CHAPITRE CHHU

# JPC

DECEMBRE 85
JANVIER 86

NUMERO 30

Le numéro 35 FF.

## A PROPOS DU CLUB


| | | |
|---|---|---|
| PIERRE DAVID | 1 | EDITORIAL |
| PIERRE DAVID | 2 | PPC PARIS SE REUNIT |
| MICHEL MARTINET | 3 | AUGMENTEZ LA MEMOIRE DE VOTRE HP71B |
| PIERRE DAVID | 4 | LES INTERFACES VIDEOS HPIL |
| ALAIN GILLET | 6 | LE DOSSIER DE PROGRAMME |
| PHILIPPE GUEZ | 6 | PETITES ANNONCES |
| | 7 | COURRIER DES LECTEURS |
| | 8 | INFORMATIONS HEWLETT-PACKARD |


## HP71 BASIC


| | | |
|---|---|---|
| FRANCOIS LE GRAND | 10 | LE LOGICIEL NAVIGATION MARITIME |
| MICHEL WEIL | 11 | LES TABLES DE LA LOI |
| MICHEL WEIL | 12 | FORMAT HPAF |
| PHILIPPE GUEZ | 13 | CALENDRIERS |
| | 15-31 | PROGRAMMES |


## HP71 FORTH-ASSEMBLEUR


| | | |
|---|---|---|
| PIERRE DAVID | 34 | LE DEBUGGER FORTH |
| ALAIN HERREMAN | 41 | BREAK (FORTH) |
| ALAIN HERREMAN | 42 | SORT (FORTH) |
| STEPHANE BARIZIEN, PIERRE DAVID, ET MICHEL MARTINET | 42 | CHAMPS ASSEMBLEUR |
| MICHEL MARTINET | 43 | ASSEMBLEUR ET MEMOIRE DE MASSE |
| JANICK TAILLANDIER | 49 | L'ORDRE EXIT (FOR STEP EXIT NEXT) |
| JEAN-FRANCOIS GARNIER | 50 | LEX PPOLL ET DEMANDE DE SERVICE |
| MICHEL MARTINET | 55 | LE COIN DES LHEX |
| PHILIPPE GUEZ | 59 | RESUME DES ARTICLES DE L'ANNEE 1985 |
| LE CLUB | 62 | COOPERATIVE |


Edité par PPC-PC: 56, rue Jean-Jacques Rousseau - 75001 PARIS

# EDITORIAL

Bonne année à tous !

Voici la nouvelle année ! L'an de grâce mil neuf cent quatre vingt six sera une très grand cru pour PPC Paris et tous ses adhérents.

La fin de l'année passée fût riche en promesses de toutes natures. Notre progression s'accélère. Cela est sans doute dû à la confiance que de plus en plus nombreux vous nous accordez, ainsi qu'au contenu des articles qui ne cessent de nous parvenir. Et sans oublier bien sûr l'incroyable effort de l'équipe qui met en forme le Journal.

Vous pouvez désormais nous contacter par téléphone. Le répondeur est au (1) 42 33 87 73. N'hésitez pas à demander des renseignements sur le Club ou sur la technique.

Notre nom commence à être connu, et notre réputation commence à s'affirmer à l'étranger. Nous recevons de plus en plus de demandes de la part de nos voisins européens et d'outre-atlantique.

Bref vous l'avez compris, en ces périodes de fêtes, l'optimisme règne au sein du Club. Le prochain numéro inaugurera la couverture jaune. Belle couleur pour 1986, n'est ce pas ?

Pierre DAVID
(P37 SIG1 CHHU616)

## 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.

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

Les dates des prochaines réunions sont: (16h-19h)
- Le Samedi 1er février 1986
- Le Samedi 1er mars 1986
- Le Samedi 19 avril 1986
- Le Samedi 3 mai 1986
- Le Samedi 7 juin 1986

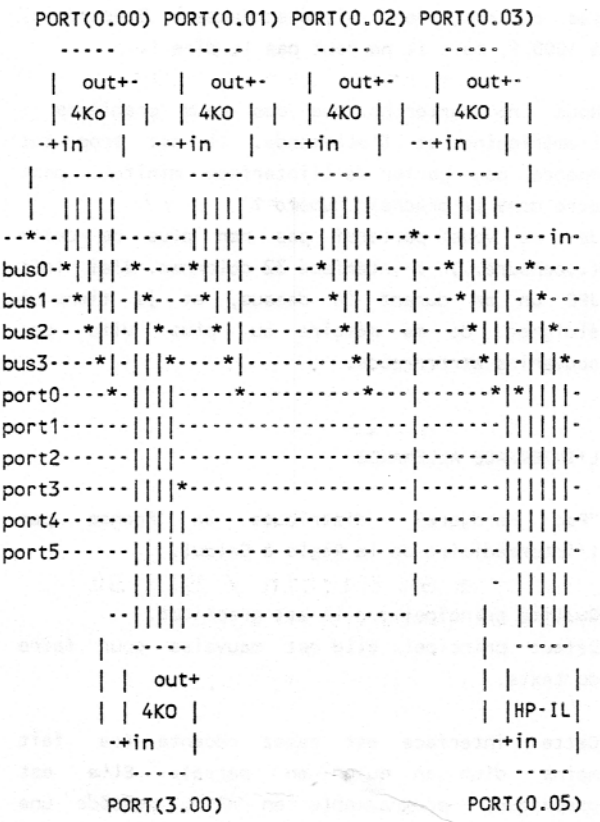
A Bientôt...
Pierre DAVID

AUGMENTEZ LA MEMOIRE DE VOTRE HP71B

I PRINCIPE

Le microprocesseur de notre ordinateur préféré peut adresser jusqu'à 512 ko.

Le schéma suivant vous indiquera comment est structuré le chaînage interne des RAM. Ce n'est pas le schéma exacte du système, loin de là, mais un simple plan synoptic simplifié.




Il existe 6 ports (connecteurs) pour le branchement de la mémoire au bus du HP71B.

- Le PORT 0 -
C'est le port interne du système, il contient les quatre modules internes de la mémoire de base (voir schéma).

- Les PORTS 1 à 4 -
Ce sont les 4 ports extérieurs se trouvant devant la machine sous le clavier. Le Port 1 n'est pas tout à fait identique aux trois autres, une de ses broches permet de générer une interruption pour inhiber les ROM internes du systèmes. En clair, nous avons la possibilité de mettre à cet endroit une ROM pouvant remplacer complètement le système d'exploitation du HP71B. C'est ce que fait la ROM diagnostic du service après vente.

- Le PORT 5 -
C'est l'emplacement du lecteur de carte. Il permet de connecter les modules de tailles plus grosses comme les nouveaux modules d'extension mémoire HHP.

II L'EXTENSION MEMOIRE INTERNE

De part la configuration du système, le HP71B peut recevoir jusqu'à 16 modules différents sur chacun de ses ports. Exemple: essayez FREE:PORT(4.07) ou FREE:PORT(0.15), le système vous renverra l'erreur "Device Not Found". Par contre lorsque vous dépassez l'argument ":PORT(X.15)", le système vous retourne l'erreur "Invalid Arg".

Les RAM sont toutes connectées de la même façon au bus mais pour que la configuration fonctionne normalement, elle sont reliées entre elles par ce que l'on appelle un Daisy Chain. En clair, elle sont branchées en parallèle sur le bus sauf deux de leurs broches qui les relies en séries. Ce sont les broches "Daisy In" et "Daisy Out". La solution pour augmenter la mémoire de Titan est donc de rajouter des modules dans la chaîne du port 0. Je suis arrivé à en rajouter 6 soit un total de 10 avec les quatre modules d'origine.

La même opération pourrait être faite sur les ports extérieurs mais cela obligerait à beaucoup trop de "bricolage". En effet, le câblage de ces ports et la connectique ne permettent pas de chaîner directement plusieurs modules. Il faudrait refaire (ou découper) quelques lignes sur la plaque du circuit imprimé. De plus, il se créé un trou entre le dernier module interne et la rom HPIL. Je pense que c'est le microprocesseur IL et son entourage interne qui occupe cette place mais sans grande conviction. Le même décalage existe avec plus de modules internes, actuellement il se trouve en PORT(0.11). Le dernier module interne occupe la position 0.09 (FREE:PORT).

Voilà quelques explications à propos de la mémoire du HP71. Un dernier point avant de nous quitter: que faire d'une si petite machine avec

une si grosse mémoire ? J'ai actuellement un HP71B dont la mémoire est poussée à 86 ko. Lorsque l'on désire faire un fichier Lex de 2 ko, il faut compter environ 30 à 40 ko de fichier source. En toute circonstance, je peux en plus charger en mémoire une FORTHRAM de 20ko contenant le debogueur de Pierre David et/ou l'assembleur rapide de Stéphane Barizien. Enfin, lorsque je n'ai pas d'imprimante, je fais le listing objet en mémoire centrale et il reste très souvent encore beaucoup de place.

Heureuse Programmation
Michel MARTINET (SIG#2, P#12, CHHU#617)

---

## LES INTERFACES VIDEO HPIL

Sans un écran, un ordinateur n'a pas grande crédibilité aujourd'hui. Essayez d'expliquer la puissance d'une HP41, d'un HP75 ou d'un HP71 à quelqu'un, vous vous heurterez toujours à des réflexions du genre :

- sans écran, moi j'pourrais pas !
- ça n'a que 12 (resp. 32, resp. 22) caractères, ton truc ?

Et je ne parle pas des gens qui demandent où est l'ordinateur...

Que répondre, face à de telles phrases catégoriques ? Qu'une HP41 peut se connecter à une interface vidéo ? C'est vouloir faire figure d'attardé mental, de menteur, etc... Vous connaissez certainement le problème, chers collègues HPistes convaincus.

Cependant, si l'on peut très bien se passer d'une interface vidéo (ne l'ais-je pas fait pendant des années ?), il est vrai que cela apporte un confort, nottament dès que l'on traite du texte, et plus spécialement lorsqu'on s'occupe d'un Club, qui édite un journal...

Néanmoins, tout le monde ne traite pas du texte, et on peut fort bien avoir besoin d'une interface vidéo pour développer du logiciel, pour présenter des résultats, que sais-je encore...

Ce mini-dossier a donc pour but de vous présenter les interfaces vidéo actuellement disponibles sur le marché.
Elles ne se bousculent pas :

A votre gauche, l'américaine (Moutain Computer - Hewlett Packard), et à ma droite, l'allemande (Pac Hardware, notez le jeu de mot subtil...). Au loin se profile l'outsider, la française, l'interface minitel. Derrière, loin derrière, on voit encore l'ERREUR de Hewlett-Packard : la HP82163B et ses 32 colonnes. Chut, elle n'existe plus ! Il n'y en a plus ? Non ! Même pour caler les chaises ? Non ! Si, chez Maubert Electronic, à 1000 F, mais il ne faut pas le dire !

Nous ne parlerons que des deux premières : l'américaine et l'allemande. Il est trop tôt encore pour parler de l'interface minitel, peut être dans le prochain numéro ?
Je ne vous parlerai pas non plus de cette (..censuré..) d'interface 32 colonnes. J'ai fait JPC un an durant là dessus, et je tiens à éloigner de ma mémoire au plus vite ces souvenirs atrrrrroces.

### L'INTERFACE ALLEMANDE

"Pac Hardware", distribuée en France par l'intermédiaire de la Règle à Calcul.

Qualité principale: elle est graphique.
Défaut principal: elle est mauvaise pour faire du texte.

Cette interface est assez récente (ça fait moins d'un an qu'on en parle). Elle est graphique, programmable en HPGL, possède une résolution de 640 points par 250, ce qui n'est pas mal du tout.

La liste des instructions HPGL est assez longue, je ne les donnerai donc pas toutes. Sachez simplement que l'échelle est définissable, ainsi que le type de ligne. Dans une moindre mesure, vous pouvez modifier la taille et l'inclinaison des caractères, et changer le type de tracé. Couleur se traduit par "tracé", "effacement", ou "exclusion", c'est à dire que tenter d'allumer un point qui l'est déjà provoque son extinction.

Tout ceci est modulable. Un certain nombre d'instructions qui ne sont pas de l'HPGL sont fournis, telle l'instruction pour vider le contenu de la page graphique sur la ThinkJet, ou les instructions pour gérer le vidage de la page graphique sur des imprimantes parallèles connectées directement sur l'interface, ce qui reste à mon avis une application rare.

Rien à dire de plus sur le coté graphique, mis à part que le tracé est assez lent. Mais c'est déjà bien d'avoir du graphique !

Du coté texte, c'est assez pitoyable. Les séquences d'échappement ne sont pas standard, du moins pas avec l'américaine, et comme déjà beaucoup de logiciels ont été écrits pour cette dernière, cela peut créer des problèmes.

Mais, ce qui est le plus grave, surtout pour des européens, est le manque total d'accents. Je sais bien qu'il y a déjà du graphique, et qu'il ne faut pas trop en demander, mais je trouve absurde de commercialiser un engin qui m'affiche un "E" en vidéo inverse à la place d'un "é". Pire encore : un "5" (cinq) en vidéo inverse à la place d'un "ç".

Précisons que d'autres séquences sont fournies pour gérer le port parallèle situé à l'arrière de la bête.

Signalons enfin que, comme sa collègue américaine, elle chauffe démesurément.

L'AMERICAINE

Mountain Computer (Oh pardon, Hewlett Packard HP92198B). Le B sigifie que c'est la version française. Il aura fallu près de deux ans pour que HP France décide de changer l'adaptateur secteur, et l'ajoute à son catalogue.

Principale qualité: Remarquable en texte
Principal défaut: Pas de graphique.

Nous ne nous attarderons pas longtemps sur le côté graphique de la chose. Sachez simplement qu'il est possible de changer les eproms pour avoir des caractères semi-graphiques. Adressez-vous au Club.

Mais, pour le texte, quel régal ! Plein de séquences d'échappement utiles, telles qu'insertion ou suppression de ligne, changement de police de caractères (vidéo inverse, ou Roman8). Des accents qui apparaissent comme des accents sur le moniteur !

Le jeu de caractères Roman8 est directement compatible avec les imprimantes HP82905B ou ThinkJet, c'est à dire avec toutes les imprimantes HP présentes ou à venir, y compris la LaserJet.

A noter une petite bogue, qui arrive lorqu'on insère des caractères sur la dernière ligne. Il se peut que le curseur se retrouve en haut de la page.

PERFORMANCES

Mitigées :

L'allemande est plus rapide que l'américaine en performances pures, mais elle ralentit la boucle HPIL. En effet, l'américaine travaille plus vite que l'HPIL, et ne ralentit donc pas les messages, comme par ailleurs la HP82905B et le lecteur de disquettes qui faisaient partie de la boucle au moment du test.

CHOIX

Allemande : environ 6000 FF
Indispensable pour le graphique

Américaine : environ 4000 FF
Indispensable pour le texte

La configuration idéale serait, pour moi, l'interface américaine et une table traçant HP. Savez-vous qu'une nouvelle vient de sortir ? La HP7450. Pas encore d'interface HPIL, d'après ce que j'ai compris.

A titre indicatif, l'interface minitel devrait arriver aux alentours de 3000 FF H.T. Pour ce prix là, 40 colonnes, et un modem. A suivre...

Pierre DAVID
(SIG#1, P#37, CHHU#616)

---

## LE DOSSIER DE PROGRAMME

Celui-ci doit comprendre:

Exposé du problème

Il faut exprimer succintement le problème afin de savoir comment le résoudre.
Masque d'écran: dessiner sur grille d'écran permet à l'auteur de simuler le travail tel qu'il devra s'effectuer.
L'organigramme: permet de matérialiser par bloc toutes les tâches que le programmeur doit prendre en charge. Il donne de plus une vision du programme permettant une bonne compréhemsion de celui-ci.
L'identification des variables: chaque programme nécessite un bon nombre de variables. Afin de s'y retourver, il faut faire l'inventaire de celle-ci. Une fiche d'identification des variables permet de gérer celle-ci.
Feuilles de programmation: elles ont un triple usage. Etablir les masques d'écran, supporter les lignes de programmation, donner en abscisse et en ordonnées les valeurs décimales de chaque point de l'écran.

Mise au point et maintenance

Mise au point: lorsque le dossier de programmation est constitué, l'introduction des séquences de traitement dans l'ordinateur peut commencer. La procédure d'introduction doit se faire par étape. Chaque test doit-être ponctué par la sauvegarde du programme sur disquette avant de passer à l'étape suivante.
Pour les séquences de traitement dont le test complet nécessite plusieurs sessions d'introductions des données, la construction d'un jeu d'essai simulant toutes les conditions rélles du travail doit-être envisagée. L'utilisation d'une imprimante permet de tester la bonne marche du programme et la bonne utilisation des masques d'écran.
La maintenance du programme: elle est aisée si le dossier programme est complet. Tenir à jour l'évolution de celui-ci en numérotant les différentes versions, archiver les sorties d'imprimante servant de test du programme, les modifications réellement effectuées.

Pour pouvoir utiliser un programme, il faut non seulement le programmer correctement mais établir un dossier sur celui-ci permettant de le suivre. Tout comme le docteur avec son patient. Ce texte ainsi que le précédent (Méthode de réalisation d'un programme) m'ont été inspiré du livre de Michel Benelfoul: La réalisation des programmes Editions du PSI.

Alain GILLET

---

Jean-Pierre FASSIOLA, 11 avenue des Lacs, 94100 St Maur (Tel.: 48 83 61 08 après 18h30). Vends une interface vidéo (TV MONITEUR) HP82163 et une imprimante thermique HP82162A (boucle interface). Matériel pratiquement neuf et prix à débattre.

Wilfrid FELIHO, 19 rue L. Casanova, 77270 Villeparisis (tel.: 64 27 53 37). Vends HP41CV + lecteur de cartes + HP82104A + imprimante HP82143A, le tout à 4800FF, état neuf. Vend d'autre part une très belle montre SEIKO neuve à 800FF.

François LEGRAND, rue Josse Impens 99, 1030 Bruxelles BELGIQUE; ou rue de la Poterie, Trévignon 29128 Trégunc FRANCE. Vends imprimante 82143 (2200FF) + lecteur de cartes 82104 (1200FF) + 2 XMEM 82181 (2*450FF) + 11 card holder avec 40 cartes 97-13141 (11*120FF) + 40 cartes 97-13140 (40*3FF) + pour HP67/97 (compatible HP41) nav pac (150FF) et math pac (150FF). Soit un total neuf de 6040FF. Le tout, soldé à 5000FF avec livres et programmes. Soit une réduction de 60% par rapport au prix neuf (tarif RAC). Vends également 80 microcassettes HP82176A pour lecteur HP82161A au prix de 90FF pièce! Ces cassettes sont neuves, d'origine HP et sous emballage d'origine scellé.

INDJEHAGOPIAN Jean-Pierre, 142 Grande Rue, 92380 Garches (Tel.: Dom. 30 38 38 00, Dom. 47 41 21 97). Vend une imprimante 24 colonnes HP82143 peu servi (+ rouleaux de papier impression noire) à 1100FF, un lecteur de

cassette HPIL HP82161A, neuf, jamais servi à 2700FF, une interface vidéo HP82163B à 1400FF, une PPCROM avec manuel d'utilisation, jamais servi à 600FF. Factures, modes d'emploi et emballages d'origine fournis.

Philippe DAVASE, 7 Rue Ho Chi Minh, 78500 SARTROUVILLE (Tel.: Dom. 39 13 51 67, Bur. 47 57 31 40 poste 451). Vends: une imprimante graphique GP100A Seikosha interfaçage Sté Sicape-Vincennes, Totalement interfacée boucle HPIL, compatible HP41-HP71-HP75 + interface centronic d'origine matrice 5*7, 116 caractères alpha+symboles graphique point par point, 80 caractères par ligne, entrainement papier par picots, papier de larg.=4,5 à 10 pouces, très peu servie et vendue à 3000FF. Une imprimante HP 24 colonnes, HPIL HP82162, vendue à 2500FF. Tout ce matériel peut-être échangé contre autre matériel HP compatible HP71B ou HPIL.
Vends d'autre part un micro-ordinateur SHARP MZ80B, 64k RAM utilisateur, 32k RAM graphique, processeur Z-80A (8 bits), écran 9 pouces, haute résolution graphique 320*200, affichage 40 caractères * 25 lignes ou 80 caractères * 25 lignes, gestion cassettes par soft, lecteur de cassettes intégré, 6 slots dans l'unité centrale, horloge intégrée, gestion totale du curseur, sortie son et haut parleur, clavier ASCII - QWERTY, majuscules et minuscules, symboles graphiques, 10 touches fonctions programmables, pavé numérique séparé. Un double lecteur de disquettes SHARP, disquettes 5"1/4, 2*350k, double face, double densité, 70 pistes, 16 secteurs/piste, soft sector. Une imprimante grahique bidirectionnelle 80P5B, 80 caractères/seconde, impression à aiguilles, entrainement à picots réglables, 230 sortes de caractères, matrice 9*8, 40-80-68-136 caractères/ligne, papier listing larg=4" à 10", fonction graphique intégrale, nombre de lignes/page programmable. Deux boites de rangement disquettes + 80 disquettes, 70 cassettes spéciales testées micro-informatique. Liste des programmes disponibles disquettes: Multiplan, Wordstar, Mailmerge, Spelstar, CP/M, Dbase2, Friday, Wordmaster, T/Maker, Mbasic + Expand + Compilateur, Forth, 5 Basics, F-Dos, Assembleur Z80, Propascal, Supercalc, Cobol, Walcore, Langage C, The Last One, Calcstar + Graphique, Datastar, Ascom, Langage PL1, Mu-Math, Mu-Simp, + 9 disquettes de jeux. Nombreux livres sur CP/M et Z80. Le lot vendu à 12000FF ou échange contre table traçante HP7470-HPIL ou tout autre matériel Hewlett-Packard.

Vincent HERLICQ (P81), Tel.: (1) 46 40 15 33. Vend périphériques HPIL en très bon état: 1 HP82161A (lecteur de cassette) de septembre 84 + 4 cassettes dont 1 neuve à 3000FF, 1 HP82162A (imprimante) de 1983 + 3 rouleaux de papier thermique à impression noire à 2200FF et 150 cartes magnétique pour HP41 (dans étuis) au prix de 120FF les 50 ou 300FF le tout.

Michel MARTINET 119, rue de Bagneux 92120 MONTROUGE - Vend HP75C + Visicalc + Formatteur de textes + très nombreux logiciels : 5000 FF . Tél. 43 36 12 05 Département Informatique.

---

Guy TOUBLANC
21, Rue Henri Becquerel
59120 LOOS-LES-LILLE

Remarques concernant le programme "COMPLEXE" (JPC 28) d'Alain Herreman:

1) On peut obtenir la commande supplémentaire de la pile "LSTX" en ajoutant:
- à la fin des lignes 60, 110, 350
@ L$=X$
- La ligne:
650 'LSTX' : T$=Z$ @ Z$=Y$ @ Y$=X$ @ X$=L$ @ GOTO 'SHOW'
- L'assignation:
KEY 'L','CONT LSTX':

2) Un avantage du module MATH du HP71B: la pile de commande mémorise toute suite d'opérations complexes enchainées donc permet les modifications sans avoir à réécrire le tout. Ce qui n'est pas le cas pour le programme "COMPLEXE". D'autre part, la remarque de Michel Weil (JPC n24 p46) concernant le programme "INTG" (JPC n23 p51) ne tient pas si on ajoute à la fin de la ligne 70:
*F (Ce qui semble être un oubli de frappe qui n'a pas été signalé par la suite.)

Guy Toublanc.

# Un nouveau système : l'instrumentation associée à l'ordinateur personnel

Hewlett-Packard présente un nouveau système de test et de mesure, dans lequel l'ordinateur personnel constitue une station de travail très souple accessible à une grande variété d'applications. L'instrumentation se présente sous la forme de différents modules faisant office de périphériques du PC.

### Compatibilité HP/IBM PC

Ces modules d'instrumentation sont compatibles avec un ordinateur HP à écran tactile (HP 150B ou HP 150 II), un IBM PC, PC/XT ou PC/AT (sous MS DOS™). La communication avec l'ordinateur s'effectue via une interface utilisateur intitulée PCIB (bus PC-instrumentation) supportant huit types d'instruments différents. Cette interface se met à l'intérieur du PC qui peut contenir plus d'une carte PCIB, selon le nombre de logements disponibles (quatre sur le HP 150 II).

### Conception modulaire

L'instrumentation associée au PC est constituée de huit types de modules différents, possédant leurs propres alimentations et inclus dans des boîtiers plastiques superposables :

- l'entrée-sortie numérique HP 61010AA,
- le multiplexeur à relais HP 61011AA,
- le double convertisseur N-A en tension HP 61012AA,
- le multimètre numérique HP 61013AA,
- le générateur de fonctions HP 61014AA,
- le compteur universel HP 61015AA,
- l'oscilloscope numériseur HP 61016AA,
- le relais de commande externe HP 61017AA

### Logiciel système

Le logiciel du système autorise son utilisation en mode manuel ou en mode programmé. En mode manuel, l'écran de l'ordinateur affiche, pour chaque module, un « panneau avant reconfigurable » reproduisant le panneau de commande d'un instrument traditionnel. Les réglages s'effectuent par effleurement de l'écran, à l'aide des touches de déplacement du curseur (HP), ou à l'aide d'une souris (IBM).

En mode programmé, il est possible de créer ses propres programmes d'application pour l'automatisation des instruments. Exemples d'applications : test de production, test de composants, contrôle d'entrée et enregistrement de données.

Le logiciel système inclut également le « Data Interchange Format » (DIF™), interface permettant de lui connecter un logiciel existant tel que Lotus 1-2-3™ ou VisiCalc®, pour le traitement des mesures en vue de compte-rendus ou de présentations graphiques.

### Logiciel en option

Le logiciel d'acquisition de données en option couvre les applications d'acquisition les plus courantes. Ce programme de contrôle par menu exécute la scrutation de tension, la scrutation de thermocouples et l'enregistrement de signaux. Il inclut également des routines de présentation graphique pour le traçage des données.

### Compatibilité HP-IB

Il est possible d'adjoindre des instruments compatibles HP-IB à un système basé sur un HP 150 ou un IBM PC : vous pouvez contrôler jusqu'à 15 instruments par carte interface HP-IB. Un ordinateur personnel peut inclure à la fois des cartes PCIB et HP-IB.

MS™ DOS est une marque déposée de MicroSoft Corporation.
DIF™ est une marque déposée de Software Arts Products Corporation.
Lotus™ 1, 2 et 3™ sont des marques déposées de Lotus Development Corporation.
VisiCalc® est une marque déposée de VisiCorp.

# La nouvelle imprimante ThinkJet est compatible avec de nombreux ordinateurs personnels

Il est possible d'utiliser le nouveau modèle d'imprimante ThinkJet de Hewlett-Packard avec de nombreux ordinateurs personnels fabriqués par Hewlett-Packard, IBM et Apple Computer. Le nouveau modèle HP 2225DB est une version de l'imprimante équipée d'une interface série RS-232-C et, comme les autres membres de la famille d'imprimantes ThinkJet, son fonctionnement est silencieux et elle exécute des travaux d'impression d'une excellente qualité a une vitesse élevée.



Grâce à ses nombreuses interfaces et au support logiciel vous pouvez utiliser l'imprimante ThinkJet sur la plupart des principaux ordinateurs personnels.

L'imprimante HP 2225DB fonctionne avec les ordinateurs personnels suivants :

- Gamme HP 150
- IBM PC, IBM PC AT et IBM PC XT
- Apple Macintosh, Apple IIc, Apple IIe.

Elle est également compatible avec les terminaux HP 2392A, HP 2623A, HP 2628A, HP 2627A et HP 2624B. (Sur les terminaux, la HP 2225DB ne permet d'imprimer que les textes car elle ne possède pas de caractères de tracé de lignes).

Vous pouvez dorénavant commander l'imprimante ThinkJet équipée de l'une quelconque de quatre interfaces. Le modèle HP 2225AB a une interface HP-IB (IEEE-488) destinée aux ordinateurs et appareils de mesure HP, la 2225BB utilise une interface HP-IL et se connecte aux calculateurs et aux ordinateurs portables HP. La 2225CB a une interface parallèle Centronics et vous pouvez la connecter aux ordinateurs personnels d'IBM, Apple, Compaq et Texas Instruments.

Un logiciel de commande spécial réalisé pour Hewlett-Packard par SoftStyle Inc, permet également d'utiliser l'imprimante avec Macintosh d'Apple pour les applications telles que MacPAINT et MacWRITE.

