

PPC-T

No6

JUILLET
AOUT
1 9 8 3

P P C

▶ PROGRAMMES ◀

p3: P. Brikké: KA et DEV(iner) . p4 François Laux: Décodage de x; Daniel Georgelin : HP en société . p5 François Laux : Décimal-hexa, décimal-binaire. p6 Jean Boschat : Tone x ; J.F. Pelanne : Intégration (Simpson) . p7 Georgelin Daniel : Entrée alpha numérique ; Jean Fèvre : Vécimal vers programme, les fichiers du X-F, pendu X-F.

▶ APPLICATIONS ◀

41C

p10 Jean Boschat : les assignements, .END.; J-D Dodin : remarques sur l'interface vidéo . p11 Serge Vaudenay : Aléa; Jean Furgerot : Tips and routines, méthodologie, le coin des U-PRO . p13 Robert Schwartz : W", Lbl délocalisés et BG . p 14 J.F. Pelanne : exécuter un programme en X-F . p15 Patrice Nolin : ST0 12 . Didier Cayrac : nouvelle 41C? ; Jean-Claude Caty : les plantages, le livre d'astronomie de Bouiges, les labels . p17 Daniel Georgelin : programmateur ; J. Pividal : ASC .

15C

p18 J-D Dodin/Bruno Piguet : suite du feuilleton-résumé des épisodes précédents

Microcode

p21 Bruno Causse : les deux premiers bits . J-D Dodin : programmer en microcode pourquoi pas ? . p22 Serge Vaudenay : listings ; Jean-Claude Bornes : MLDL p23 CCD ROM PROJECT . p24 J-D Dodin : MLDL, mise en route . p25 J-D Dodin : standard, différence entre MLDL et protocoder, le MLDL Operating System Patrice Nolin : émulateur du CPU sur la 41C ; p27 Marc Chiffolleau : assembleur à partir des mnémoniques .

75C

p28 Janick Taillandier : Forth, Benchmark, Integrate . p29 Eric Angelini : réflexions . p30 Janick Taillandier : langage machine sur 75.

▶ RUBRIQUES ◀

à partir de la page 2

BONNES VACANCES

T187 Brikké Philippe 3 chemin d'Euimont 54690 Lay-St-Christophe, tel (8) 326 88 14 urgent vend interface vidéo 82163B: 100FF (réception sur moniteur ou télé multistandard VHF CCIR). Module HP-IL 400F, le tout de 1982.

T139 Vincent Hercliq 3 rue Berteaux Dumas 92200 Neuilly/seine vend HP 41C 1300 F + 1 module RAM. Vends X-F, X-M, time, 350 F à débattre.

TTI J. BOSCHAM, 4 rue Descartes, 59750 GRANDE-SYNTHE.
RECHERCHE(URGENT):
-SP de Wickes à moins de 100FF
-IMF HF82143A à moins de 1500FF
-HF 41C System Dic. à moins de 100FF
-Lect. de CB à moins de 600FF
Faire offres...

Nouvelles brèves

Les systèmes HP-IL s'agrandissent: après le multimètre électronique et le capteur de données ("data logger"), voici un analyseur logique de table, jusqu'à 43 canaux... On annonce aussi un fréquence mètre et un analyseur de signature...

Quelques chiffres sur Hewlett-Packard en 82: 382 millions de \$ de bénéfices nets, réinvestis à 90%. croissance mondiale 19%, l'informatique fait 52% du chiffre d'affaire du groupe. La France absorbe 12% des commandes. H-P consacre 10% de son chiffre d'affaire en recherche et développement et 8% à l'auto-financement. L'effectif du personnel augmente de 15% par an. cette augmentation est de 16% par an en France. Grace à l'établissement de grenoble H-P est un des plus gros exportateurs de matériel informatique français (à vrai dire ils ne sont pas très nombreux) en exportant 1/3 de son chiffre d'affaire.

70% du chiffre d'affaire de la société est réalisé sur des produits vieux de moins de 5 ans.

Un nouveau siège social de H-P France doit s'ouvrir incessamment à Evry, sur 11 000 m².

Sur le Marché

Le CETIM (52 av Félix Louat 60300 SENLIS) diffuse gratuitement un logiciel pour HP 41C+Quad ou CV sur les règles de calcul du CODAP (appareils à pression): dossier comprenant listing, notice et jeu d'essai. Copie possible sur cartes ou bandes sur place. Demander Mr F. Osweiler.

Mountain computer vient de mettre sur le marché un interface vidéo 80 colonnes HP-IL (NTSC ou PAL) pour 300\$ (adresse dans le N°5).

CompStop vend un MLDL II (fonctionnement analogue au protocoder) vendu avec 8K de RAM + 8K de RAM optionnelle. Pas de ROM. 300\$.

CompStop 3853 N. Oracle Rd. Tucson, Arizona 85705 tel (602) 888-1532.

EDITORIAL

Un numéro très chargé, cette fois-ci. Du coup, peu de place pour le courrier. Ce numéro sera pratiquement terminé avant l'assemblée générale et je ne pourrai donc y faire écho qu'en cas d'évènement exceptionnel.

Un point sur lequel j'aimerais avoir votre avis est la future formule du journal. La cotisation qui passe à 100 F au premier juillet devrait dans notre esprit rester assez longtemps à ce prix. Vous savez que l'augmentation est due pour l'essentiel à l'augmentation du nombre de pages du journal (dont le prix, par voie de conséquence, passe à 20 F avec le présent numéro).

Mais tout dépend de l'exigence de qualité qui est la notre. J'ai devant moi un devis d'un imprimeur "sérieux" qui peut nous sortir le journal au format PPCJ, c'est à dire sous forme de cahier "piqué à cheval" c'est à dire avec deux agrafes au milieu d'une feuille double format repliée, comme un magazine. Pour 500 exemplaires (notre tirage actuel, pour 330 adhérents et 50 ex vendus à la Règle à Calcul) la facture est de 3298 F Hors taxes, bien sûr, soit 3911, pratiquement 4000 F TTC, soit 7,82 par numéro. Compte tenu des frais annexes, le journal reviendrait environ à 10 F. Les frais de port se montent probablement à 6,30 avec le nouveau tarif si on dépasse 100 grammes, soit un total d'environ 17 F de prix de revient.

Ceci nous amènerait à un prix de revient de la cotisation au moins de 150 F, compte tenu des circulaires intermédiaires.

Si nous voulons donner dans le luxe (comme les anglais avec Datafile) et une couverture "magazine", il faut compter sur une cotisation à 180 F avec un journal à 30F.

L'idiot, dans l'affaire, c'est que les 500 exemplaires suivants coûtent 4 fois moins cher! Si nous tirons à 1000 (mais encore faut-il les vendre!) il ne serait pas nécessaire d'augmenter la cotisation. Le problème est que si nous dépassons les 500 adhérents, la gestion deviendra impossible à la main, et même avec la 41 (je n'ai pas encore eu le temps de faire les programmes); et il faudra s'équiper...

PPC-J: dernier numéro reçu Avril 83 (V10 N3) avec quelques nouvelles de nos amis australiens, heureusement car ...

PPCTN: rien depuis le numéro 13

Datafile: un beau numéro (V2N2) triple (MAR/APR/MAY) avec une couverture sur le FORTH, une mise en page particulièrement belle avec une grande variété de titres fait soit au normogramme soit à l'imprimante avec un programme de "Banner". (mes programmes de ce genre sortent des caractères trop gros). La mise en page est aérée avec beaucoup de pages écrites sur une seule colonne (toujours avec réduction, mais sur toute la largeur.

Prisma: reçu le numéro 4 Avril/mai 83, toujours aussi beau.

NOP NOP NOP

POUGEON Olivier (T 62)

à

J.D. DODIN (7226 T 1)

Rennes, le 16/04/83

Cher Mr Dodin,

ERRATUM, à ma lettre du 12.04 concernant le programme ORP.

En utilisant aujourd'hui, je m'aperçois que la version de ORP que je vous ai fournie (cartes, listing et commentaires) n'est pas celle qui fonctionne.

J'ai en effet envoyé et commenté une des premières versions (ça m'apprendra à libeller correctement mes cartes) qui était incomplète et ne fonctionne donc pas.

La partie à modifier se situe entre le LBL 10 et le pas numéroté 50, le listing rectifié est le suivant :

01*LBL 10	15 GTO 11
02 0	16*LBL 12
03*LBL 11	17 ISG 00
04 SF 25	18 GTO 10
05 *LST*	19 *LST*
06 SEEKPTA	20 PURFL
07 FC? 25	21 PCL 01
08 GTO 12	22 STOF LAG
09 GETREC	23 *FINI*
10 SF 25	24 AVIEW
11 XEQ INB 00	25 RTN
12 AVIEW	26*LBL 01
13 !	27 SAVEP
14 +	28 RTN

* * * * * REACTIONS

Un mot pour vous dire ce que je pense du journal (ça sert d'avoir "l'avis de la base" comme dirait l'autre). Remarque que cette lettre est manuscrite car: c'est du bla-bla, "le ruban de ma machine n'est pas neuf" et j'ai la flemme de taper une lettre avec 2 doigts.

Tout d'abord j'ai bien aimé "Au Fond", je ne dirais pas que c'est clair (je mentirai) mais c'est en français (Cocorico!) et ça m'a permis de bien comprendre la programmation synthétique; pour l'instant je nage dans le microde. J'aurais une remarque à faire à propos du nouveau programme KA paru dans le n°5: certes il gagne 35 bytes par rapport au tien (je pense que c'est le tien) mais je n'aime pas du tout la façon de rentrer les données: code de la touche, préfixe, postfixe car pour moi c'est anti-naturel: je pense d'abord à la fonction puis je réfléchis au code la touche où je l'assigne. Certes il est simple d'insérer X<Z X<Y dans le programme pour permettre la rentrée "classique des données mais il y a plus grave/ lorsqu'on se "goure" pour rentrer ces codes ce nouveau programme fait planter la machine (+PACKING infini en général). ON s'en tire avec un Memory Lost mais quand même! L'ancien par contre m'assigne en général XROM 07,52 (=ABS) aux touches déjà assignées: c'est moins grave.

Passons au journal proprement dit: c'est fameux et je ne regrette ni mon inscription ni les anciens numéros que j'ai commandés.

J'aime les articles de "bla-bla" sur la programmation synthétique: ils sont en général bien écrits, avec humour ce qui ne gêne rien (humour "HP" diront certains). Par contre la composition du journal est assez particulière: une chatte n'y retrouverait pas ses petits... Bref c'est assez bord... Je vais être obligé, je pense, à prendre des feutres de couleur différente pour annoter les bulletins: rouge pour la programmation synthétique "haut-vol", orange pour la normale, bleu pour les programmes de maths, vert pour ceux de jeux etc... Bref l'intérêt du lecteur se disperse, se dilue dans ce dédale d'articles. C'est pour cela que les articles que j'apprécie

PROGRAMMES

P. BRIKKE (T157) 21.03.83
16, Allée du Mid
54420 Saulxures les Nancy

Cher Jean-Daniel Dodin,
Pourriez-vous m'envoyer la page 50 de votre livre car mon exemplaire de "Au Fond de la HP-41C" a eu des problèmes de tirage à cette page; Merci d'avance.

Voici 2 programmes : (Avec le X-Functions)
- KA : le classique prgm pour effectuer toutes les assignations, revu et amélioré.

-DEV : il faut deviner un mot avec le moins de coups possible. (Beaucoup plus performant que le "word guessing game" de HP)

```

K A
01 LBL "KA"          41 STO c (145,125)
02 LBL 00           42 "ADDRESS 0:OUI"
03 CF 00           43 AVIEW
04 CF 05           44 LBL 02
05 "KEY 1 ?"      45 GETKEY
06 AVIEW          46 X=O?
07 XEQ 01         47 GTO 02
08 STO 04         48 42
09 "PREPOST 1 ?" 49 X=Y?
10 PROMPT         50 GTO 00
11 STO 05         51 CLST
12 RDN            52 CLA
13 STO 06         53 CLD
14 "KEY 2 ?"     54 RTN
15 AVIEW          55 LBL 01
16 XEQ 01         56 GETKEY
17 STO 01         57 X=O?
18 "PREPOST 2 ?" 58 GTO 01
19 PROMPT         59 31
20 STO 02         60 X=Y?
21 RDN            61 GTO 03
22 STO 03         62 RDN
23 6              63 FS? 00
24 STO 00         64 CHS
25 " " (241,240) 65 "+"
26 LBL 04         66 PASN
27 RCL IND 00     67 ABS
28 XTOA          68 42
29 DSE 00        69 X=Y?
30 GTO 04        70 SF 05
31 RCL M (144,117) 71 CLx
32 RCL c (144,125) 72 43
33 X<>Y         73 X=Y?
34 " " (245,1,105,0, 74 SF 05
    240,240)     75 CLx
35 ASTO c (154,125) 76 44
36 177          77 X=Y?
37 X<>Y         78 SF 05
38 STO IND Y     79 RDN
39 RT            80 STO Y
40 RT            81 E1
  
```

```

82 /            96 +
83 INT          97 ,
84 X<>Y         98 X<>Y
85 E1           99 FS?C 05
86 MOD          100 16
87 E            101 +
88 -            102 RTN
89 16           103 LFL 03
90 *            104 FC?C 00
91 +            105 SF 00
92 ,            106 GTO 01
93 X<>Y         107 END
94 FC?C 00
95 8
  
```

Utilisation:

Clavier	Affichage
XEQ "KA"	KEY 1 ?
Appuyer sur la première touche que vous voulez assigner.	PREPOST 1 ?
Préfixe (décimal)	
ENTER↑	
Postfixe (décimal)	
R/C	KEY 2 ?
Appuyer sur la 2ème touche à assigner.	PREPOST 2 ?
Préfixe (décimal)	
ENTER↑	
Postfixe (décimal)	
R/C	AUTRES? 0:OUI

SI OUI Touche 0 (CHS) KEY 1 ? etc...
SI NON N'importe quel autre touche.

Remarques :
- Les assignations se font par paires uniquement.
- Pour RCL 00 à RCL 15 et STO 00 à STO 15 il faut les assigner sur 2 octets et non un seul. Il faut donc les assigner comme les RCL 16 à 99 ou STO 16 à 99.

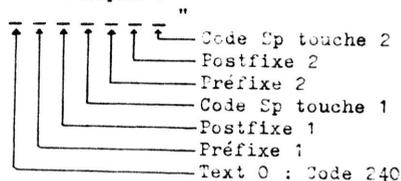
Des instructions courantes :

Préfixe	Postfixe	Fonction
4	0	CAT -
4	2	DEL -
4	3	COFY -
4	4	CLF -
4	5	R/S -
4	6	SIZE - - -
4	7	BST -
4	8	BST -
4	10	PACK
4	15	ASN -
4	19	^ } Q loaders
4	21	X }
4	25	OD }
4	27	Fournie des E
4	29	GTO " (Q-load)
4	30	XEQ " (Q-load)
4	167	eGOBEEP -

- On peut facilement introduire une partie de ce programme dans un autre afin de tout pouvoir assigner par programme :

1* introduire la séquence :
code touche à assigner (1)
" + "
PASN
code touche à assigner (2)
" + "
PASN

2* Placer en Alpha :



3* Introduire dans le prgm la séquence :
Ligne 31 jusqu'à la ligne 41 comprise.

- Le code Spécial est obtenu à partir du code normal:

Code normal = C(+,-) X(ligne) Y(colonne)
Soit A= 1 si C= -
A= 0 si C= +

et B= 1 si code norm = +42 ; +43 ; +44
B= 0 dans les autres cas
Code Spécial= X + ((Y-1+B+(A/2)) * 16)

En fait la plupart des instructions à un octets STO ou RCL peuvent s'assigner directement avec comme préfixe 4, l'affichage est farfelu mais la fonction marche en mode calcul comme en mode RUN. Il y a cependant au moins une exception (voir ailleurs dans ce numéro).

```

----- D E V -----
01 LBL DEV          31 GTO 10
02 LBL 10           32 CLx
03 SF 25           33 91
04 "MT"            34 X=Y?
05 4                35 GTO 10
06 CRFAC           36 95
07 SF 25           37 XTOA
08 CLFL            38 DSE 02
09 CLA             39 GTO 09
10 STO 00          40 AFFRC
11 ACN              41 LBL 12
12 "QUEL MOT ?"   42 E
13 TONE 9 (159, 89) 43 SEEKPT
14 TONE 2 (159, 32) 44 GETREC
15 TONE 9 (159, 89) 45 AVIEW
16 TONE 2 (159, 32) 46 TONE 9 (159, 89)
17 STOP            47 TONE 2 (159, 32)
18 ACFF            48 TONE 9 (159, 89)
19 ALBNG           49 TONE 2 (159, 32)
20 STO 01          50 LBL 08
21 STO 02          51 LBL 00
22 STO 03          52 GETKEY
23 13              53 STO Y
24 X=Y?            54 E1
25 GTO 10          55 /
26 AFFRC           56 X<>Y
27 LBL 09          57 GTO IND Y
  
```

```

62 49
63 GTO 13
64 LBL 03
65 31
66 X=Y?
67 GTO 00
68 CLX
69 35
70 X=Y?
71 GTO 00
72 CLX
73 43
74 GTO 13
75 LBL 04
76 44
77 X=Y?
78 GTO 00
79 CLX
80 37
81 GTO 10
82 LBL 05
83 30
84 GTO 13
85 LBL 06
86 24
87 GTO 13
88 LBL 07
89 73
90 X=Y?
91 GTO 00
92 CLX
93 74
94 X=Y?
95 GTO 00
96 CLX
97 18
98 LBL 13
99 +
100 CLA
101 ATOA
102 ISB CO
103 CLX
104 ,
105 SEEKPT
106 POSFL
107 RCL X
108 INT
109 X=0?
110 GTO 11
111 "----NON-----"
112 AVIEW
113 TONE 8 (159,38)
114 TONE 7 ( " ,37)
115 TONE 8 ( " ,38)
116 TONE 7 ( " ,37)
117 GTO 12

```

```

118 LBL 11
119 MATH **** ONI"
120 AVIEW
121 TONE 8 (159,70)
122 TONE 7 ( " ,57)
123 TONE 8 ( " ,70)
124 TONE 7 ( " ,57)
125 " " (241,95)
126 INDSHR
127 CLX
128 E
129 DELCHR
130 +
131 SEEKPT
132 X<>E
133 CLA
134 ATOA
135 INDSHR
136 E
137 DELCHR
138 LTR CO
139 GTO 12
140 E
141 SEEKPT
142 INTREC
143 AVIEW
144 TONE 2 (73)
145 TONE 2 (72)
146 TONE 7 (87)
147 TONE 3 (73)
148 TONE 2 (72)
149 TONE 7 (87)
150 TONE 3 (73)
151 RCL 01
152 RCL 00
153 /
154 E2
155 *
156 CF 29
157 FIX 0
158 CLA
159 ARCL X
160 FIX 2
161 SF 29
162 "1 DE BON"
163 AVIEW
164 TONE 8 (80)
165 TONE 7 (57)
166 TONE 8 (80)
167 TONE 7 (57)
168 "MT"
169 PURFL
170 CLA
171 CLST
172 STOP
173 GTO 10
174 END

```

et faire R/S entre chaque arrêt de la calculatrice.
 Comment interpréter ces nombres? C'est simple. Tout d'abord il faut les traduire en hexadécimal sur 2 chiffres. Donc 31 16 1 105 30 97 221
 = 1F 10 01 69 1E 61 DD

Rappel de la structure du registre c: en partant de la droite on a:
 Digits 0-1-2 = adresse du .END, permanent.
 Digits 3-4-5 = adresse du 1er registre de données R \emptyset .
 Digits 6-7-8 = constante de départ à chaud de la HP.
 Digits 9-10 = brouillon.
 Digits 11-12-13 = adresse du 1er registre statistique.

Découpons maintenant par tranche les 7 nombres HEXA.

1F	10	01	69	1E	61	DD
----	----	----	----	----	----	----

 Hexa = 1F1 00 169 1E6 1DD
 Déc. = 497 / Constante 486 477
 ou 497 = REG
 486 = R \emptyset
 477 = .END.

C. Q. F. D. I
 Soustrayez donc pour voir 486 de 497 = 11 = adresse du 1er registre statistique comme indiqué dans le manuel d'utilisation page 75.
 Autre exemple: le registre d qui contient les 56 FLAGS (0 à 55)

Un registre est découpé en 7 octets eux-mêmes divisés en 2 digits qui font 4 bits chacun. Donc 4x2x7 = 56
 Le compte est bon!
 Armons le FLAG 2 (SF 02)
 RCL d
 XEQ "DX"
 Sur ma HP j'ai: 32,06,52,4,128,1
 32 en binaire = 100000 soit un nombre composé de 6 chiffres. Il manque 2 chiffres (1 octet=8bits)
 Donc on rajoute 2 zéros devant 100000 = 00100000
 et nous avons le code binaire du 7ème octet composé des 8 premiers drapeaux (0 à 7) et en partant de la gauche en commençant par 0 on a bien le FLAG 2 levé.

Faites SF 05 et SF 07
 RCL d et XEQ "DX" = 37,0,6 etc
 37 décimal = 100101 + 2 zéros = 00100101 et vous avez l'état des 8 premiers drapeaux. Je vous laisse le soin de déchiffrer la suite.... Par curiosité avec SF 02 SF 05 SF 07 avec 0 en X faites X F = 164. Et que font 164 en binaire? Je vous le donne Emile! = 10100100 soit l'état des 8 premiers drapeaux mais à l'envers! Une curiosité mathématique sur les puissances ou l'on a $2^2 + 2^5 + 2^7 = 164$.

J'espère que ces quelques explications auront levé un peu le voile obscur qui recouvre la programmation synthétique pour les nonsinités ...

Amusez-vous bien !

T 207

```

12:24PM 05/30
01*LBL "DX"
02 CLA
03 ADV
04 STO E
05 X<>Y
06 STO L
07 8
08 ENTER+
09 GTO 03
10*LBL 00
11 ALENG
12 X=Y?
13 GTO 02
14 ST- Y
15 CLX
16*LBL 01
17 VIEW X
18 DSE Z
19 DSE Y
20 GTO 01
21 X<> Z
22 GTO 03
23*LBL 02
24 ATOX
25 VIEW X
26 X<> Z
27*LBL 03
28 DSE X
29 GTO 00
30 LASTX
31 CLD
32 .END.

```

Utilisation:
 XEQ"DEV" ; Entrer un mot puis R/S (1 à 12 lettres)
 Appuyer sur une lettre ; Surprise!
 Pour un nouveau jeu après la fin : R/S
 Ce jeu enthousiasme mon entourage , surtout en vitesse rapide .(Sur ma 41CV , 2 vitesses :
 1)Normale ?) Rapide : 2 X plus !!! c'est super !)



Monsieur François LAUX
 Chemin des Fontaines
 FEROLLES
 77580 CRECY la CHAPELLE

Voici un petit programme long de 8 registres inspiré des fiches pratiques de l'Ordinateur Individuel n° 30. Il s'agit de DX (Décodage de X). Il retourne les valeurs décimales des 7 octets contenus dans le registre X. Seule la pile est utilisée, Alpha est perdu, X est conservé en X et LASTX. Mais passons à un exemple en décodant le registre c.
 - Faire SIZE 026
 - CLX
 - RCL c en mode programme ou en pressant la touche prévue à cet effet en mode USER.
 Voir sur une CV ou une C avec QUAD-MEMORY.
 - XEQ"DX"
 Affichage de: 31, 16, 1, 105, 30, 97, 221.
 Si l'affichage défile trop vite, armez le FLAG 21

H.P. en société

La plupart des programmes de jeu proposés pour notre H.P. 41 C se jouent à une seule et unique personne. A croire que les utilisateurs de ces machines sont de vieux ermites craignant la compagnie. Ou encore que seuls, eux, possèdent l'art et la science de les manoeuvrer. Mais peut être est-ce les autres qui hésitent à pénétrer un univers qui leur semble compliqué et sans intérêt : " Qu'est ce

par expérience, je puis vous affirmer qu'un programme de jeu de règle simple ou connue, d'utilisation aisée, le tout appuyé d'une seule personne intéressée...et c'est la ruée ! Voici donc, pour égayer vos soirées en société, un premier programme, de jeu bien sur, qui éveille beaucoup l'intérêt et l'esprit de compétition, qui est compris très vite, mais qui est peut être un peu compliqué d'utilisation (toutes suggestions à ce sujet seront les bienvenues). La règle en est simple : on peut l'expliquer en quelques mots : mot, mélange des lettres, déplacement des lettres bons ou mauvais. Détaillons quand même un peu : un des partis donne un mot secret à H.P. (moins de douze lettres) qui vicieusement en mélange les lettres. Puis elle les restitue à l'autre parti qui, pour retrouver le mot, peut alors tenter des déplacements de lettres dont H.P. affiche le caractère bon ou mauvais. Elle compte tous les coups d'essais et en livre le nombre une fois le mot reconstitué à l'affichage.

Mais parlons un peu du programme : il est long (448 octets soit 4 pistes de 2 cartes magnétiques) mais relativement simple à comprendre car il ne possède pas d'astuces super-géniales. Il n'utilise pas de sous-programmes externes et nécessite 19 registres mémoires. Voici l'explication de ses labels : RLM comme Retrouver Le Mot, VR comme VÉRIFICATION des données entrées, ERR comme ERREUR, REC comme REConstitution et enfin FIN comme FIN (qui l'eut cru).

Donc pour rentrer ce jeu en machine il vaut mieux posséder le lecteur de cartes magnétiques. Mais attention pour y jouer, il vaut mieux l'enlever. J'ai en effet remarqué que cet accessoire intriguait beaucoup les incultes, et qu'ils finissaient toujours par trouver le moyen de l'ôter (la machine toujours allumée bien sur). Ensuite c'est relativement simple, faite : XEQ'RLM (ou RTN si vous venez d'y jouer), H.P. vous demande le "MOT SECRET ?", elle est déjà en mode alphanumérique, vous n'avez qu'à introduire ce mot mystérieux et à relancer : R/S. Donnez alors la machine à vos adversaires. Elle les fait patienter et finit par afficher, sous leur regard perplexe, les lettres mélangées du mot. Ici trois cas se présentent :

- s'ils se demandent, par exemple, si le S qui la 2^e lettre du mot mélangé ne serait pas la dernière et 7^e lettre du mot à retrouver ?

