MICRO REVUE

LA REVUE DE L'INFORMATIQUE PORTABLE

© 1985 Bimestriel

Edité par PPC-T

N° 6 MAI - JUIN 1985

40 F.

TABLE DES MATIERES

Table des matières

Page	Contenu
5	EDITORIAL
7	HP-41 PROGRAMMES
8	Assignations
9	A$, CONFIG
13	DIFPLOT
15	CB PRGM
20	SYN-SWW Mise au point de fondu-enchainé
21	Startrek 2.0
23	Id Inverse drapeaux
24	Puissance d'une matrice
26	Champ de mines
27	Nb Nombre d'octets
28	F? non perturbant
	APPLICATIONS
29	Assignations
30	Chocs
31	GTO/2
33	Livret d'application Physique
34	HP-IL, SANYO 550
38	#ae2
40	Circuit logique
41	W
43	Toulmath
	HP-71
64	Imprimer sur HP82143A
65	Les parisiens au travail
	FORTH
74	Lettre
76	Le Novix 4000 (excuses, je n'ai pas eu le temps de traduire)
76	Initiation au langage FORTH
	HP-75
92	Installation du module 8K maison
94	Agrandir
95	Ajustement de courbes
98	DUMP du PAC SCREEN sur HP82905B
99	Rébus, suite et fin
101	Poésie au mètre
	HP-16
110	Conversion flottant 32 bits (Perkin-Elmer)
115	MAGAZINE

La rue sans frais

C'est le creux du sort
Sans but ni amie
C'est une aile encore
Sans toit sans envie
C'est un corps moulu
Dans la rue des plaies
C'est un pleur vaincu
Dans la peur des fouets
Ce vent de folie

C'est l'epee de mort
Sans joie et sans vie
C'est un vol d'accord
Sans rire sans patrie
C'est un air tordu
Dans la nuit d'apres
C'est un cri decu
Dans l'ardeur sans frais
Ce vent de folie

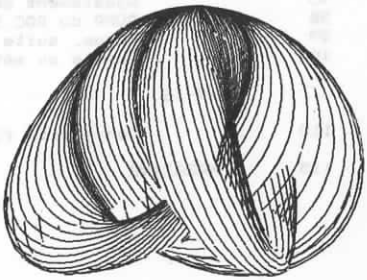
C'est un feu d'abord
Sans frein sans folie
C'est l'ami a bord
Sans rien ni genie
C'est un or perdu
Dans le monde d'acces
C'est le coeur fourbu
Dans la ville qu'on sait
Ce vent de folie

C'est un bruit dehors
Sans air sous la pluie
C'est le clan d'alors
Sans cle qu'on oublie
C'est un bras sans but
Dans l'abri qui nait
C'est un chien de plus
Dans l'instant des rets
Ce vent de folie

Garantie 3 ans
HEWLETT PACKARD

** Surface de Boy **
2/2/7/1/*/*
le 84/04/30 a 18:18

Garantie 3 ans
HEWLETT PACKARD



EDITORIAL

Quand nous avons lancé MICRO-REVUE, il y a maintenant un an, bien des gens craignaient que nous ne puissions pas tenir la distance. les autres clubs, aux USA, en Angleterre, etc... se plaignent de ne pas avoir assez de soumissions d'articles.

Et bien, il faut reconnaitre une chose : vous êtes formidables, plus le temps passe et mieux vous faites votre travail. Ce n'est pas toujours mon cas. Depuis quelques semaines j'ai de (petits) problèmes familiaux qui m'ont fait perdre beaucoup de temps et je suis en retard pour la mise en page de ce numéro 6 de MICRO-REVUE. J'étais assez inquiet du travail à faire pour ce numéro. Or j'ai pu faire la mise en page en 1 jour et demi (environ 20 heures). Ceci parce-que vous m'envoyez de plus en plus d'articles écrits au bon format, sur les imprimés étudiés pour et en respectant les indications du kit auteur. Du coup mon travail s'en trouve énormément facilité, et le niveau de la revue s'améliore. Merci et Bravo.

Dans ce numéro, de nombreux programmes. Un compte rendu du module TOULMATH ; je ne publie pour l'instant que les fiches, faute de place, mais aussi parceque mon lecteur de cartes est en panne. vous pouvez demander les programmes.

Il y a encore peu de soumissions pour le HP-71, mais les Parisiens, qu'ils soient de PPC-T ou du chapitre de Paris s'activent.

Même ceux qui n'ont pas de HP-75 devraient s'intéresser à l'article de Jean Dutertre (exemple page précédente). Pour Boy, vous attendrez le prochain numéro.

Le Forth est présent avec un article d'initiation d'un expert en la matière, Olivier SINGLA est formateur dans le cadre de l'Education Nationale, il est l'auteur d'un FORTH pour MICRAL 8022 et en prépare un pour les nanoréseaux.

Vous aurez sans doute ce numéro assez tard, les "ponts" du mois de Mai allongent les délais d'impression. Mais j'écris ceci le 8 Mai, et en ce moment votre club est représenté au SICOB de Printemps, nous y reviendrons.

Bonne lecture
J-D Dodin

ASSEMBLEE GENERALE DE PPC-T

SAMEDI 29 JUIN 1985 A 14h 30

Maison des jeunes et de la culture Jacques Prévert

Route de Seysse (à droite après le passage à niveau)

Ordre du Jour :

Renouvellement du bureau
Le point sur les perspectives
Venez nombreux

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

EDITION :
Edité par le club de Programmation Personnelle sur Calculatrice de Toulouse (PPC-T).

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

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

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

TARIFS (franco de port) :
Prix de vente au numéro : 40F (50F pour envoi par avion).
Abonnement individuel : Abonnement annuel 200F (au lieu de 240, prix de revient au numéro), 250F pour envoi par avion. Les envois à destination de l'Afrique n'arrivent pas toujours à destination, nous ne pouvons en aucun cas en assumer la responsabilité, dans la mesure du possible nous donner une adresse en Europe.

SOUMISSION D'ARTICLES :
Tous les membres du club sont invités à fournir des articles pour MICRO-REVUE. L'original à nous faire parvenir doit être tapé à la machine avec un ruban bien noir, recto seul, simple interligne, avec une largeur maximale de 15 cm. Un envoi sur disquette IBM simple face, sur disquette pour HP150, sur cassette HP82161A est recommandé.

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

Nous faire parvenir cartes ou cassettes pour les programmes, ainsi qu'un listing.

Cet original sera utilisé tel que pour la publication, soignez-le ! Ceux qui prévoient à l'avance d'écrire des articles sont invités à nous demander un "Kit Auteur" qui leur donnera les modèles et toutes les instructions nécessaires.

(c) 1985 REPRODUCTION INTERDITE

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

Ont collaboré à ce numéro :

Par ordre d'entrée en page...

Constantin Sotiropoulos, Jean Thiberge, Jacques Défendini, Robert Schwartz, D. Delefortrie, Eric Aubourg, Bruno Tredez, Yves Bertran, Laurent Laville, Georges-Noel Nicolas, Patrick Bélaire, Bernard Jousse, François Legrand, Michel Martinet, Pierre David, Jean-Jacques Dhénin, Bernard Morisseau, Charles Moore, Olivier Singla, Vincent Delorme M. Paulus, Jean-Claude Liehn, Jean Dutertre, J-P Dental, Damien Debril, Dr Leray, Michel Locusse,

HP-41

PROGRAMMES

SOTIROPOULOS C.
T 434
13 a. A. ARCHAMBAULT
95110 SANNOIS

LE PETIT PROGRAMME QUI SUIT PERMET DE FAIRE UNE ASSIGNATION SUR LA TOUCHE 11 <ET -11>.
IL PEUT ETRE UTILISER POUR TESTER DES ASSIGNATIONS RAPIDEMENT ET SANS AVOIR A INTRODUIRE LE CODE DE LA TOUCHE.
1E CODE , ENTER , 2d CODE , R/S.
DO NOT SST,DO NOT HALT EXECUTION.OK?

```
01♦LBL "1A"
02 "XTOA"
03 11
04 PASN
05 CHS
06 PASN
07 RDN
08 X<>Y
09 "♦♦♦μμμ"  240,0,0,0,12,12
10 RCL [                  12
11 X<> c
12 RDN
13 ""        240
14 XTOA
15 RDN
16 XTOA
17 "⊢=" 127,1
18 R↑
19 XTOA
20 RDN
21 XTOA
22 "⊢σ"  127,9
23 RCL [
24 STO 00
25 X<> T
26 STO c
27 CLX
28 END
```

SOTIROPOULOS C. T434

LE PETIT UTILITAIRE CI-CONTRE PERMET, GRACE A GETKEY, L'EFFACEMENT DE L ASSIGNATION DESIREE TOUT SIMPLEMENT EN APPUYANT SUR LA TOUCHE LA PORTANT.
SI VOUS APPUYEZ SUR UNE DES TOUCHES DU HAUT<ON/USER/PRGM/ALPHA> LE MESSAGE STANDARD KEYCODE ERR S'AFFICHE.

```
01♦LBL "CLKEYX"
02 "#CLKEY "
03 ATOX
04♦LBL 00
05 RCLFLAG
06 CF 21
07 AVIEW
08 STOFLAG
09♦LBL 01
10 RDN
11 GETKEY
12 X=0?
13 GTO 01
14 X≠Y?
15 GTO 03
16 "⊢-"
17 GTO 00
18♦LBL 03
19 RCLFLAG
20 FIX 0
21 CF 29
22 ARCL Y
23 AVIEW
24 STOFLAG
25 ANUM
26 CLA
27 PASN
28 END
```

A S CONFIG

Voici le programme d'assignation synthétique le plus compact que je connaisse; il demande le module XFM ou une 41 CX. Il occupe 91 octets, soit 13 registres ou 7 rangs de codes-barres.

Mode d'emploi. Entrer PREFIXE ↑ SUFFIXE ↑ Pos. touche d'assign. , R/S

Analyse du programme ligne par ligne.

1	Assignation synthétique
2	au cas où l'assignation ne pourrait être faite
3 -4	assignation-bidon (%). Voir note.
5 -6	ce que le programme va rechercher dans les registres d'assignation, stocké en L: 242, 4, 76 (αL = LBL 03 %)
7	rappel du registre c
8 -10	Mise en place du préfixe
11	devient pointeur de reg. d'ass.
12-13	adjonction du suffixe
14	devient décrément pour la ligne 19
15	stockage en T de la vraie chaîne qui remplacera la chaîne de l'assignation-bidon; jusqu'à la l. 34, la pile ne doit pas monter.
16	registre c provisoire: 245, 1, 105, 12, 0, n (n quelconque)
17	mis en place; sans danger puisque le module XFM est présent
18	début de la boucle de recherche
19	mise à jour du pointeur de registre(pas de -, L est occupé)
20-21	rappel du registre d'assignation pointé
22	adjonction de 3 nuls (244, 127, 0, 0, 0) pour être sûr que...
23	ASTO stocke 6 caractères en X
24	En-tête des registres d'assignation (241, 240)
25	le registre est complété
26-28	et renvoyé à sa place via X
29	la chaîne de l'assignation-bidon est placée en X
30	POSA la renvoie en L, donnant sa position dans le registre; si absente:
31-32	-1; retour à la ligne 18, -1 va servir à incrémenter le pointeur.
33	rotation du reg. d'ass. pour mettre la chaîne-bidon en tête
34	rappel de la bonne chaîne qui pousse la chaîne-bidon en N
35-36	d'où on la fait disparaître; voir note
37-41	calcul de l'argument de AROT qui remettra T en tête
42-43	remise en place du registre d'assignation
44-45	puis du registre c
46-47	conclusion

NOTE. Ce programme suppose que % n'a pas été assigné antérieurement; pour éviter un mauvais fonctionnement dans ce cas, on pourrait, juste après PASN, cueillir le code de la touche dans ├ et l'adjoindre à la chaîne-bidon (l. 5) il en coûte 14 octets. Si l'on cueille ce code, ne pas l'adjoindre à la vraie chaîne car il constitue (l. 36) un couvercle à des 0 possibles et le dernier AROT donnerait un mauvais résultat. Le programme suppose aussi qu'aucune assignation d'une fonction d'un octet n'a été faite en utilisant le préfixe 0.

Commentaires. Une fois le GTO 00 de la l. 32 compilé "à froid", le programme peut être entièrement SSTé.

Ligne 46. R↑ laisse la pile avec un nombre "chrétien" en X

X	n° du registre dans la table d'assignation
Y	registre c
Z, M	le registre d'assignation
T	argument de l'AROT qui mettrait en tête le 'colocataire' du registre
L	place de l'assignation faite dans le registre

Ceci laisse la voie/à toute exploitation. /ouverte

Jean THIBERGE 10292 T 167

```
01♦LBL "A$"
02 CF 25
03 "%"
04 PASN
05 "aL"
06 ASTO L
07 RCL c
08 CLA
09 R↑
10 XTOA
11 CLX
12 R↑
13 XTOA
14 CLX
15 ASTO T
16 "=ip♦"
17 ASTO c

18♦LBL 00
19 ST- Y
20 CLA
21 ARCL IND Y
22 "⊢♦♦♦"
23 ASTO X
24 ""
25 ARCL X
26 CLX
27 RCL [
28 STO IND Y
29 X<> L
30 POSA
31 X<0?
32 GTO 00
33 AROT
34 ARCL T
35 FRC
36 STO \
37 CLX
38 5
39 LASTX
40 -
41 AROT
42 RCL [
43 STO IND Z
44 R↑
45 STO c
46 R↑
47 .END.
```

```
01♦LBL "CONFIG"
02♦LBL 01
03 "1e PAGE"
04 -7

05♦LBL [
06 CF IND X
07 ISG X
08 GTO [
09 PROMPT
10 ENTER↑
11 STO Z
12 4
13 X>Y?
14 GTO 01
15 ST+ X
16 /
17 ,9
18 X<>Y
19 X>Y?
20 SF 05
21 FRC
22 ST+ X
23 X>Y?
24 SF 07
25 FRC
26 ST+ X
27 X>Y?
28 SF 00
29 FRC
30 *
31 X≠0?
32 SF 03

33♦LBL 00
34 CLX
35 "4, 8 OU 16K"
36 PROMPT
37 SF 25
38 GTO IND X
39 +
40 GTO 00

41♦LBL 04
42♦LBL 08
43♦LBL 16
44 4
45 /
46 ST+ Z
47 LASTX
48 X↑2
49 R↑
50 -
51 X<0?
52 GTO 01
53 SIGN
54 +
55 2
56 X≠Y?
57 SF 06
58 X≠Y?
59 SF 04
60 X↑2
61 X<Y?
62 SF 01
63 X>Y?
64 SF 02
65 -7
66 " "
```

```
67♦LBL \
68 FS? IND X
69 "⊢♦"
70 FC? IND X
71 "⊢—"
72 ISG X
73 GTO \
74 PROMPT
75 .END.
```

```
01♦LBL "CONFIG"
02♦LBL 01
03 "1e PAGE"
04 CF 22
05 PROMPT
06 4
07 FS?C 22
08 X>Y?
09 GTO 01
10 X<>Y
11 X<>F

12♦LBL 00
13 "4, 8 OU 16K"
14 0
15 PROMPT
16 SF 25
17 GTO IND X
18 GTO 00

19♦LBL 04
20♦LBL 08
21♦LBL 16
22 4
23 /
24 STO Y
25 LASTX
26 X↑2
27 ST+ Y
28 ST+ Y
29 X<>F
30 ST+ Z
31 +
32 X<>F
33 X<Y?
34 GTO 01
35 2,4340561
36 " "

37♦LBL B
38 FRC
39 95
40 FS? IND L
41 CLX
42 XTOA
43 CLX
44 Ei
45 *
46 X≠0?
47 GTO B
48 PROMPT
49 .END.
```

A l'invitation du confrère et sans trop me pencher par crainte de tomber, voici deux réductions de son programme CONFIG: celui de 143 octets n'utilise pas le module XFM, ce dernier permettant d'obtenir un programme de 107 octets. Certaines particularités du programme d'origine n'ont pas été reprises:

- CF 27; si la mesure est sage dans d'autres programmes, elle ne se justifie guère ici (à moins que les touches numériques aient été assignées);
- AOFF; sans objet si le programme n'est pas appelé par un autre programme où l'on serait alors en mode ALPHA, et si l'on ne va pas s'y mettre bêtement;
- TONE 6, annonçant une entrée incorrecte; le message redemandant la donnée devrait suffire;
- CF 22; les programmes ci-après imposent une entrée numérique.

Commentaires sur les programmes. A l'entrée du n° de ligne, le programme vérifie seulement que le n° de page est >= à 4; ce n'est que lorsque le nombre de Kmots est donné que le programme vérifie la cohérence de l'ensemble (p + k/4 <= 16), renvoyant sinon au début de CONFIG, car si le nombre de Kmots est correct, c'est bien le n° de la première page qui est à modifier. NB. Si, après avoir entré un n° de 1ère page, vous voulez modifier celui-ci, entrez 1 (non comme Kmots, mais comme 1ère page), R/S

Le programme avec module XFM propose une autre représentation de la position des interrupteurs; si vous voulez inverser cette position, rien n'est plus facile: rendez-vous ligne 39!

Je me suis payé une coquetterie de programmeur en utilisant des LBL locaux exclus du mode USER; les LBL locaux ordinaires ou numériques supérieurs à 16 font fort bien l'affaire.

Que le Microcode vous tienne en joie!

J. THIBERGE 10292 T 167

A$

ROW 1: LINES 1- 5

ROW 2: LINES 5- 13

ROW 3: LINES 13- 18

ROW 4: LINES 19- 24

ROW 5: LINES 24- 31

ROW 6: LINES 32- 40

ROW 7: LINES 41- 47

CONFIG

ROW1: LINES 1- 3

ROW2: LINES 3- 9

ROW3: LINES 9- 18

ROW4: LINES 18- 26

ROW5: LINES 27- 33

ROW6: LINES 33- 38

ROW7: LINES 38- 48

ROW8: LINES 49- 58

ROW9: LINES 58- 65

ROW10: LINES 66- 71

ROW11: LINES 71- 74

CONFIG

ROW1: LINES 1- 3

ROW2: LINES 3- 9

ROW3: LINES 10- 12

ROW4: LINES 12- 20

ROW5: LINES 20- 28

ROW6: LINES 29- 34

ROW7: LINES 34- 39

ROW8: LINES 39- 47

ROW9: LINES 48- 48

DIFPLDT

DEFENDINI Jacques

15 av de Tarente

29200 Brest

(T 477)

Brest le 11/10/84

Bonjour,

Ce programme long de 754 bytes permet de tracer la solution d'une équation differentielle de la forme $y' = f(x,y)$ ou $y'' = f(x,y,y')$.

Je reprends le programme du Module Math de la 41 pour le calcul de l'équation differentielle avec la méthode de Runge-Kutta :

$$\left.\begin{aligned} k_1 &= h f[x_n, y_n] \\ k_2 &= h f[x_n + \tfrac{h}{2}, y_n + \tfrac{k_1}{2}] \\ k_3 &= h f[x_n + \tfrac{h}{2}, y_n + \tfrac{k_2}{2}] \\ k_4 &= h f[x_n + h, y_n + k_3) \end{aligned}\right\} y_{n+1} = y_n + [k_1 + 2k_2 + 2k_3 + k_4]\frac{1}{6}$$

utilisation : mettre la fonction $f(x,y)$ ou $f(x,y,y')$ en memoire avec la correspondance : $x \longrightarrow X$; $y \longrightarrow Y$; $y' \longrightarrow Z'$

Puis faire XEQ "DIFPLOT" et répondre aux questions de la 41.

Ce programme a un gros defaut, il est lent, alors avis aux amateurs.

Le programme utilise l'Imprimante ZL, mais elle n'est pas indispensable.

Les memoires 0 à 23 sont utilisées ainsi que les cartes magnetiques.

En esperant que ce programme sera amelioré

Amicalement

J. Defendini

PS : je remercie Yves BERTRAND (T526) pour sa lettre et ses programmes.

```
 01♦LBL "DIFPLOT"
CF 28  CF 29  FIX 0
"NAME?"  AON  PROMPT
ASTO 09  AOFF
"MODE STANDBY"  PROMPT


 12♦LBL 05
"Y MIN?"  PROMPT  STO 10
STO 00  "Y MAX?"  PROMPT
STO 11  STO 01  X<=Y?
GTO 05  "AXIS?"  PROMPT
STO 04

 26♦LBL 04
"X MIN?"  PROMPT  STO 12
"X MAX?"  PROMPT  STO 13
X<=Y?  GTO 04  "X INC?"
PROMPT  STO 14  2  /
STO 17  "ORDER=?"
PROMPT  1  X=Y?  CF 02
CLX  2  X=Y?  SF 02
"Y0=?"  PROMPT  STO 19
FC? 02  GTO b  "Y0.=?"
PROMPT  STO 20

 58♦LBL b
ADV  "    HR PLOT OF "
ARCL 09  PRA  RCL 12
ABS  RCL 13  ABS  X<Y?
X<>Y  LOG  INT  2  -
STO 16  "    X <UNITS="
X≠0?  "├ E"  ACA  X=0?
1  ACX  ">"  ACA  7
ACCHR  PRBUF  139
STO 02  XROM "PRAXIS"
PWRDN  FIX 0  RCL 02
RCL 11  RCL 10  -  /
STO 11  RCL 10  CHS  *
STO 10  RCL 13  RCL 12
-  RCL 14  X>0?  /  ABS
10  /  RND   E3  /  1  +
STO 13  RCL Y  RCL 14
X>0?  GTO 13  CHS  /
STO 14

123♦LBL 13
.006  STO 07

126♦LBL 07
CLX  XEQ "DIFF"  RCL 11
*  RCL 10  +  RND  139
X<>Y  X>Y?  X<>Y  X<0?
0   E3  /  RCL 07  INT
+  STO IND 07  RCL 14
ST+ 12  ISG 07  GTO 07
.005  STO 07

152♦LBL 17
RCL 07  1.001  +  STO 08


157♦LBL 18
RCL IND 08  FRC
RCL IND 07  FRC  X<=Y?
GTO 19  X<> IND 08
X<> IND 07  X<> IND 08

167♦LBL 19
ISG 08  GTO 18  ISG 07
GTO 17  RCL 12  RCL 14
3  *  -  RCL 16  10↑X  /
CLA  RND  ENTER↑  ABS
X=0?  GTO 00  LOG  X<>Y
X>0?  GTO IND Y  X<>Y  1
+  X<>Y  GTO IND Y


195♦LBL 00
"├ "

197♦LBL 01
"├ "

199♦LBL 02
"├ "

201♦LBL 03
PWRUP  ARCL X  ACA
1.006  STO 07  0  STO 08
2  RCL 00  INT  Y↑X
STO 15  RCL 00  FRC


216♦LBL 27
RCL IND 07  FRC  X=Y?
GTO 28  X<>Y  1 E3  *
RCL 08  -  SKPCOL
ST+ 08  RCL 15  ACCOL  1
ST+ 08  2  RCL IND 07
INT  Y↑X  STO 15  GTO 29


238♦LBL 28
2  RCL IND 07  INT  Y↑X
ST+ 15

244♦LBL 29
RCL IND 07  FRC  ISG 07
GTO 27  RCL 06  FRC
1 E3  *  RCL 08  -
SKPCOL  RCL 15  ACCOL
PRBUF  PWRDN  .005
STO 26  ISG 13  GTO "X"
OFF

265♦LBL "X"
SF 01  ISG 26
GTO "DIFF"  CF 01
GTO 13

271♦LBL "DIFF"
FS? 02  GTO "D2"  RCL 19
RCL 12  XEQ IND 09
XEQ 21  STO 21  XEQ 20
STO 22  XEQ 20  ST+ 22
ENTER↑  +  RCL 19  +
RCL 12  RCL 17  ENTER↑
+  STO 20  +  XEQ IND 09
XEQ 21  RCL 21  +
RCL 22  ENTER↑  +  +  3
/  ST+ 19  FC? 01  GTO a
RCL 20  ST+ 12

308♦LBL a
RCL 19  FS? 01  GTO "X"
RTN

313♦LBL "D2"
RCL 20  RCL 19  RCL 12
XEQ IND 09  XEQ 21
STO 21  STO 23  RCL 17
XEQ 22  STO 22  STO 23
RCL 17  RCL 23  RCL 20
+  RCL 21  XEQ 24
ST+ 22  ENTER↑  +
STO 23  RCL 17  ENTER↑
+  XEQ 22  RCL 22
ENTER↑  +  +  RCL 21  +
3  /  RCL 20  +  STO 20
LASTX  RCL 22  RCL 21  +
3  /  +  RCL 17  ENTER↑
+  STO 21  *  ST+ 19
FC? 01  GTO c  RCL 21
ST+ 12

367♦LBL c
RCL 19  FS? 01  GTO "X"
RTN

372♦LBL 20
RCL 19  +  RCL 12
RCL 17  +  XEQ IND 09
XEQ 21  RTN

381♦LBL 22
RCL 23  RCL 20  +
RCL 23

386♦LBL 24
2  /  RCL 20  +  R↑  *
RCL 19  +  RCL 12  R↑  +
XEQ IND 09

399♦LBL 21
RCL 17  *  .END.


                    CAT 1
LBL'DIFPLOT
LBL'X
LBL'DIFF
LBL'D2
.END.        756 BYTES
```

CB PRGM

CODES-BARRES PRGM

Le PRGM "CBPG" nécessite la table traçante HP 7470A ainsi que le module PLOTTER. Je ne cite pas le XF dont la seule fonction utilisée est PSIZE, donc facultative et je pense que tous les lecteurs possèdent maintenant ce XF dont j'aurais du mal de me passer.

Le 1er affichage demande de taper sur la touche 'CHS' si l'on désire que le PRGM soit PRIVATE (Pas 05) et la valeur adéquate est STO dans le REG 01 : la partie INT indique le N° du 1er RANG à "code-barrer" et la partie FRC indique le nbre d'Octets par RANG.

Le nom du PRGM est ensuite demandé puis ASTO dans le REG 02 et les Pas 12 à 18 servent à l'initialisation. Signalons que les DRAPeaux 1 & 3 sont levés par les Pas 15 à 17 et que les Pas 13 & 14 sont destinés à tester la présence du PRGM à "code-barrer" et à le PACKer éventuellement avant d'entamer la suite de l'exécution.

Je signale qu'il pourrait être souhaitable de placer PCLBUF avant PINIT car cette dernière fonction ne modifie pas les paramètres qui intéressent les CB (BCSIZE) dans le cas où des paramètres différents (de PINIT) auraient été introduits précédemment dans le BUF.

Le LBL 00 du Pas 19 est le début de "la grande-boucle" qui s'étend jusqu'au Pas 101. Les Pas 20 à 25 délimitent la surface maximale utilisable sur la feuille de papier. On profite du zéro (Pas 20) pour initialiser le REG 03 dont nous verrons l'utilité plus loin.

Les Pas 26 & 27 précisent la taille des caractères qui seront tracés (en prévision de la réduction de format effectuée pour MICRO REVUE, j'ai adopté une dimension supérieure à celle qui est attribuée automatiquement par PINIT).

Les Pas 28 & 29 sont indispensables pour écrire dans le sens horizontal car PINIT précise d'écrire de haut en bas, ce qui ne convient pas dans la majeure partie des cas (disposition contestable).

On profite du nbre pair 90 pour faire sortir la plume 2 qui est réservée à l'écriture de l'ALPHA (numéros de RANGs compris). Il convient de réserver la plume N° 1 pour le tracé des CB afin de l'émousser le moins possible.

Les Pas 31 à 38 permettent d'écrire le nom du PRGM en tête de la feuille. MOVE place la plume en fonction des valeurs placées en X & Y et on lève le DRAPeau 17 pour interdire à la plume de sauter à la ligne suivante.

Les Pas 41 à 53 permettent, lors de la première exécution (sur la 1ère page uniquement), d'écrire éventuellement (CF 23 & FS? 23) des annotations sur la même ligne que le nom du PRGM (N° et page de MR, par exemple). L'écriture est faite avec des caractères légèrement plus petits (3 CSIZE).

Au LBL 09, le REG 00 est chargé d'une valeur qui déterminera l'origine de chacun des 27 RANGs qu'il est possible de tracer sur une feuille (137=5 fois 26 + 7, 7 étant la position du 1er RANG, 26 le nbre d'intervalles et 5 étant l'INCrément, donc la distance entre 2 RANGs.

La boucle LBL 01 (Pas 57 à 96) est parcourue à chaque RANG pour le créer. Les Pas 58 & 59 éteignent la '41' en cas de LOW BAT. De 60 à 73, on écrit le N° de

RANG en le faisant précéder d'un SPACE s'il est inférieur à 10. Ce N° est issu du REG 01. 'E' (=1) est l'intervalle qui sépare le 1er DIGit (chiffre ou SPACE) de la limite de la marge déterminée par 'LIMIT' (Pas 25) et la valeur issue du REG 00 détermine la position dans le sens horizontal comme nous l'avons vu précédemment. Le 2 du Pas 70 choisit la plume N° 2 et servira plus loin et le 5 aurait pu être placé avant le '+' du Pas 76.

Les Pas 74 à 81 déterminent où doit commencer le RANG de CB en tenant compte de l'éventualité de numéros de RANGs à 3 chiffres (supérieurs à 99). ALENG compte le nombre de chiffres (°) qui est multiplié par 2 (Pas 70) et on ajoute le 5 du Pas 69 précité. (°) J'avoue qu'un autre procédé aurait pu être adopté compte tenu du fait que ALENG ne peut répondre que 2 ou 3. On RCL le contenu du REG 00 et on y retranche 3 de façon à placer la plume au bon endroit pour le traçage des CB.

Les Pas 82 à 85 commandent le traçage d'1 RANG de CB après avoir placé en Y le nom du PRGM (cf Pas 11) et en X le pointeur "qui-va-bien". Ce pointeur sera INCrémenté de 1 par BCP et replacé en X; l'usage de la PILE étant limité, on STO cette nouvelle valeur dans le REG 01.

Lorsque tout le PRGM est "code-barré", zéro est placé en X et on saute alors au LBL 13 (Pas 87 & 88).

WHERE place en X & Y les coordonnées du point où se trouve située la plume à la fin du RANG de CB et en Z, la situation de la plume (haute = 0 ou basse = 1, 0 dans notre cas). Les Pas 89 à 94 comparent la longueur du RANG qui vient d'être tracé avec la longueur du RANG précédent et STO (reSTO ?) la plus grande valeur dans le REG 03. Nous voyons pourquoi il était nécessaire de placer zéro dans 03 dès le début. Cette longueur maximale nous permettra de déterminer où le trait vertical de droite (encadrement) devra être tracé. OK ?

Lorsque les 27 RANGs sont tracés, le TEST conditionnel du Pas 95 "enjambe" le pas suivant et l'affichage nous invite à placer une nouvelle feuille sur la table. Un 'R/S' nous redirigera vers le LBL 00.....

Nous avons vu comment s'effectue le branchement au LBL 13 où le DRAPeau 03 va être baissé de façon à indiquer dans le sous-PRGM LBL 06 (Pas 105 à END) qu'il s'agit de clore le traitement (OFF du Pas 129). Le XEQ "RS" du Pas 103 pourrait être supprimé: c'est ce qui trace mes initiales..... Signalons simplement que si vous supprimez "RS", les Pas 107, 109 & 110 devront également être supprimés. La fonction LOCATE permet de tracer le trait d'encadrement dont les dimensions doivent être précisées dans la PILE. Le trait vertical de droite ne doit pas (ne peut pas) être tracé en dehors d'une limite maximale qui est, dans notre cas, 191,25 mm (que j'ai arrondie). La plume N° 2 doit être précisée (Pas 117 & 118) pour tenir compte du cas où cette routine n'est pas exécutée à la suite de "RS" et compte tenu du fait que nous avons décidé que seuls les CB seront tracés avec la plume N° 1. La fonction FRAME trace le trait d'encadrement dont les limites ont été précisées par LOCATE. Je laisse deviner par ceux qui connaissent les fonctions du module PLOTTER, pourquoi j'ai adopté LOCATE plutôt que LIMIT (?).

La fonction CSIZEO définit les paramètres dimensions, rapport d'aspect et inclinaison des caractères en fonction des valeurs placées en X, Y & Z.

Si le PRGM "CBPG est placé en MEM centrale, la place disponible pour le PRGM à "code-barrer" est égale à :

319 REG auxquels on retranche :

- le nbre de REG occupés par le (ou les PRGM de service ("CBPG" et "RS", par exemple),
- les 4 REG du SIZE de "CBPG",
- les 26 REG du BUF (à ne pas oublier !)
- les REG utilisés par d'éventuelles ASN de fonctions.

Une solution consiste à placer les PRGM de service en EPROM ou dans la RAM d'un MLDL mais il en est une autre beaucoup plus abordable : placer et XEQécuter le PRGM de service en XMEM. Pour cela il faut respecter quelques conditions dont, en attendant la fonction MC adéquate, la présence en MEM centrale d'un petit PRGM qui commande l'exécution du PRGM situé en XM (dans les 125 premiers REG). Affaire à suivre.....

Merci de votre éventuelle attention ! rS T178)

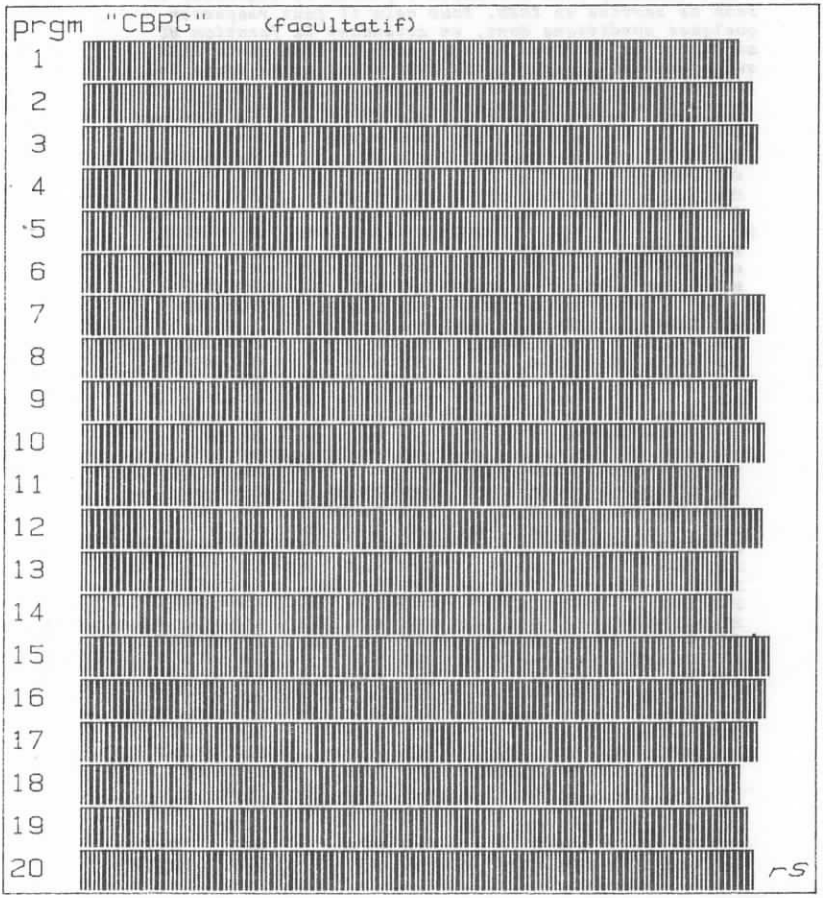
```
 01♦LBL "CBP39 FC?C 01 80 -        120 RCL 03
G"           40 GTO 09  81 MOVE     121 ,
 02 4        41 CSIZE   82 RCL 2    122 ENTER↑
 03 PSIZE    42 30      83 RCL 01   123 X<> Z
 04 1,16     43 X<>Y    84 BCP      124 X<>Y
 05 "PRIV=CH44 MOVE     85 BC       125 LOCATE
S"           45 "TITRE" 86 STO 01   126 FRAME
 06 PROMPT   46 CF 23   87 X=0?     127 FRAME
 07 STO 01   47 AON     88 GTO 13   128 FC? 03
 08 "NOM PRG48 PROMPT   89 RCL 03   129 OFF
M"           49 AOFF    90 WHERE    130 END
 09 AON      50 FS? 23  91 X<> T
 10 PROMPT   51 LABEL   92 X<>Y      01♦LBL "RS"
 11 ASTO 02  52 4       93 X>Y?      02 WHERE
 12 PINIT    53 CSIZE   94 STO 03    03 2
 13 ,        54♦LBL 09  95 ISG 00    04 ST+ Z
 14 BCP      55 7,13705 96 GTO 01    05 RDN
 15 "P♦♦1♦♦"56 STO 00   97 XEQ 06    06 3
 16 RCL [    57♦LBL 01  98 "FEUILLE07 +
 17 STO d    58 FS? 49 "             08 MOVE
 18 SETGU    59 OFF     99 BEEP      09 30
 19♦LBL 00   60 CLA    100 PROMPT    10  E
 20 ,        61 RCL 01 101 GTO 00    11 3
 21 STO 03   62  E1    102♦LBL 13    12 CSIZEO
 22 266      63 X>Y?   103 XEQ "RS"  13 "rS"
 23 ,        64 " "    104 CF 03     14 SF 17
 24 191      65  E     105♦LBL 06    15 LABEL
 25 LIMIT    66 RCL 00 106 RCL 03    16 RCL 03
 26 4        67 MOVE   107 FC? 02    17 WHERE
 27 CSIZE    68 ARCL 01108 5         18 X<> T
 28 90       69 5      109 FS? 02    19 X<>Y
 29 LDIR     70 2      110  E        20 X>Y?
 30 PEN      71 PEN    111 +         21 STO 03
 31 "prgm ""72 SF 17   112 191       22 X>Y?
 32 ARCL 02 73 LABEL   113 X>Y?      23 SF 02
 33 "⊢""     74 ALENG  114 RDN       24 END
 34  E       75 *      115 STO 03    LBL'RS
 35 3        76 +      116 RCL 00    END        45 BYTES
 36 MOVE     77 RCL 00 117 2         LBL'CBPG
 37 SF 17    78 3      118 PEN       END        274 BYTES
 38 LABEL    79 PEN    119 +         .END.      03 BYTES
```

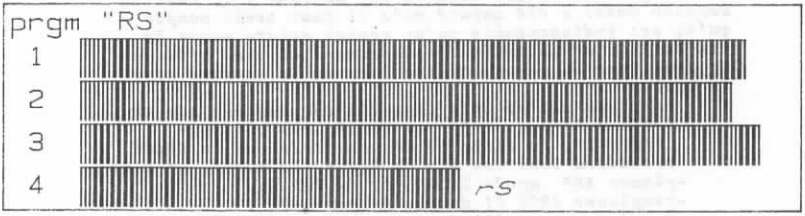
prgm "FULL" (maxi Bits a 1)

-1

Il faut noter que dans ce cas extrême (°), seul le dernier Octet a été amputé mais il faut tenir compte qu'il est indispensable qu'un espace existe entre la fin du RANG et le trait d'encadrement et ceci afin de ne pas créer de perturbations lors de la lecture avec le WAND.