JetStart™ est une marque déposée de SoftStyle Inc.

## HP71 BASIC

| | |
|---|---|
| FRANCOIS LE GRAND | LE LOGICIEL NAVIGATION |
| MICHEL WEIL | LES TABLES DE LA LOI |
| MICHEL WEIL | FORMAT HPAF |
| PHILIPPE GUEZ | CALENDRIERS |

## LE LOGICIEL NAVIGATION POUR HP71B

Voici un ensemble de 3 programmes permettant de résoudre la plupart des problèmes de navigation maritime.
Ce logiciel reprend en gros les caractéristiques du module navigation de la HP-41, avec quelques avantages en plus, dont une exécution beaucoup plus rapide.

Voici un appercu des possibilités de ce logiciel:

- Tenue de l'estime
- Distance et cap par une loxodromie
- Position par une loxodromie
- Calculs orthodromiques
- Position par une orthodromie
- Trace de l'orthodromie sur la carte
- Navigation mixte ortho-loxo-ortho
- Droite de hauteur
- Ephémérides nautiques
- Identification d'étoile

Pour les 3 derniers cas, il y a possibilité d'utiliser 3 types d'éphémérides différentes:
- Ephémérides permanentes du logiciel PERPET
- Epimenides (éditées par le S.H.O.M.) du logiciel EPIMENIS
- Ephémérides Nautiques

L'accès à ces différentes possibilités se fait soit par menu, soit directement par la touche [g[ON]] (#155).
A chaque affichage de résultat, le programme suspend son exécution. Le voyant SUSP est alors allumé. Pour accéder à la suite du programme, tapez [f[+]] (CONT).

### FORMAT DES ENTREES ET SORTIES

Lors des questions, lorsqu'il y a une valeur entre le "=" et le "?", cette valeur correspond à la valeur par défaut si vous tapez ENDLINE directement. Si vous voulez changer cette valeur, tapez la nouvelle valeur sans effacer quoi que ce soit.
Les angles (coordonnées, hauteur d'un astre) peuvent être exprimés en Degrés, Minute,Seconde (DMS) si le drapeau 0 est baissé (indicateur 0 éteint) ou en Degré,Minute,Dixièmes de minute (DMT) si le drapeau 0 est levé (si l'indicateur 0 est allumé ).
Attention, si aucune valeur par défaut n'apparait, l'angle doit être donné en Degré,Minute,Dixième de minute (DMT).
Les coordonnées géographiques + pour Nord et Ouest, - pour Sud et Est. Vous n'êtes pas obligé de spécifier le signe +. D'autre part, vous pouvez spécifier le signe simplement en indicant N, S, E, ou W apreèsla valeur ( i.e 1925'3 Est pourra être tapé 19.253E ou -19.253).

### UTILISATION

La première fois, ou lorsque l'assignation de la touche #155 est effacée, faire "RUN NAVIG" [ENDLINE]. L'initialisation commence:
[Result " or '/10 ?]: répondre " ou ' ou rien. Selon votre réponse, les angles seront affichés en DMS ou DMT. Aucune réponse: aucun changement.
[Speroid Eccentricity = .08199188998 ?]: L'excentricité de la terre qui est proposée correspond à l'excentricité internationale. Une valeur de 0 correspond à une sphère parfaite.
[Elev. de l'Oeil = 0.0 ?]: si vous vous servez d'un sextant.
[Routes]: Ceci correspond à la première option du menu.

Le menu est divisé en 3 parties: menu principal, navigation astro, routes.

Le menu principal contient les options suivantes:
Routes
Navigation astro.
Initialisation (INIT)

Le menu [Routes] contient: (le code entre parenthèses est à utiliser pour l'accès direct).

- Tenue de l'estime
  (DR) pour Dead Reckoning
- Distance & Cap Loxo.
  (Rl) pour Rhumb-Line
- Position par une Loxo.
  (RLPOS) pour Rhumb-Line Position
- Calcul Orthodromiques
  (GC) pour Great Circle
- Position sur une Ortho.
  (GCPOS) pour Great Circle Position
- Trace de l'Ortho.
  (GCPLOT) pour Great Circle Plotting
- Navigation mixte
  (GCRL) pour Great Circle & Rhumb-Line
- Menu principal

Le menu Navigation astro contient:

- Droite de hauteur

(LOP) pour Line Of Position
- Ephémérides Nautiques
(ALMAC) pour ALMAnaC
- Menu principal

Utilisation de la touche [g[ON]]:
- Sans aucun code affiche au préalable, vous entrez dans le menu à la dernière option choisie (pas de retour au début du menu).
- En affichant préalablement un des codes d'accès direct (ie. DR [g[ON]]) vous entrez directement dans le sous programme choisi, sans passer par le menu.

Utilisation des différentes éphémérides:

Le choix de l'éphéméride servant au calcul se fait à la question [Which body : ?] . Répondez par le nom (voir note) de la planète, astre ou étoile que vous avez observé au sextant, puis, si nécessaire par le nom de l'éphéméride que vous desirez utiliser (la premiere lettre suffit).

Nom des 3 éphémérides:
PERPET pour les éphmérides du programme
EPIMENIDES
INTERPOLATEUR pour les éphémérides nautiques

Si vous ne spécifez aucune éphéméride, le calcul se fera avec PERPET s'il est en mémoire, ou avec EPIMENIDE s'il est en mémoire. Si aucun de ces deux programmes ne se trouve en mémoire, alors le calcul se fera avec l'INTERPOLATEUR.
Exemple. [Which body : SIRIUS,E] indique que le calcul se fera avec les épiménides.
Les épiménides et l'interpolateur sont ce qu'il a de plus précis (1/10 de minute d'arc). Le logiciel PERPET offre généralement une précision généralement supérieure à une minute d'arc.
Le logiciel PERPET contient les éphémérides du soleil (SUN), de la lune (MOON), de 4 planètes (MARS JUPITER VENUS SATURN) et de 57 étoiles (celles des tables américaines).
Le logiciel EPIMENIS dispose d'une sous-routine permettant de stocker les paramètres mensuels. Cette sous-routine est appelée automatiquement quand ces paramètres ne sont pas déjà en mémoire. Cette sous-routine peut-être appelée du clavier par "RUN EPIMENIS".
L'option identification d'étoile se fait également à la question [Which body:] à laquelle il suffit de répondre par "?" Si l'éphéméride de travail est PERPET, après 4 minutes de travail, s'affichera par ordre de brillance, une liste d'étoiles avec pour chaque étoile, l'écart en hauteur et l'écart en azimut.

Voilà en vrac quelques indications succintes relatives à l'usage de mon programme de navigation. Pour plus de détails, ou pour vos réactions ou commentaires, écrivez moi:

François LE GRAND, rue de la poterie
Trevignon , 29128 TREGUNC FRANCE

NOTE: Vous avez la possibilité de n'introduire que le début du nom de la planète. (Exemple: JUPIT ou JUP pour JUPITER)

---

## LES TABLES DE LA LOI

Lorsque Moïse descendit le Sinaï, il vit ses coreligionnaires danser autour d'un IBM-AT (disque dur: 40 G.O.), la colère le saisit, puis il se calma et brandit son HP-71B contenant les tables de la loi. Ceci eut l'effet escompté: un peu de calcul des probabilités rendirent un peu plus humble les pêcheurs.

Voiçi, donc l'utilisation de ces fameuses tables: il s'agit des lois du KI2, de STUDENT, et de FISHER-SNEDECOR. (Quand à la loi NORMALE il est normal que je n'en fasse pas état). Certains des sous-programmes de LOISPROB ont paru dans JPC nº 27 (avec un petit bog). Les re-voiçi corrigés et surtout complètés par la relation dans l'autre sens.
Dans le premier sens les sous-programmes calculent:
dºliberté(s), X:réel(>0) ----> 1-proba([X,INF[) (suivant le cas il y a 1 ou 2 degrés de liberté, X est un nombre réel positif et on calcule la probabilité d'être supérieur à X).
Dans l'autre sens le calcule donne:
dºliberté(s),Q --->X: nombre tel que P[X,INF[=Q (toujours, suivant les cas, il y a 1 ou 2 degrés de liberté, Q est un nombre compris entre 1/2 et 1 et le résultat X est un réel tel que la probabilité d'être plus grande que X vaut Q).
Dans un sens j'utilise modérement le module

MATH (on pourrait presque s'en passer), et dans l'autre sens ce module est INDISPENSABLE. Dans le sous-programmes PROBAKI2 les lignes 620 à 655 ont été obtenues par ESTIMATIONS de méthode de régression linéaires. En effet, le problème était de résoudre des équations integrales, et, vu le temps (long) mis pour résoudre ces équations, il était nécessaire d'initialiser la recherche des solutions. D'où ces estimations.
Une autre difficulté est venue du fait que la méthode de Newton, utilisée dans FNROOT (module MATH oblige) converge (si elle converge) obligatoirement à la préçision 10^(-12), alors que 10^(-2) suffit (pour avoir d'ailleurs une préçision finale de 10^(-4) (hé-oui). D'où le trafic semi-récursif des fonctions FNK1,FNK2 et FNF1,FNF2.

SYNTAXE:

```
CALL TPROBA(N,T,Q) @ DISP Q
CALL KI2PROBA(N,X,Q) @ DISP Q
CALL FPROBA(N,M,F,Q) @ DISP Q
```

donnent le sens direct pour respectivement les lois de STUDENT(à N dº), du KI2(à N dº), et de FISHER-SNEDECOR (à N,M dº).

Le sens inverse sera donné par:

```
CALL PROBAT(N,Q,X) @ DISP X
CALL PROBAKI2T(N,Q,X) @ DISP X
CALL PROBAF(N,M,Q,X) @ DISP X
```

Voilà, essayez. Vous pouvez inclure ces lois dans vos programmes. C'est vrai que le calcul est parfois longuet (surtout pour FISHER-SNEDECOR), mais il ne fallait pas oublier vos tables. Tout le monde n'est pas comme Moïse.

Michel WEIL

---

## Format HPAF

Certains modules ou ROM du HP71 B utilisent le format HPAF pour les fichiers DATAs de tableau de données. Qu' est ce que ce format? Ce sont des fichiers DATAs d'un tableau matriciel A(RxC) contenant:

1) à l'enregistrement nº 0: une chaîne de la forme

"HPAFNNSNSS..."

où NNSNSS... représente la forme de la ligne courante de A (Ici: Nbre, Nbre, Shaîne, Nbre, Shaîne, Shaîne...).

2) A l' enregistrement nº 1: le nbre de lignes de la matrice.

3) A l'enregistrement nº 2: le nº E de l'enregistrement à partir duquel on ferra des descriptions (ou encore l'on se racontera des petites histoires), appelé bloc descriptif.

4) A partir de l'enregistrement nº 3 et avant celui nº E, il y aura bien sûr les données de A.

Le PROBLEME est le suivant: les modules dont je parlais plus haut (par ex. lissage de courbe (module, qui s'il est bien par certains aspects, est à revoir pour d'autres points), etc...) demandent l'entrée de données (par ex. à la main) et mettent automatiquement ces données sur fichier data aux normes HPAF. D'où deux ennuis:

1■ Le ss-pgm (label + gosub-return) de mise à ce format est dans un trés, trés grand SUB-pgm, donc non accesible de l'extérieur de ce pgm.

2■ Si au contraire on est à l'intérieur de ce pgm, que l'on a des résultats intermédiaires (par exemples un vecteur ou un matrice), COMMENT les conserver? (en format HPAF ou non) (le pgm est en SUSP) (Notons V(,) ce résultat [si c'est un vecteur n'oubliez pas de le redimensionner par DIM V(1,n)]).

Il y a deux façons de faire:

a) La machine étant en mode SUSP dans un SUB-pgm la 'seule' possibilité est de taper: CREATE DATA TATA @ ASSIGN #28 TO TATA @ PRINT #28;V et [f] [ENDLINE].

b) Vous voulez le sauver au format HPAF dans le fichier F$, alors vous taper

```
CALL SAVEHPAF(V(,),N$(),F$)
```

Et c'est fait.

SYNTAXE:

Revenons aux format HPAF: Pour palier aux inconvénients préçités j'ai transformé le label gosub-return H-P correspondant afin d'en faire deux SUB-programmes autonomes, appelable de n'importe où:

1º Pour sauvegarder, dans le fichier data F$, des données contenues dans le tableau A(RxC) vous taper

```
CALL SAVEHPAF(A(,),N$(),F$)
```

N$() est le vecteur OPTIONNEL portant le nom des colonnes (s'il est vide, il est vide, mais mettez le quand même comme paramètre) (et si c'est le nom

des lignes qui vous intéresse veuillez transposer la matrice A).

2º Pour rappeler les données contenues dans un fichier DATA F$ au format HPAF, c'est tout facile: vous tapez

```
DIM A(1,300),N$(20)
CALL LOADHPAF(R,C,A(,),N$(),F$)
```

et le pgm vous sort: R,C,A(,),N$().

ATTENTION n'oubliez pas de pré-DIMENSIONNER la matrice A (n'IMPORTE COMMENT, pourvu que se soit supérieur à la dimension finale de A: DIM A(n,m) @ DIM A(1,n*m) (pour normaliser la lecture du fichier DATA. Compris? (idem pour N$()).

3º En prime si le tableau A contient DEJA des données et que vous voulez le compléter par d'autres données provenant d'un fichier DATA F$, alors vous surdimensionner A et N$ et la MEME SYNTAXE vous donne le tableau complété.

Ceci permet d'ailleurs (en utilisant maintenant SAVE(A(,),F$) d'augmenter l'ancien fichier DATA F$!!

4º Le format HPAF est relativement ouvert: j'ai choisit le format

HPAFNN..N + bloc descriptif(noms de COLONNES)

5º J'aurai pu, (ce que je fait d'ailleurs souvent), ne pas utiliser le bloc déscriptif, pour pouvoir nommer les LIGNES directemment. Pour cela, il suffirait de prendre le format

HPAFNNN..NNS

où S serait le nom (optionnel) de la ligne des NNN...NN. (S peut être d'ailleurs n'importe où: NNSN...N irait aussi, mais il ne faudrait qu'UN S). Le changement est facile à faire pour ce qui est du sub-pgm SAVEHPAF (au besoin demandez moi le listing (trés petit)). Pour charger ou lire un tableau par LOADHPAF c'est DEJA FAIT!

6º A tout hasard j'ai joint un SUB programme AIDEHPAF d'aide au formattage HPAF. Syntaxe:

CALL AIDEHPAF

7º Je suis l'heureux possesseur du module MATH (qui a part l'assembleur-FORTH) est le module le plus utile et que je trouve TRES BON. Par conséquent je l'ai utilisé par: MAT A=B, MAT B=ZERO, UBND(A,i) (c'est la DIMension de la ligne ou de la colonne de la matrice A). Mais le même PGM se bâtit sans ces puissants artifices.

8º Sans les remarques, sans AIDEHPAF, sans la fonction-réponse OUINON (qui a besoin de KEYWAIT (qui devraiENT être à demeure dans votre machine), il y a environ 1550 octets.

Bon, c'est tout, bonne nuit,

Michel WEIL

---

## UN CALENDRIER POUR LES LECTEURS DU FAMEUX PPC PARIS

Le programme que je vous propose aujourd'hui, calcule les différents paramètres de tracé d'un calendrier sur THINKJET. Le programme proprement dit (CALEND), tient une place en mémoire de 4382 octets. Puis il a besoin de 240 octets pour le fichier FETES et de 878 octets pour le fichier LUNES. D'autre part, il met 776,61s à l'exécution. Enfin, pour pouvoir l'utiliser il vous faudra les fichiers lex EDLEX et KETWAIT.

Le programme fait la recherche des principales fêtes de l'année (sous forme d'un fichier pour les fêtes fixes et calcule la date de Pâque pour les fêtes mobiles) par les routines SUB DFETES et SUB PAQUES. Il calcule aussi la date et l'heure des phases de la lune avec une assez bonne approximation (à 2mn près par rapport aux valeurs qui m'ont été transmises par le bureau des longitudes).

Tous les paramètres et les équations, ont été tirés du livre de Jean Meeus (Astronomical Formulae for Calculators).

Le mode d'emploi est très simple. Vous prenez un HP71B, vous y branchez une boucle d'interface (HPIL) et à l'autre extrémité de cette boucle, une imprimante THINKJET (n'oubliez pas de mettre du papier dans l'imprimante). Puis vous lancez le programme. Un premier arrêt demande l'année pour laquelle vous désirez le tracé de votre calendrier (le 71 prendra 1986 par défaut). Puis au premier bip, le calculateur rempli le fichier FETES, aux 12 bips suivant, il rempli le fichier LUNES et au dernier bip, il trace le calendrier.

Les différents messages affichés par le HP71, sont:

```
Année?= 1986
FETES
Lunes mois nr 1
Lunes mois nr 2
Lunes mois nr 3
Lunes mois nr 4
```

```
Lunes mois nr 5
Lunes mois nr 6
Lunes mois nr 7
Lunes mois nr 8
Lunes mois nr 9
Lunes mois nr 10
Lunes mois nr 11
Lunes mois nr 12
Tracé du calendrier
```

Attention: Si vous lancez le programme près de minuit, vous risquez d'avoir un petit problème. La ligne 5010 stocke la date courante dans une variable (A$), et ne la restitue qu'à la ligne 5040 par SETDATE A$. Si minuit passe à ce moment là, A$ ne sera pas incrémenté d'un jour, et la date courante sera désormais fausse.

Le tracé ne peut être fait pour une année antérieure au calendrier grégorien, soit avant 1582 (inclus). J'ai utilisé une méthode fort simple de calcul de dates fréquement utilisée en calcul mental.

Pour ceux qui veulent briller en société: vous devez vous souvenir d'un nombre: 033614625035, où chaque chiffre correspond à un mois "janvier=0"; "février=3"; ... Pour vous souvenir de ce nombre, vous attribuez 0 à janvier, puis 0 + 31 (nombre de jours de janvier) = 31, puis MOD(31,7) = 3, vous attribuez ce 3 à février puis vous faites 3 + 28 =31, et MOD(31,7) = 3 que vous attribuez à mars, ... Supposons que nous voulons trouver le jour du 25 décembre 1985, nous devons ajouter 4 nombres : 25 + 85 + 5 (numéro de décembre) + 21 (qui est la INT(85/4)) le résultat est donc 136 on calcule ensuite 136 MODulo 7, et nous trouvons 3. L'année 1900 commençant un lundi, on compte : 1=> lundi, 2=> mardi, 3=> mercredi. Le 25 décembre 1985 est donc un mercredi. Pour un siècle antérieur à 1900, il faut ajouter 2 à la somme des 4 nombres par siècle (2 pour 1800, 4 pour 1700 ...).

La dernière ligne du programme est une fonction qui permet le calcul d'un arrondi. Nous en avons besoin à la ligne 5050, la variable K doit être arrondie à 0,25 près pour que le calcul donne une valeur exacte.

Fonction ROUND:
```
DEF FNR(X,N)
FNR=IP(X*N+.5)/N
END DEF
```
vous permet d'arrondir un nombre X à N près.

Et voilà, ma prose est terminée. J'espère que le calendrier que j'ai tracé à votre intention, vous servira toute l'année (pour penser à nous envoyer des articles pour votre journal préféré). Les membres du bureau se joignent à moi pour vous souhaiter un

JOYEUX NOEL et une
BONNE NOUVELLE ANNEE.

Philippe

**Janvier 1986**

| | | |
|---|---|---|
| MERCREDI | 1 | (Jour de l'an) |
| Jeudi | 2 | |
| Vendredi | 3 | DQ à 19h48mn |
| Samedi | 4 | |
| DIMANCHE | 5 | |
| Lundi | 6 | |
| Mardi | 7 | |
| Mercredi | 8 | |
| Jeudi | 9 | |
| Vendredi | 10 | NL à 12h24mn |
| Samedi | 11 | |
| DIMANCHE | 12 | |
| Lundi | 13 | |
| Mardi | 14 | |
| Mercredi | 15 | |
| Jeudi | 16 | |
| Vendredi | 17 | PQ à 22h14mn |
| Samedi | 18 | |
| DIMANCHE | 19 | |
| Lundi | 20 | |
| Mardi | 21 | |
| Mercredi | 22 | |
| Jeudi | 23 | |
| Vendredi | 24 | |
| Samedi | 25 | |
| DIMANCHE | 26 | PL à 0h33mn |
| Lundi | 27 | |
| Mardi | 28 | |
| Mercredi | 29 | |
| Jeudi | 30 | |
| Vendredi | 31 | |