- ils peuvent alors faire : R/S, H.P. demande "DE O=???" ils introduisent alors l'ordre (la position) de la lettre qu'ils veulent déplacer, ici 2, et ils relancent.

- mais ils peuvent aussi donner directement cette valeur et relancer.

- ou alors ils introduisent une valeur inadéquate et relancent.

Dans ce dernier cas la machine leur rappelle le nombre de lettres du mot et l'affiche à nouveau. Sinon elle leur demande "A O=???". Ce à quoi ils doivent répondre par l'ordre (la position) supposé de la lettre choisie précédemment. Mais là encore si cet ordre est inadéquat elle leur rappelle le nombre de lettres du mot et l'affiche à nouveau : il faut alors recommencer. Sinon elle réfléchit et finit par annoncer "C EST BIEN" ou "C EST MAL". Si c'est bien c'est que la première lettre choisie dans le mot mélangé (ici S) est bien dans la position donnée en seconde (?). Le S serait donc bien la dernière lettre. Et si c'est mal c'est que cette lettre serait mal placée, le S ne serait donc pas la 7^e lettre du mot mystérieux. Attention on ne peut rien dire d'autre et si vous avez deux S dans le mot la machine ne fera pas de distinction entre l'un et l'autre.

H.P. effectue alors l'échange entre les deux lettres désignées par leur position et :

- soit réaffiche le mot ainsi trituré
- soit, si celui-ci est le mystère, et a donc été découvert, annonce le nombre de coups d'essais qu'ils leur a fallu pour le percer.

Un dernier conseil avant de vous livrer le programme : expliquez bien à vos amis l'utilisation de la touche R/S car ils oublient souvent de relancer, alors que l'introduction des données leur est facile. On pourrait bien supprimer cet ennui par une boucle centrée sur une pause, mais le temps de réponse est alors long et ils ne comprennent pas pourquoi la machine met tant de temps à répondre.

- Et pour finir voici le commentaire ligne par ligne :
- 1 à 8 : demande et stockage du mot secret
 - 9 à 22 : stockage du mot lettre par lettre
 - 23 à 32 : vérification du registre aléatoire
 - 33 à 43 : initialisation des compteurs de mélange
 - 44 à 64 : mélange aléatoire des lettres
 - 65 à 66 : initialisation du compteur de coups
 - 67 à 77 : test d'identité entre l'affichage du joueur et le mot secret
 - 78 à 88 : demande de déplacement
 - 89 à 109 : réponse au déplacement testé : bien ou mal
 - 110 à 112 : incrémentation du compteur de coups et fin de boucle
 - 113 à 124 : vérification des données de déplacement
 - 125 à 139 : message d'erreur
 - 140 à 154 : reconstitution de l'affichage
 - 155 à 198 : isolation du X^{ieme} caractère du mot à trouver
 - 199 à 208 : affichage du résultat final.

01 *LBL *RIM *	53 1	105 *BIEN*	157 6
02 *MOT SECRET ?	54 +	106 X=Y?	158 RCL 03
03 AOK	55 *	107 *MAL*	159 INT
04 STOP	56 6	108 AVIEW	160 X=Y?
05 AOFF	57 +	109 TONE 5	161 -
06 ASTO 00	58 INT	110 ISG 02	162 AES
07 ASHF	59 RCL IND 1	111 CLA	163 7
08 ASTO 01	60 X<> IND 2	112 GTO 04	164 X<>Y
09 SF 25	61 STO IND Y		165 -
10 *ATTENDEZ*	62 RCL Z	113*LBL *VR*	
11 AVIEW	63 ISG Y	114 INT	166*LBL 07
12 SF 99	64 GTO 02	115 ABS	167 *+*
13 6.1	65 .	116 X=0?	168 DSE X
14 STO 02	66 STO 02	117 GTO *ERR*	169 GTO 07
15 1.012		118 RCL 05	170 RT
16 STO 03	67*LBL 04	119 1	171 6
	68 XEQ *REC*	120 +	172 RCL 03
17*LBL 00	69 RCL 00	121 X<Y?	173 INT
18 XEQ *IS*	70 ASTO Y	122 GTO *ERR*	174 X<Y?
19 ASTO IND 02	71 X=Y?	123 RDN	175 GTO 08
20 ISG 02	72 GTO 05	124 RTN	176 ASTO X
21 ISG 03	73 ASHF		177 ARCL 01
22 GTO 00	74 RCL 01	125*LBL *ERR*	178 ASTO Y
	75 ASTO Y	126 *IL N A QUE*	179 X=Y?
23*LBL 01	76 X=Y?	127 AVIEW	180 GTO 01
24 RCL 04	77 GTO *FIN*	128 TONE 7	181 GTO 09
25 SF 25		129 FIX 0	
26 ABS	78*LBL 05	130 CLA	182*LBL 08
27 FSQ 25	79 CLST	131 RCL 05	183 ARCL 00
28 GTO 02	80 CF 22	132 1	
29 *N=?*	81 *DE O=??*	133 +	184*LBL 09
30 PROMPT	82 BEEP	134 ARCL X	185 ASHF
	83 STOP	135 *+ LETTRES*	186 ASTO X
31*LBL 02	84 FC? 22	136 AVIEW	187 -
32 STO 04	85 PROMPT	137 TONE 9	188 ARCL X
33 RCL 03	86 XEQ *VR*	138 TONE 9	189 ASTO X
34 INT	87 *A O=??*	139 GTO 04	190 CLA
35 2	88 PROMPT		191 ASTO Y
36 -	89 *EH BIEN*	140*LBL *REC*	192 * -
37 STO 05	90 AVIEW	141 RCL 05	193 ARCL X
38 6	91 XEQ *VR*	142 6	194 ASHF
39 +	92 STO 03	143 +	195 ASTO X
40 1 E3	93 5	144 1 E3	196 X=Y?
41 /	94 ST+ Z	145 /	197 GTO 01
42 6	95 +	146 6	198 RTN
43 +	96 RCL IND X	147 +	
	97 X<> IND Z	148 CLA	199*LBL *FIN*
44*LBL 03	98 STO IND Y		200 *GAGNE*
45 RCL 04	99 ENTER+	149*LBL 06	201 AVIEW
46 9821	100 ENTER+	150 ARCL IND 1	202 BEEP
47 *	101 XEQ *IS*	151 AVIEW	203 *EN *
48 .211327	102 RCL Z	152 ISG X	204 FIX 0
49 +	103 *C EST *	153 GTO 06	205 ARCL 02
50 FRC	104 X=Y?	154 RTN	206 *+ FOIS*
51 STO 04		155*LBL *IS*	207 PROMPT
52 RCL 05		156 CLA	208 .END.



Monsieur François LAUX
Chemin des Fontaines
FEROLLES
77580 CRECY la CHAPELLE

Cher Jean Daniel,

Pour commencer, toutes mes excuses ! KA fonction ne à merveille à condition de ne pas se tromper en le recopiant! (PPC-T n°4 page 21).

Voici quelques routines qui peuvent faciliter la programmation synthétique: il s'agit de changements de bases.

Voyons les une par une.

D - H Décimal - Hexadécimal

D en X mais inférieur à 1 E 10 à cause de la précision des calculs.

Exemple: 1234567890

XEQ "D - H"

résultat en ALPHA = 499602D2

D - B Décimal - Binaire

D en X inférieur à 16777216 ce qui correspond à 24 1 capacité de ALPHA

Exemple: 1234

XEQ "D - B"

résultat en ALPHA = 10011010010

H - D Hexadécimal - Décimal

introduire le nombre hexa en ALPHA

Exemple: ALPHA ABC ALPHA

XEQ H - D

résultat en X = 2748

B -D Binaire - Décimal

entrer le nombre binaire en ALPHA

Exemple: ALPHA 101010 ALPHA

XEQ B - D

résultat en X = 42

Tous ces programmes n'utilisent pas de registres de données. La pile est perdue. Total environ 30 registres, donc sur une seule carte recto verso

ajoute à droite de M l'octet correspondant.
Ligne 25:ajoute à droite de M les octets 9I et 7C (la ligne est donc: F37F917C)
Lignes 26 et 27:place le pointeur de prgm en M (adresse 06) pour l'exécution du tone et le retour sur le end de TX grâce au STO b.

Remarque:on peut"SST" le programme,mais on ne peut introduire d'instructions en prgm si le pointeur se trouve en M(MEMORY LOST)
La HP affiche les tones de paramètre supérieur à 9 en ne faisant apparaître que le chiffre de l'unité.
Par exemple,TONE(26) sera affiché TONE 6.

Bonne programmation
T 207

01*LBL "D-H" 31 XEQ 02 61 X<>Y 91 ENTER+
02 RCLFLAG 32 -1 62 RTN 92 RTN
03 X<>Y 33 ARDT
04 I E10 34 RDN 63*LBL "H-D" 93*LBL "B-D"
05 CF 07 35 RDN 64 CLST 94 CLST
06 GTO 00 36 INT
37 X*0? 65*LBL 03 95*LBL 05
07*LBL "D-E" 38 GTO 01 66 ALENG 96 ALENG
08 RCLFLAG 39 X<>Y 67 1 97 1
09 X<>Y 40 STOFLAG 68 - 98 -
10 16777216 41 CLST 69 X<>? 99 X<>?
11 SF 07 42 AVIEW 70 GTO 06 100 GTO 06
43 RTN 71 ATOX 101 ATOX
12*LBL 00 44 / 72 40 102 RDN
13 X<=Y? 73 - 103 XEQ IND T
14 FACT 45*LBL 02 74 9 104 X<>Y
15 RDN 46 10 75 X<>Y? 105 2
16 FIX 0 47 X<=Y? 76 XEQ 04 106 X<>Y
17 CF 29 48 GTO 02 77 RDN 107 Y+X
18 CLA 49 ARCL Y 78 16 108 *
50 RDN 79 X<>Y 109 +
19*LBL 01 51 RTN 80 X<> Z 110 ENTER+
20 ENTER+ 81 Y+X 111 GTO 05
21 FS? 07 52*LBL 02 82 *
22 2 53 X<> T 83 +
23 FC? 07 54 X<>Y 84 ENTER+
24 16 55 STO L 85 ENTER+
25 ST/ Z 56 55 86 GTO 03
26 MOD 57 ST+ L
27 INT 58 X<> L 87*LBL 04
28 FS? 07 59 XTOR 88 RDN
29 ARCL X 60 X<> T 89 7
30 FC? 07 90 - 118*LBL 06
119 RDN
120 STO L
121 CLST
122 LASTX
123 .END.

* * * * *

Jean BOSCHAT
4,rue DESCARTES,
59760 Grande-Synthe.
T7I

Grde-Synthe le 30/4/83

Monsieur,

Je vous envoie le programme TX qui permet,en utilisant la programmation synthétique,d'entendre tous les "tones" dont le paramètre est spécifié en X.(paramètre compris entre 0 et 127).

Commentaire:

Ce programme utilise une astuce qui,à l'aide du registre b,fait lire par la HP le registre M comme un registre contenant des lignes de programme! L'essentiel du programme sert à remplir le registre M comme suit:

octet 1: 9F code de l'instruction TONE
octet 2: nombre que vous avez introduit et qui sera pris comme paramètre du TONE.

octet3: 9I code de l'instruction STO.
octet4: 7C code de l'instruction MEAN.
En fait,les trois premiers octetsà gauche de M sont nuls,et sont négligés lors de l'exécution.

Reg M: 00 00 00 9F XX 9I 7C

Ce registre équivaut donc aux lignes: TONE(XX)
STO b

Ligne2:création de 9F
(la ligne est:FI 9F)

Lignes 03 à24:vérifie que X inférieur à 128 et

Heureuse programmation...

J.Boschat.

01*LBL "TX" 15 FS?C 07
02 "" 16 SF 09
03 128 17 X<> d
04 MOD 18 X<> [
05 LASTX 19 "H*"
06 + 20 STO \
07 OCT 21 "H*"
08 X<> d 22 X<> \
09 FS?C 11 23 CLA
10 SF 12 24 STO [
11 FS?C 10 25 "H*"
12 SF 11 26 E6
13 FS?C 09 27 X<> b
14 SF 10 28 .END.

* * * * *

Je vous propose un prgm d'intégration très rapide et très précis. (3 fois + rapide que celui de Mr Leroy Il existe de très nombreuses façons d'intégrer : par Gauss, Simpson, les trapèzes, Romberg etc ... mais quelque soit celle ci elle doit être rapide, précise mais aussi utiliser le moins d'octets possibles, c'est pourquoi je vous propose un prgm d'intégration par Simpson qui me semble la plus facile à mettre en oeuvre De plus celui ci est une version très optimisée du prgm du module MATH (qui à mon avis ne propose pas des programmes très performants c'est dommage) en effet mon prgm n'utilise que trois registres de données au lieu de 8 et de plus utilise les registres d'état M,N et O.

Ci joint une carte magnétique.

Commentaire du prgm :

Si vous ne disposez pas du XFTes, supprimer les lignes 02 à 06 (mais faire SIZE 003).

Les registres 01 et 02 contiennent les bornes d'intégration a et b. Le registre 00 contient f(a)+f(b).

La ligne 0 24 contient le nombre d'intégration. Celui ci doit être pair.

Pour utiliser ce prgm c'est très simple :

1/ Programmer votre f(x) à étudier en commençant par LBL'FF ou autre (mais alors modifier la ligne 0 12 puis XEQ'IN.

2/ a enter b

3/ XEQ a (ou R/S) et automatiquement le prgm vous donnera 1 à 2 minutes après le résultat sous la forme I = n n n,a a a a a a

Voilà c'est tout

Happy programming

J.P. PELANNE

T 191

01*LBL "IH" 16 STO 00 31 ST+ 00 46 +
02 SIZE? 17 RCL 02 32 2 47 ST+ 00
03 J 18 XEQ IND [33 ST+ \ 48 RTN
04 X<>Y? 19 ST+ 00 34 60
05 PSIZE 20 RCL 02 35 RCL \ 49*LBL 00
06 CLX 21 RCL 01 36 X=Y? 50 J
07 STOP 22 STO J 37 GTO 00 51 RCL 00
23 - 38 XEQ 06 52 RCL 01
08*LBL a 24 60 39 GTO 03 53 *
09 STO 02 25 / 54 X<>Y 55 /
10 X<>Y 26 STO 01 40*LBL 06 56 "I="
11 STO 01 27 , 41 RCL 01 57 ARCL X
12 "FF" 28 STO \ 42 ST+ J 58 PROMPT
13 ASTO [29*LBL 03 43 RCL J 59 CLST
14 RCL 01 30 XEQ 06 44 XEQ IND [60 .END.
15 XEQ IND [

* * * * *

Entrée Alphanumérique

L'idée et le programme de base de cet article proviennent de l'excellent livre de W. C. WICKES : Synthetic Programming on the IBM 41 C écrit hélas en américain. Ce cher William a pensé d'utiliser le drapeau n° 40 pour rentrer dans le registre alphanumérique une réponse qui, à l'affichage, prolonge le texte de la question mais qui s'inscrit séparément dans ce registre. En effet le 40° drapeau (le n° 45) est levé lorsque l'on introduit manuellement une séquence de données, sinon il est abaissé. On peut ainsi considérer qu'il est levé à chaque fois que l'on voit le caractère souligné () à la fin d'une séquence (exemples de séquences : STO , 1983,12 , DR ou encore SIZE).

Mais attention ! Il ne suffit pas de lever ce drapeau pour voir apparaître le souligné. Par contre, dans cet état, il permet de continuer manuellement l'entrée du reste de la séquence. Mais pour peu que l'on ruse un peu, on peut faire afficher la question suivie de la réponse, si elle est alphanumérique, que l'on entre et n'écrire secrètement, dans le registre alpha, que la réponse seule. Tout simplement on utilise le registre d'affichage. C'est ce qu'a fait William.

Voici un exemple : XEQ' n'importe quoi et la machine répond : DONNEE= , on est en mode alpha et on a la surprise de constater que si l'on tape E on obtient : DONNEE=E mais si l'on sort et reentre du mode alpha on lit E tout simplement. Cela permet de nous rappeler toujours de la question quand on écrit la réponse.

Programme ô combien génial mais auquel je trouvais un petit défaut : la question devrait pouvoir posséder un point d'interrogation final ou un caractère souligné qui disparaîtrait seul lors de l'introduction de la réponse. Et après maintes recherches je puis vous affirmer que c'est possible ! D'ailleurs la preuve à suivre.

Ah ! Une petite restriction quand même : la question ne peut être composée que de 6 caractères, point d'interrogation inclus, inconvénient que n'a pas la version d'origine.

Les trois programmes dont les listes suivent sont :
- EAO : pour le programme d'Entrée Alphanumérique d'Origine de W.C. WICKES .

- EA : pour le programme d'Entrée Alphanumérique type que l'on insère dans le programme où l'on pose la question.

- SEA : pour le Sous-programme d'Entrée Alphanumérique que l'on utilise ainsi : placer en alpha la question à poser sans son point d'interrogation (moins de 5 caractères) et faire XEQ'SEA , au retour la réponse est dans le registre alpha.

Remarque : ces programmes détruisent les registres T, L (sauf pour EAO) et alpha, mais ils conservent le registre d et donc tous les drapeaux.

Programmation : seule difficulté : la chaîne alpha qui est ainsi codée : 242 4 128 en décimal et F2 04 80 en hexadécimal.

GEORGELIN Daniel T 57

P.S. : SEA détruit aussi le registre Z.

	11*LBL "EAO"	25*LBL "SEA"
	12*LBL "EA"	26 ASTO L
	13 "A=?"	27 "I=?"
01*LBL "EAO"	14 ASTO L	28 ASTO X
02 "a"	15 "a"	29 "a"
03 RCL E	16 RCL E	30 RCL E
04 X(> d	17 X(> d	31 X(> d
05 "DONNEE="	18 "A="	32 CLA
06 AVIEW	19 AVIEW	33 ARCL L
07 CLA	20 VIEW L	34 AVIEW
08 STOP	21 CLA	35 VIEW Y
09 X(> d	22 STOP	36 CLA
10 RTH	23 X(> d	37 STOP
	24 RTH	38 X(> d
		39 .END.

Jean (T250) FEVRE
100, Rue CLAUDE DECAEN
75012 PARIS

Jean daniel (TI) DODIN
77, Rue CAGIÈRE
31100 TOULOUSE

DP - Decimal to Program

PARIS 1e, 9 mai 1983

Cher Jean Daniel,

Cette lettre est divisée en trois parties:

Un rappel sur la structure des fichiers de l'extension mémoire du module X FONCTIONS.

Un programme 'DP' (Décimal to Program), qui fait la même chose que 'LB' de Lionel ANCELOT (cf PPC-TN°1), mais il est plus souple d'emploi.

Un programme 'PX' qui permet de jouer au pendu sur sa 41C.

PPC-T N°6
JUILLET AOÛT
1983 P7

-1- Introduction.

Le manuel d'utilisation du module nous dit que les deux premiers registres d'un fichier forment l'en-tête. Le premier registre d'en-tête contient le nom du fichier, complété à 7 caractères par des espaces. Le deuxième registre contient divers paramètres. C'est ce registre que nous allons étudier pour chaque type de fichier.

-2- Fichiers Programme.

Le second registre d'en-tête d'un fichier programme à la forme suivante:

10 00 00 00 BB BR RR

RRR : ce sont les digits 0,1 et 2 ils donnent la taille du programme en registres

BBB : les digits 3,4 et 5 donnent la taille du programme en octets (bytes).

I : le digit 13 indique le type du fichier, donc à I un fichier programme.

Après ce second registre on trouve le programme, sous sa forme habituelle (ce qui permet d'exécuter des programmes dans l'EM). Dans le dernier registre du programme à la suite du dernier octet du programme on trouve un octet de contrôle. Cet octet est la somme de tous les octets du programme modulo 256. Chaque fois que l'on rappelle un programme de l'EM vers la mémoire la HP contrôle ce dernier octet et si il n'y a pas correspondance on a: CHKSUM ERRL.

Dans BBB on ne tient pas compte de l'octet de contrôle, mais dans RRR on le prend en compte, donc:

RRR = INT((BBB+7)/7)

-3- Fichiers de données.

Le second registre d'en-tête d'un fichier de données à la forme:

2A AA 00 00 PP PR RR

RRR : taille du fichier en registres.

PPP : valeur actuelle du pointeur de registres.

AAA : adresse absolue du deuxième registre d'en-tête à zéro si le fichier est vide (à la création).

2 : type du fichier, 2 pour fichier de données.

Il n'y a rien d'autre à rajouter si ce n'est que les registres de données ont la même forme que les registres "normaux".

-4- Fichier ASCII.

Le second registre d'en-tête d'un fichier ASCII est de la forme:

3A AA 00 CC EE ER RR

RRR : taille du fichier en registres.

EEE : valeur actuelle du pointeur d'enregistrements.

CC : valeur actuelle du pointeur de caractères.

AAA : adresse absolue du deuxième registre d'en-tête à zéro si le fichier est vide.

3 : type du fichier, 3 pour les fichiers ASCII.

Ensuite chaque enregistrement est précédé d'un octet qui indique la taille de l'enregistrement (entre 0 et FE). Enfin le fichier se termine par un octet codé FF.

-5- Conclusion.

Enfin, la dernière chose à rajouter est que le dernier fichier dans l'EM est suivi par un registre donc tous les octets sont à la valeur FF.

-1- Fonction.

Ce programme a pour objet de rentrer des programmes (si on en a le courage!!), ou seulement des instructions octet par octet. Chaque octet étant entré en décimal.

Pour faire ces opérations, 'DP' utilise le module X FONCTIONS et son extension mémoire.

'DP' sera surtout utile lors de l'introduction de programmes avec un fort pourcentage de fonctions ou de chaînes synthétiques.

Ce programme a une mise en oeuvre plus simple que 'LB' de Lionel ANCELOT. On n'a pas besoin de savoir le nombre d'octets à introduire, et il n'utilise aucun autre programme en complément.

2- Utilisation.

01,02,...,07 représentent les octets entrés.

CLAVIER	AFFICHAGE
XEQ 'DP'	B0=
o0 R/S	B1=
o1 R/S	B2=
.	.
oN R/S	BN+1=
R/S	.

A ce moment le dernier programme présent en mémoire est celui que l'on vient de rentrer octet par octet.

-3- Observations.

- 'DP' utilise le module XF.
- Tous les fichiers de l'EM sont perdus. A la fin du programme le directory est vide.
- Tous les flags conservent leur état initial sauf le flag 01 qui est baissé.

-4- Pile.

Au cours du programme les registres X,Y,Z,T et L sont perdus.

Etat final de la pile:

T= contenu du registre OBE
 Z= valeur de c mettant R00 = (OBE)
 Y= 0
 X= valeur actuelle de c
 L= X

-5- Registres.

On utilise aucun registre. on travail en size 000.

-6- Flags.

- 01 pour entrer le END
- 22 teste l'entrée d'un octet et permet de sortir de la boucle d'introduction.
- 30 flag du catalogue, toujours à l'état bas, sert à créer la négation de la fonction ISG.

-7- Remarque.

Le registre 'a' de la pile système sert à stocker le contenu initial du registre des flags 'd'.

-8- Description du programme.

Ligne 02 : initialisation de l'EM.

Lignes 03 à 05 : création d'un fichier données de 122 registres, de nom 'x'.

Lignes 06 à 07 : mise à zero de la pile et du registre L qui sert de compteur d'octets.

Lignes 08 à 12 : sauvegarde du registre 'd' des flags dans le registre 'a'. Ce registre reste intact tant qu'il n'y a pas d'appel à un sous-programme. Ensuite on formate l'affichage et on baisse le flag 01, qui levé transforme la boucle d'entrée en sous-programme.

Lignes 13 à 42 : boucle d'entrée des octets, on rajoute chaque octet dans le registre alpha et tous les sept octets on a ainsi un registre que l'on sauve dans le fichier (ligne 41). On fait la somme de tous les octets (ligne 33) et le compte de ceux-ci (lignes 34 à 36).

On quitte la boucle d'entrée lorsqu'il n'y a plus d'entrée numérique (lignes 26 et 27).

Lignes 43 à 47 : on remet l'affichage dans son format initial (lignes 45 et 46). Puis on lève le flag 01 afin de transformer le boucle d'entrée en sous-programme.

Lignes 48 à 55 : on rentre un END à la fin du programme (octets 192,0 et 9).

Lignes 56 à 60 : réduction à huit bits de la somme des octets.

Lignes 61 à 70 : on complète le dernier registre à sept octets par des nuls puis on range ce registre dans le fichier.

Ligne 71 : les sept caractères de gauche forment le contenu d'un registre 'c' qui place le rideau en OBE. Les quatre caractères suivant forment le debut du deuxième registre d'entete d'un fichier programme.

Lignes 72 à 77 : à partir du nombre d'octets entres on calcul la taille du programme en registres.

Lignes 78 à 95 : à partir de la taille en octets et en registres du fichier on calcul les trois derniers octets du second registre d'en-tete du fichier programme contenu dans le registre alpha.

Lignes 96 à 101 : on place le rideau en OBE puis on range à cette adresse le second registre d'en-tete que l'on vient de calculer.

Lignes 102 à 103 : on rappelle le fichier en memoire programme.

Lignes 104 à 112 : sous-programme qui initialise l'extension mémoire en placent zéro à l'adresse 040 qui est le registre système de l'EM du module X-F.

01xLBL'DP'	39 GTO 01	77 INT
02 XEQ 06	40 X<> M	78 X<>Y
03 122	41 SAVEX	79 RCL X
04 'x'	42 GTO 00	80 16
05 CRFLD	43xLBL 03	81 /
06 CLST	44 STO M	82 XTOA
07 STO L	45 X<> a	83 X<> L
08 RCL d	46 STO d	84 MOD
09 STO a	47 SF 01	85 LASTX
10 FIX 0	48 CLX	86 x
11 CF 29	49 I92	87 RCL Y
12 CF 01	50 XEQ 02	88 256
13xLBL 00	51 .	89 /
14 CLA	52 XEQ 02	90 +
15 CLX	53 9	91 XTOA
16 6 E-3	54 XEQ 02	92 RDN
17xLBL 01	55 CF 01	93 256
18 FS? 01	56 LASTX	94 MOD
19 RTN	57 RCL Z	95 XTOA
20 RCL M	58 256	96 RCL N
21 'B'	59 MOD	97 X<> c
22 ARCL L	60 XTOA	98 RCL M
23 '+='	61 X<> Z	99 STO 00
24 CF 22	62xLBL 04	100 X<>Y
25 PROMPT	63 ISG X	101 STO c
26 FC? 22	64 FS? 30	102 'x'
27 GTO 03	65 GTO 05	103 GETSUB
28 X<>Y	66 ' -'	104xLBL 06
29 STO M	67 GTO 04	105 ' * * * * '
30 RDN	68xLBL 05	106 RCL M
31xLBL 02	69 RCL M	107 X<> c
32 XTOA	70 SAVEX	108 SIGN
33 ST+ Z	71 ' * * * * * '	109 STO 60
34 CLX	72 X<> Z	110 LASTX
35 E	73 RCL X	111 STO c
36 ST+ L	74 7	112 END
37 RDN	75 ST+ Y	
38 ISG X	76 /	

Ligne 71 : FB 10 00 01 69 0B E0 A0 10 00 00 00

Ligne 66 : F2 7F 00

Ligne 105: F7 10 00 01 69 00 40 02

PX - Pendu avec X fonctions

-1- Fonction.