(°) (bien que cet exemple pourrait se produire sur un RANG de N° à 3 chiffres) Remarquons que pour 'PRIVATE', je viens de m'apercevoir que le SPACE précèdera tous les N°s de RANGs. La modif. serait:
-placer ABS après le RCL 01 du Pas 61,
-remplacer ARCL 01 du Pas 68 par ARCL T
ce qui prouve qu'il faut essayer un PRGM dans toutes les configurations.....


prgm "CBPG" (facultatif)
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
rS


prgm "RS"
1
2
3
4
rS

```
01♦LBL "CBPG"
 02 4
 03 PSIZE
 04 1,16
05 "PRIV=CHS"
 06 PROMPT
 07 STO 01
08 "NOM PRGM"
 09 AON
 10 PROMPT
 11 ASTO 02
 12 PINIT
 13 ,
 14 BCP
 15 "P♦♦I♦♦"
 16 RCL [
 17 STO d
 18 SETGU
19♦LBL 00
 20 ,
 21 STO 03
 22 266
 23 ,
 24 191
 25 LIMIT
 26 4
 27 CSIZE
 28 90
 29 LDIR
 30 PEN
 31 "prgm ""
 32 ARCL 02
 33 "├""
 34  E
 35 3
 36 MOVE
 37 SF 17
 38 LABEL
 39 FC?C 01
 40 GTO 09
 41 CSIZE
 42 30
 43 X<>Y
 44 MOVE
 45 "TITRE"
 46 CF 23
 47 AON
 48 PROMPT
 49 AOFF
 50 FS? 23
 51 LABEL
 52 4
 53 CSIZE
54♦LBL 09
 55 7,13705
 56 STO 00
57♦LBL 01
 58 FS? 49
 59 OFF
 60 CLA
 61 RCL 01
 62 ABS
 63  E1
 64 X>Y?
 65 " "
 66  E
 67 RCL 00
 68 MOVE
 69 ARCL T
 70 5
 71 2
 72 PEN
 73 SF 17
 74 LABEL
 75 ALENG
 76 *
 77 +
 78 RCL 00
 79 3
 80 PEN
 81 -
 82 MOVE
 83 RCL 02
 84 RCL 01
 85 BCP
 86 BC
 87 STO 01
 88 X=0?
 89 GTO 13
 90 RCL 03
 91 WHERE
 92 X<> T
 93 X<>Y
 94 X>Y?
 95 STO 03
 96 ISG 00
 97 GTO 01
 98 XEQ 06
 99 "FEUILLE"
100 BEEP
101 PROMPT
102 GTO 00
103♦LBL 13
104 XEQ "RS"
105 CF 03
106♦LBL 06
107 RCL 03
108 FC? 02
109 5
110 FS? 02
111  E
112 +
113 191
114 X>Y?
115 RDN
116 STO 03
117 RCL 00
118 2
119 PEN
120 +
121 RCL 03
122 ,
123 ENTER↑
124 X<> Z
125 X<>Y
126 LOCATE
127 FRAME
128 FRAME
129 FC? 03
130 OFF
131 END
```

```
PRP "RS"
01♦LBL "RS"
 02 WHERE
 03 2
 04 ST+ Z
 05 RDN
 06 3
 07 +
 08 MOVE
 09 30
 10  E
 11 3
 12 CSIZEO
 13 "rS"
 14 SF 17
 15 LABEL
 16 RCL 03
 17 WHERE
 18 X<> T
 19 X<>Y
 20 X>Y?
 21 STO 03
 22 X>Y?
 23 SF 02
 24 END
```

```
CAT 1
LBL'CBPG
END
LBL'RS
END
.END.
```

```
261 BYTES
45 BYTES
09 BYTES
```

SYN-SWW

Dunkerque, le 15 OCT 84

D.DELEFORTRIE T.390
Adresse au 1 Novembre:
4 rue Mme Baleine
59181 STEENWERCK

Programme SYN-SWW

Programmes d'aide à la mise au point pour fondu-enchaîné.
Matériel: HP41 + module TIME

Le programme se compose de deux parties distinctes:

- SWW : partie enregistrement
  * initialisation et mise enroute du chronomètre (fonction SW)
  * allocation d'une SIZE suffisante :
  = la machine affiche 'N ?' : demande du nombre d'enre--gistrements et s'arrête ensuite sur le message ' C: ENRT'

  * L'enregistrement proprement dit peut commencer en appuyant sur la touche C : la machine affiche ' ENTER = SPLIT' = la mise en mémoire des temps partiels se réalise en appuyant sur ENTER
  * Une fois l'enregistrement terminé, ne pas oublier d'arrêter le chrono. = appuyer sur la touche B

- SYN : partie synchronisation-reproduction
  * mise en route : appuyer sur la touche A (message : A': RUN B : STOP) ; à chaque temps partiel arrivé à échéance, la machine émettra un BEEP
  * arrêter le chrono. après lecture : appuyer sur B

Ce programme m'est bien utile au cours de la phase de mise au point du fondu-enchaîné; il peut même servir à passer manuellement le F-E quand le matériel audio-visuel tombe en panne ...!

Très utile également quand on veut synchroniser précisèment sur une musique (évite pas mal de crises de nerfs..) Il est possible de modifier directement le registre concerné pour ajuster une 'interruption'.

Autre idée: on peut conserver les différents temps dans un fichier-data (avec un module X-Function).

Avec mes meilleurs voeux de programmation audio-visuelle!

```
12:06   15.10
01♦LBL "SYN"
02 SF 27
03 SIZE?
04 1
05 -
06 1 E3
07 /
08 STO L
09 TONE 2
10 TONE 2
11 TONE 2
12 "A: RUN B: STOP"
13 PROMPT
14♦LBL A
15 0
16 SETSW
17 RUNSW
18♦LBL 01
19 RCL IND L
20 RCLSW
21 FIX 4
22 VIEW X
23 X<=Y?
24 GTO 01
25 BEEP
26 FIX 0
27 VIEW L
28 ISG L
29 GTO 01
30♦LBL B
31 STOPSW
32 " STOPSW"
33 AVIEW
34 TONE 8
35 TONE 8
36 TONE 8
37 TONE 8
38 TONE 8
39 RTN
40♦LBL "SWW"
41 SIZE?
42 "  N ?"
43 PROMPT
44 X>Y?
45 PSIZE
46 " C: ENRT"
47 AVIEW
48 RTN
49♦LBL C
50 BEEP
51 "ENTER=SPLIT"
52 AVIEW
53 PSE
54 SW
55 .END.
```

Eric AUBOURG (T 366)
2 résidence St charles
27 rue Champ Lagarde
78000 VERSAILLES

Startrek : un grand classique des jeux sur ordinateur. Il est maintenant disponible sur HP 41, si vous avez 320 registres de libres et un module X FUNCTIONS.

Dans cette version STARTREK 2.Ø, le jeu se déroule dans une galaxie de 8x8 quadrants. Votre but est bien entendu, en temps que commandant de l'Enterprise, de détruire la flotte Klingon qui s'apprête à envahir la fédération galactique. Vous n'êtes bien sur pas seul dans cette glorieuse mission : des bases spatiales disséminées dans la galaxie vous permettent de vous ravitailler.

Voyons donc comment jouer: il faut d'abord taper le programme (ou le demander à Damien Debril). Je vous conseille de sauvegarder "initialisations" avec SF 11, puis "startrek" normalement. Il suffit alors de faire un memory lost (startrek prend 319 registres ...), de passer les cartes d'"initialisations" qui démarre, crée les fichiers XF nécessaires, et passe dans la Size adéquate. Il s'autodétruit ensuite, et vous pouvez charger startrek proprement dit. Celui ci va commencer par créer la carte de la galaxie. C'est long : faites autre chose pendant ce temps. Ensuite, la HP s'éteind : vous pouvez préparer le jeu à l'avance. Les commandes disponibles sont alors :

- A : Rapport général. Etat du vaisseau et du quadrant.
- B : Réglage du bouclier. Indispensable en présence de Klingons.
- C : Lancer d'une torpille. Votre stock en est limité ! Si il y a plusieurs klingons présents, un seul sera détruit, et les autres riposteront. Votre torpille peut d'ailleurs être absorbée par une étoile ... La probabilité en est proportionnelle au nombre d'étoiles dans le quadrant.
- D : Tir laser. Gourmand en énergie ... Le tir est automatiquement réparti entre les klingons présents. Ceux ci n'ont d'ailleurs pas de torpilles: ils vous tirent dessus au laser.
- E : Radar longue portée. Donne l'état des huit quadrants voisins sous la forme kbe : nombre de klingons, de bases, et d'étoiles. La sortie se fait en trois lignes.
- F : Déplacement. Vous indiquez votre destination sous la forme y,x, et le régime en quadrants/jour.
- G : Accostage à une éventuelle base, et ravitaillement en énergie et torpilles. Vous êtes alors protégé des tirs ennemis par le bouclier

```
01♦LBL "STARTRK"
"ALEA ?" PROMPT STO 00
XEQ 99 8 * E + INT
XEQ 99 8 * E + INT
E1 / + STO 01 3 E3
STO 03 E1 STO 04 CLX
STO 09 "M" 5 SF 25
CRFLD CF 25 CLFL "G"
02 SF 25 CRFLD CF 25
CLFL 9 SEEKPT ,063
STO [

43♦LBL 01
CLX STO 10 STO 02
XEQ 99 ,8 - X>0? E
X<=0? , ST+ 09 E2 *
ST+ 10 RDN ,15 -
X>0? E X<=0? ,
ST+ 09 E2 * ST+ 10
RDN ,03 - X>0? E
X<=0? , ST+ 09 E2 *
ST+ 10 XEQ 99 ,96 -
X>0? E1 X<=0? ,
ST+ 10 XEQ 99 8 *
INT 1 + RCL 10 +
SAVEX VIEW [ ISG [
GTO 01 XEQ 90 CLX
X<>F SF 27 OFF

105♦LBL "A"
"POSITION:" AVIEW PSE
"QUADRANT " RCL 01
FIX 1 ARCL X AVIEW
INT E - 2 * RCL 01
FRC 2 * INT + "N"
SEEKPTA GETREC RCL 01
FRC ,4 MOD "├ I"
X=0? "├V" ,2 X<=Y?
"├I" CLX ,3 X=Y?
"├I" PROMPT "ENERGIE:"
FIX 0 ARCL 03 PROMPT
"TORPILLES: " ARCL 04
PROMPT "BOUCLIER:"
RCL 04 FRC E4 *
ARCL X PROMPT
"TEMPS: " ARCL 02
PROMPT RCL 05
"ETAT DU QUADRAN" "├T:"
ARCL X PROMPT E2
X>Y? GTO 93
"ALERTE ROUGE" PROMPT
RCL 04 FRC 5 E-2
FC? 05 X<=Y? GTO 93
"BOUCLIER FAIBLE"
PROMPT GTO 93

179♦LBL "B"
FC? 06 GTO 00 "M" 1
SEEKPTA GETX X=0?
GTO 00 X>0? GTO 00
"BOUCLIER ENDOMM"
"├AGE" AVIEW GTO 93

194♦LBL 00
"BOUCLIER A:" RCL 04
FRC E4 * FIX 0
ARCL X AVIEW PSE
RCL 03 X<>Y ST+ 03 +
"E. DISPONIBLE:" ARCL X
"├ ?" PROMPT X<=Y?
GTO 02 RCL 04 INT
STO 04 GTO "B"

218♦LBL 02
ST- 03 E4 / RCL 04
INT + STO 04 GTO 93

227♦LBL "C"
RCL 05 E2 / INT
X=0? GTO 04 FC? 06
GTO 00 "M" 3 SEEKPTA
GETX X>0? GTO 00 X=0?
GTO 00
"TORPILLES ENDOM"
"├MAGEES." AVIEW
GTO 93

248♦LBL 00
RCL 04 E - X>0?
GTO 03
"PLUS DE TORPILL" "├ES"
AVIEW GTO 93

258♦LBL 03
"TORPILLES." AVIEW
STO 04 FC? 06 GTO 13
"M" , SEEKPTA GETX
X>0? GTO 13 X=0?
GTO 13 XEQ 99 ,2
X<=Y? GTO 13
"ORDINATEUR E"
"├NDOMMAGE: TI" AVIEW
"├R RATE" ASHF AVIEW
GTO 91

283♦LBL 13
XEQ 99 ,95 RCL 05 10
MOD Y↑X X<=Y? GTO 05
"KLINGON DETRUIT" AVIEW
E2 ST- 05 RCL 09 E
- X=0? GTO 95 STO 09
GTO 91

303♦LBL 04
"AUCUN KLINGON D"
"├ETECTE" AVIEW GTO 93

308♦LBL 05
"UNE ETOILE S'IN"
"├TERPOSE." AVIEW
GTO 91

313♦LBL "D"
CF 07 FC? 06 GTO 00
"M" , SEEKPTA GETX
X<0? SF 07 GETX GETX
X>0? GTO 00 X=0?
GTO 00
"LASERS ENDOMMAG"
"├ES." AVIEW GTO 93

333♦LBL 00
RCL 05 E2 / INT
X=0? GTO 04
"LASERS POINTES." AVIEW
PSE "E. DISPONIBLE:"
RCL 03 ARCL X AVIEW
"PUISSANCE DU TI"
"├R ?" PROMPT X>Y?
GTO "D" - STO 03
LASTX -FS? 07 XEQ 99
FS? 07 * RCL 05 E2
/ INT ST/ Y 6 + E3
/ 7 + STO 10 RDN
STO 12

373♦LBL 06
XEQ 99 * RCL IND 10 -
CHS STO IND 10 X>0?
GTO 08
"*** KLINGON DET"
"├RUIT" AVIEW E2
ST- 05 RCL 09 E -
X=0? GTO 95 STO 09
RCL 10 INT 8 X=Y?
GTO 08 CLX 7 X=Y?
GTO 07 RCL 07 STO 06

404♦LBL 07
RCL 08 STO 07

407♦LBL 08
RCL 12 ISG 10 GTO 06
GTO 91

412♦LBL "E"
"M" , SEEKPTA GETX
X>0? GTO 00 X=0?
GTO 00
"RADAR ENDOMMAGE" AVIEW
GTO 93

424♦LBL 00
"RADAR LONGUE PO"
"├RTEE" AVIEW "G" CLX
SEEKPTA FS? 05 32
X<>F RCL 01 FRC ,1
X=Y? SF 01 CLX ,8
X=Y? SF 03 LASTX INT
E X=Y? SF 00 CLX 8
X=Y? SF 02 FIX 0 "!"
RCL 11 9 - FC? 00
FS? 01 SF 04 FS? 04
"├***" SEEKPT GETX
FC?C 04 ARCL X "├:"
FS? 00 "├***" GETX
FC? 00 ARCL X "├:"
FC? 03 FS? 00 SF 04
FS? 04 "├***" GETX
FC?C 04 ARCL X "├!"
AVIEW "!" RCLPT 5 +
SEEKPT FS? 01 "├***"
GETX FC? 01 ARCL X
"├:" GETX ARCL 05
"├:" GETX FS? 03
"├***" FC? 03 ARCL X
"├!" STOP AVIEW "!"
RCLPT 5 + SEEKPT
GETX FC? 02 FS? 01
SF 04 FS? 04 "├***"
FC?C 04 ARCL X "├:"
GETX FS? 02 "├***"
FC? 02 ARCL X "├:"
GETX FC? 02 FS? 03
SF 04 FS? 04 "├***"
FC?C 04 ARCL X "├!"
STOP AVIEW STOP
GTO 93

538♦LBL "F"
CF 07 "M" 4 SEEKPTA
GETX X<0? SF 07
"VOUS ETES EN " FIX 1
ARCL 01 AVIEW RCL 11
"G" SEEKPTA RCL 05
SAVEX "QUADRANT D'ARRI"
"├VEE=?" PROMPT X<0?
GTO "F" INT 8 X<Y?
GTO "F" LASTX FRC ,8
X<Y? GTO "F" LASTX
STO 10 INT RCL 01 INT
- RCL 10 FRC RCL 01
FRC - 10 * R-P
"REGIME ?" PROMPT X<0?
GTO "D" FC? 07 GTO 00
"REGIME MAX ,2" ,2
X>Y? PROMPT RDN

594♦LBL 00
/ ST+ 02 LASTX 1 E2
* RCL 03 X<Y? GTO 09
X<>Y ST- 03 RCL Z
X<> 10 STO 01 RCL 02
30 X<Y? GTO 94 CF 05
FS? 06 XEQ 70 XEQ 90
GTO 93

617♦LBL 09
"PAS ASSEZ D'ENE"
"├RGIE" AVIEW RCL 04
FRC E3 * + X<Y?
GTO 93 "VOUS POUVEZ "
"├BAISSER LE B" AVIEW
ASHF ASHF "├OUCLIER"
AVIEW PSE GTO 93

637♦LBL "G"
RCL 05 E2 MOD 10 /
INT X≠0? GTO 10
"AUCUNE BASE PRE"
"├SENTE." AVIEW GTO 93

650♦LBL 10
"ACCOSTE" AVIEW 3 E3
STO 03 10 STO 04
SF 05 GTO 93

659♦LBL "H"
"RAPPORT AVARIES" AVIEW
"M" CLX SEEKPTA "MH"
SEEKPTA FIX 1

668♦LBL 15
"M" RCLPTA SF 25 GETX
FC?C 25 RTN "MH"
RCLPTA GETREC "├ : "
ARCL Y AVIEW GTO 15

682♦LBL 99
RCL 00 9821 * ,211327
+ FRC STO 00 RTN

691♦LBL 90
RCL 01 INT 8 *
RCL 01 FRC 10 * +
STO 11 "G" SEEKPTA
GETX STO 05 E3
STO 06 STO 07 STO 08
RTN

711♦LBL 91
FS? 05 GTO 93 RCL 05
E2 / INT X=0?
GTO 93 E3 / E +
STO 10

725♦LBL 11
RCL 10 5 + RCL IND X
XEQ 99 X<>Y *
STO IND Y LASTX - CHS
"TIR ENNEMI:" FIX 0
ARCL X "├ UNITES"
AVIEW RCL X RCL 04
FRC E4 * - CHS
X<0? GTO 92
"BOUCLIER:" ARCL X
AVIEW E4 / RCL 04
INT + STO 04 RDN
XEQ 99 ,6 X>Y? GTO 00
RCL Z AHUM / XEQ 99
4 * INT "M" SEEKPTA
X<>Y GETX - CHS X<>Y
SEEKPTA X<>Y SAVEX
X>0? GTO 00 "MH" X<>Y
SEEKPTA GETREC
"├ ENDOMMAGES" AVIEW
SF 06

791♦LBL 00
ISG 10 GTO 11 GTO 93

795♦LBL 92
"L'ENTERPRISE ES"
"├T DETRUIT" AVIEW PSE

800♦LBL 12
"IL RESTAIT " RCL 05
ARCL X "├ KLINGON" E
X<Y? "├S" AVIEW RTN

810♦LBL 93
"READY." AVIEW RTN

814♦LBL 94
"FIN DE VOTRE MI"
"├SSION." AVIEW GTO 12

819♦LBL 95
"FELICITATIONS" AVIEW
"VOUS AVEZ DETRU"
"├IT TOUS" AVIEW
"LES VAISSEAUX K"
"├LINGONS." AVIEW
"EN " ARCL 02
"├ JOURS SIDERAU" "├X"
AVIEW RTN

834♦LBL 70
CF 06 "M" CLX SEEKPTA

839♦LBL 71
SF 25 GETX FC?C 25
RTN X<0? FS? 30
GTO 71 RCL 10 + RCLPT
1 - SEEKPT RDN SAVEX
X<0? SF 06 GTO 71
```

de la base.

- H : Rapport des avaries : un nombre négatif indique une avarie, et le nombre de jours nécessaires pour réparer. Les effets d'une panne sont généralement un non fonctionnement total sauf pour les machines, qui autorisent un régime maximum de 0.2 q/j, et le bouclier, qui continue à vous protéger dans la mesure de sa charge, mais que vous ne pouvez plus recharger.

Munis de ces quelques renseignements, vous pouvez commencer à jouer. Une dernière précision: une carte de la galaxie me parait indispensable. Consignez-y toutes les observations radar. Bon amusement et

Heureuse Programmation

T 366

Subong

-INVERSE DRAPEAUX

Un classique, adapté ici pour module IL-Development. Id inverse l'état du drapeau 00 ≤ n ≤ 55. Le pas 02 sauve X, qui est rétabli au pas 25; Y et Z sont préservés aussi; T contient INT($\frac{n}{8}$)-7.

Si vous utilisez Id en sous-programme, les pas 02 et 23 peuvent être adaptés selon ce que vous voulez conserver de la pile.

TREDEZ b. T 120

```
01♦LBL "Id"
02 STO \
03 STO [
04 8
05 MOD
06 7
07 -
08 X<> \
09 X<> d
10 X<> [
11 8
12 /
13 INT
14 7
15 -
16 A-X%
17 X<>FLAG
18 FC?C IND \
19 SF IND \
20 X<>FLAG
21 LASTX
22 Y-A%
23 RDN
24 X<> [
25 X<> d
26 CLA
27 .END.
```

PUISSANCE

BERTRAND Yves
19, rue Principale
MURBACH
68 530 BUHL
T526

Le programme de Jacques DEFENDINI - MICRO-REVUE, N°1, page 28, concernant l'élévation d'une matrice carrée à une puissance quelconque, comporte une erreur : il utilise les registres B, C, D (103, 104, 105) pour stocker les indices i, j, k. Or, pour le calcul de A^n, A étant une matrice d'ordre 10, ces mêmes mémoires contiennent 3 coefficients de la matrice. Elles font donc double emploi et le programme donne des résultats erronés pour n = 10.

Je vous propose donc une autre version de ce programme : elle est environ 57 % plus rapide et n'utilise que 234 octets (au lieu de 357) et accepte des matrices d'ordre inférieur ou égal à 11.

Utilisation du programme :

- XEQ "P"
- la HP-41 C (V) demande EFFACEZX-F ; ne l'effacer que lors de la première utilisation
- répondre aux demandes successives de la HP-41 ;

elle met environ 25,2 s (contre 59,5) pour calculer

$$\begin{pmatrix} 1 & 2 & 5 \\ 3 & 4 & \end{pmatrix}$$

Ne possédant pas d'imprimante, je n'ai pas conçu ce programme pour qu'il imprime les C (i,j), mais 4 ou 5 lignes supplémentaires combleront aisément cette lacune.

```
   LBL "P"
   251
   PSIZE
   "EFFACEZX-F"
05 PROMPT
   CLST
   FS ? 39
   SIGN
   FS ? 38
10 2
   FS ? 37
   4
   FS ? 36
   8
15 +
   +
   STO 00
   121
   STO 04
20 "PUISSANCE = ?"
   SF 25
   CRFLD
   PROMPT
   DSE X
25 STO 08
   9,25
   STO 01
   STO + 04
   STO 03
30 "ORDRE = ?"
   PROMPT
   STO a
   STO 02
   STO 05
35 STO 06
   E3
   /
   ISG X
   STO 07
40 ENTER↗
   FIX 0
   CF 29
   LBL 02
   "A"
45 ARCL Y
   " ⊢,"
   ARCL X
   " ⊢= ? "
   PROMPT
50 STO IND T
   RDN
   ISG Z
   ISG X
   GTO 02
55 CLX
   RCL 07
   ISG Y
   GTO 02
   9,130121
60 REGMOVE
   LBL 10
   RCL 03
   RCL 04
   STO N
65 CLX
   LBL 00
   RCL IND Y
   RCL IND N
   X
70 +
   RCL a
   STO + N
   SIGN
   STO + Z
75 RDN
   DSE 02
   GTO 00
   SAVEX
   R↗
80 STO + 04
   LAST X
   STO 02
   DSE 05
   GTO 10
85 STO 05
   STO-04
   STO + 03
   DSE 06
   GTO 10
90 STO 06
   CLX
   SEEKPT
   RCL 04
   GETRX
95 CLX
   SEEKPT
   RCL 01
   STO 03
   DSE 08
100 GTO 10
   RCL 07
   ENTER↗
   LBL 01
   FIX 0
105 "A"
   ARCL Y
   " ⊢,"
   ARCL X
   " ⊢ ="
110 FIX IND 00
   ARCL IND 04
   PROMPT
   ISG 04
   ISG X
115 GTO 01
   CLX
   RCL 07
   ISG Y
   GTO 01
120 END
```

Heureuse Programmation

Bertrand

BERTRAND Yves

19, Rue Principale

68530 MURBACH

Le 26 Août 1984

======

T 526

Voici un programme ludique pour HP-41C qui allie le hasard à la déduction : un champ de mines est représenté par un carré de 10 x 10 = 100 cases où une mine occupe une case et peut avoir comme coordonnées 0,0 à 9,9, avec un maximum de 11 mines. Vous disposez d'un engin anti-mines qui comporte deux options :

- Vous tirez dans le champ de mines, mais le projectile n'atteint aucune mine (cas le plus fréquent) ; la HP-41 vous indique le nombre de mines détectées par le projectile. Pour qu'une mine soit détectée, il faut qu'elle se trouve dans une des 8 directions suivantes : nord, sud, est, ouest, nord-est, nord-ouest, sud-est, sud-ouest, par rapport au projectile.

- Vous tirez dans le champ de mines et le projectile détruit une mine ; la HP-41 affiche alors TOUCHE, élimine la mine en question de sa mémoire et indique également le nombre des mines détectées.

Lorsque toutes les mines sont détruites la HP-41 affiche FIN, XX TIRS, XX étant le nombre de tirs.

But du jeu : Vous l'avez peut-être deviné, il s'agit tout simplement de détruire toutes les mines avec le moins de tirs possibles.

Utilisation du programme :

- XEQ "MINES"

- Lorsque la HP-41C demande MINES ? entrez un nombre entier inférieur ou égal à 11 et pressez sur R/S.

- Lorsque la HP-41 affiche PRET introduisez les coordonnées de votre tir sous la forme suivante : exemple : pour 9,4 appuyez sur 94 R/5 puis attendez l'affichage du nombre de mines détectées (et éventuellement TOUCHE). Pour les tirs suivants : XY R/S.

N.B. Le module X - functions est necessaire

PROGRAMME

```
 1 LBL "MINES"
   SIZE ?
   17
   X> Y ?
 5 PSIZE
   CLRG
   CLX
   STOd
   FIX 0
10 SF 26
   RCL FLAG
   STO 16
   TIME
   HR
15 FRC
   PROMPT
   "MINES?"
   STO 12
   STO 15
20 FIX 1
   LBL 07
   X<>Y
   PI
   X
25 FRC
   RCL X
   E1
   X
   RND
30 STO IND Z
   R↗
   DSEX
   GTO 07
   "PRET"
35 FIX 0
   LBL 00
   PROMPT
   "TIR "
   ISG 14
40 ADV
   ARCL 14
   " ⊢: "
   ARCL X
   AVIEW
```

```
45 E1
   /
   STO 00
   RCL 16
   STOFLAG
50 RCL 12
   STO a
   LBL 01
   RCL IND a
   x< 0 ?
55 GTO 09
   RCL 00
   x = y ?
   GTO 08
   INT
60 x<>y
   INT
   -
   RCL 00
   FRC
65 RCL IND a
   FRC
   -
   E1
   X
70 R→P
   x<>y
   360
   MOD
   45
75 /
   12
   +
   FRC
   x = o ?
80 FS ? IND L
   GTO 09
   SF IND L
   ISG 13
   LBL 09
85 DSE a
   GTO 01
   CL X
   x<>13
   "MINES :_"
90 ARCL X
   GTO 00
   LBL 08
   "TOUCHE"
   TONE 9
95 AVIEW
   CHS
   STO IND a
   DSE 15
   GTO 09
100 "FIN, "
   ARCL 14
   " ├ TIRS"
   PROMPT
   END
```

Heureuse Programmation

Bertrand

28/3/84

Le programme "Nb" est inspiré de "PD" du PPC ROM qui mesure le nombre d'octets séparant le pointeur en RAM, du .END. Placer le pointeur à l'endroit voulu d'un programme, faire RCL b (assigné) puis XEQ"Nb. Le contenu de Y est conservé, vous pouvez donc recommencer et faire la soustraction pour connaitre la distance entre deux points d'un programme (entre GTO et LBL par exemple).

TREDEZ b. T 120

```
13:02  28.03
01♦LBL "Nb"
02 "*"
03 X<> [
04 X<> \
05 ASHF
06 CLX
07 ATOX
08 ATOX
09 X<>Y
10 STO [
11 16
12 ST/ [
13 MOD
14 LASTX
15 X↑2
16 *
17 +
18 7
19 *
20 RCL [
21 INT
22 +
23 CLA
24 .END.
```

LAVILLE Laurent (T.634)
45 Rue Riquet
75019 PARIS

Cher Jean-Daniel,

Pour ma première participation au club je vous soumets un article concernant le programme F? du PPC ROM.

En lisant les anciens numéros de PPC-T et plus particulièrement le n°3, je me suis apperçu que le programme F? faisait du dégat en perturbant les alarmes, ce qui le rendait inutilisable. Aussi je vous propose ma version en langage utilisateur de ce programme qui ne perturbe en aucun cas les alarmes et les assignations. Bien sur me direz vous, il existe maintenant un programme en microcode qui effectu le travail (PPC-T n°3 P29), mais tout le monde n'a pas la chance et les moyens financier pour acquérir et utiliser le matériel adéquat.

Ma version de F? utilise les routines 2D, QR et PD parus dans MR n°3 P30. En voici la liste accompagnée des commentaires.

```
01  LBL "F?"
02  RCL c
03  XEQ "PD"      Ces lignes servent à décoder l'adresse du .END.
04  LASTX
05  7
06  /
07  511
08  X<>Y          Nombre de registres déjà utilisés
09  -
10  320           Nombre de registres restant sans compter ceux
11  X<>Y          utilisés pour les alarmes et les assignations.
12  -
13  SIZE?         Ajoute le nombre de registres alloué aux données
14  +
15  LBL 01
16  DSE X
17  SF 25         Boucle qui a pour effets de supprimer le nombre
18  PSIZE         de registres alloué aux alarmes et assigations à
19  FC?C 25       celui calculé aux lignes 10 à 12.
20  GTO 01
21  LASTX
22  PSIZE         Restitu la partition originale entre données et
23  -             programmes.
24  END
```

Bonne programmation à tous

APPLICATIONS

Faisons une excursion dans les registres disponibles situés entre l'adresse absolue ØCØh (192d) et le .END. au-dessus duquel figurent les programmes puis, à partir du "rideau" (on désigne par rideau, en anglais "curtain", la séparation entre les programmes et les registres de données; en fait donne l'adresse absolue du registre de données RØØ), les données.

Cet espace est occupé, à votre initiative, par les assignations de fonctions (l'ensemble des assignations étant couramment appelé: table d'assignation) et les tampons - ou plus exactement les mémoires-tampon. Un tampon (en anglais: buffer) est une sorte de bloc-notes réservé à l'usage exclusif d'un périphérique où sont stockées les informations dont il a besoin pour tenir son rôle; ils peuvent être soit internes aux périphériques (cas des imprimantes ou du lecteur optique, par exemple) soit externes (cas le plus fréquent pour les modules); pour ceux-ci, c'est dans cet espace qu'ils figurent, mais un tampon ne peut être créé et maintenu qu'en présence du module en assurant le contrôle.

Pour permettre à la HP de les reconnaître, le premier registre du tampon - celui situé le plus bas, vers ØCØ -appelé en-tête (ang.: header) débute par un indicatif identifiant l'usage du tampon; pour les tampons connus, ces indicatifs sont: (en hexa)

AA pour les alarmes du module TIME
BB pour le module PLOTTER
CC pour le module HP-IL DEV

Hormis le tampon du PLOTTER dont la taille est fixe (26 registres), les autres ont une taille variant suivant le nombre d'alarmes entrées, pour le module TIME, ou au gré de l'utilisateur, pour le module HP-IL DEV. Cette taille figure dans l'en-tête , en hexadécimal, dans l'octet suivant l'indicatif du tampon; la taille des tampons est ainsi limitée en fait à 254 registres, dont le registre d'en-tête. Ceci permet à la HP de sauter aisément d'un en-tête de tampon à l'autre jusqu'à trouver l'indicatif qu'elle recherche.

Ces tampons sont accumulés dans l'ordre chronologique de création, le premier tampon occupant le premier registre disponible à partir de ØCØh. Ceci va à l'encontre de l'idée communément admise, suggérée par les informations portées sur les schémas de la mémoire vive, qui rangent toujours les alarmes avant les autres tampons (à commencer par "Au fond" -fig. 35 - de notre cher Président). S'il n'existe à ce moment aucune assignation de fonctions, le registre d'en-tête du tampon occupera donc le registre ØCØh. Si ultérieurement on veut créer une table d'assignation, celle-ci commencera toujours en ØCØh: les registres de celle-ci repousseront donc tous les autres tampons (et les registres d'assignation existants au fur et à mesure des besoins) vers le .END., tant qu'il y aura de la place disponible; sinon, il s'affichera un

PACKING, TRY AGAIN	pour les assignations et le module HP-IL DEV
NO ROOM	pour les alarmes
PL: NO ROOM	pour le module PLOTTER

Si l'on supprime un tampon, par annulation ou par enlèvement du module,(1) les tampons supérieurs - créés ultérieurement - seront automatiquement descendus à l'allumage pour combler le vide. Comme le contrôle "tampon-module contrôleur" est effectué à l'allumage, vous pouvez fort bien, la HP restant éteinte, déplacer le module contrôleur sans disparition du tampon.

Assignations. Une assignation de fonction n'occupe que 3 octets, la moitié d'un registre; La Palisse dirait alors qu'on peut en stocker 2 par registre: c'est une explication plus claire que le texte ambigu figurant à la fin de l'annexe E du manuel HP 41!

(1) C'est sans doute le fait qu'on ne pouvait enlever le module TIME d'une 41 CX qui a fait introduire la fonction CLRALMS sur ce modèle.

Un registre d'assignation se présente comme suit:

octet				
6	FØ	Indicatif des registres d'assignation de fonction		
5	LBL Ø3	(assignation normale)	code à 2 octets de la fonction	(ass. synth.)
4	f	code de la fonction		
3	k	code de la touche	code de la touche	
2	LBL Ø3	autre assignation		
1	f'			
Ø	k'			

Dans un registre nouvellement ouvert, la première assignation occupe les octets Ø à 2, la suivante les octets 3 à 5.

Lorsqu'une fonction est désassignée, son code de touche est mis à Ø mais son code de fonction reste.

Lors de l'assignation d'une nouvelle fonction, la HP recherche dans la table si une assignation a le même code de touche: si oui, elle la remplace par la nouvelle assignation; si non, elle recherche à partir de ØCØ si un code de touche est à Ø: si oui, elle remplace l'assignation annulée par la nouvelle. Il ne peut donc y avoir 2 fonctions assignées à la même touche (sauf programmation synthétique), seule la dernière subsiste; par ailleurs, une assignation nouvelle peut se trouver n'importe où dans la table.

Si, dans un registre, une assignation reste seule active, le registre est laissé en l'état; si, par contre, les 2 assignations du registre ont été annulées, celui-ci est supprimé soit à l'allumage soit par PACK, et les registres supérieurs ainsi que les tampons sont descendus vers ØCØ.

Mise à zéro de la table d'assignation.

Si vous possédez un lecteur de cartes, la procédure est la suivante: assignez une fonction à une touche quelconque, annulez la et, sans éteindre, enregistrez une carte par WSTS. Le lecteur demandera les 2 pistes: passez alors 2 fois la même piste.

Pour annuler la table, passez la carte en pressant + lorsque le lecteur vous réclamera la première piste, éteignez la HP: la table d'assignation sera vide à l'allumage.

Il importe d'entrer et d'annuler une fonction; si vous réalisiez l'opération avec une table vide, le lecteur ne vous demanderait qu'une piste, et, à la lecture, la table d'assignation resterait inchangée.

Si vous ne possédez pas de lecteur mais une imprimante, listez par PRKEYS les assignations et désassignez à la main toutes les assignations de fonction. Enfin, démuni de lecteur et d'imprimante - la vraie misère! -, il ne vous restera plus qu'à essayer toutes les touches (shiftées et non shiftées) pour désassigner, toujours à la main, les assignations de fonction qui apparaitront.

Voici rassemblées nombre d'informations concernant ces registres disponibles. Les renseignements fournis ici ont été contrôlés, en particulier avec le ZENROM dont le mode RAMED (RAM Editor) a été fort précieux.

Jean THIBERGE (10292-T165)

Après quelques sueurs froides dues à ma chère HP 41, je voudrais épargner de telles angoisses aux autres HPistes de PPC-T. En effet, sur les nouvelles HP, la plaque processeur n'est plus boulonnée mais simplement posée sur la plaque clavier. Moralité : lors d'un choc, celle-ci bouge légèrement. Si elle a beaucoup bougé, la HP ne marche plus du tout. Sinon, si il y a seulement un faux-contact intermittent, votre HP se met à dérailler complètement (par exemple vous faites tranquillement un petit calcul, et en pleine introduction numérique, elle affiche des caractères incongrus...) . Pas de panique, le remède est simple : enlevez batteries et extensions et opérez. Il suffit de presser légèrement sur la plaque processeur pour la remettre bien en place. Appliquez le même traitement au connecteur d'extensions : il a aussi tendance à bouger. Ensuite, remontez et revissez (à fond, mais sans forcer). Normalement, votre HP doit marcher à nouveau.

Heureuse programmation ...

T 366

Eric AUBOURG (T 366)
12 rue des quatre vents
79000 Niort

GTO 2

GTO / 2 octets (suite)

Un adhérent de PPC-PC m'ayant téléphoné pour me demander des précisions sur les subtilités des GTO à des LBL codés sur 1 octet, je lui ai répondu que la solution nécessitait un assez long développement qui ne pouvait se faire convenablement que par écrit et que j'écrirai donc un "papier" sur le sujet qu'il trouvera dans la revue "MR" de PPC-T. Je dois préciser que le fait que je n'adhère plus au chapitre de Paris justifie cette façon d'agir et j'espère que vous ne souhaitez pas que j'épilogue sur les raisons qui m'ont guidé à me désolidariser d'une équipe avec laquelle je ne me sentais aucune affinité.

Le "papier" précité a été intitulé "GTO codés sur 2 octets" *(cf. MR N°3P29) et n'avait pour but que de répondre à une question posée, réponse qui pouvait, je pense, intéresser d'autres lecteurs.*

Pour simplifier le contenu du papier précité, j'ai volontairement fait une sorte d'impasse sur la valeur retournée en 'X' par le PRGM "PD" de la 'PPC-ROM' en écrivant qu'il s'agissait de l'adresse DEC absolue. Cette assertion peut en fait être considérée comme une ânerie (j'avais prévenu!) mais une explication détaillée aurait demandé, comme vous allez le voir, un développement dont l'utilité ne me semblait pas indispensable. Nous ne désirions que calculer le plus facilement possible, le nombre d'octets qui séparent "la fin du GTO au début du LBL".

Notre adhérent de PPC-PC me retéléphone suite à la parution du "papier" dans MR (merci de nous avoir lus et j'espère ton adhésion!) pour me signaler son désarroi car on venait de lui indiquer, non pas que j'avais écrit une ânerie (ou des âneries, à la rigueur) mais que tout ce que j'avais écrit était parfaitement faux et je vous épargne les qualificatifs diffamants dont on m'a affublé. Tout ceci n'a eu pour seul effet que me faire sourire car il est évident qu'à mon âge, je connais assez bien ce genre de comportement et le but recherché.... Comme l'a écrit fort judicieusement notre Président : " Des gamineries. " (MR N°2P122).

Revenons à des choses beaucoup plus intéressantes!

L'adresse du pointeur PRGM est placée dans le REG 'b' et nous allons voir d'une part, de quelle façon et, d'autre part, quelle est la relation qui existe avec la valeur placée en 'X' par la routine "PD".

Si nous reprenons l'exemple qui nous donnait 1781 avec le pointeur sur LBL 11, un RCL b *suivi d'un* XEQ "DECODE" *ou* "NH" *(pour ceux qui disposent de la PPC-ROM) nous placerait en ALPHA la valeur HEX 3ØFE.* "DECODE" *(ou* "NH"*) convertit (décode) le NNN placé en 'X' par* RCL b *en son équivalent HEXadécimal (NNN = Nombre Non Normalisé). Or HEX ØFE = DEC 254 et 1781 MODulo 254 nous donne la valeur 3 qui est le 1er DIGit indiqué dans le REGistre* b *. Une autre façon permet d'écrire que 1781 = (254 * 7) + 3 . Si nous opérons*

de la même façon au niveau du GTO 11 qui nous donnait 177Ø avec "PD", *nous obtenons HEX 6ØFC où ØFC = DEC 252 et 177Ø MOD 7 nous donne 6. Nous allons voir ce que tout cela donne d'une façon que j'espère plus concrète. Représentons les octets et les registres qui correspondent à notre exemple :*

7	6	5	4	3	2	1	7	6	5	4	3	2	1	7	6	5	4	3	2	1
					1	2	3	4	5	6	7	8	9	10						
			254	LBL 11			←			253				→	GTO 11			252		

Les chiffres de la 1ère ligne ci-dessus indiquent, de droite à gauche, les numéros de 1 à 7 de chaque octet, ces octets étant groupés par RECistre.

La 2ème ligne de chiffres indique les 1Ø octets occupés par les instructions situées entre le LBL 11 *et le* GTO 11.

La 3ème ligne indique l'adresse absolue DEC de chaque octet.

Remarquons que LBL 11 *se trouve sur l'octet '3' du REG 254 (3 ØFE...) et que le 1er octet de* GTO 11 *est le N° 6 (6ØFC).*

J'espère qu'un examen attentif de ce croquis va vous permettre de comprendre le système de répartition des octets et des REG dans la RAM.

Le s/PRGM "2D" *de la* PPC-ROM *décode 2 octets en DECimal. En reprenant notre exemple, si nous replaçons le pointeur PRGM sur le* LBL 11, *qu'en mode RUN nous XEQécutons* RCL b *(ASN à une touche ou avec la* ZENROM 1) *puis XEQ* "2D", *nous obtenons en 'X' la valeur DEC 48 qui est égale à HEX 3Ø. Si ensuite, toujours en mode RUN, nous XEQécutons* RCL M, *nous obtenons en 'X' DEC 254 = HEX FE et nous retrouvons ainsi le 3ØFE que nous avions obtenu avec* RCL b *suivi de* DECODE.

Le s/PRGM "QR" *(Quotient Remainder, Remainder signifiant: "le reste", je crois?)convertit une valeur DEC en HEX mais d'une façon assez particulière mais néanmoins intéressante. Je pense qu'un exemple sera plus explicite qu'une description quelconque. Reprenons notre valeur DEC 254 obtenue ci-avant :* 254 ENTER↑ 16 XEQ "QR" *placera 14 en 'X' et 15 en 'Y'. Or DEC 15 = HEX F et DEC 14 = HEX E et nous retrouvons ainsi HEX* FE = *DEC* 254.