NAVIG

- Programme de navigation

Auteur : François LE GRAND
Rue de la Poterie Trevignon 29128 TREGUNC F

L'utilisation de ce prgm n'engage que l'utilisateur qui, en navigation, doit garder son esprit critique.

```
================================================================
100 'INIT': DELAY 2 @ 'SECD': DISP 'Results  " ";" or  '/10"; @ USER OFF @ INPUT ' ','?';A$
104 STD @ IF A$='?' THEN 112
108 IF A$='"' OR A$[1,1]="'" THEN A=FLAG(0,NUM(A$)-34) ELSE 100
112 USER ON @ 'EXC': E9=.08199188998 @ DISP 'Spheroid Eccentricity =';E9;
116 ON ERROR GOTO 120 @ INPUT '','?';E9

================================================================
120 'EYE': A=E0 @ DISP "Elev. de l'Oeil = "; @ GOSUB 'INPMT' @ E0=A
124 DEF KEY '#155','IFDISP$=""THENRUNNAVIG,"A"&M$[2,2]ELSERUNNAVIG,DISP$':
128 INTEGER N,I,J,S,M,Y @ SHORT Z,Z1,D1,D2 @ DIM M$[2],A$[22],B$[15]

================================================================
200 'A': 'AP': N=POS('RAM',M$[2,2]) @ M$='MP' @ N=N*(N>0)+1*(N<1)
204 ON FNN(N,3) GOTO 212,220,100

================================================================
208 'MP': DATA Routes,Navigation astro.,Initialisation
212 M$='MR' @ N=1

================================================================
216 'AR': N=FNN(N,8) @ ON N GOTO 'DR','RL','RLPOS','GC','GCPOS','GCPLOT','GCRL',200
220 M$='MA' @ N=1

================================================================
224 'AA': N=FNN(N,3) @ ON N GOTO 'LOP','ALMAC',200

================================================================
300 'MA': 'LOP': DATA Droite de hauteur,Ephemerides Nautiques,Menu principal
304 A$='D.R.' @ GOSUB 'IN1' @ CFLAG 11
308 GOSUB 'INPDT' @ GOSUB 'INPX' @ GOSUB 'REF' @ GOSUB 'ALM' @ P=A0-G1 @ CALL SRT(D0,L1,P,H2,Z)
316 IF B$[1,4]='MOON' THEN H2=H2-VAL(A$)*COS(H2) @ D8=ATAN(D8/(1/TAN(VAL(A$))-SIN(H2))/VAL(A$))
320 FIX 1 @ C=(H2-D8-H)*60 @ DISP B$;' Int. =';ABS(C);CHR$(65*(C>=0)+84*(C<0)); @ STD
324 A=Z @ DISP TAB(25+LEN(B$));'Az. = '; @ GOSUB 'DPC' @ DISP @ PAUSE @ GOTO 308

================================================================
350 'ALMAC': GOSUB 'INPDT' @ SFLAG 11 @ GOSUB 'ALM' @ GOSUB 'DSPCO' @ PAUSE @ GOTO 350

================================================================
1000 'MR': 'DR': DATA "Tenue de l'estime"
1004 GOSUB 'RL2' @ A$='D.R.' @ GOSUB 'DSP2' @ GOSUB 'SP' @ GOTO 'DR'

================================================================
1020 'RL': DATA Distance & Cap Loxo.
1024 GOSUB 'INP2' @ GOSUB 'RL1' @ GOSUB 'DSPDC' @ GOSUB 'SP' @ GOTO 'RL'

================================================================
1050 'RL1': GOSUB 'MA2' @ SFLAG -1 @ Z=ATAN(RED(G1-G2,360)*PI/180/(LN(A/B)+(G1=G2)))
1054 IF NOT COS(Z) THEN D1=ABS(RED(G1-G2,360))*COS(L1) ELSE D1=(L2-L1)/COS(Z)
1058 Z=MOD(Z+180*(D1<0),360) @ D1=ABS(D1*60) @ RETURN
```

```
============================================================
1100 'RLPOS': DATA Position par une loxo.
1104 GOSUB 'RL2' @ A$='Arr.' @ GOSUB 'DSP2' @ GOSUB 'SP' @ GOTO 'RLPOS'

============================================================
1150 'RL2': A$='Dep.' @ GOSUB 'IN1' @ GOSUB 'INPC' @ GOSUB 'INPD' @ L2=RED(L1+D1*COS(Z)/60,180)
1154 IF NOT COS(Z) THEN G2=RED(G1-D1*SIN(Z)/60/COS(L1),360) @ L2=L1 @ RETURN
1158 GOSUB 'MA2' @ SFLAG -1 @ G2=RED(G1-LN(A/B)*TAN(Z)*180/PI,360) @ CFLAG -1 @ RETURN

============================================================
2000 'GC': DATA Calcul Orthodromiques
2004 GOSUB 'INP2' @ CALL SRT(L2,L1,G2-G1,A,Z1) @ D2=(A-90)*60 @ A=D2 @ GOSUB 'DPD'
2008 A=Z1 @ DISP TAB(25);'Initial Head. = '; @ GOSUB 'DPC' @ DISP @ PAUSE @ A$='Vrtx.'
2012 A=ACOS(ABS(SIN(Z1))*COS(L1))*SGN(COS(Z1)) @ GOSUB 'DSPL' @ DISP TAB(25);
2016 SFLAG -1 @ A=G1-ATAN(1/(TAN(Z1)*SIN(L1)))+180*(L1*COS(Z1)<0) @ CFLAG -1
2020 GOSUB 'DSPG' @ DISP @ PAUSE @ GOSUB 'RL1' @ A=D1-ABS(D2) @ DISP 'Gain / '; @ GOSUB 'DPD'
2024 DISP @ PAUSE @ GOSUB 'GC1' @ PAUSE @ GOTO 2000

============================================================
2100 'GCPOS': GOSUB 'INP2' @ DATA Position sur une Ortho
2104 I=99 @ GOSUB 'GC2' @ L2=L @ G2=G @ GOTO 2104

============================================================
2200 'GCPLOT': DATA "Trace de l'Ortho."
2204 GOSUB 'INP2' @ GOSUB 'GC1' @ GOTO 2204

============================================================
2208 'GC1': I=1 @ GOSUB 'GC2' @ RETURN

============================================================
2300 'GC2': L=L2 @ G=G2 @ INPUT 'Long.(+) or Dist.(-) ','?';C @ IF C>0 THEN SFLAG 5
2304 FOR I=I TO 99 @ DISP ' Working...' @ CALL SRT(L,L1,G-G1,A,Z1) @ IF FLAG(5) THEN 2324
2308 CALL SRT(90+C/60,L1,Z1,L2,A) @ G2=G1-ABS(RED(A,360))*SGN(SIN(Z1))
2312 A$=STR$(I)&'.' @ IF I=99 THEN A$='Int.'
2316 IF RED(G2-G1,360)*RED(G-G2,360)<0 THEN 2336
2320 GOSUB 'RL1' @ GOSUB 'DSPDC' @ DISP @ PAUSE @ GOSUB 'DSP2' @ GOSUB 'SP' @ NEXT I @ RETURN
2324 IF I=1 OR I=99 THEN A=-SGN(SIN(Z1))*C @ C=FNR
2328 B=C @ IF I=1 AND FP(B)=0 THEN B=B*INT((G1+B)/B)-G1
2332 G2=G1+B @ L2=ATAN((SIN(L1)*COS(B)-SIN(B)/TAN(Z1))/COS(L1)) @ GOTO 2312
2336 G2=G @ L2=L @ GOSUB 'RL1' @ GOSUB 'DSPDC' @ CFLAG 5 @ A$='Arr.'
2340 DISP @ PAUSE @ GOSUB 'DSP2' @ RETURN

============================================================
2400 'GCRL': DATA Navigation mixte,Menu principal
2404 GOSUB 'INP2' @ CALL SRT(L2,L1,G2-G1,A,Z1) @ G=ACOS(ABS(SIN(Z1))*COS(L1))*SGN(COS(Z1))
2408 A$='Max.' @ A=G
2412 GOSUB 'IL' @ L=A @ IF ABS(G)<ABS(L) THEN DISP 'Max.Lat. > Vertex Lat.' @ PAUSE @ GOTO 2404
2416 G=G2 @ A=L @ L=L2 @ L2=A @ A=L1 @ GOSUB 2450 @ G2=G1-A @ A$='1.Max.' @ GOSUB 'DSP2'
2420 DISP @ PAUSE @ A=L @ GOSUB 2450 @ G2=G+A @ A$='2.Max.' @ GOSUB 'DSP2' @ DISP @ STOP
2450 A=ABS(ACOS(TAN(A)/TAN(L2)))*SGN(SIN(Z1)) @ RETURN

============================================================
4000 'DSPL': DISP A$;'Lat. = '; @ 'DPL': S=83*(A<0)+78*(A>=0)
4004 IF FLAG(0) THEN DISP USING 4008;FNH,FNM,FNT,S ELSE DISP USING 4012;FNH,FNM,FNS,S
4008 IMAGE #,ZZ,'',ZZ,"'",Z,X,B
4012 IMAGE #,ZZ,'',ZZ,"'",ZZ,'"',X,B
4016 RETURN

============================================================
```

```
4020 'DSPG': DISP A$;'Long. = '; @ S=69*(A<0)+87*(A>=0)

================================================================
4024 'DPG': IF FLAG(0) THEN DISP USING 4028;FNH,FNM,FNT,S ELSE DISP USING 4032;FNH,FNM,FNS,S
4028 IMAGE #,ZZZ,'',ZZ,"'",Z,X,B
4032 IMAGE #,ZZZ,'',ZZ,"'",ZZ,'"',X,B
4036 RETURN

================================================================
4040 'DSPDA': DISP 'Date(dmy) = '; @ DISP USING 4044;D,M,Y @ RETURN
4044 IMAGE #,ZZ,'/',ZZ,'/',4Z

================================================================
4048 'DSPTI': DISP 'Time = '; @ DISP USING 4052;FNH,FNM,FNS @ RETURN
4052 IMAGE #,2(ZZ,':'),ZZ

================================================================
4056 'DC': DISP 'Course = '; @ 'DPC': DISP USING 4060;FNH,IP(FNC/10) @ RETURN
4060 IMAGE #,3Z,'',Z

================================================================
4064 'DPD': DISP USING 4068;FNH,FNC @ RETURN
4068 IMAGE #,'Distance = ',K,"'",ZZ

================================================================
4072 'DSPMT': FIX 1 @ DISP A;'m'; @ STD @ RETURN

================================================================
4200 'DSP2': A=L2 @ GOSUB 'DSPL' @ A=G2 @ DISP TAB(25); @ GOSUB 'DSPG' @ RETURN

================================================================
4204 'SP': DISP @ PAUSE @ L1=L2 @ G1=G2 @ RETURN

================================================================
4208 'DSPDC': A=D1 @ GOSUB 'DPD' @ A=Z @ DISP TAB(25); @ GOSUB 'DC' @ DISP '    '; @ RETURN

================================================================
4212 'DSP1': A=L1 @ GOSUB 'DSPL' @ A=G1 @ GOSUB 'DSPG' @ RETURN

================================================================
4216 'DSPCO': S=32 @ DISP B$;
4220 IF FLAG(10) THEN DISP ' sha = '; @ A=MOD(A0-G0,360) @ GOSUB 'DPG' @ DISP TAB(25+LEN(B$));
4224 DISP ' gha = '; @ A=A0 @ GOSUB 'DPG'
4228 DISP TAB(25+LEN(B$)+25*FLAG(10,0));'dec. = '; @ A=D0 @ GOSUB 'DPL' @ DISP @ RETURN

================================================================
4400 'IL': GOSUB 'DSPL' @ GOSUB 'FNI' @ A=RED(FNR,180) @ RETURN

================================================================
4404 'IG': GOSUB 'DSPG' @ GOSUB 'FNI' @ A=RED(FNR,360) @ RETURN

================================================================
4408 'IN1': A=L1 @ GOSUB 'IL' @ L1=A @ A=G1 @ GOSUB 'IG' @ G1=A @ RETURN

================================================================
4412 'INP2': A$='Dep.' @ GOSUB 'IN1' @ A$='Arr.' @ A=L2 @ GOSUB 'IL' @ L2=A @ A=G2
4416 GOSUB 'IG' @ G2=A @ RETURN

================================================================
4420 'INPDT': GOSUB 'DSPDA' @ ON ERROR GOTO 4424 @ INPUT ' ','?';D,M,Y @ GOTO 4428
```

```
4424 IF ERRN=40 THEN Y=10000*FP(100*D) @ M=IP(100*FP(D)) @ D=IP(D)

============================================================
4428 'INPTI': A=T @ GOSUB 'DSPTI' @ GOSUB 'FNI' @ T=FNR2 @ RETURN

============================================================
4432 'INPC': A=Z @ GOSUB 'DC' @ GOSUB 'FNI' @ Z=A @ RETURN

============================================================
4436 'INPD': A=D1 @ GOSUB 'DPD' @ GOSUB 'FNI' @ D1=A @ RETURN

============================================================
4440 'INPX': INPUT 'Sextant height = ','?';A @ H1=FNR @ RETURN

============================================================
4444 'INPMT': GOSUB 'DSPMT' @ GOSUB 'FNI' @ RETURN

============================================================
4448 'FNI': ON ERROR GOTO 4460 @ INPUT ' ','?';B$ @ A=VAL(B$)
4452 IF POS('SE',UPRC$(B$[LEN(B$)])) THEN A=-A
4456 OFF ERROR @ RETURN
4460 POP @ OFF ERROR @ RETURN

============================================================
4600 'JD': T1=367*Y-INT(INT(INT((M+9)/12)+Y)*7/4)-INT(INT(INT(INT((M-9)/7+Y)/100)+1)*3/4)
4604 T1=T1+INT(275*M/9)+D-730516.5+T/24 @ T2=T1/36525+1 @ RETURN

============================================================
4608 'REF': H=H1+.97*(TAN(H1-ATAN((H1+3)*12))-SQR(E0/.3048))/60 @ RETURN

============================================================
4612 'MA2': A=L2 @ GOSUB 4616 @ B=A @ A=L1 @ GOSUB 4616 @ RETURN
4616 A=COS(A)/(1+SIN(A))/((1-E9*SIN(A))/(1+E9*SIN(A)))^(E9/2) @ RETURN

============================================================
4800 'ALM': GOSUB 'JD' @ D8=Y @ G0=M @ D0=D @ A0=H @ B$=''
4804 ON ERROR GOSUB 4816 @ INPUT 'Which body : ','?';A$,B$,A
4808 IF '?'=A$ THEN 'FIND'
4812 CALL B$(T,T1,T2,A$,B$,D8,A0,D0,G0) @ RETURN
4816 B$=UPRC$(B$[1,1]) @ IF ''=B$ THEN B$='P'
4820 ON ERROR GOSUB 4832 @ CALL B$
4824 B$=CHR$(MOD(NUM(B$)-58,22)+69) @ CALL B$
4828 B$='I' @ RETURN
4832 IF ERRN=36 THEN POP
4836 RETURN

============================================================
4840 'FIND': IF FLAG(11) THEN 'ALM'
4841 INPUT 'Star azimut = ','?';Z @ CALL SRT(H,L1,Z,A,B) @ IF 'P'=B$ THEN 4856
4844 SFLAG 10
4848 CALL B$(T,T1,T2,'ARIES',A$,D8,A0,D0,G0) @ D0=A @ A0=B+G1 @ B$='Star' @ GOSUB 'DSPCO'
4852 PAUSE @ GOTO 'ALM'
4856 DIM C$[100] @ FOR J=7 TO 68 @ CALL P(T,T1,T2,STR$(J),B$,D8,A0,D0,G0)
4860 IF ABS(A0-G1-B)+ABS(D0-A)<20 THEN GOSUB 4868
4864 NEXT J @ DISP C$ @ PAUSE @ DESTROY C$ @ GOTO 'ALM'
4868 A$=B$ @ B$='P' @ GOSUB 4812 @ P=A0-G1 @ CALL SRT(D0,L1,P,H2,C) @ OFF ERROR
4872 C$=C$&B$&' '&STR$(IP(H+D8-H2))&' ~ '&STR$(IP(C-Z))&' / ' @ RETURN

============================================================
8000 DEF FNH=IP(ABS(A))
```

```
================================================================
8004 DEF FNM=IP(60*FP(ABS(A)+.000000001))

================================================================
8008 DEF FNS=60*FP(60*(ABS(A)+.000000001))

================================================================
8012 DEF FNC=IP(100*FP(ABS(A)))

================================================================
8016 DEF FNT=IP(10*FP(60*(ABS(A)+.000000001)))

================================================================
8020 DEF FNR=IP(A)+FP(A)/.6*FLAG(0)+(FP(100*A)/.6+IP((A-IP(A))*100))/60*NOT FLAG(0)

================================================================
8024 DEF FNR1=IP(A)+FP(A)/.6

================================================================
8028 DEF FNR2=IP(A)+(FP(100*A)/.6+IP((A-IP(A))*100))/60

================================================================
8032 DEF FNN(J,E)
8036 RESTORE M$ @ J=1*(J<1)+E*(J>E)+J*(J>=1 AND J<=E)
8040 FOR I=1 TO J @ READ A$ @ NEXT I @ DISP A$
8044 B$=KEY$ @ IF B$='' THEN 8044
8048 IF B$='#51' THEN J=J+1 @ GOTO 8036
8052 IF B$='#50' THEN J=J-1 @ GOTO 8036
8056 IF B$='#162' THEN J=1 @ GOTO 8036
8060 IF B$='#163' THEN J=E @ GOTO 8036
8064 IF B$#'#38' THEN 8044 ELSE FNN=J
8068 END DEF

================================================================
9500 SUB I(T,U,V,A$,B$,Y,H,D,M) @ A=(POS('ARIVENMARJUPSATMOOSUN',A$[1,3])+2)/3 @ B$=A$
9504 IF A<2 THEN 9536
9508 DISP B$;' gha <';STR$(IP(T)); @ INPUT ':00> = ','?';G @ DISP B$;' gha <';STR$(CEIL(T));
9512 INPUT ':00> = ','?';B @ G=MOD(FNH(G)+RED(FNH(B)-FNH(G),360)*FP(T),360)
9516 DISP B$;' dec. <';STR$(IP(T)); @ INPUT ':00> = ','?';D @ DISP B$;' dec. <';STR$(CEIL(T));
9520 INPUT ':00> = ','?';B @ D=FNH(D)+(FNH(B)-FNH(D))*FP(T)
9524 IF A-5>0 THEN INPUT '  : ','?';Y @ Y=(Y-2*Y*('U'=B$[LEN(B$)]))*(POS('UL',B$[LEN(B$)])>0)
9528 IF MOD(A,7)>0 THEN INPUT '  : ','?';P
9532 Y=P*COS(H)+Y @ H=G @ GOTO 9552
9536 DISP 'ARIES gha <';STR$(IP(T)); @ INPUT ':00> = ','?';M @ M=FNH(M)+15.042*FP(T) @ G=M
9540 IF A=1 THEN 9552
9544 DISP B$; @ INPUT ' sha = ','?';H @ H=MOD(M+FNH(H),360)
9548 DISP B$; @ INPUT ' dec. = ','?';D @ D=FNH(D) @ SFLAG 10 @ Y=0

================================================================
9552 DEF FNH(A)=IP(A)+FP(A)/.6

================================================================
9700 SUB SRT(D,L,P,H,Z) @ ON ERROR GOTO 9708
9704 H=ASIN(SIN(L)*SIN(D)+COS(L)*COS(D)*COS(P)) @ A=ATAN(SIN(P)/(TAN(D)*COS(L)-SIN(L)*COS(P)))
9708 Z=(180*((SIN(P)>=0)+1)-ABS(A))*(A>=0)+(180*(SIN(P)>=0)+ABS(A))*(A<0)
```

**********

```
EPIMENIS


   4 INTEGER Y,M @ INPUT 'Annee = ','?';Y @ INPUT 'Mois = ','?';M @ CALL CO('?',Y,M)

================================================================
  10 SUB E(T,U,V,A$,B$,Y,H,D,M) @ DIM F$[8],G(6),O(6) @ INTEGER I
  14 IF NOT POS('SUVEMAJUSA',UPRC$(A$[1,2])) THEN B$=A$ @ A$='ARIES' @ SFLAG 10
  18 ON ERROR GOSUB 50 @ F$=UPRC$(A$[1,3])&STR$(MOD(Y,100))&'M'&STR$(M) @ A=HTD(ADDR$(F$))
  22 Y=0 @ ASSIGN #1 TO F$ @ A=ACOS((T/24+D)/16.5-1) @ IF FLAG(10) THEN D=0 @ H=0 @ GOTO 42
  26 READ #1;B$,G(),C,O(),P @ IF UPRC$(A$[4])='L' THEN Y=C
  30 IF UPRC$(A$[4])='U' THEN Y=-C
  34 Y=P*COS(H)+Y @ A=ACOS((T/24+D)/16.5-1) @ H=0 @ D=0
  38 FOR I=0 TO 6 @ H=H+G(I)*COS(A*I) @ D=D+O(I)*COS(A*I) @ NEXT I @ H=MOD(H,360) @ GOTO 54
  42 READ #1;A$,G(0),G(1) @ M=MOD(G(0)+G(1)*COS(A),360) @ IF POS(B$,'ARI') THEN 46
  44 DISP B$; @ INPUT ' sha = ','?';H @ DISP B$; @ INPUT ' dec. = ','?';D
  46 D=FNH(D) @ H=MOD(M+FNH(H),360) @ GOTO 54
  50 CALL CO(A$,Y,M) @ RETURN

================================================================
  54 DEF FNH(A)=IP(A)+FP(A)/.6

================================================================
 100 SUB CO(A$,Y,M) @ DIM G(6),D(6) @ INTEGER I @ IF FLAG(10) THEN 108
 104 ON ERROR GOTO 104 @ INPUT 'English Name : ',A$;A$
 108 ASSIGN #2 TO A$[1,3]&STR$(MOD(Y,100))&'M'&STR$(MOD(M,24))
 112 INPUT 'gha Test = ','?';T @ DESTROY H @ ON ERROR GOSUB 148
 116 FOR I=0 TO 6 @ DISP ' a (';STR$(I); @ INPUT ') = ','?';G(I) @ H=H+G(I)*COS(180*I) @ NEXT I
 120 OFF ERROR @ IF ABS(MOD(H,360)-IP(T)-FP(T)/.6)>5/6000 THEN DISP 'New '; @ GOTO 112
 124 IF UPRC$(A$[1,3])='ARI' THEN 144
 128 INPUT '  = ','?';C @ DESTROY H @ INPUT 'dec. Test = ','?';T @ ON ERROR GOSUB 148
 132 FOR I=0 TO 6 @ DISP ' d (';STR$(I); @ INPUT ') = ','?';D(I) @ H=H+D(I)*COS(180*I) @ NEXT I
 136 OFF ERROR @ IF ABS(H-IP(T)-FP(T)/.6)>5/6000 THEN DISP 'New '; @ GOTO 128
 140 INPUT '  = ','?';P @ PRINT #2;A$,G(),C,D(),P @ END
 144 PRINT #2;A$,G(0),G(1) @ END
 148 I=7 @ RETURN



*****************************************************************************************************

PERPET



================================================================
  10 SUB P(T,T1,T2,A$,B$,D8,A9,D0,H9)
  25 DIM C$[25] @ C$=A$ @ ON ERROR GOTO 1480 @ A=VAL(A$)
  26 DISP 'Working for ';A$;' ...' @ OPTION ROUND NEAR

================================================================
  40 'ARI': A1=T2-T/24/36525 @ D8=0
  50 H9=MOD(MOD((A1*.000387083+36000.76893)*A1,360)+99.69098325+T*15.04106863,360)
  60 D0=0 @ IF C$[1,3]='ARI' OR A=69 THEN A9=H9 @ B$='ARIES' @ END
  80 M1=218.31624+13.1763964644*T1 @ M2=134.96292+13.0649929524*T1 @ M3=93.27276+13.229350272*T1
  90 M4=297.85032+12.1907491128*T1 @ M5=125.04348-.0529538076*T1
 100 S1=280.46592+.9856473516*T1 @ S2=.98560026*T1-2.47464
 110 L2=181.97928+1.60216872372*T1 @ G2=50.40828+1.6021302228*T1 @ F2=105.29928+1.6021440612*T1
 120 L4=355.44708+.5240711808*T1 @ G4=19.38816+.5240207772*T1 @ F4=305.88984+.524050074*T1
```

```
130 L5=32.25888+.0830912148*T1 @ G5=20.35116+.0830912148*T1
140 L6=47.9862+.0334597356*T1 @ G6=317.87532+.0334597356*T1
150 E9=21107/900-47/3600*T2+COS(M5)/400 @ D9=17/3600*SIN(M5)
170 ON ERROR GOTO 180 @ ON A GOTO 190,200,210,290,280,295,430,490,550,630
180 ON ERROR GOTO 1340 @ GOTO A$ @ OFF ERROR

============================================================
190 'SUNL': D8=1 @ GOSUB 230 @ B$='SUNL' @ GOTO 740

============================================================
200 'SUNU': D8=-1 @ GOSUB 230 @ B$='SUNU' @ GOTO 740

============================================================
210 'SUN': GOSUB 230 @ B$='SUN' @ GOTO 740
230 X=6910*SIN(S2)+72*SIN(2*S2)-17*T2*SIN(S2)-7*COS(S2-G5)+6*SIN(M1-S1)+5*SIN(4*S2-8*G4+3*G5)
240 X=X-5*COS(2*S2-2*G2)-4*SIN(S2-G2)+4*COS(4*S2-8*G4+3*G5)+3*SIN(2*S2-2*G2)-3*SIN(G5)
250 X=(X-3*SIN(2*S2-2*G5))/3600+S1-D9 @ Y=0
260 Z=1.00014-.01675*COS(S2)-.00014*COS(2*S2) @ D8=ATAN(.00466/Z)*D8
270 CALL LBRYZX(Z,Y,X,E9) @ X5=X @ Y5=Y @ Z5=Z @ CALL XYZHAD(X,Z,Y) @ RETURN

============================================================
280 'MOONU': D8=-1 @ B$='MOONU' @ GOTO 310

============================================================
290 'MOONL': D8=1 @ B$='MOONL' @ GOTO 310

============================================================
295 'MOON': B$='MOON'
310 X=22640*SIN(M2)-4586*SIN(M2-2*M4)+2370*SIN(2*M4)+769*SIN(2*M2)-668*SIN(S2)-412*SIN(2*M3)
320 X=X-212*SIN(2*M2-2*M4)-206*SIN(M2-2*M4+S2)+192*SIN(M2+2*M4)+165*SIN(2*M4-S2)+148*SIN(M2-S2)
330 X=X-125*SIN(M4)-110*SIN(M2+S2)-55*SIN(2*M3-2*M4)-45*SIN(M2+2*M3)+40*SIN(M2-2*M3)
340 X=X-38*SIN(M2-4*M4)+36*SIN(3*M2)-31*SIN(2*M2-4*M4)+28*SIN(M2-2*M4-S2)-24*SIN(2*M4+S2)
350 X=(X+19*SIN(M2-M4)+18*SIN(M4+S2))/3600+MOD(M1,360)-D9
360 Y=18461*SIN(M3)+1010*SIN(M2+M3)+1000*SIN(M2-M3)-624*SIN(M3-2*M4)-199*SIN(M2-M3-2*M4)
370 Y=Y-167*SIN(M2+M3-2*M4)+117*SIN(M3+2*M4)+62*SIN(2*M2+M3)+33*SIN(M2-M3+2*M4)+32*SIN(2*M2-M3)
380 Y=(Y-30*SIN(M3-2*M4+S2)-16*SIN(2*M2+M3-2*M4)+15*SIN(M2+M3+2*M4))/3600
390 Z=60.36298-3.27746*COS(M2)-.57994*COS(M2-2*M4)-.46357*COS(2*M4)-.08904*COS(2*M2)
400 A$=STR$(ATAN(1/Z)) @ D8=.2725*D8*ATAN(1/Z) @ Z=Z/23454.8
410 CALL LBRYZX(Z,Y,X,E9) @ CALL XYZHAD(X,Z,Y) @ GOTO 740

============================================================
430 'VEN': GOSUB 230 @ B$='VENUS'
440 X=2814*SIN(G2)-181*SIN(2*F2)-20*T2*SIN(G2)+12*SIN(2*G2)-10*COS(2*S2-2*G2)+7*COS(3*S2-3*G2)
450 X=X/3600+MOD(L2,360)-D9 @ Y=(12215*SIN(F2)+83*SIN(G2+F2)+83*SIN(G2-F2))/3600
460 Z=.72335-.00493*COS(G2)
470 GOSUB 730 @ X=X+.0084 @ GOTO 740

============================================================
490 'MARS': GOSUB 230 @ B$='MARS'
500 X=38451*SIN(G4)+2238*SIN(2*G4)+181*SIN(3*G4)-52*SIN(2*F4)+37*T2*SIN(G4)-22*COS(G4-2*G5)
510 X=X/3600+L4-D9 @ Y=(6603*SIN(F4)+622*SIN(G4-F4)+615*SIN(G4+F4)+64*SIN(2*G4+F4))/3600
520 Z=1.53031-.1417*COS(G4)-.0066*COS(2*G4)-.00047*COS(3*G4)
530 GOSUB 730 @ X=X-.0093 @ Z=Z-.0021 @ GOTO 740

============================================================
550 'JUP': GOSUB 230 @ B$='JUPITER'
560 X=19934*SIN(G5)+5023*T2+2511+1093*COS(2*G5-5*G6)+601*SIN(2*G5)-479*SIN(2*G5-5*G6)
570 X=X-185*SIN(2*G5-2*G6)+137*SIN(3*G5-5*G6)-131*SIN(G5-2*G6)+79*COS(G5-G6)-76*COS(2*G5-2*G6)
580 X=(X-74*T2*COS(G5)+68*T2*SIN(G5)+66*COS(2*G5-3*G6)+63*COS(3*G5-5*G6))/3600+MOD(L5,360)-D9
590 Y=(-4692*COS(G5)+259*SIN(G5)+227-227*COS(2*G5))/3600
```

```
600 Z=5.20883-.25122*COS(G5)-.00604*COS(2*G5)
610 GOSUB 730 @ X=X-.0229 @ Z=Z-.0022 @ GOTO 740

============================================================
630 'SAT': GOSUB 230 @ B$='SATURN'
640 X=23045*SIN(G6)+5014*T2-2689*COS(2*G5-5*G6)+2507+1177*SIN(2*G5-5*G6)-826*COS(2*G5-4*G6)
650 X=X+802*SIN(2*G6)+425*SIN(G5-2*G6)-229*T2*COS(G6)-153*COS(2*G5-6*G6)-142*T2*SIN(G6)
660 X=X-114*COS(G6)+101*T2*SIN(2*G5-5*G6)-70*COS(2*L6)+67*SIN(2*L6)+66*SIN(2*G5-6*G6)
670 X=X+60*T2*COS(2*G5-5*G6)+41*SIN(G5-3*G6)+39*SIN(3*G6)+31*SIN(G5-G6)+31*SIN(2*G5-2*G6)
680 X=(X-29*COS(2*G5-3*G6))/3600+MOD(L6,360)-D9
690 Y=8297*SIN(G6)-3346*COS(G6)+462*SIN(2*G6)-189*COS(2*G6)+185+79*T2*COS(G6)-71*COS(2*G5-4*G6)
700 Y=(Y+46*SIN(2*G5-6*G6)-45*COS(2*G5-6*G6)+29*SIN(3*G6))/3600
710 Z=9.55774-.53252*COS(G6)-.01878*SIN(-4*G6)-.01482*COS(2*G6)+.00817*SIN(G5-G6)
720 GOSUB 730 @ X=X-.0055 @ Z=Z-.0064 @ GOTO 740
730 CALL LBRYZX(Z,Y,X,E9) @ X=X+X5 @ Y=Y+Y5 @ Z=Z+Z5 @ CALL XYZHAD(X,Z,Y) @ RETURN
740 A9=MOD(X+H9,360) @ D0=Z @ END

============================================================
760 'SIR': DATA SIRIUS,-16.6722-.0361*(T2-.69),259.0542+.0158*(T2-.69)

============================================================
770 'CAN': DATA CANOPUS,-52.6783,264.1854

============================================================
780 'VEG': DATA VEGA,38.7539,81.0279

============================================================
790 'RIGI': DATA RIGIL KENTAURUS,-60.7086+.0214*(T2-.69),140.63+.2028*(T2-.69)

============================================================
800 'BET': DATA BETELGEUSE,7.4028,271.6267

============================================================
810 'CAP': DATA CAPELLA,45.9683,281.4008

============================================================
820 'ARC': DATA ARCTURUS,19.3431-.0558*(T2-.69),146.4383+.0372*(T2-.69)

============================================================
830 'RIGE': DATA RIGEL,-8.2361,281.7383

============================================================
840 'PRO': DATA PROCYON,5.3056-.0286*(T2-.69),245.58+.0186*(T2-.69)

============================================================
850 'ACH': DATA ACHERNAR,-57.3939,335.8596

============================================================
860 'HAD': DATA HADAR,-60.2244,149.595

============================================================
870 'ALT': DATA ALTAIR,8.7853+.01*(T2-.69),62.6825-.0158*(T2-.69)

============================================================
880 'ACR': DATA ACRUX,-62.9275,173.785

============================================================
890 'ALD': DATA ALDEBARAN,16.4483,291.465

============================================================
```

```
900 'SPI': DATA SPICA,-11,159.1108

================================================================
910 'POLL': DATA POLLUX,28.1025,244.1446+.0186*(T2-.69)

================================================================
920 'ANT': DATA ANTARES,-26.3647,113.1242

================================================================
930 'REG': DATA REGULUS,12.1194,208.3192

================================================================
940 'FOM': DATA FOMALHAUT,-29.7869,16.015

================================================================
950 'DENEB': DATA DENEB,45.1689,49.9067

================================================================
960 'ADH': DATA ADHARA,-28.9289,255.6483

================================================================
970 'GAC': DATA GACRUX,-56.94,172.6413

================================================================
980 'BEL': DATA BELLATRIX,6.3231,279.1333

================================================================
990 'SHA': DATA SHAULA,-37.0831,97.1246

================================================================
1000 'ALI': DATA ALIOTH,56.1278,166.8325

================================================================
1010 'AVI': DATA AVIOR,-59.4094,234.53

================================================================
1020 'ALNI': DATA ALNILAM,-1.2203,276.34

================================================================
1030 'ELN': DATA ELNATH,28.5828,278.9175

================================================================
1040 'MIA': DATA MIAPLACIDUS,-69.5894,221.7829

================================================================
1050 'ALK': DATA ALKAID,49.4675,153.42

================================================================
1060 'MIR': DATA MIRFAK,49.7522,309.475

================================================================
1070 'ATR': DATA ATRIA,-68.9733,108.6588

================================================================
1080 'KAU': DATA KAUS AUSTRALIS,-34.4011,84.4717

================================================================
1090 'DUB': DATA DUBHE,61.9186,194.5429

================================================================
```

```
1100 'POLA': DATA POLARIS,89.1214,329.3708

============================================================
1110 'PEA': DATA PEACOCK,-53.8361,54.1988

============================================================
1120 'NUN': DATA NUNKI,-26.3369,76.6642

============================================================
1130 'RAS': DATA RASALHAGUE,12.5811,96.6267

============================================================
1140 'ALPHER': DATA ALPHERATZ,28.9194,358.3050

============================================================
1150 'ALNA': DATA ALNAIR,-47.1119,28.4283

============================================================
1160 'DENEBO': DATA DENEBOLA,14.7453,183.13+.0158*(T2-.69)

============================================================
1170 'DIP': DATA DIPHDA,-18.1564,349.4917

============================================================
1180 'HAM': DATA HAMAL,23.3167,328.645

============================================================
1190 'KOC': DATA KOCHAB,74.2822,137.3058

============================================================
1200 'ALPHA': DATA ALPHARD,-8.5233,218.4842

============================================================
1210 'SUH': DATA SUHAIL,-43.3069,223.2863

============================================================
1220 'ALPHEC': DATA ALPHECCA,26.8181,126.6527

============================================================
1230 'MENKE': DATA MENKENT,-36.2183-.0114*(T2-.69),148.7875+.0172*(T2-.69)

============================================================
1240 'ANK': DATA ANKAA,-42.4744,353.8113

============================================================
1250 'ELT': DATA ELTANIN,51.4919,91.0287

============================================================
1260 'SCH': DATA SHEDAR,56.3678,350.3163

============================================================
1270 'ENI': DATA ENIF,9.7322,34.3346

============================================================
1280 'MAR': DATA MARKAB,15.0381,14.1967

============================================================
1290 'SAB': DATA SABIK,-15.6881,102.8508

============================================================
```

```
1300 'GIE': DATA GIENAH,-17.37,176.4479

============================================================
1310 'MENKA': DATA MENKAR,3.9692,314.8358

============================================================
1320 'ZUB': DATA ZUBEN-UBI,-15.9142,137.71

============================================================
1330 'ACA': DATA ACAMAR,-40.4281,314.8358
1340 ON ERROR GOTO 1360 @ A=A-10
1350 ON A RESTORE 760,770,780,790,800,810,820,830,840,850,860,870,880,890,900
1355 GOTO 1435
1360 ON ERROR GOTO 1380 @ A=A-15
1370 ON A RESTORE 910,920,930,940,950,960,970,980,990,1000,1010,1020,1030,1040,1050
1375 GOTO 1435
1380 ON ERROR GOTO 1400 @ A=A-15
1390 ON A RESTORE 1060,1070,1080,1090,1100,1110,1120,1130,1140,1150,1160,1170,1180,1190,1200
1395 GOTO 1435
1400 ON ERROR GOTO 1420 @ A=A-15
1410 ON A RESTORE 1210,1220,1230,1240,1250,1260,1270,1280,1290,1300,1310,1320,1330
1415 GOTO 1435
1420 ON ERROR GOTO 1480
1430 RESTORE C$ @ OFF ERROR
1435 READ B$,D,H @ SFLAG 10
1440 F=(H9-15*T)/.985647-T/24
1450 A9=H+(T2-.69)*(SIN(H)*TAN(D)-2.3)/1.8-SIN(M5-H)*TAN(D)/426+SIN(M5)/208.77
1460 A9=MOD(A9+H9+COS(H-F)/COS(D)/171.4,360)
1470 D0=D+(T2-.69)*COS(H)/1.8-COS(M5-H)/426+SIN(D)*SIN(H-F)/171.4 @ END
1480 IF '?'=C$ THEN DISP 'Search aborted' @ DESTROY A9,D0 @ B$='...' @ END
1490 IF LEN(C$)<=2 THEN DISP 'No ';A$; @ INPUT ' / Correct name ','?';A$ @ GOTO 25
1510 C$=C$[1,LEN(C$)-1] @ ON ERROR GOTO 1420 @ GOTO C$

============================================================
2000 SUB XYZHAD(X,Y,Z)
2010 F=SQR(X^2+Y^2) @ G=ANGLE(X,Y) @ H=SQR(F^2+Z^2) @ Y=ANGLE(F,Z) @ X=360-G @ Z=0

============================================================
2030 SUB LBRYZX(X,Y,Z,T) @ A=X*COS(Y)*COS(Z) @ B=X*(COS(Y)*SIN(T)*SIN(Z)+SIN(Y)*COS(T))
2040 C=X*(COS(Y)*COS(T)*SIN(Z)-SIN(Y)*SIN(T)) @ Z=A @ Y=B @ X=C



**********************************************************************************************************

LES TABLES DE LA LOIS (MATHROM)


   - Tables de STUDENT,KI2,FISHER-SNEDECOR

============================================================
  10 SUB TPROBA(N,X,Q) ! unilatéral, loi de STUDENT: n,x-->q
  12 IF X<0 THEN BEEP 999 @ DISP 'ERR: X < 0 !!' @ PAUSE @ GOTO 65
  15 RADIANS @ Y=X/SQR(N) @ T=ATAN(Y)
  20 IF N=1 THEN B=T*2/PI @ GOTO 60
  25 S=COS(T)^2 @ M=2*INT(N/2) @ IF M=N THEN 45
  30 B=COS(T) @ S=B*B @ V=B @ IF N=3 THEN 40
  35 FOR I=2 TO N-3 STEP 2 @ V=V*S*I/(I+1) @ B=B+V @ NEXT I
  40 B=B*SIN(T) @ B=(B+T)*2/PI @ GOTO 60
  45 S=COS(T)^2 @ B=1 @ V=1 @ IF N=2 THEN 55
```

```
 50 FOR I=1 TO N-3 STEP 2 @ V=V*S*I/(I+1) @ B=B+V @ NEXT I
 55 B=B*SIN(T)
 60 Q=1-((1+B)/2)
 65 END SUB
```

```
================================================================
```