Ce programme est une version pour HP-41C du jeu bien connu du pendu.

La version présentée ici est la cinquième que je fais (la seconde avec le module XF). Le fait d'utiliser le module XF en plus de racourcir le programme permet d'utiliser un fichier de mots.

-2- Utilisation.

CLAVIER	AFFICHAGE
---------	-----------

XEQ 'PX'
 a- soit on rentre un mot (10 lettres max), R/S

APPLICATIONS

Jean BOSCHET,
4, rue Descartes,
59200, GRAND-BYNTHE.
771

Grde-Synthe
le 10/4/83

Cher Monsieur,

Après vous avoir envoyé la version "écourtée" du programme KA, j'ai voulu comprendre pourquoi ce programme pose des problèmes lorsqu'on a supprimé un ou plusieurs assignements. En effet, mes seules connaissances sur la mémoire d'assignation (que j'avais mises dans le commentaire) proviennent de AF, et ne permettent pas d'expliquer ce problème...

J'ai alors entrepris l'étude de la mémoire d'assignation en utilisant le programme suivant:
OI LBL TEST La ligne 02 est: F5 01 69 00 FO FO
(comme dans KA)
RCL c L'utilisation en est simple: I77 puis
ASTO c R/S nous donnera en M le registre
X<>Y OCO que l'on pourra visualiser en
RCL IND X mode alpha. I78, R/S nous donnera de
RCL Z la même façon le registre supérieur
STO c OCI et ainsi de suite...
RDN Malgré la normalisation, on peut
IO STO M ainsi comprendre le fonctionnement
de cette partie de la mémoire, mais

les assignements sont transformés en XROM 03,46 qui correspond à I/X.

Après chaque essai, il faut remettre la mémoire à zéro par un MEMORY LOST.

Finalement, je suis parvenu aux conclusions suivantes:

1) La mémoire d'assignation commence bien à partir de OCO vers les adresses supérieures.

2) Les registres d'assignement ont bien le format:

FO I 2 3 4 5 6

let2: code de fonction sur deux octets. l'octet I est rempli par un octet "bouche trou" 04 (qui correspond à LBI 03) si la fonction n'occupe qu'un octet.

3: code touche sur un octet.

Les octets 4,5 et 6 jouent des rôles analogues pour une seconde assignation.

3) Après un memory lost, si l'on assigne sans jamais effacer d'assignement, la HF remplit les registres de droite à gauche. Lors d'une assignation, elle complète la partie gauche de OCO si celle-ci est vide, sinon elle décale tous les registres d'assignation vers le haut et remplit la droite de OCO.

Si l'on effectue ASN + II après un MEMORY LOST, et donc si cette assignation est la seule en mémoire, le registre OCO est:

FO 00 00 00 04 40 01

4) Lorsqu'on efface une assignation, la HF baisse les index des registres + ou e, mais elle se contente d'annuler le code touche dans le registre d'assignement!!

5) Dans le cas général, lors d'une assignation, la HF commence par rechercher dans TOUTE la mémoire d'assignement s'il y a un code touche à zéro dans un demi-registre. Dès qu'elle en trouve un elle y place l'assignement, sinon elle décale tous les registres et le place à droite de OCO.

La recherche s'effectue dans l'ordre suivant:

partie droite de OCO	oci	6	5
partie gauche de OCO			
partie droite de OCI	oci	4	3
partie gauche de OCI	oco	2	1

et ainsi de suite..... Ceci explique pourquoi la HF met assez longtemps pour effectuer un assignement si il y en a déjà beaucoup en mémoire.

Si on a effacé un ou deux assignements, le programme KA risque donc de ne pas fonctionner; en effet:

si on a supprimé un assignement:

1^{er} cas: si le registre OCO est totalement rempli lors du message ASSIGNEZ 2 T, la première assignation occupera les 3 octets de celle qui a été supprimée, la seconde sera dans OCO après décalage des registres: le programme marchera.

2^{em} cas: si seule la partie droite de OCO était remplie, les deux touches sont bien assignées mais l'assignement qui se trouvait à droite de OCO est perturbé.

si on a supprimé 2 assignements:

les 2 nouveaux assignements sont effectués, mais deux autres sont perturbés.

Lorsqu'on reassigne une touche déjà assignée, la HF ne remplace pas le demi-registre occupé par l'ancien assignement de cette touche. Elle annule l'assignation précédente (elle met l'octet du code touche à zéro), puis effectue la seconde assignation comme décrit au 5).
La séquence: ASN + II / ASN - II a le même effet que: ASN + II / AST Alpha Alpha II / ASN - II
Cette règle reste valable même si l'on assigne à une touche la fonction qui y était déjà assignée!!
7) Lorsque l'on presse une touche en mode USER, la HF effectue les opérations suivantes:

- elle regarde si l'index correspondant est levé
- si non, elle effectue la fonction qu'on aurait obtenu en mode normal.
- si l'index est levé:

La HF recherche dans la mémoire d'assignation dans le même ordre qu'au 5) si une fonction est assignée.

si oui, elle l'exécute, si non elle recherche si un programme en mémoire vive est assigné à la touche.

A ce stade, la machine doit forcément trouver quelque chose, sinon c'est que l'index a été armé de façon artificielle (en synthétique...) et elle se "plante" lamentablement!!

8) Les assignements de programme n'utilisent pas de place en mémoire car ils ne sont pas stockés dans la mémoire d'assignement, le code touche étant enregistré dans le 4^{em} octet du label global.

Les assignations de programmes en ROM constituent une exception. Le label ne pouvant être modifié, ils sont enregistrés dans la mémoire d'assignement comme une fonction standard.

9) Un PACKING a pour effet de supprimer dans la mémoire d'assignation les registres où les deux octets de touche sont à zéro. Les registres où un seul code touche est à zéro ne sont pas modifiés!

10) Lors de l'enregistrement d'une carte d'état (par WSTS), le contenu de la mémoire d'assignement est recopié tel quel, sans PACKING préalable.

OUI! J'avais envisagé de taper les exemples qui m'ont permis de parvenir à ces nombreuses conclusions, mais il y en aurait encore pour 2 pages.....

Pour finir, voici un court programme utilitaire que certains ne connaissent peut-être pas encore. "en" place le pointeur de programme au niveau du .END. permanent, ce qui permet d'accéder facilement au dernier programme en mémoire.

01 IBI EN **Commentaire:**
RCL c lignes 01 à 07: isolent les deux
CLA derniers octets de c, qui contien-
STO 0 -nent l'adresse du .END. dans les
STO N 3 digits de droite.
ASHF
ASHF lignes 08 à 15: mettent le premier
RCL d digit à 3 (cad OCII en binaire)
ASTO d qui deviendra le pointeur d'octets
CF 08 du registre b.
CF 09
SF 10 lignes 16 à 19: stockent le registre
SF 11 ainsi constitué dans b.
ARCL d
STO d
LAAA
RCL E
STO b

Heureuse programmation...

19 END

Boschet

Quelques remarques sur l'interface vidéo II:

Cet interface ne réagit (hélas) pas toujours très bien aux sollicitations du type imprimante. Par exemple le caractère spécial | (append) est traduit par un blanc, ce qui est gênant dans les listages! on ne sait plus ce qui est ajouté au registre alpha et ce qui l'efface. D'autres caractères sont "traduits", par exemple ENTER ↑ est traduit par ENTER*.

Un des avantages principaux de cet interface est la possibilité d'avoir une copie de la pile à chaque opération, qui facilite beaucoup les mises au point de programmes. Mais, qui sait pourquoi? cette copie de la pile n'a pas lieu lors d'une entrée de chiffre, alors que cette entrée fait monter la pile et perd t.

Cette recopie de la pile est interrompue par un R/S: il reste sur l'écran une partie de la pile: celle-ci est imprimée en commençant par t puis z, puis y et enfin x (pas de Last x); un R/S peut donc vous laisser avec seulement t ou t et z... la reprise se fait à l'instruction suivante.

Les instructions illégales et les autres:

La fonction ACCRH doit envoyer à l'écran le caractère dont le numéro est en x. C'est OK sauf pour les caractères de contrôle 10 (Passer à la ligne) et 13 (retour en début de ligne) qui ne répondent pas. Il faut alors placer ces caractères en alpha et faire OUTA, ce qui détruit le registre alpha. L'indicateur 17 ne semble pas avoir d'effet là dessus, ais-je oublié quelque chose?

La fonction ACCOL, illégale, fonctionne très bien! mais bien sur elle n'accumule pas des colonnes qui ne sont pas reconnues, par contre : 1) elle écrit un blanc à l'emplacement du curseur, 2) elle envoie le caractère dont le numéro figure en x, y compris les caractères 10 et 13 !!! par contre l'espace inscrit empêche l'utilisation de ACCOL pour des séquences d'échappement. SKPCHR, illégale a comme effet d'effacer (écrire un espace...) un caractère, mais affiche DATA ERROR si le nombre en x n'est pas compris entre 1 et 23.

SKPCOL accepte des nombres de 0 à 167. De 0 à 7 SKPCOL efface 2 caractères, des 8 à 167 elle efface 3 caractères.

BLDSPEC/ACCSPEC fonctionne comme ACCOL quand on utilise le code du caractère, mais permet d'envoyer plusieurs caractères d'un coup. Ceci ne modifie pas le registre alpha et peut donc être utile.

J D D (T1)

* * * * *
* * * * *

01 BIL ABEM
02 ROL
03 B TER
04 BIE
05 =0?
06 R(X)
07 RD
08 VRC
09 BIE
10
11, 111007
12
13
14
15 BIE
16 BIE

Ne trouvez-vous pas rageant que votre machine s'arrête en affichant ALPHA DATA ou OUT OF RANGE, voire MONTREMENT quand vous essayez un programme de jeu? Cela arrive souvent à un générateur aléatoire quand R 00 contient une chaîne alpha ou un nombre astronomique, ou quand vous êtes en BIE 00.

Voici un programme qui teste les conditions d'erreurs dues au contenu de R 00 tout en conservant x et y dans Y et Z (respectivement 1). Il accoupe 5 registres et n'utilise que le registre 00.

Si vous avez le module X SUBSTP, le MONTREMENT provoqué par un BIE 00 peut être supprimé par:

00 BIE?
01
02 R(X)
03 BIE
04 RD
05 RD

L'amélioration peut être apportée si vous avez le module ME (ce n'est pas son cas): insérez un BIE 00 après le premier VRC, mettez un R 00 dans R 01 avant le B D, la fonction (je ne sais pas comment elle s'appelle) mettant en X l'heure, puis le VRC. La génération est sera encore plus aléatoire.

Le ME ABEM vous dévorera 3 octets, mais vous pouvez les ramener à 3 en recouvrant le son, ce qui est équivalent à un ASC numérique ou binaire. Un espoir de voir beaucoup de ME ABEM dans vos programmes...

T 270 Serge Vaudenay

* * * * *
* * * * *

HP KEY NOTES EST MORT
TIPS & ROUTINES SURVIT

C'est en lisant le PPC journal de Janv/Fev 83 que j'apprends la disparition de "KEY NOTES".

Aussi je propose de reprendre cette rubrique dans notre journal, en publiant régulièrement des routines remplissant, autant que possible, les conditions suivantes:

- 1/ caractère inédit,
- 2/ d'intérêt général,
- 3/ utilisation maximale des possibilités de la pile (entrées-traitements-sorties),
- 4/ ne recourant pas à la programmation synthétique.

Je m'explique:

En ce qui concerne le premier point il est bien entendu qu'il s'agit malhonnêtement de piller les excellents ouvrages qui existent tels:

"PROGRAMMER HP41" Descamps & Dhenin Editions PSI
"CALCULATORS TIPS & ROUTINES de DEARING etc...

Ceci étant tout le monde ne lit pas l'anglais et il est regrettable que des possesseurs de HP41 ignorent des routines fondamentales comme:
"IN" "OUT" "Y/N?" (Key Notes V5N1p14) *

Dans ces conditions je pense qu'il serait bon, avec l'accord de HP, de publier, après traduction, ces routines et quelques autres en les classant, par exemple selon le plan suivant:

- SAISIE: ex "IN" *
- AFFichage: ex "DISP" V5N2p13
- IMPression: ex "AV" V5N2p12
- GRAHiques: ex "DATA2" V5N2p12 (ou COURBES)
- CALCuls: ex "FAC" V6N4p10
- TRAITement: ex "SQRT" V6N4p14
- DIALogue: ex "Y/N?" *
- PROGRAMMation: ex "GOSUB" V5N3p14
- etc...

J'ouvre cette rubrique en vous proposant "MN" (MiNimum/maximum) et je lance un appel à tous les lecteurs pour l'alimenter car seul je n'irai pas loin...

Il est tout à fait sûr que PPC-T a une fonction voisine de celle de Key Notes, en peut-être un peu plus élaborée, le niveau de Key Notes n'ayant pas toujours été très relevé. Publier un résumé de tout cela est un très gros travail pour celui qui fera la compilation et me paraît justifier une version française de Calculator Tips and Routines, ne serait-ce que parcequ'il y a eu beaucoup de choses vues depuis CTR. Je suis en train de mettre en place une maison d'édition et je suis prêt à examiner tous les projets de manuscrits. Qui va s'y mettre?

MÉTHODOLOGIE DE LA
PROGRAMMATION
HP 41

Quand on souhaite soumettre à la bibliothèque HP un programme il faut le documenter à l'aide de formulaires spécifiques.

Ces documents, bien conçus, se révèlent utiles lors de la mise au point des programmes; la page consacrée aux "registres, indicateurs & modes opératoires" étant même un outil que je trouve indispensable.

Toutefois rien n'est prévu pour les labels, sauf à utiliser une feuille de "listage du programme" et à la compléter des labels locaux et autres labels.

Pour pallier cet inconvénient j'utilise le document, dont photocopie jointe, qui reprend:
-les labels numériques de 00 à 99
-les labels alpha de "AàZ" & de "a à e"
et ménage de la place pour d'autres labels

Vous remarquerez que sont surlignés d'un trait fort: les labels numériques courts de deux traits forts: les labels locaux alpha d'un trait fin: les labels numériques correspondant aux codes touches (pour une utilisation simple de "GETKEYS" (module X) Par ailleurs les registres de la pile sont entourés

Espérant que vous publierez ce document, grandeur nature, afin que vos lecteurs puissent aisément utiliser des reproductions photocopiques, je vous prie de croire à mes sentiments les meilleurs

J FURGEROT
PPC 8435
PPCT 156

Je pense que la réalisation de cette feuille est très simple à partir de vos indications, sans qu'il soit nécessaire de la publier.

"responsables" par sujets, chargés d'animer une rubrique (comme Microscene de Craig Pearce ou Viva 25, rubriques de PPC US). Jusqu'à présent les volontaires ne se sont pas pressés. Nous avons par exemple plusieurs officiers de marine et plusieurs pilotes d'avion sur les 300 membres du club. Ils ont peut-être des choses à se dire. Le journal est là pour ça. A vous

Au fait j'ai toujours à vendre un exemplaire du fameux livre de Baumgartner (70 F franco)!

NOP: correction des erreurs des précédents existe quand des erreurs me sont signalées....C'est le cas cette fois-ci.

Feedback s'appelle "Réaction" dans PPC-T et figure à chaque numéro.

Le classement que vous demandez pour le journal existe bien: L'éditorial, les clubs, les journaux, sont en page 2; suivent 4 grandes rubriques: Programmes, en nombre insuffisant pour être subdivisés, Applications concernant les matériels, les astuces de programmation, les cours destinés aux débutants (très recherchés), Réactions qui contient le courrier et les remarques concernant le journal ou le club. Il est presque impossible de subdiviser davantage, sauf à créer une nouvelle rubrique, comme NOMAS (Not manufacturer supported: sans garantie du fabricant) pour le microcode, si la matière le rend possible. En fait quand vous écrivez au club, vous déversez tout votre savoir dans votre machine à écrire et il n'est pas toujours facile de trier!!!

LE COIN DES U PRO

(LE COIN DES Utilisateurs PROfessionnels de la HP-41)

Jean FURGEROT PPC8435-PPCT156 40ans, exploitant de Banque, Trésorier d'un LEP de Photographie, prof au CFPB

utilise les programmes personnels suivants

- Dépouillement de Bilans (quad)
- % nominaux & actuariels sur placements à terme (bons & comptes - intérêts pré & postcomptés)
- PICOVISICALC: toutes opérations sur un tableau de 10x12 max, y compris libellés (configuration: Quad +X fonctions + cassette)
- Barème IRPP
- Analyse des encaisses (quad)
- Portefeuille Titres (X fonctions)
- etc...

ECHANGE CES PROGRAMMES contre

programmes &/ou algorithmes &/ou organigrammes traitant des sujets suivants:

- prêts à remboursement progressif (de 2 à n palier)
- calcul complet de l'IRPP
- tenue de comptabilité
- tous programmes utilisables dans ses activités professionnelles

RECHERCHE POUR DES ECHANGES DE VUES UTILISATEURS

- de l'excellent ouvrage de Mrs BAUMGARTNER & PETITGAND "PICO-INFORMATIQUE ET GESTION. D'ENTREPRISE" (les Editions d'Organisation PARIS 1981)
- du module FINANCIAL 1

Merci de me joindre à mon domicile 1-6556710 ou de m'écrire 68, rue G Péri 92120 MONTROUGE

XX

APPEL AUX UTILISATEURS PROFESSIONNELS DE LA HP41

En adhérant au PPC, puis au PPCT, je pensais pouvoir joindre des utilisateurs de HP41 ayant les mêmes préoccupations principales que moi: mise au point, échanges d'idées/programmes utilisables dans ma profession.

Hélas, j'attends toujours l'édition de l'annuaire du PPC et les listes d'adhérents du PPCT n'indiquent ni les professions, ni les sujets d'intérêt.

Aussi pensant ne pas être le seul dans cette situation, je propose l'ouverture d'une rubrique baptisée "le coin des utilisateurs professionnels"

ou "COIN DES "U" PRO"; rubrique largement ouverte sur l'extérieur mais qui, dans un premier temps, contiendrait la liste, par profession et/ou sujets d'intérêt des membres du PPC-T avec indication des n° de tel (domicile et/ou travail).

Bien entendu

- 1/la mise à jour de cet annuaire serait publié dans chaque numéro du Journal
- 2/ceux qui désireraient ne pas y figurer sont dès à présent invités à le faire savoir au Club

Ensuite cette rubrique pourrait traiter les thèmes suivants:

- programmes personnels utilisés par les membres
- recherches de programmes
- citations ou analyses d'articles ou d'ouvrages traitant de sujets professionnels
- propositions commerciales
 - .de séminaires, stages, conférences etc...
 - .de vente de programmes (y compris ceux proposés par HP à travers CORVALIS ou Genève)
- banc d'essai
 - .programmes
 - .stages etc...

Les beaux discours ne valent pas l'action aussi je vous montre l'exemple en me poussant à l'eau.en espérant bien ne pas être le seul à ramer.

Jean FURGEROT 68, rue Gabriel Péri 92120 MONTROUGE tel dom 655 67 10 précédé du préfixe:1 pour les appels de la province ou de l'étranger (il est permis de rêver non???)

T178

Bondy, le 16.03.83

Bien cher Jean-Daniel,

Un grand merci pour tout ce que tu m'as envoyé. J'ai commandé à Didier JEHL un lecteur d'EPROM et je vais donc commander TR-1D (ToulROM). Je vais également commander PPC ROM.

Tu trouveras ci-joint copie du "papier" que j'ai envoyé à Philippe GUEZ (P1) car je suis adhérent (récent) à PPC-PC. Tu comprendras que la proximité de mon domicile a été la raison principale de ma décision et, comme l'a écrit Jacques VAUCELLE : <<certains HPistes enragés s'abonneront probablement aux deux revues...>> (sic).

Puisque tu nous demandes notre avis sur la présentation du Journal, j'indique que le format d'écriture est particulièrement pénible à lire pour ceux qui n'ont plus le^s yeux de 20 ans et qu'il serait peut-être possible de gagner de la place en éliminant la publication des lettres ou parties de lettres ne présentant aucun intérêt telles les demandes de matériel ou d'adhésion. Autrement dit, suppression de tout ce qui ne peut être d'aucune utilité pour les lecteurs. Je sens que cette disposition pourrait s'appliquer souvent à mes bavardages.....

La publication ou plutôt la distribution de PRGM en CB aurait facilité la tâche de Damien DEBRIL avec qui je correspond^s régulièrement. Tu sembles "anti-CB" ! C'est dommage pour les heureux possesseurs de lecteur.....

Je t'envoierai un papier sur W" dès que mes recherches seront complétées....

Amicalement et respectueusement,

Recu le 21 avril, suite à la grève des postes....

J'avais écrit précédemment qu'il était possible d'obtenir des LBL, GTO & XEQ délocalisés avec le BG. La procédure que j'avais trouvée pour LBL "d" était assez longue mais JDD (°) m'a gentiment indiqué le moyen d'obtenir le même résultat d'une façon beaucoup plus simple :

```

PRGM LBL "T"           Ø1 LBL "T"
LBL "T"                Ø2 LBL "T"
x=y? x=y?             Ø4 X=Y?
GTO .ØØ1 BG SST       Ø3 LBL ØØ
SST                    Ø4 "T"
GTO .ØØ3 BG SST       Ø5 CHS
← +                   Ø5 +
X)Ø?                  Ø6 X\Ø?
PACK BST              Ø5 +
← BST                 Ø3 LBL ØØ
BG ← SST ← SST       Ø3 "d"
GTO .ØØ1 PACK BG ← SST ← Ø1 LBL "T"
SST                    Ø2 LBL "d"
    
```

Ne pas faire BST à la place de GTO .ØØ3 car nous serions parachuté dans les REG d'ASN.

CHS apparu au Pas Ø5 a le même code que T + est un code quelconque destiné à prendre la place d'1 octet et qui sera ensuite effacé pour créer le NUL (octet indispensable pour stocker la "distance").

(°) Je tiens ici à remercier Jean-Daniel Dodin (T1) pour son aide efficace et éclairée sans laquelle les lignes précédentes n'auraient pas pu être écrites. PPC-T 77, rue du Cagire 31100 TOULOUSE . (je suis T178)

Passons maintenant aux GTO "d" :

```

PRGM LBL "R"           Ø1 LBL "R"
GTO "R"                Ø2 GTO "R"
BST BG                 Ø2 "A-----"
SST                    Ø3 "R"
← "d"                  Ø3 "d"
GTO .ØØ1 BG ← SST ← SST Ø2 GTO "d"
    
```

Vous avez remarqué que le signe # qui apparait dans la chaîne issue du BG correspond au code 29 qui est GTO " " ; vous pouvez obtenir XEQ "d" en remplaçant GTO "R" par XEQ "R" .

Comme je vous le disais, l'utilisation du BG est un "sport" dont l'entraînement permet de comprendre la constitution des fonctions.

Je dois préciser que mes "découvertes" n'ont été possibles que grâce à la lecture attentive du livre "AU FOND" de JDD. Donc: un grand merci à JDD ! à bientôt, peut-être R.S.

(copie du présent "papier" envoyée à JDD)

T178 Coup d'oeil sur W"

La "fonction" #ae2 ASN/"KA" avec les codes 1,31 permet d'obtenir bien des "choses".

Je me contenterais de donner quelques exemples.

#ae2 étant ASN à la touche 5 (W) :

```

Mode CALcul
XEQ "a"      affichage:  NONEXISTENT
PRGM USER W USER      pp W" a
CAL XEQ EMDIR          DIR EMPTY
PRGM USER W USER      pp W" EMDIR
CAL XEQ DIR etc....    pp (DIR...)
    
```

La manip avec EMDIR suppose que le XF se trouve placé dans un port et pour DIR, l'interface boucle "IL". Dans ce dernier cas, le lecteur de K7 peut rester sur OFF.

Vous pourrez découvrir bien d'autres utilisations de cette "fonction" bizarre, vous avez dit bizarre!... et, comme je crois vous l'avoir dit précédemment, ce code 31 (en DEC) est INDISPENSABLE pour introduire en ALPHA (puis en X) la chaîne adéquate permettant de manipuler les 44 premiers drapeaux avec STOFLAG.

Qui a dit qu'il n'y avait plus rien à découvrir ? Je vous assure que j'ai découvert tout cela sans aucune aide quelconque mais je pense que d'autres l'auront fait avant moi.

Je vous laisse le plaisir d'exploiter ce "sauteur d'octets".

Bon entraînement, R.S.
pp indique le N° du PAS correspondant.

.....

