

JPC**PPC-PC**

JPC
 DECEMBRE-
 JANVIER
 1984/1985
 NUMERO-20

LE NUMERO 35FF

LE BUREAU	1	EDITORIAL
PIERRE DAVID	2	PPC PARIS SE REUNIT 1 FOIS / MOIS
PHILIPPE GUEZ	3	COOPERATIVE / ACHATS GROUPES
PRESSE	4	LU DANS LE HP ACTUALITES
	6	LE COURRIER DES LECTEURS

- HP41

JEAN-CLAUDE BECKER	12	LES FICHIERS ASCII DE LA 41
JEAN-CLAUDE BECKER	19	GESTION D'UN CARNET D'ADRESSE
JEAN-PIERRE TOYRE	21	TABLEAUX PRATIQUE / PRGM 'DR'
DENIS CASTELAIN	24	PRGM TAROT
JACQUES VAUCELLE	25	CONDENSATION DE DONNEES EN BASE 256
JEAN-JACQUES MOREAU	29	CHOC EN RETOUR PC n°18 : PRGM INTG

- HP75/71

FREDERIC BARNAUD	46	HP75 PUISSANCE 4 (PRGM BASIC)
DAVID / MARTINET	52	HP71 DESASSEMBLAGE DE KEYWAIT\$
PIERRE DAVID	56	HP71 EDITEUR DE TEXTE ET FRALEX
PIERRE DAVID	57	HP71 APPROXIMATION DE FRACTIONS
JEAN-PIERRR BONDU	58	HP71 CATALOGUE ETENDU (BASIC)

Les illustrations sont de Christelle

... LE**PROTOCOLE****DU MODULE****HP IL ...**

EDITORIAL

Chers programmeurs fous,

Voici arrivé entre vos mains le cadeau que PPC PARIS vous fait: Un numero double bien rempli! Un dossier sur le protocole HPIL, des informations, des programmes ... Tout ce qui a fait votre régal deux ans durant...

L'assemblée générale amène quelques changements dans le bureau. Tout d'abord, le nombre des membres du bureau s'est réduit, pour cause d'études en province, et ensuite il faut plus de gens pour s'occuper de PPC. A ce sujet, tous les volontaires sont les bienvenus, même s'ils n'ont pas le titre de membre de bureau'.

D'autre part, vous avez pu constater que le club s'informatise (EH OUI!). Nous avons un traitement de texte sur HP71, et une grande partie du journal est imprimé entièrement sur matériel HP. Ce même journal, est envoyé dans des enveloppes, et sur ces enveloppes, il y a des étiquettes. Celles-ci sont faites encore une fois, grâce au HP71, et à ce qui va derrière. Ce fichier va bientôt s'étendre afin de gérer la quasi totalité des activités du PPC PARIS. Ceci est rendu possible par l'apparition des disquettes HPIL. Comme vous êtes déjà tous fichés (HE! HE! HE!) nous vous informons que, conformément à la loi informatique et liberté, vous avez la possibilité de consulter votre fiche. Mais vous n'apprendrez rien de nouveau car, pour le moment, il n'y a que votre nom et votre adresse!

PPCmement vôtre

LE BUREAU

DIVERS

PPC PARIS SE REUNIT UNE FOIS PAR MOIS

Comme vous le savez peut être déjà, PPC PARIS se réunit une fois par mois, en plein coeur de Paris. Amenez votre matériel, votre bonne volonté et vos idées ! Plus vous en apporterez, et plus vous en trouverez chez vos collègues de PPC. Ces réunions se déroulent de manière très libre, aucun ordre du jour, discussion ou autre n'étant imposé. Un membre du Bureau est toujours présent. Ainsi, si vous désirez remettre votre article tout frais au journal, si vous avez des suggestions à faire, si vous voulez vous procurer des anciens numéros de JPC, ce sera en principe toujours possible.

Si donc cela vous intéresse, n'hésitez plus un seul instant venez nous rejoindre tous les premiers samedi de chaque mois, au CENTRE DE JEUNESSE ET DE LOISIRS JEAN VERDIER
11, rue de Lancry
75010 PARIS

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

Pour l'accès en métro, trois possibilités s'offrent à vous à savoir:

Métro Strasbourg St Denis: sortie Porte St Martin / Bld St Denis, coté pairs.

Métro République: sortie Bld St Martin, coté pairs.

Métro J. Bonsergent: Sortie Bld Magenta, coté impairs.

Ah, j'oubliais: JPC est distribué en avant-première lors de ces réunions ... A bon entendeur Salut !

Les dates des prochaines réunions sont: (16h-19h)

- Le Samedi 1 Décembre 1984

- Le Samedi 5 Janvier 1985

- Le Samedi 2 Février 1985

A Bientot...
Pierre DAVID

NOUS EN AVONS
=====

Sont en vente au club:

- Cartes magnétiques pour HP41 au prix de 150FF les 50 cartes + 3,10 de port;
- Eproms (2716 et 2732) vierges au prix de 55FF + 3,10FF de port. Pour les Eproms programmées, nous consulter;
- Convertisseur HPIL (82166A) au prix de 1750FF + 3,10FF de port. livré avec connecteur et manuel;
- module TIME (82182A) au prix de 400FF + 3,10FF de port. Livré avec le manuel français;
- module XFONCTION (82180A) au prix de 400FF + 3,10FF de port. Livré avec le manuel en français;
- Module QUADRAM (82170A) au prix de 400FF + 3,10FF de port;
- module HPIL (82160A) au prix de 970FF + 3,10FF de port. Livré avec manuel en français.
- Le MLDL hollandais (société ERAMCO) au prix de 3860FF + port. 24k Eprom + 8k RAMde microcodes et livré avec l'eprom de 4k 'ERAMCO-MLDL'. Délai 1 à 2 mois.
- Pour les commandes de la ZENROM et de la CCDROM, le prix de souscription est de 1000FF, envoyez nous 500FF minimum d'arrhes. La ROM PANAME est maintenant disponible auprès de JJ DHENIN.
- VASM (listing commenté par HP des 3 premières pages de la 41) au prix de 150FF + 13,50FF de port;
- Le manuel de service de la 41 au prix de 50FF + 6,50 de port;
- le manuel technique du convertisseur au prix de 20FF + 3,10FF de port;
- Les anciens numéros au prix de:
 - n°1 à 3 au prix de 15FF + 3,10FF de port par numéro,
 - n°4 à 10 au prix de 20FF + 3,10FF de port par numéro,
 - n°11 à 17 au prix de 25FF + 6,50FF de port par numéro,à partir du numéro 18 au prix de 35FF + 6,50FF de port par numero.

Les ports sont les tarifs PTT actuels et donnés en non urgent

ACHATS GROUPES
=====

Nous avons des commandes groupées en route. Pour ceux qui désirent du matériel, faites nous des propositions et donnez nous des promesses fermes.

En ce moment, nous avons une commande de modules FORTH/ASS pour HP71B et une commande de module CCDROM pour HP41

Le Portable de Hewlett-Packard

Pas plus volumineux qu'un classeur, le Portable de Hewlett-Packard possède, intégrées dans un petit boîtier, de puissantes fonctions informatiques. Ce nouvel ordinateur est doté du système d'exploitation MS™-DOS, d'un logiciel intégré, d'une mémoire utilisateur de grande capacité et d'un disque électronique. Il est compatible avec de nombreux ordinateurs personnels.

Le Portable « s'ouvre » sur un écran à cristaux liquides de 16 lignes de 80 colonnes, à adressage par bit, qui permet d'afficher des graphiques de 128 x 480 points image (pixels). Son clavier est semblable à celui de l'ordinateur personnel HP 150 pour ne pas vous dérouter. Une horloge temps réel vous aide à vous souvenir de vos rendez-vous ou à organiser et localiser les fichiers par nom ou par date et heure. Il possède même une alarme qui avertit des appels extérieurs ou avertit du transfert des données.

Mémoire utilisateur permanente de grande capacité

La ROM de 348 K octets du Portable contient le système d'exploitation, des écrans d'aide accessibles facilement ainsi que les logiciels suivants :

- Gestionnaire d'applications personnelles (P.A.M.). Ce menu principal, qui rend inutile l'emploi de la syntaxe complexe de l'ordinateur, vous permet d'utiliser le Portable en quelques secondes simplement en appuyant sur une touche quelconque pour sélectionner le programme désiré.
- Lotus™ 1.2.3.™. Ce progiciel vous permet de passer de l'analyse sur feuille de calcul à la gestion des informations et aux graphiques sans avoir besoin de charger des programmes ou des données supplémentaires.
- MemoMaker: ce traitement de texte compatible avec WordStar® élimine la nécessité de l'apprentissage de séquences de commande spéciales.
- Communications/émulateur de terminal : ce progiciel, utilisé avec le coupleur acoustique en HP-IL, vous permet de communiquer avec les fichiers d'autres ordinateurs ou de les partager.

Vous pouvez stocker des programmes et des données supplémentaires dans la RAM de 272 K octets du Portable. Une partie de cette mémoire peut servir d'unité de disque transistorisée, à grande vitesse, appelée « disque électronique », qui stocke et extrait les informations plus de six fois plus rapidement que les unités de disque souple classiques.

Une unité de disque alimentée par batterie pour votre ordinateur portable

Avec sa capacité formatée de 710 K octets sur un disque souple de 3,5 pouces (lecture-écriture sur les deux faces), l'unité de disque portable HP 9114A constitue le complément idéal de l'ordinateur Portable Hewlett-Packard. Elle lui permet d'avoir accès à toute une série d'applications telles que Multiplan®, dBASE II™ et WordStar®.

Alimentée par une batterie sèche étanche, la HP 9114A fonctionne, en moyenne, pendant huit heures avant d'avoir besoin d'être rechargée, opération qui prend environ cinq heures. Vous pouvez continuer à utiliser l'unité de disque pendant que la batterie se recharge, car une utilisation même prolongée sur le secteur n'a aucun effet nocif sur la batterie. Une fonction de mise



Le Portable, un ordinateur personnel Hewlett-Packard alimenté par batterie, pèse environ 4 kg et mesure 33 x 25,4 x 7,6 cm. Il présente des applications intégrées et un écran de 16 lignes et de 80 colonnes.

Une carte enfichable optionnelle, appelée Portable-Desktop Link (PDL), permet au Portable de communiquer avec le HP 150 et les ordinateurs personnels d'IBM (et ordinateurs compatibles). La carte PDL HP 82973A comprend une interface HP-IL (Hewlett-Packard Interface Loop) et le logiciel de communication du Portable avec l'ordinateur personnel d'IBM et l'accès à ses périphériques. Elle permet aussi à l'ordinateur personnel d'IBM d'accéder aux périphériques HP-IL.

A l'aide de la carte PDL HP 45463A, il est possible de charger une feuille de calcul électronique du HP 150 dans le Portable, d'y travailler chez soi, puis de recharger, le jour suivant, la feuille de calcul révisée dans le HP 150.

Vous pouvez également utiliser l'interface série intégrée RS-232-C pour relier le Portable à d'autres ordinateurs ou le connecter à d'autres périphériques RS-232-C tels que des imprimantes de qualité courrier.

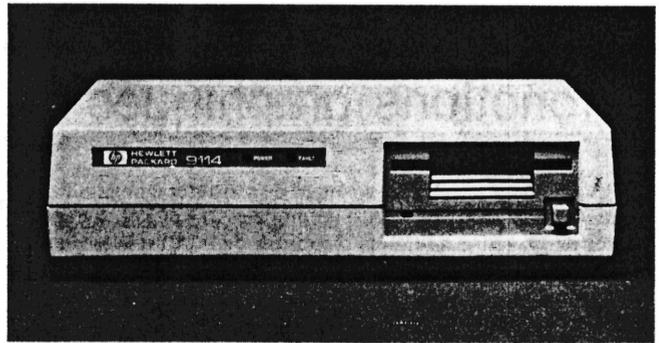
sous et hors tension automatique réduit la consommation d'énergie dès que vous en avez besoin. Cette méthode permet de prolonger la durée d'utilisation de la batterie. La fiabilité de la HP 9114A est assurée par un certain nombre de fonctions standard comprenant, en particulier, un dispositif de surveillance du support, qui signale la nécessité de remplacer les disques avant qu'ils ne soient usés.

Compatibilité

Les données stockées sur les disques HP 9114A au format simple face peuvent être lues par l'ordinateur personnel HP 150 et permettent donc d'échanger facilement des données entre les ordi-

uateurs portables et les ordinateurs personnels HP. Il est possible de communiquer avec l'ordinateur personnel IBM en lui connectant directement le HP 9114A au moyen de la carte enfichable PDL (Portable Desktop Link) qui permet à l'ordinateur personnel IBM de dialoguer avec la HP 9114A comme s'il s'agissait d'une unité de disque IBM. La compatibilité existant entre le Portable, le HP 150 et l'ordinateur personnel IBM donne la possibilité de transférer des fichiers de données entre ces ordinateurs sur des disques de 3,5 pouces. La HP 9114A peut également être connectée aux ordinateurs de poche HP-41, HP-71 et HP-75.

L'unité de disque HP 9114A ne pèse que 2,5 kg et mesure 20 x 29 x 7,6 cm.



La nouvelle unité de disque est portable, très légère et a une capacité de 710 K octets (avec formatage). Elle est compatible avec les ordinateurs personnels HP et IBM ainsi qu'avec les ordinateurs de poche HP.

Une imprimante portable silencieuse équipée d'une tête d'impression à jet d'encre

L'imprimante ThinkJet est un outil indispensable à votre ordinateur portable Hewlett-Packard. Elle est silencieuse et réalise, à grande vitesse, des travaux d'impression d'une haute qualité. Cette imprimante compacte est équipée d'un nouveau système d'impression à jet d'encre dont la tête d'impression et le dispositif de projection de l'encre sont associés dans une cartouche jetable. Comme une nouvelle tête d'impression est mise en service chaque fois que l'on renouvelle l'encre, l'impression est toujours nette et précise.

Du fait que la tête d'impression ne touche jamais le papier, l'imprimante ThinkJet est sensiblement plus silencieuse que la plupart des imprimantes et fonctionne à un niveau sonore inférieur à 50 dBA. La tête d'impression « peint » littéralement les caractères sur le papier en pulvérisant l'encre par 12 gicleurs microscopiques. La tête d'impression survole le papier au ras de celui-ci, à la vitesse de 150 caractères par seconde, soit environ 1 300 mots par minute. L'excellente qualité d'impression obtenue est également due à l'emploi d'une matrice de caractères de 11 x 12 points.

On peut utiliser du papier en continu à pliage paravent ou des feuilles simples. On obtient la meilleure qualité d'impression avec le papier spécial pour imprimante à jet d'encre fourni par HP ou les distributeurs agréés. Un mécanisme d'entraînement par picots peut être réglé de façon à employer le papier au format 8,5 x 11 pouces ou au format A4.

Compatible avec de nombreux ordinateurs personnels

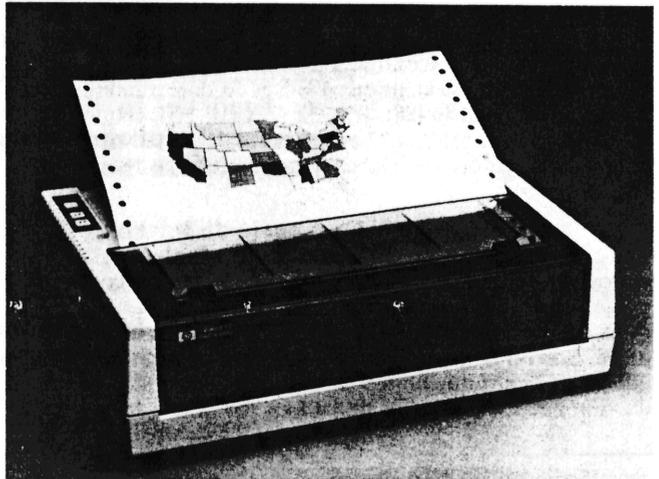
L'imprimante ThinkJet peut être commandée avec l'une des trois interfaces suivantes. Le modèle HP 2225A a une interface HP-IB (IEEE-488) destinée aux instruments et aux ordinateurs HP, tandis que le modèle HP 2225B possède une interface HP-IL qui permet de la connecter aux ordinateurs de poche HP. Pour les ordinateurs personnels des autres constructeurs, y compris l'IBM PC, l'Apple II et les modèles de Compaq et Texas Instruments, vous pouvez commander le HP 2225C qui est équipé d'une interface parallèle Centronics.

Le modèle HP 2225B est une imprimante portable, alimentée par une batterie NiCad rechargeable qui imprime jusqu'à 200 pages avant de devoir être rechargée. Il est également possible d'utiliser cette imprimante pendant le chargement de la batterie.

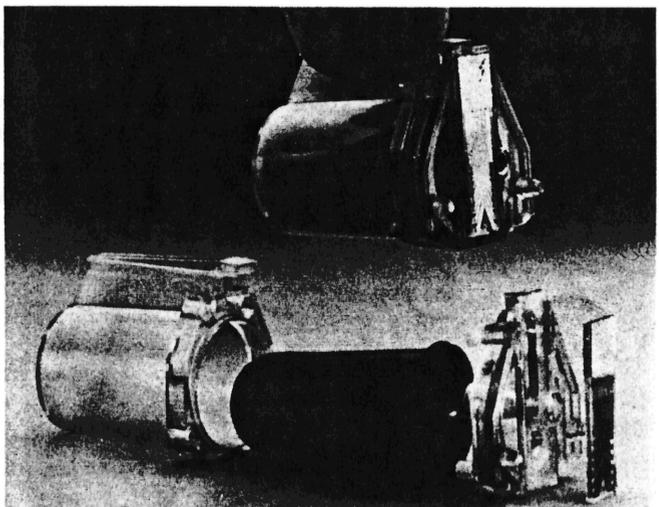
En mode graphique, la résolution de l'imprimante ThinkJet est de 96 x 96 ou 192 x 96 points par pouce. Elle utilise des trames graphiques standard ou des graphiques en colonnes.

La ThinkJet imprime en anglais, allemand, danois, espagnol, finnois, français, italien, néerlandais, norvégien, portugais et suédois. Avec l'interface Centronics du HP 2225C, vous pouvez également obtenir 6 jeux de caractères ISO. Vous avez, en outre, le choix entre quatre densités d'impression et vous pouvez utiliser les caractères gras et le soulignement sans ralentissement de la vitesse d'impression.

Les imprimantes ThinkJet mesurent 29,2 x 20,6 x 8,9 cm et leurs poids sont compris entre 2,5 et 3,4 kg.



En mode graphique, la résolution de l'imprimante ThinkJet est de 96 x 96 ou 192 x 96 points par pouce.



La tête d'impression à jet d'encre jetable utilisée par les imprimantes ThinkJet contient environ 3 cm³ d'encre, suffisamment pour imprimer 500 pages de texte. Elle est transparente ce qui permet de voir la quantité d'encre restante et de remplacer la cartouche avant qu'elle ne soit épuisée.

SAUNIER STEPHANE (PC78)
11, Allée de la Source
78480

Cher Philippe,

Tout d'abord, je te remercie d'avoir fait passer mon message au sujet de la CCDROM (celà fait plaisir de pouvoir compter sur un club!).

Voici les nouvelles concernant cette affaire; ma lettre est parvenu à Wolfgang pendant les vacances et j'ai récemment reçu le chèque que je lui avait donné. Je tiens à saluer ici l'agentillesse et l'honnêteté de ce garçon.

Voilà, mes problèmes sont finis avec cette ROM!

Je suis en pourparlé pour la vente de mon système 41CV, et sa vente est imminente. Seulement, il me reste tout une documentation sur la 41, alors, je me suis dit que je pourrai en faire bénéficier les membres du club:

- programmer HP41 (PSI)
- HP41 EXTENDED FONCTIONS MADE EASY (K. JARETT)
- calculator tips & routines (J. DEARING)
- au fond de la HP41 (J.D. DODIN)
- synthetic programming on the HP41C (W.C. WICKES)
- time module solution I (+ codes barres)

pour 250 F, et en prime, tous les JPC et presque tous les PPCT (manque 2 ou 3)!

J'espère que cette offre intéressera quelqu'un et il ne reste plus qu'à dire: Merci pour tout et longue vie à JPC.

S. SAUNIER

ADRIEN FLORENT (PPCPC 105)
47 Avenue de la division Leclerc
92320 CHATILLON

Cher président,

Je me permet de vous écrire pour vous signaler mon changement d'adresse. Je voudrais également féliciter toute l'équipe qui réalise le journal, dont la qualité s'amméliore de plus en plus. Pourriez vous me faire connaître les dates et lieu des prochaines réunions du club?

En vous remerciant et en souhaitant encore longue vie au PPC-PC, veuillez agréer, cher président, l'expression de mes sentiments les meilleurs.

F. ADRIEN

Cher ami, en ce qui concerne les dates et lieu des prohaines réunions, voir la page qui y est consacrée 'PPC PARIS SE REUNIT UNE FOIS PAR MOIS'. Merci d'autre part pour l'équipe qui réalise le journal. N'oublie pas que tu fais partie de cette équipe, et si tu as réalisé quelquechose, fait le moi parvenir

A bientôt

PHILIPPE

J.M. ANCELLIN
58, rue d'assas
75006 PARIS
tel: 268 15 00

29 / 10 / 84

Madame, Monsieur,

Utilisateur de calculatrices HP41CX, HP12C, HP15C, je suis à la recherche d'un algorithme de calcul de la fonction GAMMA (factorielles généralisées) donnant la même précision que celui, malheureusement inconnu de la HP15C.

Pourriez vous m'aider à résoudre ce problème? On peut songer à un développement en série de la formule de STIRLING, ou à une approximation par polynômes de TCHEBYCHEFF, l'approximation polynomiale fournie dans le livret HP "math haut niveau" étant très insuffisante.

J'ai par ailleurs observé une anomalie de la fonction factorielle entière pour les grandes valeurs de n. Si on calcule $\frac{n!}{n(n-1)(\dots)1}$, on trouve bien 1,000000000 jusqu'à un certain rang mais une erreur de quelques unités de la dernière décimale apparaît à partir d'un certain rang. Pouvez vous m'indiquer la raison de cette anomalie?

