contenant une définition d'un ensemble de caractère selon l'option sélectionnée.



LISTE DES LUTINS ET VISUALISATION ASCII

Ces 2.options sont destinées à consulter les tables de caractères standards ou graphiques. Dans cette configuration, le choix d'un contraste est offert, ainsi que l'affichage sélectif d'un caractère dont on connaît la valeur décimale :

Exemple : Visualisation ASCII. 'ENDLINE'.
Le jeu standard ? N 'ENDLINE'
Contraste ? 9 'ENDLINE'
Pour visualiser un caractère choisi, le caractère est présent au clavier, frapper la touche correspondante ex : !V.....

Si le caractère n'est pas immédiatement disponible (ex BETA), appuyez directement sur 5 suivi de 'ENDLINE'.
Retour au menu par 'ENDLINE'

LES ASSIGNATIONS

Le manuel de l'utilisateur contient un programme 'CINEMA', p138 qui utilisera une chaîne entrée au clavier. Il est possible d'entrer une chaîne graphique en utilisant les assignations. Afin d'assouplir l'affectation des graphismes, plusieurs options sont réservées :

Assignations. 'ENDLINE' affiche la chaîne disponible, le caractère courant est clignotant. On choisit le caractère à assigner au moyen des 2 flèches gauche et droite, 'ENDLINE' provoque l'apparition du caractère retenu suivi de la demande de la touche ; la réponse est conforme à la spécification de DEF KEY; ex FV assigne un caractère à la place de 'PURGE'.

Assignations-series. 'ENDLINE' assigne chaque caractère successivement aux touches pressées.

Assignations globales. 'ENDLINE' présente la chaîne par défaut, que l'on peut encore modifier avec l'éditeur du HP-71, 'ENDLINE' demande la touche etc...

Assignations blanches. permet de faire le ménage !

CODAGE DES LUTINS

Transcrit les graphismes en suite des nombres décimaux propres à être entrés dans le cours d'un programme.

DECODAGE GLOBAL

Est utilisé pour la création de graphisme dont les codes sont déjà connus.

Ex : 112 248 248 248 252 254 254 254 231 230 254 230 230 254 230
230 254 254 230 230 254 230 230 255 230 230 254 124 32 MEUH!

MODE HP-71.

Autorise les calculs en cours de création de chaînes. Pour reprendre le déroulement normal de l'éditeur graphique, 'CONT'.

Espérons que ce programme utilitaire favorisera l'animation et la mise au point d'une créativité non dépourvue de caractère



```

10 ! EDITEUR DE LUTINS de Serge Vaudenay
20 DESTROY ALL @ INTEGER A,L,M @ M=15
30 OPTION BASE 0 @ DIM L$(M),M$(M)(3),B$(132)
40 FOR A=0 TO M @ READ L$(A),M$(A) @ NEXT A
50 DISP L$(L) @ BEEP MAXREAL,EPS
60 AS=KEY$ @ IF AS="" THEN 60
70 IF AS="#50" THEN LET L=(L-1)*SGN(L) @ GOTO 50
80 IF AS="#51" THEN LET L=(L+1-M)*(L#M)*M @ GOTO 50
90 IF AS="#162" THEN LET L=0 @ GOTO 50
100 IF AS="#163" THEN LET L=M @ GOTO 50
110 IF AS="#38" THEN GOSUB M$(L) @ GOTO 50
120 IF AS="#43" OR AS="Z" THEN END
130 GOTO 60
140 DATA "Options de l'editeur :", "GET"
150 DATA "Effacement des lutins.", "INI"
160 DATA "Creation de lutins.", "CRL"
170 DATA "Stockage des lutins.", "STL"
180 DATA "Rappels des lutins.", "RCL"
190 "INI": CHARSET "" @ RETURN
200 "CRL": DIM B$(132) @ INTEGER W,X,Y @ B$=CHARSETS(X,132*X) @ X=1 @
Y=1
210 B=MOD(INT(NUM(B$(X))/2^Y*2),2) @ "DSB": GDISP B$
220 C=INT(FP(TIME)*2) @ GOSUB 400
230 AS=KEY$ @ IF AS="" THEN "DSB"
240 C=B @ GOSUB 400 @ IF AS="#47" THEN X=(X-2)*(X#1)-1 @ GOTO 210
250 IF AS="#48" THEN X=X+SGN(MOD(X,132)) @ GOTO 210
260 IF AS="#50" THEN Y=(Y-2)*(Y#1)-1 @ GOTO 210
270 IF AS="#51" THEN Y=Y+(Y#8) @ GOTO 210
280 IF AS="#159" THEN X=1 @ GOTO 210
290 IF AS="#160" THEN X=LEN(CHARSET$)+1 @ GOTO 210
300 IF AS="#162" THEN Y=1 @ GOTO 210
310 IF AS="#163" THEN Y=8 @ GOTO 210
320 IF AS="#38" THEN RETURN
330 IF AS="#104" THEN BS=B$(0,X-1)&B$(X+1) @ GOTO 210
340 IF AS="f" THEN BS=B$(0,X-1)&CHR$(0)&B$(X) @ GOTO 210
350 IF AS#"0" AND AS#"1" THEN "DSB" ELSE C=VAL(AS)
360 GOSUB 400 @ CHARSET B$ @ Y=Y+1 @ IF Y#9 THEN 210
370 Y=1 @ X=X+1 @ IF MOD(X-1,6)#0 THEN 210
380 DISP "CHR$(;127+INT(X/6);)"=&CHR$(127+INT(X/6)) @ WAIT 2
390 IF X=132 THEN RETURN ELSE 210
400 Z=2^(Y-1) @ T=NUM(B$(X))
410 BS=B$(0,X-1)&CHR$(INT(T/Z/2)*Z*2+C*Z+MOD(T,Z))&B$(X+1,132) @ RETURN
420 "STL": INPUT "Nom du fichier ? ";N$ @ PURGE N$ @ CREATE TEXT N$ @
ASSIGN #1 TO N$
430 PRINT #1;CHARSET$ @ ASSIGN #1 TO * @ RETURN
440 "RCL": INPUT "Nom du fichier ? ";N$ @ ASSIGN #1 TO N$
450 READ #1;B$ @ CHARSET B$ @ ASSIGN #1 TO * @ RETURN
460 DATA "Liste des lutins.", "LIL"
470 "LIL": FOR A=128 TO INT(LEN(CHARSET$)/6)+127
480 DISP "CHR$(;A;)"=&CHR$(A) @ WAIT .5
490 IF KEYDOWN("#38") THEN RETURN
500 NEXT A @ RETURN
510 DATA "Visualisation ASCII.", "VIA"
520 "VIA": INPUT "Le jeu standard ? ",N:AS @ FOR A=0 TO 127 @ IF
AS="0" THEN 480
530 INPUT "Contraste ? ",9:K @ CONTRAST K
540 AS=KEY$ @ IF AS="" THEN 540
550 IF AS="#38" THEN DISP @ RETURN
560 IF NUM(AS)<49 OR NUM(AS)>57 THEN DISP AS; @ GOTO 540
570 PUT AS @ INPUT A @ DISP CHR$(A); @ GOTO 540

```

TGV
ERIC
SINGOUX

```

580 DATA "Assignations.", 'ASS'
590 DATA "Assignations-series.", 'ASG'
600 DATA "Assignations globales.", 'AST'
610 DATA "Assignations-liste.", 'ASL'
620 DATA "Assignations blanches.", 'CLA'
630 BS=' ' @ FOR A=128 TO LEN(CHARSET$)/6-127 @ BS=BS&CHR$(A) @
NEXT A @ A=128 @ RETURN
640 DISP BS$(0,A-128)&CHR$(INT(FP(TIME)*2)*(NUM(BS$(A-127))-32)+32)
&BS$(A-126,LEN(BS)-1)
650 AS=KEY$ @ IF AS='' THEN 640
660 IF AS='#38' THEN RETURN
670 IF AS='#48' THEN A=A+(A#LEN(BS)+127)
680 IF AS='#47' THEN A=A-(A#128)
690 GOTO 640
700 'ASS': GOSUB 630 @ GOSUB 640
710 DISP CHR$(A)&' '; @ INPUT 'Touche ? ';K$ @ IF K$='' THEN 700
720 DEF KEY K$,CHR$(A);
730 RETURN
740 DATA "Codage des lutins.", 'COD'
750 DATA "Codage global.", 'COG'
760 DATA "Decodage global.", 'DEG'
770 'COG': A=1 @ 'R': DISP @ FOR B=0 TO 5 @ DISP '
&STR$(NUM(CHARSET$(A+B))); @ NEXT B
780 IF KEYDOWN('#38') OR KEYDOWN('#48') AND A+5=LEN(CHARSET$) THEN
DISP @ RETURN
790 IF KEYDOWN THEN A=A+6*KEYDOWN('#48') @ GOTO 'R' ELSE 780
800 'ASG': GOSUB 630 @ FOR A=128 TO 127+LEN(CHARSET$)/6
810 DISP BS&' TO '; @ INPUT '';K$ @ DEF KEY K$,CHR$(A); @ BS=BS$(2)
@ NEXT A @ RETURN
820 'CLA': USER OFF @ INPUT 'Touche ? ';K$ @ IF K$='' THEN USER
ON @ RETURN
830 DEF KEY K$ @ GOTO 820
840 'AST': GOSUB 630 @ INPUT 'Chaîne ? ',BS;BS @ INPUT 'Touche ?
';K$
850 DEF KEY K$,BS; @ RETURN
860 'ASL': LIST KEYS @ RETURN
870 DATA "Mode HP-71.", 'M71'
880 'M71': DISP 'ED:' @ SFLAG -26 @ WINDOW 4 @ PAUSE @ WINDOW 1 @
CFLAG -26 @ RETURN
890 'COD': GOSUB 630 @ GOSUB 640
900 A=6*(A-128)+1
910 FOR B=1 TO 6 @ DISP ' &STR$(NUM(CHARSET$(A+B-1))); @ NEXT B @
DISP @ WAIT 5 @ RETURN
920 'DEG': BS=''
930 FOR B=1 TO 6 @ INPUT C$ @ IF C$='' AND FLAG(5,0) THEN CHARSET
BS @ RETURN
940 BS=BS&CHR$(VAL(C$)) @ NEXT B
950 BEEP @ SFLAG 5 @ GOTO 930

```

Ces 4 pages ont été écrites, mises en pages, y compris le programme sur une machine à écrire EP44. Ce matériel est peu coûteux, il peut être interfacé avec l'HP-71; le traitement de texte du 71, bien que lent, fonctionne correctement.