Mr PELANNE J.F.
60 rue Saint-Flacide
75006 - PARIS
T-191

Cher Monsieur DODIN

Gd mère HP affirme que l'on ne peut exécuter un prgm directement à partir de la mémoire étendue mais moi je dis, on le peut, et ce qui suit le prouve : Procéder très précisément comme suit :

- 1/ Mémoire étendue vide (XF ss XM)
 - 2/ Faire le SIZE correspondant au prgm que l'on va enregistrer. Appelons le XX (pour que celui-ci soit exécutable).
 - 3/ Remplir la mémoire étendue avec un prgm de 100 R par ex. avec SAVEP puis l'effacer de la mémoire principale.
 - 4/ Remplir la mémoire principale de façon qu'il soit impossible de copier le prgm de 100 R par GETSUB.
 - 5/ Disposer du XROM 05,03 assigné à une touche (LN ou autre).
 - 6/ Une fois que les conditions 1/ à 4/ ont été remplies faire XROM 05,03 en mode calcul.
 - 7/ Passer en mode prgm, faire SST jusqu'à voir LBL^XX (notre prgm) label du prgm existant en mémoire étendue (soyez patient surtout si beaucoup de touches du clavier ont été assignées à des fonctions ou des prgms). Attention n'appuyez pas sur une touche autre que SST ou BST sinon MEMORY LOST, nous sommes après la zone d'assignation, certainement dans la zone mémoire du X Ftes.
 - 8/ Passer en mode calcul et faites XEQ alpha XX alpha affichage de NONEXISTENT évidemment puisque celui-ci n'est pas dans la mémoire principale.
 - 9/ Repassons en mode prgm et faisons SST jusqu'au END du prgm puis à nouveau SST, la numérotation des lignes à changer, nous sommes désormais hors de la table assignations (cette phase 9/ n'est pas indispensable).
 - 10/ Se placer sur LBL^XX puis repassons en mode calcul et faisons R/S , une chaîne de caractères synthétiques peut apparaître, surtout ne rien toucher surtout pas R/S , quelques secondes plus tard, le prgm s'exécute normalement.
- Et voilà on peut exécuter un prgm placé en MEM étendue facilement et directement sans avoir à le copier en MEM principale contrairement à ce qu'affirme Gd mère HP.

Le 21 oct 81 on ne doit pas avoir le END correspondant au XROM 05,03.

Cher Monsieur,

J'ai fait de nouvelles découvertes depuis ma dernière lettre, en complément de ce que j'ai dit précédemment. Un programme privatisé par WPRV se trouvant en MEM

trouve confronté. J'ai pu remarquer que la destruction d'une chaîne de caractères nulle dans les registres d'assignations provoquent souvent un tel plantage. Le remède est alors simple: provoquer un MEMORY LOST sans éteindre la 41! pour cela, soit on stocke une valeur bidon dans le registre d'état c, soit on se place dans les registres d'assignations et introduit des instructions quelconques ce qui amènera inévitablement au MEMORY LOST recherché

Troisième sujet: La programmation des labels contenant des ca

ractères "synthétiques" ou non. il suffit d'entrer la séquence d'instructions suivantes: RCL IND 64
SIN
Chaîne de n+1 chr

deuxième sujet de ma lettre: j'ai lu dans PPC6T ou dans l'OI (?) la lettre d'une personne se plaignant de n'avoir pas d'ouvrages sérieux sur l'astronomie de position. Je citerai donc un livre qui traite intelligemment du sujet:

CALCUL ASTRONOMIQUE POUR AMATEURS
adapté à l'emploi d'un calculateur ou d'un micro-ordinateur

AUTEUR: s. Bouiges
édité chez MASSON

Pour éclaircir les idées, en voici le plan:

1ère PARTIE/

Rappel de notions élémentaires d'astronomie

- 1..Les mouvements dans le système solaire
- 2..Systèmes de coordonnées
- 3..Le temps

2ème PARTIE:

le calcul des positions planétaires

- 4..Calcul des coordonnées écliptiques et héliocentriques d'une planète
- 5..Coordonnées géocentriques
- 6..Constantes planétaires
- 7..Mouvements de la LUNE
- 8..Amélioration de la précision
- 9..Coordonnées apparentes
- 10..Satellites de JUPITER et de SATURNE
- 11..Cycles
- 12..Déterminations de l'orbite d'une comète
- 13..Les étoiles doubles

3ème PARTIE:

Les relations ciel-terre

- 14...Temps sidéral
- 15...Calcul des coordonnées locales
- 16...Lever et coucher des asters
- 17...Le triangle parallactique
- 18...Visibilité d'un astre
- 19...Calcul d'un cadran solaire plan
- 20...Coordonnées héliographiques

4ème PARTIE:

la programmation sur calculatrice de poche

ANNEXE 1: Exemple de programmes

- HP 67: calcul de N
- 41C : coordonnées des planètes
- TI 59: " " "
- 41C : lever et coucher du soleil
- TI 59: étoiles doubles
- BA51C: éphémérides

ANNEXE 2: Note sur fonctions trigo inverses

le 1er caractère de cette chaîne est le caractère nul codé 00. Compactez ensuite la mémoire (PACK) et BG sur l'instruction précédente RCL IND 64. détruisez la chaîne "?----" qui en résulte et SST pour voir le label désiré. vous pouvez ainsi avoir des labels jusqu'à 14 caractères. Cependant seuls ceux de moins de 7 caractères peuvent faire l'objet d'un branchement par GTO ou XEQ. si vous désirez assigner un tel LBL, le 1er caractère de la chaîne synthétique devra correspondre au code de la touche à laquelle vous désirez assigner le LBL (chaîne de n+1 caractères mentionnée précédemment.). ensuite, soit vous armez le bit correspondant à cette touche dans le registre e ou t (ce que je déconseille car c'est fastidieux) soit vous stockez ledit programme sur k7 ou sur carte magnétique en prenant soin d'être en mode USER. Lorsque vous chargerez ce programme en mode USER, il sera assigné à la touche désiré.

un exemple clarifiera les choses (je l'espère) soit à créer puis à assigner LBL"ABCDEFG:" à la touche Σ+.

entrez la séquence suivante:

```
01 sin
02 RCL IND 64
03 SIN
04 "IABCDEFG:" (F9 01 41 42 43 44 45 46
47 3A)
```

le caractère 01 correspond au code de la touche Σ+. Ensuite BG sur 01sin. SST une fois pour voir apparaître LBL "ABCDEFG:". Enregistrez ensuite ce programme en prenant soin d'armer USER. Lorsque vous lirez cette carte; USER armé; le programme sera assigné à la touche Σ+. Je crois que l'intérêt de ceci est surtout ludique encore que certains y trouveront peut-être des applications.

j'ajouterais rituellement:

Happy programming ou pour les francophones
JOYEUSE PROGRAMMATION

Jean-Claude CATY 9412 T168

Dernière heure: H-P annonce "une surprise dans les magasins" pour Août ... en FRANCE aussi??

PROGRAMMATHÈQUE

Depuis quelque temps j'essaie d'organiser ma collection de programmes. Cependant je n'arrive pas à satisfaire toutes mes exigences sur la forme que devrait avoir cette future programmathèque. Pour bien faire il faudrait :

- être sûr de la fiabilité de conservation des programmes
- pouvoir retrouver celui que l'on recherche facilement
- être aisément transportable et stockable
- chaque programme doit être expliqué clairement. Un néophyte doit pouvoir le mettre en oeuvre.
- être accessible à d'autres qu'au créateur

Si je n'ai rien oublié, on peut commencer à transformer ces désirs en réalités. Alors pour être sûr de retrouver un programme, il ne suffira pas de l'enregistrer sur cartes, cassettes ou codes barres, car ces systèmes ne sont pas assez fiables pour être uniques. Il faudra donc conserver une liste de chaque programme. Cependant, elles ne seront consultées que très rarement et elles impliquent un volume important et donc une gêne. Alors on devra séparer notre programmathèque en deux parties : la première constituée de cartes magnétiques, de codes barres ou de cassettes, et la seconde composée des listes de tous les programmes. La relation entre ces deux formes de stockage se fera par l'intermédiaire de numéros qui correspondront chacun à un programme donné. On pourra même conserver, par écrit, les différentes versions d'un programme quand, par simplicité et par économie, on n'en gardera qu'une utilisable directement.

Mais maintenant, comment retrouver le programme désiré ? Eh bien, le mieux est encore de stocker tous les programmes d'une manière donnée, de référence, et de faire des fichiers faisant la liaison entre les différents classements. En effet, on peut classer : par ordre chronologique d'enregistrement, par ordre alphabétique, par auteurs, par taille, par sujet, par fréquence d'utilisation, et que sais-je encore... ? Par exemple, on pourrait stocker, en référence, par ordre chronologique d'enregistrement, ce qui impliquerait un couple : numéro d'arrivée-programme. Et créer, parallèlement, autant de fichiers de correspondance que l'on souhaiterait de classements. Ils seraient composés de couples : forme de classement particulier-numéro d'arrivée.

Mais on pourrait encore améliorer le système en dédoublant le stockage des quelques programmes utilisés quasi-quotidiennement. Cette petite annexe, trop petite pour être classée à son tour, serait contenue dans la housse de la H.P.. Ainsi on pourrait amener sa machine en voyage avec la première partie de sa programmathèque, et même en simple déplacement avec son annexe. En effet, pour être facilement transportable, il faut bien admettre que l'utilisation du lecteur de cartes s'impose, au moins pour le stockage de l'annexe de la programmathèque.

Pour qu'un programme soit utilisable par un néophyte, il faut qu'un mode d'emploi soit noté quelque part. Je pense que le mieux serait de l'écrire sur une fiche signalétique qui serait attribuée à chaque programme de la première partie. Ces fiches seraient placées à proximité du programme stocké, dans le cas d'utilisation des cartes et des codes barres. Ou bien, elles seraient classées en carnet, dans le cas de l'utilisation des cassettes. Voici un exemple de fiche signalétique :

Nom :	
Sous-programmes/	Extensions nécessaires/
Code/	Labels et affectations/
	Explications/
-	
-	

Où, après Nom on écrit : le nom du programme et son objet.
 Où, sous Sous-programmes/ on note les différents programmes nécessaires à son utilisation.
 Où, sous Extensions nécessaires/ on inscrit parmi 4R (Quad RA), K7 (lecteur de cassettes), LC (lecteur de cartes), MT (module temps), ME (module d'extension), IL (interface IL), IMP (imprimante),... les accessoires nécessaires à son fonctionnement.
 Où, sous Code/ on écrit le code correspondant aux différents nombres caractéristiques du programme :
 - 0000 pour le nombre d'Octets programmes utilisés
 - 00P pour le nombre de Pistes de cartes magnétiques occupées
 - 000M pour le nombre de registres Mémoires nécessaires
 - 000F pour le nombre de registres de Fichiers nécessaires
 Exemple : 25003P19M20F
 Et où, sous Labels et affectations/ on note les affectations des labels, exemple : SH23, CH24, TH25.
 Et enfin où sous Explications/ il faut placer le plus dur : le mode d'utilisation, l'objet précis et les performances du programme.

Et pour que tout ce travail soit publiable ou simplement utilisable par des amis, il faudrait qu'il soit écrit lisiblement ou mieux, tapé à la machine. De même, une notice d'utilisation de la programmathèque pourrait être utile. Elle contiendrait : le type du classement de référence et des autres, un schéma expli-

-quant le fonctionnement de toute la programmathèque, l'explication du numéro de code et tous les conseils que vous jugerez utiles de donner aux néophytes.

Ah ! J'oubliais ! Il ne serait plus besoin alors de noter sur les cartes ou cassettes les noms des programmes contenus, mais simplement leur référence dans le classement de référence.

Mais alors, me direz vous : pourquoi n'arrive-je pas à constituer cette programmathèque dont je rêve si précisément ? Eh bien simplement parce que je devrais travailler sur plus de 180 programmes, et aussi parce que je ne possède pas d'imprimante. Mais c'est petit à petit que l'oiseau fait son nid, alors je rattrape lentement mon retard pour atteindre mon but.

En fait, cet article a surtout été écrit pour les débutants qui ont tendance (comme moi) à accumuler toutes sortes de programmes. Attention ! Faites le tri et le classement au fur et à mesure, sinon vous glisserez doucement vers le même merdier duquel j'essaie de sortir.

GEORGELIN Daniel T 57

J. Pividal T 72.

MODE D'EMPLOI D'ASC
 APRES AVOIR CREE UN FICHER AVEC CARLAS FAIRE XEQ ASC, ENTRER LES CARACTERES .
 S'IL Y A PLUS DE 24 CAR. APRES LE BEEP SORTEZ DU MODE ALPHA APPUYEZ SUR G: POUR CREER UN ARTICLE , F: POUR RAJOUTER DES CARACTERES , H: POUR RETABLIR TOUS LES FLAGS .
 S'IL Y A MOINS DE 24 C ON PEUT EMPLOYER , G: POUR CREER UN ARTICLE , F: POUR AJOUTER DES CHAINES MAIS ON NE VERRA PAS LA FIN DE LA CHAINE EN COUR .

POUR ECRIRE CE MODE D'EMPLOI J'AI UTILISE :
 1 IMPRIMANTE RAPIDE HP-IV A220 A + HP40C
 LE PRGM ASC (GIL/PIVIDAL)
 LE PRGM ED (POLJEDU) POUR CORRIGER LES ERREURS .
 LE PRGM HLP (VALCELLE) POUR IMPRIMER

22:06 05/04
 01LBL "ASC"

22:07 05/04
 01LBL "ASC"
 02 CLA
 03 "0" (484,1280)
 04 RCL .
 05 STO L
 06 RCL d
 07 STO Z
 08 CLA
 09LBL 00
 10 ASHF
 11 ASHF
 12 ASHF
 13 RCL L
 14 X() d
 15 AVIEW
 17 STOP
 18LBL F
 19 APPCHR
 20 GTO 00
 21LBL G
 22 APPREC
 23 GTO 00
 24LBL H
 25 X() Z
 26 STO d
 27 END

est-il lisible ?
 JDD
 (T2)

05/04/83 23:02:01 LISTEUR DE CODE BARRE -GIL- PAGE 1 VERSION 4/83

PROGRAMME = ASC

ROW:1						
ROW:2						
ROW:3						
ROW:4						
ROW:5						
ROW:6						
ROW:7						

H-P 15C

Article écrit initialement pour "l'Ordinateur de Poche (qui l'a refusé!) c'est pourquoi il reprend tout au début. Ce n'est sans doute pas inutile.

J-D Dodin
HP-15C

Où nous allons relever les jupes de cette demoiselle!

Une fois n'est pas coutume, c'est la maman qui donne le mode d'emploi pour déboutonner sa fille! les séducteurs entreprenants auront sûrement été intrigués par une note à la page 263 de la notice française de la HP 15C: dans le chapitre consacré à la "vérification du fonctionnement", le §1 précise: "Si le calculateur ne répond pas aux pressions de touches... appuyez sur y'x et ON simultanément, puis relâchez les. Ceci modifie le contenu du registre X, effacez donc ce dernier ensuite."

Et, se dit notre séducteur, si je ne l'efface pas ? Avant de faire la première expérience, précisons bien les conditions. Il faut d'abord éteindre la HP, puis presser simultanément sur les touches ON et y'x (j'écris ainsi y puissance x, touche 14), puis les relâcher DANS N'IMPORTE QUEL ORDRE. La HP s'allume avec à l'affichage... ne soyez pas trop pressé!

Cette séquence de touches, nous aurons à la répéter souvent, donnons lui donc un nom: D22. D pour Demoiselle, 22 v'la la maman!

Nous pouvons maintenant commencer à lever le voile Tapez le chiffre 7,77777777 E77, donc 7, une virgule (ou un point, si vous préférez la méthode US), neuf autres 7, pressez la touche EEX puis encore deux fois 7, vous voyez: 7,77777 77. OK ?

Effectuez maintenant la séquence D22 décrite ci-dessus. avez-vous pensé à éteindre d'abord la machine ? vous devez voir: 0,000000000 si vous êtes en FIX (quelque soit le n°)

1,422422-56 si vous êtes en SCI 9. Curieux ???

Faites maintenant f PREFIX pour voir la mantisse... Oh surprise! vous voyez : Pol^uoIPPPP. P. doit être l'initiale de "Porno", car ce que nous voyons ici dépasse les limites de la bienséance! Notre HP-15C aurait-elle fauté avec une 41C? pour montrer ainsi des caractères alphanumériques ?????

Des dessous pas très catholiques :

Arrivés sous les jupes, nous allons devoir parler de matrice. Ce texte devient de plus en plus scabreux !

Vous savez sans doute , si vous possédez une HP-15C qu'elle affiche les matrices en présentant à l'écran , à gauche une lettre donnant le nom de matrice , à droite deux chiffres donnant les dimensions de celles-ci, par exemple RCL MATRIX C donne |C 0 0| (les |représentant le bord de l'écran) si la matrice C est dimensionnée à 0.

Tapez maintenant 1,014 et faites D22 DEUX FOIS DE SUITE. Vous devez voir |y n m|. y est le nom d'une matrice que vous ne connaissiez pas, n et m les dimensions de la matrice A. On constate à l'expérience que la HP 15C peut afficher 16 matrices différentes que tous les passionnés d'informatique sauront immédiatement numéroter en hexadécimal de 0 à 9, puis de A à F. Noter que les matrices 0 et b ,8 et d s'affichent pareil mais n'ont pas la même valeur, idem pour les matrices 2 et 9. Vous trouverez les 10 premières matrices en remplaçant dans le chiffre 1,01n le n qui était tout à l'heure égal à 4 par une valeur de 0 à 9, les matrices A,B,C,D et E sont les matrices standard, la matrice F existe mais ne peut être vue que par des manipulations hasardeuses.

Les matrices de la HP 15C peuvent être manipulées (oh) de deux façons : directement en les appelant par leur nom à l'aide des touches A à E ou indirectement en plaçant la matrice à traiter dans le registre d'index I.

C'est ce que nous allons faire avec notre matrice y, puisqu'aussi bien elle est disponible en x. Refaites maintenant cette matrice si vous vous en étiez écarté, puis STO I. Il faut maintenant DIMentionner les matrices convenablement. Faites 0 ENTER, f DIM C, f DIM D, f DIM E; puis I Enter 50 (en fait le plus grand chiffre accepté selon le contenu de votre machine est le meilleur. Pour commencer, prenez une machine vide) f DIM A. Si aucun élément perturbateur imprévu ne figure en mémoire, en faisant RCL (i) vous devez voir |y 1 50|.

Nous allons maintenant examiner où se trouve notre machine quand à son "état" en faisant g MEM. Nous voyons :

dd uu pp-b (cf notice, page 215)
dd= numéro du dernier registre de la zone donnée, uu nombre de registres disponibles dans la zone commune, pp=nombre de registres utilisés dans un programme, b=nombre d'octets restant avant la conversion en programme d'un nouveau registre.

Additionnez uu et pp. Vous aurez à faire cette opération souvent, n'oubliez pas que demoiselle HP sait aussi faire les additions !

faites maintenant uu+pp + 27 STO 1
(en notation HP : uu ENTER pp + 27 + STO 1).

Continuez par I STO 0

Vous venez d'initialiser les numéros de rang (R 0) et de colonne (R 1). Continuez en rentrant le nombre 1,006363 et faites STO (i). Essayer maintenant RCL E. Si vous voyez |E 99 99|, vous avez franchi un petit pas pour l'homme mais un grand pas pour la HP, elle va s'abandonner dans vos bras. Remarquez que la matrice D est dimensionnée à 1,0 ce qui est quand même curieux. Laissez le ou supprimez le, c'est tout un, vous pouvez maintenant librement disposer de vos matrices A à D. Laissez E à 99,99, cela va se révéler très utile.

Assimilez bien les manipulations ci-dessus, vous en aurez besoin chaque fois que vous aurez réinitialisé votre machine. Le risque de réinitialisation accidentelle est faible, beaucoup plus faible que dans le cas de la HP 41C, cela ne m'est arrivé (pour l'instant) qu'une fois, mais j'ai déjà fait par erreur f MATRIX 0 ou -ON.

Cette fois elle est à vous !

Vous pouvez maintenant accéder à tous les registres de votre HP directement , car la Matrice NON Normalisée (MNN) est tout à fait différente des autres. son premier rang commence avec le premier registre de la zone commune et se continue à travers la mémoire programme et au delà , dans les registres d'état de la HP 15C

Ces registres se trouvent après le registre 65 , je les ai numérotés à partir de E1. Ils sont loin d'être tous entièrement élucidés , je ne sais même pas à coup sûr combien il y en a !

il est possible de les atteindre avec la matrice E 99,99 enmettant en R1 un numéro de colonne égal à uu + pp + numéro du registre , c'est ce que vous avez fait tout à l'heure avec le registre E27 , sans le savoir... Nous y reviendrons.

Les registres E1, E2 et E3 ont un contenu évident. J'ignore encore exactement ce que contient le registre E4 . N'oubliez pas que pour voir correctement le contenu d'un registre il faut visualiser la mantisse par f PREFIX .

Le registre E5 est le dernier affichage, du coup il vous renvoie toujours le E lnn (nn est le numéro de colonne) qui s'est affiché furtivement quand vous avez effectué RCL E. le registre E8 commande le mode USER de façon curieuse : si vous y stockez 1 (par exemple), le mode USER s'allume , et vous pourrez l'éteindre qu'en remettant ce même registre à 0 , la touche f USER n'est plus efficace . E10 et E11 commandent l'affichage , nous allons y revenir. E17 et E18 sont R0 et R1 , E19 est RI, E20 est Last x, E21 contient la racine de RANDOM et E26 et E27 les dimensions de matrices, d'ou l'opération de création de E 99,99, avez-vous vu 63 (hexa)=99(décimal)

Savez-vous comment je m'appelle?

Un petit programme va nous permettre de faire parler la demoiselle:

1	LBL A	
2	STO E	le u apparait si vous entrez l'instruction en mode USER, il signale l'incrément automatique des rangs et colonnes.
3	Ø	
4	-STO E	pas de mode user ni d'incrément
5	R↓	
6	DSE 1	
7	LBL 0	
8	GTO 0	

Vous allez initialiser en mettant uu+ pp + 10 en R1 et 1 en R0; puis tapez 4.370000774, faites une fois D22 et lancez le programme A. Si tout marche bien vous allez voir clignoter le nom de la demoiselle H-P, !

en x	matrice	dim. comme
1,010	b	C
1,011	≡	A
1,012	≡	b
1,013	γ	C
1,014	y	A
1,015	=	b
1,016	γ	C
1,017	β	A
1,018	d	d
1,019	≡	C
	A	A
	B	b
	C	C
	D	d
	E	F
	blanc	b

- 001 LBL 00
- 002 LBL 01
- 003 LBL 02
- 004 LBL 03
- 005 LBL 04
- 006 LBL 05
- 007 LBL 06

La HP 15C (comme toutes les HP) utilise pour ranger les nombres des registres contenant 56 bits (les 0 ou 1 élémentaires), ou encore 14 digits (un digit est un groupe de 4 bits), ou encore 7 octets (groupe de 2 digits ou 8 bits).

Si nous étudions les digits de gauche à droite, nous trouvons :
 un digit qui contient
 0 (bits 0000) si le nombre est positif,
 1 (bits 0001) si le nombre est le nom d'une matrice,
 9 (bits 1001) si le nombre est négatif,
 toute autre valeur provoque l'affichage d'un signe négatif, mais le nombre peut se comporter bizarrement !

Les 10 digits suivants sont ceux de la Mantisse, c'est à dire les chiffres significatifs du nombre, ceux qui sont affichés par f PREFIX.

Un digit qui contient le signe de l'exposant, à 0 ou 9 selon le signe.

Deux digits d'exposant qui sont normaux pour les nombres positifs mais codés différemment pour les nombres négatifs, en plus du signe de l'exposant : -02 est codé 998 (signe et exposant). Vous obtenez donc la valeur de l'exposant en faisant 1000-valeur, le résultat vous donne même le digit de signe.

Si vous avez rappelé en x un registre d'état ou un registre de programme, il se peut que le contenu de ce registre soit un nombre hexadécimal (voir encadré 2). Dans ce cas, l'affichage "normal" est fait sous forme "normalisée", c'est-à-dire que la HP n'affiche que des valeurs décimales. Les digits hexadécimaux sont traduits par leur contre valeur décimale (A=10, B=11...), avec report de la retenue ce qui entraîne même une rectification de la mantisse quand le digit le plus à gauche (le plus significatif) de la mantisse est hexadécimal. (déplacement à droite de la mantisse avec ajout de 1 à l'exposant.)

Ces rotations, surtout par paires permettent de décoder le contenu des digits de signes. Par contre, l'exposant est affiché en hexadécimal, et c'est également le cas de la mantisse montrée par f PREFIX. La comparaison de ces deux formes vous montre les effets de la normalisation. Cette normalisation n'affecte que l'affichage et non pas le registre.

D22 : ce sont les initiales de Droite 22, c'est-à-dire rotation de 22 bits à droite. Vous avez peut-être vu que 22 n'est pas un multiple de 4, donc le déplacement coupe en deux les digits pour les reconstituer autrement, ce qui permet des décodages au niveau du bit. Pourquoi cette fonction de "RESET" a-t-elle ce résultat, je l'ignore.

Deux D22 provoquent une rotation (en fait permutation circulaire) de 44 bits, soient 11 digits vers la droite, ce qui est équivalent à une rotation de 3 digits vers la gauche.

C'est ce qui explique la création de la matrice y avec 1,014 : le 1 est là pour servir de premier caractère de la mantisse, le deuxième 1 après rotation de 3 digits à gauche prend la place du digit de signe et transforme l'affichage en affichage de matrice, le 4 est le chiffre qui est traduit par y, c'est donc le numéro de la matrice.

Décimal, hexadécimal et HP 15C

Tous ceux d'entre vous qui lisent attentivement ce journal ont déjà entendu parler de la notation hexadécimale. En bref, les ordinateurs, ceux de poche comme les autres, travaillent sur des éléments binaires, les bits. Or avec 3 bits on compte de 0 à 7 et avec 4 bits de 0 à 15. S'arrêter de compter à 10 c'est trop pour 3 bits et c'est du gaspillage pour 4. Disons également qu'avec la table ci-dessous, il est très facile de traduire un nombre, même si il contient les 56 bits du registre HP, de binaire vers l'hexadécimal et réciproquement ce qui est beaucoup plus délicat à partir du décimal. Habituellement, avec les afficheurs comme celui de la 15C, dits "à 7 segments", les lettres A,B,C,D,E et F de l'hexadécimal sont écrits A,b,c,d,e,f. Mais, qui sait pourquoi ? La HP utilise d'autres symboles :

dec	binaire	hexa	H-P
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	8	8
9	1001	9	9
10	1010	A	T
11	1011	B	-
12	1100	C	o
13	1101	D	P (u)
14	1110	E	E
15	1111	F	blanc

Sortons lui les tripes!

Avant de préciser le contenu et la structure des registres d'état il est nécessaire de nous munir d'un outil de création de nombres hexadécimaux. Ceci va nous amener à déchiffrer les codes des instructions de la zone program.

faites g MEM pour lire dd uu pp placez en R1 la valeur uu+pp, en R0 la valeur 1, et faites RCL E (n'oubliez pas qu'il est plus commode de travailler en SCI 9).