D'avance je vous remercie et je vous prie d'agréer, Madame, Monsieur, l'expression de ma parfaite considération.

J.M. ANCELIN

PHILIPPE TENAND

ANOMALIE SUR HP41: Additions graphique

Je ne sais si cette anomalie est valable sur toutes les HP41. Cela ne marche pas quand cette configuration: HP41CV, lecteur de carte, PPCROM en sortie 1, GAMES en sortie 2, MATH en sortie 3, quand un élément manque, l'anomalie se résume en une anomalie déjà connue. Mais il ne se passe pas de défilement anormal que je vais aborder.

En mode normal, faire: CAT 2 ou 3 ou 1 au choix.
S'arrêter en un endroit quelconque. Exécuter SST ou BST non pas par la touche prévue, mais lettre par lettre XEQ S puis S puis T.
Cela ressemble à une anomalie déjà connue. La suite diffère grâce à la configuration spéciale de l'ensemble utilisé.
Le catalogue se pointe donc sur la fonction suivante.
Faire ← .

LA COMMENCE L'ANOMALIE.

Un ensemble de trois caractères alphabétiques (y compris ceux que l'on ne découvre que par la programmation synthétique), toujours différents suivant le catalogue choisi, ou l'endroit choisi dans ce catalogue, viendra défiler de gauche à droite. Deuxièmement à droite, un bloc de caractères (constitué d'autant d'éléments qui devraient constituer l'affichage de la fonction ou le pointeur s'est arrêté) reste à droite de l'affichage. Les caractères sont aléatoires, mais constants à chaque essai.

Le premier bloc mobile viendra dans le bloc de droite le traverser et reviendra par la gauche.

Durant le passage, la superposition des deux blocs donnera une addition préétablie et dépendant sûrement de leur code, ceci suivant une logique rigoureuse.

Voici mon exemple: ayant choisi de m'arrêter avant ADV, avec S/S/T, le pointeur affiche ADV ← , le premier bloc est: X \$&; le second bloc C̄P. En se superposant au passage, les seuls trois derniers blocs d'affichage sont, après chaque mouvement:

C̄P; F̄P; D̄P; ADV; C̄AT; C̄CQ; C̄P.

nous remarquons tout de suite qu'à la superposition rigoureuse (3 sur 3) nous avons: X \$& + C̄P = ADV

Y a t'il une logique mathématique à tout cela?

OUI, nous le voyons à chaque décalage.

Et par la logique, nous déduisons que: 1) & + C̄ = F

2) & + P = V

3) X + C̄ = A

4) \$ + C̄ = D

5) \$ + P = T

6) X + P = Q

Une même logique, existe pour n'importe quel essai.

Voyons un peu les codes:

Prenons les trois opérations avec: $x + C̄ = y$

$& + C̄ = F$

$38 + 64 = 102$, c'est un f minuscule au lieu de majuscule.

$X + C̄ = A$

$33 + 64 = 97$, c'est un a minuscule au lieu de majuscule.

$\$ + C̄ = D$

$36 + 64 = 100$, c'est un d minuscule au lieu de majuscule. Le décalage pour ces trois exemples semble constant, et serait de 32.

Prenons les trois autres opérations avec: $x + P = y$

$& + P = V$

$38 + 80 = 118$, c'est un v minuscule (un paté) au lieu d'un V majuscule (code 86), différence 32.

$\$ + P = T$

$36 + 80 = 116$, c'est un t minuscule au lieu d'un T majuscule (code 84)

$X + P = Q$

$33 + 80 = 113$, c'est un q minuscule.

Nous voyons ici une logique mathématique, un décalage par rapport au code décimal connu pour l'addition. IL vous reste à saisir la logique du pourquoi, et à chercher une théorie en vous basant sur de multiples essais. Est ce le fait que $2 \times 32 = 64$, code C̄ a une importance dans cette exemple, pourquoi 32, ce chiffre varie t'il? Qu'est ce qui sépare les deux caractères de l'opération arithmétique du caractère du résultat?

Nous savons juste que la somme des deux donne exactement le caractère contenu dans la fonction ou le pointeur s'est arrêté quand les caractères se juxtaposent entièrement (3 caractères au dessus de trois autres dans l'exemple.)

Est ce qu'une fonction de la HP est décomposée en deux ordres suivant les colonnes et les lignes du clavier, ordres dont chacun aurait un nom particulier?

Pourquoi la configuration donnée et la PPCROM sont ils obligatoires? C'est à vous de chercher la logique de tout cela.

Notez que n'ayant pas d'imprimante, je ne me suis pas intéressé au codage graphique ou le chiffre 32 a peut être son importance. Peut être faut il trois modules différents ou les quatres sorties utilisées pour que cela marche.

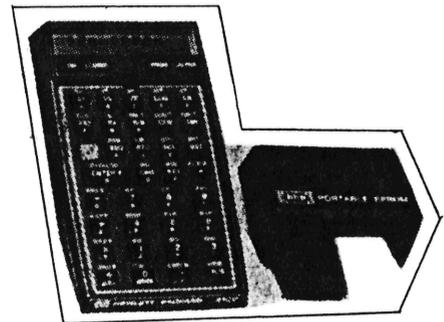
Philippe Tenand (44)
2 bis rue de l'égalité
94300 VINCENNES

Parlez nous du module HP (USA): math 2
et du module HP Paname (ajoutant des centaines de fonctions)
Donnez une photo du joli HHP (façon lecteur de cartes), faites en une critique
Demandons à HP d'éditer des cartes magnétiques enregistrées pour HP41 (en grande quantité), celles pour HP67/97 étant hors circulation.
Donnez nous des extraits techniques sur la HP41, par exemple quand il y a des trous dans JPC.
La qualité de JPC est meilleure, bel effort, BRAVO. A quand une photographie en couverture? (même noir et blanc).
Expliquez le principe du MEM-O-PACK qui permet de retirer des modules et d'en remettre sans perte d'informations (même QUAD, TIME, ...)

Philippe Tenand.

Cher ami,

En ce qui concerne le module math 2, je ne peux en parler car je ne l'ai pas, mais peut être un membre le possède t'il. Le module Paname devrait très prochainement voir un article qui lui sera consacré. Le nouveau lecteur d'EPROM de HHP peut contenir 32 k d'EPROM. Il est livré avec une EPROM de 8k et son prix approche 350\$. Quand je peux vous fournir des extraits techniques des matériels, je le fait, mais la plupart de ces document sont en copyriht par HP. Nous avons une très large marge en ce qui concerne l'épaisseur du journal, et je pense que les trous peuvent rester, ou être comblé par des dessins de PILOUSAN. Merci pour le compliment sur le journal mais à moins que tout le monde soit contre, nous garderons la page de couverture telle quelle. Envoyez moi vos idées à cesujet. En ce qui concerne le MEM-O-PACK, je demande à un membre de nous faire un petit article.



Philippe.

COMME TOUS LES ANS, NOUS ALLONS FAIRE UNE LISTE DES ARTICLES PARUS EN 1984

APPLICTIONS	MACH.	AUTEUR	N°	PAGE	TITRE
41		ROBERT SCHWARTZ	11	2	LE REGISTRE 'e'
41		ROBERT SCHWARTZ	11	3	LE REGISTRE 'q'
41		ROBERT SCHWARTZ	11	3	LE 'Q-LOADER'
41		ROBERT SCHWARTZ	11	3	LE REGISTRE 'c'
41		ROBERT SCHWARTZ	11	4	END ET COMPILATION
41		PIERRE ANTOINE	11	8	CASSETTES HP: LE CAT (suite)
41		PIERRE ANTOINE	12	11	ACCIDENTS DE CASSETTE
41		ANDRE OISEL	12	18	IMPRIMANTE 143 ET CARAC. DE CONTROLE
41		DANIEL JACOB	12	20	STRUCTURE FICHIERS
41		ROGER CHARPENTIER	13	13	LISTE DES CODES A PARTICULARITES
41		JEAN-PIERRE TOYRE	16	18	CREATION D'UN XROM
41		BERNARD MORISSEAU	16	18	BLDSPREC DEMYSTIFIE
41		DANIEL SAADA	17	19	FONCTION LN(1+X) ET EXP(X-1)
41		JEAN-PIERRE TOYRE	18	6	CREATION D'UN XROM (suite)
41		PIERRE ANTOINE	18	8	BIZZARERIES SUR HP41CX
41		BERNARD MORISSEAU	19	9	'BG' - 'BJ' COMMENT CA MARCHE
41		JEAN-CLAUDE BECKER	20		LES FICHIERS ASCII
41		JACOUES VAUCELLES	20		CONDENSATION EN BASE 256
41		PIERRE DAVID	20		EDITEUR DE TEXTE ET FRALEX

LE PETIT THEATRE DES MICROCODES

71	DAVID/MARTINET	18	25	JPCLISTE
71	MICHEL MARTINET	18	25	VOLABEL
71	PIERRE LANGLOIS	18	26	NUAGE
71	OLIVIER ARBEY	18	26	PIECE MACHINE
71	PIERRE LANGLOIS	18	27	ROUTINE D'INVERSION VIDEO
71	PIERRE DAVID	18	27	MOT FORTH: 'MSGS'
41	PIERRE DAVID	18	28	POLYCOPY
41	PHILIPPE DAVASE	19	14	ARRACHEMENT DES ANCRAGES DANS LE BETON
41	JEAN-CLAUDE BECKER	19	14	MONSTRES ET DONJONS (2° partie)
41	CHRISTIAN JEGOUZO	19	16	RENAME (FICHIERS DU XFCI)
41	CHRISTIAN JEGOUZO	19	17	BOU: BOURSE ET INVESTISSEMENT
71	JEAN-PIERRE BONDU	19	25	FORTH: INVERSION DE L'AFFICHAGE
71	LAURENT ISTRIA	19	26	ALICE ET LE MIROIR
71	PIERRE DAVID	19	27	LE 71B ET LA RECURSIVITE
41	JEAN-CLAUDE BECKER	20	34	LISTEUR DE VARIABLES
41	PIERRE DAVID	20		FICH/a
41	JEAN-PIERRE TOYRE	20		DR
41	DENIS CASTELAIN	20		TAROT
41	FREDERIC BARNAUD	20		PUISSANCE 4
71	PIERRE DAVID	20		APPROXIMATION DE FRACTIONS
71	JEAN-PIERRE BONDU	20		CATALOGUE ETENDU

BANCS D'ESSAI

71	PIERRE DAVID	13	1	LE HP71B
41	DANIEL JACOB	14	11	LE MLDL HOLLANDAIS
71	PIERRE DAVID	14	45	ADDITIF AU BANC D'ESSAI
IMP	PHILIPPE GUEZ	15	48	TABLE TRACANTE CANON X710
71	PIERRE DAVID	16	28	LE MODULE FORTH/ASS.
110	OLIVIER ARBEY	18	23	IMPRESSIONS SUR LE HP110
71	LAURENT ISTRIA	19	39	MODULE MATH.
IL	MARTINET/DAVID	20		LE PROTOCOLE DU HP11L

DUR ET MOU

41	LOHRS/OISEL	12	8	LE MLDL HOLLANDAIS
41	JACOUES VAUCELLE	12	30	HP41CX GONFLEES
41	JEAN-CLAUDE BECKER	14	5	LA 41 MENE GRAND TRAIN
41	JEAN-CLAUDE BECKER	14	7	COMMANDE D'APPAREILS ELECTRIQUES
IMP	PIERRE DAVID	14	9	URE NOUVELLE IMPRIMANTE IL
IMP	JACOUES VAUCELLE	14	10	HISTORIQUE TECHNIQUE DE LA NLE IMP
41	PHILIPPE GUEZ	14	22	PROGRAMMATION D'EPROMS
41	GABRIEL GIL	16	25	CARTE DE COMMANDE DE 16 RELAIS
41	JEAN-CLAUDE BECKER	19	22	LA 41 MENE GRAND TRAIN (suite)

LA ROUTE DE LA ROM

41	PHILIPPE GUEZ	12	12	NFCROM 1B, ASSEMBLEUR 3
41	PHILIPPE GUEZ	13	30	ERAMCO, ADELIMANN
41	PHILIPPE GUEZ	14	19	MLEPROM 1H, MLDL OS 1A
41	PHILIPPE GUEZ	15	12	CDROM

PROGRAMMES (BASIC, RPN, ...)

41	PHILIPPE GUEZ	11	4	ID55 (INVERSION DRAPEAU 55)
41	PHILIPPE GUEZ	11	5	COD ET DECOD
41	PHILIPPE GUEZ	11	5	RCLR ET STOR
41	DANIEL JACOB	11	6	RCR ET STR
41	FREDERIC POUPON	11	10	JOUONS A PUISSANCE 4
41	PIERRE DAVID	11	12	XEDIT (EDITEUR EN XMEM)
41	GILBERT TISSERAND	11	16	FLAGS
41	AIME PIERRARD	12	23	XXX (REPERTOIRE)
41	GILBERT TISSERAND	12	24	SOS CODES BARRES
41	PHILIPPE GUEZ	12	26	ACFCT (ADRESSE ET CODE HEX D'1 FCT)
71	PIERRE DAVID	13	17	CONTRAST
71	PIERRE DAVID	13	17	REMPLISSAGE DE L'AFFICHEUR GRAPHIQUE
71	PIERRE DAVID	13	17	RACINES DE L'EQUATION f(x)=0
71	PIERRE DAVID	13	19	CARACT
41	JEAN-CLAUDE BECKER	13	26	SOS (1° SOINS)
41	PIERRE DAVID	13	32	ARITHMETIQUE FRACTIONNAIRE
41	PIERRE LANGLOIS	13	34	SH, BI, DE, DIV
41	ALAIN ROSSET	14	24	SIMULATION D'ECRAN GRAPH. SUR IMP
41	PIERRE DAVID	14	32	XEQS (XEQ SYNTHETIQUE)
71	PIERRE DAVID	14	34	LA QUETE DES FLAGS
71	PIERRE DAVID	14	36	CALCUL SUR LES FRACTIONS
71	PIERRE DAVID	14	41	COURSE DE VOITURE
71	PIERRE DAVID	15	18	LABYRINTHE
41	PIERRE LANGLOIS	15	23	PILE RACCOURCIE POUR MAT. D'ORDRE 2
41	PIERRE LANGLOIS	15	25	TIMER
41	PIERRE LANGLOIS	15	25	L (GTO IND ALPHA)
41	PIERRE LANGLOIS	15	25	MN (MINUTERIE)
41	PIERRE LANGLOIS	15	26	DICHO (DICHOTOMIE)
41	ERIC DELAGNES	15	27	APAF (APPLICATIONS AFFINES)
41	GILBERT TISSERAND	15	33	SAUTE MOUTON
41	CHRISTIAN JEGOUZO	16	8	MODULATION HORAIRE DE LA TAXE TEL.
41	PIERRE DAVID	16	11	REVEIL MATIN
71	PIERRE DAVID	16	30	MOT FORTH: 'VLIST'
71	MICHEL MARTINET	16	31	DIALOGUE
71	OLIVIER ARBEY	16	35	ELECMUN
71	CHRISTIAN BACHELET	16	39	POSITIONNEMENT DE FLAGS
41	FREDERIC POUPON	17	10	LLIST (LISTAGE DES LIGNES DE TEXTE)
41	LAUR.-JOLLIA FERRIER	17	12	EDITEUR DE TEXTE
71	PIERRE DAVID	17	21	FORTH: 'COURSE DE VOITURE'
71	MICHEL MARTINET	17	24	CHARSET
71	JEAN-PIERRE BONDU	17	25	ROUTINES GRAPHIQUES
71	PIERRE DAVID	17	25	DATE
71	LAURENT ISTRIA	17	25	MATRICE
71	JEAN-PIERRE BONDU	17	28	DESSIL (DAO)
71	MICHEL MARTINET	17	29	EDITEUR DE CASSETTES
71	PIERRE DAVID	17	30	GESTION DE L'HPIL
41	PHILIPPE GUEZ	18	8	ENTREE DE CAR. EN ALPHA
41	JEAN-CLAUDE BECKER	18	10	DIRE
41	JEAN-CLAUDE BECKER	18	11	ENVOIE
41	DENIS CASTELAIN	18	13	COMMUNICATION TELEPHONIQUE
41	GILBERT TISSERAND	18	14	TRADUCTION A LA GREC DU PRGM WUMPUS
41	JEAN-JACQUES MOREAU	18	16	INTG, LEC

LA REVUE DES CODES BARRES

41	GILBERT TISSERAND	11	17	FLAGS
41	GILBERT TISSERAND	11	22	SIZE 000 à 255
41	GILBERT TISSERAND	11	23	STO..., RCL..., ASTO..., ARCL...
41	GILBERT TISSERAND	11	24	STO IND..., RCL IND..., X()IND..., ISG IND.
41	GILBERT TISSERAND	11	25	ASTOIND..., ARCLIND..., VIEWIND..., DSEIND..
41	GILBERT TISSERAND	11	26	ST+IND..., ST-IND..., ST*IND..., ST/IND...
41	GILBERT TISSERAND	11	27	TONES 10 à 127
41	GABRIEL GIL	11	29	XROM 20,0 à 23,63
41	GILBERT TISSERAND	15	35	SAUTE MOUTON
41	GILBERT TISSERAND	15	38	FONCTIONS DE LA PPCROM
41	GABRIEL GIL	15	40	XROM 24,0 à 31,63

HP. 41

RPN:

- | | |
|---|----------------------|
| 1 : LES FICHIERS ASCII DE LA 41 | (JEAN-CLAUDE BECKER) |
| 2 : GESTION D' UN CARNET D'ADRESSES | (JEAN-CLAUDE BECKER) |
| 3 : TABLEAUX PRATIQUES | (JEAN-PIERRE TOYRE) |
| 4 : PROGRAMME DR | (JEAN-PIERRE TOYRE) |
| 5 : PROGRAMME TAROT | (DENIS CASTELAIN) |
| 6 : CONDENSATION DE DONNEES EN BASE 256 | (JACQUES VAUCELLE) |

CHOC EN RETOUR:

- | | |
|----------------------------------|-----------------------|
| 1 : CORECTIONS SUR PRGM INTG/LEC | (JEAN-JACQUES MOREAU) |
|----------------------------------|-----------------------|

FICH/a 1300 BYTES SIZE 15
NECESSITE :X-FONCTIONS
OPTIONNELS : X- MEMOIRES, IMPRIMANTE, LECTEUR DE CASSETTES.

Les fichiers ASCII de la 41 semblent d'un abord assez rébarbatif, mais lorsqu'on les a domptés, ils se révèlent, combinés aux fonctions de traitements de chaînes alpha-numériques, capables de performances assez stupéfiantes; et le fait de pouvoir les chafer et de travailler (si on n'est pas pressé), sur l'équivalent d'un fichier de 130Koctets (ou 130xn si on possède n lecteurs de cassettes...) est assez agréable.

Le programme FICH/a est conçu pour exploiter ainsi un fichier de données.

Chaque fichier se compose de fiches, toutes au même format, le nombre de fiches dépendant à la fois de la quantité d'informations contenues dans chaque, et du nombre de X-MEMORYS présents.

Chaque fiche est constituée d'un nombre "l" de lignes, défini lors de la création du fichier, stocké dans le registre 00, et tel que $1 \leq l \leq 8$. (Les lignes sont numérotées de 0 à l-1).

Il est possible, lors de l'affichage d'une fiche -sur la 41, une vidéo ou une imprimante- soit de visualiser toute la fiche, soit seulement la partie des lignes déterminée à l'avance par les indicateurs binaires 0 à 1:

indicateur "i" armé: ligne "i" affichée;

indicateur "i" désarmé: ligne "i" ignorée.

De plus, si l'indicateur 9 est désarmé, seule la fiche en question sera visualisée; s'il est armé, toutes les fiches à partir de celle-ci seront visualisées, et ce conformément aux indicateurs 0 à 1.

Dans tous les cas, pour qu'un affichage ait lieu, il faut que l'indicateur 10 soit armé.

Pour que le programme tourne, il est impératif d'exécuter d'abord les instructions du début: il est donc préférable de l'enregistrer en mode départ automatique (11 armé)

CREATION DU FICHIER

LBL "b" crée un nouveau fichier en mémoire étendue et remplit son en-tête. Celle-ci doit obligatoirement comporter un nombre de lignes l. Le nom peut être différent de celui qu'il aura sur la cassette.

MODIFICATION DU FICHIER

LBL "D" ajoute le contenu du registre α comme nouvelle ligne en fin de fichier.

LBL "E" ajoute le contenu de α à la fin de la ligne désignée par le pointeur.

LBL "d" insère une ligne juste avant celle du pointeur.

LBL "e" ajoute un texte juste avant le pointeur

LBL "C" efface la ligne du pointeur

LBL "c" efface "n" caractères en commençant par celui désigné par le pointeur.

Remarque: Pour "D" et "E", il faut d'abord remplir le registre α , puis lancer le sous-programme;

Pour les autres, on lance d'abord le sous-programme, puis on répond aux questions.

Le sous-programme "Ø" permet de ne rien faire (Heureux programme !)

Le sous-programme " 14" est utilisé pour l'impression automatique d'étiquettes autocollantes (Tous les détails dans un prochain article). Il permet de sauter le nombre de lignes indiqué par le registre 14.

Ces trois registres sont remplis par les premières lignes du programmes, et peuvent être modifiés sans problème en cours de travail.

EXEMPLES:

```

GTO .000
RUN
TYPE LIGNE:
DEBUT FICHE?
- RUN → trace une ligne de ----- en début
FIN FICHE ? de fiche.
* RUN → trace une ligne de XXXXXXXX en fin
Nb. LIGNES/ 14? de fiche.
3 RUN → IBL " 14" fera avancer le papier de
PRET 3 lignes blanches.

```

IL EST OBLIGATOIRE DE REPONDRE A CES TROIS QUESTIONS.