Les dessins ont été imprimés sur table traçante TANDY au moyen du module PANAME, sur HP-41 SERGE VAUDENAY T270.

MATH

Après être fort désagréablement surpris par le module Math de la HP-41C, j'ai longuement hésité à acheter celui-ci. Pourtant, que de différences ! Quelle puissance !

Le module se présente exactement sous le même encombrement que le cabochon qu'il est nécessaire de retirer pour le mettre en place. Il abrite un fichier LEX nommé MATHROM long de 32745 octets.

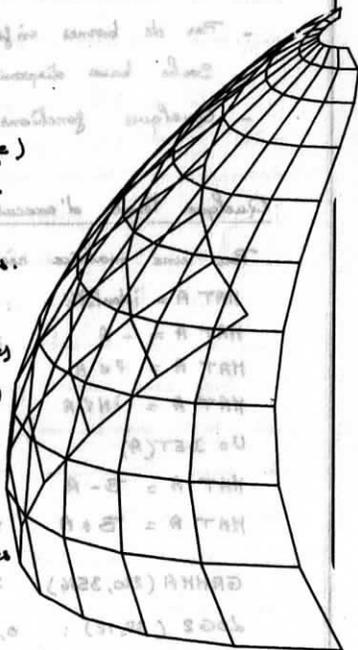
Ce type de fichier permet d'appeler les fonctions du module comme celles du F1 , c'est à dire sans passer par l'ordre CALL (appel de sous-programme). Ainsi pour obtenir le sinus hyperbolique de 30° il suffit de faire $\text{SINH}(30)$ et non $\text{CALL SINH}(30)$ (ce qui serait équivalent à XROM SINH avec la 41). Ce procédé, s'appliquant à toutes les fonctions du module, est donc très souple.

Le module est livré avec un manuel de 178 pages format 180×215 (comme ceux du F1) mais orange et blanc et sans la photo du F1 en couverture. Un guide de poche de 33 pages au format et d'une présentation habituels est également fourni. Il résume bien les diverses fonctions et indique si elles sont ou non utilisables en mode CALC. (Je reviendrai plus tard sur ce point). Un récapitulatif des messages d'erreurs avec leurs numéros et leurs causes y est également inclus.

Le manuel présente (en anglais, avec une anglophobie !) les diverses fonctions par familles avec des exemples à la fin de chaque chapitre.

Les divers chapitres présentés sont :

- Changements de base (2, 8, 10 et 16 sont)
- Définitions de variables complexes
- Fonctions scalaires (hyperboliques et inverses, fonction gamma, log de base 2, élévation à une puissance de 10, type de variable ...)
- Fonctions complexes (toutes et même davantage)
- Entrées-sorties de matrice et initialisation.
- Opérations sur les matrices
- Déterminant, inversion, Produit scalaire, Normes.
- Transposition, Résolution de système linéaire.
- Recherche des racines d'une équation $f(x)$
- Intégration numérique (méthode de Romberg)
- Recherche des zéros d'un polynôme (réel ou complexe).
- Transformation de Fourier.
- Appendices dont celui donnant les valeurs par défaut retournées par les exceptions mathématiques.



Bref, il y a profusion, et ne manque guère que les équations différentielles.

Seules les fonctions scalaires et complexes sont accessibles en mode calcul. Il n'est pas possible d'y traiter des matrices ou des choses complexes comme l'intégration.

Les bons points :

- Bonne documentation
- Toutes les fonctions accessibles directement
- Fonctions presque toutes très rapides

- intégrales multiples (style $\int_0^x \int_1^3 \cos y \, dy \, dx$) jusqu'à cinq niveaux.
- Traitement matriciel très performant.

Les mauvais points:

- Mode CALC un peu délaissé.
- Pas de matrices rectangulaires pour INV et DET.
- Pas de bornes infinies pour les intégrales.
- Seules bases disponibles : 2, 8, 10 et 16.
- Quelques fonctions "lentes".

Quelques temps d'exécution:

Pour une matrice réelle 20×20 (400 éléments!)

$$\text{MAT A} = \text{identité} : 0,03 \text{ s}$$

$$\text{MAT A} = -A : 0,35 \text{ s}$$

$$\text{MAT A} = 7 * A : 2,21 \text{ s}$$

$$\text{MAT A} = \text{INV A} : 34,58 \text{ s}$$

$$U = \text{DET}(A) : 19,54 \text{ s}$$

$$\text{MAT A} = B - A : 1,61 \text{ s}$$

$$\text{MAT A} = B * A : 35,71 \text{ s}$$

$$\text{GAMMA}(220, 3514) : 0,11 \text{ s}$$

$$\text{LOG2}(32, 12) : 0,02 \text{ s}$$

Solution de $x-1=0$ avec 125 comme

valeur initiale : 0,44 s !!

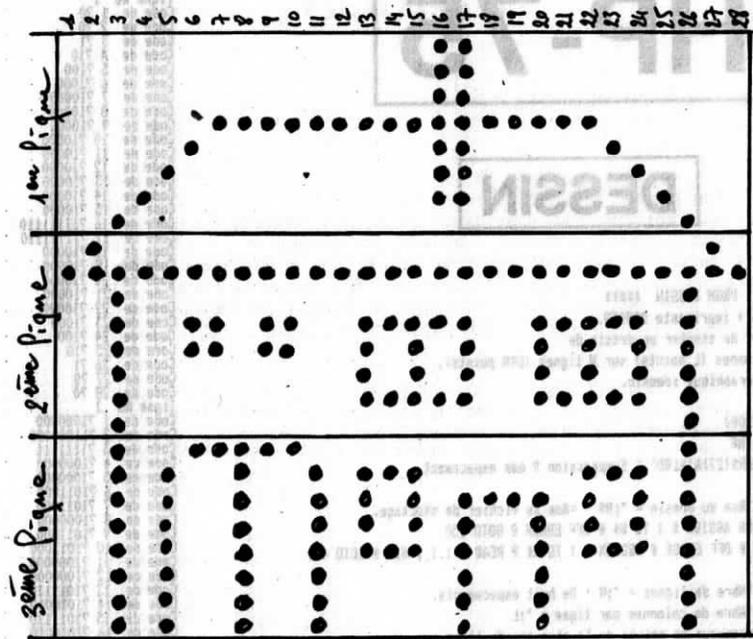
Solution de $\tan(\theta) - \theta = 0$ avec 0,35 comme

valeur initiale : 3,51 s

$$\int_0^{2\pi} \sin(\theta) \, d\theta \text{ avec 12 chiffres} : 0,28 \text{ s} !!$$

Prix du module : 362,72 F HT (qd même)

ou prix tarif HT.



Nom du dessin = Ferme
 Nombre de lignes = 3
 Nombre de colonnes par ligne = 28

Ligne N° 1 code de

1	=	0
2	=	0
3	=	1
4	=	10
5	=	100
6	=	1000
7	=	10000
...		
1	=	1000000
2	=	11000000
3	=	11111111
...		



N° 2
 Michel Pautus
 18 Rue Duperré
 75009 PARIS

TRUCS

Les cartes du HP-75 (comme celles du HP-71) ont un format peu commode car ne rentrant pas dans une enveloppe standard. Un de nos adhérents (Vincent Delorme) a trouvé une excellente solution : il faut mettre la (ou les) carte(s) dans une baguette de reliure en plastique. Vous connaissez certainement ces reliures en forme de pince (qui pincent bien mal). Fermées sur le côté et aux deux bouts avec du scotch, elles forment un étui peu couteux, solide et léger à nos cartes magnétiques.

Le club s'est procuré un module Text Formatter HP-75. Comme j'espère disposer bientôt d'une imprimante meilleure que notre actuelle Seikosha, je vous recommande de nous envoyer vos textes sur cassette et de me laisser le soin de les mettre en page. Si vous utilisez vous même le TF (et son clavier personnalisé) pour l'entrée du texte, ce n'est que mieux.

Ci-dessous une liste de fonctions du module APRIL (I/O) du 75 qui ne figurent pas dans le manuel ! Attention, le ALL RESET est au bout des erreurs, mais quelle puissance !

ACCESS\$("nom fichier") visualise les bits d'accès d'un fichier. La chaîne retournée indique l'état des bits d'accès sous la forme "RXELPCNT" où :

R= fichier RAM	r= fichier ROM
X=eXécutable (BASIC)	x=non exécutable
E=Editable	e=non éditable
L=Listable	l=non listable
P=Purgeable	p=non purgeable
C=Copiable	c=non copiable
N=avec Numéros de ligne	n=sans...
T=Token (fichier LEX)	t=non...

CHAIN "nom fichier:nom appareil" continue l'exécution d'un programme avec le fichier spécifié. Ceci purge le fichier courant, et copie le nouveau fichier de la mémoire de masse (purge seulement si le nouveau fichier existe).

CLEAR COMMON : efface la mémoire commune. Le prochain COMMON sera considéré comme une déclaration.

CLEAR SRQ : met à zéro la demande de service de tous les appareils (HP-IL). Travaille en demandant à chaque appareil son état. Peut ne pas marcher.

COMMON var, var... déclare ou accède aux variables communes, avec var=nom de la variable qui doit être commune, qu'elle soit numérique (A), chaîne (A\$), matrice (A() ou A(,)) et matrice de chaîne (A\$() ou A\$(,)). Le premier COMMON est une déclaration et tous les suivants doivent correspondre en type, nombre et taille.

CREATE\$("nom fichier", "type", "accès", taille) crée un fichier où nom fichier est le nom, type est le type tel qu'il apparaît au catalogue (B

pour Basic, etc.), accès est une chaîne règlementant l'accès (cf. ACCESS\$), taille est le nombre d'octets à allouer. L'adresse de début du fichier est retournée en hexa. Tenter d'éditer un fichier avant d'y avoir placé des données risque fort de bloquer la machine. Il est également dangereux de placer un T avant que le fichier LEX soit complet.