J'espère que toutes ces indications vont vous permettre d'analyser ligne par ligne, ce qui se passe dans le PRGM "PD" *et dans les routines associées.*

Je termine en signalant qu'il n'est plus indispensable d'utiliser la FS RCL b *(en mode RUN donc, non PRGMmable) si nous disposons de la* "MLEPROM 1H" *écrite en MC par Stéphane BARIZIEN car il suffit alors de faire:* 12 CHS NRCL DECODE *(tout ceci étant PRGMmable) pour obtenir en ALPHA la valeur HEX contenue dans le* REG b *dont l'adresse DEC absolue est 12, et ceci sans que le pointeur PRGM ait sauté de RAM en ROM, si nous opérons en mode RUN le pointeur PRGM ne bougera pas d'un octet. MERCI Stéphane ! Je me sens tout de même obligé de signaler que la* ZENROM 1 *permet d'obtenir le même résultat et que je me régale avec cette ROM (à suivre...).*

J'ai conscience que je n'ai pas encore tout dit sur le REG b *et j'attends que vous me disiez si je dois continuer dans cette voie ou si vous préférez que je laisse la place à d'autres. Vos encouragements seraient éventuellement les bienvenus car, comme je l'écris souvent, c'est plus facile à comprendre qu'à expliquer. Merci de votre attention.* *RS* (T178)

PHYSIQUE

Cher Jean-Daniel,

Pour continuer la description de la rubrique "Sur le marché", je t'envoie un résumé du livre d'applications "PHYSIQUE".

Ce manuel est assez semblable au manuel du X-F de par ses dimensions, il comporte environ 70 pages plus 20 pages de codes-barres (bande de petits veinards!). Je vais essayer de reprendre ici les caractéristiques des différents programmes.

1) Radiation thermique d'un corps noir: calcule la longueur d'onde pour un maximum de puissance émise à une température donnée et calcule la puissance émise.
2) Caractéristiques d'un trou noir: calcule la température, le rayon et la durée de vie d'un trou noir.
3) Conversions de relativité: conversions entre la masse au repos, la vitesse, l'énergie et le moment.
4) Relativité spéciale en 3 dimensions: calcule les temps, les facteurs de dilatation et les transformations de Lorentz.
5) Paradoxe des jumeaux d'Einstein (connu également sous le nom de "Paradoxe des jumeaux de Langevin", celà dépend si l'on est chauvin ou non...) : calcule les temps relatifs et réels basés sur la transformation de Lorentz.
6) Simulation d'orbite Delta-V : calcule les paramètres d'une orbite lorsque la position initiale et la vitesse sont connues.
7) Equations du mouvement: calcule les différents paramètres pour l'équation d'un mouvement rectiligne uniformément accéléré.
8) Correction du chevauchement des isotopes: calcule le chevauchement de 2 isotopes radioactifs dans un spectomètre à scintillation.
9) Formule semi-empirique de la masse nucléaire: calcule l'énergie de liaison et la masse (résultats approximatifs).
10) Coefficient de Clebsch-Gordon: utilise la formule de Racah pour évaluer le couplage de 2 états de moment angulaire.
11) Calcul des désintégrations nucléaires suivant les dates.

Une 41C suffira pour la majorité de ces programmes, seuls les programmes 4, 6 et 9 nécéssitent un module mémoire supplémentaire. L'appendice contient les codes-barres et le tout est déjà perforé afin de prendre place dans un classeur à 3 anneaux.

<u>Qu'en penser?</u> Le niveau de ces programmes est fort variable, le n°7 par exemple, peut très facilement être résolu par un lycéen (celà ce fait les 2 doigts dans le nez, et jusqu'au coude!). Quant au n°2, je n'ai pas encore compris son utilité dans les applications quotidiennes.
On pourra se demander, à juste titre d'ailleurs, si le prix est bien justifié (plus de 1000 FB) pour des programmes de cette qualité.
Pour ce même prix, j'ai acquis un livre (de physique) d'environ 500 pages en anglais (yes, sir!) qui répond mieux à mes besoins plutôt qu'une collection hétéroclite de programmes de physicien à l'eau de rose... Ceci dit, si vous avez de l'argent à placer, les produits HP augmentent rapidement...

Tout autre chose maintenant avec des suggestions pour le microcode:
- SST2 et BST2 agissent exactement comme SST et BST mais continuent à agir lorsque l'on maintient la touche enfoncée (comme sur une machine de la série 10).
- XM<>MM : échange de 2 programmes, l'un se trouvant en mémoire principale, l'autre en mémoire étendue, les 2 programmes n'ayant pas obligatoirement la même longueur. C'est décevant d'avoir parfois sa machine pleine à craquer et de devoir supprimer un programme dans la mémoire pricipale avant de pouvoir faire GETP. Messages d'erreur: TYPE ERROR, NONEXISTENT, XM TOO LONG, MM TOO LONG, DATA ERROR. Ne modifie pas X,Y,Z,T,L. On place en alpha: "NOM1,NOM2" où nom1 et nom2 sont les noms des programmes à échanger. Ce programme pourrait peut-être s'appliquer également à des données.

DANIEL MEYER T59.

Georges-Noel NICOLAS
KERRET
29160 CROZON
(T 438)

HP-IL

Cher amis

Suite à l'appel du dernier "MICRO REVUE" , je prends ma plume avec un tout petit programme. La boucle HP-IL vide et le module HP-IL DEVELOPMENT je pense avoir trouvé tous les codes de la boucle. j'en joins la liste. Le premier tableau correspond aux ordres que l'on envoie avec la fonction SEND du module I/O. Les ordres qui sont en minuscule semblent etre des bouches trous.

Voici le petit programme

```
01 LBL HP-IL     Avant de lancer le PRGM mettre en Y : -1 et en X 0,1,2
02 LBL A         etc... puis se mettre en position MONITOR (HP-IL DEVEL.)
03 X<>Y                Il ne faut plus éteindre la 41.
04  E
05 +                   Le flag 33 est LEVE.
06 VIEW          Donne le N° d'octet qui va etre traité.
07 X<>Y          j'ai essayé un ISG Y mais les chiffres pairs ne donnaient
08 PSE           rien !.
09 WFRM          place X et Y dans les REGS. 1 et 2 du circuit IL.
10 STOP
11 GTO A
```

SEND : SEND COMMAND HP-IL : CMD Envoie de commande N°

AVEC WREG Y=00 à 255 ; X=4

	0	01	02	03	04	05	06	07	08	09
0	NUL	GTL	cmd02	cmd03	SDC	PPD	cmd06	cmd07	GET	cmd09
10	cmd10	cmd11	cmd12	cmd13	cmd14	ELN	NOP	LLO	cmd18	cmd19
20	DCL	PPU	cmd22	cmd23	EAR	cmd25	cmd26	cmd27	cmd28	cmd29
30	cmd30	cmd31	LAD00	LAD01	LAD02	LAD03	LAD04	LAD05	LAD06	LAD07
40	LAD08	LAD09	LAD10	LAD11	LAD12	LAD13	LAD14	LAD15	LAD16	LAD17
50	LAD18	LAD19	LAD20	LAD21	LAD22	LAD23	LAD24	LAD25	LAD26	LAD27
60	LAD28	LAD29	LAD30	UNL	TAD00	TAD01	TAD02	TAD03	TAD04	TAD05
70	TAD06	TAD07	TAD08	TAD09	TAD10	TAD11	TAD12	TAD13	TAD14	TAD15
80	TAD16	TAD17	TAD18	TAD19	TAD20	TAD21	TAD22	TAD23	TAD24	TAD25
90	TAD26	TAD27	TAD28	TAD29	TAD30	UNT	SAD00	SAD01	SAD02	SAD03
100	SAD04	SAD05	SAD06	SAD07	SAD08	SAD09	SAD10	SAD11	SAD12	SAD13
110	SAD14	SAD15	SAD16	SAD17	SAD18	SAD19	SAD20	SAD21	SAD22	SAD23
120	SAD24	SAD25	SAD26	SAD27	SAD28	SAD29	SAD30	sad31	PPE00	PPE01
130	PPE02	PPE03	PPE04	PPE05	PPE06	PPE07	PPE08	PPE09	PPE10	PPE21
140	PPE12	PPE13	PPE14	PPE15	IFC	ppe17	REN	NRE	ppe20	ppe21
150	ppe22	ppe23	ppe24	ppe25	AAU	LPD	ppe28	ppe29	ppe30	ppe31
160	DDL00	DDL01	DDL02	DDL03	DDL04	DDL05	DDL06	DDL07	DDL08	DDL09
170	DDL10	DDL11	DDL12	DDL13	DDL14	DDL15	DDL16	DDL17	DDL18	DDL19
180	DDL20	DDL21	DDL22	DDL23	DDL24	DDL25	DDL26	DDL27	DDL28	DDL29
190	DDL30	DDL31	DDT00	DDT01	DDT02	DDT03	DDT04	DDT05	DDT06	DDT07
200	DDT08	DDT09	DDT10	DDT11	DDT12	DDT13	DDT14	DDT15	DDT16	DDT17
210	DDT18	DDT19	DDT20	DDT21	DDT22	DDT23	DDT24	DDT25	DDT26	DDT27
220	DDT28	DDT29	DDT30	DDT31	cmd224	cmd225	cmd226	cmd227	cmd228	cmd229
230	cmd230	cmd231	cmd232	cmd233	cmd234	cmd235	cmd236	cmd237	cmd238	cmd239
240	cmd240	cmd241	cmd242	cmd243	cmd244	cmd245	cmd246	cmd247	cmd248	cmd249
255	cmd250	cmd251	cmd252	cmd253	cmd254	cmd255				

AVEC WREG Y=00 à 255 ; X=5

	0	01	02	03	04	05	06	07	08	09
00	RFC	rdy01	rdy02	rdy03	rdy04	rdy05	rdy06	rdy07	rdy08	rdy09
01	rdy10	rdy11	rdy12	rdy13	rdy14	rdy15	rdy16	rdy17	rdy18	rdy19
20	rdy20	rdy21	rdy22	rdy23	rdy24	rdy25	rdy26	rdy27	rdy28	rdy29
30	rdy30	rdy31	rdy32	rdy33	rdy34	rdy35	rdy36	rdy37	rdy38	rdy39

```
 40  rdy40   rdy41   rdy42   rdy43   rdy44   rdy45   rdy46   rdy47   rdy48   rdy49
 50  rdy50   rdy51   rdy52   rdy53   rdy54   rdy55   rdy56   rdy57   rdy58   rdy59
 60  rdy60   rdy61   rdy62   rdy63   ETO     ETE     NRD     rdy67   rdy68   rdy69
 70  rdy70   rdy61   rdy72   rdy73   rdy74   rdy75   rdy76   rdy77   rdy78   rdy79
 80  rdy80   rdy81   rdy82   rdy83   rdy84   rdy85   rdy86   rdy87   rdy88   rdy89
 90  rdy90   rdy91   rdy92   rdy93   rdy94   rdy95   SDA     SST     SDI     SAI
100  TCT     rdy101  rdy102  rdy103  rdy104  rdy105  rdy106  rdy107  rdy108  sdy109
110  rdy110  rdy111  rdy112  rdy113  rdy114  rdy115  rdy116  rdy117  rdy118  sdy119
120  rdy120  rdy121  rdy122  rdy123  rdy124  rdy125  rdy126  rdy127  AAD00   AAD01
130  AAD02   AAD03   AAD04   AAD05   AAD06   AAD07   AAD08   AAD09   AAD10   AAD11
140  AAD12   AAD13   AAD14   AAD15   AAD16   AAD17   AAD18   AAD19   AAD20   AAD21
150  AAD22   AAD23   AAD24   AAD25   AAD26   AAD27   AAD28   AAD29   AAD30   AAD31
160  AEP00   AEP01   AEP02   AEP03   AEP04   AEP05   AEP06   AEP07   AEP08   AEP09
170  AEP10   AEP11   AEP12   AEP13   AEP14   AEP15   AEP16   AEP17   AEP18   AEP19
180  AEP20   AEP21   AEP22   AEP23   AEP24   AEP25   AEP26   AEP27   AEP28   AEP29
190  AEP30   AEP31   AES00   AES01   AES02   AES03   AES04   AES05   AES06   AES07
200  AES08   AES09   AES10   AES11   AES12   AES13   AES14   AES15   AES16   AES17
210  AES18   AES19   AES20   AES21   AES22   AES23   AES24   AES25   AES26   AES27
220  AES28   AES29   AES30   AES31   AMP00   AMP01   AMP02   AMP03   AMP04   AMP05
230  AMP06   AMP07   AMP08   AMP09   AMP10   AMP11   AMP12   AMP13   AMP14   AMP15
240  AMP16   AMP17   AMP18   AMP19   AMP20   AMP21   AMP22   AMP23   AMP24   AMP25
250  AMP26   AMP27   AMP28   AMP29   AMP30   AMP31
```

```
AVEC WREG Y=00 à 255 ; X=0   DONNE DAB00 à DAB255
AVEC WREG Y=00 à 255 ; X=1   DONNE DSR00 à DSR255
AVEC WREG Y=00 à 255 ; X=2   DONNE END00 à END255
AVEC WREG Y=00 à 255 ; X=3   DONNE ESR00 à ESR255
AVEC WREG Y=00 à 255 ; X=7   DONNE ISR00 à ISR255
Tout chiffre >7 pour X est identique à x = 1
```

Maintenant je vais vous parler de 2 choses. Au lieu d'avoir acheté une video et un lecteur de disquette HP-IL j'ai acheté à Jacques Vaucelle un SANYO 555/2 256Ko de mémoire vive et 2 lecteurs de disquette de 360 ko. Grace à une routine programmée par Jacques Vaucelle et d'autres par moi, j'ai asservi par l'intermédiaire de l'interface HP-IL RS232C le SANYO. Il est totalement controlé par la 41. Je fais de la VIDEO COU-. LEUR sur 80 collones!!!...Et ceci est d'autant plus facile pour moi depuis que j'ai le module ZENROM. Tous les codes du Sanyo directement sur le clavier de la 41 (de 0 à 255)!!. Meme une touche ESC !..Pour mon imprimante je la branche sur la prise centronix du Sanyo ce qui simplifie ma boucle.

Beaucoup de mes programmes 41 sont enregistrés sous 2 formes: ASCII et sous forme d'une suite de codes hexa. cette dernière étant la forme sous laquelle on réinjecte le PRGM dans la 41. Olivier Pougeon m'a fait un programme qui sous le Basic du sanyo crée un éditeur pleine page.Ce qui permet de modifier nimporte quel octet du PRGM , la traduction sous forme ASCII se fesant sur la droite de l'écran.

Le système d'exploitation du Sanyo est le MS-DOS 2.11 le meme que l'HP 150 et l'HP 110...On peut adresser jusqu'à 4 disquettes de 720Ko. et meme un disque dur de 10Mo.

La vitesse de transfert entre la 41 et le Sanyo est d'environ 190 OCTETS par seconde !!!..J'ai un programme qui fait la fonction FILES du Basic. sur mon microK7 j'envoie 80 caractères d'un coup depuis le buffeur (HP-IL DEVEL).Et bien le voyant BUSI du K7 clignotte le temps d'envoyer la ligne (10 noms de fichiers par ligne).

```
 type DIRW.41D
11:12   15.09
01>LBL "DIRW"
02 CLX
03 X<>F
04 CF 16
05 CF 15
06 CF33
07 66
08 FINDAID
09 SELECT
10 STO 00
```

A partir de cet endroit, je travaille sur le SANYO sous l'éditeur de texte du MS-DOS 2.11 DIRW est le programme qui me permet de faire la fonction "FILES" du k7 sur le sanyo. Je vais vous décrire ce programme car, c'est un bon exemple d'utilisation de la boucle.

Jusqu'à cette ligne initialisation du système
N° de code de l'interface RS232C
Donne la position de l'interface
La sélectionne
Met son adresse dans le reg 00

```
11 ADROFF          Arme le flag 34(mode émetteur/récepteur non activé
12 LAD             RS232C est réceptrice
13 CLX             0 DDL Permet dans les reg. de l'interface d'en-
14 DDL             registrer les 14 data suivantes (reg 00 à 13
15 X<> M
16 "R0003300L0200"
17 OUTAE           Envoie alpha+le caract. codé en X + END
18 ADRON           efface le flag 34
19 UNL             RS232 plus réceptrice
20 CLA
21 X<> M
22 "2J"           pour le sanyo efface l'écran 244,27,91,50,74
23 OUTA            envoie le message précédent
24 16              N° d'ordre du cassette
25 FINDAID         le sélectionne
26 STO 01
27 SF33            Arme le flag 33 Le module HP-IL ne controle plus
28 LAD             la boucle. Le cassette devient récepteur.
29 SDC             Met les périphérique à leur état initial.
30 UNL             les périf. prennent leur état naturel
31 TAD             le cassette devient émetteur
32 2               positionne la bande sur le premier enregistre-
33 DDT             ment et le copie dans le registre du cass
34 113
35 BSIZEX          crée dans la 41 un buffeur de 113 octets
36 AIPT            incrément automatique du pointeur du buffeur.
37 20              recupère les 20 premiers octets dans le buffeur
38 INBUFX          cass. et les copie dans le buf. 41
39 19              place le pointeur buf sur l'octet 20 car le
40 XEQ 02          buf. hp41 commence par l'adresse 0
41 ,8
42 *               de 42 à 46 calcule le nombre de fichiers qu'il
43 INT             y a dans le directory cassette
44  E1
45 *
46 STO 03
47 CLX
48 PT=             Met le pointeur buffeur à 0
49 "ETAT D'ECRITURE"
50 "▶  **41** "
51 A-BUF           Met alpha dans le buffeur
52 "DU "
53 DATE
54 FIX 6
55 ADATE
56 "▶ a "
57 TIME
58 FIX 4
59 ATIME24
60 A-BUF           Met alpha dans le buf. à la suite
61 FIX 0
62 " INITIALISEE a "
63 ARCL 03
64 "▶ FIC"
65 A-BUF
66 "HIER: "     248,72,73,69,82,83,13,10,10
67 A-BUF
68 CLX             place le pointeur du buf à 0. Le texte est
69 PT=             maintenant formatté en 80 collones dans le
70 UNT             buffeur. On va l'envoyer sur le SANYO
71 RCL 00
72 LAD             RS232C  est récepteur
73 80
74 STO 05
75 OUTBUFX         80 caractères envoyé du buf vers l'interface
76 UNL
77 RCL 01
78 TAD             Cassette sélecté
79 2
80 DDT
81 DDT             Le début du directory est maintenant dans le buf.
82 32              du cassette.
83 STO 06
```

```
 84>LBL 03
 85 RCL 06
 86 STO 02
 87 ,009
 88 STO 04
 89>LBL 04
 90 CLX
 91 PT=
 92 UNL
 93 RCL 01              Récupère 32 octets (un enregistrement complet)
 94 TAD                 l'incrémentation dans le buffeur cassette se
 95 RCL 06              fait automatiquement.
 96 INBUFX              Les mets dans le buf. 41
 97 11
 98 XEQ 02
 99 X=0?
100 GTO 04
101 255                 Si l'on obtient un octet de valeur 255 c'est que
102 X=Y?                l'on est en fin de fichier.
103 GTO 01
104 CLX
105 PT=
106 8
107 CLA                 Prend  les 8 premiers octets de l'enregistrement
108 BUF-AX              et les mets dans alpha.
109 RCL 02
110 PT=                 place le pointeur buf. pour mettre 10 noms de
111 A-BUF               fichiers par lignes (8*10=80) alpha dans buf.
112 PT?                 position finale du pointeur
113 STO 02              dans reg 02
114 ISG 04
115 GTO 04
116 XEQ 05
117 GTO 03
119>LBL 01
120 RCL 04
121 INT
122 8
123 *
124 STO 05
125 XEQ 05
126 RCL 01
127 UNT
128 LAD                 sélectionne la cassette
129 7
130 DDL                 Rembobine la cassette
131 RCL 02
132 ST- 03              calcule le nombre de fichiers disponible
133  E
134 ST+ 03
135 CF33                Efface le flag 33
136 ":  " code 243,13,10,10 à la ligne et 2 sauts.
137 OUTA
138 "RESTE: "
139 ARCL 03
140 ACA
141 " DISPONIBLES"
142 OUTA
143 "FINI"
144 BEEP
145 PROMPT
146>LBL 02              Dans le buffeur les nombres sont codés sous
147 PT=                 base 256. Ce sous-prgm décode la valeur de
148 CLX                 l'octet.
149 BUF-XB
150 RTN
151>LBL 05              Ce sous-prgm sélectionne comme récepteur l'in-
152 RCL 06              terface RS232C et envoie 80 caractères.C'est pour
153 PT=                 aller plus vite que 80 est mis dans un registre.
154 UNT
155 RCL 00
156 LAD        158 OUTBUFX
157 RCL 05     159 END
```

Voilà c'est tout pour aujourd'huis.Amicalement.

T438

SOTIROPOULOS C. T-434

La fonction #ae2

1)COMMENT L'OBTENIR
Pour pouvoir utiliser cette fonction il faut que le module TIME soit branche sur la 41(parfois le XF aussi).Sans le TIME la fonction est inoperante.
Les codes de cette fonction sont:0,13 et elle affiche 2 - -

2)LES ARGUMENTS DIRECTS
Pour 00 et un SIZE minimum de 17,2 - - se comporte comme le CRIC,mais en mode PRGM.
Les arguments entre 01 et 15,a l' exception d'un deplacement de l' affichage assez curieux,semblent inoperants.
Les arguments 16 a 27 fournissent,en mode PRGM ,les instructions qui correspondent aux codes decimaux(voir exemple plus loin).
L'argument 28 agit comme SST.
Les arguments 29 a 49 parfois sont inoperants,d'autres fois affichent NONEXISTENT.
L'argument 50 envoie le pointeur PRGM dans la table d'assignations.
Les arguments 51 a 69 semblent inoperants.
Par contre l'argument 70 est assez extraordinaire.Plusieurs cas sont possibles:
1)Vous faites 2 70 et vous attendez. Un CAT ultra rapide se produit et la 41 s'eteint et malgre vos tentatives desesperees elle ne se rallume plus!Pas d' affolement.Appuyez sur CLX et ON et tout va rentrer dans l'ordre.
2)Vous faites 2 70 et tout de suite apres vous appuyez sur la touche ALPHA , c'est USER qui s'eteint,vous appuyez sur USER,vous passez en mode PRGM!
Vous appuyez sur XEQ, XEQ - (et non pas XEQ - -) s'affiche et la 41 s'eteint.
3)Vous appuyez sur [,] 99999999 99 est affiche et vous etes en sequence partielle.
4)Si vous appuyez sur les touches numeriques , 99999999 X9 s'affiche ,X correspondant a la touche enfoncee.

```
                    #ae2 16
                    #ae2 17
                    #ae2 18
                    #ae2 19
                    #ae2 20
                    #ae2 21
                    #ae2 26
                    #ae2 27
                    DEL 999
01♦LBL "TEST"
02 0
03 1
04 2
05 3
06 4
07 5
08 6
                        1 E
                    #ae2 80
                    #ae2 80
                     B1⊢50
Γ                       ΓN
pX
NONEXISTENT
                    #ae2 81
YES
        #ae2 82
            CLX
 0.000000000
            ***
        #ae2 72
 0.000000000
  +20       ***
 #ae2 IND 01
            CLX
 0.000000000
            ***
 #ae2 IND 20
OUT OF RANGE
 #ae2 IND 42
YES
 #ae2 IND 43
YES
 #ae2 IND 44
NO
 #ae2 IND 59
 #ae2 IND 60
        CF 12
```

L'argument 71 provoque parfois un MEMORY LOST, tandis que 72 semble agir comme RCL b.

3)LES ARGUMENTS INDIRECTS

IND 00 parfois est inoperant, parfois il affiche NONEXISTENT.
IND 01 fait passer la HP en mode RAD.
Les arguments IND 40 a IND 45 agissent bien logiquement (codes 168 a 173) comme SF , CF , FS?C etc.

IND 16 donne un CAT "synthetique"(voir plus loin).
IND 19 envoie le pointeur PRGM dans la MEM.
IND 20 donne un nombre pathologique a l'affichage, a condition que le SIZE soit 255 ou superieufr a 257, sinon affichage OUT OF RANGE.

Bonne recherche

```
+
<=0?
@>
0?
<Y?
cD
0
00+a&GB1-W>
LN
W
10        -
S
0?
=0?
&
HS
c
H
S
;
05CF
dSΓ_vΓ
α-10>
50
STX
0?
S-
>
μ
FACT
```

```
          #ae2 IND 19
     0.000000000    ***
ROM
          #ae2 IND 20
NO
            SIZE 255
          #ae2 IND 20
YES
          #ae2 IND 20
YES
              CF 01
          #ae2 IND 20
OUT OF RANGE
           #ae2 IND X
     0.000000000    ***
          #ae2 IND 99
NO SUCH KEY
KEYCODE?
              GTO ..
PACKING
          #ae2 IND 99
NONEXISTENT
          #ae2 IND 98
NONEXISTENT
          -ae2 IND 97
NONEXISTENT
          #ae2 IND 12
                 CLX
     0.000000000    ***
```

IND 12 crée un ALARME PAST-DUE

```
                 ALMCAT
          #ae2 IND 12
                 ALMCAT
```

```
00:09   06/22
x◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
```

BELAIRE PATRICK

65,Rue Paul ELUARD,

622IO AVION

T567

Le I9/II/I984

Cher ami,

Ca y est . Maintenant je suis en possession de mon numero PPC. Je vais pouvoir t'envoyer des programmes, jusqu'en fevrier 85(date à laquelle je pars faire mon Service Nationnal,mais continues à m'envoyer Micro-Revue à la même adresse et lorsqu'il le faudra je renouvellerai mon abonnement).

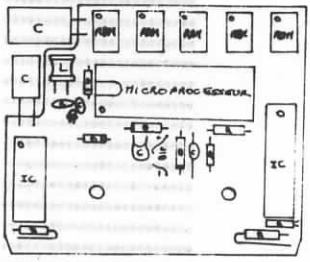
Ci-joint un programme-jeu : BASE.

Au fait, felicitations pour ton livre AU FOND DE LA H-P 4I(je l'ai racheté à Didier JEHL).Moi qui suis debutant en H-P. je l'ai trouvé extra, seulement comme j'ai une CX. il y a quelques petites differences surtout au niveau du circuit logique. Ci-joint un schéma.Mais aussi au niveau carte de la memoire vive.Aussi, s'il y a eu des modifications et des informations supplementaires pourrais-tu me les faire parvenir,c'est la 2ème édition (janvier 83).Merci.

UN GRAND BRAVO POUR LA PRESENTATION DE MICRO-REVUE;*

Amicalement,

Belaire



CIRCUIT LOGIQUE MP 44CX 1984
T567

* Dont je te commande le n° 1, je te joint donc un chèque de 40F.

W

Bernard JOUSSE (T 502)
Barzun
64530 PONTACQ

Le 02-01-85

La fonction W

Dans Micro Revue n° 2 et 3, deux articles sur la fonction W ont été publiés. A priori, ils arrivent a des conclusions différentes. Toutefois, il apparaît que ces résultats sont valables si l'on prend en compte la configuration matériel utilisée. Il est fort probable que Cyril Reigner travaille avec une HP-41 sans autres accessoires que le XF et le PPC et que Franck Lebastard dispose soit d'une CX soit d'un module TIME.

Dans mon cas j'ai CV+TIME+(XF+ZENROM) en port 1+XIO en port 3 et un Card Reader. J'ai toutefois enlevé le TIME pour arriver aux mêmes résultats que Cyril. Il semblerait donc que ce qui rentre en ligne de compte est l'accessoire de n° ROM le plus faible. Pour en être sûr, il me faudrait la réponse a deux questions: 1) Quel est l'effet de la fonction 2B0 (Clear Data Registers) en particulier sur le Carry et 2) Que se passe-t-il lorsque l'on envoie la HP a une adresse ROM vide. Je pense qu'elle exécute des NOP jusqu'a trouver une adresse non vide (boucle-t-elle a 0000 après FFFF ?).....ou effectue-t-elle un RTN ?

Dans le cas où vous n'avez pas de ROM a n° inférieur a 8, vous obtiendrez les effets décrits par Cyril. Il n'y a pas que la touche (11) qui puisse être activée: la séquence:

```
LBL TT
W @      (1F X1 11) X quelconque donne la touche 11

Si on a    (1F X1 87) on obtient la touche (-11)
```

Il y a sans doute d'autres opportunités car il y a 2 ans j'avais réussi a faire exécuter la touche (-84)

Pour essayer de comprendre W, il faut attaquer le microcode. En l'absence de réponse a mes deux questions, je me suis amusé a voir pourquoi cette fonction s'affichait W avec ses particularités.

Considérons que nous sommes en mode PRGM at que nous faisons des SST sur le programme. Pour l'affichage de la fonction, tout se passe en 0563 (DFILLF dans le VASM). La machine a déja mis le numéro de ligne daqns l'affichage, le code de la fonction a afficher (ici 1F) est en S&X de C et c'est parti.

Tout d'abord, on effectue un branchement en 040X où X est le premier digit de la fonction donc on va en 0401 qui donne un JNC *+14. Le saut nous amène en 0415 qui est un ?NC GO -> 04AD. En 04AD, on trouve la routine DEROW quit traite de l'affichage des fonctions 1X. Ces fonctions sont: entrée de nombres, GTO et XEQ alpha. Le premier test porte sur la valeur de la fonction: >1C comme c'est le cas, elle sera traitée comme un GTO/XEQ alpha. On saute donc en 04E6 (ROW0110) sans passer par la case départ. Cette routine fait appel a PROMFC (05C7) et NBYTA0 (2D04) avant de tomber dans TXSTR.

PROMFC: Routine qui affiche le nom de la fonction: pour ce faire, elle lit le mot contenu en 14XY (=141F ce qui donne 03E) met son premier digit a 0 et rajoute un 1 devant et le décrémente on obtient donc 10AD. A partir de cette adresse, PROMFC lit des caractères en remontant jusqu'a

trouver un caractère supérieur a 80h.

Or en 10AD on a le W de AVIEW qui, étant le dernier caractère du nom est supérieur a 80h. La fonction s'affichera donc W. PROMFC ajoute un espace derrière ce W et rend la main. Après cela, on charge l'adresse du byte suivant en A avec NBYTA0 et on passe a la suite soit TXSTR.

TXSTR: fonction d'affichage d'une chaîne de caractères (cf GTO/XEQ). On va donc prendre le byte suivant 1F, garder son dernier digit et afficher autant de caractères que la valeur de ce digit. (Ceci après avoir remplacé l'espace derrière le W par un super tee ⊤).

En conclusion, a l'affichage, le byte suivant 1F est traité comme le byte suivant le 1D ou 1E de GTO et XEQ alpha et on a donc un comportement analogue a celui de GTO/XEQ alpha.

Reste le problème du prochain SST.

Après plusieurs opérations commençant en 129E, on voit que le coeur du problème se situe a partir de 2AF7 (NXLSST). EN fait, la HP-41 se "plante" a peu près de la même manière que précédemment: Elle traite la fonction W comme un GTO/XEQ alpha. Or, du point de vue des longueurs de ligne programme, un GTO/XEQ aqlpha se présente comme un octet suivi d'une ligne de texte (séquence 1D ou 1E, FX.....). La 41 traite donc de la manière suivante: Si le premier digit de la fonction est un 1 et que le deuxième est supérieur a C, alors, on incrémente le pointeur de programme pour cet octet et on retourne a un décodage de fonction. Ceci étant justifié par l'exixtence de l'octet FX après le 1D/1E. On voit dons que la ligne affichée après le W dépendra de l'octet suivant le W. Si c'est un code de fonction a un octet, il sera ignoré. Si c'est un code de fonction a n octets, la HP ignorera n octets.

On peut visualiser la chose en rentrant une chaîne commencant par 1F et en faisant varier l'octet suivant.

Par exemple la chaîne 1F 21 87 86 7E 89 s'affichera:

```
W ⊤
CLA
BEEP
AVIEW
PSE
```

Si on remplace le 21 par 91, on aura:

```
W ⊤
BEEP
AVIEW
PSE
```

De même un 92 donnera:

```
W ⊤⊤
BEEP
AVIEW
PSE
```

Alors qu'un 22 donne:

```
W ⊤⊤
CLA
BEEP
AVIEW
PSE
```

Il ne reste plus qu'a déterminer l'exécution de W en sachant que le système décrit ci dessus pour le SST sera le même employé pour passer "a la ligne" après l'exécution de W

Heureuse Programmation a tous.

Programme AHYPER	Date de revision 30/04/85	Auteurs Bernard Jousse (T 502)

Description rapide du programme

Cet ensemble de fonctions microcode calcule les fonctions arguments du groupe HYPER.

ACH(x) = LN(x+SQR(x^2+1))
ASH(x) = LN(x+SQR(x^2-1))
ATH(x) = (LN((x+1)/(1-x)))/2
ACOTH(x)= (LN((x+1)/(x-1)))/2
AGD(x) = ATH(SIN(x))
ASEH(x) = ASH(1/x)
ACEH(x) = ACH(1/x)

Microcode	X	Langage RPN	
Taille	0 octets		0 Registres
Size requis	0		

Toutes les remarques faites pour HYPER sont applicables.

Algorithme employe et autres renseignements

Listing Microcode

```
084   D
007   G
001   A
2A0   SETDEC
0F8   C=REG3 (X)
365
050   GOSUB 14D9  (CHK#2)        Test alpha data
070   N=C
221
048   GOSUB 1288  (SIN)
3C4   ST=0
07B   GONC * +0F
088   H
014   T
00F   O
003   C
001   A
388   S0=1
023   GONC * +04
088   H
014   T
001   A
2A0   SETDEC
0F8   C=REGN3 (X)
365
050   GOSUB 14D9     (CHK#S2)    Test alpha data
070   N=C
10E   A=C W
08E   B=A W
009
060   GOSUB 1802   (SUBONE)       x-1
0F0   CN EX
10E   A=C W
08E   B=A W
001
060   GOSUB 1800  (ADDONE)     x+1
38C   ?S0=1
013   GONC * +02
2BE   C=-C-1 S
10E   A=C W
0B0   C=N
261
060   GOSUB 1898   (DIV2-10)
115
06C   GOSUB 1B45   (LN10)
10E   A=C W
04E   C=0 W
```

Programme AHYPER

```
35C   PT=12
090   LC 2
261
060   GOSUB 1898   (DIV2-10)
331
002   GOLONG 00CC   (NFRX)  Finito...
088   H
005   E
013   S
001   A
033   GONC * + 06
088   H
005   E
003   C
001   A
308   S1=1
388   S0=1
04B   GONC * +09
088   H
003   C
001   A
308   S1=1
023   GONC * +04
088   H
013   S
001   A
2A0   SETDEC
0F8   C=REG3 (X)
365
050   GOSUB 14D9  (CHK#S2)    Test alpha data
070   N=C
38C   ?S0=1
22D
061   GOSUBC 188B   (1/X10)
10E   A=C W
135
060   GOSUB 184D   (MP2-10)
30C   ?S1=1
009
061   GOSUBC 1802   (SUBONE)
30C   ?S1=1
001
060   GOSUBNC 1800  (ADDONE)
3C4   ST=0
2F9
060   GOSUB 18BE    (SQR10)
10E   A=C W
0B0   C=N
01D
060   GOSUB 1807    (AD2-10)
115
06C   GOSUB 1B45    (LN10)
283   GONC * -30    Vers terminaison
```

Programme HYPER	Date de revision 29/04/85	Auteurs Alain Goubault (T)

Description rapide du programme

Cet ensemble de fonctions microcde effectue les calculs des fonctions hyperboliques:

```
CH(x)    = (EXP(x)+EXP(-x))/2
SH(x)    = (EXP(x)-EXP(-x))/2
TH(x)    = (EXP(2x)-1)/(EXP(2x)+1)
GD(x)    = ASIN(TH(x))
SEH(x)   = 1/SH(x)
CEH(x)   = 1/CH(x)
COTH(x)  = 1/TH(x)
```

Microcode X	Langage RPN	
Taille	0 octets	0 Registres
Size requis	0	

Du point de vue des mouvements de pile, du registre Last x et des messages d'erreur, ces fonctions se comportent exactement comme EXP(x) ou SIN(x)

Par exemple:

```
 ASTO x  CH    donnera ALPHA DATA
5000 CH donnera OUT OF RANGE
Clx COTH  donnera DATA ERROR....
```

X se retrouve en Lastx, la pile monte si elle a le droit, la fonction s'ecrit en X et la montee de pile est autorisee. Le flag 25 a toujours les memes possibilites.

Algorithme employe et autres renseignements

Listing Microcode:

```
088   M
005   E
003   C
208   S2=1
043   GONC * +08
088   H
005   E
013   S
208   S2=1
03B   GONC * +07
088   H
013   S
308   S1=1
01B   GONC * +03
088   H
003   C
2A0   SETDEC
0F8   C=REGN3 (X)
356
050   GOSUB 14D9  (CHK#S2)  Test Alpha Data
029
068   GOSUB 1A0A  (EXP10)   Calcul EXP(x)
070   N=C
22D
060   GOSUB 188B  (1/X10)   Calcul EXP(-x)
30C   ?S1=1
013   CONC * +02
2BE   C=-C-1 S
10E   A=C W
0B0   C=N
01D
060   GOSUB 1807  (AD2-10)   Calcul EXP(x)± EXP(-x)
10E   A=C W
04E   C=0 W
35C   PT=2
090   LC 2
20C   ?S2=1
013   GONC * +02
0AE   AC EX W
261
060   GOSUB 1898  (DIV2-10)   Division finale
153   GONC * +2A              Terminaison
084   D
007   G
308   S1=1
04B   GONC * +09
088   H
```

Programme HYPER

```
014   T
00F   O
003   C
208   S2=1
01B   GONC * +03
088   H
014   T
2A0   SETDEC
0F8   C=REGN3 (X)
365
050   GOSUB 14D9    (CHK#S2)    Test Alpha data
10E   A=C W
01D
060   GOSUB 1807    (AD2-10)    Calcul de x+x
029
068   GOSUB 1A0A    (EXP10)     Calcul de EXP(2x)
070   N=C
009
060   GOSUB 1802    (SUBONE)    EXP(2x)-1
0F0   CN EX
10E   A=C W
08E   B=A W
001
060   GOSUB 1700    (ADDONE)    EXP(2x)+1
10E   A=C W
0B0   C=N
20C   ?S2=1
017   GOC * +02
0AE   AC EX W
261
060   GOSUB 1898    (DIV2-10)
30C   ?S1=1
023   GONC * +04
070   N=C
261
040   GOSUB 1098    (ASIN)
331
002   GOLONG 00CC   (NFRX)      That's all....
```

Nota: Du fait de l'absence d'arrondi dans les calculs intermediaires, il est possible que les resultats differenr legerement de ceux obtenus en calculant au clavier. Ceci est vrai pour toutes les fonctions MC de TOULMATH

TOULMATH		PPC-T
Programme GAUSS	Date de revision 26/04/85	Auteurs Bernard Jousse (T502)

Description rapide du programme

Ce programme permet d'integrer une fonction par la methode dite de GAUSS. Une des bornes de l'intervalle peut etre infinie.

La fonction doit etre definie et continue sur l'intervalle pour que le resultat ait un sens.

Dans le cas d'un intervalle fini, il est possible d'augmenter la precision par un decoupage en sous intervalles.

Microcode	Langage RPN X	
Taille	450 octets	65 Registres
Size requis	9	

Le programme peut etre appele en execution 'manuelle' par XEQ ^GAUSSM ou pour une execution en sous programme par XEQ ^GAUSS

Algorithme employe et autres renseignements

La methode employee est une methode de GAUSS en 12 points suivant:

$$\int_a^b f(x)\,dx = \frac{b-a}{2} \sum_{i=0}^{11} \left[w_i\, f\left(\frac{b-a}{2} z_i + \frac{b+a}{2} \right) \right]$$

$$\int_a^{+\infty} f(x)\,dx = 2 \sum_{i=0}^{11} \left[\frac{w_i}{(1+z_i)^2}\, f\left(\frac{2}{1+z_i} + a - 1 \right) \right] \qquad \int_{-\infty}^{a} f(x)\,dx = \int_{-a}^{+\infty} f(-x)\,dx$$

Les differentes constantes sont donnees par:

$W_{z_{i+1}} = W_{z_i}$ $\qquad$ $Z_{z_{i+1}} = -Z_{z_i}$

avec

Z0 = .981560342	W0 = .04717533639
Z2 = .9041172564	W2 = .106939326
Z4 = .7699026742	W4 = .1600783285
Z6 = .5873179543	W6 = .2031674267
Z8 = .367831499	W8 = .2334925365
Z10= .1252334085	W10= .2491470458

Programme GAUSS

Utilisation des registres

R00 a R08 inclus sont utilises. A la fin de l'execution, seuls R00 (resultat), R03 (nom de fonction) et R04 (code pour REGSWAP) contiennent des donnees utilisables.

Pnedant l'execution de la fonction a integrer, le rgistre ALPHA contient des resultats intermediaires. La fonction a integrer ne doit donc pas utiliser les registres M, N et O. Par contre, il est prevu un code pour l'execution d'un REGSWAP avant chaque appel de la fonction, et apres chaque retour.