- Loi du KI2: n,x-->q ;le flag(60) sert à enlever le bug de L30

```
================================================================
205 SUB KI2PROBA(N,X,Q) ! valable ssi Q>.5
207 IF X<0 THEN BEEP 999 @ DISP 'ERR: X < 0 !!' @ PAUSE @ GOTO 260
210 O=N/2 @ G=1 @ M=INT(O)
215 IF N=M*2 THEN GOTO 235
220 IF N=1 THEN GOTO 230
225 FOR I=1 TO 2*M-1 STEP 2 @ G=G*I/2 @ NEXT I
230 G=G*SQR(PI) @ GOTO 245
235 IF N=2 THEN GOTO 245
240 FOR I=1 TO M-1 @ G=G*I @ NEXT I
245 A=EXP(-X/2) @ F=(X/2)^O @ F=F*A/(O*G) @ E=1 @ P=1 @ I=2
250 E=E*X/(N+I) @ P=P+E @ IF E*F>.00000001 THEN I=I+2 @ GOTO 250
255 Q=1-(P*F) @ IF Q>=.999 AND NOT FLAG(60) THEN Q=.000000001
260 END SUB
```

```
================================================================
```

- Loi de FISHER-SNEDECOR: n,m,f-->p

```
================================================================
405 SUB FPROBA(N,M,F,P) @ IF N=1 THEN CALL TPROBA(M,SQR(F),P) @ P=2*P @ GOTO 430 ELSE E=.0001
410 DELAY 0,0 @ DISP 'Proba...' @ DEF FNG(N,M,T)=T^(N/2-1)*(M+N*T)^(-(N+M)/2)
415 A=GAMMA((N+M)/2)*N^(N/2)*M^(M/2)/GAMMA(N/2)/GAMMA(M/2)
420 B=INTEGRAL(O,F,E,FNG(N,M,IVAR))
425 P=A*B @ P=1-P
430 END SUB
```

```
================================================================
```

- table de la loi de STUDENT: n,q-->x

```
================================================================
505 SUB PROBAT(N,Q,X) ! 0<Q<.5
507 IF Q>.5 OR Q<0 THEN DISP 'ERR: pas bon Q' @ BEEP 1000 @ PAUSE @ GOTO 525
```

```
================================================================
510 DEF FNT(Q) @ FNT=FNROOT(0,1,FNT1(FVAR)-Q) @ END DEF
```

```
================================================================
515 DEF FNT1(Y) @ CALL TPROBA((N),Y,P) @ FNT1=P @ END DEF
```

```
================================================================
520 A=FNT(Q) @ X=N
525 END SUB
```

```
================================================================
```

- table de la loi du KI2: n,q-->y ;le flag(10) est pour sortir de Newton

```
================================================================
605 SUB PROBAKI2(N,Q,Y) @ SFLAG 60 @ SFLAG 10 @ E=.01
607 IF Q>=1 OR Q<=0 THEN DISP 'ERR: pas bon Q' @ BEEP 1000 @ PAUSE @ GOTO 705
610 IF N<12 AND Q>.2 OR N<5 THEN E=.00001
615 IF N<5 THEN S=0 @ T=1 @ GOTO 660
620 IF .075>Q AND Q>=.035 THEN S=4.1+1.3*N @ T=4.5+1.4*N @ GOTO 660
```

```
625 IF .035>Q AND Q>=.015 THEN S=6+1.3*N @ T=7+1.5*N @ GOTO 660
630 IF .015>Q AND Q>=.008 THEN S=7+1.4*N @ T=8+1.6*N @ GOTO 660
635 IF .15>Q AND Q>=.075 THEN S=2.5+1.2*N @ T=3+1.3*N @ GOTO 660
640 IF .008>Q AND Q>=.0008 THEN S=11+1.5*N @ T=14+1.6*N @ GOTO 660
645 IF .4>Q AND Q>=.15 THEN S=1.6+.9*N @ T=2+1.2*N @ GOTO 660
650 IF .7>Q AND Q>=.4 THEN S=-.6+N @ T=-1.6+.9*N @ GOTO 660
655 IF 1>Q AND Q>=.7 THEN S=-2.5+.5*N @ T=-.63+N
660 W=FNK(N,Q) @ CFLAG 60 ! le W est pour vous faire parler

================================================================
665 DEF FNK(N,Q)
670 IF FLAG(10) THEN FNK=FNROOT(S,T,FNK1(FVAR,N)-Q) ELSE FNK=FVAR
675 END DEF

================================================================
680 DEF FNK1(X,N)
685 IF NOT FLAG(10) THEN FNK1=FVAR @ GOTO 700
690 IF ABS(Y-FVAR)<E THEN Y=FVAR @ CFLAG 10 @ GOTO 700 ELSE Y=FVAR
695 CALL KI2PROBA((N),(X),P) @ FNK1=P
700 END DEF

================================================================
705 END SUB

================================================================
   - table de FISHER-SNEDECOR@ n,m,p--->y

================================================================
905 SUB PROBAF(N,M,P,Y)
906 IF P>=1 OR P<=0 THEN BEEP 999 @ DISP 'ERR: pas bon P' @ PAUSE @ GOTO 950
908 SFLAG 10 @ A=FNF2(N,M,P)

================================================================
910 DEF FNF2(N,M,P)
915 IF FLAG(10) THEN FNF2=FNROOT(1,2,FNF1(FVAR,N,M)-P) ELSE FNF2=FVAR
920 END DEF

================================================================
925 DEF FNF1(X,N,M)
930 IF NOT FLAG(10) THEN FNF1=FVAR @ GOTO 945
935 IF ABS(Y-FVAR)<.01 THEN Y=FVAR @ CFLAG 10 @ GOTO 945 ELSE Y=FVAR
940 CALL FPROBA((N),(M),X,Q) @ FNF1=Q
945 END DEF

================================================================
950 END SUB



*****************************************************************************************************

FORMAT HPAF (MATHROM + KEYWAIT)



================================================================
   5 SUB AIDEHPAF
   - pgm d'aide au formattage HPAF
     sauvegarde @ chargement au format HPAF
     A est le tableau, apriori au DIM 15x12
```

```
    N$() est le nom des # colonnes (optionnel)(15 par ex.)
    si N$(1)="" les colonnes sont inommables
    A et les noms N$ sont sauvegardés par le pgm SAVEHPAF dans un fichier-data F$
    Si R,C=0,0 LOADHPAF formatte A !!
    Si R,C#0,0 LOADHPAF complète la matrice,déja existante..
    ..par le contenu du fichier-data F$
    LEX KEYWAIT, sub-pgm OUINON (JPC n.27 ou 28), module MATH

    construction du tableau
100 DESTROY ALL @ STD @ OPTION BASE 1
110 INPUT "Nbres de lign, col.? ";R,C @ DIM A(R,C),N$(C) @ MAT INPUT A
120 FOR I=1 TO C @ DISP "Nom";I;"ième colonne"; @ INPUT N$(I) @ IF N$(I)='' THEN 140
125 NEXT I
  - sauvegarde
140 INPUT "Nom du fich.sauveur? ";F$
150 CALL SAVEHPAF(A(,),N$(),F$) ! c'est sauvé
  - chargement en tableau d'un fichier hpaf
500 DESTROY ALL @ INTEGER R,C,R9,C9
510 INPUT "DIM  apriori du tabl?";R9,C9 @ DIM A(1,R9*C9),N$(R9*C9)
520 INPUT "Nom du fich.chargeur ";F$
530 CALL LOADHPAF(R,C,A(,),N$(),F$)
540 DIM A(R,C),N$(C) ! et voilà: on a tout: R,C,A(rxc),N$()
  - DIM A(R,C),N$(r) ! si les lignes sont nommées
    vérification:
700 DELAY 8 @ DISP "R,C= ";R;C @ MAT DISP A; @ IF NOT LEN(N$(1)) THEN 720
710 FOR I=1 TO C @ DISP N$(I) @ NEXT I ! ou i=1 à R
720 DELAY 0,0
800 END SUB

================================================================
  - copie write de HEWLETT-PACARD par Michel WEIL
    flag 22 armé:les lignes du tableau A ont un nom N$ (sub-SAVE)
    flag 22 desarmé: pas de nom pour ces lignes (sub-SAVE)
    flag 21 armé: il n'y a pas de tabl.préexistant à l'enregistrement des données
    flag 21 desarmé: on veut complèter un tableau par des données d'un fichier-data.
    LEX KEYWAIT et ss-pgm OUINON utilisés (JPC nº 27 ou 28)
    module MATH utile (la version sans module est à votre disposition)
    OPTION BASE 1
    A: RxC-matrice ; N$: C-vecteur ;F$: fich.-data

================================================================
1890 SUB SAVEHPAF(A(,),N$(),F$) @ R=UBND(A,1) @ C=UBND(A,2)
1910 IF LEN(N$(1)) THEN B=R+3 @ D=C
1920 F$=UPRC$(F$) @ I=LEN(F$) @ J=POS(F$,":")
1930 IF J=1 OR NOT I OR POS(F$," ") THEN DISP "Fich. non valable" @ GOTO 2100
1940 DISP "Sauvegarde..." @ ON ERROR GOTO 1960
1950 CREATE DATA F$,3+R+D,C*8 @ ASSIGN #9 TO F$ @ OFF ERROR @ GOTO 2020
1960 OFF ERROR @ IF ERRN#59 AND ERRN#255030 THEN 2010
1970 Q$="écrase fich.-dat(O/N)?" @ CALL OUINON(Q$,Q) @ IF NOT Q THEN 2100
1980 DISP "Sauvegarde.." @ PURGE F$
1990 ON ERROR GOTO 2010
2000 GOTO 1950
2010 OFF ERROR @ DISP ERRM$ @ GOTO 2100
2020 IF MEM<300 AND NOT J THEN PURGE F$ @ DISP "manque de Mémoire!" @ GOTO 2100
2030 Q$="HPAF" @ FOR I=1 TO C @ Q$=Q$&"N" @ NEXT I
2040 PRINT #9,0;Q$
2050 PRINT #9,1;R @ PRINT #9,2;B ! B: bloc-descriptif (noms des colonnes)
2060 FOR I=1 TO R @ RESTORE #9,I+2
2070 FOR J=1 TO C @ PRINT #9;A(I,J) @ NEXT J @ NEXT I
2080 IF NOT B THEN 2100
```

```
2090 FOR I=1 TO C @ PRINT #9,B+I-1;N$(I) @ NEXT I
2100 ASSIGN #9 TO * @ END SUB

============================================================
2130 SUB LOADHPAF(R,C,A(,),N$(),F$) @ CFLAG 21 @ ASSIGN #9 TO F$
2140 LINPUT "DIM.tabl.prédonné?","0,0";R$ @ W=POS(R$,',') @ INTEGER T(20)
2150 R=VAL(R$[1,W-1]) @ C=VAL(R$[W+1])
2160 IF NOT R AND NOT C THEN SFLAG 21 @ GOTO 2190
2170 W$="Suppress.données(O/N)?" @ CALL OUINON(W$,W)
2180 IF W THEN MAT A=ZERO @ R=0 @ C=0 @ SFLAG 21
2190 J=POS(F$,":") @ IF J=1 OR POS(F$," ") THEN DISP "Fichier inadapté" @ GOTO 'FIN'
2200 DISP "ça charge.."
2210 ON ERROR GOTO 2220 @ READ #9,0;T$ @ GOTO 2230
2220 OFF ERROR @ DISP "Fich.non formatté HPAF" @ GOTO 'FIN'
2230 IF T$[1,4]#"HPAF" THEN 2220
2240 T$=T$[5] @ C2=LEN(T$)
2250 C3=0 @ FOR I=1 TO C2 @ IF T$[I,I]='N' THEN T(I)=1 @ C3=C3+1 @ GOTO 2270
2260 IF T$[I,I]='S' THEN T(I)=0 ELSE 2220
2270 NEXT I @ IF NOT FLAG(21) AND C3#C THEN DISP "ERR: Colonnes #" @ GOTO 'FIN'
2280 READ #9,1;R1 @ R=R+R1 @ C=C3 @ IF MEM-R1*C*8<300 THEN 2300
2290 ON ERROR GOTO 2220 @ GOTO 2320
2300 OFF ERROR @ DISP "trop de données!!" @ PAUSE @ GOTO 'FIN'
2310 IF NOT R1 THEN MAT A=ZERO @ R=0 @ C=0 ELSE R=R-R1
2320 OFF ERROR @ READ #9,2;B
2330 X=3 @ FOR I=R-R1+1 TO R @ C1=1 @ READ #9,X @ FOR J=1 TO C2
2340 IF T(J) THEN READ #9;A(1,(I-1)*C+C1) @ C1=C1+1 ELSE IF NOT B THEN READ #9;N$(I)
2350 NEXT J @ X=X+1 @ NEXT I
2360 IF NOT B THEN 'FIN'
2370 ON ERROR GOTO 'FIN'
2380 FOR I=1 TO C3 @ READ #9,B+I-1;N$(I) @ NEXT I

============================================================
2390 'FIN': ASSIGN #9 TO * @ END SUB
   - fct réponse

============================================================
8010 SUB OUINON(Q$,Q)
8020 DISP Q$ @ I=POS("NOFC12",UPRC$(KEYWAIT$[1,1]))-1 @ IF I<0 THEN 8020
8030 Q=MOD(I,2) @ END SUB



**********************************************************************************************

UN CALENDRIER POUR LES LECTEURS (EDLEX + KEYWAIT)


   - Calendrier
  10 DESTROY ALL @ T5=TIME @ FOR I=1 TO 3 @ PUT '#48' @ NEXT I
  20 STD @ INPUT 'année?= ','1986';A
  30 CALL DFETES(A) ! (Calcul des fêtes de l'année.)
  40 ASSIGN #1 TO FETES @ READ #1,0;F$
  50 CALL LUNE(A) ! (Calcul des phases de la lunes.)
  60 ASSIGN #2 TO LUNES @ READ #2,0;W$ @ DISP 'Tracé du calendrier' @ BEEP
  70 FOR M=1 TO 12
  80 RESTORE 510 @ FOR I=1 TO M @ READ M$ @ NEXT I ! (Nombre de jours dans le mois)
  90 RESTORE 530 @ FOR J=1 TO I-1 @ READ P @ NEXT J
 100 J1=MOD(FP(A/100)*100+FP(A/100)*100 DIV 4+P+1,7)
 110 RESTORE 500 @ FOR I=0 TO J1 @ READ J1$ @ NEXT I ! (Nom du 1 Janvier)
 120 RESTORE 520 @ FOR I=1 TO M @ READ N @ NEXT I
```

```
130 IF N#0 THEN 150 ELSE IF NOT MOD(A,4) THEN N=29 ELSE N=28
140 IF NOT MOD(A,100) AND MOD(A,400) THEN N=28
150 M$=M$&'  '&STR$(A) @ ENDLINE CHR$(13)
160 RESTORE IO @ CONTROL ON @ PRINTER IS :%35 @ CLEAR @ PRINT CHR$(27)&"&k3s1W"
170 PRINT CHR$(27)&"&l8d1l64f96P" @ ENDLINE
180 PRINT TAB((30-LEN(M$)) DIV 2);M$
190 PRINT '==============================' @ ENDLINE CHR$(13) @ PRINT CHR$(27)&'&k2S'
200 RESTORE 500 @ FOR I=1 TO J1 @ READ J$ @ NEXT I
210 FOR I=1 TO N @ READ J$ @ IF J$='Samedi' THEN RESTORE 500
220 ON ERROR GOTO 230 @ READ #1,0;F$ @ IF VAL(F$[1,4])#100*M+I THEN DESTROY F$ @ GOTO 230
225 READ #1,1;F1$ @ IF VAL(F$[1,4])=VAL(F1$[1,4]) THEN F$=F$&','&F1$[5] ELSE DESTROY F1$
230 ON ERROR GOTO 240 @ READ #2,0;W$ @ IF VAL(W$[1,2])#I THEN DESTROY W$
240 OFF ERROR @ L=11-LEN(J$)-LEN(STR$(I))
250 IF J$='DIMANCHE' THEN PRINT CHR$(27)&"(s1B" ELSE PRINT CHR$(27)&"(s0B"
260 IF F$#'' THEN PRINT CHR$(27)&"(s1B" @ J$=UPRC$(J$)
265 ENDLINE CHR$(13)&CHR$(27)&'='
270 PRINT USING ",9A,2D,' |',29A,12A";J$,I,F$[5],W$[4] @ PRINT CHR$(27)&"(s0B"
280 IF F$='' THEN 290 ELSE DESTROY F$ @ DELETE #1,0
285 IF F1$='' THEN 290 ELSE DESTROY F1$ @ DELETE #1,0
290 IF W$='' THEN 300 ELSE DESTROY W$ @ DELETE #2,0
300 PRINT '------------------------------------------------------------'
310 NEXT I @ FOR I=I TO 31 @ PRINT ' ' @ PRINT ' ' @ NEXT I @ NEXT M
320 PURGE LUNES @ PURGE FETES @ T5=TIME-T5 @ DISP T5 @ PAUSE @ END ALL

================================================================
500 DATA DIMANCHE,Lundi,Mardi,Mercredi,Jeudi,Vendredi,Samedi
510 DATA Janvier,Février,Mars,Avril,Mai,Juin,Juillet,Août,Septembre,Octobre,Novembre,Décembre
520 DATA 31,0,31,30,31,30,31,31,30,31,30,31
530 DATA 0,3,3,6,1,4,6,2,5,0,3,5

================================================================
2000 SUB DFETES(A) @ BEEP @ DISP 'FETES' @ CREATE TEXT FETES @ ASSIGN #1 TO FETES @ ENDLINE
   - (Fêtes fixes)
2020 PRINT #1;"0101(Jour de l'an)"
2030 PRINT #1;"0501(Fête du travail)"
2040 PRINT #1;"0508(Victoire 1945)"
2050 PRINT #1;"0714(Fête nationale)"
2060 PRINT #1;"0815(Assomption)"
2070 PRINT #1;"1101(Toussaint)"
2080 PRINT #1;"1111(Victoire 1918)"
2090 PRINT #1;"1225(Noël)"
   - (Fêtes mobiles)
2100 CALL PAQUES(A,N1,P1)
2110 P$='0'&STR$(N1) @ IF P1<10 THEN P$=P$&'0'
2120 INSERT #1,1;P$&STR$(P1)&'(Pâques)'
2130 INSERT #1,2;P$&STR$(P1+1)
2140 Z$=DATE$ @ READ #1,1;P$ @ SETDATE '00/'&P$[1,2]&'/'&P$[3,4]
2150 G=DATE+1-7*7 @ P=DATE+49 @ A1=P-10
2160 SETDATE G @ INSERT #1,1;DATE$[4,5]&DATE$[7]&'(Mardi gras)'
2170 SETDATE A1 @ INSERT #1,6;DATE$[4,5]&DATE$[7]&'(Ascension)'
2180 SETDATE P @ INSERT #1,7;DATE$[4,5]&DATE$[7]&'(Pentecôte)'
2190 SETDATE DATE+1 @ INSERT #1,8;DATE$[4,5]&DATE$[7] @ SETDATE Z$ @ END SUB

================================================================
3000 SUB LUNE(A) @ CREATE TEXT LUNES @ ASSIGN #2 TO LUNES @ ENDLINE
3010 FOR M=1 TO 12 @ BEEP @ DISP 'Lunes mois n'&STR$(M) @ FOR J=3 TO 28 STEP 5 @ CFLAG 5,7
3020 CALL LUNES(A,M,J,W$) @ IF W2$=W$ THEN 3040
3030 PRINT #2;W$ @ W2$=W$
3040 NEXT J @ NEXT M @ END SUB
```

```
============================================================
4000 SUB PAQUES(A,N1,P1) ! (Calcul de la date de Pâques)
4010 R1=RMD(A,19) @ Q1=A DIV 100 @ R2=RMD(A,100) @ Q2=Q1 DIV 4 @ R3=RMD(Q1,4)
4020 Q3=(Q1+8) DIV 25 @ Q4=(Q1-Q3+1) DIV 3 @ R4=RMD(19*R1+Q1-Q2-Q4+15,30)
4030 Q5=R2 DIV 4 @ R5=RMD(R2,4) @ R6=RMD(32+2*R3+2*Q5-R4-R5,7)
4040 Q6=(R1+11*R4+22*R6) DIV 451
4050 A1=R4+R6-7*Q6+114 @ N1=A1 DIV 31 @ P1=RMD(A1,31)+1 @ END SUB

============================================================
5000 SUB LUNES(A,M,J,W$)
5010 A$=DATE$ @ D$=STR$(A)&'/' @ IF M<10 THEN D$=D$&'0'
5020 D$=D$&STR$(M)&'/' @ IF J<10 THEN D$=D$&'0'
5030 SETDATE D$&STR$(J)
5040 A2=A+RMD(DATE,1000)/360 @ SETDATE A$
5050 T=A2/100-19 @ K=FNR(1236.85*T,4) @ DEGREES
5060 K2=4*FP(K) @ IF K2=0 THEN SFLAG 7 @ GOTO 5100
5070 IF K2=1 THEN 5100
5080 IF K2=2 THEN SFLAG 5 @ GOTO 5100
5090 IF K2=3 THEN SFLAG 5,7
5100 J2=2415020.75933+29.53058868*K+.0001178*T^2-.000000155*T^3
5110 J2=J2+.00033*SIN(166.56+132.87*T-.009173*T^2)
5120 M1=RMD(359.2242+29.10535608*K-.0000333*T^2-.00000347*T^3,360)
5130 M2=RMD(306.0253+385.81691806*K+.0107306*T^2+.00001236*T^3,360)
5140 F=RMD(21.2964+390.67050646*K-.0016528*T^2-.00000239*T^3,360)
5150 IF MOD(FLAG(5)+FLAG(7),2) THEN 5230 ! (Premier et Dernier quartier)
5160 C=(.1721-.0004*T)*SIN(M1)+.0021*SIN(2*M1)-.628*SIN(M2)+.0089*SIN(2*M2)-.0004*SIN(3*M2)
5170 C=C+.0079*SIN(2*F)-.0119*SIN(M1+M2)-.0047*SIN(M1-M2)+.0003*SIN(2*F+M1)-.0004*SIN(2*F-M1)
5180 C=C-.0006*SIN(2*F+M2)+.0021*SIN(2*F-M2)+.0003*SIN(M1+2*M2)+.0004*SIN(M1-2*M2)
5190 C=C-.0003*SIN(2*M1+M2) @ C1=.0028-.0004*COS(M1)+.0003*COS(M2)
5200 IF FLAG(7) THEN 5220
5210 C=C+C1 @ L$='PQ' @ GOTO 9000 ! (Premier quartier)
5220 C=C-C1 @ L$='DQ' @ GOTO 9000 ! (Dernier quartier)
5230 IF FLAG(5) THEN L$='PL' ELSE L$='NL' ! (Nouvelle et pleine lune)
5240 C=(.1734-.000393*T)*SIN(M1)+.0021*SIN(2*M1)-.4068*SIN(M2)+.0161*SIN(2*M2)-.0004*SIN(3*M2)
5250 C=C+.0104*SIN(2*F)-.0051*SIN(M1+M2)-.0074*SIN(M1-M2)+.0004*SIN(2*F+M1)-.0004*SIN(2*F-M1)
5260 C=C-.0006*SIN(2*F+M2)+.001*SIN(2*F-M2)+.0005*SIN(M1+2*M2)
9000 J2=J2+C ! (Jour julien réel)
   - (Traduction du jour julien en date et heure)
9020 J2=J2+.5 @ Z=INT(J2) @ F=FP(J2)
9030 IF Z>=2299161 THEN Y=(Z-1867216.25) DIV 36524.25 @ Z=Z+1+Y-Y DIV 4
9040 B=Z+1524 @ C=(B-122.1) DIV 365.25 @ D=INT(365.25*C) @ E=(B-D) DIV 30.6001
9050 W=B-D-INT(30.6001*E)+F ! (Jour du mois, avec décimales)
9060 IF E>13.5 THEN E=E-13 ELSE E=E-1 ! (Mois)
9070 IF E>2.5 THEN C=C-4716 ELSE C=C-4715 ! (Année)
9080 W1=24*FP(W) @ W2=60*FP(W1) @ W3=60*FP(W2) ! (Heures, minutes, secondes.)
9090 IF W<10 THEN W$='0' ELSE W$=''
9100 W$=W$&STR$(INT(W))&'/'&L$&' à '&STR$(INT(W1))&'h'
9110 IF W2<10 THEN W$=W$&'0'
9120 W$=W$&STR$(INT(W2))&'mn'

============================================================
9130 DEF FNR(X,N) @ FNR=IP(X*N+.5)/N @ END DEF ! (Fonction d'arrondi.)
```

**********


AMIS HPTISTE BONNE...
BONNE ANNÉE 1986

## HP71 FORTH-ASSEMBLEUR

| | |
|---|---|
| PIERRE DAVID | LE DEBUGGER FORTH |
| ALAIN HERREMAN | BREAK (FORTH) |
| ALAIN HERREMAN | SORT (FORTH) |
| STEPHANE BARIZIEN,<br>PIERRE DAVID,<br>ET MICHEL MARTINET | CHAMPS ASSEMBLEUR |
| MICHEL MARTINET | ASSEMBLEUR ET MEMOIRE DE MASSE |
| JANICK TAILLANDIER | L'ORDRE EXIT (FOR STEP EXIT NEXT) |
| JEAN-FRANCOIS GARNIER | LEX PPOLL ET DEMANDE DE SERVICE |

## DES BOGUES, DES BOGUES !

Ca y est, le voilà !