LINPUT\$ ("question") place une chaîne à l'affichage, vous permet de l'éditer et retourne le résultat. Taille maximale 93 caractères

NOISE matrice formelle, génère un son complexe. matrice formelle est le nom d'une matrice à une ou deux colonnes, qui contient les paramètres. Le premier élément est un drapeau désignant le type de paramètres : 0=fréquence en hertz, durée en secondes ; non zéro=durée en demi-périodes, durée en 2ⁿ-14 secondes. Le deuxième élément est le nombre de répétitions du son. Le reste de la matrice consiste en nombres pairs décrivant la fréquence et la durée. La matrice doit contenir au moins 4 éléments et avoir un nombre pair d'éléments. Sinon une erreur "MISSING PARAMETER" sera générée. L'option comptage de période permet des sons beaucoup plus rapides. Le rapport avec la fréquence est :

nombre de périodes=(28300 fréquence)-16

SET ACCESS "nom fichier", "accès" : place les bits d'accès au fichier donné.

TONE (fréquence) : exécute une tonalité de la fréquence donnée tant qu'une touche est pressée, retourne le temps d'exécution en secondes (précision au millième).

TYPE\$ (numéro de fichier) retourne le type de données actuellement pointé dans le fichier. Fin de fichier="EOF", chaîne="STR", nombre="NUM".

UPDATE "nom fichier" met à l'instant présent la date de création du fichier nommé.

Données fournies par l'auteur du ROM, Ryan Young (dit Raan, pour Rooman, de Kangaroo...). Essayez aussi DO ERROR 124 ou DO ERROR 123 (moins drôle) Zorro est arrivé ! Essayez LINPUT\$("Zarqoon.")

Voici deux petits programmes qui permettent de sécuriser en place vos propres programmes, à la manière du HP-71: vous ne pouvez plus purger ou éditer les programmes, mais vous pouvez les copier sur cassette ou carte. La protection ne vaut malheureusement qu'en RAM, elle n'est pas effective sur la cassette. Elle n'est pas non plus efficace en cas de ALL RESET.

```
10 A$=CAT$(0) @ A$=A$[1,
81
20 SET ACCESS A$,MAP$( 'P
E', 'pe', ACCESS$(A$) )
```

```
10 INPUT "Nom fichier ?
";A$
20 SET ACCESS A$,MAP$( 'P
E', 'pe', ACCESS$(A$) )
```

```
10 INPUT "Nom fichier ?
";A$
20 SET ACCESS A$,MAP$( 'P
e', 'PE', ACCESS$(A$) )
```

Sec(1)

Sec(2)

UNSEC

La deuxième version de SEC sécurise le fichier courant. Il n'est pas possible d'en faire autant pour UNSEC car on ne peut pas déclarer fichier courant les fichiers sécurisés.

Certaines lignes de MAKEFILE étaient difficiles à lire dans le précédent numéro (pas tant que ça, en fait). Voici donc une nouvelle version de ce programme, optimisé à l'aide du ROM I/O.

```

10 T#='M1' @ DIM A#C256J,B#C256J
20 T1#='TALWB' @ I#='RS'&CHR#(137)&CHR#(138)&CHR#(254)
30 INPUT 'Load file ? ' ; F1# @ IF F1#='' THEN CAT ALL @ GOTO 30
40 F#='FILL#(' ,F1#,' ',8) @ ON ERROR GOTO 60
50 CAT F1#&T# @ BEEP @ DISP 'File already on tape' @ STOP
60 OFF ERROR @ DISP 'Checking checksums'
70 ASSIGN # 1 TO F# @ READ # 1,1 ; A# @ B=0
80 A#='RTRIM#(' ,A#)
90 D=FND @ IF LEN(A#) THEN B=B+1 @ GOTO 90 ELSE W=B @ READ # 1,1 ; A#
100 X=FND+FND @ L=18+FND+256*FND @ X=FND @ T=POS(T1#,CHR#(FND))
110 IF NOT T THEN BEEP @ DISP 'Wrong file type' @ STOP ELSE C2=0 @ I=0
    @ B=1
120 I=I+1 @ READ # 1,I ; A#
130 A#='LTRIM#(' ,A#)
140 IF LEN(A#)<>4 THEN C1=0 @ GOTO 160
150 IF C2=256*FND+FND THEN 200 ELSE BEEP @ DISP 'Bad overall checksum'
    @ EDIT F# @ STOP
160 FOR B=B TO MINKL,B+W-1) @ B1=FND @ C1=MOD(C1+B1,255) @ C2=MOD(C2+B
    1,65535) @ NEXT B
170 IF C1=FND THEN 120 ELSE INPUT 'Checksums in file ? ' ,Y# ; A1#
180 A1#='UPRC#(A1#) @ N=SGN(POS(A1#,'N')) @ IF N THEN W=W+1 @ I=I+CEIL(
    L/W) @ GOTO 200
190 BEEP @ DISP 'Bad checksum on line' ; I @ EDIT F# @ STOP
200 DISP 'Creating tape file'
210 L1=I-1 @ COPY F1# TO T# @ PURGE F1#&T# @ SENDIO T#,'UNL,LAD#,DDL4'
    ,CHR#(0)&CHR#(0)
220 A#='ENTIO#(T#,'TAD#,DDT2,SDA') @ D=HTD(HEX#(A#C19,20J))
230 R1=HTD(HEX#(A#C11,12J))
240 SENDIO T#,'LAD#,DDL4',A#C11,12J @ A#='ENTIO#(T#,'TAD#,TL+,DDT2,SDA'
    ) @ I1=1
250 IF A#C11,I1+7J]=F# THEN 300
260 IF A#C11+10,I1+11J]=CHR#(255)&CHR#(255) THEN 290
270 IF I1<224 THEN I1=I1+32 @ GOTO 250
280 IF D>1 THEN D=D-1 @ I,I1=1 @ R1=R1+1 @ A#='ENTIO#(T#,'TAD#,SDA') @
    GOTO 250
290 BEEP @ DISP 'File not found' @ STOP
300 DISP 'Writing data to tape'
310 SENDIO T#,'UNL,UNT,LAD#,DDL4',A#C11+14,I1+15J @ SENDIO T#,'LAD#,DD
    L2,TL+',''
320 B,J=1 @ FOR I=1 TO L1 @ READ # 1,I ; A# @ FOR B=B TO MINKL,B+W-1)
330 B#C1J]=CHR#(FND) @ J=J+1 @ IF J=257 THEN J=1 @ SENDIO T#,'TL+',B#
340 NEXT B @ NEXT I @ IF J>1 THEN SENDIO T#,'TL+',B#C1,J-1J
350 SENDIO T#,'UNT,LAD#,DDL8,DDL4',CHR#(R1\256)&CHR#(MOD(R1,256))
360 A#='ENTIO#(T#,'TAD#,DDT2,SDA')
370 A#C11+19,I1+19J]=CHR#(CEIL(L/256)) @ A#C11+10,I1+11J]=CHR#(224)&I#C
    T,J
380 SENDIO T#,'LAD#,DDL4',CHR#(R1\256)&CHR#(MOD(R1,256))
390 SENDIO T#,'LAD#,DDL2',A# @ SENDIO T#,'LAD#,DDL8,DDL7','' @ CAT F1#
    &T# @ BEEP @ STOP
400 DEF FND
410 A#='LTRIM#(' ,A#)
420 FND=HTD(A#C1,2J) @ A#='A#C3J @ END DEF

```

HELP

*Sauvick Tailandier
Jean Belzonne
Stroobelli*

Fichier LEX HELP (source dans PPC Computer Journal V3N2P6)

Ce fichier permet d'obtenir des informations sur les ordres ou fonctions BASIC du HP-75.

HELP ['mot-clé']

Le mot-clé est optionnel, s'il est précisé les informations ne seront obtenues que sur ce mot, sinon tous les fichiers LEX en RAM puis les ROM externes puis les ROM internes sont analysés.

On passe dans ce cas d'un fichier au suivant en appuyant sur ATTN en les ordres non encore listés dans le fichier courant.

Sauvick

On obtient le listage sous le format suivant:

nnnnn V: type: mot-clé

où:

nnnnn représente le numéro de LEX-ID du fichier (c'est la même chose que le premier chiffre d'un XROM de 41 bits).

V: donne une lettre en principe de A à Z donnant la version du fichier LEX (ceci est facultatif).

type les valeurs obtenues sont:

STMT

stmt quelles différences ?

FNC# fonction donnant une valeur numérique

FNC# fonction donnant une chaîne de caractères

OTHR 'other' par exemple CMD de la commande SEND

OP#2 opérateur alpha

OP#2 opérateur numérique * / % + - , par exemple.

le mot-clé, s'il désigne une fonction, est suivi d'une représentation symbolique des paramètres, par exemple:

MAX(#,#) 2 paramètres numériques

UPRC\$(#) 1 paramètre alphabétique.

Pour résumer, c'est remarquable. Sous toutes réserves ceci serait l'oeuvre de D. Conklin.