Utilisation des drapeaux

Si F00 est leve a l'appel, on travaille avec une borne infinie. Ce sera la borne superieure si F01 est baisse, la borne inferieure si F01 est leve.
Si on leve F01, il n'est pas necessaire de lever F00.

F02 est utilise par le programme pour son usage personnel....

Utilisation en manuel

Donner les data comme demande. Pour a et b, faire R/S si borne infinie. Taper 0 R/S si valeur nulle. Il y a un test sur F22. Si pas de REGSWAP: R/S. Si une seule iteration, R/S

2 niveaux de sous programme...

Exemples de precision

Soit a integrer $-1/x^2$

Calcul de 1 a 500 en 1 passe: -.440
en 4 passes: -.861
en 10 passes:-.9806

Calcul de 100a + ∞ : -.0090826
Calcul de 100 a 2000 en 4 passes puis de 2000 a $+\infty$: -.0095710

APPEL EN SOUS-PROGRAMME

Registre Alpha : nom de la fonction

X : borne inferieure

Y : borne supérieure

Z : nombre de passes

T : code de REGSWAP

TOULMATH		PPC-T
Programme DER	Date de revision 23/02/85	Auteurs Eric Aubourg (T 366)

Description rapide du programme

Ce programme effectue les calculs de derivation locale d'une fonction.

On peut calculer les derivees a droite, a gauche ou normale.

Microcode	Langage RPN X	
Taille	136 octets	20 Registres
Size requis	6	

Algorithme employe et autres renseignements

$$f'_d(x) = \lim_{h \to 0^+} \frac{f(x+h) - f(x)}{h}$$

$$f'_g(x) = \lim_{h \to 0^+} \frac{f(x) - f(x-h)}{h}$$

$$f'_s(x) = \lim_{h \to 0^+} \frac{f(x+h) - f(x-h)}{2h}$$

Dans la pratique, le programme calcule une suite Y'n qui converge vers f'(x)
A chaque iteration, on calcule $\Delta_n = Yn - Yn-1$
Si $|\Delta n| > |\Delta n-1|$, on n'a pas de convergence.
Si apres un RND, on a $\Delta n = \Delta n-1$, la precision est atteinte

Il est a noter que si f est derivable en x, on a $f'_g(x)=f'_d(x)=f'_s(x)$ toutefois, le calcul de $f'_s(x)$ converge plus vite et est plus precis.
Si $f'_g(x) \neq f'_d(x)$, $f'_s(x)$ n'a en general aucun sens.....

Programme DER

Utilisation des registres

R00 contient h
R01 est la valeur pour laquelle on veut f'(x)
R02 f'(x)
R03 Δ
R04 provisoire
R05 nom de la fonction

Utilisation des drapeaux
F00 usage interne. Si leve au retour: pas de convergence
F01 calcul de fg
F02 clacul de fd

Labels locaux

Les LBL 00, 01, 02, 04 et 10 sont utilises

Appel manuel

Faire un FIX n pour la precision voulue
XEQ ^DERM, rentrer le nom de la fonction (FN par defaut)
Rentrer G, D ou S suivant le type de derivation
Rentrer la valeur de Xo

Le programme affiche les approximations successives et s'arrete soit lorsque la precision demandee est atteinte, soit sur divergence.(F00 leve).
Au cas ou la precision atteinte serait inferieure a celle voulue, F00 est leve

F00 leve correspond a un resultat inutilisable

Appel en sous-programme

Avoir FIX n, le nom de la fonction en R05, Xo en X, F01 et 02 baisses pour f'_s
F01 leve pour f'_g ,F02 leve pour f'_d
Memes remarques que precedemment pour F00

ERRATA

TOULMATH		PPC-T
Programme *******	Date de revision 23/02/85	Auteurs Eric Aubourg (T 366)

Description rapide du programme

Les deux programmes DEC-BSE et BSE-DEC font les changements de base 10 en base quelconque et vice-versa.

Microcode	Langage RPN X	
Taille	104 octets	15 Registres
Size requis	0	

Algorithme employe et autres renseignements

La methode utilisee utilise les fonctions XTOA et ATOX du module XF.

Vu la simplicite du programme, il n'y a pas de deuxieme page

Registres utilises: aucun
Drapeaux: utilisation de F00
Labels locaux: LBL 00 et 01 sont utilises

Appel: b ENTER^ n XEQ ^DEC-BSE au retour le registre alpha contient n en base b
^XXXX b XEQ ^BSE-DEC au retour, X contient l'equivalent decimal du registre alpha en base b

Programme GAMMA	Date de revision 18/02/85	Auteurs Eric Aubourg (T 366)
		Microcode Langage RPN X
		Taille 214 octets 31 Registres
		Size requis 0

Description rapide du programme

Ce programme calcule la fonction $\Gamma(x+1)$ pour des valeurs de 0 a environ 69.9

Pour des raisons de compatibilite avec FACT, on calcule $\Gamma(x+1)$

Algorithme employe et autres renseignements

De facon exacte la fonction GAMMA est donnee par $\Gamma(x) = \int_0^{+\infty} t^{x-1} e^{-t} dt$

Dans ce programme, 3 methodes differentes sont utilisees:

X entier: (x+1)=x!

X non entier et inferieur a 14
on utilise la recurrence (x+1)=x (x) pour se ramener dans [0,1[
puis on fait un developpement limite

X non entier et superieur a 14
application de la formule de Stirling:

$$\Gamma(x+1) = \sqrt{2\pi x}\left(\frac{x}{e}\right)^x \left[\frac{1}{288x^2}\left(1-\frac{139}{180x}-\frac{571}{8640x^2}\right)+\frac{1}{12x}+1\right]$$

NOTA: pile et registre Alpha detruits, resultat en X

Programme GAMMA

Utilisation des registres

Programme en SIZE 000. Seuls le registre Alpha et la pile sont utilises.

Utilisation des drapeaux

Aucun drapeau utilise

Labels locaux

Les LBL 00 a 04 inclus sont utilises

Appel en sous-programme

Direct par XEQ ^GAMMA apres initialisation de X

ERRATA

Programme ASYM	Date de revision 18/02/85	Auteurs Eric Aubourg (T 366)
Microcode	Langage RPN X	
Taille	183 octets	27 Registres
Size requis	6	

Description rapide du programme

Ce programme recherche, si elles existent les asymptotes a une fonction.

Algorithme employe et autres renseignements

Le programme recherche la limite (si elle existe) de la pente de la courbe representant la fonction. L'ordonnee a l'origine est alors determinee par difference.

Programme ASYM

Utilisation des registres

R00 Derniere valeur de X pour le calcul de la fonction
R01 Prochaine valeur de X a utiliser
R02 Derniere pente calculee
R03 Dernier (Test de convergence, voir DER)
R04 Provisoire
R05 Nom de la fonction

Utilisation des drapeaux

F00, 01 et 02 sont utilises par ASYM

Labels locaux

Les LBL 00 a 03 inclus sont utilises

Appel du programme

L'appel se fait en mode direct par XEQ ^ASYM
Au prompt NOM FONCTION, taper le nom (FN par defaut)
A la demande taper + ou - suivant l'asymptote cherchee (+ par defaut)

Le programme affiche les pentes successives calculees puis soit
Pas d'asymptotes
soit l'equation de l'asymptote

NOTA: ATTENTION AUX BRANCHES PARABOLIQUES HORIZONTALES
(LOG a pour asymptote Y=9.4170 !!!!!)

ERRATA

Programme CONIQUE	Date de revision 23/02/85	Auteurs
		Eric Aubourg (T 366) J-Marc Vigilant (T 448)
		Microcode Langage RPN X
		Taille 1110 octets 159 Registres
		Size requis 7

Description rapide du programme

Ce programme permet a partir d'une equation du type

$$ax^2+by^2+cxy+dx+ey+f=0$$

de determiner la nature et les parametres de la conique associee a l'equation.

Le programme est a acces manuel uniquement.....

Algorithme employe et autres renseignements

1) Calcule de $\Delta=-c^2+4ab$ Si nul, parabole. Sinon, conique a centre.

2) Cas des paraboles
Equation reduite de la forme: $y^2-2px=0$ ($\gamma=2p$)
(a+b) donne la constante multiplicative ($a=k\sin^2\theta$ et $b=k\cos^2\theta$)
$c=-\sin 2\theta$ donc $tg=-sgn(c)\sqrt{a/(b+\varepsilon)}$ ($\varepsilon\to 0$ pour $\theta=\pi/2$)
$d=-\gamma\cos\theta+2\beta'\sin\theta$
$e=-\gamma\sin\theta-2\beta'\cos\theta$
on a donc $\beta'=1/2\ (d\sin\theta-e\cos\theta)$
$\gamma=-d\cos\theta-e\sin\theta$
$\alpha'=(f-\beta'^2)/\gamma$
$\alpha=\alpha'\cos\theta-\beta'\sin\theta$
$\beta=\alpha'\sin\theta+\beta'\cos\theta$

3) Cas des coniques a centre
Equation reduite de la forme $X^2/A+Y^2/B-1=0$
$\alpha=(ce-2bd)/\Delta$ $\beta=(cd-2ac)/\Delta$
$tg\ 2=c/(a-b+\varepsilon)$ ($\varepsilon\to 0$)
$B=a\cos^2\theta+b\sin^2\theta+(c/2)\sin 2\theta$
$A=a+b-\beta$
$k=-(f+e\beta+d\alpha+a\alpha^2+b\beta^2+\alpha\beta c)/AB$
$A \leftarrow kA$ $B \leftarrow kB$

Si A=B : cercle
Si AB<0 : hyperbole
Sinon: ellipse

4) Pour les details, voir les figures jointes

Programme CONIQUE

Utilisation des registres

R00 a R06 inclus sont utilises.
En fin de programme:
R00 (angle axe focal/axe abcisses)
R01 et R02 demi grand axe
R03 et R04 demi petit axe
R05 et R06 coodonnees du centre

Utilisation des drapeaux

F00 a F07 inclus, F24 et format d'affichage modifies...

Labels locaux

Les LBL 00,01,02,04,06,07,08,16,17,98,99,A,C,D,J sont utilises

Routine d'affichage (LBL A)
Affiche successivement et suivant la conique:
-Centre
-Axe focal
-Sommet(s)
-Foyer(s)
-Parametre
-Points a la verticale du(des) foyer(s)
-Excentricite
-Directrice

Exemple d'utilisation

Soit 9x +16y -24xy+8x_94y+374=0

Apres entree des valeurs, la machine affiche 'PARABOLE', puis
Axe focal oblique =36,87 (degres) Y=0.75 X + 2.00
Sommet A: ⟨4.00:5.00⟩ Foyer F: ⟨4.40:5.30⟩
Parametre P=1.00 P ': ⟨3.80:6.10⟩ P '': ⟨5.00:4.50⟩
Directrice H=0.50 Y=-1.33 X +9.50
A la question POINTS?, un appui sur ENTER fait redemander les coefficients.....

On peut verifier que on a affaire a une parabole, d'axe focal Y=0.75 X +2.00
Le foyer est en X=4.40 Y=5.30 La directrice est Y=-1.33 X +9.50 a une distance de 0.50 du foyer.
Les tangentes horizontales et verticales passent par les points:
X=5.00 Y=4.50 et X=3.80 Y=6.10

CQFD...

ERRATA

Programme GENCONI	Date de revision 23/02/85	Auteurs
		Eric Aubourg (T 366) J-Marc Vigilant (T 448)
		Microcode Langage RPN X
		Taille 247 octets 36 Registres
		Size requis 7

Description rapide du programme

GENCONI effectue les operations inverses de celles faites par CONIQUE:

A partir des parametres d'une conique, il en fournit l'equation.

Se referer a CONIQUE pour la significati on des symboles.

Algorithme employe et autres renseignements

Suivant la nature de la conique (E, H, P), le programme demande les differem parametres necessaires aux calculs.

.....

Le programme stocke les coefficients A a F dans les registres 01 a 06.
R00 est aussi utilise, de mem que F00 et les LBL 00 a 02 inclus.

.....

Programme RLIN	Date de revision 18/02/85	Auteurs Eric Aubourg (T 366)
		Microcode Langage RPN X
		Taille 105 octets 15 Registres
		Size requis 6

Description rapide du programme

Ce programme permet de faire une regression lineaire sur un ensemble de couples (x,y) rentres au prealable par $\Sigma+$.

Il est possible de :
Calculer l'equation de la droite de regression
Calculer le coefficient de correlation
D'interpoler des valeurs de x et y

Algorithme employe et autres renseignements

On utilise les formules classiques:

$$Y = ax + b$$

$$a = \frac{cov(x,y)}{v(x)} \qquad b = \bar{Y} - a\bar{x} \qquad \bar{x} = \frac{\Sigma x}{n} \qquad \bar{y} = \frac{\Sigma Y}{n}$$

$$v(x) = \sigma_x^2 = \frac{\Sigma x^2 - n(\Sigma x)^2}{n}$$

$$cov(x) = \frac{\Sigma xy - (\Sigma x \Sigma y)/n}{n}$$

Exemple: soient les couples de valeurs:
(1981,1.2) (1982,1.5) (1983,1.9) (1984,2.1)

On effectue CLΣ puis 1.2 ↑ 1981 $\Sigma+$ 2.1 ↑ 1982 $\Sigma+$ ..... 1984 $\Sigma+$

Puis 1985 XEQ ^RLY? donne 2.45 RLCOR donne 0.993

Les resultats sont:
RLIN en X : pente de la droite de regression, en Y ordonnee a l'origine
RLCOR en X : coefficient de correlation
RLX? et RLY? en X : l'autre valeur

A la fin du programme, les registres statistiques sont en R00 a R05.
Pas de drapeaux utilises
Les LBL 00 et 01 sont utilises

Programme INTLAG	Date de revision 18/02/85	Auteurs Eric Aubourg (T 366)
Microcode	Langage RPN X	
Taille	201 octets	29 Registres
Size requis	Variable	

Description rapide du programme

Ce polynome effectue une regression polynomiale sur n couples de points (X,Y).

On obtient donc un polynome de degre n-1 'passant' par ces n points.

Algorithme employe et autres renseignements

On utilise la formule de LAGRANGE: $L_k(x) = \prod_{j \neq k} (x - x_j)$

$$P(x) = \sum_{i=1}^{n} y_j \frac{L_j(x)}{L_j(x_j)}$$

Pour calculer x connaissant y, on inverse les couples (X,Y) par REGSWAP.

Utilisation manuelle
XEQ ^INTLAG et rentrer les couples
Pour calculer y=f(x) x XEQ ^LAGY?
x=f^{-1}(y) y XEQ ^LAGX?

Utilisation en sous programme
n en R00 , les Xi en Ri (i de 1 a n) , les Yi en Ri+n
Size minimum: (2n+1)
Faire CF01 XEQ ^LAGY? ou XEQ ^LAGX?

Drapeaux:
F00 utilise par programme, F01 distingue entre y=f(x) (baisse) et x=f^{-1}(y) (leve)

Labels locaux
Les LBL 00 a 05 inclus sont utilises

Exemple:
on connait les couples (1,1.2) (2,1.5) (3,1.9) (4,2.1)
En manuel XEQ ^INTLAG
Nbre de points 4 R/S
puis 1 R/s 1.2 R/S 2 R/S ..... 2.1 R/S
2,5 XEQ ^LAGY? 1.705
En sous programme
4 STO 00 1 STO 01 1,2 STO 05 .....4 STO 04 2.1 STO 08
CF 01 2.5 XEQ ^LAGY?

TOULMATH		PPC-T
Programme *******	Date de revision 21/04/85	Auteurs Eric Aubourg (T 366)

Description rapide du programme

Ces programmes calculent la valeur en un point donne des polynomes de Tchebyschev,
Legendre,
Hermite,
Laguerre.

Microcode	Langage RPN X	
Taille	216 octets	31 Registres
Size requis	0	

Algorithme employe et autres renseignements

* Polynomes de Tchebyschev:
  Verifient les relations: $T_n(\cos x)=\cos nx$ $\quad n \neq m \rightarrow \int_{-1}^{+1} \frac{T_n T_m(t)}{\sqrt{1-t^2}} dt = 0$

  Avec $T_0=1$ et $T_1=x$, on a $T_n(x)=2xT_{n-1}(x)-T_{n-2}(x)$

* Polynomes de Legendre:
  Verifient: $n \neq m \rightarrow \int_{-1}^{1} R_m R_m(t)\, dt = 0$ $\quad R_n(x) = \frac{1}{2^n n!}\left[(x^2-1)^n\right]^n$

  Avec $R_0=1$ et $R_1=x$, on a $R_n = \frac{x(2n-1)R_{n-1}-(n-1)R_{n-2}}{n}$

  Ces polynomes sont definis a une constante pres. Ceux calcules par le programme verifienr R (1)=1. Si on veur un polynome dont le terme de plus haut soit 1, il faut multiplier le resultat par $\frac{(n!)^2\, 2^n}{(2n)!}$

* Polynomes de Hermite:
  $H_0=1$ $\quad H_1=2x$ $\quad H_n=2xH_{n-1}-2(n-1)H_{n-2}$

* Polynomes de Laguerre:
  $L_0=1$ $\quad L_1=1-x$
  $L_n=\frac{(L_{n-1}-x)L_{n-1}-(n-1)L_{n-2}}{n}$

Aucun registre de donnees. Pile et registre alpha detruit. A chaque boucle X contient P et Y contient P

Utilisation: n | X XEQ ^P.... (TCHX, LEGX, HERX ou LAGX)

Exemple: 3 ENTER| .4 XEQ ^PTCHX D'ou T (0.4)=-0.9440

Le Point sur TOULMATH

Apres un gros travail de Eric, Jean Marc, Alain et quelques autres, TOULMATH prend bonne tournure. Avant de passer aux listings, voici la liste de ce qui existe:

* RPN

CALMAT	: tous (ou presque) les calculs sur matrices réelles
DER	: derivation locale de fonction
RESO	: résolution de f(x)=0
*FN	: utilitaire pour GAUSS, RESO, DER....
CONIQUE	: de l'équation a la géometrie
GENCONI	: l'inverse de CONIQUE
DEC-BSE	: de la base 10 a la base x
BSE-DEC	: de la base x a la base 10
GAMMA	: (son nom parle tout seul..)
ASYM	: calcul des asymptotes de fonction
EQDIF1/2	: résolution d'équations différentielles
RLIN	: régression linéaire
INTLAG	: réfression polynomiale
CALPOL	: calculs sur polynomes réels
INTEG	: intégrales par Gauss en 12 points
PTCH	: calculs des polynomes de Tchebyschev, Hermite etc.. pour utilisation avec CALPOL
PTCHX	: valeur locale des polynomes ci dessus mentionnés
VPRP	: valeurs et vecteurs propres d'une matrice d'ordre n

* NC

Hyperboliques

* Developpements en cours

En RPN tout est bienvenu, en particulier les triangles sphériques, transformées de Fourier, fonctions de Bessel et autres.

EN MC, Eric est en contact avec Steen Petersen (PPC-DK 3, PPC 9583) qui transforme sa HP 41 en HP 15, mais en mieux. Il travaille aussi sur les complexes et les equivalents MC de RESO et INTEG. De toute facon RESO et INTEG seront aussi publiés. Alain travaille sur CALMAT en MC et de mon coté, je m'attaque aux changements de coordonnées dans l'espace et a deux routines d'affichage de message pour permettre de faire des ROM a message en langues différentes. A part ca, on prend toute bonne idée, en particulier *FN en MC, et VA et AVN de "Tips & Routines" en MC.

HP-71

Francois LE GRAND
T391-P171

Probleme : Vous avez depuis plusieurs annees une HP-41C, son module HP-IL, l'imprimante non IL HP-82143A et un XF . Vous venez d'acquerir un HP-71 avec module IL et vous aimeriez bien imprimer vos premieres elucubrations sans vous ruinez d'avantage .

Solution : Reunissez tout votre materiel et utilisez les programmes suivants .
Le programme PRFILE permet de lister des fichiers BASIC, TEXT, DATA ainsi que les fichiers sources FORTH .
PRBAS est une version moins encombrante de PRFILE mais qui imprime uniquement des fichiers BASICs

Avant toute utilisation du 71 et du 41 sur la meme boucle, faites CONTROL OFF sur le 71

Utilisation de PRFILE :
sur la 41 : XEQ"PR71"
suivez ensuite les instructions affichees :
- File to PRINT ?
Entrer le fichier que vous desirez imprimer .
- Type of FILE ?
Entrer le type de fichier a imprimer :
B : fichier BASIC
F : fichier source FORTH
T : fichier TEXT
D : fichier DATA , SDATA

Utilisation de PRBAS :
-sur le 71 : RUN PRBAS
- File to PRINT ?
voir PRFILE
-sur la 41 : XEQ"PR"

NOTA :
Avec un long fichier basic, il peut arriver un TRANSMIT ERR sur la 41 . Ne vous inquietez pas, ne touchez pas votre 71, faites seulement XEQ"PR" sur la 41 .

PRFILE

```
0010 ! Print BASIC
, TEXT & DATA file on HP
-82143A via HP-41C+XF
0020 OPTION BASE 0
@ DIM Q$[8],V$[120],F$[3
],T$[5],V1$[120] @ INTEG
ER A,Y
0030 INPUT 'File to
PRINT ';Q$
0040 INPUT 'Type of
FILE ','B';T$ @ T$=T$[1
,1] @ IF T$='B' THEN TRA
NSFORM Q$ INTO TEXT
0050 DISP 'Printing
 ';Q$ @ OUTPUT :LOOP ;'
α  ';Q$ @ OUTPUT :LOOP ;
'x̄'
0060 OUTPUT :LOOP ;
'α';'------------' @ OUT
PUT :LOOP ;'x̄' @ ASSIGN
#7 TO Q$
0070 ON ERROR GOTO
80 @ READ #7;V$ @ GOSUB
T$ @ GOTO 70
0080 ON ERROR GOTO
90 @ READ #7;V @ OUTPUT
:LOOP ;'*';V @ GOTO 70
0090 OUTPUT :LOOP ;
'Φ' @ ASSIGN #7 TO * @ I
F T$='B' THEN TRANSFORM
Q$ INTO BASIC
0100 DESTROY ALL @
PUT '#43' @ END
0110 'B': OUTPUT :L
OOP ;'α';V$[1,5] @ V1$=V
$[6] @ GOSUB 210 @ OUTPU
T :LOOP ;'x̄' @ RETURN
0120 'F': X=POS(V$,
':')-1 @ IF X<0 THEN A=0
 @ GOTO 170
0130 IF X<1 THEN 15
0
0140 V1$=V$[1,X] @
GOSUB 210
0150 A=POS(V$,' ',X
+3) @ OUTPUT :LOOP ;'x̄'
0160 OUTPUT :LOOP ;
'+';V$[X+1,X+2] @ OUTPUT
 :LOOP ;'α';V$[X+3,A]
0170 X=POS(V$,';')
@ IF X=0 THEN X=A @ GOTO
 190
0180 V1$=V$[A+1,X]
@ GOSUB 210 @ OUTPUT :LO
OP ;'x̄'
0190 IF X+1<LEN(V$)
 THEN V1$=V$[X+1] @ GOSU
B 210
0200 RETURN @ 'T':
'D': V1$=V$ @ GOSUB 210
@ OUTPUT :LOOP ;'x̄' @ RE
TURN
0210 FOR Y=1 TO 6 @
 OUTPUT :LOOP ;'+';V1$[1
+22*(Y-1),22*Y] @ NEXT Y
 @ RETURN
```

PRBAS

```
0010 INTEGER Y @ DI
M Q$[8],V$[120] @ INPUT
'File to PRINT ';Q$ @ T
RANSFORM Q$ INTO TEXT
0020 DISP 'Printing
 ';Q$ @ OUTPUT :LOOP ;'
α  ';Q$ @ OUTPUT :LOOP
;'x̄' @ ASSIGN #7 TO Q$
0030 ON ERROR GOTO
50 @ READ #7;V$ @ OUTPUT
 :LOOP ;'α';V$[1,5] @ V$
=V$[6]
0040 FOR Y=1 TO 6 @
 OUTPUT :LOOP ;'+';V$[1+
22*(Y-1),22*Y] @ NEXT Y
@ OUTPUT :LOOP ;'x̄' @ GO
TO 30
0050 OUTPUT :LOOP ;
'B' @ ASSIGN #7 TO * @ T
RANSFORM Q$ INTO BASIC @
 DESTROY ALL @ PUT '#43'
```

```
01♦LBL "PR71"
02 CF 12
03 REMOTE
04 "RUNPRFILE"
05 OUTA
06 ,010
07 "LOOK YOUR 71"
08 CF 21
09 AVIEW
10♦LBL 15
11 PSE
12 ISG X
13 GTO 15
14♦LBL "PR"
15 SF 21
16♦LBL 00
17 INA
18 ATOX
19 GTO IND X
20♦LBL 01
21 CF 01
22 PRA
23 GTO 00
24♦LBL 02
25 FC?C 01
26 PRBUF
27 SF 01
28 GTO 00
29♦LBL 03
30 CF 01
31 ACA
32 GTO 00
33♦LBL 04
34 CF 01
35 SF 12
36 ACA
37 CF 12
38 GTO 00
39♦LBL 05
40 CF 01
41 PRBUF
42 PRA
43 CF 21
44 "DONE"
45 AVIEW
46 SF 21
47 END
```

Voici quelques utilitaires pour les amis qui ont un HP-71. Il ne s'agit pas d'articles destinés à la publication, mais plutôt d'un mise en ciculation d'éléments de provenance divers qui offrent surtout l'intérêt d'ouvrir des pistes. Ces travaux sont souvent des ébauches. Prière de nous retourner les corrections pour que la progression des travaux continue. Cela est surtout vrai pour les LEX.

Les différentes personnes qui se sont attelées à la tâche:

STEPHANE BARIZIEN (1) 660 32 91 p 1122
Ecole centrale Anthony

MICHEL MARTINET
PIERRE DAVID PPC PARIS

J-JACQUES DHENIN (1) 545 92 80
83, Rue de GERGOVIE
75014 PARIS

OUTILS 71	Ce fichier est difficile à détruire du fait de son nom; il prémunit l'utilisateur contre le BUG décrit dans l'Addendum du manuel d'utilisation de l'interface HP-IL.
REPLACE	LEX de traitement de chaine. Remplace A$ dans B$ par C$
TM	Affiche l'heure
LOOP	Liste les périphériques présents et leurs identifications.
DIR	Imprime le catalogue de la K7, fichiers purgés compris.
VOLUM	Donne les noms des supports magnétiques présents sur la boucle.
VRENAME	Redéfinit le nom du volum.
PCAT	Imprime le cat. du HP 71
NOLBL	Supprime les labels dans un programme BASIC.
FLAGS	Liste les indicateurs binaires armés.
STKLEN	Redéfinit la pile de commande jusqu'à 16 niveaux.
KEYNAME	Donne le nom des touches au fur et à mesure de leur pression.
CAT0	Fournit la liste des mots du basic.
DCDLEX	Désassemble le début d'un LEX.
HEXDUMP	Liste sur 80 colonnes les octets secteurs par secteurs de la K7 ou de la disquette.
MCOPY	Copy secteur par secteur d'un support magnétique sur un autre.
UTILLEX	DPEEK, ESCADDR, STARTUP$, REV$, LDELAY?, CDELAY?, WIDTH?, PWIDTH?
TRIK7	Construit un fichier 'D1' contenant la liste des fichiers (y compris les fichiers purgés) d'un support magnétique.
COPTRIK7	Copie les fichiers dont les noms sont listés dans D1 d'un support magnétique sur un autre. (Utiliser edtext pour sélectionner les fichiers dans D1.).
LISTVARS	Edite la liste des variables de l'environnement principal.
VER35	Effectue une vérification de la ThinkJet.
XCAT	Catalogue automatique du contenu du HP 71.
VARLIST	Edite la liste des variables employées dans un programme.
SYSTCAT	Edite les différents LEX présents dans le HP 71.

XLIST — Edite différents CATalogues.

PENDULE — Affiche l'heure en gros caractères sur la vidéo.

VARREF — Constitue un fichier des variables d'un programme.

ASSFIELD — LEX facilitant l'écriture d'un source pour assembleur.

EVALLEX — Evalue une expression sans renvoyer de résultat.
Ex EVAL FLAG(1,NOT1).

DIFF — Compare 2 programmes.

K71 — Editeur d'octet sur la disquette ou K7.
E permet la recherche d'un secteur, O la recherche d'un octet dans un enregistrement, C la modification d'un caractère. Valider la modification en appuyant sur EndLine puis sur fléche v. Déplacements en utilisant les différentes fléches.

DOT2DOT — Impression de l'écran LCD sur la ThinkJet. Décrit dans CHHU N°1.

FRALEXV — FRALEX sans éditeur de texte en ROMAN 8.

LEXCAT — CATalogue des LEX présents, avec leur N° ID.

Disponible au club - envoyez une cassette...