CREATION D'UN NOUVEAU FICHER :

```

XEQ b
DIM. FICH=50?
RUN (1)
NOM FICH. EX?
ETEND RUN → Nom du fichier en mémoire étendue.
Nb. LIGN./FICHE?
4 RUN → Dimension de chaque fiche du fichier
NOM FICH. MASS?
MASS RUN → Nom du fichier sur la cassette.
NOM FICH. SUI.
MASS2 RUN → Nom du fichier sur cassette dans l
COMMENTAIRE? lequel le programme devra aller
EXCELLENT PROGRAMME RUN chercher les renseignements s'il
COMMENTAIRE? ne les trouve pas dans le fichier
RUN en cours.Ce nouveau fichier rem-
placera l'ancien en mémoire éten

```

due; le nom en mémoire étendu sera inchangé, et le contenu des registres modifié en conséquence.

Le commentaire est un texte sans importance stocké en fin de ligne Ø.

VISUALISATION DU FICHER

LBL "J" place le pointeur à la valeur contenue en X.

LBL "G" donne la valeur du pointeur.

(Ce pointeur est celui décrit page 22 du manuel du X-FONCTIONS)

LBL "F" affiche une ligne d'après sa position dans la fiche et le numéro de la fiche dans le fichier: (cf exemple)

LBL "a" affiche la ligne dont le numéro a été préalablement mis en X (a% est utilisable par programme)

LBL "B" fait défiler tout le fichier de puis la ligne dont le numéro est en X. (Attention aux indicateurs 00 à 1 et 9 et 10)

RECHERCHE DANS LE FICHER

LBL "A" cherche dans le fichier une chaîne identique à celle en X et l'affiche; commence à chercher à la ligne du pointeur. dès S'arrête à la 1^{ère} fiche trouvée si l'indicateur 9 est désarmé, donne toutes les fiches s'il est armé.

A% est utilisable par programme si la chaîne cherchée a été préalablement stockée dans les registres 4 et 5.

La recherche de la chaîne se fait sur l'ensemble des fichiers chaînés de la cassette. Par conséquence, IL Y A RISQUE DE DESTRUCTION DU FICHER EN MEMOIRE ETENDU PAR A% SI L'INDICATEUR 8 EST ARME (A désarme cet indicateur)

Ceci permet néanmoins de travailler en toute modestie sur 130 K...

TRANSFERT DES FICHERS

LBL "I" rappelle un fichier de la cassette à la mémoire étendue, et décode son en-tête.

I% se contente de décoder l'en-tête.

LBL "H" transfère un fichier de la mémoire étendue à la cassette.

Dans les deux cas, les noms des fichiers peuvent être différents.

H\$ évite d'entrer le nom du fichier en mémoire étendue (Ce nom doit figurer dans les registres 10 et 11)

H% est utilisable par programme si les registres 7 (dimension du fichier), 8; 9 (nom sur cassette), 10 et 11 ont été remplis au préalable.

Dans tout les cas, le fichier est sécurisé et vérifié.

H, H\$, ET H% PEUVENT DETRUIRE UN FICHER SUR CASSETTE NON SECURISE!

SI L'INDICATEUR 7 EST ARME, H% PEUT MEME/CARREMENT ASSASSINER UN PAUVRE FICHER QUI SE CROYAIT BETEMENT SECURISE SUR SA CASSETTE! (Cet indicateur est désarmé par H et H\$)

Alors, faites des copies, ou soyez très prudent si 7 est armé...

Les sous-programmes "0" ; "-" ; "X" ; " X" ; et " 14" (Les deux derniers labels commencent par un espace vide) servent à encadrer et décorer vos fiches lors de l'impression par LBL "B": si le nom du sous-programme est placé dans le registre 12, il sera effectué au début de chaque fiche; s'il est placé dans le registre 13, le travail sera effectué à la fin de chaque fiche.

STOCKAGE/RAPPEL SUR CASSETTE.

	XEQ A
NOM FICH. EXTEN?	RUN (1)
NOM FICH. MAS.?	RUN (1)
DIMENS. ? <50>	RUN (1)
UNSEC: 071	RUN
MASS ETEND, MASS	
	XEQ I
NOM F. MASSE?	RUN (1)
NOM F. ETEND?	RUN (1)
DIM. FICH=50?	RUN (1)
MASS, ETEND	

ENREGISTREMENT
sur cassette.

Si un fichier de même nom exist-
déjà sur cassette, il faut
armer 7 pour que le programme
puisse le désécuriser.

MASS : nom du fichier en
mémoire de masse
(cassette)
ETEND: nom du fichier en
mémoire étendue.

Le (1) indique que vous pouvez répondre à la question en fai-
sant directement "R/S" . Dans ce cas les anciennes valeurs
sont conservées.

RECHERCHE D'UNE DONNEE: La fiche sur MARIUS.

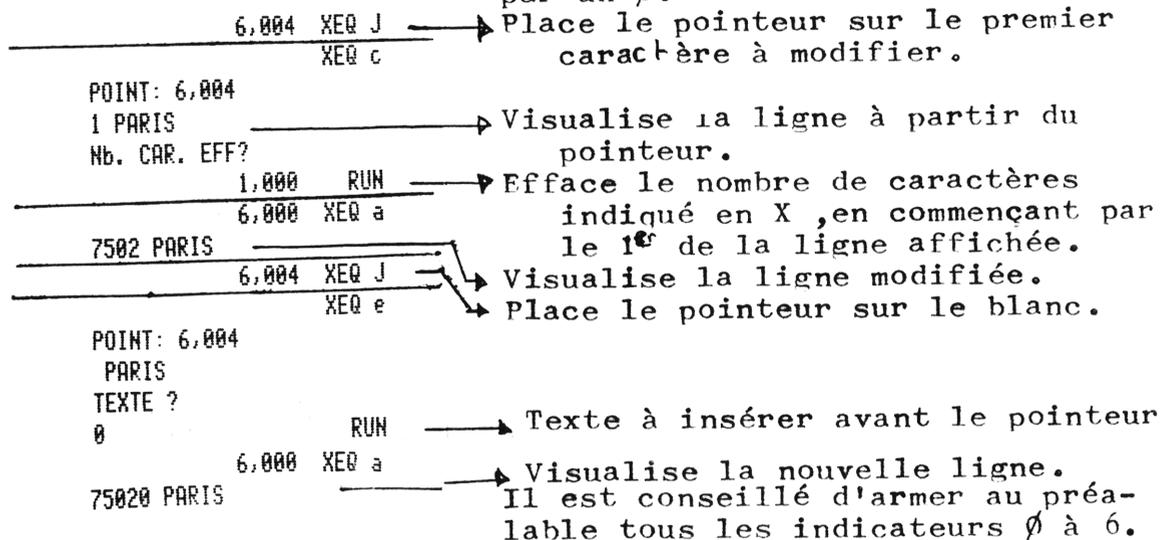
Tous les indicateurs
armés.

MARIUS
MARIUS XEQ A
MARIUS
LIGNE 8
MARIUS
2, RUE MINANT
13000 MARSEILLE
TEL: -
END OF FL

Indicateurs 1 et 2
désarmés

MARIUS XEQ A
MARIUS
LIGNE 8
MARIUS
TEL: -
END OF FL

MODIFICATION D'UNE LIGNE: Remplacement du 1 de 75021 PARIS
par un Ø.



01*LBL "FICH/a"
SIZE? 15 X>Y? PSIZE
127 X<>F CF 08 SF 09
SF 10 "TYPE LIGNE:"
AVIEW PSE"
"DEBUT FICHE?" AON
PROMPT ASTO 12
"FIN FICHE ?" PROMPT
ASTO 13 AOFF
"Nb. LIGNES/ 14?"
PROMPT STO 14 "PRET"
PROMPT

27*LBL b
RCL 07 FIX 0
"DIM. FICH=" ARCL X
"f?" PROMPT STO 07
"NOM FICH. EX?" AON
PROMPT ASTO 10 ASHF
ASTO 11 AOFF CLA
ARCL 10 ARCL 11 CRFLAS
0 SEEKPTA
"Nb. LIGN./FICHE" "f?"
PROMPT STO 00 CLA
FIX 0 CF 29 ARCL X
INSREC "*" ARCL 07
"f<" APPCHR AON
"NOM FICH. MASS?"
PROMPT ASTO 08 ASHF
ASTO 09
"NOM FICH. SUI." PROMPT
"f>" APPCHR

71*LBL 16
CF 23 "COMMENTAIRE?"
PROMPT FC? 23 CLA
APPCHR FS? 23 GTO 16
AOFF

81*LBL 04
RCL 00 2 -

85*LBL 05
RCLPT INT X>Y? RTN
"-- APPREC GTO 04

93*LBL E
APPCHR RTN

96*LBL I
APPREC RTN

99*LBL I
"NOM F. MASSE?" CF 23
AON PROMPT FC? 23
GTO 23 ASTO 08 ASHF
ASTO 09

109*LBL 23
"NOM F. a ?"
"NOM F. ETEND?" CF 23
PROMPT FC? 23 GTO 24
ASTO 10 ASHF ASTO 11

119*LBL 24
RCL 07 "DIM. FICH=" ARCL X
"f?" AOFF
PROMPT STO 07 CLA
ARCL 10 ARCL 11 SF 25
PURFL CRFLAS CLA
ARCL 08 ARCL 09 "f,"
ARCL 10 ARCL 11 AVIEW

140*LBL "I?"
GETAS 0 SEEKPT XEQ 10
0 SEEKPT GETREC ANUM
STO 00 0 SEEKPT "*" POSFL
SEEKPT GETREC ANUM
STO 07 0 SEEKPT "<"
POSFL SEEKPT GETREC
62 POSA AROT 60

168*LBL 01
ATOX PSE X<>Y X*Y?
GTO 01 ASTO 02 ASHF
ASTO 03 0 SEEKPT ">"
POSFL ,001 + XEQ 10

184*LBL 02
RCL 00 SEEKPT

187*LBL 03
XEQ 10 RCL 01 RCLPT
X<Y? GTO 03 RTN

194*LBL A
ASTO 04 ASHF ASTO 05
CF 08

199*LBL "A?"
RCL 00 SEEKPT

202*LBL 13
CLA ARCL 04 ARCL 05
AVIEW POSFL X<0?
GTO 12 INT STO Y
RCL 00 MOD - STO 06
"LIGNE " ARCL 06 AVIEW

219*LBL 11
RCL 06 RCL 00 + X<=Y?
GTO 13 X<>Y SEEKPT
RCL 00 MOD FC? IND X
GTO 14 XEQ 10 0
GTO 15

234*LBL 14
1

236*LBL 15
RCLPT + SEEKPT GTO 11

241*LBL 12
"FICH. SUIV.?" FC? 08
PROMPT CLA ARCL 10
ARCL 11 CLFL CLA
ARCL 02 ARCL 03 "f,"
AVIEW ARCL 10 ARCL 11
XEQ "I?" GTO "A?"

258*LBL F
"NUM. FICHE ?" PROMPT
RCL 00 *
"NUM. LIGNE/FICH"
"fE ?" PROMPT +

267*LBL a
268*LBL "a?"
SEEKPT

270*LBL 10
GETREC RCLPT INT
RCL 00 MOD FC? IND X
GTO 06 FS? 17 XEQ 19
FS? 17 GTO 10 XEQ 20
RCLPT 1 + INT SEEKPT
FC? 09 STOP RTN

291*LBL 19
FS? 10 GTO 21 FC? 10
GTO 21 ACA RTN

298*LBL 20
FS? 10 GTO 21 FC? 55
GTO 21 ACA PRBUF RTN

306*LBL 21
AVIEW PSE RTN

310*LBL "B?"
SEEKPTA

312*LBL B
SEEKPT

314*LBL 09
RCLPT RCL 00 MOD X=0?
XEQ IND 12 SF 25
XEQ 10 RCLPT RCL 00
MOD X=0? XEQ IND 13
FC? 25 RTN GTO 09

330*LBL 06
RCLPT INT 1 + SEEKPT
XEQ IND 14 RTN

338*LBL J
SEEKPT RTN

341*LBL G
RCLPT "POINT: " FIX 3
ARCL X AVIEW PSE RTN

349*LBL c
XEQ 18 "Nb. CAR. EFF?"
PROMPT DELCHR RTN

355*LBL C
XEQ 18 INT SEEKPT
"EFF. LIGNE?" PROMPT
DELREC RTN

363*LBL e
XEQ 18 "TEXTE ?" AON
PROMPT AOFF INSCR
RTN

371*LBL 18
XEQ G STO 01 XEQ a
RCL 01 SEEKPT RTN

378*LBL H
"NOM FICH. EXTEN" "f?"
AON CF 23 PROMPT
FC? 23 GTO "H?"
ASTO 10 ASHF ASTO 11

389*LBL "H?"
CF 07 "NOM FICH. MAS.?"
CF 23 PROMPT FC? 23
GTO 22 ASTO 08 ASHF
ASTO 09

399*LBL 22
AOFF RCL 07
"DIMENS.?" ARCL X
"f?" PROMPT STO 07
"UNSEC: 07?" PROMPT

409*LBL "H?"
CLA ARCL 08 ARCL 09
AVIEW SF 25 FS? 07
UNSEC RCL 07 SF 25
PURGE CREATE CLA
ARCL 10 ARCL 11 "f."
ARCL 08 ARCL 09 AVIEW
SAVERS CLA ARCL 08
ARCL 09 SEC VERIFY
RTN

435*LBL d
XEQ 18 "TEXTE ?" AON
PROMPT AOFF INSREC

442*LBL 18
XEQ G STO 01 XEQ a
RCL 01 SEEKPT RTN

449*LBL "0"
RTN

451*LBL "--"
FC? 55 RTN
"-----" ACA ACA
PRBUF RTN

459*LBL "*"
FC? 55 RTN
"*****" ACA ACA
PRBUF RTN

467*LBL " *"
FC? 55 RTN
" * * * * " ACA
" * " ACA PRBUF RTN

476*LBL " 14"
RCL 14

478*LBL 17
ADV DSE X GTO 17 RTN

483*LBL " "
ADV END

CATALOGUE :

LBL'FICH/a
LBL'I%
LBL'A%
LBL'a%
LBL'B%
LBL'H\$
LBL'H%
LBL'0
LBL'-
LBL'*
LBL' *
LBL' 14
LBL'
END 1300 BYTES

NUM. FICHE ? XEQ F
2,000 RUN
NUM. LIGNE/FICHE ?
2,000 RUN
13000 MARSEILLE

REPLISSAGE DU
FICHIER

GAYROCHE XEQ D
8, RUE TABAGA XEQ D
75021 PARIS XEQ D
TEL: 699 84 00 XEQ D
MARIUS XEQ D
2, RUE MINANT XEQ D
13000 MARSEILLE XEQ D
TEL: - XEQ D

LECTURE DU FICHIER
TOUS LES INDICA-
TEURS ARMES

0 XEQ B

4*50<MASS2>EXCELLENT PRO
GRAMME

GAYROCHE
8, RUE TABAGA
75021 PARIS
TEL: 699 84 00

MARIUS
2, RUE MINANT
13000 MARSEILLE
TEL: -

CF 01
CF 02
4 XEQ B

GAYROCHE
TEL: 699 84 00

MARIUS
TEL: -

CLAVIER EN MODE USER

Affiche ligne n°X	Crée l'en-tête	Efface n caractères	Insère une ligne avant celle du pointeur	Insère un texte avant le pointeur
a	b	c		e
: Cherche	: Visualise	: Efface la	: Place	: Place
: dans le fi-	: tout le	: ligne du	: nouvelle	: en fin
: chier	: fichier	: pointeur	: ligne en	: ligne
:	: depuis la	:	: fin fichier	: pointeur
: A	: ligne X	: C	: D	: E
: Visualise	: Donne la	: Envoie de	: Rappelle de	: Place le
: fiche n° X	: valeur du	: la mémoire	: la mémoire	: pointeur
:	: pointeur	: étendue à	: de masse à	: en X
:	:	: la mémoire	: étendue	:
: F	: G	: de masse	: I	: J

REGISTRES:

- ØØ : Nombre de lignes par fiche.
- Ø1 : Pointeur pour les corrections.
- Ø2 :
- Ø3 :
- Ø4 : Texte cherché par "A" dans le fichier.
- Ø5 :
- Ø6 : Numéro de la ligne trouvée par "A".
- Ø7 : Dimension du fichier.
- Ø8 :
- Ø9 : Nom du fichier sur cassette (mémoire de masse).
- 1Ø : Nom du fichier en mémoire étendue (X MEMORY).
- 11 :
- 12 : Sous-programme en début de fiche.
- 13 : Sous-programme en fin de fiche.
- 14 : Nombre de lignes blanches pour " 14".

Dans un prochain article, nous verrons comment exploiter ce programme soit pour gérer des stocks; soit pour se construire un carnet d'adresses informatisé, avec recherche de noms, impression d'étiquettes, indication en temps réel du coût d'une communication téléphonique en tenant compte automatiquement de la distance, de la durée et des taux de réductions.

A bientôt.

Jean-Claude BECKER

GESTION D'UN CARNET D'ADRESSES
grace au programme "FICH/a"

Cet ensemble de programmes (ADxxxxx, publiés dans le prochain JPC), sert, en association avec FICH/a, à effectuer automatiquement toutes les opérations faites habituellement à la main, et coûteuses en temps, dans la gestion d'un fichier d'adresses.

Cet ensemble était prévu au départ pour gérer le fichier du club, mais comme malheureusement (pour lui, pas pour nous), le 71 fait aussi bien et plus vite, il n'est donc pas utilisé dans cette tâche. Mais il peut être employé facilement, sous réserves de petites adaptations, dans la configuration vous concernant.

C'est pourquoi je ne vous livrerai pas ici, contrairement à mon habitude, des programmes très "travaillés", mais au contraire une simple "ossature", à charge pour vous de tout adapter à votre cas particulier...

En attendant les programmes d'exploitations le mois prochain, vous pouvez déjà entrer votre fichier en mémoire (C'est la partie la plus longue et la plus horrible, courage, après, ce sera un vrai régal...)

Pour que vos fichiers soient compatibles entre eux, il faut qu'ils soient tous au même format. Celui que je vous propose n'est pas plus mauvais qu'un autre, et a l'avantage d'être universel.

Nombre de lignes par fiche: 6, valeur stockée dans le registre 00.

1ère ligne: (n°0) En-tête de la personne concernée par la fiche. Pour une association, on peut y mettre par exemple le numéro d'inscription, la date d'inscription, la date d'échéance de la cotisation, éventuellement la date d'échéance d'un abonnement (cas de JPC au début de cette année). Pour un particulier, on peut y mettre la date de dernière mise à jour de la fiche.

2ème ligne: (n°1) Nom et prénom (Séparez le prénom du nom par au moins deux espaces blancs, et les autres mots (en cas de nom ou de prénom composé) par au plus un espace blanc, et il sera possible par la suite de commencer automatiquement par au choix le nom et le prénom.

3ème ligne: (n°2) Numéro et nom de la rue. Eventuellement, on peut y rajouter si nécessaire un nom de bâtiment, de résidence, etc., sur la même ligne du fichier, mais en intercalant des "retour chariot / avance papier (code ASCII 10) pour imprimer cela sur plusieurs lignes sur l'étiquette (Ne vous inquiétez pas, le programme d'édition d'étiquettes que je vous livrerai sait compter les lignes imprimées, et fait quand même une mise en page correcte, même si ce nombre de lignes varie!

4ème ligne (n°3) Code postal. On peut rajouter le nom du pays en séparant par un code 10

5ème ligne (n°4) Numéro de téléphone personnel

6ème ligne (n°5) Numéro de téléphone au travail.

Les numéros de téléphone doivent être entrés de préférence sous la forme n_XX_YYYYYY_posteZZZZ avec:
_: au moins un caractère alphanumérique
n: lettre code de la durée d'une unité (l'indication se trouve dans l'annuaire; P si pas de limitation de durée,
A une unité pour 120 secondes.

- B une unité pour 72 secondes,
 - C une unité pour 45 secondes,
 - D ^{un} une unité pour 24 secondes,
 - E une unité pour 112secondes ,
- (Valeurs pour le plein tarif)

Je vous conseille de bien respecter ces indications, car elles seront reprises dans des programmes d'exploitation de ces données.

XX: indicatif téléphonique du département.
 YYYYYYY numéro de téléphone proprement dit.
 ZZZZ éventuellement: numéro de poste.

Si vous entrez plusieurs nombres sur une même ligne de fichier ASCII, et si vous voulez récupérer leurs valeurs numériques par la suite; si vous avez la très précieuse fonction ANUMDEL (dans le module PANAME, ou le module EXTENDED I/O), aucun problème. Sinon, débutez-les toujours au même endroit, ou mettez devant un symbole ASCII non utilisé ailleurs.

Sur ce, bon courage et rendez-vous le mois prochain. Et vive la loi Informatique et Liberté!

Jean-Claude BECKER

EXEMPLE:

Dans le fichier ASCII

16.11.84	1ère ligne	(ligne n°0)
LULU LA BELLE	2ème ligne	
RESIDENCE LE PAVOIS*BATI	} 3ème ligne	
MENT G*22, ALLEE JIPECE		
75022 PARIS*FRANCE	4ème ligne	
P 1 6887171	5ème ligne	
*	6ème ligne	(ligne n°7)

(Les sont les codes 10)

RESULTAT sur imprimante 80 colonnes:

```

16.11.84
LULU LA BELLE
RESIDENCE LE PAVOIS
BATIMENT G
22, ALLEE JIPECE
75022 PARIS
FRANCE
P 1 6887171
*
```

TABLEAUX PRATIQUES -

En entreprenant la lecture de l'article "RESURRECTION D'UNE MEMOIRE MORTE" paru dans l'OI d'Avril 1983, on s'aperçoit que si l'on veut décoder beaucoup d'adresses de **MEM**, on doit remplir, entre autre, des pages de papier; aussi, je vous propose 4 tableaux reproductibles qui permettent de transcrire et d'analyser vos "TROUVAILLES".

Je joins avec un programme "DR" dérivant de "DCX" paru dans le même article, qui permet de décoder les registres d'état **e, d, c, b, a, t, Q, P** et accessoirementle registre "X".

