

JPC

1983

MAI

1983

VOLUME-1
NUMERO-2

APPLICATIONS

ROBERT SCHWARTZ	1	DEUX NOUVELLES DECOUVERTES
ROBERT SCHWARTZ	2	LBL "DELOCALISES" ET BG
ROBERT SCHWARTZ	4	-E ET LBL DELOCALISES
ROBERT SCHWARTZ	6	W ET CHAINE "STOFLAG"
ROBERT SCHWARTZ	8	COUP D'OEIL SUR LE REG c
ROBERT SCHWARTZ	9	REGISTRE ALPHA
ROBERT SCHWARTZ	11	STOCKAGE DANS LES REGISTRES "NORMAUX"
PHILIPPE GUEZ	13	REPONSE A ROBERT
YANN DOLHEN	19	DEVANT LA DIFFICULTE NE BAISSONS PAS LES BRAS
YANN DOLHEN	20	TABLEAU DES DRAPEAUX DE LA HP 41
ROBERT SCHWARTZ	22	DEUXIEME PAS
ROBERT SCHWARTZ	24	FONCTION "W"
PHILIPPE GUEZ	26	LES PREMIERS PAS DE BEBE (2°EPISODE)
PHILIPPE GUEZ	26	TEXT ENABLER
PHILIPPE GUEZ	27	Q LOADER
PHILIPPE GUEZ	27	LE ENHANCED BYTE JUMPER
PHILIPPE GUEZ	29	LE PETIT THEATRE DES MICROCODES
DIDIER JEHL	34	MACHINE LANGAGE DEVELOPPEMENT LAPORATORY

PROGRAMMES

ROBERT SCHWARTZ	16	CDXR
YANN DOLHEN	21	PRF
YANN DOLHEN	21	FL
YANN DOLHEN	21	DISPLAY

EDITORIAL

RICHARD NELSON	38	
ROBERT SCHWARTZ	25	REUNION A LA CHASSERIE

DUR ET MOU

BIEN CHER PHILIPPE	14	<u>BRAVO PEPE</u>	8
<u>RECTIFICATION</u>	17	<u>BIEN CHER PHILIPPE</u>	16
<u>CHOC EN RETOUR</u>	39	<u>NOTES</u>	33

Deux nouvelles découvertes

Page 139 du dernier OI (N°46), Stéphan Harlé propose un PRGM qui contient de la PS . Il semble utiliser le CRIC mais ce qui a principalement retenu mon attention, ce sont les 2 fonctions qui permettent d'obtenir ce qu'il appelle des LBL "délocalisés". Les codes indiqués pour obtenir les ASN de XROM 52,00 et de @ AHHH sont bien utilisables avec "KA" et le mode d'emploi indiqué permet bien la génération des LBL & GTO "délocalisés".

Cette nouveauté (pour moi) me semble intéressante particulièrement pour accélérer de façon appréciable, l'utilisation des 2 premières rangées de touches pour l'ASN à des LBL.

BG+LBL/GTO ?

Je n'avais pas eu connaissance de cette possibilité et j'avais même remarqué que l'on faisait le mutisme total sur les fonctions correspondant aux codes DEC 29 à 31 .

USER KEYS:

- 11 PACK
- 12 XROM 52,00
- 12 @+AHHH
- 14 DEL
- 15 XROM 28,63

J'ai donc complété ma carte d'ETAT comme indiqué ci-contre mais pour être complète, cette nouvelle possibilité devrait être complétée par la fonction XEQ "ld" qui devrait pouvoir être générée. (*quel français!*)

Je lance donc un "SOS" afin d'être éclairé à ce sujet. Il y aura bien un adhérent dévoué qui prendra sa machine à écrire pour nous initier à ces LBL délocalisés (?).

Je ne dis pas merci d'avance car je n'apprécie pas dutout cette expression.

R.S.

P.S.: Les codes DEC de - E (-1) sont 28 27 (c.q.f.dire); qui me dira comment l'obtenir avec le BG ?

-25 ENG

XEQ "ld" s'obtient avec l'ASN codes DEC 1 30(@NQ)

- USER KEYS: *Codes DEC*
 11 X *1, 10 à 1, 31*
 -11 #ae2
 12 HLLD
 13 Aa
 14 #ae2
 15 OD
 21 P
 -21 -
 22 a#pdc 1)52-#T+NvIAa
 -22 Hb
 23 P
 -23 #ae2
 24 M
 -24 @+AHHH
 25 DC#B00+atGE(4)
 -25 ENG → (-1,30 DEC)

?

LBL "délocalisés" et BG

A l'attention des "sportifs" du BG, nous allons nous entraîner ensemble.

Le BG étant ASN, ainsi éventuellement que PACK & DEL__ (c'est bien pratique) :

Mode PRGM

Ø1+LBL "T"
Ø2 ENTER
Ø3 "≠"
Ø4 "A"

GTO .ØØ1
PACK

PACKING
Ø2 W
Ø3 W

LIST ØØ1

GTO "A"

Ø2 "A"
Ø2 W
Ø3 DEL Ø4

GTO .ØØ1
PACK

PACKING
Ø2 W
Ø2 ENTER
Ø3 "≠"

GTO .ØØ1
PACK

PACKING
Ø2 W
Ø3 W

LIST ØØ1

GTO "A"
Ø2 "A"
Ø2 W
Ø3+LBL Ø6

GTO .ØØ1
PACK

PACKING
Ø2 W
Ø2 ENTER
Ø3 "≠"

PACK

PACKING
Ø2 W
Ø3 W

LIST ØØ1

GTO "A"

1) LBL "T" Ø1 LBL "T"
ENTER↑ Ø2 ENTER ↗
"≠" Ø3 "≠"
GTO .ØØ1 Ø4 LBL "T"
PACK BG BG ← ← SST Ø5 GTO "
CAT 1 (pour sortir de derrière le rideau !)

2) LBL "T" Ø1 LBL "T"
ENTER↑ Ø2 ENTER ↗
"≠" Ø3 "≠"
"A" Ø4 "A"
GTO .ØØ1 Ø5 LBL "T"
PACK BG BG ← ← SST Ø6 GTO "A"

3) ENTER↑ Ø1 ENTER ↗
"≠" Ø2 "≠"
"a" Ø3 "a"
GTO .ØØ1 Ø4 LBL "T"
PACK BG BG ← ← SST Ø5 LBL "a"

4) LBL "T" Ø1 LBL "T"
"A" Ø2 "A"
BST BG SST ← LBL Ø6 Ø3 LBL Ø6
GTO .ØØ1 BG ← SST ← Ø4 LBL "T"
ENTER "≠" BST BST PACK Ø5 LBL "T"
BG BG ← ← SST Ø6 GTO "A"
LIST ØØ1 .END. REG 41

et vous tenez : GTO "↓" dont je vous avais parlé.

Je pense que vous trouverez facilement LBL "x"

XEQ "x" et dans la foulée W "x" mais pour cette dernière fonction, il me reste à découvrir son utilité éventuelle. Il y aura peut-être un adhérent qui nous éclairera (?).....

à bientôt, peut-être, R.S.

Je crois pouvoir dire, contrairement à ce qu'affirment certains, qu'on doit pouvoir "tout créer" avec le BG

LBL "délocalisés" (suite)

A l'attention des non sportifs du BG et pour ceux qui ne veulent pas utiliser (?) les fonctions découvertes dans l'OI N°46 P139 ou pour ceux qui veulent faire plaisir à Lionel (?), j'indique les codes DEC à introduire avec "LB" pour obtenir des LBL, GTO et XEQ délocalisés.

Je dois préciser que j'écris ces lignes à l'attention des débutants et je demande aux "chevronnés" de ne pas sourire mais d'essayer de se souvenir du temps où "ils ne savaient pas encore".

```
XEQ "LB"  
192.000  RUN  
0.000   RUN  
242.000  RUN  
1.000   RUN  
127.000  RUN  
        RUN
```

```
LIST 001
```

```
02+LBL "P"
```

```
XEQ "LB"  
29.000  RUN  
241.000  RUN  
127.000  RUN  
        RUN
```

```
LIST 001
```

```
02.GTO "P"
```

```
XEQ "LB"  
192.000  RUN  
0.000   RUN  
247.000  RUN  
1.000   RUN  
114.000  RUN  
111.000  RUN  
98.000   RUN  
101.000  RUN  
114.000  RUN  
116.000  RUN  
        RUN
```

```
LIST 001
```

```
05+LBL "robert"
```

192,0,242,1,127 permet d'obtenir LBL "P"
C0,00,F2,01,7F (en HEX)

Les 4 premiers ½ octets sont constitués de C (12) suivi de 12 bits destinés à recevoir l'indication de la position du LBL dans la MEM.

Le 3^{ème} octet est le préfixe destiné à introduire la chaîne ou le caractère désiré.

Le 4^{ème} octet sert à enregistrer une ASN (code de la touche désignée)

Le dernier octet correspond au caractère désiré et il est évident (?) qu'il ne faut pas introduire, par exemple, 80 (P comme Paris) car il est beaucoup plus simple de faire: PRGM LBL "P" directement.....

Si vous désirez un LBL avec plus de 1 caractère, vous devrez modifier la valeur correspondant au 3^{ème} octet. Exemple: 192,0,247,1,114,111,98,101,114,116 (un peu narcissique, robert !).

29,241,127 permet d'obtenir GTO "P" et un coup d'oeil sur la TABLE des CODES (notre Bible) vous indiquera comment obtenir XEQ "P".

A quoi peuvent servir ces LBL délocalisés qui semblent ne pas pouvoir être obtenus avec XROM 205,0 etc... ?

On peut les utiliser pour une simple considération d'esthétique mais aussi, si l'on veut faire correspondre la touche ASN avec le LBL (exemple: LBL "R↓" pour faire "descendre" la PILE d'un cran). Je ne pense pas que vous ayez recours, un jour, à ces LBL, suite à l'épuisement des autres LBL.....

à bientôt, peut-être R.S. (244;82,46,83,46)

- E et LBL délocalisés

Le PRGM du dernier CI N° 45 Page 139 m'a permis de découvrir des fonctions qui me semblent intéressantes et que je vais essayer de vous faire découvrir.

- E équivalent de -1

Les codes DEC sont 28,27 et permettent effectivement d'obtenir - E avec le PRGM "LB" de Lionel mais le même résultat peut être obtenu au moyen de 2 fonctions ASN à 2 touches . Ces fonctions sont représentées par : \neq ae2 & Hb dont les codes sont respectivement :

1,28 1,27

Profitez de la présence du PRGM "KA" dans la MEM principale pour ASN toutes les fonctions dont les codes sont compris entre 1,16 & 1,31 ; vous disposerez ainsi d'une véritable panoplie de BUGS dont je n'ai pas trouvé toutes les applications.

Assignez donc les 2 fonctions précitées à 2 touches voisines ou à la même touche (une fonction sur la touche shiftée, dans ce dernier cas, bien entendu).

Je vous prends par la main et suivez moi:

LBL "T"	Ø1 LBL "T"
\neq ae2	Ø2 2 --
retaper sur la mm touche	Ø2 -
Hb	Ø3 H --
retaper sur la mm touche	Ø3 E
SST	Ø4 "
←	Ø3 - E (bravo !)
BST	Ø2 LBL "T"

Ne nous affolons pas, nous sommes encore dans le domaine d'Alice.....

CAT 1 et nous retrouvons comme par miracle :

Ø1 LBL "T"
Ø2 - E

Je pense qu'il était inutile de vous dire que ces opérations ne sont possibles que dans un PRGM "ouvert" vers le bas, c'est-à-dire non packé par un GTO .. et "fermé" par le haut, ce qui est réalisé par le LBL "T" dans l'exemple ci-dessus

USER TEST
17 XPRM 28,63
-11 ATOX
12 Ø+AHHH
-12 XROQ 52,00
13 OD
-13 DCABEE+æGØH>+
14 DEL
-14 P
15 Hb
-15 =æe2
21 PCKK
-21 \neq æe2
22 Hg
-22 P
23 HLLD
-23 Aa
24 P
-24 Aa
25 M
-25 æ+ædc'>52-\$T+HvIAa
32 X
33 -

Voir not
page 5

Ø2 GTO "T"
Ø3 X
Ø3 Ø
Ø3 DCABEE+æGØH>+
Ø3 P
Ø3 Hb
Ø4 \neq æe2
Ø3 =æe2
Ø4 XEQ ""
Ø5 P Ø7
Ø5 HLLD
Ø5 Aa ""
Ø5 M
Ø5 æ+ædc'>52-\$T+HvIAa
Ø
Ø5 Y
Ø5 -

PRP "T"

Ø1 LBL "T"
Ø2 GTO "T"
Ø3 W ""
Ø4 XEQ ""
Ø5 ""
Ø6 END

Je n'ai pas dit que ce procédé était le plus simple mais je laisse à d'autres le plaisir de nous indiquer comment ils pratiquent.

LBL, GTO et XEQ délocalisés

Si vous avez ASN toutes les fonctions indiquées précédemment, vous disposez de :

ⓐ ◆ AHMH (1,29) & ⓐ NQ (1,50)

Il vous faut également la XROM 52,00 (205,0)ASN à une touche.

L'utilisation de ces 3 fonctions est très facile et, à mon avis, beaucoup plus facile qu'avec "LB". «Excuses - moi, Lionel !» (Bon PRGM est tout de même "SUPER")

01+LBL "T"

XEQ :

En mode CALCul :

NONEXISTENT

02 GTO "e"

XEQ "A"

NONEXISTENT (on s'en fout)

XEQ :

NONEXISTENT

03+LBL "e"

Mode PRGM

01 LBL "T"

PRGM

04 Hb 05

XROM 52,00

02 LBL "A"

PRGM

05 #ae2 05

04 #ae2 05

05 Hb 05

Le même procédé utilisé avec les 2 autres fonctions ASN permet d'obtenir GTO et XEQ "A" (par exemple).

A ceux qui penseront qu'ils ne voient pas l'utilité de ces nouveaux LBL, je répondrais qu'ils n'ont pas bien réfléchi.....

NONEXISTENT

07 XEQ "e"

PRP "T"

Si cela vous intéresse, je vous indiquerais comment obtenir ces LBL, GTO & XEQ avec le BG mais cette proposition ne s'adresse bien entendu, qu'aux débutants (dont, je précise, je fais partie) aimant "cela", donc "sportifs".