2 & >help

101	STMT: HELP	16483	a	STMT: LOCAL	
16483	a	STMT: GOTOX	16483	a	STMT: REMOTE
16483	a	STMT: GOSUBX	16483	a	STMT: TRIGGER
16483	a	STMT: SHELL	16483	a	STMT: OUTPUT
16483	a	STMT: SENDIO	16483	a	STMT: KEYBOARD IS
16483	a	STMT: AUTOLOOP	16483	a	FNC#: IOSIZE?
16483	a	STMT: ASSIGN LOOP	16483	a	STMT: IOSIZE
16483	a	STMT: WIO	16483	a	STMT: ENTER
16483	a	STMT: LOCAL LOCKOUT	16483	a	STMT: TIMEOUT

```

0 3 FNC#: INF 0 3 OP#2: \
0 3 FNC#: PI 0 3 FNC#: POS(.,.)
0 3 FNC#: UPRC( ) 0 3 FNC#: DEG( )
0 3 OTHR: USING 0 3 FNC#: RAD( )
0 3 OTHR: THEN 0 3 FNC#: FLOOR( )
0 3 OTHR: TAB 1 " STMT: ALARM
0 3 OTHR: STEP 1 " STMT: LOCK
0 3 OP#2: EXOR 1 " STMT: OPTION ANGLE DEGREES
0 3 OP#1: NOT 1 " STMT: OPTION ANGLE RADIANS
0 3 OP#2: DIV 2 d STMT: TRANSFORM
0 3 FNC#: ERRN 2 d STMT: PACK
0 3 FNC#: ERRL 2 d STMT: INITIALIZE
0 3 OTHR: CARD 2 d FNC#: TIME#
0 3 OP#2: AND 2 d FNC#: DATE#
0 3 OTHR: KEYS 2 d FNC#: TIME
0 3 OTHR: ELSE 2 d FNC#: DATE
0 3 FNC#: SIN( ) 2 d FNC#: ANGLE(.,.)
0 3 FNC#: COS( ) 2 d FNC#: ACOS( )
0 3 FNC#: TAN( ) 2 d FNC#: ATN( )
0 3 OP#2: TO 2 d FNC#: ASIN( )
0 3 OTHR: I 2 d FNC#: RMD(.,.)
0 3 OTHR: J 2 d FNC#: MOD(.,.)
2 d FNC#: RND

```

(traits)



HELP

524 Bytes

Line	Data	Check
1	13 97 FA 01 8D 4C 5E 4D 24 9F 48 45 4C 50 20 20 20 20 65 00 00	
2	0A 00 12 00 0C 00 17 00 1A 00 2D 00 18 00 FF FF 48 45 4C 00 4C	
3	FF 96 80 1D 9E CE 6D 46 50 12 6C A9 B4 65 00 00 6F 0A E2 6C 80	
4	E5 9E A1 98 74 93 CE 3A 1F F9 0C CE CC FD CE 11 1F 76 0A E3 F1	
5	5C 34 A3 56 B1 A3 82 78 16 85 53 00 7E A9 60 00 C3 78 87 53 CF	
6	00 CE 48 48 EE AE 40 A9 65 00 CE 61 47 9E 74 06 E5 56 E5 5E 5D	
7	B9 A3 82 CE 8A 01 58 A9 64 84 68 B1 FB 09 18 A7 CE 5A F3 68 89	
8	B1 64 84 4C 06 E3 74 E3 98 50 B1 A3 82 18 A3 12 A3 10 85 04 25	
9	00 D3 A5 82 58 18 85 02 00 D3 A5 82 52 12 85 0A 00 D3 A5 82 40	
10	88 8B 40 A8 30 40 8A 41 C8 20 F7 F9 40 88 88 41 12 H4 42 89 5C	
11	A3 82 F6 04 58 CB 48 00 42 18 E1 56 A9 74 84 CE 48 08 F9 07 E5	
12	CE A3 07 56 0C A1 9E 42 10 E0 C8 80 F7 06 C8 FF F7 F1 F0 07 41	
13	18 E1 F0 EF 42 10 E0 43 02 A0 CF 7F 16 E4 02 C0 F7 F2 74 91 F1	
14	F7 1E 7C 3A A1 40 16 A1 CD 74 84 3E C1 F6 B0 A9 74 84 42 00 C0	
15	E0 43 3C E0 02 C0 F6 B0 7E 8B F6 F2 7E 18 A5 D3 A5 82 78 3E 8E	
16	B5 F8 FF 43 3F A0 CF C0 F7 27 C8 40 F6 08 64 A9 43 4D 4E 44 BA	
17	F0 12 C8 00 F6 00 6A A9 53 54 4D 54 F0 06 64 A9 73 74 6D 74 71	
18	42 A9 CF 01 0C C3 06 E5 9E 7F CF 3F C8 28 FA 3D C8 2C FB 20 DF	
19	64 A9 4F 50 20 20 7F C8 2A 66 FA 04 A8 24 F0 02 A8 23 F8 90 61	
20	F3 05 67 A8 32 F0 03 67 A8 31 F0 17 F7 17 C8 2F FB 13 64 A9 9C	
21	46 4E 43 20 7F 90 67 F3 04 A8 23 F0 02 A8 24 F0 06 64 A9 4F 47	
22	54 48 52 64 C8 46 F6 35 99 78 81 81 81 98 F9 2D 42 A8 28 16 0E	
23	E4 7F 90 F5 05 42 A8 24 F0 03 42 A8 23 16 E4 78 85 7F 90 F5 00	
24	05 42 A9 28 29 E5 78 85 42 A8 2C E4 9B F8 DE 56 88 42 A8 29 8B	
25	E4 64 B3 6E 84 66 A9 3A 20 B3 72 84 68 B3 65 84 67 B2 6D 84 17	
26	56 CD 65 84 B2 64 84 A9 6A 84 CE E3 07 42 18 E1 A9 C1 00 0C A9	
27	C3 06 E5 9E	4E
28	F3D5	

FRANC: 0	0	FRANC: 3	3
FRANC: 1	1	FRANC: 4	4
FRANC: 2	2	FRANC: 5	5
FRANC: 3	3	FRANC: 6	6
FRANC: 4	4	FRANC: 7	7
FRANC: 5	5	FRANC: 8	8
FRANC: 6	6	FRANC: 9	9
FRANC: 7	7	FRANC: 10	10
FRANC: 8	8	FRANC: 11	11
FRANC: 9	9	FRANC: 12	12
FRANC: 10	10	FRANC: 13	13
FRANC: 11	11	FRANC: 14	14
FRANC: 12	12	FRANC: 15	15
FRANC: 13	13	FRANC: 16	16
FRANC: 14	14	FRANC: 17	17
FRANC: 15	15	FRANC: 18	18
FRANC: 16	16	FRANC: 19	19
FRANC: 17	17	FRANC: 20	20
FRANC: 18	18	FRANC: 21	21
FRANC: 19	19	FRANC: 22	22
FRANC: 20	20	FRANC: 23	23
FRANC: 21	21	FRANC: 24	24
FRANC: 22	22	FRANC: 25	25
FRANC: 23	23	FRANC: 26	26
FRANC: 24	24	FRANC: 27	27
FRANC: 25	25	FRANC: 28	28
FRANC: 26	26	FRANC: 29	29
FRANC: 27	27	FRANC: 30	30
FRANC: 28	28	FRANC: 31	31
FRANC: 29	29	FRANC: 32	32
FRANC: 30	30	FRANC: 33	33
FRANC: 31	31	FRANC: 34	34
FRANC: 32	32	FRANC: 35	35
FRANC: 33	33	FRANC: 36	36
FRANC: 34	34	FRANC: 37	37
FRANC: 35	35	FRANC: 38	38
FRANC: 36	36	FRANC: 39	39
FRANC: 37	37	FRANC: 40	40
FRANC: 38	38	FRANC: 41	41
FRANC: 39	39	FRANC: 42	42
FRANC: 40	40	FRANC: 43	43
FRANC: 41	41	FRANC: 44	44
FRANC: 42	42	FRANC: 45	45
FRANC: 43	43	FRANC: 46	46
FRANC: 44	44	FRANC: 47	47
FRANC: 45	45	FRANC: 48	48
FRANC: 46	46	FRANC: 49	49
FRANC: 47	47	FRANC: 50	50
FRANC: 48	48	FRANC: 51	51
FRANC: 49	49	FRANC: 52	52
FRANC: 50	50	FRANC: 53	53
FRANC: 51	51	FRANC: 54	54
FRANC: 52	52	FRANC: 55	55
FRANC: 53	53	FRANC: 56	56
FRANC: 54	54	FRANC: 57	57
FRANC: 55	55	FRANC: 58	58
FRANC: 56	56	FRANC: 59	59
FRANC: 57	57	FRANC: 60	60
FRANC: 58	58	FRANC: 61	61
FRANC: 59	59	FRANC: 62	62
FRANC: 60	60	FRANC: 63	63
FRANC: 61	61	FRANC: 64	64
FRANC: 62	62	FRANC: 65	65
FRANC: 63	63	FRANC: 66	66
FRANC: 64	64	FRANC: 67	67
FRANC: 65	65	FRANC: 68	68
FRANC: 66	66	FRANC: 69	69
FRANC: 67	67	FRANC: 70	70
FRANC: 68	68	FRANC: 71	71
FRANC: 69	69	FRANC: 72	72
FRANC: 70	70	FRANC: 73	73
FRANC: 71	71	FRANC: 74	74
FRANC: 72	72	FRANC: 75	75
FRANC: 73	73	FRANC: 76	76
FRANC: 74	74	FRANC: 77	77
FRANC: 75	75	FRANC: 78	78
FRANC: 76	76	FRANC: 79	79
FRANC: 77	77	FRANC: 80	80
FRANC: 78	78	FRANC: 81	81
FRANC: 79	79	FRANC: 82	82
FRANC: 80	80	FRANC: 83	83
FRANC: 81	81	FRANC: 84	84
FRANC: 82	82	FRANC: 85	85
FRANC: 83	83	FRANC: 86	86
FRANC: 84	84	FRANC: 87	87
FRANC: 85	85	FRANC: 88	88
FRANC: 86	86	FRANC: 89	89
FRANC: 87	87	FRANC: 90	90
FRANC: 88	88	FRANC: 91	91
FRANC: 89	89	FRANC: 92	92
FRANC: 90	90	FRANC: 93	93
FRANC: 91	91	FRANC: 94	94
FRANC: 92	92	FRANC: 95	95
FRANC: 93	93	FRANC: 96	96
FRANC: 94	94	FRANC: 97	97
FRANC: 95	95	FRANC: 98	98
FRANC: 96	96	FRANC: 99	99
FRANC: 97	97	FRANC: 100	100

PPC-T

MAGAZINE

LE COIN DU CLUB

L'ORGANISATION DE PPC-TOULOUSE

Cette page sera publiée dans chaque numéro de MICRO-REVUE. Seules les indications répertoriées dans le dernier numéro du journal font foi. PPC-T étant une association sans but lucratif, elle ne fait pas de bénéfice, elle est administrée par des bénévoles et ses produits et services sont réservés à ses membres. La marge prise sur les produits vendus est minimale et destinée à couvrir les frais de fonctionnement. Une grande partie de nos produits vient des USA et leur prix est donc soumis aux variations du \$. En cas de variation brusque du \$ nous nous réservons le droit de demander un supplément.