Par contre, pour ce qui est de l'explication des résultats, se référer au chapitre et au tableau sur les registres d'états de "AU FOND DE LA HP 41 C" (2ème édition Janvier 1983).

CHERS AMIS,

Les tableaux ont été réduits 2fois. Ceux qui seraient éventuellement intéressés, doivent nous contacter, ils peuvent avoir ces tableaux au prix de la photocopie.

PHILIPPE

```

01+LBL "DR"
02 FS?C 00
03 GTO 01
04+LBL 02
05 "REG. e"
06 AVIEW
07 RCL e
08 XEQ 00
09 "REG. d"
10 AVIEW
11 RCL d
12 XEQ 00
13 "REG. c"
14 AVIEW
15 RCL c
16 XEQ 00
17 "REG. b"
18 AVIEW
19 RCL b
20 XEQ 00
21 "REG. a"
22 AVIEW
23 RCL a
24 XEQ 00
25 "REG. F"
26 AVIEW
27 RCL F
28 XEQ 00
29 "REG. Q"
30 AVIEW
31 RCL _
32 XEQ 00
33 "REG. P"
34 AVIEW
35 RCL †
36 XEQ 00
37 " " FIN"
38 AVIEW
39 STOP
00+LBL 01
41 "REG. X"
42 AVIEW
43 RCL X
44 XEQ 00
45 GTO 02
46+LBL 00
47 ENTER†
48 FIX 9
49 CLA
50 X<> [
51 "†++++"
52 X<> [
53 "†++"
54 X<> [
55 ARCL \
56 "†+"
57 X<> ]
58 "†+"
59 ARCL X

```

```

60 X<> †
61 X<> ]
62 X<> \
63 CLX
64 FIX 4
65 ARCL X
66 STO †
67 X<> ]
68 X<> \
69 X<> [
70 RDN
71 PROMPT
72 RTN
73 END

```

```

L5: 246,82,69,71,46,32,1
01,
L9: 246,82,69,71,46,32,1
00,
L13: 246,82,69,71,46,32,
99,
L17: 246,82,69,71,46,32,
98,
L21: 246,82,69,71,46,32,
97,
L25: 246,82,69,71,46,32,
127,
L29: 246,82,69,71,46,32,
81,
L33: 246,82,69,71,46,32,
80,
L37: 247,32,32,32,32,70,
73,78,
L41: 246,82,69,71,46,32,
88,
L51: 245,127,0,0,0,0,
L53: 243,127,0,0,
L56: 242,127,0,
L58: 242,127,0,

```

EXEMPLE

```

-1254.258-58 SF 00
XEQ "DR"
REG. X
1000000046494)
RUN
REG. e
100020:0107???)
RUN
REG. d
00000738048021
RUN
REG. c
204001691?91:1
RUN
REG. b
000000000031?0
RUN
REG. a
000000000000000
RUN
REG. F
10303130105000
RUN
REG. Q
9045472)205121
RUN
REG. P
000000000000000
RUN
FIN

```

```

: A
; B
< C
= D
> E
? F

```


18.10.1984

PRP "TAROT"

21:49 18.10

TOUT D'ABORD, JE
TIENS A REMERCIER TOUTES
LES PERSONNES QUI M'ONT
CONTACTE POUR M'APPORTER
DES ELEMENTS DE REPONSE,
A LA SUITE DU S.O.S. QUE
J'AVAIS LANCE DANS UN
PRECEDENT NUMERO DE JPC.

TAROT

SI VOUS AIMEZ JOUER AU
TAROT, VOTRE HP 41 ASSU-
RERA LA MARQUE A VOTRE
PLACE.

CE PGM NECESSITE LA
PRESENCE DU MODULE XF,
ET SOIT UN FICHER ASCII
DE 7 REG NOMME "TAROT"
SOIT UNE PLACE SUFFISAN-
TE POUR PERMETTRE AU PGM
DE CREER CE FICHER.

A LA QUESTION: "QUI
PREND ?" ENTREZ TOUT OU
PARTIE DU NOM (L'INITIAL
SUFFIT S'IL N'Y EN A PAS
DEUX IDENTIQUES.)

IL SUFFIT DE FAIRE RS

A: DEBUT DE CALCUL
B: REVOIR LES SCORES.

AMUSEZ-VOUS BIEN.

D. CASTELAIN

```
01+LBL "TAROT"  
CLRG . X<>F SF 27  
FIX 0 "TARO" SF 25 7  
CLFL FC?C 25 CRFLAS 5  
"NBRE JOUEURS" PROMPT  
X=Y? SF 05 STO 05  
DSE X STO 06 AON
```

```
22+LBL 01  
"NOM?" PROMPT APPREC  
DSE 05 GTO 01
```

```
28+LBL A  
AON "QUI PREND?"  
XEQ 06 STO 07 FS? 05  
GTO 04
```

```
35+LBL 05  
AOFF CF 22 FS? 05  
"LEURS" FC? 05 "SES"  
"- POINTS?" PROMPT  
FC? 22 GTO 05 STO 08  
RCL 06 FC? 05 GTO 07  
FC?C 06 DSE X * 2 /  
ST+ IND 09
```

```
56+LBL 07  
FC? 05 * ST+ IND 07  
RCL 06 1 E-3 * STO 05
```

```
64+LBL 00  
FS?C IND 05 GTO 02  
RCL 08 ST- IND 05
```

```
69+LBL 02  
ISG 05 GTO 00
```

```
72+LBL 0  
CLX SEEKPT
```

```
75+LBL 03  
GETREC "- " RCLPT RND  
ARCL IND X AVIEW BEEP  
PSE RCL 06 X=Y?  
GTO 03 STOP GTO A
```

```
89+LBL 04  
"AVEC QUI?" XEQ 06  
STO 09 RCL 07 X=Y?  
SF 06 GTO 05
```

```
97+LBL 06  
PROMPT 0 SEEKPT POSFL  
X<0? GTO 08 SF IND X  
RTN
```

```
106+LBL 08  
"NON" AVIEW BEEP  
"DEBILE" AVIEW BEEP  
PSE GTO A AVIEW .END.
```

CHERS AMIS 41CISTES,

Ce mois nous allons voir un petit programme qui n'a aucune utilité en soi. Il était destiné à l'origine à un membre qui a besoin de prendre des mesures sur le terrain en utilisant moins de mémoire que prévu. Il vous permettra une approche de l'utilisation du buffer du module HPIL devel.

CONDENSATION DE DONNEES EN BASE 256

Ce programme vous permet de condenser en base 256 et dans le buffer du défini par la fonction 'BSIZE' un système de points à 7 paramètres.

Les restrictions de ce programme, sont, d'une part, des nombres positifs et d'autre part, des nombres entiers. Mais, je pense que chacun sera en mesure de changer le programme en fonction de ses besoins.

```
TYPE DEMO256.41L
` 20:53 24.11
01LBL "DEMO256"
02 CF 29
03 CF 28
04 SIZE?
05 5
06 X>Y?
07 PSIZE
08 FIX 0
09 " * DEMO *"
10 AVIEW
11 TONE 3
12 TONE 3
13 TONE 3
14 PSE
15 PSE
16 " 3 POINTS"
17 AVIEW
18 PSE
19 PSE
20 " SERONT"
21 AVIEW
22 PSE
23 PSE
24 " SAISIS"
25 AVIEW
26 PSE
27 PSE
28 " AU CLAVIER"
29 AVIEW
30 PSE
31 PSE
32 TONE 3
33LBL 01
34 " COMPOSEZ"
35 AVIEW
36 PSE
37 PSE
38 " CHAQUE"
39 AVIEW
40 PSE
41 PSE
42 "PARAMETRE : "
43 AVIEW
44 PSE
45 PSE
46 "ABSOLU, ENTIER"
47 AVIEW
48 PSE
49 PSE
50 PSE
51 PSE
52 0
53 STO 00
54 1,003
55 STO 02
56 150
57 BSIZEX
58 AIPT
```

59LBL 02	125 AVIEW	
60 " * * *"	126 PSE	
61 AVIEW	127 PSE	
62 PSE	128 "PARAMETRES :"	
63 TONE 5	129 AVIEW	
64 TONE 5	130 PSE	
65 PSE	131 PSE	
66 "POINT NUM : "	132 PSE	
67 ARCL 02	133 PSE	
68 AVIEW	134 1,003	
69 PSE	135 STO 02	
70 PSE	136 0	
71 PSE	137 PT=	
72 RCL 00	138LBL 12	
73 3	139 3	
74 +	140 BUF-XB	
75 PT=	141 1 E7	
76 0	142 /	191 0
77 STO 03	143 STO 03	192 BSIZEX
78 1,007	144 1,007	193 "APPUYEZ R/S"
79 STO 04	145 STO 04	194 AVIEW
80LBL 04	146LBL 14	195 PSE
81 10	147 RCL 03	196 PSE
82 ST* 03	148 10	197 " POUR"
83 CLX	149 *	198 AVIEW
84 "INTRO "	150 FRC	199 PSE
85 ARCL 04	151 STO 03	200 "RECOMMENCER"
86 " ?"	152 LASTX	201 TONE 7
87 TONE 9	153 BUF-XB	202 PROMPT
88 PROMPT	154 CLA	203 GTO 01
89 PT?	155 ARCL 02	204 .END.
90 X<>Y	156 " "	
91 X-BUF	157 ARCL 04	
92 PT?	158 " "	
93 RCL Z	159 ARCL X	
94 -	160 AVIEW	
95 ST+ 03	161 TONE 3	
96 ISG 04	162 PSE	
97 GTO 04	163 PSE	
98 PT?	164 PSE	
99 X<> 00	165 ISG 04	
100 PT=	166 GTO 14	
101 RCL 03	167 ISG 02	
102 X-BUF	168 GTO 12	
103 ISG 02	169 BEEP	
104 GTO 02	170 BEEP	
105 BEEP	171 RCL 00	
106 BEEP	172 21	
107 " "	173 /	
108 ARCL 00	174 "ENCOMBREMENT"	
109 " OCTETS"	175 AVIEW	
110 AVIEW	176 PSE	
111 PSE	177 PSE	
112 PSE	178 " PAR POINT :"	
113 "MEMORISENT"	179 AVIEW	
114 AVIEW	180 PSE	
115 PSE	181 PSE	
116 PSE	182 "ENVIR. "	
117 "VOS 3 POINTS"	183 ARCL X	
118 AVIEW	184 "REG"	
119 PSE	185 AVIEW	
120 PSE	186 PSE	
121 PSE	187 PSE	
122 TONE 3	188 PSE	
123 TONE 3	189 BEEP	
124 " VOICI VOS"	190 BEEP	

XEQ "DEMO256"

* DEMO *

3 POINTS

SERONT

SATIS

AU CLAVIER

COMPOSEZ

CHAQUE

PARAMETRE :

ABSOLU, ENTIER

POINT NUM : 1

INTRO 1 ?

7 RUN

INTRO 2 ?

23 RUN

INTRO 3 ?

8 RUN

INTRO 4 ?

987 RUN

INTRO 5 ?

2 RUN

INTRO 6 ?

3 RUN

INTRO 7 ?

8 RUN

POINT NUM : 2

INTRO 1 ?

10 RUN

INTRO 2 ?

720 RUN

INTRO 3 ?

96 RUN

INTRO 4 ?

3 RUN

INTRO 5 ?

985 RUN

INTRO 6 ?

789 RUN

INTRO 7 ?

123 RUN

POINT NUM : 3

INTRO 1 ?

456 RUN

INTRO 2 ?

123 RUN

INTRO 3 ?

456 RUN

INTRO 4 ?

789 RUN

INTRO 5 ?

741 RUN

INTRO 6 ?

8520 RUN

INTRO 7 ?

963 RUN

40 OCTETS	1 ,1: 7
MEMORISENT	1 ,2: 23
VOICI VOS	1 ,3: 8
PARAMETRES	1 ,4: 987
	1 ,5: 2
	1 ,6: 3
	1 ,7: 8
	2 ,1: 10
	2 ,2: 720
	2 ,3: 96
	2 ,4: 3
	2 ,5: 985
	2 ,6: 789
	2 ,7: 123
	3 ,1: 456
	3 ,2: 123
	3 ,3: 456
	3 ,4: 789
	3 ,5: 741
	3 ,6: 8520
	3 ,7: 963

ENCOMBREMENT
PAR POINT :
ENVIR. 2 REG
APPUYEZ R/S
POUR
RECOMMENCER

PRBYTES				
00 16				
01 248	ESPACE OCC. PT N°1			
02 47	1112111 = 8octets			
03 07	POINT NUM. 1			
04 23				
05 08				
06 03) 937		
07 219				
08 02				
09 03				
10 08				
11 18				
12 123	ESPACE OCC. PT N°2			
13 85	1211221 = 10 octets			
14 10	POINT NUM. 2			
15 02) 720		
16 208				
17 96				
18 03				
19 03) 937		
20 217				
21 03) 789		
22 21				
23 123				
24 32	ESPACE OCC PT N°3			
25 97	2122222 = 13 octets.			
26 238				
27 01	POINT NUM. 3			
28 200) 456		
29 123				
30 01) 456		
31 200				
32 03) 789		
33 21				
34 02) 741		
35 229				
36 33) 8520			
37 72				
38 03) 963			
39 195				
40 00				

Pour vous aider dans l'étude de ce PRGM, vous trouverez ci-dessous quelques indications sur les différentes parties de ce programme.

LIGNES 001 à 051: Initialisation de la 41 et indications de ce qui va être fait; choix de 3 paramètres entiers et positifs.

LIGNES 052 à 065: définit le buffer utilisé pour la condensation des points en base 256. Le 'LBL 02' demande la définition des points. Le 'LBL 04' demande les paramètres des points.

LIGNE 073: démarre le stockage à la 3° ligne du BUFFER pour le premier point.

LIGNE 098: pour le premier point, stocke 1112111 (à partir de la position du pointeur en 0). Stocke donc le nombre d'octets utilisé par le point numero 1; 1+1+1+2+1+1+1= 8octets. pour le point n°2, stocke 1211221 (à partir de la position du pointeur en 11. Pour le point n°3, stocke 2122222 (à partir de la position du pointeur en 24).

LIGNES 124 à 168: rappel des points définis précédemment.

Le 'LBL 12' rappelle chaque points.
Le 'LBL 14' rappelle chaque paramètre sous la forme: "i,j : n" ou
i = numéro du point,
j = numéro du paramètre,
n = valeur du paramètre.

LIGNES 171 à 204: définit la place utilisée non seulement par chaque point, mais aussi par leur encombrement. Demande ensuite l'introduction de nouveaux points.

REMARQUES: Si vous avez besoin de plus de trois point et (ou) de plus de 7 paramètres, vous devez changer les constantes:
lignes 54 & 134 = nombre de points,
lignes 78 & 144 = nombre de paramètres,
ligne 56 = taille du buffer.

Les nombres sont stockés dans le buffer sous la forme modulo 256. Si le pointeur est en 0, 10 X-BUF stocke 10 à la ligne 0. 456 X-BUF stocke dans le buffer en ligne 0: 1 (456/256=1) et en ligne 1: 200 (456 MOD 256 = 200). 8520 X-BUF stocke dans le buffer en ligne 0: 33 (8520/256=33) et en ligne 1: 72 (8520 MOD 256 = 72)... En ce qui concerne le nombre d'octets, 2122222 X-BUF stocke en ligne 0: 32, en ligne 1: 97 et en ligne 2: 238, (soit, 2122222 MOD 256 = 238, 2122222/256 = 8289, 8289 MOD 256 = 97 et 8289/256 = 32). Soit en base 256, nous avons: 10 = 10; 456 = (1 x 256) + 200 et 2122222 = (32 x 256²) + (97 x 256) + 238.

Pour ceux qui ne possèdent pas le module HPIL DEVEL mais qui désirent tout de même étudier ce programme, voici les fonctions spéciales à ce module:

AIPT	arme le mode d'auto incrémentation du pointeur. Le pointeur utilisé par le buffer, s'auto incrémente à chaque utilisation.
BSIZEX	donne la taille du buffer égale à la valeur contenue dans le registre X.
BUF-XB	convertie la valeur binaire contenue dans le buffer en décimal et la place en X.
PRBYTES	imprime les bytes du buffer dans le format hexa.
PT=	positionne le pointeur selon la valeur contenue dans le registre X
PT?	retourne la position du pointeur en X
X-BUF	stocke la valeur de X dans le buffer en binaire à partir de la position du pointeur.

Bonne étude à tous

JACQUES VAUCELLE

```
R00= 0,000000000000
R01= 0,01234122980
R02= 0,99518722000
R03= 0,02853138863
R04= 0,97472855600
R05= 0,04427743882
R06= 0,93827455200
R07= 0,05929858492
R08= 0,88641552700
R09= 0,07334648141
R10= 0,82000198600
R11= 0,08619016153
R12= 0,74012419160
R13= 0,09761865210
R14= 0,64809365190
R15= 0,10744427010
R16= 0,54542147140
R17= 0,11550566810
R18= 0,43379350760
R19= 0,12167047290
R20= 0,31504267970
R21= 0,12583745630
R22= 0,19111886750
R23= 0,12793819530
R24= 0,06405689286
```

Chers amis ,

l'auteur

Suite à un coup de fil de l'article du JPC du mois dernier 'INTG/LEC', de JEAN JACQUES MOREAU, un certain nombre de modifications doivent être faites:

Vous trouverez ci-contre le contenu des registres R00 à R24 avec une précision de 11 chiffres après la virgule, et ci-dessous certains changements à apporter au programme INTG.

```
01 *LBL "INTG"
02 *e" ! (241-126)
03 CLX
04 SF 25
05 SEEKPTA
06 1.024
07 FC? 25
08 RDTAX
09 FS?C 25
10 GETAX
11 *F-NAME ?*
12 AON
13 STOF
14 AOFF
15 ASTO M
```

16 STOP	62 CLX
17 *LBL a	63 STO 00
18 X<Y	64 SF 06
19 STO Z	65 *LBL 13
20 -	66 24
21 2	67 STO P
22 /	68 *LBL 14
23 STO 0	69 RCL IND P
24 +	70 FS? 06
25 STO N	71 CHS
26 CLX	72 RCL a
27 STO 00	73 +
28 SF 06	74 1/X
29 *LBL 01	75 STO 0
30 24	76 ST+ X
31 STO P	77 RCL N
32 SIGN	78 +
33 CHS	79 DSE P
34 ST* 0	80 XEQ IND M
35 *LBL 09	81 RCL IND P
36 RCL IND P	82 RCL 0
37 RCL 0	83 X↑2
38 *	84 *
39 RCL N	85 *
40 +	86 ST+ 00
41 XEQ IND M	87 DSE P
42 DSE P	88 GTO 14
43 RCL IND P	89 FS?C 06
44 *	90 GTO 13
45 ST+ 00	91 RCL 00
46 DSE P	92 ST+ X
47 GTO 09	93 RTN
48 FS?C 06	94 *LBL "LEC"
49 GTO 01	95 "α" ! (242-4-128)
50 RCL 0	96 STO a
51 RCL 00	97 CLX
52 *	98 RCL M
53 RTN	99 X<> d
54 *LBL b	100 *NAME = "
55 RCL M	101 AVIEW
56 "0α" ! (242-16-09)	102 CLA
57 X<> M	103 STOP
58 STO a	104 X<> d
59 SIGN	105 CLX
60 -	106 RCL a
61 STO N	107 ASTO M
	108 GTO IND M
	109 END

D O S S I E R

L E P R O T O C O L E

D U M O D U L E

H P I L (4 1 / 7 0)

I N T R O D U C T I O N

Le dossier que nous vous proposons aujourd'hui concerne l'HPIL. Vaste domaine que celui-ci ! Ne vous attendez pas à des conseils du genre "Comment afficher un tableau sur votre imprimante". Le sujet est bien trop vaste pour nous diversifier, aussi nous ne pourrions que donner un faible aperçu de ce qu'est le Protocole HPIL. Une étude détaillée suffirait à remplir chaque JPC durant au moins une année !

Le problème, qui se pose à toute personne qui veut un peu comprendre comment fonctionne l'HPIL, est le manque de documentation. La référence s'appelle ici "HPIL Interface Specification". Vous trouverez cet ouvrage dans le "kit convertisseur", référencé HP82166C. A titre indicatif, il contient aussi le nécessaire (matériel et documentation) pour développer des périphériques HPIL vraiment dignes de ce nom, telle l'interface vidéo Mountain Computer. Ne vous faites cependant pas d'illusions: il faut de solides connaissances en informatique et en électronique, plus la modique somme de 4000 à 5000 FF...

Ce que vous allez lire dans les pages suivantes était au départ un petit aide-mémoire, puis est devenu petit à petit un dossier pour JPC. C'est donc un document technique, en français, qui vous permettra d'appréhender l'HPIL. Répétons encore que nous n'avons ni la place, ni le temps de développer le sujet comme il le mériterait. Ce sera pour plus tard, et peut être pas dans ces colonnes...

Il ne faut pas confondre message HPIL et séquences d'échappement ou de contrôle. Un message HPIL est envoyé par le contrôleur du système aux appareils de la boucle. Vous, vous ne voyez pas ces messages. Par exemple, vous ne voyez pas AAU AAD1 quand vous exécutez RESTORE IO sur HP75 ou HP71. Vous ne voyez pas non plus REN quand vous faites REMOTE sur HP41. Ce sont ces messages, pourtant, qui sont le mécanisme fondamental de la communication sur la boucle.

A l'opposé, nous avons les séquences d'échappement et de contrôle. Ces séquences ne sont pas propres à l'HPIL, mais sont plus

générales, et sont utilisées par la plupart des ordinateurs pilotant des périphériques. Malheureusement, ces séquences ne sont absolument pas standard, et peuvent varier d'une imprimante à l'autre, d'une interface vidéo à l'autre, etc... Notons aussi la place particulière occupée par le HPGL (Hewlett-Packard Graphic Language), permettant de contrôler de manière très performante les tables traçantes d'origine HP. Ceci n'est pas propre à l'HPIL.