Vous voyez 6,050403 00 qui, à l'aide de f PRGM et de D22 se traduit rapidement par 06050403020000

Il va falloir lire de droite à gauche par groupes de deux digits (un octet par instruction), la correspondance apparaît alors:

Ligne instruction	code
001 LBL 00	00
002 LBL 01	00
003 LBL 02	02
004 LBL 03	03
005 LBL 04	04
006 LBL 05	05
007 LBL 06	06

La traduction semble maintenant évidente, mais le cas du LBL 01 attire l'attention, en effet on attendrait plutôt le code 01! et de toute façon il n'est pas possible que deux instructions (LBL 00 et Lbl01) aient même code. En fait il est facile de vérifier que si on modifie l'ordre des Lbl dans le programme on obtient bien le code 01 pour Lbl 01, ceci nous amène à deux conclusions:

- le registre E 1,(uu+pp) est le premier registre de la mémoire programme,
- certains digits sont modifiés lors du rappel entre la mémoire et l'affichage (cf encadré).

Remarquez aussi que le deuxième registre de la mémoire programme est E 1,(uu+pp=1).

Et encore, que la HP écrit ses programmes en remplissant les registres de droite à gauche (les numéros de ligne les plus bas à droite).

Des essais successifs nous amènent à établir les tables d'instructions, voir N°5

La table des instructions "à un octet" est pleine, sauf la dernière colonne, celle des codes nF. Certains de ces codes annoncent des instructions "à deux octets", les autres paraissent inutilisés; et donnent des résultats bizarres quand ils sont placés (STO E) en mémoire. Je n'ai pas encore trouvé le code des instructions non programmables comme f PREFIX ou PROGRAM, mais il y a encore beaucoup de choses à essayer...

Il est relativement facile de créer un registre avec des valeurs hexadécimales en créant en début de mémoire programme la suite d'instructions ayant le bon code et en le rappelant en x. La normalisation du signe de l'exposant est cependant gênante, le D22 devient alors utile pour décaler le registre par bits ou par digits.

Reconstituer le squelette

Nous pouvons maintenant trouver le plan complet de la mémoire et détailler les registres d'état, en quelque sorte le squelette de notre demoiselle.

La matrice E 99,99 a le premier registre de son premier rang au début de la zone commune (uu) puis la zone programme (pp), puis les registres d'état. Par conséquent la création de nouvelles lignes de programme ne déplace pas l'origine de notre matrice. Par contre toute modification des zones matrice, registre complexe, solve ou integrate déplace notre origine.

Mais comment accéder à ces zones, situées au dessus de la matrice E ? tout simplement en utilisant le rang 0 de la matrice.

Rang 0? Mais 0 STO 0 RCL E donne Error 3!
 Il va falloir ruser! Voici les opérations dans l'ordre:

-créer un programme

- 001 LBL A
- 002 LBL B
- 003 LBL C

- faire g MEM et calculer la valeur uu+pp
- placer cette valeur en R1 et 1 en R0
- faire RCL E ;on rappelle ainsi le premier registre de la zone programme,

-faire D22 deux fois. On obtient un nombre non normalisé avec un exposant nul,
 -faire STO 0, ça y est, vous avez votre rang 0 de la matrice. a partir de maintenant STO ou RCL E fera apparaitre E 0,xx.
 Méfiez-vous, si xx est 99, au prochain tour (gare au mode user!) la HP incrémente RO et démoli votre édifice.
 Le rang 0 fini là où le rang 1 commence, ou plutôt un registre au-dessus. Donc E 0;99 est le dernier registre de la zone matrice.

- Il y a 64 registres à répartir au total hors les registres d'état, donc:
- la zone programme se compose des registres situés de 64 à 64-pp+1 (inclus)
- La zone commune se compose des registres 64-pp à 64-pp+1 (inclus)
- La zone spéciale se compose des registres 1 à 64-uu-pp (inclus)

Le registre n°1 a donc le rang E 0,99-(64-uu-pp)+1, soit E 0,99-64+uu+pp+1 = E 0,36+uu+pp
 Donc, en fait tous les registres sont compris entre E 0,36+uu+pp et E 1,uu+pp

Commentaire des tables des codes (cf N°5)

Les octets représentant les instructions sont indiqués comme ils apparaissent à l'affichage. le numéro de rang est le digit de gauche, le numéro de colonne le digit de droite.

Les instructions ayant un lien logique sont encadrées quand c'est possible pour faciliter leur recherche. Ainsi DSE (i) est le code 83, STO D est 4D.

Toutes les cases de la table sont occupées, sauf dans la colonne nF, qui est celle des préfixes des fonctions à deux octets.

Comme il n'y avait plus de place dans un seul octet, HP a dû utiliser deux octets pour certaines fonctions. Le premier octet de toutes les fonctions à deux octets a son digit de droite égal à F.

La deuxième table explicite toutes les fonctions légales à deux octets. Ces fonctions comportent donc 4 digits qui apparaissent de droite à gauche dans la table comme à l'affichage.

la première colonne à droite donne le premier octet de la fonction (celui de droite). Seuls sont légaux les octets AF à FF.

Le digit le plus à gauche se lit dans la dernière colonne de gauche, à chaque premier octet correspond un certain nombre de digits qui sont classés par ordre croissant dans chaque rangée correspondant à un premier octet particulier.

le digit intermédiaire est la valeur de l'argument. Les fonctions et les arguments légaux sont indiqués dans les cases quadrillées.

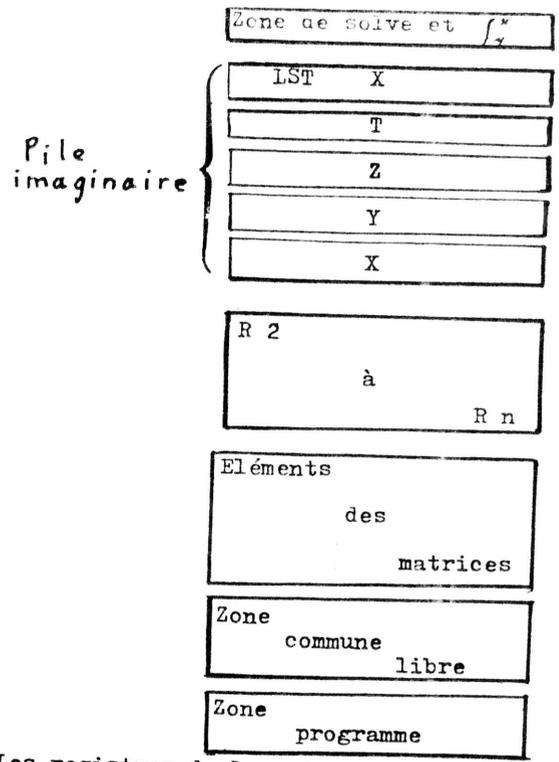
STO .9 est donc D9 DF
 DSE A est 7A EF

Il ya au total 562 fonctions programmables...

HP I5 C HP I5 C HP I5 C

Codage des adresses mémoire

D'une façon générale, la mémoire est divisée en registres de 7 octets chacun. Un registre contient soit 7 octets d'instructions programme, soit un nombre (registres de données, éléments de matrice). Pour que la machine puisse savoir ce qu'elle fait, il faut bien sûr qu'à chaque registre soit affecté une adresse. Dans la HP I5 cette adresse est composée de deux nombres hexadécimaux (en base 16, où l'on compte de 1 à F(15) ce qui fait au total 256 adresses possibles). Ainsi dans la matrice E 99.99 (pour son obtention, voir articles précédents), qui permet de visualiser la mémoire, on retrouve tous les 256 éléments les mêmes registres. Par exemple, le registre qui est en 1,1 est le même que celui qui est en 3,59 (256=2x99 +58 = 2 lignes et 58 éléments. Avant de décrire plus précisément le système d'adressage, rappelons l'organisation de la partie de la mémoire qui nous intéresse: Précisons bien la façon dont la zone programme se remplit. Au fur et à mesure que l'on écrit un programme, on "empiète" sur la zone commune libre. Le programme s'écrit donc en "remontant" et, à l'intérieur d'un programme, de la droite vers la gauche, alors que, par exemple, le registre de données 3 est après le 2, et s'écrit dans



Les registres de la matrice E 99,99 se lisent dans le sens des adresses croissantes, les registres des premiers pas de programme se liront en dernier et auront une adresse supérieure aux autres.

La numérotation des registres se fait "à rebours", à partir du registre où sont inscrits les premiers pas de programme, qui est toujours numéroté FF. Le précédent a l'adresse juste inférieure, c'est à dire FE, et ainsi de suite. Par exemple, si on utilise que sept octets de programme, si la zone commune est vide, si E est dimensionnée à 0,0 (ou Non normalisée, comme c'est en fait le cas) et si D contient 3 éléments quel sera l'adresse du premier élément de D ? Réponse : FC

Dans le registre d'état numéroté 22 (voir article précédent pour la numérotation des registres d'état visibles par l'intermédiaire de la matrice E 99,99), le 1er octet (quartet de signe + 1er quartet de la mantisse) indique en base 16 le nombre de registres utilisés avant R 2 (pile imaginaire, Solve, $\int \frac{x}{y}$).

L'octet suivant (en lisant de gauche à droite) contient l'adresse du 1er élément de la matrice A. On a ensuite la matrice Betc

L'adressage en zone programme se faisant sur une instruction, donc sur un octet, il est nécessaire d'affiner le système en rajoutant un troisième chiffre pour indiquer quel est l'octet du registre qui est adressé. Le chiffre est placé avant les deux autres et est compris entre 2 (pour l'octet "de droite", qui contient dans un registre de données les chiffres de l'exposant) et 7 (pour l'octet "de gauche", qui contient le quartet de signe et le premier chiffre de la mantisse)

C'est ainsi que l'on trouve dans le registre 24, à la place de l'exposant, l'adresse de l'octet pointé, et à la place du signe de l'exposant, le numéro de l'octet dans ce registre.

Pour les adresses de retours de sous programmes, qui sont placées dans les mantisses des registres 24 puis 23, le codage répond au même principe, à la seule différence que le chiffre des "Seizaines" est égal au chiffre des seizaines de l'adresse normale moins un, et que le numéro de l'octet est compris entre 2 (octet "de droite") à 7 (avant dernier octet "à gauche"). L'octet "de gauche" est noté comme le premier octet du registre suivant, et s'il n'y a pas d'octet suivant, il est codé OF0. Autre explication: l'adresse d'un octet dans ce code est celle, dans l'autre code de l'octet placé seize registres et un octet plus haut. (Ou II3 octets plus haut) Ou encore, pour traduire l'adresse XYZ du code sous programme au code normal il faut calculer X-I; Si X-I est égal à zéro, on remplace YZ par YZ-I et X par 7, ou bien par OF0 si on est dans le dernier

registre utilise pour le programme. Enfin, on remplace Y par Y-I. Pour tous ceux que ces explications fumeuses n'ont pas éclairé, voici quelques exemples:

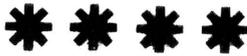
rang de l'octet dans le prgm	adresse normale	adresse pour un retours de ss-prgm
1 ^{er}	IPF	2EF
6 ^{eme}	6FF	7EF
7 ^{eme}	7FF	IEE
13 ^{eme}		
14 ^{eme}	6FE	7EE
15 ^{eme}	7FE	IED
20 ^{eme}	IFD	2ED
21 ^{eme}	6FD	7ED
22 ^{eme}	7FD	IEC
III ^{eme}	IFC	2EC
II2 ^{eme}	6FO	7EO
II3 ^{eme}	7FO	IDF
447 ^{eme}	IEF	2DF
448 ^{eme}	6CO	7BO
	7CO	OFO

Dans le cas d'une instruction à deux octet, les deux ont une adresse, l'appel de l'instruction se fait sur le premier octet des deux. Et enfin un petit calcul: de 00 à FF, cela fait 256 adresses possibles (en comptant 00 comme adresse). On a 64 registre de mémoire vive utilisables à volonté, plus R 0, R I, I, RANDOM, la pile, les registres d'état: environ une centaine de registres, Et les autres adresses? Sont elles bêtement vides? Remplissables? la machine pourrait-elle les gérer?.

Signalons que lorsque l'on remplace les vraies adresses des matrices par des fausses, on plonge la machine dans un drôle d'état: la réponse à MEM(demande de la partition mémoire) est Error IO, de même qu'a tout essai de changement de la partition mémoire (même dans le sens d'une diminution du nombre des registres alloués) Et les registres de données sont en fait des registres d'état (en général, R6 correspond à celui qui est numéroté IO)

Encore des explorations en perspective!

Bruno Piguet (T I52)



MICROCODE

Suite à l'article de Stéphane Barisien (programme CHARGE ---) j'ai tenté de comprendre la signification des deux premiers bits des deux premiers mots du nom de la fonction (de bas en haut), et je suis arrivé aux conclusions suivantes: mais ceci reste à confirmer car je ne possède ni MLDL ni RE. Je pense que les deux 1^{er} bits des deux 1^{er} mots sont étroitement liés et donnent :

1 ^{er} mot	2 ^e mot	signification possible (?)
1	0	1 tiret, ALPHA obligatoire, pas adressage indirecte (ex: ASN, COPY ...)
1	1	3 tirets, possibilité de BEX, pas IND (ex: DEL, SIZE)
1	2	Jamais rencontré ???
1	3	1 tiret, poss IND, (ex: ENG, TONE...)
2	0	2 tirets, poss de + - / X, d'adressage indirecte, de .(ST), seulement STO
2	1	2 tirets, .(ST), IND, (ex: RCL, ST+ ...)
2	2	2 tirets, IND, (ex: SF, ΣREG ...)
2	3	Jamais rencontré ???
3	0	LBL
3	1	XEQ
3	2	Jamais rencontré ???
3	3	GTO

1'après une étude du ROM interne.

T155 bruno

* * * * *

PPC-T No6
JUILLET AOUT
1983 P21

HP-41C

Programmer en microcode, pourquoi pas ?

POURQUOI ?

Je viens de faire un petit test qui me paraît significatif. Je dispose d'un programme en microcode qui fait la simple chose suivante : dans une boucle additionner 1 à chaque tour. Si une touche est pressée, formater le nombre ainsi obtenu pour pouvoir l'afficher, et l'afficher. Je lui ai comparé le programme classique suivant :

01 LBL "EDC"	Le END est là pour montrer que
02 LBL A	le programme a été "packé". Les
03 CLX	deux GTO 01 sont là pour vous
04 LBL 01	éviter le souci de savoir si
05 ISG X	le ISG saute ou non. Le programme
06 GTO 01	a tourné une fois avant l'essai
07 GTO 01	pour compiler les GTO. On peut
08 END	facilement faire plus court,

je ne pense pas qu'on puisse faire plus rapide. Noter qu'au moment de l'essai ma 41C était reliée à un grand nombre de modules, y compris une imprimante et qu'il ne faut donc pas comparer ces résultats à VOTRE machine. Simplement les deux essais étaient menés dans les mêmes conditions. J'ai fait pour chacun deux essais, je vous donne les moyennes. Pour 30 secondes de chrono la routine en microcode a compté jusqu'à 62786. Le programme ci-dessus a compté jusqu'à 272. Je n'ai pas voulu épiloguer en faisant un rapport, mais je maintiens d'un côté SOIXANTE DEUX MILLE SEPT CENT QUATRE-VINGT SIX et de l'autre DEUX CENT SOIXANTE DOUZE pour les mêmes 30 secondes.

Je ne vous donnerai pas la routine microcode ici, sachez qu'elle fait quand même 22 mots machine de long.

Cet exemple, même s'il est simpliste, montre bien un des avantages du microcode : il va VITE.

Son autre avantage est de permettre des fantaisies interdites autrement : ma machine n'a pas le même MEMORYLOST que la votre (ici prend place un pied de nez !), elle termine avec SIZE 026 (quelque soit le nombre de modules mémoire), FIX2 et notre chère virgule française.

Si tout ceci ne vous a pas donné envie de faire comme moi, c'est que vous n'êtes pas un "FAN".

COMMENT ?

Disons tout de suite que programmer en microcode une HP 41C nécessite une volonté certaine, quelques moyens financiers et surtout un coeur "gros comme ça !"

Pourtant, depuis peu, il est possible de faire tout cela sans sortir de notre territoire national pour ramener des éléments exotiques de lointaines contrées.

Pour résumer, il faut, en plus d'une 41C en état de marche, un certain matériel et une certaine documentation.

Renseignements préliminaires : NON il n'est pas possible de programmer votre 41C en microcode à partir du clavier et dans la mémoire programme. Pour une raison de moi inconnue HP utilise comme unité d'instruction machine un mot de 10 bits, au lieu de l'octet utilisé par tout le monde, HP Y compris le plus souvent.

Le programme EDC ci-dessus est composé d'instructions organisées autour d'une unité commune de HUIT bits (éléments binaires) formant un MOT de mémoire vive.

Tous les programmes en mémoire morte de la 41C utilisent l'unité de 10 bits des instructions machines. Vous ne le voyez pas, et pourtant c'est ainsi qu'elle tourne.

Quand les précurseurs du club PPC américain ont voulu programmer en microcode, ils ont dû résoudre le problème consistant à ranger des mots de 10 bits dans des rangements de 8 bits, constitués par des circuits courants dans le commerce et appelés EPROM.

Relisez plusieurs fois s'il le faut le paragraphe précédent. Comment ranger le contenu de 20 bouteilles d'un litre de vin dans des bouteilles de 75 centilitres ? n'oubliez pas qu'il y a des crus différents à ne pas mélanger !!!

L'éternel miracle de la vie informatique ayant fait son oeuvre, une solution a été trouvée : pour chaque litre de vin on va utiliser 2 petites bouteilles. Simple, non ?

Pour la HP, c'est pareil, on utilise, pour construire 1 module de mémoire morte, 2 circuits électroniques EPROM. Il ne reste plus qu'à construire une boîte de triage qui sera capable de reconstituer le module initial à partir des deux EPROM, et cela mot par mot.

A la programmation chaque mot d'instruction est coupé en deux, un morceau va dans une EPROM, un autre morceau va dans une autre EPROM. A la lecture un circuit reconstitue le mot initial à partir des morceaux.

C'est Jim de Arras (PPC N° 4706) qui, le premier a su réaliser ce travail et fabriquer l'appareil capable de ce travail que l'on appelle depuis l'EPROM BOX, en français Boîte d'EPROM ou lecteur d'EPROM.

Si vous êtes toujours avec nous c'est que le microcode vous intéresse vraiment, bravo !

une boîte d'EPROM est actuellement disponible au sein du club français PPC-Toulouse. Sa fabrication est aujourd'hui artisanale et réservée aux membres du club, mais le succès aidant une fabrication pourrait être envisagée par le concepteur, Didier Jehl (8116 T 80). Le prix de cet appareil est d'environ 800 F, donc du même ordre de grandeur qu'un module courant.

Mais attention, cet appareil est un lecteur, il ne peut pas programmer. Il ne peut qu'utiliser les eproms actuellement diffusées par le club.

PROGRAMMER SOI-MEME

Le seul moyen largement accessible actuellement pour programmer soi-même est d'acquérir un programmeur d'EPROM. Le seul modèle d'un prix accessible qui soit indépendant d'un ordinateur est un excellent petit appareil appelé SOFTY et distribué par Pentasonic au prix de 2250 F. Un jeu d'EPROM (une 2716 + une 2732) coûte entre 100 et 150 F selon la source d'approvisionnement, les EPROM préprogrammées diffusées au sein du club le sont au prix de 200 F.

Bien sûr un même lecteur d'EPROM peut utiliser successivement autant d'EPROM que l'on veut, ces appareils sont munis de connecteurs spéciaux qui permettent de changer les EPROM presque aussi facilement que les modules d'origine.

L'appareil lui-même est de l'encombrement de la 41C et relié à celle-ci sans aucune soudure par un câble plat et un connecteur dans un des ports de sortie.

ECRIRE DES PROGRAMMES MICROCODE :

Nous venons de voir le matériel nécessaire. Il y a également des appareils plus élaborés, mais ils ne sont pas actuellement disponibles en France, je n'en parlerai donc pas.

Programmer en microcode, c'est d'abord connaître à fond sa HP-41C. Pour cela le livre écrit par votre serviteur (au fond de la 41C) est à peu près indispensable.

Vous y trouverez tout le jeu d'instructions du microprocesseur (tout cela a été décodé par des utilisateurs de la machine, membres de PPC, en particulier Paul Lind (6157). Brutalement on peut considérer que cela suffit. En fait, il faudrait être vraiment très fort pour s'en contenter. Heureusement à cela vient s'ajouter une très volumineuse documentation accessible par l'intermédiaire (toujours, excusez m'en !) du club PPC-Toulouse. La documentation se compose essentiellement de la liste sur papier des programmes contenus dans tous les modules HP courants, à commencer par les modules de mémoire morte contenus dans la HP depuis l'origine. Cela représente près de mille pages de photocopies !

Hewlett Packard a été tellement surpris de voir le club déchiffrer et utiliser le microcode qu'ils nous ont fait cadeau de leurs propres notes sur le contenu des modules internes de la 41C. Pas toujours facile à déchiffrer, mais où serait le plaisir si c'était facile !!!

Ce qui rend l'élaboration des programmes en microcode difficile c'est l'absence totale d'aides à la programmation. En fait, nous avons bien des programmes conçus par le club, qui sont une aide précieuse, mais pas de mode trace, pas de possibilité d'insertion et d'effacement d'une ligne à moins de refaire une EPROM, pas de messages d'erreur, et le plus souvent Memory Lost au bout !

Et bien cette absence d'aides doit être retournée en bien et nous amener à programmer proprement du premier coup. Je me souviens d'avoir lu dans les livres d'Arsac qu'un bon programme doit tourner du premier coup. Combien d'entre nous respectent cette condition ? Et bien en microcode vous apprendrez à structurer une analyse descendante, à partir d'un problème bien défini pour arriver à un résultat bien défini, à découper votre problème en modules (réutilisables si possible) parfaitement analysés, et je suis persuadé que vous ferez des progrès spectaculaires même dans votre programmation de tous les jours.

EN GUISE DE CONCLUSION :

Il n'est pas possible pour un amateur d'écrire des programmes en microcode très longs. Le meilleur usage qui peut être fait du microcode est la création de nouvelles fonctions très puissantes, permettant des programmes très efficaces.

Un programmeur d'EPROM pour une seule personne est un jouet coûteux, acquis par un groupe d'une dizaine d'amis, cela devient un outil décisif. Pour un usage professionnel les possibilités sont illimitées.

MISE A JOUR

L'article ci-dessus avait été écrit il y a quelques mois pour une revue qui n'en a pas voulu. Depuis bien des choses se sont passées: Le lecteur d'eproms du club (enfin, celui de Didier Jehl) est moins cher, une version professionnelle est disponible à la SICAPE, elle coûte je crois 1300 F, mais peut-être gonflée jusqu'à 16 Koctets. Elle représente une excellente solution pour conserver des programmes professionnels d'usage fréquent.

L'appareil plus élaboré dont je ne voulais pas parler est maintenant disponible en France. La souscription lancée dans la dernière circulaire n'est valable que jusqu'à la fin de Juin et il sera sans doute trop tard quand vous lirez ces lignes, mais en Septembre la construction sera reprise, avec

cependant un prix plus élevé, de l'ordre de 2500 F. Mais cet appareil constitue à la fois un lecteur d'EPROM et une mémoire vive qui fait également 4 Koctets, donc au total 8 Koctets de place disponible. Surtout c'est un outil prodigieux pour la programmation en langage machine.

J-D Dodin (7226 T 1)

* *

2 270

Jerge Vaudenay
20 av Arrouquin B3
94420 Le Plessis Trévise

le 11/01/83

bonjour,

J'aimerais savoir si il serait possible de se procurer le listing des ROM de la 41C (l'adresse 0, 1 et 2 (!!!)). Ma demande paraît énorme (voir calcul ci-dessus) mais je crois connaître un moyen pour que tous vos lecteurs y tirent avantage: en consacrant six pages de listing par journaux bimensuels et en les complétant par un article expliquant le fonctionnement de la machine.

Mais parlons chiffres. Nous supposons que chaque module va jusqu'à la ligne FF en hexal soit 4096 lignes. En prenant comme unité le paragraphe (10 lignes), cela fait 256 pages. En regardant la page 22 de "Au fond de la HP 41C", on voit qu'une page peut contenir 4 colonnes de 2 pages facilement. Cela ramène à 128 colonnes ou 32 pages.

Un module (4Koctets) aura un listing de 32 pages. Les 3 modules (12Koctets) feront 96 pages (!!!). Mais 4Koctets font 3 273,6 lignes de 10 bits au lieu de 4 096, ce qui fait 76,8 pages. A raison de 6 pages par journaux, la HP 41C aura livré ses secrets en 2 ans...

Cette collection de listings sera sans doute une collection unique. Le "dérivage" et l'interprétation de ces listings seront fastidieux et devront sans doute être fait à plusieurs. Je ne pourrais volontairement pour effectuer le déasseblage (à partir du listing des codes des instructions car ce n'est pas le matériel pour travailler en microcode). Les travaux pourraient commencer dès la parution du n°5 avec l'accord de tout le monde.

En souhaitant une plus grande croissance des inscriptions ...

Amicablement

A votre avis, qu'est-ce qu'il y a dans les photocopies "Microcode PPC"? la lecture des anciens numéros vous aurait renseigné...

* *

La page suivante est très importante, c'est pourquoi je la passe tel que, je ne suis pas capable de traduire l'allemand, mais il est facile de voir qu'il s'agit du projet d'un ROM custom, c'est à dire fabriqué par HP pour un particulier et prévu pour contenir des instructions "club". Si quelqu'un peut m'en faire un résumé en français il passera dans le prochain numéro.

Les remarques sont à faire à Wolfgang Baltes Gardinistr. D 8000 MUNCHEN 70, RFA. Vous pouvez écrire ou téléphoner en allemand, anglais ou français (!).

* *

Jean Claude BORNES
PPC 9250 / T 175
10, Allée de la Brie
LISSES
91000 EVRY

Lisses, le 22 avril 1983

Cher Jean Daniel

J' ai beaucoup tardé à envoyer cette lettre, mais pour installer la machine, il fallait que je range mon MLDL.

J'ai bien reçu le jeu d'EPROM. Malheureusement, c'est lui qui a été la source de mes soucis. Je décris donc mes problèmes:

CCD-ROM-Projekt

Heute möchte ich Euch ein Projekt vorstellen, um das ein paar andere Mitglieder und die Unterzeichner sich schon seit einiger Zeit Gedanken gemacht haben: die Erstellung eines Moduls für den HP-41 mit in Maschinensprache programmierten Funktionen.