ASSOCIATION PPC-T

Association à buts non lucratifs (loi de 1901), PPC-T accepte des adhérents dans toute la France et à l'étranger. La cotisation est de 50 F plus 150 F d'abonnement groupé pour MICRO-REVUE, pour un total de 200 F. DOM TOM et reste du monde par avion ajouter 50F. Règlements par chèques et mandats postaux à l'ordre de PPC-T (Eurochèques acceptés, mandats postaux uniquement pour les DOM TOM). L'inscription prend effet au premier jour du bimestre qui précède la réception au club du bulletin d'adhésion (1er Janvier, 1er Mars...). En cas d'adhésion d'un abonné, l'adhésion est de 50 F et prends effet à la date de début de l'abonnement. Les adhérents s'engagent à apporter leur contribution au journal et au club dans la mesure de leurs moyens, même modestement.

RESPONSABLES DE PPC-T

Président: Jean-Daniel Dodin (T1) 77 rue du Cagire 31100 Toulouse (nouvelles adhésions, journaux de l'année en cours, édition du journal, réception des articles).

Trésorier: Jean-François Sibille (T15) Résidence du Pays d'Oc, rue des Genets 31500 Toulouse (renouvellement d'adhésions, problèmes financiers...).

Trésorier adjoint: Jean-Pierre Baudoin (T131) Lieu dit "Les Carabiniers" Castelnaud d'Estretetfond 31620 Fronton (fichier).

Secrétaire: Olivier Monachon (T13) 19 rue de Coulmier 31000 Toulouse

Secrétaire adjoint: Gilles Barret (T22) 92 av de Muret 31300 Toulouse (photocopies).

Secrétaire adjoint: Jean-Yves Pasquier (T18) Apt. 10, 2 rue de Grèce 31000 Toulouse (librairie et coopérative).

Programmathèque: Damien Debrill (T33) 38 rue du 8 Mai 1945 59190 Hazebrouk (envoyez les cartes vierges, elles reviennent pleines).

Coordinateurs régionaux:

Bretagne: Franck Lebastard (T35) 10 rue de la Jalousie 35600 Redon

Paris: Olivier Arbey (T164) 34 av de la République 94100 St Maur, Robert Schwartz (T178) 141 av Carnot 93140 Bondy, Eric Gengoux 8 rue de Furstenberg 75006 Paris, J.J. Dhenin 88 rue de Gergovie 75014 Paris, Jean Thiberge (T165) 104 Quai Louis Blériot 75016 Paris, Serge Vaudenay (T270) 62 av Ardouin B3 94420 Le Plessis Tréville, Stéphane Barizien (T149) 39 rue St Fargeau 75020 Paris

Lorraine: Dominique Talon (T311) 90100 Courcelle

Sud Est: Eric Angelini (T102) 14 av Marianna 06000 Nice

Besançon: E. Piérunek (T76) chez Mme Faivre 27 rue O. David 25000 Besançon.

Picardie: Bruno Tredez (T120), 16 chemin St Jean Allonville 80260 Villers Bocage.

Région Rhône: Yves Alajouanine (T144), 108 bis rue Hénon 69004 Lyon.

BELGIQUE: Bernard Branquart (T372) 32 Thiarfont, 7190 Ecaussines Belgique.

Des coordinateurs sont recherchés pour toutes les autres régions. Contactez le coordinateur de votre région pour organiser avec lui des réunions, démonstrations, ... Vous pouvez obtenir un budget du club sur présentation d'un projet. Nous pouvons vous imprimer vos convocations, des affiches, ... Contactez vous-même les concessionnaires locaux, les journaux, les autres clubs. Contactez les maires pour avoir des salles de réunions (montrez leur cette page du journal pour confirmer votre position, vous nous représentez officiellement dans votre région !)

LES JOURNAUX

Reçus depuis MICRO-REVUE 1 :

Le nouveau catalogue d'EDUCALC (cf. "sur le Marché") 27953 Cabot Road, Laguna Niguel, CA 92677.

PPC Calculator Journal Mai, Juin et Juillet 84. Me paraît de plus en plus intéressant. Computer Journal of PPC; plein de choses très intéressantes sur le HP-75 et le HP-71. Ces deux journaux vont sans doute fusionner en Janvier 85. En attendant le montant des abonnements/adhésion sont de \$37 pour chacun des journaux, plus \$8 de droit d'entrée. PPC POB 9599 Fountain valley, California 92728-9599 USA.

Le troisième numéro du journal Danois, toujours aussi beau, 71 pages. PPC Danmark Postboks 2, 3500 Vagløse, Danemark.

Hewlett-Packard Journal, deux numéros, le Juillet 84 consacré entièrement au HP-71 (très intéressant, en anglais), et le Aout 84 entièrement consacré au HP-150. Disponible gratuitement chez HP-France (et non pas chez les revendeurs).

Les numéros de Juin et Juillet de PRISMA, couverture couleur, toujours aussi beau.

Le deuxième numéro de JEDI (cf. dossier FORTH).

Les numéros de Mai, Juin et Juillet de JPC, journal du chapitre de Paris, avec couverture cartonnée rouge pour le numéro de Juillet (en fait Juillet-Aout). Annoncent une présentation reliée pour le prochain numéro. Philippe Guez 56 rue JJ Rousseau 75001 Paris. Ils n'ont toujours pas publié notre petit encart publicitaire, ni leur fameux "dossier" sur leurs relations avec nous qu'ils annoncent avec des sous entendus désagréables depuis quelque temps. Des gamineries.

PETITES ANNONCES

Gilles Barret (T22, service photocopies) vend 1 module HP-IL (600F) + un livret d'application Fluid Dynamics and Hydraulics avec codes barre (70F) (pour HP-41).

Edouard Cohen (T102), 11 rue Wagner 93150 Blanc-Mesnil vend Lecteur de cartes Avril 83 (1100F), module Mathe (200F), Calculator tips and routines (100F), PPC Journal (US) années 81,82,83 (250F/an).

Laurent DANE
52 Le Hameau d'EPson
Bouliac
33 270 FLOIRAC
Tél: (56) 20 53 19

Cette lettre a été écrite Par une imprimante a aiguilles SEIKOSHA GP 100 A adaptée Pour la boucle HP-IL Par la SICAPE. La Lettre est mise en fichier ASCII sur HP 41 et un Programme Permet de la Personnaliser.

Si les Paperasses Presques identiques vous énervent et vous font perdre du temps, cette imprimante Pourra vous aider Pour un Prix Pas trop élevé. La SICAPE Produit d'ailleurs un module en microcode de traitement de texte Pour HP 41.

Désirant Passer à un niveau d'ordinateur supérieur à celui de la HP 41 cette imprimante ne sera Plus tres adaptée à mes besoins. J'ai donc décidé de la vendre au Prix de 3500 F .Elle a été achetée à La FNAC en Janvier 84 et a tres Peu servi.

Si vous Possédez une imprimante thermique {HP 82143A} non HP IL un échange avec contrepartie serait Possible Puisque Je souhaite Garder ma HP 41. D'autre Part Je vends également mon module HP IL 500F, il a été acheté en novembre 83.

En espérant que cette offre Pourra vous intéresser, je vous souhaite de bons Programmes.

2-0-2

SOTIROPOULOS T 434

13, av. A. ARCHAMBAULT 95110 SANNOIS

VENDS:

- HP 41CV(1982) REVISEE
- PRINTER 82143A
- WAND
- CARD READER
- X FUNCTIONS
- X MEMORY
- TIME
- HP IL (sous garantie)
- MODULE NAVIGATION
- PPC ROM (sans manuel)
- LIURES
- ACCUS
- CLAVIER SOUPLE
- MANUELS

L'ensemble en parfait état , 8000 FRs.

VENDS EGALEMENT:

- TI 58C 350 FRs
- CASIO FX 702P+PRINTER+INTERF.CASS.900 F
- CANON X07+IMPRIMANTE 4 COULEURS X710
- +CHARGEURS+MANUELS 3000 FRs.(Sous GARANTIE)

T 123 - Gilbert Papin - 9 rue du Hameau de Bellevue, 91230 MONTGERON

Vends pour HP41 : Lecteur de code barre (400), modules Extended i/o (400), plotter (400), time (350), xfunction (350), finance et 4 modules simples (300), un quad (350), extension de port de la sicape (500). L'ensemble : 2.600 Envoi franco si chèque à la commande ou la moitié des frais en contre remboursement. SVP : me contacter de préférence par écrit.

T341 VDS HP41 1500 REGS + TIME 5000 FF à DISCUTER. LA MACHINE EST EN TRÈS BON ETAT
ET DATE D'AVRIL DERNIER. ME CONTACTER AU 700 54 24.

Eric Hansmege 105 rue de la convention 75015 Paris

Vds Imprimante HP B2143A pour HP-41C
Prix à débiter: 2000 FF
(Date d'achat: JANU 82)
S'adresser à Mr COSSEVIN ERIC
3, rue du cdt FILLOL 17000 La Rochelle
Tel: (46) 44.15.36

S-O-S

COIN DES U.PRO

Le 6 mai 1984

T400 Grégoire JOUVE 7, rue Adolphe Leray 35100 RENNES
(à partir d'août 84)

Je me propose pour coordonner les efforts des adhérents qui pratiquent des DISPOSITIFS D' EXPERIMENTATION en agriculture ou dans d'autres domaines pour établir un ensemble de programmes d'analyses statistiques à plusieurs variables -variances, covariances... pour des dispositifs expérimentaux comme Blocs de Fisher carrés latins, réseaux équilibrés carrés ou non, factoriels etc.. avec des sous-programmes pour les calculs des tables des distributions: F de Snedecor, Bartley, N. et K., Duncan etc.. Je suis persuadé que la plupart des programmes existent déjà et qu'ils pourraient se standardiser avec des sous-programmes communs pour trouver place dans les 30x80x140 mm de notre machine ou une EPROM si nous sommes une demi-douzaine.