Vous trouverez donc dans les pages suivantes:

- Un résumé des messages HPIL en français. Ces messages sont accessibles séparément avec le module Extended I/O ou HPIL Dev du HP41, avec le IOUtilities ou le module APRIL du HP75, ou tout simplement avec le HP71 et le programme "SEND". C'est un résumé expliqué de l'HPIL Interface Specification (et en Français, s'il vous plaît !).
- Ensuite, vous avez un petit aide mémoire, ainsi que la syntaxe HP71 correspondant à chaque commande. Cette syntaxe est très proche de celles du HP75 et du HP41, donc vous n'aurez pas de mal à l'adapter pour votre machine. Cette liste est en anglais, car on conserve ainsi la signification mnémotechnique de l'abréviation employée.
- Enfin, pour les possesseurs de HP71 + HPIL, nous avons fait le programme SEND. Il peut être intéressant de l'étudier, même si vous n'avez pas d'HPIL, même si vous avez un HP75, et même si vous n'avez pas l'intention de pénétrer la signification intrinsèquement fondamentale de l'HPIL...

Nous espérons vous fournir, par l'intermédiaire de ce dossier, quelques éléments pour commencer l'étude du protocole HPIL. Si certains points vous résistent, n'oubliez pas qu'il existe une rubrique "SOS". Si vous avez tout compris, ce que nous espérons, n'oubliez pas que les colonnes de JPC vous sont ouvertes !

Joyeuse Programmation et Compréhension !

Michel MARTINET et Pierre DAVID

: RESUME DES MESSAGES HPIL :

GROUPE DONNEES

- DAB (0 à 255) - Octet de donnée
- DATA Byte
 - Les octets de donnée sont à la base de la communication sur la boucle d'interface. Ce sont généralement des codes ASCII. L'instruction SEND DATA du HP71 contient plusieurs ordres DAB.
- END (0 à 255) - Octet de fin
- END byte
 - Les octets de fin sont identiques aux DAB à l'exception d'un bit, ce qui permet d'identifier une fin d'enregistrement. Ne pas confondre avec les EDT.

GROUPE ORDRE

- NUL
- Commande nulle
 - NUL command
 - Cette instruction sert à vérifier la continuité de la boucle. Aucune opération n'est effectuée.
- GTL
- Local
 - Go To Local
 - Les récepteurs passent à l'état local.
- SDC
- Appareil récepteur libre
 - Selected Device Clear
 - Place les récepteurs dans leur condition initiale, généralement la condition après la mise sous tension.
- PPD
- Déconfiguration reconnaissance parallèle
 - Parallel Poll Disable
 - Les récepteurs ne réagissent plus à la reconnaissance parallèle. Ceci est sans signification à l'heure actuelle, le HP71 étant capable de répondre, mais non pas d'exploiter la reconnaissance parallèle (LEX ?).

- GET
 - Déclenchement
 - Group Execute Trigger
 - Cet ordre permet le déclenchement simultané d'une action sur tous les récepteurs y répondant. Par exemple: déclenchement de plusieurs mesures au même instant t. Sans signification pour le HP71 en tant que périphérique.

- ELN
 - Récepteurs validés pour pas prêt
 - Enable Listener Not ready
 - Les récepteurs sont autorisés par le contrôleur à interrompre l'émetteur durant une transmission de données.

- PPE0 (0 à 7)
 - Validation reconnaissance parallèle
- et
 - Parallel Poll Enable
- PPE1 (0 à 7)
 - Autorise les récepteurs à modifier les messages IDY conformément au PPE reçu. La syntaxe utilisée depuis quelques années est : PPE 0 à 15. Le programme SEND utilise cette syntaxe.

- DDL (0 à 31)
 - Ordre récepteur dépendant
 - Device Dependant Listener
 - La réponse dépend des caractéristiques des différents périphériques. Voir à ce sujet "K71" (JPC septembre 84), "VOLABEL" (JPC octobre 84) pour plus de détails.

- DDT (0 à 31)
 - Ordre émetteur dépendant
 - Device Dependant Talker
 - Message jumeau du précédent...

- NOP
 - Commande non-opérante
 - No Operation
 - Utilisée par le contrôleur pour resynchroniser la boucle à la suite de demandes asynchrones.

- LLD
 - Blocage local
 - Local LockOut
 - Inhibe les contrôles locaux, tels que boutons poussoirs permettant la remise en mode local. Sans signification pour le HP71, celui-ci ne disposant pas de bouton poussoir !

- DCL
 - Appareils libres
 - Device CLear
 - Tous les périphériques sont remis dans leur état initial. (Equivalent à un SDC généralisé).

- EAR
 - Validation demande asynchrone
 - Enable Asynchronous Request
 - Ou l'art de semer la pagaille sur une boucle HPIL, et d'attendre jusqu'à ce que NOP débloque la situation... En clair, permet à tous les périphériques d'émettre simultanément leur IDY ??

- IFC - Interface libre
 - InterFace Clear
 - Permet de prendre d'autorité le contrôle de la boucle, et supprime tous les états récepteurs et émetteur. Le HP71 ne peut, par l'ordre SEND IFC, reprendre le contrôle.
- REN - Validation télécommande
 - Remote ENable
 - REN valide le mode télécommande sur la boucle. Les périphériques après réception sont toujours en mode local mais ils peuvent être désormais télécommandés dès qu'ils sont adressés récepteurs (LAD 0 à 30). Leurs commandes locales (claviers, interrupteurs, avance papier etc.) sont alors inactives. Tous les périphériques n'ont pas la possibilité d'être télécommandés. Se référer aux caractéristiques de chacun d'eux.
- NRE - Annulation télécommande
 - Not Remote Enable
 - NRE déconfigure la télécommande de la boucle: supprime les effets de REN et LLO.
- AAU - Déconfiguration adresse auto
 - Auto Address Unconfigure
 - Cette instruction est envoyée par le contrôleur à tous les périphériques. Elle déconfigure l'adressage de la boucle et est généralement envoyée avant d'opérer un nouvel adressage. Les périphériques possédant une adresse par défaut activent cette adresse. Le HP71 répond à l'adresse 21. Voir à ce sujet le programme POLYCOPY paru dans JPC octobre 84.
- LPD - Extinction de la boucle
 - Loop Power Down
 - Cette instruction est utilisée par le contrôleur pour éteindre les périphériques qui en ont la possibilité. Le lecteur de cassettes et l'imprimante HP82162 sont des exemples de périphériques répondant à LPD.
- LAD (0 à 30) - Adresse récepteur
 - Listen Address
 - L'adresse est contenue dans l'instruction. Si le numéro correspond à son adresse, le périphérique se met dans l'état récepteur.
- UNL (LAD31) - Non récepteur
 - UNListen
 - Cette instruction remet tous les récepteurs actifs de la boucle en état d'attente.

- TAD (0 à 30) - Adresse émetteur
 - Talk Address
 - L'adresse est contenue dans l'instruction. Si le numéro correspond à son adresse, le périphérique se met dans l'état émetteur. Si l'adresse ne correspond pas, il efface un éventuel état émetteur antérieur.
- UNT - Non émetteur
 - UNTalk
 - Cette instruction annule l'état émetteur de la boucle.
- SAD (0 à 30) - Adresse secondaire
 - Secondary Address
 - Cette instruction permet de valider la mise en état récepteur ou émetteur des périphériques répondant à l'adressage étendu. Dans ce cas, les LAD ou TAD précédant définissent l'adresse primaire et SAD l'adresse secondaire.
- ISA - Adresse secondaire illégale
 - Illegal Secondary Address
 - Cette instruction ne fait pas partie des spécifications HP-IL mais est écrite pour assurer une homogénéité avec les instructions du groupe Auto-Adressage.

 GROUPE PRET

- RFC - Prêt pour commande (ordre)
 - Ready For Command
 - Cette instruction suit systématiquement l'envoi d'une commande.
- ETO - Fin de transmission - ok
 - End of Transmission - Ok
 - ETO est envoyée par l'émetteur à la suite d'une transmission de données pour signaler au contrôleur qu'il a terminé sa transmission.
- ETE - Fin de transmission - erreur
 - End of Transmission - Error
 - L'instruction ETE est envoyée par l'émetteur pour signaler au contrôleur une erreur dans la transmission de données. Il arrête donc la transmission.
- NRD - Pas prêt pour données
 - Not Ready for Data
 - Cette instruction est émise par le contrôleur ou par un récepteur si le contrôleur le lui permet pour arrêter une transmission de données.

- SDA - Envoi Données
 - Send DATA
 - Cette instruction est utilisée par le contrôleur pour informer l'émetteur de la boucle qu'il peut commencer l'envoi de données.
- SST - Envoi état
 - Send STATUS
 - A la réception de SST, l'émetteur envoie son (ou ces) octet(s) d'état.
- SDI - Envoi identification périphérique
 - Send Device Identification
 - SDI est envoyée par le contrôleur à destination de l'émetteur. Si l'émetteur ne répond pas à SDI, il renvoie le message au contrôleur, autrement il le remplace par le premier octet de sa chaîne d'identification. Lorsque le premier octet lui est retourné, il envoie le second, etc.. La chaîne est généralement suivie de (CR)(LF) et il termine sa transmission par ETO.
- SAI - Envoi identification appareil
 - Send Accessory Identification
 - Ems par le contrôleur, destiné à l'émetteur. A la réception de SAI, l'émetteur le remplace par son octet d'identification appareil et termine la transmission par ETO.
- TCT - Prendre Contrôle
 - Take Control
 - A la réception de TCT, l'émetteur devient immédiatement contrôleur. Il ne renvoie pas l'instruction et la remplace par une commande. Le contrôleur d'origine reste en mode contrôle jusqu'à la réception de cette commande: c'est le seul moyen qu'il ait pour savoir si un autre contrôleur se trouve sur la boucle. A ce moment, il abandonne son rôle de contrôleur.
- AAD (0 à 30) - Adresse auto
 - Auto Address
 - Adressage simple (30 périphériques). Cette instruction est envoyée à tous les périphériques de la boucle. Chacun prend comme adresse le code reçu et l'incrémente avant retransmission.
- IAA - Adresse auto illégale
 - Illegal Auto Address
 - Cette instruction est le résultat de l'incrémementation de AAD lorsque trop de périphériques se trouvent en même temps sur la boucle. Lorsque un périphérique reçoit IAA, il garde son adresse par défaut s'il en a une.

- AEP (0 à 30) - Primaire étendue auto
 - Auto Extended Primary
 - Après avoir assigné leurs adresses secondaires à un groupe de périphériques, le contrôleur envoie AEP n pour leur assigner une même adresse primaire.
- IEP - Primaire étendue illégale
 - Illegal Extended Primary
 - Cette instruction n'est autre que AEP 31. Le contrôleur n'a pas à l'envoyer sur la boucle puisqu'elle est illégale. Elle fait partie du protocole pour qu'il n'y ait pas de trou.
- AES (0 à 30) - Secondaire étendue auto
 - Auto Extended Secondary
 - Chaque appareil prend comme adresse secondaire le code reçu et l'incrémente avant retransmission.
- IES - Secondaire étendue illégale
 - Illegal Extended Secondary
 - Cette instruction est le résultat de l'incrémentation de AES. Lorsque l'instruction atteint la valeur 31, elle est appelée IES et n'est plus acceptée par les périphériques.
- AMP (0 à 30) - Primaire multiple auto
 - Auto Multiple Primary
 - Adressage multiple (30 périphériques). D^e AAU: sans signification à l'heure actuelle pour le HP71.
- IMP - Primaire multiple illégale
 - Illegal Multiple Primary
 - Cette instruction est le résultat de l'incrémentation de AMP lorsque 30 ou trop de périphériques se trouvent sur la boucle. Sans signification pour le HP71.
- ZES (AES0) - Secondaire étendue zéro
 - Zero Extended Secondary
 - Instruction utilisée pour l'assignation de l'adresse secondaire des périphériques répondant à l'adressage multiple. Sans signification pour le HP71.

 GROUPE IDENTIFICATION

- IDY - Identification
 - IDentify
 - Cette instruction est envoyée par le contrôleur à tous les périphériques pour savoir si un ou plusieurs d'entre eux demandent service.

PSEUDO-MESSAGES

MTA - Adresse émetteur
- My Talk Address
- Ce n'est pas à proprement parler un message, mais le message UNT suivi de l'activation de la fonction Emetteur. Ceci permet au HP71 de se déclarer comme l'émetteur de la boucle.

MLA - Adresse récepteur
- My Listen Address
- Encore plus simple que le précédent: active simplement la fonction Récepteur. A noter que celle-ci détruit la fonction Emetteur si elle existait auparavant.

DSR, ESR, et ISR.

- Ces messages sont un peu particuliers dans le sens où ils convoient une information importante: la demande de service. En fait, ils sont des DAB, END et IDY émis lorsque le bit numéro 6 défini par REQUEST est à 1.



à la recherche de
nouveaux horizons par l'HP71

Christelle G. 13 ans

LISTE DES COMMANDES HPIL

GRUPE DONNEES

```

-----
: DAB (0 à 255)  : Data Byte                : OUTPUT ;LOOP; CHR$(0 à 255):
: DSR (0 à 255)  : Data byte - Service Request : OUTPUT ;LOOP; CHR$(0 à 255):
:                :                            : si BIT(REQUEST,6)=1.      :
: END (0 à 255)  : END byte                  : SEND END 0 à 255         :
: ESR (0 à 255)  : End byte - Service Request : SEND END 0 à 255         :
:                :                            : Si BIT(REQUEST,6)=1.     :
-----

```

GRUPE ORDRE

```

-----
: ACG                : Addressed Command Group   :
-----
: NUL                : NUL command               : SEND CMD                  :
: GTL                : Go To Local               : SEND GTL                  :
: SDC                : Selected Device Clear     : SEND SDC                  :
: PPD                : Parallel Poll Disable     : SEND CMD 4                :
: GET                : Group Execute Trigger     : TRIGGER                   :
: ELN                : Enable Listener Not ready : SEND CMD 15               :
: PPE0 (0 à 7)      : Parallel Poll Enable 0    : SEND CMD 128 à 135       :
: PPE1 (0 à 7)      : Parallel Poll Enable 1    : SEND CMD 136 à 143       :
: DDL (0 à 31)      : Device Dependent Listener : SEND DDL 0 à 31          :
: DDT (0 à 31)      : Device Dependent Talker   : SEND DDT 0 à 31          :
-----
: UGC                : Universal Command Group   :
-----
: NOP                : No Operation command      : SEND CMD 16               :
: LLD                : Local Lockout             : LOCAL LOCKOUT             :
: DCL                : Device Clear              : CLEAR                     :
: PPU                : Parallel Poll Unconfigure  : SEND CMD 21               :
: EAR                : Enable Asynchronous Request : SEND CMD 24               :
: IFC                : InterFace Clear          : SEND IFC                  :
: REN                : Remote ENable            : REMOTE                    :
: NRE                : Not Remote Enable         : LOCAL                     :
: AAU                : Auto Address Unconfigure  : SEND CMD 154              :
: LPD                : Loop Power Down          : SEND LPD                  :
-----
: LAG                : Listen Address Group      :
-----
: MLA                : My Listen Address         : SEND MLA                  :
: LAD (0 à 30)      : Listen Address            : SEND LISTEN 0 à 30       :
: UNL                : UNListen                  : SEND UNL                  :
-----
: TAG                : Talk Adress Group        :
-----
: MTA                : My Talk Address           : SEND MTA                  :
: TAD (0 à 30)      : Talk Address              : SEND TALK 0 à 30         :
: UNT                : UNTalk                    : SEND UNT                  :
-----
: SAG                : Secondary Address Group   :
-----
: SAD 0 à 30        : Secondary Address         : SEND SAD 0 à 30          :
: ISA                : Illegal Secondary Address  : SEND SAD 31               :
-----

```

 GROUPE PRET

RFC	Ready For Command	SEND RDY	

ARG-EOT	Addressed Ready Group - End Of Transmission		

ETO	End of Transmission - Ok	SEND RDY 64	
ETE	End of Transmission - Error	SEND RDY 65	
NRD	Not Ready for Data	SEND RDY 66	

ARG - SOT	Addressed Ready Group - Start Of Transmission		

SDA	Send Data	ENTER :LOOP	
SST	Send Status	SEND RDY 97	
SDI	Send Device Identification	SEND RDY 98	
SAI	Send Accessory Identification	SEND RDY 99	
TCT	Take Control	PASS CONTROL	

ARG - AAG	Addressed Ready Group - Auto Address Group		

AAD (0 à 30)	Auto Address	SEND RDY 128 à 158	
IAA	Illegal Auto Address	SEND RDY 159	
AEP (0 à 30)	Auto Extended Primary	SEND RDY 160 à 190	
IES	Illegal Extended Primary	SEND RDY 191	
AES (0 à 30)	Auto Extended Secondary	SEND RDY 192 à 222	
IES	Illegal Extended Secondary	SEND RDY 223	
AMP (0 à 30)	Auto Multiple Primary	SEND RDY 224 à 254	
IMP	Illegal Multiple Primary	SEND RDY 255	
ZES (AES 0)	Zero Extended Secondary	SEND RDY 192	

 GROUPE IDENTIFICATION

IDY (0 à 255)	Identify	SEND IDY 0 à 255	
ISR (0 à 255)	Identify - Service Request	SEND IDY 0 à 255	
		Si BIT(REQUEST,6)=1.	

Le programme SEND permet d'envoyer une séquence de messages HPIL, sans avoir à se préoccuper d'obscurs calculs pour avoir le nombre à mettre derrière SEND CMD, ou autre SEND RDY.

Le sous-programme SEND, par contre, a pour vocation d'étendre la fonction SEND standard à tous les messages décrits dans les spécifications de l'HPIL. Il ne traite qu'un seul message à la fois, ce qui permet de lui conserver sa simplicité.

Faites [RUN]. Vous avez alors à l'affichage: "SEND ".

Les messages peuvent être tous écrits en ligne, ou envoyés les uns après les autres.

Vous pouvez par exemple envoyer : IFC REN AAU LAD21 MTA, [END LINE], ou alors IFC [END LINE], puis REN [END LINE], AAU [END LINE], LAD21 [END LINE], etc.

Dans les deux cas, notez qu'il ne faut point mettre d'espace entre le message et son paramètre éventuel. Il faut également ne laisser qu'un seul espace pour séparer deux messages.

Pour le sous-programme SEND, pour utilisation à partir d'un autre programme, faites: CALL SEND ('IFC'), si vous voulez envoyer IFC par exemple. Simple, non ?

```

100 DIM S$(91)
  - S$: contient la chaîne entrée au clavier.
110 DISPLAY IS #
  - Pour éviter les interférences...
120 LINPUT 'SEND ';S$
  - LINPUT pour éviter l'erreur en cas d'introduction de virgule.
130 SFLAG -24
  - Pour éviter que le HP71 réassigne la boucle lors de toute opération HPIL.
140 IF S$=' ' THEN CFLAG -24 @ END
150 P=POS(S$, ' ')
  - Le séparateur entre plusieurs ordres HPIL est l'espace.
160 IF NOT P THEN P=LEN(S$)
170 DISP S$[1,P]
  - On prend seulement un message à la fois.
180 CALL SEND(S$[1,P])
190 DIM S$(91)
200 S$=S$[P+1]
  - On garde les autres messages pour les traiter ultérieurement.
210 IF LEN(S$) THEN 60 ELSE 30

=====
1000 SUB SEND(S$)
  - Sous forme de sous-programme: pour être appelé de n'importe quel programme, et ainsi
  devenir une réelle extension de la fonction SEND.
1010 DIM C$(3),K$(72),L$(69)
  - C$: le message proprement dit (sans paramètre éventuel).
  K$ et L$: 2 variables pour définir une seule chaîne qui contient tous les messages
  (lignes 1530 et 1540).
1020 C$=S$[1,3]
  - On extrait le message proprement dit.
1030 RESTORE 'CODE'
1040 READ K$
1050 READ L$
1060 P=(POS(K$&L$,C$)-1)/3
  - P est le numéro du message dans la chaîne K$&L$. Vive POS, qui évite une longue

```

```

recherche !
1070 IF FP(P) THEN 'ERREUR'
  - Il faut éviter un faux message, du genre 'ADT', 'DTA', ou tout autre se situant à cheval
  sur deux vrais messages.
  Si le message n'est pas trouvé, POS vaut 0, donc P vaut -1/3, non entier.
1080 IF P<6 THEN 'CG'
  - Command Group, avec paramètre.
1090 IF P<24 THEN 'C'
  - Command Group, sans paramètre.
1100 IF P<28 THEN 'RG'
  - Ready Group, avec paramètre.
1110 IF P<42 THEN 'R'
  - Ready Group, sans paramètre.
1120 P=P-41
  - Sinon, on ramène P à une valeur plus facilement exploitable.
1130 IF P=1 THEN SEND DATA FNV
  - FNV est une fonction utilisateur, qui retourne le paramètre du message, et 0 s'il n'y en
  a pas.
1140 IF P=2 THEN SEND END FNV
  - Le message END définit une fin d'enregistrement. Ne pas confondre avec:
  - EOT: End Of Transmission (Fin de transmission), qui signifie l'arrêt de la transmis-
  sion par l'émetteur. EOT n'est pas un ordre HPIL, mais un nom générique pour ETO, ETE.
  - EOL: End Of Line (Fin de ligne), qui n'est pas un message HPIL, mais l'envoi
  automatique de la séquence définie par ENDLIN. C'est une "Macro Instruction".
1150 IF P=3 THEN SEND IDY FNV
  - IDY ne peut être envoyé, lorsque le HP71 est non contrôleur, que si la boucle a reçu une
  validation demande asynchrone (EAR).
1160 IF P=4 THEN SEND MLA
  - MLA n'est pas à proprement parler un message, mais active la fonction récepteur (fonction
  Listener), et donc annule l'éventuel état émetteur du HP71.
1170 IF P=5 THEN SEND MTA
  - récepteur antérieur.
  active sa fonction émetteur (fonction Talker), et donc annule un éventuel état
1180 GOTO 'FIN'

```

```

=====
1190 'CB':
1200 C=FNV
  - C est l'argument pour la commande.
1210 IF C>31 OR C$='PPE' AND C>15 THEN 'ERREUR'
  - L'envoi de LAD31 (UNL), par exemple, est tout à fait possible.
1220 C=(P+1)*32+C
  - C est transformé en le numéro à associer à SEND CMD pour fournir le bon message.
1230 GOTO 'CMD'

```

```

=====
1240 'C':
1250 RESTORE 'CD'
  - Pour connaître le numéro particulier de la commande sans paramètre.
  NUL correspond à 0, GTL à 1, SDC à 4, etc...
1260 FOR I=0 TO P-6
1270 READ C
1280 NEXT I

```

```

=====
1290 'CMD':
1300 SEND CMD C
  - L'envoi de la commande proprement dit.
1310 GOTO 'FIN'

```

```
=====
1320 'CD':
1330 DATA 0,1,4,5,8,15,16,17,20,21,24,63,95,144,146,147,154,155
```

```
=====
1340 'RG':
- Le traitement des messages RDY est similaire à celui des CMD, sauf qu'il n'y a pas de
  Parallel Poll.
1350 C=FNV
1360 IF C>31 THEN 'ERREUR'
1370 C=(P-16)*16+C
1380 GOTO 'RDY'
```

```
=====
1390 'R':
1400 RESTORE 'RG'
1410 FOR I=0 TO P-28
1420 READ C
1430 NEXT I
```

```
=====
1440 'RDY':
1450 SEND RDY C
1460 GOTO 'FIN'
```

```
=====
1470 'RD':
1480 DATA 0,64,65,66,96,97,98,99,100,159,191,192,223,255
```

```
=====
1490 'ERREUR':
- C'est le même message pour toutes les erreurs (elles sont toutes du même type).
1500 DISP 'Arg. non valide'
1510 BEEP 500,.5
- Il n'y a pas de GOTO 'FIN', car tout ce qui reste n'est qu'étiquette, DATA, ou fonction
  utilisateur...
```

```
=====
1520 'CODE':
1530 DATA LADTADSADPPEDDLDDTNULGTLSDCPPDGETELNNOPLLODCLPPUEARUNLUNTIFCRENNREAAULPD
1540 DATA AADAEPAESAMPRFCETOETENRDSASSTSDISAITCTIAAIEPZESIESIMPDABENDIDYMLAMTA
```

```
=====
1550 DEF FNV
- Cette fonction donne l'argument du message contenu dans S$. S'il n'y en a pas, il est
  par défaut à 0.
1560 IF LEN(S$)>3 THEN FNV=VAL(S$[4])
1570 END DEF
```

```
=====
1580 'FIN':
1590 END SUB
```

HP.75 / 71

HP 75:

1 : PUISSANCE 4

(FREDERIC BARNAUD)

ASSEMBLEUR:

1 : HP71 DESASSEMBLAGE DE KEYWAIT\$

(DAVID / MARTINET)

DIVERS:

1 : HP71 EDITEUR DE TEXTE ET FRALEX

(PIERRE DAVID)

BASIC:

1 : HP71 APPROXIMATION DE FRACTIONS

(PIERRE DAVID)

2 : HP71 CATALOGUE ETENDU

(JEAN-PIERRE BONDU)

La rubrique 75 est couplée au HP71. La raison en est que les programmes (s'ils sont enbasic), de l'un peuvent aller, moyennant quelques modifications, sur l'autre.