01+LBL "T"

GTO "e"

03+LBL "e"

- E E XEQ "e" .END

Ces LBL.... peuvent également être obtenus avec "LB" et les codes DEC sont : 29,1,97 & 30,1,97 pour respectivement : GTO "a" & XEQ "a". Vous retrouverez dans la TABLES des CODES les codes correspondant aux caractères désirés et vous vous apercevrez alors qu'il est possible également d'utiliser des LBL du genre : LBL "\$" ou XEQ "↓" Je vous laisse le soin de découvrir les codes des LBL "délocalisés" (?).

USER KEYS:

11 R (1,16)

-11 a+RdC)52-\$T+HPIA0

12 R (1,18)

-12 W (1,19)

13 DCABEEa&GB1+W *

-13 Y (1,21)

14 HLLI (1,22)

-14 A0 (1,23)

15 #ae2 (1,24)

-15 00 (1,25)

21 - (1,26)

-21 Hb (1,27)

22 #ae2 (1,28)

-22 #AHMH (1,29)

23 #NQ (1,30)

-23 #ae2 (1,31)

01 "KA"

Au plaisir de lire vos découvertes et peut-être à bientôt; R.S.

4,29 semble équivalent à 1,29 et vous découvrirez (peut-être) la fonction MU (μ^{Γ}) qui.....

* 13 DCABEEa&GB1+W

*faites des impressions plus forcées
Note de l'imprimeur!*

11 1,16

-13 1,21

04 11 a -23 1196/13

Les chercheurs (?) ont essayé d'analyser le contenu des chaînes utilisées avec STOFLAG.

Reprenons la chaîne : 247,31,241,128,Ø,2,64,8 et, en mode PRGM, utilisons BG. Nous obtenons les octets suivants :

```
Ø1+LBL "T"
Ø2 W ""
Ø3+LBL Ø1
Ø4 +
Ø5+LBL Ø7
Ø6 .END.
```

```
W "" (il est bien là, notre code 31 !)
LBL Ø1 (code 2)
+ (code 64)
LBL Ø7 (code 8)
```

Seuls les codes 241, 128 et le NUL ne sont pas retrouvés. Nous savons que 241 est constitué de F complété par les 4 bits correspondant aux 4 premiers drapeaux et que 128 correspond aux 8 bits suivants.

Il semble donc que W "" intégrerait les 2 octets suivants, ou tout du moins les 3 digits contenant l'état des 12 premiers drapeaux.

Malheureusement, je ne peux pas vous en dire plus. au plaisir de vous lire, R.S.

STOFLAG et PS

Nous pouvons utiliser STOFLAG du XF pour placer les 44 premiers drapeaux dans une configuration déterminée. Il suffit de générer la chaîne de caractères adéquate et de placer en X les codes correspondants.

Le 1^{er} octet est toujours 31 (DEC); "tiens, tiens, W^T fait surface !".

Le ½ octet suivant (digit) est toujours F et les 11 ½ octets suivants contiennent la position des 44 drapeaux.

```
XEQ "T"
Ø1+LBL "T"
"X<>["
"q+" ***
STOFLAG
END
```

Soit à lever les drapeaux suivants :

```
SF Ø3 Ø4 26 29 4Ø
```

Représentons en BIN chacun des ½ octets sup à Ø:

```
0001 1000 0010 0100 1000
  ↑   ↑   ↑   ↑   ↑
  Ø3  Ø4  26  29  4Ø (drapeaux)
```

Construisons notre chaîne complète en HEX, sans oublier les ½ octets à zéro :

```
1F F1 8Ø ØØ Ø2 4Ø Ø8
```

Convertissons ces octets en DEC :

```
Ø31 241 128 ØØØ ØØ2 Ø64 ØØ8
```

et, au moyen de "LB", générons la chaîne ALPHA : 247,31,241,128,Ø,2,64,8

```
X<>M STOFLAG
```

et vous avez placé vos 44 drapeaux dans la configuration désirée.

Signalons que pour cette manip. , 11 octets sont

```
Ø1+LBL "T"
Ø2 "X<>["
Ø3 X<> [
Ø4 STOFLAG
Ø5 END
```

```
Ø2 "X<>["
```

STOFLAG et PS (suite)

Ceux qui ont eu la curiosité d'essayer de remplacer X<>M par ASTO X ont eu la surprise d'avoir

DATA ERROR .

En effet, le transfert d'une chaîne Non Normalisée entraîne la modif du contenu de cette chaîne et la bande ci-jointe vous indique que le "8" du dernier octet a été oublié ainsi que le "ø" (NUL) qui était placé entre "128" & "2".

Mise à part la suppression des nuls qui est indiquée dans le manuel de belle-mère HP, l'origine des autres modif. n'est pas évidente et semblerait issue du code 31 (W") dont les "effets" restent à découvrir.

Toujours avec le PRGM "LB" de Lionel, faites la chaîne : 246,255,ø,ø,2,8ø,8 suivie de ASTO X

Mode CALcul, ô surprise, la fenêtre est vide !

ALPHA CLA ARCL X

et voila notre chaîne revenue, "Coucou, me revoilou !".....

PRX nous indique "┘", qui est le caractère correspondant au code 255 .

Etant incapable de vous donner une explication à ces "subtilités", je lance donc un SOS. Il se trouvera bien un (mais ne regardez pas votre voisin!) adhérent pour éclairer notre lanterne.....

R.S.

DUR ET MOU

Francis me signale qu'il cherche à vendre son imprimante HP-82143A au prix de 1600FF (à débattre) .Que tous les membres interressés le contactent :

FRANCIS ROZANGE

13, RUE PELEE

75011 PARIS

Coup d'oeil sur le REG c

Le REG c est réputé "dangereux" mais rien ne vous empêche de le visualiser, ne serait-ce que par simple curiosité.

En attendant des explications plus détaillées, nous allons donc examiner ce fameux REG.

Comme vous l'indique la bande ci-jointe, vous pouvez sans crainte faire RCL c puis STO M et PRA (si vous avez une IMP). Vous avez donc la représentation des 7 octets de ce REG plein de mystères et soyez rassurés, débutants, je n'en sais pas plus que vous mais nous allons ensemble jouer les voyeurs.

Nous ne voyons imprimé dans cet exemple que 6 caractères mais un coup d'oeil affûté dans la fenêtre, nous indique un espace situé avant le 1^{er} caractère. Ceux qui n'ont pas la chance d'avoir le MF, prendront la TABLE des CODES et découvriront les valeurs DEC correspondant à chaque caractère.

La conversion en HEX donnera :

20 50 01 69 1F 81 97

et signalons qu'il n'est pas nécessaire de faire ces conversions puisque la TABLE des CODES permet de lire les codes en HEX (regardez donc autour de chaque tableau).

La conversion en BIN peut être faite directement mais le passage en HEX permet d'obtenir, d'une façon plus claire, la valeur (qui peut être une lettre) correspondant à chaque ½ octet.

Nous n'irons pas plus loin aujourd'hui car je n'en sais guère plus..... R.S.

Bravo PÉPÉ !

Je tiens à féliciter bien vivement notre collègue Daniel JACOB pour son merveilleux PRGM "PP" qui a été accueilli avec enthousiasme par tous les fanas de la "41" réunis chez Jacques VAUCELLE le Samedi de Pâques.

Je suis très intéressé par les GTO compilés et je demande donc à "ceux qui savent" de nous initier à cette technique qui semble, pour l'instant, réservée à qq. initiés.

Donc au plaisir de vous lire, R.S.(P20)

TRC

-0.300169258-03

0=i=0

SCI 9
RCL c

STO I
FIX 0
ATOX
32, ***
ATOX
48, ***
ATOX
1, ***
ATOX
105, ***
ATOX
31, ***
ATOX
129, ***
ATOX
151, ***

0=i=0

Registre ALPHA, affichage et REG internes.

Le REG ALPHA est constitué de 5 REG complets M,N,O et de 5 octets du REG P.

Nous allons travailler de droite à gauche et la figure ci-jointe en dira plus qu'un long (et mauvais) bavardage.

$(5 \times 7) + 5 = 24$ octets correspondent à ce que nous pouvons entrer dans le REG ALPHA que nous connaissons.

Affichage

La fenêtre ne permet la visualisation que de 12 caractères.

Si après le 8) nous faisons : CLA RCL M

nous aurons : OMBENT? et : CLA RCL N nous donnera : C

Je pense que maintenant vous êtes en mesure de jongler avec les REG internes constituant le REG ALPHA .

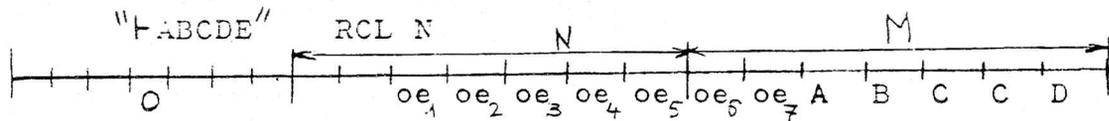
Signalons que les 4 autres octets du REG P peuvent également être utilisés, ce qui nous fait $4 \times 7 = 28$ caractères; mais pas trop longtemps, car notre chienne se sert de ces 4 octets à d'autres fins.

Exemple d'utilisation

Si nous faisons : CLA RCL e STO M nous aurons :



et si nous voulons isoler les 5 premiers octets :



Nous ayons ainsi isolé les $5 \times 8 = 40$ premiers bits du REG e . Ce ne sont peut-être pas ceux dont vous avez besoin mais je pense que maintenant vous pouvez vous débrouiller.

Je tiens à renouveler toute ma reconnaissance à Daniel GEORGELIN de Quiberon qui a eu la gentillesse de m'initier avec beaucoup de compétence et de gentillesse. Le tableau (croquis) ci-joint est d'ailleurs une copie de celui qu'il m'a envoyé. Donc, un grand merci à Daniel !

- 01+LBL "VY"
- 02 FIX 0
- 03 SF 29
- 04 CF 20
- 05 "XXXXXX ?"
- 06 PROMPT
- 07 CLA
- 08 ARCL X
- 09 "+ABCDEF"
- 10 RCL \
- 11 CLA
- 12 STO I
- 13 PROMPT
- 14 END

Ci-contre: petite routine permettant de supprimer la virgule inutile sur tout nbre de 6 chiffres au maximum. On peut faire mieux et + court avec AROT & ATOX mais il faut disposer du XF . Cherchons pour + de 6 chiffres (sans XF).

à bientôt, peut-être....

R.S.

Mode ALPHA CLA

0) 5 7 7 7 (fenêtre)
P O N M

1) Tapons C C _

2) Tapons O C O _

3) Tapons M C O M _

4) Tapons M C O M M _

5) Tapons E C O M M E _

6) Tapons N C O M M E N _

7) Tapons T C O M M E N T _

8) Tapons ? C O M M E N T ? _

9) T R E S F A C I L E M E N T , O K ? _

Stockage dans les REGistres-"normaux"

Nous pouvons stocker dans les REG normaux, c'est-à-dire ceux dont on dispose après avoir indiqué le SIZE désiré (SIZE 009 = 9 REG de 00 à 08, par ex.), des valeurs alphanumériques.

Stockage des valeurs numériques

Lorsque nous stockons un nbre. dans un REG, notre Diva ne le place pas dans le REG indiqué sous la forme présentée à l'affichage après STO rr ou STO IND ii .

Si nous tapons en mode CALCul 12345 STO 00 . cette valeur deviendra :

0 | 1 2 3 4 5 0 0 0 0 0 0 | 0 4

Signalons qu'un REG comprend 7 octets ou 14 ½ octets.

Le 1^{er} ½ octet indique le signe (0 pour + & 9 pour -).

Les 11 ½ octets suivants contiennent la mantisse du nbre, et les 2 derniers ½ octets contiennent la valeur de l'exposant en notation SCI.

Notre "41" utilise donc SCI 9 pour le stockage des val. avec une particularité que vous avez tous remarquée : la virgule est toujours placée entre le 1^{er} et le 2^{ème} chiffre. Cette disposition astucieuse permet de savoir "gratuitement" où est placée la virgule.

<<J'en vois un qui lève le doigt.>>

Si vous rentrez 0,12345678 cette valeur sera stockée sous la forme : 0 | 1 2 3 4 5 6 7 8 0 0 0 | 9 9

<<Baissez le doigt, élève Phénix !>>

99 n'est pas une erreur mais simplement le complément à 100 de l'exposant. La valeur rentrée affichée en SCI 9 est bien 1,2345678-01 (0,123456780 en FIX 9) mais est stockée de la façon précitée et la chienne y retrouve ses petits....

0,12345678 .10⁻²²

sera représenté à l'affichage :

-1,2345678-23

et sera stocké sous la forme :

9|1 2 3 4 5 6 7 8 0 0 0|7 7

Remarquons que chaque chiffre de la mantisse est un entier compris entre 0 & 9 (excepté le 1^{er} qui n'est jamais inf. à 1) qui occupe un ½ octet et, étant donné que sur 4 bits on peut compter jusqu'à 15 : <<QUEl gâchis !>>

« Elève Goupil, vous allez trouver le moyen d'utiliser ce rab. » On peut effectivement stocker plus qu'une valeur numérique dans un REG mais malheureusement d'autres y ont pensé avant moi. Crampez-vous et vous allez trouver le système mais pour vous mettre sur la voie, je vous indique que ce n'est pas au niveau du 14 octet qu'il faut "travailler" mais au niveau de l'octet. Nous pouvons placer dans chaque octet un code ASCII correspondant à un entier positif compris entre 0 & 255. Vous voici sur des rails qui ne mènent pas à une voie de garage..... Ceci n'étant qu'un exemple car si vous travaillez directement en BIN, vous aurez un autre itinéraire mais pour cela, maîtrisez bien les opérations dans ce mode et principalement, les soustractions et vous devrez bientôt voir la gare d'arrivée (je ne suis pas cheminot).

Stockage ALPHA

Nous disposons toujours de 7 octets ou 14 1/2 octets.