CASSETTE BARIZI

NAME	S	TYPE	LEN	DATE	TIME
OUTIL 71B		LEX	1	00/00/00	00:00
TM		BASIC	109	02/22/85	20:00
LOOP		BASIC	78	02/22/85	20:00
DIR		BASIC	918	02/22/85	20:01
VOLUM		BASIC	254	02/22/85	20:01
VRENAME		BASIC	296	02/22/85	20:02
PCAT		BASIC	214	02/22/85	20:02
NOLBL		BASIC	830	02/22/85	20:03
FLAGS		BASIC	55	02/22/85	20:03
STKLEN		BASIC	329	02/22/85	20:04
KEYNAME		BASIC	125	02/22/85	20:04
CAT0		BASIC	141	02/22/85	20:05
DCDLEX		BASIC	750	02/22/85	20:05
HEXDUMP		BASIC	373	02/22/85	20:06
MCOPY		BASIC	378	02/22/85	20:06
UTILLEX		LEX	355	02/22/85	20:07
TRIK7 .		BASIC	629	02/22/85	20:07
COPTRIK7		BASIC	200	02/22/85	20:08
LISTVARS		BASIC	1528	02/22/85	20:09
VER35		BASIC	146	02/22/85	20:09
XCAT		BASIC	541	02/22/85	20:10
VARLIST		BASIC	548	02/22/85	20:11
PATTERN		LEX	171	03/08/85	10:00
SYSTCAT		BASIC	1370	02/22/85	20:12
XLIST		BASIC	225	02/22/85	20:13
PENDULE		BASIC	1160	02/22/85	20:13
VARREF		BASIC	1715	02/22/85	20:14
ASSFIELD		LEX	122	02/22/85	20:15
EVALLEX		LEX	46	02/22/85	20:16
DIFF		BASIC	276	02/22/85	20:16
K71		BASIC	1493	02/22/85	20:17
REPLACE		LEX	295	02/22/85	20:53
DOT2DOT		BASIC	633	02/22/85	21:00
FRALEXV		LEX	2027	02/22/85	21:00
LEXCAT		BASIC	430	02/22/85	21:36

```
FRALEXV
ID=5C
 08 REPLACE$ Function
     Parm1   String
     Parm2   String
     Parm3   String
     Parm4   Opt String

ID=5E
 06 CDELAY?  Function

 01 DPEEK    Function
     Parm1   Numeric
     Parm2   Numeric

02 ESCADDR  Function

05 LDELAY?  Function

08 PWIDTH?  Function

04 REV$     Function
    Parm1   String

03 STARTUP$ Function

07 WIDTH?   Function
```

```
DIFF
10 DIM A$[200],B$[200]
20 ASSIGN #1 TO R21 @ ASSIGN #2 TO RENTLINE
30 ON ERROR GOTO 70
40 READ #1;A$ @ READ #2;B$
42 IF A$[1,4]=B$[1,4] THEN 44 ELSE 46
44 IF A$#B$ THEN PRINT "R21:";A$ @ PRINT "RT:";B$ @ PRINT @ GOTO 40 ELSE 40
46 IF VAL(A$[1,4])<VAL(B$[1,4]) THEN PRINT "R21:";A$[1,4] @ PRINT @ READ #1;A$ @
 GOTO 42
48 IF VAL(A$[1,4])>VAL(B$[1,4]) THEN PRINT "RT:";B$[1,4] @ READ #2;B$ @ GOTO 42
70 END
TM
5 DELAY 0
10 ON TIMER #1,1 GOSUB 30
20 A$=KEY$ @ IF A$="" THEN 20 ELSE PUT A$ @ END
30 DISP CHR$(27)&"H"&TIME$[1,2]&" h  "&TIME$[4,5]&" mn  "&TIME$[7,8]&" s" @ RETU
RN

VER35
4 S=SPOLL('%35') @ S=BINAND(S,255)
5 IF S=225 OR S=161 THEN DISP "IMP OK" @ END
10 IF NOT BIT(SPOLL('%35'),5) THEN DISP "PAPIER SVP"
20 IF BIT(SPOLL('%35'),6) THEN DISP "LOW BAT IMP"
30 A$=KEYWAIT$ @ GOTO 4

PENDULE
10 ON TIMER #1,60 GOSUB 50 @ CFLAG 0 @ DISP CHR$(27)&"E"
25 GOSUB 50 @ CALL XYTAB(45,10) @ DISP ""&CHR$(8)&CHR$(10)&CHR$(10)&""
30 A$=KEY$ @ IF NOT LEN(A$) THEN 30 ELSE DISP CHR$(27)&"E" @ PUT A$ @ END
50 CALL TDIG(62,8,5) @ IF FNT(5) THEN RETURN ELSE CALL TDIG(50,8,4)
70 IF FNT(4) THEN RETURN ELSE CALL TDIG(32,8,2) @ IF NOT FNT(2) THEN CALL TDIG(2
0,8,1)
80 RETURN
85 DEF FNT(N)=VAL(TIME$[N,N]) AND FLAG(0,(N=2)+FLAG(0))
90 SUB XYTAB(X,Y) @ DISP CHR$(27)&"%"&CHR$(X)&CHR$(Y);
100 SUB TDIG(X,Y,N) @ ON VAL(TIME$[N,N])+1 RESTORE 1000,1001,1002,1003,1004,1005
,1006,1007,1008
110 FOR I=0 TO 6 @ READ A$ @ CALL XYTAB(X,Y+I) @ DISP USING "#,9A";A$
112 NEXT I @ DISP CHR$(27)&"H";
1000 DATA "  ",        ,        ,        ,        ,        ,"  "
1001 DATA "    ","    ","    ","    ","    ","    ","  "
1002 DATA "  ",        ,"        ","    ","  ",,
1003 DATA "  ",        ,"        ","    ","        ",        ,"
"
1004 DATA "      ","    ","    ",        ,,"      ","      "
1005 DATA ,,," ","        ",        ,"  "
1006 DATA "    ","  ",,,        ,        ,"  "
1007 DATA ,"        ","      ","    ","  ","  ","  "
1008 DATA "  ",        ,        ,"  ",        ,        ,"  "
1009 DATA "  ",        ,        ,"  ","        ","      ","
"
```

```
CAT0
10 DIM A$[2],B$[7],X,B @ B$=""
20 FOR X=126625 TO 127964 STEP 2
30 A$=PEEK$(DTH$(X),1)&PEEK$(DTH$(X-1),1)
40 B=HTD(A$)
50 IF B>35 AND B<91 THEN B$=B$&CHR$(B) ELSE DISP B$ @ B$="" @ X=X+1
60 NEXT X @ BEEP

SYSTCAT
10 SUB SYSTCAT
11 ON ERROR GOTO 'EOF'
20 !
30 DEF FNR$(R$)
40 R1$=""
50 FOR J=LEN(R$) TO 1 STEP -1
60 R1$=R1$&R$[J,J]
70 NEXT J
80 FNR$=R1$
90 END DEF
100 !
110 A=HTD(FNR$(PEEK$("2F571",5)))
120 'NEXT': S$=FNR$(PEEK$(DTH$(A),7))
130 IF S$[4,6]#"BFC" THEN A=A+7+HTD(S$[1,3]) @ GOTO 'NEXT'
140 A=A+7
150 !
160 'LEXID': T$=FNR$(PEEK$(DTH$(A),11))
170 PRINT "ID="&T$[10,11]
180 C=HTD(T$[1,5])
190 M=HTD(T$[8,9])
200 B=C-13
210 B=B+HTD(FNR$(PEEK$(DTH$(B),4)))
220 'LEXENTRY': A$=PEEK$(DTH$(B),16)
230 IF A$[1,2]="FF" THEN 'EOF' ! test for end of LEX file
240 W$="" ! word name
250 L=HTD(PEEK$(DTH$(B-1),1))+1 ! # chrs in name
260 !
270 FOR I=1 TO L STEP 2 ! build name
280 W$=W$&CHR$(HTD(FNR$(A$[I,I+1])))
290 NEXT I
300 !
310 S$=FNR$(PEEK$(DTH$(B+L),2)) ! entry tape
320 IF S$="00" THEN PRINT " -- ";W$;TAB(14);"Spec Word/FFN" @ GOTO 'NXTENTRY'
330 PRINT " ";S$;" ";W$;
340 D=C+(HTD(S$)-M)*9
350 E=HTD(PEEK$(DTH$(D+8),1))
360 W$="Statement"
370 IF E=15 THEN W$="Function" ELSE IF E=0 THEN W$="Word"
380 PRINT TAB(14);W$
390 IF E=15 THEN GOSUB 'FATTRIB' ELSE IF E#0 THEN GOSUB 'SATTRIB'
400 'NXTENTRY': B=B+L+3
410 PRINT
420 GOTO 'LEXENTRY'
430 !
440 'FATTRIB': E=D+1+HTD(FNR$(PEEK$(DTH$(D+3),5)))
450 IF E>1048575 THEN E=E-1048576
460 P=HTD(PEEK$(DTH$(E),1))
470 Q=HTD(PEEK$(DTH$(E+1),1))
480 !
490 FOR I=1 TO Q
500 PRINT "     Parm";STR$(I);TAB(14);
510 E=E-1
520 X=HTD(PEEK$(DTH$(E),1))
530 IF I>P THEN PRINT "Opt ";
540 IF X>11 THEN PRINT "Num or String"; ELSE IF X>7 THEN PRINT "Numeric"; ELSE P
RINT "String";
550 IF MOD(X,4) THEN PRINT "Array";
560 PRINT
570 NEXT I
580 RETURN
590 !
600 'SATTRIB': PRINT '    Legal:";
610 IF E>8 THEN PRINT TAB(14);"In program";
```

```
620 IF MOD(E,8) DIV 4 THEN PRINT TAB(14);"After IF";
630 IF MOD(E,2) THEN PRINT TAB(14);"From KBD";
640 PRINT
650 RETURN
660 !
670 'EOF': IF T$[10,11]#"00" THEN A=A+11 @ PRINT @ GOTO 'LEXID'
680 END SUB

LEXCAT
5 ON ERROR GOTO 150
10 DIM L$[11],H$[2],K$[4],L,I
20 I=1
30 'BOUCLE':
40 L$=XFN92007(DTH$(HTD(XFN92001('BFC'))-4+I*11),11)
50 H$=XFN92006(L$[1,2])
60 DISP 'Id #';H$;'/';STR$(HTD(H$));TAB(15);XFN92006(L$[3,4]);'<tk<';XFN92006(L$
[5,6])
70 K$=XFN92004
80 IF K$='#48' THEN DISP 'Main Table: ';XFN92006(L$[7]) @ GOTO 70
90 IF K$='#47' THEN 60
100 IF K$='#50' THEN I=I-(I#1) @ GOTO 'BOUCLE'
110 IF K$#'#51' THEN PUT K$ @ END
120 IF L$[1,2]#'00' THEN I=I+1
130 GOTO 'BOUCLE'
140 IMAGE "ID #",2A,"/",3D,2X,2A,"<tk<",2A
150 IF ERRN=34 THEN DISP 'FILE "DESAL" NECESSAIRE' ELSE DISP ERRM$

LOOP
10 FOR I=1 TO 30 @ A=DEVAID(I) @ D$=DEVID$(I) @ IF A<0 AND D$="" THEN END
20 DISP USING "2Z,3X,5D,3X,K";I,A,D$ @ NEXT I

DIR
10 U=DEVADDR("TAPE") @ DIM N$[10],T$[2]
20 SEND MTA LISTEN U DDL 4 DATA 0,0 DDL 3 DATA 0 UNL UNT
30 GOSUB 'RDY'
40 SEND MLA TALK U DDT 2
50 ENTER :LOOP USING "#,2X,6A";V$ @ SEND UNL UNT @ PRINT "Volume label : ";V$ @
PRINT
60 SEND UNL MTA LISTEN U DDL 4 DATA 0,2 DDL 3 DATA 0 UNL UNT
65 GOSUB 'RDY'
70 SEND MLA TALK U DDT 2
80 ENTER :LOOP USING "#,10A,2A,20X";N$,T$ @ SEND UNL UNT
89 U$=DTH$(NUM(T$[1,1])*256+NUM(T$[2]))[2]
90 RESTORE @ ON ERROR GOTO 120
100 READ T1$,T2$ @ IF T1$#U$ THEN 100 ELSE 150
120 T2$="("&U$&")" @ IF U$="FFFF" THEN SEND LISTEN U MTA DDL 7 UNL UNT @ END
150 OFF ERROR
151 PRINT USING "10A,2X,8A";N$,T2$
180 SEND UNL MLA TALK U DDT 0
190 GOTO 80
200 'RDY': S=SPOLL(U) @ IF BIT(S,5) THEN 'RDY' ELSE RETURN
250 DATA 0000,Purged,0001,TEXT,E0D0,SDATA,E0F0,DATA,E204,BIN
251 DATA E208,LEX,E20C,KEY,E214,BASIC
252 DATA E0F1,DATA [S],E205,BIN [S],E206,BIN [P],E207,BIN [E],E209,LEX [S]
253 DATA E20D,KEY [S],E215,BASIC [S],E216,BASIC [P],E217,BASIC [P]
254 DATA E088,75BASIC,E089,75LEX,E053,75APPT,E052,75KEY,E08A,75VSW
255 DATA E040,41WA,E050,41KE,E060,41ST,E080,41PR

PCAT
5 RESTORE @ DIM A$[50] @ PRINT CHR$(12);
6 ON ERROR GOTO 100
10 I=1 @ READ A1$ @ IF A1$#"@" THEN A1$=":"&A1$ ELSE END
20 A$=CAT$(I,A1$) @ IF A$="" THEN 10
30 PRINT A$ @ I=I+1 @ GOTO 20
100 IF ERRN=64 THEN 10
1000 DATA MAIN,PORT(0),PORT(.01),PORT(.02),PORT(.03),PORT(.04),PORT(.05),@

NOLBL
10 DELAY 0,0
20 DISP "SUPPRESS LBLS AT SOL?"
30 IF KEYDOWN("Y") THEN F=1 ELSE IF KEYDOWN("N") THEN F=0 ELSE 30
40 DEF FNG$ @ IF FLAG(0) THEN FNG$="'" ELSE FNG$='"'
```

```
50 END DEF
60 DIM A$[128]
70 PURGE TX1 @ PURGE LBLS @ PURGE TX2
80 N=0 @ M=0
90 INPUT "FILE NAME : ";N$ @ TRANSFORM N$ INTO TEXT TX1
100 CREATE DATA LBLS
110 ASSIGN #1 TO TX1 @ ASSIGN #2 TO LBLS
120 ON ERROR GOTO 'E1'
130 READ #1;A$ @ DISP A$[1,4] @ N=N+1 @ P=POS(A$,":") @ IF NOT P THEN 130
140 B$=A$[6,6] @ IF B$="!" OR B$#"'" AND B$#'"' THEN 130
150 IF A$[P-1,P-1]#"'" AND A$[P-1,P-1]#'"' THEN 130
160 PRINT #2;A$[7,P-2],VAL(A$[1,4]) @ M=M+1 @ GOTO 130
170 'E1': IF ERRN#54 THEN DISP "L";ERRL;" : ";ERRM$ @ STOP
180 OFF ERROR @ CREATE TEXT TX2 @ ASSIGN #3 TO TX2 @ RESTORE #1
190 FOR I=1 TO N @ READ #1;A$ @ RESTORE #2 @ IF A$[6,6]="!" OR A$[6,9]="REM " TH
EN 270
200 FOR J=1 TO M @ READ #2;B$,B @ DISP B$;TAB(10);A$[1,4]
210 CFLAG 0
220 P=POS(A$,FNG$&B$&FNG$)
230 IF P AND A$[P+LEN(B$)+2,P+LEN(B$)+2]#":" THEN A$[P,P+LEN(B$)+1]=STR$(B) @ GO
TO 210
240 IF P=6 AND F THEN A$[6,LEN(B$)+8]="" @ GOTO 210
250 P=FLAG(0,FLAG(0)-1) @ IF FLAG(0) THEN 220
260 NEXT J
270 PRINT #3;A$ @ NEXT I
280 ASSIGN #3 TO *
290 PURGE TX1 @ PURGE LBLS @ TRANSFORM TX2 INTO BASIC N$&"2" @ PURGE TX2
```

```
FLAGS
10 FOR I=-64 TO 63 @ IF FLAG(I) THEN DISP USING "#,4D";I
20 NEXT I
25 PUT "#43" @ DISP
```

```
STKLEN
5 INPUT "Nb de niveaux ?";N @ CALL SWEET16(N) @ END
10 SUB SWEET16(X) @ POKE "2F441","1"
20 DEF FNR$(A$)=A$[5]&A$[4,4]&A$[3,3]&A$[2,2]&A$[1,1]
30 X=IP(X) @ IF X<1 THEN X=1 ELSE IF X>16 THEN X=16
40 DIM S$[X*6] @ A=HTD(FNR$(PEEK$("2F576",5)))
50 FOR Y=1 TO X @ S$=S$&"000300" @ NEXT Y
60 E$=FNR$(DTH$(A+X*6)) @ POKE "2F580",E$&E$&E$
70 POKE DTH$(A),S$ @ POKE "2F976",DTH$(X-1)[5]
80 POKE "2F441","0" @ END SUB
```

```
KEYNAME
10 DISP "Press ATTN to exit"
20 POKE "2F441","1" @ DIM A$[4]
30 A$=KEY$ @ IF LEN(A$) THEN DISP "'";A$;"'" ELSE 30
40 IF A$#"#43" THEN 30
50 POKE "2F441","0"
```

```
DCDLEX
10 INPUT "FILE : ";N$ @ O=HTD(ADDR$(N$))
20 IF FNP$(16,4)#"E208" THEN DISP MSG$(63) @ END
30 DISP "FiLeNd at ";FNP1$(32,5)
40 DISP "ID : ";FNP$(37,2);"  Tokens ";FNP$(39,2);" to ";FNP$(41,2)
45 T9=HTD(FNP1$(43,5)) @ T8=HTD(FNP$(48,1)) @ IF T8=15 THEN T8=49 ELSE T8=128
50 T1=HTD(FNP1$(T8,4))-1 @ M=T8+13
60 DISP @ DISP "Keyword";TAB(15);"Token";TAB(25);"Char.";TAB(33);"Runs at"
70 E=T1+HTD(FNP$(M,3)) @ N=(HTD(FNP$(E,1))+1)/2 @ K$="" @ FOR J=1 TO N
80 K$=K$&CHR$(HTD(FNP$(E+2*J-1,2))) @ NEXT J
90 DISP K$;TAB(15);FNP$(E+2*N+1,2);TAB(25);FNP$(M+8,1);TAB(33);FNP1$(M+3,5)
100 M=M+9 @ IF M#T1 THEN 70
110 DISP @ IF FNP$(53,4)="0000" THEN DISP " No MSG Table" @ GOTO 130
120 DISP "MSG Table at ";FNP1$(53,4)
130 DISP @ IF FNP$(57,5)="00000" THEN DISP "No POLL handler" @ GOTO 150
140 DISP "POLL handler at ";FNP1$(57,5)
150 STOP
9000 DEF FNP$(R,L)=FNR$(PEEK$(DTH$(O+R),L))
9010 DEF FNP1$(R,L)=DTH$(R+HTD(FNP$(R,L)))
9020 DEF FNR$(A$) @ L=LEN(A$) @ DIM R$[L] @ FOR I=1 TO L
9030 R$[I,I]=A$[L-I+1,L-I+1] @ NEXT I @ FNR$=R$ @ END DEF
```

```
HEXDUMP
10 U=DEVADDR("TAPE")
20 INPUT "Start Record : ";R$ @ R=HTD(R$)
60 SEND UNL MTA LISTEN U DDL 4 DATA R DIV 256,RMD(R,256) DDL 3 DATA 0 UNL UNT
65 GOSUB 'RDY'
70 SEND MLA TALK U DDT 2 @ R=R*256
80 ENTER :LOOP USING "#,32A";N$ @ SEND UNL UNT
90 IF NOT RMD(R,256) THEN PRINT USING "#,4a,' ":DTH$(R DIV 256)[3] ELSE PRINT "
   ";
100 FOR I=1 TO 32 @ PRINT DTH$(NUM(N$[I,I]))[4]; @ NEXT I @ PRINT
110 R=R+32
180 SEND UNL MLA TALK U DDT 0
190 GOTO 80
200 'RDY': S=SPOLL(U) @ IF BIT(S,5) THEN 'RDY' ELSE RETURN

MCOPY
5 INPUT "Origine : ";A1$ @ T1=DEVADDR(A1$)
6 INPUT "Destination : ";A2$ @ T2=DEVADDR(A2$)
10 SEND LISTEN T1 MTA DDL 4 DATA 0,0 UNL UNT @ GOSUB 'RDY1'
20 SEND LISTEN T2 MTA DDL 4 DATA 0,0 UNL UNT @ GOSUB 'RDY2'
30 SEND TALK T1 LISTEN T2 DDL 2 DDT 2 RDY 96
40 IF BIT(STATUS,7) THEN 40
50 SEND UNL UNT @ GOSUB 'RDY1' @ GOSUB 'RDY2'
60 DISP U1,U2 @ STOP
70 'RDY1': U1=SPOLL(T1) @ IF BIT(U1,5) THEN 'RDY1' ELSE RETURN
80 'RDY2': U2=SPOLL(T2) @ IF BIT(U2,5) THEN 'RDY2' ELSE RETURN

XCAT
5 ON ERROR GOTO 900 @ INTEGER P,I,U,J,V,K,W @ DIM A$[43],C$(11)[9] @ POKE "2F441
","F"
10 C$(1)="MAIN" @ I=2 @ FOR P=0 TO 5 @ GOSUB 1000 @ NEXT P
15 FOR P=100 TO 400 STEP 100 @ GOSUB 1000 @ NEXT P @ I=1 @ U=1
20 DISP CHR$(27)&"E" @ FOR P=1 TO 23
30 A$=CAT$(I,":"&C$(U)) @ IF NOT LEN(A$) THEN U=U+1 @ I=1 @ GOTO 30 ELSE IF P=1
THEN W=U @ K=I .
40 I=I+1 @ DISP A$ @ NEXT P
50 A$=KEY$ @ IF NOT LEN(A$) THEN 50 ELSE IF A$="#43" THEN POKE "2F441","0" @ PUT
 A$ @ END
70 IF A$="#51" THEN I=I+1 @ P=V @ V=W @ W=P @ P=J @ J=K @ K=P @ GOTO 20
72 IF A$#"#50" THEN BEEP @ GOTO 50 ELSE U=V @ I=J @ GOTO 20
900 IF ERRN=64 THEN U=U+1 @ I=1 @ GOTO 30
910 IF ERRN#28 THEN 920 ELSE IF P#1 THEN 50 ELSE U=W @ I=K @ GOTO 20
1000 C$(I)="PORT("&STR$(P/100)&")" @ I=I+1 @ RETURN

XLIST
10 CFLAG 0,1 @ ON ERROR GOTO 200 @ DIM A$[132]
20 IF FLAG(0) THEN INPUT "PLIST ";A$ ELSE INPUT "LIST ";A$
30 IF NOT LEN(A$) THEN A$=STR$(FLAG(0,NOT FLAG(0))) @ GOTO 20
40 ASSIGN #1 TO A$
50 READ #1;A$ @ SFLAG 1 @ IF FLAG(0) THEN PRINT A$ ELSE DISP A$
60 GOTO 50
200 IF ERRN#54 THEN 205 ELSE IF FLAG(1) THEN 210 ELSE PURGE A$ @ LIST A$
205 DISP ERRM$ @ BEEP
210 DESTROY A$


VARREF
10 DESTROY F$,P
20 INPUT "TEXT FILE ? ";F$
30 INPUT "PRINTER WIDTH ? ";P
40 P$=PEEK$("2F958",2)
50 PWIDTH P
60 DISP "WORKING"
70 FOR I=1 TO P @ PRINT "="; @ NEXT I
80 PRINT "Variable Cross Reference";
90 IF P>25+LEN(F$) THEN PRINT TAB(P-LEN(F$)):F$ ELSE PRINT @ PRINT TAB((P-LEN(F$
))/2);F$
100 FOR I=1 TO P @ PRINT "="; @ NEXT I @ PRINT
110 PWIDTH P DIV 5*5
120 CALL VARIABLE(F$)
130 POKE "2F958",P$ @ PUT "$38"
140 END
```

```
150 !
160 SUB VARIABLE(F$)
170 ASSIGN #255 TO F$
180 DIM L$[100],L0$[4],A$[26],D$[23]
190 INTEGER I,J,K,L,M,O
200 O=FLAG(-16,1) @ F0=FLAG(0)
210 D1$=PEEK$("2F6DC",2) @ STD
220 A$="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
230 D$="+-*/( )<>[]/\?#@%^& :;=,"
240 J=0
250 !
260 'NEXTLINE': ON ERROR GOTO 'ERROR' @ READ #255;L$ @ OFF ERROR
270 L$=UPRC$(L$) @ I=6 @ L0$=L$[1,4]
280 !
290 'NEXTCHAR':
300 IF L$[I,I]="!" OR L$[I][1,4]="REM " THEN "NEXTLINE"
310 IF L$[I][1,4]="SUB " AND L$[I-2,I-2]#"D" THEN GOSUB "SBPRGM"
320 IF L$[I,I]="'" OR L$[I,I]='"' THEN I=POS(L$,L$[I,I],I+1)+1
330 IF NOT (POS(A$,L$[I,I]) AND NOT POS(A$,L$[I+1,I+1])) THEN 'REPEAT'
340 K=I
350 'VARS': IF NOT POS(D$,L$[I,I]) AND I<=LEN(L$) THEN I=I+1 @ GOTO 'VARS'
360 SFLAG 0
370 FOR L=1 TO J
380 IF N$(L)=L$[K,I-(L$[I,I]#"(")] THEN 'OUT'
390 NEXT L
400 CFLAG 0
410 'OUT': IF FLAG(0,0) THEN "PUTIN"
420 J=J+1
430 DIM N$(J)[4]
440 N$(J)=L$[K,I-(L$[I,I]#"(")]
450 CREATE TEXT "VAR"&STR$(L)
460 ASSIGN #L TO "VAR"&STR$(L)
470 'PUTIN': PRINT #L;CHR$(VAL(L0$[1,2]))&CHR$(VAL(L0$[3]))
480 'REPEAT': IF NOT POS(D$,L$[I,I]) AND I<=LEN(L$) THEN I=I+1 @ GOTO 'REPEAT'
490 I=I+1
500 IF I<=LEN(L$) THEN "NEXTCHAR" ELSE "NEXTLINE"
510 'SBPRGM': IF J THEN GOSUB "PRINT"
520 PRINT "-------" @ PRINT STR$(VAL(L0$))&" "&L$[6] @ PRINT
530 RETURN
540 'PRINT': IF J=0 THEN RETURN
550 FOR M=1 TO J @ ASSIGN #M TO * @ NEXT M
560 CALL PRINTOUT(N$(),J)
570 RETURN
580 'ERROR':
590 IF ERRN#54 THEN DISP ERRM$ @ BEEP 1000,.25 @ OFF ERROR ELSE GOSUB 'PRINT'
600 O=FLAG(-16,O) @ F0=FLAG(0,F0) @ POKE "2F6DC",D1$
610 END SUB
620 !
630 SUB PRINTOUT(N$(),L)
640 DIM A$[2],B$[2],T$[4]
650 INTEGER I,J,K
660 FOR I=1 TO L
670 T$="x"
680 FOR J=1 TO L
690 IF N$(J)<T$ THEN T$=N$(J) @ K=J
700 NEXT J
710 N$(K)="x" @ B$=""
720 IF T$#"x" THEN PRINT T$
730 ASSIGN #K TO "VAR"&STR$(K)
740 'MORE': ON ERROR GOTO 'OUT' @ READ #K;A$ @ OFF ERROR
750 IF A$#B$ THEN PRINT USING "#,X0DDD";NUM(A$)*100+NUM(A$[2])
760 B$=A$
770 GOTO 'MORE'
780 'OUT': ASSIGN #K TO *
790 PURGE "VAR"&STR$(K)
800 PRINT @ PRINT
810 NEXT I
820 L=0
830 END SUB
```

FORTH

Bernard MORISSEAU Ingenieur ECP
22 Avenue de FRIEDLAND
75008 PARIS

Toulon le 21 juillet

Cher camarade

Récent adhérent du PPC-T, j'apprends à la lecture de Micro revue N°1 que le numero de septembre sera en grande partie consacré au langage Forth dont je suis depuis quelques mois un utilisateur convaincu, comme je vous en faisais part lors de mon adhésion (à ce propos, je serais heureux de voir mon nom figurer dans la liste du groupe PPC-T Forth)

Bien qu'étant en vacances et privé de ma machine à écrire, je n'attends pas mon retour à Paris pour vous faire part de quelques reflexions sur l'avenir des études communes du PPC sur ce langage

Il s'agit certes d'un langage offrant des possibilités très vastes, quelque soit le domaine pour lequel on l'utilise et le materiel sur lequel il est implanté, mais cette variété ne peut manquer de constituer un grave ecueil et de rendre difficile la communication entre membres du club si un minimum de discipline n'est pas respecté par les différents correspondants.

Il me paraît indispensable de partir des sources, c'est à dire du noyau Forth et de ne traiter des extensions que d'une manière très progressive pour éviter que les utilisateurs de HP75, HP71, Apple, Jupiter ACE ou autres materiels (personnellement j'utilise FORTH sur un ORIC 1) ne présentent des communications inexploitables pour ceux qui travaillent sur des materiels différents

Je souhaiterais vivement pouvoir disposer, comme cela semble être envisagé, d'un FORTH pour HP 41, mais ne vois pas très bien comment cela peut être realisé.

En effet, FORTH a été conçu pour fonctionner sur micro ordinateurs munis d'interface video et offrant la possibilité de sauvegarde de programmes source et fichiers sur mémoire de masse

Le X function de la HP41 et les modules d'extension mémoire pourront-ils resoudre ce problème ?
D'autre part, bien des utilisateurs de HP41 ne disposant pas de lecteurs d'Eprom, la réalisation d'une ROM serait à mon sens indispensable.

Un autre problème a trait à l'utilisation, indispensable pour les applications scientifiques, de nombres réels en virgule flottante.
Quelle notation adopter ? La notation binaire des micro ordinateurs ou la notation DCB de la HP41 ? La première ayant le grand avantage de permettre de travailler dans une base quelconque, la deuxième étant sans doute plus facile à implanter sur HP.

La cassette FORTH, d'origine anglaise, vendue pour l'oric contient le FIG FORTH, des écrans d'extension éditeur et assembleur excellents (assembleur 6502) ainsi que des extensions d'instructions musicales et graphique haute résolution.

J'ai par ailleurs utilisé un certain nombre de routines de la ROM Basic, pour la création de variables virgule flottante, les opérations arithmétiques et les fonctions trigonométriques. Ceci m'a permis, avec utilisation de routines de calcul matriciel, de réaliser des animations graphiques d'objets en trois dimensions.

Je rencontre cependant beaucoup de difficultés à manipuler les buffers et les blocs pour la sauvegarde de fichiers sur cassette, les documents dont je dispose manquant totalement de clarté sur ce sujet.

Le Forth est un langage de spécialiste, mais remarquable par sa rapidité, sa possibilité qu'il offre de simplifier l'utilisation de l'assembleur, et sa philosophie qui permet la création d'objets et de vocabulaires variés adaptés à tous les besoins et permettant une structuration parfaite des applications

Cette philosophie est très voisine de celle de la HP 41, ce qui permet d'espérer une heureuse combinaison de l'une et l'autre et justifie pleinement l'intérêt que semble devoir lui porter notre association.

PRELIMINARY DATA

QUESTIONS FORTH PROGRAMMERS ASK ABOUT THE NOVIX 4000

WHAT IS THE NOVIX 4000?

The 4000 is a high-speed microprocessor, the first in a family of Novix super high-speed processing engines. Like other microprocessors, the 4000 is versatile and programmable, yet in performance it compares with discrete logic systems, which are costly and time-consuming to design.

Ordinary Forth systems emulate the Forth model with software. But the 4000 is a realization of the Forth machine in silicon. The return stack and the data stack are each implemented via a 16-pin data bus to separate external memory regions (no more run-away stacks!). The chip executes subroutine-threaded code as its normal mode of operation, eliminating the need for a "NEXT routine" in software, or even in microcode. (NEXT is the heartbeat of Forth; the routine that manages the execution of the threaded code.)

HOW FAST WILL IT RUN?

As of this writing (Nov. '84) the 4000 has been designed and is now being manufactured in prototype form by Mostek. Until we receive the first prototypes, we cannot make any guarantees about clock speed.

Nevertheless, we are expecting the chip to operate at 10MHz, with one machine instruction being executed per clock cycle. Many instructions are combinable within a single opcode; hence we expect code to run well over 10 million *operations* per second.

We expect the 4000 to execute Forth code roughly 100 times faster than today's generation of microprocessors can run Forth.

And 10MHz is nothing compared with what's to come! The prototype is being implemented using standard gate-array technology; no special optimization for speed has been incorporated. Second-generation versions using ECL technology or high-speed custom CMOS should give us another order of magnitude improvement in speed.

WHAT MAKES IT FASTER THAN OTHER MICROS?

One reason for the chip's incredible performance compared with other micros is its lack of internal microcode; it is "externally microcoded." Due to the 4000's full 16-bit instruction field, individual bits directly control chip functions with a minimum of decoding.

The effect is as if you were writing microcode using a high-level, fully-structured, modular-design language.

The elimination of internally-microcoded instructions is made possible by the chip's (and Forth's) simplicity. As RISC (reduced instruction set computers) machines are now proving, more work can be done in less time when using a small set of simple, fast instructions, than when using a larger set of slower, more powerful instructions.

The Forth architecture itself contributes to the 4000's performance. Because the data and return stacks are implemented in hardware, and because the processor is set up to execute threaded code, much of the "housekeeping" of moving data and negotiating jumps, loops and subroutine calls can occur *in parallel* with the operation of the ALU. Thus, many of the machine instructions can be executed simultaneously (within the same clock cycle). The combinations "SWAP −", "OVER SWAP −", "@ +", "DUP @ SWAP nn +" (incrementing I/O fetch) are just a few examples.

Because the full 16-bit opcode field allows instruction bits to directly control internal logic with very little decoding, these combinations can be assembled into single opcodes by a "smart" Forth compiler.

An indication of the 4000's unique parallelism is its 124 external pins. Most micros have 40 or 64 pins.

The top two items on the data stack reside in on-chip registers (T and N) at all times. A push or pop causes the appropriate exchange with data stack memory.

The chip operates on a "decode and load" instruction cycle. In execution of straight Forth code, the instruction register (called "L") will contain the next instruction to be executed by the time the current instruction is completed. If the next instruction is a jump, the destination will appear on the address bus before the next clock cycle.

The **TIMES** instruction allows the instruction latched in L to be repeated once per cycle until a count (held on the return stack) becomes zero.

HOW IS IT DIFFERENT FROM OTHER "FORTH CHIPS"?

Rockwell's R65F11 has been called a "Forth chip." It is essentially a 6502 microprocessor with a Forth nucleus embedded in on-chip ROM. This nucleus, like that of any conventional Forth system, consists of a combination of high-level definitions and code definitions. The code definitions incorporate normal 6502 instructions; the Forth virtual machine is being emulated.

In contrast, the 4000 is an actual realization of the Forth machine in silicon. The data stack and the return stack are real, they are not emulated. The "NEXT routine" is no longer an instruction; it is the *background* to the chip's every operation.

The 4000, like the other so-called Forth chips, offers the user the convenience of a high-level language, but at hundreds of times the speed.

WHAT INSTRUCTIONS ARE INCLUDED AMONG THE OPCODES?

The 4000's instruction set includes the stack manipulation operators DUP, SWAP, OVER and DROP, >R, R>, and I; addition, and subtraction (with or without carry), multiplication-steps, division-steps (16 of these steps are required for a 16-bit multiply or divide), even square root, ORing, ANDing, XORing, left-and-right logical shifts, jump instructions, a repeat instruction, data fetch and store, literal data fetch, extended-address, local and internal data fetch and store.

But the most significant instruction of all is one that you won't find on any other processor: it is the single-cycle jump to subroutine. If an opcode contains a "0" in its most-significant bit, then the remaining 15 bits are interpreted as a word address, where execution continues as a subroutine on the very next clock cycle. Traditional colon definitions, instead of consisting purely of lists of addresses, now consist of lists of opcodes—some of which are machine instructions such as DUP and +, and the rest of which are single-cycle jumps to subroutine.

Since all definitions are, in effect, executing "machine code," there is no longer a need for the traditional "code field" in dictionary headers. The processor uses direct, not indirect, threaded code, yet allows more compact code.

The "return from subroutine" operation is controlled by a single bit in the opcode field, which in most cases can be combined with existing opcodes for zero time cost.

Philosophically, it can be said that the driving force behind Forth is "optimization of subroutine calls" (through the use of threaded code via the return stack, and the simple passing of data arguments via the data stack). The effect of this optimization is to change the very way we think about programming.

The 4000 is the next logical step in the Forth progression: a chip that is optimized for subroutine calls to the utmost degree possible—one subroutine call per clock cycle.

HOW WIDE ARE THE BUSSES?

All registers in the 4000 are 16-bits wide. The address bus to main memory is 16 bits, enabling the processor to access a full 64K *words* of memory (128K bytes). In addition, extended address instructions are available for accessing a full 4 Megabytes of main memory, via a 5-bit field embedded within fetch and store instructions.

The data bus for main memory is 16-bits wide as well, for single-cycle continuous fetches or stores. (Performed independently, fetches and stores are two cycle operations.)

What makes the 4000 unusual is that the data stack and return stack each possess a 16-bit data bus and an 8-bit address bus. If we take into account the 16-bit I/O port along with the main memory bus and the two stack busses, the 4000's four-way addressing effectively yields a *64-bit data path*, with the expected corresponding increase in performance.

WHAT OTHER FEATURES CAN YOU TELL ME ABOUT?

The 4000 has a set of "local data" fetch and store operations, using a 5-bit imbedded literal field in the opcode to access the first 32 words of main memory in a single instruction (two cycles). Thus low memory can be treated as a set of 32 fast 16-bit pseudo-registers.

In addition, the chip features 15 true internal registers which can be accessed with the single-clock "internal data" fetch and store operations. These registers contain the stack pointers, I, the "times" count, arithmetic registers, program counter, and four special registers for each of the I/O ports: data, mask, direction, and 3-state. An additional read-only register contains hex FFFF for the quick generation of logical "true."

One-level interrupt capability is provided via a single pin, causing execution of interrupt code at octal 40.

The operation of the processor is asynchronous. Since it is implemented in static logic, it may be operated at any clock rate that does not exceed the maximum clock frequency.

WHAT APPLICATIONS IS IT DESIGNED FOR?

The 4000 is a general purpose microprocessor, capable of tackling any problem that Forth is capable of (in other words, any problem at all). Its blazing speed makes it appropriate for applications that might otherwise be designed in MSI or bit-slice, but with a vastly reduced development time. Such applications include signal processing and graphics.

Novix also provides customized versions of the 4000 architecture for dedicated applications, where specialized capabilities are added as needed. One beauty of the Forth architecture is that it can be applied to an infinite variety of chip configurations—all with equal success. Novix is committed to providing custom versions of the chip for any high-volume, high-speed application.

WHO DESIGNED IT? WHO IS NOVIX?

The 4000 architecture was designed by Charles H. Moore, the creator of Forth, and Robert Murphy of Torric Corp. in Scottsdale, Arizona. The prototype chip is being manufactured by Mostek.

Novix is a partnership consisting of Sysorex International and Technology Industries.

WHEN, AND IN WHAT FORM WILL THE 4000 BE AVAILABLE?

We currently expect the first prototype to appear in silicon in early January 1985, for internal testing. Assuming this run meets our performance expectations, a small quantity will be available at that time, complete with an evaluation/development board that will incorporate a PROM-based Forth operating system with fast RAM and serial port. Full-custom versions are currently being planned.

For more information, contact John Golden, General Manager, Novix, 981 University Avenue, Suite A, Los Gatos, California 95030, or phone 408/399-2325.

M. Olivier SINGLA
Ecole Normale d'Instituteurs
Bd. Montauriol
82000 MONTAUBAN
tél: (63) 03-07-27
no. personnel: (63) 66-10-90

Montauban, le 27 avril 85

Cher collègue,

Suite à notre entretien téléphonique du 26 courant, je vous envoie ma version de FORTH, 'prochainement' distribuée par le CNDP.... Notez que cette version fonctionne sous le système d'exploitation EDEN, et est ainsi indépendante du matériel (micral 8022G, non G, Ix 529, Ix549 ou sil'z II. Je ne souhaite pas en rester là car je projette de réaliser une implantation de FORTH sur MO5, dans le cadre du nano-réseau, mais les négociations d'ores et déjà engagées avec l'ADI semblent mal parties (problèmes financiers...ou pressions de Thomson qui commercialise son FORTH...?).

Je vous envoie également un article que j'ai envoyé à l'EPI, et qui devrait paraître dans le numéro de Juin. Peut-être peut-il s'insérer dans le cadre du bulletin de l'association que vous présidez (et à laquelle j'adhérerai avec plaisir).

S'il est encore temps, demandez à faire le stage FORTH en septembre: nous serons ainsi entre gens de bonne compagnie !

Meilleures salutations,

Olivier SINGLA

Initiation au langage FORTH - volet # 1

1. Préliminaire

L'objet de cette série d'articles consacrés à l'initiation au FORTH n'a pour objectif que de vous aider dans l'appropriation de ce fabuleux langage, et principalement au travers de la pratique de la version de FORTH disponible sur les machines 8 bits z80 agrées par l'Education Nationale. Nous renvoyons les lecteurs au numéro de l'EPI de décembre 84 pour les généralités sur le langage FORTH, et une description rapide de *zFORTH* (qui devrait donc être distribué par le CNDP quand cet article paraîtra). Cette série d'articles s'adresse à tout le monde, mais nous supposerons néammoins que les lecteurs potentiels ont une expérience de la programmation (quel que soit le langage), et qu'ils connaissent la signification de termes tels que: bits, octets, adresse-mémoire, etc... En outre, pour ceux qui disposeraient de *zFORTH* et de son manuel de référence, nous les invitons fermement à lire et à pratiquer le chapitre IV ("Notes pour travailler un programme sous *zFORTH*"). Ceci étant dit, lançons-nous dans l'aventure sans plus attendre !

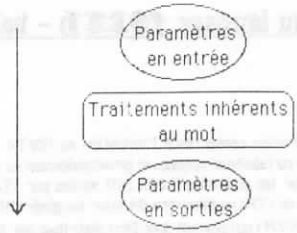
2. Les différentes couches de logiciels

Il est très important de situer le langage par rapport aux autres logiciels pouvant cohabiter dans la machine. Nous parlerons donc en terme de **couches de logiciels** , représentées par le shéma suivant:

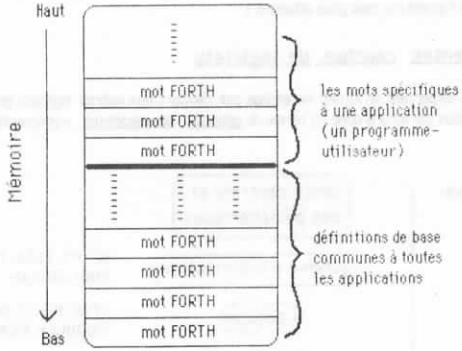
Bien sûr, dans chaque élément, nous pourrons retrouver plusieurs couches de logiciels. C'est le cas par exemple pour un système d'exploitation tel que CP/M avec 4 couches de logiciels (TPA, CCP, BIOS et BDOS), pour un langage ou un programme d'application.

3. Les composantes de base de FORTH

Le premier concept à définir est celui de **MOT** (encore appelé **DEFINITION**). Un mot FORTH correspond à une parcelle de programme. Nous pourrons le définir par le shéma suivant:



Un "programme" FORTH sera donc constitué par un ensemble cohérent et ordonné de plusieurs définitions ou mots. Les définitions FORTH sont regroupées dans ce qui est appelé le **DICTIONNAIRE** . On peut représenter le dictionnaire FORTH de la manière suivante:



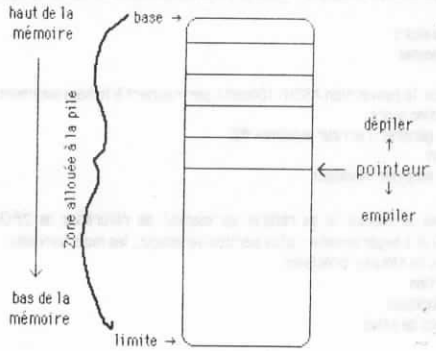
On notera que les nouvelles définitions viendront s'ajouter en haut du dictionnaire: celui-ci croît donc vers le haut de la mémoire. Nous pouvons déjà faire la distinction entre 2 couches de mots: la couche des mots disponibles après le chargement de votre version du langage FORTH, et la couche des mots que vous pourrez définir et qui constitueront votre ou vos programmes. La première couche contiendra des mots standards que l'on retrouvera quelle que soit la version de FORTH.

On peut remarquer une analogie certaine avec des langages procéduraux tels que PASCAL, LSE ou LOGO (un programme = un ensemble de procédures).

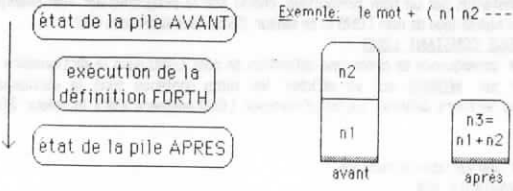
La première différence vient dans la définition elle-même d'un mot FORTH : au lieu d'utiliser un jeu d'instructions figé, avec une syntaxe spécifique pour chaque instruction, on fera simplement référence à d'autres mots FORTH, *déjà existants* ... Ainsi, nous pouvons dire, shématiquement, qu'une définition FORTH est constituée de réferences à un ensemble d'autres définitions FORTH. Première conséquence: on peut dire que FORTH est un langage globalement sans syntaxe, dans la

mesure où les mots référencés sont tapés librement, avec au moins un espace séparateur (pour que le système FORTH puisse les isoler). Si syntaxe il y a à respecter, cela ne dépend que du mot alors invoqué... Deuxième conséquence: une définition FORTH ne pourra faire référence qu'à des mots FORTH existants (déjà définis) lors de la création de cette définition.

La deuxième différence tient dans la manière de passer les paramètres entre les différents mots. Est privilégiée l'emploi d'une pile, appelée **PILE DES PARAMETRES**. Comment définir une pile ? Disons que c'est une structure de donnée à partir de laquelle deux types d'accés sont possibles : empiler un objet ou dépiler un objet. La pile est utilisé par tous les langages de programmation (exemple: en BASIC, chaque fois qu'un GOSUB est rencontré, est empilée sur la pile de retour l'adresse de retour au programme appelant, et chaque fois que RETURN est rencontré, est alors dépilée l'adresse de retour). Le FORTH sur lequel nous allons travailler étant implanté sur un micro-ordinateur architecturé autour d'un micro-processeur 8 bits avec une capacité d'adressage de 64 Koctets (compteur ordinal sur 16-bits), les objets manipulés sur la pile seront, de manière générale, des MOTS de 16-bits. Décrivons la structure et le fonctionnement de la pile des paramètres en FORTH grâce au shéma suivant:



La plupart des mots FORTH seront ainsi définis par rapport à la pile. Cela peut être illustré par le shéma suivant:



Nous pouvons remarquer comment est présentée, de manière symbolique, l'action d'un mot FORTH sur la pile des paramètres: (état avant --- état après). Par convention, n désignera un entier codé sur 16-bits (soit un entier compris entre -32768 et +32767, devinez pourquoi...).

4. Opérations élémentaires.

Nous avons défini dans le chapitre précédent le mot + : celui-ci dépile 2 entiers 16-bits, effectue leur somme, et empile alors le résultat. Le mot . (point, mais prononcez "dot") affichera alors ce nombre sur le terminal.

```
123 -10 + .  113 ok
```

Expliquons brièvement les mécanismes mis en jeu lorsque nous validons après avoir tapé ceci: chaque mot est isolé, l'espace étant considéré comme le séparateur, puis est recherché dans le dictionnaire. Si le mot est trouvé, son exécution est alors invoquée, mais si la recherche a échoué, le système tente une conversion ASCII->binaire, par rapport à la base courante (pour le moment la base 10). Si cette conversion s'avère impossible, une erreur est générée («mot non trouvé et nombre incorrect»), sinon la valeur trouvée est empilée sur la pile des paramètres. Le shéma suivant illustre l'algorithme utilisé par l'interprète en état exécution (pour chaque mot isolé):

```
si trouvé alors
  | l'exécuter
sinon
  | tenter la conversion ASCII->binaire par rapport à la base courante
  | si échec alors
  |   | générer l'erreur système #0
  | sinon
  |   | empiler la valeur
```

Nous invitons le lecteur à se référer au manuel de référence de *zFORTH* ("Glossaire rapide", en annexe G et à expérimenter, plus particulièrement, les mots suivants:

- arithmétique en simple-précision
- entrées-sorties
- opérations logiques
- manipulations de piles

5. Variables et constantes numériques

FORTH offre, comme la plupart des langages de haut-niveau, la possibilité de manipuler des variables et des constantes par un nom symbolique, choisi par le programmeur. Par exemple, pour créer une constante numérique de nom LONG et de valeur 365, nous tapertons ceci:

```
365 CONSTANT LONG
```

Ceci a pour conséquence de créer une définition de nom LONG dans le dictionnaire. On peut d'ailleurs vérifier par **WORDS** qui va afficher les mots contenus dans le dictionnaire (en commençant par les derniers définis). Le fait d'invoquer LONG amènera alors la valeur 365 sur la pile.

Pour créer une variable de nom AGE :

```
VARIABLE AGE
```

Une définition de nom AGE est alors crée dans le dictionnaire, et son invocation amènera sur la pile, non la valeur de la variable, mais l'adresse de cette valeur. Deux mots vont nous permettre de déposer une valeur à cette adresse, ou à l'inverse de la lire. Ce sont:

@ (fetch) (adr --- n) et ! (store) (n adr ---)

Par exemple:

24 AGE ! (initialise la variable AGE à la valeur 24)
AGE @ . (lit la valeur de AGE et l'affiche)

De ceci, nous pouvons tirer 2 conclusions importantes...

Le première est que tout en manipulant sur la pile des valeurs entières codées sur 16-bits, l'utilisation qui en faite par différents mots permet de les considérer comme des valeurs, des adresses mémoire, etc... Remarquons au passage que bien que manipulant l'adresse de la variable, nous pouvons parfaitement ignorer oû elle se situe physiquement en mémoire : les mots @ et ! rendent transparents les accès à une variable.

La deuxième est que, à partir d'un mot FORTH (ici VARIABLE et CONSTANT), nous avons crée d'autres mots dans le dictionnaire. Les mots VARIABLE et CONSTANT seront appelés des MOTS GENERATEURS. FORTH offre ainsi la possibilité remarquable de se définir ses propres mots générateurs. Pour cela, il faut donner une réponse aux 2 questions suivantes:

☞ que faire lors de la création du mot dans le dictionnaire ?

☞ que faire lors de l'invocation des mots générés ?

Ceci nous permet de dire que le contexte FORTH nous permet d'écrire un compilateur (ou plutôt des compilateurs...) de manière très simple, en éliminant tous les problèmes syntaxiques qui peuvent se poser dans un langage compilé traditionnel.

6. Créons d'autres mots FORTH

Supposons que nous désirions créer un mot FORTH qui calcule le complément à 100 d'une valeur donnée sur la pile. La définition en serait la suivante:

```
: COMPLEMENT    ( n1 --- n2 )
  100 SWAP - ;
```

Nous vous conseillons de regarder dans le manuel de référence de *zFORTH* le fontionnement des mots suivants : (SWAP - ;

Si vous ne comprenez toujours pas, laissez tomber le FORTH et revenez à votre LSE chéri...