Depuis le temps qu'on l'attendait ! Il fallait bien qu'il arrive un jour ! Tous les fous de l'assembleur vont pouvoir se régaler. Ils vont connaître le plaisir suprême : savoir comment sont vraiment les registres ! Plus de suppositions hasardeuses sur le contenu de tel ou tel octet. Et en plus, ils vont pouvoir suivre à la trace tous leurs petits (ou grands) programmes... Les veinards !

### HISTORIQUE

Tout à commencé par ce mot clé "DEBUG" dans la ROM Forth/Assembleur (le machin). Quoi de plus irritant que de savoir que Ca a failli exister ?

Les idées fleurissaient, mais aucune réalisation ne voyait le jour. Et puis est arrivé "User", la revue de PPC (CHHU) Danemark. Et dedans, un débogueur de quelques 900 octets, tout en assembleur ! Wouah !

Seulement, voilà : après s'en être un peu servi, les restrictions apparurent. En effet, la seule chose possible était de visualiser les registres. En dehors, point de salut.

Alors, je pris mon 71 dans une main, mes IDS dans l'autre, et je partis en quête d'un débogueur plus utilisable.

L'avantage de l'assembleur est la rapidité. Soit. Mais l'inconvénient, c'est qu'on est prisonnier du programme. Et à moins de faire un superbe gros bout d'assembleur, le problème n'est toujours pas résolu !

Alors, il faudrait le faire en langage évolué ? Oui. En Basic ? Heu... non. Basic est un horrible destructeur de Math-Stack, d'AVMEMS et autres pointeurs rocambolesques. Et bien d'autres problèmes encore.

Il n'y a donc pas de solution ? Si. J'ai nommé Forth.

Les avantages de Forth sont nombreux. D'abord, et à égalité en tête, nous trouvons :
- Forthram est à une adresse fixe, et
- Forth n'écrase pas tous les pointeurs !

Les autres avantages sont, pêle-mêle : rapidité, compacité, modularité, langage évolué...

### PRINCIPE

Accrochez vos ceintures !

Vous voulez voir le contenu de vos registres et l'état de la mémoire à un point donné du programme. Simple, vous insérez GOSBVL #2F9DF à ce point, vous assemblez et vous lancez !

En #2F9DF se situe la fin de la Reserved Ram. Sept octets sont nécessaires pour réaliser un GOVLNG dans la Forthram. Là, un bout d'assembleur sauve les registres et quelques bouts de mémoire (ce peut être utile), et donne le contrôle à un mot (QUIT ou CONTROLE EXECUTION), qui se charge d'arrêter la machine. Dorénavant, vous avez la main, vous pouvez faire ce que bon vous semble : afficher les registres, visualiser des zones mémoire, modifier le contenu de ce qui vous passe par la tête, y compris et surtout les registres ! Vous êtes le seul maître à bord après le HP71...

### UTILISATION

Insérez vos points d'arrêt dans votre fichier source. Ceci se fait par GOSBVL #2F9DF.

Assemblez. Tant que vous êtes en Forth, faites DEBOG. Une seule fois par Memory Lost (la reserved ram n'est modifiée que dans ce cas).

Mettez 1 dans CONTROLE-EXECUTION, et 1 dans PAS, éventuellement le nom de votre fichier dans la variable FICHIER.

Sortez de Forth (ou restez-y) et exécutez votre fonction, programme binaire (ou mot Forth). Les registres s'affichent : vous êtes sous contrôle du débogueur.

Là, faites ce qu'il vous plait de faire.

Pour le retour à votre programme assembleur, jusqu'au prochain point d'arrêt (ou jusqu'à la fin), tapez RETOUR.

Voilà. Simple, non ?

NOTES

Un certain nombre de choses sont à savoir.

Le programme est prévu pour être utilisé avec une interface vidéo. Si vous ne disposez que d'une ligne d'affichage, vous êtes vivement invités à modifier .REG et le message d'invite à votre gré.

A, B, C, D, D1 et D0 ne représentent plus des nombres hexa, mais des adresses. Si vous travaillez en hexa (ce que je crois), vous aurez intêret à insérer des 0 en tête (le chiffre C devient 0C).

Vous pouvez (devez) modifier ce débogueur, et faire part aux autres utilisateurs des modifications. Forth est une boule d'argile que chacun peut modeler à sa guise. Ne l'oubliez jamais.

Ce débogueur mange une partie de la Math-Stack. Je n'ai pas eû le temps d'empêcher cela. Il vaut mieux avoir un débogueur qui détruit la Math-Stack que pas de débogueur du tout !

Je vous conseille d'éplucher le source de ce programme (peut-être pas la partie assembleur, qui n'est pas passionnante). En effet, pour bien l'utiliser, il faut savoir comment ça marche. Je suis à votre entière disposition pour toute aide sur ce débogueur. N'hésitez pas à écrire (avec une enveloppe timbrée pour la réponse, ça va beaucoup plus vite !), je suis là pour ça !

REMERCIEMENTS

Je tiens à remercier ici nos amis Danois, sans qui cela n'aurait vraissemblablement pas vu le jour aussi vite.

Je tiens aussi à remercier Jean-Jacques Dhénin, Michel Martinet, et Janick Taillandier pour leur soutien, et leur aide dans le débogage de ce débogueur !

Allez, déboguez bien, et à bientôt !

(dites le vite, une fois !)

Pierre DAVID (SIG#1, P#37, CHHU#616)

Pierre DAVID
33 Bld St Martin
75003 PARIS

```
        FORTH
        TITLE  **** DEBOGUEUR EN FORTH ****
MLFFLG  EQU    #2F870
TRFMBF  EQU    #2F8C5
CSRC5   EQU    #1B41B
CSLC5   EQU    #1B435
oONERR  EQU    #2FBB6
oERR?   EQU    #2FBBB
DSTKAD  EQU    #2FAFD
ACTIVE  EQU    #2FB0C
oFLD    EQU    #2FB7F
oR0     EQU    #2FB16
oBASIC  EQU    #2FBA2
SVSTAT  EQU    #E47C7
GTSTAT  EQU    #E1BA5
NEXT00  EQU    #E71DD
DSPCHC  EQU    #01C3C

        WORD   'DEBOG'
        A0EX
        D0=(5) #2F9DF
        LC(2)  #D8
        DAT0=C B
        D0=D0+ 2
        LC(5)  CNTRL
        DAT0=C A
        D0=A
        RTNCC

**********
STOD1   C=DAT1 W
        D1=D1+ 16
STO     DAT0=C W
        D0=D0+ 16
        RTN
STO5D1  C=DAT1 A
        D1=D1+ 5
        DAT0=C A
        D0=D0+ 5
        RTN
        STITLE **** ASSEMBLEUR > FORTH ****
CNTRL   R4=C
        CSLC
        CSLC
        CSLC
```

```
        CSLC
        CSLC
        CDOEX
        DO=(5) REG
        DAT0=C A        DO
        C=0    B
        GONC   CNT10
        C=C+1  B
CNT10   DO=DO+ 5
        CD1EX
        DAT0=C A        D1
        DO=DO+ 5
        CD1EX
        DAT0=C 1        Cy
        DO=DO+ 1
        C=P    15
        DAT0=C S        P
        DO=DO+ 1
        C=0    S
        C=C-1  S
        DAT0=C S        F si Hex, 9 si Déc
        DO=DO+ 1
        C=ST
        DAT0=C 3        ST
        DO=DO+ 3
        C=0    A
        P=     7
        ?ST=0  15
        GOYES  CNT41
        C+P+1
CNT41   P=     3
        ?ST=0  14
        GOYES  CNT42
        C+P+1
CNT42   P=     1
        ?ST=0  13
        GOYES  CNT43
        C+P+1
CNT43   P=     0
        ?ST=0  12
        GOYES  CNT44
        C+P+1
CNT44   DAT0=C 1        PST (Prog. STatus: S15 à
        DO=DO+ 1            S12)
        C=0    A
        ?XM=0
        GOYES  CNT20
        C+P+1
CNT20   P=     1
        ?SB=0
        GOYES  CNT21
        C+P+1
CNT21   P=     3
        ?SR=0
        GOYES  CNT22
        C+P+1
CNT22   P=     7
        ?MP=0
        GOYES  CNT30
        C+P+1
CNT30   DAT0=C 1        HST (MP,SR,SB,XM)
        DO=DO+ 1        (Poids fort -> faible)
CNT40   C=RSTK
        DAT0=C A        RSTK
        DO=DO+ 5
        P=P-1
        GONC   CNT40
        GOSBVL CSRC5
        RSTK=C
        C=A    W
        GOSUB  STO      A
        C=B    W
        GOSUB  STO      B
        C=R4
        C=RSTK
        GOSUB  STO      C
        C=D    W
        GOSUB  STO      D
        C=R0
        GOSUB  STO      R0
        C=R1
        GOSUB  STO      R1
        C=R2
        GOSUB  STO      R2
        C=R3
        GOSUB  STO      R3
        D1=(5) 1+(MLFFLG)   Une autre facon de
        GOSUB  STOD1        dire STMTR0...
        GOSUB  STOD1
        GOSUB  STO5D1
        GOSUB  STO5D1
        GOSUB  STOD1
        GOSUB  STOD1
        GOSUB  STO5D1
        GOSUB  STO5D1
*
* Maintenant, nous rendons le contrôle à
* l'interpréte FORTH
*
        SETHEX
        P=     0
        DO=(5) oONERR
        C=0    A
        DAT0=C A
        DO=(5) oERR?
        DAT0=C A
        DO=(5) DSTKAD
        C=DAT0 A
        D1=C
        DO=(5) oR0
```

```
        C=DAT0 A
        R3=C
        D0=(5) ACTIVE
        A=DAT0 A
        C=0    A
        C=C+1  A
        DAT0=C A
        D0=(5) oFLD
        DAT0=A A
        ?A#0   A
        GOYES  DEJA
        GOSBVL SVSTAT
DEJA    D0=(5) oBASIC
        C=0    A
        DAT0=C A
        C=R3
        B=C    A
        GOSUB  SETI
PINIT   BSS    5        CFA du mot central du
SETI    C=RSTK          débogueur, connu plus tard
        CD0EX
        GOVLNG NEXT00

        STITLE **** FORTH > ASSEMBLEUR ****
        WORD   'RETOUR'
        CD1EX
        D1=(5) DSTKAD
        DAT1=C A
        D0=(5) oFLD
        C=DAT0 A
        D0=(5) ACTIVE
        DAT0=C A
        ?C#0   A
        GOYES  DEJA2
        GOSBVL GTSTAT
DEJA2   D0=(5) HST
        C=DAT0 1
        GOSUB  XM=1
        SB=0
        CSRB
        ?SB=0
        GOYES  RET05
        XM=0
RET05   D0=(5) REGFIN
        D1=(5) TRFMBF
        GOSUB  RCL5D1
        GOSUB  RCL5D1
        GOSUB  RCLD1
        GOSUB  RCLD1
        GOSUB  RCL5D1
        GOSUB  RCL5D1
        GOSUB  RCLD1
        GOSUB  RCLD1
        GOSUB  RCL
        R3=C            R3
        GOSUB  RCL
        R2=C            R2
        GOSUB  RCL
        R1=C            R1
        GOSUB  RCL
        R0=C            R0
        GOSUB  RCL
        D=C    W        D
        D0=D0- 16
        GOSUB  RCL
        B=C    W        B
        GOSUB  RCL
        A=C    W        A
        P=     7
RET10   D0=D0- 5
        C=DAT0 A        RSTK
        RSTK=C
        P=P-1
        GONC   RET10
        D0=D0- 1        HST (MP,SR,SB,XM)
        C=DAT0 1
        CSRB
        SB=0
        CSRB            Maintenant, SB est remis
        SETHEX
        D0=D0- 1        PST (Program STatus: S15 à
        C=DAT0 S             S12)
        C=C+C  S
        ST=0   15
        GONC   RET15
        ST=1   15
RET15   C=C+C  S
        ST=0   14
        GONC   RET14
        ST=1   14
RET14   C=C+C  S
        ST=0   13
        GONC   RET13
        ST=1   13
RET13   C=C+C  S
        ST=0   12
        GONC   RET12
        ST=1   12
RET12   D0=D0- 3
        C=DAT0 3        ST
        ST=C
        D0=D0- 1        Mode Hex ou Déc
        C=DAT0 S
        C=C+1  S
        GOC    HEX      C valait #F
        SETDEC
HEX     D0=D0- 1        P
        C=DAT0 S
        P=C    15
        D0=D0- 1        Cy
```

```
        C=DAT0 S
        ?C#0   S
        GOYES  RET20
RET20   D0=(5) D1       D1
        C=DAT0 A
        D1=C
        D0=(5) D0       D0
        C=DAT0 A
        D0=(5) RET99
        DAT0=C A
        D0=(5) C
        C=DAT0 W
        NIBHEX 1B       D0=(5) #00000
RET99   BSS    5
        RTN
RCL     D0=D0- 16
        C=DAT0 W
XM=1    RTNSXM
RCLD1   GOSUB  RCL
        D1=D1- 16
        DAT1=C W
        RTN
RCL5D1  D0=D0- 5
        C=DAT0 A
        D1=D1- 5
        DAT1=C A
        RTN
        STITLE **** REGISTRES DU 1LF2  ****
REG
D0      BSS    5
D1      BSS    5
Cy      BSS    1
P       BSS    1
MODE    BSS    1
ST      BSS    3
PST     BSS    1
HST     BSS    1
PC      BSS    5
RSTK    BSS    5
        BSS    5
        BSS    5
        BSS    5
        BSS    5
        BSS    5
        BSS    5
        BSS    5
A       BSS    16
B       BSS    16
C       BSS    16
D       BSS    16
R0      BSS    16
R1      BSS    16
R2      BSS    16
R3      BSS    16
SR0     BSS    16
SR1     BSS    16
SD0     BSS    5
SD1     BSS    5
FR0     BSS    16
FR1     BSS    16
FD0     BSS    5
FD1     BSS    5
REGFIN
        STITLE ****    UTILITAIRES     ****
        WORD   'PMUD'
        CD0EX
        RSTK=C
        C=B    A
        RSTK=C
        C=DAT1 A
        D1=D1+ 5
        A=DAT1 A
        D1=D1+ 5
        A=A+C  A
        A=A-1  A
        D0=A
        CD1EX
        RSTK=C
        CD1EX
        GONC   PMUD40   B.E.T.
PMUD10  RSTK=C
        A=0    B
        A=DAT0 1
        D0=D0- 1
        CD0EX
        RSTK=C
        LC(2)  9
        ?A>C   B
        GOYES  PMUD20
        LCASC  '0'
        GONC   PMUD30     B.E.T.
PMUD20  LC(2)  'A'-10
PMUD30  C=C+A  B
        GOSBVL DSPCHC
        C=RSTK
        D0=C
        C=RSTK
PMUD40  C=C-1  A
        GONC   PMUD10
        C=RSTK
        D1=C
        C=RSTK
        B=C    A
        C=RSTK
        D0=C
        RTNCC

        WORD   'DCST'   Debogger ConSTants
        RTNCC
        CON(5) PINIT
        CON(5) REG
```

```
        END


( ********************************************** )
( *      Module d'affichage des registres      * )
( ********************************************** )

( But: exploiter l'image des registres stockée en)
( mémoire. Ce module fournit l'ensemble des      )
( secondaires pour visualiser les registres et   )
( la zone "Statement & Function Scratch" .       )

( Nous emploierons abusivement, dans ce qui suit )
( le terme registre pour désigner la zone de     )
( scratch. La différence se situe au niveau      )
( conceptuel ; mais au niveau strictement        )
( physique, nous raisonnons sur une image de ce  )
( qui existait à un instant donné, et tout est   )
( traité de manière identique.                   )

( Principe : On crée une table de constantes,    )
( pour désigner les adresses des registres. Les  )
( mots qui permettront d'afficher ces registres  )
( seront les mêmes, avec un point "." devant.    )
( Ainsi, R0 désigne le registre R0, mais .R0 est )
( l'action affichant le contenu de R0.           )

BASE @ DECIMAL

( Adresse de base de la zone image               )
' DCST 7 + DUP @ SWAP 5 + @    ( PINIT REG )

( ---------------------------------------------- )
( -      Table des adresses de registres       - )
( ---------------------------------------------- )

DUP CONSTANT REG
DUP CONSTANT D0 5+
DUP CONSTANT D1 5+
DUP CONSTANT Cy 1+
DUP CONSTANT P 1+
DUP CONSTANT MODE 1+
DUP CONSTANT ST 5+
DUP CONSTANT PC 5+
DUP CONSTANT RSTK 7 5 * +
DUP CONSTANT A 16 +
DUP CONSTANT B 16 +
DUP CONSTANT C 16 +
DUP CONSTANT D 16 +
DUP CONSTANT R0 16 +
DUP CONSTANT R1 16 +
DUP CONSTANT R2 16 +
DUP CONSTANT R3 16 +
DUP CONSTANT SR0 16 +
DUP CONSTANT SR1 16 +
DUP CONSTANT SD0 5 +
DUP CONSTANT SD1 5 +
DUP CONSTANT FR0 16 +
DUP CONSTANT FR1 16 +
DUP CONSTANT FD0 5 +
CONSTANT FD1

( Variable FICHIER : Elle sert à stocker le nom  )
( du fichier courant. Ainsi, tous les affichages )
( du pointeur programme (mais pas de la pile)    )
( seront relatifs au début du fichier. Les       )
( adresses sont alors compatibles avec le        )
( listing issu de l'assemblage.                  )
8 STRING FICHIER

: 2SP SPACE SPACE ;
: .A ." A =" A 16 PMUD ;
: .B ." B =" B 16 PMUD ;
: .C ." C =" C 16 PMUD ;
: .D ." D =" D 16 PMUD ;
: .R0 ." R0=" R0 16 PMUD ;
: .R1 ." R1=" R1 16 PMUD ;
: .R2 ." R2=" R2 16 PMUD ;
: .R3 ." R3=" R3 16 PMUD ;
: .D0 ." D0=" D0 5 PMUD ;
: .D1 ." D1=" D1 5 PMUD ;
: .Cy ." Cy=" Cy 1 PMUD ;
: .P  ." P=" P 1 PMUD ;
: .MODE ." Mode=" MODE N@
  9 = IF ." Dec" ELSE ." Hex" THEN ;
: .ST " ST=0123456789012345XBRP" ( adr nb)
  PAD SMOVE PAD
  ST @ 1 ( adr ST masque )
  23 3 DO
    2DUP   ( adr ST masque ST masque )
    AND    ( adr ST masque bit_correspondant )
    0= IF  ( adr ST masque )
      3 PICK I 2* + 45 SWAP C! THEN
    2*     ( adr ST masque*2)
  LOOP
  2DROP 23 TYPE ;
: .PC BASE @ HEX
." PC=" PC @
FICHIER DUP 0= IF 2DROP 0 ELSE FINDF THEN
- 0 <# # # # # # #> TYPE
BASE !
;
: .RSTK RSTK
   7 0 DO
     ." S" I 48 + EMIT ." ="
     DUP 5 PMUD 2SP 5+
  LOOP DROP ;
: .SR0 ." STMTR0=" SR0 16 PMUD ;
```

```
: .SR1 ." STMTR1=" SR1 16 PMUD ;
: .SD0 ." STMTD0=" SD0 5 PMUD ;
: .SD1 ." STMTD1=" SD1 5 PMUD ;
: .FR0 ." FUNCR0=" FR0 16 PMUD ;
: .FR1 ." FUNCR1=" FR1 16 PMUD ;
: .FD0 ." FUNCD0=" FD0 5 PMUD ;
: .FD1 ." FUNCD1=" FD1 5 PMUD ;
: .REG
  .A  SPACE .B  SPACE .C  SPACE .D  CR
  .R0 SPACE .R1 SPACE .R2 SPACE .R3 CR
  .PC 2SP .D0 2SP .D1 2SP .MODE 2SP
  .ST 2SP .Cy 2SP .P CR
  .RSTK CR ;

( Attention, on met QUIT comme mot de contrôle   )
( du débogueur. Ainsi, l'utilisateur qui a       )
( oublié de charger le module suivant n'aura pas )
( de beau Memory Lost (du moins pas à cause de   )
( moi !). Ce module est donc utilisable sans le  )
( suivant...                                     )
' QUIT SWAP !       ( PINIT )
BASE !



( ********************************************** )
( *     Module de contrôle de l'exécution      * )
( ********************************************** )

( But: prendre le contrôle à chaque fois que     )
( l'on rentre dans le débogueur, pour déterminer )
( si on doit s'arrêter ou non.                   )

( Modes gérés par le module :                    )
( Mode 1 = T (Trace) : Passage à travers un      )
(   certain nombre de points d'arrêts, avec      )
(   affichage des registres ".REG" à chaque      )
(   point d'arrêt sauté.                         )
( Mode 2 = U (Untrace) : similaire à T, sans     )
(   affichage des registres à chaque passage.    )
( Mode 3 = G (Go) : Exécution directe, jusqu'au  )
(   point d'arrêt dont l'adresse est indiquée    )
(   soit en absolu (variable "Fichier" vide),    )
(   soit en relatif (variable "Fichier" contient )
(     le nom du fichier)                         )

( Variables :                                    )
( MODE-EXEC : Contient le mode d'exécution       )
(   choisi, soit une valeur de 1 à 3.            )
( PAS : suivant la valeur de MODE-EXEC,          )
(   MODE-EXEC=1 ou 2 : PAS indique le nombre de  )
(     points d'arrêt à sauter.                   )
(   MODE-EXEC=3 : PAS indique l'adresse où doit  )
(     s'arrêter le débogueur (en relatif ou en   )
(     absolu, suivant le contenu de FICHIER).    )

VARIABLE MODE-EXEC       ( 1:T, 2:U, 3:G )
VARIABLE PAS             ( Nombre de sauts )

0 MODE-EXEC ! 1 PAS !    ( Initialisations )

( ---------------------------------------- )
( -                G-STOP?                 - )
( ---------------------------------------- )
( But: décider de l'arrêt si on est en mode "G". )
( Pour cela, G-STOP? compare la valeur courante  )
( du pointeur programme (PC) et la valeur de PAS )
( éventuellement ajoutée à l'adresse du fichier. )
( Si égalité, on renvoie le flag vrai (-1) pour  )
( signaler l'arrêt.                              )

: G-STOP?    ( -> f )
  FICHIER FINDF          ( adr. du fichier ou 0 )
  PAS @ +                ( ajout de l'adresse   )
  PC @                   ( pointeur programme   )
  =                  ( Vrai si stop, fx si cont.)
;

( ---------------------------------------- )
( -                T-U-STOP?               - )
( ---------------------------------------- )
( But: renvoyer la valeur vraie (-1) si les PAS  )
( points d'arrêt ont été sautés. PAS est         )
( décrémenté jusqu'à atteindre la valeur 0.      )
( Important: une valeur négative de PAS provoque )
( l'arrêt. C'est le cas typique où on oublie de  )
( remettre une bonne valeur avant le retour.     )
( Comme ce genre de bévue arrivera souvent (!!!) )
( je fais les tests en mode "signé": une valeur  )
( négative arrête donc bien le débogueur, et il  )
( n'est pas besoin de réactualiser PAS à chaque  )
( arrêt. Attention, ne le faites que 2^19 fois ! )

: T-U-STOP?              ( -> f )
  PAS @ 1- DUP PAS !     ( pas )
  DUP 0= SWAP 0< OR  ( Vrai si stop, fx si cont.)
;

( ---------------------------------------- )
( -                STOP?                   - )
( ---------------------------------------- )
( But: décider de l'arrêt en fonction du mode    )
( choisi. Une autre valeur que 1, 2 ou 3         )
( provoque l'arrêt. C'est en fait le groupement  )
( des deux mots précédents.                      )

: STOP?                  ( -> f )
  MODE-EXEC @            ( mode-exec courant )
  CASE
   1 OF T-U-STOP? ENDOF  ( mode T )
```

```
   2 OF T-U-STOP? ENDOF  ( mode U )
   3 OF G-STOP?   ENDOF  ( mode G )
        -1               ( Vrai ! )
  ENDCASE ;

( ------------------------------------------ )
( -            CONTROLE-EXECUTION            - )
( ------------------------------------------ )
( But: agir à chaque entrée dans le débogueur.  )
( Ce mot prend le contrôle et décide de l'arrêt, )
( et de l'affichage éventuel des registres.     )
( C'est le mot appelé par la routine assembleur, )
( et vous noterez qu'il contient un "ABORT".    )

: CONTROLE-EXECUTION
  STOP?                ( vrai: stop, faux: cont. )
  IF .REG ABORT" --- Debogueur ! ---"
     ELSE
       MODE-EXEC @ 1 = IF .REG CR THEN
       RETOUR
     THEN
;

( maintenant, il s'agit de mettre l'adresse du   )
( mot CONTROLE-EXECUTION au bon endroit...       )
' CONTROLE-EXECUTION
' DCST 5+ 2+ @
!
( Ca y est, maintenant le débogueur est          )
( utilisable !                                   )
```

---

BREAK - FORTH

Cet article a pour but de vous présenter l'adaptation d'un mot paru dans le Forth Dimension Vol. 5 no1 pour le forth du 71. Souvenez vous du temps que vous avez passé à faire votre dernier mot forth, éh oui ! Votre samedi après-midi et peut être même une partie du dimanche. Et pourquoi tant de temps ? Simplement parcequ'il vous manquait un DUP, ou au contraire un nombre se trouvait dans la pile alors qu'il n'aurait pas dû y être. Mais comment s'en rendre compte ?

Voici l'objet de BREAK : un BREAK placé dans un endroit stratégique de votre mot interrompt MOMENTANEMENT son exécution en vous permettant d'exécuter tout les mots que vous voulez (entre autre, vérifier l'état des piles, et si nécessaire dans changer les valeurs).

Comment cela est il possible ?

Malheureusement ce mot est désespérément simple, et je n'impressionerai personne en l'expliquant.

Le principe est très simple: BREAK n'est qu'une boucle (donc interruption de l'exécution) qui exécute ce que vous tapez au clavier, ce qui revient à "créer" un nouvel interpréteur. Pour cela deux mots: QUERY et INTERPRET.

QUERY interrompt la boucle et vous permet de rentrer ce que vous voulez au clavier jusqu'à la pression de END-LINE puis dépose tout cela dans le T.I.B. et INTERPRET intreprête (étonnant non ?) ce qui se trouve dans le T.I.B. ( c'est-à-dire ce que vous avez rentré au clavier) car on a précédemment chargé 0 dans la variable BLK ( sinon il exécuterait ce qui se trouve dans le "Masse Storage Buffer"... ). Pour sortir de cette boucle (i.e. continuer l'exécution) il suffit de taper le mot CONT qui change la valeur d'une variable qui est testée lors de la boucle de BREAK et donc permet d'en sortir (boucle BEGIN... WHILE... REPEAT).