Lorsque notre chienne place une chaîne ALPHA dans un REG, cette chaîne peut contenir des chiffres ou (et) des lettres ou caractères. Il faut donc qu'elle place un repère qui lui permette de savoir qu'il s'agit d'une chaîne ALPHA. Elle utilise donc le 1^{er} octet à cette fin.

Nous comprenons tout de suite pourquoi nous ne pouvons stocker que 6 caractères dans un REG.

Le 1^{er} octet "étiquette" indique donc une chaîne alphanumérique de la façon suivante :

Le 1^{er} 1/2 octet est toujours F et le second représente le nbre. de caractères que contient la chaîne.

ROBERT sera stocké sous la forme :

α	R	O	B	E	R	T	
F'6	52	4F	42	45	52	54	(HEX)
246	082	079	066	069	082	084	(DEC)

Remarquons que 246 est TEXT 6 dans la TABLE des CODES qui devient notre bréviaire.

Guez sera stocké sous cette forme :

(oui, je sais, je fayotte !)... ..

α	G	u	e	z			
244	071	117	101	122			(DEC)

```

PACKING
      XEQ "LB"
242,000 RUN
183,000 RUN
      RUN
LIST 001

```

-241 devant supprimer
le "losange". Essayez!

```

03 *
**
LIST 001
02 *
*****
Codes?

```

Si vous avez compris, n'attrapez pas pour autant la grosse tête et pour vous en convaincre, essayez donc la chaîne suivante avec le PRGM "LB" de Lionel ANCELET : XEQ "LB" 242 R/S 183 R/S R/S vous obtiendrez dans la fenêtre : \overline{TK} mais avec l'IMP une chaîne sup. à 15 caractères (?). Je laisse le soin à un adhérent plus doué que moi, de développer le sujet, c'est-à-dire, le "pourquoi et le comment ?". Les "sportifs" pourront obtenir une chaîne identique avec le BG ; c'est très enrichissant mais je crois qu'il faut être un peu maso. Je pense que je dois l'être..... à bientôt peut-être, si vous aimez ça... R.S.

Mon cher Robert en réponse à une de tes lettres, les quelques assignations que tu me demandes:

```

CLKEYS
ASN "AK" 11
      AK
      AK
PRKEYS

```

```

USER KEYS:
11 AK
12 XROM 28,63
13 -PRINTER-

```

programme d'assignation contenu dans le NFCROM assignation des codes décimaux 167, 63 (en hexa A7 3F) pour assigner le BG à autre chose que LN tu peux faire les codes F7 40; pour générer les autres prendre: 41, 42, ... tous les BG ont la même utilisation.

```

PACKING
01+LBL "BG"
02 RCL IND 16
03 RDN
      GTO .001
02 W
      DEL 001
      PRP ""

```

impression après l'utilisation du BG

Je n'ai pas trouvé d'utilisation des codes A7 3F donnant le XROM 28,63

```

01+LBL "BG"
02 RCL I
03 .END.
      CLP ""

```

```

PACKING

```

bonne programmation
PHILIPPE
R/S

Bien cher Philippe,

J'avais indiqué PAGE8 du N°1 de JPC que :
"si nous voulions que notre Club "prolifère", il
fallait aider les débutants".

Je me permets de donner quelques précisions à
ce sujet et essayer d'apporter ma modeste contribution
pour l'éventuelle application de la thèse que je
propose.

Je dois donc préciser que si je souhaite que
PPC-PC ait le plus d'adhérents possible, ce ne doit
être au détriment d'aucun autre Club et en particulier
du PPC-T auquel j'adhère sous le N° 178.

Il est tout à fait hors de question que j'en-
tame une polémique entre les "unionistes" et les
"séparatistes" car chaque partisan a ses propres argu-
ments pour défendre son point de vue mais je pense
qu'une des premières qualités d'un Homme digne de ce
nom, est de respecter les opinions d'autrui.

Je vais essayer de résumer mon point de vue
sur la question :

émulation : OUI

rivalité : NON

coopération : SOUHAITABLE

en précisant qu'une UNION est tout à fait possible et
j'indique à ses partisans, qu'ils ont la possibilité
d'oeuvrer dans ce sens car ils doivent savoir que
dans des Clubs "loi de 1901", les Présidents sont
bien loin d'avoir tous les pouvoirs.

Donc, pas de critiques stériles ni de propos
indélicats ou mesquins mais des actes réfléchis et
constructifs ! De mon côté, j'essaierai de garder
cette ligne de conduite.

Passons au problème beaucoup plus passionnant des
"Débutants".

Etant moi-même un débutant en PS, je pense con-
naître les problèmes auxquels se heurte tout débutant.

<<Me voici en possession d'un merveilleux PRGM
que je souhaiterais vivement utiliser mais, ô honte !,
je n'arrive pas à générer cette mystérieuse fonction
X<>IND L ! Ah, cette PS !>>(?)

J'estime qu'il n'y a aucune honte à avoir dans
quelque cas que ce soit et, comme personne n'a la
science infuse, toute personne dite chevronnée, est
passée par le stade de l'ignorance, ce dernier mot
ne devant avoir aucun sens péjoratif.

Si notre collègue arrive à vaincre ses éventuels
"complexes", il va écrire à son Club "préféré" pour
obtenir la solution à son problème. S'il doit attendre
la réponse dans le prochain N° du Journal, il y a fort
à parier que, soit il aura trouvé seul la solution
avant de la lire, soit il aura perdu son calme (pour-
tant indispensable à tout PRGMmeur) et il aura peut-
être piétiné de rage sa chère (dans les 2 sens du mot)
"41".....

Pour éviter de pareils "sacrilèges", je te propose
mon cher Président de me charger de répondre directe-
ment à toute question concernant la PRGMmation
"classique", c'est-à-dire, PS exclue.

Sous réserve de m'adresser une enveloppe self-
adressée et affranchie, je m'engagerais donc, si tu

retiens ma proposition, à répondre par retour de courrier à toute question semblable à celle que j'ai donnée comme exemple et je vais même indiquer sa "réponse" :

```

XEQ ----- pp XEQ --
ALPHA ----- pp XEQ _
X ----- pp XEQ X_
SHIFT COS ----- pp XEQ X<_
SHIFT TAN ----- pp XEQ X<_>
ALPHA ----- pp X<> --
SHIFT ----- pp X<> IND --
. ----- pp X<> IND ST _
L ----- pp X<> IND L

```

<<Ne souriez pas, les "cracks", avez-vous réussi du "premier coup" sans difficultés ? >>

Il est bien entendu que je m'en tiendrais à des renseignements de ce genre et que je refuserais, par exemple, de chercher dans un PRGM, "pourquoi il ne tourne pas"..... Une question demandant un développement appréciable telle que :<<je ne vois pas ce qu'on peut faire avec X<>IND T ?>>, ferait l'objet d'une réponse publiée dans le Journal suivant, sans citer le nom de la personne qui l'a posée (ménageons les susceptibilités!...). J'espère que tout le monde aura compris le sens de ma démarche .

Je vous demande de m'excuser d'avoir été aussi bavard mais, comme vous le constaterez, je ne sais pas m'exprimer avec concision et je crois même, avec "simplicité".... et ce n'est plus à mon âge que je vais évoluer....

Je vous souhaite à tous, bonnes PRGMmations,

Robert SCHWARTZ

R.S. (P2Ø+T178)

141, avenue Carnot

93140 BONDY

P.S. : J'allais oublier de te remercier, mon cher Philippe, de m'avoir prêté cette merveilleuse PPC-ROM accompagnée de son bouquin de 492 pages. Une véritable "mine d'OR"! De quoi se priver de Guy Lux plus d'une soirée ! Si tu le désires, je peux te faire un "papier" sur cette ROM. Je vais l'acheter mais le contrôle des changes ne va pas faciliter les échanges (?) avec nos collègues US !...

01*LBL "CDXR"
02 OF 27
03 ACY
04 ENTER†
05 ENTER†
06 4
07 /
08 INT
09 160
10 +
11 "CDXR"
12 ARCL X
13 RDN
14 4
15 MOD
16 64
17 *
18 +
19 ARCL X
20 PROMPT
21 END
SIZE 000

20. ENTER†
63. SF 27
XEQ "CDXR"

CDXR=167,63,

=FAUX BG

44. ENTER†
0. SF 27
XEQ "CDXR"

CDXR=171,0,

= QL

5. ENTER†
3. SF 27
XEQ "CDXR"

CDXR=161,67,

= ORIC

1. ENTER†
53. SF 27
XEQ "CDXR"

CDXR=160,117,

= ROL M

PRKEYS

USER KEYS:

11 PACK
12 XROM 20,63
13 "CDXR"
22 eGOBEEP
51 XROM 44,00

Bien cher Philippe,

J'ai reçu avec beaucoup de plaisir le N°1 de JPC et je pense que maintenant, le train est mis sur les rails. Il ne reste plus à nos collègues, de maintenir la machine sous pression en envoyant le résultat de leurs travail ou de leurs découvertes.

J'indique que j'ai été surpris par la place que prenait ma prose. Je ne pensais pas en avoir écrit autant.....

Je vais me permettre quelques remarques qui ne seront en fait, que des conseils.

Il est tout à fait normal d'accepter le courrier non tapé à la machine étant donné que tout le monde ne dispose pas de cet outil, notamment les étudiants qui ne sont pas les moins "actifs" et "créateurs". Je conseille cependant d'écrire uniquement à l'encre NOIRE car cela se reproduit mieux.

A ceux qui tapent à la machine, j'indique qu'il faut de temps en temps, remplacer le ruban encreur. Ce n'est pas une grosse dépense et si vous faites du charme à une secrétaire, ce serait bien le diable qu'elle vous refuse cela.... Ayant une machine ancienne c'est-à-dire sans K7, je sais que ce n'est pas agréable et que cela n'est pas facile. Regardez bien où passe ce ruban avant de le retirer car un "montage" non prévu dans le manuel (?) risque de donner un résultat décevant. Malgré la publicité : tout le monde n'a pas fait Pigier !.....(moi le premier).

Je suggère que nous séparions distinctement le courrier non destiné à être publié, des textes qui sont d'intérêt général. Comme me l'écrivait un de mes correspondants : <<le Journal n'a pas besoin de rubrique "courrier du coeur">>

Je saute du coq à l'âne :

Les codes DEC à utiliser avec un PRGM genre "KA" pour ASN le BG que notre Président semble avoir baptisé -PRINTER- (personnellement, je préfère "Bouffeur d'octets"), sont 247,63. eGOBEEP s'obtient avec DEC-Ø,167 et Q-LOAD qui semble être l'équivalent du μ ou (μ F) français (?). J'ai trouvé cela dans "HP-41 Synthetic Programming Made Easy de Keith JARETT" mais je vous assure que je n'ai pas tout compris.....

A la demande de notre Président, j'envoie une copie du "papier" que j'ai fait sur la HP-16C pour le PPC-T Journal (je suis T178).

Les drapeaux 36 à 39 servent à stocker le nombre de DEC spécifié avec FIX, SCI ou ENG et il semblerait donc possible de demander par exemple, FIX 14 (?). Essayez donc ! Les autres questions "S.O.S." de JPC N°1 restent valables.

Je viens de faire à la hâte le petit PRGM "CDXM" qui permet de trouver les codes DEC correspondant aux XROM et j'indique à ceux qui ont la chance d'avoir un XFunctions, qu'il existe des PRGM "LB" & "KA" qui sont beaucoup plus courts et plus rapides que ceux que vous connaissez. Ils sont publiés dans le bouquin précité et, si vous êtes intéressés, je peux les proposer.

Je demande à ce que tous les termes (et à fortiori les phrases) anglais soient traduits car, si je ne me trompe pas, notre revue est française..... of course !

JPC

RECTIFICATION

Dans le précédent N° de JPC, j'avais indiqué la publication des PRGM "EDF" & "V" à la demande de notre PRésident.

Ayant fait ces PRGM pour une utilisation professionnelle dans le cadre de la Société qui m'emploie, je suis au regret de vous annoncer qu'il ne m'est pas possible de publier ces PRGM.

La législation du Travail considère comme faute grave : quiconque diffusera..... Je vous fais grâce de la suite et je pense que vous comprendrez que je ne veuille pas prendre de risques étant donné que ces PRGM utilisent une méthode et des formules qui peuvent être assimilées à des secrets (?).

Rassurez-vous cependant car je vous présenterai des PRGM qui ne seront pas susceptibles de tomber sous le coup de la LOI.....

Pour commencer, vous trouverez un petit PRGM qui m'a été donné par notre collègue Damien DEBRIL (P22) et qui peut remplacer "CLXM" publié P25 JPC N°1 et dont l'auteur est également notre collègue Patrick ZAGURY.

Je vous livre également un CATALOG 1 disons, "particulier" puisqu'il indique 1 028 REG de PRGM et je vous signale qu'on peut faire encore mieux.... Je crois même pouvoir vous dire que je vous ferai un CAT 1 qui vous donnera le vertige.....(?).

01+LBL "CXM"			3.615,	CF 28
02 "V"				SF 29
03 E			3.157,	+
04 CRFLD			6.772,	***
05 PURFL		CAT 1	171,	+
06 SF 25		3615 BYTES	6.943,	***
07 SEEKPT		LBL 02 3157 BYTES	21,	+
08 END		LBL "MKX"	6.964,	***
KAX		END 171 BYTES	123,	+
		LBL "CXM"	7.007,	***
	SAVEP	END 21 BYTES	54,	+
MKX		LBL "KAX"	7.141,	***
	SAVEP	END 123 BYTES	41,	+
	ENDIR	LBL "DC"	7.182,	***
KAX P018		END 54 BYTES	11,	+
MKX P025		LBL "CDXR"	7.193,	***
	XEQ "CXM"	END 41 BYTES	7,	
	ENDIR	LBL "T"	1.028,	***
DIR EMPTY		END 08 BYTES		
		.END. 03 BYTES		

R.S. (P20)

Philippe,

Cher ami, je t'envoie comme promis l'article sur les drapeaux que j'ai complété avec une petite boîte à outils très pratique pour démonter et mieux comprendre le mécanisme du registre d.

J'espère que cet article passera dans le numéro de Mai, car je l'ai écrit en particulier pour répondre au S.O.S. de Robert Schwartz du numéro d'Avril. Sans en attendre de compliment j'aimerais connaître ton avis sur cet article qui peut je pense être critiqué.