Während des Ende Februar in Frankfurt stattgefundenen "1. Personal Computer Symposiums" von Hewlett-Packard, auf dem unser Club einen Stand hatte, bot sich reichlich Gelegenheit, mit den verschiedensten Leuten von HP zu sprechen. Dabei wurde uns zugesichert, die Anfrage in den USA, ob und in welcher Form HP ein solches Modul produzieren will, zu unterstützen und desweiteren uns eventuell mit weiteren Unterlagen über den HP-41 die Arbeit zu erleichtern.

Am Samstag, den 26. Februar 1983 fand in Glashütten ein Treffen zwischen Wolfgang Baltes (155), Roland Freytag (1064), Ulrich Jansen (283), Peter Kiefer (958), Wilfried Kötz (?), Andreas Marktscheffel (69) und Ralf Mulch (886) statt, wobei eine erste Liste von Funktionen zusammengestellt wurde, um das Projekt zu umreißen.

Hier nun, was dabei herauskam:

Systemroutinen:

CATALOG - Anzeige der eingesteckten Module, Auflistung der Funktionen eines einzelnen Modules

SAVE STACK - Temporäres Abspeichern der Stackregister

RECALL STACK - Zurückholen der abgespeicherten Stackregister

SWAPSTK - Vertauschen des aktuellen mit einem vorher abgespeicherten Stack

PASSWORD - Rechner nur nach Eingabe einer (selbstdefinierten) Zeichenfolge nutzbar

CRASH RECOVER - Erzeugen eines definierten Rechnerstatus, Retten von 'verlorenen' Programmen

SWAPKA - Vertauschen zweier Blöcke von Tastenzuordnungen

DEFAULTKA - Tastenzuordnungen für die oberen beiden Tastenreihen aktivieren/deaktivieren

PACKKA - Packen der Tastenzuordnungsregister (eventuell automatisch beim Einschalten des Rechners)

FREE - Ausgabe der Anzahl der noch unbenutzten Register

RUN - Ausführen eines vordefinierten Programmes, das dabei entweder im Hauptspeicher oder im 'Erweiterten Speicher' (X-Memory) stehen.

ALPHAroutinen:

Generell soll ein besonderer Alphamodus implementiert werden der Art, daß bei zusätzlich gesetztem USER-Modus ungeschiftet alle Kleinbuchstaben und geschiftet weitere Sonderzeichen vorhanden sind. Dieser Zustand wird bei eingestecktem Modul immer vorhanden sein, so daß er sowohl beim Programmieren als auch bei der üblichen Eingabe in das ALPHA-register aktiv ist. Dies erlaubt zum Beispiel die Eingabe von Kleinbuchstaben in Programmtexten ohne jede Schwierigkeit.

OUTREC - Ausgabe eines ASCII-Records (X-Functions)

EDIT - Direktes Editieren eines ASCII-Files

PRA - Print Alpha mit automatischem Unterstreichen beziehungsweise beim Video-Interface in Invers-Video-Darstellung

ARCLINT - ARCL X im Format FIX 0, ohne Dezimalpunkt, jedoch ohne den aktuellen Displaystatus wirklich zu verändern

ARCLHMS - ARCL X im Format Grad/Min/Sek.

Mathematische Routinen:

XYZCW - Zusammenfassen der drei Stackregister X, Y und Z in das Format der Standardisierten Kontrollzahl xxxx.yyyzz

CWXYZ - Zerlegen der Standardisierten Kontrollzahl in die drei Komponenten

D-H - Zahlenkonvertierung von der Basis 10 zur Basis 16

H-D - Zahlenkonvertierung von der Basis 16 zur Basis 10

D-B - Zahlenkonvertierung von der Basis 10 zur Basis 2

B-D - Zahlenkonvertierung von der Basis 2 zur Basis 10

REC-SPH - Koordinatentransformation von rechtwinklig nach sphärisch

SPH-REC - Koordinatentransformation von sphärisch nach rechtwinklig

EULER - Euler-, Kardanwinkelberechnung

HMSRND - Runden im Sexagesimalsystem (hh.mmss)

SIGMA - Recall der einzelnen Statistikregister

LINREG - Lineare Regression

SOLVE - Kernroutine für Nullstellenprogramme

INTEGRATE - Kernroutine für Integrationsprogramme

HORNER - Funktionsberechnung eines Polynoms

SUM - Summation von Funktionswerten

MATRIX - Ein-/Ausgabe, +, -, x, /, Pivotisierung, Umwandlung zwischen Index- und Registeradressierung

X-Memory-Routinen:

RENAME - Umbenennen eines Files

COMPRES - Verkürzen eines Daten- oder ASCII-Files

EXPAND - Verlängern eines Daten- oder ASCII-Files

CHANGE - Verändern eines Dateityps

IL-Routinen:

SLCTPR - Automatisches Selektieren eines Druckers

SLCTVI - Automatisches Selektieren eines Video-Gerätes

Spezielle Routinen:

LB - Editieren einzelner Bytes im HEXcode oder als ALPHAzeichen

KA - Zuordnen von beliebigen Codes auf eine Taste

PEEK - Liefert den Wert eines Bytes aus einer absoluten Adresse

POKE - Speichert ein Byte in eine absolute Adresse

PBG - Schnelle Meßdatenerfassung mit dem PBG-Gerät

RCLABS - Nichtnormalisierender Rückruf aus einer wahlweise absoluten oder relativen Adresse

STOABS - Speicher in ein absolut adressiertes Register

CODE - Erzeugen eines beliebigen HEXcodes

DECODE - Darstellen eines Registerinhaltes im HEXcode

AND - Bool'sche Funktion auf die 56 Bit von X- und Y-Register

OR - Bool'sche Funktion auf die 56 Bit von X- und Y-Register

XOR - Bool'sche Funktion auf die 56 Bit von X- und Y-Register

NOT - Bool'sche Funktion auf die 56 Bit von X- und Y-Register

Natürlich ist das nur eine vorläufige Liste. Das Modul soll so anwenderfreundlich wie möglich werden; so werden Druckfunktionen sowohl mit den drei Druckern als auch mit dem Video-Interface kompatibel sein; Fehlermeldungen sollen möglichst genau auf die Ursache hinweisen. Dieses Anforderungen und die Fülle der Funktionen sind so speicherplatzintensiv, daß ein 8k-Modul, wenn nicht sogar ein 12k-Modul erwogen wird. In letzterem Falle würde auch ein Systembereich des Rechners, nämlich der des Service-Moduls, mit in Anspruch genommen, wodurch sich weitere zusätzliche Möglichkeiten eröffnen. So kann damit ein wirkungsvoller Schutz des Rechners gegen unbefugtes Benutzen realisiert werden. Insgesamt können in einem solchen Modul bis zu 127 Funktionen untergebracht werden. Um ein möglichst nutzbringendes Modul entwerfen zu können, bitten wir hiermit jeden einzelnen von Euch, Vorschläge und Wünsche der zu implementierenden Funktionen an

Ulrich Jansen
Mühlenstraße 16
5140 Erkelenz

einzusenden. Wohlgedacht: Es dreht sich dabei nur um die Artikulierung von Zielvorstellungen; ob und wie diese zu realisieren sind, wird dann in einem daran anschließenden Schritt von den Mikrocode-Programmierern des Moduls erarbeitet. Ideal wäre, wenn Eure Vorschläge Folgendes beinhalten würden:

- Art der Funktion; was soll die Funktion bewirken

- Eingabeparameter

- Ausgabeparameter

- vorgesehene Fehlermeldungen

- ein Beispiel (in normalem HP-41-Tastencode) zum vorgesehenen Gebrauch der vorgeschlagenen Funktionen

Dabei sollte beachtet werden, daß wegen der Mikrocode-Programmierung dieser Routinen damit dem HP-41 auch ganz neue, ungewöhnliche Eigenschaften und Fähigkeiten entlockt werden können. Mit anderen Worten: durch Mikrocode-Routinen kann auch bislang Unmögliches auf dem HP-41 bewirkt werden.

Die Details zu unserem neuen ROM-Projekt sind noch nicht geklärt; wir werden Euch im PRISMA darüber auf dem Laufenden halten. Sicher ist nur, daß das Gelingen eines solch aufwendigen Projektes von Eurer Mitarbeit abhängt und von Eurem Interesse getragen sein muß. Deshalb hier nochmals unsere Bitte: Artikuliert Euch; entweder an Ulrich (Anschrift siehe oben) oder an Wolfgang (Telefon 089/7001540 ab 19 Uhr).

Wolfgang Baltes (155)
Ulrich Jansen (283)
Andreas Marktscheffel (69)

END.

Cas 1: Lors d'un Catalog 2, la machine se plante en arrivant à X<ROM qui n'est pas listé. Le listing en SST montre des affichages fantaisistes OF JO dans le port 2, NY= dans le port 4, d dans le port 3, ces anomalies toujours à la place des mêmes noms, la machine se plantant en arrivant à CODE. Si je la fais repartir sans MEMORY LOST, elle se plante à DE, PPACK, PKEY. quand je repart après CODE, la plupart du temps l'indicateur PGRM est mis.

Cas 2: Lors d'un CAT 2, la machine affiche OF JO (OF JO sur l'imprimante) ou autre, se plantant en fin de Catalogue, la dernière fonction (X<ROM) ne s'affichant pas.

Cas 3: Le CAT 2 se déroule normalement, la machine se plantant sans afficher la dernière fonction.

* En résumant, quelques cas de figure:

Circuit à l'envers pour les mesures, cas 1

Circuit soulevé, cas 2.

Circuit à l'endroit, cas 3, sauf des cas erratiques où j'ai pu voir un catalogue complet!

Je passe sur les nombreuses mesures concernant tous les circuits, les échanges, le changement de modèle des condensateurs de filtrage etc... sans aucun résultat, sans trouver une explication, tous les tests donnant les résultats attendus.

Avant de lire la suite, je suggère de marquer un temps d'arrêt et de réflexion sur les causes possibles car j'ai enfin trouvé...

* La réponse à tous mes problèmes qui n'apporte pas la solution est que ces EPROM ont besoin pour fonctionner, de lumière. Oui, de lumière! La 2716 en veut plus que la 2732. La 2716 cause le crash en fin de catalogue, la 2732 causant les autres troubles, cas 1 ou 2 et j'ai:

Pas de lumière ou très peu, cas 1. Un peu plus, cas 2. Encore un peu plus, cas 3 (assez de lumière pour la 2732). Enfin suffisamment de lumière pour la 2716 (lampe de 60 W à 50 cm), fonctionnement normal... enfin presque!

* Ce qui reste, car il reste des problèmes:

- XCAT ne marche que pour les ROM d'adresse inférieure à celle de l'EPROM. Pour l'EPROM et les ROM suivants, la machine renvoie le numéro sans rien faire d'autre (elle répond normalement NONEXISTANT quand il faut)

- Quand je place une pause trop près d'une fonction de l'EPROM la machine fait une pause infinie, la distance à respecter dépendant de la fonction (distance nulle pour certaines).

De toutes façons; comme la fourniture de lumière ne fait pas partie des spécifications, ces EPROM ont l'air (!) déf ectueuses.

Côté projets, le programmeur d'EPROM a bien avancé.. sur le papier. Programmation directe des 10 bits (2716 et 2732 en parallèle) par une instruction WRITE (la fonction X<ROM suivie d'une boucle de temporisation) et EPROM directement utilisable sans la bouger (liaison bidirectionnelle), le test de la routine est immédiat. La RAM est presque inutile!

Je vous tiendrais au courant. Pour ceux qui veulent monter un MLDL j'ai trouvé des 6504 (de 200 ns excusez du peu!) à 27 F TTC. Je n'ai trouvé ni 6116 ni 6132 à prix Honnête.

Pour en revenir à mon problème avec mes EPROM, que puis-je faire? Moi qui avais prévu un double fond dans ma malette pour y loger le MLDL !

Enfin, il existe des 27C16 et 27C32, au moins chez N.S. cela éviterait d'envoyer des signaux à des circuits non alimentés (qui semblent bien le supporter, contrairement aux CMOS).

Les Eproms sont des animaux capricieux, en particulier le niveau de lecture dépend de l'appareil lecteur. Mon programmeur vérifie chaque jeu d'eproms, je les "brûle" deux fois, je les essaie sur la machine avant de les envoyer, mais il y a encore parfois des problèmes. N'hésitez pas à m'en parler.

LISSES, le 4 mai 1983

AHHHHH..!

Cher Jean Daniel!

J'avais terminé le câblage de la partie RAM de mon MLDL hier soir ... et constatais un très mauvais résultat. En particulier avec des MEMORY LOST (volontaires) qui se plantaient avec l'affichage lassique au lieu d'afficher le SIZE 026 attendu, les essais n'étaient pas facilités. Après une nuit de repos, les résultats n'étaient pas meilleurs ce matin, avec des plantages itératifs, même avec la RAM inhibée.

Et puis le facteur m'a apporté un petit paquet contenant le jeu d'EPROM. Je me rue sur ma machine et... miracle, tout marche!

Je fais un CAT 2 (sans lumière!) OK, un memory lost OK, un 12 XCAT OK, et tout ça en ayant fermé la boîte! Bref formidable!

J'ai aussitôt purgé la RAM, enregistré ce qu'il faut, avec une petite routine, et ça marche (après des problèmes de n° XROM mal choisis, la hâte m'ayant empêché de vérifier).

Un grand merci, donc, mes problèmes étant réglés. Je vais pouvoir monter la partie programmeur d'EPROM, travailler sur un assembleur-éditeur, et finir mes capteurs optiques (pour réaliser une base de chronométrage avec deux barrières optiques branchées sur le port d'entrée et le module TIME) la maquette me donne plus de trois mètres de portée, un temps de commutation de 200µs c'est long mais ça devrait aller) et des interruptions de faisceau de moins de 2 ms détectables. Pour un chrono au 1/100 de s, ça devrait marcher...

Après la lecture du journal, je me suis précipité à la FNAC.. qui n'avait plus rien et vendait le module HP-IL 1160 F!

Jene me suis pas découragé et suis allé à la règle à calcul et y ai acheté un XM à 270 F et l' HP-IL à 270 F et OUI! je n'ai plus qu'a trouver un inter face vidéo... dans mes prix, et un kit convertisseur au bon prix!

Bref je replonge de ce pas dans le MCODE avec un grand merci pour une réponse si rapide et si satisfaisante pour moi!

Lisses, le 25 mai 1983

Cher Jean Daniel

D'abord des nouvelles de mon MLDL qui peuvent intéresser ceux qui en ont fait un et ceux qui veulent en faire un.

Tout d'abord une négligence dans le schéma fait que les entrées des mémoires vives se retrouvent à certains moments "voyant" des sorties haute impédance, avec pour résultat une consommation faramineuse (plusieurs dizaines de milliampères) et des séquelles dures quand j'ai laissé le circuit au repos pendant 24 heures (avant de m'apercevoir de ce défaut) En effet, le lendemain, le circuit était complètement bloqué. Après la lettre enthousiaste que j'avais envoyée, on devine mon désespoir! Je passe sur les étapes et donne la solution.

Pour régler le problème des sorties haute impédance, il faut relier U6,15 et U7,15 au niveau haut (V+) quand on utilise des 6504 x 10. Dans le cas de la modification décrite par Chris Rath dans PPCCJ V9N4p25 et reproduite par Didier Jehl dans PPC-T4p28 et 29, il suffit de relier DL8 et 9 à la masse par une résistance tire-bas, mais à 40µA par résistance vous êtes prévenus! Une solution satisfaisante est plus complexe à trouver et je ne peux de toutes façons pas l'expérimenter.

Mes 6504 sont des Harris de 200 ns (B6). D'autres modèles peuvent très bien supporter les choses.

Ce n'est pas tout! Les EPROMS ne supportent pas, comme on pouvait s'y attendre, de recevoir des signaux sans être alimentées. Dans ces conditions, la broche 20 (OE) de ma 2732 consomme 100µA et ne répond plus de rien si je laisse trainer. La solution: valider en permanence les sorties en reliant OE et CE de la 2716 et de la 2732 à la masse, en tamponnant les sorties avec deux 4503 qui reçoivent, eux, le signal de validation PROMO*. Seules les sorties de la 2732 doivent être tamponnées, celles de la 2716 étant tamponnées par U8, le 4052.

Ces modifications faites, mon circuit consomme environ 1µA au repos, ce qui est correct et laisse du temps pour changer les piles (tous sélecteurs en position ouverte), opération rare avec une autonomie de ... 100 ans! Ceci pour dire que seule compte la durée de vie en stock dans le choix des piles.

Le défaut de consommation du MLDL est semble-t-il un défaut d'origine, je ne suis pas compétent pour apprécier la valeur de la modification que tu propose, mais si ça marche pour toi... Je n'ai jamais eu de problème d'usure de piles. Il faut dire que je sauve tout sur cassette. Je me contente de débrancher l'appareil quand je ne m'en sers pas.

* *

MLDL-MISE EN ROUTE

Dans toute cette étude, nous considérerons le MLDL comme posé devant vous les eproms en haut (reconnaisables à la fenêtre qu'elles ont sur le dessus, le commutateur gauche).

Dans ce cas l'eprom 2732 doit être placée à droite et la 2716 à gauche, leur repère en u en haut.

Le cordon de liaison est placé dans le support en bas (celui du milieu s'il y en a plusieurs).

Pour fonctionner les eproms et le cordon de liaison doivent être en place et l'ensemble branché sur un port quelconque de la 41C.

Il faut savoir que: l'adresse des eproms dépend du port dans lequel le MLDL est branché et elles se trouvent à l'adresse haute de celui-ci (par exemple adresse F dans le port 4).

canaux. Nous les numérotions de 1 à 6 et de haut en bas, quelle que soit l'inscription qui est dessus.

- l'interrupteur n°1 (en haut) permet l'utilisation éventuelle d'une eprom 2732 à la place d'une 2716. Il est normalement OFF.
- l'interrupteur n°2 valide ou non la lecture de la mémoire vive. Quand il est sur OFF on peut y écrire mais pas la lire. Au premier branchement, si il n'y a pas eu d'initialisation, il doit être sur OFF.
- les autres interrupteurs donnent l'adresse de la mémoire vive en binaire, le bit le plus significatif en haut, le moins significatif en bas. L'adresse 8 s'écrit donc de haut en bas ON OFF OFF OFF = 1 0 0 0 faites attention de ne pas utiliser une adresse déjà occupée!

Mise en route:

Il vaut toujours mieux commencer avec une machine nue, sans aucun autre périphérique.

les branchements faits, exécutez "ØRAM" ce qui nettoie tout le contenu de la mémoire vive, aléatoire à ce stade. Puis placer l'interrupteur 2 sur ON. C'est prêt.

Remarque: même la HP sur OFF, le MLDL consomme du courant. Le débrancher quand on ne s'en sert pas est prudent. L'utilisation pour le maintien de la mémoire de piles de 3V fait qu'elles s'usent peu. Pour changer les piles brancher le MLDL sur la 41C (éteinte), celle ci prendra le relai (facultatif).

N'oubliez pas, quand vous travaillez avec le MLDL que vous devez tenir à jour la partie catalogue (n°XROM, nombre de fonctions, adresse du point d'entrée, etc...)



STANDARD

Dans PPCJ V9 N3 Avril 1982 Paul Lind(6157) et Lynn Wilkins (7344) ont proposé un standard pour l'échange des programmes en microcode sur cartes magnétiques ou cassettes.

Il s'agit de ranger 5 mots de 10 bits dans un seul registre de données, sous forme de valeur alphanumérique de la façon suivante

|0001xx|mot n|mot n+1|mot n+2|mot n+3|mot n+4|

xx = n'importe quoi, les mots sont rangés dans le sens normal de lecture, bit le plus significatif à gauche.

Le premier registre du fichier ainsi constitué est un chapeau qui donne les informations sur le fichier:
digits: 1000RRRAAANN

RRR= nombre total de registres dans le fichier, chapeau compris

AAAA= adresse originale du premier mot du fichier)

NNN nombre de mots moins 1 (ajouté à AAAA donne l'adresse du dernier mot du fichier

000 est libre pour l'instant. Il serait très intéressant d'y mettre l'adresse du point d'entrée (par exemple en nombre de mots par rapport au début du fichier, 1 seul digit serait alors suffisant. Ceci permettrait d'écrire une routine en microcode qui, d'une part mettrait sur cartes une routine microcode à partir de son nom (donné en alpha ou prompté), il suffirait de convenir d'un code de fin de routine (genre END à mettre comme dernier mot de la routine), et de permettre à la relecture la mise en place de la routine à la suite du contenu précédent du MLDL, avec mise à jour du catalogue automatique. Cela reste à écrire.

Par contre les routines mettant en application le standard de Lind et de Wilkins ont été données par ceux-ci et figurent dans le MLDL Operating System.



Différences entre MLDL et Protocoder

Ces deux versions de simulateur sont disponibles en différentes versions. Leur différence essentielle est le mode d'écriture des instructions dans le simulateur (mémoire vive).

Le MLDL tel que décrit pour la première fois dans PPCJ par Lynn Wilkins et Paul Lind utilise une instruction inutilisée du microprocesseur de la 41C (un "NOP", le code 040) comme instruction de déclenchement du MLDL. Il faut donc pour utiliser cet appareil avoir disponible cette instruction, en général dans un lecteur d'eproms, d'où la nécessité d'ajouter ce lecteur d'eproms au système ce qui accroît son coût. Mais ce lecteur peut contenir bien plus que cette seule instruction.

Le Protocoder utilise une instruction peu utilisée de la HP (SIGN) pour activer la RAM. Il n'est alors plus besoin de lecteur d'eproms, mais gare aux réactions si un SIGN passe dans le programme. Les accidents sont évités par le contrôle

d'un bit particulier qui doit être présent en X et qui ne peut exister que dans un Nombre Non Normalisé.

Chaque système a ses partisans.



LE MLDL OPERATING SYSTEM

Les eproms MLDL OS livrées avec le MLDL de PPC-T ont un numéro XROM de 04 (celui du module finance, attention!).

Elles ont été mises en place par Didier Jehl et Janick Taillandier.

Les fonctions de BCDBIN à DECODE, ainsi que DIS, XCAT, sont celles de TOULROM (avec des numéros XROM différents, attention!).

Elles ont été décrites dans PPC-T.

MNEM, DISASM, PROMT, "LIST" fournissent un désassembleur en microcode avec, hélas,

les mnémoniques HP et sont dues à Nelson Crowle, le fabriquant du Protocoder. Le nom "LIST" est très mal choisi et il faut être astucieux pour utiliser la fonction. En effet elle nécessite l'imprimante et XEQ "LIST" vous branche sur le programme standard de l'imprimante. Il faut: 1) débrancher l'imprimante; 2) brancher le MLDL; 3) faire ASN "LIST" xx. On peut ensuite rebrancher l'imprimante et utiliser la touche assignée pour obtenir LIST - - - . Répondre en pressant les touches qui donnent en hexadécimal l'adresse où doit commencer la liste (ne pas passer par alpha).

ØRAM efface complètement le MLDL (partie RAM).

"LOAD" permet de remplir la RAM du MLDL. XEQ "LOAD" vous demande l'adresse de début de chargement en hexadécimal (presser simplement les touches 0-9 et A-F sans autre formalité). La HP affiche alors l'adresse suivie du code préexistant à cette adresse, puis trois tirets qui vous invitent à fournir un code. Ce code fourni (code 244) l'adresse suivante est affichée de la même façon.

Voyons maintenant les instructions fournies à l'origine par l'auteur du MLDL, Lynn Wilkins (7344) et publiées dans PPCJ (US) V9N3 Avril 1982

ROM>X Etant donnée une adresse hexadécimale dans les 16 bits de droite de X (obtenue par CODE ou "HN") retourne en X les 10 bits du mot du ROM. L'adresse est présente en L incrémentée de 1.

X>ROM Etant donnée en X une adresse de 16 bits + un mot de 10 bits (le tout obtenu par CODE), écrit le code à l'adresse indiquée. Pas d'incrément en L.

X>A Met en alpha (à gauche) le caractère dont le code est en X

ROM>REG Crée un fichier de données dans les registres de la 41C au format indiqué plus haut.

Le numéro de premier registre doit être en x et un nombre hexadécimal BBBBEEEE où BBBB est l'adresse du premier mot du microcode et EEEE l'adresse du dernier. En fin de travail Last X contient le numéro du dernier registre utilisé, le reste de la pile n'est pas modifié. S'il n'y a pas assez de registres disponibles le programme s'arrête sur NONEXISTENT.

REG>ROM Met dans le MLDL le contenu des registres. Avant l'exécution, X doit contenir le numéro du premier registre de données, Y contenant l'adresse de chargement du microcode:

- si y=000 le fichier est chargé à l'adresse d'où il a été lu
- si y = 0001 à 000F, le fichier est chargé dans la page dont le numéro est ainsi donné, à son emplacement d'origine.
- si y = 0010 à FFFF, le fichier est chargé à partir de cette adresse.

Le catalogue du ROM, les XEQ et GTO sont à adapter par vous même.



Cher M. DCBIN

Dans le dernier numéro du journal j'ai lu que Jean-Christophe BENOIST a écrit un programme sur TRS 80 pour simuler le microprocesseur de la HP. J'ai essayé de faire la même chose sauf que j'ai essayé directement sur la 41. D'où le prgm qui accompagne cette lettre. J'espère que vous pourrez l'imprimer et le publier car il tient tout juste dans ma 41 munie de 2 modules de mev. Par cette voie il pourrait être amélioré et ré-

... qui permettant ainsi d'inclure d'autres ordres du microprocesseur.

Pour l'instant en effet c'est pas tellement l'idéal pour construire ses routines en micro-code car voici ses (nombreuses) limitations:

Il ne peut manier que les registres suivants du microprocesseur C A B M N P Q R ST et le carry qui correspondent aux mémoires 09 à 18 de la machine. Les mémoires 00 à 04 représentent la pile dans l'ordre: T Z Y X L. Quant aux mémoires 05 à 08 elles étaient censées représenter le registre Alpha mais malheureusement un AVIEW ne les montre pas.

L'éditeur est des plus simples: SST BST on remplace un pas par un autre par écrasement il n'y a pas d'insertion.

La mémoire programme est limitée à 50 pas qui sont en fait les mémoires 20 à 70.

Tous les ordres du microprocesseur ne sont pas disponibles, loin de là, je les détaillerai après vous avoir expliqué le fonctionnement du programme.