Le fond étant compris voici quelques raffinements:

Lors de la boucle de BREAK ce n'est plus OK { X } qui est affiché mais OK' { X' } pour bien vous indiquer que vous êtes dans la boucle de BREAK et le X' vous indique le nombre de valeurs que vous avez rajoutées dans la pile (c'est-à-dire le nombre réel de valeurs dans la pile moins celui des valeurs dont se sert le programme). Si vous voulez voir les valeurs dont se sert le mot il suffit de taper .S .

Par sécurité (éviter le Memory Lost) on vérifie que le nombre de valeurs dans la pile de retour est inchangé, c'est-à-dire que vous pouvez changer leur valeur mais pas leur quantité.

Au total nous avons besoin de trois variables; une qui contient la position de la Return Stack avant la boucle, une qui contient le nombre de valeurs de la Data Stack avant la boucle et une qui contient la valeur de la variable testée pour sortir de la boucle, et donc pour éviter de

créer trois entêtes on crée une "super variable" pouvant contenir trois nombres (VBREAK).

Alain HERREMAN

```
CREATE CBREAK F ALLOT
: BREAK
RP@ F - CBREAK A + ! 0 BLK !
1 CBREAK 5+ ! DEPTH CBREAK !
BEGIN QUERY INTERPRET CBREAK 5+ @ WHILE ."
OK' { " DEPTH CBREAK @ - . ." }" CR REPEAT ;
: CONT 0 CBREAK 5+ ! CBREAK A + @ RP@ <>
IF ." Can't resume !" QUIT THEN ;
```

---

TRI EN FORTH

Voici un exemple de récursivité en forth, il s'agit de trier par ordre décroissant les nombres de la Data Stack (SORT).

Autant vous dire que ce mot ne me sert absoluement à rien, mais il m'a amené à créer d'autres mots plus utiles (déjà parcequ'ils servent à créer SORT ! ), il s'agit de NDROP et PICK> . NDROP supprime le nième nombre de la pile de données et PICK> fait l'équivalent de PICK : rappeler le nième nombre de la pile mais sans le dupliquer, c'est-à-dire le supprime de son ancienne place. Bravo à ceux qui avaient fait le rapprochement entre PICK et PICK> (et colegramme...) avant que je ne l'explique ! SORT quant à lui, il faut bien en parler un peu, détermine le max. des éléments de la pile en comparant les éléments 2 par 2 et en laissant le plus grand sur le dessus, le max. déterminé il le met dans la Return Stack puis recommence à déterminer le max. des nombres qui restent (d'où la récursivité)... Quand il ne reste plus qu'un nombre dans la pile (c'est le min.) on remet tous les nombres de la Return dans la Data Stack, mais cette fois dans l'ordre (silence émerveillé...). Si vous préférez avoir les nombres dans l'ordre croissant il suffit de changer MIN en MAX et MAX en MIN, enfin on se sert de MYSELF qui permet la récursivité en rappelant le mot dans lequel il est placé cela en chargeant lors de la compilation le CFA du mot.

Alain HERREMAN

```
: NDROP
DUP DEPTH 2- >
IF . ." Trop grand." " BEEP"BASICX ELSE
SP@ 5+ DUP 5+ ROT 1- 5 * NMOVE> DROP THEN ;
: PICK>
1+ DUP PICK SWAP NDROP ;
: SORT
DEPTH 1- IF DEPTH 1+ 2 DO I PICK> 2DUP
MIN ROT ROT MAX LOOP >R MYSELF R> THEN ;
```

---

RESPECTEZ LES CHAMPS DE VOS FICHIERS SOURCES

Voici un petit Lex qui nous le pensons sera très utile à tout ceux qui en on marre de chercher ou se trouve leur curseur lorsqu'ils font de l'assembleur en plein format.

PRINCIPE:
Ce lex vous permet de faire avancer le curseur aux bonnes colonnes, pour respecter les champs Label, Mnémonique, Modificateur et Commnentaire lors que voulez faire un fichier source.
Pour activer ce mode "champs" il vous suffit de presser [f[,]] et le tour est joué. Pour en sortir vous faites de même: c'est bi-stable !
Nous sommes encore en présence de ce bon vieux poll "pKYDF" qui cette fois-ci est intercepté avec une redéfinition de touche.

Voilà, le listing source n'est pas très grand mais si vous êtes trop fatigués reportez vous à la rubrique "le coin des Lhex".

Bon assemblage
A bientôt
Heureuse Programmation

Stéphane BARIZIEN
Pierre DAVID
Michel MARTINET

```
       LEX    'ASFLEX'
       ID     #E1
       MSG    0
       POLL   POLHND
       ENDTXT
        *
pKYDF  EQU    #1B
ASRW5  EQU    #0ED0A
RESERV EQU    #2F986
DEFADR EQU    #2F967
ATNDIS EQU    #2F441
KcCALC EQU    #17
FLGREG EQU    #2F6E9
DSPBFS EQU    #2F480
DSPBFE EQU    #2F540
CURSOR EQU    #2F47E
SPCKEY EQU    #31
MNEMF  EQU    7
MODF   EQU    14
COMF   EQU    23
        *
************************************************
* Auteur: Stéphane Barizien 03/85 - Interception
*         de pKYDF et champs assembleur.
* 1ère modification: M.MARTINET et P.DAVID 10/85 -
*       transformation en FIELD ON | OFF.
* 2ème modification: M.MARTINET 12/85 - mise en
*       place de la bascule touche CALC et suppres-
*       sion token FIELD ON | OFF.
************************************************
        *
POLHND LC(2)  pKYDF
       ?B=C   B
       GOYES  POLL10   Ok: on exécute le Lex
       RTNSXM          Autrement fin.
POLL10 D0=(5) ATNDIS   Vérification de l'état
       C=DAT0 S        de
       ?C=0   S        la touche ATTN
       GOYES  rtn      Si elle est libre: fin
       D0=(4) 14+FLGREG ) Est-on sous éditeur
       C=DAT0 S        de
       C=C+C  S        textes ?
       GONC   rtn      Non: retour.
       A=R0            A[A] = touche pressée
       C=0    A
       LC(2)  KcCALC   Est-ce
       ?A#C   A        la touche CALC ?
       GOYES  FXQ      Non: exécution suite prgm.
       D0=(4) RESERV   Autrement: changement
       A=DAT0 P        d'état
       LC(1)  2        du flag
       B=A    A        FIELD
       C=C&A  A        "
       ?C=0   P
       GOYES  Fon
       C=0    A
       GOC    Fok
Fon    LC(1)  2
Fok    C=C+B  P
       DAT0=C P
       GOTO   rtnxm0   et fin d'exécution
rtn    RTNSXM
FXQ    D0=(4) RESERV   Exécution du programme
       A=DAT0 1        Est-on en mode field on ?
       LC(1)  2
       A=A&C  P
       ?A=0   P
       GOYES  rtn      Non: fin de programme
       D0=(4) DSPBFS   Oui: continu
       LC(3)  (DSPBFE)-(DSPBFS)-1
       B=C    X
SP05   A=DAT0 B
       ?A=0   B
       GOYES  SP08
       LCASC  ' '
       ?A#C   B
       GOYES  SP06
       D0=D0+ 2
       B=B-1  X
       GONC   SP05
rtn1   GOC    rtn      B.E.T.
SP06   LCASC  '*'      Y a-t-il des remarques ?
       ?A=C   B
       GOYES  rtn      Oui: fin de programme
SP08   A=R0            Autrement: la touche
       GOSBVL ASRW5    pressée est-elle
       LC(2)  SPCKEY   un espace ?
       ?A#C   B        Si non:
       GOYES  rtn      fin de programme
       D0=(4) CURSOR   Vérification
       A=DAT0 B        position du curseur
       LC(2)  (MNEMF)-1 en fonction
       ?A<=C  B        des
       GOYES  SP10     différents
       LC(2)  (MODF)-1 champs:
       ?A<=C  B        MNEMF: mnémo-field
       GOYES  SP10     MODF : modification-field
       LC(2)  (COMF)-1 COMF : comment-field
       ?A>C   B
       GOYES  rtn1
SP10   C=C-A  A        Redéfinition
       C=C+1  A        de la touche
       D0=(4) DEFADR   espace
       DAT0=C B        avec
       D0=D0+ 2        le
       LCHEX  4        bon
       DAT0=C 1        nombre
       GOSUB  SKPTBL   de
       NIBASC '        ' CARACTERES
```

```
       NIBASC ' '          BLANCS
SKPTBL C=RSTK              Sortie de table
       D0=D0+ 1
       DAT0=C A
       ST=1   0            ST[0]=1 => définit. touche
rtnxm0 XM=0                ST[0]=0 => touche inhibée
       RTNCC               Puis interception du Poll.

       END                 Fin du Lex
```

---

ASSEMBLEUR ET MEMOIRES DE MASSE

Voici comme je vous le disais dans notre dernier numéro quatre petites fonctions complémentaires HPIL.

MAXD(<Spécificateur d'appareil>)

Cette fonction renvoie le nombre maximum d'entrées dans le directory du support. Ce nombre est celui que vous avez indiqué en initialisant le support.
Par défaut, les cassettes sont initialisées à 128 entrées, les diskettes simple face à 264 entrées et les diskettes double face à 616 entrées.

MAXM(<Spécificateur d'appareil>)

Cette fonction renvoie la capacité maximale du support. Les cassettes font 131072 octets, les diskettes simple face 270336 octets et les diskettes doubles faces 630784 octets.

MEMD(<Spécificateur d'appareil>)

Cette fonction renvoie le nombre exact d'entrée non utilisées sur le support.

MEMM(<Spécificateur d'appareil>)

Cette fonction renvoie la mémoire (réelle) disponible sur le support. EXEMPLE: \Supposons que nous prenons une diskette double face venant juste d'être initialisée.

MAXM(":TAPE") retourne 630784, c'est la capacité maxi de la diskette.
630784 = 2464 enregistrements de 256 octets.

MAXD(":TAPE") retourne 616, c'est le nombre maxi d'entrée dans le directory. Il y a 8 entrées par enregistrements d'où l'on obtient la longueur totale de celui-ci : 616/8 = 77 enregistrements (soit 19712 octets).

MEMD(":TAPE") renvoie 616 puisque aucun fichier n'y est enregistré.

MEMM(":TAPE") renvoie 610560, c'est la mémoire réellement disponible sur le support. En plus du directory, les 512 premiers octets de tout support de mémoire de masse sont réservés "au service", c'est-à-dire pour coder toutes les informations nécéssaires pour la lecture/écriture sur le support.
630784 - (19712 + 512) = 610560
MAXM DIR RES MEMM

Voilà, je me tiens à votre entière disposition pour tout autres renseignements.

Michel MARTINET (SIG#2, PC#12, CHHU#617)
T.A.O. B.C.M.W. Conseil

```
       LEX    'MMLEX'
       ID     #E1
       MSG    0
       POLL   0
       ENTRY  MXD
       CHAR   #F
       ENTRY  MXM
       CHAR   #F
       ENTRY  MDIR
       CHAR   #F
       ENTRY  MEM
       CHAR   #F
       KEY    'MAXD'
       TOKEN  57
       KEY    'MAXM'
       TOKEN  58
       KEY    'MEMD'
       TOKEN  59
       KEY    'MEMM'
       TOKEN  60
       ENDTXT

A-MULT EQU    #1B349
ATNFLG EQU    #2F442
Attn   EQU    12
```

```
BSERR  EQU    #0939A
CHKMAS EQU    #425C
CSRW5  EQU    #0ED2C
D=AVMS EQU    #1A460
DEVPAR EQU    #1BF0
ENDTAP EQU    #44D9
FNRTN4 EQU    #0F238
Flg    EQU    11
GDIRST EQU    #4843
GETERR EQU    #6791
GETMBX EQU    #3B62
HDFLT  EQU    #1B31B
I/OFND EQU    #118BA
LEXPIL EQU    #FF
MEMERR EQU    #0944D
R<RSTK EQU    #014DD
READR# EQU    #44FF
RSTK<R EQU    #014A8
SNAPBF EQU    #2F7F0
bLEX   EQU    #BFC
eABORT EQU    4
eNODEF EQU    3
ePIL   EQU    2
eTAPE  EQU    1
eXWORD EQU    #23
memflg EQU    10

************************************************
* La routine VERTAP verifie si le paramètre
* pointé par D1 correspond bien à une mémoire
* de masse d'identificateur appareil 16.
************************************************

Dvnfnd P=     ePIL     Classe d'erreur 32 à 47
       C=0    A        Erreur 32.
       GOC    erreur   B.E.T.

VERTAP GOSUB  JUMPER
       CON(5) DEVPAR   Routine d'analyse du
        *              spécificateur d' appareil
        *              de la fonction. En sortie
        *              D1 ^ Math Stack - 16 pour
        *              un paramètre numérique.
       GOC    erreur   Si carry: ERREUR
       ?D=0   X        D[X] contient l'adresse du
        *              périphérique. Si D[X] = 0,
       GOYES  Dvnfnd   le périphérique n'est pas
        *              présent. On renvoie
        *              "Device Not Found"
       CD1EX           ) Sauvegarde de D1 pour
       R0=C            ) le retour à BASIC (fin).
       GOSUB  JUMPER   Vérification de l'AID du
       CON(5) CHKMAS   périphérique (16).
       RTNNC           Cy=0 : OK, Cy=1 : ERREUR

************************************************
* Traitement des erreurs HPIL. P indique la
* classe d'erreur (mémoire de masse... etc.) et
* C[0] le numéro de l'erreur. Le retour à la
* main loop se fait via BSERR.
************************************************

erreur ?P=    eTAPE
       GOYES  ERROR1
       ?P=    ePIL
       GOYES  ERROR1
       ?P#    eABORT
       GOYES  ERROR0
       GOSUB  JUMPER
       CON(5) GETMBX
       GOSUB  atnchk
       GOC    ERROR0
       GOSUB  JUMPER
       CON(5) GETERR
       GONC   ERROR-
       ?P#    eABORT
       GOYES  ERROR1
ERROR- P=     eABORT
ERROR0 C=P    0
       P=     eNODEF
ERROR1 C=P    1
       P=     2
       LCHEX  FF
       A=C    A
       C=A    A
       P=     0
bserr  GOVLNG BSERR    Envoyez l'erreur !

************************************************
* M A X D
************************************************

       NIBHEX C11      Alpha ou un numérique.
MXD    CD0EX
       RSTK=C
       GOSUB  r<rstk
       GOSUB  VERTAP
       GOSUB  JUMPER
       CON(5) GDIRST   Get DIRectory STart
       GOC    erreur   Erreur si carry
       C=D    W        D[5:8] = l. du dir. (enr.)
       GOSBVL CSRW5    C[A] = longueur du dir.
       A=C    A        A[A] = longueur du dir.
       ASL    A        A[A] = 16 fois l. du dir.
       ASRB            A[A] = l. du dir. * 8

************************************************
* SORTIE: commune, RETOUR A BASIC : Replace
* 4 niveaux de pile des retours, le compteur
* programme dans D0, transforme A[A] en nombre
```

```
* flottant dans A[W], replace D1 à sa place
* sur la Math-Stack puis retour à Basic via
* FNRTN4. (Rembobine le support)
************************************************

SORTIE GOSUB  JUMPER   Rembobine le support
       CON(5) ENDTAP   de mémoire de masse
       GOC    Erreur
       P=     4        ) Restitution de 5 niveaux
       GOSBVL RSTK<R   ) de pile
       C=RSTK          ) puis de
       D0=C            ) D0  (4 niveaux & PC)
       GOSBVL HDFLT    A[W] = longueur
       C=R0            R0 = D1 (MS-16)
       D1=C            D1 = MS-16
       C=A    W        C[W] = longueur
       GOVLNG FNRTN4   retour à BASIC

Erreur GOTO   erreur   Ralonge

memerr GOVLNG MEMERR

************************************************
* M E M M
************************************************

       NIBHEX C11
MEM    ST=0   memflg
       ST=1   Flg
       GONC   MXMC

************************************************
* M E M D
************************************************

       NIBHEX C11
MDIR   ST=1   memflg
       ST=0   Flg
       GONC   MXMC

************************************************
* M A X M
************************************************

       NIBHEX C11      Alpha ou un numérique
MXM    CLRST           Ni memflg ni Flg
MXMC   CD0EX
       RSTK=C
       GOSUB  r<rstk
       GOSUB  VERTAP
       A=R0            R0 = MS-16
       C=D    A        ) D[X] = device address
       RSTK=C          ) RSTK = D[X]
       C=0    A
       LC(3)  #1F0     1F0 = 512-16 = 496
       A=A-C  A        A[A] = MS-512
       GOSBVL D=AVMS
       C=A    A        C[A] = AVMEMS
       ?C<D   A        Y a-t-il assez de mem. ?
       GOYES  memerr   Non: Erreur
       R1=C            Sauvegarde de MS-512
       D1=C            D1 = Math Stack - 512
       A=0    A        A[A] = # enreg. à lire
       C=RSTK          ) Restitution de
       D=C    A        ) dev. add. dans D[X]
       GOSUB  JUMPER   ) Lecture de l'enreg. A[A]
       CON(5) READR#   ) en commencant à D1 ^
       GOC    Erreur
       A=R1            A[A] = MS-512
       ?ST=1  memflg   Est-on entré via MEMDIR ?
       GOYES  MDIRXQ
       C=0    A
       LC(2)  48
       C=C+A  A
       D1=C            ^ pointe ^ l. max support
       P=     4        ) Calcul de
       GOSUB  GETALR   ) la longueur maximale
       GOSUB  getA*    ) du support
       GOSUB  getA*    ) de mémoire de masse
       ?ST=0  Flg      Est-on entré via MAXMEM
       GOYES  MXM10
       GOSUB  MEM00    Autrement exécution de MEMM
MXM10  ASL    A        ) multiplication de A[A]
       ASL    A        ) par 256 (dclg. d'1 octet)
       GOTO   SORTIE   pré retour à BASIC

MEM00  B=A    A        B[A] = # enreg. maxi
       A=R1
MDIRXQ D1=A            D1^ MS-512
       D1=D1+ 16       D1^ codage # Rec. Start DIR
       P=     4
       GOSUB  GETALR   => A[A]= 00002 dir. start
       B=B-A  A        B[A] = # maxrec - vol.label
       R2=A            R2= dir. start
       D1=D1+ 8        D1^ code long. directory
       P=     4
       GOSUB  GETALR   => A[A]= 0xxxx Dir.Len
       B=B-A  A        B[A]=maxmem après format.
       ?ST=1  Flg      Est-on entré via MEMM
       GOYES  MEM10
       C=A    A        Sauvegarde de DirLen dans C
MEM10  A=A-1  A        ) A => R2 = compteur
       AR2EX           ) A[A] = DIR START
       ABEX   A        A[A]=Maxmem après format.
       R3=A            R3=compteur mem :tape
       ?ST=1  Flg      Est-on entré via MEMM
       GOYES  MEM20
       CSL    A        ) Non: division de C[A] par
       CSRB            ) 8 pour obtenir le max dir
       R3=C            ) R3 compteur mem dir
```

```
MEM20  GOSUB  GET#FL   Routine commune de comptage
       A=R3           A= résulat GET#FL
       ?ST=1  Flg     Est-on entré via MEMM ?
       RTNYES         Retour de MEMM
       GOTO   SORTIE  Retour de MEMDIR

GET#FL A=R1
       D1=A           D1^ MS-512
       A=B    A       A[A] = # enregis. à lire
       GOSUB  JUMPER
       CON(5) READR#
       GOC    ERReur
       B=B+1  A       B[A] = prochain # enr.
       A=R1
       D1=A           D1 ^ MS - 512
       GOSUB  GET#E   Routine de comptage des 8
        *             entrées directory d'un enr.
       A=R2           ) Décrémentation de R2
       A=A-1  A       ) # d'enregistrements du
       RTNC           ) dir & retour si fin
       R2=A           Autrement sauvegarde
       GONC   GET#FL  B.E.T

ERReur GOTO   erreur

GET#E  LC(5)  #FFFF   C[3:0]=FFFF, masque de com-
        *             paraison du rideau de fin d
        *             de directory
       P=     8       8 = 8 entrées / enregist.
       A=0    A
get#e  P=P-1          Compteur
       RTNC           Sortie si fin
       D1=D1+ 16
       D1=D1+ 4       D1^ type de fichier
       A=DAT1 4       A[3:0] =type de fichier
       ?A=C   A       Est-ce le rideau (FFFF)
       GOYES  R3=OK   Oui: fin du s.prog.
       ?A#0   A       A=0 => fichier purgé !
       GOYES  R3-1
incré  D1=D1+ 12      ) On passe à l'entrée
       D1=D1+ 16      ) suivante
MEM30  D1=D1+ 16      )
       GONC   get#e   B.E.T

R3-1   ?ST=1  Flg     MEMM ??
       GOYES  R3-M    Oui => R3- long fichier
       A=R3           )autrement R3
       A=A-1  A       ) -1 seulement pour
       R3=A           ) MEMDIR
       GONC   incré   B.E.T.

R3-M   D1=D1+ 12      D1^ long. du fich. en #rec.
       C=P    4       Sauveg. pointeur # dir
       RSTK=C
       P=     4
       GOSUB  GETALR  A[A] long. # rec. du fich.
       C=RSTK
       CPEX   4       Restitution
       CR3EX          )
       C=C-A  A       ) R3 = R3 - # rec
       CR3EX          )
       D1=D1+ 8       Ajustement
       GONC   MEM30   B.E.T

R3=OK  C=RSTK         Fin pop et retour
       RTN

************************************************
* Cette routine sauvegarde 5 niveaux de pile
************************************************

r<rstk P=     4
       GOVLNG R<RSTK   I N D I P E N S A B L E !!

************************************************
* Cette routine a été pompée dans les IDS V, elle
* n'est pas supportée. Elle permet la sortie
* de l'erreur Aborted lorsque l'on arrête l'exécu-
* tion des routines HPIL en appuyant deux fois sur
* la touche ATTN.
************************************************

atnchk ?ST=0  Attn
       GOYES  ATNCHc
       RSTK=C
       CD0EX
       D0=(5) ATNFLG
       C=DAT0 S
       D0=C
       C=RSTK
       ?C=0   S
       GOYES  ATNCHc
       C=C+1  S
       GOC    ATNCHc
       P=     eABORT
       RTNSC

ATNCHc RTNCC

************************************************
* Routine de calcul de la longueur maxi
* du support (multiplication)
************************************************

getA*  R2=A
       P=     4
       GOSUB  GETALR
       C=R2
       GOSBVL A-MULT
       RTN
```

```
*************************************************
* GETALR: GET A Left-Right.
* Cette routine permet de charger dans le registre
* A n octets. Les octets sont décalés vers la gau-
* che à chaque nouvel arrivant.
*
* Entrée: D1^ premier octet à charger dans A
*         P= Nombre d'octets à charger
*
* Sortie: A[2*(Pentrée):0] = octets, le dernier
*           étant toujours dans A[B].
*         P=0
*         D1 ^ après dernier octet
*
* Utilise : C[B], A[W], P , D1
*
*************************************************

GETALR C=DAT1 B
       D1=D1+ 2
       ASL    W
       ASL    W
       A=C    B
       P=P-1
       ?P#    0
       GOYES  GETALR
       RTN

*************************************************
* ROUTINE JUMPER
* Cette routine permet l'accès aux points d'entrée
* supportés du module HPIL. Elle peut en réalité
* servir à n'importe quelque fichier Lex.
*************************************************

JUMPER RSTK=C
       CD1EX
       D1=(5) SNAPBF
       DAT1=C A
       D1=(2) (SNAPBF)+5
       C=RSTK
       DAT1=C W
       D1=(4) (SNAPBF)+21
       DAT1=A W
       D1=(2) (SNAPBF)+37
       C=B    A
       CPEX   5
       P=     6
       C=0    P
       GONC   JUMP05
       C=C-1  P
JUMP05 P=     7
       C=0    P
       C=C-1  P
       DAT1=C 8
       SETHEX
       P=     0
       LC(3)  bLEX
       GOSBVL I/OFND
       GONC   JUMP90
       LC(2)  LEXPIL
       B=C    A
       A=0    A
       A=A+1  A
JUMP10 C=DAT1 6
       ?C=0   B
       GOYES  JUMP90
       ?B#C   B
       GOYES  JUMP20
       CSR    W
       CSR    A
       ?A<C   B
       GOYES  JUMP20
       CSR    A
       CSR    A
       C=C-A  B
       GONC   JUMP30
JUMP20 D1=D1+ 11
       GONC   JUMP10
JUMP90 LC(4)  eXWORD
       GOTO   bserr
JUMP30 D1=D1+ 6
       C=DAT1 A
       B=C    A
       C=RSTK
       D1=C
       D1=D1+ 5
       CD1EX
       RSTK=C
       C=DAT1 A
       C=C+B  A
       RSTK=C
       D1=(5) (SNAPBF)+21
       A=DAT1 W
       D1=D1+ 16
       C=DAT1 8
       B=C    A
       P=     7
       C=C+1  P
       GOC    JUMP50
       SETDEC
JUMP50 P=     6
       ?C#0   P
       GOYES  JUMP60
JUMP60 P=C    5
       D1=(4) (SNAPBF)+5
       C=DAT1 W
       D1=(2) SNAPBF
       RSTK=C
```

```
C=DAT1 A
D1=C
C=RSTK
RTN

END
```

---

## LA STRUCTURE FOR STEP EXIT NEXT

JANICK TAILLANDIER (SIG#7 P#253 CHHU#202)

La fonction 'EXIT' présentée ici a pour but essentiel d'améliorer la lisibilité de vos programmes. Elle permet de sortir proprement d'une boucle avant la fin théorique de celle-ci telle que définie par l'ordre FOR.
La syntaxe est la suivante:

```
EXIT variable
```

La variable en question est l'indice de boucle utilisé dans les ordres FOR et NEXT. Lors de l'exécution de cet ordre il y a débranchement à l'instruction qui suit l'ordre NEXT de même variable.

Ceci sera plus clair avec un exemple:

```
FOR I=1 TO 10
      .
      .
FOR J=1 TO 10
      .
      .
IF ... THEN EXIT I
      .
      .
NEXT J
      .
NEXT I          <= après EXIT l'execution
                   se poursuit ici
```

Le code proprement dit est essentiellement le résultat d'une paraphrase du code de FOR et NEXT généreusement fournit par HP.
Le cas de EXIT exécuté depuis le clavier n'est pas traité car un tel ordre n'a d'intérêt que dans le cadre d'un programme assez compliqué et, par ailleurs, le traitement de cette condition complique le code.

```
      LEX   'EXITLEX'
      ID    #E1
      MSG   0
      POLL  0
      ENTRY exit
      CHAR  12        non utilisable à partir
       *              du clavier
      KEY   'EXIT'    Syntaxe: EXIT N
      TOKEN 61
      ENDTXT

NXTDC  EQU   #0528A
NXTP   EQU   #03455
RANGE  EQU   #1B07C
AVMEME EQU   #2F599
FORSTK EQU   #2F59E
RUNRTN EQU   #074EA
TKSCN7 EQU   #08A99
eFwoNX EQU   #2A
MFERR  EQU   #09393
tNEXT  EQU   #C4
PRGMEN EQU   #2F567

EXITd  GOVLNG =NXTDC
EXITp  GOVLNG =NXTP

************************************************
*
* structure de la pile FOR-NEXT
*
* Low mem.
*
*   5 quartets   adresse de retour (DO)
*  16 quartets   incrément
*  16 quartets   limite (TO)
*   2 quartets   ASCII letter or alpha-digit
*   2 quartets   00 or ASCII letter
*
* High mem.
*
************************************************

      REL(5) EXITd
      REL(5) EXITp

* Le code suivant est tiré de la routine
* INDXSN point d'entrée non supporté

exit  ST=1  8       alpha-digit variable flag
      A=DAT0 4      lecture de l'index variable
```

```
       LCASC  'ZA'
       GOSBVL =RANGE
       P=     3
       GOC    indx02
       ST=0   8          Simple chr. alpha-digit
       B=0    A
       P=     1
indx02 B=A    WP
       P=     3
       D1=(5) =FORSTK
       C=DAT1 A
       R0=C              Sauveg. du haut de la pile
        *                FOR NEXT
       D1=D1+ 5
       A=DAT1 A          ^ le bas de la pile F/NEXT
indx04 D1=C
       ?C>=A  A          Fin de recherche ?
       GOYES  indx07
       D1=D1+ 16
       D1=D1+ 16
       D1=D1+ 5
       C=DAT1 A          Lecture de la variable
       ?B=C   WP         ok ?
       GOYES  indx08
       D1=D1+ 4
       CD1EX
       GONC   indx04     B.E.T.
indx07 P=P-1             =>  carry=0
indx08 P=     0

* Fin de INDXSN

       GONC   efwo
       R0=C              sauvegarde de la variable
       D1=D1+ 4          D1 @ bas de entry
       AD1EX             Efface toutes les entrées
       D1=(5) =AVMEME    précédentes
       DAT1=A A          .
       D1=D1+ (=FORSTK)-(AVMEME)
       DAT1=A A          .

       D1=(5) =PRGMEN    D[A] prêt pour  TKSCN7
       C=DAT1 A          .
       D=C    A          .

       P=     3
       ?ST=1  8          variable alpha-digit ?
       GOYES  exit01
       P=     1
exit01 CPEX   15
       D0=D0- 4          pour tXWORD et lex ID
exit10 D0=D0- 4          D0 @ long. octets
       A=0    A
       A=DAT0 B          long. ligne
       B=A    A
       ADOEX
       A=A+B  A          pointe EOL ou @
       D0=A
       P=     0
       LC(2)  =tNEXT     . limite ?
       GOSBVL =TKSCN7
       GOC    exit02     token trouvé
efwo   LC(2)  =efwoNX
       GOVLNG MFERR
exit02 P=C    15
       A=R0              restaure nom variable EXIT
       D0=D0+ 2          ^ variable courante NEXT
       C=DAT0 WP
       ?A#C   WP
       GOYES  exit10     essai le prochain NEXT
       D0=D0- 4
       A=0    A
       A=DAT0 B          A[B]= long. stmt
       CD0EX
       A=A+C  A
       D0=A
       GOVLNG =RUNRTN

       END
```

---

## LEX PPOLL ET DEMANDES DE SERVICE SUR LA BOUCLE HPIL

Le module HPIL du HP71 fait de cette machine le plus puissant contrôleur de boucle HPIL actuellement disponible. Cependant, il y manquait quelques fonctions pour être réellement complet, justifiant ainsi l'écriture du lex PPOLL.