De mon côté je crois que certains reprints du système de la HP 41 mériteraient d'être expliqués longuement. A vos claviers...

A ce sujet tu as peut-être remarqué que je m'étais trouvé une secrétaire dévouée, qui soit me tape mes articles ou me prête sa machine.

J'admet que c'est agréable, pour une meilleure lisibilité.

Pourtant la non compréhension d'un texte nuit aussi à sa lecture, et dans le numéro d'Avril, beaucoup d'abréviations m'ont semblées très mystérieuses:

le CPU? etc... de toute manière je n'est rien compris du théâtre des microcodes quels sont ces registres, pas ceux de la machine ?

Enfin cela devrait venir à la suite des petits théâtres.

Le GTC compilé c'est quoi, ça se crée comment? C'est pour le numéro de Mai je crois.

Sinon pourrait-on obtenir des pages moins brouillonnes au sujet du matériel, et de certaines annotations sur les codes barres (C.B.). Je ne sais pas comment interpréter la page 25.

Erratum : Les LBL alphanumériques ne peuvent avoir au maximum que 14 caractères. C'est du moins ce j'en ai déduit après avoir longtemps disséqué cette fonction. Je demande à voir un LBL 15 CARACTERES.

Je vous enverrais peut-être un article sur ces LBL avec 2 programmes Morse2 (traduit 24 caractères en morse sonore)

APPRH 5 (écrit un message dans le sens de la longueur via l'imprimante en petites et grandes lettres. Limité seulement à la longueur du rouleau de papier)

Mais quand JPC (?) c'est SUPER!!!

même A

A bientôt : Yann DOLHEN

R/S.

Devant la difficulté, ne baissons pas les bras

Notre chère HP 41, nous le savons, possède de nombreux secrets. Beaucoup sont si complexes à expliquer que le secret persiste après les explications.

Cependant, je tiens à tenter une expérience avec vous tous, en dévoilant le mécanisme d'une des serrures les plus simples du système interne de la HP 41 : les drapeaux (ou flags).

Nombreux sont ceux qui savent déjà que le bloc des drapeaux est jalousement contenu par le registre d. Or, grâce à la programmation synthétique (P.S.), nous pouvons facilement programmer un RCL d.

Malheureusement, la valeur ainsi obtenue est particulièrement rebutante. On peut quand même, avec cette valeur, exécuter de nombreux calculs (sinus, logarithmes.. etc...) mais, essayer d'en faire l'inverse ($1/x$ rapidement).

L'affichage disparaît. Dès lors il y a deux solutions :

-enlevez et remettez les piles et appuyez par exemple sur ENTER[↑]. Ceux qui n'auraient pas trouvé cela amusant peuvent ne pas recommencer mais sinon essayez encore et au lieu de taper ENTER[↑], appuyez plusieurs fois sur SHIFT (touche jaune). Où est donc l'affichage ?

-ou alors rentrez le programme : *(er attendez)*
LBL PAUSE, PAUSE, AVIEW, RCL d, $1/x$, BEEP. *(des fois Alpha Data)*
qui vous montrera que la séquence RCL d, $1/x$ provoque une pause de 23'36" que l'on peut vérifier avec le chronomètre du module TIME (si on le possède bien sûr).

Passons sur ce seul amusement que peut nous procurer la valeur de d et abordons un domaine plus sérieux.

Introduisez le programme : LBL ESSAIS, CLA, RCL d, STO M, AON, STOP.
En l'exécutant vous pouvez obtenir de 7 caractères à rien du tout (7 octets nuls). De toute manière, il est temps d'en venir aux explications théoriques.

On obtient donc en ALPHA une chaîne de la forme :

$\alpha_1 \mid \alpha_2 \mid \alpha_3 \mid \alpha_4 \mid \alpha_5 \mid \alpha_6 \mid \alpha_7$

où chaque caractère a un code Ascii compris entre 0 et 255 qu'il faut décomposer en binaire (base 2) exemple pour $\alpha_1=Y$

Le code Ascii de Y est 89, en base 2, $89 = 01011001$ (toujours 8 chiffres ou digits) qui correspond à l'état de 8 drapeaux, pour α_1 les drapeaux 0 à 7.

On a 7 caractères possibles ($7 \times 8 = 56$ flags) et donc tous les drapeaux sont représentés dans le registre d (voir tableau).

Remarque : Attention dans tous les cas, il faut prendre la notation suivante (exemple pour une chaîne de 6 caractères : α_1 n'est pas considéré).

Affichage ALPHA (attention au sens de la flèche " ← drapeaux ")
"droite" | X | α_2 | α_3 | α_4 | α_5 | α_6 | α_7 | "gauche"
| 0 à 7 | 8 à 15 | 16 à 23 | 24 à 31 | 32 à 39 | 40 à 47 | 48 à 55 | ←

Ainsi pour lever uniquement des drapeaux de 48 à 55 un seul caractère alphanumérique suffit (inutile d'en mettre trop).

Par contre pour lever des drapeaux de 0 à 7, il faut un caractère suivi de 6 octets nuls.

Tableau des Drapeaux de la HP 41

α code	flag	remarques
α_1	128	00 usage général
	64	01 "
	32	02 "
	16	03 "
	8	04 "
	4	05 "
	2	06 "
1	07 "	
α_2	128	08 "
	64	09 "
	32	10 "
	16	11 exécution automatique
	8	12 double largeur (imprimante)
	4	13 enregistrement d'une carte protégée (lecteur de cartes)
	1	15
α_3	128	16
	64	17
	32	18
	16	19
	8	20
	4	21
	2	22
1	23	
α_4	128	24
	64	25
	32	26
	16	27
	8	28
	4	29
	2	30
1	31	
α_5	128	32
	64	33
	32	34
	16	35
	8	36
	4	37
	2	38
1	39	
α_6	128	40
	64	41
	32	42
	16	43
	8	44
	4	45
	2	46
1	47	
α_7	128	48
	64	49
	32	50
	16	51
	8	52
	4	53
	2	54
1	55	

spécia listes
à un ou plusieurs périphériques

normalement levé avec l'imprimante, sinon AVIEW devient PROMPT
levé dès que l'on tape une valeur) -numérique {Précaution : à baisser avant
-alphanumérique {chaque nouveau test.

saut d'une ou plusieurs erreurs de dépassement, sans se baisser
saut d'une seule erreur (de toutes sortes)
message sonore ou non
USER
affichage de la virgule décimale ou du point.
découpage par groupe de 3 chiffres (point ou virgule, inverse de 28)
levé lors de l'affichage d'un catalogue
périphériques)

Correspondance binaire de 0 à 32, codage des XROM

(nombre de chiffres après la virgule, codé en binaire
(levés : 39 = 1, 38 = 2, 37 = 4, 36 = 8)
particularité : compteur de 0 à 15 alors que FIX, SCI, ENG, de
0 à 9, mais 127 arguments possibles pour chaque (à connaître)

FIX
ENG dominant sur FIX) sinon SCI
GRAD
RAD dominant sur GRAD) sinon DEG

Affichage permanent obtenu avec ON
Affichage du tiret inférieur (introduction de l'argument) cumulable
IND particularité : annule une pression de n'importe quelle touche
SHIFT

ALPHA
Affichage de BAT
Message d'erreur ou affichage du canard
SST (inutilisable)
PRGM
d'après HP : entrée / sortie ???
PSE
Imprimante connectée

Il est très facile ainsi de modifier l'état des drapeaux en plaçant en alpha, la chaîne de caractères désirée (avec le module XFunctions par la fonction XTPOA ou encore à l'aide de la P.S. en générant la chaîne en programme et en faisant exécuter la ligne).

Ensuite on exécute le petit programme : LBL F, RCL M, STO d, qui réinitialise tous les drapeaux suivant la chaîne alphanumérique choisie.

Je vous soumetts quelques programmes parmi lesquels seul PRF est une création personnelle. J'ai modifié DISPLAY (2 pas en moins, et il conserve l'état initial des flags). Je ne connais pas les créateurs de DISPLAY et de FL mais je les félicite de leur travail et les en remercie.

```
01+LBL "PRF"
0 + RCL d ENTER↑
" 0123456789" SF 12
RVIEW 55 .009
```

```
11+LBL 00
CLX CF 29 FS? 55
CF 21 RVIEW FIX 0
ARCL L "F"
```

```
20+LBL 01
R↑ STO d RDN
FS? IND X "F"
FC? IND X "F" FC? 55
RVIEW X? GTO 02
ISG X GTO 01
```

```
34+LBL 02
SF 12 RVIEW .01 ST+ Y
CLX RCL T RDN ISG L
X<> X X<=Y? GTO 00 R↑
STO d CLST END
```

```
0 0123456789
0 *-----*---
1 *-----*---
2 -*-----*---
3 -----***
4 *-----*---
5 -----*
```

* LEVE
- BAISSÉ

```
01+LBL "FL"
ENTER↑ ENTER↑ ABS 3
ST+ Z MOD 16 + RCL d
STO [ SF 29 SCI IND Z
ARCL Y "F" X<> \
X<> d RCL Z X?
SF IND Z X<? CF IND Z
RCL d STO \ RDN ABS
INT 7 X<>Y -
FIX IND X ARCL X RCL ]
STO d END
```

[= M , \ = N ,] = 0

```
01+LBL "DISPLAY"
CF 21 "++0+"
RCL E
"++:++" ASTO Y ARCL Y
ARCL Y ARCL Y RVIEW
X<> d PSE PSE X<> d
CLX RTN END
```

[= N

FL lève tous les drapeaux sauf le drapeau 00

Faire : n
(0<n<56)
Exécuter FL

Si vous pensez que votre affichage est en panne, faites simplement :

XEQ DISPLAY
afin de vous assurer le bon fonctionnement de votre machine

```
( 03↑ FE 00 00 10 00 21 EB
  05↑ 10 BA 10 BA 10 BA
```

PRF fonctionne en SIZE 000
C'est simple mais il suffisait d'y penser
Remarque : PRF tournée avec ou sans l'imprimante

Une recommandation cependant : baissez tous les drapeaux que vous voulez mais attention en les levant. Dame 41 est parfois susceptible. Surtout quand on stoppe la machine en cours de programme. Prudence est d'attendre la fin de l'exécution.

J'espère en tout cas que ces explications vous auront appris et que cet article que je dédie à Robert Schwartz qui m'en a donné l'idée, vous aura beaucoup intéressé. Si certains n'ont pas tout compris, qu'ils m'écrivent à :

DOLHEN Yann 9 rue BERANGER 60200 COMPIEGNE

.R/S.

"DEUXIEME PAS"

Nous disposons du fameux BG et nous allons nous en servir.
 << Donnes-moi la main, collègue, avant de traverser ! >>

PRGM (affiché) GTO .. (SHIFT RCL . .) _____	(PACKING)
LBL "Vd" (SHIFT STO ALPHA V SHIFT LOG ALPHA) _____	Ø1 LBL "Vd"
RCL IND 16 (RCL SHIFT 1 6) _____	2 RCL IND 16
AVIEW (ALPHA SHIFT R/S ALPHA) _____	Ø3 AVIEW
BST BST PACK (XEQ "PACK") _____	(PACKING)
BG _____	Ø1 LBL "Vd"
← SST _____	Ø2 "A"-----E
RCL IND 17 _____	Ø2 RCL d
RDN (touche R↓) _____	3 RCL IND 17
BST BST PACK BG ← SST _____	Ø4 RDN
BST _____	Ø3 STO M
CLA (ALPHA SHIFT ← ALPHA) _____	Ø2 RCL d
SST _____	Ø3 CLA
PROMPT (XEQ ALPHA PROMPT ALPHA) _____	Ø4 STO M
	Ø5 PROMPT

La routine "Vd" nous permet de visualiser les CARactères placés en ALPHA et représentant les 7 octets du REG d qui contient l'ETAT des 56 drapeaux. Repérons les éventuels NULS représentés dans la fenêtre par un "surligné" (barre supérieure) et décodons les autres CAR au moyen de la fonction ATOX du XF (les codes obtenus sont DEC). Les 7 codes DEC seront convertis en HEX pour obtenir les 14 codes correspondant aux DIGits (½ octets). Chaque DIG (code HEX) sera converti en BIN qui représentera, sur 4 Bits, l'ETAT des drapeaux correspondants (1= "levé", 0= "baissé"). Si, par exemple, le premier DIG à Gauche est "2"= 0010, cela indiquera que le DRAPEau 03 était levé lors de l'exécution de la fonction RCL d. Ne pas oublier que le premier DRAP est 00.

Nous avons crée 2 FS (Fonctions Synthétiques) RCL d et STO M mais nous pouvons, de façon semblable, obtenir toutes les fonctions codées sur 2 octets.

Le premier octet (PREfixe) est toujours constitué de RCL suivi de ce qui est indiqué au centre des cases de la TABLE des CODES (notre bréviaire), cases dont les codes DEC sont : 144 à 191 ainsi que 206 et 207. Ces cases sont représentées sur fond jaune sur la très belle plaquette "HP-41 QUICK REFERENCE CARD FOR SYNTHETIC PROGRAMMING" qui a judicieusement, le format de notre "41".

Le deuxième octet (suffixe) est constitué de ce qui est indiqué en Haut des cases ayant les codes DEC 0 à 143, cases représentées sur fond blanc sur la plaquette précitée, exceptés 29 à 31 (DEC) qui sont représentés sur fond bleu.

Les codes DEC 0 et 27 à 31 sont "particuliers" et feront l'objet d'un prochain "papier".

Exemples : RCL IND 22 RDN nous donneront ISG M
 RCL IND 78 CLX d° d° X<>0
 RCL IND 27 X=Y? d° d° ARCL P
 RCL IND 31 LBL 07 d° TONE 8 mais cette

dernière fonction peut s'obtenir plus simplement (?).