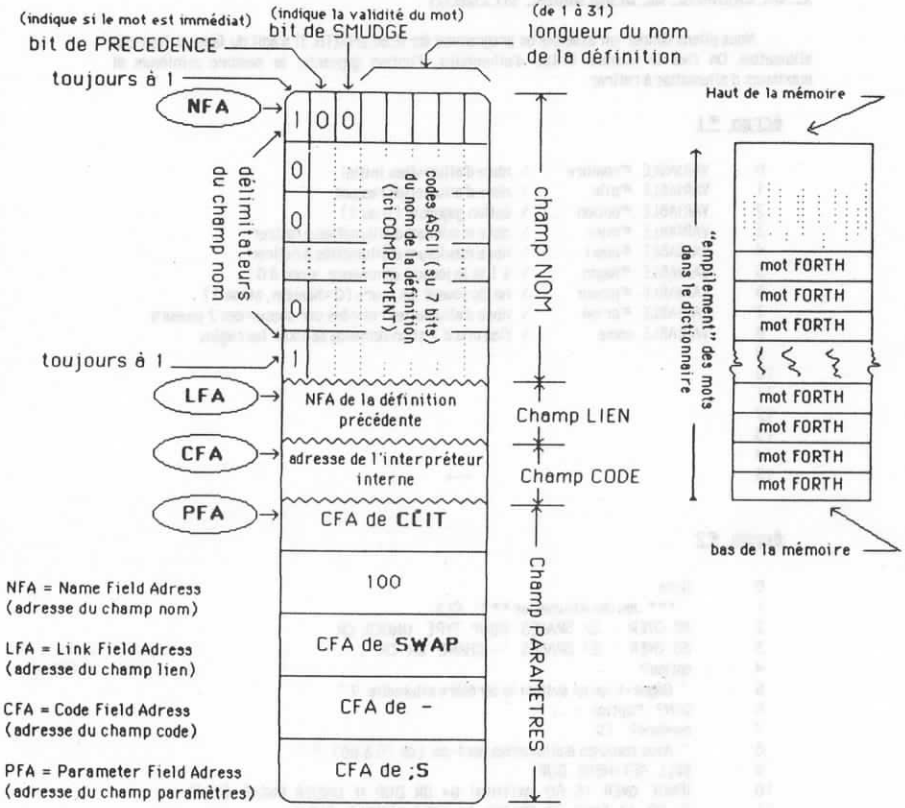
Le rôle du : est de créer l'entête du mot COMPLEMENT dans le dictionnaire (avec le chaînage aux mots précédents), puis de faire basculer le système en état COMPILATION. Résumons le fonctionnement de l'interprète FORTH en état compilation (pour chaque mot qui a été isolé):

```
rechercher le mot dans le dictionnaire
si trouvé alors
  | si mot immédiat alors
  |    | l'exécuter
  | sinon
  |    | "compiler" son adresse au sein de la définition en cours de création
sinon tenter la conversion en binaire (ASCII->binaire)
si succès alors
  | "compiler" CLIT, LIT ou DLIT et la valeur sur 8, 16 ou 32 bits
sinon
  | générer l'erraur-système #0 ("mot non trouvé et nombre invalide")
```

Analysons, à l'aide du shéma suivant, la structure des mots dans le dictionnaire, et le code généré par le compilateur FORTH:

Structures des mots dans le dictionnaire

(modèle zFORTH pour z80 sous EDEN)



Définition... : COMPLEMENT
100 SWAP -

(Dans cet exemple, l'interpréteur interne est l'interpréteur de mots FORTH de haut-niveau, soit la procédure d'exécution compilée par :)

7. Un exemple de programme en FORTH

Nous allons donner un exemple de programme écrit en zFORTH. Il s'agit du fameux jeux des allumettes. On fixe le nombre initial d'allumettes, l'option gagnante, le nombre minimum et maximum d'allumettes à retirer.

écran #1

```
 0 :  VARIABLE #nombre     \ nbre d'allumettes initial
 1 :  VARIABLE #pile       \ nbre d'allumettes restant
 2 :  VARIABLE #option     \ option gagnante (0 ou 1)
 3 :  VARIABLE #mini       \ nbre minimum d'allumettes à retirer
 4 :  VARIABLE #maxi       \ nbre maximum d'allumettes à retirer
 5 :  VARIABLE #begin      \ à 1 si le joueur commence, sinon à 0
 6 :  VARIABLE #joueur     \ no. du joueur en cours (0=humain, sinon 1)
 7 :  VARIABLE #prise      \ nbre d'allumettes retirées par chacun des 2 joueurs
 8 :  VARIABLE same        \ flag mis à 1 si on demande de fixer les règles
 9 :
10 :
11 :
12 :
13 :
14 :
15 :                       -->
```

écran #2

```
 0 :  : titre
 1 :    " *** Jeu des Allumettes ***"  CLS
 2 :    80 OVER - 2/ SPACES DDUP TYPE UNDER CR
 3 :    80 OVER - 2/ SPACES "- CHARS CR CR ;
 4 :  : option?
 5 :    ." Gagne-t-on en évitant la dernière allumette ? "
 6 :    O/N? #option ! ;
 7 :  : nombre? CR
 8 :    ." Avec combien d'allumettes part-on (de 15 à 60) ? "
 9 :    BELL GET/NBRE DUP
10 :    IFNOT OVER 15 60 [WITHIN] 0= OR DUP IF UNDER ENDIF ENDIF
11 :    IF CR 11 EMIT 79 SPACES 11 EMIT MYSELF ENDIF
12 :    #nombre ! ;
13 :  : begin? CR
14 :    ." Désirez-vous jouer en premier ? " O/N? #begin !
15 :                                  -->
```

écran #3

```
 0  :  : mini? CR
 1  :    ." Quel est le minimum à retirer (de 1 à 7) ? "
 2  :    BELL GET/NBRE DUP
 3  :    IFNOT OVER 1 7 [WITHIN] 0= OR DUP IF UNDER ENDIF ENDIF
 4  :    IF CR 11 EMIT 79 SPACES 11 EMIT MYSELF ENDIF
 5  :    #mini ! ;
 6  :  : maxi? CR
 7  :    ." Quel est le maximum à retirer (de " #mini @ 1+ .
 8  :    ." à 9) ? " BELL GET/NBRE DUP IFNOT
 9  :      OVER #mini @ 1+ 9 [WITHIN] 0= OR DUP IF UNDER ENDIF
10  :    ENDIF  IF CR 11 EMIT 79 SPACES 11 EMIT MYSELF ENDIF
11  :    #maxi ! ;
12  :  : paramètres? same @0<> IF
13  :      CR ." Fixons les règles du jeu, voulez-vous..." CR CR
14  :      option? nombre? mini? maxi?
15  :    ENDIF  begin?  ;              -->
```

écran #4

```
 0  :  : mise-à-jour
 1  :    @CURS #prise @ #pile -!
 2  :    7 0 ICURS 80 SPACES
 3  :    7 80 #pile @ - 2/ ICURS #pile @ "! CHARS
 4  :    6 30 ICURS #pile ?  ICURS  ;
 5  :  : page
 6  :    titre ." On gagne en "
 7  :    #option @0<> IF ." évitant" ELSE ." prenant" ENDIF
 8  :    ."  la dernière allumette..." CR
 9  :    ." Minimum à retirer= " #mini ? CR
10  :    ."  --- Maximum à retirer= " #maxi ? CR CR
11  :    ." Nombre d'allumettes restant = " #pile ? CR
12  :    mise-à-jour  ;
13  :
14  :
15  :                          -->
```

écran #5

```
 0 :  : (Xjeu)
 1 :    ." Combien retirez-vous d'allumettes ? " BELL GET/NBRE
 2 :    IF #prise 0! ELSE #prise ! ENDIF
 3 :    #prise @  1 #mini @ [WITHIN]  #prise #pile @= AND  IFNOT
 4 :      #prise @  #mini @  #maxi @  [WITHIN] 0=  #prise @  #pile @ >
 5 :      OR IF  CR 11 EMIT 79 SPACES 11 EMIT MYSELF  ENDIF
 6 :    ENDIF   ;
 7 :  : Xjeu
 8 :    10 0 !CURS  ." A vous de jouer..."  25 SPACES (Xjeu)
 9 :    CR 2 11 CHARS  160 SPACES  2 11 CHARS
10 :    ." Vous venez donc de retirer "  #prise ?
11 :    ." allumettes..."   mise-à-jour   ;
12 :
13 :
14 :
15 :                          -->
```

écran #6

```
 0 :  : (Ojeu)
 1 :    3000 0  DO LOOP   \ on fait semblant de bosser dur...
 2 :    #pile @  #mini @ <= IF #pile #prise @! EXIT ENDIF  #prise 0!
 3 :    #maxi @ 1+  #mini @  DO
 4 :      #pile @ 1 -  #mini #maxi @+  MOD
 5 :      #option @0=  IF  0=  ELSE  1  #mini  [WITHIN]  ENDIF
 6 :      IF 1 #prise !  ENDIF   \ hé hé, c'est une situation gagnante
 7 :    LOOP
 8 :    #prise @0=  IF 9 RND #prise !  ENDIF
 9 :    #prise @  #mini @  #maxi @  [WITHIN]  IFNOT MYSELF ENDIF
10 :    #prise @  #pile @ > IF MYSELF ENDIF   ;
11 :  : Ojeu
12 :    14 0  !CURS  ." A moi donc de jouer..."  (Ojeu)  25 SPACES
13 :    CR  2 11 CHARS  160 SPACES  2 11 CHARS
14 :    ." Je viens donc de retirer "  #prise ?
15 :    ." allumettes..."   mise-à-jour     ;      -->
```

écran #7

```
 0 :  : diagnostic
 1 :    19 0 !CURS
 2 :    #joueur #option @= IF
 3 :      ." Bravo, vous avez gagné !!"
 4 :    ELSE
 5 :      ." ha ha, je vous ai bien eu, avouez-le !!"
 6 :    ENDIF ;
 7 :  : again?
 8 :    21 0 !CURS ." Une autre partie ? " 0/N? ;
 9 :  : same?
10 :    22 0 !CURS ." On garde les mêmes règles ? " 0/N? 0= ;
11 :
12 :
13 :
14 :                          -->
15 :
```

écran #8

```
 0 :  : (jouer)
 1 :    titre paramètres? #nombre #pile @! #prise 0! page
 2 :    #begin #joueur @!
 3 :    BEGIN
 4 :      #joueur #0<> IF Xjeu ELSE Ojeu ENDIF
 5 :      #joueur @0= #joueur ! \ un tour chacun, quoi...
 6 :    #pile @0= UNTIL
 7 :    diagnostic again?
 8 :    IF same? same ! MYSELF ENDIF ;
 9 :  : JOUER
10 :    same 1! (jouer) ;
11 :
12 :  :S  \ That's all folks !!
13 :
14 :
15 :
```

Nous ne saurions trop vous engager de disséquer soigneusement ce petit programme sans prétention, manuel de référence de zFORTH et "glossaire rapide" en mains...

Dans les prochains numéros de ce même bulletin, nous aborderons d'autres aspects du langage FORTH: machine-virtuelle FORTH, structures de contrôles, structures de données, structures des vocabulaires, gestion de la mémoire virtuelle, etc... Lancez-vous dans la programmation en FORTH, et vous redécouvrirez votre bon vieux micral (ou logabax ou sil'z), et en outre, je pense que, comme moi-même, vous vous apercevrez que vous êtes plus efficace, à tous les points de vue (qualitatif et quantitatif).

Pour terminer, voici une bonne nouvelle pour tous les passionnés du langage FORTH dans l'Education Nationale : je vais réaliser une implantation de FORTH sur MO5, dans le cadre du nano-réseau. Cette version sera opérationnelle dès la rentrée scolaire 85. Elle tiendra compte de la mémoire auxiliaire de 64K (organisée en 4 banques de 16K, commutables, et situées à la même adresse que la ROM BASIC). Seront intégrés les outils suivants: éditeur d'écrans-FORTH et éditeur de texte, macro-assembleur 6809, débugger, décompilateur FORTH, gestion d'overlays relogeables, virgule flottante. Les entêtes des mots FORTH seront séparés, ce qui permet de supprimer les entêtes des mots d'un programme terminé (pour ne garder que le code exécutable). En outre, ce FORTH sera en code direct chaîné, c'est-à-dire que le CFA (champ code) contiendra du code machine au lieu d'une adresse. Le fait d'éliminer ainsi une indirection fait gagner un temps appréciable à l'exécution. Je désire que ce FORTH soit un système de développement complet, relativement indépendant du matériel sur lequel il sera implanté. C'est d'ailleurs à partir de ce FORTH que sera écrit le logiciel d'EAO GEOMETRIX, sur MO5. Tout courrier (suggestions, etc...) sera le bienvenu !

Adresse personnelle:

Olivier SINGLA
Ecole Normale d'Instituteurs
Bd. Montauriol
82000 MONTAUBAN

18 av. Mayenne
82000 MONTAUBAN
(63) 66-10-90

HP-75

Paris, le 19 Janvier 1985

Mr VINCENT DELORME (T 350)
5, rue GUSTAVE COURBET
75116 PARIS

Module BR, suite du n° MR4
p 88

Cher Jean-Daniel,

La procédure d'installation du module est la suivante : (la machine est éteinte, et les programmes sauvé

1 - le module doit être introduit de façon à ce que l'étiquette rouge soit visible pendant l'opération

2 - il faut l'enfoncer à fond dans son logement (1 mm environ du support en époxy doit dépasser dans le compartiment batterie).

3 - le module doit être installé horizontalement (le dos du gros circuit intégré (mémoire 8 Koctets) doit se trouver à environ 2 mm au dessus de la plaque en époxy qui constitue le fond du logement du module (et qui forme le clavier du 75). Il est recommandé de placer une cale [en carton, par exemple] pour respecter la condition précédente.

On réinsère la batterie, et on allume la machine :

4 - si l'écran reste blanc, enlever la batterie et retourner à l'étape 1. Si rien ne se produit au bout de quelques essais, le module doit être dépanné

5 - si la routine d'initialisation s'affiche (demande de l'heure et de la date), passer en mode BASIC et taper ou charger la commande MEM, puis presser RTN

le 75 détermine la taille de mémoire dont il dispose par incrémento de 2 Koctets. Environ 2 Koctets sont utilisés par le système donc la commande MEM doit conduire à l'affichage de :

14 XXX s'il n'y a pas de module inséré
22 xxx si le module RAM est inséré
(x est un chiffre).

donc si l'affichage est différent de 22 xxx il faut retourner à l'étape 3 et réajuster le module.

Test du fonctionnement du module :

frapper le petit programme suivant :

faire EDIT "Q", puis :

```
10  ASSIGN # 1 TO "W"
20  PRINT # 1 ;"< chaîne de caractères quelconque>"
30  GOTO 20
```

(ce programme permet de tester toute la mémoire puisqu'il la remplit complètement).

faire RUN. au bout de quelques dizaines de secondes doit s'afficher le message d'erreur :

ERROR line 20 : not enough memory

faire EDIT "W", aller à la fin du fichier et vérifier que les dernières lignes contiennent bien la chaîne de caractère définie à la ligne 20.

Si cette procédure s'effectue comme elle est décrite, alors le module est "bon pour le service".

[En cas de problème, tu peux me joindre au téléphone au 16 (20) 88 05 47 à partir de 19 heures du lundi au jeudi]

AGRANDIR

```
>pl.
10 ! **** AGRANDIR ****
20 !
21 ! Ce programme permet d'agrandir un dessin
22 ! obtenu a partir du programme DESSIN
23 ! cf Micro revue N 2 Pages 113 et 114.
24 !
30 INPUT "Nom du dessin = ";N$
40 INTEGER S
50 ON ERROR OFF ERROR @ DISP "Dessin n'existe pas" @ END
60 OFF ERROR
70 CAT N$ @ ASSIGN # 1 TO N$ @ M$=N$&"B" @ ASSIGN # 2 TO M$ @ READ # 1,I ; N,L
80 FOR I=1 TO L*N
90 READ # 1,I+1 ; X
100 B$=""
110 X1=IP(X/2) @ X2=X-X1*2 @ X2$=STR$(X2) @ B$=X2$&B$
120 IF X1#0 THEN X=X1 @ GOTO 110
130 B=VAL(B$)
140 PRINT # 2,I ; B
150 NEXT I
160 ASSIGN # 1 TO *
170 INPUT "Agrandissement = ";X
180 ASSIGN # 3 TO N$&STR$(X) @ PRINT # 3,1 ; X*N,L
190 FOR I=1 TO N
200 DIM L2$[240]
210 FOR J=1 TO L
220 READ # 2,(I-1)*L+J ; L1 @ L1$=STR$(L1) @ M=8-LEN(L1$)
230 FOR K=1 TO M @ L1$="0"&L1$ @ NEXT K
240 FOR K=0 TO 7
250 L2$[I+K*X,I+K*X]=L1$[K+1,K+1]
260 NEXT K
270 FOR K=0 TO 7
280 FOR M=1 TO X-1
290 L2$[I+K*X+M,I+K*X+M]=L2$[I+K*X,I+K*X]
300 NEXT M
310 NEXT K
320 FOR K=1 TO X
330 S=0 @ Y=VAL(L2$[8*K-7,8*K])
340 FOR M=1 TO 8
350 Y=Y/10 @ F=FP(Y)*10*2^(M-1) @ S=S+F @ Y=IP(Y)
360 IF Y=0 THEN 380
370 NEXT M
380 PRINT # 3,(I-1)*L*X+L*(K-1)+J+1 ; S
390 NEXT K
400 NEXT J
410 NEXT I
420 PURGE M$
430 DIM A$[200]
440 PWIDTH INF
450 PRINT CHR$(27)&"&l9D"
460 FOR I=1 TO X*N
470 A$=""
480 FOR J=1 TO L
490 READ # 3,(I-1)*L+J+1 ; Y
500 Y$=CHR$(Y)
510 Y1$=""
520 FOR K=1 TO X @ Y1$=Y1$&Y$ @ NEXT K
530 A$=A$&Y1$
540 NEXT J
550 PRINT TAB(40);CHR$(27)&"*b"&STR$(X*L)&"G"&A$
560 NEXT I
570 ASSIGN # 3 TO *
580 PRINT CHR$(27)&"&l6D"
```

```
>
Nom du dessin = ferme
FERME        B  784 13:26 06/26/84
Agrandissement = 1
```

```
>
Nom du dessin = ferme
FERME        B  784 13:26 06/26/84
Agrandissement = 2
```

```
>
Nom du dessin = ferme
FERME        B  784 13:26 06/26/84
Agrandissement = 3
```

M. Daulus
8 imp. Bastian
88000 Epinal

Jean Claude LIEHN (T488)
8 rue de l'écaille
51100 REIMS
Tel (26) 85 50 52

Cher président,

Voici ma contribution à MICRO-REVUE : il s'agit d'un programme qui me parait présenter un double intérêt.

D'une part il propose une solution assez efficace au problème classique de l'ajustement des courbes. Si le cas des modèles linéarisables est bien traité dans l'ouvrage de W.M. KOLB (Editions du CAGIRE), le cas des modèles non linéaires nécessite l'utilisation de méthodes itératives. Le programme proposé permet de travailler avec des modèles quelconques. J'espère qu'il consolera les utilisateurs du HP75 de ne pas avoir droit au module CURVE FITTING du 71.

Le second intérêt de ce programme est qu'il illustre la puissance du module MATH . Malgré les commentaires détaillés, le programme n'occupe que 3K. Le programme est la traduction lettre à lettre des formules mathématiques décrivant la méthode utilisée. Quel autre matériel permet d'en faire autant ?

Les abondants commentaires et l'exemple me semblent suffisants pour utiliser le programme sans notice supplémentaire.

Bonne programmation (non linéaire !).

J.C. LIEHN

PS : La cassette ci-jointe contient le programme CRFIT, le fichier de données DATAFILE utilisé comme exemple (somme de deux exponentielles), le programme LI qui édite le fichier courant assez proprement. (Ceci au cas où vous modifieriez mon programme pour sa publication, LI n'a rien à voir avec CRFIT, il nécessite le module I/O) et enfin cette LETTRE avec les commandes du TEXT FORMATTER.

```
100   ################################################################################
      # ! CRFIT : Ajustement des paramètres d'un modèle à une courbe expérimentale.  #
      ################################################################################
110 REM
120 REM   Matériel = HP75 + module MATH.
130 REM   Données dans le fichier 'DATAFILE' sous la forme : #ligne DATA Xi,Yi
140 REM   Le nombre de paramètres (P) du modèle est écrit à la ligne 390.
150 REM   Le modèle est écrit à la ligne 400.
160 REM   Pour changer de modèle, il faut changer ces deux lignes.
170 REM   La méthode de MARQARD est utilisée.
180 REM   Le modèle peut être linéaire ou non (exemple non linéaire = somme d'exponentielles)
190 REM   Auteur : J.C. LIEHN (T488) 8 Rue de l'écaille 51100 REIMS Tel (26) 85 50 52
200 REM
210 OPTION BASE 1 @ DELAY 0
220 REAL X(30),Y(30) !   Données expérimentales.
230 REAL Y0(30) !        Valeurs du modèle.
240 REAL Y1(30) !        Ecart données-modèle.
250 REAL J0(8,30) !      Jacobien.
260 REAL J1(30,8) !      Transposé du Jacobien.
270 REAL H1(8,8) !       Matrice de curvature.
280 REAL H2(8,8) !       Matrice de curvature modifiée.
290 REAL B(8) !          Matrice Béta.
300 REAL A0(8) !         Valeurs initiales des paramètres.
310 REAL A(8) !          Valeurs courantes des paramètres.
320 REAL S(8) !          Incrément des paramètres.
330 INTEGER N !          Nombre de points expérimentaux.
340 INTEGER P !          Nombre de paramètres du modèle.
350 INTEGER I,J,T
360 REAL F1,F2,F,D,K0,K1,K
370 REM
```

```
380     ################################################################################
        # ! Définition du modèle.                                                      #
        ################################################################################
390  P=4
400  DEF FNY(X) = A(1)*EXP(-A(2)*X)+A(3)*EXP(-A(4)*X)
410     ################################################################################
        # ! Calcul du Jacobien (dérivée du modèle par rapport aux paramètres.)         #
        ################################################################################
420  DEF FND(X,J)
430  D=MAX(A(J)*.000001,.000001) @ A(J)=A(J)+D @ F1=FNY(X) @ A(J)=A(J)-2*D @ F2=FNY(X)
440  FND=(F1-F2)/(2*D) @ A(J)=A(J)+D @ END DEF
450     ################################################################################
        # ! Cacul du Khi 2 (Distance modèle-données)                                   #
        ################################################################################
460  DEF FNK2
470  FOR I=1 TO N @ Y0(I)=FNY(X(I)) @ NEXT I
480  MAT Y1=Y-Y0 @ FNK2=DOT(Y1,Y1)/(N-P) @ END DEF
490  REM
500  GOTO 770 ! saut au programme principal.
510  REM
520     ################################################################################
        # ! Construction des matrices.                                                 #
        ################################################################################
530  FOR I=1 TO N @ FOR J=1 TO P @ J0(J,I)=FND(X(I),J) @ NEXT J @ NEXT I
540  MAT J1=TRN(J0) @ MAT H1=J0*J1 @ MAT B=J0*Y1 @ RETURN
550     ################################################################################
        # ! Incrémentation des valeurs des paramètres.                                 #
        ################################################################################
560  MAT H2=H1 @ FOR I=1 TO P @ H2(I,I)=(1+F)*H2(I,I) @ NEXT I
570  MAT H2=INV(H2) @ MAT S=H2*B @ MAT A=A0+S @ K=FNK2 @ RETURN
580     ################################################################################
        # ! Lecture des données.                                                       #
        ################################################################################
590  ASSIGN # 1 TO "DATAFILE"
600  N=0 @ ON ERROR GOTO 620
610  N=N+1 @ READ # 1 ; X(N),Y(N) @ GOTO 610
620  OFF ERROR @ ASSIGN # 1 TO * @ N=N-1 @ RETURN
630     ################################################################################
        # ! Initialisation des paramètres.                                             #
        ################################################################################
640  REDIM X(N),Y(N),Y0(N),J0(P,N),H1(P,P),A0(P)
650  DISP "*** CRFIT ***" @ DISP "ESTIMATIONS INITIALES : " @ MAT INPUT A0 @ F=.001 @ RETURN
660     ################################################################################
        # ! Edition des résultats.                                                     #
        ################################################################################
670  BEEP 3000,.2 @ DELAY 100
680  DISP "*** RESULTATS ***"
690  FOR I=1 TO P
700       DISP USING "'A(',DD,') = ',D.DDDE" ; I,A0(I)
710  NEXT I
720  DISP
730  FOR I=1 TO N
740       DISP USING "'#',DD,' Ye=',D.DDDE,' Ym=',D.DDDE" ; I,Y(I),FNY(X(I))
750  NEXT I
760  GOTO 670
770     ################################################################################
        # ! Programme principal.                                                       #
        ################################################################################
780  GOSUB 580 ! Lecture des données.
790  GOSUB 630 ! Initialisation.
800  MAT A=A0 @ K0=FNK2 @ T=0 @ K1=K0
810  DISP USING "'Khi Deux = ',D.DDDDE" ; K0
820  GOSUB 520 ! Calcul de la matrice béta.
830  GOSUB 550 ! Incrémentation.
840  IF K>=K0 AND T=0 THEN F=F*10 @ GOTO 830
850  IF K>=K0 AND T=1 THEN 870
860  IF K<K0 THEN MAT A0=A @ K0=K @ F=F/10 @ T=1 @ GOTO 870
870  IF (K1-K)/K1>.01 THEN 800
880  MAT A=A0 @ GOTO 660 ! Edition des résultats.
```

```
>LIST"DATAFILE"
1 DATA 0,100
2 DATA 5,67.5
3 DATA 10,47.5
4 DATA 15,35.1
5 DATA 20,27.2
6 DATA 25,22.1
7 DATA 30,18.8
8 DATA 35,16.5
9 DATA 40,14.9
10 DATA 45,13.6
```

```
>RUN"CRFIT"
*** CRFIT ***
ESTIMATIONS INITIALES :
A0(1)? 50
A0(2)? .2
A0(3)? 50
A0(4)? .005
Khi Deux = 6.1878E+002
Khi Deux = 3.3332E+002
Khi Deux = 1.2858E+002
Khi Deux = 1.0985E+001
Khi Deux = 3.0821E+000
Khi Deux = 2.0606E-002
Khi Deux = 1.1030E-003
*** RESULTATS ***
A( 1) = 7.978E+001
A( 2) = 1.003E-001
A( 3) = 2.022E+001
A( 4) = 1.025E-002

# 1 Ye=1.000E+002 Ym=9.999E+001
# 2 Ye=6.750E+001 Ym=6.752E+001
# 3 Ye=4.750E+001 Ym=4.751E+001
# 4 Ye=3.510E+001 Ym=3.505E+001
# 5 Ye=2.720E+001 Ym=2.720E+001
# 6 Ye=2.210E+001 Ym=2.214E+001
# 7 Ye=1.880E+001 Ym=1.880E+001
# 8 Ye=1.650E+001 Ym=1.650E+001
# 9 Ye=1.490E+001 Ym=1.486E+001
#10 Ye=1.360E+001 Ym=1.362E+001
```

DUMP

dump

DUMP B 806 20:35 15\04\85

```
5     ##########################################################################
      # ! DUMP POUR ADVANCED PAC SCREEN ET HP82905B                            #
      ##########################################################################
10  PRINTER IS ":P1"
20  PRINT CHR$(27)&"&k2S"
30  PRINT CHR$(27)&"&l9D"
40  DIM A$[90],B$[90],C$[640]
50  INTEGER C(640),I,J,K,Z,N,T
60  N=0 @ T=0
70  PRINTER IS ":G1"
80  PRINT "DL"
90  ENDLINE CHR$(10)
100  ENTER ":G1" ; A$
110  IF LEN(A$)<86 THEN ENTER ":G1" ; B$ @ A$=A$&CHR$(10)&B$ @ GOTO 110
120  N=N+1
130  GOSUB 150
140  IF SPOLL(":G1")#34 THEN T=1 @ GOSUB 150 @ ENDLINE @ PRINTER IS ":P1" @ END
145  GOTO 100
150  Z=2^(8-MOD(N,8)) @ IF Z=256 THEN Z=1
160  IF MOD(N,8)=1 THEN MAT C=ZER
170  FOR I=7 TO 86
180        K=(I-7)#8
190        D$=A$[I,I]
200        IF D$="X" THEN GOTO 240
210        FOR J=7 TO 0 STEP -1
220              IF FLAG?(D$,J)=1 THEN C(K+8-J)=C(K+8-J)+Z
230        NEXT J
240  NEXT I
250  IF T=1 THEN N=8
260  IF MOD(N,8)=0 THEN C$="" @ FOR I=1 TO 640 @ C$=C$&CHR$(C(I)) @ NEXT I
270  IF MOD(N,8)=0 THEN OUTPUT ":P1" ; CHR$(27)&"*b640G"&C$&CHR$(13)
280  RETURN
```

```
5 INTEGER P,C,R,F5,V5,V0,F1,X0,Y0,X1,Y1,Z0,Z1,Z2,S1,S1,S5,S6,S7
10 RCLVAR
100 DIM D$[96],BO$[96],T$[74],T1$[74],D1$[96],FB$[41]
200 M2,M0=0 @ M1=1 @ M2=2 @ M3=3 @ M4=4 @ M5=5
210 T1$="[+-(.0123456789*ABCDEFGHIJKLMNOPQRSTUVWXYZ @%%*/!$)`AAOOEEII"
230 T1$[17,173]=""" @ T2$=CHR$(13)&"ÂÊÎ Ô`Û&"
700 IF F5=M4 THEN F5=M2 @ P1,M=M0 @ T$=UPRC$(CHOICE$(12)) @ GOSUB 8810 @ GOTO 2930
702 IF V0 AND P<M0 THEN GOSUB 9680
810 T$=T1$ @ M$="" @ IF V0 AND M0! V5 THEN CURSOFF
811 PUTSTATUS M5,C @ PUTSTATUS 6,R @ IF M0! Z0 THEN Z1=C @ Z2=R
812 W0=M1
814 MARGIN 91 @ GOSUB 9860
820 F1=CELLTYPE(C,R) @ F5=M1 @ IF F1>M3 THEN D$=GETLABEL$(C,R) @ GOTO 900
827 IF S1&(F1&M1) THEN D$=GETFORMULA$(C,R,S4) ELSE D$=GETVALUE$(C,R,S6)
900 IF M0! V0 THEN 950
910 IF M0! V5 THEN GOSUB 9415
920 IF V5=M1 THEN GOSUB 9500 @ GOSUB 9610 @ GOSUB 9415
930 IF V5=M3 THEN CURSON
950 GOSUB 8950 @ M,U1=M0 @ D1=M1 @ V5=M2
955 GOSUB 8705
960 F5=M1 @ GOSUB 8810
965 IF P>66 THEN ON P-66 GOSUB 1510,1520,1530,1540,1550,1560,1570,1580 ELSE 985
970 IF F5 THEN GOSUB 9710 @ GOSUB 9720 ELSE 960
975 IF NOT F5 THEN GOSUB 9990 @ GOTO 960 ELSE V5=M0 @ GOTO 810
985 IF P>62 THEN ON P-62 GOSUB 1810,1820,1830,1840 @ GOTO 955
990 IF P=62 THEN GOSUB 9800 @ GOTO 812
995 IF P=61 THEN GOSUB 9850 @ GOTO 814
1000 F5=M2 @ IF P>50 THEN ON P-50 GOTO 2910,9000,8500,7510,7010,1210,8100,8000,7310,7400
1010 M=M3+(P>16) @ GOTO 2100
1210 BO$="" @ M$=MESSAGE$(20)
1220 GOSUB 9115 @ R9=ROW(BO$) @ C9=COL(BO$) @ G=LEGAL(BO$)
1230 IF G=M2 AND NOT C9=R9 OR NOT (C9=R9) THEN BEEP @ GOTO 1220
1240 IF R9 THEN R=R9
1250 IF C9 THEN C=C9
1272 V5=M0 @ GOTO 810
1510 R=R-M1 @ RETURN
1520 R=R+M1 @ RETURN
1530 C=C-M1 @ RETURN
1540 C=C+M1 @ RETURN
1550 IF R=M1 THEN F5=M0 @ RETURN ELSE R=M1 @ RETURN
1560 IF R=MAXROW THEN F5=M0 @ RETURN ELSE R=MAXROW @ RETURN
1570 IF C=M1 THEN F5=M0 @ RETURN ELSE C=M1 @ RETURN
1580 IF C=MAXCOL THEN F5=M0 @ RETURN ELSE C=MAXCOL @ RETURN
1810 U1=-M1 @ RETURN
1820 U1=M1 @ RETURN
1830 U1=-WINSIZ @ RETURN
1840 U1=WINSIZ @ RETURN
2099 M=M5
2100 BO$=K$ @ P0=M2 @ W0=M1 @ F0=M0
2105 IF POS(""&"",K$) OR M=M5 THEN BO$="" @ P0=M1
2106 IF NOT LEN(MESSAGE$(M4)) THEN M=M4
2110 PUTSTATUS 12,M2 @ GOSUB 9860 @ GOSUB 9870
2115 F5=-M3 @ IF M=M4 THEN MARGIN GETWIDTH(C)
2116 T$=T2$ @ IF M=M3 THEN T$[M5,M5]="!"
2130 GOSUB 9117 @ IF P&M4 THEN 2145
2133 GOSUB 9850 @ GOSUB 9860 @ GOSUB 9870
2137 W3=LEN(BO$)+LEN(M$)-2 @ IF W0>W3 AND W2<M1 THEN W0=W3
2139 GOTO 2130
2145 IF P=M5 THEN 2182
2155 IF NOT LEN(BO$) THEN BLANK C,R @ GOTO 2165
2160 IF M=M3 OR POS(BO$[M1,M1],"@")=(M=M4) THEN 2182
2164 IF M=M4 THEN PUTLABEL C,R,BO$ ELSE PUTFORMULA C,R,BO$
2165 IF F1<M4 THEN F5=-M1
2167 GOTO 2193
2182 IF LEN(BO$)>91 THEN P0=92 @ GOTO 2188
2183 P0=PARSE(BO$) @ IF NOT P0 THEN 2190
2184 IF W0>P0 THEN W0=P0
2188 F0=M0 @ BEEP @ GOTO 2130
2190 IF P=M5 THEN F5=-M2 @ BO$=INTERP$ @ W0=M1 @ F0=M0 @ P0=LEN(BO$)+M1 @ GOTO 2130
2192 PUTFORMULA C,R @ F5=-M1
2193 V5=M3 @ IF CELLTYPE(M0,M0)=6 THEN 2196 ELSE S2=GETSTATUS(12) @ PUTSTATUS 12,M2
2194 IF V0 THEN CURSOFF
2195 RECALC M0,M0 @ IF GETSTATUS(12)>M2 THEN PUTSTATUS 12,S2
2196 U0=GETSTATUS(18)=LEN(DSP$)>M0 @ IF GETSTATUS(12)>M2 THEN 9020
2199 IF F5<(NOT GETSTATUS(K1))=-M1 THEN GOSUB 9982 @ GOTO 9000 ELSE 810
2910 M1=21 @ GOSUB 8795
2930 IF P>11 THEN P=P+M4 @ GOTO 9030
2940 ON P GOTO 9030,4310,9030,4910,5310,9030,9030,6310,6210,9030,2099
4310 M1=28 @ GOSUB 8795 @ PUTFORMAT C,R,P-M1 @ V5=M3 @ GOTO 810
4910 M1=38 @ GOSUB 8795 @ ON P GOTO 5000,5020,4940,5200
4940 M$=MESSAGE$(42) @ BO$=COLWIDTH$(M0)
4955 GOSUB 9115 @ IF NOT LEN(BO$) THEN GOSUB 9980 @ PUTWIDTH M0 @ GOTO 5350
4980 IF VALCHK(BO$)<M0 THEN 4955 ELSE GOSUB 9980 @ PUTWIDTH M0,VAL(BO$) @ GOTO 5350
5000 T3$="II" @ M=40 @ X=M1 @ Y=M0 @ GOTO 5050
5020 T3$="ÎÊ" @ M=41 @ X=M0 @ Y=M1
5050 F7=M0 @ C9=C @ R9=R @ SETWIN @ M1$=MESSAGE$(M)&" " @ L1=LEN(M1$)+M1 @ DISP M1$;
5052 T$=T3$&T1$[M1,5036]""
5060 SETWIN L1 @ CLRLCD L1 @ M2$=COORD$(X*C,Y*R,M3) @ IF NOT POS(M2$,"[") THEN M2$=""
5064 DISP CHR$(22);"Z@";COORD$(X*C,Y*R,M0);": ";M2$;
5065 F5=-6 @ GOSUB 8810 @ IF P>M4 THEN 5090
5070 IF M=41 THEN ON P GOSUB 1510,1520,1550,1560 ELSE ON P GOSUB 1530,1540,1570,1580
5075 IF F5 THEN GOSUB 9710 @ GOSUB 9720 ELSE 5065
5080 IF NOT F5 THEN GOSUB 9990 @ GOTO 5065 ELSE 5060
5090 IF P=21 OR P=22 OR P=55 AND LEN(M2$) THEN BO$="" @ P0=M1 @ GOTO 5140
5095 IF P=56 AND LEN(M2$) THEN BO$=GETLABEL$(X*C,Y*R) @ P0=M1 @ GOTO 5140
5099 IF P>54 THEN GOTO 5065
5135 BO$=K$ @ P0=M2
5140 W0=M1 @ F0=M0 @ M$=M1&COORD$(X*C,Y*R,M0) @ MARGIN 8
5150 GOSUB 9115 @ F7,F5=6 @ IF P=M2 THEN 5052
5162 IF NOT LEN(BO$) THEN BLANK X*C,Y*R @ GOTO 5052
5163 IF LEN(BO$)>8 THEN D1$=BO$[M1,8] ELSE D1$=BO$
5165 IF D1$="[]" THEN BO$="" @ GOTO 5190 ELSE D1$="["&D1$&"]"
5167 IF LEGAL(D1$)=M1 THEN 5175
5170 OFF ERROR @ IF ERRN=16 THEN 5190 ELSE BEEP @ GOTO 5150
5175 ON ERROR GOTO 5170
5190 PUTLABEL X*C,Y*R,BO$ @ OFF ERROR @ GOTO 5052
5200 M1=43 @ M2=23 @ GOSUB 8795 @ GOSUB 9980 @ PUTSTATUS 13,N2-P @ GOTO 5350
5310 M1=44 @ M2=25 @ GOSUB 8795
5340 IF P=M1 AND MAXCOL<255 AND COLWIDTH$(255)="" THEN GOSUB 9982 @ INSCOL C @ GOTO 5350
5345 IF P=M2 AND MAXROW<255 THEN GOSUB 9982 @ INSROW R @ GOTO 5350
5348 BEEP @ GOTO 5310
5350 V5=M1 @ GOTO 810
6210 IF NOT LEN(DSP$) THEN BEEP @ GOSUB 8810 @ GOTO 2930
6211 M1=61 @ M2=23 @ GOSUB 8795 @ W0=NOT (P-M1) @ PUTSTATUS 18,W0
6230 IF W0 THEN GOSUB 9982 @ CLRSCR @ GOSUB 9680 ELSE CURSOFF
6250 GOTO 810
6310 M$=MESSAGE$(56) @ BO$=COORD$(C,R,S5) @ W0=M1 @ F0=M0 @ P0=LEN(BO$)+M1 @ F5=-M4
6320 GOSUB 9115 @ GOSUB 6600 @ SOURCE C8,R8,C9,R9 @ C6=C8 @ R6=R8 @ C7=C9 @ R7=R9
6350 BO$="" @ F5=-M5 @ P0,W0=M1 @ F0=M0
6360 M$=MESSAGE$(57) @ GOSUB 9115 @ IF P&6 THEN 6360
6363 T$="" @ M$="" @ U9=W0 @ P9=P0
6370 M$=MESSAGE$(56) @ D$=COORD$(C6,R6,S5) @ D1$=COORD$(C7,R7,S5)
6371 IF D$&D1$ THEN D$=D$&"..."&D1$
6373 W0=M1 @ GOSUB 6680 @ ON P-M4 GOTO 810,6374,6376
6374 W0=U9 @ P0=P9 @ GOTO 6360
6376 GOSUB 9850 @ GOTO 6370
6380 GOSUB 6600 @ TARGET C8,R8,C9,R9
6390 FOR C3=C6 TO C7 @ FOR R3=R6 TO R7
6405 IF LEN(FIRSTREF$(C3,R3,S4)) THEN C6=C3 @ R6=R3 @ C3,R3=INF
6410 NEXT R3 @ NEXT C3 @ IF C3=INF THEN 6430
6417 GOSUB 9982
6420 V5=M1 @ REPLICATE @ GOTO 9012
6430 M1=58 @ GOSUB 8795 @ P8=P @ IF P8=M1 THEN 6417
6435 IF P8=M2 THEN GOSUB 9982
6445 FOR C4=C6 TO C7 @ FOR R4=R6 TO R7 @ P9,L5=M1 @ G=M0
6450 IF P8=M2 THEN L5=M0 @ GOTO 6460
6457 D1$=GETFORMULA$(C4,R4,S4=L5) @ IF LEN(D1$)<92 THEN 6460
6458 IF NOT L5 THEN 6580 ELSE L5=M0 @ GOTO 6457
6460 B0$=FIRSTREF$(C4,R4,S4=L5)
6465 IF NOT LEN(B0$) THEN 6580
6470 IF P8=M2 THEN 6565
6480 D$=D1$ @ IF LEN(B0$)>M1 THEN B0$=F8$ @ GOTO 6520
```