Avant d'en détailler les fonctions, je crois qu'il n'est pas inutile de rappeler les possibilités offertes par le protocole hpil pour gérer les demandes de services.

Comme chacun sait, les messages HPIL sont codés sur 11 bits: 3 bits de contrôle définissant le type de message ( donnée, commande, synchro, identification) suivis de 8 bits de données. Dans le cas des messages de données et d'identification, un bit de contrôle est disponible pour permettre aux périphériques de signaler une demande de service. Ansi lorsque le contrôleur communique sur la boucle, il peut se

rendre compte à chaque message de donnée de la présence d'une demande de service et agir en conséquence.

Un problème survient lorsque le contrôleur est inactif et ne communique pas sur la boucle : comment se rendre compte de la présence d'une demande de service ? Deux solutions sont possibles pour le contrôleur :

* générer périodiquement des messages d'identification IDY. Ceux-ci lui signaleront la présence d'une demande de service. Les calculateurs HP de la série 80, l'interface HPIB/HPIL, ainsi que la HP41 (lorsqu'elle a trouvé une imprimante) utilisent ce mode. Le HP71 en est capable, mais il est plus astucieux d'utiliser la seconde solution :

* mettre la boucle dans un mode particulier dans lequel les périphériques peuvent générer leur propre message d'identification de façon asynchrone lorsqu'ils veulent signaler une demande de service. Le contrôleur envoie la commande EAR (Enable Asynchronous Request) pour valider ce mode. Lorsqu'il veut reprendre la direction des opérations, le contrôleur doit s'assurer que la boucle est revenue en mode synchrone, ceci se fait généralement par l'envoi d'une série de messages NOP.

Le HP71 gère le mode asynchrone, bien que ce point soit peu décrit dans la documentation. Voici un exemple :

```
          ON INTR GOTO 'SERVICE'
'DORS':   ENABLE INTR 8 ! valide intr sur SRQ
          SEND CMD 24 ! envoie EAR
          OFF
          STOP
'SERVICE':DISP "SRQ reçue !" @ BEEP
          ! traitement de la demande de service
          GOTO 'DORS'
```

Le HP71 se réveille à la réception d'une demande de service (par exemple lorsqu'un modem a détecté un appel), la traite, puis se rendort.

A ce point, le contrôleur a la possibilité de prendre connaissance de la présence d'une demande de service. Si plusieurs périphériques sont susceptibles de demander un service, le contrôleur doit tout d'abord identifier le demandeur. Il peut le faire en interrogeant successivement l'octet d'état de chaque périphérique : c'est le 'serial polling' (SPOLL), ou il peut interroger simultanément jusqu'à 8 demandeurs potentiels : c'est le 'parallel polling' (PPOLL).

L'interrogation parallèle se fait en envoyant un message d'identification IDY 0. Les périphériques demandeurs peuvent, en plus de lever le bit de demande de service, modifier un des 8 bits de données disponibles. La mise en oeuvre de l'interrogation parallèle se fait donc en assignant chacun des bits à un ou plusieurs périphériques grâce aux commandes PPE (Parallel Poll Enable).

DESCRIPTION DU LEX PPOLL

- PPOLL

- syntaxe : 100 P=PPOLL
100 IF BIT(PPOLL,3) THEN 'DEVICE3'
- messages envoyés : IDY 0
- retourne la réponse à une interrogation parallèle sous la forme d'un octet (0-255).
Les périphériques doivent avoir été configurés par les commandes PPE (commandes 128 à 143)

Exemple : SEND UNL LISTEN 1 CMD 136

SEND UNL LISTEN 3 CMD 137

assigne le bit 0 (PPE1,0) au périphérique 1 et le bit 1 (PPE1,0) au périphérique 3.

- SRQ

- syntaxe : 100 S=SRQ
100 IF SRQ THEN 'SERVICE'
ELSE 'ATTEND'
- messages envoyés : IDY 0
- retourne 0 si aucun périphérique ne demande de service
- retourne 1 si au moins un périph. demande un service

- NLOOP

- syntaxe : 100 N=NLOOP
100 FOR I=1 TO NLOOP
- messages envoyés : aucun
- retourne le nombre de périphériques sur la boucle
- Le résultat n'est valide que si le 71 est en mode d'adressage simple (flag -22 baissé).
Cette fonction n'envoie aucun message, elle ne

fait que retourner la valeur conservée par l'interface. Cette valeur est actualisée lors de l'auto-adressage de la boucle, par exemple lors d'un RESTORE IO.

Exemple :

```
RESTORE IO
DISP "Device Aid   Id#"
FOR I=1 TO NLOOP
DISP USING "2(3D,4X),K";I,DEVAID(I),DEVID$(I)
NEXT I
```

- SLEEP

- syntaxe : 100 SLEEP
    - met le 71 en sommeil léger, diminuant la consommation dans un facteur de l'ordre de 50.

Les causes provoquant le réveil du 71, et la poursuite du programme sont :

* les 3 timers hardware (différents des 3 timers software ON TIMER# qui se partagent en fait le même timer hardware)
* le HPIL
* le clavier
* autres ...

Le 71 utilise les timers hardware pour ses propres besoins. Toutes les minutes, un des timers interrompt le 71 (et éventuellement le réveille) pour lui faire penser à vérifier le bon état des batteries. De même, la gestion du clignotement du curseur est assurée par des interruptions timers.
CES INTERRUPTIONS REVEILLENT LE 71 ET PROVOQUENT DONC LA SORTIE DE L'INSTRUCTION SLEEP !

Un programme utilisant l'ordre SLEEP se présente sous la forme :

```
ON TIMER #1,T1 GOSUB 'IRQTIM' ! interruption timer
ON INTR GOSUB 'IRQIL' ! interruption hpil
ENABLE INTR M ! déf. le masque d'interruption hpil
              ! éventuellement, envoyer EAR si le
              ! 71 est contrôleur
              ! et veut surveiller les demandes
              ! de services
'DORS': SLEEP
K$=KEY$
IF K$="" THEN 'DORS'
              ! traitement du clavier
GOTO 'DORS'
              !
'IRQTIM':     ! traitement du timer
RETURN
              !
'IRQIL':      ! traitement du HPIL
ENABLE INTR M
RETURN
```

La mise en sommeil léger par SLEEP a par rapport à la mise en sommeil profond (OFF) les avantages suivants :

* possibilité de gérer le clavier
* possibilité d'afficher un message (ou un menu)
* grande vitesse de réponse aux demandes de services hpil et aux timers.

Exemple d'utilisation de l'ordre SLEEP :

Le programme suivant affiche l'heure et la date à la façon de la fonction CLOCK de la HP41. Si vous assignez 'CALL CLOCK' à la touche gATTN par
DEF KEY '#155','CALL CLOCK':
la similitude sera (presque) parfaite. La pression d'une touche quelconque (y compris ATTN) provoque le retour du HP71 à son contexte antérieur.

```
10 ! --- CLOCK ---
20 POKE "2F441","F"
30 ON TIMER #1,1 GOTO 40
40 DISP CHR$(13);DATE$;"      ";TIME$;
50 SLEEP
60 K$=KEY$ @ IF K$="" THEN 50 ELSE PUT K$
70 OFF TIMER #1
80 DISP
90 POKE "2F441","0"
```

Note : les trois fonctions acceptent un numéro de boucle optionnel.
ex: SRQ(2), PPOLL(3)

Jean-François GARNIER

```
      LEX    'PPOLL'
      ID     #E1
```

```
        MSG     0
        POLL    0
        ENTRY   NLOOP
        CHAR    #F
        ENTRY   PPOLL
        CHAR    #F
        ENTRY   Sleep
        CHAR    #D
        ENTRY   SRQ
        CHAR    #F
        KEY     'NLOOP'
        TOKEN   62
        KEY     'PPOLL'
        TOKEN   63
        KEY     'SLEEP'
        TOKEN   64
        KEY     'SRQ'
        TOKEN   65
        ENDTXT

eXWORD  EQU     #23
bLEX    EQU     #BFC
LEXPIL  EQU     #FF
BSERR   EQU     #0939A
FNRTN1  EQU     #0F216
I/OFND  EQU     #118BA
HDFLT   EQU     #1B31B
SNAPBF  EQU     #2F7F0
FUNCD0  EQU     #2F8BB
GETLPs  EQU     #1D15
PUTC    EQU     #6B1C
GET     EQU     #6751
ERRORX  EQU     #342C
OUTELA  EQU     #05303
NXTSTM  EQU     #08A48
SLEEP   EQU     #006C2
CKSREQ  EQU     #00721

        NIBHEX 801
NLOOP   GOSUB   JUMPER   cherche la boite
        CON(5)  GETLPs   au lettre hpil
        GOC     err      erreur ?
        LC(4)   #0100    demande le
        GOSUB   SEND     nombre de périphériques
        A=C     A        résultat
        GONC    SRQ1     B.E.T.

        NIBHEX 801
PPOLL   GOSUB   SIDY     envoie IDY 0
        A=C     B        résultat du ppoll
        GONC    SRQ1     B.E.T.

        NIBHEX 801
SRQ     GOSUB   SIDY     envoie IDY 0
        CSR     A
        A=C     P        (P=0)
        LC(1)   1
        A=A&C   P        isole bit SRQ
SRQ1    GOSBVL  HDFLT    convertion en  flottant
        D0=(5)  FUNCD0
        C=DAT0  A
        CD0EX            restaure D0
        C=A     W        résultat
        GOVLNG  FNRTN1   fin

SIDY    GOSUB   JUMPER   cherche la boite
        CON(5)  GETLPs   au lettre hpil
        GOC     err      erreur?
        LC(4)   #1E00
SEND    GOSUB   JUMPER   envoie la commande
        CON(5)  PUTC     au processeur d'e/s
        GOC     err      erreur?
        GOSUB   JUMPER   recoie la
        CON(5)  GET      réponse
        GOC     err      erreur?
        A=0     A        initialise A pour
        RTNCC            la suite

err     GOSUB   JUMPER
        CON(5)  ERRORX

 * ERRORX n'est pas une routine supportée
 * mais est valable pour les ROM versions A & B

************************************************
* NOTE DE LA REDACTION : La routine ERRORX est
* recopiée dans le Lex MMLEX de Michel MARTINET
* sous le label [erreur]. Dans l'avenir, il nous
* semblerait judicieux de rassembler tous les Lex
* traitant de la boucle d'interface dans un seul
* pour éviter d'être obligé de recopier à chaque
* fois le traitement des erreurs et la routine
* JUMPER (accès au routines internes de la ROM
* HPIL).
************************************************

        REL(5)  Sleepd
        REL(5)  Sleepp
Sleep   CD0EX            sauve D0
        R0=C             et D1 en
        CD1EX            R0 et R1
        R1=C
        GOSBVL  SLEEP    en sommeil léger
        GOSBVL  CKSREQ   traite les SRQ
        C=R1
        D1=C             restaure D1
        C=R0               et
        D0=C               D0
        GOVLNG  NXTSTM
```

```
Sleepp RTNCC           parse: rien
Sleepd GOVLNG OUTELA   décompile: rien

JUMPER RSTK=C
       CD1EX
       D1=(5) SNAPBF
       DAT1=C A
       D1=(2) (SNAPBF)+5
       C=RSTK
       DAT1=C W
       D1=(4) (SNAPBF)+21
       DAT1=A W
       D1=(2) (SNAPBF)+37
       C=B    A
       CPEX   5
       P=     6
       C=0    P
       GONC   JUMP05
       C=C-1  P
JUMP05 P=     7
       C=0    P
       C=C-1  P
       DAT1=C 8
       SETHEX
       P=     0
       LC(3)  bLEX
       GOSBVL I/OFND
       GONC   JUMP90
       LC(2)  LEXPIL
       B=C    A
       A=0    A
       A=A+1  A
JUMP10 C=DAT1 6
       ?C=0   B
       GOYES  JUMP90
       ?B#C   B
       GOYES  JUMP20
       CSR    W
       CSR    A
       ?A<C   B
       GOYES  JUMP20
       CSR    A
       CSR    A
       C=C-A  B
       GONC   JUMP30
JUMP20 D1=D1+ 11
       GONC   JUMP10
JUMP90 LC(4)  eXWORD
       GOVLNG BSERR
JUMP30 D1=D1+ 6
       C=DAT1 A
       B=C    A
       C=RSTK
       D1=C
       D1=D1+ 5
       CD1EX
       RSTK=C
       C=DAT1 A
       C=C+B  A
       RSTK=C
       D1=(5) (SNAPBF)+21
       A=DAT1 W
       D1=D1+ 16
       C=DAT1 8
       B=C    A
       P=     7
       C=C+1  P
       GOC    JUMP50
       SETDEC
JUMP50 P=     6
       ?C#0   P
       GOYES  JUMP60
JUMP60 P=C    5
       D1=(4) (SNAPBF)+5
       C=DAT1 W
       D1=(2) SNAPBF
       RSTK=C
       C=DAT1 A
       D1=C
       C=RSTK
       RTN

       END
```

## LE COIN DES LHEX

Au programme en ce début d'année, six LHEX:

| | | |
|---|---|---|
| KEYWAIT | KEYWAIT$ | 82001 |
| EDLEX | DELETE# | 240001 |
| | EDPARSE$ | 240007 |
| | EDTEXT | 240002 |
| | FILESZR | 240003 |
| | INSERT# | 240004 |
| | REPLACE# | 240005 |
| | SEARCH | 240006 |
| | SCROLL | 82002 |
| | MSG$ | 82003 |
| MMLEX | MAXD | 225057 |
| | MAXM | 225058 |
| | MEMD | 225059 |
| | MEMM | 225060 |
| ASFLEX | (POLL pKYDF) | |
| EXITLEX | EXIT | 225061 |
| PPOLL | NLOOP | 225062 |
| | PPOLL | 225063 |
| | SLEEP | 225064 |
| | SRQ | 225065 |

BONNE ANNEE
Heureuse programmation à tous
Michel MARTINET

PROGRAMME MAKELEX

```
 10 CALL MLEX
  @ SUB MLEX @ SFLAG -1 @ PURGE AH @ INPUT "Nb. d'octets: ";N @ LC OFF
 20 CREATE DATA AH,1,N-4 @ A=HTD(ADDR$("AH")) @ B=A @ GOSUB 130
 30 Q=1 @ X=0 @ INPUT "000: ",P$;A$ @ C$=A$ @ S=0 @ GOSUB 90
 40 Q=2 @ X=1 @ GOSUB 80 @ A$=A$&C$ @ A=A+37 @ N=N*2+37
  @ Q=3 @ SFLAG 5 @ FOR X=2 TO N DIV 16-1
 50 GOSUB 80 @ C$=C$[5*FLAG(5)+1] @ POKE DTH$(A),C$
  @ A=A+16-5*FLAG(5,0) @ NEXT X @ Q=4
 60 DISP DTH$(X)[3]; @ INPUT ": ",P$[1,MOD(N,16)];C$ @ GOSUB 90
 70 POKE DTH$(A),C$ @ POKE DTH$(B),A$ @ CFLAG -1 @ END
 80 DISP DTH$(X)[3]; @ INPUT ": ",P$;C$
 90 DISP DTH$(X)[3]; @ INPUT " sm ","---";D$
100 M=S @ FOR Z=1 TO LEN(C$) @ M=NUM(C$[Z])+M+1 @ NEXT Z
110 IF D$=DTH$(MOD(M,4096))[3] THEN GOSUB 130 @ S=M @ RETURN
120 DISP "Erreur de somme" @ BEEP @ P$=C$ @ POP @ ON Q GOTO 30,40,50,60
130 P$="----------------" @ RETURN
```

```
KEYWAIT   ID#52  55 octets

     0123456789ABCDEF sm

000: B454957514944502 366
001: 802E001041111058 6AB
002: 3700025101000000 9CE
003: F710000000000000 CFC
004: 0E1000FFB4549575 087
005: 14944542101FF001 3E7
006: 361081371098F2C6 757
007: 0045111913511813 A93
008: 48D8ACA18F127006 E2F
009: BDF              EFE

EDLEX     ID#F0  2557 octets

     0123456789ABCDEF sm

000: 5444C45485020202 354
001: 802E002041111058 69A
002: FF3100F107031310 A00
003: FD400CB00F910000 D83
004: 04A200D220BE100D 0FE
005: 130CE100F240D920 476
006: 0D150EB200D26076 7F2
007: 800FF003FE00FB44 B94
008: 54C454455410F544 F03
009: 4051425355442708 258
00A: 5444455485420D6 5BD
00B: 494C45435A52530B 938
00C: 94E4355425454OD2 CA9
00D: 55405C414345450B 00F
00E: 355414253484601F 36C
00F: F10B001504C496E6 6F0
010: 5602CC010254F666 A67
011: CF120BB344454648 DF1
012: 494C4D405253545C 171
013: D0301F302D60C314 4E9
014: 0EAE4E616D656A3C 8A6
015: E060334D646A3CB2 C3A
016: 70AF4B40247F6024 FC1
017: 456C65647656F302 33B
018: 95F2E4A3CC080295 6DC
019: E415C9190ECE734D A90
01A: 6460235472776CA1 E00
01B: A0975F627B696E67 1A4
01C: 6E2E2E2CE0B0344F 55A
01D: 6E656CFF96950503 8FA
01E: 11B1351121CB1378 C64
01F: B6A113510B3B14A3 FE8
020: 4544540215DB0031 344
021: E296140003410000 687
022: 8A240006E8224500 9E6
023: 393008DEAD814111 D6B
024: 36068FDD41074B08 0FA
025: F59B904C27FC18F8 4B9
026: 3DB0137C21357390 82E
027: 33A300854F2F6DA6 BC8
028: C207C708F36F9047 F5F
029: ED0CC10113374358 2E2
02A: 448AE1D110E47C71 67A
02B: 8FB13B186450BCC0 A20
02C: 41CF15178F8A4100 DA8
02D: 71368DC32F08B400 131
02E: 554008488596D304 4A4
02F: A4004440085B8496 80F
030: 9208D8BB818D10FA BBF
031: 170F16301A7400A1 F26
032: 40084B8498F53410 298
033: 1F678F28FF237178 63C
034: CF7241175D074467 9C0
035: D655DB1F678F28F3 D7F
036: F2718793710417 0F6
037: DA08F13DB0AF1D88 4BF
038: 1D1371F995F2CA14 85E
039: 1134183D48F42A71 BD8
03A: 1583E581DE5C5C51 F7B
03B: C4143C0141E01318 2E5
03C: F261B11F178F2143 66E
03D: 1748F3F2711011FF 9FA
03E: 69F214B77B35018A D8A
03F: E823363005F1AC48 114
040: F08371421A4F462A 49C
041: 4F40278E38DA9390 835
042: 16173EE871F08DE4 BDC
043: 591878DB13706132 F4F
044: 86B40DAEE10B1F69 30B
045: 8F214713517C1477 67E
046: 7AE7F3105B0D507D A27
047: 947A1FF69F21438F DD6
048: 7541117F17414307 131
049: E217F17214517514 49C
04A: 7113CA1418F53410 80D
04B: 8C66F0AF2AF0E610 BC6
04C: 936575840110B14A F2F
04D: 311F966C3160142F 2A9
04E: 416410114A311F96 60A
04F: 6C116115A3103119 965
050: 8BE0133B00002111 CC2
051: 1031117527101113 FF5
052: 7B171031337A7010 351
053: A7CD257082103173 6C7
054: 13312311A74C2137 A22
055: 1354607A431B688F DA3
056: 2144DB068FE3F801 13F
057: 1B1FF85F21451350 4C7
058: 7D77A13108DB068F 868
059: 71B608FDA6701351 BF2
05A: 731375CC8FD1F60D FA5
05B: 7174137111018D20 2F5
05C: F203F4C544445548 678
05D: 5453F8F124501361 9E4
05E: BB98F2144164133I D5D
05F: 401341318F5CC304 0CE
060: CB8FE7A204607801 46A
061: 1BB98F2146164142 7E5
062: 13413174308FDC63 B5E
063: 05C0859732056078 EC2
064: 00318F68A078A079 243
065: 907590311E64908F 5BC
066: 625308F728408F9D 94A
067: F30873911FB49F21 CDE
068: 5B090C907A6059C8 071
069: F740301347E408F4 3EF
06A: 60307D7045631229 74A
06B: 62C5307962458FFD AE6
06C: C2014B7B50491879 E66
06D: BD3102962B031C29 1E7
06E: 669C730053A31228 552
06F: D8EC208D271308DD 901
070: 96308548DB2E2085 C88
071: 48DF3E208FEA2304 035
072: 0054E31D0962008D 3A8
073: 29A208588F653304 71E
074: 4C17174BF7BCF7CA AEE
075: F8782A312F966AA8 E99
076: FBEC208DD973017F 25C
077: 8F2295017131C27D 5D9
078: 6F8F229501738D00 95C
079: 3503132745F8D394 CCF
07A: 508F754111781534 031
07B: AC88FE7371400172 3C0
07C: 1574AC7A46463170 73C
07D: 14313117FD215F3C ABF
07E: E8AE521737BBD722 E6E
07F: 0500D28680031630 1BC
080: 233C3000233F3000 50B
081: 2CC101AF0CC10084 89F
082: 7848712040013411 BE2
083: 0E41001198A27587 F4D
084: 8001365DDD11358B 2D4
085: F44173137887E315 657
086: B38F42A7123A98D9 9FD
087: B1520400DD81EE6C D9F
088: 6C6133131C28BB50 120
089: 8580385702CD8580 49A
08A: 21C3137847848789 80D
08B: F40017303760017F B84
08C: 8F322B1400664A88 F01
08D: 884551368EB58077 283
08E: DF10070DF190A140 60F
08F: 75CF16414010178B 987
090: FCC540D084516414 D11
091: 0CC45085513710A1 080
092: 641F698F21471441 3F6
093: 107C6E4A31BFA8F2 7AE
094: 1461969144DB19FA B42
095: 14411A1371641447 E94
096: 84148273A05F26BA 220
097: A8AE9F11A1358FBD 5EE
098: 3B1133846102AF26 95F
099: 2F08DDF8E0184142 CFE
09A: 8B29E16E146D7164 095
09B: 146729E44D8780B1 41D
09C: 44173137164144D7 77A
09D: C9C910B18E865E01 B1C
09E: 468BD81C3C318E14 EBC
09F: 6135D57B215E1845 23C
0A0: 7900E6144609F1B5 5BD
```

```
0A1: A8F214601AF27DEF 979
0A2: 7B440410B199BAF2 D0F
0A3: 146DF111EFEE81E7 0D1
0A4: C2404DF81EE67F14 483
0A5: 11BBF2BF2BF2AB6B 860
0A6: F2BF2BF232800ABF C22
0A7: 2EBF2A3F90A7FABB 009
0A8: 0411286660E4E413 371
0A9: 11CF15571B0C8F21 705
0AA: 421308CD28F8468F A9E
0AB: 13DB01371351BB98 E22
0AC: F2144C210A8A8001 19C
0AD: 3418114A31C5966F 51A
0AE: 21811331311368BE 87B
0AF: 0013414A31C5962F BEC
0B0: 011ACECE10A8561B F97
0B1: 495F2142D23151CA 31C
0B2: 1000384B8407B713 67E
0B3: 1C5966B316114A31 9F3
0B4: 4296613161132 11A D43
0B5: 9663211B86B708A7 0C8
0B6: 00D710903793131C 42B
0B7: 596644D413016114 786
0B8: A31C5962E4E5E572 B2A
0B9: 218606487BE17A01 EA5
0BA: 31E5966A18750085 21B
0BB: BE5E55268600279E 5B3
0BC: 031E296205304962 90C
0BD: 843104962F37BC03 C89
0BE: 1C59622311A8B580 002
0BF: DB617F11B8BB00DB 3C2
0C0: 13414E962F087B00 73C
0C1: E7E761EF85777204 ADE
0C2: 6187B00E7E711B8B E7C
0C3: B00548038DD44908 1FB
0C4: D2D2B11371351108 568
0C5: B66E1C41191451C4 8F0
0C6: D91451C4DB145877 C85
0C7: B0119D7CFCF8471C 046
0C8: 4071451C00915501 394
0C9: 1B10911A8B582DB1 724
0CA: 0963B0E5E5D41301 AA9
0CB: 4A03870708500184 E01
0CC: 00373EF31C596632 17F
0CD: 72DF11A8B5A0CFCF 557
0CE: 6C6031C59626073C 8D6
0CF: F8609331E2966E01 C5C
0D0: 198B7C3623031049 FC9
0D1: 666068A03142966E 33E
0D2: 01198B7316320DB1 6B0
0D3: 3414E962908496EF A41
0D4: 0E5E511A8B521DBE DF5
0D5: 6E610985962E0E7E 194
0D6: 71198BB606B2F793 52B
0D7: FE5E5860A2310496 8BD
0D8: 6D011A8B9DC51E31 C63
0D9: 429665A11A8B99B5 FFA
0DA: A931C5966198505E 386
0DB: BE7E71198BB27E5E 74F
0DC: 511A8B9F88604378 ADC
0DD: DE31049665051E31 E56
0DE: E2962CC314296661 1DA
0DF: 11ACECE8B5A0695F 5A7
0E0: 76AE31C596262119 92A
0E1: 13418114E9625111 C7F
0E2: 9CECE1098B75D6C0 045
0E3: F711E58E6D1F0915 3EB
0E4: 700A170147061741 73A
0E5: 47D7174147D51741 AAE
0E6: 4717487900109038 E02
0E7: D4BCE084BD113713 1A6
0E8: 58BB001C114B3102 519
0E9: 9628E87BD030C85B 8C1
0EA: 9628DE58F670B15E C70
0EB: 2E531E2962003039 FD7
0EC: 6200E530B96200E5 349
0ED: 31F396200E502727 6B3
0EE: 084AAE6AF5137135 A4C
0EF: 8BBF31C114B8F670 DF8
0F0: B14C281181190980 16C
0F1: 7B00AF5D4AE84DC3 535
0F2: B939393939393021 8A2
0F3: 717D10AF49765085 C29
0F4: A7ADF9FA50AFAD16 014
0F5: 98F3F03030303030 373
0F6: 3030303411361B6 6B8
0F7: B8F2144137108332 A29
0F8: 00F1088FA0521FE1 DBB
0F9: B109F2779011B135 133
0FA: 8F83DB0137134D68 4C7
0FB: F992101EBB8F1411 861
0FC: 741411318F771B11 BC8
0FD: 361B995F2144737F F53
0FE: 10910A10BAC23302 2C0
0FF: 021088F064A10885 62A
100: B747E87AC17CA2E8 9F1
101: 04307205802019C0 D3D
102: 0C0C0C11880FF667 0D7
103: 01188DBA990741FA 475
104: E6AFA101850772E8 814
105: 7A0D726289147076 B8B
106: 0B308B0F7087A3B1 F1D
107: 038527DFD909C130 2AD
108: 590581874317E107 614
109: 3ED90D6067511182 992
10A: F307108206E91D88 D16
10B: 11811110AE8D4100 07B
10C: 854037A8EAE6AFA1 439
10D: 028517D9D304901D 7B6
10E: 39858B8FAA2511B1 B55
10F: 09F2D831142B14A9 EDE
110: 6045E61610D50F11 251
111: 8AE92F3044A87D8F 60D
112: 725D305901CB6F6F 9AE
113: 1371358551B0C8F2 D2A
114: 1440177EF1841461 096
115: 356D90110AEA1002 40D
116: 031A4962DDE69626 7A1
117: D710D7141770EEED B4F
118: 1FA1F71F30017111 ECA
119: 03124966DA729F1C 25A
11A: 11371358B3D214B3 5CC
11B: 10296221308962A0 92A
11C: 30C9668D17113713 C99
11D: 51841447B9C30390 008
11E: 56062AE909606FAE 3AB
11F: 7A10634140733030 6F6
120: 6300400040406307 A2B
121: 110D1A88C5C91341 DB0
122: 4A090E66968D1D28 149
123: 22E681C8328F1108 4C9
124: 16ACA1008F156811 84A
125: 291CB15DB11A1CB1 BF1
126: 5DB11B1CB15DB118 F9C
127: 1C114DF6F61C114D 340
128: 3038121C114D1BBB 6C6
129: 8F21461641428F40 A35
12A: 1B11B6B8F2146134 DB7
12B: 8408D7B181D98D62 153
12C: 4208118E7C6F8EEE 50D
12D: 5F570831E2D01000 87C
12E: 417F8F625308FA05 C10
12F: 21FE8FC2B90723D8 FC4
130: D608908FFACE0346 379
131: 52009BE3C100BF4F 719
132: 4F48BE4B8F943B11 ACE
133: 18AD2CA740940A25 E62
134: 203000000F020000 18F
135: 0000000B00D3000D 4DC
136: 00036FFFF7D43574 87F
137: 4230B353425F4C4C C04
138: 4201FF8DE6A20E04 FAA
139: FF4FFFF8ED6FE681 3B3
13A: F08F156811438E70 738
13B: 5FCC540D0D1AE81B AFC
13C: 374F214AD231F5B6 E95
13D: 29E140D9AEA34084 231
13E: F2C2C213414E96E9 5D2
13F: 0CC57ED01AC74F14 988
140: 81831460B8410B14 CED
141: 48FE21208D84A80  049

MMLEX     ID#E1  526 octets

                  0123456789ABCDEF sm

000: D4D4C45485020202 373
001: 802E006041111058 6BD
002: 024001E93C300000 A0B
003: F230000000000000 D36
004: 0CC000FB0044100F 0B3
005: 610F2100F120A110 40F
006: 0F7D4148544937D4 799
007: 1485D4A37D454D44 B2B
008: 4B37D454D4D4C31F ED9
009: F22D245272B20FB1 26A
00A: 049193BAE1371087 5E7
00B: B92C524050089153 951
00C: 8920388452708226 CAB
00D: B307F12451707219 014
```

```
00E: 760570884A02480C 383
00F: 02380C12231FFDAD 721
010: 6208DA9390C11136 A9A
011: 0677D17E7F703234 E1F
012: 840479AFB8FC2DE0 1E4
013: DAF081C73129D440 57D
014: 472248F8A4100713 8E7
015: 48FB13B1118135AF C7C
016: 68D832F0625F8DD4 028
017: 490C1184A85B531C 3B2
018: 1185A84B570C1108 729
019: 1360674517BFE110 AA0
01A: DB06D2320F1EA8F0 E52
01B: 64A1D68B33B10913 1D8
01C: 5D007D77381FF440 56A
01D: 47911187A13D2310 8CF
01E: 3C2135247A417131 C2F
01F: 7D2186B607800F0F FBE
020: 0653FD811113117F 334
021: 247021E810217724 689
022: 7211E887B40D6CC1 A27
023: 22DC10387BA0F281 DB9
024: E10B7C0011387B00 131
025: 66EE111131D47EE0 4C8
026: FF4404C1E5111131 846
027: 7210112CC4001025 B96
028: 4D683E34FFFF028D F61
029: 00D40017F17315B3 2CD
02A: 8A2648ACE017B17F 67B
02B: 17F5DD87BD0113CC A36
02C: 10356E17B80C4062 DAB
02D: 471600780F412BE2 122
02E: 12B17759C0701248 48D
02F: DDD41086C6206136 817
030: 1B244F2156413407 B7B
031: 94AC0B4646024020 EEA
032: 3102247C0011A8F9 25A
033: 43B10114F171BF0B 5E3
034: F0AEA0D880CE0106 993
035: 1371F0F7F21451D5 D1E
036: F0715571E508F151 09D
037: 71D51D980F526A82 432
038: 550A0E27A82A0E15 7C2
039: D7042032CFB8FAB8 B7C
03A: 1154331FFD5D0E41 F11
03B: 5F596A1296561BF6 2AD
03C: F69E2C0F6F6B6252 665
03D: 117A5AD333200652 9D1
03E: D175147D50713517 D3F
03F: 413706147C9061F5 0AE
040: 08F2153717F15F7D 443
041: 527B06440052690E 7AC
042: 2080D51E5F7F1577 B41
043: 1D0F061471350701 E9F
044: F                EE6

ASFLEX    ID#E1  160 octets

     0123456789ABCDEF sm

000: 143564C454850202 358
001: 802E007041111058 6A3
002: 541001E000000000 9D3
003: FE00000000800001F D2D
004: F31B196140001B14 096
005: 4F2156494AB41A7F 435
006: 6F1564A465B3110D 7BC
007: 231718A6F21A689F B50
008: 1520302D80EF290A ED0
009: 70D2450302A09154 22F
00A: 06EB0001A689F15A 5C2
00B: 03020E06908AE1A0 93B
00C: 84F32FB0AB514A96 CE7
00D: 8023102966E0161A 04A
00E: 3D58E4FB31A29626 3ED
00F: B1108FA0DE031139 77A
010: 663A1AE74F14A316 B12
011: 09EA4131D09EAB03 EB2
012: 1619E6FBE2E61A76 267
013: 9F14C16130415C07 5DD
014: 2100202020202020 8FC
015: 2020207160144850 C36
016: 82103            D39

EXITLEX   ID#E1  155 octets

     0123456789ABCDEF sm

000: 54859445C4548502 36B
001: 802E007041111058 6B6
002: A31001ED3D300000 A1F
003: F710000000000000 D4D
004: 0C2000C754859445 0B8
005: D31FF8DA82508D55 46E
006: 4302FFFF4FFFF858 850
007: 15A33314A58FC70B BE5
008: 123490848D121A98 F56
009: 231FE95F21471081 2D3
00A: 741431358BAC117F 65C
00B: 17F174147911D017 9C9
00C: 313750E0D2059510 D2B
00D: 81731331F995F214 0A0
00E: 11741411F765F214 409
00F: 7D723878402180FF 792
010: 183183D014AD8132 B06
011: C013020314C8F99A E8B
012: 804D031A28D39390 206
013: 80DF110161156191 569
014: 62C183D014A136CA 8F8
015: 1308DAE470F      B7A

PPOLL     ID#E1  285 octets

     0123456789ABCDEF sm

000: 0505F4C4C4020202 368
001: 802E008041111058 6B4
002: F32001EE31400000 A12
003: F230000000000000 D3D
004: 075000FD00C6000F 0B2
005: A101E000D7207600 414
006: 0F9E4C4F4F405E39 7CD
007: 0505F4C4C4F3935C B6C
008: 4545405045352515 EB5
009: 141FF8017EC051D1 24A
00A: 04F733001077500A 5BA
00B: 5428017930AEA571 935
00C: 8017C20F6F6A8A30 CCF
00D: 10E068FB13B11BBB 079
00E: 8F2146136AF68D61 40E
00F: 2F0737051D104423 76F
010: 300E17160C1B6042 AD8
011: 1755015760460D00 E2B
012: 37540C2430930002 178
013: 3000136108137109 4B3
014: 8F2C6008F1270011 826
015: 91351181348D84A8 B9B
016: 0038D30350061371 EE7
017: F0F7F21451D5F071 284
018: 5571E508F15171D5 606
019: 1D980F526A82550A 995
01A: 0E27A82A0E15D704 D29
01B: 2032CFB8FAB81154 0CF
01C: 331FFD5D0E415F59 482
01D: 6A1296561BF6F69E 82F
01E: 2C0F6F6B6255117A BCA
01F: 5AD3332008DA9390 F51
020: 175147D507135174 2AF
021: 13706147C9061F50 61A
022: 8F2153717F15F7D5 9B4
023: 27B06440052690E2 D1A
024: 080D51E5F7F15771 0AE
025: D0F061471350701  3DA
```