- 1) on fait XEQ "MICRO-C" et on voit MOTS ?
- 2) on rentre alors un mot en code 442 et on fait R/S= même message, on tape le second mot etc...
- 3) lorsque le programme est entièrement tapé on appuie sur R/S sans rien répondre à l'interrogation et le programme s'exécute.
- 4) au bout d'un temps plus ou moins long on peut regarder l'effet du prgm sur les registres que j'ai décrits plus haut.

5) si on veut faire défiler le prgm on tape XEQ "LIST" et on voit le numéro du pas ainsi que le code qu'il contient.

Pour avancer: XEQ "PAV"

POUR reculer: XEQ "PAR"

Et voilà pour le fonctionnement. Voyons maintenant le décorticage.

LE PROGRAMME MICRO-C

1-3: initialisation

4-13: introduction des codes 442

14-18: mise en place du "pointeur" dans le registre 19

19-44: décomposition du code courant et stockage dans R71 R72 et R73

45-51: incrémentation du pointeur, si on n'est pas à la fin du prgm on saute au LBL 02 de la ligne 19, sinon on affiche TERMINE pour faire savoir à l'utilisateur que le boulot est fini.

LE PROGRAMME Ø

Il interprète les instructions dont les 2 bits de droite sont à zéro.

2-3: on traduit le paramètre en décimal.

5-6: on décode l'instruction.

7: on exécute l'instruction, le paramètre est en R73.

8: c'est fini on revient au prgm MICRO-C pour l'instruction suivante.

à partir de la ligne 9 ce sont les différentes fonctions de type Ø que le prgm comprend.

LE PROGRAMME 1

Il devait interpréter les fonctions de type 1 mais le manque de place fait qu'il se compose uniquement du label global.

LE PROGRAMME 2

Il interprète les fonctions de type 2.

1-7: on traduit le paramètre on le range en R73

et on fait la même chose pour l'instruction

8-18: on récupère le 5^{ème} bit de l'instruction et on retraduit en décimal.

19-27: on spécifie le champ mais faute de place il sera toujours de 3 c'est à dire ALL.

28: on exécute l'instruction.

29: on revient à MICRO-C

à partir de la ligne 30 ce sont les instructions de type 2.

LE PROGRAMME 3

Il s'occupe des sauts relatifs.

1-7: on traduit en décimal bien sûr.

8-27: on calcule la longueur du saut.

28-30: on détermine la condition: JNC ou JC.

31: on VA au label ainsi déterminé.

32-40: cas où C=0. On vérifie que la condition est réalisée.

si oui on laisse la longueur du saut intacte en R71.

sinon on la met à zéro.

dans les 2 cas on saute à l'étiquette 2.

41-48: cas où C=1. ON agit de la même façon.

49-54: on rappelle la valeur du saut.

si F15 est baissé il s'agit d'un saut positif on ajoute R71 à R19.

si F15 est levé on le baisse et on retranche R71 de R19.

55: on retourne à MICRO-C.

LE PROGRAMME LC

Il traduit un caractère alphabétique en décimal.

LE PROGRAMME LIST

Il montre la 1^{ère} ligne du prgm entré.

LE PROGRAMME PAV

Il montre la ligne suivante.

LE PROGRAMME PAR

Il montre la ligne précédente.

Voyons maintenant les ordres compris ORDRES DE TYPE Ø

NOP

CLRF- SETF- ?FSET

?R=

CATEGORIE 6/0

Tous sauf T=ST ST=T ST()T Méfiance cependant car les flags ne sont pas levés en conséquence.

R=

WRIT

READ

CATEGORIE 8/0

SLCT P SLCT Q ?P=Q A=B=C=0

CATEGORIE C/0

N=C C=N C()N

ORDRES DE TYPE 2

Tous sauf RSHFA RSHFB RSHFC LSHFA

ORDRES DE TYPE 3

Tous

Et voilà vous savez tout. Pendant que j'y suis je voudrais poser une question:

Pourrait-on créer un module style Maths PPC-Rom ou autre grâce auquel on pourrait programmer en microcode soit dans la HP en lui faisant croire que sa MEV est une MEM soit dans le module lui-même qui serait comme un MLDL ?

Où peut-on se procurer un labo pas trop cher pour un étudiant ?

Dans PPCT N° 4 les dernières pages en décrivent un: peut-on se le procurer dans le club ? Si oui à quel prix ?

Merci puor tout d'avance

Nolin Patrice T263

01*LBL "MLRU"	88 20	172 RCL 73	257 STO 11	338*LBL 24	428 FS?C 15	442 X?Y?	456 "F"	478 GTO 00
02 20	89 +	173 60	258 RTN	339 RCL 09	429 ST- 19	443 RTN	457 ARCL IND 71	
03 AOH	90 XEQ IND Y	174 +		340 RCL 10		444 1	458 PROMPT	471*LBL I
	91 RTN	175 XEQ IND X	259*LBL 00	341 X?Y?	430*LBL "LC"	445 -		472 AOH
04*LBL 00		176 RTN	260 RCL 09	342 SF 15	431 SF 25		459*LBL "PAY"	473 STOP
05 "MOT ?"	92*LBL 25		261 STO 10	343 0	432 CLA	446*LBL "LIST"	460 ISG 72	474 AOFF
06 CF 23	93 RCL 09	177*LBL 61	262 RTN	344 FS?C 15	433 ARCL X	447 20	461 "	475 ASTO IND 71
07 STOP	94 STO 12	178 RCL 09		345 1	434 "++++"	448 STO 71	462 ISG 71	476 RCL IND 71
08 FC? 23	95 RTN	179 STO 13	263*LBL 09	346 STO 10	435 RCL E	449 1	463 "	477 XEQ "LC"
09 GTO 01		180 RTN	264 RCL 11	347 RTN	436 E2	450 STO 72	464 GTO 00	478 X#0?
10 ASTO IND X	96*LBL 26		265 ST+ 10		437 *	451 FIX 0		479 GTO "PAY"
11 ISG X	97 RCL 12	181*LBL 62	266 RTN	348*LBL 25	438 30	452 CF 29	465*LBL "PAR"	480 ,
12 "	98 STO 09	182 RCL 13		349 RCL 11	439 -		466 DSE 72	481 STO IND 71
13 GTO 00	99 RTN	183 STO 09	267*LBL 10	350 RCL 10	440 10		467 "	482 GTO "PAY"
		184 RTN	268 RCL 09	351 X?Y?	441 X?Y?	453*LBL 00	468 DSE 71	483 .END.
14*LBL "RUH"	100*LBL 27		269 ST+ 10	352 SF 15		454 CLA		
15*LBL 01	101 RCL 12	185*LBL 63	270 RTN	353 0		455 ARCL 72	469 "	
16 AOFF	102 X? 09	186 RCL 09		354 FS?C 15				
17 20	103 STO 12	187 X? 13	271*LBL 11	355 1				
18 STO 19	104 RTN	188 STO 09	272 ISG 10	356 STO 10				
		189 RTN	273 "	357 RTN				
19*LBL 02	105*LBL 33		274 RTN					
20 CLA	106 RCL 09	190*LBL 14		358*LBL 26				
21 ARCL IND 19	107 LOG	191 RCL IND 73	275*LBL 12	359 RCL 10				
	108 INT	192 STO 09	276 RCL 11	360 X#0?				
22*LBL 04	109 STO 17	193 RTN	277 ST- 10	361 SF 15				
23 "H"	110 RTN		278 RTN	362 0				
24 RCL \		194*LBL "1"		363 FS?C 15				
25 X#0?	111*LBL 34	195*LBL "2"	279*LBL 13	364 1				
26 GTO 04	112 RCL 09	196 RCL 73	280 1	365 STO 18				
27 ,	113 LOG	197 XEQ "LC"	281 ST- 10	366 RTN				
28 X? \	114 FRC	198 STO 73	282 RTN					
29 "H"	115 RCL 17	199 RCL 72		367*LBL 27				
30 ,	116 +	200 XEQ "LC"	283*LBL 14	368 RCL 10				
31 X? \	117 101X	201 STO 72	284 RCL 09	369 RCL 09				
32 "H"	118 STO 09	202 8	285 ST- 10	370 X#Y?				
33 RCL \	119 RTN	203 X?Y?	286 RTN	371 SF 15				
34 RDH		204 SF 15		372 0				
35 CLA	120*LBL 35	205 0	287*LBL 15	373 FS?C 15				
36 STO [121 RCL 17	206 FS?C 15	288 RCL 09	374 1				
37 ASTO 72	122 RCL 09	207 1	289 ST+ 09	375 STO 18				
38 RDH	123 LOG	208 RCL 73	290 RTN					
39 STO [124 INT	209 2		376*LBL "3"				
40 ASTO 73	125 ST- L	210 *	291*LBL 16	377 RCL 73				
41 Rt	126 STO 17	211 +	292 RCL 10	378 XEQ "LC"				
42 Rt	127 X?Y	212 STO 73	293 ST+ 09	379 STO 73				
43 STO [128 ST+ L	213 8	294 RTN	380 RCL 72				
44 ASTO 71	129 LASTX	214 RCL 72		381 XEQ "LC"				
45 XEQ IND 71	130 101X	215 X?Y?	295*LBL 17	382 STO 72				
46 ISG 19	131 STO 09	216 -	296 ISG 09	383 2				
47 "	132 RTN	217 X?Y?	297 "	384 /				
48 RCL IND 19		218 -	298 RTN	385 INT				
49 SF 25	133*LBL 07	219 40		386 STO 71				
50 X#0?	134 RCL 73	220 +	299*LBL 18	387 RCL 73				
51 GTO 02	135 STO IND 16	221 STO 72	300 RCL 10	388 8				
52 "TERMINE"	136 RTN	222 XEQ IND 73	301 RCL 09	389 X?Y?				
53 PROMPT		223 RTN	302 -	390 SF 15				
			303 STO 09	391 X?Y?				
54*LBL "0"	137*LBL 08	224*LBL 00	304 RTN	392 -				
55 RCL 73	138 RCL 73	225 ,		393 FC? 15				
56 XEQ "LC"	139 40	226 STO 10	305*LBL 19	394 RDH				
57 STO 73	140 +	227 RTN	306 DSE 09	395 8				
58 RCL 72	141 XEQ IND X		307 "	396 *				
59 XEQ "LC"	142 RTN	228*LBL 01	308 RTN	397 ST+ 71				
60 XEQ IND X	143*LBL 42	229 ,		398 64				
61 RTN	144 14	230 STO 11	309*LBL 20	399 RCL 71				
	145 STO 16	231 RTN	310 -1	400 FS? 15				
62*LBL 00	146 RTN		311 ST+ 09	401 -				
63 RTN		232*LBL 02	312 RTN	402 STO 71				
		233 ,		403 RCL 72				
64*LBL 01	147*LBL 43	234 STO 09	313*LBL 21	404 2				
65 CF IND 73	148 15	235 RTN	314 RCL 09	405 MOD				
66 RTN	149 STO 16		315 CHS	406 GTO IND X				
	150 RTN	236*LBL 03	316 1					
67*LBL 02	151*LBL 44	237 RCL 11	317 -	407*LBL 00				
68 SF IND 73	152 RCL 14	238 X? 10	318 STO 09	408 RCL 18				
69 RTN	153 RCL 15	239 STO 11	319 RTN	409 X#0?				
	154 X=Y?	240 RTN		410 SF 16				
70*LBL 03	155 SF 15		320*LBL 22	411 ,				
71 0	156 0	241*LBL 04	321 RCL 11	412 FS?C 16				
72 FS? IND 73	157 FS?C 15	242 RCL 10	322 X#0?	413 RCL 71				
73 1	158 1	243 STO 11	323 SF 15	414 STO 71				
74 STO 10	159 STO 18	244 RTN	324 0	415 GTO 02				
75 RTN	160 RTN		325 FS?C 15					
		245*LBL 05	326 1	416*LBL 01				
76*LBL 05	161*LBL 46	246 RCL 09	327 STO 10	417 RCL 10				
77 RCL 73	162 ,	247 X? 10	328 RTN	418 X#0?				
78 RCL IND 16	163 STO 09	248 STO 09		419 SF 16				
79 X=Y?	164 STO 10	249 RTN	329*LBL 23	420 ,				
80 SF 15	165 STO 11		330 RCL 09	421 FS?C 16				
81 0	166 RTN	250*LBL 06	331 X#0?	422 RCL 71				
82 FS?C 15		251 RCL 11	332 SF 15	423 STO 71				
83 1	167*LBL 10	252 STO 06	333 0					
84 STO 10	168 RCL 09	253 RTN	334 FS?C 15	424*LBL 02				
85 RTN	169 STO IND 73		335 1	425 RCL 71				
	170 RTN	254*LBL 07	336 STO 10	426 FC? 15				
86*LBL 06		255 RCL 11	337 RTN	427 ST+ 19				
87 RCL 73	171*LBL 13	256 X? 09		428 +				

Marc Chiffolleau (9710-T189)
8 rue Auber
59046 Lille

22.03.83



Cher Jean-Daniel,

Je te fais parvenir une modification de ton programme assembleur paru dans "Au fond" page 97. Je l'ai modifié pour pouvoir entrer directement en ALPHA les numériques PPC: ce qui nécessite un "petit" fichier ASCII (110 REG). Pour la seconde partie, j'ai carrément pompé ton programme ! Ci-joint la liste du Programme et du fichier ASCII. Les lignes 206 et 207 sont en Hexa :
- FFOEOC08410102050A040903060D0B07
- F10F

En insérant une routine entre 247 et 248, on devrait pouvoir programmer un MLDL en utilisant directement les numériques ? A ceux qui en ont un d'en faire la preuve.

PPC Rom required

01*LBL "M"	47*LBL 71	91 X?Y?	136 16	184 XEQ 62
02 XEQ 99	48 64	92 GTO 19	137 STO 06	185 X#0?
03 CLA	49 -	93 RDH	138 XEQ "BD"	186 CLST
04 XEQ 60		94 128	139 256	187 X#0?
05 XEQ 62	50*LBL 72	95 X?Y?	140 XEQ "QR"	188 GTO 50
06 GETREC	51 FC? 07	96 FS?C 01	141 XEQ 18	189 "POST ?"
07 "-----"	52 128	97 -	142 1	190 PROMPT
08 RCL [53 FC? 07	98 32	143 +	191 16
09 CLA	54 +	99 XEQ "QR"	144 XEQ IND X	192 MOD
10 STO [55 CLA	100 XEQ IND Y	145 ASTO 01	193 X?Y
11 "F----	56 XEQ 61	101 2	146 CLA	194 4
12 RCL \	57 SF 07	102 XEQ "QR"	147 RDH	195 X=Y?
13 CLA	58 GTO 40	103 XEQ IND Y	148 RDH	196 GTO 50
14 STO [104 2	149 XEQ 18	197 RDH
15 GTO 40		105 *	150 RTN	198 10
	59*LBL "ASS"	106 1		199 X=Y?
	60 CF 08	107 X?Y?	151*LBL B	200 GTO 50
	61 "1ST ADR"	108 FS?C 00	152 XEQ 99	201 RDH
	62 XEQ 60	109 +	153 "PREF"	202 14
	63 XEQ 63	110 4	154 XEQ 60	203 X=Y?
	64 16	111 *	155 XEQ 62	204 GTO 50
	65 STO 06	112 3	156 16	205 X? 2
	66 XEQ "BD"	113 GTO 20	157 -	206 "VAR+X000+G+L+J"
	67 1		158 "POST"	207 "1+1"
	68 -	114*LBL F	159 XEQ 60	208 3
	69 STO 00	115 XEQ 17	160 XEQ 62	209 X=Y?
	70 12288	116 ,	161 48	210 CLA
	71 X?Y?	117 GTO 20	162 -	211 RDH
	72 GTO 19		163 CLA	212 AROT
	73 SF 08	118*LBL G	164 X?Y	213 ATOX
		119 XEQ 17	165 8	214 X? 1
		120 1	166 XEQ "QR"	215 RDH
		121 GTO 20	167 XEQ IND Y	
			168 X?Y	216*LBL 50
		122*LBL H	169 RDH	217 RDH
		123 XEQ 17	170 8	218 X?Y
		124 2	171 *	219 CLA
		125 GTO 20	172 +	220 XEQ 16
			173 4	221 ,
		126*LBL I	174 XEQ "QR"	
		127 XEQ 17	175 XEQ IND Y	222*LBL 20
		128 3	176 4	223 +
		129 GTO 20	177 *	224 XEQ IND X
			178 2	225 FC? 06
		130*LBL 17	179 GTO 20	226 GTO 40
		131 CF 07		227 ASTO X
		132 SF 06		228 X? 01
		133 "ADR"	180*LBL A	229 CLA
		134 XEQ 60	181 XEQ 99	
		135 XEQ 63	182 "PREF"	230 ARCL X
			183 XEQ 60	

231*LBL 40	266 RDN	297*LBL 07	327 STOP
232 19G 00	267 16	298 *F7*	328 ROFF
233 DEG	268 *	299 RTN	329 RTN
234 RCL 00	269 +		
235 4096	270 4		
236 XEQ *QR*	271 XEQ *QR*	300*LBL 08	330*LBL 61
237 XEQ IND Y	272 XEQ IND Y	301 *F8*	331 256
238 256	273 4	302 RTN	332 XEQ *QR*
239 XEQ *QR*	274 *		333 XEQ IND Y
240 XEQ IND Y	275 RTN	303*LBL 09	334 16
241 16		304 *F9*	335 XEQ *QR*
242 XEQ *QR*	276*LBL 00	305 RTH	336 XEQ IND Y
243 XEQ IND Y	277 *F0*		337 XEQ IND X
244 XEQ IND X	278 RTN	306*LBL 10	338 RTN
245 *F -		307 *F1*	
246 3	279*LBL 01	308 RTN	339*LBL 62
247 AROT	280 *F1*		340 POSFL
248 AVIEW	281 RTN	309*LBL 11	341 X*0?
249 FC? 00		310 *F2*	342 X*0?
250 GTO 65	282*LBL 02	311 RTN	343 RTN
251 FC?C 06	283 *F2*		344 *SYNTAX ERR*
252 STOP	284 RTN	312*LBL 12	345 PROMPT
253 GETKEY		313 *F3*	
254 CLA	285*LBL 03	314 RTN	346*LBL 63
255 ARCL 01	286 *F3*		347 ALENG
256 GTO 40	287 RTN	315*LBL 13	348 4
		316 *F4*	349 X=Y?
	288*LBL 04	317 RTN	350 RTN
257*LBL 18	289 *F4*		
258 16	290 RTN	318*LBL 14	351*LBL 19
259 XEQ *QR*		319 *F5*	352 *ADR ERR*
260 X<>Y	291*LBL 05	320 RTN	353 PROMPT
	292 *F5*		
261*LBL 16	293 RTN	321*LBL 15	354*LBL 65
262 4		322 *F6*	355 *LOAD ERR*
263 XEQ *QR*	294*LBL 06	323 RTN	356 PROMPT
264 XEQ IND Y	295 *F6*		357 .END.
265 X<>Y	296 RTN	324*LBL 60	
		325 *F ?*	
		326 AON	

HP-75 HP-75 HP-75 HP-75 HP-75 HP-75 HP-75 HP-75

Mon intention, à partir de ce numéro de PPCT, est de créer une rubrique aussi régulière que possible, consacrée au HP-75. Rassurons tout de suite les fanatiques de la HP-41, je n'ai pas l'intention de prendre un espace considérable dans le journal avec du BASIC ou autre (ce qui laissera toute la place voulue aux articles sur le BYTE GRABBER, CRIC et autre W^T, qui constituent bien entendu, la caractéristique fondamentale du HP-41 (?)).

Il me paraît toutefois intéressant de vous faire connaître ce qui existe dans le domaine logiciel et matériel sur cette machine, c'est ainsi que je vous présenterai les nouveaux modules ou logiciels (HP ou non) au fur et à mesure de leur parution et si, bien entendu, il m'est possible de les essayer car il est peu vraisemblable que je les achète tous (le 75 est déjà assez cher comme cela). De même, vous pouvez vous attendre à avoir des articles sur l'architecture de la machine.

Au menu de ce premier numéro, vous trouverez:

- FORTH sur le HP-75,
- un résumé des performances du 75 comparées avec d'autres machines BASIC,
- un petit programme-maison.

Comme Jean-Daniel a eu l'occasion de vous le faire connaître par une des dernières circulaires, des Américains ont mis au point un interpréteur FORTH pour le 75. J'ai commandé aux USA, et j'ai beaucoup de mérite car une telle opération est vraiment devenue difficile pour les particuliers, la version "commercialisée" du produit (V1.00). Cette version, a priori, ne devrait pas avoir de bugs majeurs, et est accompagnée d'une documentation. Il existe toutefois dans la nature une version précédente (V0.02) et j'ai demandé aux auteurs l'autorisation de la donner à ceux qui m'en feraient la demande*. Cette version de développement n'est ni complète ni juste et vient sans documentation, elle peut toutefois constituer pour les personnes intéressées une bonne introduction. Si vous la voulez, faites moi parvenir une cassette ou 11 cartes et je vous l'envierai dès que possible. Le logiciel complet coûte \$165 et peut être commandé à l'adresse suivante:

FORTH Age Products
Suite 213
23 Altarinda Road
ORINDA, California

* REFUSEE

La revue DECISION INFORMATIQUE du 10 Janvier 1983 a publié un banc d'essai du HP-75 très intéressant, dans lequel un certain nombre de programmes test (benchmark simplifié) ont été essayés sur diverses machines; ceci complète l'étude similaire parue dans le Calculator Journal de PPC en Août dernier:

	Apple II	IBM PC	HP-75
10 FOR I=1 TO 5000	6.7 sec	6.4 sec	12.5 sec
20 NEXT I			
10 A=2.71828	29 sec	23.8 sec	55 sec
20 B=3.14159			
30 FOR I=1 TO 5000			
40 C=B/A			
50 NEXT I			
10 A=2.71828	13.9 sec	12.4 sec	28 sec
20 B=3.14159			
30 FOR I=1 TO 5000			
40 GOSUB 1000			
50 NEXT I			
10 A\$ =	32.3 sec	23 sec	47 sec
"abcdefghijklm"			
20 FOR I=1 TO 5000			
30 B\$=MID\$(A\$,6,6) (1)			
40 NEXT I			

(1) remplacé sur le 75 par B\$=A\$(6,12).

LISTE DU FICHIER ASCII POUR "ASS"

222 (9710)

"NOP"	← Pour des commodités d'affichage, certaines syntaxes ont été modifiées:
"CLRF"	- a remplace @ .
"SETF"	- + remplace & .
"FSET?"	
"LDAR--"	
"?R="	
"R="	← ATTENTION: Ne pas oublier l'espace!
"Z"	
"SELP"	"ST=03C4"
"WRIT"	"CLRKEY3C8"
"?FI"	"?KEY3CC"
"Z"	"R=R-13D4"
"Z"	"R=R+13DC"
"READ"	"G=C0958"
"RCR"	"C=G098"
"A=0"	"C<>G0D8"
"B=0"	"M=C158"
"C=0"	"C=ML98"
"A<>B"	"C<>M1D8"
"B=A"	"T=ST258"
"A<>C"	"ST=T298"
"C=B"	"ST<>T2D8"
"C<>B"	"ST=C358"
"A=C"	"C=ST398"
"A=A+B"	"C<>ST3D8"
"A=A+C"	"XQ-G0020"
"A=A+1"	"POW0FF060"
"A=A-B"	"SLCTP0A0"
"A=A-1"	"SLCTQ0E0"
"A=A-C"	"?P=Q120"
"C=C:C"	"?LWDA T260"
"C=C+A"	"A=B=C=01A0"
"C=C+1"	"GOTOADR1E0"
"C=A-C"	"C=KEY220"
"C=C-1"	"SETHEX260"
"C=0-C"	"SETDEC2A0"
"C=-C-1"	"DSP0FF2E0"
"?B#0"	"DSPT0G320"
"?C#0"	"RTN3E0"
"?A<C"	"?CRTN360"
"?A<B"	"?NCRN3A0"
"?A#0"	"N=C070"
"?A#C"	"C=N0B0"
"RSHFA"	"C<>N0F0"
"RSHFB"	"LDI130"
"RSHFC"	"PUSH170"
"LSHFA"	"POPLB0"
"aR"	"GOTOKEY230"
"S+X"	"RAMSLCT?70"
"R<"	"WRITED2F0"
"ALL"	"FETCH330"
"P-Q"	"C=CORA370"
"XS"	"C=CANDA3B0"
"M"	"PRPHSLCT3E0"
"MS"	

Résultats vraiment excellents puisque le rapport de vitesses 75 / IBM PC n'est que de 2.17. Des tests similaires étaient conduits avec des micros portables, les résultats cumulés donnaient:

Sharp 1211: 22 mn 27 sec. (21.05)
 Sharp 1500: 4 mn 54 sec. (4.59)
 Casio FX702P: 7 mn 43 sec. (7.23)
 HP-75C: 1 mn 04 sec. (1.00)

Reste, bien entendu, à relativiser ces tests par des critères de coût, de précision de calculs etc... Par ailleurs il est vraisemblable que l'Apple II et l'IBM PC utilisaient un interpréteur et non un compilateur.

* * *

Pour conclure, le petit programme annoncé. Il s'agit d'une reprise du programme IG du PPC ROM, (Voir manuel P. 220 et suivantes) de calculs d'intégrales et permet d'illustrer très bien les avantages et inconvénients du BASIC et de la notation polonaise. Le BASIC est sans commune mesure plus facile à écrire, à mettre au point, et à maintenir: ce programme a été écrit en un peu moins d'une heure à un moment où le BASIC était pour moi une nouveauté, et il est d'une grande lisibilité puisque les variables ont le même nom que dans l'algorithme rapellé plus loin; par ailleurs la fonction à intégrer est donnée "en clair" à la ligne 230. En revanche il est très gourmand en mémoire: avec la fonction $SRQ(X)/(X-1) - 1/LN(X)$, il faut très exactement 2381 octets (programme + variables). Le programme est relativement rapide (c'est à dire beaucoup plus qu'une 41 et beaucoup moins qu'un IBM 370/168), moins toutefois que la fonction similaire, en assembleur, prévue dans le module Mathématiques (à sortir à l'automne si l'on en croit HP).