PRGM GTO ..	(PACKING)
LBL "P"	Ø1 LBL "P"
"ABCDE" (ALPHA A B C D E ALPHA)	Ø2 "ABCDE"
BST BG	Ø2 "A"-----E
SST	Ø3 -
DEL 005 (5=5 CAR "ABCDE")	Ø2 "A"-----E
LN	Ø3 LN
ABS (XEQ ALPHA A B S ALPHA)	Ø4 ABS
PI (SHIFT 0)	Ø5 PI
FRC (XEQ ALPHA F R C ALPHA)	Ø6 FRC

y ^x (SHIFT 1/x)	07 Y↗ X
GTO . 001 (SHIFT RCL . 0 0 1).	01 LBL "P
PACK BG	02 " A
SST	03 STO 15
← ← SST	02 "PaMS
ENTER↑ENTER↑ENTER↑	05 ENTER↑
GTO . 001 PACK BG	02 " A
SST	03 LN
SST	04 ABS
SST	05 PI
SST	06 FRC
SST	07 Y↗ X
GTO . 001 PACK BG SST ← ← SST	02 "PaMS
« Coucou, me revoilou ! » (Polnareff)	
BST PACK BG SST	03 LN
SST	04 ABS
← RCL 14	04 RCL 14
SST	05 PI
← SST	05 FRC
← SST	05 Y↗ X
RCL 14	06 RCL 14
RCL 00	07 RCL 00
GTO . 001 PACK BG SST ← ← SST	02 "P.S.

Si vous n'avez pas compris ce qui s'est passé, recommencez et observez avec attention en consultant votre bréviaire; vous devrez finir par comprendre....

Pour créer une chaîne de CAR (15 au maximum), il suffit donc de faire une chaîne de nnCAR quelconques, BST, BG, SST, DEL 0nn, introduire les fonctions correspondant aux codes désirés, revenir au début, PACK, BG, SST, ←, ←, SST.

Le NUL du code 0 s'obtient simplement en introduisant une fonction quelconque (- par exemple) et de la supprimer après avoir "PACKé" (par XEQ "PACK"). Vous avez compris pourquoi je vous avais conseillé d'ASN à une touche la fonction PACK ainsi que DEL. PACK sert à supprimer les octets nuls "baladeurs" ou créés par des rectifications. DEL facilite l'effacement des fonctions mais peut être remplacé par autant de ← que nécessaires.

Le code DEC 27 est facile à obtenir :

PRGM (affiché) GTO .. LBL "EE"	01 LBL "EE
EEX	02 1 E
BST PACK BG ← SST	02 E

qui sera l'équivalent du EEX représenté dans la case 27 et que nous pouvons utiliser également pour remplacer "1" (c'est plus joli et il paraît que ça va plus vite ?)

BST "A"	02 "A
BST PACK BG SST	03 -
← BST PACK BG ← SST ← SST	02 "E

qui est le CAR Æ correspondant au code ASCII 27.

Le NEG du code DEC 28 ne s'obtient pas avec CHS et il ne faut pas confondre ce "moins" avec celui du code DEC 45 qui se crée avec RCL 13 .

PRGM (affiché) GTO .. LBL "-"	01 LBL "-
EEX CHS	02 1 E-
BST PACK BG SST	03 E-
BST BG ← ← SST	02 -
BST "C"	02 "C
BST BG SST	03 /
← BST PACK BG ← SST ← SST	02 "E

qui est le CAR œ correspondant au code 28 (1C en HEX).

Nous créerons les fonctions des codes 29 à 31 dans le N° 3 de JPC. A bientôt, R.S. (P20+T178)

Fonction W''

J'ai ramené de la Chasserie (voir article dans ce mm N°) un PRGM que m'a donné ce "sorcier" de la PRGMmation qu'est notre charmant ami, Jacques VAUCELLE.

Les "recherches" que j'avais faites sur la fonction W'' m'avaient simplement permis de trouver une application concrète pour la manipulation des 44 premiers drapeaux avec STOFLAG mais j'étais à peu près persuadé qu'il existait au moins une autre application de cette fonction ayant le code DEC 31. Mon manque de perspicacité ne m'a pas permis de découvrir ce que notre "enchanteur" avait mijoté... Je le remercie donc bien vivement de tout ce qu'il m'a appris et j'ai donc appelé ce PRGM "WJV".

Quelques indications sont utiles si vous désirez obtenir un résultat correct :
Le Pas 10 peut être obtenu facilement avec le PRGM "LB" mais je vous conseille d'écrire tous les Pas suivants (exceptés le END, bien entendu) avant de placer LBL "++", etc... Comme vous le constaterez (non curieux s'abstenir), cette fonction est pour le moins curieuse et, si elle semble "bouffer" des octets comme le BG, je me permettrais de dire qu'elle se contente de les "ruminer" pour les "régurgiter" ensuite. Vous m'excuserez, je l'espère, pour cette image un peu "vache" (?) mais je pense que ceux qui feront des essais me suivront dans mes raisonnements d'étables.....

Le CF 27 est destiné à désarmer le mode USER si vous ASN "WJV" à une touche; il est donc facultatif.

Les codes DEC 31 & 12 créent: W "-----" et vous pouvez ensuite "packer" par GTO .. puis supprimer les + inutiles (etc...) et repacker de la mm façon.

L'utilisation de ce PRGM avec une IMP nécessite le désarmement du drapeau 55 placé avant l'exécution de cette fonction. Je ne vous ferai pas l'affront de vous indiquer comment on désarme le "55"....

Je vous indique également qu'il ne faut pas placer un STOP après cette fonction car R/S (qui ne sont pas mes initiales) est une fonction bizarre, vous avez dit....

Le Pas 10 après GTO ... doit vous donner à l'affichage dans la fenêtre ~~MM~~ qui représente 12 octets.

UTILISATION (essais) :

ENTER↑ USER J (si "WJV" ASN/touche 25=J) et vous obtenez l'affichage fugitif de: GTO IND 00
puis: FINI "dans un silence lugubre"....
7 ENTER↑ 8 USER J et là, quel surprise !
Notre BEEP familier (si vous entendiez celui que m'a "fait" Jacques!....) suivi de l'affichage de FINI complété par RAD et les drapeaux 0 à 4 sont levés comme au passage du Général.....

Je vous laisse trouver ce qui se passe dans ces 2 cas et je suis sûr que vous trouverez des

```
PRP --
01+LBL "WJV"
02 CF 27
03 CF 00
04 CF 01
05 CF 02
06 CF 03
07 CF 04
08 DEC
09 X=Y?
10 W "
11 BEEP
12 SF 00
13 SF 01
14 SF 02
15 SF 03
16 SF 04
17 RAD
18 "FINI"
19 AVIEW
20 END
PAS 10: 1F 00 44 BYTES
/LB
10+LBL "++"
25 +
26 XROM "LB"
XROM "Lg"
DEC/HEX INPT
#1 OF 7?
#2 OF 7?
#3 OF 7?
RUN
RUN
RUN
SST. DEL 004
DEL 004
LIST 001
10 W "+++++"
GTO ..
PACKING
```

ENTER*
SF 27
XEQ "WJY"
FINI
7) ENTER*
8) SF 27
XEQ "WJY"
FINI
XROM "VF"
FLAGS SET:
0 1 2 3
4 21 26 29
40 43 50 55

applications à cette formule W".

Le code 31 peut être suivi d'une valeur comprise entre 0 (?) et 15 mais vous pouvez essayer au delà et j'indique aux "sportifs" du BG (y'en'a-t'il ?) que W" peut s'obtenir avec STO 31.

Ceux qui possèdent le XF peuvent remplacer les Pas 03 à 07 par , X<>F ou 0 X<>F mais ne faites pas comme votre serviteur: n'oubliez pas RDN ! C'est la farce involontaire que j'avais faite à notre ami Jacques qui "n'y comprenait plus rien".....

Merci de votre attention et à bientôt le plaisir de lire vos découvertes et vos applications sur cette curieuse fonction W".

R.S.(P20+T178)

Réunion à la Chasserie

Suite à l'aimable invitation de notre collègue, Franck LEBASTARD, contact régional Bretagne PPC-T, nous nous sommes retrouvés, une bonne poignée de fanas de la "41" chez notre ami Jacques VAUCELLE le Samedi de Pâques.

Je pense pouvoir avancer que tous garderont un très bon souvenir de cette rencontre et nous devons remercier bien vivement Jacques pour l'accueil chaleureux et éclairé qu'il nous a réservé. Merci également à tous ceux qui se sont déplacés ainsi qu'à ceux qui ont eu la malchance d'avoir un accident sur la route "de la Casserie".

Les membres PPC de Bretagne ont témoigné ainsi de leur profond intérêt à la vie du Club.
<<Bravo les Bretons !>>

Tous ont manifesté un parfait esprit de camaraderie et le vétéran de l'assemblée, votre serviteur, s'est senti tout rajeuni par la présence de tous ces jeunes (oui papa!...) d'une correction exemplaire et c'était plaisir à les entendre parler de leur passion commune.

Rassurez-vous, je ne me suis pas mis à l'écart et, bavard comme je suis, j'ai également participé aux conversations avec mon exubérance habituelle.....

Donc : un grand MERCI bien sincère à tous et vous pouvez compter sur ma présence si vous acceptez de me réinviter (?)....

Confraternelles pensées amicales à tous les "beaux-frères" de la Famille HP 41... R.S.(P20)

PACKING

OLKEYS
ASN *AK* 11
AK
AK
AK
PRKEYS

LES PREMIERS PAS DE BEBE (2° épisode)

Dans ce numéro nous allons voir quelques fonctions synthétiques connues des anglophones grace au livre de WICKES

LE TEXT ENABLER

USER KEYS

11 AK
12 XROM 01,53
13 XROM 05,53
14 XROM 05,62

23PR48488

CLA
CODE
.END.
CLA
STO C

R<ond

ASTO 00
GTO ..

PACKING

01+LBL *TEN*
02 *ABCDE*

GTO ..

PACKING

CLA
ARCL 00

R<ond

RCL C

0.0002-12 ***

AK

PRKEYS

USER KEYS:

11 AK
12 XROM 01,53
13 XROM 05,53
14 XROM 05,62
15 XROM 05,57

GTO ."TEN"
GTO .002

03 X<=Y?

04 Z+

05 Z-

PACK

PACKING

CLA
ARCL 00

R<ond

RCL C

0.0002-12 ***

GTO .002

STO -

STO d

02 *ABCDEFGH*

PRP *TEN*

01+LBL *TEN*

02 *ABCDEFGH*

03 END

Qui pourrait être traduit par "plus de texte" qui peut être un procédé intéressant pour combler les chaînes alpha en mode programme. En mode PRGM-ALPHA le CPU commence par écrire un byte (F1) correspondant à un texte contenant 1 caractère suivi par le byte de caractère dans la mémoire. Pour chaque caractère suivant, le CPU doit changer le byte de texte. L'information requise par le processeur est enregistrée dans le registre Q pour garder la suite des opérations. Les 4 premiers bits (1° nybble) de Q représentent le nombre courant de caractères dans la ligne. Les 4 derniers bits, l'adresse du dernier byte entré. D'autre part le drapeau 45 (data entry flag) est armé durant l'entrée de la ligne puis immédiatement désarmé à la première instruction qui termine l'entrée de caractères, si bien qu'il est désormais impossible de rajouter d'autres caractères à cette ligne. Grâce à la programmation synthétique, on peut armer ce drapeau en stockant le Nombre Non Normalisé approprié dans le registre d en particulier si on utilise STO d pour armer simultanément les drapeaux 45 DATA ENTRY 48 MODE ALPHA 52 MODE PRGM, il est possible de résoudre notre problème. Ce nombre NNN peut être appelé comme celui des anglophones TEN (text enabler number). Tous les TEN possibles conviennent selon l'état dans lequel vous désirez garder votre calculateur.

On pourrait choisir par exemple un NNN qui "0" comme premier nybble 488 pour les trois suivants. Attention, le registre d affecte les 56 drapeaux du calculateur nous devons donc choisir des nybbles bien spécifiques qui gardent là HP dans un état agréable. Le NNN utilisé dans l'exemple est donné par le code 00 00 02 3C 04 84 88, il arme les drapeaux 26

- 26 AUDIO
- 27 MODE USER
- 28 FIX 4
- 29 MODE DEGRE

Utilisez un programme CODE (HN PPC ROM) pour générer le TEN, on obtient en ALPHA une chaîne de caractères " * (* * " que nous pouvons stocker en 00 (ASTO 00) pour une utilisation future.

Si vous désirez utiliser le TEN, procédez selon les instructions suivantes

- SHIFT ALPHA CLA
- SHIFT ARCL 00
- SHIFT ALPHA RCL M, qui est meilleur que RCL 00, pour

éliminer le byte d'identification ALPHA que ASTO 00 ajoute au NNN. Maintenant faites GTO.. pour débiter un nouveau programme, mode PRGM puis entrez la ligne " ABCDE " si on désame le mode ALPHA après l'entrée de C, la ligne est terminée. Mettez vous en mode RUN puis exécutez à la lettre les instructions.

Après avoir entré les instructions des pas de programme 03 04 et 05 n'oubliez pas le GTO . 002 . STO Q STO d et pour entrer la chaîne appuyez sur un caractère quelconque vous lirez "ABCDEFGH" en continuant vous entrerez des caractères nuls.

01+LBL *E2*
02 I E2
PRP *E2*
GTO .002
CLA
ARCL 00
01+LBL *E2*
02 E2
03 .END.

R<ond
RCL C
GTO .002
STO -
STO d



LE Q LOADER (normé par l'OI)

Utilise aussi la propriété du registre de stocker temporairement des informations suite à l'enregistrement d'une chaîne ALPHA. Il est possible d'exploiter cette propriété pour le chargement de chaînes ALPHA-PRGM de 7 caractères. LeQ loader est l'assignation du chiffre 9 (9 pour la fonction μ) à une touche. Nous avons besoin d'assigner STO Q le Q LOADER (4,25 avec un programme d'assignation) Le Q loader est l'assignation du nombre 9 à une touche soit byte 19. l'assignation de tout byte de 10 à 1C marchera comme un Q LOADER mais 19 est facile à utiliser et donne un caractère facilement identifiable OD. en mode RUN l'utilisation du Q loader entre 9 dans le registre X mais en mode programme il entre 2 lignes la première un 9 mais la seconde une chaîne alpha; le contenu du registre Q. Pour l'utilisation de ce registre nous ne sommes pas limités dans les NNN possibles qui peuvent être générés par un programme CODE.

```

                                CLKEYS
                                GTO ..
PACKING
                                ASN "AK" 11
                                ASN "CODE" 21
                                AK
                                AK
                                AK
                                AK
                                PRKEYS
                                PRKEYS

USER KEYS:
11 AK
12 CODE
13 XROM 05,57
14 OD
01+LBL "QLOADER"
41 CLA
4544434
                                CODE
                                STO ..
02 OD
                                DEL 001
                                PRP **
                                01+LBL "QLOADER"
                                02 "ABCDE"
                                03 .END.
105040660
                                CODE
                                STO ..
                                DEL 001
                                GTO .002
                                PACK
                                PACKING
                                02 OD
                                DEL 001
                                GTO .002
                                PACK
                                CLA
                                ARCL 00
                                x<000
                                (RCL M) -> RCL I
                                (STO Q) -> STO -
                                STO d
                                02 "(#)"
                                PRP **
                                01+LBL "QLOADER"
                                02 "(#)"
                                (ASTO M) -> 03 ASTO I
                                04 .END.