Plus d'une dizaine de Ko ce mois-ci! Beau succès que cette rubrique!

PUISSANCE 4, POUR HP 75

A la maison, le seul individu à ne rien connaître à Puissance 4 était mon HP 75. Il fallait donc y remédier, car comment aurait-il pu vivre ainsi plus longtemps ?

Voici ce programme, PUISSSS (avec quatre "s", normal !), qui a terminé quatrième (normal !) au tournoi de l'Institut Industriel Du Nord (IDN) le 20 Janvier 1984, qui regroupait 40 ordinateurs de table en catégorie interprétés.

Grosso modo, il se compose de six parties :

- * 10 à 90 : neuf lignes de DATA nécessaires à l'analyse des coups dans le pgm de réflexion <A>.
- ** 100 à 130 : initialisation nécessaire pour le pgm de réflexion .
- *** 140 à 230 : sous-pgm d'analyse utilisé par <A> et .
- **** 240 à 520 : ces lignes regroupent le début de la partie, la saisie et la vérification des coups de l'adversaire, ainsi qu'une petite bibliothèque sur 5/2 coups.
- ***** 530 à 1430 : le pgm <A>, qui prend en charge la partie à partir du 12ième 1/2 coup. Les organes vitaux de sa stratégie sont les quatre sous-pgm placés de 820 à 1250.
- ***** 1440 à 2060 : le pgm , qui s'occupe des coups 3 à 6, moins performant que <A> mais plus rapide et amplement suffisant en début de partie.

Stratégie utilisée par PUISSSS :

* pgm : pgm relativement classique qui analyse ses différents coups possibles, en considérant les alignements "les plus visibles" ainsi que la réponse éventuelle de l'adversaire: gagnante ou pas ?

** pgm <A> : recherche très complète sur 2 1/2 coups avec une stratégie assez évoluée pour qu'il puisse tendre des pièges ou jouer finement : les positions gagnantes à tous les coups (ex: fig 1), par exemple, ne lui échappent pas.

Notez bien qu'il n'y a qu'une fonction RND par pgm d'analyse : j'ai horreur du hasard dans un pgm de jeu de réflexion.

Principales variables utilisées :

- N9 : nombre de 1/2 coups depuis le début de partie.
- M1,M2,M3 : utilisées pour la bibliothèque d'ouverture.
- A(i,j) : matrice représentant la situation en cours, incluant les lignes 0 et 7 ainsi que la la colonne 0 (plus facile pour <A>).
- L(i) : nombre de pions dans la colonne i.
- T,T1,T2 : respectivement temps cumulé de réflexion, valeurs de TIME à l'instant où débute la nouvelle recherche et où elle se termine.
- O=1 et X=2 : "couleur" des pions; O concerne l'ordinateur et X le joueur.
- J : "couleur" du pion sur lequel réfléchit le pgm.
- A,Ai et B,Bi : des numéros soit de lignes, soit de colonnes.
- H,Hi,D1 et V,Vi,D2 : représentent des déplacements verticaux ou horizontaux à partir de la position étudiée.
- V(i) : notes possibles dans l'évaluation de .
- N(i) et M(j) : notes attribuées dans <A> respectivement pour la colonne i possible pour le 75, et pour la réponse éventuelle j de l'adversaire. La note globale, encore appelée N(i), prend en considération les deux notes partielles ci-dessus.
- V : équivalent des N(i), dans , V1 étant le maximum des notes des coups déjà évalués.
- D'autre part il y a énormément de variables indescriptibles, dont beaucoup sont utilisées comme drapeaux.

PROGRAMME PUISSSS

```

=====
10 DATA 0,-1,0,-2,0,-3,0,1,0,-1,0,1,0,-2,0,2,0,1,0,2,0,-1,0,3,1,1,2,2,3,3,-1,-1,1,1,-1,-1
20 DATA 2,2,-2,-2,-1,-1,-2,-2,1,1,-3,-3,1,-1,2,-2,-1,1,3,-3,-1,1,1,-1,2,-2,-2,2
30 DATA -1,1,-2,2,-3,3,1,-1
40 DATA 0,-1,0,-3,0,-2,0,1,0,-2,0,-1,0,2,0,3,0,1
50 DATA 1,1,3,3,2,2,2,-1,-1,1,1,-2,-2,-3,-3,-1,-1
60 DATA 1,-1,3,-3,2,-2,2,-2,-1,1,1,-1,-2,2,-3,3,-1,1
70 DATA 0,2,0,3,2,2,3,3,0,1,1,1,0,2,0,3,-2,2,-3,3,0,1,-1,1
80 DATA 2,3,2,3,2,3,2,-1
90 DATA 1,1,1,2,1,3,0,1,-1,2,-2,3,2,1,3,2,4,3

```

```

=====
100 DIM L(7),A(7,7),N(7),M(7),S(4),F(4),V(16)

```

```

=====
110 DATA 1,100,500,1.E20,1,800,4000,1.E20
120 DATA 1,75,900,1.E18,1,450,3000,1.E18

```

```

=====
130 RESTORE 110 @ FOR I=1 TO 16 @ READ V(I) @ NEXT I @ GOTO 240
140 C=0 @ H=0 @ V2=-1 @ GOSUB 180
150 C=0 @ H=1 @ V2=0 @ GOSUB 180 @ H=-1 @ V2=0 @ GOSUB 180
160 C=0 @ H=1 @ V2=1 @ GOSUB 180 @ H=-1 @ V2=-1 @ GOSUB 180
170 C=0 @ H=-1 @ V2=1 @ GOSUB 180 @ H=1 @ V2=-1 @ GOSUB 180 @ RETURN
180 K=0
190 K=K+1 @ A1=A+K*V2 @ B1=B+K*H
200 IF A1<1 OR B1<1 OR A1>6 OR B1>7 THEN RETURN
210 IF A(A1,B1)<>J THEN RETURN ELSE C=C+1
220 IF C=3 THEN POP @ RETURN
230 IF K=3 THEN RETURN ELSE GOTO 190
240 FOR A=0 TO 7 @ FOR B=0 TO 7 @ A(A,B)=0 @ NEXT B @ NEXT A @ FOR A=1 TO 7 @ L(A)=0 @ NEXT A
250 M3=0 @ M2=0 @ N9=0 @ FOR I=1 TO 7 @ A(0,I)=3 @ A(I,0)=3 @ NEXT I
260 T1=0 @ T2=1 @ O=1 @ X=2 @ T=0
270 M1=0 @ INPUT "Desirez-vous commencer ? ";A$
280 IF A$[1,1]="N" THEN A(1,4)=1 @ L(4)=1 @ BEEP @ PRINT "Je joue 4 ( 0 ),"; @ N9=1
290 IF N9<>42 THEN INPUT "votre choix:";B ELSE 1790
300 IF B<1 OR B>7 THEN PRINT "Position non jouable" @ GOTO 290
310 IF L(B)=6 THEN PRINT "Position non jouable" @ GOTO 290
320 L(B)=L(B)+1 @ N9=N9+2
330 A=L(B) @ A(A,B)=2
340 J=2 @ GOSUB 140 @ IF C=3 THEN PRINT "Vous avez gagne !! en";T @ END
- * Bibliotheque d'ouverture *
360 IF N9=6 AND M2=43 AND B=6 THEN M3=436
370 IF N9=8 AND M3=436 AND B=5 THEN M=4 @ T2=T1+2 @ BEEP @ GOTO 1430
380 IF N9=6 AND M2=45 AND B=2 THEN M3=452
390 IF N9=8 AND M3=452 THEN M=4 @ BEEP @ T2=T1+2 @ GOTO 1430
400 IF N9=6 AND M2=33 AND B=4 THEN M3=334
410 IF N9=8 AND M3=334 AND B=6 THEN M=6 @ BEEP @ T2=T1+2 @ GOTO 1430
420 IF N9=6 AND M2=55 AND B=4 THEN M3=554
430 IF M3=554 AND N9=8 AND B=2 THEN BEEP @ M=2 @ T2=T1+2 @ GOTO 1430
440 IF N9=2 OR N9=3 OR N9=5 THEN M1=B @ M=4 @ BEEP @ GOTO 1430
450 IF N9>11 THEN 530

```

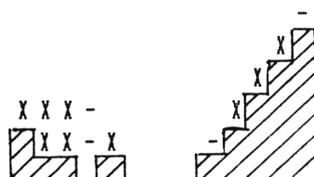
Il y a un tel fouillis de variables, car je n'ai pas conçu <A> en même temps que , et le temps me manquait ensuite pour synchroniser tous les noms des variables des deux pgm ; d'autant plus qu'à l'époque PUISSSS n'était pas destiné à être publié. Mais le temps a suivi son cours...

Notez enfin que la répartition de <A> et au cours de la partie est laissée au libre choix de l'utilisateur : il suffit de modifier les lignes 440 et 450. Cela confère à PUISSSS une certaine souplesse ; à vous de trouver une répartition meilleure que la mienne !

Maintenant, chers confrères, à vous de jouer et bonne chance à ... PUISSSS.

auteur : Frederic BARNAUD

Fig 1:



Note de la "Rédaction":

Compte tenu de la compatibilité HP 75/HP 71, les possesseurs de ce dernier ordinateur peuvent entrer ce programme, à condition toutefois de modifier quelques détails:

- Tout d'abord, puisque le HP71 n'accepte pas que tableaux et variables aient le même nom, il faut changer le nom de tous les tableaux ("tous" pour ne pas avoir de distinction à faire). Je vous propose, pour simplifier, de remplacer les tableaux par le même nom suivi de "0". Ex: A devient A0. Remplacez donc A par A0 à la ligne 100, puis dans toutes les lignes où le tableau A intervient. Si vous disposez de l'Editeur, rien de plus simple !
- Vous devez remplacer les affectations multiples par autant d'affectations que de variables: à la ligne 1180, par exemple, mettez:
1180 A1=A @ A2=A @ A6=A @ A3=A-1 @ A4=A3 @ A5=A3 @ A7=A3
compris ? Oui ? Alors recommencez pour les lignes 1190 et 1240.
- Remplacez aussi INF par MAXREAL à la ligne 740.

Voilà, ça doit suffire. Bonne chance...

Pierre DAVID

```

460 IF N9>5 AND N9<12 THEN 1450
470 M2=10*M1+B
480 IF M2=43 OR M2=12 OR M2=21 OR M2=22 OR M2=26 THEN BEEP @ M=2 @ GOTO 1430
490 IF M2=33 OR M2=42 THEN BEEP @ M=3 @ GOTO 1430
500 IF M2=55 OR M2=46 THEN BEEP @ M=5 @ GOTO 1430
510 IF M2=45 OR M2=62 OR M2=66 OR M2=67 OR M2=76 THEN BEEP @ M=6 @ GOTO 1430
520 M=4 @ BEEP @ GOTO 1430
- * Pgm de reflexion <A> *
540 C1=0 @ Z=0 @ FOR I=1 TO 7 @ N(I)=0 @ NEXT I @ T1=INT(TIME)
550 FOR F=1 TO 7
560 IF L(F)=6 THEN Z=Z+1 @ N(F)=-INF @ GOTO 1330
570 L(F)=L(F)+1 @ A=L(F) @ B=F @ J=1 @ GOSUB 140
580 IF C=3 THEN M=F @ F=7 @ C1=1 @ GOTO 1330 ELSE A(A,B)=1
590 GOSUB 820 @ N(F)=N(F)+Y
600 IF A<3 OR A=6 THEN 620
610 IF A(A-1,B)=1 AND A(A-2,B)=1 THEN N(F)=N(F)+100010
620 GOSUB 950 @ N(F)=N(F)+Y
630 RESTORE 90 @ FOR K=1 TO 3 @ READ H1,V1,H2,V2,H3,V3 @ C2=0
640 A1=A+H1 @ A2=A+H2 @ A3=A+H3 @ B1=B+V1 @ B2=B+V2 @ B3=B+V3
650 IF A3<1 OR A3>6 OR B3<1 OR B3>7 THEN 670
660 IF A(A1,B1)=1 AND A(A2,B2)=1 AND A(A3,B3)=1 THEN N(F)=N(F)-10000000 @ C2=1
670 IF C2<>0 THEN 690
680 B1=B-V1 @ B2=B-V2 @ B3=B-V3 @ C2=1 @ GOTO 650
690 NEXT K
700 IF N9<12 OR N9>20 THEN 720
710 GOSUB 1040 @ N(F)=N(F)+Y
720 FOR I=1 TO 7 @ M(I)=0 @ NEXT I @ W2=0
730 FOR B=1 TO 7 @ IF L(B)=6 THEN 1280 ELSE A=L(B)+1 @ J=2 @ GOSUB 140
740 IF C=3 THEN N(F)=-INF/10 @ B=7 @ GOTO 1280
750 GOSUB 820 @ M(B)=M(B)+Y
760 IF C2=1 THEN B=7 @ GOTO 1280
770 IF A<3 OR A=6 THEN 790
780 IF A(A-1,B)=2 AND A(A-2,B)=2 THEN M(B)=M(B)+100000
790 GOSUB 950 @ M(B)=M(B)+Y
800 IF N9<12 OR N9>29 THEN 1270
810 GOSUB 1040 @ M(B)=M(B)+Y @ GOTO 1270
- * Ss-pgm d'analyse 1 *
830 RESTORE 10 @ C2=0 @ Y=0
840 FOR K=1 TO 9 @ READ H1,V1,H2,V2,H3,V3,H4,V4 @ Z1=0 @ N8=0
850 A1=A+H1 @ A2=A+H2 @ A3=A+H3 @ A4=A+H4 @ B1=B+V1 @ B2=B+V2 @ B3=B+V3 @ B4=B+V4
860 IF A1<1 OR A2<1 OR B1<1 OR B2<1 OR A1>6 OR A2>6 OR B1>7 OR B2>7 THEN 940
870 IF A3<1 OR B3<1 OR A3>6 OR B3>7 THEN N8=1 @ GOTO 890
880 IF A(A1,B1)=J AND A(A2,B2)=J AND A(A3,B3)=0 THEN Y=Y+1000000
890 IF A4<1 OR A4>6 OR B4<1 OR B4>7 THEN 940
900 IF A(A1,B1)=J AND A(A2,B2)=J AND A(A4,B4)=0 THEN Y=Y+1000000
910 IF N8=1 THEN 940
920 IF A(A1,B1)=3-J AND A(A2,B2)=3-J AND A(A3,B3)=0 AND A(A3-1,B3)<>0 THEN Z1=1
930 IF Z1=1 AND A(A4,B4)=0 AND A(A4-1,B4)<>0 THEN Y=INF/100 @ K=9 @ C2=1
940 NEXT K @ RETURN
- * Ss-pgm d'analyse 2 *
960 Y=0 @ FOR K=1 TO 9 @ C2=0 @ READ H1,V1,H2,V2,H3,V3
970 A1=A+H1 @ A2=A+H2 @ A3=A+H3 @ B1=B+V1 @ B2=B+V2 @ B3=B+V3
980 IF A1<1 OR A2<1 OR A3<1 OR B1<1 OR B2<1 OR B3<1 THEN 1010
990 IF A1>6 OR A2>6 OR A3>6 OR B1>7 OR B2>7 OR B3>7 THEN 1010
1000 IF A(A1,B1)=J AND A(A2,B2)=J AND A(A3,B3)=0 THEN Y=Y+1000000
1010 IF C2=1 THEN 1030

```

```

1020 A1=A-H1 @ A2=A-H2 @ A3=A-H3 @ B1=B-V1 @ B2=B-V2 @ B3=B-V3 @ C2=1 @ GOTO 980
1030 NEXT K @ RETURN
- * Ss-pgm d'analyse 3 *
1050 RESTORE 70 @ Y=0
1060 FOR K=1 TO 2 @ C2=0 @ Z1=0 @ READ H1,V1,H2,V2,H3,V3,H4,V4,H5,V5,H6,V6
1070 A1=A+H1 @ A2=A+H2 @ A3=A+H3 @ A4=A+H4 @ A5=A+H5 @ A6=A+H6
1080 B1=B+V1 @ B2=B+V2 @ B3=B+V3 @ B4=B+V4 @ B5=B+V5 @ B6=B+V6
1090 IF A4<1 OR A4>6 OR B2<1 OR B2>7 THEN 1120
1100 IF A(A1,B1)=J AND A(A2,B2)=J AND A(A3,B3)=J AND A(A4,B4)=J THEN Z1=1
1110 IF Z1=1 AND A(A5,B5)=0 AND A(A6,B6)=0 THEN Y=Y+J*1000000000 @ C2=1
1120 IF C2<>0 THEN 1150
1130 A1=A-H1 @ A2=A-H2 @ A3=A-H3 @ A4=A-H4 @ A5=A-H5 @ A6=A-H6 @ C2=1
1140 B1=B-V1 @ B2=B-V2 @ B3=B-V3 @ B4=B-V4 @ B5=B-V5 @ B6=B-V6 @ GOTO 1090
1150 NEXT K
- * Ss-pgm d'analyse 3bis *
1170 Y=0 @ FOR K=1 TO 2 @ Z1=0 @ C2=0 @ READ V1,V2,V3,V4
1180 A1,A2,A6=A @ A3,A4,A5,A7=A-1
1190 B1=B+V1 @ B2=B+V2 @ B3=B+V3 @ B4=B+V4 @ B5=B @ B6,B7=B+1
1200 IF A=1 OR B2<1 OR B2>7 OR B4<1 OR B4>7 THEN 1230
1210 IF A(A1,B1)=J AND A(A2,B2)=J AND A(A3,B3)=J AND A(A4,B4)=J AND A(A5,B5)=J THEN Z1=1
1220 IF Z1=1 AND A(A6,B6)=0 AND A(A7,B7)=0 THEN Y=Y+J*1000000000 @ C2=1
1230 IF C2<>0 THEN 1250
1240 B1=B-V1 @ B2=B-V2 @ B3=B-V3 @ B4=B-V4 @ B6,B7=B-1 @ C2=1 @ GOTO 1200
1250 NEXT K @ RETURN
- * Suite de <A> *
1270 IF N(F)-M(B)<W1 AND F<>1 THEN N(F)=N(F)-M(B) @ B=7 @ W2=1
1280 NEXT B
1290 A=L(F) @ L(F)=L(F)-1 @ A(A,F)=0
1300 IF W2=1 THEN 1320
1310 W=M(1) @ FOR I=2 TO 7 @ W=W+M(I) @ NEXT I @ N(F)=N(F)-W
1320 IF F=1 THEN W1=N(1) ELSE W1=MAX(W1,N(F))
1330 NEXT F @ T2=INT(TIME) @ T=T+T2-T1 @ BEEP
1340 N(2)=N(2)+1 @ N(3)=N(3)+5 @ N(4)=N(4)+10 @ N(5)=N(5)+5 @ N(6)=N(6)+1
1350 IF Z=7 THEN PRINT "Partie nulle en";T @ END
1360 IF C1=1 THEN PRINT "Je joue en";M;"et je gagne en";T @ END
1370 W=MAX(N(1),N(2)) @ FOR I=3 TO 7 @ W=MAX(W,N(I)) @ NEXT I
1380 R=INT(RND*2) @ IF R=0 THEN 1410
1390 FOR F=1 TO 7 @ IF N(F)=W THEN M=F @ F=7
1400 NEXT F @ GOTO 1430
1410 FOR F=7 TO 1 STEP -1 @ IF N(F)=W THEN M=F @ F=1
1420 NEXT F
1430 PRINT "Je joue";M;"(";"T2-T1-1;""); @ L(M)=L(M)+1 @ A=L(M) @ A(A,M)=1 @ GOTO 290
- * Pgm de reflexion <B> *
1450 M=B @ T1=INT(TIME) @ M9=0 @ V1=0 @ N1=1
1460 FOR M4=1 TO 7 @ L=L(M4)+1
1470 IF L>6 THEN 1770
1480 V=1 @ P=0 @ W=0 @ M=M4
1490 GOSUB 1850
1500 FOR Z1=1 TO 4 @ N(Z1)=0 @ NEXT Z1
1510 FOR Z=1 TO 4
1520 S=S(Z)
1530 IF S-W>3 THEN T2=INT(TIME) @ GOTO 1810
1540 U=S+F(Z)
1550 IF U<4 THEN 1570
1560 V=V+4 @ N(S)=N(S)+1
1570 NEXT Z