Heureuses Programmations



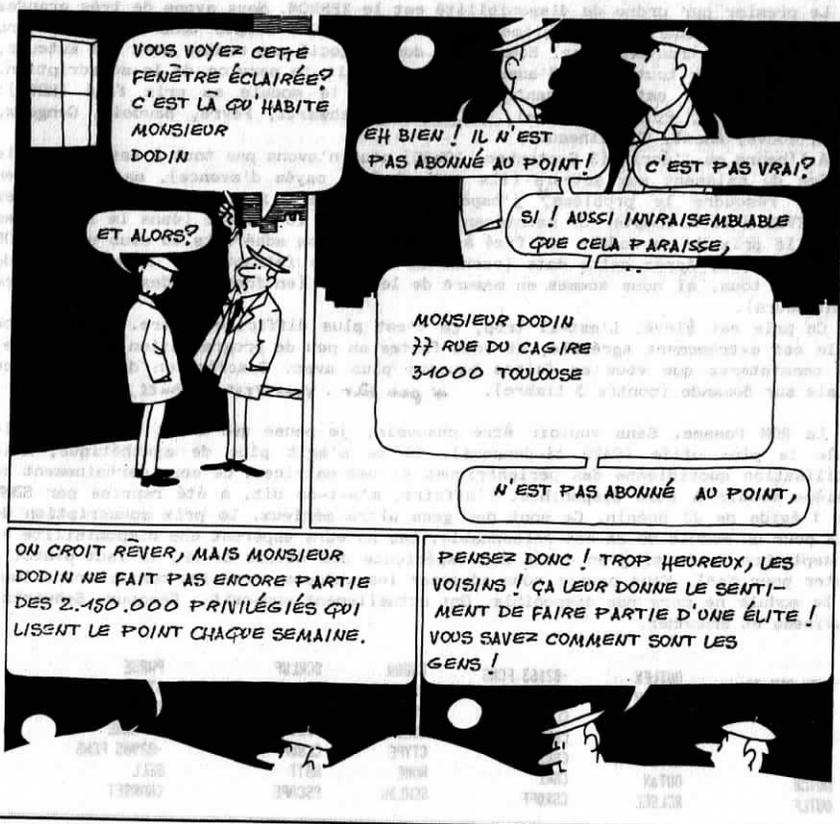
Janick Taillandier (9 rue Delezienne 59000 Lille) est un de nos principaux fournisseurs de documents d'origine américaine. La plupart de ces documents sont achetés à ses frais (et d'abord pour ses propres besoins, bien sûr). Mais ses moyens financiers ont des limites et actuellement il demande de l'aide pour l'acquisition des "IDS" du 71. Il s'agit des livres fournis par HP et contenant les spécifications du 71. Au total (6 volumes) il y en a pour plus de 5000F. Si on s'y met à 10, le prix de revient est raisonnable. Que tous ceux qui sont intéressés par un achat en commun de ces ouvrages se mettent en rapport avec Janick. Merci.

NOUVELLES BREVES

Hewlett-Packard diffuse en France un certain nombre de produits (modules...) avec des notices en anglais. En fait ceci est contraire à la loi française, mais le prix de revient d'une traduction suivie d'une édition à quelques centaines d'exemplaires est trop élevé, et c'est ça ou ne pas avoir le produit. Je crois savoir cependant que HP pourrait être intéressé par la diffusion par photocopie de traductions de ces notices. Si par hasard vous avez fait une telle traduction, prenez contact avec Mme Chantal Berthier à Evry. Une traduction "professionnelle" est facturée 80F les 250 mots du texte d'origine. Vous ne pourrez sans doute pas en obtenir autant, mais n'en faites quand même pas cadeau.

Les nouveaux manuels de la HP-41CX sont parus. Il y a trois très beaux manuels : principes de base (145p), fonctionnement en détails (452p), guide de poche (44p). Très bien faits. Ils vous seront envoyés **gratuitement** sur simple demande. Si vous pouvez joindre une facture d'achat de la 41-CX, ce serait quand même mieux. HP France : Avenue du Lac, 91040 Evry Cedex (16 (6) 077 83 83).

Les imprimantes font des progrès : j'ai reçu cette publicité du journal "Le Point" (je ne m'y suis pas abonné pour autant !):



ERREUR

Dans l'article "gestion de fichier", paru dans le numéro 1 de MICRO-REVUE, je me suis aperçu qu'une petite erreur s'était glissée dans le programme : ENR-FI, à la ligne 48 de celui-ci il faut lire RCL 06 et non RCL 04.

Marcel Trimborn (T86)
18 rue des Jardins
57990 Nousseviller St Nabor

SUR LE MARCHÉ

Il y a beaucoup d'opérations en cours avec des fortunes diverses, et il me paraît indispensable d'en faire le point.

LES ROMS HP-41 :

A ma connaissance, actuellement, il y a 4 modules clients HP en préparation, à des degrés divers.

Le premier par ordre de disponibilité est le ZENROM. Nous avons de très grandes difficultés à résoudre les problèmes posés par ce ROM. Compte tenu de son contenu effectif nous le trouvons cher. Nous avons donc négocié une remise avec les auteurs, qui nous permettra tout juste d'assurer sans déficit le service de la souscription. Cette souscription est maintenant close. Auront le module au prix fixé (800F) : Thiberge, Nicolas, Pelanne, Wrobel, Collewaert, Schwartz, Fèvre, Baudoin, Gengoux, Adrien, Jouve, Ancel, Chauvineau.

A l'heure où j'écris (3 Septembre, 10h30) nous n'avons pas tout à fait résolu le problème du paiement aux auteurs (ils veulent être payés d'avance), mais sommes en voie de résoudre le problème, j'espère que ce sera fait quand vous recevrez MICRO-REVUE N°2. A compter de maintenant, et jusqu'au 15 Octobre (dans la limite des stocks) le prix de ce module est fixé à 1000 F pour les adhérents du club et à 1200F pour les autres. Après cette date (cachet de la poste faisant foi) le prix sera de 1200F pour tous, si nous sommes en mesure de le tenir (en fonction des exigences des fournisseurs).

Ce prix est élevé. L'est-il trop, ça c'est plus difficile à dire. A l'usage ce module est extrêmement agréable, si vous faites un peu de programmation synthétique, vous constaterez que vous en faites beaucoup plus avec. Description de 4 pages en anglais sur demande (contre 3 timbre). ** pas sûr. Vous serez informés.*

La ROM Paname. Sans vouloir être chauvin, je pense que c'est de beaucoup le module le plus utile (CAT2 ci-dessous). Il ne s'agit plus de synthétique, mais d'utilisation quotidienne des périphériques et des matrices. Ce sera certainement le deuxième module à être disponible. L'affaire, m'a-t-on dit, a été reprise par SBMW sous l'égide de JJ Dhénin. Ce sont des gens ultra sérieux. Le prix souscription de 1100F pour un module de 8K est raisonnable. Les auteurs espèrent une disponibilité en fin septembre, mais si j'en crois mon expérience des délais de HP, il faut plutôt y compter pour Noël. Vous pouvez nous adresser les chèques, nous les conserverons tant que le module ne sera pas disponible. Ont actuellement souscrit : Gengoux, Schwartz, Chauvineau et Brandner.

-ADV PRT 3C	OUTLFX	-82163 FCMS	CSRON	SCRLUP	PARSE
ATD	OUTSPX	CLEAR	CSRR	SCRLX	STATUS
ID	OUTXB	CLEARO	CSRXX	XYTAB	TABCOL
FINDRID	OUTYBX	CSRDN	CSRUP	-82162 FCMS	UNPARSE
OUTAX	OUTa	CSRHX	CTYPE	CLBUF	-82905 FCMS
OUTCR	OUTaX	CSRL	HOME	8BIT	BELL
OUTLF	RCLSEL	CSROFF	SCRLDN	ESCAPE	CHARSET

FFEE	*DRAW	AD-LC	LC-AD	RCOPY	X=NH?
FORNLEN	*HONE	ALENG	LINPT	RGINIT	X=NH?
GRAPHX	*LABEL	ANUM	NOP	RCMB	X<NH?
MODE	*LDIR	ANUNDEL	OUT	RCORD	X<NH?
SKIPOFF	*LTYPE	APPX	POSA	RCXTR	X<NH?
SKIPON	*MOVE	AROT	PSIZE	RCSUM	X<NH?
TEXTLEN	*PLRECX	ATOKL	READEM	RCVIEW	X<NH?
YSPAC	RDRAN	ATOXR	RG	SAVERGX	X<F
-PLOT FCMS	RESET	ATOPX	RG+	SIZE?	XTOAL
AXIS	REVL	BLDPT	RG*	SORT	XTOAR
BACKSP	REVLFX	BRKPT	RG/	STO\L	Y/N
BACKSPX	RMOVE	CHFLAG	RG+Y	SUB\$	YTOAR
BOX	SETORG	CLINC	RG+Y	TF55	
COLOR	-UTILITIES	COLPT	RG/Y	VKEYS	
*Csize	/MOD	GETRCX	RCAX	WRTEH	

Le module CCD (club allemand). Ce module est présenté comme une pure merveille (8 ou 12K ?) et certainement ses programmeurs sont de première qualité. Il doit comprendre une bonne partie des commodités du ZENROM (double emploi préjudiciable, à mon avis), ainsi que des utilitaires et des fonctions mathématiques. Mais il n'est pas encore fini d'écrire, ce qui fait qu'à mon humble avis il ne sera pas disponible au mieux avant Janvier 85, c'est bien tard. Son prix sera de l'ordre de 1100F. Nous n'avons encore reçu directement aucune proposition (mais des articles ont été publiés dans les journaux allemand et américain).