```

Dans la génération d'une chaîne entrer le code en ALPHA en HEXADECIMAL et à l'envers de celle que l'on veut générer car le registre Q inverse toutes les chaînes. Puis transformer ce code en un NNN le charger en Q puis en mode PRGM presser le Q-loader (OD) effacer le 9 et lister le programme.

Dans le deuxième cas procéder de même avec le code 759A292328 puis utilisez le TEN sur la chaîne ainsi construite. Après le STO d appuyez trois fois sur un caractère quelconque puis faire un SST vous devez voir ASTO M dans votre programme.

Les assignations de :

- 04 1C 4/29 GTO(Alpha)
- 04 1D 4/30 XEQ(Alpha)
- 0D 00 205/00 LBL(Alpha)

LE ENHANCED BYTE JUMPER (eJUMP)

le défaut principal du byte jumper est l'impossibilité d'altérer le 2° byte d'une ligne programme. Pour cela nous devons feinter et utiliser un générateur qui permet de "cacher" le préfixe et insert le postfixe dans le programme. Il demande une double utilisation du CRIC. Cette méthode laisse beaucoup d'instructions à gauche qui peuvent être ensuite effacées ainsi que le générateur. L'impossibilité d'effacer le 2° byte vient du fait que les bytes sont normalement insérés dans le programme après le dernier, cette n'est pas appliquée si le numéro de la ligne programme est "00". Le ejump nous permet cette possibilité. La méthode la plus simple pour aller à la ligne 000 est d'exécuter en mode RUN un RTN, il place le pointeur programme au sommet du programme courant;



et la séquence suivante " PRGM off " ; " RTN " ; " RCL e " ; " GTO. xyz " ; " STO e " , changer le numero "xyz" de la ligne programme en un numéro fictif "00" . Le registre e contient le numéro de la ligne programme ainsi que le RTN suivant, il faut donc charger un numéro de ligne arbitraire, ici "00" . Se placer en mode PRGM et suivre la séquence qui donnera toujours à l'affichage " 00 REG xyz " . on obtient le même résultat avec :

" GTO. xyz " ; " RCL b " ; " RTN " ; " STO b "

Cette méthode simplifie l'introduction de nouvelles fonctions

```
eJUMP, xyz 1. PRGM off
            2. RTN
            3. RCL e
            4. GTO, xyz
            5. CRIC
            6. STO e
            7. PRGM on
```

Pour l'édition de programmes la première méthode (ci contre) est préférée au moins tant que le contenu du registre e n'est pas changé (par exemple, ne pas changer d'assignation de touche shiftée).

Pour illustrer l'utilisation du eJUMP :

```
ou 1. PRGM off
    2. GTO, xyz
    3. CRIC
    4. RCL b
    5. RTN
    6. STO b
    7. PRGM on
```

Création de ISG N

Les fonctions de 2 bytes autres que STO RCL et LBL sont plus faciles à créer 'ils sont initialement entrés avec un postfix de 00 le premier DEL n'est pas nécessaire. CLP ""

Création d'une chaîne de 9 caractères

Cette méthode n'est pas limitée à la création des fonctions de 2 bytes.

Création de RCL M.

```
PACKING
GTO ..
ASN "AK" 11
  AK
  AK
  AK
  AK
  AK
ASN "DEL" 22
  PRKEYS
```

```
PACKING
GTO ..
```

```
PACKING
01+LBL "eJUMP"
02 STO 01
03 ISG 00
GTO .003
W
RCL b
0.0000+= ***
RTN
STO b
GTO .002
DEL 001
PRP ""
```

```
PACKING
CLP ""
```

```
PACKING
GTO ..
```

```
PACKING
01+LBL "eJUMP"
02 STO 01
03 "ABCDEFGH1"
RTN
RCL e
0.0000 ***
GTO .003
W
STO e
DEL 009
```

USER KEYS:

```
11 AK
12 XROM 01,63
13 XROM 05,63
14 XROM 01,60
15 XROM 05,60
21 -PRINTER-
22 DEL
01+LBL "eJUMP"
02 STO 01
03 RCL 99
```

Tout numéro de registre > 15

```
GTO .003
RTN
RCL e
0.0000 ***
GTO .003
W
STO e
DEL 001
```

01 RDN

```
GTO .001
GTO .002
DEL 001
PRP ""
```

ATTENTION différent de ←

on peut aussi utiliser:

"SST" , "←" .

```
01+LBL 00
02+LBL 01
03+LBL 02
04+LBL 03
05+LBL 04
06+LBL 05
07+LBL 06
08+LBL 07
09+LBL 08
```

```
GTO .002
DEL 001
```

```
02 TAN
PRP ""
```

```
01+LBL "eJUMP"
"xyz←PR↓&0" .END.
```

01+LBL "eJUMP"

END

LE PETIT THEATRE DES MICROCODES

Faisant suite à l'article du mois dernier, nous verrons ce mois l'ensemble des mnémoniques utilisées par tous les membres du PPC.

Il existe quatre types de classe dans le microcode de la HP41

I - INSTRUCTIONS DE TYPE 0 : "DIVERS"

Mot type : $\overline{P_3} \overline{P_2} \overline{P_1} \overline{P_0} \overline{I_3} \overline{I_2} \overline{I_1} \overline{I_0} \emptyset\emptyset$
 bits de bits d'
 paramètre instruction

TABLE 0

I:	3	2	1	0	MNEMONIQUES	NOTES
0	0	0	0	0	NOP	fonction inoperante
0	0	0	1	0	CLRF f	effacer le drapeau f (note 1)
0	0	1	0	0	SETF f	armer le drapeau f (note 2)
0	0	1	1	0	?FSET f	armer le carry si le drapeau f est armé (note 3)
0	1	0	0	0	LD@R- d	mettre d dans C au pointeur R (note 4)
0	1	0	1	0	?R= f	armer le carry si le pointeur à f (note 5)
0	1	1	0	0	-----	voir table 0.6
0	1	1	1	0	R= f	placer le pointeur en f (note 6)
1	0	0	0	0	-----	voir table 0.8
1	0	0	1	0	SELP f	sélectionner le périphérique f (note 7)
1	0	1	0	0	WRIT r	écrire C dans la MEV
1	0	1	1	0	?FI f	test des drapeaux des périphériques
1	1	0	0	0	-----	voir table 0.C
1	1	0	1	0	-----	inutilisé
1	1	1	0	0	READ r	lire C de la MEV
1	1	1	1	0	RCR f	permutation circulaire à droite de C de f digits
					-note 1 : sauf 3C4 ST=0	effacer les registres d'état
					-note 2 : sauf 3C8 CLRKEY	efface le drapeau de clavier KEY
					-note 3 : sauf 3CC ?KEY	armer le carry si le drapeau KEY est levé
					-note 4 : retire 1 au pointeur après chargement	
					-note 5 : sauf 3D4 R=R-1	decremente le pointeur de 1
					-note 6 : sauf 3DC R=R+1	incrémente le pointeur de 1
					-note 7 : le CPU est mis en attente pendant l'exécution des instructions spéciales	

DEFINITION DES PARAMETRES

P:	3	2	1	0	d	f	r
0	0	0	0	0	0	3	0 (T)
0	0	0	1	0	1	4	1 (Z)
0	0	1	0	0	2	5	2 (Y)
0	0	1	1	0	3	A(10)	3 (X)
0	1	0	0	0	4	8	4 (L)
0	1	0	1	0	5	6	5 (M)
0	1	1	0	0	6	B(11)	6 (N)
0	1	1	1	0	7	E(14)	7 (O)
1	0	0	0	0	8	2	8 (P)
1	0	0	1	0	9	9	9 (Q)
1	0	1	0	0	A	7	10(+)
1	0	1	1	0	B	D(13)	11(a)
1	1	0	0	0	C	1	12(b)
1	1	0	1	0	D	C(12)	13(c)
1	1	1	0	0	E	0	14(d)
1	1	1	1	0	F	F(15)	15(e)

Exemples:

Adresse	Code	Mnémonique
000C	14C	?FSET 6
008F	0F0	C()N ALL
00A8	0D4	?R= 10

TABLE 0.6 : "DIVERS-INSTRUCTIONS DE CLASSE 0 - TYPE 6"

CODE HEX	MNEMONIQUES	NOTES
018	-----	inutilisé
058	G=C CR,+	place 2 digits de C dans G
098	C=G CR,+	place G dans C à R et R+1
0D8	C()G CR,+	échangerG avec 2 digits de C
118	-----	inutilisé
158	M=C ALL	remplacer M par C
198	C=M ALL	remplacer C par M
1D8	C()M ALL	échanger C et M
218	-----	inutilisé
258	T=ST	remplacer le registre Tone par le registre SStatut
298	ST=T	remplacer le registre ST par le registre T
2D8	ST()T	échanger les registres ST et T (chacun font 8 bits)
318	-----	inutilisé
358	ST=C X	remplacer ST par l'exposant de C
98	C=ST X	remplacer l'exposant de C par ST
3D8	C()ST X	échanger l'exposant de C avec le registre ST

-note : le registre ST (statut) contient les drapeaux
0 à 7, les autres peuvent être individuellement
armés ou désarmés.

TABLE 0.8 : "DIVERS- INSTRUCTIONS DE CLASSE 0 - TYPE 8"

CODE HEX	MNEMONIQUES	NOTES
020	XQ-)GO	échanger XQ en GO en plaçant le contenu du registre de la pile
060	POWOFF	mettre la HP 41C en mode d'attente (note) dans l'ADR de C
0A0	SLCT P	sélectionner le pointeur en P (R)
0E0	SLCT Q	sélectionner le pointeur en Q (R)
020	?P=Q	armer le carry si les pointeurs sont égaux
160	?LOWBAT	armer le carry si l'indicateur BAT est allumé
1A0	A=B=C=0	effacer les registre A, B et C
1E0	GOTO ADR	charger le registre PC (contient l'adresse)
220	C=KEY KY	charger
260	SETHex	mettre le CPU en mode HEXadécimal
2A0	SETDEC	mettre le CPU en mode DECimal
2E0	DSPOFF	éteindre l'affichage
320	DSPTOGGLE	changer le mode d'affichage (ON () OFF)
360	?C RTN	retour de la sous routine si le carry est armé
3A0	?NC RTN	retour de la sous routine si le carry est désarmé
3E0	RTN	retour sans condition

-note : n'éteint pas l'affichage

TABLE 0.C : "DIVERS-INSTRUCTIONS DE CLASSE 0 - TYPE C

CCDE HEX	MNEMONIQUES	NOTES
030	-----	inutilisé
070	N=C ALL	remplacer N par C (tous les bits)
0B0	C=N ALL	remplacer C par N (tous les bits)
0F0	C(·)N ALL	échanger C et N (tous les bits)
130	LDI S&X	charger immédiatement le mot (10 bits) suivant dans C
170	PUSH ADR	mettre l'adresse de C dans la pile
1B0	POP ADR	"éclater" la pile dans le champ d'adresse de C
1F0	-----	inutilisé
230	GOTO KEY	charger PC du buffer (contenu de KEY)
270	RAM SLCT	mettre l'adresse en RAM
2B0	-----	inutilisé
2F0	WRITE DATA	écrire C dans la RAM (56 bits)
330	FETCH S&X	charger le mot situé à l'adresse ADR en C
370	C=C OR A	"OU" logique de C et A
3B0	C=C AND A	"ET" logique de C et A
3F0	PRPH SLCT	sélectionner le périphérique spécifié par C (S & X)

2 - INSTRUCTIONS DE CLASSE 1 : "GOTO / XEQ ABSOLU"

Mot 1 : $\bar{A}7 \bar{A}6 \bar{A}5 \bar{A}4 \bar{A}3 \bar{A}2 \bar{A}1 \bar{A}0$ 01
 Mot 2 : $\bar{A}15 \bar{A}14 \bar{A}13 \bar{A}12 \bar{A}11 \bar{A}10 \bar{A}9 \bar{A}8$ GC
 bits d'adresse //condition
 / GOTO / XEQ

TABLE 1 : INSTRUCTIONS DE LA CLASSE 1

GC	MNEMONIQUES	NOTES
00	?NC XQ	si le carry est désarmé exécuter la sous routine
01	?C XQ	si le carry est armé exécuter la sous routine
00	?NC GO	si le carry est désarmé aller à la sous routine
11	?C GO	si le carry est armé aller à l' adresse absolue

-note : le carry est automatiquement baissé après une instruction qui le teste ou non. Ainsi, ?NC XQ et ?NC GO sont souvent utilisés pour des branchements inconditionnels

Exemple : ADRESSE INSTRUCTION TRADUCTION BINAIRE MNEMONIQUE

0000 201 - 801 10 0000 0001 ?NC GO
 0001 006 - 006 00 0000 0110 -)180

aller à l'adresse 0180

0509 02D - 0B1 00 0010 1101 ?C XQ
 050A 0B5 - 2D1 00 1011 0101 -)2D0B

exécuter l'adresse 2D0B

REMARQUE:après l'exécution de la sous routine 2D0B, le pointeur programme revient à l'instruction d'adresse 050B par une instruction RTN

3 - INSTRUCTIONS DE CLASSE 2 : OPERATIONS ARITHMETIQUES ET LOGIQUES

Mot type : $\overline{I4}$ $\overline{I3}$ $\overline{I2}$ $\overline{I1}$ $\overline{I0}$ $\overline{F2}$ $\overline{F1}$ $\overline{F0}$ $1\emptyset$
 bits instruction de bits champs