REBUS - suite

```
6515 GOSUB 6705 @ IF P7 THEN FB0=BO$[P7+N3] @ BO$=BO$[M1,P7-M1]
6520 WO=POS(D$[P3],BO$)+P9-M1 @ P9=WO+M1 @ PO=WO
6527 D$[WO,LEN(BO$)+WO-M1]=HIGH$(D$[WO,LEN(BO$)+WO-M1])
6528 M$=COORD$(C4,R4,S5=S7) @ T9$=UPRC$(CHOICE$(KO))&"A "
6529 M$=MESSAGE$(60) @ GOSUB 6680 @ ON P-M4 GOTO 6575,6570,6565,6530,6534,6575
6530 IF G OR NOT L5 THEN 6529 ELSE GOSUB 9800 @ P9=M1 @ GOTO 6457
6534 GOSUB 9850 @ GOTO 6528
6565 MARK
6570 BO$=NEXTREF$ @ G=M1 @ GOTO 6465
6575 UNMARK @ C4,R4=INF
6580 NEXT R4 @ NEXT C4
6585 IF C4@INF THEN 6590
6587 IF P=M5 THEN 810 ELSE 9005
6590 IF P@=M2 THEN 6420 ELSE 6417
6600 GOSUB 6700 @ IF NOT P7 THEN C8$=BO$ @ C9$=BO$ @ GOTO 6627
6610 C8$=BO$[M1,P7-M1] @ C9$=BO$[P7+N3]
6627 C8=COL(C8$) @ R8=ROW(C8$) @ IF NOT C8 OR NOT R8 THEN PO=M1 @ GOTO 6750
6633 C9=COL(C9$) @ R9=ROW(C9$) @ IF NOT C9 OR NOT R9 THEN PO=P7+N3 @ GOTO 6750
6640 IF R8@R9 AND C8@C9 OR R8>R9 OR C8>C9 THEN WO,PO=M1 @ GOTO 6750
6648 RETURN
6680 I$=T1$[63,63]&"%T9$ @ U1=NO @ GOSUB 8950
6685 GOSUB 8705 @ GOSUB 8810 @ IF P<M5 THEN ON P GOSUB 1810,1820,1830,1840 @ GOTO 6685
6697 RETURN
6700 IF LEN(BO$)>41 THEN WO,PO=M1 @ POP @ GOTO 6750
6705 P7=POS(BO$,"...") @ IF NOT P7 THEN RETURN
6715 IF LEN(BO$)>P7+N2 THEN POP ELSE RETURN
6750 BEEP @ POP @ FO=NO @ IF PO<WO THEN WO=PO
6760 IF M$=MESSAGE$(56) THEN 6325 ELSE 6360
7010 IF F1>M5 THEN M$=" " @ GOTO 7050
7018 IF F1<M4 THEN M=17 ELSE M=F1+14
7020 M$=" "&MESSAGE$(M)&" "
7050 H$=COORD$(C,R,NO) @ D$=MESSAGE$(GETFORMAT(C,R)+6) @ M$=M$&CTIVE$ @ FB$=STR$(MEM)
7055 C8$=MESSAGE$(GETSTATUS(N3)+6) @ C9$=MESSAGE$(GETSTATUS(N2)+13)
7057 D1$=MESSAGE$(GETSTATUS(M1)+15) @ SETWIN @ CLRLCD
7060 DISP H$;" ";D$;T$;" ";C8$;" ";C9$;" ";D1$;" ";F8$ @ T$="E" @ GOSUB 8810 @ GOTO 81
7310 PUTSTATUS 11,NOT S1 @ S1=GETSTATUS(11) @ GOTO 820
7400 PUTSTATUS 16,NOT S6 @ S6=GETSTATUS(16) @ GOTO 820
7510 IF F1>M5 THEN 960
7520 IF F1<M4 THEN BO$=GETFORMULA$(C,R,S4) @ M=N3 ELSE BO$=GETLABEL$(C,R) @ M=F1
7523 PO=WO @ FO=NO @ GOTO 2110
8000 PUTSTATUS 9,C @ PUTSTATUS 10,R @ GOTO 810
8100 ZO=NOT ZO @ IF ZO THEN C=GETSTATUS(9) @ R=GETSTATUS(10) ELSE C=Z1 @ R=Z2
8110 GOTO 1272
8500 BO$="" @ WO,PO=M1 @ FO=NO @ IF F1>M5 THEN 960
8510 IF F1<M4 THEN M=N3 ELSE M=F1
8520 GOTO 2110
8705 IF L<WINSIZ THEN U1=NO @ P1=M1 @ P2=L @ GOTO 8730
8707 IF WO+WINSIZ-M1>L THEN D1=M1 ELSE D1=-M1
8709 IF U1>NO THEN D1=M1
8710 IF U1<NO THEN D1=-M1
8715 P1=WO+U1 @ P2=P1+WINSIZ-M1 @ IF P1<M1 OR P2>L THEN U1=U1-D1 @ GOTO 8715
8730 WO=P1 @ DISP CHR$(27);"Z@@";D$[P1,P2]; @ RETURN
8795 IF NOT M2 THEN M2=M1+M1
8796 SETWIN @ DISP MESSAGE$(M1);": ";MESSAGE$(M2)
8800 T$=UPRC$(CHOICE$(M2)) @ M2=NO
8810 K$=WAITKEY$(-M1) @ P=POS(T$,UPRC$(K$)) @ IF P AND K$<>"" THEN RETURN
8830 IF POS(" ",K$) THEN POP ELSE 8810
8835 IF K$=" " THEN 9005
8840 IF P1<F5>N1 THEN 960
8845 IF F5=M1 THEN 812
8850 IF F5=-6 THEN C=C9 @ R=R9 ELSE 810
8851 IF F7 THEN U5=M1 @ GOSUB 9980
8852 GOTO 810
8950 L=LEN(D$) @ L1=LEN(H$) @ SETWIN @ DISP H$;
8955 IF (NOT S1)<>F5=M1 AND (F1=N2 OR F1=N3) THEN DISP "D";ELSE DISP ":";
8956 DISP " "; @ IF NOT LEN(M$) THEN SETWIN L1+N3 @ CLRLCD L1+N3 @ RETURN
8958 R6=32-LEN(M$)-N1 @ DISP CHR$(27);"X";CHR$(R6);"@";M$; @ SETWIN L1+N3,R6-N1
8961 CLRLCD L1+N3,R6-N1 @ M$="" @ RETURN
9000 P=12 @ GOTO 9030
9005 P=13 @ GOTO 9030
9012 P=14 @ GOTO 9030
9020 P=15
9030 STOVAR @ END
9115 T$=T2$
9117 SETWIN @ DISP M$;": "; @ CLRLCD LEN(M$)+N3
9121 IF F5>-N1 THEN WO,PO=M1 @ FO=NO
9123 SETWIN LEN(M$)+N3
9125 K$=INLINE$(BO$,WO,PO,T$,FO)
9130 BO$=INPBUF$ @ WO=INPWIN @ PO=INPPTR @ FO=INFLAG @ L=LEN(BO$)+M1
9180 P=POS(T$,K$) @ IF P>10 THEN 9122
9235 IF P=M1 OR P=6 AND F5=-M5 OR P=M5 AND M=N3 OR P=N4 AND (M>N2 AND M<6) THEN RETURN
9380 IF P=10 THEN 9005
9381 IF P@N2 THEN 9123
9382 IF F5=-N2 THEN POP @ GOTO 2196
9385 IF ABS(F5)=6 THEN RETURN ELSE POP @ GOTO 810
9415 R=-M1
9416 IF C<X0 THEN X0=C @ GOSUB 9510 @ GOSUB 9500
9420 IF C>X1 THEN X1=C @ GOSUB 9500
9425 IF R<Y0 THEN Y0=R @ GOSUB 9610
9430 IF R>Y1 THEN Y0=Y0+R-Y1 @ GOSUB 9610
9455 IF R=-M1 AND U5$N1 THEN 9465
9460 IF NOT U5 THEN GOSUB 9983
9462 GOSUB 9650
9465 CURSON @ RETURN
9500 X0=UPCOL(X0,-M1,-M1)
9510 X1=UPCOL(X0,M1,-M1) @ R=M1 @ RETURN
9610 R=N2 @ Y1=UPROW(Y0) @ RETURN
9650 G=DUMP(X0,Y0,X1,Y1,-M1)
9659 PUTSTATUS 7,X0 @ PUTSTATUS 8,Y0 @ RETURN
9680 U5=N2 @ GOSUB 9510 @ GOSUB 9500 @ GOSUB 9610 @ GOSUB 9416 @ RETURN
9710 IF NOT C OR C>256 THEN F5=NO @ C=C-SGN(C-M1)
9715 RETURN
9720 IF NOT R OR R>256 THEN F5=NO @ R=R-SGN(R-M1)
9725 RETURN
9800 PUTSTATUS 14,NOT S4 @ S4=GETSTATUS(14) @ RETURN
9850 PUTSTATUS 15,NOT S5 @ S5=GETSTATUS(15) @ RETURN
9860 H$=COORD$(C,R,S5=S7) @ IF Z0 THEN H$=HIGH$(H$)
9865 RETURN
9870 M$=H$&MESSAGE$(M) @ RETURN
9980 IF NOT U0 THEN RETURN
9982 IF U0 THEN CURSOFF
9983 SETWIN @ DISP MESSAGE$(M2) @ RETURN
9990 BEEP 110,.05 @ RETURN
```

Fin du REBUS

LA POESIE AU METRE

sur HP-75 par Jean Dutertre, T416.

Un ordinateur n'est pas fait uniquement pour élaborer des calculs scientifiques, encore qu'il y excelle, ni se consacrer à la gestion des fichiers des clients, ce qu'il fait avec la plus scrupuleuse et la plus obstinée des stupidités, la plus efficace aussi. Aux confins évasifs de la combinatoire et de la syntaxe, un peu dans l'esprit des jeux, existe le jeu à la fois bien réglé et soumis au miraculeux hasard de la poésie. Il y a quelques années j'ai pensé que l'on pouvait informatiser aisément les **Cent mille milliards de poèmes** de Raymond Queneau. Ce recueil, paru en 1961 chez Gallimard, était hélas épuisé, et j'en avais offert mon unique exemplaire à une personne dont les beaux yeux, à l'époque, me paraissaient indispensables. Le principe était le suivant: 10 sonnets présentent la même syntaxe, les mêmes rimes, jamais les mêmes mots. On peut à volonter échanger les vers, de sorte que l'on peut ainsi composer 10 à la puissance 14, soit cent mille milliards de poèmes. Il ne me restait comme seule solution de réinventer, sinon des sonnets, et sinon dix, du moins sept fois huit vers d'aimable fantaisie. Il y aura donc 7 à la puissance 8, soit à peu près quarante mille douzaines de douzaines de poèmes "presque" différents. C'est beaucoup moins, je le sais, que l'on me pardonne il s'agit d'un essai.

Le programme QUARANTE permet ainsi d'exploiter le fichier "perdreau" qui contient les vers. La sortie se fait sur console (un moniteur et son interface), sur imprimante thermique, sur Olympia, ou sur le traceur de la Sicape.

Le recueil de Queneau venant d'être reédité, le programme CENTMIL vous permet d'exploiter un fichier "Queneau", que le copyright interdit de fournir: recopiez les vers, sonnet après sonnet, dans un fichier TEXT. Ce sera long, je sais je l'ai fait, mais on en a pour un million de siècles de lecture, "plus quelques plombes et broquilles", comme dit Queneau. C'est rentable.

On peut se demander s'il n'existerait pas un autre procédé que celui qui consiste à interchanger des vers entiers.

Un des textes les plus curieux d'Edgar Poe est peut-être justement cet "essai" par lequel il donne les moyens pratiques de composer un chef-d'oeuvre de la poésie: le choix du sujet, le choix des rimes, le choix d'un refrain, les termes à reprendre en écho, la sonorité... On atteint au sublime, dans le domaine de l'humour glacé, lorsqu'il prend comme exemple de cette construction froide et mécanique son **Corbeau**, le sommet de la poésie américaine et sans doute un des chefs-d'oeuvres de la littérature de tous les temps. A l'en croire, il suffira à tout un chacun de s'appliquer avec soin, en suivant son mode de construction pas à pas, pour y parvenir. C'est le "kit" en pièces détachées du génie littéraire.

Il y a des poètes dont les ficelles paraissent plus à portée de main. Qui ne connait la très belle **Mélancolie** de Léo Ferré:

C'est un chien de plus
Dans le monde qu'on sait
La mélancolie

Il y a là un choix assez heureux de termes désolants, un effet de litanie, les sonorités en première ligne et les idées si elles peuvent suivre. Voila ce que l'on peut essayer de confier au microprocesseur.

Il y a dans **Creative computing**, volume 2, page 223, un article de David H. Ahl, "Roses are red, computers are blue", avec un programme nommé "poet" qui utilise des fragments du **Raven** de Poe. Le programme "bard" fait des poèmes avec n'importe quoi. Page 106, "producing computer poetry" de Margaret Chisman est plus linguistiquement élaboré. Il n'y a pas de programme, mais un choix de termes pour en alimenter un au besoin, sous le nom de "myself manifest". L'idée la plus intéressante consiste à affiner l'effet du hasard en imposant un tirage exhaustif sans remise. La monotonie est ainsi moins évidente. **Pour quelques douzaines de plus** est issu de ces considérations. Le programme s'appelle DOUZPLUS et le fichier des mots est "douze". On peut s'en offir factorielle huit à la puissance quatre versions différentes, deux ou trois milliards de milliards. On s'en lasse avant.

```
10  DISP CHR$(27)&"E"
20  !
30  DISP "*     QUARANTE     *"
40  DISP "* mille douzaines *"
50  DISP "*  de  douzaines  *"
60  DISP "*     de  poemes   *"
70  DISP "* pas tristes par *"
80  DISP "*   J.Dutertre    *"
90  DISP "*  18 juillet 79  *"
100 !
110 DELAY 0 @ DISP
120 WIDTH 32
130 DIM V$[80],V1$[80],V2$[80]
140 INTEGER I,J,K,L,N,R,S
150 K=0
160 INPUT "!video 2therm 3oly 4trac 5bro?";S9$
170 IF S9$="" THEN S=1 @ GOTO 190
180 S=VAL(S9$) @ IF S<1 OR S>5 THEN 160
190 RANDOMIZE
200 ASSIGN # 1 TO "perdreau"
210 IF S=1 THEN DISP CHR$(27)&"E";
220 IF S=2 THEN PRINTER IS ":P1" @ PRINT CHR$(27)&"&k1H"
230 IF S=3 THEN PRINTER IS ":I1" @ CALL "oly" @ PRINT
    CHR$(27)&CHR$(23)
240 IF S=4 THEN PRINTER IS ":P1"
250 IF S=5 THEN PRINTER IS ":I1" @ CALL "bro09"
260 FOR J=0 TO 90 STEP 10
270 IF S=1 OR S=2 OR S=4 THEN 290
280 FOR K=0 TO 3
290 IF S=3 AND K=0 THEN PRINT CHR$(27)&CHR$(23)
300 GOSUB 400
310 IF S=1 OR S=2 OR S=4 THEN 360
320 IF K#3 THEN J=J+10
330 NEXT K
340 DISP "papier et 'P'."
350 D$=KEY$ @ D$=UPRC$(D$) @ IF D$#"P" THEN 350
360 IF S=1 THEN WAIT 5 @ DISP CHR$(27)&"E";
370 NEXT J
380 GOTO 390
390 END
400 IF S#1 THEN PRINT @ PRINT "     ";J/10+1 @ PRINT
410 IF S=1 THEN DISP
420 FOR I=0 TO 490 STEP 70
430 L=INT(RND*7+1)*10
440 READ # 1,I+L ; V$
450 IF S=2 OR S=3 OR S=5 THEN 570
460 N=30 @ IF S=4 THEN N=40
470 IF S=4 THEN 490
480 IF LEN(V$)<=N THEN DISP V$[2] @ GOTO 600
490 IF LEN(V$)<=N THEN PRINT V$[2] @ GOTO 590
500 B$=V$[N,N] @ IF B$#" " THEN N=N-1 @ GOTO 500
510 V1$=V$[2,N-1] @ V2$="  "&V$[N+1]
520 IF S=4 THEN 560
530 IF I=0 THEN DISP CHR$(27)&"H"&V1$&CHR$(27)&CHR$(60);ELSE DISP
    V1$&CHR$(27)&CHR$(60);
540 DISP V2$&CHR$(27)&CHR$(60);
550 GOTO 570
560 PRINT V1$ @ PRINT TAB(40-LEN(V2$));V2$
570 IF S=2 OR S=3 OR S=5 THEN PRINT V$ @ GOTO 590
580 IF S=1 AND I=210 THEN DISP @ DISP @ GOTO 600
590 IF I=210 OR I=490 THEN PRINT @ GOTO 610
600 IF I=1000 THEN DISP
610 NEXT I
620 RETURN
630 ! maj 850321
```

1

Modeste et decide remontant ses bretelles
Et trainant sans espoir ses pas sous le preau
Il chaussait pour marcher ses souliers sans semelles
Et voyait les perdrix devaler le coteau

Elle etait l'hirondelle
Il se sentait nigaud
Comment le savait-elle
La bulle est de niveau

2

Quittant a tout jamais les abords de Sarcelles
Et trainant sans espoir ses pas sous le preau
Il marchait et sautait ainsi qu'a la marelle
Et parvenait enfin au donjon du chateau

Elle etait tire d'aile
Il etait soliveau
Comment le savait-elle
Ainsi du fil de l'eau

3

Fredonnant sans effort le chant du violoncelle
Protege du grand froid par un ample manteau
Il rechauffait sans hate un mechant vermicelle
Et parvenait enfin au donjon du chateau

Elle etait l'etincelle
Il plumait l'artichaut
Du moins le disait-elle
C'est Noel en sabots

4

Reveur et affame pensant a sa gamelle
Le long des chemins creux en poussant son velo
Il chantait des chansons ironiques et belles
Et par dela les jours approchait du tombeau

Elle etait parallele
Il etait nid d'oiseau
Comment le savait-elle
Tovaritch nitchevo

```
PERDREAU   T 2072 08:40 15\04\84
:p list
10 En ecoutant sans fin le cri des hirondelles
20 Quittant a tout jamais les abords de Sarcelles
30 Fredonnant sans effort le chant du violoncelle
40 Modeste et decide remontant ses bretelles
50 Reveur et affame pensant a sa gamelle
60 Suivant en fremissant le vol des bartavelles
70 Saluant d'un sourire un joueur de crecelle
80 Poussant de lourds soupirs a l'ombre des ormeaux
90 Et trainant sans espoir ses pas sous le preau
100 Et malgre la chaleur un frisson sur la peau
110 Protege du grand froid par un ample manteau
120 Pensif a contretemps reveur les pieds dans l'eau
130 Le long des chemins creux en poussant son velo
140 Toujours sur les chemins et par monts et par vaux
150 Il chantait des chansons ironiques et belles
160 Il tentait avec soin d'illustrer les voyelles
170 Il mimait en silence emu Polichinelle
180 Il choyait des pensees qui lui donnaient des ailes
190 Il chaussait pour marcher ses souliers sans semelles
200 Il marchait et sautait ainsi qu'a la marelle
210 Il rechauffait sans hate un mechant vermicelle
220 Et parvenait enfin au donjon du chateau
230 Et malgre les errances atteignait le poteau
240 Et par dela les jours approchait du tombeau
250 Et voyait les perdrix devaler le coteau
260 Et s'amusait longtemps du lourd vol des perdreaux
.270 Et relevait le tete en tirant son chapeau
280 Et meditait sans joie sur la fin des heros
290 Elle etait parallele
300 Elle etait l'hirondelle
310 Elle etait la plus belle
320 Elle etait pele-mele
330 Elle etait immortelle
340 Elle etait l'etincelle
350 Elle etait tire d'aile
360 Il etait tirant d'eau
370 Il etait soliveau
380 Il etait le corbeau
390 Il se sentait nigaud
400 Il plumait l'artichaut
410 Il etait au tombeau
420 Il etait nid d'oiseau
430 Comment le savait-elle
440 Que ne le disait-elle
450 Pourquoi le pensait-elle
460 On y pensait pour elle
470 Qu'en sait-il qu'en sait-elle
480 Malgre lui malgre elle
490 Du moins le disait-elle
500 Ainsi du fil de l'eau
510 On s'enfuit au galop
520 Miracle et berlingot
530 C'est Noel en sabots
540 Tovaritch nitchevo
550 La bulle est de niveau
560 C'est l'ecole de l'echo
```

A suivre...

```
DOUZPLUS                    page 1                    85/05/02

10  DISP CHR$(27)&"E"
20  !
30  DISP " DOUZPLUS"
40  DISP " Pour quelques"
50  DISP " douzaines de plus."
60  DISP " Un poeme pas gai"
70  DISP " de Jean Dutertre"
80  DISP " le 19 sept 1979"
90  !
100  DISP
110  INPUT "1video 2therm 3oly 4trac 5bro?":R$
120  IF R$="" THEN R$="1"
130  R=VAL(R$)
140  IF R=1 THEN PRINTER IS ":D1" @ GOTO 200
150  IF R=2 THEN PRINTER IS ":P1" @ PRINT CHR$(27)&"&k1H" @ GOTO 200
160  IF R=3 THEN PRINTER IS ":I1" @ CALL "oly" @ GOTO 200
170  IF R=4 THEN PRINTER IS ":P1" @ GOTO 200
180  IF R=5 THEN PRINTER IS ":I1" @ CALL "BRO09" @ GOTO 200
190  GOTO 100
200  DIM R(8,8)
210  S$="DansSans" @ O=80
220  ASSIGN # 1 TO "douze"
230  RANDOMIZE
240  FOR J=1 TO 8 @ FOR I=1 TO 8
250  R(I,J)=0
260  NEXT I @ NEXT J
270  READ # 1,INT(8*RND)*10+170 ; T1$
280  READ # 1,INT(8*RND)*10+250 ; T2$
290  T1$[2,2]=UPRC$(T1$[2,2])
300  DISP CHR$(27)&"E"
310  PRINT @ PRINT "  "&T1$&T2$ @ PRINT
320  FOR K=1 TO 4
330  FOR M=4 TO 0 STEP -4
340  FOR L=1 TO 2
350  J=1+M @ GOSUB 480 @ N=I*10+(J-1)*O @ READ # 1,N ; T1$
360  J=2+M @ GOSUB 480 @ N=I*10+(J-1)*O @ READ # 1,N ; T2$
370  J=3+M @ GOSUB 480 @ N=I*10+(J-1)*O @ READ # 1,N ; T3$
380  J=4+M @ GOSUB 480 @ N=I*10+(J-1)*O @ READ # 1,N ; T4$
390  PRINT "C'est"&T1$&T2$ @ PRINT S$[M+1,M+4]&T3$&T4$
400  NEXT L
410  NEXT M
420  PRINT " Ce vent de folie" @ PRINT
430  IF R=1 THEN WAIT 10 @ DISP CHR$(27)&"E"&CHR$(27)&"<"
440  NEXT K
450  FOR L=1 TO 4 @ PRINT @ NEXT L
460  ASSIGN # 1 TO *
470  END
480  I=INT(8*RND+1)
490  IF R(I,J)=0 THEN R(I,J)=1 @ RETURN
500  I=I+1 @ IF I=9 THEN I=1
510  GOTO 490
520  ! maj 850321

CENTMIL                     page 1                    85/05/02

10  DISP CHR$(27)&"E"
20  DELAY 0
30  !
40  DISP "*  CENTMIL   *"
50  DISP "* R.Queneau  *"
60  DISP "* (surtout)  *"
70  DISP "* J.Dutertre *"
80  DISP "* 19 mars 83 *"
90  !
100  DISP
110  DIM V$[80],V1$[80],V2$[80]
120  INTEGER I,J,K,N,R,S
130  K=-1
```

```
140  DISP " Les sonnets dans le..." @ DISP
150  INPUT "1 desordre 2 ordre? ";R9$
160  IF R9$="" THEN R=1 @ GOTO 180
170  R=VAL(R9$) @ IF R<1 OR R>2 THEN 150
180  DISP @ INPUT "1video 2therm 3oly 4trac 5bro?";S9$
190  IF S9$="" THEN S=1 @ GOTO 220
200  S=VAL(S9$) @ IF S<1 OR S>5 THEN 180
210  DISP CHR$(27)&">"
220  RANDOMIZE
230  ASSIGN # 1 TO "queneau"
240  IF S=1 THEN DISP CHR$(27)&"E"
250  IF S=2 THEN PRINTER IS ":P1" @ PRINT CHR$(27)&"&k1H"
260  IF S=3 THEN PRINTER IS ":I1" @ CALL "OLY" @ PRINT
     CHR$(27)&CHR$(23)
270  IF S=4 THEN PRINTER IS ":P1"
280  IF S=5 THEN PRINTER IS ":I1" @ CALL "BR009" @ PRINT
290  IF R=1 THEN 360
300  FOR J=0 TO 90 STEP 10
310  GOSUB 410
320  IF S=1 THEN WAIT 10
330  DISP CHR$(27)&"E"&CHR$(27)&"H";
340  NEXT J
350  GOTO 400
360  GOSUB 410
370  IF S=1 THEN WAIT 10
380  IF S=1 THEN DISP CHR$(27)&"E";
390  GOTO 360
400  END
410  K=K+1 @ IF R=1 THEN J=-10
420  IF (S=3 OR S=5) AND MOD(K,2)=0 THEN DISP "papier, 'CONT' et
     'RTN'." @ STOP
430  IF S#1 THEN PRINT @ PRINT "     ";J/10+1 @ PRINT
440  IF S=1 THEN DISP CHR$(27)&"<"
450  FOR I=0 TO 1300 STEP 100
460  IF R=1 THEN J=INT(RND*9)*10
470  READ # 1,I+J : V$
480  IF S=2 OR S=3 OR S=5 THEN 590
490  IF S=4 THEN 520
500  N=32 @ IF LEN(V$)<=N THEN DISP V$[2]&CHR$(27)&"<"; @ GOTO 590
510  IF S=1 THEN 530
520  N=40 @ IF LEN(V$)<=N THEN PRINT V$[2] @ GOTO 590
530  B$=V$[N,N] @ IF B$#" " THEN N=N-1 @ GOTO 530

540  V1$=V$[2,N-1] @ V2$=V$[N+1]
550  IF S=4 THEN 580
560  IF I=0 THEN DISP CHR$(27)&"H"&V1$:ELSE DISP V1$;
570  DISP TAB(32-LEN(V2$));V2$;
580  IF S=4 THEN PRINT V1$ @ PRINT TAB(40-LEN(V2$));V2$
590  IF S=2 OR S=3 OR S=5 THEN PRINT V$
600  IF S=1 THEN 620
610  IF I=300 OR I=700 OR I=1000 OR I=1300 THEN PRINT @ GOTO 640
620  IF I=700 THEN WAIT 12 @ DISP CHR$(27)&"E" @ DISP CHR$(27)&"<" @
     GOTO 640
630  IF S=1 THEN DISP
640  NEXT I
650  RETURN
660  ! Raymond Queneau
670  ! cent mille milliards de poemes
680  ! NRF Gallimard 1961, 1982.
690  ! maj 850321
```

L'instant d'acces

C'est une aile encore
Sans rire sans folie
C'est l'epee du sort
Sans joie et sans vie
C'est un chien sans but
Dans l'ardeur qu'on sait
C'est un pleur tordu
Dans le monde des plaies
Ce vent de folie

C'est le clan de mort
Sans frein sans envie
C'est un vol dehors
Sans rien sans patrie
C'est un bras de plus
Dans l'abri des rets
C'est un air perdu
Dans la rue sans frais
Ce vent de folie

C'est l'ami d'abord
Sans toit sous la pluie
C'est un bruit a bord
Sans cle ni genie
C'est un or vaincu
Dans l'instant d'apres
C'est un cri decu
Dans la ville des fouets
Ce vent de folie

C'est le creux d'accord
Sans air qu'on oublie
C'est un feu d'alors
Sans but ni amie
C'est le coeur moulu
Dans la nuit d'acces
C'est un corps fourbu
Dans la peur qui nait
Ce vent de folie

```
:cat
DOUZE        T  722 18:53 27\05\84
:list
10 un chien
20 un cri
30 un air
40 le coeur
50 un or
60 un corps
70 un pleur
80 un bras
90 perdu
100 de plus
110 vaincu
120 decu
130 moulu
140 fourbu
150 sans but
160 tordu
170 le monde
180 l'abri
190 la rue
200 la ville
210 l'instant
220 la nuit
230 la peur
240 l'ardeur
250 qu'on sait
260 des rets
270 sans frais
280 d'apres
290 des fouets
300 d'acces
310 qui nait
320 des plaies
330 un vol
340 une aile
350 l'ami
360 un bruit
370 le creux
380 un feu
390 le clan
400 l'epee
410 de mort
420 d'abord
430 d'accord
440 a bord
450 dehors
460 d'alors
470 encore
480 du sort
490 toit
500 air
510 cle
520 but
530 joie
540 rire
550 frein
560 rien
570 sans envie
580 ni genie
590 sans folie
600 et sans vie
610 sous la pluie
620 ni amie
630 sans patrie
640 qu'on oublie
:
```

o

Le vieux marin breton de tabac prit sa prise
pour le fin fond du nez exciter les arceaux
une toge il portait qui n'etait pas de mise
il se trouve aussi sec qu'un vieux sac de fayots

Le cheval Parthenon frissonnait sous la bise
le vulgaire s'entete a vouloir des vers beaux
d'une etrusque inscription la pierre etait incise
les Grecs et les Romains en vain cherchent leurs mots

Du Gange au Malabar le lord anglais zozotte
le chat fait un festin de tetes de linotte
le chemin vicinal se nourrit de crottin

On a bu du pinard a toutes les epoques
on mettait sans facon ses plus infectes loques
le mammifere est roi nous sommes son cousin

o

Lorsqu'un jour exalte l'aede prosaise
depuis que lord Elgin negligea ses naseaux
la decouverte alors voila qui traumatise
elle soufflait bien fort par dessus les coteaux

On etait bien surpris par cette plaine grise
qui se plait a flouer de pauvres provinciaux
aller a la grand ville est bien une entreprise
a tous n'est pas donne d'aimer les chocs verbaux

Du Gange au Malabar le lord anglais zozotte
on comptait les esprits aceres a la hotte
lorsque Socrate mort passait pour un lutin

L'Amerique du Sud seduit les equivoques
frere je t'absoudrai si tu m'emberlucoques
mais rien ne vaut grille le morceau de boudin

Queneau ?

HP-1x

SBM OFFSHORE SYSTEMS

SINGLE BUOY MOORINGS INC.

Member of the IHC SBM Group

Head Office
5, route de Fribourg
P.O. Box 129
CH-1723 Marly, Switzerland
Telephone (37) 46 11 61
Télex 36207 sbm ch

Engineering Office
24, Avenue de Fontvieille
P.O. Box 199
MC 98007 Monaco Cedex
Téléphone (93) 50 45 67
Télex sboy 469887 mc

Your ref

Our ref

Date le 16 Fevrier 1985

Re

Monsieur,

Je me permets de vous envoyer un programme pour H.P. 16. c

Ce programme effectue la traduction d'un nombre flottant 32 bits (PERKIN - ELMER) dans le format H.P. 16. C.

Comme vous le demandez dans votre manuel d'utilisation : " Je soumets volontairement ces informations à la société Hewlett - PACKARD ces informations ne sont pas confidentielles et Hewlett Packard peut en avoir l'usage de son choix sans obligations envers moi ou toute autre personne."

Veuillez agréer, Monsieur, l'expression de mes sentiments les meilleurs.

J.P DENTAL
Computer. Dept.

Publié avec l'aimable autorisation de l'auteur
JDD

SBM OFFSHORE SYSTEMS

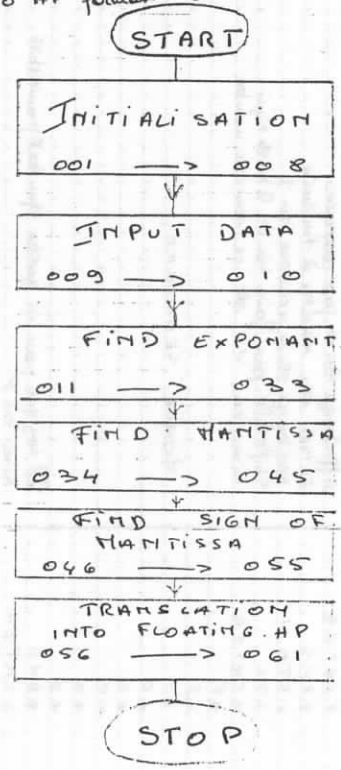
SINGLE BUOY MOORINGS INC.

CALCULATION SHEET

SIGNATURE DENTAL J.P — DATE 12-FEB-85

TITLE HP 16c program — JOB No.

SUBJECT translation floating point number PERKIN into HP format — SHEET No.



[redacted]

PROGRAMME "~~FORTRAN~~"

ASSEMBLEUR HP 16 C

GXF2 0033

PROBLEME	Traduction nombre flottant 32 bits P.E.D.S → H.P.		PAGE 1 de 3
PROGRAMMEUR	DENTAL	DATE 12 - FEB - 1985	IDENTIFICATION

TYPE N°	Suite	ORDRE FORTRAN	
001		43.22. A *LBL A	
002		24 *DEC	changement de la
003		3 *3	longueur du champ
004		2 *2	à des mots de 32 bits.
005		42 44 *WSIZE	
006		23 *HEX	mise en hexa
007		0 *0	
008		36 *ENTER	Affichage de 0 pour commencer
009		31 *R/S	Entrée du nombre à traduire
010		44 1 *STO 1	Stockage de ce nombre en 1
011		42 A *SL	Shift left pour faire tomber le bit de signe
012		36 *ENTER	mise en Y de ce nouveau nombre
013		F *F	
014		E *E	
015		0 *0	Masque FE000000
016		0 *0	
017		0 *0	
018		0 *0	
019		0 *0	
020		0 *0	
021		42 20 *AND	Logique pour ne garder que les 7 premiers bits
022		36 *ENTER	mise en Y
023		7 *7	rotation des sept bits de gauche
024		42 E *RLn	à droite
025		36 *ENTER	mise en Y.

GXF2 0033

IBM

PROGRAMME "FORTRAN"

ASSEMBLEUR HP 16 C

GXF2-0093 0

PROBLEME	TRADUCTION nombre flottant PERKIN ELMER → H.P. format.	PAGE	2 de 3
PROGRAMMEUR	DENTAL	DATE	12 - FEB - 1985
IDENTIFICATION			

ORDRE FORTRAN

TYPE	Suite			
026		4	*4	Soustraction de 40 en raison
027		0	*0	de la méthode du signage de
028		30	*-	l'Exposant chez P.E.D.S. (bit de signe sur 7 bits)
029		6	*6	Soustraction de 6 pour avoir tout
030		30	*-	en puissance de 16 et passer la mantisse en hexa
031		4	*4	multiplication par 4 pour avoir l'exposant
032		20	*x	en puissance de 2 (requis par HP pour traduction)
033		44 2	*STO 2	Stockage en 2.
034		45 1	*RCL 1	On rappelle le nombre original.
035		36	*ENTER	mise en Y.
036		0	*0	
037		0	*0	
038		F	*F	
039		F	*F	masque 00FFFFFF
040		F	*F	
041		F	*F	
042		F	*F	
043		F	*F	
044		42 20	*AND	et logique pour ne garder que les 24 derniers bits.
045		44 3	*STO 3	Stockage en 3 de ce nombre.
046		45 1	*RCL 1	Rappel du nombre original à nouveau
047		42 A	*SL	Mise du bit de signe dans le bit de Report.
048		43 6 4	*F? 4	Test du bit de signe
049		22 C	*GTO C	Si 1 alors on va en C.
050		22 B	*GTO B	Si 0 alors on va en B.

GXF2 0093 0

IBM

GXF2-0093-0

PROGRAMME "FORTRAN"

ASSEMBLEUR HP 16 C

PROBLEME TRADUCTION nombre flottant PERKIN ELMER → HP format.		PAGE 3 de 3
PROGRAMMEUR DENTAL	DATE	IDENTIFICATION

TYPE	Suite	ORDRE FORTRAN	
051		43.28.C *LBLC	LABEL C le nombre est negatif
052		45.3 *RCL 3	on rappelle 3
053		49 *CHS	on change le signe
054		44.3 *STO 3	on le remet en 3
055		43.28.b *LBL B	LABEL B le nombre est positif pas de changement
056		24 *DEC	passage en decimal
057		45.3 *RCL 3	on rappelle la mantisse
058		36 *EXTER	on la met en X
059		45.2 *RCL 2	on rappelle l'exposant
060		42.45.48 *FLOAT .	on passe en mode flottant cachant $X = X*2^X$
061		43.21 *RTN	fin du programme

PPC-T

MAGAZINE

LE COIN DU CLUB

LES JOURNAUX

Reçus depuis MICRO-REVUE No5

Disons d'abord que tous les journaux "club" à qui nous avons envoyé la publicité de IND41 l'ont publiée. En a résulté une commande ferme par EDUCALC de 20 exemplaires, commande qui, nous l'èspérons sera suivie de beaucoup d'autres. Merci à tous.

PPC Journal, Janvier 85 avec un article sur la communication HP-75 réseau genre "the Source" ; un article sur le "translator pack" dont, avec un peu de chance, vous trouverez la traduction dans les pages qui suivent. Disco SKWID et Butcher blocs sont deux rubriques très intéressantes destinées aux débutants. SKWID traite dans ce numéro du langage machine HP-41, Butcher Blocs ("La rubrique du Boucher") traite en général de modifications matérielles. Dans le présent numéro David White (auteur de la rubrique et éditeur de PPC Journal) donne son avis sur l'Intégral PC. Avis mitigé. Je ne peux pas tout donner, les numéros sont riches, en ce moment.

PPC Journal, Février 1985, très intéressant. Un article inattendu sur le format des disquettes IBM-PC, motivé par la disponibilité du lecteur de disquettes STEIMETZ IBM/HP-IL (dont je n'ai jamais réussi à importer un exemplaire, malgré une commande ferme, voir peut-être par EDUCALC). Pour l'instant, la concurrence entre PPC et CHHU nous donne des numéros très intéressants.

Comme vous le savez peut-être, il n'y a plus qu'un journal à PPC, qui regroupe les anciens Calculator et Computer. Il y a maintenant page 2 un "OURS" donnant les tarifs et autres indications. Je ne sais pas si c'est une erreur, mais PPC annonce 12 numéros par an, au lieu de 10 d'habitude. Le prix pour nous est de $37 et il ne semble plus être question de droit d'entrée. PPC PO Box 9599, Fountain Valley, CA 92728-9599, USA.

Je disais plus haut que la concurrence est actuellement bénéfique, je le vois en examinant CHHU CHRONICLE, numéros Décembre 84 et Janvier/février 85. Remarquez au passage, aussi bien pour PPC que pour CHHU, que les dates de parution réelles ne sont guère en correspondance avec la date du journal...

Il est presque impossible, faute de temps et de place, de décrire le contenu de ces journaux. Citons quand même dans CHHU une rubrique régulière SCJ Corner, concernant particulièrement les topographes, rédigée par l'animateur du surveying journal US.

CHHU Chronicle abonnement pour un an et 10 numéros, $34.75 ($41 la première année).

Rappelons que ces deux clubs traitent de tous les portables HP, de la 41 au 71 en passant par le 75. l'Intégral PC et le Portable ne sont pas prévus pour l'instant, la série 80 semble délaissée.

Prisma, numéro de Février 85, couverture orange. CCD est à la recherche d'un nouveau LOGO. Le module CCD semble sorti, mais je n'ai recu aucune indication officielle (il semble que le manuel ne soit pas encore prêt) ; de toute façon, très peu d'entre vous se sont manifestés à ce sujet. Le prix annoncé par EDUCALC est très èlevé (environ 2000F rendu en France), même si les frais de douane sont à déduire dans le cas d'un achat en allemagne, cela semble bien cher. Il se dit que PPC-Paris peut fournir ce module à 1000F. Si c'est encore vrai, précipitez-vous, c'est donné (adresse ci-dessous). CCD Rolf Hansmann, Computer Club Deutschland e. V. Limburger Strasse 15, 6242 Kronberg 2, RFA.

Numéro 10 de JEDI, 15 pages sur FORTH, Pascal, Langage LPB, et la communication par ligne téléphonique. JEDI, 8 rue Poirier de Narcay, 75014 Paris.

Dans le Hewlett-Packard Journal de Avril 85, les français sont à l'honneur, en effet ce sont des français de Grenoble qui ont conçu le terminal vidéo HP 2392A, et qui le décrivent ici (hélas, en anglais !). Il s'agit de Jean-Louis Chapuis, Michèle Prieur, Michel Cauzid, Jean-

Jacques Simon, Christian-Marcel Dulphy, Jean-Yves Chartron, René Martinelli. la Brochure "Mesure actualité" de HP annonce l'Intégral PC.

JPC, journal du Chapitre de Paris, numéros de Mars et d'Avril 85. La présentation a fait de gros progrès depuis que tout est sorti sur traitement de texte HP-71, ce qui doit représenter pas mal d'heures de dactylographie. Le 71 est traité de façon particulièrement importante. Une équipe à l'intention de faire un catalogue des fichiers LEX (langage machine) en cours de réalisation en France. Il n'y en a pas encore beaucoup... JPC/ Philippe GUEZ, 56 rue Jean-Jacques Rousseau 75001 Paris.

Le HP-club Spiegel de Suisse en est à son 13ème numéro. Je suis très agréablement surpris de la qualité du travail de ce club dont la zone de recrutement est réduite. Franco dal Molin, Plattenstraat 44, 8152 Glattenbrugg.

C'est tout pour aujourd'hui.