Le module FORTH pour HP-41. C'est de loin le projet que je trouve le plus excitant. Il a été lancé par Serge Vaudenay qui s'est programmé son module FORTH de 4K tout seul comme un grand (il vient d'entrer en première, il promet !). Cette première version était semble-t-il très lente. Mais J.J. Dhénin et Stéphane Barizien, une fois fini le ROM Paname, s'y sont mis aussi. Connaissant leur niveau, il n'est pas étonnant que les résultats soient intéressants. Par contre il est presque impossible d'obtenir d'eux une description écrite du module. J'espère néanmoins pouvoir lancer dans le prochain numéro de MICRO-REVUE une souscription internationale pour ce module. Si nous pouvons trouver 150 clients à 1000F chacun, la réalisation sous forme de module client est faisable. Sinon il faudra se contenter d'une EPROM. Ecrivez-leur, écrivez-moi ce que vous en pensez (adresses à la coordination de Paris dans l'organigramme du club). *voir p 21*

Le PPC ROM. Celui-là est fabriqué depuis longtemps, mais il est incroyablement difficile de l'obtenir, le club ne fournissant jusqu'à présent aucune facture et demandant un paiement d'avance. Une commande de 10 PPCROMS a été payée à PPCUSA le 20 Juillet dernier, et nous n'avons encore rien reçu. Sont enregistrées les commandes de Ancel, Lecluse, Duquesne, Gamper, Albagnac, Schmoker, Desjonquère, Baconnier, soient 8 commandes. Il va rester 3 ROMS avant la prochaine commande. Le prix est toujours (sauf mauvaise surprise) de 1050F. *Problème résolu, la banque n'avait pas manqué le bon à commande avec le chèque.*

LES DISQUETTES :

5 ou 6 personnes ont manifesté leur intérêt pour des disquettes HP-11 format IBMPC. Compte tenu de cela les Editions du Cagire ont fait une demande de tarif au fabricant, sans réponse pour l'instant. Vous serez informé. Si vous voulez des détails, écrivez aux Editions, ceci n'est pas une opération du club.

Résultat obtenu - chez un 110V... suivre quand nous

LES DOCUMENTS FORTH

Nous avons eu quelques difficultés à obtenir les originaux des documents FORTH annoncés dans le dernier numéro, mais c'est maintenant chose faite sauf pour le 68000, du fait d'une erreur d'expédition. Ce dernier problème sera réglé rapidement. reportez vous au dernier numéro pour la liste des documents disponibles.

AVOIRS DE PPC-T

Quand vous envoyez un chèque pour une commande à PPC-T, il peut se faire que le montant du chèque soit trop élevé. Pour des montants faibles (disons moins de 30F), si la commande faite est livrée en totalité il n'est pas délivré d'avoir, et le solde

est considéré comme une subvention au club (merci). Si nous ne pouvons pas vous livrer un produit dans un délai raisonnable, nous vous adressons un avoir. Nous ne pouvons pas vous rembourser immédiatement car ceux qui vous expédient les commandes ne disposent pas de carnet de chèques. Les avoirs en question peuvent être utilisés pour régler toute autre commande à PPCT, ou bien sur être remboursés sur simple demande adressée au trésorier. Ce dernier point nous paraissait évident, mais certains se fâchent facilement. Bien sûr, il faut nous retourner l'avoir pour en bénéficier. Nous ne conservons pas de trace des avoirs. Sont valides les avoirs signés par le trésorier, le président ou les responsables de la coopérative ou du service photocopies, dûment signés et portant le tampon du club. Merci de votre compréhension.

NE SONT PLUS DISPONIBLES A TOULOUSE

Notre fournisseur de cassettes a augmenté ses prix de 50%. Nous ne pouvons donc plus vous fournir ces cassettes (nous avons du mal à nous en procurer pour nous à un prix décent).

La vente de papier noir pour imprimante se fait par à-coups. Après n'avoir rien vendu pendant plusieurs semaines, nous avons épuisé notre stock au mois de Juillet. Compte-tenu des congés et des délais de notre fournisseur, nous serons en rupture de stock au moins jusqu'en Octobre. Ne nous passez plus de commande avant le prochain numéro, merci.

Extended functions made easy en anglais n'est plus disponible. Synthetic programming made easy est encore disponible en anglais à quelques exemplaires.

NOUVEAUX PRODUITS

Deux nouveaux interfaces vidéo sont maintenant disponibles, l'un Danois, l'autre allemand. Pour un prix à peu près identique, le modèle allemand semble incomparablement plus performant, c'est donc le seul dont l'importation est actuellement en cours. Les Editions du Cagire vous proposent donc cet appareil au prix de 3700F TTC. Il s'agit d'un interface vidéo 80 colonnes/graphique/interface imprimante. Il contient deux appareils HP-IL, l'un est un interface vidéo 80 colonnes, l'autre une "table tracante" utilisant une version simplifiée du langage HP-GL, le tout sortant sur le même écran. L'appareil dispose en plus d'une prise imprimante parallèle qui permet outre le branchement d'une imprimante en parallèle sur l'écran une recopie d'écran graphique sur imprimante (!). Il est également possible de sortir le contenu de la mémoire graphique sur l'HP-IL. Le boîtier fait 22x14x5cm et pèse un peu moins d'un kilo. Il contient deux circuits imprimés de qualité professionnelle et une alimentation généreuse. L'ensemble donne confiance. Branché, l'appareil apparaît sous la forme de deux appareils HP-IL. Photocopie de la notice (actuellement en allemand) sur demande aux Editions du Cagire 77 rue du Cagire 31100 Toulouse (photocopie également de la notice de l'appareil danois sur demande) joignez 3 timbre poste. Si quelqu'un peut me faire une traduction en français pour un prix raisonnable, qu'il me contacte (il y a 18 petites pages). Délai de livraison de l'appareil : environ un mois.

Vincent Delorme est prêt à commencer en petite série la fabrication d'un interface HP-IL/RS232 qui aurait les caractéristiques suivantes :

- deux ports de sortie RS232 et 1 port parallèle
- programmation interne par microprocesseur et EPROM (donc ajustable à vos besoins),
- prix environ 2500F

Si vous êtes intéressés, vous pouvez nous écrire pour que nous puissions vous tenir au courant de l'avancement des travaux (j'attends un prototype pour fin Septembre).

DIVERS

Une bibliothèque de programmes utilitaires pour HP-71 est envoyée gratuitement par HP à ceux qui s'inscrivent à la librairie de Corvallis. Le temps que vous receviez ce journal, un de vos amis en aura sans doute une copie.

CHEZ EDUCALC

L'utilisation des cartes de crédit est maintenant à nouveau autorisée pour tous, vous pouvez donc commander à EDUCALC simplement en indiquant le numéro de votre carte et sa date de fin de validité (une photocopie de la carte fait l'affaire) n'oubliez pas qu'il faut compter \$6 par commande + \$15 par kg de frais d'envois. Le montant sera directement calculé et prélevé par Educalt, vous pouvez leur faire toute confiance. N'oubliez pas des frais de douane et de TVA de 35% environ à payer au facteur à la réception. Si la commande dépasse 5000F il faut prendre contact avec un transitaire et prévoir des frais supplémentaires.

Pieds d'expansion/modem pour HP-75C, avec disque électronique et toutes les commandes pour le modem (full duplex) et le disque, 32K : \$650.95, 64K : \$895.95

Gestion de fichier pour HP-75, programme de base de données, VISICALC indispensable, ref. C-312, \$39.95

Thinkjet printer HP-IL : \$410 (avec port)

Disquette HP-IL HP9114 (double face simple drive, 630K, portable) : \$650

LES LIVRES

Nous avons reçu un nouveau livre sur le FORTH : Introduction au ZX FORTH de Marc Petreman et Michel Rousseau (Eyrolles). La préface commence par citer les évangiles et Göddel. Si j'avais dû acheter ce livre je ne serais sans doute pas allé plus loin et je l'aurais reposé en rayon. C'aurait été dommage, car il n'est pas sans intérêt. Il décrit les deux versions de FORTH disponibles pour le ZX80 (il n'y est pas question, sauf erreur de ma part de la possibilité d'utiliser les listings assembleur pour Z80, disponibles par exemple à PPC-T. Il est vrai que le prix des cassettes ne doit pas être beaucoup supérieur à celui du listing !). Il fournit d'assez nombreux exemples et consacre plus de soins que d'autres livres à la définition de tous les mots utilisés ce qui fait qu'il doit être utilisable même avec d'autres appareils. Ceci dit, dans l'exemplaire que j'ai reçu, figurent, de la main de l'auteur (je suppose) quelques corrections et compléments. Si l'auteur en question nous envoie une liste publiable, nous nous ferons un plaisir de publier cet erratum dans notre prochain numéro (vu, Marc ?).

LA PROGRAMMATION SYNTHETIQUE, C'EST FACILE ! par Keith Jarett, traduction par Gilles Barret. Après La programmation synthétique par Bill Wickes, premier ouvrage sur la question, un classique (toujours disponible, 100F aux Editions, 80F à la librairie du club pour les adhérents) les Editions du Cagire sont heureuses de vous annoncer la sortie prochaine de La programmation synthétique, c'est facile ! de Keith Jarett. Cet ouvrage, sans faire double emploi avec le précédent vous présente une approche plus simple, avec des moyens plus récents, de la programmation synthétique. Avec Au fond de la HP-41 par J-D Dodin (100F, 80F au club), et l'arrivée du ZENROM, la programmation synthétique devient accessible à tous et aussi simple que la programmation normale. Vous disposez maintenant de toutes les sources utiles pour en tirer le meilleur parti.

LES FONCTIONS D'EXTENSION, C'EST FACILE ! par Keith Jarett. Cet ouvrage est le premier et actuellement le seul à être consacré au module de fonctions d'extension mémoire et aux modules de mémoire d'extension. Il traite également des fonctions nouvelles de la HP-41CX. Ces modules font de la HP-41C un ordinateur particulièrement performant, mais son utilisation se heurte souvent à des difficultés de compréhension. Avec cet ouvrage, particulièrement soigné au point de vue pédagogique, vous maîtriserez rapidement votre module.

PRIX ET DISPONIBILITE : Les deux livres ci-dessus seront disponibles en principe le 15 Octobre (la traduction est faite, il reste à faire une relecture et l'impression),

au prix de 150F Chacun (le premier sera livré avec une carte des codes plastifiée en couleur). Pour avantager ceux d'entre vous qui acceptent de nous faire confiance, nous vous proposons un tarif spécial souscription de 120F pour chaque ouvrage pour toutes les commandes expédiées (cachet de la poste faisant foi) avant le 15 Octobre 84.