```

```

1580 FOR I=1 TO 4 @ N=N(I)-1
1590 IF N=-1 THEN 1620
1600 I1=8*N+4*SGN(N)+1
1610 V=V+V(I1)+N*V(8*N+I)
1620 NEXT I
1630 IF N=1 THEN 1660
1640 M=1 @ P=X
1650 GOTO 1490
1660 FOR B=1 TO 7 @ A=L(B)+1
1670 IF A>6 THEN 1710
1680 J=X @ GOSUB 140
1690 IF C=3 AND B<>M4 THEN V=V-1.E19
1700 IF C=3 AND B=M4 THEN V=.5
1710 NEXT B
1720 IF V<V1 THEN 1770
1730 IF V>V1 THEN N1=1 @ GOTO 1760
1740 N1=N1+1
1750 IF RND>1/N1 THEN 1770
1760 V1=V @ M9=M4
1770 NEXT M4 @ T2=INT(TIME) @ T=T+T2-T1
1780 IF M9<>0 THEN 1800
1790 BEEP @ PRINT "Partie nulle en";T-60 @ END
1800 M=M9
1810 BEEP @ PRINT "Je joue";M;"(";ABS(T2-T1);");";
1820 L=L(M)+1 @ L(M)=L(M)+1
1830 A(L,M)=0 @ A=L @ B=M @ J=0
1840 GOSUB 140 @ IF C=3 THEN PRINT " et je gagne en";T @ END ELSE 290
1850 Q=X
1860 IF P=X THEN Q=0
1870 D2=1 @ D1=0 @ Z=0 @ GOSUB 1920
1880 D1=1 @ D2=1 @ GOSUB 1920
1890 D2=0 @ D1=1 @ GOSUB 1920
1900 D1=1 @ D2=-1 @ GOSUB 1920
1910 RETURN
1920 D=1 @ S=1 @ U=0 @ Z=Z+1
1930 C=0 @ FOR K=1 TO 3
1940 M5=M+K*D1 @ L1=L+K*D2
1950 IF M5<1 OR M5>7 OR L1<1 OR L1>6 THEN 2020
1960 B=A(L1,M5)
1970 IF C=0 THEN 2000
1980 IF B=Q THEN K=3 @ GOTO 2020
1990 U=U+1 @ GOTO 2020
2000 IF B=P THEN S=S+1 @ GOTO 2020
2010 C=1 @ GOTO 1980
2020 NEXT K
2030 IF D=0 THEN 2060
2040 D=0 @ D1=-D1 @ D2=-D2
2050 GOTO 1930
2060 S(Z)=S @ F(Z)=U @ RETURN

```

DESASSEMBLAGE DE KEYWAIT*

Le problème qui se pose aux nouveaux possesseurs du module Forth/Assembleur est le manque d'exemples fournis. La documentation n'est pas vraiment limpide, et le HP71 trop jeune pour qu'il y ait des prédecesseurs en matière d'assemblage.

Néanmoins, il ne faut pas oublier quelques personnes, qui connaissent bien le HP71: ceux qui l'ont fait.

En effet, nous avons à notre disposition quantité de fichiers LEX: ceux compris dans l'interpréteur Basic, et ceux des différents modules enfichables ou fichiers séparés. Seul problème: la Rom Basic fait 64 Ko, le module HPIL 16 Ko, le module Maths 32 Ko (il est privatisé, mais ce n'est pas une raison pour ne pas le regarder...). Fralex, à lui seul, fait 2,5 Ko environ. Le fichier EDLEX (génial) occupe 2,5 à 3 Ko suivant le module (Editeur de texte, ou Forth). Il faut donc, pour un début, s'attaquer à des fichiers plus petits...

Or, il en est un que HP, dans son infinie bonté, met "à notre disposition". On le trouve, embusqué dans des modules tels que "Circuit Analysis", ou "Financial"... Son principal avantage: 55 octets, toutes taxes comprises. La simplicité en prime !

Nous voilà donc partis, un beau matin pluvieux de septembre... Le désassemblage fût hardu, la quête de la vérité universelle, longue. Armés de notre seule patience, de quelque PEEK bien choisi, et de quelques IDS, nous partîmes sur la route de la connaissance suprême (snif).

Une fois le désassemblage réalisé, il ne restait plus qu'à réassembler... Oh miracle ! Pas de "Memory lost" à l'horizon, même pas de vulgaire "plantage", tout fonctionnait à merveille ! Nous avons réussi ! Aujourd'hui, il nous paraît intéressant de vous présenter le fruit de ces investigations... en attendant de vous donner nos propres fonctions, le mois prochain... Si si, ça avance !

! Pour que vous compreniez bien le fonctionnement
! du fichier LEX, nous avons reproduit un exemple
! de "LISTING FILE", c'est à dire un compte rendu
! détaillé de l'assemblage. Vous pouvez l'obtenir,
! si vous possédez le module Forth bien sûr, en
! exécutant (en Forth) : " KWASSEMB" LISTING S!

! Le listing source est plus clair, mais détaille
! moins les recoins obscurs du LEX. Nous avons,
! dans la mesure du possible, commenté les deux
! listages.

! Rappelons le processus, pour assembler:

! - Tout d'abord, il faut créer un fichier texte
! qui contiendra les mnémoniques d'assemblages,
! éventuellement des commentaires, et aussi (et
! surtout) des directives d'assemblages, ou
! pseudo-opérations. Ces dernières ne sont pas des
! codes compris par le microprocesseur, mais des
! ordres donnés au programme "assembleur".

! - Ensuite, il faut lancer le programme
! Assembleur. Il se charge alors de transformer
! toutes vos mnémoniques en codes binaires, et de
! les insérer dans le contexte d'un fichier LEX,
! BIN, ou d'une primitive Forth. Le fichier qui
! contient les codes binaires est appelé
! "fichier objet".

! - Il est possible "d'espionner" la phase
! d'assemblage. C'est le rôle du "Listing file".
! L'espion peut être soit une imprimante, soit un
! fichier texte en mémoire.

! Notons que le désassemblage a été fait à la main.
! Ce fut d'autant plus désagréable que les IDS ne
! sont vraiment pas prévus pour les pauvres
! désassembleurs que nous sommes. Le comble de
! l'hypocrisie, c'est que la fonction DEBUG existe.
! Essayez donc, depuis Basic, de taper:
! DEBUG [END LINE]. Eh oui !

! Quand ?

! A Bientôt, et Heureuse Programmation...

! Michel MARTINET et Pierre DAVID

KEYWAIT\$: Fichier source

```

LEX   'KEYWAIT' - Ce fichier a pour nom KEYWAIT (Copyright Hewlett-Packard 1984).
ID    #52      - Identification du fichier LEX parmi 256 possibles. Pour une applica-
  $      tion donnée, il est possible de réserver un numéro auprès d'HP. Les
  $      numéros 52 à 5B sont attribués à la bibliothèque des utilisateurs.
  $      Pour la conception, nous avons à notre disposition les ID 5C à 5E.
MSG   0        - Il n'y a pas de message dans ce LEX. Eventuellement, les messages
  $      seront appelés par leur numéro, ce qui permettra leur traduction...
POLL  0        - Il n'y a pas de routine d'exception (Poll handler) dans ce LEX. Une
  $      telle routine intercepte le contrôle de l'appareil, lors d'une
  $      situation déterminée (exemple: "COPY vers un fichier non reconnu" est
  $      intercepté par le module HPIL).
SLEEP EQU #006C2 - Cette routine teste le clavier et dispose le microprocesseur en état
  $      de "sommeil" à moins qu'une touche soit enfoncée ou que la mémoire
  $      clavier ne soit pas vide.
CKSREQ EQU #00721 - (Check Service REQuest) Cette routine traite les interruptions telles
  $      que BAT, horloge ... car KEYWAIT$ attend indéfiniment. C'est la seule
  $      routine qui permet l'interruption de la fonction.
KEY$  EQU #IACAB - C'est la fonction KEY$.
      ENTRY KW$   - KW$ est une étiquette locale, pointant le début de la zone d'exécution
      CHAR #F     - C'est une fonction, F signifie que KEYWAIT$ sera utilisable directe-
  $      ment au clavier, après IF..THEN..ELSE, et programmable.
      KEY 'KEYWAIT$' - Le nom.
TOKEN 1        - Le code, dans le fichier LEX. Assimilable à XROM HP41 (2ème paramètre)
ENDTXT                                     - Fin de la partie texte.
NIBHEX 00      - Nombre maximum et minimum de paramètres pour cette fonction (1 quartet
  $      chacun).
KW$    CDOEX   - Les registres D0 et D1 doivent être sauvegardés lors de l'exécution de
      RO=C     la fonction. Ils sont stockés respectivement dans R0 et R1 avant l'ap-
      CD1EX    pel des routines SLEEP et CKSREQ. Ils sont restitués à la fin pour le
      R1=C     branchement à KEY$.
LBL1   GOSBVL SLEEP - Début de la boucle.
      GOC LBL2  - La retenue (carry) est armée par la routine SLEEP si la mémoire
  $      clavier est vide.
      C=R1     - Sinon, on restitue les registres D0 et D1 avant le branchement à la
      D1=C     fonction KEY$.
      C=R0
      D0=C
      GOVLNG KEY$ - Fin de la fonction KEYWAIT$.
LBL2   GOSBVL CKSREQ - La mémoire clavier est vide, on fait les vérifications d'usage,
      GOTO LBL1 et on boucle.
      END      - Pseudo-opération optionnelle.

```

0001 00000	LEX 'KEYWAIT'	‡ Cf. listage source.
0001 00000 B4549575	NIBASC 'KEYWAIT'	‡ Nom du fichier,
00008 14944502		‡ sur 16 quartets.
0001 00010 802E	NIBHEX 802E	‡ Type de fichier (LEX), 4 quartets (LIF).
0001 00014 00	CON(2) 0	‡ Privatisation, sécurisation.
0001 00016 61	CON(2) #39	‡ Minute,
0001 00018 10	CON(2) #19	‡ heure,
0001 0001A 80	CON(2) #16	‡ jour,
0001 0001C 90	CON(2) #11	‡ mois,
0001 0001E 48	CON(2) #84	‡ et année de création.
0001 00020 37000	REL(5) FiLeNd	‡ Adresse relative fin de fichier, 20+73.
0002 00025	ID #52	‡ Cf. listage source.
0002 00025 25	CON(2) #52	‡ Numéro d'identification du LEX.
0002 00027 10	CON(2) 0001	‡ Plus petit,
0002 00029 10	CON(2) 0001	‡ et plus grand numéros de fonctions.
0002 0002B 00000	CON(5) 0	‡ Fin de la table de LEX.
0002 00030 F	NIBHEX F	‡ Pas de table d'accès rapide.
0002 00031 7100	REL(4) 1+TxTbSt	‡ Adr. relative début de la partie texte.
0003 00035	MSG 0	‡ Cf. listage source.
0003 00035 0000	CON(4) 0	‡ Pas de message dans ce LEX,
0004 00039	POLL 0	‡ Cf. listage source.
0004 00039 00000	CON(5) 0	‡ et pas de routine d'exception.
0005 0003E	SLEEP EQU #006C2	‡ Cf. listage source.
0006 0003E	CKSREQ EQU #00721	‡ "
0007 0003E	KEY\$ EQU #1ACAB	‡ "
0008 0003E	ENTRY KW\$	‡ "
0008 0003E	***MAIN TABLE***	‡ Début de la table principale.
0008 0003E 000	CON(3) (TxEn01)-(TxTbSt)	‡ Adr. rel. du texte dans la partie texte.
0008 00041 E1000	REL(5) KW\$	‡ Adr. relative de l'étiquette d'entrée.
0009 00046	CHAR #F	‡ Cf. listage source.
0009 00046 F	CON(1) #F	‡ Chargement de F.
0010 00047	KEY 'KEYWAIT\$'	‡ Cf. listage source.
0010 00047	***TEXT TABLE***	‡ Début de la partie texte.
0010 00047	TxTbSt	‡ Début de la partie texte (Text Table).
0010 00047	TxEn01	‡ Fin de la table texte.
0010 00047 F	CON(1) 15	‡ Longueur de 'KEYWAIT\$' en quartets - 1.
0010 00048 B4549575	NIBASC 'KEYWAIT\$'	‡ Code correspondant à
00050 14944542		‡ la pseudo-opération KEY.
0011 00058	TOKEN 1	‡ Cf. listage source.
0011 00058 10	CON(2) 1	‡ Code correspondant à token.
0012 0005A	ENDTXT	‡ Cf. listage source.
0012 0005A 1FF	NIBHEX 1FF	‡ Marqueur de fin de la partie texte.
0013 0005D 00	NIBHEX 00	‡ Cf. listage source.
0014 0005F 136	KW\$ CDOEX	‡ Cf. listage source, jusqu'à 'END'.
0015 00062 108	RO=C	‡
0016 00065 137	CD1EX	‡
0017 00068 109	R1=C	‡
0018 0006B 8F2C600	LBL1 60SBVL SLEEP	‡
0019 00072 451	60C LBL2	‡
0020 00075 119	C=R1	‡
0021 00078 135	D1=C	‡
0022 0007B 118	C=RO	‡
0023 0007E 134	DO=C	‡
0024 00081 8DBACA1	GOVLNG KEY\$	‡
0025 00088 8F12700	LBL2 60SBVL CKSREQ	‡
0026 0008F 6BDF	GOTO LBL1	‡
0027 00093	END	‡

PAGE 002

Forth Assembler **** SYMBOL TABLE ****

CKSREQ 00721
FiLeNd 00093
KEY\$ 1ACAB
KW\$ 0005F
LBL1 0006B
LBL2 00088
SLEEP 006C2
TxEn01 00047
TxTbSt 00047

SOURCE : KWSOURCE * Le fichier texte entré par EDTEXT.
*
OBJECT : KEYWAIT * Le fichier LEX, créé par l'assembleur.
*
LISTING : KWASSEMB * Le fichier que vous avez, imprimé, sous vos yeux. (optionnel)
*
DATE : 01:17:54 on 84/09/08 * Information utile pour l'archivage...
*
ERRORS : 000 * Vive nous ! (C'est la première fois que cela nous arrive ...)

EDITEUR DE TEXTE et FRALEX

Pour vous qui possédez le fichier FRALEX (traduction des messages en français), ainsi que le programme EDTEXT du module Forth/Assembleur ou Editeur de Texte, voici aujourd'hui un résumé des commandes de l'éditeur de texte. En effet, vous avez pu constater que FRALEX fait bien son travail, c'est à dire qu'il ne traduit pas seulement les messages d'erreurs, mais aussi toutes les commandes. Ça désoriente beaucoup au début, mais ensuite FRALEX devient vite indispensable ! Pour vous épargner une recherche fastidieuse, je vous propose le petit tableau suivant:

----- Commandes de l'éditeur de texte du HP71B (EDTEXT)

B: Bouger	[ligne début [ligne fin]] B [nom de fichier]
C: Chercher	[ligne début [ligne fin]] [?] C/ Chaîne [/]
D: Dupliquer	[ligne début [ligne fin]] D [nom de fichier]
E: Entrer du texte	[numéro de ligne] E
I: Imprimer	[ligne début [ligne fin]] I [nombre de lignes] [N]
L: Insérer	[numéro de ligne] L
R: Remplacer	[ligne début [ligne fin]] [?] R/ Chaîne1 / Chaîne2 [/]
S: Supprimer	[ligne début [ligne fin]] S [nom de fichier [+]]
T: Terminer	
V: Visualiser	[ligne début [ligne fin]] V [nombre de lignes] [N]

Les conventions d'écriture sont les mêmes que celles des manuels d'utilisation. Vous pouvez constater que la traduction est très bonne, mis à part la commande "L" (insertion), dont je n'ai pas saisi le sens profond...

N'oubliez pas que si vous en avez assez, le seul moyen de remettre les messages en anglais (si si, il y a des masochistes) est d'effacer le fichier FRALEX. Plus tard, quand nous aurons 1 Mo de mémoire centrale, et que nous saurons faire les traducteurs commutables décrits par les IDS, nous pourrons envisager des solutions plus "douces". On peut toujours espérer ...

Voilà, je vous laisse là. Ne perdez pas le nord !

Pierre DAVID

APPROXIMATION DE FRACTIONS POUR HP71 (HP75)

Dans JPC No 14 (Mai 1984), je vous proposais un programme de calcul sur les fractions, en utilisant la notation Forth. Ce programme est malheureusement long, et donc encombrant pour les 16 pauvres petits Kilo-octets de cette merveilleuse machine qu'est le HP71B.

Mea culpa, donc. Depuis, je suis tombé sur un "truc et astuces" de l'Ordinateur Individuel. Je l'ai un peu compacté, car c'est dommage de gaspiller quelques demi-octets. Nous n'en avons que 2416955 à notre disposition ! Ce sous-programme a le mérite trouver la fraction correspondant (à peu de choses près) à un nombre réel donné. L'utilisation est des plus simples:

Faites [RUN], entrez votre nombre, puis [END LINE]. C'est fini, c'est très rapide.

Par exemple: -1.23456789 donnera: -100/81. Si nous effectuons maintenant la division, nous voyons que le résultat est précis à 10^{-6} près. Si vous voulez une précision plus grande, il suffit de modifier le CALL.

Quelques remarques à propos de INPUT: vous pouvez très bien rentrer 5/3+2/5 [ENDLINE]. Vous aurez ainsi le bon résultat (31/15). Ce programme est donc parfait pour remplacer tous les calculs sur les fractions !

Je ne peux pas vous détailler l'algorithme, mais sachez que son nom est "méthode des réduites successives". Le programme original est dû à M. Serge Boisse, et est paru dans l'OI No 52.

```
-----  
0010 INTEGER A,B @ REAL X @ INPUT X @ CALL FR(X,A,B,10^(-6)) @ DISP USING 'K,A,K';A,'/',B @ END  
0100 SUB FR(G,B,E,P)  
0110 H=G @ I=INT(H) @ B=I @ E=1 @ A=1 @ B=I @ IF NOT FP(ABS(G)) THEN 130  
0120 H=1/(H-I) @ I=INT(H) @ C=B*I+A @ A=B @ B=C @ F=E*I+D @ D=E @ E=F @ IF ABS(G-C/F)>P THEN 120  
0130 END SUB  
-----
```

J'ai placé un SUB, afin de faciliter l'appel de ce sous programme par des programmes extérieurs.

Ce fichier occupe 209 octets dans la mémoire du HP71B. Il est adaptable sur HP75. Pour ma part, il est à demeure dans ma machine.

A Bientôt

Pierre DAVID

CATALOGUE ETENDU SUR HP71B

Nom : XCAT version 2.0 (30/07/1984)
Auteur: Jean-Pierre BONDU
Objet : aide à l'édition des fichiers
(sauvegarde, rappel, effacement ...)

XCAT est un programme qui vous aidera à gérer vos fichiers, quel qu'en soit le support :

- i - Mémoire vive (Mev)
- ii - " " de Masse (Mst)

Le support est le type de mémoire sur lequel vous avez choisi de travailler. Ce peut être la Mev du 71, ou une mémoire de masse de la boucle HP-IL. Par la suite nous emploierons le terme 'Mst' pour mémoire de masse (de l'anglais 'Mass Storage').

FONCTIONNEMENT:

S'il y a plusieurs mémoires de masse sur l'IL, XCAT vous demande d'en choisir une. A partir de ce moment toutes les commandes disponibles sont indifférenciées, c'est à dire qu'elles accompliront la même action quel que soit le support courant. Ex: si vous avez sélectionné la Mst, la commande <E>ffacement détruira le fichier indiqué de l'unité de mémoire de masse (sélectionnée auparavant). On se déplace dans le CAT du support à l'aide des touches habituelles: [^], [v], [g][^], [g][v].

Toutes les commandes agissent sur le fichier affiché, en voici la liste.

COMMANDES: elles sont au nombre de 10

- 1- <E>: Efface le fichier.
- 2- <T>: Transfere (copie) le fichier du support courant vers l'autre
- 3- <R>: Rappelle le fichier indiqué de la Mst. Ne fonctionne que si le support courant est la Mev
- 4- <P>: Protection. Permet de 'Secure/Unsecure' le fichier.
- 5- <W>: 'Write All'. Recopie intégrale de la Mev sur la Mst.
- 6- <C>: Configuration. Affiche les caractéristiques du système.
- 7- <A>: Aide. Affiche le menu des commandes disponibles.
- 8- <S>: Support. Changement de support (Mev <-> Mst).
- 9- <F>: Fin. Fait le ménage !...
- 10- <I>: Initialisation. Revient à lancer le programme.