$I = \lim_{k \rightarrow \infty} M(k,k) \quad k=1,2,3,..$ avec:

$$u_0 = -1 + 2^{-k}$$

$$u_i = u_{i-1} + 2^{1-k}$$

$$x_i = ((b-a)/4u_i(3 - u_i^2) + (b + a)/2)$$

$$S_0 = f((a + b)/2)$$

$$S_k = \sum_i f(x_i)(1-u_i^2) + S_{k-1}$$

$$M(k,0) = (3(b-a)/4)*2^{-k}*S_k$$

$$M(k,j) = M(k,j-1) + (M(k,j-1) - M(k-1,j-1))/(4j-1)$$

Ceci termine ce premier article; encore une fois, j'invite les possesseurs d'un 75 à me contacter (ceci permettra, entre autre, de connaître la taille du "marché" du ROM danois, si celui-ci voit le jour) et/ou à me transmettre leurs articles et autres programmes, ceci évitera également de faire de cette colonne un monologue!

J. TAILLANDIER T8/7451
 9 Rue DELEZENNE
 59000 LILLE
 (20) 93-78-60

INTECP B 734 22:57
 03:05:03

```
10 DIM M(12:12),S(12)
20 FOR I=0 TO 12 @ FOR J
  =0 TO 12 @ M(I,J)=0 @ NE
  XT J @ NEXT I
30 INPUT 'Limites A,B?'
  :A,B
40 INPUT 'Precision?'
  :L @ DISP ..
60 S(0)=FNY((A+B)/2)
70 M(0,0)=3*(B-A)*S(0)/4
80 FOR K=1 TO 12 @ DISP
  K:M(K-1,K-1)
100 U=-1+2^(-K)
110 S(K)=S(K-1)
120 FOR I=1 TO 2^K
```

```
130 S(K)=S(K)+(1-U*U)*FN
  Y((B-A)*U*(3-U*U)/4+(B+A
  )/2)
140 V=U+2^(1-K) @ U=V
150 NEXT I
160 M(K,0)=3*(B-A)*S(K)*
  2^(1-K)/4
170 FOR J=1 TO 12
180 M(K,J)=M(K,J-1)+M(K
  ,J-1)-M(K-1,J-1)/(4*J-1
  )
190 NEXT J
200 IF ABS(M(K,K)-M(K-1,
  K-1))<L THEN DISP M(K,K)
  @ BEEP @ END
210 NEXT K
230 DEF FNY(X) = SQR(X)/
  (X-1)-1/LOG(X)
```

* *

Cher Jean-Daniel,

Je t'écris pour ouvrir, en quelque sorte,

la rubrique HP 75 .

PPC-T No6
 JUILLET AOUT
 1983 P29

Et oui, je viens de faire cette folie ! (9 400 F). Je me suis plongé aussitôt dans le confortable manuel (354 pages, index compris...). Je ne vais pas me lancer dans la description du 75 qui a été très bien faite dans l'OI et l'OP, mais émettre quelques critiques qui s'avèreront peut-être constructives :

- Absence d'accès en LM directement pour l'instant, mais je compte sur tous les possesseurs de 75 qui ont envie de démonter le bloc d'accus pour volatiliser partiellement la MEV .

- Peu d'accès en programme dans le calendrier étendu . On aurait pu l'espérer, en regard du module time du 41 . En effet, hormis TIME / TIME \$ / DATE / DATE \$ qui sont les 4 fonctions donnant heure et date du jour sous 2 formes différentes il n'y a pas possibilité de calcul du nombre de jours entre 2 dates, ni de chrono à lecture directe.

- Enfin quelques fonctions du basic de la série 80 ne sont pas adressables indirectement : c'est-à-dire que l'on est obligé de spécifier la valeur numérique comme argument . Exemple :

DIM V (N) est refusé même si l'on a défini au préalable la variable N, "étonnant non" ?

Dans un premier temps, j'ai imaginé créer la ligne de programme en stockant sur un fichier TEXT. une variable \$ contenant le libellé de la fonction, puis, après l'avoir transformée en Basic, le "MERGE" au programme principal :

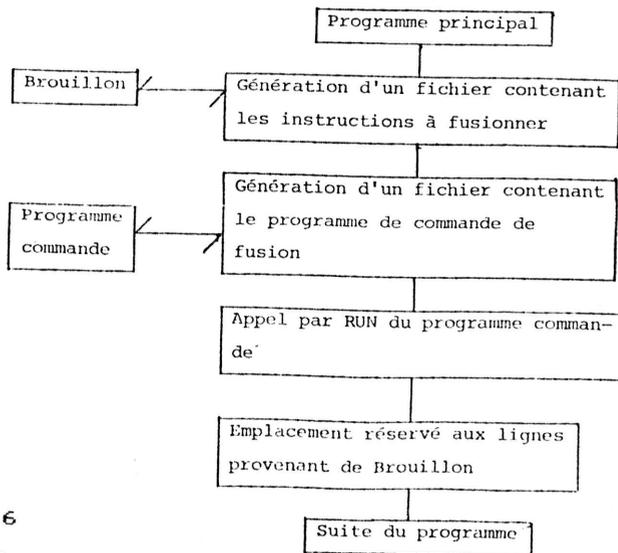
```
100 ASSIGN # 1 TO "Brouillon", TEXT
110 INPUT N
120 A $= "DIM V (" & STR $(N) & ")"
130 PRINT # 1, 160; A$
140 TRANSFORM "Brouillon" INTO BASIC
145 EDIT "Brouillon"
150 MERGE "BROUILLON"
160 ! Place réservée à la ligne d'instruction
170 ! Suite du programme
```

Trop beau pour être vrai ?

Vous avez raison, le manuel (p. 172) me l'avait bien dit: "La fusion d'un fichier d'un programme alloué désalloue ce dernier" !!

Que faire ?

Et bien, sur le même principe, on crée un 2° fichier qui commande les modifications du 1° vers le programme principal. De plus, pour une raison de commodité, le programme principal doit contenir tous les futurs fichiers temporaires pour économiser les cartes magnétiques .



Enfin, si ce n'est toujours pas réglé, voici un listing exemple; quelques dernières précisions : l'instruction CALL (appel du programme) s'arrête sur une instruction MERGE; j'ai donc dû faire appel à RUN, avec préservation sur un 3° fichier temporaire des variables que j'avais préalablement entrées, comme N, pour V (N).

"EXEMPLE"

```
10 ASSIGN # 1 TO "DATA"
20 ASSIGN # 2 TO "BROUIL", TEXT
30 ASSIGN # 3 TO "COMMAND",TEXT
40 INPUT N
50 PRINT # 1,1;N
60 A$ = "DIM V ("% STR $(N) &")"
70 PRINT # 2, 130; A$
80 A$ = 'EDIT "EXEMPLE" @ MERGE "BROUIL" @ RUN "EXEM-
  PLE", 130'
90 PRINT # 3; A$
100 TRANSFORM "BROUIL" INTO BASIC
110 TRANSFORM "COMMAND" INTO BASIC
120 RUN "COMMAND"
130 ! .....
140 READ # 1, 1; N
150 ! Suite du programme
```

Voilà !

Je réclame votre indulgence, je n'ai la machine que depuis

8 jours. J'espère avoir été assez clair.

Merci d'avoir tenu la distance.

Bravo pour le journal
Synthétiquement vôtre,

Eric ANGELINI T 210

14, Avenue Marianna

06000 NICE



T 210

P.S. Je vends un VIDEO GENIE SYSTEM (achat : Août 1981), avec sortie son du lecteur cassette intégré, avec quelques programmes commerciaux de jeux, avec manuels et V1 pratique du TRS 80. Prix = 3 000 F avec Monitor (TV NB modifiée).



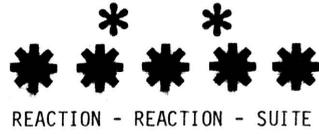
Janick Taillandier m'a envoyé un petit programme permettant de faire du langage machine sur HP 75, et cela en transformant un fichier "Texte". Je n'ai pas de 75, donc mes explications sont peut-être sommaires, mais je pense qu'il ne faut pas laisser ce genre d'information ignorée, même si tout n'est pas au point.

Janick utilise un LEX FILE du commerce. Il charge en fichier texte un programme assembleur (pour l'instant trouvé dans PPCJ-computer), range ce programme sur cassette puis lance le programme ci-joint qui transforme le programme texte en programme assembleur. Le ruban d'imprimante est en bleu, j'espère qu'il sera quand même lisible, c'est beaucoup trop compliqué pour être retapé à la machine!

Je voudrais ajouter qu'il doit être possible de ressortir la vieille méthode de la "black box" (boite noire) utilisée par PPC pour truander les machines précédentes la HP 41: il s'agit d'un simple potentiomètre bobiné de quelques dizaines d'ohms intercallé à l'aide d'isolant papier et de conducteurs en feuilles d'aluminium ménager entre un pôle de l'alimentation et le pôle correspondant de la machine. Une rotation du potentiomètre fait varier la tension d'alimentation de la machine et permet d'arriver à la tension de décrochage du microprocesseur avec des résultats inattendus....

```
40 SENDIO A$;LAD#;DOUT
  CHR$(0)
50 O$(1,2)=CHR$(0)/CHR$(
  2)
60 SENDIO A$;LAD#;IDU4
  ,0$
70 B$=ENTIO$(A$;TAD#;P
  DTZ;SDA#)
80 B$(11,12)=CHR$(224)
  CHR$(127)
90 SENDIO A$;LAD#;DILE
  ,CHR$(0)
100 SENDIO A$;LAD#;DOL4
  ,0$
110 SENDIO A$;LAD#;DOL2
  ,0$
120 SENDIO A$;LAD#;DOL4
  ,0$
200 ON ERROR GOTO 340
220 ASSIGN # 1 TO "INPUT"
```

```
400 SENDIO A$;LAD#;DOL4
  ,0$
410 SENDIO A$;LAD#;DOL4
  ,0$
420 SENDIO A$;LAD#;DOL4
  ,0$
430 SENDIO A$;LAD#;DOL4
  ,0$
440 SENDIO A$;LAD#;DOL4
  ,0$
450 SENDIO A$;LAD#;DOL4
  ,0$
460 SENDIO A$;LAD#;DOL4
  ,0$
470 SENDIO A$;LAD#;DOL4
  ,0$
480 SENDIO A$;LAD#;DOL4
  ,0$
490 SENDIO A$;LAD#;DOL4
  ,0$
500 SENDIO A$;LAD#;DOL4
  ,0$
```



total). Donc : vive Duriez !
Mon revendeur "local" m'annonce une baisse de 60 % du système HP-IL (l'HP-IL, le lecteur de K7, l'interface vidéo, la nouvelle imprimante etc...) pour le début des vacances. Voilà qui met en particulier l'interface vidéo à portée de pas mal de monde.. Je me suis mis à la "Users' Library" car étant étudiant en électronique, certains programmes sont très intéressants. Les derniers trucs que j'ai sont :

Programmer's Reference Guide
Catalog of Contributed Programs (contient le dernier key note, le n° 7). Si par le plus grands des hasards tu ne les as pas, je peux te les prêter : je te les envoie, tu me les fais parvenir quand tu peux. D'autre part, j'ai cru comprendre que tu avais de la famille ici à Montpellier : si à l'occasion tu passes et que tu as du temps de libre, contacte-moi j'aimerais discuter et voir "en chair et en os" "mon président". Demande Jean-Philippe au téléphone.

Je reviens au programme KA : la différence de réaction face à une erreur de programmation doit venir du fait que dans le nouveau programme, la pile contient déjà un nombre important avant l'introduction des données et il faut que ce nombre se retrouve en T après l'introduction des données sinon ça foire. Dans l'ancien programme, c'est le registre L qui contient un nombre important donc même si on ne rentre que 1 ou 2 données, ce "nombre important" n'est pas perdu. Malheureusement, je suis incapable d'effectuer une synthèse des deux programmes.

Pour finir, je te commande :
2xCl 2 cartes des codes HP 41 C en couleur 25F X 2.
1Xc5 Autocollants toutes fonctions HP 41 et periph.

Fais donc parvenir au trésorier le chèque bancaire de 55F ci-joint.

Dernier mot : comment avoir d'anciens numéros de PPC (USA), comment s'y inscrire ?

En espérant que je ne t'ai pas trop saoulé de questions et de bla-bla...

T - 284 te salut bien.
P.S. "n'est pas président qui veut"

1 X C5 + 2 X C1 à :

J. PH. IMBACH
1108, rue de Las Sorbes
34000 Montpellier
Tél : (16 67) 63 00 43

Cette lettre m'a paru valloir la publication.

Quand à la mise en page du journal, les réflexions que tu fais sont intéressantes. En fait je pense que tu emploie la bonne méthode: souligner en couleur les rubriques selon le classement qui te plaît.

Pour le club j'ai suis amené à m'occuper de plusieurs choses, comme tu peux le voir en dernière page. En fait "l'épicerie" me prend en moyenne une heure par jour. C'est à la fois peu et beaucoup, car c'est autant de temps que je ne peux consacrer à la mise en page du journal.

Cependant, je ne tiens pas pour l'instant à être déchargé de cet aspect du travail car, dans la structure actuelle du club, c'est le seul moyen pour moi de rester en contact avec les adhérents.

Une entreprise habituelle fonctionne de façon centralisée, tout le courrier arrive dans le même local où les administrateurs peuvent le consulter avant d'y répondre. Rien de tel pour nous. Nous n'avons pas de local et je ne souhaite pas que nous en ayons! Pour des raisons familiales, je tiens essentiellement à avoir des activités qui me permette de rester chez moi. De toute façon, un local est absolument au dessus de nos moyens.

Du coup l'administration est dispersée. Ce n'est pas grave pour le débat ou la prise de décisions, car le téléphone fonctionne bien! Mias je n'ai à peu près aucun écho de ce que vous écrivez au trésorier, au responsable polycop, etc...

C'est normal. Mais les réflexions dont vous émaillez vos lettres sont pour moi le meilleur des sondages sur la vivacité du club. Si j'abandonnais toute activité liée au courrier, je perdrais donc une source précieuse de renseignements. De plus, meilleure mise en page suppose parcellisation

du journal en rubriques de quelques lignes. Comme tout cela est alimenté par vos lettres, cela supposerait que vous fassiez autant de feuilles que de réflexions, et manifestement ce n'est pas dans vos habitudes, ou que je retape ces réflexions après ventilation, autant tout réécrire.

Certains regrettent les mentions sur les commandes dans les lettres publiées. Elles ne sont sans doute pas indispensables mais généralement soigneusement mêlées au reste, et d'ailleurs cela donne une idée de la vie du club, non?

La "budine" est un anneau prévu pour recevoir un cadenas pour éviter la fauche dans les bureaux, efficace?

J'espère que vous avez raison quand à la baisse dans les magasins, mais 60% me paraît bien optimiste!

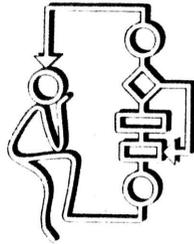
Remarque: ta lettre est un exemple des commandes "imbriquées dans le texte!!!

amicalement

J-D Dodin (T1)



PPC L



Lausanne, le 7 mai 1983

Philippe Komascano (T69)

Grand-Vennes 39 IOIO Lausanne
CH

Cher Jean-Daniel,

Voici quelques nouvelles du Club

P.P.C. en Suisse. Après cinq mois de recherches, nous avons enfin trouvé un local de réunions à Lausanne. Si l'adresse vous intéresse, la voici : " Le Cazard ", 15 rue Pré-du-Marché. Nous nous y réunissons en général les premiers lundis de chaque mois. Nous essayons de présenter lors de ces rencontres le plus d'informations et de matériels. Quelques membres ayant un 75C, nous pensons bientôt l'inclure dans les articles de notre journal.

Hewlett-Packard Genève m'a aimablement prêté un plotter 7470A pour deux semaines. Je pense que c'est un appareil fantastique pour l'édition de codes-barres, mais il dépasse les possibilités de la 41 pour le tracage de courbes. Lorsque l'on pense avec quelle facilité le Basic vous permet d'y parvenir, je ne peux que dire que la 41 n'est pas à la hauteur de ce plotter ; heureusement qu'il est disponible en HP-IB...

Dans le dernier numéro de votre journal, un de vos membres, Monsieur Pelanne de Paris (T191) signale les difficultés qu'il y a en France pour la commande des PPC ROM aux U.S.A. Pour ce qui est de la Suisse, je prends moi-même les commandes pour le Club; j'obtiens les ROM de Richard Nelson au prix de environ 135 francs suisses (= 400FF ?). Comme il n'y a aucune restriction de change en Suisse, voici ce que je vous propose : je pourrai me charger de prendre les commandes de vos adhérents au prix ci-dessus et de leur faire parvenir une ROM; je précise que notre Club n'a aucun but lucratif et que nous ne touchons rien sur le prix des ROM; Je pense que de cette manière tous les adhérents de P.P.C. de Suisse ou de France pourront disposer de ce merveilleux module qu'est le PPC ROM.

Côté pratique, voici ce que je vous suggère : **PPC-T No6**
JUILLET AOUT
1983 P31

publiez cette lettre, soit dans votre prochain journal, soit, pourquoi pas, sous la forme d'une de vos circulaires mensuelles.

Je demande à toutes les personnes intéressées de me faire parvenir la somme de 140 francs suisses, qui incluerait les frais de ports pour la France. Je crois que les Eurochèques sont le meilleur moyen pour cela. Voici à nouveau mon adresse :

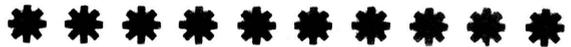
Philippe Komascano Grand-Vennes 39

IOIO Lausanne Suisse. Joignez à votre commande une enveloppe portant vos nom et adresse, ainsi que l'équivalent de 2irs 40 FF, que je puisse vous envoyer un reçu, comme garantie de réception. Lors de la dernière commande du Club, pour 15 ROM, tout a été réglé en 2,5 mois, les ROM étant arrivées en Suisse 3 semaines après commande, et les manuels 1 mois après, par bateau. Je ne peux rien garantir comme délai de livraison, car les postes américaines sont trop souvent atteintes de paresse... Afin d'éviter une commande trop importante si beaucoup de monde est intéressé, nous pourrions envisager de faire les commandes par groupes de 20 ROM, au maximum une fois par mois. De cette manière, je ne risque pas de me retrouver avec 50 demandes à la fois !

Voilà, j'espère, Jean-Daniel, que vous n'accueillerez pas mal ma proposition, car encore une fois il ne s'agit pas de faire du fric, mais de donner accès au plus de monde possible à cette ROM.

Je pense que pour une fois je vous ai envoyé une lettre au format correct pour être publiée. Je vous envoie mes meilleures salutations et amitiés.

Philippe Komascano



REMARQUE

Vu le volume de la matière dans ce journal, un bon nombre de textes intéressants sont "restés sur le carreau", y compris des compte-rendus de la réunion de Bretagne, des lettres, la traduction de l'excellent article de Richard Nelson sur l'électricité statique et plein de programmes. Ce sera pour le prochain numéro. Celui-ci vous parviendra début Septembre, je ne veux pas faire d'envoi pendant le mois d'Aout, il y a trop d'absents à l'adresse indiquée.

NOTE SUR LE FORMAT

Pour être publiables, vos textes doivent être tapés à la machine (à la rigueur calligraphiés sur quadrillage 5x5) en 13,5 cm de large maxi, simple interligne, recto seul, et bien noirs. Je ne suis pas sûr de ce que vont donner dans ce numéro des textes tapés en ocre ou en gris (j'allais dire en blanc!)

Tant que la liste des adhérents n'est pas publiée, il est souhaitable que vous indiquiez vos nom et adresse en début ou en fin de texte.

Envoyez obligatoirement les programmes sur carte magnétique ou cassette (retournées à la demande, toujours pour les cassettes), avec une liste du programme au cas où le support magnétique serait défectueux.

Ne faites pas de mise en page, vous ne pouvez pas savoir comment est ma page.

Merci



Cette page sera publiée dans chaque numéro de PPC-T. Seuls les produits répertoriés sont disponibles. PPC-T étant une association sans but lucratif elle ne fait pas de bénéfice et ses produits et services sont réservés aux adhérents.

Association PPC-T

Association à but non lucratif (loi de 1901), PPC-T accepte des adhérents dans toute la France et à l'étranger. La cotisation, abonnement au journal inclu, est de: France et Europe 100 F (français) DOM-TOM et reste du monde par avion 150 F Modes de règlement: 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 du bulletin d'adhésion (1er Janvier, 1er Mars....).

Responsables de PPC-T

Président: Jean-Daniel Dodin 77 rue du Cagire 31100 Toulouse (nouvelle adhésions, coopérative, eproms, numéros du journal de l'année en cours).
Trésorier: Jean-François Sibille Résidence du Pays d'OC rue des genêts 31500 Toulouse (librairie, renouvellement d'adhésions, fichier, problèmes financiers).
Secrétaire: Jean-Luc Basso 4 rue Jacques Darré 31300 Toulouse (coordination des chapitres, organisation des réunions).
Programmiste: Damien Debril 38 rue du 8 Mai 1945 59190 Hazebrouk.
Photocopies, album 82 du journal: Gilles Barret 92 av de Muret 31300 Toulouse

Coordinateurs régionaux:

Bretagne: Frank Lebastard 10 rue de la Jalousie 35600 Redon

Des coordinateurs sont recherchés pour toutes les régions (démonstrations, collections de journaux, réunions à organiser...)

Disponible à Toulouse

Chez le trésorier uniquement:

Librairie:

L1: William Kolb, Curve Fitting 2^e édition 120 F
L2: John Dearing, Calculator tips and routines 130 F
L4: J-D Dodin, Au fond de la HP 41C 80 F
L5: Keith Jarett, Synthetic programming made easy 130 F

Chez le président uniquement:

Coopérative:

C1: Carte des codes HP-41 couleur 25 F
C3: Overlay perforés en bristol, les 10 30 F
C5: autocollants toutes fonctions 41C et périphériques 5 F
C7: cartes magnétiques HP avec classeur, les 80 250 F
C8: classeur de cartes magnétiques de poche 20 F

Anciens journaux:

N°3, 4 ou 5, chacun 15 F

N°6 20 F

Eproms:

Il s'agit d'un jeu d'eproms (2716+2732) prêt à fonctionner sur n'importe quel lecteur d'eproms comme le BE-01-4K, ou le MLDL. Le prix unique de 200 F comprend la fourniture des eproms, le port en recommandé et les mises à jour éventuelles (par retour des eproms à Toulouse) pendant un an. Pour ceux d'entre vous qui ont un lecteur acceptant 8K, des combinaisons peuvent être réalisées (ex: TR1D + monitor). Le jeu de trois eproms (2x2732 + 1 2716) est alors facturé 300 F. Si vous fournissez vous même les eproms, la copie est facturée 70 F. Règlements à l'ordre de J-D Dodin.

Sont disponibles:

HP service module 1C (non compatible avec d'autres eproms, fonctionnement décrit dans P11)

Copie du module X-fonctions

PPC ROM 2C (microcode) dans sa forme originale (fonctions incluses dans Toulouse)

HP-IL Monitor: version précoce du HP-IL Dev. Module)

Toulrom, dernière version courante.

Possibilité de copier n'importe quelle eprom fournie par vous.

Cassettes:

Pour ceux qui le souhaiteraient, je peux vous copier n'importe quel jeu d'eproms sur cassette. Comptez 20 F + 20 F par jeu

PPC-T N°6
JUILLET AOUT
1983 P32

d'eprom. références ci-dessus uniquement. Pour usage avec le MLDL.

Fournissez une cassette déjà initialisée (NEWM 70 par exemple).

Uniquement chez Gilles Barret:

Photocopies:

P1: Data base, extraits de la notice	25 F
P2: PPC East cost conference (avril 82)	50 F
P7: Microcode PPC:tous les ROMS HP jusqu'en 82	120 F
P5: Microcode 3, restent 2 exemplaires (chez Dodin)	40 F
P8: Microcode HP	220 F
P9: documentation sur HP-IL Monitor	10 F
P10: trésors cachés de la HP-41C (1'01)	15 F
P11: manuel technique HP-41C	60 F
P12: manuel du convertisseur HP-IL	25 F
P13: PPC Southwest conference (janvier 83)	35 F
P14: HP Journal janv 83 en anglais (HP-IL)	20 F
P15: dossier Hard: tout ce qui est paru sur le hard jusqu'en 82	45 F
P16: Album PPC-T 1982 (1ère année du journal)	50 F

REMARQUES IMPORTANTES

Passer commande à un autre que le responsable nous fait perdre un temps précieux et retarde votre livraison de une semaine à 15 jours.

Tous les prix sont franco de port.

Les anciens numéros de PPCJ (USA) sont maintenant disponibles, même pour les non adhérents au club américain (mais il faut les commander là-bas). Liste avec contenu et prix sur demande. Nous pourrions sans doute agir de la façon suivante: faire en France les photocopies, payer au club américain un droit de reproduction et diffuser nous même ces journaux. ils n'en seraient pas moins en anglais, mais plus faciles à obtenir. Mais compte tenu du (très gros) travail que cela représente, nous ne pourrions faire la vente que par années complètes et pour un montant entre 200 et 250 F chaque année (de 150 à 300 pages au format PPC-T). Ecrivez-moi vite pour me donner votre avis (Il y a au total 10 ans de PPC, 4ans pour la 41C).

05.05.1983 21:23:25

PROGRAMMES DISPONIBLES

CHEZ T33

DAMIEN DEBRIL

38 RUE DU 8 MAI 1945

59190 HAZEBROUCK

FRANCE

POUR LES CARTES PREVOIR

ENVELOPPE RETOUR AFFR.

A 1.60F OU 2.00F.

POUR LES CASSETTES

METTRE 2.30F OU 3.30F

PREVOIR UN EMBALLAGE

SOLIDE CAR IL Y A BEJA

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.

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 10 -2 C-INTER

V1N4 P 11 -2 C-TAB

V1N4 P 11 -2 C-651

V1N4 P 9 -1 C-T

NO 1 P 16 -1 C-QUOT

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 -4 C-CFA

NO 1 P 13 -1 C-LFA

NO 1 P 13 -1 C-BH

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 0 -1 C-ASS

NO 4 P 0 -8 C-RC

NO 4 P 0 -1 C-RCKEYS

NO 4 P 0 -1 C-RCDATA

NO 4 P 0 -8 C-R

NO 4 P 0 -1 C-D-H

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

Journal PPC-T

Titre: PPC-T; adresse 77 rue du CAGIRE 31100 Toulouse

N°ISSN 0754-832X (1982)

Directeur de publication-éditeur Jean-Daniel Dodin