Registre C : $\overline{13}$ $\overline{12}$ $\overline{11}$ $\overline{10}$ $\overline{09}$ $\overline{08}$ $\overline{07}$ $\overline{06}$ $\overline{05}$ $\overline{04}$ $\overline{03}$ $\overline{02}$ $\overline{01}$ $\overline{00}$
 digits champ : MS ←----- M -----> | XS | X |
 | ← ADR → |
 | < KY > |

TABLE 2

I :	4 3 2 1 0	MNEMONIQUES	NOTES
	0 0 0 0 0	A=0	effacer le registre A
	0 0 0 0 1	B=0	effacer le registre B
	0 0 0 1 0	C=0	effacer le registre C
	0 0 0 1 1	A()B	échanger les registre A et B
	0 0 1 0 0	B=A	remplacer B par A
	0 0 1 0 1	A()C	échanger les registre A et C
	0 0 1 1 0	C=B	remplacer C par B
	0 0 1 1 1	C()B	échanger C et B
	0 1 0 0 0	A=C	remplacer A par C
	0 1 0 0 1	A=A+E	ajouter B à A
	0 1 0 1 0	A=A+C	ajouter C à A
	0 1 0 1 1	A=A+1	incrémenter A
	0 1 1 0 0	A=A-B	soustraire B de A
	0 1 1 0 1	A=A-1	décrémenter A
	0 1 1 1 0	A=A-C	soustraire C de A
	0 1 1 1 1	C=C+C	doubler C
	1 0 0 0 0	C=C+A	ajouter A à C
	1 0 0 0 1	C=C+1	incrémenter C
	1 0 0 1 0	C=A-C	remplacer C par A-C
	1 0 0 1 1	C=C-1	décrémenter C
	1 0 1 0 0	C=0-C	prendre le complément arithmétique de C
	1 0 1 0 1	C=-C-1	prendre le complément à 1 ou 9
	1 0 1 1 0	?B≠0	armer le carry si B est différent de 0
	1 0 1 1 1	?C≠0	armer le carry si C est différent de 0
	1 1 0 0 0	?A(C	armer le carry si A est inférieur à C
	1 1 0 0 1	?A(B	armer le carry si A est inférieur à B
	1 1 0 1 0	?A≠0	armer le carry si A est different de 0
	1 1 0 1 1	?A≠C	armer le carry si A est different de C
	1 1 1 0 0	RSHFA	(Right SHiFt A) décaler A de 1 digit à droite
	1 1 1 0 1	RSHFB	(Right SHiFt B) décaler B de 1 digit à droite
	1 1 1 1 0	RSHFC	(Right SHiFt C) décaler C de 1 digit à droite
	1 1 1 1 1	*LSHFA	(Left SHiFt A) décaler A de 1 digit à gauche

-note : les opérations prennent la place indiquée par le champ:uniquement.Les ppérations peuvent être en HEX ou DCB selon le mode du CPU. Le mode du CPU est armé par SETHEX (260) ou SETDEC (2A0)'

DEFINITION DU CHAMP

F :	2 1 0	MNEMONIQUES	NOTES
	0 0 0	ER	digit indiqué par R
	0 0 1	S&X	signe de l'exposant et exposant
	0 1 0	RC-	au dessus du pointeur à droite
	0 1 1	ALL	tous les 14 digits
	1 0 0	P-Q	entre les pointeurs
	1 0 1	XS	signe de l'exposant (digit 2)
	1 1 0	M	mantisse uniquement (digit 3 à 12)
	1 1 1	MS	signe de la mantisse uniquement (digit 13)

-note : R signale le pointeur utilisé



4 - INSTRUCTIONS DE CLASSE 3 : PC, SAUT RELATIF

Mot type : $\overline{D6}$ $\overline{D5}$ $\overline{D4}$ $\overline{D3}$ $\overline{D2}$ $\overline{D1}$ $\overline{D0}$ \overline{C} 11
 bits de déplacement /condition

TABLE 3

C	MNEMONIQUES	NOTES
0	JNC *sdd	saut si carry baissé
1	JC *sdd	saut si carry armé

le saut n'est pas effectué si la condition du carry n'est pas remplie
 s signe du déplacement : - vers le haut
 + vers le bas
 dd déplacement : nombre d'instructions sautées

exemple :	ADRESSE	INSTRUCTIONS	MNEMONIQUE
	000D	05B - 163	JNC *+0B
	0025	39B - E63	JNC *-0D
	01D8	3B7 - ED3	JC *-0A

Explication du 1° cas

05B 00 0101 1011 --- table 3
 -----condition 0 = JNC
 000 1011 signe + car le 1°bit est à 0
 0 B
 soit JNC +0B

Explication du 3° cas

3B7 11 1011 0111 --- table 3
 -----condition 1 =JC
 111 0110 signe - car le 1° bit est à 1
 prendre le complément du nombre formé par les 7 premiers
 bits à 2 soit 000 1001 puis ajouter 1 à ce résultat soit :

 000 1010 = 0A
 soit JC -0A

Lethéâtre des microcodes est terminé pour aujourd'hui mais avant de refermer complètement le rideau je à répondre à certaines lettres reçues depuis le JPC n°1

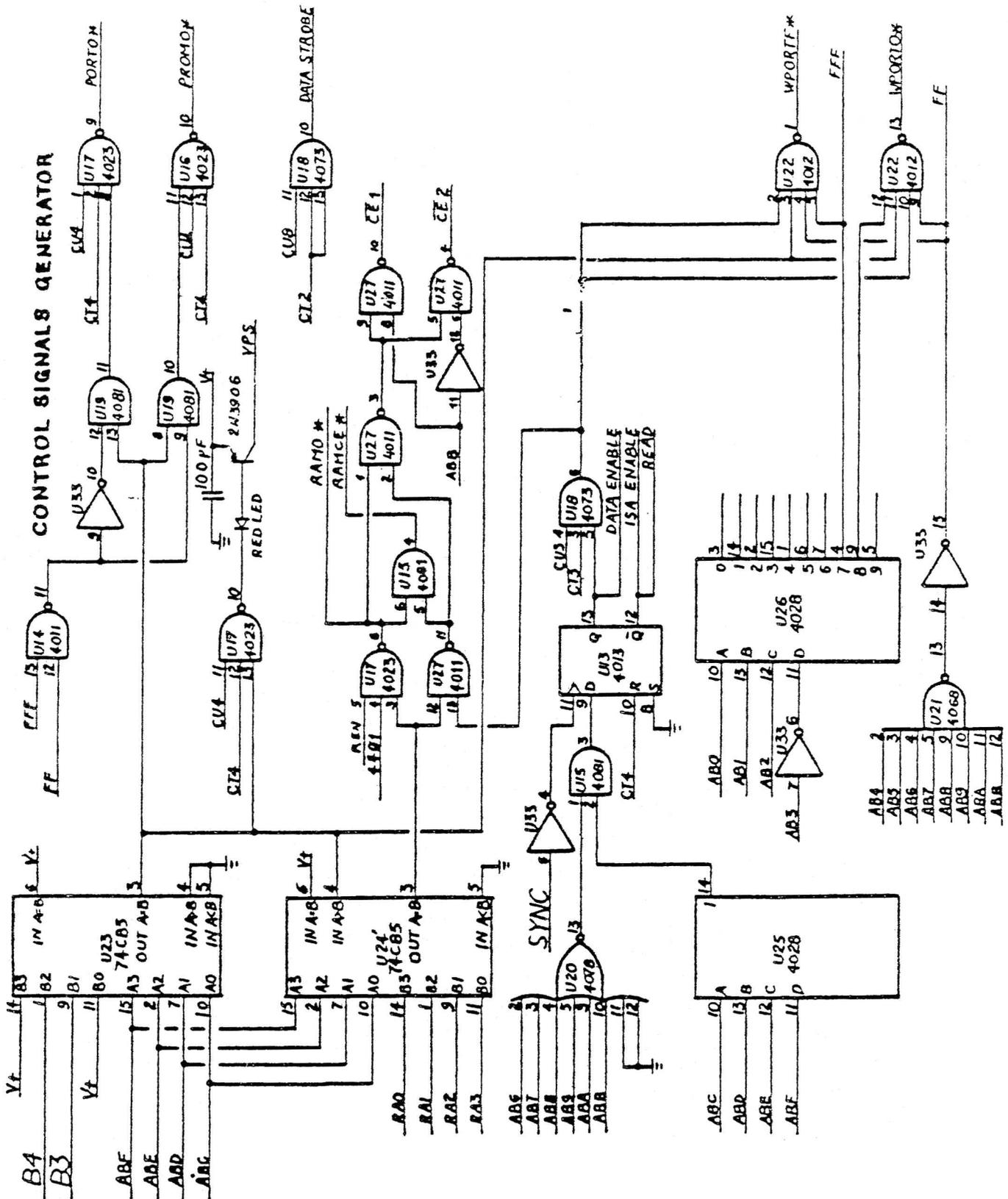
Les différents registres A, B, C, ... ne sont accessibles qu'avec une boîte d'eprom ou simulateur de ROM (du genre du MLDL) car HP n'a pas officialisé le langage machine et ne vous livrera rien sur ce sujet les seules documentations possibles ont été faites par les membres du PPC (PPCAméricain et PPC australie pour ne citer que les plus avancés)

Liste des microcodes PPC
 Liste des microcodes HP (livre de 371 pages appelé VSAM)

PHILIPPE (7858 - 1)