Note: Si une commande risque d'altérer un

fichier, confirmation vous en est demandée. Les remarques dans le programme suivent les lignes concernées.

MODE D'EMPLOI:

- Lancez le programme (si, si ...)
- Choisissez votre mémoire de masse (si vous en avez ...) Le témoin 'BUSY' clignote .
- Répondez aux questions.
- N'oubliez pas d'appuyer sur <F> pour en finir. C'est tout !...

VARIABLES:

- T: Nombre de Mst sur la boucle
- K: Adresse de la Mst sélectionnée
- D: Numéro d'ordre de la Mst sélectionnée
- P: " " du fichier affiché (courant)
- M: Nombre de fichiers sur le support courant
- M1: " " en Ram
- M2: " " sur la Mst
- D\$: Spécificateur d'unité de la Mst
- N\$: Nom du fichier affiché (courant)
- A\$: CAT comprimé de N\$
- K\$: Tampon clavier

DRAPEAUX:

- 5: (1)mémoire de masse sélectionnée
- 5: (0)pas de " " " "
- 6: cf.sous programme 'WALL' (ligne 2940)
- 8: (1)support courant= Mst (IL)
- 8: (0)support courant= Ram (71)

Au plaisir de vous lire...

Jean-Pierre BONDU (P#33)

PROGRAMME XCAT

```

=====
1000 'DEBUT':
1010 RESET @ DELAY 0
1020 DIM A$(42),D$(12),M$(5),N$(8),NO$(8),K$(4)
1030 INTEGER D,P,M1,M2,M,I,K,R,T
1040 DISP "Initialisation"
1050 GOSUB 'SELMASS'
1060 GOSUB 'INIT'
1070 GOSUB 'AFFICHE'

=====
1080 'MENU': R=0
1090 IF NOT KEYDOWN THEN 1090
1100 K$=KEY$
1110 IF K$=#50' THEN P=P-1 @ R=1
1120 IF K$=#51' THEN P=P+1 @ R=1
1130 IF K$=#162' THEN P=1 @ R=1
1140 IF K$=#163' THEN P=M @ R=1
1150 IF K$='E' THEN R=2
1160 IF K$='P' THEN R=3
1170 IF K$='T' THEN R=4
1180 IF K$='S' THEN R=5
1190 IF K$='A' THEN R=6
1200 IF K$='F' THEN R=7
1210 IF K$='C' THEN R=8
1220 IF K$='R' THEN R=9
1230 IF K$='W' THEN R=10
1240 IF K$='I' THEN 'DEBUT'
1250 IF R=0 THEN 'MENU'
1260 ON R GOSUB 'P','EFFACE','PROTEGE','TRANSF','SUPPORT','AIDE','FIN','CONFIG','RAPPEL','WALL'
1270 GOSUB 'AFFICHE'
1280 GOTO 'MENU'

=====
1290 'P': RETURN

=====
1300 'AFFICHE':
1310 IF FLAG(8) AND M2=0 THEN GOSUB 'SUPPORT'
    - S'il n'y a plus de fichiers sur la mémoire de masse, revenir à la Ram.
1320 IF P>M THEN P=M @ GOTO 1360
1330 IF P<1 THEN P=1 @ GOTO 1360
1340 IF FLAG(8) THEN A$=CAT$(P,D$) @ A$(9,10)=" ELSE A$=CAT$(P)
1350 A$(15,17)=" @ N$=A$(1,POS(A$,' ')-1)
    - Analyse du fichier pointé
1360 DISP USING "2D,':',19A";P,A$
1370 RETURN

=====
1380 'SUPPORT':
    - Changement de support

```

```

1390 IF D=0 THEN GOSUB 'ERR1' @ RETURN
  - Pas de Mst sur la boucle
1400 IF M2=0 THEN GOSUB 'ERR2' @ SFLAG B
  - Il y a une Mst, mais vierge de tout fichier valide
1410 I=FLAG(B,NOT FLAG(B))
  - Inversion du support
1420 DISP "Support: ";
1430 IF FLAG(B) THEN DISP "Mém. de masse" ELSE DISP "Mémoire vive" @ SEND LPD
1440 BEEP 999,.02 @ WAIT .5
1450 IF FLAG(B) THEN M=M2 @ RADIANS ELSE M=M1 @ DEGREES
1460 P=1
  - On sélectionne le premier fichier du nouveau support
1470 RETURN

```

```

=====
1480 'INIT':
  - Analyse la configuration du système
1490 GOSUB 'M1'
  - D'abord la Ram...
1500 IF D#0 THEN GOSUB 'M2'
  - Puis la Mst le cas échéant
1510 P=1 @ M=M1 @ CFLAG B
  - Ram sélectionnée - Système initialisé
1520 RETURN

```

```

=====
1530 'M1':
  - Nombre de fichiers en Ram
1540 FOR I=1 TO INF
1550 A$=CAT$(I)
1560 IF A$="" THEN M1=I-1 @ RETURN
1570 NEXT I

```

```

=====
1580 'M2': DIM B$(256) @ R=0
1590 SFLAG -23
  - La séquence 'EDL' (End Of Line) valide la fin d'une entrée sur la boucle
1600 SEND MLA TALK K DDT 2
  - MLA : My Listen Address - le 71 est récepteur
  TALK K: périphérique d'adresse K est l'émetteur
  DDT 2 : envoie l'enregistrement pointé
1610 ENTER :K ;B$
  - Le 71 reçoit l'enr. 0 (=directory)
  NOTE: l'ordre RESTORE IO (ligne 1800) avait placé le lecteur au début de l'enr. 0
1620 M2=NUM(B$(20))+256*NUM(B$(19))
  - Nombre d'enr. déclarés lors de l'initialisation
1630 ENTER :K ;B$
  - Saute l'enr. 1
1640 FOR I=2 TO M2+1
1650 ENTER :K ;B$
  - Stocke l'enr. I dans B$
1660 CALL FLPURGE(B$,R)
  - Compte (en R) le nombre de fichiers purgés
1670 J=POS(B$,CHR$(255))
  - Position de la fin du Directory

```

```

1680 IF J#0 THEN 1700
1690 NEXT I
1700 M2=8*(I-2)+(J-1)/32-R
  - R= Nbre de fichiers purgés   I-2= Taille du directory (en enr.)
  J= Permet de déduire le nbre de fichiers enregistrés (mais pas forcément valides)
  Rappelons qu'au niveau du directory:
  1 enr. contient les informations de 8 fichiers, et un fichier occupe 32 octets
1710 DESTROY B$
1720 SEND LPD
1730 CFLAG -23
  - 'EOL' non valide (cf 1590)
1740 IF M2=0 THEN GOSUB 'ERR2'
  - Pas de fichier valide
1750 RETURN

```

```

=====
1760 'SELMASS':
  - Sélectionne la mémoire de masse
1770 T=0
  - T= Nombre de mémoires de masse sur la boucle
1780 ON ERROR GOSUB 'SELMERR'
1790 RESET HPIL
1800 RESTORE IO
1810 FOR A=1 TO 30
1820 I=DEVAID(A)
1830 IF I<32 AND I>15 THEN T=T+1 @ GOSUB 'SELECT'
1840 NEXT A @ OFF ERROR
1850 IF T=0 THEN GOSUB 'ERR1' @ RETURN
  - Pas de Mst
1860 DISP T;"mémoire"&FNP$(T)&" de masse" @ WAIT 2
1870 DISP "Mst:"&STR$(D);TAB(10);"Adr:"&STR$(K);TAB(20);"OK"
  - Mst: numéro d'ordre de l'appareil dans sa catégorie
  Adr: adresse dans la boucle
1880 D$=":MASSMEM("&STR$(D)&")"
  - Spécificateur du périphérique
1890 RETURN

```

```

=====
1900 'SELECT':
  - Si une Mst a été trouvée, permet de la sélectionner
1910 IF FLAG(5) THEN RETURN
  - Si c'est déjà fait, inutile de recommencer
1920 DISP "Mst n "&STR$(T)&": (0) (N) ?"
1930 GOSUB 1970
1940 IF R THEN K=A @ D=T @ SFLAG 5 @ GOSUB 'CHKMASS' @ DISP "Sélectionnée" ELSE DISP "Suivante"
  - Initialise les variables si sélection
1950 OFF ERROR
1960 RETURN
1970 ON TIMER #1,10 GOTO 2040
1980 K$=KEY$
1990 IF K$='D' THEN 2040
2000 IF K$='N' THEN R=0 @ GOTO 2050
2010 SEND LISTEN A @ WAIT .5
2020 SEND UNL @ WAIT .5
  - Clignotement du témoin 'BUSY' sur le périphérique concerné

```

```
2030 GOTO 1980
2040 R=1
  - Si pas de réponse, le sélectionne
2050 OFF TIMER #1
2060 RETURN
```

```
=====
2070 'SELMERR': A=MOD(ERRN,255000)
2080 IF A=60 OR A=188 OR A=35 OR A=163 THEN 2100
2090 GOSUB 'ERR'
2100 RETURN
```

```
=====
2110 'CHKMASS':
  - La mémoire de masse est-elle prête ?
2120 R=SPOLL(A)
2130 IF R<16 OR R>31 OR R=23 THEN BEEP ON @ RETURN
  - Pas prêt:
2150 BEEP 500,.2
2160 BEEP OFF
2170 DISP "Unité";I;"inopérante"
  - Boucle jusqu'à ce que l'unité soit OK
2180 GOTO 'CHKMASS'
```

```
=====
2190 'EFFACE':
2200 ON ERROR GOTO 'EFFERR'
2210 IF NOT FLAG(8) THEN M#=ADDR$(N#)
  - Test la validité du fichier (# workfile, keys ...)
2220 DISP "Eff de: ";N#;
2230 IF FLAG(8) THEN DISP " -Mst" ELSE DISP " -Ram"
2240 BEEP 1000,.02 @ WAIT .7
2250 GOSUB 'CONFIRM'
2260 IF I THEN DISP N#;" éffacé" @ GOSUB 2300 ELSE DISP "Effacement Annulé"
2270 WAIT .5
2280 OFF ERROR
2290 RETURN
2300 UNSECURE FNS$(N#,D#)
2310 PURGE FNS$(N#,D#)
2320 M=M-1 @ IF P>1 THEN P=P-1
2330 IF FLAG(8) THEN M2=M ELSE M1=M
  - Réinitialise la variable M (nombre total de fichiers)
2340 RETURN
```

```
=====
2350 'EFFERR':
2360 GOSUB 'ERR'
2370 OFF ERROR
2380 RETURN
```

```
=====
2390 'CONFIRM':
2400 DISP "(C)onfirmmez (A)nnulez"
2410 IF KEYDOWN('A') THEN I=0 @ DISP @ RETURN
2420 IF KEYDOWN('C') THEN I=1 @ DISP @ RETURN
```

2430 GOTO 2410

=====

2440 'TRANSF':

2450 IF D=0 THEN GOSUB 'ERR1' @ RETURN

- Pas de mémoire de masse

2460 ON ERROR GOTO 'TRSFERR'

2470 IF FLAG(8) THEN COPY FNS\$(N\$,D\$) ELSE COPY N\$ TO D\$ @ SECURE N\$&D\$

2480 IF FLAG(8) THEN DISP N\$;" Copié en Ram" @ M1=M1+1 ELSE DISP N\$;" Sauvegardé" @ M2=M2+1

2490 IF FLAG(8) THEN M=M2 ELSE M=M1

- Cf ligne 2330

2500 BEEP 2000,.01 @ WAIT 1

2510 OFF ERROR

2520 SEND LPD

2530 RETURN

=====

2540 'TRSFERR': OFF ERROR

2550 SEND LPD

2560 Z=MOD(ERRN,255000)

2570 IF Z#16 AND Z#30 AND Z#144 AND Z#158 THEN GOSUB 'ERR' @ RETURN

- Erreur non prévue

2580 DISP "Fichier ";

2590 IF Z=16 OR Z=144 THEN DISP "protégé" ELSE DISP "existant"

2600 BEEP 500,.1 @ WAIT 1

2610 GOSUB 'CONFIRM'

2620 IF NOT I THEN DISP "Transfert Annulé" @ WAIT .5 @ RETURN

2630 DISP N\$;"Trsf confirmé" @ WAIT .5

- Tansfert confirmé

2640 UNSECURE FNC\$(N\$,D\$)

2650 IF FLAG(8) THEN PURGE N\$ @ M1=M1-1 ELSE M2=M2-1

2660 GOTO 2470

- Retour à la procédure de sauvegarde

=====

2670 'AIDE': DELAY INF,.15

2680 DISP " Appuyez sur une touche"

2700 DISP " Mst =mémoire de masse"

2710 DISP "Commandes disponibles:"

2720 DISP "<E>: Effacer"

2730 DISP "<T>: Transférer"

2740 DISP "<P>: Protection"

2750 DISP "<S>: Support (71/ Mst)"

2760 DISP "<W>: Wall (Sauve tout)"

2770 DISP "<R>: Rappel"

2780 DISP "<C>: Configuration"

2790 DISP "<F>: Fin"

2800 DISP "<A>: Aide !..."

2810 DISP "<I>: Initialisation"

2820 DELAY 0

2830 RETURN

=====

2840 'PROTEGE':

2850 DISP "(S)ecure (U)nsecure"

```
2860 IF KEYDOWN('S') THEN SECURE FNS$(N$,D$) @ RETURN
2870 IF KEYDOWN('U') THEN UNSECURE FNS$(N$,D$) @ RETURN
2880 GOTO 2860
```

```
=====
2890 'WALL':
2900 IF FLAG(8) THEN GOSUB 'ERR3' @ RETURN
    - Il faut que la Ram soit le support courant
2910 IF D=0 THEN GOSUB 'ERR1' @ RETURN
2920 N$=CAT$(0)[1,POS(CAT$(0)," ")-1]
    - Nom du programme en cours d'exécution
2930 DISP "(A)vec (S)ans Effacmt?"
2940 IF KEYDOWN('A') THEN SFLAG 6 @ GOTO 2970
2950 IF KEYDOWN('S') THEN CFLAG 6 @ GOTO 2970
2960 GOTO 2940
2970 ON ERROR GOTO 'WALLERR'
2980 P=1
2990 GOSUB 'AFFICHE'
3000 COPY N$ TO D$
3010 SECURE N$&D$
3020 IF FLAG(6) AND N$#N0$ THEN UNSECURE N$ @ PURGE N$ @ M1=M1-1 @ P=P-1
    - Efface le fichier sauvegardé (si demandé)
3030 M2=M2+1
3040 M=M1
3050 P=P+1 @ IF P<=M THEN 2990
3060 OFF ERROR
3070 SEND LPD
3080 DISP "Wall Terminé" @ BEEP 2000,.02 @ BEEP 2000,.02 @ WAIT .5
3090 RETURN
```

```
=====
3100 'WALLERR':
3110 IF ERRN=57 OR ERRN=1057 THEN 3050
    - Fichier non valide (=workfile, keys ...)
3120 IF ERRN=255016 OR ERRN=255144 THEN UNSECURE N$&D$ @ M2=M2-1 @ GOTO 3000
3130 GOSUB 'ERR'
3140 DISP "Wall Abandonné" @ WAIT 1
3150 OFF ERROR
3160 SEND LPD
3170 RETURN
```

```
=====
3180 'RAPPEL':
3190 IF FLAG(8) THEN GOSUB 'ERR3' @ RETURN
    - Ram doit être le support courant
3200 IF D=0 THEN GOSUB 'ERR1' @ RETURN
3210 IF M2=0 THEN GOSUB 'ERR2' @ RETURN
3220 INPUT "Fichier: ",N$
3230 IF N$="" THEN RETURN
3240 ON ERROR GOSUB 'RAPPERR'
3250 COPY N$&D$
3260 DISP N$;" Copié en Ram" @ WAIT 1
3270 OFF ERROR
3280 M1=M1+1 @ M=M1
3290 SEND LPD
```

3300 RETURN

```
=====
3310 'RAPPERR': OFF ERROR
3320 SEND LPD
3330 Z=ERRN
3340 IF Z#255016 AND Z#255144 AND Z#255030 AND Z#255158 THEN POP @ GOSUB 'ERR' @ RETURN
    - Erreur non prévue
3350 DISP N$;" existe déjà" @ BEEP 2000,.1 @ WAIT .5
3360 GOSUB 'CONFIRM'
3370 IF NOT I THEN POP @ DISP "Rappel Annulé" @ WAIT .5 @ RETURN
3380 DISP N$;": Rap confirmé"
3390 UNSECURE N$ @ PURGE N$
3400 COPY N$&D$
3410 M1=M1-1
3420 RETURN
```

```
=====
3430 'CONFIG':
3440 DELAY 2
3450 DISP T;"mémoire"&FNP$(T)&" de masse"
3460 IF D#0 THEN DISP "Mst: ";D;TAB(13);"Adr: ";K
3470 IF D#0 THEN DISP "Mst: ";M2;"fichier"&FNP$(M2)
3480 DISP "Ram: ";M1;"fichier"&FNP$(M1)
3490 DISP "Ram: ";MEM;"oct. libres"
3500 DELAY 0
3510 RETURN
```

```
=====
3520 'FIN': POP
3530 IF D#0 THEN SEND LISTEN K DDL 7 IFC LPD
3540 DESTROY ALL @ RESET
3550 BEEP 999,.1
3560 DISP "-Terminé"
3570 END
```

```
=====
4000 'ERR':
4010 DISP ERRM$
4020 BEEP 500,.1 @ WAIT 1
4030 RETURN
```

```
=====
4040 'ERR1':
4050 DISP "Pas de Mém. de masse"
4060 BEEP 500,.1 @ WAIT 1
4070 RETURN
```

```
=====
4080 'ERR2':
4090 DISP "Mém. de masse vierge"
4100 BEEP 500,.05 @ WAIT 1
4110 RETURN
=====
```

```
4120 'ERR3':
4130 DISP "Support non Valide"
4140 BEEP 500,.1 @ WAIT 1
4150 RETURN
```

```
=====
5000 DEF FNS$(N$,D$)
5010 IF FLAG(8) THEN FNS$=N$&D$ ELSE FNS$=N$
5020 END DEF
```

```
=====
5030 DEF FNC$(N$,D$)
5040 IF FLAG(8) THEN FNC$=N$ ELSE FNC$=N$&D$
5050 END DEF
```

```
=====
5060 DEF FNP$(N)
- Pluriel !....
5070 IF N>1 THEN FNP$="s"
5080 END DEF
```

```
=====
5090 SUB FLPURGE(C$,S)
- C$= 1 enregistrement de 'directory'
5110 FOR I=1 TO 225 STEP 32
5120 A=NUM(C$[I+10])+NUM(C$[I+11])
- A= Type de fichier
5140 IF A=510 THEN 5200
- A=510: fin de 'directory'
5160 IF NOT A THEN S=S+1
- A=0: fichier purgé
S: nombre de fichiers purgés
5190 NEXT I
5200 END SUB
```

Note: avant impression, J.P. BONDU, l'auteur de ce fabuleux programme, nous à fait parvenir par la voie téléphonique un certain nombre de modifications. Remplacer les lignes 2850 à 2880 par:

```
=====
2850 ON ERROR GOTO 2870
2860 IF POS ('ES',A$ 10,10 ) THEN UNSECURE FNS$(N$,D$) ELSE SECURE
FNS$(N$,D$)
2870 OFF ERROR
2880 RETURN
=====
```

Grâce à ceci, la touche agit comme une bascule pour la sécurisation du fichier. Il faut également rajouter la ligne:

```
=====
2565 IF Z=57 OR Z=1057 THEN DISP MSG$(67) RETURN
=====
```

nb.: pour ceux qui n'ont pas le fichier EDLEX, ils peuvent remplacer MSG\$ par la fonction FNM\$ décrite dans le JPC n°18 page 27.

PHILIPPE

PPC PARIS CHAPTER

ASSOCIATION REGIE PAR LA LOI DE 1901, ENREGISTREE A PARIS LE 2 DECEMBRE 1982 SOUS LE NUMERO 82/3240

BULLETIN D'ADHESION

NOM _____
 PRENOM _____ DATE DE NAISSANCE ____/____/____
 ADRESSE _____

 COMMUNE _____
 CODE POSTAL _____ PAYS _____
 TELEPHONE DOMICILE ____/____/____ BUREAU ____/____/____

PROFESSION _____
 INTERETS _____

MATERIEL HP EN VOTRE POSSESSION _____

AUTRE MATERIEL MICRO-INFORMATIQUE _____

COMMENT AVEZ-VOUS CONNU PPC PARIS CHAPTER ?

PUBLICITE _____ MAGAZINE _____
 AUTRE CLUB _____ HP _____
 RELATIONS, MEMBRES DU CLUB, AUTRES _____

QUE RECHERCHEZ-VOUS AU SEIN DU PPC PARIS CHAPTER ?

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

DATE ____/____/19____
 SIGNATURE, PRECEDEE DE LA MENTION "LU ET APPROUVE" _____

LE MONTANT DE LA COTISATION AU PPC PARIS CHAPTER S'ELEVE A 250.00 FF.
 PAIEMENT EFFECTUE LE ____/____/19____ A L'ORDRE DE PPC PARIS CHAPTER.
 PAR CHEQUE BANCAIRE N° _____ BANQUE _____
 CHEQUE POSTAL 3 VOLETS N° _____
 MANDAT LETTRE _____

EVENTUELLEMENT: JE M'ABONNE A COMPTER DU ____/____/19____
 JOINDRE A VOTRE INSCRIPTION UNE ENVELOPPE TIMBREE A VOTRE ADRESSE.

VEUILLEZ ENVOYER TOUTE CORRESPONDANCE A:
 MR PHILIPPE GUEZ, 56 RUE J.J. ROUSSEAU, 75001 PARIS (FRANCE)

Le journal JPC est un bulletin de liaison entre les membres du PPC PC. Le PPC PC est imprimeur et éditeur du JPC, son siège est à PARIS, au 56 rue jean-jacques Rousseau - 75001 PARIS
Directeur de la publication: Philippe GUEZ, President du PPC PC.

N° ISSN: 0762 - 381X