PETITES ANNONCES

VENDS SYSTEME HP-41/HP-IL

936 registres-6552 octets de mémoire vive et 300 fonctions dans:

1 HP-41CV: calculateur-ordinateur de poche alphanumérique programmable (336 registres-2352 octets de mémoire vive, 128 fonctions, etc.);

1 Module de fonctions d'extension
2 Modules d'extension de mémoire } 47 fonctions, 600 registres-4200 octets;

1 Module horloge de précision avec 53 fonctions;

1 Module HP-II: 64 fonctions de contrôle de la boucle d'interface;

1 Imprimante thermique HP-IL: graphiques, traçage (points), codes barres, minuscules, caractères spéciaux, formattage, intensité var., mémoire (101 car.), mode attente, bat. ou sect., etc.;

1 Lecteur optique de codes barres (8 fonctions);

1 Mallette de système et divers accessoires. Tout pour utilisation immédiate;

16 Bouquins: manuels, programmes, codes barres, explications, périodiques de HP;

Acheté neuf (factures fournies). A peine utilisé (vérifiable). En parfait état. Prix actuel du neuf: env. 15000. Prix d'occasion: 9000 FF.

T360: Vends HP-41C Déc 80 (Réinsé HP-Sept 83)
+ Module Quad : 1000 Frs (Machine Accélérée avec interrupteur)
Module TIME : 300 Frs
Module XFUNCTIONS : 300 Frs
ou le tout pour 1500 Frs

Tel: 530-24-38 LOCUSSE Michel
116 rue de l'Abbé Groult 75015 PARIS.

Vends HP82143A + lecteur optique + module Time + module X Function + module Zenrom 1 (tout de matériel est du début 1984 ou du cours de l'an) valeur environ 5600F vendu 3500F l'ensemble. Urgent.
T445
WROBEL Jacques
118 rue des Alliés
57740 Longeville lès St Avold

Janick Taillandier, Résidence St Clair, 27 chemin du Malvan, 06800 Cagnes/ mer vends HP-75 version bbbbbb avec rom I/O, FORTH (cassette) pour 6000F ; vends également interface RS232/HP-IL, 2500F.

François Eckert, 45 rue du dr Babin, St Germain les Arpajon, tel (6) 016 48 54, Vends HP-41CV, imprimante HP 82143A, lecteur de cartes, modules Time, X-F, X-M, Math, Financial, accessoires divers, le tout 4500F.

T252 Bruno Piguet vends 3 modules simples pour HP-41, excellent état. prix 75F pièce.

F. Darin (T6) vends HP-71 1984 3650F, module HP-IL/71 1200, module 4K 510F, module financier 530F, module FORTH 1200F, interface vidéo HP-IL (tout contrôleur HP-IL) HP82163 2320F, Editeur de texte 71 530F, Lecteur de cartes 71 1030F, franco de port. tel bureau 62.62.75.

NOUVELLES BREVES

Henri Bourboulon, T259, 156 av de la libération, 59140 Dunkerque, a rédigé deux petits programmes pour utiliser la connaissance des temps avec un HP-41 et les communiquera volontier à qui en fera la demande (il n'a pas paru utile de les publier dans la revue, car ils ne doivent intéresser que bien peu de monde.

Le module Translator Pack pour HP-41 qui permet la récupération de programmes de HP-41 sera disponible incessamment chez Hewlett-Packard, au prix de 1542F environ.

Conférence du FORML (Forth Modification Laboratory) en Europe, très précisément à Stettenfels Castle, Heilbronn, République Fédérale d'allemagne, 25 et 27 Octobre 1985. Le thème de la conférence est "la métrologie des programmes", Programmes et méthodes pour mesurer les performances, la complexité, la structure, des programmes, la productivité, les méthodes de développement, les modèles, les outils des programmeurs, les aides et les procédures de vérification des programmes. Informations auprès de EuroFORML, Forth Interest Group, Post office Box 8231, San José, CA 95155, USA phone Fig Hot Line (408) 277-0668. Si l'un d'entre vous peut y assister, je suis très intéressé par un compte rendu. JDD.

Philippe Romascano, animateur du club PPC-Lausanne, a été embauché par Hewlett-Packard. Il est remplacé à la tête du club par Luzius Auer, Simplon 26, 1020 RENENS, Suisse. Il y a longtemps que je n'ai pas vu un journal de PPC Lausanne, mais je suis content de voir que les activités locales continuent. PPC Lausanne Case Postale 22, CH-1049 Assens, Suisse.

François Le Grand (T391) Rue Josse Impens 99, 1030 Bruxelles Belgique, se propose d'établir un ensemble de programmes de navigation marine pour le HP-71, d'abord en BASIC, ensuite en FORTH éventuellement. Si vous êtes intéressés, écrivez-lui. Si vous êtes nombreux, pourquoi pas une EPROM Navigation ? (cf "sur le Marché" pour le lecteur d'Eprom).

Le FIG-FORTH des Alpes (Georges Seibel) vient d'inaugurer un local, Immeuble "La citadelle", 19 Av. des Hirondelles, 74000 Annecy. Téléphone (50) 57 02 80.

Franck Lebastard, l'animateur bien connu du groupe Bretagne de PPC-T, qui a émigré à Lyon, vient de lancer avec d'autres membres de PPC-T un club IBM-PC et compatibles qu'il veut animer dans l'esprit de PPC. Certains compatibles IBM, comme le SANYO 550 sont très bon marché, et peuvent servir de support aux machines HP. Cet aspect sera rapporté ici. On peut faire du FORTH, bien sur, sur ces machines. Qui a les références d'un FORTH IBM facilement disponible et à un prix raisonnable? je suis preneur. Profitons de cette information pour dire que le club va être en mesure de traiter vos articles sur disquettes compatibles IBM (je suis maintenant, entre autre, revendeur SANYO), ainsi que sur HP-150. Me contacter pour les détails pratiques.

La SCIP d'Etienne Poupée a des problèmes. Ses pratiques commerciales (en tous cas vues par notre bout de la lorgnette) ne s'étant pas améliorées, est-ce étonnant ? En tout cas ils ont sponsorisé un bateau à la course de l'EDHEC et semble-t-il pour le tour de France à la voile, pour la promotion de leur module navigation. SCIP 10-12 rue Deltéral 93310 Le Pré st Gervais. Tel 840 58 29.

Un groupe FORTH en Allemagne : Common Interface Alpha, Schanzenstrasse 27, 2000 Hamburg 6, RFA. Il édite un journal, VIERTE DIMENSION.

Certains produits viennent d'être retirés du catalogue des photocopies et de la coopérative de PPC-T. Il s'agit de P7 Microcode PPC-1, qui fait double emploi avec le "microcode HP", mais les originaux restent disponibles si vous le voulez vraiment. Par contre les modules PPC-ROM et ZENROM ne seront plus disponibles chez nous. Les difficultés que nous avons à obtenir des livraisons (couramment 3 mois de travail et au moins autant de délai de livraison) et le prix (trop) élevé qui en résulte nous ont découragé. Pour le PPC ROM, vous pouvez vous adresser directement à EDUCALC 27953 Cabot Road, Laguna Niguel, CA 92677 USA, prix $84.95 + $15 pour l'avion + 30% de TVA et de douane, auxquels vous avez une chance d'échapper si les douaniers ne voient pas le module (caché par le manuel). Nous nous n'y échappons pas. Vous pouvez payer par carte VISA ou Master Charge. Pour le ZENROM, voyez ZENGRANGE Ltd, Greenfield Road Leeds LS9 8DB, West Yorkshire, Angleterre. Prix 85 livres + 4 livres de port + 20% de TVA à l'arrivée. Les livres sont à payer d'avance, les cartes de crédit ne sont pas acceptées.

PROGRAMMATHEQUE

PROGRAMMES DISPONIBLES
CHEZ T33
DAMIEN DEBRIL
38 RUE DU 8 MAI 1945
59190 HAZEBROUCK
FRANCE
T:16(28)41.40.68

POUR LES CARTES PREVOIR
ENVELOPPE RETOUR AFFR.
A 1,70F OU 2,10F.
POUR LES CASSETTES
METTRE 2,40F OU 3,70F.
PREVOIR UN EMBALLAGE
SOLIDE CAR IL Y A DEJA
EU DES DEGATS DURANT LE
TRANSPORT PAR LA POSTE.
J'AI MIS LES NOMS SOUS
LESQUELS LES PROGRAMMES
SONT STOCKES.
VOUS POUVEZ PREPARER LES
CARTES EN Y INSCRIVANT
LES NOMS DES PROGRAMMES.
PREVOIR QUELQUES CARTES
SUPPLEMENTAIRES POUR
REMPLACER CELLES QUI
SERAIENT DEFAILLANTES.

H P 4 1

V1N2 P 3 -1 C -CREA
V1N2 P 3 -1 C-LISTE
V1N2 P 0 -1 C-ED
V1N3 P 4 -1 C-DF
V1N3 P 5 -4 C-CDR
V1N4 P 9 -1 C-T
V1N4 P 10 -2 C-INTER
V1N4 P 11 -2 C-TAB
V1N4 P 11 -2 C-651

No 1 P 5 -1 C-FACTO
No 1 P 5 -1 C-ALFA
No 1 P 6 -1 C-ALFA2
No 1 P 6 -1 C-ALFA3
No 1 P 7 -1 C-LB
No 1 P 7 -1 C-DN
No 1 P 8 -2 C-PILOTE
No 1 P 13 -3 C-CFA
No 1 P 13 -1 C-DD
No 1 P 13 -1 C-LFA
No 1 P 13 -1 C-DH
No 1 P 16 -1 C-QUOT

No 2 P 3 -3 C-WI
No 2 P 4 -3 C-RE
No 2 P 5 -2 C-BASES
No 2 P 5 -1 C-PAGE

No 3 P 14 -2 C-CALAN
No 3 P 15 -2 C-WATT
No 3 P 17 -3 C-BAIRST
No 3 P 17 -3 C-FERRAIL
No 3 P 18 -2 C-TAB-MUL
No 3 P 18 -1 C-MGN
No 3 P 19 -2 C-HIDDEN
No 3 P 19 -5 C-LEM
No 3 P 21 -6 C-DISBIN
No 3 P 23 -1 C-LON

No 4 P 3 -2 C-LEFT
No 4 P 4 -1 C-HIDDEN2
No 4 P 4 -4 C-MIND
No 4 P 5 -1 C-DPC
No 4 P 6 -1 C-MA
No 4 P 7 -1 C-CREAT
No 4 P 7 -1 C-PD
No 4 P 25 -2 C-HDUMP
No 4 P 26 -3 C-SPEC
No 4 P 27 -3 C-FERME C
No 4 P 8 -1 C-ASS
No 4 P 8 -8 C-RC
No 4 P 8 -1 C-RCKEYS
No 4 P 8 -1 C-RCDATA
No 4 P 8 -8 C-R

No 5 P 3 -5 C-CAISSE
No 5 P 3 -3 C-CA
No 5 P 4 -2 C-BLJACK
No 5 P 5 -1 C-P/
No 5 P 5 -1 C-IGR
No 5 P 6 -2 C-TRANS
No 5 P 6 -1 C-OM
No 5 P 9 -2 C-EDTEXTE
No 5 P 10 -1 C-FACTO2
No 5 P 10 -2 C-MM
No 5 P 14 -1 C-KAT71

No 6 P 4 -1 C-DX
No 6 P 5 -2 C-RLM
No 6 P 6 -1 C-D-H
No 6 P 6 -1 C-TX
No 6 P 6 -1 C-IH
No 6 P 7 -1 C-EAO
No 6 P 8 -1 C-DP
No 6 P 9 -2 C-PX
No 6 P 27 -4 C-MICRO-C
No 6 P 27 -4 C-M

No 7 P 3 -2 C-C-A
No 7 P 3 -1 C-REVEIL
No 7 P 4 -1 C-P
No 7 P 8 -3 C-CBG1
No 7 P 8 -1 C-IP
No 7 P 11 -6 C-AY
No 7 P 11 -4 C-AYDATA
No 7 P 12 -1 C-TSYNT
No 7 P 12 -1 C-SL
No 7 P 12 -1 C-PPC-T
No 7 P 13 -1 C-B-D
No 7 P 13 -1 C-DATA
No 7 P 13 -1 C-CDE
No 7 P 14 -1 C-ACC
No 7 P 15 -1 C-CAN
No 7 P 15 -1 C-PND
No 7 P 16 -2 C-PCB
No 7 P 18 -1 C-TRI
No 7 P 19 -2 C-SAV
No 7 P 21 -1 C-FAC
No 7 P 21 -1 C-ORP
No 7 P 22 -2 C-SAVE
No 7 P 24 -1 C-DEV

No 8 P 43 -5 C-CODEHP
No 8 P 44 -1 C-DESAS
No 8 P 44 -1 C-INPUT
No 8 P 45 -2 C-T0
No 8 P 45 -1 C-C=10

No 8 P 46 -1 C-XQ
No 8 P 47 -1 C-CHR$
No 8 P 47 -5 C-ASSEMB
No 8 P 50 -1 C-INPUT2
No 8 P 51 -2 C-CHR$2

No 9 P 56 -2 C-PEL
No 9 P 57 -1 C-PEL2

No 10 P 20 -2 C-TIR
No 10 P 21 -3 C-JACK
No 10 P 54 -1 C-IN2
No 10 P 55 -1 C-CA/CR
No 10 P 56 -1 C-CA/CRDA
No 10 P 56 -2 C-GAM+
No 10 P 61 -2 C-APRO

No 11 P 47 -1 C-DES
No 11 P 48 -1 C-RENAME
No 11 P 49 -1 C-F+

H P 7 5

No 10 P 13 - ENPAGOLY
No 10 P 14 - ENPAGE75
No 10 P 15 - LISTOLY
No 10 P 16 - AZERTAC
No 10 P 16 - OLY
No 10 P 17 - COPCAS
No 10 P 17 - NETTCAS
No 10 P 18 - DUMPCAS

M I C R O - R E V U E

M R 1

41 P 8 -1 C-DCB
41 P 15 -2 C-SYMB
41 P 18 -4 C-INVC
41 P 21 -2 C-HIST
41 P 25 -3 C-TRIPLOT
41 P 26 -1 C-ROM
41 P 28 -1 C-PUISS
41 P 29 -1 C-VM
41 P 29 -1 C-PM
41 P 30 -1 C-CHRO
41 P 30 -1 C-CHROV
41 P 33 -1 C-AS/B
41 P 33 -2 C-AS/C
75 P 36 - -DUMP
75 P 37 - -MAKEFILE
75 P 38 - -UTIL3
75 P 38 - -MEMDIR
75 P 40 - -MEMDUMPY
75 P 42 - -CATALPHA
41 P 93 -2 C-BA
41 P 94 -1 C-BCPLOT
41 P 98 -2 C-BC162
41 P 111 -1 C-CB
41 P 111 -1 C-XR
41 P 0 -2 C-CBPG

M R 2

41 [illegible] -1 C-DC
41 P [illegible] -1 C-DCPGDV
41 P 63 -1 C-DCPGH

M R 3

41 P 40 -1 C-DERIVEE
41 P 42 -1 C-TF
41 P 49 -7 C-DONSOTI
41 P 52 -6 C-DON
41 P 57 -1 C-LST
41 P 57 -3 C-BAPBORN
41 P 61 -2 C-TRACE
41 P 67 -2 C-CBD
41 P 67 -2 C-CBD
41 P 70 -1 C-CBNN
41 P 83 -1 C-CAR
75 P 107 - -CREECAS!
75 P 108 - -ECRICAS!
75 P 110 - -LIRECAS!
75 P 111 - -UNECOPIE

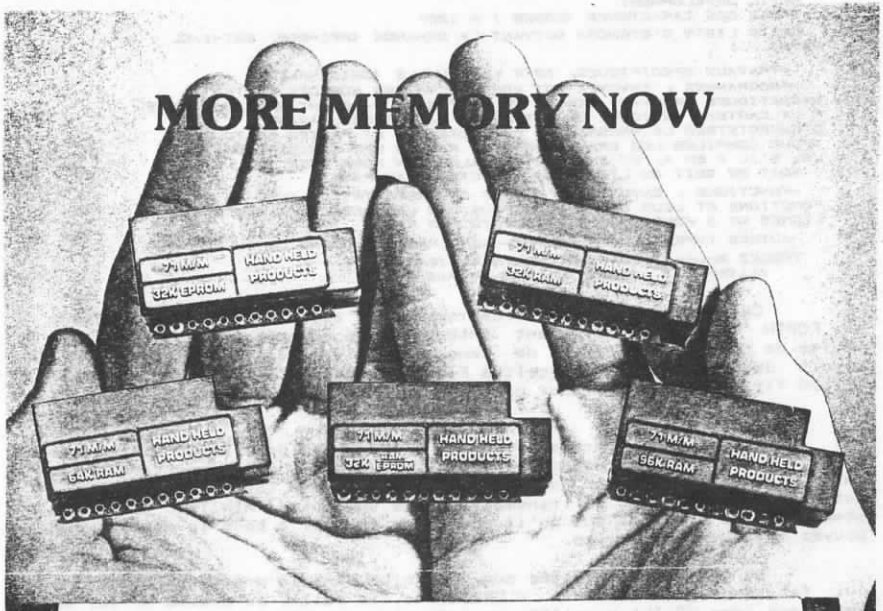
M R 4

41 P 11 -1 C-VR
41 P 11 -1 C-INI
41 P 12 -3 C-DGA
41 P 12 -3 C-DGA1
41 P 12 -3 C-NOM
41 P 15 -4 C-DOMINO
41 P 17 -6 C-CHECKUP
41 P 18 -1 C-TEX
41 P 18 -1 C-TT
41 P 20 -1 C-IDRE
41 P 23 -4 C-DIR
41 P 23 -2 C-ARRA
41 P 25 -2 C-VMIND
41 P 30 -2 C-CA
41 P 46 -1 C-STOD
41 P 47 -1 C-PURDIR
41 P 70 -8 C-CHIESA
75 P 93 - -ROMPEEK
75 P 98 - -ROMDIR
75 P 100 - -MEMLEX
75 P 100 - -DISPLEX
75 P 103 - -DUMCAS3
75 P 104 - -CREECAS2
75 P 105 - -ECRICAS4
75 P 107 - -LIRECAS!
75 P 108 - -ENPAGE62

M R 5

41 P 8 -1 C-DECR
41 P 12 -1 C-LABYDA
41 P 12 -2 C-LABY
41 P 15 -1 C-ST
41 P 18 -1 C-PKDTE
41 P 18 -1 C-UPDTE
41 P 22 -1 C-DL
41 P 22 -1 C-DOUTE
41 P 35 -3 C-TR
41 P 38 -6 C-CALPOL

SUR LE MARCHÉ



NEW FROM HAND HELD PRODUCTS

Upgrade your Hewlett-Packard HP-71 hand held computer instantly! Hand Held Products has responded to the clamor of all HP-71 users who need more memory. The HP-71 is a strong tool for a host of applications but is limited to only 17.5K of Random Access Memory (RAM). With increased memory, the HP-71 is transformed into a powerhouse whose potential is finally realized! In response to the user demand, Hand Held Products is pleased to announce not only one, but five ways to dramatically increase the memory of your HP-71.

All Memory Modules are enclosed within the card reader space of the HP-71, are entirely self-contained, and can be literally installed in seconds. The 64K RAM and 32K RAM modules have enclosed battery back-up and can be removed from the HP-71 without fear of memory loss. Hand Held Products also provides other products to support the HP-71 and 71 Memory Modules. A cross assembler which allows you to download assembly language programs into your HP-71 is available as well as ROM development software to enable you to burn your own EPROMS. Hand Held Products is very proud to add the 71 Memory Modules and support products to its rapidly growing family of state-of-the-art, custom, portable electronics.

HHP-71 M/M 96K	RAM	— $995.00
HHP-71 M/M 64K	RAM	— $695.00
HHP-71 M/M 32K	RAM	— $395.00
HHP-71 M/M 32K	RAM EPROM CARRIER	— $495.00
HHP-71 M/M 32K	EPROM CARRIER	— $ 99.00

Available Now! For complete information:

HAND HELD PRODUCTS, INC.
P.O. Box 2388, Charlotte, North Carolina 28211

Telephone (704) 541-1380
Volume Discounts Available

TWX 810-621-0380 PROTECH
VISA and MasterCard Accepted

SERVICE CODES-BARRES

CODES-BARRES HP-41 SUR TRACEUR 7470A POUR LES ADHERENTS PPC-T.
DEUX SORTES SONT PROPOSEES :

+PLANCHES DE FONCTIONS CLASSEES, 4 F LA FEUILLE (PHOTOCOPIE)
X-FONCTIONS ET TIME (Y COMPRIS CX)
HP-IL ET LECTEUR OPTIQUE
MODULE GRAPHIQUE ET LECTEUR OPTIQUE
LECTEUR DE CARTES ET EXPOSANTS REDUITS (E)
FONCTIONS DE BASE (CAT 3) ET SYNTHETIQUES (Q LOADER, BYTE-G...)
HP-IL DEVELOPMENT
TABLE DES CARACTERES (CODES 1 A 126)
CETTE LISTE S'ETENDRA SUIVANT LA DEMANDE (PPC-ROM, EXT-I/O, MATHS...)

+TRAVAUX SPECIFIQUES, 10 F LA FEUILLE (ORIGINALE)
-PROGRAMMES : ENVOYEZ MOI VOS PROGRAMMES SUR DES CARTES MAGNETIQUES (POUVANT SERVIR EN PARTIE AU PAIEMENT A RAISON DE 3 F LA CARTE). INDIQUEZ UN TITRE <25 CARACTERES. AVANT D'ENREGISTRER LE PROGRAMME, COMPACTEZ LE ET FAITES LE TOURNER (POUR COMPILER LES BRANCHEMENTS). FAITES LES ASSIGNATIONS DE LBL S'IL Y EN A, ET ENREGISTREZ ALORS EN MODE USER.
SOIT 22 SOIT 26 LIGNES DE 13 OCTETS PAR PAGE.
-FONCTIONS : ENVOYEZ UN MODELE DE PRESENTATION AVEC LE NOM DES FONCTIONS ET LEUR CODE XROM (POUR LES MODULES). MAXIMUM : 21 LIGNES DE 6 FONCTIONS DE 2 OCTETS PAR PAGE.
-AUTRES TYPES : DATA, ALPHA, ASN, PANACHE... ME CONSULTER :
TREDEZ Bruno T120 16 CHEMIN St JEAN
ALLONVILLE 80260 VILLERS-BOCAGE

Ceux d'entre vous qui comprennent l'allemand et s'intéressent au FORTH seront certainement intéressés de connaitre l'adresse d'une source de documents FORTH et de langages, livres... en Allemagne. Il s'agit de FORTH-SYSTEME Angelika Flesh, Schützenstr. 3, postfach 1311, D-7820 Titisee-Neustadt. Leur catalogue est un des plus complet que j'ai vu.

Jean-Claude Bornes (T175) 10 allée de la Brie 91000 Lisses est en mesure de fabriquer des interfaces HP-IL/MINITEL qui permettent d'une part d'utiliser l'écran du minitel comme vidéo (avec quelques limitations), d'autre part de transmettre vers un autre MINITEL équipé de même. Je ne suis pas sur qu'il pourra en fournir à tout le monde. Vous pouvez toujours lui écrire.

La société américaine que j'ai cité dans le précédent numéro et qui fabrique des lecteurs d'EPROM de la taille du module HP-IL, un interface (non HP-IL) parallèle pour la 41, un accessoire (non HP-IL) qui transforme la 41 en multimètre numérique programmable, est CMT 33815 Eastgate Circle, Corvallis, OR 97333. tel (503) 752-5456. Les lecteurs d'EPROM seront des 16K, 12K par pagination (une seule adresse). prix public prévu, $195, remise par quantité. disponible été 85.

NOTICE DE LOGICIEL
DE COMPTABILITE MEDICALE.

VERSION DE BASE:

Comptabilité journalière des honoraires;

Enregistrement possible de 50 actes dont 40 caractères réservés pour la saisie des noms et autres renseignements complémentaires.
Les saisies des actes,du montant des honoraires et du mode de paiement s'effectuent à l'aide d'un minimum de frappe au clavier.

L'horloge fournit une chronologie horaire des actes de la journée.

Comptabilité des dépenses;
Permet l'enregistrement de 7 dépenses dans la journée dont 40 caractères réservés pour l'identification de la dépense.
A la fin du mois,l'imprimante effectue un relevé des dépenses mensuelles jusqu'à 70 dépenses.

-Toutes les entrées et sorties chiffrées acceptent des nombres au moins 100 fois supérieurs aux chiffres habituels.
-Il existe des procédures de corrections d'erreurs à chaque entrée au clavier.

-Les capacités de la version de base peuvent être doublées ou triplées suivant les besoins particuliers.

Les résultats journaliers et mensuels s'affichent sur l'imprimante à jet d'encre sur 80 colonnes au format standard.

L'ordinateur envoie la date et l'heure de la sortie sur l'imprimante de la comptabilité des honoraires et des dépenses.
SONT AFFICHES:
-La quantité de chaque catégorie d'actes pratiqués avec le total des honoraires correspondant.
-La ventilation fiscale des actes de la journée avec le montant de chaque honoraire,le mode de paiement de chaque acte,et l'honoraire correspondant.
-Les montants dans chaque catégorie de mode de paiement:
-espèces
-chèques bancaires
-chèques postaux
-accidentsdu travail
-aides médicales
-impayés
-La recette de la journée.
-Les liquidités en caisse.
-Le total des chèques bancaires et postaux.
- Les effets à recevoir.
- Les avoirs crédités.
- Le reste des avoirs à recevoir.
- Le solde mensuel cumulé.
- Le ventilation des dépenses de la journée.
- Le total journalier des dépenses et le solde mensuel.
- Le solde annuel cumulé.

A la fin du mois,l'imprimante affiche:
- Le relevé des dépenses du mois.
- Le relevé des actes du mois,les totaux dans chaque rubrique est le pourcentage de ces sommes par rapport à la recette mensuelle.
- La recette encaissée du mois.
- La recette des actes du mois.
- Le total des dépenses du mois.
- Le pourcentage des frais par rapport aux recettes du mois.

A tout moment et à la fin du trimestre,peuvent s'afficher les indices d'activité des TSAP que le médecin peut ainsi comparer avec ses propres relevés des caisses.

Enfin,il est possible d'effectuer des adaptations personnalisées pour des besoins particuliers notamment pour des cabinets de groupes de médecins généralistes ou spécialistes.

Docteur LERAY
29 253 ILE de BATZ
(98) 61 7919

T360: Michel LOCUSSE 116 rue de l'Abbé Groult 1/2
75015 PARIS.

Je commercialise pour les possesseur d'une HP41 C ou CV (uniquement), d'un lecteur d'Eproms, d'un module XFUNCTIONS, et à la rigueur d'un module TIME, un jeu d'Eproms contenant toutes les fonctions de la HP-41CX. (sauf. SWPT), ainsi que la plus part des ses caractéristiques c.a.d.

NCAT	1	idem CAT 1
	2	— 2
	3	— 3
	4	— ENDIR
	5	— NALMCAT
	6	— CAT 6 de la 41-CX (Catalogue des Assignations)

NPOSFL idem POSFL (41CX)

NALMCAT: idem ALMCAT (CAT EMPTY ok vide)

+ 14 nouvelles fonctions:

CLEMX efface tous les fichiers à partir du Xième fichier (compris)
CLEMA " " " à partir du fichier nommé en Alpha
CLREM " " les fichiers ! (DIR EMPTY)
FLNAME Donne le nom du fichier travail
FLPACK 'Pack' le fichier ASCII courant
FLTYPE Donne le type du fichier nommé en Alpha : PR, DA, AS...
XBST } Automatique si la touche reste enfoncée
XSST }
XRENAME Renomme les fichiers d'extension
MANT Donne la mantisse de X.
REG? Donne le nb de registres disponibles
PURFLX Purge le Xième fichier
TYPE X change le type du fichier nommé en Alpha M, PR, AS.
RCLEM. Uniquement pour les possesseurs du module XFCW version 1B
Récupère les fichiers d'extension détruits par une fonction exécutée sans fichier travail
Ex: GETX, SAVEX, RCLPT...

CATALOG:

-CX ROM 1A	-CX TIME
NALMCAT	CLALMA
NCAT	CLALMX
NPOSFL	CLRALMS
SWPTX	RCLALM
-CX EXT FCN	-EXT FNS- +
ASROOM	CLEMA

KRISTAL INFORMATIQUE
Chemin des Clos.2KST
38240 MEYLAN
tel: (76)90.38.13

JFG 20/02/85

INTERFACE GRAPHIQUE HPGL

(HPIL ---> CENTONICS)

L' interface HPGL permet d'utiliser une mini table traçante type Tandy CGP 115 sur la boucle HPIL avec un controleur utilisant le langage graphique Hewlett-Packard (HPGL). Elle est principalement destinée aux machines des séries 40 et 70. Elle est utilisable soit en imprimante (compatible avec toute imprimante type Centronics), soit en table traçante (avec l'imprimante CGP115 uniquement). Elle est reconnue comme telle par le controleur suivant le cas (identificateurs accessoires AID = 46 et 96). Ceci permet l'utilisation des fonctions de haut niveau du controleur: mode AUTOIO sur la 41, identificateurs PRINTER et GRAPHIC avec le 71.

CARACTERISTIQUES GENERALES :

Mémoire tampon de 1,5 ko. (possibilité de 7,5 ou 15,5 ko)

Interface HPIL :
Entièrement compatible HPIL.
Réponds aux demandes d'identification.

Centronics :
8 bits parallèles, utilise BUSY et STROBE.

Alimentation externe 9 V ou par le Centronics.

Mode HPGL :
Compatible avec la CGP115.
Zone de trace de 96 x 133 mm par défaut, 96 x 200 max
4 couleurs de tracé
Résolution de 0,2 mm

A la mise sous tension, l'interface est dans le mode spécifié par un microinterrupteur. Le passage du mode imprimante au mode graphique HPGL se fait par l'envoi du caractère de controle 19. L'interface convertit alors les instructions HPGL issues d'un controleur en instructions élémentaires pour la CGP115. Le retour au mode imprimante se fait par l'envoi de la chaine 'EX'.

Compatibilité avec le langage HPGL :

Réponds aux commandes suivantes :

AF : advance full
CP : character plot
DI : direction
DF : default
DP : digitize point
EX : exit (sortie mode HPGL)
IP : input points
IN : initialize
IW : input window

LB : label
LT : length
OA : output actual
OC : output commanded
OD : output digitized
OE : output error
OI : output indentifier
OP : output points
OS : output status
PA : plot absolute
PD : pen down
PR : plot relative
PU : pen up
SP : select pen
SR : size relative
XT : X tic
YT : Y tic

Utilisation avec la HP 41 :

Mode imprimante :

Le mode imprimante est utilisable avec toute imprimante comportant un connecteur sortie parallèle (type Centronics).

En mode imprimante, les fonctions générales du module HPIL sont utilisables en mode AUTOIO. Cependant , certaines fonctions prévues pour l'imprimante HP82162 ne donnent pas de résultat valide, en particulier les fonction semi-graphiques: ACCOL, ACSPEC, FMT, REGPLOT, SKPCHR, SKPCOL, STKPLOT.

L'interface assure la conversion des caractères spéciaux de la HP41. Ceci est particulierement important pour l'impression de liste de programmes.
Les caractères spéciaux de la HP 41 sont remplacés par les caractères suivants :

◆ -> *
≠ -> #
Σ -> s
⊢ -> &

Les caractères de controle d'origine de l'imprimante sont conservés, y compris le mode graphique (se référer au manuel de l'imprimante). La seule variante concerne le code de changement de couleur (en mode texte) qui est maintenant égal à 31 afin d'éviter le conflit avec le caractère ≠ de la HP41 (code 29). Tous les logiciels écrits pour la CGP115 sont donc exécutables par l'intermédiaire de l'interface.

Mode graphique HPGL :

L'utilisation du module PLOTTER, sans être strictement nécessaire, est fortement recommandé. Toutes les fonctions de ce module sont utilisables, avec quelques limitations dues a la table tracante utilisee.

Instructions éxécutables sans limitation :

CLIPUU, DRAW, FRAME, GCLEAR, IDRAW, IMOVE, IPLOT, LABEL, LOCATE, LORG, LXAXIS, LYAXIS, MOVE, PCLBUF, PDIR, PEN, PENDN, PENUP, PINIT, PLOT, PLREGX, PRCL, RATIO, RPLOT, SCALE, SETGU, SETUU, UNCLIP, WHERE, XAXIS, XAXISO, YAXIS, YAXISO.

Instructions éxécutables avec quelques limitations :

CSIZE : La taille de caractères est arrondie à la plus proche valeur possible de la table tracante.
CSIZEO : idem CSIZE, de plus pas d'inclinaison possible.
DGTIZE : la table ne peut pas être utilisée en digitaliseur,

l'exécution de DGTIZE renvoie la position actuelle du stylo.

LDIR : seules 4 directions sont possibles : haut, bas, droite, gauche.

LIMIT : les limites physiques de tracé sont 199,8 mm en x et 96 mm en y. L'exécution de LIMIT avec des valeurs supérieures entraine l'apparition du message d'erreur "PL:RANGE ERR".

LOCATD : non utilisable : définit une fenêtre de tracé nulle.

LTYPE : 2 types de tracé : continue ou pointillés.

LTYPEO : idem LTYPE, cependant permet de définir la longueur des pointillés.

TICLEN : sans effet.

Fonctions de tracé de codes barres :

Toutes les fonctions de tracés de codes barres sont utilisables. Du fait de l'utilisation par la HP41 d'unités absolues, les codes barres ont la même taille que ceux tracés sur d'autres tables, seule la mise en page est à modifier (programmes PLOTBC et XBC du manuel PLOTTER).

Exemple sur HP41 : (avec module Plotter)

```
01*LBL"EXEMPL"
02 19
03 ACCHR          envoie caractère 19
04 FRAME          trace un cadre
05 2
06 PEN            change de couleur
07 20
08 80
09 20
10 80
11 LOCATE         nouvelle zone de tracé
12 FRAME          trace un cadre d'une autre couleur
13 GCLEAR         avance le papier
14 "EX"
15 OUTA           sort du mode graphique HPGL
16 PRSTK          imprime le contenue de la pile
17 PRA            et le registre ALPHA.
18 END
```

INTERFACE HPIL :

L'interface répond aux demandes d'identification accessoire et appareil :
id. accessoire : 46 en mode imprimante, 96 en mode HPGL.
id. appareil : "KR9604A".

L'interface possède un octet d'état qui peut être lu par le controleur par une interrogation série.

En mode imprimante, l'octet d'état est toujours nul.

En mode HPGL, les bits ont la signification suivante :

bit 0 : Etat du stylo : 0 = levé, 1 = baissé.
bit 1 : Nouvelles limites. Mis à 1 apres IP, mis à 0 apres OP.
bit 2 : Digitalisation. Mis à 1 apres DP, mis à 0 apres OD.
bit 3 : Mis à 1 pendant l'initialisation, mis à 0 apres OS.
bit 4 : Traceur pret. Toujours 1.
bit 5 : erreur commande. Mis à 1 lors de la réception d'une commande invalide, mis à 0 apres OE.
bit 6 : toujours 0.
bit 7 : toujours 0.

A la recéption des ordres HPIL DCL (Device Clear) et SDC (Selected Device Clear), l'interface efface sa mémoire tampon, mais ne change pas de mode de fonctionnement.

L'ORGANISATION DE PPC-TOULOUSE

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

ASSOCIATION PPC-T

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

RESPONSABLES DE PPC-T

Président: Jean-Daniel Dodin (T1) 77 rue du Cagire 31100 Toulouse (nouvelles adhésions, journaux de l'année en cours, édition du journal, réception des articles).
Trésorier: Jean-François Sibille (T15) Résidence du Pays d'Oc, rue des Genets 31500 Toulouse (renouvellement d'adhésions, problèmes financiers...).
Trésorier adjoint: Jean-Pierre Baudoin (T131) Lieu dit "Les Carabiniers" Castelnau d'Estretefond 31620 Fronton (fichier).
Secrétaire: Olivier Monachon (T13) 1 rue Louis Vignes 31200 Toulouse (librairie et coopérative)
Secrétaire adjoint: Gilles Barret (T22) 92 av de Muret 31300 Toulouse.
Secrétaire adjoint: Jean-Yves Pasquier (T18) Apt. 10, 2 rue de Grèce 31000 Toulouse.
Programmathèque: Damien Debril (T33) 38 rue du 8 Mai 1945 59190 Hazebrouk (envoyez les cartes vierges, elles reviennent pleines).
Responsable service photocopies : Bruno Piguet, Ch. B34 Ecole Nationale de la Météorologie, 42 Av. de Coriolis, 31057 Toulouse Cedex.
Coordinateurs régionaux:
Bretagne: Olivier Pougeon 6 Bd Chanard 56170 Quiberon
Paris: Olivier Arbey (T164) 34 av de la République 94100 St Maur, Robert Schwartz (T178) 141 av Carnot 93140 Bondy, Eric Gengoux 8 rue de Furstenberg 75006 Paris, J.J. Dhenin 83 rue de Gergovie 75014 Paris, Jean Thiberge (T165) 104 Quai Louis Blériot 75016 Paris, Serge Vaudenay (T270) 62 av Ardouin B3 94420 Le Plessis Trévise, Stéphane Barizien (T149) 39 rue St Fargeau 75020 Paris
Lorraine: Dominique Talon (T311) 90100 Courcelle
Sud Est: Eric Angelini (T102) 14 av Marianna 06000 Nice
Besançon: E. Piérunek (T76) chez Mme Faivre 27 rue O. David 25000 Besançon.
Picardie: Bruno Tredez (T120), 16 chemin St Jean Allonville 80260 Villers Bocage.
Région Rhône: Yves Alajouanine (T144), 108 bis rue Hénon 69004 Lyon, et Franck Lebastard Apt 1085 15 Chemin des Plates 69120 Vaulx en Velin tel (7) 880 77 34.
BELGIQUE: Bernard Branquart (T372) 32 Thiarmont, 7190 Ecaussines Belgique.
Des coordinateurs sont recherchés pour toutes les autres régions. Contactez le coordinateur de votre région pour organiser avec lui des réunions, démonstrations,...Vous pouvez obtenir un budget du club sur présentation d'un projet. Nous pouvons vous imprimer vos convocations, des affiches,...Contactez vous-même les concessionnaires locaux, les journaux, les autres clubs. Contactez les mairies pour avoir des salles de réunions (montrez leur cette page du journal pour confirmer votre position, vous nous représentez officiellement dans votre région !)

Renseignements facultatifs

==========

Sujets d'intéret, domaines préférentiels d'utilisation de votre ordinateur de poche :

==========

Je dispose des matériels informatiques suivants (type et modèle, date d'achat) :

==========

J'ai eu connaissance de PPC-T par :

==========

N°PPCUS : N°CHHU :

==========

Observations :

club de Programmation Personnelle
sur Calculatrice de Toulouse
77 rue du Cagire 31100 Toulouse France

Nouvel adhérent
Renouvellement
Changement d'adresse
(rayer la mention
inutile)

PPC-T N°_____

BULLETIN D'ADHESION

NOM _____
Prénom _____
ADRESSE _____
_____
Pays _____
Profession _____
Téléphone: 16( ): : : :

Ecrivez clairement en majuscules ; indiquez l'adresse à laquelle vous souhaitez recevoir les publications.

==========

Je demande à rejoindre PPC-T dans le seul but d'échanger des programmes et des idées sur la programmation. Mes seules obligations sont d'aider les nouveaux venus, d'encourager l'information sur les activités de PPC-T et d'offrir mon expérience aux autres adhérents. J'ai compris que l'adhésion à PPC-T entraine une participation active à l'échange d'informations.

Date signature

==========

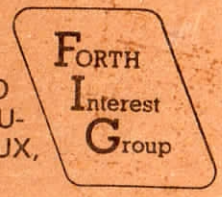
paiement par: Chèque - CCP - mandat - espèces -
à l'ordre de PPC-T

Votre adhésion sera prise en compte à réception par PPC-T du présent bulletin correctement rempli.

==========

Remplir le verso, SVP, merci.

PROGRAMMATION SYNTHETIQUE, LCD, IMPRIMANTE, REGISTRES, MODULE, TOUCHES, OVER LAYS, MATRICES, MEMOIRE CONSTANTE, LECTEUR DE CODES A BARRES, LECTEUR DE CARTES, BASIC, RPN, FORTH, LANGAGE SPECIALISE, MICROCODE, ASSEMBLEUR, MATHEMATIQUES, SOUS PROGRAMME, NOMBRE ALEATOIRE, DRAPEAUX, LUCASIEWICZ, PILE, ALPHANUMERIQUE, PARASITES, RIDEAU, SWAP, NOTATION POLONAISE INVERSE, DUP, SHIFT, ECRAN, MOT, COMPILER, EXECUTION, TESTS, BOUCLES, DO LOOP, BEGIN UNTIL, IF ELSE THEN, : ;, BOUCLES CONDITIONNELLES, LABELS, TABLEAUX, TRACE, SST, BST, R/S



Edité par PPC-T - TITRE: MICRO-REVUE
77, rue du Cagire 31100 TOULOUSE
ISSN 0754-6335
Directeur de Publication : J.-D. Dodin
Imprimé par la Société Pyrénéenne d'Impression S.A.
20 Av. du Lauragais 31400 TOULOUSE