R/S

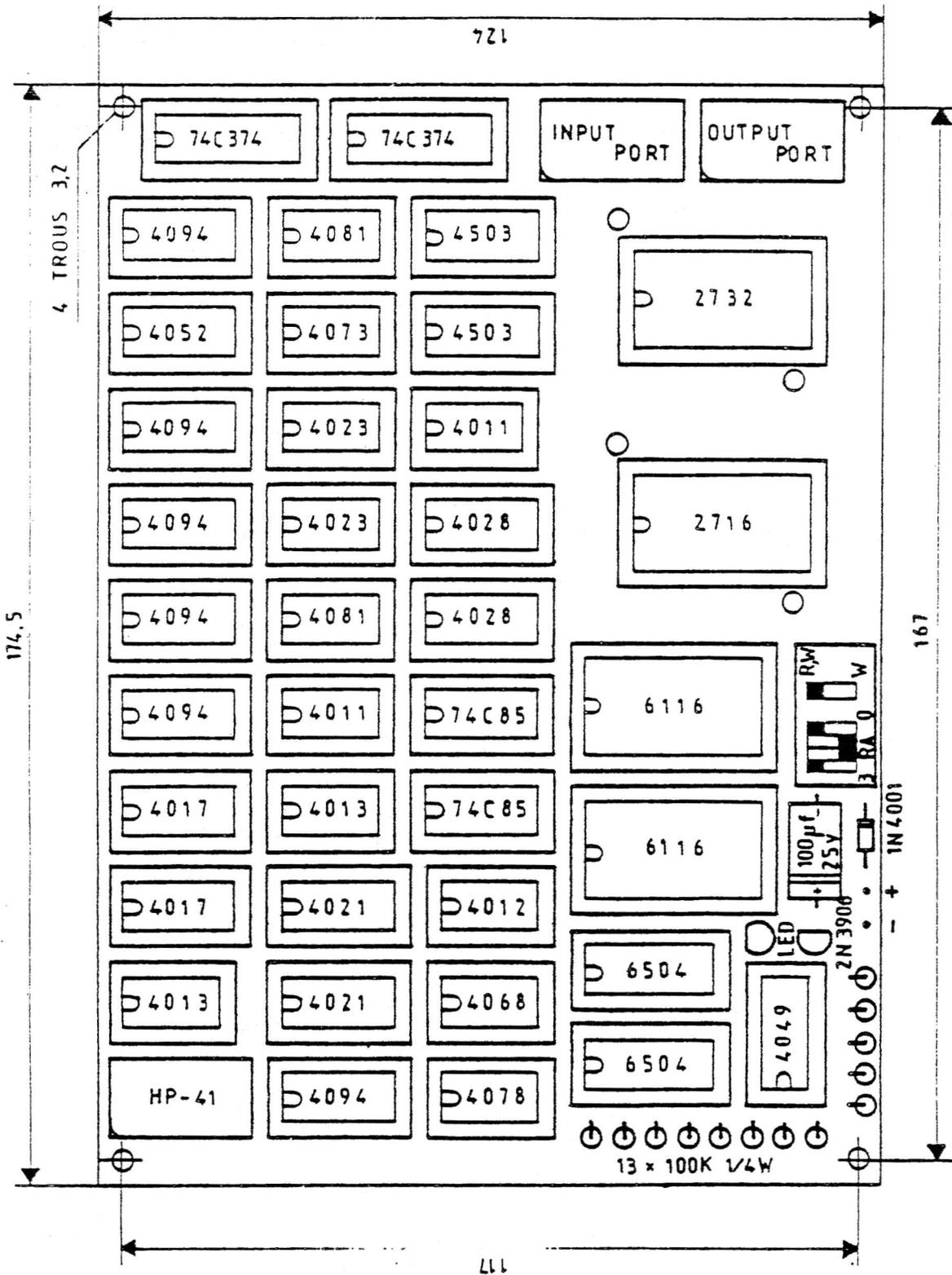




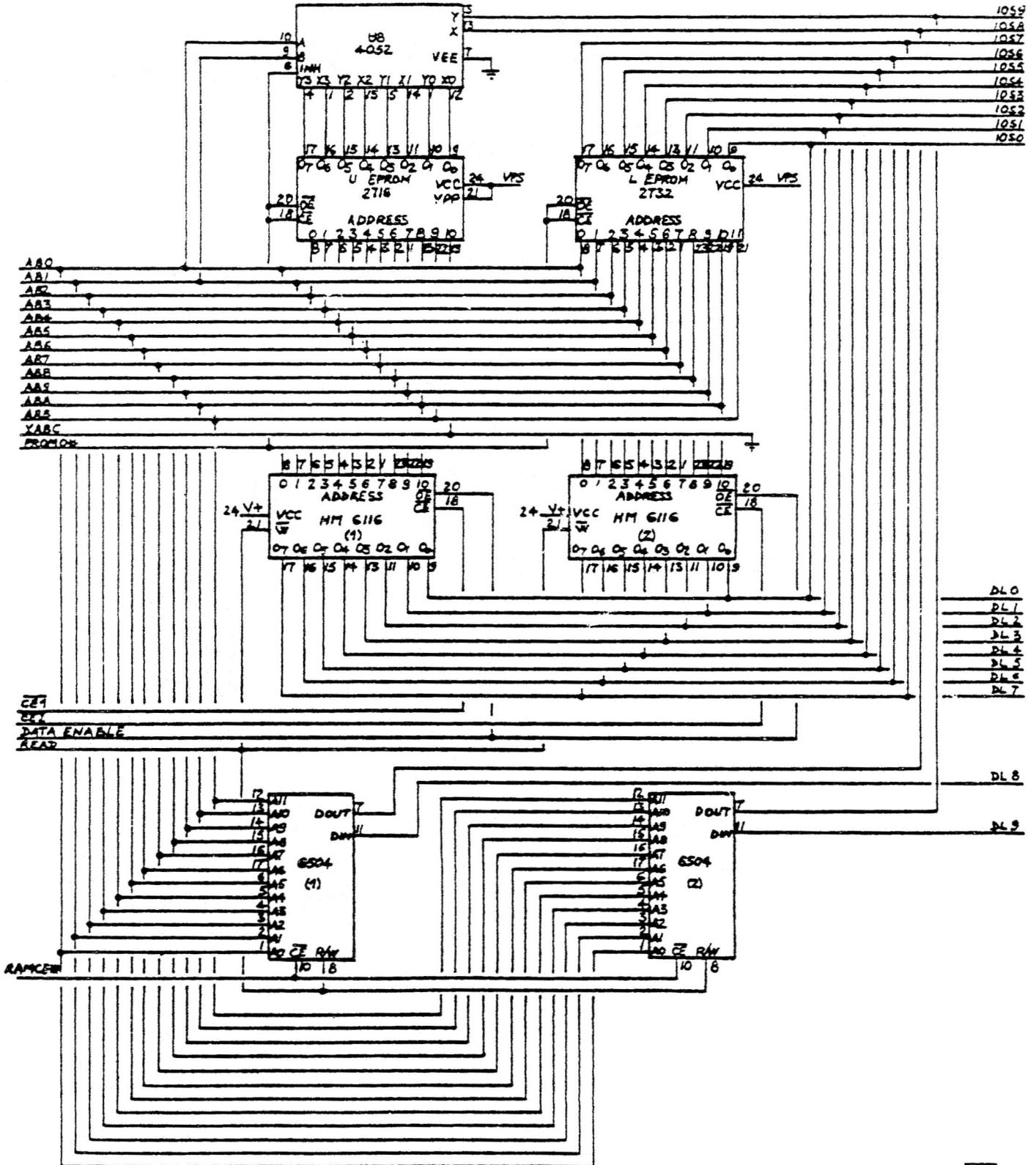
Pour finir le petit théâtre des microcodes vous trouverez ci après 4 pages de description du MLDL (Machine Langage Développement Laboratory) soit la première version d'un simulateur de ROM vous permettant d'étudier le langage machine. Version modifiée par Didier Jehl notre fameux fabricant de la boîte d'eprom (BE 01 4K) ces 4 pages devraient vous permettre la fabrication du MLDL ATTENTION il y a pour plus de mille francs de pièces et le travail est assez ardu la grande majorité des pièces sont très fragiles!!!

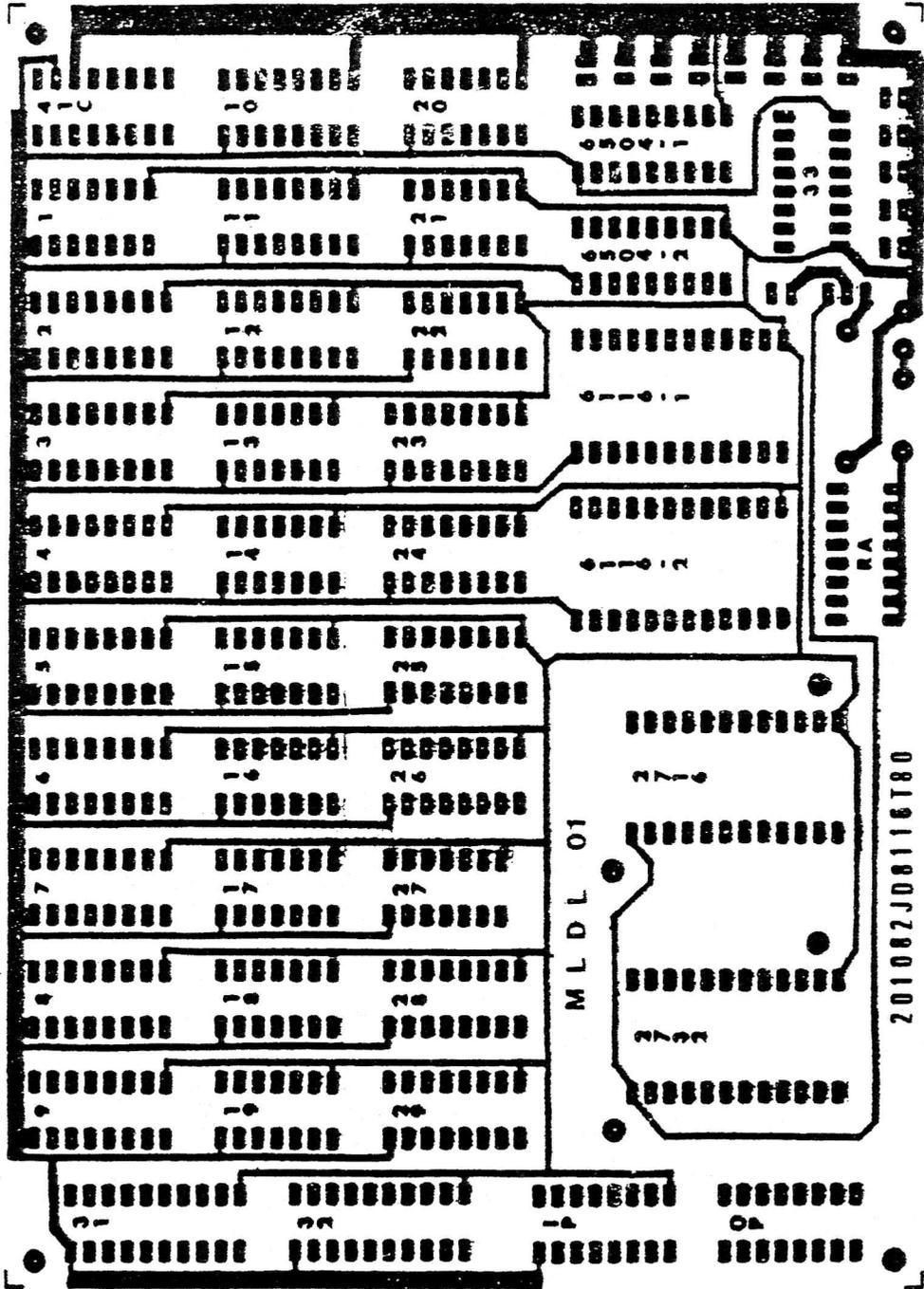
AVIS AUX AMATEURS ...! et bon courage

MLDL: Implantation Mécanique



4K EPROM AND 4K CMOS RAM





201062J08116T80



Board of Directors

Edward J. ...
President

Richard J. Nelson
Vice-President

Richard Schwartz
Secretary

Richard K. ...
Treasurer

Ernie ...
...

February 26, 1983

Philippe Guez (7853)
56 Rue Jean Jacques Rousseau
Paris 75001
France

Dear Philippe,

I will try to answer your questions as I understand them. I do not read or write French, and also have difficulty with hand written letters.

1. PPC ROM. The PPC ROM may be purchased in quantities of 10 or more for \$61.75 each. If you pick the ROM's up at the air port and clear customs they may be shipped by air at the lowest cost. We often use "K" Line Air Service to Europe for othr countries outside the US. The Board of Directors will have to address any proposals regarding manufactured ROMs for you. I doubt that they would allow others to use the mask.
2. ROM Manual. Any translation effort would have to be done by you. If you wish to do this we have no objections, but any commercial activity will have to receive formal agreement from the PPC Board. The PPC trademark and ROM Manual are official copyright of PPC.
3. PPC CJ Articles. You may reprint translated articles from PPC CJ provided: a) they are not sold for profit, b) they reference the source and author and c) PPC and the author receive a copy. (PPC will forward a copy to the author if provided.)
4. The VSAM Listing (371 pages) is available for a printing cost of US \$12.00. It weighs 3 pounds for shipping purposes. It is also copyrighted by PPC.

So far there is no book on 41 M-CODE. The best resource at present is the PPC Southwest Conference Proceedings. See the Chapter Notes Column for comments. Cost is \$3.50, weight is 5 ounces, which would cost \$2.14 to send by air to France.

The new chapter message will be in the January/February (late) Calculator Journal.

If you are serious about 100 + PPC ROMs, I will address the PPC Board for a special price.

Happy programming,

Richard J. Nelson
Richard J. Nelson

2545 West Camden Place, Santa Ana, California 92704 USA
Telephone (714) 754-6226

voici la réponse d'une lettre écrite à R.Nelson

- 1 des prix sont possibles en cas d'une commande groupée de PPC ROM me contacter.
- 2 il nous est possible de faire une traduction du livre de la ROM apres quelque accord. Que tous ceux qui sont interresses et qui seraient en mesure de traduire une partie me contactent
- 3 nous pouvons traduire certains articles du PPC journal apres envoi d'une copie au PPC et à l'auteur
- 4 VSAM est un listing microcode



NOM	PROFESSION - INTERETS	VILLE	PPC n°	PPCPC n°
JEAN REIBEL	ENSEIGNANT 9, square van Klinging 92350	LE PLESSIS ROBINSON		31
PIERRE LAMY	CONSEILLÉ EN INFORMATIQUE 12, rue Carpeaux 92400	COURBEVOIE		32
JEAN-PIERRE BONDU	ETUDIANT PROGRAMME ALLO 1, allée des juchere 78121	CRESPIERES		33
THIERRY BRAVIER	LYCEEN MICROCODE 60, rue du chateau d'eau 75010	PARIS		34
FREDERIC POUPON	ETUDIANT INTERPACKAGES MICROCODE 1, allée Pierre Fresnay 94400	VITRY SUR SEINE		35
FRANCOIS MESTOUDJIAN	ETUDIANT PROGRAMMATION 3, cité d'hauteville 75010	PARIS		36
PIERRE DAVID	ETUDIANT MICROCODE HARD 33, boulevard saint Martin 75003	PARIS		37
JEAN PRUNEL	MEDECIN PROGRAMMATION 6, avenue du bois guimier 94100	ST MAUR		38
DIDIER JEHL	PROF. D'ELECTRONIQUE MICROCODE HARD Appt n°8 9, rue utrillo 62000	ARRAS		39
GABRIEL GIL	ETUD. INFOR. ET ELCT. MICRO., HARD, ECHANGE 44, chemin de barberet 69700	GIVORS		40
BERNARD MORISSON	INGENIEUR ENSEIGNANT EN INFORMATIQUE 22, avenue de Friedland 75008	PARIS		41
CLAUDE FOURNIER	INGENIEUR BATIMENT BATIMENT TRAV. PUBLICS 323, rue de charenton 75012	PARIS		42
FRANCK LEBASTARD	ETUDIANT MATHEMATIQUES PROGRAMMATION 10, rue de la jalousie 35600	REDON		43

CHOC EN RETOUR AU JPC V1N1
=====

PHILIPPE GUEZ 56, rue Jean Jacques Rousseau 75001 PARIS
 LIONEL ANCELET Résidence universitaire A. Camus CH V119 cité scientifique 59650 V. d'ASQ
 LAURENT ISTRIA 18, rue Pierre et Marie Curie 75005 PARIS
 OLIVIER DANCER 39, rue saint Fargeau 75020 PARIS
 JEAN DANIEL DODIN 77, rue cagire 31100 TOULOUSE
 PATRICE DROIN 10, boulevard Malesherbes 75008 PARIS
 ERIC DELAGNES 14, rue Merlin de Thionville 92150 SURESNES
 JACOB DANIEL 4, rue des blanchisseurs 92370 CHAVILLE
 MICHEL MARTINET 17, rue Eugène Henaff 93200 ST DENIS
 CHARLES MEGNIN 40, rue faidherbe 75011 PARIS
 PASCAL MELLY 32, villa des sorbiers 91800 BOUSSY SAINT ANTOINE
 YVES ALAJOUANINE 108, bis rue Hénon 69004 LYON
 PIERRE BENNING 4, allée de la Boissière 92350 LE PLESSIS ROBINSON
 JACQUES DURAND 88, boulevard Ney bâtiment 1E 75018 PARIS
 JEAN PIERRE SANDOZ 44, avenue H. De Montmort 78320 LE MESNIL SAINT DENIS
 FRANKLIN KHAZINE 35, rue de la rangée 92380 GARCHES
 ROBERT SCHWARTZ 141, avenue Carnot 93140 BONDY
 JACQUES VAUCELE "La Chasserie" 35133 ROMAGNE
 DAMIEN DEBRIL 38, rue du 8 mai 1945 59190 HAZEBROUCK
 JOSE PIDIVAL Lieu dit saint Martin 69700 ECHALAS
 PASCAL DUMONT 6, résidence le pré au bois 92420 VAUCRESSON
 YANN DOLHEN 9, rue Béranger 60200 COMPIEGNE
 GUY DUPRAZ 27, rue du faubourg CH 1286 SORAL/GE SUISSE
 EDOUARD COHEN-TANNOUDI 11, rue Richard Wagner 93150 BLANC-MESNIL
 BRUNO LALINE boîte postale 73 6200 GOSSELIES BELGIQUE
 MARCO BENEDETTI 132, rue staempfli 2503 BIENNE SUISSE
 DIDIER MARTIAL 50 rue du disque 75013 PARIS

Ci dessus les adresses des adhérents au PPCPC dont les noms sont parus dans le JPC v1n1