APPLICATIONS

| | | | | |
|---|---|---|---|---|
| HP41 | P. TENAND | 21 | 13 | Défilement alphanumérique logique |
| | P. LANGLOIS | 22 | 10 | La rapidité des basics |
| HP41 | P. DAVASE | 22 | 16 | Accès direct en ROM par STO b |
| HPIL | L. ANCELET | 22 | 23 | Les messages du module I/O |
| HP75 | DAVID/MARTINET | 22 | 25 | Analyseur de boucle IL |
| HP71 | J.-J. MOREAU | 22 | 30 | Assemblage sans assembleur |
| HP71 | WEIL/DAVID | 22 | 44 | FORTH récursif |
| HP71 | P. DAVID | 23 | 30 | Les buffers du 71 |
| HP71 | M. MARTINET | 24 | 35 | Le registre HARDWARE STATUS |
| HP71 | O. ARBEY | 25 | 42 | Débuter en assembleur |
| HP71 | M. MARTINET | 25 | 48 | Les variables de la MATH Stack |
| HP71 | M. MARTINET | 25 | 57 | Programmation synthétique |
| HP71 | J.-J. DHENIN | 25 | 61 | Pour la rubrique débutant |
| HP41 | P. ANTOINE | 26 | 25 | Inversion du registre C |
| HP71 | P. GUEZ | 26 | 60 | La X710 et le PLOTLEX |
| HP71 | J.-P. BONDU | 26 | 62 | A propos de DISP$ |
| HP71 | DAVID/MARTINET | 27 | 26 | La décomparse |
| HP71 | J.-P. BONDU | 28 | 6 | Optimisation |
| HP71 | J. BAUDIER | 28 | 19 | A propos de PASSWORD |
| HP71 | S. VAUDENAY | 28 | 36 | Assembleur: Stop/Encore |
| | A. GILLET | 29 | 8 | Méthode de réalisation d'un programme |
| HP41 | G. TISSERAND | 29 | 12 | Programmer des XROM avec le ZENROM |
| HP71 | F. LEGRAND | 29 | 21 | BEEP tout le temps (bis) |
| HP71 | F. LEGRAND | 29 | 22 | 1/x à l'échappement |
| HP71 | M. MARTINET | 29 | 28 | Remise à jour des LEX |
| HP71 | J.-J. DHENIN | 29 | 29 | Accès aux variables à partir d'un programme binaire |
| | A.GILLET | 30 | 6 | Le dossier de programme |

DUR ET MOU

| | | | | |
|---|---|---|---|---|
| HPIL | L. ANCELET | 21 | 10 | Carte Modem |
| | P. DAVID | 21 | 12 | Du nouveau chez HP |
| HP71 | P. ANTOINE | 23 | 8 | Module AC pour 71 |
| HP71 | P. ROMASCANO | 23 | 9 | Module de traduction HP71-HP41 |
| HP71 | R. OUVRAY | 25 | 60 | Anomalies du calendrier |
| HPIL | E. GENGOUX | 27 | 10 | Renforcement alimentation unité de disquette HP9114 |
| HP41 | HP/GUEZ | 28 | 11 | Module HP41 Advantage |
| HP71 | M.MARTINET | 30 | 3 | Augmentez la mémoire de votre HP71 |
| HPIL | P. DAVID | 30 | 4 | Les interfaces vidéos HPIL |

PROGRAMMES

| | | | | | | |
|---|---|---|---|---|---|---|
| HP41 | P. DAVID | 21 | 14 | F1 (course) | jeux | RPN |
| HP41 | G. TISSERAND | 21 | 18 | Les tours de Hanoï | jeux | RPN |
| HP41 | J.-P. TOYRE | 21 | 21 | Traductions Hyerogliphes | divers | RPN |
| HP41 | A. OISEL | 21 | 24 | XA & AX (X«=»ASCII) | appli. | RPN |
| HP41 | J.-J. MOREAU | 21 | 25 | SPINVS | jeux | RPN |
| HP15 | D. SAADA | 21 | 27 | Générateur de nombres aléatoires | math. | RPN |
| HP15 | D. SAADA | 21 | 27 | Fonction GAMMA | math. | RPN |
| HP15 | D. CONNAN | 21 | 27 | Méthode du simplexe simpliste | math. | RPN |
| HP75 | F. BARNAUD | 21 | 29 | Matrices | math. | BASIC |
| HP71 | M. MARTINET | 21 | 31 | Fonctions de manipul. de chaînes | appli. | Assembleur |
| HP71 | S. VAUDENAY | 21 | 35 | Histoire de Startup | astuce | BASIC |
| HP71 | J.-P. BONDU | 21 | 35 | Récréation | jeux | BASIC |
| HP71 | P. LANGLOIS | 21 | 35 | Codage en mémoire | appli. | BASIC |
| HP71 | S. VAUDENAY | 21 | 35 | Matrices | math. | BASIC |

| | | | | | | |
|---|---|---|---|---|---|---|
| HP71 | V. HERLICQ | 21 | 36 | Graphique sur 71B | divers | BASIC |
| HP41 | A. PIERARD | 22 | 19 | Affichage des données ASCII | appli. | RPN |
| HP41 | A. PIERARD | 22 | 19 | Caractères spéciaux sur 41CX | appli. | RPN |
| HP75 | DAVID/MARTINET | 22 | 25 | Analyseur de boucle (SCOPE) | appli. | BASIC |
| HP71 | L. ISTRIA | 22 | 30 | Contraste? | appli. | Assembleur |
| HP71 | M. MARTINET | 22 | 32 | Désassembleur | divers | ASS/BASIC |
| HP71 | J.-J. MOREAU | 22 | 48 | Intégrales | math. | BASIC |
| HP71 | J.-P. BONDU | 22 | 49 | Matrices | math. | BASIC |
| HP71 | O. ARBEY | 22 | 50 | VALPROP (Matrices) | math. | BASIC |
| HP41 | P. LANGLOIS | 23 | 12 | Routines complexes | math. | RPN |
| HP41 | M. CLABOT | 23 | 13 | Biorythmes | jeux | RPN |
| HP41 | J.-J. DHENIN | 23 | 15 | Le PANAME ROM et la VIDEO | appli. | RPN |
| HP71 | L. ISTRIA | 23 | 32 | Interception de POLL | appli. | Assembleur |
| HP71 | M. MARTINET | 23 | 34 | REPLACE$ | divers | Assembleur |
| HP71 | S. BARIZIEN | 23 | 36 | FILE? | divers | Assembleur |
| HP71 | IST./MART./DAV. | 23 | 37 | PEEK/POKE partout | divers | Assembleur |
| HP71 | P. DAVID | 23 | 37 | BIP tout le temps | divers | Assembleur |
| HP71 | DAVID/MARTINET | 23 | 38 | ATTN | divers | Assembleur |
| HP71 | M. MARTINET | 23 | 39 | Manipulation de chaînes alphanum. | appli. | Assembleur |
| HP71 | M. MARTINET | 23 | 47 | Chainage de fichiers LEX | divers | BASIC |
| HP71 | D. JACOB | 23 | 49 | Intégration numérique de Gauss | math. | BASIC |
| HP71 | J.-C. FOURES | 23 | 49 | Utilitaires | divers | BASIC |
| HP71 | P. DAVASE | 23 | 50 | Visualisation de la RAM | divers | BASIC |
| HP71 | S. VAUDENAY | 23 | 50 | Débilomètre | jeux | BASIC |
| HP41 | J.-J. MOREAU | 24 | 10 | Matrices | math. | RPN |
| HP41 | G. TISSERAND | 24 | 17 | Monaco | jeux | RPN |
| HP41 | M. CLABOT | 24 | 23 | Vos impots en 85 | divers | RPN |
| HP71 | J.-J. DHENIN | 24 | 30 | PLOTLEX | divers | Assembleur |
| HP71 | M. MARTINET | 24 | 30 | DRIVELEX | divers | Assembleur |
| HP71 | P. DAVID | 24 | 33 | CURLEX | divers | Assembleur |
| HP71 | J.-P. BONDU | 24 | 35 | FKEY | divers | Assembleur |
| HP71 | J.-J. MOREAU | 24 | 37 | Graphisme sur HP71B | divers | FORTH |
| HP71 | J.-C. FOURES | 24 | 46 | LOTO | jeux | BASIC |
| HP71 | J.-C. FOURES | 24 | 46 | Devinez le nombre | jeux | BASIC |
| HP71 | F. BARNAUD | 24 | 46 | Méthode de Romberg | math. | BASIC |
| HP71 | M. WEIL | 24 | 47 | Listage de catalogue | appli. | BASIC |
| HP71 | S. VAUDENAY | 24 | 47 | Récifs | jeux | BASIC |
| HP41 | C. JEGOUZO | 25 | 18 | HP41 et récursivité | appli. | RPN |
| HP41 | D. CASTELAIN | 25 | 26 | Ordinateur de bord | divers | RPN |
| HP41 | PETERSEN/PPCDEN | 25 | 29 | proposition d'1 standard MC en CB | divers | assembleur |
| HP71 | J.-J. MOREAU | 25 | 43 | ENDUP | divers | Assembleur |
| HP71 | J.-P. BONDU | 25 | 47 | Pile CMDS à 16 niveaux | divers | Assembleur |
| HP71 | L. ISTRIA | 25 | 50 | Combinaison arrangement (COMBARR) | math. | Assembleur |
| HP71 | M. MARTINET | 25 | 52 | Heure minute seconde (HMS +, -) | math. | Assembleur |
| HP71 | M. MARTINET | 25 | 57 | Pile CMDS à n niveaux (STKLEX) | divers | Assembleur |
| HP71 | S. VAUDENAY | 25 | 61 | Musique | jeux | BASIC |
| HP71 | X. BILLE | 25 | 63 | Master mind ammélioré | jeux | BASIC |
| HP71 | P. DAVID | 25 | 64 | D.A.O. | jeux | BASIC |
| HP71 | S. VAUDENAY | 25 | 65 | Hantise sur Saturne | jeux | BASIC |
| HP41 | P. COLLIGNON | 26 | 16 | Transformation étoile-triangle | élect. | RPN |
| HP41 | P. LANGLOIS | 26 | 20 | taille du catalogue cassette | divers | RPN |
| HP41 | P. ANTOINE | 26 | 21 | MAJMIN et MIN | appli. | RPN |
| HP41 | C. MORLOT | 26 | 22 | MAJ-MIN | appli. | RPN |
| HP41 | C. MORLOT | 26 | 23 | Balistique | divers | RPN |
| HP41 | C. MORLOT | 26 | 25 | TI-COR | divers | RPN |
| HP71 | J.-J. MOREAU | 26 | 29 | REPEAT | divers | Assembleur |
| HP71 | M. MARTINET | 26 | 33 | MARGIN | divers | Assembleur |
| HP71 | J.-J. DHENIN | 26 | 34 | MENU | divers | Assembleur |
| HP71 | O. ARBEY | 26 | 37 | PRIMLEX | math. | Assembleur |
| HP71 | P. DAVID | 26 | 39 | PRINTLEX | appli. | Assembleur |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| HP71 | J.-J. MOREAU | 26 | 47 | ONKEYLEX | appli. | Assembleur |
| HP71 | J.-J. DHENIN | 26 | 49 | INDATE$ | appli. | Assembleur |
| HP71 | MARTINET/DAVID | 26 | 50 | Formattons, Formattons ! | appli. | ASS/BASIC |
| HP71 | J.-M. SIMON | 26 | 58 | Navigation | divers | BASIC |
| HP71 | L. ISTRIA | 26 | 63 | Suprime les remarques (SUPREM) | divers | BASIC |
| HP71 | J.-C. FOURES | 26 | 64 | TRI BULLE | math. | BASIC |
| HP71 | L. SERANO | 26 | 64 | Compte Bancaire | divers | BASIC |
| HP71 | L. SERANO | 26 | 64 | Compteur téléphone | divers | BASIC |
| HP71 | X. BILLE | 26 | 65 | Caractères Ethiopiens | divers | BASIC |
| HP41 | C. JEGOUZO | 27 | 13 | KIBUR | jeux | RPN |
| HP41 | C. JEGOUZO | 27 | 16 | NIM | jeux | RPN |
| HP71 | MORANDO | 27 | 18 | Phases de la lune | astro. | RPN |
| HP71 | F.B.I. | 27 | 22 | Redéfinition du clavier | divers | Assembleur |
| HP71 | P. DAVID | 27 | 26 | CALCLEX | divers | Assembleur |
| HP71 | M. MARTINET | 27 | 34 | DESAL | divers | Assembleur |
| HP71 | J.-C. FOURES | 27 | 40 | La recherche dichotomique | appli. | BASIC |
| HP71 | O. ARBEY | 27 | 40 | Des polynomes et des croissants | math. | BASIC |
| HP71 | X. BILLE | 27 | 42 | Calcul de dates | divers | BASIC |
| HP71 | J.-C. FOURES | 27 | 43 | Carrés magiques | divers | BASIC |
| HP71 | M. WEIL | 27 | 43 | Tests statistiques | stati. | BASIC |
| HP71 | M. MARTINET | 27 | 71 | MAKELEX | appli. | BASIC |
| HP41 | F. POUPON | 28 | 10 | WRXM | appli. | RPN |
| HP41 | F. POUPON | 28 | 10 | READXM | appli. | RPN |
| HP41 | F. POUPON | 28 | 10 | APPXM | appli. | RPN |
| HP71 | M. WEIL | 28 | 16 | Test de Kolmogorov et Smirhov | math. | BASIC |
| HP71 | J. BAUDIER | 28 | 17 | Translater program 41TO71 | divers | BASIC |
| HP71 | J. BAUDIER | 28 | 19 | Password | divers | FORTH/BASI |
| HP71 | J.-P. BONDU | 28 | 22 | Ecrémez vos programmes (ECREM) | divers | BASIC |
| HP71 | S. VAUDENAY | 28 | 22 | Equate | Appli. | FORTH/ASS. |
| HP71 | A. HERREMAN | 28 | 22 | Développement limité en RPN | math. | BASIC |
| HP71 | A. HERREMAN | 28 | 23 | HP71 Tellement complexe ! | math. | BASIC |
| HP71 | L. ISTRIA | 28 | 24 | Un bridge ce soir | jeux | BASIC |
| HP71 | S. VAUDENAY | 28 | 36 | Assembleur: stop, encore | Appli. | FORTH/ASS. |
| HP71 | S. VAUDENAY | 28 | 37 | GANYMEDE (OTHELLO) | jeux | FORTH/ASS. |
| HP71 | L. ISTRIA | 28 | 40 | Des dates, encore des dates | Appli. | Assembleur |
| HP71 | M.MARTINET | 28 | 46 | Le coin des Lhex | Divers | Assembleur |
| HP41 | C. JEGOUZO | 29 | 12 | Spécial bourse | Appli. | RPN |
| HP71 | J.-P. TOYRE | 29 | 15 | Côtes franç.<=>pied,pouce,x/64 | divers | RPN |
| HP71 | J.-J. DHENIN | 29 | 21 | Carrés magiques | math. | BASIC |
| HP71 | PPC DANEMARK | 29 | 23 | Editeur de texte | divers | BASIC |
| HP71 | F. LE GRAND | 30 | 10 | Le logiciel navigation pour HP71 | navig. | BASIC |
| HP71 | M. WEIL | 30 | 11 | Les tables de la loi | Stat. | BASIC |
| HP71 | M. WEIL | 30 | 12 | Format HPAF | Divers | BASIC |
| HP71 | P. GUEZ | 30 | 13 | Calcul et édition de calendriers | astro. | BASIC |
| HP71 | P. DAVID | 30 | 34 | Le Débogueur Forth | Utili. | ASS/FORTH |
| HP71 | A. HERREMAN | 30 | 41 | Break | Utili. | FORTH |
| HP71 | A. HERREMAN | 30 | 42 | Sort (Tris) | Utili. | FORTH |
| HP71 | M. MARTINET | 30 | 44 | Mémoires de masse | HP-IL | Assembleur |
| HP71 | BARIZ/DAV/MARTI | 30 | 42 | Champs assembleur sous EDTEXT | Utili. | Assembleur |
| HP71 | J. TAILLANDIER | 30 | 49 | L'ordre EXIT (FOR/NEXT) | Utili. | Assembleur |
| HP71 | J.-F. GARNIER | 30 | 50 | Lex PPOLL et demande de service | HP-IL | Assembleur |

## NOUS EN AVONS

Sont en vente au Club:

- Cartes magnétiques pour HP41 au prix de 150FF + 3,10FF de port les 50 cartes;
- Cartes magnétiques pour HP71B au prix de 91FF + 3,10FF de port les 10 cartes;
- Cassettes digitales pour HP82161A: 100 FF pièce, plus 3,10 FF de port par cassette.
- Eproms (2716 & 2732) vierges au prix de 60FF + 3,10FF de port pièce. Pour les Eproms programmées, nous consulter;
- Module TIME (82182A) au prix de 400FF + 6,50FF de port. Livré avec manuel en français;
- Module QUADRAM (82170A) au prix de 400FF + 6,50FF de port;
- Module HPIL (82160A) au prix de 970FF + 6,50FF de port. Livré avec manuel en français;
- Pour les commandes des ZENROM, CCDROM et ROM PANAME nous envoyer 500FF minimum d'arrhes;
- VASM (listing en anglais des 3 premières pages de la 41) au prix de 170FF + 13,50FF de port;
- Le manuel de service de la 41 (en anglais) au prix de 50FF + 6,50FF de port;
- Le manuel technique du convertisseur (en anglais) au prix de 20FF + 3,20FF de port;
- HPIL Interface Specifications (en anglais) au prix de 80 FF + 9,50 FF de port
- The HPIL Integrated Circuit User's manual (en anglais): 40 FF + 6,50 FF de port
- Interface Kit Technical Guide (en anglais): 15 FF + 3,50 FF de port
- Manuel technique du Minitel R (Ref. 250 Alcatel): 40 FF + 6,50 FF de port
- Les anciens numéros au prix de:
  No 1 à 3 au prix de 15FF + 3,20FF de port par numéro,
  No 4 à 10 au prix de 20FF + 3,20FF de port par numéro,
  No 11 à 17 au prix de 25FF + 6,50FF de port par numéro,
  à partir du No 18 au prix de 35FF + 6,50FF de port par numéro.

Les ports sont les tarifs PTT actuels en non urgent.

## ACHATS GROUPES

Nous assurons des commandes groupées pour ceux qui désirent profiter de notre réduction "spécial Club", soit 25% par rapport au prix HP. Pour ceux qui le désirent, nous assurons l'envoi postal en urgent ou en non urgent. Ces achats concernent aussi bien des modules que des calculatrices et autres.

NOTA: ces deux rubriques "nous en avons" et "achats groupés", sont des services "Club" que nous rendons exclusivement aux membres effectivement inscrits au club et possédant leur carte de membre PPC-PARIS.

# 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 |_|_|_|

Le Journal JPC est le bulletin de liaison entre les membres de l'association "PPC-PC", régie par la loi de 1901. Le Club est éditeur du JPC, et son siège est au 56, rue Jean-Jacques Rousseau, 75001 PARIS.

La maquette de ce journal a été préparée par Michel MARTINET et Philippe GUEZ. Elle a été réalisé par Michel MARTINET avec la participation de Jean-Jacques DHENIN sur un système comprenant un HP71B, un lecteur de disques HP9114A et une imprimante "LaserJet".

Directeur de la publication Philippe GUEZ
Numéro ISSN: 0762 - 381X