Compte tenu du nombre de programmes synthétiques contenus dans les ouvrages, et bien que les codes barre de ces programmes figurent dans les livres, nous vous proposons une cassette K. Jarrett, contenant tous les programmes des deux ouvrages. Nous pensons pouvoir tenir le prix de 150F pour cette cassette, mais nos difficultés d'approvisionnement nous obligent au conditionnel. En tous cas nous essaierons au maximum de respecter ce prix pour les commandes reçues avant le 15 Octobre. Les programmes de ces cassettes ne seront pas privatisés, et nous ne voyons aucun inconvénient à ce que vous fassiez une copie de la cassette pour un ami, à condition que vous n'en fassiez pas commerce. La copie des livres reste strictement interdite.

PROGRAMMATHEQUE

PROGRAMMATHEQUE

PROGRAMMES DISPONIBLES

CHEZ T33

DAMIEN DEBRIL
38 RUE DU 8 MAI 1945
59190 HAZEBROUCK
FRANCE
T:16(28)41.40.68

POUR LES CARTES PREVOIR

ENVELOPPE RETOUR AFFR.

A 1.70F OU 2.10F.

POUR LES CASSETTES

METTRE 2.40F OU 3.70F.

PREVOIR UN ENBALLAGE

SOLIDE CAR IL Y A DEJA

EU DES NEGATS DURANT LE

TRANSPORT PAR LA POSTE.

J'AI MIS LES NOMS SOUS

LESQUELS LES PROGRAMMES

SONT STOCKES.

VOUS POUVEZ PREPARER LES

CARTES EN Y INSCRIVANT

LES NOMS DES PROGRAMMES.

PREVOIR QUELQUES CARTES

SUPPLEMENTAIRES POUR

REMPLACER CELLES QUI

SERAIENT DEFAILLANTES.

HP 41 (suite)

(cf catalogue)

No 6 P 6 -1 C-TX

No 6 P 6 -1 C-IN

No 6 P 7 -1 C-EO

No 6 P 8 -1 C-DP

No 6 P 9 -2 C-PX

No 6 P 27 -4 C-MICRO-C

No 6 P 27 -4 C-M

No 7 P 3 -2 C-C-A

No 7 P 3 -1 C-REVEIL

No 7 P 4 -1 C-P

No 7 P 8 -3 C-CBG1

No 7 P 8 -1 C-IP

No 7 P 11 -6 C-AV

No 7 P 11 -4 C-AVDATA

No 7 P 12 -1 C-TSYNT

No 7 P 12 -1 C-SL

No 7 P 12 -1 C-PPC-T

No 7 P 13 -1 C-B-D

No 7 P 13 -1 C-DATA

No 7 P 13 -1 C-CDE

No 7 P 14 -1 C-ACC

No 7 P 15 -1 C-CAN

No 7 P 15 -1 C-PND

No 7 P 16 -2 C-PCB

No 7 P 18 -1 C-TRI

No 7 P 19 -2 C-SAV

No 7 P 21 -1 C-FAC

No 7 P 21 -1 C-ORP

No 7 P 22 -2 C-SAVE

No 7 P 24 -1 C-DEV

No 8 P 43 -5 C-CODEHP

No 8 P 44 -1 C-BESAS

No 8 P 44 -1 C-INPUT

No 8 P 45 -2 C-T0

No 8 P 45 -1 C-C=10

No 8 P 46 -1 C-XQ

No 8 P 47 -1 C-CHR\$

No 8 P 47 -5 C-ASSEMB

No 8 P 50 -1 C-INPUT2

No 8 P 51 -2 C-CHR\$2

No 9 P 56 -2 C-PEL

No 9 P 57 -1 C-PEL2

No 10 P 20 -2 C-TTR

No 10 P 21 -3 C-JACK

No 10 P 54 -1 C-IN2

No 10 P 55 -1 C-CR/CRD

No 10 P 56 -1 C-CR/CRD

No 10 P 56 -2 C-GAM+

No 10 P 61 -2 C-APRO

No 11 P 47 -1 C-DES

No 11 P 48 -1 C-RENAME

No 11 P 49 -1 C-F+

MR 1 P 8 -1 C-DCB

MR 1 P 15 -2 C-SYMB

MR 1 P 18 -4 C-INVC

MR 1 P 21 -2 C-HIST

MR 1 P 25 -3 C-TRIPL0T

MR 1 P 28 -1 C-PUISS

MR 1 P 29 -1 C-VN

MR 1 P 29 -1 C-PH

MR 1 P 30 -1 C-CHRO

MR 1 P 30 -1 C-CHROV

MR 1 P 33 -1 C-AS/B

MR 1 P 33 -2 C-AS/C

MR 1 P 93 -2 C-BA

MR 1 P 94 -1 C-BCPLOT

MR 1 P 98 -2 C-BC162

MR 1 P 111-1 C-CB

MR 1 P 111-1 C-XR

MR 1 P 0 -2 C-CBPG

HP 7 5

No 10 P 13 - EMPAGOLY

No 10 P 14 - EMPAGE75

No 10 P 15 - LISTOLY

No 10 P 16 - AZERTAC

No 10 P 16 - OLY

No 10 P 17 - COPCAS

No 10 P 17 - NETTCAS

No 10 P 18 - DUMPCAS

MR 1 P 36 - BUMP

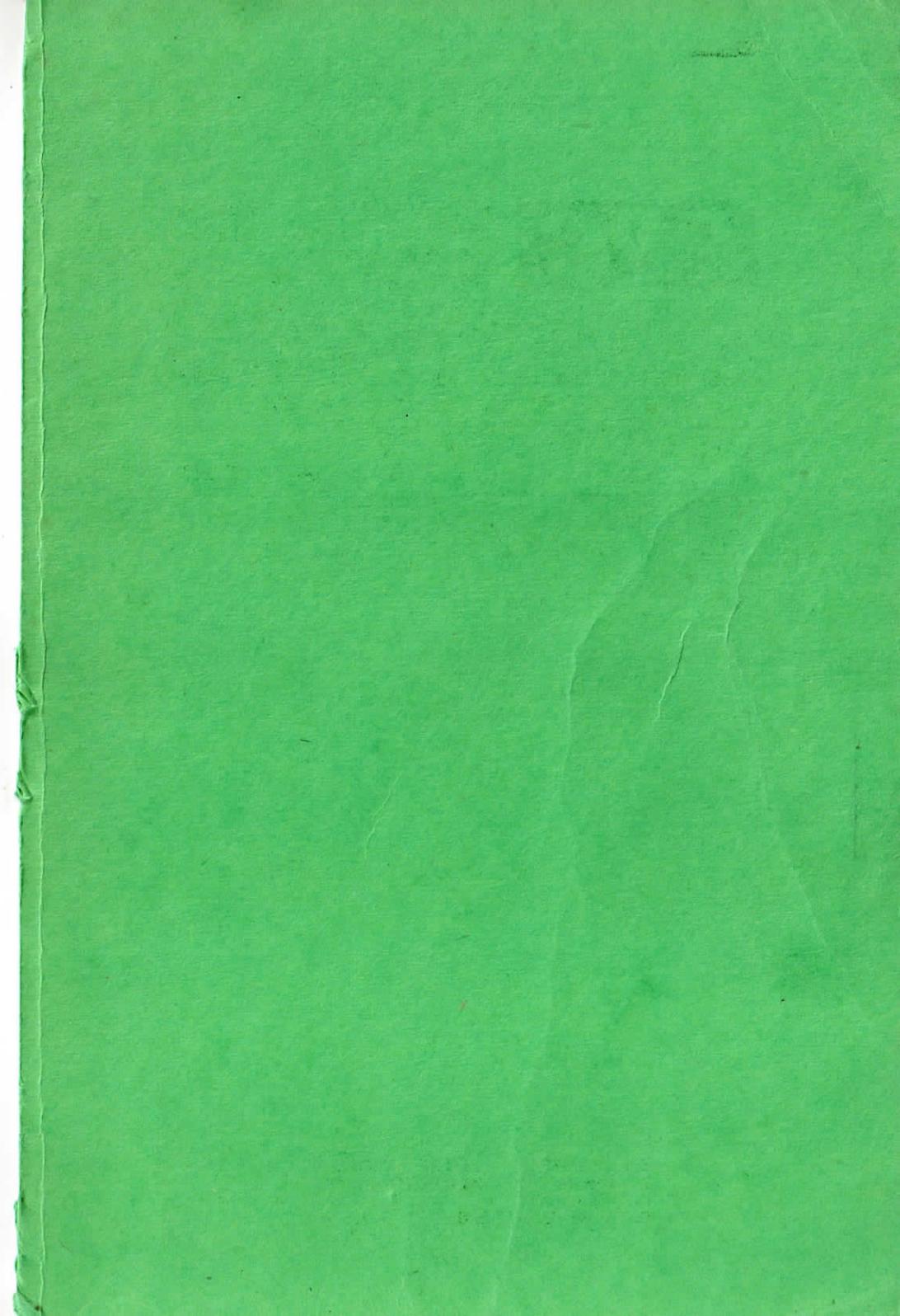
MR 1 P 37 - MAKEFILE

MR 1 P 38 - UTIL3

MR 1 P 38 - MENDIR

MR 1 P 40 - NEMBUMPX

MR 1 P 42 - CATALPHA



PPC

PROGRAMMATION SYNTHETIQUE, LCD, IMPRIMANTE, REGISTRES, MODULE, TOUCHES, OVER LAYS, MATRICES, MEMOIRE CONSTANTE, LECTEUR

DE *CODES A BARRES, LECTEUR DE CARTES, BASIC, RPN, FORTH, LANGAGE SPECIALISE, MICROCODE, ASSEMBLEUR, MATHEMATIQUES, SOUS PROGRAMME, NOMBRE ALEATOIRE, DRAPEAUX, LUCASIEWICZ, PILE, ALPHANUMERIQUE, PARASITES, RI-DEAU, SWAP, NOTATION POLONAISE INVERSE, DUP, SHIFT, ECRAN, MOT, COMPILER, EXECUTION, TESTS, BOUCLES, DO LOOP, BEGIN UNTIL, IF ELSE THEN, : ;, BOUCLES CONDITIONNELLES, LABELS, TABLEAUX, TRACE, SST, BST, R/S

FORTH
Interest
Group

Edité par PPC-T — TITRE : MICRO-REVUE
77, rue du Cagire 31100 TOULOUSE
ISSN en cours
Directeur de publication : J. D. Dodin